Podstawy metodyczne klasyfikacji danych hiperspektralnych

W niniejszym rozdziale zaprezentowano algorytmy, które posłużyły do klasyfikacji danych DAIS 7915 obszaru Tatr Wysokich. Do przygotowania wzorca do uczenia sieci wybrano homogeniczne spektralnie poligony poszczególnych zbiorowisk roślinnych. Były to piksele, których odpowiedź spektralna pochodziła tylko od danego, analizowanego zbiorowiska roślinnego (piksele czyste spektralnie, tzw. *endmembers*). Procedura ta bazowała na pakiecie ENVI i analizie PPI (*Pixel Purity Index*). Wyniki analizy PPI zostały nałożone na mapę roślinności rzeczywistej (Kozłowska, 2006) i załadowane w pamięć odbiornika GPS Trimble GeoXT w celu ich rzeczywistej lokalizacji podczas badań terenowych. Z terenowego rozpoznania obiektów wybrano poligony, które posłużyły do uczenia sieci neuronowych.

W kolejnym kroku wytypowane poligony posłużyły do wyboru charakterystyk spektralnych spośród reprezentujących je pikseli czystych spektralnie. Uzyskane charakterystyki wykorzystano do wykonania półautomatycznej klasyfikacji SAM (*Spectral Angle Mapper*). Algorytm ten pozwolił wybrać obszary¹ zgodne spektralnie z poligonami pozyskanymi w trakcie badań terenowych do uczenia sztucznych sieci neuronowych. Po wykonaniu klasyfikacji SAM, uzyskane wyniki zostały ponownie załadowane do odbiornika GPS i w trakcie kolejnej sesji terenowej pozwoliły wybrać większą liczbę poligonów poszczególnych zbiorowisk roślinności rzeczywistej do utworzenia wzorca weryfikującego dane poklasyfikacyjne z symulatora fuzzy ARTMAP.

Wprowadzenie każdego dodatkowego kanału obrazu do symulatora sztucznych sieci neuronowych powoduje zwiększenie liczby neuronów, co rozbudowuje sieć, zwiększając liczbę powiązań i komplikując możliwości przesyłania sygnałów. W efekcie wydłuża to proces klasyfikacji, dlatego istotne jest zredukowanie liczby kanałów najmniej informacyjnych. Procedura ta wykonywana jest poprzez analizę informacyjności poszczególnych kanałów lub też dekorelację szumu zawartego w poszczególnych kanałach i analizę głównych składowych (PCA), jest to tzw. transformacja Minimum Noise Fraction (MNF). Efektem tej transformacji jest utworzenie jakościowo nowych kanałów MNF, które pozwalają wybrać najmniej skorelowane informacyjnie kanały obrazowe, ograniczając znacząco liczbę powiązań w sieci neuronowej (kompresja danych).

Do klasyfikacji danych hiperspektralnych wykorzystane zostały 2 symulatory sztucznych sieci neuronowych MLP (perceptron wielowarstwowy ze wsteczną propagacja błedów) oraz fuzzy ARTMAP (FAM). Pierwszy symulator dostępny jest w darmowym pakiecie SNNS, drugi natomiast został udostępniony przez prof. Paolo Gambę (Uniwersytet w Pawii, Włochy). Właściwa klasyfikacja została wykonana w FAM, wynikało to głównie z szybszego, prostszego przygotowania danych (zarówno do klasyfikacji, jak i wizualizacji uzyskanych wyników). Symulator MLP został wykorzystany na wstępnym etapie klasyfikacji do określenia optymalnej architektury sieci (dobór liczby kanałów oraz jakość klasyfikacji względem liczby wykorzystanych danych). Zaletą symulatora MLP z pakietu SNNS jest możliwość dowolnego przerywania procesu klasyfikacji i ponownego uruchamiania wytrenowanej sieci. Ma to ogromne znaczenie w automatyzacji klasyfikacji tych samych klas na różnych zdjęciach tego samego sensora, gdyż raz wytrenowana sieć może zostać wykorzystana do klasyfikacji tej samej formy lub obiektu na różnych scenach (po uprzednim wykonaniu korekcji atmosferycznej).

Wskaźnik czystości piksela PPI (Pixel Purity Index)

Techniki hiperspektralne pozwalają na ekstrakcję obiektów z obrazów (*feature extraction*). Jednym

¹ Piksele po transformacji PPI i weryfikacji terenowej na zgodność występowania z danym zbiorowiskiem roślinności rzeczywistej.



Ryc. 3. Rzut 3-wymiarowej przestrzeni (xyz) utworzonej z 1000 pikseli w projekcji kanałów 1, 2 i 3 MNF na płaszczyznę. Barwne piksele oznaczają piksele czyste spektralnie (tzw. endmembers) Fig. 3. Projection of 3D space, which is built from 1000 pixels of MNF 1,2,3 bands. Pixels of endmembers are coulored



Ryc. 4. Wybrane charakterystyki spektralne z pikseli czystych spektralnie pozyskane z analizy PPI (endmembers) z obrazu DAIS 7915 do klasyfikacji SAM Fig. 4. Selected spectral characteristics derived from endmembers (PPI analysis) from DAIS 7915 images for SAM classification

z podstawowych algorytmów jest identyfikacja homogenicznych pikseli, które służą jako wzorce klasyfikacji poszczególnych obiektów (Kruse i inni, 1993; Boardman, 1994; Boardman, Kruse, 1994; Schwengerdt, 1997). Cała grupa algorytmów zajmujących się pozyskaniem takich wzorcowych pikseli i charakterystyk spektralnych nosi ogólną nazwę algorytmów ekstrakcji pikseli czystych spektralnie (EEAs²), a opracował je J.W. Boardman (Kruse i inni, 1993; Boardman, 1994). Wskaźnik czystości piksela (PPI)³ jest obliczany w kolejnych rotacjach n-wymiarowej przestrzeni (n odpo-

 $^{^{\}scriptscriptstyle 2}$ Endmember Extraction Algorithms.

³ Dzięki ogólnej dostępności algorytmów oraz kodów źródłowych transformacji PPI (w środowisku C++, IDL oraz Matlab) istnieje obecnie szereg modyfikacji PPI, do których należy zaliczyć algorytm Fast Iterative PPI (FIPPI). Wskaźnik FIPPI został zaproponowany przez C.-I. Chang, A. Plaza (2006) i należy do tzw. wirtualnej wymiarowości przestrzeni (VD – Virtual Dimensionality) (Chang, 2003; Chang, Du, 2004). Algorytm FIPPI bazuje na wykonywaniu kolejnych, niezależnych iteracji, co pozwala na szybszą oraz dokładniejszą inicjalizację procedur poszukiwania pikseli czystych spektralnie.



Ryc. 5. Fragment zobrazowania DAIS 7915 Doliny Gąsienicowej po transformacji PPI. Jasne obszary przedstawiają homogeniczne spektralnie piksele

Fig. 5. A part of Gasienicowa Valley on the DAIS 7915 after the PPI transformation. Bright polygons present homogeneous spectrally pixels

wiada liczbie kanałów) na przypadkowo generowanych wektorach. Piksele, które znajdą się w najbardziej zewnętrznych częściach wielowymiarowego histogramu, są zapisywane jako czyste spektralnie⁴ (ryc. 3; Boardman, Kruse, 1994; Zagajewski i inni, 2009). Aby przyśpieszyć ten proces wykonuje sie redukcje zbednych kanałów, np. za pomocą transformacji MNF (Minimum Noise *Fraction*). Pierwsze kanały MNF (po transformacji) zawierają najmniej skorelowane informacje względem kanałów, które posłużyły do wykonania transformacji)5. Zautomatyzowanym algorytmem analizy pikseli jest APPI (Automatic Pixel Purity Index), procedura opracowana na Uniwersytecie Maryland (Chaudhry, 2005). Metoda APPI polega na tym, że w sposób automatyczny inicjuje się kilka oddzielnych procedur wyłaniania pikseli czystych spektralnie. Do fazy finałowej algorytmu kierowane są tylko piksele wyłonione we wszystkich pojedynczych analizach.

Wybór pikseli metodą PPI pozwala na wyróżnienie na obrazie pól treningowych dowolnie wybranych klas, np. form pokrycia terenu czy zbiorowisk roślinnych (ryc. 4).

Końcowe etapy transformacji PPI pozwalają przeprowadzić jakościową analizę wyekstrahowanych pikseli (ryc. 5).

Transformacja Minimum Noise Fraction (MNF)

Źródła transformacji MNF sięgają końca lat 1980. kiedy to zespół A.A. Greena (1988) poszukiwał alternatywnych algorytmów do analizy głównych składowych (PCA⁶) w celu redukcji 10 kanałowych danych ATM⁷, które miały posłużyć łączeniu z innymi danymi. Główna idea MNF polegała na wykonaniu podwójnej analizy głównych składowych PCA. Obecnie w pierwszym etapie analizowana jest macierz kowariancji szumu w celu jego dekorelacji i przeskalowania, a w drugim etapie wykonywana jest klasyczna analiza PCA. Analiza oraz usunięcie szumu odbywa się za pomocą tzw. metody przesunięcia różnic (shift difference method). Polega ona na stworzeniu wirtualnego szumu (oscylacji sygnału wokół wybranej wartości) danego kanału i porównaniu go ze zmiennością sygnału każdego kolejnego piksela rzeczywistego zobrazowania. Koncepcja MNF, polega na liniowej transformacji głównych składowych⁸ pod kątem analizy wskaźnika sygnał/szum. Efektem jest zestaw nowych danych, które są uporządkowane względem szumu. Do analiz wykorzystuje się liczbę kanałów MNF wskazaną przez wypoziomowanie się linii informacyjności (w niniejszym opracowaniu wykorzystanych zostało 20 pierwszych kanałów MNF, ryc. 6). Pierwsze kanały MNF są pozbawione szumu i zawierają najmniej skorelowane informacje poszczególnych kanałów (ryc. 7). Pozwala to na wykorzystanie do analiz mniejszej liczby kanałów (Olesiuk, Zagajewski, 2008) niż by miało to miejsce po zastosowaniu innych algorytmów kompresji danych (Ramachandra, Uttam, 2005). W przypadku wykorzystania symulatorów sztucznych sieci neuronowych jest to znaczące ułatwienie, gdyż ilość wprowadzanych do symulatora sygnałów redukuje czas procesu uczenia i klasyfikacji sieci.



Ryc. 6. Wykres informacyjności kolejnych kanałów MNF 5 linii DAIS 7915 (ośX – kolejne kanały MNF) Fig. 6. Eigenvalue of the MNF transformation of the DAIS 7915 flight line no. 5

⁴ Wartość PPI zależy także od liczby wykonanych iteracji reprojekcji pikseli w *n*-wymiarowej przestrzeni i założonych progów położenia wektorów, na które są rzutowane piksele.

⁵ Proces MNF polega na jednoczesnej kompresji danych i eliminacji najbardziej zaszumionych kanałów (analiza SNR – Signal to Noise Ratio).

⁶ Principal Component Analysis.

⁷ Airborne Thematic Mapper.

⁸ Gdyby wartość szumu była identyczna we wszystkich kanałach spektralnych zobrazowania, transformacja MNF odpowiadałaby transformacji głównych składowych PC (Green i inni, 1988).



Ryc. 7. Fragment 5 linii zobrazowania DAIS 7915 po transformacji MNF: A – 1 kanał MNF, B – 10 kanał MNF, C – 20 kanał MNF, D – 30 kanał MNF

Fig. 7. A part of the 5th flight line of the DAIS 7915 image after MNF transformation: A - 1st MNF band, B - 10th MNF band, C - 20th MNF band, D - 30th MNF band

Klasyfikacja Spectral Angle Mapper (SAM)

Koncepcja klasyfikacji wykorzystującej cechy spektralne obiektu, zapisane w bardzo wielu kanałach tego samego piksela, pojawiła się w laboratoriach *C*entre for the Study of Earth from Space (CSES) Cooperative Institute for Research in Environmental Sciences (CIRES) University of Colorado na początku lat 1990. Bazowano wtedy na pierwszych hiperspektralnych zobrazowaniach GERIS, HIRIS⁹, AIS oraz AVIRIS (Kruse i inni, 1993; Kruse, Lefkoff, 1993). Jednym z przełomowych algorytmów tego pakietu EEAs (*Endmember Extraction Algorithms*) był *Spectral Angle Mapper* (SAM)¹⁰ opra-

⁹ High Resolution Imaging Spectrometer.

¹⁰ Pomysł ten był rozwijany w języku programowania IDL na platformie UNIX w pakiecie Spectral Image Processing System (SIPS).

cowany przez dr J.W. Boardmana, stypendystę CIRES (Boardman, Kruse, 1994).

Klasyfikator SAM jest nadzorowana i automatyczna metoda do bezpośredniego porównywania spektrów, pozyskanych z każdego piksela obrazu hiperspektralnego, z dowolnym wzorcem (np. z pomiarów laboratoryjnych, naziemnych) lub badanego obrazu (najczęściej z tzw. pikseli czystych spektralnie). Analizowane spektrum danego piksela (B) i referencyjna charakterystyka spektralna (A) (np. piksela czystego spektralnie lub charakterystyki z biblioteki spektralnej) są traktowane jako wektory spektralne, pomiędzy nimi istnieje kąt spektralny (θ). Na osiach x,y (ryc. 8) zaprezentowane zostały współczynniki odbicia (ρ_{1}, ρ_{2}) dla kolejnych kanałów. Spektrum A jest wzorcem do porównania charakterystyki spektralnej danego piksela (B), f reprezentuje frakcje piksela czystego spektralnie (wzorca). Odległość pomiędzy punktami $f_A \cdot B_1$ i $f_A \cdot B_2$ określa różnice spektralne w poszczególnych długościach fal danych kanałów obrazu. Błąd odległości pomiędzy kanałami spektralnymi obrazu jest mierzony wartością RMSE (Kruse i inni, 1993; van der Meer i inni, 1997).

Pierwsze klasyfikacje SAM z wykorzystaniem pikseli czystych spektralnie i grupowaniu ich w biblioteki spektralne były prowadzone w latach 1980. w Stanach Zjednoczonych. Pionierskie prace dotyczyły identyfikacji minerałów (Goetz i inni, 1985; Lang i inni, 1987; Pieters, Mustard, 1988; Kruse, 1988; Kruse, Lefkoff, 1993; Crowley, 1993; Boardman, Kruse, 1994), roślinności (Gamon i inni, 1993; Roberts i inni, 1993; Elvidge i inni, 1993), pokrywy śnieżnej i lodowców (Nolin, Dozier, 1993), gazów zawartych w atmosferze (Gao, Goetz, 1990; Carrère, Conel, 1993), wód powierzchniowych (Hamilton i inni, 1993; Carder i inni, 1993).



Ryc. 8. Porównanie charakterystyk spektralnych pozyskanych ze wzorca oraz klasyfikowanego piksela (źródło: Kruse i inni, 1993; Dennison i inni, 2004, zmienione)

Fig. 8. Comparission of spectral characteristics of an analysed pixel and a reference characteristic (source: Kruse et al., 1993; Dennison et al., 2004, modified) Długość wektora spektralnego L_p określa się na podstawie wzoru (1), gdzie M oznacza liczbę kanałów, natomiast kąt spektralny θ pozwala obliczyć zależność (2) (Kruse i inni, 1993):

$$L_{p} = \sqrt{\sum_{\lambda=1}^{M} \rho_{\lambda}^{2}}$$
(1)

$$\theta = \cos^{-1}\left(\frac{\sum_{\lambda=1}^{M} \rho_{\lambda} \dot{\rho_{\lambda}}}{L_{\rho} L_{\rho'}}\right)$$
(2)

gdzie:

- heta kąt spektralny (błąd metryczny SAM),
- L_{p} długość wektora spektralnego wzorca,
- $L_{p'}^{c}$ długość modelowanego wektora z danego piksela, ρ_{λ} – współczynnik odbicia dla danej długości fali charakterystyki wzorcowej,
- ρ'_{λ} współczynnik odbicia dla danej długości fali danego piksela.

Wartość kąta spektralnego θ podaje tylko różnicę pomiędzy spektrami (danego piksela i wzorca), natomiast długość wektora podaje informację o albedo spektralnym. W przypadku klasyfikacji, gdy wartość kąta spektralnego θ znajduje się poza zadanym przez użytkownika zakresem danej klasy, dany piksel jest nieklasyfikowany (Dennison i inni, 2004).

Proces analizy podobieństwa charakterystyk spektralnych podczas klasyfikacji SAM przebiega w dwóch etapach (Kruse i inni, 1993; Schwarz, Staenz, 2001):

- określenie błędu metrycznego SAM (kąta spektralnego θ),
- zakwalifikowanie lub odrzucenie danego piksela do/z danej klasy.

W praktyce dość znaczącym problemem jest wybór obiektów wzorcowych, których charakterystyki spektralne mogłyby posłużyć do identyfikacji poszukiwanych obiektów. Znalezienie prawidłowych charakterystyk polega na stworzeniu z pomiarów referencyjnych biblioteki charakterystyk spektralnych i wykonaniu pełnej korekcji atmosferycznej obrazów lub – co jest często spotykane – pozyskaniu pikseli czystych spektralnie za pomocą wskaźnika czystości pikseli PPI (*Pixel Purity Index*).

Sztuczne sieci neuronowe (SSN)

Budowa i funkcjonowanie mózgu leżą od dawna w centrum uwagi, naukowe podstawy neurologii sięgają drugiej połowy XIX w. Głównym elementem tych zainteresowań jest zdolność człowieka do rozpoznawania i zapamiętywania obrazów i dźwięków. Ludzki mózg bez najmniejszych problemów potrafi rozpoznać te same twarze pomimo różnego oświetlenia, ekspozycji, czy procesów starzenia się. Uwaga cybernetyki koncentruje się wokół aktywności mózgu, jego sposobu analizy szczegółów obrazu (np. jak to się dzieje, że potrafimy znaleźć podobieństwo dziecka do rodzica?). Na szczególną uwagę w cybernetycznych rozważaniach zasługuje fakt, iż mózg analizuje i porównuje całe układy, np. kolor i oprawa oczu, kształt nosa, itd.

Pierwsze badania nad stworzeniem sztucznej inteligencji¹¹ wiazały się z pracami W.S. McCullocha i W. Pittsa¹². Opracowali oni model komórki nerwowej (McCulloch, Pitts, 1943) i rozwinęli koncepcję cybernetyki¹³, która legła u podstaw sztucznych sieci neuronowych. Idea ta była rozwijana w latach 1950. przez W.S. McCullocha po przejściu do Research Laboratory of Electronics, Massachusetts Institute of Technology. Przedmiotem badań był przepływ sygnałów w mózgu żaby (Lettvin i inni, 1968). Równolegle z tymi pracami D. Hebb rozwijał koncepcję dotyczącą połączeń komórek mózgowych, zapamiętywania i uczenia mózgu, w tym aktywacji połączeń pomiędzy neuronami (tzw. uczenie Hebba lub reguła Hebba). D. Hebb zauważył, że informacja może być zapamiętywana dzięki połączeniom synaptycznym (Hebb, 1949).

Operacyjne początki sztucznych sieci neuronowych sięgają roku 1951 oraz prac M. Minsky'ego, który w Massachusetts Institute of Technology współzałożył laboratorium sztucznej inteligencji i stworzył pierwszą stochastyczną sieć neuronową SNARC (*Stochastic Neural Analog Reinforcement Calculator*)¹⁴. Program ten bazował na metodzie uczenia synaps opracowanej przez Hebba (Russell, Norvig, 2003).

 12 Jest to powszechnie znana koncepcja przedstawiająca komórkę nerwową jako komórkę biologiczną składającą się z somy z wewnątrz znajdującym się jądrem (centrum komór-ki) oraz licznymi wypustkami. Długie i cienkie wypustki to aksony, które stanowią wyjście impulsu nerwowego, zaś krót-ki i cienkie to dendryty (wejścia impulsów); wypustki łączą po-szczególne komórki w sieć. Według tego modelu sygnał w sieci neuronów przepływa w jednym kierunku (od dendrytów, przez centrum obliczeniowe jądra, aż do aksonów i poprzez synapsy do kolejnej komórki sieci). Szacuje się, że sygnał elektryczny o amplitudzie około 100 mV (od -70 mV do +30 mV) pokonuje odlegość od licznych synaps dendrytów do jądra w czasie około 1 milisekundy, tam po wykonaniu przetworzeniu sygnału kierowany jest zawsze w jednym kierunku do wzgórka aksonu (Osowski, 2006).

¹³ Było to rozwinięcie prac Alana Turinga z roku 1937, który opracował koncepcję i algorytm pojęciowy (*Turing Machine*). Ze względów technicznych nie mogła ona zostać zastosowana do automatycznej symulacji procesów logicznych. Pierwszym modelem, który mógł zostać technicznie zrealizowany, był Universal Turing Machine (*Universal Machine*). Model ten został ujęty w zapis matematyczny przez A. Churcha (tzw. teoria Church-Turing). Była to pierwsza koncepcja, która pozwalała w pełni, w sposób logiczny i matematyczny przeprowadzić procesy obliczeniowe.

14 http://aima.cs.berkeley.edu/

W 1957 w Cornell Aeronautical Laboratory opracowana została koncepcja perceptronu¹⁵ (Rosenblatt, 1958). Był to pierwszy tzw. klasyfikator liniowy, czyli najprostsza sieć neuronowa bazująca na modelu komórki nerwowej McCullocha i Pittsa z grupy sieci jednowarstwowych jednokierunkowych. Przepływ sygnałów realizowany był operacyjnie za pomocą rozwiązań optomechanicznych i elektronicznych. Istotą tego modelu była możliwość klasyfikacji obrazów służąca rozpoznawaniu znaków. Symulator składał się dowolnej ilości komórek wejściowych, każda z tych komórek przekazywała sygnał do komórek (neuronów) wyjścia w postaci binarnej (1, 0), czyli możliwy był przepływ sygnału do wszystkich kolejnych neuronów lub blokowanie poszczególnych komórek. Wszystkie sygnały wejścia były sumowane na wyjściu (y) według wzoru (3) (Osowski, 1996):

$$y_i = f(\sum_{j=1}^N W_{ij} x_j) \tag{3}$$

gdzie:

- y_i sygnał wyjściowy neuronu,
- f funkcja aktywacji (w tym przypadku jest to funkcja skoku jednostkowego),
- $x_{j}\,$ sygnały wejściowe (j odpowiada liczbie kolejnych warstw wejściowych, np. liczbie kanałów obrazu),
- $N\,$ całkowita liczba neuronów wejściowych,
- W_{ij} współczynniki wagowe połączeń komórek (wagi synaptyczne – odpowiadają za kierunki i intensywność przepływu sygnału pomiędzy kolejnymi neuronami).

Schemat ten legł u podstaw budowy wiekszości symulatorów sztucznych sieci neuronowych i składał się z bloku sumującego wagowe sygnały wejściowe oraz drugiego bloku, który nieliniowo generował sygnał wyjściowy. Ciągłość funkcji nieliniowych determinuje dobór wag pomiędzy poszczególnymi neuronami, czyli ma wpływ na technikę uczenia sieci. W przypadku strategii uczenia sztucznych sieci neuronowych wyróżnia się uczenie nadzorowane (supervised learning) i uczenie nienadzorowane (unsupervised learning). W pierwszym przypadku przygotowuje się zestaw danych zawierających wektory wejściowe x, (poszczególne kanały zobrazowania) oraz pożądane wektory wyjściowe d_i (wzorce poszczególnych klas będących przedmiotem klasyfikacji). Celem sieci jest takie dopasowanie wag (połączeń pomiędzy poszczególnymi neuronami), by sygnał wyjściowy neuronu y_i był najbliższy wartości pożądanej d_i (Osowski, 2006). W przypadku gdy nie ma możliwości zapewnienia wektorów wyjściowych (brak danych referencyjnych), pozostaje zastosowanie uczenia nienadzorowanego (proceduralnie odpowiada to klasyfikacji nienadzorowanej). Dobór wag następuje według zasady konkurencyjności neuronów

¹¹ Sztuczne sieci neuronowe stanowią podzbiór tzw. sztucznej inteligencji (*Artifical Inteligence – AI*) i różnią się od innych metod przetwarzania danych tym, że w trakcie przetwarzania powstają sygnały (na podstawie danych wejściowych), które nie są prostą pochodną przetworzenia algebraicznego, lecz stanowią uogólniony, jakościowo nowy sygnał, wytworzony na podstawie procedur generalizacji wyuczonych przez algorytm przetwarzania sygnału. Uzyskane dane stanowią istotną część wyniku sieci.

¹⁵ Były to elektroniczne maszyny, konstruowane specjalnie do pierwszych sztucznych sieci neuronowych; poźniej nazwą tą objęte zostały algorytmy tworzone na komputery.

względem siebie¹⁶ lub korelacji sygnałów uczących¹⁷) (Osowski, 2006).

W prezentacji historii rozwoju sztucznych sieci neuronowych warto cofnąć się do roku 1960 – wtedy to prof. B. Widrow wraz ze swoim studentem T. Hoffem rozwinęli model ADALINE (*ADAptive LInear Neuron*), który później został przemianowany na *Adaptive Linear Element*, by w końcu dojść do powielenia i połączenia poszczególnych modułów ADALINE, co dało pakiet MA-DALINE (*Many ADALINE*). Była to pierwsza operacyjna jednowarstwowa sieć neuronowa do klasyfikacji danych obrazowych i prognozowania pogody. System ten był pierwszym oferowanym komercyjnie, zorientowanym na aplikacje radarowe, sonarowe i telekomunikacyjne. Koncepcja sieci bazowała na modelu komórki McCulloch-Pitts z liniową funkcją aktywacji oraz użyciem algorytmu najmniejszych kwadratów (4) (ADALINE¹⁸):

$$y = \sum_{j=1}^{N} W_j x_j + \theta \tag{4}$$

gdzie:

y – neuron wyjściowy,

x – neuron wejściowy,

W – waga neuronu,

N - liczba neuronów,

 θ – stała sieci.

Założenia: $x_{n+1} = 1$ oraz $w_{n+1} = \theta$

Podejścia związane z sieciami jednowarstwowymi oraz ograniczonymi możliwościami obliczeniowymi ówczesnych maszyn liczących nie przyniosły satysfakcjonujących wyników. Spotykało się to z krytyką; najgłośniej uwagi zaprezentował M. Minsky, wykazując liczne ograniczenia sieci jednowarstwowych oraz wielowarstwowych. Wykazał, że tego rodzaju sieci są przydatne tylko do klasyfikacji obiektów identyfikowalnych w postaci funkcji liniowej. Ponadto S. Papert wskazał, że rozbudowanie sieci do układu wielowarstwowego daje się matematycznie sprowadzić do sieci jednowarstwowej, co umożliwi klasyfikacje obiektów jedynie funkcjami liniowymi (Minsky, Papert, 1969). Krytyka ta w znacznym stopniu przyczyniła się do spadku finansowania prac związanymi z sieciami neuronowymi na blisko dekadę. Jednocześnie był to ostatni moment naukowych prób stworzenia sztucznego mózgu (w kolejnych latach wzrosła wiedza na temat struktury i biologicznego funkcjonowania mózgu, co zaowocowało stwierdzeniem prof. E. Pugha, że jeśli ludzki umysł byłyby wystarczająco prosty do zrozumienia go, to my bylibyśmy zbyt prymitywni, by to zrozumieć¹⁹).

Odpowiedzią na krytykę M. Minsky'ego i S. Paperta była praca doktorska *The roots of backpropagation* P. Werbosa z 1974 r. (Werbos, 1994). Autor ten po raz pierwszy rozbudował koncepcję perceptronów wielowarstwowych oraz zaproponował algorytm wstecznej propagacji nauki sieci. Problem ten został doceniony i szerzej rozbudowany dopiero w 1986 r. przez J. McCleelanda i D. Rumelharda (Rumelhart i inni, 1986; McClelland i inni, 1986). Metoda ta, ze względu na wykorzystanie jej w niniejszej pracy, zostanie szczegółowo omówiona w kolejnych rozdziałach.

Kolejne postępy prac nad sieciami neuronowymi przypadają na lata 1970. Do ważniejszych należą prace z zakresu matematyki, opublikowane przez S.I. Amari (1972) i dotyczące uczenia sieci z elementami progowymi. Miały one ogromne znaczenie dla kolejnych badań, ponieważ zostało udowodnione, że nie wszystkie elementy sieci muszą zostać pobudzone, by efektywnie działać. Dowiedziono, że poniżej pewnej wartości progowej część komórek nie przekazuje sygnału, powodując wygaszenie danej części klasyfikacji. Pozwala to prowadzić procesy analityczne tym częściom sieci, które są najbliższe zadanemu wzorcowi.

Innym przykładem wykorzystania symulatorów SSN mogą być osiągnięcia amerykańskich naukowców nad wykorzystaniem sieci neuronowych do, np., rozpoznawania mowy, czy sterowania robotami. Za przykłady moga służyć prace S. Grossberga nad siecia Avalanche do sterowania ramieniem robota oraz rozpoznawania mowy lub J. Albus, A. Pollioneze i D. Marr z Massachusetts Institute of Technology, którzy opracowali sieć Cerebellatron służącą do sterowania robotami (Hines, 2009). Na uwagę zasługuje sieć Brain State in the Box, opracowana w roku 1977 przez zespół J. Andersona (Anderson i inni, 1977; Sevrani, Abe, 2000). Był to rodzaj sieci z pamięcią asocjacyjną (skojarzeniową) z dwustronnym dostępem BAM (Bidirectional Associative Memory). Służyła do klasyfikacji obiektów o cechach nieliniowych. Pomimo że jej układ był bardzo prosty - składał się z bezpośrednich relacji wejście-wyjście, bez analiz w kolejnych iteracjach klasyfikacyjnych – sieć ta stanowi podstawę do tzw. sztucznych sieci rekurencyjnych (RNN), które w latach 1983-1986 za sprawą prac J. Hopefielda stały się podstawą symulatorów ze sprzężeniem zwrotnnym miedzy neuronami sieci²⁰. Do tego typu należą jeszcze sieci Hamminga, RTRN, Elmana (Mandic, Chambers, 2001).

Na przełomie lat 1970. i 1980. przełomowym projektem badawczym była wspomniana już pamięć asocjacyjna oraz stworzenie sieci rezonansowych do klasyfikacji bez nauczyciela (np. typu ART). Znacząca była tu praca T. Kohonena, J.A. Andersona, S. Grossberga i G. Carpentera (Carpenter, Grossberg, 1987). Pamięć asocjacyjna (skojarzeniowa) odpowiadała za wzajemne kojarzenie neuronów (Osowski, 2006).

¹⁶ Strategie WTA (*Winner Takes All*) lub WTM (*Winner Takes Most*).

¹⁷ Waga połączeń między neuronami jest wzmacniana przy stanach uaktywnienia neuronów (metody hebbowskie).

¹⁸ ADALINE opis modelu: http://davinci.newcs.uwindsor. ca/~angom/cs574/le2.pdf.

¹⁹ "If the human mind was simple enough to understand, we'd be too simple to understand it" http://www.quotationspage.com/quotes/Emerson_Pugh/

²⁰ Oznacza to, że sygnały wychodzące z warstwy ukrytej bądź wyjściowej mogą zostać przesłane do warstwy wejściowej.

Początek lat 1980. to okres gwałtownego i wielokierunkowego rozwoju sieci neuronowych. Do najważniejszych prac należą wspomniane już odkrycia Kohonena - dwuwymiarowe sieci SOM (Self Organising Maps) z roku 1982 (Kohonen, 1990) i prace J. Hopfielda nad różnymi typami i modelami sieci autoasocjacyjnych z lat 1983-86. Kolejny znaczący krok to reguła optymalizacji globalnej (symulowane wyżarzanie). Idea ta nawiązuje do wyżarzania ciał stałych, które podczas krzepniecia powinny powoli się krystalizować minimalizując lokalne naprężenia, w praktyce gwarantuje to ograniczenie powstawania minimów lokalnych, czyli miejsc narażonych na pęknięcia, przed zakończeniem całego procesu krystalizacji. Analogicznie w sieciach neuronowych daży się do ograniczenia lokalnych minimów, kończących klasyfikację, przed uzyskaniem optymalnych wyników dla całej sieci (Kirkpatrick i inni, 1983).

30

W drugiej połowie lat osiemdziesiątych ogromne ożywienie w rozwoju sieci spowodowała wspomniana już publikacja McClellanda i Rumelharta (Rumelhart i inni, 1986; McClelland i inni, 1986). Autorzy zaproponowali rozwiązania problemów, które legły u podstaw krytyki M. Minsky'ego i S. Paperta (1969).

Od końca lat 1980. zaczyna się powszechny dostęp do komputerów i błyskawiczny rozwój metod badawczych i aplikacyjnych z wykorzystaniem sztucznych sieci neuronowych. Dzieje się to za sprawą darmowego dostępu do symulatorów SNN oraz w wielu przypadkach udostępniane są kody źródłowe programów, co umożliwia samodzielne modyfikowanie i rozwijanie programów oraz po rozwoju internetu możliwości wsparcia technicznego, np. poprzez bardzo dobrze funkcjonujące grupy dyskusyjne.

Podsumowując należy stwierdzić, że sztuczne sieci neuronowe najsilniej rozwijane były w Stanach Zjednoczonych, do głównych ośrodków należy zaliczyć Massachusetts Institute of Technology, Cornell Aeronautical Laboratory, AT&T Bell Labs, uniwersytety (np. Standforda, w Bostonie, Browna, Harvarda). Także naukowcy pracujący poza Stanach Zjednoczonych wnieśli ogromny wkład do tej części nauki. Do tego grona niewątpliwie należy zaliczyć prof. Teuvo Kohonen z Helsinki University of Technology (TKK), który w latach 1982-84 był pierwszym wiceprzewodniczącym International Association for Pattern Recognition, a następnie w latach 1991-92 pierwszym przewodniczacym European Neural Network Society²¹. W wielu krajach (Japonia, Niemcy, Szwecja, Włochy) narodowe organizacje wykazują się bardzo dużą aktywnością w propagowaniu i rozwoju cybernetyki²². Polska ma także znaczący dorobek w tej dziedzinie, gdyż od początku rozwoju sztucznych sieci neuronowych Polacy zajmowali się tymi algorytmami i narzędziami (Greniewski, 1959; Lange, 1965; Gawroński, 1970; Kulikowski, 1972; Kempisty, 1973; Szostak, 1978; Morecki, Ekiel, 1979). Podobnie jak na świecie, także w Polsce od drugiej połowy lat 1980. datują się intensywne prace nad cybernetyką, algorytmami matematycznymi i sztucznymi sieciami neuronowymi. Do pionierów należy zaliczyć m.in.: R. Tadeusiewicza²³ (Jaworowski, Tadeusiewicz, 1974; Tadeusiewicz, Flasiński, 1991; Tadeusiewicz, 1993; Tadeusiewicz i inni, 2007), St. Osowskiego (Osowski, 1996; 2006), J. Korbicza (Korbicz i inni, 1994; Korbicz, Kowal, 2007; Korbicz, 2008) oraz D. Rutkowską i L. Rutkowskiego ²⁴ (Rutkowska i inni, 1997)²⁵.

Architektura sztucznych sieci neuronowych

Szacuje się, że mózg człowieka składa się z około 1011 neuronów, każdy z nich pozyskuje sygnał z synapsy i poprzez dendryty (każda komórka może mieć ich do 1000) przekazuje do centrum komórki. Tam następuje sumowanie sygnałów wejściowych i porównanie z wartościami progowymi, które są charakterystyczne dla lokalizacji i funkcji neuronu (Osowski, 2006). Jedną z większych tajemnic mózgu jest sposób łaczenia poszczególnych neuronów. Wiadomo (Tadeusiewicz, 2007), że poszczególne części mózgu cechują się różnymi schematami połączeń. Podobnie optymalnie dobrana struktura sieci neuronowych znacząco zwiększa przepływ informacji ułatwiając rozwiązanie problemu. Niemniej optymalny dobór sieci i jej parametrów jest trudnym zadaniem i wymaga doświadczenia oraz wykonania kilku prób pozwalajacych na wyłonienie najlepszego zestawu wskaźników. Próbując zoptymalizować struktury sieci neuronowych przyjmuje się w symulatorach różne sposoby połączeń poszczególnych neuronów, budując je w różne warstwy i układy. Powoduje to powstanie różnych typów sieci, w których neurony łączą się w charakterystyczny sposób dla danego typu sieci i według odpowiedniej metody doboru wag w połączeniach między neuronami. Najprostszym sposobem jest połączenie wszystkich neuronów ze wszystkimi. Owocuje to stworzeniem sieci umożliwiającej rozwiązanie szerokiego spektrum problemów, jednakże kosztem ogromnego wysiłku obliczeniowego i jest niezmiernie czasochłonne. Dlatego korzysta się z różnych architektur sieci (Osowski, 1996; 2006; Tadeusiewicz, 2007), z których najczęściej spotykane to:

- jednokierunkowe jednowarstwowe. Sieć taka składa się z neuronów ułożonych w jednej warstwie. Sygnał wejściowy dostarczany jest bezpośrednio do każdego neuronu warstwy, skąd przekazywany jest do wyjścia. Sposób doboru metody wagowania neuronów i uczenia sieci decyduje o nazwie sieci (np. sieci jednowarstwowe Kohonena, perceptron jednowarstwowy);
- jednokierunkowe wielowarstwowe. W sieci tego typu sygnał przepływa od wejścia do wyjścia poprzez wiele warstw, w tym co najmniej jedną warstwę ukrytą, która pośredniczy pomiędzy warstwą wejściową

²¹ http://www.e-nns.org.

²² Opracowano na podstawie strony internetowej IEEE Computational Intelligence Society (http://ieee-cis.org).

²³ http://www.uci.agh.edu.pl/uczelnia/tad//dorobek_naukowy.php?id=nauka.

²⁴ http://kik.pcz.pl/pracownik.php?ID=4, http://kik.pcz.pl/ pracownik.php?ID=3.

²⁵ http://kik.pcz.czest.pl/~rutkowski/publications.html.

a wyjściową. Neurony ukryte umożliwiają analizę statystyczną związków łączących poszczególne sygnały. Sieci wielowarstwowe w ujęciu matematycznym pełnia rolę aproksymacji stochastycznej funkcji wielu zmiennych, odwzorowując zmienne wejściowe w wyjściowe. Zastosowanie różnych metod przekształcania danych różnicuje sieci na kolejne grupy, np. sieci sigmoidalne, radialne (o radialnej funkcji bazowej). Specjalna grupa rozwiązań stanowią techniki wektorów podtrzymujących (SVM - Support Vector Machine)²⁶. Obejmują one grupę rozwiązań stosujących różne metody aktywacji neuronów oraz cechujących się specjalnym sposobem ich uczenia. Ten typ sieci został dalej przedstawiony dokładniej, gdyż wykorzystano go w niniejszej pracy;

- rekurencyjne, gdzie występuje sprzężenie zwrotne między warstwą wyjściową a wejściową. Polega to na tym, że sygnał dochodzący do wyjścia może zostać skierowany ponownie do wejścia sieci. Formowanie się sygnałów wyjściowych jest procesem dynamicznym nieliniowo i wynika z występowania jednostkowych operatorów opóźnienia. Ma to za zadanie doprowadzenie sieci do wyboru optymalnej konfiguracji (uzyskanie maksymalnego zysku przy minimalnych nakładach). W praktyce takie rozwiązania spotyka się przy stosunkowo prostych sieciach, które przez możliwość tworzenia "pętli" sygnałów potrafią dość szybko i skutecznie zaproponować optymalne rozwiązanie. Przykładem sieci rekurencyjnej jest sieć Hopfielda;
- samoorganizujące się. Jest to przykład sieci działającej bez nauczyciela (wzorców uczących sieć). Systemy automatycznie tworzą pewne struktury sygnałów wejściowych, bazując na maksymalnym ograniczeniu powstawania lokalnych minimów (zaniku sygnału), gdyż uporządkowanie całej sieci może mieć miejsce w przypadku samoorganizacji sieci w każdej jej części. Odwzorowanie tych struktur prowadzi do uzyskania określonych zbiorów na wyjściu z sieci (mapowanie sygnałów). Proces uczenia sieci polega na zwiększaniu wartości poszczególnych wag na połączeniach między neuronami. Efektem jest zwiększenie sygnałów pobudzających, co doprowadza do współpracy pomiędzy poszczególnymi częściami sieci tworzac konkurencyjne oddziaływania z innymi częściami sieci. W efekcie daje to dwie możliwości samoorganizacji sieci: opartej na regule asocjacji Hebba oraz współzawodnictwa według reguły Kohonena:

- rozmyte²⁷. Ten typ sieci obejmuje zestawy neuronów do analiz danych nieliniowych ciągłych (Zadeh, 1965). Jest to konkurencyjne rozwiązanie w stosunku do klasycznych sieci jednokierunkowych. Do nauki sieci wykorzystuje się metody propagacji wstecznej, samoorganizacji lub metodę tabeli przejść, która jest charakterystyczna dla tego typu sieci neuronowych. Sieć tego typu dokładnie przedstawiono w kolejnych częściach pracy, ponieważ symulator fuzzy ARTMAP, który jest kombinacją sieci rozmytej i samoorganizującej się został wykorzystany do klasyfikacji zaprezentowanych w niniejszym opracowaniu;
- specjalizowane. Nazwą tą obejmuje się modyfikacje poszczególnych rodzajów sieci do konkretnych algorytmów i zadań. Jako przykłady mogą służyć: sieć kaskadowej korelacji Fahlmana (sieć wielowarstwowa, w której wraz z uczeniem sieci na każdym etapie dokładany jest kolejny ukryty neuron; ma to na celu wybór optymalnej struktury i wytrenowania sieci), sieć Volterry (rodzaj sieci dynamicznej do nieliniowego przetwarzania danych, które są z opóźnieniem dostarczane do sieci).

Reasumując należy stwierdzić, że dostępność poszczególnych struktur sztucznych sieci neuronowych jest ogromna; w literaturze spotyka się ponad 50 różnych, dobrze opisanych rodzajów sieci. Fakt ten wynika z powszechnej dostępności kodów źródłowych poszczególnych symulatorów. Powoduje to, że każdy może modyfikować i dostosowywać wybraną sieć względem własnych potrzeb, które są bardzo zróżnicowane, także względem danych wejściowych. Generalnie sieci są dzielone na kilka grup (ryc. 9), uwzględniających:

- typ danych wejściowych (dane dyskretne i ciągłe),
- metody uczenia (nadzorowane, ze wzmocnieniem oraz nienadzorowane),
- topologie sieci (jednokierunkowe, ze sprzężeniem zwrotnym).

Na potrzeby niniejszej pracy wykorzystano 2 rodzaje sieci:

- wielowarstwowe jednokierunkowe ze wsteczną propagacją błędów (architektura takiej sieci składa się z warstwy wejściowej, kilkudziesięciu warstw ukrytych oraz warstwy wyjściowej), bazując na symulatorze SNNS (*Stuttgart Neural Network Simulator*)²⁸,
- rozmyte, bazując na pakiecie FuzzyARTMAP²⁹.

²⁶ S. Osowski (2006) przytacza opinię niektórych badaczy, że techniki podtrzymywania wektorów (SVM) nie należą do sztucznych sieci neuronowych. W ramach tych metod stosuje się oddzielnie proces klasyfikacji dla danych typu dyskretnego na wyjściu sieci (w którym dąży się do maksymalnego rozróżnienia poszczególnych klastrów wejściowych i wyjściowych) oraz aproksymacji (regresji) danych o ciągłych danych na wyjściu sieci. W systemie tym stosuje się system nagród (za poprawnie wykonaną czynność) oraz kar.

²⁷ Pojęcie zbiorów rozmytych zostało wprowadzone w 1965 r. przez L. Zadeha i dotyczyło próby algorytmicznego ujęcia pojęć, które z definicji są trudne do jednoznacznego określenia, np. pojęcia lingwistyczne (dużo, mało, dobrze, źle).

 $^{^{28}}$ Symulator SNNS jest darmowym pakietem, dostępnym ze strony http://www.ra.cs.uni-tuebingen.de/SNNS/

²⁹ Symulator sieci fuzzy ARTMAP został udostępniony przez prof. Paolo Gambę z Uniwersytetu w Pawii. Jednostka kierowana przez P. Gambę jest partnerem Katedry Geoinformatyki i Teledetekcji WGiSR UW w projekcie EU Hyper-i-net (www.hyperinet.eu) w kontrakcie nr 6 (zaawansowane algorytmy przetwarzania danych).



Ryc. 9. Klasyfikacja najważniejszych algorytmów uczenia sieci Fig. 9. Classification of the most important teaching algorithms of ANNs

Zastosowanie sieci neuronowych pozwoliło przeanalizować potencjał poszczególnych rodzajów danych wejściowych do klasyfikacji, które – ze względu na dużą liczbę kanałów oraz wysoką rozdzielczość radiometryczną (15 bitów) – stawiają dość duże wyzwanie dla procesu klasyfikacji, szczególnie z powodu wielkości zbiorów i wymogów sprzętowych.

Sieci wielowarstwowe jednokierunkowe

Sieci wielowarstwowe, nazywane także perceptronem wielowarstwowym (MLP)³⁰ bazują na nieliniowych składowych i w większości wykorzystują algorytmy wstecznej propagacji błędów do ich uczenia. Stanowią one obecnie podstawowe narzędzia w praktycznym wykorzystaniu sztucznych sieci neuronowych.

Jak już wspomniano, przepływ sygnału odbywa się w jednym kierunku, od wejścia (x_i) do wyjścia (d_j) poprzez liczne warstwy ukryte (ryc. 10). Sieci wielowarstwowe oraz algorytm wstecznej propagacji błędów jest precyzyjnie udokumentowany matematycznie; stanowi to ważny element analizy procesów oraz uzyskiwanych dokładności (Żurada i inni, 1996; Osowski, 1996; 2006; Tadeusiewicz, 2007).

Sieć składa się z warstwy wejściowej (x) zawierającej klasyfikowany obraz oraz wzorce klasyfikacyjne, według których będzie uczona sieć lub odbywać się właściwa klasyfikacja (gdy sieć jest już wytrenowana i na wejściu znajdują się tylko obrazy, które mają być klasyfikowane). W *i* warstwach ukrytych³¹ następuje właściwa klasyfikacja obrazu (liczba warstw ukrytych jest zależna od liczby warstw wejściowych i jest definiowana przez użytkownika podczas tworzenia architektury sieci przed pierwszą klasyfikacją). Proces klasyfikacji odbywa się poprzez wagowanie sygnałów na wszystkich połączeniach neuronów. Wagi połączeń w warstwie ukrytej oznaczone zostały od w_{i} do w_{i} (indeks górny 1 oznacza, że dotyczy to warstwy ukrytej). Dla warstwy wyjściowej wagi połączeń oznaczone zostały od w_1 do w_{w} . Indeks górny 2 oznacza, że wagi te dotyczą warstwy wyjściowej. Wyniki klasyfikacji przekazywane są do warstwy wyjściowej (d). Liczba neuronów warstwy wyjściowej jest zależna od liczby klasyfikowanych klas $(d_{1,i}, d_{2,i}, d_{3,i}, ..., d_{i,i})$. Każdy neuron łączący kolejne neurony oznaczony został kolorami (w przypadku połączeń niewidocznych na schemacie neuronów – przerywanymi wektorami). Funkcje aktywacji neuronów (u) mają liniowe, skokowe lub sigmoidalne właściwości (Chyliński i inni, 2009).

Całość procesu uczenia sieci polega na wypracowaniu optymalnego wagowania sygnału pomiędzy poszczególnymi neuronami w warstwie ukrytej i wyjściowej (w_{ij}^1, w_w^2) . "Optymalne" oznaczać powinno maksymalną zgodność pomiędzy sygnałem jakim cechuje się warstwa wyjściowa $(d_{1,j}, d_{2,j}, d_{3,j}, ..., d_{i,j})$ oraz sygnałem warstwy wejściowej $(x_{1,1}, x_{2,1}, x_{3,1}, ..., x_{i,1})$, czyli np. pomiędzy zadanym wzorcem a obrazem. Funkcja aktywacji neuronu warstwy ukrytej przybiera postać (5) (Osowski, 2006) i jest niezależna od sygnałów warstwy wyjściowej (6), natomiast funkcja aktywacji warstwy wyjściowej zależy od sygnałów warstw ukrytych:

$$\upsilon_i = f\left(\sum_{j=0}^N w_{ij}^1 x_j\right) \tag{5}$$

gdzie:

 $N-{\rm liczba}$ neuronów wejściowych,

x – wektory wejściowe, w – wagi sygnału.

³⁰ Multi Layer Perceptron.

³¹ Warstwami ukrytymi nazywa się warstwy sieci, do których nie ma bezpośredniego dostępu (Żurada i inni, 1996).



Ryc. 10. Standardowa architektura sieci jednokierunkowej wielowarstwowej. Wektory jednostkowe warstwy wejściowej: $\mathbf{x}_{1,1}$, $\mathbf{x}_{2,1}$, $\mathbf{x}_{3,1}$, ..., $\mathbf{x}_{N,1}$; wagi połączeń: \mathbf{w}_1 ..., \mathbf{w}_w ; neurony wyjściowe warstwy wyjściowej ($\mathbf{d}_{1,j}$, $\mathbf{d}_{2,j}$, $\mathbf{d}_{3,j}$, ..., $\mathbf{d}_{M,j}$). Indeksem górnym 1 oznaczone zostały wagi warstw ukrytych, indeksem górnym 2 – warstwy wyjściowej. Indeks dolny i oznacza kolejny neuron w danej warstwie, natomiast indeks j oznacza kolejną warstwę. Objaśnienia w tekście. Fig. 10. A standard architecture of the multi-layer perceptron. Input vectors: $x_{1,p}$, $x_{2,p}$, $x_{3,p}$, ..., $x_{N,p}$; weights of connections: w_1 ...,

Fig. 10. A standard architecture of the multi-layer perceptron. Input vectors: $x_{1,1}, x_{2,1}, x_{3,1}, ..., x_{N,1}$; weights of connections: $w_1, ..., w_w$; output vectors: $d_{1,j}, d_{2,j}, d_{3,j}, ..., d_{M,j}$. Upper index = 1 marks weights of hidden layers, upper index = 2 – output layer. Lower index = i marks a following neuron in a layer, and lower index = j - a following layer.

Sygnał wyjściowy neuronu d_{ij} z warstwy wyjściowej ma Fuzzy ARTMAP postać (4) (Osowski, 2006):

 $d_{ij} = f\left(\sum_{i=0}^{j} w_{ij}^2 v_i\right) \tag{6}$

Podczas tworzenia sieci jednym z kluczowych momentów jest definicja liczby warstw ukrytych (N_{ww}). Zbyt mała liczba ogranicza dokładność trenowania sieci, zbyt duża natomiast przedłuża procedury uczenia. Wydaje się, że optymalnym rozwiązaniem jest empiryczny dobór liczby warstw ukrytych polegający na tym, iż w pierwszych etapach klasyfikacji dobiera się mniejszą liczbę warstw. Jako przykład może służyć wzór (7) (Chyliński i inni, 2009):

$$N_{wu} = \sqrt{N_{wwe} * N_{wwy}} \tag{7}$$

gdzie:

- $N_{\scriptscriptstyle wwe}$ liczba warstw wejściowych (np. liczba kanałów obrazu do klasyfikacji),
- N_{wwy} liczba warstw wyjściowych (np. liczba klasyfikowanych klas pokrycia terenu).

Nie spotyka się większej liczby warstw ukrytych niż (8) (Kavzoglu, Mather, 2003):

$$N_{wu} = 3N_{wwe} + 1 \tag{8}$$

Sieci neuronowe ARTMAP oraz fuzzy ARTMAP należą do grupy sieci nadzorowanych (wymagają dostarczenia wzorca do klasyfikacji) i współzawodniczacych³²

czenia wzorca do klasyfikacji) i współzawodniczących³², tj. wewnątrz sieci znajdują się dwa symulatory, które równolegle klasyfikują dane i poprzez współzawodnictwo decyduje się, która sieć dostarczy wynik zbliżony najbardziej do wzorca.

Geneza tego typu sieci wiąże się z teorią Adaptive Resonance Theory (ART) opracowaną przez G.A. Carpentera oraz S. Grossberga (1987). Teoria ART składa się z kilku koncepcji (ART1, ART2, ART2-A, ART3)³³, opisujących sposób przetwarzania informacji przez mózg. W modelach tych zakłada się zarówno nadzorowany, jak i nienadzorowany sposób przepływu informacji, mających na celu analizę i rozpoznanie wzorców (Carpenter, Grossberg, 1990; Carpenter, 1997).

W roku 1991 zaproponowany został model ARTMAP – najbardziej zaawansowana sieć zawierająca 2 konkurujące ze sobą moduły ART, mające za zadanie uczyć

³² Równorzędnym typem sieci są architektury jednokierunkowe oraz rekurencyjne, czyli ze sprzężeniem zwrotnym.

³³ Podstawowym modelem jest ART (zwany też po generalizacji ART1) i jego modyfikacją jest ART2, który do analiz wykorzystuje dane o charakterze ciągłym. Model ART 2-A jest uproszczoną kontynuacją ART2 i oferuje znacznie szybsze obliczenia. Rozwinięciem tego modelu jest ART 3, który w analizie uwzględnia złożone modele zachodzące na złączu poszczególnych neuronów (symuluje funkcjonowanie neuroprzekaźnika).



Ryc. 11. Schemat funkcjonalny sieci (fuzzy) ARTMAP. ARTa – moduł odpowiadający za przetwarzanie danych wejściowych do klasyfikacji; ARTb – moduł odpowiadający za końcowe etapy klasyfikacji; dane wejściowe a – obrazy będące przedmiotem klasyfikacji; dane wzorcowe b – dane referencyjne służące uczeniu sieci; p_M – neurony wejść (odpowiadają one poszczególnym kanałom zobrazowania i wzorców wykorzystanych do uczenia sieci); STM³⁴ – pamięć krótkotrwała; LTM – pamięć długotrwała; warstwa 1³⁵ – warstwa porównująca; warstwa 2 – warstwa rozpoznająca; x_i – neurony warstwy 1; y_j – neurony warstwy 2; w_{ij} – wagi połączeń neuronów sprawdzających poprawność klasyfikacji; warstwa asocjacyjna – odpowiada za identyfikację klasyfikowanych jednostek w module ARTa ze wzorcem przetwarzanym w module ARTb; moduł tworzenia nowego wzorca³⁶ (opracowano na podstawie Carpenter, Grossberg, 2003; Dagher, 2006; Rocki, 2007; Trianni, 2007)

Fig. 11. Processing chain of the (fuzzy) ARTMAP simulator. ARTa – input processing module; ARTb – output processing module; p_M – input neurons; STM – short-term memory; LTM – long-term memory; layer 1 – comparison layer; layer 2 – recognising layer; x_i –neurons of layer 1; y_j – neurons of layer 2; w_{ij} – weights of connections of output image classification module; v_{ji} – weights of proofing neurons; match traking – module of signal comparison between input and reference units (sources: Carpenter, Grossberg, 2003; Dagher 2006; Rocki, 2007; Trianni, 2007)

³⁵ Warstwa 1 – pobiera dane wejściowe w postaci wektora sygnału i na podstawie funkcji wyboru odszukuje w warstwie 2 wzorzec najbardziej odpowiadający wektorowi wejściowemu z warstwy 1. Po odnalezieniu odpowiednich wektorów następuje klasyfikacja, a po jej zakończeniu odbywa się porównanie przez moduł kasujący z zadanym na wstępie współczynnikiem czujności ρ (prawdopodobieństwa) (ang. vigilance). Im wyższy współczynnik czujności p, tym system rozpoznaje więcej obiektów (tworzy więcej klastrów, które w efekcie dokładniej odpowiadają wzorcom), przy obniżeniu jego wartości uzyskane wyniki są bardziej zgeneralizowane. W przypadku gdy uzyskana wartość sygnału jest zgodna z zadanym współczynnikiem p, następuje trening sieci. W przypadku braku jakiegokolwiek neuronu, którego sygnał spełniłby warunki narzucone przez współczynnik p, tworzony jest nowy neuron, który reprezentuje nową klasę. Trening sieci spełnia taką samą rolę jak w sposób nadzorowany i klasyfikować dane w czasie rzeczywistym (Carpenter i inni, 1991). Ważne w tym modelu było to, że podczas treningu do obu modułów ART dopływały sygnały i poszczególne moduły ART mogły indywidualnie zmieniać swoją architekturę w odpowiedzi na napływające sygnały (ryc. 11).

Różnica pomiędzy ARTMAP i fuzzy ARTMAP polega na tym, że ta pierwsza przyjmuje wartości binarne, natomiast druga dane o wartościach ciągłych.

³⁴ STM – Short Term Memory odpowiada za szczegółowe rozpoznanie i dopasowanie sygnałów; LTM – Long Term Memory – za tworzenie i aktualizacje wag połączeń pomiędzy warstwami.

w sieciach wielowarstwowych: ma zoptymalizować połączenia neuronów, czyli zaadaptować wagi do danej klasyfikacji. Proces ten odbywa się w pamięci długotrwałej i bazuje na dwóch metodach: szybkiej (algebraiczne dopasowanie sygnału) oraz wolnej, symulującej procesy neurologiczne (w przypadku sieci stosuje się równania różniczkowe) (Rocki, 2007).

³⁶ W przypadku gdy między danymi generowanymi przez ARTa i danymi wejściowymi do sieci ARTb nie znajduje się odpowiednich sygnałów tworzona jest nowa kategoria.

Ważnym elementem systemów ART jest możliwość pracy na danych typu ciągłego, co pozwala analizować dane w sposób "rozmyty" (ang. fuzzy), czyli taki, w którym brakuje wyraźnych granic pomiędzy analizowanymi jednostkami (np. analiza pojęć cicho i głośno, ciepło – zimno) (Zadeh, 1965, 1972). Wersje fuzzy istnieja zarówno dla pojedynczych modeli ART (tzw. fuzzy ART), jak tandemów fuzzy ART pracujących w postaci ARTMAP (tzw. fuzzy ARTMAP lub FAM) (Carpenter i inni, 1991, 1992). Sieci neuronowe bazujące na zestawach "rozmytych" charakteryzuje występowanie wartości pośrednich pomiędzy wartościami całkowitymi (0, 1 - typowymi dla obrazów)³⁷. Zastosowanie sieci fuzzy ART znaczaco ogranicza liczbę pomyłek i błędów powstałych podczas analiz na każdym poziomie rozpoznania, analizy i klasyfikacji danych. Dzieje się to za sprawa dużej elastyczności systemu³⁸, automatycznie generowanej zmiennej progu uczenia³⁹, maksymalnego wykorzystania pamięci komputera oraz możliwości skalowania⁴⁰. Dużą zaletą sieci tego typu jest tworzenie list rankingowych możliwych przynależności klasyfikowanego obiektu, gdyż moduł ARTa klasyfikuje dane wedle zasad klasyfikacji nienadzorowanej, polegającej na grupowaniu wektorów $p_{\scriptscriptstyle M}$ w klastry według zadanego parametru współczynnika czujności ρ^{41} (Carpenter i inni, 1992, 1995). W efekcie

³⁷ Podobna sytuacja miała miejsce w przypadku sieci wielowarstwowej SNNS ze wsteczną propagacją błędu – tam poszczególnym wartościom jasności obrazu dopisane zostały wartości wyrażone w liczbach rzeczywistych z zakresu <0,1>.

³⁸ Sieci typu ART mogą zmieniać strukturę względem dopływających sygnałów wejściowych, cechują się dwoma systemami pamięci (długo- i krótkotrwałej), odpowiadającej pamięci człowieka. W pierwszym przypadku zachowują się ogólne informacje, które potrafią szybko odtworzyć ogólne informacje o obiekcie. Pamięć krótkotrwała zawiera bardzo wiele szczegółowych danych, które znacząco obciążają system, ale dostarczają bardzo dokładne informacje. W przypadku wstępnych procedur uczenia sieci wykorzystuje się długotrwałą pamięć, która naprowadza system na prawidłowe wartości wag połączeń neuronów, a szczegółowe ulepszenia systemu wykonuje się w pamięci krótkotrwałej. Cecha ta powoduje, że w krótkim czasie modele ART cechuje dużo większa efektywność uczenia w porównaniu do sieci wielowarstwowych ze wsteczną propagacją błędu, gdyż system powinien pamiętać wyuczone już wzorce.

³⁹ W przypadku sieci wielowarstwowych ze wsteczną propagacją błędu parametr uczenia jest stały dla całego procesu klasyfikacji.

⁴⁰ Możliwość podziału wielowymiarowych danych na mniejsze części, co znacząco zwiększa efektywność obróbki i klasyfikacji danych.

⁴¹ Współczynnik czujności ρ powinien cechować się możliwie niewielką wartością, co pozwoli na szczegółowe wydzielanie klas w sieci ARTa. Jednakże zbyt szczegółowe wyróżnienie klastrów w ARTa może spowodować, że w sprzężeniu zwrotnym z siecią referencyjną ARTb, gdzie zapisane są wzorce nadzorujące klasyfikacje w ARTMAP, warstwa asocjacyjna nie znajdzie prawidłowego sygnału zwrotnego i dane klastry nie zostaną prawidłowo zaklasyfikowane do tego, czego życzyłby sobie operator sieci. Zbyt duża wartość współczynnika ρ może spowodować zbytnią generalizację klasyfikowanej mapy, dlatego dobór współczynnika powinien się odbyć empirycznie. powoduje to automatyczne tworzenie nowych neuronów. Proces ten odbywa się w ten sposób, że w pierwszym przypadku sprawdzana jest pamięć długotrwała (LTM), jeśli ona nie wykaże obecności znanego sygnału lub poziom sygnału nie będzie akceptowalny przez zadany parametr wartości progowej współczynnika czujności ρ^{42} , to powstają nowe klasy (Rocki, 2007). W przypadku nowo uczonej sieci ARTa proces ten polega na tym, że po załadowaniu pierwszego neuronu x_1 tworzony jest pierwszy wzorzec referencyjny y_1 , jeśli po załadowaniu kolejnego neuronu x_2 podobieństwo tego sygnału znajdzie się w zakresie zdefiniowanym współczynnikiem czujności ρ , neuron zaklasyfikowany zostanie do zdefiniowanej klasy y_1 , jeśli nie, to do y_2 (ryc. 12, Carpenter i inni, 1992).



Ryc. 12. Najważniejsze etapy klasyfikacji obrazów z wykorzystaniem symulatora fuzzy ARTMAP (źródło: Rocki, 2007, zmodyfikowane)

Fig. 12. The main stages of a fuzzy ARTMAP image classification (source: Rocki, 2007, modified)

Pierwsze etapy klasyfikacji z wykorzystaniem symulatora fuzzy ARTMAP są zbliżone do klasyfikacji z wykorzystaniem symulatora SNNS i polegają na przygotowaniu danych wejściowych oraz wytrenowaniu sieci. Różnica polega na tym, że symulator fuzzy ARTMAP akceptuje dane zapisane w formacie środowiska ENVI, natomiast SNNS wymaga przygotowania zbiorów liczb

⁴² Sprzyja temu rozmycie wartości wejściowych, co jest istotne w przypadku heterogenicznych danych teledetekcyjnych (duży udział mikseli, reprezentujących różne formy pokrycia terenu).

rzeczywistych zapisanych w formacie ASCII. Różnica dotyczy jedynie formatu zapisu danych, co z jakościowego punktu widzenia klasyfikacji nie odgrywa żadnej roli, jest to tylko znaczące ułatwienie dla osoby klasyfikującej, gdyż format ENVI jest jednym z podstawowych, w których się zapisuje dane teledetekcyjne i nie wymaga żadnych dodatkowych procedur.

Po wyborze kanałów do klasyfikacji należy przygotować warstwę wzorcowa, np. form pokrycia terenu. Symulator fuzzy ARTMAP - podobnie jak SNNS - wymaga, by była to warstwa rastrowa z identyczną liczbą pikseli (kolumn oraz wierszy) jak dane obrazowe będące przedmiotem klasyfikacji. Akceptowalnym formatem danych wzorcowych jest plik *.bmp. Na tym etapie przygotowania danych różnica pomiędzy omawianymi symulatorami jest istotna, ponieważ fuzzy ARTMAP wykonuje w trakcie całej procedury klasyfikacji także ocenę dokładności, która jest podawana w postaci macierzy błędów, współczynnika kappa oraz dokładności całkowitej, natomiast symulator SNNS wymaga oddzielnej procedury określającej dokładność wykonanej klasyfikacji. Symulator SNNS oferuje natomiast możliwość bieżącego śledzenia aktualnego błędu klasyfikacji (co pozwala modyfikować dane i parametry klasyfikacji, zaś w fuzzy ARTMAP należy czekać do końca klasyfikacji). Ogromną zaletą podejścia zastosowanego w symulatorze SNNS jest to, że sieć można wytrenować na dowolnym zestawie danych, a klasyfikację wykonać na geograficznie niezwiązanych ze sobą terenach⁴³.

Po przygotowaniu zestawu klasyfikacyjnego (dane obrazowe, wzorzec oraz obraz referencyjny do określenia dokładności klasyfikacji) kolejny etap obejmuje ustalenie parametrów klasyfikacji. Są to: β_a oraz β_b (współczynniki szybkości uczenia sieci ARTa i ARTb), ρ_a ; ρ_b (próg czujności (prawdopodobieństwa)⁴⁴ sieci ARTa i ARTb) oraz parametru wyboru α (choice parametr)⁴⁵.

Reasumując: przygotowanie danych do klasyfikacji fuzzy ARTMAP jest szybsze i pozwala na wizualizacje danych na każdym etapie klasyfikacji, co umożliwia eliminację przypadkowych błędów (np. przypadkowej podmiany kanału), jednakże w symulatorze SNNS powstałe błędy – czy generalnie mówiąc, jakość klasyfikacji – można analizować w dowolnym momencie klasyfikacji poprzez analizę wartości SSE. Końcowe wyniki symulatora SNNS wymagają analizy i podjęcia decyzji, który poziom prawdopodobieństwa wystąpienia danego obiektu jest satysfakcjonujący dla użytkownika. Daje to użytkownikowi większą elastyczność pracy i zwiększa szansę wyboru właściwych wyników. System fuzzy ART-MAP przedstawia końcowe wyniki w postaci binarnej (0, 1), co znacząco upraszcza procedurę, szczególnie gdy klasyfikowana jest znacząca liczba wydzieleń. Istotnym ułatwieniem symulatora fuzzy ARTMAP jest podanie końcowych wyników dokładności klasyfikacji (w przypadku SNNS należy wykonać tę czynność dodatkowo). Znaczącą zaletą klasyfikacji SNNS jest możliwość przerywania klasyfikacji w dowolnym momencie i zmieniania parametrów, co w przypadku fuzzy ARTMAP jest niedopuszczalne.

Zastosowanie logiki rozmytej eliminuje sytuacje, w których sygnał wektora wejściowego odpowiadałby 2 różnym wzorcom (stopień przynależności sygnału do dwu klas byłby identyczny). Dzieje się to za sprawą decyzyjności (0, 1) sygnału wyjściowego względem zadanego wektora wejściowego, czyli jednoznacznego zaklasyfikowania wektora do jednej klasy, a nie jak ma to miejsce w innych typach sieci, określenie prawdopodobieństwa przynależności do każdej klasy. Takie postępowanie eliminuje nakładające się poligony z poszczególnych wydzieleń⁴⁶.

Uczenie sieci metodą wstecznej propagacji błędów

Po stworzeniu architektury sieci gotowej do klasyfikacji pierwszym krokiem jest jej wytrenowanie, czyli określenie metody doboru współczynników wagowych na połączeniu pomiędzy poszczególnymi neuronami. Istnieją trzy metody uczenia sieci: nadzorowana (do sieci wprowadzany jest wzorzec, który wskazuje jakie wartości są żądane na wyjściu), ze wzmocnieniem (systemowi podaje się ocenę jakości dopasowania wag) oraz nienadzorowana (na wyjściu nie podaje się liczby klas wyjściowych, a sieć tworzy dowolną liczbę asocjacji/klastrów pikseli).

Obecnie jednym z najbardziej efektywnych algorytmów jest metoda wstecznej propagacji błędów, która polega na opracowaniu funkcji celu $E(w)^{47}$, mającej za zadanie minimalizację różnic pomiędzy aktualnymi wartościami sygnałów (*x*) przepływających przez połączenia neuronów (synapsy) a zadanymi we wzorcu klasyfikacyjnym (d)⁴⁸ (propagowanie przesyłania błędów uczenia sieci powstałych na wyjściu do warstwy wejściowej)⁴⁹

⁴³ Symulator fuzzy ARTMAP wymaga, by wzorzec znajdował się na terenie geograficznie powiązanym z obrazem, który jest w danej procedurze klasyfikowany.

⁴⁴ Parametr ten jest modyfikowany przez sieć ARTa. Efektem tego procesu jest dynamiczna zmiana liczby kategorii wyjściowych do warstwy asocjacyjnej.

⁴⁵ Parametr ten determinuje wykasowanie, zapętlenie bądź przesłanie sygnałów neuronów pochodzących z ARTa i ARTb.

⁴⁶ Ma to także negatywne skutki, gdyż w przypadku klasyfikacji gatunków, sieć fuzzy ARTMAP nie pozwala na stworzenie poligonu, który składałby się z różnych gatunków, jeśli nie były one zdefiniowane we wzorcu. Sieć SNNS oraz ARTMAP pozwalają na stworzenie takiej mapy z niezależnych wzorców, np. występowania kosówki, borówczysk i muraw. W sieciach SNNS i ARTMAP można utworzyć kompleks tych trzech klas, natomiast w fuzzy ARTMAP są tworzone oddzielne, nienakładające się poligony.

⁴⁷ Najczęściej funkcja celu jest definiowana jako suma kwadratów różnic między neuronami wzorcowymi a aktualną wartością sygnału (Osowski, 2006).

⁴⁸ Uczenie sieci polega na określeniu wag poszczególnych połączeń neuronów (we wszystkich warstwach sieci) w taki sposób, by wektory wejściowe *x* na wyjściu *y* uzyskiwały wartości maksymalnie zbliżone do wzorców *d* (Werbos, 1994, Żurada i inni, 1996).

⁴⁹ Istnieje szereg metod określających dokładność dopasowania, a tym samym uzyskiwany błąd uczenia sieci.

(Werbos, 1994, Kavzoglu, Mather, 2003; Osowski, 2006). Funkcję celu definiuje się za pomocą wzoru (9):

$$E(w) = \frac{1}{2} \sum_{j=1}^{p} \sum_{k=1}^{M} \left(y_k^j - d_k^j \right)^2$$
(9)

gdzie:

- $j = 1, 2, 3, \dots p$ numer kolejnych próbek uczących (wzorcowych),
- $k = 1, 2, 3, \dots M$ numer kolejnych wartości wyjściowych,
- y wartość wyjściowa,
- d wartość zadana (wzorcowa), która jest klasyfikowana.

Uczenie sieci przebiega według następującego algorytmu (Żurada i inni, 1996; Osowski, 2006):

- analiza wartości neuronów wejściowych x,
- obliczenie wartości sygnałów wyjściowych warstwy ukrytej w¹,
- obliczenie wartości sygnałów warstwy wyjściowej y_i.
 Dla wielu neuronów odbywa się to według wzoru (10) (Osowski, 2006),
- obliczenie aktualnej wartości funkcji celu (*E (w)*).
 W kolejnych iteracjach klasyfikacji wartość (*E(w)*)
 powinna maleć. Ciągłość procesu pozwala na zastosowanie gradientowych algorytmów optymalizacji.

$$y_{k} = f(\sum_{i=0}^{K} w_{ki}^{2} f(\sum_{j=0}^{N} w_{ij}^{1} x_{j}))$$
(10)

gdzie:

- ¹ indeks górny 1 mają warstwy ukryte,
- ² indeks górny 2 mają warstwy wyjściowe,
- K końcowa liczba neuronów warstwy ukrytej,
- M końcowa liczba neuronów warstwy wyjściowej,
- N końcowa liczba neuronów wejściowych,
- x neurony warstwy wejściowej,
- y neurony warstwy wyjściowej,
- d neurony warstwy zadanej (wzorcowej),
- w wagi połączeń pomiędzy neuronami,
- i, j kolejne neurony.

Gradientowe metody optymalizacji należą do najskuteczniejszych rozwiązań podczas uczenia sieci (Żurada i inni, 1996). Proces ten odbywa się według wzoru (11) (Osowski, 2006):

$$w(k+1) = w(k) + \eta p(w)$$
 (11)

gdzie:

w – wagi połączeń pomiędzy neuronami,

- k neurony warstwy ukrytej,
- η współczynnik uczenia,
- p(w) kierunek minimalizacji w przestrzeni wielowymiarowej w.

Współczynnik uczenia (ŋ) dobierany jest empirycznie. Zbyt duża jego wartość (np. 5) może doprowadzić do przeregulowania sieci i dużego końcowego błędu klasyfikacji, gdyż wywołuje znaczące oscylacje. Natomiast zbyt mała wartość współczynnika (np. 0, 1) eliminuje oscylacje, powodując jednakże powolny spadek błędu uczenia (*Sum Square Error*) i znacząco wydłużając proces uczenia. Aby osiągnąć optymalny współczynnik uczenia (zadany poziom błędu), należy wykonać kilka prób klasyfikacji na niewielkiej liczbie iteracji.

Wyznaczenie kierunku minimalizacji p(w) opiera się na analizie wszystkich wag danej iteracji, co pozwala na określenie gradientu wag i wyznaczenie ich wektora⁵⁰. Proces ten nazywa się algorytmem wstecznej propagacji. Najważniejsze etapy tej procedury to (Osowski, 2006):

- analiza sieci. Zakłada się, że sygnały przepływają od wejścia do wyjścia, pozwala to uzyskać wartość chwilową sygnału wyjścia warstwy ukrytej i wyjściowej z sygnałów wejścia (równym sygnałom neuronów wejścia *x*),
- utworzenie sieci propagacji wstecznej. Etap ten jest inicjowany po uzyskaniu sygnałów wyjścia i polega na odwróceniu kierunku przepływu sygnałów. Podczas odwrócenia następuje porównanie sygnału, który ma być kierowany ku wejściu z sygnałem warstw zadanych (wzorcowych). W tym etapie następuje zastąpienie funkcji aktywacji przez ich pochodne (*m* oznacza liczbę warstw sieci) (12) (Osowski, 2006); pozwala to obliczyć różnice pomiędzy wzorcem a wartością aktualną sygnału podczas analizy wstecznego przepływu sygnału:

$$\frac{df(u_i^1)}{du_i^1}, \frac{df(u_i^2)}{du_i^2}, \dots, \frac{df(u_i^m)}{du_i^m}$$
(12)

modyfikacje wag przepływu sygnału przez synapsy sieci (uczenie sieci). Odbywa się to zgodnie z przedstawioną powyżej procedurą (przepływ sygnału od wejścia ku wyjściu i odwrotnie – propagacja wsteczna). Proces ten zachodzi w sposób ciągły, zostanie zatrzymany po wykonaniu zadanej liczby iteracji lub osiągnięciu zaprogramowanej przez użytkownika wartości błędu *ɛ* (SSE⁵¹). Podstawy teoretyczne i matematyczne zapisy tych procesów są dokładnie opisane w literaturze (Werbos, 1994; Osowski, 1996, 2006; Żurada i inni, 1996).

Najpowszechniej stosowane mierniki to: średni błąd kwadratowy (*mean squared error*), suma błędów kwadratowych (*sum squared error*) i średni błąd bezwzględny (*mean absolute error*).

⁵⁰ Proces ten nie dotyczy warstwy wyjściowej.

⁵¹ Sum Squared Error.