

Sieć jednowarstwowa

1 Funkcje aktywacji

Dotychczas rozważaliśmy perceptron ze skokową funkcją aktywacji. Oczywiście jest to tylko jedna z wielu możliwości. Funkcje aktywacji możemy podzielić przede wszystkim na ciągłe/dyskretne oraz unipolarne/bipolarne.

Najczęściej spotykane funkcje aktywacji (dla $net = \sum_i w_i x_i - \theta$):

- **signum:**

$$f(net) = \text{sgn}(net)$$

- **progowa:**

$$f(net) = \begin{cases} 1 & \text{dla } net \geq 0 \\ 0 \text{ (lub } -1) & \text{w.p.p.} \end{cases}$$

- **sigmoidalna (unipolarna):**

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

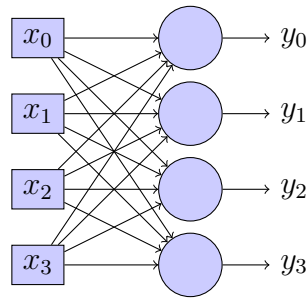
- **sigmoidalna (bipolarna):**

$$f(net) = \frac{2}{1 + e^{-net}} - 1$$

- **liniowa:**

$$f(net) = net$$

2 Sieć jednowarstwowa



Rysunek 1: Przykład sieci 1-warstwowej z czterema wejściami i czterema wyjściami.

Perceptron jako klasyfikator może dzielić na co najwyżej dwie klasy. W przypadku większej liczby musimy użyć wielu perceptronów, z których budujemy warstwę.

W sieci neuronowej klasy możemy reprezentować na dwa sposoby:

- **Lokalnie:** każdy neuron reprezentuje jedną klasę. Dla każdego wektora wejściowego wartość 1 powinien mieć tylko jeden neuron warstwy wyjściowej, a pozostałe wartość 0. Dla danego przykładu wybieramy klasę reprezentowaną przez neuron z najwyższą aktywacją.
- **Globalnie:** każdej klasie przypisana jest jedna kombinacja wyjść. K neuronów warstwy wyjściowej może reprezentować 2^K klas.

Dla sieci jednowarstwowej wprowadzamy następującą notację:

- wektor wejść: $\mathbf{x} = (x_0 \ \dots \ x_J)^T$
- wektor wyjść: $\mathbf{y} = (y_0 \ \dots \ y_K)^T$
- macierz wag:

$$\mathbf{W} = \begin{pmatrix} w_{11} & w_{12} & \dots & w_{1J} \\ w_{21} & w_{22} & \dots & w_{2J} \\ \vdots & \vdots & \ddots & \vdots \\ w_{K1} & w_{K2} & \dots & w_{KJ} \end{pmatrix}$$

gdzie w_{kj} to waga j -ego wejścia k -tego neuronu.

- wektor odchyłeń: $\boldsymbol{\theta} = (\theta_0 \ \dots \ \theta_K)^T$, gdzie θ_k to odchylenie k -tego neuronu.
- Macierz funkcji aktywacji $\boldsymbol{\Gamma} = \text{diag}[f(\cdot)]$. Mnożenie tej macierzy przez wektor jest równoważne z wyznaczeniem wartości funkcji dla każdego elementu.

Wartość wyjściową sieci obliczamy jako:

$$\mathbf{y} = \boldsymbol{\Gamma}[\mathbf{W}\mathbf{x} - \boldsymbol{\theta}]$$

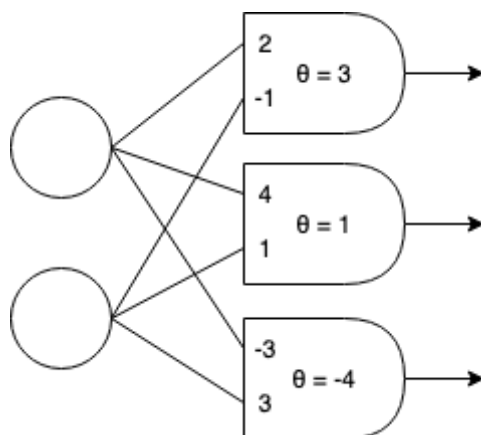
Reguła delta dla warstwy:

$$\mathbf{W}' = \mathbf{W} + \alpha(\mathbf{d} - \mathbf{y})\mathbf{x}^T$$

$$\boldsymbol{\theta}' = \boldsymbol{\theta} - \alpha(\mathbf{d} - \mathbf{y})$$

Zadania

Zadanie 1.



Dla powyższej sieci jednowarstwowej, w której neurony mają progową funkcję aktywacji:

1. Zapisz macierz wag.
 2. Wykonaj jeden krok uczenia dla następujących wektorów wejściowych i oczekiwanych wektorów wyjściowych ($\alpha = 1$).
- $\mathbf{x} = (3, 1)$, $\mathbf{d} = (0, 0, 1)$
 - $\mathbf{x} = (2, 2)$, $\mathbf{d} = (1, 0, 1)$
 - $\mathbf{x} = (2, -8)$, $\mathbf{d} = (1, 1, 1)$
 - $\mathbf{x} = (0, 1)$, $\mathbf{d} = (1, 1, 0)$

Mini-projekt: Sieć jednowarstwowa

Celem projektu jest stworzenie sieci jednowarstwowej identyfikującej język, w jakim napisany jest tekst wejściowy.

W pliku `lang.zip` znajduje się zbiór tekstów w trzech językach – angielskim, polskim i niemieckim. Aby zaklasyfikować dany tekst należy zliczyć częstotliwość występowania każdej z liter alfabetu łacińskiego. Na potrzeby tego zadania ignorujemy wszystkie nie-standardowe litery (polskie znaki, etc.). Zliczamy tylko częstotliwości 26 podstawowych liter alfabetu, pomijając wszystkie inne znaki.

Dla tekstu wejściowego generujemy 26-elementowy wektor wejściowy zawierający liczbę wystąpień każdej z liter. Następnie należy znormalizować wektor:

$$\hat{\mathbf{v}} = \frac{\mathbf{v}}{|\mathbf{v}|}.$$

Wyjście sieci powinno mieć reprezentację lokalną: do każdego neuronu przypisujemy jeden z języków. Dla danego tekstu wartość wyjściową 1 powinien mieć neuron reprezentujący język tekstu, a pozostałe wartości 0. Aby klasyfikować język tekstu wejściowego, wybieramy neuron z maksymalną aktywacją. Możemy wykorzystać neurony ze skokową funkcją aktywacji (w tym wypadku możemy użyć implementacji z ostatniego projektu), lub sigmoidalną.

W pliku `lang.test.zip` znajduje się zbiór testowy, którym testujemy wytrenowaną sieć.

Program powinien zapewniać interfejs umożliwiający wklejenie tekstu i rozpoznanie jego języka.

Termin oddania: dwa tygodnie.