

AKADEMIJA TEHNIČKO-UMETNIČKIH STRUKOVNIH
STUDIJA BEOGRAD

ODSEK VISOKA ŠKOLA ELEKTROTEHNIKE I
RAČUNARSTVA

Marković Dušan

**Jedan pristup mapiranja podataka iz tabela u grafičku
dvodimenzionalnu reprezentaciju za potrebe treniranja
modela konvolucione neuronske mreže**

- master rad -



Beograd, jul 2022.

Kandidat: **Marković Dušan**

Broj indeksa: **RIN-56/17**

Studijski progra: **Računarsko inženjerstvo**

Tema: **Jedan pristup mapiranja podataka iz tabela u grafičku dvodimenzionalnu reprezentaciju za potrebe treniranja modela konvolucione neuronske mreže**

Osnovni zadaci:

- 1. Jedan pristup pripreme ulaznih podataka za svrhu obrade**
- 2. Jedan način generisanja slike na osnovu podataka iz tabela**
- 3. Razvijanje modela i procena performansi obučenog modela**

Mentor:

Beograd, jul 2022 godine.

dr Nemanja Maček

REZIME:

U ovom radu objašnjen je postupak pripreme i metodologija transformacije tabelarnih podataka u dvodimenzionalnu grafičku reprezentaciju. Za novodobijene podatke razvijen je model konvolucione neuronske mreže koji treba da izvrši binarnu klasifikaciju. U radu je urađena i analiza performansi modela kako bi se uočile prednosti i mane koje donosi ovakav pristup.

Ključne reči: transformacija tabečarnih podataka u sliku, konvolucione neuronske mreže

ABSTRACT:

This paper explains the data preparation procedure and methodology for transforming tabular data into a two-dimensional graphic representation. A convolutional neural network model was developed for the newly obtained data, which should perform binary classification. The paper also analyzed the performance of the model in order to see the advantages and disadvantages of this approach.

Keywords: transformation of tabular data into an image, convolutional neural networks

SADRŽAJ

1. Uvod.....	5
2. Neuronske mreže	6
2.1 Treniranje mreže.....	9
2.2 Optimizatori.....	10
2.3 Podela neuronskih mreža	13
2.4 Konvolucione neuronske mreže	14
2.5 Metodologija mapiranja podataka u dvodimenzionalnu grafičku prezentaciju	20
3. Priprema i analiza podataka	22
3.1 Metode filtriranja	22
3.2 Metode testiranja.....	24
3.3 Kombinovane metode	25
4.Praktična realizacija	28
4.1 Programski jezik	28
4.2 Radno okruženje	29
4.3 Eksperiment	30
5. Zaključak	45
6. IZJAVA O AKADEMSKOJ ČESTITOSTI	46
7. Literatura.....	47

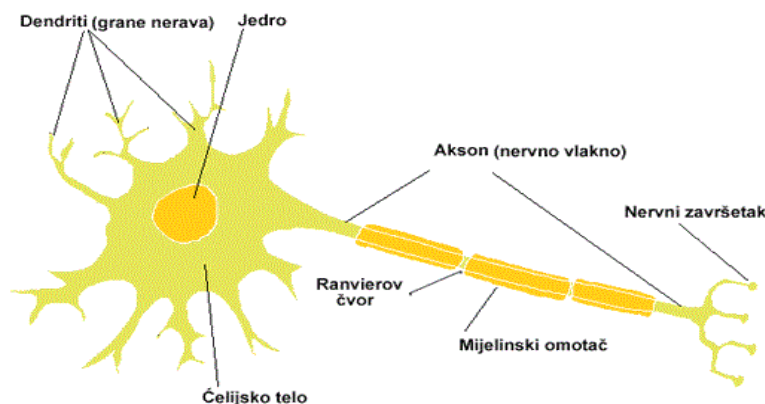
1. Uvod

Pojavom prvih sistema za automatsko donošenje odluka, binarna klasifikacija se javlja kao jedna od oblasti koja se konstatno razvija. Binarna klasifikacija rešava problem kategorizacije (razdvajanje) uzoraka na jednu od dve moguće klase (pouzdan ili loš kupac, zdrav ili bolestan pacijent, uključi ili isključi uređaj itd). Kako bi se performanse poboljšale, razvijani su novi i unapređivani su postojeći sistemi. U zavisnosti od tipa i oblika ulaznih podataka, specifični sistemi i pristupi dali su svoj doprinos u unapređenju performansi. Analiza i priprema podataka ima takođe bitnu ulogu u postizanju što boljih rezultata. Ovaj postupak je možda i jednako bitan ako ne i bitniji od razvijanja modela klasifikacije. Trenutno je najaktuelniji pristup rešavanja pomoću stabala odlučivanja. Ensambl algoritmi kao što su slučajne šume (engl. Random Forest), poboljšana stabla (engl. Extra Trees), iksbust (engl. XGBoost) daju zapažene rezultate kada su ulazni podaci predstavljeni u tabelarnom obliku. Veštačke neuronske mreže, iako dosta obećavajuće, za sad nisu dale očekivane rezultate kada su u pitanju tabelarni podaci. Razlog tome je što su ove mreže veoma dobri aproksimatori funkcija, ali imaju slabu generalizaciju. To dovodi do velike preprilagođenosti modela koji na nepoznatim podacima ne daje očekivano dobre rezultate. Glavna ideja ovog rada jeste da se ispita kakvi rezultati se mogu postići novim pristupom u kome se tabelarni podaci najpre transformišu u dvodimenzionalni grafički ekvivalent, a zatim provlače kroz konvolucione neuronske mreže. Eksperimentom je potrebno utvrditi da li i u kojoj meri modeli konvolucionih mreža pate od preprilagođavanja u ovakvim situacijama kao i da li se stepen istog problema može korigovati. Osim ovoga, eksperiment treba da pokaže kakve još uticaje i eventualne pogodnosti ovakav pristup može da postigne. Zbog velikih brzina koje konvolucione mreže postižu korišćenjem grafičke procesorske jedinice (GPU), procene su da bi ova metodologija mogla da da dobre rezultate pogotovo kada se radi o podacima sa velikim brojem obeležja.

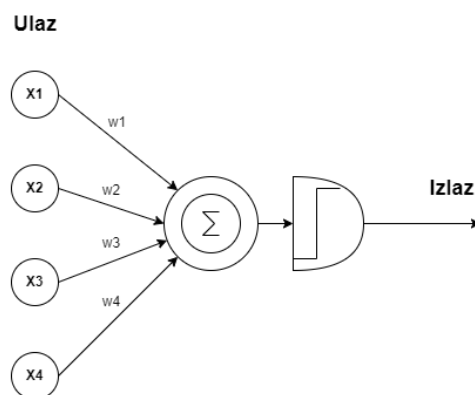
2. Neuronske mreže

¹Veštačka neuronska mreža spada u jedan od trenutno najaktuelnih alata mašinskog učenja koji se koristi u sistemima veštačke inteligencije. To je skup logički povezanih jedinica koji ima za cilj da na osnovu ulaznih podataka prognozira sledeći događaj (regresioni model) ili izračuna verovatnoću da ulaz pripada nekoj od već definisanih klasa (klasifikacioni model). Osnovni gradivni element veštačke neuronske mreže je logička jedinica neuron koji po strukturi podseća na biološki neuron koji se nalazi u mozgu živih bića. Ono što treba naglasiti je da veštački neuron ne predstavlja potpunu aproksimaciju pravog (biološkog) i da po načinu funkcionisanja nemaju preterane sličnosti. Razlog tome je jednostavan, a oslanja se na činjenici da ni stručnjaci iz domena neurologije ne mogu sa sigurnošću da tvrde kako u potpunosti funkcioniše biološki neuron. I pored svih razlika, veštačke neuronske mreže daju zapažene rezultate u različitim domenima istraživanja i praktične primene veštačke inteligencije u realnim uslovima.

²Nastanak neuronskih mreža vezuje se za 1943. godinu i istraživački rad Voltera Pirsa i Vorena Mekolaha. Njih dvojica su predstavili koncept neurona koji je i dan danas u osnovi isti. Pojavljivanjem ovakvog otkrića pobudio je veliko interesovanje za dalje istraživanje stručnjaka neurologije i računarskih nauka. I pored velikog revolucionarnog otkrića i entuzijazma za dalje razvijanje, koncept neuronskih mreža doživeće par „zima veštačke inteligencije“ (eng. AI winter) tokom svoje istorije koje su uzrokovane što zbog logičkih nedosletka što zbog tehničkih (ne)mogućnosti datog vremena. Ipak, krajem 90-ih godina razvijanje i korišćenje neuronskih mreža vraća se na velika vrata što je direktna posledica eksponencijalnog rasta računarskih sposobnosti. Od tada neuronske mreže nalaze široku primenu u različitim oblastima veštačke inteligencije: računarski vid, obrada prirodnog jezika (tekst, glas), generisanje novih sadržaja, itd. Na slici 2.1. prikazana je ilustracija anatomije biološkog neurona dok je na slici 2.2. prikazana logička struktura veštačkog neurona.

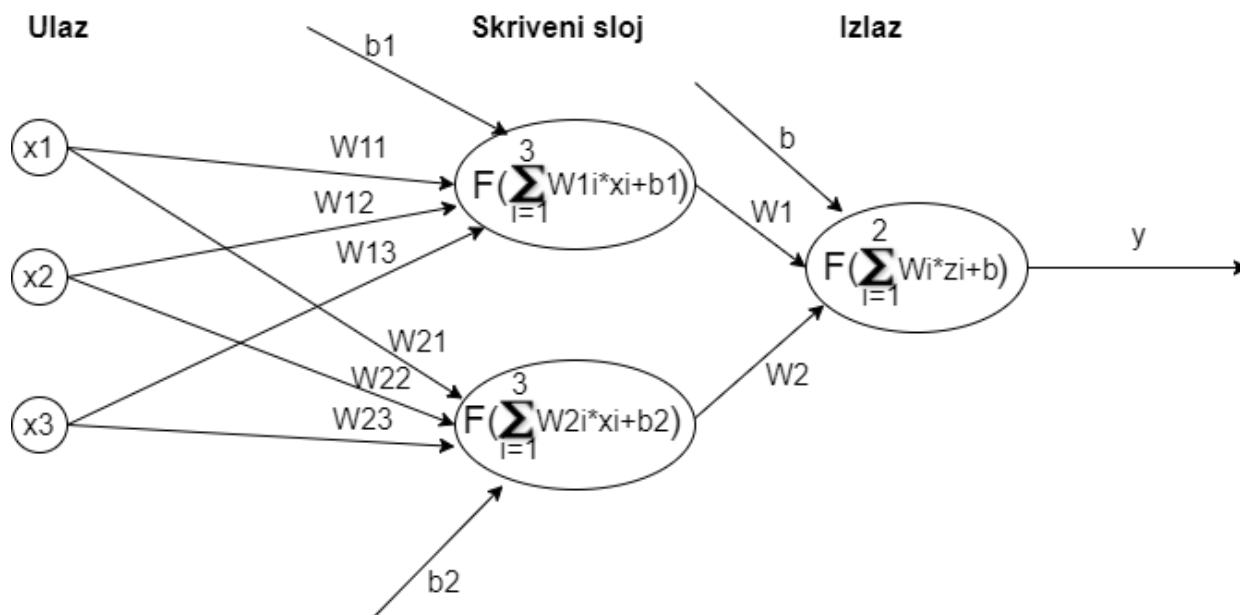


Slika 2.1- Ilustracija anatomije biloškog neurona [3]



Slika 2.2- Logička struktura veštačkog neurona



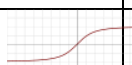

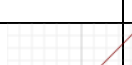
⁴Najosnovniji oblik neuronske mreže sastoji se od tri sloja: ulaz, skriveni sloj i izlaz. Ulazni sloj je takođe poznat i kao čvor ulaznih informacija koje se iz spoljašnjeg sveta prosleđuju modelu. Prosleđivanje se može vršiti u funkciji učenja ili izvođenja zaključaka. Ulazni čvor prosleđuje informacije sledećem čvoru tj. skrivenom sloju. Skriveni sloj predstavlja kolekciju neurona koji obrađuju ulazne podatke. Izlazni sloj se sastoji, u zavisnosti od tipa modela, od jednog ili više čvorova koji na osnovu informacija koje dobija od skrivenog sloja daje rezultat tj. izlaz modela. Ukoliko se radi o regresionom ili modelu binarne klasifikacije, izlaz se sastoji od jednog čvora u slučaju višeklasnog klasifikacionog modela izlaz ima po jedan čvor za svaku od mogućih klasa izlaza. Na slici 1.3 prikazana je logička struktura jednostavne mreže koja se sastoji od tri ulazna čvora, dva neurona u skrivenom sloju i jednog izlaza.



Slika 2.3- Jednoslojna neuronska mreža

Na slici 2.3 ulazni podaci obeleženi su simbolima x_1, x_2, x_3 i simbolom z za izlaz iz neurona skrivenog sloja. Težinski koeficijenti ulaza obeleženi su simbolom W , dok je korektivni faktor označen simbolom b . Težinski koeficijent i korektivni faktor inicijalno mogu imati slučajne vrednosti, vrednost 1, slučajne vrednosti u normalnoj distribuciji itd. U procesu učenja ove vrednosti se adaptiraju tako da se na izlazu dobije željeni rezultat. Krajnji rezultat obeležen je simbolom y . Svaki od neurona u mreži, osim gorespomenutih vrednosti, ima i aktivacionu funkciju koja je na slici obeležena sa F .⁵ Aktivaciona funkcija obezbeđuje stvaranje nelinearne zavisnosti u modelu. Ulaz aktivacione funkcije je zbir sume proizvoda ulaznih vrednosti i njihovih težinskih koeficijenata i korektivnog faktora neurona. Svaki neuron iz skrivenog sloja je povezan sa ulaznim vrednostima tako da broj ulaznih vrednosti određuje i broj težinskih koeficijenata neurona. Menjanje vrednosti težinskog i korektivnog faktora utiče da izlazna vrednost modela bude bliže ili dalje očekivanoj vrednosti. Ovaj postupak se zove učenje ili treniranje modela. Kriterijum biranja aktivacione funkcije postavlja se na osnovu modela i optimizatora koji se koristi. Za aktivacionu funkciju izlaznog čvora/ova klasifikacionog modela obično se bira *softmax* funkcija koja ima osobinu da su izlazne vrednosti u opsegu od 0-1 i da je ukupan zbir svih izlaza 1, tako da pojedinačna vrednost zapravo predstavlja verovatnoću datog ishoda. U tabeli 2.1. prikazane su neke od najkorišćenijih aktivacionih funkcija, kao i izgled i njihove osnovne osobine.

Tabela 2.1- Aktivacione funkcije

Naziv	Jednačina	Graf	Interval	Monotona
Sigmoidna funkcija	$f(x) = \sigma(x) = \frac{1}{1 + e^{-x}}$		(0,1)	Da
Hiperbolički tangens	$f(x) = \tanh(x) = \frac{(e^x - e^{-x})}{(e^x + e^{-x})}$		(-1,1)	Da
ISRU	$f(x) = \frac{x}{\sqrt{1 + \alpha x^2}}$	)	Da
ISRLU	$f(x) \begin{cases} \frac{x}{\sqrt{1 + \alpha x^2}} & \text{za } x < 0 \\ x & \text{za } x \geq 0 \end{cases}$		$(\frac{-1}{\sqrt{\alpha}}, \infty)$	Da
ReLU	$f(x) = \begin{cases} 0 & \text{za } x < 0 \\ x & \text{za } x \geq 0 \end{cases}$			Da

2.1 Treniranje mreže

⁶Postupak učenja formalizovan je kao postupak minimiziranja funkcije gubitka. Funkcija gubitka igra vitalnu ulogu u korišćenju neuronskih mreža. Ona definiše zadatak koji mreža treba da uradi i obezbeđuje meru kvaliteta reprezentacije potrebne za učenje. Izbor odgovarajuće funkcije gubitka zavisi od tipa problema i tipa neuronske mreže koja se koristi. Funkcija gubitka predstavlja zbir funkcije greške i regulacionog izraza. Funkcija greške (greška predikcije) je značajan deo izraza gubitka i pokazuje u kojoj meri izlaz modela odstupa od očekivane vrednosti.

⁷Greška predikcije može se podeliti u dve kategorije shodno tipu podataka koji se koristi. Jedna se odnosi na grešku koja se dobija nad podacima za treniranje, a druga na grešku nad test podacima. Neke od najznačajnijih funkcija za grešku predikcije date su u sledećem delu.

Srednja kvadratna greška

Srednja kvadratna greška (eng. Mean squared error, MSE) predstavlja prosečnu vrednost kvadrirane razlike između izlaza modela i očekivanog rezultata.

$$MSE = \frac{\sum (izlaz - očekivanaVrednost)^2}{ukupanBrojUzoraka}$$

Normalizovana kvadratna greška

Normalizovana kvadratna greška (eng. Normalized squared error, NSE) predstavlja količnik kvadrirane razlike, između izlaza modela i očekivanog rezultata, i normalizacionog koeficijenta. Ako je vrednost ove greške jednaka nuli onda se radi o idealnoj predikciji. Ukoliko model treba da rešava regresioni problem onda je ovo podrazumevani tip greške koji će se koristiti u procesu učenja.

$$NSE = \frac{\sum (izlaz - očekivanaVrednost)^2}{normalizacioniFaktor}$$

Težinska kvadratna greška

Težinska kvadratna greška (eng. Weighted squared error, WSE) koristi se kod modela binarne klasifikacije gde postoji neuravnoteženost između broja pozitivnih i negativnih uzoraka. Postoje dva težinska faktora, jedan za pozitivne uzorke, a drugi za negativne.

$$WSE = faktorPozitivnihVrednosti \sum (izlaz - očekivanaPozitivnaVrednost)^2 \\ + faktorNegativnihVrednosti \sum (izlaz - očekivanaNegativnaVrednost)^2$$

Greška unakrsne entropije

Greška unakrsne entropije (eng. Cross entropy error, COE) koristi se u situacijama kada se rešava binarna klasifikacija gde su jedino moguće izlazne vrednosti 0 i 1. Glavna prednost ove formule je što izrazito povećava vrednost ako je očekivana vrednost 1, a dobijeni izlaz je 0 i obrnuto. Idealni model treba da ima grešku unakrsne entropije jednaku nuli.

$$COE = \frac{-1}{brojUzoraka * \sum izlaz * \log(očekivanaVrednost)}$$

Greška Minkovskog

Greška Minkovskog (eng. Minkowski error, ME) je veoma slična srednjoj kvadratnoj, s tim da se kod ove funkcije stepenovanje vrši vrednostima od 1 do 2, obično 1.5. Ova vrednost se naziva minkovski parametar (mp).

$$ME = \sum \frac{(izlaz - očekivana_{vrednost})^{mp}}{ukupan_broj_uzoraka}$$

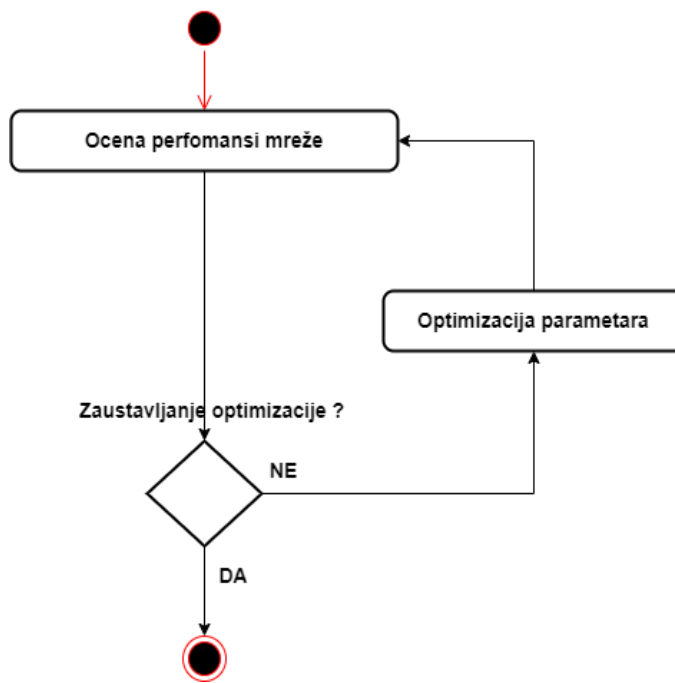
2.2 Optimizatori

⁸Algoritam optimizacije (optimizator) obavlja najveći deo posla u postupku učenja neuronske mreže. Postoje puno različitih algoritama optimizacije koji se međusobno razlikuju po brzini, preciznošću i memorijskim zahtevima. U ovom poglavlju biće ukratko opisani neki od

najznačajnijih tipova optimizatora. Kao što je već rečeno, proces učenja neuronske mreže ogleda se u tome da se optimizuju težinski faktori tako da minimiziraju funkciju gubitka koja je definisana. Funkcija gubitka je, matematički posmatrano, nelinearna funkcija težinskih faktora tako da je nemoguće definisati idealan algoritam za pronalaženje minimuma. Algoritam optimizacije vrši pretragu u prostoru težinskih i korektivnih faktora (u daljem tekstu parametri) tražeći vrednosti koje će usloviti smanjenje funkcije gubitka. Postupak učenja se odvija iterativno tako da se u svakom koraku (epohi) računa kako su promene na parametrima uticale na performanse. Na ovaj način, učenje neuronske mreže započinje dodeljivanjem početnih vrednosti parametrima, a nakon toga se odvija iterativni postupak optimizacije parametara. Proces učenja se izvršava sve dok se ne dostigne neki od definisanih kriterijuma. Razlozi za zaustavljanje postupka optimizacije parametara/učenja mogu biti različiti. Najčešći razlozi navedeni su u sledećoj listi:

- Poboljšanje funkcije gubitka u jednoj epohi manje je od definisanog kriterijuma
- Funkcija gubitka je dostigla željenu vrednost
- Dostignut je maksimalni broj epoha
- Prekoračeno je maksimalno vreme obrade

Na slici 2.2.1 prikazan je dijagram toka koji opisuje postupak učenja.



Slika- 2.2.1- Tok procesa učenja mreže

Kako ne postoji idealan algoritam optimizacije, razvili su se različiti algoritmi koji se razlikuju po performansama, memorijskim zahtevima, brzinama itd. Odluka o optimizatoru koji će se koristiti donosi se u zavisnosti od tipa arhitekture mreže, podataka koji se koriste i kako su pripremljeni. U sledećem delu navedeni su neki od najkorišćenijih optimizatora koji se koriste sa njihovim osnovnim osobinama.

Gradijentni spust (engl. Gradient descent, GD)

Ovo je jedan od najkorišćenijih algoritama koji se koristi u nepovratnim mrežama. Parametri se optimizuju u svakom koraku u pravcu negativnog gradijenta funkcije gubitka.

$$noviParametri = parametri - gradijentGubitka * stopaUčenja$$

Stopa učenja obično se prilagođava u svakoj epohi učenja.

Konjugovani gradijent (engl. Conjugate gradient, CG)

Pretraga najboljih parametara u ovom algoritmu vrši se u konjugovanom pravcu što rezultira bržom konvergencijom nego što čini algoritam gradijentnog spusta.

$$noviParametri = parametri - konjugovaniGradijent * stopaUčenja$$

Stopa učenja obično se prilagođava u svakoj epohi učenja.

Kvazi-Njutnov method (engl. Quasi-Newton method, QNM)

Njutnov metod koristi matricu drugog derivata funkcije gubitka kako bi izračunao pravac učenja. Zbog korišćenja informacija višeg reda, metod daje pravac promena parametara sa većom tačnošću ali ujedno je i komplikovaniji i zahtevniji za računanje. Kvazi Njutnov metod baziran je na originalnom metodu ali ne zahteva računanje matrice drugog derivata. Umesto toga ovaj metod računa približnu vrednost inverzne Hesijanove matrice (*hsm*) u svakoj iteraciji algoritma koristeći samo informacije o gradijentu.

$$noviParametri = parametri - hsm * gradijent * stopaUčenja$$

Stopa učenja obično se prilagođava u svakoj epohi učenja.

Levenbergov-Markvardov algoritam (engl. Levenberg-Marquardt algorithm, LM)

Ovaj algoritam dizajniran je da postigne visoku preciznosti učenja bez računanja Hesijanove matrice. Algoritam se može primeniti samo u slučajevima kad se koristi funkcija greške koje

računaju sumu kvadratnih grešaka (SSE, MSE, NSE). Takođe zahteva i računanje gradijenta i Jakobijeve matrice (jm) funkcije gubitka.

$$noviParametri = parametri - faktorPrigušenja * jm * gradijent$$

Stohastički gradijentni spust (engl. Stochastic gradient descent, SGD)

Ovaj algoritam se razlikuje po prirodi obrade od svih gorepomenutih algoritama. Odlikuje ga obrada parametara više puta u toku svake epohe koristeći serije podataka.

$$noviParametri = parametri - serijaGradijenta * stopaUčenja + momentum$$

Prilagodljivi linearni impuls (engl. Adaptive linear momentum, ADAM)

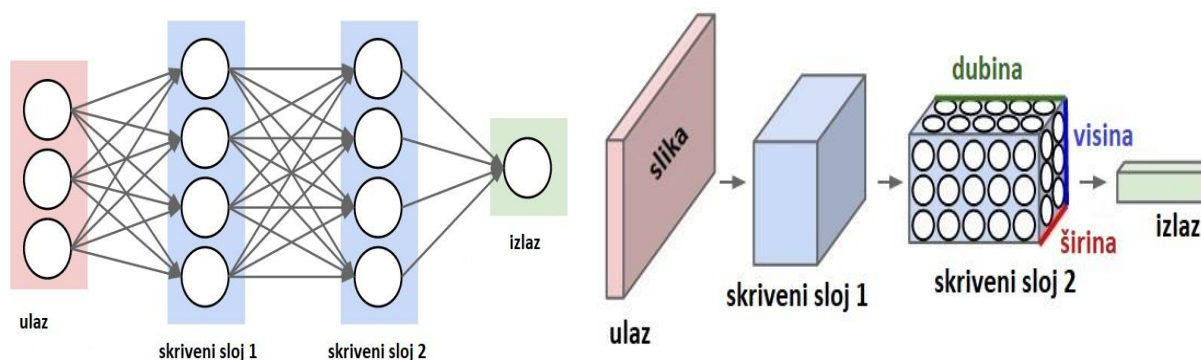
Algoritam je sličan gradijentnom spustu s tim da implementira sofisticiranije metode za određivanje pravca učenja. Ova složenost omogućava bržu konvergenciju u odnosu na gorespomenuti algoritam.

2.3 Podela neuronskih mreža

Od samog nastanka pa do danas, neuronske mreže se neprestano razvijaju i poboljšavaju tako da se mogu naći u puno različitih oblika. Svaka od njih ima neke svojstvene karakteristike koje omogućavaju da se bolje ili gore ponašaju u određenim okolnostima. Okolnosti mogu biti uslovljene fizičkim zahtevima, obliku podataka koji se obrađuje, kao i vrsti problema koji treba rešiti. Podela neuronskih mreža može se izvršiti u nekoliko kategorija: načinu učenja, toku informacija, dubini. Prema načinu učenja imamo tri vrste mreža: nenadgledano, nadgledano i polunadgledano učenje. Mreža koja ima nadgledano učenje u potpunosti zavisi od podataka koji joj se moraju obezbediti i obeležiti. Nenadgledano i polunadgledano učenje zahteva delimično ili uopšte ne zahteva pripremu podataka. Kod podele prema toku informacija delimo ih na nepovratne (engl. Feedforward) i na povratne, rekurentne (engl. Feed back). Kod nepovratnih se prenos signala vrši samo u jednom smeru, bez petlji što kod rekurentnih nije slučaj. Prema dubini razlikujemo jednoslojne i duboke neuronske mreže. Jednoslojne mreže imaju samo jedan skriveni sloj, dok se kod dubokih pojavljuju dva ili više skrivenih slojeva. Za potrebe ovog rada detaljnije će biti objašnjena konvoluciona neuronska mreža (engl. *Convolutional Neural Network*) jer će se upravo ovaj tip mreže koristiti za kreiranje modela binarne klasifikacije. Ovaj tip mreže spada u duboku, nepovratnu kategoriju mreže koja koristi nadgledani sistem učenja.

2.4 Konvolucione neuronske mreže

⁹Konvolucione neuronske mreže (engl. Convolutional neuron network, CNN) predstavljaju poseban tip neuronske mreže koja se takođe sastoji od neurona raspoređenih u više nivoa isto kao i klasična neuronska mreža. Specifičnost ove mreže ogleda se u tome da na ulazu prihvata dvodimenzionalne grafičke prezentacije koje mogu imati različite dubine. Dubina predstavlja broj kanala pomoću kojih je kreirana slika, najčešće 3 (crvena, zelena i plava). Kod klasičnih neuronskih mreža, ulaz je doveden na oblik jednodimenzionalnog vektora, što u slučaju slike može predstavljati jako veliku količinu podataka, a samim tim i veliki broj parametara koji treba da se optimizuju. Na primer, slika od $100 \times 100 \times 3$ (širina, visina i dubina respektivno) imaće $100 \times 100 \times 3 = 30\,000$ težinskih koeficijenata samo za jedan neuron u prvom nivou neuronske mreže. U praksi, rezolucija slike može biti još veća što dodatno povećava broj parametara, a samim tim i otežava proces učenja. Kod konvolucionih mreža broj parametara je znatno manji tako da omogućava bržu i efikasniju upotrebu. Osnovni princip funkcionisanja konvolucione mreže ogleda se u tome da, pomoću posebno dizajniranih slojeva u mreži, detektuje specifične regione na slici koji će dovesti do uspešne detekcije ili klasifikacije. Na slici 2.4.1 može se videti razlika arhikteture klasične (levo) i konvolucione neuronske mreže (desno).



Slika 2.4.1- Klasična i konvolucionna neuronska mreža [10]

Najopštija konvolucionna mreža sastoji se od sledećih slojeva: Konvolucionog sloja (engl. *Convolutional Layer*), sloja sažimanja (engl. *Pooling layer*), relu sloja (engl. *RELU layer*) i potpuno povezanog sloja (sloj klasične neuronske mreže). Iako se nalaze u mreži, neki slojevi ne zahtevaju treniranje parametara. Takvi slojevi su relu i sloj sažimanja. Sve računске operacije optimizacije rade se u okviru konvolucionog i potpuno povezanog sloja. Kombinacije slojeva raspoređeni su na više nivoa i najčešće se ponavljaju više puta.

Konvolucioni sloj je osnovni gradivni element konvolucione mreže i obavlja najbitniji i najzahtevnije računске operacije. Sloj se sastoji od niza filtera koji se mogu prilagoditi (istrenirati) odgovarajućim težinskim koeficijentima. Svaki filter predstavlja malu površinu (matricu) koja se primenjuje na ulaznu veličinu (sliku) po dužini i širini. Na primer, neka filter

prvog konvolucionog sloja ima veličinu $5 \times 5 \times 3$ (5 po širini/visini i 3 za dubinu koja obično ima tri kanala boja). Tokom svakog prolaza, filter se pomera po širini i visini i računa skalarni proizvod male površine ulaza i filtera za svaki pomeraj. Proizvod sa svake pozicije se čuva i obrazuje novu sliku koja se još zove i aktivaciona mapa. U njoj su sačuvani rezultati primenjenog filtera sa svake pozicije na kojoj je primenjen. Postupak učenja ogleda se u tome da se parametri filtera istreniraju tako da su sposobni da detektuju specifične regije, oblike ili površine određene boje koje će dovesti do ispravne detekcije/klasifikacije. Svaki konvolucionni sloj obično se sastoji od više filtera tako da će se za svaki filter izgenerisati aktivaciona mapa. Na primer, ukoliko sloj ima 17 filtera njegov izlaz imaće dubinu 17. Svaki od filtera tražiće neku određenu specifičnost koja se nalazi negde na ulazu. Pošto se radi o višedimenzionalnim ulazima kao što su slike ili izlazi drugih konvolucionih slojeva nije efikasno povezati neurone jednog sloja sa svim neuronima prethodnog sloja. Efikasnije rešenje je povezati svaki neuron samo sa određenom površinom ulaza. Ova površina je hiperparametar i predstavlja zapravo veličinu filtera (receptivno polje). Dubina filtera mora uvek odgovarati dubini ulaza. Filter se pomerajima dovodi na svaku poziciju ulaza (po širini i visini) i na svakoj poziciji primenjuje se u potpunoj dubini. Na primer, ako ulaz ima dimenzije $32 \times 32 \times 3$ (širina, visina i dubina) i polje recepcije filtera 5×5 , onda će svaki neuron imati $5 \times 5 \times 3 = 75$ težinskih koeficijenata i 1 korektivni faktor. Takođe u situaciji da imamo ulaz dimenzija $16 \times 16 \times 20$ (izlaz iz prethodnog konvolucionog sloja) i veličinu filtera 3×3 , tada će svaki neuron imati $3 \times 3 \times 20 = 180$ veza (težinskih koeficijenata) i 1 korektivni faktor. Ukupan broj neurona određuje se na osnovu veličine izlaza i broja filtera koji će se primeniti. Dimenzija izlaza određuje se na osnovu broja pozicija za koliko se vrši pomeranje filtera, kao i od opcije za popunjavanje ivica slike. U slučaju da je broj pomeraja jedan, tada se filter pomera za jednu poziciju (piksel) u svakom koraku. Pomeranje se vrši po horizontalnoj osi dok se ne dosegne desna ivica, a zatim se spušta za jednu poziciju niže (vertikalno) i ponavlja horizontalni prolaz. Postupak se ponavlja sve dok filter ne prođe kroz celu površinu i ne dosegne donju desnu ivicu. Najčešće vrednosti pomeraja su 1, 2 i 3, dok su se veći pomeraji pokazali nepraktičnim i nisu česti u praksi. Popunjavanje ivice slike vrši se kako bi se filter efikasnije primenio na krajnje ivice ulaza (slike). U tom slučaju dodaju se novi pikseli po horizontalnim i vertikalnim ivicama. Pomoćni pikseli obično se popunjavaju nulama kako ne bi uticali na operaciju filtera. Broj kolona i redova koji će se dodati definiše se kao vrednost hiperparametra. Kada su poznate vrednosti za broj pomeraja i popunjavanje ivica moguće je izračunati dimenzije izlaza. [10] Formule za izračunavanje date su u sledećem delu.

$$W_o = \frac{W_i - F_w + 2P}{S} + 1$$

$$H_o = \frac{H_i - F_h + 2P}{S} + 1$$

Gde je:

Wo- širina izlaza **Ho**- visina izlaza

Wi- širina ulaza **Hi**- visina ulaza

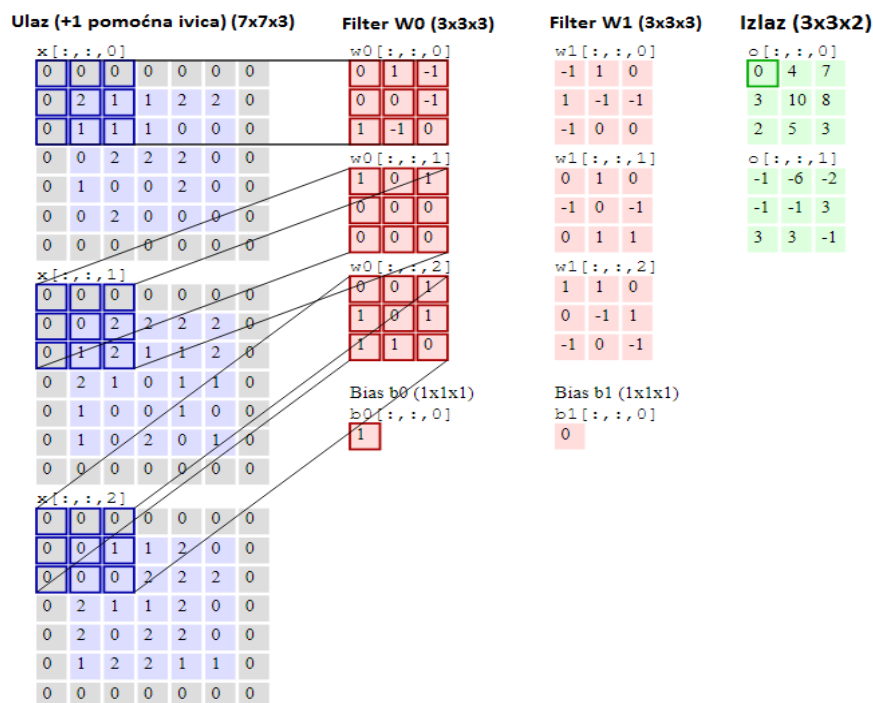
Fi- širina filtera **Fh**- visina filtera

P- broj pomoćnih ivica **S**- broj pomeraja

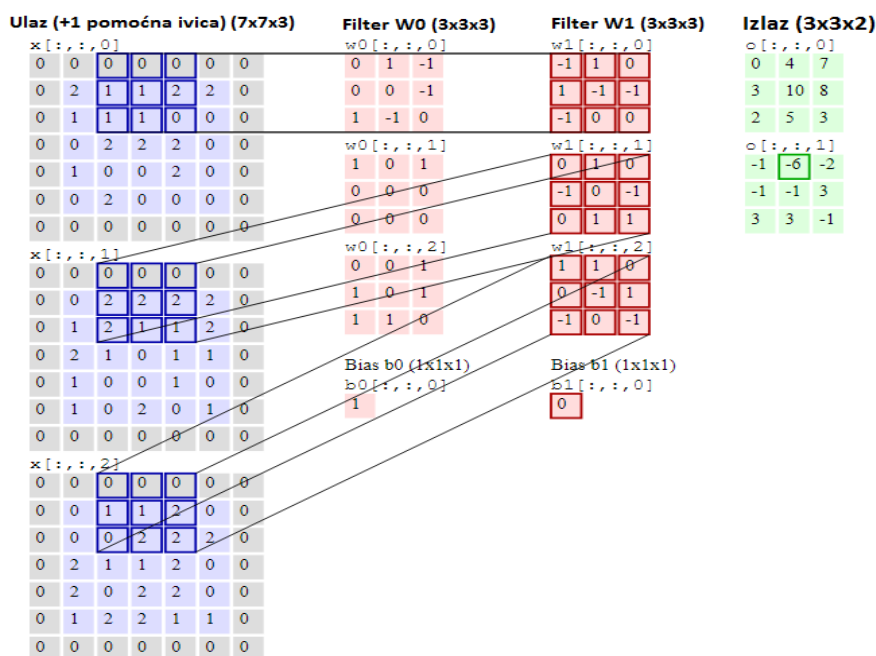
Ukoliko je cilj da se dimenzije ulaza i izlaza ne razlikuju i ako se pomeraj vrši za jednu poziciju onda broj pomoćnih ivica treba izračunati po sledećoj formuli[10]: $P=(F-1)/2$.

Na primer, ulaz je dimenzija 10x10 i filter veličine 2x2, pomeraj 2 i broj pomoćnih ivica 1. Dimenzije izlaza, izračunate po prethodnim formulama, su $W_o=(10-2+2*1)/2+1=6$, $H_o=(10-2+2*1)/2+1=6$. U ovom primeru nije definisana dubina izlaza tako da nema treće dimenzije. Dubina izlaza određena je brojem filtera koji će se primeniti na ulaz. Ukoliko se primeru doda korišćenje dva različita filtera onda bi potpuna dimenzija izlaza bila 6x6x2.

Prilikom biranja vrednosti za broj pomeraja treba voditi računa da li ta vrednost vodi do nelogične dimenzije izlaza. Na primer, ako je dimenzija ulaza 11x11 , veličina filtera 2x2, broj pomeraja 2 i broj pomoćnih ivica 0. Tada bi rezultujuća dimenzija izlaza imala vrednost 5.5x5.5. Kako su širina i visina predstavljene pomoću decimalnih brojeva, jasno je da se ova vrednost pomeraja ne može koristiti. U ovakvim slučajevima najčešće se broj pomoćnih ivica adaptira kako bi rezultujuća dimenzija izlaza bila validna, a da se pritom ne menjaju ostali parametri. Svakako je najbolje izvršiti proračun i videti da li vrednosti parametara dovode do neželjene situacije koja može uticati na rad i funkcionisanje čitave mreže. Na slikama 2.4.2 i 2.4.3 ilustrovan je postupak obrade slike u konvolucionom sloju gde je ulaz dimenzija 5x5x3, filter 3x3x3, broj pomeraja 2, broj pomoćnih ivica 1, broj filtera 2.



Slika 2.4.2- Računanje elementa izlaza konvolucione neuronske mreže [10]



Slika 2.4.3- Računanje elementa izlaza konvolucione neuronske mreže [10]

Na slici 2.4.2 opisan je početni korak konvolucije sa prvim filterom. Dubina ulaza je 3 što se može posmatrati i kao broj kanala boja, ako je prvi konvolucionni sloj u pitanju. Zbog toga dubina

svakog filtera mora biti isto 3. Krajnji izlaz za jednu regiju dobijamo kada sumiramo sve vektorske proizvode svakog kanala receptivnog polja i filtera. Na ukupnu sumu dodaje se i vrednost korektivnog faktora. Treba napomenuti da su u ilustraciji korišćene male vrednosti zbog lakšeg računanja. Takođe, filteri imaju već podešene težinske faktore tako da je ovde prikazan prolaz unapred. U formuli 2.4.1 prikazan je obrazac računanja rezultata prvog elementa izlaza (slika 2.3.1).

$$o[0,0,0] = \left(\sum_{i=0}^2 \sum_{j=0}^2 x[i,j,0] * w0[i,j,0] + \sum_{i=0}^2 \sum_{j=0}^2 x[i,j,1] * w0[i,j,1] + \sum_{i=0}^2 \sum_{j=0}^2 x[i,j,2] * w0[i,j,2] \right) + b0$$

Formula 2.4.1- Obrazac za izračunavanje elementa izlaza konvolucionog sloja

Na slici 2.4.3 ilustrovan je postupak računanja drugog elementa izlaza drugog kanala. Računanje vrednosti se vrši na isti način s tim da se u tom slučaju koriste težinski koeficijenti drugog ($w1$) filtera.

U zavisnosti od situacije, ponekad je dobra praksa da se ubaci sloj sažimanja posle konvolucionog sloja. Sloj sažimanja smanjuje izlaz prethodnog sloja na manju veličinu što direktno smanjuje i broj parametra za treniranje u sledećem nivou. Osim smanjivanja dimenzija, sloj doprinosi većoj generalizaciji što ponekad nije poželjno, pogotovo kada se radi o ulazima malih dimenzija pa smanjivanje dovodi do gubitka bitnih informacija. U takvim slučajevima sloj sažimanja se izostavlja. Slično konvolucionom sloju i ovde se obrada vrši sekvencijalno po malim površinama sve dok se ne pokrije cela površina ulaza. Razlika u ovom sloju je u tome što se umesto vektorskog množenja ovde vrši funkcija maksimizacije. Operacija je sasvim jednostavna i ogleda se u tome da se iz date regije (receptivnog polja) uzima samo polje (piksel) sa najvećom vrednošću. Ta vrednost se dalje upisuje na pripadajućoj poziciji izlaza. Pomeraj i veličina regije takođe mogu biti različiti, a dimenzija izlaza dobija se po sledećoj formuli:

$$W_o = \frac{(W_i - F_w)}{S} + 1$$

$$H_o = \frac{H_i - F_h}{S} + 1$$

Gde je:

W_o- širina izlaza

H_o- visina izlaza

W_i- širina ulaza

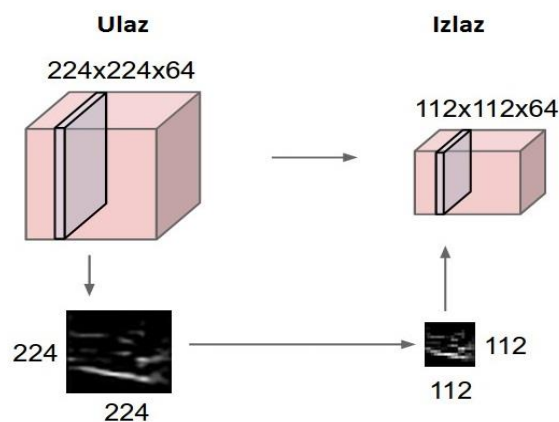
H_i- visina ulaza

F_i- širina filtera

F_h- visina filtera

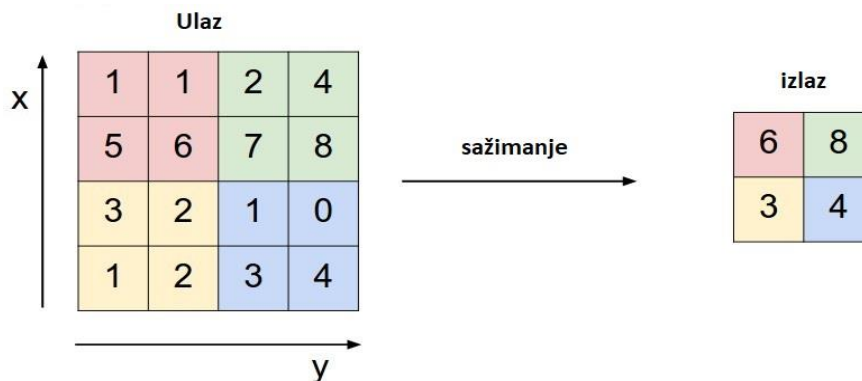
S- broj pomeraja

U ovom sloju vrši se isključivo redukcija prve dve veličine (širina, visina) dok dubina ostaje nepromenjena. Iz ovog razloga dimenzija regije po kojoj će se vršiti obrada imaće samo dve veličine. Uobičajene veličine koje se koriste za parametre i koje u praksi daju dobre rezultate su: veličina regije 3x3 sa pomerajem 2 i regija veličine 2x2 sa pomerajem 2. Slojevi sažimanja sa većim receptivnim poljem obično dovode do gubitka bitnih informacija. Na slici 2.4.4 prikazana je arhitektura jednog sloja sažimanja.



Slika 2.4.4- Sloj sažimanja [10]

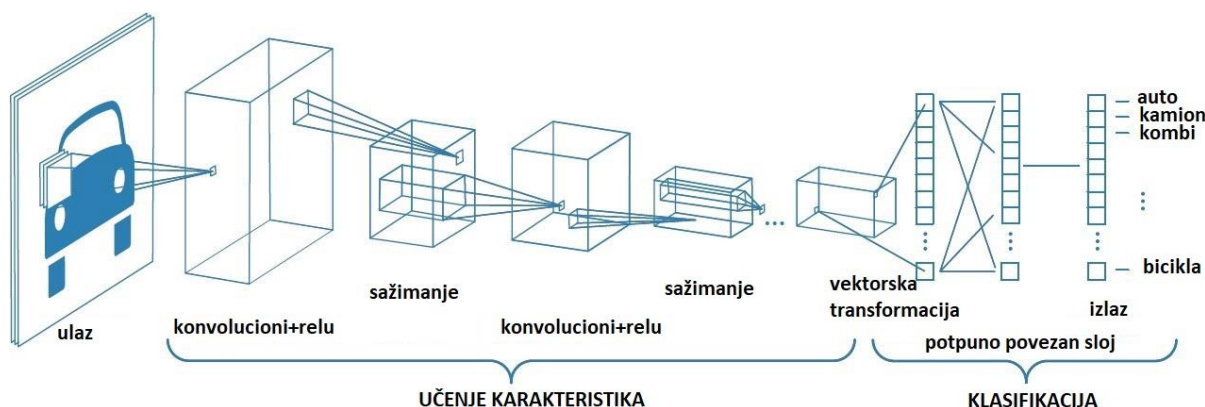
Prilikom obrade receptivnog polja osim funkcije maksimizacije mogu se koristiti i druge funkcije kao što su funkcija srednje vrednosti ili L2 normalizacija. Funkcija srednje vrednosti računa srednju vrednost svih piksela receptivnog polja i to daje kao izlazni rezultat. Ova funkcija ne nalazi preteranu primenu u praksi jer se funkcija maksimiziranja pokazala kao efikasnija. Na slici 2.4.5 prikazana je obrada jednog ulaza dimenzija 4x4 i receptivnog polja 2x2.



Slika 2.4.5- Ilustracija procesa sažimanja [10]

Relu (engl. Rectified Linear Unit, ReLU) sloj je takođe sloj u kome nema trenirajućih parametara, ali za razliku od sloja sažimanja, on ne smanjuje dimenzije ulaza. Osnovna uloga ovog sloja je da izvrši obradu pojedinačnih elemenata ulaza i da na njima primeni ReLU funkciju. Ova funkcija je opisana u prvom poglavlju u tabeli 2.1. Funkcija zapravo daje izlaz nula ako je ulazna vrednost negativna, u suprotnom vraća ulaznu vrednost ($\max(0, x)$).

Potpuno povezan sloj (engl. Fully Connected layer, FC) ima isti oblik i funkcionalnost kao sloj u običnoj neuronskoj mreži. Njegov zadatak je da na osnovu informacija i saznanja koja su dobijena pomoću slojeva konvolucije, relu i sažimanja izračuna/definiše verovatnoću detekcije objekta na slici ili raspodelu verovatnoća klasa ukoliko se radi o klasifikacionom modelu. Ovaj sloj se nalazi na kraju mreže i može imati više ponavljanja. Ulaz u prvi potpuno povezan sloj je izlaz iz poslednjeg konvolucionog dela pa je potrebno prevesti ulaz u vektorski oblik. Na slici 2.4.6 dat je primer klasifikacionog modela koji se sastoji od kombinacije više konvolucionih, relu, slojeva sažimanja i jednog potpuno povezanog sloja za generisanje izlaza.



Slika 2.4.6- Ilustracija konvolucione neuronske mreže [11]

Iako se u osnovi sve konvolucione mreže sastoje od prethodnonavedenih slojeva, moguće je izvesti mnoštvo različitih varijacija u arhitekturi modela od broja i sastava slojeva do različitih vrednosti parametara koji će se koristiti u svakom od njih (veličina receptivnog polja, dubina, pomeraj,..). Različite vrednosti i raspored rezultovaće boljim ili lošijim perfomansama. U sledećoj listi navedene su neke od arhitektura konvolucionih mreža koje su dale zapažene rezultate: ¹²LeNet, ¹³AlexNet, ZF Net, GoogLeNet, VGG Net, ¹⁴ResNet.

2.5 Metodologija mapiranja podataka u dvodimenzionalnu grafičku prezentaciju

Metod mapiranja koji će se koristiti u ovom radu bazira se na korišćenju podataka koji su dovedeni na oblik u kojima su sva obeležja predstavljena pomoću kategorija/grupa. Dimenzija rezultujuće slike mora da bude identična za svaki zapis u skupu podataka. Zbog ovog uslova,

širina slike određuje se brojem obeležja koja se koriste, dok je visina određena najvećim brojem različitih kategorija koje poseduje neko obeležje. Slika koja se generiše biće u crno-belom varijanti tako da je dubina 1. Ideja je da se u prvom koraku kreira slika datih dimenzija i da svaki piksel ima vrednost 0. Nakon toga sledi postupak mapiranja piksela sa vrednošću 255. Ovi pikseli signaliziraju pripadnost podatka određenom obeležju i grupi. Svaka kolona označava jedno obeležje, dok svaki red označava pripadnost jednoj grupi. Piksel oznake postavlja se u preseku obeležja i grupe kojoj vrednost obeležja pripada. Svaka kolona slike mora imati jedan piksel sa vrednošću 255 jer vrednost obeležja ne može imati pripadnost više grupa. Ciljno obeležje (A15) izuzima se iz ove obrade jer je to informacija koju model treba da predvidi. Vrednost ovog obeležja prosleđuje se modelu u procesu učenja kao očekivani izlaz. Sa ovom informacijom model može da izračuna funkciju gubitka, a samim tim i da optimizuje parametre u odgovarajućem pravcu kako bi se utvrdila veza između različitih šablona, koje obrazuju pikseli slike, i izlazne vrednosti.

3. Priprema i analiza podataka

Pre samog razvijanja modela, treba izvršiti pravilnu pripremu i analizu podataka koji će se koristiti za treniranje i testiranje modela. Ovaj korak je možda bitniji i od samo razvijanja modela jer ukoliko podaci nisu ispravni ili nisu dovedeni u poželjan oblik ni rezultati ne mogu biti relevantni. Pre svega potrebno je proveriti da li u prikupljenim podacima postoje zapisi koji nisu poželjni. Takvim podacima obično nedostaje neki podatak koji može dovesti do anomalija ili onemogućavanja procesa učenja. U zapisima gde nedostaje podatak nekog obeležja (kolone), dodeljuje se vrednost (srednja vrednost na primer) ili se briše. Brisanje se preporučuje ukoliko u skupu podataka nema puno ovakvih zapisa. Takođe, ukoliko je nekom obeležju dodeljena izuzetno velika ili mala vrednost, koja očigledno odstupa od ostalih zapisa, treba proveriti da li je možda napravljena greška kako ti zapisi ne bi doveli do šuma ili greške prilikom treniranja. U zavisnosti od tipa i metode učenja, nekada je potrebno izvršiti i pretvaranje simboličkih vrednosti u numeričke ekvivalente. Ovo je veoma bitno jer ukoliko se ovaj korak ne izvrši greška prilikom obrade je neminovna. Sa druge, strane kod numeričkih podataka nekada je poželjno izvršiti normalizaciju ili standardizaciju kako bi se opseg vrednosti doveo u manji obim. Izbor da li će se i koja obrada izvršiti (normalizacija ili standardizacija) zavisi od algoritma koji će se koristiti u procesu učenja. Duplirani zapisi, takođe, nisu poželjni i neophodno ih je pronaći i izbaciti kako ne bi došlo do neželjenog preprilagođavanja modela¹⁵.

Nakon ovih provera i korekcija prelazi se na odabir obeležja koje sadrže bitne informacije za problem koji se rešava. Dobar odabir direktno će uticati na smanjenje preprilagođavanja modela, ubrzaće se proces treniranja i povećaćće se tačnost predikcije. Neke kolone se mogu odmah izostaviti jer se bez ikakvih analiza može zaključiti da ne mogu doprineti pravilnom radu modela. Primer takvih kolona su :broj telefona, id, broj računa, broj ulice itd. U praksi se generalno ne može preterano oslanjati na zdravorazumsko rasuđivanje ukoliko se ne zna na šta se podaci obeležja odnose pa je potrebno sprovesti detaljnije analize i metode koje će dovesti do pravilnog izbora obeležja. Prema načinu biranja podesnih kolona, metode možemo podeliti u tri kategorije¹⁶.

- Metode filtriranja (engl. Filter methods)
- Metode testiranja (engl. Wrapper methods)
- Kombinovane metode (engl. Embedded methods)

3.1 Metode filtriranja

¹⁷Metode filtriranja koriste se kao pripremni korak što znači da se vrši odabir kolona pre kreiranja modela. Ove metode odlikuje statistička obrada korelacija između obeležja i cilja predikcije. Statističke metode (testovi) računaju u kojoj meri cilj predikcije zavisi od vrednosti nekog

obeležja. Rezultati testova predstavljeni su u numeričkom obliku pa je lako izvršiti rangiranje najboljih obeležja. Izbor statističke analize koja će se primeniti zavisi od tipa obeležja i ciljne vrednosti. U osnovi obeležje i cilj mogu biti neprebrojive (kontinualne) vrednosti (plata, visina, cena,...) ili prebrojive (kategoričke) (pol, zaposlenost, količina...). Metode filtriranja su dosta brže od metode testiranja i lake su za izračunavanje. U situacijama kada je broj obeležja u podacima veliki (nekoliko hiljada) onda su metode filtriranja jedino moguće rešenje. U sledećem delu navedeni su neki od najkorišćenijih statističkih testova koji se koriste u praksi.

Informaciona dobit (engl. Information gain)

Informaciona dobit računa smanjenje entropije ukoliko se obeležju dodeli neka slučajna vrednost iz skupa mogućih vrednosti. U teoriji informacija, entropija slučajne promenljive je prosečni nivo „informacija“, „iznenađenja“ ili „neizvesnosti“ svojstvenih mogućim ishodima promenljive. S obzirom na diskretnu slučajnu promenljivu, sa mogućim ishodima, koji se javljaju sa verovatnoćom, entropija je formalno definisana kao:

$$Xx_1, \dots, x_n P(x_1), \dots, P(x_n), X$$

$$H(X) = - \sum_{i=1}^n P(x_i) * \log P(x_i)$$

gde **n** označava zbir mogućih vrednosti promenljive. Entropija meri koliko informacija ima u slučajnoj promenljivoj ili tačnije njenu distribuciju verovatnoće. Iskrivljena distribucija ima nisku entropiju, dok raspodela gde događaji imaju jednaku verovatnoću ima veću entropiju. Informaciona dobit omogućava da se iskoristi entropija za izračunavanje kako promena obeležja utiče na čistoću skupa podataka, npr. raspodela klasa. Manja entropija sugerise više čistoće ili manje iznenađenja. Formula izračunavanja ima sledeći oblik:

$$IG(S, a) = H(S) - H(S \vee a)$$

Gde je **IG(S, a)** informacioni dobitak nad skupom **S** sa slučajnim promenljivama **a**. **H(S)** predstavlja entropiju skupa **S** bez promena, a **H(S | a)** entropiju sa izvršenim promenama promenljive (obeležja) u skupu **S**. Skup **S** predstavlja jedan uređen par obeležje-cilj. Promene se vrše isključivo u obeležju, a postupak se ponavlja za svako obeležje iz ukupnog skupa podataka. Ovu analizu moguće je primeniti ako su i obeležje i cilj prebrojivog tipa. Informaciona dobit se još naziva i međusobno informisanje (engl. Mutual information) kada se koristi za procenu korisnosti obeležja nekog skupa podataka.

Hi-kvadratni test (engl. Chi-square test)

Hi kvadratni test se takođe koristi ukoliko je obeležje i cilj prebrojivog tipa. Test pokazuje zavisnost dve promenljive pa je postupak potrebno ponoviti za svaki par obeležje- cilj. Što je

rezultat veći to je veća zavisnost ciljne promenljive i obeležja što je indikacija da to obeležje treba zadržati. U standardnim primenama ovog testa, zapažanja su svrstana u međusobno isključive klase i postoji neka teorija, ili nulta hipoteza, koja daje verovatnoću da bilo koje opažanje pripada odgovarajućoj klasi. Svrha testa je da se proceni koliko su verovatne opservacije, pod pretpostavkom da je nulta hipoteza tačna.

Koeficijent korelacije (engl. Correlation Coefficient)

Koeficijent korelacije nam daje informacije koliko promena jednog obeležja utiče na promenu drugog. Ukoliko između obeležja postoji visoka korelacija možda bi trebalo razmotriti izbacivanje jednog od njih kako bi smanjili broj podataka koji ne doprinose novim saznanjima. Treba naglasiti da je poželjna visoka korelacija između obeležja i cilja, a da međusobna korelacija između obeležja treba biti što manja. U praksi se obično koristi Pirsonov koeficijent korelacije. Pirsonov koeficijent korelacije računa korelaciju između dve neprebrojive (*kontinualne*) promenljive sa intervalne i razmerne skale. Obično se predstavlja malim slovom r a formula za izračunavanje ima sledeći oblik:

$$r = \frac{\sum xy}{NS_x S_y}$$

gde je x i y odstupanja rezultata od aritmetičkih sredina promenljivih X i Y , N - broj uzoraka, S_x - standardna greška srednje vrednosti promenljive X , S_y - standardna greška srednje vrednosti promenljive Y .

Rezultat se kreće u rasponu od -1 do 1. Vrednost 0 pokazuje da ne postoji međusobni uticaj između dve promenljive. Vrednosti -1 i 1 predstavljaju potpunu negativni i potpunu pozitivnu korelaciju (respektivno). Negativna vrednost korelacije takođe ukazuje na zavisnost između promenljivih tako da je idealno da korelacija između obeležja bude približno nuli.

3.2 Metode testiranja

¹⁸Metode testiranja ogleda se u biranju različitih podskupova celog skupa obeležja i odabirom onog koji daje najbolje rezultate. Za potrebe ovog pristupa neophodno je postojanje modela kako bi se vršila evaluacija posle svakog koraka. U svakom koraku model mora da se istrenira i testira pomoću trenutnog podskupa. Ovo je izuzetno spor postupak pogotovo kada je broj obeležja veliki. Sa druge strane, ovakav pristup daje bolje rezultate u odnosu na metode tesiranja jer se testira jako veliki broj različitih kombinacija. U sledećem delu dati su različiti pristupi u biranju različitih podskupova obeležja.

Iterativno dodavanje obeležja (engl. Forward feature selection)

Ovaj pristup se ogleda u tome da je polazna tačka jedno obeležje. Odabir tog obeležja donosi se na osnovu upoređivanja rezultata koje daje svako od njih i biranje najboljeg. U sledećem koraku meri koliko dodavanje novog obeležja iz skupa preostalih utiče na poboljšanje performansi. Postupak se ponavlja dok se ne dosegne neki željeni kriterijum (veličina podskupa, maksimalni broj iteracija) ili dok se ne isprobaju sve moguće permutacije što je u praksi redak slučaj. Za potpunu pretragu potrebno je $N!$ (faktoriyel) iteracija gde je N ukupan broj obeležja. Tako je na primer za skup od 12 obeležja potrebno 479001600 iteracija.

Iterativno izbacivanje obeležja (engl. Backward feature elimination)

Postupak je potpuno obrnut od onog koji je opisan u prethodnom pristupu. U ovom slučaju polazna tačka je skup od svih obeležja. U svakoj iteraciji meri se koliko bi se rezultat poboljšao ukoliko bi neko obeležje bilo izbačeno. Postupak se ponavlja sve dok se dosegne željeni kriterijum ili dok performanse ne krenu da opadaju.

Rekurzivno izbacivanje obeležja (engl. Recursive feature elimination, RFE)

Ovakav pristup pre svega zahteva korišćenje nekog eksternog estimatora koeficijenta uticaja obeležja (npr. koeficijenti linearnog modela). Postupak se ogleda u tome da se rekurzivnom obradom bira sve manji skup obeležja. Inicijalno svako obeležje dobija svoj koeficijent uticaja (korisnosti) od estimatora. Nakon toga, iz skupa se izbacuje obeležje sa najmanjim uticajem i ponovo se računa koeficijenti za novi podskup. Postupak se ponavlja sve dok se ne dostigne željeni broj obeležja.

3.3 Kombinovane metode

¹⁹Kombinovane metode koriste prednosti metoda i filtriranja i testiranja. Cilj je optimizovanje procesa biranja obeležja tako da se kombinacijom statistike i iterativnog biranja nađe kompromis između konačnog rezultata i troškova obrade. Primer ovakvog pristupa je metod Random Forest Importance. Algoritam slučajne šume (engl. Random Forest) je vrsta ensambl algoritma koji se sastoji od proizvoljnog broja stabla odlučivanja (engl. Decision trees). Slučajni podskupovi obeležja dodeljuju se svakom stablu u šumi. Svako stablo poseduje čvorove koji su organizovani na osnovu svoje čistoće (engl. impurity). Čvorovi sa najvećim smanjenjem čistoće postavljaju se za prve čvorove u stablu dok se za poslednje postavljaju čvorovi sa najmanjim smanjenjem čistoće. Na osnovu ovoga može se izvršiti statistička analiza svakog stabla u šumi i da se izvede zaključak koja obeležja imaju veći uticaj od ostalih.

Koeficijent opravdanosti (engl. Weight of Evidence, WoE)

Zbog specifičnosti metodologije koja se obrađuje u ovom radu potrebno je prebaciti sve neprebrojive vrste u prebrojivu kategoriju. Postupak grupisanja se odvija tako što određeni opseg vrednosti dobija svoju grupu. Na primer obeležje visina, koja je verovatno neprebrojiva, imaće sledeće grupe: nizak (<160), prosečna visina[160-190] i visok (>190). Kako se radi o binarnoj klasifikaciji, iskoristiće se koeficijent opravdanosti²⁰ koji će oceniti koliko dobro je izvršena raspodela zapisa za svaku grupu. Nakon toga je moguće izračunati informacionu vrednost (engl Information value, IV) koja govori o prediktivnoj moći preuređenog obeležja. Koeficijent opravdanosti se računa za svaku novokreiranu grupu kao prirodni logaritam količnika distribucije pozitivnih i distribucije negativnih rezultata date grupe. Distribucija se računa kao procenat pozitivnih/negativnih rezultata u grupi u odnosu na ukupan broj pozitivnih tj. negativnih rezultata (respektivno).

$$WoE = \ln\left(\frac{Dp}{Dn}\right) \quad Dp = \frac{Bp}{Np} \quad Dn = \frac{Bn}{Nn}$$

Bp- broj pozitivnih rezultata koji pripadaju istoj grupi

Np- ukupan broj pozitivnih uzoraka

Bn- broj negativnih rezultata koji pripadaju istoj grupi

Nn- ukupan broj negativnih uzoraka

Pozitivna vrednost koeficijenta govori da je distribucija pozitivnih rezultata veća od distribucije negativnih, a negativna govori da je distribucija negativnih veća od distribucije pozitivnih. Nakon prebacivanja neprebrojivih vrednosti u grupe treba izračunati koeficijent opravdanosti za svaku od novodobijenih grupa. Ukoliko postoje grupe sa sličnim vrednostima onda te grupe treba spojiti. Razlog tome je što imaju sličnu proporciju distribucija (pozitivnih i negativnih) pa će grupe imati slično ponašanje. Da bi koeficijent opravdanosti bio primenljiv mora se ispuniti uslov da svaka grupa sadrži najmanje 5% ukupnih uzoraka i da nema grupe u kojoj se ne nalazi bar jedan pozitivan ili negativan uzorak.

Kada je izvršena kategorizacija i izračunat koeficijent opravdanosti može se izračunati i informaciona vrednost. Ova vrednost može pomoći prilikom odabira korisnih obeležja. Ovo nije optimalan način kada se radi o klasifikaciji sa više klasa ali u slučaju binarne klasifikacije daje dobre rezultate. Vrednost se dobija tako što se razlika između distribucije pozitivnih i negativnih uzoraka grupe množi sa pripadajućim koeficijentom opravdanosti. Sumiranjem vrednosti svih kategorija dobija se ukupna informaciona vrednost za dato obeležje. Formula izračunavanja:

$$IV = \sum (Dp - Dn) * WoE$$

Dobijene vrednosti govore kolike se prediktivne moći kriju u obeležju. Interpretacija vrednosti vrši se po kriterijumima koji su navedeni u tabeli 3.1.

Table 3.1- Kriterijum ivformacione vrednosti

Informaciona vrednost	Prediktivnost
<0.02	Nije korisna za predikciju
0.02- 0.1	Slaba prediktivna sposobnost
0.1-0.3	Solidna prediktivna sposobnost
0.3- 0.5	Jaka prediktivna sposobnost
>0.5	Sumnjiva prediktivna sposobnost

Za vrednosti veće od 0.5 treba proveriti da li je kategorizacija pravilno izvršena jer veće vrednost obično ukazuju na takve nedoslednosti. Sa povećanjem broja kategorija povećava se i informaciona vrednost mada treba biti obazriv da u grupama ostane dovoljan broj pozitivnih i negativnih uzoraka jer u suprotnom može dovesti do nemerodavne prediktivne sposobnosti obeležja.

4. Praktična realizacija

4.1 Programski jezik

Praktična realizacija ovog rada odrađena je u Python programskom jeziku. **Python** kao programski jezik, nastao je početkom devedesetih godina 20-og veka u National Research Institute for Mathematics and Computer Science u Holandiji od strane autora Guido van Rossum. Nastao je kao kombinacija više jezika, kao što su ABC, Modula-3, C, C++, Algol-68, SmallTalk, Unix shell. Odlikuje ga laka sintaksa sa relativno malim brojem rezervisanih reči, velika brzina i strog način pisanja koda što se odražava na urednost koda. Spada u grupu viših, interpreterskih, interaktivnih, objektno orijentisanih jezika a sadrži i određene koncepte iz funkcionalnog programiranja. Ima široku primenu u obrazovnim ustanovama, web i klasičnom programiranju ali i u oblasti veštačke inteligencije. Veoma je popularan među stručnjacima mašinskog učenja i analize podataka tako da su razvijene korisne biblioteke koje mogu pomoći pri radu u ovim oblastima. U sledećem delu navedene su neke od biblioteka koje su korišćene u ovom radu.

TensorFlow je besplatna biblioteka otvorenog koda namenjena za razvijanje sistema veštačke inteligencije i mašinskog učenja. Primena biblioteke ima širok spektar ali fokus primene usmeren je na razvijanju i upravljanje procesa kreiranja i treniranja dubokih neuronskih mreža. Biblioteka se razvija pod okriljem Google korporacije za potrebe istraživanja i produkcije. Dostupna je u različitim verzijama za različite programske jezike (Java, C++, Javascript), ali prevashodno za primenu u Pajthon jeziku. Može se izvršavati na različitim operativnim sistemima kao što su: Windows, macOS, Linux, iOS i Android. Omogućava istovremeno izvršavanje na više centralnih procesorskih jedinica (CPU) ili grafičkih procesorskih jedinica (GPU). Mogućnost izvršenja pomoću GPU-a je posebno bitna za ovaj rad jer se koriste konvolucione mreže koje obrađuju grafičke prikaze gde GPU ostvaruje veće brzine od klasičnih CPU-a.

Keras je biblioteka višeg nivoa razvijena u Python jeziku koja koristi alate iz TensorFlow ili Theano biblioteke. Ideja koja stoji iza ove biblioteke da se postupak razvijanja i treniranje modela neuronske mreže dodatno ubrza i poboljša. Centralni je deo TensorFlow2 ekosistema sa kojim je uskopovezana. Pokriva svaki korak obuke modela od upravljanja podacima do optimizacije hiperparametara sistema.

NumPy je osnovni paket alata za nepredne računске operacije u Pythonu. Ova biblioteka obezbeđuje rad sa višedimenzionalnim nizovima objekata, matricama, kao i pun asortiman za efikasne i brze operacije nad nizovima kao što su matematičke i logičke operacije, manipulacije

dimenzionalnosti, sortiranje, selekcija, statistička ispitivanja, diskretne Furijeove transformacije, slučajne simulacije itd.

Pandas je biblioteka otvorenog koda sa BSD-licenciranjem koja nudi alate za efikasno i brzo upravljanje strukturama skupova podataka. Alati olakšavaju posao učitavanja, prikaza i pripreme podataka. Osim toga u biblioteci se nalaze i alati namenjeni za analizu obrađenih podataka.

Matplotlib je sveobuhvatna biblioteka za kreiranje statičkih, animiranih i interaktivnih vizuelizacija u Pajtonu. Biblioteka nudi alate koji olakšavaju poslove kao što su prikaz strukture određenih podataka, dijagrama, histograma, 3D objekata, slika itd.

Seaborn je biblioteka višeg nivoa koja koristi alate Matplot biblioteke. Fokus ove biblioteke je vizuelizacija složenih statističkih podataka.

4.2 Radno okruženje

Collaboratory ili skraćeno Colab je proizvod Google Research odeljenja Google korporacije koji nudi korisnicima besplatno pisanje i izvršenje Python koda. Pisanje i izvršenje koda vrši preko internet pretraživača (engl. Browser) bez potrebe za instaliranjem dodatnih aplikacija ili alata. Okruženje je posebno prilagođeno razvijanju modela veštačke inteligencije, analizi i vizuelizaciji podataka. U tehničkom pogledu, Colab je klad (engl. Cloud) platforma na kojoj se izvršava Jupyter servis. Jupyter je web-bazirano okruženje koje nudi mogućnost obrade skripti/koda i podataka u mnoštvu različitih jezika uključujući Python, R, Julia i Scala. Njegov fleksibilan dizajn omogućava korisnicima da konfigurišu i organizuju radne zadatke u pojedinačnim nezavisnim jedinicama. Ovakav pristup pogodan je prilikom rada u računarskom novinarstvu, analizi podataka i mašinskom učenju. Colab osim ove usluge nudi i mogućnost korišćenja serverskih fizičkih resursa kao što su CPU i GPU. Jedini je uslov za korišćenje ovih usluga je posedovanje Google naloga čije je otvaranje takođe besplatno. Iz ovih razloga kao i zbog modularnosti, ovo okruženje je korišćeno prilikom praktične realizacije.

4.3 Eksperiment

Prvi korak praktične realizacije odnosi se na pripremu i analizu podataka. Potrebno je proveriti da li u zapisima postoje takvi da im nedostaje neka vrednost obeležja (prazno polje/null). Nakon toga treba proveriti postojanje dupliranih zapisa. Duplirani podaci dovode do preprilagođenja modela pa ih treba ukloniti. Takođe, treba izvršiti i proveru da li postoje zapisi sa ekstremnim odstupanjima u nekom od obeležja. Ova odstupanja nastala su verovatno greškom pri unosu pa ih je potrebno zameniti prosečnom vrednošću ili izbrisati iz skupa.

Podaci koji će se koristiti u ovom eksperimentu prikupljeni su od aplikanata kreditnih kartica koji su se pokazali kao pouzdani ili nepouzdani klijenti. Radi zaštite proverljivosti podataka, imena obeležja i vrednosti transformisani su u beznačajne simbole. Prikupljeni podaci su pogodni eksperimentu jer sadrže dobru pomešanost neprebrojivih i prebrojivih obeležja. U skupu se nalazi ukupno 690 zapisa koji su opisani pomoću 14 obeležja i jednog koji se odnosi na ciljno obeležje. Šest obeležja je neprebrojivog tipa dok je preostalih 8 prebrojivo. Vrednosti neprebrojivih obeležja promenjene su u numeričke vrednosti tako da obeležje koje je imalo vrednosti „p“, „g“, „gg“ sada ima vrednosti 1, 2, 3 (respektivno). U skupu se nalaze i 37 zapisa (5%) kojima nedostaje vrednost jednog ili više obeležja. Takva obeležja dobila su najčešću kategoriju kada su pitanju prebrojivi tipovi dok je neprebrojivima dodeljena prosečna vrednost. Na ovaj način izbegnuto je brisanje zapisa sa izostalim vrednostima. U tabeli 4.3.1 prikazana su imena obeležja sa njihovim osobinama.

Tabela 4.3.1- Pregled obeležja

Naziv obeležja	Tip	Vrednosti
A1	prebrojiv	0, 1
A2	neprebrojiv	[13.75- 80.25]
A3	neprebrojiv	[0- 28.0]
A4	prebrojiv	1, 2, 3
A5	prebrojiv	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14
A6	prebrojiv	1, 2, 3, 4, 5, 6, 7, 8, 9
A7	neprebrojiv	[0- 28.5]
A8	prebrojiv	0, 1
A9	prebrojiv	0, 1
A10	neprebrojiv	[0- 67]
A11	prebrojiv	0, 1
A12	prebrojiv	1, 2, 3
A13	neprebrojiv	[0- 2000]
A14	neprebrojiv	[1- 100001]
A15	prebrojiv	0, 1

Klijent može pripadati jednoj od dve moguće klase. Obeležje cilja(klase) nazvano je A15 i ima dve moguće vrednosti (0,1). Ove vrednosti mogu se posmatrati kao pouzdan i nepouzdan klijent

mada se, zbog zaštite poverljivosti, ne zna šta koja vrednost predstavlja. Distribucija klasa data je u sledećoj proporciji: 307 uzoraka (44.5%) prve klase (vrednost 1) i 383 uzoraka (55.5%) druge klase (vrednost 0). U tabeli 4.3.2 prikazani su primeri zapisa i skupa.

Tabela 4.3.2- Primeri zapisa iz skupa podataka

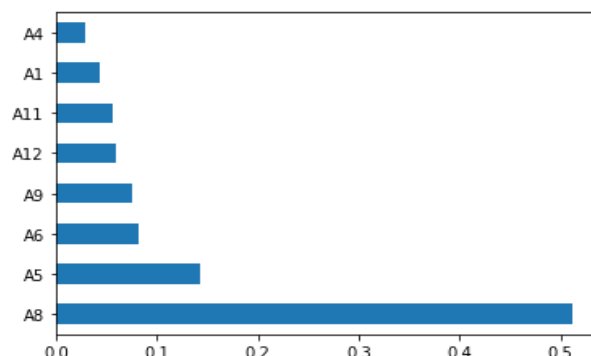
A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11	A12	A13	A14	A15
1	22.08	11.46	2	4	4	1.585	0	0	0	1	2	100	1213	0
0	22.67	7.0	2	8	4	0.165	0	0	0	0	2	160	1	0
0	29.58	1.75	1	4	4	1.25	0	0	0	1	2	280	1	0
0	21.67	11.5	1	5	3	0.0	1	1	11	1	2	0	1	1
1	20.17	8.17	2	6	4	1.96	1	1	14	0	2	60	159	1
0	15.83	0.585	2	8	8	1.5	1	1	2	0	2	100	1	1
1	17.42	6.5	2	3	4	0.125	0	0	0	0	2	60	101	0
0	58.67	4.46	2	11	8	3.04	1	1	6	0	2	43	561	1
1	27.83	1.0	1	2	8	3.0	0	0	0	0	2	176	538	0

Uz pomoć alata iz sklearn biblioteke, izvršena je statistička analiza obeležja. Rezultati pokazuju u kojoj meri određeno obeležje može da doprinese dobroj klasifikaciji. Uz pomoć matplotlib i seaborn biblioteka pojedini rezultati su grafički predstavljeni kako bi se stvorila jasnija slika uticajima obeležja. U tabeli 4.3.3 prikazani su rezultati *hi* kvadratnog testa obavljenog nad prebrojivim obeležjima. Rezultati su sortirani u rastućem poretku.

Tabela 4.3.3– Rezultati *hi* kvadratnog testa

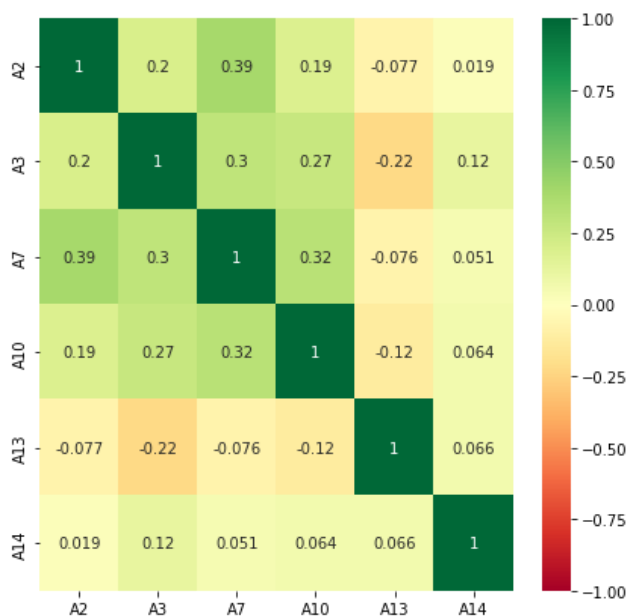
Obeležje	Rezultat
A5	177.0702 68
A8	170.7463 88
A9	82.96584 4
A6	35.43068 6
A4	2.723335
A12	0.423695
A11	0.374048
A1	0.042875

Osim *hi* kvadratnog testa, izvršen je proračun informacione dobiti svakog prebrojivog obeležja u skupu. Rezultati ove analize prikazani su na dijagramu koji se nalazi na slici 4.3.1.



Slika 4.3.1– Dijagram informacione dobiti

Nad neprebrojivim obeležjima urađena je analiza međusobne korelacije. Na slici 4.3.2 prikazana je matrica korelacije. Svako polje matrice obeležava presek dva obeležja u kojima je upisana vrednost korelacije. Naglasenim bojama (jarko crvena, tamno zelena) obeležene su jake korelacione veze, dok su svetlijom naglašene slabe veze.



Slika 4.3.2– Matrica korelacije

Prethodne analize govore da obeležja A1, A4, A11, A12 nemaju preveliki uticaj na ciljno obeležje tako da su ovo kandidati za eventualno izbacivanje iz skupa. Osim njih iz matrice korelacije je primećeno da A2 i A7 imaju određeno uzajamno dejstvo, ali ne preterano pa bi se moglo razmotriti testiranje bez jednog od ova dva obeležja. Za testiranje je neophodan model koji bi pokazao da li i u kojoj meri izbacivanje nekog obeležja utiče na performanse. Model koji će se

koristiti zahteva da ulazni podatak bude dvodimenzionalna grafička prezentacija pa će se zbog toga tabelarni podaci transformisati u ovaj oblik. Zbog specifičnosti metode koja će se koristiti za transformaciju, potrebno je prebaciti sva obeležja neprebrojivog u prebrojiv tip.²¹ Ocenu kvaliteta grupisanja vrednosti obeležja meriće se pomoću koeficijenta opravdanosti i informacione koristi. U tabeli 4.3.4 prikazan je primer kako izgleda raspodela po grupama sa opisnim svojstvima.

Tabela 4.3.4– A3 obeležje nakon grupisanja

Ime grupe	Opseg vrednosti	N	Np	Nn	Dp	Dn	WoE	IV
A3_1	(-0.001, 0.247]	46	23	23	0.074919	0.060052	0.221187	0.003288
A3_2	(0.247, 0.5]	52	20	32	0.065147	0.083551	-0.24882	0.004579
A3_3	(0.5, 0.75]	46	11	35	0.035831	0.091384	-0.93627	0.052013
A3_4	(0.75, 1.084]	40	17	23	0.055375	0.060052	-0.08109	0.000379
A3_5	(1.084, 1.5]	55	11	44	0.035831	0.114883	-1.16511	0.092104
A3_6	(1.5, 1.934]	37	16	21	0.052117	0.05483	-0.05075	0.000138
A3_7	(1.934, 2.4]	55	16	39	0.052117	0.101828	-0.66979	0.033295
A3_8	(2.4, 3.0]	38	16	22	0.052117	0.057441	-0.09727	0.000518
A3_9	(3.0, 4.016]	45	15	30	0.04886	0.078329	-0.47196	0.013908
A3_10	(4.016, 5.055]	46	25	21	0.081433	0.05483	0.395541	0.010523
A3_11	(5.055, 6.75]	47	29	18	0.094463	0.046997	0.698111	0.033136
A3_12	(6.75, 9.432]	45	27	18	0.087948	0.046997	0.626652	0.025662
A3_13	(9.432, 11.0]	49	28	21	0.091205	0.05483	0.508869	0.01851

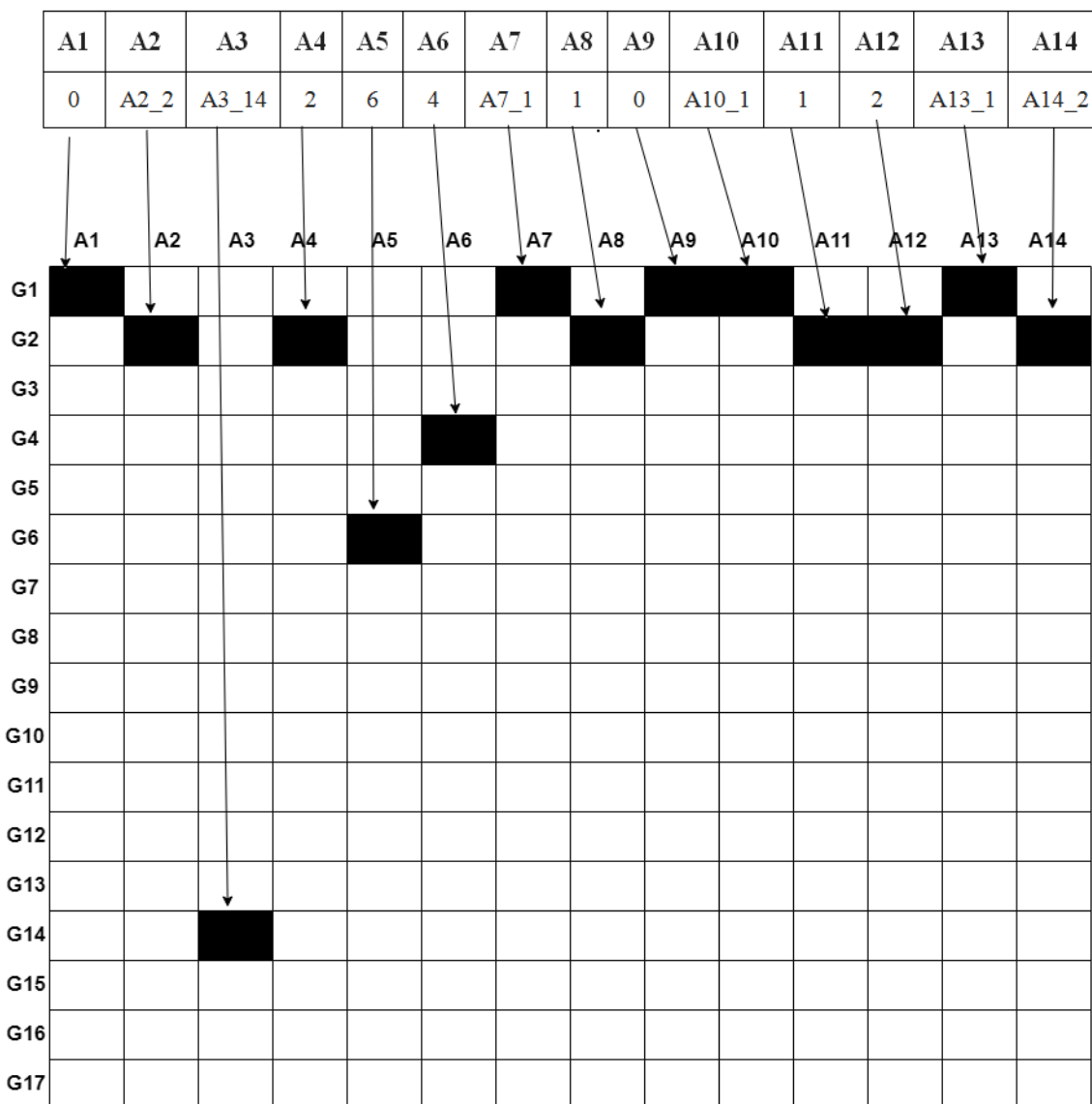
Ime grupe	Opseg vrednosti	N	Np	Nn	Dp	Dn	WoE	IV
A3_14	(11.0, 13.0]	45	24	21	0.078176	0.05483	0.354719	0.008281
A3_15	(13.0, 28.0]	44	29	15	0.094463	0.039164	0.880433	0.048686
Ukupno IV								0.34502

Na isti način urađeno je grupisanje za ostala neprebrojiva obeležja s tim da je broj grupa različit za svako pojedinačno obeležje. Broj grupa je odabran tako da se maksimizuje informaciona dobit (IV), a da ostane dovoljan broj uzoraka po grupama. U tabeli 4.3.5 date su informacije o broju grupa po obeležju kao ukupni IV koji ostvaruju.

Tabela 4.3.5– Broj grupa po obeležju sa ukupnim IV

Naziv obeležja	Broj grupa	Ukupno IV
A2	17	0.23174
A3	15	0.34502
A7	5	0.57963
A10	2	1.26324
A13	3	0.25326
A14	4	0.23168

U ovom eksperimentu koristiće se svih 14 obeležja tako da će širina biti ove veličine. Najveći broj grupa poseduje obeležje A2 (17) pa će podaci, kojim će se trenirati/testirati model, biti dimenzija 14x17. Na slici 4.3.3 ilustrovan je postupak mapiranja jednog zapisa iz skupa podataka. Slika je prikazana u negativ varijanti zbog bolje preglednosti.



Slika 4.3.3– Postupak mapiranja podataka u sliku

Postupak transformacije se ponavlja za svaki zapis ukupnog skupa podataka tako da će trening i test podaci činiti 690 slika.

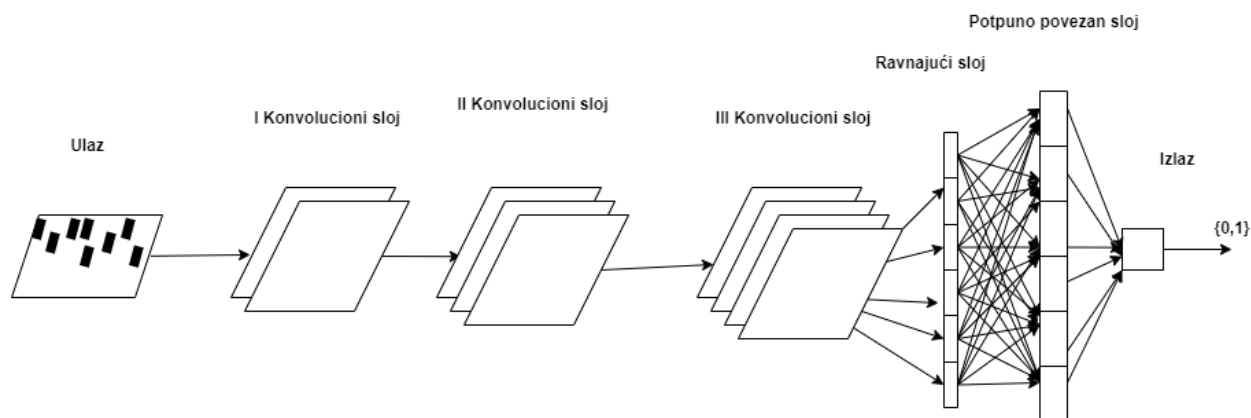
Razvijanje modela

Razvijanje modela je iterativni proces u kome je cilj kreiranje arhitekture koja je prilagođena obliku ulaznih podataka. Specifičnost ovog eksperimenta zahteva još veću preciznost jer je ulazni podatak jako malih dimenzija gde svaki piksel može u velikoj meri da utiče na rezultat predikcije. Osim toga, potrebno je optimizovati i hiperparametre koji se koriste u slojevima mreže i u procesu učenja. Ovo je takođe iterativni postupak koji zahteva puno testiranja kako bi

hiperparametri dobili vrednosti koji doprinose većoj tačnosti modela. Nakon brojnih testiranja, model koji je dao najbolje rezultate sastoji se od sledećih slojeva:

- Konvolucionni sloj I (veličina filtera: 5X5, broj filtera: 128, pomeraj: 2, aktivaciona funkcija: ReLU)
- Konvolucionni sloj II (veličina filtera: 3X3, broj filtera: 256, pomeraj:2, aktivaciona funkcija: ReLU)
- Konvolucionni sloj III (veličina filtera: 2X2, broj filtera: 512, pomeraj: 1, aktivaciona funkcija: ReLU)
- Ravnajući sloj
- Potpuno povezan sloj (broj čvorova: 16, aktivaciona funkcija: ReLU)
- Izlazni sloj (broj čvorova: 1, aktivaciona funkcija: sigmoid)

U konvolucionim slojevima korišćene su i pomoćne ivice čiji je broj prilagođen tako da izlaz sloja bude istih dimenzija kao i ulaz. Ravnajući sloj je element mreže koji izlaz konvolucionog sloja prilagođava ulazu potpuno povezanog sloja. Ovaj sloj vrši transformaciju višedimenzionalnih podataka u jednodimenzionalni oblik. Na slici 4.3.4 prikazana je arhitektura modela koji se koristi u eksperimentu.

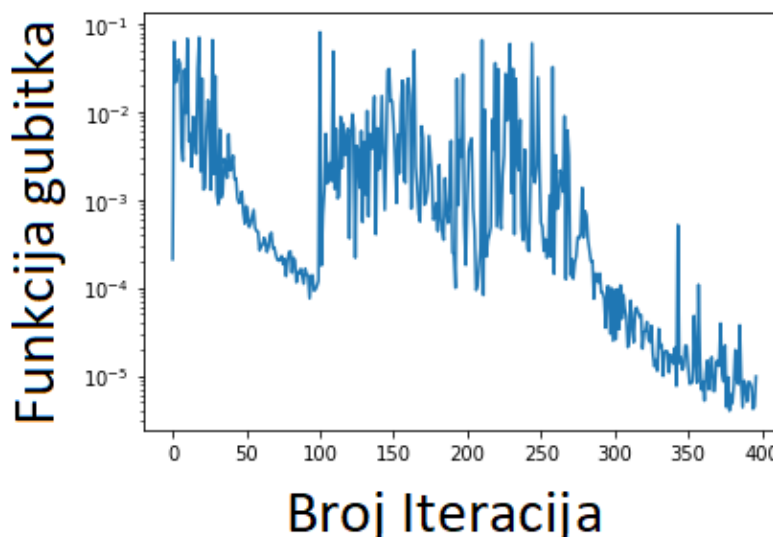


Slika 4.3.4—Arhitektura modela

Da bi model mogao da započene proces učenja potrebno je, osim arhitekture, definisati funkciju gubitka (greške), optimizator, kao i hiperparametre potrebne da bi se postavili kriterijumi treniranja i progresije učenja. Za funkciju greške koristi se greška unakrsne entropije. Kako se ne koristi regularizacija, ovo će ujedno biti i funkcija gubitka. Ostale vrednosti parametra prikazani su u sledećoj listi:

- Optimizator: Adam
- Korak učenja: $5e-4$
- Broj epoha: 100
- Veličina serije: 115

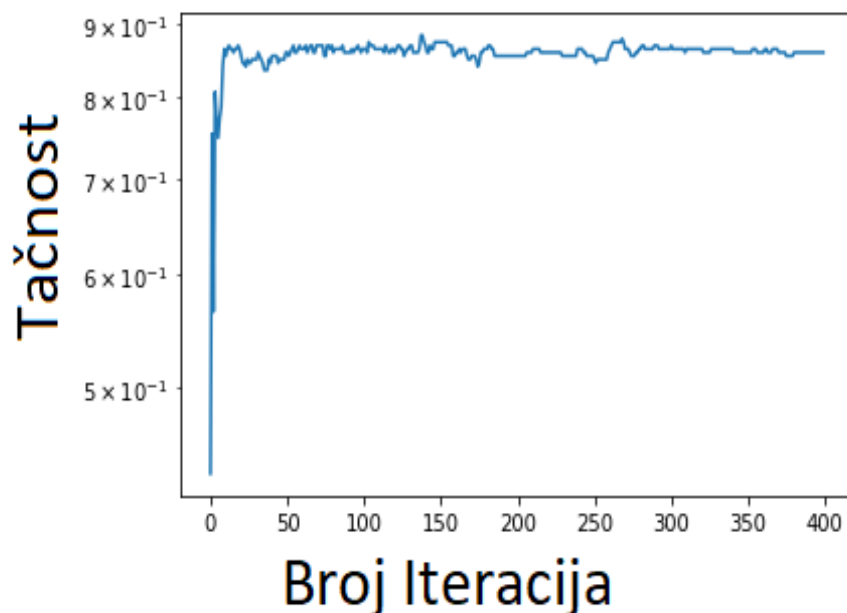
Proces treniranja ponavljan je nekoliko puta kako bi se dobila prosečna tačnost. U svakom ponavljanju, težinski koeficijenti i korektivni faktori imaju isto početno stanje. Raspodela na trening i test podatke vršena je u odnosu 70:30 (respektivno). U svakom ponavljanju, trening i test podaci birani su na slučajan način kako bi se osigurala nepristrasnost modela. Na slici 4.3.5 prikazan je dijagram toka kretanja funkcije gubitka tokom treniranja.



Slika 4.3.5- Dijagram toka funkcije gubitka

Evaluacija modela

Ocena efikasnosti modela vršena je na osnovu prosečne vrednosti pojedinačnih rezultata dobijenih u svakom ponavljanju procesa učenja. Metrika koja je korišćena za ocenu modela je tačnost (engl. Accuracy) koja se računa kao odnos pravilno klasifikovanih i ukupnog broja uzoraka korišćenih pri testiranju. Da bi se odredilo da li i u kojoj meri model pati od preprilagođavanja, vršena je i evaluacija modela sa trening podacima. Odnos između rezultata nad test i trening podacima opisuje stepen pristrasnosti modela. Na slici 4.3.6 prikazan je dijagram toka tačnosti modela tokom treniranja nad test podacima.



Slika 4.3.6– Dijagram toka tačnosti modela

U procesu evaluacije uzimaće se samo najviša vrednost koju je model postigao u nekom trenutku treniranja. Postupak ponovnog treniranja i testiranja obavljen je 7 puta. U tabeli 4.3.6 prikazani su pojedinačni rezultati kao i prosečna vrednost.

Tabela 4.3.6- Rezultati testiranja

Redni broj	Tačnost (test podaci)	Tačnost (trening podaci)
1	0.9959	0.884058
2	1.0000	0.8888889
3	0.9689	0.85507244
4	0.9979	0.89855075
5	1.0000	0.8888889
6	1.0000	0.87439615
7	1.0000	0.87922704
Prosek	0.994671	0.881297

Iz priloženog se može videti da je model klasifikovao tačno približno 89% test primera. Ovaj podatak pokazuje da metodologija transformacije tabelarnih podataka u sliku funkcioniše, a isto tako i da je novodobijene podatke moguće iskoristiti za model konvolucione neuronske mreže. Sa druge strane, može se uočiti i velika razlika između tačnosti nad trening i test podacima, što je znak da se i ovde javlja preprilagođavanje modela. Sledeći korak baviće se načinima kojima se može smanjiti pristrasnost modela. Prvi način je da se iz skupa obeležja izostave ona koja imaju veliku međusobnu korelaciju ili ona koja nemaju preveliki uticaj na ciljnu promenljivu. Ovim se

postize da model na osnovu manje podataka napravi bolju generalizaciju i na taj način izvrši klasifikaciju po merodavnijim i ispravnijim kriterijumima. U prethodnom poglavlju urađena je analiza kojom su obeležja A1, A4, A11, A12 okarakterisana kao slabi prediktori pa će se pri treniraju narednog modela ova obeležja izostaviti. Analizom je takođe ustanovljeno da obeležja A2 i A7 imaju veći stepen međusobne korelacije pa će se jedno od ova dva obeležja izostaviti. Na osnovu informacione dobiti koju imaju ova obeležja, A2 je izostavljeno jer ima manju vrednost. U tabeli 4.3.7 prikazani su rezultati pojedinačnih testova kao i prosečne vrednosti tačnosti modela nad test i trening podacima nakon selekcije obeležja.

Tabela 4.3.7- Rezultati testiranja

Redni broj	Tačnost (test podaci)	Tačnost (trening podaci)
1	0.9876	0.884058
2	0.9876	0.87922704
3	0.9876	0.87439615
4	0.9876	0.884058
5	0.9876	0.87922704
6	0.9876	0.884058
7	0.9876	0.87439615
Prosek	0.9876	0.879917197

Na osnovu novodobijenih rezultata može se zaključiti da izmene u skupu obeležja koja su se primenile nemaju preveliki uticaj na krajnje rezultate. Ovo može biti indikacija da model može sam da prepozna koja su obeležja merodavnija od ostalih. Ipak i dalje je ostala velika privrženost modela trening podacima, tako da će se u narednom koraku primeniti tehnika regularizacije.

²²Regularizacija je postupak u kome se teži smanjivanju preprilagođenosti modela. Uopšteno, postupak se ogleda u tome da se velike magnitude parametara modela penalizuju kako bi im se vrednosti smanjile. Pod parametrima modela u ovom slučaju se podrazumevaju težinski koeficijenti neurona u sloju mreže. Regulacioni izrazi direktno utiču na povećanje funkcije gubitka tako da će se sa pojavom velikih magnituda, u težinskim koeficijentima, povećati i funkcija gubitka. Na ovaj način se reguliše veličina parametara što direktno vodi i do smanjenja privrženosti modela podacima. Svaki regularizacioni model ima stepen regularizacije koji kontroliše u kojoj meri se vrši kontrola veličina. Step regularizacije se obično obeležava simbolom λ . Odabir ove vrednosti treba da se izvrši tako da regularizacioni izraz dovede do veće generalizacije modela, ali ne previše. Prevelika vrednost ovog parametra može da dovede do nedovoljno prilagođenog (engl. Underfitting) modela. Krajnji cilj je da se izbalansiraju performanse modela nad trening i test podacima. U zavisnosti od toga koji se model koristi i koje sporedne efekte je potrebno postići, biraju se odgovarajući regularizacioni izrazi od kojih su sledeća dva najkorišćenija: L1 i L2.

²³**L1** regularizacija poznata i kao L1 ili Lasso (engl. least absolute shrinkage and selection operator) kada se primenjuje u linearnoj regresiji, smanjuje preprilagođavanje modela tako što smanjuje vrednosti parametara (težinskih koeficijenata) ka nuli što može dovesti da neke karakteristike/obeležja budu ignorisane u toku procesiranja podataka u mreži. Kada se stepen regularizacije (λ) povećava, povećava se i broj težinskih koeficijenata (W_j) koji su blizu ili imaju vrednost jednaku nuli. Formula za izračunavanje vrednosti L1 regularizacionog izraza ($R(w)$) koji će se dodati funkciji gubitka ima sledeći oblik:

$$R(w) = \lambda \sum_{j=1}^n |W_j|$$

²⁴**L2** regularizacija poznata i kao grebena (engl. Ridge), kada se primenjuje u linearnoj regresiji, smanjuje preprilagođavanje modela tako što smanjuje vrednosti parametara (W_j), ali ne do te mere da dobiju vrednost jednaku nuli. Formula za izračunavanje vrednosti L2 regularizacionog izraza ($R(w)$) ima sledeći oblik:

$$R(w) = \lambda \sum_{j=1}^n W_j^2$$

U poređenju prethodno navedenih izraza za regularizaciju može se zaključiti da je L1 manje stabilna gde male promene stepena regularizacije mogu dovesti do većih promena modela dok su modeli sa L2 otporniji na ove efekte. Izbor izraza i stepena regularizacije treba vršiti na osnovu validacije modela sa različitim vrednostima. Regularizacija se može izvršavati nad pojedinačnim slojevima mreže u kojima se mogu i kombinovati različiti regularizacioni izrazi, tako da u eksperimentima treba primeniti i ovakve varijacije.

Nakon višestrukih testiranja ustanovljeno je da L2 regularizacioni izraz postavljen na svim slojevima mreže sa stepenom 0.4 dovodi do povećanja performansi modela i veće generalizacije. Prilikom testiranja primećeno je da model i u kasnijim iteracijama (epohama) pravi proboje u tačnosti nad test podacima pa je broj epoha povećan na 300. Konačni rezultati pojedinačnih testiranja prikazani su u tabeli 4.3.8.

Tabela 4.3.8- Rezultati testiranja

Redni broj	Tačnost (test podaci)	Tačnost (trening podaci)
1	0.93719804	0.9669
2	0.89855075	0.9586
3	0.89855075	0.9503
4	0.9227053	0.971
5	0.90338165	0.9607
6	0.89855075	0.9545
7	0.93719804	0.9627
Prosek	0.913733611	0.960671429

Na osnovu ovih rezultata može se videti uticaj regularizacije na perfomanse modela gde se primećuje da je smanjena tačnost nad trening podacima, a da je povećana nad test podacima što je posledica veće generalizacije modela.

U narednom eksperimentu ponoviće se prethodni koraci, ali nad novim skupom podataka kako bi se proverila validnost prethodnih uvida. Novi skup podataka sastoji se od 32 obeležja, od kojih je jedna ciljna promenljiva, a obuhvata 569 instanci podataka. Osim ciljne promenljive i identifikacionog broja, sva obeležja su neprebrojivog tipa. Skup sadrži biometrijske nalaze tkiva tumora dojke od kojih je 63% označeno kao benigni, a 37% kao maligni. Model treba na osnovu ulaznih podataka da zaključi da li je tumor zloćudan ili nije.

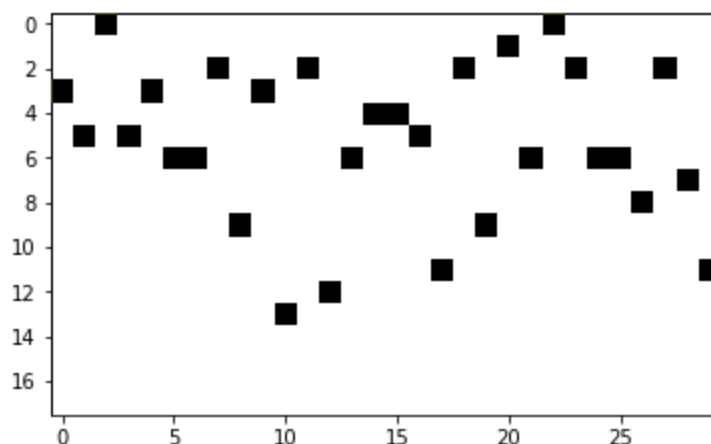
Nakon izvršene pripreme i analize podataka, iz celokupnog skupa obeležja izbačeno je obeležje „id“ (identifikacioni broj) jer sigurno ne utiče na rezultat predikcije, dok su ostala zadržana bez obzira na rezultate statističkih analiza koje su izvršene. Ideja je da se u prvom modelu ostave sva obeležja dok će se u drugom izbaciti ona koja su se analizom pokazala kao loši prediktori. Upoređivanjem perfomansi ova dva modela, utvrdiće se da li je ovakav pristup obrade otporan na podatke koje nemaju merodavne informacije što je pokazano i u prethodnom eksperimentu. U tabeli 4.3.9 prikazana su obeležja koja će se koristiti kao i broj kategorija koje svako od njih poseduje nakon izvršene grupacije vrednosti. Osim broja kategorija prikazane su i pojedinačne informacione dobiti za svako obeležje.

Tabele 4.3.9- Broj grupa po obeležju sa ukupnim IV

Naziv obeležja	Broj grupa	Ukupno IV
radius_mean	5	4.673082805240479
texture_mean	17	1.2717340651548492
perimeter_mean	4	4.071093623780412
area_mean	6	4.453006430691186
smoothness_mean	17	0.8169275633931412
compactness_mean	17	2.739599477020948
concavity_mean	10	4.7771332247176
concave points_mean	3	4.483923894549508
symmetry_mean	17	0.7108211332745628
fractal_dimension_mean	17	0.31661440672696045
radius_se	17	4.2375691369518815
texture_se	17	0.18342735932389082
perimeter_se	17	3.843439520735495
area_se	7	4.093871907512801
smoothness_se	17	0.2337505977351828
compactness_se	17	1.3423875374733305
concavity_se	17	2.4537028096151623
concave points_se	17	1.9108252729359272
symmetry_se	17	0.10917242697170587

fractal_dimension_se	17	0.37922489388133807
radius_worst	3	4.879222074678701
texture_worst	17	1.4001790106935454
perimeter_worst	3	4.978936795623927
area_worst	3	4.867913014979207
smoothness_worst	17	1.0263619143744316
compactness_worst	17	2.8712581223321627
concavity_worst	17	4.8248355804585845
concave points_worst	3	4.759565347075222
symmetry_worst	17	1.0378823179948147
fractal_dimension_worst	17	0.6169723846878854

Tranformacija tabelarnih podataka u sliku vrši se na isti način kao i u prethodnom primeru s tim da se zbog broja obeležja i najvećeg broja grupa, dimenzije rezultujućih slika razlikuju. Dimenzije slika u ovom primeru ce biti 31x10. Na slici 4.3.10 prikazana je ilustracija jednog od uzoraka iz skupa podataka nakon transformacije u sliku. Slika je prikazana u negativ varijanti zbog bolje preglednosti.



Slika 4.3.10- Mapirani tabelarni podaci u sliku

Arhitektura modela je ostala ista kao i u prošlom eksperimentu s tim da su promenjene neke od vrednosti hiperparametara kako bi se poboljšale performanse. Nove vrednosti date su u sledećem delu:

Broj epoha: 200

Stepen regularizacije: 0.1.

Korak učenja: 4e-4

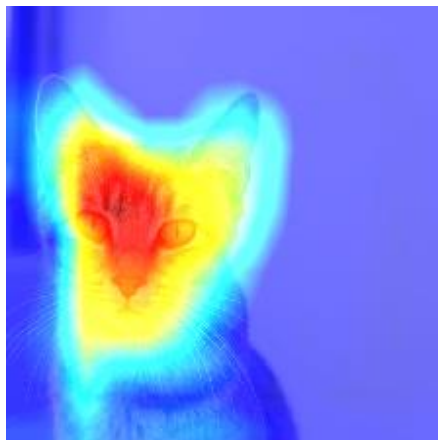
Nakon evaluacije prvog modela dobijena je prosečna tačnost nad test podacima od 0.9736 i 0.997

. Drugi model, u kome su izbačeni slabi prediktori, ostvario je tačnost od 0.974 za test i 0.9972 trening podatke. Iz ovoga se može videti da slabi prediktori ne utiču na model ni u ovom slučaju. Takvo saznanje se možda može iskoristiti u primeni ovakvog pristupa u neke sekundarne svrhe u odnosu na prvobitnu namenu. Osim toga model je pokazao jednako dobre, a negde i bolje rezultate u odnosu na algoritme koji se trenutno aktuelno koriste u praksi. Na repozitorijumu odakle su preuzeti podaci nalaze se i rezultati koji su postignuti drugim tehnikama i algoritmima mašinskog učenja. U tabeli 4.3.11 prikazi su rezultati različitih algoritama koji su postignuti nad ovim podacima. Evaluacija modela se zasnivala na različitim metrikama: tačnost, preciznost, odziv i rezultati F1 testa. U poslednjem redu prikazani su rezultati koje je ostvario model čije je razvijanje prikazano u ovom radu bez selekcije obeležja.

Tabela 4.3.11- Perfomanse modela različitih metoda učenja

Metod učenja	Odziv	Tačnost	Preciznost	F1
SVM	0.946429	0.972028	0.981481	0.963636
KNN	0.928571	0.958042	0.962963	0.945455
Adaboost	0.928571	0.965035	0.981132	0.954128
Logistic Regr	0.910714	0.965035	1	0.953271
Random Forest	0.910714	0.944056	0.944444	0.927273
Gradien Boosting	0.910714	0.951049	0.962264	0.93578
Decision Tree	0.892857	0.909091	0.877193	0.884956
model	0.97	0.9766082	0.96	0.96

I pored dobrih rezultata koje model postiže, teško se može iskoristiti za upotrebu u realnim uslovima kada je na primer reč o klasifikaciji pacijenata ili klijenata banke. Razlog tome je jer model ne može tačno da definiše pravila na osnovu kojih je doneo neki zaključak što u praksi predstavlja problem. Klasifikacioni modeli najčešće se koriste u svojstvu asistenata ili dodatne pomoći prilikom donošenja odluke što ovaj model može da pruži, ali bez objašnjenja tako da nije od neke pomoći. S druge strane, nakon uvida u kojima se pokazalo da model ima iste performanse sa podacima u kojima se nalaze sva obeležja i model koji je obučen bez loših prediktora, otvaraju se mogućnosti da se ovaj pristup možda iskoristi u svojstvu odabira i rangiranja obeležja. Postoje različite tehnike mapiranja ²⁵istaknutosti (engl. Saliency map) koja kreira grafički prikaz u kome su naznačene regije na slici koje utiču na model da donese datu odluku. Osim toga jačina boje kojom je obeležena regija, pokazuje u kojoj meri utiče na rezultat. Ovo se može iskoristiti kod modela i pristupa, koji je opisan u ovom radu, za primenu u analizi kojom bi se pokazalo koja obeležja, koje vrednosti i u kojoj meri utiču na izlaz modela tj. ciljnu promenljivu. Na slici 4.3.12 je ilustrovan jedan primer mape istaknutosti koja je prikazana preko originalne slike. Mapa istaknutosti kreirana je primenom ²⁶Grad CAM (engl. Gradient-weighted Class Activation Mapping) algoritma. Slika ilustruje koje regije na ulaznoj slici su zaslužne da model pravilno klasifikuje da li se na njoj nalazi mačka.



Slika 4.3.12- Ilustracija Grad CAM algoritma [27]

U narednim radovima treba ispitati mogućnosti da se ovakav pristup iskoristi za rangiranje obeležja iz skupa podataka. Nakon toga novodobijeni skup se može iskoristiti u drugim modelima koji imaju mogućnost da pruže adekvantne razloge zbog kojih je doneta neka odluka. Osim toga, treba ispitati druge tipove arhitekture mreže koje imaju dobre rezultate u postizanju veće generalizacije. Neki od primera takvih mreža je ²⁸CAE (engl. Convolutional Autoencoder) arhitektura koja se može iskoristiti da se poveća stepen generalizacije modela konvolucione neuronske mreže.

5. Zaključak

Problem binarne klasifikacije javlja se sa pojavom prvih sistema za automatizovano donošenje odluka. Vremenom su se ovi sistemi razvijali sa ciljem da se postignu sve bolji rezultati i da se u većem obimu nađu u praktičnoj primeni. Rešavanje problema binarne klasifikacije je tako dobilo mnoštvo novih pristupa u rešavanju i unapređivanju postojećih sistema. Trenutno postoje algoritmi koji postižu dobre rezultate u ovoj oblasti, ali prostor za unapređivanje i dalje postoji. U ovom radu opisan je nov pristup rešavanja problema u kojem se želi postići veća iskorišćenost potencijala neuronskih mreža. Konvolucione neuronske mreže, kao specifičan tip, nude niz pogodnosti koje treba razmotriti. Ove mreže zahtevaju da ulazni tip podatka bude u formatu dvodimenzionalne grafičke prezentacije pa je u ovom radu predstavljena metodologija pretvaranja tabelarnih podataka u sliku i analiza stepena korisnosti koji se na taj način postiže. Ipak problem kod konvolucionih mreža je što ne mogu dati jasno definisane kriterijume i razloge zbog kojih je doneta neka odluka. U najvećem broju praktične primene modela veštačke inteligencije ovo predstavlja problem jer se koriste u službi dodatne pomoći pri odlučivanju gde bez opisa razloga, odluka nema merodavnost i opravdanost.

Eksperimentom je pokazano da ovaj pristup može imati dobre rezultate i da postoji prostor za dalje unapređivanje. Ipak iz razloga koji su ranije navedeni, ovakav pristup se ne može efikasno primeniti u praksi, ali postoje indikacije da se može iskoristiti u drugim primenama kao što je analiza pojedinačnih obeležja nekog skupa podataka.

6. IZJAVA O AKADEMSKOJ ČESTITOSTI

IZJAVA O AKADEMSKOJ ČESTITOSTI

Student (ime, ime jednog roditelja i prezime):

Dušan, Goran, Marković

Broj indeksa:

RIN-56/17

Pod punom moralnom, materijalnom, disciplinskom i krivičnom odgovornošću izjavljujem da je master rad, pod naslovom:

Jedan pristup mapiranja podataka iz tabela u grafičku dvodimenzionalnu reprezentaciju za potrebe treniranja modela konvolucione neuronske mreže

1. rezultat sopstvenog istraživačkog rada;
2. da ovaj rad, ni u celini, niti u delovima, nisam prijavljivo/la na drugim visokoškolskim ustanovama;
3. da nisam povredio/la autorska prava, niti zloupotrebio/la intelektualnu svojinu drugih lica;
4. da sam rad i mišljenja drugih autora koje sam koristio/la u ovom radu naznačio/la ili citirao/la u skladu sa Uputstvom;
5. da su svi radovi i mišljenja drugih autora navedeni u spisku literature/referenci koji je sastavni deo ovog rada, popisani u skladu sa Uputstvom;
6. da sam svestan/svesna da je plagijat korišćenje tuđih radova u bilo kom obliku (kao citata, prafraza, slika, tabela, dijagrama, dizajna, planova, fotografija, filma, muzike, formula, vebajtova, kompjuterskih programa i sl.) bez navođenja autora ili predstavljanje tuđih autorskih dela kao mojih, kažnjivo po zakonu (Zakon o autorskom i srodnim pravima), kao i drugih zakona i odgovarajućih akata Visoke škole elektrotehnike i računarstva strukovnih studija u Beogradu;
7. da je elektronska verzija ovog rada identična štampanom primerku ovog rada i da pristajem na njegovo objavljivanje pod uslovima propisanim aktima Visoke škole elektrotehnike i računarstva strukovnih studija u Beogradu;
8. da sam svestan/svesna posledica ukoliko se dokaže da je ovaj rad plagijat.

U Beogradu, __. __. 2022. godine.

Својеручни потпис студента

7. Literatura

- ¹ F. Rosenblatt, Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms (a version of a 1961 report of the same title done for Cornell Aeronautical Labs.). Washington, DC: Spartan Books, 1962.
- ² Eberhart, R. C., & Dobbins, R. W. (1990). Early neural network development history: the age of Camelot. IEEE Engineering in Medicine and Biology Magazine, 9(3), 15-18.
- ³ http://www.biologija.rs/nervni_sistem.html
- ⁴ Guez, A., Eilbert, J. L., & Kam, M. (1988). Neural network architecture for control. IEEE control systems Magazine, 8(2), 22-25.
- ⁵ Sharma, S., Sharma, S., & Athaiya, A. (2017). Activation functions in neural networks. towards data science, 6(12), 310-316.
- ⁶ Günther, F., & Fritsch, S. (2010). Neuralnet: training of neural networks. R J., 2(1), 30.
- ⁷ Cortés-Ciriano, I., & Bender, A. (2018). Deep confidence: a computationally efficient framework for calculating reliable prediction errors for deep neural networks. Journal of chemical information and modeling, 59(3), 1269-1281.
- ⁸ Bera, S., & Shrivastava, V. K. (2020). Analysis of various optimizers on deep convolutional neural network model in the application of hyperspectral remote sensing image classification. International Journal of Remote Sensing, 41(7), 2664-2683.
- ⁹ Albawi S., Mohammed T., Al-azawi S. Understanding of a convolutional neural network. Proceedings of 2017 International Conference on Engineering & Technology (ICET'2017); August 2017; Antalya, Turkey: Akdeniz University; pp. 274–279. [Google Scholar]
- ¹⁰ <https://cs231n.github.io/convolutional-networks>)
- ¹¹ <https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53>
- ¹² Zhang, C. W., Yang, M. Y., Zeng, H. J., & Wen, J. P. (2019). Pedestrian detection based on improved LeNet-5 convolutional neural network. *Journal of Algorithms & Computational Technology*, 13, 1748302619873601.
- ¹³ Sharma, N., Jain, V., & Mishra, A. (2018). An analysis of convolutional neural networks for image classification. Procedia computer science, 132, 377-384.

-
- ¹⁴ Targ, S., Almeida, D., & Lyman, K. (2016). Resnet in resnet: Generalizing residual architectures. *arXiv preprint arXiv:1603.08029*.
- ¹⁵ Yu, L., Wang, S., & Lai, K. K. (2005). An integrated data preparation scheme for neural network data analysis. *IEEE Transactions on Knowledge and Data Engineering*, 18(2), 217-230.
- ¹⁶ Dash, M., & Liu, H. (1997). Feature selection for classification. *Intelligent data analysis*, 1(1-4), 131-156.
- ¹⁷ Liang, J., Wan, X., Liu, Q., Li, C., & Li, J. (2016). Research on filter selection method for broadband spectral imaging system based on ancient murals. *Color Research & Application*, 41(6), 585-595.
- ¹⁸ Khendek, F. B., Fujiwara, S., Bochmann, G. V., Khendek, F., Amalou, M., & Ghedamsi, A. (1991). Test selection based on finite state models. *IEEE Transactions on software engineering*, 17(591-603), 10-1109.
- ¹⁹ Jović, A., Brkić, K., & Bogunović, N. (2015, May). A review of feature selection methods with applications. In 2015 38th international convention on information and communication technology, electronics and microelectronics (MIPRO) (pp. 1200-1205). Ieee.
- ²⁰ Wod, I. J. (1985). Weight of evidence: A brief survey. *Bayesian statistics*, 2, 249-270.
- ²¹ <https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=9386102>
- ²² Wan, L., Zeiler, M., Zhang, S., Le Cun, Y., & Fergus, R. (2013, May). Regularization of neural networks using dropconnect. In *International conference on machine learning* (pp. 1058-1066). PMLR.
- ²³ Ranstam, J., & Cook, J. A. (2018). LASSO regression. *Journal of British Surgery*, 105(10), 1348-1348.
- ²⁴ McDonald, G. C. (2009). Ridge regression. *Wiley Interdisciplinary Reviews: Computational Statistics*, 1(1), 93-100
- ²⁵ Monroy, R., Lutz, S., Chalasani, T., & Smolic, A. (2018). Salnet360: Saliency maps for omnidirectional images with cnn. *Signal Processing: Image Communication*, 69, 26-34.
- ²⁶ Selvaraju, R. R., Cogswell, M., Das, A., Vedantam, R., Parikh, D., & Batra, D. (2017). Grad-cam: Visual explanations from deep networks via gradient-based localization. In *Proceedings of the IEEE international conference on computer vision* (pp. 618-626).
- ²⁷ <https://tf-explain.readthedocs.io/en/latest/methods.html>
- ²⁸ Guo, X., Liu, X., Zhu, E., & Yin, J. (2017, November). Deep clustering with convolutional autoencoders. In *International conference on neural information processing* (pp. 373-382). Springer, Cham.