Na prawach rękopisu

INSTYTUT INFORMATYKI, AUTOMATYKI I ROBOTYKI
POLITECHNIKI WROCŁAWSKIEJ

Raport serii: PREPRINTY nr 7/2006

Efekty występowania pamięci krótko- i długoterminowej w pulsującej sieci neuronowej

(rozprawa doktorska)

Tymoteusz BIŁYK

PROMOTOR:

Dr hab. inż. Jan Magott, prof. PWr

Słowa kluczowe:

– pulsujące sieci neuronowe
– algorytmy uczenia
– efekty pamięci długo- i krótkoterminowej
– teoria przetwarzania bez stanów ustalonych

WROCŁAW
MARZEC 2006

Chciałbym podziękować prof. Janowi Magottowi za opiekę nad pracą, dr Urszuli Markowskiej-Kaczmar za swobodę w doborze modelu, prof. Ryszardowi Tadeusiewiczowi za internetowe konsultacje, prof. Ludmile Borodulin-Nadzieji za selekcję oraz wyjaśnienie faktów fizjologicznych.

Spis treści:

Wykaz oznaczeń i skrótów:

AND – operator binarny lub logiczny "i"

ADP – kwas adenozynodifosforowy

EEG - elektroencefalografia

FFT – szybka transformacja Fouriera

FSM- maszyna skończonych stanów

HCM- zespoły komórek Hebba

IAF- model neuronu pulsującego

IMPL – nazwa modeli sieci neuronowych pierwszego symulatora w języku Borand C++

LSM – maszyna ciekłych stanów

MDI- aplikacja wykonana w technice obsługi wielu dokumentów

OCO – częstotliwość odpaleń neuronu z ostatniego okresu pomiarowego

OR – operator binarny lub logiczny "lub"

PNN – pulsująca sieć neuronowa

PRA – mechanizm samoregulacji obejmujący próg, refrakcję i moment aktywacji

PRISM – jeden z symulatorów sieci pulsujących neuronowych

RTCWSS – przetwarzanie danych w czasie rzeczywistym bez dążenia do osiągnięcia stanu ustalonego

SFC – łańcuchy synchronicznych pobudzeń

SCO – średnia częstotliwość odpaleń neuronu

SNN – nazwa modeli sieci neuronowych symulatora obsługującego tryb chroniony procesora w języku Visual C++

SOM- samoorganizowalna mapa, technika neuronowa

XOR – operator negacja “albo”

ZSCO = SCO-OCO – zmiana częstotliwości pracy neuronu

1. Wstęp

Natura często zadziwiała badaczy swoją doskonałością. Na szczególną uwagę zasługuje układ nerwowy, który pomimo szybkiego rozwoju informatyki jest obecnie niedoścignionym ideałem. Fascynacja jego modelowaniem zaczęła się dość dawno, jednakże często spektakularne sukcesy oraz potrzeba zastosowań technicznych sprowadzała badania na obszary odległe wzorcowi (np. burzliwy rozwój metod back-propagation). Nienaturalne założenia dotyczyły większości cech sieci neuronowych: budowy neuronu, mechanizmu pracy i uczenia neuronu, struktury sieci, rodzaju przesyłanych sygnałów międzyneuronowych, formy wprowadzanych danych wejściowych oraz wymuszanych odpowiedzi. Świadomie wykonywane posunięcia odchodzenia od modelowania biologicznego modelu neuronu często prowadziły w „ślepe uliczki", co zamrażało prace na długie lata, po czym startowano z innego punktu rokującego określone techniczne nadzieje, zatem sytuacja się powtarzała.

Jednakże równolegle prowadzone są biocybernetyczne badania z zakresu neurofizjologii polegające na budowaniu możliwie wiernego modelu neuronu i sieci neuronowej. Sama neurofizjologia dostarcza nam ogromnej masy faktów, często pozostających zupełnie bez związku z przetwarzaniem informacji w neuronie. Naturalny neuron posiada bowiem wiele równoległych procesów i nie wszystkie z nich są związane z przetwarzaniem informacji np. funkcje metaboliczne komórki. Modelowanie cech nie posiadających jakichkolwiek związków z przetwarzaniem informacji jest niepożądane w tym przypadku, ponieważ następuje zwykle wydłużenie czasu symulacji, jest potrzebne użycie większej mocy obliczeniowej, oraz powoduje przesłonienie istoty modelu. Podstawowym zadaniem jest więc selekcja znanych faktów dotyczących neuronów, ich dekompozycja w modelu oraz sprawdzenie podczas symulacji. Bazowym założeniem symulowanych modeli jest fakt, iż odpowiedź aktywnego neuronu jest pojedynczym impulsem o stałej maksymalnej amplitudzie. Modele tego typu zostały określone w ostatnich latach etykietką pulsujących sieci neuronowych oraz nastąpił silny ich rozwój, pomimo że nie mają obecnie zbyt

wielu zastosowań technicznych. Zjawisko to świadczy o tym, że jest widziana szansa jakościowego skoku w przetwarzaniu informacji przy pomocy tej techniki. Pierwszorzędnymi właściwościami tego podejścia jest bardzo wysokie zrównoleglenie procesów przetwarzania, wykorzystanie informacji zawartych w pojedynczych impulsach, gubionych podczas operowania na danych pochodzących tylko ze statystyk aktywności danego wyjścia (np. perceptron bazuje na tej zasadzie), możliwość wykorzystania chwilowego stanu sieci i układu aktywnych neuronów do modyfikowania procesu przetwarzania oraz bardzo krótkiego czasu odpowiedzi takiego systemu na zdarzenia krytyczne czasowo. Kilka symulacji przeprowadzonych w tej pracy potwierdza te właściwości.

Patrząc na prowadzone prace z pewnej perspektywy, oraz biorąc pod uwagę fakt dostarczania przez fizjologów układu nerwowego ogromnej masy nieuporządkowanych danych, zależności i korelacji pomiędzy różnymi procesami wewnątrz neuronu i sieci, część badaczy utrzymuje stanowisko, że nie da się stworzyć modelu układu nerwowego idąc od tej strony (bottom-up), pomimo że teoretycznie ta metoda umożliwia wykrywanie własności na coraz wyższych poziomach organizacji sieci. Przeszkodą jest właśnie ogrom różnych danych i wzajemnych zależności, często jeszcze w fazie niezweryfikowanych hipotez, które w tej metodzie stawałyby się elementami testowanego systemu, aby zapewnić przebadanie całej przestrzeni możliwych hipotetycznych modeli. Proponują oni natomiast obserwację poszczególnych partii mózgu, określenie ich funkcji oraz wzajemnych połączeń, z kolei kreowanie w miarę spójnych hipotez, które powinny być w następnym etapie poddawane weryfikacji np. poprzez symulacje. Podejście to wydaje się metodologicznie dobre oraz ma jedną zasadniczą zaletę: posiadając hipotezę możemy określić jakich efektów się spodziewamy oraz możemy wcześnie wykonać aparat to mierzący. Wadą tej metody jest niewątpliwie trudność tworzenia hipotez, w szczególności spójnych.

Światowa literatura na dzień rozpoczęcia badań w dość skromnym zakresie porusza temat powstawiania w pulsującej sieci neuronowej pamięci długo- lub krótko- terminowej. Nie znalazłem żadnego opracowania opisującego kompletny i weryfikowalny model wykształcający pamięci długo– i krótko– terminową w czasie równoległego procesu pracy i uczenia z lokalną regułą uczenia. Samo pojecie pulsującej sieci neuronowej nie było zbyt

popularne do 1996 roku. W wielu pracach odnoszono się do pojedynczych neuronów o takiej konstrukcji nazywając je odpalającymi się lub pulsującymi. Same prace również dotyczyły zwykle poszczególnych neuronów, ich zachowania oraz możliwości reprezentowania się danych w tego typu sieciach. Można przyjąć, że bardzo bliska przedmiotowi praca ukazała się w 1998 roku [138] i bardziej przypominała sieć Hopfielda z neuronem wzorowanym na pulsującym. Sieć pracowała w trybie synchronicznym, ale nadano już właściwą powagę dynamicznej reprezentacji informacji, jaką były tam wyłącznie sygnały międzyneuronowe, dokonano analizy procesów z nimi związanych i przypisano im funkcję pamięci. Sieć nie przetwarzała kolejnych danych wejściowych podczas pracy, choć autor juz dwa lata wcześniej brał taką teoretyczną możliwość pod uwagę [114], oraz nie miała algorytmu uczenia zdefiniowanego na pracującej sieci.

Praca [4] choć nie odnosi się bezpośrednio do pamięci, zawiera analizę wielu czynników z nią związanych: wpływ połączeń rekurencyjnych, opis dynamiki sieci, metody modyfikacji synaps i ich wpływ na całość działania sieci. Autorzy zwracają dużą uwagę na zgodność modelu z biologicznym wzorcem, a teoretyczne wnioski popierają wynikami z symulacji. Założenia tego opracowania są spójne z podejściem IMPL.

Model kory mózgowej został omówiony w dość obszernej publikacji [21]. Bazuje on na synchronicznych łańcuchach pobudzeń (SFC) w przeciwieństwie do modeli zespołów komórek Hebba (HCA). Zawierał on wiele pożądanych czynników jak np. stałą średnią częstotliwość pracy sieci (przy przyjętych 10 tys neuronach) i jednocześnie stany synchroniczne, stabilność pracy, algorytm modyfikacji wag, a przede wszystkim konkurencyjność poszczególnych synchronicznych łańcuchów pobudzeń w walce o przyporządkowanie danego wzorca do klasy reprezentowanej przez swój łańcuch. Pomimo wielu podobnych elementów (w tym np. podobnego kroku symulacji) i obszerności materiału dotyczącego tego modelu nie udało się zweryfikować go doświadczalnie w tej pracy.

Celem pracy jest zbudowanie modelu neuronu pulsującej sieci neuronowej, bazującego na wytycznych neurofizjologicznych, w której zostaną zaobserwowane zjawiska odwzorowywania sygnału wejściowego w strukturze

sieci. Zapisana w ten sposób informacja powinna być użyteczna w procesie przetwarzania kolejnych sygnałów np. pod względem klasyfikacji.

Teza pracy: Istnieje model neuronu spełniający wymagania podane w celu pracy. Kodowanie wzorca w pamięci długo- i krótko- terminowej wymaga zróżnicowania funkcji wielu neuronów w sieci i ich współpracy (synergii), pomimo iż posiadają one bardzo podobne sygnały wejściowe.

Zakres pracy to symulacja komputerowa budowanych modeli, która jest podstawowym kryterium oceny sprawności działania sieci w tej pracy oraz potwierdzeniem odpowiedniego wyboru zbioru modelowanych cech. Prace nad modelami mogły przebiegać kilkoma torami, dlatego w rozdziale drugim została omówiona metodologia podejścia do przedmiotu tej pracy oraz motywacja jej wyboru. Rozdział trzeci zawiera zebrane z różnych źródeł informacje na temat budowy oraz działania naturalnego neuronu, omówione pod kątem wykorzystania ich w modelach sieci neuronowej. Bazowe założenia modeli, przyjęta platforma sprzętowa symulatora oraz budowa samego symulatora zawiera rozdział czwarty. Przedstawia on również system interfejsów pomiędzy jądrem sztucznej sieci neuronowej a środowiskiem modelu. Rozdział piąty zawiera wyniki i opisy przeprowadzonych symulacji. Ze względu na swą obszerność został podzielony na trzy podstawowe podrozdziały dotyczące odpowiednio prostych modeli binarnych, modeli uwzględniających symulację ładunków postsynaptycznych oraz modeli uwzględniających symulację ładunków postsynaptycznych o dużej liczbie neuronów. Nie bez przyczyny te dwa ostatnie typy modeli omówione są osobno, gdyż występują w nich zupełnie inne efekty synergii pracy neuronów, do ich oceny zastosowano nowe mechanizmy, a samo drastyczne zwiększenie liczby neuronów wymagało zbudowania nowego symulatora. Rozdział szósty odnosi efekty przedstawione w poprzednim rozdziale do literatury współczesnej. Pracę kończy podsumowanie oraz spis literatury.

2. Metodologia pracy

Zakres niniejszej pracy, uwzględniając analizę, zaprojektowanie oraz implementację specjalizowanych symulatorów, mieści się w pojęciu informatyki. Sam modelowany obiekt, jakim jest neuron lub grupa neuronów interaktywnie reagująca na bodźce z otoczenia oraz przesyłające dane do niego wkracza na obrzeża biocybernetyki, która jest nauką o sterowaniu, w szczególności samosterowaniu, układów biologicznych. Dodatkowo można stwierdzić, iż zakres biologiczny został tutaj zawężony tylko do neurofizjologii. Jak wiemy, ta część biocybernetyki jest nauką przyrodniczą opartą na obserwacji i formalnie nie ma możliwości udowodnienia obserwacji. Reguły ustala się na podstawie odpowiednio dużej liczby przeprowadzonych doświadczeń oraz interpretacji wyników przy pomocy przyjętych aparatów statystycznych. Postępowanie takie nie jest jednak w dalszym ciągu formalnym dowodem. Z drugiej strony, informatyka zawiera elementy teorii, jednak dotyczą one w większości bliskich matematyce numerycznych metod przetwarzania lub przetwarzania symbolicznego. Samo tworzenie oprogramowania, jakim w tej pracy jest budowanie i kompilowanie specjalizowanego symulatora dla każdego badanego modelu, jak do tej pory nie ma dobrej teorii. Użyteczne stają się w tym przypadku różnego rodzaju metodologie programowania, jednakże one też nie dają możliwości formalnego udowodnienia poprawności programu, w tym przypadku - symulatora. Oczywiście istnieją metody formalne udowadniania poprawności oprogramowania, jednakże przy stosowanym tu stopniu złożoności są zupełnie nieprzydatne, czego potwierdzenie uzyskujemy w bardziej zaawansowanych metodologiach. Z tych dwóch powodów wynika, iż dowód formalny działania takiego systemu jest praktycznie niemożliwy, szczególnie jeśli miałby dotyczyć jakiejś ogólnej klasy modeli (trudności dodatkowo z wyspecyfikowaniem i formalnym opisem cech wspólnych i niezależnych od cech jednostkowych modelu) lub każdego modelu z osobna (duża ilość dowodów).

Z drugiej strony, gdyby został stworzony wysoce pojemny, szybki w działaniu oraz optymalnie wykorzystujący swoją historię model pamięci długo-

i krótko- terminowej i jedynym jego problemem byłby jego formalny dowód, oznaczałoby to niebywały sukces. Jak do tej pory model taki nie istnieje, a cały ciężar badań pulsujących sieci neuronowych skupia się nad doborem odpowiednich cech do modelowania, co jest celem tej pracy.

Ramy obszaru, w którym prowadzone są badania, są zatem ograniczone z jednej strony zbiorem znanych cech neuronu i sieci, które przekładane są do modelu. Z drugiej strony znając pożądany efekt można skonstruować aparat badający istnienie badanej przez nas cechy wyższego poziomu, którą w tym przypadku jest pamięć długo- i krótko- terminowa. Ponieważ istnienie pamięci może być stwierdzone wtedy, kiedy ona funkcjonuje (a w szczególności - funkcjonuje dobrze), a prace polegają właśnie na jej skonstruowaniu, zatem na podstawie danych zebranych tym ww. aparatem badający nie mógłby podjąć decyzji ustalającej kierunek dalszych badań, szczególnie w najistotniejszych, pierwszych etapach konstruowania modelu. Nie jest możliwe zatem prowadzenie badań tą metodą, że względu na zbyt wiele stopni swobody modelu. W tym momencie została przyjęta następująca hipoteza, która przyniosła efekty w praktyce. Otóż, jeden neuron jest w stanie zawrzeć znikome ilości informacji, ponieważ siła sieci polega na ilości połączeń międzyneuronowych, i tam należy się spodziewać zasobów pamięci. Żeby tego typu pamięć mogła powstać, neurony muszą się komunikować ze sobą oraz ich zachowanie musi wykazywać jakikolwiek porządek. Mówimy w tym przypadku o efekcie synergii neuronów. Jak pokazują doświadczenia, przy stosunkowo małej ilości neuronów (do 10 000 szt.), jedna z trzech podstawowych metod bazuje na analizie wykresu średniej częstotliwości aktywnych neuronów, a w szczególności opiera się na pewnych okresowościach w nim. Ten wyznacznik jest podstawowym, automatycznym testem, który wraz z obserwacją matrycy wag oraz matrycy potencjałów postsynaptyczynch, o ile dany model takowe posiada, przesądza o jakości modelu. (rozdział 5.4)

Obszar badania ograniczyliśmy zatem do zbioru cech biologicznych neuronu, implementowanych w modelu celem osiągnięcia maksymalizacji wartości mierzonego parametru (np. wariancji wartości średniej częstotliwości odpaleń neuronów pulsującej sieci jako funkcji czasu), wykrywanego określonym narzędziem (np. modułem przeliczająco-wizualizującym zbudowanego symulatora). Dodatkowo zbiór modelowanych cech został

rozszerzony o hipotetyczne cechy (np. normalizacja wag synaptycznych, gromadzenie statystyki zmiany częstotliwości pulsowania oraz wykorzystywanie jej podczas procesu uczenia, założenia warunkowej modyfikacji wag synaptycznych itp.), które neuron może posiadać, a nie są obecnie znane (w podobny sposób wykryto np. istnienie synaps hamowania, co zostało później potwierdzone przez neurofizjologów).

W pracy użyte jest połączenie tych dwóch metod, wstępującej oraz zstępującej. W pierwszej znane fakty fizjologiczne implementowane są w modelu oraz przeprowadzana jest symulacja. Opcja ta jest wybierana, gdy nie jesteśmy w stanie nic przypuszczać o dalszych efektach prac. Druga metoda, polegająca na doborze symulowanych elementów pod kątem osiągnięcia celu / hipotezy, jest używana w etapach, w których model wymaga pewnego poprawienia i jesteśmy w stanie wnioskować o brakującej albo nadmiarowej cesze ze zbioru symulowanych elementów.

Słuszność podjętego podczas rozpoczynania doświadczeń tej pracy kroku można po czasie uwiarygodnić identycznymi stwierdzeniami czołowych badaczy tej dziedziny T. Natschlägera, H. Markrama, i W. Maassa 2003 [141].

Praca, choć głównie wykorzystuje fizjologię i w mniejszym stopniu anatomię neuronu, przekracza granice biocybernetyki wkraczając na pole informatyki nie tylko na obszarze konstrukcji symulatora. Dzieje się tak również w momencie, gdy model zgodnie z celem pracy zaczyna wykazywać właściwości związane z zaistnieniem pamięci krótko- lub długo- terminowej. Obserwowanie pamięci odbywa się podczas sprawdzania jej funkcjonowania, czyli jej zapisywania, odczytywania, przechowywania, gromadzenia czy przetwarzania danych, co stanowi kluczowe zagadnienia informatyki.

Przy pracy z sieciami neuronowymi istnienie pamięci jest związane z określoną funkcjonalnością np. użyciem sieci jako klasyfikatora. Dla usystematyzowania, zawężenia i ukierunkowania badań w powyższym obszarze zostały przyjęte następujące cztery kryteria definiujące zgodność sygnału wejściowego z wyuczonym wzorcem:

1) Definicja pokrycia zupełnego wzorca

$$\exists y \in C \forall n \in \{1,2,\ldots N\} \quad F(P_n, X_{y+n}) = F(P_n, P_n) \qquad (1)$$

gdzie:

N - ilość wektorów wzorca

P_n - *n'ty* wektor wzorca

P – wzorzec

C – zbiór liczb całkowitych

X_t - wektor sygnału wejściowego w chwili *t*. Czas w modelu jest zdefiniowany dyskretnie. Kolejna wartość całkowita *t* oznacza przesunięcie się w modelowanej sieci o ok. 1,85 ms.

t,y - wartości należące do dziedziny czasu

$F(P_n, X_t)$- funkcja pokrycia *n'tego* wektora wzorca przez wektor wejściowy w chwili *t* zgodnie z następującą definicją:

$$\begin{gathered} P = (P_1, P_2, P_3, \ldots, P_N) \\ P_n = (a_{n,1}, a_{n,2}, \ldots, a_{n,L}) \\ X = (\ldots X_{-2}, X_{-1}, X_0, X_1, X_2 \ldots) \\ X_n = (x_{n,1}, x_{n,2}, \ldots, x_{n,L}) \\ x, a \in \{0,1\} \\ def.\ F(P_n, X_m) = \sum_{i=1}^{L} \begin{cases} 1 & \Leftrightarrow a_{n,i} = 1 \ \wedge x_{m,i} = 1 \\ -1 & \Leftrightarrow a_{n,i} = 1 \ \wedge x_{m,i} = 0 \\ 0 & \Leftrightarrow a_{n,i} = 0 \end{cases} \end{gathered} \qquad (2)$$

Funkcja *F* wykazuje pewną niesymetryczność związaną z mniejszym wpływem na jej wartość pól przyjmujących we wzorcu wartość zero. Wyjaśnienie tego założenia znajduje się w rozdziale 5.1.1, a jego interpretację zawiera rys. 14.

Zgodnie z definicją (1) sygnał wejściowy musi być idealnie zsynchronizowany w czasie przetwarzania całego wzorca oraz zachowywać zgodność kolejnych wektorów w myśl funkcji *F*.

2) Definicja pokrycia wzorca na zdefiniowanym poziomie

$$\exists y \in C \, \forall n \in \{1,2,\ldots N\} \quad F(P_n, X_{y+n}) > cF(P_n, P_n) \qquad (3)$$
$$c - \textit{ułamek właściwy}$$

Definicja (3) w przeciwieństwie do (1) nie wymaga pełnej zgodności poszczególnych wektorów sygnału wejściowego z wzorcem, lecz akceptuje poziom wyznaczony przez parametr *c*. Synchronizacja i kolejność wektorów musi być w pełni zachowana, podobnie jak w (1).

3) Definicja pokrycia wzorca przy określonej tolerancji nieliniowości podstawy czasu lub deformacji sygnału wejściowego w wymiarze czasu

$$\exists y \in C \;\; \exists e \in \{1,2,\ldots N\} \forall n \in \{1,2,\ldots N\} \exists g \in \{1,2,\ldots e\} \quad F(P_n, X_{y+g}) = F(P_n, P_n) \qquad (4)$$

Definicja (4) pozwala sygnałowi wejściowemu na zmiany w synchronizacji na odcinkach czasowych wyznaczonych przez parametr *e*. Dopuszczalna jest również zmiana kolejności wektorów w tym zakresie, natomiast zgodność odpowiednich wektorów musi być pełna.

4) Definicja pokrycia wzorca przy określonej tolerancji nieliniowości podstawy czasu lub deformacji sygnału wejściowego w wymiarze czasu przy założeniu zdefiniowanego stopnia pokrycia wzorca

$$\exists y \in C \;\; \exists e \in \{1,2,\ldots N\} \forall n \in \{1,2,\ldots N\} \exists g \in \{1,2,\ldots e\} \quad F(P_n, X_{y+g}) > cF(P_n, P_n) \qquad (5)$$
$$c - \textit{ułamek właściwy}$$

Ostatnia definicja (5) zgodności sygnału wejściowego z wzorcem wydaje się najlepiej opisywać rzeczywistość z wymienionych powyżej. Dopuszczone są przesunięcia czasowe sygnału lub nierównomierności do poziomu wyznaczonego parametrem *e*, oraz stopień zgodności poszczególnych wektorów może być mniejszy niż w (4) i definiowany parametrem *c*.

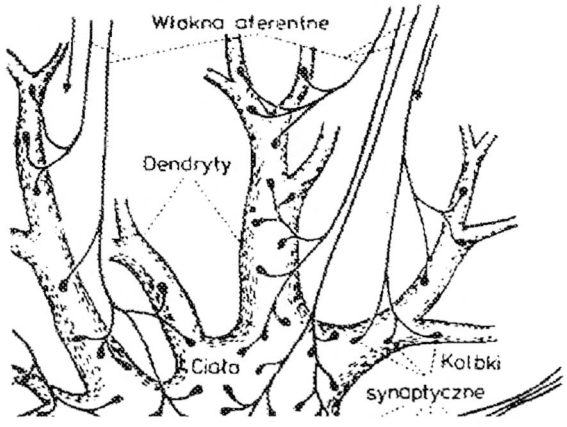
W praktyce parametr *e* może przyjmować nawet dość duże wartości, ale mniejsze od *N*, o ile *y* jest momentem rozpoczęcia sygnału podobnego do wzorca. Założenie, że *y* wskazuje środek sygnału podobnego do wzorca jest

formalnie precyzyjniejsze. Jeśli występuje zmiana kolejności dwóch wektorów, ich odległość nie powinna przekraczać 8 ms.

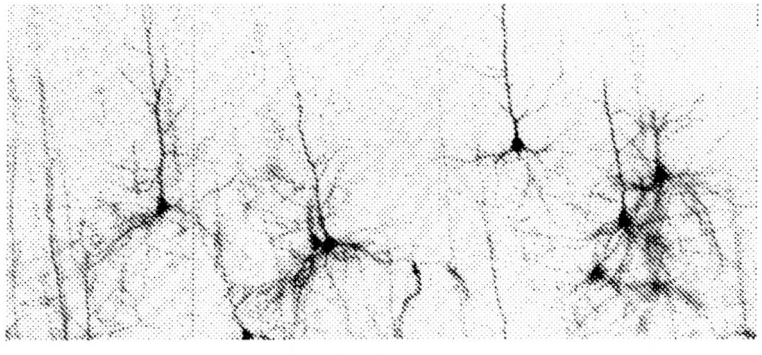
3. Elementy budowy neuronu oraz opis zjawisk fizjologicznych związanych z tematem pracy

3.1 Elektrofizjologia oraz budowa neuronu

Podstawowym elementem budulcowym układu nerwowego jest neuron. Jest on komórką, czyli grudką cytoplazmy zawierającą jądro, otoczoną błoną komórkową. (rys. 1) Błona komórkowa jest cienką submikroskopową warstwą białkowo-lipidową. Ciała tłuszczowe i białka tworzą bardzo cienką molekularną błonę, która ma zdolność do wybiórczego przepuszczania pewnych substancji. Możliwe jest więc zróżnicowanie stężeń jonów wewnątrz komórki względem otoczenia. Pierwszoplanowe znaczenia mają anion chlorkowy Cl^- oraz kation sodowy Na^+. Kation sodowy występuje w środowisku zewnętrznym komórki w stężeniu dziesięciokrotnie większym niż we wnętrzu. Jon potasowy K^+ natomiast występuje w stężeniu trzydziestokrotnie większym we wnętrzu komórki niż na zewnątrz. Stężenia są utrzymywane przez tzw. pompę sodową. Wydatek energetyczny komórki jest konieczny nie tylko, aby utrzymać gradient stężeń, ale również gradient elektrochemiczny. Jest on



Rys.1. Ciało komórki otoczone błoną komórkową z zaznaczonymi kolbkami synaptycznymi

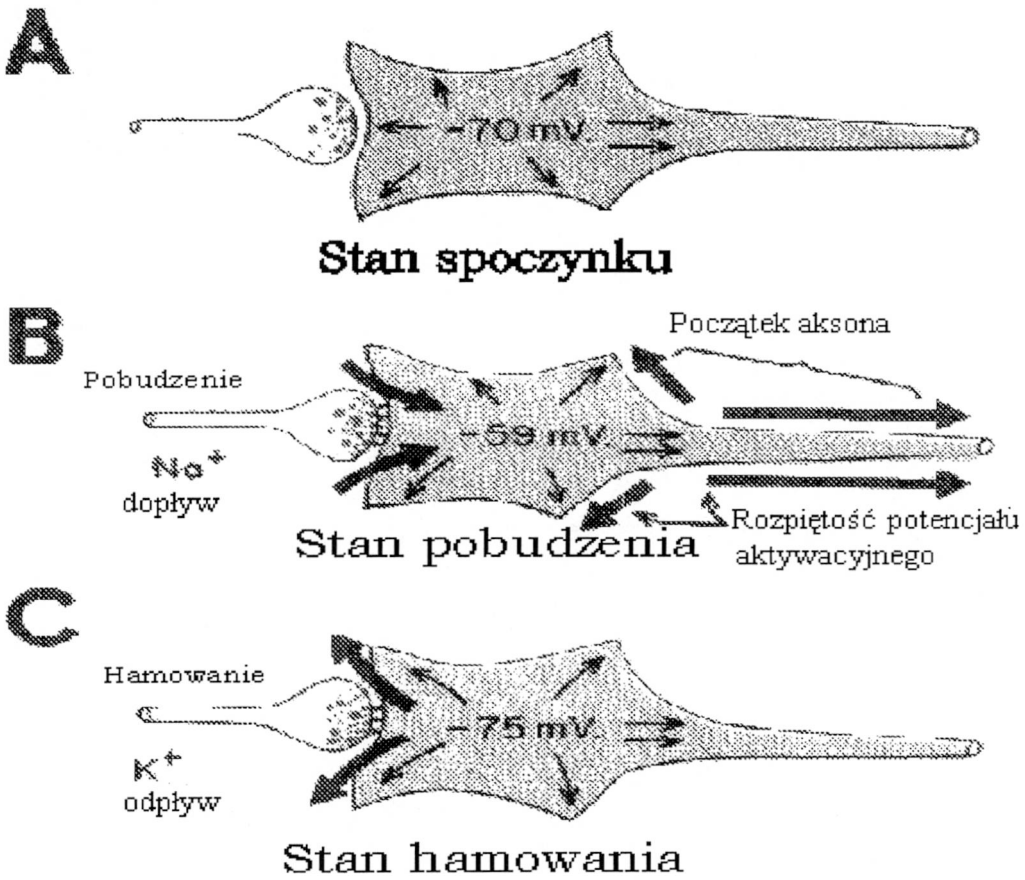

Rys.2. Neurony uwidocznione barwieniem metodą Golgiego

związany z następującą zależnością: we wnętrzu komórki znajdują się białka, które posiadają stosunkowo liczne grupy kwaśne -COOH. Po dysocjacji w roztworze wodnym jon H^+ odłączony od grupy karboksylowej może zgodnie z gradientem stężeń przenikać przez błonę komórkową, ponieważ jest bardzo mały. Może być on wyparty również przez jon sodowy lub potasowy. Jon potasowy w porównaniu z uwodnionym jonem sodowym jest na tyle mały, że może przenikać przez błonę komórkową. Jony potasowe zobojętniają ujemne grupy karboksylowe we wnętrzu komórki, tworząc sole potasowe białek. Gradient stężeń natomiast powoduje tendencje do uciekania tych jonów z komórki. Tak więc kationy potasowe oscylują przez błonę komórkową w efekcie wypadkowej sił gradientu stężeń oraz gradientu elektrochemicznego, tak że pewna ilość jonów K^+ jest zawsze poza komórką. We wnętrzu komórki względną przewagę posiadają więc ładunki ujemne $-COO^-$. Wykazywany potencjał spoczynkowy pomiędzy wnętrzem komórki a powierzchnią zewnętrzną błony jest równy ok. -70 mV. Ponieważ ta różnica wywołana jest oscylacją kationów K^+, mówimy o baterii potasowej.

3.2 Zmiany potencjału komórki

Potencjał spoczynkowy neuronu ulega niewielkim wahaniom związanym z metabolizmem komórki. Jednak może on znacznie zmienić swoją chwilową wartość pod wpływem pracy synaps, znajdujących się na powierzchni komórki. Substancja chemiczna zwana neuromediatorem może spowodować zmianę stanu błony i tym samym całej komórki. Wyróżnić można dwa typy neuromediatorów: hamujące i pobudzające. Mediator hamujący powoduje uszczelnienie błony komórkowej i w rezultacie różnica potencjałów pomiędzy wnętrzem i otoczeniem jeszcze bardziej wzrasta. Przeciwnie dzieje się pod wpływem mediatora pobudzającego. Błona komórkowa staje się przepuszczalna dla jonów sodowych i w rezultacie różnica potencjałów maleje. Jeśli pod wpływem działań mediatorów synaptycznych potencjał komórki zmieni się z -70 mV do ok. -59 mV (rys. 3 A,B), zwanego potencjałem generatorowym, to błona komórkowa zmienia swoje właściwości separacyjne. Staje się ona całkowicie przeźroczysta dla kationów w związku z występowaniem dodatniego

sprzężenia zwrotnego: zwiększenie potencjału komórki powoduje wzrost przepuszczalności błony i odwrotnie, czyli wzrost przepuszczalności błony powoduje zwiększenie potencjału komórki. W wyniku tego procesu komórka osiąga potencjał iglicowy ok. +50 mV. Potencjał ten bardzo szybko spada, aż osiągnięty zostanie stan poniżej potencjału pierwotnego. Stan ten zwany jest hiperpolaryzacją. Cały proces pobudzenia jonowego wraz z okresem hiperpolaryzacji tworzącej refrakcję bezwzględną błony trwa około 2 ms, czyli w ciągu sekundy może powstać nawet 500 potencjałów iglicowych, średnio jednak powstaje ich 200-300.



Rys.3. Trzy podstawowe procesy pracującej komórki nerwowej. A- stan spoczynku, B- stan pobudzenia C- stan hamowania

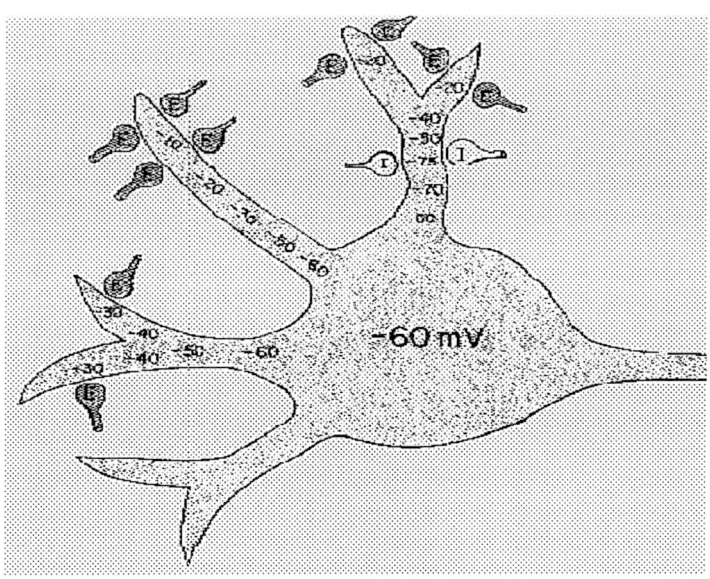
Jeśli komórka jest w fazie generowania potencjału iglicowego, to nie reaguje na dochodzące do niej inne sygnały. Dzieje się tak również przez ok. 2 ms po tym fakcie. Okres ten nazywamy refrakcją bezwzględną, ponieważ w tym czasie nie jest możliwe powstanie kolejnego potencjału iglicowego. W ciągu następnych ok. 8 ms już ma tę możliwość, jednak warunki aktywacji są

trudniejsze do spełnienia. W tym czasie komórka wraca do stanu spoczynkowego i jej siła reagowania na bodźce jest mniejsza. Rozkład potencjałów jest podobny jak w przypadku hamowania, co prezentuje to rys. 3 fragment C. Okres ten jest nazwany refrakcją względną.

Po depolaryzacji w komórce jest nadmiar jonów sodowych. Część z nich zostaje usunięta przez działanie pompy sodowej, a większość pozostałych dostaje się do neurytu tworząc falę biegnącą wzdłuż włókna osiowego.

Należy zwrócić uwagę, że komórka może być pobudzona do potencjału mniejszego niż generatorowy. Jest ona wtedy w stanie pobudzenia podprogowego. Ma wówczas tę właściwość, że łatwiej reaguje na następne bodźce pobudzające. W zależności od fazy pracy komórki suma pobudzeń musi mieć określoną wartość, aby wygenerowany został potencjał iglicowy. Występują trzy fazy pracy: depolaryzacja, hiperpolaryzacja oraz stan przeciętny. Podczas hiperpolaryzacji pobudzenie komórki jest trudniejsze niż podczas stanu przeciętnego, ponieważ wymagana jest znacznie większa ilość neuromediatora pobudzającego.

3.3 Neuron jako układ przeliczeniowy



Rys.4. Przykładowy rozkład potencjałów we wnętrzu komórki w zależności od typu synaps

Pierwszoplanowym czynnikiem modyfikującym siłę oddziaływania synaps na polaryzację komórki jest czynnik przestrzenny. Synapsy zarówno pobudzające, jak i hamujące zlokalizowane na ciele komórki mają większe znaczenie bodźcotwórcze niż synapsy na odległych krańcach dendrytów (rys.

4). Wiąże się to ze słabym oddziaływaniem na zmianę potencjału komórki nawet mocno spolaryzowanych synaps z odległych dendrytów. Jednak, jeżeli jest spełniony drugoplanowy czynnik - czynnik czasowy, oddziaływania polaryzacyjne dendrytów nabierają nowej jakości. Jeśli jednocześnie, lub z drobnym przesunięciem czasowym, podrażniona zostanie większość synaps pobudzających, to nastąpi aktywacja (odpalenie) neuronu, czyli wygenerowany zostanie impuls iglicowy.

Możemy więc uogólnić, że neuron dokonuje czasowo-przestrzennego sumowania arytmetycznego docierających bodźców podprogowych. Omawiane stany spoczynku i pobudzenia należą do zjawisk elektrycznych, z tym że należy podkreślić, że nie chodzi tu o przepływ prądu elektrycznego niosącego energię ze źródła do celu, ale o przesunięcie jonów, propagujące się podobnie jak efekt domina, gdzie każda komórka podlega prawom własnego, osobnego bilansu energetycznego.

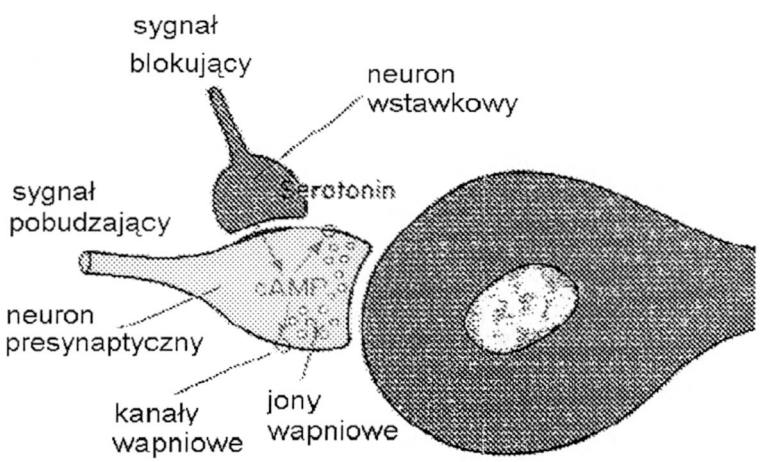
3.4 Czynniki modyfikujące pracę neuronu

Neuron jest wrażliwy na wiele substancji chemicznych. Mogą one istotnie zmieniać pracę neuronu (np. C_2H_5OH). Jedną z najważniejszych klas tego typu stanowi rozlane pobudzenie/ hamowanie, ponieważ pozostawia ślady w strukturze neuronowej na długi czas. Jest ono czynnikiem potrafiącym zmieniać wagi synaptyczne, czyli siłę oddziaływania, w zakresie od zaniku do kilkunastokrotnego zwiększenia oraz pozostawić taki stan po swoim zaniku. Proces zmieniający wagi jest stosunkowo długotrwały i zachodzi podczas odpalania się neuronu. Można go opisać następująco: Jeśli neurony są poddane działaniu rozlanego pobudzenia i podczas tego odpalają się (gdyż pracują aktywnie cały czas), to droga impulsu zostaje udrożniona. Neuron zostaje jeszcze bardziej uwrażliwiony na docierające do niego sygnały poprzez zwiększenie odpowiednich synaps. Obserwujemy zjawisko torowania [58].

Przeciwnie działa rozlane hamowanie. Podczas odpalania neurony zmniejszają wartości wag aktywnych synaps. Po dłuższym czasie neuron nie jest wrażliwy na pobudzenia z odpowiednich synaps. Zjawisko to nazywamy blokowaniem.

3.5 Analiza zjawisk fizjologicznych pod kątem wykorzystania w modelach sieci neuronowej

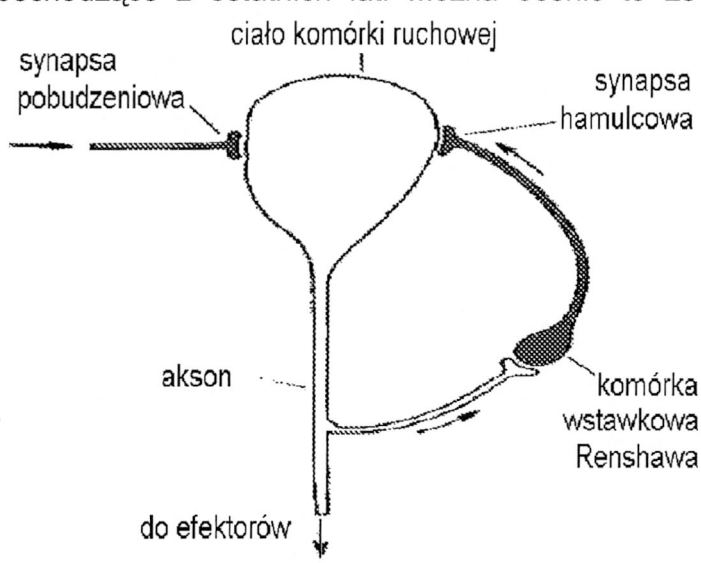
Pomimo tak dobrej znajomości określonych zjawisk elektrofizjologicznych neuronu, niektóre procesy nie są w ogóle wyjaśniane. Jak wykazały pierwsze symulacje, reguła Hebba nie przystawała do uczenia synaps hamujących. [rozdział 5] Należałoby się więc zastanowić, czy hamowanie presynaptyczne oraz odpowiadająca mu struktura synaps wykształca się w procesie uczenia, czy jest ona zdeterminowana genetycznie? Jeśli w procesie uczenia, to w jakim? Można przypuszczać, że nie może to być wyłącznie określone genetycznie, ponieważ organizm żywy wykazuje duże możliwości uczenia, w tym na pewno znalazłyby się funkcje niemonotoniczne. Jak twierdzi znany specjalista z tej dziedziny, prof. Ryszard Tadeusiewicz, (Konferencja PAN Wrocław 1996) nie była znana jakakolwiek spójna praca na temat powstawania lub pochodzenia synaps hamujących (rys. 5), mimo że synapsy hamowania presynaptycznego są widoczne w preparatach i były badane elektrochemicznie przez J. Ecclesa [53], za co otrzymał nagrodę Nobla. Można przypuszczać, że w synapsie hamowania presynaptycznego występuje mechanizm torowania w przypadku aktywności neuronu pobudzającego oraz wstawkowego (hamującego). Czy torowanie to występuje na linii neuron pobudzający - neuron pobudzany, czy też na linii neuron pobudzający - neuron wstawkowy, jeśli neuron pobudzany nie odpalił się, pozostaje to nadal w fazie domysłów.



Rys.5. Synapsa hamowania presynaptycznego

Pełniejsze zamodelowanie pewnych zjawisk fizjologicznych w sztucznej sieci neuronowej ma na celu możliwe najlepsze naśladowanie biologicznego pierwowzoru, z drugiej zaś strony pozwoli być może wyjaśnić pewne fakty biologiczne. Selekcji materiału tego rozdziału oraz niektórych generalizacji dokonała neurolog i fizjolog prof. Ludmiła Borodulin- Nadzieja. Nierozwiązanym problemem jest metoda uczenia synaps hamujących, zarówno hamowania presynaptycznego, jak i bezpośredniego. Fakt, który rzuca światło na ten temat to uwarunkowanie genetyczne struktury układu nerwowego. Jeśli chodzi o procent synaps pobudzających i hamujących uwarunkowanych genetycznie w stosunku do posiadających możliwości adaptacji w procesie uczenia, to można faktycznie wyróżnić dwie grupy neuronów związanych z tym podziałem: te które rosną wyłącznie ze schematem zapisanym genetycznie, oraz te z większymi możliwościami adaptacyjnymi. Pierwsza grupa jest odpowiedzialna za ogólną strukturę układu. Są w niej elementy, bez których organizm skazany byłby na zagładę oraz inne ważne grupy łączące stosunkowo bardzo odległe ośrodki (coś w rodzaju magistral). Szacuje się genetyczne uwarunkowanie na ok. 80% liczby neuronów w najprostszych partiach układu nerwowego. Sam mózg ze względu na swoją plastyczność oraz złożoność nie jest przebadany.

Wysoce pożądane byłoby poznanie stosunku ilości synaps pobudzających do hamujących. Znane są jedynie szacunkowe dane pochodzące z ostatnich lat. Można ocenić to ze względu na ilość chorób występujących w obu przypadkach - związanych z mediatorami pobudzającymi i hamującymi. Prawdopodobnie na hamowanie zużywane jest 60% energii przetwarzania lub więcej. Jest to wskazówka, jak ważny jest to proces. Pewne wnioski

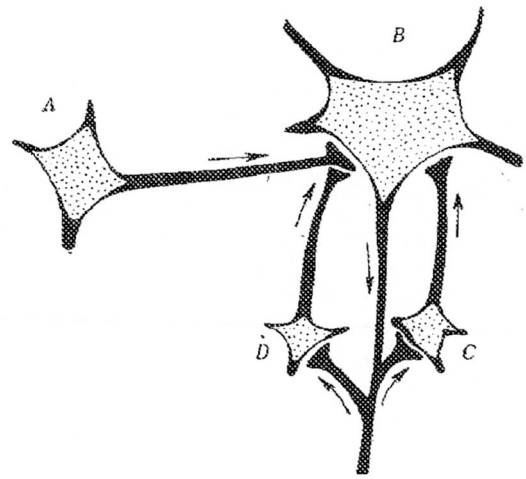


Rys.6. Hamowanie bezpośrednie przez komórkę wstawkową Renshawa

dotyczące uczenia w kierunku blokowania (zmniejszania wartości wag) można wyciągnąć obserwując działanie rozlanego hamowania. Droga impulsu może stawać się mniej drożna w wyniku tego procesu, jednak cały czas dotyczy to tylko synaps pobudzających. Wpływ rozlanego pobudzenia / hamowania na neurony jest znany (lub wydaje się być znanym). Do całościowego przebadania tego mechanizmu brakuje niestety jednej niemalże podstawowej rzeczy: jaki mechanizm steruje generowaniem rozlanego pobudzenia / hamowania? Czy jest on sterowany tylko chemicznie, czy również elektrycznie? Brak odpowiedzi na to pytanie komplikuje wysnuwanie hipotez na temat metod uczenia całych sieci (aby wystąpiła jakakolwiek synergia neuronów). Sam proces hamowania impulsów elektrycznych jest natomiast znany. Siła hamowanie presynaptycznego jest ustalana w procesach chemicznych. Sam proces hamowania jest procesem elektrycznym. Aspekt chemiczny podczas pojedynczego aktu hamowania jest pomijalny, gdyż nie może on hamować tak skutecznie (długo i dokładnie) jak hiperpolaryzacja. Znana jest też funkcja pomiędzy potencjałem z neuronu hamującego a skutecznością hamowania. Była ona badana kilka lat temu na podstawie prac Ecclesa przez doktoranta prof. Tadeusiewicza - Janusza Majewskiego [157]. Jego symulacje potwierdzają przydatność liniowej aproksymacji skuteczności hamowania względem potencjału hamującego. Gdy porównamy ilościowo oba typy hamowania synaptycznego: hamowania presynaptycznego oraz typu komórek Renshawa, działających bezpośrednio depolaryzacyjnie na błonę (rys. 6]), to znakomita większość jest typu pierwszego. Typ drugi występuje jedynie w kilku miejscach układu nerwowego i jest wysoce zdeterminowany genetycznie, ponieważ występuje wysoka powtarzalność tych elementów u różnych osobników. W literaturze podawany jest schemat samoblokowalnego wejścia (rys.7]), typu hamowania presynaptycznego. Jest to schemat stabilizatora częstotliwości, obrazuje np. hamowanie oboczne na siatkówce. Po analizie dochodzi się do wniosku, że akson neuronu C musi mieć właściwości hamowania bezpośredniego, gdyż w przeciwnym wypadku układ uległby nie kończącemu się wzbudzeniu. Oznacza to, że jedna komórka może otrzymywać sygnały hamujące różnych typów!

Hipoteza dziedziczenia umiejętności nabytych mogłaby rzucić pewne światło na uwarunkowanie genetyczne budowy organizmów, jednak

mechanizmu takiego do tej pory nie wykryto. Jedynym obecnie uznawanym procesem dokonującym przystosowywania się gatunku do środowiska jest selekcja naturalna. Jest to jednak wysoce zastanawiające, ze względu na statystycznie obserwowane szybsze przebieganie procesu adaptacji. Omawiane dane statystyczne są niestety niekompletne.



Rys.7. Schemat samoblokowalnego wejścia

Powróćmy znów do pojedynczych neuronów. W jakim stopniu można zagwarantować binarność przesyłanych sygnałów międzyneuronowych, czy na wyjściu mogą pojawiać się inne stany? Sygnał na wyjściu może przybierać pewne wartości również podczas nieodpalania się neuronu. Napięcia te nie przekraczają jednak 1 mV (mniej niż 1/100 amplitudy potencjału sygnału iglicowego), czyli prawdziwa jest teza o ich binarności. Są one związane bardziej z metabolizmem komórki niż z faktycznym przetwarzaniem.

Jeśli prześledzimy układ nerwowy pod kątem zakresu sprzężeń zwrotnych, generatorów i ich odpowiedników, to okazuje się że sprzężenia zwrotne występują na wszystkich szczeblach szczegółowości. W większości są to sprzężenia ujemne, ale wykryto również dodatnie (nie mówiąc o sprzężeniach wewnątrz neuronu). Jeśli chodzi o generatory to występują one dosyć często, np. jako elementy sprzężeń zwrotnych. Ogólnie mówiąc są one często zależne. Oprócz tego możemy znaleźć w mózgu stabilizatory(!) częstotliwości. Istnieje jeden wyspecjalizowany obszar zajmujący się głównie pobudzaniem określonych partii (nadający zmienne tempo pracy każdemu osobnemu modułowi mózgu). Zwany jest on układem siatkowatym. Nadaje on tempo pracy mózgu (sen, koncentracja uwagi). Sieć neuronowa nie może wygenerować impulsu sama z siebie (z wyjątkiem specjalizowanej struktury sterującej sercem). Istnieje więc wiele zabezpieczeń, aby w mózgu nie zabrakło pobudzeń. Chwilowy brak odpaleń wszystkich neuronów jest równoznaczny ze śmiercią.

Ważną rolę w budowaniu modeli mają również statystyczne dane na temat pracy neuronów, częstotliwości pracy oraz ich czułość na zmiany częstotliwości podczas sumowania czasowego. Co prawda nie można oszacować jaki procent neuronów odpala się w ciągu podanego odcinka czasowego i tym podobnych przydatnych statystyk, ale każde znane parametry muszą być wykorzystane. Maksymalna częstotliwość ograniczona jest refrakcją, ale wynosi nawet 1000 Hz w niektórych typach neuronów. Człowiek jednak słyszy i 8000 Hz, a po przesunięciach fazowych potrafi identyfikować kierunek. Jak się okazuje, obie w.w. jakości są przetwarzane przez osobne układy neuronowe i nie ma tu żadnej sprzeczności. Rzuca to jednak pewne światło, iż w sytuacjach skrajnych natura stosuje nieintuicyjne rozwiązania, a ignorowanie ich istnienia może doprowadzić do błędnych wniosków.

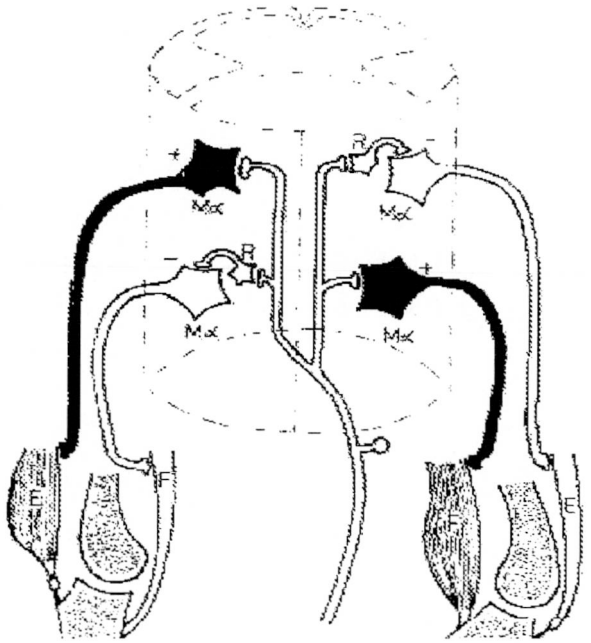
Jeśli chodzi o dokładność sumowania, to nie jest ona dokładnie zbadana. Rozpatrując poszczególne elementy sumowania potencjałów w neuronie należy uwzględnić, iż każdy neuron tego układu posiada możliwości adaptacyjne. Poza tym występuje również zjawisko habituacji, czyli selekcjonowanie powtarzających się informacji. Stopień habituacji jest prawdopodobnie ustalany genetycznie. Zwykle neurony za receptorami posiadają częstotliwości pulsowania rzędu 20 - 200 Hz (Częstotliwość graniczna sygnału jest znacznie większa ze względu na stromy przebieg "iglicy"). Równie ważnym elementem jest czułość neuronów, przy czym należy ją również rozpatrzyć jako funkcję częstotliwości. Zmiana częstotliwości o pewną wartość jest w granicy błędu, gdy nie wywoła zmian w funkcjonowaniu sieci. Trudno ocenić, jaka jest czułość neuronów na zmianę częstotliwości wejściowej. Z pewnych przesłanek wynika, że może tu występować wrażliwość na dość subtelne przesunięcia fazowe (np. przy mechanizmie lokalizacji źródła dźwięku w słyszeniu dwuusznym). Zwykle przyjmuje się, że błąd działania analogowych mechanizmów w sieci (sumowanie czasowe w synapsach) jest rzędu 1 - 5%.

Można próbować budować założenia dwóch różnych koncepcji maszyn neuronowych. W pierwszej większość neuronów jest ciągle pobudzana (chociaż podprogowo), w drugim panowałaby 'pustka elektryczna' z krążącymi pojedynczymi impulsami. Natura jest bliższa temu pierwszemu modelowi.

Pomijane przez neurofizjologię, a posiadające znaczenie dla modelu jest również wzajemne oddziaływanie na siebie pracujących neuronów. Wiążą się z tym pośrednio dynamiczne zmiany potencjałów otoczenia neuronów oraz zmiany stężeń związków chemicznych (np. tlenek azotu) wpływające na działanie sąsiadów oraz zmiany plastyczności (szybkości) uczenia sąsiadów - neuronów. Mówiąc 'sąsiad' mamy na myśli wzajemne fizyczne położenie neuronów w przestrzeni. W naturalnych sieciach, jak wiadomo, przestrzeń ta jest trójwymiarowa. W modelu przestrzeń musi być sztucznie i dość dowolnie zdefiniowana, ale po jej zdefiniowaniu musi być konsekwentnie przestrzegana. Fizjologicznie zmiany plastyczności uczenia sąsiadów neuronu aktywnego są tłumaczone obecnie przenikaniem przestrzennym tlenku azotu między neuronami. Wzajemne oddziaływania neuronów były rozważane przez ks. Sedlaka [151], nie są to jednak dane czysto fizjologiczne. Faktem jest, że istnieją zależności pomiędzy ośrodkami np.: Jeżeli czujemy głód, to systematyczny wzrost częstotliwości odpaleń neuronów w tym ośrodku powoduje zmniejszanie średniej częstotliwości odpaleń w innych ośrodkach, tak że po jakimś czasie jesteśmy już tylko w stanie myśleć o głodzie, w żadnym przypadku o pięknie.

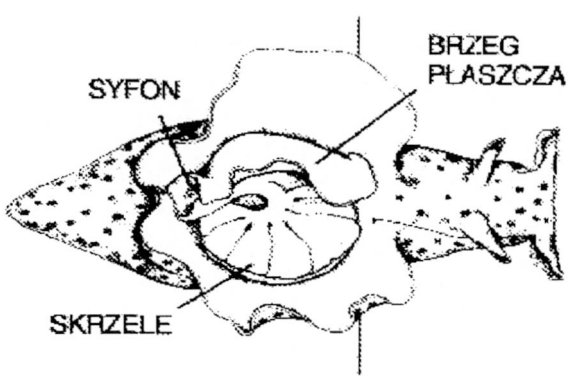
Należałoby rozpatrzyć również problem uczenia wysokopoziomowego i zjawiska z tym skorelowane. Na pewno bezpośredni związek ma częstotliwość układu siatkowatego. Pobudza on różne części mózgu zmuszając go do podwyższonej aktywacji. Układ siatkowaty jest natomiast pobudzany ilością różnorodnych bodźców docierających do mózgu. Pod względem chemicznym zasada torowania jest jedna. Generowanie określonych faz (np. udrażniania / hamowania) jest związane z utrzymaniem homeostazy i jest sterowane korą mózgową, a móżdżek to kontroluje. Dziedzina rozpatrująca człowieka nie jako obiekt, lecz jako proces (approach system) ma za ambicje zbadać prawa rządzące homeostazą. Ogólną zasadą działającą w zdrowych organizmach jest optymalizacja zużycia energii oraz wykonywanej pracy.

Wróćmy do synaps hamujących. Bardzo ważną rolę (jak się okazało na podstawie prac z IMPL5B, IMPL6) [24] pełnią różne typy hamowania impulsów w sieciach neuronowych. W przyrodzie znane są obecnie dwa podstawowe typy hamowania, którym warto się jest przyjrzeć ze szczególną uwagą pod kątem biochemicznym oraz występowania w organizmie. Można znaleźć je jednocześnie w układach nerwowych.



Rys.8. Motoneurony alfa oraz komórki Renshawa w odruchu obronnym zginania

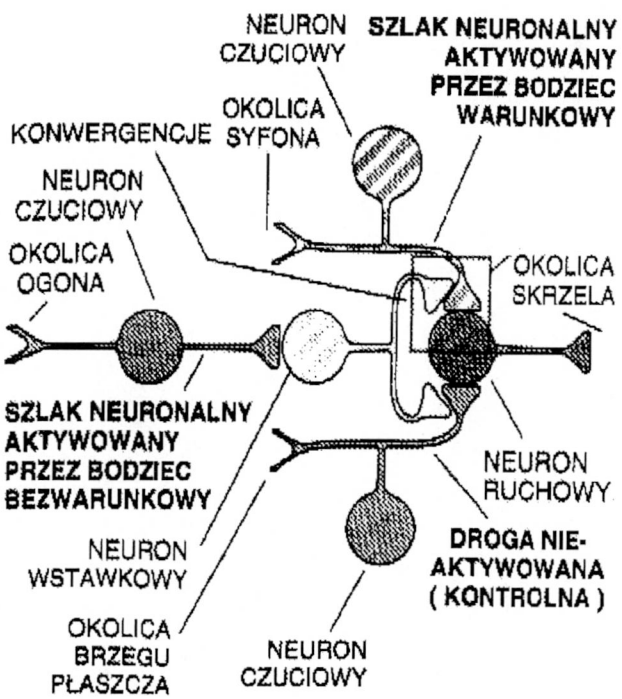
Pierwszym z nich jest synapsa posiadająca neuromediator hamujący [58] str.573, [156] str.115. Przy aktywacji tego układu hiperpolaryzacja wnętrza komórki (a często tylko dendrytu) może zmaleć z –60mV do -75mV. Tego typu synapsy występują przede wszystkim na ciele komórki (soma) lub na dendrytach w części zbliżonej do środka komórki. Powstały więc w procesie dość skomplikowanym (np. uczenia), gdyż inne ich umiejscowienie nie dawałoby odpowiedniego działania, a z punktu widzenia anatomii jest ono równie prawdopodobne. Hamowanie to może wpływać selektywnie na określone synapsy, ale również ma wpływ na całe pobudzenie komórki. Jako przykład biologiczny można podać niektóre elementy sterujące np. motoneurony alfa w okolicy rogów brzusznych. W układzie tym występuje hamowanie generowane przez charakterystyczną



Rys.9. Budowa ślimaka Aplysia

komórkę wstawkową Renshawa na rys. 8 oznaczoną symbolem R. Ma ona za zadanie stabilizację i pracuje w pętli sprzężenia zwrotnego ujemnego (rys. 8). Dokładniejszy opis działania jest następujący: motoneuron alfa produkuje acetylocholinę, która przez swoje kolaterale zakończone w komórce wstawkowej będzie ją pobudzać. Ona z kolei wytworzy glikol (do neuromediatorów hamujących należy również substancja GABA czyli kwas gamma-aminomasłowy), który ma za zadanie hiperpolaryzację motoneuronu.

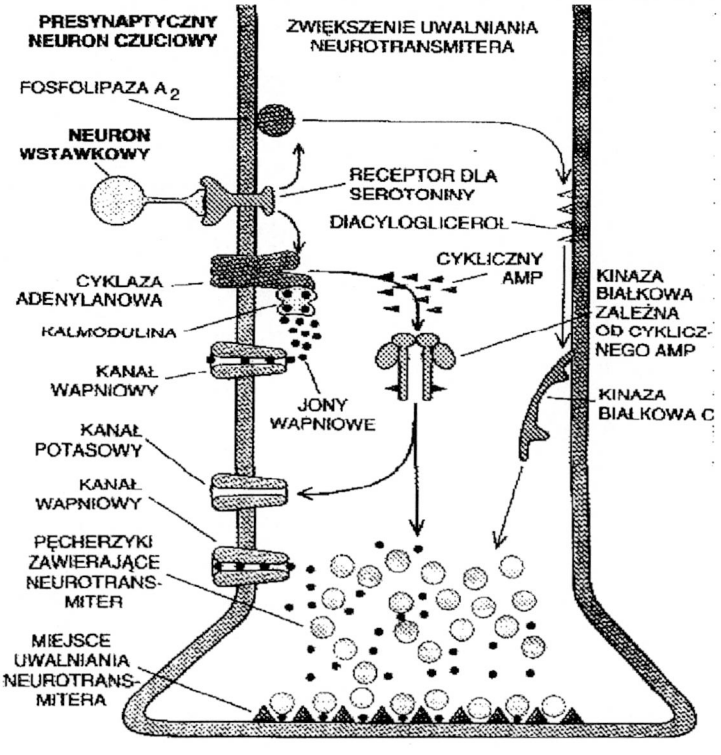
Drugim typem hamowania jest wejście hamujące (Noxious stimulus). Zapobiega wytworzeniu się pobudzenia postsynaptycznego wprowadzając hiperpolaryzację w synapsie. Hamowanie to jest wyłącznie selektywne. Dzieje się tak dlatego, że noxious stimulus działa serotoniną na gospodarkę cAMP we „właściwej synapsie" (Sensory stimulus). Zablokowanie neuromediatora powoduje zablokowanie drogi dla propagacji sygnału. Układy tego typu wykryto w ślimaku „Aplysia" [rys. 9]. [58] str. 693 [158] str. 56. Ślimak *Aplysia californica* jest jednym z pierwszych niebanalnych pod względem złożoności układu nerwowego obiektem przebadanym co do neuronu. Posiada on tylko 20 000 neuronów i to w dodatku wyjątkowo dużych. Jego układ nerwowy posiada odruch chowania skrzeli, oraz możliwości kojarzenia prymitywnych faktów. Dlatego stał się również wdzięcznym obiektem badań nad mechanizmami uczenia. Na podstawie analiz biochemicznych neuronu można wyróżnić kilka mechanizmów uczenia w układzie nerwowym. Niektóre zależności mogą być powiązane z synapsami hamowania. Zgodnie z zasadą Donalda O. Hebba z



Rys.10. Schemat odruchu ślimaka *Aplysia*

1949 roku, aby doszło do wzmocnienia siły połączeń pomiędzy neuronami, muszą one się pobudzać jednocześnie.

Po badaniach Aplysia w 1963 roku zaproponowano pewną modyfikację tej reguły. Może dojść do wzmocnienia połączenia również bez aktywowania neuronu postsynaptycznego. Mechanizm ten wymaga jednak jednoczesnej aktywacji innego neuronu (najczęściej wstawkowego torowania). Proces ten można opisać następująco: pod wpływem bodźca bezwarunkowego następuje aktywacja odpowiadającego mu szlaku neuronowego i w rezultacie pobudzenie neuronu wstawkowego. Ten z kolei za pomocą serotoniny powoduje aktywacje cyklazy adenylanowej w neuronie czuciowym. Jeśli jest on akurat w tym momencie pobudzany, to wzrasta stężenie jonów wapniowych. Łączą się one z kalmoduliną i z kolei te z cyklazą adenylową. Aktywując ten enzym zwiększa się synteza cAMP i dalsze reakcje są łatwe do przewidzenia (aktywacja kinaz białkowych i zwiększenie wydzielania neurotransmitera). Tak więc istnieje zasada kojarzenia nie objęta przez Hebba. Rys. 10 przedstawia schemat ogólny dróg neuronalnych odruchu warunkowego i bezwarunkowego chowania skrzela. Procesy zachodzące w synapsie w okolicach skrzela, omówione powyżej, prezentuje rys. 11.



Rys.11. Mechanizm warunkowania klasycznego u *Aplysia*

Wymieniony wyżej mechanizm występuje w trakcie uczenia nieświadomego. Układ nerwowy jest w stanie zapamiętać kolejność czasową występowania określonych bodźców (potencjalnych warunkowych) oraz

skorelować je z określonymi efektorami bez udziału świadomości człowieka. Natomiast zasada Hebba występuje w neuronach zaliczanych do obszaru zwanego hipokampem. Jest to twór, który posiada trzy główne drogi neuronalne i jest w stanie kojarzyć informacje z dłuższego czasu. Jest to też jedyny układ w mózgu potrafiący zapisać tę informację do pamięci długotrwałej abstrakcyjnej. Jego działanie jest związane ze świadomością, gdyż jeżeli jakiegoś wydarzenia nie spostrzeżemy to oznacza, że hipokamp nie zadziałał efektywnie. Wydarzenie takie nie będzie nam się mogło nigdy przypomnieć (pomimo, że informacja o nim przewędrowała przez znaczną część naszego mózgu), ponieważ nie przeszła przez „układ zapisu do pamięci".

Można określić dwa typy pamięci niskiego poziomu, jaką może dysponować model sieci neuronowej:

1. pamięć krótkotrwała - jest to pamięć, w której główną rolę pełnią krążące w sieci impulsy w sprzężeniach zwrotnych oraz wartości potencjałów postsynaptycznych w neuronach w danej chwili czasu. Jednocześnie proces zapamiętywania jest bardzo szybki. Może wystarczyć jeden, dwa sygnały wejściowe, aby zaistniała informacja w tej pamięci. Równie szybko może ulegać kasowaniu.

2. pamięć długotrwała - działanie tej formy pamięci oparte jest na zmianach wartości wag synaptycznych. Proces zapamiętywania (w tym wypadku uczenia) jest długotrwały. Modyfikacja wag podczas działania sieci jest stosunkowo niewielka, dlatego uczenie wymaga długotrwałej prezentacji wzorców wejściowych. Ten rodzaj pamięci opiera się na tworzeniu struktury połączeń międzyneuronowych. Same pojedyncze sygnały nie mają tu statystycznie dużego znaczenia. Sygnały pojawiają się jedynie podczas dostępu do informacji.

4. Założenia modeli pulsujących sztucznych sieci neuronowych

4.1 Założenia podstawowe budowanych modeli

Modele sztucznych sieci neuronowych zostały konstruowane ze szczególnym zwróceniem uwagi na następujące elementy:

- sztuczna sieć neuronowa powinna posiadać właściwości maszyny z pamięcią, w przeciwieństwie do ograniczania się do przekształceń funkcyjnych aktualnych danych wejściowych
- rozproszony algorytm uczenia wykształcającego pamięć długo- i krótko-terminową. Poprzez rozproszoność rozumie się autonomiczność pracy pojedynczego elementu, w tym przypadku neuronu, zarówno w procesach pracy jak i uczenia. Pojedynczy neuron ma zatem dostęp do części zbioru wszystkich danych systemu. Dane docierające do neuronu można podzielić na dwie grupy. Pierwsza zawiera informacje pochodzące z sąsiedztwa neuronu, np. neuronów z którymi istnieje połączenie synaptyczne. Druga grupa to dane dotyczące globalnego stanu sieci (np. modelowaniu rozlanego pobudzenia), oraz w jakimś zakresie, danych wejściowych całego systemu .
- wysoka równoległość procesów uczenia i przetwarzania. Jest ona zapewniona przez odpowiednio zorganizowany przepływ danych w modelu.
- tania przenośność do postaci sprzętowej, (brak szybkich magistral, niskie częstotliwości pracy elementów, powtarzalność schematu)
- tolerancja sieci na uszkodzenia neuronów.

4.2 Symulator oraz jego platforma sprzętowa

Badane modele zostały podzielone na dwie grupy i roboczo nazwane IMPL i SNN, adekwatnie do użytego symulatora: modele binarne IMPL 1 - 4 i modele z symulacją ładunków postsynaptycznych IMPL 5 - 8 oraz SNN 9 – 12. Symulacje zostały przeprowadzone na specjalnie skonstruowanych pod każdą grupę sieci symulatorach bazujących na platformie PC. Pierwszy z nich (IMPL) działa w środowisku BORLAND C pod dyskowym systemem operacyjnym.

Drugi jest w pełni 32 bitowy, działający w środowisku chronionym MS Windows, oparty o MS Visual C++, zwany dalej SNN. Oba symulatory charakteryzują się wysoką specjalizacją oraz przystosowaniem do tej dziedziny. Również inni badacze [141] podążyli w stronę specjalizowania symulatorów w ostatnich latach z powodu ograniczoności możliwości np. Matlaba. Obiekt symulowany, jego właściwości oraz algorytm zachowania, jest zapisywany w języku C a następnie kompilowany razem z symulatorem. Wytworzony kod jest całkowicie kodem zapisanym w języku maszynowym procesora, zatem podczas symulacji nie występują żadne procesy z zakresu interpretacji obiektu, co znacznie przyśpiesza symulację. Wewnątrz symulatora występują odpowiednie moduły, które przygotowują dane wejściowe do sieci z próbkowanego sygnału dźwiękowego (normalizacja, FFT, kodowanie częstotliwościowe), oraz procedury, które umożliwiają rekonstrukcję dźwięku z wyjścia sieci, czyli zakodowanego częstotliwościowo widma FFT. Wszystkie operacje są przeprowadzane z archiwizowaniem wyników pośrednich. Możemy je otrzymać w postaci plików.

Formalny zapis przekształceń wstępnych sygnału ma następującą postać:

$$
\begin{gathered}
V_t = FFT\left(W\left(t,t+k\right)\right) \\
V_t = \left(x_{1,t}, x_{2,t}, \ldots, x_{L,t}\right) \\
x_{1,t}, \ldots x_{L,t} \in \mathbb{R}^+ \\
def.\, G\left(x_{i,t}\right) = \left(w_{i,1,t}, w_{i,2,t} \ldots, w_{i,k,t}\right), D_G = \mathbb{R}^+, P_G = \left(0,1\right)^k \;:\; x_{i,t_1} \le x_{i,t_2} \Rightarrow \\
\Rightarrow \left|G\left(x_{i,t_1}\right)\right| \le \left|G\left(x_{i,t_2}\right)\right| \\
\forall\, i,j,t \quad w_{i,j,t} \in \left(0,1\right) \\
X_t = \left(w_{1,1,t}, w_{2,1,t}, \ldots, w_{L,1,t}\right) \\
\vdots \\
X_{t+k} = \left(w_{1,k,t}, w_{2,k,t}, \ldots, w_{L,k,t}\right)
\end{gathered}
\tag{6}
$$

gdzie:

W(t,t+k) – pierwotny ciągły sygnał wejściowy z przedziału czasowego *(t,t+k)*

FFT – funkcja realizująca szybką transformację Fouriera

V_t – wektor o *L* współrzędnych zawierający wynik FFT z chwili *t+k*

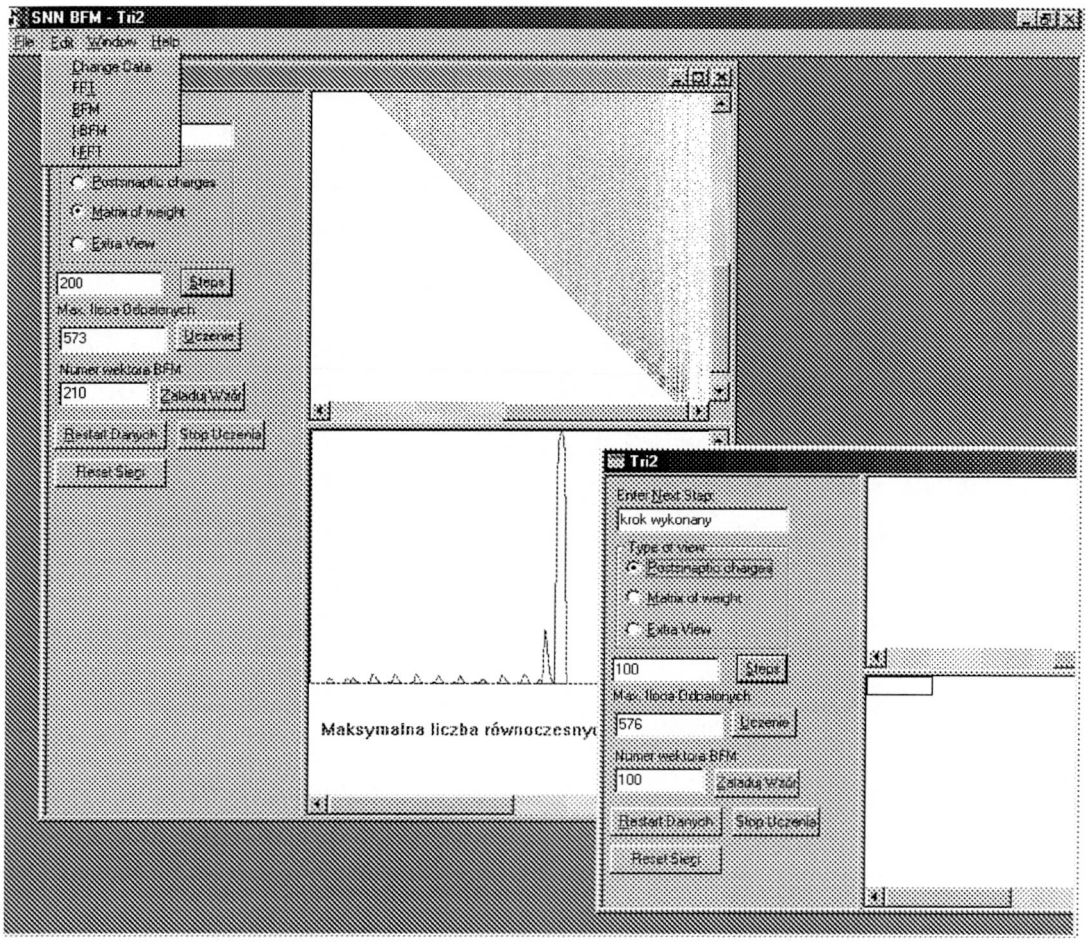
G – funkcja zachowująca monotoniczność względem długości wektora wyjściowego

X_t – wektor wejściowy do pulsującej sieci neuronowej w chwili *t*

Symulator SNN umożliwia ponadto stałe monitorowanie matrycy wag oraz potencjałów postsynaptycznych zarówno podczas pracy jak i uczenia sieci. Obie matryce można oglądać w postaci graficznej oraz możemy przemieszczać się po nich używając myszki. Aktualny stan sieci, wag, progów, potencjałów postsynaptycznych oraz rozkładu sygnałów międzyneuronowych może być w dowolnym kroku nagrany do pliku, co umożliwia powroty do ciekawszych stanów sieci oraz kontynuowanie doświadczenia rozpoczynając z tego stanu. Symulator monitoruje częstotliwość odpaleń neuronów w sieci oraz przedstawia to na automatycznie skalowanym wykresie. Podczas symulacji możemy wstrzymywać procesy uczenia, zaburzać sekwencje wejściowe itp.

4.2.1 Okno aplikacji Symulatora SNN

Przykładowe okno symulatora przedstawione jest na rys. 12. Aplikacja wykonana jest w technice MDI. Symulator może jednocześnie symulować działanie wielu sieci neuronowych przetwarzających te same dane lub inne, co jest ustawiane przy uruchamianiu danego okienka za pomocą definiowania pliku wejściowego i wyjściowego danej sieci. Przykładowe okienko aplikacji symulatora zawiera dwa dokumenty. Pierwsze zawiera sieć, której wizualizacja matrycy wag została wybrana w lewym okienku dialogowym i automatycznie wyświetlona nad prawym spliterem. Poniżej znajduje się wykres częstotliwości neuronów z pierwszych dwustu kroków symulacji. Okno dialogowe podaje nam również maksymalną liczbę odpalonych jednocześnie neuronów oraz numer aktualnie przetwarzanego wektora wejściowego. W menu głównym widzimy rozwiniętą kolumnę odpowiedzialną za cztery przekształcenia przystosowujące sygnał do formatu użytecznego przez modelowaną sieć, co zostanie szczegółowo omówione w następnej sekcji.



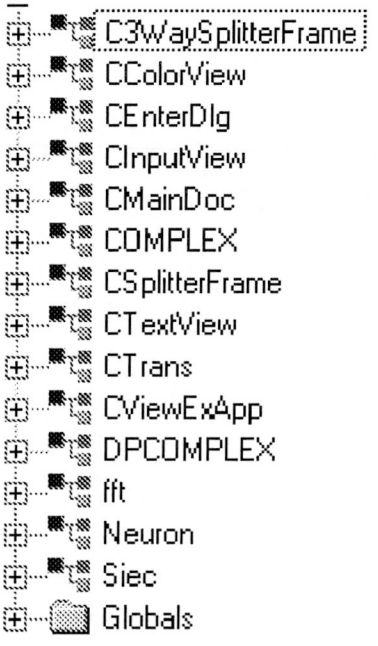
Rys. 12 – Okno aplikacji symulatora SNN z dwiema równolegle symulowanymi sieciami neuronowymi.

4.2.2 Konstrukcja modelu

Niezależnie od przyjętego symulatora (IMPL czy SNN) jądro modelu składające się z kilku obiektów, w myśl metodologii programowania obiektowego, jest identyczne. Jedynie metody tych obiektów związane z interfejsem użytkownika mają inną treść, ze względu na zupełnie inne możliwości obu platform. Różnicą symulatora SNN względem IMPL również jest możliwość użycia zasobów wirtualnych systemu operacyjnego, jak i możliwości zaadresowania i używania dużych, ciągłych segmentów danych. Pojemność pamięci wirtualnej czy kilkakrotnie większe zmienne indeksujące w rezultacie przekładają się na zwiększenie ilościowe elementów powtarzalnych modelowanego obiektu takich jak poszczególne neurony lub sieci. Pomimo, że

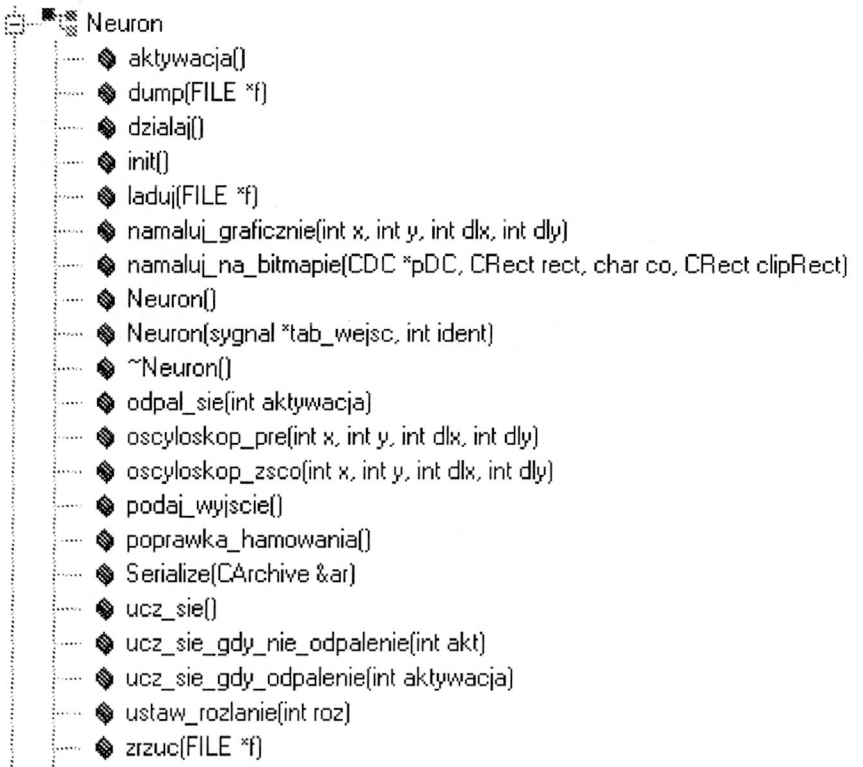
występuje tu jedynie zmiana ilościowa względem IMPL, efekty synergii neuronów w takich sieciach nabierają jakościowo innych właściwości, co jest omówione w rozdziale 5.3.

Symulator SNN zawiera następujące klasy obiektów:



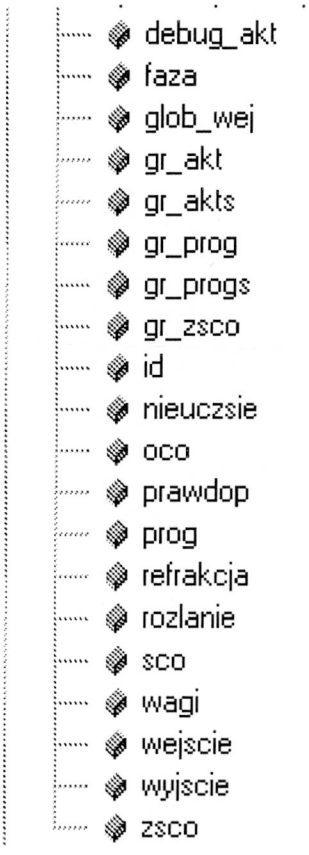
Samo jądro systemu tworzą klasy „*Neuron*", „*Siec*" oraz „*fft*". Są one wspólne dla symulatora SNN i IMPL. Klasa CTrans występuje tylko w symulatorze SNN i zapewnia automatyczną konwersję dźwięku poprzez kilka transformacji do postaci akceptowalnej przez pulsującą sieć neuronową zgodnie z wzorem (6). Pozostałe klasy służą do komunikacji z użytkownikiem np. okna dialogowe, okna wyświetlające matryce wag oraz potencjałów postsynaptycznych, wykresy częstotliwości odpaleń neuronów i matryce aktywnych neuronów na tle sieci. Ich szczegółowe omawianie wydaje się być w tym miejscu niecelowe. Podczas trwania symulacji odpowiednie przełączniki zlokalizowane w oknie dialogowym pozwalają użytkownikowi symulatora na szybką zmianę prezentacji dynamicznie zmieniającej się zawartości sieci neuronowej (np. matrycy wag czy potencjałów postsynaptycznych) za pomocą myszki.

Najmniejszym występującym obiektem jest neuron. Posiada on następujące metody:



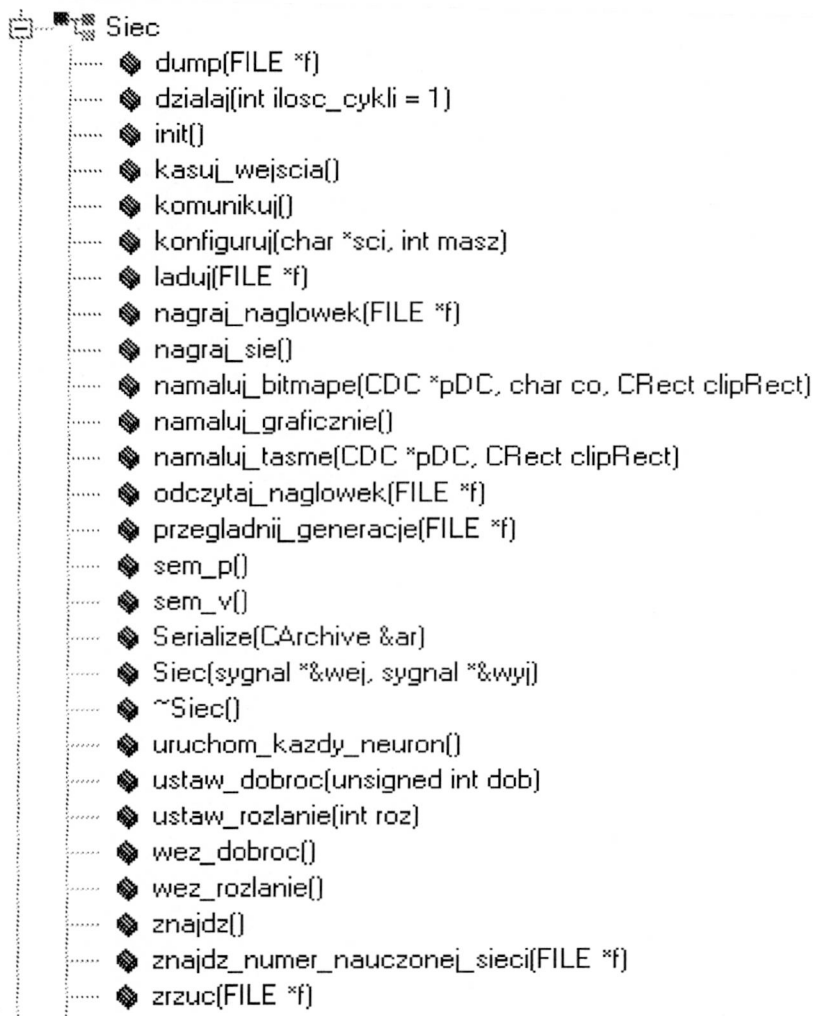
Najistotniejsze metody związane z pracą neuronu to funkcja *aktywacja()*, wyliczająca aktualne pobudzenie neuronu, zwracająca informację o przekroczeniu progu. Wartość jej liczona jest na podstawie wartości potencjałów postsynaptycznych oraz odpowiednich im wag. Metoda *działaj()* używając tej funkcji podejmuje decyzję o przekazaniu sterowania metodom *ucz_się_gdy_odpalenie()* i *ucz_się_gdy_nie_odpalenie()*. Niezależnie od tego wyboru, w każdym kroku symulacji wywoływana jest metoda *ucz_się()*, która zmienia również wartości wag i progu np. aby nie dopuścić do wyłączenia się neuronu z pracy sieci z powodu zbyt wysokiej wartości wylosowanego progu. Działania tych metod opisane są formułami matematycznymi (np. rozdział 5.3.2). Jedną z metod istotnych przy testowaniu synaps hamujących jest *poprawka_hamowania()* – metoda zapobiega powstawaniu niedozwolonych synaps w danym modelu np. synaps hamowania presynaptycznego z kilkoma synapsami hamującymi itp.

Atrybuty obiektu *neuron*, wymienione już po części, jak próg czy wartości postsynaptyczne, przedstawia kompletna lista:



Neuron posiada swój identyfikator *id* używany przy obsłudze autoasocjacji połączeń. Wartość refrakcji przechowywana jest w zmiennej *refrakcja*, a w przypadku modelowania tylko refrakcji bezwzględnej używana jest zmienna *faza*. *Glob_wej* jest wskaźnikiem na bufor wyjść neuronów, stanowiący jednocześnie tablicę wartości sygnałów wejściowych do sieci, w tym do każdego neuronu. W ten sposób praca neuronów jest synchroniczna. *Wejście* stanowi tablicę wartości potencjałów wygenerowanych przez impuls wejściowy na synapsie o odpowiednim indeksie oraz uwzględnia pozostałości potencjałów po poprzednich impulsach tej synapsy. Powstałe w ten sposób wartości są mnożone przez wartości *wagi* o tych samych indeksach przez funkcję obliczającą aktywację. Neuron zbiera też statystyki na temat swojej aktywności w dalszej i bliższej przeszłości w zmiennych *sco* i *oco*. Ich różnica określa ostatnią tendencję i jest przechowywana w atrybucie *zsco*. *Rozlanie* wraz z wartością aktualnego *zsco* jest elementem modyfikującym kierunek zmian częstotliwości, do którego dąży algorytm uczenia neuronu.

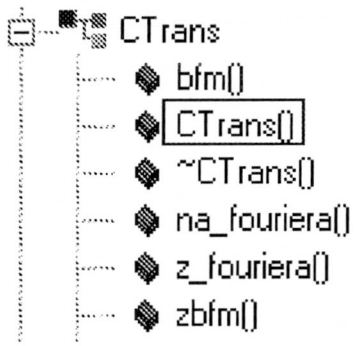
Obiektem nadrzędnym względem *Neuronu* jest *Sieć*. Koncentruje ona dowolnie wielką ustaloną liczbę neuronów w zakresie umożliwianym przez platformę sprzętową. Ponieważ struktura sieci umożliwia połączenia każdy z każdym, obiekt *Sieć* musi wygenerować neurony o odpowiedniej ilości synaps, aby każdy neuron mógł obsługiwać sygnały z wszystkich neuronów sieci, w tym swój sygnał, oraz wszystkie wejścia globalne do systemu (do sieci).
Metody obiektu *Sieć*:



Konstruktor sieci generując odpowiednie neurony przekazuje im wskaźnik na bufor sygnałów wyjściowych neuronów. Metoda *init()* ustawia parametry startowe wag, progu, faz i innych zmiennych neuronu przy pomocy identycznej metody obiektu *Neuron*. Po tych operacjach *Sieć* jest w stanie wykonywać metodę *działaj()*, co sprowadza się do wywołania przeliczenia każdego neuronu (*uruchom_każdy_neuron()*) oraz aktualizacji buforów wejściowych zgodnie z przeliczonymi wartościami wyjść neuronów (metoda:

komunikuj()). Obiekt *Sieć* posiada poza tym kilka metod wizualizacyjnych oraz procedury archiwizacji i przeglądu drzewa generacji sieci. Zasoby *Sieci* nie będące *Neuronami* sprowadzają się do zmiennych organizujących przeglądanie drzewa generacji oraz archiwizację sieci i z powodu mało istotnego znaczenia dla modelu nie będą omawiane.

Na uwagę zasługuje również klasa *CTrans.* Wykorzystuje on obiekt *fft* oraz zawiera następujące własne metody:



Spróbkowany sygnał podlega Szybkiej Transformacji Fouriera w metodzie *na_fouriera()* i w rezultacie powstają wektory o współrzędnych stanowiących wartości energii przypadającej na dany przedział częstotliwości. Współrzędne tego wektora są z kolei kodowane na binarne wektory o tej samej ilości współrzędnych za pomocą dość dowolnej ale monotonicznej funkcji w metodzie *bfm()* Taki format danych jest możliwy do podania na wejście omawianego modelu sieci. Aby możliwa była obserwacja procesów zachodzących w sieci za pomocą dźwięku zostały opracowane również transformacje odwrotne do wymienionych powyżej o nazwach *zbfm()* oraz *z_fouriera()* w wyniku których wyjście sieci może być z powrotem przekształcone w sygnał (np. dźwiękowy).

4.3 Nowe jakości w pulsującej sieci neuronowej

4.3.1 Kodowane danych wejściowych w pulsującej sieci neuronowej

Głównym założeniem pulsujących sieci neuronowych jest forma przekazywanego sygnału międzyneuronowego, określona jako impuls o

zdefiniowanym przebiegu. W znakomitej większości sieci oraz we wszystkich modelach tej pracy została założona identyczność tych impulsów, co oznacza brak możliwości zmian amplitudy sygnału. Fakt ten jest potwierdzony przez fizjologię. Istnieją również podejścia z modulacją amplitudy do PNN (Pulsed Neural Network [112] – Pulsujące Sieci Neuronowe), które nie są omówione w tej pracy.

Zakładając kodowanie impulsowe, obok metod fazowych, zwykle zastanawiamy się nad kodowaniem częstotliwościowym. Aby nie doprowadzić do nieporozumień warto zdać sobie sprawę, że używane czułe mierniki często przeprowadzają pomiar częstotliwości w określonej liczbie kolejnych impulsów. Należy w tym miejscu uściślić dla każdego modelu techniczny sposób mierzenia częstotliwości, ponieważ zwykle jako pomiar częstotliwości traktujemy ilość impulsów w jednostce czasu. Kolejną nieścisłością, która może wkraść się pod pojęcie częstotliwości odpaleń, bywa zwykle nie częstotliwość odpaleń neuronu, ale średnia częstotliwość odpaleń grupy lub wszystkich neuronów sieci. W odniesieniu do odpowiedzi sieci neuronowej podczas badań możemy też mówić o częstotliwości wyjścia sieci z jednego doświadczenia, lub średniej częstotliwości wyjścia sieci z kilku identycznych prób. Ta ostatnia metoda jest dość często wybierana, szczególnie wtedy, kiedy odpowiedź sieci jest zaledwie kilkoma impulsami i pojedyncze wyniki charakteryzują się dość dużymi błędami. W krańcowych przypadkach niektórzy badacze dla udowodnienia swoich hipotez podają częstotliwość wyliczoną jako odwrotność czasu pomiędzy dwoma odpaleniami neuronu, co w istocie rzeczy jest błędem. Zwracając uwagę, która z ww. metod mierzenia częstotliwości została użyta (np. przez neurofizjologów), oszczędzamy czas nad szukaniem potencjalnych błędów w modelu sieci.

Przy kodowaniu impulsowym możemy nadać znaczenie wielu czynnikom, nie tylko częstotliwości. Najprostszą metodą jest dekompozycja binarnej danej na pojedynczy impuls, oznaczający wystąpienie danego abstraktu. Drugim czynnikiem, który może zawierać informację, jest czas, który upływa do pierwszego pojawienia się impulsu (np. względem ostatniego odpalenia się danego neuronu). Standardowo podczas kodowania częstotliwościowego w określonym odcinku czasu dt należy umieścić taka liczbę impulsów x, aby stosunek x/dt był równy zadanej częstotliwości.

Wprowadzając nierównomierności w rozmieszczeniu tych impulsów (o ile to możliwe), możemy jednocześnie kodować dodatkowe dane. Nierównomierności rozkładu impulsów można formować tak, żeby były one skorelowane z nierównomiernościami z innych wejść lub nie, w ten sposób sieć może przypisać określone wejścia do jednej klasy. Widać tu już powolne przejście do metod kodowania fazowego. Podstawową metodą tej kategorii jest generowanie ramki synchronizującej. W tej ramce w odpowiednim momencie jest umieszczany impuls. Neuron na podstawie przesunięcia względem ramki określa jego fazę i tę wartość dalej przetwarza. Jest też możliwe kodowanie danych na kilku wejściach neuronu. Wtedy też wygodnie jest użyć przesunięć czasowych impulsów na różnych wejściach neuronu do przedstawienia określonych abstraktów. Oczywiście nie bez znaczenia jest też czas pomiaru. Jak wykazują badania najwięcej informacji niosą pomiary realizowane w okresach od 100 ms do 500 ms.

Powyższy przegląd nie wyczerpuje oczywiście wszystkich metod kodowania pojedynczego wejścia używanych w PNN (występują np. metody mieszane), stanowi jedynie przegląd najpopularniejszych. W badanych modelach używana jest standardowe kodowanie częstotliwościowe przy założeniu określonego odcinka czasowego oraz kodowanie bezpośrednie przy pomocy pojedynczego impulsu (np. przy uczeniu prostych funkcji binarnych).

Po zdefiniowaniu najniższej warstwy kodowania danych należy określić i przystosować strumień informacji wejściowych do formatu akceptowanego przez sieć. Proces ten jest zależny w każdym przypadku od rodzaju przetwarzanych informacji i często przesądza o powodzeniu całego przedsięwzięcia. Polega on na wyeliminowaniu, o ile to możliwie, symboliczności informacji i skonstruowaniu w zamian odpowiedniej przestrzeni wartości. Przez znakomitą większość badaczy jest on robiony intuicyjnie, a jedynie wieloletnie doświadczenie specjalisty sprawia, że wybrana funkcja przekształcająca dane wejściowe na przestrzeń danych wejścia sieci, jest zwykle w tej roli nie najgorsza [52]. Błędy przy wyborze tej funkcji mogą być w określonym zakresie niwelowane poprzez samą sieć neuronową, jednakże okupione jej dodatkowym nakładem pracy. Biorąc pod uwagę, iż pulsujące sieci neuronowe znakomicie nadają się do przetwarzania sygnałów, sztandarowym błędem z omawianego tematu jest reprezentacja próbek sygnału w systemie

dwójkowym i podanie ich w takiej postaci do sieci. Jak wiemy, system dwójkowy jest kodem, czyli symbolem, a nie badaną wartością, którą on przedstawia. W przypadku kiedy informacja wejściowa z natury jest symboliczna, np. kolor, definiowanie przestrzeni wejścia powinno być powiązane z potencjalnym znaczeniem danego abstraktu, o ile posiadamy jakiekolwiek hipotezy na ten temat.

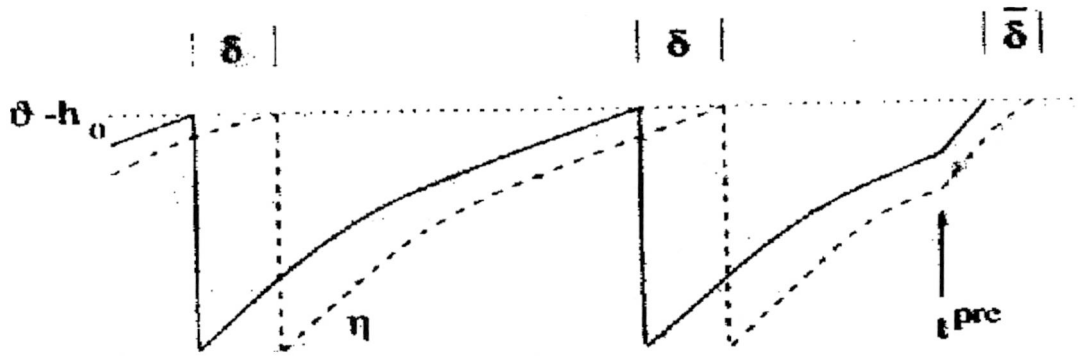
Po zbadaniu, czy przetwarzane dane nie mają właściwości symbolicznych, lub po opracowaniu odpowiedniego przekształcenia, celem zoptymalizowania pracy sieci należy ocenić, czy przypadkiem zbyt duży strumień informacji nie koncentruje się na jednym lub kilku wejściach do sieci pozostawiając wejścia spoza tego zbioru z bardzo małą ilością informacji. Może to prowadzić w krótkim okresie do przetrenowania zbyt utylizowanych wejść oraz zastój w uczeniu. Jest to dość poważny problem i nie ma na niego uniwersalnej recepty. Z pomocą, obok doświadczenia obecnie posiadanego w tej dziedzinie, przychodzi często natura i jej rozwiązania np. w przypadku przetwarzania dźwięku warto oba kanały dźwiękowe poddać określeniu amplitudy, normalizacji, przekształceniu Fouriera i z kolei wynik tego przekształcenia oraz wielkość amplitudy dostosować do formatu danych wybranego przy danym modelu sieci, czyli analogicznemu przekształceniu do realizowanego w biologicznym uchu.

4.3.2 Efekt synchronizacji lokalnej oraz wpływ szumów

W procesie przetwarzania w PNN (Pulsed Neural Network – Pulsująca sieć Neuronowa) mamy często do czynienia z następującym zjawiskiem: w zależności od funkcji pełnionej w sieci oraz od aktualnie przetwarzanych danych neuron X odpala się z określoną częstotliwością. W sąsiedztwie X znajduje się neuron Y, który posiada część synaps do identycznych neurytów co neuron X. Jeśli neurony pełnią podobne funkcje w sieci, a różnica polega na tym, iż przetwarzają dane z trochę innego obszaru wejściowego (np. inna część siatkówki), to zwykle średnia częstotliwość pracy X i Y są podobne. Jeśli faza tych odpaleń wynosi $\delta \neq 0$ oraz gdy zostanie aktywowany region wspólny, tzn. do obu neuronów dotrze impuls x w chwili t^{pre}, to kolejne mierzone czasy

pomiędzy odpaleniami wyniosą $\overline{\delta}$, gdzie $\overline{\delta} \leq \delta$. Jest to zjawisko, które może być wykorzystane w procesie przetwarzania oraz uczenia: neurony posiadające bardziej zbliżone fazy odpaleń mogą wzmacniać bardziej wzajemne synapsy niż synapsy do neuronów o innych fazach. Logiczną interpretacją może być hipoteza, iż skoro dane pochodzące z dwóch różnych źródeł nigdy nie zbiegają się w tym samym czasie, to znaczy że one są niezależne i nie powinny być syntezowane. I odwrotnie - jeśli dwa niezależne źródła informacji zwykle aktywują się w tym samym czasie, to znaczy iż prawdopodobnie dotyczą tego samego zdarzenia.

Sam mechanizm neuronowy występujący w tym zjawisku jest dość prosty. Podczas gdy sygnał x dociera do neuronów X i Y, X oraz Y są w określonej fazie aktywacji. Automatycznie dodatkowy sygnał powoduje jednoczesny szybszy wzrost potencjału aktywacji w obu neuronach względem sytuacji początkowej. Szybszy wzrost potencjału aktywacji powoduje większy kąt nachylenia funkcji potencjału aktywacji względem osi czasu (druga pochodna tej funkcji jest w tym momencie dodatnia). Jeśli w czasie oddziaływania sygnału x neurony przekroczą swoje progi, to czas pomiędzy ich odpaleniami będzie mniejszy względem poprzedniego. Obrazuje zdarzenie rys 13.



Rys.13. Mechanizm synchronizacji momentów odpaleń neuronów

Rozważanie przeprowadzone powyżej, podobnie jak kolejne w dalszej części pracy, pomimo iż operują na pojedynczych neuronach i pojedynczych okresach mają znaczenie statystyczne, a uproszczenie to jest użyte celem wyjaśnienia samej istoty zjawiska. W naturze napotykamy na różnego rodzaju szumy oraz zakłócenia i nie da się tego faktu zignorować, a być może ma on też wpływ na sam proces przetwarzania i uczenia. Po pierwsze, wartość progu

danego neuronu może być zakłócana tzn. zdarzają się odpalenia neuronu gdy nie zostanie osiągnięty potencjał progowy, jak i braki odpaleń nawet po przekroczeniu wartości progowej (wliczając refrakcję). Po drugie, mechanizm integracji sygnałów może być zakłócany różnymi czynnikami, szczególnie tymi, które działają na błonę komórkową. Może wystąpić wtedy sumowanie w „zaszumionym" fragmencie błony i wynik może być inny niż zwykle. Może nastąpić też przesunięcie sygnałów sumowanych w czasie, co niesie z sobą nieprzewidywalne wyniki pracy neuronu. Po trzecie, szum może być również wynikiem niezupełnego wykasowania potencjałów postsynaptycznych podczas odpalenia neuronu. Startowanie z akumulacją potencjałów nie od stanu zerowego może zaburzyć zarówno częstotliwość pracy jak i jej fazę.

5. Symulacje wybranych procesów fizjologicznych pod kątem wykształcania pamięci długo- i krótkotrwałej

5.1 Modele binarne

5.1.1 Modele IMPL1 i IMPL2 - podstawowe elementy budowy, działania oraz metod uczenia

Budowa tych modeli jest po części wzorowana na binarnych pamięciach Hopfield'a [63]. Ich działanie jest prawie identyczne, z tym że kolejne dane wejściowe są przetwarzane przed osiągnięciem stanu ustalonego. W ten sposób dynamiczny rozkład impulsów w czasie przetwarzania może interferować się z danymi wejściowymi. Głównym elementem modelu jest metoda uczenia bez nauczyciela, inna niż przyjmowana w pamięci Hopfield'a. Ma ona niestety jeszcze wady.

Założenia IMPL1 są bardzo ubogie i należy je zaliczyć do bazowych założeń kolejnych modeli. Jednym z elementów występujących we wszystkich omawianych symulacjach IMPL jest sposób kodowania danych podawanych do sieci. Polega on na tym, że elementarna dana wprowadzana do sieci (0 lub 1) reprezentowana jest przez wektor dwuelementowy. Ma on źródła w fizjologii i na takim też przykładzie zostanie wyjaśniony. Rozpatrzymy percepcję ciepła i zimna przez organizm. Gdy receptor ciepła przestaje wysyłać sygnały o tym fakcie, nie oznacza to wystąpienia zimna w środowisku, ponieważ za wykrycie zimna odpowiedzialny jest inny receptor. W ten sposób możemy wyznaczyć tabelę stanów (rys. 14), która będzie stosowana do kodowania. Oczywiście nie wszystkie naturalne abstrakty wejściowe wymagają takiego kodowania.

Wartość bitu	Reprezentacja	opis
0	(0,1)	zero
1	(1,0)	jeden
	(0,0)	stan dozwolony
	(1,1)	stan zabroniony

Rys. 14. Tabela kodowania sygnałów

Wspólnym elementem modeli IMPL jest również typ sygnałów

przesyłanych między neuronami. Mają tę samą semantykę i formę jak dane podawane do sieci.

Cechą charakterystyczną jest też symulowana równoległość pracy neuronów. W przypadku sieci ze sprzężeniami zwrotnymi pobudzenie neuronu jest funkcją wartości wyjść innych neuronów. Aby zatem wyniki pracy sieci nie były zależne od kolejności przeliczania neuronów, zastosowane zostały bufory przechowujące stan z danego kwantu czasu symulacji aż do następnego. Podczas pracy neuron korzysta z obrazu sieci zapamiętanego w buforach. Gwarantuje to symulację równoległości.

Analizowane sieci neuronowe oparte na metodzie uczenia wyłącznie za pomocą modyfikacji wag (IMPL1 – tak został nazwany pierwszy model) zgodnie z poniższym wzorem:

$$w_{i,j,t+1} = \begin{cases} w_{i,j,t} + c \Leftrightarrow y_{j,t} > 0 \\ w_{i,j,t} - c \Leftrightarrow y_{j,t} \leq 0 \end{cases} \qquad (7)$$

$w_{i,j,t}$ - wartość i'tej wagi neuronu j w chwili t, $-W < w_{i,j,t} < W$

$y_{j,t}$ - wartość wyjścia neuronu j w chwili t,

c, W – stała modelu o wartości dodatniej

miały następujące właściwości:

- włączały sprzężenia zwrotne, blokujące poprawne działanie sieci: nie zostały zauważone żadne efekty synergii neuronów, nie zauważono żadnego związku danych wejściowych z wartościami wag,
- zbytnio poszerzały rozpoznawane klasy abstrakcji, po dłuższym czasie prawie każdy ciąg wejściowy był akceptowany,
- włączały sprzężenia w obrębie pojedynczego neuronu,
- niektóre neurony przez cały czas pozostawały nieaktywne,

Aby uniknąć pojawiających się cech znanych z IMPL1, w założeniach modelu IMPL2 oprócz ww. elementów wspólnych zawarto:

- architekturę sieci umożliwiającej sprzężenia zwrotne z wyjątkiem autoasocjacyjnych,
- dwa podstawowe typy budowy synaps,
- metodę uczenia zdefiniowaną na tej strukturze,

- refrakcję bezwzględną.

W rezultacie otrzymuje się monolityczną strukturę sieci, w której np. warstwy to rzecz wyłącznie umowna. Każdy neuron ma dostęp do wszystkich danych wejściowych sieci. Zamodelowano dwa rodzaje synaps. Pierwszy opisuje synapsy pobudzające, drugi - hamujące analogiczne do synapsy hamulcowej komórki wstawkowej Renshawa. Nie występuje tu synapsa hamowania presynaptycznego. Model nie zawiera symulacji gromadzenia się ładunków postsynaptycznych. Refrakcja bezwzględna w tym modelu oznacza, że neuron przez *n* taktów po odpaleniu, nie odpala się lub kwant czasu symulacji jest równy czasowi refrakcji bezwzględnej. Ta druga opcja przyjęta jest w większości modeli. Uczenie realizuje się podczas pracy sieci zgodnie z następującymi zasadami:

- podczas odpalenia neuronu modyfikują się wagi synaps aktualnie pobudzanych zgodnie ze wzorem (8).

$$w_{i,j,t+1} = \begin{Bmatrix} w_{i,j,t} + c \Leftrightarrow w_{i,j,t} > 0 \\ w_{i,j,t} - c \Leftrightarrow w_{i,j,t} < 0 \\ rnd\{0,1,-1\} \Leftrightarrow w_{i,j,t} = 0 \end{Bmatrix} \qquad (8)$$

gdzie :

rnd() - funkcja losowa o rozkładzie płaskim

c – parametr z przedziału (W/1000, W/5). Najlepsze wyniki osiągano dla (W/100,W/30).

Wagi synaps niepobudzanych ($x_{i,j,t}$=0) nie modyfikują się. Zmienia się natomiast próg podczas odpalenia zgodnie ze wzorem (9).

$$h_{j,t+1} = \begin{Bmatrix} h_{j,t} + \dfrac{\sum_i w_{i,j,t} * x_{i,j,t} - h_{j,t}}{k} \Leftrightarrow y_{j,t} > 0 \\ h_{j,t} - a \Leftrightarrow y_{j,t} \le 0 \end{Bmatrix} \qquad (9)$$

gdzie :

$h_{j,t}$ - wartość progu neuronu *j* w chwili *t*

$w_{i,j,t}$ - wartość *i*'tej wagi neuronu *j* w chwili *t*

$x_{i,j,t}$ - wartość i'tego wejścia neuronu *j* w chwili *t*

$y_{j,t}$ - wartość wyjścia neuronu j w chwili t

a – stała dodatnia proporcjonalna do $c * I$, ale kilka rzędów mniejsza. Najciekawsze zachowania modelu zaobserwowano dla wartości 2,3 rzędów mniejszej.

k – stała dodatnia większa od 1. Najlepsze właściwości zaobserwowano dla wartości k=3.

- w wypadku nie odpalenia się modyfikowany jest jedynie próg. Jest on zmniejszany.

W tak zdefiniowanym modelu widać wyraźną synergię neuronów. Współpraca ich prowadzi bowiem do powstania w strukturze sieci automatu zsynchronizowanego z sekwencją wejściową. Model ten został zaprojektowany z myślą o silnej polaryzacji wag neuronów. Po kilku do kilkunastu tysięcy przeliczeniach każdego neuronu (jeśli nie wpadniemy w minimum lokalne) otrzymujemy dwa typy wag synaptycznych

a) (0, 10% W)

b) (90% W, W), gdzie W jest maksymalną możliwą wartością wagi.

Możemy zatem zaznaczyć jedynie zbiór B=$\{w_i: w_i>0.9 *W\}$ na mapie połączeń neuronów, a zbiór A =$\{w_i: w_i<0.1 *W\}$ traktować jako brak połączenia do neuronu za którego odpowiada synapsa z wagą w_i. Próg natomiast wpływa w ten sposób na działanie neuronu, że możliwe jest jego uproszczenie do bramki AND z czasem przeliczenia równym 2 ms. Otrzymujemy więc schemat elektryczny omawianego automatu.

Przy zastosowaniu uczenia ze zmianą progu (IMPL2) w obrębie tego modelu można zaobserwować, że sieć odzwierciedla w swojej strukturze częściej pojawiające się wzorce (dokładna analiza wymienionego przekształcenia znajduje się w rozdziale 5.1.1.1), nie wpada w sprzężenia bezwyjściowe (większość może być przerwana przez zmianę danych wejściowych). Również jeśli chodzi o stopień wykorzystania sieci, to jest on większy (nie ma neuronów, które nigdy nie zadziałają). Negatywnym elementem są zastoje i wzbudzenia sieci przekraczające często 50 cykli symulatora w początkowym okresie uczenia, co możemy zapisać następująco:

$$\exists t \; \exists p \in \langle 0,1 \rangle \mid \exists q > 50 \; \forall j \; \forall k \in (t, t+q) \cap N \quad y_{j,k} = p \tag{10}$$

Podsumowując - model IMPL2, po uproszczeniu zapisów i nie tracący ww. właściwości, możemy przedstawić formalnie w następujący sposób:

$$y_{j,t+1} = \begin{Bmatrix} 1 \Leftrightarrow \sum_{i=0}^{I-1} w_{i,j,t} * x_{i,j,t} > h_{j,t} \\ 0 \Leftrightarrow \sum_{i=0}^{I-1} w_{i,j,t} * x_{i,j,t} \leq h_{j,t} \end{Bmatrix} \qquad (11)$$

gdzie:

$h_{j,t}$ - wartość progu neuronu j w chwili t

$w_{i,j,t}$ - wartość i'tej wagi neuronu j w chwili t

$x_{i,j,t}$ - wartość i'tego wejścia neuronu j w chwili t

$$x_{i,j,t+1} = z_{i,t} \qquad (12)$$

$$z_{i,t} = \begin{Bmatrix} Z_{i,t} \Leftrightarrow i < L \\ y_{i-L,t} \Leftrightarrow i \geq L \end{Bmatrix} \qquad (13)$$

gdzie:

$Z_{i,t}$ – wartość i'tego wejścia do sieci w chwili t

L – ilość wejść do sieci. Kolejne wejścia sieci skojarzone są z kolejnymi indeksami synaps rozpoczynając od najmniejszej.

Połączenia rekurencyjne z wyjść neuronów trafiają na wejściowe synapsy o indeksie równym indeksie neuronu wyjściowego powiększonym o ilość wejść do sieci L.

$$\begin{aligned} &I = L + J \\ &0 \leq j < J \end{aligned} \qquad (14)$$

J – ilość neuronów w sieci

I – ilość synaps neuronu

$$h_{j,t+1}=\begin{Bmatrix} h_{j,t}+\dfrac{\sum_i w_{i,j,t}*x_{i,j,t}-h_{j,t}}{k} \Leftrightarrow y_{j,t}>0 \\ h_{j,t}-a \Leftrightarrow y_{j,t}\leq 0 \end{Bmatrix} \qquad (15)$$

$$w_{i,j,t+1}=\begin{Bmatrix} w_{i,j,t}+c \Leftrightarrow y_{j,t}>0 \\ w_{i,j,t}-c \Leftrightarrow y_{j,t}\leq 0 \end{Bmatrix} \qquad (16)$$

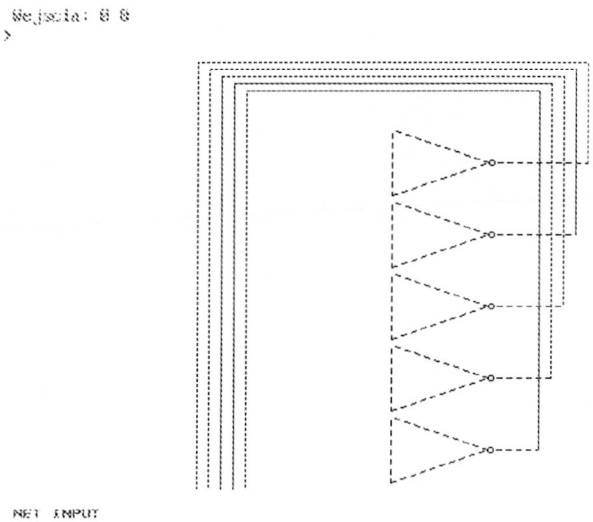
gdzie:

a,c,k – stałe danego modelu o proporcjach jak w (9).

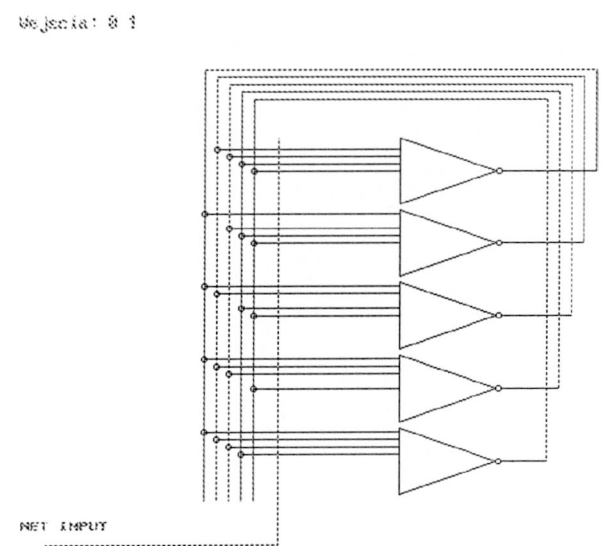
Warunek aktywacji neuronu przedstawia wzór nr 11. Formuła nr 12 i 13 opisuje strukturę sieci oraz mechanizm buforowania pomiędzy wejściem oraz wyjściem neuronów. Modyfikacja wag (wzór nr 16) oraz zmiany progów (wzór nr 15) są elementami mechanizmu uczenia.

5.1.1.1 Przykładowe doświadczenia

Pierwsze doświadczenie dotyczy uczenia ciągu dwuwektorowego. Podczas działania sieci prezentowane były wektory wejściowe {(0,1), (1,0)}. W czasie procesu uczenia, który jest wykonywany równolegle z pracą sieci, możemy zauważyć kilka faz. Będą one kolejno omawiane oraz prezentowane na rysunkach. Każdy rysunek ma podobny wygląd: neurony są reprezentowane jako trójkąty, przy czym narysowane linią ciągłą oznaczają, że w danym kroku symulacji są one aktywne. Dane wejściowe są reprezentowane linią o początku pod napisem NET INPUT i odpowiednim położeniu. W górnej części rysunku są wypisane elementy wektora wejściowego. Synapsy pobudzające są obrazowane okręgami przy wejściu do neuronu, hamujące - kwadratami.



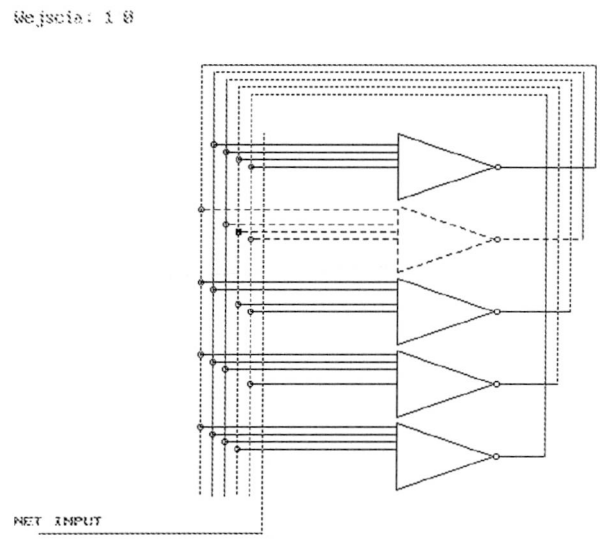
Rys.15. Stan początkowy



Rys.16. Pierwsza faza uczenia

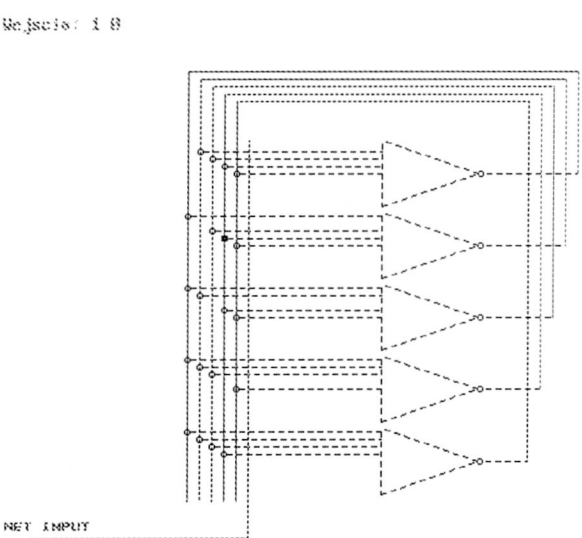
Rys. 15 przedstawia stan początkowy. Neurony nie posiadają połączeń zwrotnych, gdyż wagi wylosowane są z wąskiego przedziału bliskiemu 0. Brak jest sygnałów wejściowych (net input). Neurony

narysowane przerywaną kreską oznaczają, że nie były one aktywne w tym cyklu.



Rys. 17. Przejście do stanu biernego

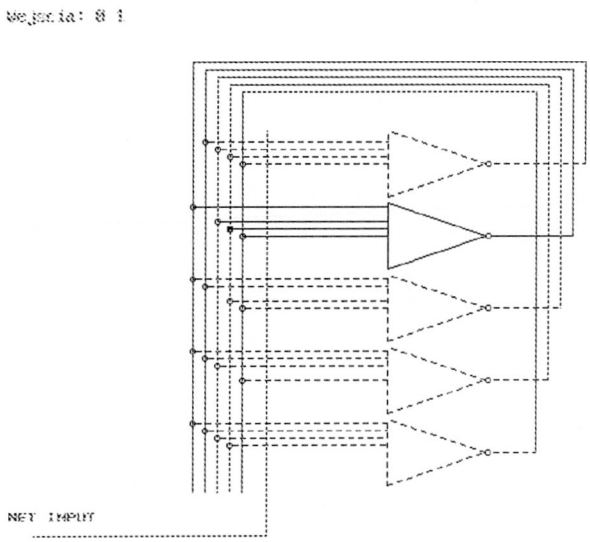
Pierwsza faza uczenia jest zaprezentowana na rys. 16 . Wszystkie neurony są aktywne, wykształcają bardzo dużo połączeń zwrotnych, ignorując jednocześnie wejścia sieciowe, które są aktywne stosunkowo rzadko w porównaniu do częstotliwości aktywacji pozostałych neuronów. Biorąc pod uwagę algorytm uczenia dla neuronu nie ma znaczenia, czy sygnały pochodzą od innych neuronów, czy też są one sygnałami wejściowymi sieci. W tej fazie próg rośnie asymptotycznie do wartości aktywacji neuronu. Gdy proces osiągnie moment, w którym dojdzie do małej różnicy między wartością progu a wartością aktywacji, sygnały wejściowe mają wpływ na aktywację (mogą ją zwiększyć), co pociąga za sobą zwiększenie progu. Brak danego sygnału wejściowego będzie mógł wpłynąć na wyłączenie neuronu.



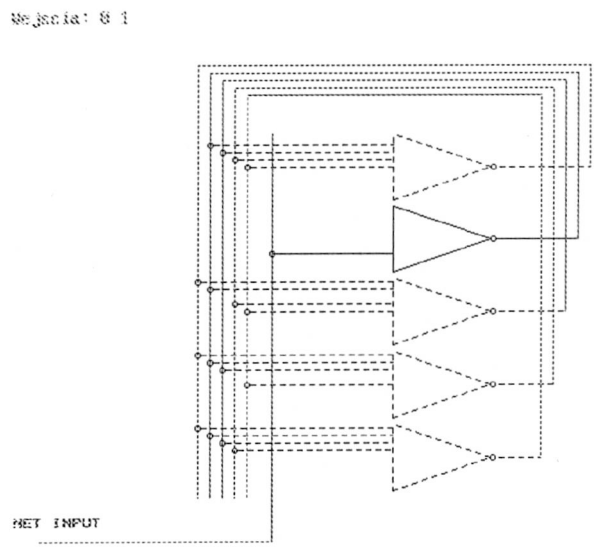
Rys.18. Faza nieaktywności neuronów

Rys. 17 przedstawia przejście ze stanu aktywacji do stanu biernego. Został wyłączony drugi neuron. Duża ilość połączeń zwrotnych pociągnie za

sobą wyłączenie całej sieci. Jednocześnie w tej fazie (przejściowej) następuje uczenie sieci. Wszystkie aktywne neurony (4 w tym przypadku) odnotowują na swoich wagach zachowanie drugiego neuronu.



Rys.19. Włączenie się odpowiedniego neuronu

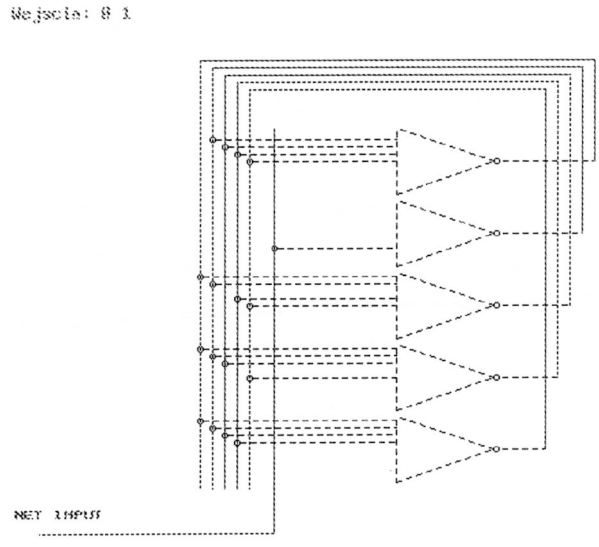


Rys.20. Uwrażliwienie wybranego neuronu na bodziec wejściowy

Rys. 18 przedstawia fazę nieaktywności. Jedynym procesem w tym czasie jest opadanie progów, co następuje powoli, aby drobne różnice w aktywacji neuronów mogły być użyte do uczenia podczas przechodzenia do kolejnej fazy aktywacji sieci. Stosunkowo powolne opadanie progów ma za zadanie rozsynchronizować momenty pierwszych aktywacji neuronów. Wpłynie to korzystnie na zróżnicowanie klas wzorców rozpoznawanych przez różne neurony.

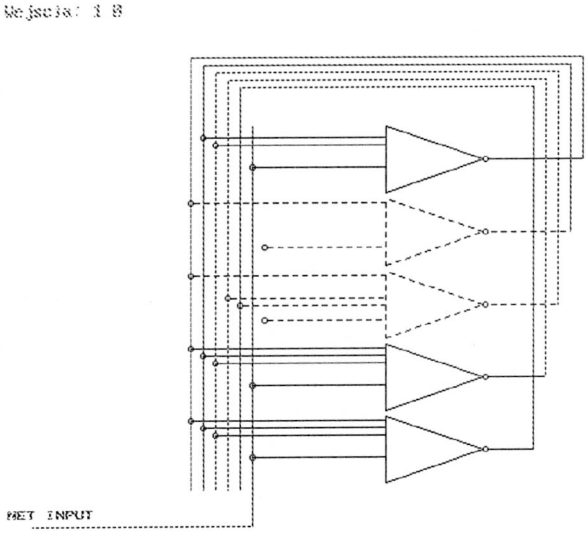
Jeden z neuronów jako pierwszy zaczął się aktywować - rys. 19. Jest on zaznaczony na rysunku jako drugi neuron od góry, jest to ten sam, który najwcześniej się wyłączył. Neuron ten wykształcił połączenie ujemne do neuronu 4, oznaczone prostokątem. Jest to w tym modelu bez znaczenia dla przetwarzania.

Jak widać na rysunku 20 wykształcił się neuron z podłączeniem do wejścia zewnętrznego sieci, gdyż było ono jedynym wejściem z sygnałem dostępnym w czasie aktywacji tego neuronu, w jego stanach działania. Pozostałe neurony pozostawały cały czas nieaktywne, wskutek czego ich wagi nie były modyfikowane. Jest to początek procesu specjalizacji neuronów.



Rys.21. Kolejna faza dłuższej nieaktywacji neuronów

Kolejna faza nieaktywacji jest przedstawiona na rysunku 21. Jak widać sieć powoli się organizuje, tzn. różne klasy rozpoznawanych wzorców pojawiających się w czasie przyporządkowują się różnym neuronom.

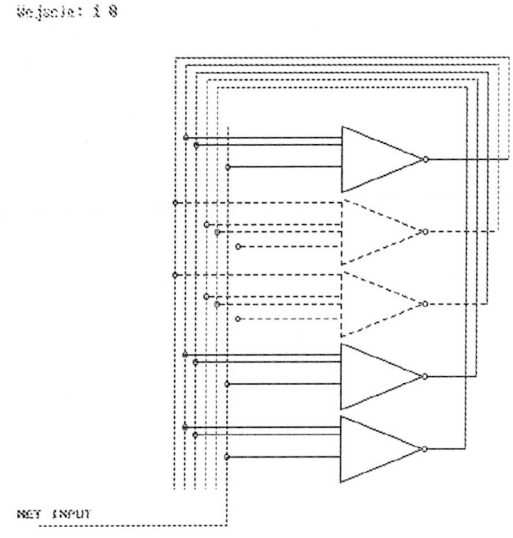


Rys.22. Zakończenie procesu uczenia

Po kilku lub kilkunastu fazach wzbudzeń oraz nieaktywacji, wartości wag ulegają silnej polaryzacji. Oznacza to zakończenie procesu uczenia w tym modelu. Jednocześnie synchronizacja z sekwencją wejściową też jest ustalana podczas uczenia. Na rys. 22 sieć już pracuje, jednak neurony nie są podzielone wyraźnie na dwie klasy. Klasa neuronów charakteryzuje się jednoczesnym pobudzeniem. Jak można

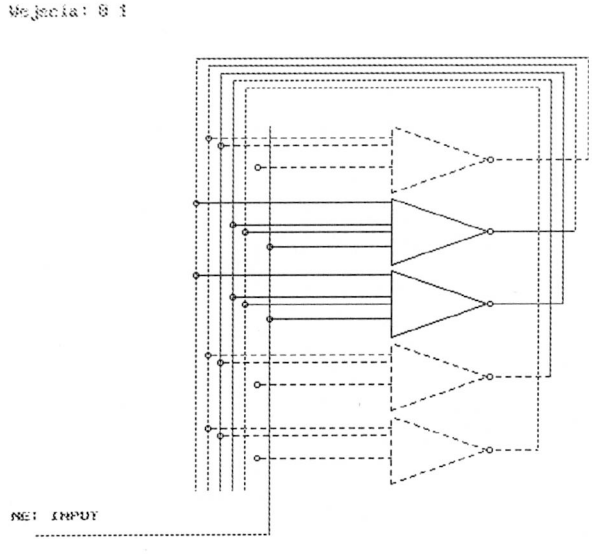
sądzić, pierwsza klasa będzie reprezentowana przez neurony 1, 4 i 5 licząc od góry, klasa druga przez neuron 2 i 3.

Neuron nr 2 ma pewną właściwość bezpośrednio związaną ze swoją klasą. Mianowicie uwrażliwił się na drugie wejście sieciowe, które powoduje jego odpalanie. Neuron 3 natomiast uwrażliwił synapsę połączenia neuronu 5. Ponieważ neuron 5 należy do pierwszej klasy, a neuron 3 odpala się chwilę później i sekwencja wejściowa jest dwuelementowa, to automatycznie neuron 3 zostaje przyporządkowany do drugiej klasy. Mechanizm klasyfikacji w obu przypadkach działał w różny sposób. Kolejne fazy pracy i uczenia spowodują powolne upodobnianie się neuronu 2 i 3 w swojej budowie i funkcji.



Rys.23. Pierwsza faza pracy sieci

Rysunki 23 oraz 24 przedstawiają sieć podczas pracy. Mimo, iż proces uczenia ciągle trwa, nie wpływa to już na dalsze zmiany w sieci. Oznacza to zakończenie procesu uczenia. Neurony zgrupowały się w dwie klasy, dokładnie tyle ma bowiem sekwencja wejściowa. Oba elementy cyklu powtarzają się na zmianę. Gdy zabraknie jednego z sygnałów



Rys.24. Druga faza pracy sieci

wejściowych do sieci, następuje wyłączenia sprzężeń zwrotnych (stan nieaktywacji). Efektem tego doświadczenia jest wykształcenie się automatu zsynchronizowanego z ciągiem wejściowym.

Należy też zwrócić uwagę, że wynik mało zależy od wylosowanych wartości początkowych wag. Proces uczenia jest zbieżny.

Jeśli przeanalizujemy zależność ilości wejść i prędkości uczenia, to ilość wejść w tym modelu może być dowolnie duża. Sieć będzie działała równie skutecznie. Nie jest to pokazane na przykładzie, gdyż byłoby to mało czytelne. Wydłużenie czasu uczenia jest powodowane najczęściej przez wydłużenie sekwencji, zwłaszcza gdy elementy tej sekwencji się powtarzają.

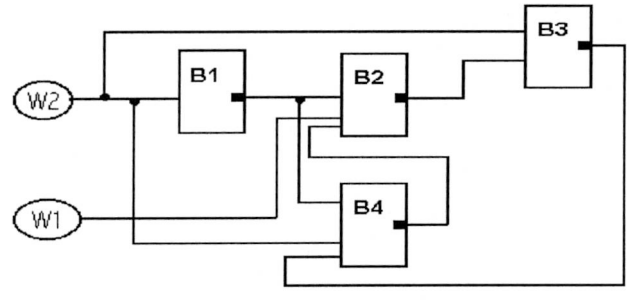
Złożoność przeprowadzonego powyżej doświadczenie jest niewielka, ale pozwoliło ono pokazać procesy zachodzące w sieci podczas uczenia i pracy.

Czas	Wejście 1	Wejście 2
1	0	1
2	1	0
3	0	1

Rys.25. Prosta sekwencja wejściowa z przykładu do IMPL2

Sieć podobnie się zachowuje w przypadku innych, większych problemów. Przykładem prostego automatu powstałego w strukturze IMPL2 może być sieć o podobnej strukturze jak wyżej, na wejścia której podano inną sekwencję wejściową.

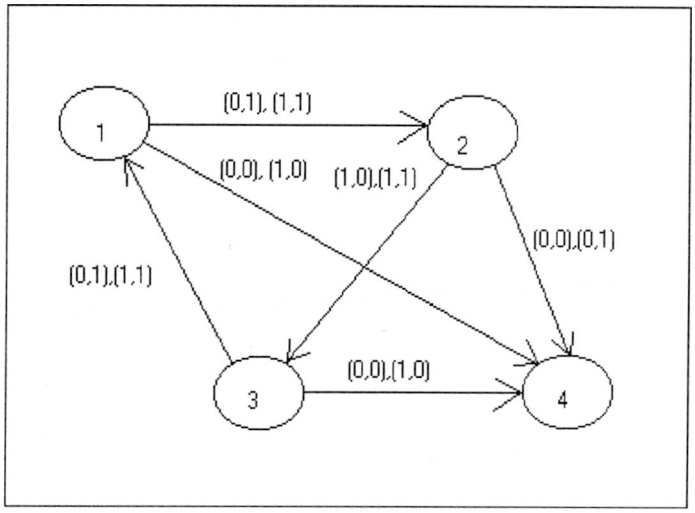
Prosta sieć 5 neuronów i 2 wejść będzie możliwa do przeanalizowania bez stosowania programów pomocniczych.



Rys.26. Schemat elektryczny powstałego automatu

Przykładowa sekwencja trójelementowa jest określona jest w tabeli rys. 25. Schemat powstały w sieci pod wpływem sekwencyjnie pojawiającego się wzorca przedstawia rys 26. Znajdują się tam tylko 4 bramki, ponieważ piąta była nadmiarowa (dublowała bramkę nr 3) i nie została naniesiona dla zachowania przejrzystości rysunku.

Pierwsza bramka B1 tego automatu opóźnia ciąg pobudzeń pochodzących z wejścia W2 tak, aby bramka druga B2 wykryła iloczyn W1(2)*W2(1). W chwili 3 (rys. 27) wynik ten jest obliczony, jednocześnie pojawia się pobudzenie na wejściu W2, co z kolei wykrywa trzecia bramka B3. Czwarta bramka B4 czeka na ten sygnał oraz sprawdza czy ostatnie dwa razy wystąpiło pobudzenie na W2. Pomyślny wynik jest przesyłany na nie omówione dotychczas wejście bramki B2. W ten sposób powstaje zachowanie historii w pętli bramek trzech B2,B3 i B4. Dokładnie bowiem tyle wektorów wejściowych ma przykładowa sekwencja wejściowa.



Rys. 27. Schemat powstałego automatu

Wszelkie możliwe operacje w sieci niezbędne do identyfikacji ciągu wejściowego przeprowadzane są z maksymalną równoległością, co jest cechą bardzo pożądaną. Proces uczenia i przetwarzania jest całkowicie rozproszony. Po dłuższym czasie uczenia nie obserwuje się neuronów nieaktywnych (nie odpalających się nigdy). Dublują one inne neurony, mimo to w tym modelu nie wpływa to na potencjalne zwiększenie odporności sieci na uszkodzenia lub wadliwą pracę, w związku z wysoką polaryzacją połączeń międzyneuronowych.

Model IMPL2 nie wymaga wydłużenia czasu uczenia, gdy zwiększamy ilość wejść. Zależy on natomiast od długości sekwencji wejściowej. Proces uczenia jest zbieżny w ok. 85% dla sekwencji kilkuwektorowych . Należy zwrócić uwagę na to, że pozostawienie części nadmiarowej (więcej neuronów niż faktycznie potrzebnych bramek do zbudowania danego automatu) wpływa korzystnie na skrócenie czasu uczenia. Przeciwnie, jeśli długość sekwencji wejściowej jest większa od pojemności sieci, to generowane są rozwiązania częściowe, wyróżniane klasy sekwencji wzorców uznawanych za poprawne są szersze od docelowych np. jeżeli sekwencja zawiera podsekwencje (a,b,c,d,e) i

b = d to sekwencje (a,b,e), (a,b,c,b,c), (a,b,c,b,e) itp. będą prawdopodobnie również rozpoznawane jako prawidłowe wzorce wejściowe.

W wyniku przeprowadzonych doświadczeń nowego znaczenia nabierają niektóre pytania z zakresu fizjologii. Do jednych z najważniejszych należy stwierdzenie faktu pochodzenia synaps hamowania. Czy hamowanie sygnałów (reprezentowane w tym modelu jako wagi ujemne) może być wykształcone podczas uczenia? Jak wykazują badania tego modelu (zgodnie z jego założeniami), połączenia ujemne są dyskryminowane i zanikające. Może to być wynikiem przyjętej metody uczenia. Na podstawie działania rzeczywistego neuronu wydaje się, że hamowanie synaptyczne powstaje w przypadku, gdy odpali się neuron presynaptyczny i nie odpali się neuron postsynaptyczny. W takim przypadku waga połączenia powinna maleć. Nasuwa się pytanie, czy w tym przypadku sieć nie zacznie zapominać to, co się przed chwilą nauczyła. Gdy sieć pracuje, ale nie rozpoznaje niczego, postępują procesy zapominania wyrażające się w zmniejszaniu wartości wag dodatnich.

Podczas badania wyżej wymienionych modeli sieci, można zauważyć że mają one zdolności do zapamiętywania cyklicznego pojawiania się impulsów w czasie (z wejść ze środowiska lub z wyjść z innych neuronów). Model ten ma natomiast małe szanse nauczyć się rozpoznawania ciągów pojawiających się w sposób asynchroniczny. Taki stan rzeczy spowodowany może być przez przewagę znaczenia stanu sieci (to znaczy impulsów w sieci) nad znaczeniem wartości wag. Wydaje się to być również niezgodne ze wzorcem biologicznym. Podstawowe pytanie dotyczy więc sposobu zapamiętywania informacji w sieci.

Do negatywnych cech tego modelu należy następująca właściwość. Po obniżeniu progów sieć wchodzi w pobudzenie i wtedy uczy się intensywnie. Uczy się, a wcale wtedy nie musi wchodzić na wejście sieci ciąg pojawiający się asynchronicznie, który powinien być zapamiętany. Wejścia te (przeważnie nieaktywne w chwili uczenia) są odłączane. W ten sposób tracona jest ta i tak minimalna zapamiętana informacja na temat rozpoznawanego ciągu. Informacja nie ma szans kumulować się.

Kolejnym zaobserwowanym faktem jest wykorzystywanie przez sieć wszystkich neuronów. Poszczególne fragmenty sieci są dublowane i w ten sposób powstaje nadmiarowość, co w przyszłości należy wykorzystać do bardziej niezawodnej pracy sieci (realizacja w postaci układu elektronicznego).

Obecnie niezawodność z tego powodu nie jest podniesiona (nie ma ona znaczenia, ponieważ zakładamy że pracuje na niezawodnym komputerze).

Podsumowując IMPL2 ma cechy pamięci krótko- i długo- terminowej, której nie posiadał model IMPL1. Pamięć ta nie jest pozbawiona pewnych mankamentów, jak np. niemożliwość rejestrowania sygnałów pojawiających się asynchronicznie. Jednak pamięć krótkoterminowa, prezentowana w tym modelu przez ciąg wektorów wyjść neuronów w kolejnych krokach symulacji, może reprezentować dane wejściowe sieci będące ciągiem pojawiającym się sekwencyjnie. Zmiana danych wejściowych, sekwencji wejściowej oznacza w tym przypadku (zgodnie z tabelą kodowania) niezgodność ze wzorcem, wykrywaną przez sieć. Efektem wykrycia niezgodności ze wzorcem wejściowym jest ciąg wektorów zerowych na wyjściu sieci (wektory wyjściowe są binarne).

Pamięć długoterminowa jest umiejscowiona w wartościach wag synaptycznych międzyneuronowych oraz wejściowych sieci. Wykształca się ona pod wpływem pojawiającego się sekwencyjnie ciągu wektorów uczących. W niej zapisana jest informacja na temat zależności wykrytych przez sieć w ciągu wektorów wejściowych. Sieć używając obu rodzajów pamięci jest w stanie stwierdzić, czy pojawiający się wektor wejściowy należy do wyuczonej sekwencji wejściowej oraz czy pojawił się w odpowiednim momencie (np. czy wektory nie zostały zamienione miejscami).

Z perspektywy czasu można przypuszczać, czy forma przetwarzania zaobserwowana w tym bardzo prostym modelu IMPL2 (jak prawidłowe używanie sprzężeń zwrotnych, spójność jednostki, odrębność części przetwarzającej sieci od części nieaktywnej) nie jest zalążkiem modelu podstawowej jednostki obliczeniowej mózgu[1][141], jeśli nie jest nią sam neuron.

5.1.2 Model IMPL3 - uczenie podczas okresu nieaktywności

Model ten miał za zadanie sprawdzenie jednej z hipotez uczenia sieci, w których uczenie neuronów nie ograniczałoby się wyłącznie do momentów odpaleń neuronów. Neurony mogą się uczyć również podczas okresów nieodpaleń. Został zmodyfikowany algorytm uczenia. Było to również ściśle

[1] tzw. microcircuit

powiązane z problemem niewykształcania w procesie uczenia w poprzednich modelach znaczących wag ujemnych. Zmianą w porównaniu do poprzedniego modelu jest to, że podczas odpalenia wagi ujemne aktywnych synaps są zwiększane i dążą do 0. Natomiast podczas nieodpalenia, wagi aktywnych synaps są zmniejszane (teoretyczna możliwość powstania wag ujemnych).

Algorytm działania neuronu można przedstawić następująco:

a) w przypadku odpalenia następuje

- zwiększenie wag aktywnych synaps

$$x_{i,t-1}>0 \Rightarrow w_{i,t}=w_{i,t-1}+a \tag{17}$$

- zwiększenie wag ujemnych nieaktywnych synaps

$$x_{i,t-1}=0 \wedge w_{i,t-1}<0 \Rightarrow w_{i,t}=w_{i,t-1}+a \tag{18}$$

- zmniejszenie wag dodatnich nieaktywnych synaps

$$x_{i,t-1}=0 \wedge w_{i,t-1}>0 \Rightarrow w_{i,t}=w_{i,t-1}-a \tag{19}$$

b) w przypadku nieodpalenia następuje

- zmniejszenie wag aktywnych synaps

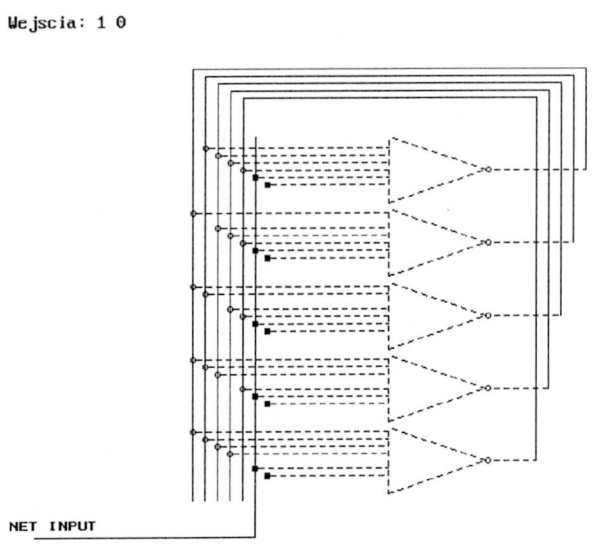
$$x_{i,t-1}>0 \Rightarrow w_{i,t}=w_{i,t-1}-b \tag{20}$$

- pozostawienie bez zmian nieaktywnych synaps

$$x_{i,t-1}=0 \Rightarrow w_{i,t}=w_{i,t-1} \tag{21}$$

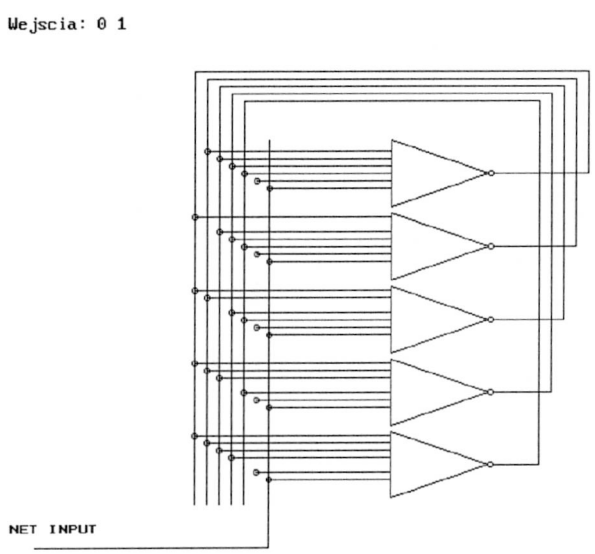
gdzie: a,b - współczynniki uczenia.

W modelu tym jak i w poprzednim możemy wyróżnić występujące po sobie fazy wzbudzeń i nieaktywności w podobnych proporcjach. Podobnie również wyglądały mechanizmy uczenia. Jednak podczas nieaktywności sieci, gdy sieć otrzymywała sygnały z wejścia następowało ich wyłączanie (tzn. wagi były zmniejszane do 0 a następnie do wartości -MIN_WAGI) i ustawianie



Rys.28. Pierwsza z faz cyklu pracy modelu IMPL3

odpowiednich wag na ujemne. Wagi na połączeniach zwrotnych między neuronami pozostawały bez zmian i w ten sposób zachowywały duże znaczenie, co jest wysoce niepożądane. Wagi wytworzone na sprzężeniach zwrotnych mają zbyt duże wartości w porównaniu do wag sygnałów wejściowych sieci, aby następował poprawny proces uczenia. Sieć wykształcała wagi ujemne, ale jej praca była beznadziejna: proces uczenia nie był zbieżny, był okresowy. Podczas aktywacji neuronów połączenia ujemne zanikały i pojawiały się dodatnie, podczas nieaktywacji proces przebiegał odwrotnie.



Rys.29. Druga z faz cyklu pracy modelu IMPL3

Model ten został przekształcony w inny, w którym wcześniej obliczana waga była tylko kierunkiem zmian prawdziwej wagi o jednostkę. Model ten ma dziwne zachowania, po części wydawałoby się chaotyczne i nie został jeszcze dostatecznie przetestowany i przeanalizowany.

Badania tego modelu nie potwierdziły istnienia efektów synergii neuronów. Badania pamięci w tym modelu mają więc wynik negatywny.

5.1.3 Model IMPL4 - interpretacja rozlanego pobudzenia.

Dotychczasowe modele neuronów nie zakładały żadnego związku działania neuronu z otoczeniem. Neuron miał wbudowane funkcje, które miały na celu jego adaptację do danych wejściowych. Tak działał, gdy doprowadzaliśmy do jego synchronizacji z sekwencją wejściową. Nie było za to żadnej możliwości bezpośredniego wpływu na to czego i kiedy neuron będzie

się uczył. Pewne efekty może przynieść połączenie działania neuronu ze środowiskiem oraz wprowadzenie funkcji oceny poprawności działania.
Jedną z metod możliwą do wykorzystania w tym przypadku jest stosowanie rozlanego pobudzenia/ hamowania. Działanie rozlanego pobudzenia można traktować podobnie jak funkcji błędu w metodach wstecznej propagacji, ale jego interpretacja jest inna. Przyjmuje ona wartości ze zbioru {-1,0,+1}. Wartości te oznaczają:

k = -1 - błędne działanie sieci,

k = +1 - poprawne,

k = 0 - brak oceny sieci.

Na podstawie tego sygnału, który musi być generowany spoza sieci neuronowej, następuje modyfikacja procesu uczenia. Sygnał ten ukierunkowuje uczenie w określoną stronę. W tym rozumieniu można sygnał ten interpretować następująco :

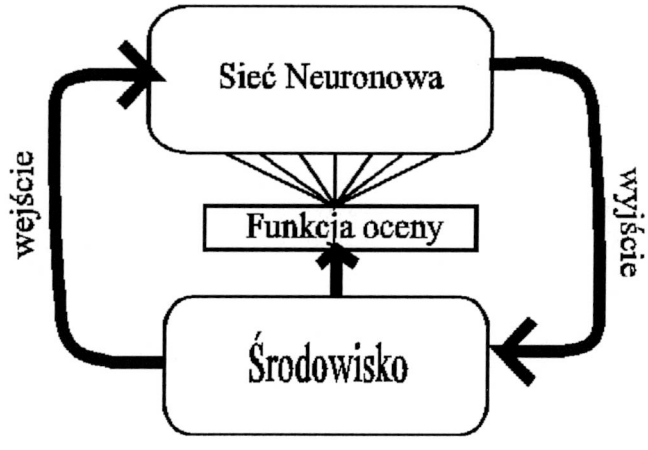
k = -1 hamowanie uczenia właściwego (np. reguła Hebba)

k = +1 przyspieszanie uczenia właściwego

k = 0 obojętny

Być może ważną cechą jest uśrednione działanie rozlanego pobudzenia, tzn. sygnał ten powinien działać z dosyć dużą bezwładnością. Jest to bardziej ogólna ocena działania, a nie ocena konkretnego przypadku.

Analiza wpływu rozlanego pobudzania na proces uczenia jest przeprowadzona poniżej. Ze względu na złożoność algorytmu działania modelu, jest on dostępny w całości w formie elektronicznej. Proces uczenia rozpoczyna się od losowego ustawienia neuronów (losowe wagi ich połączeń). Neurony pracują niepoprawnie. Ważne jest, aby miały szansę odpalić się. Zapewnia to mechanizm podobny jak w IMPL3, polegający na zmianie wag. Neuron zapamiętuje jakie wejścia doprowadziły do jego



Rys.30. Zależność uczenia względem środowiska

pobudzenia tylko wtedy, gdy wartość rozlanego pobudzenia jest większa lub równa 0. Gdy k=+1 zmiana wag będzie o X% większa, co spowoduje faworyzowanie przez sieć takiego stanu. Wpływowi rozlanego pobudzenia podlegają wszystkie neurony sieci, dlatego będą uczone wszystkie neurony odpowiedzialne za powstanie odpowiedzi w sieci.

Należy określić funkcję oceny działania sieci. Funkcję oceny trzeba traktować jako zewnętrzny sygnał oceny generowany na podstawie działania sieci w perspektywie dosyć długiego okresu. Z drugiej strony jest to sygnał kierunkujący działanie sieci, dlatego można go traktować jako znak pochodnej funkcji oceny. Jeśli układ wykazuje poprawę działania do jej poprzedniego stanu, to następuje przyspieszenie uczenia. W tym momencie sieć neuronowa będzie optymalizowała tak swoje działanie, aby funkcja oceny osiągała maksimum. Istnieje oczywiście możliwość znalezienia lokalnego maksimum funkcji oceny (zamiast pożądanego maksimum globalnego), ale gdy założymy, że rozwiązanie nie jest złożonym problem (w celach testowych), można to zbagatelizować.

Implementacja modelu IMPL4 zawiera moduł nadrzędny względem modułu uczenia. Są to pierwsze kroki z rozlanym pobudzeniem. Działanie neuronu było następujące:

Jeśli k=1 i neuron się nie odpali, to się uczy.

Jeśli k=-1 i neuron się odpali, to się uczy, gdzie k oznacza wartość rozlanego pobudzenia.

Uczenie natomiast zostało rozszerzone o modyfikację o progu. Od progu jest odejmowana wartość k. Wagi synaps aktywnych zmieniają się zgodnie z założeniami rozlanego pobudzenia/ hamowania. Po pierwszych próbach okazało się, że sieć jako pierwsza z badanych sieci prawidłowo operowała na wagach ujemnych. Posiadała możliwości wyuczenia się funkcji logicznych niemonotonicznych. Przykładowa funkcja (tutaj negacja implikacji) o skomplikowaniu podobnemu funkcji XOR jest zaprezentowana na rysunku 31.

Lp	wej 1	Wej 2	Wyj
1	0	0	0
2	1	0	1
3	0	1	0
4	1	1	0

Rys.31. Przykładowa funkcja niemonotoniczna

Próby uczenia modelu IMPL4 funkcją XOR

zakończyły się niestety niepowodzeniem. Prawdopodobnie jest to efektem złego rozumienia wpływu rozlanego pobudzenia na neurony.

Po przeprowadzonych doświadczeniach symulowania sieci neuronowych można wyciągnąć dwa wnioski dotyczące modelu IMPL4:

- rozlane pobudzenie rzeczywiście powoduje ukierunkowanie uczenia.
- nie ma tendencji do tworzenia bardziej złożonych układów neuronalnych.

Uczenie tą metodą jest ograniczone tylko do jednego neuronu. Nie następuje podział funkcji neuronów w sieci, lecz każdy próbuje rozwiązywać cały problem sam. Podczas uczenia powinny wykształcać się szlaki neuronalne wielu neuronów rozwiązujących dany problem.

Należy więc opracować nową metodę modyfikacji wag i progu, która wykorzystywałaby skutecznie informacje zawarte w pobudzeniu rozlanym. Najprawdopodobniej model ten zazębia się z kumulowaniem impulsów jako ładunków na wejściach neuronu i wtedy może mieć dopiero pożądane działanie. Działanie rozlanego pobudzenia opiera się ściśle na złożeniu operowania sygnałami średnimi. Wtedy prawdopodobnie możliwa będzie specjalizacja grup neuronów.

W modelu nie występuje pamięć krótkotrwała, ponieważ efekty synergii neuronów są niewielkie, lub ich nie ma (zależy od problemu). Praca neuronu wyjściowego jest stymulowana rozlanym pobudzeniem/ hamowaniem, tak że potrafi on rozwiązywać również problemy funkcji niemonotonicznych. Informacja zawarta w jego wagach może być bardzo prostą formą pamięci długotrwałej na temat korelacji rozlanego pobudzenia oraz danych wejściowych.

5.1.4 Wnioski z doświadczeń oraz wstępne założenia projektowania kolejnych modeli IMPL

Kolejne modele mają za zadanie poprawić błędne założenia i niedociągnięcia modeli sieci wykryte w eksperymentach. Poprawy wymaga sposób zmian wag neuronu, aby umożliwić powstawanie wag ujemnych. W przeciwnym wypadku należy dowieść, że nie są one konieczne. Kolejnym elementem jest rozdzielenie pamięci na dwa odrębne rodzaje pamięci:

krótkotrwałą i długotrwałą. Na tym etapie sieć wydaje się dysponować jedynie pamięcią krótkotrwałą i ulotną. Ulotność polega na tym, że trudno jest utrzymać zapamiętany stan. Każdy niepoprawny impuls powoduje utratę pamięci. Powinien również zostać wyodrębniony proces uczenia się, spośród czasu istnienia sieci. Może to być osiągnięte przez sztywne określenie czasu uczenia lub przez przyjęcie założenia, że neurony będą płynnie tracić możliwość uczenia się wraz z upływem czasu. Drugą metodę zrealizować można przez określenie prawdopodobieństwa zmiany danej wagi. Jeśli nie ulega ona zmianom przez pewien okres czasu to można założyć, że jej wartość jest ustabilizowana i nie powinno się jej zbytnio zmieniać. Kolejną potrzebną rzeczą jest wprowadzenie modelowania upływu czasu do modelu sieci. Umożliwi to modulowanie siły impulsu częstotliwością jego powstawania. Powstawanie aktywacji poprzez odbieranie impulsów będzie uwarunkowane czasem. Impulsy nie będą musiały występować jednocześnie ponieważ pobudzenie neuronu opadać będzie dopiero po pewnym czasie.

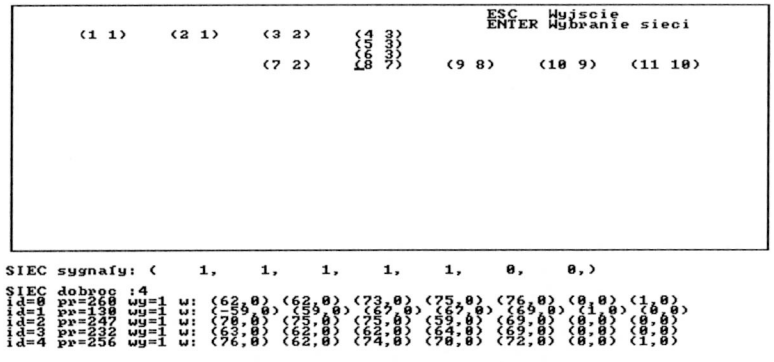
Jak wynika z obserwacji układu nerwowego człowieka, sieć neuronowa powinna przeprowadzać dość wyraźną selekcję sygnałów pojawiających się zbyt często. Sygnały te nie powinny być przesyłane do warstw wyższych. Naturalnym przykładem może być receptor dźwiękowy. Neurony występujące za mechanizmem rozkładu Fouriera bieżącego dźwięku w uchu są odpowiedzialne za określony ton. Jeśli dana częstotliwość trwa dłuższy okres czasu człowiek przestaje na dany dźwięk reagować. Nie jest zauważane też zaniknięcie danego dźwięku. Jest to zachowanie sieci, dające odpowiednie światło na działanie pojedynczego neuronu. Właściwości te powinny posiadać również modele sieci. Powinno to wpłynąć na zmniejszenie znaczenia sprzężeń zwrotnych, a skoncentrowanie się na przetwarzaniu sygnałów pojawiających się asynchronicznie, co jest wysoce potrzebne w tym modelu.

Sieć aby pełnić bardziej skomplikowane funkcje, musi być podzielona na określone części zwane modułami. Można przyjąć taką definicję, że moduł rozpoznaje określony ciąg danych wejściowych, jeśli występuje w nim co najmniej jeden neuron pobudzany tylko po wystąpieniu rozpoznawanego ciągu.

Aby sieć skutecznie rozpoznawała sekwencje niepotrzebna jest równoległość przetwarzania w module. Praktycznie w chwili t przetwarzania określonego ciągu w pojedynczym module znaczenie ma tylko jeden neuron

(jeśli wyeliminujemy nadmiarowość; dotyczy to modeli binarnych). Można to zaobserwować w doświadczeniach. Skonstruowanie odpowiedniego pobudzania neuronów, wymuszonego np. fazami pracy lub położeniem geometrycznym może wpłynąć na bardziej szeregowe przetwarzanie oraz zmniejszenie nadmiarowości. Jeśli te mechanizmy zadziałają to powstaną odpowiednie linie przetwarzania bodźców wejściowych. Jeśli będą one pracować w tym samym czasie symulacji (co jest już wykonane) i będzie występowała korelacja między odpowiednimi liniami przetwarzania, to powinno być to zaznaczone w strukturze sieci po odpowiednim czasie. Aby ukierunkować pracę modułu można użyć globalnego sygnału, oceniającego pracę sieci. Ma on odpowiednik w naturalnej sieci zwany rozlanym pobudzeniem lub rozlanym hamowaniem. Należy wyszczególnić specjalizowane pobudzenia dla każdego modułu. Ma to za zadanie przeprowadzanie selekcjowanego uczenia, tak jak to w naturze selekcjonują odpowiednie neurotransmitery i substancje na nie wpływające (obniżające lub zwiększające aktywację).

Jedną z metod budowania modułów może być położenie geometryczne neuronów w zdefiniowanej przestrzeni. Na pewno nie będzie to 'zwykła' przestrzeń, ponieważ liczenie odległości wymagałoby operacji zbyt długo trwających (pierwiastki). W przestrzeni tej będzie określona funkcja propagacji zaburzenia pola elektrostatycznego, co będzie wpływało na prawdopodobieństwo aktywacji sąsiednich neuronów. Można też ograniczyć ilość synaps do bliższego lub dalszego sąsiedztwa.



Rys.32. Fragment drzewa genealogicznego metody ewolucyjnej
para (xx , yy) oznacza: xx-nr sieci yy-numer przodka sieci

Gdy określony model będzie posiadał pamięć krótko- i długoterminową oraz efektywnie ją wykorzystywał, będzie poddany uczeniu według opracowywanej metody ewolucyjnej (nie zawartej w pracy). Prawdopodobnie w

tej metodzie konieczne będzie wprowadzenie określonych operatorów wymuszających większą modułowość sieci. Pokrótce metodę tę można przedstawić następująco. W procesie wstępnym zostaje ustalona maksymalna liczba potomków danej sieci. Przez potomka rozumie się sieć poddaną modyfikacji w procesie uczenia przez określony czas. Potomek jest podczas działania (i uczenia jednocześnie) poddawany kolejnym testom sprawdzającym jego poprawność (przystosowane do środowiska, rozwiązywanie zadanego problemu funkcyjnego itp.). Testy funkcyjne przeprowadzane są w losowej kolejności, aby sieć nie nauczyła się sekwencji. Zostaje policzona ilość trafnych odpowiedzi na testy - jest to dobroć sieci. Następnie dobroć sieci zostaje porównana z dobrocią innych sieci. Z wszystkich wcześniej wygenerowanych sieci wybrana zostaje sieć, która charakteryzuje się najlepszą dobrocią, pod warunkiem, że maksymalna liczba potomstwa danej sieci nie zostaje przekroczona. Dana sieć może mieć wielu potomków, ale tylko z tego, który ma najlepszą dobroć tworzony jest nowy potomek. Jeśli potomek daje gorsze wyniki od przodka to jest generowany następny potomek tego przodka. Jeśli wszyscy potomkowie dają gorsze wyniki, to tworzony jest następny „brat" przodka. Proces ten można w każdym momencie przeanalizować za pomocą drzewa genealogicznego sieci. Fragment przykładowego drzewa jest przedstawiony na rys. 32.

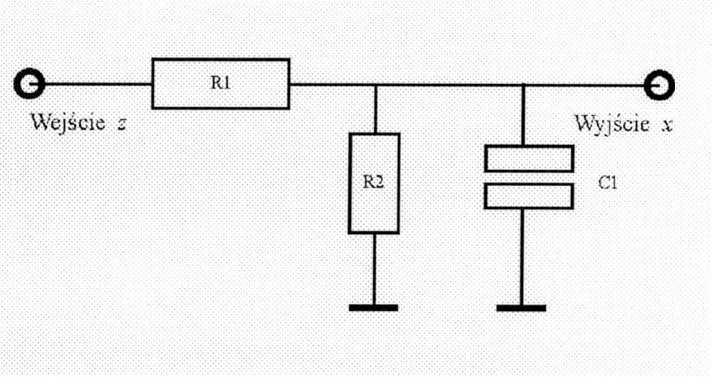
5.2 Modele z symulowaniem przepływu ładunków

Jak wykazują badania, jedynym zjawiskiem które może zmienić wartość wagi synapsy zarówno w kierunku zwiększającym, jak i zmniejszającym o tysiące procent jest zjawisko torowania. Prawdopodobnie jest to podstawowy mechanizm uczenia. Torowanie jest zależne od sygnału wejściowego, jednak uśrednionego w pewnym okresie czasu oraz od rozlanego pobudzenia/ hamowania. Aby symulacja komputerowa tych zjawisk była zbliżona do procesów naturalnych, należy w dużym stopniu zbliżyć się do modelu ciągłego. Nie należy jednak przesadzać z dokładnością modelu, gdyż prawdopodobnie nie wpływa to drastycznie na jakość obserwowanych zjawisk, a może spowodować sytuację, że na w miarę szybkiej maszynie będzie możliwa symulacja 1 neuronu, a nie całej sieci neuronowej. Zjawisko uczenia można

obserwować na przykładzie 1 neuronu, ale fakt dążenia pojedynczego neuronu do określonego rozwiązania nie oznacza, że sieć składająca się z takich elementów będzie jako całość również do czegoś dążyła.

5.2.1 IMPL5 - uproszczony ciągły model komórki nerwowej

Celem prawidłowego zaobserwowania zjawisk związanych z kumulowaniem ładunków w neuronach oraz ich wzajemnych zależności, najlepiej użyć modelu ciągłego. Jednak nie wszystkie zjawiska tego modelu posiadają tę sam ciężar gatunkowy. Aby praca sieci możliwa była do symulowania, przyjęto w IMPL5 kwantowanie czasu co okres równy połowicznemu rozładunkowi potencjału postsynaptycznego. Jest to ok. 1,8 ms. Synapsę można zaprezentować schematem rys. 33. Elementy *R1 i R2* to rezystory, *C1* to kondensator. Określa on nam deformację sygnału przechodzącego przez synapsę, jednak nie wyznacza kształtu sygnału. Upraszczając to przekształcenie przy zachowaniu zdolności sumowania czasowego oraz przybliżonym krokiem czasowym równym 2 ms kumulowanie ładunku postsynaptycznego możemy przedstawić następująco:



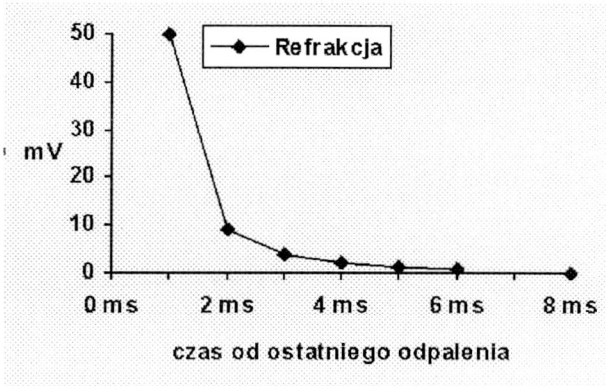
Rys. 33 - Model synapsy - schemat elektryczny

$$x_{i,j,t+1} = \frac{x_{i,j,t}}{2} + z_{i,j,t} \qquad (22)$$

$x_{i,j,t}$ – wartość *i*'tego wyjścia z synapsy neuronu j w chwili *t*

$z_{i,j,t}$ – wartość *i*'tego wejścia do synapsy neuronu j z innego neuronu w chwili *t*

Jeśli próbki sygnału wyjściowego z neuronu w kolejnych taktach przedstawiałyby się następująco: 0, 16, 8, 4, 2, 1, 0, 0, 0, ... to potencjał postsynaptyczny miałby wartości: 0, 16, 16, 10, 6, 4, 2, 1, 0, ..., czyli jego kształt nawet po zamodelowaniu rozładowywania w czasie posiada błąd w początkowej fazie [16,16]. Najlepszym rozwiązaniem jest więc trwanie sygnału wyjściowego przez 1 takt: 50, 0, 0, ... , wtedy potencjał postsynaptyczny przyjmie wartości: 25,12,6,3,1,0.



Rys.34. Refrakcja jako modyfikacja progu wyrażona w mV

W celu zamodelowania zwłoki w pracy komórki nerwowej, równej ok. 2 ms, nie trzeba stosować żadnych dodatkowych elementów w algorytmie, ponieważ neurony przeliczane są co takt, czyli posiadają wymagane opóźnienie. Poza tym przy tej ziarnistości czasu można przyjąć brak zwłoki na propagacji sygnału w aksonie.

Kolejnym elementem zamodelowanym jest refrakcja względna. Wymaga ona odpowiedniej implementacji. Refrakcja jest siłą blokującą pobudzanie się komórki bezpośrednio po pobudzeniu i jest zmienna w czasie. Można ją wyrazić jako chwilowe zwiększenie wartości progu o x mV. W tym modelu w kolejnych taktach będzie ona równa: nieskończoność, 9, 2, 1, 0, (rys. 34) czyli neuron ma duże szanse wzbudzać się co 6ms, a nawet co 4ms, gdyż neuron N_j jest aktywny (odpala się), jeśli w danym momencie zachodzi nierówność:

$$\sum_{i=0}^{I-1} w_{i,j,t} * x_{i,j,t} > h_{j,t} + r(dt_{r,j}) \qquad (23)$$

gdzie:

$h_{j,t}$ - wartość progu neuronu j w chwili t

$w_{i,j,t}$ - wartość i'tej wagi neuronu j w chwili t

$x_{i,j,t}$ - wartość i'tego wejścia neuronu j w chwili t

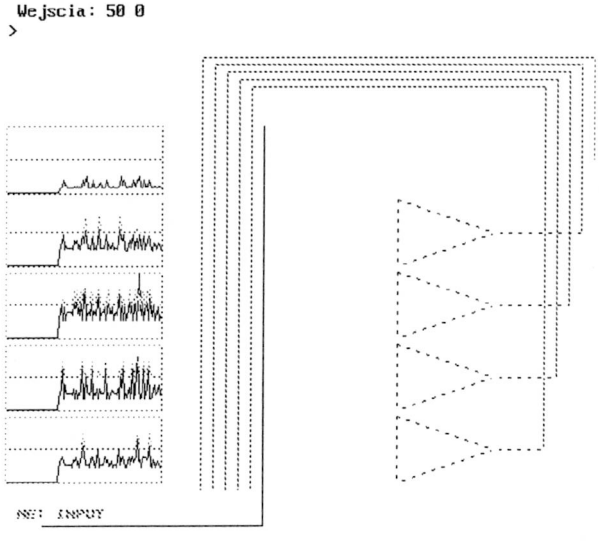
$r(dt_{r,j})$ -wartość funkcji refrakcji, gdy odpalenie neuronu j wystąpiło dt czasu temu. Dla $0<dt<1$ms modeluje refrakcję bezwzględną.

$dt_{r,j}$ - czas, który upłynął od ostatniego odpalenia neuronu j

Refrakcja nie jest jedynym czynnikiem wpływającym na powodowanie przerw pomiędzy odpaleniami określonego neuronu, gdyż podczas odpalenia neuron rozładowuje się tzn. jego potencjał postsynaptyczny jest równy 0 lub nawet ujemny w modelu (tzn. <-59mV wewnątrz komórki). Aby ponownie się wyzwolić, potencjał musi wzrosnąć do wartości ponadprogowej, co musi trwać określony czas.

Wszystkie powyższe wartości parametrów modelu zostały dobrane tak, aby:

- synapsa otrzymująca sygnał z dużą częstotliwością miała potencjalne szanse pobudzenia neuronu,
- występowały sumowania sygnałów różnych synaps tego samego neuronu przesuniętych w czasie,
- pobudzenie neuronu było adekwatne do częstotliwości impulsów wyjściowych.



Rys.35. Efekty drażnienia jednej synapsy z dużą częstotliwością

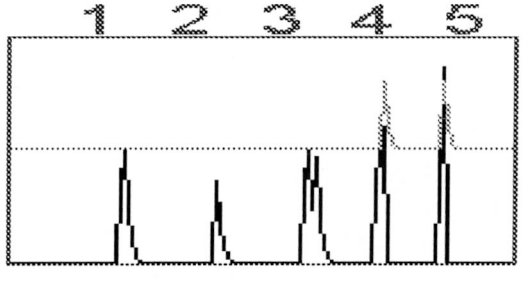
Przykład wzrostu pobudzenia postsynaptycznego przy pobudzaniu tylko jednej synapsy z dużą intensywnością (maksymalną częstotliwością) obrazuje rysunek 35. Semantyka symboli z rysunku jest podobna do poprzednio omówionych. Poza tym z lewej strony każdego neuronu znajduje się oscylogram pobudzenia neuronu oraz wartość sumy progu i refrakcji. Ponieważ wartość progu jest prawie stała w tak krótkim czasie, względne zmiany

oznaczają wpływ refrakcji. Neurony 3 i 4 (od góry, gdzie pierwszy neuron nie jest zaznaczony, ponieważ zawsze był pobudzany podprogowo, zgodnie z odpowiadającym mu oscylogramem) ładowały się dość szybko pod wpływem pobudzeń 6 synapsy (pierwsze wejście z zewnątrz do sieci). Neuron 3 wyzwolił się pierwszy i z powodu wzajemnych sprzężeń w neuronie 4 można zaobserwować wyraźny 'pik' w potencjale postsynaptycznym.

Oscylogram 36 prezentuje inną klasę zależności czasowych. Pobudzane były dwie synapsy jednego neuronu w różnych odstępach czasowych. Na wykresie widnieje 5 akcji.

1. pobudzenie od synapsy B (pierwszej z lewej)
2. pobudzenie od synapsy A
3. pobudzenie synapsy B + 4ms przerwy + pobudzenie synapsy A
4. pobudzenie synapsy B + 2ms przerwy + pobudzenie synapsy A
5. pobudzenie synapsy B + pobudzenie synapsy A

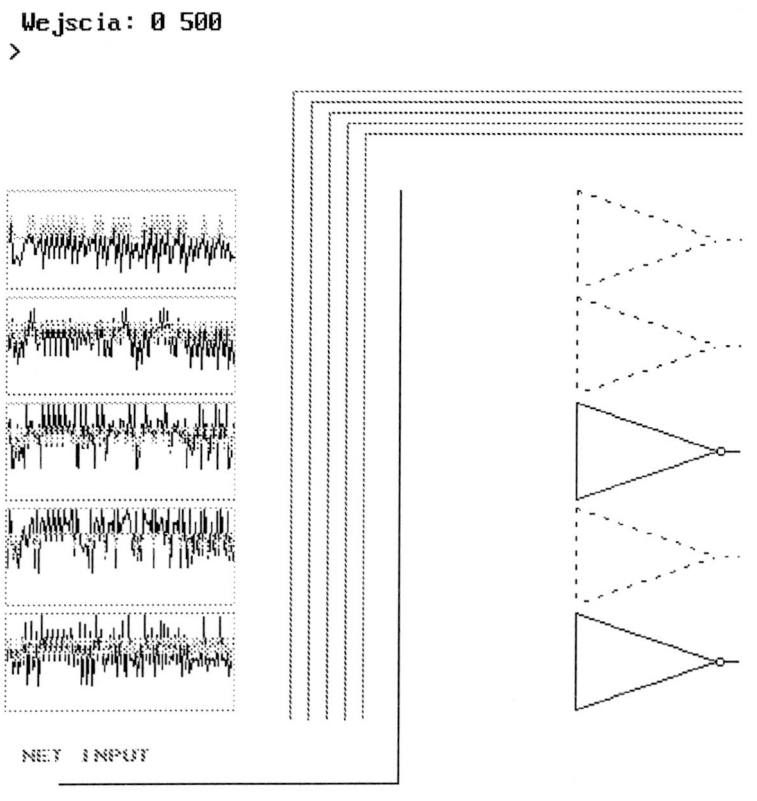
Jak widać na oscylogramie tylko w przypadku 4 i 5 wystąpiło odpalenie neuronu. Granica rozbieżności czasowych pobudzeń wywołujących pobudzenie wynosi ok. 3ms. Sieć ma więc potencjalne możliwości określania przynależności wzorca do określonej klasy abstrakcji również pod kątem tego kryterium.



Rys.36. Pobudzanie dwóch synaps z przesunięciem czasowym

W modelu IMPL5 można zaobserwować nie spotykane wcześniej zachowania sieci. Wszystkie dotychczasowe modele sieci miały tendencje do wpadania w cykle podczas uczenia lub minima lokalne. Dopiero podanie odpowiednio spreparowanych danych uczących (w miarę równomiernie i losowo rozłożonych) zapewniało odpowiednio duże prawdopodobieństwo nauczenia. Okazuje się, że ten model ma zupełnie inne właściwości. Wykazuje zachowania z pogranicza chaosu deterministycznego. Nawet po dłuższym przebywaniu w cyklu uczenia może z niego wyjść. W pracy sieci można zauważyć wyraźnie powtarzające się sekwencje porozdzielane wstawkami chaotycznymi, aczkolwiek deterministycznymi.

Ponieważ sieć pracuje obecnie bardziej w oparciu o częstotliwości niż pojedyncze sygnały -wystąpiła potrzeba określenia odpowiednich metod oceniania sieci. Pierwsza jest nazwana „dobrze/ źle" i jest ona związana z interpretacją pobudzenia rozlanego, gdzie przyjmuje się, że pobudzenie *k(t)* ma trzy stany {-1,0,1}. Odpowiadają one następującym zjawiskom: hamowaniu, zwanemu również blokowaniem, brakowi oceny oraz pobudzeniu. Trudno jednak zgodzić się z taką kategoryczną interpretacją, bowiem sieć neuronowa nawet po osiągnięciu określonego wymaganego poziomu działania może nadal się douczać, co jest pominięte w tej koncepcji. Prawdopodobnie zbadanie mechanizmu powstawania pobudzenia rozlanego może uzupełnić ten model.



Rys.37. Prosta sieć z przykładowymi oscylogramami z elementami chaosu

Inną interpretacją rozlanego pobudzenia może być metoda „lepiej/ gorzej". W porównaniu do poprzedniej metody możliwa jest rozbudowa w dwóch kierunkach:

- pozostawienie obiektu neuronu bez zmian, wprowadzenie mechanizmu różnicującego wyniki do algorytmu oceniającego,
- zmiana algorytmu działania neuronu, związana z inną interpretacją rozlanego pobudzenia (współ. *k(t)*).

Ta druga opcja ma tę zaletę, że możliwe jest wykorzystanie indywidualnych stanów dynamicznych neuronów do modyfikacji mechanizmu

uczenia. Wydaje się, że może to wpłynąć na lepszą współpracę neuronów w rozwiązywaniu określonego zadania.

Przykładowy algorytm uczenia neuronu przy wybraniu drugiej opcji musi zawierać pewne statystyki na temat pracy neuronu. Aby wagi mogły być odpowiednio wyliczane pod wpływem *k(t)* interpretowanym 'lepiej / gorzej', każdy neuron powinien pamiętać swoją średnią częstotliwość odpalania (SCO) oraz ostatnią średnią częstotliwość odpalania (OCO), np. w czasie *(t-d,t)*, gdzie *t*- chwila obecna, *d*- ilość kroków symulatora w których liczona jest OCO. Zmiana średniej częstotliwości odpaleń (ZSCO) jest równa więc różnicy OCO oraz SCO.

$$ZSCO = OCO - SCO \qquad (24)$$

- SCO reprezentuje średnią częstotliwość odpaleń w ostatnich stu kilkunastu cyklach.
- OCO jest średnią częstotliwością odpaleń w ostatnich kilkunastu taktach.
- ZSCO=OCO-SCO i jest zmianą częstotliwości odpaleń.

W zależności od wyliczonego ZSCO możemy przeprowadzać uczenie, zgodnie z zasadami:

- (ZSCO > 0 i k = +1) lub (ZSCO < 0 i k = -1) -> torowanie
- (ZSCO > 0 i k = - 1) lub (ZSCO < 0 i k = +1)-> blokowanie

Udrożnienie polega na zmodyfikowaniu wag aktywnych synaps o wielkość wprost proporcjonalną do iloczynu wejścia *x(t)* oraz *k(t)* liczonym w określonym czasie, oraz obniżeniu progu. Blokowanie jest podobne, lecz próg zostaje podniesiony. Jak wynika z tego algorytmu, synapsy nigdy nieaktywne nie są zmieniane w procesie uczenia. Aby wyuczona sieć nie popadała w dziwne zachowania w zmienionym środowisku musi z czasem następować zanik takich synaps, czyli np. wyzerowanie wag. Proces ten jest również zamodelowany, jego wzorce można odnaleźć w przyrodzie.

Ponieważ z synapsami hamowania presynaptycznego w poprzednich modelach były duże problemy, jest tu zamodelowany hipotetyczny mechanizm pozaneuronowy. W dotychczasowych modelach hamowanie presynaptyczne (o

ile było implementowane) miało być jednym z elementów wykonywanych przez neuron. Realizacja zakładała istnienie wag ujemnych, które w wyniku uczenia zwiększałyby swoje wartości bezwzględne. Mimo sprawdzenia wielu metod uczenia, nie udało się znaleźć algorytmu pracy i uczenia neuronu, który umiałby zarządzać siecią z takimi założeniami. Być może problem tkwi nie w algorytmie, lecz w strukturze sieci. Może wagi ujemne (rozumiane standardowo, że $x_i * w_i < 0$, który to iloczyn powoduje zmniejszenie globalnego pobudzenia komórki) nie są podstawowym typem synaps, lecz miejsce te powinny zajmować synapsy hamowania presynaptycznego. Jeśli tak, to musi być to mechanizm występujący poza neuronem, którego miejsce występowania należy zaznaczyć w algorytmie. Hipotetyczny model hamowania presynaptycznego można przedstawić następująco: Jeżeli waga $w_i < 0$ i w chwili t występuje pobudzenie synapsy nr i, to składnik pobudzenia (liczonego standardowo) $w_{i+1} * x_{i+1}$ nie powinien być wliczany do globalnego pobudzenia neuronu, lub można zastosować zależność liniową. Jeśli w_i dotyczy ostatniej synapsy neuronu to z przyczyn formalnych mechanizm ten nie działa, lub hamuje sygnał x_1 , co z globalnego punktu widzenia nie ma jakościowego znaczenia. Wykształcanie zaś tego typu synaps ma szansę zaistnieć podczas uczenia w procesie torowania, a nie hamowania, jak można by początkowo przypuszczać. Jest to podstawowa nowość względem poprzednich metod. Zarówno synapsy hamowania, jak i pobudzania, wykształcałyby się podczas torowania, podczas blokowania oba typy synaps ulegałyby zanikaniu. Jest to intuicyjnie zgodne z traktowaniem pobudzenia rozlanego jako oceny pracy sieci. Blokowanie oznacza w tym wypadku negatywne tendencje w dążeniu sieci do rozwiązania. Torowanie oznacza ocenę pozytywną, sieć będzie dążyć dalej w tym samym kierunku. Synapsy hamowania presynaptycznego można ustawić w sieci poprzez wylosowanie wag ujemnych (warunkujące hamowanie presynaptyczne) przed procesem uczenia i pozostawić je bez możliwości zmiany znaku. Wartość bezwzględna tych wag jest natomiast zmieniana w procesie uczenia.

W związku z tym, że zaszła potrzeba przebadania kilku wariantów w ramach wyżej przyjętych założeń, z modelu IMPL5 powstały pochodne IMPL5A, IMPL5B i IMPL5C. Zostaną omówione skrótowo dwa pierwsze. Części algorytmów symulacyjnych zastały pogrupowane w moduły podstawowych

algorytmów uczenia. W poniżej przedstawionych doświadczeniach, związanych z metodami uczenia sieci neuronowych oraz pojedynczych neuronów, omówione są różne algorytmy. Mają one jednak pewne części (moduły) wspólne, których określenie ułatwi dalszą analizę. Moduły te są związane z wywołaniem ich w różnych momentach pracy neuronu. Blokowo można to przedstawić następująco:

1. Obliczenie rozładowania się ładunków postsynaptycznych w związku z symulacją czasu.
2. Moduł uczenia nr 1.
3. Obliczenie pobudzenia neuronu, jeśli nie występuje odpalenie to przejście do punktu 6.
4. Moduł uczenia nr 2.
5. Symulacja zmian ładunków podczas odpalenia.
6. Obliczenie niezbędnych statystyk.

Moduł uczenia nr 1 to zasadniczy element modelujący zjawisko torowania. Występuje on niezależnie od odpalania neuronu. Działanie jest ściśle związane z przekazanym z zewnątrz sieci parametrem k - rozlanym pobudzeniem/ hamowaniem. Jednak ostateczna interpretacja parametru k nie mieści się w tych granicach. Jednakowo ważny wpływ na mechanizmy uczenia każdego neuronu mają lokalnie obliczane statystyki, które to dopiero wraz z parametrem k przesądzają o kierunku uczenia. Udrożnienie przepływu ładunków następuje, gdy spełniony jest warunek:

$$k>0 \wedge zsco \geq 0 \vee k<0 \wedge zsco<0 \qquad (25)$$

zaś blokowanie, gdy:

$$k>0 \wedge zsco<0 \vee k<0 \wedge zsco \geq 0 \qquad (26)$$

Moduł uczenia nr 2 jest uaktywniany podczas odpalenia neuronu. Ma on za zadanie wyłączanie synaps, które prawie nigdy nie są aktywne, a wagi na nich mają na tyle duże wartości, że przypadkowe sygnały mogłyby zakłócać pracę. Jest to uzupełnienie modułu pierwszego, który zajmuje się tylko synapsami aktywnymi.

5.2.2 IMPL5A - rozlane pobudzenie, wykorzystanie historii odpaleń neuronu

Model ten wyposażony jest jedynie w moduł uczenia nr 2, czyli posiada następujące założenia:

- uczenie tylko podczas odpalenia neuronu,
- uczenie tylko synaps pobudzanych,
- brak modyfikacji progu,
- uczenie na podstawie rozlanego pobudzenia / hamowania (w zależności od historii każdego neuronu),

Doświadczenia przeprowadzone na kilku neuronach nie były jednoznaczne, sieć nie wpadała w sprzężenia bezwyjściowe lub stany 'ciszy', ale jej funkcjonowanie nie dążyło do rozwiązania. Efekty synergiczne nie zostały dostrzeżone. Nie można zatem mówić o występowaniu pamięci krótko- ani długotrwałej w tym modelu.

5.2.3 IMPL5B- moduł uczenia podczas odpalenia oraz podczas nieaktywności

Model ten został wzbogacony w stosunku do IMPL5A o moduł uczenia nr 1. Nieaktywne synapsy mają szansę zaniknąć podczas aktywacji neuronu. Progi nadal nie są zmieniane. Wyznaczanie współczynnika *k* dla sieci nie jest sprawą łatwą. W tym modelu wygląda to następująco (jeśli chodzi o odwzorowania funkcyjne):

$$\begin{aligned} zo>0 \wedge zsco<0 &\Rightarrow k=-1 \\ zo>0 \wedge zsco>0 &\Rightarrow k=1 \\ zo<0 \wedge zsco<0 &\Rightarrow k=1 \\ zo<0 \wedge zsco>0 &\Rightarrow k=-1 \\ zo=0 \wedge zsco<0 \wedge n>0 &\Rightarrow k=-1 \\ zo=0 \wedge zsco\geq 0 \wedge n>0 &\Rightarrow k=1 \\ zo=0 \wedge zsco<0 \wedge n=0 &\Rightarrow k=1 \\ zo=0 \wedge zsco\geq 0 \wedge n=0 &\Rightarrow k=-1 \end{aligned} \qquad (27)$$

gdzie:

- *zo(t) = zo(t-2) - zo(t-1)* - zmiana oczekiwanego wyjścia (teoretycznego) przy zmianie danych wejściowych
- *n* - sygnał wyjściowy oczekiwany ze zbioru {0,1}
- *k* - ustawienie współczynnika rozlania *k*

W momencie rozpoczynania pracy sieci pojawia się sztucznie wprowadzony okres, w którym zablokowane są procesy uczenia, a jedynym zadaniem w tym czasie jest ustalenie się statystyk *SCO* i *OCO*. W poprzednich modelach nie występowały fazy takie jak: działanie i uczenie, ale były one jednoczesne. Tutaj sytuacja nieco się zmienia. Występują na zmianę fazy uczenia i testowania(działania), ale uczenie wykonywane jest na działającej sieci. Algorytm wygląda następująco:

1. wprowadzenie losowych danych wejściowych (ustalanie statystyk),
2. podawanie określonych danych wejściowych (testowanie),
3. wprowadzenie rozlanego pobudzenia (uczenie) oraz praca,
4. powrót do punktu 2 lub 1,

Ważne jest, aby punkt 3 występował bezpośrednio po 2, i nie było w tym miejscu innych działań.

Model ten ma problemy w uczeniu funkcji AND (iloczynu logicznego). W modelu IMPL2 nie następowało prawidłowe uczenie neuronów, które powinny wykonywać operację OR (sumę logiczną). W tym modelu jest odwrotnie. Bramki OR wykształcają się, natomiast nie występują AND'y. Jest to związane z przyjętą logiką sieci. O ile w IMPL2 było to łatwo wytłumaczalne (asymptotycznym dążeniem progu do aktywacji) to tu można próbować wyjaśnić ten problem następująco: punkty 1-4 opisują stany, w jakich może znaleźć się neuron podczas pracy. Sytuacje te są równie prawdopodobne. W każdym punkcie następuje pewna modyfikacja wag wejściowych, w zależności od danych wejściowych oraz *ZSCO*, zgodnie z algorytmem uczenia.

1. wejście = (0,0) - neuron się nie uczy, bo (a) nie ma aktywnych synaps, (b) neuron się nie odpala,
2. wejście = (1,0)

- jeśli się odpala, to zmniejszane są obie wagi,
- jeśli się nie odpala, to zmniejszana jest waga przy aktywnej synapsie,

3. wejście = (0,1)

- jak w punkcie 2,

4. wejście = (1,1)

- wzmacniane są oba wejścia, czyli zwiększane są wartości wag obu synaps

Po dłuższym czasie szanse zwiększania się wag synaptycznych są nikłe, bo średnia zmiana wagi zgodnie z punktami 1-4 wynosi

$$0 * \frac{1}{4} - \frac{1}{4} * 1 - \frac{1}{4} * \frac{1}{2} + \frac{1}{4} * 1 < 0 \qquad (28)$$

co jest potwierdzalne doświadczalnie.

Testowanie wpływu torowania i blokowania przeprowadzono w kilku wariantach. Aby ocenić skrajne warunki pracy sieci przetestowano ją przy trwającym cały czas pobudzeniu rozlanym oraz w drugim przypadku, przy rozlanym hamowaniu. W pierwszej klasie doświadczeń wszystkie sieci niezależnie od stanu początkowego po dłuższym czasie wpadały w sprzężenia zwrotne. W pierwszej fazie każdy neuron ustawił swoje wagi na wartości maksymalne lub zero. Gdy *ZSCO* po pewnym czasie osiągnęło 0 wszystkie wagi ustawiły się na wartości zbliżone do maksymalnych przyjętych w modelu. W drugiej klasie sieć zachowywała się aperiodycznie, nie dążyła do całkowitego wygaszenia. Jednocześnie neurony nie wykształcały określonych rozpoznawanych wzorców, ale ich wagi oscylowały w okolicach zera. Jest to prawidłowe, gdyż neurony dążyły do zmiany swojego *ZSCO*. Innymi słowy, gdy w neuron zaczął odpalać się rzadziej, to dążył do zwiększenia tej częstotliwości i odwrotnie. Powyższe fakty świadczą o statystycznie prawidłowym oddziaływaniu rozlanego pobudzenia na neurony i pracę sieci, zgodnie z założeniami.

Omawiany model jest kolejnym pośrednim krokiem w kierunku budowy maszyny z pamięcią. Wykazuje on pewne właściwości w przekształcaniu sygnału wejściowego na jego wewnętrzną reprezentację, nie wykorzystując przy tym jeszcze mechanizmów związanych z modyfikacjami progów. Sieć potrafi podobnie jak model IMPL2 odpowiedzieć sekwencją zsynchronizowaną z ciągiem wejściowym, nie dochodząc do osiągnięcia synchronizacji za pomocą mechanizmów uczenia (IMPL2), ale wynika to już z jej zasady działania. Wpływa to znacznie na szybkość procesu synchronizacji. Najczęstszą niedoskonałością jest tu wpadanie w bezwyjściowe sprzężenia zwrotne (ok.25% doświadczeń).

Analizując pamięć długoterminową w tym modelu, należy zauważyć że posiada tę samą ideę co IMPL2, jednak inną budowę. Różnica polega na tym, że IMPL2 operował logicznymi bramkami AND, ten zaś model - bramkami OR.

Potencjalne możliwości obu modeli są równe, ponieważ stosujemy przyjętą wcześniej tabelę kodowania sygnałów wejściowych. Problem dotyczy w tym przypadku sygnałów międzyneuronowych. Czy sieć może na nich równie skutecznie operować jak w IMPL2? Trudno jest to porównać, ponieważ rozważany model jest ciągły (mimo iż przesyłane informacje międzyneuronowe są binarne) i nie jest sprawą oczywistą przekształcanie go w cyfrowy schemat elektroniczny.

Pamięć krótkoterminowa jest reprezentowana w tym modelu przez rozkład potencjałów postsynaptycznych oraz stany neuronów (np. stany aktywacji). W porównaniu do IMPL2 ma tę przewagę, że z powodu możliwości gromadzenia potencjałów podprogowych szybciej zyskuje synchronizację z danymi wejściowymi. Pojawia się natomiast problem przedefiniowania samej istoty reprezentowania danych przez automat neuronowy. Ogólnie mówiąc, ciągi wyjściowe nie tworzą zdefiniowanego ciągu, ale można ustalić prawdopodobieństwa określonych ciągów wyjściowych jako odpowiedzi na odpowiednie ciągi wejściowe. Sieć realizuje określone przekształcenie.

W czasie doświadczeń okazało się, że podczas zmian w wielkości sieci zmieniają się właściwości pracy neuronów. W przypadku większych sieci bowiem refrakcja ma statystycznie zbyt małe znaczenie. O ile z badań biologicznych wynika, że refrakcja powinna wynosić 9,2,1,0 mV (w kwantach czasu tego modelu = 1,8 ms), to druga strona w tej nierówności akt>prog+refrakcja, gdzie 'akt' to globalne pobudzenie neuronu liczone jako

$$akt = \sum_{i \neq id} w_i * x_i / W \qquad (29)$$

gdzie:

0<i<ILOŚĆ_SYGNALOW +1 - ilość danych wejściowych sieci,

id - numer aktualnego neuronu,

x_i - wartość potencjału postsynaptycznego,

w_i - wartość i-tej wagi,

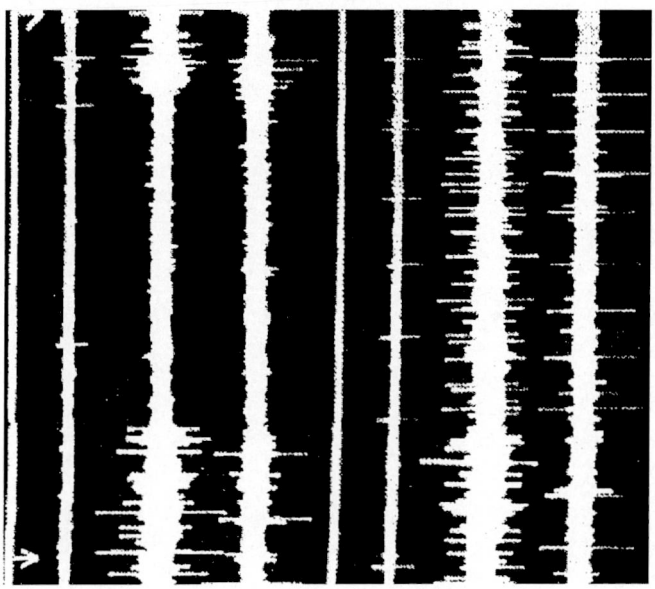
W- maksymalna dopuszczalna wartość wagi

nie odpowiada biologicznemu pobudzeniu, ze względu na sumowanie geometryczne na błonie komórki potencjałów pobudzenia całkowitego [158]. Statystycznie suma uwzględniająca geometrię błony tych potencjałów jest na

pewno mniejsza. Lepszym przybliżeniem Natury będzie kolejne przeskalowanie:

$$akt\_sk=akt / ILOSC\_SYGNALOW; \qquad (30)$$

Ważnym elementem tej pracy jest uwzględnianie w modelu zjawisk fizjologicznych. Można więc teraz próbować odnieść zachowania modeli IMPL do statystycznych danych fizjologicznych z rozdziału pierwszego. Modele IMPL mają pewne cechy naturalnych sieci neuronowych. Oscylogramy zebrane z IMPL5 świadczą niewątpliwie o wielu niepożądanych procesach, np. jest zbyt wiele korelacji w całej sieci, czyli sprzężeń zwrotnych dodatnich. Jeśli w naturalnych sieciach neuronowych lekarze obserwują takie makroskopowo obserwowalne zachowania w EEG (rys. 38), to są one objawem patologicznym. Należy tu zauważyć, że sieć przechodzi w stany wysokiej aktywacji a następnie długich wyciszeń. Przechodzenie między tymi stanami jest konieczne, lecz nie powinny one trwać tak długo. Rozwiązaniem tego problemu jest prawdopodobnie wbudowanie 'zegara neuronowego' (na wzór układu siatkowatego) oraz uwzględnienie centralnego sterowania układu limbicznego. Sam fakt istnienia neuronowego zegara i stabilizatora częstotliwości mówi dużo o budowie pojedynczego neuronu oraz może zweryfikować przyjętą kwantyzację czasu.



Rys.38. Prawidłowe oscylogramy komórek nerwowych

Jak wynika z przeprowadzonych doświadczeń zmniejszenie nacisku na uczenie w momencie aktywności neuronu zmniejszyło faktycznie

prawdopodobieństwo stanów bezwyjściowych (tzn. takich, w których neurony wcale się nie odpalały, albo cały czas odpalały się ze swoją maksymalną częstotliwością). Zniszczone zostały jednak efekty synergiczne w sieci, czyli jedna z najistotniejszych własności sieci neuronowej. Uczenie podczas odpalania neuronów wydaje się więc być podstawową i nienaruszalną zasadą. Potwierdzają to również badania neurofizjologów. Jakie więc typy uczenia w odniesieniu do określonych struktur neuronowych należałoby rozważyć, aby nie ominąć żadnego ważnego aspektu? W kolejnych modelach zaprezentowano kilka możliwych koncepcji podziału.

5.2.4 Model IMPL6 - kilka koncepcji synaps

W związku z potrzebą usystematyzowania algorytmów uczenia, można je pogrupować względem struktur, na których one operują. Zatem algorytm uczenia może być związany z:

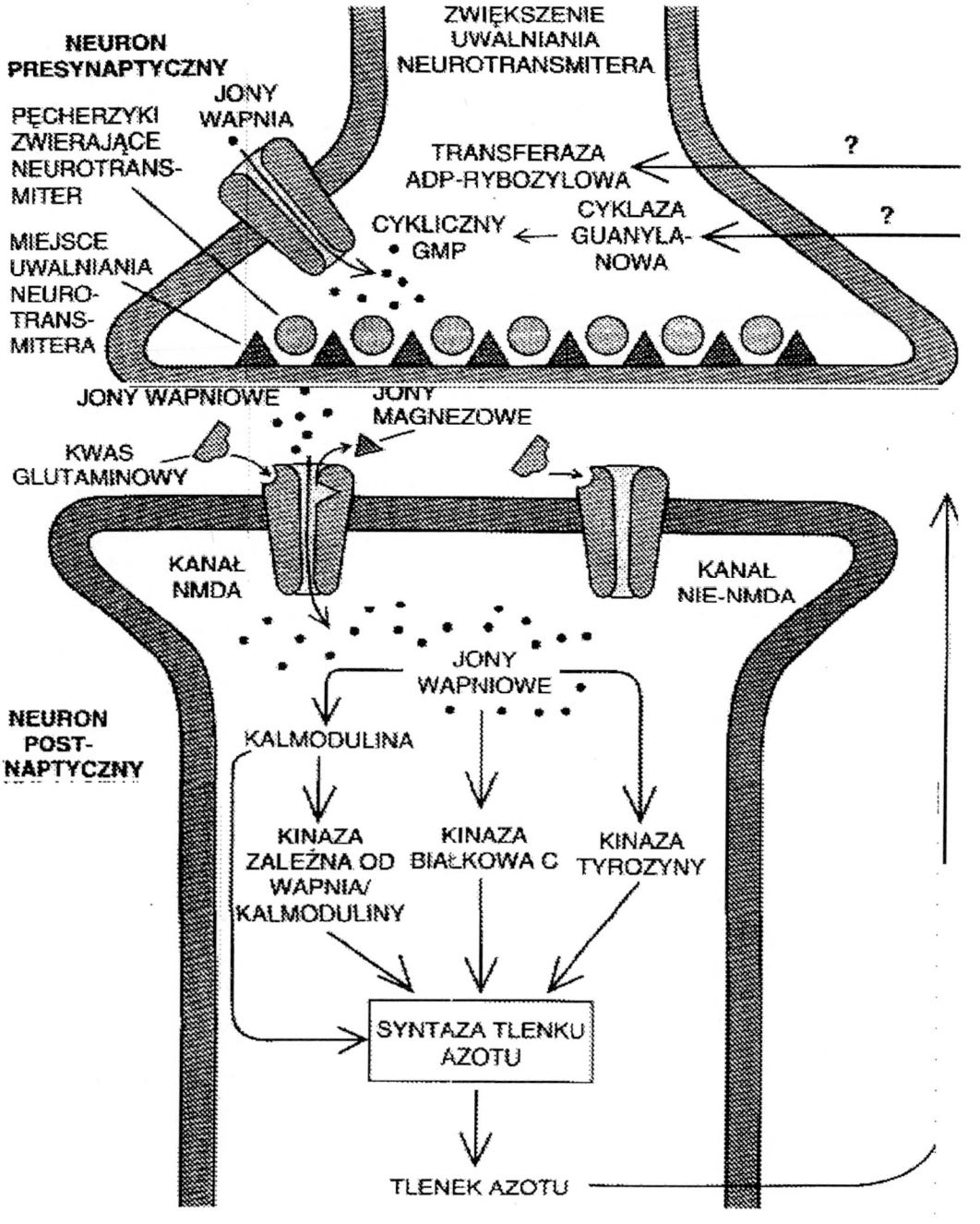
- Synapsą hamującą (typu komórki Renshawa). Pełni ona znaczącą funkcję w stabilizowaniu częstotliwości. Ani reguła Hebba, ani proponowane przez badaczy modyfikacje, nie gwarantują skutecznego uczenia, które wpływałoby na zwiększenie synergii w pracy neuronów.
- Synapsą pobudzającą. Tutaj model wydaje się być najprostszy. Algorytmem uczenia w tym przypadku może być reguła Hebba.
- Ułatwianiem presynaptycznym. Jest to synapsa w której występują trzy możliwe drogi przepływu impulsu. Dwie wejściowe i jedna wyjściowa. Jest to stosunkowo dobrze zbadane i jest również określony częściowo mechanizm uczenia. O ile jest znany wpływ neuronu modulującego na zmianę wagi synapsy neuronu czuciowego i postsynaptycznego, to nieopisany jest proces wykształcający wagę w synapsie aksonowo -aksonowej.

W tym momencie warto omówić rolę tlenu azotu w procesie uczenia (i związanych z nim reakcji), gdyż niewątpliwie jest on związany z synapsą ułatwienia presynaptycznego, a możliwe, że jest to proces o jeszcze szerszym zasięgu. Zamodelowanie tego procesu mogłoby się przyczynić do powstania pewnej cechy samoorganizowalności (na podobieństwo SOM) oraz prymitywnego kojarzenia faktów posiadających korelację czasową. Sam

hipotetyczny mechanizm wygląda następująco: Długotrwała depolaryzacja błony znosi blokadę kanału jonowego, przez który napływają jony wapniowe do neuronu postsynaptycznego. Wzrost poziomu wapnia powoduje aktywację kinaz białkowych i w rezultacie indukcji LTP. Neuron postsynaptyczny wypuszcza wsteczny przekaźnik - tlenek azotu, który to z kolei działa wzmacniająco na pracę zakończeń presynaptycznych, ponieważ wydzielana jest większa ilość neurotransmitera (kwasu glutaminowego). Wydzielany jest on pod wpływem cyklazy guanylowej albo transferazy ADP-rybozylowej. Jest to zobrazowane na rysunku 39.

Nawet jeśli ten mechanizm okaże się nieprawdziwy, to i tak musi być jakiś jego odpowiednik spełniający te podstawowe funkcje, jak plastyczność, samoorganizowalność, ułatwienie presynaptyczne. W tej pracy mechanizm ten z powodu niekompletności wzorca do modelowania, nie jest symulowany.

- Hamowaniem presynaptycznym. Budowa synapsy jest podobna jak w powyższym punkcie. Reakcje chemiczne są również podobne, ale powodują hamowanie. Czyżby więc jeden typ anatomiczny spełniał dwie funkcje? Pytanie pozostaje na razie bez odpowiedzi. Nie znane są też odpowiednie metody uczenia. Być może są one powiązane z ułatwieniami presynaptycznymi.



Rys.39. Hipoteza dwukierunkowości synaps

Jak można zauważyć, zarysowuje się w tych wszystkich elementach pewna symetria. Wszystkie te przypadki występują w układzie nerwowym i potrafią dostrajać się (uczyć) w tym środowisku. Po dokładniejszym przeanalizowaniu tych przypadków, będzie można się pokusić na określenie pewnej ogólnej metody uczenia (o ile nie koliduje ona z uwarunkowaniami genetycznymi).

Główna część procesu uczenia w tym modelu została z powrotem przeniesiona na moment odpalania neuronu, a jako całość - sieć posiada zalety poprzednich modeli (tzn. nie wpada w stany bezwyjściowe sprzężenia zwrotnego lub zaniku impulsów, a jeżeli to w nieporównywalnie małym stopniu). Uczenie ma zalety typu IMPL2, sieć potrafi się organizować. W wielu doświadczeniach okazywało się, że nawet po odłączeniu sygnałów wejściowych po sieci krążyły impulsy, co nie zdarzało się w IMPL2. Jeśli chodzi o korelację zmian średnich częstotliwości odpaleń, to została ona w dużym stopniu zmniejszona. Zalety te zostały uzyskane za pomocą modelowania lokalnej synapsy hamowania presynaptycznego z mechanizmem uczenia zapożyczonym z identycznego elementu anatomicznego, ale o innej funkcji: synapsy ułatwiania presynaptycznego. Pozostałe synapsy uczą się według zmodyfikowanej dla wag ujemnych reguły Hebba, przedstawionej poniżej.

Znaczenie symboli użytych w poniższych wzorach:

i- liczba naturalna

$w_{i,t}$ - i'ta waga neuronu w chwili *t*,

$x_{i,t}$ - sygnał wejściowy neuronu odpowiedni i'tej wadze w chwili *t*,

L - ilość sygnałów wejściowych neuronu,

S,-*S* – maksymalna i minimalna wartość wagi,

M - stała określająca wielkość modyfikacji wag,

Moduł nr 2 przeprowadza proces zmniejszania wag nieużywanych:

$$\begin{Bmatrix} \forall i<L \;\; x_{i,t}=0 \;\wedge w_{i,t}>0 \;\Rightarrow w_{i,t+1}=w_{i,t}-1 \\ \forall i<L \;\; x_{i,t}=0 \;\wedge w_{i,t}<0 \;\Rightarrow w_{i,t+1}=w_{i,t}+1 \end{Bmatrix} \qquad (31)$$

Moduł nr 1 – udrażnianie:

$$\begin{Bmatrix} \forall i<L \;\; x_{i,t}>0 \;\wedge w_{i,t}>0 \;\wedge w_{i,t}\leq S-M \Rightarrow w_{i,t+1}=w_{i,t}+M \\ \forall i<L \;\; x_{i,t}>0 \;\wedge w_{i,t}<0 \;\wedge w_{i,t}\geq -S+M \Rightarrow w_{i,t+1}=w_{i,t}-M \end{Bmatrix} \qquad (32)$$

Fragment odpowiedzialny za uczenie synaps aksonowo - aksonowych znajduje się w części kodu programu nie związanego z czasem odpalania neuronu. Neuron postsynaptyczny nie musi się w tym przypadku odpalać, a to on jest odpowiedzialny za modelowanie lokalnego hamowania. Aby nastąpiło uczenie

według tego schematu wystarczy aktywacja neuronu modulującego (tu hamującego) oraz pobudzającego. Można to przedstawić następującą zależnością:

$$\forall i<L-1 \quad x_{i,t}>0 \ \wedge x_{i+1,t}>0 \ \wedge w_{i,t}<0 \ \wedge w_{i,t}\geq -S+M \wedge w_{i+1,t}\leq S-M \Rightarrow \\ w_{i+1,t+1}=w_{i+1,t}+M \wedge w_{i,t+1}=w_{i,t}-M \tag{33}$$

Model IMPL6 posiada lepsze właściwości synergiczne neuronów niż IMPL5, skutkiem czego pamięć krótko- jak i długoterminowa ma lepsze właściwości tzn. na jej wykształcenie potrzeba mniej czasu symulacji oraz znacznie większy procent doświadczeń jest udanych (głównie dzięki modelowaniu synaps hamowania presynaptycznego z metodą uczenia zaczerpniętą z synapsy ułatwiania presynaptycznego). Typ oraz budowa tych pamięci są identyczne jak w IMPL5. W tym modelu jako pierwszym można zaobserwować pamięć krótkotrwałą utrzymującą się pewien czas, również po skończeniu rozpoznawanej sekwencji wejściowej sieci.

5.2.5 IMPL7 - próba zebrania pozytywnych cech modeli ciągłych IMPL

Oprócz elementów wspólnych dla modeli ciągłych IMPL7 zawiera symulację:

- dwóch typów synaps,
- odpowiednich dla danych struktur metod uczenia oraz pracy,
- okresowego wyłączania sprzężeń zwrotnych,
- refrakcji względnej i bezwzględnej.

Ładunek postsynaptyczny rozładowuje się według funkcji wykładniczej, zgodnie z założeniami modelu ciągłego. Jeśli synapsa jest dostatecznie często aktywowana (sygnały przesyłane między neuronami mają wartości binarne), to występuje na niej sumowanie czasowe. Może się zdarzyć, że dostatecznie częste drażnienie jednej synapsy wywoła odpalenie neuronu. Rozładowywanie się ładunków postsynaptycznych może przebiegać również w trybie natychmiastowym. Sytuacja ta występuje w momencie aktywacji neuronu postsynaptycznego. Po odpaleniu się komórka nie posiada już wcześniej

gromadzonych ładunków elektrycznych. Proces gromadzenia ładunku zaczyna się od początku.

W modelu występują dwa rodzaje synaps: pobudzająca oraz hamująca. Ta druga jest typu hamowania presynaptycznego, tzn. w tej synapsie znajdują się elementy (dwa aksony i dendryt) należące do trzech różnych neuronów. Działanie takiej synapsy polega na tym, iż pojawienie się potencjału na aksonie hamującym wpływa blokująco na przepływ impulsu na drodze akson pobudzający - dendryt. Przyjmuje się, że blokowanie jest wprost proporcjonalne do potencjału na aksonie blokującym.

Praca neuronu polega na sumowaniu arytmetycznym wszystkich potencjałów na synapsach po uwzględnieniu hamowania oraz porównania tej wartości z sumą progu oraz wartości refrakcji. Jeśli neuron się odpali, to wysyła impuls zerując jednocześnie potencjały na synapsach. Wysłany impuls powoduje powstanie nowych potencjałów postsynaptycznych w innych neuronach o wartości proporcjonalnej do odpowiednich wag.

Algorytm uczenia można podzielić na 3 części:

1) wykonywany w każdym takcie symulacji
2) wykonywany podczas odpalenia neuronu
3) wykonywany podczas nieaktywności neuronu.

Punkt pierwszy dotyczy uczenia synaps hamujących. Jeśli zostało wykryte hamowanie presynaptyczne niezależnie od stanu neuronu, to siła hamowania zostaje wzmacniana.

$$\text{jeśli } x_{i,t}>0 \ \wedge x_{i+1,t}>0 \ \wedge w_{i,t}<0 \ \Rightarrow w_{i,t+1}=w_{i,t}-a \wedge w_{i+1,t+1}=w_{i+1,t}+a \qquad (34)$$

gdzie:

a - stała określająca prędkość uczenia. (np. 5)

$x_{i,t}$ - wartość potencjału na synapsie nr i w chwili t (0-50mV)

$w_{i,t}$ - wartość wagi synapsy nr i w chwili t <-128,127>

W punkcie drugim zawarty jest proces dążenia wartości wag do 0 w przypadku nieaktywnych synaps, zarówno pobudzających, jak i hamujących. Próg dąży

asymptotycznie, rosnąco do określonego ułamka (właściwego) aktywacji. Zwiększane są wagi aktywnych synaps pobudzających i hamujących.

$$\text{jeśli} \quad \left\{ \begin{array}{l} x_{i,t}=0 \Rightarrow w_{i,t+1}=w_{i,t}-b*sign(w_{i,t}) \\ x_{i,t}>0 \Rightarrow \left\{ \begin{array}{l} w_{i,t+1}=w_{i,t}+a*sign(w_{i,t}) \\ h_{t+1}=h_t+(c*\sum_i w_{i,t}*x_{i,t}-h_t)/d \end{array} \right. \end{array} \right\} \qquad (35)$$

gdzie:

b - stała określająca prędkość zmniejszania wag, zwykle o rząd wielkości mniejsza od stałej *a*

c - stała, najlepiej ułamek właściwy ok. 0.66

d - stała, np.3

h_t – wartość progu w chwili *t*

sign() - funkcja signum

Punkt trzeci służy jedynie do modyfikacji progu w kierunku malejącym.

$$h_{t+1}=h_t-f \qquad (36)$$

gdzie:

f - stała, np. 0,01

Bardzo ważną rolę w algorytmie uczenia pełnią proporcje zmian wag, wywoływane różnymi przyczynami np. modyfikacja progu w tym punkcie ma dwa rzędy mniejsze znaczenie, niż wykrywanie hamowania lub wzmacnianie synaps aktywnych podczas odpalenia.

Porównując modele IMPL2 i IMPL7 można powiedzieć, że IMPL2 wykształcał zbyt wąską klasę rozpoznawanych wzorców. Właściwości generalizacji były dosyć ubogie. Zostało to poprawione przez wprowadzenie dodatkowych założeń w modelu IMPL7. Zachowany został typ synergii neuronów obserwowany w IMPL2, lecz zmieniona została skala przetwarzanych danych. Podstawową prezentacją danej wejściowej nie jest pojedyncze pobudzenie synapsy, lecz cała seria pobudzeń zgromadzonych na przestrzeni określonego odcinka czasu. Jest to jednocześnie krok wykonany w kierunku zbliżenia się do zjawisk znanych z badań fizjologicznych. Sieć o takiej strukturze posiada wysokie zdolności do generalizacji. Przypisuje ona różną ważność dla różnych elementów wzorców wejściowych tzn. klasyfikuje cechy

rozpoznawanych wzorców na pierwszorzędne, drugorzędne itd., Własność ta jest możliwa do osiągnięcia dzięki gromadzeniu się w neuronach statystyk na temat ciągów wejściowych odzwierciedlonych w wagach. Przetwarzanie wykazuje cechy wysokiej równoległości. Pod względem niezawodności też wystąpiła poprawa w stosunku do IMPL2. Zniszczenie losowo wybranych neuronów nie wpływa tak drastycznie na pracę, jak miało to miejsce w modelu IMPL2. Okupione to zostało zwiększeniem ilości neuronów w pracującej sieci niezbędnych do przetworzenia podobnej ilości informacji oraz wyłączeniem częściowym sprzężeń zwrotnych, gdyż zmieniona została skala danych wejściowych. Zwiększenie złożoności sieci modelu IMPL7 powoduje jednocześnie okresowe zachowania z pogranicza chaosu deterministycznego, co można w naturalny sposób wykorzystać do wychodzenia z minimów lokalnych błędu rozwiązania.

Model IMPL7 został sprawdzony również pod kątem możliwości zbudowania z tak zdefiniowanych neuronów układów znanych z fizjologii jak: **stabilizator częstotliwości, komparator częstotliwości, generator wzbudzany pobudzeniem**. Doświadczenia wypadły pozytywnie!

Omawiany model jest pewną próbą zebrania pozytywnych cech IMPL5 i 6, oraz ich uogólnienia. Pamięć krótko- i długotrwała ma właściwości zbliżone do IMPL6. Żadnych istotnie nowych zjawisk nie zaobserwowano.

5.3 Efekty synergii w sieciach o dużej złożoności

Omówione wyżej doświadczenia dotyczyły dość ograniczonej liczby neuronów, zwykle kilkunastu (nigdy nie więcej niż sto), co było ograniczeniem szybkiego symulatora IMPL. Przy kilkudziesięciu neuronach - obserwacje zachowania sieci w tym symulatorze jest bardzo żmudne i mało efektywne. Biorąc pod uwagę, iż niektóre efekty mogą być zauważone przy określonej pojemności pamięci krótkoterminowej (np. kojarzenie przesunięć czasowych sygnału), został zaprojektowany, wykonany i przetestowany mocniejszy symulator o nazwie SNN. Umożliwia on użycie całych zasobów PC, zarówno pamięci jak i procesora, ponieważ jest on mocno skalowalny. Na prostym PC 166 MHz i 64MBRAM przy użyciu SNN jest możliwa symulacja nawet do 10 000 neuronów z włączoną symulacją ładunków. Jest też bardziej pomocny przy obserwacji oraz analizie tak dużej sieci. Dla porównania symulator LSM [141], będący modułem napisanym w C++ do Matlaba w 2003 roku symuluje 1000 neuronów, ale tylko 100 tys. synaps. SNN już przy 500 neuronach posiada ponad 2,5 raza więcej synaps.

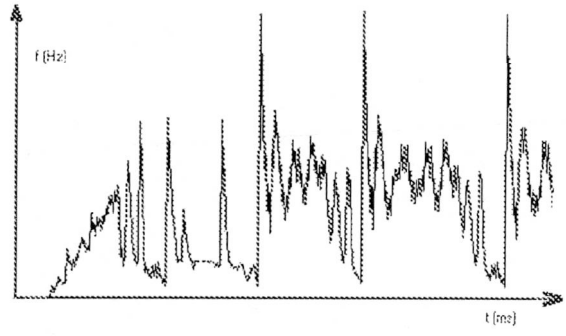
Oceńmy maksymalną pojemność pamięci krótkoterminowej sieci o wielkości *n* neuronów. W skrajnym przypadku każdemu ładunkowi postsynaptycznemu możemy przypisać 1 bit. Ponieważ sieć posiada strukturę połączeń „każdy z każdym", ilość synaps wynosi V=n*n. Dla n=400, V=160000 bitów, co daje 20000 bajtów. Jest to pojemność bardzo duża, lecz praktycznie nie do wykorzystania przez sieć w obecnych modelach, ze względu na trudność sterowania sprzężeniami zwrotnymi bezpośrednimi oraz pośrednimi. Po uwzględnieniu tego faktu oraz wyeliminowaniu autoasocjacyjności neuronu otrzymujemy V=(n-1)*n/2. (dla n=400, V=9975 bajtów). W tym momencie oceniliśmy maksymalną pojemność pamięci krótkotrwałej, przy założeniu że każdy potencjał postsynaptyczny reprezentuje jeden bit. Doświadczenia wykazują, że efektywnie z tego jest używane jedynie 10% objętości, ponieważ potencjały postsynaptyczne są używane do generalizacji rozwiązań, czyli kilka potencjałów reprezentuje jeden element (bit) przetwarzanego wzorca. Drugim faktem jest duża nadmiarowość sieci w większości modeli, szczególnie w ściśle

określonych obszarach matrycy wagowej, co zostanie omówione na przykładzie modelu SNN.

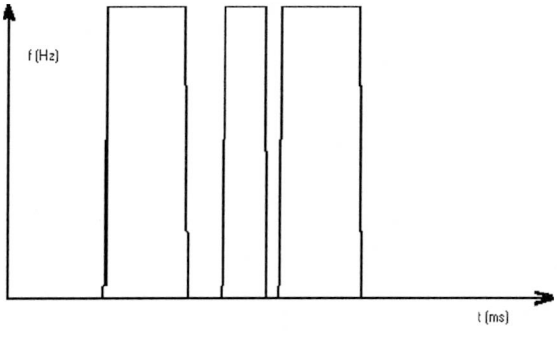
Wymieniony wyżej ułamek użytecznej pamięci ma jednak inne znaczenie niż można się spodziewać. Sieć nie ma konieczności pamiętania przetwarzanych danych ze stuprocentową dokładnością. Właściwie świadczyłoby to o braku przetwarzania wzorca. W pamięci krótkoterminowej znajduje się zatem przetworzona reprezentacja sygnału zgodnie z pamięcią długoterminową powstała z chwilowych danych wejściowych.

5.3.1 Modele SNN10 – podstawowe zależności w sieciach o dużej liczbie neuronów

Przenosząc modele sieci oraz neuronów z poprzednich doświadczeń do bardziej złożonego środowiska napotykamy na kilka trudności, związanych z samą strukturą sieci. Znaczny przyrost ilości synaps zwiększa wariancję aktywacji, co za tym idzie, funkcja zmiany progu musi uwzględniać ten fakt. Model SNN10a został przeniesiony z IMPL 7 i wykazywał słabą aktywność na dane wejściowe. Na rys. 40 można zauważyć, iż odpowiedź sieci po wyuczeniu wzorca nie jest wyraźna, mimo iż reaguje silnie na niektóre elementy wzorca (tutaj: jego początek). Objawia się to względnie dużą ilością odpaleń neuronów. Na wykresie częstotliwości widzimy trzy takie momenty.



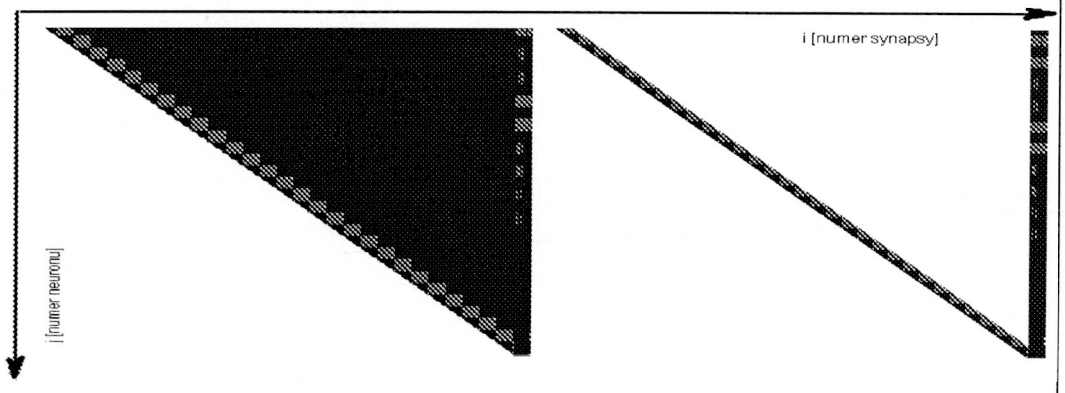
Rys. 40 – Reakcja SNN10a na powtarzające się dane wejściowe.



Rys. 41 – Model SNN10c wykazuje się niezwykle silną polaryzacją. Kategoryczność odpowiedzi tego modelu jest okupiona małymi zdolnościami do generalizacji.

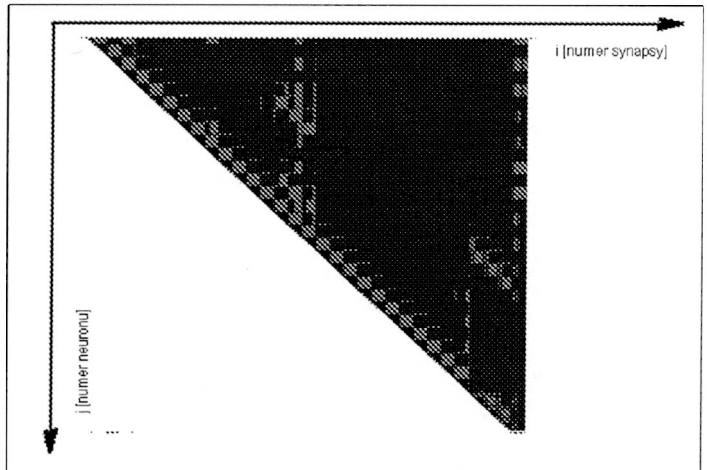
Po wstępnej analizie funkcję modyfikacji progu ustalono na bardzo szybko dążącą do aktywacji. Potencjały postsynaptyczne zostały usunięte z algorytmu uczenia, pozostały jedynie w procesie pracy. Model (SNN10b) po wyuczeniu wzorca wykazywał wysoką średnią częstotliwość odpaleń neuronów, a na dane wejściowe reagował zmniejszeniem tej częstotliwości.

Odpowiednikiem IMPL 2 w obecnym symulatorze jest SNN10c. Wyłączono w nim całkowicie symulacje potencjałów postsynaptycznych oraz



Rys. 42 – Porównanie matryc wag modelu SNN10c (po prawej) oraz SNN10d (po lewej). Można zauważyć, iż obszar wewnątrz matrycy SNN10d nie zakodował żadnych danych. Dodatkowo zauważamy większą integrację danych w SNN10c.

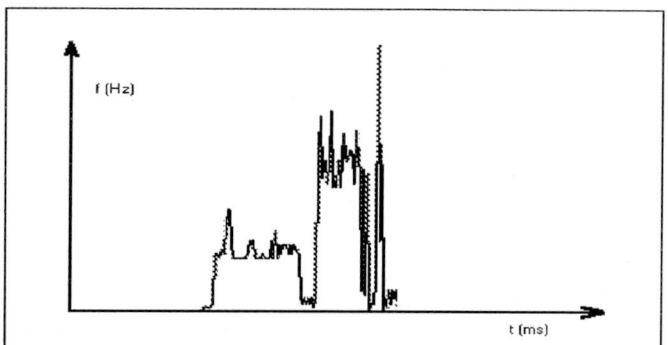
wyeliminowano wagi ujemne. Model podobnie jak IMPL2 nie ma zdolności do generalizacji, jednak łatwiej synchronizuje się z sekwencją wejściową, co z pewnością jest wynikiem braku sprzężeń zwrotnych. Neuron tego modelu komunikuje się ze swoim sąsiedztwem o określonej wielkości oraz z globalnym wejściem sieci. Sieć jest w stanie przetwarzać znacznie dłuższe sekwencje wejściowe oraz czas uczenia jest znacznie krótszy. Model ten będzie wyjściowym do kolejnych modyfikacji.



Rys. 43 – SNN10e – wnętrze matrycy wag nie jest puste.

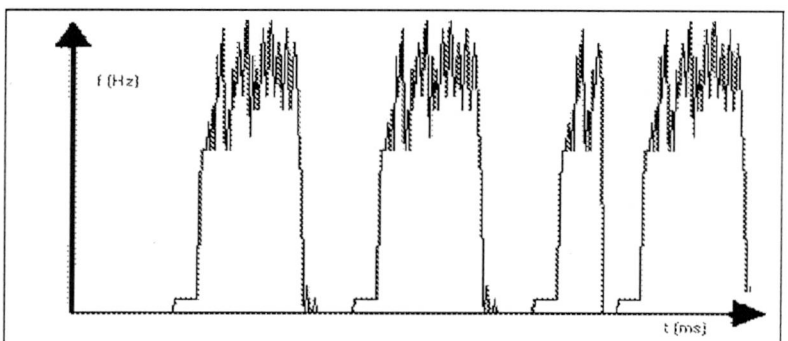
Przy obecnych założeniach rozszerzenie sąsiedztwa neuronu do wielkości maksymalnej nie przynosi żadnych zmian (SNN10d). Duże zmiany powoduje włączenie symulacji potencjałów postsynaptycznych (SNN10e). W tym modelu pobudzenie nie przemieszcza się już swobodnie, a synchronizacja bywa gubiona. Dzieje się tak, iż napływające dane globalne są w stanie w dowolnym momencie zinterferować się z wcześniejszymi danymi. Oznacza to

niedopasowanie algorytmu uczenia. Zmniejszenie w tym momencie sąsiedztwa nie rozwiązuje problemu, gdyż co prawda sygnał wejściowy nie jest w stanie zinterferować się z poprzednimi danymi (SNN10f), ale poprzez sieć przebiegają samowyindukowane impulsy nie związane z danymi wejściowymi, które wprowadzają algorytm uczenia na złe ścieżki (minima lokalne).



Rys. 44 – Odpowiedź SNN10f

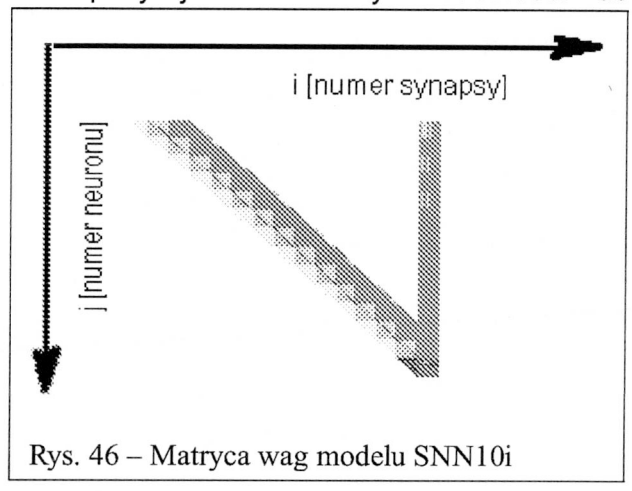
Pomocniczą w tym momencie okazuje stała charakteryzująca wielkość modyfikacji wagi w procesie uczenia. Jej znaczne zmniejszenie powoduje pewne zestrojenie systemu i sieć jest w stanie odwzorowywać sekwencję wejściową zarówno z, jak i bez włączonych potencjałów postsynaptycznych (SNN10g). Rys. 45 obrazuje test przeprowadzony na tym modelu, poddający sieć próbie wprowadzenia nieprawidłowych danych pomiędzy prawidłowymi wzorcami. Jak można zaobserwować na wykresie częstotliwości, widać wyraźną reakcję odrzucenia danych podczas przetwarzania trzeciej serii wektorów. Widoczny spadek częstotliwości do zera oznacza jednocześnie zerwanie synchronizacji.



Rys. 45 – SNN10g – odpowiedź częstotliwościowa sieci na 3 wzorce i jeden zakłócony

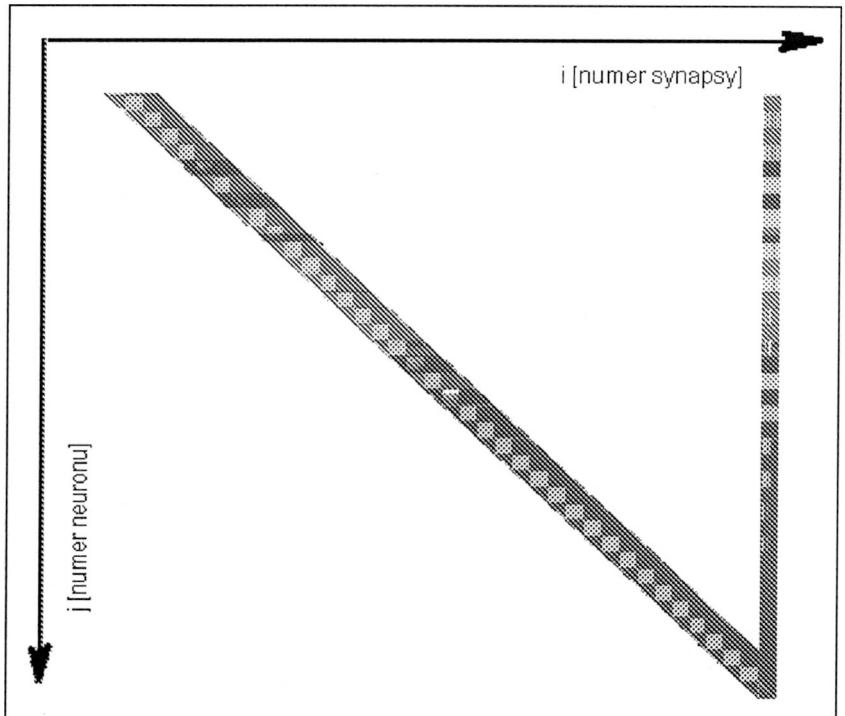
Aby bardziej kategorycznie odróżnić wzorce od podobnych im sekwencji wejściowych w modelu SNN10h wyłączono refrakcję, mechanizm uczenia neuronu skupiono na jego pierwszym odpaleniu, utrzymano dość skromne sąsiedztwo neuronu oraz symulację ładunków postsynaptycznych. Efekt

zamierzony został osiągnięty. Ucierpiały jednak na tym zdolności do generalizacji.



Rys. 46 – Matryca wag modelu SNN10i

Aby temu zapobiec, należy powrotem przenieść ciężar uczenia na wszystkie odpalenia neuronu, a nie tylko pierwsze (SNN10i). Jak obserwujemy na wykresie średniej częstotliwości odpaleń, wzorzec jest zapisywany w matrycy wag, jednak początek sieci jest atakowany stosunkowo dużą liczbą danych i po określonym czasie się



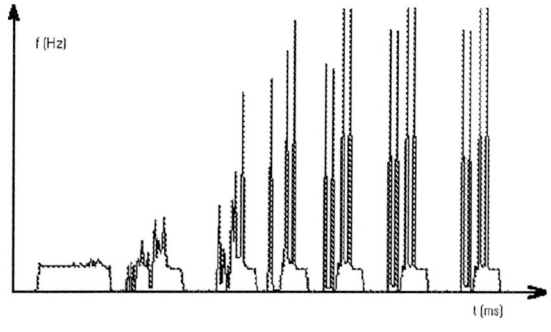
Rys. 47 – Matryca wag modelu SNN10j

rozprogramowuje (rys. 46). Zacierana jest zatem informacja o początku wzorca, a ponieważ sieć działa bardzo kategorycznie, wzorzec nie jest rozpoznawany ponownie. Cała sieć czeka na wystartowanie neuronów początkowych, podczas gdy one są w tym momencie przeprogramowane.

Skonstruowanie modelu pośredniego pomiędzy SNN10h i SNN10i nie jest zadowalające. Co prawda sieć potrafi rozpoznać wzorzec ponownie, zatem ma możliwości rekonstrukcji wzorca, jednak procesy rozprogramowujące

początek sieci są tylko znacznie spowolnione. Co za tym idzie, po dłuższej pracy nie działa ona prawidłowo (SNN10j). Na rys. 47 widzimy wyraźnie ciemniejsze miejsca na przekątnej o względnie dużych powierzchniach. Oznacza to, iż w tych miejscach nawet przy przetwarzaniu poprawnego wzorca synchronizacja może zostać zerwana, ponieważ ciemniejsze miejsca wskazują na wagi o niższych wartościach.

Aby zmniejszyć efekt odrzucania wektorów nieznacznie odbiegających od wzorca została zmieniona funkcja modyfikacji progu, z dążącej do aktywacji na dążącą do 4/5 z aktywacji (SNN10k), jednak dopiero wyłączenie potencjałów postsynaptycznych (SNN10l) przywróciło stabilność zapamiętania wzorca.



Rys. 48 – Procesy rozprogramowujące SNN10j

Posiadając wymienione możliwości operowania na danych wejściowych przy pomocy modelu sieci neuronowej należy się zastanowić nad parametryzacją niektórych cech przetwarzania jak stopień generalizacji rozwiązań, zależność obecnie przetwarzanych danych od wcześniejszego kontekstu, stopień integracji danych przypadających na synapsę / neuron pochodzącej z wzorca wejściowego.

Pierwszy parametr można scharakteryzować w najprostszym przypadku poprzez procent zgodnych elementów wektora z wektorem wzorcowym. W ogólności, każdemu elementowi wektora jest przypisana wartość oznaczająca istotność danego bitu danych i te wartości są używane do wyliczenia zgodności ze wzorcem.

Kontekst w wielu implementacjach jest traktowany jako dodatkowy wektor wejściowy, wyliczany poprzez sieć w poprzedniej iteracji i podawany w kolejnej do określonych wejść sieci. Przy tej strukturze sieci, można powiedzieć, że nie ma takiego ścisłego podziału, ponieważ zabronione są sprzężenia zwrotne, zatem każdy neuron ma dostęp tylko do wyników obliczeń neuronów o mniejszych indeksach. Cechą, którą w tym punkcie chcemy operować, jest istotność kontekstu przy przetwarzaniu kolejnego wektora

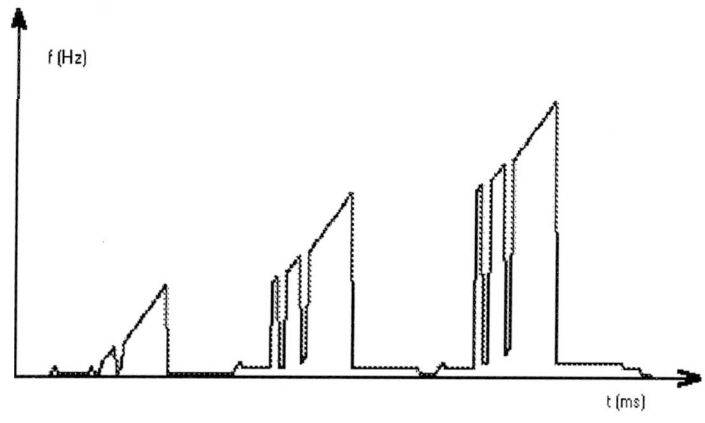
wejściowego. W skrajnych wypadkach pożądane jest: przy ignorowaniu kontekstu – uznanie danych za zgodnych ze wzorcem, gdy x ostatnich wektorów wejściowych jest zgodne z x ostatnimi wektorami wzorca; przy priorytetowaniu kontekstu – uznanie danych za niezgodnych ze wzorcem, jeśli pierwszy wektor danych i wzorca nie był identyczny.

Trzecią cechą sieci, na która powinna być możliwa do ustalenia, to średnia ilość neuronów kodująca jeden wektor wejściowy. Jak pokazują doświadczenia, ma to bezpośredni związek z pracą sieci i jej zdolnościami do uogólniania rozwiązania. W skrajnym przypadku jeden wektor wejściowy powinien być kodowany na jednym neuronie. Idąc w przeciwnym kierunku, jednemu wektorowi wejściowemu powinno odpowiadać x neuronów, x>1.

Pierwsza próba sparametryzowania ww. cech oraz przekształcenia ich na parametry budowy sieci zostały przeprowadzone w modelu SNN10m. Jak się okazuje, zmiana wielkości sąsiedztwa danego neuronu ma bezpośredni wpływ na drugą i trzecią cechę, a pośrednio również i na pierwszą (głównie poprzez trzecią). Model ten przetestowano przy bardzo małym parametrze sąsiedztwa począwszy od wartości Z=2, oznaczającej możliwość połączenia się z sobą samym oraz swoim poprzednikiem. Ponieważ autoasocjacyjność jest też zabroniona, neuron posiada możliwości przetwarzania jedynie kontekstu pochodzącego od jego poprzednika, oraz oczywiście ma dostęp do aktualnie przetwarzanego wzorca wejściowego. Warunek taki znacznie zmniejsza prędkość przepływu informacji przez sieć, a biorąc pod uwagę, iż dane wejściowe napływają ze stałą prędkością, stopień ich koncentracji w sieci musi być większy i tak faktycznie jest.

Kontekst w tym modelu też zachowuje się zupełnie inaczej niż w poprzednim. Należy pamiętać, sygnały międzyneuronowe mają postać binarną, a cały kontekst jest przekazywany z jednego neuronu do następującego po nim w postaci jednego takiego sygnału. Oznacza to, iż może on zawierać informację na temat zgodności poprzednich wektorów danych ze wzorcem na wybranym poziomie zgodności lub nie i jest to w tym przypadku informacja binarna, bez możliwości przekazania parametru o stopniu zgodności itp.

Stopień koncentracji informacji możemy ocenić następująco: jak zostało założone w modelu, dane są podawane do wejść w kolejnych punktach czasowych, odległych od siebie czasem refrakcji pojedynczego neuronu (większe częstotliwości po prostu fizycznie poprzez neuron nie mogą być przetwarzane) przyjętym jako 2 ms. W tym samym czasie informacja w sieci o zgodności wektora z wzorcem jest w stanie przejść drogę tylko do jednego (kolejnego) neuronu. Oznacza to, iż jeden wektor musi być kodowany na jednym neuronie. Po przeglądnięciu poszczególnych neuronów oraz porównaniu wag synaps odpowiedzialnych za połączenia z wejściem sieci z wzorcem, widać wyraźnie odbicie wyuczonego wzorca na tych synapsach.



Rys. 49 – SNN10m – odpowiedź sieci przedstawiona na wykresie średniej częstotliwości odpaleń neuronów świadczy o możliwości startowania impulsu synchronizacji z pominięciem kontekstu lub nawet początku wzorca.

Jak widać na rys. 49 dane wejściowe są w stanie wygenerować impuls oznaczający zwykle zgodność poprzednich wektorów ze wzorcem, nawet gdy taka zależność nie wystąpiła. Dzieje się tak szczególnie wtedy, kiedy przez dłuższy czas wejście sieci jest nieaktywne i wszystkie tego typu impulsy zanikną. Zjawisko to jest pożądane, zgodnie z wymienionymi wyżej założeniami i świadczy o możliwości ustalania wpływu kontekstu na proces przetwarzania. Tym przypadku jest on krańcowo mały. Przy założeniach przyjętych w SNN10m kontekst jest reprezentowany jako jeden bit. Oczywiście waga synapsy w kolejnym neuronie odpowiedzialna za utrzymanie kontekstu może być duża, ale statystyczne ilość bitów przenoszących kontekst w pierwszych krokach uczenia przesądza o priorytecie tej informacji. Dokładniej analizując pracę sieci, przekłada się to na stosunek ilości bitów kontekstu do

ilości wejść sieci (co jest równoważne z ilością bitów wejścia sieci danego momentu w czasie). W badanym modelu stosunek ten wynosi 1/12. Pamiętając, iż próg podczas odpalenia neuronu w przyjętym modelu modyfikuje się zgodnie ze wzorem:

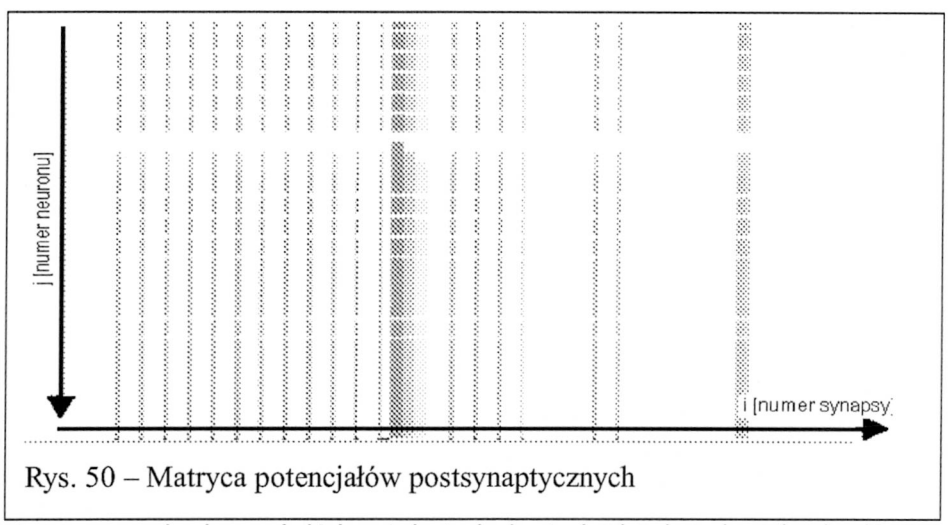
$$h_{j,t+1} = \sum_i w_{i,j,t} * x_{i,j,t} * \frac{4}{5} \qquad (37)$$

gdzie:

$h_{j,t}$ - wartość progu neuronu *j* w chwili *t*

$w_{i,j,t}$ - wartość *i*'tej wagi neuronu *j* w chwili *t*

$x_{i,j,t}$ - wartość i'tego wejścia neuronu *j* w chwili *t*

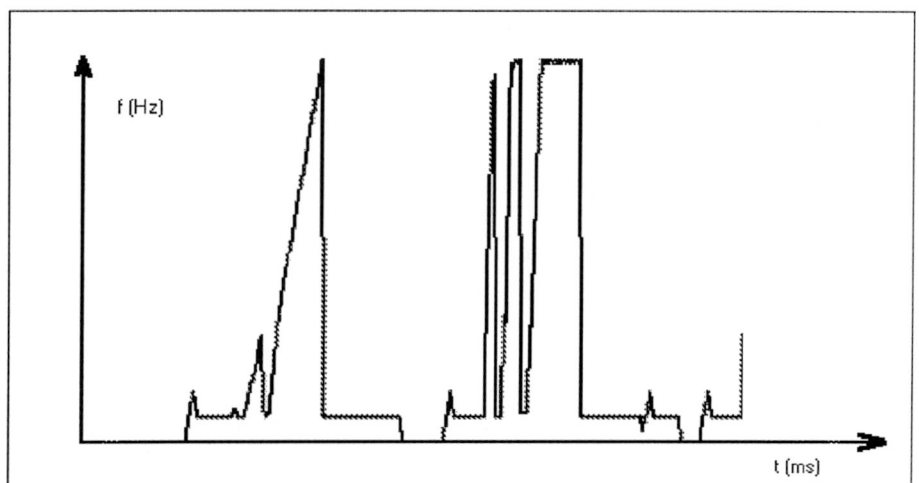


Rys. 50 – Matryca potencjałów postsynaptycznych

możemy wywnioskować, iż kontekst nie jest niezbędny do odpalenia neuronu, a co się za tym kryje, do odpalenia impulsu oznaczającego zgodny kontekst. W dalszej części pracy, że względu na specyfikę przemieszczania się tego impulsu pobudzającego kolejne neurony w kolejnych chwilach czasowych, będzie on nazywany impulsem synchronizującym. Dodatkowo przy rozbudowanym kontekście (kilku lub kilkunastobitowym), który podlega pod te same prawa co impuls synchronizujący, zbiór propagujących się aktywacji neuronów tworzy falę synchronizującą. Propagujące się impulsy kontekstu są dobrze obserwowalne na matrycy potencjałów postsynaptycznych, ponieważ powodują efekt przepływu przez sieć „fali" potencjałów postsynaptycznych - rys. 50.

Opisany efekt widziany na matrycy potencjałów postsynaptycznych ma bardzo duże znaczenie podczas przeprowadzania doświadczeń. Podczas procesu uczenia nie od razu mamy do czynienia z przepływem takiej fali. Organizacja neuronów trwa czasami dość długo oraz następują różne przegrupowania, co za tym idzie sygnały propagują się np. tylko przez krótki odcinek sieci, albo gasną w różnych nieokreślonych momentach, lub sygnały powstają w różnych miejscach sieci i odpalenia następują asynchronicznie, nie tworząc zwartej grupy i tym bardziej propagującej się fali. Na tej podstawie po określonej liczbie kroków symulacji jest podejmowana decyzja, czy model wykazuje efekty synergii neuronów i czy doświadczenie warto kontynuować.

W modelu SNN10n sąsiedztwo zostało zwiększone do Z=7, co jest również dość małą wartością. Prędkość propagacji informacji w sieci zostaje

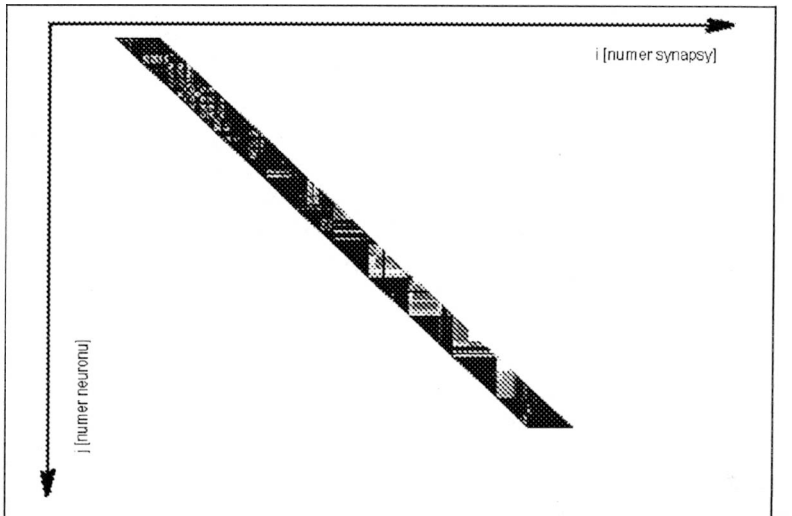


Rys. 51 – Wykres częstotliwości neuronów modelu SNN10n

automatycznie zwiększona, stopień integracji informacji na element sieci jest zmniejszony. Na wykresie średniej częstotliwości odpaleń można zauważyć, że wielkości te nie rosną tak liniowo jak w poprzednim modelu, ale następują gwałtowne spadki częstotliwości. Oznacza to zwiększenie priorytetu kontekstu. Dokładniej analizując, wzorzec wejściowy posiadał element okresowy wewnątrz swojej struktury. Jest to dość trudne zadanie dla sieci neuronowej, ponieważ dość łatwa jest pomyłka, w którym momencie przetwarzania wzorca aktualnie sieć się znajduje. Ponieważ zgodność jest dość duża, wykres częstotliwości wskazuje ich wzrost. Jednakże w odpowiednim momencie (po zakończeniu wewnętrznej okresowości wektorów wzorca) częstotliwość spada do wartości, która jeszcze oznacza zgodność ze wzorcem (rys. 51). Gdyby zgodność

danych ze wzorcem była nieprawdziwa, częstotliwość odpaleń spadałby do częstotliwości bliskiej zero.

Wprowadzona w modelu SNN10j cecha, polegająca na kilkukrotnie silniejszym uczeniu neuronu przy jego pierwszym odpaleniu pozwoliła uzyskać wiele zamierzonych efektów oraz przyśpieszyć wiele doświadczeń, jednakże ogranicza właściwości generalizacji metawzorca, ze względu na zbyt silne wyuczenie pierwszych danych, (pierwszego przedstawienia wzorca, co nie musi być reprezentatywne). Wyłączenie tej cechy sieci nie jest procesem banalnym, jakby wyglądało to pozornie, ponieważ może zostać zaburzony porządek synchronizacji np. przy drugiej prezentacji przedstawiciela wzorca. Cykliczne wzajemne zaburzanie synchronizacji poprzez dwa po części sprzeczne przedstawienia wzorca zwykle nie prowadzą do uogólnienia rozwiązania. Oczywiście pozostaje do rozstrzygnięcia problem, czy aż tak różne dwa przedstawienia wzorca można uznać jako przedstawienia wzorca? Z założenia, wektory będące przedstawieniami wzorca powinny być podobne. W modelach od SNN10j pierwszy pokazany wzorzec był zdecydowanie wiodącym, kolejne dostosowywały się do pierwszego. W modelu SNN10o zlikwidowano tę zasadę, a dodatkowo jako podstawowy mechanizm generalizacji- włączono symulację ładunków postsynaptycznych, refrakcję oraz zwiększono sąsiedztwo neuronu. Nie udało się wykryć w tym modelu efektów synergii neuronów, nawet przy wyłączeniu sterowania wartością progu. W modelu SNN10p pracą progu oraz prędkościami jego zmian dokonano pewnego strojenia systemu, jednak dalej nie można stwierdzić jego jakiejkolwiek pracy jako całości. Ponieważ wszystkie



Rys. 52 – Efekty synergii neuronów widziane na matrycy wag modelu SNN10r

wprowadzone elementy do modelu są pożądane, ale nie występuje jakakolwiek synergia neuronów, pierwszym elementem do strojenia systemu musi być algorytm zarządzania pracą progu neuronu. Ponieważ statyczne funkcje zostały w dużej mierze przebadane i nie przyniosły rezultatu, kolejnym krokiem może być uzależnienie zmian progu od innych parametrów neuronu. Biorąc pod uwagę fakt, iż próg jest elementem pamięci długoterminowej, jego modyfikacje muszą być odwracalne. Oznacza to, iż chwilowa zmiana progu musi zostać po dłuższym czasie anulowana. Zatem planowane zmiany progu muszą mieć tę samą naturę co refrakcja, tylko odbywać się zarówno w stronę wzrostu jak i obniżania wartości progu.

Niepożądaną cechą SNN10p jest okresowy brak pobudzeń występujący po wzbudzeniu sieci (i dopasowaniu się neuronów do pracy podczas wzbudzenia). Zgodnie z faktami neurofizjologicznymi, każdy neuron musi się odpalać chociażby z określoną minimalną częstotliwością, nie mogą natomiast następować dłuższe okresy braku aktywacji. Biorąc pod uwagę, iż możemy zmodyfikować funkcję refrakcji oraz znamy kierunek jej zmian (ma prowadzić do utrzymania minimalnej częstotliwości odpaleń neuronu), oznacza to iż funkcja refrakcji w określonym czasie *dt* dąży do przeciwnej wartości progu. Załóżmy wstępnie liniowość tego procesu. Dodatkowo zmniejszmy tempo wzrostu progu w chwilach odpaleń zgodnie ze wzorem:

$$h_{j,t+1} = h_{j,t} + \frac{\frac{4}{5}\sum_i w_{i,j,t} * x_{i,j,t} - h_{j,t}}{3} \qquad (38)$$

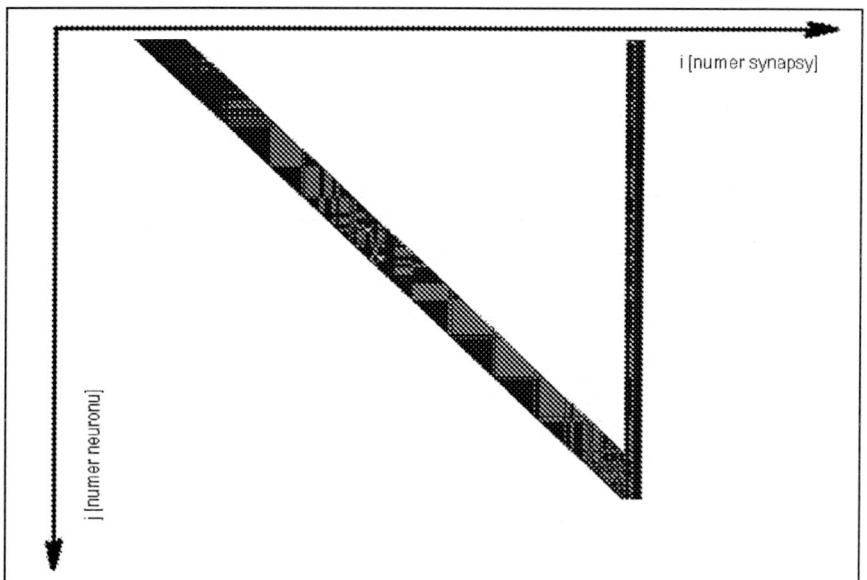
gdzie:

$h_{j,t}$ - wartość progu neuronu *j* w chwili *t*

$w_{i,j,t}$ - wartość *i*'tej wagi neuronu *j* w chwili *t*

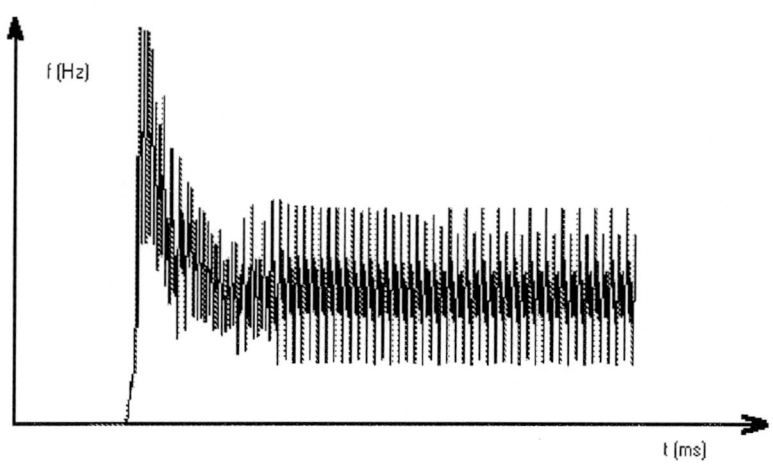
$x_{i,j,t}$ - wartość i'tego wejścia neuronu *j* w chwili *t*

Po doświadczalnie dobranym *dt* w modelu SNN10r można zaobserwować pewną samoorganizowalność neuronów (rys. 52) ale pojawił się nowy jakościowo problem. Otóż zgodnie z założonym celem, neurony powinny



Rys. 53 – Niezerowe wartości wag wejściowych systemu w modelu SNN10t są potwierdzeniem hipotezy

utrzymywać określoną minimalną częstotliwość. Jeśli natomiast dane wejściowe sieci statystycznie będą miały mały udział podczas aktywacji neuronów (np. gdy częstotliwość pobudzeń z wejścia systemu będzie mniejsza niż omawiana minimalna częstotliwość pracy neuronu) od może dojść do odłączenia przez sieć sygnałów wejściowych, co obserwujemy w SNN10r.



Rys. 54 – SNN10s na bardzo krótką sekwencję wejściową odpowiada częstotliwością o równie krótkim okresie – model pracuje prawidłowo

Dla potwierdzenia hipotezy o zależnościach częstotliwościowych fm oraz fd w modelu SNN10s zostały przygotowane i podane do sieci sygnały wejściowe o stosunkowo dużym średnim pobudzeniu. Niektóre wejścia zostały z dodatkową siłą nasycone pobudzeniami (SNN10t – rys. 52). Jak można zauważyć na wykresy średniej częstotliwości odpaleń neuronów sieci, okres wyrównał się z okresem częstotliwości prezentacji wzorca, rys 53, co oznacza, że neurony podzieliły się na liczbę klas odpowiednią wzorcowi. Wystąpiły zatem efekty synergii, czyli hipoteza się potwierdziła i zidentyfikowaliśmy problem do rozwiązania.

Podsumowując serię modeli SNN10 obserwujemy w nich wiele ciekawych procesów, niemożliwych do zauważenia przy mniejszej skali symulowanej sieci. Odkrycie zależności panujących w takiej strukturze mówi wiele na temat przetwarzania informacji w takiej strukturze oraz pozwala sterować niektórymi jej parametrami np. SNN10m. Jak pokazują doświadczenia, strojeniu wymaga duża liczba parametrów, chociaż większość procesu strojenia udaje się zautomatyzować. Generalnie modele SNN10 charakteryzują się dość małą wrażliwością na niektóre różnice w sygnale wejściowym, co jest jednym z ważniejszych wątków badań kolejnej serii SNN11.

5.3.2 Modele SNN11 – próba znalezienia optymalnego algorytmu uczenia

Modele SNN11 bazują na modelu wyjściowym posiadającym następujące cechy: sąsiedztwo neuronu jest zdefiniowane jako trzykrotna wartość wejść do systemu, symulacja ładunków postsynaptycznych jest włączona, nie ma priorytetu uczenia w pierwszych odpaleniach neuronu, próg rośnie dość wolno do ułamka z aktywacji zgodnie ze wzorem

$$h_{j,t+1} = h_{j,t} + \frac{\frac{4}{5}\sum_{i} w_{i,j,t} * x_{i,j,t} - h_{j,t}}{3} \qquad (39)$$

gdzie:

$h_{j,t}$ - wartość progu neuronu j w chwili t

$w_{i,j,t}$ - wartość i'tej wagi neuronu j w chwili t

$x_{i,j,t}$ - wartość i'tego wejścia neuronu j w chwili t

Sieć jest traktowana danymi o dużej koncentracji sygnałów.

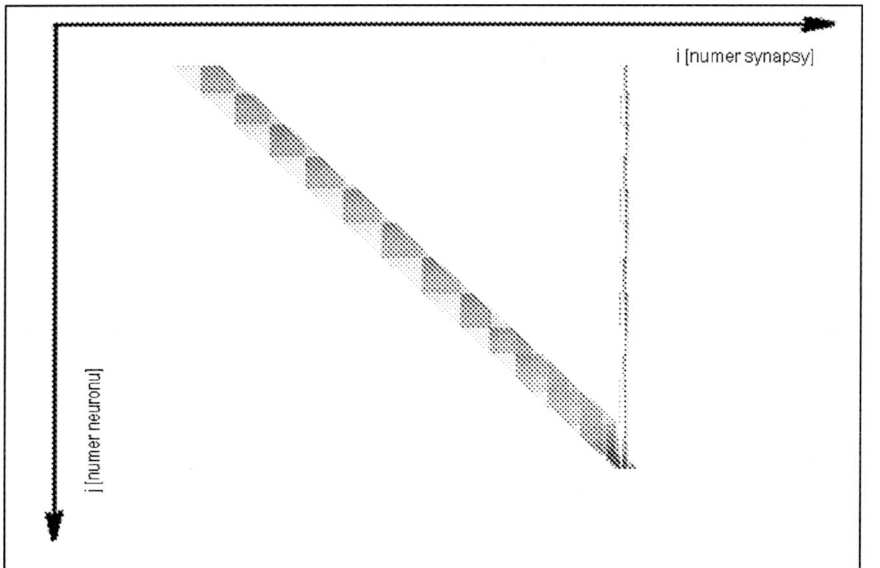
Jak zostało powiedziane we wnioskach poprzedniego paragrafu, modele SNN10 powinny zostać wyposażone w mechanizm, który bardziej szczegółowo analizowałby cechy wzorca. W modelach SNN11 ciężar prac zostanie skupiony na algorytmie uczenia.

Podczas transformacji modeli przystosowującej je do symulowania ładunków postsynaptycznych zarówno wyliczanie aktywacji jak i modyfikacji wag zostało oparte na ww. wartościach ładunków. O ile podczas wyliczania aktywacji wartość ładunku była w dość oczywisty sposób użyta (wzór), o tyle podczas poprawki wag nadal była traktowana binarnie czyli inaczej gdy na synapsie nie było żadnego ładunku i inaczej gdy był chociaż minimalny ładunek. W modelach SNN okres utrzymywania się ładunku na synapsie wynosi ok. 14 ms, zanim użyta rozdzielczość liczb wykaże wartość 0. Prowadziło to przy przyjętym kroku równym 2 ms do powstania ok. 7 razy większej ilości aktywacji synaps w sieci, co bez wcześniejszego strojenia algorytmu uczenia prowadzi do szybkiego wzrostu wag oraz ogólnego wzbudzenia. Pierwszą próbą zminimalizowania tego efektu jest zwiększanie

wag proporcjonalnie do potencjału postsynaptycznego określonej synapsy. Ponieważ ładunek zawsze ulega częściowemu rozładowaniu w czasie, wzrosty wag zostały znacznie spowolnione (SNN11a), co jednak nie rozwiązało problemu, m.in. dlatego, iż wagi ujemne zanikały, gdyż zgodnie algorytmem uczenia tej postaci – mogły tylko rosnąć.

W modelu SNN11b z tego powodu została określona wartość minimalna ładunku q, poniżej którego synapsę uczono w kierunku hamowania. W modelu tym występują dwie przeciwstawne siły, które mogą zmienić wartość synapsy zarówno w górę jak i w dół. Niestety pomimo wielu prób nie udało się ustalić prawidłowej wartości q, być może ona nie istnieje. Może o tym świadczyć fakt istotnej nieliniowości uczenia związanej z wartością q, nie mającej odpowiednika w naturze. Otóż wartości ładunku bliskie q, ale mniejsze od q powodują podczas odpalenia bardzo duże zmniejszenie wartości wagi. Zupełnie odwrotnie dzieje się dla q bliskiego 0. Można przyjąć, iż założenie modyfikacji wagi proporcjonalnie do potencjału postsynaptycznego w połączeniu z zaburzeniem wprowadzanym przez ustaloną wartość q nie przynosi zamierzonego efektu oraz odbiega znacząco od natury, zatem ta droga badań zostanie w tym punkcie zakończona.

Nie oznacza to, iż proporcjonalność szybkości uczenia synaps do



Rys. 55 – SNN11d – polaryzacja wag dość mała, ale widać wyraźny postęp względem SNN11a-c

ładunków na nich zgromadzonych jest skazana na niepowodzenie.

Rozwiązaniem, które wprowadza wiele ograniczeń, ale może przynieść dobre efekty w walce ze zjawiskiem spotkanym w SNN11a, jest normalizacja wag po każdym odpaleniu neuronu.

$$\begin{gathered}\vec{W}\, n_{j,t} = p_{j,t} * \vec{W}_{j,t} \\ \vec{W}_{j,t} = \left(w_{1,j,t}, \ldots, w_{Ij,t}\right) \\ \left(\vec{W}\, n_{j,t}\right) = 1\end{gathered} \qquad (40)$$

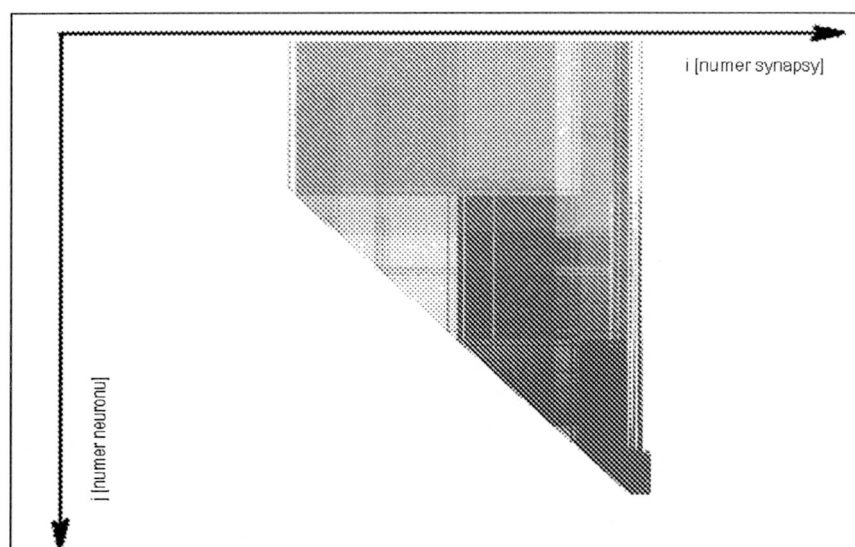
gdzie:

$Wn_{j,t}$ – wektor wag *j*-tego neuronu po normalizacji

$W_{j,t}$ - wartość wektora wag *j*-tego neuronu w chwili *t* po modyfikacjach w czasie uczenia

$p_{j,t}$ – liczba rzeczywista

W ten sposób zachowujemy pełną liniowość uczenia dla wszystkich potencjałów, tracimy możliwość operowania na wagach ujemnych, jednakże możemy z góry ustalić ogólną aktywność neuronu (SNN11c). Dodatkowo pojawiło się kilka mankamentów ze skalowaniem sieci, np. neurony na początku sieci, które z założenia struktury wykorzystują tylko część synaps, na tych synapsach ustawiły bardzo wysokie wagi, ponieważ pozostałe były nieużywane. W modelu SNN11d zwiększono wartości docelowe progu zgodnie ze wzorem



Rys. 56 – Matryca wag SNN11g zawiera elementy świadczące o podziale neuronów na kilka różnych grup

$$h_{j,t+1} = h_{j,t} + \frac{\frac{7}{8}\sum_i w_{i,j,t} * x_{i,j,t} - h_{j,t}}{3} \qquad (41)$$

gdzie:

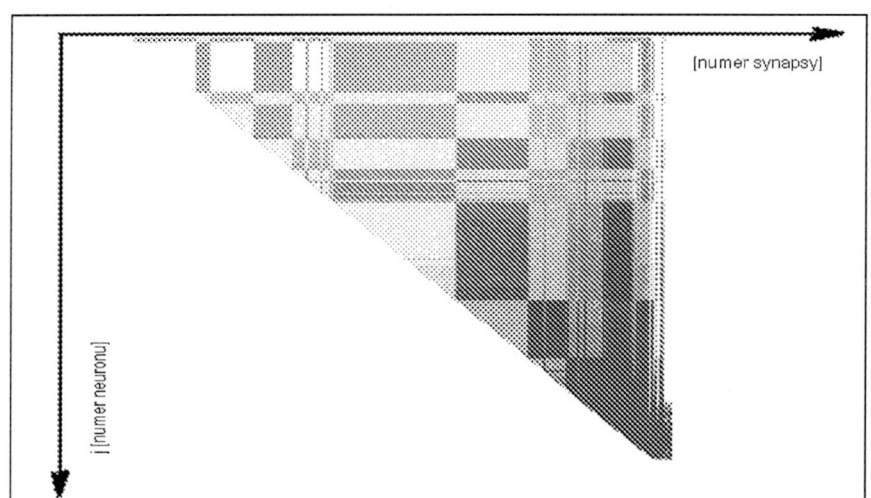
$h_{j,t}$ - wartość progu neuronu j w chwili t

$w_{i,j,t}$ - wartość i'tej wagi neuronu j w chwili t

$x_{i,j,t}$ - wartość i'tego wejścia neuronu j w chwili t

oraz zwiększono o 20% wartość maksymalnej modyfikacji wagi, co spowodowało większe stopniowanie i uwypuklenie wpływu wartości potencjałów postsynaptycznych. Efekty widać na matrycy wag jako zwiększenie polaryzacji względem poprzedniego modelu (rys. 55)

Model SNN11e zawierał nowy element w postaci wylosowanych wag



Rys. 57 – Zmiany w modelu SNN11h spowodowały znaczne urozmaicenie matrycy wag względem SNN11g

początkowych. Element ten okazał się zupełnie nieistotny dla pracy sieci. Zwiększenie sąsiedztwa w SNN11f wywołało podobne efekty jak w modelach SNN10. Głównym mankamentem stał się zbyt mały udział sygnałów wejściowych w ogólnym pobudzeniu sieci. Jednak zmieniając funkcję modyfikacji progu –

$$h_{j,t+1} = h_{j,t} + \frac{\sum_i w_{i,j,t} * x_{i,j,t} - 1 - h_{j,t}}{3} \qquad (42)$$

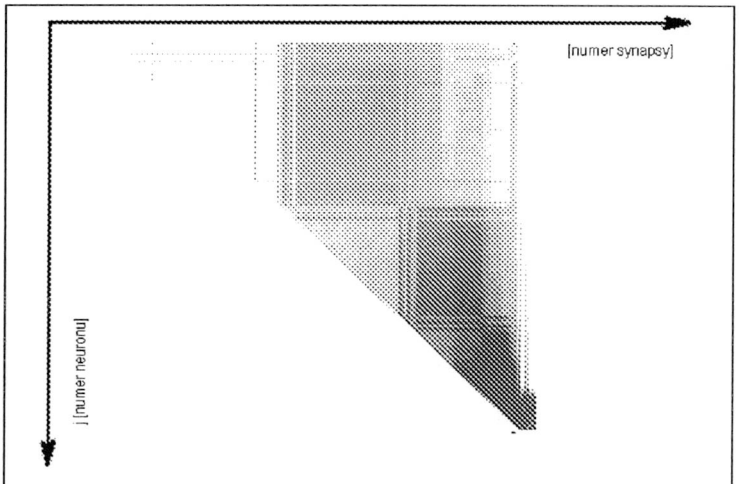
gdzie:

$h_{j,t}$ - wartość progu neuronu j w chwili t

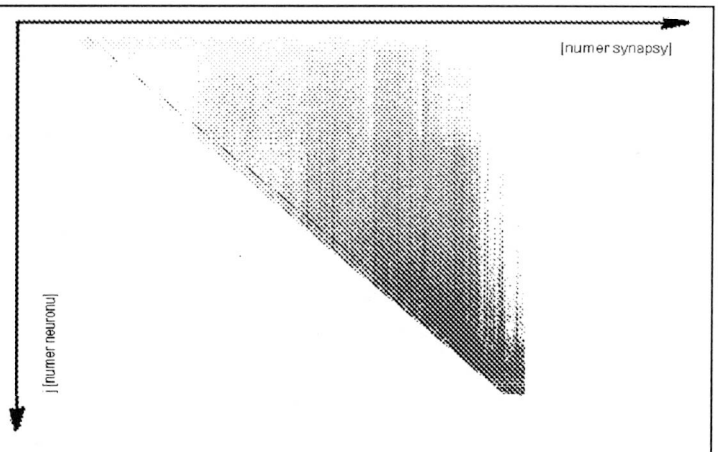
$w_{i,j,t}$ - wartość i'tej wagi neuronu j w chwili t

$x_{i,j,t}$ - wartość i'tego wejścia neuronu j w chwili t

oraz modyfikując mechanizm normalizacji, określając sumę wag neuronu jako założoną wartość i zmniejszając ją względem poprzedniego modelu, w SNN11g uzyskano znacznie lepsze wyniki a efekty synergii neuronów możemy obserwować na matrycy wag rys. 56. Lekkie zwiększenie różnorodności neuronów zostaje też uzyskane w SNN11h poprzez zwiększenie stałej oznaczającą maksymalną modyfikację wagi podczas odpalenia. Rys. 57.

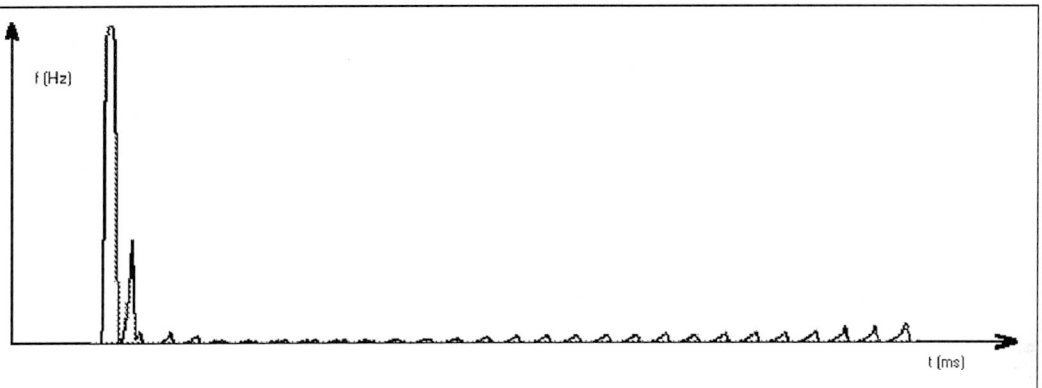


Rys. 58 – Duże jednolite powierzchnie w matrycy wag modelu SNN11i świadczą o jego małych możliwościach przetwarzania danych wejściowych



Rys. 59 – Matryca wag modelu SNN11j w początkowym stadium przetwarzania, wczesny etap uczenia

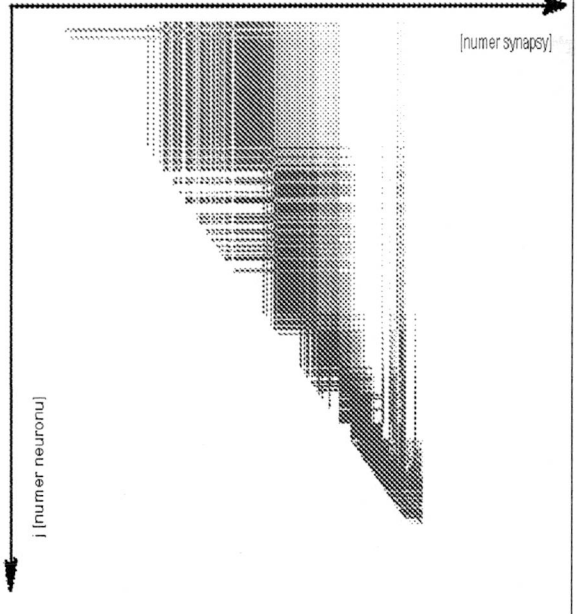
Przetwarzane dane wejściowe o względnie dużej częstotliwości, wprowadzone jeszcze podczas problemów w SNN10 zostały zmienione na dane o znacznie dłuższym okresie i mniejszej częstotliwości średniej odpaleń neuronów, co zbliżyło model do naturalnego. W modelu SNN11i zmniejszono jednocześnie maksymalną możliwą modyfikację wagi podczas odpalenia do wielkości używanej w SNN11g. Rys. 58 nie wykazuje żadnych efektów synergii. Jednakże w wyłączając w obecnym modelu mechanizm wcześniej używany do strojenia częstości odpaleń (zmodyfikowana funkcja refrakcji) i pozostawiając



Rys. 60 – Pierwsze efekty synergii są najwyraźniej zauważalne na wykresie średniej częstotliwości odpaleń neuronów w sieci

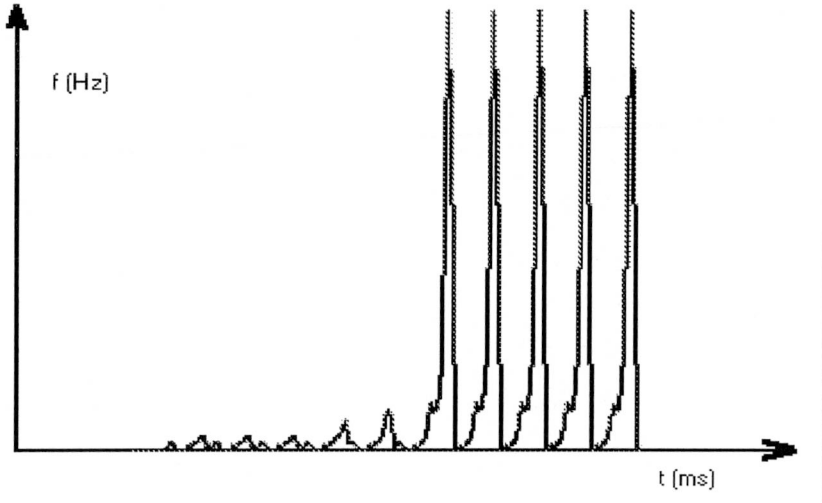
pozostałe elementy jak w SNN11i otrzymujemy model SNN11j, który posiada właściwości powolnej i systematycznej samoorganizowalności, nawet przy utrzymaniu danych wejściowych o względnie małej częstotliwości i stosunkowo dużym okresie, do której to cechy dążyliśmy w ostatnim okresie doświadczeń.

Ponieważ SNN11j wykazuje się ciekawymi



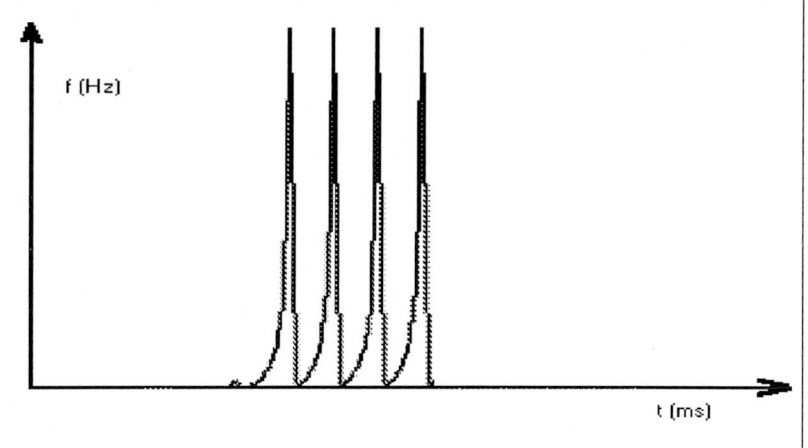
Rys.61 – Matryca wag modelu SNN11j po dłuższym okresie uczenia

właściwościami, jego algorytm uczenia kryje dość duże możliwości a działająca i użyteczna w tym zestawieniu symulacja ładunków postsynaptycznych daje szerokie możliwości do uogólniania rozwiązania, został on poddany kilku bardziej zaawansowanym próbom polegającym na zbadaniu prędkości dostosowania się do nowego wzorca, o dużym podobieństwie do pierwotnego wzorca. Interesującym w tym przypadku jest czas dostosowania się sieci, a właściwie najistotniejsza jest tu proporcja tego czasu do okresu podawanego wzorca. Drugim doświadczeniem jest prędkość synchronizacji z danymi podanymi do sieci asynchronicznie o zawartości zgodnej ze wzorcem. Jest to jednocześnie parametr mówiący o szybkości rozpoznawania.



Rys. 62 – Odpowiedź sieci podczas adaptacji do nowego wzorca o podobnej budowie

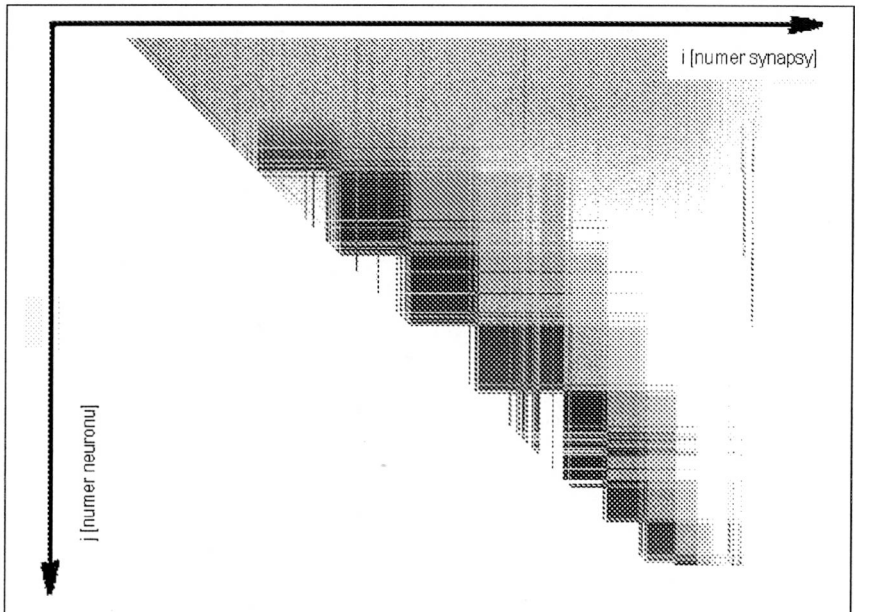
W pierwszym doświadczeniu dane wejściowe zostały uszczuplone względem wzorca o jedną aktywację. Sieć posiada ciągłą zdolność do adaptacji, ponieważ ma włączony algorytm uczenia. Na wykresie rys.62 obserwujemy, iż po pięciu prezentacjach danych odpowiedź sieci stała się



Rys. 63 – Odpowiedź sieci podczas adaptacji do nowego wzorca o podobnej budowie

identyczna do wcześniej wywoływanej przez wzorzec. Wynik ten można uznać za bardzo dobry. W drugim doświadczeniu jako danych użyto już nowo wyuczonego wzorca. Wyuczony wzorzec został podany do sieci zupełnie asynchronicznie, bez żadnej sekwencji startowej. Sieć odpowiedziała pełną mocą już przy pierwszej prezentacji danych. Jest to na tego typu model bardzo dobry wynik.(por. IMPL2). Sytuację tę obrazuje rysunek Rys. 63.

Testując różnego typu dane wejściowe, można zauważyć, iż w modelu SNN11j występuje pewna nieefektywność pracy niektórych obszarów neuronów. Jest to zjawisko, które zostało już wcześniej zidentyfikowane, polegające na niedopasowaniu systemu normalizacji wag do ilości możliwych użytych synaps ze względów struktury, ograniczeń w sprzężeniach itp. Ustawiając współczynnik sumy wartości wag na względnie dużym poziomie niszczymy przetwarzanie sieciowe już na samym początku sieci, gdyż początkowe neurony wszystkie używane wagi ustawiają na wartości wysokie, a same z kolei generują sygnały zupełnie nie przydatne podczas przetwarzania, wprowadzając do systemu zbędne szumy. Ustawienie natomiast mniejszych wartości współczynnika sumy wagi powoduje sytuację w której neurony znajdujące się w ostatnich partiach sieci, które posiadają kilkaset aktywnych synaps, nie są wstanie zróżnicować wag w sposób odpowiedni prawidłowej pracy.



Rys. 64 – Odpowiedź sieci podczas adaptacji do nowego wzorca o podobnej budowie

Zapobiegając temu zjawisku, w modelu SNN 11k zostaje dla każdego neuronu obliczony współczynnik aktywnych synaps. Współczynnik ten

modyfikuje z kolei docelową sumę wszystkich wag. Jest to widoczne jako równomierne rozłożenie wag na matrycy wag rys. 64. Modyfikacja ta umożliwia przetwarzanie większych wzorców przy tej samej wielkości sieci.

Formalny opis neuronu powstałego modelu SNN11k możemy przedstawić następująco:

$$y_{j,t+1}=\left\{\begin{array}{l} 1 \Leftrightarrow \sum_{i=0}^{I-1} \frac{w_{i,j,t} * x_{i,j,t}}{W * I} > h_{j,t}+r(j,t) \\ 0 \Leftrightarrow \sum_{i=0}^{I-1} \frac{w_{i,j,t} * x_{i,j,t}}{W * I} \leq h_{j,t}+r(j,t) \end{array}\right\} \qquad (43)$$

gdzie:

$h_{j,t}$ - wartość progu neuronu j w chwili t

$r(j,t)$ – wartość refrakcji neuronu j w chwili t

$w_{i,j,t}$ - wartość i'tej wagi neuronu j w chwili t

$x_{i,j,t}$ - wartość i'tego wejścia neuronu j w chwili t

$$x_{i,j,t+1}=\frac{x_{i,j,t}}{2}+z_{i,t} \qquad (44)$$

$$z_{i,t}=\left\{\begin{array}{l} Z_{i,t} \Leftrightarrow i<L \\ y_{i-L,t} \Leftrightarrow i \geq L \end{array}\right\} \qquad (45)$$

gdzie:

Z_{it} – wartość i'tego wejścia do sieci w chwili t

L – ilość wejść do sieci

$$\begin{array}{l} I=L+J \\ 0 \leq j<J \\ \forall t \, \forall j \, \forall i \quad w_{i,j,t} \leq W \end{array} \qquad (46)$$

$$h_{j,t+1}=\begin{Bmatrix} h_{j,t}+\dfrac{\sum_i w_{i,j,t}*x_{i,j,t}-h_{j,t}}{k} \Leftrightarrow y_{j,t}>0 \\ h_{j,t}-a \Leftrightarrow y_{j,t}\leq 0 \end{Bmatrix} \quad (47)$$

$$w_{i,j,t+1}=\begin{Bmatrix} w_{i,j,t}*b*x_{i,j,t} \Leftrightarrow y_{j,t}>0 \\ w_{i,j,t} \Leftrightarrow y_{j,t}\leq 0 \end{Bmatrix} \quad (48)$$

gdzie:

c_j – parametr stały danego neuronu

a,k – stałe danego modelu

b – współczynnik zależny

Warto zwrócić uwagę na wzór nr 44. Pomimo jego prostoty jest on wyznacznikiem podstawy czasu całego modelu, co ma podstawowe znaczenie w sieciach pulsujących. Opisuje on zachowanie się potencjału postsynaptycznego w synapsie z uwzględnieniem docierających nowych pobudzeń oraz upływającego czasu. W praktyce, krok symulacji został tak dobrany, aby ten wzór był możliwie prosty, co ma bezpośredni wpływ na czas symulacji.

5.3.3 Unikalny automatyczny mechanizm dostrajania neuronu

Jak już zostało wspomniane przy omawianiu wyników poszczególnych symulacji, neurony modeli SNN oraz części IMPL posiadają wiele zależności funkcyjnych pomiędzy wartością progu, amplitudy refrakcji oraz wartości aktywacji liczonej podczas aktywacji neuronu. Te zależności funkcyjne tworzą w modelach SNN specjalnie dla nich opracowane mechanizmy autoregulacji nazwane PRA. Skrót posiada swą genezę w pierwszych literach nazw obiektów uczestniczących w mechanizmie, czyli „próg, refrakcja i aktywacja". Jest to, jak do tej pory, unikalny mechanizm występujący w SNN, którego odpowiednika nie udało się odnaleźć w literaturze. W początkowym okresie konstruowania ww. mechanizmu były to niespójne atrapy przystosowujące nadmiarowe neurony do

zmiany swojej funkcji w sieci, lub, przede wszystkim, zmuszające neurony nieużywane do wykonania jakiejkolwiek akcji, która mogłaby doprowadzić do wkomponowania nigdy lub bardzo rzadko aktywnego neuronu w strukturę sieci, a w rezultacie powstało z tego bardzo finezyjne narzędzie. Podstawowym trikiem było systematyczne lub losowe zmniejszanie wartości progu takich neuronów o bardzo małe wartości (IMPL2). Neuron zatem miał szansę powolnego zbliżania warunku aktywacji do jego spełnienia oraz wystartowanie algorytmu uczenia w tym bardzo ciekawym z punktu widzenia zawartości synaps momencie, który to praktycznie przesądzał o całej przyszłości neuronu, jego funkcji i klastrze do którego został przyporządkowany lub który właśnie on stworzył. Kolejnym mechanizmem autoregulującym, zapobiegającym zbyt mocnym lub bezwyjściowym sprzężeniom zwrotnym jest modyfikowanie wartości progu o wartość proporcjonalną do różnicy wartości aktywacji i progu podczas momentu aktywacji. Współczynnik proporcjonalności był dobierany empirycznie (i zwykle jego wartość wynosząca *3* przynosiła najlepsze efekty). Wyjście ze sprzężenia w dość krótkim czasie jest podstawowym warunkiem, w przeciwnym razie pamięć długoterminowa zdekomponowana na wagi synaptyczne ulega zniszczeniu. Kolejnym elementem jest wprowadzenie refrakcji. Powoduje ona dodatkową dynamikę w sieci . W naturalny sposób rozbija ona dodatnie sprzężenia zwrotne, czyniąc je znacznie słabszymi. Do tej pory nie zostało określone jednoznacznie, czy jej działanie jest pozytywne w procesach przetwarzania informacji[2], na pewno jednak zmienia działanie sieci. Można stwierdzić, że refrakcja w modelach IMPL i SNN wprowadza elementy losowe, co z jednej strony powoduje różnicowanie neuronów, przyczyniając się do powstawania większej ilości klastrów, z drugiej strony zmniejsza prędkość lub zabrania neuronowemu wektorowi wag dążyć w określonym kierunku, przesuwając jego momenty aktywacji względem pierwotnych. Refrakcja rozbija dodatnie sprzężenia zwrotne, jednakże powoduje powstawanie nieoczekiwanych i powtarzających się innych sprzężeń powstałych z rozbicia sprzężenia globalnego. Walka z nieoczekiwanymi sprzężeniami jest bardzo trudna i zawsze wprowadza mniejsze lub większe spustoszenia zarówno w pamięci krótko- jak i długo- terminowej. Badaniem dynamiki zachowania sieci nie podlegającej uczeniu oraz jej przechodzeniem ze stanu synchronicznych

[2] z wyjątkiem jednej zależności, która zostanie opisana w tym rozdziale

niezależnych odpaleń do stanu synchronicznej aktywacji jako funkcji ilości połączeń zwrotnych, zarówno dodatnich jak i ujemnych, oraz średnich częstotliwości odpaleń neuronów zajęli się Y. Aviel, C. Mehring, M. Abeles i D. Horn w pracy [14]. Nie można zatem jednoznacznie określić przydatności wprowadzenia refrakcji do modelu, jednakże w modelach z refrakcją został zaobserwowany w tej pracy następujący schemat zależności, którego opisu nie znalazłem w omawianej powyżej literaturze, a który ma bardzo istotne znaczenie dla procesów informacyjnych neuronu. Wszystkie wymienione elementy zaczęły funkcjonować zgodnie z następującym schematem:

1.wartość progu neuronu zmodyfikowana o wartość refrakcji dąży (zgodnie z wybraną arbitralnie w danym modelu funkcją) chwilowo do wartości progu, aby po określonym czasie refrakcja przyjęła wartość 0. Wartości chwilowe amplitudy refrakcji są zwykle proporcjonalne do progu dla danego neuronu, aby refrakcja miała równie duże oddziaływanie w neuronach zarówno bardzo aktywnych i przyporządkowanych do wyróżniania określonej klasy abstrakcji jak i neuronów niedotrenowanych.

2.refrakcja wpływa bezpośrednio na moment pobudzenia neuronu, a dokładniej na jego przesunięcie w czasie względem momentu pobudzenia, który byłby wyliczony bez uwzględnienia wartości refrakcji

3.pobudzenie neuronu wyznacza moment liczenia całkowitej aktywacji, która to aktywacja będzie przekazana do algorytmu uczenia. Jeśli moment pobudzenia został nieznacznie przesunięty, wartość aktywacji może być różna od tej z pierwotnego momentu (ale muszą być spełnione warunki aktywacji, czyli przekroczenia progu zmodyfikowanego o refrakcję), a co ważniejsze, wektor aktywnych synaps może mieć zupełnie inną wartość.

4.zmieniona względem pierwotnej wartość aktywacji wpływa bezpośrednio na modyfikację progu, ponieważ zwykle wartość progu dąży do wartości aktywacji lub do wartości proporcjonalnej do aktywacji

Jak można zauważyć, w dalszym ciągu najważniejsze pozostają dane wejściowe wyznaczające zgrubnie moment pobudzenia neuronu, jednakże ww. mechanizm stabilizuje częstotliwość pracy neuronu, a co ważniejsze – aktywnie wpływa na informację zapisaną w pamięci krótko- a przede wszystkim długo-

terminowej. Kolejność przyczynowo-skutkowa tworzy listę elementów działających w silnej pętli sprzężenia zwrotnego, a jest to kolejno: „**p**róg, **r**efrakcja, **a**ktywacja", gdzie kolejne elementy powiązane są silnymi zależnościami funkcyjnymi, umożliwiając powstanie finezyjnego mechanizmu synchronizującego pulsującą sieć neuronową z danymi wejściowymi.

5.4 Kryteria weryfikacji algorytmów uczenia oraz pomiary ich skuteczności

Dążąc do zobiektywizowania porównań jakości poszczególnych modeli i przyśpieszenia eliminacji modeli nie rokujących nadziei na powstanie w nich pamięci długo- lub krótko- terminowej zostały opracowane trzy metody pozwalające przeprowadzić te operacje w sposób automatyczny.

Pierwsza z nich bazuje na wykresie ilości odpaleń neuronów w czasie pracy (i uczenia) sieci. Pamiętając, że symulacja sieci odbywa się ze zdefiniowanym krokiem równym 2ms, każdemu takiemu okresowi możemy przypisać określoną liczbę neuronów, które uległy aktywacji w tym okresie. Podczas tego testu wzorzec lub ciąg testowy powinien być przedstawiany sieci cyklicznie.

Pomocniczo zdefiniujemy funkcję *G* badającą korelację :

$$def.\ G(a,b)=\underset{t=0}{\overset{T_I}{corr}}\left(z(a+t),z(b+t)\right) \tag{49}$$

gdzie:

T_I – czas trwania wzorca

$z(t)$ – liczba odpalonych neuronów w kroku symulacji nr t

$$G(t_0,t_I)\gg G(t_0,t_z) \tag{50}$$

$$(t_z-t_n)>T_I,t_n\in\{t_0,\ t_1,\ t_2,\ \ldots.\} \tag{51}$$

gdzie:

t_n - chwila rozpoczęcia prezentacji wzorca

We wzorze (50) zakładamy, że sygnał nie będący wzorcem jest od niego statystycznie różny w myśl kryterium (1),(3),(4) lub (5).

Jeżeli wystąpi korelacja pomiędzy ilością odpalonych neuronów oraz aktualnym wektorem wzorca np. pojawią się pewne okresowości na wykresie częstotliwości odpaleń neuronów i okres będzie równy czasowi trwania wzorca możemy stwierdzić, że sieć wykazuje oznaki prawidłowej reakcji na wzorzec. Zbadanie tej właściwości pozwala stwierdzić, że sieć nie wpadła w globalne, niekończące się sprzężenie zwrotne.

Metoda ta nie może być stosowana podczas kilku pierwszych prezentacji zbioru uczącego lub wzorca, kiedy następuje wstępne organizowanie się sieci, co często prowadzi do bardzo burzliwych reakcji sieci.

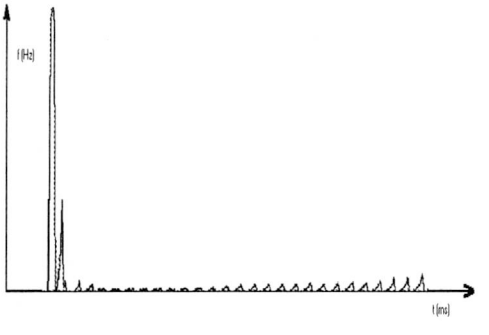
Jeśli korelacja istnieje możemy pokusić się o sprawdzenie tendencji jej zmian:

$$G\left(t_{n+1},t_n\right)-G\left(t_n,t_{n-1}\right)=c_n \qquad (52)$$

W ten sposób otrzymujemy kolejny parametr z tej metody. Jeśli stopień korelacji z czasem wzrasta, oznacza to, że badany algorytm jest lepszy niż mogliśmy ocenić w pierwszych pomiarach. Prędkość wzrostu korelacji daje nam trzeci parametr:

$$c'_n=c_n-c_{n-1} \qquad (53)$$

Parametry mogą być użyte do automatycznego porównywania jakości poszczególnych algorytmów uczenia. Przykładowy wykres przedstawia rys. 65 [24]



Rys. 65 – Wykres częstotliwości podczas udanego procesu uczenia

Drugie kryterium oceny jakości algorytmów opiera się na analizie zmian zachodzących w matrycy potencjałów postsynaptycznych podczas uczenia sieci oraz w mapie aktywnych neuronów w każdym kolejnym kroku symulacji. Zarówno na matrycy potencjałów, jak i na mapie aktywnych neuronów, możemy zaobserwować mniej

lub bardziej skoncentrowaną falę odpowiednio potencjałów postsynaptycznych o wysokiej wartości lub pobudzonych neuronów, co możemy zapisać następująco:

$$\forall t{:}t-t_n<T_I \quad \exists b\in\langle 1, I\rangle$$
$$\sum_{j=0}^{J}\begin{Bmatrix}0 & :q_{b,j,t}=0\\ 1 & :q_{b,j,t}>0\end{Bmatrix}>d*J \qquad (54)$$

gdzie:

q_{ijt} – wartość ładunku postsynaptycznego *i*'tej synapsy *j*'tego neuronu w chwili *t*

d – stała bliska 1 np. *d* = 0.9

Mierzenie spójności fali uformowanej w sieci, w myśl wzoru (54), w odpowiedzi na na wzorzec lub ciąg testowy może być kolejnym parametrem porównawczym.

Trzecia metoda jest ściśle związana z modyfikacjami matrycy wag synaptycznych. Przy założeniu sąsiedztwa neuronów oraz ilości klas, na które podzieliły się neurony w procesie specjalizacji swojej funkcji w procesie dekompozycji problemu globalnego, następujący wzór szacuje stopień rozdrobnienia klas:

$$v=\frac{1}{W^2 I*J}\sum_{j=0}^{J-1}\sum_{i=1}^{I-1}\left(w_{i,j,t}-w_{i-1,\,j,t}\right)^2 \qquad (55)$$

Używanie trzeciej metody powinno być poprzedzone użyciem dwóch poprzednich w odniesieniu do danego modelu, o ile to jest możliwe, a w szczególności pierwszej. Wysoka wartość parametru *v* oznacza dobrą jakość algorytmu uczenia.

Dość ciekawe procesy prowadzące do wykształcenia się dużej liczby klas neuronów możemy również obserwować w postaci wizualnej, przekształcając matrycę wag do postaci czarno-białej bitmapy, gdzie

odpowiedni stopień zaciemnienia danego punktu jest odpowiedni do wartości wagi reprezentowanej synapsy. Jeżeli po określonym czasie pracy sieci bitmapa będzie zawierała wiele detali i nieregularnych kształtów - oznacza to iż algorytm rokuje nadzieje na wykształcenie pamięci długoterminowej.

Zgodnie z założonymi kryteriami pokrycia wzorca (rozdział drugi) przez sygnał wejściowy zostały spreparowane odpowiednie dane testowe oraz przeprowadzone symulacje celem obiektywnego zweryfikowania skuteczności oceny wyżej wymienionych trzech metod. Jako obowiązujące i nie preferujące określonego typu sieci kryterium danych wejściowych wybrano (3), gdyż modele sieci posiadające symulacje ładunków postsynaptycznych znacznie lepiej radzą sobie z nieliniowością podstawy czasu (4),(5), szczególnie gdy lokalne zaburzenie nie przekracza 8 ms, co w modelu przekłada się na $e<4$. Modele te wykazują bardzo dużą tolerancję nawet na zmianę kolejności wektorów w ww. okresie. Każdy rodzaj sieci był testowany od czterdziestu do stu razy. Wyniki zostały zamieszczone w tabeli 66. *Mała sieć* została zdefiniowana jako zbiór mniej niż 30 neuronów. *Duża sieć* była testowana na zbiorze 400 i 4000 neuronów.

Rodzaj sieci/ kryterium weryfikacyjne	Metoda pierwsza, bazująca na ilości odpaleń	Metoda druga, bazująca na istnieniu fali ładunków	Metoda trzecia, bazująca na matrycy wag
Mała sieć binarna	a) nieużyteczna	b) nieużyteczna	c) 0,570
Duża sieć binarna	d) 0,963	e) nieużyteczna	f) 0,9375
Sieci z symulacją ładunków postsynaptycznych	g) 0,978	h) 0,989	i) 0,945
Sieci z symulacją refrakcji	j) 0,905	k) nie użyteczna	l) 0,554
Sieci z symulacją ładunków postsynaptycznych oraz refrakcji	m) 0,932	n) 0,988	o) 0,629

Tabela 66 – Wyniki skuteczności metod weryfikacji algorytmów uczenia

Objaśnienie skrajnych przypadków testu i błędnych przyporządkowań:

a) Takie rodzaje sieci mają małą maksymalną częstotliwość odpaleń neuronów ze względu na małą ich ilość. Czyni to tę metodę stosunkowo mało czułą.
b) Sieć jest zbyt mała aby móc mówić o skoncentrowaniu ładunków na danym obszarze i wytworzeniu przesuwającej się fali.
c) Bardzo słabe wyniki tej metody w przypadku małych sieci są skutkiem akceptowania dużej liczby błędnych algorytmów. Mała matryca wag powoduje dość małą skalę rozpiętości parametru *d* obarczoną względnie dużym błędem statystycznym.
d) Jedynie kilka sieci o bardzo małej amplitudzie częstotliwości odpaleń udało sie błędnie zaakceptować jako dobre. Sieci te transmitowały dane z wejścia do wyjścia bez jakichkolwiek transformacji, nie posiadały pamięci.
e) Metoda badania potencjałów postsynaptycznych może być użyta tylko do sieci, które je posiadają.
f) Kilka sieci z małą ilością detali w matrycy wag z powodu zbyt prostego wzorca posiadało prawidłowo działający algorytm, a zostało odrzuconych. Jednocześnie zdarzyło się kilka algorytmów nie wykształcających pamięci, a bardzo różnicujących matrycę wag, które metoda błędnie zaakceptowała.
g) Podobnie jak w d)
h) Jedna sieć nie wykazywała koncentracji ładunków tworzących falę, ale jednocześnie wykazywała właściwości pamięci krótkoterminowej, co oznacza, że algorytm uczenia był poprawny, a został odrzucony przez tę metodę.
i) Podobnie jak w f)
j) Kilka sieci posiadających niską wartość częstotliwości odpaleń neuronów, które przesyłały sygnały wejściowe bezpośrednio na wyjście, zostało błędnie zaakceptowanych. Jednocześnie pomyłki zdarzyły się również w przeciwnym kierunku, tzn. refrakcja stabilizowała częstotliwościową odpowiedź sieci i kilka prawidłowo działających algorytmów zostało odrzuconych.
k) Podobnie jak w e)
l) Dynamiczny model refrakcji zaburza w pewnym stopniu integralność sąsiedztwa w topologicznym podejściu, ponieważ potrafi ono być podzielone np. na dwa obszary sąsiedztwa wzajemnie się przeplatające i zajmujące ten

sam obszar. Logicznie są jednak dwoma niezależnymi sąsiedztwami. Jeśli sąsiedztwa się pokrywające należą do różnych klas neuronów - powstaje bardzo zróżnicowana matryca wag, co nie jest proporcjonalne do ilości faktycznych klas wykształconych w sieci. Metoda nie jest akceptowalna dla tej struktury sieci.

m) Kilka sieci wykazywało spełnienie tego kryterium, ale ich algorytm uczenia utykał w pewnym punkcie i nie doprowadzał skutecznie do powstania pamięci, co oznaczało błędną klasyfikacje tego kryterium. Jeden model sieci o złym algorytmie uczenia spełnił kryterium przez przypadek – nie było to powtarzalne w przyjętej próbce.

n) Podobnie jak w h)

o) Podobnie jak w l)

Podsumowując wyniki pomiarów skuteczności metod weryfikacji algorytmów uczenia możemy jednoznacznie powiedzieć, że żadna z metod nie nadawała się do weryfikacji działania małych sieci neuronowych, posiadających do kilkudziesięciu neuronów. Metoda druga, z natury rzeczy, nie ma możliwości operowania na sieciach nie posiadających włączonej symulacji potencjałów postsynaptycznych. Metoda trzecia wypadła bardzo słabo przy sieciach z modelowaniem refrakcji względnej i bezwzględnej. Większość tych przypadków dotyczy zatem albo przypadków mało ciekawych lub skrajnych np. użycie takich metod do badania małych sieci jest pewnym przerostem narzędzi nad potrzebami, lub brak możliwości obserwacji fali potencjałów postsynaptycznych w miejscach nie posiadających tych potencjałów. W pozostałych przypadkach skuteczność metod jest bardzo wysoka, maksymalnie sięgająca niemalże 99%, a oscylująca dla różnych testów na poziomie 96-97%! Co ważniejsze, dotyczy to właśnie najbardziej obiecujących obszarów badań, czyli sieci dużych, posiadających modelowanie ładunków synaptycznych. Nawet w przypadku analizy sieci wykorzystujących różne rodzaje refrakcji, które z założenia wykazują większe zróżnicowanie i są trudniejsze do weryfikacji, metody z grupy pierwszej i drugiej wykazują skuteczność pomiędzy 90 a 98%. Tworząc kryteria wykorzystujące jednocześnie po dwie lub trzy metody możemy jeszcze zwiększyć precyzje weryfikacji algorytmów.

5.5 Zestawienie składowych modeli oraz algorytmów uczenia

a) Warunek aktywacji neuronu

$$y_{j,t+1}=\begin{cases} 1 \Leftrightarrow \sum_{i=0}^{I-1} \frac{w_{i,j,t} * x_{i,j,t}}{W * I} > h_{j,t} + r(j,t) \\ 0 \Leftrightarrow \sum_{i=0}^{I-1} \frac{w_{i,j,t} * x_{i,j,t}}{W * I} \leq h_{j,t} + r(j,t) \end{cases} \quad (56)$$

gdzie:

$h_{j,t}$ - wartość progu neuronu *j* w chwili *t*

$r(j,t)$ – wartość refrakcji neuronu *j* w chwili *t*

$w_{i,j,t}$ - wartość *i*'tej wagi neuronu *j* w chwili *t*

$x_{i,j,t}$ - wartość i'tego wejścia neuronu *j* w chwili *t*

W- maksymalna wartość wagi

I- ilość synaps każdego neuronu

Refrakcja przyjmowała wartości:

- funkcji stałej o wartości zero, gdy refrakcja nie była modelowana
- funkcji przyjmującej wartości nieskończoność,9,4,2,1,0 mV dla danego neuronu *j* licząc dyskretnie w kolejnych krokach symulacji o długości 2 ms i początku w momencie aktywacji neuronu
- malejącej funkcji linowej (w tym również przyjmującej wartości ujemne) o wartości początkowej w momencie odpalenia neuronu zależnej od wartości jego aktywacji. Nachylenie funkcji jest zależne od wartości progu, tak że po ustalonej dla danego modelu ilości kroków osiąga wartość przeciwną do progu.

b) Przepływ ładunków przez synapsę

Przepływ ładunków przez synapsę i ich akumulacja po stronie postsynaptycznej jest ściśle związana z krokiem symulacji. Przy kroku równym 2ms możemy przyjąć:

- brak symulacji ładunków postsynaptycznych
- zgodnie ze wzorem:

$$x_{i,j,t+1}=\frac{x_{i,j,t}}{2}+z_{i,t} \tag{57}$$

c) Wejście danych do systemu

$$z_{i,t}=\begin{Bmatrix} Z_{i,t} \Leftrightarrow i<L \\ y_{i-L,t} \Leftrightarrow i \geq L \end{Bmatrix} \tag{58}$$

gdzie:

Z_{it} – wartość *i*'tego wejścia do sieci w chwili *t*

L – ilość wejść do sieci

$$\begin{array}{l} I=L+J \\ 0 \leq j<J \\ \forall t \forall j \forall i \quad w_{i,j,t} \leq W \end{array} \tag{59}$$

d) Zmienność progu neuronu

$$h_{j,t+1}=\begin{Bmatrix} h_{j,t}+\dfrac{g\sum_{i=0}^{I} w_{i,j,t}*x_{i,j,t}-h_{j,t}}{k} \Leftrightarrow y_{j,t}>0 \\ h_{j,t}-a \Leftrightarrow y_{j,t}=0 \end{Bmatrix} \tag{60}$$

gdzie:

g – stała modelu, ułamek właściwy lub 1

k – stała modelu, k>1

a – stała modelu

Parametr g jest wyznacznikiem, jak szerokie klasy abstrakcji mogą zostać wykształcone podczas procesu uczenia. Im parametr ten jest bliższy 1, tym bardziej klasa będzie pokrywała jedynie zbiór uczący. Wartość nie powinna być niższa niż 0,3.

Parametr k definiuje tempo zmian progu. W wielu modelach przekłada się to bezpośrednio na czułość sieci na drobne zmiany sygnału wejściowego. Im k przybiera większa wartość, tym model staje się czulszy, ale tym wolniej przebiega uczenie.

e) Zmiany wag synaptycznych

W tym punkcie nie da się przedstawić uogólnienia. Modyfikacje wag to jedna z najważniejszych części modelu odpowiedzialna za powstawanie pamięci długoterminowej, a jednocześnie jedna z najszerszych pól do zbadania. Występują tu różne algorytmy, począwszy od modelujących hamowanie i torowanie, modyfikacji opartych o wyliczenia zmian częstotliwości odpaleń neuronów, sprawdzających hipotezy synaps hamujących i hamowania presynaptycznego, co zostało omówione przy poszczególnych modelach. Możemy jednak z dużym przybliżeniem powiedzieć, że wartości wag zwykle są proporcjonalne do swoich poprzednich wartości oraz sygnału wejściowego (w tej pracy w większości reprezentowanego przez potencjał postsynaptyczny).

$$w_{i,j,t+1} = \begin{Bmatrix} w_{i,j,t} * b * x_{i,j,t} \Leftrightarrow y_{j,t} > 0 \\ w_{i,j,t} \Leftrightarrow y_{j,t} \leq 0 \end{Bmatrix} \qquad (61)$$

gdzie:

c_j – parametr stały danego neuronu

a,k – stałe danego modelu

b – współczynnik zależny

Wektor wag danego neuronu może być normalizowany. O ile na temat zwiększania się wag w procesie uczenia jest sporo hipotez, o tyle bazujących na danych fizjologicznych i dotyczących zmniejszania jest znacznie mniej.

Normalizacja tego wektora wydaje się być biologicznie uzasadniona, jako odzwierciedlająca fizyczne możliwości czułości neuronu. W praktyce proces powinien być nieco sparametryzowany, aby zachować odpowiednią rozpiętość wartości wag:

$$w_{i,j,t+1} = \begin{cases} p_{i,j,t} * w_{i,j,t} * x_{i,j,t} \Leftrightarrow y_{j,t} > 0 \\ p_{i,j,t} * w_{i,j,t} \Leftrightarrow y_{j,t} = 0 \end{cases}$$
$$\vec{W}_{j,t} = (w_{1,j,t}, \ldots, w_{I,j,t})$$
$$(\vec{W}_{j,t+1}) = \sqrt{(I_a)} * \frac{W}{q} \qquad (62)$$

gdzie:

$W_{j,t}$ – wektor wag *j*-tego neuronu

$p_{j,t}$ – liczba rzeczywista

$I_a = S_G + S_d$ – liczba synaps neuronu używanych w danym modelu zgodnie ze wzorem (63)

q- stała modelu

f) Sąsiedztwo oraz dozwolony stopień sprzężeń zwrotnych

Podstawową własnością pulsującej sieci neuronowej, bez której zawładnięcia nie można kroczyć w stronę wykształcania pamięci, jest utrzymanie stanu asynchronicznych pobudzeń podczas procesu uczenia. Jak zostało opisane w poprzednich rozdziałach, zachowanie tej stabilności jest trudne w sieciach ze sprzężeniami zwrotnymi. Natomiast wyeliminowanie sprzężeń zwrotnych diametralnie zmieniłoby możliwości sieci i zbliżyło do prostych sieci warstwowych jednokierunkowych przeliczających określona funkcję.

Doświadczenia wskazują, że priorytetowa okazuje sie w tym zakresie struktura sieci, tzn.:

- definicja sąsiedztwa w przyjętym monolicie,
- zasięg i ilość synaps do neuronów o mniejszych identyfikatorach
- fakt istnienia w strukturze możliwości ciągłego przekazywania kontekstu do sąsiednich obszarów.

Uzasadnienie tych tez wyjaśniam w odwrotnej kolejności niż zostały wymienione. Rekurencyjna sieć pulsująca ma predyspozycje do zachowań synchronicznych. Jeśli owa własność zostanie wykorzystana do synchronizacji z danymi wejściowymi i będzie dotyczyła całej sieci, inne słabsze sprzężenia zwrotne nie będą w stanie "przebić się" w gąszczu tych sprzężeń i będą okresowo przerywane przez pętle silniejsze, z czasem doprowadzając do zaniknięcia tych słabych poprzez przejęcie ich neuronów do swoich funkcji. Ważne jest w tym przypadku, aby sąsiedztwo zdefiniowane w strukturze tworzyło spójny twór, np. w najprostszym przypadku było linowe.

Druga własność wymaga określenia potencjalnej ilości sprzężeń zwrotnych. Aby odróżnić synapsy zwrotne od utrzymujących pierwotny kierunek w architekturze monolitycznej każdemu neuronowi przyporządkowano jego identyfikator, będący kolejną liczbą naturalną. Synapsy do neuronów o mniejszym identyfikatorze są synapsami zwrotnymi. Ich ilość jest określona. Zasięg oznacza maksymalną różnicę wartości identyfikatorów neuronów posiadających synapsę zwrotną. Oczywiście ww. definicje ma tylko charakter umowny na cele tego rozumowania. Zarówno ilość jak i zasięg synaps zwrotnych przekładają się na stabilność pracy sieci w stanie asynchronicznym.

Przechodząc do punktu pierwszego możemy znaleźć dopełnienie konstrukcji. Sąsiedztwo właśnie, budowane celem koncentracji określonych funkcji sieci w określonym topologicznie obszarze, jest jednocześnie ściśle związane z zasięgiem i ilością synaps, w tym zarówno przepływu danych do przodu jak i do tyłu (do neuronów o wyższych i o niższych identyfikatorach względem identyfikatora neuronu wysyłającego sygnał). Wynika z tego, że za właściwości dynamiczne sieci odpowiada definicja sąsiedztwa w modelu.

W testowanych modelach sąsiedztwo było definiowane w oparciu o identyfikator neuronu *j* i miało następującą postać:

$$\begin{matrix} 0 < i < I \\ j - S_D < i < j + S_G \end{matrix} \qquad (63)$$

gdzie:

I – ilość synaps neuronu

j – identyfikator neuronu

S_D – sąsiedztwo dolne

S_G – sąsiedztwo górne

W większości modeli sąsiedztwo było definiowane asymetrycznie. Parametr S_D przybierał zwykle wartość równą I, choć część modeli miało nawet zawężane np. do 12. S_G zwykle wynosił kilkanaście procent z liczby synaps, choć były testowane modele z tym parametrem o wartości od 1 do I.

W modelach ze zdefiniowanym sąsiedztwem ładunki synaps o indeksie i spoza zakresu określonego wzorem (63) nie były brane do liczenia aktywacji oraz nie dokonywano modyfikacji wag im odpowiednich.

6. Porównanie badanych modeli do znanych z literatury

6.1 Wytyczne teorii przetwarzania przy pomocy PNN w czasie rzeczywistym bez stanów ustalonych

Analizując literaturę światową, dotyczącą podmiotu tej pracy, można zauważyć, że fakty wynikające z przedstawionych symulacji i publikowane na konferencjach zostały potwierdzone przez niezależnych badaczy w latach 1995-2001, a część nadal jest przedmiotem analiz. Do podstawowych problemów można zaliczyć teorię budowy synapsy, algorytmu działania synapsy oraz jej modyfikacji w procesie uczenia, dynamiki pracy systemu złożonego z dużej liczny neuronów (większość interesujących efektów powstaje przy liczbie neuronów większej od kilku tysięcy), oraz ściśle z nim związanego algorytmu uczenia.

Problemem fundamentalnym teorii systemów bazujących na motorze pulsujących sieci neuronowych jest konieczność zmiany podejścia do sieci neuronowych jako mniej lub bardziej wyrafinowanej funkcji i uznanie dynamiki sieci oraz braku konieczności osiągania stanu ustalonego jako podstawy do skonstruowania brakującej w tym miejscu teorii. Jej brak, o którym zostało wyraźnie powiedziane w 1996 roku [24][3] jako teorii (a właściwie jej braku) przetwarzania w czasie rzeczywistym bez stanów ustalonych, doczekał się wstępnego opracowania w październiku 2001 roku przez Thomasa Natshlager'a oraz Wolfganga Maass'a [131], co prawdopodobnie wpłynie na zintensyfikowanie prac w tym zakresie przez różne ośrodki badające PNN. Powstałe opracowanie „Neurony pulsujące i wstęp do maszyny skończonych stanów" [131] nie tylko mówi o dużej pustce w tym obszarze, ale proponuje stosunkowo prostą teorię, która przystaje do sieci jednokierunkowych. W tej samej pracy jest podkreślona również rola lokalnej reguły uczenia, co w przyszłości powinno być fundamentem algorytmu uczenia. Przedstawiony

[3] [24] – w rozdziale 3.4: „Stan dynamiczny sieci może być również reprezentacją danych wejściowych, o ile zapewni się mechanizmy to wykorzystujące." oraz w rozdziale 4 wraz z przykładami działających tego typu systemów, powstałych w procesie uczenia, bazującym na lokalnej regule uczenia.

model neuronu pulsującego przez Natschlagera posiada bardzo duże podobieństwa do modeli z symulacją ładunków postsynaptycznych prezentowanych w poprzednich rozdziałach: praca synapsy, refrakcja bezwzględna i względna. Ważnym wkładem [131] jest też usystematyzowanie pojęć poprzez podanie ich definicji oraz ujednolicenie słownictwa w zakresie opisu PNN jako maszyny skończonych stanów. Pulsująca sieć neuronowa jest w tym rozumieniu maszyną przekształcająca ciąg wejściowy w ciąg wyjściowy na bieżąco, nie arbitralnie lecz używając w tym procesie skończonej liczby stanów wewnętrznych maszyny skończonych stanów (FSM). Jest to bardzo istotne z tego punktu widzenia, iż nie traktuje się tu dynamiki sieci jako coś tymczasowego, coś czego celem jest doprowadzenie do ustabilizowania się (jeśli proces będzie zbieżny) i określenia wyniku, jak np. w sieci Hopfielda, ale sieć jako proces staje się głównym obiektem zainteresowania. Sieć nie jest tu też automatem, który odrzuca lub akceptuje określone ciągi wejściowe (jak maszyna Turinga). Spojrzenie jest znacznie szersze.

Wcześniejsza praca W. Maassa z 1996 roku [114], najbardziej zbliżona do tego tematu (PNN jako FSM), zawiera formalne metody przekształcenia maszyny w pulsującą sieć neuronową, ale proces ten nie ma nic wspólnego z algorytmem uczenia. Nie zawiera też praktycznej realizacji, jedynie teoretyczny proces na potrzeby oszacowania możliwości sieci. Ponadto wyniki były też niezgodne z przesłankami biologicznymi np. nie założono jakiejkolwiek redundancji, nienaturalnie złożona funkcja pojedynczego neuronu w systemie itp., co sprawia, że zbytnio odbiegała od naturalnego wzorca PNN. Praca przedstawia natomiast formalne dowody na potencjalną moc PNN jako maszyny. Dowody te obrazują, jak duże możliwości posiada mechanizm PNN np. w jednym z nich jest pokazane, jak maszyna Turinga może być przekształcona i emulowana przez PNN. Jest to zatem ważna cegiełka w teorii działania PNN.

W świetle rysujących się wymagań stawianych przez współcześnie powstającą teorię przetwarzania w czasie rzeczywistym przy pomocy pulsujących sieci neuronowych bez osiągania stanów ustalonych (ostatnio opisywanego skrótem RTCWSS) warto cofnąć się do klasycznych architektur sieci i określić ich pozytywne i negatywne cechy pod kątem możliwości zastosowania w modelach pulsujących. Siecią najbardziej przypominającą w

swym działaniu PNN ze sprzężeniami zwrotnymi jest popularna sieć Hopfielda [63]. Jej atutem jest występowanie sprzężeń zwrotnych i w rezultacie procesu dynamicznego przetwarzania po podaniu danych wejściowych. Jest to cecha, którą posiadają naturalne sieci neuronowe i wysoce pozytywna w świetle w.w. teorii. Jednak podczas tego procesu nie są przetwarzane przez sieć kolejne dane wejściowe, jakby to wynikało z wzorca biologicznego. Kolejna partia danych może być przetworzona w następnym cyklu pracy sieci, a on następuje po osiągnięciu stanu ustalonego w cyklu bieżącym, gdzie czas trwania cyklu nie jest dodatkowo stały ani łatwo przewidywalny. Nie jest to zatem przetwarzanie w czasie rzeczywistym. Poza tym oczekuje się, że sieć Hopfielda osiągnie stan ustalony i to oznacza zakończenie przetwarzania. Samo odłączanie wejść sieci po pobraniu wzorca wejściowego jest tworem bardzo sztucznym, nie mówiąc o tym, że aby zapewnić zbieżność przetwarzania muszą być spełnione określone warunki wag, często możliwe do sprawdzenia tylko centralnie (sprzeczne z lokalną regułą uczenia). Ograniczenia te wynikają z faktu, że za wszelką cenę jest zapewniana zbieżność przetwarzania do stanu ustalonego, co zgodnie z teorią RTCWSS nie jest konieczne.

W maszynie Boltzmana [159] podczas pracy występuje proces dynamiczny, jego efekty działania są pożądane w rozumieniu RTCWSS. Jednak maszyna ta posiada ten mankament, że zgodnie z uogólnionym algorytmem uczenia podanym przez Hintona i Sejnowskiego [61] wymaga scentralizowanego zgromadzenia statystyk na temat sieci oraz wyliczenia poprawek wag. Jest to w konflikcie z lokalną regułą uczenia, gdyby sieć miała uczyć się podczas pracy. Efektem zakończenia przetwarzania jest też stan równowagi, podobnie jak w sieci Hopfielda. Stan ustalony nie może być formą reprezentacji danych zmiennych w czasie.

Klasycznym przykładem braku nauczyciela w sieciach neuronowych jest mapa cech Kohonena [77], a właściwie jej pierwowzór - sieć Wilshawa i von der Malsburga, która dodatkowo nie wymaga przetwarzania scentralizowanego i jest zgodna z lokalną regułą uczenia. Zarówno samoorganizowalność jak i decentralizacja przydatna w RTCWSS ma swój pierwowzór w naturze, dokładniej naśladowanie przyrody w odwzorowaniach świata zmysłów organizmów w ich korze mózgowej np. połączenia kory mózgowej z okiem

tworzą mapę reintopową, z uchem - mapę tonotopową, z receptorami dotyku - mapę somatosensoryczną. Samoorganizowalność tych map jest osiągnięta poprzez wzajemne oddziaływanie neuronów w zakresie pracy i uczenia. Siła wzajemnego oddziaływania jest funkcją zwaną „kapeluszem meksykańskim" [79] str.79, której argumentem jest odległość na zdefiniowanej przestrzeni. Naturalna przestrzeń jest trójwymiarowa, nie jest ona jednak jedyną przydatną w tej sytuacji. Pomimo, że samoorganizowalność nie jest jednoznacznie związana z RTCWSS, powinna stanowić jej nierozłączną część, zwłaszcza że została tutaj użyta pożądana lokalna reguła uczenia.

Bardzo interesująca jest, współcześnie powstała do modelu SNN, praca Massa i Natschlagera [133]. Jest ona rozwinięciem poprzedniej pracy [131] dotyczącej przetwarzania informacji bez konieczności osiągania stanów ustalonych w pulsujących sieciach neuronowych (RTCWSS). Postępowość tej pracy jest ogromna. Zwykle likwidując konieczność dążenia do stanu ustalonego pierwszym przychodzącym na myśl abstraktem odpowiednim jest powstanie określonego atraktora z wektorów generowanych przez wyjście sieci lub przez wektor aktywnych neuronów sieci. Tak również możemy domniemywać było w poprzedniej pracy tych autorów, jak i we wszystkich zaprezentowanych modelach IMPL. I jest to błąd spowodowany prawdopodobnie ukrytym dążeniem do zamodelowania stanu ustalonego przystającego do pracującego modelu PNN. Zgodnie z założeniami, nie jest potrzebny (a dokładniej - jest niedozwolony) jakikolwiek stan ustalony, aby przetwarzać zmienne w czasie informacje za pomocą tego typu struktury. W pracy jest podjęta próba zdefiniowania alternatywnego paradygmatu względem stanów ustalonych w pojęciu Maszyny Turinga lub atraktorów sieci neuronowych. Paradygmat odrzuca konieczność sekwencyjnych przejść pomiędzy ściśle zdefiniowanymi dyskretnymi stanami wewnętrznymi. Co więcej, proponowana maszyna nie posiada ściśle określonych stanów wewnętrznych i jest to jak najbardziej uzasadnione – mówi nam o tym neurobiologia. Jako model PNN proponowana jest maszyna ciekłych stanów (LSM) [150], która w swej postaci jest bardzo dobrym odzwierciedleniem procesów występujących w PNN. LSM posiada natomiast uniwersalną moc obliczeniową (przy spełnieniu kilku warunków) niezależną od jej fizycznej implementacji. Własności LSM zatem przenoszą się na PNN. W zamian poprzednio używanego abstraktu

atraktora w PNN powstają tutaj dwa nowe kryteria wystarczające do kompletnego przetwarzania danych, mianowicie: *zdolność separacji danych* oraz *zdolność uogólnia*nia (aproksymowania).

Zdolność separacji danych przejawia się różnymi trajektoriami stanów wewnętrznych pod wpływem różnych ciągów wejściowych.

Zdolność aproksymacji jest wyznaczana za pomocą jakości mechanizmu przekształcającego określoną grupę zmian stanów wewnętrznych w jeden określony ciąg wyjściowy. Grup zmian stanów wewnętrznych, z przyporządkowanym im ciągiem wyjściowym, może być wiele.

Dość ważną zmianą jest również podejście do autoasocjacyjności neuronu oraz możliwości tworzenia sprzężeń zwrotnych. W poprzedniej pracy na temat RTCWSS [131] domyślenie główna uwaga została skupiona na sieciach jednokierunkowych. Dla tego typu sieci jesteśmy w stanie obecnie zaimplementować dość efektywne metody algorytmu uczenia, w przeciwieństwie do sieci ze sprzężeniami zwrotnymi. Jednakże sieci z pętlami zwrotnymi mają potencjalnie nieporównywalnie większe możliwości zarówno pod względem efektywności pracy sieci, ilości przekształceń matematycznych możliwych w odpowiednio licznej grupie neuronów oraz zupełności możliwych przekształceń sygnału wejściowego. Problemem jest natomiast dopasowanie algorytmu uczenia, co zostało szczegółowo omawiane przy modelach SNN. Zmiana polega na domyślnym założeniu, że sieć powinna mieć bardzo dużo małych obwodów sprzężenia zwrotnego, (szacunkowo ok. 80% synaps powinna pracować w takich obwodach). Zwiększa to diametralnie dynamikę zachowania takiej sieci, niestety autorzy nie podają, jak powinna pracować taka struktura. Wydaje się to samym sednem problemów PNN i wyzwaniem dla wszystkich badaczy tego obszaru. Opracowanie przystającego do tej struktury algorytmu uczenia spełni warunki stawiane LSM i wyzwoli w PNN uniwersalną moc obliczeniową. Zgodnie z symulacjami omówionymi w poprzednich rozdziałach wydaje się, że jednak liczba obwodów sprzężenia zwrotnego powinna być ok. czterokrotnie mniejsza.

Z drugiej strony należy mieć na uwadze, że jeśli to się powiedzie, powstała maszyna z mocą obliczeniową będzie na tyle różna od znanych nam obecnie komputerów, że prawdopodobnie nigdy ona nie zastąpi PC a jedynie go uzupełni (lub odwrotnie – technika PC będzie jej uzupełnieniem). Przemawia

za tym wysoka dynamika systemu i związane z tym ryzyko nieprzewidywalności przy określonych ciągach wejściowych oraz określonych stanach wewnętrznych. Wydaje się też, że składowanie baz danych w których wymagana jest stuprocentowa dokładność, będzie też niemożliwe lub co najmniej okupione nieproporcjonalnie dużym nakładem pracy takiej sieci i znacznie wyższym niż w przypadku klasycznego przetwarzania.

Warto w tym miejscu zaznaczyć, że nie tylko technika PNN natrafiła na problemy z teorią przetwarzania informacji w ostatnich latach. Prace nad skonstruowaniem komputera DNA wprowadzają nas w analogiczne dylematy. Łańcuch kwasu DNA w komórce jest przetwarzany równolegle w wielu miejscach, podobnie jak w wielu miejscach informacja jest przetwarzana przez równolegle pracujące neurony. Już obecnie możemy zaprogramować taki komputer DNA aby określił, czy w danym ciągu jest np. parzysta liczba zer, czy ciąg się zaczyna i/lub kończy określonym symbolem itp. Nikt jednak nie planuje zastosować tego do skonstruowania PC, pomimo że technika ta w wielu miejscach może mieć możliwości nigdy nie osiągalne przez prosta linię rozwoju obecnych PC.

Odchodzenie w wielu dziedzinach od koncepcji komputera zgodnego z maszyną Turinga potwierdza zapotrzebowanie na nowe teorie przetwarzania informacji.

6.2 Inne pulsujące systemy neuronowe – teoretyczne i zaimplementowane

Pulsujące sieci neuronowe to bardzo obiecujący dział sieci neuronowych, pomimo iż w zastosowaniach aplikacyjnych technika ta obecnie przegrywa z klasycznym podejściem SN lub nawet z algorytmami nie neuronowymi. Opisane w tej pracy symulacje tworzą ciąg doświadczeń powiązanych ze sobą związkami przyczyno-wynikowymi. W naturalny sposób stanowią niezależną linię pulsujących sieci neuronowych względem prac innych badaczy zajmujących się tym tematem. Pomimo, że model tworzy odpowiednio dobrany zbiór zaimplementowanych cech i odpowiednie proporcje

zwykle stanowią o jego mocy, możliwe jest też porównywanie poszczególnych elementów występujących w modelach różnych autorów.

Jak pokazują najnowsze badania, sam neuron zintegrowany i pulsujący powszechnie uznawany obecnie jako bazowy (IAF), został już w 1907 roku opracowany przez Lapicque! [1]. Zaproponował on dość prostą synapsę składająca się z kondensatora i rezystora połączonego równolegle do potencjału zewnętrznego. Nie znał on oczywiście wszystkich procesów występujących w neuronie, ale zaproponował właśnie neuron typu IAF, pomijając np. zmienność przepływności membrany w zależności od aktualnego jej potencjału. Przydatność tego modelu można ocenić dopiero obecnie, po tak długim czasie, ponieważ coraz więcej procesów neuronu wydaje się nie mieć związku z przetwarzaniem informacji. Weryfikacja takich faktów nie jest łatwa i wymaga wielu doświadczeń, aby pokusić się o stwierdzenie, że właściwość *x* oraz *y* nie mają ze sobą związku, lub że występujące zależności nie mają znaczenia w interesujących nas procesach.

Jednym z pierwszych symulatorów, w którym potwierdzono fale określonej częstotliwości koncentrujące się topologicznie w określonym miejscu sieci, jest PRISM [71]. Fakt przepływu np. fali potencjałów postsynaptycznych, jako cechy świadczącej o skutecznym przetwarzaniu danych, został omówiony w modelach SNN i jest symptomem wystąpienia synergii neuronów oraz procesu kreowania pamięci krótkoterminowej. System PRISM[4] zawiera elementy zmiany synapsy, uwarunkowane jej historią w systemie, używane do rozpoznawania statycznego wzorca np. liter. Zastosowanie procesów Markowa przy modyfikacji synapsy daje dość duże możliwości, ale prawdopodobnie nie jest wystarczające, ponieważ nie tylko informacje docierające do synapsy mają wpływ na zmianę jej wagi. Brak zależności czasowych lub chwilowych w tym systemie znacznie ogranicza możliwości sieci neuronowej. Natomiast cechą wyróżniającą ten system jest przyjęte kryterium przy określaniu odpowiedzi sieci (zarówno pozytywnej jak i negatywnej), mianowicie: **koncentracji impulsów o kreślonym miejscu sieci**: „Mathematically the wave packet is created when a group of waves all near the same frequency clumps together. The wave packet represents a region where a localized concentration of energy occurs. We're assuming that the coherent, energetic synapses filled

[4] PRISM: http://www.neuralmachines.com/prism/prism.html#Prism

with all kinds of neural dynamics can be modelled by this mathematical construct."[5] („Matematycznie, pakiet fal powstaje gdy grupa fal o podobnej częstotliwości koncentruje się wspólnie. Pakiet fal reprezentuje region gdzie występuje wymieniona koncentracja energii. Zakładamy, że spójne, energetyczne synapsy wraz z różnymi typami dynamiki neuronu mogą być modelowane przez ten konstrukt"). Jest to cechą dość rzadko do tej pory spotykaną a jednocześnie wspólną z modelami SNN. Oznacza to użycie podobnych mechanizmów klasyfikujących w obu rozwiązaniach.

Porównując SNN do współczesnych mu symulatorów, można powiedzieć, że osiągnięta skala sieci w modelach SNN przy złożoności pojedynczego neuronu (np. symulacja potencjałów postsynaptycznych) oraz spełnieniu założeń neuronu typu IAF w tym czasie stawia pracę [26] na pionierskiej pozycji, a zastosowanie PNN z ww. mechanizmami do rozpoznawania względnie dużych ilości danych wejściowych (pojedyncze dźwięki, sylaby) [28] w 1998r nie znajduje odbicia w literaturze. Współczesne jej aplikacje sieciowe nie wykazują globalnych efektów synergicznych, a ich zasadę działania ([112] str. 310)[6] można raczej oprzeć w tych przypadkach na idei sieci komórkowej dostosowanej do założeń PNN, co niewątpliwie nie jest pożądanym przeznaczeniem PNN. Nie licząc prac teoretycznych większość doświadczeń tego okresu prowadzona jest na pojedynczych neuronach. Niektóre z nich są bardzo nowatorskie (np. postępowe prace z 1997 roku Axela Janke [75], Tima Schoenauera,...), ale nie są w stanie określić dynamiki sieci złożonej z wielu neuronów. Z drugiej strony prace na większych zbiorach neuronów nie wykorzystują globalnych właściwości sieciowych celem przetwarzania informacji. Jako przykład tego typu obiektu przeanalizujmy sieć skonstruowaną przez wybitnych badaczy Horn'a i Opher'a [71] w 1997 roku, składającą się z 150 neuronów (150IAF) o topologii identycznej do założonej w tej pracy. Po określonym czasie symulacji neurony wpadały w sprzężenie zwrotne i momenty odpaleń neuronów, zgodnie z regułą opisaną w rozdziale 4.3.2, zbliżały się do siebie, by po dłuższym czasie doprowadzić do wzajemnej synchronizacji. W przypadku ustawienia stosunkowo dużej ilości wag ujemnych sprzężenia nie miały charakteru dodatniego i efekt synchronizacji nie następował. Zachowania

[5] PRISM http://www.neuralmachines.com/prism/neural.html

[6] sekcja autorstwa Lindblad z 1997 roku

te są znane już z modeli IMPL [24], przy czym sieć 150IAF nie przetwarzała żadnych danych wejściowych! Działo się tak, gdyż nie zapewniono żadnych mechanizmów likwidacji sprzężenia oraz różnicowania neuronów. Oczywiście nie są to elementy proste do wkomponowania w model. W IMPL2 była to odpowiednia praca progu, natomiast w modelach SNN odpowiednie mechanizmy synchronicznego pobudzenia lokalnego (a nie jak 150IAF – globalnego) tworzone podczas pracy sieci opierając się na możliwości użycia funkcyjnych zależności od sąsiedztwa neuronu. 150 IAF nie posiada algorytmu uczenia, ponieważ model nie wychodzi ze sprzężenia (nawet częściowo), a jest to podstawą do opracowania algorytmu uczenia.

Przesuwając się do roku 2000 można stwierdzić, iż prace profesora Davida Horn wraz z współpracownikami (w tym Nir Levy, Eytan Ruppin, Sharon Levanda, Isaac Meilijson, Ofer Hendin, Mishy V. Tsodyks) z ośrodkiem w Tel Aviv'ie stanowią obecnie mocną podstawę do badań sieci składających się pulsujących oraz zintegrowanych neuronów. Tego typu konstrukcje są stosunkowo bliskie modelom SNN z dostępnych w literaturze i internecie pod względem budowy, cech oraz efektów działania, a co najważniejsze – występuje tu również zgodność w metodologii działania. W modelu 150 IAF z 1997 roku w przeciągu ostatnich 4 lat Opher i Horn wraz z zespołem wprowadził wiele zmian. Wzmocniła się też platforma sprzętowa a rozmiar 150 neuronów jest dość swobodnie przekraczany. Analizując jedną z ostatnich prac [86] tego ośrodka można wywnioskować na jakim poziome zostały zrealizowane poszczególne mechanizmy modelu IAF oraz wykazać mocne strony modeli SNN.

Sama dynamika sieci, szczególnie w modelu rozwijanym przez Horn'a i współpracowników (nieco uproszczonych neuronach IAF [14]), była i jest dość poważnym problemem z powodu praktycznego bi-stabilnego stanu ich systemu, tj. asynchronicznie i niezależnie odpalających się neuronów lub pełnej synchronizacji przy wysokiej aktywności neuronów. Tak naprawdę interesującym stanem jest właśnie stan pośredni, tzn. wyzwalanie się pojedynczych lub skończonych serii synchronicznych ciągów odpaleń. Jak wykazano w pracy [14] z 2003 roku, przy założeniu, że neurony są połączone w sposób dość losowy oraz występuje duża ich liczba i sieć jest stosunkowo gęsta (a tak jest w przyrodzie), można przyjąć że wyjście z systemu w dość

małym stopniu zależy od pojedynczego neuronu. Ponieważ wyjście z sieci przy wielu połączeniach zwrotnych jest też praktycznie tym samym, co odpowiednie neurony otrzymują na swoje synapsy, można rozwiązać równanie w których zrównamy wartość wyjścia i wejścia, co jest warunkiem granicznym utrzymywania stabilności sieci. Jak wynika z obliczeń, częstotliwość pracy neuronów nie ma wpływu na stabilność pracy sieci. Jak się można domyślać, jest ona już efektem, a nie warunkiem. Sama stabilność i zdolność do wygenerowania pojedynczych fal synchronicznych pobudzeń zależy natomiast od proporcji synaps pracujących w sprzężeniach zwrotnych, zarówno ujemnych jak i dodatnich, do całej ilości synaps. Oczywiście jest to cenne spostrzeżenie, ale elementem potrzebnym w tym systemie byłby algorytm uczenia zachowujący te proporcje. Jak wiadomo, występują tu bardzo duże kłopoty tych algorytmów, gdyż bardzo często korelują dane pochodzące właśnie z własnych sprzężeń (np. przesuniętych o jakiś wektor w czasie poprzez przejście sygnału przez kilka czy kilkanaście szeregowo pracujących neuronów), co jest zjawiskiem bardzo negatywnym. W tej pracy autorzy nie proponują żadnego algorytmu uczenia. Warto jednak zwrócić uwagę, że symulowana sieć była dość duża, ok. 10-20 razy większa od możliwości SNN, co w dużym zakresie poprawia wyniki statystycznych obserwacji i czyni je jeszcze bardziej realnymi. Z drugiej strony neurony tego modelu zostały jeszcze bardziej uproszczone (w tym również względem SNN) i choć autorzy wyciągają wnioski dla sieci IAF, nie do końca można się z tym zgodzić. Być może wnioski będą prawdziwe również dla IAF, ale w.w. symulacje nie pozwalają tego powiedzieć. Z pewnością tak mocne zredukowanie budowy neuronu i jednocześnie zwiększenie liczby neuronów jest obszarem interesującym do przebadania. Można powiedzieć - jest jakby kolejnym krokiem, bardziej radykalnym, niż uproszczenie klasycznego neuronu do pulsującego, choć nie wiadomo czy słusznym.

Zarówno sieć IAF jak i SNN posiada strukturę połączeń umożliwiającą sprzężenia zwrotne. Neurony jednej jak i drugiej sieci posiadają specjalny mechanizm całkujący sygnał wejściowy w czasie na swoich synapsach, co zapewnia fizyczną bazę dla pamięci krótkoterminowej, liczonej w kilkunastu milisekundach. Ze względu na przyśpieszenie symulacji (oraz zmniejszenie zajętości pamięci) twórcy IAF również (podobnie jak w SNN) używają większych częstotliwości wejściowych i w rezultacie międzyneuronowych niż

występujące naturalnie. „As a result the span of the kernel is somewhat smaller than the experimentally observed ones. In future more realistic neuronal dynamics one should aim for both larger time-span of the kernel and lower sustained firing rates of excitatory neurones, thus getting closer to experimental observations." [86] str. 5. („W rezultacie okres podstawowy jest trochę mniejszy niż obserwowany eksperymentalnie. W przyszłości dla bardziej realistycznej dynamiki neuronów powinien być wydłużony oraz zmniejszona stała częstotliwość odpalania pobudzonych neuronów. W ten sposób zbliżymy się obserwacji eksperymentalnych"). Oszczędność pamięci wynika z następującego faktu: używanie większych częstotliwości umożliwia zakodowanie większej ilości danych jednostce czasu, zatem pamięć krótkoterminowa o krótszym czasie pamiętania (proporcjonalnym do zwiększenia częstotliwości względem naturalnych) ma takie same możliwości. Zabieg ten w sumie doprowadza do bardziej efektywnego wykorzystywania neuronów pod względem przetwarzania informacji, jednakże może powodować inne komplikacje związane z oddalaniem się od naturalnego wzorca, jak np. potrzebę przeskalowania innych stałych występujących w modelu, zmniejszenie tolerancji na uszkodzenia sieci itp. Podstawę algorytmu uczenia obu typów neuronów jest reguła Hebba oraz testowane są jej czasowo zależne modyfikacje. Zmieniając podstawę czasu musimy uwzględnić jej różnice analizując wyniki pracy pamięci krótkoterminowej, ponieważ z definicji pamięć ta jest nierozłącznie związana z upływem czasu. Z drugiej strony zwiększanie częstotliwości celem zwiększenia efektywności sieci nie może postępować w nieskończoność, a nieprzekraczalną granicę wyznacza tu dyskretny krok symulacji. Dla modeli SNN jest on równy 2 ms.

Problem skalowania podstawy czasu nie jest do chwili obecnej rozwiązany i rykoszetem odbija się w innych aspektach sieci np. niezbędnej wielkości sieci do określonego zadania. Bo czy faktycznie natura pozwoliłaby na kodowanie sekwencji na tym większej liczbie neuronów im wolniejsza dana sekwencja by była (jej zmienność)? Poprzez prędkość sekwencji rozumiemy ilość poszczególnych zdarzeń występujących w sekwencji w przyjętej jednostce czasu. Zdarzeniem może być np. zmiana tonu czy uderzenie w kolejny klawisz instrumentu klawiszowego. Prawdą jest też, że wolniejsze sekwencje są trudniejsze do zapamiętania (potocznie mówi się, że szybkie rytmy łatwiej

„wpadają w ucho"). Ptaki i podobnie rozwinięte zwierzęta wydając dźwięki nie zmieniają nigdy tempa (z kilkoma wyjątkami). Być może zapamiętywanie, porównywanie i odtwarzanie z pamięci sekwencji o różnej podstawie taktującej wymaga większego zaangażowania naturalnej sieci neuronowej, a w przypadku prostych sieci jest to niewykonalne. Pozostaje zatem konstruowanie hipotez, jak to natura realizuje.

Zarówno przedstawione modele sieci IAF jak i SNN mają problemy z zapamiętaniem kilku zapisów w tym samym miejscu sieci, ponieważ zbyt często dochodzi do uśrednienia wzorców w pamięci sieci. Jest to tzw. problem **multipamięci** w PNN, nad którym prowadzone są obecnie intensywne prace.

Rezultatem pracy sieci IAF jak i SNN, jak do tej pory, jest wygenerowanie dynamicznego atraktora w przestrzeni pobudzeń neuronów. Jak już zostało wspomniane w poprzednim rozdziale, atraktor nie jest dobrym reprezentantem wyniku przetwarzania, jednakże przeprowadzone doświadczenia symulacyjne SNN w latach 1996-1999 bazowały właśnie na takim podejściu. Natomiast nawet szczegółowa analiza atraktora (stopień powtarzalności określonych części lub całych wektorów, okres tych powtórzeń, tendencje zmian, uśrednianie statystyczne...) dostarcza ogromnej ilości informacji na temat procesów przebiegających wewnątrz sieci. Sam fakt skupienia się na analizie atraktora w tym czasie, jego powstawaniu, stabilności itp. jest pierwszą jaskółką, że on sam nie jest efektem przetwarzania. Dodatkowo obserwacja matrycy wag uzupełnia wiedzę o przetwarzaniu sieci PNN.

W tym momencie zauważamy wyraźne różnice w złożoności tej matrycy sieci IAF Horn'a z 2000 r. (zwanej dalej IAF) oraz SNN. O ile IAF jest w stanie wygenerować podczas pracy przykładowe pięć klastrów neuronów ([86] str.9) (IMPL2 miało już te możliwości, doświadczenie rozdział 5.1.1.1), o tyle SNN może generować ich dowolnie wiele nie zmniejszając szans osiągnięcia celu ani nie powiększając złożoności obliczeniowej. Na tym praktycznie kończy się moc algorytmu uczenia IAF. Oznacza to w skali której operujemy, iż wzorcem dla IAF jest sygnał sinusoidalny o stałej częstotliwości (tutaj z przedziału 100-500 Hz). Zawartość informacji w takim sygnale jest oczywiście niewielka. Oczywiście istnieje możliwość ustawienia wag przed rozpoczęciem pracy sieci IAF. Wtedy mamy sposobność otrzymania dowolnie skomplikowanej matrycy

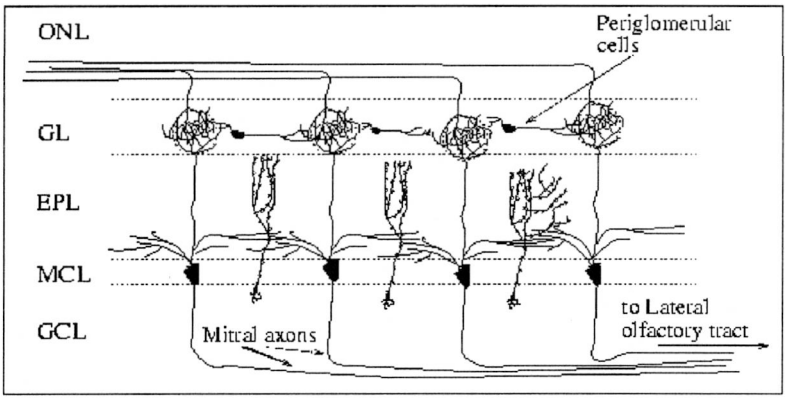
wag oraz zaobserwowania pracy sieci z tą matrycą. Niestety nie możemy w tym przypadku wykonać sedna doświadczenia, tzn. uwolnienia modyfikacji wag zgodnie z przyjętym algorytmem uczenia, co potwierdza autor: „In fact, if we allow synaptic learning to occur under the same conditions listed above, the learning process will destroy the segmented synaptic matrix structure and merge the two cell-assemblies." ([86] str. 17) (W praktyce, gdy zezwolimy na uczenie synaptyczne zgodnie z przyjętymi regułami, proces uczenia zniszczy posegmentowaną strukturę oraz połączy obie grupy komórek").

W modelach SNN praca wag nie jest blokowana, a co istotniejsze – tak względnie skomplikowane matryce wag powstały w wyniku działania sieci, a w tym algorytmu uczącego, również bazującego na regule Hebba. Oczywiście opracowanie takiego algorytmu, bazującego na regule Hebba, nie było trywialne i wymagało dość pokaźnej serii doświadczeń, omówionych w tej pracy. Algorytm uczenia SNN potrafi odwzorować dowolnie długi wzorzec wejściowy w matrycy wag neuronów, oczywiście pod warunkiem spełnienia wymagań pamięciowych (np. wzorzec 500 wektorów 12 elementowych jest w stanie reprezentować 1 sekundę dźwięku, czyli przeciętny wyraz). Nie potrafi tego wykonać IAF, gdzie jak można przypuszczać na podstawie doświadczeń z modelami IMPL, długość wzorca nie powinna przekraczać 5-6 wektorów, a doświadczenia z dłuższymi wzorcami w aż ok. 96%-98% kończą się niepowodzeniami.

Kolejnym dość istotnym elementem SNN względem IAF jest traktowanie wzorca jako danych pojawiających się w czasie pracy sieci. IAF dość ściśle trzyma się linii wytyczonej przez sieć Hopfield'a, pomimo zastosowania jako podstawowych elementów neuronów pulsujących. Kierunek ten zapewnia komfort związany ze zbieżnością przetwarzania itp. natomiast ogranicza siłę neuronów pulsujących, które z samej zasady swojego działania wymagają pracy w czasie. Pomimo, iż Horn i jego współpracownicy przełamali w ostatnich pracach opory przed dopuszczeniem niesymetryczności w matrycy wag, o tyle wzorzec o określonej wielkości musi być podany w całości w jednym momencie, ponieważ tylko taki mechanizm ta sieć posiada. Jeśli chodzi o wzorzec związany z sekwencyjnym podawaniem do sieci, to w przypadku IAF[86] jest to ton o ustalonej częstotliwości z zakresu 100-500 Hz, ew. po pewnych

przeróbkach IAF model mógłby przetwarzać prawdopodobnie kilkuwektorowe wzorce (np. 5 wektorowe).

W tak złożonych systemach, oraz o tak dużym stopniu dynamiki, nie tylko wymienione wyżej cechy mają znaczenie. Analizując wcześniejsze prace Horn'a oraz badaczy z nim współpracujących [71] możemy dowiedzieć się więcej szczegółów na temat drugoplanowych cech modeli oraz pewnych założeń konstrukcyjnych. W pracy „Associative Memory and Segmentation in an Oscillatory Neural Model of the Olfactory Bulb" [60] jest poruszony problem budowy systemu pod kątem wykorzystania go jako pamięci asocjacyjnej. Jest to dość wczesna praca i nie możemy powiedzieć, że podstawowym elementem jest tu już zintegrowany neuron pulsujący pomimo wprowadzenia dodatkowych cech. Zarówno mechanizm działania jak i uczenia bazuje na założeniach sieci Hopfielda z 1982 r [64]. Sieć skonstruowano z dwóch istotnych warstw (oraz trzeciej pomocniczej), pierwszej dokonującej wstępnego porównania, oraz drugiej działającej jako mechanizm klasyfikujący. Sieć o wielkości ok. 1000 neuronów umożliwiała zapamiętanie tylko dwóch wzorców. Elementem wartym szczególnego podkreślenia tej pracy było traktowanie wyjścia systemu jako atraktora, w przeciwieństwie do oczekiwania stanu ustalonego. Był to pierwszy duży krok jak owe czasy i niezbędny przy pracy z pulsującymi sieciami neuronowymi nie dążącymi do stanu ustalonego. Bardzo interesujące podejście do tematu przetwarzania pulsacyjnego zostało zaprezentowane w pracy „Temporal Coding in an Olfactory Oscillatory Model"[144]. Pomimo iż większość zagadnień jest przedyskutowana tylko na



Rys. 7.1 – Model sieci o warstwowej strukturze ziarnistej („Olfactory Bulb layered architecture" [60]). Zaznaczone warstwy: ONL nerwy węchowe, GL kępki nerwowe, EPL sploty zewnętrzne, MCL komórki dwudzielne, GCL komórki ziarniste.

podstawie rozważań teoretycznych, nie traktuje się w tej pracy sieci neuronowej jako rodzaju pamięci, ale jako pewne przekształcenie sygnału wejściowego w wyjściowy, jako filtr o określonych możliwościach. Jest to zupełnie odmienne podejście do tematu niż w poprzednio wymienionej pracy, i jak sam autor wskazuje – dość bliskie wzorcom biologicznym. Brak wykorzystywania pamięci krótkoterminowej podczas przetwarzania oraz brak spójności z poprzednimi pracami uniemożliwia zastosowania wypracowanych wcześniej mechanizmów, co z kolei znacznie ogranicza dyskutowane potencjalne możliwości tego rozwiązania. Jak wskazują modele SNN jest to produktywne podejście do zagadnienia, pod warunkiem ujednolicenia teorii pamięci oraz jednoczesnego przetwarzania z nie nadzorowanym uczeniem.

Istotna jest też praca z 2000 roku Davida Horn'a i jego współpracowników pod tytułem: „Neuronal Regulation and Hebbian Learning" [50] w której autor stwierdza między innymi: „Hebbian mechanisms per-se fail to provide robust and effective learning, both in supervised and unsupervised scenarios", „Hebbian learning leads to poor associative memory capacity that does not grow with the size of the network" („Reguła Hebba nie dostarcza silnego i efektywnego uczenia zarówno w uczeniu z nauczycielem jak i bez". „Reguła Hebba prowadzi do słabej pamięci autoasocjacyjnej, której pojemność nie rośnie wraz ze wzrostem sieci neuronowej"). Pomimo wielkiego autorytetu pracy Hebb'a zgadzam się z tą opinią, o czym świadczy odrzucenie czystej zasady Hebb'a począwszy od modelu SNN11a [1999] poprzez **wprowadzenie normalizacji wag synaptycznych**. Horn również proponuje identyczne posunięcie, a jako jego zalety wymienia:

- zwiększenie pojemności pamięci
- możliwość powstrzymywania dodatniej pętli sprzężenia wzrostu wag uczenia regułą Hebba
- możliwość powstawania przypadkowych połączeń międzyneuronowych, czego nie zapewniał paradygmat Hebba!
- modelowanie degradacji pamięci powodowanej procesami metabolitycznymi
- większą zgodnością z wzorcem biologicznym (np. wzrost wagi jest ograniczony)

W tej pracy autor jednak nie podaje jak użyć tego mechanizmu w najbardziej przydatnym zestawieniu - w środowisku, w którym neuron ma dostęp tylko do

swojego otoczenia i do informacji w nim zawartych. Dokładniej, autor nie widzi szybkiej możliwości adaptacji: „These observations show that it is difficult to implement effective learning with local synapse-specific learning rules" („Obserwacje pokazują, że trudnym do zaimplementowania jest efektywny algorytm uczenia bazujący na regułach uczenia uwzględniających lokalną specyfikację synapsy"). Efektywne uczenie wymaga uwzględnienia w omawianym algorytmie liczby odpalanych neuronów w sieci w danym momencie, co jest globalnym parametrem sieci, a nie lokalnym. Zatem pojedynczy neuron nie posiada dostępu do tego parametru. Jak wynika z prac nad modelami SNN11, zaimplementowanie algorytmu uczenia spełniającego te ostre warunki i bazującego wyłącznie na lokalnych danych jest możliwe (lokalna reguła uczenia).

Pewną konsekwencją sceptycznego podejścia do możliwości adaptacji algorytmu uczenia typu czasu rzeczywistego w sieciach opartych o neurony IAF stało się przyczyną położenia nacisku przez D. Horna na inne obszary badań tych modeli, w tym szczególności ich dynamiki oraz pojemności, czego dowodem jest praca [15]. W tym modelu synapsy nie modyfikują się zarówno w zakresie kierunku połączeń jak i siły przewodzenia. Wzmocnienie połączenia pomiędzy neuronami, o ile taka konieczność wystąpi, realizowana jest poprzez zwielokrotnienie synapsy. Może się tak zdarzyć, gdy dany neuron uczestniczy w odtwarzaniu nie tylko jednego wzorca zapamiętanego w sieci. Wprowadzenie wzorców do sieci odbywa się jednorazowo przed przystąpieniem sieci do pracy poprzez obliczenie wartości synaps, podobnie jak w przypadku sieci Hopfielda. Dość istotna modyfikacją jest ustalenie wartości wag (tym bardziej, że wszystkie wagi maja tę samą wartość w tym modelu), jako funkcji ilości synaps pobudzających, co ma diametralny wpływ na dynamikę pracy sieci. Przyjęto tutaj, że waga jest funkcją pierwiastkową o argumencie ilości ww. synaps. Założeniem ustawienia synaps jest reguła, że neuron nie uczestniczący w przetwarzaniu aktualnego wzorca powinien otrzymywać zwykle aktywację podprogową, z kilkoma wyjątkami związanymi z pracą neuronów w tle. Autor definiuje przydatne pojęcie tzw. **Asynchronus State** (AS) czyli stan asynchroniczny sieci, w którym neurony odpalają się stosunkowo rzadko i dość losowo w porównaniu do stanu synchronicznego (Synchronus State), który jest traktowany jako zjawisko niekorzystne, jeśli czas jego trwania przekroczy

zadany okres krytyczny - przyjmowany tu jako 100 do 300 ms. Podejście takie jest jak najbardziej uzasadnione, gdyż przy omawianych modelach i wielkościach sieci IAF pomiędzy 1000 a 100 000 neuronów stan synchronicznego pobudzenia trwający dużej prawdopodobnie nigdy by się nie zakończył w sposób samoczynny. Najciekawsze stany natomiast to są stany przejściowe ze stanu asynchronicznego do synchronicznego, a zjawiska temu towarzyszące mówią dużo o naturze pamięci. Jeśli sieć pracująca w trybie asynchronicznym posiada wyuczony wzorzec (tutaj wpisany na stałe) to przy rozpoznaniu wzorca wejściowego przez kilkaset milisekund pracuje w stanie synchronicznym, zwiększając jednocześnie średnią częstotliwość odpaleń neuronów w sieci. Zjawiska te były omówione podczas analizy zjawisk modeli SNN tej pracy. W szczególności wartym podkreślenia jest fakt, że SNN zapamiętuje wzorzec podczas pracy i nauki w czasie rzeczywistym, czemu towarzyszy równoległe i systematyczne wytwarzanie w sieci coraz mocniejszych (pochłaniających więcej neuronów i trwających dłużej) skończonych stanów synchronicznych, czego omawiana praca D.Horna nie zawiera. Zgodnie natomiast z autorem uważam, że podejście zaprezentowane w [33], polegające na odzyskiwaniu zawartości pamięci w trybie synchronicznego stałego pobudzenia jest możliwe, aczkolwiek nie jest to kierunek rozwoju, który byłby obiecujący.

Badając dynamikę sieci autor pokusił się nie tyko o zapewnienie warunku na strukturę oraz ilość synaps, zabraniającego powstawanie samoistne stanu synchronicznego, ale również skonstruował dodatkowy balans w postaci dodatkowej warstwy neuronów hamujących (nazwanych warstwą cieni). Tak skonstruowaną sieć, o dodatkowej ilości sygnałów hamujących pojawiających się jednocześnie ze wzmożoną aktywnością neuronów pobudzających, nazwano podwójnie zbalansowaną siecią (Double Balanced Network). Oczywiście warstwa cieni w aktywny sposób zapobiega powstawaniu stanu synchronicznego, jednakże podobnie jak poprzednie warunki, nie jest pewnym rozwiązaniem, a dodatkowo konstrukcja jest coraz odleglejsza od wzorców naturalnych, co może wprowadzić badania w ślepą uliczkę. Warstwa cieni jest przydatna głównie przy bardzo mocnym, choć krótkim (np. 5 ms), pobudzeniu wejścia sieci.

Autor systematyzuje jednocześnie dwa podejścia przeplatające się w konstrukcjach IAF, mianowicie SFC oraz HCA (tj. synfire chains (SFC) oraz Hebbian cell assemblies (HCA)) oraz przydatne słownictwo, np. połączenia synaptyczne „do przodu" w modelach SFC zostały nazwane łańcuchami, a sama propagująca się tą „doliną" grupa pobudzeń – „synfire wave" lub po porostu falą. Pomimo, że HCA wykazują się większą gęstością zapamiętywanej informacji, SFC stają się coraz częstszym obiektem badań, co niewątpliwie świadczy o pokładanych w nich nadziejach. Wszystkie modele SNN tej pracy wspierały strukturę SFC podczas symulacji, choć mają potencjalna możliwość pracy zarówno w SFC, jak i HCA.

Dość ciekawą analizę, wykazującą dużą moc zintegrowanego, pulsującego neuronu, przeprowadzono w pracy [118] z 1999 roku. Analiza została przeprowadzona, zgodnie z tytułem, pod kątem opracowania algorytmu uczenia, wykorzystującego fakt zawarty w twierdzeniu 4.2 i rozwinięty w 4.3 tej pracy, a mianowicie, że problem integralności dla neuronu z binarnymi opóźnieniami oraz ustalonymi wagami jest NP-zupełny. W szczególe, jeżeli pojedynczy neuron posiada przynajmniej opóźnienie jednostkowe, z odpowiedniej liczby neuronów możemy skonstruować odpowiednie do zadania opóźnienie i zsynchronizowanie dwóch lub większej ilości przesuniętych w czasie ciągów danych celem ich dalszej obróbki. Autor nie proponuje jednak żadnego algorytmu uczenia. Stwierdza natomiast, że jest to duże wyzwanie, a szczególnie trudne byłoby udowodnienie poprawności takiego algorytmu. Tezy te jak najbardziej potwierdzają doświadczenia tej pracy przeprowadzone na symulatorze IMPL! Algorytm uczenia w IMPL, choć niedoskonały, pokazał że jest w stanie skorelować i przeprowadzić operacje logiczne na danych pojawiających się na wejściach systemu z przesunięciem czasowym. Dodatkowo IMPL posiada sprzężenia zwrotne, dzięki czemu może się synchronizować z sekwencją powtarzalną, natomiast przykłady tej pracy [118] znajdujące się w rozdziale trzecim świadczą, że domyślnie założono jednokierunkowość przepływu danych w sieci. Interesująca i warta podkreślenia jest natomiast kolejna próba zdefiniowania, co w neuronie pulsującym jest cyfrowe, a co analogowe. Oczywiście, w większości jest to kwestia interpretacji, czy operujemy na pojedynczych impulsach, czy na częstotliwościach, które te impulsy występujące w określonych odstępach czasu tworzą. W tym przypadku

potrzebujemy tez funkcji przekształcającej te częstotliwości w abstrakty, które ona reprezentuje. Możemy zatem definiować neurony o analogowych wejściach i cyfrowym wyjściu albo odwrotnie, zupełnie binarne itp.

Ważnym elementem neuronu pulsującego, ściśle związanym z algorytmem uczenia i w rezultacie tworzącym pamięć krótko- i długoterminową., którego nie sposób nie potraktować specjalnie, jest pojedyncza synapsa – jej praca, parametry transmisji, gromadzone dane statystyczne dotyczące jej historii, modyfikacje wagi itd. W modelach SNN występują teoretycznie 3 rodzaje synaps: synapsa pobudzająca, synapsa hamująca oraz synapsa hamowania presynaptycznego. W tej ostatniej występują 2 aksony i jeden dendryt. Skuteczne przetwarzanie danych udało się zaobserwować jedynie na pierwszej z wymienionych synaps (przy założeniu struktury PNN oraz stałego kształtu impulsu wysyłanego przez neuryt). Identyczną ilość typów synaps (nazwane *F1,F2* i *F3*) proponują Maass oraz Natschlager w pracy [132]. Proponowane jest jednak inne sparametryzowanie synapsy - poprzez trzy zmienne charakteryzujące jej dynamikę, nazwanymi *U,F,D* (prawdopodobnie inspiracją tego podejścia była praca [126]). Zmienne te przesądzają o typie danej synapsy. Zmienne te są ciągłe, co oznacza, że istnieją różne typy synaps pośrednich łączących w różnym stopniu cechy synaps skrajnych. Podejście takie jest o tyle ciekawe, iż definiuje nam stosunkowo różne synapsy, wszystkie pracujące w polaryzacji dodatniej. Żadna z nich nie dokonuje hamowania innych sygnałów czy bezpośredniego ujemnego wpływu na potencjał wewnątrzkomórkowy neuronu postsynaptycznego. Jedyna synapsa z w.w. trzech, nazwana depresyjną (*F2*), najbardziej przyjmująca rolę synapsy hamującej tak naprawdę ją nie jest. Jej działanie można polega na zmniejszeniu czułości przez określony czas po otrzymaniu pobudzenia, ale proces ten dotyczy tylko impulsów docierających do tej właśnie synapsy. Nie jest to zatem synapsa hamująca. Pamiętając o kłopotach algorytmów uczenia synaps hamujących, powstaje tu nowa przestrzeń do zagospodarowania przez algorytmy uczenia synaps pobudzających. Pomimo mnogości takich algorytmów autor nie przystosowuje żadnego z nich do tej struktury. Podaje jedynie metody cząstkowych optymalizacji synaps do zdefiniowanych ciągów wejściowych charakterystycznych dla „skrajnych" synaps. Można przypuszczać, że algorytm uczenia, o ile powstanie, będzie modyfikował w synapsie tylko

wymienione parametry *U,F,D.* Przyglądając się tym parametrom możemy uznać, że *U* jest odpowiednikiem wagi, natomiast *F,D* – są wyrażone w ms i definiują dynamikę synapsy, jej reakcje na skupioną bądź rozproszoną w czasie serię bodźców. Autor natomiast bardzo dokładnie opracował ciągi optymalnie pobudzające synapsę przy ustalonych parametrach *U,D,F.* Ciągi te zostały wyliczone przy pomocy metod optymalizacyjnych i można je uznać za swoiste klucze do synaps, tzn. np. 3 różne dendryty odbierające sygnał z jednego neurytu mogą być w miarę selektywnie pobudzane. Praca też jest zgodna z przesłankami biologicznymi, gdyż obserwujemy w układzie nerwowym różne modulacje i konfiguracje impulsów przy zachowaniu określonej częstotliwości (w granicach błędu oraz przy wyborze odpowiednio dużego okna czasowego). Wydaje się zatem prawdopodobne, że informacja może być również kodowana w ten sposób.

Omawiając synapsę należy ująć kwestię jej znaczenia jako pamięci tymczasowej, reprezentowanej przez stan ładunku postsynaptycznego. Sekwencja wejściowa synapsy jest transformowana w wartość ładunku uwzględniając parametry danej synapsy np. jak poprzednio zaproponowane *U,F,D.* Analiza zależności odwrotnej tzn. określania ciągu wejściowego przy znanych parametrach synapsy jest przeprowadzona w pracy [111]. Z twierdzenia *2.1* tej pracy wynika, że nie można jednoznacznie określić sekwencji wejściowej na podstawie wartości potencjału oraz parametrów synapsy, nawet jeśli znamy ilość impulsów $k>=3$. Przy $k<3$ sytuacja jest znacznie prostsza, ale jednocześnie mało interesująca. Potencjał postsynaptyczny jest zależny od impulsów docierających do synapsy w ostatnich kilkuset milisekundach [111], a jest ich zwykle co najmniej kilkanaście. Fakt nieistnienia funkcji przekształcenia odwrotnego nie przesądza oczywiście o braku możliwości użycia potencjałów postsynaptycznych jako pamięci krótkoterminowej. Jak wskazują doświadczenia – fakt ten jest jak najbardziej potwierdzalny. Należy jednak pamiętać o tym twierdzeniu, ponieważ rzuca ono pewne światło na strukturę pamięci krótkoterminowej.

W pracy [141] przedstawiona jest koncepcja wprowadzenia sigmoidalnej funkcji przepuszczalności błony komórkowej synapsy w zależności od przepływającego przez nią ładunku. Całość dotyczy synapsy neuronu pulsującego, co dodatkowo komplikuje sytuację i rodzi pytanie, czy

przypadkiem nie została wprowadzona nieliniowość przetwarzania w neuronie podwójnie, gdyż neuron pulsujący już takową posiada, ze względu na symulację potencjałów postsynaptycznych. W modelu tym można zaobserwować jeden element zupełnie nietestowany w SNN. Synapsy pobudzane zarówno przed, jak i krótko po momencie aktywacji (+/- 50ms) neuronu są silnie modyfikowane przez algorytm uczenia. Trudno ocenić korzyści z tego zjawiska, jednakże może to być właściwość przydatna. W modelach SNN tylko synapsy będące przyczyną wywołania impulsu postsynaptycznego silnie były modyfikowane wykonaniem procedury uczenia. Zakładając prawidłowe działanie synapsy i jej gromadzenie ładunku, właściwość ta może przyczynić się do kojarzenia sygnałów, które co prawda nie dotarły na czas do danej synapsy, ale są związane z daną klasą.

Przedstawiony powyżej przegląd literatury dotyczy jedynie prac najbliżej związanych z tematem pamięci w pulsujących sieciach neuronowych, gdyż specjalizacja w zakresie użycia sygnałów płynących wstecznie jest dość wąska. Jednocześnie pokazuje wiele szczegółowych elementów, które w przyszłości mogą tworzyć z mniejszym lub większym powodzeniem sprawny system przetwarzania informacji - system o globalnych efektach synergicznych neuronów oraz zawierający w sobie różne formy reprezentacji swojego środowiska zewnętrznego (pamięć długo- i krótko- terminowa) z możliwością przetwarzania zapisanych w swojej pamięci informacji. Nie jest obecnie możliwe określenie wagi każdego z elementów lub przeprowadzenia pełnej ich klasyfikacji, ale można pokusić się na podstawie tej pracy o następujący ***podział opisanych pulsujących sieci neuronowych względem wykorzystania potencjałów postsynaptycznych:***

- sieci zawierające neurony bez symulacji ładunków postsynaptycznych
- sieci zawierające neurony z symulacją ładunków postsynaptycznych
    - użytą do pracy neuronu
    - użytą do pracy neuronu oraz w algorytmie uczenia
        - metody progowe
        - metody proporcjonalne

Potencjał postsynaptyczny jako pewna forma przedstawienia historii danej synapsy jest możliwy do zapisania za pomocą kilku bitów informacji, a jego wykorzystanie może diametralnie zmienić pracę sieci. Może on być użyty

podczas przetwarzania danych przez neuron, przy obliczaniu globalnego pobudzenia , jak również podczas wyliczania poprawek wag. Ponieważ ładunek postsynaptyczny, będący odpowiedzią synapsy na pobudzenie neuronu, ma wartość nieujemną (nie uwzględniając procesów podczas refrakcji bezwzględnej i przy ustalonym poziome potencjału „0") wynikają z tego pewne komplikacje. Stosując mechanizmy uczenia bazujące na regule Hebb'a - brakuje procesu przeciwnego do zwiększania wag. Pierwszym rozwiązaniem jest przyjęcie określonej wartości ładunku jako wartości progowej synapsy. Ładunki nie przekraczające tej wartości mogą być traktowane jako "zero" w myśl reguły Hebba. Rozwiązanie to może operować również na wagach ujemnych, choć efektywne wykorzystanie wag ujemnych nie zostało eksperymentalnie potwierdzone w tej pracy. Alternatywną metodą jest modyfikowanie proporcjonalne wag do ich ładunków postsynaptycznych. Jest to zgodne z regułą Hebb'a, ale znany problem zmniejszania wag pozostaje [8]. Dość dobrym podejściem jest normalizacja wektora wag bezpośrednio po poprawkach wprowadzonych algorytmem uczenia. Na podstawie tej pracy ***metody zmniejszania wag i/lub umożliwiające powstawanie wag ujemnych możemy podzielić na:***

- modyfikacje reguły Hebba
- normalizacja wektora wag po każdej iteracji algorytmu uczenia

Pierwsze podejście jest o tyle lepsze o tyle, iż umożliwia powstanie wag ujemnych. Jednakże w połączeniu z bardzo istotnym mechanizmem samoregulacji neuronu opracowanym w modelach IMPL oraz SNN nazwanym PRA[7], może prowadzić do powstawania neuronów o wartościach ujemnych progu. Neurony takie obecnie bardzo szybko uciekają spod kontroli algorytmu uczenia i niszczą efekty synergii (we wszystkich zbadanych modelach, w których takie neurony powstały).

Drugie podejście, polegające na normalizacji wektora wag, co prawda nie wspiera powstania wag ujemnych, jednak jak pokazują modele SNN11 oraz najnowsze publikacje (np. [50]) – posiada wiele niezbitych atutów oraz staje się coraz bardziej popularny. Zastosowanie tego mechanizmu w modelu SNN11 i wszystkich następnych jest niezbitym i potwierdzonym dowodem na postępowość myśli zawartej w skonstruowanych modeli SNN. Praca [50] jest

[7] Mechanizm PRA został omówiony w rozdziale 5.3.3.

datowana dopiero na wrzesień 2000 roku, gdzie modele SNN posiadały tę cechę od ponad 2 lat.

7. Podsumowanie

Przyroda w swym geniuszu często zadziwiała ludzi oraz dawała i daje inspiracje do naśladowania. Jak wynika z przeprowadzonych doświadczeń, modelowanie struktur znanych z fizjologii może dostarczyć nie tylko ciekawych faktów do symulacji, ale i rzucać pewne światło na sam przedmiot modelowania. Przykładowo, pośrednio z przeprowadzonych symulacji wynika, że powstawanie naturalnego mózgu musi przebiegać równolegle z jego pracą w takim zakresie, w jakim on istnieje. W przeciwnym razie uległby krótszemu, dłuższemu lub nieskończonemu wzbudzeniu. W każdym z tych przypadków skutki dla organizmu mogłyby być katastrofalne. Wnioski takie można wyciągnąć na podstawie różnych form synergii w pracy neuronów zaprezentowanych w modelach IMPL oraz SNN. Nie są one pozbawione mankamentów, jednak już niektóre właściwości, jak kumulowanie informacji w czasie w sieci, równoległość przetwarzania, niezawodność i wysokie rozproszenie przetwarzania wskazuje na potrzebę dogłębnego przebadania tych zjawisk. Właściwości te wskazują perspektywę w budowaniu tego typu układów elektronicznych o niskich kosztach i wysokiej szybkości pracy.

Rozważając właściwości adaptacyjne należy zauważyć, że maszyny potrafiące odpowiednio się zachowywać w różnych środowiskach można budować wykorzystując różne podejścia do sztucznej inteligencji. Nabywanie wiedzy w systemach ekspertowych jest bardzo trudne, inaczej niż w przypadku sieci neuronowych, ponieważ konstruowanie bazy wiedzy jest niezwykle żmudne nawet dla wysokiej klasy specjalistów. Maszyna na tym miejscu radzi sobie jeszcze gorzej. Sieć neuronowa potrafi natomiast budować reprezentację środowiska, nawet z pozornie sprzecznych danych, posiada również zdolności do generalizacji, co można zaobserwować w modelach SNN. Cechy te leżą bowiem już w naturze struktur neuronowych. Jednakże obecnie technika neuronowa z różnych przyczyn nie doprowadziła do zbudowanie takiej sieci neuronowej, która swobodnie wykształciłaby w procesie uczenia i operowałaby podczas pracy na abstraktach, które powszechnie znamy z wiedzy werbalnej.

Dużą przewagą sieci neuronowych względem klasycznych algorytmów jest równoległość przetwarzania. Jest ona omówiona na przykładzie kilku sieci w rozdziale 5.1.1.1. Przetwarzanie jest niemal perfekcyjnie równoległe. Zwiększa to znacznie możliwości obliczeniowe przy zastosowaniu tej samej technologii wykonania. Warto zauważyć, że neurony układu nerwowego człowieka pracują zwykle z częstotliwością do 500 Hz (max. 1000 Hz), a najszybsze obecne komputery często przegrywają w rozmaitych testach. Na niezwykłą równoległość pracy naturalnych sieci neuronowych wskazuje fakt, że niektóre reakcje układu nerwowego są bardzo szybkie (np. impuls jest w stanie pokonać drogę równą 10 neuronom), a w odpowiedzi są uwzględnione wyniki z wysokich abstrakcji np. z analizy figur przestrzennych, zasady etyki itp. Z dosyć dużym zapasem można ocenić, że przyśpieszenie z powodu równoległości przetwarzania będzie bardzo duże, szczególnie w procesach rozpoznawania, widzenia przestrzennego itp.

Również swą naturalną rozproszoność przetwarzania naturalne sieci neuronowe wykorzystały do poprawienia swojej niezawodności. Systemy scentralizowane posiadają wiele zalet, jednak ogranicza je moc jednostki centralnej. Jeżeli ma ona właściwości częściowo równoległe, to trudno jest to wykorzystać do określonego zadania. Poza tym jednostka przetwarzająca jest wrażliwa na błędy i uszkodzenia, które często doprowadzają do przerwania pracy całego systemu. Przetwarzanie w sieciach neuronowych jest wyjątkowo wysoko rozproszone, a odporność na uszkodzenia leży także w ich naturze, ponieważ elementy podstawowe (neurony) są wysoce zawodne. Potwierdzają to badania modeli ciągłych, w przeciwieństwie do binarnych.

Przetwarzanie danych jest uwarunkowane procesem uczenia. W sztucznych sieciach bardzo często najpierw sieć uczy się, a po osiągnięciu określonego poziomu nadaje się do przetwarzania. Inaczej wygląda to w przypadku sieci naturalnych. Procesy uczenia oraz pracy przebiegają równolegle oraz oddziaływają na siebie, choć jak wiemy w różnych proporcjach w różnych okresach rozwoju organizmu. Bez cechy równoległości uczenia i przetwarzania już model IMPL2 nie wykazywałby żadnych efektów synergii neuronów, co jest niezbędnym elementem wytworzenia pamięci. Jest to jedna z ważniejszych cech naturalnych sieci, której zamodelowanie daje szansę na zbliżenie się ku naturze. Procesy uczenia i pracy powinny przebiegać w

modelu równolegle, aczkolwiek z różną intensywnością. Intensywność uczenia może być wyznaczona jako funkcja przystosowania do środowiska, co było wstępnie testowane w modelu IMPL5A. Równie ważna jest rozproszoność procesu uczenia naturalnej sieci. Wpływa to korzystnie na możliwość implementacji sprzętowej, oraz na zgodność z przyjętymi założeniami. Jednym z ważniejszych założeń pracy było niewykorzystywanie centralnych mechanizmów uczenia. W pracy zostało pokazane, że jest możliwe zbudowanie algorytmu uczenia spełniającego te założenia.

Naturalna sieć neuronowa wydaje się być najlepszym przykładem do naśladowania, zwłaszcza jeśli mamy perspektywę budowania układów scalonych tego typu. Modele IMPL i SNN są również konstruowane pod tym kątem. Do ostatecznej weryfikacji działania pozytywnie ocenionych modeli wymagane będzie użycie wielkiej liczby neuronów. Wiąże się z tym długi czas symulacji. Jedną z możliwości jest sprawdzanie układu w postaci elektronicznej. Założenia projektu mają za zadanie umożliwić przeniesienie modelu do schematu elektronicznego możliwie niskim kosztem. Zastosowanie mechanizmów odporności na błędy w przetwarzaniu umożliwia zwiększenie skali integracji (nie jest wymagana niezawodność podstawowych elementów przetwarzających) oraz umożliwi w przyszłości zastosowanie najnowszych choć zawodnych technologii (może np. nanokomputerów, opartych na ułożeniu przestrzennym określonych atomów).

Praca jest krokiem w kierunku maszyn z pamięcią zrealizowanych w technice neuronowej, wzorowanych na przesłankach z psychologii fizjologicznej. Nie daje ona ostatecznej recepty na taką maszynę, ale wskazuje jak mogłaby ona funkcjonować (RTCWSS), jak mogą być reprezentowane dane pochodzące z środowiska w takiej maszynie np. maszynie ciekłych stanów. Zapożyczając z medycyny takie terminy jak pamięć krótko- i długoterminowa z dosyć dobrym rezultatem dają się one dopasować do określonych zjawisk w symulowanych sieciach neuronowych. Z pewnością modele są jeszcze dalekie od wzorca biologicznego, ale niektóre cechy pamięci krótkoterminowej, jak ulotność i zależność od pamięci długoterminowej, wydają się zgadzać z faktami fizjologicznymi.

W pracy można znaleźć modele sieci neuronowych charakteryzujące się występowaniem wysokiej równoległości procesów uczenia oraz przetwarzania,

modele zawierające pamięć krótko- jak i długoterminową oraz rozproszone metody uczenia. W żadnym z modeli nie występują elementy szczególnie kosztowne w realizacji sprzętowej.

Ostatnie modele serii IMPL nie były wyłącznie obiektami badań, ale nakreśliły wytyczne potomnych modeli SNN, które mogą zawierać znacznie większe ilości neuronów oraz zawierają mechanizmy skalujące. Do szczególnie interesujących wskazówek można zaliczyć sprawdzenie dwukierunkowości synaps oraz dynamiki pracy synapsy, szczególnie przy synapsach hamowania presynaptycznego, określenie metody uczenia oraz formy reprezentowania informacji w sieci neuronowej.

Modele SNN okazały się dość trudnymi do zestrojenia, jednakże po serii wielu prób udało się zorganizować neurony używając dość precyzyjnie działającego mechanizmu PRA, stanowiącym jedną z najmocniejszych stron tych modeli. SNN można obecnie stosować w aplikacjach typu przetwarzanie sygnałów, dźwięku oraz innych danych których natura jest zmienna w czasie. Jednocześnie podczas wprowadzania sygnału do sieci jest już możliwość odczytania odpowiedzi częściowej przed zakończeniem strumienia danych, ponieważ sieć ta pracuje tym samym czasie i nie wymaga kompletności danych do rozpoczęcia procesu przetwarzania. Jest to bardzo istotny krok względem innych podejść, gdzie problemy segmentacji danych, nie występujące w modelach SNN, często skupiały w sobie ponad połowę procesu przetwarzania np. w procesach klasyfikacji.

Cel pracy został osiągnięty. Modele zawierają jedynie elementy, które albo są bezpośrednio potwierdzone przez neurofizjologię, lub zachodzi przypuszczenie, że zostaną odkryte. Nie zostały implementowane żadne mechanizmy i cechy, które byłyby definitywnie wykluczone w naukach biologicznych. Sieć jest wykonana w technice neuronów pulsujących. Sygnał wejściowy jest kodowany w strukturze sieci na różne sposoby. Każdy model SNN posiadał pewną indywidualną reprezentacje pamiętanych danych, co jest omówione przy poszczególnych modelach. W fazie weryfikacji zapamiętane informacje były bazą do porównania nadchodzących danych z wzorcem odtworzonym z zapamiętanych szczegółów. Wynik tego porównania w zależności od modelu jest binarny lub ciągły, a wartość wyniku jest możliwa do zaobserwowania w postaci przedstawionych w pracy trzech prezentacji:

wykresu w czasie chwilowej częstotliwości odpaleń wszystkich neuronów sieci, wielkości przepływającej przez sieć fali potencjałów postsynaptycznych oraz istnienia krążącego impulsu synchronizacji z danymi wejściowymi, w przypadku odpowiedzi binarnej systemu.

Teza pracy, również została potwierdzona, a dowodem jest choćby istnienie modelu SNN11j. Neuron tego modelu spełnia warunki wymagane w tezie pracy. Kodowanie wzorca w pamięci długo- i krótkoterminowej w tym modelu różnicuje funkcje neuronów w sieci oraz wymaga współpracy pomiędzy nimi. Neurony otrzymują identyczne dane wprowadzane do systemu oraz oczywiście posiadają połączenia międzyneuronowe.

Przyjęte neurofizjologiczne obostrzenia, minimalizacja złożoności neuronu pod względem przyszłej możliwości implementacji sprzętowej przy wysokiej liczbie modelowanych procesów, oraz fakt zupełnego braku teorii przetwarzania bez stanów ustalonych, w tym również w odniesieniu do pulsujących sieci neuronowych, w czasie, w którym przeprowadzono konstrukcje i symulacje nowoczesnych jak na współczesny im okres modeli SNN, jest dużym osiągnięciem. Ale jak to w tej dziedzinie bywa, wymagania szybko rosną. Następnym krokiem, którego długości obecnie określić nie sposób, jest problem wkomponowania mulipamięci w sieć typu pulsującego. Prawdopodobnie będzie to wymagało ścisłego powiązania algorytmu uczenia ze zdefiniowaną przestrzenią sieci neuronowej oraz wyodrębnienia osobnych sprzężeń zwrotnych w tak zwanych mikrokomórkach (mikrokolumnach) pamięci, co mam nadzieję będzie kolejnym etapem rozwoju sieci bazujących na symulatorze SNN.

8. INDEKS

acetylocholina - ester kwasu octowego i choliny, neurohormon produkowany na stykach (synapsach) oraz zakończeniach nerwów układu parasympatycznego (autonomiczny układ nerwowy) i neuronów ruchowych tworzących płytki nerwowo-mięśniowe. Acetylocholina działa pobudzająco na mięśniówkę gładką (tkanka mięśniowa) naczyń krwionośnych (powodując ich rozkurcz obniża ciśnienie krwi), przewodu pokarmowego (zwiększa perystaltykę) i dróg oddechowych (wywołuje skurcz oskrzeli), a także hamująco na mięsień sercowy (zwalnia czynność serca), pobudza czynność wydzielniczą gruczołów trawiennych i gruczołów dróg oddechowych (zwiększa wydzielanie śluzu). Jest rozkładana przez enzym esterazę acetylocholinową (cholinesterazę). Pochodne acetylocholiny mają zastosowanie jako leki.

dendryt - wypustka drzewkowata komórki nerwowej doprowadzająca bodźce do tejże komórki. Dendryty sąsiednich komórek nerwowych łączą się ze sobą za pomocą synaps.

istota biała - biała substancja, obszary mózgu i rdzenia kręgowego zbudowane z aksonów oraz oligodendrocytów, czyli gleju skąpokomórkowego (tkanka glejowa). Nie występują w niej ciała neuronów. Aksony otoczone są osłonką mielinową, która nadaje im biały kolor. Włókna istoty białej tworzą drogi nerwowe łączące poszczególne struktury układu nerwowego w jedną funkcjonalną całość.

istota szara - szara substancja, substancja występująca w mózgu i rdzeniu kręgowym (obok istoty białej). W kresomózgowiu mieści się na zewnątrz istoty białej i zbudowana jest z ciał komórek nerwowych (neuron) tworzących korę mózgu i móżdżku oraz ośrodki podkorowe (jądra podkorowe), natomiast w rdzeniu przedłużonym istota szara położona jest wewnętrznie. Również w rdzeniu kręgowym leży wewnątrz istoty białej, wyróżnia się w niej na przekroju rogi przednie, skąd biorą początek włókna ruchowe przednich korzeni ruchowych, i rogi tylne, do których wnikają włókna czuciowe korzeni tylnych.

komórka Schwanna – komórka należąca do tkanki glejowej, tworząca osłonki nerwowe: osłonkę Schwanna i osłonkę mielinową. Osłonka Schwanna powstaje zawsze jako pierwsza i jest utworzona z cytoplazmy (otaczającej dookoła włókno nerwowe) szeregowo ułożonych komórek Schwanna. Ten typ osłonki

pełni nie tylko funkcje izolacyjne ale również wspomaga metabolicznie otaczany akson. Osłonka mielinowa powstaje tylko na włóknach, które wcześniej miały już osłonkę Schwanna. Jest ona zbudowana z wielokrotnie okręconego wokół aksonu podwójnego fałdu błony komórkowej.

neuron, neurocyt - razem ze swoimi wypustkami - dendrytami, których jest zazwyczaj więcej niż jeden i zawsze jednym neurytem (aksonem), przystosowana do przewodzenia i przetwarzania, a także wytwarzania bodźców nerwowych. Charakteryzuje się tym, że przewodzi bodźce zawsze w jednym kierunku od dendrytów do ciała komórki (perikarionu) i z komórki dalej przez neuryt (wypustkę osiową).

Komórki nerwowe można podzielić ze względu na:

a) kształt ich perikarionu oraz obszaru utworzonego przez wypustki (dendryty i neuryt) na: neurony ziarniste, gwiaździste, piramidowe i gruszkowate.

b) liczbę wypustek i wyróżniamy tu neurony: wielobiegunowe (najczęściej występujące), dwubiegunowe (np.: w siatkówce, błonie węchowej), pseudojednobiegunowe (komórki zwojowe) i jednobiegunowe (bardzo rzadko występujące u kręgowców).

Wypustki nerwowe są odizolowane od otoczenia osłonkami nerwowymi. Natomiast cały neuron (perikarion jak i jego wypustki) jest pokryty wypustkami astrocytów (tkanka glejowa), które biorą udział w jego procesach metabolicznych i regeneracyjnych.

neuryt – akson, zwany również włóknem osiowym (nerwowe) - wypustka neuronu, zwykle długa (od kilku mikrometrów do ponad 1 m), przez którą przewodzone są i przekazywane dalszym komórkom impulsy nerwowe. Pęczek tysięcy aksonów tworzy nerw.

pamięć - jeden z podstawowych procesów psychicznych, dzięki któremu osobnik gromadzi informacje i zdobywa doświadczenie co wpływa na jego aktualne zachowania. Do podstawowych funkcji pamięci należą: zapamiętywanie, przechowywanie i odpamiętywanie (przypominanie). Zapamiętywanie polega na wytworzeniu w wyższych ośrodkach nerwowych śladu pamięciowego i kojarzeniu nowych informacji z wytworzonymi wcześniej śladami pamięciowymi. Przechowywanie to utrzymywanie się w układzie nerwowym zapamiętanego materiału. Odpamiętywanie polega na aktywizowaniu śladów pamięciowych i przybierać może formę przypomnienia,

rozpoznania lub reprodukcji. Ze względu na czas pamiętania, wyróżnia się: 1) pamięć bezpośrednią (krótkotrwałą) - pamiętanie bodźców bezpośrednio po ich zadziałaniu. 2) pamięć długotrwałą - pamiętanie bodźców, przedmiotów, czynności przez długi a czasem nieograniczony okres czasu.

pamięć długotrwała - działanie tej formy pamięci oparte jest na zmianach wartości wag synaptycznych. Proces zapamiętywania (w tym wypadku uczenia) jest długotrwały. Modyfikacja wag podczas działania sieci jest stosunkowo niewielka, dlatego uczenie wymaga długotrwałej prezentacji wzorców wejściowych. Ten rodzaj pamięci opiera się na tworzeniu struktury połączeń międzyneuronowych. Same pojedyncze sygnały nie mają tu statystycznie dużego znaczenia. Sygnały pojawiają się jedynie podczas próby dostępu do informacji.

pamięć krótkotrwała - jest to pamięć, w której główną rolę pełnią krążące w sieci impulsy w sprzężeniach zwrotnych oraz wartości potencjałów postsynaptycznych w neuronach w danej chwili czasu. Jednocześnie proces zapamiętywania jest bardzo szybki. Może wystarczyć jeden, dwa sygnały wejściowe, aby zaistniała informacja w tej pamięci. Równie szybko może ulegać kasowaniu.

sieć neuronowa - urządzenie techniczne lub algorytm, którego działanie wzorowane jest w pewnym stopniu na działaniu sieci zwierzęcych komórek nerwowych. Zazwyczaj składa się z siatki połączonych ze sobą elementów, z których każdy posiada pewną liczbę wejść i wyjść lub symuluje działanie takiej siatki. Wyjścia z poszczególnych elementów są połączone z wejściami innych. Zależność pomiędzy wejściami i wyjściami jest modyfikowana dla każdego elementu z osobna w procesie tzw. uczenia sieci. Nauczona sieć przetwarza informację poprzez powielanie sygnałów pomiędzy elementami. Zależność pomiędzy sygnałem wejściowym a wyjściowym jest następnie interpretowana jako rozwiązanie jakiegoś problemu.

serotonina - C10H12ON2, 5-hydroksytryptamina, enteramina, 5-HT, hormon tkankowy, pochodna tryptofanu. Jest wydzielana przez błonę śluzową jelit oraz neuroprzekaźnikiem w neuronach mózgowych. W tkankach obwodowych zwęża naczynia krwionośne i powoduje skurcz mięśni gładkich macicy i przewodu pokarmowego. Metylowe pochodne serotoniny mają działanie halucynogenne (omamy). Rozkład serotoniny jest katalizowany przez

monoaminooksydazę (MAO). Serotonina ma wpływ na ośrodkowy i obwodowy układ nerwowy np.: działa przeciwdepresyjnie (depresja), jest odpowiedzialna za sen fizjologiczny.

synapsa - (gr. sýnapsis 'połączenie') biol. złącze między komórkami nerwowymi w ośrodkowym układzie nerwowym, w zwojach nerwowych oraz między komórką nerwową a komórką efektora; s. jest miejscem, w którym zachodzi przekazywanie pobudzenia jednej komórki na drugą.

9.Literatura

[1] Abbott, L.F.: „Lapicque's Introduction of the Integrate-and-Fire Model", Brain Research Bulletin 50:303-304, 1999

[2] Abbott, L.F.: „The Analytical Bend of the Leech", Nature 391:18-19, 1997

[3] Abbott, L.F. „Fiddling with the Neural Code", Neuron 19:5-6

[4] Abbott, L.F.: „Decoding Neuronal Firing and Modeling Neural Networks", Quart. Rev. Biophys. 27:291-331, 1994

[5] Abbott, L.F., Blum, K.I.: „Functional Significance of Long-Term Potentiation for Sequence Learning and Prediction", Cerebral Cortex 6:406-416, 1996

[6] Abbott, L.F. and Dayan, P.: „The Effect of Correlated Variability on the Accuracy of a Population Code", Neural Comp. 11:91-102, 1998

[7] Abbott, L.F., Varela, J.A., Sen, K. and Nelson, S.B.: "Synaptic Depression and Cortical Gain Control", Science 275:220-223, 1997

[8] Abbott, L.F., Song, S.: „Temporally Asymmetric Hebbian Learning, Spike Timing and Neuronal Response Variability", w Kearns, M.S., Solla, S.A. and Cohn, D.A.: „Advances in Neural Information Processing Systems 11", MIT Press, Cambridge MA, 1999

[9] Abramowicz, H., Horn, D., Naftali, U., Sahar-Pikielny, C.: „An Orientation Selective Neural Network for Pattern Identification in Particle Detectors", w M. C. Mozer, M. J. Jordan and T. Petsche: „Advances in Neural Information Processing Systems", 9, str. 925—931, MIT Press, 1997

[10] Auer, P., Burgsteiner, H., and Maass, W.: „Reducing communication for distributed learning in neural networks", In Proc. ICANN'2002, Springer-Verlag, 2002.

[11] Auer, P., Burgsteiner, H. and Maass, W.: „The p-delta learning rule for parallel perceptrons", submitted for publication, 2001

[12] Auer, P., Holte, R.C. and Maass, W.: "Theory and applications of agnostic PAC-learning with small decision trees", In Proc. of the 12th International Machine Learning Conference, Tahoe City (USA), pages 21-29. Morgan Kaufmann (San Francisco), 1995

[13] Auer, P., Kwek, S., Maass, W., and Warmuth, M. K.: „Learning of depth two neural nets with constant fan-in at the hidden nodes", In Proc. of the 9th Conference on Computational Learning Theory 1996, pages 333-343. ACM-Press (New York), 1996

[14] Aviel, Y., Mehring, C., Abeles, M., i Horn, D.: "On embedding synfire chains in a balanced network", Neural Computation, 15, str.1321-1340, 2003

[15] Aviel, Y., Horn, D., Abeles, M.: "Memory Capacity of Balanced Networks", submitted to Neural Computation

[16] Baddeley, R., Abbott, L.F., Booth, M.J.A., Sengpiel, F., Freeman, T., Wakeman, E.A. and Rolls, E.T.: "Responses of Neurons in Primary and Inferior Temporal Visual Cortices to Natural Scenes", Proc. Roy. Soc. B264:1775-1783, London, 1997

[17] Ben-Hur, A., Horn, D., Siegelmann, H., Vapnik, V.: „Support Vector Clustering", Journal of Machine Learning Research 2, 125-137, 2001

[18] Bienenstock E.: „On the Dimensionality of Cortical Graphs", „J. Physiol.", Paris, 90: 251-256, 1996.

[19] Bienenstock, E.: „Composition In: Brain Theory - Biological Basis and Computational Theory of Vision", Ad Aertsen and Valentino Braitenberg, Elsevier, str. 269-300,1996

[20] Bienenstock, E., Geman, S.: „Compositionality in Neural Systems" w: M. Arbib: „The Handbook of Brain Theory and Neural Networks", Bradford Books/MIT, Press, str. 223-226, 1995

[21] Bienenstock, E.: **„**A Model of Neocortex", w „Network: Computation in Neural Systems", 6: 179-224 (1995)

[22] Bienenstock, E., Geman, S., Potter, D.: „Compositionality, MDL Priors, and Object Recognition", w M.C. Mozer, M.I. Jordan, and T. Petsche: „Advances in Neural Information Processing Systems 9" , MIT Press, str. 838-844,1997

[23] Bienenstock, E., Lehmann, D.: „Regulated Criticality in the Brain?", „Advances in Complex Systems", 1:361-384,1998

[24] Biłyk, T.: „Wykształcanie pamięci długo – krótkoterminowej w sieci neuronowej", Politechnika Wrocławska, Wrocław 1996

[25] Biłyk, T.: „The synergetic effects of neurons in neural network models based on physiological approach", Colloquia in artificial intelligence theory and applications CAI96, Łódź, December 19-21 1996

[26] Biłyk, T.: „Wykształcanie pamięci długo– i krótkoterminowej w modelu sieci neuronowej"; Proceedings of the Third Conference Neural Networks and their Applications and Summer School on Neural Networks Applications to Signal Processing, Częstochowa, 1997

[27] Biłyk, T.: „Methods of defect detecting in a natural neural network", Neural networks and their applications, Częstochowa, 1999

[28] Biłyk, T.: „Short term memory as a form of coding sound data", Neural networks and their applications, Częstochowa, 1999

[29] Biłyk, T.: „Coding input sequence in few pulsed neural networks", Neural Networks and Soft Computing, Częstochowa, 2000

[30] Biłyk T.: "The verification's cryterion of learning algorythms ", Intelligent Systems Design and Applications, IEEE Computer Society, Wrocław, 2005

[31] Blum, K.I. and Abbott, L.F.: „A Model of Spatial Map Formation in the Hippocampus of the Rat", Neural Comp. 8:85-93, 1996

[32] Bower, J. M.: "Computational Neuroscience: Trends in Research", 1997, str. 267-272, Plenum Press, New York, 1997

[33] Brunel, N., Wang: „Effects of neuromodulation in a cortical network model of object working memory dominated by recurrent inhibition", Journal of Computational Neuroscience, 63-85. X.-J, 2001

[34] Burkitt, A. N.: „An information-theoretic analysis of the coding of a periodic synaptic input by integrate-and-fire neurons", Neurocomputing 44-46, 69-74, 2002

[35] Burkitt, A. N.: „Balanced neurons: Analysis of leaky integrate-and-fire neurons with reversal potentials", "Biological Cybernetics 85", 247-255, 2001

[36] Burkitt, A. N.: "Interspike interval variability for balanced networks with reversal potentials for large numbers of inputs", "Neurocomputing 32-33", 313-321, 2000

[37] Burkitt, A.N.: "Retrieval properties of attractor neural networks incorporating biological features - a self-consistent signal-to-noise analysis", w

[38] Burkitt, A.N.: „Retrieval properties of attractor neural networks that obey Dale's law using a self-consistent signal-to-noise analysis", „Network: Computation in Neural Systems 7", 517-532, 1996

[39] Burkitt, A.N.: „Retrieval properties of attractor neural networks that obey Dale's law using a self-consistent signal-to-noise analysis", „Network: Computation in Neural Systems 7", 517-532, 1996

[40] Burkitt, A. N., and Clark, G. M.: „Synchronization of the neural response to noisy periodic synaptic input", „Neural Computation 13(12)", 2639-2672, 2001

[41] Burkitt, A. N., and Clark, G. M.: „Calculation of interspike intervals for integrate-and-fire neurons with Poisson distribution of synaptic inputs", „Neural Computation 12(8)", 1789-1820, 2000

[42] Burkitt, A. N. and Clark, G. M.: „Analysis of synchronization in the response of neurons to noisy periodic synaptic input", „Neurocomputing 32-33", 67-75, 2000

[43] Burkitt, A. N., Clark, G. M.: „Analysis of integrate-and-fire neurons: synchronization of synaptic input and spike output", „Neural Computation 11(4)", 871-901, 1999

[44] Burkitt, A. N., Clark, G. M.: „Modelling the responses of neurons to auditory stimuli: differences between acoustical and electrical stimulation", w „Proceedings of the IEEE Engineering in Medicine and Biology (EMBS) Conference", str. 46-49, 22-23 Luty 1999, Monash University, Australia, 1999

[45] Burkitt, A. N. and Clark, G. M.: „New technique for analyzing integrate-and-fire neurons", Neurocomputing 26-27 (1999) 93-99, 1999

[46] Burkitt, A. N., Clark, G. M.: „New method for analyzing the synchronization of synaptic input and spike output in neural systems", „Australian Journal of Intelligent Information Processing Systems 5", 50-54, 1998

[47] Chance, F.S., Abbott, L.F. „A Recurrent Network Model for the Phase-Invariance of Complex Cell Responses", w Neurocomputing 32-33:339-344, 2000

[48] Chance, F.S., Abbott, L.F.:"Divisive Inhibition in Recurrent Networks", Network: Comp. Neural Sys., 11:119-129, 2000

[49] Chance, F.S., Nelson, S.B., Abbott, L.F.: „Recurrent Cortical Amplification Produces Complex Cell Responses", w Kearns, M.S., Solla, S.A. and Cohn, D.A.: „Advances in Neural Information Processing Systems", 11, MIT Press, Cambridge MA, 1999

[50] Chechik, G., Horn, D., Ruppin, E.: „Neuronal Regulation and Hebbian Learning", Handbook of Brain Theory and Neural Networks, 2nd. ed. 2000

[51] Date, A., Bienenstock, E., Geman, S.: „On the Temporal Resolution of Neural Activity", „Technical Report", Division of Applied Mathematics, Brown University, 1998

[52] Duch, Wodzisław, Grudziński, Karol, Stawski, Grzegorz: „Symbolic Features in Neural Networks", Neural Networks and Soft Computing, Zakopane 2000

[53] Eccles, S., Carew, J. „The Physiology of Nerve Cells", Herter Lectures of Johns Hopkins University, 1957

[54] Fregnac, Y., and Bienenstock, E.: „Correlational Models of Synaptic Plasticity: Development, Learning and Cortical Dynamics of Mental Representations", w T.J. Carew, R. Menzel, C.J. Shatz: „Mechanistic Relationships between Development and Learning", Wiley and sons, str. 113-148, 1998

[55] Gerstner, W. and Abbott, L.F.: „Learning Navigational Maps Through Potentiation and Modulation of Hippocampal Place Cells", J. Computational Neurosci. 4:79-94, 1997

[56] Goldman, M.S., Nelson, S.B., Abbott, L.F.: „Decorrelation of Spike Trains by Synaptic Depression", w Neurocomputing 26-27:147-153, 1999

[57] Gottlieb, A., Horn, D.: „Algorithm for data clustering in pattern recognition problems based on quantum mechanics", Phys. Rev. Lett. 88 18702, 2002

[58] Guython, A.: „Textbook of medical physiology", sixth edition W.B.Saunders company 1981

[59] Hendin, O., Horn, D., Tsodyks, M.:"The Role of Inhibition in an Associative Memory Model of the Olfactory Bulb", Journal of Computational Neuroscience 4, 173-182 (1997).

[60] Hendin, O., Horn, D., Tsodyks, M.: „Associative Memory and Segmentation in an Oscillatory Neural Model of the Olfactory Bulb", Journal of Computational Neuroscience, 5.1998

[61] Hinton, G. E. and Sejnowski, T. J.: „Learning and relearning in Boltzmann machines", In Rumelhart, D. E. and McClelland, J. L., editors, Parallel Distributed Processing: Explorations in the Microstructure of Cognition. Volume 1: Foundations, MIT Press, Cambridge, MA.

[62] Hohn, N. and Burkitt, A. N.: „Shot noise in the leaky integrate-and-fire neuron", „Phys. Rev. E 63", 031902, 2001

[63] Hopfield, J.J.: „ Transforming neural computations and representing time", PNAS, 1996

[64] Hopfield J.J.: „Neural networks and physical systems with emergent collective computational abilities", Proc. of the Nat. Acad. of Sc., USA 1982

[65] Horn, D., Axel, I.: "Quantum clustering of microarray expression data in a truncated SVD space", Bioinformatics, w druku

[66] Horn, D., Levanda, S.: „Fast Temporal Encoding and Decoding with Spiking Neurons", Neural Computation, 10, 1705—1720, 1998

[67] Horn, D., Levy, N., Ruppin, E.: „The importance of nonlinear dendritic processing in multimodular memory networks", Neurocomputing 26, 389-394, 1999

[68] Horn, D., Levy, N. and Ruppin, E.: „Multimodular Networks and Semantic Memory Impairments", w J.A. Reggia, E. Ruppin, D.L. Glanzman: „Brain, Behavioral and Cognitive Disorders: The Neurocomputational Perspective", „Progress in Brain Research", Elsevier Science Pub., 67-72, 1999

[69] Horn, D., Levy, N., Ruppin, E.: „Memory Maintenance Via Neuronal Regulation", Neural Computation, 10, 1-18 (1998)

[70] Horn, D., Levy, N., Ruppin, E.,: „Neuronal Regulation vs Synaptic Unlearning in Memory Maintenance Mechanisms", Network: Comp. in Neural Systems, 8, 577-586, 1998

[71] Horn, D. i Opher, I.:"Solitary Wawes of Integrate and Fire Neural Fields", Neural Computation, 1997

[72] Horn, D., Opher, I.: „Complex Dynamics of Neuronal Thresholds", Neurocomputing 32, str. 161-166, 2000

[73] Horn, D., and Opher, I.: „Collective Excitation Phenomena and their Applications", W. Maass and C. B. Bishop: „Pulsed Neural Networks", rozdział 11, MIT Press, 1998

[74] Horn, D., and Opher, I.: „Segmentaion and Binding in Oscillatory Neural Systems", w D. Levine, V. Brown and T. Shirey: „Oscillations in Neural Systems", (Erlbaum), 1996

[75] Jahnke, A. Roth, U., i Klar H.: „A SIMD/dataflow arcitecture for a nurocomputer for spike-processing neural networks", IEEE Computer Society Press, 1996

[76] Kacprzyk, T. i Ślot, K.: „Sieci neuronowe komórkowe", WN PWN, Warszawa- Łódź 1995

[77] Kohonen T: „Self-Organisation and Associative Memory", Springer Verlag, Hilderberg 1984

[78] Kandel, E.R., Hawkins, R.D.:Świat nauki,"Umysł a mózg", Scientific American, Listopad 1992

[79] Korbicz, J., Obuchowicz, A., Uciński, D.: „Sztuczne sieci neuronowe", AOW PLJ, Warszawa 1994

[80] Kudrewicz, J.: „Fraktale i chaos", WN-T, Warszawa 1993

[81] Kuhlmann, L., Burkitt, A. N., Paolini, A. G. and Clark, G.M. : "Summation of spatiotemporal input patterns in leaky integrate-and-fire neurons: Application to neurons in the cochlear nucleus receiving converging auditory nerve fiber input", „J. Comput. Neurosci. 12", 55-73, 2002

[82] Legenstein, R. A., Maass, W.: „Optimizing the layout of a balanced tree", submitted for publication, 2001

[83] Legenstein, R. A., Maass, W.: „Neural circuits for pattern recognition with small total wire length", Theoretical Computer Science, 2001

[84] Legenstein, R. A., Maass, W.: „Total wire length as a salient circuit complexity measure for sensory processing", Submitted for publication, 2001

[85] Legenstein, R. A., Maass, W.: „Foundations for a circuit complexity theory of sensory processing", In Todd K. Leen,Thomas G. Dietterich, and Volker Tresp, editors: „Advances in Neural Information Processing Systems", (NIPS '2000), volume 13, pages 259-265, MIT Press, Cambridge, 2001

[86] Levy, N., Horn, D., Meilijson, I., Ruppin, E.: „Distributed Synchrony in a Cell Assembly of Spiking Neurons", Neural Networks, 14, 815-824, 2001 (Internet, 28 wrzesień 2000)

[87] Levy, N., Horn, D., Ruppin, E.: „Multi-modular Associative Memory", w Michael I. Jordan, Michael J. Kearns, and Sara A. Solla: „Advances in Neural Information Processing Systems", 10, pp 52—58, MIT Press, Cambridge, 1998

[88] Levy, N., Horn, D., Ruppin, E.: „Associative Memory in a Multi-modular Network", Neural Computation 11, 1717-1737 (1999)

[89] Maass, W.: "Computing with spikes", Special Issue on Foundations of Information Processing of TELEMATIK, 8(1):32-36, 2002

[90] Maass, W.: „Neural computation: a research topic for theoretical computer science? Some thoughts and pointers", In Rozenberg G., Salomaa A., and Paun G., editors: „Current Trends in Theoretical Computer Science, Entering the 21th Century", pages 680-690, World Scientific Publishing, 2001

[91] Maass, W.: "Neural computation with winner-take-all as the only nonlinear operation", In Sara A. Solla, Todd K. Leen, and Klaus-Robert Müller, editors: "Advances in Information Processing Systems", volume 12, pages 293-299. MIT Press (Cambridge), 2000

[92] Maass, W.: "Computation with spiking neurons", In M. A. Arbib, editor: "The Handbook of Brain Theory and Neural Networks", MIT Press (Cambridge), 2nd edition, 2001

[93] Maass, W.: "On the computational power of winner-take-all", Neural Computation, 12(11):2519-2536, 2000

[94] Maass, W.: "Paradigms for computing with spiking neurons", In Leo van Hemmen, editor: "Models of Neural Networks", volume 4. Springer (Berlin), 1999

[95] Maass, W.: "On the role of time and space in neural computation", In Proc. of the Federated Conference of CLS'98 and MFCS'98, Mathematical Foundations of Computer Science 1998, volume 1450 of "Lecture Notes in Computer Science", pages 72-83. Springer (Berlin), 1998

[96] Maass, W.: "Computing with spiking neurons", In W. Maass and C. M. Bishop, editors: "Pulsed Neural Networks", pages 55-85. MIT Press (Cambridge), 1999

[97] Maass, W. : "On the relevance of time in neural computation and learning", In M. Li and A. Maruoka, editors: "Proc. of the 8th International Conference on Algorithmic Learning Theory in Sendai (Japan)", volume 1316 of Lecture Notes in Computer Science, pages 364-384. Springer (Berlin), 1997

[98] Maass, W.: "Noisy spiking neurons with temporal coding have more computational power than sigmoidal neurons", In M. Mozer, M. I. Jordan, and T. Petsche, editors: "Advances in Neural Information Processing Systems", volume 9, pages 211-217. MIT Press (Cambridge), 1997

[99] Maass, W.: "Networks of spiking neurons: the third generation of neural network models", Neural Networks, 10:1659-1671, 1997

[100] Maass, W.: "A model for fast analog computations with noisy spiking neurons. In James Bower, editor: "Computational Neuroscience: Trends in research", pages 123-127, 1997

[101] Maass, W.: "Fast sigmoidal networks via spiking neurons", Neural Computation, 9:279-304, 1997

[102] Maass, W.: "On the computational power of noisy spiking neurons", In D. Touretzky, M. C. Mozer, and M. E. Hasselmo, editors: "Advances in Neural Information Processing Systems", volume 8, pages 211-217. MIT Press (Cambridge), 1996

[103] Maass, W.: "On the computational complexity of networks of spiking neurons", In G. Tesauro, D. S. Touretzky, and T. K. Leen, editors: "Advances in Neural Information Processing Systems", volume 7, pages 183-190. MIT Press (Cambridge), 1995

[104] Maass, W.: "On the complexity of learning on neural nets", In J. Shawe-Taylor and M. Anthony, editors: "Computational Learning Theory: EuroColt'93" , pages 1-17. Oxford University Press (Oxford), 1994

[105] Maass, W.: "Efficient agnostic PAC-learning with simple hypotheses", In Proc. of the 7th Annual ACM Conference on Computational Learning Theory, pages 67-75, 1994

[106] Maass, W.: "Computing on analog neural nets with arbitrary real weights", In V. P. Roychowdhury, K. Y. Siu, and A. Orlitsky, editors: "Theoretical Andvances in Neural Computation and Learning", pages 153-172. Kluwer Academics Publisher (Boston), 1994

[107] Maass, W.: "Vapnik-Chervonenkis dimension of neural nets", In M. A. Arbib, editor: "The Handbook of Brain Theory and Neural Networks", pages 1000-1003. MIT Press (Cambridge), 1995

[108] Maass, W.: "Perspectives of current research about the complexity of learning on neural nets", In V. P. Roychowdhury, K. Y. Siu, and A. Orlitsky, editors: "Theoretical Advances in Neural Computation and Learning", pages 295-336. Kluwer Academic Publishers (Boston), 1994

[109] Maass, W., Markram, H.: "Temporal integration in recurrent microcircuits", In M. A. Arbib, editor, "The Handbook of Brain Theory and Neural Networks", MIT Press (Cambridge), 2nd edition, 2002

[110] Maass W., Markram, H.: "On the computational power of recurrent circuits of spiking neurons", submitted for publication, 2002

[111] Maass W., Markram, H.: "Synapses as dynamic memory buffers", Neural Networks, 15:155-161, 2002

[112] Mass, W., Bishop, Ch.: „Pulsed Neural Networks",MIT Press, London 1999

[113] Maass, W., Legenstein, R.A., Markram, H.: „A new approach towards vision suggested by biologically realistic neural microcircuit models", In Proc. of the 2nd Workshop on Biologically Motivated Computer Vision, Lecture Notes in Computer Science. Springer, Nov. 2002.

[114] Maass, W.: „Lower bounds for the computational power of networks of speaking neurons", Neural Computation, 1996

[115] Maass, W., Orponen, P.: „On the effect of analog noise in discrete-time analog computations", Neural Computation, 10:1071-1095, 1998

[116] Maass, W., Ruf, B.: „On computation with pulses", Information and Computation, 148:202-218, 1999

[117] Maass, W., Ruf, B.: „On the relevance of the shape of postsynaptic potentials for the computational power of networks of spiking neurons", In Proc. of the International Conference on Artificial Neural Networks ICANN, pages 515-520, Paris, 1995

[118] Maass, W., Schmitt, M.: „On the complexity of learning for spiking neurons with temporal coding", Information and Computation, 153:26-46, 1999.

[119] Maass, W., Schmitt, M.: „On the complexity of learning for a spiking neuron", In Proc. of the 10th Conference on Computational Learning Theory 1997, pages 54-61. ACM-Press (New York), 1997

[120] Maass, W., Sontag, E.D.: „A precise characterization of the class of languages recognized by neural nets under Gaussian and other common noise distributions", In M. S. Kearns, S. S. Solla, and D. A. Cohn, editors: „Advances in Neural Information Processing Systems", volume 11, pages 281-287. MIT Press (Cambridge), 1999

[121] Maass, W., Sontag, E.D.: „Neural systems as nonlinear filters. Neural Computation", 12(8):1743-1772, 2000

[122] Maass, W., Sontag, E.D.: „Analog neural nets with Gaussian or other common noise distribution cannot recognize arbitrary regular languages", Neural Computation, 11:771-782, 1999

[123] Maass, W., Steinbauer, G., and Koholka, R.: „Autonomous fast learning in a mobile robot", In G. D. Hager, H. I. Christensen, H. Bunke, and R. Klein, editors „Sensor Based Intelligent Robots", International Workshop, Dagstuhl Castle, Germany, October 15-25, 2000, Selected Revised Papers, volume 2238 of Lecture Notes in Computer Science, pages 345-356, Springer (Berlin), 2002

[124] Maass, W., Zador, A,: „Computing and learning with dynamic synapses", In W. Maass and C. Bishop, editors: „Pulsed Neural Networks", pages 321-336. MIT-Press (Cambridge), 1998

[125] Maass, W., Zador, A.M.: „Dynamic stochastic synapses as computational units", Neural Computation, 11(4):903-917, 1999

[126] Markam, H., Wang, Y, Tsodyks, M.: "Differential signaling via the same axon of neocortical pyramidal neurons", Proc. Natl. Acad. Sci., 1998

[127] Miller, G., Horn, D.: "Probability Density Estimation using Entropy Maximization", Neural Computation, 10, 1925--1938 (1998)

[128] Naftaly, U., Intrator, N., Horn, D.: "Optimal Ensemble Averaging of Neural Networks", Computation in Neural Systems, 8, 283-296 (1997)

[129] Natschläger, T., Maass, W., Zador, A.: „Efficient temporal processing with biologically realistic dynamic synapses", Computation in Neural Systems, 12:75-87, 2001

[130] Natschläger, T., Maass, W., Markram, H.:" The "liquid computer": A novel strategy for real-time computing on time series", Special Issue on Foundations of Information Processing of TELEMATIK, 8(1):39-43, 2002

[131] Natschlager, T., Maass, W.: "Spiking Neurons and the Introduction of Finite State Machines", Elsevier Preprint, October 2001

[132] Natschlager, T., Maass, W.: "Computing the Optimally Fitted Spike Train for a Synapse", Neural Computation 13(11):2477-2494, 2001.

[133] Natschlager, T., Maass, W.: "Real-Time Computing Without Stable States: A New Framework for Neural Computation Based on Perturbations", nie opublikowane, 2002

[134] Natschläger, T., Maass, W.: "Finding the key to a synapse", In Todd K. Leen, Thomas G. Dietterich, and Volker Tresp, editors: "Advances in Neural

Information Processing Systems" (NIPS '2000), volume 13, pages 138-144, MIT Press, Cambridge, 2001

[135] Natschläger, T., Maass, W.: "Fast analog computation in networks of spiking neurons using unreliable synapses", In ESANN'99 Proceedings of the European Symposium on Artificial Neural Networks, pages 417-422, Bruges, Belgium, 1999

[136] Natschläger, T., Maass, W.: "Emulation of Hopfield networks with spiking neurons in temporal coding", In J. M. Bower, editor: "Computational Neuroscience": Trends in Research, pages 221-226. Plenum Press, 1998

[137] Natschläger, T., Maass, W.: "A model for fast analog computation based on unreliable synapses", Neural Computation, 12(7):1679-1704, 2000

[138] Natschläger, T., Maass, W.: "Associative memory with networks of spiking neurons in temporal coding", In L. S. Smith and A. Hamilton, editors: "Neuromorphic Systems: Engineering Silicon from Neurobiology", pages 21-32. World Scientific, 1998

[139] Natschläger, T., Maass, W.: "Networks of spiking neurons can emulate arbitrary Hopfield nets in temporal coding", Network: Computation in Neural Systems, 8(4):355-372, 1997

[140] Natschläger, T. ,Maass, W., Sontag, E.D., Zador, A.: „Processing of time series by neural circuits with biologically realistic synaptic dynamics", In Todd K. Leen, Thomas G. Dietterich, and Volker Tresp, editors: „Advances in Neural Information Processing Systems 2000" (NIPS '2000), volume 13, pages 145-151, MIT Press, Cambridge, 2001

[141] Natschläger, T., Markram, H., Maass, W.: "Computer models and analysis tools for neural microcircuits", In R. Kötter, editor "A Practical Guide to Neuroscience Databases and Associated Tools" chapter 9. Kluver Academic Publishers (Boston), 2002

[142] Opher, I., Horn, D., Quenet, B.: „Clustering with Spiking Neurons", prezentacja ustna, ICANN-99, 1999

[143] Quenet B., Horn, D.: „The Dynamic Neural Filter: A Binary Model of Spatiotemporal Coding", Neural Computation, 2002

[144] Quenet, B., Horn, D., Dreyfus, G.: „Temporal Coding in an Olfactory Oscillatory Model", CNS00, 2000

[145] Salinas, E. and Abbott, L.F.:"Do Simple Cells in Primary Visual Cortex Form a Tight Frame?", Neural Comp. 12:313-336, 2000

[146] Salinas, E. and Abbott, L.F.: „Invariant Visual Perception from Attentional Gain Fields", J. Neurophysiol. 77:3267-3272, 1997

[147] Salinas, E., Abbott, L.F.: „A Model of Multiplicative Neural Responses in Parietal Cortex", Proc. Natl. Acad. Sci. USA 93:11956-11961, 1996

[148] Salinas, E. and Abbott, L.F.: „Transfer of Coded Information from Sensory to Motor Networks", J. Neurosci. 15:6461-6474, 1995

[149] Salinas, E. and Abbott, L.F.: „Vector Reconstruction from Firing Rates", J.Computational Neurosci. 1:89-107, 1994

[150] Savage, J.: „Models of Computation: Exploring the Power of Computing", Addison-Wesley, Reading, MA 1998

[151] Sedlak, W.: „Homo electronicus", PIW, Warszawa, 1980

[152] Sen, K., Jorge-Rivera, J.C., Marder, E. and Abbott, L.F.: „Decoding Synapses", J. Neurosci. 16:6307-6318, 1996

[153] Song, S. and Abbott, L.F.: „Temporally Asymmetric Hebbian Learning and Neuronal Response Variability", w Neurocomputing 32-33: 523-528, 2000

[154] Steinbauer, G., Koholka, R., Maass, W.: „A very short story about autonomous robots", Special Issue on Foundations of Information Processing of TELEMATIK, 8(1):26-29, 2002

[155] Steward, I.: „Czy Bóg gra w kości? - matematyka chaosu", WN PWN, Warszawa 1994

[156] Strojnowski,J.: „ Psychologia fizjologiczna",KUL, Lublin 1989

[157] Tadeusiewicz R., Izworski A., Majewski J.: "Biometry", AGH publ., Kraków, 1993

[158] Tadeusiewicz, R.: „Problemy biocybernetyki", PWN, Warszawa 1994

[159] Tadeusiewicz, R.: „Sieci neuronowe", AOW, Warszawa 1993

[160] Takanishi, G.: "Computation in Neural Clusters", draft, Kaneohe USA, Maj 2001

[161] Takanishi, G.: "Computation in Neural Clusters", draft, Kaneohe USA, Maj 2001

Załącznik nr 1 - Opis formatów danych

Dane przetwarzane w tej pracy, z powodu ich objętości, zostały załączone w formie cyfrowej na dołączonej płycie CD.

Płyta zawiera również kody źródłowe poszczególnych modeli IMPL, SNN oraz kody źródłowe odpowiednich tym modelom symulatorów.

Symulator IMPL oraz SNN operuje na różnych formatach danych związanych z dodatkowymi informacjami zawartymi w odpowiednich plikach, co nie oznacza, że wspólna idea dotycząca kodowania zgodnie ze wzorem (6) nie została zachowana.

Załączona płyta CD zawiera przykładowe dane wejściowe użyte w pracy w formacie wejściowego sygnału dźwiękowego oraz kolejnych jego przekształceń. Symulator SNN posiada również zaimplementowane procedury odtwarzania sygnału dźwiękowego z formatu danych jakim operuje pulsująca sieć neuronowa. Ponieważ przekształcenie to przebiega w kilku etapach, załączone są również pliki zawierające dane z poszczególnych transformacji.

Precyzując format danych docierający do sieci neuronowej, który powstaje po przetworzeniu spróbkowanego sygnału dźwiękowego (np. fonetycznej reprezentacji cyfr od zera do dziewięć) możemy zauważyć, że pomimo zdefiniowania klasy funkcji *G* we wzorze (6) w dalszym ciągu jest stosunkowo duża swoboda w określeniu jej danej reprezentacji. Ważne jest dodatkowo, aby funkcja była monotoniczna, najlepiej niemalejąca, w miarę liniowo przekształcająca wartości dziedziny na przeciwdziedzinę, a przedziały w których funkcja jest funkcją stałą - równej długości oraz równomiernie rozmieszczone w dziedzinie. Dodatkowo, wartości *1* oraz *0* powinny być względnie równomiernie rozmieszczone w wektorze wyjściowym.

Przepływ danych w symulatorze oraz odpowiednie pliki są następujące:

1) sygnał dźwiękowy .pcm
2) wyniki FFT .fft
3) modulacja zgonie z (6) – wejście do sieci .bfm
4) wyjście z sieci .bfm
5) wyliczenie energii w przedziałach widma .fft

6) złożenie sygnału dźwiękowego .pcm

Pliki zawierają dane zgodnie z opisanymi przekształceniami, bez formatujących struktur nagłówkowych. Większość modeli była testowana z użyciem podziału widma sygnału wejściowego na dwanaście przedziałów częstotliwości, co implikuje dwanaście wejść do systemu zawierającego pulsująca sieć neuronową.

Symulator potrafi przechowywać sieć wraz z jej stanem dynamicznym (m.in. wartością refrakcji neuronów, konfiguracją aktywnych neuronów, bieżącym wektorem przetwarzanych danych wejściowych itp.) w pliku typu .SNN.

Plik ten ma następujący format:

1) słowo "Dokument"
2) wskaźnik na bieżący wektor przetwarzanych danych
3) słowo "Sieć"
4) zawartość buforów miedzy wejściem a wyjściem neuronów
5) sekcja powtarzająca się zgodnie z ilością neuronów:
    1. słowo "Neuron"
    2. wektor wag
    3. wartość progu
    4. wartość refrakcji
    5. faza pracy neuronu
    6. wartość blokady uczenia

W odpowiednich katalogach na załączonej płycie znajdują się:

1) przekształcenia wejściowe – zbiór plików obrazujący kolejne etapy przetwarzania dźwiękowego sygnału wejściowego, oraz przetwarzanie obrazujące proces odwrotny. Procedury wykonujące przetwarzanie dostępne są jako samodzielny program lub jako odpowiednie funkcje wywoływane z symulatora SNN.
2) modele IMPL – zawiera katalogi i archiwa modeli sieci szesnastobitowego symulatora IMPL. W odpowiednich plikach w języku C++ można znaleźć poszczególne implementacje.

3) modele SNN – podobnie jak wyżej, ale dotyczy trzydziestodwubitowego symulatora SNN. Archiwa zawierają implementacje poszczególnych modeli.
4) katalog roboczy symulatora – zawiera pliki wejściowe do sieci jak i pliki wyjściowe, oba typu .bfm; pliki definiujące stan różnych modeli sieci w różnych stadiach symulacji oraz pliki poszczególnych transformacji dźwiękowego sygnału wejściowego.