

# Agentske tehnologije – predlog projekta

## Clanovi:

Veljko Vukotic RA70 2020

Nikola Sehovac RA115 2020

## Problem koji se resava:

Ideja projekta je da se napravi model koji ce vrsiti klasifikaciju rucno pisanih brojeva na osnovu fotografika uz pomoc federativnog ucenja.

## Algoritam:

Koristićemo konvolutivnu neuronsku mrežu (CNN) za prepoznavanje ručno pisanih brojeva. CNN je prikladan za obradu slika i ima dobre performanse u zadacima prepoznavanja oblika.

## Arhitektura CNN-a:

1. Ulazni sloj: Slike brojeva dimenzija 28x28 piksela.
2. Konvolutivni slojevi: Očekujemo da će model imati 2 do 3 konvolutivna sloja, svaki praćen slojem sažimanja (pooling layer).
3. Potpuno povezani slojevi: Nakon konvolutivnih slojeva, sledi nekoliko potpuno povezanih slojeva.
4. Izlazni sloj: Softmax sloj za klasifikaciju 10 klasa (brojevi od 0 do 9).

## Skup podataka

Koristićemo skup podataka "Handwritten Digits Dataset" koji se sastoji od ručno pisanih brojeva. Skup sadrži 2061 slika u formatu PNG, prikupljenih iz različitih izvora. Slike su grayscale i dimenzija 28x28 piksela.

## Način distribucije treniranja algoritma

Federativno učenje će se koristiti za distribuiranje treniranja modela na lokalnim uređajima. Sistem će biti podeljen na više lokalnih modela, svaki će se trenirati na uređajima korisnika. Ažuriranja lokalnih modela će se slati na centralnu lokaciju radi kombinovanja i

formiranja globalnog modela, omogućavajući tako distribuirano učenje bez deljenja privatnih podataka.

### **Metod evaluacije rezultata**

Metod evaluacije rezultata biće tačnost (accuracy) modela u klasifikaciji brojeva na testnom skupu podataka. Kako bismo ocenili performanse modela, koristićemo test skup podataka koji nije korišćen tokom treniranja modela. Ocena tačnosti će nam pružiti uvid u sposobnost modela da ispravno prepozna brojeve.

### **Aktori**

#### **Različite vrste aktora i njihove uloge:**

##### **Centralni Aktor (CentralActor):**

- Koordinira proces distribuiranog učenja.
- Prikuplja ažuriranja od lokalnih modela.
- Kombinuje ažuriranja kako bi formirao globalni model.
- Šalje ažuriranja nazad lokalnim modelima.

##### **Lokalni Aktori (LocalActor):**

- Predstavljaju lokalne modele na uređajima korisnika.
- Treniraju se na lokalnim podacima.
- Šalju ažuriranja svog modela centralnom akтору.

#### **Poruke i njihova namena:**

##### **Ažuriranje modela (ModelUpdateMessage):**

- Lokalni aktori šalju ova ažuriranja centralnom akтору nakon svakog ciklusa treniranja.
- Centralni aktor koristi ova ažuriranja za formiranje globalnog modela.

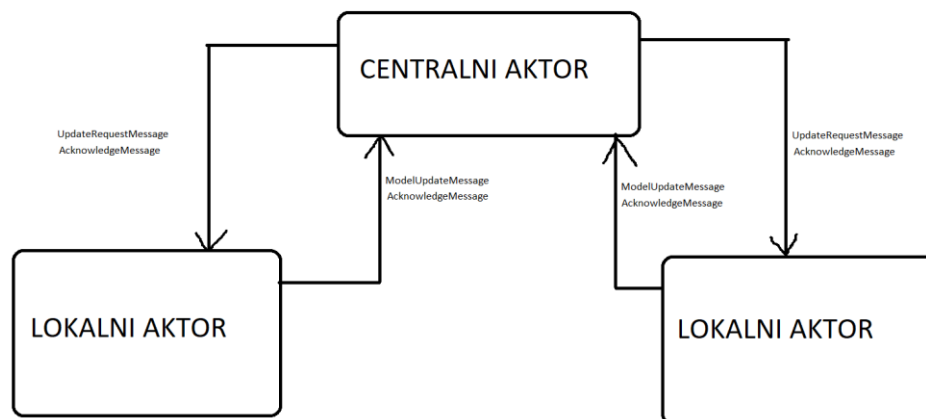
##### **Zahtev za ažuriranjem (UpdateRequestMessage):**

- Centralni aktor šalje zahteve lokalnim aktorima za slanje ažuriranja svojih modela.

##### **Potvrda o primanju (AcknowledgeMessage):**

- Služi za potvrdu o primanju poruke.

## SKICA:



## Detalji implementacije:

### Programski jezik:

- Koristićemo Go jezik za implementaciju celog projekta. Go jezik je efikasan, brz i pogodan za razvoj distribuiranih sistema, što ga čini idealnim izborom za ovakav projekat.

### Biblioteke:

- Za implementaciju aktorske modela, koristićemo biblioteku kao što je "[github.com/AsynkronIT/protoactor-go](https://github.com/AsynkronIT/protoactor-go)", koja omogućava lak razvoj distribuiranih sistema zasnovanih na aktorima u Go jeziku.

### Federativno učenje:

- Implementacija federativnog učenja će zahtevati integraciju algoritama za treniranje CNN modela u Go jeziku, kao i mehanizama za distribuirano treniranje i kombinovanje ažuriranja modela.

### Komunikacija:

- Komunikacija između aktora će se odvijati putem poruka, a za ovu svrhu koristićemo mehanizme koji su dostupni u biblioteci za aktorski model koju smo odabrali.