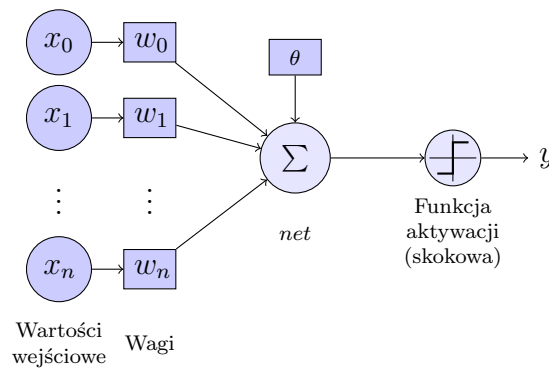


Perceptron

Semestr letni 2019/2020

1 Perceptron



Rysunek 1: Perceptron ze skokową funkcją aktywacji.

Perceptron składa się z:

- Wektora wejść $\mathbf{x} = (x_0 \ x_1 \ \dots \ x_n)^T$,
- Wektora wag $\mathbf{w} = (w_0 \ w_1 \ \dots \ w_n)^T$,
- Progu/odchylenia θ ,
- Funkcji aktywacji f .

1.1 Kalkulacja wartości wyjściowej

Wartość net to iloczyn skalarny wektora wag oraz wektora wejść pomniejszony o wartość odchylenia:

$$net = \mathbf{w}^T \mathbf{x} - \theta = w_0 \cdot x_0 + w_1 \cdot x_1 + \dots + w_n \cdot x_n - \theta$$

Wartość wyjściowa to:

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

2 Perceptron - interpretacja geometryczna

Perceptron tworzy hiperpłaszczyznę dzielącą przestrzeń decyzyjną na dwie części, której wektorem normalnym jest wektor wag. Wartość wyjściowa jest równa 1, kiedy punkt wejściowy znajduje się po tej samej stronie co wektor wag. Poniżej równanie hiperpłaszczyzny dla perceptronu z wektorem wag (w_0, w_1, \dots, w_n) i odchyleniem θ :

$$w_0x_0 + w_1x_1 + \dots + w_nx_n - \theta = 0$$

3 Reguła delta

Wagi i odchylenia modyfikujemy zgodnie z następującą regułą:

$$\begin{aligned}\mathbf{w}' &= \mathbf{w} + \eta(d - y)\mathbf{x} \\ \theta' &= \theta - \eta(d - y)\end{aligned}$$

Gdzie:

- \mathbf{w}' to nowy wektor wag;
- \mathbf{w} to stary wektor wag;
- θ' to nowa wartość odchylenia;
- θ to stara wartość odchylenia;
- η to współczynnik uczenia (zwykle między 0 a 1);
- d to oczekiwana wartość wyjściowa;
- y to otrzymana wartość wyjściowa;
- \mathbf{x} to wektor wejściowy.

4 Algorytm uczenia perceptronu

Dane wejściowe to zbiór treningowy składający się z wektorów wejść \mathbf{x}_i i oczekiwanych wartości wyjściowych d_i :

$$\{(\mathbf{x}_0, d_0), (\mathbf{x}_1, d_1), \dots, (\mathbf{x}_n, d_n)\}$$

1. Wybieramy wartość współczynnika uczenia η i losujemy wagi początkowe.
2. Dla każdego wektora \mathbf{x}_i ze zbioru uczącego:
 - (a) Wyznacz wartość wyjściową y .
 - (b) Zaktualizuj wagi oraz odchylenie zgodnie z regułą delta podaną powyżej.
3. Wyznacz wartość błędu iteracji (liczbę przykładów, dla których perceptron dał błędną odpowiedź):

$$E = \sum_{i=0}^n |d_i - y_i|$$

4. Wracamy do punktu 2, albo zakańczamy uczenie, kiedy E jest poniżej ustalonej granicy ($E < E_{max}$), lub po określonej liczbie iteracji.

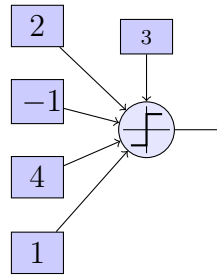
Zadania

Zadanie 1.

Dany jest następujący perceptron z wektorem wag $\mathbf{w} = (2, -1, 4, 1)$, odchyleniem $\theta = 3$, oraz unipolarną skokową funkcją aktywacji. Podaj wyjście perceptronu dla następujących wektorów wejściowych.

- $\mathbf{x} = (7, -2, -5, 4)$

- $\mathbf{x} = (2, 0, 2, 8)$



Zadanie 2.

Podaj równanie hiperpłaszczyzny dla perceptronu z Zadania 1.

Zadanie 3.

Zaprojektuj perceptron implementujący następujące funkcje logiczne: AND, OR, NAND, NOR, XOR.

Zadanie 4.

Jak zostaną zmodyfikowane wagi i odchylenie perceptronu z Zadania 1 dla podanych oczekiwanych wartości wyjściowych ($\eta = 0.5$).

- $\mathbf{x} = (7, -2, -5, 4), d = 1$

- $\mathbf{x} = (2, 0, 2, 8), d = 0$

- $\mathbf{x} = (5, 8, -1, -2), d = 1$

- $\mathbf{x} = (3, 1, 9, -3), d = 0$

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

Mini-projekt: Perceptron

W pliku `perceptron.data` znajdują się dane treningowe: zbiór *Iris* ograniczony do *Iris-setosa* i *Iris-versicolor*. W pliku `perceptron.test.data` znajduje się zbiór testowy. Zaimplementuj perceptron i wytrenuj, aby rozróżniał te dwa gatunki. Testuj zbiorem testowym i wypisz liczbę poprawnie zaklasyfikowanych przykładów.

Program powinien umożliwiać:

- Wczytanie dowolnego zbioru treningowego z pliku w formacie csv, gdzie ostatnia kolumna to atrybut decyzyjny. Powinien dostosowywać liczbę wag do wczytanego zbioru.
- Wybór stałej uczenia.
- Prosty interfejs pozwalający na podanie ręczne wektorów do klasyfikacji.

Termin oddania: 03-04-2019