Predviđanje genskog izražaja modelima dubokih neuronskih mreža

Dokić, Mateo

SVEUČILIŠTE U RIJECI
TEHNIČKI FAKULTET
Sveučilišni prijediplomski studij računarstva

Završni rad

PREDVIĐANJE GENSKOG IZRAŽAJA MODELIMA DUBOKIH NEURONSKIH MREŽA

Rijeka, rujan 2024.

Mateo Dokić
0069092165

SVEUČILIŠTE U RIJECI
TEHNIČKI FAKULTET
Sveučilišni prijediplomski studij računarstva

Završni rad

PREDVIĐANJE GENSKOG IZRAŽAJA MODELIMA DUBOKIH NEURONSKIH MREŽA

Mentor: Prof. dr. sc. Goran Mauša

Rijeka, rujan 2024.

Mateo Dokić
0069092165

SVEUČILIŠTE U RIJECI
TEHNIČKI FAKULTET
POVJERENSTVO ZA ZAVRŠNE ISPITE

Rijeka, 17.03.2024.

Zavod: Zavod za računarstvo

Predmet: Programsko inženjerstvo

ZADATAK ZA ZAVRŠNI RAD

Pristupnik: **Mateo Dokić (0069092165)**

Studij: Sveučilišni prijediplomski studij računarstva (1035)

Zadatak: **Predviđanje genskog izražaja modelima dubokih neuronskih mreža / Prediction of gene expression by deep neural network models**

Opis zadatka:

Istražiti i usporediti različite modele dubokih neuronskih mreža u kontekstu predviđanja genskog izražaja. Analizirati način rada postojećih modela poput Enformera koji prilagođava arhitekuru transformera za transkripcijske ojačivače. Pregledati relevantne modele iz literature, pripremiti podatke za učenje i testiranje modela, implementirati odabrane modele te vrednovati njihove performanse. Komentirati prednosti i ograničenja odabranih modela u kontekstu analize genoma.

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
završni ispit:
prof. dr. sc. Miroslav Joler

Mentor:
izv. prof. Goran Mauša

Izjava o samostalnoj izradi rada

Izjavljujem da sam samostalno izradio ovaj rad.

Rijeka, rujan 2024.

_____

Ime Prezime

Zahvala

Zahvaljujem profesoru i mentoru Goranu Mauši na podršci tijekom pisanja ovoga rada, korisnim raspravama, kvalitetnim savjetima i konstruktivnim kritikama.

Sadržaj

Popis slika

Poglavlje 1

UVOD

Genski izražaj, ključan element u molekularnoj biologiji, odnosi se na procese kojima stanice transkriptiraju genetske informacije iz DNA u RNA, što u konačnici vodi do sinteze proteina u organizmu. Razumijevanje ovih procesa omogućuje ne samo dublji uvid u osnovne mehanizme života, već i razvoj novih terapija za bolesti kao što su rak i genetski poremećaji. S obzirom na kompleksnost i obim podataka uključenih u analizu genskog izražaja, bioinformatici su potrebne sofisticirane metode za obradu i interpretaciju ove vrste podataka.

U posljednjih nekoliko godina, modeli dubokih neuronskih mreža postali su izuzetno važni u bioinformatici zahvaljujući svojoj sposobnosti učenja složenih obrazaca iz velikih količina podataka. Ove metode omogućuju ne samo bolje razumijevanje bioloških procesa na molekularnoj razini, već i predviđanje promjena u genskom izražaju pod različitim uvjetima, što ima ključnu ulogu u personaliziranoj medicini i razvoju lijekova.

Ovaj završni rad izrađen je u okviru projekta pod naslovom Primjena umjetne inteligencije u predviđanju genskog izražaja (oznaka: UNIRI-INOVA-3-23-1) s ciljem istraživanja, implementacije i usporedbe naprednih modela dubokih neuronskih mreža u kontekstu predviđanja genskog izražaja. Fokus rada bit će na modelima Enformer [1] i CRMnet [2], koji predstavljaju najnovije pristupe u ovom području. Enformer, adaptacija arhitekture transformera specijalizirana za analizu transkripcijskih ojačivača, i CRMnet, koji koristi konvolucijske neuronske mreže za mapiranje regija bogatih cis-regulatornim motivima, su dva primjera kako moderna istraživanja pristupaju problemu predviđanja genskog izražaja. Svaki od

ovih modela ima svoje specifične prednosti i ograničenja, koje će biti detaljno analizirane kroz praktičnu implementaciju i usporedbu njihovih performansi na stvarnim skupovima podataka. Skupovi podataka uključuju milijune nasumično uzorkovanih promotorskih DNA sekvenci i njihovih izmjerenih razina genskog izražaja u kvascu *Saccharomyces cerevisiae*.

CRMnet je nova arhitektura neuronske mreže koja precizno predviđa razine genske ekspresije uzrokovane promotorima kvasca. Arhitektura modela je inspirirana biološkim saznanjima da promotor sekvence sadrže više susjednih TFBS (transkripcijskih faktorskih veznih mjesta) motiva koji zajedno koordinirano reguliraju gensku ekspresiju. Prema istraživanju, CRMnet pokazuje poboljšane performanse u odnosu na prethodne modele, kao što je model temeljen na transformatorima (Vaishnav et al., 2022). Ovo poboljšanje je rezultat korištenja kombinacije konvolucijskih neuronskih mreža i dodatnih transformator stupnjeva, vođeni rezultatima treniranja i testiranja na velikim visokoprotočnim skupovima podataka.

Model Enformer razvio je tim organizacije Google Deepmind pod vodstvom Žige Avseca [1], dok je model CRMnet razvijen od strane istraživača predvođenih Alexandrom Vu [2]. Enformer predstavlja značajan napredak u točnosti predviđanja genske ekspresije iz DNA sekvenci korištenjem duboke neuronske mreže koja može integrirati informacije iz dugometražnih interakcija (do 100 kb udaljenosti) u genomu. Ovaj napredak omogućava točnije predikcije utjecaja varijanti na genski izražaj za prirodne genetske varijante. Enformer je sposoban izravno iz DNA sekvence predvidjeti interakcije ojačivača i promotora, konkurentno metodama koje koriste izravne eksperimentalne podatke kao ulaz. Ova poboljšanja omogućuju učinkovitije fino mapiranje asocijacija ljudskih bolesti i pružaju okvir za interpretaciju evolucije cis-regulatora. Model Enformer ne samo da poboljšava predikcije genske ekspresije, već i omogućuje bolje razumijevanje kako nekodirajuća DNA određuje gensku ekspresiju u različitim tipovima stanica, što je ključni problem u ljudskoj genetici s važnim primjenama u medicini i biologiji.

Kroz ovaj rad, posebna pažnja bit će usmjerena na vrednovanje točnosti, učinkovitosti i praktične primjene ovih modela u predviđanju genskog izražaja. Diskusija će uključivati analizu kako ovi modeli doprinose unapređenju personalizirane medicine i koje su mogućnosti za daljnji razvoj ovih tehnologija u svjetlu trenutnih izazova i ograničenja.

Poglavlje 2

PREDVIĐANJE GENSKOG IZRAŽAJA

Gensko izražavanje je osnovni molekularni proces koji omogućuje da se genetička informacija zapisana u DNA pretoči u funkcionalne proteinske i ne-proteinske proizvode. Ovaj proces je temelj za razumijevanje kako geni kontroliraju fiziološke i biokemijske funkcije unutar organizma.

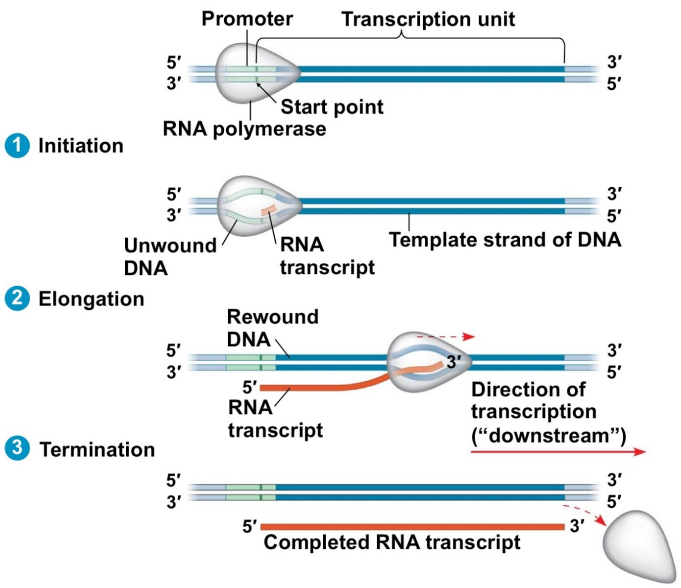
2.1 Faze i mehanizmi genskog izražaja

Transkripcija

Transkripcija je prva faza u procesu genskog izražaja gdje se dio DNA sekvence prepiše u glasnički RNA (mRNA) (Slika 2.1). Ovaj proces započinje kada se transkripcijski faktori vežu za specifične DNA sekvence poznate kao promotorske regije koje su locirane neposredno uz gen. Ova vezanja iniciraju sastavljanje kompleksa RNA polimeraze na promotoru, što dovodi do lokalnog "otvaranja" dvostrukog lanca DNA kako bi se očitala njegova kodirajuća sekvencija. RNA polimeraza zatim sintetizira jednolančanu RNA molekulu koja je komplementarna matičnoj DNA sekvenci, čime nastaje primarna mRNA (pre-mRNA).

Nakon što se sintetizira, pre-mRNA mora proći kroz procese *cappinga*, spajanja i poliadenilacije prije nego što postane zrela mRNA, koja je spremna za translaciju. Capping uključuje dodavanje modificirane guanin nukleotide na 5' kraj pre-mRNA, što štiti mRNA od degradacije i pomaže u inicijaciji translacije. *Splicing* je proces u kojem se introni (nepotrebne sekvence) uklanjaju, a eksoni

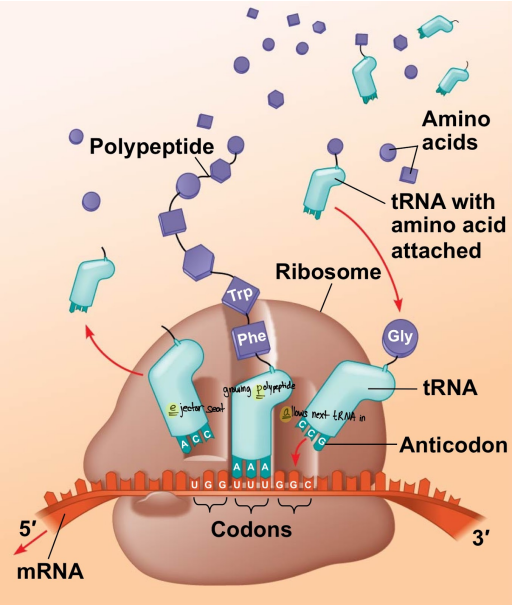
(kodirajuće sekvence) spajaju kako bi formirali kontinuiranu kodirajuću sekvencu. Na kraju, polyadenilacija dodaje rep od adenin nukleotida na 3' kraj mRNA, što također pomaže u stabilnosti i izvozu mRNA iz jezgre u citoplazmu.



Slika 2.1 Proces transkripcije (preuzeto iz [3])

Translacija

Translacija, prikazana na slici 2.2, je proces u kojem ribosomi čitaju sekvencu mRNA i sintetiziraju proteine prema kodirajućim uputama. Ribosomi se vežu na 5' kraj mRNA i kreću se duž njezine sekvence, čitajući kodone (tri nukleotida) koji specificiraju aminokiseline. Molekule transportne RNA (tRNA) prenose odgovarajuće aminokiseline prema kodonima na mRNA, gdje ribosomi kataliziraju formiranje peptidnih veza između aminokiselina, stvarajući polipeptidni lanac koji se preklapa u funkcionalni protein.



Slika 2.2 Proces translacije (preuzeto iz [3])

Regulacija genskog izražaja

Regulacija genskog izražaja može se odvijati na različitim razinama, uključujući transkripciju, post-transkripcijske modifikacije, translaciju i post-translacijske modifikacije. Na razini transkripcije, faktori okoline i signalne molekule mogu utjecati na aktivnost promotora i dostupnost transkripcijskih faktora, čime direktno kontroliraju sintezu mRNA. Post-transkripcijske modifikacije kao što su alternativni splicing i RNA interference također igraju ključnu ulogu u kontroli koje mRNA molekule će biti prevedene u proteine. Na razini translacije, različiti mehanizmi mogu odrediti koliko često i učinkovito ribosomi mogu pristupiti mRNA za sintezu proteina. Post-translacijske modifikacije proteina, kao što su fosforilacija ili glikozilacija, dalje mogu modificirati aktivnost, stabilnost ili lokaciju proteina unutar stanice.

2.2 Metode analize genskog izražaja

Za detaljno razumijevanje genskog izražaja koriste se različite eksperimentalne metode koje pružaju uvid u različite aspekte regulacije i funkcije gena [4]. Osim tehnika kao što su Northern blotting, RT-PCR, DNA microarray, i RNA-seq, važne su i metode kao što su DNAse-seq, CAGE i ChIP-seq. Svaka od ovih metoda ima specifičnu primjenu i pruža jedinstvene informacije o genskom izražaju i regulaciji.

Northern blotting

Northern blotting je klasična metoda za detekciju i kvantifikaciju specifičnih RNA molekula unutar uzorka [5]. Proces uključuje separaciju RNA po veličini pomoću gel elektroforeze, a zatim transferiranje RNA iz gela na membranu, koja se naknadno izlaže specifičnim označenim sondama koje hibridiziraju s ciljanom RNA. Nakon hibridizacije, signal se vizualizira koristeći radioaktivne ili fluorescentne etikete na sondama. Northern blotting omogućuje ne samo detekciju prisutnosti i količine određene mRNA, već i pruža informacije o veličini transkripta, što može ukazivati na alternativni splicing ili druge post-transkripcijske modifikacije. Rezultati se analiziraju usporedbom intenziteta signala na blotu, što omogućuje kvantitativnu analizu izražaja gena.

RT-PCR

Reverse Transcription Polymerase Chain Reaction (RT-PCR) je tehnika koja omogućava kvantitativno mjerenje razine mRNA [6]. Prvi korak uključuje stvaranje komplementarne DNA (cDNA) iz mRNA pomoću enzima reverzne transkriptaze. Nakon toga se provodi PCR amplifikacija specifičnih segmenata cDNA koristeći specifične *primer*-e. RT-PCR može biti kvantitativan qRT-PCR ako se koristi za mjerenje količine početne RNA, što se obično postiže detekcijom fluorescentnih signala tijekom PCR ciklusa. Rezultati qRT-PCR-a omogućuju vrlo precizne i osjetljive mjere izražaja gena, a interpretacija se temelji na vrijednosti praga ciklusa (Ct), gdje niže Ct vrijednosti ukazuju na višu početnu koncentraciju mRNA.

DNA microarray

DNA microarray je tehnologija koja omogućuje istovremeno mjerenje izražaja tisuća gena [7]. Tehnika uključuje hibridizaciju označenih cDNA molekula na mikročipove koji sadrže tisuće specifičnih oligonukleotidnih sondi, fiksiranih na čvrstu površinu. Hibridizacija označenih cDNA s komplementarnim sondama na čipu rezultira fluorescentnim signalima koji se čitaju i analiziraju. Intenzitet signala odražava razinu izražaja svakog gena u uzorku. Analiza podataka obično uključuje normalizaciju signala i statističku obradu za identifikaciju značajnih promjena u izražaju gena između različitih uzoraka ili tretmana.

RNA-seq

RNA sequencing (RNA-seq) je tehnika sekvenciranja nove generacije koja omogućuje visoko-protočno kvantitativno profiliranje cijelog transkriptoma [8]. Uključuje konverziju izolirane RNA u knjižnicu cDNA, koja se zatim sekvencira pomoću platformi za sekvenciranje visokog protoka. RNA-seq pruža kompletne transkriptome, uključujući nekodirajuće RNA i alternativno splajcane varijante, s visokom rezolucijom i preciznošću. Analiza RNA-seq podataka uključuje mapiranje čitanja na referentni genom, kvantifikaciju izražaja gena, i identifikaciju diferencijalno izraženih gena. Rezultati se često vizualiziraju u obliku točkastih dijagrama (engl. scatter plots) ili toplinskih karata (engl. heatmaps), što omogućuje lako prepoznavanje uzoraka izražaja i biološki relevantnih uvida.

DNAse-seq

DNAse I hypersensitive sites sequencing (DNAse-seq) je tehnika koja se koristi za identifikaciju regija DNA koje su osjetljive na razgradnju DNase I enzimom, što ukazuje na njihovu dostupnost i potencijalnu regulacijsku ulogu u genskoj ekspresiji [9]. Tijekom DNAse-seq eksperimenta, DNAse I tretira se kromatinom kako bi se prepoznale i izolirale regije koje su slobodne od pakovanja histona i stoga dostupne transkripcijskim faktorima. Nakon tretmana, DNAse I osjetljive regije se sekveniraju kako bi se stvorila mapa dostupnih regija u genomu. Analiza rezultata DNAse-seq omogućuje identifikaciju promotorskih regija, enhancera i

drugih regulatornih elemenata koji su aktivni u određenom staničnom tipu ili pod određenim uvjetima. Rezultati se obično vizualiziraju u amplitudama na genomskim mapama, gdje više amplitude predstavljaju regije s većom osjetljivošću na DNAse I.

CAGE

Cap Analysis of Gene Expression (CAGE) je metoda koja omogućuje precizno mapiranje transkripcijskih početnih mjesta (TSS) i kvantifikaciju genskog izražaja [10]. Tehnika se temelji na sekveniranju 5' krajeva mRNA molekula koji su označeni kapicom, što omogućuje identifikaciju početnih točaka transkripcije za pojedine gene. U CAGE eksperimentima, mRNA se izolira iz stanica, a zatim se koristi obrnuta transkripcija i adaptor ligation za pripremu cDNA za sekvenciranje. Analiza CAGE podataka omogućuje detekciju promjena u TSS korištenju između različitih stanica ili uvjeta, pružajući uvid u alternativne promotorne upotrebe i regulaciju na razini transkripcije. Rezultati CAGE-a se obično prikazuju kao grafovi gustoće TSS-ova duž genoma, gdje su više gustoće povezane s većom aktivnošću promotora.

ChIP-seq

Chromatin Immunoprecipitation sequencing (ChIP-seq) je moćna tehnika za proučavanje interakcija proteina s DNA, posebice za mapiranje lokacija vezanja transkripcijskih faktora i modificiranih histona u cijelom genomu. U ChIP-seq eksperimentu, prvo se korištenjem formaldehida kemijski povezuju proteini i DNA unutar stanica, a zatim se kromatin fragmentira i imunoprecipitira pomoću specifičnih antitijela protiv ciljanog proteina. DNA fragmenti povezani s precipitiranim proteinima se zatim izoliraju i sekveniraju. Analizom ChIP-seq podataka moguće je identificirati specifične regije u genomu koje su ciljane od strane određenih transkripcijskih faktora ili koje nose specifične post-translacijske modifikacije histona. Rezultati se obično prikazuju kao amplitude na genomskim lokacijama, gdje svaka amplituda predstavlja regiju povećanog vezanja proteina.

2.3 Uloga i važnost predviđanja genskog izražaja

Predviđanje genskog izražaja ima ključnu ulogu u biomedicinskim znanostima, jer pruža uvide koji su esencijalni za razumijevanje bolesti i razvoj terapeutika. Sposobnost predviđanja kako će geni reagirati na određene tretmane ili promjene u okolišu može voditi ka personaliziranoj medicini, gdje se tretmani mogu prilagoditi genetskom profilu pojedinca. Također, predviđanje genskog izražaja omogućuje identifikaciju novih terapeutskih meta i biomarkera bolesti. Na primjer, u onkologiji, razumijevanje genskog izražaja tumora može pomoći u identifikaciji ključnih gena koji potiču rast tumora i otpornost na lijekove, što može voditi ka razvoju ciljanih terapija koje specifično ciljaju te gene [11]. U genetskim bolestima, predviđanje učinka mutacija na gensko izražavanje može pomoći u dijagnozi i planiranju tretmana. Ovaj pristup predstavlja rješenje za problem generičkih lijekova koji se temelje na principu 'jedan lijek za sve' zbog kojeg dolazi do manje učinkovitosti rezultata i većeg broja nuspojava.

Predviđanje genskog izražaja igra ključnu ulogu i u otkrivanju novih biomarkera za bolesti [12]. Biomarkeri su biološki pokazatelji bolesti ili terapijskog odgovora koji se mogu koristiti za rano otkrivanje bolesti, praćenje napretka bolesti, ili predviđanje odgovora na liječenje. Analiza genskog izražaja može otkriti promjene u ekspresiji određenih gena koje su specifične za određene bolesti, što može voditi do razvoja novih dijagnostičkih testova i terapija.

U farmaceutskoj industriji, predviđanje genskog izražaja koristi se za razvoj novih lijekova [13]. Identificiranje ključnih gena koji sudjeluju u patogenezi bolesti može pomoći u ciljanju tih gena novim terapijskim agentima. Na primjer, ako se otkrije da određeni gen igra ključnu ulogu u razvoju otpornosti na lijekove, može se razviti novi lijek koji specifično cilja taj gen kako bi se poboljšala efikasnost postojećih terapija.

U poljoprivredi, tehnike predviđanja genskog izražaja koriste se za poboljšanje otpornosti i produktivnosti biljaka [14]. Na primjer, razumijevanje kako biljke izražavaju određene gene u odgovoru na stresne uvjete, poput suše ili visokih temperatura, može voditi razvoju genetski modificiranih biljaka koje su bolje prilagođene tim uvjetima. Ovo može rezultirati biljkama koje proizvode više hrane ili koje bolje podnose promjenjive klimatske uvjete, čime se osigurava veća prehrambena

sigurnost.

2.4 Duboke neuronske mreže

DNN predstavljaju naprednu kategoriju algoritama strojnog učenja koji su inspirirani strukturom i funkcijom ljudskog mozga. Ove mreže se sastoje od više slojeva umjetnih neurona, odnosno čvorova, koji obrađuju ulazne podatke kroz složene transformacije kako bi generirali željeni izlaz. Duboke neuronske mreže su postale izuzetno popularne u širokom rasponu primjena zbog svoje sposobnosti prepoznavanja i učenja iz složenih uzoraka u velikim količinama podataka [15].

2.4.1 Arhitektura dubokih neuronskih mreža

Na slici 2.3 je prikazana temeljna arhitektura dubokih neuronskih mreža koja uključuje tri glavna tipa slojeva: ulazni sloj, skriveni slojevi i izlazni sloj. Ulazni sloj prima sirove ili predobrađene podatke, skriveni slojevi obrađuju te podatke kroz nelinearne transformacije, a izlazni sloj generira konačan rezultat predstavljem numeričkom varijablom koja može biti interpretirana kao vjerojatnost (u slučaju klasifikacije) ili veličina koju predviđamo (u slučaju regresije).

Ulazni sloj

Ulazni sloj duboke neuronske mreže primarno služi kao pristupna točka za ulazne podatke. Ovi podaci mogu biti u različitim formatima, ovisno o specifičnoj primjeni, poput slika, zvuka ili tabličnih podataka.

Postoje značajne razlike u konfiguraciji ulaznog sloja ovisno o vrsti podataka koja se koristi. Za slike, ulazni sloj obično ima dimenzije koje odgovaraju veličini slike. Na primjer, za slike dimenzija 28x28 piksela s jednim kanalom (npr. crnobijele slike), ulazni sloj imat će 784 neurona (28x28), dok će za slike s tri kanala (RGB slike), ulazni sloj imati 28x28x3 neurona.

Za zvukovne podatke, ulazni sloj može biti konfiguriran tako da prihvaća vremenske serije amplituda zvučnog signala, gdje svaki neuron odgovara jednoj točki u vremenskoj seriji. U nekim slučajevima, zvukovni podaci se prvo transformiraju

u spektrograme prije nego što se unesu u mrežu, pri čemu ulazni sloj ima dimenzije koje odgovaraju dimenzijama spektrograma.

Tablični podaci, koji se često pojavljuju u poslovnim i znanstvenim aplikacijama, obično imaju ulazni sloj čiji broj neurona odgovara broju značajki (featurea) u tablici. Na primjer, za tablične podatke s 10 značajki, ulazni sloj će imati 10 neurona.

Svaka vrsta podataka zahtijeva specifičnu prilagodbu ulaznog sloja kako bi se osigurala učinkovita obrada i ekstrakcija značajki, što je ključno za postizanje visokih performansi modela u različitim primjenama.

Skriveni slojevi

Skriveni slojevi čine jezgru dubokih neuronskih mreža. Broj skrivenih slojeva i broj neurona u svakom sloju mogu se značajno razlikovati i određuju se na temelju specifičnosti problema koji se rješava. Neuroni u skrivenim slojevima koriste težinske koeficijente i pristranosti, koje se prilagođavaju tijekom procesa učenja, za transformaciju ulaznih podataka u izlaz koji služi kao ulaz za sljedeći sloj. Ova svojstva omogućuju DNN-ima modeliranje složene funkcije.

Izlazni sloj

Izlazni sloj duboke neuronske mreže odgovoran je za generiranje konačnog rezultata. Broj neurona u izlaznom sloju ovisi o vrsti zadatka — na primjer, u zadacima klasifikacije, broj neurona često odgovara broju klasa koje model treba prepoznati.

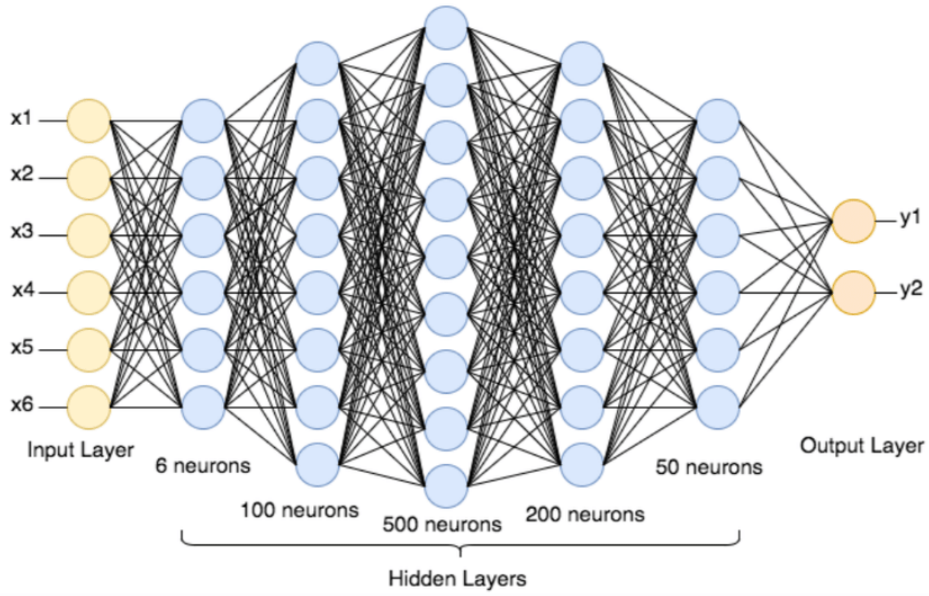
Kod različitih zadataka, izlazni sloj koristi različite aktivacijske funkcije kako bi prilagodio izlazne vrijednosti odgovarajućem rasponu. Jedna od najčešće korištenih funkcija je sigmoidna funkcija, koja se koristi za binarnu klasifikaciju gdje je potrebno odrediti je li primjer pripadajući jednoj od dvije klase. Sigmoidna funkcija transformira izlazne vrijednosti u raspon od 0 do 1, omogućujući interpretaciju rezultata kao vjerojatnosti pripadnosti određenoj klasi. Matematički, sigmoidna funkcija se definira kao $\sigma(x) = \frac{1}{1+e^{-x}}$, gdje $\sigma(x)$ predstavlja sigmoidnu funkciju.

Za klasifikaciju u veći broj disjunktnih razreda koristi se softmax funkcija. Softmax funkcija normalizira izlazne vrijednosti tako da njihov zbroj bude 1, pre-

tvarajući ih u vjerojatnosti pripadnosti svakoj klasi. Svaki neuron u izlaznom sloju predstavlja jednu klasu, a izlazna vrijednost neurona interpretira se kao vjerojatnost da ulazni podatak pripada toj klasi. Funkcija softmax se definira kao $\text{softmax}(x_i) = \frac{e^{x_i}}{\sum_{j=1}^{k} e^{x_j}}$, gdje x_i predstavlja ulaznu vrijednost neurona, a k ukupan broj klasa.

U regresijskim zadacima, gdje je cilj predviđanje kontinuiranih vrijednosti, koristi se linearna funkcija. Linearna funkcija omogućuje izlaznim vrijednostima zadržavanje kontinuiranog raspona, što je ključno za precizne regresijske modele. Matematički, linearna funkcija se definira kao $f(x) = x$.

Pravilnim odabirom aktivacijske funkcije u izlaznom sloju, model može učinkovito prilagoditi svoje izlazne vrijednosti specifičnim zahtjevima zadatka, bilo da se radi o klasifikaciji ili regresiji.



Slika 2.3 Arhitektura DNN-a sa 2 izlaza, 6 ulaza i 5 skrivenih slojeva sa sveukupno 856 čvorova (preuzeto iz [16])

2.4.2 Mehanizmi učenja

Duboke neuronske mreže koriste algoritme učenja kako bi optimizirale svoje težinske koeficijente. Najčešći algoritam učenja za DNN je unazadna propagacija (engl. backpropagation) u kombinaciji s optimizacijskim algoritmima kao što su Stohastički gradijentni spust (SGD).

Unazadna propagacija

Unazadna propagacija je metoda kojom se greška izračunata na izlaznom sloju propagira unazad kroz mrežu, omogućujući prilagodbu težina i pristranosti kako bi se minimizirala greška. Tijekom ovog procesa, izvodi funkcije gubitka se koriste za prilagodbu parametara mreže u svrhu optimizacije performansi modela.

Optimizacija

Optimizacijski algoritmi kao što su SGD, Adam i RMSprop koriste se za efikasno ažuriranje težina u dubokim neuronskim mrežama tijekom treninga. Ovi algoritmi pomažu u minimiziranju funkcije gubitka kroz iterativno ažuriranje težina na temelju gradijenta funkcije gubitka.

2.5 Primjena dubokih neuronskih mreža u bioinformatici

DNN imaju široku primjenu u polju bioinformatike, posebno u analizi i interpretaciji bioloških podataka koji se generiraju u velikim količinama zbog napredovanja tehnologija visokog protoka. Njihova sposobnost učenja izuzetno složene obrasce čini ih idealnim za zadatke kao što su predviđanje strukture proteina [17], analiza genetskih varijacija [18], i posebno predviđanje genskog izražaja [1].

Predviđanje genskog izražaja jedan je od ključnih izazova u bioinformatici, gdje duboke neuronske mreže igraju vitalnu ulogu. Gensko izražavanje, proces u kojem se informacije iz DNA transkribiraju u mRNA i prevedu u proteine, podložno je složenoj regulaciji i varijacijama koje mogu biti uzrokovane različitim genetskim i okolišnim faktorima. Duboke neuronske mreže omogućavaju modeliranje tih složenih interakcija kroz nekoliko pristupa:

2.5.1 Modeliranje transkripcijske aktivnosti

DNN se koriste za predviđanje aktivnosti promotorskih i enhancerskih regija u genomu, koje igraju ključne uloge u regulaciji transkripcije. Modeli kao što je DeepSEA koriste se za predviđanje vezanja transkripcijskih faktora i epigenetskih markera u tim regijama, što može ukazati na njihovu aktivnost.

2.5.2 Predviđanje utjecaja mutacija

Mutacije u genomu mogu značajno utjecati na izražavanje gena. DNN se koriste za analizu kako specifične mutacije mogu promijeniti gensko izražavanje i potencijalno dovesti do bolesti. Ovi modeli mogu identificirati patogene mutacije analizirajući promjene u obrascima vezanja transkripcijskih faktora ili promjene u strukturalnoj dostupnosti DNA.

2.5.3 Integracija više vrsta podataka

U modernoj bioinformatici, često je potrebno integrirati više vrsta podataka (npr. genomske, transkriptomske, proteomske podatke) za preciznije predviđanje genskog izražaja. DNN su posebno dobre u ovoj integraciji zbog svoje sposobnosti efikasne obrade i kombiniranja složenih i raznolikih podatke.

Poglavlje 3

MODELI ZA PREDVIĐANJE GENSKOG IZRAŽAJA

3.1 Enformer

Enformer, inovativni model razvijen od strane DeepMind-a, predstavlja značajan napredak u području prediktivnog modeliranja genskog izražaja. Dizajniran kako bi efikasno uhvatio složenu regulatornu arhitekturu genoma, Enformer koristi duboke tehnike učenja za precizno predviđanje ishoda genskog izražaja. Ovaj model izdvaja se svojom primjenom arhitekture transformera, koja omogućava obradu velikih sekvencijalnih podataka uz izvanrednu točnost i efikasnost.

3.1.1 Arhitektura i funkcionalnost

Enformer je razvijen koristeći naprednu varijantu arhitekture transformera, specifično prilagođenu za analizu genoma. Ova arhitektura uključuje kombinaciju konvolucijskih slojeva i slojeva transformera, zajedno s posebno dizajniranim izlaznim slojevima za specifične organizme (3.1), što omogućuje modelu preciznu analizu i predviđanje genskog izražaja.

Enformer koristi sedam konvolucijskih slojeva na početku svoje arhitekture. Konvolucijski slojevi su ključni za preradu ulaznih genetskih sekvenci jer omogućuju modelu efektivnu identifikaciju i izolaciju lokalnih značajki unutar širokog spektra genomskih podataka. Ovi slojevi primjenjuju različite filtre na ulazne sek-

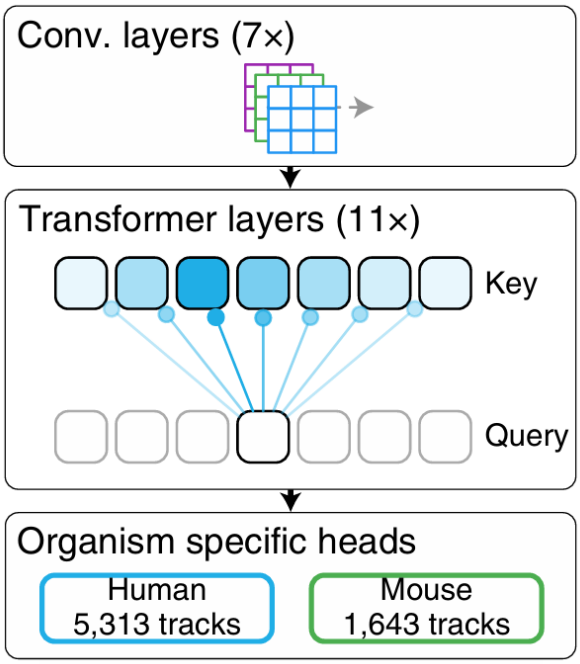
vence, generirajući skup značajki koje će se dalje obraditi u slojevima transformera. Primjena konvolucijskih slojeva omogućava modelu ekstrakciju bitnih informacija, relevantne za predviđanje genskog izražaja, iz sekvencijalnih podataka.

Nakon početnih konvolucijskih slojeva, Enformer implementira jedanaest slojeva transformera. Ovi slojevi koriste mehanizme pažnje kako bi omogućili modelu analizu i sintezu informacija iz različitih dijelova genetske sekvence istovremeno. Mehanizam pažnje u transformera omogućuje modelu "usmjeriti fokus" na relevantne dijelove ulaznih podataka, što je posebno korisno u kontekstu genoma gdje distalni elementi mogu imati značajan utjecaj na regulaciju gena. Slojevi transformera u Enformeru pomažu u modeliranju kompleksnih interakcija između različitih genetskih regija, uključujući enhancere i promotore, doprinoseći preciznijem predviđanju genskog izražaja.

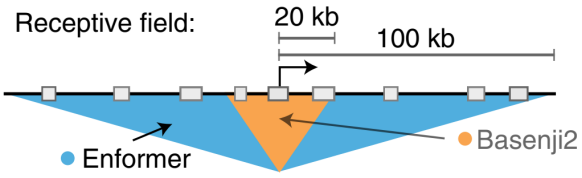
Jedinstvena značajka Enformera su dva izlazna sloja prilagođena specifičnim organizmima: jedan za ljudski genom sa 5313 staza (engl. tracks) i drugi za mišji genom sa 1643 staze. Ovi izlazni slojevi omogućuju modelu pružanje predviđanja genskog izražaja koja su prilagođena specifičnim karakteristikama i potrebama različitih organizama. Prilagodba na razinu vrsta omogućava modelu izuzetnu preciznost u svojim predviđanjima, uzimajući u obzir razlike u genetskim regulacijama koje se mogu značajno razlikovati između različitih organizama.

3.1.2 Prednosti u odnosu na prethodne modele

Jedna od ključnih prednosti Enformera u odnosu na druge modele, poput Basenji2 [19], jest njegova sposobnost obrade sekvencijskih elementa udaljenih do 100 kb od TSS (3.2). Ova značajka čini Enformer iznimno moćnim u identifikaciji i analizi regulatornih interakcija koje se događaju na velikim udaljenostima, što je često ključno za razumijevanje kompleksnih obrazaca genske regulacije u višim eukariotima, uključujući ljude.



Slika 3.1 Arhitektura modela Enformer (preuzeto iz [1])



Slika 3.2 Usporedba receptivnog polja Enformer i Basenji2 modela (preuzeto iz [1])

3.2 CRMnet

CRMnet je model razvijen za predviđanje genskog izražaja s fokusom na identifikaciju cis-regulatornih modula (CRM), koji igraju ključnu ulogu u regulaciji transkripcije. CRMnet koristi kombinaciju dubokog učenja i bioinformatičkih tehnika kako bi precizno identificirao i kvantificirao utjecaj CRM-ova na genski izražaj.

3.2.1 Arhitektura

Arhitektura modela CRMnet sastoji se od nekoliko ključnih komponenti: Squeeze and Excitation (SE) Encoder Blokova, Transformer Encoder Blokova, SE Decoder Blokova, SE Bloka i Multi-Layer Perceptron (MLP) (3.3). Ova složena struktura omogućuje modelu učinkovito učenje i analizu različitih uzoraka u genskim podacima. Dalje je naveden detaljan opis ključnih komponenata modela CRMnet:

1. **SE Encoder Blokovi:**

   SE blokovi se koriste za poboljšanje reprezentacije značajki tako što adaptivno rekalibriraju značajke kanala. Ovom metodom model pridaje veću važnost relevantnim značajkama, poboljšavajući njegovu sposobnost prepoznavanja složenih uzoraka u podacima.

2. **Transformer Encoder Blokovi:**

   Transformer encoder blokovi koriste mehanizam pažnje kako bi model mogao efikasno obrađivati sekvencijalne podatke. Ovo omogućuje modelu uzimanje u obzir udaljene interakcije između elemenata u sekvenci, što je ključno za ispravno predviđanje genskog izražaja.

3. **Decoder Blokovi SE:**

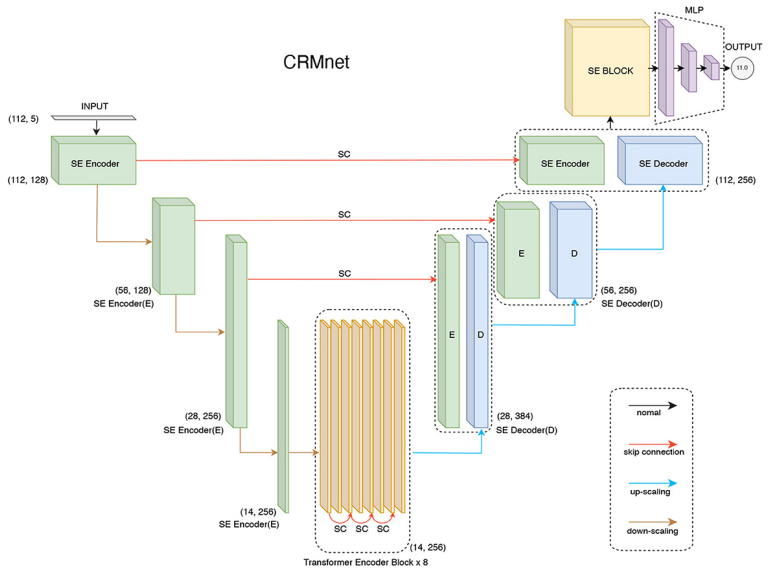
   SE decoder blokovi koriste mehanizam SE za ponovno sastavljanje značajki u dekoderu. Ovi blokovi, zajedno sa spojevima preskoka (engl. skip connections) između enkodera i dekodera, omogućuju modelu zadržavanje visoke rezolucije značajki kroz cijeli proces dekodiranja.

4. **Blok SE:**

   Blok SE je dodatni modul koji dodatno poboljšava reprezentaciju značajki kroz cijelu mrežu. Ovaj blok adaptivno prilagođava značajke kanala, omogućujući modelu bolji fokus svoje pažnje na ključne informacije.

5. **MLP:**

   MLP se koristi na kraju mreže za finalnu klasifikaciju ili regresiju. Sastoji se od nekoliko slojeva potpuno povezanih (fully connected) neurona, koji obrađuju konačne značajke i donose završnu odluku o predviđanju.



Slika 3.3 Arhitektura modela CRMnet (preuzeto iz [2])

3.2.2 Funkcionalnost

CRMnet je dizajniran kako bi integrirao različite vrste podataka i iskoristio složene uzorke u vremenskim serijama podataka o genskom izražaju. Vremenske serije (engl. time series) su nizovi podataka koji su zabilježeni ili promatrani u redoslijedu kroz vrijeme. U kontekstu genskog izražaja, vremenske serije mogu predstavljati razine genskog izražaja mjerene u različitim pozicijskim točkama unutar sekvence, omogućujući pozicijsku analizu izražaja gena.

CRMnet koristi arhitekturu sličnu U-Net arhitekturi [20], gdje enkoder i odgovarajući dekoder na istoj razini imaju skip konekciju. Ovo znači da dekoder koristi konkatenaciju uzorkovane mape značajki s odgovarajućom mapom značajki višeg rješenja iz enkodera na toj razini. Ova struktura omogućuje modelu zadržavanje detaljnih informacija tijekom cijelog procesa dekodiranja, poboljšavajući preciznost i robusnost predviđanja.

3.2.3 Primjena CRMnet-a u istraživanju genskog izražaja

CRMnet je posebno koristan za istraživanje genskog izražaja u kontekstu složenih regulatornih mreža. Njegova sposobnost prepoznavanja specifičnih sekvencijskih obrazaca i kvantificiranja njihovog utjecaja na genski izražaj omogućuje istraživačima detaljno proučavanje kako različiti cis-regulatorni elementi doprinose regulaciji gena. Ovo je posebno važno u biomedicinskim istraživanjima, gdje razumijevanje regulacije genskog izražaja može voditi do novih terapijskih ciljeva i intervencija.

3.3 Implementacija i treniranje modela

U ovoj sekciji će se opisati proces implementacije i treniranja oba modela koristeći priloženi programski kod. Oba modela su implementirana u Pythonu. Enformer koristi Sonnet [21] za implementaciju, dok CRMnet koristi Tensorflow Keras API [22]. Ovdje ćemo detaljno objasniti ključne dijelove koda i njihovu funkcionalnost.

3.3.1 Implementacija i treniranje modela Enformer

Sav dalje naveden programski kod i konstante funkcijskih parametara i varijabli su dostupne na službenom Github repozitoriju otvorenog pristupa Deepmind-a [23]. Model se može trenirati pomoću GPU-a ili CPU-a, ali se preporuča korištenje GPU-a radi kraćeg vremena treniranja.

Učitavanje podataka

Prvi korak u treniranju modela Enformer je učitavanje skupa podataka. U ovom primjeru, koriste se podaci za ljudske ('human') i mišje ('mouse') sekvence.

```
df_targets_human = get_targets('human')

human_dataset = get_dataset('human', 'train').batch(1)
                                              .repeat()
mouse_dataset = get_dataset('mouse', 'train').batch(1)
                                              .repeat()
```

```
human_mouse_dataset = tf.data.Dataset.zip((human_dataset,
                                           mouse_dataset))
                                           .prefetch(2)

it = iter(mouse_dataset)
example = next(it)

it = iter(human_mouse_dataset)
example = next(it)
for i in range(len(example)):
  print(['human', 'mouse'][i])
  print({k: (v.shape, v.dtype) for k,v in example[i].items()
   })
```

Isječak koda 3.1 Učitavnje skupa podataka za treniranje modela Enformer

Ovdje koristimo funkcije `get_dataset` za dobivanje podataka za ljudske i miševe sekvence, koje su zatim grupirane i ponovljene kako bi se omogućilo kontinuirano treniranje. Funkcija `tf.data.Dataset.zip` koristi se za kombiniranje dva skupa podataka (ljudski i mišji) u jedan skup podataka koji se može koristiti za treniranje modela. Funkcija `prefetch` osigurava učitavanje podataka unaprijed kako bi treniranje bilo što efikasnije. Primjeri podataka se iteriraju i ispisuju njihove oblike i tipovi kako bi se potvrdilo da su podaci pravilno učitani.

Definiranje funkcije treniranja

Sljedeći korak je definiranje funkcije za treniranje modela. Ova funkcija koristi TensorFlow-ov `GradientTape` za automatsko računanje gradijenata i optimizaciju parametara modela.

```

def create_step_function(model, optimizer):

  @tf.function
  def train_step(batch,
                 head,
                 optimizer_clip_norm_global=0.2):
```

```
    with tf.GradientTape() as tape:
      outputs = model(batch['sequence'], is_training=True)
                [head]
      loss = tf.reduce_mean(
          tf.keras.losses.poisson(batch['target'], outputs))

    gradients = tape
                .gradient(loss, model.trainable_variables)
    optimizer.apply(gradients, model.trainable_variables)

    return loss
  return train_step
```

Isječak koda 3.2 Definicija funkcije za treniranje modela Enformer

Funkcija `create_step_function` kreira i vraća funkciju `train_step` koja izvodi jedan korak treniranja. Unutar `train_step` funkcije, `GradientTape` se koristi za praćenje operacija kako bi se omogućilo računanje gradijenata gubitka u odnosu na trenirajuće varijable. Nakon izračuna gradijenata, oni se primjenjuju na varijable modela pomoću optimizatora.

Postavljanje modela

Postavljanje modela uključuje određivanje optimizatora, algoritma za određivanje optimalnih vrijednosti parametara modela, čime se definiraju stopa učenja i ostali parametri modela.

```
learning_rate = tf.Variable(0., trainable=False,
                            name='learning_rate')
optimizer = snt.optimizers.Adam(learning_rate=learning_rate)
num_warmup_steps = 5000
target_learning_rate = 0.0005

model = enformer.Enformer(channels=1536 // 4,
                          num_heads=8,
                          num_transformer_layers=11,
                          pooling_type='max')
```

```
11
12 train_step = create_step_function(model, optimizer)
```

Isječak koda 3.3 Postavljanje modela Enformer

Ovdje definiramo promjenjivu vrijednost parametra `learning_rate` koji se koristi za postavljanje stope učenja. Odabrani optimizator je Adam [24] (*Adaptive Moment Estimation*), a inicijalno postavljena stopa učenja je nula i kasnije se povećava tijekom treniranja. Optimizator Adam je jedan od najčešće korištenih optimizatora za treniranje dubokih neuronskih mreža [25]. Kombinira najbolje osobine dvaju drugih ekstenzivno korištenih optimizatora: AdaGrad i RMSProp. Glavna prednost Adam optimizatora je sposobnost automatske prilagodbe stope učenja za svaki parametar modela. Model je instanca klase Enformer sa specifičnim parametrima kao što su broj kanala, broj glava u transformatoru i broj slojeva transformatora. Na kraju, kreiramo `train_step` funkciju za treniranje koristeći definirani model i optimizator.

Treniranje modela

Treniranje modela provodi se kroz definirani broj epoha i koraka po epohi. Stopa učenja se prilagođava tijekom treniranja, a gubitak se računa za svaki korak treniranja.

```
1 steps_per_epoch = 20
2 num_epochs = 5
3
4 data_it = iter(human_mouse_dataset)
5 global_step = 0
6 for epoch_i in range(num_epochs):
7   for i in tqdm(range(steps_per_epoch)):
8     global_step += 1
9
10    if global_step > 1:
11      learning_rate_frac = tf.math.minimum(
12          1.0, global_step / tf.math
13                                .maximum(
14                                1.0,
```

```
                                num_warmup_steps))
    learning_rate.assign(target_learning_rate *
                         learning_rate_frac)

  batch_human, batch_mouse = next(data_it)

  loss_human = train_step(batch=batch_human, head='human')
  loss_mouse = train_step(batch=batch_mouse, head='mouse')

# End of epoch.
print('')
print('loss_human', loss_human.numpy(),
      'loss_mouse', loss_mouse.numpy(),
      'learning_rate', optimizer.learning_rate.numpy()
      )
```

Isječak koda 3.4 Treniranje modela Enformer

Ovdje se definira broj koraka po epohi i broj epoha za treniranje. Iterira se kroz kombinirani skup podataka (ljudski i mišji) i za svaki korak treniranja se prilagođava stopa učenja. Podaci za treniranje se grupiraju u šarže, što omogućuje primaanje veće skupove podataka odjednom, rezultirajući bržim treniranjem. Svaka šarža (engl. batch) podataka se koristi za računanje gubitka za ljudske i mišje sekvence koristeći funkciju `train_step`. Na kraju svake epohe ispisujemo gubitke i trenutnu stopu učenja kako bismo pratili napredak treniranja.

3.3.2 Implementacija i treniranje modela CRMnet

Prije samog treniranja modela, potrebno je predobraditi podatke za treniranje [26] što će u konačnici rezultirati u skupu podataka od 250GB. Treniranje modela CRMnet zahtjeva ili GPU ili Tensorflow Processing Unit (TPU). Treniranje na CPU-u je moguće, ali će rezultirati iznimno dugačkim vremenima treniranja. Sav kod i konstante su preuzete sa službene Github stranice modela CRMnet [27].

Postavljanje strategije distribuiranog treniranja

Kako bi se ubrzalo treniranje, koristi se strategija distribuiranog treniranja na GPU-ima ili TPU-ima.

```
    if tf.config.list_physical_devices('GPU'):
        strategy = tf.distribute.MirroredStrategy()
        gpus = tf.config.list_logical_devices('GPU')
        print("All devices: ", gpus)
        n_hardwares = len(gpus)
    else:  # Use the TPU Strategy
        ##TPU
        resolver = tf.distribute.cluster_resolver
                              .TPUClusterResolver(tpu='local')
        tf.config.experimental_connect_to_cluster(resolver)

        # This is the TPU initialization code that has to be
    at the beginning.
        tf.tpu.experimental.initialize_tpu_system(resolver)
        strategy = tf.distribute.TPUStrategy(resolver)
        tpus = tf.config.list_logical_devices('TPU')
        print("All devices: ", tpus)
        n_hardwares = len(tpus)
```

Isječak koda 3.5 Postavljanje strategije distribuiranog treniranja za treniranje modela CRMnet

Ovdje provjeravamo dostupnost GPU-a i postavljamo `MirroredStrategy` za distribuirano treniranje na više GPU-a. Ako GPU nije dostupan, koristimo TPU strategiju pomoću `TPUClusterResolver` i `TPUStrategy`.

Postavljanje TensorBoard profiler-a i učitavanje podataka

Za praćenje treniranja koristimo TensorBoard profiler i učitavamo prethodno obrađene podatke.

```
## tensorboard profiler
tf.profiler.experimental.server.start(6000)
```

```
3
4  if experiment_media == "complex_media":
5      saved_tf_dataset_path="preprocessed_data/complex_media/"
6      model_path="saved_model/complex_media_model/"
7      result_path="output/complex_media/"+model_arch
8  else:
9      saved_tf_dataset_path="preprocessed_data/defined_media/"
10     model_path="saved_model/defined_media_model/"
11     result_path="output/defined_media/"+model_arch
12
13 if not os.path.isdir(model_path):
14     os.makedirs(model_path, exist_ok=True)
15 if not os.path.isdir(result_path):
16     os.makedirs(result_path, exist_ok=True)
17
18 train_dataset = tf.data.experimental.load(os.path.join(
19                   saved_tf_dataset_path + "train_dataset"))
20 val_dataset = tf.data.experimental.load(os.path.join(
21                   saved_tf_dataset_path + "val_dataset"))
```

Isječak koda 3.6 Postavljanje TensorBoard profiler-a i učitavanje podataka za treniranje modela CRMnet

TensorBoard profiler pokrećemo na portu 6000 za praćenje performansi. Ovisno o eksperimentalnom mediju (`complex_media` ili `defined_media`), postavljamo putanje za učitavanje podataka, spremanje modela i rezultate. Ako putanje ne postoje, stvaramo ih. Učitavamo trenirajući i validacijski skup podataka pomoću `tf.data.experimental.load` funkcije.

Priprema skupa podataka za treniranje

Priprema skupa podataka uključuje postavljanje opcija, grupiranje podataka u šarže i predmemoriranje podataka za efikasnije treniranje.

```
1 ## Disable AutoShard.
2 options = tf.data.Options()
3 options.experimental_distribute
```

```
        .auto_shard_policy = tf.data
                                .experimental
                                .AutoShardPolicy.OFF

train_dataset = train_dataset.with_options(options)
val_dataset = val_dataset.with_options(options)

batch_size=1024*n_hardwares #number of TPU cores or GPUs
train_dataset = train_dataset.batch(batch_size,
                                    drop_remainder=True)
val_dataset = val_dataset.batch(batch_size,
                                drop_remainder=True)

train_dataset.cache()
val_dataset.cache()

train_dataset.prefetch(tf.data.AUTOTUNE)
val_dataset.prefetch(tf.data.AUTOTUNE)
```

Isječak koda 3.7 Priprema skupa podataka za treniranje modela CRMnet

Postavljamo opcije za skup podataka kako bismo onemogućili automatsko dijeljenje podataka na dijelove (`AutoShard`). Veličina šarži je postavljena proporcionalno broju dostupnih hardvera (TPU ili GPU). Podaci se grupiraju u šarže, predmemoriraju (`cache`) i unaprijed učitavaju (`prefetch`) radi poboljšanja performansi.

Kompajliranje i treniranje modela

Model se kompajlira i trenira unutar distribuirane strategije.

```
with strategy.scope():
    r_square = tfa.metrics
                  .r_square
                  .RSquare(dtype=tf.float32, y_shape=(1,))
    rmse = tf.keras.metrics.RootMeanSquaredError()
    model = return_model(model_arch)
```

```
    model.compile(optimizer=Adam(),
                  steps_per_execution=50,
                  loss = tf.keras.losses.Huber(),
                  metrics=[r_square,rmse])

model.summary()
```

Isječak koda 3.8 Kompajliranje modela CRMnet

Unutar distribuirane strategije definiramo metrike (R^2 i RMSE) te kreiramo i kompajliramo model koristeći Adam optimizator, Huber gubitak i specificirane metrike. Model se zatim ispisuje naredbom `model.summary()` kako bismo dobili pregled arhitekture.

Postavljanje rasporeda učenja i treniranje modela

Postavljamo raspored učenja, callback funkcije i započinjemo treniranje modela.

```
scheduler = CosineScheduler(max_update=50,
                            base_lr=0.001*n_hardwares,
                            final_lr=0.001*n_hardwares,
                            warmup_steps=10,
                            warmup_begin_lr=0.0001*
                                            n_hardwares)
learning_rate = tf.keras
                  .callbacks
                  .LearningRateScheduler(scheduler)
early_stop = tf.keras
                .callbacks
                .EarlyStopping(monitor='val_r_square',
                               patience=10,
                               mode='max',
                               restore_best_weights=True)

log_dir = path + "logs/fit/" +
          model_arch + "_" + experiment_media +
          "_" + datetime
```

```
            .datetime.now().strftime("%Y%m%d-%H%M%S")
tensorboard_callback = tf.keras
                          .callbacks
                          .TensorBoard(log_dir=log_dir,
                                       histogram_freq=1)

tik = time.time()
history = model.fit(x=train_dataset,
                    epochs=50,
                    batch_size=batch_size,
                    verbose=2,
                    validation_data=val_dataset,
                    callbacks=[tensorboard_callback,
                               early_stop,
                               learning_rate])
tok = time.time()

model.save(model_path + model_arch)

result_dic = model.evaluate(val_dataset,
                            batch_size=batch_size,
                            return_dict=True)
result_dic["training_time"] = tok-tik
save_result(result_dic, result_path)
```

Isječak koda 3.9 Treniranje i spremanje modela CRMnet

Definiramo raspored učenja (`CosineScheduler`), callback za rano zaustavljanje (`EarlyStopping`) i TensorBoard callback za praćenje treniranja. Započinjemo treniranje modela pomoću `model.fit`, postavljajući broj epoha na 50. Nakon treniranja, spremamo model i evaluiramo ga na validacijskom skupu podataka, spremajući rezultate i vrijeme treniranja u odgovarajuće direktorije.

3.4 Metode vrednovanja modela

U ovom radu koristit ćemo predtrenirane modele Enformer i CRMnet za usporedbu njihovih performansi u predviđanju genske ekspresije. Performanse modela bit će vrednovane pomoću tri ključne metrike: Pearson-ov koeficijent korelacije (R), koeficijent determinacije (R^2) i Spearman-ov koeficijent korelacije (ρ).

3.4.1 Pearsonov koeficijent korelacije

Pearsonov koeficijent korelacije R mjeri linearnu korelaciju između dviju varijabli, u ovom slučaju između predviđenih i stvarnih vrijednosti genske ekspresije. Vrijednost R kreće se u rasponu od -1 do 1, gdje 1 označava savršenu pozitivnu linearnu korelaciju, -1 savršenu negativnu linearnu korelaciju, a 0 odsutnost linearne korelacije [28].

Ulazni podaci za izračunavanje R uključuju:

- Predviđene vrijednosti genske ekspresije ($\hat{y}$)
- Stvarne vrijednosti genske ekspresije (y)

Proces računanja R:

$$\bar{\hat{y}} = \frac{1}{n}\sum_{i=1}^{n}\hat{y}_i, \quad \bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i \tag{3.1}$$

$$(\hat{y}_i - \bar{\hat{y}}), \quad (y_i - \bar{y}) \tag{3.2}$$

$$\sum_{i=1}^{n}(\hat{y}_i - \bar{\hat{y}})(y_i - \bar{y}) \tag{3.3}$$

$$\sqrt{\sum_{i=1}^{n}(\hat{y}_i - \bar{\hat{y}})^2}, \quad \sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2} \tag{3.4}$$

$$R = \frac{\sum_{i=1}^{n}(\hat{y}_i - \bar{\hat{y}})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(\hat{y}_i - \bar{\hat{y}})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}} \tag{3.5}$$

gdje je:

- R - Pearsonov koeficijent korelacije

- $\hat{y}_i$ - Predviđena vrijednost za i-ti uzorak
- y_i - Stvarna vrijednost za i-ti uzorak
- n - Ukupan broj vrijednosti

Interpretacija rezultata R je sljedeća: vrijednosti bliže 1 ili -1 ukazuju na snažnu linearnu korelaciju, dok vrijednosti bliže 0 ukazuju na slabu ili nikakvu linearnu korelaciju. Visoka vrijednost R ukazuje na učinkovito predviđanje genskog izražaja modela, koji je linearno povezan sa stvarnim vrijednostima, što znači da model dobro razumije osnovne obrasce u podacima.

3.4.2 Spearmanov koeficijent korelacije

Spearmanov koeficijent korelacije mjeri monotonu povezanost između dviju varijabli. Za razliku od Pearsonovog koeficijenta korelacije, Spearmanov koeficijent ne zahtijeva linearni odnos između varijabli, već se fokusira na to jesu li varijable monotono povezane, tj. ako jedna varijabla raste, raste li i druga (ili obrnuto). Vrijednosti Spearmanovog koeficijenta kreću se također od -1 do 1 [29].

Ulazni podaci za izračunavanje ρ uključuju:

- Predviđene vrijednosti genske ekspresije ($\hat{y}$)
- Stvarne vrijednosti genske ekspresije (y)

Proces računanja ρ:

$$\text{Rangiranje } R_{\hat{y}}, R_y \tag{3.6}$$

$$d_i = R_{\hat{y}_i} - R_{y_i} \tag{3.7}$$

$$\rho = 1 - \frac{6 \sum_{i=1}^{n} d_i^2}{n(n^2 - 1)} \tag{3.8}$$

gdje je:

- ρ - Spearmanov koeficijent korelacije
- $R_{\hat{y}}$ - Rang predviđenih vrijednosti
- R_y - Rang stvarnih vrijednosti
- d_i - Razlika između rangova predviđenih i stvarnih vrijednosti za i-tu vrijed-

nost

- n - Ukupan broj vrijednosti

Interpretacija rezultata ρ je sljedeća: vrijednosti bliže 1 ili -1 ukazuju na snažnu monotonu povezanost, dok vrijednosti bliže 0 ukazuju na slabu ili nikakvu monotonu povezanost. Visoka vrijednost Spearmanovog koeficijenta ukazuje na uspješno predviđanje genskog izražaja modela koji je monotono povezan sa stvarnim vrijednostima, što znači da model dobro razumije osnovne rangove u podacima.

3.4.3 Koeficijent determinacije

Koeficijent determinacije R^2 mjeri udio varijance stvarnih vrijednosti koji je objašnjen predviđenim vrijednostima modela. Vrijednost R^2 kreće se od 0 do 1, gdje 1 označava savršeno objašnjenje varijance, dok 0 označava da model ne objašnjava varijancu podataka bolje od prosječne vrijednosti [30].

Ulazni podaci za izračunavanje R^2 uključuju:

- Predviđene vrijednosti genske ekspresije ($\hat{y}$)
- Stvarne vrijednosti genske ekspresije (y)

Proces računanja R^2:

$$\text{TSS} = \sum_{i=1}^{n}(y_i - \bar{y})^2 \tag{3.9}$$

$$\text{RSS} = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{3.10}$$

$$R^2 = 1 - \frac{\text{RSS}}{\text{TSS}} \tag{3.11}$$

gdje je:

- TSS - Ukupna varijanca stvarnih vrijednosti (engl. Total Sum of Squares)
- RSS - Rezidualna varijanca između stvarnih i predviđenih vrijednosti (engl. Residual Sum of Squares)
- R^2 - Koeficijent determinacije

- y_i - Stvarna vrijednost za i-ti uzorak
- $\hat{y}_i$ - Predviđena vrijednost za i-ti uzorak
- $\bar{y}$ - Prosječna vrijednost stvarnih vrijednosti
- n - Ukupan broj vrijednosti

Interpretacija rezultata R^2 je sljedeća: vrijednosti bliže 1 ukazuju na visoku sposobnost objašnjavanja varijance podataka modela, dok vrijednosti bliže 0 ukazuju na nisku sposobnost objašnjavanja varijance podataka. Visoka vrijednost R^2 sugerira ne samo da model uspješno predviđa gensku ekspresiju, već i da većina varijabilnosti u stvarnim vrijednostima može biti objašnjena modelom.

3.4.4 Vrednovanje modela

Vrednovanje modela Enformer i CRMnet bit će izvedeno na skupu podataka koji nije korišten tijekom treniranja modela kako bismo osigurali da rezultati odražavaju stvarnu sposobnost modela generalizacije znanja. Upotrebom predtrenirane verzije modela Enformer i CRMnet omogućit će nam ispitivanje njihovih performansi bez potrebe za dugotrajnim procesom treniranja.

Za svaki model izračunat ćemo vrijednosti R, R^2 i ρ na testnom skupu podataka [26]. Ove vrijednosti će nam pružiti kvantitativni uvid u to koliko dobro svaki model predviđa gensku ekspresiju.

3.4.5 Podaci i procedure

Ulazni podaci za vrednovanje uključivat će stvarne vrijednosti genske ekspresije, koje su mjerene eksperimentalno, te predviđene vrijednosti koje generiraju modeli. Proces vrednovanja uključuje sljedeće korake:

1. **Priprema podataka:** Podaci će biti obrađeni za oba modela kako bi se mogle generirati predviđene vrijednosti
2. **Predviđanje:** Izgrađeni modeli će generirati predviđene vrijednosti genske ekspresije za testni skup podataka.

3. **Računanje metrika:** R, R^2 i ρ bit će izračunati koristeći stvarne i predviđene vrijednosti iz testnog skupa.
4. **Vrednovanje modela:** Dobivene vrijednosti metrika bit će interpretirane kako bi se procijenila učinkovitost svakog modela.

Korištenjem ovih postupaka, dobit ćemo jasnu i preciznu sliku o tome koliko su modeli Enformer i CRMnet učinkoviti u predviđanju genskog izražaja, te koje su njihove relativne prednosti i nedostatci.

Poglavlje 4

REZULTATI

U ovom poglavlju detaljno će se analizirati performanse prethodno treniranih modela Enformer i CRMnet u predviđanju genskog izražaja. Modeli će biti testirani na različitim skupovima podataka, zato što je domena primjene CRMneta genom kvasca, dok se Enformer može primijeniti na ljudskom i mišjem genomu. Kako bi se dobile najbolje moguće i autentične performanse modela, svaki će biti testiran na skupu podataka iz svoje domene primjene.

CRMnet će biti testiran na skupu podataka preuzet iz [26] koji sadrži milijune sekvenci promotora duljine 112 baza, gdje je za svaku vezana jedinstvena vrijednost genskog izražaja kvasca *Saccharomyces cerevisiae*. S druge strane, Enformer će biti testiran na skupu podataka iz [31] koji sadrži nekoliko milijuna sekvenci duljine 393,216 baza te njihovog genskog izražaja za 5,313 različitih "ciljeva" (engl. targets) koji predstavljaju gen koji se izražava te kojom metodom je dobiven izražaj (npr. DNASE: srce odraslog muškarca (dob 27)).

Korištene metrike za vrednovanje modela su Pearsonov koeficijent korelacije (R), koeficijent determinacije (R^2) i Spearmanov koeficijent korelacije (ρ). Osim ovih metrika, dodatno ćemo izračunati i srednju apsolutnu pogrešku (MAE, engl. mean absolute error), srednju kvadratnu pogrešku (MSE, engl. mean square error), relativnu srednju apsolutnu pogrešku (RMAE, engl. relative mean absolute error) te relativnu srednju kvadratnu pogrešku (RMSE, engl. relative mean square error). Ove dodatne metrike pružaju dublji uvid u točnost i preciznost predviđanja modela, omogućujući nam kvantifikaciju njihovih pogrešaka na različite načine.

Sve metrike će biti izračunate za skup primjeraka koji čini testni skup poda-

taka, također se mogu izračunati za svaki primjerak genoma posebno. R mjeri linearni odnos između predviđanja modela i stvarnih vrijednosti, dok R^2 prikazuje koliko varijacija u stvarnim vrijednostima može biti objašnjeno modelom. ρ mjeri monotoni odnos između varijabli, bez pretpostavke o linearnoj povezanosti. MAE pokazuje prosječnu apsolutnu razliku između predviđenih i stvarnih vrijednosti, dok MSE naglašava veće pogreške zbog kvadriranja razlika. RMAE i RMSE prilagođene su varijacije ovih metrika koje omogućuju usporedbu performansi modela s različitim skalama podataka.

Ove metrike pružaju sveobuhvatan prikaz kvalitete predviđanja modela, omogućujući nam detaljnu analizu njihovih performansi i sposobnosti generalizacije na testnim skupovima podataka iz njihovih specifičnih domena primjene.

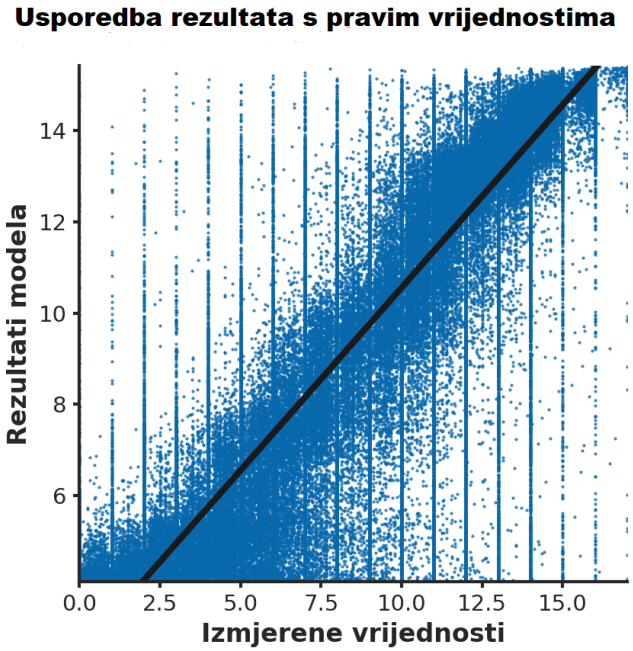
4.1 Performanse modela CRMnet

Model CRMnet je validiran na spomenutom skupu podataka kako bi se procijenila njegova učinkovitost u predviđanju genskog izražaja. Dobivene vrijednosti vrednovanih metrika za CRMnet su sljedeće:

- **R**: 0.8797
- $\mathbf{R^2}$: 0.7516
- ρ: 0.8835
- **MAE**: 1.4985
- **MSE**: 4.2849
- **RMAE**: 0.167
- **RMSE**: 0.4774

Ove metrike pokazuju visok stupanj slaganja između predviđenih i stvarnih vrijednosti genskog izražaja, što sugerira uspješno prepoznavanje obrazaca u podacima i generiranje preciznih predviđanja modela CRMnet.

Na slici 4.1 prikazana je povezanost između pravih vrijednosti genske ekspresije i vrijednosti koje je predvidio model CRMnet. Uglavnom je vidljiv sklad predviđenih vrijednosti s pravim vrijednostima, što dodatno potvrđuje visoke vrijednosti



Slika 4.1 Usporedba rezultata modela CRMnet s istinitim vrijednostima

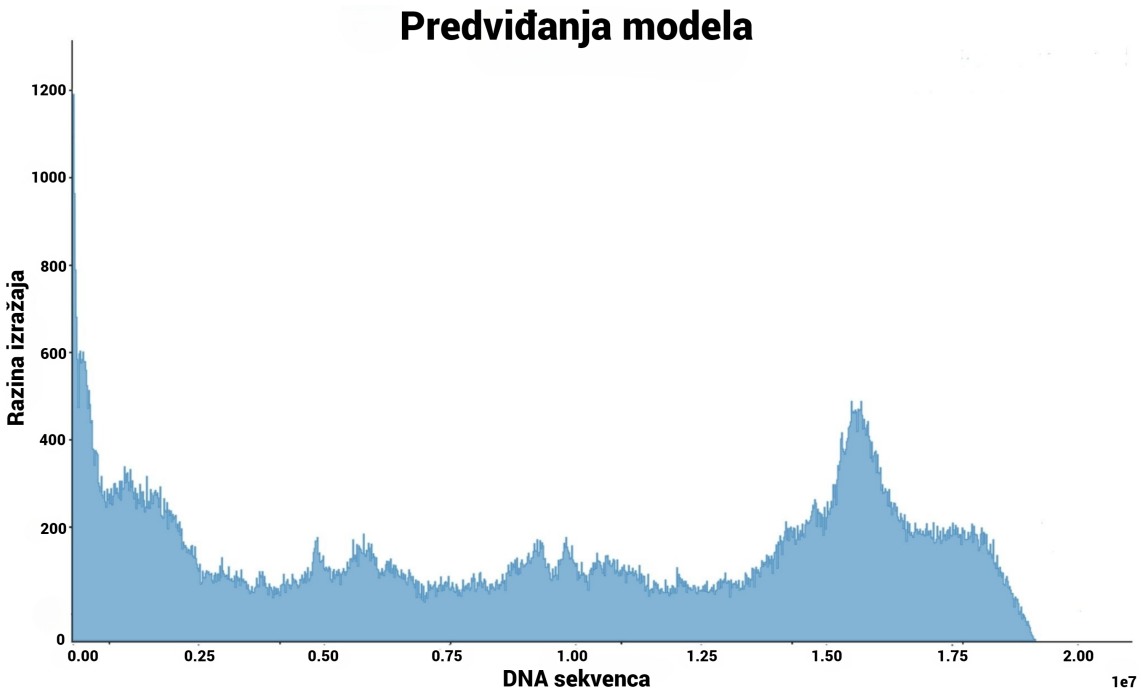
R i ρ metrika.

Vrijednost R od 0.8797 ukazuje na snažnu pozitivnu linearnu korelaciju između predviđenih i stvarnih vrijednosti. Ova visoka korelacija implicira precizno predviđanje genskog izražaja te značajnu linearnost između predviđanja modela i stvarnih mjerenja.

R^2 od 0.7516 znači da model objašnjava oko 75.16% varijance u stvarnim podacima. To je znak solidne sposobnosti generalizacije i pouzdanog predviđanja genskog izražaja na neviđenim podacima.

ρ od 0.8835 dodatno potvrđuje jaku monotonu povezanost između predviđenih i stvarnih vrijednosti. Ova mjera, koja je osjetljivija na nelinearne odnose, potvrđuje uspješno prepoznavanje obrazaca i u slučajevima gdje odnosi nisu striktno linearni.

Rezultati pokazuju visoku točnost i sposobnost generalizacije u predviđanju genskog izražaja. Kombinacija visokih vrijednosti Pearsonovog koeficijenta korelacije, koeficijenta determinacije, Spearmanovog koeficijenta korelacije te niskih vrijednosti metrika pogrešaka sugerira precizno hvatanje ključnih obrazaca u po-



Slika 4.2 Predviđene vrijednosti genskog izražaja modela CRMnet

dacima i stvaranje pouzdanih predviđanja.

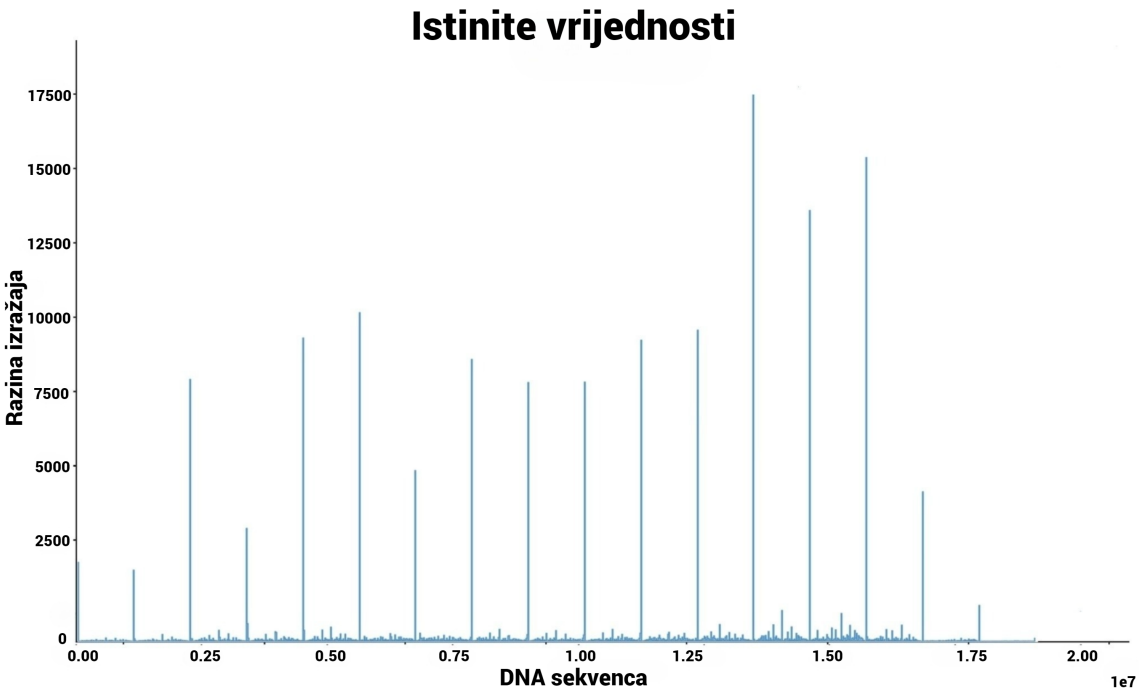
Ovaj detaljan prikaz performansi modela CRMnet omogućuje nam shvaćanje njegove učinkovitosti i pouzdanosti u zadatku predviđanja genske ekspresije. Nakon ovog, u sljedećoj sekciji, prikazat ćemo performanse modela Enformer, te ćemo potom usporediti oba modela kako bismo donijeli zaključke o njihovim relativnim prednostima i nedostacima.

Uz ove metrike, analizirali smo i metrike pogrešaka kako bismo dobili dublji uvid u preciznost modela.

MAE od 1.4985 pokazuje prosječnu apsolutnu razliku između predviđenih i stvarnih vrijednosti. Ovo ukazuje na prosječno odstupanje predviđanja modela za oko 1.5 jedinica od stvarnih vrijednosti.

MSE iznosi 4.2849, što naglašava veće pogreške zbog kvadriranja razlika. Ovo je korisno za razumijevanje koliko velike pogreške model može napraviti, jer kvadriranje daje veću težinu većim odstupanjima.

RMAE iznosi 0.167, što je normalizirana verzija MAE koja omogućuje uspo-

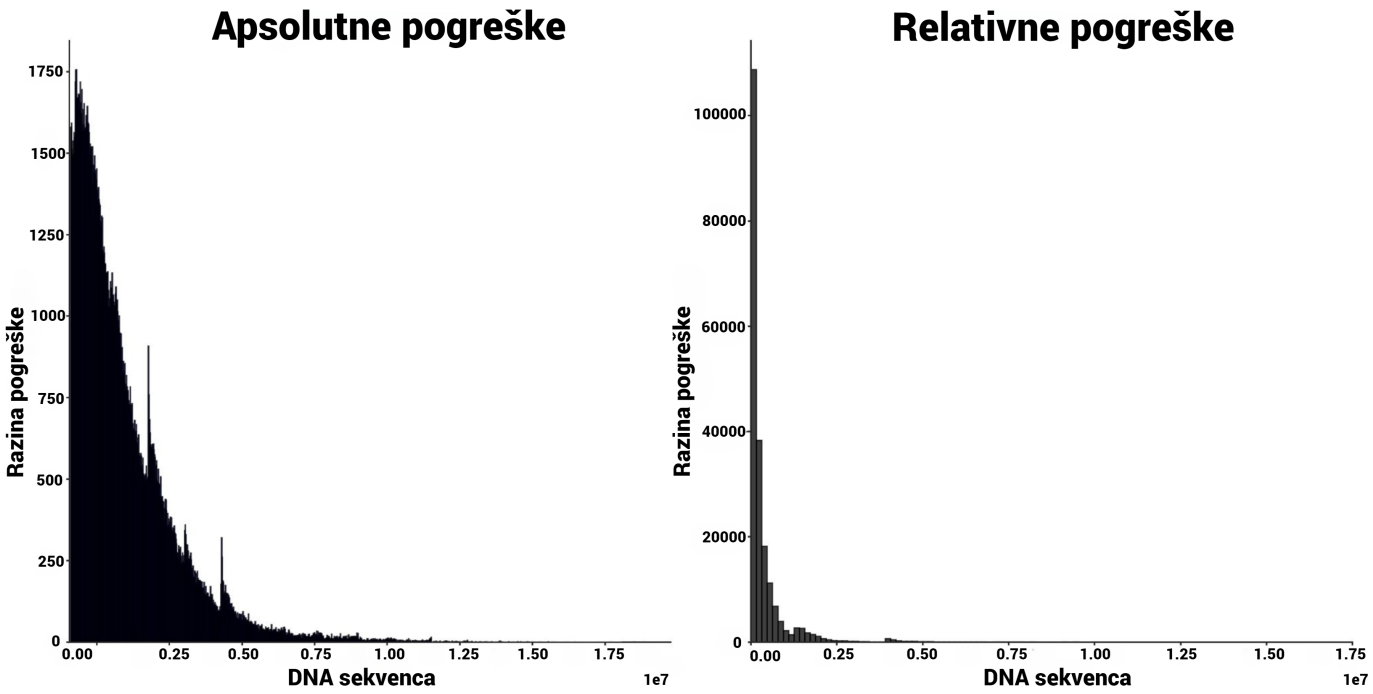


Slika 4.3 Prave vrijednosti genskog izražaja modela CRMnet

redbu performansi modela s različitim skalama podataka. Relativno mala vrijednost ukazuje na dobru točnost modela s obzirom na veličinu stvarnih vrijednosti.

RMSE od 0.4774 pruža dodatni uvid u preciznost modela, normalizirajući MSE na sličan način kao što RMAE normalizira MAE. Niska vrijednost ukazuje na dosljedno predviđanje vrijednosti bliske stvarnim vrijednostima, s rijetkim velikim pogreškama.

Rezultati pokazuju visoku točnost i sposobnost generalizacije u predviđanju genskog izražaja kod modela CRMnet. Kombinacija visokih vrijednosti Pearsonovog koeficijenta korelacije, koeficijenta determinacije i Spearmanovog koeficijenta korelacije sugerira precizno hvatanje ključnih obrazaca modela CRMnet u podacima i daje pouzdana predviđanja.



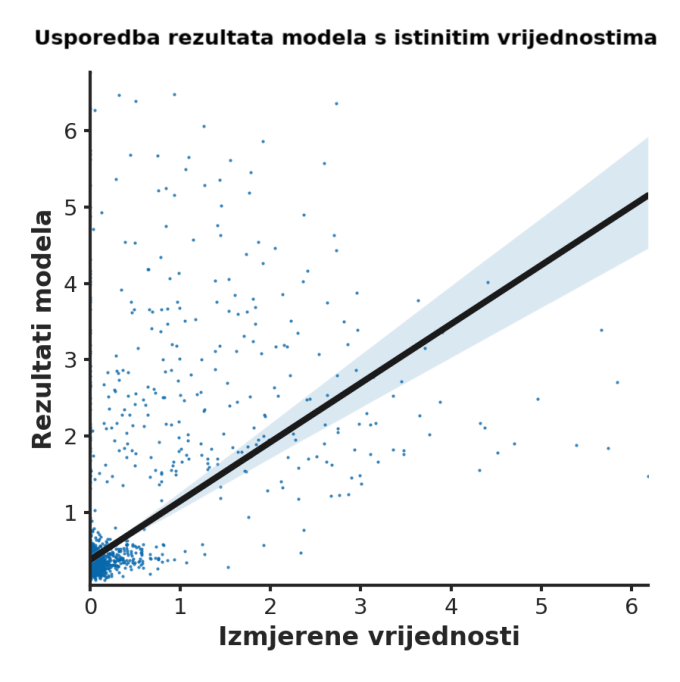
Slika 4.4 Apsolutne i relativne pogreške modela CRMnet

4.2 Performanse modela Enformer

Model Enformer je vrednovan na istom skupu podataka kako bi se procijenila njegova učinkovitost u predviđanju genskog izražaja. Dobivene vrijednosti vrednovanih metrika za Enformer su sljedeće:

- **R**: 0.6130
- $\mathbf{R}^2$: -0.0756
- ρ: -0.0847
- **MAE**: 0.8081
- **MSE**: 5.6006
- **RMAE**: 0.8619
- **RMSE**: 5.9736

Ove metrike ukazuju na slabiju uspješnost modela Enformer u predviđanju



Slika 4.5 Usporedba rezultata modela Enformer s istinitim vrijednostima

genskog izražaja u usporedbi s modelom CRMnet. Iako je model uspješno treniran, rezultati pokazuju poteškoće u prepoznavanju obrazaca u podacima i generiranju točnih predviđanja, posebno kada se radi o DNA sekvencama ljudskog genoma i pripadajućim razinama genskog izražaja za te sekvence.

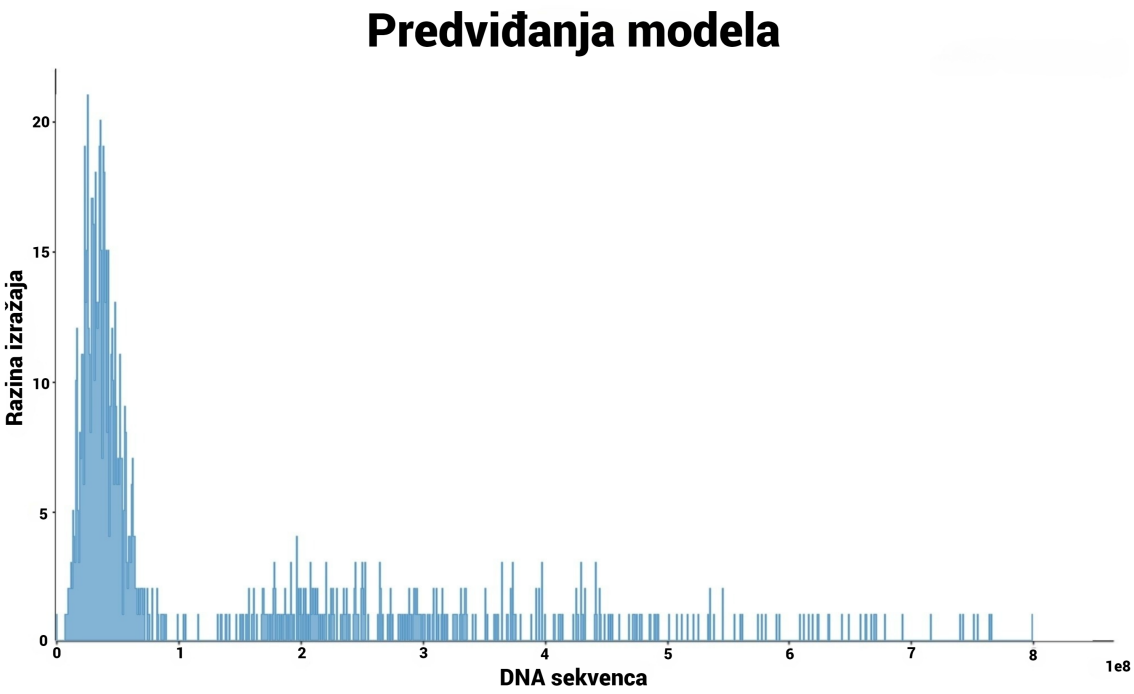
Međutim, treba naglasiti da rad s velikim skupovima podataka, poput onog korištenog za vrednovanje modela Enformer, nosi sa sobom određene izazove koji mogu značajno utjecati na performanse modela, kao i na samu evaluaciju rezultata.

Kao što je spomenuto, skup podataka korišten za vrednovanje modela Enformer bio je znatno veći od onog korištenog za model CRMnet. Ovo povećanje veličine skupa podataka donosi sa sobom nekoliko ključnih izazova. Veliki skupovi podataka zahtijevaju znatno više vremena za obradu i analizu – konkretno, obrada na HPC arhitekturi [32] može trajati nekoliko dana dulje, ovisno o kompleksnosti zadataka i količini podataka. U slučaju modela Enformer, vrijeme potrebno za provođenje evaluacije bilo je 3 dana duže od vremena evaluacije modela CRMnet, što je zahtijevalo korištenje snažnijih računalnih resursa. Ovaj problem može biti

posebno izražen kada se rade iterativne optimizacije ili kada se provode višestruki eksperimenti, što može produžiti istraživački proces. Jedno moguće rješenje ovog problema je korištenje distribuiranih sustava obrade podataka, poput clustera ili cloud platformi, koje omogućuju paralelizaciju zadataka i smanjenje ukupnog vremena obrade.

Kako veličina skupa podataka raste, postaje sve teže grafički prikazati dobivene rezultate na način koji je informativan i pregledan. Prikaz rezultata za model Enformer zahtijevao je značajne prilagodbe, pošto jednostavne grafičke metode poput raspršnih dijagrama mogu postati pretrpane i nečitljive s velikim brojem točaka. U ovom slučaju su podaci neravnomjerno prikazani na dijagramu na slici 4.5, što može otežati interpretaciju rezultata. Na dijagramu se može primijetiti pretrpanost točkama na određenim područjima, što otežava razlikovanje pojedinačnih rezultata, dok su druga područja gotovo prazna. Ova neravnomjernost može biti posljedica prirode podataka korištenih za vrednovanje modela ili načina na koji su podaci agregirani za prikaz. U takvim slučajevima, potrebno je koristiti naprednije tehnike vizualizacije, poput toplinske karte (engl. heatmap), sumarnih statistika ili tehnika smanjenja dimenzionalnosti, kao što su PCA i t-SNE. PCA (engl. Principal Component Analysis) je statistička metoda koja transformira visoko-dimenzionalne podatke u manji broj dimenzija dok zadržava što je više moguće varijance u podacima. Tako se podaci prikazuju u obliku nekoliko glavnih komponenti koje sažimaju ključne informacije, što olakšava vizualizaciju i analizu. S druge strane, t-SNE (t-distributed Stochastic Neighbor Embedding) je metoda nelinearnog smanjenja dimenzionalnosti koja je posebno korisna za vizualizaciju podataka u dvije ili tri dimenzije. t-SNE funkcionira tako što smanjuje dimenzionalnost podataka dok pokušava zadržati relativne udaljenosti između podatkovnih točaka, čime se zadržava njihova lokalna struktura. Ova metoda je vrlo učinkovita za prikazivanje složenih obrazaca u podacima, ali može biti računalno intenzivna i sklona različitim rezultatima ovisno o parametrima.

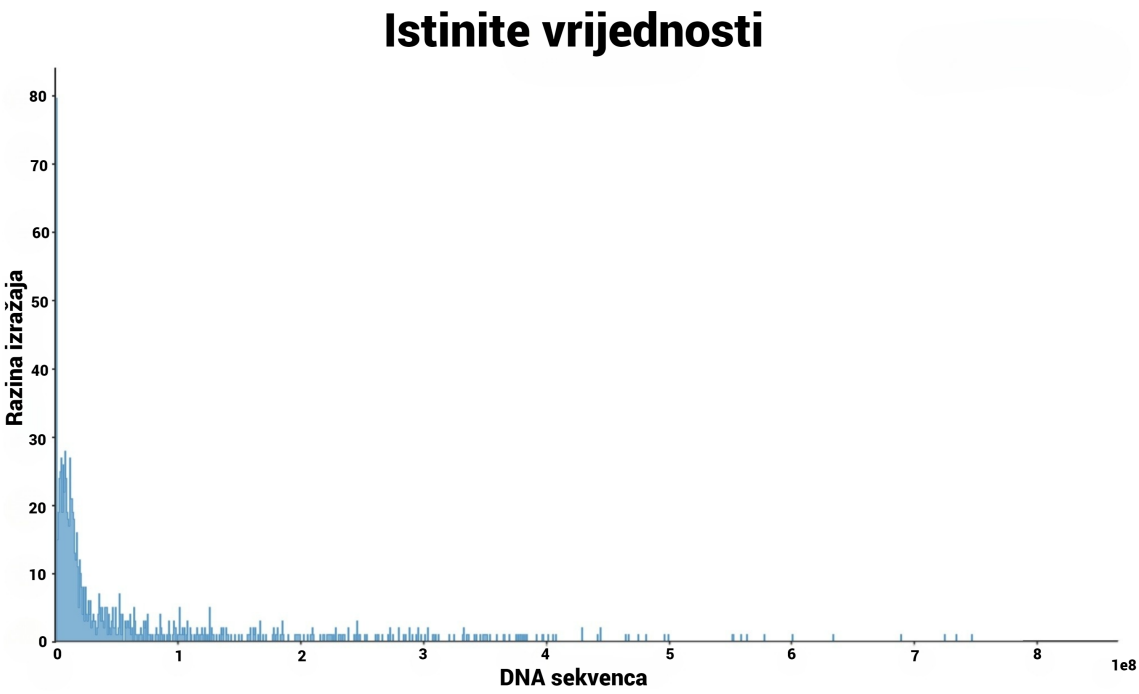
Veći skupovi podataka često su raznovrsniji i mogu uključivati više varijabilnosti, uključujući prisutnost izoliranih točaka (engl. outlier), koji mogu negativno utjecati na performanse modela. U modelu Enformer, ovo se moglo odraziti na visoke vrijednosti pogrešaka poput MSE i RMSE. Identifikacija i obrada takvih izoliranih točaka može biti izazovna, no korištenje statističkih metoda ili tehnika



Slika 4.6 Predviđene vrijednosti genskog izražaja modela Enformer

poput odstranjivanja izoliranih točaka, ponderiranja točaka podataka ili korištenja regularizacijskih tehnika može pomoći u ublažavanju ovog problema.

Iako se može očekivati da veći skupovi podataka poboljšavaju sposobnost modela za generalizaciju, u stvarnosti često dolazi do problema poput prenaučenosti (engl. overfitting). Ovaj problem se može pojaviti zbog visoke dimenzionalnosti podataka i kompleksnosti modela, što može kod modela Enformer uzrokovati učenje specifičnih obrazaca koji se pojavljuju samo u skupu podataka za treniranje, umjesto generalnih pravila. Rješenje može biti korištenje tehnika regularizacije, kao što su L1 i L2 regularizacija, dropout ili unakrsna validacija (engl. cross-validation). L1 regularizacija (Lasso) [33] dodaje apsolutne vrijednosti koeficijenata modela kao penalizaciju u funkciju gubitka, čime se preferira rješenja s manjim brojem značajki, tj. pojedine značajke mogu biti eliminirane. L2 regularizacija (Ridge) [33] koristi kvadrat koeficijenata modela kao penalizaciju, čime smanjuje utjecaj pojedinih značajki i sprječava ekstremne vrijednosti koeficijenata. Dropout [34] je tehnika koja se često koristi u dubokim neuronskim mrežama, a funkcionira tako da se tijekom treniranja nasumično "isključuje" postotak neurona u mreži, što



Slika 4.7 Prave vrijednosti genskog izražaja modela Enformer

sprječava pretjeranu ovisnost modela o određenim putevima kroz mrežu i time poboljšava generalizaciju. Unakrsna validacija [35] je metoda procjene performansi modela gdje se podaci podijele u više skupova, te se model trenira i testira na različitim kombinacijama tih skupova, čime se dobiva robusnija procjena njegove sposobnosti generalizacije na neviđene podatke.

Kako raste veličina skupa podataka, povećavaju se i tehnički izazovi u rukovanju takvim podacima, uključujući probleme skalabilnosti u memoriji i skladištenju podataka. Veliki skupovi podataka mogu zahtijevati napredne tehnike za upravljanje memorijom i procesiranje, poput korištenja distribuiranih sustava kao što su Hadoop ili Spark za paralelnu obradu podataka, kao i optimizaciju korištenja memorije kroz tehnike poput *chunking* ili *mini-batch* obrade. Hadoop [36] je okvir za distribuirano skladištenje i obradu velikih skupova podataka s pomoću skupa jednostavnih programskih modela, omogućujući skalabilnu analizu podataka na klasterima računalnih strojeva. Spark [37] je također distribuirana platforma za obradu podataka, ali je optimizirana za brže procesiranje podataka u memoriji (engl. in-memory processing), što je korisno za iterativne zadatke kao što su treni-

ranje modela strojnog učenja. *Chunking* [38] je tehnika koja razbija velike skupove podataka u manje dijelove (engl. chunks), koji se potom obrađuju pojedinačno, čime se smanjuje memorijski otisak i olakšava upravljanje podacima. *Mini-batch* [39] obrada je slična kao *chunking*, ali se koristi posebno u kontekstu treniranja modela strojnog učenja, gdje se podaci dijele na manje grupe (engl. mini-batches) koje se zatim koriste za ažuriranje modela, omogućujući brže treniranje i smanjenje potrošnje memorije.

Vrijednost R od 0.6130 ukazuje na umjerenu pozitivnu linearnu korelaciju između predviđenih i stvarnih vrijednosti, ali s obzirom na značajna odstupanja, točnost modela je daleko ispod očekivanog. Ova korelacija implicira neuspješnost modela kod preciznog predviđanja genskog izražaja i postojanje ograničene linearne veze između predviđanja modela i stvarnih mjerenja.

R^2 od -0.0756 ukazuje na negativnu vrijednost, što ukazuje na neuspješnost modela kod objašnjavanja varijacije u stvarnim podacima. Ova vrijednost ukazuje na modelove poteškoće s generalizacijom i mogućnost pogoršanja predviđanja u odnosu na jednostavne metode poput korištenja prosječne vrijednosti.

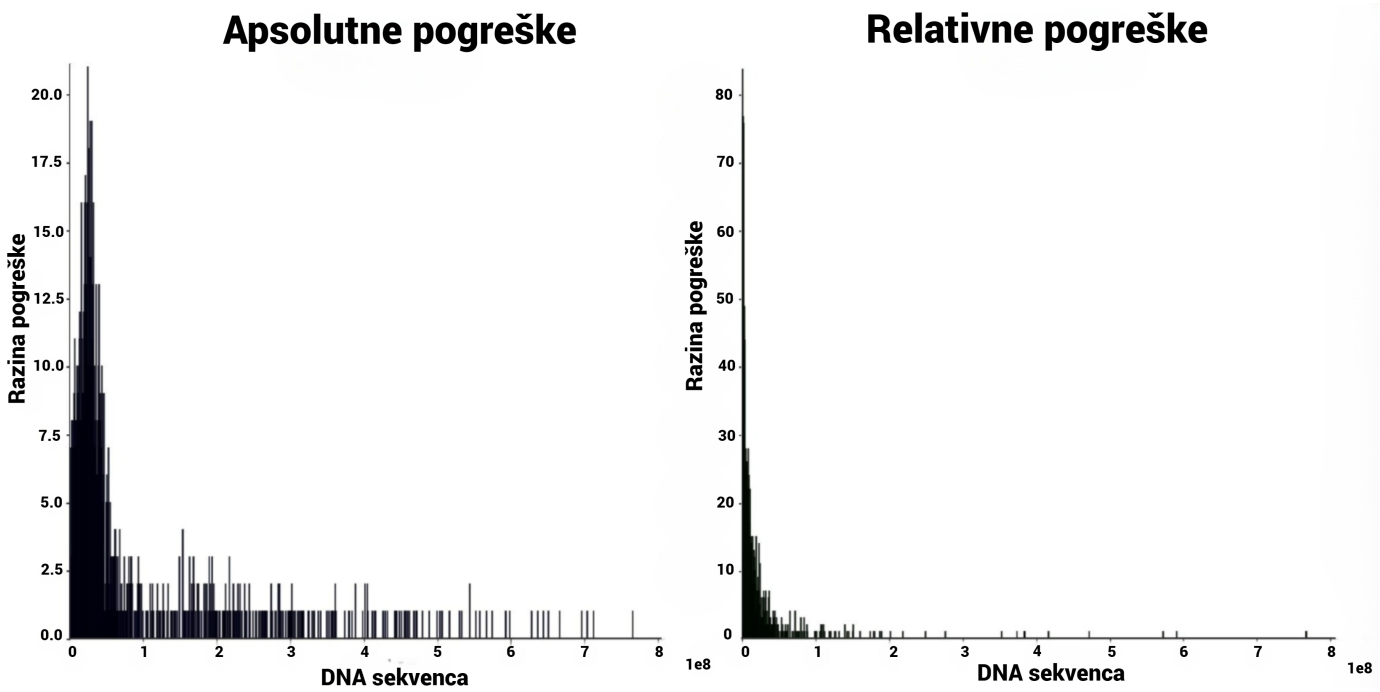
ρ od -0.0847 dodatno potvrđuje negativnu monotonu povezanost između predviđenih i stvarnih vrijednosti. Ova mjera, koja je osjetljiva na nelinearne odnose, ukazuje na nemogućnost pravilnog prepoznavanja obrazaca, čak i u slučajevima gdje odnosi nisu striktno linearni.

Rezultati sugeriraju pokazivanje značajnih poteškoća u točnom predviđanju genskog izražaja kod modela Enformer. Niske vrijednosti Pearsonovog i Spearmanovog koeficijenta korelacije, kao i negativna vrijednost koeficijenta determinacije, ukazuju na probleme u prepoznavanju obrazaca u podacima i generiranju točnih predviđanja.

Uz ove metrike, analizirali smo i metrike pogrešaka kako bismo dobili dublji uvid u preciznost modela.

MAE od 0.8081 pokazuje prosječnu apsolutnu razliku između predviđenih i stvarnih vrijednosti. Ova vrijednost ukazuje prosječno odstupanje predviđanja modela za manje od jedne jedinice od stvarnih vrijednosti, što je iznenađujuće s obzirom na slabije rezultate ostalih metrika.

MSE iznosi 5.6006, što naglašava veće pogreške zbog kvadriranja razlika. Ova



Slika 4.8 Apsolutne i relativne pogreške modela Enformer

vrijednost sugerira generiranje značajnih pogrešaka kod modela, što potvrđuje nisku razinu točnosti predviđanja.

RMAE iznosi 0.8619, što je normalizirana verzija MAE. Relativno visoka vrijednost ukazuje slabiju preciznost modela u usporedbi s drugim modelima, što dodatno potvrđuje njegovu ograničenu sposobnost u prepoznavanju obrazaca u podacima.

RMSE od 5.9736 pruža dodatni uvid u preciznost modela, ukazujući na značajna odstupanja u predviđanjima, s naglaskom na veće pogreške. Visoka vrijednost RMSE pokazuje prisutnost poteškoća u točnom predviđanju i često generira velike pogreške.

Zaključno, rezultati pokazuju slabiju točnost i sposobnost generalizacije modela Enformer u predviđanju genskog izražaja. Niske vrijednosti Pearsonovog i Spearmanovog koeficijenta korelacije, te negativna vrijednost koeficijenta determinacije, ukazuju na ograničenu sposobnost modela u preciznom hvatanju ključnih obrazaca u podacima i davanju pouzdanih predviđanja.

Poglavlje 5

ZAKLJUČAK

Ovaj rad istraživao je primjenu dubokih neuronskih mreža, konkretno modela Enformer i CRMnet, u predviđanju genskog izražaja, što predstavlja ključni izazov u području bioinformatike i molekularne biologije. Predviđanje genskog izražaja ima ključnu ulogu u biomedicinskim znanostima, osobito u kontekstu personalizirane medicine, identifikacije novih terapeutskih meta i biomarkera bolesti, te razvoja novih lijekova. Ovi modeli također mogu biti primjenjivi u poljoprivredi za poboljšanje otpornosti i produktivnosti biljaka. Stoga, sposobnost preciznog predviđanja genskog izražaja modela kao što su Enformer i CRMnet ima dalekosežne implikacije u mnogim područjima znanosti i industrije. Kroz analizu performansi ovih modela na specifičnim skupovima podataka, prikazana je njihova sposobnost učenja složenih obrazaca i generiranja preciznih predviđanja.

Rezultati su pokazali nadmoć modela CRMnet nad modelom Enformer u preciznosti predviđanja na skupu podataka za kvasac *Saccharomyces cerevisiae*, s visokim vrijednostima Pearsonovog koeficijenta korelacije (Pearson-ov koeficijent korelacije (R)), koeficijenta determinacije (Koeficijent determinacije (R^2)) i Spearmanovog koeficijenta korelacije (Spearman-ov koeficijent korelacije (ρ)). S druge strane, Enformer je imao poteškoća u postizanju sličnih rezultata, što ukazuje na izazove koje nosi rad s većim i složenijim skupovima podataka kao što su ljudski i mišji genomi. Ovi rezultati upućuju na specifične prednosti CRMnet-a u obradi podataka koji se odnose na organizme s jednostavnijom genetičkom strukturom, dok Enformer, unatoč svojoj naprednoj arhitekturi, pokazuje određena ograničenja kada je suočen s velikim i heterogenim skupovima podataka.

Jedan od ključnih doprinosa ovog rada jest osvjetljavanje prednosti i nedostataka ovih modela. CRMnet se istaknuo u domeni genoma kvasca, zahvaljujući svojoj arhitekturi optimiziranoj za prepoznavanje cis-regulatornih modula. Enformer, s druge strane, unatoč svojim naprednim mogućnostima za integraciju dugometražnih interakcija unutar genoma, imao je izazove s preciznošću na velikim skupovima podataka, što ukazuje na potrebu za daljnjim istraživanjem i optimizacijom modela u specifičnim domenama primjene, posebno kada se radi o kompleksnijim organizmima s velikim genomima.

Prilikom rada s velikim skupovima podataka, poput onih korištenih za vrednovanje modela Enformer, uočeni su izazovi poput produženog vremena obrade, potrebe za većim računalnim resursima, kao i poteškoća s prikazom i interpretacijom rezultata. Na primjer, grafički prikazi rezultata Enformera bili su neravnomjerni, otežavajući interpretaciju zbog pretrpanosti dijagrama i prisutnosti izoliranih točaka. Takvi problemi zahtijevaju primjenu naprednih tehnika vizualizacije i obrade podataka kako bi se osigurala preciznost i jasnoća rezultata.

U zaključku, iako Enformer i CRMnet nude značajne mogućnosti za predviđanje genskog izražaja, njihova primjena dolazi s izazovima, osobito kada je riječ o radu s velikim i kompleksnim skupovima podataka. Daljnja istraživanja i optimizacija ovih modela bit će ključna za poboljšanje njihove učinkovitosti i primjenjivosti u različitim biomedicinskim i industrijskim kontekstima.

Literatura

[1] Z. Avsec, V. Agarwal, D. Visentin i dr., "Effective gene expression prediction from sequence by integrating long-range interactions," *Nat Methods*, sv. 18, str. 1196–1203, 2021. DOI: 10.1038/s41592-021-01252-x.

[2] K. Ding, G. Dixit, B. J. Parker i J. Wen, "CRMnet: A deep learning model for predicting gene expression from large regulatory sequence datasets," *Frontiers in Big Data*, sv. 6, 2023., ISSN: 2624-909X. DOI: 10.3389/fdata.2023.1113402. adresa: https://www.frontiersin.org/articles/10.3389/fdata.2023.1113402.

[3] "Mastering Biology." (11. svibnja 2024.), adresa: https://pmark.pearsoncmg.com/northamerica/masteringbiology/index.html.

[4] F. Rapaport, R. Khanin, Y. Liang i dr., "Comprehensive evaluation of differential gene expression analysis methods for RNA-seq data," en, *Genome Biol*, sv. 14, br. 9, R95, 2013.

[5] S. L. He i R. Green, "Northern blotting," en, *Methods Enzymol*, sv. 530, str. 75–87, 2013.

[6] S. Deepak, K. Kottapalli, R. Rakwal i dr., "Real-Time PCR: Revolutionizing Detection and Expression Analysis of Genes," en, *Curr Genomics*, sv. 8, br. 4, str. 234–251, lipanj 2007.

[7] R. Bumgarner, "Overview of DNA microarrays: types, applications, and their future," en, *Curr Protoc Mol Biol*, sv. Chapter 22, Unit 22.1. Siječanj 2013.

[8] Z. Wang, M. Gerstein i M. Snyder, "RNA-Seq: a revolutionary tool for transcriptomics," en, *Nat Rev Genet*, sv. 10, br. 1, str. 57–63, siječanj 2009.

[9] L. Song i G. E. Crawford, "DNase-seq: a high-resolution technique for mapping active gene regulatory elements across the genome from mammalian cells," en, *Cold Spring Harb Protoc*, sv. 2010, br. 2, db.prot5384, veljača 2010.

[10] M. S. Morioka, H. Kawaji, H. Nishiyori-Sueki i dr., "Cap Analysis of Gene Expression (CAGE): A Quantitative and Genome-Wide Assay of Transcription Start Sites," *Methods Mol Biol*, sv. 2120, str. 277–301, 2020.

[11] N. Borisov, M. Sorokin, V. Tkachev, A. Garazha i A. Buzdin, "Cancer gene expression profiles associated with clinical outcomes to chemotherapy treatments," *BMC Med Genomics*, sv. 13, br. Suppl 8, str. 111, rujan 2020.

[12] W. DeGroat, H. Abdelhalim, K. Patel, D. Mendhe, S. Zeeshan i Z. Ahmed, "Discovering biomarkers associated and predicting cardiovascular disease with high accuracy using a novel nexus of machine learning techniques for precision medicine," *Scientific Reports*, sv. 14, br. 1, str. 1, siječanj 2024., ISSN: 2045-2322. DOI: 10.1038/s41598-023-50600-8. adresa: https://doi.org/10.1038/s41598-023-50600-8.

[13] J. P. F. Bai, A. V. Alekseyenko, A. Statnikov, I.-M. Wang i P. H. Wong, "Strategic applications of gene expression: from drug discovery/development to bedside," en, *AAPS J*, sv. 15, br. 2, str. 427–437, siječanj 2013.

[14] S. Mansoor, E. M. B. M. Karunathilake, T. T. Tuan i Y. S. Chung, "Genomics, Phenomics, and Machine Learning in Transforming Plant Research: Advancements and Challenges," *Horticultural Plant Journal*, 2024., ISSN: 2468-0141. DOI: https://doi.org/10.1016/j.hpj.2023.09.005. adresa: https://www.sciencedirect.com/science/article/pii/S2468014124000098.

[15] K. Choudhary, B. DeCost, C. Chen i dr., "Recent advances and applications of deep learning methods in materials science," *npj Computational Materials*, sv. 8, br. 1, str. 59, travanj 2022., ISSN: 2057-3960. DOI: 10.1038/s41524-022-00734-6. adresa: https://doi.org/10.1038/s41524-022-00734-6.

[16] M. Bahi i M. Batouche, "Deep Learning for Ligand-Based Virtual Screening in Drug Discovery," *2018 3rd International Conference on Pattern Analysis*

and Intelligent Systems (PAIS), 2018., str. 1–5. DOI: 10.1109/PAIS.2018.8598488.

[17] J. Jumper, R. Evans, A. Pritzel i dr., "Highly accurate protein structure prediction with AlphaFold," *Nature*, sv. 596, br. 7873, str. 583–589, kolovoz 2021., ISSN: 1476-4687. DOI: 10.1038/s41586-021-03819-2. adresa: https://doi.org/10.1038/s41586-021-03819-2.

[18] D. Quang, Y. Chen i X. Xie, "DANN: a deep learning approach for annotating the pathogenicity of genetic variants," en, *Bioinformatics*, sv. 31, br. 5, str. 761–763, listopad 2014.

[19] D. R. Kelley, "Cross-species regulatory sequence activity prediction," *PLOS Computational Biology*, sv. 16, br. 7, str. 1–27, srpanj 2020. DOI: 10.1371/journal.pcbi.1008050. adresa: https://doi.org/10.1371/journal.pcbi.1008050.

[20] O. Ronneberger, P. Fischer i T. Brox, *U-Net: Convolutional Networks for Biomedical Image Segmentation*, 2015. arXiv: 1505.04597 [cs.CV]. adresa: https://arxiv.org/abs/1505.04597.

[21] *Sonnet*, Google Deepmind, 3. srpnja 2024. adresa: https://sonnet.readthedocs.io/en/latest/.

[22] *Keras*, Keras Team, 3. srpnja 2024. adresa: https://keras.io/.

[23] G. Deepmind, *deepmind-research*, 26. svibnja 2024. adresa: https://github.com/google-deepmind/deepmind-research.

[24] D. P. Kingma i J. Ba, *Adam: A Method for Stochastic Optimization*, 2017. arXiv: 1412.6980 [cs.LG]. adresa: https://arxiv.org/abs/1412.6980.

[25] C. Leo. "The Math Behind The Adam Optimizer." (4. srpnja 2024.), adresa: https://towardsdatascience.com/the-math-behind-adam-optimizer-c41407efe59b.

[26] E. D. Vaishnav, C. de Boer i A. Regev, *The evolution, evolvability and engineering of gene regulatory DNA*, siječanj 2021. DOI: 10.1038/s41586-022-04506-6. adresa: https://doi.org/10.1038/s41586-022-04506-6.

[27] jiayuwen, *CRMnet*, 23. svibnja 2024. adresa: https://github.com/jiayuwen/CRMnet.

[28] P. Samuels i M. Gilchrist, *Pearson Correlation*, travanj 2014.

[29] S. K. Eden, C. Li i B. E. Shepherd, "Nonparametric estimation of Spearman's rank correlation with bivariate survival data," en, *Biometrics*, sv. 78, br. 2, str. 421–434, ožujak 2021.

[30] D. Chicco, M. J. Warrens i G. Jurman, "The coefficient of determination R-squared is more informative than SMAPE, MAE, MAPE, MSE and RMSE in regression analysis evaluation," en, *PeerJ Comput Sci*, sv. 7, e623, srpanj 2021.

[31] *Basenji barnyard*, Calico, 26. svibnja 2024. adresa: `https://console.cloud.google.com/storage/browser/basenji_barnyard/data`.

[32] Centar za napredno računanje i modeliranje, Sveučilište u Rijeci, *HPC Bura*, 1. rujna 2024. adresa: `https://cnrm.uniri.hr/hr/bura/`.

[33] Y. Shizuya, "Understanding L1 and L2 Regularization with Analytical and Probabilistic Views," *Intuition*, svibanj 2024. adresa: `https://medium.com/intuition/understanding-l1-and-l2-regularization-with-analytical-and-probabilistic-views-8386285210fc`.

[34] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever i R. Salakhutdinov, "Dropout: A Simple Way to Prevent Neural Networks from Overfitting," *Journal of Machine Learning Research*, sv. 15, str. 1929–1958, lipanj 2014. adresa: `https://www.jmlr.org/papers/volume15/srivastava14a/srivastava14a.pdf`.

[35] S. Bates, T. Hastie i R. Tibshirani, "Cross-Validation: What Does It Estimate and How Well Does It Do It?" *Journal of the American Statistical Association*, sv. 119, br. 546, str. 1434–1445, svibanj 2023., ISSN: 1537-274X. DOI: `10.1080/01621459.2023.2197686`. adresa: `http://dx.doi.org/10.1080/01621459.2023.2197686`.

[36] Apache Software Foundation, *Hadoop*, 1. rujna 2024. adresa: `https://hadoop.apache.org`.

[37] Apache Software Foundation, *Apache Spark*, 1. rujna 2024. adresa: `https://spark.apache.org`.

[38] S. Saxena, R. Raperya i N. K. Malik, "MACHINE LEARNING USING CHUNKING," *International Journal of Advance Research in Science and Engineering*, sv. No.6, Issue No. 02 veljača 2017. adresa: `https://www.ijarse.com/images/fullpdf/1486979648_G1034ijarse.pdf`.

[39] J. C. Olamendy, "Mini-Batch Gradient Descent: Optimizing Machine Learning," *Medium*, studeni 2023. adresa: `https://medium.com/@juanc.olamendy/mini-batch-gradient-descent-optimizing-machine-learning-98ef238c5225`.

Pojmovnik

ρ Spearman-ov koeficijent korelacije. 30–37, 40, 45, 47

CAGE Cap Analysis of Gene Expression. 8

cDNA Komplementarna DNA. 6, 7

ChIP-seq Chromatin Immunoprecipitation sequencing. 8

CRM Cis-regulatorni moduli. 17

Ct Prag ciklusa. 6

DNAse-seq DNAse I hypersensitive sites sequencing. 7

DNN Duboke neuronske mreže. viii, 10–14

MAE Srednja apsolutna pogreška. 36, 38–40, 45, 46

MLP Multi-Layer Perceptron. 18

mRNA Glasnički RNA. 3–6, 8, 13

MSE Srednja kvadratna pogreška. 36, 38–40, 42, 45

PCR Polymerase Chain Reaction. 6

pre-mRNA Primarna mRNA. 3

qRT-PCR Quantitive Reverse Transcription Polymerase Chain Reaction. 6

R Pearson-ov koeficijent korelacije. 30, 31, 33–37, 40, 45, 47, 56

$\mathbf{R}^2$ Koeficijent determinacije. 28, 30, 32–37, 40, 45, 47

RMAE Relativna srednja apsolutna pogreška. 36, 38–40, 46

Pojmovnik

RMSE Relativna srednja kvadratna pogreška. 36, 39, 40, 42, 46

RNA-seq RNA sequencing. 7

RT-PCR Reverse Transcription Polymerase Chain Reaction. 6

SE Squeeze and Excitation. 18

SGD Stohastički gradijentni spust. 13

TSS Transkripcijska početna mjesta. 8, 16

Sažetak

Ovaj rad istražuje primjenu dubokih neuronskih mreža, konkretno modela Enformer i CRMnet, u predviđanju genskog izražaja, što je ključno u području bioinformatike i molekularne biologije. Predviđanje genskog izražaja ima značajne primjene u biomedicinskim znanostima, uključujući personaliziranu medicinu, identifikaciju terapeutskih meta, razvoj novih lijekova, te poboljšanje otpornosti i produktivnosti biljaka u poljoprivredi. Kroz analizu performansi ovih modela na specifičnim skupovima podataka, CRMnet se pokazao superiornim u preciznosti predviđanja na skupu podataka za kvasac, primjerice s Pearsonovim koeficijentom korelacije (R) od 0.8797, dok je Enformer postigao nižu vrijednost od 0.6130. Unatoč naprednoj arhitekturi, Enformer je pokazao ograničenja pri radu s velikim i kompleksnim skupovima podataka, kao što su DNA sekvence ljudskog i mišjeg genoma. Rad također ističe izazove s obradom i interpretacijom rezultata, posebice kod većih skupova podataka, te naglašava potrebu za daljnjom optimizacijom ovih modela kako bi se poboljšala njihova primjenjivost u različitim znanstvenim i industrijskim kontekstima.

Ključne riječi **— predviđanje genskog izražaja, duboke neuronske mreže, Enformer, CRMnet**

Abstract

This thesis explores the application of deep neural networks, specifically the Enformer and CRMnet models, in gene expression prediction, which is crucial in the fields of bioinformatics and molecular biology. Gene expression prediction has significant applications in biomedical sciences, including personalized medicine, the identification of therapeutic targets, drug development, and improving crop resistance and productivity in agriculture. Through the analysis of these models' performance on specific datasets, CRMnet proved to be superior in prediction accuracy on the yeast dataset, with a Pearson correlation coefficient (R) of 0.8797, while Enformer achieved a lower value of 0.6130. Despite its advanced architecture, Enformer demonstrated limitations when working with large and complex datasets, such as DNA sequences of the human and mouse genomes. The paper

also highlights the challenges in processing and interpreting results, particularly with larger datasets, and emphasizes the need for further optimization of these models to improve their applicability in various scientific and industrial contexts.

Keywords **— gene expression prediction, deep neural networks, Enformer, CRMnet**