



Sztuczne sieci neuronowe

Sieci nieliniowe wielowarstwowe BP



Zagadnienia:

1. Sieci liniowe- budowa neuronu, uczenie, klasy rozwiązywanych zadań.
2. Sieci nieliniowe – budowa (struktura), metoda uczenia, klasy rozwiązywanych zadań.
3. Oprogramowanie sieci nieliniowych w pakiecie Matlab, przykłady.

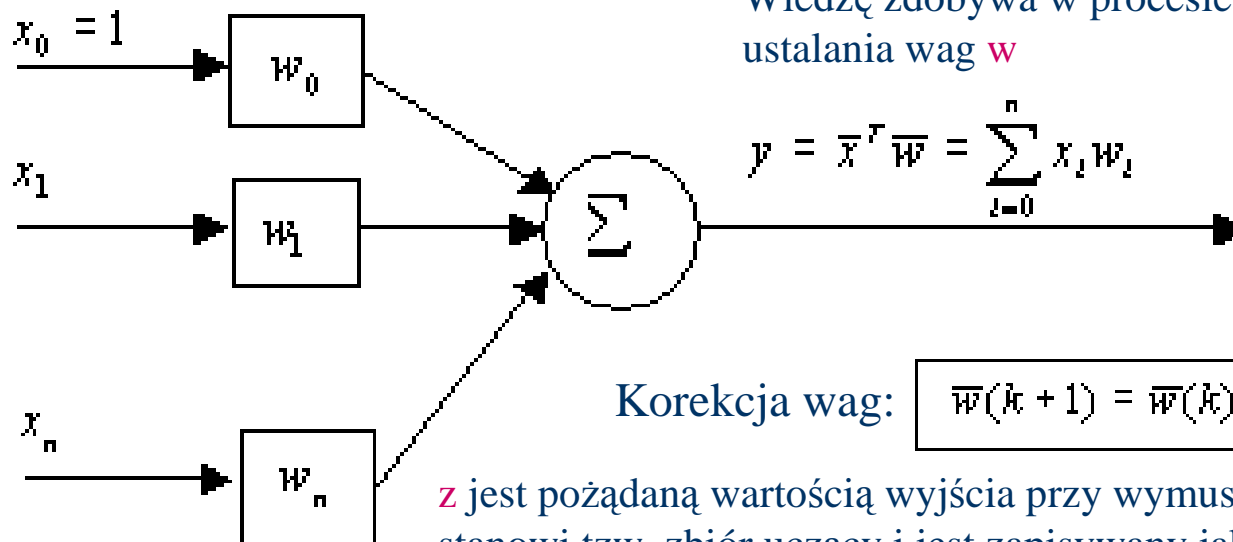
Sieci liniowe

Model podstawowej komórki sieci liniowych -

- neuronu liniowego

Neuron liniowy jest zdolny podzielić przestrzeń cech na dwie klasy (separowalne liniowo)

Wiedzę zdobywa w procesie uczenia czyli ustalania wag \mathbf{w}



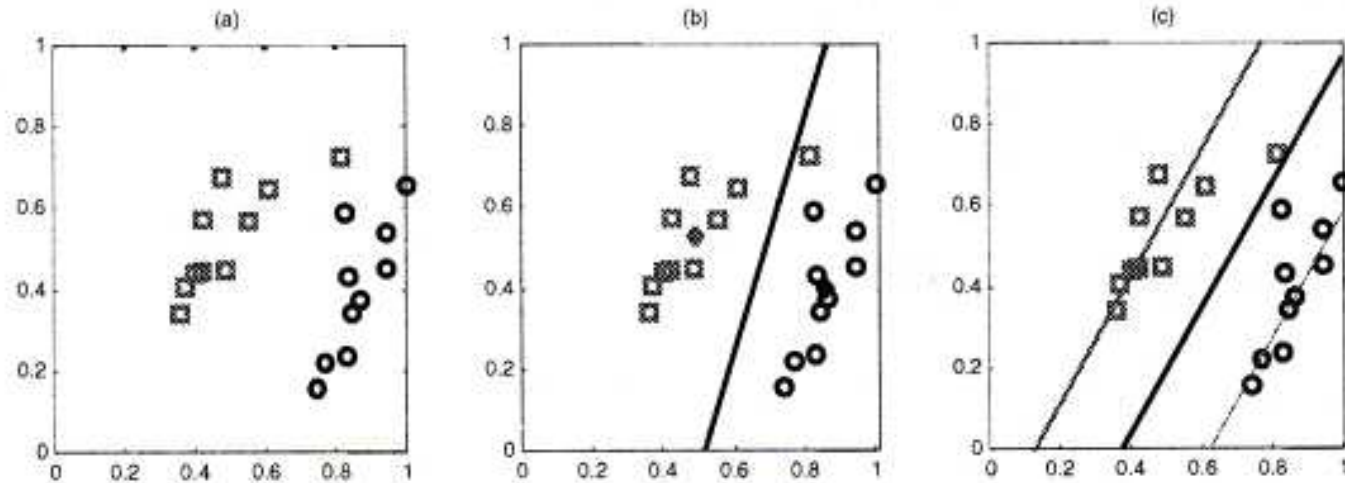
Korekcja wag:

$$\bar{\mathbf{w}}(k+1) = \bar{\mathbf{w}}(k) - \eta(z - y)\bar{\mathbf{x}}$$

z jest pożądaną wartością wyjścia przy wymuszeniu \mathbf{x} , zbiór par $\langle \mathbf{x}, z \rangle$ stanowi tzw. zbiór uczący i jest zapisywany jako $\langle \mathbf{X}, \mathbf{z} \rangle$

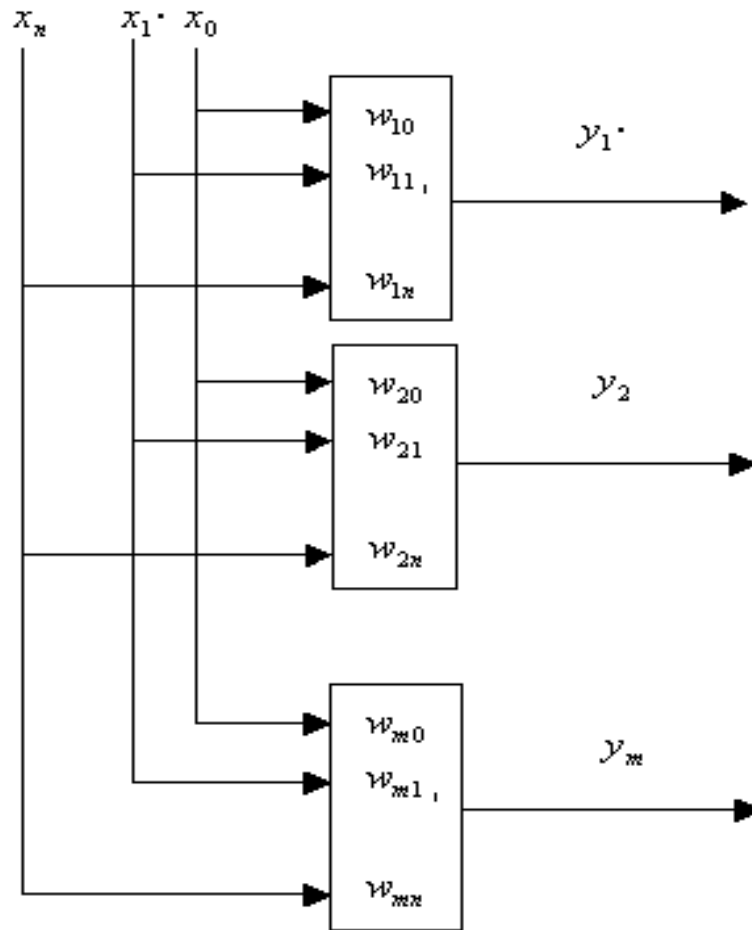
(\mathbf{X} - kolumnowa macierz wzorców, \mathbf{z} - wektor pożądanых wyjść)

Zasada klasyfikacji w neuronie liniowym



Porównanie wyników klasyfikacji metodą: b) reprezentantów klas, c) neuronem liniowym (inaczej aproksymacji klas)

Sieć liniowa – warstwa neuronów liniowych



Warstwa neuronów liniowych
pozwala na rozwiązanie zadań:

klasyfikacji – jeden neuron dokonuje
detekcji 1 klasy ,

predykcji m chwil wprzód.

Uczenie jest naturalnym
rozszerzeniem uczenia pojedynczego
neuronu, jedynie mamy tu do
czynienia z macierzą żądanych wyjść.
Tak więc zbiór uczący dany jest jako:
 $\langle X, Z \rangle$



Sieć liniowa – zastosowania

1. Proste zadania klasyfikacji, gdzie klasy są separowalne liniowo.
2. Zagadnienia predykcji (przewidywania) .



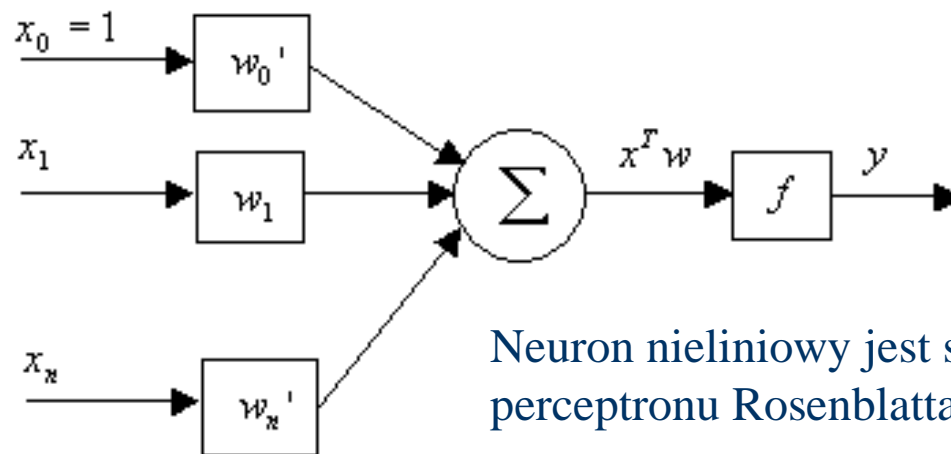
Sieć liniowa – wady



1. Rozwiązuje jedynie proste zadania klasyfikacji (separowalność liniowa jest znacznym ograniczeniem zastosowalności).
2. Nie można zbudować sieci liniowej wielowarstwowej – mającej znacznie większe możliwości (sieć liniowa wielowarstwowa zawsze daje się zastąpić siecią jednowarstwową).

Sieć neuronowa nieliniowa wielowarstwowa

Neuron nieliniowy



Neuron nieliniowy jest swoją budową zbliżony do perceptronu Rosenblatta.

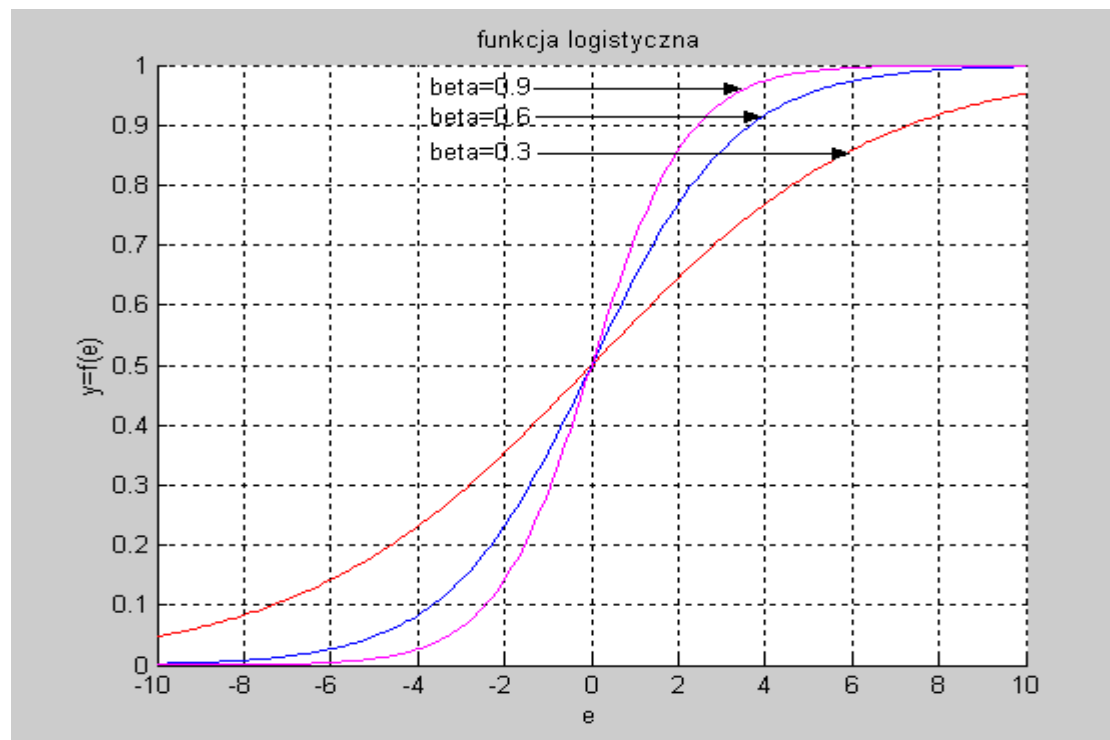
Bardzo istotna różnica polega na tym, że w tym przypadku funkcja f (zwana funkcją aktywacji lub propagacji) jest różniczkowalna (posiada pochodną).

Uczenie neuronu nieliniowego jest oparte najczęściej na metodzie gradientowej (obierania kierunku najszybszego spadku błędu sumarycznego)

Sieć neuronowa nieliniowa wielowarstwowa

Funkcje aktywacji (propagacji) f

1. Funkcja logistyczna

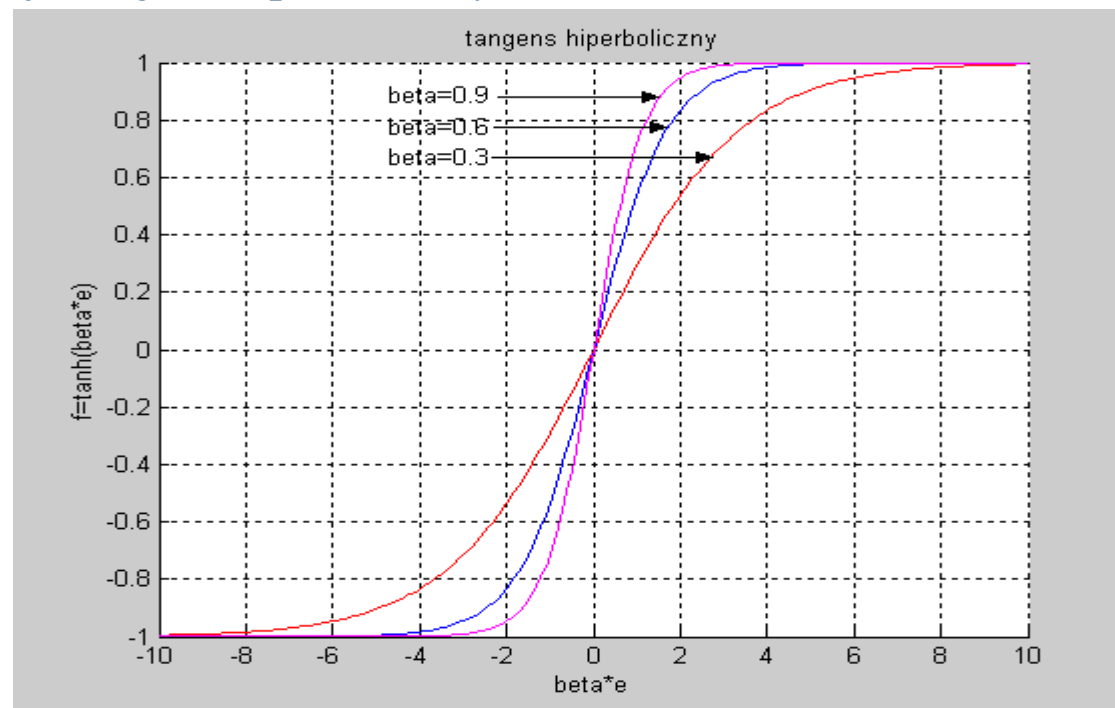


$$f(e) = \frac{1}{1 + \exp(-\beta e)} \quad e = w^T x \quad e - \text{łączne pobudzenie neuronu}$$

Sieć neuronowa nieliniowa wielowarstwowa

Funkcje aktywacji (propagacji) f

1. Funkcja tangens hiperboliczny



$$f(e) = \tanh(\beta \cdot e) \quad e = w^T x \quad e - \text{łączne pobudzenie neuronu}$$



Sieć neuronowa nieliniowa wielowarstwowa

Nowe możliwości



Z neuronów nieliniowych można budować sieci wielowarstwowe pozwalające na realizację bardzo skomplikowanych odwzorowań.

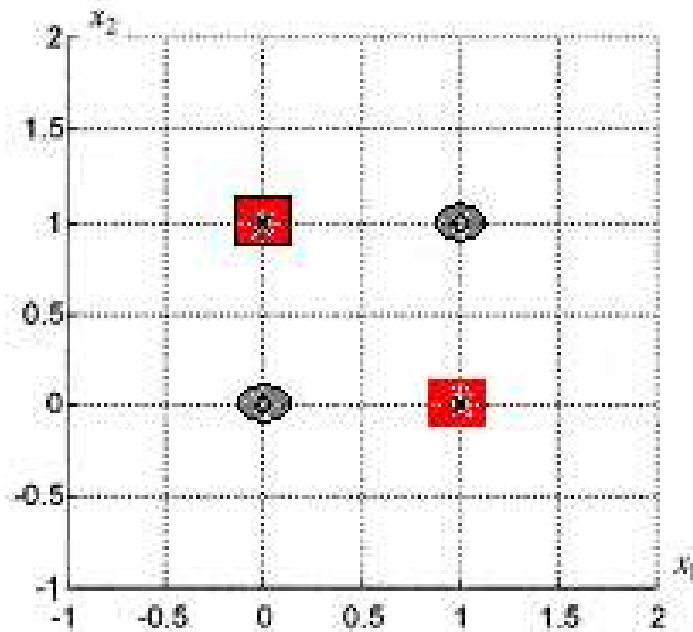
Dla sieci nieliniowych wielowarstwowych istnieje efektywna metoda uczenia (w przeciwieństwie do sieci perceptronowej) określana jako:

metoda wstecznej propagacji błędów

(ang. backpropagation - BP)

Sieć neuronowa nieliniowa wielowarstwowa

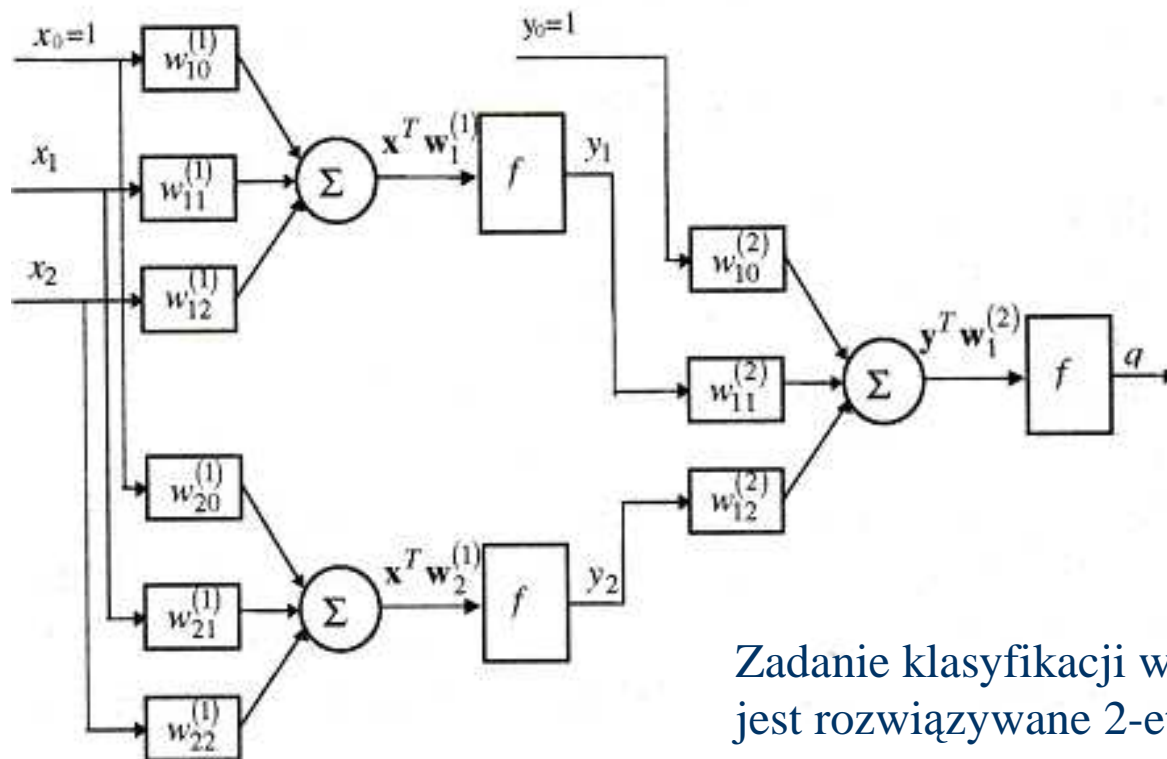
Zadanie klasyfikacji – klasy nieseparowalne liniowo



xor

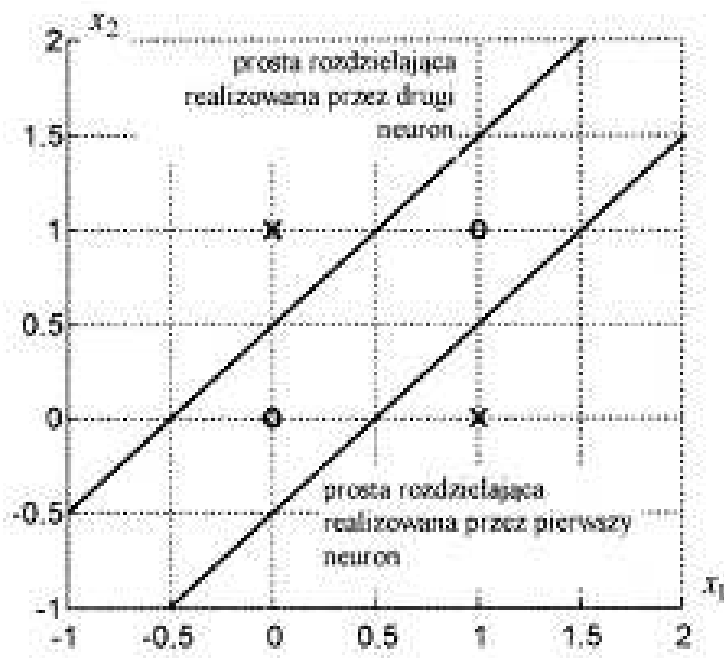
Rozwiązanie ?

Sieć neuronowa nieliniowa. Zadanie klasyfikacji – klasy nieseparowalne liniowo. Metoda – sieć wielowarstwowa

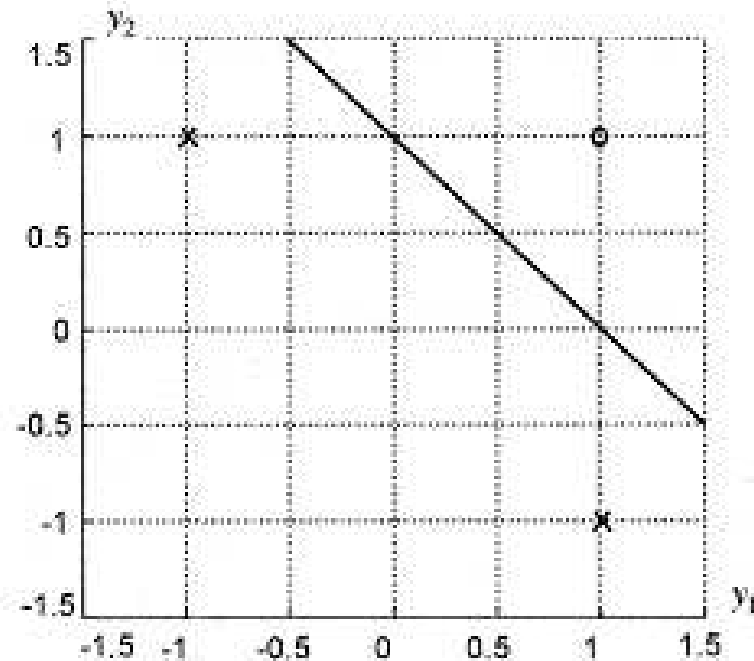


Zadanie klasyfikacji w tym przypadku jest rozwiązywane 2-etapowo

Sieć neuronowa nieliniowa. Zadanie klasyfikacji – klasy nieseparowalne liniowo. Metoda – sieć wielowarstwowa



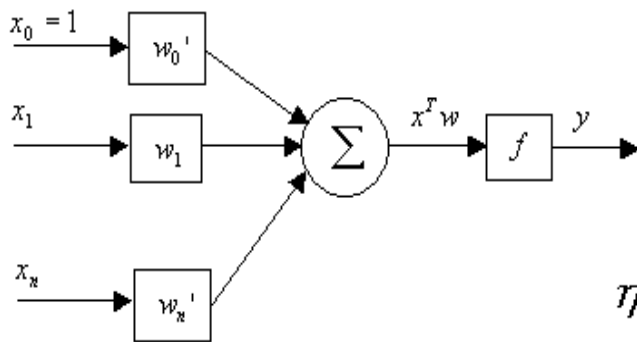
Sposób rozdziatu sygnałów pierwszego etapu klasyfikacji



Sposób rozdziatu sygnałów drugiego etapu klasyfikacji

Sieć neuronowa nieliniowa wielowarstwowa

korekcja wag pojedynczego neuronu



Korekta wag:

$$w_i(k+1) = w_i(k) - \eta(y - z) f' x_i$$

η - współczynnik szybkości uczenia

$y - z$ - odchylenie wartości wyjścia neuronu od wartości pożądanej - błąd



Do wyznaczenia korekty wag potrzebna
jest pochodna funkcji aktywacji f

Pochodna funkcji logistycznej

$$f' = y' = y(1 - y)$$


Pochodna funkcji tanh

$$f' = y' = 1 - y^2$$



Sieć neuronowa nieliniowa wielowarstwowa

algorytm wstecznej propagacji błędów



Podstawowy problem :



Jak wyznaczyć wartości błędów dla neuronów położonych w **warstwach wewnętrznych (ukrytych)** ?

Rozwiązanie – rok 1982 – algorytm wstecznej propagacji błędów.

Podstawowa zasada algorytmu:



Wartość błędu neuronu warstwy ukrytej jest wyznaczana na podstawie błędów neuronów do których dany neuron wysyła sygnały (wielkość przejmowanej części błędu zależy od tego z jaką wagą sygnał jest przyjmowany).

Wniosek: propagacja błędów (tym samym korekcja wag) przebiega od **warstwy ostatniej (wyjściowej)** poprzez wszystkie warstwy ukryte do **warstwy wejściowej**.



Sieć neuronowa nieliniowa wielowarstwowa

Architektura sieci neuronowej, parametry treningu sieci



Na architekturę sieci składają się:

- liczba warstw oraz liczby neuronów w warstwach
- typ (typy) funkcji propagacji w sieci

Parametry treningu sieci:

- współczynnik szybkości uczenia
- graniczny sumaryczny błąd na wyjściu sieci
- liczba epok (epoka – jednokrotne podanie wszystkich wzorców na sieć w procesie uczenia)




Sieć neuronowa nieliniowa wielowarstwowa


Klasy rozwiązywalnych zadań



1. Zadania klasyfikacji (rozpoznawania), klasy nieseparowalne liniowo.
2. Skomplikowane zadania predykcji przebiegów zarówno wolno-
jak i szybkozmiennych.
3. Zadania pochodne.



Matlab – oprogramowanie do symulacji działania sieci liniowych i nieliniowych wielowarstwowych



Obejmuje :

1. Inicjalizację sieci – struktury (liczby warstw, liczb neuronów w poszczególnych warstwach) oraz wag początkowych
2. Uczenie (trenowanie) sieci na podstawie skonstruowanego uprzednio zbioru uczącego (konstrukcja zbioru uczącego wynika z typu rozwiązywanego zadania)
3. Symulację działania sieci (dla danych nietreningowych)

Inicjalizacja sieci nieliniowej

Wywołanie: `net = newff(PR,[S1 S2...SNI],{TF1 TF2...TFNI},BTF,BLF,PF)`

PR - Rx2 matrix of min and max values for R input elements.

Si - Size of ith layer, for NI layers.

TFi - Transfer function of ith layer, default = 'tansig'.

BTF - Backprop network training function, default = 'trainlm'.

BLF - Backprop weight/bias learning function, default = 'learngdm'.

PF - Performance function, default = 'mse'.

inne funkcje propagacji: TANSIG, LOGSIG, PURELIN.

inne metody trenowania: TRAINLM, TRAINBFG, TRAINRP, TRAINGD.

Uczenie (trenowanie) sieci

Wywołanie: [Pf,Tr,Y,E,Pf,Af] = train(NET,P,T,Pi,Ai,VV,TV)

Parametry wejściowe:

NET - Network.

P - Network inputs.

T - Network targets, default = zeros.

Pi - Initial input delay conditions, default = zeros.

Ai - Initial layer delay conditions, default = zeros.

VV - Structure of validation vectors, default = [].

TV - Structure of test vectors, default = [].

Uczenie (trenowanie) sieci cd.

Wyniki działania (parametry wyjściowe)

NET - New network.

Tr - Training record (epoch and perf).

Y - Network outputs.

E - Network errors.

Pf - Final input delay conditions. Af - Final layer delay conditions.

Parametry treningu:

net.trainparam.goal – błąd graniczny.

net.trainparam.epoch – liczba epok uczenia



Symulacja działania sieci

funkcja sim



Wywołanie: $Y = \text{sim}(\text{net}, P)$

net - sieć

P – dane wejściowe

Inicjalizacja sieci liniowej

Wywołanie: `net = newlin(PR,S,ID,LR)`

PR - $R \times 2$ matrix of min and max values for R input elements.

S - Number of elements in the output vector.

ID - Input delay vector, default = [0].

LR - Learning rate, default = 0.01;



Inicjalizacja sieci perceptronowej



Wywołanie: `net = newp(PR,S)`

PR - **Rx2** matrix of min and max values for **R** input elements.

S - Number of elements in the output vector.