

# Wstęp do uczenia maszynowego

## Perceptron - przypomnienie

Tomasz Derek

KMS

Listopad 6, 2019

Perceptron - uczenie z nadzorowane

# Ogólny zapis perceptronu progowego

$$O(x_1, \dots, x_n) = f\left(\sum_{i=1}^n x_i w_i + w_0\right)$$

lub też

$$O(x_1, \dots, x_n) = f\left(\sum_{i=1}^n x_i w_i - \theta\right)$$

# Funkcja progowa

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

SPLA = Simple Perceptron Learning Algorithm

- Wylosuj wagi o wartościach bliskich 0
- Wybieramy losowy/kolejny 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

# Pocket Learning Algorithm

- Wylosuj wagi o wartościach bliskich 0, przypisujemy układowi wag zerowy czas życia i zapisujemy go w kieszonce
- Przebiegnij w sposób losowy po przykładach uczących
- Oblicz błąd sieci
- Jeżeli błąd jest równy zero, zwiększ czas życia o jeden, jeżeli wynik jest lepszy od rekordzisty to obecny perceptron zostaje rekordzistą, w kieszonce zapisz 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 i wróć do punktu 2.

- Zakończ po przebiegnięciu odpowiedniej liczby iteracji.

# Pocket Learning Algorithm with Ratchet

- Wylosuj wagi o wartościach bliskich 0, przypisujemy układowi wag zerowy czas życia i zapisujemy go w kieszonce
- Przebiegnij w sposób losowy po przykładach uczących
- Oblicz błąd sieci
- Jeżeli jest to wynik lepszy od rekordzisty i i **klasyfikuje więcej przykładów niż rekordzista**, to staje się nowym rekordzistą, zapisz jego 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 i wróć do punktu 2.

- Zakończ po przebiegnięciu odpowiedniej liczby iteracji.

# Czego będziemy dziś potrzebować?

ALGEBRY





# Czego będziemy dziś potrzebować?

Metody z modułu NumPy:

- `dot()`
- `random.randn()` lub `zeros()`

# Czego będziemy dziś potrzebować?

Metody z modułu DataLoader:

- `create_dataset()`
- `random_sample()`

Koniec?



## WHEN YOUR MODEL PERFORMS POOR ON TEST DATA

