

Podstawy sieci neuronowych

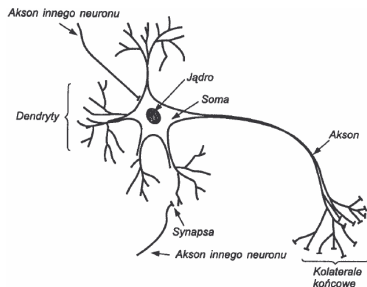
Aleksander Byrski

Katedra Informatyki AGH Kraków

- Lata 40-te XX w. początek, pierwszy model neuronu (McCulloch, Pitts), reguła uaktualniania wag połączeń (Hebb).
- Lata 50-te/60-te XX w. Perceptron (Rosenblatt, Wightman), ADALINE (Widrow Hoff).
- Lata 60-te/70-te okres zastouju - wykrycie ograniczeń perceptronów jednowarstwowych (Minsky, Papert).
- Lata 80-te ponowny rozkwit, prace Hopfielda, odkrycie metody uczenia perceptronów wielowarstwowych (McClelland, Rumelhard, ale: Werbos opublikował podobną metodę w 1974).

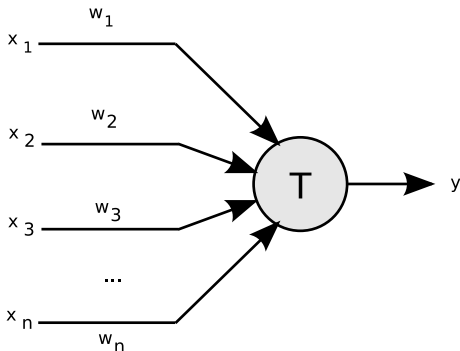
- Rozpoznawanie obrazów.
- Rozpoznawanie i synteza mowy.
- Kompresja obrazów.
- Prognozowanie sprzedaży, giełdy, wyścigów konnych.
- Interpretacja badań biologicznych i medycznych.
- Diagnostyka układów elektronicznych.
- Dobór pracowników.
- Selekcja celów śledztwa w kryminalistyce.

- Objętość 1400cm^3 , średnia masa $1,5\text{kg}$ (w większości woda).
- Kora mózgowa grubości 3mm zawiera 10^{10} komórek nerwowych i 10^{12} komórek glejowych.
- Liczba połączeń 10^{15} , dystans $0,01\text{ mm}$ do 1 m .
- Przekazywanie informacji – impulsy elektryczne $1 - 100\text{Hz}$, czas trwania $1 - 2\text{ms}$, napięcie 100mV , szybkość propagacji $1 - 100\text{m/s}$.
- Szacowana szybkość pracy mózgu 10^{18} operacji na sekundę, najszybsze komputery – 10^{14} operacji na sekundę.
- Typowa operacja wymaga co najwyżej 100 kroków, czas reakcji co najmniej 300ms .
- Pojemności informacyjne kanałów zmysłów: wzrok 100Mb/s , dotyk 1Mb/s , słuch 15kb/s , węch 1kb/s , smak 100b/s .



- Neuron przekazuje informacje zakodowane w postaci impulsów nerwowych.
- Impuls nerwowy to przesuwanie się fali depolaryzacji od miejsca pobudzenia do zakończeń neuronu, przekazywane są pomiędzy neuronami za pomocą dendrytów (w kierunku ciała neuronu) i aksonu (od ciała neuronu).
- Synapsa miejsce styku błony komórkowej zakończenia aksonu z błoną komórkową neuronu.

Neuron McCullocha-Pittsa



Prosty sumator z następującą regułą pobudzenia:

$$y = \begin{cases} 1 & \text{gd}y \sum_{i=1}^n w_i x_i \geq T \\ 0 & \text{gd}y \sum_{i=1}^n w_i x_i < T \end{cases}$$

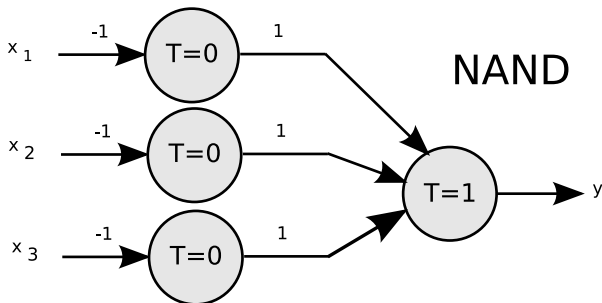
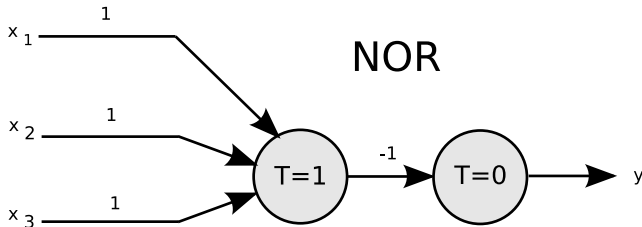
gdzie:

$$w_i = \pm 1$$

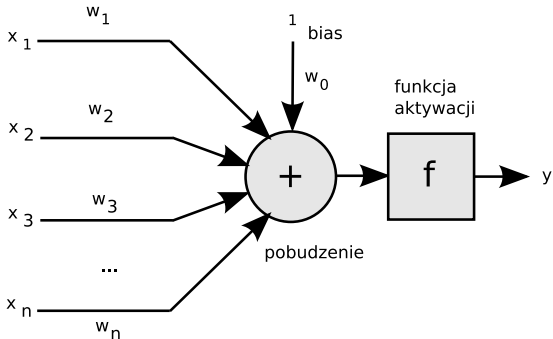
$$i = 1, 2, \dots, n$$

Brak procesu uczenia, wagi synaptyczne dobierane arbitralnie.

Neurony McCullocha Pittsa jako proste sumatory logiczne



Perceptron (Rosenblatt)



Funkcja przejścia:

$$y = f(\sum_{i=1}^n w_i x_i + w_0)$$

lub:

$$y = f(\sum_{i=0}^n w_i x_i)$$

gdzie:

f – funkcja aktywacji

$i = 0, 1, 2, \dots, n$

Uczenie neuronu/sieci neuronowej

- Jest to proces dobrania wag synaptycznych, najczęściej wykonywany iteracyjnie.
- Uczenie z nauczycielem – istnieje nadzorca, który porównuje otrzymaną odpowiedź, z odpowiedzią pożądaną (np. metody największego spadku, wsteczna propagacja błędów).
- Cykl pracy neuronu/sieci uczonej z nauczycielem: prezentacja danych na wejściu, obliczenie odpowiedzi neuronu/sieci, porównanie z odpowiedzią pożądaną, modyfikacja wag synaptycznych na podstawie błędu.
- Uczenie bez nauczyciela – neuron/sieć ma za zadanie wykryć istniejące związki w prezentowanych danych i samoistnie dostroić wagi synaptyczne (np. reguła Instar, reguła Hebba).
- Cykl pracy neuronu/sieci uczonej bez nauczyciela: prezentacja danych na wejściu, obliczenie odpowiedzi neuronu/sieci, modyfikacja wag synaptycznych na podstawie pewnej funkcji danych wejściowych i wyjściowych.

Uczenie perceptronu

W kolejnym (k-tym) kroku uczenia oblicza się tzw. błąd średniokwadratowy

$$E^k(\mathbf{w}) = \frac{1}{2} \sum_j (d_j - y_j^k)^2 = \frac{1}{2} \sum_j (d_j - f(\sum_i w_{ij} x_{ij}))^2$$

czyli różnicę między pożądanymi a otrzymanymi wartościami na wyjściu. Błąd ten może być liczony dla wielu kroków (aktualizacja wag synaptycznych może przebiegać po każdym kroku lub po serii):

$$E(\mathbf{w}) = \sum_k E^k(\mathbf{w})$$

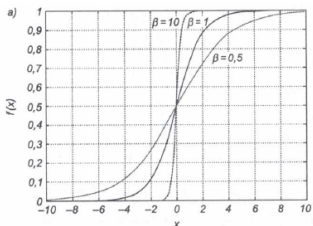
Przesunięcie w przestrzeni wag realizowane jest w kierunku maksymalnego spadku funkcji błędu, określonym przez ujemny gradient:

$$-\frac{\delta E^k}{\delta w_{ij}} = -2 \frac{1}{2} (d_j - y_j^k) x_{ij}^k \frac{\delta f}{\delta w_{ij}}$$

Przytoczona formuła zależy od wartości wejściowych, wyjściowych oraz pochodnej funkcji aktywacji.

Funkcja aktywacji

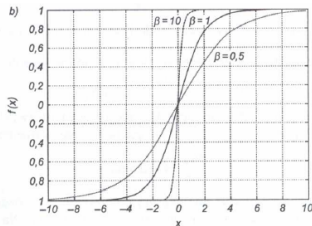
Początkowo stosowano progowe funkcje aktywacji. Ze względu na niemożność policzenia pochodnych, zaczęto stosować aproksymacje nieliniowe tych funkcji:



unipolarna (logistyczna)

$$f(x) = \frac{1}{1 + e^{-\beta x}}$$

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



bipolarna

$$f(x) = \tanh(\beta x)$$

$$f'(x) = f(x)(1 - f^2(x))$$

Pochodne tych funkcji są bardzo łatwe do policzenia.

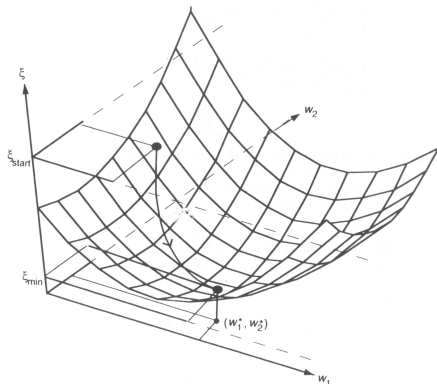
W procesie uczenia, zmiana wag neuronów obliczana jest wg następujących wzorów. Dla sigmoidy unipolarnej:

$$\Delta w_{ij} = -\eta(y_i - d_i)(1 - y_i)y_i x_j$$

Dla sigmoidy bipolarnej:

$$\Delta w_{ij} = -\eta(y_i - d_i)(1 - y_i^2)y_i x_j$$

η jest współczynnikiem szybkości uczenia, jego wartość steruje wielkością kroku wykonywanego w jednej iteracji uczenia (zwykle $\eta \in (0, 05; 2)$).



- Płaszczyzna obrazuje kształt funkcji błędów opisanej w przestrzeni wag.
- Współrzędne startowe to (w_1, w_2) , wartość funkcji błędów wynosi J_{start}
- Po wykonaniu odpowiedniej liczby kroków osiąga się ekstremum funkcji błędów (w_1^*, w_2^*) , dla którego wartość funkcji błędów wynosi J_{min} .

Współczynnik bezwładności (momentum)

Wielkość aktualnej zmiany wagi synaptycznej zależy od jej ostatniej wartości:

$$\Delta w_{ij}(t+1) = -\eta(y_i - d_i)(1 - y_i)y_i x_j + \alpha \Delta w_{ij}(t)$$

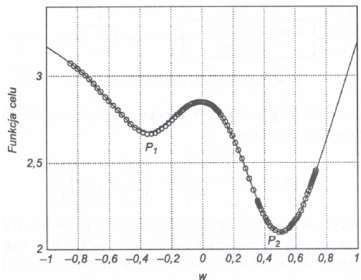
w przypadku, gdy obserwuje się niewielkie zmiany błędu sieci:

$$e_i(t+1) < 1,05e_i(t) \Rightarrow \alpha \neq 0$$

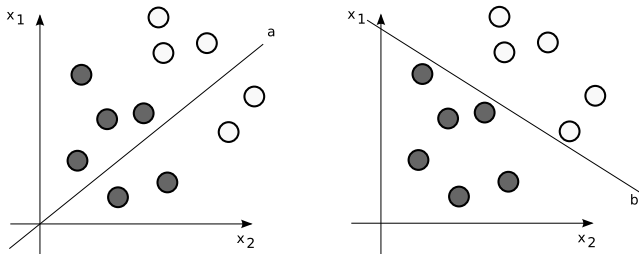
w przeciwnym przypadku:

$$e_i(t+1) \geq 1,05e_i(t) \Rightarrow \alpha = 0$$

Zwykle $\alpha \in (0, 1)$.



Pojedynczy neuron dzieli płaszczyznę danych wejściowych za pomocą prostej decyzyjnej (hiperpodpłaszczyzny o 1 wymiar mniejszej).



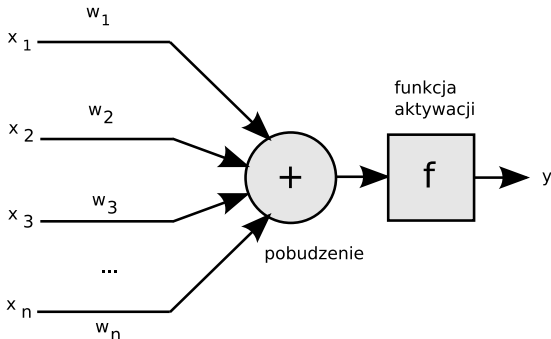
W przypadku braku biasu (zwanego też poziomem odniesienia) wszystkie proste decyzyjne muszą przechodzić przez początek układu współrzędnych:

$$a: 0 = w_1 x_1 + w_2 x_2$$

dzięki dodatkowej synapsie możliwe jest przesunięcie prostej, i odseparowanie danych współliniowych względem początku układu współrzędnych:

$$b: 0 = w_1 x_1 + w_2 x_2 + w_0$$

Instar (Grossberg)



Funkcja przejścia:

$$y = f(\sum_{i=1}^n w_i x_i)$$

gdzie:

f – funkcja aktywacji (zwykle liniowa)

$$i = 0, 1, 2, \dots, n$$

Neurony instar uczone są zwykle bez nauczyciela.

Dane wejściowe powinny być znormalizowane, czyli: $\forall \mathbf{x}, \|\mathbf{x}\| = 1$, można to uzyskać następująco:

$$\mathbf{x} = [x_1, \dots, x_n], x_j = \frac{x_j}{\sqrt{x_1^2 + \dots + x_n^2}}, j \in \langle 1, n \rangle$$

Neuron uczony jest za pomocą następującej reguły (Grossberga):

$$\Delta w_{ij} = \eta y_i (x_j - w_{ij})$$

wielkość modyfikacji wagi maleje do 0, gdy \mathbf{w} dąży do \mathbf{x} .

Interpretacja geometryczna działania instar

W przypadku, gdy neuron został wytrenowany do rozpoznania wzorca x_1 , zachodzi:

$$\mathbf{w} = [w_{i1}, \dots, w_{in}]^T = x_1$$

Po prezentacji innego wektora na wejściu (x_2) zachodzi:

$$y = \mathbf{w}^T x_2 = x_1^T x_2 = ||x_1|| ||x_2|| \cos \varphi_{12}$$

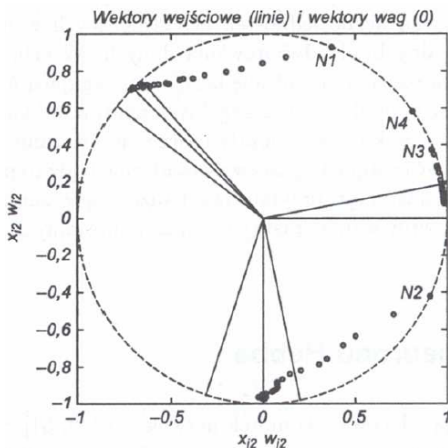
ponieważ wektory wejściowe są znormalizowane:

$$y = \cos \varphi_{12}$$

odpowiedź neuronu jest proporcjonalna do cosinusa kąta między x_1 a x_2 .
Neuron nie jest trenowany jest w celu zapamiętania jednego wzorca, lecz w celu uśrednienia wielu (klastrowanie).

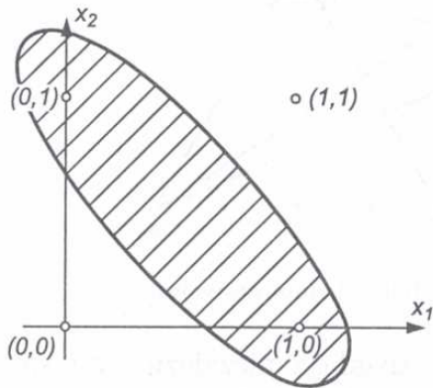
W trybie odtwarzania, odpowiedź neuronu oznacza jak daleko prezentowany wektor jest odległy od środka klastra prezentowanego przez neuron.

Wizualizacja uczenia neuronów instar



- Neurony N1, N2, N3, N4 rozpoczynają naukę z losowych miejsc w przestrzeni wag.
- Po odpowiedniej liczbie iteracji wagi N1 oraz N2 zostają ustawione w centrach dwóch klastrów danych.
- Efektem inicjacji wag jest ustawienie się wag N3 oraz N4 w centrum trzeciego klastra danych.
- Po prezentacji danych zbliżonych do dowolnego z klastrów, najbardziej pobudzone zostaną neurony, których wagi są ustawione w jego centrum.

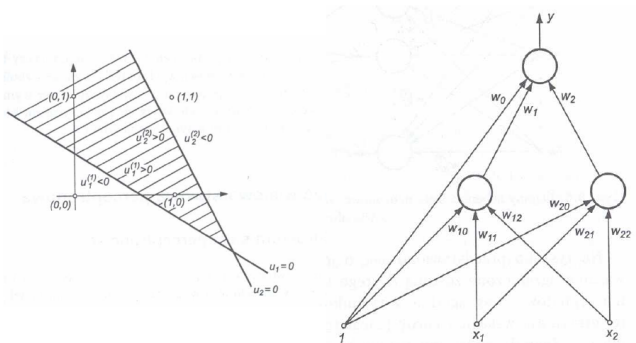
Problem XOR



| | | | | |
|-------|---|---|---|---|
| x_1 | 0 | 0 | 1 | 1 |
| x_2 | 0 | 1 | 0 | 1 |
| d | 0 | 1 | 1 | 0 |

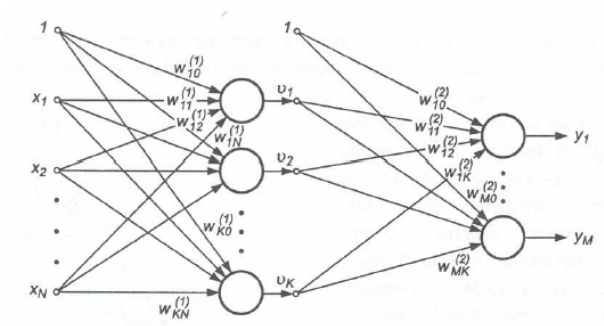
Symulacja działania prostego funktora logicznego XOR przerasta możliwości pojedynczego neuronu, oraz pojedynczej warstwy neuronów (nie jest to problem liniowo separowalny).

Rozwiązanie problemu XOR



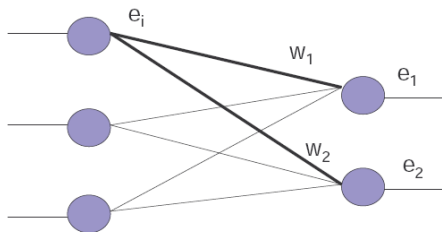
Rozwiązanie polega na wprowadzeniu dodatkowego neuronu, oraz dodatkowej warstwy z neuronem pełniącym rolę sumatora logicznego dla zbiorów odseparowanych przez dwa neurony z warstwy pierwszej.

Perceptron wielowarstwowy



Zwyczajowo złożony z neuronów sigmoidalnych. Pierwsza warstwa: buforująca, kolejne warstwy: ukryte, ostatnia: wyjściowa.


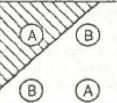
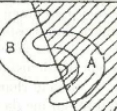
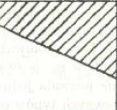

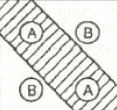
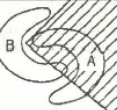
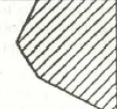

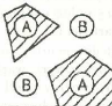
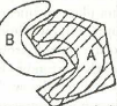
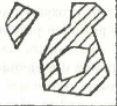
Algorytm wstecznej propagacji błędów



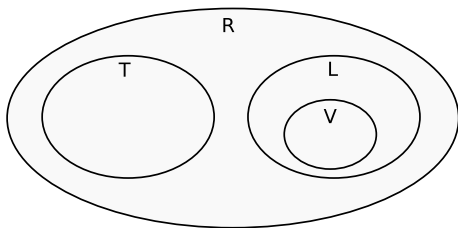
$$e_i = \sum_{k \in WY} w_k i e_k$$

Błąd z wyjścia sieci przekazywany jest proporcjonalnie do neuronów poprzedniej warstwy tak, aby również tam była możliwość policzenia błędu, gradientu i modyfikacji wag synaptycznych.

Uniwersalna aproksymacja

| Structure | Type of Decision Regions | Exclusive-OR Problem | Classes with Mesned Regions | Most General Region Shapes |
|---|---|---|---|--|
| Single-layer  | Half plane bounded by hyperplane |  |  |  |
| Two-layers  | Convex open or closed regions |  |  |  |
| Three-layers  | Arbitrary (Complexity limited by number of nodes) |  |  |  |

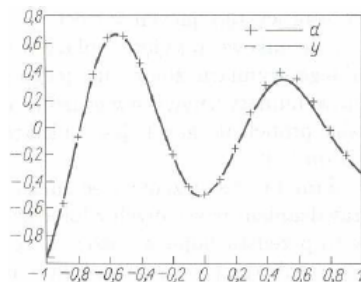
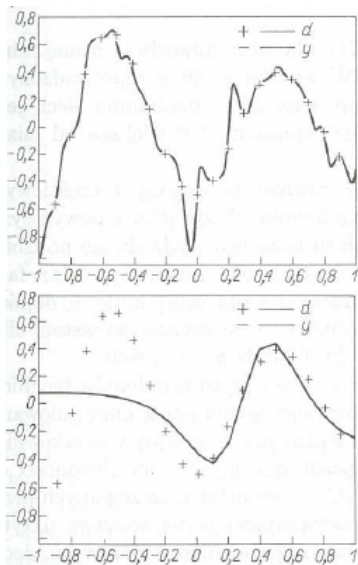
Udowodniono, że trójwarstwowy perceptron jest uniwersalnym aproksymatorem, co oznacza że może nauczyć się dowolnego odwzorowania z dowolną dokładnością o ile posiada odpowiednią liczbę neuronów.



- R - zbiór wszystkich możliwych danych wejściowych.
- L - zbiór danych uczących.
- V - zbiór danych weryfikujących.
- T - zbiór danych testujących.

Sieć posiada zdolność generalizacji, jeśli po procesie uczenia za pomocą danych ze zbioru L jest w stanie prawidłowo zaklasyfikować dane należące do zbioru T (które nie zostały wcześniej zaprezentowane). Zwykle $\frac{\#T}{\#L} \approx \frac{1}{5}$.

Niedouczenie i przeuczenie sieci



Prezentowane przypadki:

- Przeuczenie (za duża liczba neuronów).
- Niedouczenie (za mała liczba neuronów).
- Prawidłowo nauczona sieć (generalizująca).

- S. Osowski „Sieci neuronowe do przetwarzania informacji”.
- S. Osowski „Sieci neuronowe w ujęciu algorytmicznym”.
- R. Tadeusiewicz „Sieci neuronowe”.
- S. Haykin „Neural Networks – A Comprehensive Foundation”.