

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 138

**ISPITIVANJE UTJECAJA HIPERPARAMETARA DUBOKIH
NEURONSKIH MREŽA NA UČINKOVITOST KLASIFIKACIJE
POREMEĆAJA IZ 12-KANALNOG
ELEKTROKARDIOGRAMA**

Marin Matjanec

Zagreb, lipanj 2021.

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 138

**ISPITIVANJE UTJECAJA HIPERPARAMETARA DUBOKIH
NEURONSKIH MREŽA NA UČINKOVITOST KLASIFIKACIJE
POREMEĆAJA IZ 12-KANALNOG
ELEKTROKARDIOGRAMA**

Marin Matjanec

Zagreb, lipanj 2021.

ZAVRŠNI ZADATAK br. 138

Pristupnik: **Marin Matjanec (0036514463)**

Studij: Elektrotehnika i informacijska tehnologija i Računarstvo

Modul: Računarstvo

Mentor: izv. prof. dr. sc. Alan Jović

Zadatak: **Ispitivanje utjecaja hiperparametara dubokih neuronskih mreža na učinkovitost klasifikacije poremećaja iz 12-kanalnog elektrokardiograma**

Opis zadatka:

Slobodno dostupan skup podataka PTB-XL (<https://physionet.org/content/ptb-xl/1.0.1/>) sastoji se od velikog broja zapisa 12-kanalnih elektrokardiograma (EKG) koji sadržavaju 23 moguća poremećaja rada srca. Za računalnu klasifikaciju srčanih poremećaja iz ovih zapisa mogu se koristiti različite kombinacije algoritama strojnog učenja i obrade signala, no unazad zadnjih nekoliko godina najbolje rezultate daju algoritmi dubokog učenja. U ovom završnom radu potrebno je ispitati utjecaj arhitekture dubokih neuronskih mreža i njihovih hiperparametara na učinkovitost klasifikacije poremećaja iz 12-kanalnog EKG-a. Pritom se ispitivanje neuronskih mreža treba provesti na sirovim signalima, uz eventualno korištenje prozora određene širine kao metode za segmentaciju signala. Za učenje i razvoj modela mogu se koristiti vlastiti računalni resursi ili platforma Google Colab. Moguće je koristiti implementaciju neuronskih mreža u bilo kojem programskom jeziku po izboru, a osim vrednovanja, rezultat rada treba biti i aplikacija koja omogućava učitavanje zapisa, njihovo učenje ili testiranje te ispis rezultata.

Rok za predaju rada: 11. lipnja 2021.

Zahvala

Zahvaljujem se svom mentoru izv. prof. dr. sc. Alanu Joviću na uloženom trudu i savjetima koje mi je pružio tijekom izrade ovog rada. Također, zahvaljujem svojoj obitelji i prijateljima koji su mi bili velika podrška.

Hvala Vam!

Sadržaj

Uvod	1
1. Srce i elektrokardiogram	2
1.1. Srce.....	2
1.2. Elektrokardiogram.....	3
1.2.1. Elektrode i kanali.....	4
1.2.2. Značajke elektrokardiograma	7
2. Skup podataka PTB-XL.....	9
3. Ispitivanje hiperparametara	14
3.1. Korištene tehnologije	14
3.2. Početna mreža.....	16
3.3. Broj epoha i veličina grupe.....	17
3.3.1. Broj epoha.....	17
3.3.2. Veličina grupe	19
3.4. Isključivanje čvorova grafa	20
3.5. Hiperparametri konvolucijskog sloja	23
3.6. Hiperparametri LSTM sloja.....	26
3.7. Hiperparametri algoritma optimizacije.....	28
4. Aplikacija u Pythonu.....	35
Zaključak.....	37
Literatura	39
Sažetak	40
Summary.....	41

Uvod

Kardiovaskularne bolesti pripadaju najopasnijim bolestima na svijetu. Procijenjeno je da one svake godine uzmu 17.9 milijuna ljudskih života, što predstavlja od prilike 31% svih smrti godišnje. Ova statistika čini kardiovaskularne bolesti prvima na listi najsmrtonosnijih bolesti s kojima se ljudi susreću. Pristup osnovnim tehnologijama za analizu rada srca nužan je uvjet za pružanje adekvatne pomoći i liječenja osoba kojima je to potrebno.[1] Elektrokardiograf jedna je od najvažnijih tehnologija u tom području. To je uređaj koji bilježi električnu aktivnost srca u vremenu, a njegov proizvod je crtež koji se naziva elektrokardiogram (EKG). Analiza EKG-a problem je koji se može primijeniti u računarstvu koristeći umjetnu inteligenciju.

Ovaj završni rad bavi se dubokom neuronskom mrežom koja uči na temelju zapisa elektrokardiograma iz slobodno dostupnog skupa podataka PTB-XL koji sadrži 23 moguća poremećaja rada srca, te potom radi klasifikaciju na temelju danog zapisa.[2] Glavna tema rada je analiza utjecaja hiperparametara te neuronske mreže na uspješnost njene klasifikacije zapisa na odgovarajuće bolesti. Duboke neuronske mreže imaju velik broj hiperparametara, a svaki od njih važan je na svoj način i utječe na točnost klasifikacije zapisa. Optimizacija hiperparametara često zahtjeva cijeli tim inženjera i velike računalne resurse, zato će ovaj rad samo zagrebat površinu ovog jako velikog i vremenski skupog problema. Ali unatoč tome, potrudit će se jasno objasniti najvažnije hiperparametre i njihov utjecaj na neuronsku mrežu koju oni grade.

Rad sadrži četiri glavna poglavlja. U prvom poglavlju iznose se osnovne informacije o srcu i elektrokardiogramu. Razumijevanje elektrokardiograma i njegove reprezentacije kao signala bitan je dio ovog problema. U drugom dijelu opisuje se slobodno dostupan skup podataka koji je korišten za učenje i testiranje uspješnosti duboke neuronske mreže. Treći dio je glavni dio rada. On sadrži istraživanje i analizu utjecaja hiperparametara dubokih neuronskih mreža na uspješnost klasifikacije oboljenja. Osim analize, opisani su najvažniji hiperparametri koji su bili ispitani u istraživanju. Zadnji dio rada opisuje aplikaciju koja korisniku nudi mogućnost klasifikacije oboljenja iz danog zapisa EKG-a, mogućnost učenja duboke neuronske mreže, te ispis rezultata.

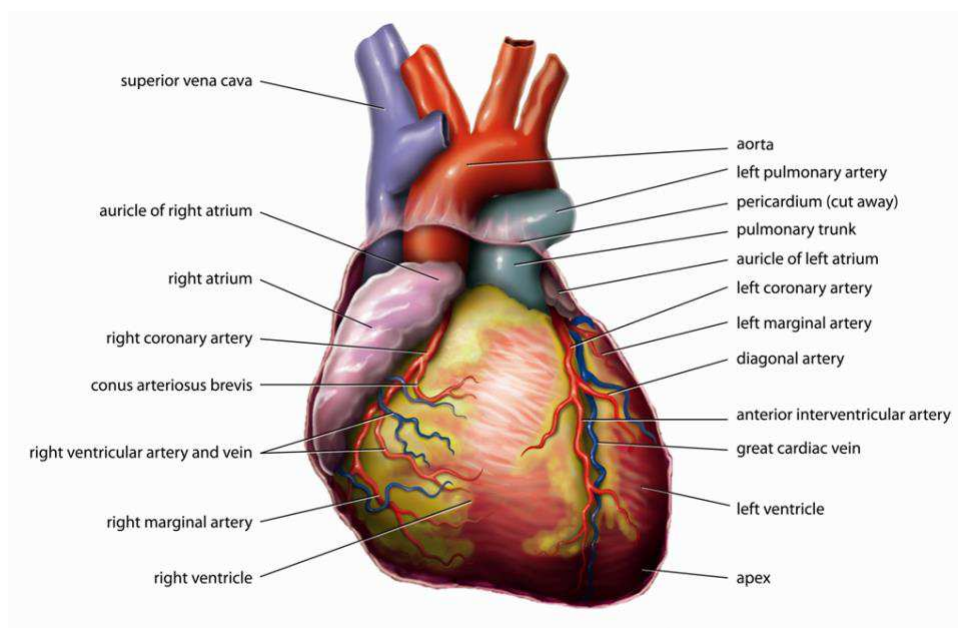
1. Srce i elektrokardiogram

EKG najvažniji je zapis za analizu kardiovaskularnih oboljenja. Njegovo razumijevanje bitno je za ovaj završni rad, jer duboka neuronska mreža koju učimo na svoj ulaz prima upravo zapise EKG-a. U sljedećim poglavljima bit će opisani EKG i srce. Većina teksta preuzeta je iz [3] i [6].

1.1. Srce

Srce je organ veličine šake smješten s lijeve strane prsnog koša čovjeka i jedan je od najvažnijih organa ljudskog tijela. Srce osigurava neprekidan protok krvi kroz ljudsko tijelo. Ono je podijeljeno u četiri glavne komore. Gornje dvije komore, lijeva i desna pretklijetka, te donje dvije komore, lijeva i desna klijetka. Pretklijetka se još naziva i atrij, dok se klijetka još naziva i ventrikul. Desna strana srca ima funkciju prikupljanja krvi bogate ugljikovim dioksidom u desnoj pretklijetki koja dolazi iz tijela preko gornje i donje šuplje vene, te izbacivanja te krvi iz desne klijetke u pluća. Lijeva strana prikuplja krv bogatu kisikom u lijevoj pretklijetki, a izbacuje ju kroz lijevu klijetku preko aorte do organa u tijelu.[4]

Pogled na srce izvana prikazan je na slici (Slika 1.1). Na njoj se jasno vide gornja šuplja vena i aorta, zajedno s ostalim dijelovima srca vidljivim izvana.



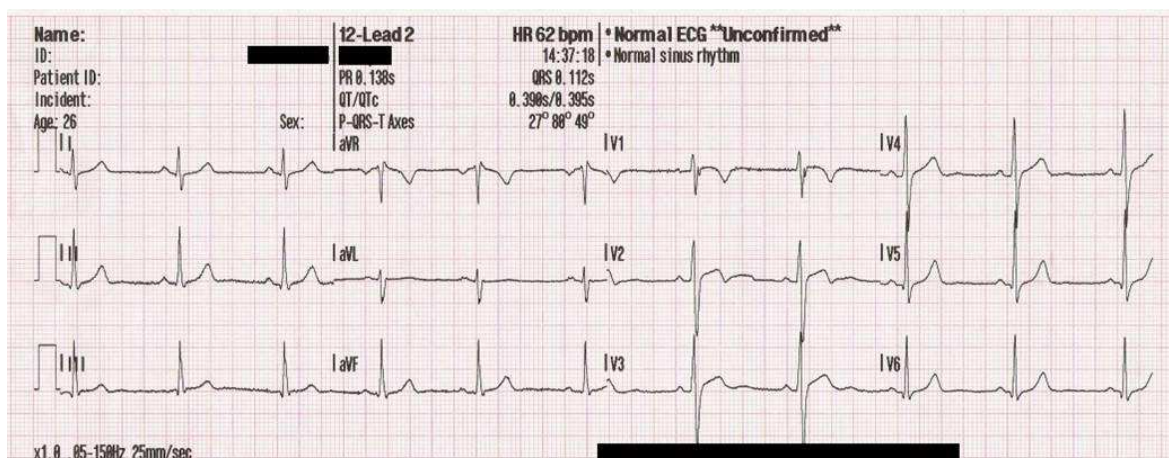
Slika 1.1 Pogled na srce izvana [4]

Pretilost, arterijska hipertenzija i visoki kolesterol mogu povećati rizik od razvoja srčane bolesti.[4]

Srce je organ koji nikad ne prestaje sa svojim kontrakcijama tijekom života, a za njegovu stimulaciju zaslužna je električna energija koja se javlja u sinoatrijalnom čvoru. Prije svakog normalnog otkucaja srca, ili sistole, val električne struje prođe kroz cijelo srce, koji potiče kontrakciju srčanog mišićnog tkiva (miokarda). Električni impuls ne širi se slučajno, već se širi po strukturi srca u koordiniranom uzorku što dovodi do učinkovite, koordinirane sistole. Ovo uzrokuje razliku potencijala koju je moguće izmjeriti i zabilježiti s površine ljudskog tijela. Rezultat tog zapisa nakon filtriranja i pojačanja predstavlja EKG.

1.2. Elektrokardiogram

Zbog bitne uloge srca u ljudskom zdravlju, a i povezanosti s oboljenjima, analiza nalaza EKG-a nužan je i rutinski dio svakog kompletnog medicinskog pregleda. EKG izumio je Willem Einthoven 1901. godine i za to 1924. godine dobio Nobelovu nagradu za fiziologiju ili medicinu. Slika 1.2 prikazuje jedan normalan zapis EKG-a.

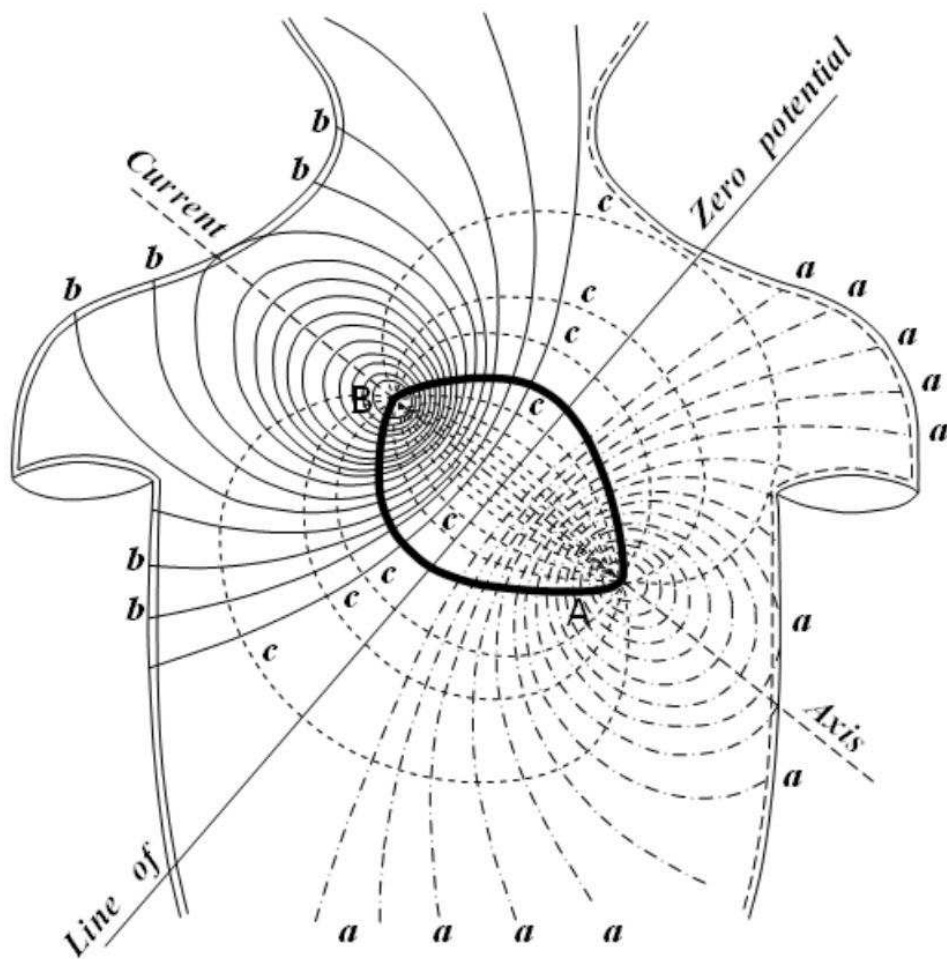


Slika 1.2 Zapis EKG-a

Razumijevanje osnova normalnog EKG-a zahtijeva uvažavanje četiri fenomena: elektrofiziologije pojedine stanice, kako se val električne struje širi kroz miokard, fiziologije specifičnih struktura srca kroz koje električni val putuje i posljednje kako to dovodi do mjerljivog signala na površini tijela, proizvodeći normalan EKG.

1.2.1. Elektrode i kanali

Slika 1.3 prikazuje električno polje srca na površini tijela. Srce smatramo dipolom čiji pozitivni i negativni polovi su smješteni u točkama *A* i *B*. Krivulje označene *a* i *b* su pozitivne i negativne izopotencijalne linije. Ako na površinu tijela stavimo elektrode, moći ćemo izmjeriti razliku potencijala uzrokovanih promjenama dipola srca.[5]



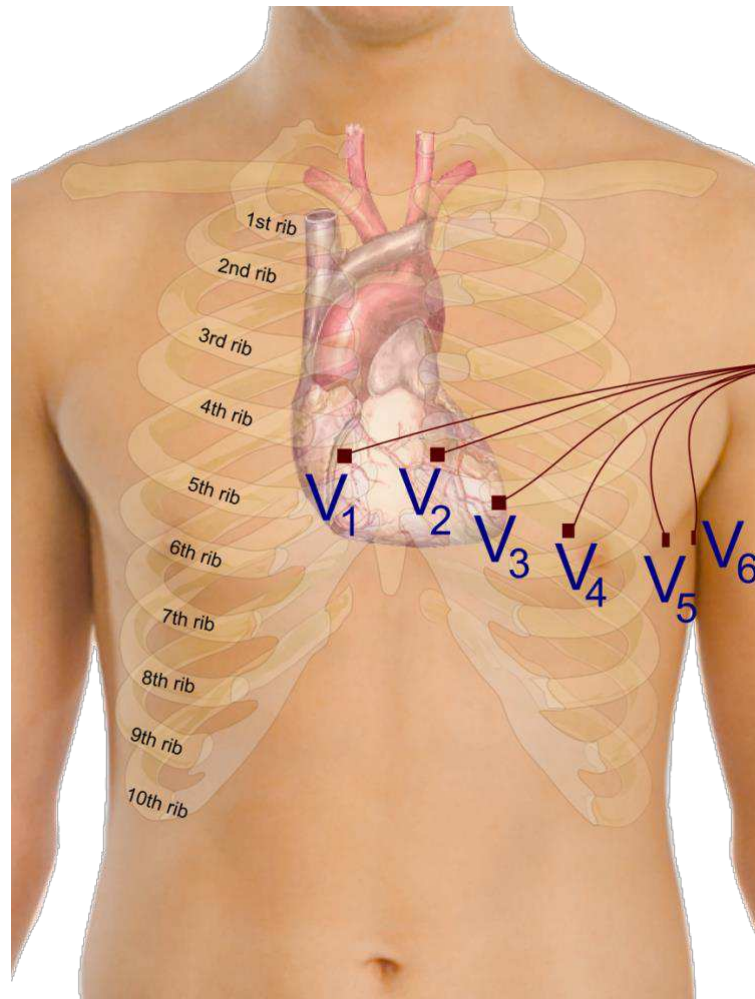
Slika 1.3 Električno polje srca na površini tijela [5]

Vektor srca je usmjerena veličina koja pokazuje u kojem smjeru srce širi električnu energiju. Kako se svaki val depolarizacije širi srcem, vektor srca mijenja veličinu i smjer u ovisnosti o vremenu. Kako se vektor srca širi, skuplja i okreće u trodimenzionalnom prostoru, tako se projicira na 12 različitih kanala dobro definirane orijentacije. Svaka elektroda otkriva veličinu srčanog vektora u smjeru te elektrode u svakom trenutku.

Elektrode su vodljiva tijela pričvršćena na površinu kože subjekta. Bilo koji par elektroda može izmjeriti razliku potencijala između dva odgovarajuća mjesta pričvršćenja. Takav par tvori jedan odvod ili kanal. Kanali se također mogu formirati između fizičke elektrode i virtualne elektrode, poznate kao Wilsonov središnji terminal, čiji se potencijal definira kao prosječni potencijal izmjeren pomoću tri elektrode ekstremiteta koje su pričvršćene na desnu ruku, lijevu ruku i lijevu nogu.[6]

Poznajemo I, II i III standardni odvod (diferencijalni kanal) i tri pojačana odvoda (aVR, aVL, aVF) koji se nazivaju i periferni odvodi, te šest prekordijalnih odvoda (V1, V2, V3, V4, V5, V6).

Šest prekordijalnih kanala prate aktivnosti u vodoravnoj ravnini. U praksi to zahtijeva postavljanje šest elektroda oko trupa subjekta (Slika 1.4), a EKG predstavlja razliku između svake od tih elektroda i Wilsonovog središnjeg terminala.



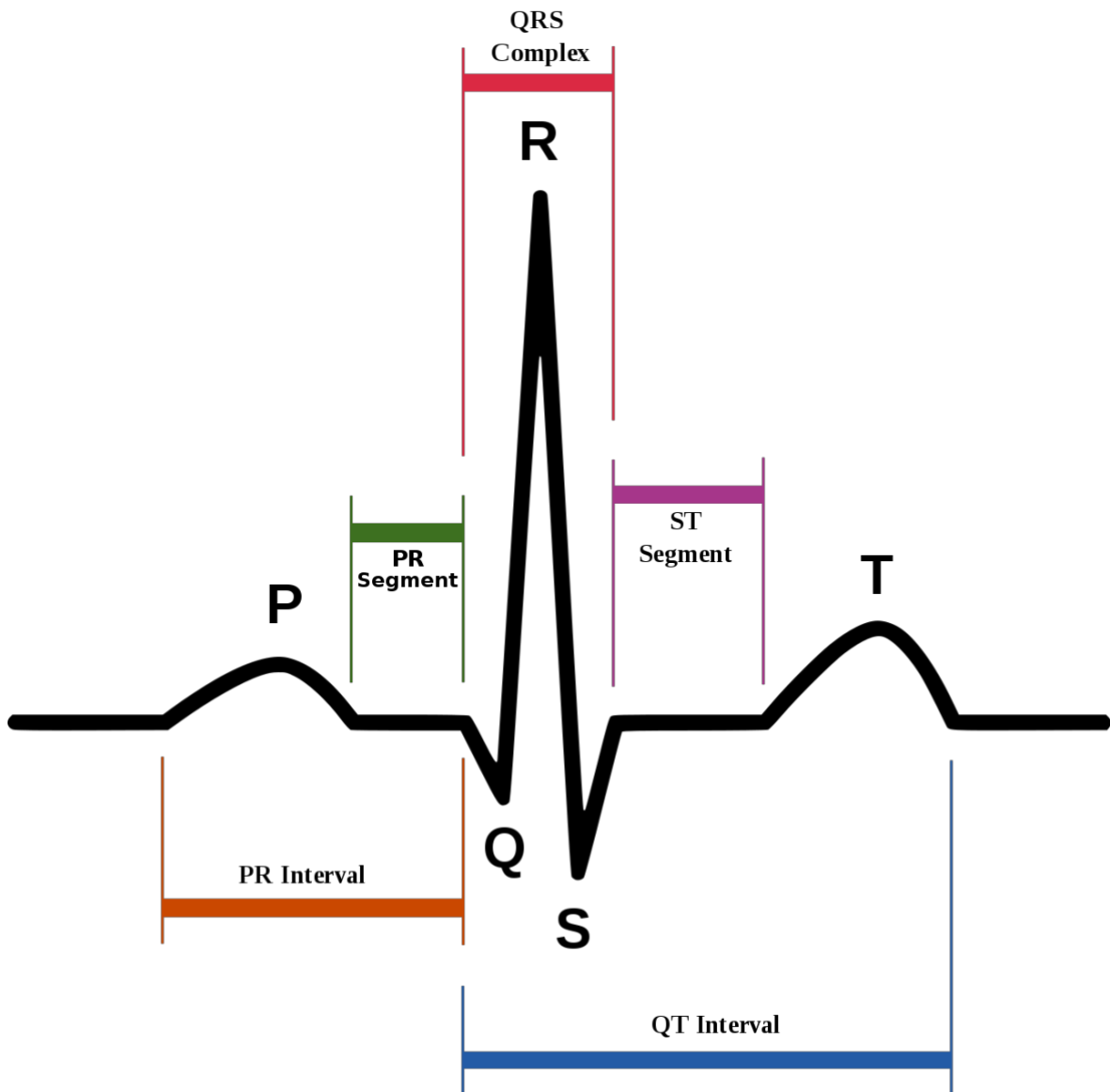
Slika 1.4 Položaj prekordijalnih elektroda [6]

Ostale četiri elektrode postavljaju se na lijevu ruku, desnu ruku, lijevu nogu i desnu nogu subjekta. Razlike potencijala između elektroda ekstremiteta i središnjeg terminala osnova su za ostala tri standardna kanala EKG-a. Kanal I predstavlja razliku između elektrode lijeve ruke(eng. LA) i elektrode desne ruke(eng. RA), kanal II je razlika između lijeve noge(eng. LL) i desne ruke, dok je kanal III razlika između lijeve noge i lijeve ruke.

Ostala tri kanala, koja započinju slovom „a“, predstavljaju razliku potencijala pojedinog ekstremiteta i prosjeka potencijala druga dva ekstremiteta. aVF je razlika između potencijala lijeve noge i prosjeka potencijala ruku, aVR je razlika između desne ruke i prosjeka lijeve noge i lijeve ruke, dok je aVL razlika između lijeve ruke i prosjeka desne ruke i lijeve noge.

1.2.2. Značajke elektrokardiograma

Slika 1.5 ilustrira normalne kliničke značajke EKG-a, koje uključuju valne amplitude i vremenske intervale valova.



Slika 1.5 Normalne značajke elektrokardiograma [6]

Položaji različitih valova na EKG-u označeni su slovima P, Q, R, S i T. Osim ovih, postoji i U val, iako je taj val često teško prepoznati, jer može biti odsutan, imati malu amplitudu ili biti prikriven naknadnim otkucajima.

P val predstavlja depolarizaciju pretkljetki koja se širi od desne pretkljetke do lijeve. P val je obično uspravan u većini kanala, osim kod aVR-a. Ako je P val neobično dugog trajanja, to može predstavljati povećanje atrija. Normalno trajanje P vala je manje od 80 ms.

PR interval označava se od početka P vala do početka QRS kompleksa. PR segment obično je potpuno ravan, ali može biti spušten kod perikarditisa. Očekivano trajanje PR intervala je između 120 i 200 ms.

QRS kompleks predstavlja brzu depolarizaciju desne i lijeve kljetke. Kljetke imaju veću mišićnu masu od pretkljetki, pa QRS kompleks obično ima mnogo veću amplitudu od P vala. Ako je QRS kompleks širok (dulji od 120 ms), to sugerira poremećaj srčanog provodnog sustava ili poremećaj ritmova kljetki poput ventrikularne tahikardije. Neobično visok QRS kompleks može predstavljati hipertrofiju lijeve kljetke, dok QRS kompleks vrlo male amplitude može predstavljati perikardijalni izljev. Normalno trajanje QRS kompleksa je između 80 i 100 ms.

ST segment predstavlja razdoblje kada su kljetke depolarizirane. On povezuje QRS kompleks i T val. Obično je ravnog oblika, ali može poprimiti depresivan ili povišen oblik, a tada ukazuje na srčani udar ili ishemiju.

T val predstavlja repolarizaciju kljetki. Obično je uspravan kod svih kanala, osim kod aVR-a i V1. Obrnuti T valovi mogu biti znak ishemije, hipertrofije lijeve kljetke, ili metaboličkih abnormalnosti. Visoki T valovi mogu biti znak hiperkalemije ili ranog srčanog udara.

2. Skup podataka PTB-XL

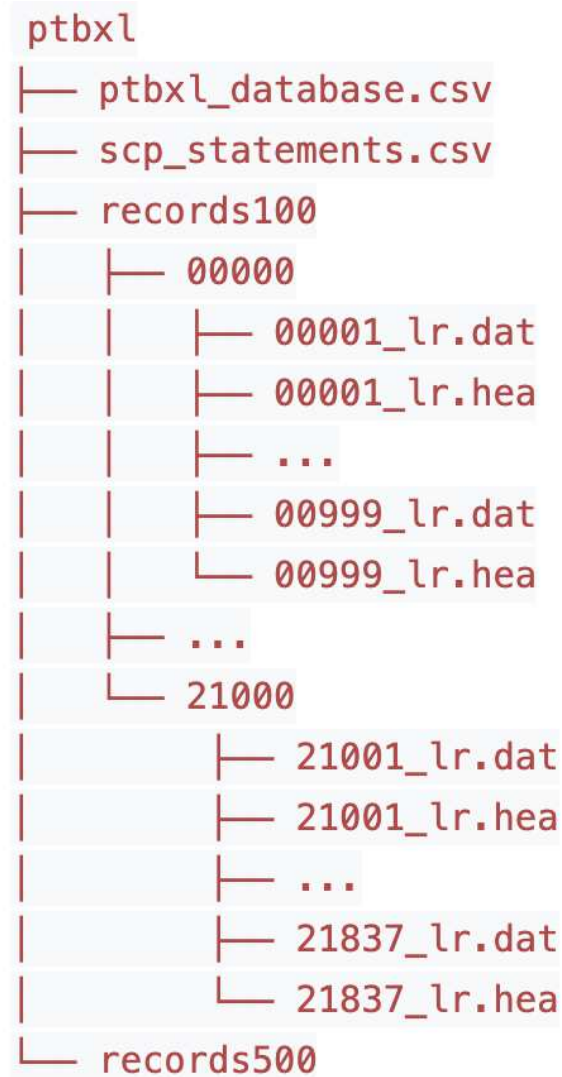
Neuronska mreža na kojoj se provodi eksperiment za ovaj završni rad uči na velikom, slobodno dostupnom skupu podataka PTB-XL koji sadrži razne zapise EKG-a. Ovo poglavlje opisuje taj skup podataka. Opis je uglavnom inspiriran službenom stranicom PTB-XL-a [2].

Skup EKG zapisa PTB-XL velik je skup podataka od 21837 kliničkih 12-kanalnih EKG-a od 18885 subjekata dužine 10 sekundi. Sirove podatke u valnom obliku bilježilo je do dva kardiologa, koji su svakom zapisu dodijelili potencijalno više EKG izjava. Skup podataka dolazi s opsežnim metapodacima od kojih su neki: demografski podaci, karakteristike infarkta, vjerojatnosti za dijagnostičke EKG izjave, kao i svojstva označenih signala.

Zapisi na kojima se temelji PTB-XL skup podataka prikupljeni su uređajima tvrtke Schiller AG tijekom gotovo sedam godina. Prikupljali su se između listopada 1989. godine i lipnja 1996. godine.

Sirovi signali zabilježeni su i pohranjeni u komprimiranom formatu. Za sve signale pruža se standardni skup od 12 kanala (I, II, III, aVL, aVR, aVF, V1, V2, V3, V4, V5, V6) s referentnim elektrodama na desnoj ruci. Uz svaki zapis dolaze i općeniti podaci o subjektima kao što su dob, spol, težina i visina. Svaki je zapis označen nizom izvještaja (generiran od strane kardiologa ili automatskom interpretacijom pomoću EKG-uređaja) koji je pretvoren u standardizirani skup SCP-ECG izjava (`scp_codes`). Velik dio zapisa potvrđen je i od drugog kardiologa, a sve zapise potvrdio je tehnički stručnjak koji se većinom fokusirao na karakteristike signala. EKG zapisi i subjekti identificiraju se jedinstvenim identifikatorima (`ecg_id` i `patient_id`).

Skup podataka organiziran je kao na Slika 2.1.



Slika 2.1 Organizacija skupa podataka

Skup podataka sadrži 21837 12-kanalnih EKG zapisa u trajanju od 10 sekundi od 18885 subjekata. 52% subjekata su muškarci, a 48% su žene. Raspon godina subjekata je od 0 (nula znači manje od 1) do 95 godina (medijan 62 i interkvartilni raspon 22).

Distribucija dijagnoza prikazana je na Slika 2.2. Zbog jednostavnosti dijagnoze su ograničene na iskaze skupljene u superklase. Zbroj dijagnoza premašuje broj zapisa zbog potencijalno više dijagnoza po zapisu.

#Records	Superclass	Description
9528	NORM	Normal ECG
5486	MI	Myocardial Infarction
5250	STTC	ST/T Change
4907	CD	Conduction Disturbance
2655	HYP	Hypertrophy

Slika 2.2 Distribucija dijagnoza u skupu podataka

Moguća je pojava pet glavnih dijagnoza: NORM – Normalni EKG, MI – Srčani udar (engl. *Myocardial Infarction*), promjene u ST segmentu (engl. STTC-ST/T *Change*), CD – Poremećaji provođenja (engl. *Conduction Disturbance*), HYP – Hipertrofija (engl. *Hypertrophy*) i sedamnaest njihovih najčešćih kombinacija.

Valni zapisi pohranjuju se u WaveForm DataBase (WFDB) formatu s 16-bitnom preciznošću i frekvencijom uzorkovanja od 500 Hz (records500/). Osim frekvencije od 500 Hz, svi zapisi su dostupni i u frekvenciji uzorkovanja od 100 Hz (records100/). Za ovaj završni rad korišteni su podaci s frekvencijom uzorkovanja od 100 Hz.

WaveForm Database ili WFDB je softver za pregled, analizu i stvaranje snimaka fizioloških signala, u koje spada i valni zapis EKG-a. [7]

Svi dostupni metapodaci pohranjuju se u datoteku ptb_xl_database.csv s jednim redom po zapisu kojeg identificira `ecg_id`. Sadrži 28 stupaca koji se mogu kategorizirati u:

1. Identifikatore: Svakom zapisu pripada jedinstven `ecg_id`. Odgovarajući subjekt identificira se preko `patient_id`-a. Putovi do izvornog zapisa (500 Hz) i prilagođenog zapisa (100 Hz) pohranjeni su u `filename_hr` i `filename_lr`.
2. Opće metapodatke: demografski podaci i metapodaci snimanja kao što su `dob`, `spol`, `visina`, `težina`, `pripadajuća medicinska sestra`, `mjesto`, `uređaj` i `datum_snimanja`
3. EKG izjava: temeljna komponenta je `scp_codes` (SCP-ECG izrazi kao rječnik s oblikom izjava: vjerojatnost, gdje je vjerojatnost postavljena na 0 ako je nepoznata) i

izvještaj (string izvještaja). Dodatna polja su: heart_axis, infarction_stadium1, infarction_stadium2, validated_by, second_opinion, initial_autogenerated_report i validated_by_human.

4. Metapodatke signala: kvaliteta signala poput šuma (static_noise i burst_noise), zanošenja osnovne linije (baseline_drift) i drugih informacija poput problema s elektrodama. Također postoji polje extra_beats za brojanje dodatnih sistola i pacemaker za uzorke signala koji ukazuju na aktivan pacemaker.

Slika 2.3 prikazuje jedan manji isječak iz datoteke ptb1_database.csv. Na njoj se vide neki od osnovnih metapodataka i početna 32 retka u skupu podataka.

ecg_id	patient_id	age	sex	height	weight	nurse	site	device	recording_date	report
1	15709.0	56.0	1	63.0	2.0	0.0	CS-12	E	1984-11-09 09:17:34	sinusrhythmus periphere niederspannung
2	13243.0	19.0	0	70.0	2.0	0.0	CS-12	E	1984-11-14 12:55:37	sinusbradykardie sonst normales ekg
3	20372.0	37.0	1	69.0	2.0	0.0	CS-12	E	1984-11-15 12:49:10	sinusrhythmus normales ekg
4	17014.0	24.0	0	82.0	2.0	0.0	CS-12	E	1984-11-15 13:44:57	sinusrhythmus normales ekg
5	17448.0	19.0	1	70.0	2.0	0.0	CS-12	E	1984-11-17 10:43:15	sinusrhythmus normales ekg
6	19005.0	18.0	1	58.0	2.0	0.0	CS-12	E	1984-11-28 13:32:13	sinusrhythmus normales ekg
7	16193.0	54.0	0	83.0	2.0	0.0	CS-12	E	1984-11-28 13:32:22	sinusrhythmus linkstyp t abnormal, wahrscheinlich inferiorer myokardschaden qt-verlängerung
8	11275.0	48.0	0	95.0	2.0	0.0	CS-12	E	1984-12-01 14:49:52	sinusrhythmus linkstyp qrs(t) abnormal inferiorer infarkt alter unbest.
9	18792.0	55.0	0	70.0	2.0	0.0	CS-12	E	1984-12-08 09:44:43	sinusrhythmus normales ekg
10	9456.0	22.0	1	56.0	2.0	0.0	CS-12	E	1984-12-12 14:12:46	sinusrhythmus normales ekg
11	11243.0	20.0	1	57.0	2.0	0.0	CS-12	E	1984-12-16 14:20:39	sinus arrhythmie sonst normales ekg
12	11031.0	43.0	1	44.0	2.0	0.0	CS-12	E	1984-12-21 08:02:26	sinusbradykardie sonst normales ekg
13	19953.0	58.0	1	54.0	2.0	0.0	CS-12	E	1985-01-05 12:13:05	sinusrhythmus normales ekg
14	12925.0	19.0	1	58.0	2.0	0.0	CS-12	E	1985-01-10 11:45:19	sinusrhythmus normales ekg
15	13375.0	17.0	1	67.0	2.0	0.0	CS-12	E	1985-01-12 10:39:41	sinus arrhythmie sonst normales ekg
16	10999.0	49.0	0	79.0	2.0	0.0	CS-12	E	1985-01-16 08:27:38	sinusrhythmus normales ekg
17	13619.0	56.0	0		2.0	0.0	CS-12	E	1985-01-20 14:25:25	vorhof:hf 280 tachykardes vorhofflimmern ueberdrehter linkstyp qrs(t) abnormal inferiorer infarkt
18	13619.0	56.0	0		2.0	0.0	CS-12	E	1985-01-20 15:17:24	vorhofflattern 2:1 Überleitung,jetzt nach isoptin 2amp+ 1 amp digitalis teilweise 3:1 vorhof:ca. hf:265
19	11116.0	20.0	0	85.0	2.0	0.0	CS-12	E	1985-01-23 11:23:00	sinusrhythmus p-verbreiterung
20	13619.0	56.0	0		2.0	0.0	CS-12	E	1985-01-23 12:55:32	supraventrikuläre ersatzstole(n) interponierte supraventrikuläre extrasystole(n) supraventrikuläre ta
21	17102.0	19.0	1	58.0	2.0	0.0	CS-12	E	1985-01-25 10:31:52	sinusrhythmus normales ekg
22	20978.0	17.0	1	53.0	2.0	0.0	CS-12	E	1985-01-25 12:33:18	sinusrhythmus linkstyp unspezifisches abnormales t
23	13619.0	56.0	0		2.0	0.0	CS-12	E	1985-01-25 12:49:40	2:1 Überleitung bei vorhofflattern,jetzt unter 3x1 cordichin + 1x1 novodigal 0,2 etwas langsamer hf v
24	9012.0	41.0	0	69.0	2.0	0.0	CS-12	E	1985-01-26 10:15:03	sinusrhythmus normales ekg
25	10962.0	19.0	1	64.0	2.0	0.0	CS-12	E	1985-01-26 11:15:05	sinusrhythmus normales ekg
26	13619.0	56.0	0		2.0	0.0	CS-12	E	1985-01-27 12:45:19	sinusrhythmus verdacht auf p-sinistocardiale linkstyp unspezifisches abnormales t qt-verlängerung :
27	10316.0	56.0	0	87.0	2.0	0.0	CS-12	E	1985-01-30 07:02:34	sinusrhythmus linkstyp sonst normales ekg
28	13619.0	56.0	0		2.0	0.0	CS-12	E	1985-02-01 12:58:55	sinusrhythmus verdacht auf p-sinistocardiale linkstyp unspezifisches abnormales t qt-verlängerung :
29	14340.0	18.0	1	47.0	2.0	0.0	CS-12	E	1985-02-01 14:33:55	sinusrhythmus normales ekg
30	19053.0	25.0	0	83.0	2.0	0.0	CS-12	E	1985-02-02 09:30:29	sinusrhythmus unspezifischer intraventrikulärer block linkshypertrophie
31	15348.0	24.0	1	58.0	2.0	0.0	CS-12	E	1985-02-03 13:06:23	sinusrhythmus normales ekg
32	11154.0	18.0	0	56.0	2.0	0.0	CS-12	E	1985-02-07 10:38:43	sinusrhythmus rechtstyp linksposterioer hemiblock

Slika 2.3 Isječak datoteke ptb1_database.csv

U nastavku prikazan je osnovni kod u programskom jeziku Python korišten za učitavanje podataka. Podaci su sa poveznice dohvaćeni u trenutni Jupyter Notebook.

```

url = 'https://storage.googleapis.com/ptb-xl-
1.0.1.physionet.org/ptb-xl-a-large-publicly-available-
electrocardiography-dataset-1.0.1.zip'
r = requests.get(url, allow_redirects=True)

open('baza.zip', 'wb').write(r.content)
from zipfile import ZipFile
with ZipFile('baza.zip', 'r') as zipObj:
    zipObj.extractall()

path = '/content/ptb-xl-a-large-publicly-available-
electrocardiography-dataset-1.0.1/'

Y = pd.read_csv(path+'ptbx1_database.csv',
index_col='ecg_id')

Y.scp_codes = Y.scp_codes.apply(lambda x:
ast.literal_eval(x)) #scp_codes pretvara iz stringa u dict

data = [wfdb.rdsamp(path+f) for f in Y.filename_lr]

```

Nakon učitavanja skup podataka podijeljen je na skupove za učenje, validaciju i testiranje.

3. Ispitivanje hiperparametara

Hiperparametri neuronske mreže su varijable koje utječu na strukturu mreže i zadaju način na koji mreža uči. Oni se zadaju prije procesa učenja, a budući da učenje zahtijeva određenu količinu vremena, problem optimizacije hiperparametara je vremenski skup.

U nastavku slijedi izvještaj ispitivanja raznih hiperparametara dubokih neuronskih mreža i analiza njihovih utjecaja na učinkovitost klasifikacije oboljenja iz zapisa EKG-a.

3.1. Korištene tehnologije

Eksperimenti su provedeni preko platforme Google Colab [8]. To je besplatna platforma koja nudi mogućnost izvođenja programskog koda na poslužiteljima kompanije Google. Ona je pogodna za sve znatiželjne programere i istraživače u području umjetne inteligencije.

Kod je pisan u programskom jeziku Python [9], koji je u današnje vrijeme najbolji odabir za provođenje projekata u području umjetne inteligencije, u što spadaju i duboke neuronske mreže. To je upravo zato što postoji platforma otvorenog koda TensorFlow[10] za strojno učenje dostupna za programski jezik Python. TensorFlow pruža mogućnost jednostavne izgradnje i učenja modela neuronskih mreža koristeći intuitivna programska sučelja visoke razine poput Kerasa [11].

Svi modeli dubokih neuronskih mreža i ispitivanje njihovih hiperparametara u ovom radu provedeni su preko programskog sučelja Keras. Ono nudi dosljedne i jednostavne API-je, minimalizira broj korisničkih radnji potrebnih za uobičajene obrasce uporabe i pruža jasne poruke o pogreškama. Također ima opsežnu dokumentaciju i vodiče za programere. [11]

Budući da ispitivanje utjecaja hiperparametra na točnost modela zahtijeva mnoge probe i pokušaje, ono je vremenski zahtijevan posao. Zbog toga je za automatiziranu provjeru hiperparametara i pokušaje optimizacije modela korišten radni okvir Optuna [12] dostupan u Pythonu. Optuna u svom radu koristi objekte. Najvažniji objekti su `optuna.Study` i `optuna.Trial`. Objekt `Study` predstavlja jedno istraživanje. On se jednostavno kreira i može se spremirati u datoteku za lagano ponovno korištenje i nastavak istraživanja. Kreiranje objekta `Study` prikazano je u nastavku.

```
study_name = "example" # Unique identifier of the study.
```

```
storage_name = "sqlite:///{}.db".format(study_name)
study = optuna.create_study(study_name=study_name,
storage=storage_name, direction='maximize')
```

„SQLite“ datoteka `example.db` automatski se inicijalizira novim zapisom objekta `Study`. Sada je iz te datoteke moguće učitati postojeće istraživanje za nastavak korištenja kao što je prikazano.

```
study_name = "example"
storage_name = "sqlite:///{}.db".format(study_name)
study = optuna.create_study(study_name=study_name,
storage=storage_name, load_if_exists=True)
```

Kako bi mogli koristiti Optunu, model neuronske mreže omata se u funkciju koja se naziva `objective`. Ta funkcija se onda predaje u funkciju `optimise` koja se poziva nad objektom `Study`. Primjer u nastavku pokazuje jednostavno korištenje Optune za minimiziranje matematičke funkcije.

```
def objective(trial):
    x = trial.suggest_float("x", -10, 10)
    return (x - 2) ** 2

study.optimize(objective, n_trials=100)
```

Na isti način se optimiraju hiperparametri neuronskih mreža ako se kreiranje modela omata u funkciju `objective`, a iz funkcije se vrati vrijednost koja se želi optimirati. Najčešće je to točnost modela na validacijskom skupu podataka (`val_accuracy`). Hiperparametre zadaje sam radni okvir pomoću funkcija koje se pozivaju nad objektom `Trial`. Neke od njih su `suggest_float()`, `suggest_uniform()`, `suggest_categorical()`... Kako bismo sami zadali parametre koje želimo isprobati, nad objektom `Study` zovemo funkciju `enqueue_trial()` kojoj predajemo željene vrijednosti hiperparamtera.

Većina grafova koji će biti prikazani iscrtani su pomoću biblioteke za Python Matplotlib ili pomoću vizualizacijskog sučelja koji nudi Optuna. Matplotlib je sveobuhvatna biblioteka za stvaranje statičkih, animiranih i interaktivnih vizualizacija u Pythonu [12].

3.2. Početna mreža

Neuronska mreža od koje kreće ispitivanje hiperparametara napravljena je na predmetu Projekt R akademske godine 2020./2021. Slika 3.1 prikazuje arhitekturu te mreže.

```
Model: "sequential"
Layer (type)                Output Shape                Param #
=====
conv1d (Conv1D)             (None, 989, 256)           37120
batch_normalization (Batch Normalization) (None, 989, 256)           1024
max_pooling1d (MaxPooling1D) (None, 494, 256)           0
conv1d_1 (Conv1D)           (None, 483, 128)           393344
batch_normalization_1 (Batch Normalization) (None, 483, 128)           512
max_pooling1d_1 (MaxPooling1D) (None, 241, 128)           0
conv1d_2 (Conv1D)           (None, 230, 128)           196736
batch_normalization_2 (Batch Normalization) (None, 230, 128)           512
max_pooling1d_2 (MaxPooling1D) (None, 115, 128)           0
lstm (LSTM)                 (None, 115, 200)           263200
lstm_1 (LSTM)               (None, 115, 200)           320800
lstm_2 (LSTM)               (None, 200)                 320800
dense (Dense)               (None, 23)                  4623
=====
Total params: 1,538,671
Trainable params: 1,537,647
Non-trainable params: 1,024
```

Slika 3.1 Prikaz početnog modela mreže

Mrežu čine tri konvolucijska sloja, gdje nakon svakog od njih slijedi batch normalization sloj te max pooling sloj. Nakon konvolucijskih slojeva slijede tri LSTM sloja. Nakon LSTM slojeva slijedi zadnji sloj koji je normalan gusti sloj s 23 izlaza od kojih svaki predstavlja jednu od mogućnosti klasifikacije.

Sljedeći kod je prikaz izgradnje modela u Kerasu.

```

model = keras.models.Sequential()
model.add(keras.layers.Conv1D(filters=256, kernel_size=12,
activation="relu", input_shape=(1000, 12)))
model.add(keras.layers.BatchNormalization())
model.add(keras.layers.MaxPool1D())
model.add(keras.layers.Conv1D(filters=128, kernel_size=12,
activation="relu"))
model.add(keras.layers.BatchNormalization())
model.add(keras.layers.MaxPool1D())
model.add(keras.layers.Conv1D(filters=128, kernel_size=12,
activation="relu"))
model.add(keras.layers.BatchNormalization())
model.add(keras.layers.MaxPool1D())
model.add(keras.layers.LSTM(200, return_sequences=True))
model.add(keras.layers.LSTM(200, return_sequences=True))
model.add(keras.layers.LSTM(200))
model.add(keras.layers.Dense(23, activation="softmax"))

```

Ova mreža imala je približnu točnost od 60% na validacijskom skupu podataka.

U nastavku ćemo razmotriti neke od njenih hiperparametara i analizirati njihov utjecaj na uspješnost klasifikacije oboljenja iz zapisa EKG-a.

3.3. Broj epoha i veličina grupe

Prva optimizacija koja je napravljena je odabir broja epoha kod učenja mreže i veličina serije (engl. *batch size*) koji se daje mreži prilikom svake epohe.

3.3.1. Broj epoha

Epoha je jedan prolaz svih podataka iz skupa za učenje kroz neuronsku mrežu. Broj epoha onda određuje koliko puta će neuronska mreža vidjeti cijeli skup za učenje.

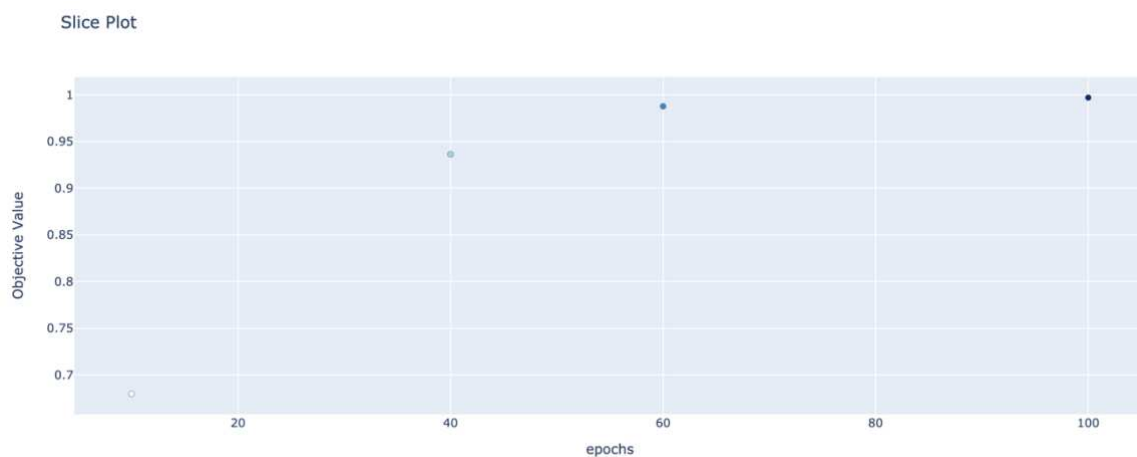
Čest problem kod dubokih neuronskih mreža je prenaučenosť. Prenaučenosť je problem koji se događa kada mreža dosegne skoro savršenu točnost na skupu za učenje, ali nema ni približno sličnu točnost na skupu za validaciju. To znači da se mreža potpuno prilagodila podacima na kojima je učila, a daje loše rezultate na podacima koji joj nisu poznati.

Prenaučen model mreže je kao takav beskoristan. LSTM slojevi, koji su sastavni dio ove neuronske mreže, općenito su skloni prenaučnosti.

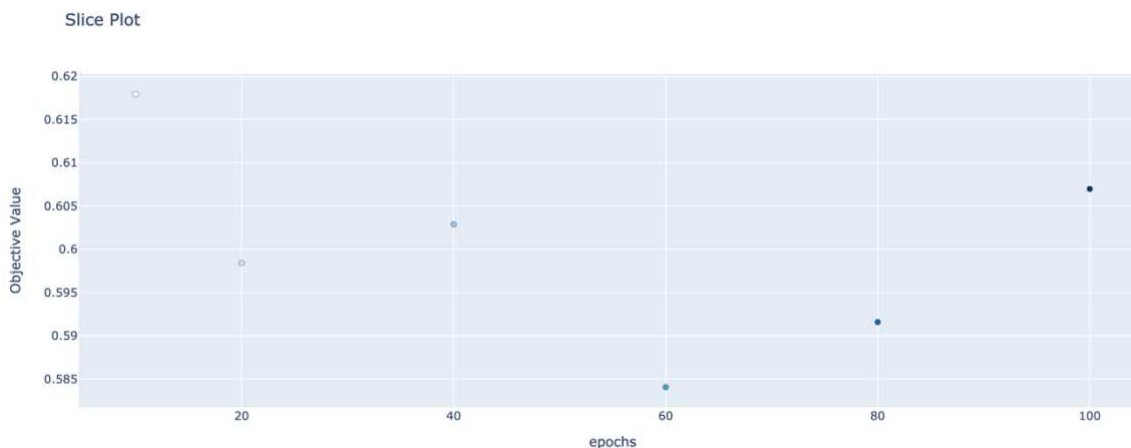
Broj epoha također utječe na prenaučnost. Iz tog razloga želimo postepeno povećavati broj epoha i pratiti do kada će točnost na skupu za validaciju biti slična točnosti na skupu za učenje. Kad točnost na skupu za validaciju počne stagnirati ili čak opadati, a točnost na skupu za učenje i dalje raste, tada želimo prestati povećavati broj epoha, kako bi smo izbjegli prenaučnost.

Početna mreža imala je broj epoha postavljen na 100. U ovom eksperimentu krenut ćemo sa manjim brojem epoha i pratiti točnosti na skupu za učenje i validaciju. U Kerasu točnost na skupu za učenje spremljena je u varijablu `accuracy`, a točnost na skupu za validaciju u varijablu `val_accuracy`. Brojevi epoha koji su ispitani su 10, 20, 40, 60, 80 i 100.

Slika 3.2 prikazuje točnost mreže na skupu za učenje, dok Slika 3.3 prikazuje točnost na skupu za validaciju.



Slika 3.2 Točnost na skupu za učenje



Slika 3.3 Točnost na skupu za validaciju

Na grafovima se vidi da već nakon 40 epoha točnost na skupu za učenje je skoro 95%, dok je točnost na skupu za validaciju približno 60%. Također se čini da broj epoha ne utječe puno na točnost validacije, jer je ona uvijek približno 60%. Za sada se čini da bi bilo dobro odabrati 40 kao broj epoha, ali uz uvjet da riješimo problem prenaučivosti nekim drugim hiperparametrom.

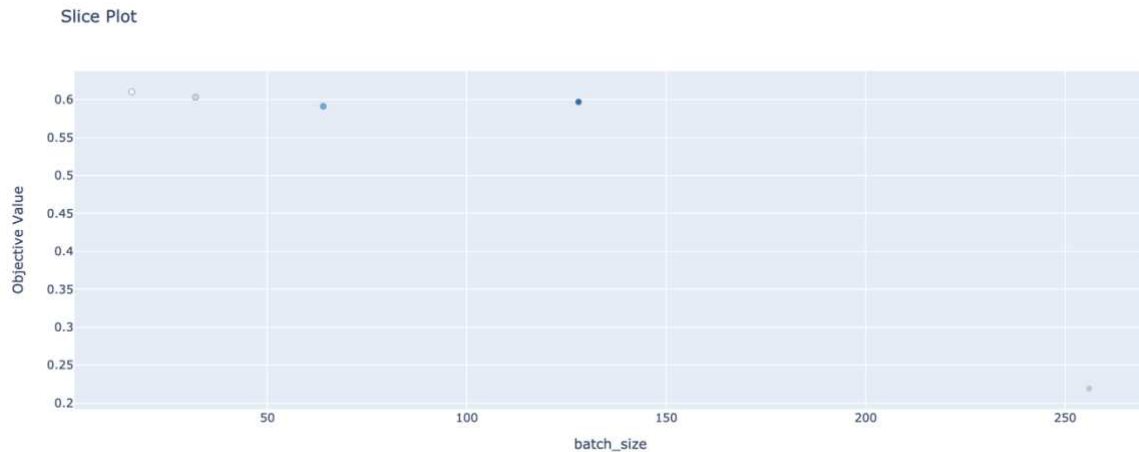
3.3.2. Veličina grupe

Veličina grupe (eng. *batch size*) je broj poduzoraka danih mreži nakon čega se događa ažuriranje parametara. Parametri su koeficijenti modela koje algoritam tijekom učenja sam optimira. Parametri mreže su srž njenog djelovanja. Model nakon učenja sadrži niz parametara koji minimiziraju pogrešku predikcije.

Ako je veličina grupe jednaka 1, parametri će se ažurirati nakon svakog zapisa EKG-a koji je dan mreži u trenutnoj epohi. Ako je veličina grupe jednaka veličini cijelog skupa podataka, parametri će se ažurirati tek nakon što svi zapisi EKG-a prođu kroz mrežu. Što je veličina grupe manja, to će učenje mreže zahtijevati više vremena, jer će biti potrebno više iteracija i više ažuriranja parametara neuronske mreže.

Pokazalo se dobrom praksom da veličina grupe bude mala, kao na primjer 16 ili 32. Početna mreža imala je veličinu grupe zadanu na 16. U ovom eksperimentu isprobane veličine grupe su 16, 32, 64, 128 i 256. Kreće se od manjih veličina grupe i prati se točnost na skupu za validaciju.

Slika 3.4 prikazuje točnost mreže na skupu za validaciju. Iz grafa se vidi da se većom veličinom grupe ne dobiva bolja točnost. Zato kao veličinu serije odabiremo početnu veličinu koja je bila 16.



Slika 3.4 Točnost na skupu za validaciju

3.4. Isključivanje čvorova grafa

Pojedinačni model neuronske mreže može se koristiti za simulaciju velikog broja različitih mrežnih arhitektura slučajnim isključivanjem čvorova tijekom učenja. To se naziva isključivanje čvorova grafa (eng. *dropout*) i nudi računski vrlo jeftinu i izuzetno učinkovitu metodu regularizacije kako bi se smanjilo prekomjerno prilagođavanje podacima i smanjila pogreška generalizacije u dubokim neuronskim mrežama svih vrsta. Ispuštanjem čvora mislimo na njegovo privremeno uklanjanje iz mreže, zajedno sa svim dolaznim i odlaznim vezama. [13]

Uvodimo novi hiperparametar koji određuje vjerojatnost isključivanja izlaza sloja. Ispadanje je jedan od najvažnijih hiperparametara kada se želimo riješiti prenaučivosti. Budući da LSTM slojevi imaju taj problem, isključivanje ćemo dodati kao njihov hiperparametar.

Pratit ćemo što se događa s funkcijama gubitka i točnostima na skupu za učenje i skupu za validaciju. Početi ćemo od vjerojatnosti ispadanja 0, pa ćemo ju postepeno povećavati. Isprobane vjerojatnosti su 0, 0.2, 0.4, 0.6, 0.8. Rezultati slijede u Tablica 1.

Tablica 1 Utjecaj vjerojatnosti ispadanja na točnost

Vjerojatnost isključivanja	Točnost validacijskog skupa
0	0.5809979438781738
0.2	0.6021872758865356
0.4	0.633287787437439
0.6	0.6390977501869202
0.8	0.6630212068557739

Vidi se da veća vjerojatnost isključivanja čvorova pogoduje većoj točnosti mreže. Uvjerimo se još da smo uspjeli smanjiti prenačenost. Slika 3.5 prikazuje funkcije gubitka za različite vrijednosti vjerojatnosti ispadanja. Slika 3.6 prikazuje točnosti na skupu za učenje u usporedbi sa točnosti na skupu za validaciju. Vidi se da se sa većom vjerojatnošću isključivanja lako postiže da točnost na skupu za učenje ne dolazi skoro do savršenih 100% kao što je to slučaj bez isključivanja. Najboljom se pokazala vrijednost 0.8, pa će se ta vrijednost koristiti u budućim eksperimentima. Sada su točnosti u odnosu u kojem bi trebale biti. Točnost na skupu za učenje veća je od točnosti na skupu za validaciju, ali više ne dolazi do prenaučivosti nakon malog broja epoha.

Budući da smo u mrežu dodali hiperparametar ispadanja, njoj je sada potreban veći broj epoha da bi postigla optimalnu točnost. Iz tog razloga broj epoha povećan je na 100 i dodan je povratni poziv (eng. *callback*) ranijeg zaustavljanja koji nudi Keras pri učenju modela. Time se postiže da učenje modela prestaje nakon određenog broja epoha ako se točnost modela nije poboljšala.

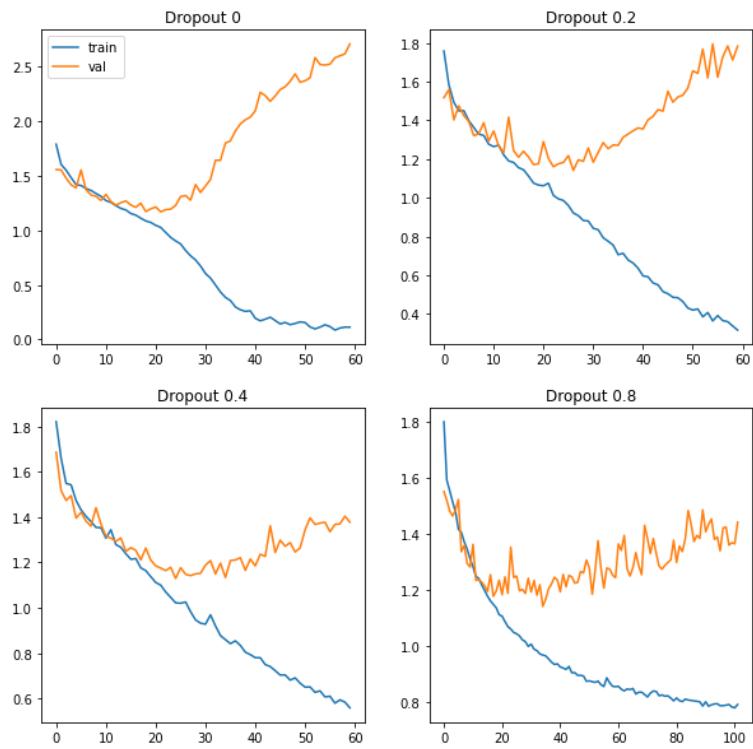
```

earlystopping = keras.callbacks.EarlyStopping(
    monitor = "val_accuracy",
    mode = "max", patience = 30,
    restore_best_weights = True)

```

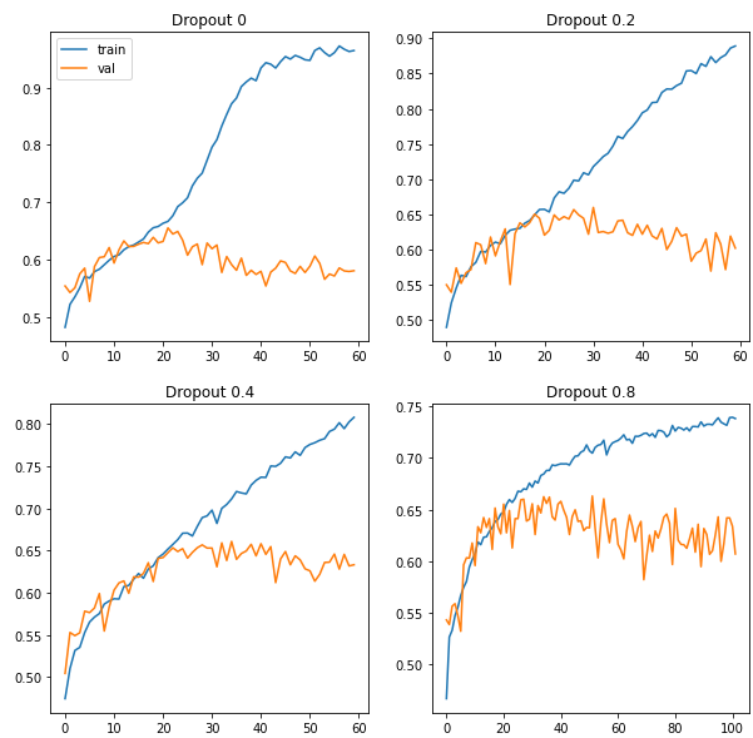
Učenje mreže će se zaustaviti ako ne bude napretka u rasponu od 30 epoha. Time ćemo malo uštedjeti na vremenu.

Train and Validation loss by epochs with different dropout values



Slika 3.5 Funkcije gubitka za različita ispadanja

Train and Validation accuracy by epochs with different dropout values



Slika 3.6 Točnosti za različita ispadanja

3.5. Hiperparametri konvolucijskog sloja

U ovom dijelu razmatra se utjecaj hiperparametara konvolucijskog sloja na točnost klasifikacije oboljenja. Ispitani hiperparametri su veličina filtera (engl. *kernel size*), broj filtera, aktivacijske funkcije te broj konvolucijskih slojeva.

Konvolucijski sloj koristi takozvane filtere ili jezgre s kojima skenira i prolazi po ulaznim podacima. Taj sloj kreira izlaz koji se naziva karta značajki (engl. *feature map*). Ovakvi slojevi prikladni su za korištenje kad ulazni podaci imaju topologiju rešetke. Konvolucijski slojevi razlikuju se po dimenzionalnosti podataka koje primaju na ulazu. Tako postoje konvolucijski slojevi jedne dimenzije, dvije dimenzije i tri dimenzije. U početnom modelu opisanom u poglavlju 3, korišteni su konvolucijski slojevi jedne dimenzije.

Neki od hiperparametara takvog sloja su broj jezgri, širina jezgre, aktivacijska funkcija i broj koraka (engl. *strides*). Broj koraka označava za koliko rešetki će se jezgra pomaknuti nakon svake operacije. Korišteni broj koraka u početnom modelu je 1. Širina jezgre postavljena je na 12, a broj jezgri svih slojeva osim prvog je 128. Prvi sloj ima 256 jezgri.

Svakom konvolucijskom sloju moguće je pridijeliti aktivacijsku funkciju. Aktivacijske funkcije označavaju na koji način slojevi neuronske mreže ulaze pretvaraju u izlaze. U današnje vrijeme najpoznatija aktivacijska funkcija je ReLU (engl. *rectified linear activation function*). Postoji još aktivacijskih funkcija od kojih su neke tanh (tangens hiperbolni) i sigmoida.

U nastavku se promatra kako navedeni hiperparametri utječu na točnost na skupu za validaciju.

Prvi na redu je broj jezgri. Tablica 2 prikazuje odnos broja jezgri i točnost na skupu za validaciju.

Tablica 2 Utjecaj broja jezgri na točnost

Broj jezgri	Točnost na skupu za validaciju
1	0.468557745218277
8	0.6322624683380127
16	0.6582365036010742
64	0.6606288552284241

128	0.6630212068557739
256	0.6657552719116211
512	0.6551606059074402
1024	0.6578947305679321

Vidi se da je idealan broj jezgri za ovaj problem 128 ili 256. Povećanjem broja jezgri dobiva se veća točnost, sve dok kod 256 jezgri točnost počne stagnirati. Budući da veći broj jezgri zahtijeva više vremena za učenje mreže, uzima se 128 kao optimalan broj.

Tablica 3 prikazuje odnos širine jezgre i točnosti na skupu za validaciju.

Tablica 3 Odnos širine jezgre i točnosti

Širina jezgre	Točnost na skupu za validaciju
1	0.6449077129364014
2	0.6452494859695435
3	0.6558441519737244
5	0.6534518003463745
8	0.6507176756858826
12	0.6630212068557739
24	0.6349965929985046
36	0.630553662776947
48	0.622351348400116
60	0.6062884330749512
80	0.6028708219528198

Iz tablice se vidi da je najbolji izbor za širinu jezgre 12. Vidi se da s većom jezgrom od 12 točnost polako počinje opadati.

Sljedeće slike prikazuju odnose nekih od najboljih brojeva jezgri i veličina jezgre.

Tablica 4 prikazuje odnos aktivacijske funkcije konvolucijskog sloja i točnosti na skupu za validaciju.

Tablica 4 Odnos aktivacijske funkcije i točnosti

Aktivacijska funkcija	Točnost na skupu za validaciju
ReLU	0.6630212068557739
Tanh	0.6161996126174927

ReLU se pokazuje kao bolja aktivacijska funkcija, što je i bilo očekivano s obzirom da je ona najčešće korištena u konvolucijskim slojevima. Zbog toga i dalje će se koristiti ReLU.

Za kraj ostaje provjeriti kako broj konvolucijskih slojeva utječe na točnost na skupu za validaciju. Poslije svakog sloja konvolucije slijedit će sloj normalizacije te sloj maksimalnog sažimanja (engl. *max pooling*).

Normalizacijski sloj (engl. *batch normalization*) koristi se za stabilizaciju i ubrzanje procesa učenja, to se postiže transformacijom koja održava srednju aktivaciju blizu nule, a standardno odstupanje aktivacije blizu jedinice. Jednodimenzionalan sloj sažimanja nastoji smanjiti senzibilitet izlazne mape naučenih značajki na način da sažme prosječnu prisutnost značajke u promatranoj sekvenci podataka kako bi mreža bila otpornija na neispravne podatke. [15]

Tablica 5 prikazuje odnos broja slojeva i točnosti na skupu za validaciju.

Tablica 5 Odnos broja konvolucijskih slojeva i točnosti

Broj konvolucijskih slojeva	Točnost na skupu za validaciju
1	0.6319206953048706
2	0.6353383660316467
3	0.6630212068557739
4	0.663287082195282

5	0.6531100273132324
6	0.6404647827148438

Četiri konvolucijska sloja pokazuju najbolju točnost, za više slojeva od četiri točnost počinje opadati. Odabire se broj 4 kao optimalan broj slojeva konvolucije.

3.6. Hiperparametri LSTM sloja

U ovom dijelu razmatra se utjecaj hiperparametara *Long Short Term Memory* sloja na učinkovitost klasifikacije oboljenja. Hiperparametri koji su ispitani su isključivanje čvorova, broj jedinica LSTM sloja te broj LSTM slojeva.

LSTM sloj spada u vrstu rekurentnih slojeva. Rekurentni slojevi omogućuju mreži da pamti stanja između više korištenja. Takvi slojevi sadrže veze između neurona u samom sloju. Po tome se razlikuju od običnih slojeva. Jedna od osnovnih komponenti LSTM sloja je memorijska ćelija. Memorijska ćelija sadrži ključne informacije koje je naučila tijekom vremena, a mreža je stvorena da učinkovito održava korisne informacije u memorijskoj ćeliji tijekom mnogih vremenskih koraka. U svakom vremenskom koraku, kad do sloja dođu podaci, LSTM sloj modificira memorijsku ćeliju s novim informacijama. Izlaz iz LSTM sloja je funkcija ulaza, prošlog izlaza te stanja memorijske ćelije. [14]

LSTM ćelija može sadržavati više jedinica.

Početna mreža sadrži 3 LSTM sloja svaki s 200 jedinica. U poglavlju 4.4. u svaki LSTM sloj dodano je isključivanje čvorova s vjerojatnošću 0.8.

Tablica 6 prikazuje odnos broja jedinica LSTM slojeva i točnost na skupu za validaciju.

Tablica 6 Odnos broja jedinica LSTM sloja i točnosti

Broj jedinica LSTM sloja	Točnost na skupu za validaciju
10	0.5475051403045654
20	0.6582365036010742
50	0.657211184501648

100	0.6568694710731506
200	0.6630212068557739
300	0.649350643157959
400	0.6558441519737244
500	0.6582365036010742
600	0.649350643157959
800	0.6520847678184509

Čini se da broj jedinica LSTM slojeva ne utječe puno na točnost modela. Ipak, 200 jedinica dalo je najbolji rezultat pa se taj broj odabire kao broj jedinica.

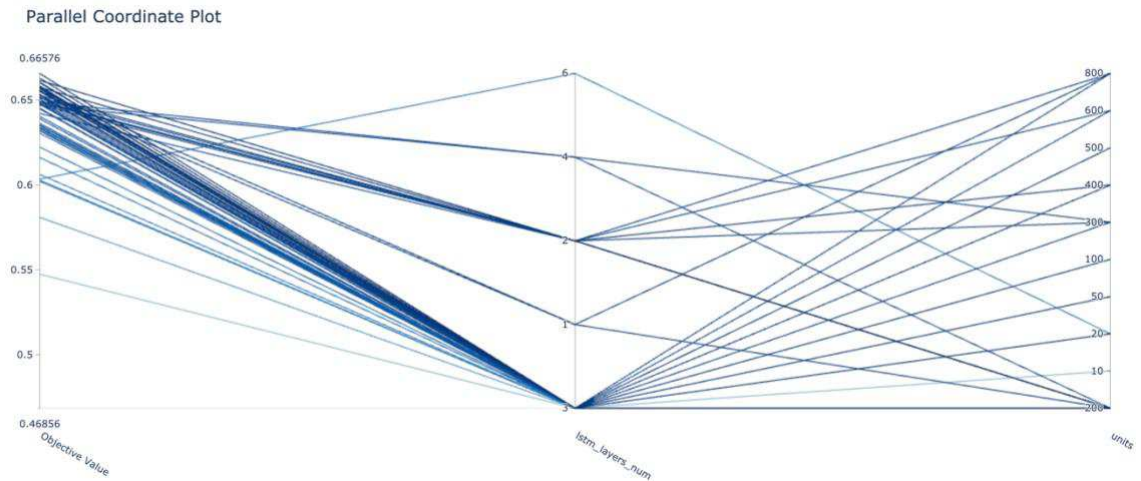
Tablica 7 prikazuje utjecaj različitog broja LSTM slojeva na točnost na skupu za validaciju.

Tablica 7 Utjecaj broja LSTM slojeva na točnost

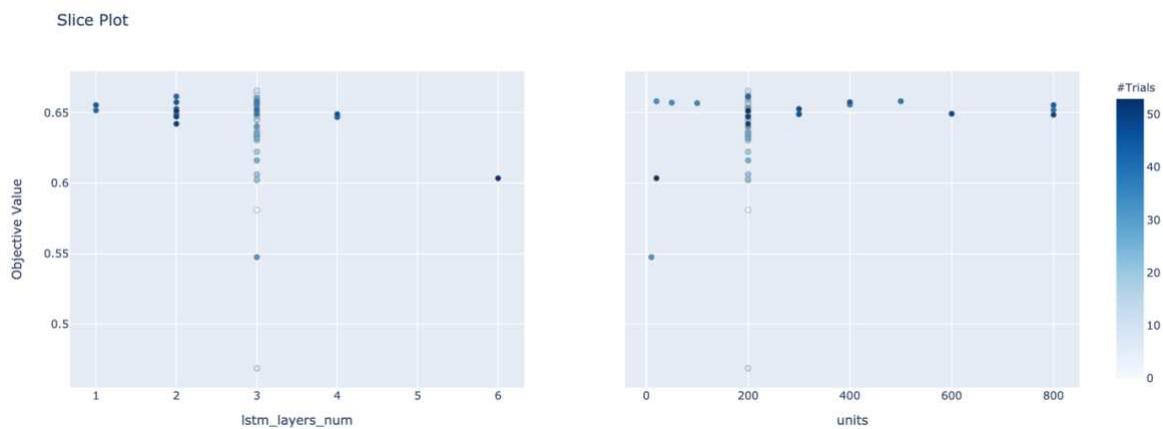
Broj LSTM slojeva	Točnost na skupu za validaciju
1	0.6517429947853088
2	0.6616541147232056
3	0.6630212068557739
4	0.6469582915306091

Odabiremo tri kao optimalni broj LSTM slojeva, jer se većim brojem slojeva ne postiže bolja točnost.

Slika 3.7 prikazuje graf paralelnih koordinata, a Slika 3.8 prikazuje odrezani graf za broj jedinica u LSTM sloju i broj slojeva. Grafovi prikazuju svih 54 pokušaja optimizacije koji su zabilježeni u Optuna istraživanju. Ovo konkretno Optuna istraživanje bilježilo je učenja mreže te rezultate dobivenog modela za sve isprobane hiperparametre konvolucijskog sloja te LSTM sloja. U svrhu lakšeg praćenja utjecaja hiperparametara LSTM sloja na točnost, samo ti hiperparametri prikazani su na grafovima. Ovi grafovi potvrđuju da tri LSTM sloja s 200 jedinica daju najbolju točnost.



Slika 3.7 Graf paralelnih koordinata



Slika 3.8 Odrezani graf

3.7. Hiperparametri algoritma optimizacije

Algoritmi optimizacije važan su dio učenja svake duboke neuronske mreže. Oni određuju na koji način se parametri mreže ažuriraju nakon svake primljene serije podataka. Njihov cilj je minimiziranje funkcije gubitka, a za to koriste vektor koji se naziva gradijent. Gradijent je vektor koji pokazuje u smjeru najvećeg spusta funkcije gubitka.

Opisi raznih algoritama optimizacije uglavnom su preuzeti iz [15].

Za svako učenje koje se provodi nad mrežom, mora biti zadan algoritam optimizacije. U Kerasu je to objekt koji se naziva `optimizer`. Do sad se u eksperimentima koristio Adam kao algoritam optimizacije.

Temeljni optimizacijski algoritam je algoritam stohastičkog gradijentnog spusta (engl. *Stochastic Gradient Descent* – SGD). Njegovi hiperparametri su stopa učenja (engl. *learning rate*) i moment.

Pravilo ažuriranja za parametar w s gradijentom g bez momenta [16]:

$$w = w - \text{learning_rate} * g$$

Pravilo ažuriranja s momentom većim od 0 [16]:

$$\begin{aligned} \text{velocity} &= \text{momentum} * \text{velocity} - \text{learning_rate} * g \\ w &= w + \text{velocity} \end{aligned}$$

Pravilo ažuriranja s Nesterovim momentom [16]:

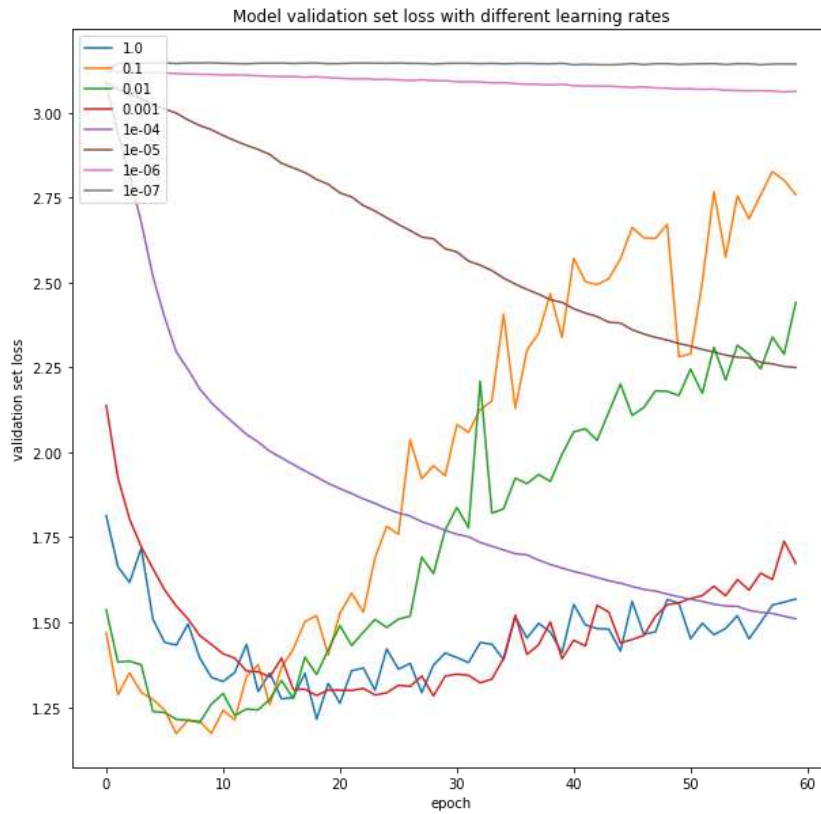
$$\begin{aligned} \text{velocity} &= \text{momentum} * \text{velocity} - \text{learning_rate} * g \\ w &= w + \text{momentum} * \text{velocity} - \text{learning_rate} * g \end{aligned}$$

Najčešća zadana vrijednost stope učenja za stohastički gradijentni spust je 0.01. U ovom eksperimentu ispitane su različite vrijednosti stope učenja, a praćene su funkcija gubitka te točnost na skupu za validaciju. Slika 3.9 prikazuje utjecaj stope učenja na funkciju gubitka na skupu za validaciju, dok Slika 3.10 prikazuje odnos stope učenja i točnosti na skupu za validaciju.

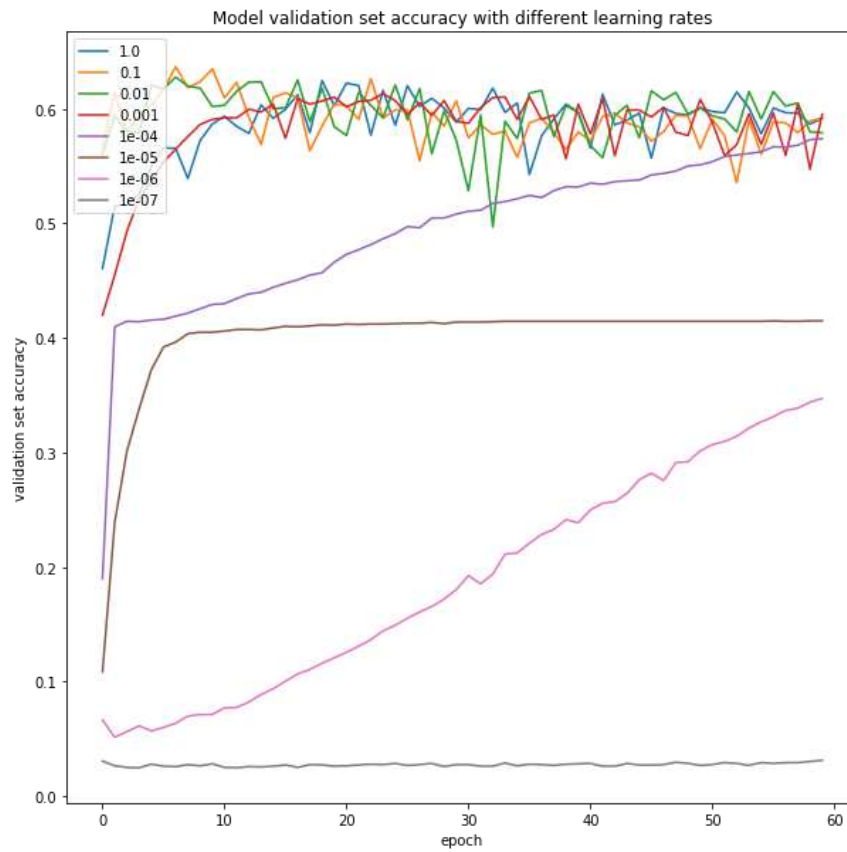
Po grafovima zaključujemo da model zaista sporije uči što je stopa učenja manja.

Prevelika stopa učenja može izazvati velike oscilacije, nestabilan postupak učenja, a možda čak i divergenciju. Za male stope, postupak učenja napreduje sporo, ali dovodi do optimalnog rješenja. Za premale stope, učenje može zaglaviti vrlo rano u lošem lokalnom minimumu, što se vidi i s grafova.

Pokazuje se da su idealne stope učenja za ovaj model upravo najčešće zadane vrijednosti poput 0.01 ili 0.001.



Slika 3.9 Odnos stope učenja i funkcije gubitka



Slika 3.10 Odnos stope učenja i točnosti

Osim stohastičkog gradijentnog spusta postoje i noviji algoritmi optimizacije koji koriste adaptivne stope učenja. Neki od najpoznatijih su AdaGrad, RMSProp i Adam.

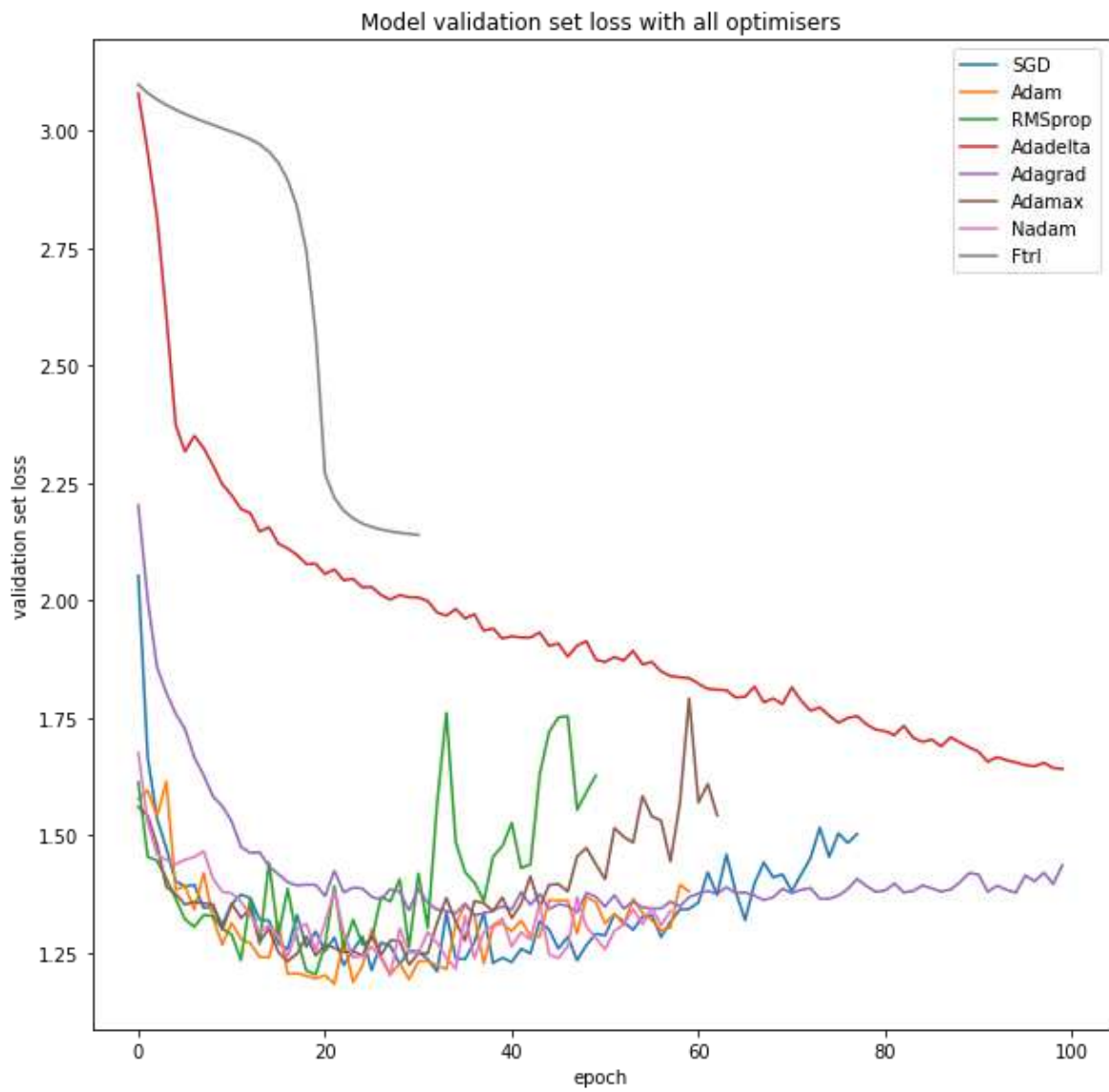
AdaGrad koristi stope učenja specifične za svaki parametar. On adaptira stope učenja parametara skalirajući ih obrnuto proporcionalno kvadratnom korijenu sume svih kvadriranih vrijednosti pripadne parcijalne derivacije funkcije gubitka kroz povijest. Parametri koji imaju velike parcijalne derivacije brzo će početi koristiti male stope učenja, dok parametri koji imaju male parcijalne derivacije dugo vremena će koristiti razumno velike stope učenja. Skaliranje stope je konzistentno sve jače i jače, što znači da nema mogućnosti oporavka nakon skaliranja.

RMSProp mijenja akumuliranje gradijenata tako da dodaje eksponencijalno prigušenje starih vrijednosti, pa onda novije vrijednosti imaju veći utjecaj. Ako komponenta gradijenta u nekom trenutku padne, relativno brzo će pasti i njeno prigušenje pa će se korekcije parametara ponovno povećati.

Adam (od *Adaptive Moments*) prati prosjek kretanja gradijenta i kvadrata gradijenta uz eksponencijalno zaboravljanje. Moment je implicitno ugrađen u praćenje kretanja gradijenta. Korekcija parametara se radi prema uprosječenom gradijentu a ne izračunatom. Osnovna inačica Adama je dosta robusna implementacija koja radi sa širokim skupom parametara.

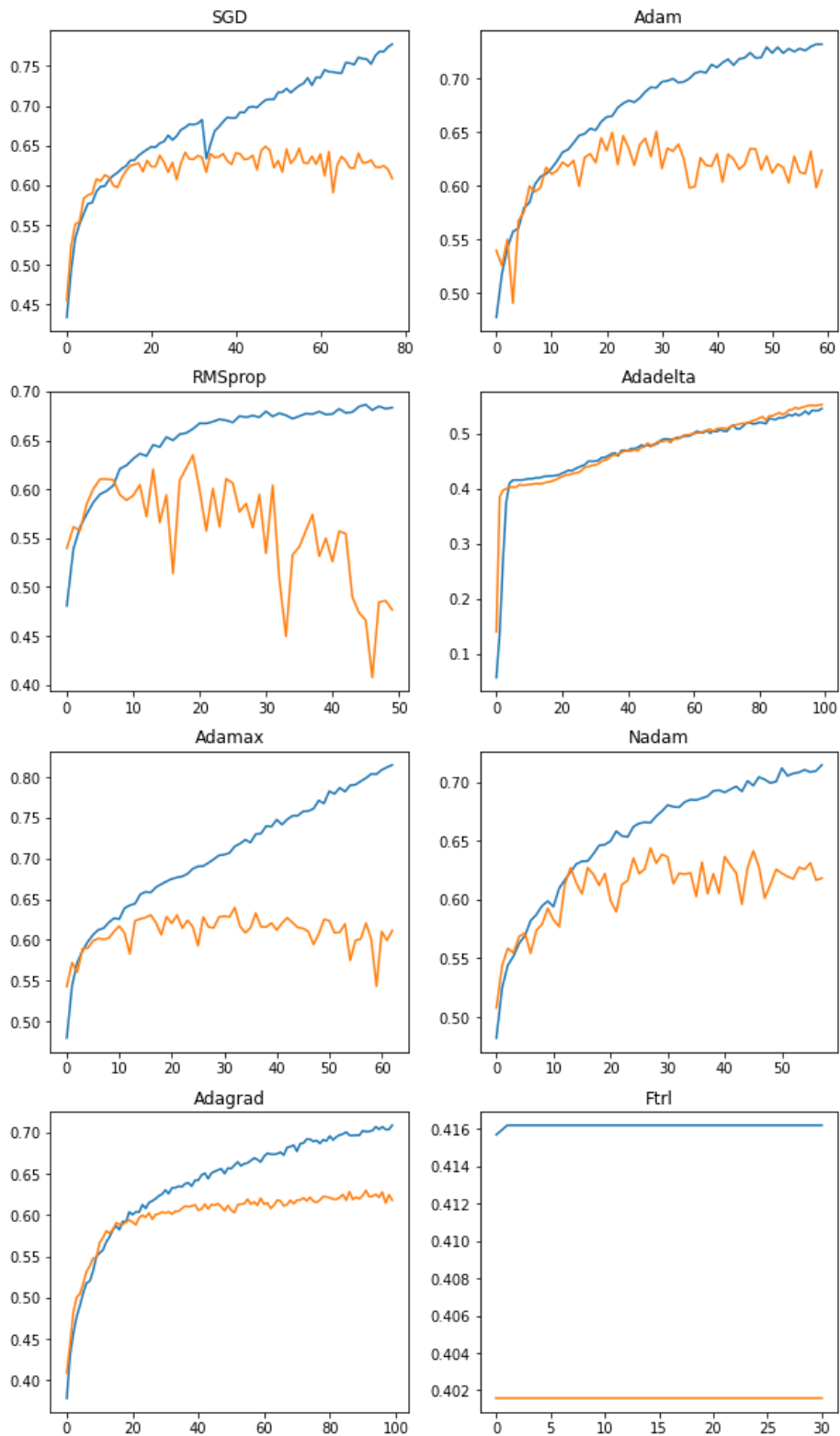
Osim osnovne inačice Adama, postoje i razne modifikacije tog algoritma. Neke od njih su Adamax i Nadam. Kod Adamaxa drugi moment ne prati kvadrat članova gradijenta već maksimume. Nadam je kombinacija Adama i Nesterovog momenta.

U ovom eksperimentu praćen je utjecaj različitih algoritama optimizacije na funkciju gubitka te točnosti na skupovima za učenje i validaciju. Isprobani su svi algoritmi optimizacije dostupni u Kerasu. Slika 3.11 prikazuje odnos različitih algoritama optimizacije i funkcije gubitka na skupu za validaciju. Slika 3.12 prikazuje grafove koji opisuju utjecaj algoritama optimizacije na točnost na skupu za validaciju, te točnost na skupu za učenje. Točnost na skupu za validaciju prikazana je žutom bojom, a točnost na skupu za učenje plavom.



Slika 3.11 Odnos optimizacijskih algoritama i funkcije gubitka

Train and Validation accuracy by epochs



Slika 3.12 Odnos algoritama optimizacije i točnosti

Na grafovima vide se razlike u ponašanju funkcije gubitka i točnosti na skupovima za učenje i validaciju tijekom učenja po broju epoha. Svaki od ovih algoritama provodi ažuriranje parametara modela na drugačiji način, a to onda utječe i na točnost klasifikacije oboljenja.

Tablica 8 prikazuje točnosti na skupu za validaciju za različite algoritme optimizacije.

Tablica 8 Točnosti na skupu za validaciju za razne optimizatore

Algoritam optimizacije	Točnost na skupu za validaciju
SGD	0.6490088701248169
Adam	0.6507176756858826
RMSProp	0.6349965929985046
Adadelata	0.5533151030540466
Adagrad	0.6179084181785583
Adamax	0.6397812962532043
Nadam	0.6438824534416199
Ftrl	0.4015721082687378

Kao najbolji algoritmi pokazali su se SGD, Adam, Nadam i Adamax. Odabire se Adam kao najbolji algoritam optimizacije za ovaj problem, koji je i inače najbolji izbor za većinu problema kojima se bave duboke neuronske mreže.

4. Aplikacija u Pythonu

Osim analize utjecaja hiperparametara dubokih neuronskih mreža na učinkovitost klasifikacije zapisa EKG-a, kao rezultat ovog završnog rada izrađena je i jednostavna aplikacija u Programskom jeziku Python. Ona omogućava učivanje zapisa EKG-a, njihovo učenje ili testiranje te ispis rezultata.

Sustav je napravljen kao lokalna konzolna aplikacija u programskom jeziku Python. Nudi korisniku da klasifikira oboljenja temeljem unaprijed ugrađenog ili vlastitog modela mreže koji unese u program i njegovog nalaza EKG-a. Korisnik također može odabrati učenje mreže odabirom mreže koju želi naučiti i datoteka sa zapisima.

Nakon pokretanja aplikacije korisnik treba upisati relativni put do modela neuronske mreže (koji mora biti pohranjen u .h5 formatu) koju želi koristiti u aplikaciji. Zatim će ući u petlju koja će ga svaki puta pitati želi li klasificirati bolest, učiti mrežu ili izaći iz aplikacije.

Ako odabere klasifikaciju bolesti, aplikacija će ga pitati za put do zapisa EKG-a u WFDB .dat formatu. Nakon što unese put do zapisa aplikacija će provesti klasifikaciju te obavijestiti korisnika o rezultatu klasifikacije te vjerojatnosti za svaku moguću bolest, čime će korisnik imati jasni uvid u oboljenje koje je najizglednije. Aplikacija će također nacrtati graf na temelju vjerojatnosti kako bi korisnik lakše vizualizirao rezultat.

Slika 4.1 prikazuje isječak koda koji služi za klasifikaciju bolesti iz danog zapisa EKG-a.

```
def classify(model):
    print("Classifying with model:")
    model.build()
    model.summary()
    pathToEKG = input_data("Please enter the path to the ECG WFDB .dat file you wish to classify: ", False)
    if(pathToEKG[-4:]!=".dat"):
        pathToEKG=pathToEKG[0:-4]
    print("Reading data")
    try:
        data = wfdb.rdsamp(pathToEKG)
    except:
        print("An error occurred during the loading of data. Returning to main menu.")
        return

    X = np.array([signal for signal, meta in data])
    print("Data loaded")
    classified = model.predict(X)

    for k in range(classified.size):
        print("%s%2d%s"%(MapOfDisease[k]+":").ljust(72),math.trunc(classified[0][k]*100),"%")

    print("Model shows that the most likely classification is:",MapOfDisease[np.argmax(classified[0])])
    ListOfDisease[0]="Unknown"
    names = ListOfDisease
    values = classified[0]
    plt.plot(names, values)
    plt.xticks(rotation=90)
    plt.suptitle('Categorical Plotting')
    plt.show()
    print("Operation complete, returning to main menu")
    return
```

Slika 4.1 Isječak koda iz aplikacije

Korisnik može odabrati i opciju učenja. Tada će mu aplikacija ponuditi mogućnost dodatnog učenja već postojećeg ugrađenog modela ili novog modela. Aplikacija će ga pitati za ime modela kojeg vraća, broj epoha za učenje, koliko zapisa korisnik želi dati modelu na učenje, te na kraju za put do .csv datoteke iz koje će uzeti putanje do datoteka iz kojih će model učiti. Kao rezultat dobit će dodatno naučeni postojeći model ili naučeni novi model. Dani model nakon učenja spremljen je na disk u isti direktorij gdje se nalazi i aplikacija. Tako će ga korisnik dalje moći koristiti za svoje potrebe.

Zaključak

Cilj ovog rada bio je analizirati utjecaj hiperparametara dubokih neuronskih mreža na učinkovitost klasifikacije oboljenja iz 12 kanalnog elektrokardiograma. Optimizacija hiperparametara dubokih neuronskih mreža problem je koji zahtijeva velike računalne resurse. Za svaki promijenjeni hiperparametar potrebno je ponovno pokrenuti učenje mreže, koje ovisno o modelu može trajati i do nekoliko sati. Zbog toga je ovaj problem jako vremenski zahtijevan, te je za potpunu optimizaciju često potreban cijeli tim inženjera.

Ovo istraživanje dotaklo se najbitnijih hiperparametara početnog modela. Početna mreža imala je tri konvolucijska sloja gdje je nakon svakog slijedio sloj normalizacije te sloj sažimanja. Nakon konvolucijskih, slijedila su tri sloja dugog kratkoročnog pamćenja te izlazni sloj. Početna mreža imala je točnost približno 60%.

Nakon provedenog istraživanja i analize hiperparametara kroz 73 ispitane konfiguracije mreže, točnost na skupu za validaciju je dovedena na 66,3%, što čini napredak od 6,3%. Na skupu za testiranje točnost je dovedena na 65%. Kroz istraživanje ispitan je broj epoha za učenje, veličina serije podataka koja se daje mreži pri svakoj iteraciji u učenju. Ispitani su hiperparametri konvolucijskog sloja koji su širina jezgre, broj jezgri, aktivacijske funkcije slojeva, te broj konvolucijskih slojeva. Od hiperparametara slojeva dugog kratkoročnog pamćenja ispitani su broj jedinica u svakom sloju, te broj takvih slojeva. Također su ispitani algoritmi optimizacije učenja duboke neuronske mreže.

Kao najbolji model pokazao se onaj koji je bio učen sa 100 epoha s ranim zaustavljanjem, veličinom grupe 16, te optimizacijskim algoritmom Adam. Taj model sadrži 4 konvolucijska sloja gdje prvi ima 256 jezgri, a ostali 128 jezgri širine 12 nakon kojih slijede slojevi normalizacije i sažimanja, 3 sloja s dugim kratkoročnim pamćenjem od kojih svaki sadrži 200 LSTM jedinica.

Usporedimo li rezultate modela dobivenog ovim istraživanjem s najboljim modelima s natjecanja provedenog od PhysioNet/CinC Challenges (Classification of 12-lead ECGs, 2020) [17], zaključujemo da su rezultati doista dobri. Tri najbolja modela s tog natjecanja imali su točnosti na skupu za testiranje 49,2%, 41,2%, te 39,6%. Unatoč tome, mora se uzeti u obzir da su na tom natjecanju korišteni EKG zapisi s čak pet dodatnih izvora. Zbog toga teško možemo pravilno usporediti ovaj model s modelima s natjecanja. Premda ova

usporedba nije najpravičnija, ona služi kao pozitivna smjernica koja pruža informaciju da je istraživanje uspješno provedeno.

Rezultati ovog završnog rada su jasni, a mogućnost poboljšanja učinkovitosti modela uvijek postoji. Kroz još više uloženo vrijeme i ispitivanja ostalih hiperparametara kojih ima mnogo, zasigurno se točnost može još povećati. Unatoč tome, 6,3% na skupu za validaciju i 5% na skupu za učenje napredak je koji pokazuje da je optimizacija hiperparametara jedan bitan dio kod učenja dubokih neuronskih mreža.

Literatura

- [1] World Health Organization, Cardiovascular diseases. Poveznica: <https://www.who.int>; pristupljeno 30. svibnja 2021.
- [2] Wagner, P., Strodthoff, N., Bousseljot, R., Samek, W., & Schaeffter, T. (2020). PTB-XL, a large publicly available electrocardiography dataset (version 1.0.1). *PhysioNet*. Poveznica: <https://doi.org/10.13026/x4td-x982>; pristupljeno 23. svibnja 2021.
- [3] Clifford G.D., Azuaje F., McSherry P.E. *Advanced Methods and Tools for ECG Data Analysis*, Artech House, 2006.
- [4] Wikipedia, Čovječje srce. Poveznica: https://hr.wikipedia.org/wiki/Čovječje_srce; pristupljeno 30. svibnja 2021.
- [5] Magjarević R., *Tehnologija u medicini*, UNIZG FER, 2021.
- [6] Wikipedia, Electrocardigraphy. Poveznica: <https://en.wikipedia.org/wiki/Electrocardigraphy>; pristupljeno 30. svibnja 2021.
- [7] WFDB, The WFDB software package. Poveznica: <https://archive.physionet.org/physiotools/wfdb.shtml>; pristupljeno 25. svibnja 2021.
- [8] Google Colab, Colaboratory. Poveznica: <https://colab.research.google.com/>; pristupljeno 19. travnja 2021.
- [9] Python, The official home of the Python Programming Language. Poveznica: <https://www.python.org/>; pristupljeno 1. lipnja 2021.
- [10] Tensorflow, An end-to-end machine learning platform. Poveznica: <https://www.tensorflow.org/>; pristupljeno 1. lipnja 2021.
- [11] Keras, Simple. Flexible. Powerful. Poveznica: <https://keras.io/>; pristupljeno 1. lipnja 2021.
- [12] Matplotlib, Matplotlib: Visualization with Python. Poveznica: <https://matplotlib.org/>; pristupljeno 29. svibnja 2021.
- [13] Machine Learning Mastery, A gentle introduction to Dropout for Regularizing Deep Neural Networks. Poveznica: <https://machinelearningmastery.com/dropout-for-regularizing-deep-neural-networks/>; pristupljeno 2. lipnja 2021.
- [14] Buduma N., Locascio N. *Fundamentals of Deep Learning*, 1. izdanje. O'Reilly, 2017.
- [15] Šegvić S., Čupić M., Tutek M., *Duboko učenje*, UNIZG FER, 2021.
- [16] SGD, Gradient descent optimizer. Poveznica: <https://keras.io/api/optimizers/sgd/>; pristupljeno 3. lipnja 2021.
- [17] Perez Alday EA, Gu A, Shah AJ, Robichaux C, Wong AI, Liu C, Liu F, Rad AB, Elola A, Seyed S, Li Q, Sharma A, Clifford GD*, Reyna MA*. Classification of 12-lead ECGs: the PhysioNet/Computing in Cardiology Challenge 2020. *Physiol Meas*. 2020 Nov 11. Poveznica: <https://iopscience.iop.org/article/10.1088/1361-6579/abc960>; pristupljeno 9. lipnja 2021.

Sažetak

Naslov:

Ispitivanje utjecaja hiperparametara dubokih neuronskih mreža na učinkovitost klasifikacije poremećaja iz 12-kanalnog elektrokardiograma.

Sažetak:

Kardiovaskularne bolesti pripadaju najopasnijim bolestima s kojima se ljudi susreću. Elektrokardiogram najbitniji je nalaz kod svih kardioloških pregleda. Analiza EKG-a problem je koji se može primijeniti u računarstvu koristeći umjetnu inteligenciju, a kao najbolja metoda pokazale su se duboke neuronske mreže. U ovom radu provedeno je istraživanje nekih od najbitnijih hiperparametara dubokih neuronskih mreža, te njihov utjecaj na uspješnost klasifikacije oboljenja iz zapisa EKG-a. Kroz istraživanje točnost početnog modela na skupu za validaciju poboljšana je za 6,3%, a na skupu za testiranje za 5%. Osim analize, kao rezultat ovog rada napravljena je i aplikacija koja korisnicima nudi učitavanje zapisa, njihovo učenje ili testiranje te ispis rezultata.

Ključne riječi:

Elektrokardiogram, srce, duboka neuronska mreža, hiperparametar, konvolucijski sloj, sloj s dugim kratokročnim pamćenjem

Summary

Title:

Evaluating the Influence of Deep Neural Network Hyperparameters on Disorder Classification Efficiency from 12-Channel Electrocardiogram

Summary:

Cardiovascular diseases belong to the most dangerous diseases that people encounter. The electrocardiogram is the most important finding in all cardiac examinations. ECG analysis is a problem that can be applied in computing using artificial intelligence, and deep neural networks have proven to be the best method. In this paper, a research of the most important hyperparameters of deep neural networks was conducted, and their impact on the success of the classification of diseases from ECG records. Through research, the accuracy of the initial model on the validation set was improved by 6.3% and on the test set by 5%. In addition to the analysis, as a result of this paper, an application was created that offers users to load records, learn or test them and print the results.

Keywords:

Electrocardiogram, heart, deep neural network, hyperparameter, convolutional layer, long short term memory layer