

# **Zadanie nr 1 – reguła Delta**

## **Inteligentna analiza danych**

Robert Radczyc, 203976

Dawid Michałowski, 203942

22.03.2017

# 1 Cel zadania

Napisanie programu, który implementuje pojedynczy neuron liniowy przeszkolony z regułą delta z zastosowaniem pojedynczych i wielu wzorców szkoleniowych.

## 2 Wstęp teoretyczny

Pojedynczy neuron składa się z określonej ilości wejść ( $x_1, x_2, \dots, x_N$ ) oraz odpowiednich wag ( $w_1, w_2, \dots, w_N$ ), a także wyjścia  $y = \sum_{i=0}^N x_i w_i$  [1].

Na wejście podajemy  $n$  wzorców treningowych, składających się z wejść i wyjścia  $z$ , na podstawie, których obliczamy przybliżone wartości wag.

Ich trening odbywa się w określonej ilości  $K$  epok. Różnicą między wartością obliczoną wyjścia  $y$ , a wartością oczekiwaną  $z$  w danej epoce jest  $\delta$ , która w każdej epoce ma wpływ na zmianę wartości wagi.

## 3 Eksperymenty i wyniki

### 3.1 Eksperyment nr 1

#### 3.1.1 Założenia

Przyjmujemy stałe wartości wejść i epok i losową wartość wag.

#### 3.1.2 Przebieg

Sprawdzamy dokładność metody dla trzech przypadków:

- a) liczba wzorców treningowych mniejsza niż liczba wag  $n < N$ ,
- b) liczba wzorców treningowych równa liczbie wag  $n = N$ ,
- c) liczba wzorców treningowych większa niż liczba wag  $n > N$ ,

#### 3.1.3 Rezultat

Rezultaty przedstawiono na Rys. 1 – 3.

```
Trening 1:
X: [-4.3, -6.3, 4.6, -9.6, 1.3]
Wynik oczekiwany (z): 44
Wynik neuronowy (y): 44.0
abs(z-y): 0.0

Trening 2:
X: [0.1, -5.3, -4.9, -5.2, 6.4]
Wynik oczekiwany (z): 44
Wynik neuronowy (y): 43.999999999999986
abs(z-y): 1.4210854715202004e-14

Trening 3:
X: [2.4, 3.0, 10.0, -9.4, 9.2]
Wynik oczekiwany (z): 44
Wynik neuronowy (y): 43.999999999999999
abs(z-y): 7.105427357601002e-15

Wagi przed: [0.5123463695454274, 0.9470413156781942, 0.4336038580188477, 0.20594031211338837, 0.9790467481356809]
Wagi po: [-0.3910743762023524, -1.8655978630634087, -0.24589584922400257, -2.926579870089736, 2.770052152768736]
```

Rys. 1 Zrzut ekranu dla przypadku a)  $n < N$  (1000 Epok)

```

Trening 1:
X: [0.9, -8.4, 2.7, -3.8, 3.9]
Wynik oczekiwany (z): 44
Wynik neuronowy (y): 43.989437147102926
abs(z-y): 0.010562852897074038

Trening 2:
X: [-8.6, -6.6, -1.7, 0.6, -9.6]
Wynik oczekiwany (z): 44
Wynik neuronowy (y): 44.02530313437657
abs(z-y): 0.02530313437657128

Trening 3:
X: [9.4, -8.8, -7.9, -6.4, 4.8]
Wynik oczekiwany (z): 44
Wynik neuronowy (y): 43.94425276772047
abs(z-y): 0.05574723227952916

Trening 4:
X: [5.8, -9.3, -9.6, -6.2, 6.0]
Wynik oczekiwany (z): 44
Wynik neuronowy (y): 44.072814360271124
abs(z-y): 0.0728143602711242

Trening 5:
X: [-7.5, -5.9, -7.5, -4.7, -5.1]
Wynik oczekiwany (z): 44
Wynik neuronowy (y): 43.96223830054299
abs(z-y): 0.03776169945700758

Wagi przed: [0.8589053549368629, 0.9371494272717286, 0.2678945169279965, 0.5489630455590125, 0.4121165357786586]
Wagi po: [-0.20729351393205983, -4.687861868339862, 0.47697389786646094, -2.3708250839724787, -1.409991622718835]

```

*Rys. 2 Zrzut ekranu dla przypadku b)  $n=N$  (1000 Epok)*

```

Trening 1:
X: [-9.3, 0.2, 7.5, 2.3, -5.4, -5.8, -2.4]
Wynik oczekiwany (z): 44
Wynik neuronowy (y): 43.999999999999999
abs(z-y): 7.105427357601002e-15

Trening 2:
X: [-7.1, 6.1, 7.9, 5.7, -6.7, 2.4, 7.9]
Wynik oczekiwany (z): 44
Wynik neuronowy (y): 44.000000000000001
abs(z-y): 7.105427357601002e-15

Trening 3:
X: [-2.6, 0.3, 1.5, -2.2, 3.8, 0.6, 6.7]
Wynik oczekiwany (z): 44
Wynik neuronowy (y): 43.999999999999997
abs(z-y): 2.842170943040401e-14

Wagi przed: [0.38267079592778264, 0.2719541140385985, 0.8725105519745623, 0.1555716136195533, 0.005249526211477562, 0.9910769800923893, 0.9301099334544038]
Wagi po: [-3.5953851963130994, -0.7322923513379841, 3.2053196104202493, -1.6722619206702707, 1.9324177997925334, -1.391492381991941, 2.9666325886366978]

```

*Rys. 3 Zrzut ekranu dla przypadku b)  $n>N$  (1000 Epok)*

Po przeprowadzeniu testów łatwo zauważyć, że dokładność metody jest najlepsza w przypadku gdy liczba wag jest większa niż liczba wzorców  $n < N$ . W miarę zwiększania ilości wzorców treningowych lub zmniejszania ilości wag dokładność ta maleje.

## 3.2 Eksperyment nr 2

### 3.1.1 Założenia

Przyjmujemy stałe wartości wejść i losową wartość wag. Ich ilości są sobie równe.

### 3.1.2 Przebieg

Sprawdzamy dokładność metody dla trzech przypadków:

- a) liczba epok równa 100,
- b) liczba epok równa 10 000,
- c) liczba epok równa 100 000.

### 3.1.3 Rezultat

Rezultaty przedstawiono na Rys. 4 – 6.

```
Trening 1:
X: [0.2, -2.4, 4.3, -8.0, -1.4]
Wynik oczekiwany (z): 44
Wynik neuronowy (y): 25.466507985754667
abs(z-y): 18.533492014245333

Trening 2:
X: [-8.1, -9.5, 0.9, -9.2, 2.2]
Wynik oczekiwany (z): 44
Wynik neuronowy (y): 34.59659585603389
abs(z-y): 9.403404143966107

Trening 3:
X: [8.2, -1.2, 6.3, -4.3, 5.9]
Wynik oczekiwany (z): 44
Wynik neuronowy (y): 54.121025999664994
abs(z-y): 10.121025999664994

Trening 4:
X: [-1.9, 1.2, 4.6, 0.5, -2.6]
Wynik oczekiwany (z): 44
Wynik neuronowy (y): 1.6402693762636629
abs(z-y): 42.35973062373634

Trening 5:
X: [8.3, 0.3, -6.4, 6.6, 4.3]
Wynik oczekiwany (z): 44
Wynik neuronowy (y): 13.075974855351774
abs(z-y): 30.924025144648226

Wagi przed: [0.8071832694758514, 0.42397023024221936, 0.6625470363823747, 0.16763426004730375, 0.014028480174843594]
Wagi po: [4.613693709837329, -9.25979023782811, 3.483149305470737, 1.8702631304966026, -1.668070671299171]
```

Rys. 4 Zrzut ekranu dla przypadku a)  $K=100$

```
Trening 1:
X: [8.8, 9.7, 9.3, -9.3, 8.1]
Wynik oczekiwany (z): 44
Wynik neuronowy (y): 43.99999999999998
abs(z-y): 1.9895196601282805e-13

Trening 2:
X: [-0.6, -7.5, 2.3, 6.7, 8.1]
Wynik oczekiwany (z): 44
Wynik neuronowy (y): 44.000000000000027
abs(z-y): 2.7000623958883807e-13

Trening 3:
X: [-2.2, 5.1, -4.4, 4.4, -1.2]
Wynik oczekiwany (z): 44
Wynik neuronowy (y): 43.999999999999968
abs(z-y): 3.197442310920451e-13

Trening 4:
X: [2.8, -6.3, 4.5, 0.6, -3.6]
Wynik oczekiwany (z): 44
Wynik neuronowy (y): 43.999999999999971
abs(z-y): 2.913225216616411e-13

Trening 5:
X: [0.4, -9.8, -3.1, 6.3, 4.9]
Wynik oczekiwany (z): 44
Wynik neuronowy (y): 43.999999999999961
abs(z-y): 3.907985046680551e-13

Wagi przed: [0.9527547136768828, 0.7756042478938279, 0.40402941730741526, 0.5856560170717833, 0.17405613089461558]
Wagi po: [16.325498446853864, 3.9602761318692057, 1.5239976621090234, 14.51527615645877, -2.130885779505293]
```

Rys. 5 Zrzut ekranu dla przypadku b)  $K=10\ 000$

```

Trening 1:
X: [-6.7, -5.5, -4.6, 3.4, 2.5]
Wynik oczekiwany (z): 44
Wynik neuronowy (y): 43.999999999996554
abs(z-y): 3.446132268436486e-12

Trening 2:
X: [-0.1, 3.3, -5.0, -1.1, -9.8]
Wynik oczekiwany (z): 44
Wynik neuronowy (y): 43.99999999999905
abs(z-y): 9.521272659185342e-13

Trening 3:
X: [7.8, 4.9, 0.8, -4.4, 5.0]
Wynik oczekiwany (z): 44
Wynik neuronowy (y): 43.999999999995005
abs(z-y): 4.995115432393504e-12

Trening 4:
X: [1.3, 3.3, 6.8, 9.1, -6.5]
Wynik oczekiwany (z): 44
Wynik neuronowy (y): 43.99999999999905
abs(z-y): 9.521272659185342e-13

Trening 5:
X: [5.0, 6.8, -3.0, -1.7, 7.7]
Wynik oczekiwany (z): 44
Wynik neuronowy (y): 44.00000000000233
abs(z-y): 2.3305801732931286e-12

Wagi przed: [0.1575330906804998, 0.9738633104953137, 0.23174942261477494, 0.3538300050705582, 0.5169033510092608]
Wagi po: [33.57493624465944, -19.263800427171482, -21.391629322755737, 21.03374360421375, -2.7660210596925916]

```

*Rys. 6 Zrzut ekranu dla przypadku c)  $K=100\ 000$*

Po przeprowadzeniu testów łatwo zauważyć, że dokładność rośnie wraz ze wzrostem liczby epok. Dla  $K = 100\ 000$  otrzymane wyniki są prawie identyczne oczekiwany wartości

## 4 Wnioski

Po przeprowadzeniu eksperymentów można z łatwością stwierdzić implementacja pojedynczego neuronu liniowego przeszkolonego z regułą delta jest najbardziej efektywna i zbliżona do pożądanego wyniku, gdy posiada liczbę wag większą niż liczba wzorców oraz wystarczająco dużą liczbę epok. Łatwo jest więc stwierdzić, że nim większa liczba epok oraz znacząco więcej wag niż wzorców tym wynik jest bardziej zbliżony do wyniku oczekiwanego.

## 5 Bibliografia

[1] Instrukcja do zadania z platformy WIKAMP - Laboratory 1 The Delta rule.