

**WŁASNOŚCI WYBRANYCH TYPÓW SIECI NEURONOWYCH  
WYKORZYSTYWANYCH DO DETEKCJI POŁOŻENIA  
OCZU PACJENTA**

**PROPERTIES OF SELECTED NEURAL NETWORK TYPES  
USED FOR THE DETECTION OF THE POSITION OF PATIENT'S EYES**

**Robert Koprowski, Zygmunt Wróbel**

Zakład Komputerowych Systemów Biomedycznych, Uniwersytet Śląski

**SŁOWA KLUCZOWE:** sieci neuronowe, przetwarzanie obrazów, rozpoznawanie obrazów, oczy, detekcja

**STRESZCZENIE:** W pracy przedstawiono zastosowanie sieci neuronowych ze wsteczną propagacją błędów do zadania rozpoznawania oczu pacjenta widocznych w dowolnej projekcji na obrazie w świetle widzialnym. Rozpatrywana analiza dotyczy wskazywania z możliwie największą dokładnością środków oczu pacjenta dla różnych warunków pomiaru np. zmian oświetlenia. Przeprowadzono weryfikację błędów uczenia sieci neuronowej w zależności od liczby neuronów w poszczególnych warstwach, liczby warstw, typów neuronów oraz wybranych ustawień jej parametrów. Uzyskane wyniki potwierdzają konieczność losowego doboru przypadków pozytywnych i negatywnych (zawierających i nie zawierających obrazu oka) w wektorze uczącym. Najlepsze rezultaty (ok. 10 % błędnych rozpoznań dla 1 300) uzyskano dla sieci neuronowej ze wsteczną propagacją błędów dla jednej warstwy ukrytej składającej się z 10 neuronów po 400 epokach.

## **1. WSTĘP**

Rozpowszechnione metody weryfikacji zarówno cech biometrycznych jak też pomiarów ułożenia przestrzennego pacjentów wymagają pomiaru położenia oczu. W treści pozyskanego obrazu musi być widoczna cała sylwetka pacjenta, fakt ten powoduje, że nawet przy dużej rozdzielczości samego obrazu, jest mała liczba pikseli przypadających na oko pacjenta. Obraz taki jest dość specyficznym obrazem, w którym niejednokrotnie zachodzi konieczność analizy całej sceny pod względem występowania poszukiwanej pary oczu. Zwężenie obszaru poszukiwania pomimo istnienia markerów nie jest zagadnieniem elementarnym ze względu na występujące zakłócenia (liczne fałszywie wykryte markery jak też występująca część markerów nie wykrytych). Podawanych jest w tym miejscu w literaturze wiele metod bazujący na podejściu hierarchicznym – analiza globalna obrazu (wykrywanie położenia głowy) a następnie analiza lokalna (wykrywająca położenie oczu). Trudność wykorzystania podejść tego typu wynika nie tylko z wyprofilowania ich do konkretnych potrzeb, ale też ścisłym powiązaniu z interpretacją, np. kierunek ułożenia głowy w stosunku do osi kamery w oparciu np. o kontur.

Jedną z metod przynoszących oczekiwane rezultaty przy wykrywaniu położenia oczu pacjenta na scenie jest wykorzystanie do tego celu sieci neuronowych. Liczne pozycje literatury opisują niejednokrotnie zarówno implementacje typowych sieci neuronowych jak też metod hybrydowych składających się tylko w części z sieci neuronowych. Jedną z takich rozwiązań jest metoda opisana w publikacjach (Ranawera, 1996) i (Verschae *et al.*, 2003), w której skład wchodzi dopiero w ostatnim etapie implementacja sieci neuronowych. Niniejsza praca stanowi kontynuację opisaną metody w (Koprowski, *et al.*, 2006a) przeznaczonej do automatycznej analizy pomiaru ułożenia pacjenta na scenie 3D dla potrzeb rehabilitacji medycznej. W pracy zwrócono uwagę zarówno na przykładową implementację sieci neuronowych dla tego typu zastosowania jak też na weryfikację jej własności, zwracając szczególną uwagę na odpowiedni dobór liczb warstw ukrytych i liczby neuronów w poszczególnych warstwach. Dodatkowo nie jest z założenia wykorzystywana informacja przestrzenna wynikająca z analizy obu zdjęć stereowizyjnych (stereopary).

## 2. CEL PRACY

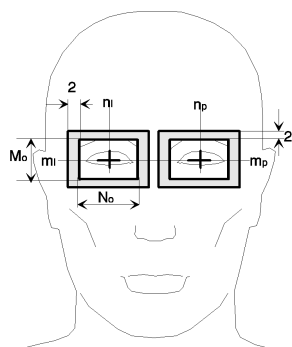
Celem pracy było zaproponowanie i weryfikacja własności sieci neuronowych ze wsteczną propagacją błędów wykorzystywanych do detekcji oczu pacjenta widocznych w dowolnej projekcji na obrazie w świetle widzialnym. Wyodrębniono następujące cele cząstkowe:

1. Zaproponowanie struktury sieci neuronowej oraz weryfikacja dokładności znajdowania oczu pacjenta dla różnych konfiguracji sieci.
2. Wyznaczenie położenia oczu wykorzystując wyznaczony typ i strukturę sieci oraz dodatkowe (zwiększające dokładność wyznaczonego położenia) elementy przetwarzania obrazów.

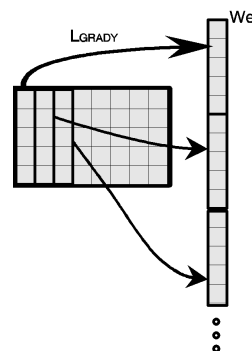
## 3. WYZNACZENIE OBSZARU ANALIZY ORAZ METODYKI TWORZENIA ELEMENTÓW WEKTORA UCZĄCEGO I UCZENIA SIECI

Specyfika wykorzystania sieci neuronowych do tego celu zastosowań niemal narzuca sieci ze wsteczną propagacją błędów. Sieci tego typu z definicji umożliwiają bowiem uczenie z nauczycielem i mogą zawierać w tym przypadku jedno lub dwa wyjścia – prawda, fałsz. Inne typy sieci uczące się bez nauczyciela (uczenie nienadzorowane), ze względu na swoją specyfikę (nieznane są klasy, do których należą mierzone wektory cechy), są rzadko wykorzystywane w tego typu zastosowaniach. Na wejście sieci podawane są zazwyczaj sekwencje kolejnych obrazów, dla których przeprowadzany jest proces rozpoznawania oczu jako wektor wejściowy. Składa się on z (jako kolejne kolumny) pikseli obrazów o rozdzielczości  $M_o \times N_o$ , zarówno zawierających jak też nie zawierających oka. Jak wykazano w (Koprowski *et al.*, 2006b) oraz (Wróbel *et al.*, 2001) w celu uzyskania zadowalających rezultatów obrazu te muszą zostać zmodyfikowane np. wyznaczeniem gradientu  $L_{GRADY}$  w osi  $oy$  monochromatycznego obrazu oczu  $L_{GRAY}$ . Ciąg uczący stworzono z obrazów zawierających oko, pozyskiwanych w wyniku interakcji z operatorem oraz obrazów nie zawierających oka dla pozostałych obszarów (nie zaznaczonych przez operatora). Operator wskazując środek położenia oczu jako parę współrzędnych  $m_1, n_1$  ( $m$  -

wiersz,  $n$ - kolumna) dla oka lewego i odpowiednio  $m_p$ ,  $n_p$  dla oka prawego tworzył automatycznie szereg obrazów w formie wektora wejściowego podawanych na wejście sieci. Możliwe zmiany położenia wskazywanych punktów środka oka wyeliminowano tworząc dodatkowe obrazy w promieniu  $\pm 2$  piksele w osi  $oy$  oraz  $\pm 2$  piksele w osi  $ox$ . Powstałe w ten sposób fragmenty obrazu  $L_{GRAY}$  nazwano dalej jako  $L_{GRADY}^{(i)}$  gdzie  $i$  oznacza numer fragmentu obrazu.



Rys. 1. Głowa z znacznymi obszarami położenia oczu



Rys. 2. Schemat poglądowy tworzenia wektora wejściowego z dwuwymiarowego obrazu  $L_{GRADY}$

Na podstawie tego przykładu oraz analizy treści obrazu wejściowego obliczono szacunkowo, iż obszarów zawierających oko jest znacznie mniej (ok. 7 % pikseli obrazu wejściowego) niż obszarów nie zawierających oka. Ponieważ najlepsze rezultaty są otrzymywane przy uczeniu sieci ciągiem uczącym zawierającym tyle samo obrazów oka ludzkiego, co ich nie zawierających (Koprowski *et al.*, 2000a), to ciąg uczący zmodyfikowano następująco:

- utworzono zestaw obrazów zawierających oko (ok. 25 obrazów dla każdego pacjenta);
- utworzono zestaw obrazów nie zawierających oka ludzkiego. Losowo wybrano obrazy nie zawierające oka tak, by ich liczba była zgodna z liczbą obrazów zawierających oko;
- utworzono ciąg wyjściowy dla dwóch wyjść takich, że jedno z wyjść wskazywało na obecność oka (0 - fałsz, 1 - prawda) natomiast drugie wskazywało na brak oka w obrazie  $L_{GRADY}^{(i)}$ ;
- uczono sieć neuronową losując dla pierwszej iteracji wartości wag;
- przeprowadzono 3 000 (lub testowo 5 000) epok;
- testując sieć wybrano te przypadki z ciągu uczącego, które sieć rozpoznała nieprawidłowo;
- utworzono zestaw obrazów nie zawierających oka z obrazów nieprawidłowo rozpoznanych, dobierając brakującą resztę z pozostałych obrazów jeszcze nie biorących udziału w uczeniu;
- przeprowadzono taki proces 60 krotnie (czyli sumarycznie 3000\*60 epok) modyfikując co 1 000 lub 3 000 epok ciąg uczący.

Oznaczając ciąg wyjściowy wzorcowy  $W_w^k$  oraz ciąg otrzymywany na wyjściu dla kolejnych elementów ciągu uczącego jako  $W_o^k$  można śledzić jakość uczenia co (przykładowo)  $k=3000$  epok obliczając błąd uczenia zgodnie z zależnością:

$$\delta(k) = \frac{1}{I} \sum_{i=1}^I |W_w^k(i) - W_o^k(i)| \quad (1)$$

Zależność ta będzie wykorzystywana do weryfikacji sieci ze wsteczną propagacją błędów w różnych konfiguracjach uczonych zgodnie z podaną sekwencją. Wartości błędu  $\delta(k)$  wpływa na liczbę obszarów otrzymywanych na wyjściu sieci (Koprowski, *et al.*, 2006a) a ich analiza na dokładność pomiaru położenia źrenic oczu.

#### 4. DOBÓR ORAZ WERYFIKACJA DOKŁADNOŚCI DZIAŁANIA SIECI DLA RÓŻNYCH KONFIGURACJI

Przedstawioną zależność na obliczanie błędu (1) wykorzystano do weryfikacji i doboru odpowiedniego typu sieci dla tego zastosowania. Ponieważ nie ma żadnych jednoznacznych przesłanek dotyczących doboru liczby neuronów w warstwach, liczby warstw sieci oraz typu wykorzystywanych neuronów rozpoczęto pomiary błędu dla następujących warunków:

- liczba warstw ukrytych  $v_u = 1 \div 2$ ,
- liczba neuronów w każdej warstwie ukrytej od  $v_{n,u} = 30 \div 90$ ,
- liczba epok dla wylosowanej postaci wektora uczącego  $v_i = 1 \div 5\ 000$ ,
- liczba losowań po każdej  $i$ -tej epoce –  $v_l = 1 \div 100$ ,
- funkcja aktywacji - sigmoidalna w postaci ogólnej:

$$f(x) = \frac{1}{1 + \exp(-\beta \cdot x)} \quad (2)$$

gdzie  $\beta=1$ .

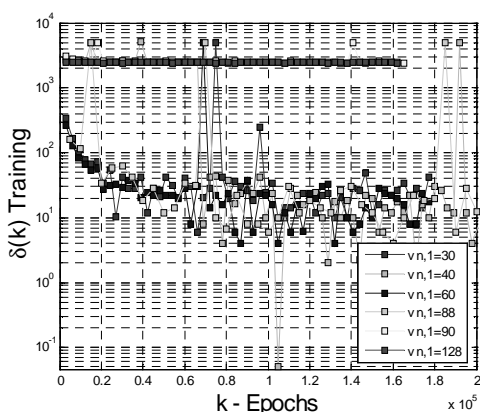
- współczynnik momentum  $v_m = 0.5 \div 1$ ,
- zatrzymanie procesu uczenia gdy: błąd uczenia będzie mniejszy od  $1 \cdot 10^{-4}$ , liczba epok  $i$  przekroczy dopuszczalną wartość lub pochodna szybkości uczenia będzie mniejsza od  $1 \cdot 10^{-20}$ ,
- warstwa wejściowa składa się z 258 neuronów a warstwa wyjściowa z 2.

Liczbę obrazów tworzących wektor uczący, ze względu na dużą liczbę mierzonych cech, ustalono arbitralnie na 4 960 (ok. 200 pacjentów). W tym 2 480 obrazów zawierających oko i 2 480 obrazów ich nie zawierających, umieszczonych zgodnie z podanym w poprzednim rozdziale opisie w losowej kolejności w wektorze uczącym. Wektor walidacyjny oraz testowy utworzono w analogiczny sposób odpowiednio o rozmiarach 4 960 (ok. 200 pacjentów – podobnie jak wektor uczący) oraz 19 500 (ok. 13 000 pacjentów).

W kolejnych podrozdziałach zostanie przedstawiona analiza wpływu wspomnianych cech ( $v_{n,u}$ ,  $v_u$ ,  $v_i$  oraz  $v_l$ ) na błąd całkowity  $\delta(k)$ .

#### 4.1. Liczba neuronów i liczba warstw

Na podstawie przesłanek będących rezultatem pomiarów wykonanych w poprzednim podrozdziale przeprowadzono pomiary zmian wartości błędu dla różnej liczby neuronów oraz warstw ukrytych. Na podstawie wstępnych pomiarów oraz przesłanek opisanych w (Down, 2002), (Hippert, 2001), (Tolstoy *et al.*, 2002) ustalono, że zgrubnie liczba neuronów w warstwie ukrytej powinna oscylować wokół pierwiastka z iloczynu liczby neuronów wejściowych i liczby neuronów wyjściowych ( $\sqrt{258 * 2} \approx 32$ ). Przyjęto zatem liczbę neuronów  $v_{n,1} = 30, 40, 60, 88, 90$  i  $128$  dla jednej warstwy ukrytej ( $v_u=1$ ) oraz liczbę neuronów  $v_{n,1} = 40, 60$  i  $v_{n,2} = 4, 8, 20$  (dla  $v_u=2$ ) dla dwóch warstw ukrytych, a wyniki przedstawiono poniżej (rys. 3, rys. 4).

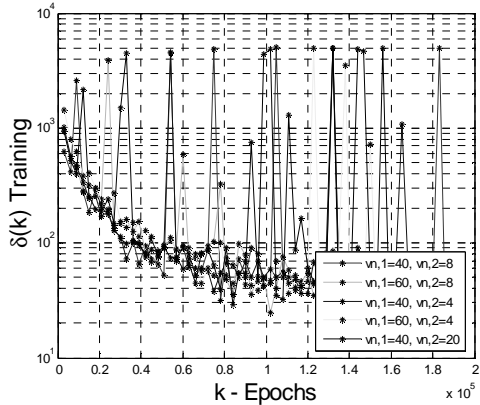


Rys. 3. Przebieg zmian wartości błędu  $\delta(k)$  w procesie uczenia dla kolejnych epok  $k=1,2,..,2*10^5$ ,  $v_{n,1}=30, 40, 60, 88, 90$  i  $128$  oraz  $v_u=1$ ,  $v_m=0.95$

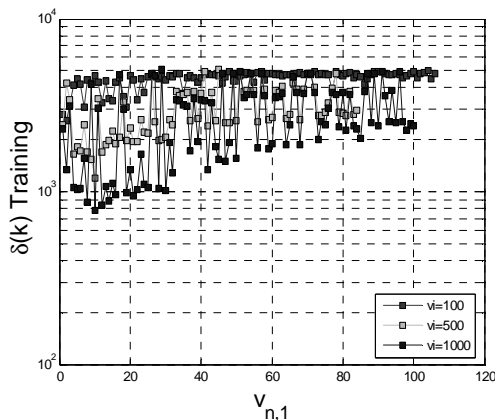
zbyt mała, przez co błąd uczenia sieci utrzymuje się na poziomie  $\delta(k) \approx 550$ . Podsumowując najbardziej optymalnym doborem wydają się sieć z  $v_{n,1}=30$  oraz ok.  $k=0,5*10^5$  epok.

Odmiernym zagadnieniem jest wpływ liczby neuronów w warstwie ukrytej i samej liczby warstw ukrytych na proces uczenia. Otrzymane wyniki dla  $k=1,2,..,2*10^5$ ,  $v_{n,1} = 40, 60$  oraz  $v_{n,2} = 4, 8$  przy  $v_u=2$  pokazano na rysunku 4.

Przedstawiony na wykresie (rys. 3) przebieg zmian wartości błędu  $\delta(k)$  w funkcji  $k$  dla różnej liczby neuronów w warstwie ukrytej na charakter eksponentialny. Dla wartości  $v_{n,1}=30, 40, 60$  oraz  $88$  sieci uczą się prawidłowo osiągając błąd uczenia  $\delta(k) \approx 20$ . Przy  $k=1.1*10^5$  widoczne jest pierwsze przeuczenie sieci  $v_{n,1}=40$  wynikające ze zbyt dużej liczby epok ( $k$ ). Podobne przeuczenia sieci obserwowane już są coraz częściej dla większej liczby epok. Oznacza to, że szacowana liczba epok dla wektora uczącego o rozmiarze  $4\ 960$  wynosi  $k=1*10^5$ . Odmienne wartości błędów  $\delta(k)$  są widoczne dla  $v_{n,1}=90$  oraz  $128$ . Dla tak dużej liczby neuronów w warstwie ukrytej liczba epok jest



Rys. 4. Przebieg zmian wartości błędu  $\delta(k)$  w procesie uczenia dla kolejnych iteracji  $k=1,2,\dots,2 \cdot 10^5$ ,  $v_{n,1}=40, 60$  oraz  $v_{n,2}=4, 8$  przy  $v_u=2$ ,  $v_m=0.95$



Rys. 5. Przebieg zmian wartości błędu  $\delta(k)$  w funkcji liczby neuronów w warstwie ukrytej  $v_{n,1}$  przy  $v_u=1$  oraz  $v_i=100,500,1000$ ,  $v_m=0.95$

zagadnieniem jest obserwacja zmian wartości błędu dla różnej liczby iteracji przy założeniu stałego wektora uczącego.

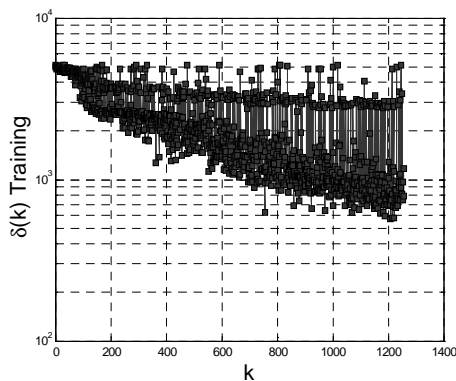
#### 4.2. Liczba iteracji

Dla wylosowanej postaci wektora uczącego przeprowadzono pomiary wpływu liczby epok na błąd uczenia. Wyniki przedstawiono na rysunku 6.

W tym przypadku (rys. 4) proces uczenia wygląda podobnie do wykresu przedstawionego na rysunku 3, jednak nie są widoczne miejsca przeuczenia sieci. Widoczne są jednak liczne punkty  $\delta(k) > 1 \cdot 10^3$ , dla których obserwowane jest niedouczenie sieci. Liczba punktów, w których sieć jest niedouczona maleje wraz ze wzrostem liczby epok  $k$  oraz nieznacznie ze zmniejszaniem się liczby neuronów w pierwszej i drugiej warstwie ukrytej. Przedstawiony przebieg na rysunku 4 i wcześniej na rysunku 3 oraz ich porównanie potwierdza przypuszczenie, że optymalną konfiguracją sieci jest  $v_{n,1} < 30$  w jednej warstwie ukrytej. Dowodem tego jest wykres przedstawiony na rysunku 5.

Przedstawiony na nim przebieg zmian wartości błędu ( $k$ ) w funkcji liczby neuronów w warstwie ukrytej  $v_{n,1}$  przy  $v_u=1$  oraz  $v_i=100,500,1000$  wskazuje na najmniejszy osiągalny błąd ( $\delta(k) < 1 \cdot 10^3$ ) dla  $v_{n,1} \in (10,30)$ . Liczba epok dla każdorazowego losowania wektora jest zależna odwrotnie proporcjonalnie (wzrost  $v_i$  powoduje zmniejszanie  $\delta(k)$ ).

Przedstawione powyżej zależności zostały zrealizowane dla kolejnych losowań tworzących inną postać wektora uczącego realizowanych, co zamierzoną liczbę epok (obserwowano też otrzymywane rezultaty dla wektora walidacyjnego). Interesującym

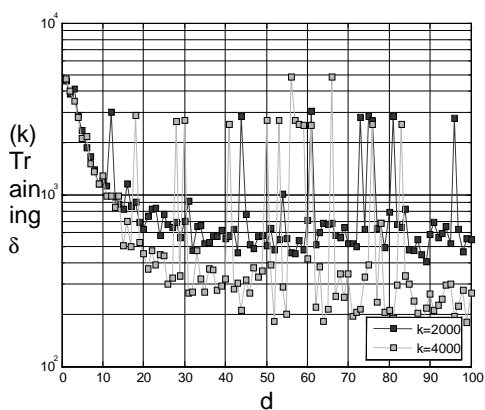


Rys. 6. Przebieg zmian wartości błędu  $\delta(k)$  w funkcji zmiennej wartości liczby epok,  $k=1,2,..1400$ ,  $v_{n,1}=10$  przy  $v_u=1$ ,  $v_m=0.95$

testującego w następujący sposób:

$$W_{T\_NOISE}^k = W_T^k + \frac{W_{NOISE}^k}{d} \quad (3)$$

gdzie:  $W_{T\_NOISE}^k$  - zakłócony wektor danych wejściowych,  
 $W_T^k$  - wektor danych wejściowych.



Rys. 7. Przebieg zmian wartości błędu  $\delta(k)$  w zależności od wartości szumu na przedziale  $\langle -0.5, 0.5 \rangle$  dla sieci  $k=2000$  i  $k=4000$ ,  $v_{n,1}=10$  i  $v_u=1$ ,  $v_m=0.95$

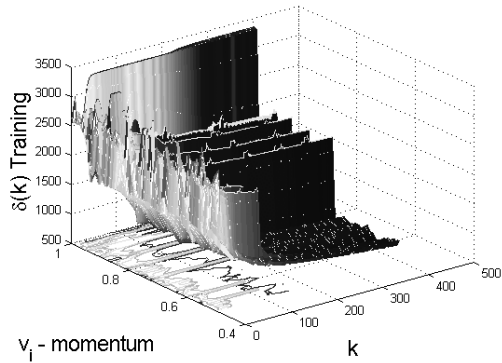
Na przedstawionym wykresie zależności błędu uczenia od liczby epok widoczna jest (spodziewana) silna proporcjonalna zależność. Wzrastająca liczba epok powoduje zmniejszanie się błędu uczenia, co jest też doskonale widoczne na wynikach otrzymywanych dla wektora walidacyjnego.

#### 4.3. Wpływ szumu na wynik uczenia

Przeprowadzone powyżej pomiary nie uwzględniają wpływu szumu zawartego w obrazie na wyniki rozpoznawania. Pomiary tego typu zostały prowadzone dla sieci o  $v_{n,1}=10$  i  $v_u=1$  której badano błąd rozpoznawania wprowadzając sztucznie szum do wektora

Otrzymane wyniki dla dodawanego szumu o rozkładzie jednostajnym na przedziale  $\langle -0.5, 0.5 \rangle$  przedstawiono na rysunku 7. Z przedstawionego na nim wykresu wynika jednoznacznie, że szum o rozkładzie jednostajnym na przedziale  $\langle -0.5, 0.5 \rangle$  o poziomie  $d > 30$  (tj.  $< 3\%$ ) nieznacznie (na poziomie dokładności nauczania się sieci neuronowych) wpływa na wynik rozpoznania. Zwiększenie liczby epok do  $k=4000$  zmniejsza błąd, zachowując zbliżony kształt obwiedni wykresu (rys. 7).

#### 4.4. Wpływ stałej momentum



Rys. 8. Przebieg zmian wartości błędu  $\delta(k)$  w kolejnych epokach  $k$  w zależności od wartości momentum  $v_m$  dla sieci  $k=400$ ,  $v_{n,1}=10$  i  $v_u=1$

Bezładność sieci pod względem uzyskiwanych wyników procesu uczenia nadaje współczynnik *momentum*. Zmieniając wpływ zmiany wag na proces uczenia pozwala na uzyskiwanie większego bądź mniejszego stopnia uogólnienia nie wpadając w niegłębokie minima lokalne, co poprawia efektywność uczenia. Zazwyczaj, zgodnie z przesłankami (Bharath *et al.*, 1994; Foster, 1992; Ranawera, 1996) momentum mieści się w zakresie  $v_m=0.5 \div 1$ . Dla tego właśnie przedziału, co 0.1 przeprowadzono badania symulacyjne dla sieci o liczbie epok  $k=400$ ,  $v_{n,1}=10$  przy  $v_u=1$ ,  $v_m=0.5 \div 1$  (rys. 8).

Obserwując szybkość uczenia oraz wartość błędu po 400 epokach łatwo zauważyć, że wartości momentum winny być zawarte w zakresie  $v_m=0.5 \div 0.7$ . Dla takiego przedziału otrzymuje się najlepsze wyniki (najmniejszy błąd przy najszybciej uczącej się sieci).

#### 4.5. Wybór funkcji aktywacji

Przeprowadzona analiza dotycząca liczby epok, częstotliwości losowania po każdej epoce, wybór liczby neuronów w warstwie ukrytej, aż w końcu liczby warstw i wartości momentum zakładały typ funkcji przejścia neuronu jako sigmoidalną. Poniżej, w tabeli 1 przedstawiono badane typy sieci.

Tabela 1. Typ sieci oraz ich funkcja przejścia

Nazwa funkcji	Funkcja	Skrót
sigmoidalna	$f(x) = \frac{1}{1 + \exp(-x)}$	logsig
Tangens hiperboliczny	$f(x) = \frac{2}{1 + \exp(-2 \cdot x)} - 1$	tansig
liniowa	$f(x) = x$	purelin
Symetryczna z nasyceniem	$f(x) = \begin{cases} -1 & \text{dla } x \leq -1 \\ x & \text{dla } -1 \leq x \leq 1 \\ 1 & \text{dla } 1 \leq x \end{cases}$	satlin
Symetryczna z nasyceniem i zerem	$f(x) = \begin{cases} 0 & \text{dla } x \leq 0 \\ x & \text{dla } 0 \leq x \leq 1 \\ 1 & \text{dla } 1 \leq x \end{cases}$	satlins
Dodatnia liniowa	$f(x) = \begin{cases} x & \text{dla } x \geq 0 \\ 0 & \text{dla } x \leq 0 \end{cases}$	poslin

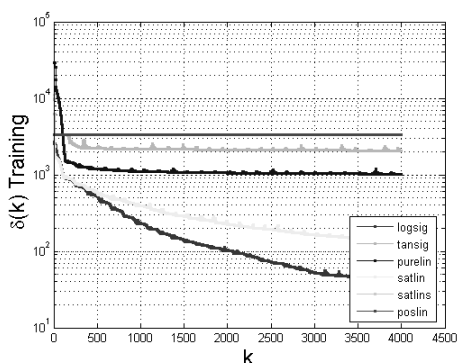


Uzyskiwane wyniki dla przedstawionych w tabeli 1 funkcji przejścia są przedstawione na rysunku 9. Przedstawiony na nim wykres potwierdza poprawność doboru funkcji przejścia sigmoidalnej ze względu na najmniejszy osiągnięty błąd  $\delta(k)$  sieci w porównaniu z innymi typami dla tych samych ustawień zarówno liczby neuronów jak też liczby warstw ukrytych.

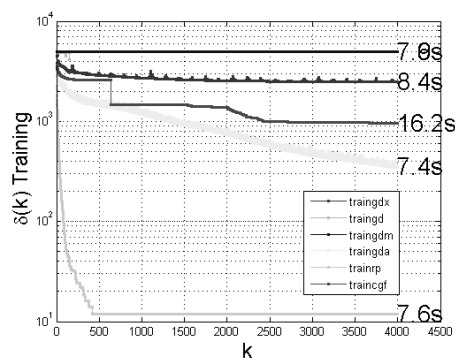
#### 4.6. Wybór sposobu uczenia

Pomiary wpływu metody uczenia na otrzymywane rezultaty (błąd  $\delta(k)$  oraz czas uczenia) przeprowadzono dla następujących metod:

- *traingdx* - algorytm największego spadku z momentum i adaptacyjnym doborem współczynnika uczenia,
- *traingcp* - metoda gradientów Fletcher-Reeves'a,
- *traingd* - algorytm największego spadku,
- *traingdm* - algorytm największego spadku z momentum,
- *traingda* - algorytm największego spadku i adaptacyjnym doborem współczynnika uczenia,
- *trainrp* - algorytm RPROP,
- *traingcf* - algorytm gradientów sprzężonych Fletcher-Powell.



Rys. 9. Przebieg zmian wartości błędu  $\delta(k)$  dla kolejnych typów sieci przy  $k=4000$ ,  $v_{n,1}=10$  i  $v_u=1$ ,  $v_m=0.95$



Rys. 10. Przebieg zmian wartości błędu  $\delta(k)$  dla kolejnych metod uczenia sieci przy  $k=4000$ ,  $v_{n,1}=10$  i  $v_u=1$ ,  $v_m=0.95$  (na końcach wykresów podano czas uczenia)

Otrzymane rezultaty przedstawiono na rysunku 10. Z wykresu wynika, że najlepsze rezultaty w sensie najmniejszej wartości błędu  $\delta(k)$  uzyskano dla metody gradientów sprzężonych Fletcher-Powell. Jednak w tym przypadku występuje, przy nadmiernej liczbie epok, prawdopodobieństwo przeuczenia sieci. Z tego powodu proponowana jest metoda gradientowa (*traingda*).

## 5. PODSUMOWANIE

Dla sieci neuronowej z 10 neuronami w warstwie ukrytej oraz współczynnika momentum  $v_m=0.7$ , liczby epok  $k=400$  i liczby warstw ukrytych  $v_u=1$  uzyskano dla wektora testowego ok. 10 % błędnych rozpoznań dla 1 300 (pacjentów) obrazów  $L_{GRAY}$  poddawanych procesowi rozpoznania. Przez błędne rozpoznanie uznano sumę wyników fałszywie pozytywnych i fałszywie negatywnych w stosunku do wszystkich przypadków. Zarówno jakość rozpoznawania rozumianą w sensie polepszenia dokładności pomiaru położenia oczu jak też zmniejszenie liczby błędnych rozpoznań zmniejszono stosując dodatkowe przekształcenia obrazu wynikowego z sieci. Przykładowym kryterium polepszającym uzyskane wyniki może być miara prawidłowego rozpoznania, zdefiniowana jako największa znormalizowana powierzchnia utworzona z wyników sieci i największa odpowiadająca jej wartość. W praktyce dla tego samego typu sieci i tej samej wielkości wektora uczącego otrzymano, stosując wspomniane algorytmy przetwarzania obrazów, jedynie ok. 3 % błędnych rozpoznań.

## 6. LITERATURA

- Bharath R. Drosen J., 1994. *Neural Network Computing*. Windcrest, USA.
- Down T., 2002. Introduction to Neural Computing & Pattern Recognition, COMP3700 Machine Learning website, University of Queensland, Australia.
- Foster W., 1992. *Neural network forecasting of short, noisy time series*. Computers Chem. Engng., Vol. 16, No. 4, s. 293-297.
- Hippert H., 2001. Neural Networks for Short-Term Load Forecasting. *A Review and Evaluation, IEEE Trans. on Power Systems*, Vol. 16, No. 1, s. 44-54.
- Koprowski R., Tokarczyk R., Wróbel Z., 2006a. Zastosowanie metod analizy obrazów w trójwymiarowym pozycjonowaniu ciała pacjenta. *Archiwum Fotogrametrii, Kartografii i Teledetekcji*, Vol. 16 (w druku).
- Koprowski R., Wojaczyńska-Stanek K., Wróbel Z., 2000b. Wyznaczanie konturu człowieka na zdjęciach termowizyjnych. SYBIOSIS 2000.
- Ranawera D., 1996. Effect of probabilistic inputs on neural network-based electric load forecasting. *IEEE Trans. on Neural Networks*, Vol. 7, No. 6, Nov. s. 1528-1532.
- Tolstoy L., Parsjani H. 2002. Application of Fourier Descriptors and Neural Networks to Shape Recognition, CRC 2000 Proceeding, Mayaguez, Puerto Rico.
- Verschae R. Ruiz-del-Solar J., 2003. A Hybrid Face Detector based on an Asymmetrical Adaboost Cascade Detector and a Wavelet-Bayesian-Detector. Int'l Work-Confer. on Artificial and Natural Neural Networks, s. 742-749.
- Wang C.Y., Li M.X., 2005. Convergence property of the fletcher-reeves conjugate gradient method with errors. *Journal Of Industrial And Management Optimization*. Vol. 1, No 2.
- Wróbel Z., Koprowski R. 2001. *Przetwarzanie obrazów w programie Matlab*. Wyd. Uniwersytet Śląski.
- Zhang B., Dong Z., 2001. An Adaptive Neural-Wavelet Model for Short Term Load Forecasting. *Electric Power Systems Research*, 59, s. 121-129.

Praca powstała w ramach grantu nr 4 T12E 052 27 pt. „Automatyzacja pomiaru na obrazach cyfrowych w zastosowaniu do fotogrametrycznego systemu trójwymiarowego pozycjonowania ciała dla celów rehabilitacji leczniczej”.

**PROPERTIES OF SELECTED NEURAL NETWORK TYPES  
USED FOR THE DETECTION OF THE POSITION OF PATIENT'S EYES**

KEY WORDS: neural networks, image processing and recognition, eyes detection

**Summary**

This paper presents the application of a backpropagation neural network to the isolation of patient's eyes visible in any picture in visible light. The method properties were analysed depending on: the number of neurons in the individual layers, the number of layers, neuron types and selected network types. The research covered in this paper is a part of a project aimed at developing methods for the three-dimensional mapping of a patient's body. The project is funded from grant no. 4 T12E 052 27: The automation of the measurements of digital images for a photogrammetric system for a 3D patient's body positioning employed in medical rehabilitation.

The obtained results confirm the usefulness of neural networks in eye recognition, in which there were 10 % incorrect recognitions for 1 300 patients. Due to the large number of measured properties, the number of images that formed the learning vector was set to 4 960 (about 200 examined patients). The learning vector included 2 480 images containing an eye, as well as 2 480 images without eyes, put in random order. The validation and test vector were created in a similar way, using 4 960 and 19 500 images (of 200 and 1 300 patients), respectively. The quality of recognition was improved and the number of incorrect recognitions was reduced by employing additional transformations of the result images from the neural network. For instance, comparing the maximum area determined by the network and the respective maximum output value from the network allowed reducing the error to approximately 3 %.

Additionally, the paper proposes various neural network structures and configurations and evaluates their effectiveness in this application. The methodology covered in this paper can be extended to other problems of a similar type.

Dr inż. Robert Koprowski  
e-mail: koprow@us.edu.pl  
tel. +32 2918381 w. 741

Prof. dr hab. inż. Zygmunt Wróbel  
e-mail: wrobel@us.edu.pl  
tel. +32 2918381 w. 736