



---

# DUBOKO UČENJE

---

Neuralne mreže – prvi projektni zadatak



JANUARY 24, 2024

Kristina Kragović 2021/0270

Aleksandar Ilić 2021/0495

## Sadržaj

Postavka problema.....	2
Opis dataseta.....	3
Podela podataka .....	4
Pretprocesiranje podataka.....	5
Arhitektura modela.....	6
Preobučavanje neuralne mreže.....	10
Performanse neuralne mreže .....	11
Konfuzionna matrica .....	12
Primeri dobro i loše klasifikovanih primeraka klasa .....	14

## Postavka problema

- Ovaj skup podataka sadrži slike gestova ruku iz igre Kamen – Papir – Makaze. Dataset je podeljen u tri podfoldera koji se nalaze u folderu PROJEKAT.
- Cilj našeg projekta je kreiranje neuralne mreže koja treba da izvrši klasifikaciju zadatog dataseta. Naš dataset se sastoji iz tri klase (papir, kamen, makaze) od kojih svaka sadrži približno 750 primeraka. Slike su dimenzija 300x200 piksela koje će kasnije biti skalirane radi bržeg obučavanja neuralne mreže.



*Primeri klase*

## Opis dataseta

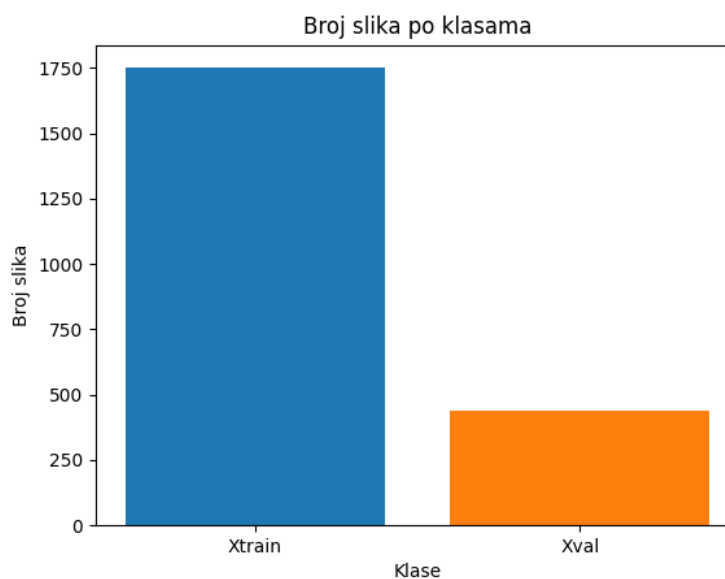
- Svaka klasa u našem datasetu sadrži približan broj odbiraka tako da nije potrebno koristiti tehnike balansiranja klasa poput: oversampling, undersampling ili class weights.
- Takođe nijedna klasa ne sadrži manje od 5% ukupnog broja odbiraka tako da nijedna nije zanemarena u klasifikaciji.



*Prikaz broja odbiraka po klasama*

## Podela podataka

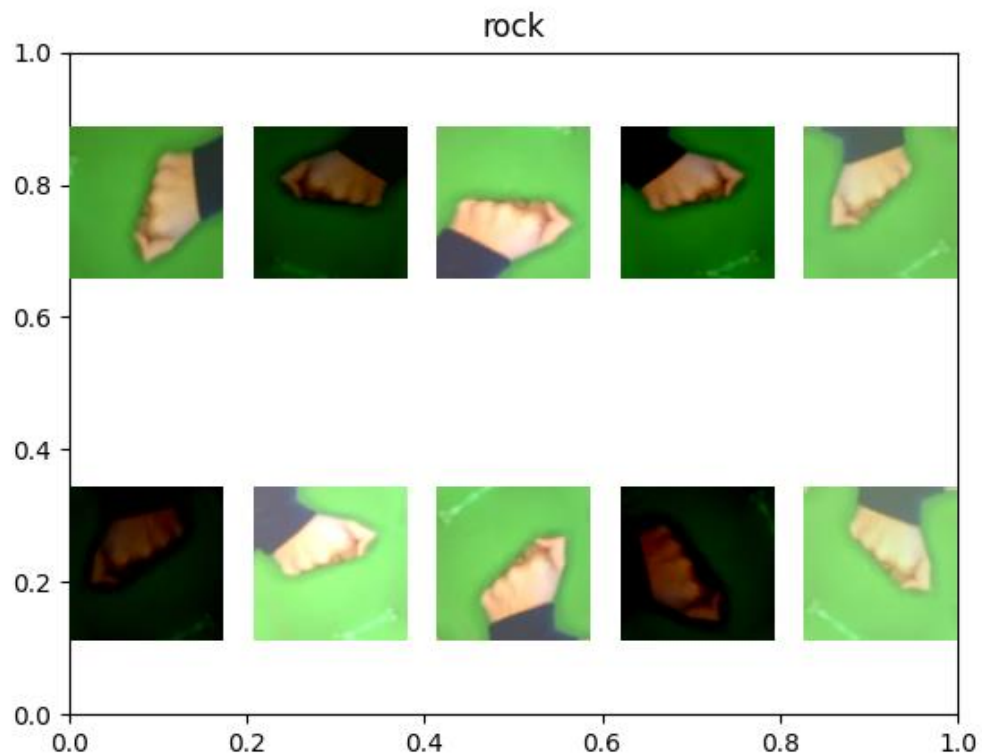
- Podatke delimo na dva skupa, skup za treniranje (Xtrain) i validacioni skup (Xval).
- Podela je izvršena korišćenjem funkcije `image_dataset_from_directory` iz biblioteke `keras.utils`.
- Skup za treniranje sadrži 80% ukupnog broja primeraka, a validacioni preostalih 20%.
- Podela je važna kako se uspešnost neuralne mreže ne bi procenjivala na osnovu podataka koji su korišćeni prilikom treniranja, već na podacima koji se ne nalaze u trening skupu da bismo dobili realniju sliku efikasnosti naše mreže. Takođe, važno je korišćenje validacionog skupa kako bismo prilikom treniranja mreže mogli da detektujemo preobučavanje mreže ukoliko do njega dolazi.



*Prikaz broja podataka u trening i validacionom skupu*

## Pretprocesiranje podataka

- Data augmentation predstavlja tehniku koju koristimo kada imamo ograničen skup podataka kako bismo proširili sam skup primenom raznih modifikacija. Sama tehnika bi trebala da pomogne neuralnoj mreži da bolje izvrši uopštavanje. Modifikacije koje su primenjene u projektu su: Rotacija, zumiranje, povećanje osvetljenosti slike kao i horizontalno i vertikalno obrtanje.



*Transformacije nad jednom slikom*

## Arhitektura modela

- Konvolucione mreže su pogodne za klasifikaciju podataka koji su oblika mreže tj. slike. Mreža sadrži konvolucione slojeve koji se koriste za ekstrakciju obeležja. Imamo više konvolucionih slojeva u modelu kako bismo postigli različit nivo apstrakcije podataka. Na primer: prvi sloj može detektovati jednostavna obeležja poput ivica, lukova, krugova i slično, drugi sloj može da prepozna oblike, a treći složene objekte (strukture) i tako dalje.
- Pored konvolucionih slojeva imamo i pooling slojeve koji smanjuju prostorne dimenzije slika i broj parametara. Pooling tehnika se koristi za feature selection (izbor obeležja) dok se konvolucionni slojevi koriste za feature extraction (izdvajanje obeležja).
- Naša neuralna mreža se sastoji iz tri konvoluciona sloja i pooling sloja. Nakon toga dodajemo i dropout sloj koji služi da predupredi preobučavanje. Dropout sloj podrazumeva da će se prilikom obučavanja isključiti pojedini neuroni kako se mreža ne bi previše oslonila na određena obeležja. Na kraju dodajemo i full connected (FC) odnosno potpuno povezane slojeve gde prvi ima 128 neruona sa aktivacionom funkcijom ReLU, a drugi 3 sa aktivacionom funkcijom softmax koja pretvara skup rezultata neuralne mreže u verovatnoće na osnovu kojih se donosi odluka.
- Poptuno povezani slojevi služe za donošenje završnih odluka zasnovanih na prethodnim slojevima (konvolucionim, pooling). U našem projektu u izlaznom sloju imamo tri neurona gde svaki neuron predstavlja određenu klasu – aktivacija određenog neurona znači pripadnost klasi.

*Kreiranje naše konvolucione mreže*

```

model = Sequential([
    data_augmentation,
    layers.Rescaling(1. / 255, input_shape=(80, 80, 3)),
    layers.Conv2D(filters=16, kernel_size=5, padding='same', activation='relu'),
    layers.MaxPooling2D(),
    layers.Conv2D(filters=32, kernel_size=5, padding='same', activation='relu'),
    layers.MaxPooling2D(),
    layers.Conv2D(filters=64, kernel_size=5, padding='same', activation='relu'),
    layers.Dropout(0.2),
    layers.Flatten(),
    layers.Dense(units=128, activation='relu'),
    layers.Dense(num_classes, activation='softmax')
])

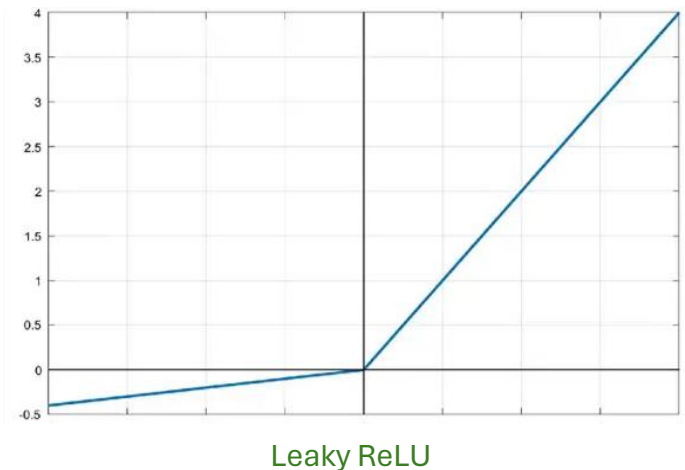
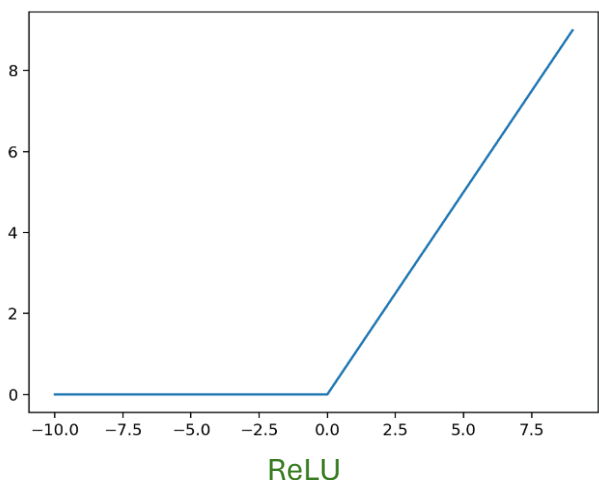
```

Ukupno imamo:

- 1 sloj augmentacije podataka
- 1 sloj pretprocesiranja
- 3 konvolutivna sloja
- 2 max pooling sloja
- 1 sloj dropouta
- 1 sloj flatten
- 2 potpuno povezana sloja



- ReLU je nelinearna aktivaciona funkcija i jedna je od najkorišćenijih funkcija u dubokom učenju. ReLU aktivaciona funkcija se definiše kao:  $f(x) = \max(0, x)$  odakle se može zaključiti da daje vrednost 0 za sve ulaze koji su 0 ili manji od nula dok za pozitivne vrednosti vraća istu vrednost – brza i efikasna implementacija.
- ReLU aktivaciona funkcija takođe daje vrlo brzu konvergenciju mreže tokom treninga i rešava problem „nestajućeg“ gradijenta koje imaju neke druge nelinearne funkcije poput sigmoida.
- Napomena: Treba imati u vidu da i ReLU aktivaciona funkcija ima svoj nedostatak poznat kao „dying ReLU“. „Dying ReLU“ podrazumeva da za različite ulaze dobijemo iste izlaze. Do problema dolazi ako je *bias* suviše negativan ili ako je inicijalno postavljena velika konstanta obučavanja. Rešenje za ovaj problem bi mogla biti funkcija Leaky ReLU koja je mala modifikacija funkcije ReLU.

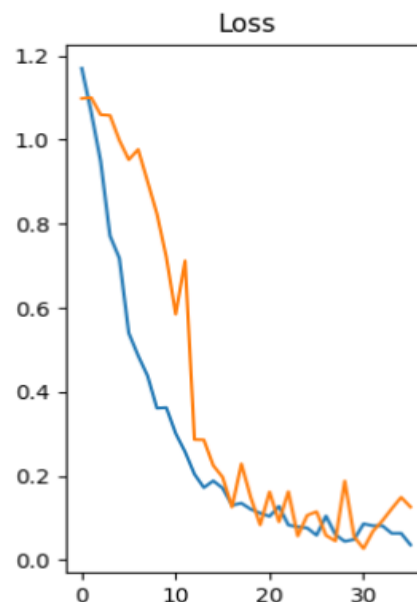


- Kriterijumska funkcija je jedan od bitnijih parametara zato što direktno utiče na računanje promene pojačanja, a samim tim i na brzinu konvergencije ka optimalnom rešenju, kao i uspešnost rada neuralne mreže. Za problem klasifikacije najčešće se koristi crossentropy kriterijumska funkcija. U projektu je korišćena funkcija SparseCategoricalCrossentropy koja je pogodna da se koristi kada imamo indekse (cele brojeve) koji predstavljaju klasu kojoj pripada neki podatak.
- Optimizator predstavlja metodu traženja minimuma kriterijumske funkcije tj. optimizaciju traženja pojačanja. Jedna od prvih metoda koja se koristila jeste *gradient descent* metoda optimizacije koja ima dosta mana. Da bi otklonili te mane koristimo u projektu Adamov optimizator. Adamov optimizator je algoritam koji kombinuje prednosti drugih popularnih algoritama optimizacije kao što su AdaGrad i RMSProp.

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) \left[ \frac{\delta L}{\delta w_t} \right] \quad v_t = \beta_2 v_{t-1} + (1 - \beta_2) \left[ \frac{\delta L}{\delta w_t} \right]^2$$

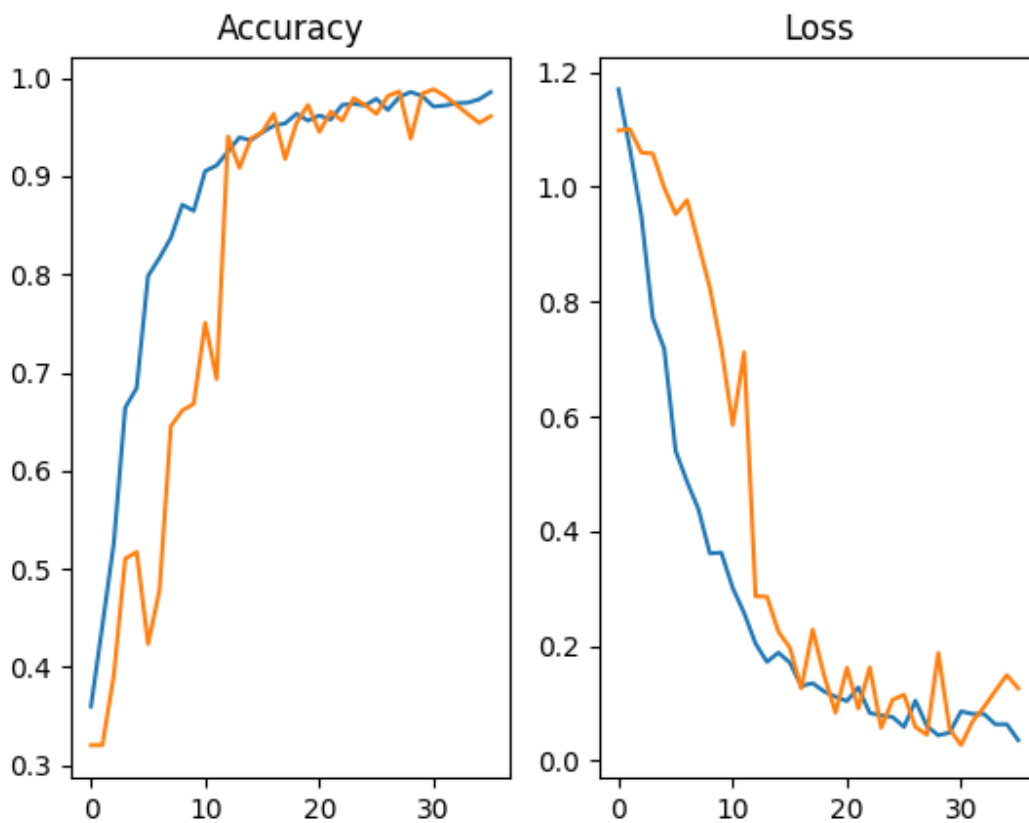
## Preobučavanje neuralne mreže

- Preobučavanje neuralne mreže (overfitting) podrazumeva da mreža gubi mogućnost generalizacije podataka tj. da model postiže visoku tačnost na trening skupu, ali to znanje ne može podjednako da iskoristi nad podacima koje do tada nije video. Do preobučavanja može doći ako je model previše kompleksan ili ukoliko se model trenira dovoljno dugo tako da mreža ima vremena da precizno namesti težinske faktore.
- Rano zaustavljanje (early stopping) je tehnika sprečavanja preobučavanja koja podrazumeva da se obučavanje neuralne mreže zaustavi kada greška na podacima koje neuralna mreža nije videla krene da raste.
- Tokom treniranja, greška na trening skupu će konstantno da opada dok će greška na validacionom skupu u jednom trenutku krenuti da raste – taj trenutak predstavlja trenutak ranog zaustavljanja.



## Performanse neuralne mreže

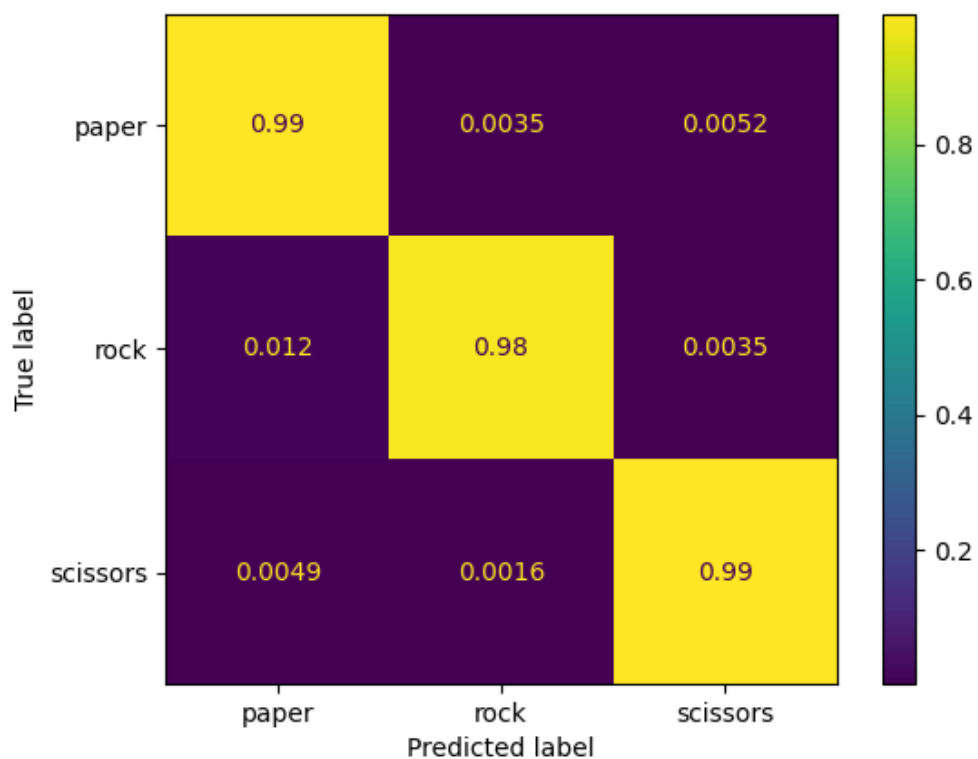
- Pratimo parametre accuracy i loss kroz epohe nad trening i validacionim skupom.
- Sa slike grafa vidimo da je tačnost mreže visoka što pokazuju i numerički podaci.



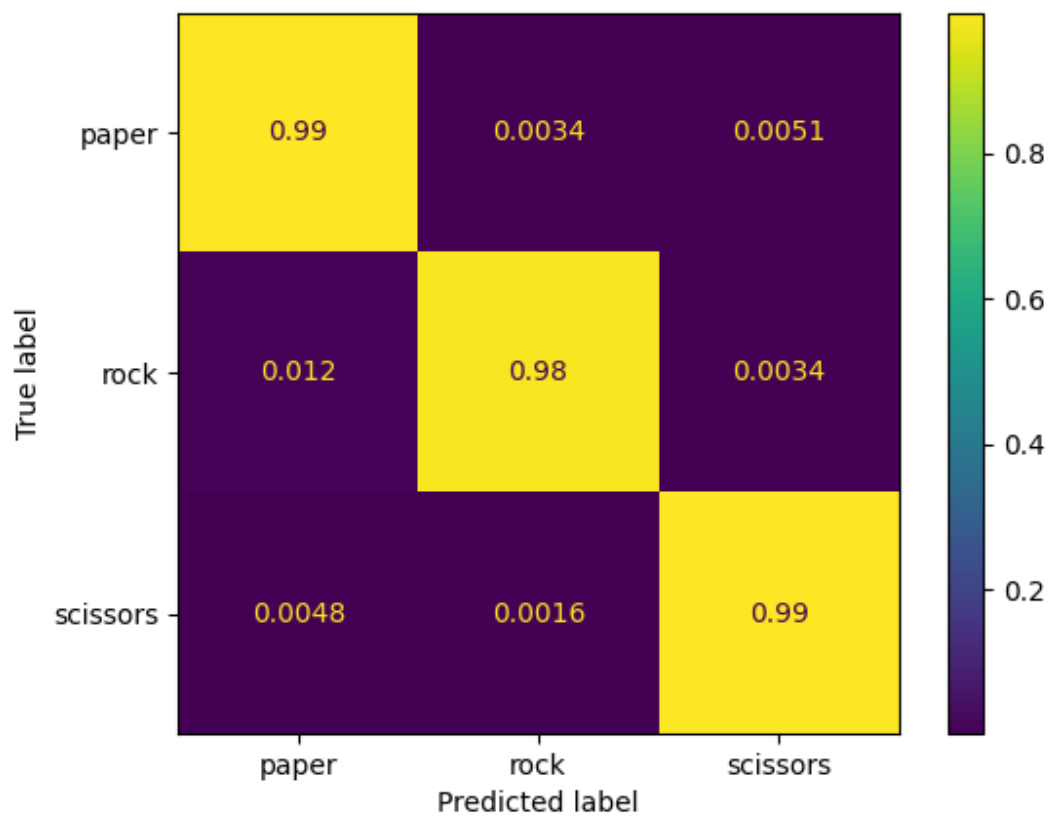
Tačnost modela je: 98.8558352402746%

## Konfuzionna matrica

Na glavnoj dijagonali konfuzione matrice nalazi se broj tačno klasifikovanih odbiraka za svaku klasu.



*Konfuzionna matrica za Xtrain skup*



*Konfuzionna matrica za Xval skup*

## Primerci dobro i loše klasifikovanih primeraka klasa

True: paper  
Pred: paper



True: scissors  
Pred: scissors



True: rock  
Pred: rock



True: scissors  
Pred: paper



True: paper  
Pred: rock



True: scissors  
Pred: paper

