

Usporedba modela strojnog učenja za klasifikaciju poremećaja iz spektra autizma korištenjem slika funkcijske magnetske rezonancije mozga

Pavelić, Dominik

Undergraduate thesis / Završni rad

2024

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Zagreb, Faculty of Electrical Engineering and Computing / Sveučilište u Zagrebu, Fakultet elektrotehnike i računarstva**

Permanent link / Trajna poveznica: <https://urn.nsk.hr/urn:nbn:hr:168:867009>

Rights / Prava: [In copyright](#) / [Zaštićeno autorskim pravom](#).

Download date / Datum preuzimanja: **2025-03-15**



Repository / Repozitorij:

[FER Repository - University of Zagreb Faculty of Electrical Engineering and Computing repository](#)



SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 1272

**USPOREDBA MODELA STROJNOG UČENJA ZA
KLASIFIKACIJU POREMEĆAJA IZ SPEKTRA AUTIZMA
KORIŠTENJEM SLIKA FUNKCIJSKE MAGNETSKE
REZONANCIJE MOZGA**

Dominik Pavelić

Zagreb, lipanj 2024.

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 1272

**USPOREDBA MODELA STROJNOG UČENJA ZA
KLASIFIKACIJU POREMEĆAJA IZ SPEKTRA AUTIZMA
KORIŠTENJEM SLIKA FUNKCIJSKE MAGNETSKE
REZONANCIJE MOZGA**

Dominik Pavelić

Zagreb, lipanj 2024.

ZAVRŠNI ZADATAK br. 1272

Pristupnik: **Dominik Pavelić (0036543749)**
Studij: Elektrotehnika i informacijska tehnologija i Računarstvo
Modul: Računarstvo
Mentorica: doc. dr. sc. Jelena Božek

Zadatak: **Usporedba modela strojnog učenja za klasifikaciju poremećaja iz spektra autizma korištenjem slika funkcijske magnetske rezonancije mozga**

Opis zadatka:

Kombinacijom strojnog učenja s podacima slika funkcijske magnetske rezonancije (MR) mozga mogu se otkriti skriveni obrasci i povezanosti u velikim skupovima podataka. Strojno učenje pruža alate za analizu i obradu ovih podataka te za identifikaciju uzoraka povezanih s određenim mentalnim stanjima ili poremećajima, poput autizma. Autizam predstavlja kompleksni razvojni poremećaj mozga koji manifestira poteškoće u socijalnim interakcijama, oblikovanju ponašanja, razumijevanju te progresiji kognitivnih sposobnosti. U radu će se opisati metode strojnog učenja primijenjenih za detekciju i klasifikaciju poremećaja iz spektra autizma temeljem informacija iz MR slika mozga. Posebnu pažnju usmjeriti na metode koje koriste funkcijske MR slike. Cilj rada je implementirati i usporediti modele strojnog učenja temeljene na neuronskim mrežama koje klasificiraju poremećaj iz spektra autizam koristeći funkcijske MR slike dostupne u bazi ABIDE.

Rok za predaju rada: 14. lipnja 2024.

Sadržaj

1. Uvod	3
2. Pregled literature	5
3. Duboko učenje	7
3.1. Umjetne neuronske mreže	7
3.2. Umjetni neuroni	7
3.3. Prijenosne funkcije	8
3.4. Vrste slojeva AAN	12
3.5. Koraci učenja	14
3.5.1. Podjela podataka	14
3.5.2. Treniranje modela	14
3.5.3. Validacija modela	15
3.5.4. Testiranje modela	16
3.6. Optimizacija AAN	16
3.6.1. Funkcije gubitka	17
3.6.2. Funkcije optimizacije	18
3.7. Podnaučenost i prenaučенost	18
3.8. Procjena uspješnosti	19
4. Materijali i metode	21
4.1. Korišteni podatci	21
4.2. Pretprocesiranje podataka	21
4.3. Regije interesa i atlas	23
4.4. Ekstrakcija značajki	26
4.4.1. <i>Nilearn</i> biblioteka	27

5. Implementacija modela	28
5.1. <i>Tensorflow</i> i <i>Keras</i> biblioteke	28
5.2. Arhitekture modela	28
5.3. Treniranje, optimizacija i evaluacija modela	35
6. Rezultati	36
7. Zaključak	42
Literatura	44
Sažetak	50
Abstract	51

1. Uvod

Poremećaji iz spektra autizma jedni su od najzastupljenijih neurorazvojnih poremećaja. Ovi poremećaji, od osobe do osobe, različito utječu na njihove kognitivne sposobnosti te ponašanje. Osobe s autizmom najčešće nailaze na izazove u komunikaciji te društvenoj interakciji [1]. S obzirom na to da autizam nije moguće izliječiti, rana i točna dijagnoza ključna je u pružanju pravilne njege i potpore. Tradicionalne metode dijagnoze oslanjaju se na strukturirana promatranja i analizu ponašanja djeteta, uz opsežne intervjue roditelja te testiranja motoričkih, govornih i kognitivnih sposobnosti [2]. Kao rezultat toga što su tradicionalne metode dijagnoze subjektivne i često prekasno uspostavljaju dijagnozu, povećava se interes za objektivnim i podacima vođenim načinima dijagnoze. Kao obećavajući pristup pokazala se analiza uzoraka aktivnosti mozga pomoću funkcijske magnetske rezonancije (engl. *functional magnetic resonance imaging*, fMRI).

Funkcijska magnetska rezonancija je neinvazivna tehnika snimanja aktivnosti mozga temeljena na mjerenju signala ovisnog o promjeni zasićenosti krvi kisikom (engl. *blood oxygen level-dependent*, BOLD). Ovaj signal je moguće izmjeriti jer molekule hemoglobina u krvi posjeduju različita magnetska svojstva ovisno o tome koliko je molekula kisika vezano na njih [3]. U području dijagnoze autizma, fMRI može otkriti specifične uzorke unutarnje povezanosti i aktivnosti mozga koji mogu služiti kao biomarkeri ovog poremećaja.

Duboko učenje, podgrana strojnog učenja, pokazalo se kao odličan alat za probleme klasifikacije raznih vrsta. Višeslojni perceptroni (engl. *multilayer perceptrons*, MLPs), jedna od osnovnih arhitektura modela dubokog učenja, pokazali su obećavajuće rezultate u području analize podataka dobivenih fMRI-jem s ciljem dijagnoze poremećaja iz spektra autizma [2].

U ovome radu napraviti će se usporedba performansi različitih višeslojnih perceptrona na problemu klasifikacije poremećaja iz spektra autizma pomoću slika dobivenih fMRI-jem. Istražit će se efikasnost različitih arhitektura MLP-a, uključujući različit broj skrivenih slojeva, različite vrste skrivenih slojeva te različit broj umjetnih neurona u pojedinom sloju.

2. Pregled literature

Aghdam et al. [4] koristili su konvolucijske neuronske mreže za klasifikaciju autizma iz podataka dobivenih fMRI-jem, istraživanje su specifično usmjerili na dijagnozu autizma u dječjoj dobi. U ovome istraživanju autori su koristili dinamičku i statičku kombinaciju klasifikatora te prijenosnog učenja, metoda u kojoj je neuronska mreža trenirana na podacima iz jedne domene, a primjenjuje se na podatke iz druge, kako bi riješili problem ograničenog skupa podataka. Istraživanje je provedeno na skupu podataka ABIDE I i ABIDE II te na njihovoj kombinaciji. Na podacima iz ABIDE I postignuta je točnost od 72,73%, odziv od 71,20% i preciznost od 73,48%, dok su na ABIDE II skupu podataka autori postigli točnost od 70,00%, odziv od 58,20% te preciznost od 80,40%. Na kombinaciji ABIDE I i ABIDE II postigli su 70,45% točnosti, 67,90% odziva i 74,21% preciznosti.

Bayram et al. [5] koristili su mrežu s dugoročnim kratkotrajnim pamćenjem (engl. *long short-term memory*, LSTM), konvolucijske neuronske mreže (engl. *convolutional neural network*, CNN), neuronske mreže s odzivnom vezom (engl. *recurrent neural network*, RNN) te hibridne modele kako bi istražili kompleksnost podataka dobivenih fMRI-jem i pokazali potencijal dubokog učenja u dijagnozi autizma. U istraživanju je korišteno 871 uzoraka iz ABIDE I skupa podataka te je RNN mreža iskazana kao najbolja što se tiče točnosti (74,74%) i odziva (72,95%). U preciznosti je bolja bila LSTM mreža preciznošću od 81,41%, a RNN mreža je imala preciznost od 76,28%. LSTM mreža pokazala je 74,4% točnosti i 66,25% odziva. Također, CNN mreža imala je točnost od 70,26%, odziv od 66,01% i preciznost od 73,93%, a kombinacija CNN i RNN mreža ostvarila je 67,28% točnosti, 57,57% odziva te 75,64% preciznosti.

Heinsfeld et al. [6] su pomoću *Craddock 200* atlasa ekstrahirali vremenske serije iz 200 regija interesa za svaki uzorak, od ukupno 1035, iz ABIDE I skupa podataka. Zatim su primijenili dva autoenkodera (enlg. *autoencoders*) za nenadzirano učenje te koriste znanje prikupljeno autoenkoderima za inicijalizaciju težina višeslojnog perceptrona kako bi model imao bolju sposobnost generalizacije. Postigli su točnost od 70%, odziv od 74% te preciznost od 63%. Također, pokazali su da postoji antikorelacija između prednjih i stražnjih djelova mozga te da postoji snažna korelacija regija u stražnjem dijelu mozga kod osoba s poremećajem iz spektra autizma.

Yang et al. [7] primijenili su klasične modele strojnog učenja kao što su logistička regresija, metodu potpunih vektora te *ridge* klasifikator na ABIDE I skup podataka. Od 1112 uzoraka u ABIDE I korišteno je njih 1035. Vremenske serije ekstrahirane su pomoću sedam različitih atlasa: *AAL*, *Dosenbach*, *Eickoff-Zilles*, *Harvard-Oxford Atlas*, *Talairach-Tournoux*, *Craddock 200*, *Craddock 400*. Ovi atlasii redom imaju 116, 160, 116, 110, 110, 200 te 400 regija interesa. U svim metodama klasifikacije korištenje *Craddock 400* atlas dalo je najbolje rezultate, pogotovo za *ridge* klasifikator gdje je dobivena točnost od 71,98%, odziv od 70,89% te preciznost od 71,53%.

Yang et al. [2] također su proveli istraživanje gdje su uspoređivali kako metoda pretprocesiranja podataka dobivenih fMRI-jem utječe na rezultate klasifikacije. U ovome istraživanju korišteno je 1035 uzoraka iz ABIDE I skupa podataka, a vremenske serije iz uzoraka ekstrahirane su pomoću *Craddock 400* atlasa. Prije ekstrakcije vremenskih serija podaci su pretprocesirani s četiri različita cjevovoda: CPAC, CCS, DPRSF, NIAK. Za klasifikaciju korišten je višeslojni perceptron s četiri različite konfiguracije. Kao najbolja konfiguracija za sve cjevovode pretprocesiranja pokazala se konfiguracija gdje skriveni slojevi imaju 128, a zatim 64 umjetna neurona. Od navedenih cjevovoda najbolji rezultati ostvareni su za CPAC cjevovod s točnosti od 75,27%, odzivom od 74% te preciznošću od 76,88%.

3. Duboko učenje

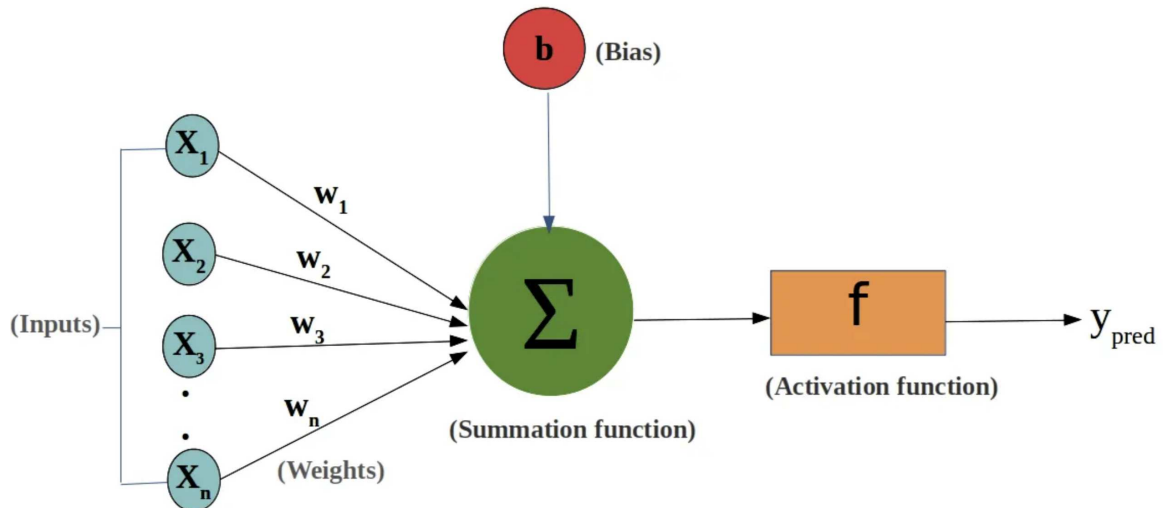
Strojno učenje je podgrana umjetne inteligencije čiji je glavni cilj razvitak algoritama koji računalnim programima omogućuju da "uče" iz podataka. Za razliku od tradicionalnog programiranja, gdje se programima eksplicitno definira znanje ili pravila izvođenja, algoritmi strojnog učenja samostalno identificiraju veze i obrasce među podacima kojima su izloženi, što im omogućuje da izvode predikcije i klasifikacije bez da su za to eksplicitno programirani [8]. Duboko učenje je podgrana strojnog učenja koja koristi više slojeva procesiranja za učenje reprezentacije podataka s povećanjem apstrakcije u svakom sljedećem sloju. Duboko učenje posebno se iskazalo u područjima prepoznavanja slika i govora, detekcije i raspoznavanja objekata te mnogim drugim. Uspjeh dubokog učenja leži u sposobnosti učenja kompleksnih značajki podataka koje modeli strojnog učenja ne mogu raspoznati [9].

3.1. Umjetne neuronske mreže

Umjetne neuronske mreže (engl. *artificial neural networks*, ANN) su matematički modeli inspirirani unutarnjim funkcioniranjem ljudskog mozga. Građene su od više slojeva od kojih svaki sadrži više podjedinica koje se nazivaju neuroni [10].

3.2. Umjetni neuroni

Umjetni neuroni su jednostavni matematički modeli, tj. funkcije, koji predstavljaju osnovne gradivne jedinice svake umjetne neuronske mreže. Njihov dizajn i funkcionalnost proizlazi iz proučavanja bioloških neurona [11]. Građa umjetnog neurona može se vidjeti na slici 3.1.



Slika 3.1. Prikaz građe umjetnog neurona [12].

Ulaz u umjetne neurone su vektori proizvoljnih duljina. Svaki element ulaznog vektora ima svoju pripadnu težinu s kojom se množi pri ulasku u neuron. Zatim se u "tijelu" neurona ulazi pomnoženi s težinama zbrajaju te se na tu vrijednost dodaje pomak (engl. *bias*). Ta se suma zatim prosljeđuje prijenosnoj funkciji. Na kraju se dobivena vrijednost predaje sljedećem sloju neurona ili na izlaz neuronske mreže [11].

$$y = F\left(\sum_{i=0}^m x_i w_i + b\right) \quad (3.1)$$

Jednadžba 3.1 predstavlja matematički zapis umjetnog neurona, gdje je m broj elemenata ulaznog vektora, x_i predstavlja jedan element ulaznog vektora, w_i je težina pridružena elementu x_i , b predstavlja pomak, a F je prijenosna funkcija.

3.3. Prijenosne funkcije

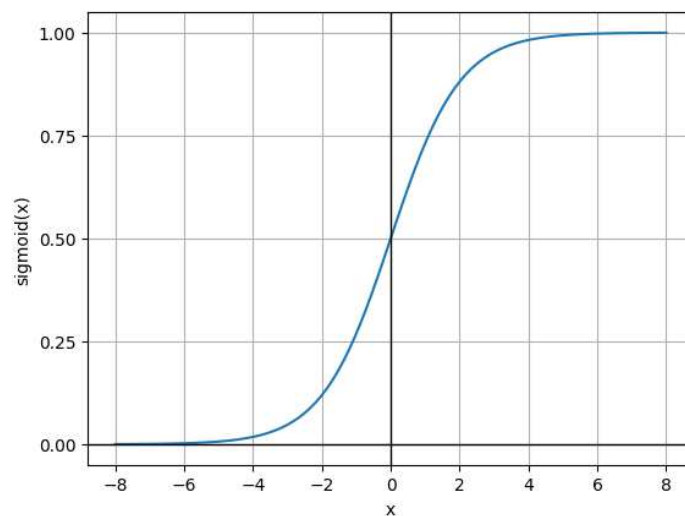
Prijenosne funkcije, još zvane aktivacijske funkcije, igraju važnu ulogu u umjetnim neuronskim mrežama na način da u model unose nelinearnost koja je potrebna kako bi model mogao naučiti složene odnose u podacima [13]. Slijede neke od najčešće korištenih prijenosnih funkcija:

1. Sigmoidna funkcija

Sigmoidna funkcija je jedna od najkorištenijih nelinearnih prijenosnih funkcija, koja preslikava vrijednosti iz intervala $\langle -\infty, +\infty \rangle$ u interval $\langle 0, 1 \rangle$. Vrijednosti bliže $-\infty$ preslikavaju se bliže 0, a one bliže $+\infty$ preslikavaju se u vrijednosti bliže 1.

$$f(x) = \sigma(x) = \frac{1}{1 + e^{-x}} \quad (3.2)$$

Jednadžba 3.2 predstavlja definiciju sigmoidne funkcije, a na slici 3.2. može se vidjeti graf sigmoidne funkcije.



Slika 3.2. Graf sigmoidne funkcije

2. Tangens hiperbolični

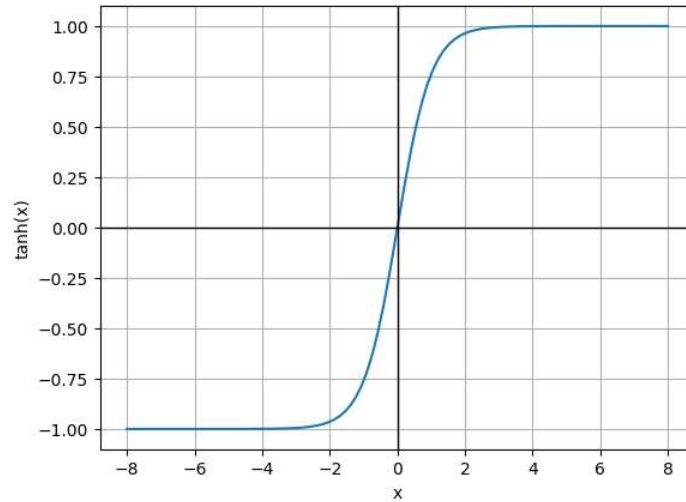
Tangens hiperbolični sličan je sigmoidnoj funkciji, razlikuju se po tome što tangens hiperbolični vrijednosti iz intervala $\langle -\infty, +\infty \rangle$ preslikava u vrijednosti iz intervala $\langle -1, 1 \rangle$ te time na izlazu možemo dobiti negativne vrijednosti, dok sigmoidna funkcija može proizvesti samo pozitivne vrijednosti.

$$f(x) = \tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (3.3)$$

Definicija tangensa hiperboličnog može se vidjeti u jednadžbi 3.3, tangens hiperbolični također se može definirati pomoću sigmoidne funkcije što je vid-

ljiwo u jednadzbi 3.4 Na slici 3.3. vidljiv je graf tangensa hiperboličnog.

$$f(x) = \tanh(x) = 2\sigma(2x) - 1 \quad (3.4)$$

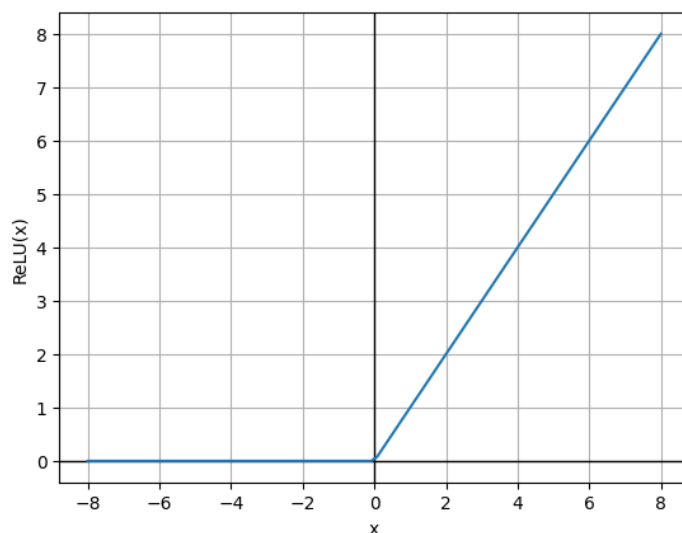


Slika 3.3. Graf tangensa hiperboličnog

3. Zglobnica

Zglobnica (engl. *Rectified Linear Unit*, ReLU) je također jedna od najčešće korištenih prijenosnih funkcija. Prednost zglobnice je u tome što neuroni jednog sloja mreže ne aktiviraju svi u isto vrijeme, neuron je aktivan samo kada je mu je vrijednost različita od nule [14]. Na slici 3.4. može se vidjeti graf zglobnice.

$$f(x) = \text{ReLU}(x) = \max(0, x) = \begin{cases} x, & \text{ako } x \geq 0 \\ 0, & \text{ako } x < 0 \end{cases} \quad (3.5)$$

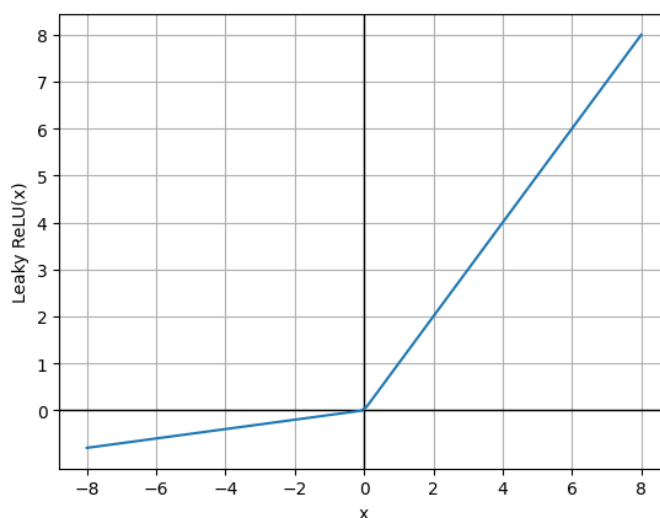


Slika 3.4. Graf zglobnice

4. Propusna zglobnica

Propusna zglobnica (engl. *Leaky Rectified Linear Unit*, LReLU) je modifikacija zglobnice gdje se ulazne vrijednosti manje od nule množe s konstantom 0.01, što omogućava dobivanje negativnih vrijednosti na izlazu iz neurona. Na slici 3.5. vidljiv je graf propusne zglobnice.

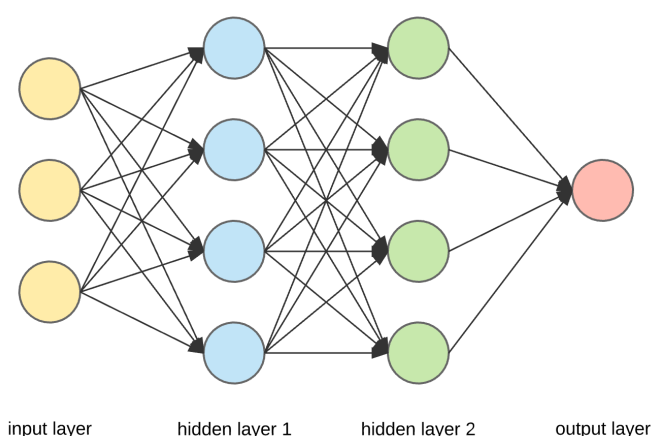
$$f(x) = \text{ReLU}(x) = \max(\alpha x, x) = \begin{cases} x, & \text{ako } x \geq 0 \\ 0.01x, & \text{ako } x < 0 \end{cases} \quad (3.6)$$



Slika 3.5. Graf propusne zglobnice

3.4. Vrste slojeva AAN

Svaki sloj neuronske mreže građen je od određenog broja neurona, slojevi se međusobno nadovezuju te tako tvore neuronsku mrežu. Ulazni sloj ima onoliko neurona koliko vektor značajki ima elemenata, a broj neurona u izlaznom sloju ovisio o broju klasa, ako se radi o klasifikacijskom problemu, ili ovisi o broju ciljnih varijabli, ako se radi o regresiji. Svi slojevi osim ulaznog i izlaznog sloja nazivaju se skrivenim slojevima (engl. *hidden layers*) jer ne možemo izravno vidjeti njihovo djelovanje jer se njihovi izlazi samo prosljeđuju dalje sljedećem sloju mreže. Na slici 3.6. može se vidjeti građa jednostavne neuronske mreže.



Slika 3.6. Prikaz građe jednostavne neuronske mreže [15]

Sljede neki od osnovnih vrsta slojeva neuronskih mreža:

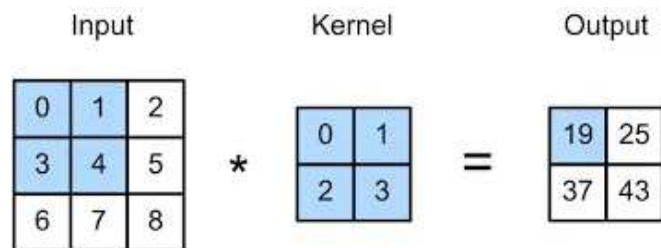
1. Potpuno povezani slojevi

Potpuno povezani (engl. *dense/fully connected*) slojevi su vrsta sloja u kojoj su izlazi iz jednog sloja povezani na ulaz svakog neurona iz sljedećeg sloja. Skriveni slojevi na slici 3.6. primjer su potpuno povezanih slojeva.

2. Konvolucijski slojevi

Konvolucijski slojevi najčešće se primjenjuju kada se neuronskim mrežama za ulazne podatke daju slike. Ovi slojevi nad tim podacima vrše matematičku operaciju konvolucije. Ta operacija sastoji se od primjerne filtra, odnosno jezgara (engl. *kernels*), koji pokušavaju "naučiti" vizualne obrasce. Filtri su ma-

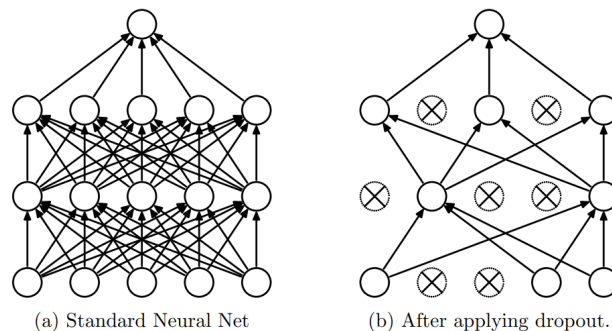
trice koje se pomiču po slikama te nad dijelovima slike koje pokrivaju rade skalarni umnožak [16]. Na slici 3.7. može se vidjeti jednostavna operacija konvolucije.



Slika 3.7. Prikaz jednostavne konvolucije

3. Sloj ispuštanja

Slojevi ispuštanja (engl. *dropout layers*) služe kako bi se spriječila prenaučenosť neuronske mreže (više o tome u 3.7.). Ovi slojevi rade na način da privremeno ispuste neke neurone te sve njihove ulaze i izlaze prema drugim neuronima. Odluka koji neuroni će se ispustiti, tj. zadržati, ovisi o vjerojatnosti p koja se najčešće postavlja na 0.5 [17]. Na slici 3.8. može se vidjeti stanje neuronske mreže prije i nakon ispuštanja pojedinih neurona.



Slika 3.8. Prikaz neuronske mreže prije i nakon ispuštanja neurona [17]

4. Normalizacijski sloj

Normalizacijski sloj služi za stabilizaciju distribucije ulaza u sloj neuronske mreže tijekom treniranja. Ovo se postiže tako da se srednja vrijednost distribucije svake aktivacije sloja postavi na 0, a standardna devijacija se postavlja na 1 [18].

3.5. Koraci učenja

3.5.1. Podjela podataka

Najprije se proučavani skup podataka dijeli na nekoliko skupova: skup za treniranje, skup za validaciju i skup za testiranje. Svrha podjele podataka na podskupove je sprječavanje pretreniranja modela. Ovi podskupovi moraju biti međusobno neovisni te među njima ne smije postojati preklapanje. Skupovi se dijele u različitim omjerima ovisno o veličini skupa podataka. Skupovi od 100 do 1 000 000 uzoraka najčešće se dijele u omjeru 60:20:20, 60% podataka za treniranje, 20% za validaciju i 20% za testiranje. Ako je veličina skupa podataka veća od 1 milijuna uzoraka podaci se dijele u omjerima 98:1:1 ili 99:0.5:0.5 [19].

3.5.2. Treniranje modela

Treniranje modela je postupak u kojemu se model strojnog učenja izlaže podaci iz skupa za treniranje te model zatim pokušava učiti iz tih podataka. Svaki model ima svoje unutarnje parametre, na primjer u neuronskim mrežama to su težine i pomaci, koje mijenja i prilagođuje tijekom treniranja. Težine određuju koliko neka značajka ulaznih podataka utječe na izlaz, a pomaci su konstantne vrijednosti koje osiguravaju da čak i ako su sve ulazne vrijednosti u neuron 0 da se neuron i dalje aktivira [20]. Cilj treniranja modela je izgraditi što precizniji matematički prikaz funkcijski veza značajki ulaznih podataka i njihovih oznaka [21].

Treniranje neuronskih mreža je iterativan proces, gdje se jedna iteracija naziva epohom (engl. *epoch*) te označava koliko će puta model vidjeti cijeli ulazni skup podataka. Ulazni skup podataka dijeli se na manje podskupove (engl. *batch*), model se u jednoj epohi izlaže jednom po jednom podskupu te nakon svakog izlaganja se ažuriraju unutarnji parametri modela, epoha završava kada je model vidio sve podskupove. Koliko će se promijeniti unutarnji parametri modela određuje stopa učenja (engl. *learning rate*).

3.5.3. Validacija modela

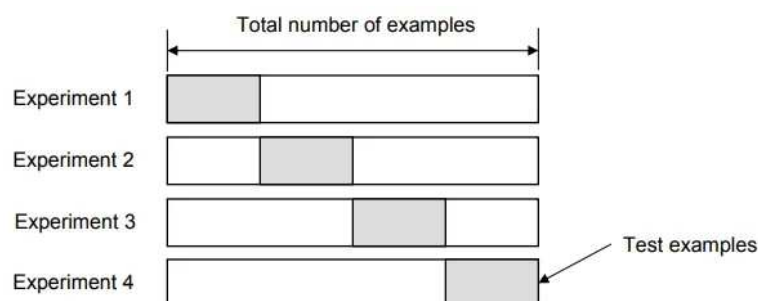
Validacija (engl. *validation*) je proces provjere performansi modela na skupu podataka za validaciju, tj. podacima koje model još nije vidio tijekom treniranja. Svrha validacije je ugađanje hiperparametara modela kao što su broj epoha, veličina *batcha* i stopa učenja. Kod treniranja neuronskih mreža validacija se provodi nakon završetka svake epohe [22]. Neke od često korištenih metoda validacije su:

1. Validacija izdvajanjem

Validacija izdvajanjem (engl. *holdout validation*) je jedna od najkorištenijih metoda validacije modela. U ovoj metodi početni skup podataka dijeli se na trening skup, validacijski skup i skup za testiranje u omjeru koji ovisi o veličini početnog skupa. Često se koristi kod treniranja modela dubokog učenja s velikim skupom podataka jer nije računalno skupa [22].

2. K-struka unakrsna validacija

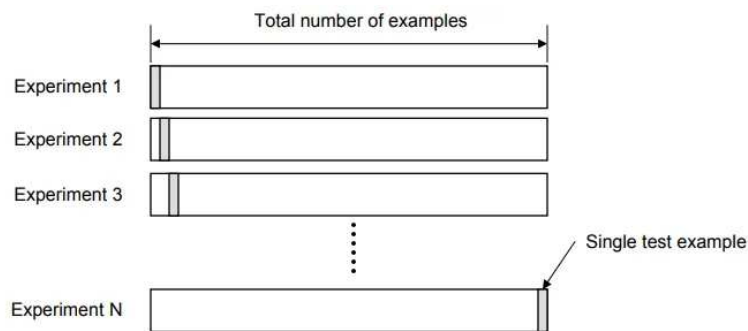
U k -strukoj unakrsnoj validaciji (engl. *k-fold cross-validation*) početni se skup podataka dijeli na dva dijela, skup za treniranje i skup za testiranje. Trening skup se zatim dijeli na k dijelova, od kojih se jedan koristi za validaciju, a ostalih $k - 1$ dijelova se spajaju u trening skup. Ovaj postupak se ponavlja k puta tako da je svaki dio jednom korišten za validaciju [22]. Na slici 3.9. može se vidjeti kako se početni skup podataka dijeli za k -struku unakrsnu validaciju.



Slika 3.9. Prikaz podjele skupa podataka u k -fold validaciji [23]

3. Izdvoji jednoga unakrsna validacija

Izdvoji jednoga unakrsna validacija (engl. *leave-one-out cross validation*) je varijacija k -struke unakrsne validacije gdje se skup podataka umjesto na k dijelova dijeli na n dijelova, gdje je n broj uzoraka u skupu. Ova se metoda koristi kada su skupovi podataka mali jer zahtjeva treniranje modela n puta [22]. Na slici 3.10. može se vidjeti kako se početni skup podataka dijeli za izdvoji jednoga unakrsnu validaciju.



Slika 3.10. Prikaz podjele skupa podataka u k -fold validaciji [23]

3.5.4. Testiranje modela

Završni korak učenja modela je testiranje. U ovome koraku modelu se daju podaci iz skupa za testiranje, koje model do sada nikada nije vidio niti tijekom treniranja niti tijekom validacije. Svrha testiranja je provjera koliko model dobro generalizira na novim podacima. Neke od mjera performansi modela su točnost, preciznost, *recall* i *F1-score* (vidi poglavlje 3.8.).

3.6. Optimizacija AAN

Prilikom treniranja neuronskih mreža provode se dva algoritma. Prvi je propagacija unaprijed, neuroni iz svojih ulaza pomoću jednadžbe 3.1 računaju svoje izlaze koji se prosljeđuju sljedećem sloju sve dok se ne izračuna izlaz iz mreže. Iz izračunanog izlaza mreže se pomoću funkcije gubitka računa gubitak, tj. pogreška, mreže u usporedbi s stvarnim izlazom za ulazne podatke. Zatim nastupa unazadna propagacija koja ovaj gubitak prosljeđuje unazad prema početku mreže te optimizacijska funkcija u svakom neuronu na

temelju ovog gubitka prilagođava težine na ulazima neurona kako bi u sljedećoj iteraciji gubitak bio manji.

3.6.1. Funkcije gubitka

Funkcija pomoću koje se računa koliko izlaz kojeg je mreža predvidjela odstupa od stvarne vrijednosti izlaza za neki ulaz naziva se funkcija gubitka (engl. *loss function*). Vrijednost ove funkcije pokušava se minimizirati [10]. Neke od najčešće korištenih funkcija gubitka:

1. Srednja kvadratna pogreška

$$MSE = \frac{1}{n} \sum_{i=1}^n (\hat{y}_i - y_i)^2 \quad (3.7)$$

2. Srednja apsolutna pogreška

$$MAE = \frac{1}{n} \sum_{i=1}^n |\hat{y}_i - y_i| \quad (3.8)$$

U jednadžbama 3.7 i 3.8 mogu se vidjeti formule za računanje srednje kvadratne i srednje apsolutne pogreške, gdje n broj uzoraka za treniranje, y_i je stvarna vrijednost izlaza, a \hat{y}_i je predviđena vrijednost za uzorak i . Ove funkcije gubitka koriste se kada se neuronske mreže primjenjuju na regresijske probleme [24].

3. Unakrsna entropija

$$L = -\frac{1}{n} \sum_{i=1}^n y_i \cdot \log(\hat{y}_i) \quad (3.9)$$

4. Binarna unakrsna entropija

$$L = -\frac{1}{n} \sum_{i=1}^n (y_i \cdot \log(\hat{y}_i) + (1 - y_i) \cdot \log(1 - \hat{y}_i)) \quad (3.10)$$

Jednadžbe 3.9 i 3.10 prikazuju funkcije gubitka unakrsne entropije te binarne unakrsne entropije. Binarna unakrsna entropija koristi se kada se neuronskom mrežom podaci klasificiraju u dvije različite klase, a kada se podaci klasificiraju u više klasa koristi se unakrsna entropija [24].

3.6.2. Funkcije optimizacije

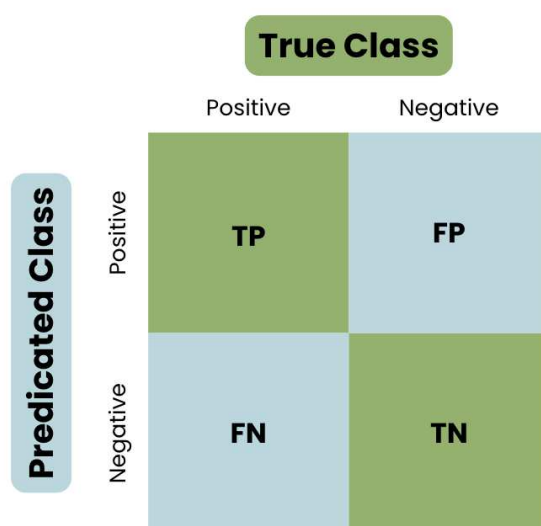
Funkcije optimizacije su funkcije koje pokušavaju minimizirati funkciju gubitka, kako bi to postigle mijenjaju unutarnje parametre modela, tj. težine na ulazima neurona. Za funkciju gubitka $f(x)$ računa se njena prva derivacija $f'(x)$, derivacija funkcije daje informaciju o nagibu funkcije u točki x te kako promijeniti ulaz x da bi se dobilo poboljšanje na izlazu funkcije. Tehnika kada se x mijenja u malim koracima u suprotnom smjeru od derivacije naziva se gradijentni spust (engl. *gradient descent*) [10]. U dubokom učenju funkcije na ulazu primaju vektor vrijednosti, pa se za njih računaju parcijalne derivacije $\frac{\partial}{\partial x_i} f(\mathbf{x})$, koja daje informaciju kako se vrijednost funkcije f mijenja s obzirom na promjenu vrijednosti x_i u točki \mathbf{x} . Vektor koji sadrži sve parcijalne derivacije funkcije f naziva se gradijentom, $\nabla_{\mathbf{x}} f(\mathbf{x})$ [10].

3.7. Podnaučenost i prenaučenost

Glavni cilj strojnog učenja općenito je da modeli dobro rade, daju dobre rezultate, na podacima koje model nikada nije vidio za vrijeme treniranja. Ova sposobnost modela da dobro radi na novim podacima naziva se generalizacija [10]. Podnaučenost je situacija koja se događa pri treniranju modela kada je gubitak na skupu podataka za treniranje prevelik, to znači da model ne može "shvatiti" vezu među podacima te da nema dobru sposobnost generalizacije. Podnaučenost modela se najčešće pojavljuje kad je model prejednostavan [25]. Prenaučenost je situacija suprotna podnaučenosti, događa se kada je razlika gubitka na skupu podataka za treniranje i skupu za testiranje prevelika. Prenaučenost znači da se model previše prilagodio podacima za treniranje te time model gubi sposobnost generalizacije. Neki od čestih uzroka prenaučivosti su da je model presložen, model se trenirao predugo pa je naučio "šum", tj. nebitne informacije, u podacima [26].

3.8. Procjena uspješnosti

Procjena uspješnosti modela služi da bi se provjerile performanse modela na skupu podataka za testiranje, drugim riječima služi kao provjera sposobnosti generalizacije modela na neviđenim podacima. Također, služi tome da se modeli s dobrim performansama mogu dalje unaprijediti, a da se oni s lošim mogu odbaciti [27]. U klasifikacijskim problemima, predikcije modela mogu se svrstati u četiri kategorije: istinito pozitivno (engl. *true positive*, TP) - model točno predviđa pozitivnu klasu, istinito negativno (engl. *true negative*, TN) - model točno predviđa negativnu klasu, lažno pozitivno (engl. *false positive*, FP) - model netočno predviđa pozitivnu klasu te lažno negativno (engl. *false negative*, FN) - model netočno predviđa negativnu klasu [28]. Ove vrijednosti mogu se skupno prikazati matricom zabune (engl. *confusion matrix*), koja se može vidjeti na slici 3.11.



Slika 3.11. Prikaz matrice zabune [29]

Iz matrice zabune mogu se izračunati četiri mjere procjene uspješnosti modela:

1. Točnost

Točnost (engl. *accuracy*) je mjera koja prikazuje omjer ukupnog broja točnih (istinitih) predikcija i ukupnog broja predikcija [30].

$$\text{Točnost} = \frac{TP + TN}{TP + TN + FP + FN} \quad (3.11)$$

2. Preciznost

Preciznost (engl. *precision*) je mjera točnosti pozitivnih predikcija modela, prikazuje koliko, od ukupno predviđenih pozitivnih klasa, je uistinu pozitivno. Računa se kao omjer broja istinito pozitivnih predikcija i broja svih pozitivnih predikcija [30].

$$\text{Preciznost} = \frac{TP}{TP + FP} \quad (3.12)$$

3. Odziv

Odziv (engl. *recall*), još nazivan osjetljivost (engl. *sensitivity*), je mjera koja prikazuje koliko dobro model klasificira istinito pozitivne uzorke, tj. koliko dobro model predviđa relevantne podatke. Računa se kao omjer broja istinito pozitivnih predikcija modela te zbroja istinito pozitivnih i lažno negativnih predikcija [30].

$$\text{Odziv} = \frac{TP}{TP + FN} \quad (3.13)$$

4. F1-mjera

S obzirom na to da je u isto vrijeme teško postići visoki odziv i visoku preciznost, kao procjena uspješnosti model, često se koristi F1-mjera (engl. *F1-score*) koja predstavlja ravnotežu odziva i preciznosti. Računa se kao harmonijska srednja vrijednost odziva i preciznosti [30].

$$\text{F1-mjera} = 2 \cdot \frac{\text{Odziv} \cdot \text{Preciznost}}{\text{Odziv} + \text{Preciznost}} \quad (3.14)$$

4. Materijali i metode

4.1. Korišteni podatci

U ovome radu korišteni su podaci iz *Autism Brain Imaging Data Exchange (ABIDE)* skupa podataka [31]. Ovaj skup podataka sastoji se od funkcijskih i strukturalnih MRI podataka 1112 pojedinaca, prikupljenih sa 17 internacionalnih institucija, od kojih je 539 dijagnosticirano s autizmom, a preostalih 573 čine kontrolnu grupu. Od 1112 pojedinaca u originalnom skupu, nakon prolaska kontrole kvalitete podataka, u ovom radu korišteno je njih 871. Od tih 871 (727 muškaraca, 144 žene) njih je 403 dijagnosticirano s autizmom, a kontrolnu grupu sačinjava preostalih 468 pojedinaca. Svi pojedinci su u rasponu od 6 do 58 godina. Sažetak podataka (institucija, grupa, spol i godine) vidljiv je u tablici 4.1.

4.2. Pretprocesiranje podataka

Podaci u skupu podataka ABIDE pretprocesirani su s četiri različita cjevovoda: *Connectome Computation System (CCS)*, *Configurable Pipeline for the Analysis of Connectomes (CPAC)*, *Data Processing Assistant for Resting-State fMRI (DPARSF)* i *Neuroimaging Analysis Kit (NIAK)*. Podaci pretprocesirani na sva četiri načina dostupni su za preuzimanje na [32]. Koraci i načini pretprocesiranja podataka u sva četiri cjevovoda su poprilično slični, ono po čemu se razlikuju su algoritmi korišteni za pojedini korak te njihova softverska implementacija. Sažet prikaz koraka pretprocesiranja za svaki cjevovod može se vidjeti u tablici 4.2. U ovome radu korišteni su podatci pretprocesirani s cjevovodom CPAC jer su Yang et al. [2] pokazali da daje najbolje rezultate u usporedbi s ostalim cjevovodima.

Tablica 4.1. Sažetak ABIDE podataka

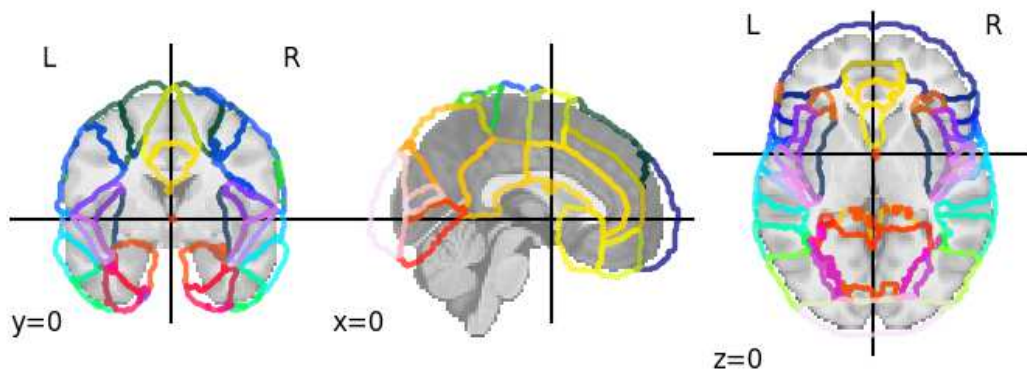
Institucija	Autizam	Kontrola	Muškarci	Žene	Godine	Ukupno
PITT	24	26	43	7	9~35	50
OLIN	14	14	23	5	10~24	28
OHSU	12	13	25	0	8~15	25
SDSU	8	19	21	6	9~17	27
TRINITY	19	25	44	0	12~26	44
UM	47	73	93	27	8~29	120
USM	43	24	67	0	9~50	67
YALE	22	19	25	16	7~18	41
CMU	6	5	7	4	19~33	11
LEUVEN	26	30	49	7	12~32	56
KKI	12	21	24	9	8~13	33
NYU	74	98	136	36	6~39	172
STANFORD	12	13	18	7	8~13	25
UCLA	48	37	74	11	8~18	85
MAX_MUN	19	27	42	4	7~58	46
CALTECH	5	10	10	5	17~56	15
SBL	12	14	26	0	20~49	26
Ukupno	403	468	727	144	6~58	871

Tablica 4.2. Sažet prikaz korak pretprocesiranja [2].

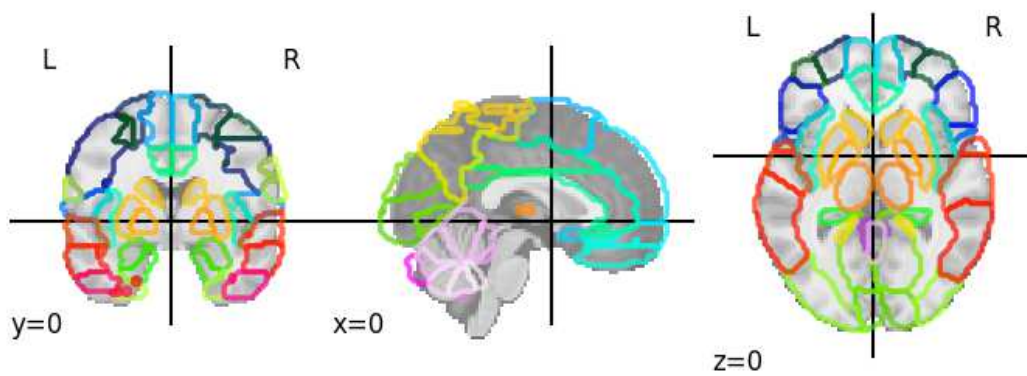
Step	CCS	CPAC	DPARF	NIAC
Drop first "N" volumes	4	0	4	0
Slice timing correction	Yes	Yes	Yes	Yes
Motion realignment	Yes	Yes	Yes	Yes
Motion	24-param	24-param	24-param	scrubbing
Tissue signals	mean WM and CSF signals	CompCor (5 PCs)	mean WM and CSF signals	mean WM and CSF signals
Low-frequency drifts	linear and quadratic trends	linear and quadratic trends	linear and quadratic trends	discrete cosine basis with a 0.01 Hz high-pass cut-off
Functional to Anatomical	boundary-based rigid body (BBR)	boundary-based rigid body (BBR)	rigid body	rigid body
Anatomical to Standard	FLIRT + FNIRT	ANTs	DARTEL	CIVET

4.3. Regije interesa i atlas

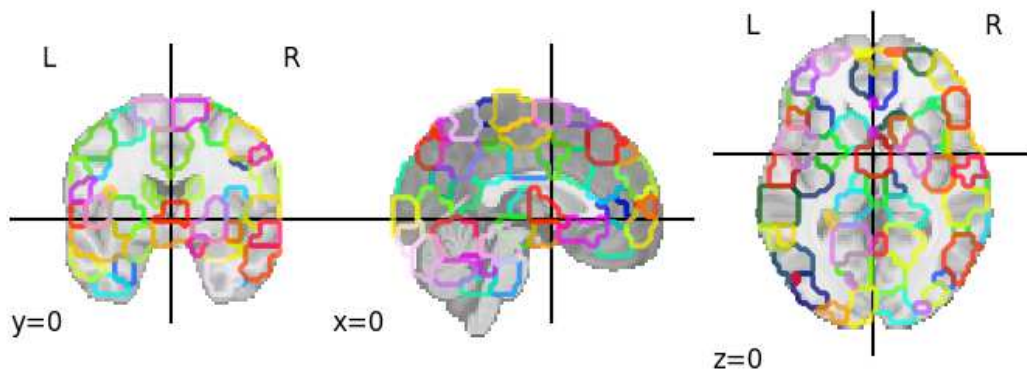
Ljudski je mozak mreža velikog broja različitih regija, od kojih svaka obavlja neki specifični zadatak ili funkciju. Ove regije su međusobno povezane strukturalno i funkcijski. Ključnu ulogu u kognitivnim procesima igra funkcijska komunikacija ovih regija [2]. S obzirom na to da ljudski mozak sadrži velik broj regija, prilikom analize podataka dobivenih funkcijskom magnetskom rezonancijom, odabire se manji broj regija koje se nazivaju regije interesa (engl. *regions of interest*, ROIs). Regije interesa koriste se kako bi se pobliže proučila aktivnost mozga unutar određenih skupova voksela (engl. *voxels*) koji su funkcijski usko povezani [33]. Riječ voksel dolazi od volumni piksel (engl. *volume pixel*) jer se prilikom snimanja mozga fMRI-jem mozak dijeli na male "kocke" koje su najčešće dimenzija 3 mm x 3 mm x 3.5 mm [34]. Više regija interesa sakupljeno je u atlasu. Atlasi se razlikuju po kojem kriteriju je podijeljen mozak. Neke od vrsta atlasa su: strukturalni atlasi koji dijele mozak u anatomske regije prema različitom tipu tkiva ili različitoj gustoći stanica [35]; funkcijski atlasi mozak razdjeljuju na regije prema njihovoj funkciji, koja je najčešće određena fMRI-jem [36]; vjerojatnosni atlas koji stvaraju statističku mapu mozga koja predstavlja vjerojatnost da određena regija bude prisutna na nekoj lokaciji [37]. Pri klasifikaciji autizma pomoću slika funkcijske magnetske rezonancije često korišteni atlasi su *Craddock 200* (CC200) s 200 ROI [38, 39, 6], *Craddock 400* (CC400) s 392 ROI [2, 40], *Automated Anatomical Labeling* (AAL) sa 116 ROI [7, 41] te *Harvard-Oxford Atlas* (HOA) s 110 ROI [7, 41]. Od ovih atlasa HAO je vjerojatnosni, AAL je strukturalni, a CC200 i CC400 su funkcionalni atlas. Na slikama 4.1., 4.2., 4.3. te 4.4. mogu se vidjeti vizualizacije ovih atlasa. U ovome radu korišten je atlas CC400 jer su Yang et al. [7] pokazali da daje najbolje rezultate.



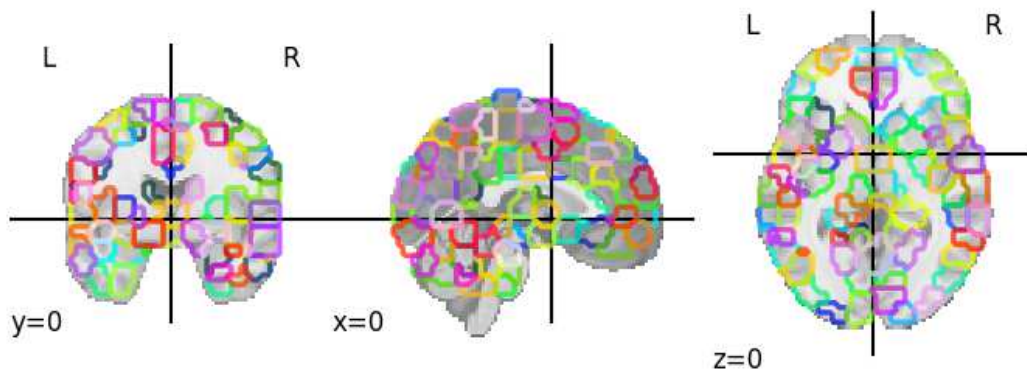
Slika 4.1. Vizualizacija HOA atlasa



Slika 4.2. Vizualizacija AAL atlasa



Slika 4.3. Vizualizacija CC200 atlasa



Slika 4.4. Vizualizacija CC400 atlasa

4.4. Ekstrakcija značajki

Kako bi se klasificirali pojedinci s autizmom i oni koji služe kao kontrola koristi se funkcijska povezanost različitih regija mozga. Funkcijska povezanost predstavlja snagu koaktivacije različitih regija temeljene na vremenskim serijama dobivenih iz fMRI snimaka [6]. Vremenske serije dobivene su na način da se najprije s pomoću atlasa, u ovome slučaju CC400 atlasa, odrede regije interesa. Zatim se za svaku regiju računa srednja vrijednost BOLD signala vokselu koji pripadaju određenoj regiji [42]. Iz dobivenih vremenskih serija zatim je izračunata matrica povezanosti korištenjem Pearsonove korelacije parova regija interesa. Svaka značajka povezanosti u matrici povezanosti predstavlja Pearsonov koeficijent korelacije dvije regije, ovi koeficijenti izračunati su po formuli 4.1 gdje su $x(t)$ i $y(t)$ vrijednosti vremenskih serija regija x i y u vremenskim točkama $t = 1, 2, \dots, T$, \bar{x} i \bar{y} predstavljaju srednje vrijednosti signala u regijama x i y [39].

$$r = \frac{\sum_{t=1}^T (x(t) - \bar{x})(y(t) - \bar{y})}{\sqrt{\sum_{t=1}^T (x(t) - \bar{x})^2} \sqrt{\sum_{t=1}^T (y(t) - \bar{y})^2}} \quad (4.1)$$

Vrijednosti koeficijenta su u intervalu od -1 do 1, gdje vrijednosti blizu 1 označavaju snažnu korelaciju između dvije regije, a vrijednosti blizu -1 označavaju anti-korelaciju regija. S obzirom na to da je matrica povezanosti simetrična, vrijednosti u gornjem trokutu i donjem trokutu matrice su jednake, donji trokut matrice je uklonjen te se samo vrijednosti gornjeg trokuta koriste kao značajke. Dodatno, na glavnoj dijagonali matrice nalaze se vrijednosti koje predstavljaju samokorelaciju, pa je ona također uklonjena. Na kraju je gornji trokut matrice vektoriziran, dvodimenzionalna matrica pretvorena je u jednodimenzionalni vektor značajki koji je u 5. poglavlju korišten kao ulaz u modele. Konačan broj značajki definiran je formulom 4.2, gdje je R broj značajki sadržanih u atlasu.

$$N = \frac{(R - 1)R}{2} \quad (4.2)$$

Za ekstrakciju značajki korišten je *Craddock 400* atlas, koji sadrži 392 značajke pa resultantni vektor sadrži 76.636 značajke.

4.4.1. *Nilearn* biblioteka

Za ekstrakciju vremenskih serija i računanje matrica povezanosti korištena je Python biblioteka *Nilearn*¹. Ova biblioteka napravljena je za analizu i primjenu metoda strojnog učenja na podatke dobivene magnetskom rezonancijom. Preciznije, iz *Nilearn* biblioteke korišten je objekt *NiftiLabelsMasker*² te njegova metoda *fit_transform()* za ekstrakciju vremenskih serija, a za računanje matrica povezanosti korišten je objekt *ConnectivityMeasure*³ te njegova metoda *fit_transform()*. Niže se može vidjeti inicijalizacija ova dva objekta.

```
1  masker = NiftiLabelsMasker(  
2  labels_img=atlas_filename,  
3  standardize="zscore_sample"  
4  )  
5  
6  correlation_measure = ConnectivityMeasure(  
7  kind="correlation",  
8  vectorize=True,  
9  discard_diagonal=True,  
10 standardize=True,  
11 )
```

Pri inicijalizaciji, objektu *NiftiLabelsMasker* kao vrijednost atributa *labels_img* prosljeđuje se datoteka u kojoj je zapisan atlas, vrijednost "zscore_sample" atributa *standardize* označava da će ekstrahirana vremenska serija biti standardizirana na načina da srednja vrijednost serije iznosi 0, a da standardna devijacija iznosi 1. Objektu *ConnectivityMeasure* se, pri inicijalizaciji, kao vrijednost atributa *kind* prosljeđuje vrijednost "correlation", što označava da će biti izračunata korelacija, također se mogu proslijediti vrijednosti "covariance", "partial correlation", "tangent" i "precision". Vrijednost *True* u atributima *vectorize*, *discard_diagonal* i *standardize* označavaju da će se odbaciti gornji trokut i dijagonala matrice te da će matrica biti standardizirana tako da joj je srednja vrijednost 0, a standardna devijacija 1.

¹<https://nilearn.github.io/index.html>

²<https://nilearn.github.io/dev/modules/generated/nilearn.maskers.NiftiLabelsMasker.html>

³<https://nilearn.github.io/dev/modules/generated/nilearn.connectome.ConnectivityMeasure.html>

5. Implementacija modela

5.1. *Tensorflow* i *Keras* biblioteke

Za implementaciju svih korištenih modela korištene su Python biblioteke *Tensorflow*¹ i *Keras*². *Tensorflow* je *open-source* Python biblioteka za strojno učenje koju je razvio Google. Često se koristi za izgradnju modela dubokog učenja zbog svoje fleksibilnosti i jednostavnosti korištenja. *Keras* je dio *Tensorflow* biblioteke te je napravljen za jednostavnu implementaciju, treniranje i testiranje modela dubokog učenja.

5.2. Arhitekture modela

Implementirano je šest različitih arhitektura modela koje slijede jedna iz druge. Svi modeli imaju jednake ulazne i izlazne slojeve. Broj umjetnih neurona ulaznog sloja jednak je broju elemenata u vektoru značajki opisanog u poglavlju 4.4., za njegovu implementaciju korišten je *InputLayer()* objekt iz biblioteke *Keras*. Izlazni sloj iz modela sadrži jedan neuron koji je povezan na sve izlaze iz neurona prethodnog sloja, za njegovu implementaciju korišten je objekt *Dense()* biblioteke *Keras*. U izlaznom sloju korištena je aktivacijska funkcija sigmoida koja je opisana u poglavlju 3.3. Ova funkcija određuje koliko je vjerojatnost da se uzorak nalazi u nekoj od klasa, ako joj je izlazna vrijednost veća od 0.5 uzorak pripada klasi 1, a ako je manja pripada klasi 0.

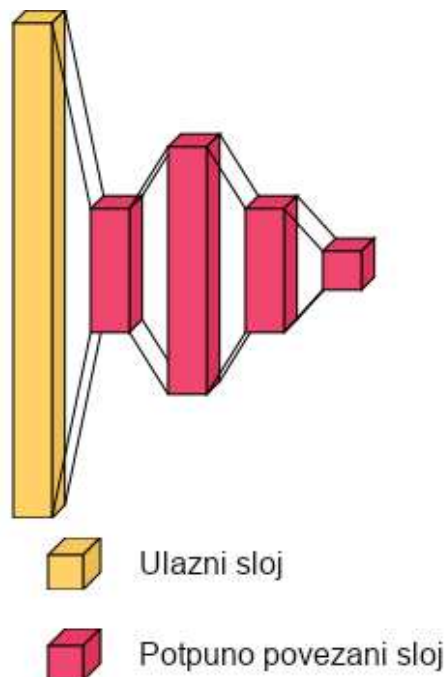
¹<https://www.tensorflow.org/>

²<https://keras.io/>

1. Model-1

Ova jednostavna arhitektura sastoji se od tri potpuno povezana skrivena sloja. Prvi od ovih slojeva ima 16 umjetnih neurona, sljedeći sloj sastoji se od 32 neurona, a posljednji sloj također ima 16 neurona. Niže se može vidjeti isječak koda koji predstavlja implementaciju ovog modela, a na slici 5.1. može se vidjeti vizualizacija arhitekture ovog modela.

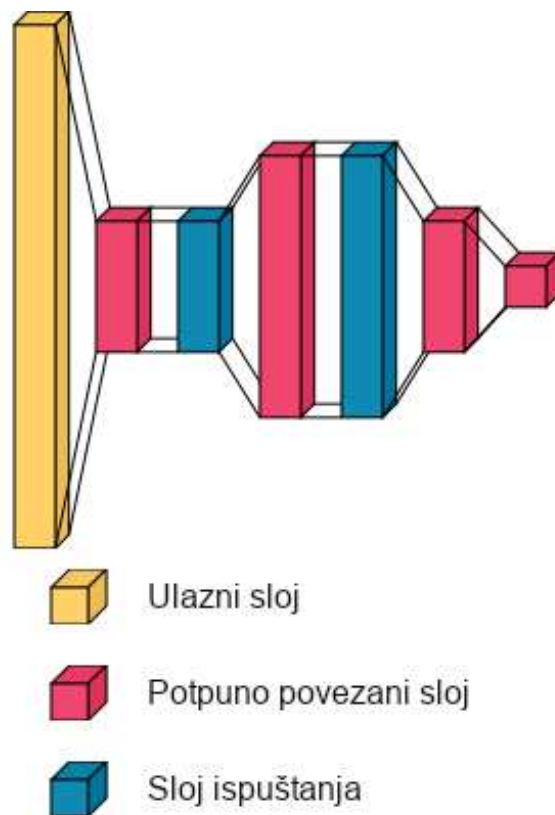
```
1     model = Sequential()  
2     model.add(InputLayer(shape=X_train.shape))  
3     model.add(Dense(16, activation="relu"))  
4     model.add(Dense(32, activation="relu"))  
5     model.add(Dense(16, activation="relu"))  
6     model.add(Dense(1, activation="sigmoid"))
```



Slika 5.1. Vizualizacija arhitekture modela-1

2. Model-2

Kako bi se spriječila prenaučenost modela arhitektura modela-1 proširena je sa slojevima ispuštanja (vidi poglavlje 3.4.). Osnovni skriveni slojevi su jednaki, no između svakog potpuno povezanog sloja dodan je sloj ispuštanja s vjerojatnošću ispuštanja neurona od 0.5. Na slici 5.2. može se vidjeti vizualizacija arhitekture ovog modela.

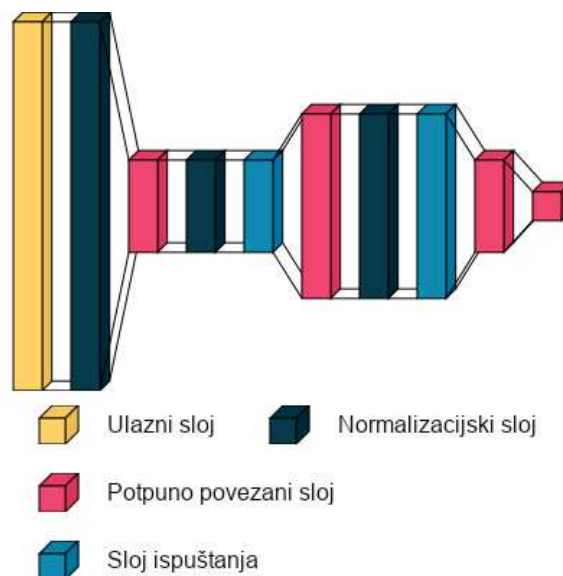


Slika 5.2. Vizualizacija arhitekture modela-2

3. Model-3

Kako bi se dodatno smanjila prenaučenosť modela, prethodni model se dodatno proširuje sa slojevima normalizacije (vidi poglavlje 3.4.). Slojevi normalizacije dodani su prije svakog sloja ispuštanja dodanog u prethodnom modelu te odmah nakon ulaznog sloja u mrežu. Niže se vidi prikaz implementacije ovog modela, a na slici 5.3. može se vidjeti vizualizacija arhitekture ovog modela..

```
1     model = Sequential()  
2     model.add(InputLayer(shape=X_train.shape))  
3     model.add(BatchNormalization())  
4     model.add(Dense(16, activation="relu"))  
5     model.add(BatchNormalization())  
6     model.add(Dropout(0.5))  
7     model.add(Dense(32, activation="relu"))  
8     model.add(BatchNormalization())  
9     model.add(Dropout(0.5))  
10    model.add(Dense(16, activation="relu"))  
11    model.add(Dense(1, activation="sigmoid"))
```

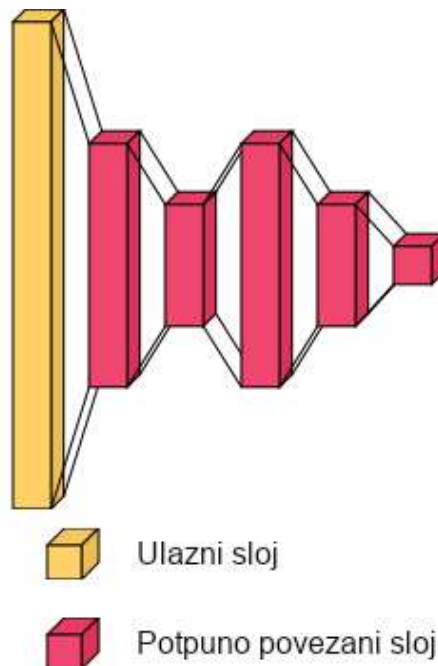


Slika 5.3. Vizualizacija arhitekture modela-3

4. Model-4

Arhitektura ovog modela je varijacija arhitekture modela-1. Ovaj model sadrži četiri skrivena potpuno povezana sloja. Prvi mu se sloj sastoji od 32 umjetna neurona, drugi sloj od 16 neurona, treći sloj ponovno ima 32 neurona te posljednji sloj ima 16 neurona. Na slici 5.4. može se vidjeti vizualizacija arhitekture ovog modela.

```
1     model = Sequential()  
2     model.add(InputLayer(shape=X_train.shape))  
3     model.add(Dense(32, activation="relu"))  
4     model.add(Dense(16, activation="relu"))  
5     model.add(Dense(32, activation="relu"))  
6     model.add(Dense(16, activation="relu"))  
7     model.add(Dense(1, activation="sigmoid"))
```

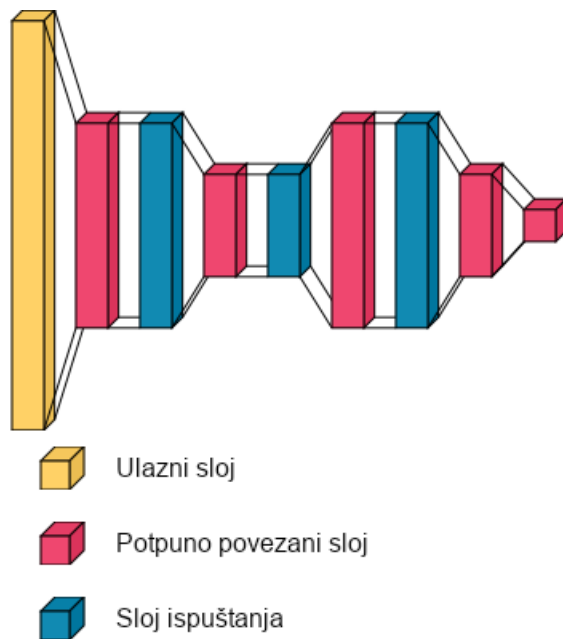


Slika 5.4. Vizualizacija arhitekture modela-4

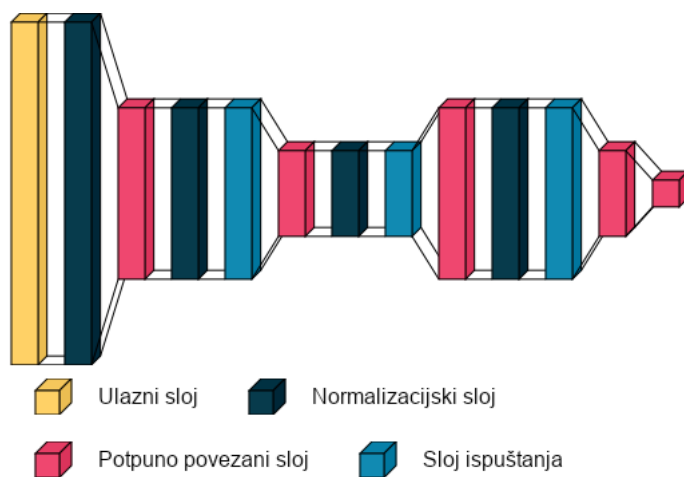
5. Model-5 i Model-6

Ovi modeli proširuju arhitekturu modela-4 slično kako model-2 i model-3 proširuju model-1. Arhitektura modela-5 na model-4 nadodaje slojeve ispuštanja između svakog potpuno povezanog sloja modela. Model-6 arhitekturu proširuje tako da, kao i model-3, između svakog potpuno povezanog sloja prije sloja ispuštanja te odmah nakon ulaznog sloja, dodaje sloj normalizacije. Niže se može vidjeti isječak koda koji prikazuje implementaciju modela-6, dok se na slici 5.5. može vidjeti vizualizacija arhitekture modela-5, a na slici 5.6. modela-6.

```
1     model = Sequential()
2     model.add(InputLayer(shape=X_train.shape))
3     model.add(BatchNormalization())
4     model.add(Dense(32, activation="relu"))
5     model.add(BatchNormalization())
6     model.add(Dropout(0.5))
7     model.add(Dense(16, activation="relu"))
8     model.add(BatchNormalization())
9     model.add(Dropout(0.5))
10    model.add(Dense(32, activation="relu"))
11    model.add(BatchNormalization())
12    model.add(Dropout(0.5))
13    model.add(Dense(16, activation="relu"))
14    model.add(Dense(1, activation="sigmoid"))
```



Slika 5.5. Vizualizacija arhitekture modela-5



Slika 5.6. Vizualizacija arhitekture modela-6

5.3. Treniranje, optimizacija i evaluacija modela

Prilikom treniranja modela korištene su različite vrijednosti hiperparametara. Za stope učenja modela korištene su vrijednosti 0,001 i 0,0001, a za veličine podskupa na koje se ulazni podaci dijele korištene su vrijednosti 64 te 256. Modeli su trenirani na svim kombinacijama ovih hiperparametara.

Za računanje gubitka modela na validacijskom skupu podataka korištena je zglobnica (vidi poglavlje 3.6.1.). Za optimizaciju modela korišten je objekt *Adam()* iz *Keras* biblioteke, ovaj objekt implementira verziju algoritma gradijentnog spusta 3.6.2., koristeći stopu učenja koja mu se zadaje pri inicijalizaciji za odluku koliko promijeniti unutarnje parametre modela kako bi optimizirao funkciju gubitka.

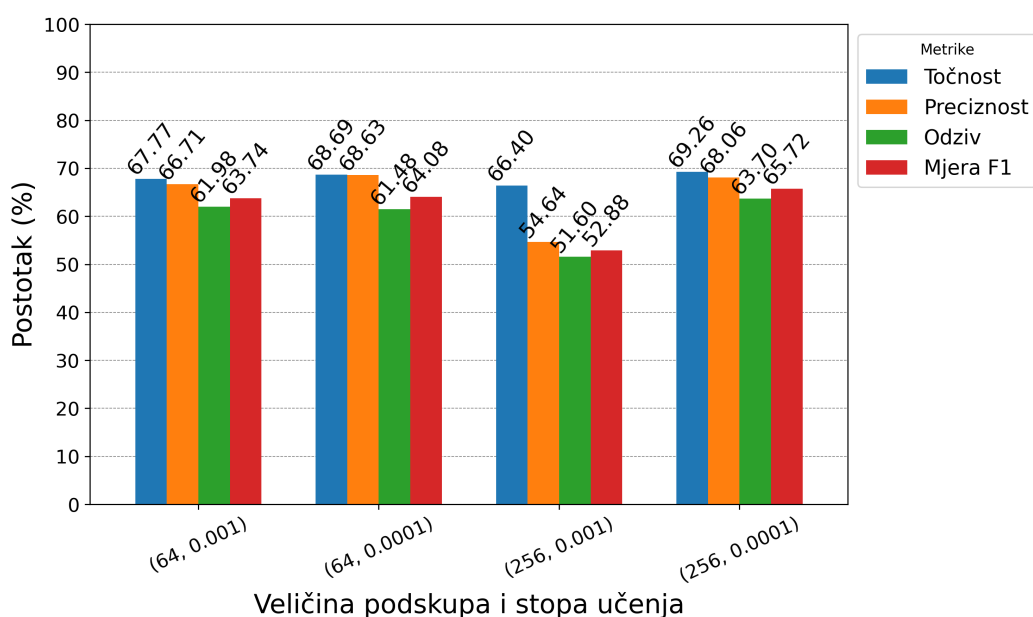
Kako bi se dodatno pokušalo spriječiti prenaučенost modela korišten je *callback ModelCheckpoint()* koji tijekom treniranja modela promatra neku vrijednost te sprema model kada je promatrana vrijednost najveća ili najmanja viđena do tog trenutka. U ovom slučaju bila je promatrana vrijednost validacijskog gubitka te se gledalo da je ta vrijednost najmanja. Taj se spremljeni model kasnije koristi za evaluaciju.

Zbog ograničene veličine početnog skupa podataka od 871 uzoraka, za evaluaciju modela korištena je k -struka unakrsna validacija, vrijednost k iznosila je 5. Na početku je iz početnog skupa podataka izdvojen skup podataka za testiranje koji je 20% veličine ukupnog skupa. Zatim je preostali dio podataka podijeljen na 5 dijelova od kojih je 4 služilo za treniranje, a 1 za validaciju. Prilikom treniranja modela na određenom preklopu pomoću *ModelCheckpoint callback*-a spreman je najbolji model, koji je na kraju evaluiran na skupu za testiranje. Nakon treniranja na svakom preklopu, kao ukupna procjena uspješnosti modela korištena je srednja vrijednost mjera uspješnosti na određenom preklopu.

6. Rezultati

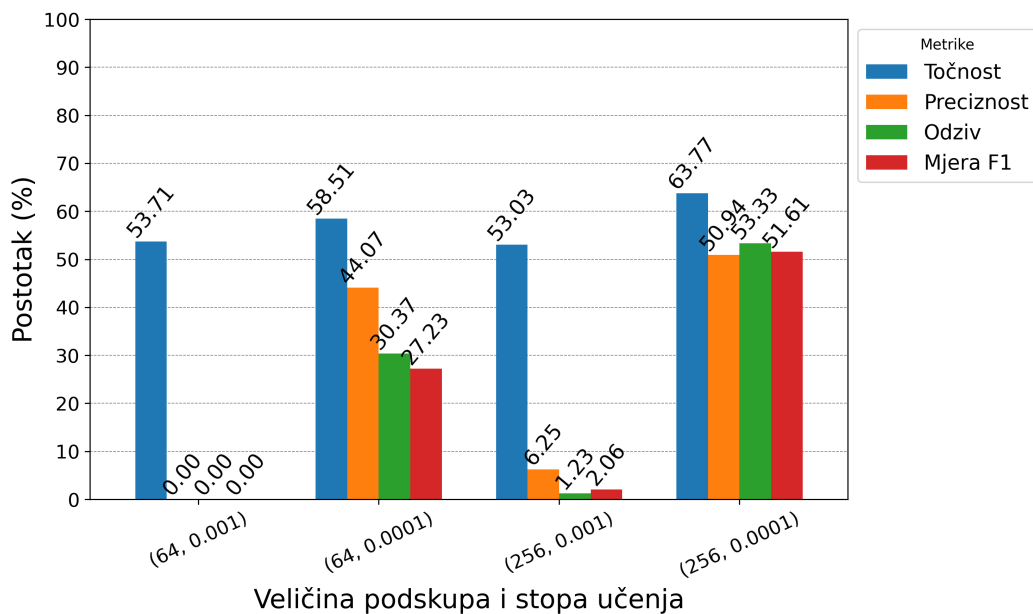
U tablici 6.1. može se vidjeti cjelokupan prikaz rezultata svih implementiranih modela. Modeli su testirani za različite kombinacije veličine podskupa na koje se dijele podaci za treniranje i stopa učenja. Za svaku tu kombinaciju izračunate su različite mjere uspješnosti modela (vidi poglavlje 3.8.). Zbog sažetosti pisanja različitih kombinacija veličine podskupa i stopa učenja označavat će se kao $[x, y]$, gdje x označava veličinu podskupa, a y stopu učenja.

Model-1 najbolje je rezultate u prosjeku ostvario za kombinaciju $[256, 0,0001]$, ostvarena je točnost od 69,26%, preciznost od 68,06%, odziv od 63,70% te mjeru F1 od 65,72%. Najlošiji rezultati dobiveni su za kombinaciju $[256, 0,001]$, ostvareno je 66,40% točnosti, 54,64% preciznosti, 51,60% odziva i 52,88% mjere F1. Slika 6.1. prikazuje rezultate modela-1.



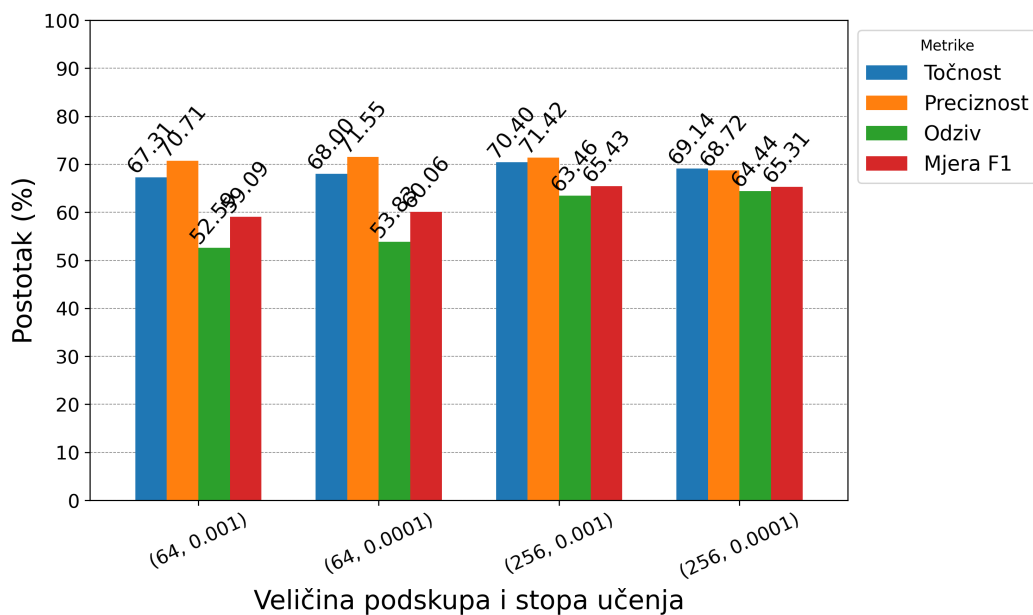
Slika 6.1. Vizualizacija rezultata modela-1

S namjerom poboljšanja uspješnosti i smanjenja prenaučeniosti modela-1, u njegovu arhitekturu dodani su slojevi ispuštanja, što čini arhitekturu modela-2. No, model-2 ostvario je znatno lošije rezultate od modela-1 u svim kombinacijama veličina podskupa i stopa učenja. Model-2 najlošije rezultate ostvario je za [64, 0,001] kombinaciju, gdje točnost iznosi 53,71%, a preciznost, odziv i mjera F1 svi iznose 0,00%, što govori da model nije predvidio niti jednu pozitivnu klasu. Najbolji rezultati koje je model-2 ostvario su za kombinaciju [256, 0,0001], točnost iznosi 63,77%, preciznost 50,94%, odziv 53,33%, a mjera F1 51,61%. Zanimljivo je uočiti da je sloj ispuštanja pogoršao uspješnost modela, suprotno očekivanjima, što ukazuje na to da su slojevi ispuštanja unijeli nepoželjnu promjenu u model. Analiza ove pojave ima potencijal biti proširenje ovog rada. Rezultati modela-2 mogu se vidjeti na slici 6.2.



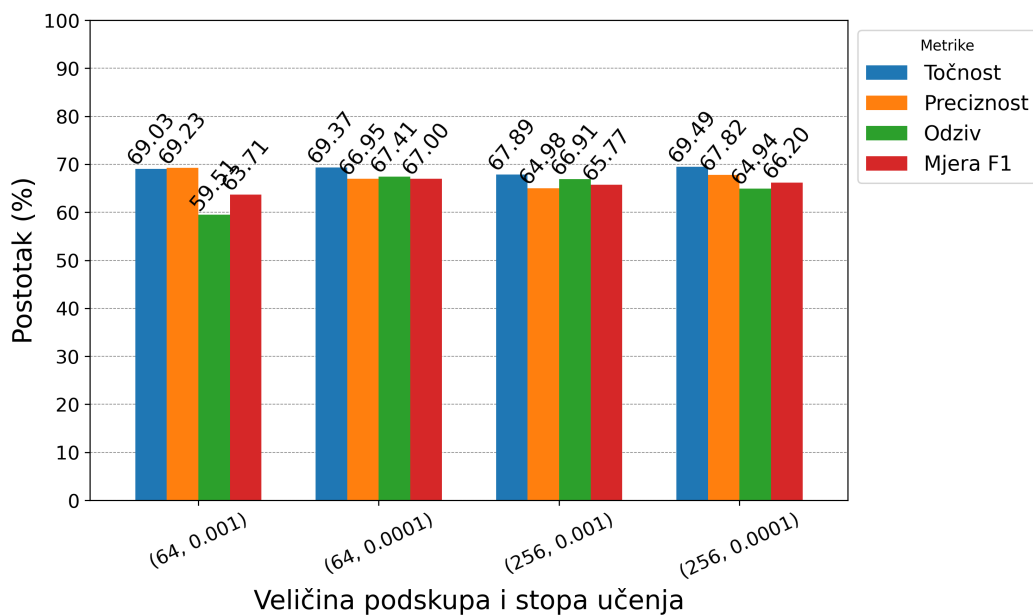
Slika 6.2. Vizualizacija rezultata modela-2

Model-3 proširuje arhitekturu modela-2 dodavanjem normalizacijskih slojeva. Rezultati ovog modela slični su kao rezultati modela-1, za kombinaciju [256, 0,0001] ovi modeli ostvaruju gotovo jednake rezultate. Također, može se vidjeti da model-3 ima znatno poboljšanje u preciznosti za sve kombinacije veličine podskupa i stope učenja, a najveća preciznost iznosi 71,55% za kombinaciju [64, 0,0001]. Rezultati ovog modela mogu se vidjeti na slici 6.3.



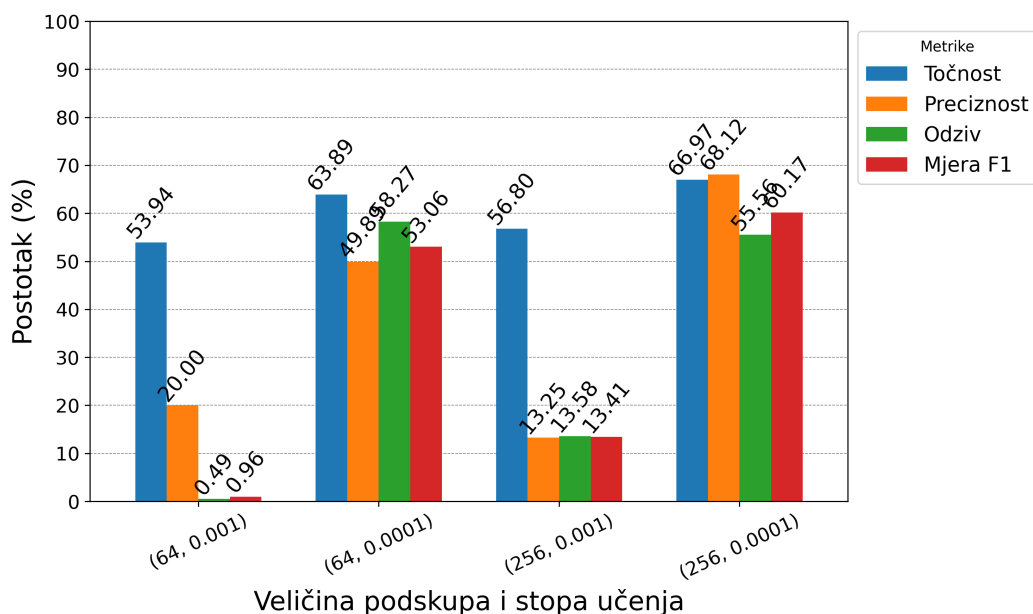
Slika 6.3. Vizualizacija rezultata modela-3

Model-4 u prosjeku ostvaruje bolju uspješnost od prethodno navedenih modela, no model-3 je uspješniji od njega po mjeri preciznosti. Model-4 najbolju točnost od 69,49% ostvaruje za kombinaciju [256, 0,0001], najbolju preciznost od 69,23% za [64, 0,001], a najbolji odziv od 67,41% i mjeru F1 od 67,00% postiže za kombinaciju [64, 0,0001]. Najlošija točnost od 67,89% i preciznost od 64,98% dobivena je za kombinaciju [256, 0,001], dok su najlošiji odziv od 59,51% i mjera F1 63,71% postignuti za kombinaciju [64, 0,001]. Slika 6.4.



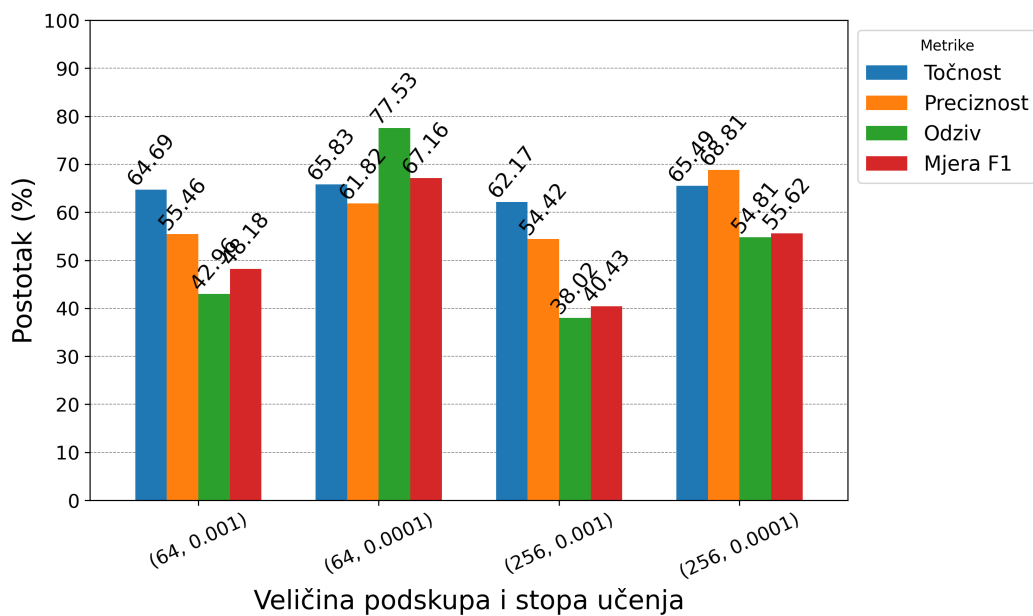
Slika 6.4. Vizualizacija rezultata modela-4

Za model-5, u kojemu se dodavanjem slojeva ispuštanje trebala poboljšati uspješnost modela, dogodila se slična pojava kao i za model-2, uspješnost modela znatno je pala. Za kombinaciju [64, 0,001] model je ostvario najgore rezultate; 53,94% točnosti, 20,00% preciznosti te samo 0,49% odziva i 0,96% mjere F1. Najbolje rezultate model je ostvario za kombinaciju [256, 0,0001]; točnost od 66,97%, preciznost od 68,12%, odziv od 55,56% i mjeru F1 od 60,17%. Rezultati modela-5 mogu se vidjeti na slici 6.5.



Slika 6.5. Vizualizacija rezultata modela-5

Dodavanjem normalizacijskih slojeva u modelu-6 uspješnost modela poboljšala se u usporedbi s modelom-5. U usporedbi s modelom-4, model-6 u prosjeku ostvaruje manju uspješnost; jedina mjera za koju je model-6 znatno bolji od modela-4 je odziv od 77,53%, dok model-4 ima 67,41% preciznosti za kombinaciju [64, 0,0001]. U prosjeku model-6 najbolju uspješnost ostvaruje za kombinaciju [64, 0,0001] s ostvarenom točnošću od 65,83%, preciznošću od 61,82%, odzivom od 77,53% te mjerom F1 od 67,16%. Rezultati ovog modela prikazani su na slici 6.6.



Slika 6.6. Vizualizacija rezultata modela-6

Tablica 6.1. Prikaz rezultata svih implementiranih modela (najbolje vrijednosti su podebljane)

Naziv modela	Veličina podskupa	Stopa učenja	Točnost	Preciznost	Odziv	Mjera F1
Model-1	64	0,001	67,77%	66,71%	61,98%	63,74%
		0,0001	68,69%	68,63%	61,48%	64,08%
	256	0,001	66,40%	54,64%	51,60%	52,88%
		0,0001	69,26%	68,06%	63,70%	65,72%
Model-2	64	0,001	53,71%	0,00%	0,00%	0,00%
		0,0001	58,51%	44,07%	30,37%	27,23%
	256	0,001	53,03%	6,25%	1,23%	2,06%
		0,0001	63,77%	50,94%	53,33%	51,61%
Model-3	64	0,001	67,31%	70,71%	52,59%	59,09%
		0,0001	68,00%	71,55%	53,83%	60,06%
	256	0,001	70,40%	71,42%	63,46%	65,43%
		0,0001	69,14%	68,72%	64,44%	65,31%
Model-4	64	0,001	69,03%	69,23%	59,51%	63,71%
		0,0001	69,37%	66,95%	67,41%	67,00%
	256	0,001	67,89%	64,98%	66,91%	65,77%
		0,0001	69,49%	67,82%	64,94%	66,20%
Model-5	64	0,001	53,94%	20,00%	0,49%	0,96%
		0,0001	63,89%	49,89%	58,27%	53,06%
	256	0,001	56,80%	13,25%	13,58%	13,41%
		0,0001	66,97%	68,12%	55,56%	60,17%
Model-6	64	0,001	64,69%	55,46%	42,96%	48,18%
		0,0001	65,83%	61,82%	77,53%	67,16%
	256	0,001	62,17%	54,42%	38,02%	40,43%
		0,0001	65,49%	68,81%	54,81%	55,62%

7. Zaključak

U ovome radu duboko učenje je primijenjeno na problem klasifikacije poremećaja iz spektra autizma korištenjem slika dobivenih funkcijskom magnetskom rezonancijom mozga. U radu je korišten skup podataka ABIDE, preciznije korišteno je 871 uzorak iz tog skupa. Od 871 uzorka njih 403 dijagnosticirano je s autizmom, preostalih 468 su kontrolni uzorci. Preuzeti podaci bili su unaprijed preprocesirani cjevovodom CPAC, iz tih podataka ekstrahirane su vremenske serije pomoću *Craddock 400* atlasa za svaki uzorak te su iz toga zatim izračunate matrice funkcijske povezanosti koje su se zatim vektoralizirale te služile kao ulaz u modele dubokog učenja. Implementirano je šest različitih arhitektura modela dubokog učenja, postoje dvije osnovne arhitekture modela koje su se nadograđivale dodavanjem slojeva za ispuštanje, a zatim dodavanjem normalizacijskih slojeva.

Modeli su evaluirani 5-strukom unakrsnom validacijom te kao ukupna mjera uspješnosti modela korištena je srednja vrijednost mjera uspješnosti najboljeg modela svakog preklopa na skupu podataka za testiranje. Bilo je pretpostavljeno da će se dodavanjem slojeva za ispuštanje uspješnost modela povećati te da će se smanjiti prenaučenos modela, no dogodilo se upravo suprotno. Ovo može biti posljedica toga što ulazni podatci imaju veliku dimenzionalnost, pa ako se previše neurona modela ispusti u isto vrijeme model ne može naučiti sve potrebne značajke što dovodi do loše uspješnosti.

Što se tiče točnosti, model-3 pokazao je najveću točnost od svih modela. Model je postigao točnost od 70,40% za veličinu podjele skupa podataka za treniranje od 256 i stope učenja od 0.001. Ovaj rezultat može se usporediti s modelom koji su implementirali Yang et al. [2] koji ima arhitekturu od dva skrivena sloja, jednog s 128 neurona te drugog s 64

neurona. Njihov model postigao je točnost od 75,27%, što je veće od točnosti modela-3, međutim, autori su koristili 1035 uzoraka umjesto 871 što može biti uzrok veće točnosti.

S obzirom na to da se u ovome radu radi klasifikacija autizma, kao najbolja mjera procjene uspješnosti modela bila bi mjera F1, koja objedinjuje preciznost i odziv modela, tj. ona je njihova harmonijska sredina. Preciznost modela je omjer uspješno klasificiranih pacijenata s poremećajem iz spektra autizma i svih pacijenata klasificiranih s autizmom, dok je odziv modela omjer uspješno klasificiranih pacijenata s autizmom i svih pacijenata s autizmom. Model-6 ostvario je najbolju mjeru F1 od 67,16% za [64, 0.0001], a model-4 ostvario je nešto nižu mjeru F1 od 67,00% također za [64, 0.0001].

Bolja uspješnost modela mogla bi se postići korištenjem složenijih arhitektura modela, kao što su konvolucijski modeli ili modeli s povratnom vezom te korištenjem neke naprednije metode optimizacije. Također, za poboljšanje sposobnosti klasifikacije mogao bi se povećati skup podataka, bilo pronalaženjem potpuno novih uzoraka ili augmentacijom već postojećih kako bi se dobilo više podataka za treniranje.

Literatura

- [1] S. Sharma, X. Gonda, i F. Tarazi, “Autism Spectrum Disorder: Classification, diagnosis and therapy”, *Pharmacology & Therapeutics*, sv. 190, svibanj 2018. <https://doi.org/10.1016/j.pharmthera.2018.05.007>
- [2] X. Yang, P. T., i N. Zhang, “A Deep Neural Network Study of the ABIDE Repository on Autism Spectrum Classification”, *International Journal of Advanced Computer Science and Applications*, sv. 11, siječanj 2020. <https://doi.org/10.14569/IJACSA.2020.0110401>
- [3] “How FMRI works”, [pristupljeno: 6. 4. 2024.]. [Mrežno]. Adresa: <https://www.open.edu/openlearn/body-mind/health/health-sciences/how-fmri-works>
- [4] M. A. Aghdam, A. Sharifi, i M. M. Pedram, “Diagnosis of Autism Spectrum Disorders in Young Children Based on Resting-State Functional Magnetic Resonance Imaging Data Using Convolutional Neural Networks”, *Journal of Digital Imaging*, sv. 32, br. 6, str. 899–918, prosinac 2019. <https://doi.org/10.1007/s10278-019-00196-1>
- [5] M. A. Bayram, I. Özer, i F. Temurtaş, “Deep Learning Methods for Autism Spectrum Disorder Diagnosis Based on fMRI Images”, *Sakarya University Journal of Computer and Information Sciences*, sv. 4, br. 1, str. 142–155, travanj 2021. <https://doi.org/10.35377/saucis.04.01.879735>
- [6] A. S. Heinsfeld, A. R. Franco, R. C. Craddock, A. Buchweitz, i F. Meneguzzi, “Identification of autism spectrum disorder using deep learning and the ABIDE dataset”, *NeuroImage: Clinical*, sv. 17, str. 16–23, siječanj 2018. <https://doi.org/10.1016/j.nicl.2017.08.017>

- [7] X. Yang, M. S. Islam, i A. M. A. Khaled, “Functional connectivity magnetic resonance imaging classification of autism spectrum disorder using the multisite ABIDE dataset”, u *2019 IEEE EMBS International Conference on Biomedical & Health Informatics (BHI)*, svibanj 2019., str. 1–4, iSSN: 2641-3604. <https://doi.org/10.1109/BHI.2019.8834653>
- [8] Y. Baştanlar i M. Özuysal, “Introduction to Machine Learning”, u *miRNomics: MicroRNA Biology and Computational Analysis*, M. Yousef i J. Allmer, Ur. Totowa, NJ: Humana Press, 2014., sv. 1107, str. 105–128, series Title: Methods in Molecular Biology. https://doi.org/10.1007/978-1-62703-748-8_7
- [9] Y. LeCun, Y. Bengio, i G. Hinton, “Deep learning”, *Nature*, sv. 521, br. 7553, str. 436–444, svibanj 2015., publisher: Nature Publishing Group. <https://doi.org/10.1038/nature14539>
- [10] I. Goodfellow, Y. Bengio, i A. Courville, *Deep Learning*. MIT Press, 2016., [pristupljeno: 14. 5. 2024.]. [Mrežno]. Adresa: <https://www.deeplearningbook.org>
- [11] K. Suzuki, *Artificial Neural Networks - Methodological Advances and Biomedical Applications*, travanj 2011.
- [12] K. S. Ganesh. What’s The Role Of Weights And Bias In a Neural Network? [pristupljeno: 14. 5. 2024.]. [Mrežno]. Adresa: <https://towardsdatascience.com/whats-the-role-of-weights-and-bias-in-a-neural-network-4cf7e9888a0f>
- [13] A. D. Rasamoelina, F. Adjailia, i P. Sinčák, “A Review of Activation Function for Artificial Neural Network”, u *2020 IEEE 18th World Symposium on Applied Machine Intelligence and Informatics (SAMI)*, siječanj 2020., str. 281–286. <https://doi.org/10.1109/SAMI48414.2020.9108717>
- [14] S. Sharma, S. Sharma, i A. Athaiya, “ACTIVATION FUNCTIONS IN NEURAL NETWORKS”, *International Journal of Engineering Applied Sciences and Technology*, sv. 04, br. 12, str. 310–316, svibanj 2020. <https://doi.org/10.33564/IJEAST.2020.v04i12.054>
- [15] G. Ognjanovski, “Everything you need to know about Neural Networks and

Backpropagation — Machine Learning Made Easy...”, [pristupljeno: 14. 5. 2024.]. [Mrežno]. Adresa: <https://towardsdatascience.com/everything-you-need-to-know-about-neural-networks-and-backpropagation-machine-learning-made-easy-e5285bc2be3a>

- [16] Jorgecardete, “Convolutional Neural Networks: A Comprehensive Guide”, [pristupljeno: 15. 5. 2024.]. [Mrežno]. Adresa: <https://medium.com/thedeephub/convolutional-neural-networks-a-comprehensive-guide-5cc0b5eae175>
- [17] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, i R. Salakhutdinov, “Dropout: A Simple Way to Prevent Neural Networks from Overfitting”.
- [18] S. Santurkar, D. Tsipras, A. Ilyas, i A. Madry, “How Does Batch Normalization Help Optimization?” u *Advances in Neural Information Processing Systems*, sv. 31. Curran Associates, Inc., [pristupljeno: 15. 5. 2024.]. [Mrežno]. Adresa: https://proceedings.neurips.cc/paper_files/paper/2018/hash/905056c1ac1dad141560467e0a99e1cf-Abstract.html
- [19] I. Muraina, *Ideal Dataset Splitting Ratios In Machine Learning Algorithms: General Concerns for Data Scientists and Data Analysts*, veljača 2022.
- [20] “Model Training”, [pristupljeno: 16. 5. 2024.]. [Mrežno]. Adresa: <https://c3.ai/glossary/data-science/model-training/>
- [21] “Model Training”, [pristupljeno: 16. 5. 2024.]. [Mrežno]. Adresa: <https://c3.ai/glossary/data-science/model-training/>
- [22] F. Maleki, N. Muthukrishnan, K. Ovens, C. Reinhold, i R. Forghani, “Machine Learning Algorithm Validation: From Essentials to Advanced Applications and Implications for Regulatory Certification and Deployment”, *Neuroimaging Clinics*, sv. 30, br. 4, str. 433–445, studeni 2020., publisher: Elsevier. <https://doi.org/10.1016/j.nic.2020.08.004>
- [23] J. R. Choudhary, “What is Model Validation.” [pristupljeno: 16. 5. 2024.]. [Mrežno]. Adresa: <https://medium.com/analytics-vidhya/what-is-model-validation-257686d0253e>

- [24] J. Brownlee, “Loss and Loss Functions for Training Deep Learning Neural Networks”, [pristupljeno: 14. 4. 2024.]. [Mrežno]. Adresa: <https://machinelearningmastery.com/loss-and-loss-functions-for-training-deep-learning-neural-networks/>
- [25] “What Is Underfitting? | IBM”, [pristupljeno: 18. 4. 2024.]. [Mrežno]. Adresa: <https://www.ibm.com/topics/underfitting>
- [26] “What is Overfitting? | IBM”, [pristupljeno: 18. 4. 2024.]. [Mrežno]. Adresa: <https://www.ibm.com/topics/overfitting>
- [27] O. Rainio, J. Teuvo, i R. Klén, “Evaluation metrics and statistical tests for machine learning”, *Scientific Reports*, sv. 14, br. 1, str. 6086, ožujak 2024., publisher: Nature Publishing Group. <https://doi.org/10.1038/s41598-024-56706-x>
- [28] “Classification: True vs. False and Positive vs. Negative | Machine Learning”, [pristupljeno: 5. 5. 2024.]. [Mrežno]. Adresa: <https://developers.google.com/machine-learning/crash-course/classification/true-false-positive-negative>
- [29] “What is A Confusion Matrix in Machine Learning? The Model Evaluation Tool Explained”, [pristupljeno: 19. 4. 2024.]. [Mrežno]. Adresa: <https://www.datacamp.com/tutorial/what-is-a-confusion-matrix-in-machine-learning>
- [30] P. Huilgol, “Precision and Recall | Essential Metrics for Machine Learning (2024 Update)”, [pristupljeno: 17. 4. 2024.]. [Mrežno]. Adresa: <https://www.analyticsvidhya.com/blog/2020/09/precision-recall-machine-learning/>
- [31] “ABIDE”, [pristupljeno: 19. 3. 2024.]. [Mrežno]. Adresa: https://fcon_1000.projects.nitrc.org/indi/abide/
- [32] “ABIDE Preprocessed”, [pristupljeno: 19. 3. 2024.]. [Mrežno]. Adresa: <http://preprocessed-connectomes-project.org/abide/index.html>
- [33] R. A. Poldrack, “Region of interest analysis for fMRI”, *Social Cognitive and Affective Neuroscience*, sv. 2, br. 1, str. 67–70, ožujak 2007. <https://doi.org/10.1093/scan/nsm006>

- [34] F. G. Ashby, “An Introduction to fMRI”, u *An Introduction to Model-Based Cognitive Neuroscience*, B. U. Forstmann i E.-J. Wagenmakers, Ur. New York, NY: Springer New York, 2015., str. 91–112. https://doi.org/10.1007/978-1-4939-2236-9_5
- [35] P. K. Mandal, R. Mahajan, i I. D. Dinov, “Structural Brain Atlases: Design, Rationale, and Applications in Normal and Pathological Cohorts”, *Journal of Alzheimer’s disease : JAD*, sv. 31, br. 0 3, str. S169–S188, 2012. <https://doi.org/10.3233/JAD-2012-120412>
- [36] R. C. Craddock, G. James, P. E. Holtzheimer III, X. P. Hu, i H. S. Mayberg, “A whole brain fMRI atlas generated via spatially constrained spectral clustering”, *Human Brain Mapping*, sv. 33, br. 8, str. 1914–1928, 2012., _eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1002/hbm.21333>. <https://doi.org/10.1002/hbm.21333>
- [37] P. Thompson, “Probabilistic Brain Atlases”, [pristupljeno: 2. 4. 2024.]. [Mrežno]. Adresa: http://users.loni.ucla.edu/~thompson/prob_atlas.html
- [38] A. Karampasi, I. Kakkos, S.-T. Miloulis, I. Zorzos, G. N. Dimitrakopoulos, K. Gkiatis, P. Asvestas, i G. Matsopoulos, “A Machine Learning fMRI Approach in the Diagnosis of Autism”, u *2020 IEEE International Conference on Big Data (Big Data)*, prosinac 2020., str. 3628–3631. <https://doi.org/10.1109/BigData50022.2020.9378453>
- [39] Y. Liang, B. Liu, i H. Zhang, “A Convolutional Neural Network Combined With Prototype Learning Framework for Brain Functional Network Classification of Autism Spectrum Disorder”, *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, sv. 29, str. 2193–2202, 2021., conference Name: IEEE Transactions on Neural Systems and Rehabilitation Engineering. <https://doi.org/10.1109/TNSRE.2021.3120024>
- [40] Z. Sherkatghanad, M. Akhondzadeh, S. Salari, M. Zomorodi-Moghadam, M. Abdar, U. R. Acharya, R. Khosrowabadi, i V. Salari, “Automated Detection of Autism Spectrum Disorder Using a Convolutional Neural Network”, *Frontiers in Neuroscience*, sv. 13, siječanj 2020., publisher: Frontiers. <https://doi.org/10.3389/fnins.2019.01325>

- [41] M. Khosla, K. Jamison, A. Kuceyeski, i M. R. Sabuncu, “Ensemble learning with 3D convolutional neural networks for connectome-based prediction”, lipanj 2019., arXiv:1809.06219 [cs, stat]. <https://doi.org/10.48550/arXiv.1809.06219>
- [42] G. Varoquaux i R. C. Craddock, “Learning and comparing functional connectomes across subjects”, *NeuroImage*, sv. 80, str. 405–415, listopad 2013. <https://doi.org/10.1016/j.neuroimage.2013.04.007>

Sažetak

Usporedba modela strojnog učenja za klasifikaciju poremećaja iz spektra autizma korištenjem slika funkcijske magnetske rezonancije mozga

Dominik Pavelić

Poremećaji iz spektra autizma jedni su od najzastupljenijih neurorazvojnih poremećaja, tradicionalne metode dijagnoze često su spore i subjektivne. U ovome radu uspoređene su različite arhitekture modela dubokog učenja za klasifikaciju poremećaja iz spektra autizma korištenjem slika funkcijske magnetske rezonancije (fMRI). Analiza je provedena na 871 uzoraku iz ABIDE (*Autism Brain Imaging Data Exchange*) skupa podataka. Za treniranje modela korištene su matrice funkcijske povezanosti dobivene iz vremenskih serija ekstrahiranih iz 4D fMRI slika. Za model s tri skrivena sloj postignute su najveća točnost od 70,40% i preciznost od 71,55%, a najveći odziv od 77,53% i mjera F1 od 67,16% postignuti su za model s četiri skrivena sloja.

Ključne riječi: duboko učenje; klasifikacije; autizam; fMRI; ABIDE

Abstract

Comparison of machine learning models for the autism spectrum disorder classification using brain functional magnetic resonance images

Dominik Pavelić

Autism spectrum disorder (ASD) is among the most prevalent neurodevelopmental disorders, and traditional diagnostic methods are often slow and subjective. In this study various deep learning model architectures were compared for classifying ASD using functional magnetic resonance imaging (fMRI) images. The analysis was conducted on 871 samples from the ABIDE (Autism Brain Imaging Data Exchange) dataset. Functional connectivity matrices derived from time series extracted from 4D fMRI images were used for training the models. The highest accuracy of 70,40% and precision of 71,55% were achieved by a model with three hidden layers, while the highest recall of 77,53% and F1-score of 67,16% were achieved by a model with four hidden layers.

Keywords: deep learning; classification; ASD; fMRI; ABIDE