
**XIV Seminarium
NIENISZCZĄCE BADANIA MATERIAŁÓW
Zakopane, 4-7 marca 2008**

**SZTUCZNE SIECI NEURONOWE. MOŻLIWOŚCI ZASTOSOWANIA W
BADANIACH NIENISZCZĄCYCH**

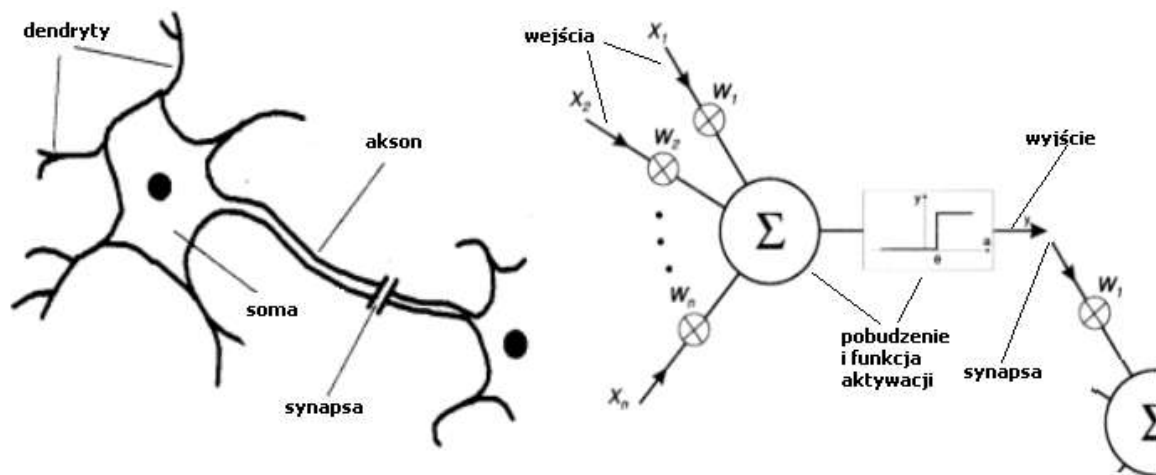
Piotr Gołabek
Politechnika Radomska, Radom, pgolab@poczta.onet.pl

1. Uwagi wstępne

Sztuczne Sieci Neuronowe (SSN) są dziedziną badawczą o bardzo bogatym obszarze aplikacyjnym, która rozwijana jest konsekwentnie od ponad 60 lat. Wyłoniła się ona na gruncie odwiecznego zainteresowania człowieka charakterem jego własnych procesów myślowych. Pozostając zawsze bardzo blisko inspiracji neurobiologicznych czy też psychologicznych, badania nad SSN wprowadziły szereg ścisłych, formalnych modeli obliczeniowych, które z jednej strony odnoszą się bezpośrednio do biologicznego pierwowzoru – mózgu, a z drugiej stanowią bardzo efektywne narzędzie rozwiązywania trudnych, niealgorytmicznych problemów. Współczesny obszar zastosowań SSN jest bardzo bogaty i obejmuje m.in. rozpoznawanie wzorców w sygnałach różnego rodzaju (serie czasowe, obrazy wizyjne), klasyfikację danych wielowymiarowych (grupowanie danych, wyłanianie kategorii, wspomaganie procesów decyzyjnych), aproksymację funkcji (modelowanie systemów typu *black-box*, adaptacyjne sterowanie i regulację), adaptacyjną filtrację (przetwarzanie sygnałów, obrazów), redukcję danych wielowymiarowych (kompresję danych, wykrywanie najbardziej znaczących komponentów), itd. Trudno nie zauważać narastających chronologicznie coraz ściślejszych związków SSN z konceptami wypracowanymi w ramach innych dziedzin: statystyki, teorii przetwarzania sygnałów czy też badań nad nieliniowymi układami dynamicznymi. Jednak zdecydowanie zachowana została koncepcyjna odrębność SSN, wynikająca z mocnego ich osadzenia w obszarze Sztucznej Inteligencji, a bardziej specyficznie, Inteligencji Obliczeniowej. W niniejszym artykule przedstawione zostaną najbardziej charakterystyczne koncepcje tej bogatej dziedziny wraz z jej zastosowaniami do rozwiązywania problemów Badań Nieniszczących.

2. Matematyczny model neuronu

Charakterystyczne, bazowe koncepcje dziedziny najłatwiej poznać, śledząc jej ewolucję w czasie. Dla SSN przełomową datą był rok 1943, w którym neurobiolog Warren McCulloch i matematyk Walter Pitts opublikowali artykuł, proponujący matematyczny model neuronu - podstawowego elementu budulcowego mózgu [1]. Zachowana została uproszczona analogia morfologiczna i funkcjonalna (rys. 1). W zaproponowanym modelu sygnały wprowadzone na wejścia neuronu („dendryty”) ulegały pomnożeniu przez tzw. wagi połączeń (odpowiednik przewodności elektrochemicznej dendrytów) i po zsumowaniu tworzyły tzw. pobudzenie neuronu. Schodkowa funkcji aktywacji (funkcjonalnie – „soma”) wywoływała przy odpowiednio silnym pobudzeniu aktywację wyjścia neuronu („aksonu”) i przekazanie bodźca (przez „synapsę”) na wejście następnego neuronu.



Rys. 1 Model McCullocha-Pittsa

Funkcjonalność neuronu McCullocha-Pittsa można wyrazić związłym modelem matematycznym:

$$z = \sum_i x_i w_i \quad y = \begin{cases} -1, & z < \theta \\ 1, & z \geq \theta \end{cases}, \quad (1)$$

przy czym x_i to wejścia neuronu, w_i to wagi poszczególnych wejść, z to pobudzenie, y to wyjście (aktywacja) neuronu, a θ to próg aktywacji.

Ten uproszczony model służył udowodnieniu jednej tezy – neuron, przy odpowiednim doborze wag i progu aktywacji może realizować elementarną funkcję logiczną algebry Boole’a (AND, OR, NOT), a złożenie neuronów może zrealizować proces obliczeniowy (względnie implikacyjny proces wnioskowania). Np. dwuwejściowy neuron McCullocha-Pittsa z wagami równymi jeden i progiem aktywacji 1 realizuje funkcję logiczną OR. W ten sposób położony został most pomiędzy mózgiem i jego hipotetycznym działaniem, a formalnym modelem obliczeniowym. Dalszy rozwój dziedziny zajmował się doskonaleniem takich modeli, tak aby w sztucznym systemie reprodukować procesy obliczeniowe, wnioskowania czy też decyzyjne, podobne do tych, do jakich jest zdolny mózg.

3. Reguły uczenia

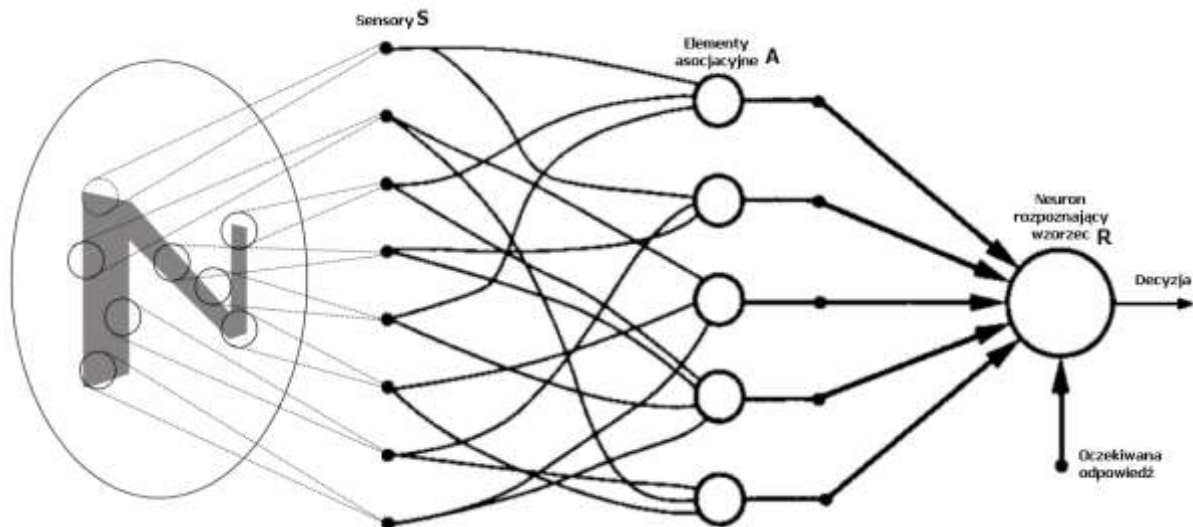
Dobór wag w modelu McCullocha-Pittsa był arbitralny. Jednak już w 1949 roku neurofizjolog Donald Hebb sformułował postulat dotyczący podstawowej neurofizjologicznej zasady działania mózgu: przepustowość połączeń synaptycznych zmienia się cały czas, w zależności od ich aktywności (im bardziej aktywna synapsa, tym bardziej się wzmacnia). Innymi słowy – mózg się zmienia, uczy. Ta korelacyjna reguła przeniosła aktywność badawczą w stronę poszukiwania formalnych reguł adaptacji dla obliczeniowych modeli neuronu. Pierwsza z nich została nazwana regułą Hebba [2]. Formułowała zmianę wagi neuronu proporcjonalną do iloczynu (korelacji) wejścia skojarzonego z tą wagą oraz wyjścia neuronu:

$$\Delta w_i = \mu x_i y, \quad (2)$$

gdzie μ to tzw. szybkość uczenia. W 1954 roku Dennis Gabor zaproponował strukturę „uczącego się filtru”, w którym wagi zmieniały się tak, aby zminimalizować błąd między faktyczną a oczekiwaną wartością wyjścia. Zasada uczenia została nazwana regułą największego spadku. Reguły Hebba i Gabora są właściwe dla dwóch komplementarnych trybów uczenia – nienadzorowanego oraz nadzorowanego.

Sztuczny neuron mógł się więc uczyć. W 1958 roku psycholog Frank Rosenblatt [3] opisał eksperyment demonstrujący użyteczne zastosowanie reguły uczenia. Elektromechaniczna struktura, którą nazwał perceptronem (rys. 2), o zachowaniu zgodnym z modelem

McCullocha-Pittsa, była zdolna w drodze repetetywnego stosowania reguły uczenia podobnej do reguły Hebba nauczyć się rozróżniać proste obrazy wizyjne. Struktura składa się z warstwy fotosensorów (elementy S), tworzącej „siatkówkę” perceptronu. Wyjścia S podłączone są do warstwy tzw. elementów asocjacyjnych (A), prostych dyskryminatorów progowych, zamieniających ciągły sygnał z sensorów na logiczne wyjście $\{0,1\}$. Zrealizowana jest tylko bardzo mała część ze wszystkich możliwych połączeń między S a A, przy czym wybór połączeń jest losowy. Wyjścia A są podłączone do wejść neuronu McCullocha-Pittsa, którego wagi połączeń mogą być zmieniane. Jako, że połączenia między S a A nie zmieniają się w czasie eksperymentu, struktura efektywnie sprowadza się do jednego neuronu ze wstępnym uformowaniem danych wejściowych. Celem struktury jest nauczenie się realizowania dychotomii – rozróżniania między dwoma klasami obrazów.



Rys. 2 Perceptron Rosenblatta

Na siatkówkę S podawana jest cyklicznie seria obrazów, przy czym każdemu obrazowi towarzyszy informacja do której z dwóch klas należy. Algorytm kontrolny sprawdza, czy ta oczekiwana odpowiedź jest zgodna z faktycznym wyjściem neuronu R. Jeśli nie, dokonywana jest drobna zmiana wag wejść R. Jeśli oczekiwana odpowiedź to -1 , a faktyczna to 1 , wagi są zmniejszane, w przeciwnym wypadku – zwiększane. Wyraża to tzw. perceptronowa reguła uczenia (bardzo podobna do reguły Hebba):

$$\Delta w_i = \mu x_i (d - y), \quad (3)$$

gdzie d to oczekiwana wartość wyjścia.

Ten prosty schemat procesu uczenia, powtarzany przez pewien czas prowadzi do takiego ustalenia się wag neuronu R, że poprawnie klasyfikuje on wszystkie wzorce. Rosenblatt modyfikował swój eksperyment, wyposażając strukturę w wiele neuronów R, a zatem umożliwiając rozpoznanie więcej niż tylko dwóch klas. Zaproponował też inne możliwe realizacje warstwy sensorowej, np. bank filtrów sygnału elektrycznego, nastrojonych na różne częstotliwości czy też dokonujących dyskryminacji czasowo-amplitudowej. W takim ujęciu perceptron mógł rozpoznawać np. charakterystyczne sygnatury w sygnale echa.

Opisany eksperyment obrazuje jedną z podstawowych strategii rozwiązywania problemów przy użyciu SSN, tzw. uczenie nadzorowane. Wiedza o tym, jak rozwiązać problem (rozróżnić dwie klasy obrazów) nabywana jest w drodze prezentacji maszynie uczącej się (perceptronowi) przykładów poprawnie przeprowadzonego przypisania. Obraz wraz z prawidłową (oczekiwaną) odpowiedzią stanowi tzw. parę trenującą. Neuronowi przedstawiane jest sukcesywnie wiele takich par. Tworzą one tzw. zbiór trenujący. Co więcej, zbiór ten jest prezentowany wielokrotnie. Jednokrotna prezentacja to tzw. epoka uczenia.

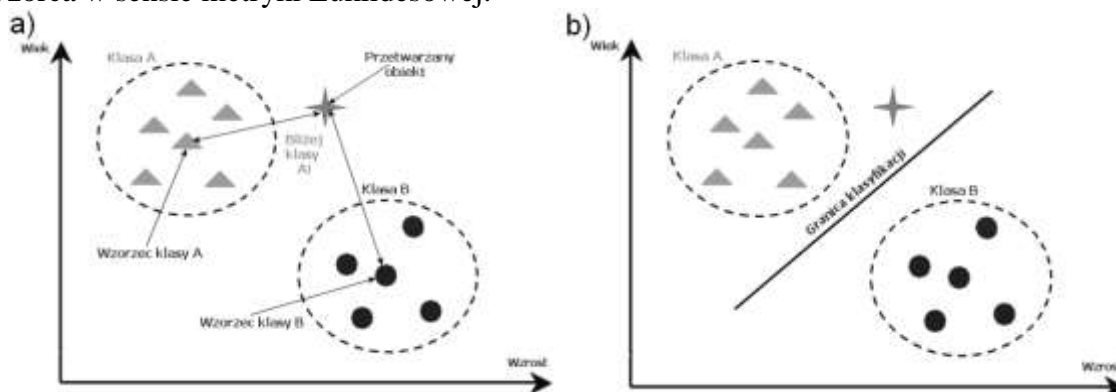
Proces uczenia wymaga wielu takich epok, aż do osiągnięcia stanu wytrenowania. Zachodzące w tym czasie zmiany parametrów maszyny (wag) dążą do zmniejszenia błędu między oczekiwaną a faktyczną wartością na wyjściu. Gwarantuje to odpowiednio sformułowana reguła uczenia.

4. Zagadnienie klasyfikacji

Eksperyment Rosenblatta jest również przykładem realizacji jednego z fundamentalnych problemów inteligencji obliczeniowej, mianowicie zadania klasyfikacji, lub rozpoznawania wzorców (ang. *pattern recognition*). Obiekt (obraz, dana) podlegający przetwarzaniu ma być zaliczony do jednej z predefiniowanych klas obiektów, co w praktyce oznacza skojarzenie go z jednym ze wzorcowych obiektów, reprezentujących pewną klasę, do którego jest najbardziej podobny. Klasyfikacją jest więc np. diagnoza medyczna stawiana poprzez porównanie wyników testów diagnostycznych do „wzorcowego” obrazu klinicznego danej jednostki chorobowej. Albo może nią być określenie, jaką literę alfabetu przedstawia fragment obrazu poprzez porównanie go z wzorcowymi szablonami liter. Jest nią także np. werdykt, czy w badanym obiekcie występuje ukryta wada na podstawie porównania spektrum odebranego sygnału ultradźwiękowego ze spektrum sygnału obiektu bez wad. Do przeprowadzenia klasyfikacji potrzebne jest określenie miary podobieństwa dwóch obiektów, tzw. metryki. Przetwarzany obiekt jest reprezentowany przez ciąg liczb, tzw. atrybutów. Mogą to być wyniki testów diagnostycznych, kolory kolejnych pikseli z ustalonego fragmentu obrazu, czy też amplitudy kolejnych harmonicznych spektrum. Na rys. 2 widać, że dla decyzyjnego neuronu R strumień danych o przetwarzanym obrazie ogranicza się do 5 wartości podawanych na jego wejścia z elementów asocjacyjnych A. Oznacza to, że każdy obraz jest w tym wypadku reprezentowany przez 5 atrybutów. Atrybuty obiektu są traktowane jako jego współrzędne w wielowymiarowej tzw. przestrzeni cech (sam obiekt jest wektorem tej przestrzeni). Najnaturalniejszą metryką określającą odległość dwóch takich n -atrybutowych obiektów jest metryka Euklidesa (odległość w linii prostej):

$$d = \sqrt{\sum_{i=1}^n (x_{1i} - x_{2i})^2}, \quad (4)$$

gdzie x_{1i} , x_{2i} to kolejne atrybuty/współrzędne, odpowiednio, obiektu \mathbf{x}_1 oraz \mathbf{x}_2 . W praktyce klasyfikacyjnej stosuje się bardziej wyrafinowane metryki, zazwyczaj wyposażone w dodatkowy komponent, reprezentujący wstępną, aprioryczną wiedzę o charakterze przetwarzanego zadania (np. taką, że występowanie pewnego wzorca jest w praktyce bardzo mało prawdopodobne). Na rys. 3a zilustrowana jest koncepcja klasyfikacji. Osie wykresu to dwa atrybuty opisujące przetwarzany obraz. Różne wzorce naniesione na wykres tworzą dwa skupiska, dwie klasy. W każdej klasie można wyznaczyć najbardziej reprezentatywny obiekt – wzorec klasy. Przetwarzany obiekt został zaliczony do klasy A, ponieważ jest bliżej jej wzorca w sensie metryki Euklidesowej.

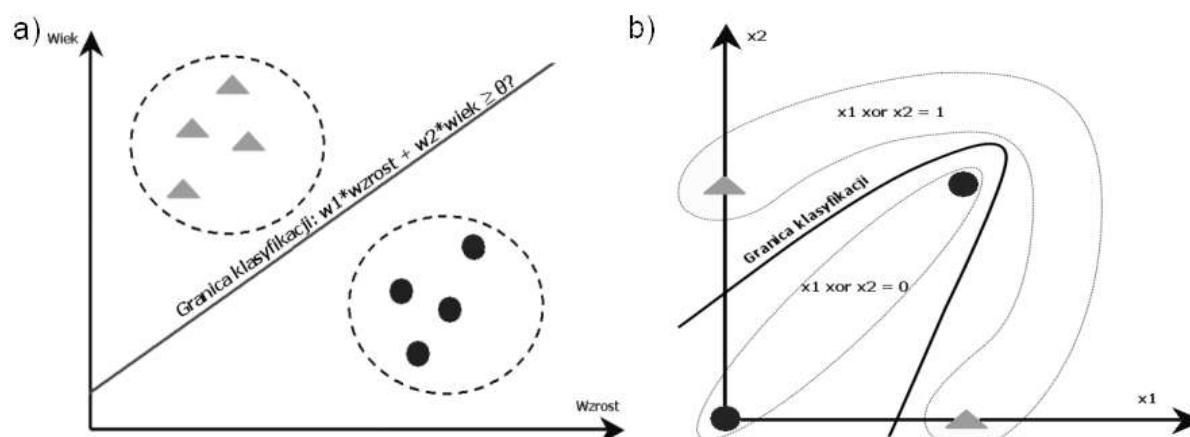


Rys. 3 Ilustracja zasad klasyfikacji

Innym spojrzeniem na ten sam proces jest wytyczenie granicy separującej najlepiej w sensie statystycznym dwie klasy w przestrzeni cech i określenie, po której stronie granicy leży obiekt – rys. 3b.

Perceptron Rosenblatta realizuje takie właśnie rozgraniczenie dwóch klas. Dychotomia wynika bezpośrednio z faktu, że wyjście neuronu R jest dwustanowe (-1 lub 1). Wartość na wyjściu zależy od tego, czy pobudzenie neuronu przekracza próg aktywacji. Pobudzenie jest liniową kombinacją wejść (p. wzór 1). Aktywacja neuronu jest więc informacją o spełnieniu liniowej nierówności. Jest to zilustrowane na rys. 4a. Separacja między klasami jest możliwa za pomocą granicy liniowej – prostej (w przypadku większej liczby atrybutów mówimy o granicy w postaci płaszczyzny lub hiperpłaszczyzny). Co więcej, położenie granicy między klasami dobierane jest automatycznie w trakcie uczenia perceptronu.

W 1969 roku Seymour Papert i Marvin Minsky wykazali znaczące ograniczenie perceptronu. Nie było możliwe za jego pomocą rozgraniczenie klas, których nie da się rozseparować liniowo. Prosty przykład takiego problemu to funkcja XOR (ang. *exclusive or* – suma rozłączna). Jak to widać na rys. 4b nie da się liniowo rozseparować klasy obiektów, dla których funkcja XOR ma wartość 0 od klasy obiektów dla których funkcja ma wartość 1. Ten trywialny przykład był zaledwie czubkiem góry lodowej. Perceptron Rosenblatta i wszystkie pokrewne rozwiązania wypracowane w latach 60-tych nie były w stanie wypracowywać koniecznej dla trudnych (czyt. rzeczywistych) problemów separacji nieliniowej.



Rys. 4 Liniowa i nieliniowa separacja klas

Drugim problemem badawczym był brak dobrych algorytmów uczenia dla sieci złożonych z wielu neuronów. Krokiem w tę stronę były struktury ADALINE (ang. *ADAPtive LInear NEuron*) oraz MADALINE (ang. *Many ADALINEs*) zaproponowana przez Bernarda Widrowa. ADALINE była właściwie równoważna perceptronowi Rosenblatta, ale wprowadzała trochę inną regułę uczenia, tzw. μ -LMS [4], opartą na minimalizacji średniokwadratowego błędu między oczekiwaną a faktyczną wartością wyjścia:

$$E = \frac{1}{2} \sum_p \left(d_p - y_p \right)^2, \quad (5)$$

przy czym wyjście y było próbkowane jeszcze przed progowaniem (czyli było równe pobudzeniu z). Sumowanie po indeksie p oznacza tu uśrednienie po wszystkich parach z zbioru trenującego. Występująca we wzorze wartość wyjściowa y oczywiście zależy od wag neuronu. Z tego powodu wyrażenie (5) określane jest mianem funkcji błędu. Sformułowany w ten sposób błąd średniokwadratowy (MSE - ang. *Mean Square Error*) poddawany jest klasycznej minimalizacji, tj. poszukiwaniu miejsca (zestawu wag), dla którego pochodna MSE względem wag jest równa zero (warunek ekstremum). Skutkuje to tzw. gradientową

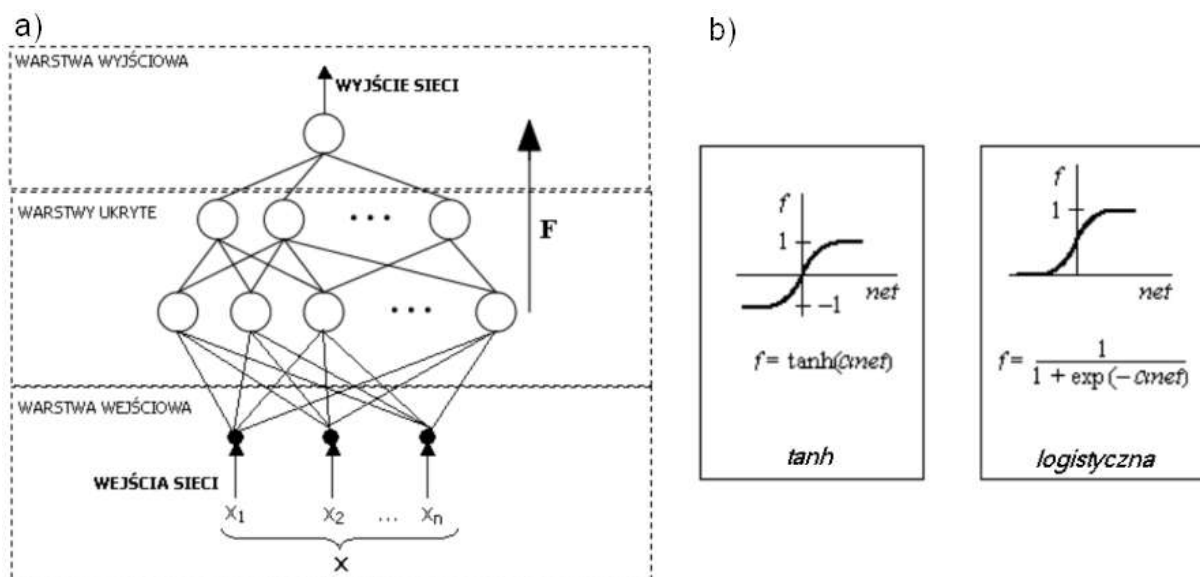
regułą uczenia, w której wagi są zmieniane zgodnie z kierunkiem gradientu MSE, tzn. w kierunku największego spadku wartości MSE (często obrazowo określane jako reguła schodzenia z gór):

$$\Delta w_i = \mu \sum_p \delta_p - y_p \underline{x}_i \quad (6)$$

Warto zwrócić uwagę na podobieństwo do reguły perceptronowej (wzór 3). W stosunku do niej następuje tu uśrednienie błędu po wszystkich parach trenujących. Zabieg ten nazywany jest często uczeniem z gradientem deterministycznym, tj. z ograniczonymi losowymi fluktuacjami gradientu od pojedynczych, zaburzonych par trenujących. Według tej nomenklatury reguła Rosenblatta to uczenie z gradientem stochastycznym. Struktura MADALINE stanowiła kaskadowe połączenie struktur ADALINE. Problem algorytmiczny polegał na sposobie wyznaczenia gradientu MSE względem wag z głębszych warstw. Te dwie przeszkody – liniowość rozwiązań i brak dobrego algorytmu uczenia spowalniało w dużym stopniu rozwój dziedziny aż do lat 80-tych.

5. Sieci wielowarstwowe

W 1986 roku Rumelhart, Hinton i Williams opublikowali algorytm wstecznej propagacji błędu (BP z ang. *error BackPropagation*), rozwiązujący opisane powyżej problemy. Z czasem okazało się, że analogiczne sformułowanie zostało opublikowane już w 1974 roku, ale przeszło niezauważone. Algorytm tzw. warstwowych sieci neuronowych. Najogólniej sieć neuronowa to dowolne połączenie neuronów. Jeśli w strukturze występują sprzężenia zwrotne, tzn. jeśli wyjście choć jednego neuronu jest podłączone bezpośrednio lub pośrednio do jego wejścia, mamy do czynienia z tzw. sieciami rekurencyjnymi. Przy braku sprzężeń zwrotnych wprowadza się zazwyczaj dodatkowe, porządkujące kryterium, tj. pogrupowanie neuronów w tzw. warstwy. Wymaga się najczęściej aby wyjścia pewnej warstwy były połączone tylko z wejściami warstwy następnej. Taka struktura, warstwowa sieć z przetwarzaniem w przód jest niekwestionowanie jedną z najpopularniejszych i najczęściej wykorzystywanych. Przedstawiona została na rys. 5a. Wyjście każdego neuronu warstwy i jest połączone z jednym wejściem każdego neuronu warstwy $i+1$.



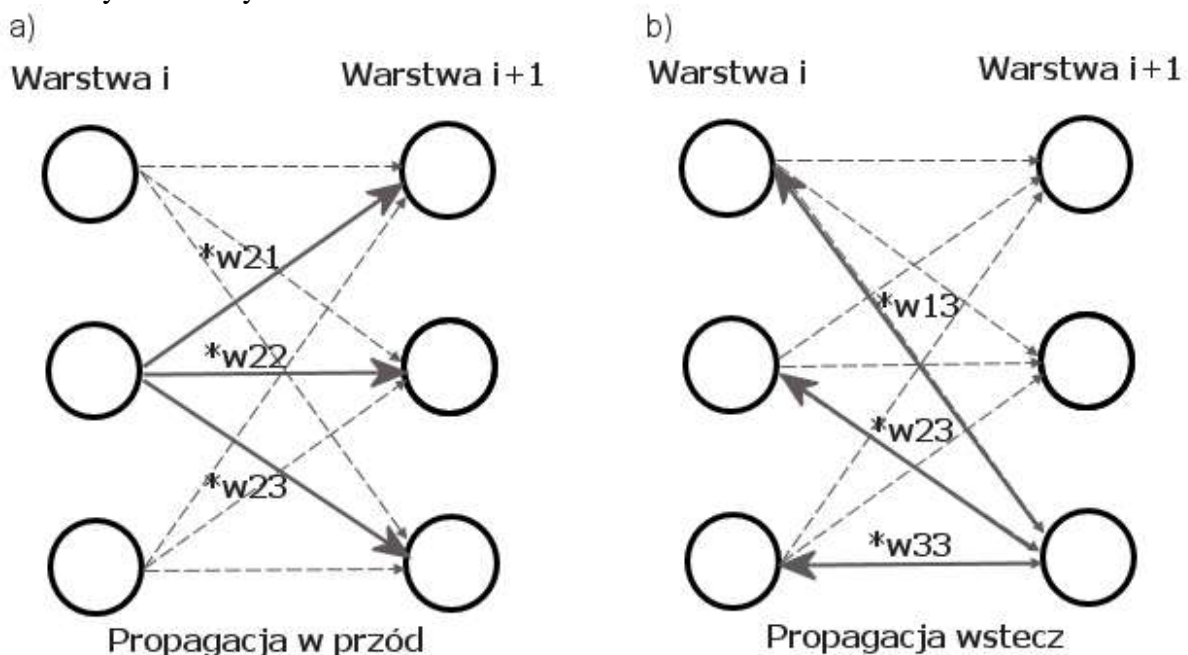
Rys. 5 Struktura sieci warstwowej. Sigmoidalne funkcje aktywacyjne

ADALINE upraszczała zagadnienie obliczenia gradientu funkcji błędu (MSE) poprzez wprowadzenie wyjścia z części liniowej neuronu (przed progowaniem). Zabieg ten brał się

stąd, że schodkowa funkcja aktywacyjna z modelu McCullocha-Pittsa nie jest różniczkowalna. Algorytm BP wymaga stosowania nieco innych, różniczkowalnych funkcji aktywacyjnych. Zbiorczo nazywa się je, ze względu na kształt, funkcjami sigmoidalnymi. Przykłady dwóch takich funkcji pokazane są na rys. 5b. Przy ich zastosowaniu reguła LMS dla wag warstwy wyjściowej przybiera zmodyfikowaną postać (por. wz. 6):

$$\Delta w_i = \mu \sum_p f'(z_p) (d_p - y_p) x_j, \quad (7)$$

gdzie $f'(z_p)$ to wartość pochodnej funkcji aktywacyjnej. Algorytm BP rozwiązuje problem wyliczenia gradientu funkcji błędu dla głębszych warstw sieci. W powyższym wyrażeniu $f'(z_p)(d_p - y_p)$ powstaje w wyniku przeniesienia błędu obliczonego na wyjściu sieci (czynniki $(d_p - y_p)$) na stronę wejść neuronów ostatniej warstwy (przeskalowanie przez pochodną $f'(z_p)$). Pytanie, jak przepropagować ten błąd głębiej, poprzez połączenia międzywarstwowe, na wyjścia neuronów warstwy przedostatniej, drugiej od końca, itd. Dokładne wyprowadzenia algorytmu obejmują używanie reguł różniczkowania funkcji wielu zmiennych. Nie ma potrzeby przytaczania ich tutaj. Wystarczy intuicyjnie powiedzieć, że propagacja błędu wstecz obejmuje określenie dla każdego neuronu i warstwy poprzedniej, w jakim stopniu jest on odpowiedzialny za błędy powstałe na wejściu neuronu j w warstwie następnej. Miarą tej odpowiedzialności jest waga połączenia i - j , czyli w_{ij} . Liczenie błędów na wyjściach warstwy poprzedniej staje się propagacją błędów poprzez połączenia w_{ij} w sposób całkowicie symetryczny do tego, jak propagują się w przód pobudzenia neuronów. Jest to zilustrowane schematycznie na rys. 6.



Rys. 6 Symetria algorytmu wstecznej propagacji błędu

Elegancja i efektywność algorytmu BP polega właśnie na tej symetrii. Propagacja w przód przez warstwę polega na przemnożeniu wektora wyjść warstwy poprzedniej przez macierz \mathbf{W} wag połączeń międzywarstwowych, natomiast propagacja wstecz polega na przemnożeniu wektora błędu przez transponowaną macierz \mathbf{W}^T .

Algorytm BP określa, jak obliczyć gradient funkcji błędu względem wag w dowolnym miejscu sieci, jednak to, jak z tego gradientu skorzystać, rozszczepia ten algorytm na całą rodzinę pochodnych, zmodyfikowanych metod uczenia. Należy pamiętać o tym, że w całym tym procesie chodzi o minimalizację błędu na podstawie lokalnej informacji gradientowej. Procedura taka jest narażona na szereg niebezpieczeństw, typu utknięcie w płytkim minimum

lokalnym, oscylacje wokół minimum, wolna zbieżność w obszarach plateau, itp. Najróżniejsze warianty BP usiłują zabezpieczyć się przed takimi problemami. Np. metoda *momentum* dodaje do reguły uczenia składnik „bezwładnościowy”, usiłujący podtrzymać obrany w poprzedniej epoce uczenia kierunek zmian wag. Taki zabieg „rozpędza” proces w obszarach plateau, i pomaga wyrwać się z płytkich lokalnych minimum.

Algorytm BP pozostaje powszechnie stosowanym, najpopularniejszym algorytmem uczenia sieci neuronowych.

6. Uwagi dotyczące stosowania SSN

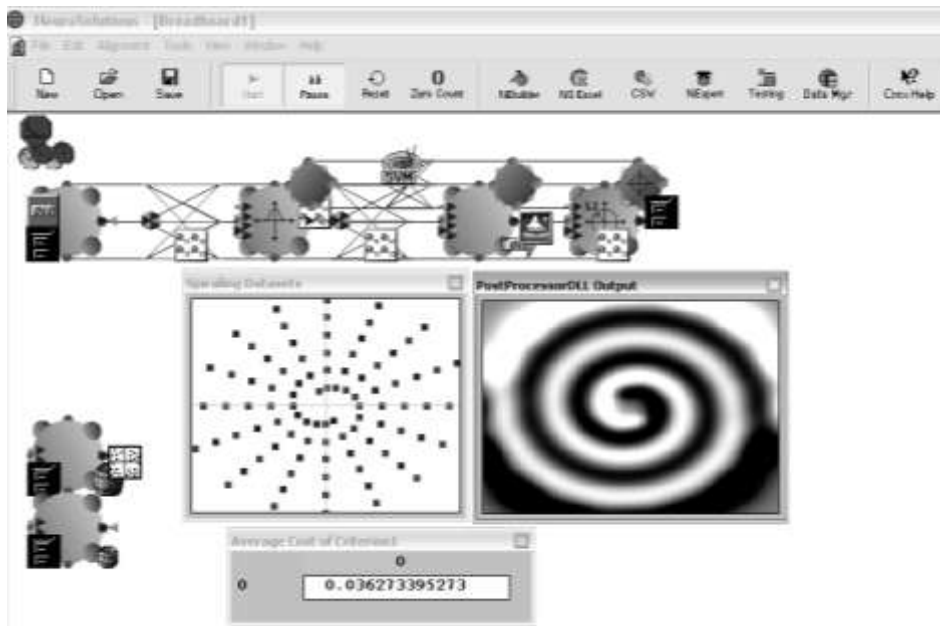
Jak wynika z dotychczasowych rozważań, proces uczenia sieci neuronowej polega na iteracyjnej zmianie wartości wag na podstawie informacji zaczerpniętej ze zbioru trenującego, pod nadzorem zastosowanej reguły uczenia. Cała wiedza nabyta w tym procesie przez sieć zakodowana jest w wartościach wag. Jest to reprezentacja liczbowa i rozproszona po całej macierzy wag. Jako taka nie nadaje się do interpretacji przez człowieka, co zdecydowanie odróżnia sieci neuronowe od innych systemów Inteligencji Obliczeniowej, bazujących na przetwarzaniu informacji symbolicznej. Sieć neuronowa nie potrafi uzasadnić wypracowanej decyzji. Jest to zasadnicza wada tej techniki. Jednak dzięki zrównoległonemu, rozproszonemu kodowaniu wiedzy, sieć wykazuje dużą odporność na zaburzenia (uszkodzenia) i dużą plastyczność, tj. zdolność do zmiany zakodowanej wiedzy małym kosztem przy zmieniających się warunkach zewnętrznych.

Cała nabyta przez sieć wiedza pochodzi z prezentowanego jej zbioru trenującego. Krytyczną wagę ma zatem odpowiednia jakość danych uczących. Przede wszystkim powinny one być reprezentatywne, tzn. powinny pokrywać całą dziedzinę uczonego problemu. Jeśli tak nie jest, to sieć neuronowa zastosuje interpolację, która nie musi zgadzać się z faktycznym charakterem modelowanego fenomenu. Po drugie, sieć potrzebuje odpowiednio dużej ilości danych trenujących – tym większej, im większa jest liczba atrybutów opisujących przetwarzane dane oraz im większa jest sama sieć (w sensie liczby neuronów). Większe sieci są bardziej elastyczne – potrafią wytwarzać bardzo złożone rozwiązania, dlatego potrzebują znacznej ilości danych trenujących. Jeżeli to możliwe, należy dążyć do jak najmniejszych sieci i jak najmniej wymiarowych danych. Zwłaszcza w tym drugim względzie jest wiele do zrobienia. Dane uczące powinny być poddawane wszelkim dostępnym technikom ekstrakcji istotnych atrybutów przed zaprezentowaniem ich sieci. Dla sygnałów czasowych może to być np. zastosowanie rozkładu spektralnego i wybranie ograniczonej liczby dominujących składowych spektrum. Dla obrazów wizyjnych może to być przeskalowanie, redukcja rozdzielczości kolorów, przygotowanie wszelkiego rodzaju deskryptorów, takich jak np. histogram. Bardzo pożądane jest stosowanie technik statystycznych takich jak np. PCA (ang. *Primary Component Analysis*) polegających na wskazaniu atrybutów odpowiadających maksymalnej energii (wariancji) przetwarzanych danych. Im mniej wymiarowe dane, tym większa szansa na sprawne wytrenowanie sieci.

Należy pamiętać, że prawdziwym celem procesu uczenia sieci jest uzyskanie systemu o dobrej generalizacji, tj. takiego, który wytrenowany na przygotowanym zbiorze danych będzie dobrze spisywał się w fazie eksploatacji, przetwarzając dane, które nigdy nie były mu prezentowane. Tak elastyczne systemy, jak sztuczne sieci neuronowe, są zdolne dopasować się do nawet najdrobniejszych fluktuacji w danych, a że dane są zawsze obciążone jakimiś zaburzeniami – szumem, błędami systematycznymi, to istnieje duże ryzyko, że sieć nauczy się losowych fluktuacji występujących niepowtarzalnie w zbiorze trenującym, ze szkodą dla dokładności przetwarzania w fazie eksploatacji. Istnieją różne techniki zapobiegania takiemu zjawisku (zwanemu przetrenowaniem sieci) – wczesne przerywanie uczenia, ograniczanie struktury sieci, wprowadzanie do funkcji błędu tzw. składników regularyzacyjnych. Zawsze niezbędne jest też operowanie w fazie uczenia dwoma zbiorami danych – zbiorem trenującym i, nie używanym do uczenia a tylko do kontrolowania jakości wypracowanego rozwiązania,

zbiorem testującym. Jako praktyczną regułę często podaje się, że wszystkie dostępne dane powinny być dzielone w proporcjach 60% na zbiór trenujący i 40% na zbiór testujący. Zabieg jest tak ważny, że nawet, gdy danych jest bardzo mało, i trzeba użyć je wszystkie do uczenia, to stosuje się metodę zwaną walidacją krzyżową, polegającą na wielokrotnym uczeniu sieci za pomocą zbioru zubożonego o pewien mały fragment testujący, za każdym razem inny.

Aby używać SSN nie trzeba pisać własnego oprogramowania. Istnieje szereg bardzo dobrych pakietów oprogramowania, tzw. symulatorów SSN. Określenie „symulator” pojawia się tu głównie ze względu na to, że ze swej istoty równoległa struktura obliczeniowa sieci neuronowej jest siłą rzeczy konwertowana na uszeregowany proces obliczeniowy wykonywany na pojedynczym procesorze komputera. Dobrym przykładem takiego symulatora jest system Neurosolutions firmy NeuroDimensions (rys. 7). Całe oprogramowanie ma organizację obiektową. Sieci neuronowe buduje się z komponentów rozmieszczonych na wielu paletach. Dostępne są różnego rodzaju warstwy neuronów, połączeń, reguły uczenia, harmonogramy zmiany parametrów, źródła danych, elementy wizualizacji. Istnieje możliwość rozszerzania funkcjonalnego systemu poprzez dołączanie własnych bibliotek DLL a nawet włączanie fragmentu kodu źródłowego w języku C. Zbudowana i wytrenowana sieć może też zostać wyeksportowana jako kod źródłowy w C, do włączenia we własnym projekcie programistycznym.



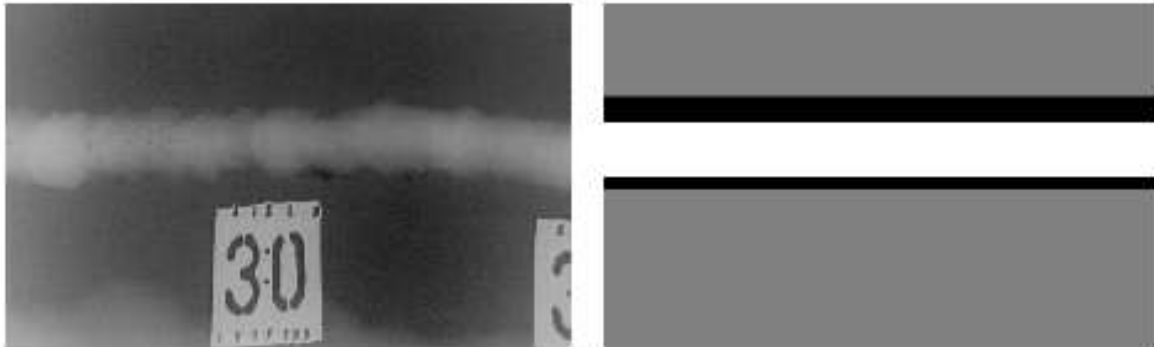
Rys. 7 Oprogramowanie NeuroSolutions

7. Przykład zastosowania – przetwarzanie obrazów

Czas na prezentację dwóch przykładów zastosowania SSN w Badaniach Nieniszczących. Opisywane w literaturze przypadki, to w większości zastosowanie warstwowych sieci neuronowych, uczonych przy użyciu algorytmu BP do problemu rozpoznawania wzorców.

Artykuł [5] opisuje zastosowanie SSN do automatycznej inspekcji obrazów radiograficznych spawów w rurach ze szwami. System zdolny do samodzielnego wykrywania wad w spawach w czasie rzeczywistym musi przetwarzać ogromną ilość informacji wizualnej. Pożądane jest ograniczenie strumienia informacji poprzez przeprowadzenie segmentacji, tj. oddzielenia obrazu spawu od tła. Dalszej analizie podlega jedynie użyteczna część zdjęcia (spaw). Segmentacja w zaproponowanej przez autorów metodzie opiera się na zastosowaniu trójwarstwowej sieci neuronowej, wypracowującej dla każdego piksela decyzję, czy należy on do obrazu spawu, czy do tła. Jest to więc realizacja dychotomicznej klasyfikacji. Tak

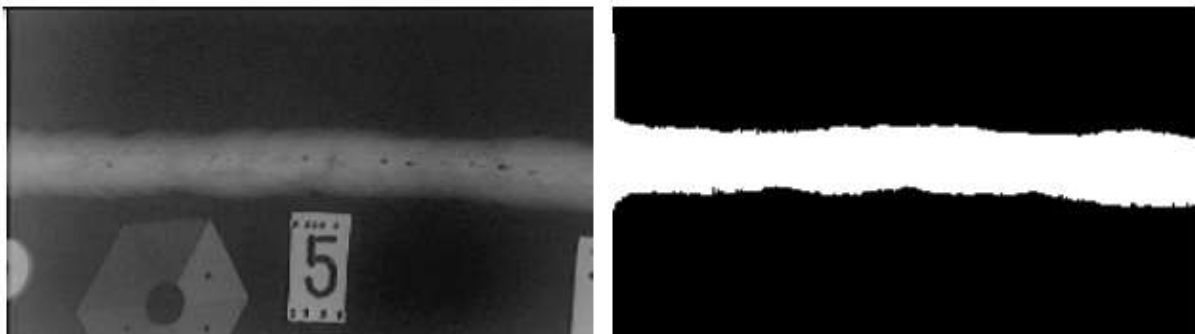
naprawde, zaproponowana siec ma wyjscie ciagle przeskalowane do zakresu od 0 do 255, które można interpretować, jako przeskalowaną miarę prawdopodobieństwa, że rozpatrywany piksel należy do spawu. Dychotomia jest uzyskiwana poprzez dyskryminację progową tego wyjścia. Sieć jest trenowana przy w trybie nadzorowanym. Zbiór trenujący pochodzi z kilku par zdjęć, zawierających obraz rzeczywisty (negatyw zdjęcia radiograficznego) oraz przygotowany przez doświadczonego operatora schematyczny szablon spawu. Rys. 8 przedstawia taką parę.



Rys. 8 Zdjęcie radiograficzne spawu i szablon

Obszar biały w szablonie znakuje spaw, szary – tło, czarny – strefę nieokreśloną. Każda para odpowiadających sobie pikseli z obu obrazów stanowi parę trenującą dla sieci neuronowej. Wartością oczekiwaną jest poziom szarości w szablonie. Obiektem przetwarzanym – deskryptor piksela w obrazie oryginalnym. Autorzy stosują 6 atrybutów na określenie tego piksela: jego współrzędne x,y , jego jasność, oraz średnią, medianę i maksymalną wartość jasności z 8-pikselowego lokalnego otoczenia. W artykule zaprezentowane są rezultaty działania sieci wytrenowanej za pomocą 40000 par trenujących pochodzących ze trzech różnych zdjęć radiograficznych. Na rys. 9 pokazany jest przykład segmentacji wypracowanej przez sieć neuronową. Oryginalne zdjęcie przed przetworzeniem przez sieć zostało odszumione za pomocą dolnoprzepustowej filtracji z okienkiem 5x5 pikseli. Następnie dla każdego piksela obrazu 6 opisanych powyżej atrybutów zostało przetworzonych przez wytrenowaną sieć. Wyjście sieci poddane dyskryminacji z progiem 128 określa kolejne piksele obrazu wynikowego. Dobrze o skuteczności metody świadczy fakt zaklasyfikowania znaczników radiograficznych jako tła obrazu.

Artykuł nie poprzestaje na tym, ale opisuje zastosowanie podobnej metody do segmentacji użytecznego fragmentu zdjęcia na obszary normalnego spawu i obszary defektów. Także i w tym przypadku zastosowane jest uczenie nadzorowane na podstawie par trenujących wygenerowanych z zestawów obejmujących zdjęcie oryginalne i szablon określający położenie defektów.



Rys. 9. Segmentacja przeprowadzona przez SSN

Tym razem rolę „eksperta” pełni kosztowna obliczeniowo, ale automatyczna procedura detekcji defektów, zaczerpnięta z praktyki przetwarzania obrazów w medycynie. Użyty algorytm Kehou określa piksel jako należący lub nie do obszaru defektu na podstawie wskaźników wariancyjnych i gradientowych wyznaczonych z otoczenia piksela o rozmiarze 15 na 15 punktów. W eksperymencie przytoczonym w artykule, jako źródło danych trenujących użyte zostało 5 zdjęć 128x128, zawierających przykłady różnych defektów wraz z szablonami uzyskanymi poprzez zastosowanie algorytmu Kehou. Tym razem atrybutami charakteryzującymi każdy piksel obrazu było 81 wartości jasności wszystkich pikseli należących do otoczenia 9x9. Sieć neuronowa o 81 wejściach, dwóch warstwach ukrytych po 10 neuronów każda i jednym wyjściu została poddana treningowi z użyciem 50000 par trenujących wygenerowanych z wyżej wspomnianych 5 zestawów.

Rys.10 pokazuje rezultat detekcji defektów przeprowadzonej za pomocą wytrenowanej sieci neuronowej. Autorzy nadmienią, że mimo mniejszego okienka (9x9) i przy szybszym działaniu, detekcja zrealizowana przez sieć wydaje się lepsza od tej przeprowadzonej przez referencyjny algorytm Kehou.



Rys. 10. Przykłady detekcji defektów przez SSN. Po lewej stronie – zdjęcia radiograficzne

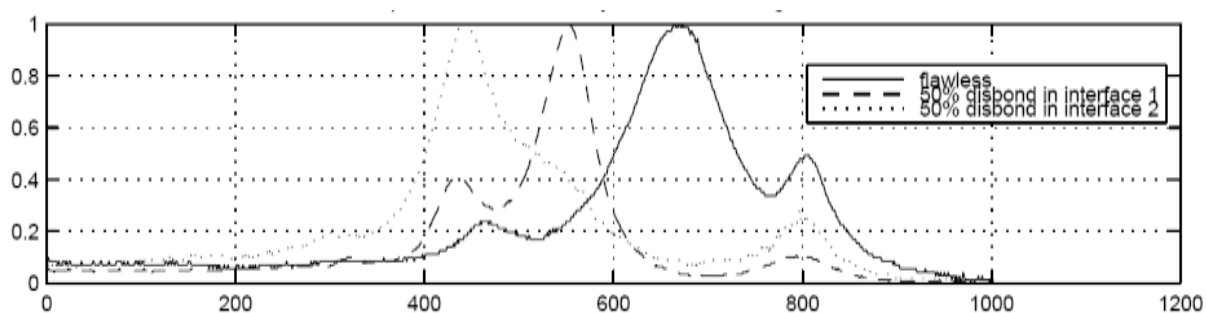
8. Przykład zastosowania – rozpoznawanie wzorców w przebiegach

W artykule [6] opisane jest zastosowanie SSN do rozpoznawania defektów na podstawie spektrum wyznaczonego przez rezonansową spektroskopię ultradźwiękową. W metodzie tej badany obiekt jest wprowadzany w rezonans mechaniczny przez przetwornik ultradźwiękowy o płynnie zmieniającej się częstotliwości emitowanej fali. Przetwornik odbiorczy rejestruje amplitudę i względną fazę odebranego sygnału, dając w wyniku spektrum częstotliwościowe. Kształt tego spektrum świadczy o wewnętrznej strukturze badanego obiektu, m.in. o występujących defektach. Sygnał spektrum może podlegać ocenie (klasyfikacji) przy użyciu SSN. Artykuł wskazuje trzy możliwe, potwierdzone eksperymentami, aplikacje metody. Pierwszy przypadek to rozpoznawanie defektów w kompozytowych panelach używanych w przemyśle lotniczym. Do klasyfikacji spektrum została użyta warstwowa sieć neuronowa, uczona za pomocą algorytmu BP. Jako zbioru trenującego użyto zestawu 23 spektrum z dobrych paneli oraz 29 spektrum pochodzących z badania uszkodzonych paneli (nacięcia w warstwie włókien węglowych z delaminacją lub bez, uszkodzenia cieplne, uszkodzenia udarowe). Każde spektrum obejmowało 1000 próbek z zakresu częstotliwości od 50kHz do 3MHz. Po wytrenowaniu sieci była ona w stanie poprawnie rozpoznać defekty na podstawie spektrum w 82% danych ze zbioru testowego. Brak jest danych o strukturze sieci, ale w innym, podobnym eksperymencie używana jest mała sieć neuronowa (3 neurony w warstwie ukrytej) z krótkim czasem uczenia (55 epok).

Innym przykładem jest rozpoznawanie defektów w łożyskach kulkowych. Tym razem sieć była trenowana za pomocą 130 spektrum pochodzących z 20 sprawnych łożysk oraz 208 z wadliwych. Spektra w tym wypadku to 100 punktów z zakresu częstotliwości od 1kHz do

10MHz. Autorzy podają, że po wytrenowaniu, 96.5% łożysk była poprawnie klasyfikowana przez sieć neuronową.

Wreszcie trzecią przytoczoną aplikacją jest wykrywanie rozwarstwień w strukturach warstwowych. Tym razem przedmiotem eksperymentu było nie tylko rozróżnienie między dobrymi i wadliwymi obiektami, ale także identyfikacja granicy, na której nastąpiło rozwarstwienie. Zastosowana sieć miała 3 do 5 wyjść, w zależności od liczby warstw. Wartość na wyjściu sieci była interpretowana jako prawdopodobieństwo wystąpienia rozwarstwienia w warstwie skojarzonej z tym wyjściem. Źródłem danych był obiekt z szeregiem sztucznych wad, badany handlowym spektroskopem. W artykule brak danych o rozmiarze zbioru trenującego, wzmiankowane jest jedynie, że 50% wszystkich danych zostało użytych do testowania wytrenowanej sieci. Przykłady spektrum podlegających klasyfikacji pokazane są na rys. 11. Wytrenowana sieć w 100% poprawnie zaklasyfikowała wszystkie dane ze zbioru testowego w dychotomicznym rozróżnieniu obiektów wadliwych i dobrych oraz 97.5% poprawność wskazania numeru granicy z rozwarstwieniem. Cechą szczególną było to, że wykrywane były nawet rozwarstwienia o powierzchni mniejszej od powierzchni przetwornika piezoelektrycznego.



Rys. 11. Spektra uzyskane ze spektrometru ultradźwiękowego

9. Uwagi końcowe

W artykule jedynie zasygnalizowane zostało całe bogactwo dziedziny badań nad SSN. W szczególności całkowicie pominięty został obszar uczenia nienadzorowanego oraz sieci rekurencyjnych, które odgrywają bardzo wielką rolę w zagadnieniach ekstrakcji użytecznych atrybutów ze zbioru danych oraz rozpoznawania wzorców. Pominięte zostały specyficzne techniki dotyczące przetwarzania przebiegów czasowych, czy też bardzo ostatnio popularne, osadzone w statystyce techniki takie jak SVM czy RBF. I wiele innych zagadnień. Sztuczne Sieci Neuronowe są po prostu niezwykle bogatą dziedziną z ogromnym potencjałem zastosowań i trudno je przeoczyć w dzisiejszej praktyce inżynierskiej i badawczej.

10. Literatura

- [1] Freeman J., Skapura D., Neural Networks, Algorithms, Applications and Programming Techniques, Addison-Wesley Publishing Company, 1991
- [2]. Haykin S., Neural Networks. A Comprehensive Foundation, Prentice Hall International, 1999
- [3] Rosenblatt F., Audio Signal Pattern Perception Device, U.S. Patent nr 3287649, 1996
- [4] Hassoun M., Fundamentals of Artificial Neural Networks, MIT Press 1995
- [5] Shaun W., Lawson and Graham A. Parker, Automatic detection of defects in industrial ultrasound images using a neural network, Proceedings of SPIE, vol. 2786, 1996, pp. 37-47
- [6] Stepinski T., Ericsson L., Vagnhammar B., Gustafsson M., Neural Network Based Classifier for Ultrasonic Resonance Spectra, Proceedings of 7th European Conference on Non-Destructive Testing, 26-29 May 1998.