



**UNIVERZITET U NIŠU**  
**ELEKTRONSKI FAKULTET**



**TEMA**

**MASTER RAD**

Studijski Program: Računarstvo i informatika

Mentor:

Prof. dr Leonid Stoimenov

Kandidat:

Aleksandra Stanojević 918

Niš, oktobar 2020. godine

Master rad  
TEMA

Zadatak:  
*Proučiti....*

Student:

Aleksandra Stanojević 918

Predsednik

Komisija za odbranu:

Datum prijave rada: \_\_\_\_\_

Datum predaje rada: \_\_\_\_\_

Datum odbrane rada: \_\_\_\_\_

\_\_\_\_\_  
Član

\_\_\_\_\_  
Član

TEMA

SAŽETAK

//TODO

**Ključne reči: Duboko učenje, Klasifikacija, Koronavirus, COVID-19, Konvolucijske neuronske mreže, Rendgenske slike pluća**

## TEMA NA ENGLSKOM

### ABSTRACT

//TODO

**Key words:** Deep learning, Classification, coronavirus, COVID-19, Convolutional Neural Network, Chest X-Ray images

# Sadržaj

<b>Uvod</b>	<b>6</b>
<b>Detekcija virusa COVID-19</b>	<b>8</b>
Prednosti CXR slika u odnosu na CT slike pluća	9
<b>Povezani radovi</b>	<b>10</b>
<b>Duboko učenje</b>	<b>12</b>
Kako funkcioniše?	13
Algoritam Bekpropagacije	14
Stohastički opadajući gradijent	15
<b>Digitalna obrada slike korišćenjem Dubokog učenja (Deep learning)</b>	<b>15</b>
Veštačke neuronske mreže (ANN)	15
Konvolucijske neuronske mreže (CNN)	15
<b>Korišćene tehnologije</b>	<b>17</b>
Python	17
Anaconda	17
Spyder	17
TensorFlow	18
Instalacija	20
Korišćenje	20
Keras	21
<b>Implementacija</b>	<b>23</b>
Opis skupa podataka	23
Augmentacija podataka	23
Metrika performansi	23
Transfer učenja	23
<b>Rezultati istraživanja</b>	<b>23</b>
<b>Zaključak</b>	<b>23</b>
<b>Literatura</b>	<b>23</b>



# Uvod

COVID-19 je novi koronavirus koji se pojavio krajem 2019. godine u Kineskoj oblasti Wuhan, a u 2020. godini proširio na ceo svet i postao svetski zdravstveni problem [1-5]. Ovaj virus inficira pluća i uzrokuje potencijalno smrtonosne respiratorne sindrome [6]. Prema poslednjim podacima ukupan broj potvrđenih slučajeva širom sveta je dostigao 33,502,322, od kojih je 1,005,281 smrtnih slučajeva, a države koje su najviše pogođene ovom pandemijom su USA, Indija, Brazil i Italija [7]. Glavna metoda za detekciju COVID-19 je lančana reakcija polimeraznom reverznom transkripcijom (skr. RT-PCR) koja može detektovati SARS-CoV-2 RNA iz respiratornih uzoraka (nazofaringealni i orofaringealni brisevi) [8]. Međutim, ova metoda je dosta specifična, dugotrajna (od nekoliko sati do celog dana), predstavlja naporan ručni postupak, a javlja se i problem nedostatka kompleta za testiranje usled veoma velike potražnje. Kako je testu potrebno više vremena za dobijanje rezultata u poređenju sa vremenom za širenje virusa među ljudima, uz ponekad i pogrešne rezultate (lažni negativni ili lažni pozitivni), mnogo istraživača radi na automatskoj detekciji virusa korišćenjem rendgenskih slika pluća (engl. Chest X-Ray - CXR) ili slika pluća korišćenjem metode kompjuterizovane tomografije (engl. CT scan). Neki radovi su čak i došli do zaključka da dijagnostikovanje korišćenjem slika pluća daje bolje rezultate od RT-PCR metode i da se treba uzeti u razmatranje da bude primarna tehnika za detekciju COVID-19 virusa [9]. Ovoj ideji doprinosi i činjenica da su medicinska istraživanja već pokazala da je infekcija pluća koju izaziva COVID-19 drugačija u odnosu na druge infekcije i da je na snimcima pluća radiolozi mogu detektovati [9-12].

Klasifikacija rendgenskih slika pluća, međutim, nije novi problem u polju Veštačke inteligencije. Naime, konvolucijske neuronske mreže (*skr. CNN*) su jedan od najpopularnijih modela dubokog učenja koji se koristi za digitalnu obradu slike. Do prodora ove metode došlo je sa ImageNet takmičenjem [13], na kome je stopa greške za prepoznavanje objekata prepolovljena u odnosu na dotadašnje metode. U najnovijim istraživanjima akcenat je na korišćenju CNN-a ili radi kreiranja metoda koji će najkorektnije detektovati segmente na snimcima pluća koji ukazuju na COVID-19 ili radi klasifikacije slika pluća na ona sa i bez COVID-19 infekcije, a sve sa ciljem, rane i brze detekcije i izdvajanja pacijenata koji su pozitivni na novi virus radi njihovog daljeg lečenja i izolacije koja bi sprečila širenje zaraze.

Ovaj rad se bavi proučavanjem digitalne obrade rendgenskih slika pluća korišćenjem dubokog učenja, odnosno konvolucijske neuronske mreže radi klasifikacije slika pluća sa COVID-19 infekcijom, bakterijskom infekcijom i slikama čistih pluća. Najpre je objašnjeno koje su to karakteristike koje se na slikama pluća mogu uočiti, a na osnovu kojih se mogu razlikovati pluća sa COVID-19 infekcijom. Zatim je dat osvrt na ranije radove na istu temu, a potom je

predstavljeno duboko učenje kao metoda za digitalnu obradu slika i detaljno obrađen metod konvolucijske neuronske mreže i načina njenog rada od prihvatanja ulaznih podataka do izdavanja rezultata. Takođe, predstavljene su tehnike augmentacije podataka i transfera znanja, jer je zbog male količine podataka bilo neophodno, na neki način, izbeći preterano podudaranje s podacima za trening (*engl. overfitting*).

Teorijski koncepti obrađeni u radu su zatim korišćeni prilikom realizacije praktičnog dela zadatka. Najpre su predstavljene korišćene tehnologije, a zatim i sama implementacija, od skupa podataka koji su korišćeni, preko preprocesiranja, kreiranja konvolucijske neuronske mreže do prikaza matrica, dijagrama i evaluacije korišćenjem standardne metrike. Korišćena je tehnika augmentacije podataka, kao i tehnika transfera učenja i to za nekoliko modela CNN-a, a zatim vršeno i poređenje njihovih performansi.

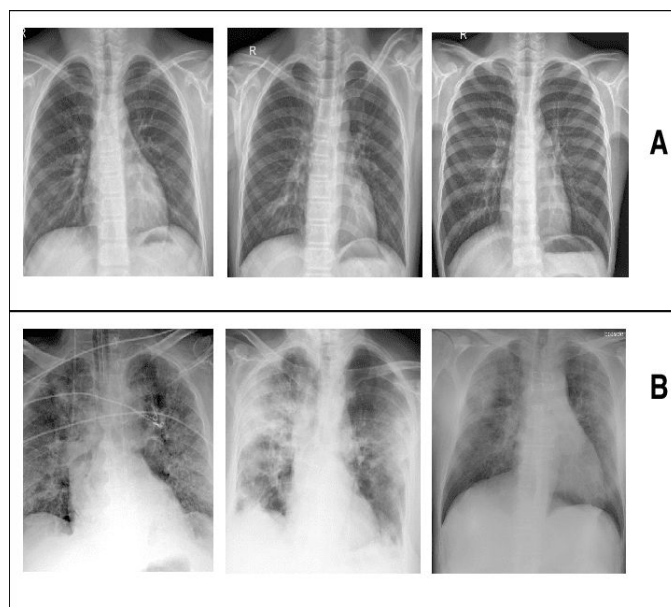


# Detekcija virusa COVID-19

Kao što je već rečeno, trenutno je RT-PCR osnovna metoda za detekciju virusa COVID-19, ali mnogo istraživača radi na automatskoj detekciji virusa korišćenjem radiografskih slika pluća (CXR i/ili CT slike) kao alternativnom metodu. Praksa je da je prvi korak korišćenje RT-PCR metode, a zatim se dalje lečenje, nakon što je virus detektovan, nastavlja uz stalno praćenje radiografskih slika od strane radiologa.

Medicinska istraživanja su pokazala da se kod pacijenata, koji su inficirani, mogu pomoću radiografskih snimaka uočiti karakteristične abnormalnosti. Na primer, Huang i drugi [11] identifikovali su da većina COVID-19 pozitivnih slučajeva iz njihovog istraživanja imaju bilateralne radiografske abnormalnosti u CXR slikama, dok su Guan i ostali [12] takođe uočili abnormalnosti u vidu “mlečnog stakla” (*engl. ground-glass opacity - GGO*), bilateralnih i intersticijskih (međuprostorna tečnost) abnormalnosti na CXR i CT slikama.

Naime, kada bolest napreduje u telu pacijenta pozitivnog na COVID-19 virus, dolazi do pojave tečnosti u malim vazдушnim vrećicama u plućima zvanima alveole [14]. Prisustvo ove tečnosti izaziva zapaljenje pluća, a napredak upale može se pratiti preko CXR i CT slika. Upala pluća se pojavljuje u vidu “mlečnih stakala”, a moguće je i stvaranje “mlečnih konsolidacija” (spajanje više “mlečnih stakala” u jedno).



**Slika XX:** Poređenje slika čistih i pluća sa COVID-19 infekcijom. Skup A su čista, a skup B COVID-19 pluća.

Medicinsko osoblje mora imati određeni kriterijum na osnovu koga će doneti odluku da li je pacijentu potrebna terapija kiseonikom ili priključenje na respirator ili isključenje sa respiratora

(ako je u fazi oporavka). Vizuelizacija “mlečnih stakala” u CT slikama igra veoma važnu ulogu i od velike je pomoći medicinskom osoblju prilikom donošenja odluka. Takođe je veoma važno tačno utvrditi o kojoj vrsti plućne bolesti se radi pre primene bilo koje terapije, pa je stoga analiza korišćenjem CXR i CT slika pluća od višestrukog značaja.

## Prednosti CXR slika u odnosu na CT slike pluća

Kao što je već prethodno rečeno, postoje dve mogućnosti kada govorimo o radiografskim snimcima, a to su CXR i CT slike pluća. U slučaju CT slika, imamo detaljniji prikaz grudnog koša u odnosu na CXR. Na primer, na CXR slikama se mogu uočiti abnormalnosti, dok CT treba da bude u stanju da pokaže tačnu lokaciju i ispita formaciju. CXR je 2D, a CT slike mogu biti i 3D. Prve se koriste radi ispitivanja gustih tkiva, dok CT može istovremeno da prikaže i kosti, meka tkiva i krvne sudove. Rentgenska oprema je mnogo manja i manje složena od CT skeniranja, jer se CT skener mora okretati oko pacijenta koji se skenira. [15]

Zbog ovih razlika se kroz dosta radova, od kojih će neki biti spomenuti u narednom poglavlju, provlači polemika oko izbora vrste radiografskih snimaka za detekciju COVID-19 virusa. U praktičnom delu ovog rada ispitivanja će se vršiti na CXR slikama, pa je stoga potrebno napomenuti koje su to karakteristike CXR snimaka koje su bile presudne prilikom izbora.

- **Brza trijaža<sup>1</sup>** - CXR snimci omogućavaju brzu trijažu pacijenata za koje se sumnja da su zaraženi COVID-19 infekcijom i može se vršiti paralelno sa virusnim testiranjem (za koje je potrebno više vremena) kako bi se broj pacijenata smanjio, naročito kada su u pitanju područja sa popunjenim kapacitetima (npr. New York, Italija, Španija). Takođe, ukoliko virusno testiranje nije moguće zbog nedostatka zaliha, CXR snimci predstavljaju dobru alternativu.
- **Dostupnost i pristupačnost** - CXR snimanje je dostupno i pristupačno u većini kliničkih centara i centara za snimanje, a pored toga smatra se standardnom opremom zdravstvenih sistema. Konkretno, CXR snimanje je mnogo dostupnije u odnosu na CT snimanje, naročito ako govorimo o zemljama u razvoju gde je trošak za opremu i održavanje CT skenera prevelik.
- **Prenosivost** - Postojanje prenosivih CXR sistema znači da se snimanje može izvoditi u domu osobe koja je u kućnoj izolaciji, čime se značajno smanjuje rizik od prenosa COVID-19 infekcije tokom transporta do zdravstvenih ustanova. Ovo nije slučaj sa CT skenerom koji predstavlja fiksni sistem i ne može se prenositi van zdravstvene ustanove.

---

<sup>1</sup> **Trijaža** je jedan od medicinskih sistema razvrstavanja pacijenata koji se primenjuje u uslovima masovnih događaja (velike nesreće, pandemije i sl), kako bi se omogućilo brzo prepoznavanje osoba koje će imati najveću korist ranog zbrinjavanja i brzog transporta u odgovarajuću zdravstvenu ustanovu.

## Povezani radovi

Analiza i detekcija COVID-19 infekcije su obimno istraživane teme u poslednjih nekoliko meseci. Jedan deo radova bavi se procenom broja novih slučajeva infekcije, broja oporavljenih i broja umrlih, odnosno vrši predviđanje daljeg širenja pandemije. Drugi deo radova bavi se pitanjima vezanim za otkrivanje infekcije na osnovu različitih pristupa dubokog učenja koristeći CXR i/ili CT slike pluća. Kako je detekcija virusa COVID-19 koristeći duboko učenje i CXR snimke tema i ovog rada u nastavku će biti dat osvrt na do sada postignute rezultate na ovom polju i to koristeći skup podataka koji će biti korišćen i u praktičnom delu ovog rada.

Kao što je već prethodno rečeno klasifikacija rendgenskih slika pluća nije novi problem u polju dubokog učenja. Objavljeno je dosta skupova podataka i nad njima obučeno puno neuronskih mreža koje su dale jako dobre rezultate. Nedavno su Cohen i saradnici objavili otvoreni skup podataka COVID-19 [16] koji sadrži CXR i CT slike pluća ljudi za koje je potvrđena infekcija ovim virusom ili se sumnja na COVID-19 ili neku drugu bakterijsku ili virusnu infekciju. Od objavljivanja skupa podataka COVID-19, mnogi istraživači pokušali su da klasifikuju te slike i kreirali su test skupove kombinovanjem ovog skupa podataka i drugih skupova CXR slika. Neki od tih pristupa dati su u nastavku.

Narin i saradnici [17] su kreirali mali skup podataka sa 50 COVID-19 pozitivnih slika pluća preuzetih iz Cohen-ovog skupa i 50 slika zdravih pluća preuzetih sa Kaggle-a<sup>2</sup> i vršili klasifikaciju na te dve klase. Predložili su CNN model zasnovan na transferu učenja korišćenjem ImageNet skupa podataka (ResNet50, InceptionV3 and Inception-ResNetV2) da bi se klasifikacija mogla izvršiti na tako malom skupu podataka, a da pritom može da da dobre rezultate predviđanja. Istraživanje je pokazalo superiornost ResNet50 modela u pogledu tačnosti i u fazi obuke i u fazi testiranja. Koristili su 5-cross validaciju i dobili tačnost od 98%.

Apostolopoulos i saradnici [18] su kombinovali snimke iz Cohen-ovog skupa i nekoliko drugih izvora da bi kreirali veći skup od 224 COVID-19 pozitivnih slika pluća, 700 slika pluća sa infekcijama koje nisu COVID-19 i 504 slika zdravih pluća. Oni su takođe predložili CNN model zasnovan na transferu učenja korišćenjem ImageNet skupa podataka za pet modela. Rezultati su pokazali da VGG19 dostiže najveću tačnost i to 93.48%, a korišćena je 10-cross validacija.

---

<sup>2</sup> **Kaggle**, ćerka firma kompanije Google LLC, je online zajednica praktičara mašinskog učenja. Omogućava korisnicima da pronađu, preuzmu i objave skupove podataka, istražuju i grade modele mašinskog učenja, rade sa drugim članovima zajednice i učestvuju u takmičenjima za rešavanje aktuelnih izazova.

Wang i saradnici [19] su trenirali Covid-Net, novu arhitekturu predstavljenu u njihovom radu. Koristili su dosta veliki skup podataka sa 358 COVID-19 pozitivnih slika pluća, 5 538 slika pluća sa infekcijama koje nisu COVID-19 i 8 066 slika zdravih pluća. Dizajn CNN-a konstituisana je od kombinacije  $1 \times 1$  konvolucija i residualnih modela da bi se omogućila dublja arhitektura, a korišćen je i obrazac projektovanje-ekspanzija-projektovanje-ekstenzija (*skr. PEPX*). Dostignuta tačnost u ovom radu je 92%.

Hemdan i saradnici [20] su kreirali programski okvir za duboko učenje, Covidx-Net i trenirali ga na skupom podataka sačinjenom od 50 slika pluća, 25 COVID-19 pozitivnih slika pluća i 25 slika pluća koja nisu zaražena virusom COVID-19. Polovina slika preuzeto je iz Cohen-ovog skupa. Ovaj rad koristi transfer učenja korišćenjem ImageNet skupa podataka za sedam modela i predstavlja uporednu studiju različitih arhitektura dubokog učenja. Rezultati su pokazali da su superiornije VGG19 i DenseNet201 u odnosu na ostale i da dostižu tačnost od 90%. Korišćena je 5-cross validacija.

Naravno postoji još radova koji koriste slike iz Cohen-ovog skupa ili pak onih koji su kreirali sopstveni skup snimaka i koji je ostao zatvoren za javnost, te im se ne može pristupiti, ali ovde su spomenuti samo neki od važnijih radova. Tabela XX predstavlja poređenje opisanih modela.

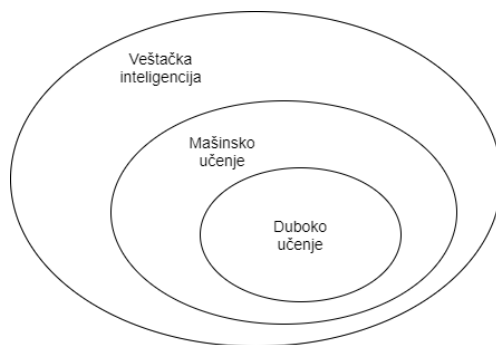
Studija	Skup podataka	Transfer učenja	Korišćeni model	Tačnost
Narin i saradnici [19]	50 COVID-19 50 zdravih pluća	Da (ImageNet skup)	ResNet50	98%
Apostolopoulos i saradnici [20]	224 COVID-19 700 pneumonia 504 zdravih pluća	Da (ImageNet skup)	VGG19	93.48%
Wang i saradnici [21]	358 COVID-19 5538 pneumonia 8066 zdravih pluća	Da (COVIDx skup)	Covid-Net	92%
Hemdan i saradnici [22]	25 COVID-19 25 zdravih pluća	Da (ImageNet skup)	Covidx-Net	90%

**Tabela XX:** Poređenje opisanih modela dubokog učenja koji za klasifikaciju koriste CXR slike iz Cohen-ovog skupa podataka COVID-19

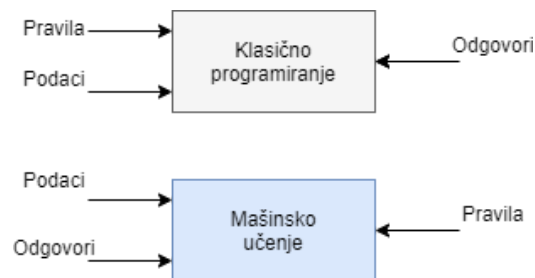
# Duboko učenje

U poslednjih nekoliko godina veštačka inteligencija je bila predmet velike medijske pažnje. Mašinsko učenje, duboko učenje i veštačka inteligencija pojavljuju se u bezbroj članaka i to vrlo često izvan tehnološki orijentisanih publikacija. Međutim ova grana računarstva nije mlada i nova, već njeni začeci datiraju iz sredine prošlog veka.

**Veštačka inteligencija** nastala je još 1950-ih godina kada se prvi put javilo pitanje automatizacije intelektualnih zadataka koje obično obavljaju ljudi. Ona predstavlja širu oblast koja obuhvata mašinsko učenje i duboko učenje, kao i neke od pristupa koji uopšte ne uključuju učenje. Prvi programi bili su za igranje šaha sa računarom i oni su podrazumevali praćenje pravila, što je bila i dominantna paradigma sve do 1980-ih godina, a ovaj pristup poznat je kao *simbolička veštačka inteligencija*. Međutim, kako ovaj pristup nije mogao da reši složenije logičke zadatke, razvio se novi pristup, poznat kao mašinsko učenje. [21]



*Slika XX: Veštačka inteligencija, mašinsko učenje i duboko učenje*



*Slika XX: Mašinsko učenje kao novi pristup u računarstvu*

**Mašinsko učenje** počinje svoj razvoj sa pitanjem koje je postavio pionir veštačke inteligencije Alan Tjuring, a to je sposobnost mašine da samostalno uči kako da izvrši određen zadatak. U mašinskom učenju čovek dostavlja mašini podatke i odgovore koje očekuje, a ona sama definiše pravila na osnovu kojih će od podataka dobiti odgovore. Ono što je ideja ovog pristupa je pronalaženje transformacije koju je potrebno izvršiti nad podacima tako da se dobije odgovarajuće rešenje (odgovori). [21]

Jedan od načina podele algoritama za mašinsko učenje je na algoritme za :

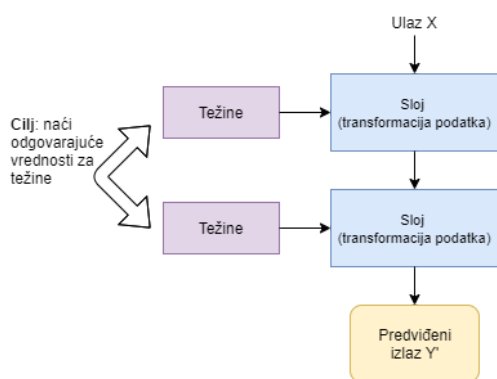
- Nadgledano učenje - algoritmi koji grade matematički model iz skupa podataka koji sadrže i ulaze i željene izlaze;

- Polunadgledano učenje – algoritmi koji razvijaju matematičke modele iz nepotpunih podataka, gde deo ulaznih podataka nema ulazne labele;
- Nenadgledano učenje - algoritmi koji razvijaju matematičke modele iz skupova podataka koji sadrže samo ulaze i ne sadrže željene izlazne modele;

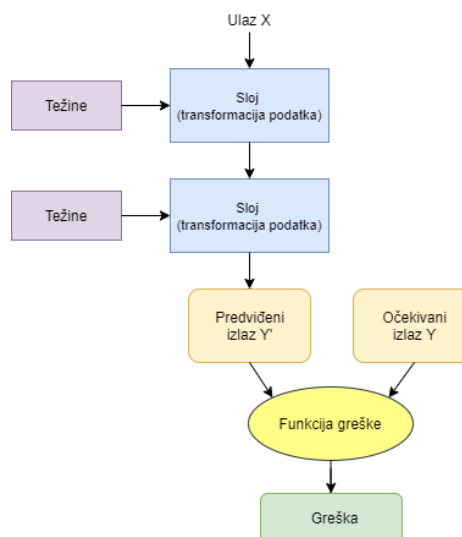
**Duboko učenje** je specifična podoblast mašinskog učenja koja podrazumeva pristup učenju kroz slojeve odnosno sukcesivne transformacije podataka. Dubina modela predstavlja koliko slojeva čini posmatrani model. Neki modeli uključuju desetine ili čak stotine uzastopnih slojeva, a učenje je automatsko prilikom izlaganja podacima za treniranje. Kod dubokog učenja, ovi slojeviti modeli su poznati pod nazivom neuronske mreže.

## Kako funkcioniše?

Način na koji duboko učenje funkcioniše podrazumeva da se specifikacija onoga što sloj radi sa ulaznim podacima čuva u njegovim težinama ili parametrima sloja (Slika XX 1.7). U ovom kontekstu, učenje znači pronalaženje skupa vrednosti za težine svih slojeva u mreži koji će omogućiti pravilno mapiranje ulaznih podataka u odgovarajuće izlazne podatke. Međutim duboka neuronska mreža može sadržati desetine miliona parametara, pa ju je zbog toga nije jednostavno kontrolisati. Da bi mogli upravljati izlazom neuronske mreže, neophodno je izmeriti koliko je on daleko od očekivanog rezultata. Ovo je zadatak funkcije greške (*engl. loss function*) ili kako se drugačije naziva ciljne funkcije (*engl. objective function*). [21]



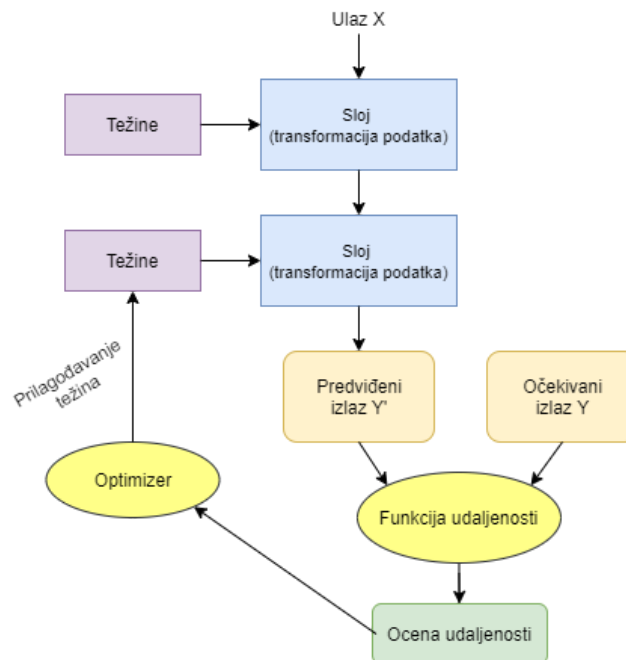
*Slika XX1.7: Parametri neuronske mreže, odnosno težine*



*Slika XX1.8: Funkcija greške meri kvalitet predviđanja neuronske mreže*

**Funkcija greške** uzima predviđanja koja su izlazne vrednosti mreže i stvarne ciljeve (izlazne vrednosti koje želimo da dobijemo od mreže) i izračunava grešku, beležeći postignuće mreže za

konkretni skup ulaznih podataka. (Slika XX 1.8) Osnovni štos u dubokom učenju je upotreba ove greške kao vodilje za prilagođavanje težina u takvom smeru da će to izazvati smanjenje greške za posmatrani skup ulaznih podataka (Slika XX 1.9). Ovo prilagođavanje težina je posao **optimizera** (engl. *optimizer*) koji implementira takozvani **Algoritam Bekpropagacije**, centralni algoritam u dubokom učenju.



*Slika XX1.9: Optimizer koristi grešku da bi prilagodio težine*

U početku se težinama mreže dodeljuju slučajne vrednosti, tako da mreža sprovodi niz slučajnih transformacija. Naravno, rezultat je tada daleko od onog kakav bi trebao biti, pa je samim tim i greška velika. Ali sa svakim podatkom koji mreža obradi, težine se prilagođavaju i greška povećava. Ovo je petlja treninga koja, ponovljena dovoljan broj puta, daje vrednosti težinama koje minimizuju grešku. Mreža sa minimalnom greškom je ona čiji su izlazni podaci najbliži onima koje očekujemo, odnosno predstavlja istreniranu mrežu.

## Algoritam Bekpropagacije

Iako su osnovne ideje o neuronskim mrežama nastale još 1950-ih godina, nekoliko decenija je bilo potrebno da se pronade efikasan način za treniranje velikih neuronskih mreža. Naime ovaj problem rešen je pojavom Algoritma Bekpropagacije.

Najteži zadatak dubokog učenja je prilagođavanje težina modela (neuronske mreže) u takvom smeru da će to izazvati smanjenje greške. Međutim, postavlja se pitanje kako odrediti za pojedinačni koeficijent težine u neuronskoj mreži da li ga je potrebno smanjiti ili povećati i za koliko. Odgovor na ovo pitanje leži u činjenici da su sve operacije koje se koriste u neuronskoj mreži diferencijabilne, te se može izračunati gradijent na osnovu koeficijenata težina mreže. Zatim je koeficijente potrebno pomeriti suprotno od gradijenta, izazivajući smanjenje greške.

*Stohastički opadajući gradijent*

## **Digitalna obrada slike korišćenjem Dubokog učenja (Deep learning)**

Veštačke neuronske mreže (ANN)

Konvolucijske neuronske mreže (CNN)



# Korišćene tehnologije

U ovom poglavlju opisani su programski jezici, platforme, biblioteke i okruženja koji su korišćeni za praktičnu realizaciju rada.

## Python

Python je programski jezik interpretatorskog tipa, objektno-orijentisani jezik visokog nivoa, interaktivni, poseduje dinamičku semantiku i opšte je namene. Nastao je 1991. godine, stvorio ga je Gvido van Rosum (*engl. Guido van Rossum*), a ime je dobio po kultnoj britanskoj komediji „Monty Python”. Akcenat kod ovog jezika je na njegovoj jednostavnoj sintaksi, preglednosti koda i velikoj fleksibilnosti. Koristi se za veb razvoj, veštačku inteligenciju, mašinsko učenje, operativne sisteme, razvoj mobilnih aplikacija i video igrice.

## Anaconda

Anaconda je besplatna distributivna platforma otvorenog koda (*engl. open-source*) za jezike Python i R. Ova platforma donosi puno alata koji se koriste u Nauci o podacima (*engl. Data science*) i Mašinskom učenju (*engl. Machine learning*) samo jednom instalacijom, tako da pruža kratku i jednostavnu postavku. Anaconda koristi koncept kreiranja promenljivih okruženja, tako da može da izoluje različite biblioteke i njihove verzije. Obuhvata veliki broj paketa kao što su conda, numpy, scipy, ipython, notebook i mnoge druge, a može se koristiti na Windows, Linux i macOS operativnim sistemima.

## Spyder

Spyder je integrisano razvojno okruženje (*engl. Integrated Development Environment - IDE*) za naučno programiranje u programskom jeziku Python i otvorenog je koda. Inicijalno ga je kreirao i razvio Pierre Raybat 2009. godine, a od 2012. godine na njegovom održavanju i kontinuiranom unapređenju radi tim naučnika i programera, kao i čitava python zajednica.

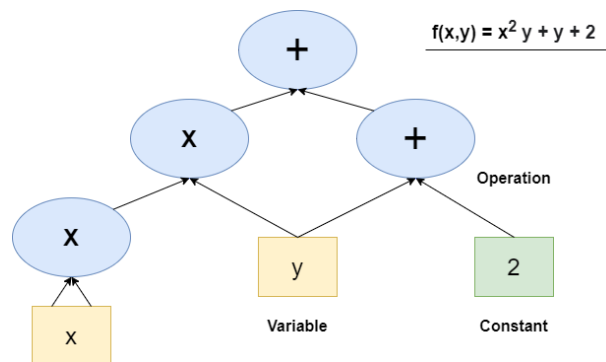
Spyder uključuje uređivanje, interaktivno testiranje, debugiranje i introspektivne karakteristike, a dostupan je preko Anaconda platforme. Pokreće se ili upotrebom komande - *spyder* - iz Anaconda terminala ili jednostavnim klikom na ikonicu Spyder u Anaconda Navigator-u<sup>3</sup>, koji je deo Anaconda platforme. Može se koristiti na Windows, Linux i macOS operativnim sistemima.

---

<sup>3</sup> **Anaconda Navigator** je grafički-korisnički interfejs (*engl. Graphical User Interface - GUI*) koji je uključen u Anaconda platformu i omogućava pokretanje aplikacija i jednostavno upravljanje paketima i promenljivama okruženja bez potrebe za upotrebom komandi komandne linije (*engl. command-line commands*).

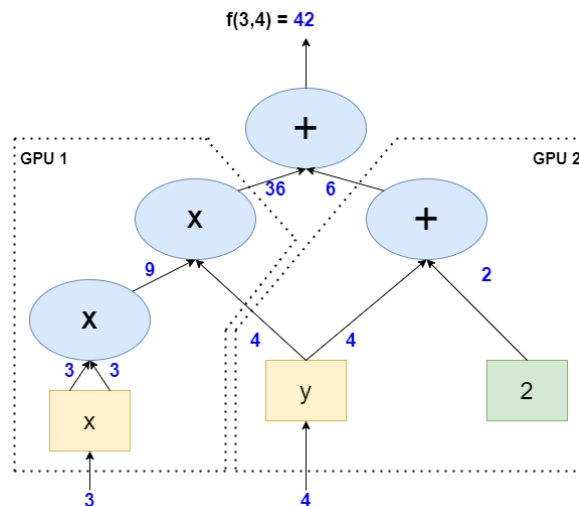
## TensorFlow

TensorFlow je softverska biblioteka otvorenog koda (*engl. open-source*) za numerička izračunavanja, a posebno je pogodna i prilagođena za jako velike zadatke iz oblasti mašinskog učenja (*engl. machine learning*). Princip rada ove biblioteke podrazumeva da se u programskom jeziku Python definiše graf operacija koje je potrebno obaviti (kao na primeru sa Slika XX), a zatim TensorFlow uzima taj graf i efikasno ga izvršava koristeći optimizovani C++ kod. [100]



Slika XX: Jednostavni graf operacija

Još važnije je to da je moguće raspardati graf operacija na nekoliko delova i izvršavati ih paralelno na većem broju CPU-a ili GPU-a (kao što je prikazano na Slika XX).



Slika XX: Paralelno izvršavanje na više CPU-a/GPU-a/servera

TensorFlow takođe podržava distribuirano izvršavanje, tako da se mogu trenirati velike neuronske mreže na ogromnim skupovima podataka u razumnom vremenskom roku podelom

izračunavanja na nekoliko hiljada servera. Ova biblioteka može da trenira mrežu od nekoliko miliona parametara na skupu obuke sačinjenog od nekoliko biliona instanci gde svaka ima po nekoliko miliona karakteristika. Ovo nije iznenađenje, ako uzmemo u obzir činjenicu da je TensorFlow razvio Google Brain tim stručnjaka i da on pokreće Google-ove servise kao što su Google Cloud Speech, Google Photos i Google Search.

Kada je TensorFlow postao otvorenog koda (*engl. open source*) Novembra 2015. godine, postojalo je već puno popularnih i dostupnih biblioteka za duboko učenje (*engl. Deep Learning*) i one su sadržale dosta funkcionalnosti koje ima i TensorFlow. Ipak, čist dizajn, skalabilnost, fleksibilnost i veoma dobra dokumentacija (da se ne spomene i ime Google-a) su jako brzo podigli TensorFlow do ogromne popularnosti.

Neke od najzanimljivijih odlika TensorFlow-a date su u nastavku [100]:

- Pored Windows-a, Linux-a i macOS-a može se pokrenuti i na mobilnim uređajima i to i na IOS-u i na Android-u;
- Pruža veoma jednostavan Python API - TF.Learn2 (tensorflow.contrib.learn), koji je kompatibilan sa Scikit-Learn<sup>4</sup> bibliotekom;
- Pruža još jedan jednostavan API - TF-slim (tensorflow.contrib.slim) za pojednostavljeno kreiranje, treniranje i evaluaciju neuronskih mreža;
- Pruža i još nekoliko API-a visokog nivoa, koji su kreirani nezavisno, kao što su Keras i Pretty Tensor;
- Njegov osnovni Python API nudi dosta fleksibilnosti u kreiranju svih vrsta izračunavanja, uključujući bilo koju arhitekturu neuronskih mreža;
- Uključuje veoma efikasnu C++ implementaciju za veliki broj operacija mašinskog učenja (*engl. Machine Learning*), kao i C++ API za definisanje korisničkih operacija visokih performansi;
- Pruža nekoliko naprednih čvorova za optimizaciju radi traženja parametara koji minimizuju troškove operacija. Jako su jednostavni za upotrebu, jer TensorFlow automatski računa gradijent funkcija koje korisnik definiše (automatsko diferenciranje);
- Dolazi sa veoma dobrim alatom za vizuelizaciju - TensorBoard, koji omogućava navigaciju kroz graf operacija, krive učenja i slično;
- Za izvršavanje TensorFlow grafova, Google je kreirao klad servis (*engl. Cloud service*);
- Ima rastuću zajednicu koja radi na njegovom unapređenju i jedan je od trenutno najpopularnijih projekata otvorenog koda.

---

<sup>4</sup> **Scikit-learn** je besplatna biblioteka za mašinsko učenje u programskom jeziku Python. Sadrži algoritme za klasifikaciju, regresiju, klasterizaciju, kao i mnoge druge poznate algoritme, a prilagođena je saradnji sa Python numeričkim bibliotekama NumPy i SciPy.

## *Instalacija*

Korišćenje TensorFlow biblioteke je malo komplikovanije u odnosu na druge Python biblioteke, jer vrlo često izaziva konflikte ukoliko se ne uklapa sa verzijama drugih prisutnih biblioteka. Da bi se sprečili potencijalni konflikti potrebno je kreirati posebnu promenljivu okruženja na Anaconda platformi. Ta će promenljiva sadržati verziju Pythona koji se uklapa sa TensorFlow bibliotekom i drugim paketima kao što su NumPy, Pandas i Matplotlib. [101]

Potrebno je otvoriti Anaconda terminal ili komandnu liniju i instalirati TensorFlow okruženje izdavanjem sledeće komande:

```
conda create -n tensorflow_env tensorflow
```

Ovo novo okruženje neće imati pristup drugim Python/Anaconda paketima koje smo već instalirali u drugom okruženju, te je potrebna ponovna instalacija svih paketa koje nameravamo da koristimo. Potrebno je najpre aktivirati TensorFlow okruženje, zatim instalirati sve potrebne pakete i nakon toga deaktivirati okruženje. Komanda za aktiviranje je:

```
conda activate tensorflow_env
```

Nakon aktiviranja okruženja na početku komandne linije ili terminala trebalo bi da vidimo naziv okruženja. Zatim, sledećom komandom vrši se instalacija najčešće korišćenih paketa:

```
conda install -n tensorflow_env scikit-learn pandas matplotlib
```

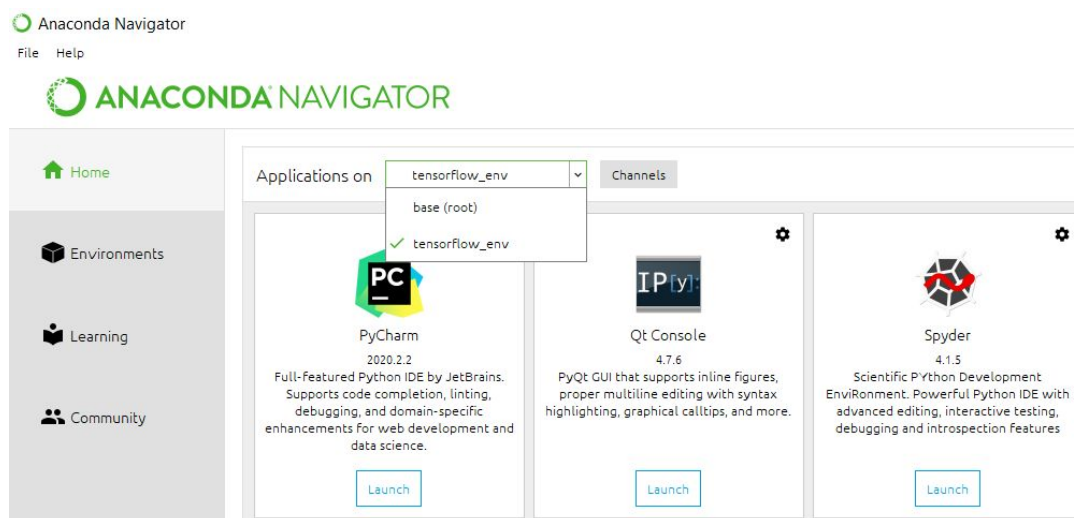
Nakon toga, sledećom komandom vršimo deaktiviranje okruženja, čiju smo instalaciju završili:

```
conda deactivate
```

## *Korišćenje*

Moguće je instaliranom okruženju i nadalje pristupati upotrebom Anaconda terminala ili komandne linije korišćenjem komandi za aktivaciju i deaktivaciju, ali daleko jednostavnije je korišćenjem Anaconda Navigator-a.

Na Slika XX prikazan je padajući meni koji nudi dve opcije, jednu za TensorFlow i drugu za okruženje koje smo imali pre instalacije.



*Slika XX: Prikaz dostupnih okruženja u Anaconda Navigator-u*

U novom okruženju potrebno je instalirati Spyder, da bi smo mogli da unosimo kod, a nakon njegovog instaliranja možemo uvesti TensorFlow i otpočeti njegovo korišćenje komandom:

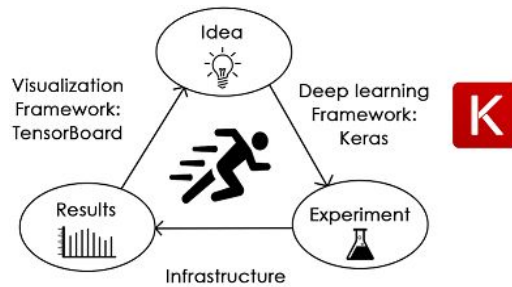
```
import tensorflow as tf
```

## Keras

Keras je biblioteka otvorenog koda za neuronske mreže napisana u programskom jeziku Python. Dizajnirana je tako da omogući brzo eksperimentisanje sa dubokim neuronskim mrežama, a fokus je na tome da bude jednostavna za korišćenje, modularna i proširljiva.

Neke od najzanimljivijih karakteristika ove biblioteke date su u nastavku [102]:

- Keras prati najbolje prakse za smanjenje kognitivnog opterećenja: nudi konzistentne i jednostavne API-je, minimizuje broj korisničkih akcija koje su neophodne za uobičajene slučajeve korišćenja i omogućava jasne i primenljive poruke o greškama. Takođe je prate obimna dokumentacija i niz tutorijala za programere;
- Keras je najviše korišćeno programsko okruženje za duboko učenje od top-5 poredničkih timova na Kaggle-u. Razlog je taj što Keras olakšava izvođenje novih eksperimenata i omogućava brže isprobavanje više ideja od konkurencije;



*Slika XX: Petlja napretka korišćenjem Keras API-a*

- Kako je Keras izgrađena nad Tensorflow 2.0 bibliotekom, predstavlja jako snažno okruženje koje može da se skalira na ogromne klastere GPU-a ili na čitav TPU pod<sup>5</sup> i to vrlo jednostavno;
- Keras modeli se mogu eksportovati na JavaScript i izvršavati direktno u pretraživaču ili na TF Lite<sup>6</sup> radi izvršavanja na IOS-u, Androidu ili embedded uređajima;
- Keras je centralni deo jako spregnutog TensorFlow 2.0 sistema, jer pokriva svaki korak procesa razvoja modela mašinskog učenja, od upravljanja podacima do hiperparametara za treniranje i krajnjeg rešenja;
- Keras koriste CERN i NASA, kao i mnoge druge naučne organizacije u svetu, jer ima veliku fleksibilnost čak i na najnižim nivoima za implementaciju proizvoljnih istraživačkih ideja, dok nudi opcione pogodnosti na visokom nivou radi ubrzavanja ciklusa eksperimentisanja;
- Zbog činjenice da je jako jednostavan za korišćenje i fokusiran na korisnička iskustva Keras je rešenje za duboko učenje koje biraju mnogi univerziteti i naširoko se preporučuje kao jedan od najboljih načina za upoznavanje sa dubokim učenjem.

<sup>5</sup> **TPU pod** je konfiguracija u Google-ovom centru za podatke koji ima niz TPU (*engl. Tensor Processing Unit*) uređaja međusobno povezanih veoma brzom mrežom. Kada je koristimo za izvršavanje sva izračunavanja mašinskog učenja koja su nam potrebna se distribuiraju na sve TPU uređaje.

<sup>6</sup> **TF Lite** (*engl. TensorFlow Lite*) je skup alata koji programerima omogućava izvršavaje TensorFlow modela na mobilnim, embedded i IoT (*engl. Internet of Things*) uređajima.

# Implementacija

Opis skupa podataka

Augmentacija podataka

Metrika performansi

Transfer učenja

## Rezultati istraživanja

## Zaključak

## Literatura

- [1] C.D.C. COVID, R. Team (February 12 - March 16, 2020) ‘Severe outcomes among patients with coronavirus disease 2019 (COVID-19)—United States’, *MMWR Morb Mortal Wkly Rep.* 69, 343–346.
- [2] A. Remuzzi, G. Remuzzi (March 13, 2020) ‘COVID-19 and Italy: what next?’, *The Lancet*, 395(10231), 1225-1228.
- [3] Joe Parkin Daniels (July 25, 2020) ‘COVID-19 cases surge in Colombia’, *The Lancet*, 396(10246), 227.
- [4] The Lancet (September 26, 2020) ‘COVID-19 in India: the dangers of false optimism’, *The Lancet*, 396(10255), 867.
- [5] E. Mahase (2020) ‘Coronavirus: covid-19 has killed more people than SARS and MERS combined, despite lower case fatality rate’, *BMJ*, 368:m641.
- [6] C. Huang, Y. Wang, X. Li, L. Ren, J. Zhao et al. (2020) ‘Clinical features of patients infected with 2019 novel coronavirus in Wuhan, China’, *The Lancet*, 395(10223), 497–506.
- [7] <https://www.worldometers.info/coronavirus/>, (n.d.).

- [8] V.M. Corman, O. Landt, M. Kaiser, R. Molenkamp et al. (2020) ‘Detection of 2019 novel coronavirus (2019-nCoV) by real-time RT-PCR’, *Eurosurveillance*, 25 2000045.
- [9] T. Ai, Z. Yang, H. Hou, C. Zhan et al. (Feb 26, 2020) ‘Correlation of Chest CT and RT-PCR Testing for Coronavirus Disease 2019 (COVID-19) in China: A Report of 1014 Cases’, *RSNA*. 296(2), E32–E40.
- [10] M.Y. Ng, E. YP Lee, J. Yang, F. Yang, X. Li, H. Wang et al. (Feb 13 2020) ‘Imaging profile of the COVID-19 infection: Radiologic findings and literature review’, *Cardiothoracic Imaging*, 2(1).
- [11] C. Huang Y. Wang et al. (2020) ‘Clinical features of patients infected with 2019 novel coronavirus in Wuhan, China’, *The Lancet*, 395, 497–506.
- [12] W. Guan, Z. Ni, Y. Hu, W. Liang, C. Ou, J. He et al. (2020) ‘Clinical characteristics of coronavirus disease 2019 in China’, *N Engl J Med*, 382, 1708-1720
- [13] A. Krizhevsky, I. Sutskever, and G. Hinton (2012) ‘ImageNet classification with deep convolutional networks’, *Advances in Neural Processing Systems*, 25, 1097–1105.
- [14] Neha Patha (September 23, 2020) ‘What Does COVID-19 Do to Your Lungs?’, *WebMD*. Available at: <https://www.webmd.com/lung/what-does-covid-do-to-your-lungs#1> (Accessed 30 September 2020)
- [15] C. Rachna (April 5, 2017) ‘Difference between X-ray and CT Scan’, *Bio Differences*. Available at: <https://biodifferences.com/difference-between-x-ray-and-ct-scan.html> (Accessed 02 October 2020)
- [16] J.P. Cohen, P. Morrison, L. Dao (2020) ‘COVID-19 image data collection’, *ArXiv Prepr.* ArXiv2003.11597.
- [17] A. Narin, C. Kaya, Z. Pamuk (2020) ‘Automatic detection of coronavirus disease (COVID-19) using X-ray images and deep convolutional neural networks’, *ArXiv Prepr.* ArXiv2003.10849.
- [18] I.D. Apostolopoulos, T.A. Mpesiana (2020) ‘Covid-19: automatic detection from x-ray images utilizing transfer learning with convolutional neural networks’, *Phys Eng Sci Med* 43, 635–640.
- [19] L. Wang, A. Wong (2020) ‘Covid-net: A tailored deep convolutional neural network design for detection of covid-19 cases from chest radiography images’, *ArXiv Prepr.* ArXiv2003.09871.
- [20] E.E.-D. Hemdan, M.A. Shouman, M.E. Karar (2020) ‘Covidx-net: A framework of deep learning classifiers to diagnose covid19 in x-ray images’, *ArXiv Prepr.* ArXiv2003.11055.
- [21] François Chollet (2018) ‘Deep Learning with Python’, 1st edn. United States of America: Manning Publications Co.
- [100] Aurélien Géron (2017) ‘Hands-On Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems’, 1st edn. United States of America: O'Reilly Media.
- [101] Michael Allen (2018) ‘Python for healthcare modelling and data science - 106. Installing and using TensorFlow in Anaconda’ Available at:



<https://pythonhealthcare.org/2018/12/19/106-installing-and-using-tensorflow-using-anaconda/> (Accessed 26 September 2020)

[102] François Chollet (2015) ‘Keras: the Python deep learning API’ Available at: <https://keras.io/> (Accessed 27 September 2020)