Sadržaj

Uvod

Bioinformatika je znanost koja proučava i sakuplja izrazito komplekse biološke podatke kao što je to genetički kod. Genetički kod je zapisan u slijedu četiri različita nukleotida u molekuli DNA i to su adenin (A), gvanin (G), citozin (C) i timin (T). Slijed tih nukleotida kodira za proteine koji obavljaju mnoge različite funkcije u stanici. Proučavanjem sekvence tih gena možemo dobiti važne informacije o njihovoj regulaciji, funkciji ili o njihovoj evoluciji. Uspoređivanjem sekvenci nekih konzerviranih gena možemo otkriti i srodnost različitih vrsta te tako dobiti važne informacije o evoluciji i razvoju pojedinih vrsta.

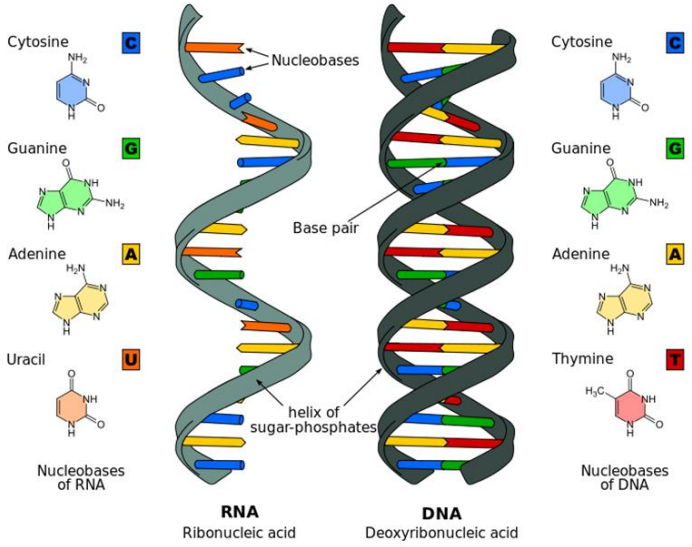
Zbog razvoja tehnologija poput NGS (eng. Next Generation sequencing), količina sekvenciranih genoma raste iz dana u dan pa je računalna analiza jedini mogući način za adekvatno proučavanje ovih velikih setova podataka. Nastankom velikih baza podataka sekvenciranih gena u kojoj su zapisane konsenzus sekvence početnih i završnih slijedova tih gena, otvara se mogućnost za prepoznavanje gena iz same sekvence gena bez popratnih eksperimentalno dobivenih podataka o njihovoj ekspresiji, tj. o prisustvu produkata tih gena. Takva ab initio anotacija gena (možda dodatno opisat što je anotacija gena), ukoliko je uspješna, može biti bolja od anotacije gena na bazi dokaza o njihovoj ekspresiji koja može propustiti neke gene zbog utišane ekspresije.

Cilj ovog rada je implementirati model duboke neuronske mreže koji ima sposobnosti ab initio indentifikacije gena na širem genomu te opisati biološke i tehnološke koncepte koji stoje iz njegove implementacije

1. Biološki kontekst

1.1 DNA i RNA

DNA (deoksiribonukleinska kiselina) i RNA (ribonukleinska kiselina) su biopolimeri na kojima se zasniva život. DNA (prikazana desno na Sl. 1.1) služi kao glavno spremište instrukcija za razvoj i funkciju svih poznatih živih organizama. DNA je velika molekula sastavljena od ponavljajućih jedinica nukleotida. Svaki nukleotid sastoji se od tri glavna dijela: dušične baze, petoatomnog šećera (deoksiriboza) i barem jedne fosfatne skupine. Dušične baze dolaze u četiri varijante - adenin (A), citozin (C), guanin (G) i timin (T). U strukturi DNA, citozin uvijek formira par s guaninom, a adenin s timinom. Ti lanci čine dvostruku uzvojnicu DNA, stabilnu strukturu koja omogućuje pohranu velike količine informacija u malom prostoru unutar stanice.



Sl. 1.1: Prikaz strukture DNA I RNA

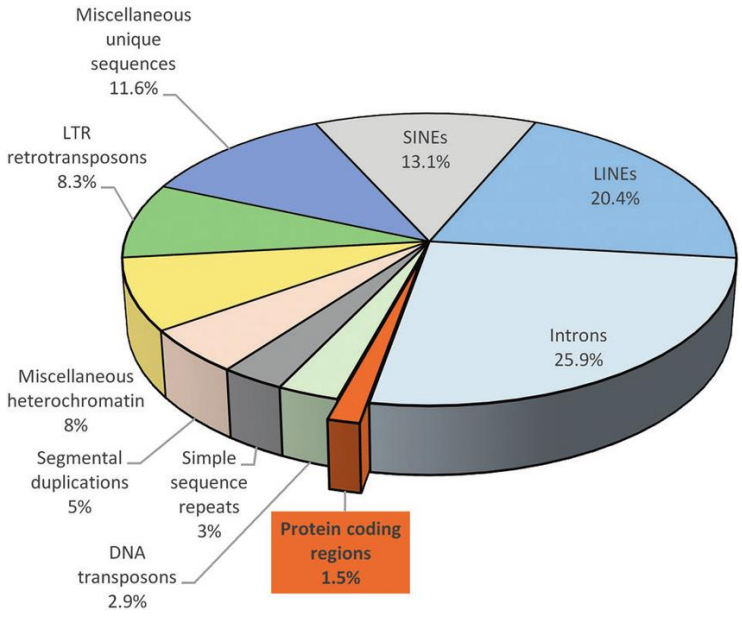
S druge strane RNA (prikazana na lijevoj strain Sl. 1.1) ima ključnu ulogu u dekodiranju, regulaciji i ekspresiji tih genetskih instrukcija. RNA se, kao i DNA, sastoji od nukleotida, ali za razliku od DNA je građena od jednog lanca i građena je od šećera pentoze. Nukleotidne baze koje tvore lanac RNA su adenin, citozin, gvanin i uracil, koje označavamo sa slovima A, C, G i U . Postoji nekoliko vrsta RNA molekula koje dijelimo ovisno o njihovim funkcijama. Najvažnije su glasnička RNA

(mRNA), transportna RNA (tRNA), ribosomska RNA (rRNA) te regulacijske RNA poput mikro RNA (miRNA), male jezgrene RNA (snRNA) i male interferirajuće RNA (siRNA).[1]

1.2 Genom

Genom organizma predstavlja njegov cijeli genetski material te sadrži sve informacije koje su organizmu potrebne za rast i razvoj. Genomi variraju veličinom od ljudskog genoma koji sadrži preko tri milijarde parova baza do genoma bakterije *Mycoplasma genitalium* koji sadrži manje od 600.000 parova baza. Kompletan slijed nukleotida koji čini DNA naziva se sekvenca genoma. Vrlo je slična između pojedinaca iste vrste te varira samo u vrlo malim regijama, koji su većinom u nekodirajućim regijama. U potpunosti su sekvencirani genom više od 34.000 vrsta [2].

Sastav ljudskog genoma prikazan je na Sl. 1.2.



Sl. 1.2: Sastav ljudskog genoma

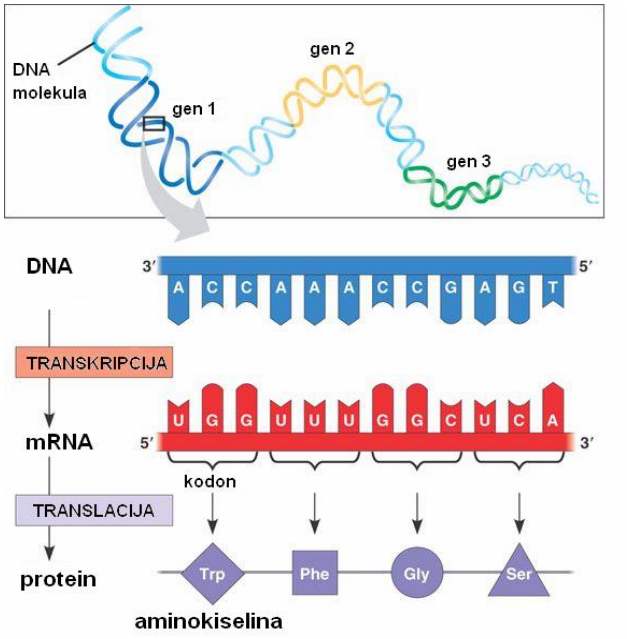
Regije genoma mogu se podijeliti na dvije skupine:

- Sekvence koje ne kodiraju proteine, koje uključuju elemente kao što su introni, pseudogeni, transponibilni elementi, dugački interspersirani nuklearni elementi (LINE), kratki interspersirani nuklearni elementi (SINE), regulatorna DNA i druge elemente s manjim udjelima.

- Sekvence koje kodiraju proteine, koje nose potrebne informacije za sintezu proteina to jest gene

1.3 Sinteza proteina

Geni su temeljne jedinice nasljeđivanja u svim živim organizmima. Oni služe kao nacrti za proizvodnju proteina. Geni variraju u veličini, krećući se od nekoliko stotina do više od dva milijuna parova baza. Sinteza proteina (Sl. 1.3) je centralni događaj u ekspresiji gena i može se podijeliti u dva glavna koraka: transkripciju i translaciju. Transkripcija je proces prepisivanja DNA u mRNA. Kod prokariota, koji nemaju staničnu jezgru, transkripcija gena koji kodiraju proteine stvara mRNA koja je odmah spremna za translaciju. Eukarioti, koji imaju staničnu jezgru, imaju složeniju strukturu gena. Eukariotski gen sadrži nekodirajuće regije (introne) koje stvaraju prostor da se isti gen različitim prespajanjem kodirajućih regija (egzona) koristi za sintezu više različitih proteina. Zbog kompleksnije strukture gena eukariota nakon čitanja primarnog transkripta RNA koji sadrži eksone i introne ta RNA se mora doraditi izrezivanjem introna kako bi postala zrela mRNA. Zrela mRNA se po izlasku iz jezgre spaja na ribosom te započinje process translacije. Translacija je proces pri kojem se na ribosomu gradi protein čitanjem genetskog koda s mRNA sekvence sve dok se ne naiđe na STOP kodon. U procesu translacije je bitna transportna RNA (tRNA) koja služi kao veza između mRNA i aminokiselina od kojih se proizvode proteini.[1]



Sl. 1.3: Ilustracije sinteze proteina

1.4 Anotacija gena

Genomska anotacija je process pronalaska lokacija kodirajućih regija na genomu te određivanje njihovih bioloških funkcija. Anotacija gena određena je strukturalnom i funkcionalnom komponentom. Strukturalni dio anotacije sadrži fizičku lokaciju gena unutar genoma i njegovu konstituciju. Paralelno, funkcionalna anotacija nastoji opisati biološku aktivnost gena i način na koji se izražava u protein. Otkrivanje gena obuhvaća niz tehnika i metoda koje se mogu smatrati korakom u anotaciji gena. Cilj predviđanja gena je pronaći početne i krajnje pozicije gena i drugih funkcionalnih regija genoma. Te informacije se potom mogu koristiti za daljnje proučavanje regije genoma, što vodi do anotacije te regije.

Ab initio ili statističke metode

Ab initio metode nastoje pronaći gene na temelju sastava sekvence koristeći statistički značajne regije u kodirajućim sekvencama poput regija početka i završetka. Ove vrste metoda se oslanjaju na informacije poznate o već poznatim genima. Detektiraju znakove u određenim regijama DNA sekvence, ti znakovi ovise o tipu organizma. U slučaju da je organizam prokariot, zadatak je

relativno jednostavan jer prokarioti nemaju introne i imaju prepoznatljiv promotor dok kod eukariota pronalazak gena je obično teži zbog velikih udaljenosti između eksona i ograničenije količine znanja o promotorima.

Komparativne metode

Komparativne metode koriste informacije prikupljenje iz već dokumentiranih genoma različitih organizama. Komparativne metode funkcioniraju zahvaljujući činjenici da organizmi koji nisu taksonomski jako udaljeni imaju vrlo slične gene. Većina komparativnih metoda započinje poravnanjem sekvenci genoma dva organizma koji se uspoređuju i pokušavaju pronaći sličnosti među njima. Poravnanja mogu biti globalna - koja uspoređuju sekvence po njihovoj punoj duljini ili lokalna - koja traže samo regije visoke sličnosti.[3]

1.4.1 Postojeći alati za predviđanje gena

AUGUSTUS

AUGUSTUS je alat za predviđanje gena u eukariotskim organizmima koji se koristi Generaliziranim skrivenim Markovljevim modelom koji uzima u obzir intrinzične i ekstrinzične infromacije te je obučen na više od sedamdeset organizama uključujući životinje i bakterije. [4]

GENSCAN

GENSCAN je alat koji koristi ab initio metode za indentificiranje eksona i introna u genomu. Koristi algoritme Markovljevog modela petog reda za nekodirajuće regije i nehomogeni triperiodički Markovljev model petog reada za kodirajuće regije. GENSCAN može predvidjeti višestruke gene u sekvenci te rukovati sa djelomičnim ili potpunim genima. Alat prihvaća sekvence do milijun znakova duljine. [5]

CESAR

CESAR je alat koji se temelji na skrivenom Markovljevom modelu koji za razliku od ostalih spomenutih modela koristi komparativne umjesto ab inition metoda. CESAR je brži i manje memorijski zahtjevan of većine drugih alata. Valja spomenuti da iako se CESAR može koristiti za predikciju gena glavni fokus njegove implementacije je uparivanje i poravnanje sekvenci.[6]

2. Duboko učenje

Duboko učenje je područje strojnog učenja inspirirano strukturom i funkcijom ljudskog mozga, revolucioniralo je način na koji računalni sustavi uče i interpretiraju velike količine podataka. Kroz dubinsku neuronsku mrežu, algoritmi dubokog učenja koriste više slojeva za izvođenje složenih oblika obrade podataka. Sposobni su identificirati i učiti obrasce iz podataka na vrlo visokoj razini abstrakcije, što im omogućuje da obavljaju zadatke s velikom točnošću, bilo da se radi o prepoznavanju slika, razumijevanju prirodnog jezika ili predviđanju budućih trendova.

2.1 Umjetne neuronske mreže

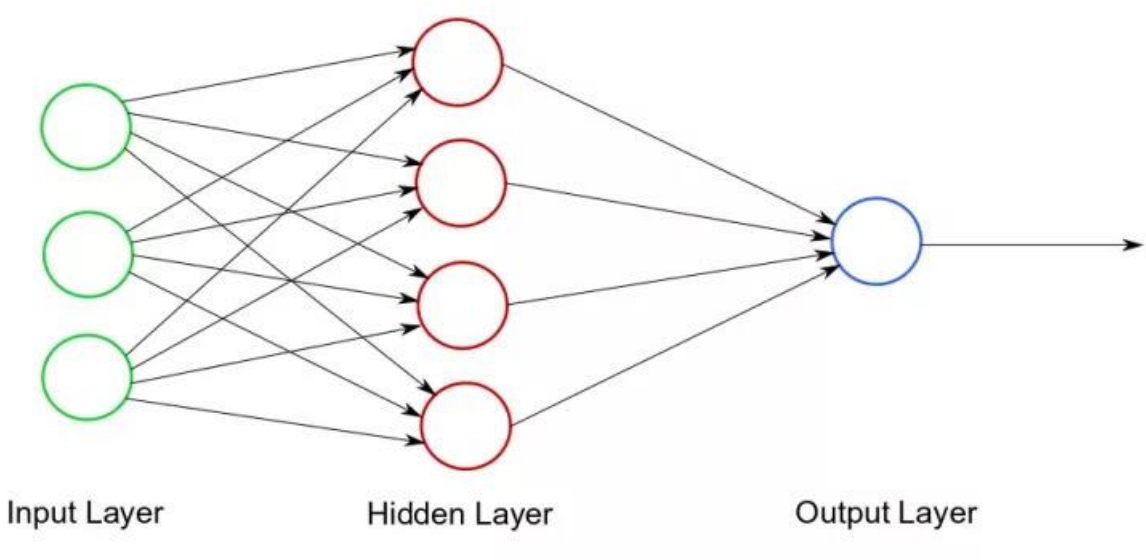
Umjetne neuronske mreže su računalni modeli inspirirani strukturom i funkcijom bioloških neuronskih mreža. Osnovni elementi umjetnih neuronskih mreža su umjetni neuroni, analogno biološkim neuronima. Ovi umjetni neuroni primaju ulazne informacije slično kako dendriti primaju signale od drugih neurona u biološkoj neuronskoj mreži. Primljeni ulazi u umjetnoj

neuronskoj mreži tada se obrađuju koristeći matematičku funkciju, simulirajući proces integracije koji se događa u tijelu biološkog neurona. Ako obrađeni signal premaši određeni prag - oponašajući inicijaciju akcijskog potencijala u biološkom neuronu - umjetni neuron se aktivira i šalje signal dalje. Ovo prosljeđivanje signala slično je načinu na koji akson prenosi električni impuls drugim neuronima. Naposljetku, težine određuju razinu utjecaja jednog neurona na ulaz drugog te predstavljaju jačinu veza između umjetnih neurona slično kako sinaptička snaga funkcionira u biološkim sinapsama. Važna je činjenica da unatoč sličnostima umjetne neuronske mreže su samo pojednostavljena apstrakcija stvarne složenosti bioloških neuronskih mreža [7].

2.2 Osnovna arhitektura umjetne neuronske mreže

Kako bi razumijeli kako umjetna neuronska mreža funkcionira moramo promotriti arhitekturu jednostavne neuronske mreže (Sl. 2.1). Prvi sloj neuronske mreže je ulazni podatak, to jest numerička reprezentacija ulaznog podataka u slučaju da je taj podatak tekst,slika ili neka druga vrsta informacije.Nakon ulaznog sloja slijedi skriveni sloj koji se sastoji od skupa neurona gdje svaki neuron prima ulazne podatke te izvršava matematičku operaciju nad njima, a zatim primjenjuje aktivacijsku funkciju na rezultat. Izlazi dobiveni obradom u skrivenom sloju se potom šalju izlaznom sloju. On sadrži jedan ili više neurona, ovisno o vrsti problema koji se neuronskom mrežom rješava. Na primjer u slučaju regresijskog problema izlazni sloj obično sadrži samo jedan neuron čiji je izlaz kontinuirana numerička vrijednost koja predstavlja predviđeni odgovor za dane ulazne podatke. S druge strane, za probleme klasifikacije broj neurona u izlaznom sloju obično odgovara broju mogućih klasa s iznimkom za binarnu klasifikaciju gdje se koristi samo jedan neuron.

Prikazana neuronska mreža je potpuno-povezana neuronska mreža što znači da je izlaz svakog neurona povezan sa svim neuronima u slijedećem sloju, ujedno je i unaprijedna neuronska mreža jer nema veza koje su povratne ili koje preskaću slojeve



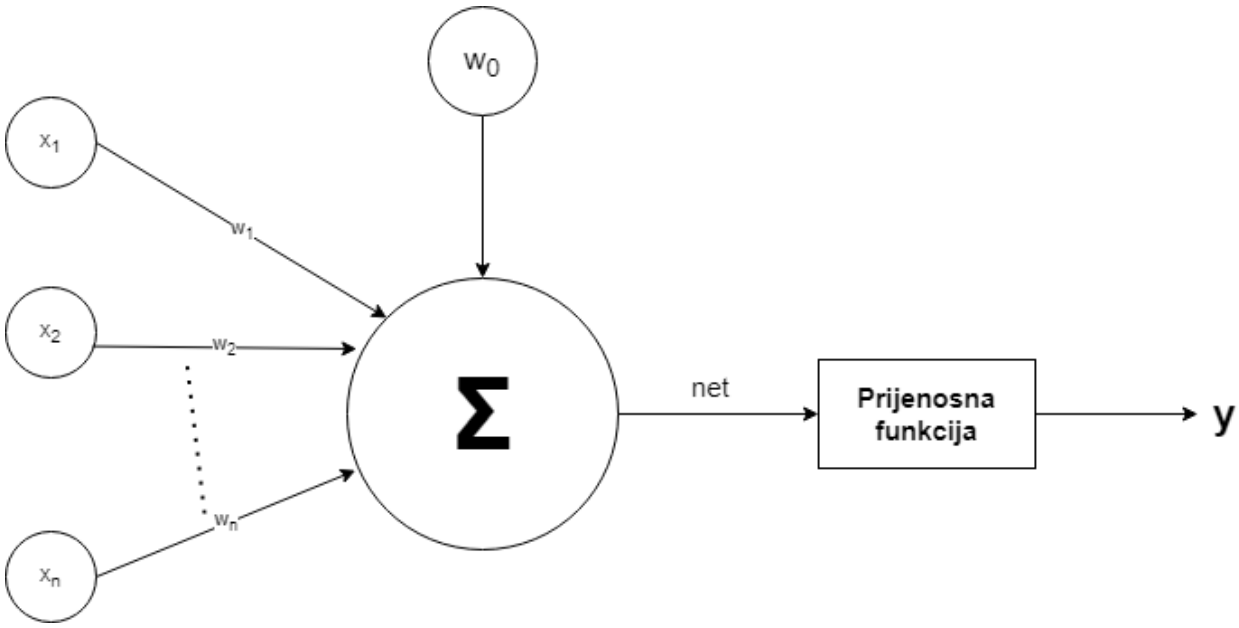
Sl. 2.1: Arhitektura jednostavne neuronske mreže

2.3 Umjetni neuron

Umjetni neuron je temeljni element umjetne neuronske mreže (Sl. 2.2). Umjetni neuron ima niz ulaza označenih sa x_i te se svaki od tih ulaza množi sa svojom težinom w_i. Dobivene vrijednosti se zatim zbrajaju te im se pribraja pomak w_0.

$$\sum_{i=1}^{n} w_i * x_i + w_0 \quad . \tag{1}$$

Prikazani izraz (1) predstavlja jednađbu umjetnog neurona gdje je n broj elemenata tenzora, a na njegov rezultat se primjenjuje aktivacijska funkcija. Rezultat aktivacijske funkcije je izlaz y kao što se vidi na slici [7].

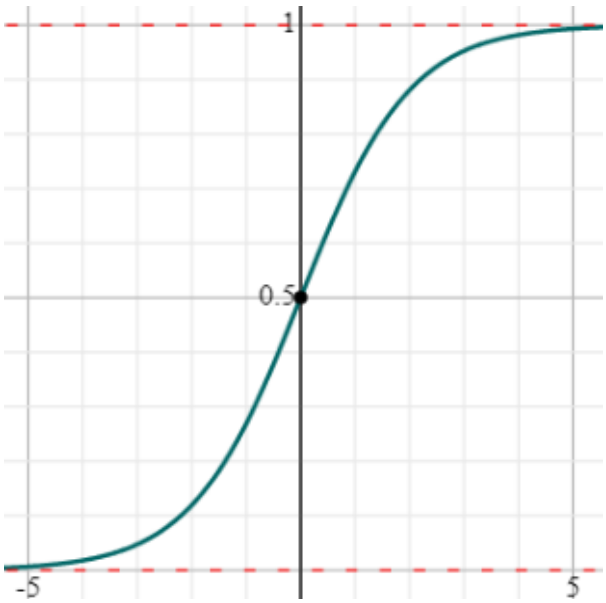


Sl. 2.2: Prikaz umjetnog neurona

2.4 Aktivacijska funkcija

Zadnji korak u obradi ulaza neurona je aktivacijska funkcija (na slici označena kao prijenosna funkcija) koja se primjenjuje na vrijednost dobivenu zbrajanjem umnožaka ulaza i njihovih težina. Odabir aktivacijske funkcije igra veliku ulogu u karakteristikama model te postoji više različitih aktivacijskih funkcija [7].

Sigmoidalna funkcija

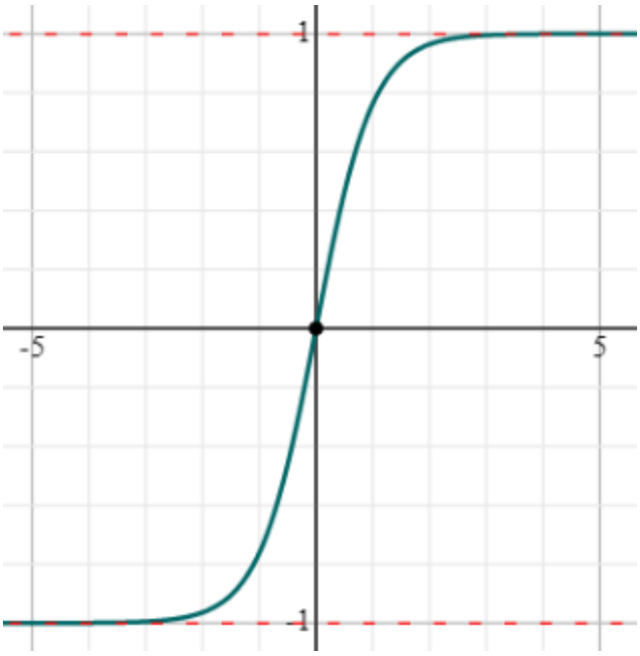


Sl. 2.3: Graf sigmoidalne funkcije

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

Sigmoidalna funkcija (2) je nelinearna aktivacijska funkcija kojoj je domena skup realnih brojeva, a kodomena interval [0,1] što se može se može vidjeti na Sl. 2.3. Nelinearnost sigmoidalne funkcije proširuje spektar problema koji su modeli koji je koriste sposobni riješiti. No sigmoidalna funkcija ima svoje nedostatke. Kada su argumenti sigmoidalne funkcije vrlo velike ili vrlo male vrijednosti velike promjene u ulaznim podacima rezultiraju malim promjenama izlaza što dovodi do problema nestajućeg gradijenta koji otežava učenje.

Tangens hiperbolni



Sl. 2.4: Graf tanh funkcije

$$\tanh(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}} \tag{3}$$

Kao i sigmoidalna funkcija tangens hiperbolni (3) je nelinearna aktivacijska funkcija, no za razliku od sigmoidalne funkcije kodomena joj je [-1,1] (Sl. 2.4) što omogućava lakšu optimizaciju modela. Tangens hiperbolni pati od nestajućeg gradijenta slično kao i sigmoidalna funkcija, ali na manjoj razini zbog svojeg strmijeg gradijenta.

ReLu funkcija

Sl. 2.5: Graf ReLu funkcije

$$\mathrm{ReLu}(x) = \begin{cases} 0 \ , & x < 0 \\ \mathrm{x} \ , & x \geq 0 \end{cases} \tag{4}$$

ReLu (zglobnica) (4) je trenutačno jedna od najčešće korištenih aktivacijskih funkcija u dubokom učenja. ReLu funkcija sve negativne vrijednosti pretvara u nulu i ne mijenja pozitivne vrijednosti što se vidi na Sl. 2.5 . ReLu aktivacijska funkcija je korisna jer ne pati od problema nestajućeg gradijenta za pozitivne vrijednost. No i ReLu ima svoje probleme jer zbog načina na koji funkcija tretira negativne ulaze može doći do problema "umirućeg ReLu" ako je velik broj ulaza negativan.

2.5 Unazadna propagacija greške

Unazadna propagacija greške je metoda optimizacije koja se koristi u dubokom učenju. Ova metoda koristi gradijentni spust kako bi prilagodila težine u neuronskoj mreži, s ciljem minimiziranja greške između stvarnih vrijednosti i izlaznih vrijednosti mreže.

Osnovni algoritam unazadne propagacije greške:

1. Unaprijedno širenje: Ulazni podaci se prosljeđuju kroz mrežu od ulaznog do izlaznog sloja

2. Izračun greške: Na izlaznom sloju mreže se uz pomoč funkcije gubitka računa ukupna greška to jest razlika između stvarne i predviđene vrijednosti.

3. Unazadna propagacija greške: Greška se širi unatrag kroz neuronsku mrežu, od izlaznog sloja prema ulaznom. Izračunava se koliko svaka težina pridonosi ukupnoj grešci. To se postiže izračunom derivacije funkcije gubitka u odnosu na svaku težinu, time se stvara gradijent funkcije gubitka

4. Ažuriranje težina: Težine se ažuriraju koristeći algoritam gradijentnog spusta to jest težine se prilagođavaju u smjeru suprotnom od gradijenta jer je to smjer u kojem najbrže opada greška

Ovaj algoritam se ponavlja kroz određeni broj epoha dok mreža ne prilagodi svoje težine tako da minimizira grešku [7].

2.6 Arhitekture dubokih modela

Duboko učenje ima više različitih arhitektura za rješavanje problema. U ovoj sekciji je opis nekoliko uobičajenih struktura koje su relevantne za model napravljan u svrhu ovog rada.

2.6.1 Višeslojni perceptroni

Višeslojni perceptroni su unaprijedne neuronske mreže sa više od dva skrivena sloja te su jedna od najčešćih arhitektura dubokog učenja. Višeslojni perceptroni koriste nelinearne aktivacijske funkcije kako bi mogli naučiti nelinearne veze između ulaznih i izlaznih podataka, a unazadnu propagaciju greške za ažuriranje težina [2].

2.6.2 Konvolucijske neuronske mreže

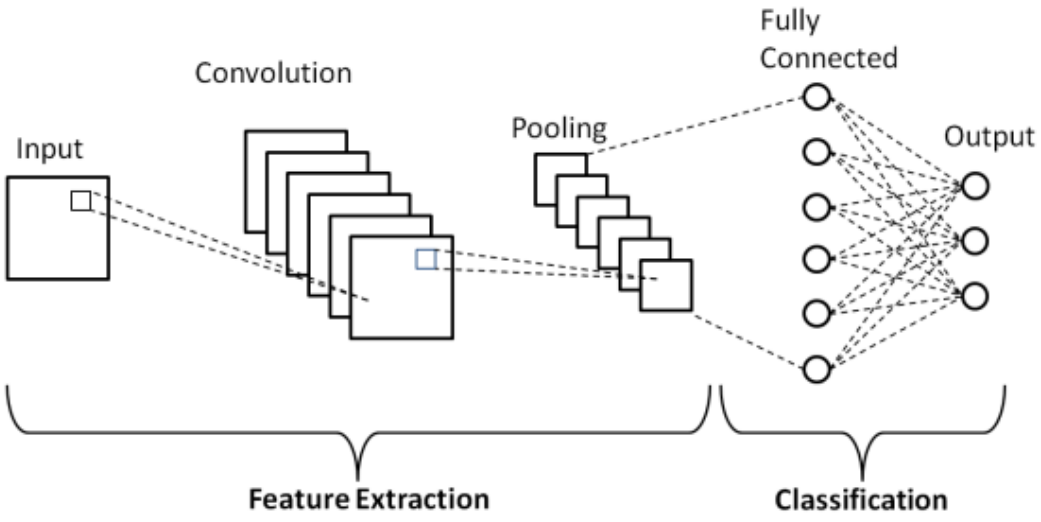
Konvolucijske neuronske mreže su vrsta duboke umjetne neuronske mreže koja se primarno koristi u problemima klasifikacije slika, klasteriranja ili prepoznavanja obrazaca.

Ove mreže imaju primjene u različitim stvarnim područjima, od prepoznavanja prometnih znakova do osnaživanja vida u autonomnim vozilima.

Modeli temeljeni na arhitekturi konvolucijske neuronske mreže mogu imati mnogo različitih oblika, ali obično se temelje na četiri glavna zadatka:

- Konvolucija - ekstrakcija značajki iz ulaznih podataka, Ekstrakcija se provodi filterima značajki koji se primjenjuju na ulazne podatke kako bi stvorili mape značajki
- Uvođenje nelinearnih funkcija nakon svake konvolucije
- Poduzorkovanje – smanjenje dimenzionalnosti mapa značajki što rezultira lakše obradivim reprezentacijama podataka sa smanjenim brojem parametara što smanjuje vjerojatnost prenaučenosti
- Konstrukcija potpuno povezanog sloja, umjetne neuronske mreže s nelinearne funkcije aktivacije koja koristi rezultate poduzorkovanja kako bi klasificirala naučene podatake.

Sl. 2.6 prikazuje arhitekturu konvolucijske neuronske mreže koja se temelji na opisanim zadacim [8].



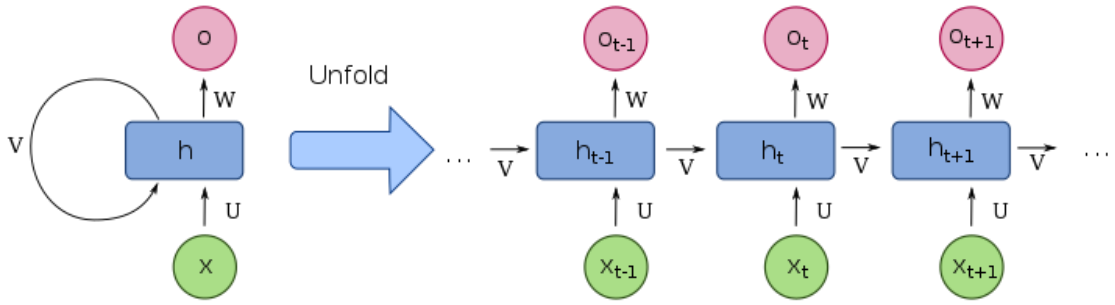
Sl. 2.6: Arhitektura konvolucijske neuronske mreže

2.6.3 Rekurentne neuronske mreže

Rekurentne neuronske mreže su vrsta arhitekture koja se uobičajeno koristi za probleme sa sekvencijalnim podacima kao što su prepoznavanje govora i prirodni jezik.

Raspored slojeva rekurentne neuronske mreže je sličan rasporedu slojeva u unaprijednoj neuronskom mreži s redoslijedom slojeva: ulazni – skriveni – izlazni. Glavna razlika između rekurentne neuronske mreže i unaprijedne neuronske mreže je mogućnost povratnih veza između čvorova, pri čemu je izračun svakog čvora u skrivenom sloju kombiniran izračun ulazne vrijednosti i informacija iz prethodnih čvorova kao što je prikazano na Sl. 2.7.

Jedan od problema koji se mogu javiti u rekurentnim neuronskim mrežama je problem nestajućeg gradijenta. "Long Short-Term Memory" (LSTM) mreže su varijacija rekurentnih neuronskih mreža dizajnirane da bi se riješio problem nestajućeg gradijenta. Ove mreže mogu "pamtiti" informacije tijekom dužih vremenskih koraka zahvaljujući posebnim strukturama zvanim "vrata". Ta poboljšanja omogućuju LSTM mrežama da efikasnije obrađuju sekvencijalne podatke, čineći ih idealnim za kompleksne sekvencijalne zadatke [9].



Sl. 2.7: Arhitektura rekurentne neuronske mreže

3. Implementacija i Evaluacija

3.1 Podatci

Podaci predstavljaju kritičnu komponentu svakog zadatka dubokog učenja. Kvaliteta i kvantiteta dostupnih podataka može značajno utjecati na performanse modela dubokog učenja. Modeli se treniraju na velikim skupovima podataka kako bi naučili prepoznati **što** složen**ije** obrasce i veze između ulaznih i ciljnih varijabli.

3.1.1 Repozitoriji genetskih podataka.

Kako bi mogali efikasno trenirati i evaluirati model potrebni su kvalitetni i točni podaci. U slučaju genomskih podataka postoji više javnih besplatnih repozitorija s kojih se mogu preuzeti takvi podatci.

GenBank

GenBank je besplatna javna kolekcija DNA sekvenci. Održava se putem online platforme koja omogućava lagan pristup svim sekvencama u njihovoj bazi podataka. GenBank održava NCBI ogranak Nacionalnog instituta za zdravlje Sjedinjenih Američkih Država. GenBank omogućava korisnicima preuzimanje velikih količina raznih genetskih zapisa odjednom.

Ensembl

Projekt Ensemble započeo je 1999. Godine. Stvoren je s namjerom da omogući sveobuhvatnu anotaciju genoma. Danas ima online platformu na koju korisnici mogu predati svoje doprinose pretraživati informacije o određenom položaju u genomu. Opcije koje Ensemble online platforma pruža se prostiru od pretraživanja gena do preuzimanja tekstualnih prikaza genoma.

3.1.2 Formati podataka

Pri radu sa DNA sekvencama uobičajeno se koristi nekoliko popularnih formata genetskih podataka. U ovoj sekciji je opis formata koji su relevantni u kontekstu ovog rada.

FASTA

FASTA format je jedan od najrasprostranjenijih formata u bioinformatici, koristi se za reprezentaciju nukleotidnih (DNA, RNA) ili peptidnih (proteinskih) sekvenci. Format se sastoji od jedne ili više sekvenci te svaka sekvenca započinje zaglavljem (koje počinje simbolom ">") nakon kojeg slijedi niz nukleotida ili aminokiselina. Simplificirana struktura omogućava lako manipuliranje sekvencama, dok široka primjena ovog formata omogućava kompatibilnost s velikim brojem bioinformatičkih alata [1].

GFF

Za razliku od FASTA formata koji se fokusira na same sekvence, GFF (General Feature Format) format se koristi za detaljno opisivanje gena i ostalih značajki na DNA i RNA sekvencama, ili proteinima. GFF format predstavlja tablični prikaz podataka sastavljen od devet kolona koje pružaju informacije o različitim aspektima svake značajke, uključujući njen položaj, tip, izvor, među ostalim atributima. Omogućuje složeniju analizu i interpretaciju genetskih podataka, uključujući lokalizaciju gena, anotaciju egzona i introna, kao i druge genomske značajke.

3.1.3 Priprema podataka

Sekvenca genoma i odgovarajuća anotacija gena za *Salmonellu enterica* su preuzeti u formatima GFF i FASTA. Prvi korak je učitavanje tih podataka. Pritom su ekstrahirane informacije o genima iz GFF datoteke, gdje je svaki gen opisan početkom i krajem njegove pozicije na genomu.

Zatim je učitana sekvenca genoma iz FASTA datoteke. Sekvenca je transformirana u numerički format koristeći "one-hot encoding" tehniku, gdje svaki nukleotid ima odgovarajuću jedinstvenu binarnu reprezentaciju.

Paralelno je kreirana i oznaka za svaku poziciju u genomu pritom je svaka pozicija označena kao 0 ili 1, gdje 1 označava da se na toj poziciji nalazi nukleotid koji je dio kodirajuće regije. Oznake su kreirane usporedbom položaja gena (dobivenih iz GFF datoteke) s dužinom genomne sekvencije.

Nakon toga, genom i odgovarajuće oznake su podijeljeni na blokove od 500 nukleotida, s korakom od 250 nukleotida. Time se stvara prozor koji klizi preko genoma, što omogućava dubinsko učenje na podacima.

Svaka sekvenca i odgovarajuće oznake su zatim spojeni u parove i pomiješani kako bi se osigurala nasumična distribucija podataka.

Na kraju, skup podataka je podijeljen na skup za treniranje i skup za testiranje u omjeru 80:20. Ova podjela omogućava validaciju modela na podacima koji nisu korišteni tijekom treniranja.

3.2 Biblioteke za duboko učenje

Biblioteke za duboko učenje predstavljaju korisnu osnovu za razvoja modela dubokog učenja jer nude veliki broj funkcionalnosti na različitim razinama kompleksnosti.

Na visokoj razini ovi alati pružaju mogućnost brzog i efikasnog eksperimentiranja, s njima je moguće u nekoliko linija koda stvoriti modele složenih arhitektura.

Na nižoj razini omogućuju finu kontrolu nad svim aspektima modela i procesa učenja, što je korisno za istraživačke projekte i specifične primjene.

TensorFlow

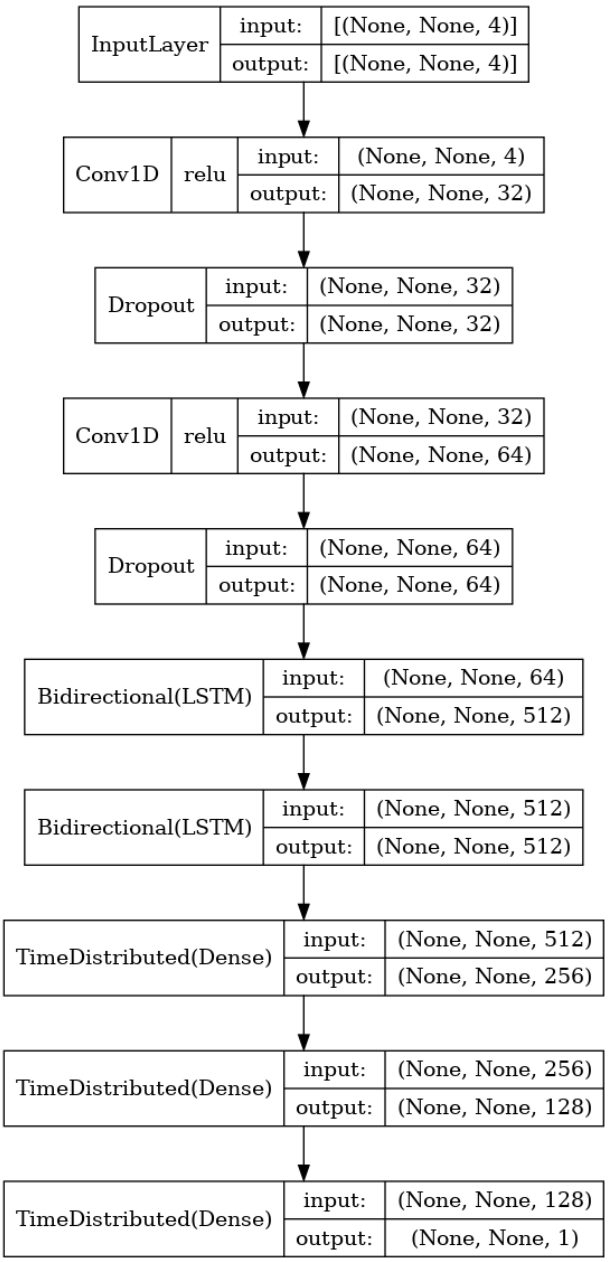
TensorFlow je besplatna open-source biblioteka koju je razvio Google. Osim što nudi alate za strojno učenje nudi i opsežnu matematičku biblioteku za druge svrhe znanstvene analize podataka. Koristi structure koje se zovu tenzori koji simuliraju skalare, vektore i matrice te omogućava matematičke operacije između njih. TensorFlow je jedna od najčešće korištenih biblioteka za duboko učenje unatoč tome što se pokazala sporijom od drugih.[10]

Keras

Keras je otvorena biblioteka za duboko učenje koja radi na osnovi TensorFlow-a. Izvorno je razvijen kao korisnički orijentirano sučelje za ublažavanje složenosti TensorFlow biblioteke, s ciljem omogućavanja brzog prototipiranja. Keras podržava sve osnovne tipove modela dubokog učenja - potpuno povezane, konvolucijske i rekurentne mreže, kao i njihove kombinacije. Nudi i mnoštvo alata za rad s podacima, treniranje modela, procjenu učinkovitosti i optimizaciju. Ova biblioteka je posebno popularna zbog svoje jednostavnosti i pristupačnosti, što je čini idealnom za početnike u području dubokog učenja.[11]

3.3 Model

Implementirani model(Sl. 3.1) je duboka neuronska mreža koja kombinira arhitekture konvolucijskih neuronskih mrežo i rekurentnih neuronskih mreža. Zadatak mreže je indentifikacija kodirajućih sekvenci u genomu. U ovoj sekciji će biti objašnjena struktura modela.



Sl. 3.1: Arhitektura implementiranog modela

3.3.1 Konvolucijski Slojevi

Model počinje s dva konvolucijska sloja. Ovi slojevi koriste jednodimenzionalne konvolucije koje su posebno korisne za sekvencijske podatke. Prvi konvolucijski sloj ima 32 filtera i veličinu jezgre od 9. Ovaj sloj može identificirati lokalne uzorke u sekvenciji koji obuhvaćaju 9 uzastopnih točaka podataka. Nakon prvog konvolucijskog sloja, koristimo Dropout sloj koji nasumično postavlja neke ulaze na nulu kako bi spriječio prenaučenost modela. Drugi konvolucijski sloj koristi 64 filtera kako bi uhvatio složenije uzorke u podacima. Oba konvolucijska sloja koriste aktivacijsku funkciju ReLU, koja omogućava modelu da uhvati nelinearne obrasce u podacima.

3.3.2 Slojevi s Dugoročnom Kratkoročnom Memorijom (LSTM)

Nakon konvolucijskih slojeva, koristimo dvosmjerni LSTM sloj. Ovi slojevi su dizajnirani za rad s sekvencijskim podacima, i mogu zapamtiti uzorke tijekom dužih sekvenci. Upotrebom dvosmjernosti, LSTM obrađuje sekvenciju u oba smjera, što znači da može koristiti kontekstualne informacije kako iz prošlosti tako i iz budućnosti. Nakon toga, dodajemo još jedan LSTM sloj koji nastavlja proces hvatanja vremenskih ovisnosti u sekvenci podataka.

3.3.3 Potpuno Povezani Slojevi

Nakon LSTM slojeva, naš model koristi seriju potpuno povezanih slojeva. Koristimo TimeDistributed omotač kako bi primjenili potpuno povezanih sloj na svaki vremenski korak u sekvenci. Prva dva Dense sloja imaju, redom, 256 i 128 neurona i koriste ReLU aktivacijsku funkciju.

3.3.4 Izlazni Sloj

Na posljetku, model koristi Dense sloj s jednim neuronima i sigmoidalnom aktivacijskom funkcijom. Sigmoidalna funkcija će ograničiti izlaz između 0 i 1, što je idealno za zadatke binarne klasifikacije na primjer pripada li nukleotid kodirajućoj sekvenci ili ne.

3.3.5 Hiperparametri

Hiperparametri su konfiguracijske varijable koje određuju strukturu modela i kako će se model trenirati. U razvoju modela dubokog učenja je odabir pravih hiperparametara ključan za postizanje optimalne učinkovitosti

Dimenzije Ulaznih Podataka

Prvi bitan hiperparametar su dimenzije ulaznih podataka. U našem modelu svaki ulaz je sekvenca od 500 točaka podataka, svaka sa 4 značajke koje korespondiraju nukleotidima A,C,G,T

Optimizator, stopa učenja I funkcija gubitka

Optimizator modela je Adam – učinkovita metoda stohastičke optimizacije. Stopa učenja kontrolira koliko se brzo model prilagođava podacima te je u modelu 0,001 što je uobičajena stopa učenja za Adam. Korištena funkcija gubitka je binarna unakrsna entropija koja je prikladna za probleme binarne klasifikacije.

Metrika

Metrika koja je korištena za evaluaciju učinkovitosti modela je točnost. Točnost je postotak točno klasificiranih uzoraka te je praktična jer je jednostavna za interpretaciju.

Broj Epoha i veličina skupa

Model treniramo kroz 100 epoha pritom epoha predstavlja jedan prolazak kroz cijeli set podataka za treniranje, a veličina skupa je 32, što znači da model ažurira težine svaki put kad obradi 32 uzorka.

3.4 Evaluacija modela

Kako bi model bio koristan mora moći dobro predviđati gene na podacima na kojima nije treniran. U ovoj sekciji će biti opisani podaci na kojima se testira model i metrike prema kojima će rezultati biti evaluirani

3.4.1 Podaci za evaluaciju

Model se evaluira na genomu *Escherichie coli.* Proces obrade genomskih podataka *E. coli* započinje ekstrakcijom punog genoma *E.coli* iz FASTA datoteke i pozicija svih gena unutar genoma iz GFF datoteke. Potom se genom enkodira "one-hot encoding" metodom. Nakon pripreme podataka, genomski niz i označeni niz se dijele na blokove fiksne veličine (500 baza u ovom slučaju) s pomakom (250 baza u ovom slučaju).

Zatim se testni set preoblikuje u format prihvatljiv za model i pokreće se proces predviđanja. Model za svaki blok genoma daje predviđanje koje se zatim dodaje u konačnu predikciju koja predstavlja predviđanje položaja gena duž cijelog genoma.

3.4.2 Rezultati evaluacije

U ovom dijelu analiziramo rezultate našeg modela za identifikaciju gena temeljenog na neuronskim mrežama. Model je evaluiran na testnom skupu podataka koristeći nekoliko metrika, uključujući točnost, preciznost, odziv, F1 ocjenu i negativnu prediktivnu vrijednost.

Prije nego što predstavimo konkretne rezultate, važno je razumjeti što svaka od ovih metrika mjeri:

Točnost mjeri omjer ispravno predviđenih instanci u odnosu na ukupan broj instanci. Dok je ova metrika korisna, ona može biti zavaravajuća u slučaju nebalansiranih klasa, što je često slučaj u biološkim skupovima podataka.

Preciznost je omjer ispravno pozitivno predviđenih instanci u odnosu na ukupan broj instanci koje su predviđene kao pozitivne. Ova metrika je posebno važna u situacijama gdje je važnije imati manje lažno pozitivnih rezultata nego lažno negativnih.

Odziv je omjer ispravno pozitivno predviđenih instanci u odnosu na ukupan broj stvarnih pozitivnih instanci. Ova metrika je važna u situacijama gdje je bitno prepoznati što više stvarnih pozitivnih instanci.

F1 ocjena je harmonijski prosjek preciznosti i odziva. Ona kombinira oba ova aspekta u jednu metriku koja daje jednaku važnost oba aspekta.

Negativna prediktivna vrijednost(NPV)je omjer ispravno negativno predviđenih instanci u odnosu na ukupan broj instanci koje su predviđene kao negativne. Ova metrika je korisna u situacijama gdje je važno imati manje lažno negativnih rezultata.

Tablica 3.1: Tablica metrika korištenih za evaluaciju

	Točnost	Preciznost	Odziv	F1 ocjena	NPV
Salmonella enterica	0.9748	0.9894	0.9823	0.9857	0.9098
Escherichia coli	0.9040	0.9306	0.9610	0.9417	0.5969

Usporedba između jednog predviđanja kodirajućih sekvenci od strane modela i stvarnih kodirajućih sekvenci na istom segment DNA. (Sl. 3.2 i Sl. 3.3)

```
GGCGTTXTAXXCATAXXXXCTCCTTXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXTTAXXCCGTAATCTGATCCATG
GCCTGTAAAATACGCTTATCGGAAATAGGATAAGGCGTACCGAGCTGTTGGGCGAAATAACTGACACGCAGTTCTTCAATCATCCAACGGATCTCTTTA
ACATCATCATCTTCGCGACGCGCTGGAGGCAATTTATTTATCCATTGTTGCCACGCCTGCTGGACGCTTTCGACTTTCAGCATTTGCGCCCGATCGCGG
TGCGGATCGACAGCCAGTTTCTCCAGCCGCTTTTCAATTGCCTGCAGATAACGCAGCGTATCGCCGAGACGTTTAAAGCCGTTGCCAGTCACAAAACCA
CGGTATACCAGACCGCTCATCTGCGCCTTGATATCGGAAAGCCCCAACGCCATGCTCATATCCACCCGCCCTTTCAGACGCTTATTGATATTGAATACG
GTCGT
```

Sl. 3.2: Predviđene kodirajuće sekvence

```
GGCGTTGTATTCATAATTCCTCCTTXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXTTAGGCCGTAATCTGATCCATG
GCCTGTAAAATACGCTTATCGGAAATAGGATAAGGCGTACCGAGCTGTTGGGCGAAATAACTGACACGCAGTTCTTCAATCATCCAACGGATCTCTTTA
ACATCATCATCTTCGCGACGCGCTGGAGGCAATTTATTTATCCATTGTTGCCACGCCTGCTGGACGCTTTCGACTTTCAGCATTTGCGCCCGATCGCGG
TGCGGATCGACAGCCAGTTTCTCCAGCCGCTTTTCAATTGCCTGCAGATAACGCAGCGTATCGCCGAGACGTTTAAAGCCGTTGCCAGTCACAAAACCA
CGGTATACCAGACCGCTCATCTGCGCCTTGATATCGGAAAGCCCCAACGCCATGCTCATATCCACCCGCCCTTTCAGACGCTTATTGATATTGAATACG
GTCGT
```

Sl. 3.3: Stvarne kodirajuće sekvence

4. Zaključak

Cilj ovog rada bio je implementacija i evaluacija modela dubokog učenja za identifikaciju kodirajućih regija u genomu. Kroz proces razvoja modela, testirao sam razne konfiguracije hiperparametara i arhitektura. Arhitektura implementiranog modela kombinira konvolucijske neuronske mreže i rekurentne neuronske mreže, koristeći prednosti obje tehnike za analizu sekvencijalnih podataka. Kroz brojne iteracije i prilagodbe, model je uspio postići relativno zadovoljavajuću točnost u prepoznavanju kodirajućih sekvenci na genomu Javno dostupni repozitoriji genetskih podataka su omogućili da se model trenira na stvarnim genomskim sekvencama. Iako je implementirani model pružio korisne rezultate, još uvijek postoji prostor za poboljšanja. Budući rad može istražiti upotrebu različitih arhitektura modela, kao i integraciju dodatnih vrsta genetskih podataka.

5. Literatura

[1] Šikić,M.,Domazet-Lošo,M.,Skripta iz Bioinformatike, FER,Zagreb,2013.

[2] Martins, P. V. L. (2018). Gene prediction using deep learning (Magistarski rad). FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO. Mentor: Silva, R. C. C. d. S. F.

[3] Yandell, M., & Ence, D. (2012). A beginner's guide to eukaryotic genome annotation. Nature reviews. Genetics, 13(5), 329–342. https://doi.org/10.1038/nrg3174

[4] Stanke, M., & Morgenstern, B. (2004). AUGUSTUS: a web server for gene prediction in eukaryotes that allows user-defined constraints. Nucleic Acids Research, 32(Web Server issue), W465-W467. DOI: 10.1093/nar/gkh379

[5] Burge, C., & Karlin, S. (1997). Prediction of complete gene structures in human genomic DNA. Journal of Molecular Biology, 268(1), 78-94. DOI: 10.1006/jmbi.1997.0951

[6] Baeza-Centurion, P., & Stormo, G. D. (2019). CESAR: a versatile method for the prediction of cis-regulatory modules. Bioinformatics, 35(10), 1679-1686. DOI: 10.1093/bioinformatics/bty859

[7] Haykin, S. (2009). Neural Networks and Learning Machines. Pearson. ISBN: 978-0131471399

[8]Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep Learning. MIT Press. Dohvaćeno s http://www.deeplearningbook.org

[9] Ilya Sutskever, Training Recurrent Neural Networks, Dissertation, Toronto, Ont., Canada, Canada, 2013, ISBN 978-0-499-22066-0, AAINS22066.

[10] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey

Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al.

Tensorflow: a system for large-scale machine learning. In OSDI, volume 16, pages 265–283, 2016.

[11] Keras documentation. Dostupno na https://keras.io/. Pristupljeno: lipanj. 2023

Sažetak

Abstract