

Wstęp do uczenia maszynowego

Perceptron

Tomasz Derek

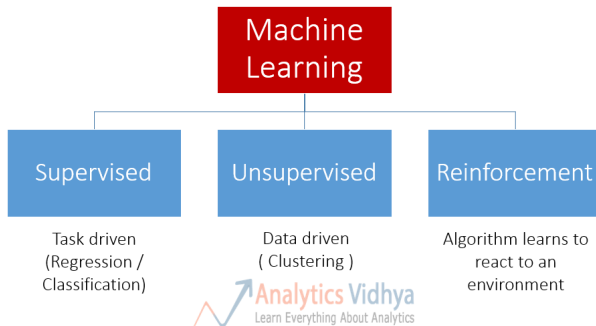
KMS

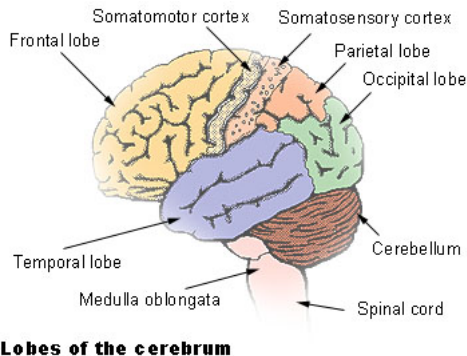
Październik 23, 2019

Typy uczenia

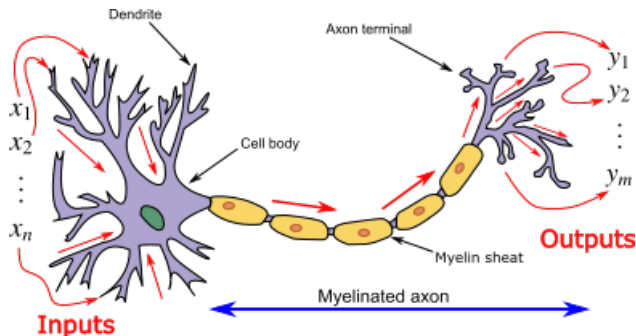
- Uczenie nadzorowane
- Uczenie nienadzorowane
- Ucznienie przez wzmacnianie

Types of Machine Learning

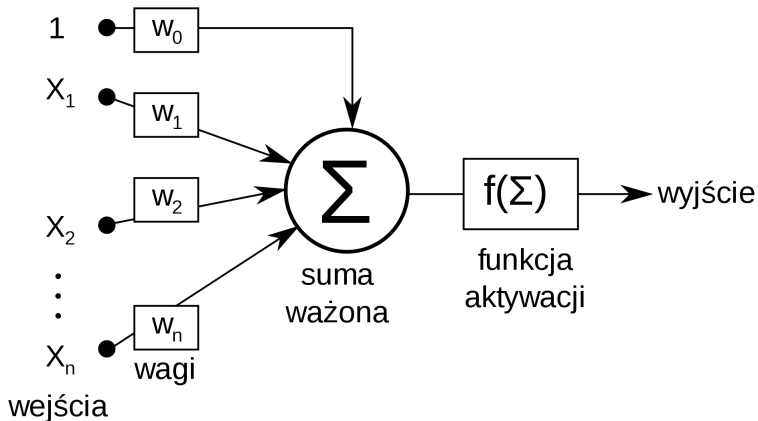




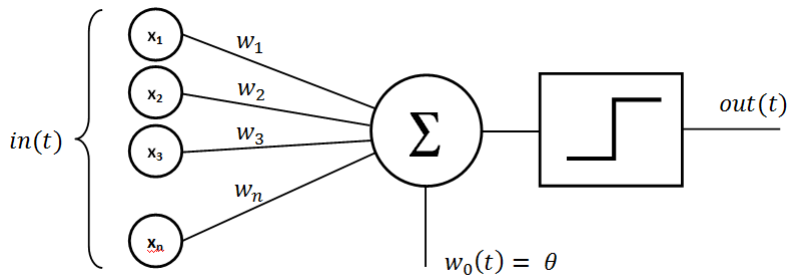
Biologiczny neuron



Neuron McCullocha Pittsa



Perceptron



Perceptronem nazywamy układ składający się z:

- wektor wejściowy $x = [x_1, x_2, \dots, x_n]$
- wektor wag w stworzony z wejściem, gdzie $w_1, \dots, w_n \in \mathbb{R}$
- funkcja aktywacji $f : \mathbb{R} \rightarrow \mathbb{R}$

Perceptron dla wejścia x zwraca

$$O(x_1, \dots, x_n) = f(\sum_{i=1}^n w_i x_i) = f(w^t x)$$

Pseudokod funkcji aktywacji progowej:

$$f(n) = \begin{cases} -1 & \text{if } x < \theta \\ +1 & \text{if } x \geq \theta \end{cases}$$

Dla wejścia $x = (x_1, x_2, \dots, x_n)$ perceptron progowy zwróci:

$$O(x_1, \dots, x_n) = \begin{cases} -1 & \sum_{i=1}^n w_i x_i < \theta \\ +1 & \sum_{i=1}^n w_i x_i \geq \theta \end{cases}$$

Pseudokod funkcji znakowej:

$$f(x) = \begin{cases} -1 & \text{if } x < 0 \\ +1 & \text{if } x \geq 0 \end{cases}$$

Relu:

$$f(n) = \begin{cases} 0 & \text{if } x \leq 0 \\ x & \text{if } x > 0 \end{cases}$$

Perceptron z biasem

- wektor wejściowy o rozmiarze n : x_1, \dots, x_n
- wektor wag o rozmiarze $n + 1$: w_0, w_1, \dots, w_n
- dodatkowe wejście $x_0 = +1$
- perceptron z biasem jest równoważny jednostce z progową funkcją aktywacji

Interpretacja geometryczna

Rozważmy jednostkę z funkcją progową.

Przypadek 1d - jedno wejście x_1 , jedna waga w_1 i próg θ

$$O(x_1) = \begin{cases} -1 & w_1 x_1 < \theta \iff x_1 < \theta/w_1 \\ +1 & w_1 x_1 \geq \theta \iff x_1 \geq \theta/w_1 \end{cases}$$

W ten sposób otrzymujemy punkt, który dzieli nam prostą na dwie półproste.

Interpretacja geometryczna

Rozważmy jednostkę z funkcją progową.

Przypadek 1d - jedno wejście x_1 , jedna waga w_1 i próg θ

$$O(x_1) = \begin{cases} -1 & w_1 x_1 < \theta \iff x_1 < \theta/w_1 \\ +1 & w_1 x_1 \geq \theta \iff x_1 \geq \theta/w_1 \end{cases}$$

W ten sposób otrzymujemy punkt, który dzieli nam prostą na dwie półproste.

Interpretacja geometryczna

Rozważmy jednostkę z funkcją progową.

Przypadek 2d - wejście x_1, x_2 , dwie wagi w_1, w_2 oraz próg θ

$$O(x_1) = \begin{cases} -1 & w_1 x_1 + w_2 x_2 < \theta \iff x_2 < \frac{-w_1}{w_2} x_1 + \frac{\theta}{w_2} \\ +1 & w_1 x_1 + w_2 x_2 \geq \theta \iff x_2 \geq \frac{-w_1}{w_2} x_1 + \frac{\theta}{w_2} \end{cases}$$

Czy nie wydaje się być podobne?

$$ax + by = c \iff y = -\frac{a}{b}x + \frac{c}{b}$$

Uczenie perceptronu

SPLA = Simple Perceptron Learning Algorithm

- Wylosuj wagi o wartościach bliskich 0
- Wybieramy losowy przykład E z danych uczących i odpowiadającą mu odpowiedź T
- Obliczamy pobudzenie sieci dla wybranego przykładu
- Obliczamy błąd
- Jeżeli błąd jest równy zero wróć do kroku 2
- W przeciwnym wypadku zaktualizuj wagi zgodnie ze wzorem

$$w_i = w_i + \eta * ERR * E$$

$$\theta = \theta - \eta * ERR$$

- Wróć do kroku 2

Uczenie perceptronu

Pocet Learning Algorithm

- Wylosuj wagi o wartościach bliskich 0, przypisujemy układowi wag zerowy czas życia i zapisujemy go w kieszonce
- Przebiegamy przykład **losując** z listy
- Obliczamy błąd
- Jeżeli błąd jest równy zero, zwiększ czas życia o jeden, jeżeli wynik jest lepszy od rekordzisty to obecny perceptron zostaje rekordzistą, w kieszonce zapisujemy wagi, a następnie wróć do kroku 2
- W przeciwnym wypadku zaktualizuj wagi zgodnie ze wzorem

$$w_i = w_i + \eta * ERR * E$$

$$\theta = \theta - \eta * ERR$$

Przypisz zerowy czas życia wróć do punktu 2.

- Zakończ po przebiegnięciu odpowiedniej liczby iteracji.

Uczenie perceptronu

Pocet Learning Algorithm with Ratchet

- Wylosuj wagi o wartościach bliskich 0, przypisujemy układowi wag zerowy czas życia i zapisujemy go w kieszonce
- Przebiegamy przykład **losując** z listy
- Obliczamy błąd
- Jeżeli jest to wynik lepszy od rekordzisty i i **klasyfikuje więcej przykładów niż rekordzista** to staje się nowym rekordzistą, a wagi zapisujemy.
- W przeciwnym wypadku zaktualizuj wagi zgodnie ze wzorem

$$w_i = w_i + \eta * ERR * E$$

$$\theta = \theta - \eta * ERR$$

Przypisz zerowy czas życia wróć do punktu 2.

- Zakończ po przebiegnięciu odpowiedniej liczby iteracji.