

Sebastian KURLETO, Kazimierz TRYBALSKI
Akademia Górniczo-Hutnicza, Kraków

MODELOWANIE UKŁADU MIELENIA I KLASYFIKACJI RUDY MIEDZI Z ZASTOSOWANIEM SIECI NEURONOWYCH

Streszczenie. W publikacji przedstawiono próbę zastosowania sieci neuronowych – jednej z uniwersalnych metod – do modelowania złożonego procesu mielenia i klasyfikacji. Rozpatrywane procesy wchodzą w skład węzła przygotowania nadawy w Zakładach Wzbogacania Rudy „Rudna” KGHM „Polska Miedź” SA. W badaniach zastosowano sieć neuronową uczoną metodą wstecznej propagacji błędów.

THE MODELLING OF THE COPPER ORES GRINDING SYSTEM AND CLASSIFICATION BY USING NEURAL NETWORKS

Summary. The attempt of neural networks application (one of the universal methods) – for the complex grinding and classification process modelling is presented in the paper. The considered processes are the part of the centre of the feed preparation in the ZWR „Rudna” KGHM „Polska Miedź” SA. In the researches the neural network, learned by the method of the backward error propagation is used.

1. Wstęp

Technologiczne procesy przeróbcze mają różnorodny charakter. Różne są zasady ich przebiegu, zachodzą one według różnych praw fizycznych, fizykochemicznych i chemicznych. Różne są też ich zadania i wskaźniki oceny, np.:

1. rozdrabnianie - oceniane jest składem ziarnowym produktu,
2. wzbogacanie - zawartością składnika użytecznego w uzyskiwanych produktach,
3. odwadnianie - gęstością produktu.

Złożoność i różnorodność procesów i schematów technologicznych przeróbki surowców mineralnych są przyczyną trudności ich analizy i oceny. Analiza i modelowanie procesów

różnorodnych, jakie występują w przeróbce kopalin, najefektywniejsze są, a często jedynie możliwe - przy zastosowaniu uniwersalnych metod opisu.

W niniejszej publikacji zostanie przedstawiona próba zastosowania sieci neuronowych do modelowania złożonego procesu mielenia i klasyfikacji.

W modelowaniu procesów przeróbczych sieci neuronowe znalazły zastosowanie w latach dziewięćdziesiątych (Stęgowski 1997 r., Cubillos 1996 r., Watano 1997 r.).

2. Wprowadzenie

Sieci neuronowe są strukturami wzorowanymi na biologicznych systemach nerwowych.

Jednostkowe elementy sieci neuronowych, nazywane neuronami, są odpowiednikami neuronów biologicznych [4]. Neurony te powiązane są połączeniami stanowiącymi analogię do aksonów i dendrytów w systemach biologicznych. Połączenia pomiędzy neuronami mogą mieć charakter pobudzający lub hamujący, a w procesie uczenia efektywność tych połączeń może ulegać zmianie, tak aby dana sieć realizowała określone zadania.

Sieć neuronowa jest równoległym systemem złożonym z pojedynczych neuronów, które można traktować jako elementarne procesory. Każdy procesor wykonuje pewien prosty program, na który składa się obliczanie sumy ważonej danych wejściowych i niekiedy dodanie pewnego dodatkowego stałego składnika. Sieci neuronowe są nowym i pod wieloma względami atrakcyjnym narzędziem informatycznym, jednocześnie wybitnie wygodnym i wydajnym.

Zalety sieci neuronowych to dwa podstawowe atuty:

1. stanowią one wygodną i tanią możliwość wykorzystania wieloprocessorowego systemu o bardzo wielu elementach przetwarzających równolegle dostarczane informacje,
2. nie wymagają programowania, wykorzystują natomiast proces uczenia.

Sieci znalazły skuteczne zastosowanie głównie dzięki zdolnościom przystosowania się do modelowanych nieliniowych funkcji wielu zmiennych. Ta cecha pozwala na modelowanie różnych procesów na zasadzie czarnej skrzynki "black box", używanej do korelacji zespołu sygnałów wejściowych i wyjściowych.

Sieci neuronowe są użytecznym narzędziem do kompleksowego modelowania systemów w wielu dziedzinach inżynierii. W literaturze wiele jest doniesień o różnych udanych zastosowaniach techniki sieci neuronowych, jest to więc technika bardzo obiecująca. Podejmowane są badania i próby rozwiązywania problemów naukowych i praktycznych przez

stosowanie tej techniki. Niekiedy osiągane są znaczące sukcesy, często jednak zdarza się, że obiecująco zapowiadające się badania nie przynoszą oczekiwanych rezultatów.

Jak już wspomniano, poszczególne neurony występujące w sieci są połączone pomiędzy sobą, efektywność modelowania zależy od przyjętych wartości modyfikowalnych wag. Waga jest współczynnikiem liczbowym mogącym przyjmować zarówno wartości dodatnie dla połączeń pobudzających, jak i ujemne dla połączeń hamujących. Wartość sygnału wyjściowego z neuronu obliczana jest w dwu etapach.

W pierwszym etapie sygnały wejściowe przemnażane są przez odpowiadające im wagi i następnie sumowane. Etap ten nazywany jest funkcją wewnętrznego przetwarzania i w praktyce jest to najczęściej funkcja sumowania, można jednak realizować także inne dowolne funkcje, takie jak: iloczyn, maksimum, minimum, itp.

W drugim etapie wynik funkcji wewnętrznego przetwarzania podlega działaniu określonej funkcji wejścia – wyjścia, zwanej funkcją aktywacji. Ostatecznie dla pojedynczego neuronu sygnał wyjściowy wyliczany jest z następującej zależności:

$$y = f\left(\sum_{i=1}^n w_i x_i\right) \quad (1)$$

gdzie:

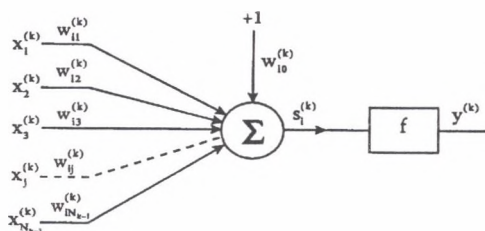
- x_i – wektor danych wejściowych,
- w_i – wektor wag,
- f – funkcja aktywacji,
- y – sygnał wyjściowy,
- n – liczba sygnałów wejściowych.

Na rysunku 1 przedstawiono schemat pojedynczego neuronu. Najczęściej stosowane w praktyce funkcje aktywacji to:

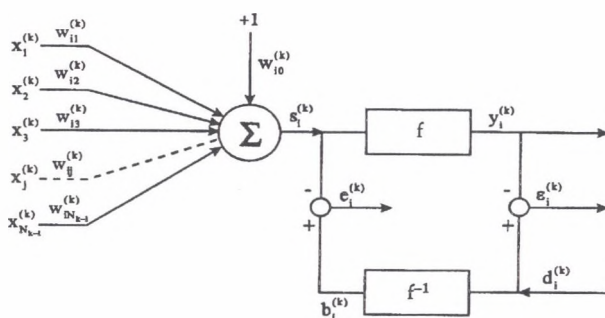
- funkcja liniowa,
- funkcja skoku jednostkowego,
- funkcja sigmoidalna,
- funkcja tangens hiperboliczny.

Na rysunku 2 przedstawiono model neuronu, stosowany w często wykorzystywanym w modelowaniu, algorytmie RLS (z ang.: Recursive Least Squares - metody najmniejszych kwadratów). Program obliczeniowy, który wykorzystano w tym przypadku, bazuje natomiast na algorytmie LMS (z ang.: Least Mean Squares) - z którego wywodzi się metoda wstecznej

propagacji błędów [2]. Opis metody wstecznej propagacji błędów zastosowanej do uczenia sieci neuronowych w niniejszej publikacji przedstawiono w rozdziale 5.



Rys. 1. Schemat pojedynczego neuronu
Fig. 1. The scheme of a single neuron



Rys. 2. Schemat pojedynczego neuronu wykorzystywany w algorytmie RLS
Fig. 2. The scheme of a single neuron, used in algorithm RLS

Poszczególne symbole umieszczone na rysunku 2 oznaczają:

$x^{(k)} = [x_0^{(k)}, \dots, x_{N_{k-1}}^{(k)}]^T$ – wektor sygnałów wejściowych warstwy k , $k = 1, \dots, L$,

$w_{ij}^{(k)}$ – waga i -tego neuronu, $i = 1, \dots, N_k$, warstwy k , $k = 1, \dots, L$, łącząca go z j -tym wejściem $x_j^{(k)}$, $j = 0, \dots, N_{k-1}$,

$s_i^{(k)}(n) = \sum_{j=0}^{N_{k-1}} w_{ij}^{(k)}(n) x_j^{(k)}(n)$ – sygnał wyjściowy części liniowej i -tego neuronu, $i = 1, \dots, N_k$,

warstwy k , $k = 1, \dots, L$,

$b_i^{(k)}$ – sygnał wzorcowy części liniowej i -tego neuronu, $i = 1, \dots, N_k$, warstwy k , $k = 1, \dots, L$,

$b_i^{(k)} = f^{-1}(d_i^{(k)})$,

$d_i^{(k)}$ – sygnał wzorcowy i -tego neuronu, $i = 1, \dots, N_k$, warstwy k , $k = 1, \dots, L$,

$e_i^{(k)}(n) = b_i^{(k)}(n) - f^{-1}(y_i^{(k)}(n))$ – błąd części liniowej i -tego neuronu, $i = 1, \dots, N_k$, warstwy k , $k = 1, \dots, L$,

$\varepsilon_i^{(k)}(n) = d_i^{(k)}(n) - y_i^{(k)}(n)$ – błąd sygnału wyjściowego i -tego neuronu, $i = 1, \dots, N_k$,

warstwy k , $k = 1, \dots, L$,

$y_i^{(k)}(n) = f(s_i^{(k)}(n))$ – funkcja wyjściowa neuronu,

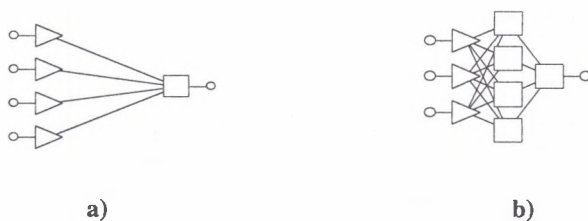
L – liczba warstw sieci,

N_k – liczba neuronów w warstwie k , $k = 1, \dots, L$,

N_0 – liczba sygnałów wejściowych sieci.

W praktyce najczęściej stosowane są sieci wielowarstwowe. W takiej sieci rozróżniamy warstwę wejściową, warstwy ukryte i warstwę wyjściową.

Na rysunkach 3 przedstawiono przykładowe schematy sieci liniowej oraz sieci typu MLP (z ang.: Multilayer Perception – perceptron wielowarstwowy).



Rys. 3. Schematy sieci: a) liniowej, b) MLP

Fig. 3. The schemes of a network: a) linear, b) MLP

3. Opis obiektu modelowania

Obiektem modelowania był węzeł mielenia i klasyfikacji rudy miedzi w ZWR „Rudna”. Na rysunku 4 przedstawiono fragment schematu IV ciągu technologicznego w tym zakładzie, przerywaną linią zaznaczono badany układ mielenia i klasyfikacji [3].

Ruda po wstępnym rozdrobnieniu podawana jest przenośnikiem PT60 wraz z wodą do młyna prętowego MP62. Po wstępnym zmieleniu ruda kierowana jest do klasyfikatora zwojowego KZ6A6B, z którego wylew transportowany jest do młyna kulowego MK61, a następnie poprzez rzapie RZ6B trafia na baterię hydrocyklonów HC691, HC692. Wylew hydrocyklonów HC691, HC692 kierowany jest poprzez rzapie do domielania w młynach cylpepsowych MC02, MC04, natomiast przelew tych hydrocyklonów wraz z domielonym wylewem (produkt z MC02 i MC04) podawany jest również poprzez rzapie do flotacji wstępnej MF45, 46, 47, 48.

Przelew klasyfikatora KZ6A6B przesyłany jest poprzez rzapie RZ6A na baterię hydrocyklonów HC695. Wylew baterii hydrocyklonów HC695 kierowany jest do domielania w młynach cylpepsowych MC06, MC08, z których trafia do flotacji wstępnej piasków MF49, MF50. Przelew HC695 przechodzi poprzez rzapie do flotacji piasków MF27, MF28. Miejsca pobierania próbek zaznaczono na rysunku 4 symbolami „x”. Zestawienie i opis danych pomiarowych przedstawiono w tablicy 1. Otrzymało je w następujący sposób [3]:

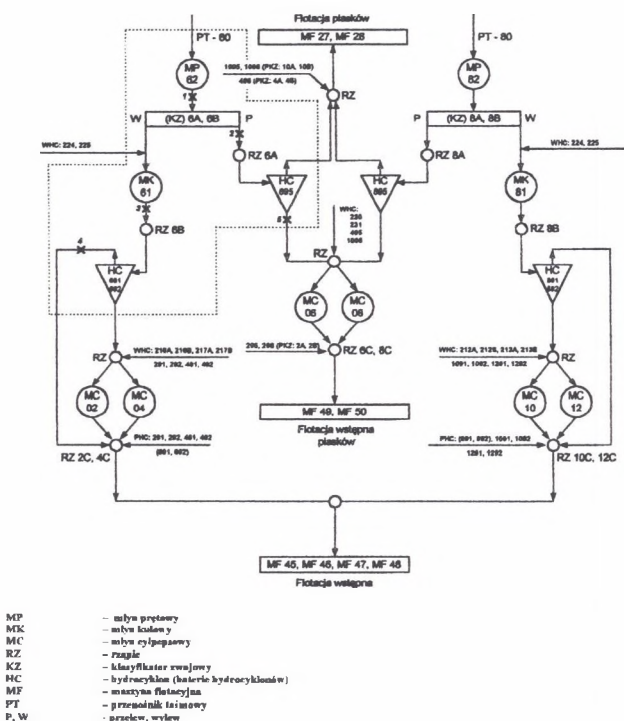
Dane nr. 1 – 4 – przez bezpośrednie określenie masy 1 dm³ pobranej ręcznie próbki,

Dane nr. 5 – 7 – jako odczyt wyniku pomiaru analizatora składu ziarnowego PSM,

Dane nr. 8 – 12 – jako odczyt z automatycznych urządzeń pomiarowych,

Dane nr. 13 – 27 – określono dla pobranych ręcznie próbek, w trakcie analizy sitowej, przeprowadzonej w DKJ Zakładu,

Dane nr. 40 – 54 – określono dla klas ziarnowych pobranych ręcznie próbek, metodą analizy chemicznej przeprowadzonej przez DKJ.



Rys.4. Fragment schematu IV ciągu technologicznego ZWR „Rudna”

Fig.4. The fragment of the production process ZWR „Rudna” scheme

Tablica I

Opis zebranych danych pomiarowych

Nr	Zmienna	Symbol [Wymiar]
1	Gęstość wylewu młyna prętowego MP62	D_{MP} [g/dm ³]
2	Gęstość przelewu klasyfikatora 6A6B	D_{KZ} [g/dm ³]
3	Gęstość wylewu młyna kulowego MK61	D_{MK} [g/dm ³]
4	Gęstość wylewu hydrocyklonu HC695	D_{WHC} [g/dm ³]
5	Gęstość przelewu hydrocyklonu HC691/692 (PSM)	D_{PHC} [g/dm ³]
6	Wychód klasy – 0,071 w przelewie HC691/692 (PSM)	$X_{-0.071 HC (PSM)}$ [%]
7	Wychód klasy + 0,1 w przelewie HC691/692 (PSM)	$X_{+0.1 HC (PSM)}$ [%]
8	Natężenie przepływu rudy MP62	F_{MP} [Mg/h]
9	Ciśnienie nadawy HC691/692	P_{HC} [%]
10	Natężenie przepływu wody do KZ6A6B	F_{WKZ} [%]
11	Natężenie przepływu wody do MP	F_{WMP} [%]
12	Natężenie przepływu wody do RZ6B	F_{WRZ} [%]
13	Wychód klasy + 0,1 w wylewie młyna prętowego MP62	$X_{+0.1MP}$ [%]
14	Wychód klasy + 0,071 w wylewie młyna prętowego MP62	$X_{+0.071MP}$ [%]
15	Wychód klasy – 0,071 w wylewie młyna prętowego MP62	$X_{-0.071MP}$ [%]
16	Wychód klasy + 0,1 w przelewie klasyfikatora 6A6B	$X_{+0.1KZ}$ [%]
17	Wychód klasy + 0,071 w przelewie klasyfikatora 6A6B	$X_{+0.071KZ}$ [%]
18	Wychód klasy – 0,071 w przelewie klasyfikatora 6A6B	$X_{-0.071KZ}$ [%]
19	Wychód klasy + 0,1 w wylewie młyna kulowego MK61	$X_{+0.1MK}$ [%]
20	Wychód klasy + 0,071 w wylewie młyna kulowego MK61	$X_{+0.071MK}$ [%]
21	Wychód klasy – 0,071 w wylewie młyna kulowego MK61	$X_{-0.071MK}$ [%]
22	Wychód klasy + 0,1 w wylewie hydrocyklonu HC695	$X_{+0.1HC}$ [%]
23	Wychód klasy + 0,071 w wylewie hydrocyklonu HC695	$X_{+0.071HC}$ [%]
24	Wychód klasy – 0,071 w wylewie hydrocyklonu HC695	$X_{-0.071HC}$ [%]
25	Wychód klasy + 0,1 w przelewie hydrocyklonu HC691/692	$X_{+0.1HC}$ [%]
26	Wychód klasy + 0,071 w przelewie hydrocyklonu HC691/692	$X_{+0.071HC}$ [%]
27	Wychód klasy – 0,071 w przelewie hydrocyklonu HC691/692	$X_{-0.071HC}$ [%]
28	Gęstość wylewu młyna prętowego MP42	D_{MP42} [g/dm ³]
29	Gęstość przelewu klasyfikatora 4A4B	D_{KZ4A4B} [g/dm ³]
30	Gęstość wylewu młyna kulowego MK41	D_{MK41} [g/dm ³]
31	Wychód klasy + 0,1 w wylewie młyna prętowego MP42	$X_{+0.1MP42}$ [%]
32	Wychód klasy + 0,071 w wylewie młyna prętowego MP42	$X_{+0.071MP42}$ [%]
33	Wychód klasy – 0,071 w wylewie młyna prętowego MP42	$X_{-0.071MP42}$ [%]
34	Wychód klasy + 0,1 w przelewie klasyfikatora 4A4B	$X_{+0.1MPKZ4A4B}$ [%]
35	Wychód klasy + 0,071 w przelewie klasyfikatora 4A4B	$X_{+0.071KZ4A4B}$ [%]
36	Wychód klasy – 0,071 w przelewie klasyfikatora 4A4B	$X_{-0.071KZ4A4B}$ [%]
37	Wychód klasy + 0,1 w wylewie młyna kulowego MK41	$X_{+0.1MK41}$ [%]
38	Wychód klasy + 0,071 w wylewie młyna kulowego MK41	$X_{+0.071MK41}$ [%]
39	Wychód klasy – 0,071 w wylewie młyna kulowego MK41	$X_{-0.071MK41}$ [%]
40	Zawartość miedzi w klasie + 0,1 wylewu MP62	$Cu_{+0.1MP}$ [%]
41	Zawartość miedzi w klasie + 0,071 wylewu MP62	$Cu_{+0.071MP}$ [%]
42	Zawartość miedzi w klasie – 0,071 wylewu MP62	$Cu_{-0.071MP}$ [%]
43	Zawartość miedzi w klasie + 0,1 wylewu KZ6A6B	$Cu_{+0.1KZ}$ [%]
44	Zawartość miedzi w klasie + 0,071 wylewu KZ6A6B	$Cu_{+0.071KZ}$ [%]
45	Zawartość miedzi w klasie – 0,071 wylewu KZ6A6B	$Cu_{-0.071KZ}$ [%]
46	Zawartość miedzi w klasie + 0,1 wylewu MK61	$Cu_{+0.1MK}$ [%]
47	Zawartość miedzi w klasie + 0,071 wylewu MK61	$Cu_{+0.071MK}$ [%]
48	Zawartość miedzi w klasie – 0,071 wylewu MK61	$Cu_{-0.071MK}$ [%]
49	Zawartość miedzi w klasie + 0,1 wylewu HC695	$Cu_{+0.1WHC}$ [%]
50	Zawartość miedzi w klasie + 0,071 wylewu HC695	$Cu_{+0.071WHC}$ [%]
51	Zawartość miedzi w klasie – 0,071 wylewu HC695	$Cu_{-0.071WHC}$ [%]
52	Zawartość miedzi w klasie + 0,1 wylewu HC691/692	$Cu_{+0.1PHC}$ [%]
53	Zawartość miedzi w klasie + 0,071 wylewu HC691/692	$Cu_{+0.071PHC}$ [%]
54	Zawartość miedzi w klasie – 0,071 wylewu HC691/692	$Cu_{-0.071PHC}$ [%]

Dla analizowanego obiektu – węzła mielenia i klasyfikacji – poszukiwano modelu, którego ogólną postać można przedstawić następująco:

$$X_{-0,071HC} = f(D_{MP}, D_{KZ}, F_{MP}, P_{HC}) \quad (2)$$

Zależność ta oznacza, że wychód klasy $-0,071$ mm w przelewie hydrocyklonu HC691/692 ($X_{-0,071}$) jest funkcją:

- gęstości wylewu młyna prętowego MP62 (D_{MP}),
- gęstości przelewu klasyfikatora 6A6B (D_{KZ}),
- natężenia przepływu rudy młyna prętowego MP62 (F_{MP}),
- ciśnienia nadawy hydrocyklonów HC691/692 (P_{HC}).

Jak z powyższej zależności widać, na tym etapie modelowania wybrano tylko niektóre wielkości zawarte w tablicy 1.

4. Opis programu obliczeniowego

Obliczenia dla zadanej funkcji modelu przeprowadzono wykorzystując komputerowy program do modelowania sieci neuronowych *Statistica Neural Networks* firmy StatSoft [5].

Statistica Neural Networks posiada okno główne oraz pewną liczbę innych okien. Wśród tych okien wyróżnić można *okna dokumentów*, które wyświetlane są w obrębie okna głównego (ich wielkość może być modyfikowana przez użytkownika), oraz *okna dialogowe* (o stałych rozmiarach), które wyświetlane są niezależnie od okna głównego. Okna należące do tej drugiej grupy wyświetlane są zawsze „na pierwszym planie”. Zawsze są one widoczne na tle okna głównego, które jednak również pozostaje dostępne. W górnej części okna głównego znajduje się menu i pasek narzędziowy, zaś w dolnej jego części można znaleźć pole przeznaczone do wyświetlania komunikatów i wskaźnika zaawansowania realizowanych procedur. Środkowa część okna tworzy obszar roboczy charakterystyczny dla programów wyposażonych w interfejs wielodokumentowy, w którym wyświetlane są okna o modyfikowalnych przez użytkownika rozmiarach. Zadania wymienionych elementów interfejsu użytkownika są następujące.

- a) **Menu** pozwala na dostęp do wszystkich funkcji oferowanych przez *Statistica Neural Networks*, wybór większości opcji znajdujących się w menu powoduje wyświetlenie innych okien, w których realizowane jest bardziej precyzyjne sterowanie pracą programu.
- b) **Pasek narzędzi** umożliwia szybki dostęp do niektórych (najpopularniejszych) opcji w menu. Jeden ze znajdujących się na nim przycisków umożliwia dostęp do pomocy kontekstowej, po jego wybraniu wywołany zostaje podręcznik elektroniczny, z którego

można skorzystać w celu wyjaśnienia ewentualnych wątpliwości wiążących się ze wskazanym elementem okna. Na pasku narzędzi znajduje się między innymi szczególnie użyteczny przycisk służący do wywołania *Automatycznego projektanta sieci*.

- c) **Pole komunikatów** służy do wyświetlania tych informacji, które nie mają krytycznego znaczenia (komunikaty o charakterze krytycznym wyświetlane są za pomocą pozostających na pierwszym planie okien informacyjnych).
- d) **Wskaźnik zaawansowania** realizowanych procedur pokazuje stopień zaawansowania wykonywanych czasochłonnych operacji, takich jak algorytmy uczenia. Pozwala to zorientować się, jak długo trzeba czekać na wyniki obliczeń. Wygląd interfejsu użytkownika może być w dużej mierze zdefiniowany przez użytkownika, w zależności od indywidualnych potrzeb, upodobań i preferencji.

Większość elementów w *Statistica Neural Networks* zachowuje się w podobny sposób jak w innych programach działających w środowisku Windows. Jednym ze specjalnych i wartych zauważenia narzędzi jest *arkusz danych*, który posiada specjalne możliwości edycyjne. *Statistica Neural Networks* może być w pełni obsługiwany za pomocą klawiatury, ale też alternatywnie – większość operacji sterowania (nie wymagających wprowadzania danych) może być obsługiwana samą tylko myszką.

Więcej informacji na temat programu, jego działania oraz obsługi można uzyskać przeglądając dołączony do programu szczegółowy przewodnik problemowy [5].

5. Opis procedury obliczeniowej

Jako zbiór danych pomiarowych – przyjętych do obliczeń i określenia zależności (2) – wybrano, z danych zawartych w tablicy 1, zbiór odpowiadający zmiennym: 1, 2, 6, 8, 9 składający się z 72 serii wyników pomiarów. W procedurze obliczeniowej modyfikacji wag i wartości progowych wykorzystuje się zwykle algorytmy uczenia działające w oparciu o zbiory danych uczących. W niniejszym przypadku do uczenia sieci neuronowej zastosowano algorytm *wstecznej propagacji błędów* – który należy do najlepiej poznanych metod służących do uczenia sieci neuronowych. Nazwa algorytmu pochodzi od sposobu obliczania błędów w poszczególnych warstwach. W algorytmie tym najpierw obliczane są błędy w warstwie ostatniej, następnie w poprzedzającej ją i dalej aż do warstwy pierwszej.

Dla wielowarstwowej sieci neuronowej o wyjściach [2]:

$$y_i^{(k)}(n) = f(s_i^{(k)}(n)) \quad (3)$$

minimalizowana jest miara błędu określona następująco:

$$Q(n) = \sum_{i=1}^{N_L} \varepsilon_i^{(L)^2}(n) = \sum_{i=1}^{N_L} (d_i^{(L)}(n) - y_i^{(L)}(n))^2 \quad (4)$$

Dla dowolnej wagi sieci stosuje się regułę najszybszego spadku:

$$w_{ij}^{(k)}(n+1) = w_{ij}^{(k)}(n) + \mu (-\nabla_{ij}^{(k)}(n)) \quad (5)$$

gdzie:

μ - stała,

$$\nabla_{ij}^{(k)}(n) = \frac{\partial Q(n)}{\partial w_{ij}^{(k)}(n)} = \frac{\partial Q(n)}{\partial s_i^{(k)}(n)} \frac{\partial s_i^{(k)}(n)}{\partial w_{ij}^{(k)}(n)} = \frac{\partial Q(n)}{\partial s_i^{(k)}(n)} x_j^{(k)}$$

Oznaczając:

$$\delta_i^{(k)}(n) = -\frac{1}{2} \frac{\partial Q(n)}{\partial s_i^{(k)}(n)} \quad (6)$$

otrzymuje się:

$$\frac{\partial Q(n)}{\partial w_{ij}^{(k)}(n)} = -2\delta_i^{(k)} x_j^{(k)} \quad (7)$$

zatem algorytm (5) przyjmie postać:

$$w_{ij}^{(k)}(n+1) = w_{ij}^{(k)}(n) + 2\mu \delta_i^{(k)} x_j^{(k)} \quad (8)$$

Dla warstwy ostatniej otrzymuje się:

$$\begin{aligned} \delta_i^{(L)}(n) &= -\frac{1}{2} \frac{\partial \sum_{m=1}^{N_L} \varepsilon_m^{(L)^2}(n)}{\partial s_i^{(L)}(n)} = -\frac{1}{2} \frac{\partial \varepsilon_i^{(L)^2}(n)}{\partial s_i^{(L)}(n)} = \\ &= -\frac{1}{2} \frac{\partial (d_i^{(L)}(n) - y_i^{(L)}(n))^2}{\partial s_i^{(L)}(n)} = \varepsilon_i^{(L)}(n) \frac{\partial y_i^{(L)}(n)}{\partial s_i^{(L)}(n)} = \varepsilon_i^{(L)}(n) f'(s_i^{(L)}(n)) \end{aligned} \quad (9)$$

Dla dowolnej innej warstwy:

$$\begin{aligned} \delta_i^{(k)}(n) &= -\frac{1}{2} \frac{\partial Q(n)}{\partial s_i^{(k)}(n)} = -\frac{1}{2} \sum_{m=i+1}^{N_{k+1}} \frac{\partial Q(n)}{\partial s_m^{(k+1)}(n)} \frac{\partial s_m^{(k+1)}(n)}{\partial s_i^{(k)}(n)} = \\ &= \sum_{m=i+1}^{N_{k+1}} \delta_m^{(k+1)}(n) w_{mi}^{(k+1)}(n) f'(s_i^{(k)}(n)) = f'(s_i^{(k)}(n)) \sum_{m=i+1}^{N_{k+1}} \delta_m^{(k+1)}(n) w_{mi}^{(k+1)}(n) \end{aligned} \quad (10)$$

Oznaczając:

$$\varepsilon_i^{(k)}(n) = \sum_{m=1}^{N_{k+1}} \delta_m^{(k+1)}(n) w_{mi}^{(k+1)}(n) \quad (11)$$

wyrażenie (10) przyjmuje postać:

$$\delta_i^{(k)}(n) = \varepsilon_i^{(k)}(n) f'(s_i^{(k)}(n)) \quad (12)$$

W rezultacie algorytm wstecznej propagacji błędów można zapisać następująco:

$$y_i^{(k)}(n) = f(s_i^{(k)}(n))$$

$$\varepsilon_i^{(k)}(n) = \begin{cases} \varepsilon_i^{(L)}(n) & \text{dla } k = L \\ \sum_{m=1}^{N_{k+1}} \delta_m^{(k+1)}(n) w_{mi}^{(k+1)}(n) & \text{dla } k = 1, \dots, L-1 \end{cases} \quad (13)$$

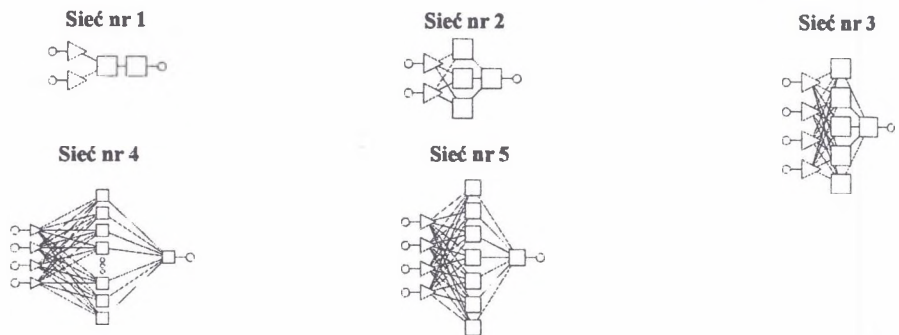
$$\delta_i^{(k)}(n) = \varepsilon_i^{(k)}(n) f'(s_i^{(k)}(n))$$

$$w_{ij}^{(k)}(n+1) = w_{ij}^{(k)}(n) + 2\mu d_i^{(k)} x_j^{(k)}$$

Dla zadanego zbioru danych *Automatyczny projektant sieci* znalazł zbiór składający się z pięciu sieci neuronowych (rysunek 5), które mogą być wykorzystane do modelowania rozpatrywanego obiektu badań. Z sieci tych sieć nr 5 okazała się najlepsza, charakteryzowała się bowiem najmniejszą wartością błędu. Podstawowe charakterystyki rozpatrywanych sieci zawarto w edytorze zbioru sieci, przedstawionym w tablicy 2.

W tablicy 2 w kolumnach 5, 6 oraz 7 przedstawiono wartości błędów zbiorów uczących, walidacyjnych oraz testowych dla poszczególnych sieci. Umieszczono tam błędy otrzymane przy uruchamianiu sieci na podzbiórze uczącym, walidacyjnym i testowym. Są to pierwiastki sum kwadratów błędów poszczególnych przypadków wyznaczanych przez funkcję błędu sieci [5].

Wzrost błędu walidacyjnego na pewnym etapie uczenia sieci wskazuje na przeuczenie sieci. Wartość tego błędu należy przede wszystkim brać pod uwagę przy podejmowaniu decyzji, której sieci użyć w modelowaniu. Błąd uczący bardzo niewiele mówi o zdolności do znajdowania poprawnych wartości wyjściowych dla nowych danych na wejściu. W niniejszym przypadku podzbiór uczący liczył trzydzieści sześć serii danych, natomiast podzbiory walidacyjny i testowy zawierały po osiemnaście serii danych. Podział posiadanego zbioru danych na wymienione kategorie ma istotne znaczenie – w trakcie uczenia i testowania zbiory: walidacyjny i testowy są traktowane inaczej niż zbiór uczący. Zbiór testowy nie jest w



Rys. 5. Schematy sieci neuronowych znalezionych przez program
Fig. 5. The neural networks schemes, found by the program

Tablica 2

Edytor zbioru sieci

1	2	3	4	5	6	7	8	9	10
Nr. sieci	Typ	Wejści a	Ukryte	Bł. ucz.	Bł. wal.	Bł. tst.	Jakość ucz.	Jakość wal.	Jakość tst.
1	MLP	2	1	17,47871	17,21150	17,74191	0,698568	0,636271	0,772872
2	MLP	2	3	9,85576	7,61664	11,67299	0,394081	0,279199	0,514260
3	MLP	4	5	3,77053	4,37903	3,04266	0,150764	0,163236	0,134660
4	MLP	4	10	3,70936	4,25221	3,07205	0,148318	0,158655	0,136237
5	MLP	4	7	3,23638	3,21600	3,25663	0,129406	0,120022	0,144580

Tablica 3

Statystyki zagadnień regresyjnych

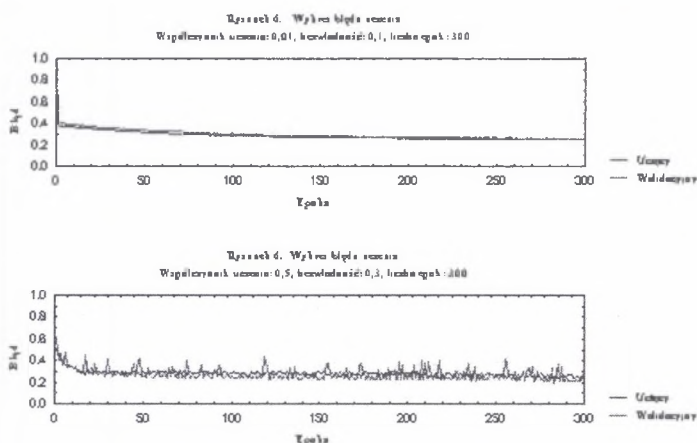
1	2	3	4
	Uc. var 6	Wa. var 6	Te. var 6
Średnia	48,26944	51,92222	44,61667
Odchylenie standardowe	25,3642	27,5603	23,16687
Średni błąd	0,003428	-0,09311	0,0999604
Odchylenie błędu	3,282285	3,307854	3,349469
Średni błąd bezwzględny	2,606907	2,672084	2,54173
Iloraz odchylenia	0,1294062	0,1200221	0,1445801
Korelacja	0,9915941	0,9927751	0,9895257

ogóle wykorzystywany w trakcie uczenia i jest przeznaczony do niezależnego określenia sprawności sieci po całkowitym zakończeniu procedury projektowania sieci. W kolumnach 8, 9 oraz 10 przedstawiono jakość uczenia, jakość walidacji i jakość testowania, w postaci liczbowej miary jakości sieci. Sposób mierzenia jakości zależy od natury sieci. Kolumna 4 podaje liczbę ukrytych warstw dla poszczególnych sieci.

Tablica 3 zawiera statystyki regresyjne wyznaczone na podstawie zbiorów danych. Statystyki regresyjne wyznaczone są niezależnie dla zbioru uczącego, walidacyjnego i testowego (kolumny 2, 3 oraz 4 w tablicy 3). Najważniejszym wskaźnikiem jakości

zbudowanego modelu jest iloraz odchyłeń standardowych. Dla bardzo dobrych modeli iloraz ten nie przekracza wartości 0,1. Dla znalezionej modelu iloraz ten nieznacznie przekracza wartość 0,1, lecz o dobrej jakości modelu świadczy bardzo wysoka wartość korelacji – bliska jedności.

Po zastosowaniu algorytmu wstecznej propagacji błędów dla sieci nr 5 i zmianie parametrów uczenia oraz bezwładności (współczynnik uczenia 0,1, bezwładność 0,01 oraz współczynnik uczenia 0,3, bezwładność 0,5) otrzymano wartości błędów w trakcie uczenia, których wykresy przedstawiono na rysunkach 6. Liczbę epok procesu uczenia określono na trzysta. Z rysunków wynika, że zarówno dla przypadku „a” jak i dla przypadku „b” proces uczenia sieci przebiegł prawidłowo, co widać po wartościach błędów uczących i walidacyjnych. Proces uczenia przebiegł lepiej dla przypadku „a”, ponieważ można zaobserwować mniejszy „rozrzut” wartości błęd uczącego od walidacyjnego.



Rys. 6. Wykres błędu uczenia. Współczynnik uczenia: 0,01, bezwładność: 0,1, liczba epok: 300.

Wykres błędu uczenia. Współczynnik uczenia: 0,5, bezwładność: 0,3, liczba epok: 300

Fig. 6. The graph of the learning error. Learning coefficient: 0,01, inertia: 0,1, number of epochs: 300.

The graph of the learning error. Learning coefficient: 0,5, inertia: 0,3, number of epochs: 300

6. Wnioski końcowe

Wykorzystanie do modelowania procesów sieci neuronowych – dla zgromadzonych danych pochodzących z przemysłowych procesów przerobczych dało bardzo dobre rezultaty, przy czym dzięki *Automatycznemu projektantowi sieci* można uzyskać sieć optymalną,

charakteryzującą się małą wartością ilorazu odchylenia oraz wartością korelacji zbliżoną do jedności.

Sieci neuronowe można więc stosować do skutecznego modelowania skomplikowanych procesów przemysłowych, w tym procesów przeróbczych, z dobrymi osiągnięciami. Istotny jest przy tym odpowiedni dobór w procesie uczenia zarówno współczynnika uczenia, jak i bezwładności, tak aby otrzymany model charakteryzował się jak najmniejszymi wartościami błędów i aby błędy te w znaczący sposób nie odbiegały wartościami od siebie.

LITERATURA

1. Kunysz J.: „Analiza możliwości wykorzystania modeli adaptacyjnych i sieci neuronowych w opisie procesów przeróbki polskich rud miedzi”, praca doktorska, AGH Kraków 2001 r.
2. Praca zbiorowa pod redakcją Leszka Rutkowskiego: „Sieci neuronowe i neurokomputery”, Politechnika Częstochowska, Częstochowa 1996.
3. Sprawozdanie pt. „Określenie parametrów technologicznych i modeli jako podstawy sterowania procesem mielenia i klasyfikacji IV i VI ciągu ZWR Rudna” – praca pod kierownictwem K. Sztaby, Przedsiębiorstwo Naukowo - Techniczne Omega, Kraków 2000 (praca niepublikowana).
4. Stęgowski Z., Tora B., Horn Carsten S., Stannek W.: „Model procesu przemysłowego z zastosowaniem sieci neuronowej”, *Gospodarka Surowcami Mineralnymi*, t.13, Zeszyt Specjalny, 1997.
5. „Statistica Neural Networks PL. Przewodnik problemowy”, StatSoft Polska Sp. z o.o., 2001.
6. Tadeusiewicz R.: „Sieci neuronowe”, Akademicka Oficyna Wydawnicza, Warszawa 1993.
7. Trybalski K., Ciepły J.: „Model typu ARMA flotacji rudy miedzi”, *Gospodarka Surowcami Mineralnymi*, t.3, str.175, Wydawnictwo CPPGSMiE PAN, Kraków 1997.
8. Trybalski K., Wszolek W.: „Modelowanie przemysłowych procesów flotacji przy pomocy sieci neuronowych”, *Mechanizacja i Automatyzacja Górnictwa*, Nr 4 – 5 (354), Katowice 2000.
9. Tumidajski T., Kunysz J., Trybalski T.: „Adaptive mathematical models of grinding and classification processes of Polish copper ores”, *Proceedings of the 14TH International Conference on Automation in Mining ICAMC 2001*, 3 – 5 September 2001, Tampere, Finland.
10. Watano S., Sato Y., Miyanami K.: „Application of neural network to granulation scale up”, *Powder Technology* 90, pp.153 – 159, 1997.
11. Cubillos F. A., Alvarez P. I., Pinto J. C., Lima E. I.: „Hybrid – neural modelling for particulate solid drying processes”, *Powder Technology* 87, pp.153 – 160, 1996.

Abstract

The analyze and modelling of various processes, which occur in the raw material processes is the most effective, and sometimes only possible, by the use of universal description methods. The attempt of neural networks application (one of the universal methods) – for the complex grinding and classification process modelling is presented in the paper. The considered processes are the part of the centre of the feed preparation in the ZWR „Rudna” KGHM „Polska Miedź” SA. In the researches the neural network, learned by the method of the backward error propagation is used. The results are obtained in the network structure and their parameters and coefficients form, which characterize a very high level of the data discernment.

Pracę wykonano w ramach pracy statutowej nr 10.10.100.658