prof. dr hab. inż. Stanisław BIEDUGNIS
dr inż. Mariusz SMOLARKIEWICZ
Katedra Programowania i Zarządzania Bezpieczeństwem
Zakład Zarządzania Ryzykiem, SGSP
dr Marcin M. SMOLARKIEWICZ
Katedra Programowania i Zarządzania Bezpieczeństwem
Zakład Zarządzania Kryzysowego, SGSP

SIECI NEURONOWE W ZAGADNIENIACH BEZPIECZEŃSTWA ZAOPATRZENIA W WODĘ

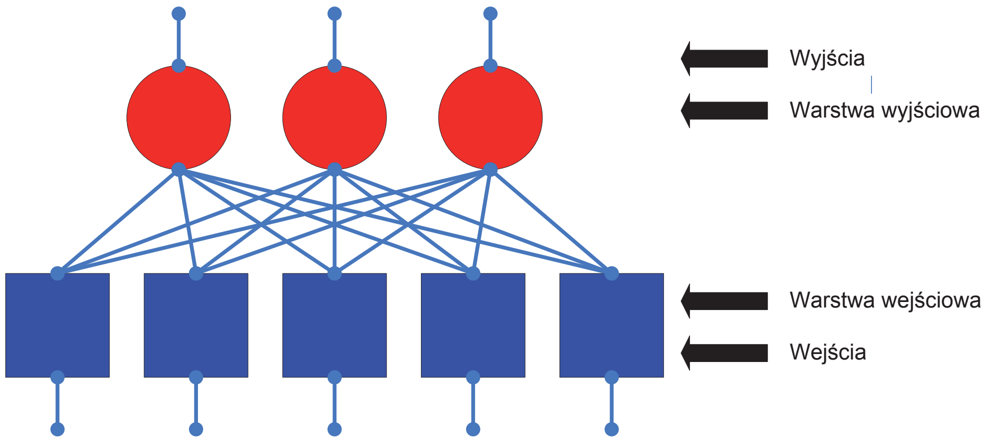
W niniejszym artykule przedstawiono zastosowanie sieci neuronowych typu Rosenblatt w modelowaniu on-line systemów zaopatrzenia w wodę, które pozwala na zwiększenie bezpieczeństwa w odniesieniu do ich prawidłowego modelowania, efektywności procesu.

In this article the use of neural network concept proposed by Rosenblatt in on-line modeling of water supply systems was presented. The use of neural network in modeling process increases the level of safety and efficiency of modeling itself.

1. Wstęp

W roku 1958 Rosenblatt [5] opracował i zbudował sztuczną sieć neuronową nazwaną perceptronem. Ten częściowo elektromechaniczny, a częściowo elektroniczny układ przeznaczony był do rozpoznawania znaków alfanumerycznych z procesem uczenia jako metodą programowania systemu. Ważnym rezultatem Rosenblatta było ponadto udowodnienie tzw. Twierdzenia o zbieżności perceptronu, które gwarantuje skończoną liczbę iteracji procesu uczenia, o ile dla zagadnienia modelowanego przy użyciu tego typu sieci optymalny układ wag istnieje. Pomimo iż działanie perceptronu nie było zadowalające z punktu widzenia zasadniczego celu (układ wykazywał dużą wrażliwość na zmianę skali rozpoznawanych obiektów oraz ich położenia w polu widzenia), był to ogromny sukces badań prowadzonych w tym zakresie [4]. Przede wszystkim był to pierwszy fizycznie skonstruowany układ symulujący sieć nerwową, który wraz ze zdolnością do uczenia się wykazywał zdolność do poprawnego działania nawet po uszkodzeniu części jego elementów. Zastosowanie tego typu sieci neuronowych w modelowaniu on-line systemów zaopatrzenia w wodę pozwala na zwiększenie bezpieczeństwa w odniesieniu do ich prawidłowego modelowania, efektywności procesu on-line i jego niezawodności.

W swojej najprostszej postaci perceptron zbudowany był z dwóch oddzielnych warstw neuronów reprezentujących odpowiednio wejście i wyjście (rys. 1). Zgodnie z przyjętą zasadą neurony warstwy wyjściowej otrzymują sygnały od neuronów warstwy wejściowej, lecz nie odwrotnie [7]. Oprócz tego neurony z danej warstwy nie komunikują się między sobą.



Rys. 1. *Perceptron*

2. Perceptron prosty

Idea perceptronu jest zawarta w czterech podstawowych zasadach:

1. Elementem składowym perceptronu jest sztuczny neuron, którego model matematyczny może być opisany funkcją aktywacji unipolarną:

$$y = \begin{cases} 1, & \varphi > 0 \\ 0, & \varphi \leq 0 \end{cases} \tag{1}$$

lub bipolarną:

$$sgn(\varphi) = \begin{cases} +1, & \varphi > 0 \\ -1, & \varphi \leq 0 \end{cases} \tag{2}$$

gdzie:

$$\varphi = \sum_{i=1}^{m} w_i u_i - \theta \tag{3}$$

przy czym w_i oznacza wagę i-tego połączenia wstępującego do elementu; u_i – wartość i-tego wejścia; θ – wartość progową funkcji aktywacji.

2. Sieć perceptronową można podzielić jednoznacznie na ściśle uporządkowane i rozłączne klasy elementów zwane warstwami, wśród których wyróżnić można warstwę wejściową i wyjściową. Pozostałe noszą nazwę warstw ukrytych [10].
3. Perceptron nie zawiera połączeń pomiędzy elementami należącymi do tej samej warstwy.
4. Połączenia pomiędzy warstwami są asymetryczne i skierowane zgodnie z ich uporządkowaniem, tzn. od warstwy wejściowej do pierwszej warstwy ukrytej, następnie od pierwszej do drugiej warstwy ukrytej, itd. aż do warstwy wyjściowej. Nie ma połączeń zwrotnych [10].

Warstwa wejściowa posiada elementy o nieco uproszczonej funkcji przejścia i jednym wejściu. Jest to swego rodzaju układ receptorów odbierających sygnały wejściowe i po wstępnym ich przetworzeniu (np. normalizacji, filtracji) przesyłających je do elementów warstwy następnej [12]. Umownie jest to warstwa zerowa sieci, stąd perceptron zawierający jedynie warstwę wejściową i wyjściową nazywany jest perceptronem jednowarstwowym lub perceptronem prostym.

W praktycznych zastosowaniach SSN (sztucznych sieci neuronowych) ustala się dla każdego elementu przetwarzającego wartość progu $\theta = 0$. Jego rolę przejmuje waga wstępującego do niego połączenia od dodatkowego elementu wejściowego (tzw. elementu progowego, *ang. bias*). Wówczas dla $w_0 = -\theta$ wzór (3) przyjmuje postać:

$$\varphi = \sum_{i=1}^{m} w_i u_i \tag{4}$$

Zaletą powyższej modyfikacji jest uzyskanie jednolitych wzorów określających zmiany wag i progów aktywacji w procesie uczenia sieci (dostrajania parametrów sieci).

Najprostszą siecią jednokierunkową jest perceptron prosty. Zbudowany jest jedynie z warstwy wejściowej i wyjściowej. Ponieważ nie istnieją połączenia między elementami warstwy wyjściowej, każdy z nich można potraktować niezależnie jako osobną sieć o m+1 wejściach i jednym wyjściu [13] (rys. 2).

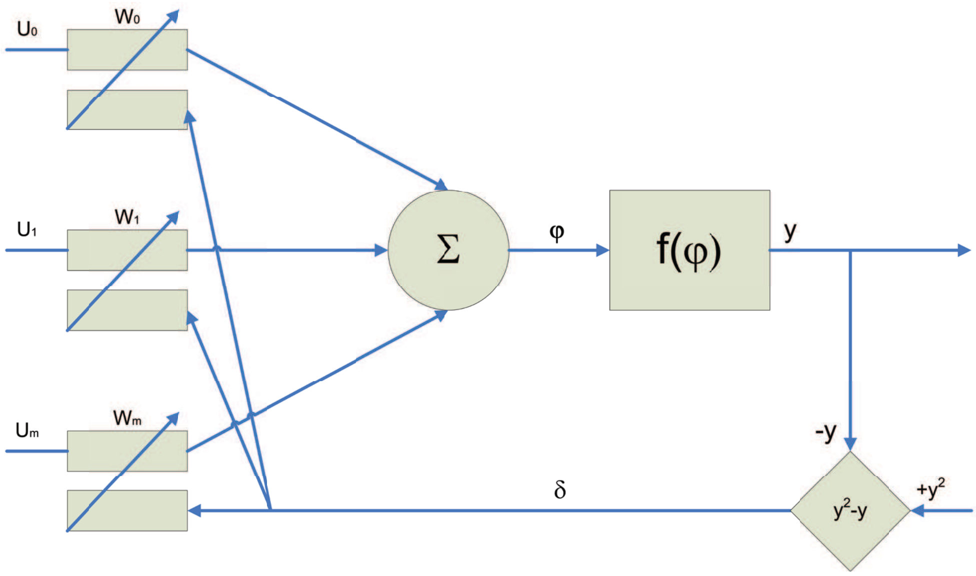
Przyjmijmy, że w bloku aktywacji elementu przetwarzającego realizowana będzie funkcja unipolarna (1).

Dany jest pewien zbiór wzorcowych wektorów wejściowych $u \in R^{m+1}$ i odpowiadających im oczekiwanych wartości wyjściowych y_z.

Algorytm uczenia można sformułować następująco:

1. Pokaż obraz wejściowy u i wyznacz wyjście y.
2. Podejmij jedną z poniższych decyzji:
   – jeśli obraz wyjścia jest poprawny, to wróć do punktu 1 i pokaż inny obraz wejściowy u;
   – jeśli sygnał wyjścia jest niepoprawny i równy 0, to dodaj wartość każdego wejścia u^i pomnożoną przez pewną liczbę η do wartości odpowiedniego współczynnika wag, w_i;
   – jeśli sygnał wyjścia jest niepoprawny i równy 1, to odejmij wartość każdego wejścia u^i pomnożoną przez pewną liczbę η od wartości odpowiedniego współczynnika wag, w_i.
3. Wróć do punktu pierwszego i pokaż inny obraz wejściowy u.

Proces uczenia według tego algorytmu kontynuuje się tak długo, aż w kolejnych iteracjach nie następuje zmiana wektora wag połączeń wstępujących do elementu przetwarzającego. Warto również ograniczyć z góry liczbę iteracji procesu na wypadek ewentualnej jego rozbieżności. Ostatnim problemem jest ustalenie kolejności prezentacji wzorców.



Rys. 2*. Element sieci – proces uczenia nadzorowanego*

Istnieją trzy możliwości:

1. Prezentować dany wzorzec dopóty, dopóki uzyska się stabilny wektor wag i następnie przejść do kolejnego wzorca.
2. Prezentować wzorce cyklicznie zgodnie z ustaloną z góry kolejnością.
3. Do każdej prezentacji losować wzorzec ze zbioru:

$$\Gamma = \left\{ \left(u^{\mu}, y_{\mu}^{z} \right) \right\}_{\mu=1}^{L} \tag{5}$$

z prawdopodobieństwem *1/L*.

Pierwsza z metod jest niewskazana [11], ponieważ sieć, ucząc się danego wzorca, traci zdolność udzielania poprawnych odpowiedzi na wzorce nauczone wcześniej. Drugi wariant unika niedogodności pierwszego, lecz istnieje obawa utraty zbieżności poprzez generowanie cyklicznych stanów sieci. Preferowany jest zatem wariant trzeci.

Jedną z podstawowych operacji w procesie uczenia elementu perceptronowego jest wyznaczanie współczynników wag w_i. Istnieje kilka metod korekcji wag:

- za pomocą ustalonego kroku wzrostu lub spadku;
- za pomocą kroku proporcjonalnego do różnicy pomiędzy sumą ważoną otrzymaną w sumatorze a oczekiwaną wartością wyjściową; w tym przypadku proces uczenia może być niestabilny;
- za pomocą kombinacji dwóch powyższych metod.

Najprostsza jest metoda ustalonego kroku wzrostu, którą można opisać równaniem:

$$w_i(k+1) = w_i(k) + \eta\delta_\mu u_\mu^i \tag{6}$$

gdzie:

$$\delta\mu = y_\mu^z - y\mu = \begin{Bmatrix} -1, y_\mu^z = 0 \wedge y_\mu = 1 \\ 0, y_\mu^z = y_\mu \\ 1, y_\mu^z = 1 \wedge y_\mu = 0 \end{Bmatrix} \tag{7}$$

przy czym k oznacza numer kroku algorytmu, $w_i(k)$ – wagę i-tego połączenia na k-tym kroku, u_i^μ – i-tą składową μ-tego wektora wejściowego u^μ, y_μ^z – żądaną odpowiedź na μ-ty wektor wejściowy u^μ, y_μ – aktualną odpowiedź na μ-ty wektor wejściowy u^μ, η – parametr uczenia.

Dla każdego kroku k losuje się odpowiadający mu numer wzorca μ. Powyższa zasada nosi nazwę reguły delty. Łatwo zauważyć, że proces uczenia to iteracyjny proces minimalizacji błędów δ_μ, $\mu = 1, \ldots, L$.

Jedna z interesujących właściwości perceptronu została sformułowana przez Rosenblatta [8] w postaci twierdzenia: jeżeli tylko istnieje taki wektor wag w, przy pomocy którego element perceptronowy odwzorowuje w sposób poprawny zbiór wzorcowych wektorów wejściowych $\{u^\mu\}$ na odpowiadający mu zbiór wartości wyjściowych $\{y_\mu{}^z\}$, to istnieje metoda uczenia tego elementu gwarantująca zbieżność do wektora w.

W roku 1969 Minsky i Papert [8] zauważyli, że wiele interesujących funkcji nie może być modelowanych przez perceptron prosty, gdyż nie jest spełniony warunek istnienia wektora wag w z twierdzenia Rosenblatta.

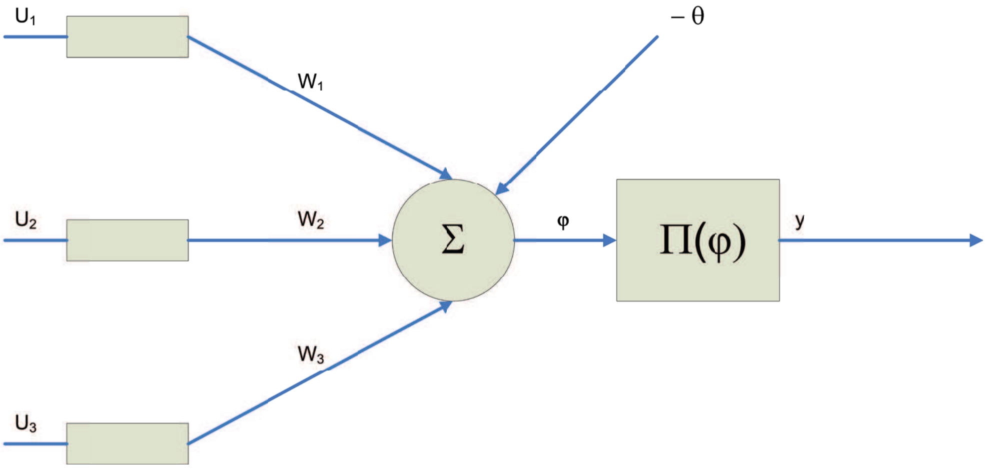
Rozważmy element o m wejściach i progu aktywacji θ (rys. 3.).

Odpowiedź y-tego elementu wyraża się wzorem:

$$\prod(\varphi) = \begin{cases} 1, \varphi > 0 \\ 0, \varphi \le 0 \end{cases} \tag{8}$$

Zatem element perceptronowy dzieli m-wymiarową przestrzeń wektorów wejściowych na dwie półprzestrzenie rozdzielone $(m-1)$ – wymiarową hiperpłaszczyzną o równaniu:

$$w^\tau u - \theta = 0 \tag{9}$$



Rys. 3. *Element o m wejściach i progu aktywacji θ*

Dla wektorów wejściowych z jednej półprzestrzeni element jest aktywny *($y = 1$)*, dla drugiej – nie *($y = 0$)*. Wymiarowa hiperpłaszczyzna nosi nazwę ***granicy decyzyjnej***.

3. Perceptron wielowarstwowy

Ograniczenie liniowej separowalności perceptronu prostego można usunąć poprzez wprowadzenie warstw ukrytych [3]. Oto struktura perceptronu wielowarstwowego (rys. 4.), przy czym $i = 1, 2, \ldots, m$; m – liczba elementów w warstwie wejściowej; $j = 1, 2, \ldots, n$; n – liczba elementów w warstwie wyjściowej; $h = 1, 2, \ldots, H$; H – liczba warstw ukrytych; k_h – 1, 2, …, K_H; K_H – liczba elementów w h-tej warstwie ukrytej; $w^{h}_{k_{h-1},\, k_h}$ – waga połączenia pomiędzy elementami k_{h-1}-tym oraz k_h-tym odpowiednio zawartymi w warstwach (h-1) i h.

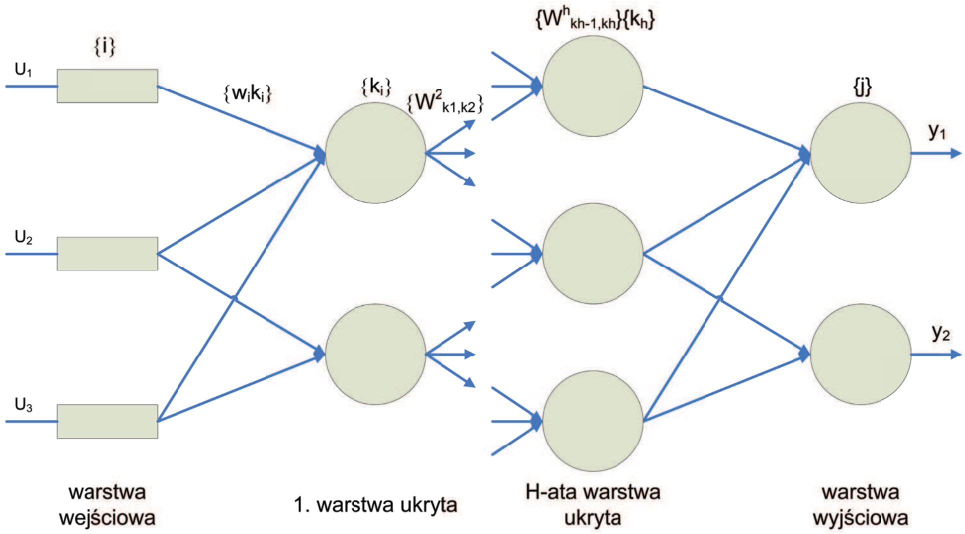
Korzystając z braku połączeń pomiędzy elementami wyjściowymi y_j, $j = 1, 2, \ldots, n$, podobnie jak w przypadku perceptronu prostego, rozważymy przypadek sieci z jednym wyjściem (rys. 5.). Jest to perceptron dwuwarstwowy [3], opisany następująco:

$$y = \prod(v^T u' - \theta_0) = \prod\left(\sum_{k=1}^{k_1} v_k u'_k - \theta_0\right) \tag{10}$$

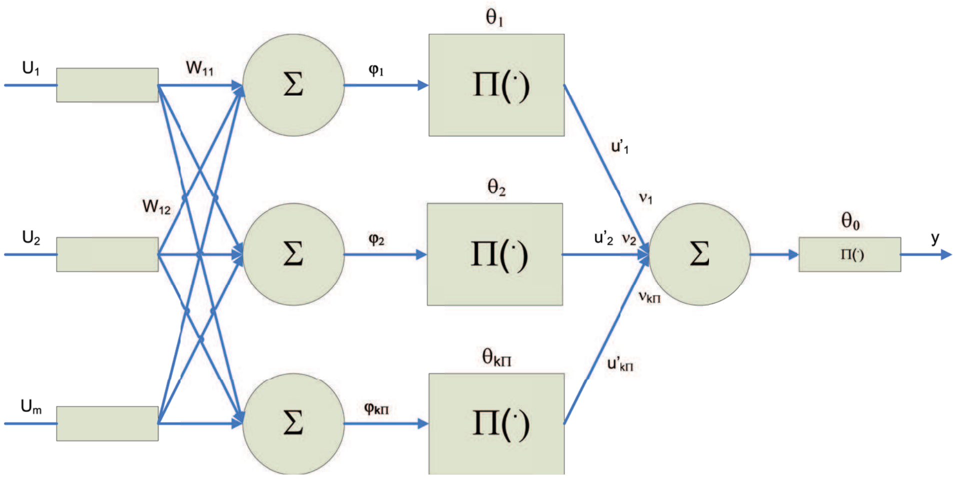
oraz:

$$u'_k = (w_k^T u - \theta_k) = \left(\sum_{i=1}^{T} w_{k_i} u_i - \theta_k \right), \quad k = 1, 2, \ldots, k_1 \tag{11}$$

przy czym θ_0, θ_1, …, θ_{k1} są progami bloków aktywacji elementów sieci (0 odpowiada elementowi wyjściowemu, a 1, …, k_1 – elementom warstwy ukrytej).



Rys. 4. *Struktura perceptronu wielowarstwowego*



Rys. 5. *Przypadek sieci z jednym wyjściem*

Każdy element warstwy ukrytej, podobnie jak opisany wcześniej pojedynczy element perceptronowy, dzieli przestrzeń stanów wejściowych na dwie półprzestrzenie rozdzielone hiperpłaszczyzną o równaniu:

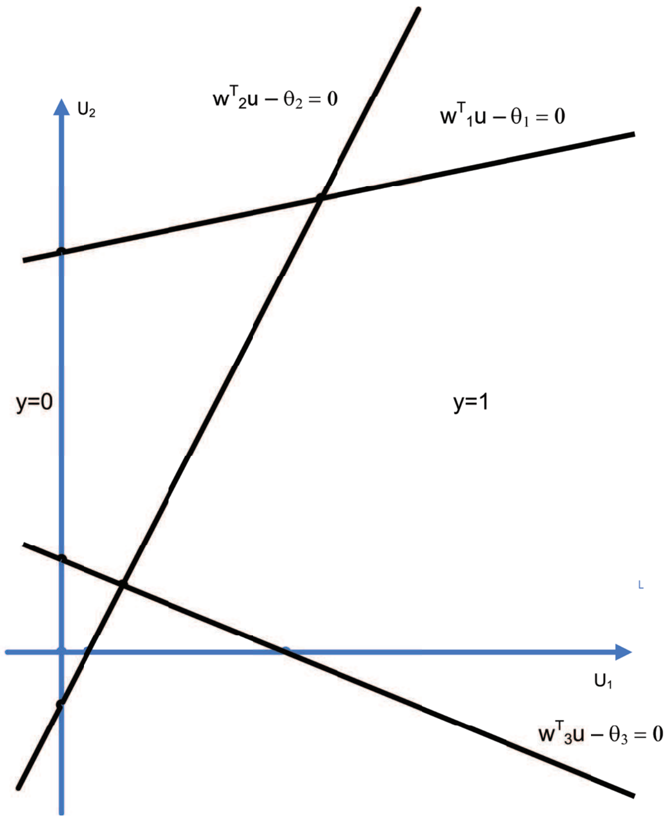
$$w^T u - \theta = 0 \tag{12}$$

Dla wszystkich stanów z jednej półprzestrzeni dany element jest aktywny, dla stanów z drugiej – nie. Element wyjściowy, działając zgodnie ze wzorem:

$$y = \prod(v^T u' - \theta_0) = \prod\left(\sum_{k=1}^{k_1} v_k u'_k - \theta_0\right) \tag{13}$$

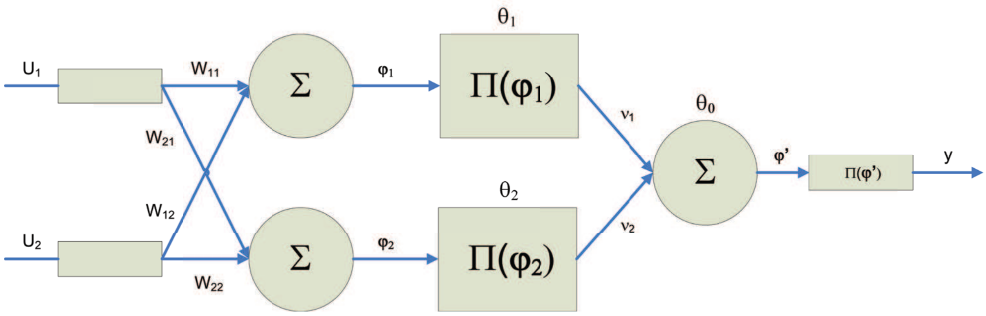
dzieli przestrzeń wektorów wejściowych u na dwie podprzestrzenie uzyskane w wyniku kombinacji iloczynów i sum mnogościowych zbiorów stanów aktywnych poszczególnych elementów warstwy ukrytej. Ponieważ w przypadku perceptronu jednowarstwowego są to półprzestrzenie, jedna z podprzestrzeni stanów aktywności elementu wyjściowego musi być zbiorem wielościennym wypukłym. Wartości wag v połączeń wstępujących do elementu wyjściowego i jego próg θ_0 decydują, które z określonych w warstwie ukrytej hiperpłaszczyzn decyzyjnych ograniczać będą utworzony zbiór wypukły i jaka będzie aktywność wyjściowa dla stanów wewnątrz tego zbioru: $y = 0$, czy też $y = 1$.

Wynika stąd, że jeśli jeden z dwóch zbiorów wzorcowych wektorów wejściowych odpowiadających wartościom funkcji aktywacji 0 i 1 zawiera się w zbiorze wypukłym [3], to dane zagadnienie może być rozwiązane przez perceptron dwuwarstwowy. Na rys. 6. przedstawiono graficzną ilustrację zagadnienia realizowanego przez perceptron dwuwarstwowy na płaszczyźnie.

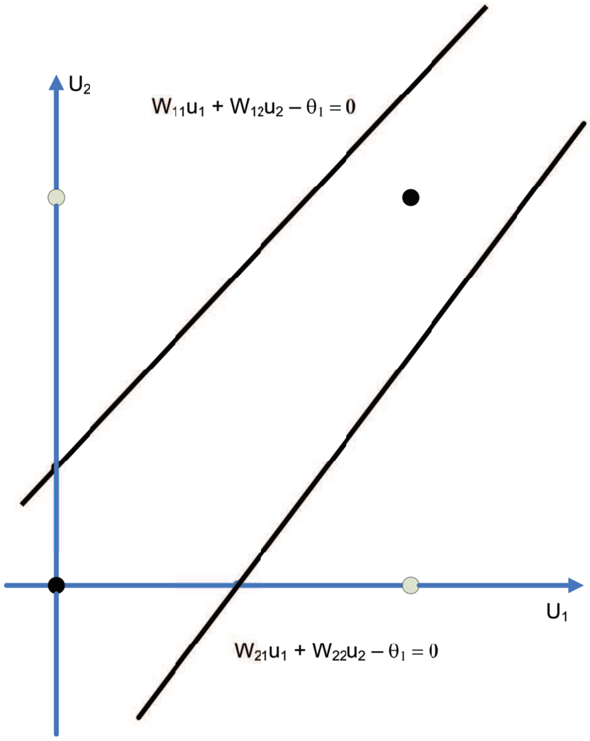


Rys. 6. *Graficzna interpretacja perceptronu dwuwarstwowego*

Korzystając z własności perceptronu wielowarstwowego, rozwiążmy z jego pomocą problem XOR (warunku logicznego). Na rys. 7. i 8. przedstawiono schemat sieci i realizowany przez nią podział wektorów wejściowych:



Rys. 7*. Schemat sieci*



Rys. 8. *Podział wektorów wejściowych*

Dalszym stadium rozbudowy omawianej sieci jest perceptron trójwarstwowy [3], tzn. sieć z dwiema warstwami ukrytymi. Preceptron o takiej strukturze może realizować dowolne odwzorowania przestrzeni stanów wejściowych w przestrzeń stanów wyjściowych. Innymi słowy, perceptron trójwarstwowy gwarantuje istnienie układu wag sieci realizującej poprawnie dowolne odwzorowanie zbioru wzorcowych wektorów wejściowych $\{u^{\mu}\}$ na odpowiadający mu zbiór oczekiwanych wektorów wyjściowych $\{y_{\mu}{}^{z}\}$. Każdy z elementów pierwszej warstwy ukrytej dzieli, podobnie jak elementy wyjściowe w perceptronie prostym, przestrzeń stanów wejściowych na dwie półprzestrzenie rozdzielone pewną hiperpłaszczyzną decyzyjną. Następnie każdy z elementów drugiej warstwy, podobnie jak elementy wyjściowe w perceptronie dwuwarstwowym, dokonuje pewnej transformacji mnogościowej półprzestrzeni z warstwy pierwszej, dzieląc przestrzeń stanów wejściowych na dwie podprzestrzenie, z których jedna stanowi zbiór wielościenny wypukły. Następnie każdy z elementów warstwy wyjściowej dokonuje transformacji mnogościowej powyższych zbiorów wypukłych, w wyniku której można uzyskać zbiór o dowolnej postaci. Jedynym ograniczeniem jest dostatecznie duża liczba elementów przetwarzających w sieci i połączeń między nimi.

4. Podsumowanie i wnioski końcowe

Bezpieczeństwo systemów zaopatrzenia w wodę zależy od sprawnego i wydajnego systemu nadzoru i kontroli [1].

Budowa systemu nadzoru i kontroli w oparciu o rozbudowane modele neuronowe pozwala na zwiększenie ich efektywności i w wielu przypadkach na wyeliminowanie człowieka jako najsłabszego ogniwa procesu decyzyjnego w stanach zagrożeń.

Zastosowanie sieci neuronowych opartych na teoriach Rosenblatta pozwala na zbudowanie bezpiecznego systemu zarządczego, który w razie własnej autoktonicznej awarii nadal może pracować i funkcjonować prawidłowo.

PIŚMIENNICTWO

1. S. Biedugnis, P. Podwójci: Studium na temat danych wejściowych modeli optymalizacyjnych regionalnej gospodarki odpadami komunalnymi. VI Sem. Inst. ZWiBW PW. Oficyna Wyd. PW, Warszawa 1999, s. 177–189.
2. J. Jóźwiak: Statystyka od podstaw. Polskie Wyd. Ekonomiczne, Warszawa 1997.
3. R. A. Koniński: Sztuczne sieci neuronowe. WNT, Warszawa 2002.
4. D. H. Marks, J. C. Liebman: Mathematical analysis of solid waste collection. Bureau of Solid Waste Management, 1970.
5. J. McCutcheon, W. Scott: An Introduction to the Mathematics of Finance, Butterworth-Heinemann, 1991.
6. D. Morrison: Wielowymiarowa analiza statystyczna. PWN, Warszawa 1990.
7. S. Osowski: Sieci neuronowe do przetwarzania informacji. WFUJ, Kraków 2002.
8. K. A. Shuster: A Five-Stage Improvement Process for Solid Waste Collection Systems, US EPA, Washington D.C., 1974.
9. M. Siudak: Badania operacyjne. Oficyna Wyd. PW, Warszawa 1997.

10. R. Staniszewski: Teoria systemów. Wszechnica Polskiej Akademii Nauk, Ossolineum, Wrocław 1988.
11. J. Stevens: Applied multivariate statistics for the social science. Hillsdale, NJ Erlbaum, 1986.
12. R. Tadeusiewicz: Wprowadzenie do sieci neuronowych. Kraków 2001.
13. D. Witkowska: Sztuczne sieci neuronowe i metody statystyczne. C. H. Beck, Warszawa 2002.

SUMMARY

prof. dr hab. inż. Stanisław BIEDUGNIS
dr inż. Mariusz SMOLARKIEWICZ
dr Marcin M. SMOLARKIEWICZ

NEURAL NETWORKS IN THE SAFETY OF WATER SUPPLY ISSUE

In this article the use of neural network concept, proposed by Rosenblatt in 1958, in on-line modeling of water supply systems was presented. This partly electro-mechanical and partly electronic system was invented to recognize alphanumerical symbols with self-learning procedure. It was the first constructed neural network which could work even if part of the network was broken. The use of neural network of Rosenblatt type in modeling process of water supply systems increases the level of safety and efficiency of modeling itself.