

Wprowadzenie do uczenia maszynowego część I

Franciszek Górski 2021

Czym jest uczenie maszynowe?

- Uczenie maszynowe - proces poprawy wyników algorytmu wraz ze zdobywanym doświadczeniem, które to doświadczenie pozyskiwane jest z danych.
- Proces pozyskiwania doświadczenia nazywany jest uczeniem się.
- W czasie uczenia tworzony jest model na podstawie danych uczących.

Dane uczące

Jeżeli model ma zdobywać doświadczenie z danych, to dane muszą zawierać informację. Dlatego w Informatyce dane przedstawiane są w postaci zbioru parametrów np. - dane opisujące różne komputery

Pamięć RAM	Pamięć VRAM GPU	Liczba rdzeni CPU	Taktowanie CPU
8	0	4	2,4
16	6	8	2,5
8	2	8	2,3

Etykiety danych

Przedstawione dane mają z kolei przypisane etykiety. Załóżmy, że prezentowany przez nas zbiór ma służyć opisaniu cen komputerów w zależności od ich parametrów:

Pamięć RAM	Pamięć VRAM GPU	Liczba rdzeni CPU	Taktowanie CPU	Cena
8	0	4	2.4	1500
16	6	8	2.5	3000
8	2	8	2.3	1900

Dane uczące

Pamięć RAM	Pamięć VRAM GPU	Liczba rdzeni CPU	Taktowanie CPU
8	0	4	2.4
16	6	8	2.5
8	2	8	2.3

= **X**

Cena
1500
3000
1900

= **y**

X - macierz liczb rzeczywistych o wymiarach 3x4 (3 przykłady danych, każdy opisany 4 parametrami)

y - wektor liczb rzeczywistych 3x1

Dane uczące - nasze uproszczenie

Pamięć RAM
8
16
8

= X

X - macierz (wektor) liczb rzeczywistych o wymiarach 3×1 na nasze potrzeby założmy 1 parametr opisujący dane

y - wektor liczb rzeczywistych 3×1

Cena
1500
3000
1900

= y

Hipoteza $h(x)$

Przykładowa hipoteza dla naszego przypadku - cena laptopów na rynku:

$$h(x) = a \cdot x + b$$

Nasze dane:

$$X[0] = [8]$$

$$h(X[0]) = 8a + b$$

$$X[1] = [16]$$

$$h(X[1]) = 16a + b$$

$$X[2] = [8]$$

$$h(X[2]) = 8a + b$$

Inicjalizacja wag w hipotezie $h(x)$

Nasza hipoteza: $h(x) = a \cdot x_1 + b$

Nasze wyniki:

$$h(X[0]) = 8a + b$$

$$h(X[1]) = 16a + b$$

$$h(X[2]) = 8a + b$$

Inicjalizacja wag - wektor parametrów θ (theta)

$$\theta = [a, b]$$

$$\theta = [100, 150]$$

Etykiety danych:

$$y[0] = 1500$$

$$y[1] = 3000$$

$$y[2] = 1900$$

Wyniki predykcji:

$$y_pred[0] = 800 + 150 = 950$$

$$y_pred[1] = 1600 + 150 = 1750$$

$$y_pred[2] = 800 + 150 = 950$$



Jak zmierzyć błąd algorytmu? - funkcja kosztu $L(y, y_pred)$

Nasze wyniki predykcji:

$$y_pred[0] = 800 + 150 = 950$$

$$y_pred[1] = 1600 + 150 = 1750$$

$$y_pred[2] = 800 + 150 = 950$$

Przykładowa funkcja kosztu $l(y[i], y_pred[i])$:

$$l(y[i], y_pred[i]) = |y[i] - y_pred[i]|$$

lub

$$l(y[i], y_pred[i]) = (y[i] - y_pred[i])^2$$

Posiadane etykiety danych:

$$y[0] = 1500$$

$$y[1] = 3000$$

$$y[2] = 1900$$

MSE (mean square error):

$$L(y, y_pred) = 1/n * \sum_i \{(y[i] - y_pred[i])^2\}$$

Jak zmierzyć błąd algorytmu? - funkcja kosztu $L(y, y_pred)$ c.d.

Nasze wyniki predykcji:

$$y_pred[0] = 800 + 150 = 950$$

$$y_pred[1] = 1600 + 150 = 1750$$

$$y_pred[2] = 800 + 150 = 950$$

Nasza funkcja kosztu $L(y, y_pred)$

MSE (mean square error):

$$L(y, y_pred) = 1/n * \sum_i [(y[i] - y_pred[i])^2]$$

Posiadane etykiety danych:

$$y[0] = 1500$$

$$y[1] = 3000$$

$$y[2] = 1900$$

Funkcja kosztu $L(y, y_pred)$

$$\begin{aligned} L(y, y_pred) &= \frac{1}{3} * [(1500 - 950)^2 + (3000 - 1750)^2 + (1900 - 950)^2] = \\ &= \frac{1}{3} * [302500 + 1562500 + 902500] = \\ &= \frac{1}{3} * 2767500 \sim \mathbf{922500} \end{aligned}$$

Funkcja kosztu $L(y, y_pred)$

Funkcja kosztu $L(y, y_pred) = 922500$ - to nie jest dobry wynik :)

Co jest naszym celem? - minimalizacja funkcji kosztu $L(y, y_pred)$ czyli **$\min L(y, y_pred)$**

Co jest naszym celem? - minimalizacja funkcji kosztu $L(y, y_pred)$ czyli **$\min L(y, y_pred)$**

Jak to zrobić? - zmieniając parametry $\theta = [a = 100, b = 150]$ przy pomocy metody gradientu prostego - ***Gradient Descent***

Metoda gradientu prostego (Kocioł pod Polskim Grzebieniem, Tatry Słowackie)



Autor: Franciszek Górski

Metoda gradientu prostego

Jest wiele metod optymalizujących funkcję, my skupimy się na jednej z najpopularniejszych z nich - metodzie gradientu prostego.

Wyobraźmy sobie dolinę wśród gór w której chcemy znaleźć najniżej położony punkt, który określimy jako minimum globalne.

W tym celu zastosujemy właśnie metodę gradientu prostego, która licząc pochodne cząstkowe parametrów optymalizowanej funkcji wskazuje kierunek **wzrostu** funkcji!

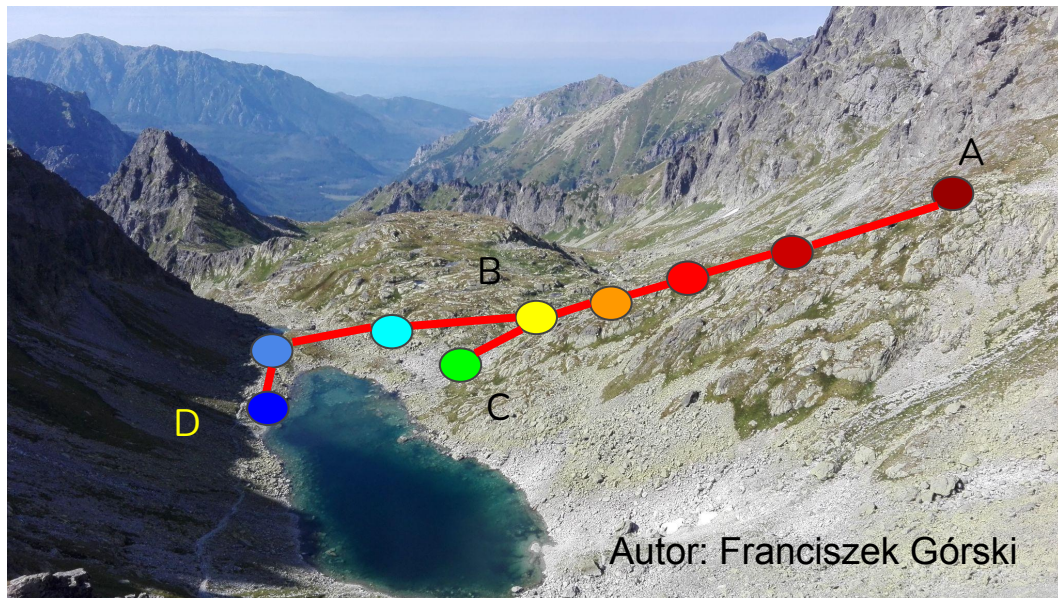
Metoda gradientu prostego

Wzrostu? Ale my chcemy minimalizować funkcję! - Dlatego będziemy wykorzystywali **zanegowaną** wartość gradientu - **negacja wzrostu == spadkowi**

Startujemy w punkcie A i korzystając z zanegowanej wartości gradientu kierujemy się w dół zbocza doliny.

Aż do punktu D, który jest najniżej położonym punktem w dolinie.

Uwaga na lokalne minima np. w punkcie C!



Autor: Franciszek Górski

Metoda gradientu prostego - obliczenia

$$L(y, y_pred) = 1/n * \sum_i \{(y[i] - y_pred[i])^2\}$$

$$\nabla L(a, b) = \{\partial a, \partial b\} =$$

$$h(x) = a*x_1 + b, h(x[i]) = y_pred[i]$$

$$\partial a = - 2/n * \sum_i \{x[i]*(y[i] - a*x[i] + b)\}$$

$$\theta = [a, b]$$

$$\partial b = - 2/n * \sum_i \{(y[i] - a*x[i] + b)\}$$

$$L(y[i], x[i], \theta) = 1/n * \sum_i \{(y[i] - a*x[i] - b)^2\}$$

$$\theta_0 = [a_0 = 100, b_0 = 150]$$

$$a_1 := a_0 - \eta * \partial a = a_0 - \eta * (- 2/n * \sum_i \{x[i]*(y[i] - a_0*x[i] + b_0)\}) = a_0 - \eta * (- 2/n * \sum_i \{x[i]*(y[i] - y_pred[i])\})$$

$$b_1 := b_0 - \eta * \partial b = b_0 - \eta * (- 2/n * \sum_i \{(y[i] - a_0*x[i] + b_0)\}) = b_0 - \eta * (- 2/n * \sum_i \{(y[i] - y_pred[i])\})$$

η (eta)- współczynnik uczenia (ang. *learning rate*)

Metoda gradientu prostego - obliczenia

$$\theta_0 = [a_0 = 100, b_0 = 150]$$

$$y = [1500, 3000, 1900]$$

$$y_{pred} = [950, 1750, 950]$$

$$\eta = 0.001$$

Nasze nowe parametry θ_1 :

$$\theta_1 = [a_1 = -78,66, b_1 = -175]$$

$$\begin{aligned} a_1 &:= 100 - 0.001 * \frac{2}{3} * [8 * (1500 - 950) + 16 * (3000 - 1750) + 8 * (1900 - 950)] = 100 - 0.001 * \frac{2}{3} * \\ &[8 * 550 + 16 * 1250 + 8 * 950] = 100 - 0.001 * \frac{2}{3} * (4400 + 20000 + 7600) = 100 - 0.001 * \frac{2}{3} * 32000 \\ &= 100 - \frac{2}{3} * 32 \approx 100 - 21,33 = -78,66 \end{aligned}$$

$$\begin{aligned} b_1 &:= 150 - 0.001 * \frac{2}{3} * [150 * (1500 - 950) + 150 * (3000 - 1750) + 150 * (1900 - 950)] = 150 - 0.001 * \\ &\frac{2}{3} * [150 * 550 + 150 * 1250 + 150 * 950] = 150 - 0.001 * \frac{2}{3} * (82500 + 187500 + 142500) = 150 - \\ &0.001 * \frac{2}{3} * 412500 = 150 - \frac{2}{3} * 412,5 \approx 150 - 275 = -175 \end{aligned}$$

Regresja liniowa jednej zmiennej

Pamięć RAM
8
16
8

= X

Omawiany przez nas przykład określany jest jako **regresja liniowa jednej zmiennej**.

Regresja oznacza, że **zbiorem wartości** funkcji są **liczby rzeczywiste**.

Cena
1500
3000
1900

= y

W ramach uproszczenia przedstawiliśmy regresję dla hipotezy z jedną zmienną niezależną x .

Regresja liniowa wielu zmiennych

Pamięć RAM	Pamięć VRAM GPU	Liczba rdzeni CPU	Taktowanie CPU
8	0	4	2.4
16	6	8	2.5
8	2	8	2.3

= X

Cena
1500
3000
1900

= y

W prawdziwych problemach spotkacie się jednak z danymi zawierającymi wiele zmiennych niezależnych x, czyli wieloma parametrami danych.

Regresja liniowa wielu zmiennych

$$h(x) = a*x4 + b*x3 + c*x2 + d*x1 + e$$

I wtedy taka hipoteza zostanie wykorzystana do treningu modelu, reszta kroków pozostaje niezmienna ...

Jednak w przypadku wielu zmiennych model liniowy może okazać się niewystarczający, ale o tym kiedy indziej ...

Klasyfikacja

- Zbiorem wartości zamiast liczb rzeczywistych są dyskretne (z góry określone) wartości np. liczby [1, 2, 3]
- Klasyfikacja polega na przypisania danych do konkretnych klas
- Tak samo jak w regresji tutaj też jest hipoteza, jej parametry, funkcja kosztu i optymalizacja parametrów np. metodą gradientu prostego
- Inne są jednak hipotezy i funkcje kosztu

Klasyfikacja - dane

Pamięć RAM	Pamięć VRAM GPU	Liczba rdzeni CPU	Taktowanie CPU
8	0	4	2.4
16	6	8	2.5
8	2	8	2.3

= X

Klasa jakości (1 -3)
1
3
2

= y

Jak widać teraz te same dane zostały przypisane do jednej z 3 klas określających poziom “możliwości” poszczególnych laptopów

Klasyfikacja - hipoteza

Jako hipotezy w klasyfikacji wykorzystuje się funkcje logistyczne takie jak sigmoid, tangens hiperboliczny czy ReLu.

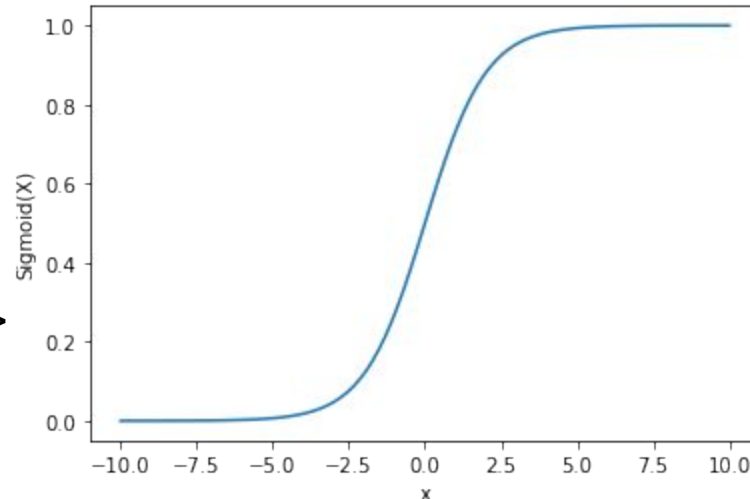
$$\text{sigmoid}(\mathbf{z}) = 1 / (1 + e^{-\mathbf{z}})$$

$$\mathbf{z} = \boldsymbol{\theta} * \mathbf{x}^T = \theta_0 * x_0 + \theta_1 * x_1$$

$$\boldsymbol{\theta} = [\theta_0, \theta_1]$$

$$\mathbf{x}^T = [x_0, x_1]^T$$

Zbiór wartości : $\langle 0, 1 \rangle$ albo $\langle -1, 1 \rangle$



Klasyfikacja - funkcja kosztu

Funkcja kosztu ma postać:

$$L(y, y_pred) = 1/n * \sum_i [y[i] * (-\log(y_pred[i])) + (1 - y[i]) * (-\log(1 - y_pred[i]))]$$

gdzie $y_pred[i] = \text{sigmoid}(z) = 1 / (1 + e^{-z})$

$y = 1, L(y, y_pred) = -\log(y_pred)$

$y = 0, L(y, y_pred) = -\log(1 - y_pred)$

Pytania?

Jeśli nie to ...

pora na krótką prezentację w Colabie