

# SA specifikacija - Federativno učenje

Ivan Mikić RA44/2020  
Luka Rakin RA205/2020

## Specifikacija algoritma koji će biti implementiran

Feed forward neuronska mreža sa N slojeva.

Prvih  $N - 1$  slojeva ima Relu aktivacionu funkciju, a poslednji sloj Softmax aktivacionu funkciju da bi pretvorio izlaz u vektor verovatnoća. Kao optimizator bi koristili cross entropy loss.

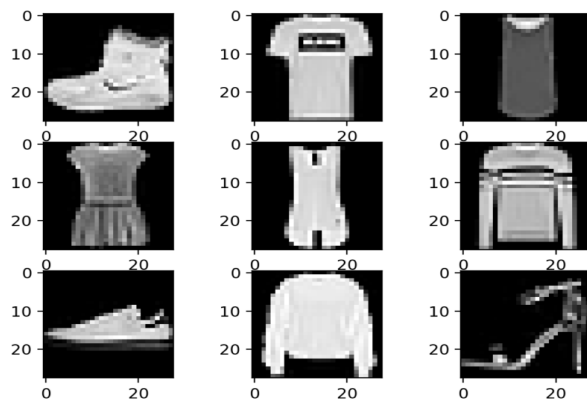
## Skup podataka

Fashion MNIST dataset

<https://github.com/zalandoresearch/fashion-mnist>

Skup podataka koji ima 60000 slika odeće veličine 28 x 28 sa 10 klasa:

Label	Description
0	T-shirt/top
1	Trouser
2	Pullover
3	Dress
4	Coat
5	Sandal
6	Shirt
7	Sneaker
8	Bag
9	Ankle boot



## Način distribucije treniranja algoritma

Podela slika dataseta na slučajan način na N aktora. Ideja je da skup podelimo na jednake veličine podskupova, ali bez stratifikacije podataka.

## Metod evaluacije rezultata

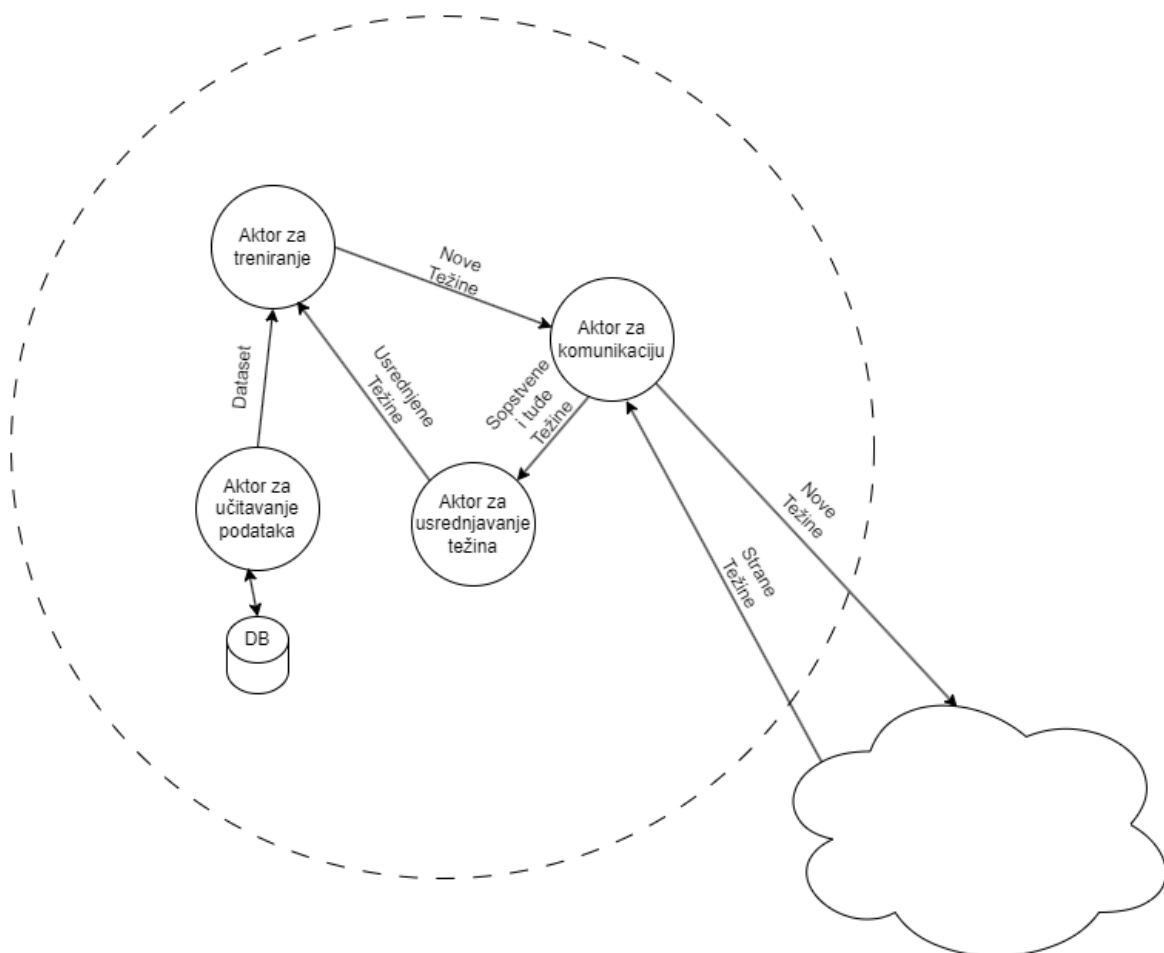
Performanse modela bi bile evaluirane putem accuracy metrike i matrice konfuzije. Takođe bi rezultati i proces treniranja mogli biti upoređeni naspram mreže trenirane na jednoj mašini.

## Aktori

Aktori će biti raspoređeni u klastere koji će međusobno komunicirati i vršiti treniranje.



Skica funkcionisanja člana klastera:



Vrste aktora koji se nalaze unutar klastera:

- Aktori za trening  
Prima težine i podatke za treniranje  
Šalje rezultat na svakih nekoliko epoha akтору za komunikaciju
- Aktori za komunikaciju  
Prima i koordiniše težine od ostalih klastera i od aktora za treniranje
- Aktori za učitavanje podataka  
Vrše komunikaciju sa nekom vrstom baze podataka  
Pripremaju podatke za treniranje  
Šalju podatke akтору za treniranje
- Aktori za usrednjavanje težina  
Primaju težine i usrednjavaju ih  
Šalju usrednjene težine na treniranje

## Poruke

- Težine
- Podaci za treniranje

## Detalji implementacije

Implementacija uz pomoć `proto.actor` biblioteke u jeziku Golang.  
Neuronska mreža će takođe biti implementirana u Golangu.