



Bezpieczeństwo zaopatrzenia w wodę

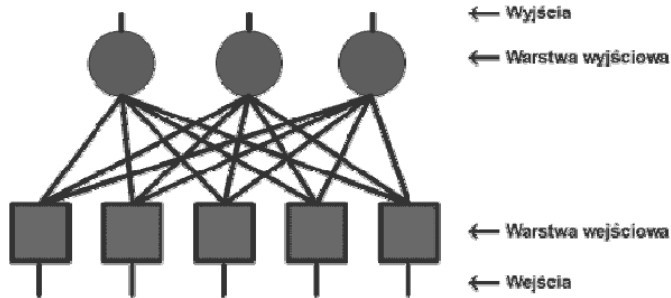
Stanisław Biedugnis, Mariusz Smolarkiewicz
Szkoła Główna Służby Pożarniczej

Andrzej Czapczuk
Politechnika Warszawska

1. Wstęp

W roku 1958 Rosenblatt [8] 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. 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).



Rys. 1. Perceptron

Fig. 1. Perceptron

Zgodnie z przyjętą zasadą neurony warstwy wyjściowej otrzymują sygnały od neuronów warstwy wejściowej, lecz nie odwrotnie. Oprócz tego neurony z danej warstwy nie komunikują się między sobą.

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:

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

lub bipolarną:

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

gdzie:

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

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.
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.

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. 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 progów $\theta = 0$. Jego rolę przejmuje waga wstępującego do niego połączenia od dodatkowego elementu wejściowego (tzw. elementu progowego, ang. bias) o wartości u_0 zawsze równej 1.4. Wówczas dla $w_0 = -\theta$ wzór:

$$\varphi = \sum_{i=1}^m w_i u_i - \theta \quad (4)$$

przyjmuje postać:

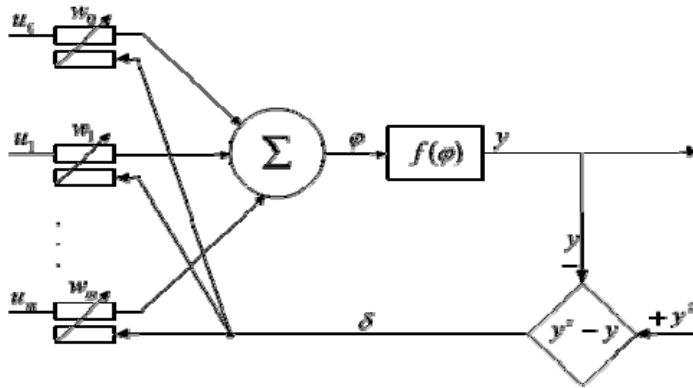
$$\varphi = \sum_{i=0}^m w_i u_i \quad (5)$$

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 osobna sieć o $m+1$ wejściach i jednym wyjściu (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 .



Rys. 2. Element sieci – proces uczenia nadzorowanego
Fig. 2. Element of network – supervised teaching process

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. Istnieją trzy możliwości:

1. Prezentować dany wzorec 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:

$$\Gamma = \{(\mathbf{u}^\mu, y_\mu^z)\}_{\mu=1}^L \quad (6)$$

z prawdopodobieństwem $1/L$.

Pierwsza z metod jest niewskazana, 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:

$$w_i(k+1) = w_i(k) + \eta \delta_\mu u_i^\mu \quad (7)$$

gdzie:

$$\delta_\mu = y_\mu^z - y_\mu = \begin{cases} -1, & y_\mu^z = 0 \wedge y_\mu = 1 \\ 0, & y_\mu^z = y_\mu \\ 1, & y_\mu^z = 1 \wedge y_\mu = 0 \end{cases} \quad (8)$$

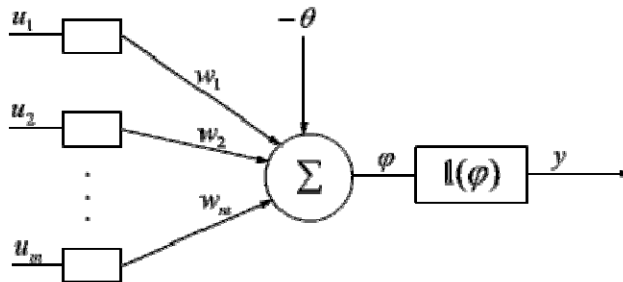
przy czym k oznacza numer kroku algorytmu, $w_i(k)$ – wagę i -tego połączenia na k -tym kroku, u_i^μ – i -tą składową μ -tego wektora wejściowego u^μ , y_μ^z – żadaną odpowiedź na μ -ty wektor wejściowy u^μ , y_μ – aktualną odpowiedź na μ -ty wektor wejściowy u^μ , η – parametr uczenia. Dla każdego kroku k losuje się odpowiadający mu numer wzorca μ . Powyższa zasada nosi nazwę reguły delty. Łatwo zauważyć, że proces uczenia to iteracyjny proces minimalizacji błędów $\delta_{i\mu}$, $\mu = 1, \dots, L$.

Jedna z interesujących właściwości perceptronu została sformułowana przez Rosenblatta [12] 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 [12] 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):



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

Fig. 3. Element with m inputs and activation threshold θ

Odpowiedź y tego elementu wyraża się wzorem:

$$l(\varphi) = \begin{cases} 1, & \varphi > 0 \\ 0, & \varphi \leq 0 \end{cases} \quad (9)$$

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

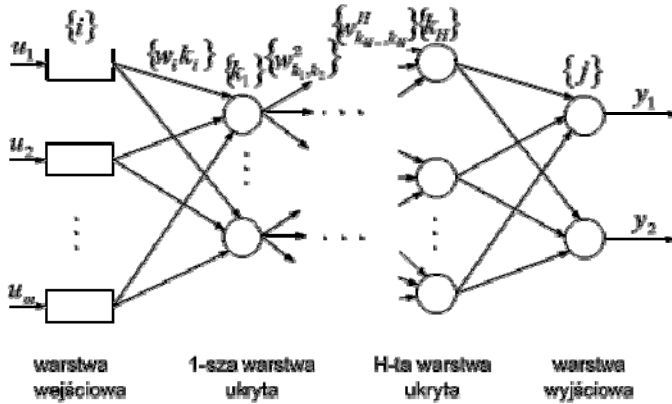
$$\mathbf{w}^T \mathbf{u} - \theta = 0 \quad (10)$$

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

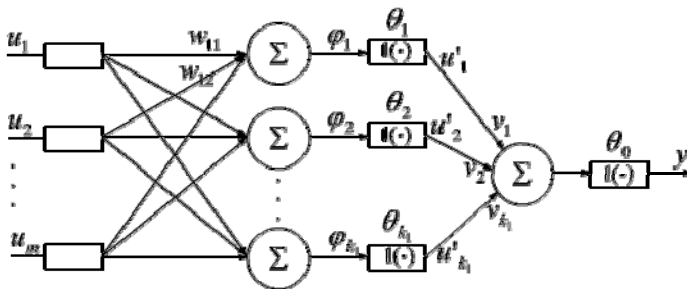
Ograniczenie liniowej separowalności perceptronu prostego można usunąć poprzez wprowadzenie warstw ukrytych. Oto struktura perceptronu wielowarstwowego (rys. 4). Przy czym $i = 1, 2, \dots, m$; m – liczba elementów w warstwie wejściowej; $j = 1, 2, \dots, n$; n – liczba elementów

w warstwie wyjściowej; $h = 1, 2, \dots, H$; H – liczba warstw ukrytych; $k_h = 1, 2, \dots, K_H$; K_H – liczba elementów w h -tej warstwie ukrytej; w_{k_h-1, k_h}^h , k_h – waga połączenia pomiędzy elementami k_{h-1} -tym oraz k_h -tym odpowiednio zawartymi w warstwach $(h-1)$ i h .



Rys. 4. Struktura perceptronu wielowarstwowego
 Fig. 4. Structure of multi layer perceptron

Korzystając z braku połączeń pomiędzy elementami wyjściowymi y_j , $j = 1, 2, \dots, n$, podobnie jak w przypadku perceptronu prostego, rozważymy przypadek sieci z jednym wyjściem (rys. 5).



Rys. 5. Przypadek sieci z jednym wyjściem
 Fig. 5. Network with one output

Jest to perceptron dwuwarstwowy opisany następująco:

$$y = \mathbf{l}(\mathbf{v}^T \mathbf{u}' - \theta_0) = \mathbf{l}\left(\sum_{k=1}^h v_k u'_k - \theta_0\right) \tag{11}$$

oraz:

$$u'_k = (\mathbf{w}_k^T \mathbf{u} - \theta_k) = \left(\sum_{i=1}^{n_i} w_{ki} u_i - \theta_k \right), \quad k = 1, 2, \dots, k_1 \quad (12)$$

przy czym $\theta_0, \theta_1, \dots, \theta_{k_1}$ są progami bloków aktywacji elementów sieci (0 odpowiada elementowi wyjściowemu, a $1, \dots, k_1$ – elementom warstwy ukrytej).

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

$$\mathbf{w}^T \mathbf{u} - \theta = 0 \quad (13)$$

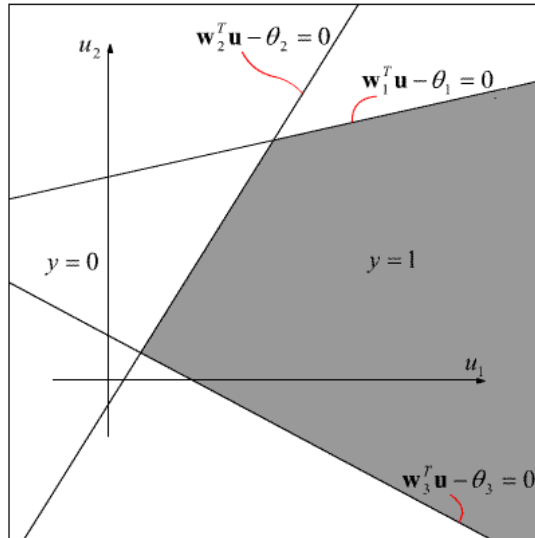
Dla wszystkich stanów z jednej półprzestrzeni dany element jest aktywny, dla stanów z drugiej – nie. Element wyjściowy, działając zgodnie ze wzorem:

$$y = \mathbf{l}(\mathbf{v}^T \mathbf{u}' - \theta_0) = \mathbf{l}\left(\sum_{k=1}^{k_1} v_k u'_k - \theta_0\right) \quad (14)$$

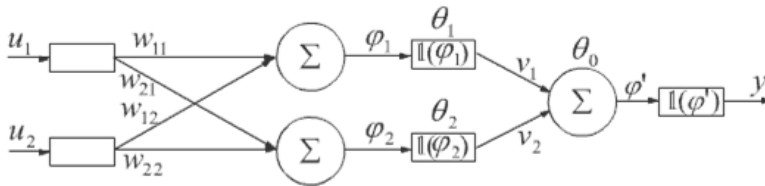
dzieli przestrzeń wektorów wejściowych \mathbf{u} na dwie podprzestrzenie uzyskane w wyniku kombinacji iloczynów i sum mnogościowych zbiorów stanów aktywnych poszczególnych elementów warstwy ukrytej. Ponieważ w przypadku perceptronu jednowarstwowego są to półprzestrzenie, jedna z podprzestrzeni stanów aktywności elementu wyjściowego musi być zbiorem wielościnnym 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, to dane zagadnienie może być rozwiązane przez perceptron dwuwarstwowy. Na rys. 6 przedstawiono graficzną ilustrację zagadnienia realizowanego przez perceptron dwuwarstwowy na płaszczyźnie.

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.



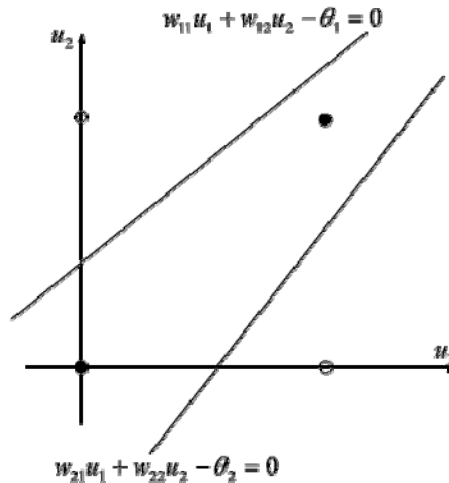
Rys. 6. Graficzna interpretacja perceptrona dwuwarstwowego
Fig. 6. Graphical interpretation of two layer perceptron



Rys. 7. Schemat sieci
Fig. 7. Network diagram

Dalszym stadium rozbudowy omawianej sieci jest perceptron trójwarstwowy, tzn. sieć z dwiema warstwami ukrytymi. Perceptron 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 rozdzielona 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 jedną stanowi zbiór wielościenne 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.



Rys. 8. Podział wektorów wejściowych

Fig. 8. Division of input vectors

4. Podsumowanie i wnioski końcowe

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

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 o teorie Rosenblatt pozwala na zbudowanie bezpiecznego systemu zarządczego, który w razie własnej autoktonicznej awarii nadal może pracować i funkcjonować prawidłowo.

Literatura

1. **Biedugnis S., Podwójci P.:** *Studium na temat danych wejściowych modeli optymalizacyjnych regionalnej gospodarki odpadami komunalnymi*. VI Sem. Inst. ZWiBW PW, Oficyna Wyd. PW, Warszawa, 177-189, 1999.
2. **Gomółka B.:** *Podstawy ochrony środowiska*. Wyd. Politechniki Wrocławskiej, 1980.
3. **Harrison E., Angell R.:** *Waste Prevention Tool Kit for Local Governments*. Cornell Waste Management Institute, 1992.
4. **Hicman H.:** *Collection of Residential Solid Waste*. In: *The Solid Waste Handbook: A Practical Guide*, ed. by Robinson W., John Wiley & Sons, NY, 1986.
5. **Jóźwiak J.:** *Statystyka od podstaw*. Polskie Wyd. Ekonomiczne, Warszawa, 1997.
6. **Koniński R.A.:** *Sztuczne sieci neuronowe*. WNT, Warszawa, 2002.
7. **Marks D.H., Liebman J.C.:** *Mathematical analysis of solid waste collection*, Bureau of Solid Waste Management, 1970.
8. **McCutcheon J., Scott W.:** *An Introduction to the Mathematics of Finance*. Butterworth-Heinemann, 1991.
9. **Morrison D.:** *Wielowymiarowa analiza statystyczna*. PWN, Warszawa, 1990.
10. **Oowski S.:** *Sieci neuronowe do przetwarzania informacji*. WFUJ, Kraków 2002.
11. **Podwójci P., Biedugnis S.:** *Studium na temat danych wejściowych modeli optymalizacyjnych regionalnej gospodarki odpadami komunalnymi*. Grant KBN nr 3 T09C 052 15, Warszawa, 2000.
12. **Shuster K.A.:** *A Five-Stage Improvement Process for Solid Waste Collection Systems*. US EPA, Washington D.C., 1974.
13. **Siudak M.:** *Badania operacyjne*. Oficyna Wydawnicza PW, Warszawa, 1997.
14. **Staniszewski R.:** *Teoria systemów*. Wszechnica Polskiej Akademii Nauk, Ossolineum, Wrocław, 1988.
15. **StatSoft Polska:** *Statistica Pl dla Windows*. Kraków, 1997.
16. **Stevens J.:** *Applied multivariate statistics for the social sciences*. Hillsdale, NJ Erlbaum, 1986.
17. **Szajna J.:** *Turbo Prolog. Programowanie w języku logiki*. WNT, Warszawa, 1991.
18. **Tadeusiewicz R.:** *Wprowadzenie do sieci neuronowych*. Kraków, 2001.
19. **Witkowska D.:** *Sztuczne sieci neuronowe i metody statystyczne*. C.H.Beck, Warszawa, 2002.
20. **Wiśniewski M.:** *Elementy matematyki finansowej*. maszynopis, Wydział Nauk Ekonomicznych, Uniwersytet Warszawski, Warszawa, 1999.
21. **Zeigler B.:** *Teoria modelowania i symulacji*, PWN, Warszawa, 1984.

22. <http://www.lpa.co.uk> – strona firmy LPA.
23. <http://www.hallogram.com/science/lpaprolog/index.html> – strona firmy Hallogram.

Safety of Water Supply

Abstract

In 1958 Rosenblatt developed and built an artificial neural network, called perceptron. This partly electro-mechanical and partly electronic system was intended to recognize the alphanumeric signs with learning process as a method of programming the system. An important result of Rosenblatt was also to proving of so called theorem of perceptron convergence, which guarantees a finite number of iterations of the learning process, if for the modeled problem using this type of network there is a optimal arrangement of weights. Although the operation of perceptron was not satisfactory from the point of view of the fundamental goal (the system showed high sensitivity to change of scale of recognized objects and their location in the field of view), it was a great success of research in this area. First of all, it was the first physically constructed system of simulating neural network, which together with the ability to learn demonstrated ability to work properly even after the damage of part of its components. Using this type of neural networks in on-line modeling of water supply networks helps to increase safety in relation to their proper modeling, the effectiveness of on-line process and its reliability.

In its simplest form perceptron was constructed of two separate layers of neurons representing the input and output, respectively (Fig. 1). According to the assumed rule, neurons of output layer receive signals from neurons of input layer, but not vice versa. In addition, neurons of given layer do not communicate between themselves.

In 1969 Minsky and Papert noticed that a lot of interesting functions cannot be modeled by a simple perceptron, because condition of existence of the weights vector from Rosenblatt theorem is not met. Limitation of linear separability of perceptron can be removed simply by introducing hidden layers. Structure of multilayer perceptron is shown in Fig. 4.

Safety of water supply systems depends on efficient system of supervision and control. Construction of supervision and control system based on advanced neural models allows you to increase their effectiveness and in many cases to eliminate man as the weakest link in decision-making process during emergencies.

Application of neural networks based on the theories of Rosenblatt allows to build a safe management system which, in the event of their auto failure can still work and function properly.