

POLITECHNIKA WROCŁAWSKA

Instytut Sterowania i Techniki Systemów

**MODELOWANIE OBIEKTÓW
ZŁOŻONYCH Z WYKORZYSTANIEM
SIECI NEURONOWYCH**

mgr inż. Grzegorz Drałus

(Rozprawa doktorska)

Słowa kluczowe:

- sieci neuronowe wielowarstwowe
- algorytmy uczenia sieci
- modele lokalnie optymalne
- modele globalnie optymalne
- wielokryterialne wskaźniki jakości

Promotor:

dr hab. inż. Jerzy Świątek

Profesor Politechniki Wrocławskiej

Wrocław 2001

Spis treści

Wykaz ważniejszych oznaczeń	4
1 Modelowanie obiektów złożonych z wykorzystaniem sieci neuronowych – pojęcia podstawowe	5
1.1 WSTĘP	5
1.2 SIECI NEURONOWE	7
1.2.1 Opis sztucznego neuronu.....	7
1.2.2 Sieć jednowarstwowa jednokierunkowa	8
1.2.3 Sieć wielowarstwowa jednokierunkowa	9
1.3 MODELOWANIE POJEDYNCZEGO ELEMENTU OBIEKTU	11
1.3.1 Uczenie sieci neuronowej.....	12
1.3.2 Uczenie pod nadzorem sieci jednowarstwowej	13
1.3.3 Uczenie pod nadzorem sieci wielowarstwowej	14
1.4 OBIEKTY ZŁOŻONE.....	16
1.4.1 Modelowanie obiektów złożonych.....	17
1.4.2 Modelowanie obiektów złożonych z wykorzystaniem sieci neuronowych	20
1.5 CEL I ZAKRES PRACY	22
2 Modele i algorytmy wyznaczania parametrów w modelowaniu obiektów złożonych	23
2.1 MODEL LOKALNIE OPTYMALNY	23
2.2 MODEL GLOBALNIE OPTYMALNY	25
2.3 MODEL GLOBALNIE OPTYMALNY Z UWZGLĘDNIENIEM JAKOŚCI MODELI LOKALNYCH ...	30
2.3.1 Syntetyczny wskaźnik jakości.....	31
2.3.2 Zadowalające modele lokalne	34
• Zewnętrzna funkcja kary	35
• Wewnętrzna funkcja kary	40
3 Badania symulacyjne.....	45
3.1 PLAN I ZAKRES BADAŃ.....	45
3.2 WYNIKI BADAŃ SYMULACYJNYCH	47
3.3 OMÓWIENIE WYNIKÓW	55
4 Modelowanie procesu produkcji azotynu amonu	58
4.1 OPIS PROCESU	58

4.2 DANE UCZĄCE I TESTUJĄCE.....	59
4.3 MODELE PROCESU I WYNIKI MODELOWANIA.....	61
4.3.1 Modelowanie procesu – wariant „I”	61
4.3.2 Modelowanie procesu – wariant „II”	73
4.3.3 Modelowanie procesu – wariant „III”	80
4.4 OMÓWIENIE WYNIKÓW	90
5 Podsumowanie.....	95
Literatura	100

Wykaz ważniejszych oznaczeń

b	- przesunięcie
$e^{(r)}$	- błąd uczenia r -tego modelu lokalnego ($e^{(r)} = \bar{y}^{(r)} - y^{(r)}$)
$\hat{e}$	- błąd uczenia modelu globalnego ($\hat{e} = \hat{y} - y$)
f	- funkcja aktywacji neuronu
u	- wektor wejściowy obiektu złożonego
$u^{(r)}$	- wektor wejściowy r -tego elementu obiektu
$\hat{u}^{(r)}$	- wektor wejściowy r -tego modelu częściowego
w	- macierz współczynników wagowych
Δw	- wektor przyrostu wag
y	- wektor wyjściowy obiektu złożonego
$\hat{y}$	- wektor wyjściowy modelu globalnego
$y^{(r)}$	- wektor wyjściowy r -tego elementu obiektu
$\bar{y}^{(r)}$	- wektor wyjściowy r -tego modelu lokalnego
$\hat{y}^{(r)}$	- wektor wyjściowy r -tego modelu częściowego
B	- wewnętrzna funkcja kary
G	- funkcja kary
K	- zewnętrzna funkcja kary
T	- zmodyfikowana funkcja kryterialna
Q	- wskaźnik jakości modelu globalnego
Q_1, Q_2, Q_3	- wskaźniki jakości modeli lokalnych
Q_u	- wskaźnik jakości modelu globalnego – szczególny przypadek
Q_w	- syntetyczny wskaźnik jakości
APE	- błąd względny
α_r	- r -ty współczynnik w syntetycznym wskaźniku jakości
β_r	- r -ty współczynnik w globalnym wskaźniku jakości
λ_r	- ograniczenie na wskaźnik jakości r -tego modelu lokalnego
η	- współczynnik uczenia

1. Modelowanie obiektów złożonych z wykorzystaniem sieci neuronowych – pojęcia podstawowe

1.1 Wstęp

Identyfikacja i modelowanie matematyczne systemów należą do grupy problemów o ważnym znaczeniu zarówno poznawczym, jak i praktycznym. Zagadnieniom tym poświęcono sporo miejsca w literaturze, np. w [3, 39] podano m.in. metodologię identyfikacji obiektów statycznych i dynamicznych, w [13, 34, 49, 51] podano szeroki zakres metod deterministycznych i probabilistycznych (metody: najmniejszych kwadratów, błędu predykcji, zmiennych instrumentalnych, największej wiarygodności, itp.). Powyższe metody modelowania matematycznego wymagają szerokiej znajomości i dogłębnej wiedzy matematycznej, jak również wiedzy z innych dziedzin takich fizyka, chemia czy automatyka i sterowanie, jak i pewnego doświadczenia w prowadzeniu zadań identyfikacji.

Wydaje się, że obszar aktualnych zagadnień z zakresu metodologii modelowania, jak i również technicznych środków służących do otrzymywania modeli, leży na pograniczu kilku dziedzin, w których informatyka odgrywa kluczową rolę.

Z czasem, gdy sieci neuronowe stawały się coraz popularniejsze, ich ważne własności takie, jak przetwarzanie sygnałów dyskretnych, praca równoległa, odwzorowywanie nieliniowych funkcji oraz uczenie się i adaptacja, znalazły naturalne zastosowanie do modelowania obiektów. Wystarczy znajomość danych wejściowych i wyjściowych modelowanego obiektu, aby zbudować model neuronowy. Proces doboru współczynników modelu neuronowego, które często są trudne do zinterpretowania, odbywa się automatycznie, tzn., nieabsorbując od strony projektanta. Potrzebna jest jednak pewna wiedza o zasadach doboru struktury modelu (funkcji aktywacji, liczby neuronów i warstw). W zasadzie sieci neuronowe mogą modelować obiekty statyczne [5, 20, 21, 22, 27, 29, 40, 41, 43, 48, 52, 53, 61]. Jednak opracowano pewne ich modyfikacje, które mogą modelować również obiekty dynamiczne [20, 29, 38, 40, 53, 55, 61]. Wydaje się jednak, że w tym zakresie jest jeszcze wiele do zrobienia.

Opis modelowania matematycznego obiektów złożonych w literaturze zawierają prace [2, 3, 4, 33]. Zawarto tam opis modeli matematycznych obiektów złożonych, identyfikację obiektu złożonego z agregatów oraz podano uwagi o identyfikacji kompleksu operacji. Z kolei praca [54] zawiera opis modelowania obiektów złożonych heterogenicznych. Inne publikacje dotyczą matematycznych metod modelowania, takich jak metody celowo

zorientowane [7], dekompozycji [16], czy graficzne w oparciu o proces globalny uwzględniający kilka dziedzin nauki [50]. Istnieje jeszcze szereg innych publikacji na temat modelowania matematycznego obiektów złożonych oraz metod identyfikacji, np. [1, 56].

Nietypowe wykorzystanie sieci neuronowych do modelowania obiektów złożonych polegające na zasadzie złożenia sieci prostych zawiera praca [35]. Praca ta nie uwzględnia jednak wielokryterialnych ocen w proponowanych modelach.

Warto też wspomnieć o zastosowaniu logiki rozmytej w połączeniu z sieci neuronowymi w identyfikacji i modelowaniu obiektów [18, 19, 42, 46, 60].

Matematyczne metody identyfikacji obiektów złożonych polegają najczęściej na modelowaniu części składowych systemu. Identyfikacja lokalna poszczególnych elementów składowych obiektów, bez uwzględnienia faktu, że obiekty te są częściami składowymi pewnego obiektu złożonego daje modele *lokalnie optymalne*. Zastąpienie poszczególnych elementów obiektów ich modelami lokalnie optymalnymi nie daje w efekcie modelu *globalnie optymalnego*, czyli najlepszego modelu obiektu złożonego traktowanego jako całość.

Zamiarem autora pracy jest opracowanie metod i narzędzi tworzenia modeli w postaci sieci neuronowych dla nieliniowych statycznych obiektów złożonych o strukturze szeregowej, ze szczególnym uwzględnieniem problemu konstrukcji modelu lokalnie optymalnego, modelu globalnie optymalnego. Ponadto, korzystając z podejścia wielokryterialnego autor zamierza uwzględnić jakość modeli lokalnych w modelowaniu globalnym. Proponowane różne podejścia powinny w efekcie umożliwić zastosowanie zaproponowanych modeli do projektowania algorytmów sterowania obiektami złożonymi.

Przegląd treści pracy

W rozdziale pierwszym przedstawiono wybrane wiadomości dotyczące sieci neuronowych. Podano metody modelowania pojedynczego obiektu przy pomocy sieci neuronowych oraz podstawowe algorytmy uczenia sieci wielowarstwowych bez sprzężeń zwrotnych. Następnie przedstawiono systemy złożone oraz metody ich modelowania za pomocą sieci neuronowych. Na końcu tego rozdziału podano cel i zakres pracy.

Rozdział drugi zawiera modele obiektów złożonych o strukturze szeregowej, algorytmy uczenia sieci wielowarstwowych wykorzystanych do modelowania globalnego oraz modelowania globalnego z uwzględnieniem jakości modeli lokalnych.

W rozdziale trzecim przedstawiono wyniki modelowania pewnego statycznego nieliniowego obiektu złożonego o jednym wejściu i jednym wyjściu. Symulacje wykorzystują przedstawione wcześniej metody modelowania przy zastosowaniu kilku wybranych struktur sieci neuronowych.

W rozdziale czwartym przedstawiono wyniki modelowania fragmentu procesu produkcji azotynu amonu z Zakładów Azotowych w Tarnowie - Mościcach. Zaproponowano kilka wariantów modeli tego procesu.

Pracę kończy rozdział podsumowujący dokonania autora i kierunki dalszych badań. Omawiane w pracy metody modelowania, algorytmy wyznaczania parametrów modeli obiektów złożonych, wyniki badań symulacyjnych zostały częściowo zaprezentowane we wcześniejszych publikacjach autora [10, 11, 12].

1.2 Sieci neuronowe

Sieci neuronowe są wykorzystywane do rozwiązywania szeregu złożonych zadań, ponieważ posiadają wiele pożytecznych właściwości:

- zdolność uczenia się,
- zdolność zapamiętywania i uogólniania (predykcji),
- możliwość aproksymacji funkcji nieliniowych z dowolną dokładnością,
- możliwość odwzorowania funkcji wektorowych wielu zmiennych,
- sieci neuronowe mogą przetwarzać dane w postaci numerycznej i mogą doskonalić swoje działanie zarówno dla danych, które w procesie uczenia są dostępne na raz lub są dostarczane sukcesywnie.

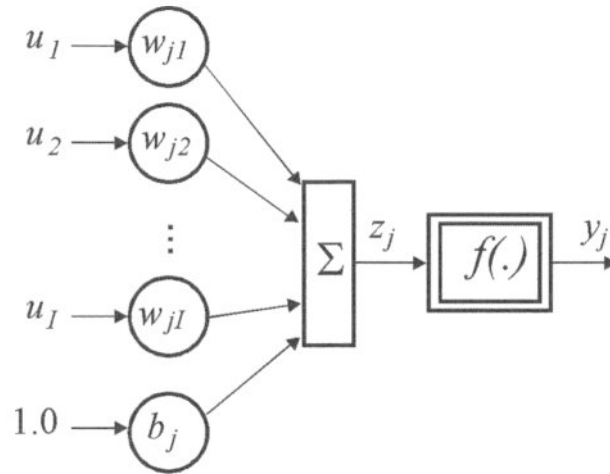
Zastosowanie sieci neuronowych w modelowaniu i identyfikacji wynika przede wszystkim z możliwości aproksymacji „dowolnych” nieliniowości (por. p.1.4.2 niniejszej pracy) oraz dostrajania parametrów na podstawie danych pomiarowych, względnie danych uczących pochodzących z innych źródeł.

1.2.1 Opis sztucznego neuronu

Podstawowym elementem sieci neuronowej jest neuron. Większość modeli neuronów oparta jest na modelu neuronu autorstwa McCullocha i Pittsa [32]. Na jego podstawie można zbudować inne modele neuronów takie, jak Perceptron [37, 45], Adaline, Madaline i inne [21, 31, 57, 58]. Neuron zdefiniowany jest przez kompozycję dwóch elementów [25]:

- ważona suma wejść,

- nieliniowa funkcja przejścia (inaczej - funkcja aktywacji).



Rys. 1.1. Podstawowy model sztucznego neuronu

Rysunek 1.1 przedstawia model pojedynczego neuronu. Sygnały wejściowe u_i ($i = 1, 2, \dots, I$) są sumowane z odpowiednimi wagami w_{ji} a do utworzonej sumy dodawane jest przesunięcie b_j . Wynik sumowania opisany jest następująco:

$$z_j = \sum_{i=1}^I w_{ji} u_i + b_j \quad (1.1)$$

gdzie: I –liczba wejść neuronu. Sygnał wyjściowy neuronu y_j jest odpowiedzią funkcji przejścia $f(\cdot)$ na sygnał z_j .

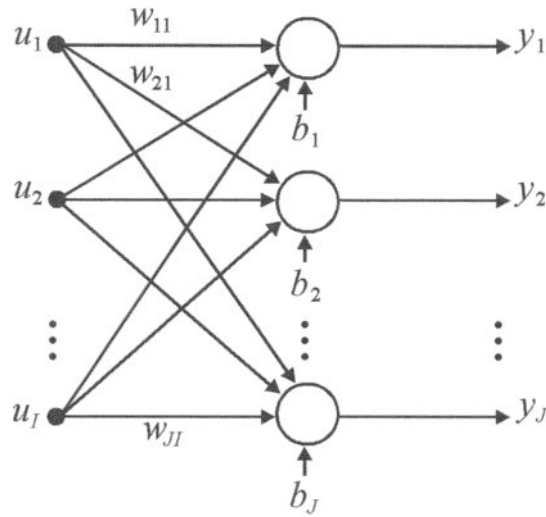
$$y_j = f(z_j) \quad (1.2)$$

1.2.2 Sieć jednowarstwowa jednokierunkowa

Sztuczna sieć neuronowa jest zbudowana z wzajemnie połączonych neuronów. Nieliniowa część neuronu może być wybierana w zależności od potrzeb. Zwykle dla nieliniowej funkcji przejścia wybiera się funkcję tangens hiperboliczny lub funkcję sigmoidalną, które są funkcjami różniczkowalnymi. Stosuje się również funkcje skokowe, które z kolei nie są różniczkowalne. W większości sieci prezentowanych w literaturze, dokonywane są uproszczenia poprzez wybieranie neuronów tego samego typu w tych samych warstwach.

Dla J statycznych neuronów "zasilanych" równolegle przez wektor wymuszeń $\mathbf{u} = [u_1 \dots u_I]^T$ i generujących wektor wyjściowy $\mathbf{y} = [y_1 \dots y_J]^T$, tworzy się jedną warstwę

sieci, poprzez połączenia jednokierunkowe (rys.1.2). Ta jedna warstwa sieci realizuje statyczne odwzorowanie nieliniowe wiążące wejście u z wyjściem y .



Rys.1.2. Jednowarstwowa sieć z połączeniami jednokierunkowymi

Może ona być opisana za pomocą dwóch odwzorowań (nieliniowego i liniowego):

$$y = \Phi(z) \quad (1.3)$$

i

$$z = w \cdot u + b \quad (1.4)$$

lub krótko:

$$y = \Phi(w \cdot u + b)$$

gdzie w jest macierzą wag o wymiarach $J \times I$ (J – liczba neuronów, I – liczba sygnałów wejściowych):

$$w = \begin{bmatrix} w_{11} & w_{12} & \cdots & w_{1I} \\ \vdots & \vdots & \ddots & \vdots \\ w_{J1} & w_{J2} & \cdots & w_{JI} \end{bmatrix} \quad (1.5)$$

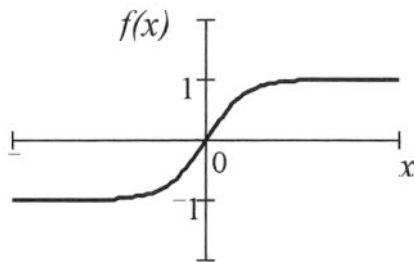
i funkcja wektorowa

$$\Phi = \text{diag}(f_1(z_1), \dots, f_J(z_J)) \quad (1.6)$$

1.2.3 Sieć wielowarstwowa jednokierunkowa

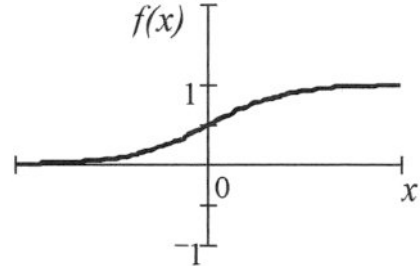
Sieci wielowarstwowe tworzone są przez połączenie sieci jednowarstwowych. Sieć wielowarstwowa posiada co najmniej dwie warstwy neuronów. Występują tu połączenia pomiędzy warstwami neuronów typu każdy z każdym. Neurony, które są użyte w

wielowarstwowych sieciach neuronowych są z reguły typu McCullocha-Pittsa i posiadają nieliniowe funkcje aktywacji. Funkcje te są ciągłe, monotonicznie rosnące, ograniczone i najczęściej różniczkowalne. Opis takich funkcji jest następujący:



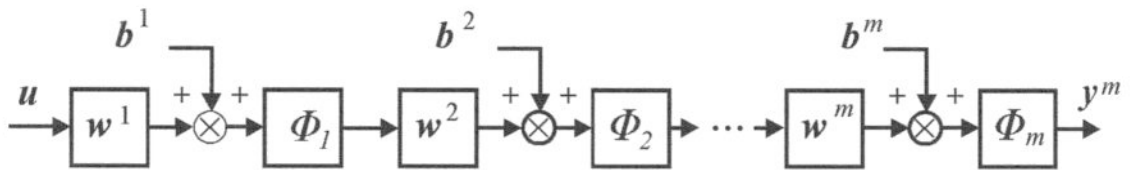
$$f(x) = \frac{1 - e^{-x}}{1 + e^{-x}} \quad (1.7)$$

Rys. 1.3. Funkcja tangens hiperboliczny



$$f(x) = \frac{1}{1 + e^{-x}} \quad (1.8)$$

Rys. 1.4. Funkcja sigmoidalna



Rys. 1.5 Struktura wielowarstwowej sieci neuronowej

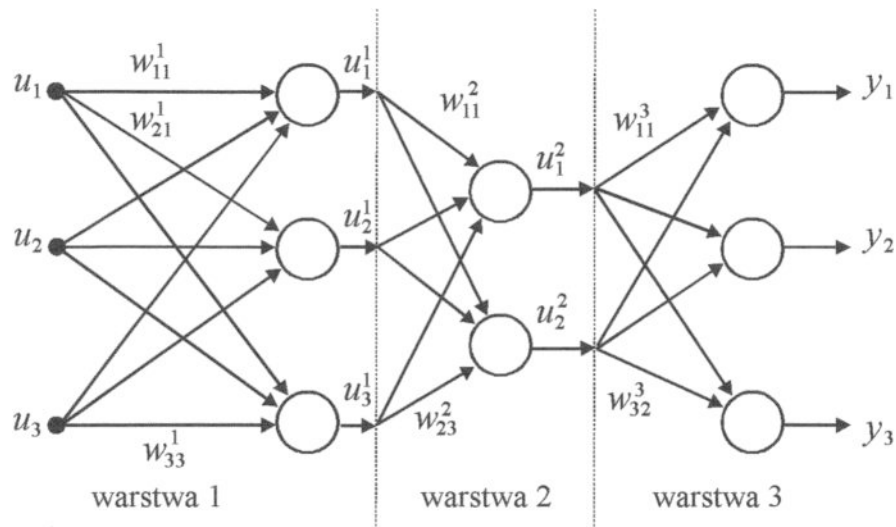
Każda m -ta warstwa sieci neuronowej zawiera macierz wag w^m , nieliniowy operator Φ_m i wektor przesunięć b^m . Wyjściem warstwy y^m jest odpowiedź na wektor wejściowy u^m

$$y^m(u^m) = \Phi_m(w^m u^m + b^m) \quad (1.9)$$

Wyjście to może być opisane przez złożenie m -funkcji:

$$y^m = \Phi_m(w^m \Phi_{m-1}(w^{m-1} \dots \Phi_1(w^1 u + b^1) + \dots + b^{m-1}) + b^m) \quad (1.10)$$

gdzie y^m jest wektorem wyjściowym sieci. Na rysunku 1.5 przedstawiony jest ogólny schemat sieci m -warstwowej.



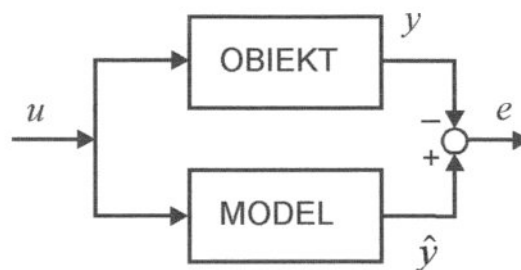
Rys. 1.6 Przykład trójwarstwowej sieci neuronowej z połączeniami jednokierunkowymi

Przykład trójwarstwowej sieci neuronowej pokazano na rysunku 1.6. Pierwsza warstwa zawiera trzy neurony. Wyjściowy wektor warstwy wejściowej jest równy $\mathbf{u}^1 = [u_1^1 \ u_2^1 \ u_3^1]^T$. Stanowi on wejście dla następnej warstwy (zwanej *ukrytą*), która z kolei składa się z dwóch neuronów. Trzecia warstwa zwana wyjściową generuje wyjściowy wektor sieci $\mathbf{y} = [y_1 \ y_2 \ y_3]^T$. Wyjściowa warstwa jest zasilana przez wektor $\mathbf{u}^2 = [u_1^2 \ u_2^2]^T$. Jeżeli przyjmiemy, że wektory przesunięć we wszystkich warstwach $\mathbf{b}^1, \mathbf{b}^2, \mathbf{b}^3$ są równe zero, wówczas sieć realizuje odwzorowanie nieliniowe wejścia $\mathbf{u}$ na wyjście $\mathbf{y}$ w następujący sposób:

$$\mathbf{y} = \Phi_3(\mathbf{w}^3 \Phi_2(\mathbf{w}^2 \Phi_1(\mathbf{w}^1 \cdot \mathbf{u})))$$

gdzie: Φ_1, Φ_2, Φ_3 i $\mathbf{w}^1, \mathbf{w}^2, \mathbf{w}^3$ są odpowiednio wektorowymi funkcjami aktywacji i macierzami wagowymi poszczególnych warstw.

1.3 Modelowanie pojedynczego elementu obiektu



Rys. 1.7 Modelowanie obiektu

Rysunek 1.7 przedstawia ogólną zasadę modelowania. Dla danego obiektu tworzymy optymalny model, którym może być sieć neuronowa.

Zadanie modelowania polega na takim doborze parametrów modelu (wag, przesunięć) aby ocena różnicy między wyjściem modelu $\bar{y}$ a wyjściem obiektu y była minimalna. Różnica e jest wektorem:

$$e = \bar{y} - y \quad (1.11)$$

W przypadku K -elementowego zestawu danych uczących wektory wejściowe $u^1, u^2, \dots, u^K$ można zapisać w postaci macierzy $u = [u^1 u^2 \dots u^K]$. Na dany wektor wejściowy u^k odpowiedzią obiektu jest y^k , zatem wektory odpowiedzi tworzą macierz $y = [y^1 y^2 \dots y^K]$. Podobnie wektory wyjściowe sieci neuronowej można zapisać w postaci macierzy $\bar{y} = [\bar{y}^1 \bar{y}^2 \dots \bar{y}^K]$.

Problem sprowadza się do wyznaczenia wartości parametrów sieci neuronowej, które minimalizują wskaźnik jakości Q dla pełnego zestawu uczącego K :

$$Q = \frac{1}{2} (e)^T e = \frac{1}{2} \sum_{k=1}^K (\bar{y}^k - y^k)^2 \quad (1.12)$$

1.3.1 Uczenie sieci neuronowych

W tej pracy celem procesu uczenia jest taki dobór parametrów (wag) sieci neuronowej, który umożliwi dostosowanie działania sieci do postawionego zadania modelowania obiektu, określonego w postaci odpowiednich wymagań co do odwzorowania danych wejściowych w dane wyjściowe. Dla pojedynczej wagi w_{ji} zmiana ta może być zapisana następująco:

$$w_{ji}(l+1) = w_{ji}(l) + \Delta w_{ji}(l) \quad (1.13)$$

gdzie l oznacza kolejny numer cyklu uczenia ($l=0, 1, 2, \dots$).

Sposób wyznaczenia przyrostu $\Delta w_{ji}(l)$ zależy od wybranej metody uczenia. W niniejszej pracy została wybrana metoda uczenia pod nadzorem (z nauczycielem), która zakłada, że każdym wejściowym wartościom sygnałów uczących towarzyszą pożądane wartości sygnałów na wyjściu sieci.

1.3.2 Uczenie pod nadzorem sieci jednowarstwowej

Rozważmy obecnie sieć jednowarstwową o nieliniowych funkcjach aktywacji (rys.1.2). W uczeniu pod nadzorem dla danego k -tego wektora wejściowego $\mathbf{u}^k = [u_1^k \ u_2^k \ \dots \ u_I^k]^T$ znany jest wektor odpowiedzi obiektu $\mathbf{y}^k = [y_1^k \ y_2^k \ \dots \ y_J^k]^T$. Sieć na wektor wejściowy $\mathbf{u}^k$ odpowiada wektorem wyjściowym $\bar{\mathbf{y}}^k = [\bar{y}_1^k \ \bar{y}_2^k \ \dots \ \bar{y}_J^k]^T$. Różnica między wektorem wyjściowym modelu i obiektu stanowi wektor błędu:

$$\mathbf{e}^k = \bar{\mathbf{y}}^k - \mathbf{y}^k \quad \text{dla } k = 1, 2, \dots, K \quad (1.14)$$

Celem uczenia jest uzyskanie najmniejszej normy wektora błędu poprzez minimalizację funkcji celu.

Dla wszystkich próbek uczących K funkcja celu $Q(\mathbf{w})$ najczęściej przyjmuje postać:

$$Q(\mathbf{w}) = \sum_{k=1}^K Q^k = \frac{1}{2} \sum_{k=1}^K \sum_{j=1}^J (\bar{y}_j^k - y_j^k)^2 \quad (1.15)$$

gdzie

$$Q^k = \frac{1}{2} (\mathbf{e}^k)^T \mathbf{e}^k = \frac{1}{2} \sum_{j=1}^J (\bar{y}_j^k - y_j^k)^2$$

Dla k -tej próbki uczącej dla j -tego neuronu wyjściowego błąd jest równy:

$$e_j^k = \bar{y}_j^k - y_j^k = f(z_j^k) - y_j^k = f\left(\sum_{i=0}^I w_{ji} u_i^k\right) - y_j^k \quad (1.16)$$

gdzie: y_j^k , $\bar{y}_j^k$, z_j^k oznaczają odpowiednio wyjście obiektu i wyjście j -tego neuronu w sieci oraz sumę ważoną wejść wyznaczoną w jego sumatorze przy prezentacji k -tej pary uczącej; u_i^k jest i -tą składową wektora wejściowego, w_{ji} oznacza wagę połączenia pomiędzy j -tym neuronem warstwy wyjściowej i i -tym wejściem; I - liczba wejść oraz $w_{j0}^k = b_j^k$, $u_0^k = 1$.

Zadanie uczenia sieci polega na minimalizacji funkcji celu Q . Jedną z najbardziej popularnych metod minimalizacji funkcji celu jest gradientowa metoda największego spadku. Jest to metoda iteracyjna, która poszukuje kolejnego lepszego punktu w kierunku przeciwnym do gradientu funkcji celu. Stosując powyższą metodę do uczenia sieci, zmiana wag odbywa się według wzoru:

$$\Delta \mathbf{w} = -\eta \nabla Q(\mathbf{w}) \quad (1.17)$$

gdzie η jest ustaloną liczbą z przedziału $(0,1)$, a $\nabla Q(\mathbf{w})$ jest gradientem funkcji celu Q .

Dla pełnego zbioru uczącego K , ograniczając się do wagi w_{ji} otrzymujemy:

$$\Delta w_{ji} = -\eta \frac{\partial Q}{\partial w_{ji}} = -\eta \sum_{k=1}^K \frac{\partial Q^k}{\partial w_{ji}} \quad (1.18)$$

Ograniczając się do k -tej pary uczącej otrzymujemy:

$$\Delta w_{ji}^k = -\eta \frac{\partial Q^k}{\partial w_{ji}} \quad (1.19)$$

Uwzględniając złożoną zależność Q^k od w_{ji} oraz od sygnałów $\bar{y}_j^k$ i u_i^k możemy zapisać:

$$\Delta w_{ji}^k = \frac{\partial Q^k}{\partial \bar{y}_j^k} \frac{\partial \bar{y}_j^k}{\partial z_j^k} \frac{\partial z_j^k}{\partial w_{ji}} \quad (1.20)$$

Ostatni czynnik wynosi:

$$\frac{\partial z_j^k}{\partial w_{ji}} = u_i^k \quad (1.21)$$

W przypadku nieliniowych neuronów z różniczkowalną funkcją aktywacji mamy:

$$\frac{\partial \bar{y}_j^k}{\partial z_j^k} = \frac{\partial f(z_j^k)}{\partial z_j^k} = f'(z_j^k) \quad (1.22)$$

Pierwszy czynnik jest równy:

$$\frac{\partial Q^k}{\partial \bar{y}_j^k} = \bar{y}_j^k - y_j^k \quad (1.23)$$

Wzór (1.18) ma zatem postać:

$$\Delta w_{ji} = -\eta \sum_{k=1}^K f'(z_j^k) (\bar{y}_j^k - y_j^k) u_i^k \quad (1.24)$$

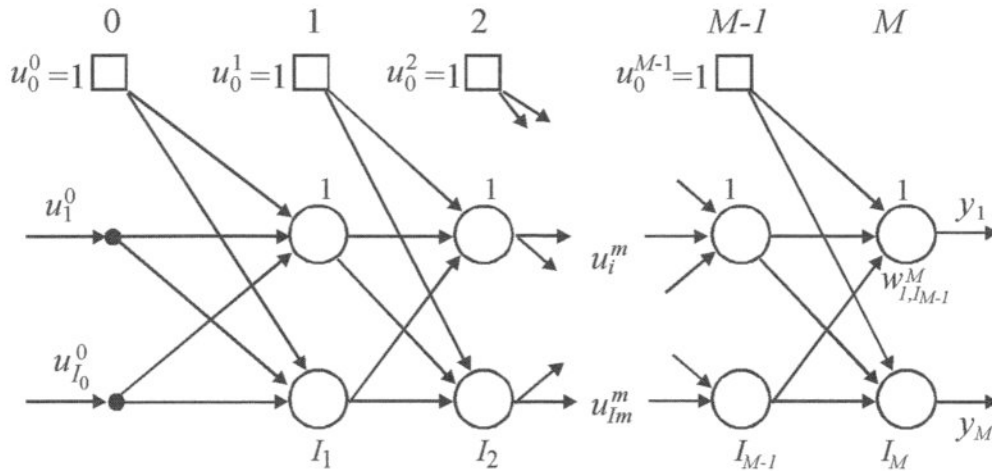
Uaktualnianie wag następuje po zaprezentowaniu wszystkich par uczących (tzw. *skumulowana reguła delta*). Uaktualnianie wag może następować również po prezentacji każdego wzorca, co upraszcza praktyczną realizację algorytmu. Wtedy mamy do czynienia z *regułą delta*:

$$\Delta w_{ji}^k = -\eta f'(z_j^k) (\bar{y}_j^k - y_j^k) u_i^k \quad (1.25)$$

Warto dodać, że reguła delta (1.25) jest alternatywą, a w pewnych przypadkach koniecznością, gdy w momencie rozpoczęcia uczenia nie dysponujemy wszystkimi parami uczącymi.

1.3.3 Uczenie pod nadzorem sieci wielowarstwowej

Do uczenia pod nadzorem sieci wielowarstwowej zostanie użyta metoda wstecznej propagacji błędów, która jest często stosowana i dobrze opisana w literaturze.

Rys. 1.8 Sieć neuronowa M -warstwowa

Rozważmy teraz M -warstwową sieć neuronową o jednakowych niemalejących i różniczkowalnych funkcjach aktywacji (rys.1.8). Jeżeli przez $u_j^{m,k}$ oznaczmy wartość wyjścia i -tego neuronu m -tej warstwy przy prezentacji k -tego zestawu uczącego ($m=1, 2, \dots, M$; $i=0, 1, \dots, I_m$; $k=1, 2, \dots, K$), to wyjście j -tego neuronu jest obliczane na podstawie wzoru:

$$u_j^{m,k} = f(z_j^{m,k}) = f\left(\sum_{i=0}^{I_{m-1}} w_{ji}^m u_i^{m-1,k}\right) \quad (1.26)$$

oraz oznaczenia w warstwie wyjściowej $u_j^{M,k} = \bar{y}_j^k$ i warstwie wejściowej $u_i^{0,k} = u_i^k$ jest i -tym wejściem sieci neuronowej.

Zdefiniujemy błąd „delta” $\delta_j^{m,k}$ powstały w j -tym neuronie m -tej warstwy dla k -tej prezentacji zestawu uczącego ($m=0, 1, \dots, M$; $i=0, 1, \dots, I_m$; $k=1, 2, \dots, K$):

$$\delta_j^{m,k} = \frac{\partial Q}{\partial u_j^{m,k}} \frac{\partial u_j^{m,k}}{\partial z_j^{m,k}} = \frac{\partial Q}{\partial u_j^{m,k}} f'(z_j^{m,k}) \quad (1.27)$$

Dla warstwy wyjściowej zgodnie ze wzorem (1.15) zapiszemy:

$$\frac{\partial Q^k}{\partial u_j^{M,k}} = \frac{\partial Q^k}{\partial \bar{y}_j^k} = \bar{y}_j^k - y_j^k \quad (1.28)$$

Dla warstw ukrytych obliczenie pochodnej cząstkowej jest bardziej złożone. Ponieważ $u_j^{m,k}$ jest jednym ze składników sum ważonych liczonych we wszystkich neuronach warstwy $m+1$ zatem możemy zapisać ($m=1, 2, \dots, M-1$):

$$\begin{aligned}
\frac{\partial Q^k}{\partial u_j^{m,k}} &= \sum_{l=1}^{I_{m+1}} \frac{\partial Q^k}{\partial z_l^{m+1,k}} \frac{\partial z_l^{m+1,k}}{\partial u_j^{m,k}} = \\
&= \sum_{l=1}^{I_{m+1}} \frac{\partial Q^k}{\partial z_l^{m+1,k}} \frac{\partial}{\partial u_j^{m,k}} \sum_{i=1}^{I_m} w_{li}^{m+1,k} u_i^{m,k} \\
&= \sum_{l=1}^{I_{m+1}} \frac{\partial Q^k}{\partial z_l^{m+1,k}} w_{lj}^{m+1} = \sum_{l=1}^{I_{m+1}} \delta_l^{m+1,k} w_{lj}^{m+1}
\end{aligned} \tag{1.29}$$

Stąd na podstawie zależności (1.27), (1.28), (1.29) możemy dla każdego neuronu obliczyć:

- dla warstwy wyjściowej ($m=M$):

$$\delta_j^{M,k} = f'(z_j^{M,k})(\bar{y}_j^k - y_j^k) \tag{1.30}$$

- dla warstw ukrytych ($m=1, \dots, M-1$):

$$\delta_j^{m,k} = f'(z_j^{m,k}) \sum_{l=1}^{I_{m+1}} \delta_l^{m+1,k} w_{lj}^{m+1} \tag{1.31}$$

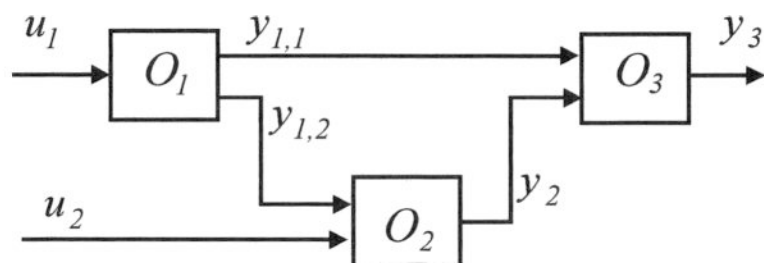
Przy takich definicjach błędu ogólną regułę delty na przyrosty wag możemy zapisać jako:

$$\Delta w_{ji}^{m,k} = -\eta \delta_j^{m,k} u_i^{m-1,k} \tag{1.32}$$

Uaktualnianie wag w regule delty odbywa się w dwóch etapach. W pierwszym etapie na wejście sieci podawany jest jeden z wektorów uczących $\mathbf{u}^k$ i dla każdej warstwy wyznacza się aktualne wartości wyjść wszystkich neuronów zgodnie ze wzorem (1.26). Na koniec oblicza się różnicę między wyjściami neuronów w ostatniej warstwie $\bar{y}_j^k$ a wyjściami z obiektu y_j^k . W drugim etapie na podstawie błędu warstwy wyjściowej $\delta_j^{M,k}$ (1.30) są obliczane błędy warstw ukrytych zgodnie ze wzorem (1.31) począwszy od ostatniej warstwy ukrytej aż do warstwy pierwszej. Po obliczeniu wartości błędu każdego neuronu wyznacza się przyrosty wag połączeń zgodnie ze wzorem (1.32).

1.4. Obiekty złożone

Przez element obiektu będziemy rozumieć taki obiekt, który daje się zamodelować na podstawie obserwacji jego wejść i wyjść. Element obiektu jest częścią składową większego systemu, który będziemy nazywali obiektem złożonym. Jako przykład może posłużyć obiekt przedstawiony na rys. 1.9.



Rys.1.9 Przykład obiektu złożonego

W odróżnieniu od elementów, systemy złożone charakteryzują się tym, że wyodrębnia się w nich oddzielne części składowe, z oddzielnymi, własnymi modelami. Na model złożonego systemu składają się zatem:

- a) Model struktury obiektu, tj. sposób powiązania w całość części składowych systemu złożonego.
- b) Modele poszczególnych części składowych.

Konieczność uzyskania modeli systemów złożonych wymuszona jest potrzebami kompleksowego sterowania złożonymi procesami przemysłowymi. Modele elementów obiektu wystarczą do sterowania procesów stacjonarnych przy zadanych wartościach wyjściowych oraz do optymalizacji statycznej procesów przy określonym wskaźniku jakości. Przy kompleksowym sterowaniu procesami przemysłowymi zachodzi konieczność określenia algorytmów sterowania w różnych fazach procesu (rozruch, praca ustalona, zatrzymanie) oraz w sytuacjach awaryjnych, sterowania skomplikowanym zestawem różnych operacji technologicznych i pomocniczych. Dla wyznaczenia odpowiednich algorytmów sterowania konieczne jest znalezienie modelu sterowanych procesów. Modele elementów opisują fragmenty złożonego systemu w pewnych tylko fazach jego przebiegu.

Problemy ustalania modeli i identyfikacji dla elementów obiektu można już uznać w zasadzie za dostatecznie opracowane, jednak problemy opisu matematycznego złożonych obiektów sterowania nie zostały jeszcze dobrze rozwiązane [3].

1.4.1 Modelowanie obiektów złożonych

W złożonych procesach przemysłowych istnieje duża różnorodność sytuacji, które należy zamodelować. Typowe, najczęściej spotykane sytuacje można sprowadzić do takiego modelu, w którym występują oddzielne części składowe powiązane ze sobą przepływem sygnałów (materiałów i energii), bądź różnymi uwarunkowaniami czasowymi.

Jeśli w złożonym procesie można wyodrębnić poszczególne części zwane także elementami (rys.1.8), w których odbywa się określony proces technologiczny, to powiązania między elementami spowodowane są tym, że materiały lub energia dla określonego elementu mogą

być produktami innych elementów. W efekcie niektóre wielkości wyjściowe wybranego elementu są wielkościami wejściowymi innych elementów. Są to wielkości charakteryzujące materiały (lub energię) przemieszczane pomiędzy elementami.

Innego typu model systemu złożonego otrzymamy traktując poszczególne jego części składowe jako oddzielne *operacje*, tj. określone czynności wykonywane w określonym czasie. Operacjami mogą być oddzielne czynności tzw. dyskretnego procesu technologicznego a także części składowe procesu ciągłego zachodzące w różnych blokach oraz różne fazy procesu zachodzące w tym samym bloku. Złożony proces technologiczny traktujemy zatem jako *kompleks operacji*, którego struktura wynika z odpowiednich następstw czasowych poszczególnych operacji. Sterowanie kompleksem operacji sprowadza się zazwyczaj do odpowiedniego rozdziału zadań i zasobów na poszczególne operacje w czasie realizacji kompleksu. Sterowanie to może być w określonym sensie optymalne, np. może polegać na takim rozdziale ograniczonych zasobów, który minimalizuje czas trwania całego procesu.

Na model matematyczny kompleksu składają się [3]:

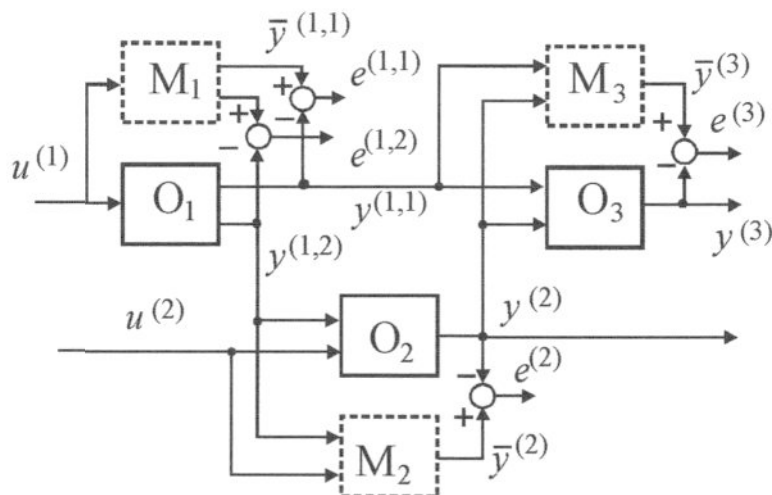
- Modele poszczególnych operacji;
- Model struktury kompleksu.

Modelowanie kompleksu operacji, krótko opisane powyżej, nie jest tematem rozważań niniejszej pracy.

Ustalenie modelu złożonego obiektu polega na określeniu jego struktury oraz na określeniu modeli poszczególnych części składowych. Zadanie pierwsze w przypadku obiektu złożonego z elementów polega na wyodrębnieniu poszczególnych elementów procesu i ustaleniu ich wzajemnych powiązań. Dla złożonych procesów taka ich wstępna charakterystyka, a także ustalenie celów sterowania i grup wielkości charakterystycznych dla części składowych może być zadaniem bardzo trudnym, wymagającym dobrej znajomości i dokładnej analizy rozpatrywanego procesu. Nie jest to jeszcze zadanie identyfikacji, lecz wstępny etap pracy polegający na zastosowaniu metod *analizy systemowej* do określenia struktury procesu sterowanego.

Następny etap polega na określeniu modeli poszczególnych elementów w wyniku ich identyfikacji. W ten sposób zadanie identyfikacji obiektu złożonego sprowadza się w zasadzie do zadania identyfikacji poszczególnych elementów, na jakie został podzielony obiekt złożony. Występuje tu jednak nowy problem związany z faktem, że obecnie nie mamy do czynienia z pojedynczym elementem obiektu, lecz z obiektem wyodrębnionym z pewnego systemu. Modele tych wyodrębnionych obiektów powinny być takie, aby w rezultacie

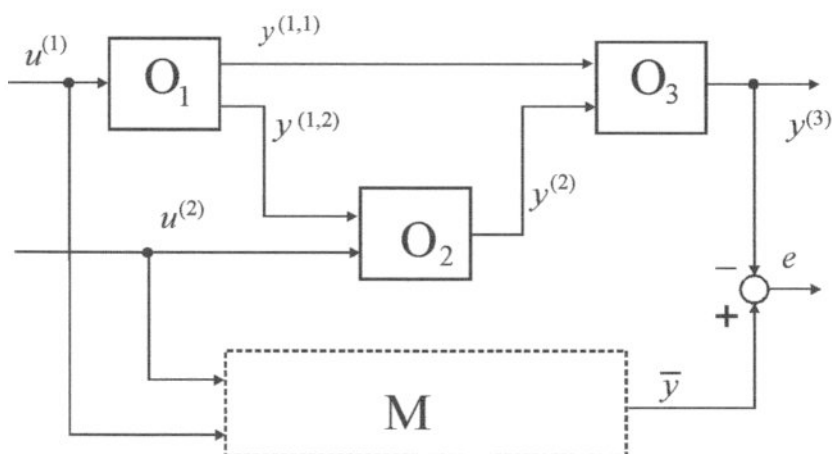
najlepszy był model (neuronowy) całego systemu. Wiąże się to z *oceną lokalną* jakości modelu elementu składowego i *oceną globalną* obiektu złożonego [3].



Rys. 1.10. Przykład obiektu złożonego z modelami jego elementów składowych

Konstruując modele obiektu złożonego można sformułować wielorakie kryteria oceny jakości tych modeli.

Niezależnie od struktury obiektu można zaproponować modele elementów obiektu (rys. 1.10), dla których oceniamy różnice między wyjściem modelu a wyjściem elementu obiektu osobno i nazywamy ją identyfikacją lokalną.

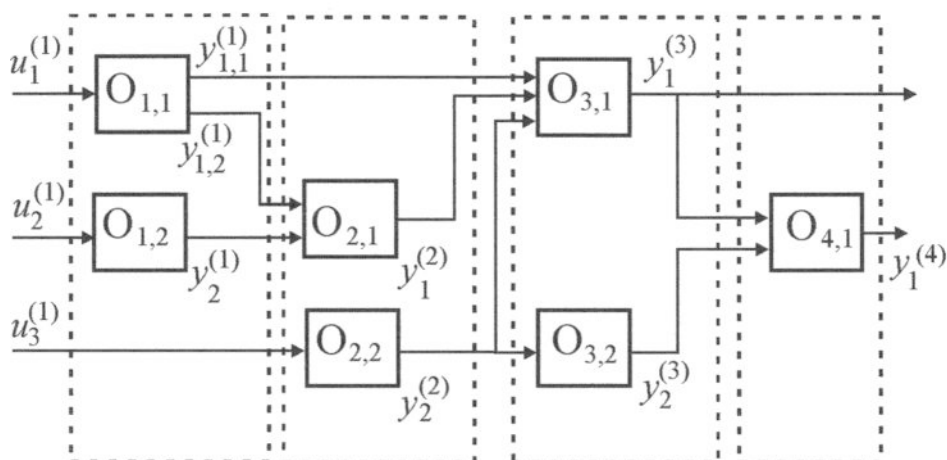


Rys. 1.11 Przykład obiektu złożonego z modelem globalnym

Z drugiej strony możemy sformułować model globalnie optymalny, dla którego oceniamy różnice pomiędzy wybranymi wyjściami obiektu złożonego dla zadanych wejść zewnętrznych, a odpowiednimi odpowiedziami wyjściowymi modelu złożonego (rys. 1.11).

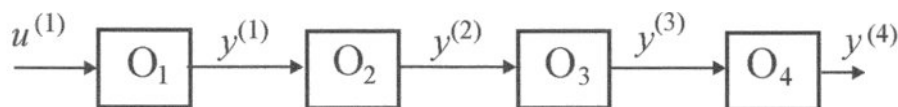
W praktycznych przypadkach modele lokalnie optymalne jak i modele globalnie optymalne mogą być przydatne. W zależności od potrzeb możemy używać jednego lub drugiego.

Wielokryterialna ocena prowadzi do różnych sformułowań optymalizacyjnych. Możemy sformułować na bazie ocen jakości modeli lokalnych i modeli globalnych wypadkowe kryterium, które doprowadzi do powstania modelu w pewnym stopniu globalnie optymalnego, równocześnie modele lokalne będą również w pewnym stopniu zoptymalizowane.



Rys.1.12 Przykład obiektu złożonego bez sprzężeń zwrotnych

Jeżeli w obiekcie złożonym o dowolnej strukturze, np. jak na rys.1.12, nie występują sprzężenia zwrotne, to taki obiekt można sprowadzić do obiektu o strukturze szeregowej, jak na rysunku 1.13. W takiej strukturze wejściem do kolejnego elementu obiektu jest wyjście z obiektu poprzedniego. Struktura szeregowa jest strukturą podstawową. Występuje w wielu różnorodnych procesach przemysłowych. W niniejszej pracy będziemy rozważać obiekty złożone tylko o strukturze szeregowej.



Rys.1.13. Przykład obiektu złożonego o strukturze szeregowej

1.4.2 Modelowanie systemów złożonych za pomocą sieci neuronowych

W tej sekcji zajmiemy się zadaniem modelowania obiektu złożonego statycznego, szeregowego. Przyjmujemy, że modelami obiektu złożonego będą modele w postaci sieci

neuronowych wielowarstwowych bez sprzężeń zwrotnych. Prace na temat użycia sieci neuronowych w procesie modelowania, identyfikacji i sterowania przyciągnęły dużą uwagę. Zaprojektowanie odpowiedniego schematu sterowania złożonego procesu, przy braku modelu tego procesu lub istnieniu modelu aproksymującego jest bardzo trudne. Brak modeli może wynikać z niedostatecznej wiedzy o procesie. Zdolności uczenia sieci można wykorzystać w sytuacji, gdy nasza znajomość procesu nie jest dogłębna.

W oparciu o twierdzenie Stone’a-Weierstrass’a [6] i twierdzenie Kolmogorova [28] Cybenko [8], Funahashi [17] i Hornik, Stinchcombe i White [23, 24] wykazali, że sieci neuronowe są zdolne aproksymować z dowolną dokładnością praktycznie *każde odwzorowanie wejście-wyjście*.

Cybenko udowodnił, że sieć neuronowa jest uniwersalnym aproksymatorem w sensie normy maksymalnej, którą definiuje się następująco:

$$\|f\| = \sup_{x \in X} |f(X)| \quad (1.33)$$

gdzie f jest funkcją ciągłą określoną na $X = [0,1]^L$, L – liczba wejść sieci.

Twierdzenie 1 [36] Dla dowolnej ciągłej funkcji $g: R^L \rightarrow R^M$ i dowolnie małego $\varepsilon > 0$ istnieje funkcja f będąca wyjściem trójwarstwowej sieci neuronowej o jednej warstwie ukrytej, L -neuronach wejściowych i M -neuronach wyjściowych taka, że

$$\|g - f\| < \varepsilon \quad (1.34)$$

Twierdzenie 2 [22] Jeżeli funkcja aktywacji neuronów jest ciągła, ograniczona i różna od stałej, to dla podzbioru zawartego w $X \subseteq R^n$, wielowarstwowa jednokierunkowa sieć neuronowa może aproksymować dowolną funkcję ciągłą określoną na podzbiorku X z dowolną dokładnością pod warunkiem, że warstwa ukryta posiada wystarczającą liczbę neuronów.

W związku z tym, sieci neuronowe mogą być zastosowane do identyfikacji statycznych odwzorowań nieliniowych. Zalety modelowania przy użyciu wielowarstwowych sieci neuronowych stają się jeszcze bardziej widoczne w tych sytuacjach, gdzie funkcja wejście-wyjście, której sieć jest uczona, nie jest ograniczona do ściśle określonej jednej klasy funkcji. Wektor wejściowy jak również wektor wyjściowy sieci może zawierać takie rodzaje informacji jak wartości sygnału sterującego, wartości sygnałów procesu, dane historyczne, dane z czujników, itp.

Na podstawie pojęć podanych w poprzedniej sekcji dotyczących modeli lokalnych, modeli globalnych oraz uwzględniając wielokryterialne oceny modelu złożonego, skupiając się na obiektach złożonych o strukturze szeregowej proponuje się następujące modele: lokalnie optymalny, globalnie optymalny oraz globalnie optymalny z uwzględnieniem jakości modeli lokalnych. Uściślenie tych pojęć nastąpi w rozdziale drugim.

1.5 Cel i zakres pracy

Celem pracy jest opracowanie metod i narzędzi tworzenia modeli przy wykorzystaniu sieci neuronowych dla obiektów złożonych, statycznych, o strukturze szeregowej. W szczególności zwraca się uwagę na modele lokalnie optymalne, modele globalnie optymalne oraz globalnie optymalne z uwzględnieniem jakości modeli lokalnych. Opracowana metodyka tworzenia modeli lokalnych i globalnych powinna być przydatna do projektowania komputerowych systemów sterowania systemów złożonych.

Zakres pracy

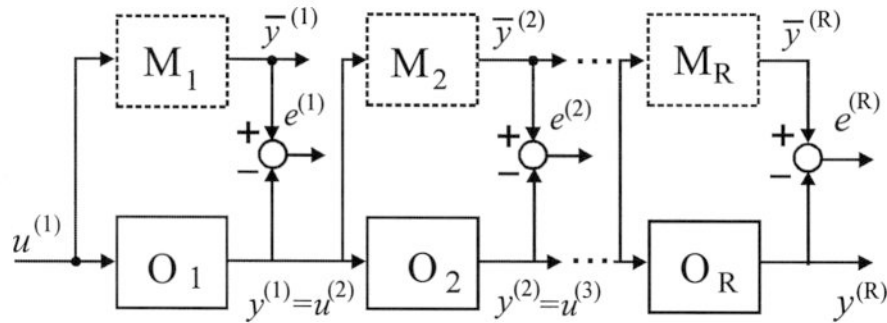
1. Opracowanie metod tworzenia modeli w postaci sieci neuronowych dla obiektów złożonych:
 - model lokalnie optymalny,
 - model globalnie optymalny,
 - model globalnie optymalny z uwzględnieniem jakości modeli lokalnych.
2. Opracowanie algorytmów i programów dla tworzenia modeli opisanych w punkcie 1.
3. Badania symulacyjne oceniające skuteczność opracowanych metod i algorytmów na przykładzie statycznego nieliniowego obiektu złożonego.
4. Opracowanie przykładu praktycznego – modelowanie procesu produkcji azotynu amonu.

2. Modele i algorytmy wyznaczania parametrów modeli obiektów złożonych

W rozdziale tym zostaną wyprowadzone algorytmy uczenia sieci neuronowych modelujących obiekty złożone. Przez parametry sieci będziemy rozumieć wagi neuronów.

2.1 Model lokalnie optymalny

Rozważmy obiekt złożony z R elementów przedstawiony na rysunku 2.1. Dla każdego elementu obiektu tworzony jest osobny model lokalny, nie uwzględniając struktury obiektu złożonego.



Rys. 2.1 Struktura szeregowego obiektu złożonego z uwzględnieniem modeli lokalnych

Modele lokalne opisane są następującymi równaniami:

$$\begin{aligned} \bar{y}^{(1)} &= \Phi_1(u^{(1)}, w^{(1)}) \\ &\vdots \\ \bar{y}^{(r)} &= \Phi_r(u^{(r)}, w^{(r)}) \\ &\vdots \\ \bar{y}^{(R)} &= \Phi_R(u^{(R)}, w^{(R)}) \end{aligned} \quad (2.1)$$

gdzie $w^{(r)}$ jest wektorem parametrów sieci neuronowej r -tego modelu ($r=1, 2, \dots, R$), natomiast $u^{(r)}$, $\bar{y}^{(r)}$ są odpowiednio wejściami i wyjściami modeli lokalnych.

Różnica między wektorem odpowiedzi r -tego modelu lokalnego a wektorem wyjściowym r -tego elementu obiektu stanowi błąd:

$$e^{(r)} = \bar{y}^{(r)} - y^{(r)} \quad (2.2)$$

Modele elementów obiektów nazywamy modelami lokalnymi. Na podstawie błędów między poszczególnymi modelami a obiektami lokalnymi $e^{(r),k} = \left[\bar{y}^{(r),k} - y^{(r),k} \right]$ przy prezentacji k -tej ($k=1, 2, \dots, K$) pary uczącej tworzona jest funkcja kryterialna dla r -tego modelu lokalnego:

$$Q_r^k(\mathbf{w}^{(r)}) = \frac{1}{2} \left(\mathbf{e}^{(r),k} \right)^T \mathbf{e}^{(r),k} \quad (2.3)$$

Dla wszystkich K par uczących:

$$Q_r(\mathbf{w}^{(r)}) = \frac{1}{2} \sum_{k=1}^K \left(\mathbf{e}^{(r),k} \right)^T \mathbf{e}^{(r),k} \quad (2.4)$$

Gdy r -ta sieć neuronowa posiada J_r wyjść, wtedy wzór (2.4) można zapisać w postaci:

$$Q_r(\mathbf{w}^{(r)}) = \frac{1}{2} \sum_{k=1}^K \sum_{j=1}^{J_r} \left(\bar{y}_j^{(r),k} - y_j^{(r),k} \right)^2 \quad (2.5)$$

W celu uzyskania modelu lokalnie optymalnego należy zminimalizować wskaźniki jakości $Q_r(\mathbf{w}^{(r)})$, odrębnie dla każdego elementu systemu. Optymalny zbiór parametrów sieci neuronowej w tym przypadku będzie oznaczony przez $\bar{\mathbf{w}}^{(r)}$

$$\begin{aligned} \bar{\mathbf{w}}^{(1)} &\rightarrow Q_1(\bar{\mathbf{w}}^{(1)}) = \min_{\mathbf{w}^{(1)}} Q_1(\mathbf{w}^{(1)}) \\ \bar{\mathbf{w}}^{(2)} &\rightarrow Q_2(\bar{\mathbf{w}}^{(2)}) = \min_{\mathbf{w}^{(2)}} Q_2(\mathbf{w}^{(2)}) \\ &\vdots \\ \bar{\mathbf{w}}^{(R)} &\rightarrow Q_R(\bar{\mathbf{w}}^{(R)}) = \min_{\mathbf{w}^{(R)}} Q_R(\mathbf{w}^{(R)}) \end{aligned} \quad (2.6)$$

gdzie $\mathbf{w}^{(r)}$ mają współrzędne będące liczbami rzeczywistymi.

Optymalne wartości parametrów modeli lokalnych można zapisać w postaci układu macierzy $\bar{\mathbf{W}} = \left\{ \bar{\mathbf{w}}^{(1)}, \bar{\mathbf{w}}^{(2)}, \dots, \bar{\mathbf{w}}^{(R)} \right\}$.

Po wyznaczeniu optymalnych parametrów (wag) sieci neuronowych modelujących R elementów systemu złożonego o strukturze szeregową otrzymujemy R modeli lokalnie optymalnych.

Do znalezienia optymalnego zbioru parametrów poszczególnych modeli lokalnych stosuje się algorytm wstecznej propagacji błędów opisany w sekcji 1.3. Stosując się do notacji na rys. 2.1 dla r -tego modelu i na podstawie zależności (1.27), (1.28), (1.29), (2.1) oraz (2.5), błędy „delta” są następujące:

- dla warstwy wyjściowej

$$\delta_j^{(r),M,k} = f'(z_j^{M,k})(\bar{y}_j^{(r),k} - y_j^{(r),k}) \quad (2.7)$$

- dla warstw ukrytych

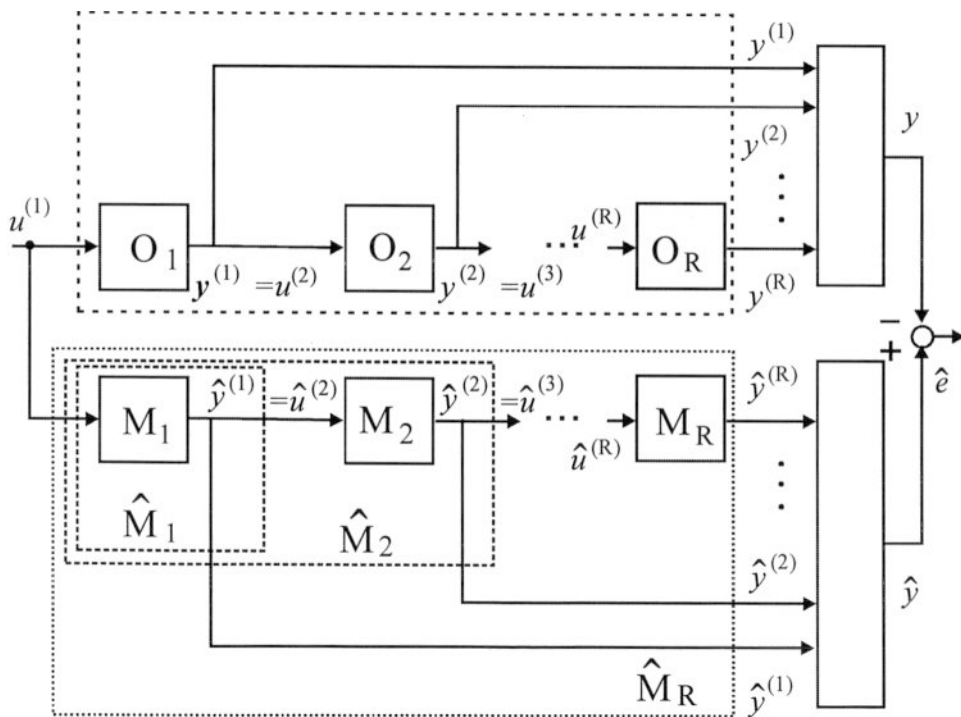
$$\delta_j^{(r),m,k} = f'(z_j^{m,k}) \sum_{l=1}^{I_{m+1}} \delta_l^{(r),m+1,k} w_{lj}^{(r),m+1,k} \quad (2.8)$$

Przy takich definicjach błędu możemy zapisać skumulowaną regułę delty na przyrost parametrów czyli wag w r -tym modelu lokalnym:

$$\Delta w_{ji}^{(r),m} = \sum_{k=1}^K \Delta w_{ji}^{(r),m,k} = -\eta \sum_{k=1}^K \delta_j^{(r),m,k} u_i^{(r),m-1,k} \quad (2.9)$$

2.2 Model globalnie optymalny

Rozważmy obiekt złożony z R elementów przedstawiony na rysunku 2.2. Dla rozważanego obiektu tworzymy model globalny, którego struktura odpowiada strukturze obiektu. W takiej strukturze wyjście aktualnego elementu modelu jest wejściem do następnego elementu modelu.



Rys.2.2 Struktura szeregowego obiektu złożonego oraz jego modelu globalnego

Pojedynczy element modelu opisany jest równaniem:

$$\hat{y}^{(r)} = \Phi_r(\hat{u}^{(r)}, w^{(r)}) \quad (2.10)$$

gdzie $\hat{u}^{(r)} = \hat{y}^{(r-1)}$ wejście r -tego elementu modelu jest wyjściem poprzedniego elementu.

Modele częściowe opisane są następującymi równaniami:

$$\begin{aligned}\hat{y}^{(1)} &= \Phi_1(u^{(1)}, w^{(1)}) = \hat{\Phi}_1(u^{(1)}, w^{(1)}) \\ \hat{y}^{(2)} &= \Phi_2(\hat{y}^{(1)}, w^{(2)}) = \Phi_2(\Phi_1(u^{(1)}, w^{(1)}), w^{(2)}) = \hat{\Phi}_2(u^{(1)}, w^{(1)}, w^{(2)}) \\ &\vdots \\ \hat{y}^{(R)} &= \Phi_R(\hat{y}^{(R-1)}, w^{(R)}) = \Phi_R(\Phi_{R-1}(\dots \Phi_1(u^{(1)}, w^{(1)}), \dots, w^{(R-1)}), w^{(R)}) \\ &= \hat{\Phi}_R(u^{(1)}, w^{(1)}, \dots, w^{(R)})\end{aligned}\quad (2.11)$$

Wektory odpowiedzi obiektu y^k i modelu $\hat{y}^k$ przy k -tej ($k=1, 2, \dots, K$) prezentacji danych uczących będziemy oznaczać przez:

$$y^k = \begin{bmatrix} y^{(1),k} \\ y^{(2),k} \\ \vdots \\ y^{(R),k} \end{bmatrix}, \quad \hat{y}^k = \begin{bmatrix} \hat{y}^{(1),k} \\ \hat{y}^{(2),k} \\ \vdots \\ \hat{y}^{(R),k} \end{bmatrix} \quad (2.12a) \text{ i } (2.12b)$$

Każdy wektor odpowiedzi r -tego obiektu zawiera J_r składowych $y^{(r)} = [y_1^{(r)} y_2^{(r)} \dots y_{J_r}^{(r)}]^T$.

Podając na wejście modelu k -ty wektor wejściowy u^k otrzymujemy odpowiedzi dla r -tego modelu częściowego $\hat{y}^{(r),k} = [\hat{y}_1^{(r),k} \hat{y}_2^{(r),k} \dots \hat{y}_{J_r}^{(r),k}]^T$.

Błąd modelowania globalnego dla k -tej próbki ciągu uczącego ma następującą postać:

$$\hat{e}^k = \hat{y}^{(k)} - y^{(k)} \quad (2.13)$$

Poszukujemy modelu globalnie optymalnego wraz optymalizacją modeli częściowych. Wypadkowe kryterium jakości dla całego ciągu uczącego K elementowego może mieć następującą postać:

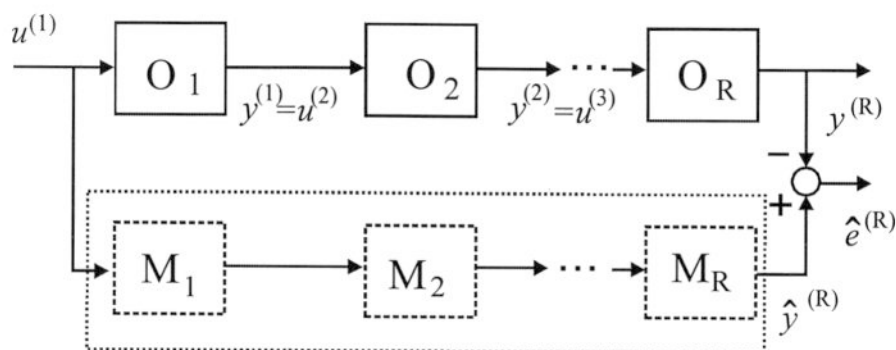
$$Q(W) = \frac{1}{2} \sum_{k=1}^K (\hat{e}^k)^T \beta \hat{e}^k \quad (2.14)$$

gdzie

$$\beta = \begin{bmatrix} \beta_1 & & & 0 \\ & \beta_2 & & \\ & & \ddots & \\ 0 & & & \beta_R \end{bmatrix}$$

oraz $\beta_r \in [0,1]$, $r = 1, \dots, R$ i $\sum_{r=1}^R \beta_r = 1$.

$W = \{w^{(1)}, w^{(2)}, \dots, w^{(R)}\}$ jest zbiorem parametrów sieci neuronowych modelujących elementy częściowe obiektu złożonego. Zbiór W nie musimy porządkować, gdyż ustalenie porządku nie ma wpływu na dalsze obliczenia (nie tworzymy więc jednej macierzy lub wektora). Współczynniki β_r określają wpływ poszczególnych obiektów częściowych na wypadkowe kryterium jakości. Poprzez odpowiedni dobór wartości współczynników możemy określić znaczenie elementów obiektu w modelu systemu złożonego.



Rys.2.3. Struktura obiektu złożonego i jego modelu globalnego - przypadek szczególny

Szczególnym przypadkiem jest model globalnie optymalny bez uwzględnienia modeli częściowych (rys.2.3). Uzyskuje się to przez wyzerowanie współczynników β_r od r równego 1 do $R-1$. Zostaje tylko ostatni współczynnik $\beta_R = 1$. Wtedy wzór (2.14) upraszcza się do postaci:

$$Q(W) = Q_u(W) = \frac{1}{2} \sum_{k=1}^K (\hat{y}^{(R),k} - y^{(R),k})^2 \quad (2.15)$$

Optymalne wartości parametrów modelu $\hat{W}$ uzyskujemy po zminimalizowaniu funkcji kryterialnej:

$$\hat{W} \rightarrow Q(\hat{W}) = \min_W Q(W) \quad (2.16)$$

Algorytm wstecznej propagacji błędu dla kryterium (2.14)

Kryterium (2.14) w postaci sumowań po K wzorcach, R elementach, z których każdy posiada J_r wyjść ma postać:

$$Q(W) = \sum_{k=1}^K Q^k(W) = \frac{1}{2} \sum_{k=1}^K \sum_{r=1}^R \beta_r \sum_{j=1}^{J_r} (\hat{y}_j^{(r),k} - y_j^{(r),k})^2 \quad (2.17)$$

Ograniczając się do k -tej pary uczącej możemy zdefiniować następującą funkcję błędu:

$$\begin{aligned} Q^k(W) = & \frac{1}{2} \beta_1 \sum_{j=1}^{J_1} \left(\hat{y}_j^{(1),k} - y_j^{(1),k} \right)^2 + \dots \\ & + \frac{1}{2} \beta_r \sum_{j=1}^{J_r} \left(\hat{y}_j^{(r),k} - y_j^{(r),k} \right)^2 + \dots + \frac{1}{2} \beta_R \sum_{j=1}^{J_R} \left(\hat{y}_j^{(R),k} - y_j^{(R),k} \right)^2 \end{aligned} \quad (2.18)$$

Poszukiwanie minimum funkcji błędu $Q^k(W)$ może być dokonane podobnie jak w rozdziale pierwszym metodą gradientową. Stosując tę metodę do uczenia sieci, przyrost wagi jest równy:

$$\Delta w_{ji}^k = -\eta \frac{\partial Q^k(W)}{\partial w_{ji}} \quad (2.19)$$

gdzie: η jest liczą z przedziału $(0,1)$ zwaną też współczynnikiem uczenia.

Rozważmy M -warstwową sieć o niemalejących i różniczkowalnych funkcjach aktywacji. Jeżeli przez $\hat{u}_i^m$ oznaczymy wartość wyjścia i -tego neuronu m -tej warstwy przy prezentacji k -tego wzorca ($m=0, 1, \dots, M$; $i=0, 1, \dots, I_m$), to wyjście j -tego neuronu jest obliczane na podstawie wzoru:

$$\hat{u}_j^{m,k} = f(z_j^{m,k}) = f\left(\sum_{i=0}^{I_{m-1}} w_{ji}^{(r),m,k} \hat{u}_i^{(m-1),k}\right) \quad (2.20)$$

Zdefiniujemy błąd „delta” $\delta_j^{(r),m,k}$ powstały w j -tym elemencie m -tej warstwy wchodzącej w skład r -tego elementu modelu:

$$\delta_j^{(r),m,k} = \frac{\partial Q^k}{\partial z_j^{m,k}} = \frac{\partial Q^k}{\partial \hat{u}_j^{m,k}} \frac{\partial \hat{u}_j^{m,k}}{\partial z_j^{m,k}} = \frac{\partial Q^k}{\partial \hat{u}_j^{m,k}} f'(z_j^{m,k}) \quad (2.21)$$

Dla warstwy wyjściowej zgodnie ze wzorem (2.18) zapiszemy:

$$\frac{\partial Q^k(W)}{\partial \hat{u}_j^{M,k}} = \beta_R \frac{\partial Q^k(W)}{\partial \hat{y}_j^{(R),k}} = \beta_R \left(\hat{y}_j^{(R),k} - y_j^{(R),k} \right) \quad (2.22)$$

Dla warstw ukrytych obliczenie pochodnej cząstkowej jest bardziej złożone. Musimy podzielić sieć na „normalne” warstwy ukryte oraz „dodatkowe” warstwy ukryte.

Warstwy ukryte „dodatkowe” to są te warstwy, w których wyjścia są wyjściami modeli częściowych. Różnice między wyjściami „dodatkowych” warstw ukrytych a wyjściami odpowiednich elementów obiektu złożonego są błędami, które wchodzą w skład wskaźników jakości modeli.

„Normalne” warstwy ukryte są to warstwy, których wyjścia są wejściami kolejnych warstw - nie są one wykorzystywane poza siecią.

W „normalnych” warstwach ukrytych $\hat{u}_j^{m,k}$ jest jednym ze składników sum ważonych liczonych we wszystkich neuronach warstwy $m+1$, zatem możemy zapisać:

$$\begin{aligned} \frac{\partial Q^k(W)}{\partial \hat{u}_j^{m,k}} &= \sum_{l=1}^{I_{m+1}} \frac{\partial Q^k(W)}{\partial z_l^{m+1,k}} \frac{\partial z_l^{m+1,k}}{\partial \hat{u}_j^{m,k}} = \\ &= \sum_{l=1}^{I_{m+1}} \frac{\partial Q^k(W)}{\partial z_l^{m+1,k}} \frac{\partial}{\partial \hat{u}_j^{m,k}} \sum_{i=1}^{I_m} w_{li}^{(r),m+1} \hat{u}_i^{m,k} \\ &= \sum_{l=1}^{I_{m+1}} \frac{\partial Q^k(W)}{\partial z_l^{m+1,k}} w_{lj}^{(r),m+1} = \sum_{l=1}^{I_{m+1}} \delta_l^{(r),m+1,k} w_{lj}^{(r),m+1} \end{aligned} \quad (2.23)$$

W „dodatkowych” warstwach ukrytych $\hat{u}_j^{m,k} = \hat{y}_j^{(r),k}$ jest jednym ze składników sum ważonych, liczonych we wszystkich neuronach warstwy $m+1$, zatem możemy zapisać:

$$\begin{aligned} \frac{\partial Q^k(W)}{\partial \hat{u}_j^{m,k}} &= \sum_{l=1}^{I_{m+1}} \frac{\partial Q^k(W)}{\partial z_l^{m+1,k}} \frac{\partial z_l^{m+1,k}}{\partial \hat{u}_j^{m,k}} + \frac{\beta_r}{2} \frac{\partial (\hat{u}_j^{m,k} - u_j^{m,k})^2}{\partial \hat{u}_j^{m,k}} \\ &= \sum_{l=1}^{I_{m+1}} \frac{\partial Q^k(W)}{\partial z_l^{m+1,k}} \frac{\partial}{\partial \hat{u}_j^{m,k}} \sum_{i=1}^{I_m} w_{li}^{(r+1),m+1} \hat{u}_i^{m,k} + \beta_r (\hat{u}_j^{m,k} - u_j^{m,k}) \\ &= \sum_{l=1}^{I_{m+1}} \delta_l^{(r+1),m+1,k} w_{lj}^{(r+1),m+1} + \beta_r (\hat{y}_j^{(r),k} - y_j^{(r),k}) \end{aligned} \quad (2.24)$$

Stąd na podstawie zależności (2.22), (2.23), (2.24) możemy dla każdego neuronu obliczyć błąd:

- dla warstwy wyjściowej:

$$\delta_j^{(R),M,k} = \beta_R f'(z_j^{M,k}) (\bar{y}_j^{(R),k} - y_j^{(R),k}) \quad (2.25)$$

- dla warstw ukrytych normalnych:

$$\delta_j^{(r),m,k} = f'(z_j^{m,k}) \sum_{l=1}^{I_{m+1}} \delta_l^{(r),m+1,k} w_{lj}^{(r),m+1} \quad (2.26)$$

- dla warstw ukrytych dodatkowych:

$$\delta_j^{(r),m,k} = f'(z_j^{m,k}) \sum_{l=1}^{I_{m+1}} \delta_l^{(r+1),m+1,k} w_{lj}^{(r+1),m+1} + \beta_r f'(z_j^{m,k}) (\hat{y}_j^{(r),k} - y_j^{(r),k}) \quad (2.27)$$

Przy takich definicjach błędu możemy zapisać ogólną regułę delty na przyrosty wag:

$$\Delta w_{ji}^{(r),m} = \sum_{k=1}^K \Delta w_{ji}^{(r),m,k} = -\eta \sum_{k=1}^K \delta_j^{(r),m,k} \hat{u}_i^{m-1,k} \quad (2.28)$$

Przy uczeniu sieci neuronowych można nie używać informacji o wartości gradientu, lecz tylko o jego znaku, co pozwala na znaczne przyspieszenie procesu uczenia [40], w literaturze

algorytm ten znany jest pod nazwą *Rprop* [44]. Wówczas wzory na przyrost parametrów modelu w kolejnych cyklach (l) uczenia są następujące [11]:

$$\Delta w_{ji}^{(r),m,k} = -\eta_{ji}^l \operatorname{sgn}(\delta_j^{(r),m,k} \hat{u}_i^{m-1,k}) \quad (2.29)$$

gdzie

$$\eta_{ji}^l = \begin{cases} \min(a \cdot \eta_{ji}^{(l-1)}, \eta_{\max}^l) & \text{dla } S_{ji}^l \cdot S_{ji}^{(l-1)} > 0 \\ \max(b \cdot \eta_{ji}^{(l-1)}, \eta_{\min}^l) & \text{dla } S_{ji}^l \cdot S_{ji}^{(l-1)} < 0 \\ \eta_{ji}^{(l-1)} & \text{dla } S_{ji}^l \cdot S_{ji}^{(l-1)} = 0 \end{cases} \quad (2.30)$$

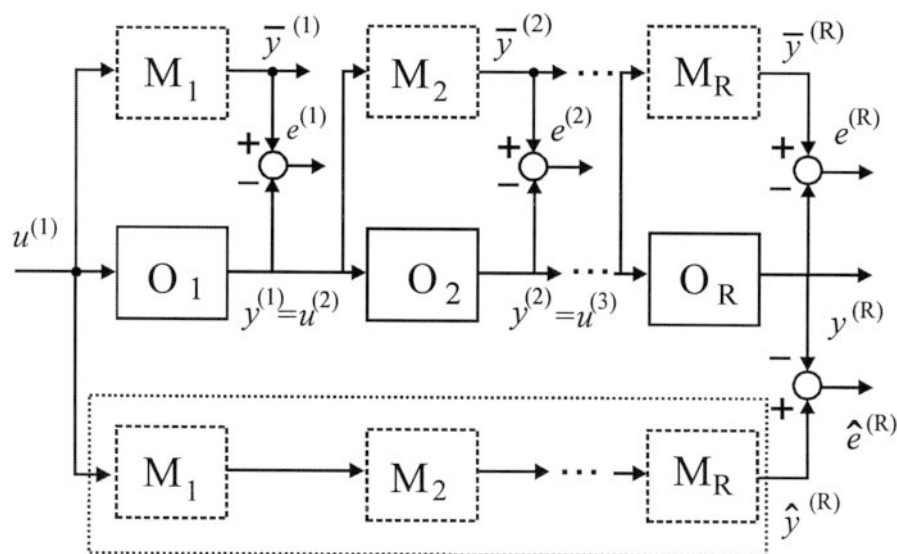
oraz

$$S_{ji}^l = \frac{\partial Q(W(l))}{\partial w_{ji}} \quad (2.31)$$

przy czym a i b są stałymi, których wartości przyjmuje się arbitralnie; np. zgodnie z [59]: $a=1.2$, $b=0.5$; $\eta_{\max} = 50$ oznacza maksymalną wartość współczynnika uczenia a $\eta_{\min} = 10^{-6}$ jest minimalną wartością współczynnika uczenia.

2.3 Model globalnie optymalny z uwzględnieniem jakości modeli lokalnych

Rozważmy system złożony z R elementów przedstawiony na rysunku 2.4.



Rys.2.4 Struktura obiektu złożonego i jego modelu globalnego z uwzględnieniem modeli lokalnych - przypadek szczególny

Dla obiektu przedstawionego na rysunku 2.4 budujemy model globalny o takiej samej strukturze jaką posiada modelowany obiekt. W tym podrozdziale jako model globalny będziemy przyjmować model globalny - szczególny przypadek, w którym nie są uwzględniane

wyjścia modeli częściowych $\hat{y}^{(r)}$ dla $r = 1, 2, \dots, R-1$, lecz wykorzystujemy tylko wyjście globalne modelu $\hat{y}^{(R)}$. W modelu globalnym można wyróżnić modele lokalne M_r (rys.2.4). Wyjścia modeli lokalnych $\bar{y}^{(r)}$ opisane są równaniami (2.1). Różnice między wyjściami modeli lokalnych a wyjściami odpowiadających im elementów obiektu złożonego wykorzystano do sformułowania wskaźników jakości modeli lokalnych.

2.3.1 Syntetyczny wskaźnik jakości

W tym przypadku szukamy kompromisu między modelem globalnie optymalnym obiektu złożonego a optymalnymi modelami lokalnymi. Szukamy w pewnym stopniu optymalnego modelu globalnego – szczególny przypadek jak i optymalnych w pewnym stopniu modeli lokalnych. Wypadkowe kryterium jakości może mieć następującą postać syntetyczną:

$$Q_w(W) = \alpha_0 Q_u(W) + \alpha_1 Q_1(w^{(1)}) + \dots + \alpha_r Q_r(w^{(r)}) + \dots + \alpha_R Q_R(w^{(R)}) \quad (2.32)$$

gdzie: $0 \leq \alpha_r \leq 1$ dla $r = 0, 1, \dots, R$, $\sum_{r=0}^R \alpha_r = 1$ oraz $Q_u(W)$ jest kryterium globalnym (2.15) a

$Q_r(w^{(r)})$ są wskaźnikami jakości modeli lokalnych (2.5).

Współczynniki $\alpha = [\alpha_1 \alpha_2 \dots \alpha_R]$ określają wpływ poszczególnych obiektów lokalnych na wypadkowe kryterium jakości. Współczynnik α_0 określa wpływ kryterium globalnego na wypadkowe kryterium jakości. Poprzez odpowiedni dobór wartości współczynników możemy określić znaczenie elementów prostych w modelu systemu złożonego.

Optymalne wartości parametrów modelu $\tilde{W}$ uzyskujemy po zminimalizowaniu wypadkowej funkcji kryterialnej:

$$\tilde{W} \rightarrow Q_w(W) = \min_W Q_w(W) \quad (2.33)$$

Algorytm wstecznej propagacji błędu dla kryterium (2.32)

Rozwijając wzór (2.32) na wypadkowy wskaźnik jakości możemy zapisać:

$$Q_w(W) = \sum_{k=1}^K Q_w^k(W) \quad (2.34)$$

gdzie

$$Q_w^k(W) = \alpha_0 \beta_R \frac{1}{2} \sum_{j=1}^{J_R} \left(\hat{y}_j^{(R),k} - y_j^{(R),k} \right)^2 + \frac{1}{2} \sum_{r=1}^R \alpha_r \sum_{j=1}^{J_r} \left(\bar{y}_j^{(r),k} - y_j^{(r),k} \right)^2 \quad (2.35)$$

W modelu globalnym - szczególny przypadek, współczynniki w β_r , w kryterium globalnym są równe zero oprócz $\beta_R = 1$, wtedy wypadkowe kryterium ma postać:

$$\begin{aligned} Q_w^k(\mathbf{W}) = & \alpha_0 \frac{1}{2} \sum_{j=1}^{J_R} \left(\hat{y}_j^{(R),k} - y_j^{(R),k} \right)^2 + \alpha_1 \frac{1}{2} \sum_{j=1}^{J_1} \left(\bar{y}_j^{(1),k} - y_j^{(1),k} \right)^2 + \dots \\ & + \alpha_r \frac{1}{2} \sum_{j=1}^{J_r} \left(\bar{y}_j^{(r),k} - y_j^{(r),k} \right)^2 + \dots + \alpha_R \frac{1}{2} \sum_{j=1}^{J_R} \left(\bar{y}_j^{(R),k} - y_j^{(R),k} \right)^2 \end{aligned} \quad (2.36)$$

Stosując metodę gradientową do uczenia sieci, przyrost wagi dla kryterium (2.36) obliczane są według:

$$\Delta w_{ji}^k = -\eta \frac{\partial Q_w^k(\mathbf{W})}{\partial w_{ji}} = -\eta \frac{\partial Q_w^k(\mathbf{W})}{\partial z_j^m} \frac{\partial z_j^m}{\partial w_{ji}} \quad (2.37)$$

gdzie: η jest liczą z przedziału (0,1).

Rozważmy M -warstwową sieć o niemalejących i różniczkowalnych funkcjach aktywacji.

Przez $\tilde{u}_i^m$ oznaczmy wartość wyjścia i -tego neuronu m -tej warstwy przy prezentacji k -tego wzorca ($m=0, 1, \dots, M$; $i=0, 1, \dots, I_m$; $k=1, 2, \dots, K$; gdzie I_m – liczba neuronów m -tej warstwy). Wyjście j -tego neuronu jest obliczane na podstawie wzoru:

$$\tilde{u}_j^{m,k} = f(z_j^{m,k}) = f\left(\sum_{i=0}^{I_{m-1}} w_{ji}^m \tilde{u}_i^{m-1,k}\right) \quad (2.38)$$

Błąd „delta” $\delta_j^{(r),m,k}$ powstały w j -tym neuronie m -tej warstwy dla kryterium (2.36) ma postać:

$$\delta_j^{(r),m,k} = \frac{\partial Q_w^k(\mathbf{W})}{\partial z_j^{m,k}} = \frac{\partial Q_w^k(\mathbf{W})}{\partial \tilde{u}_j^{m,k}} \frac{\partial \tilde{u}_j^m}{\partial z_j^{m,k}} = \frac{\partial Q_w^k(\mathbf{W})}{\partial \tilde{u}_j^{m,k}} f'(z_j^{m,k}) \quad (2.39)$$

Dla warstwy wyjściowej zgodnie ze wzorem (2.36) zapiszemy:

$$\begin{aligned} \frac{\partial Q_w^k(\mathbf{W})}{\partial \tilde{u}_j^M} = & \alpha_0 \frac{1}{2} \frac{\partial \left(\hat{y}_j^{(R),k} - y_j^{(R),k} \right)^2}{\partial \hat{y}_j^{(R),k}} + \alpha_R \frac{1}{2} \frac{\partial \left(\bar{y}_j^{(R),k} - y_j^{(R),k} \right)^2}{\partial \bar{y}_j^{(R),k}} \\ = & \alpha_0 \left(\hat{y}_j^{(R),k} - y_j^{(R),k} \right) + \alpha_R \left(\bar{y}_j^{(R),k} - y_j^{(R),k} \right) \end{aligned} \quad (2.40)$$

Podobnie jak w sekcji 2.2, w „normalnych” warstwach ukrytych $\tilde{u}_j^{m,k}$ wg. (2.38) możemy zapisać:

$$\begin{aligned}
 \frac{\partial Q_w^k(W)}{\partial \tilde{u}_j^{m,k}} &= \sum_{l=1}^{I_{m+1}} \frac{\partial Q_w^k(W)}{\partial z_l^{m+1,k}} \frac{\partial z_l^{m+1,k}}{\partial \tilde{u}_j^{m,k}} = \\
 &= \sum_{l=1}^{I_{m+1}} \frac{\partial Q_w^k(W)}{\partial z_l^{m+1,k}} \frac{\partial}{\partial \hat{u}_j^{m,k}} \sum_{i=1}^{I_m} w_{li}^{(r),m+1} \tilde{u}_i^{m,k} \\
 &= \sum_{l=1}^{I_{m+1}} \frac{\partial Q_w^k(W)}{\partial z_l^{m+1,k}} w_{lj}^{(r),m+1} = \sum_{l=1}^{I_{m+1}} \delta_l^{(r),m+1,k} w_{lj}^{(r),m+1}
 \end{aligned} \tag{2.41}$$

W „dodatkowych” warstwach ukrytych wyjście warstwy jest $\tilde{u}_j^{m,k} = \bar{y}_j^{(r),k}$ jest też wyjściem odpowiedniego modelu lokalnego. Jeżeli modele lokalne składają się z trzech warstw sieci neuronowej każdy, to wskaźniki r i m są ze sobą związane relacją $m = 3 \cdot r$, wtedy np. $\tilde{u}_j^{6,k} = \bar{y}_j^{(2),k}$. Pierwszy czynnik wzoru (2.39) jest proporcjonalny do sumy błędów propagowanych z warstw poprzednich oraz do pochodnej kryterium jakości odpowiedniego modelu lokalnego. Zatem możemy zapisać:

$$\begin{aligned}
 \frac{\partial Q_w^k(W)}{\partial \tilde{u}_j^{m,k}} &= \sum_{l=1}^{I_{m+1}} \frac{\partial Q_w^k(W)}{\partial z_l^{m+1,k}} \frac{\partial z_l^{m+1,k}}{\partial \tilde{u}_j^{m,k}} + \frac{\alpha_r}{2} \frac{\partial (\bar{y}_j^{(r),k} - y_j^{(r),k})^2}{\partial \tilde{u}_j^{m,k}} \\
 &= \sum_{l=1}^{I_{m+1}} \frac{\partial Q_w^k(W)}{\partial z_l^{m+1,k}} \frac{\partial}{\partial \tilde{u}_j^{m,k}} \sum_{i=1}^{I_m} w_{li}^{(r+1),m+1} \tilde{u}_i^{m,k} + \frac{\alpha_r}{2} \frac{\partial (\bar{y}_j^{(r),k} - y_j^{(r),k})^2}{\partial \bar{y}_j^{(r),k}} \\
 &= \sum_{l=1}^{I_{m+1}} \delta_l^{(r+1),m+1,k} w_{lj}^{(r+1),m+1} + \alpha_r (\bar{y}_j^{(r),k} - y_j^{(r),k})
 \end{aligned} \tag{2.42}$$

Stąd na podstawie zależności (2.40), (2.41), (2.42) możemy dla każdego neuronu obliczyć błąd „delta”:

- dla warstwy wyjściowej

$$\delta_j^{(R),M,k} = f'(z_j^{M,k}) \left(\alpha_0 (\hat{y}_j^{(R),k} - y_j^{(R),k}) + \alpha_R (\bar{y}_j^{(R),k} - y_j^{(R),k}) \right) \tag{2.43}$$

- dla warstw ukrytych „normalnych”

$$\delta_j^{(r),m,k} = f'(z_j^{m,k}) \sum_{l=1}^{I_{m+1}} \delta_l^{(r),m+1,k} w_{lj}^{(r),m+1} \tag{2.44}$$

- dla warstw ukrytych „dodatkowych”

$$\delta_j^{(r),m,k} = f'(z_j^{m,k}) \sum_{l=1}^{I_{m+1}} \delta_l^{(r+1),m+1,k} w_{lj}^{(r+1),m+1} + \alpha_r f'(z_j^{m,k}) (\bar{y}_j^{(r),k} - y_j^{(r),k}) \tag{2.45}$$

Przy takich definicjach błędu możemy zapisać skumulowaną regułę delta na przyrosty wag:

$$\Delta w_{ji}^{(r),m} = \sum_{k=1}^K \Delta w_{ji}^{(r),m,k} = -\eta \delta_j^{(r),m,k} \tilde{u}_i^{m-1,k} \quad (2.46)$$

Wykorzystując wzory (2.43), (2.44), (2.45) algorytm *Rprop* [44] po modyfikacji ma następującą postać:

$$\Delta w_{ji}^{(r),m,k} = \begin{cases} -\eta_{ji}^l \operatorname{sgn}(\delta_j^{(R),M,k} \hat{u}_i^{m-1,k}) & \text{w warstwie wyjściowej} \\ -\eta_{ji}^l \operatorname{sgn}(\delta_j^{(r),m,k} \hat{u}_i^{m-1,k}) & \text{w warstwach ukrytych} \end{cases} \quad (2.47)$$

gdzie η_{ji}^l obliczane jest według (2.30) i (2.31).

2.3.2 Zadawalające modele lokalne

Ponownie rozważmy system złożony przedstawiony na rysunku 2.4. W modelowaniu globalnym z ograniczeniami na modele lokalne poszukuje się optymalnego zbioru parametrów $\tilde{W}$ przy zadanej funkcji kosztu $Q(W)$ wg. (2.14) uwzględniając ograniczenia nałożone na funkcje błędu modeli lokalnych $Q_r(w^{(r)})$ (2.4)

$$\tilde{W} \rightarrow Q(\tilde{W}) = \min_{W \in D_W} Q(W) \quad (2.48)$$

gdzie

$$D_W = \{w^{(1)}, \dots, w^{(R)} : Q_1(w^{(1)}) \leq \lambda_1, Q_2(w^{(2)}) \leq \lambda_2, \dots, Q_R(w^{(R)}) \leq \lambda_R\} \quad (2.49)$$

Wektor dodatnich parametrów $\lambda = [\lambda_1 \lambda_2 \dots \lambda_R]$ zawiera wartości dopuszczalnych wskaźników jakości poszczególnych modeli lokalnych.

W tym przypadku szukamy optymalnego modelu globalnego przy narzuconych ograniczeniach na jakość modeli lokalnych. Zgadza się na model globalny nieco gorszy od optymalnego, lecz jakość modeli lokalnych nie może być gorsza od dopuszczalnych wartości.

Jako miarę błędu sieci Q w ogólnym przypadku można przyjąć wzór (2.14).

W szczególnym przypadku, który rozważamy zgodnie z rys.2.3, w przypadku, gdy $\sum_{r=1}^{R-1} \beta_r = 0$

i $\beta_R = 1$ wzór (2.14) przyjmuje postać:

$$Q_u(W) = \frac{1}{2} \sum_{k=1}^K \sum_{j=1}^{J_R} \left(\hat{y}_j^{(R),k} - y_j^{(R),k} \right)^2 \quad (2.50)$$

Funkcje ograniczeń h wynikają z ograniczeń na wskaźniki jakości poszczególnych modeli lokalnych:

$$\begin{aligned}
h_1(\mathbf{w}^{(1)}) &= \frac{1}{2} \sum_{k=1}^K \sum_{j=1}^{J_1} \left(\bar{y}_j^{(1),k} - y_j^{(1),k} \right)^2 - \lambda_1 \leq 0 \\
h_2(\mathbf{w}^{(2)}) &= \frac{1}{2} \sum_{k=1}^K \sum_{j=1}^{J_2} \left(\bar{y}_j^{(2),k} - y_j^{(2),k} \right)^2 - \lambda_2 \leq 0 \\
&\vdots \\
h_R(\mathbf{w}^{(R)}) &= \frac{1}{2} \sum_{k=1}^K \sum_{j=1}^{J_R} \left(\bar{y}_j^{(R),k} - y_j^{(R),k} \right)^2 - \lambda_R \leq 0
\end{aligned} \tag{2.51}$$

Jednym ze sposobów poszukiwania minimum wielokryterialnego wskaźnika jakości z ograniczeniami jest użycie funkcji kary [14, 47].

W metodach funkcji kary tworzymy zmodyfikowaną funkcję kryterialną przez dodanie do funkcji kryterialnej Q_u pewnej funkcji zwanej funkcją kary. Wartość funkcji kary reprezentuje „karę” za niespełnienie warunków ograniczających lub stanowi „barierę” uniemożliwiającą opuszczenie zbioru punktów dopuszczalnych.

Ogólnie zmodyfikowaną funkcją kryterialną jest:

$$T(\mathbf{W}, p) \equiv Q_u(\mathbf{W}) + G(\mathbf{W}, p), \quad p \in (0, +\infty) \tag{2.52}$$

przy czym p – parametr, $G(\mathbf{W}, p)$ jest funkcją kary.

Funkcję kary $G(\mathbf{W}, p)$ dobieramy tak, żeby przy ustalonej wartości parametru p wyznaczony punkt minimum funkcji kryterialnej $T(\mathbf{W}, p)$ można było przyjąć za przybliżenie $\mathbf{W}^*$ minimum funkcji $Q_u(\mathbf{W})$ przy warunkach ograniczających. Dokładność takiego przybliżenia zależy od postaci funkcji kary oraz od wartości parametru p . Przybliżenie takie jest tym dokładniejsze, im szybciej funkcja kary rośnie w miarę oddalania się od zbioru punktów dopuszczalnych lub naruszania ograniczeń. Wzrost parametru p powoduje szybszy wzrost funkcji kary.

W zależności od tego jaką postać funkcji kary $G(\mathbf{W}, p)$ wybierzemy, zewnętrzną czy wewnętrzną mamy, *metodę zewnętrznego funkcji kary* oraz *metodę wewnętrznego funkcji kary*.

- **Metoda zewnętrznego funkcji kary**

Funkcja kary przyjmuje postać [47]:

$$G(\mathbf{W}, p) = p \cdot K(\mathbf{W}), \quad p \in (0, +\infty) \tag{2.53}$$

gdzie: funkcja $K(\mathbf{W})$ jest równa zero, jeżeli spełnione są warunki ograniczające lub przyjmuje wartości większe od zera równe funkcji kary przy niespełnieniu ograniczeń. Ma ona sens kary nakładanej za niespełnienie ograniczeń.

Funkcja kary może mieć postać sumy kar dla poszczególnych ograniczeń:

$$K(\mathbf{W}) = \sum_{r=1}^R K_r(\mathbf{w}^{(r)}) \quad (2.54)$$

W szczególnym przypadku funkcja K_r może mieć postać:

$$K_r(\mathbf{w}^{(r)}) = \max(0, h_r(\mathbf{w}^{(r)})) \quad (2.54)$$

przy czym

$$\max(0, h_r(\mathbf{w}^{(r)})) \equiv \begin{cases} 0, & \text{jeżeli } h_r(\mathbf{w}^{(r)}) \leq 0 \\ h_r(\mathbf{w}^{(r)}), & \text{jeżeli } h_r(\mathbf{w}^{(r)}) > 0 \end{cases} \quad (2.55)$$

Funkcje kar mogą przyjmować różnorodne postaci, dlatego rozważymy przypadek ogólny. Zmodyfikowaną funkcję kryterialną tworzymy przez dodanie do funkcji celu (2.50) funkcji kary w postaci ogólnej:

$$T(\mathbf{W}, p) \equiv Q_u(\mathbf{W}) + G(\mathbf{W}, p) = \frac{1}{2} \sum_{k=1}^K \sum_{j=1}^{J_R} (\hat{y}_j^{(R),k} - y_j^{(R),k})^2 + p \sum_{r=1}^R K_r(\mathbf{w}^{(r)}) \quad (2.56)$$

Algorytm wstecznej propagacji błędu dla kryterium (2.56)

Podobnie jak w poprzedniej sekcji zostanie użyta metoda gradientowa minimalizacji zmodyfikowanej funkcji kryterialnej $T(\mathbf{W}, p)$. Stosując powyższą metodę do uczenia sieci, przyrost wagi dla (2.56) obliczamy według:

$$\Delta w_{ji} = -\eta \frac{\partial T(\mathbf{W}, p)}{\partial w_{ji}} = -\eta \sum_{k=1}^K \frac{\partial T(\mathbf{W}, p)}{\partial z_j^k} \frac{\partial z_j^k}{\partial w_{ji}} \quad (2.57)$$

Wzór (2.57) możemy zapisać w następującej postaci:

$$\begin{aligned} \Delta w_{ji} = & -\eta \frac{\partial Q_u(\mathbf{W})}{\partial w_{ji}} - \eta \frac{\partial G(\mathbf{W}, p)}{\partial w_{ji}} = \\ & -\eta \sum_{k=1}^K \frac{\partial Q_u(\mathbf{W})}{\partial z_j^k} \frac{\partial z_j^k}{\partial w_{ji}} - \eta \sum_{k=1}^K \frac{\partial G(\mathbf{W}, p)}{\partial z_j^k} \frac{\partial z_j^k}{\partial w_{ji}} \end{aligned} \quad (2.58)$$

gdzie: η jest liczbą z przedziału (0,1).

W sieci M -warstwowej w j -tym elemencie m -tej warstwy pierwszy składnik wzoru (2.58) ma postać:

$$\begin{aligned} \sum_{k=1}^K \frac{\partial Q_u(\mathbf{W})}{\partial z_j^{m,k}} \frac{\partial z_j^{k,m}}{\partial w_{ji}} &= \sum_{k=1}^K \frac{\partial Q_u(\mathbf{W})}{\partial \tilde{u}_j^{m,k}} \frac{\partial \tilde{u}_j^{m,k}}{\partial z_j^{m,k}} \frac{\partial z_j^{k,m}}{\partial w_{ji}} \\ &= \sum_{k=1}^K \frac{\partial Q_u(\mathbf{W})}{\partial \tilde{u}_j^{m,k}} f'(z_j^{m,k}) \tilde{u}_i^{m-1,k} = \sum_{k=1}^K q_j^{(r),m,k} \tilde{u}_i^{m-1,k} \end{aligned} \quad (2.59)$$

Podobnie drugi składnik wzoru (2.58) w j -tym elemencie m -tej warstwy ma postać:

$$\begin{aligned} \sum_{k=1}^K \frac{\partial G(\mathbf{W}, p)}{\partial z_j^{m,k}} \frac{\partial z_j^{m,k}}{\partial w_{ji}} &= \sum_{k=1}^K \frac{\partial G(\mathbf{W}, p)}{\partial \tilde{u}_j^{(m,k),k}} \frac{\partial \tilde{u}_j^{(m,k),k}}{\partial z_j^{m,k}} \frac{\partial z_j^{m,k}}{\partial w_{ji}} \\ &= \sum_{k=1}^K \frac{\partial G(\mathbf{W}, p)}{\partial \tilde{u}_j^{m,k}} f'(z_j^{m,k}) \tilde{u}_i^{m-1,k} = \sum_{k=1}^K g_j^{(r),m,k} \tilde{u}_i^{m-1,k} \end{aligned} \quad (2.60)$$

Zatem wzór (2.57) wyraża się jako:

$$\Delta w_{ji}^{(r),m} = \sum_{k=1}^K \left(q_j^{(r),m,k} + g_j^{(r),m,k} \right) \tilde{u}_i^{m-1,k} = \sum_{k=1}^K \delta_j^{(r),m,k} \tilde{u}_i^{m-1,k} \quad (2.61)$$

gdzie $\delta_j^{(r),m,k} = q_j^{(r),m,k} + g_j^{(r),m,k}$.

Dla warstwy wyjściowej pierwszy czynnik wzoru (2.59) na podstawie (2.50) zapiszemy:

$$\frac{\partial Q_u(\mathbf{W})}{\partial \tilde{u}_j^{M,k}} = \frac{\partial \frac{1}{2} \sum_{k=1}^K \sum_{j=1}^{J_R} \left(\hat{y}_j^{(R),k} - y_j^{(R),k} \right)^2}{\partial \hat{y}_j^{(R),k}} = \left(\hat{y}_j^{(R),k} - y_j^{(R),k} \right) \quad (2.62)$$

Podobnie pierwszy czynnik wzoru (2.60) jest następujący:

$$\frac{\partial G(\mathbf{W}, p)}{\partial \tilde{u}_j^{M,k}} = p \frac{\partial \sum_{r=1}^R K_R(\mathbf{w}^{(R)})}{\partial \hat{y}_j^{(R),k}} = p \frac{\partial K_R(\mathbf{w}^{(R)})}{\partial \hat{y}_j^{(R),k}} \quad (2.63)$$

Podobnie jak w poprzednich sekcjach warstwy ukryte dzielimy na „normalne” i „dodatkowe”.

W „normalnych” warstwach ukrytych na podstawie (1.29), (2.56) oraz (2.61) możemy zapisać:

$$\begin{aligned} \frac{\partial T(\mathbf{W}, p)}{\partial \tilde{u}_j^{m,k}} &= \sum_{l=1}^{I_{m+1}} \frac{\partial T(\mathbf{W}, p)}{\partial z_l^{(m+1),k}} \frac{\partial z_l^{m+1,k}}{\partial \tilde{u}_j^{m,k}} = \\ &= \sum_{l=1}^{I_{m+1}} \frac{\partial T(\mathbf{W}, p)}{\partial z_l^{m+1,k}} \frac{\partial}{\partial \tilde{u}_j^{m,k}} \sum_{i=1}^{I_m} w_{li}^{(r),m+1} \tilde{u}_i^{m,k} \\ &= \sum_{l=1}^{I_{m+1}} \frac{\partial T(\mathbf{W}, p)}{\partial z_l^{m+1,k}} w_{lj}^{(r),m+1} = \sum_{l=1}^{I_{m+1}} \delta_l^{(r),m+1,k} w_{lj}^{(r),m+1} \end{aligned} \quad (2.64)$$

W „dodatkowych” warstwach ukrytych $\tilde{u}_j^{m,k} = \bar{y}_j^{(r),k}$ zatem:

$$\begin{aligned}
 \frac{\partial T(\mathbf{W})}{\partial \tilde{u}_j^{m,k}} &= \frac{\partial T(\mathbf{W}, p)}{\partial \tilde{u}_j^{m,k}} + \frac{\partial G(\mathbf{W}, p)}{\partial \bar{y}_j^{(r),k}} \\
 &= \sum_{l=1}^{I_{m+1}} \frac{\partial T(\mathbf{W}, p)}{\partial z_l^{m+1,k}} \frac{\partial z_l^{m+1,k}}{\partial \tilde{u}_j^{m,k}} + p \frac{\partial \sum_{r=1}^R K_r(\mathbf{w}^{(r)})}{\partial \bar{y}_j^{(r),k}} \\
 &= \sum_{l=1}^{I_{m+1}} \frac{\partial T(\mathbf{W}, p)}{\partial z_l^{m+1,k}} \frac{\partial}{\partial \tilde{u}_j^{m,k}} \sum_{i=1}^{I_m} w_{li}^{(r+1),m+1} \tilde{u}_i^{m,k} + p \frac{\partial K_r(\mathbf{w}^{(r)})}{\partial \bar{y}_j^{(r),k}} \\
 &= \sum_{l=1}^{I_{m+1}} \delta_l^{(r+1),m+1,k} w_{lj}^{(r+1),m+1} + p \frac{\partial K_r(\mathbf{w}^{(r)})}{\partial \bar{y}_j^{(r),k}}
 \end{aligned} \tag{2.65}$$

Stąd na podstawie zależności (2.61), (2.62), (2.63), (2.64), i (2.65) możemy dla każdego neuronu obliczyć:

- dla warstwy wyjściowej ($m=M$):

$$\delta_j^{(R),M,k} = (\hat{y}_j^{(R),k} - \hat{y}_j^{(R),k}) f'(z_j^{m,k}) + p \frac{\partial K_R(\mathbf{w}^{(R)})}{\partial \bar{y}_j^{(R),k}} f'(z_j^{m,k}) \tag{2.66}$$

- dla warstw ukrytych normalnych ($m=1, 2, \dots, M-1$)

$$\delta_j^{(r),m,k} = f'(z_j^{m,k}) \sum_{l=1}^{I_{m+1}} \delta_l^{(r),m+1,k} w_{lj}^{(r),m+1} \tag{2.67}$$

- dla warstw ukrytych „dodatkowych” ($m=1, 2, \dots, M-1$) i ($r=1, 2, \dots, R-1$)

➤ przy spełnieniu ograniczenia, tj. $h_r(\mathbf{w}^{(r)}) \leq 0$

$$\delta_j^{(r),m,k} = f'(z_j^{m,k}) \sum_{l=1}^{I_{m+1}} \delta_l^{(r+1),m+1,k} w_{lj}^{(r+1),m+1} \tag{2.68a}$$

➤ przy nie spełnieniu ograniczenia, tj. $h_r(\mathbf{w}^{(r)}) > 0$

$$\delta_j^{(r),m,k} = f'(z_j^{m,k}) \left(\sum_{l=1}^{I_{m+1}} \delta_l^{(r+1),m+1,k} w_{lj}^{(r+1),m+1} + p \frac{\partial K_r(\mathbf{w}^{(r)})}{\partial \bar{y}_j^{(r),k}} \right) \tag{2.68b}$$

Przy takich definicjach błędu „delta” możemy zapisać skumulowaną regułę delta na przyrosty wag w warstwach:

$$\Delta w_{ji}^{(r),m} = \begin{cases} -\eta \sum_{k=1}^K \delta_j^{(R),M,k} \tilde{u}_i^{m-1,k} & \text{wyjściowej} \\ -\eta \sum_{k=1}^K \delta_j^{(r),m,k} \tilde{u}_i^{m-1,k} & \text{w ukrytych} \end{cases} \tag{2.69}$$

Po wstawieniu błędów „delta” wzór (2.69) ma postać

- w warstwie wyjściowej ($m=M$):

$$\Delta w_{ji}^{(r),m} = -\eta \sum_{k=1}^K f'(z_j^{M,k}) \left(\hat{y}_j^{(R),k} - y_j^{(R),k} \right) + \begin{cases} 0, & \text{gdy } h_R(\mathbf{w}^{(R)}) \leq 0 \\ p \frac{\partial K_R(\mathbf{w}^{(R)})}{\partial \tilde{y}_j^{(R),k}}, & \text{gdy } h_R(\mathbf{w}^{(R)}) > 0 \end{cases} \tilde{u}_i^{m-1,k} \quad (2.70a)$$

- w warstwach ukrytych „normalnych” ($m=1, 2, \dots, M-1$):

$$\Delta w_{ji}^{(r),m} = -\eta \sum_{k=1}^K f'(z_j^{m,k}) \sum_{l=1}^{I_{m+1}} \delta_l^{(r),m+1,k} w_{lj}^{(r),m+1} \tilde{u}_i^{m-1,k} \quad (2.70b)$$

- w warstwach ukrytych „dodatkowych” ($m=1, 2, \dots, M-1$) i ($r=1, 2, \dots, R-1$):

$$\Delta w_{ji}^{(r),m} = -\eta \sum_{k=1}^K f'(z_j^{m,k}) \left(\sum_{l=1}^{I_{m+1}} \delta_l^{(r+1),m+1,k} w_{lj}^{(r+1),m+1} + \begin{cases} 0, & \text{gdy } h_r(\mathbf{w}^{(r)}) \leq 0 \\ p \frac{\partial K_r(\mathbf{w}^{(r)})}{\partial \tilde{y}_j^{(r),k}}, & \text{gdy } h_r(\mathbf{w}^{(r)}) > 0 \end{cases} \right) \tilde{u}_i^{m-1,k} \quad (2.70c)$$

W szczególnym przypadku, gdy funkcja kary podana jest w postaci wzoru (2.55), dla warstwy wyjściowej wzór (2.63) przyjmuje przybliżoną postać ze względu na nieróżniczkowalną funkcję kary:

$$\begin{aligned} \frac{\partial G(\mathbf{W}, p)}{\partial \tilde{u}_j^{m,k}} &= p \frac{\partial \sum_{r=1}^R K_r(\mathbf{w}^{(r)})}{\partial \tilde{y}_j^{(R),k}} = p \frac{\partial \sum_{r=1}^R \max(0, h_r(\mathbf{w}^{(r)}))}{\partial \tilde{y}_j^{(R),k}} \\ &= p \frac{\partial \max \left(0, \frac{1}{2} \sum_{k=1}^K \sum_{j=1}^{J_R} \left(\bar{y}_j^{(R),k} - y_j^{(R),k} \right)^2 - \lambda_R \right)}{\partial \tilde{y}_j^{(R),k}} \\ &= p \begin{cases} 0, & \text{gdy } h_R(\mathbf{w}^{(R)}) \leq 0 \\ \max(0, \left(\bar{y}_j^{(R),k} - y_j^{(R),k} \right)) & \text{gdy } h_R(\mathbf{w}^{(R)}) > 0 \end{cases} \end{aligned} \quad (2.71)$$

W dodatkowych warstwach ukrytych wzór (2.63) ze względu na nieróżniczkowalną funkcję kary (2.55) również przyjmuje postać przybliżoną:

$$\begin{aligned}
 \frac{\partial T(\mathbf{W}, p)}{\partial \tilde{u}_j^{m,k}} &= \frac{\partial T(\mathbf{W}, p)}{\partial \tilde{u}_j^{m,k}} + \frac{\partial G(\mathbf{W}, p)}{\partial \tilde{y}_j^{m,k}} = \sum_{l=1}^{I_{m+1}} \frac{\partial T(\mathbf{W}, p)}{\partial z_l^{m+1,k}} \frac{\partial z_l^{m+1,k}}{\partial \tilde{u}_j^{m,k}} + p \frac{\partial \sum_{r=1}^R K_r(\mathbf{w}^{(r)})}{\partial \tilde{y}_j^{m,k}} \\
 &= \sum_{l=1}^{I_{m+1}} \frac{\partial T(\mathbf{W}, p)}{\partial z_l^{m+1,k}} \frac{\partial}{\partial \tilde{u}_j^{m,k}} \sum_{i=1}^{I_m} w_{li}^{(r+1),m+1} \tilde{u}_i^{m,k} \\
 &\quad \partial \max \left(0, \left(\frac{1}{2} \sum_{k=1}^K \sum_{j=1}^{J_r} \left(\bar{y}_j^{(r),k} - y_j^{(r),k} \right)^2 - \lambda_r \right) \right) \\
 &\quad + p \frac{\partial \left(\frac{1}{2} \sum_{k=1}^K \sum_{j=1}^{J_r} \left(\bar{y}_j^{(r),k} - y_j^{(r),k} \right)^2 - \lambda_r \right)}{\partial \tilde{y}_j^{(r),k}} \\
 &= \sum_{l=1}^{I_{m+1}} \delta_l^{(r+1),m+1,k} w_{lj}^{(r+1),m+1} + p \begin{cases} 0, & \text{gdy } h_r(\mathbf{w}^{(r)}) \leq 0 \\ \max \left(0, \left(\bar{y}_j^{(r),k} - y_j^{(r),k} \right) \right), & \text{gdy } h_r(\mathbf{w}^{(r)}) > 0 \end{cases}
 \end{aligned} \tag{2.72}$$

Ostatecznie skumulowana reguła delty ma postać:

- w warstwie wyjściowej:

$$\begin{aligned}
 \Delta w_{ji}^{(R),M} &= -\eta \sum_{k=1}^K f'(z_j^{M,k}) \cdot \left(\hat{y}_j^{(R),k} - y_j^{(R),k} \right) + \\
 &\quad p \begin{cases} 0, & \text{gdy } h_R(\mathbf{w}^{(R)}) \leq 0 \\ \max \left(0, \left(\bar{y}_j^{(R),k} - y_j^{(R),k} \right) \right), & \text{gdy } h_R(\mathbf{w}^{(R)}) > 0 \end{cases} \tilde{u}_i^{m-1,k}
 \end{aligned} \tag{2.73a}$$

- w „normalnych” warstwach ukrytych:

$$\Delta w_{ji}^{(r),m} = -\eta \sum_{k=1}^K f'(z_j^{m,k}) \sum_{l=1}^{I_{m+1}} \delta_l^{(r),m+1,k} w_{lj}^{(r),m+1} \tilde{u}_i^{m-1,k} \tag{2.73b}$$

- w „dodatkowych” warstwach ukrytych:

$$\begin{aligned}
 \Delta w_{ji}^{(r),m} &= -\eta \sum_{k=1}^K f'(z_j^{m,k}) \cdot \left(\sum_{l=1}^{I_{m+1}} \delta_l^{(r+1),m+1,k} w_{lj}^{(r+1),m+1} \right. \\
 &\quad \left. + p \begin{cases} 0, & \text{gdy } h_r(\mathbf{w}^{(r)}) \leq 0 \\ \max \left(0, \left(\bar{y}_j^{(r),k} - y_j^{(r),k} \right) \right), & \text{gdy } h_r(\mathbf{w}^{(r)}) > 0 \end{cases} \tilde{u}_i^{m-1,k} \right)
 \end{aligned} \tag{2.73c}$$

- **Metoda wewnętrznej funkcji kary**

Funkcja kary przyjmuje postać [47]:

$$G(\mathbf{W}, p) = \frac{B(\mathbf{W})}{p}, \quad p \in (0, +\infty) \tag{2.74}$$

przy czym $B(\mathbf{W})$ musi być funkcją ciągłą, oraz nie może przyjmować wartości ujemnych dla parametrów spełniających ograniczenia, oraz musi dążyć do nieskończoności jeżeli wskaźniki jakości zbliżają się do ograniczeń.

Wewnętrzna funkcja kary nazywana jest też funkcją barierową, ponieważ odgrywa rolę bariery nie pozwalającej na opuszczenie zbioru punktów dopuszczalnych. Często wewnętrzna funkcja kary przyjmuje postać:

$$B(\mathbf{W}) = \sum_{r=1}^R B_r(\mathbf{w}^{(r)}) \quad (2.75)$$

gdzie B_r są funkcjami ograniczeń w postaci:

$$B_r(\mathbf{w}^{(r)}) = -\frac{1}{h_r(\mathbf{w}^{(r)})} \quad (2.76)$$

Tworzymy zmodyfikowaną funkcję kryterialną poprzez dodanie do funkcji celu Q_u (2.50) wewnętrznej funkcji kary:

$$T(\mathbf{W}, p) \equiv Q_u(\mathbf{W}) + G(\mathbf{W}, p) = \frac{1}{2} \sum_{k=1}^K \sum_{j=1}^{J_R} \left(\hat{y}_j^{(R),k} - y_j^{(R),k} \right)^2 + \frac{\sum_{r=1}^R B_r(\mathbf{w}^{(r)})}{p} \quad (2.77)$$

Algorytm wstecznej propagacji błędów dla kryterium (2.77)

Do minimalizacji funkcji kryterialnej $T(\mathbf{W}, p)$ wykorzystano iteracyjną metodę gradientową. Stosując powyższą metodę do uczenia sieci przyrosty wag obliczmy zgodnie z (2.57):

$$\Delta w_{ji} = -\eta \frac{\partial T(\mathbf{W}, p)}{\partial w_{ji}} = -\eta \sum_{k=1}^K \frac{\partial T(\mathbf{W}, p)}{\partial z_j^k} \frac{\partial z_j^k}{\partial w_{ji}}$$

Wzór (2.57) w tej sekcji także można zapisać w postaci (2.58), (2.61) oraz (2.59) i (2.60). Dla sieci M -warstwowej dla warstwy wyjściowej wyrażenie (2.62) także się nie zmienia. Natomiast wyrażenie (2.63) zgodnie z (2.74) i (2.75) ma postać:

$$\frac{\partial G(\mathbf{W}, p)}{\partial \tilde{u}_j^{M,k}} = \frac{\partial G(\mathbf{W}, p)}{\partial \tilde{y}_j^{(R),k}} = \frac{1}{p} \frac{\partial B_R(\mathbf{w}^{(R)})}{\partial \tilde{y}_j^{(R),k}} \quad (2.78)$$

W normalnych warstwach ukrytych pochodna funkcji $T(\mathbf{W}, p)$ względem $\tilde{u}_j^m$ ma postać wyrażenia (2.64)

$$\begin{aligned}
 \frac{\partial T(\mathbf{W}, p)}{\partial \tilde{u}_j^m} &= \sum_{l=1}^{I_{m+1}} \frac{\partial T(\mathbf{W})}{\partial z_l^{(m+1)}} \frac{\partial z_l^{m+1}}{\partial \tilde{u}_j^m} = \\
 &= \sum_{l=1}^{I_{m+1}} \frac{\partial T(\mathbf{W})}{\partial z_l^{m+1}} \frac{\partial}{\partial \tilde{u}_j^m} \sum_{i=1}^{I_m} w_{li}^{(r),m+1} \tilde{u}_i^m \\
 &= \sum_{l=1}^{I_{m+1}} \frac{\partial T(\mathbf{W})}{\partial z_l^{m+1}} w_{lj}^{(r),m+1} = \sum_{l=1}^{I_{m+1}} \delta_l^{(r),m+1} w_{lj}^{(r),m+1}
 \end{aligned}$$

W dodatkowych warstwach ukrytych gdzie $\tilde{u}_j^{m,k} = y_j^{(r),k}$ na podstawie (1.29) oraz (2.77) przyjmuje postać:

$$\begin{aligned}
 \frac{\partial T(\mathbf{W}, p)}{\partial \tilde{u}_j^{m,k}} &= \frac{\partial T(\mathbf{W}, p)}{\partial \tilde{u}_j^{m,k}} + \frac{\partial G(\mathbf{w}^{(r)}, p)}{\partial \tilde{u}_j^{m,k}} \\
 &= \sum_{l=1}^{I_{m+1}} \frac{\partial T(\mathbf{W}, p)}{\partial z_l^{m+1,k}} \frac{\partial z_l^{m+1,k}}{\partial \tilde{u}_j^{m,k}} + \frac{1}{p} \frac{\partial B_r(\mathbf{w}^{(r)})}{\partial \tilde{u}_j^{m,k}} = \\
 &= \sum_{l=1}^{I_{m+1}} \frac{\partial T(\mathbf{W}, p)}{\partial z_l^{m+1,k}} \frac{\partial}{\partial \tilde{u}_j^{m,k}} \sum_{i=1}^{I_m} w_{li}^{(r+1),m+1} \tilde{u}_i^{m,k} + \frac{1}{p} \frac{\partial B_r(\mathbf{w}^{(r)})}{\partial \tilde{u}_j^{m,k}} \\
 &= \sum_{l=1}^{I_{m+1}} \delta_l^{(r+1),m+1,k} w_{lj}^{(r+1),m+1} + \frac{1}{p} \frac{\partial B_r(\mathbf{w}^{(r)})}{\partial \tilde{y}_j^{(r),k}}
 \end{aligned} \tag{2.79}$$

Stąd na podstawie zależności (2.61), (2.62), (2.64), (2.78) i (2.79) możemy dla każdego neuronu obliczyć błędy „delta”:

- dla warstwy wyjściowej:

$$\delta_j^{(R),M,k} = f'(z_j^{M,k}) \left(\hat{y}_j^{(R),k} - y_j^{(R),k} + \frac{1}{p} \frac{\partial B_R(\mathbf{w}^{(R)})}{\partial \tilde{y}_j^{(R),k}} \right) \tag{2.80}$$

- dla warstw ukrytych „normalnych”:

$$\delta_j^{(r),m,k} = f'(z_j^{m,k}) \sum_{l=1}^{I_{m+1}} \delta_l^{(r),m+1,k} w_{lj}^{(r),m+1} \tag{2.81}$$

- dla warstw ukrytych „dodatkowych”:

$$\delta_j^{(r),m,k} = f'(z_j^{m,k}) \left(\sum_{l=1}^{I_{m+1}} \delta_l^{(r+1),m+1,k} w_{lj}^{(r+1),m+1} + \frac{1}{p} \frac{\partial B_r(\mathbf{w}^{(r)})}{\partial \tilde{y}_j^{(r),k}} \right) \tag{2.82}$$

Przy takich definicjach błędów „delta” możemy zapisać skumulowaną regułę delta na przyrosty wag w warstwach:

$$\Delta w_{ji}^{(r),m} = \begin{cases} -\eta \sum_{k=1}^K \delta_j^{(R),M,k} \tilde{u}_i^{M-1,k} & \text{wyjściowej} \\ -\eta \sum_{k=1}^K \delta_j^{(r),m,k} \tilde{u}_i^{m-1,k} & \text{ukrytych} \end{cases} \quad (2.83)$$

W szczególnym przypadku, gdy funkcja kary ma postać (2.76) a funkcja ograniczeń ma postać (2.15), to dla warstwy wyjściowej wzór (2.63) przyjmuje postać:

$$\begin{aligned} \frac{\partial G(\mathbf{W}, p)}{\partial \tilde{u}_j^{M,k}} &= \frac{\partial G(\mathbf{W}, p)}{\partial \tilde{y}_j^{(R),k}} = + \frac{1}{p} \frac{\partial B_R(\mathbf{w}^{(R)})}{\partial \tilde{y}_j^{(R),k}} \\ &= - \frac{1}{p} \frac{\partial \left(\frac{1}{(Q_R(\mathbf{w}^{(R)}) - \lambda_R)} \right)}{\partial \tilde{y}_j^{(R),k}} \\ &= + \frac{1}{p} \frac{(\bar{y}_j^{(R),k} - y_j^{(R),k})}{(Q_R(\mathbf{w}^{(R)}) - \lambda_R)^2} \end{aligned} \quad (2.84)$$

W dodatkowych warstwach ukrytych wzór (2.65) przyjmuje postać:

$$\begin{aligned} \frac{\partial T(\mathbf{W}, p)}{\partial \tilde{u}_j^{m,k}} &= \frac{\partial T(\mathbf{W}, p)}{\partial \tilde{u}_j^{m,k}} + \frac{\partial G(\mathbf{w}^{(r)}, p)}{\partial \tilde{u}_j^{m,k}} \\ &= \sum_{l=1}^{I_{m+1}} \frac{\partial T(\mathbf{W})}{\partial z_l^{m+1,k}} \frac{\partial z_l^{m+1,k}}{\partial \tilde{u}_j^{m,k}} + \frac{1}{p} \frac{\partial B_r(\mathbf{w}^{(r)})}{\partial \tilde{u}_j^{m,k}} = \\ &= \sum_{l=1}^{I_{m+1}} \frac{\partial T(\mathbf{W})}{\partial z_l^{m+1,k}} \frac{\partial}{\partial \tilde{u}_j^{m,k}} \sum_{i=1}^{I_m} w_{li}^{(r+1),m+1} \tilde{u}_i^{m,k} - \frac{1}{p} \frac{\partial \left(\frac{1}{(Q_r(\mathbf{w}^{(r)}) - \lambda_r)} \right)}{\partial \tilde{y}_j^{(r),k}} \\ &= \sum_{l=1}^{I_{m+1}} \delta_l^{(r+1),m+1,k} w_{lj}^{(r+1),m+1} + \frac{1}{p} \frac{(\bar{y}_j^{(r),k} - y_j^{(r),k})}{(Q_r(\mathbf{w}^{(r)}) - \lambda_r)^2} \end{aligned} \quad (2.85)$$

Zatem ostatecznie skumulowana reguła delta dla ograniczeń (2.51) i wewnętrznej funkcji kary (2.76) ma postać:

- w warstwie wyjściowej:

$$\Delta w_{ji}^{(R),M} = -\eta \sum_{k=1}^K f'(z_j^{M,k}) \left((\hat{y}_j^{(R),k} - y_j^{(R),k}) + \frac{1}{p} \frac{(\bar{y}_j^{(R),k} - y_j^{(R),k})}{(Q_R(\mathbf{w}^{(R)}) - \lambda_R)^2} \right) \tilde{u}_i^{M-1,k} \quad (2.86a)$$

- w „normalnych” warstwach ukrytych

$$\Delta w_{ji}^{(r),m} = -\eta \sum_{k=1}^K f'(z_j^{m,k}) \sum_{l=1}^{I_{m+1}} \delta_l^{(r),m+1,k} w_{lj}^{(r),m+1} \tilde{u}_i^{m-1,k} \quad (2.86b)$$

- w „dodatkowych” warstwach ukrytych:

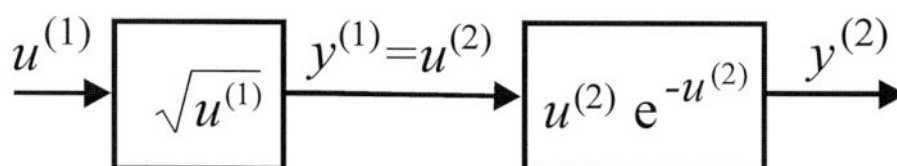
$$\Delta w_{ji}^{(r),m} = -\eta \sum_{k=1}^K f'(z_j^{m,k}) \left(\sum_{l=1}^{I_{m+1}} \delta_l^{(r+1),m+1,k} w_{lj}^{(r+1),m+1} + \frac{1}{P} \frac{(\bar{y}_j^{(r),k} - y_j^{(r),k})}{(Q_r(\mathbf{w}^{(r)}) - \lambda_r)^2} \right) \tilde{u}_i^{m-1,k} \quad (2.86c)$$

W rozdziale tym zostały opracowane algorytmy wyznaczania parametrów sieci wielowarstwowych modelujących obiekty złożone. Algorytm wstecznej propagacji błędu oparty na metodzie gradientowej nie gwarantuje szybkiej zbieżności procesu uczenia. Szybkość uczenia tego algorytmu i jego stabilność zależy od odpowiedniego doboru współczynnika uczenia η . Istnieje szereg skuteczniejszych i szybszych algorytmów uczenia sieci [29, 40, 53] lecz dotyczą one uczenia typowych sieci wielowarstwowych. Nie można tych algorytmów użyć wprost do uczenia sieci o nietypowej strukturze i wielokryterialnych wskaźnikach jakości zaproponowanych w niniejszej pracy. Istnieje jednak możliwość ich implementacji. Przykład takiej implementacji algorytmu heurystycznego, który jest odmianą metody gradientowej, zwanego *Rprop* [44], pozwalającego na znaczne przyspieszenie procesu uczenia podano w niniejszej pracy.

3. Badania symulacyjne

3.1 Plan i zakres badań

W tym rozdziale poprzez badania symulacyjne zostanie sprawdzona poprawność proponowanych sposobów modelowania obiektów złożonych. Na wybranym przykładzie obiektu złożonego zostanie zaproponowana struktura sieci i ilość neuronów w warstwach. Następnie, wykorzystując wcześniej opracowane algorytmy uczenia sieci poszukiwane będą optymalne modele obiektu złożonego: lokalnie optymalny, globalnie optymalny, globalnie optymalny bez uwzględniania modeli częściowych, globalnie optymalny z uwzględnieniem modeli lokalnych przy syntetycznym wskaźniku jakości oraz przy ograniczeniach na modele lokalne.



Rys.3.1 Przykład obiektu złożonego

Na rysunku 3.1 przedstawiono obiekt złożony, który składa się szeregowego połączenia dwóch nieliniowych elementów. Pierwszy element opisany jest funkcją $y^{(1)}(u^{(1)}) = \sqrt{u^{(1)}}$, a drugi - funkcją: $y^{(2)}(u^{(2)}) = u^{(2)} \cdot e^{-u^{(2)}}$.

Wektor uczących danych wejściowych u zawiera 31 elementów, które zostały wygenerowane znormalizowaną funkcją losową w przedziale wartości (0,6). Dla takiego wektora danych wejściowych zostały obliczone wektory: wyjściowy elementu pierwszego $y^{(1)}$ oraz drugiego $y^{(2)}$. Wektory u , $y^{(1)}$ i $y^{(2)}$ wykorzystano do uczenia sieci dla trzech wybranych modeli sieci.

Dane wejściowe testujące zawierają również 31 elementów, rozłożonych równomiernie w przedziale (0,6) z krokiem 0.2, tj. $[0, 0.2, \dots, 6]$. Dla takiego wektora danych wejściowych został obliczony wektor testujących danych wyjściowych.

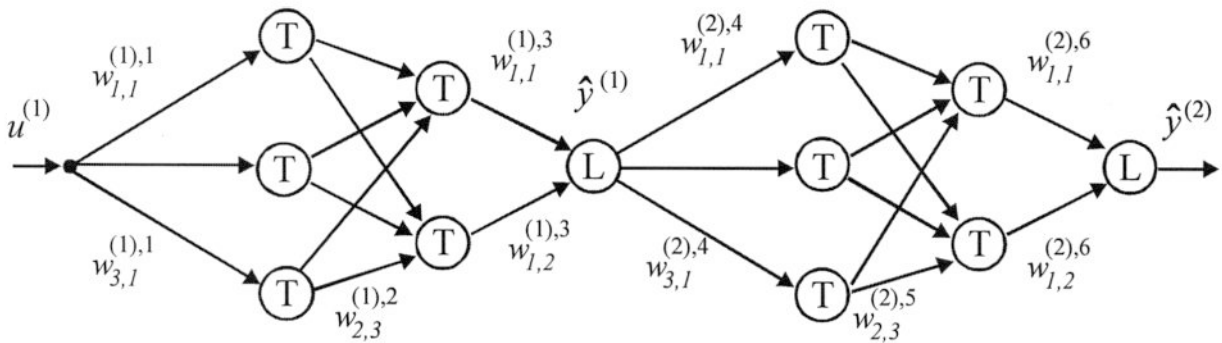
Obiekt złożony zamodelowano sieciami statycznymi wielowarstwowymi wg tabeli 1. Przyjęto trzy modele, których struktury odpowiadają strukturze obiektu. Modele A i B różnią

się tylko tym, że w modelu *B* są skalowane dane uczące. Model *C* ma większą liczbę neuronów w warstwach ukrytych: pierwszej i czwartej oraz inne wartości początkowe wag.

Tab.3.1. Struktura sieci neuronowych dla modelowania obiektu złożonego

Model	Struktura	Skalowanie
<i>A</i>	1-3T-2T-1L-3T-2T-1L	Brak
<i>B</i>	1-3T-2T-1L-3T-2T-1L	Min-Max
<i>C</i>	1-4T-2T-1L-4T-2T-1L	Brak

Kolejne liczby w środkowej kolumnie oznaczają: liczba wejść sieci, liczba neuronów w pierwszej warstwie ukrytej, liczba neuronów w drugiej warstwie ukrytej, liczba neuronów w pierwszej dodatkowej warstwie ukrytej, „T” - oznacza neurony z funkcją aktywacji tangens hiperboliczny, „L” – neurony o liniowej funkcji aktywacji.



Rys.3.2 Schemat modelu w postaci sieci neuronowej dla modelowania obiektu złożonego (model *A*)

Na początku wagi sieci zostały wygenerowane losowo. Następnie do uczenia wykorzystano algorytmy zmodyfikowane podane w rozdziale drugim. Po 1200 cyklach optymalizacyjnych przerwano proces uczenia. Wagi początkowe w poszczególnych modelach były jednakowe dla wszystkich typów modelowania.

Podstawowym kryterium oceny jakości modelowania są funkcje kryterialne zdefiniowane dla poszczególnych modeli w rozdziale drugim. Wszystkie metody modelowania zostaną ocenione według globalnego wskaźnika jakości (2.17), który daje nam obraz modelu jako całości. Drugim kryterium oceny będzie wskaźnik jakości (2.15), który ocenia tylko różnice między wyjściem globalnym modelu a wyjściem zewnętrznym obiektu złożonego. Ocenie podlegać będą modele według lokalnych wskaźników jakości (2.5). Zostaną także obliczone rzeczywiście minimalizowane wskaźniki jakości w poszczególnych modelach.

Dodatkowym kryterium oceny jakości modelowania przyjęto względny błąd procentowy APE:

$$APE = \frac{\sum_{k=1}^K |\bar{y}^k - y^k|}{\sum_{k=1}^K |y^k|} \cdot 100\% \quad (3.1)$$

gdzie $K=31$ - liczba elementów w wektorze wyjściowym, $\bar{y}^k$ - wyjście modelu, y^k - wyjście obiektu.

Kryterium APE pozwala łatwo ocenić jakość modelowania oraz porównywać wielkość błędów niezależnie od ilości i zakresu danych wejściowych i wyjściowych

3.2 Wyniki badań symulacyjnych

Tabele od 3.2 do 3.13 przedstawiają wyniki modelowania obiektu złożonego przedstawionego na rysunku 3.1.

Kolumna *Wskaźnik faktyczny optymalizowany* zawiera wartości kryterium, według którego praktycznie optymalizowano poszczególne modele. Kolumna Q zawiera wskaźnik globalny obliczony według wzoru (2.17) czyli sumę kwadratów błędów modeli częściowych. Kolumna Q_u zawiera sumę kwadratów błędów obliczoną według wzoru (2.15) tylko między wyjściem globalnym modelu złożonego a wyjściem zewnętrznym obiektu złożonego. W kolumnie Q_1 znajduje się wskaźnik jakości modeli lokalnych pierwszego elementu obiektu, obliczony według wzoru (2.5). Kolumna Q_2 zawiera wskaźnik jakości modeli lokalnych drugiego elementu obiektu obliczony również według wzoru (2.5). W tabelach wytłuszczono wskaźniki jakości faktycznie optymalizowane podczas uczenia poszczególnych modeli.

Współczynniki we wzorze (2.17) przyjęto następujące: $\beta_2 = 0.5$, $\beta_1 = 0.5$ oraz we wzorze (2.36) $\alpha_0 = 0.5$, $\alpha_1 = 0.25$, $\alpha_2 = 0.25$.

Tab.3.2. Wskaźniki jakości danych uczących po 1200 cyklach optymalizacji

Lp	Uczenie Model A	Wskaźnik faktycznie optymalizowany	Q	Q_u	Q_2	Q_I
		$\times 10^{-6}$	$\times 10^{-6}$	$\times 10^{-6}$	$\times 10^{-6}$	$\times 10^{-6}$
1	Lokalnie optymalny		2000	414	133	1590
2	Globalnie optymalny	2330	2330	336	313	1990
3	Globalnie optymalny szczególny przypadek	57	13900000	57	268000	13900000
4	Globalnie optymalny szczególny przypadek - syntetyczny wskaźnik jakości	2370	2120	332	256	1790
5	Globalnie optymalny szczególny przypadek - ograniczenia na modele lokalne – zewnętrzna funkcja kary $\lambda_1=2000$ $\lambda_2=200$	118	2120	118	170	2000

Tab.3.3 Wskaźniki jakości danych testujących po 1200 cyklach optymalizacji

Lp	Test Model A	Wskaźnik faktycznie optymalizowany	Q	Q_u	Q_2	Q_I
		$\times 10^{-6}$	$\times 10^{-6}$	$\times 10^{-6}$	$\times 10^{-6}$	$\times 10^{-6}$
1	Lokalnie optymalny		10400	4110	347	6250
2	Globalnie optymalny	10400	10400	3290	441	7090
3	Globalnie optymalny szczególny przypadek	2340	15700000	2340	317000	15700000
4	Globalnie optymalny szczególny przypadek - syntetyczny wskaźnik jakości	10600	10300	3350	308	6760
5	Globalnie optymalny szczególny przypadek - ograniczenia na modele lokalne – zewnętrzna funkcja kary $\lambda_1=2000$ $\lambda_2=200$	2350	9280	2350	791	6930

Tab.3.4 Wskaźniki jakości danych uczących po 1200 cyklach optymalizacji

Lp	Uczenie Model B	Wskaźnik faktycznie optymalizowany	Q	Q_u	Q_2	Q_I
		$\times 10^{-6}$	$\times 10^{-6}$	$\times 10^{-6}$	$\times 10^{-6}$	$\times 10^{-6}$
1	Lokalnie optymalny		2020	768	248	1250
2	Globalnie optymalny	20000	20000	360	2330	19600
3	Globalnie optymalny szczególny przypadek	13	140000000	13	227000	140000000
4	Globalnie optymalny szczególny przypadek - syntetyczny wskaźnik jakości	24000	21300	861	2700	20400
5	Globalnie optymalny szczególny przypadek - ograniczenia na modele lokalne - zewnętrzna funkcja kary $\lambda_1=36000$ $\lambda_2=2250$	3550	39500	3550	2250	35900

Tab.3.5 Wskaźniki jakości dla danych testujących po 1200 cyklach optymalizacji

Lp	Test Model B	Wskaźnik faktycznie optymalizowany	Q	Q_u	Q_2	Q_I
		$\times 10^{-6}$	$\times 10^{-6}$	$\times 10^{-6}$	$\times 10^{-6}$	$\times 10^{-6}$
1	Lokalnie optymalny		10600	5040	1230	5530
2	Globalnie optymalny	18100	18100	391	464	17700
3	Globalnie optymalny szczególny przypadek	3590	159000000	3590	266000	169000000
4	Globalnie optymalny szczególny przypadek - syntetyczny wskaźnik jakości	40400	36600	6800	3760	29800
5	Globalnie optymalny szczególny przypadek - ograniczenia na modele lokalne – zewnętrzna funkcja kary $\lambda_1=36000$ $\lambda_2=2250$	7240	56200	7240	4200	45300

Tab.3.6 Wskaźniki jakości danych uczących po 1200 cyklach optymalizacji

Lp	Uczenie Model C	Wskaźnik faktycznie optymalizowany	Q	Q_u	Q_2	Q_I
		$\times 10^{-6}$	$\times 10^{-6}$	$\times 10^{-6}$	$\times 10^{-6}$	$\times 10^{-6}$
1	Lokalnie optymalny		2990	640	175	2350
2	Globalnie optymalny	2650	2650	103	709	2550
3	Globalnie optymalny szczególny przypadek	5	37800000	5	1980000	37800000
4	Globalnie optymalny szczególny przypadek - syntetyczny wskaźnik jakości	2330	2190	165	146	2020
5	Globalnie optymalny szczególny przypadek - ograniczenia na modele lokalne – zewnętrzna funkcja kary $\lambda_1=3500$ $\lambda_2=300$	190	3690	190	143	3500

Tab.3.7 Wskaźniki jakości po 1200 cyklach optymalizacji

Lp	Test Model C	Wskaźnik faktycznie optymalizowany	Q	Q_u	Q_2	Q_I
		$\times 10^{-6}$	$\times 10^{-6}$	$\times 10^{-6}$	$\times 10^{-6}$	$\times 10^{-6}$
1	Lokalnie optymalny		13500	5110	1310	8350
2	Globalnie optymalny	11900	11900	3200	254	8670
3	Globalnie optymalny szczególny przypadek	3120	42800000	3120	217000	42800000
4	Globalnie optymalny szczególny przypadek - syntetyczny wskaźnik jakości	13700	11300	4200	4450	7060
5	Globalnie optymalny szczególny przypadek – ograniczenia na modele lokalne - zewnętrzna funkcja kary $\lambda_1=3500$ $\lambda_2=300$	5000	16400	5000	1530	11400

Tab.3.8 Błędy względne APE po 1200 cyklach optymalizacji

Lp	Uczenie Model A	Q	Q_u	Q_2	Q_1
		[%]	[%]	[%]	[%]
1	Lokalnie optymalny	1.32	0.92	0.64	0.40
2	Globalnie optymalny	1.41	0.96	0.96	0.46
3	Globalnie optymalny szczególny przypadek	39	0.39	29	38.5
4	Globalnie optymalny szczególny przypadek -syntetyczny wskaźnik jakości	1.24	0.87	0.76	0.37
5	Globalnie optymalny szczególny przypadek - ograniczenia na modele lokalne – zewnętrzna funkcja kary	0.97	0.51	0.66	0.46

Tab.3.9 Błędy względne APE po 1200 cyklach optymalizacji

Lp	Test Model A	Q	Q_u	Q_2	Q_1
		[%]	[%]	[%]	[%]
1	Lokalnie optymalny	2.00	1.47	0.83	0.53
2	Globalnie optymalny	2.09	1.50	1.02	0.59
3	Globalnie optymalny szczególny przypadek	40	1.00	32	39.5
4	Globalnie optymalny szczególny przypadek -syntetyczny wskaźnik jakości	2.08	1.51	0.93	0.57
5	Globalnie optymalny szczególny przypadek - ograniczenia na modele lokalne – zewnętrzna funkcja kary	1.83	1.20	0.90	0.63

Tab.3.10 Błędy względne APE po 1200 cyklach optymalizacji

Lp	Uczenie Model B	Q	Q_u	Q_2	Q_1
		[%]	[%]	[%]	[%]
1	Lokalnie optymalny	1.41	1.12	0.87	0.29
2	Globalnie optymalny	2.17	0.96	1.58	1.22
3	Globalnie optymalny szczególny przypadek	133	0.17	26	133
4	Globalnie optymalny szczególny przypadek -syntetyczny wskaźnik jakości	2.22	0.92	1.43	1.31
5	Globalnie optymalny szczególny przypadek - ograniczenia na modele lokalne – zewnętrzna funkcja kary	3.91	2.21	1.84	1.09

Tab.3.11 Błędy względne APE po 1200 cyklach optymalizacji

Lp	Test Model B	Q	Q_u	Q_2	Q_l
		[%]	[%]	[%]	[%]
1	Lokalnie optymalny	1.96	1.62	1.19	0.34
2	Globalnie optymalny	2.19	1.05	0.91	1.14
3	Globalnie optymalny szczególny przypadek	135	0.92	27	134
4	Globalnie optymalny szczególny przypadek -syntetyczny wskaźnik jakości	2.82	1.51	1.31	1.31
5	Globalnie optymalny szczególny przypadek - ograniczenia na modele lokalne – zewnętrzna funkcja kary	4.25	2.48	1.96	1.77

Tab.3.12 Błędy względne APE po 1200 cyklach optymalizacji

Lp	Uczenie Model C	Q	Q_u	Q_2	Q_l
		[%]	[%]	[%]	[%]
1	Lokalnie optymalny	1.44	0.96	0.71	0.48
2	Globalnie optymalny	1.02	0.53	1.03	0.49
3	Globalnie optymalny szczególny przypadek	65.4	0.11	84	65.3
4	Globalnie optymalny szczególny przypadek -syntetyczny wskaźnik jakości	1.11	0.64	0.63	0.48
5	Globalnie optymalny szczególny przypadek - ograniczenia na modele lokalne – zewnętrzna funkcja kary	1.31	0.72	0.62	0.60

Tab.3.13 Błędy względne APE po 1200 cyklach optymalizacji

Lp	Test Model C	Q	Q_u	Q_2	Q_l
		[%]	[%]	[%]	[%]
1	Lokalnie optymalny	2.21	1.69	1.14	0.62
2	Globalnie optymalny	1.84	1.23	0.78	0.62
3	Globalnie optymalny szczególny przypadek	67.1	0.77	88	66.4
4	Globalnie optymalny szczególny przypadek -syntetyczny wskaźnik jakości	1.82	1.24	1.16	0.58
5	Globalnie optymalny szczególny przypadek - ograniczenia na modele lokalne – zewnętrzna funkcja kary	2.42	1.61	1.04	0.81

W tabeli 3.14 znajdują się wskaźniki jakości po 5000 cyklach optymalizacji dla modelu lokalnie optymalnego i modelu globalnie optymalnego z ograniczeniami na modele lokalne – wewnętrzna funkcja kary. Wyniki zamieszczone zostały osobo ponieważ sposób postępowania w poszukiwaniu optymalnego modelu przy użyciu wewnętrznej funkcji kary jest inny niż w pozostałych metodach. W metodzie wewnętrznej funkcji kary parametry modelu muszą być tak dobrane aby wskaźniki jakości modeli lokalnych były mniejsze od dopuszczalnych wartości. Aby uzyskać takie parametry można wykorzystać parametry otrzymane w modelu lokalnie optymalnym. Następnie, celu minimalizacji globalnego wskaźnika jakości model poddaje się dalszej optymalizacji metodą wewnętrznej funkcji kary.

W przeprowadzonych badaniach symulacyjnych początkowe wartości parametrów wzięto z modelu lokalnie optymalnego po 1200 cyklach optymalizacji (model *A*). Następnie modele: lokalnie optymalny i globalnie optymalny – z ograniczeniami na modele lokalne – wewnętrzna funkcja kary poddano dodatkowej optymalizacji przez 3800 cykli. Łączna suma cykli optymalizacji wynosi 5000.

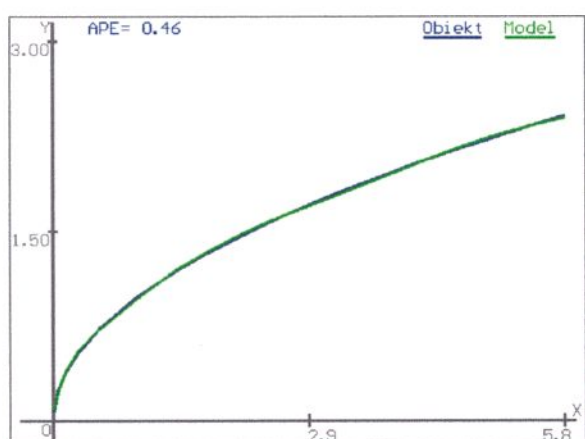
Tab.314 Wskaźniki jakości danych uczących po 5000 cyklach optymalizacji

Lp	Uczenie	Wskaźnik faktycznie optymalizowany	Q	Q_u	Q_2	Q_I
	Model <i>A</i>	$\times 10^{-6}$	$\times 10^{-6}$	$\times 10^{-6}$	$\times 10^{-6}$	$\times 10^{-6}$
1	Lokalnie optymalny		1350	240	40	1100
2	Globalnie optymalny szczególny przypadek – ograniczenia na modele lokalne - wewnętrzna funkcja kary $\lambda_1=1600$ $\lambda_2=135$	298	1680	298	38	1380

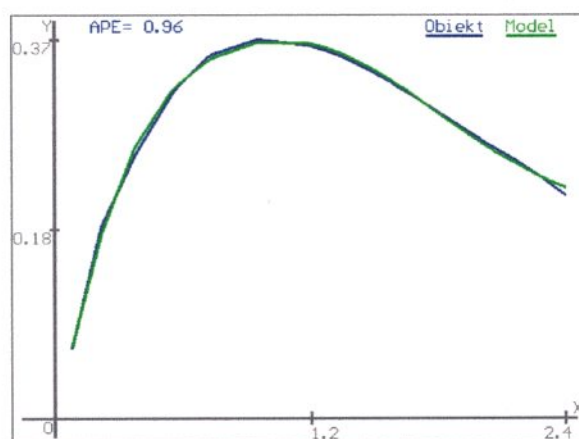
Wybrane charakterystyki obiektu i jego modeli dla danych uczących

Poniższe wykresy przedstawiają charakterystyki symulacyjnego obiektu złożonego i jego modelu otrzymane po 1200 cyklach optymalizacji. Przedstawiono wyniki modelowania w postaci graficznej dla sieci typu *A*, dla danych uczących dwóch wybranych modeli: modelu globalnie optymalnego i modelu globalnego – szczególny przypadek. Wykresy zostały wykreślone dla posortowanych danych uczących.

Model globalnie optymalny

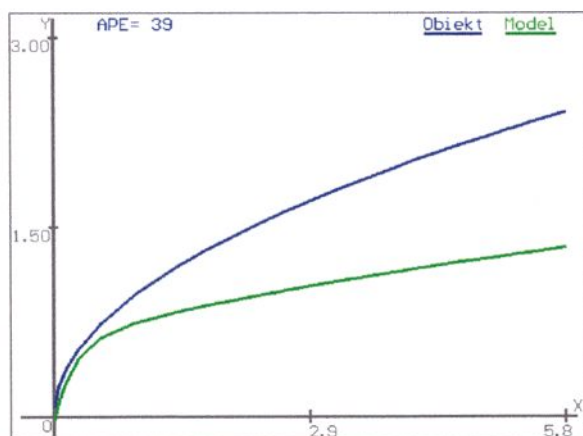


Rys. 3.3 Wyjście pierwszego elementu modelu i obiektu w funkcji jego wejścia

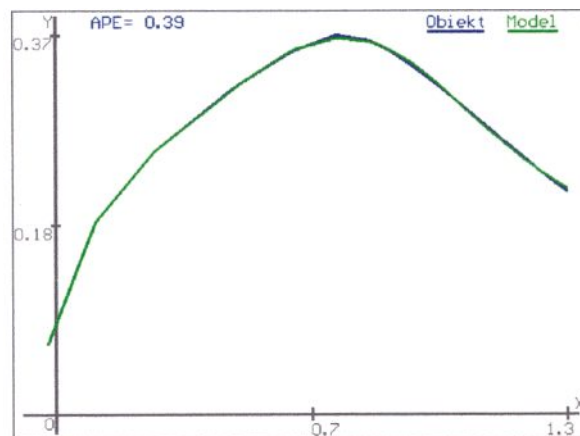


Rys. 3.4 Wyjście drugiego elementu modelu i obiektu w funkcji jego wejścia

Model globalnie optymalny- szczególny przypadek



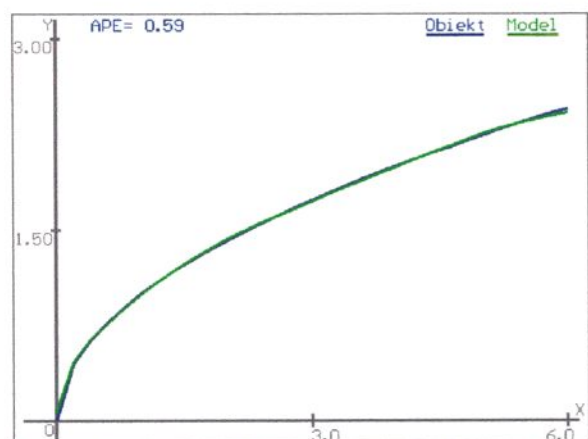
Rys. 3.5 Wyjście pierwszego elementu modelu i obiektu w funkcji jego wejścia



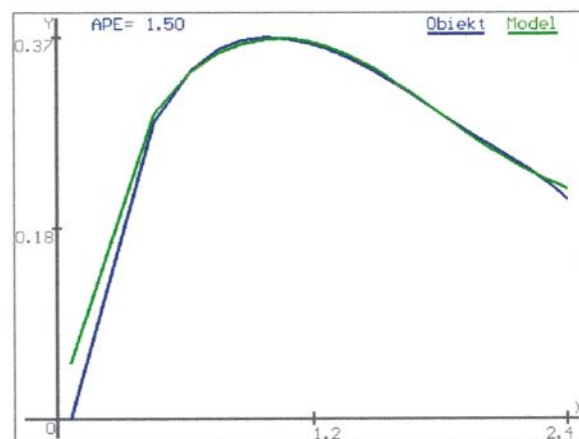
Rys. 3.6 Wyjście drugiego elementu modelu i obiektu w funkcji jego wejścia

Wybrane charakterystyki obiektu i jego modeli dla danych testujących

Model globalnie optymalny

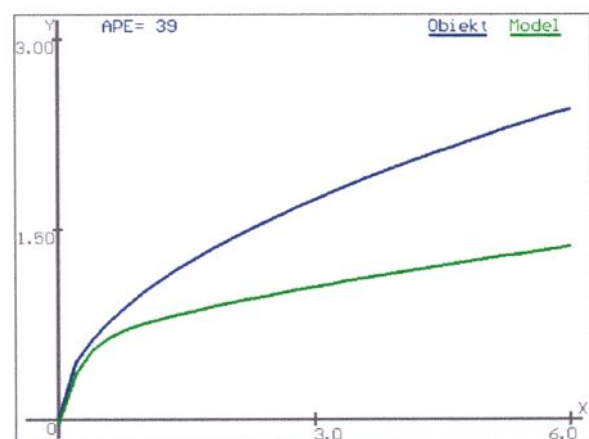


Rys. 3.7 Wyjście pierwszego elementu modelu i obiektu w funkcji jego wejścia

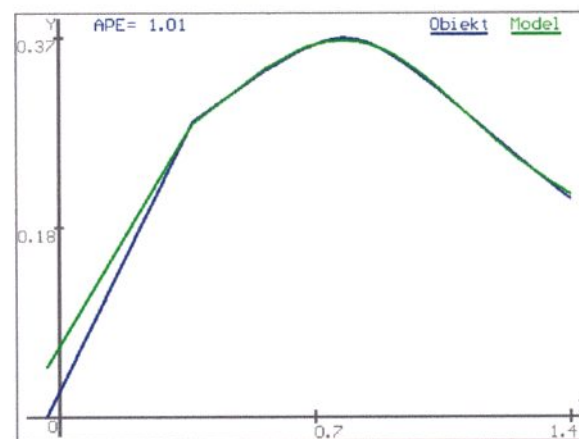


Rys. 3.8 Wyjście drugiego elementu modelu i obiektu w funkcji jego wejścia

Model globalnie optymalny - szczególny przypadek



Rys. 3.9 Wyjście pierwszego elementu modelu i obiektu w funkcji jego wejścia



Rys. 3.10 Wyjście drugiego elementu modelu i obiektu w funkcji jego wejścia

3.3 Omówienie wyników

Optymalizację wszystkich modeli przerwano po 1200 cyklach optymalizacyjnych, tzn. 1200 razy modyfikowano parametry sieci po zaprezentowaniu pełnego ciągu uczącego. Po tylu cyklach gradient wskaźników jakości był niewielki i poprawki parametrów były bardzo małe. Dalsze uczenie nie poprawiło już istotnie jakości wszystkich modeli.

Wyniki dla danych uczących znajdują się w tabelach 3.2, 3.4, 3.6, 3.8, 3.10, 3.12, a dla danych testujących znajdują się w tabelach 3.3, 3.5, 3.7, 3.9, 3.11, 3.13.

Dla skrócenia opisu wniosków i większej ich przejrzystości wprowadźmy następujące oznaczenia:

- LO – model lokalnie optymalny
- GO - model globalnie optymalny
- GSP - model globalnie optymalny – szczególny przypadek
- GWS - model lokalnie optymalny z ograniczeniami na modele lokalne –syntetyczny wskaźnik jakości
- GZK - model lokalnie optymalny z ograniczeniami na modele lokalne – zewnętrzna funkcja kary

W tabelach przedstawione zostało faktyczne kryterium, według którego optymalizowane były modele, lecz ono nie mówi w pełni o jakości modelu.

Kryterium, które mówi o jakości całego modelu złożonego jest kryterium globalne Q . Według tego kryterium dla danych uczących najmniejsze wartości uzyskał model lokalnie optymalnie (LO) oraz model globalnie optymalny z uwzględnieniem jakości modeli lokalnych – syntetyczny wskaźnik jakości (GWS). Zdecydowanie największy wskaźnik (błąd) posiada model globalnie optymalny szczególny przypadek (GSP), co jest zrozumiałe, ponieważ modele częściowe nie były optymalizowane. Pozostałe modele osiągają wartości nieco większe od minimalnego. Można też zauważyć, że dla danych uczących, w modelu globalnie optymalnym z uwzględnieniem jakości modeli lokalnych – z ograniczeniami na modele lokalne przy zewnętrznej funkcji kary (GZK), wskaźnik jakości osiąga większe wartości niż w pozostałych w modelach, z wyjątkiem modelu GSP. Wynika to właśnie z narzuconych ograniczeń na jakość modeli lokalnych, które powodują ograniczenie obszaru poszukiwań optymalnych parametrów. Zaletą zaś jest to, że modele lokalne spełniają narzucone im ograniczenia dopuszczalnych błędów.

Dla danych testujących wskaźnik Q osiąga najmniejsze wartości w modelach LO, GWS i GZK. W modelach globalnie optymalnych, chociaż wskaźnik Q nie osiągał najmniejszych wartości dla rozważanych typów sieci, to był on z reguły niewiele większy od minimalnego.

Oceniając jakość modeli według błędów obliczonych tylko na wyjściu według kryterium Q_u to dla danych uczących najmniejsze błędy uzyskiwały modele globalnie optymalne-szczególny przypadek. Drugi wynik, czyli nieco większe wskaźniki Q_u uzyskały modele globalnie optymalne. Modele lokalnie optymalne posiadały największe wartości wskaźnika

jakości Q_u . Dla danych testujących najmniejsze wartości błędów posiadały także modele globalnie optymalne - szczególny przypadek, oraz modele globalnie optymalne.

Najmniejsze wartości wskaźników modeli lokalnych Q_1 i Q_2 zarówno dla danych uczących jak i testujących posiadały z reguły modele lokalnie optymalne. Jest to uzasadnione, gdyż w tych modelach dokonywano optymalizacji lokalnej. W pozostałych modelach mniejsze wskaźniki niż pozostałe (oprócz LO) posiadają modele GWS. Można to uzasadnić tym, że w skład rzeczywiście optymalizowanego kryterium wchodziły wskaźniki jakości modeli lokalnych. Miały więc one wpływ na takie wyznaczenie parametrów modelu globalnego, dla których wskaźniki jakości modeli są nieco mniejsze niż w pozostałych modelach globalnych.

Tabele od 3.8 do 3.13, które zawierają z błędy APE dają możliwość oszacowania, na ile poszczególne modele różnią się od modelowanego obiektu. Dla danych uczących dla wskaźnika Q_u , błędy wahają się od 0.11 do 0.96 procent (tylko raz 2.21 procent). Dla danych testujących błędy wahają się od 0.77 do 1.61 procent (tylko raz osiąga najgorszy wynik 2.28 procenta) dla kryterium Q_u . Maksymalne i minimalne wartości błędów APE w poszczególnych modelach korespondują z wartościami kryteriów jakości zawartych w tabelach 3.2-3.7.

4. Modelowanie procesu produkcji azotynu amonu

W rozdziale tym opisane zostanie modelowanie fragmentu procesu produkcji azotynu za pomocą wcześniej opisanych metod. Dane do modelowania pochodzą z przemysłowej linii produkcyjnej z Zakładów Azotowych w Tarnowie.

Celem autora jest opracowanie optymalnych modeli procesu przemysłowego proponowanymi w rozdziale drugim metodami. Zakres modelowania obejmuje: model lokalnie optymalny, model globalnie optymalny, model globalnie optymalny – szczególnie przypadek, model globalnie optymalny z uwzględnieniem jakości modeli lokalnych – przy syntetycznym wskaźniku jakości i dla ograniczeń na modele lokalne.

4.1 Opis procesu

Całość procesu otrzymywania azotynu amonu składa się z trzech faz

1. Spalanie amoniaku
 - a) pobieranie i przygotowanie surowców
 - b) otrzymywanie mieszanki amoniakalno - powietrznej i utlenianie amoniaku
2. Absorpcja tlenków azotu w roztworze węglanu amonu
 - a) mieszanie części strumienia gazów nitrozowych po kondensatorze z tlenem,
 - b) dotlenianie tlenku do dwutlenku azotu,
 - c) mieszanie dotlenionych gazów nitrozowych z gazami procesowymi przed drugą i trzecią wieżą absorpcyjną,
 - d) absorpcja tlenków azotu.
3. Otrzymywanie roztworu NH_3 lub/i CO_2 w wodzie

Mieszanka amoniaku i powietrza o zawartości 10-11.5 [% obj. NH_3] utlenia się katalitycznie w reaktorze na siatkach platynowo – rodowych i PdAuRh w temperaturze ok. 1093.15 [K] (820 °C) pod ciśnieniem ok. 50 kPa. Metoda utleniania NH_3 jest metodą klasyczną powszechnie stosowaną w instalacjach kwasu azotowego bezciśnieniowego. Układ spalania jednostki AAI/2 może współpracować z dowolnym ciągiem absorpcyjnym.

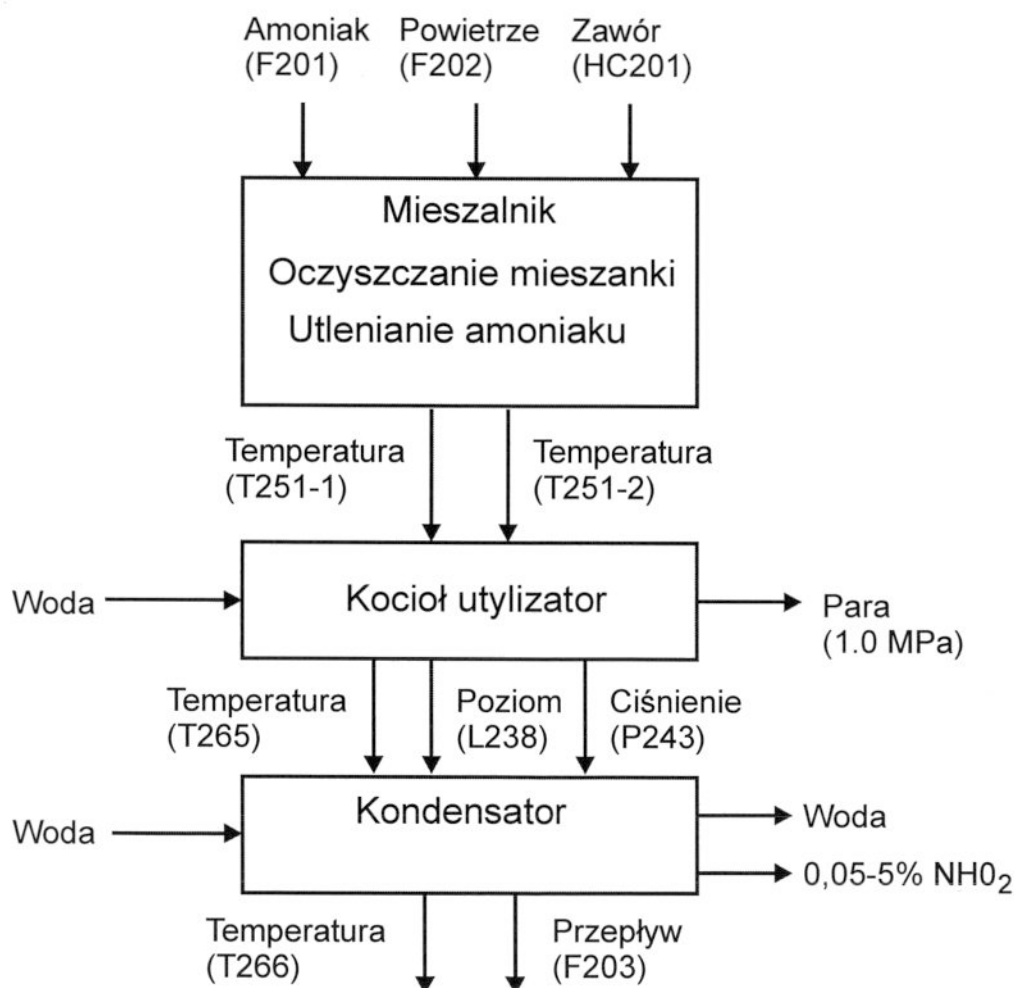
Gazy nitrozowe otrzymane w wyniku utleniania amoniaku schładza się w kondensatorze, co powoduje kondensację pary wodnej poreakcyjnej i powstanie wodnego roztworu HNO_3 . Główny strumień gazów nitrozowych (90% obj.) kieruje się do trójstopniowego węzła absorpcji. Absorpcję prowadzi się w temperaturze 273.15-293.15 [K] (0-20 °C). Ciepło absorpcji odbiera się przez schładzanie roztworu cyrkulacyjnego w chłodnicach amoniakalnych.

Wydzielona (ok. 10% obj.) część strumienia gazów procesowych po kondensatorze kierowana jest do dotleniacza, w którym gazy nitrozowe dotlenia się tlenem. Następnie część dotlenionych gazów kierowana jest do mieszalnika przed II wieżą, a pozostałą część do mieszalnika przed III wieżą absorpcyjną gdzie mieszają się ze strumieniem gazów procesowych.

Węglan amonu powstaje poprzez absorpcję amoniaku i dwutlenku węgla w wodzie. Ciepło reakcji, które składa się z ciepła absorpcji NH_3 oraz ciepła absorpcji CO_2 odbierane jest przez chłodzenie roztworu cyrkulacyjnego w chłodnicy wodnej.

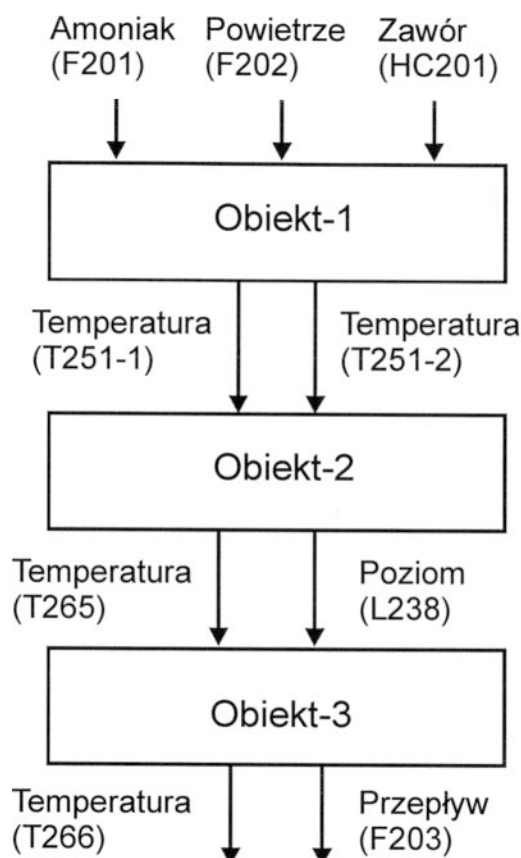
4.2 Dane uczące i testujące

Dane z fragmentu instalacji produkcji azotynu amonu zostały użyte do modelowania procesu zarejestrowane w głównej sterowni instalacji. Dane te obejmują cztery kolejne dni (17.08.2000 - 20.08.2000). z raportów chwilowych, rejestrowanych co dwie godziny. Modelowany fragment przedstawiony jest na rysunku 4.1.



Rys.4.1 Fragment instalacji produkcji azotynu amonu

Na rysunku 4.2 został przedstawiony schemat blokowy fragmentu instalacji z rys.4.1. W schemacie blokowym pominięto niemierzalne wielkości (woda, para, NHO_2), potraktowano je jako stałe zakłócenia. Pominięto też pomiar ciśnienia (P243), ponieważ było utrzymywane na stałym poziomie przez inną instalację zewnętrzną. Nazwy poszczególnych urządzeń instalacji zastąpiono obiektami z kolejnymi numerami.



Rys.4.2 Schemat blokowy procesu produkcji azotynu amonu

Przygotowanie danych do uczenia i testowania modelu.

Plik danych uczących powstał przez połączenie danych z dwóch dni (17.08.2000 i 18.08.2000), zawierają one po 24 elementy. Plik danych testujących powstał przez połączenie danych także z dwóch dni (19.08.2000 i 20.08.2000).

- ❖ Dane wejściowe obiektu pierwszego składają się z trzech sygnałów: $u_1^{(1)}$ - przepływu amoniaku (F201), $u_2^{(1)}$ - przepływu powietrza (F202), $u_3^{(1)}$ - stosunek (iloraz) NH_3 /powietrze (HC201)
- ❖ Dane wyjściowe obiektu pierwszego składają się z dwóch sygnałów, temperatur na siatkach katalizatora: $y_1^{(1)}$ - temperatura (T251-1) oraz $y_2^{(1)}$ - temperatura (T251-2). Dane

z wyjścia obiektu pierwszego są jednocześnie wejściem dla obiektu drugiego lub połączonego drugiego i trzeciego.

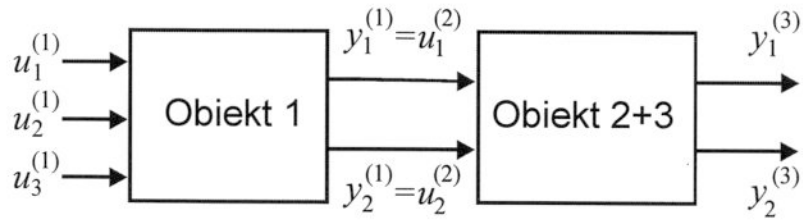
- ❖ Dane wyjściowe obiektu drugiego składają się z trzech sygnałów: $y_1^{(2)}$ - temperatura w kotle (T265), $y_2^{(2)}$ - poziom ciekłego roztworu w kotle (L238), $y_3^{(2)}$ - ciśnienie w kotle utylizatorze (P243). Dane z wyjścia obiektu drugiego są też wejściem do obiektu trzeciego.
- ❖ Dane wyjściowe do obiektu trzeciego składają się z dwóch sygnałów: $y_1^{(3)}$ - temperatura gazów po kondensatorze (T266), $y_2^{(3)}$ - przepływ strumienia gazów do kolumny absorpcyjnej (F203).

4.3 Modele procesu i wyniki modelowania

Dla procesu produkcji azotynu amonu przedstawionego na rysunku 4.1 zaproponowano trzy warianty modelowania. Wariant „I” obejmuje cały proces ale nie uwzględnia sygnałów na wyjściu obiektu drugiego (temperatury T265 i poziomu L238). Dla tego wariantu schemat blokowy procesu ma dwa elementy (rys.4.3). Wariant „II” obejmuje tylko dwa pierwsze obiekty procesu (rys.4.2), nie uwzględniamy obiektu trzeciego, dlatego schemat blokowy w tym wariantcie także ma dwa elementy (rys.4.14). Wariant „III” obejmuje cały proces, składający się z trzech elementów (rys.4.25). Każdemu wariantowi obiektu złożonego zaproponowano modele w postaci sieci neuronowych, których struktura blokowa odpowiada strukturze schematów blokowych poszczególnych wariantów.

4.3.1 Modelowanie procesu – wariant „I”

Rysunek 4.3 przedstawia uproszczony model fragmentu instalacji wytwarzającej azotyn amonu, składający się z dwóch elementów. Na pierwszy element składa się mieszalnik amoniaku z powietrzem oraz utlenianie tej mieszanki w kotle spalania, wyjściem tego obiektu jest temperatura na siatkach. Na drugi obiekt składa się kocioł utylizator i kondensator chłodnicy, wyjściem tego elementu jest temperatura gazów nitrozowych (T266) oraz ich przepływ (F203).



Rys.4.3 Wariant „I”. Obiekt złożony jako połączenie obiektu pierwszego oraz trzeciego z drugim

Obiekt złożony, wariant „I”, przedstawiony na rysunku 4.3 zamodelowano neuronową siecią statyczną, wielowarstwową o strukturze jak w tabeli 4.1. Struktura modelu (tj. sieci) odpowiada strukturze modelowanego obiektu złożonego. Początkowe wagi sieci wygenerowano losowo. Do uczenia wykorzystano algorytmy zmodyfikowane zwane *Rprop*. Uczenie przerwano po 400 cyklach optymalizacji.

Wszystkie modele miały jednakowe wagi początkowe. Również trzy typy modeli: *A*, *B*, *C* również miały te same wagi początkowe, lecz inny sposób skalowania danych. Zakres danych z procesu był bardzo duży, wartości danych mieściły się w zakresie od 10^{-1} do 10^4 . Neurony z warstw ukrytych posiadały nieliniowe funkcje aktywacji: tangens hiperboliczny, które nasycają się, jeżeli argumenty funkcji przekraczają wartość jeden, dlatego aby temu zapobiec wszystkie dane w trakcie uczenia sieci i testowania zostały przeskalowane do przedziału $[0,1]$ (skalowanie „Min-Max”) lub w pobliże tego przedziału (skalowanie „Max”, „Średnie”).

Skalowanie „Max” polegało na podzieleniu danych wejściowych i wyjściowych przez wartości maksymalne odpowiednie dla tych danych.

$$y_j^{(s)} = \frac{y_j}{y_{\max}} \quad (4.1)$$

gdzie: $y_j^{(s)}$ j -ta składowa wektora y po przeskalowaniu, y_j aktualna wartość j -tego elementu wektora, $y_{\max}$ wartość największego elementu wektora.

Skalowanie „Średnie” polegało na dzieleniu danych wejściowych i wyjściowych przez ich wartości średnie:

$$y_j^{(s)} = \frac{y_j}{\frac{1}{J} \sum_{j=1}^J y_j} \quad (4.2)$$

gdzie: J – liczba elementów wektora y .

Skalowanie „Min-Max” danych wejściowych i wyjściowych odbywało się według następującego wzoru:

$$y_j^{(s)} = \frac{y_j - y_{\min}}{y_{\max} - y_{\min}} \quad (4.3)$$

gdzie: $y_{\min}$ - wartość minimalnego elementu wektora,

Wszystkie dane miały wartości dodatnie, dlatego przeskalowane zostały do przedziału [0,1].

Wyniki uczenia i testowania zawarto w tabelach od 4.2 do 4.18, obliczając dla poszczególnych metod wskaźniki jakości oraz błędy względne APE wg. (3.1).

Początkowe wartości parametrów do optymalizacji modelu globalnego z ograniczeniami na modele lokalne – wewnętrzna funkcja kary, pochodzą z modelu lokalnie optymalnego po 400 cyklach optymalizacji. Dodatkowa liczba cykli optymalizacji dla modeli zawartych w tabeli 4.19 wyniosła 100.

Tab.4.1. Struktura sieci neuronowych dla modelowania obiektu – wariant „I”

Nazwa typu modelu	Skalowanie	Struktura
<i>A</i>	Min-Max	3-9T-3T-2L-9T-3T-2L
<i>B</i>	Max	3-9T-3T-2L-9T-3T-2L
<i>C</i>	Średnie	3-9T-3T-2L-9T-3T-2L
<i>D</i>	Min-Max	3-7T-4T-2L-5T-3T-2L

W kolumnie „struktura” w tabeli 4.1 zostały zawarte dane dotyczące zaproponowanych typów modeli (ilość warstw, ilość neuronów w poszczególnych warstwach, oraz rodzaj funkcji aktywacji w warstwach) zgodnie z opisem zamieszczonym pod tabelą 3.1.

Tab.4.2 Wskaźniki jakości po 400 cyklach optymalizacji

L P	Uczenie	Wskaźnik faktycznie optymalizowany	Q	Q_u	Q_2	Q_I
	Model <i>A</i>					
1	Lokalnie optymalny		23700	23200	13200	491
2	Globalnie optymalny	4100	4100	3413	43400	690
3	Globalnie optymalny szczególny przypadek	1370	681000	1370	218000	679000
4	Globalnie optymalny szczególny przypadek - syntetyczny wskaźnik jakości	56200	17700	16300	38500	1220
5	Globalnie optymalny szczególny przypadek - ograniczenia na modele lokalne – zewnętrzna funkcja kary $\lambda_1=910$ $\lambda_2=40000$	16700	17600	16700	27900	904

Tab.4.3 Wskaźniki jakości po 400 cyklach optymalizacji

L P	Test	Wskaźnik faktycznie optymalizowany	Q	Q_u	Q_2	Q_1
	Model A					
1	Lokalnie optymalny		27300	24600	51200	3740
2	Globalnie optymalny	57300	57300	55700	56100	1580
3	Globalnie optymalny szczególny przypadek	23100	970000	23100	363000	968000
4	Globalnie optymalny szczególny przypadek - syntetyczny wskaźnik jakości	15500	83800	80100	71700	3680
5	Globalnie optymalny szczególny przypadek - ograniczenia na modele lokalne – zewnętrzna funkcja kary $\lambda_1=910$ $\lambda_2=40000$	36300	38500	36300	58300	2130

Tab.4.4 Wskaźniki jakości po 400 cyklach optymalizacji

L P	Uczenie	Wskaźnik faktycznie optymalizowany	Q	Q_u	Q_2	Q_1
	Model B					
1	Lokalnie optymalny		47000	43200	30800	3700
2	Globalnie optymalny	39700	39700	36000	38600	3710
3	Globalnie optymalny szczególny przypadek	25100	125000000	25100	422000000	125000000
4	Globalnie optymalny szczególny przypadek - syntetyczny wskaźnik jakości	77700	36800	33200	40900	3550
5	Globalnie optymalny szczególny przypadek - ograniczenia na modele lokalne – zewnętrzna funkcja kary $\lambda_1=18000$ $\lambda_2=75000$	35200	52400	35200	38500	17200

Tab.4.5 Wskaźniki jakości po 400 cyklach optymalizacji

L P	Test	Wskaźnik faktycznie optymalizowany	Q	Q_u	Q_2	Q_1
	Model B					
1	Lokalnie optymalny		81000	75300	56200	5740
2	Globalnie optymalny	65200	65200	59600	61400	5530
3	Globalnie optymalny szczególny przypadek	30800	120000000	30800	417000000	120000000
4	Globalnie optymalny szczególny przypadek - syntetyczny wskaźnik jakości	120000	55800	51100	64000	4590

Tab.4.5 cd. Wskaźniki jakości po 400 cyklach optymalizacji

L P	Test	Wskaźnik faktycznie optymalizowany	Q	Q_u	Q_2	Q_I
	Model B					
5	Globalnie optymalny szczególny przypadek - ograniczenia na modele lokalne – zewnętrzna funkcja kary $\lambda_1=18000$ $\lambda_2=75000$	47800	52800	47800	63600	4970

Tab.4.6 Wskaźniki jakości po 400 cyklach optymalizacji

L P	Uczenie	Wskaźnik faktycznie optymalizowany	Q	Q_u	Q_2	Q_I
	Model C					
1	Lokalnie optymalny		46800	43700	31100	3100
2	Globalnie optymalny	40400	40400	36500	36400	3960
3	Globalnie optymalny szczególny przypadek	28100	573000000	28100	199000000	573000000
4	Globalnie optymalny szczególny przypadek - syntetyczny wskaźnik jakości	80500	36800	33700	43700	3180
5	Globalnie optymalny szczególny przypadek - ograniczenia na modele lokalne – zewnętrzna funkcja kary $\lambda_1=17500$ $\lambda_2=78000$	38400	55700	38400	41400	17300

Tab.4.7 Wskaźniki jakości po 400 cyklach optymalizacji

L P	Test	Wskaźnik faktycznie optymalizowany	Q	Q_u	Q_2	Q_I
	Model C					
1	Lokalnie optymalny		85600	79600	57600	5950
2	Globalnie optymalny	67200	67200	61200	61400	5950
3	Globalnie optymalny szczególny przypadek	30700	572000000	30700	195000000	572000000
4	Globalnie optymalny szczególny przypadek - syntetyczny wskaźnik jakości	121000	56200	52000	65300	4150
5	Globalnie optymalny szczególny przypadek - ograniczenia na modele lokalne – zewnętrzna funkcja kary $\lambda_1=17500$ $\lambda_2=78000$	52200	56400	52200	64400	4190

Tab.4.8 Wskaźniki jakości po 400 cyklach optymalizacji

L P	Uczenie Model D	Wskaźnik faktycznie optymalizowany	Q	Q_u	Q_2	Q_I
1	Lokalnie optymalny		30300	29800	14400	502
2	Globalnie optymalny	6880	6880	6170	57800	718
3	Globalnie optymalny szczególny przypadek	24500	787000	2450	245000	785000
4	Globalnie optymalny szczególny przypadek - syntetyczny wskaźnik jakości	82800	14900	13900	67900	977
5	Globalnie optymalny szczególny przypadek - ograniczenia na modele lokalne – zewnętrzna funkcja kary $\lambda_1=1220$ $\lambda_2=3600$	15900	17100	15900	23600	1220

Tab.4.9 Wskaźniki jakości po 400 cyklach optymalizacji

L P	Test Model D	Wskaźnik faktycznie optymalizowany	Q	Q_u	Q_2	Q_I
1	Lokalnie optymalny		159000	155000	46600	4120
2	Globalnie optymalny	109000	109000	104000	65800	4720
3	Globalnie optymalny szczególny przypadek	23500	704000	23500	400000	680000
4	Globalnie optymalny szczególny przypadek - syntetyczny wskaźnik jakości	146000	86200	83500	59800	2710
5	Globalnie optymalny szczególny przypadek - ograniczenia na modele lokalne – zewnętrzna funkcja kary $\lambda_1=1120$ $\lambda_2=3600$	20600	22600	20600	54100	1980

Tab.4.10 Błędy względne po 400 cyklach optymalizacji

Lp	Uczenie	Q	Q_u	Q_2	Q_I
	Model A				
		[%]	[%]	[%]	[%]
1	Lokalnie optymalny				
2	Globalnie optymalny	1.32	0.99	0.72	0.33
3	Globalnie optymalny szczególny przypadek	14.1	0.25	3.70	14.0
4	Globalnie optymalny szczególny przypadek -syntetyczny wskaźnik jakości	1.35	0.76	1.28	0.59
6	Globalnie optymalny szczególny przypadek - ograniczenia na modele lokalne – zewnętrzna funkcja kary	1.26	0.81	1.06	0.46

Tab.4.11 Błędy względne po 400 cyklach optymalizacji

Lp	Test	Q	Q_u	Q_2	Q_I
	Model A				
		[%]	[%]	[%]	[%]
1	Lokalnie optymalny	1.73	0.98	1.50	0.75
2	Globalnie optymalny	1.80	1.21	1.54	0.58
3	Globalnie optymalny szczególny przypadek	18.4	0.96	4.82	17.2
4	Globalnie optymalny szczególny przypadek -syntetyczny wskaźnik jakości	2.33	1.50	1.85	0.83
5	Globalnie optymalny szczególny przypadek - ograniczenia na modele lokalne – zewnętrzna funkcja kary	1.66	0.98	1.58	0.68

Tab.4.12 Błędy względne po 400 cyklach optymalizacji

Lp	Uczenie	Q	Q_u	Q_2	Q_I
	Model B				
		[%]	[%]	[%]	[%]
1	Lokalnie optymalny	2.38	1.35	1.09	1.02
2	Globalnie optymalny	2.22	1.19	1.21	1.03
3	Globalnie optymalny szczególny przypadek	207	0.82	168	207
4	Globalnie optymalny szczególny przypadek -syntetyczny wskaźnik jakości	2.09	1.10	1.26	0.99
5	Globalnie optymalny szczególny przypadek - ograniczenia na modele lokalne – zewnętrzna funkcja kary	3.33	1.15	1.21	2.18

Tab.4.13 Błędy względne po 400 cyklach optymalizacji

Lp	Test	Q	Q_u	Q_2	Q_I
	Model B				
		[%]	[%]	[%]	[%]
1	Lokalnie optymalny	2.88	1.78	1.55	1.11
2	Globalnie optymalny	2.62	1.53	1.56	1.09
3	Globalnie optymalny szczególny przypadek	205	1.02	166	205
4	Globalnie optymalny szczególny przypadek -syntetyczny wskaźnik jakości	2.37	1.40	1.58	0.97
5	Globalnie optymalny szczególny przypadek - ograniczenia na modele lokalne – zewnętrzna funkcja kary	4.15	1.47	1.62	2.68

Tab.4.14 Błędy względne po 400 cyklach optymalizacji

Lp	Uczenie	Q	Q_u	Q_2	Q_I
	Model C				
		[%]	[%]	[%]	[%]
1	Lokalnie optymalny	2.43	1.35	1.08	1.08
2	Globalnie optymalny	2.26	1.20	1.20	1.06
3	Globalnie optymalny szczególny przypadek	440	0.92	116	440
4	Globalnie optymalny szczególny przypadek -syntetyczny wskaźnik jakości	2.05	1.11	1.30	0.94
6	Globalnie optymalny szczególny przypadek - ograniczenia na modele lokalne – zewnętrzna funkcja kary	3.12	1.18	1.27	1.94

Tab.4.15 Błędy względne po 400 cyklach optymalizacji

Lp	Test	Q	Q_u	Q_2	Q_I
	Model C				
		[%]	[%]	[%]	[%]
1	Lokalnie optymalny	2.95	1.82	1.57	1.13
2	Globalnie optymalny	2.68	1.55	1.56	1.13
3	Globalnie optymalny szczególny przypadek	444	1.04	116	443
4	Globalnie optymalny szczególny przypadek -syntetyczny wskaźnik jakości	2.35	1.41	1.60	0.94
5	Globalnie optymalny szczególny przypadek - ograniczenia na modele lokalne – zewnętrzna funkcja kary	2.47	1.46	1.58	1.00

Tab. 4.16 Błędy względne po 400 cyklach optymalizacji

Lp	Uczenie Model D	Q	Q_u	Q_2	Q_1
		[%]	[%]	[%]	[%]
1	Lokalnie optymalny	1.38	1.03	0.75	0.35
2	Globalnie optymalny	0.94	0.53	1.59	0.41
3	Globalnie optymalny szczególny przypadek	15.5	0.31	3.94	15.2
4	Globalnie optymalny szczególny przypadek -syntetyczny wskaźnik jakości	1.28	0.80	1.66	0.48
5	Globalnie optymalny szczególny przypadek - ograniczenia na modele lokalne – zewnętrzna funkcja kary	1.19	0.67	1.02	0.52

Tab. 4.17 Błędy względne po 400 cyklach optymalizacji

Lp	Test Model D	Q	Q_u	Q_2	Q_1
		[%]	[%]	[%]	[%]
1	Lokalnie optymalny	2.97	2.12	1.42	0.85
2	Globalnie optymalny	2.66	1.71	1.76	0.95
3	Globalnie optymalny szczególny przypadek	15.1	0.86	5.10	14.2
5	Globalnie optymalny szczególny przypadek -syntetyczny wskaźnik jakości	2.51	1.71	1.65	0.80
6	Globalnie optymalny szczególny przypadek - ograniczenia na modele lokalne – zewnętrzna funkcja kary	1.50	0.79	1.54	0.70

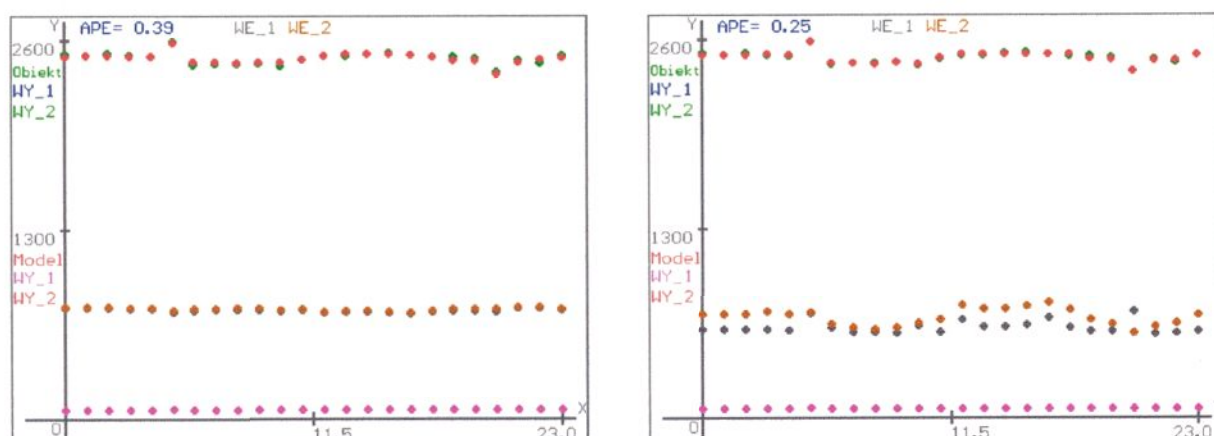
Tab. 4.18 Wskaźniki jakości po 500 cyklach optymalizacji

Lp	Uczenie Model A	Wskaźnik faktycznie optymalizowany	Q	Q_u	Q_2	Q_1
1	Lokalnie optymalny		23200	22300	12700	478
2	Globalnie optymalny szczególny przypadek - ograniczenia na modele lokalne – wewnętrzna funkcja kary $\lambda_1=910$ $\lambda_2=40000$	10700	11200	10700	22300	495

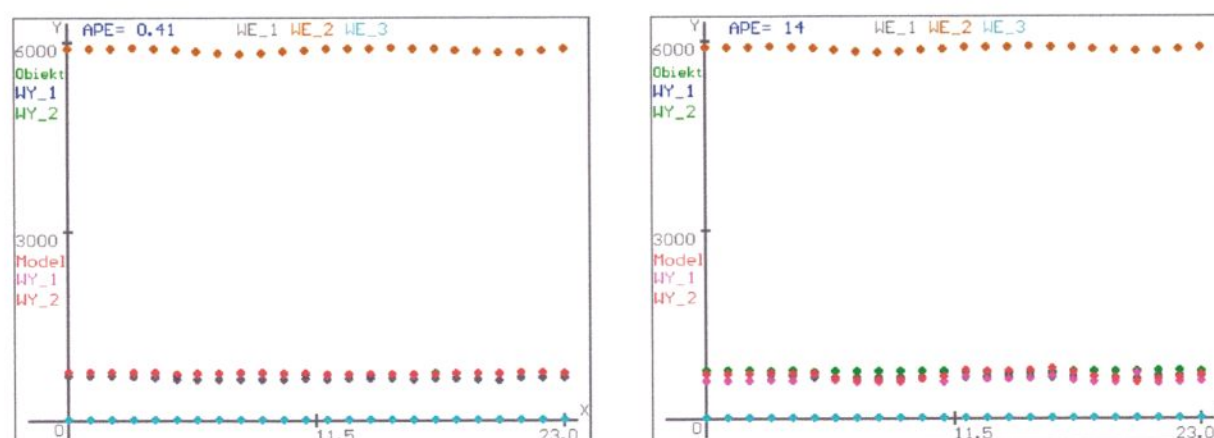
Przebiegi sygnałów uczących i testujących dla procesu wariant „I” oraz w jego modelu

Do ilustracji w postaci graficznej wybrano przebiegi w modelu sieciowym typ A, po 400 cyklach optymalizacyjnych. W kolumnie lewej zostały przedstawione wyniki uzyskane w modelu globalnie optymalnym. W prawej kolumnie wyniki z modelu globalnie optymalnego – szczególny przypadek. Przedstawiono dane wejściowo wyjściowe obiektu i procesu w skali 1:1. Następnie dla przedstawiono przebiegi wyjściowe poszczególnych elementów w skali zbliżonej do ich wartości nominalnych dla zaobserwowania różnic między wyjściem modeli a wyjściem obiektów.

Model globalnie optymalny

Model globalnie optymalny –
szczególny przypadek***Dane uczące***

***Rys.4.4 Wejścia i wyjścia obiektu i modelu złożonego
(WY_1 – temperatura T266, WY_2 – przepływ F03)***

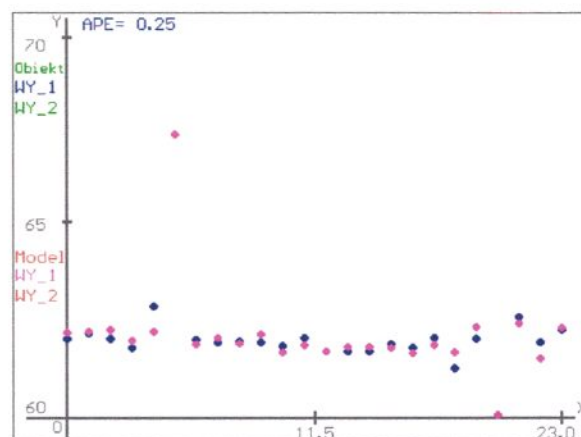
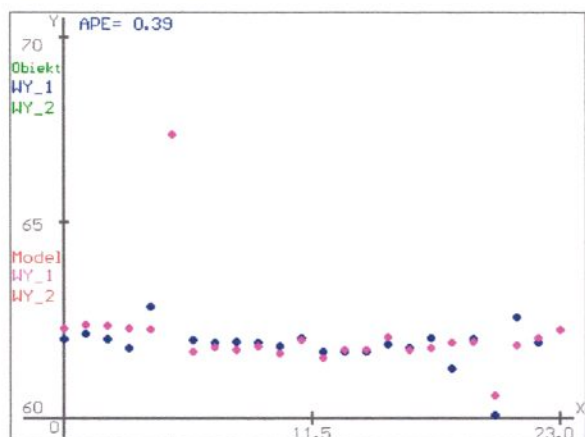


***Rys.4.5 Wejścia i wyjścia pierwszego elementu obiektu i modelu
(WY_1 – temperatura T251-1, WY_2 – temperatura T251-2)***

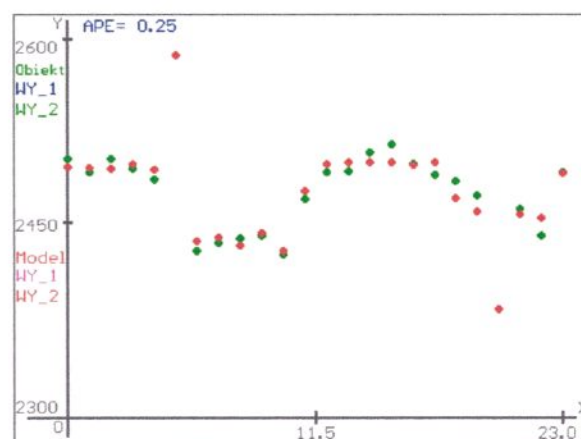
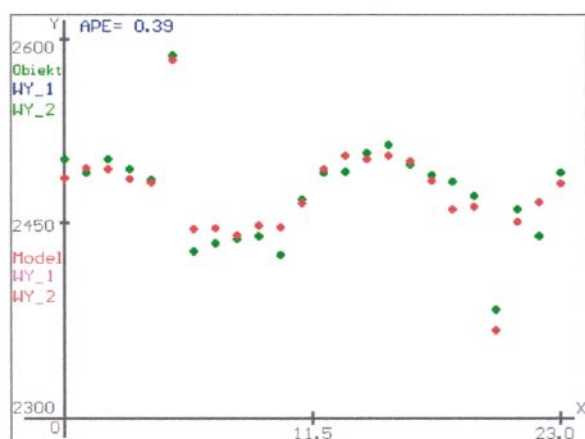
Model globalnie optymalny

Model globalnie optymalny –
szczególny przypadek

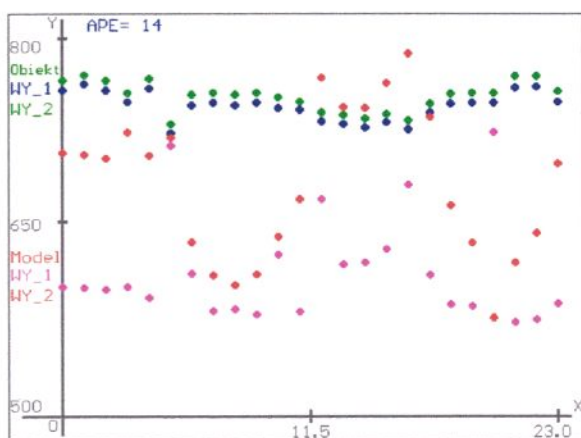
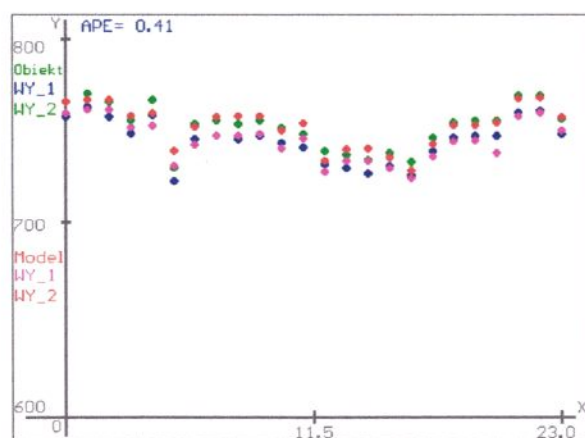
Dane uczące



Rys. 4.6 Wyjście obiektu i modelu złożonego
(WY_1 – temperatura T266)



Rys. 4.7 Wyjście pierwszego elementu obiektu i modelu
(WY_2 – przepływ F03)

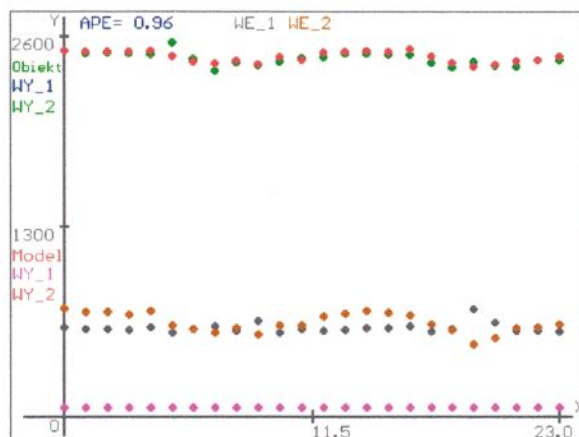
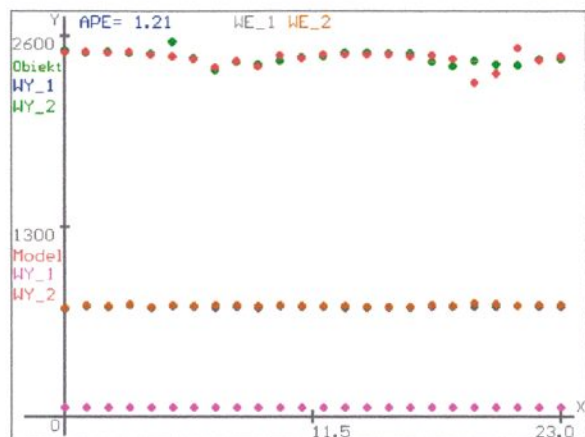


Rys. 4.8 Wyjścia pierwszego elementu obiektu i modelu
(WY_1 – temperatura T251-1, WY_2 – temperatura T251-2)

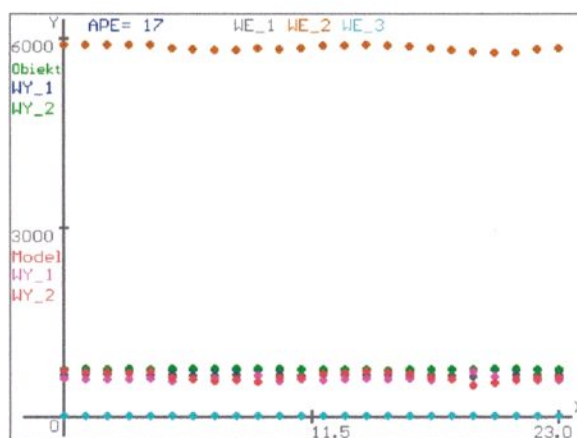
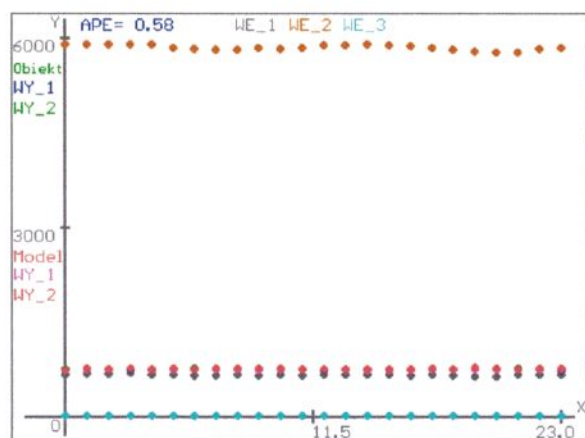
Model globalnie optymalny

Model globalnie optymalny –
szczególny przypadek

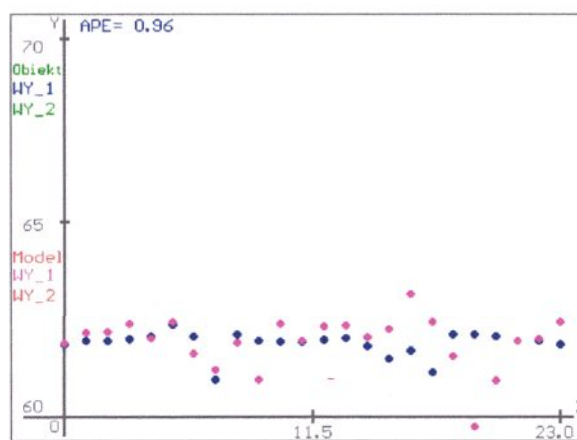
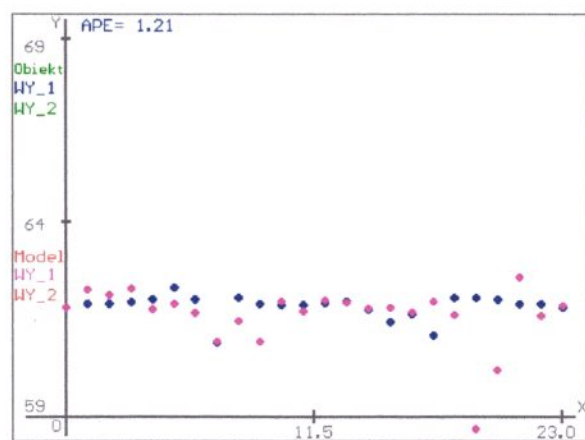
Dane testujące



*Rys. 4.9 Wejścia i wyjścia obiektu i modelu złożonego
(WY_1 – temperatura T266, WY_2 – przepływ F03)*



*Rys. 4.10 Wejścia i wyjścia pierwszego elementu obiektu i modelu
(WY_1 – temperatura T251-1, WY_2 – temperatura T251-2)*

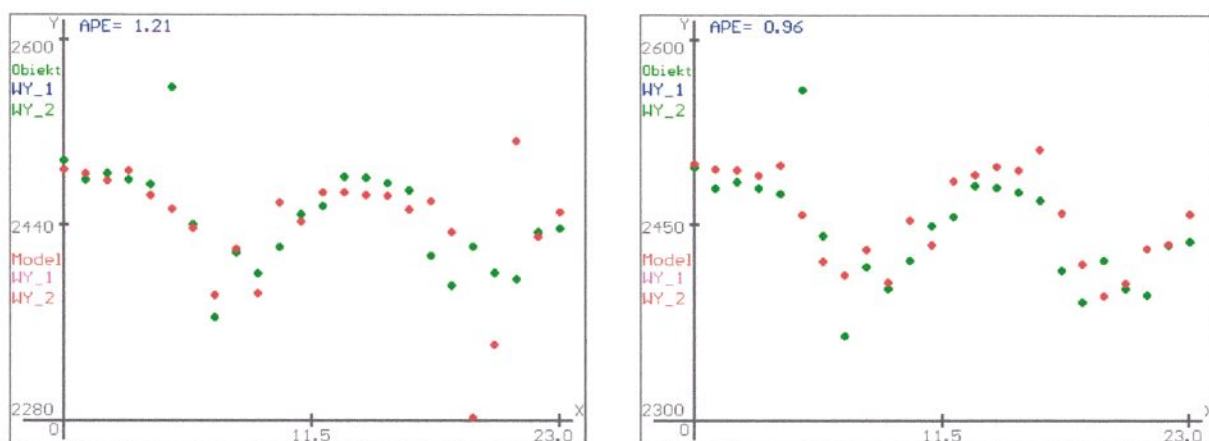


*Rys. 4.11 Wyjście obiektu i modelu złożonego
(WY_1 – temperatura T266)*

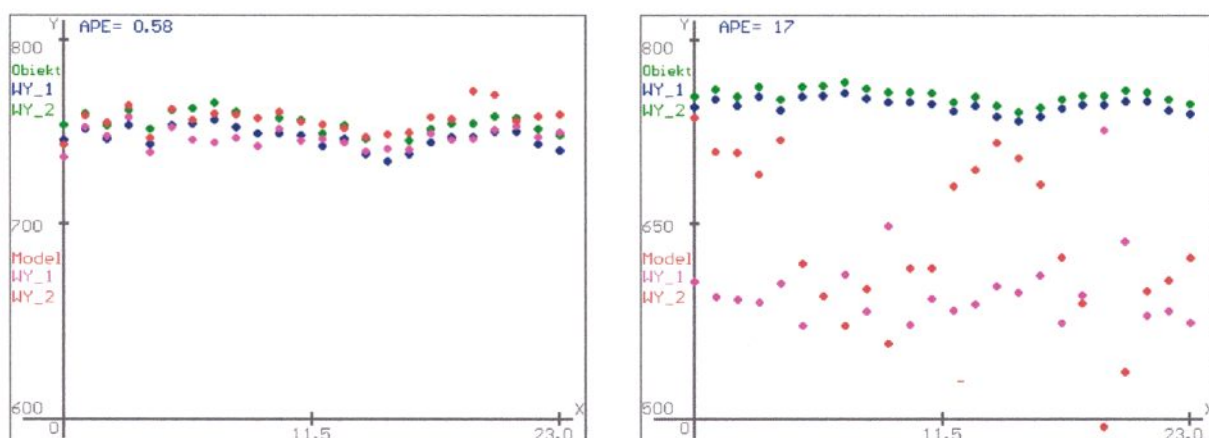
Model globalnie optymalny

Model globalnie optymalny –
szczególny przypadek

Dane testujące



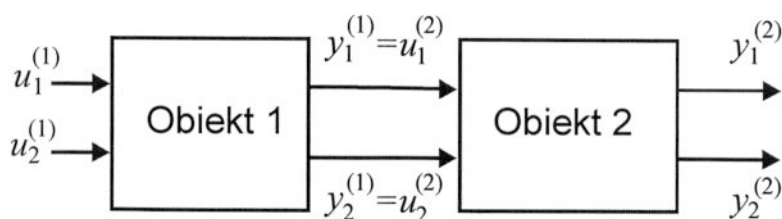
*Rys. 4.12 Wyjście pierwszego elementu obiektu i modelu
(WY_2 – przepływ F03)*



*Rys. 4.13 Wyjścia pierwszego elementu obiektu i modelu
(WY_1 – temperatura T251-1, WY_2 – temperatura T251-2)*

4.3.2 Modelowanie procesu – wariant „II”

Rysunek 4.14 przedstawia uproszczony model instalacji wytwarzającej azotyn amonu, składający się z dwóch elementów. Na pierwszy element składa się mieszalnik amoniaku z powietrzem oraz utlenianie tej mieszanki w kotle spalania, wyjściem tego obiektu jest temperatura na siatkach. Drugim elementem jest kocioł utylizator. Wyjściem tego elementu jest temperatura roztworu w kotle (T265) oraz poziom roztworu w kotle (L238).



Rys 4.14. Wariant „II”. Obiekt złożony jako połączenie obiektu pierwszego oraz drugiego

Dla obiektu złożonego jak na rysunku 4.14 przyjęto sieć o strukturze jak w tabeli 4.19. Początkowe wagi sieci wygenerowano losowo. Do uczenia wykorzystano algorytm zmodyfikowany (Rprop). Dane wejściowe i wyjściowe skalowano wg. (4.3), uczenie przerwano po 200 cyklach optymalizacji. Uczenie dla wszystkich metod zaczynało się od jednakowych wag początkowych.

Początkowe wartości parametrów do optymalizacji modelu globalnego z ograniczeniami na modele lokalne – wewnętrzna funkcja kary pochodzą z modelu lokalnie optymalnego po 200 cyklach optymalizacji. Dodatkowa liczba cykli optymalizacji dla modeli zawartych w tabeli 4.24 wyniosła 100.

Wyniki uczenia i testowania znajdują się w tabelach 2.20 – 2.23 oraz 4.24.

Tab.4.19. Struktura sieci neuronowej dla modelowania obiektu –wariant „II”

Nazwa modelu	Skalowanie	Struktura
<i>E</i>	Min-Max	2-5T-3T-2L-5T-3T-2L

Współczynniki we wzorze syntetycznym (2.17) przyjęto następujące: $\beta_2 = 0.5$, $\beta_1 = 0.5$ oraz $\alpha_0 = 0.5$, $\alpha_1 = 0.25$, $\alpha_2 = 0.25$ we wzorze (2.36).

Tab.4.20 Wskaźniki jakości po 200 cyklach optymalizacji

Lp	Uczenie	Wskaźnik faktycznie optymalizowany	Q	Q_u	Q_2	Q_I
	Model <i>E</i>					
1	Lokalnie optymalny		690	122	34	570
2	Globalnie optymalny	775	775	89.5	132	686
3	Globalnie optymalny szczególny przypadek	39	315000	39	410	314000
4	Globalnie optymalny szczególny przypadek - syntetyczny wskaźnik jakości	1060	948	102	111	846
5	Globalnie optymalny szczególny przypadek - ograniczenia na modele lokalne – zewnętrzna funkcja kary $\lambda_1 = 900$ $\lambda_2 = 166$	104	1010	104	113	900

Tab.4.21 Wskaźniki jakości po 200 cyklach optymalizacji

Lp	Test Model <i>E</i>	Wskaźnik faktycznie optymalizowany	Q	Q_u	Q_2	Q_I
1	Lokalnie optymalny		3150	103	64	3050
2	Globalnie optymalny	3320	3320	78	64	3240
3	Globalnie optymalny szczególny przypadek	138	309000	138	509	309000
4	Globalnie optymalny szczególny przypadek - syntetyczny wskaźnik jakości	2500	2430	82	66	2350
5	Globalnie optymalny szczególny przypadek - ograniczenia na modele lokalne – zewnętrzna funkcja. kary $\lambda_1=900$ $\lambda_2=166$	108	4280	108	75	4170

Tab.4.22 Błędy względne po 200 cyklach optymalizacji

Lp	Uczenie Model <i>E</i>	Q	Q_u	Q_2	Q_I
		[%]	[%]	[%]	[%]
1	Lokalnie optymalny	1.00	0.63	0.43	0.36
2	Globalnie optymalny	1.0	0.60	0.70	0.40
3	Globalnie optymalny szczególny przypadek	9.30	0.48	1.48	8.85
4	Globalnie optymalny szczególny przypadek -syntetyczny wskaźnik jakości	1.07	0.63	0.68	0.44
5	Globalnie optymalny szczególny przypadek - ograniczenia na modele lokalne – zewnętrzna funkcja kary	1.13	0.67	0.71	0.46

Tab.4.23 Błędy względne po 200 cyklach optymalizacji

Lp	Test Model <i>E</i>	Q	Q_u	Q_2	Q_I
		[%]	[%]	[%]	[%]
1	Lokalnie optymalny	1.59	0.76	0.65	0.83
2	Globalnie optymalny	1.47	0.65	0.63	0.82
3	Globalnie optymalny szczególny przypadek	9.30	0.80	1.60	8.50
4	Globalnie optymalny szczególny przypadek -syntetyczny wskaźnik jakości	1.43	0.68	0.65	0.74
5	Globalnie optymalny szczególny przypadek - ograniczenia na modele lokalne – zewnętrzna funkcja kary	1.64	0.77	0.67	0.87

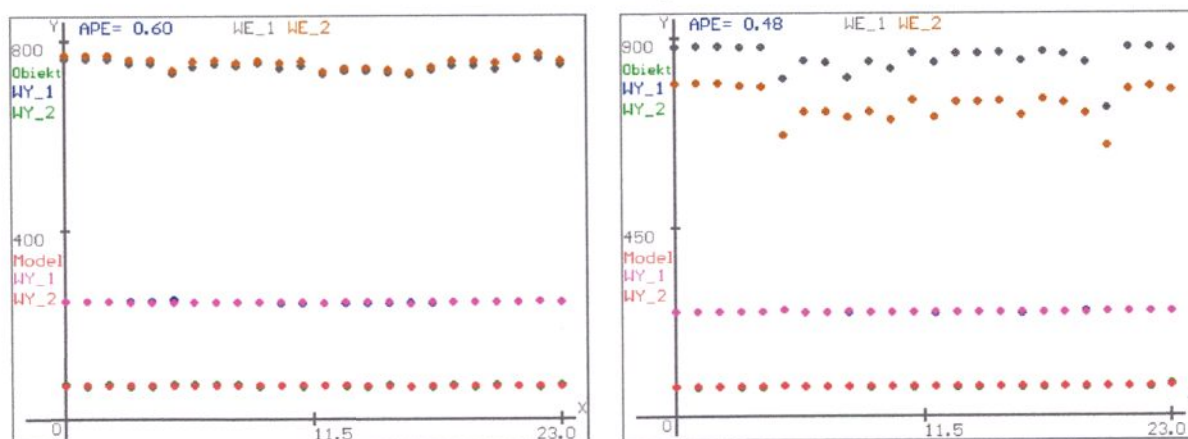
Tab.4.24 Wskaźniki jakości po 300 cyklach optymalizacji

L p	Uczenie Model A	Wskaźnik faktycznie optymalizowany	Q	Q_u	Q_2	Q_1
1	Lokalnie optymalny		720	208	33	511
2	Globalnie optymalny szczególny przypadek - ograniczenia na modele lokalne – wewnętrzna funkcja kary $\lambda_1=900$ $\lambda_2=166$	102	654	102	63	542

Przebiegi sygnałów uczących i testujących dla procesu wariant „II” oraz w jego modelu

Na wykresach od 4.15 do 4.24 przedstawiono w postaci graficznej przebiegi w modelu sieciowym typ E , po 200 cyklach optymalizacyjnych. W kolumnie lewej zostały przedstawione wyniki uzyskane w modelu globalnie optymalnym. W prawej kolumnie wyniki z modelu globalnie optymalnego – szczególny przypadek. Przedstawiono dane wejściowo wyjściowe obiektu i procesu w skali 1:1. Następnie dla przedstawiono przebiegi wyjściowe poszczególnych elementów w skali zbliżonej do ich wartości nominalnych.

Model globalnie optymalny

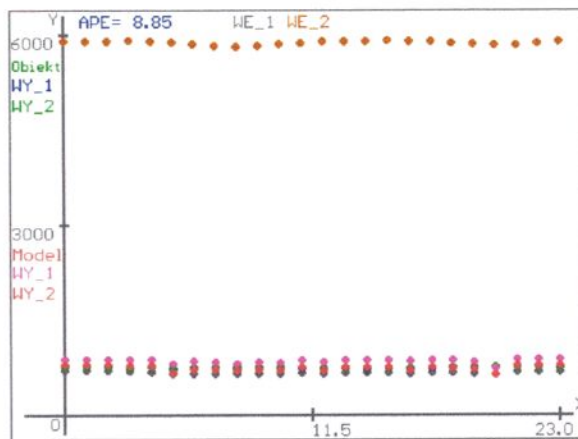
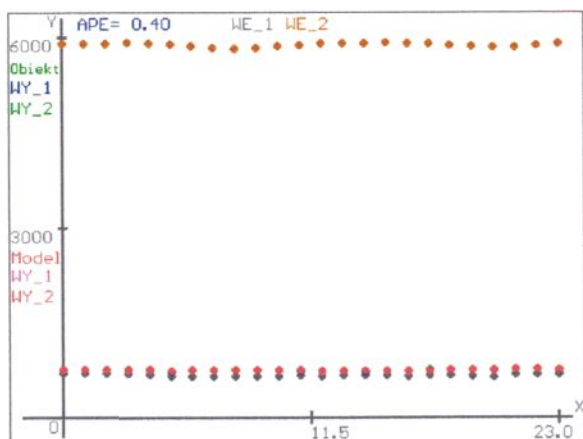
Model globalnie optymalny –
szczególny przypadek**Dane uczące**

Rys.4.15 Wejścia i wyjścia obiektu i modelu złożonego
(WY_1 – temperatura T265, WY_2 – poziom L238)

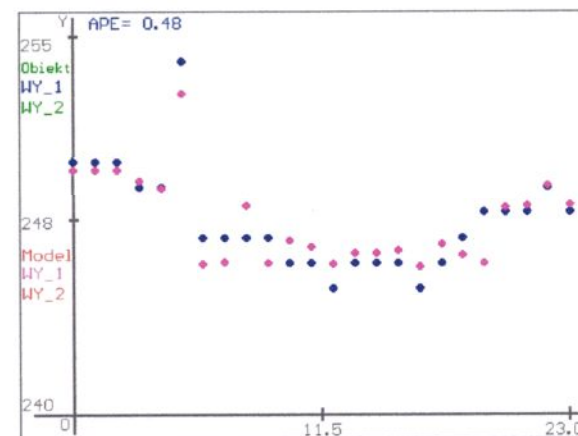
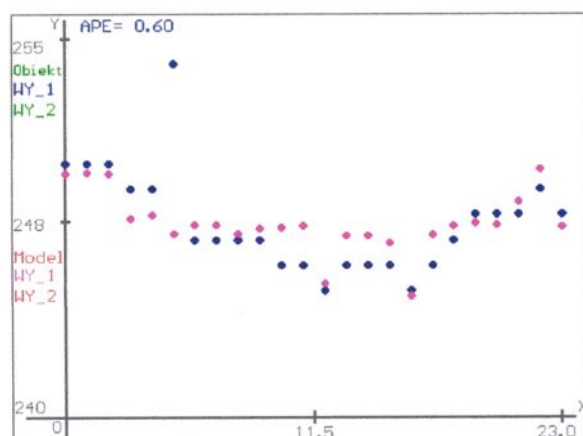
Model globalnie optymalny

Model globalnie optymalny –
szczególny przypadek

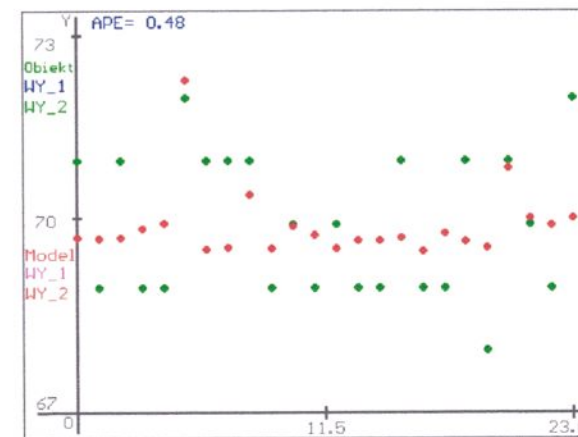
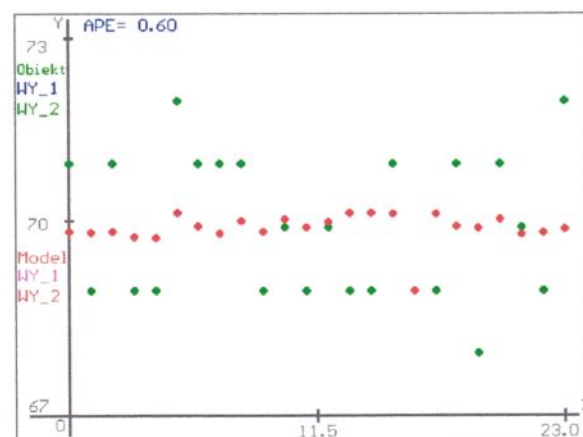
Dane uczące



*Rys. 4.16 Wejścia i wyjścia pierwszego elementu obiektu i modelu
(WY_1 – temperatura T251-1, WY_2 – temperatura T251-2)*



*Rys. 4.17 Wyjście pierwszego elementu obiektu i modelu
(WY_1 – temperatura)*

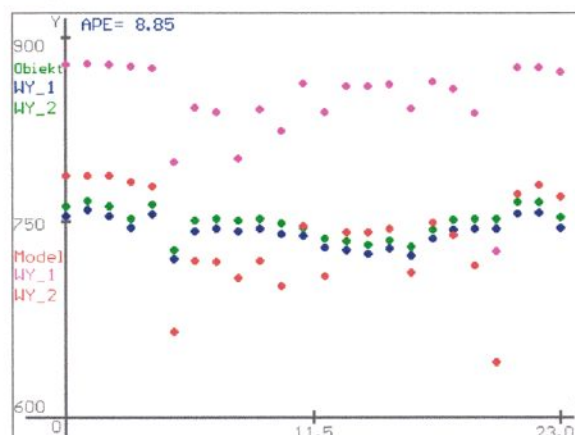
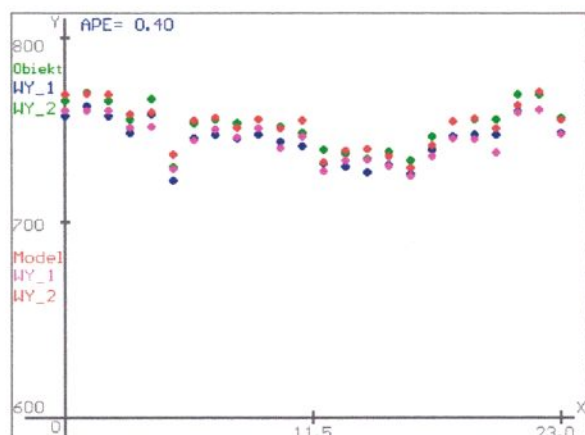


*Rys. 4.18 Wyjście pierwszego elementu obiektu i modelu
(WY_2 – poziom L238)*

Model globalnie optymalny

Model globalnie optymalny –
szczególny przypadek

Dane uczące

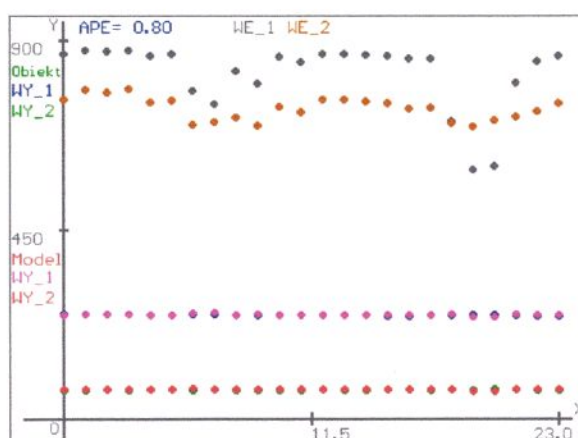
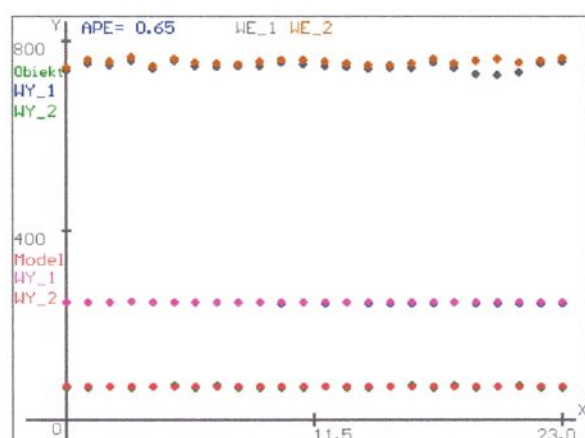


*Rys.4.19 Wyjścia pierwszego elementu obiektu i modelu
(WY_1 – temperatura T251-1, WY_2 – temperatura T251-2)*

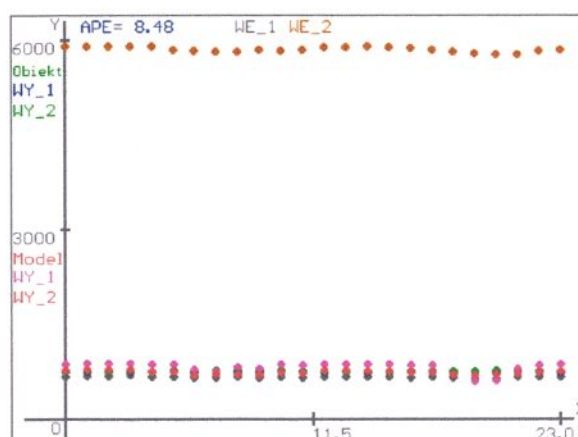
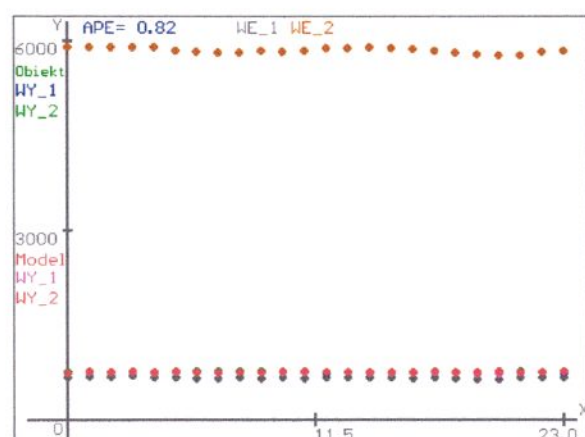
Model globalnie optymalny

Model globalnie optymalny –
szczególny przypadek

Dane testujące



*Rys.4.20 Wejścia i wyjścia obiektu i modelu złożonego
(WY_1 – temperatura T265, WY_2 – poziom L238)*

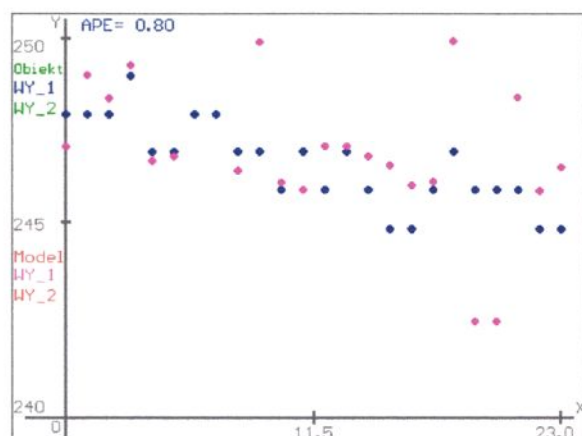
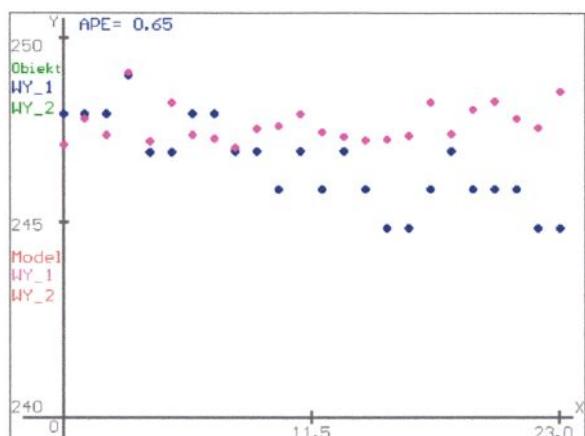


*Rys.4.21 Wejścia i wyjścia pierwszego elementu obiektu i modelu
(WY_1 – temperatura T251-1, WY_2 – temperatura T251-2)*

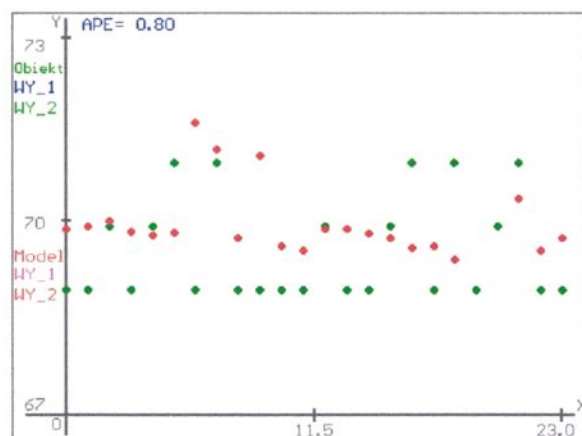
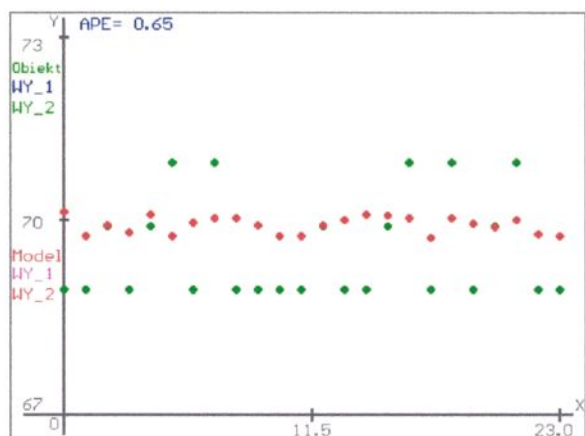
Model globalnie optymalny

Model globalnie optymalny –
szczególny przypadek

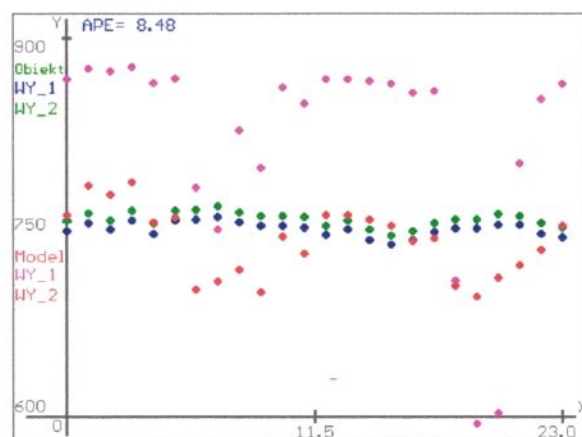
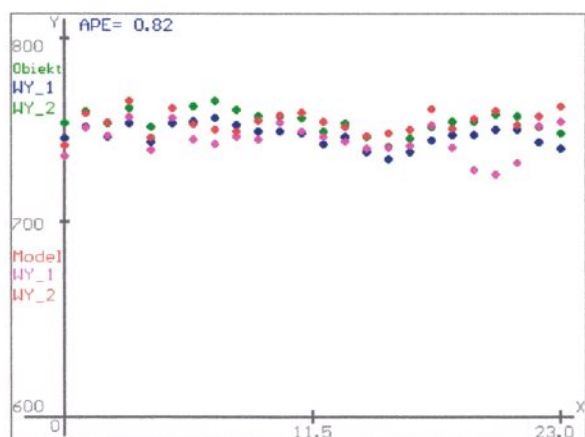
Dane testujące



Rys. 4.22 Wyjście pierwszego elementu obiektu i modelu
(WY_1 – temperatura)



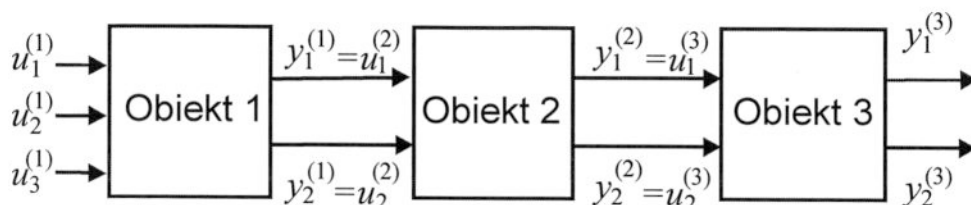
Rys. 4.23 Wyjście pierwszego elementu obiektu i modelu
(WY_2 – poziom L238)



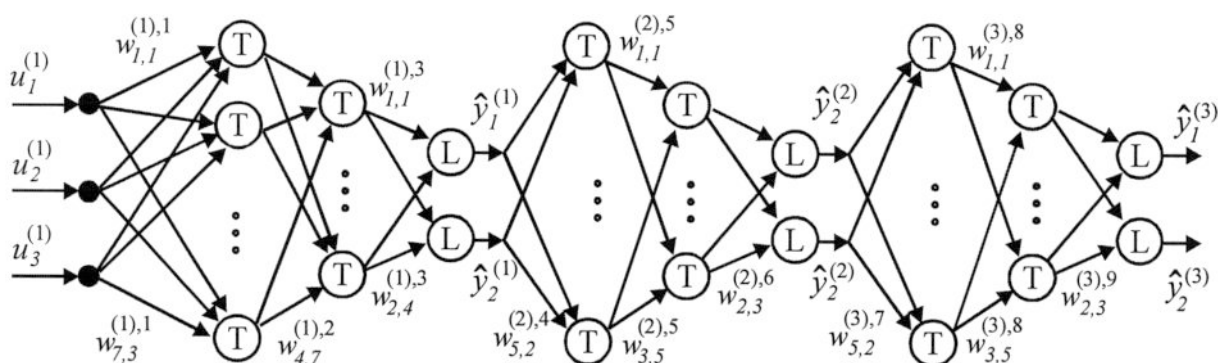
Rys. 4.24 Wyjścia pierwszego elementu obiektu i modelu
(WY_1 – temperatura T251-1, WY_2 – temperatura T251-2)

4.3.3 Modelowanie procesu – wariant „III”

Rysunek 4.25 przedstawia model dostępnego fragmentu instalacji produkcji azotynu amonu, składający się z trzech elementów. Model ten odpowiada schematowi z rys.4.2, który został szczegółowo opisany w sekcji 4.2.



Rys.4.25 Wariant „III”. Obiekt złożony jako połączenie obiektu pierwszego, drugiego i trzeciego



Rys.4.26 Struktura sieci neuronowej dla modelowania obiektu złożonego wariant „III” (model F)

Dla obiektu złożonego jak na rysunku 4.25 przyjęto sieć o strukturze wg. w tabeli 4.25, pokazanej na rysunku 4.26. Wagi sieci generowano losowo. Do uczenia wykorzystano również algorytm zmodyfikowany. Dane wejściowe i wyjściowe skalowano wg. (4.3), uczenie przerwano po 100 prezentacjach danych uczących. Uczenie sieci dla wszystkich metod zaczynało się od jednakowych wag początkowych.

Początkowe wartości parametrów do optymalizacji modelu globalnie optymalnego z ograniczeniami na modele lokalne – wewnętrzna funkcja kary pochodzą z modelu lokalnie optymalnego po 100 cyklach optymalizacji. Dodatkowa liczba cykli optymalizacji dla modeli zawartych w tabeli 4.30 wyniosła 50. Łączna suma cykli optymalizacji wynosi 150.

Wyniki uczenia i testowania zawarto w tabelach 4.26 – 4.29 obliczając dla poszczególnych metod wskaźniki jakości oraz błąd procentowy APE.

Tab.4.25 Struktura sieci neuronowych dla modelowania procesu – wariant „III”

Nazwa modelu	Skalowanie	Struktura
F	Min-Max	3-7T-4T-2L-5T-3T-2L-5T-3T-2L

Współczynniki we wzorze syntetycznym (2.17) przyjęto następujące: $\beta_3 = 0.33$, $\beta_2 = 0.33$, $\beta_1 = 0.33$, oraz $\alpha_0 = 0.25$, $\alpha_1 = 0.25$, $\alpha_2 = 0.25$, $\alpha_3 = 0.25$ we wzorze (2.36).

Tab.4.26 Wskaźniki jakości po 100 cyklach optymalizacji

Lp	Uczenie Model F	Wskaźnik faktycznie optymalizowany	Q	Q_u	Q_3	Q_2	Q_1
1	Lokalnie optymalny		35400	34300	202000	50	884
2	Globalnie optymalny	29800	29800	28700	53300	120	959
3	Globalnie optymalny szczególny przypadek	9070	1590000	9070	110000	22700	1570000
4	Globalnie optymalny szczególny przypadek - syntetyczny wskaźnik jakości	81000	32200	31100	48900	91	986
5	Globalnie optymalny szczególny przypadek - ograniczenia na modele lokalne – zewnętrzna funkcja kary $\lambda_1 = 2730$ $\lambda_2 = 100$ $\lambda_3 = 49000$	31000	34100	31000	39000	100	2720

Tab.4.27 Wskaźniki jakości po 100 cyklach optymalizacji

Lp	Test Model F	Wskaźnik faktycznie optymalizowany	Q	Q_u	Q_3	Q_2	Q_1
1	Lokalnie optymalny		57300	55500	58500	53	1700
2	Globalnie optymalny	56200	56200	54400	69500	61	1670
3	Globalnie optymalny szczególny przypadek	47200	1730000	47200	65000	22800	167000
4	Globalnie optymalny szczególny przypadek - syntetyczny wskaźnik jakości	128000	64700	62800	63800	48	1690
5	Globalnie optymalny szczególny przypadek - ograniczenia na modele lokalne – zewnętrzna funkcja kary $\lambda_1 = 2730$ $\lambda_2 = 100$ $\lambda_3 = 49000$	60700	64400	60700	58400	53	3290

Tab.4.28 Błędy względne po 100 cyklach optymalizacji.

Lp	Uczenie Model F	Q	Q_u	Q_3	Q_2	Q_1
		[%]	[%]	[%]	[%]	[%]
1	Lokalnie optymalny	2.27	1.25	0.90	0.54	0.48
2	Globalnie optymalny	2.27	1.07	1.51	0.69	0.48
3	Globalnie optymalny szczególny przypadek	31	0.61	2.40	10.6	22.1
4	Globalnie optymalny szczególny przypadek - syntetyczny wskaźnik jakości	2.43	1.12	1.48	0.64	0.48
5	Globalnie optymalny szczególny przypadek - ograniczenia na modele lokalne – zewnętrzna funkcja kary	2.97	1.12	1.32	0.65	0.78

Tab.4.29 Błędy względne po 100 cyklach optymalizacji

Lp	Test Model F	Q	Q_u	Q_3	Q_2	Q_1
		[%]	[%]	[%]	[%]	[%]
1	Lokalnie optymalny	2.82	1.55	1.61	0.62	0.65
2	Globalnie optymalny	2.83	1.49	1.66	0.62	0.63
3	Globalnie optymalny szczególny przypadek	32	1.39	1.71	10.1	23.2
4	Globalnie optymalny szczególny przypadek - syntetyczny wskaźnik jakości	3.16	1.61	1.55	0.55	0.66
5	Globalnie optymalny szczególny przypadek - ograniczenia na modele lokalne – zewnętrzna funkcja kary	3.45	1.56	1.54	0.57	0.89

Tab.4.30 Wskaźniki jakości po 150 cyklach optymalizacji

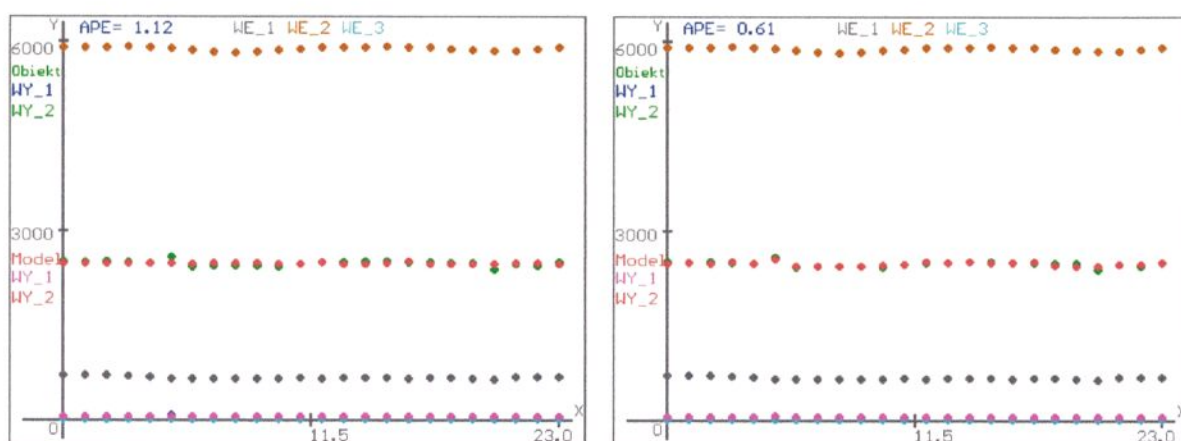
Lp	Uczenie Model F	Wskaźnik faktycznie optymalizo wany	Q	Q_u	Q_3	Q_2	Q_1
1	Lokalnie optymalny		32500	31700	19200	44	728
2	Globalnie optymalny szczególny przypadek - ograniczenia na modele lokalne – wewnętrzna funkcja kary $\lambda_1=2730$ $\lambda_2=100$ $\lambda_3=49000$	31400	32400	31400	20200	57	933

Przebiegi sygnałów uczących i testujących dla procesu wariant „III” oraz w jego modelu

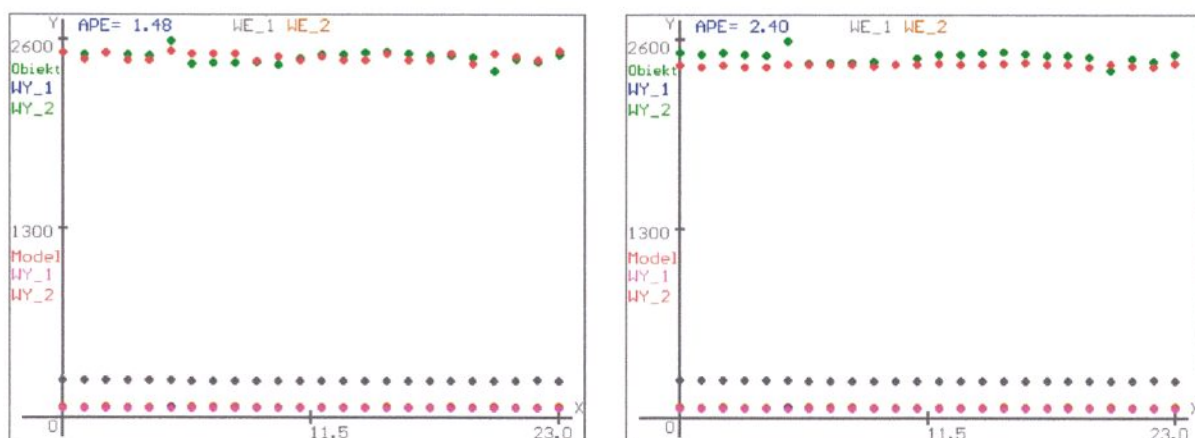
Na rysunkach od 4.27 do 4.48 przedstawiono w postaci graficznej przebiegi w modelu F , po 100 cyklach optymalizacyjnych. W kolumnie lewej zostały przedstawione wyniki uzyskane w modelu globalnie optymalnym z uwzględnieniem jakości modeli lokalnych – syntetyczny wskaźnik jakości. W prawej kolumnie wyniki z modelu globalnie optymalnego – szczególnie przypadek. Przedstawiono dane wejściowo wyjściowe obiektu i procesu w skali 1:1. Przedstawiono także przebiegi na wyjściu poszczególnych elementów obiektu i ich modeli lokalnych.

Model globalnie optymalny z
z uwzględnieniem jakości
lokalnych - wzór syntetyczny

Model globalnie optymalny
- szczególnie przypadek

Dane uczące

Rys.4.27 Wyjścia globalne modelu i obiektu złożonego
(WY_1 – przepływ F203, WY_2 – temperatura T266)

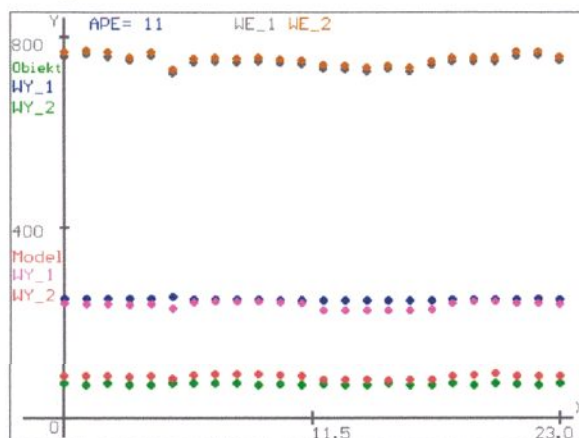
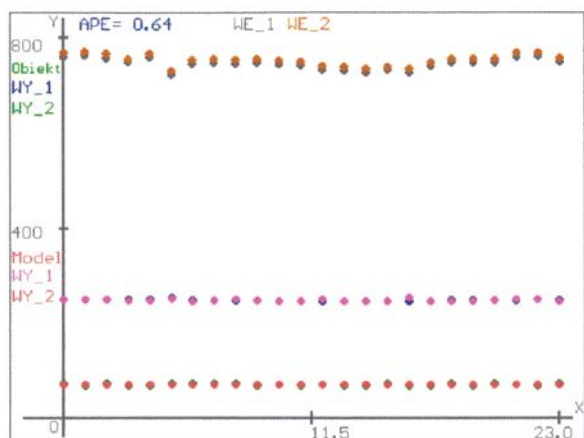


Rys.4.28 Wyjście lokalne trzeciego elementu modelu i obiektu
(WY_1 – przepływ F203, WY_2 – temperatura T266)

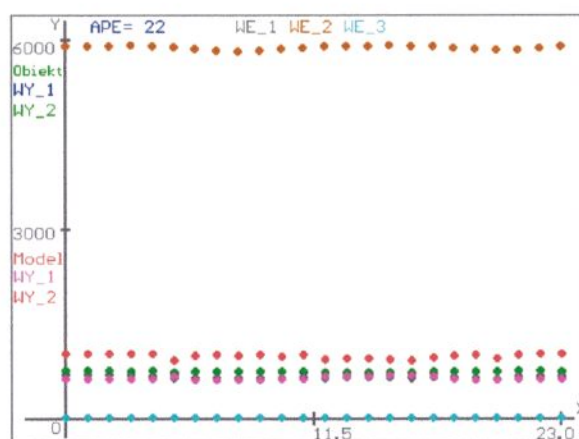
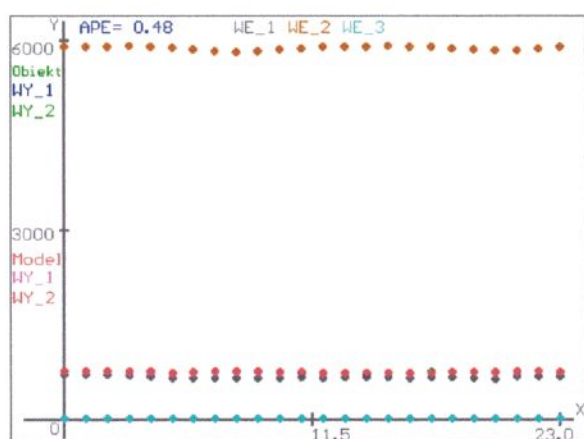
Model globalnie optymalny z
z uwzględnieniem jakości
lokalnych -wzór syntetyczny

Model globalnie optymalny
- szczególny przypadek

Dane uczące

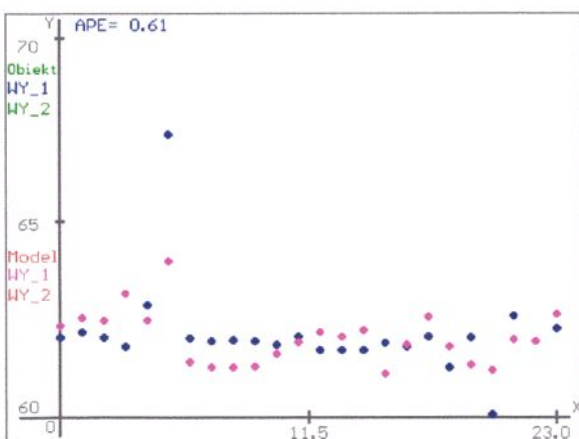
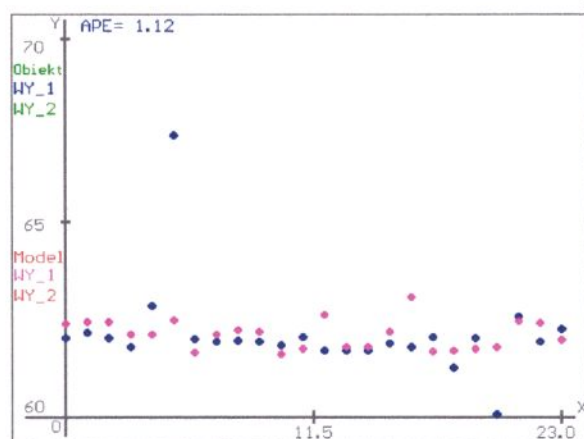


Rys.4.29 Wyjście lokalne drugiego elementu modelu i obiektu
(WY_1 – temperatura T265, WY_2 – poziom L238)



Rys.4.30 Wyjście lokalne drugiego elementu modelu i obiektu
(WY_1 – temperatura T251-1, WY_2 – temperatura T251-2)

Dane uczące – skalowane

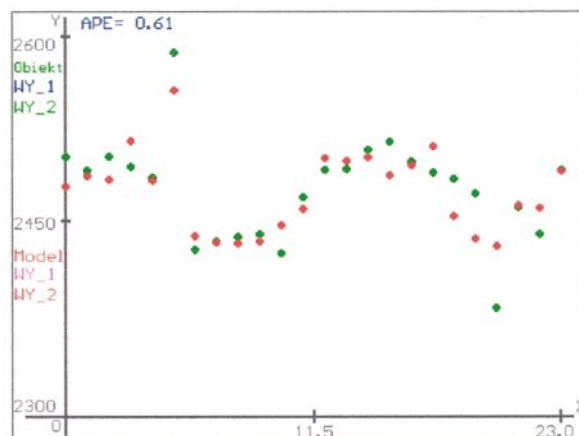
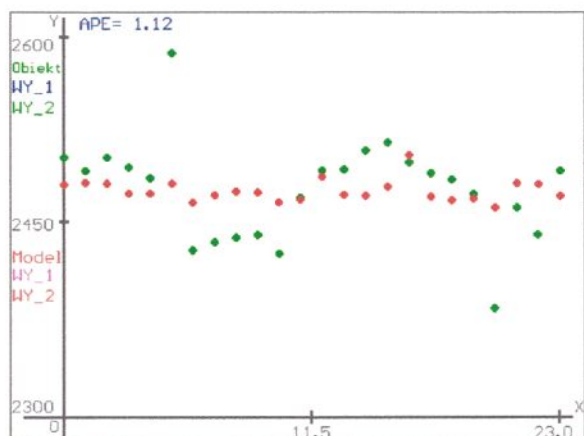


Rys.4.31 Wyjście globalne modelu i obiektu złożonego (WY_1— temperatura T266)

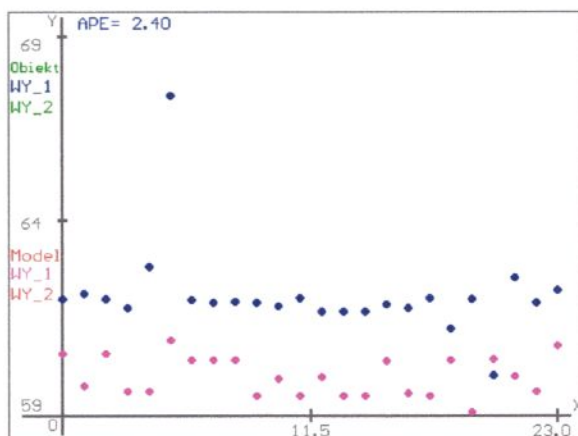
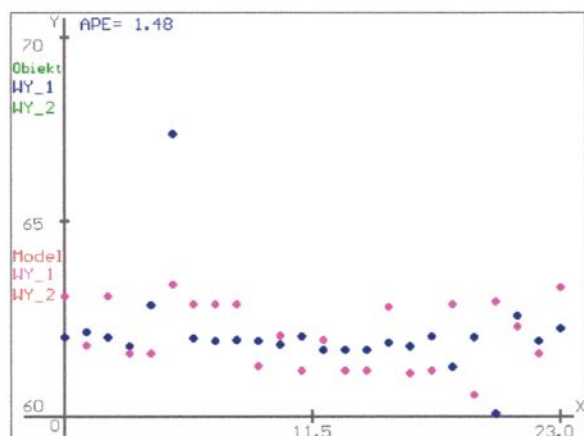
Model globalnie optymalny z
z uwzględnieniem jakości modeli
lokalnych - wzór syntetyczny

Model globalnie optymalny
szczególny przypadek

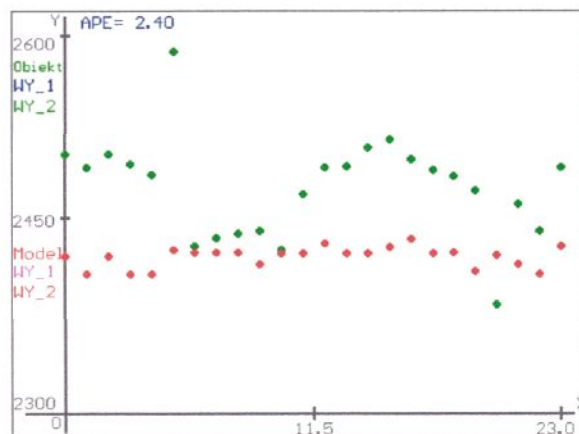
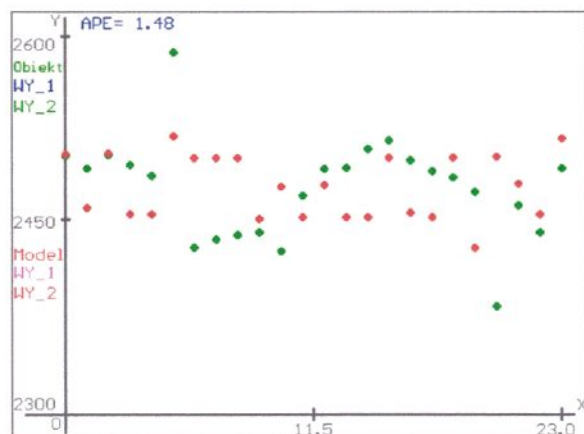
Dane uczące – skalowane



Rys. 4.32 Wyjście globalne modelu i obiektu złożonego (WY_2 - przypływ F203)



Rys. 4.33 Wyjście lokalne trzeciego elementu modelu i obiektu (WY_1 - temperatura T266)

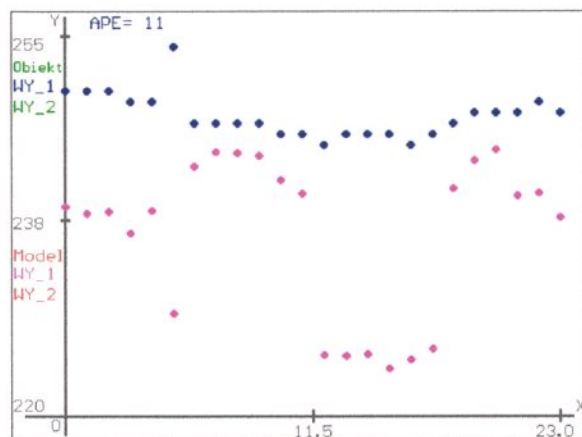
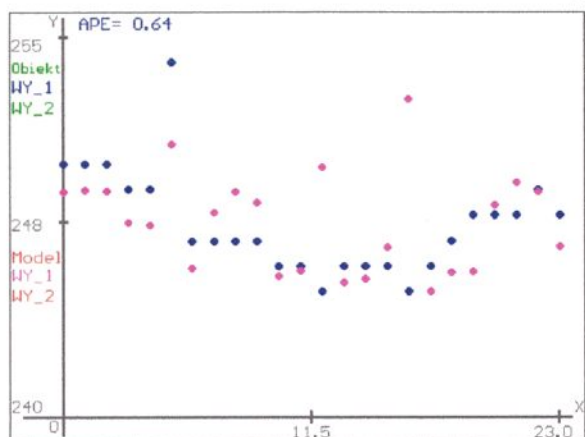


Rys. 4.34 Wyjście lokalne trzeciego elementu modelu i obiektu (WY_2 - przypływ F203)

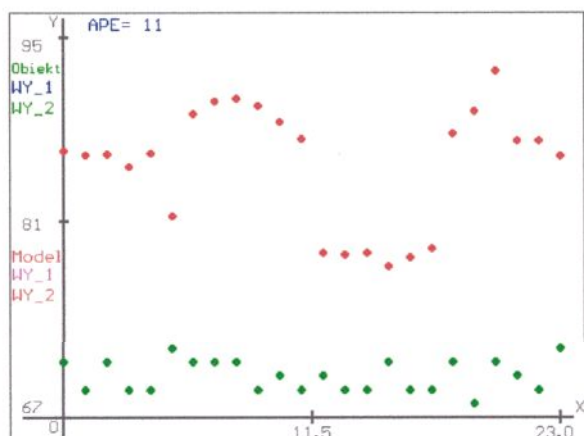
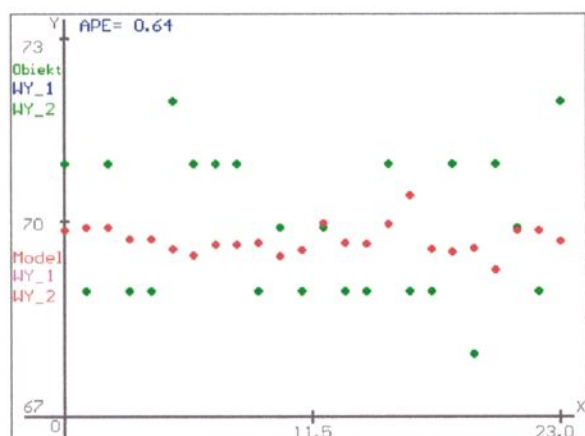
Model globalnie optymalny z
z uwzględnieniem jakości modeli
lokalnych - wzór syntetyczny

Model globalnie optymalny
szczególny przypadek

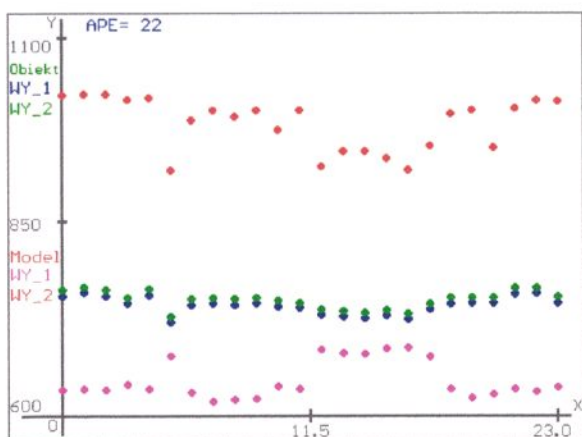
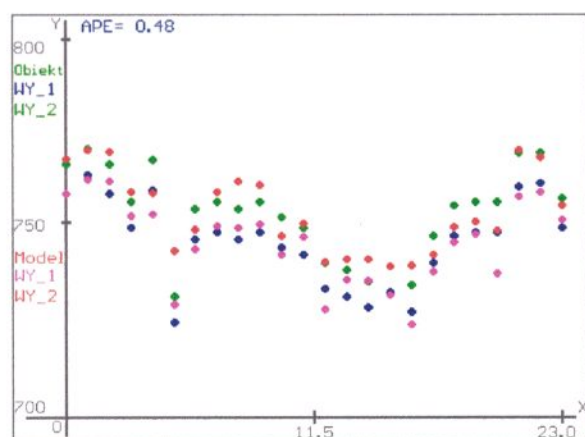
Dane uczące – skalowane



Rys.4.35 Wyjście lokalne drugiego elementu modelu i obiektu (WY_1 - temperatura T265)



Rys.4.36 Wyjście lokalne drugiego elementu modelu i obiektu (WY_2 - poziom L238)

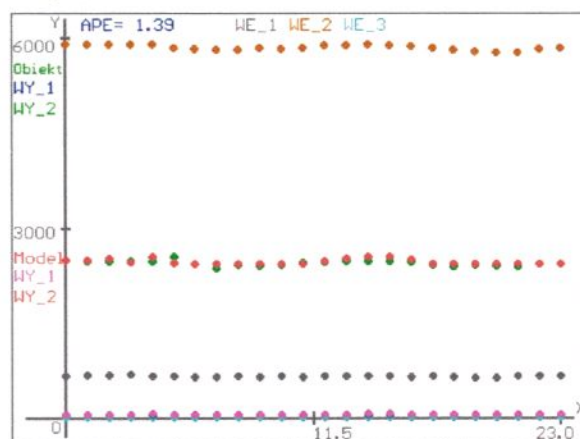
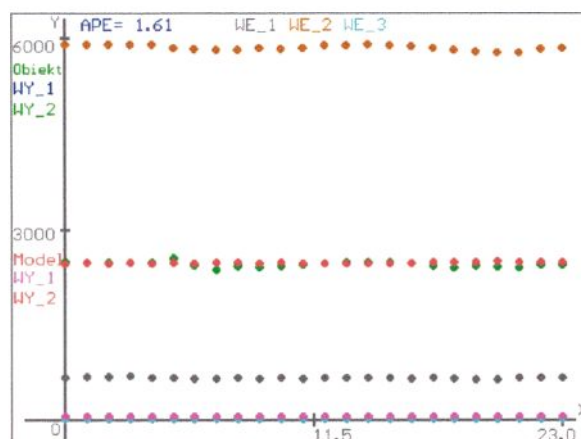


Rys.4.37 Wyjścia lokalne pierwszego elementu modelu i obiektu
(WY_1 i WY_2 - temperatury T251-1 i T251-2)

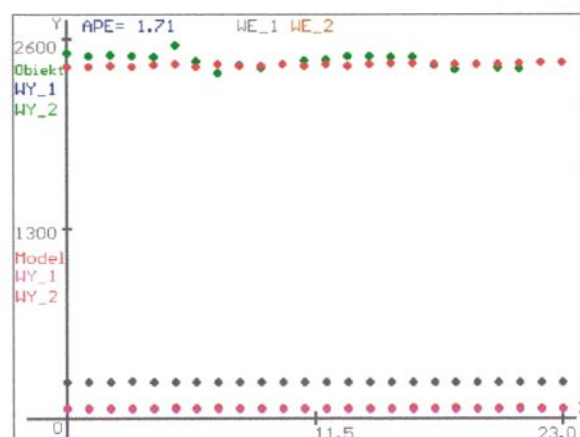
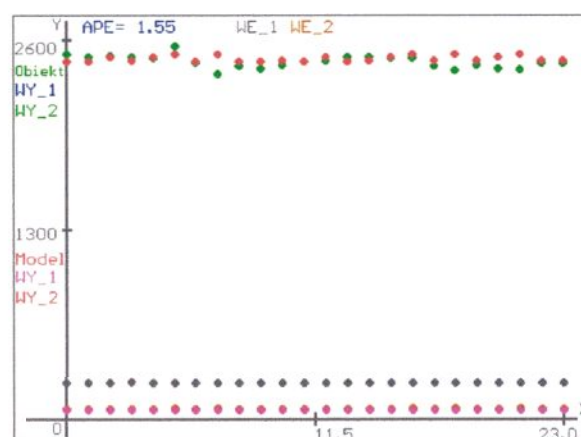
Model globalnie optymalny z
z uwzględnieniem jakości modeli
lokalnych - wzór syntetyczny

Model globalnie optymalny
szczególny przypadek

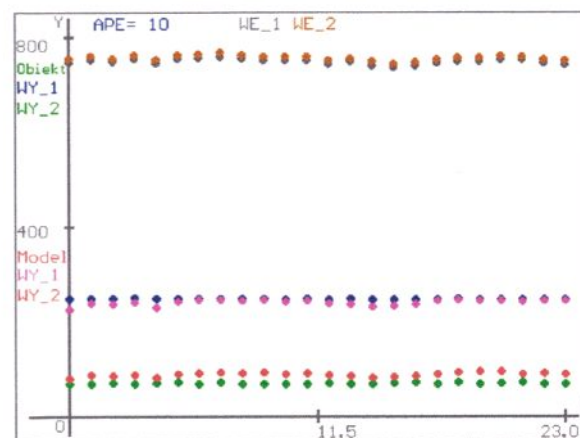
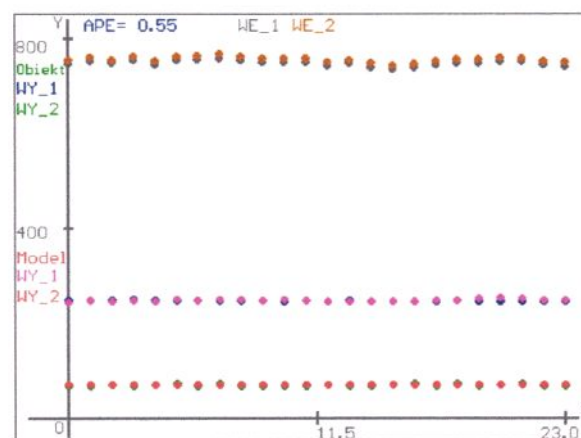
Dane testujące



*Rys.4.38 Wyjścia globalne modelu i obiektu złożonego
(WY_1 – przepływ F203, WY_2 – temperatura T266)*



*Rys.4.39 Wyjście lokalne trzeciego elementu modelu i obiektu
(WY_1 – przepływ F203, WY_2 – temperatura T266)*

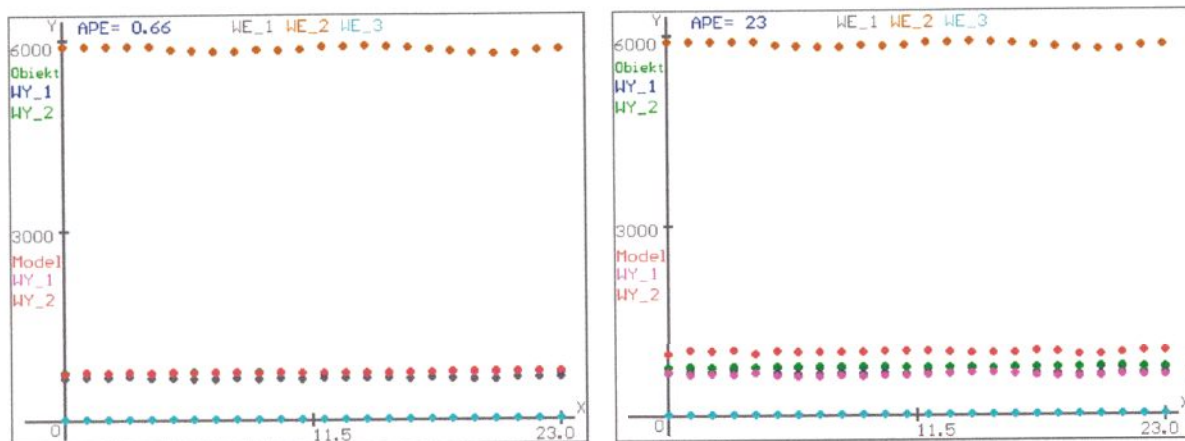


*Rys.4.40 Wyjście lokalne drugiego elementu modelu i obiektu
(WY_1 – temperatura T265, WY_2 – poziom L238)*

Model globalnie optymalny z
z uwzględnieniem jakości modeli
lokalnych - wzór syntetyczny

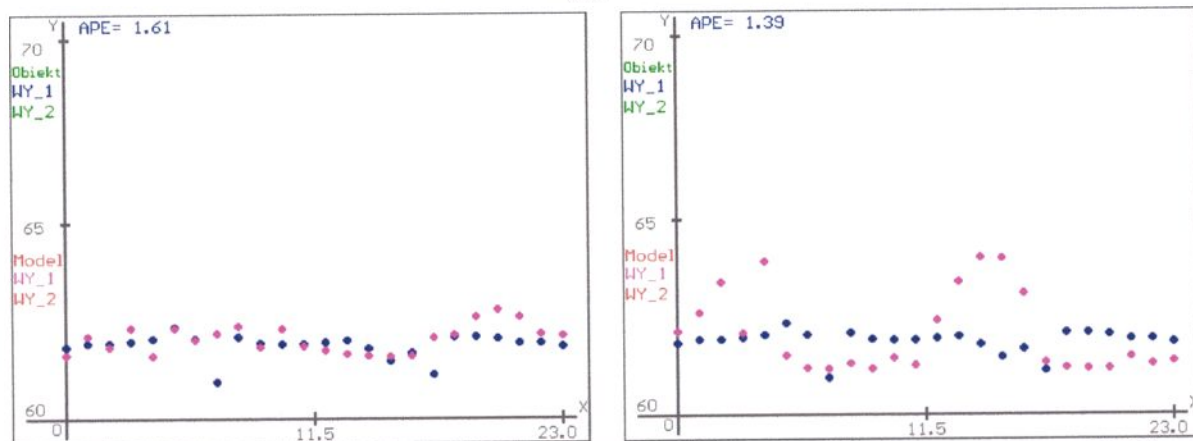
Model globalnie optymalny
szczególny przypadek

Dane testujące

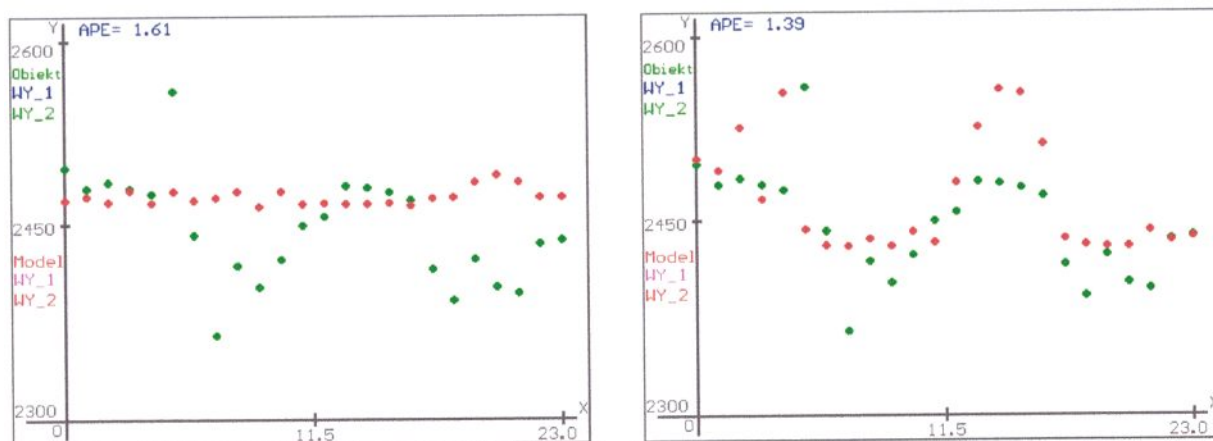


Rys.4.41 Wyjście lokalne drugiego elementu modelu i obiektu
(WY_1 – temperatura T251-1, WY_2 – temperatura T251-2)

Dane testujące – skalowane



Rys.4.42 Wyjście globalne modelu i obiektu złożonego (WY_1 – temperatura T266)

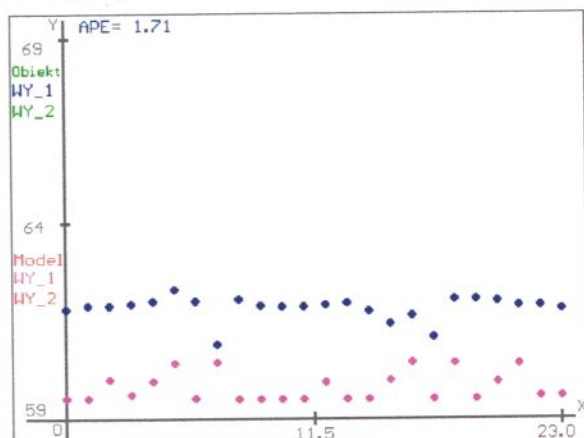
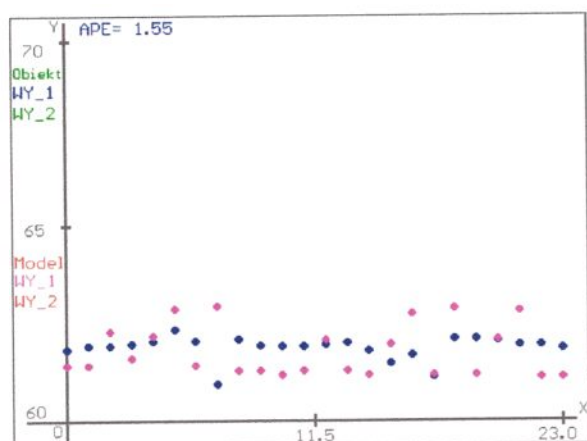


Rys.4.43 Wyjście globalne modelu i obiektu złożonego (WY_2 – przepływ F203)

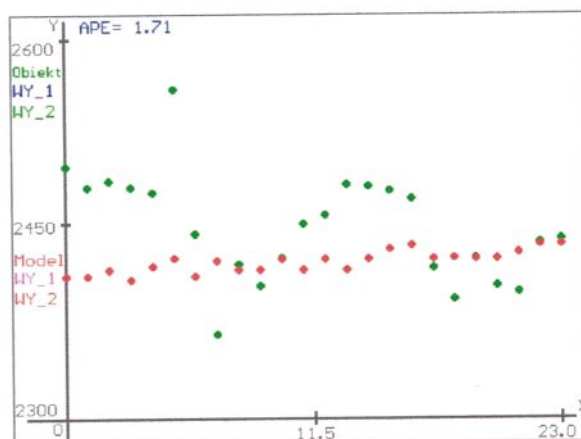
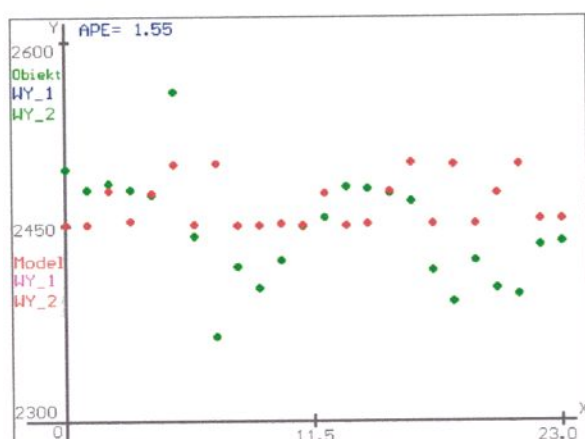
Model globalnie optymalny z
z uwzględnieniem jakości modeli
lokalnych - wzór syntetyczny

Model globalnie optymalny
szczególny przypadek

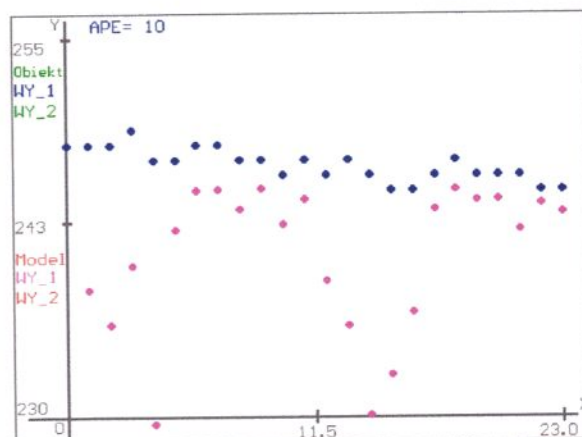
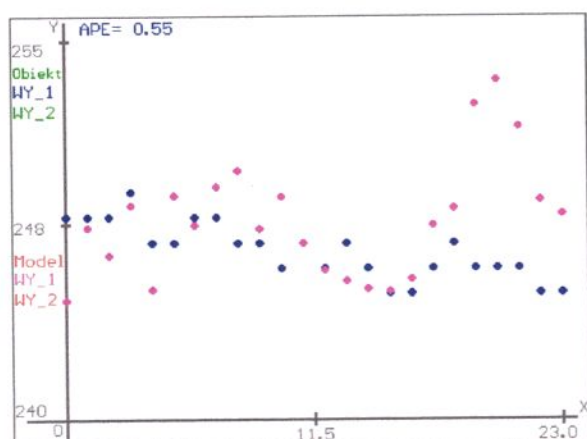
Dane testujące - skalowane



Rys. 4.44 Wyjście lokalne trzeciego elementu modelu i obiektu (WY_1 - temperatura T266)



Rys. 4.45 Wyjście lokalne trzeciego elementu modelu i obiektu (WY_2 - przepływ F203)

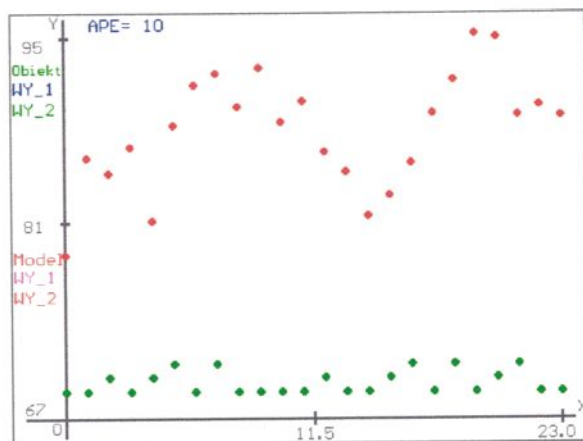
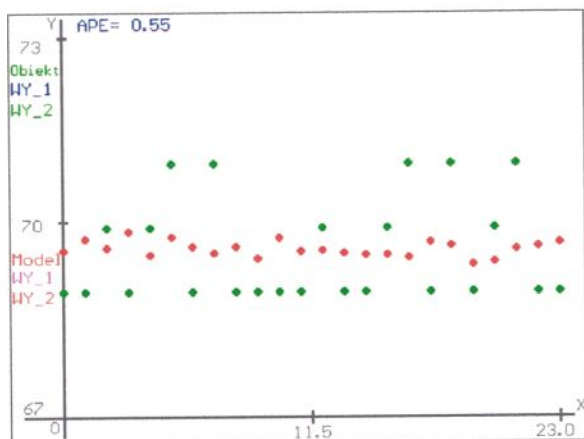


Rys. 4.46 Wyjście lokalne drugiego elementu modelu i obiektu (WY_1 - temperatura T265)

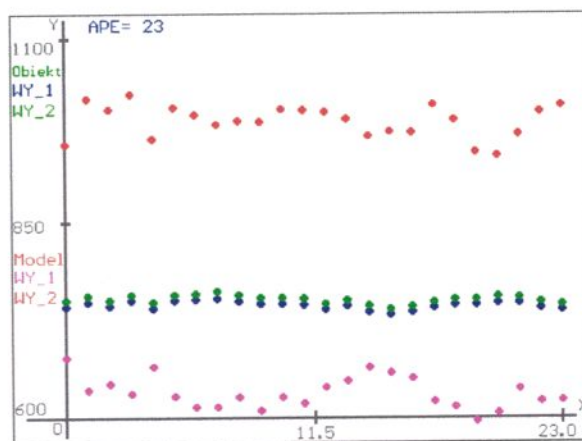
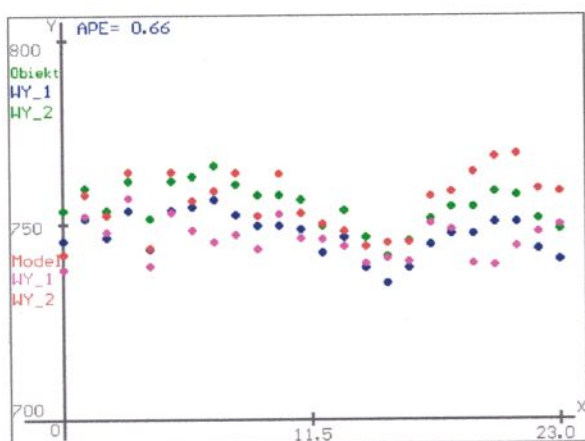
Model globalnie optymalny z
z uwzględnieniem jakości modeli
lokalnych - wzór syntetyczny

Model globalnie optymalny
szczególny przypadek

Dane testujące – skalowane



Rys. 4.47 Wyjście lokalne drugiego elementu modelu i obiektu (WY_2 - poziom L238)



Rys. 4.48 Wyjścia lokalne pierwszego elementu modelu i obiektu (WY_1 i WY_2 - temperatury T251-1 i T251-2)

4.4 Omówienie wyników

Wariant „II”

Tabele 4.20 i 4.21 przedstawiają wskaźniki jakości a 4.22 i 4.23 błędy względne APE modelowania obiektu złożonego - wariant „II” przedstawionego na rysunku 4.14.

Według kryterium Q dla danych uczących najmniejsze wskaźniki jakości uzyskały modele: lokalnie optymalny (LO oznaczenia wg p.3.3). globalnie optymalny (GO). Zdecydowanie największe wskaźniki posiada model globalnie optymalny – szczególny przypadek (GSP), ponieważ modele częściowe nie były optymalizowane. Dla danych testujących najmniejszą wartość wskaźnika jakości Q uzyskał model GWS.

Oceniając jakość modeli według błędów obliczonych tylko na wyjściu według kryterium Q_u to dla danych uczących najmniejszy błąd posiada model GSP. Drugi wynik uzyskał model GO. Największy wskaźnik Q_u uzyskał model LO.

Dla danych testujących najmniejszy Q_u posiada model globalnie optymalny oraz model GWS, a największy wskaźnik Q_u uzyskał model GZK. Najlepsze wskaźniki jakości modeli lokalnych Q_1 i Q_2 dla danych uczących uzyskał model lokalnie optymalny, a dla danych testujących model GWS.

Tabele z błędami APE dają możliwość oszacowania, o ile poszczególne modele różnią się od modelowanego obiektu. Dla danych uczących dla wskaźnika Q_u wahają się od 0.48 do 0.68 procent. Dla danych testujących błędy obliczone według kryterium Q_u wahają się od 0.64 do 1.04 procent.

Wariant „I”

Tabele od 4.3 do 4.18 przedstawiają błędy bezwzględne i względne APE modelowania obiektu złożonego - wariant „I” przedstawionego na rysunku 4.3.

Według kryterium Q dla danych uczących najmniejsze wartości błędu uzyskały modele globalny (2 razy najlepszy, dwa razy drugi) i GWS (2 razy najlepszy, raz drugi) dla czterech modeli sieciowych. Największe wartości wskaźnika Q uzyskały modele LO oraz zdecydowanie modele GSP. Dla danych testujących najmniejsze wskaźniki jakości Q posiadały modele GZK (2 razy najlepszy i raz drugi). Nieco większe wartości tego wskaźnika posiadają modele GWS.

Modelem globalnie optymalnym z ograniczeniami na modele lokalne – wewnętrzna funkcja kary (GWK) uzyskał mniejsze wskaźniki Q i Q_u dla danych uczących w porównaniu z modelem LO, natomiast model LO uzyskał mniejsze wartości wskaźników Q_1 i Q_2 niż GWK.

Oceniając jakość modeli według błędów obliczonych tylko na wyjściu według kryterium Q_u to dla danych uczących najmniejszy błąd posiadają modele globalnie optymalne – szczególny przypadek (4 razy najmniejszy). Drugi wynik uzyskały modele globalnie optymalne. Największe wartości wskaźnika Q_u uzyskano w modelach GZK i LO. Dla danych testujących najmniejsze wskaźniki jakości posiadają również modele globalnie optymalne – szczególny przypadek (4 razy najmniejszy). Nieco większe wskaźniki jakości uzyskały modele GO i GWS.

Dla danych uczących najmniejsze wartości wskaźników jakości modeli lokalnych Q_1 , Q_2 uzyskano w modelach lokalnie optymalnych. Dla danych testujących najmniejsze wartości wskaźnika Q_1 posiadają modele GWS, a wskaźnika Q_2 modele LO.

Porównując model lokalnie optymalny z modelem globalnie optymalnym z ograniczeniami na modele lokalne – wewnętrzną funkcję kary (GWK), to model GWK uzyskał mniejsze wskaźniki Q i Q_u w porównaniu z modelem LO, natomiast większe wartości wskaźników Q_1 i Q_2 .

Błędy APE dla danych uczących dla wskaźnika Q_u wahają się od 0.25 do 2.55 procenta, a dla danych testujących błędy wahają się od 0.86 do 2.88 procent.

Wariant „III”

Zarówno dla danych uczących jak i testujących najmniejszą wartość kryterium Q uzyskał model globalnie optymalny. Po nim, nieco większe wartości Q posiada model GWS dla danych uczących i LO dla danych testujących. Największe wartości wskaźnika Q dla danych uczących występują w modelu GSP i LO, dla testujących w GSP.

Oceniając jakość modeli według kryterium Q_u to dla danych uczących jak i testujących najmniejszy jego wartości posiada model GSP. Drugie wyniki uzyskał model globalnie optymalny. Największe wartości Q_u dla danych uczących posiada model LO i GWS a dla danych testujących modele GWS i GZK.

Najmniejsze wskaźniki jakości modeli lokalnych Q_1 , Q_2 i Q_3 posiada dla danych uczących model lokalnie optymalny, nie najmniejsze, ale i nie największe dla danych testujących. Pozostałe modele uzyskiwały zróżnicowane wartości wskaźników jakości modeli lokalnych: dla jednych modeli mniejsze dla pozostałych większe. Warto zauważyć, że w modelu globalnie optymalnym – z ograniczeniami - zewnętrzną funkcję kary, dla danych uczących wskaźniki jakości na modele lokalne nie przekroczyły dopuszczalnych wartości, a w pozostałych modelach oprócz LO, wskaźniki te były przekraczane. Podobnie dla danych testujących, w modelu GZK wskaźniki dla modeli lokalnych były najmniejsze oprócz Q_1 . Modelem globalnie optymalnym z ograniczeniami na modele lokalne – wewnętrzną funkcję kary uzyskał mniejsze wskaźniki Q i Q_u w porównaniu z modelem LO, natomiast model LO uzyskał mniejsze wartości wskaźników Q_1 i Q_2 niż GWK.

Błędy APE dla danych uczących dla Q_u wahają się od 0.61 do 1.32 procent, a dla danych testujących wahają się od 1.39 do 2.16 procent.

Podsumowując rozważania dotyczące zaproponowanych modeli rozważanego procesu po zakończeniu procesu uczenia sieci wskaźniki jakości dla danych testujących nie różnią się znacząco od wskaźników dla danych uczących. Świadczy to o uzyskaniu prawidłowych modeli procesu. Modele nabyły zdolność uogólniania.

Na podstawie otrzymanych wyników można zauważyć, że w modelu globalnie optymalnym błędy obliczane według przyjętych kryteriów są zazwyczaj najmniejsze. Z kolei w modelu lokalnie optymalnym błędy są często jedne z największych, zwłaszcza Q i Q_u , a najmniejsze dla Q_1 , Q_2 , Q_3 .

Modele globalnie optymalne miały małe wskaźniki, zwłaszcza Q , a wartości pozostałych wskaźników były nieco większe od minimalnych. Modele globalnie optymalne z uwzględnieniem jakości modeli lokalnych – syntetyczny wskaźnik jakości, uzyskały bardzo dobre wskaźniki dla modeli lokalnych, choć nie najlepsze, bo te ma model LO. Pozostałe wskaźniki w modelu GWS są zadowalające. W modelu globalnie optymalnym z uwzględnieniem jakości modeli lokalnych – zewnętrzna funkcja kary, wskaźniki Q i Q_u są zazwyczaj większe od wskaźników w innych modelach, zwłaszcza dla danych uczących. Jednak dla danych testujących często jedne z najmniejszych. Z drugiej strony wskaźniki jakości modeli lokalnych w GZK są równe lub mniejsze od dopuszczalnych, co nie zawsze dzieje się w innych modelach. Dla danych testujących zazwyczaj wszystkie wskaźniki nieco większe, więc i ograniczenia są przekraczane. Są jednak przypadki, w których nie wszystkie ograniczenia są przekraczane.

Najmniejsze wskaźniki Q_u posiadają modele globalnie optymalne - szczególny przypadek. Pozostałe wskaźniki modele te miały największe. Wynika to z faktu, że w modelach globalnie optymalnych – szczególny przypadek minimalizowane było kryterium oceniające tylko różnice sygnałów na wyjściu modelu i obiektu złożonego, inne błędy były pomijane.

Przedstawione w tabelach wyniki dają także informacje o szybkości zbieżności parametrów modeli do ich wartości optymalnych. Szybsza zbieżność w modelu lokalnie optymalnym wynika z tego, że modele lokalne, które są z osobna optymalizowane, posiadają mniej parametrów (są elementami modelu złożonego). Brak ograniczeń dla modeli lub mniejsza liczba składników w kryterium wpływa również na zwiększenie szybkości uczenia modeli (sieci).

Modele sieciowe A , B , C miały jednakowe wagi początkowe i taką samą liczbę neuronów, ale dane uczące zostały inaczej przeskalowane dla każdego modelu (patrz sekcja 4.2) po 400 cyklach optymalizacji uzyskano różniące się wskaźniki jakości w tych modelach. Najmniejsze wskaźniki uzyskano w modelu A , w którym dane skalowano wg. „Min-Max” (4.3). Największe wskaźniki miał model C , w którym skalowanie odbywało się wg. „Średniej” (4.2). W modelu B , wskaźniki były większe niż w modelu A ale mniejsze niż w modelu C .

W modelach A i D skalowanie było jednakowe wg. (4.3), lecz modele te różniły się liczbami neuronów w warstwach ukrytych (tab. 4.1) o jeden lub dwa neurony. Modele te miały jednak różne wagi początkowe. Po nauczaniu modeli wskaźniki jakości nie różniły się znacząco. Jednak najmniejsze wartości wszystkich wskaźników można znaleźć w modelu A dla danych uczących a w modelu D dla danych testujących.

Uzyskane wyniki modelowania wybranego procesu przemysłowego potwierdzają poprawność proponowanych metod modelowania. Można zauważyć to po analizie błędów względnych APE, które wartości często nie przekraczają jednego procenta. Nie dotyczy to wskaźnika Q , który tu jest sumą pozostałych wskaźników. Wyniki modelowania można zobaczyć na wykresach dla wybranych metod modelowania.

5. PODSUMOWANIE

Głównym celem niniejszej pracy było opracowanie metod modelowania obiektów złożonych przy użyciu sieci neuronowych. Zastosowano sieci neuronowe wielowarstwowe jednokierunkowe o nietypowej strukturze oraz opracowano algorytmy ich uczenia. Następnie praktycznie zweryfikowano te metody na obiekcie symulacyjnym. W końcu podano zastosowanie opracowanych metod do modelowania obiektu rzeczywistego. Rozważono dwa algorytmy uczenia oparte na metodzie wstecznej propagacji błędu: typowy algorytm gradientowy oraz algorytm zmodyfikowany.

Rozważania teoretyczne zweryfikowano symulacyjnie biorąc jako przykład obiekt złożony składający się z szeregowego połączenia dwóch nieliniowych elementów. Obiekt ten miał jedno wejście, jedno wyjście i jedno wejście dla drugiego elementu. Na podstawie wyników: wskaźników jakości i błędów względnych obliczonych w rozdziale trzecim, można stwierdzić, że wyniki modelowania są zadowalające. Opracowane algorytmy rozwiązują postawione zadania minimalizacji kryteriów dla poszczególnych metod modelowania. Prowadzą do wyznaczenia optymalnych parametrów modeli obiektu symulacyjnego.

Opracowane metody i algorytmy wykorzystano do zamodelowania obiektu rzeczywistego. Tym obiektem był fragment linii produkcyjnej azotynu amonu w Zakładach Azotowych w Tarnowie. Dostępne dane z procesu były zbierane w dużych odstępach czasowych (co 2 godziny), do zadań modelowania potraktowano ten obiekt jako quasi-statyczny. Wiele procesów chemicznych jest procesami dynamicznymi wolnozmiennymi, prowadzonymi w stałym punkcie pracy. Problem modelowania i identyfikacji takich procesów sprowadza się do znalezienia ich modeli statycznych. Zatem do modelowania procesu produkcji azotynu amonu wykorzystano model i metody modelowania obiektów statycznych.

Przyjęto trzy warianty modeli. W pierwszych dwóch wariantach zamodelowano obiekt złożony składający się z dwóch elementów. W trzecim wariantcie obiekt złożony składający się z trzech szeregowo połączonych elementów instalacji przemysłowej. Wskaźniki jakości otrzymanych modeli i błędy względne znajdują się w rozdziale czwartym. Przedstawiono także wybrane przebiegi modelowanego procesu wybranymi metodami dla danych uczących i testujących.

W otrzymanych modelach procesu wskaźniki jakości dla danych testujących nie różnią się znacząco od wskaźników dla danych uczących. Świadczy to o uzyskaniu poprawnych modeli procesu, które nabyły zdolność uogólniania.

Po analizie wskaźników jakości modeli można zauważyć, że w modelu globalnie optymalnym błędy obliczane według przyjętych kryteriów są zazwyczaj najmniejsze. Z kolei w modelu lokalnie optymalnym wskaźniki Q i Q_u są często jedne z największych, ale wskaźniki modeli lokalnych Q_1 , Q_2 , Q_3 są najmniejsze dla danych uczących, i jedne z najmniejszych dla danych testujących.

Można także zauważyć, że wskaźniki jakości w modelach lokalnie optymalnych różnią się znacznie od wskaźników modelach globalnie optymalnych. W modelu lokalnie optymalnym modeluje się poszczególne elementy obiektu z osobna a ich wskaźniki jakości są najmniejsze. Pozostałe wskaźniki w tym modelu osiągają większe wartości w porównaniu ze wskaźnikami w innych modelach. Z kolei model globalnie optymalny ma małe wartości wskaźników, zwłaszcza Q , a pozostałe nieco większe od najmniejszych. Model globalnie optymalny z uwzględnieniem jakości modeli lokalnych – syntetyczny wskaźnik jakości, uzyskuje bardzo dobre wskaźniki dla modeli lokalnych, choć nie najlepsze, bo te ma model lokalnie optymalny. Pozostałe wskaźniki w tym modelu są zadowalające. W modelu globalnie optymalnym z uwzględnieniem jakości modeli lokalnych – zewnętrzna funkcja kary wskaźniki Q i Q_u nie są najlepsze, zwłaszcza dla danych uczących, to jednak dla danych testujących wskaźniki te są mniejsze od innych. Z drugiej strony wskaźniki jakości modeli lokalnych w modelu globalnie optymalnym – zewnętrzna funkcja kary są równe lub mniejsze od dopuszczalnych, co nie zawsze dzieje się w innych modelach. Dla danych testujących generalnie wszystkie wskaźniki nieco większe, więc i ograniczenia są przekraczane. Występują jednak przypadki, w których nie wszystkie ograniczenia są przekraczane.

Najmniejsze wartości wskaźnika Q_u posiadają modele globalnie optymalne - szczególnie przypadek. Pozostałe wskaźniki modele te miały zdecydowanie największe. Wynika to z faktu, że w modelu globalnie optymalnym – szczególny przypadek, minimalizowane jest kryterium oceniające tylko różnice sygnałów na wyjściu globalnym modelu i wyjściu zewnętrznym obiektu złożonego, inne błędy są pomijane.

Trudno jednoznacznie ocenić model globalnie optymalny z ograniczeniami na modele lokalne – wewnętrzna funkcja kary, ponieważ otrzymano inne wyniki podczas badań symulacyjnych a inne podczas modelowania procesu. Wskaźniki jakości Q i Q_u w tym modelu otrzymane w modelach procesu są mniejsze od tych samych wskaźników w modelu lokalnie optymalnym. Są to prawidłowe wyniki. Natomiast, wskaźniki te otrzymane w

badaniach symulacyjnych są większe niż w modelu lokalnie optymalnym przy tej samej liczbie cykli optymalizacji. Można to wytłumaczyć tym, że w modelu lokalnie optymalnym proces uczenia jest szybszy, osobno optymalizowane modele lokalne posiadają mniej parametrów.

Zgodnie z oczekiwaniami, wskaźniki jakości Q_1 , Q_2 oraz Q_3 modeli lokalnych w modelu globalnie optymalnym z ograniczeniami na modele lokalne – wewnętrzna funkcja kary były większe niż te same wskaźniki w modelu lokalnie optymalnym zarówno w badaniach symulacyjnych jak i modelach procesu.

Przedstawione w tabelach wyniki po ustalonej liczbie cykli optymalizacji dają także informacje o szybkości zbieżności parametrów modeli do ich wartości optymalnych. Szybsza zbieżność w modelu lokalnie optymalnym wynika z tego, że modele lokalne, które są z osobna optymalizowane, posiadają mniej parametrów (są elementami modelu złożonego). Brak ograniczeń dla modeli lub mniejsza liczba składników w kryterium wpływa również na zwiększenie szybkości uczenia modeli (sieci). Tak jest w modelu globalnie optymalnym – szczególny przypadek. Istnienie ograniczeń, wieloskładnikowe kryteria oceny spowalniają szybkość zbieżności podczas uczenia. Można to zaobserwować w modelach: globalnie optymalnym, globalnie optymalnym z ograniczeniami na modele lokalne – syntetyczny wskaźnik jakości oraz w zewnętrznej funkcji kary.

Większa liczba składników w kryterium jakości lub ograniczeń również powoduje, że modele mają lepszą zdolność uogólniania. Można to zaobserwować dla danych testujących przy wskaźniku Q , który najmniejsze wartości osiąga najczęściej w modelu globalnie optymalnym z ograniczeniami na modele lokalne zarówno przy syntetycznym wskaźniku jakości, jaki i przy zewnętrznej funkcji kary.

Modelowany rzeczywisty obiekt złożony miał wiele wejść i wyjść w poszczególnych jego elementach. Otrzymano modele rzeczywistego obiektu złożonego o wielu wejściach, wyjściach, i wielu wejściach w jego elementach. Zatem opracowane modele i algorytmy są przydatne i skuteczne w modelowaniu obiektów o wielu wejściach i wyjściach. Również otrzymano jeden model (wariant „III”) o trzech elementach. Generalnie opracowane metody nie mają ograniczeń na liczbę wejść, wyjść czy liczbę elementów obiektu. Ograniczenia mogą jednak zaistnieć w implementacji w postaci narzędzia komputerowego.

Wydaje się, że przedstawione metody modelowania i algorytmy uczenia sieci sprawdzone na obiektach SISO i MIMO mogą być zastosowane do modelowania obiektów o innej lub bardziej złożonej strukturze niż przedstawione w niniejszej pracy.

Opisane w pracy algorytmy uczenia nietypowych sieci modelujące obiekty złożone zostały napisane przez autora w języku Borland Pascal, ponieważ nie można było użyć standardowych procedur uczenia sieci znanych pakietów z sieciami neuronowymi. Powstał symulator wykorzystujący kompilator Borland Pascal do modelowania obiektów złożonych mogący pracować na komputerze PC z systemem operacyjnym DOS lub Windows.

Do elementów nowości niniejszej pracy autor zalicza:

- opracowanie metod modelowania globalnego oraz globalnego z uwzględnieniem jakości modeli lokalnych
- zaproponowanie struktury sieci neuronowej dla odpowiedniego modelowania obiektu złożonego
- opracowanie algorytmu uczenia modelu globalnie optymalnego (metody: gradientowa i zmodyfikowana *Rprop*)
- opracowanie algorytmu uczenia modelu globalnie optymalnego z uwzględnieniem jakości modeli lokalnych (metoda: gradientowa i zmodyfikowana *Rprop*) dla:
 - - syntetycznego wskaźnika jakości,
 - - zewnętrznej funkcji kary,
 - - wewnętrznej funkcji kary.

Otrzymane modele obiektów złożonych można by wykorzystać na wiele sposobów. Przede wszystkim można je wykorzystać do opracowania optymalnych algorytmów sterowania obiektami złożonymi. Można na takich modelach bezpiecznie przeprowadzać szerokie badania symulacyjne, symulować stany awaryjne. Wykorzystanie modeli złożonych będzie zależeć od użytkownika.

Opracowane metody dają użytkownikowi dużą swobodę w wyborze metod modelowania w zależności od potrzeb. Różne metody dają inne modele obiektów. Jednymi metodami można otrzymać modele z dobrymi wskaźnikami globalnymi, czyli zadowalający model globalny. Inne metody zapewniają dobrą jakość modeli lokalnych. Inne jeszcze rozważane metody pozwalają otrzymać zadowalający model globalny, ale z zapewnieniem dostatecznych jakości wybranych wskaźników jakości.

Mimo że, metody modelowania i algorytmy były przeznaczone do obiektów statycznych to udało się uzyskać model procesu z „wolną” dynamiką, tj. model procesu produkcji azotynu

amoni. Jednak metody te raczej nie dały by dobrych wyników w modelowaniu złożonego obiektu dynamicznego. Nie zostało to jeszcze sprawdzone.

Wykorzystując metody i algorytmy przedstawione w niniejszej pracy w przyszłości można opracować metody i algorytmy modelowania nieliniowych dynamicznych obiektów złożonych. Istnieje szereg prac na temat modelowania pojedynczego dynamicznego elementu obiektu, za pomocą statycznych sieci neuronowych o odpowiedniej strukturze [29, 38, 40]. Modyfikując strukturę sieci tak, aby wykorzystać nie tylko sygnały w aktualnej chwili czasu, ale także ich opóźnienia, oraz przedstawione w niniejszej pracy algorytmy uczenia sieci o rozbudowanej strukturze i uwzględniającej wielokryterialność wydają się możliwe i celowe opracowanie metod i algorytmów modelowania dynamicznych obiektów złożonych o strukturze szeregowej.

W niniejsze pracy do modelowania wykorzystano sieci neuronowe wielowarstwowe, które mają wiele zalet, ale również pewne wady (np. optymalizacja struktury sieci, dobór odpowiednich współczynników tempa uczenia, wiedza zakodowana w postaci liczb rzeczywistych często trudnych do interpretacji). W celu wyeliminowania niektórych wad, przynajmniej częściowego, łączy się je z innymi elementami „sztucznej inteligencji” takimi, jak systemy rozmyte czy algorytmy genetyczne [30, 46]. Dlatego sieci neuronowe wykorzystane w niniejszej pracy warto połączyć z innymi elementami sztucznej inteligencji.

Literatura

- [1] Alfonso M.R., Salzano M., *The effect of public sector on the financial sector; an NN approach in a view of complexity*, Neural Nets WIRN VIETRI-98, Proc. Of the 10th Italian Workshop on Neural Nets, Springer-Verlag, London, UK, 1999, s.248-54.
- [2] Bubnicki Z., *Global Modelling and Identification of Complex Systems*, Proc. of 7th IFAC/IFORS Symp. York, UK, 1985, Identification and System Parameter Estimation, Pergamon Press, Oxford-New York-Toronto-Sydney-Frankfurt, 1985, s.161-263.
- [3] Bubnicki Z., *Identyfikacja obiektów sterowania*, PWN, Warszawa, 1974.
- [4] Bubnicki Z., *Optimization problems in large-scale systems modelling and identification*, In. A. Straszak Ed. Large Scale Systems: Theory and Applications, 1983, Pergamon Press, Oxford 1984, s.411-416.
- [5] Chen S., Bilings S.A., Grant P.M., *Nonlinear System Identification Using Neural Network*, International Journal of Control, 1990, Vol.51, No.6, s.1191-1214.
- [6] Cotter N.E., *The Stone –Weierstrass Theorem and Its Application to Neural Networks*, IEEE Transactions on Neural Networks, 1990, Vol. 1, No. 4, s.290-295.
- [7] Coulette B., Marcaillou-Ebersold S., *View based object oriented approach for complex systems modelling*, IEEE Computer. Soc. Press. Los Alamitos, USA, 1997, s.246-53
- [8] Cybenko G., *Approximation by superposition of a sigmoidal function*, Mathematic of Control, Signal, and Systems, 1989, vol.2, s.303-314.
- [9] Demuth H., Beale M., *Neural Network Toolbox User's Guide*, The MathWorks, Inc. Natick, 1993.
- [10] Drałus G., *Modelowanie globalne statycznych obiektów złożonych z wykorzystaniem sieci neuronowych*, Materiały XXIII Międzynarodowej Konferencji Z Podstaw Elektrotechniki i Teorii Obwodów, Gliwice-Ustroń, 2000, s.449-452
- [11] Drałus G., Świątek J., *A modified backpropagation algorithm for modelling static complex systems using neural network*, The 5th International Conference Neural Networks and Soft Computing, Zakopane, 2000, s. 463-468.
- [12] Drałus G., Świątek J., *Static neural network in global modelling of complex systems*, The 14th International Conference on Systems Engineering, Coventry, UK, 2000, s.547-551.
- [13] Eykhoff P., *Identyfikacja w układach dynamicznych*, PWN, Warszawa, 1980.
- [14] Findeisen W., Szymanowski J., Wierzbicki A., *Teoria i metody obliczeniowe optymalizacji*, PWN, Warszawa, 1977.
- [15] Findeisen W., *Wielopoziomowe układy sterowania*, PWN, Warszawa, 1974.
- [16] Fresewinkel T., *Modelling of complex systems by sequential identification of subsystems*, Regelungstechnik, vol.32, no.2, Feb. 1984, s.51-55, West Germany.
- [17] Funahashi K., *On the approximate realization of continuous mappings by neural networks*, Neural Networks, 1989, 2, s.183-192.

- [18] Gorzałczany M.B., *A neuro-fuzzy approach to system modelling. I. Methodology*, Archives of Control Sciences, Warsaw Univ. Technology, vol.7(43), no.1-2, Jan.-June 1998, s.121-40.
- [19] Gorzałczany M.B., *A neuro-fuzzy approach to system modelling. I. Applications*, Archives of Control Sciences, Warsaw Univ. Technology, vol.7(43), no.3-4, Jan.-June 1998, s. 267-84.
- [20] Haykin S., *Neural Networks. A Comprehensive Foundation*, Macmillan College Publishing Company, New York, Inc. 1994.
- [21] Hertz J., Krogh A., Palmer R., *Wstęp do teorii obliczeń sieci neuronowych*, WNT, Warszawa, 1993.
- [22] Hornik K., *Approximation Capabilities of Multilayer Feedforward Networks*, Neural Networks, 1991, 4, 251-257.
- [23] Hornik K., *Multilayer Feedforward Networks are Universal Approximators*, Neural Networks, 1989, 2, s.359-366.
- [24] Hornik K., *Some New Results on Neural Network Approximation*, Neural Networks, 1993, 6, s.1069-1072.
- [25] Hunt K.J., Sbarbaro D., Zbikowski R., Gawthrop P.J., *Neural Networks for Control Systems – A Survey*, Automatica, Pergamon Press Ltd, U.K., 1992, Vol.28, No.8, s.1083-1112.
- [26] Kaczorek T., *Teoria układów regulacji automatycznej*, WNT, Warszawa, 1977.
- [27] Kalogirou S.A., Panteliou S., Dentsoras A., *Artificial neural networks used for the performance prediction of a thermosiphon solar water heater*, Renewable Energy, 18, Pergamon, 1999, s.87-99.
- [28] Kolmogorov A.N., *On the representation of continuous functions of many variables by superposition of continuous functions of one variable and addition*, Doklady Akademii Nauk SSSR, 1957, s.114-:953-956, Translation in American Mathematical Society Translations, 1963, Series 2, 28, s. 55-59.
- [29] Korbicz J., Obuchowicz A., Uciński D., *Sztuczne sieci neuronowe- podstawy i zastosowania*, Akademicka Oficyna Wydawnicza PLJ, Warszawa, 1994.
- [30] Kwaśnicka H., *Obliczenia ewolucyjne w sztucznej inteligencji*, Oficyna Wydawnicza Politechniki Wrocławskiej, Wrocław, 1999.
- [31] Lippmann R.P., *An Introduction to computing with neural net*, IEEE ASSP Magazine, 1987, s.4-22.
- [32] McCulloch W.S., Pitts W.H., *A logical calculus of ideas immanent in nervous activity*, Bull., Math., Biophysics, 1943, Vol. 5, s. 115-119
- [33] Mańczak K., *Metody identyfikacji wielowymiarowych obiektów sterowania*, WNT, Warszawa, 1979.
- [34] Mańczak K., Nahorski Z., *Komputerowa identyfikacja obiektów sterowania*, PWN, Warszawa, 1983.
- [35] Masini R., Padovani E., Ricotti M.E., Zio E., *Dynamic simulation of a steam generator by neural networks*, Nuclear Engineering and Design, Elsevier, 1999, 187, s.197-213.

- [36] Meltser M., Shoham M., Manevitz L.M., *Approximating Functions by Neural Networks: A Constructive Solution in the Uniform Norm*, Neural Networks, 1996, 26, 3, 464-496.
- [37] Minsky M., Papert S., *Perceptron: An Introduction to Computational Geometry*, Cambridge, MA 1988.
- [38] Narendra K.S., Parthasarathy K., *Identification and Control of Dynamic Systems Using Neural Network*, IEEE Trans. On Neural Networks, 1990, vol 1. No.1, s.4-27.
- [39] Niederliński A., *Systemy komputerowe automatyki przemysłowej, t.2 Zastosowania*, WNT, Warszawa, 1985.
- [40] Osowski S., *Sieci neuronowe w ujęciu algorytmicznym*, WNT, Warszawa, 1994.
- [41] Osowski S., *Sieci neuronowe*, Oficyna Wydawnicza Politechniki Warszawskiej, Warszawa 1996.
- [42] Ouliddren K., Nowe A., *An approach to incremental fuzzy modeling of dynamic behaviour of functions in complex systems*, Intelligent Systems and Soft Computing for Nuclear Science and Industry, Proc. of 2nd International FLINS Workshop, World Scientific, Singapore, 1996, s.224-32.
- [43] Raj S., Diwakar S., Ranjana G., Ashish D., *Potential applications of artificial neural networks to thermodynamics: vapor-liquid equilibrium predictions*, computers and Chemical Engineering, 23, Pergamon, 1999, s.385-390.
- [44] Riedmiller M., Braun H., *RPROP – a fast adaptive learning algorithm*, Technical Report, University Karlsruhe, 1992.
- [45] Rosenblatt F., *Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms*, Spartan Books, Washington DC., 1961.
- [46] Rutkowska D., Piliński M., Rutkowski L., *Sieci neuronowe, algorytmy genetyczne i systemy rozmyte*, PWN, Warszawa-Łódź, 1997.
- [47] Seidler J., Badach A., Molisz W., *Metody rozwiązywania zadań optymalizacji*, WNT, Warszawa, 1980.
- [48] Smagt P.P., *Minimisation Methods for Training Feedforward Neural Networks*, Neural Networks, 1994, 7, 1, s.1-11.
- [49] Soderstrom T., Stoica P., *Identyfikacja systemów*, PWN, Warszawa, 1997.
- [50] Stemmalen T., Gissinger G.L., Menard C., *Bond graph modelling of complex and pluridisciplinary systems*, Proc. from the IFAC Conference, Pergamon, Oxford, UK, 1997, Part vol.1, s.133-8.
- [51] Świątek J., *Identyfikacja dwustopniowa oraz jej zastosowania w technice i biomedycie*, Prace Naukowe Instytutu Sterowania i Techniki Systemów Politechniki Wrocławskiej, Wrocław, 1987.
- [52] Tadeusiewicz R., *Elementarne wprowadzenie do techniki sieci neuronowych z przykładowymi programami*, Wydawnictwo PLJ, Warszawa, 1998,
- [53] Tadeusiewicz R., *Sieci neuronowe*, Akademicka Oficyna Wydawnicza, Warszawa, 1993.
- [54] Tzafestas S.G., *Engineering Systems with Intelligence. Concepts, Tools and Applications*, Kluwer Academic Publisher, Dordrecht, NL, 1991, s.125-32.

- [55] Tzafestas S.G., Verbruggen H.B.: *Artificial Intelligence in Industrial Decision Making, Control and Automation*, Kluwer Academic Publisher, London, 1998.
- [56] Venkatesh Sr., Dahleh MA., *System Identification of Complex Systems; Problem Formulation and Results*, IEEE Proceedings, 1997.
- [57] Widrow B., Hoff M.E., *Adaptive Switching Circuits*, In *Convention Record*, Part 4, IRE Wescon Connection record, New York, 1960, s.96-104.
- [58] Widrow B., Lehr M., *30 years of adaptive neural networks*, Proc. IEEE, 1990, vol.78, s.1414-1441.
- [59] Zell A., *SNNS – Stuttgart Neural Network Simulator, User Manual*, Stuttgart, 1993.
- [60] Zhang P., Li R., *A New Approach to Fuzzy Identification for Complex Systems*, FUZZ-IEEE'96, New York, 1996, Part. vol.2, s.1308-13.
- [61] Żurada J., Barski M., Jędruch W., *Sztuczne sieci neuronowe*, PWN, Warszawa, 1996.

