Primjena dubokog učenja u genetici

Županović, Luka

SVEUČILIŠTE U RIJECI

TEHNIČKI FAKULTET

Diplomski sveučilišni studij računarstva

Diplomski rad

Primjena dubokog učenja u genetici

Rijeka, studeni 2024.

Luka Županović

0069085842

SVEUČILIŠTE U RIJECI

TEHNIČKI FAKULTET

Diplomski sveučilišni studij računarstva

Diplomski rad

Primjena dubokog učenja u genetici

Mentor: izv.prof.dr.sc. Goran Mauša

Rijeka, studeni 2024.

Luka Županović

0069085842

SVEUČILIŠTE U RIJECI
TEHNIČKI FAKULTET
POVJERENSTVO ZA DIPLOMSKE ISPITE

Rijeka, 15.03.2024.

Zavod: Zavod za računarstvo
Predmet: Evolucijsko računarstvo

ZADATAK ZA DIPLOMSKI RAD

Pristupnik: **Luka Županović (0069085842)**
Studij: Sveučilišni diplomski studij računarstva (1400)
Modul: Programsko inženjerstvo (1441)

Zadatak: **Primjena dubokog učenja u genetici / Applications of deep learning in genetics**

Opis zadatka:

Analizirati mogućnosti primjene dubokog učenja u području genetike, s naglaskom na predviđanje genskog izražaja. Objasniti važnost korištenja metoda dubokog učenja i potencijal koji nudi u boljem razumijevanju genoma i medinskom napretku kroz pregled najnovijih istraživačkih radova. Objasniti važnost nekodirajuće DNA pri genskoj ekspresiji i izazove vezane uz istu koji se potencijalno mogu riješiti pomoću dubokog učenja. Analizirati prednosti i izazove postojećih modela.

Rad mora biti napisan prema Uputama za pisanja diplomskih / završnih radova koje su objavljene na mrežnim stranicama studija.

Zadatak uručen pristupniku: 20.03.2024.

Predsjednik povjerenstva za diplomski ispit:
prof. dr. sc. Miroslav Joler

Mentor:
izv. prof. Goran Mauša

Izjava o samostalnoj izradi rada

Izjavljujem da sam samostalno izradio ovaj rad.

Rijeka, studeni 2024.

_____

Luka Županović

Zahvala

Ovom prilikom zahvaljujem mentoru izv.prof. dr. sc. Goranu Mauši na uloženom vremenu, korisnim smjernicama i na svoj pomoći pri izradi ovog diplomskog rada.

Također, velike zahvale upućujem svojoj obitelji i prijateljima na podršci tijekom studiranja.

Sadržaj

Popis slika

Popis tablica

1. Uvod

U posljednje vrijeme umjetna inteligencija, a posebice duboko učenje koje po mišljenju mnogih posjeduje ogroman potencijal za revolucionarizirati gotovo sva područja ljudskog djelovanja, profilirali su se kao najaktualnije teme u svijetu računarstva, a i tehnologije općenito. Ovo je posebice izraženo od pojave značajnijih napredaka velikih jezičnih modela (engl. *large language model* - LLM). Područja u kojem duboko učenje nalazi svoju primjenu svakako su genetika i genomika te ono, uz preduvjet poznavanja ostalih disciplina poput biologije, statistike i matematike, predstavlja moćan alat bioinformatičarima za važne znanstvene i tehnološke inovacije. Zbog svoje prirode učenja kompleksnih struktura iz podataka, duboko učenje se nudi kao izvrstan odabir kada je u pitanju izvlačenje novih spoznaja iz izuzetno kompleksnih i visoko dimenzionalnih podataka ljudskog genoma [30]. Tijekom posljednjih desetljeća razvijene su razne strategije za proučavanje mRNK (engl. *messenger ribonucleic acid*, mRNA) sa svrhom dobivanja boljeg uvida u funkcioniranje stanica. Iste se mogu podijeliti u 3 općenite kategorije. Prvu kategoriju čine klasične metode poput primjerice QRT-PCR-a. One su prethodile drugim dvjema kategorijama - niskopropusnim i danas često korištenim visokopropusnim metodama [7]. Analiza kompleksnih podataka ljudskog genoma bitna je iz razloga što može dovesti do otkrivanja raznih uzoraka ili novih spoznaja koje, među ostalim, mogu biti korištene i za bolje razumijevanje uzroka bolesti, a posljedično i liječenja. Podaci koji su sredstvo novih spoznaja dolaze u mnoštvu oblika, a kao jedan od njih u kojem duboko učenje nalazi svoju primjenu jesu podaci genske ekspresije koji su i sami po sebi raznoliki.

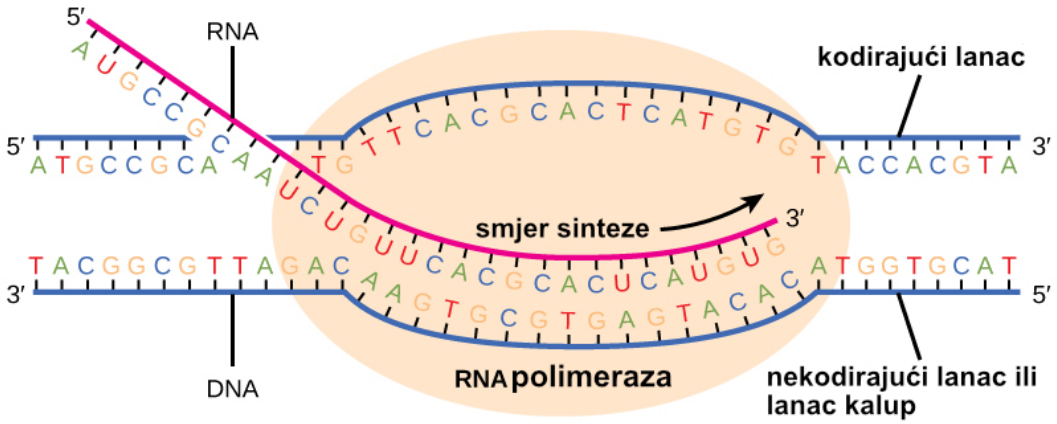
Svrha ovoga rada, izrađenog u okviru interdisciplinarnog istraživačkog projekta *Artificial Intelligence for Gene Expression Prediction* (UNIRI-INOVA-3-23-1), jest sagledati primjene tehnika dubokog učenja nad podacima ovog tipa. Na samom početku rada opisani su osnovni biolo-

ški koncepti koji će pomoći boljem razumijevanju problematike te su sagledani neki od javno dostupnih skupova podataka. Kao praktični dio rada implementirani su modeli višeslojnog perceptrona (engl. *multilayer perceptron* - MLP) i propusne povratne ćelije(engl. *gated recurrent unit* - GRU) za rješavanje problema binarne klasifikacije prisustva raka dojke na temelju izmjerenih razina genske ekspresije pojedinog uzorka. Nadalje, razmotrena je mogućnost primjene polunadziranog strojnog učenja nad ovakvim podacima na način da je implementiran model koji je polučio obećavajuće rezultate u području računalnog vida i klasifikacije slika. Ova će metoda usrednjenog učitelja (engl.*Mean teacher*, MT) [1] također biti objašnjena u daljnjim poglavljima ovoga rada. Navedeni će modeli biti analizirani i vrednovani korištenjem standardnih metrika strojnog učenja.

2. Genom i genska ekspresija

Kada je riječ o primjeni strojnog i dubokog učenja na određeni problem, prije samog modeliranja i treniranja modela nužno je poznavanje domene problema. U ovom će se poglavlju čitatelj imati priliku upoznati upravo s domenom problema te će mu biti objašnjeni s biološke perspektive ključni pojmovi korišteni kroz ovaj rad. Ljudski organizam u sebi sadrži kodiranu informaciju koju nasljeđuje od roditelja, a ona je zapisana u jedinicama: genima. Čitava genetska informacija jednog organizma jest genom, a svaki je gen sekvencija deoksiribonukleinske kiseline ili skraćeno DNK(engl. *deoxyribonucleic acid* - DNA), u kojoj se nalazi uputa za sintezu proteina. DNK je dvolančana molekula čiji je svaki lanac građen od fosfatne skupine, deoksiriboze i dušične baze (adenin, gvanin, citozin, timin). Upravo je redoslijed dušičnih baza ključan za nasljednu uputu. Također valja napomenuti kako su adenin i gvanin purinske, a citozin i timin piramidinske dušične baze. Proces u kojem gen vrši sintezu nekog produkta značajnog za funkcioniranje organizma poznat je kao genska ekspresija. Pritom su neki od mogućih produkata proteini, mRNK, tRNK(engl. *Transfer ribonucleic acid*) itd. Genska ekspresija se sastoji od dvije faze: transkripcije i translacije. Kada je riječ o genskoj ekspresiji, neophodno je spomenuti i, uz DNK i proteine, treću biološku makromolekulu nužnu za život upravo zbog svoje uloge u ovom procesu i uloge kod regulacije gena. Riječ je o ribonukleinskoj kiselini koja predstavlja sponu između informacije zapisane u DNK i sinteze proteina. Obzirom na svoju ulogu kod prijenosa informacije mogu se razlikovati tri vrste RNK: glasnička (mRNK), ribosomska (rRNK) i transportna(tRNK). Prva je zadužena za prijenos genske informacije iz deoksiribonukleinske kiseline do citoplazme. Sinteza mRNK na jednom nekodirajućem lancu(onaj s kojeg se prepisuje poruka) prva je od spomenute dvije faze: transkripcija. Novonastala sekvencija suprotna je nekodirajućem lancu na kojem je nastala, a jednaka kodirajućem lancu DNK uz iznimku da se na mjestima timina u RNK nalazi uracil. Navedeno je

vidljivo na slici 2.1.



Slika 2.1 *Proces transkripcije: RNK polimeraza sintetizira RNK lanac na nekodirajućem DNK lancu, pri čemu je RNK komplementarna nekodirajućem, a identična kodirajućem DNK lancu (uz iznimku dušične baze Uracil(U) umjesto Timina(T)). Preostale baze nukleotida označene slovima duž lanca jesu Adenin(A), Gvanin(G) i Citozin(C). Slika preuzeta iz [35]*

Transkripcija ne bi bila moguća bez enzima RNK polimeraza čija je uloga prepoznati početnu sekvenciju gena(promotorska sekvencija). U glasničkoj RNK tri susjedna nukleotida su grupirana zajedno. Ovakav slijed naziva se kodon, a on je bitan jer kodira aminokiselinu. Također, genska šifra ima svojstvo degenerativnosti: većinu je aminokiselina moguće dobiti s više kodona. Nakon transkripcije slijedi translacija, a u ovoj fazi sudjeluju transportna i ribosomska RNK i proteini. Izvan jezgre proteini tumače sekvencu mRNK te prikupljaju odgovarajuće dostupne aminokiseline vežući ih u lanac. Za svaki triplet molekule mRNK biva vezana komplementarna molekula tRNK(antikodon) na čiji je drugi jednolančani kraj vezana odgovarajuća aminokiselina. Faza translacije odvija se na ribosomima, a ishod ove faze jest stvaranje proteina na istima.

Unatoč činjenici da je proces sinteze proteina od presudne važnosti kod nasljeđivanja i održavanja životnih funkcija, isto tako treba napomenuti kako je samo 1% ljudske DNK sačinjen od gena koji kodiraju proteine. Ostatak čini nekodirajuća DNK za koji su znanstvenici dugo vremena smatrali kako nije ni od kakvog značaja za funkcioniranje ljudskog organizma. Iako dobar dio nekodirajuće DNK i dalje ostaje misterij, postalo je jasno da to nije slučaj, već da nekodirajuća DNK ima značajnu ulogu u kontroli genske aktivnosti. Primjerice, ona sadrži sekvence koje se ponašaju kao regulatorni elementi te kontroliraju u kojem trenutku i gdje

geni postaju ili prestaju biti aktivni. U regulatore spadaju promotori, pojačivači, izolatori i utišavači. Oni omogućuju vezivanje transkripcijskih faktora koji su u suštini specijalizirani proteini koji kontroliraju način na koji se informacija iz gena pretvara u proteine. Promotori se obično nalaze na početku gena te služe kao veznivna mjesta za RNK polimerazu i omogućuju početak transkripcije. Pojačivači mogu biti smješteni daleko od gena koji reguliraju, a funkcija im je povećanje stope transkripcije gena, dok utišivači imaju oprečnu ulogu. Izolatori su sekvence DNK koje sprječavaju interakciju između gena i njihovih regulatora te osiguravaju da regulacijski signali utječu samo na ciljane gene.

Kada je riječ o primjeni umjetne inteligencije u ovom području, njome se značajno olakšava analiza genskih podataka te su omogućene nove spoznaje, posebice u razumijevanju mehanizma bolesti. Jedna od spoznaja do koje modeli strojnog učenja mogu dovesti jest identifikacija biomarkera. Identificiranje gena ili obrazaca ekspresije često je ključno za dijagnostiku i liječenje. Također, u posljednje vrijeme primjetan je ubrzani razvoj personalizirane medicine gdje modeli na temelju podataka identificiraju najprikladnije tretmane za liječenje. Strojno učenje je izuzetno korisno i za otkrivanje regulatornih mreža gena. Dobivanjem uvida u to kako se različiti geni koordinirano aktiviraju tijekom bioloških procesa ili u uvjetima bolesti moguće je bolje razumijevanje kompleksnih mehanizama koji utječu na stanične procese kao što su diferencijacija stanica i apoptoza.

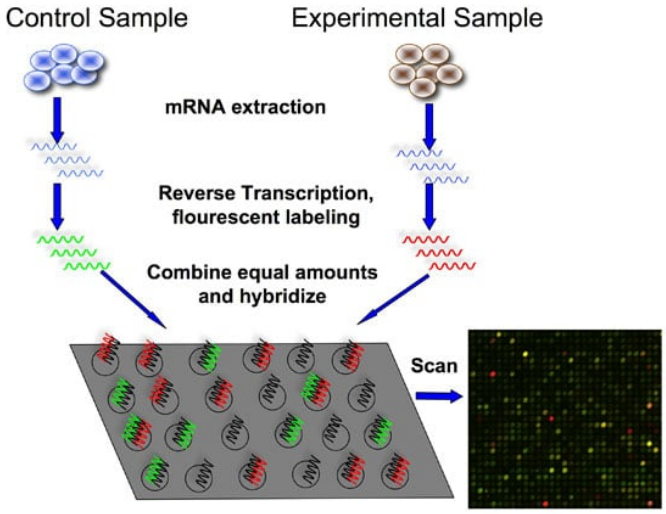
2.1 Metode dobivanja podataka genske ekspresije

U današnje vrijeme razvijene su i učestalo korištene visoko propusne metode za dobivanje podataka za gensku ekspresiju. One omogućuju brzu analizu velikog broja uzoraka istovremeno. Dvije takve koje se najčešće korsite danas objašnjene su u idućem djelu rada.

2.1.1 Hibridizacija

Jačina genske ekspresije u velikoj mjeri određuje karakteristike jedinke. Ona ovisi o broju faktora, između ostalog o međugenskim regulatornim odnosima. Danas postoji više tehnika

za određivanje iste, pri čemu je moguće izmjeriti razinu ekspresije više tisuća gena u jednom mjerenju. Često su korištene metode molekularne biologije koje vrše sekvencioniranje mRNK ili kojima se mRNK poslije pretvorbe u cDNK(engl. *complementary DNA*) hibridizira na sitne pločice koje sadrže veliki broj molekula cDNK specifičnih za gene koji kodiraju proteine(DNK čip). Ustaljeni termin za navedeno mnoštvo jednolančanih molekula jesu probe(engl. *probes*). Probe su pričvršćene za podlogu u određenim točkama. Svaka od ovih točaka predstvalja pojedini gen. Molekule mRNK iz promatrane stanice se izoliraju nakon čega slijedi postupak RT-PCR(engl. *reverse transcription polymerase chain reaction*). Ovaj je postupak suprotan transkripciji te se njime kreira komplementarni lanac DNK molekule. Zatim se pomoću enzima DNK polimeraza vrši sinteza drugog lanca pripadajuće DNK molekule i takva se DNK zove cDNK. Svaka je ova molekula označena fluorescentnom bojom i hibridiziraju se s dostupnom komplementarnom sekvencom. Nadalje, s obzirom na boju pojedinih točaka može se odrediti koji su geni aktivni u određenoj stanici. Intenzitet fluorescentnog signala u svakoj točki odražava količinu mRNK prisutnu za taj gen, što daje uvid u razinu ekspresije gena. Analizom ovih podataka može se razumjeti koji su geni uključeni u specifične biološke procese ili reakcije na različite uvjete, što je ključno za istraživanje funkcionalne genomike i molekularne biologije. Glavna prednost ove metode jest moguće paralelno odvijanje višestrukih analiza pomoću kojih se može analizirati čitavi genom. Slika 2.2 ilustrira navedenu tehniku.



Slika 2.2 *Prikaz tipičnog eksperimenta DNK čipa komparativne hibridizacije, preuzeto iz [33]*

2.1.2 RNK-Seq

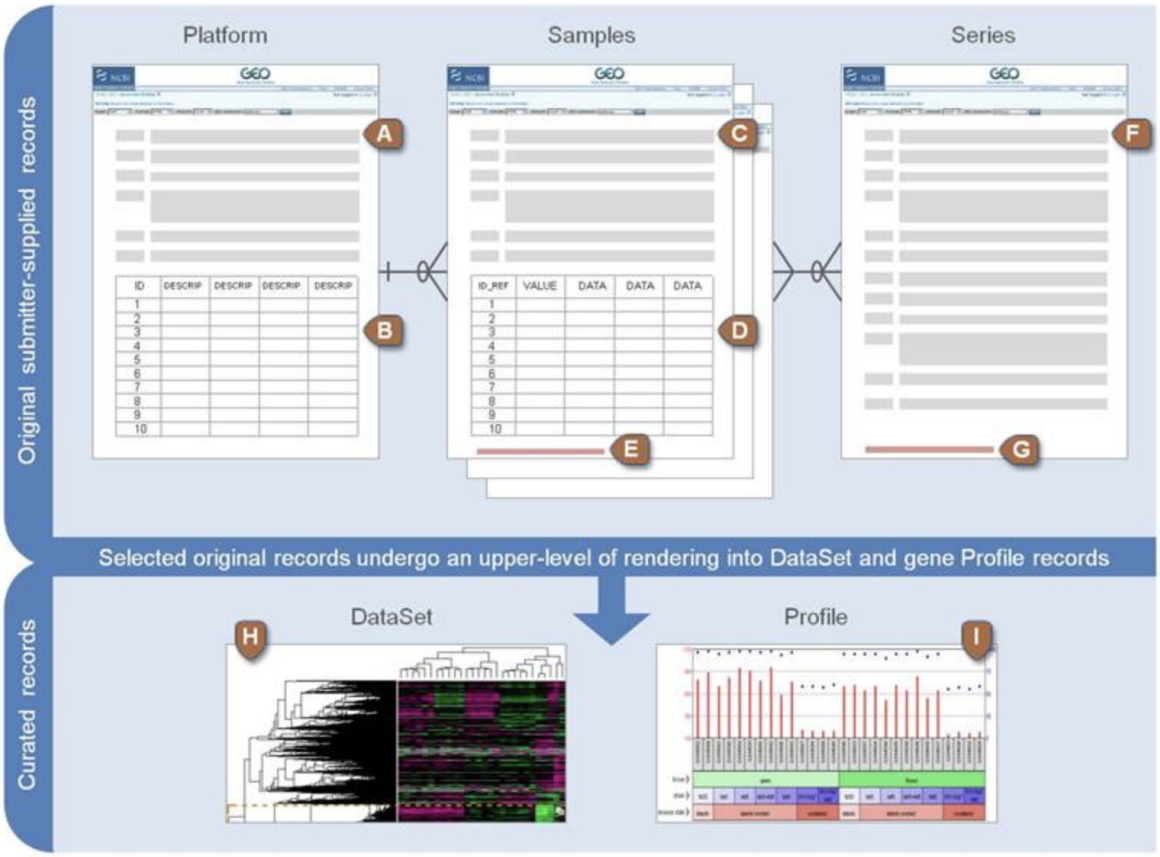
Još jedna često korištena metoda za dobivanje podataka genske ekspresije posljednjih godina je RNK-Seq[2]. Ova tehnika koristi visoko propusno sekvencioniranje (engl. *high-throughput sequencing*) ili sekvencioniranje iduće generacije (engl. *next generation sequencing*) za profiliranje čitavog transkriptoma (skup svih RNK molekula). Pristupi temeljeni na hibridizaciji nisu uspjeli u potpunosti sistematizirati i odrediti razine različitih RNK molekula u genomima. [3] Pomoću ove metode moguće je steći uvid u količinu transkripata pojedinih stanica tijekom specifičnog stanja ili određene razvojne faze što ima važnu ulogu u razumijevanju bolesti. Za razliku od metode opisane u prethodnom poglavlju, ova metoda izravno utvrđuje cDNK sekvencu. Početni korak jest izolacija RNK nakon čega slijedi utvrđivanje cDNK i amplifikacija. Nakon pripreme transkripta, slijedi njihovo sekvencioniranje koje se može vršiti u jednom ili oba smjera. Dobiveni rezultati potom bivaju procesuirani metodama koje ovise o konkretnoj studiji koja se provodi i njezinim ciljevima. Glavne prednosti metode RNK-Seq jesu bolje otkrivanje niske razine genske ekspresije te opsežnija i raznolika analiza transkriptoma. Nadalje, ova je metoda od svoje pojave omogućila otkrivanje novih transkripta. Kao glavni nedostatak metode može se izdvojiti visina troška. Također, u odnosu na ostale metode zahtjeva više resursa te su skupovi podataka kompleksniji, a generirane vrijednosti zahtjevaju visoku razinu stručnosti za ispravnu interpretaciju.

2.2 Podaci o genskoj ekspresiji

Uspjeh modela strojnog učenja uvelike je ovisan o kvaliteti i količini podataka. Kod bioloških baza podataka ustaljena je podjela na dvije kategorije: primarne i sekundarne. Primarne su sačinjene od podataka dobivenih eksperimentom poput primjerice nukleotidnih sekvenci ili proteinskih sekvenci, dok se sekundarne sastoje od podataka dobivenih analizom primarnih.[7] U ovom će poglavlju fokus biti stavljen na javno dostupne primarne skupove podataka za projekte ove domene.

2.2.1 GEO

Gene Expression Omnibus (GEO) najveći je javno dostupni resurs podataka genske ekspresije[10]. Do 2020. godine je sadržavao 27856 skupova podataka ove vrste[26]. Osnovan je 2000. godine te sadrži visoko propusne podatke genske ekspresije poput mikročip i RNAseq pogodne za istraživanja. Korisnicima je omogućeno da rezultate svojih eksperimenata objave na toj platofmi. Za bazu je zaslužan NCBI (*National Center for Biotechnology Information*), a karakterizira ju usklađenost s MIAME (*Minimum Information About a Microarray Experiment*) standardom. Isti propisuje minimalnu količinu informacija koju je potrebno dostaviti sa svrhom pružanja kompletne slike o provedenom eksperimentu. Način na koji su podaci u bazi organizirani vidljiv je na slici 2.3.



Slika 2.3 *Organizacija podataka u GEO repozitoriju, preuzeto iz [5]*

Platforma (engl. *Platform*) sadrži detalje o konkretnom načinu provedbe eksperimenta kojim su podaci pribavljeni(npr. opis *array-a*). Uzorak (engl. *Sample*) sadrži informacije o uvjetima i načinu manipuliranja individualnog uzorka. Pojedini uzorak se mora referencirati na samo jednu platformu, a može biti uključen u više serija. Serija (engl. *Series*) predstavlja skup grupiranih uzoraka te je upravo ona najčešće od interesa istraživačima. Svaka platforma, uzorak i serija imaju svoj jedinstveni identifikator (oblika redom GPLxxx, GSMxxx, GSExxx). Iako je otprilike 90% podataka vezano uz gensku ekspresiju, postoje i podaci za druge vrste studija (npr. varijacije genoma, vezivanja genoma...).

2.2.2 TCGA

The Cancer Genome Atlas još je jedan poznati izvor podataka za bioinformatičare. Sadrži više od 20000 uzoraka 33 različite vrste raka i njima pripadajuće zdrave uzorke[4]. Projekt je pokrenut 2006. godine kao zajednički program NCI-a (*National cancer institute*) i NHGRI-a (*National Human Genome Research Institute*). Podaci dolaze u raznim oblicima (genomski, trankriptomski, proteomski itd.) te su rezultirali svakakvim uspjesima poput revolucioniariziranja načina klasificiranja raka i identifikacije novih biomarkera čime je omogućen značjan napredak personalizirane medicine u onkologiji. TCGA se ističe po svojoj rigoroznoj metodologiji i standardiziranim protokolima za prikupljanje, obradu i analizu uzoraka. Projekt je uspostavio stroge kriterije kontrole kvalitete kako bi osigurao visoku pouzdanost podataka. Postoje 4 načina za preuzimanje TCGA podataka: Matrica podataka, *Bulk* preuzimanje, Pristup HTTP direktoriju i Pretraga datoteke. Tok koji rezultira da podaci postanu dostupni javnosti jest sljedeći:

- Uzorci tumora i normalnih tkiva te klinički podaci pacijenata oboljelih od raka prikupljaju se i šalju u *Biospecimen Core Resource* (BCR).
- BCR šalje podatke u *Data Coordinating Center* (DCC). DNK/RNK bivaju izuzeti iz uzoraka i označeni barkodom poslani u *Genome Characterization Center* (GCC) i *Genome Sequencing Centers* (GSC)
- GCC analizira genomske promjene uključene u razvoj raka te šalje eksperimentalne re-

zultate u DCC.

- GSC provodi temeljito sekvenciranje DNK i šalje rezultate u *Cancer Genomics Hub* (CGHub)
- DCC potvrđuje podatke i standardizira njihovu formu i podaci bivaju dostupni zajednici

2.2.3 1000 genomes

Iako po pitanju podataka genske ekspresije nije mjerljiv s prethodna dva navedena izvora, vrijede spomena je i jedan od najpoznatijih javno dostupnih skupa podataka kada je u pitanju ljudski genom - *1000 genomes*. Ovaj je projekt pokrenut 2008. godine u doba kad je napredak u tehnologiji sekvenciranja značajno umanjio troškove istog što je rezultiralo prvim projektom koji je za cilj imao sekvencioniranje genoma velikog broja ljudi kako bi se dobio uvid u ljudsku genetičku varijabilnost. Projekt je u konačnici rezultirao bazom koja obuhvaća većinu genetičkih varijacija u proučavanoj populaciji (95% potpuna za varijacije s frekvencijom alela od 1%, a za učestalije ovaj postotak raste)[6]. Nastavno ovom projektu osnovan je IGSR (engl. *International Genome Sample Resource*) čija je zadaća održavanje i proširenje skupa podataka navedenog projekta te uvođenje novih skupova.

3. Strojno učenje

Svrha ovog poglavlja jest objasniti koncepte strojnog učenja koji su potrebni za temeljito razumijevanje ovoga rada. Strojno učenje jest grana umjetne inteligencije koja omogućuje računalima da uče na način da optimiziraju neki kriterij uspješnosti temeljem dostupnih podataka ili prethodnog iskustva. Po definiciji Mitchella[8]: "Program uči iz iskustva E s obzirom na klasu zadataka T i mjeru performanse P ako se performanse u zadacima unutar skupa T, mjerene po P, poboljšaju s iskustvom E". Proces strojnog učenja se u pravilu odvija kroz sljedeće korake:

1. Priprema podataka

   Učitavanje podataka

   Ekstrakcija značajki

   Redukcija dimenzionalnosti

2. Odabir modela

3. Učenje modela

4. Vrednovanje modela

5. Ispravljanje modela (engl. *debugging*)

6. Instalacija modela (engl. *deployment*)

Strojno učenje svoju primjenu danas nalazi u raznim područjima ljudskog djelovanja: od znanosti, medicine, financijske industrije pa do primjena kojima je za cilj olakšati svakodnevni

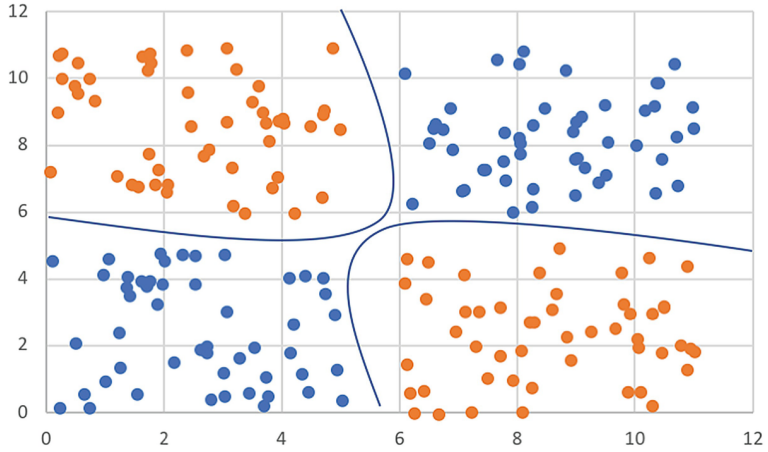
život ili automatizirati poslovne procese (obrada prirodnog jezika, prepoznavanje slika, prilagođena preporuka proizvoda i sl.). S obzirom na označenost podataka strojno učenje se dijeli na nadzirano, nenadzirano i polunadzirano.

3.1 Nadzirano učenje

Kod nadziranog su učenja ulazne i izlazne varijable definirane, a algoritam pokušava naučiti kako mapirati ulaz u izlaz. Ulazni podaci posjeduju određeni broj značajki (nezavisne varijable) te ciljne (zavisne) varijable. Svaki primjer ima n značajki te može biti predstavljen vektorom značajki $\mathbf{x} = (x_1,x_2...,x_n)$. Ukoliko je zavisna varijabla, tj. oznaka $\mathbf{y}$ diskretna vrijednost radi se o klasifikaciji, a ukoliko je kontinuirana riječ je o regresiji. Neki od poznatih primjera algoritama nadziranog učenja su Linearna regresija (LR), K-najbližih susjeda (engl. *K nearest neighbors*, KNN), Metoda potpornih vektora (engl.*Support vector machine*, SVM) i Algoritam slučajnih šuma(engl. *Random Forests*).

3.2 Klasifikacija

U okviru ovoga rada izazov je razviti modele koji će biti uspješni pri obavljanju zadatka klasifikacije. Drugim riječima, cilj je na temelju dostupnih podataka naučiti model koji će s obzirom na ulazni vektor značajki svrstati odgovarajući primjer u jednu od mogućih klasa. S obzirom da je broj mogućih klasa u ovom radu 2, riječ je o binarnoj klasifikaciji. Na slici 3.1 prikazan je primjer binarne klasifikacije. Ulazni prostor dijeli se na više regija (u slučaju binarne klasifikacije dvije) te u ovisnosti o regiji kojoj pripada pojedini primjer svrstan je u odgovarajuću kategoriju.



Slika 3.1 *Grafički prikaz binarne klasifikacije(označene plavim i narančastim točkama) i granice odluke(označena plavom linijom), preuzeto iz [36]*

Svaki se algoritam strojnog učenja može razmotriti kroz 3 različite komponente: model, funkcija gubitka i optimizacijski postupak. Model je skup hipoteza[1] h parametriziranih s θ:

$$\mathcal{H} = \{h(x;\theta)\}$$

Tako će primjerice model u slučaju jednostavnog problema linearne regresije biti skup svih mogućih pravaca (za neke druge probleme skup svih ravnina i sl.). Kako bi bilo moguće odrediti optimalnu hipotezu koja će najmanje griješiti prilikom klasifikacije (ili regresije) potrebno je ustanoviti način procjene koliko je hipoteza zadovoljavajuća. Iz tog se razloga definiraju empirijska pogreška i funkcija gubitka. Za klasifikaciju je empirijsku pogrešku moguće definirati na sljedeći način: $E(\hat{h}|D) = \frac{1}{N}\sum_{i=1}^{N} 1(h(x^{(i)}) \neq y^{(i)})$. Ovako definirana empirijska pogreška zapravo pruža informaciju o postotku pogrešno klasificiranih primjera. Matematičko očekivanje funkcije pogreške jest funkcija gubitka (engl. *loss function*) L. Općenito vrijedi:

$$E(h|D) = \frac{1}{N}\sum_{i=1}^{N} L(y^{(i)}, h(x^{(i)};\theta))$$

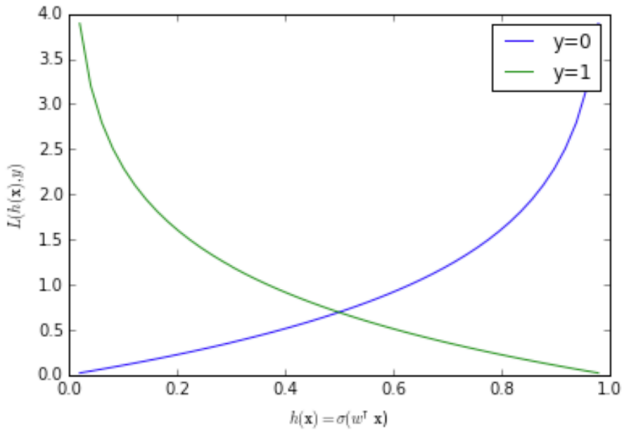
U strojnom učenju ne postoji univerzalna funkcija gubitka, već njezin odabir ovisi o raznim faktorima. Neki od njih su odabrani algoritam, lakoća derivabilnosti i u manjoj mjeri postotak

[1] funkcija koja za cilj ima naučiti preslikavanje iz X u Y

vrijednosti koje značajno odskaču unutar skupa podataka[23]. Najčešće korištena funkcija gubitka kod klasifikacijskih problema jest gubitak unakrsne entropije (engl. *cross entropy loss*, slika 3.2):

$$L(y, h(\mathbf{x})) = -y * lnh(\mathbf{x}) - (1 - y) * ln(1 - h(\mathbf{x}))$$

Valja primjetiti kako ukoliko je oznaka primjera $\mathbf{y} = 1$ drugi će pribrojnik biti jednak nuli pa je gubitak posljedično -ln(h($\mathbf{x}$)) što je 0 ako vrijedi h($\mathbf{x}$) = 1 ili teži k ∞ ako vrijedi h($\mathbf{x}$) = 0. Bitno je napomenuti kako ova funkcija gubitka izrazito kažnjava pogrešna predviđanja donesena s velikom sigurnošću. Slična logika vrijedi i za slučaj kad je $\mathbf{y} = 0$. PyTorch i slične knjižnice omogućuju jednostavno računanje velikog broja definiranih funkcija gubitaka uz jednu liniju koda. Za potrebe klasifikacije su za izdvojiti klase *CrossEntropyLoss* i *BCELoss*. Prva je prikladnija za višeklasnu klasifikaciju, dok se druga koristi za binarnu i daje jedan izlaz u obliku vjerojatnosti(*CrossEntropyLoss* pak određuje vjerojatnosti preko softmax funkcije).



Slika 3.2 *Grafički prikaz funkcije gubitka unakrsne entropije u ovisnosti o predviđanju(zelenom bojom označene pozitivne, a plavom negativne instance), preuzeto iz [34]*

Funkcija gubitka temelj je za posljednju komponentu nužnu za proces učenja: optimizacijski postupak. Njime se nastoje pronaći parametri koji će minimizirati pogrešku klasifikatora(ili općenito modela).

$$\theta^* = \arg\min_{\theta} E(\theta|D)$$

Najpoznatiji odabir u strojnom učenju kada je u pitanju optimizacijski postupak je gradijentni spust. On funkcionira postepenim spuštanjem od početne točke u smjeru suprotnom od

gradijenta u toj točki sve dok se ne dosegne minimum(kad gradijent iznosi 0).

$$\mathbf{w} \leftarrow \mathbf{w} - \eta \nabla E(\mathbf{w}|\mathcal{D})$$

Koeficijent η je pritom konstanta koja određuje brzinu učenja, odnosno veličinu učinjenog koraka(stopa učenja). U ovome radu korišten je Adam optimizator. *Adaptive moment estimation* je proširenje stohastičkog gradijentnog spusta(SGD) i često je korišten u dubokom učenju. Za razliku od SGD-a stopa učenja se mijenja tijekom treninga. Autori ovog algoritma navode kako on objedinjuje prednosti algoritama AdaGrad(stopa učenja po pojedinom parametru) i RMSProp(ažuriranje stope učenja po parametru s obzirom na prosjek nedavnih magnituda gradijenata)[24]. Za razliku od RMSProp-a koji gleda prosjek, Adam je baziran na adaptivnim izračunima momenata[2] nižih redova, tj. koristi prosjek drugog momenta gradijenata(necentriranu varijancu). Njegove prednosti su brzina izvođenja, prilagodljivost i jednostavnost implementacije.

3.3 Polunadzirano učenje

Polunadzirano učenje koristi i označene i neoznačene podatke te je nastavno tome svojevrsna mješavina između nadziranog i nenadziranog učenja. Nad neoznačenim se podacima vrše predviđanja nakon čega se takvi podaci koriste za predviđanje novih podataka nadziranim učenjem. Ovaj je pristup izuzetno bitan u strojnom učenju. U stvarnom svijetu većina problema se može efektivno riješiti na ovaj način iz razloga što je skupljanje označenih podataka često neefikasno i vremenski zahtjevno. Nadalje, ideja polunadziranog učenja najsličnija je metodi učenja kod ljudi gdje je situacija takva da ljudska i životinjske vrste uče svijet promatrajući ga (velika većina učenja je nenadzirana).

[2]sredstvo ubrzavanja optimizacije smanjivanjem oscilacija u potrazi za globalnim minimumom. Metoda je temeljena na eksponencijalnom težinskom prosjeku. Kod korištenja momenta u praksi eksperimentira se odabirom postotka, a kao česta početna vrijednost se uzima 0.9

3.3.1 Metoda usrednjenog učitelja

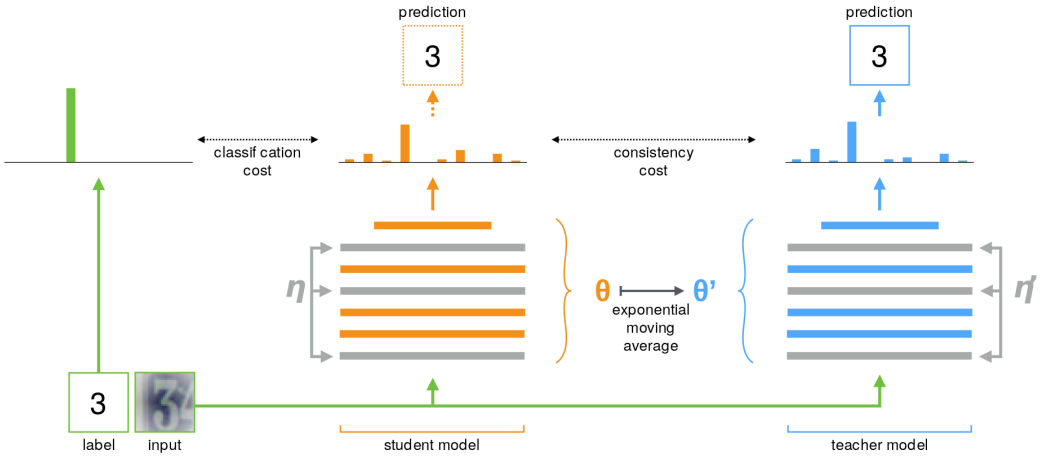
S obzirom na ogromni broj parametara koji modeli mogu imati, u strojnom učenju lako dolazi do problema prenaučenosti (engl. *overfitting*). Kako bi se poboljšala prediktivna sposobnost modela na neviđenim podacima, tj. sprječila loša generalizacija, koriste se metode regularizacije. Primjer takve bi bio dodavanje šuma (engl. *noise*) ulazu modela. Pritom je cilj da, ukoliko se primjeri ne razlikuju znatno, bivaju isto klasificirani. Međutim, za neoznačene podatke pogreška klasifikacije nije definirana što je veliki izazov u polunadziranom učenju. Metoda usrednjenog učitelja u suštini predstavlja odličan pokušaj rješavanja problema efikasnog korištenja neoznačenih podataka u polunadziranom učenju čime se smanjuje stupanj prenaučenosti. Ova je metoda svoju inspiraciju našla u *state-of-the-art* Temporal Ensembling metodi koja računa eksponencijalni težinski prosjek predviđanja svakog pojedinog primjera i kažnjava ona koja su nekonzistentna[22]. Promjenu pristupa koju autori MT metode predlažu jest da, umjesto praćenja eksponencijalnog težinskog prosjeka predviđenih oznaka pojedinih primjera se računa EMA težina modela. Na ovaj se način ciljana oznaka (engl. *target*) neće mijenjati samo jednom po epohi. U metodi su prisutna dva modela: student i učitelj (engl. *teacher*). Student i učitelj su inicijalizirani kao isti model. Student je onaj koji će učiti na označenim podacima, a učitelj je prosjek uzastopnih student modela. Korištenje prosjeka težina tijekom vremena u pravilu polučuje bolje rezultate nego korištenje završnih težina. Kako je cilj da predviđanja ova dva modela budu konzistentna, klasifikacijskoj se pogrešci student modela zbraja tzv. *consistency cost*. On je ovisan o udaljenosti predviđanja student modela (s parametrima θ i šumom η) s učiteljevim(s parametrima θ' i šumom η'):

$$J(\theta) = E_{x,\eta',\eta}\left[\|f(x,\theta',\eta') - f(x,\theta,\eta)\|^2\right]$$

Parametri modela učitelja u vremenskom trenutku t definirani su na sljedeći način:

$$\theta'_t = \alpha\theta'_{t-1} + (1-\alpha)\theta_t$$

Hiperparametar α određuje koji postotak modela učitelja će ostati nepromijenjen. Pri eksperimentiranju s ovom vrijednosti kao dobra početna točka se navodi 0.999. Na slici 3.3 nalazi se vizualni prikaz ove metode na trening skupu (engl. *batch*) s jednim označenim primjerom.



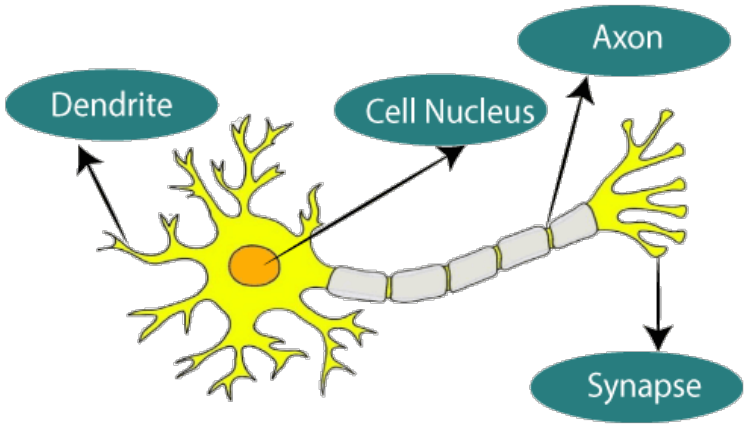
Slika 3.3 *Metoda usrednjenog učitelja, preuzeto iz [1]*

Kako je navedeno u radu, izlazi oba se modela mogu koristiti za predviđanja, ali kako trening odmiče učitelj se sve više stabilizira i izglednije je da će u konačnici njegova predviđanja biti točnija. Valja naglasiti važnost šuma u ovom modelu. Naime, zbog načina definiranja gubitka konzistentnosti, lako je moguće naići na izazove prilikom treniranja modela. Zbog prirode stohastičkog gradijentnog spusta model može zapasti u trivijalno rješenje i konvergirati u vrlo jednostavno rješenje koje neće biti od koristi. Glede dodavanja šuma, u eksperimentu(koji spada u domenu računalnog vida) autora uvode se tri vrste šuma: nasumične translacije i rotacije ulaznih slika, Gaussov šum u ulaznom sloju te *dropout*. Na Github repozitoriju autori ohrabruju korištenje i drugih vrsta šumova koje mogu biti relevantne za dani problem. Za *consistency cost* funkciju može se koristiti srednja kvadratna pogreška (engl. *Mean Squared Error*) ili *KL-Divergence*. U sklopu ovog diplomskog rada korištena je prva s obzirom da za manje skupove podataka uglavnom polučuje bolje rezultate.

3.4 Duboke neuronske mreže

Koncept dubokih neuronskih mreža predstavlja užu granu strojnog učenja koja je zaslužna za veliku većinu značajnih napredaka i spoznaja u području umjetne inteligencije. Osnovna ideja je svoju inspiraciju našla u interakcijama ljudskog živčanog sustava. Iako je ljudski model

znatno kompleksniji, u dubokim neuronskim mrežama su vidljivi određeni mehanizmi istog. Pojednostavljeni model neuronske mreže čovjeka prikazan je na slici 3.4. Dendriti prenose ulazne signale, a aksoni su zaduženi za prijenos informacija iz jedne u drugu stanicu putem sinapsi.

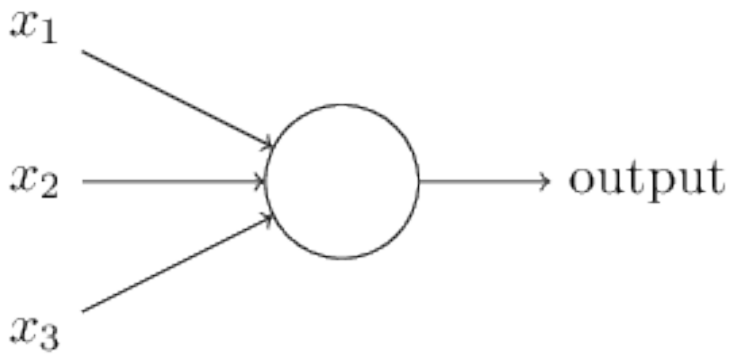


Slika 3.4 *Biološka neuronska mreža: dendriti primaju informacije od neurona, aksonom prolaze električni impulsi te završava sinapsom koja ih prenosi do idućeg neurona , preuzeto iz [9]*

Duboke neuronske mreže se sastoje od više skrivenih slojeva što rezultira modelom koji je sposoban rješavati kompleksnije zadatke. Postoji više vrsta skrivenih slojeva ovisno o konkretnom problemu (npr. konvolucijski sloj, normalizacijski, sloj pažnje...). Proces učenja ovisan je naravno o ulaznim podacima te se odvija kontinuiranim promjenama težina sinapsi.

3.4.1 Višeslojni perceptron

50-tih godina prošlog stoljeća Frank Rosenblatt predstavlja koncept perceptrona inspiriran ranijim radom Warrena McCullocha i Waltera Pittsa[27]. Perceptron je rani pokušaj implementacije umjetnog neurona koji se danas u praksi ne koristi često. Njegov primjer je dan na slici 3.5.

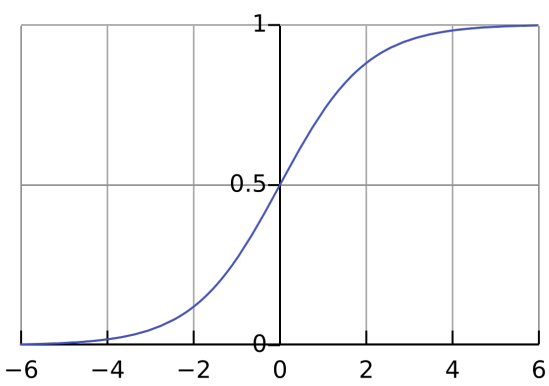


Slika 3.5 *Prvotni model perceptrona, preuzeto iz [27]*

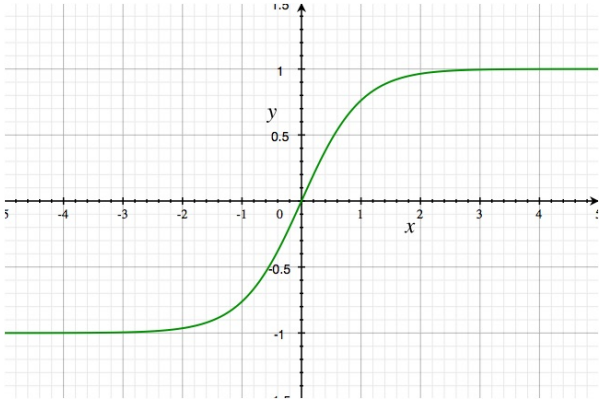
Kao što je vidljivo iz slike, ovaj model prima određeni broj binarnih ulaza x_1,x_2**...**, a kao izlaz vraća 0 ili 1. U konkretnom primjeru sa slike prisutna su 3 ulaza i svakom je pridružena određena težina(w_1,w_2,w_3) i određeni pomak (engl.*bias*)(b_1,b_2,b_3). Pravilo određivanja izlaza dano je sljedećom formulom:

$$f(x) = \begin{cases} 0, & \text{ako } w * x + b \leq 0 \\ 1, & \text{inače} \end{cases}$$

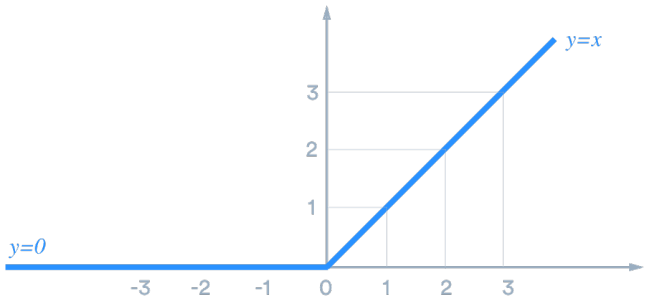
Ovdje su $\boldsymbol{w}$ i $\boldsymbol{x}$ vektori čije komponente su težine i ulazi, a pomak($\boldsymbol{b}$) je vektor koji na neki način pokazuje koliko je "lako" da izlaz perceptrona bude 1. Ovako definirani model perceptrona pogodan je za proces donošenja odluka. Primjerice, svaki neuron može predstavljati određeni faktor pri donošenju odluka te mu se u ovisnosti koliko je on bitan, pridodijeli određena težina. Što je odluka kompleksnija, to se više neurona nalazi u mreži. Međutim, za proces učenja je poželjno da mala promjena težine rezultira malom promjenom izlaza. S ovako definiranim modelom to nije slučaj. Nadalje, većina podataka u stvarnome svijetu je nelinearna, a ovaj model ne omogućava učenje kompleksnih nelinearnih uzoraka. Jedno od rješenja ovih problema jest sigmoidalni neuron. Ulazi i izlaz više neće biti 0 ili 1, već mogu poprimit bilo koju vrijednost unutar tog raspona. Izlaz $\boldsymbol{w}$ * $\boldsymbol{x}$ + $\boldsymbol{b}$ je argument funkcije sigmoide ili neke druge nelinearne funkcije. Uz nju, za uvođenje nelinearnosti najčešće su korištene funkcije tanh i ReLU (engl. *Rectified Linear Unit*). U nastavku slijede njihovi grafički prikazi i jednadžbe.



(a) *Sigmoida*



(b) *tanh*



(c) *ReLU*

Slika 3.6 *Aktivacijske funkcije korištene za uvođenje nelinearnosti u model:* ***a) Sigmoida*** *– komprimira ulazne vrijednosti u raspon [0,1]* ***b) Tanh*** *– skalirana verzija sigmoide, preslikava ulaze u raspon [-1,1],* ***c) ReLU*** *– uvodi nelinearnost na način da daje izlaz 0 za sve negativne vrijednosti, a pozitivne ostavlja istim*

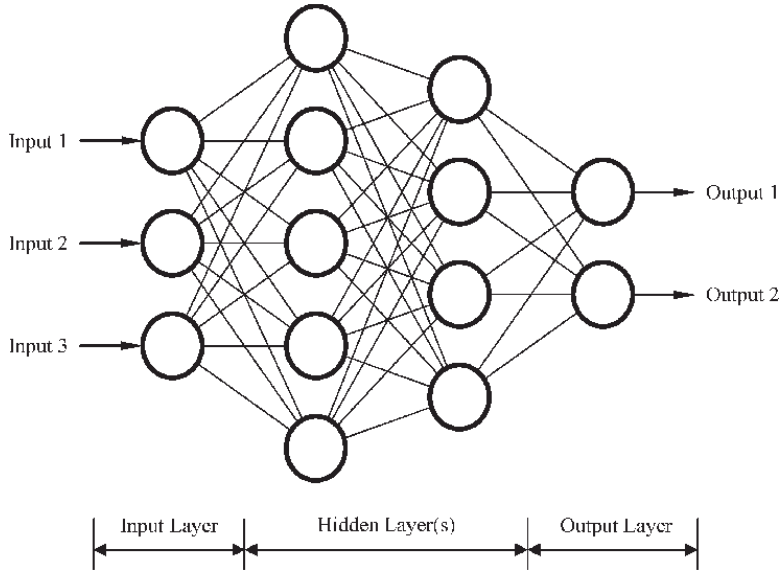
$$\text{sigmoid}(x) = \frac{1}{1 + e^{-x}} \tag{3.1}$$

$$\tanh(x) = \frac{e^{2x} - 1}{e^{2x} + 1} \tag{3.2}$$

$$\text{ReLU}(x) = \max(0, x) \tag{3.3}$$

U dubokom učenju pojam višeslojnog perceptrona obuhvaća sve duboke mreže koje se sastoje od više slojeva u potpunosti povezanih neurona. Broj skrivenih slojeva u pravilu se određuje

eksperimentalno. Što je on veći mrežu je teže trenirati i sklonija je prenaučenosti. Na slici 3.7 nalazi se primjer višeslojnog perceptrona s dva skrivena sloja i dva neurona u izlaznom sloju (pogodan za binarnu klasifikaciju).



Slika 3.7 *Višeslojni perceptron čiju strukturu čine ulazni sloj s 3 neurona, 2 skrivena sloja te izlazni sloj s 2 neurona što ga čini pogodnim za binarnu klasifikaciju, preuzeto iz [9]*

3.4.2 Algoritam unazadne propagacije

Unazadna propagacija (engl. *backpropagation*) jedan je od temeljnih algoritama dubokog učenja. U procesu učenja neuronskih mreža ovaj je algoritam neophodan jer omogućuje efikasno računanje gradijenata funkcije gubitka s obzirom na parametre (težine) mreže, a baziran je na lančanom pravilu derivacije (engl. *chain rule*). Prije same unazadne propagacije kroz mrežu potrebno je spremati aktivacije pojedinih neurona s obzirom da su one potrebne za računanje njegovog gradijenta (engl. *forward pass*). Derivacijom funkcije gubitka s obzirom na pojedini parametar spoznaje se kako malene promjene tog parametra utječu na gubitak. Ista ta promjena parametra utjecat će na izlaz promatranog neurona što će utjecati na ulaz u idući neuron i na taj se način kreira svojevrsni domino efekt kroz ostatak mreže. S matematičke strane gledano, ako je y_j^l izlaz j-tog neurona u l-tom sloju i ako je w_{jk}^l težina veze k-tog ne-

urona iz *l-1*-tog sloja s j-tim neuronom u l-tom sloju, onda se promjena u izlazu neurona može izraziti kao:

$$\Delta y_j^l = \frac{\partial y_j^l}{\partial w_{jk}^l} \Delta w_{jk}^l \tag{3.4}$$

Ovo će utjecati na promjenu izlaza q-tog neurona u idućem sloju te će za nju vrijediti:

$$\Delta y_q^{l+1} = \frac{\partial y_q^{l+1}}{\partial y_j^l} \frac{\partial y_j^l}{\partial w_{jk}^l} \Delta w_{jk}^l \tag{3.5}$$

Ovaj se postupak ponavlja sve do izlaznog sloja gdje je vidljiva ovisnost promjene funkcije gubitka o promjeni parametara. Unazadna propagacija se temelji se na 4 jednadžbe čiji opis slijedi u nastavku[27].

Grešku j-tog neurona u l-tom sloju moguće je definirati kao $\frac{\partial C}{\partial z_j^l}$. Intuitivno, navedeno vrijedi jer ukoliko se ulaz u neuron poveća ili smanji za malu vrijednost Δz_j^l, a ukoliko je navedena parcijalna derivacija velika, funkcija gubitka se može značajnije smanjiti promjenom vrijednosti parametra u smjeru suprotnom od gradijenta. Stoga $\frac{\partial C}{\partial z_j^l}$ predstavlja mjeru pogreške neurona i označava se s δ_j^l. Vektor grešaka u izlaznom sloju definiran je kao

$$\delta_j^l = \nabla_{y_j^l} L \odot f'(z^L) \tag{3.6}$$

što je zapravo Hadamardov produkt gradijenta funkcije gubitka i derivacije aktivacijske funkcije u ovisnosti o z^L(umnožak matrica jednakih dimenzija po elementima). Druga bitna jednadžba izražava isti vektor greške u ovisnosti o sljedećem sloju i može se interpretirati kao propagacija greške unazad:

$$\delta_j^l = (\mathbf{W}^{l+1})^T \delta_q^{l+1} \odot f'(z_j^l) \tag{3.7}$$

Množenjem greške s transponiranom matricom težina $(\mathbf{W}^{l+1})^T$ dobiva se učinak unazadne propagacije. Za δ_q^{l+1} se primjenom Hadamardovog produkta greška provlači nazad kroz aktivacijsku funkciju kako bi se dobila greška δ_j^l. Posljednje dvije jednadžbe izražavaju mjeru promjene funkcije gubitka u ovisnosti o pomacima mreže, odnosno o težinama mreže:

$$\frac{\partial C}{\partial b_j^l} = \delta_j^l \tag{3.8}$$

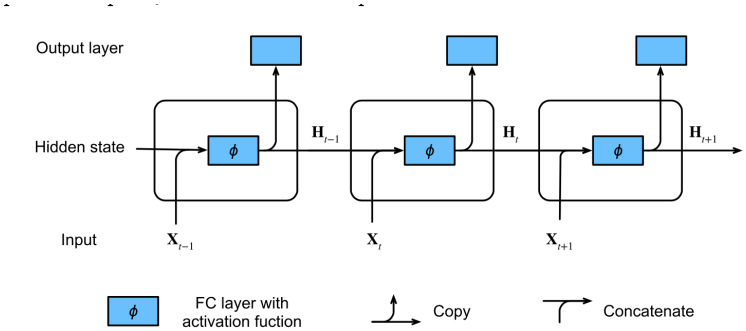
$$\frac{\partial C}{\partial w_{jk}^{l}} = \delta_j^l y_k^{l-1} \tag{3.9}$$

3.4.3 Propusna povratna ćelija

Poseban tip umjetnih neuronskih mreža jesu povratne mreže. Ove mreže uvode varijable stanja za pohranu prošlih informacija te ih koriste u kombinaciji s trenutnim ulazom za određivanje izlaza u datom trenutku. Mnogi modeli strojnog učenja za pretpostavku uzimaju da su podaci neovisni i izvučeni iz iste distribucije. Međutim, to često nije tako, a kao primjer za to može se izdvojiti ovaj tekst gdje su riječi ovisne jedna o drugoj i napisane određenim redoslijedom koji, ako promijenjen, otežava interpretaciju istog. Ono po čemu se povratne mreže ističu je sposobnost obrade sekvencijalnih informacija. One sadrže povratne veze i omogućuju da operacije izvedene u prethodnim koracima imaju utjecaj na trenutnu operaciju. Povratna neuronska mreža se prvi puta spominje u radu [11] i za nju vrijedi:

$$h_t = \tanh(W^h h_{t-1} + W^x x_t + b) \tag{3.10}$$

Pritom je $x_t \in R^{d_x}$ ulazni vektor, $W^x \in R^{d_x \times d_h}$ matrica težina, $W^h \in R^{d_h \times d_h}$ matrica stanja, a $b \in R^{d_h}$ vektor pomaka. Povratna mreža uči stanje h_t za svaki ulazni vektor x_t iz niza $t = 1, 2, \ldots, n$. Svako stanje ovisi o prethodnom h_{t-1} pa se često stanja nazivaju i vremenskim koracima (engl. *time steps*). Osim prethodnog stanja h_{t-1} i ulaznog vektora x_t, prikriveno stanje također ovisi i o matrici stanja W^h. Slika 3.8 ilustrira logiku povratne mreže kroz tri uzastopna vremenska koraka.



Slika 3.8 *Povratna neuronska mreža sa skrivenim stanjem, preuzeto iz [9]*

Nastavno na gornju sliku valja napomenuti kako će stanje $\mathbf{H}_t$ u vremenskom trenutku t biti korišteno za računanje idućeg skrivenog stanja $\mathbf{H}_{t+1}$ te biti ulaz sloju za računanje izlaza $\mathbf{O}_t$ trenutnog stanja t. Glavni nedostatak ovakve arhitekture jest pojava nestajućeg ili eksplodirajućeg gradijenta kad su skupovi podataka dovoljno veliki. Prvi je posljedica množenja mnogo uzastupnih težina manjih od 1, a drugi većih od 1 što rezultira prevelikim odnosno premalim pomacima tijekom stohastičkog gradijentnog spusta čime je onemogućeno efikasno treniranje. Jedan od modela koji nastoji riješiti ovaj problem, a koji je implementiran u sklopu ovoga rada jest model propusne povratne ćelije (engl. *gated recurrent unit - GRU*). Ovaj model odlikuje dobra kontrola memorije, a njegova matematička osnova su sljedeće 4 jednadžbe čije objašnjenje slijedi u nastavku.

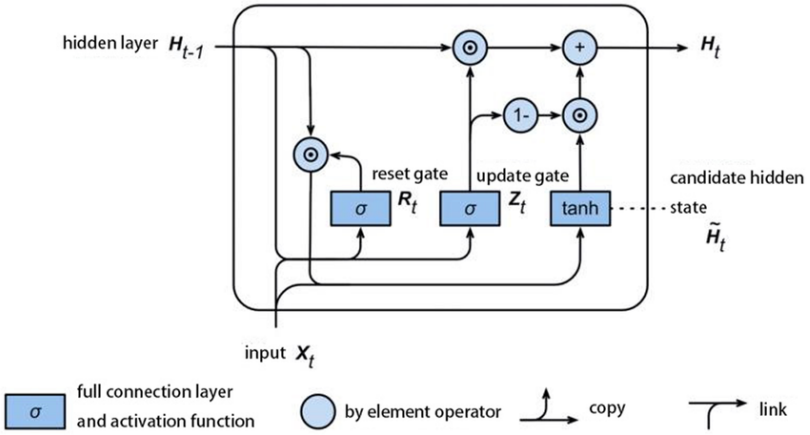
$$\vec{z}_t = \sigma(W_z\vec{x}_t + U_z\vec{h}_{t-1} + \vec{b}_z) \tag{3.11}$$

$$\vec{r}_t = \sigma(W_r\vec{x}_t + U_r\vec{h}_{t-1} + \vec{b}_r) \tag{3.12}$$

$$\tilde{\vec{h}}_t = \tanh(W\vec{x}_t + U(\vec{r}_t \odot \vec{h}_{t-1}) + \vec{b}) \tag{3.13}$$

$$\vec{h}_t = (1 - \vec{z}_t) \odot \vec{h}_{t-1} + \vec{z}_t \odot \tilde{\vec{h}}_t \tag{3.14}$$

Za početak valja objasniti kocept propusnica (engl. *gates*) koji ovaj model uvodi. To su vektori ($g \in R^{d_h}$) koji određuju koliko će se trenutnog stanja ažurirati, a koliko ostati isto. Prva jednadžba definira prvu kontrolnu strukturu GRU-a z_t (engl. *update gate*). Ona određuje omjer u kojem će se kombinirati prethodno stanje (h_{t-1}) i novo kandidatsko stanje. Sigmoida osigurava da su vrijednosti vektora z_t između 0 i 1, gdje vrijednosti bliže 1 znače veći utjecaj novih informacija. Matrice težina W_z i U_z, zajedno s vektorom pomaka b_z, uče se tijekom treninga mreže. Na sličan način funkcionira i druga propusnica r_t kojom je definirano koliko će se prethodnog stanja uzeti u obzir pri računanju novog kandidatskog stanja. Kada je r_t blizu 0, prethodno stanje se gotovo potpuno zanemaruje, omogućujući mreži da zaboravi informacije koje više nisu relevantne. U svakom vremenskom koraku se razmatra mogućnost zamjene vrijednosti memorijske ćelije h_t s vrijednošću $\tilde{h}_t \in R^{d_h}$. Kandidatsko stanje je vidljivo na jednadžbi 3.13, a konačno na 3.14. Vizualni prikaz GRU arhitekture vidljiv je na slici 3.9.



Slika 3.9 *Struktura propusne povratne ćelije, preuzeto iz [9]*

4. Metodologija

4.1 Skupovi podataka

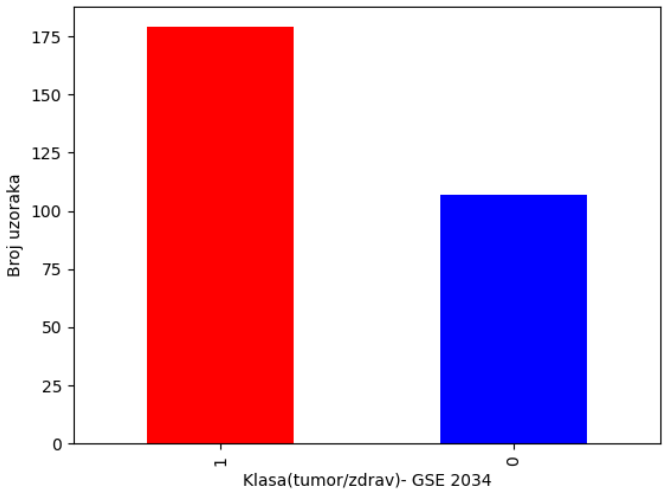
Bolest na kojoj su bazirani skupovi podataka korišteni pri izradi ovoga rada jest rak dojke (engl. *breast cancer*, BRCA), a preuzeti su iz repozitorija Mendeley Data[12]. Rak dojke je odabran jer čini najveći udio svih rakova kod žena. Iako se smtrnost posljednjih godina smanjila, mentalne i fizičke posljedice ove bolesti značajno ugrožavaju zdravlje pogođenih. Još jedan razlog odabira leži u činjenici da je ovaj rak jedan od onih čiji javno dostupni podaci čine velik udio u ukupnoj količini podataka. Ono što je svim skupovima zajedničko i što ovaj problem čini izazovnim jest da ih karakterizira malen broj uzoraka i visoka dimenzionalnost. U nastavku je dan opis GEO serija prisutnih u spomenutom izvoru na kojima su modeli trenirani. Neke vrijednosti koje su nedostajale su bile popunjene prosječnom vrijednošću pojedine grupe. Vrijednosti genske ekspresije svakog gena su bile standardizirane *Z-score-om*[13]. Z score transformacija česta je praksa kada su u pitanju ovakvi podaci. Nakon $\log_{10}$ transformacije Z score se računa oduzimanjem prosječnog intenziteta gena od vrijednosti svakog pojedinog gena i dijeljenja te vrijednosti sa standardnom devijacijom sviju intenziteta. Vrijedi:

$$Z_{score} = (intensity_G - meanIntensity_{G1...Gn})/SD_{G1...Gn}$$

gdje je G bilo koji gen, a n ukupan broj gena[29].

4.1.1 GSE2034

Ova je serija prvi puta javno objavljena 2005. godine nakon eksperimenta znanstvenika Wanga i njegovih suradnika[14]. Sadrži profile genske ekspresije 22283 različitih gena(286 uzoraka). Pacijentima je dijagnosticiran primarni BRCA koji nije zahvatio limfne čvorove. Praćeni su u postoperativnom razdoblju od 5 godina kako bi se procjenio njihov klinički ishod te su u ovisnosti o procjeni klasificirani s obzirom na to je li im se rak ponovno pojavio ili ne. Ekspresija gena mjerena je korištenjem Affymetrix oligonukleotidnog *microarray* U133a GeneChip-a. Geni s prosječnim intenzitetom ekspresije manjim od 40 jedinica ili s pozadinskim signalom većim od 100 jedinica su isključeni. Za normalizaciju čipova, setovi sondi skalirani su na ciljanu vrijednost intenziteta od 600 jedinica. Slika 4.1 prikazuje omjer pozitivnih i negativnih uzoraka u seriji.

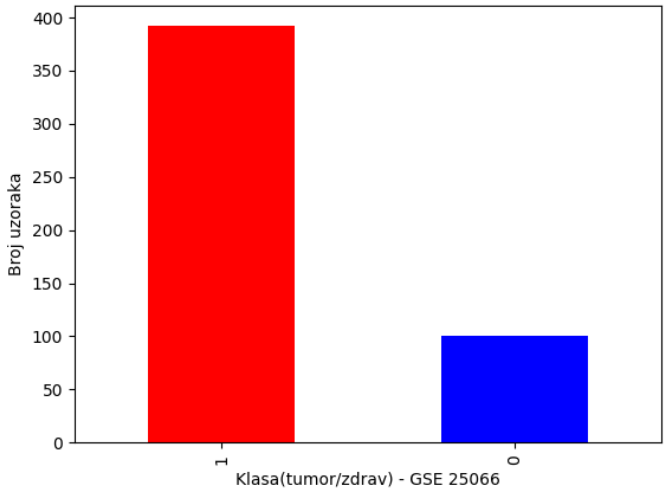


Slika 4.1 *GSE2034 - omjer pozitivnih uzoraka(1) i negativnih(0) glede ponovnog pojavljivanja tumora*

4.1.2 GSE25066

GSE25066 [15] nastao je za potrebe istraživanja koje je za cilj imalo ustanoviti prediktore preživljavanja nakon liječenja BRCA pacijenata kod kojih je tumor nedavno dijagnosticiran.

492 uzoraka sadrži podatke o odgovoru pojedinog pacijenta na neoadjuvantnu kemoterapiju. Pacijenti su praćeni u postoperativnom periodu od minimalno 3 godine. Jednako kao i kod GSE2034, svaki uzorak sadrži brojčane podatke o genskoj ekspresiji 12634 različitih gena. Također, način prikupljanja podataka je isti kao i u GSE2034 seriji - Affymetrix oligonukleotidni *microarray* U133a GeneChip. Podaci su normalizirani korištenjem MAS5 algoritma. Omjer zastupljenosti pojedine klase prikazan je na slici 4.2.



Slika 4.2 *GSE25066 - omjer uzoraka koji su pozitivni(1) i negativni(0) po pitanju nezadovoljavajućeg kliničkog ishoda neoadjuvantne kemoterapije*

4.2 Podjela podataka

Kao što je već spomenuto, jedan od najvećih izazova kod skupova podataka genske ekspresije jest maleni broj uzoraka. Kako bi se ovaj problem minimizirao i kako bi mreže mogle učiti na što je većem moguće broju podataka omjer trening podataka i testnih podataka postavljen je na 73:27. Ovo vrijedi za MLP i GRU modele kao i za eksperiment u okviru MT metode gdje se neoznačeni podaci uzimaju iz skupa koji nije korišten za treniranje i testiranje. Nad trening podacima je nakon podjele provedena unakrsna validacija u 10 preklopa. Nakon unakrsne validacije model je još jednom treniran na svim trening podacima. Što se tiče polunadziranog

učenja, u jednom je eksperimentu veličina testnog seta postavljena na 17%. Razlog dodatnog smanjenja testnog seta leži u načinu tretiranja podataka: podaci za trening podijeljeni su na označene i neoznačene (u omjeru 50:50). Dakle polovici podataka za trening izbrisana je odgovarajuća oznaka. Valja napomenuti kako ovo nije praksa u stvarnom svijetu, ali zbog potrebe provedbe eksperimenta u kojem će svi podaci biti jednakog biološkog značenja korišten je navedeni pristup.

4.3 Odabir značajki i redukcija dimenzionalnosti

Kao što je već rečeno, veliki broj značajki(12,634) predstavlja prepreku za učinkovitu klasifikaciju, posebice kada su skupovi podataka ovako maleni. Stoga je od ključne važnosti provesti odabir gena koja će odrediti gene koji su od značaja za dani problem i zanemariti one manje informativne kako bi se maksimizirali rezultati klasifikacije i smanjila prenaučenost.

Kada su u pitanju metode za odabir značajki, moguć je odabir između sljedećih kategorija: metode filtera, metode omotača (engl. *wrapper methods*), ugradbene metode (engl. *embedded methods*) i hibridne metode. Metode filtera se baziraju na filtriranju značajki koje nisu izgledne da pridonesu poboljšanju rezultata modela. Svakoj je značajki pridodijeljen određeni *score* i one s niskim bivaju zanemarene. Metode omotača evaluiraju važnost značajki na način da generiraju nasumične skupove značajki i svaki skup se ispita na klasifikacijskom algoritmu. Ovo rezultira odabirom značajki koji je specifičan za korišteni algoritam klasifikacije, ali mana ovog pristupa jest da je računalno zahtjevniji od prethodno opisanog. Ugradbene metode kombiniraju prednosti metoda filtera i omotača i paralelno vrše odabir značajki i treniranje algoritma. Posljednja kategorija koja se pojavljuje proteklih godina jesu hibridne metode koje na razne načine kombiniraju pristupe ove tri vrste ciljajući na dobar balans između računalne zahtjevnosti i kvalitetnog odabira značajki.

Osim odabira značajki, ostvarivanje boljih rezultata smanjenjem kompleksnosti ulaznih primjera i rizika prenaučenosti moguće je i ekstrakcijom značajki. U tom se slučaju izvorne značajke transformiraju u novi skup značajki čime se smanjuje dimenzionalnost. Cilj je izvući najbitnije informacije iz skupa podataka i predstaviti ih u novom manje-dimenzionalnom

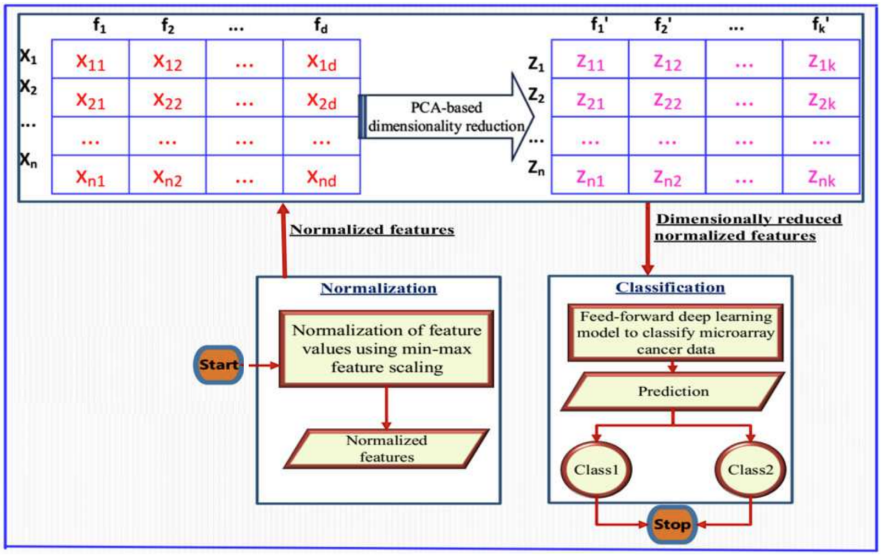
prostoru. Često je i kombiniranje odabira značajki s ekstrakcijom. Valja napomenuti kako fokus ovoga rada nije na pronalaženju optimalne metode odabira značajki niti se uspoređuju rezultati klasifikacije korištenjem različitih metoda. Izgledno je da bi neke od složenijih metoda odabira značajki poboljšale rezultate klasifikacije pa se zainteresirani čitatelj poziva na pregled relevantnih članaka.

Metoda odabira značajki korištena u ovom radu je filter metoda *variance thresholding* koja odbacuje sve značajke koje su ispod zadanog praga varijance s obzirom da nisu od velike koristi za klasifikaciju. Nakon eksperimentiranja s ovom vrijednošću prag je postavljen na 20% što je rezultiralo brisanjem 1961 gena za GSE2034, odnosno 662 gena za GSE25066 skup podataka. Za redukciju dimenzionalnosti odabrana je Analiza glavnih komponenti (engl. *Principal Component Analysis - PCA*). Ova je poznata metoda često korištena u literaturi za analizu podataka genske ekspresije. Iako se na ovu tehniku često gleda kao na tehniku redukcije dimenzionalnosti, ona je zapravo tehnika transformacije podataka nakon koje su podaci dobro "namješteni" za redukciju iz razloga što su poredani po postotku udjela varijance. Drugim riječima, generalna ideja analize glavnih komponenti jest pronaći drukčiji pogled na podatke od interesa u kojem su ti podaci jasnije razdvojivi i kad se takav pogled pronađe moguće je izvršiti redukciju dimenzionalnosti. Slika 4.3 prikazuje poopćeni prikaz cjelokupnog postupka klasifikacije te je iz nje vidljivo u kojoj fazi PCA igra ulogu.
Neka je $\mathbf{X} \in R^{n \times p}$ matrica podataka gdje n označava broj uzoraka, a p broj značajki. Svaka značajka $\mathbf{x}_j$ centrirana je tako da ima srednju vrijednost nula, a standardnu devijaciju 1.

$$\mathbf{x}_j = \mathbf{x}_j - \bar{\mathbf{x}}_j, \quad \text{gdje je} \quad \bar{\mathbf{x}}_j = \frac{1}{n}\sum_{i=1}^{n} x_{ij}. \tag{4.1}$$

Nakon centriranja, cilj PCA-a je pronaći ortogonalne smjerove (glavne komponente) $\mathbf{w}_1, \mathbf{w}_2, \ldots, \mathbf{w}_p$ koji maksimiziraju varijancu projekcija uzoraka na te smjerove. Glavne komponente definirane su kao linearne kombinacije izvornih značajki:

$$\mathbf{z}_k = \mathbf{X}\mathbf{w}_k, \tag{4.2}$$



Slika 4.3 *Radni okvir pristupanja klasifikaciji korišten u ovome radu, preuzeto i prilagođeno iz [31]*

Pritom je $\mathbf{z}_k$ k-ta glavna komponenta, a $\mathbf{w}_k$ vektor težina koji specificira smjer u prostoru značajki. U praksi se PCA može dobiti pronalaženjem svojstvenih vektora kovarijacijske matrice $\mathbf{S}$ iz podataka:

$$\mathbf{S} = \frac{1}{n-1}\mathbf{X}^\top\mathbf{X}. \tag{4.3}$$

Svojstveni vektori $\mathbf{w}_k$ kovarijacijske matrice $\mathbf{S}$ odgovaraju smjerovima glavnih komponenti, dok svojstvene vrijednosti λ_k odgovaraju varijancama duž tih smjerova. Prva glavna komponenta $\mathbf{w}_1$ je svojstveni vektor povezan s najvećom svojstvenom vrijednošću λ_1, druga glavna komponenta $\mathbf{w}_2$ povezana je s drugom najvećom svojstvenom vrijednošću λ_2, i tako dalje. Varijanca koju objašnjava k-ta glavna komponenta dana je svojstvenom vrijednošću λ_k. Ukupna objašnjena varijanca s prvih k komponenti računa se kao:

$$\frac{\sum_{i=1}^{k}\lambda_i}{\sum_{i=1}^{p}\lambda_i}. \tag{4.4}$$

Nakon eksperimentiranja s nekoliko vrijednosti odlučeno je kako će za oba skupa podataka biti objašnjeno 94% varijance za sve klasifikatore($n\_components$ argument PCA klase postavljen

na 0.94). Vrijednosti uzete u obzir za ovaj argument su bile 0.9, 0.92, 0.94, 0.97 i odabrana je u pravilu pružala najbolje rezultate. Kod odabira ove vrijednosti je također bitno paziti da ne dođe do podnaučenosti/prenaučenosti. Odabrana vrijednost od 94% rezultirala je konačnim brojem od 174 značajke neposredno prije klasifikacije za GSE2034 skup, odnosno 288 značajki za GSE25066. Na slikama 4.4 vidljiva je raspodjela podataka iz opisanih skupova nakon primjene PCA-a. Valja napomenuti kako zbog velike dimenzionalnosti tri dimenzije neće biti dostatne da se savršeno odvoje klase pošto je njima u oba slučaja opisano ispod 20% ukupne varijance.

(a) *GSE2034 - Projekcija podataka na prve tri glavne komponente s pripadajućim udjelima objašnjene varijance*

(b) *GSE25066 - Projekcija podataka na prve tri glavne komponente s pripadajućim udjelima objašnjene varijance*

Slika 4.4 *Trodimenzionalni prikaz transformiranih podataka korištenih skupova*

4.4 Struktura višeslojnog perceptrona i opis metodologije

Višeslojni perceptron, kao i sve neuronske mreže izrađene u PyTorchu (između ostalog i GRU model opisan u nastavku rada), nasljeđuje osnovnu nn.Module klasu. Prilikom dizajniranja neuronske mreže jedan od bitnijih odabira jest broj slojeva. Što je on veći to mreža bolje može učiti podatke, no isto se tako i uvodi opasnost od prenaučenosti (kod mreža s 5 ili više skrivenih slojeva ovo je jako čest slučaj). Tijekom eksperimentiranja na jednom skupu podataka je zaključeno da se mreža od 3 skrivena sloja čini optimalnom za ovaj problem te je konačan

FNN tako modeliran. Broj neurona ulaznog sloja jednak je broju značajki, dok je broj neurona izlaznog sloja 2 (odgovara broju mogućih predviđanja). Broj neurona u skrivenim slojevima je jedan od hiperparametara koji se nastoji optimizirati. Isto tako primjetno je znatno poboljšanje rezultata klasifikacije korištenjem normalizacije po grupi (engl. *batch normalization*) te stoga ona slijedi iza skrivenih slojeva. Normalizacijom izlaznih vrijednosti skrivenih slojeva prije iduće aktivacijske funkcije postiže se stabilizacija, smanjuje utjecaj ekstremnih vrijednost (engl. *outliers*) i ubrzava konvergencija. Višeslojni perceptron koristi ReLU aktivacijsku funkciju za sve slojeve osim posljednjeg kod kojeg je očit odabir sigmoide kao aktivacijske funkcije. U svaki skriveni sloj uveden je i *dropout* sloj koijm se nasumično isključuju neuroni tijekom treninga što doprinosi boljoj generalizaciji. Što se tiče optimizacije hiperparametara, dvije često korištene tehnike su nasumično pretraživanje (engl. *random search*) i mrežno pretraživanje (engl. *grid search*). Za ovaj je projekt korištena prva spomenuta tehnika, a razlog ovog odabira jest da nemaju svi hiperparametri jednaku važnost za uspjeh modela, a nasumično pretraživanje omogućuje isprobavanje više mogućih vrijednosti te je samim time povećana vjerojatnost pronalaska kvalitetnog rješenja. Ukupni broj isprobanih kombinacija postavljen je na 36 a idući isječak koda pokazuje moguće vrijednosti ili mogući raspon svakog hiperparametra.

```
def get_random_hyperparams():
    random_hyperparams = {
        "learning_rate": 10 ** random.uniform(np.log10(1e-4), np.log10(1e-2)
    ),
        "momentum": random.uniform(0.9, 0.999),
        "batch_size": random.choice([8,16,32,64]),
        "num_epochs": random.choice([28,30,32,34,36,40,42]),
        "dropout": random.uniform(0.05, 0.25),
        "hidden_size": random.choice([66,90, 100, 120,150, 200]),
        "regularization": 10 ** random.uniform(np.log10(1e-6), np.log10(1e
    -3))
    }
    return random_hyperparams
```

U tablici koja slijedi dan je prikaz optimalnih hiperparametara za skupove GSE2034 i GSE25066,

Hiperparametar	GSE2034	GSE25066
Stopa učenja	0.0065	0.0028
Batch size	64	64
Dropout	0.2462	0.0731
Neuroni u skrivenom sloju	120	100
L1 Regularizacija	3.045e-5	1.4974e-5

Tablica 4.1 *Najbolja kombinacija hipeparametara za pojedini skup podataka pronađena nasumičnom pretragom*

Za kriterij vrednovanja, odnosno *score function* uzeta je linearna kombinacija točnosti (engl. *accuracy*) i F1 (omjer 75:25). Kod ovog i sličnih problema u pravilu je cilj maksimizirati broj pogođenih pozitivnih uzoraka s obzirom da je veća greška ne otkriti uzorak kod kojeg dolazi do ponovne pojave raka nego lažno klasificirati negativan uzorak kao pozitivan. Dakle cilj je maksimizirati odziv (engl. *recall*). Međutim, s obzirom na činjenicu da oba skupa podataka sadrže više pozitivnih uzoraka nego negativnih odziv nije bio uzet u obzir kod odabira optimalnih hiperparametara jer je model ionako skloniji predviđanju pozitivnih uzoraka. Shodno tome nisu korištene tehnike koje bi smanjile nebalansiranost skupa podataka poput dodjele težinskih faktora pojedinoj klasi ili *upsampling*/*downsampling* (koji kod ovako malih skupova podataka nije praktičan). Detaljniji opis značenja metrika slijedi u poglavlju s rezultatima. Također, valja napomenuti kako model koristi Xavier inicijalizaciju težina za sve linearne slojeve, što omogućava stabilniji početak treninga s ujednačenijim vrijednostima težina i pomaže pri izbjegavanju zasićenja gradijenata. Svaki parametar pomaka slojeva postavljen je na nisku vrijednost 0.01, čime se modelu pruža inicijalna prednost prema blagim gradijentima i stabilnijem učenju.

4.5 Primjena metode usrednjenog učitelja

Iako analiza podataka genske ekspresije pruža veliki potencijal pri predviđanju različitih vrsta raka, malen broj dostupnih uzoraka ostaje znatna prepreka kada je u pitanju razvoj pouzdanijih klasifikatora. Tradicionalnim nadziranim učenjem velik broj nepotpunih podataka genske ekspresije biva zanemaren. Jedan od ciljeva ovoga rada jest razviti model polunadziranog

učenja koji pruža zadovoljavajuće rezultate, a navedeno se pokušava postići već spomenutom MT metodom. Za *student* i *teacher* modele odabran je u prethodnom poglavlju opisan model višeslojnog perceptrona. Na slici 4.5 je prikazan ovaj cjelokupni algoritam.

Algorithm 1: Mean Teacher Algorithm

Data: train set $(\mathcal{X}, \mathcal{Y})$, Unlabel data($\mathcal{Z}$)
Hyper parameters: r, α, $p1$, $p2$, $\rho 1$, $\rho 2$, $epochs$;
Create Model : $student(\theta)$, $teacher(\theta')$;
Train *teacher* for 1 epoch;
while *epochs* **do**
  while *steps* **do**
    1: Insert noise with probability as per strategies ($p1$, $p2$, $\rho 1$, $\rho 2$) in $\mathcal{X}$ i.e. $\mathcal{X}_\eta$;
    2: $student(\mathcal{X}_\eta) = \mathcal{Y}_\eta$;
    3: Classification cost ($C(\theta)$) =Binary Cross Entropy($\mathcal{Y}$,$\mathcal{Y}_\eta$);
    4: Again create different noise data as mentioned in step 1 i.e. $\mathcal{X}_{\eta'}$;
    5: $teacher(\mathcal{X}_{\eta'}) = \mathcal{Y}_{\eta'}$;
    6: Calculate Consistency cost $J(\theta)$=Mean Squared Error($\mathcal{Y}_\eta$,$\mathcal{Y}_{\eta'}$);
    7: Calculate Overall cost $O(\theta) = \lambda C(\theta) + (1 - \lambda)J(\theta)$;
    8: Calculate *gradients*, $O(\theta)$ w.r.t θ ;
    9: Apply *gradients* to θ;
    10: Update Exponential Moving average of θ to θ' i.e. $\theta'_t = \alpha\theta'_{t-1} + (1 - \alpha)\theta_t$;
  end
end

Slika 4.5 *Algoritam metode usrednjenog učitelja, preuzeto iz [28]*

Kao što je već rečeno, ova je metoda bolje prilagođena velikim skupovima podataka. Neuspješno učenje kompleksnih podataka tijekom inicijalnih epoha lako može dovesti do toga da učitelj kasnije tijekom treninga potiče netočna predviđanja. Također pronalaženje optimalnih hiperparametara znatno je otežano i stabilnost eksponencijalnog težinskog prosjeka student modela predstavlja također izazov. Metoda je primijenjena na dva načina. U prvom je nakon podjele na trening i testni skup izbrisana oznaka polovici trening podataka. U drugom načinu pristup je sljedeći: *skup* GSE2034 se dijeli kao u MLP modelu na trening i testni skup(omjera 73:27). Nakon toga se podaci iz skupa GSE25066 dodaju trening skupu kao neoznačeni. Odluka da se podaci iz GSE25066 skupa tretiraju kao neoznačeni, a ne obrnuto jest da taj skup sadrži više uzoraka, a korištenje polunadiranog učenja u praksi najčešće podarzumijeva skupove podataka u kojima prevladavaju neoznačeni podaci. Iako ovaj pristup omogućuje više podataka za treniranje, valja napomenuti neke njegove nedostatke. S obzirom na drukčije biološko značenje ova dva skupa podataka (jedan razlikuje uzorke s obzirom na ponovno pojavljivanje,

a drugi s obzirom na odgovor organizma na kemoterapiju), biološki mehanizmi i karakteristike koje utječu na ishod klasifikacije nisu nužno isti. Iako je broj gena isti, obrasci ekspresije jednog skupa nisu nužno primjenjivi na drugi što bi potencijalno modelu moglo predstavljati izazov (u polunadziranom učenju općenito neoznačeni podaci mogu rezultirati dodatnim šumom). Namjera ovog pristupa jest da neka zajednička svojstva ekspresijskih profila ipak pridonesu boljem učenju modela. Glede broja epoha, tijekom oba pristupa isprobane su vrijednosti u rasponu od 20 do 60, dok su konačni rezultati opisani u idućem poglavlju postignuti s 30 epoha. Ono čemu se težilo jest da u ranim fazama treniranja naglasak bude na pogrešci klasifikacije, a kasnije, kako trening odmiče *consistency cost* raste sve više i više. Ovo je rezultiralo stabilnijim modelom učitelja u odnosu na ostale izrađene klasifikatore. Ključnu ulogu u treniranju ima težinski faktor konzistencije λ, koji omogućuje kontrolu stupnja konzistencije između studenta i učitelja tijekom epoha. Na sljedećem isječku koda prikazane su *sigmoid_rampup* i *get_current_consistency_weight()* funkcije te u nastavku slijedi njihov opis.

```
def sigmoid_rampup(current, rampup_length):
    if rampup_length == 0:
        return 1.0
    else:
        current = np.clip(current, 0.0, rampup_length)
        phase = 1.0 - current / rampup_length
        return float(np.exp(-5.0 * phase * phase))

def get_current_consistency_weight(const_weight, const_rampup, epoch):
    cw = const_weight * sigmoid_rampup(epoch, const_rampup)
    return cw
```

Prva funkcija omogućuje postupno povećanje *consistency cost*-a tijekom definiranog broja epoha(za trening od 30 epoha ovaj je parametar postavljen na 12). Već spomenuti parametar λ množi se s povratnom vrijednošću *get_current_consistency_weight()* funkcije. Tijekom prvih 5 epoha ovaj je hiperparametar postavljen na 0.003 čime se efektivno zanemaruje konzistentnost. Ova vrijednost raste do 60 do 12. epohe i s tako postavljenim parametrima *consistency cost* krajem treninga dosegne vrijednost od oko 0.4. U većini primjera primjene ove metode kažnjavanje nekonzistentnosti se uvodi brže nego što je implementirano u ovom konkretnom problemu, no takav pristup ovdje dovodi do nasumičnih predviđanja. Ažuriranje težina *teacher*

modela se također odvija postepeno na način da se koristi stvarni prosjek sve dok eksponencijalni prosjek ne bude točniji. U ranoj fazi treninga parametar α koji kontrolira ažuriranje težina bude postavljen na 0.99 i ostaje toliki do kraja treninga.

Također je već naglašena važnost dodavanja šuma ovom modelu. Prilikom odabira strategije za dodavanje šuma od koristi je bilo razumjeti prirodu šuma koji je uveden prilikom dobivanja podataka genske ekspresije koristeći već opisanu tehniku koja je korištena za dobivanje GSE2034 i GSE25066 *skupova*. S obzirom na navode i grafičke prikaze iz [21] vrijedi nekoliko temeljnih spoznaja. Šum dobiven tijekom pripreme uzoraka jest malen i zanemariv u odnosu na šum hibridizacije. Šum hibridizacije ovisi o jačini ekspresije na sljedeći način: za visoke vrijednosti raspodjela šuma je Poissonova, a za niske je kompleksnija i nema pravila. Manje su vrijednosti genske ekspresije sklonije većem šumu. Nastavno novostečenim spoznajama, uvedene su tri razine šuma prema kvartilima vrijednosti genske ekspresije. Veći šum se primjenjuje na vrijednosti ispod prvog kvartila (Q1) korištenjem Gaussove distribucije radi jednostavnosti. Ista je distribucija primijenjena i na vrijednosti između prvog i trećeg kvartila (Q1 i Q3), ali s nešto manjim šumom. Najmanji šum se primijenjuje na vrijednosti iznad 3. kvartila te je on izvučen iz Poissonove distribucije.

4.6 Metodologija propusne povratne ćelije

Posljednji model implementiran u PyTorchu za potrebe ovog projekta jest povratna neuronska mreža - GRU. Ovaj je model općenito često korištenu u mnogim područjima genomike uključujući obradu skupova podataka koji sadrže razine genske ekspresije uzoraka tijekom više vremenskih jedinica(engl. *time series gene expression data*). Iako korišteni skupovi podataka nisu ove prirode, GRU i LSTM su često implementirani modeli i za klasifikacijske probleme slične kao u ovom radu. Povratne mreže imaju sposobnost učenja složenih korelacija među genima i uočavanja uzoraka koji omogućuju bolje razumijevanje genskih puteva(engl. *gene pathway*). Postupak modeliranja bio je temeljen na optimizaciji dvaju hiperparametara povratne mreže: broj povratnih slojeva i broj neurona u povratnom sloju. Algoritam postupka za pronalaženje ovih hiperparametara je sljedeći:

1. Priprema ulaznih podataka - matrice $E = (e_{ij})_{n \times m}$, gdje je n broj uzoraka koji se analiziraju, a m broj gena, tj. ekspresija gena koja određuje stanje svakog pojedinog uzorka

2. Definiranje raspona hiperparametara i koraka promjene: slojevi layers $= 1, 2, 3$ (broj povratnih slojeva u RNN-u); $k = 50, 55, \ldots, 85$, $dk = 5$ (raspon i korak promjene broja neurona u rekurentnim slojevima).

3. Podjela skupa podataka podskup za treniranje i testiranje u omjeru 73:27

4. Inicijalizacija broja povratnog sloja: $layer = 1$.

5. Inicijalizacija početne vrijednosti broja neurona u povratnim slojevima: $k = 50$.

6. Treniranje modela. Vrši se unakrsna validacija u 10 preklopa i računaju se relevantne metrike i vrijednost funkcije gubitka.

7. Testiranje modela na testnom skupu.

8. Ako je $k < k_{\max}$, povećava se broj neurona u povratnim slojevima za 5 (tj. $k = k + 5$) i ponavlja se 6. korak ovog postupka. Ukoliko su metrike najuspješnije sprema se navedena kombinacija parametara.

9. Ako je broj povratnih slojeva manji od maksimalnog broja ($layer < \text{layers}_{\max}$), povećava se broj sloja za 1, te se ponavlja 5. korak ovog algoritma.

Navedeni je algoritam pokazao da od spomenutih mogućih kombinacija najbolje rezultate daje mreža s 2 GRU sloja od kojih svaki ima po 75 neurona. Također razmotrena je i mogućnost dvosmjerne propusne povratne ćelije, međutim nekoliko uzastopnih postupaka učenja je pokazalo da se ovime rezultati pogoršaju. Dvosmjerna ćelija ima dva skrivena stanja te praktički funkcionira spajanjem dvaju neovisnih GRU modela. Kao i u običnoj neuronskoj mreži, korištena je *dropout* tehnika.

5. Rezultati

U ovom se poglavlju navode postignuti rezultati prethodno opisanih modela. Za validaciju performansi predloženih pristupa su korištene standardne metrike poput točnosti (engl. *accuracy*), preciznosti (engl. *precision*), odziva (engl. *recall*), F1 mjere, krivulje operativnih karakteristika (engl. *Receiver operating characteristic*, ROC) i matrice konfuzije. Točnost procjenjuje ukupnu prediktivnu sposobnost modela uzimajući u obzir četiri veličine: *True Positive (TP), True Negative (TN), False Positive (FP)* i *False Negative (FN)*. Predstavlja omjer ispravno predviđenih uzoraka u odnosu na ukupan broj uzoraka u skupu podataka:

$$\text{ACC} = \frac{TP + TN}{TP + TN + FP + FN}$$

gdje TP predstavlja ispravna pozitivna, TN ispravna negativna, FP pogrešna pozitivna, a FN pogrešna negativna predviđanja. Preciznost je omjer ispravno predviđenih pozitivnih opažanja u odnosu na ukupan broj predviđenih pozitivnih:

$$\text{PREC} = \frac{TP}{TP + FP}$$

Odziv ili osjetljivost predstavlja omjer ispravno predviđenih pozitivnih opažanja u odnosu na sva opažanja u stvarnoj klasi:

$$\text{REC} = \frac{TP}{TP + FN}$$

U konačnici F1 mjera neutralizira pristranost odziva i preciznosti i dana je sljedećom formulom:

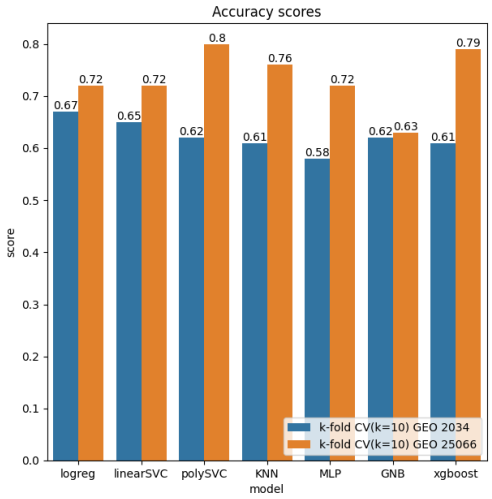
$$\text{F1} = 2 \cdot \frac{\text{PREC} \cdot \text{REC}}{\text{PREC} + \text{REC}}$$

Već je spomenuto da su skupovi podataka korišteni u ovom radu prvi puta spomenuti u [13] gdje se klasifikacija vršila kombiniranjem *Random Projection-a*(RP) s ostalim metodama za redukciju dimenzionalnosti. Konačni rezultati ovoga rada dani su u idućoj tablici. Kod donošenja zaključaka treba imati na umu da je primarna svrha RP-a pojednostavljivanje podataka, a ne maksimizacija rezultata klasifikacije.

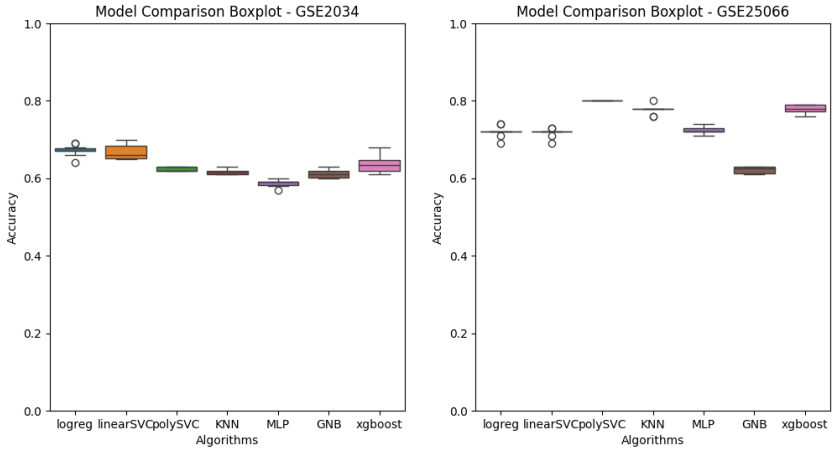
Metoda	**GSE2034**	**GSE25066**
RP	59.59	66.90
FS + RP	61.25	**71.07**
FS + RP + LDA	60.56	69.56
FS + RP + PFS	**61.55**	67.06
RP + FS	60.89	68.65
RP + PCA	54.25	60.44
RP + LDA	58.87	69.21

Tablica 5.1 *Najviša vrijednost točnosti klasifikacije na skupovima GSE2034 i GSE25066[13]*

Nadalje, prije samog osvrtanja na rezultate u prethodnom poglavlju opisanih modela, izvršena je unakrsna validacija u 10 preklopa koristeći najpoznatije klasifikacijske algoritme strojnog učenja kako bi se dobio bolji uvid u to koji se rezultati mogu očekivati i koji algoritmi pokazuju najviše potencijala. Postupak je ponovljen 10 puta. Grafički prikaz rezultata prve izvedene unakrsne validacije kao i kutijasti dijagram točnosti svih mjerenja na oba skupa podataka vidljivi su na slikama. 5.2.



Slika 5.1 *Točnost klasifikacije nekih od najčešćih algoritama strojnog učenja na skupovima GSE2034 i GSE25066*



Slika 5.2 *Kutijasti dijagram točnosti nekih od najčešćih algoritama strojnog učenja na skupovima GSE2034 i GSE25066*

Korišteni algoritmi su redom: logistička regresija, Linearni SVC(*Support Vector Classification*), Poly SVC, K Nearest Neighbours, MLP, naivan Bayesov klasifikator te XGBoost. Treba napomenuti kako su svi prikazani modeli instance gotovih sklearn klasa gdje su pritom korištene zadane postavke, a optimizacija hiperparametara također nije vršena niti na jednom

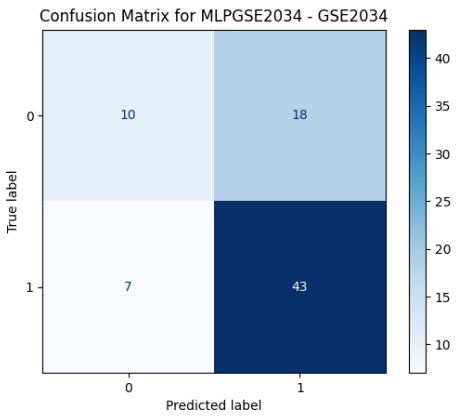
modelu. Vidljivo je da je MLP, uz naivan Bayesov klasifikator, bio najneuspješniji kad je u pitanju klasifikacija, dok su se algoritmi poput SVC-a i XGBoosta pokazali dobrim potencijalnim rješenjima za ovaj problem.

Model višeslojnog perceptrona implementiran u PyTorchu pokazao je određeno poboljšanje rezultata u odnosu na gore opisan sklearn model. Glavni modeli ovoga rada navedeni u prethodnom poglavlju trenirani su i evaluirani na testnom skupu 10 puta (s 10 različitih podjela trening i testnog skupa, tj. korištenjem 10 različitih *seed* vrijednosti). Na tablici 5.3 su dane prosječne performanse modela višeslojnog perceptrona, kao i maksimalne vrijednosti pojedine metrike.

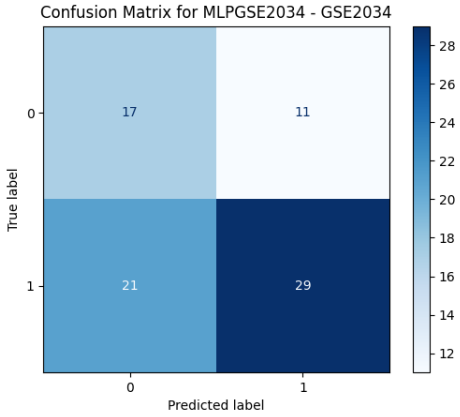
Skup podataka	**ACC**	**REC**	**PREC**	**F1**
GSE2034 - Prosječna vrijednost	0.6347	0.7788	0.6913	0.7274
GSE2034 - Najviša postignuta vrijednost	0.6795	0.94	0.725	0.7748
GSE25066 - Prosječna vrijednost	0.8038	0.9037	0.8634	0.8831
GSE25066 - Najviša postignuta vrijednost	0.8421	0.9174	0.8929	0.905

Tablica 5.2 *Rezultati klasifikacije višeslojnog perceptrona na testnom skupu*

Slike 5.3 i 5.4 prikazuju matrice konzufije krajnjih vrijednosti, odnosno klasifikacija s najvećom i najmanjom točnošću.

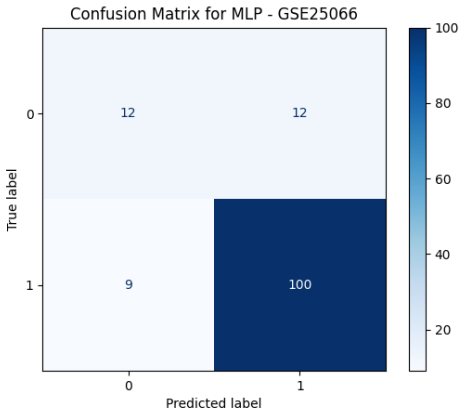


(a) *GSE2034 - Matrica konfuzije najtočnije klasifikacije*

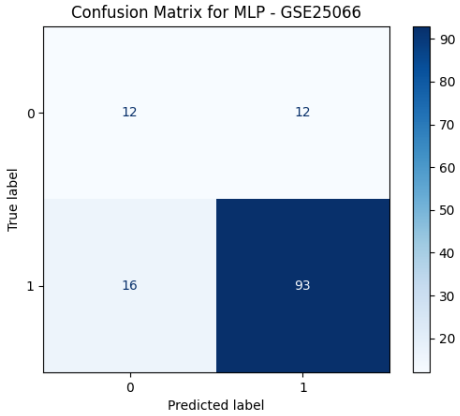


(b) *GSE2034 - Matrica konfuzije najmanje točne klasifikacije*

Slika 5.3 *Matrice konfuzije MLP modela- GSE2034*



(a) *GSE25066 - Matrica konfuzije najtočnije klasifikacije*



(b) *GSE25066 - Matrica konfuzije najmanje točne klasifikacije*

Slika 5.4 *Matrice konfuzije MLP modela- GSE25066*

Kao što je već rečeno, ovaj je model korišten i kao model studenta i učitelja u MT metodologiji. U prvom eksperimentu(tablica 5.3) 17% podataka otpada na testni skup, a u trening skupu polovica podataka biva tretirana kao neoznačena. Model je treniran i evaluiran na testnom skupu 10 puta jednako kao i u drugom eksperimentu(tablica 5.4) gdje se 27% podataka koristi za testiranje, a neoznačeni se podaci uzimaju iz GSE25066 skupa.

Skup podataka	ACC	REC	PREC	F1
GSE2034 - Prosječna vrijednost(S)	0.6518	0.8502	0.7046	0.767
GSE2034 - Najviša postignuta vrijednost(S)	0.7143	0.94	0.7931	0.8219
GSE2034 - Prosječna vrijednost(T)	0.6552	0.8408	0.7055	0.7639
GSE2034 - Najviša postignuta vrijednost(T)	0.7347	0.94	0.8148	0.8295
GSE25066 - Prosječna vrijednost(S)	0.7952	0.8985	0.8567	0.8761
GSE25066 - Najviša postignuta vrijednost(S)	0.8314	0.956	0.8841	0.8967
GSE25066 - Prosječna vrijednost(T)	0.7963	0.9059	0.8618	0.8773
GSE25066 - Najviša postignuta vrijednost(T)	0.8382	0.9559	0.9265	0.9028

Tablica 5.3 *Metoda usrednjenog učitelja - performanse modela studenta(S) i učitelja(T) tretiranjem polovice podataka za trening kao neoznačenih*

Skup podataka	ACC	REC	PREC	F1
GSE2034 - Prosječna vrijednost(S)	0.6499	0.876	0.6735	0.7623
GSE2034 - Najviša postignuta vrijednost(S)	0.6923	0.98	0.7018	0.9065
GSE2034 - Prosječna vrijednost(T)	0.6518	0.898	0.6702	0.7671
GSE2034 - Najviša postignuta vrijednost(T)	0.6795	0.9641	0.6825	0.8

Tablica 5.4 *Metoda usrednjenog učitelja - performanse modela studenta(S) i učitelja(T) na GSE2034 skupu korištenjem uzoraka iz GSE25066 skupa kao neoznačenih*

Rezultati prvog eksperimenta pokazuju slične slične performanse u odnosu na model višeslojnog perceptrona iako je korišten manji broj podataka. Drugi eksperiment nije pridonio značjnom poboljšanju rezultata, no kombiniranje skupova podataka nije ni naštetilo performansama u velikoj mjeri. Također valja primjetiti kako razlika između performansi modela studenta i učitelja nije značajna te je bio nerijetki slučaj da student poluči bolje rezultate. Treba napomenuti kako su rezultati gore prikazanih tablica uzeti na samom kraju treninga odnosno u 30. epohi. Gotovo uvijek je maksimalna vrijednost točnosti postignuta u nekoj od epoha koje joj prethode. Tablica 5.5 se odnosi na prvi eksperiment(skup GSE25066) te prikazuje jedan od primjera kretanje metrika od interesa s povećanjem broja epoha. Da se primjetiti kako mreža studenta uči u početnim epohama(vrijednosti naglašene u tablici), dok se metrike modela učitelja poboljšavaju u nešto kasnijim fazama treninga(krajem treninga često daljnje poboljšanje

nije moguće pa se kao potencijalno rješenje može razmotriti *early stopping*). Plavom i crvenom bojom u tablici su označene najviše postignute vrijednosti točnosti za model studenta i učitelja. U ovom konkretnom primjeru student postiže najbolje rezultate u 6. epohi dok model učitelja kao eksponencijalni težinski prosjek njegovih težina svoj vrhunac postiže dvije epohe kasnije. Kako bi tablica bila što sažetija neke su epohe u kasnijim fazama izostavljene iz prikaza, ali uzorak je isti kao i u prisutnim.

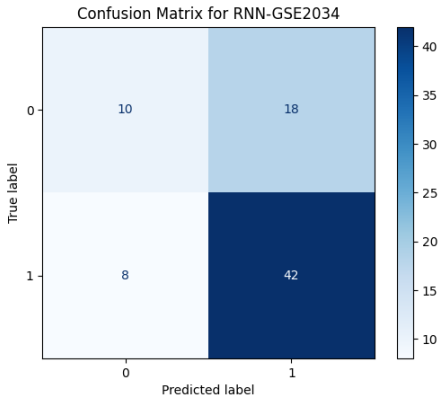
Epoha	*Student Model*				*Teacher Model*			
	ACC	REC	PREC	F1	ACC	REC	PREC	F1
Prije treninga	**0.3308**	0.2294	0.8333	0.3597	0.3083	0.1651	0.9474	0.2812
1	**0.5714**	0.5596	0.8714	0.6816	0.4737	0.4128	0.8824	0.5625
2	**0.6692**	0.7248	0.8495	0.7822	0.6241	0.6239	0.8831	0.7312
3	**0.7068**	0.7706	0.8571	0.8116	0.6541	0.6972	0.8539	0.7677
4	**0.7368**	0.8165	0.8558	0.8357	0.7068	0.7706	0.8571	0.8116
5	**0.7744**	0.8624	0.8624	0.8624	0.7293	0.7982	0.8614	0.8286
6	0.7895	0.8807	0.8649	0.8727	0.7444	0.8257	0.8571	0.8411
7	0.7895	0.8899	0.8584	0.8739	0.7594	0.8440	0.8598	0.8519
8	0.7744	0.8807	0.8496	0.8649	0.7820	0.8807	0.8571	0.8688
9	0.7594	0.8716	0.8407	0.8559	0.7820	0.8807	0.8571	0.8688
10	0.7669	0.8807	0.8421	0.8610	0.7744	0.8807	0.8496	0.8649
11	0.7519	0.8624	0.8393	0.8507	0.7820	0.8899	0.8509	0.8700
12	0.7594	0.8807	0.8348	0.8571	0.7744	0.8807	0.8496	0.8649
15	0.7594	0.8716	0.8407	0.8559	0.7744	0.8807	0.8496	0.8649
16	0.7519	0.8624	0.8393	0.8507	0.7744	0.8807	0.8496	0.8649
17	0.7519	0.8624	0.8393	0.8507	0.7669	0.8807	0.8421	0.8610
20	0.7444	0.8440	0.8440	0.8440	0.7744	0.8716	0.8559	0.8636
25	0.7519	0.8532	0.8455	0.8493	0.7669	0.8624	0.8545	0.8584
30	0.7519	0.8532	0.8455	0.8493	0.7519	0.8532	0.8455	0.8493

Tablica 5.5 *Evaluacija student i učitelj modela kroz epohe(GSE25066)*

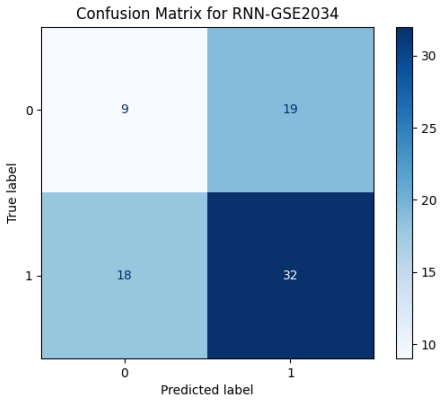
Najslabije performanse pružao je posljednji GRU model, posebice kad je u pitanju GSE2034 skup podataka. Primjetno je da svi algoritmi bolje klasificiraju skup podataka GSE25066 što je, naravno, posljedica veličine skupa za treniranje. Općenito, treniranje na skupu GSE2034 je bilo izazovno po pitanju stabilnosti i konzistentnosti modela te je modelima bilo teško učiti uzorke iz tako malog skupa podataka. Za GRU model je ovo još više izraženo jer su povratne mreže kompleksnije i zahtjevaju više podataka za efikasno treniranje. Rezultati propusne povratne ćelije su dani u tablici 5.6, a nakon njih slijede matrice konzufije klasifikacija s najvećom i najmanjom točnošću(slike 5.5 i 5.6).

Skup podataka	**ACC**	**REC**	**PREC**	**F1**
GSE2034 - Prosječna vrijednost	0.6103	0.718	0.6838	0.6993
GSE2034 - Najviša postignuta vrijednost	0.6667	0.84	0.7174	0.7636
GSE25066 - Prosječna vrijednost	0.791	0.9156	0.8434	0.8772
GSE25066 - Najviša postignuta vrijednost	0.8271	0.9633	0.8609	0.9004

Tablica 5.6 *Rezultati klasifikacije višeslojnog perceptrona na testnom skupu*

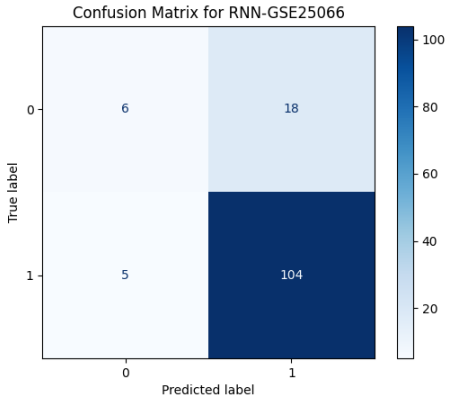


(a) *GSE2034 - Matrica konfuzije najtočnije klasifikacije*

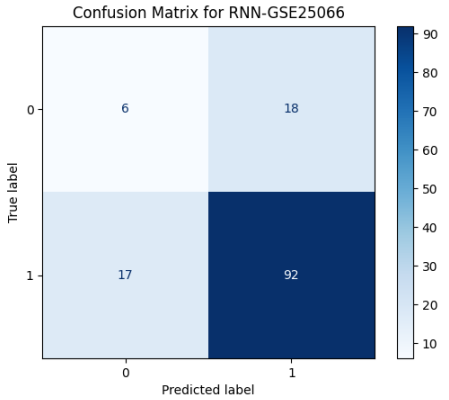


(b) *GSE2034 - Matrica konfuzije najmanje točne klasifikacije*

Slika 5.5 *Matrice konfuzije GRU modela - GSE2034*



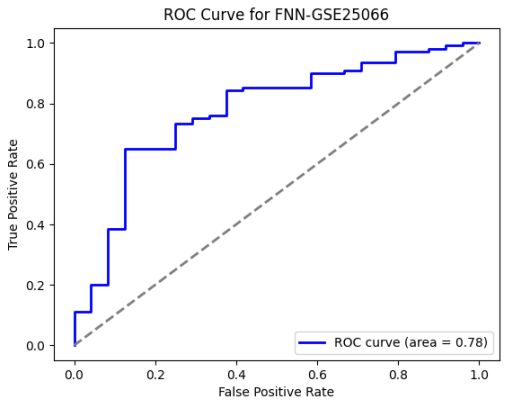
(a) *GSE25066 - Matrica konfuzije najtočnije klasifikacije*



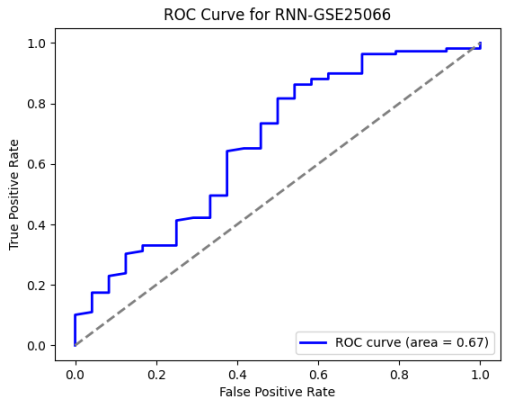
(b) *GSE25066 - Matrica konfuzije najmanje točne klasifikacije*

Slika 5.6 *Matrice konfuzije GRU modela - GSE25066*

Da je FNN uspješniji u klasifikaciji ukazuje i ROC krivulja. S njom usko povezana metrika AUC (engl. *Area Under The Curve*) pokazuje koliko je model uspješan u odvajanju klasa. Pritom veća vrijednost AUC-a ukazuje na bolju sposobnost klasifikacije, dok vrijednost od 0.5 predstavlja razinu slučajnosti. Na ovoj krivulji, os y predstavlja osjetljivost modela (*True Positive Rate*), dok os x predstavlja specifičnost (*False Positive Rate*). Ključna svrha ROC krivulje je vizualizirati kompromis između osjetljivosti i specifičnosti modela, omogućujući procjenu njegove sposobnosti razlikovanja između pozitivnih i negativnih klasa. ROC krivulje (slike 5.7) pokazuju da je FNN model postigao AUC od 0.78, dok je RNN model postigao AUC od 0.67. AUC vrijednost od 0.78 kod FNN modela sugerira umjereno snažnu klasifikacijsku sposobnost, što ukazuje da model može razumno razlikovati slučajeve slučajeve pozitivnog i negativnog odgovora na kemoterapiju. S druge strane, AUC od 0.67 za RNN model, iako viši od razine slučajnosti, upućuje na slabiju sposobnost razlikovanja.



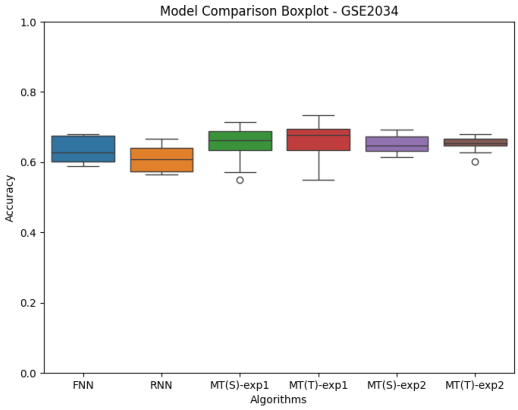
(a) *ROC krivulja MLP klasifikatora*



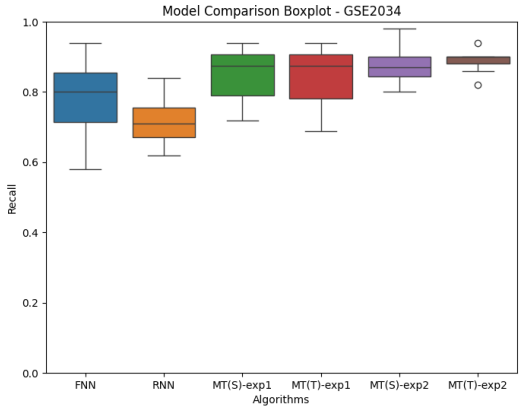
(b) *ROC krivulja GRU klasifikatora*

Slika 5.7 *ROC krivulje - GSE25066*

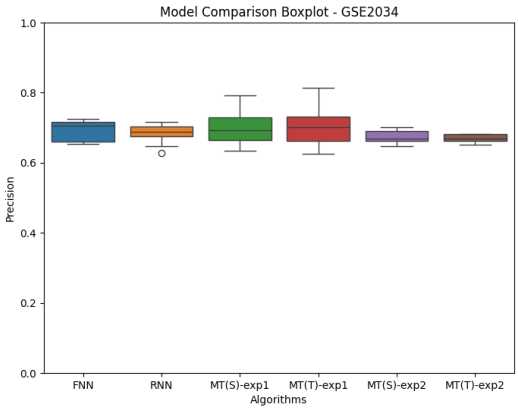
U nastavku(slike 5.8 i 5.9) slijede grafički prikazi performansi pojedinog algoritma na oba skupa podataka pomoću kutijastog dijagrama. Iz slika se daju očitati medijalna vrijednost(horizontalna crta u sredini kutije), prvi kvartil i treći kvartil(rubovi kutija) te vanjske linije(engl. *whiskers*) koje predstavljaju raspon podataka. Točke na grafičkom prikazu(engl. *outliers*) predstavljaju podatke koji se nalaze izvan normalne distribucije raspona. S obzirom na činjenicu da je na GSE2034 skupu podataka izvršen eksperiment više nego na skupu GSE25066, grafički prikazi sadrže dva modela više(student i učitelj modele mjerenja koje je na grafu notirano s "exp2", a podrazumijeva korištenje GSE25066 skupa kao izvora neoznačenih podataka). Na skupu podataka GSE25066 gdje su klasifikacijski rezultati konzistentniji u odnosu na GSE2034 interkvartilni je raspon MLP-a uži u odnosu na ostale modele, posebice povratnu propusnu ćeliju što ukazuje na manju varijabilnost performansi.



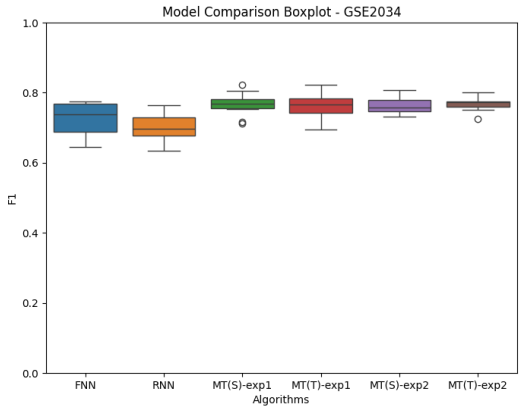
(a) *Točnost*



(b) *Odziv*

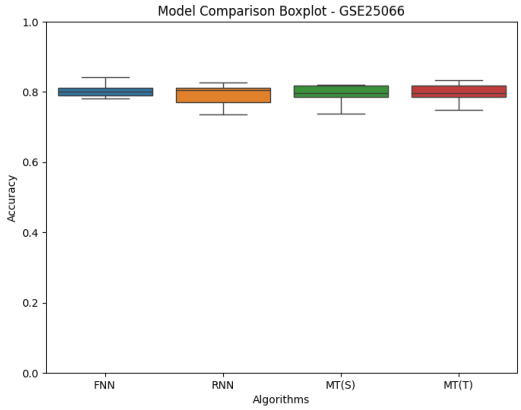


(c) *Preciznost*

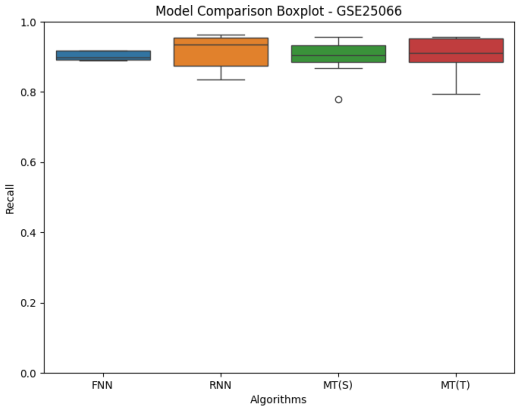


(d) *F1 mjera*

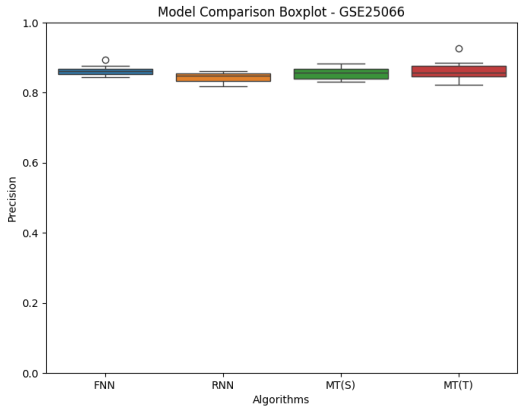
Slika 5.8 *GSE2034 - kutijasti dijagram*



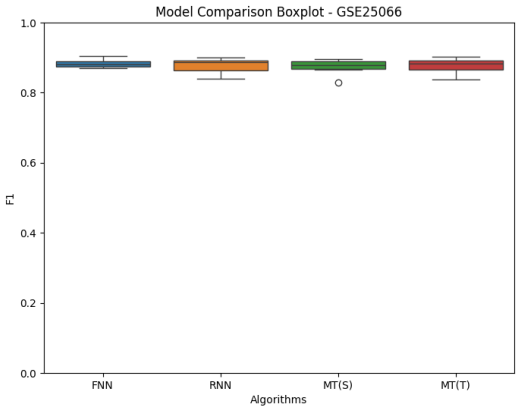
(a) *Točnost*



(b) *Odziv*



(c) *Preciznost*



(d) *F1 mjera*

Slika 5.9 *GSE25066 - kutijasti dijagram*

6. Zaključak

Potencijal dubokog učenja u istraživanju ljudskog genoma već je neko vrijeme neupitan, a s obzirom na brzinu novih spoznaja u umjetnoj inteligenciji i strojnom učenju, za očekivati je značajne napretke u razumijevanju bolesti i područjima poput personalizirane medicine. Nada je da će razvoj dubokog učenja pružiti odgovore na pitanja na koja ih zbog kompleksnosti podataka ove prirode, dosad znanstvenici još nisu pronašli. Jedna od prepreka koja usporava napredak jest ograničenost podataka koja dovodi do težeg učenja. U ovom je radu fokus stavljen na razvijanje modela dubokog učenja koji će za cilj imati vršenje binarne klasifikacije raka dojke na temelju podataka o genskoj ekspresiji. Osim razvoja i optimizacije modela višeslojnog perceptrona i propusne povratne ćelije, jedan od glavnih ciljeva ovoga rada bio je i poboljšati stabilnost predviđanja kao i generalizacijsku sposobnost modela kada je u pitanju polunadzirano učenje.

Za ostvarenje ovih ciljeva u ovom se radu predlaže primjena metode usrednjenog učitelja koja je ponajprije u području računalnog vida polučila dobre rezultate. S obzirom na postignute rezultate u ovome radu, može se reći da navedena metoda ima potencijala uz dobru optimizaciju polučiti zadovoljavajuće rezultate, međutim manjak podataka je definitivno predstavljao ogroman izazov u okviru ovoga rada. Rezultati dobiveni ovom metodom uspoređeni su s rezultatima klasifikatora MLP i GRU . Zadnji je navedeni model pružao nešto slabije performanse s prosječnom točnošću klasifikacije od 61.02% ze GEO seriju GSE2034, odnosno 79.1% za GSE25066 seriju, Rezultati modela višeslojnog perceptrona bili su konzistentniji s prosječnom točnošću od 63.46%, odnosno 80.38%. Ovaj je model korišten i kao model studenta i učitelja u MT metodi koja je uz manji broj korištenih podataka za učenje na skupu GSE2034 rezultirala povećanjem točnosti (65.18% za studenta te 65.52% za model učitelja). Na skupu GSE25066

klasifikacijski su rezultati bili vrlo slični običnom MLP modelu (prosječna točnost tek neznatno slabija). Predloženi pristup korištenja jednog od skupova podataka kao izvora neoznačenih nije rezultirao poboljšanjem točnosti. Također je na ovakvom skupu bilo bitno maksimizirati odziv što zbog uravnoteženosti klasa u skupovima nije bilo teško postići (odziv je za sve modele bio visok u odnosu na ostale metrike te je primjerice za MLP model iznosio 77.8% za GSE2034 i 90.37% za GSE25066). Također treba napomenuti kako, pogotovo u slučajevima kada su podaci ograničeni, nema univerzalnog rješenja koje će uvijek predviđati točno, ali primjena predložene metode može u odgovarajućim uvjetima na efikasan način iskoristiti neoznačene podatke.

Bibliografija

[1] Antti, Tarvainen., Harri, Valpola. (2017). Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results. 30:1195-1204.

[2] Wang Z, Gerstein M, Snyder M. RNK-Seq: a revolutionary tool for transcriptomics. Nat Rev Genet. 2009 Jan;10(1):57-63. doi: 10.1038/nrg2484. PMID: 19015660; PMCID: PMC2949280.

[3] Lowe R, Shirley N, Bleackley M, Dolan S, Shafee T. Transcriptomics technologies. PLoS Comput Biol. 2017 May 18;13(5):e1005457. doi: 10.1371/journal.pcbi.1005457. PMID: 28545146; PMCID: PMC5436640.

[4] https://www.cancer.gov/ccg/research/genome-sequencing/tcga/history, posjećeno 28. srpnja 2024.

[5] https://www.ncbi.nlm.nih.gov/geo/, posjećeno 28. srpnja 2024.

[6] The 1000 Genomes Project Consortium. A map of human genome variation from population-scale sequencing. Nature 467, 1061–1073 (2010). https://doi.org/10.1038/nature09534

[7] Pankaj Barah, Dhruba Kumar Bhattacharyya, Jugal Kumar Kalita, "*Gene Expression Data Analysis A Statistical and Machine Learning Perspective*", 2021.

[8] Tom Mitchell, "*Machine Learning*", 1997.

[9] Aston Zhang, Zachary C. Lipton, Mu Li, "*Dive Into Deep Learning*", 2023.

[10] Tanya Barrett, Dennis B. Troup, Stephen E. Wilhite, Pierre Ledoux, Dmitry Rudnev, Carlos Evangelista, Irene F. Kim, Alexandra Soboleva, Maxim Tomashevsky, Ron Edgar, NCBI GEO: mining tens of millions of expression profiles—database and tools update, Nucleic Acids Research, 2007

[11] J. L. Elman, "Finding structure in time," Cognitive Science, vol. 14, no. 2, pp. 179–211, 1990.

[12] Haznedar, Bulent; Arslan, Mustafa Turan; KALINLI, Adem (2017), "Microarray Gene Expression Cancer Data", Mendeley Data, V4, doi: 10.17632/ynp2tst2hh.4

[13] Haozhe Xie, Jie Li, Qiaosheng Zhang, Yadong Wang, Comparison among dimensionality reduction techniques based on Random Projection for cancer classification, Computational Biology and Chemistry, 2016.

[14] Wang Y, Klijn JG, Zhang Y, Sieuwerts AM, Look MP, Yang F, Talantov D, Timmermans M, Meijer-van Gelder ME, Yu J, Jatkoe T, Berns EM, Atkins D, Foekens JA. Gene-expression profiles to predict distant metastasis of lymph-node-negative primary breast cancer. Lancet. 2005.

[15] Hatzis C, Pusztai L, Valero V, Booser DJ, Esserman L, Lluch A, Vidaurre T, Holmes F, Souchon E, Wang H, Martin M, Cotrina J, Gomez H, Hubbard R, Chacón JI, Ferrer-Lozano J, Dyer R, Buxton M, Gong Y, Wu Y, Ibrahim N, Andreopoulou E, Ueno NT, Hunt K, Yang W, Nazario A, DeMichele A, O'Shaughnessy J, Hortobagyi GN, Symmans WF. A genomic predictor of response and survival following taxane-anthracycline chemotherapy for invasive breast cancer. JAMA. 2011.

[16] Shi M, Zhang B. Semi-supervised learning improves gene expression-based prediction of cancer recurrence. Bioinformatics. 2011 Nov 1;27(21):3017-23. doi: 10.1093/bioinformatics/btr502. Epub 2011 Sep 4. PMID: 21893520; PMCID: PMC3198572.

[17] Babichev S, Liakh I, Kalinina I. Applying a Recurrent Neural Network-Based Deep Learning Model for Gene Expression Data Classification. Applied Sciences. 2023; 13(21):11823. https://doi.org/10.3390/app132111823

[18] Alharbi, F.; Vakanski, A. Machine Learning Methods for Cancer Classification Using Gene Expression Data: A Review. Bioengineering 2023, 10, 173. https://doi.org/10.3390/bioengineering10020173

[19] Golcuk, Guray & Tuncel, Mustafa & Canakoglu, Arif. (2018). Exploiting Ladder Networks for Gene Expression Classification. 10.1007/978-3-319-78723-723.

[20] Naik, Sharath. (2023). Microarray Cancer Data Classification using Deep Learning-Methods. Tuijin Jishu/Journal of Propulsion Technology. 44. 3544-3567. 10.52783/tj-jpt.v44.i5.3312.

[21] Tu, Y., et al. "Quantitative Noise Analysis for Gene Expression Microarray Experiments." Proceedings of the National Academy of Sciences - PNAS, vol. 99, no. 22, 2002, pp. 14031–36, https://doi.org/10.1073/pnas.222164199.

[22] Laine, Samuli and Timo Aila. "Temporal Ensembling for Semi-Supervised Learning." ArXiv abs/1610.02242 (2016): n. pag.

[23] "Common Loss functions in machine learning", Medium `https://towardsdatascience.com/common-loss-functions-in-machine-learning-46af0ffc4d23`, posjećeno 17. lipnja 2024.

[24] Kingma, Diederik & Ba, Jimmy. (2014). Adam: A Method for Stochastic Optimization. International Conference on Learning Representations.

[25] Cheadle C, Vawter MP, Freed WJ, Becker KG. Analysis of microarray data using Z score transformation. J Mol Diagn. 2003 May;5(2):73-81. doi: 10.1016/S1525-1578(10)60455-2. PMID: 12707371; PMCID: PMC1907322.

[26] Bobak CA, McDonnell L, Nemesure MD, Lin J, Hill JE. Assessment of Imputation Methods for Missing Gene Expression Data in Meta-Analysis of Distinct Cohorts of Tuberculosis Patients. Pac Symp Biocomput. 2020;25:307-318. PMID: 31797606.

[27] Neural networks and deep learning `http://neuralnetworksanddeeplearning.com/`, posjećeno 26. lipnja 2024.

[28] https://bksaini078.medium.com/machine-learning-understanding-mean-teacher-model-ffb8d07819a4, posjećeno 23. lipnja 2024.

[29] Cheadle C, Vawter MP, Freed WJ, Becker KG. Analysis of microarray data using Z score transformation. J Mol Diagn. 2003 May;5(2):73-81. doi: 10.1016/S1525-1578(10)60455-2. PMID: 12707371; PMCID: PMC1907322.

[30] Montesinos-López, O.A., Montesinos-López, A., Pérez-Rodríguez, P. et al. A review of deep learning applications for genomic selection. BMC Genomics 22, 19 (2021). https://doi.org/10.1186/s12864-020-07319-x

[31] Basavegowda, Hema Shekar, and Guesh Dagnew. "Deep learning approach for microarray cancer data classification." CAAI Transactions on Intelligence Technology 5.1 (2020)

[32] https://medium.com/nerd-for-tech/removing-constant-variables-feature-selection-463e2d6a30d9, posjećeno 25. rujna 2024.

[33] https://bitesizebio.com/7206/introduction-to-dna-microarrays/, posjećeno 3. studenog 2024.

[34] https://www.fer.unizg.hr/_download/repository/SU1-2022-P06-LogistickaRegresija.pdf, posjećeno 3. studenog 2024.

[35] https://www.genetika.biol.pmf.hr/docs/sadrzaj/jedanaesto-poglavlje/transkripcija/, posjećeno 3. studenog 2024.

[36] https://towardsdatascience.com/what-makes-logistic-regression-a-classification-algorithm-35018497b63f, posjećeno 3. studenog 2024.

Sažetak

Kao jedan od problema u domeni genomike u kojem je strojno učenje često primijenjeno jest klasifikacija bolesti temeljena na podacima genske ekspresije. Ovaj diplomski rad bavi se predviđanjem raka uz pomoć takvih podataka. Na samom početku rada čitatelj je uveden u tematiku te je objašnjena biološka pozadina potrebna za razumijevanje ovog rada. Nadalje, istraženi su i navedeni javno dostupni resursi podataka genske ekspresije. Kao praktični dio ovoga rada razvijena su i optimizirana tri klasifikacijska modela dubokog učenja korištenjem PyTorch knjižnice. Pri kraju rada izvršena je detaljna analiza performansi modela.

Ključne riječi **— genska ekspresija, rak dojke, duboko učenje, neuronske mreže, Metoda usrednjenog učitelja**

Abstract

One of the key challenges in the field of genomics, where machine learning is frequently applied, is disease classification given the gene expression data. This thesis aims to predict the recurrence of breast cancer using such data. The work begins by introducing the reader to the subject, providing the biological background necessary to understand the scope of this study. Furthermore, publicly available gene expression data resources are explored and listed. As the practical component of this thesis, three deep learning classification models were developed and optimized using the PyTorch library. Toward the end of this thesis, a detailed performance analysis of the models is conducted.

Keywords **— gene expression, breast cancer, deep learning, neural networks, Mean Teacher**