

# Sieć jednowarstwowa

## 1 Funkcje aktywacji

Funkcje aktywacji możemy podzielić przede wszystkim na ciągłe/dyskretne oraz unipolarne/bipolarne.

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

- **signum:**

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

- **progowa:**

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

- **sigmoidalna (unipolarna):**

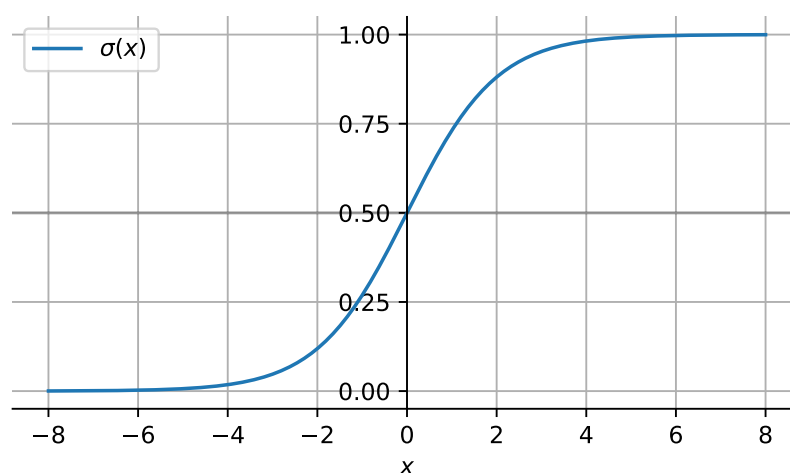
$$f(net) = \frac{1}{1 + e^{-net}}$$

- **sigmoidalna (bipolarna):**

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

- **liniowa:**

$$f(net) = net$$



Sigmoidalna funkcja aktywacji

## 2 Metoda gradientowa

Przyjmijmy następującą miarę błędu dla perceptronu z ciągłą funkcją aktywacji:

$$E = \frac{1}{2}(d - y)^2 = \frac{1}{2}(d - f(net))^2 = \frac{1}{2}(d - f(\mathbf{w}^T \mathbf{x} - \theta))^2$$

Celem jest zmodyfikowanie wag tak, aby zminimalizować wartość funkcji błędu. Metoda gradientowa polega na przesuwaniu wag w kierunku odwrotnym do gradientu funkcji:

$$\begin{aligned}\mathbf{w}' &= \mathbf{w} - \alpha \nabla E(\mathbf{w}), \\ \theta' &= \theta - \alpha \frac{\partial E}{\partial \theta},\end{aligned}$$

gdzie

$$\begin{aligned}\nabla E(\mathbf{w}) &= -(d - y)f'(net)\mathbf{x}, \\ \frac{\partial E}{\partial \theta} &= (d - y)f'(net).\end{aligned}$$

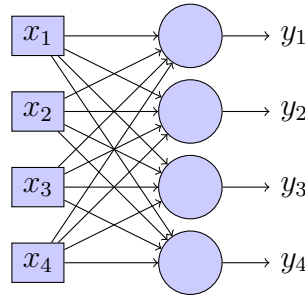
Pochodna funkcji sigmoidalnej to

$$\sigma'(x) = \sigma(x)(1 - \sigma(x)).$$

Wagi i odchylenie perceptronu z sigmoidalną funkcją aktywacji aktualizowane są zatem następujący sposób:

$$\begin{aligned}\mathbf{w}' &= \mathbf{w} + \alpha(d - y)\sigma'(net)\mathbf{x} = \mathbf{w} + \alpha(d - y)y(1 - y)\mathbf{x}, \\ \theta' &= \theta - \alpha(d - y)\sigma'(net) = \theta - \alpha(d - y)y(1 - y).\end{aligned}$$

### 3 Sieć jednowarstwowa



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

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:  $\mathbf{x} = (x_1 \ \dots \ x_J)^T$
- wektor wyjść:  $\mathbf{y} = (y_1 \ \dots \ y_K)^T$
- 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:  $\boldsymbol{\theta} = (\theta_1 \ \dots \ \theta_K)^T$ , gdzie  $\theta_k$  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:

$$\begin{aligned} \mathbf{W}' &= \mathbf{W} + \alpha(\mathbf{d} - \mathbf{y})\mathbf{x}^T \\ \boldsymbol{\theta}' &= \boldsymbol{\theta} - \alpha(\mathbf{d} - \mathbf{y}) \end{aligned}$$

## Zadania

### Zadanie 1.

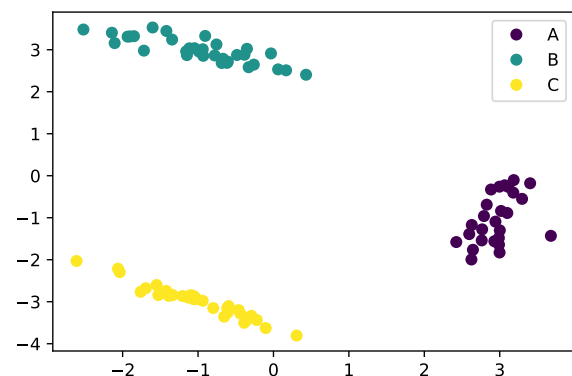
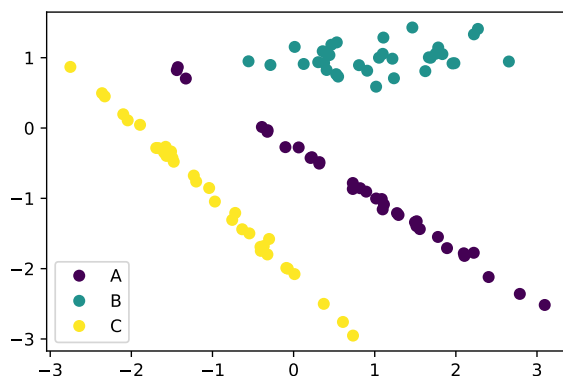
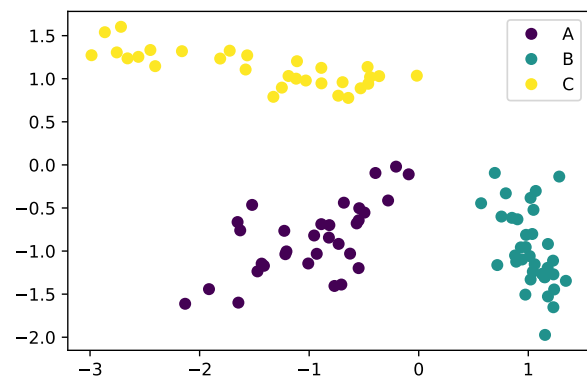
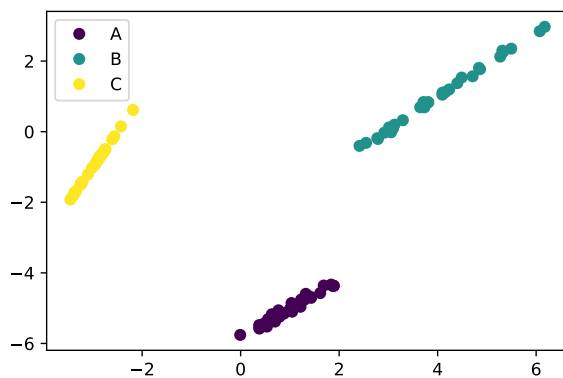
Dany jest perceptron z wektorem wag  $\mathbf{w} = (2, -1, 4, 1)$ , odchyleniem  $\theta = 3$ , oraz **sigmo-idealną** funkcją aktywacji. Oblicz wartość wyjściową dla następujących wektorów wejściowych i wykonaj jeden krok uczenia, zakładając podane oczekiwane wartości wyjściowe oraz stałą uczenia  $\alpha = 1$ .

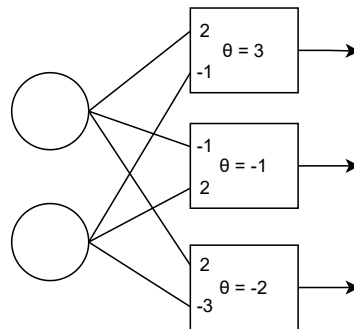
- $\mathbf{x} = (7, -2, -4, 3), d = 1$
- $\mathbf{x} = (1, 0, 1, -2), d = 0$

### Zadanie 2.

Zaprojektuj sieć jednowarstwową, która będzie klasyfikowała następujące zbiory.

- Jaka reprezentacja wyjścia sieci (lokalna/globalna) będzie odpowiednia dla każdego przykładu?
- Narysuj płaszczyznę decyzyjną każdego perceptronu.
- Zapisz mapowanie klas na wyjście sieci.



**Zadanie 3.**

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 plikach `lang.train.csv` i `lang.test.csv` znajduje się zbiór tekstów w czterech językach – angielskim, niemieckim, polskim i hiszpańskim. Aby zaklasyfikować dany tekst należy zliczyć częstotliwość występowania każdej z liter alfabetu łacińskiego. Na potrzeby tego zadania można zignorować wszystkie niestandardowe litery (znaki diakrytyczne, etc.) i zliczać tylko częstotliwości wystąpień 26 podstawowych liter alfabetu, pomijając wszystkie inne znaki.

Dla każdego tekstu wejściowego należy wygenerować 26-elementowy wektor zawierający liczbę wystąpień każdej z liter i następnie go znormalizować:

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

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 wartość 0.

Można wykorzystać funkcję aktywacji progową lub liniową ( $f(\text{net}) = \text{net}$ ) (w przypadku obu funkcji reguła modyfikacji wag jest identyczna, ponieważ dla funkcji liniowej  $f'(\text{net}) = 1$ ). Aby klasyfikować język tekstu, wybieramy perceptron z maksymalną aktywacją.

**Program powinien:**

- Trenować sieć danymi z pliku `lang.train.csv` i następnie wypisać dokładność klasyfikacji dla danych z pliku `lang.test.csv`.

- Dostosować się do dowolnego zbioru danych w podobnym formacie, np. z inną liczbą języków.
- Zapewniać interfejs umożliwiający wklejenie nowego tekstu (np. w konsoli) i rozpoznanie języka.
- (Opcjonalnie) wypisać te teksty ze zbioru testowego, dla których klasyfikacja była błędna.

**Wskazówki:**

- W tekstach mogą znajdować się przecinki, więc przy wczytywaniu plików nie działa dzielenie każdej linii `split(",")` (ale można użyć np. `split(",", 1)` (Python) lub `split(",", 2)` (Java)).
- Użycie liniowej funkcji aktywacji zmniejsza szanse na niejednoznaczną wartość wyjściową sieci (kiedy więcej niż jeden perceptron ma wartość wyjściową 1).
- Można wykorzystać implementację perceptronu z wcześniejszego projektu, lub implementować sieć od początku przy użyciu operacji na macierzach.