Metody detekcji uszkodzeń oparte na wiedzy — Sieci neuronowe

Dariusz Uciński

Instytut Sterowania i Systemów Informatycznych
Universytet Zielonogórski

Wykład 7

Wprowadzenie

- Okres kształtowania się teorii sztucznych sieci neuronowych: 1943–195
- Prace pionierskie
  - McCulloch i Pitts (1943): idea sieci neuronowych jako maszyn przetwarzających; matematyczny model komórki nerwowej
  - Hebb (1949): reguła uczenia samoorganizującego
  - Rosenblatt (1958): propozycja sieci neuronowej (perceptron) uczonej metodą nadzorowaną (z nauczycielem)

Założenia projektowe

1. Podstawowym elementem składowym sieci jest model neuronu McCullocha-Pittsa:

$$\varphi = \sum_{p=1}^{P} w_p u_p - \Theta$$

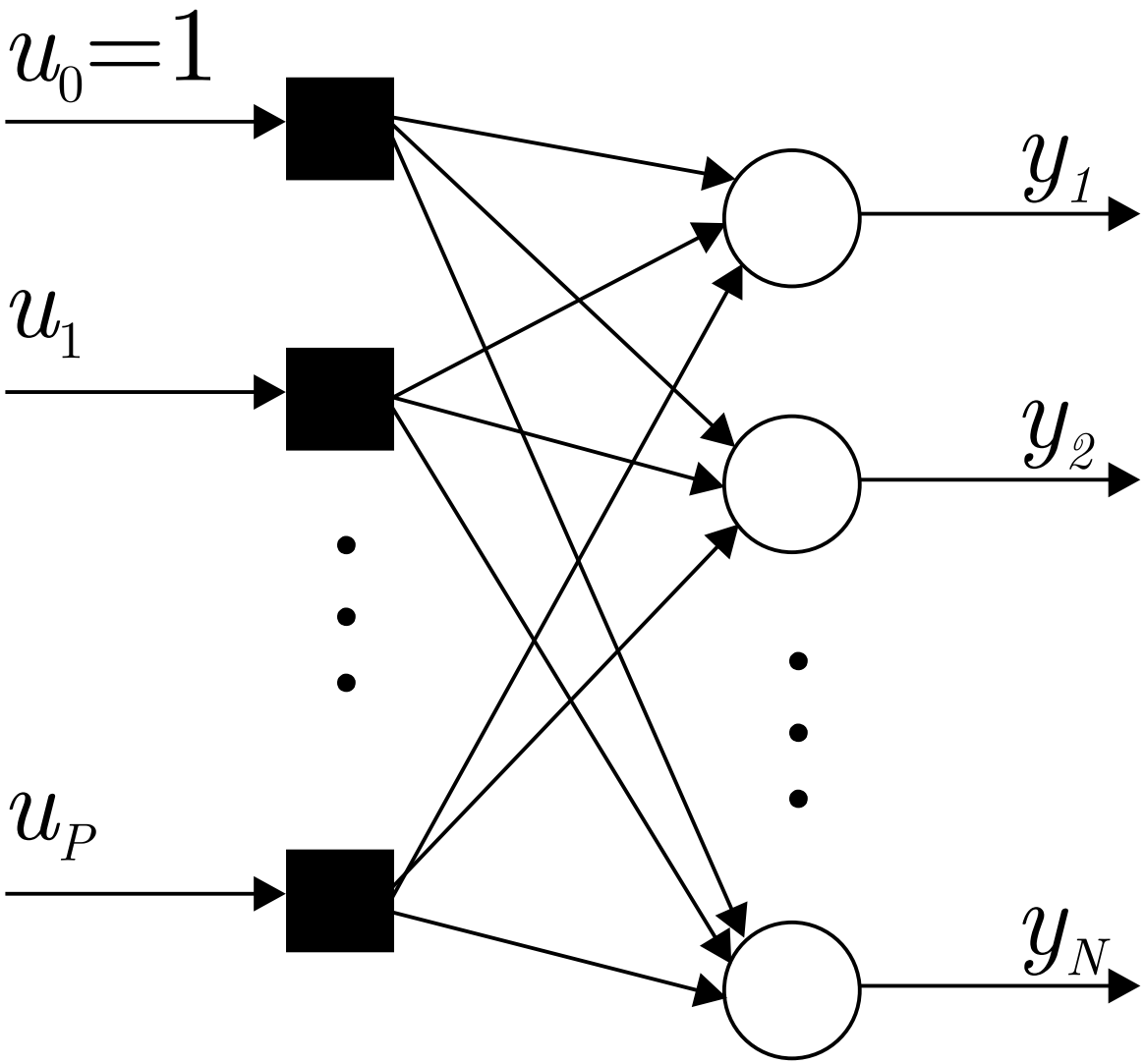
$$y = 1(\varphi) = \begin{cases} 1 & \text{dla } \varphi > 0 \\ 0 & \text{dla } \varphi \leqslant 0 \end{cases}$$

2. sieć można podzielić na ściśle uporządkowane i rozłączne klasy elementów zwanych warstwami (wejściowa, wyjściowa i ukryte)
3. perceptron nie zawiera połączeń między elementami należącymi do tej samej warstwy
4. połączenia pomiędzy warstwami są asymetryczne i skierowane zgodnie z ich uporządkowaniem, od warstwy wejściowej, poprzez ukryte do wyjściowej

Struktura sieci

1. Warstwa wejściowa
   - elementy o uproszczonej funkcji przejścia i jednym wejściu
   - układ receptorów odbierających sygnały wejściowe
   - po wstępnym przetworzeniu (skalowanie, normalizacja, filtracja) przekazuje dane do warstwy następnej
2. Perceptron złożony tylko z warstw wejściowej i wyjściowej — perceptron prosty (perceptron jednowarstwowy)
3. W praktyce ustala się dla każdego elementu przetwarzającego wartość progu $\Theta = 0$. Jego rolę przejmuje waga $w_0 = -\Theta$ dodatkowego wejścia o wartości $u_0 = 1$
4. Każdy element może być traktowany jako niezależna sieć o $P + 1$ wejściach i jednym wyjściu

Struktura sieci



Proces uczenia perceptronu

- Uczenie sieci kontynuowane jest tak długo, aż w kolejnych iteracjach nie nastąpi zmiana wektora wag połączeń wstępujących do elementu przetwarzającego
- Ustala się również zadaną z góry liczbę iteracji na wypadek rozbieżności algorytmu
- Ważnym zagadnieniem jest kolejność prezentacji wzorców uczących
  1. prezentacja danego wzorca dopóki nie uzyska się stabilnego wektora wag i następnie przejście do kolejnego
  2. prezentacja wzorców cyklicznie zgodnie z ustaloną kolejnością
  3. do każdej prezentacji losowanie wzorca ze zbioru $\Gamma = \{(u^\mu, y_z^\mu)\}_{\mu=1}^{L}$ z prawdopodobieństwem $\frac{1}{L}$

Reguła uczenia perceptronu

- Procedura podaje na wejście sieci wzorce wejściowe u_i^μ ($\mu = 1, \ldots, L$) i sprawdza jednostki wyjściowe czy ich odpowiedzi y_s^μ są poprawne
- Jeżeli odpowiedzi sieci są poprawne $y_s^\mu = y_z^\mu$ to pozostawia się bez zmian tylko te połączenia, które prowadzą do tej jednostki
- Jeżeli odpowiedzi sieci są niepoprawne $y_s^\mu \neq y_z^\mu$ to do każdego połączenia prowadzącego do tej jednostki dodaje się pewną liczbę proporcjonalną do iloczynu sygnału wejściowego i pożądanego sygnału wyjściowego
- Matematyczny zapis reguły perceptronu

$$w_{ik}^{nowe} = w_{ik}^{stare} + \Delta w_{ik}$$

$$\Delta w_{ik} = \begin{cases} \eta y_{zi}^\mu u_k^\mu & \text{jeżeli } y_{zi}^\mu \neq y_{si}^\mu \\ 0 & \text{w pozostałych przypadkach} \end{cases}$$

Reguła uczenia perceptronu

- Alternatywny zapis

$$\Delta w_{ik}^{\mu} = \eta(y_{zi}^{\mu} - y_{si}^{\mu})u_k^{\mu}$$

gdzie: η — współczynnik uczenia

$$y_{zi}^{\mu} - y_{si}^{\mu} = \begin{cases} -1 & \text{dla } y_{zi}^{\mu} = 0 \text{ i } y_{si}^{\mu} = 1 \\ 0 & \text{dla } y_{zi}^{\mu} = y_{si}^{\mu} \\ 1 & \text{dla } y_{zi}^{\mu} = 1 \text{ i } y_{si}^{\mu} = 0 \end{cases}$$

Przykład 1

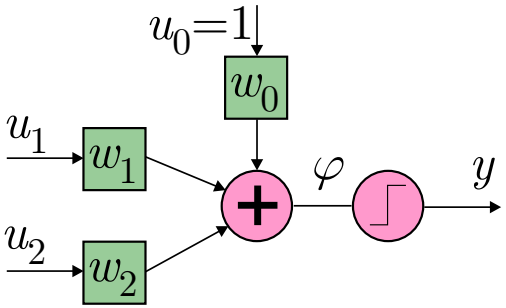
Korzystając z reguły uczenia perceptronu wyznaczyć jego wagi tak, aby rozpoznawał elementy dwóch zbiorów

u_0	u_1	u_2	y_z
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

Przyjąć początkowe wartości wag jako wektor $\mathbf{w}_0 = [1 \quad -1 \quad 1]$
Krok uczenia przyjąć $\eta = 0.5$

Przykład 1

- Struktura sieci



- Początkowa granica decyzyjna

Granica decyzyjna to hiperpłaszczyzna rozdzielająca elementy należące do dwóch różnych klas

granicę decyzyjną wyznacza się dla $\varphi = 0$

$$\varphi = w_0 + w_1 u_1 + w_2 u_2 \Rightarrow 0 = w_0 + w_1 u_1 + w_2 u_2$$

$$u_2 = -u_1 \frac{w_1}{w_2} - \frac{w_0}{w_2}$$

początkowa granica decyzyjna

$$d_0 : \quad u_2 = u_1 - 1$$

Przykład 1

- Odpowiedź sieci na poszczególne wzorce

$$
\begin{array}{lll}
y_s^1 = 1, \times & y_z^1 - y_s^1 = -1, & \Delta\mathbf{w}^1 = [-1 \;\; 0 \;\; 0] \\
y_s^2 = 1, \checkmark & y_z^2 - y_s^2 = 0, & \Delta\mathbf{w}^2 = [0 \;\; 0 \;\; 0] \\
y_s^3 = 0, \times & y_z^3 - y_s^3 = 1, & \Delta\mathbf{w}^3 = [1 \;\; 1 \;\; 0] \\
y_s^4 = 1, \checkmark & y_z^4 - y_s^4 = 0, & \Delta\mathbf{w}^4 = [0 \;\; 0 \;\; 0]
\end{array}
$$

- Wynikowy wektor korekcyjny

$$\Delta\mathbf{w} = \Delta\mathbf{w}^1 + \Delta\mathbf{w}^2 + \Delta\mathbf{w}^3 + \Delta\mathbf{w}^4 = [0 \;\; 1 \;\; 0]$$

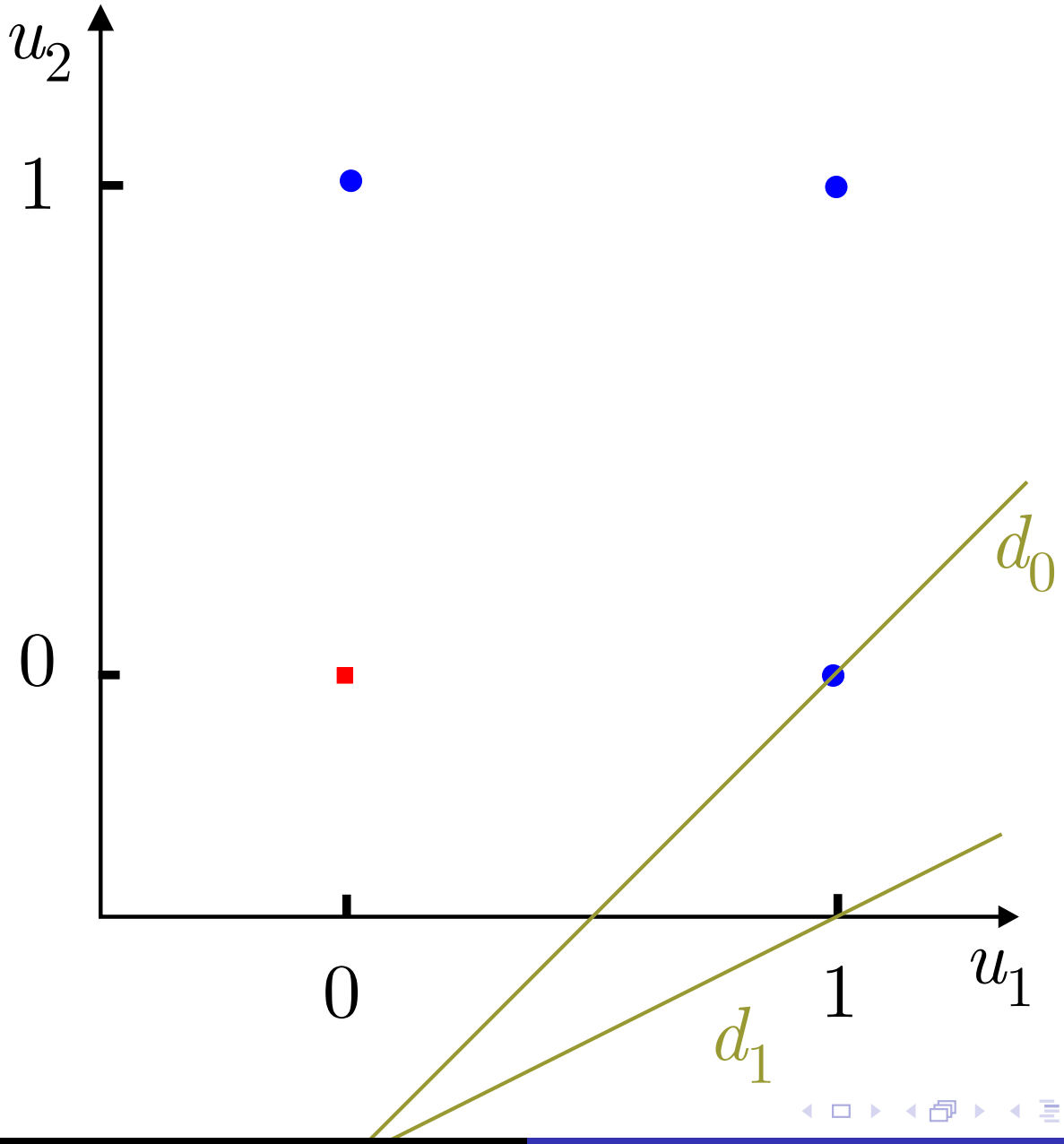
nowy wektor wag

$$\mathbf{w}^{nowe} = \mathbf{w}^{stare} + \eta\Delta\mathbf{w} = [1 \;\; -1 \;\; 1] + 0.5[0 \;\; 1 \;\; 0] = [1 \;\; -0.5 \;\; 1]$$

- Nowa granica decyzyjna

$$d_1 : \quad u_2 = u_1 \frac{0.5}{1} - \frac{1}{1} = 0.5u_1 - 1$$

Przykład 1



Separowalność liniowa

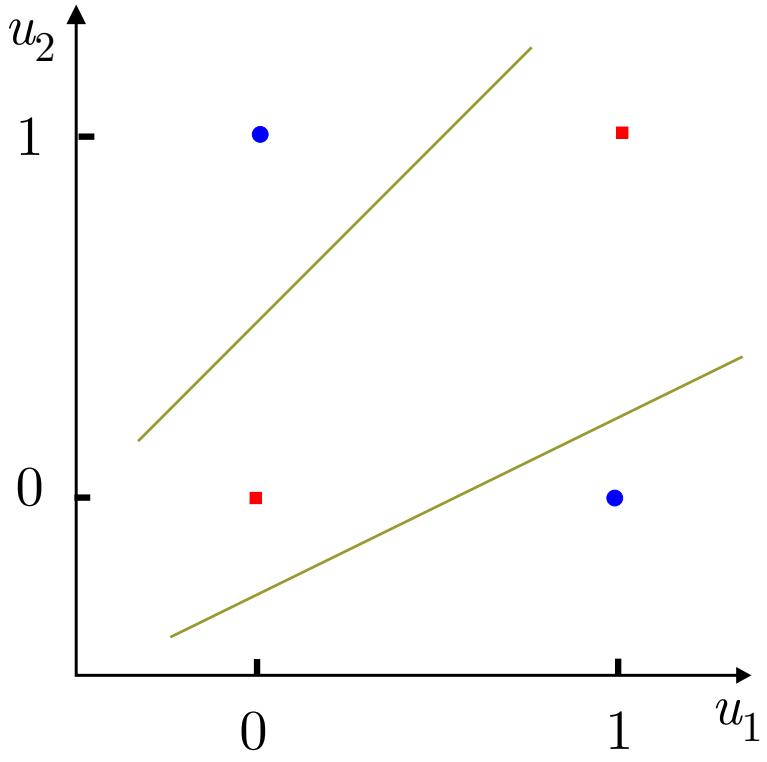
1. Zbiory wektorów wejściowych dla których funkcja aktywacji przyjmuje wartości 0 i 1 muszą być liniowo rozdzielne, tzn. muszą leżeć w dwóch różnych półpłaszczyznach rozdzielonych hiperpłaszczyzną
2. Warunek separowalności liniowej nie zawsze jest spełniony

Rozważmy np. problem XOR:

u_1	u_2	y_z
0	0	0
0	1	1
1	0	1
1	1	0

Separowalność liniowa

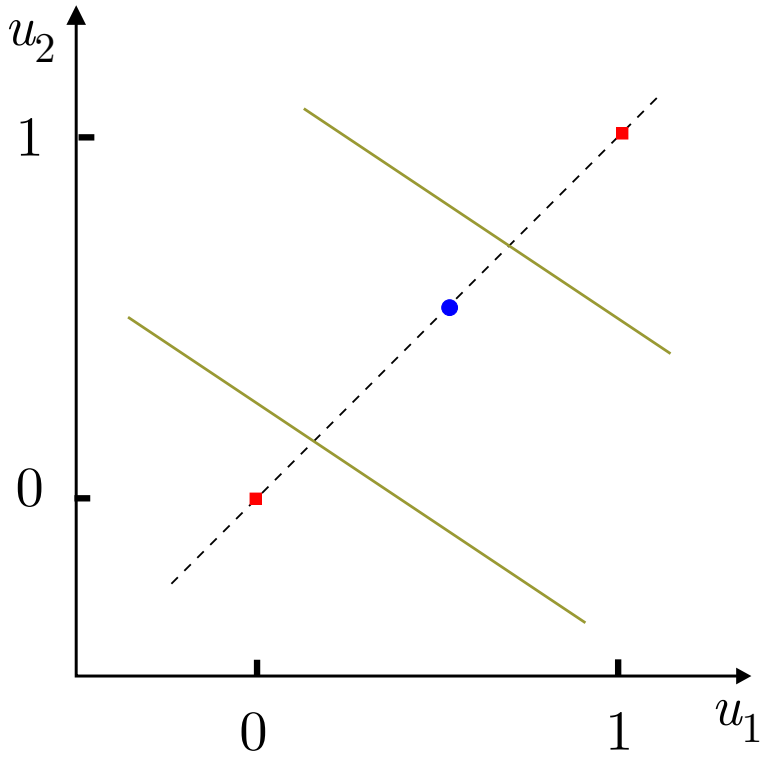
Dwa zbiory punktów nie są rozdzielne liniowo



Potrzebne są co najmniej 2 linie do rozróżnienia tych zbiorów

Separowalność liniowa

Punkty nie będące w położeniu ogólnym



Nie można rozgraniczyć tych punktów bo leżą na jednej linii

Jednostki liniowe

- Zamiast funkcji aktywacji w postaci skoku jednostkowego można zastosować funkcję liniową
- Funkcja kosztu przyjmuje postać funkcji ciągłej i różniczkowalnej – do jej minimalizacji można zastosować cała gamę technik optymalizacji, np. metody gradientowe
- Dla jednostek z liniową funkcją aktywacji postaci

$$f(\varphi) = \varphi$$

wyjście neuronu dane jest zależnością

$$y^{\mu} = \sum_{k=1}^{K} w_k u_k^{\mu}$$

gdzie u_k^{μ} – μ-ty wzorzec wejściowy na k-tym wejściu, w_k – waga k-tego połączenia synaptycznego, y^{μ} – odpowiedź neuronu na μ-ty wzorzec wejściowy

Jednostki liniowe

- Należy znaleźć regułę uczenia zdolną do znalezienia takiego zbioru wag aby odpowiedź neuronu była jak najbardziej zbliżona do wartości zadanej
- Definiuje się funkcję kosztu

$$J = \frac{1}{2}\sum_{\mu}(y_z^{\mu} - y^{\mu})^2$$

- Do minimalizacji J można użyć metody gradientowej najszybszego spadku

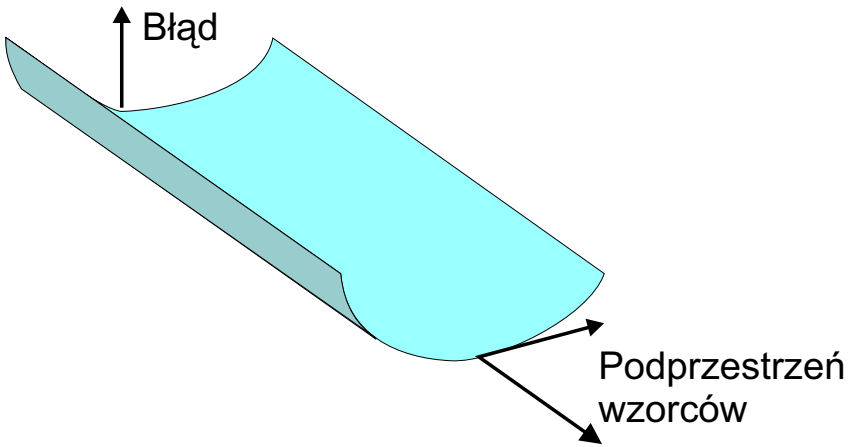
$$\Delta w_k = -\eta\frac{\partial J}{\partial w_k} = -\eta\frac{\frac{1}{2}\sum_{\mu}\partial(y_z^{\mu} - y^{\mu})^2}{\partial w_k} = \eta\sum_{\mu}(y_z^{\mu} - y^{\mu})\frac{\partial y^{\mu}}{\partial w_k}$$

$$= \eta\sum_{\mu}(y_z^{\mu} - y^{\mu})\frac{\partial\sum_j w_j u_j^{\mu}}{\partial w_k} = \eta\sum_{\mu}(y_z^{\mu} - y^{\mu})u_k^{\mu} = \eta\sum_{\mu} e^{\mu}u_k^{\mu}$$

- Powyższe wyrażenie nazywa się *regułą delty* (adaline, Widrowa-Hoffa, najmniejszych kwadratów)

Jednostki liniowe

- Funkcja kosztu jest formą kwadratową wejść (suma kwadratów błędów)
- W podprzestrzeni wzorców jej powierzchnia jest paraboliczną czaszą z pojedynczym minimum
- W przypadku gdy wejścia są liniowo niezależne, minimum występuje dla $J = 0$



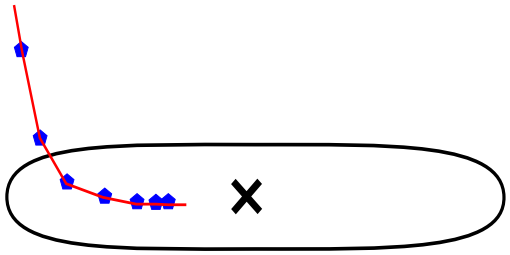
Jednostki liniowe

- Reguła delty bezwzględnie zmniejsza błąd, jeżeli η jest dostatecznie małe

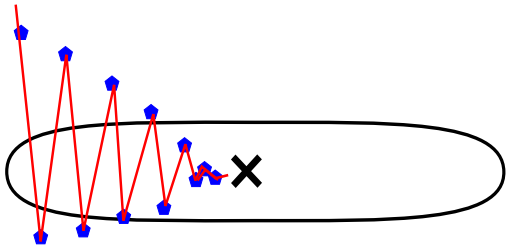
Przykład 2

Funkcja $J = x^2 + 20y^2$

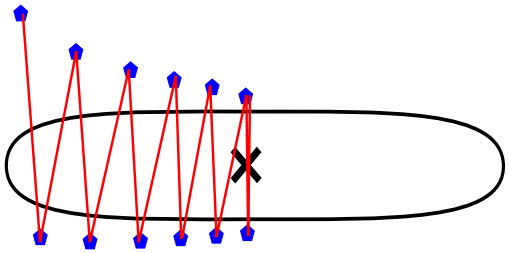
$\eta = 0,02$



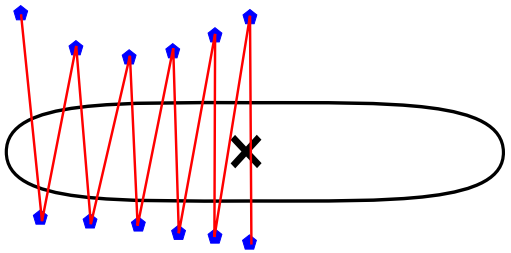
$\eta = 0,0476$



$\eta = 0,049$



$\eta = 0,0505$



Jednostki nieliniowe

- Właściwości neuronu można polepszyć poprzez wprowadzenie nieliniowej funkcji aktywacji
- Funkcja kosztu przyjmuje postać jak dla jednostek liniowych
- Reguła aktualizacji

$$\Delta w_k = -\eta \frac{\partial J}{\partial w_k} = -\eta \frac{\frac{1}{2}\sum_\mu \partial (y_z^\mu - y^\mu)^2}{\partial w_k} = \eta \sum_\mu (y_z^\mu - y^\mu) \frac{\partial y^\mu}{\partial w_k}$$

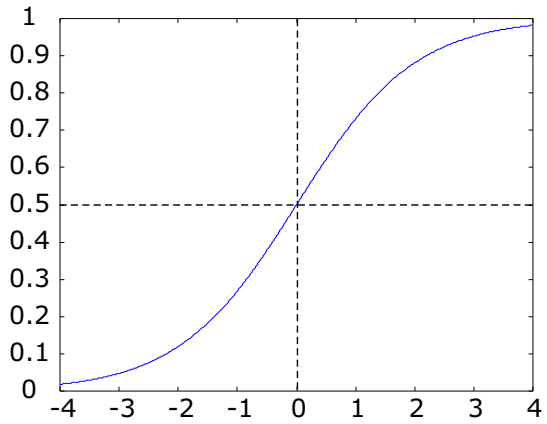
$$= \eta \sum_\mu (y_z^\mu - y^\mu) \frac{\partial f(\sum_j w_j u_j^\mu)}{\partial w_k} = \eta \sum_\mu (y_z^\mu - y^\mu) f'(y^\mu) u_k^\mu$$

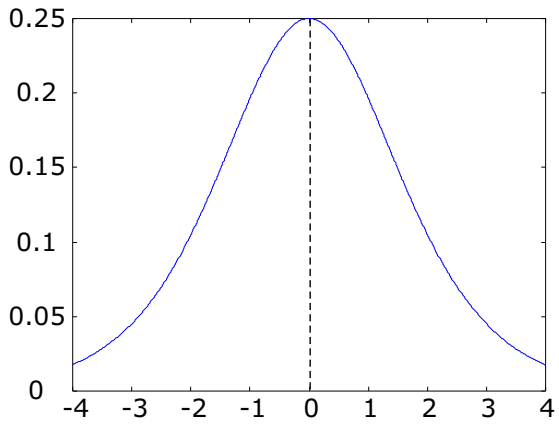
$$\Longrightarrow \Delta w_k = \eta \sum_\mu e^\mu f'(y^\mu) u_k^\mu$$

- Powyższą regułę nazywa się *uogólnioną regułą delty*.

Jednostki nieliniowe

- Dla funkcji sigmoidalnej





- Pochodna f' ma wartość największą, gdy potencjał membranowy ma wartość małą
- Największe zmiany wag następują gdy na wejściu jednostek pojawiają się małe sygnały, tzn. gdy istnieją wątpliwości dotyczące sygnału na wejściu

Jednostki nieliniowe

- Dla funkcji sigmoidalnej

$$f(x) = \frac{1}{1 + \exp(x)}$$

jej pochodna może zostać wyznaczona

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

upraszcza to złożoność obliczeniową reguły uczenia

- W przypadku zastosowania jednostek nieliniowych funkcja kosztu staje się funkcją multimodalną i reguła może utknąć w jednym z minimów lokalnych

Pojemność perceptronu prostego

- Problem: Ile losowych par wejście-wyjście można w niezawodny sposób zmagazynować w sieci o danym rozmiarze?
- Dla jednostek ciągłych liniowych i nieliniowych odpowiedź jest prosta. Jeżeli wybierze się p wzorców losowych, to będą one liniowo niezależne gdy $p \leqslant N$. Stąd pojemność maksymalna $p_{max} = N$ (N — liczba jednostek wejściowych)
- Dla jednostek progowych odpowiedź jest trudniejsza ze względu na separowalność liniową. Odpowiedź dla ciągłych losowych wejść była wyprowadzona przez Covera i wynosi $p_{max} = 2N$ $(*)$
- Liczba jednostek wyjściowych może być mała i ustalona (niezależnie od N). Precyzyjniej, równanie $(*)$ jest prawdziwe w granicy dla $N \to \infty$