

ZASTOSOWANIE SIECI NEURONOWYCH W AUTOMATYCE. PRZEGLĄD WYBRANYCH MODELI I STRUKTUR

Józef KORBICZ

Wyższa Szkoła Inżynierska
Instytut Robotyki i Inżynierii Oprogramowania
ul. Podgórna 50, 65-246 Zielona Góra
e-mail: korbicz@irio.wsi.zgora.pl

W danej pracy o charakterze przeglądowym rozpatruje się wybrane problemy zastosowania sztucznych sieci neuronowych do modelowania/identyfikacji oraz sterowania układów dynamicznych. W pierwszej części opisuje się modele stosowanych sztucznych neuronów: statyczne, dynamiczne oraz z pamięcią. Następnie pokazane są możliwe architektury sieci neuronowych i ich zastosowanie do identyfikacji nieparametrycznej, jak również w układach sterowania adaptacyjnego nieliniowych obiektów dynamicznych. W pracy nie rozpatruje się algorytmów uczenia omawianych architektur sieci neuronowych.

1. Wstęp

Intensywny rozwój w ostatnich latach metod i technik sztucznej inteligencji, w tym systemów ekspertowych, zbiorów rozmytych, algorytmów genetycznych oraz sztucznych sieci neuronowych wywiera ogromny wpływ na zasady projektowania układów sterowania obiektami dynamicznymi. Tradycyjne (analityczne) metody projektowania układów sterowania dotyczą przede wszystkim obiektów liniowych oraz wybranych klas obiektów nieliniowych. Trudności związane z projektowaniem układów sterowania silnie nieliniowych obiektów metodami tradycyjnymi określają potencjalne obszary zastosowań metod sztucznej inteligencji, a w szczególności technik sieci neuronowych [6, 9, 13, 25].

O atrakcyjności sieci neuronowych jako nowych metod i technik do projektowania układów sterowania decydują przede wszystkim takie ich własności jak możliwość aproksymacji dowolnych nieliniowości oraz możliwość uczenia się, czyli adaptacji do zmian charakterystyk obiektu i zakłóceń. W ostatnich latach zagadnienia związane z możliwościami zastosowania sieci neuronowych do rozwiązywania różnych problemów układów sterowania były rozpatrywane w wielu pracach. Stosunkowo szeroki przegląd takich możliwości omawia się w pracy zbiorowej [16], artykule przeglądowym [10] oraz rozdziale książki [26].

W niniejszej pracy rozpatruje się różne architektury sieci neuronowych budowanych z wykorzystaniem statycznych oraz dynamicznych modeli sztucznych neuronów. Wybór omawianych architektur dokonano z punktu widzenia możliwości ich zastosowania do rozwiązywania zagadnień modelowania/identyfikacji nieparametrycznej, oraz sterowania adaptacyjnego nieliniowymi obiektami dynamicznymi. Rozpatruje się różne struktury neuronowych układów identyfikacji nieparametrycznej (równoległe, szeregowo-równoległe), oraz neuronowych układów sterowania adaptacyjnego (sterowanie bezpośrednie oraz pośrednie). W pracy nie omawia się algorytmów uczenia analizowanych architektur sieci neuronowych.

2. Modele neuronów

W ogólnym przypadku architektury sieci neuronowych zależą od sposobu połączenia w jedną sieć zadanej liczby sztucznych neuronów. Architektury sieci należy rozpatrywać zarówno z punktu widzenia stosowanych modeli matematycznych neuronów jak i sposobu ich połączenia.

Model statyczny. Podstawowym modelem matematycznym sztucznego neuronu jest model McCullocha i Pittsa [25] opisany równaniem:

$$y = f\left(\sum_{i=1}^n w_i u_i + u_0\right), \quad (1)$$

gdzie u_i , $i = 1, 2, \dots, n$ są wejściami neuronu, u_0 – wejściem progowym, $w_i = \text{const.}$ współczynniki wag, $f(\cdot)$ – nieliniowa funkcja aktywacji w postaci skoku jednostkowego, funkcji sigmoidalnej lub tangensu hiperbolicznego [13]. Przedstawiony model neuronu (1) jest modelem statycznym, w którym nie uwzględnia się opóźnień w przesyłaniu sygnałów.

Modele dynamiczne. W ogólnym przypadku modele dynamiczne neuronów można otrzymać poprzez wprowadzenie do modelu statycznego (1) jednego z możliwych rodzajów sprzężeń zwrotnych [27].

i) Sprzężenie w synapsie. W danym przypadku zmodyfikowana wersja modelu McCullocha-Pittsa przyjmuje postać [27]:

$$y(k) = f\left(\sum_{i=0}^n G_i(z^{-1}) u_i(k)\right), \quad (2)$$

gdzie $k=0, 1, 2, \dots$ oznacza dyskretny czas, $G_i(z^{-1})$ – liniową dyskretną transmitancję operatorową o zadanej postaci.

W porównaniu z modelem (1), w danym przypadku współczynniki wagowe w_i modelowane są przy pomocy transmitancji $G_i(z^{-1})$. Przykładem modelu neuronu ze sprzężeniem w synapsie jest model opisany równaniem [8]:

$$\varphi(k) = -b_1 \varphi(k-1) - b_2 \varphi(k-2) + a_0 u(k) + a_1 u(k-1) + a_2 u(k-2), \quad (3a)$$

$$y(k) = f(a_s, \varphi(k)), \quad (3b)$$

gdzie $\mathbf{a} = [a_0 \ a_1 \ a_2]^T$ oraz $\mathbf{b} = [b_1 \ b_2]^T$ są wektorami współczynników wag, $\varphi(k)$ – sygnał potencjału membranowego (aktywacji), a_s – współczynnik wzmocnienia regulujący pochylność charakterystyki nieliniowej $f(\cdot)$. Alternatywnie model (3) możemy opisać transmitancją operatorową o postaci:

$$G(z^{-1}) = \frac{a_0 + a_1 z^{-1} + a_2 z^{-2}}{1 + b_1 z^{-1} + b_2 z^{-2}} \quad (4)$$

Należy zauważyć, że w modelu (3), (4) dla uproszczenia zapisu przyjęto tylko jedno wejście neuronu $u(k)$.

ii) Sprzężenie wyjściowe. W najprostszym przypadku model neuronu dynamicznego ze sprzężeniem zwrotnym wyjściowym opisany jest równaniem [7, 8]:

$$y(k) = f\left(c y(k-1) + \sum_{i=1}^n w_i u_i(k)\right), \quad (5)$$

gdzie $c = \text{const}$ jest współczynnikiem. W danym modelu sygnał sprzężenia zwrotnego jest sygnałem wyjściowym opóźnionym o jeden krok czasowy, $y(k-1)$. W ogólniejszym przypadku sygnał wyjściowy $y(k)$ może być opóźniony o m kroków i wówczas równanie (5) przepiszemy do postaci

$$y(k) = f \left(\sum_{j=1}^m c_j y(k-j) + \sum_{i=0}^n w_i u_i(k) \right), \quad (6)$$

gdzie $c_j, j=1, 2, \dots, m$ są współczynnikami określającymi intensywność sprzężenia zwrotnego wyjściowego.

iii) Sprzężenie aktywacyjne. W takich modelach neuronu wejścia u_l przyjmuje się jako opóźnione sygnały potencjału membranowego, czyli $u_l(k) = \varphi(k-l)$, $l=1, 2, \dots, n_l$ przy czym $n > n_l$, co oznacza że tylko część wejść neuronu $u_i, i=1, 2, \dots, n$ stanowią sygnały $u_i, i=1, 2, \dots, n_l$.

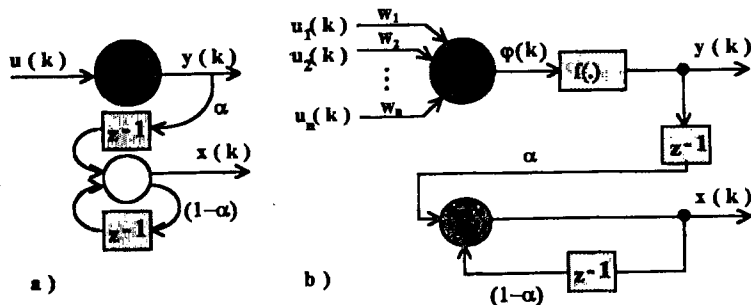
Innym modelem dynamicznym neuronu jest model z elementami pamięciowymi [23, 27]. Taki model można opisać równaniem [27]:

$$y(k) = f \left(\sum_{i=0}^n w_i u_i(k) + \sum_{i=0}^{n^*} d_i x_i(k) \right), \quad (7)$$

gdzie $x_i(k), i=0, 1, \dots, n$ są wyjściami neuronów pamięciowych z poprzedniej warstwy sieci, $d_i = \text{const}$ – współczynniki wzmocnienia. Sygnał neuronu pamięciowego określa równanie:

$$x_i(k) = \alpha y(k-1) + (1-\alpha)x_i(k-1), \quad (8)$$

gdzie $\alpha = \text{const}$. Formalnie model neuronu z pamięcią składa się z neuronu McCullocha-Pittsa oraz neuronu pamięciowego (rys. 1) [23].



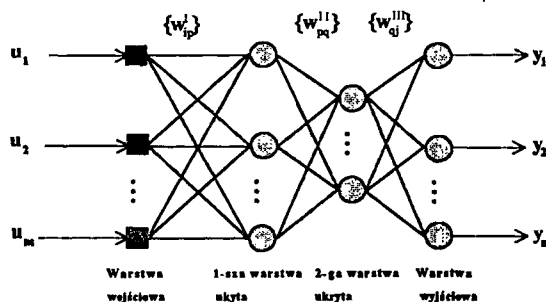
Rys. 1. Struktury modeli neuronów z pamięcią: a) schemat ogólny, b) schemat szczegółowy

3. Architektury sieci neuronowych

Powyższe modele matematyczne neuronów z punktu widzenia ich możliwości przetwarzania informacji charakteryzują się nieznaczoną mocą obliczeniową. Jedynie odpowiednie ich połączenie w struktury bardziej złożone daje układy sieci o znacznych możliwościach obliczeniowych. Biorąc pod uwagę różne możliwe sposoby połączenia neuronów można w zasadzie wyróżnić dwie grupy sieci neuronowych [19], które zarówno z punktu widzenia ich zastosowania w automatyce jak i innych dziedzinach zasługują na uwagę.

3.1. Sieci jednokierunkowe

Sieci jednokierunkowe modelują zależności statyczne. Uzasadnienie teoretyczne ich stosowania wynika z twierdzenia Cybenki [2] o aproksymacji funkcji nieliniowej z dowolnie dużą dokładnością za pomocą sieci jednokierunkowej z jedną warstwą ukrytą. Typową strukturę takiej sieci przedstawia rys.2.



Rys.2. Struktura sieci jednokierunkowej – perceptron wielowarstwowy

Formalnie nieliniowe przetwarzanie neuronowe przez sieć pokazaną na rys.2 można opisać zależnością [13, 25]:

$$y = \phi_3 \{ W^III \phi_2 [W^II \phi_1 (W^I u)] \}, \quad (9)$$

gdzie ϕ_1 , ϕ_2 i ϕ_3 oznaczają nieliniowe operatory macierzowe definiujące neuronowe przetwarzanie sygnałów przez kolejne warstwy: 1-szą, 2-gą i wyjściową; W^I , W^II i W^III – macierze współczynników wagowych określających intensywności połączeń pomiędzy neuronami w sąsiednich warstwach; $u = [u_1, u_2, \dots, u_m]^T$, $y = [y_1, y_2, \dots, y_n]^T$ – odpowiednio wektory wejść i wyjść. Jak widać z rys.2 sieci tego typu nie posiadają sprzężeń zwrotnych, a jednokierunkowy przepływ sygnałów od wejścia do wyjścia uzasadnia nazwę takich sieci. Aktualnie sieci jednokierunkowe znalazły najszersze zastosowanie w wielu dziedzinach, w tym i w automatyce [16].

Do grupy sieci jednokierunkowych należą również sieci neuronowe z radialnymi funkcjami bazowymi [3, 22]. Struktura takich sieci jest dwuwarstwowa i składa się z warstwy ukrytej oraz warstwy wyjściowej (rys.3). Warstwę ukrytą tworzą neurony opisane przy pomocy scentrowanych funkcji bazowych w przestrzeni wejść u . Zasadę działania takiej sieci opisują zależności [17]:

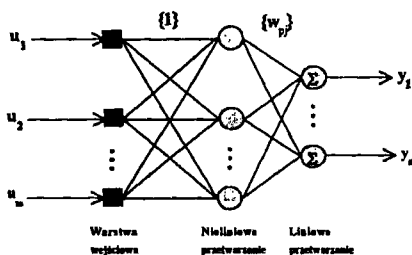
$$y_j = \sum_{p=1}^{N_h} w_{pj} h_p, \quad j=1,2,\dots,n, \quad (10)$$

gdzie y_j jest wyjściem j -tego neuronu w warstwie wyjściowej, N_h – liczbą neuronów w warstwie ukrytej, w_{pj} – współczynniki wag połączeń pomiędzy warstwą ukrytą i wyjściową, oraz h_p – wyjściem p -tego neuronu warstwy ukrytej zadanym zależnością [3, 17]:

$$h_p = \sum_{i=1}^{N_h} \phi_p(\|u - u_p^c\|) = \sum_{i=1}^{N_h} \phi_p(r_p), \quad p=1,2,\dots,N_h, \quad (11)$$

przy czym $r_p = \|u - u_p^c\|$ oznacza odległość euklidesową pomiędzy punktem $u \in U$ z przestrzeni U z centrum u_p^c radialnej funkcji bazowej ϕ_p . Do najczęściej wybieranych

funkcji bazowych zalicza się funkcje gaussowskie [3, 22] : $\phi(r)=\exp(-r^2/\rho^2)$ oraz odwrotne funkcje kwadratowe: $\phi(r)=(r^2+\rho^2)^{-1/2}$, przy czym współczynnik ρ określa szerokość parametryzującą geometryczny kształt funkcji $\phi(r)$.



Rys.3. Struktura sieci z radialnymi funkcjami bazowymi

W porównaniu z sieciami jednokierunkowymi typu perceptronowego, sieci z radialnymi funkcjami bazowymi zawierają dodatkowo dwa nieznanne parametry u_p^c oraz ρ , które należy wyznaczyć. W konkretnych zastosowaniach parametry te przyjmuje się jako wielkości stałe [3, 17, 22].

3.2. Sieci rekurencyjne

Sieci rekurencyjne, nazywane także sieciami dynamicznymi, różnią się zasadniczo od sieci statycznych tym, że w ich strukturze występują sprzężenia zwrotne. Są to systemy dynamiczne o dużych i wciąż niecałkowicie zbadanych możliwościach modelowania.

Ze względu na sposób działania sieci rekurencyjne można umownie podzielić na dwie grupy: relaksacyjne oraz standardowe. Sieci relaksacyjne z zadanego stanu początkowego lub przy zadanej kombinacji sygnałów wejściowych po zakończeniu procesu przejściowego osiągają stan równowagi, np. sieć Hopfielda [13]. Z kolei zadaniem sieci standardowych jest przetwarzanie sekwencji sygnałów wejściowych na odpowiednią sekwencję sygnałów wyjściowych.

Z uwagi na możliwości występowania różnych sprzężeń zwrotnych, sieci rekurencyjne dzielimy na:

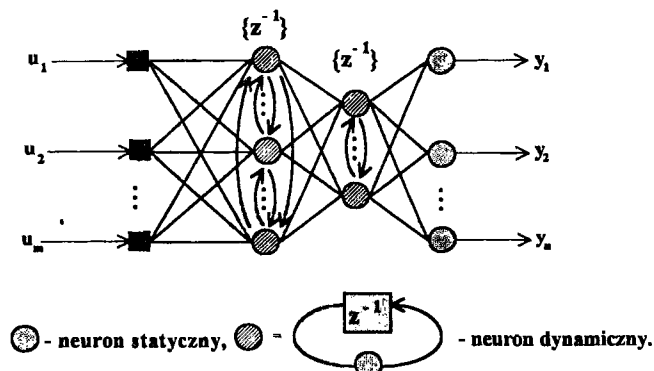
- *sieci lokalnie rekurencyjne* [14, 27] – sprzężenia zwrotne występują tylko wewnątrz modeli neuronów. Sieci takie posiadają strukturę analogiczną jak sieci statyczne, lecz zbudowane są z wykorzystaniem modeli dynamicznych neuronów.

- *sieci globalnie rekurencyjne* [14, 20] – sprzężenia zwrotne mogą występować pomiędzy neuronami różnych warstw lub tej samej warstwy. W szczególnym przypadku gdy sprzężenia zwrotne łączą warstwę wyjściową z wejściową, sieci takie nazywamy sieciami zewnętrznymi rekurencyjnymi.

Architekturę o najbardziej ogólnym charakterze posiada sieć neuronowa zaproponowana przez Williamsa i Zipsera [28]. W sieci tej zakłada się, że istnieje możliwość połączenia każdego neuronu z dowolnym innym, czyli dopuszcza się występowanie zarówno lokalnych jak i globalnych sprzężeń zwrotnych. Pomimo wielu zalet np. możliwość aproksymacji szerokiej klasy zależności dynamicznych, sieci tego typu są wolno zbieżne oraz czasami niestabilne. Mniej ogólny charakter posiada architektura sieci zaproponowana przez Jordana i Elmana [4]. W sieci takiej oprócz warstwy ukrytej znajduje się dodatkowa warstwa neuronów, w której przechowywane są bieżące wartości potencjałów membranowych (sygnałów aktywacji) neuronów warstwy ukrytej. Zadaniem

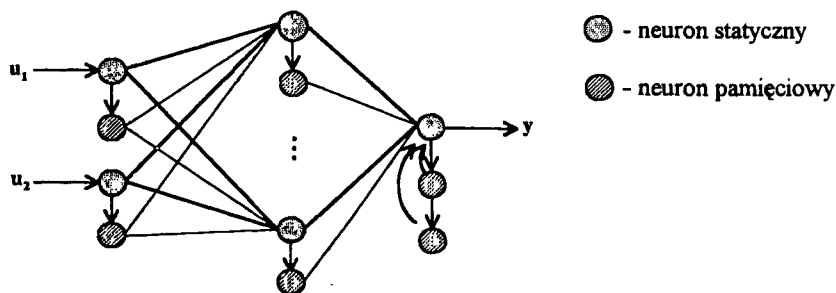
dotodkowej warstwy jest opóźnianie tych wartości o jeden takt, a następnie ich przesyłanie na dodatkowe wejścia neuronów warstwy ukrytej. Przepustowość tych połączeń nie podlega adaptacji, tj. współczynniki wagowe są równe jedności.

Rekurencyjny perceptron wielowarstwowy [20]. Architektura rekurencyjnego perceptronu wielowarstwowego różni się od standardowego perceptronu statycznego występującymi sprzężeniami zwrotnymi pomiędzy neuronami tej samej warstwy oraz dodatkowo stosowanymi dynamicznymi modelami neuronów. Przykładowa struktura takiego perceptronu pokazana jest na rys.4. W strukturach tego typu wszystkie dodatkowe połączenia pomiędzy neuronami tej samej warstwy opóźniają sygnały o jeden takt, co umownie oznaczono przez $\{z^{-1}\}$. W przypadku braku sprzężeń pomiędzy różnymi neuronami, rekurencyjny perceptron wielowarstwowy nazywa się diagonalną siecią rekurencyjną [14]. W sieciach diagonalnych efekty dynamiki uzyskuje się dzięki zastosowaniu dynamicznych modeli neuronów.



Rys.4. Struktura rekurencyjnego perceptronu wielowarstwowego,

Interesującym przykładem sieci lokalnie rekurencyjnej z zachowaniem ogólnej struktury perceptronu wielowarstwowego jest sieć z **neuronami pamięciowymi** (rys. 5). W sieci tego typu (rys. 5) [23] oprócz lokalnych sprzężeń zwrotnych wynikających z przyjętego modelu neuronu z pamięcią, występują dodatkowe połączenia pomiędzy neuronem pamięciowym danej warstwy a neuronami statycznymi warstwy następnej. Z kolei każde wyjście neuronu warstwy wyjściowej jest podawane na wejście kaskadowej struktury złożonej z dwóch neuronów pamięciowych. Sieć z neuronami pamięciowymi jest nieliniowym układem dynamicznym, którego stabilność wymaga wprowadzenia dodatkowych ograniczeń na współczynniki wagowe [23].



Rys.5. Struktura sieci z neuronami pamięciowymi

4. Identyfikacja obiektów dynamicznych

Mimo, że znanych jest obecnie wiele różnych metod i algorytmów identyfikacji parametrycznej i nieparametrycznej dynamicznych obiektów liniowych i wybranych klas obiektów nieliniowych [5], wciąż brak jest ogólnych i efektywnych metod identyfikacji nieliniowych obiektów dynamicznych o nieznanej strukturze. W ostatnich latach problem ten intensywnie rozwiązywany jest między innymi przy pomocy sieci neuronowych [11, 19, 23], zarówno statycznych [13] jak i dynamicznych [24].

Dla uproszczenia problem identyfikacji przedstawimy dla obiektu o jednym wejściu $u(k)$ i jednym wyjściu $y_p(k)$, przyjmując opis w postaci:

$$y_p(k+1) = f(y_p(k), y_p(k-1), \dots, y_p(k-n+1); u(k+1), u(k), \dots, u(k-m+1)), \quad (12)$$

gdzie $f(\cdot)$ jest nieliniową nieznaną funkcją obiektu. Dla obiektu (12) można rozpatrzeć dwa modele identyfikacji z neuronowymi układami [11, 13, 19]: model równoległy oraz szeregowo-równoległy.

W modelu równoległym wyjście neuronowego identyfikatora $y_p^m(k+1)$ w chwili $(k+1)$ -tej zależy od opóźnionych wartości wyjść $\{y_p^m(k), y_p^m(k-1), \dots, y_p^m(k-n+1)\}$, oraz wejść $\{u(k+1), u(k), \dots, u(k-m+1)\}$. Korzystając z równania (12) zapiszemy:

$$y_p^m(k+1) = f(y_p^m(k), y_p^m(k-1), \dots, y_p^m(k-n+1); u(k+1), u(k), \dots, u(k-m+1)), \quad (13)$$

przy czym $y_p(k)$ (wyjście obiektu) należy wykorzystywać jako sygnał uczący w chwili k -tej. Stosowanie modelu równoległego (13) jest ograniczone i utrudnione, gdyż nawet przy założeniu stabilności identyfikowanego obiektu nie można zapewnić stabilności neuronowego układu identyfikacji. Z tego względu częściej stosowanym modelem identyfikacji jest model szeregowo-równoległy.

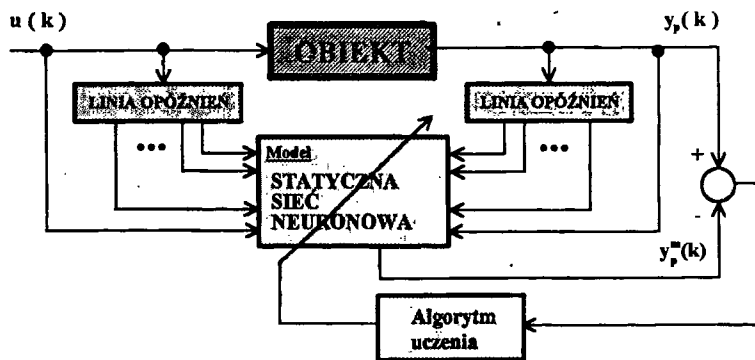
W modelu szeregowo-równoległym opisanym równaniem:

$$y_p^m(k+1) = f(y_p(k), y_p(k-1), \dots; y_p^m(k), y_p^m(k-1), \dots; u(k+1), u(k), \dots), \quad (14)$$

wyjście sieci neuronowej $y_p^m(k+1)$ zależy od opóźnionych wyjść obiektu $\{y_p(k), y_p(k-1), \dots\}$, wyjść sieci (niekoniecznie) $\{y_p^m(k), y_p^m(k-1), \dots\}$ oraz wejść $\{u(k+1), u(k), \dots\}$. Model ten należy do najczęściej stosowanych modeli identyfikacji. Modele równoległy i szeregowo-równoległy stają się tożsame w przypadku spełnienia warunku $y_p(k) = y_p^m(k)$. Wynika stąd możliwość stosowania dwuetapowego algorytmu identyfikacji: na pierwszym etapie wykorzystuje się model szeregowo-równoległy, a na drugim po spełnieniu warunku $y_p(k) \approx y_p^m(k)$ model równoległy.

4.1. Sieci statyczne

Zastosowanie neuronowych sieci statycznych w układach identyfikacji nieliniowych obiektów dynamicznych możliwe jest dzięki wprowadzeniu specjalnej linii opóźnień sygnałów wejściowych $u(k)$, oraz wyjściowych $y_p(k)$ obiektu. Struktura takiego układu identyfikacji przedstawiona jest na rys.6. Zaletą rozwiązania jest możliwość stosowania standardowej wersji (statycznej) algorytmu propagacji wstecznej do uczenia sieci (adaptacji jej współczynników wag), z uwagi na brak sprzężeń zwrotnych.

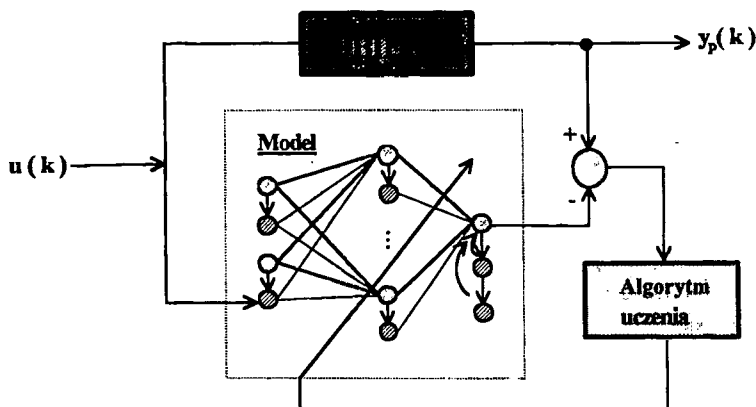


Rys.6. Układ identyfikacji ze statyczną siecią neuronową

Do istotnych wad stosowania sieci statycznych w danym przypadku należy rosnąca złożoność sieci neuronowej wraz ze wzrostem liczby wejść i wyjść obiektu oraz długości linii opóźnień.

4.2. Sieci rekurencyjne

Podstawową zaletą sieci rekurencyjnych w układach identyfikacji jest ich zdolność do przechowywania informacji oraz dynamicznego odwzorowywania wejście-wyjście. Przykładem zastosowania sieci rekurencyjnych jest szeregowo-równoległy układ identyfikacji z siecią o strukturze perceptronu z neuronami pamięciowymi (rys. 7) [23].



Rys.7. Neuronowy układ identyfikacji z neuronami pamięciowymi

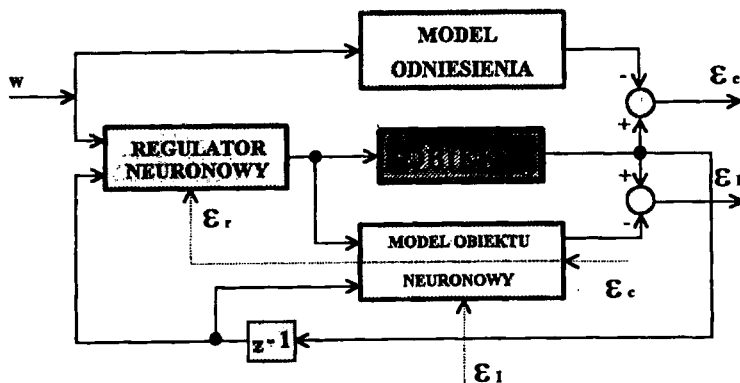
W danym rozwiązaniu szereg trudności związanych ze stosowaniem sieci statycznych zostało usuniętych, lecz pewnemu dodatkowemu skomplikowaniu uległ algorytm uczenia [23]. Podobne wyniki otrzymano w pracy [14], korzystając z innego modelu dynamicznego neuronu.

5. Sterowanie adaptacyjne

W porównaniu z układami liniowymi dla których istnieje rozbudowana teoria sterowania adaptacyjnego, tylko dla wybranych przypadków można zaprojektować efektywny układ sterowania adaptacyjnego dla obiektów nieliniowych. Neuronowe układy sterowania stwarzają możliwość rozwiązania tego złożonego problemu w szczególności dla układów adaptacyjnych z modelem odniesienia [18]. W danym przypadku dla obiektu z zadanym modelem i wejściem odniesienia należy określić wejście obiektu w taki sposób aby wyjścia obiektu i modelu odniesienia były równe sobie z zadaną dokładnością ; obiekt powinien śledzić wyjście modelu odniesienia. W zasadzie istnieją dwie strategie rozwiązania problemu sterowania adaptacyjnego z modelem odniesienia [10, 19, 23].

Układ sterowania bezpośredniego. W takim układzie parametry regulatora są bezpośrednio dostrajane na podstawie wartości sygnału błędu jaki występuje pomiędzy wyjściem obiektu i modelu odniesienia. Należy jednak zauważyć, że ze względu na nieznany i nieliniowy model obiektu nie jest możliwa adaptacja parametrów regulatora neuronowego. Nie jest możliwe przesłanie właściwego sygnału błędu z wyjścia do regulatora neuronowego ponieważ nieznany obiekt znajduje się pomiędzy regulatorem a dostępnym sygnałem błędu na wyjściu.

Układ sterowania pośredniego. W układzie z taką strategią najpierw przeprowadza się identyfikację parametrów wybranego modelu, a następnie na jej podstawie dokonuje się adaptacji parametrów regulatora. Podstawowa trudność stosowania sieci neuronowych do syntezy takich układów sterowania polega na tym, że identyfikator neuronowy jest identyfikatorem nieparametrycznym [23] i nie ma prostych zależności pomiędzy adaptowanymi wagami sieci a parametrami obiektu [19]. Jedną z metod rozwiązania tego problemu jest zastosowanie metody nazywanej *modelowaniem wprost i odwrotnym* [16] (rys.8).



Rys.8. Struktura neuronowego układu sterowania adaptacyjnego z modelem odniesienia

W układzie przedstawionym na rys.8 neuronowy model obiektu służy jako kanał do propagowania błędu z wyjścia obiektu na jego wejście. Tak uzyskany błąd na wejściu wykorzystywany jest do uczenia sieci regulatora. Przy takiej strategii uczenia model obiektu powinien być dostrajany na podstawie błędu ε_t w sposób ciągły.

Podobnie jak w przypadku neuronowej identyfikacji obiektów dynamicznych, w neuronowych układach sterowania adaptacyjnego wykorzystuje się sieci statyczne [19, 29] z dodatkowym wprowadzeniem linii opóźnień, sieci z radialnymi funkcjami bazowymi [3, 17, 22] oraz różnego typu sieci rekurencyjne np. sieci z neuronami pamięciowymi [23], sieci diagonalne [14], oraz sieci typu perceptron wielowarstwowy z dynamicznymi neuronami [8, 27].

6. Podsumowanie

W pracy o charakterze przeglądowym pokazano możliwości stosowania sztucznych sieci neuronowych do rozwiązywania nieliniowych problemów identyfikacji obiektów dynamicznych oraz układów adaptacyjnych z modelem odniesienia. Główną uwagę zwrócono na wybrane modele neuronów oraz architektury sieci, które dotychczas znalazły zastosowanie przy rozwiązywaniu wymienionych problemów teorii sterowania i automatyki. Techniki sieci neuronowych mogą być stosowane zarówno dla obiektów o znanej jak i całkowicie nieznanej strukturze. Należy zauważyć, że metody sieci neuronowych to nieparametryczne techniki identyfikacji obiektów dynamicznych: sieci neuronowe odwzorowują zależności wejście-wyjście obiektu lecz wyznaczone współczynniki wag sieci nie znajdują żadnego odniesienia do wartości współczynników identyfikowanego obiektu.

Oddzielnym zagadnieniem i bardzo ważnym jest problem uczenia wybranych struktur sieci neuronowych tj. wyznaczania współczynników wag połączeń między neuronami. Dla sieci statycznych efektywnym algorytmem uczenia jest algorytm wstecznej propagacji błędów [6, 13, 25] opracowany na podstawie gradientowej metody optymalizacji największego spadku. Z uwagi na większą różnorodność i złożoność struktur sieci dynamicznych ze sprzężeniami zwrotnymi, problem ich uczenia jest znacznie trudniejszy. Ponadto własności sieci rekurencyjnych są, w przeciwieństwie do sieci statycznych zależne od rodzaju zastosowanego algorytmu uczenia. Do znanych metod uczenia sieci dynamicznych zalicza się [11, 12]: metodę wstecznej propagacji w czasie [24], metodę modeli wrażliwości [19], oraz metody wyższego rzędu np. wykorzystujące algorytm filtru Kalmana [21]. Obszerny przegląd gradientowych algorytmów uczenia sieci jednokierunkowych i rekurencyjnych zamieszczony jest w pracy [1]. Wymienione metody uczenia różnią się złożonością obliczeniową, wymaganiami dotyczącymi pamięci komputerowej, dokładnością obliczania gradientu funkcji błędów i szybkością uczenia.

Pomimo znacznego rozwoju teorii sieci neuronowych i intensywnie prowadzonych badań nad ich zastosowaniem do nieliniowej teorii sterowania wiele istotnych problemów pozostaje nierozwiązanych i otwartych. Znanych jest wiele struktur sieci neuronowych lecz problem wyboru najlepszej nie jest sformalizowany. Otwartym i trudnym problemem jest analiza stabilności nieliniowych układów sterowania z neuronowymi regulatorami oraz modelami.

Literatura

- [1] Baldi P.: *Gradient descent learning algorithm overview. A general dynamical systems perspective*. IEEE Trans. Neural Networks, 1995, vol.6, No.1, pp.182-195.
- [2] Cybenko G.: *Approximation by superpositions of a sigmoidal function*. Math. Control Signal Systems, 1989, v.2, pp.303-314.
- [3] Elanayar S., Shin Y.C.: *Radial basis function neural network for approximation and estimation of nonlinear stochastic dynamic systems*. IEEE Trans. Neural Networks, 1994, vol.5, No.4, pp.594-603.
- [4] Elman J. L.: *Finding structure in time*. Cognitive Science, 1990, vol.14, pp.179-211.
- [5] Eykhoff P.: *Identyfikacja w układach dynamicznych*. PWN, Warszawa, 1980.
- [6] Fausett L.: *Fundamentals of Neural Networks*. Prentice Hall, Englewood Cliffs, NJ, 1994.
- [7] Frasconi P., Gori M., Soda G.: *Local feedback multilayered networks*. Neural Computation, 1992, vol.4, pp.120-130.
- [8] Gupta M.M., Rao D.H.: *Dynamic neural units with applications to the control of unknown nonlinear systems*. J. Intelligent and Fuzzy Systems, 1993, vol.1, No.1, pp.73-92.

- [9] Hertz J., Krogh A., Palmer R.G.: *Wstęp do teorii obliczeń neuronowych*. WNT, Warszawa, 1993.
- [10] Hunt K.J., Sbarbaro D., Żbikowski R., Gawthrop P.J.: *Neural networks for control systems. A survey*. Automatica, 1992, vol.28, No.6, pp.1083-1112.
- [11] Janczak A.: *Zastosowanie sztucznych sieci neuronowych do identyfikacji systemów dynamicznych. Przegląd metod*. Raport Inst. Robotyki i Inżynierii Oprogramowania, WSIInż., Zielona Góra, 1995, Nr 6/95.
- [12] Korbicz J.: *Neural network architectures used in modelling/identification and control of dynamic systems*. Proc. 8th Int. Symp. SYSTEM MODELLING CONTROL, Zakopane, 1995, vol.3, pp.54-59.
- [13] Korbicz J., Obuchowicz A., Uciński D.: *Sztuczne sieci neuronowe. Podstawy i zastosowania*. Akademicka Oficyna Wydawnicza PLJ, Warszawa 1994.
- [14] Ku Ch.-Ch., Lee K.Y.: *Diagonal recurrent neural networks for dynamic systems control*. IEEE Trans. Neural Networks, 1995, vol.6, No.1, pp.144-155.
- [15] Kuschewski J.G., Hui S., Żak S.: *Application of feedforward neural networks to dynamical system identification and control*. IEEE Trans. Control Systems Technology, 1993, vol.1, No.1, pp.37-49.
- [16] Miller W.T., Sutton R.S., Werbos P.J. (Eds.): *Neural Networks for Control*. The MIT Press, Cambridge, MA.
- [17] Moakes P.A., Beet S.W.: *Adaptive control using radial basis function networks*. Proc. IEEE Int. Conf. CONTROL'94, Coventry, UK, March, 1994, pp.1453-1457.
- [18] Narendra K.S., Annaswamy A.M.: *Stable Adaptive Systems*. Prentice Hall, Englewood Cliffs, NY, 1989.
- [19] Narendra K.S., Parthasarathy K.: *Identification and control of dynamical systems using neural networks*. IEEE Trans. Neural Networks, 1990, vol.1, No.1, pp.4-27.
- [20] Parlos A.G., Chong K.T., Atiya A.F.: *Application of the recurrent perceptron in modeling complex process dynamics*. IEEE Trans. Neural Networks, 1994, vol.5, No.2, pp.255-266.
- [21] Puskorius G.V., Feldkamp L.A.: *Neurocontrol of nonlinear dynamical systems with Kalman filter trained recurrent networks*. IEEE Trans. Neural Networks, 1994, vol.5, No.2, pp.279-297.
- [22] Sanner R.M., Slotine J.J.: *Gaussian networks for direct adaptive control*. IEEE Trans. Neural Networks, 1992, vol.3, No.6, pp.837-863.
- [23] Sastry P.S., Santharam G., Unnikrishnan K.P.: *Memory neuron networks for identification and control of dynamical systems*. IEEE Trans. Neural Networks, 1994, vol.5, No.2, pp.306-319.
- [24] Srinivasan B., Prasad U.R., Rao N.J.: *Back propagation through adjoints for identification of nonlinear dynamic systems using recurrent neural models*. IEEE Trans. Neural Networks, 1994, vol.5, No.2, pp.213-227.
- [25] Tadeusiewicz R.: *Sieci neuronowe*. Akademicka Oficyna Wydawnicza RM, Warszawa, 1993.
- [26] Trentelman H.L., Willems J.C. (Eds.): *Essays in Control: Perspectives in the Theory and its Applications*. Birkhäuser, Boston, 1993.
- [27] Tsoi A.Ch., Back A.D.: *Locally recurrent globally feedforward networks: A critical review of architectures*. IEEE Trans. Neural Networks, 1994, vol.5, No.2, pp.229-239.
- [28] Williams R.J., Zipser D.: *A learning algorithm for continually running fully recurrent neural networks*. Neural Computation, 1989, vol.1, pp.270-289.
- [29] Yabuta T., Yamada T.: *Neural network controller characteristics with regard to adaptive control*. IEEE Trans. Systems, Man, and Cybernetics, 1992, vol.22, No.1, pp.170-177.