

NENR DZ7 Izvještaj

Krešimir Topolovec 0036485747

1.) Neuron TIP1 s jednim ulazom

Neuroni tipa 1 na izlazu y generiraju mjeru udaljenosti ulaza x_i u odnosu na vrijednost težina w_i

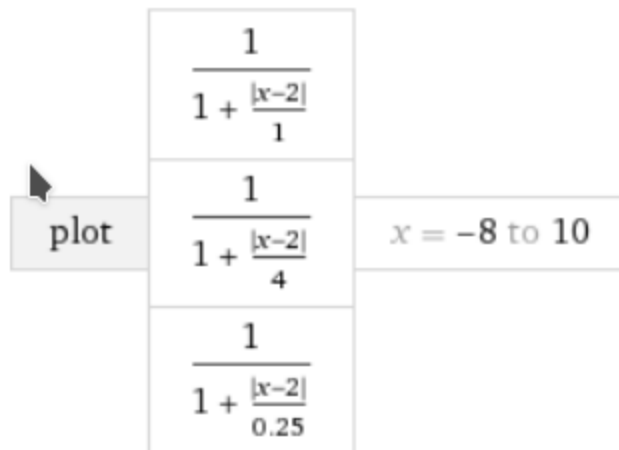
Prijenosna funkcija tog neurona glasi:

$$*TIP\ 1: y = \frac{1}{1 + \sum_{i=1}^n \frac{|x_i - w_i|}{s_i}}$$

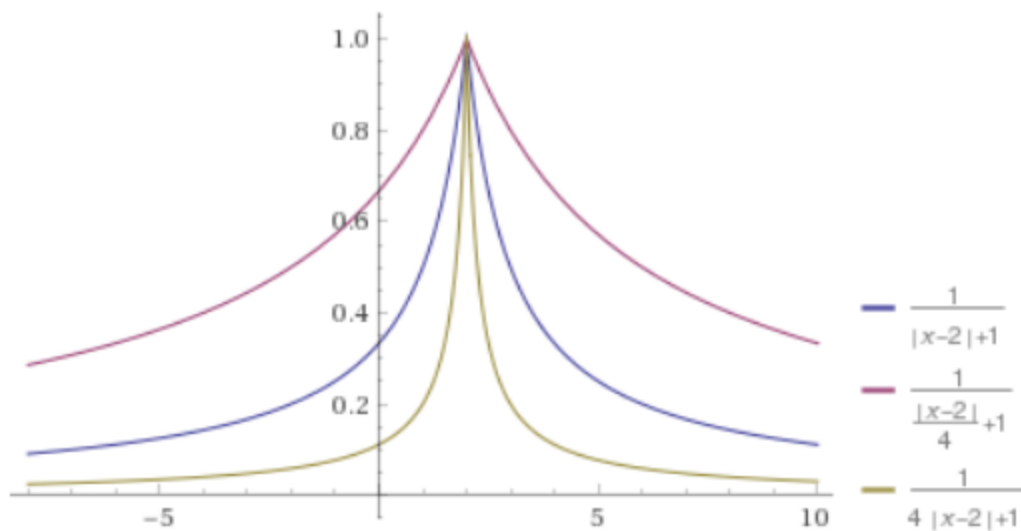
Prijenosna funkcija sigmoidalnog neurona korištenog kod učenja algoritmom backpropagation

$$TIP\ 2: y = \frac{1}{1 + e^{(net)}} \quad net = \sum_{i=0}^n w_i * x_i$$

Graf prijenosne funkcije neurona TIP1:



Plot:

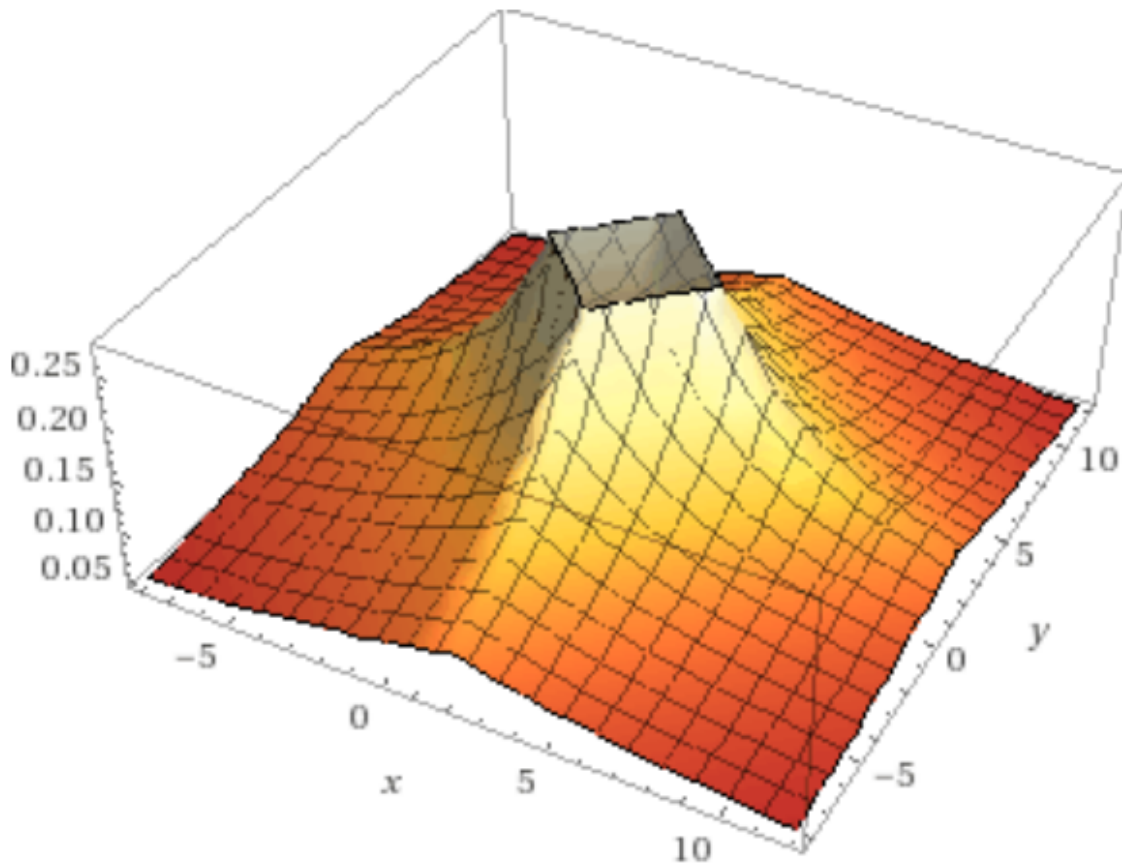


Interpretacija funkcije neurona tipa 1 i grafa:

- Parametar s_i utječe na to uolikoj mjeri će apsolutna udaljenost između x_i i w_i utjecati na izlaz y
- Kod velikog s_i udaljenost između x_i i w_i mora biti jako mala, odnosno ulaz x_i mora biti jako sličan ili jednak parametru w_i pohranjenom u neuronu da bi izlaz bio blizak broju 1.
- Izgled prijenosne funkcije neurona s 2 ulaza

- Parametrima s_1 i s_2 kontrolira se utjecaj udaljenosti ulaznog para/točke (x_1, x_2) i pohranjenog para/točke (w_1, w_2) na izlaz neurona

U primjeru sa slike: $y = \frac{1}{1 + \frac{|x-2|}{0.25} + \frac{|y-2|}{0.25}}$



Karakteristika neurona TIP1 sa 2 ulaza

2.) Podaci

Podatke vizualiziramo jednostavnom skriptom u ulazne podatke za učenje prosljeđujemo gnuplot alatu.

```
require 'gnuplot'

filepath = File.join(File.dirname(__FILE__), '../data/train-data.txt')

dataset = DataSet.read_from_file(filepath)
```

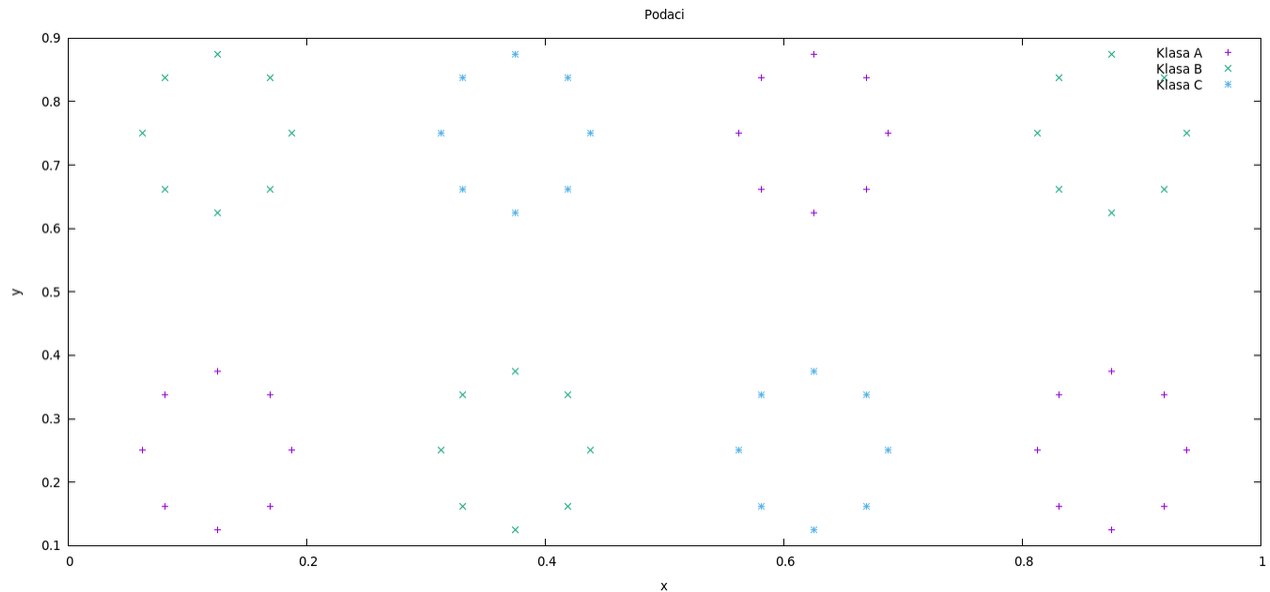
```

Gnuplot.open do |gp|
  Gnuplot::Plot.new(gp) do |plot|
    plot.xrange "[0:1]"
    plot.title "Podaci"
    plot.xlabel "x"
    plot.ylabel "y"

    [:a, :b, :c].each do |sample_class|
      group = dataset.data.filter{ |sample| sample[:sample_class] == sample_class }
      plot.data << Gnuplot::DataSet.new(
        [group.map{ |sample| sample[:x] },
        group.map{ |sample| sample[:y] }
      ]) do |ds|
        ds.with = "points"
        ds.title = "Klasa #{sample_class.to_s.capitalize}"
      end
    end
  end
end
end
end

```

Na grafu je vidljivo kako su podaci grupirani u 3 klase te svaka klasa u pod grupe koje nisu linearno odvojive. Ne možemo pravcem odvojiti niti 2 od navedene 3 klase.



3.) Ručno određivanje parametara težina w_i neurona skrivenog sloja

Pri određivanju parametara w_1 i w_2 , ako razumijemo neuron tipa 1 sa dva ulaza intuitivno je jasno da će to biti x i y koordinate mediana svake podgrupe. Za određivanje parametara s_1 i s_2 gledamo razmake između svake podgrupe po osima x i y . Npr. Razmak između podgrupe oko točke $(0.125, 0.25)$ i podgrupe oko točke $(0.375, 0.75)$ po x osi je 0.1 te po y osi 0.25 . Slično i za sve ostale grupe

Vrijednosti parametara neurona:

```
#--w1----w2----s1---s2--
```

```
[0.125, 0.25, 0.1, 0.25], # A # 1. neuron skrivenog sloja
```

```
[0.125, 0.75, 0.1, 0.25], # B
```

```
[0.375, 0.25, 0.1, 0.25], # B
```

```
[0.375, 0.75, 0.1, 0.25], # C
```

```
[0.625, 0.25, 0.1, 0.25], # C
```

```
[0.625, 0.75, 0.1, 0.25], # A
```

```
[0.875, 0.25, 0.1, 0.25], # A
```

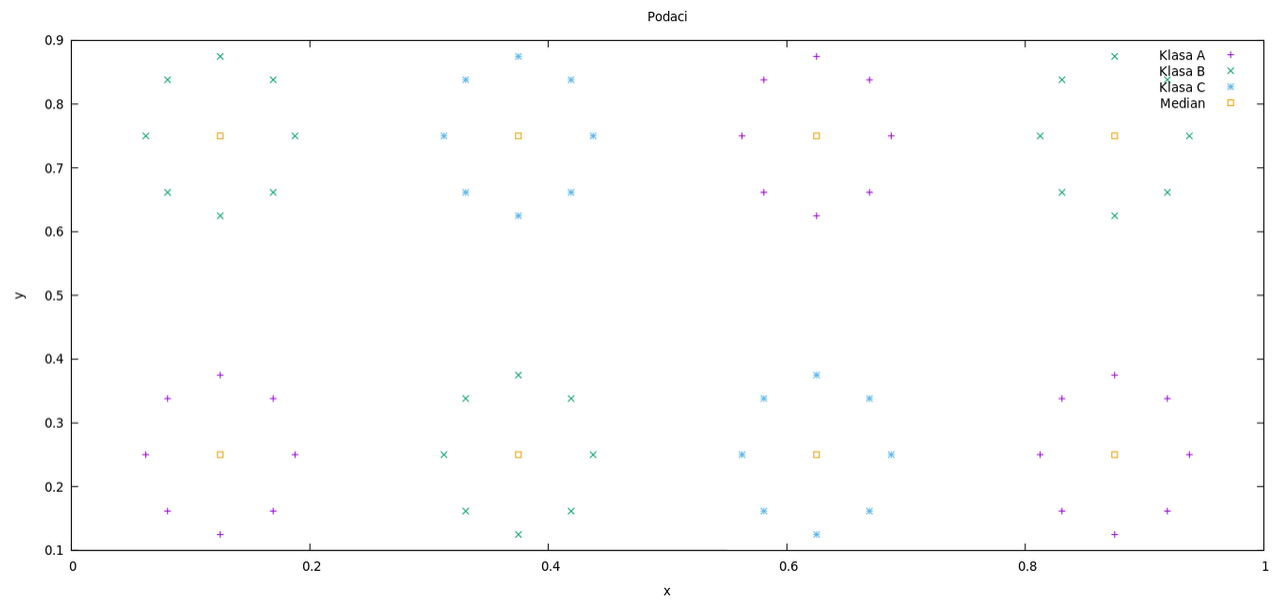
```
[0.875, 0.75, 0.1, 0.25], # B # zadnji neuron skrivenog sloja
```

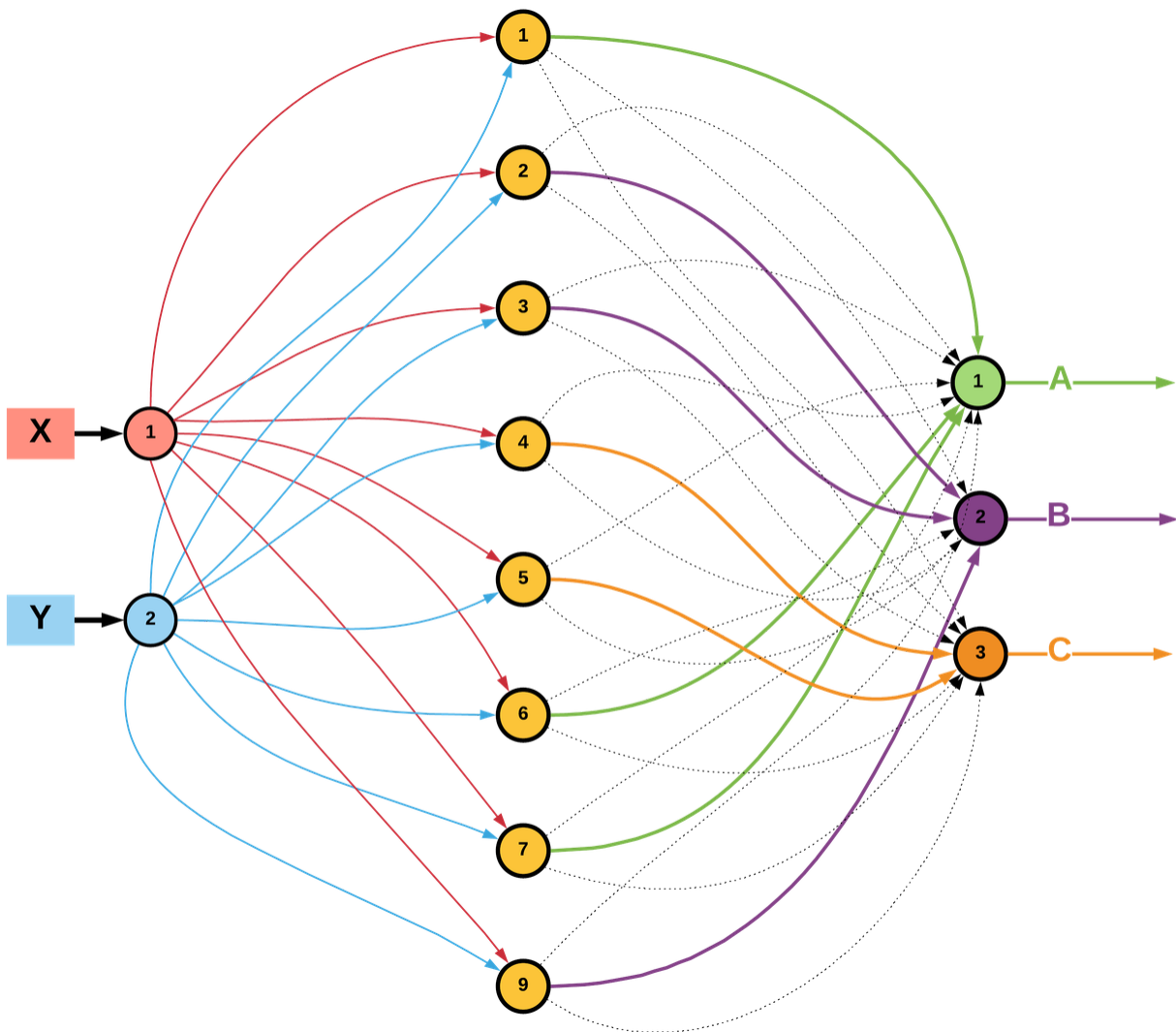
#-w1---w2----w3----w4----w5----w6---w7---w8

[1.0, -1.0, -1.0, -1.0, -1.0, 1.0, 1.0, -1.0], # A

[-1.0, 1.0, 1.0, -1.0, -1.0, -1.0, -1.0, 1.0], # B

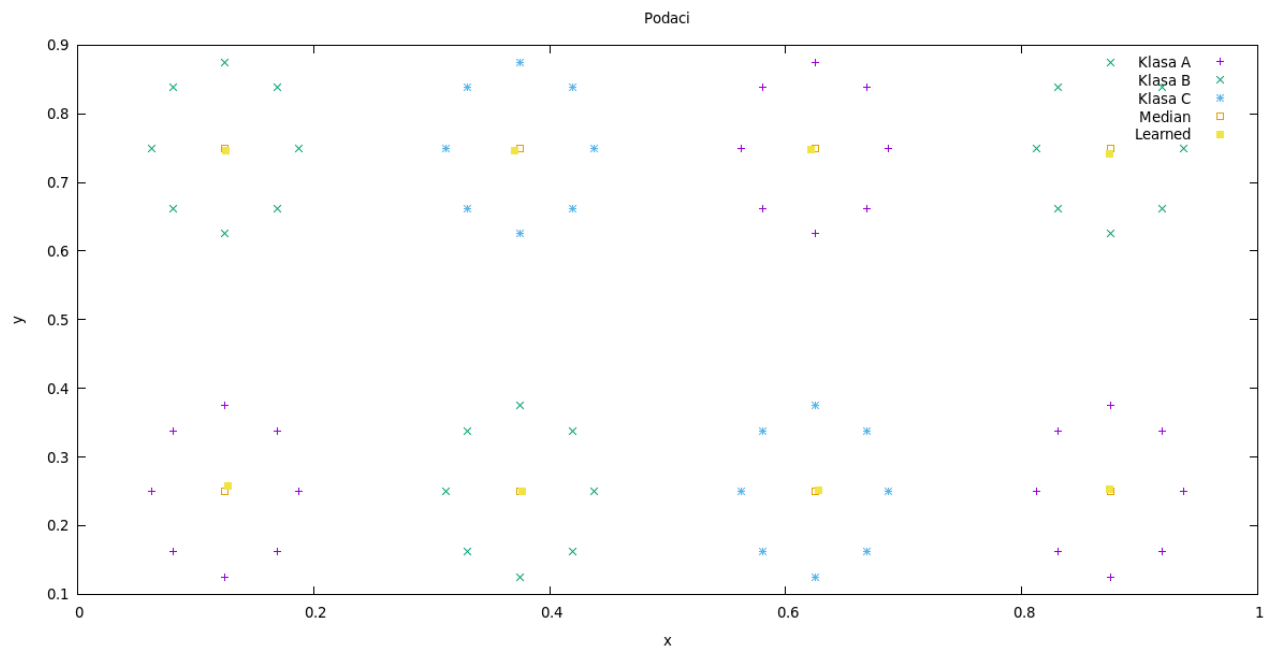
[-1.0, -1.0, -1.0, 1.0, 1.0, -1.0, -1.0, -1.0] # C



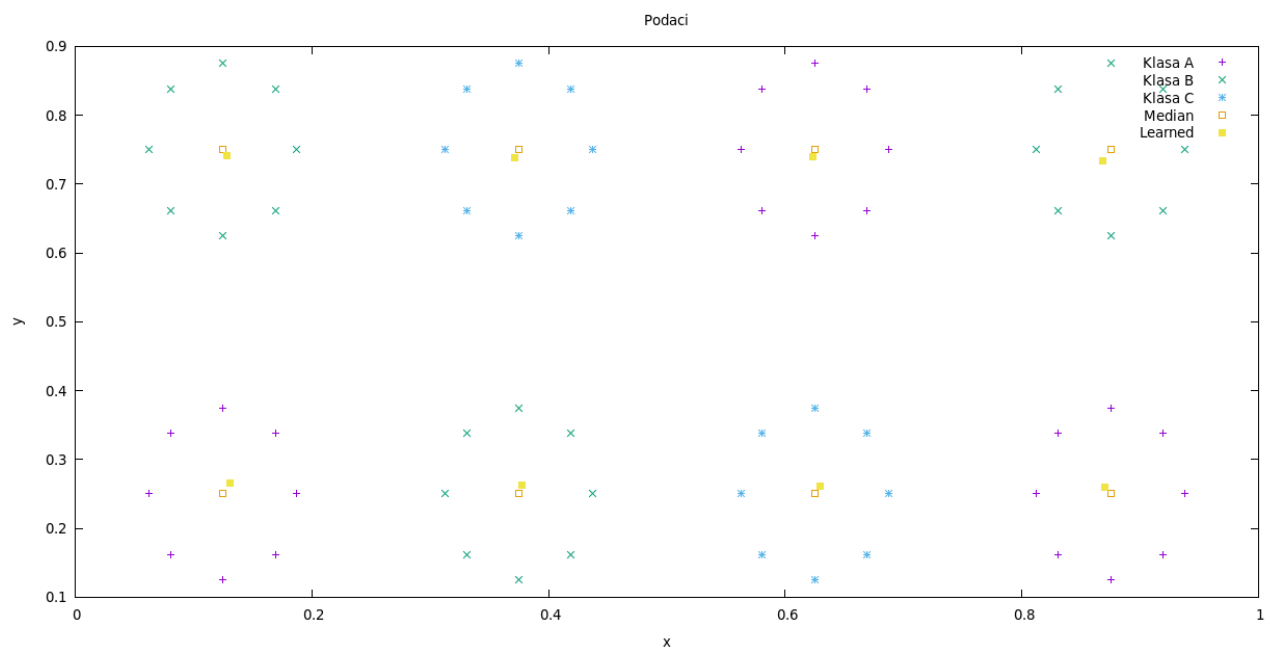


4.) Učenje mreže 2x8x3

Nakon 200k iteracija, greška je 0.0006



Nakon ~550k iteracija, $e < 1 * 10^{-7}$



Primjer treniranja mreže 2x8x3

Iteracija 539083; fitness: 9.98e-08

Parametri pri učenju:

- Veličina populacije: 50 jedinki
- k-turnirska selekcija: $k = 5$
- Mutacija (definirana u zadatku: gauss-add + gauss-replace) s parametrima

M1: prob1: 0.04, dev1: 0.15, t1: 1

M2: prob2: 0.04, dev2: 0.35, t2: 1

M3: prob3: 0.01, dev3: 0.8, t3: 1

Naučeni parametri w_1, w_2 te s_1, s_2

```
//w1          w2
[0.13097906082275665, 0.266266836074276] // A neuron 1
[0.12782407416328434, 0.7411576091786676] // B
[0.37758105610934856, 0.26269457559122433] // B
[0.3718463198904148, 0.7388380416781534] // C
[0.6296473637793707, 0.26121674026931574] // C
[0.6237505017641237, 0.7395735818485183] // A
[0.8705504659125491, 0.25935166268711346] // A
[0.8679498204047134, 0.7343524747531631] // B neuron 8
```

```
//s1          s2
[0.11749185338950978, 0.24654126593086984] // neuron 1
[-0.09593444441676535, -0.1949629027291016]
[0.08916208269967182, 0.22156677590231566]
[-0.0885566655937955, 0.22836384781486418]
[0.10361380894192361, 0.2487590593174898]
[0.1011605299828382, 0.2138764876290365]
[0.10374559280956698, -0.20199983223375123]
[-0.13610850985720582, -0.24484103523516795] // neuron 8
```

Već nakon par tisuća iteracija inicijalni parametri w_1 i w_2 počinju biti bliski ručno određenim parametrima (medijanima pojedinih grupa).

Naučeni parametri (w_1, w_2) neurona sloja 1 tj. skrivenog sloja nalaze se oko mediana pripadne podgrupe što je očekivano ponašanje mreže ovakve arhitekture. Svaki od 8 neurona skrivenog sloja mreže ovakve arhitekture klasificira svoju podgrupu. Probamo li za test postaviti broj neurona u skrivenom sloju na npr. 4, mreža neće postići ni približno dobre rezultate kao sa 8 neurona u skrivenom sloju.

```
Iteracija 100000; fitness: 0.40042212053270465 # Arhitektura  
2x4x3
```

Parametri (s_1, s_2) su različiti za x i y komponentu, ali vrlo slični za sve neurone i vrlo bliski ručno podešenim parametrima u zadatku 3.

Napomena: predznak parametara s_i nije bitan jer se u izračunu izlaza uzima [apsolutna vrijednost](#).

5.) Učenje mreže arhitekture 2x8x4x3

Učenje mreže ovakve arhitekture je komputacijski zahtjevnije zbog većeg broja neurona:

- Mnogo više računanja nego kod arhitekture 2x8x3
 - množenje, djeljenje, sumiranje
- Izlazi za svak neuron se računaju slijedno, za svaki sloj također slijedno
- Jedinke s kojima barata genetski algoritam su također mnogo duže
 - Sve operacije mutacije i križanja kod kojih se iterira po članovima jedinke traju duže

Svaka iteracija učenja zbog gornjih razloga vremenski traje duže te je memorijsko zauzeće veće

Međutim, sposobnost klasifikacije ovakve mreže je veća, tj. mreža lakše uči rješavati kompleksnije probleme, što je vidljivo i u testnom primjeru:

Primjer treniranja mreže 2x8x4x3

```
Pronađeno zadovoljavajuće rješenje u 174520 iteracija !: fitne  
s: 9.63e-08
```

Parametri:

- Veličina populacije: 50 jedinki

- k-turnirska selekcija: $k = 5$
- Mutacija (definirana u zadatku: *gauss-add + gauss-replace*) s parametrima:

M1: prob1: 0.04, dev1: 0.15, t1: 1

M2: prob2: 0.04, dev2: 0.35, t2: 1

M3: prob3: 0.01, dev3: 0.7, t3: 1

Dakle, sa istim parametrima te uz jednaku veličinu populacije u gotovo trostruko manje iteracija pronađeno je zadovoljavajuće rješenje. 😊

Zaključak: Dodatni skriveni sloj može korigirati rezultate prethodnog sloja (sloja koji sadrži 8 neurona TIP 1). Neuroni tipa 1 ni ne moraju jako dobro naučiti središta podgrupa klasa da bi krajnji rezultat bio dobar.

6.) Arhitektura 2x6x4x3

Moguće je dobiti ispravnu klasifikaciju svih uzoraka u arhitekturi koja ima broj skrivenih neurona TIP 1 u prvom sloju manji od 8 ($N_1 < 8$) iz razloga navedenog u zaključku prethodnog zadatka.

Primjer treniranja mreže 2x6x4x3

- Svi ostali parametri jednaki kao u zadatku 4 i 5

Pronađeno zadovoljavajuće rješenje u 473882 iteracija !: fitne ss: 9.96e-08

Vrijednosti parametara neurona sloja 1:

```
[#<Neural::Type1Neuron:0x00005592c843e9b8 @biases=[-0.1089841
1798496764, -0.17834987766921168], @num_of_inputs=2, @weights=
[0.13739391313494273, 0.2491119907973972]>,
  #<Neural::Type1Neuron:0x00005592c843e918 @biases=[0.06425775
567186717, 0.49593374732370915], @num_of_inputs=2, @weights=[
0.5951899759591625, 0.7462379522014531]>,
  #<Neural::Type1Neuron:0x00005592c843e878 @biases=[-0.1370994
798321767, 0.08269225821067644], @num_of_inputs=2, @weights=[
0.8748661188138257, 0.23629525031169307]>,
```

```
#<Neural::Type1Neuron:0x00005592c843e7d8 @biases=[-0.21484081908572444, -0.4771411504329601], @num_of_inputs=2, @weights=[0.5461858119421933, 0.25472474471119905]>,
```

```
#<Neural::Type1Neuron:0x00005592c843e738 @biases=[-0.18531282083795625, 0.5872177573842784], @num_of_inputs=2, @weights=[0.11772237343583104, 0.2920383344836824]>,
```

```
#<Neural::Type1Neuron:0x00005592c843e698 @biases=[0.15073374040893267, 2.0313242040360753], @num_of_inputs=2, @weights=[0.3378359597715847, 0.7039550870403213]>],
```