



Laboratorij za analizu teksta i inženjerstvo znanja

Text Analysis and Knowledge Engineering Lab

Sveučilište u Zagrebu, Fakultet elektrotehnike i računarstva

Unska 3, 10000 Zagreb, Hrvatska



Zaštićeno licencijom

Creative Commons Imenovanje-Nekomercijalno-Bez prerada 3.0 Hrvatska

<https://creativecommons.org/licenses/by-nc-nd/3.0/hr/>

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 4752

Primjena modela dubokog učenja na analizu sentimenata

Bruno Gavranović

Zagreb, lipanj 2016.

Zagreb, 11. ožujka 2016.

ZAVRŠNI ZADATAK br. 4752

Pristupnik: **Bruno Gavranović (0036478324)**
Studij: Računarstvo
Modul: Računarska znanost

Zadatak: **Primjena modela dubokog učenja za analizu sentimenta**

Opis zadatka:

Duboko učenje naziv je za skup algoritama strojnoga učenja koji kroz višeslojnu obradu podataka modeliraju apstrakcije visoke razine. Ti su se modeli pokazali iznimno učinkoviti na nizu zadataka obrade prirodnog jezika. To se posebno odnosi na metode temeljene na povratnim neuronskim mrežama, koje se u posljednje vrijeme preuzele vodstvo nad tradicionalnim pristupima u analizi teksta. U okviru završnoga rada potrebno je upoznati se s osnovama neuronskih mreža, a zatim proučiti modele dubokog učenja temeljene na tradicionalnim povratnim neuronskim mrežama (eng. recurrent neural networks, RNN), modele dugog kratkoročnog pamćenja (engl. long short term memory, LSTM) te modele s propusnim povratnim jedinicama (gated recurrent units, GRU). Upoznati se s problemom analize sentimenta u korisnički generiranom sadržaju. Razviti i implementirati model za analizu sentimenta temeljen na dubokom učenju te ga primijeniti na prikladne skupove podataka na hrvatskome i engleskome jeziku. Provesti eksperimentalno vrednovanje modela, uključivo usporedbu s referentnim modelom i statističku obradu rezultata. Radu priložiti izvorni i izvršni kod razvijenog sustava, označene skupove podataka i potrebnu dokumentaciju te citirati korištenu literaturu.

Zadatak uručen pristupniku: 18. ožujka 2016.

Rok za predaju rada: 17. lipnja 2016.

Mentor:

Doc. dr. sc. Jan Šnajder

Djelovođa:

Doc. dr. sc. Tomislav Hrkać

Predsjednik odbora za
završni rad modula:

Prof. dr. sc. Siniša Srblić

Neizmjerne zahvale dugujem Ljudmili i Đurđi jer su bile nesebične u našim druženjima u sitnim noćnim satima ;)

SADRŽAJ

1. Uvod	1
2. Umjetne neuronske mreže	3
2.1. Motivacija	3
2.2. Uvod u neuronske mreže	4
2.2.1. Visoka razina apstrakcije	4
2.2.2. Kratak pregled povijesnog razvoja	4
2.3. Relevantni koncepti	5
2.3.1. Neuron i kompozicije neurona	6
2.3.2. Prijenosna funkcija	8
2.3.3. Funkcija gubitka	9
2.3.4. Euklidski prostor težina i metode optimizacije	10
2.3.5. Algoritam unazadne propagacije	13
3. Vrste neuronskih mreža	15
3.1. Tradicionalne neuronske mreže	15
3.2. Povratne neuronske mreže	16
3.2.1. Mreže LSTM	19
4. Analiza sentimenata i korišteni alati	22
4.1. Problem analize sentimenata	22
4.2. Skup podataka	23
4.3. Analiza i obrada podataka	23
4.3.1. Vektorska reprezentacija riječi	25
4.3.2. Obrada recenzija	26
4.4. Korišteni alati	27
5. Eksperimenti, evaluacija i analiza	28
5.1. One-hot reprezentacija	29

5.2. Distribuirano predstavljanje riječi	30
5.2.1. Pretraživanje po rešetki	31
5.3. Analiza	33
5.3.1. Vizualizacija aktivacija skrivenog sloja	33
6. Zaključak	38
Literatura	39
A. Daljni razvoj neuronskih mreža	42

1. Uvod

Eksplozijski rast količine podataka koje čovječanstvo generira ima neka zanimljiva svojstva. Dnevne količine generiranih podataka danas se mjere u *trilijunima* bajtova dnevno.¹ Količina podataka kojih generiramo *danas* je veća od količine podataka generiranih ukupno do godine 2000., a brzina, pa čak i akceleracija kojom generiramo nove podatke samo nastavlja rasti.

Svi se ti podaci, iz naših osobnih profila na internetu, tekstova, narudžbi, elektroničkih poruka, pa čak i znanstvenih radova nagomilavaju u raznim bazama podataka. Tek se nedavno shvatilo da u svim tim podacima postoje određene *strukture, uzorci i pravilnosti* koji nam mogu otkriti puno o svijetu u kojem se nalazimo i sustavima koje smo stvorili.

Iako informacije postoje, skrivene duboko u podacima, mi ne raspoložemo resursima niti efikasnim načinima obrade takvih količina podataka. Standardni algoritmi, iako korisni za veliki skup slučajeva, za ovakvu su vrstu posla beznadni. Oni traže strogo definiran zadatak i, ako imaju mogućnost pronalaska uzoraka, ti uzorci trebaju biti točno precizirani. U ovom slučaju, ponekad niti sami ne znamo kakve uzorke tražimo niti kako da ih eksplicitno definiramo.

Korak u pravom smjeru je rad na novonastalom području nazvanom *znanost o podacima* (engl. *data science*). To je interdisciplinarno područje znanosti kojem je cilj "izvaditi" znanje i informacije iz velike količine podataka. Ono što to područje razlikuje od ostalih je činjenica da *semantika podataka nije bitna*. Nije cilj riješiti specifičan problem iz određene domene, nego generalizirati proces učenja i pronalaženja uzoraka na način da bude primjenjiv na bilo kakvom skupu podataka. Područje je usko povezano sa *strojnim učenjem* (engl. *machine learning*), granom računalne znanosti fokusiranoj na konkretnoj implementaciji ideja znanosti o podacima. Strojno učenje proučava i stvara algoritme koji mogu učiti na temelju podataka.

Strojno učenje enkapsulira razne pristupe rješavanju problema, od kojih se mogu navesti poznatiji: stabla odluke (engl. *decision trees*), grupiranje (engl. *clustering*),

¹<http://www-01.ibm.com/software/data/bigdata/what-is-big-data.html>

neuronske mreže, strojevi s potpornim vektorima (engl. *support vector machines*), duboko učenje (engl. *deep learning*) i genetski algoritmi.

Svaki od tih algoritama je jedna velika cjelina koja je dovoljno složena sama po sebi. Duboke neuronske mreže su nedavno postale iznimno popularne zbog svoje moći modeliranja apstrakcija visoke razine. Modeli su se pokazali iznimno učinkovitim na nizu zadataka obrade slika, prirodnog jezika i zvuka.

To se posebno odnosi na metode temeljene na povratnim neuronskim mrežama (engl. *recurrent neural networks*), koje se još konkretnije mogu implementirati kao modeli dugog kratkoročnog pamćenja (engl. *long short-term memory, LSTM*) te kao modeli s propusnim povratnim jedinicama (engl. *gated recurrent units, GRU*).

U ovom radu cilj je bio dobiti bolje razumijevanje dubokih neuronskih mreža i njihovog procesa učenja kroz implementaciju rješenja za konkretan problem.

Konkretan problem koji je rješavan je analiza sentimenata u korisnički generiranom sadržaju. Analiza sentimenata odnosi se na korištenje obrade prirodnog jezika, analize teksta i računalne lingvistike za identifikaciju i klasifikaciju subjektivne informacije u izvornim tekstovima. Cilj analize sentimenata je klasificirati razne vrste korisnički generiranih sadržaja u nekoliko kategorija (pozitivni, negativni, neutralni). Izvor podataka koji se analizirao je skupina korisničkih recenzija na filmove iz internetske baze filmova (engl. *Internet Movie Database, IMDb*). Recenziju predstavlja kratak tekst od 100-300 riječi koji prirodnim jezikom opisuje pozitivan ili negativan doživljaj filma. Svako recenziji je pridružena ocjena 1-10 koja predstavlja subjektivni dojam. Ocjena 10 predstavlja iznimno pozitivan dojam, dok ocjena 1 predstavlja iznimno negativan dojam filma.

Konačan implementacijski cilj je razviti model koji bi trebao biti sposoban procijeniti je li recenzija pozitivna ili negativna, samo na temelju ulaznog teksta.

Taj cilj u sebi obuhvaća više komponenti: upoznavanje s osnovama neuronskih mreža, proučavanje modela dubokog učenja, a pogotovo LSTM modela, upoznavanje s alatima potrebnim za pokretanje simulacije i, možda najbitnije, upoznavanje s problematikom primjera koji se rješava (eksplorativna analiza podataka i čišćenje podataka).

U ostatku rada opisana je teorijska podloga neuronskih mreža, razmotrene su različite arhitekture dubokih mreža i detaljno je opisan skup podataka. Provedeni su eksperimenti i napravljena je evaluacija dobivenih rezultata. Doneseni su zaključci o ograničenjima performansi dubokih neuronskih mreža i ograničenju našeg razumijevanja istih.

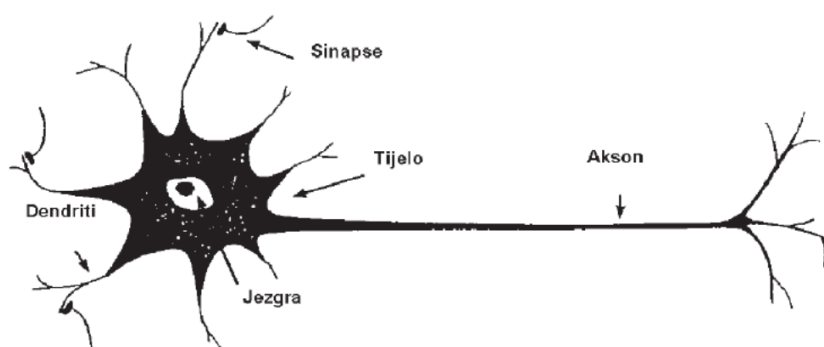
2. Umjetne neuronske mreže

2.1. Motivacija

Uvod u umjetne neuronske mreže ne bi bio potpun bez obaveznog spominjanja analogije s ljudskim mozgom.

Ljudi su oduvijek bili fascinirani misterioznim načinom rada ljudskog uma. Teorija o načinu rada mozga ima napretek, ali tek nedavnim razvojem znanosti su se neke od tih teorija uspjele usidriti u luku razuma. Danas je općepoznato da je najjednostavniji, a ujedno i najbitniji dio biološke mreže jedan neuron. Neuron se može opisati kao osnovna građivna stanica mozga koja prenosi i procesira informacije pomoću električnih i kemijskih reakcija.

Nedugo nakon njegovog otkrića nastala je ideja simuliranja neurona na "umjetan" način. Njegov cilj je formalno opisati njegov matematički model i interakciju s ostalim neuronima kako bi se na kraju što efikasnije modelirao proces učenja ljudskog mozga.



Slika 2.1: Prikaz strukture jednog neurona¹

Iako je analogija s biološkim neuronskim mrežama primamljiva, postoji i drugi razlog rada na umjetnim neuronskim mrežama. One se mogu smatrati kao prijelazni korak ka točnijem i formalnijem opisu procesa učenja. Biološki mozak fascinira sa

¹<http://www.mathos.unios.hr/mdjumic/uploads/diplomski/dum05.pdf>

svojim sposobnostima učenja, no nemamo razloga pretpostaviti da on implementira ujedno i *najbolji* proces učenja. Poznata je izrečica:

Aeronautical engineering texts do not define the goal of their field as making “machines that fly so exactly like pigeons that they can fool even other pigeons.”

Shodno tome, ostatak rada će biti usredotočen na matematički i formalan opis neuronskih mreža.

2.2. Uvod u neuronske mreže

2.2.1. Visoka razina apstrakcije

Umjetne neuronske mreže su skupina modela za procesiranje podataka inspirirani biološkim neuronskim mrežama. Objedinjuju razna područja znanosti: matematiku, statistiku, računalnu znanost, teoriju informacije i već spomenutu biologiju. Njihov proces "učenja" i stvaranja različitih razina apstrakcija ulaznih podataka je i danas uvelike nepoznat. Zbog njihove iznimne kompleksnosti, postoji više konkurentnih teorija koje pokušavaju objasniti što je to točno duboko učenje, kao što je objašnjeno u sjajnom članku Cristophera Olaha [13].

Prva je neuroznanstvena teorija, koja povlači analogiju s biološkim neuronskim mrežama. Druga je *reprezentacijska teorija* (engl. *representation theory*) koja objašnjava duboko učenje kao transformaciju podataka u visokodimenzionalnom prostoru i tezu da je svaka klasa podataka samo jedna mnogostrukost (engl. *manifold*) u tom prostoru. Jedan pak drugi način je smatranje dubokog učenja procesom optimizacija kompozitnih funkcija. Četvrta teorija je statističke prirode, koja objašnjava neuronske mreže kao pronalaženje skrivenih varijabli. Navedene teorije nisu međusobno isključive, ali predstavljaju veoma različite poglede na duboko učenje.

2.2.2. Kratak pregled povijesnog razvoja

Područje analize umjetnih neuronskih mreža može se na prvi pogled činiti kao novo područje nastalo nedavnim razvojem računala. Ipak, koncept *umjetnih* neuronskih mreža nastao je prije pojave računala i područje je preživjelo barem jednu veliku stagnaciju i nekoliko era. Ideja simuliranja rada neurona nastala je još davne 1943., kada su neurofiziolog Warren McCulloch i matematičar Walter Pitts prvi napisali rad o mogućem radu neurona te modelirali neuronsku mrežu koristeći električne krugove. Iako je bila i tada fascinantna, ideja je u potpunosti zaživjela tek u kasnim osamdesetima.

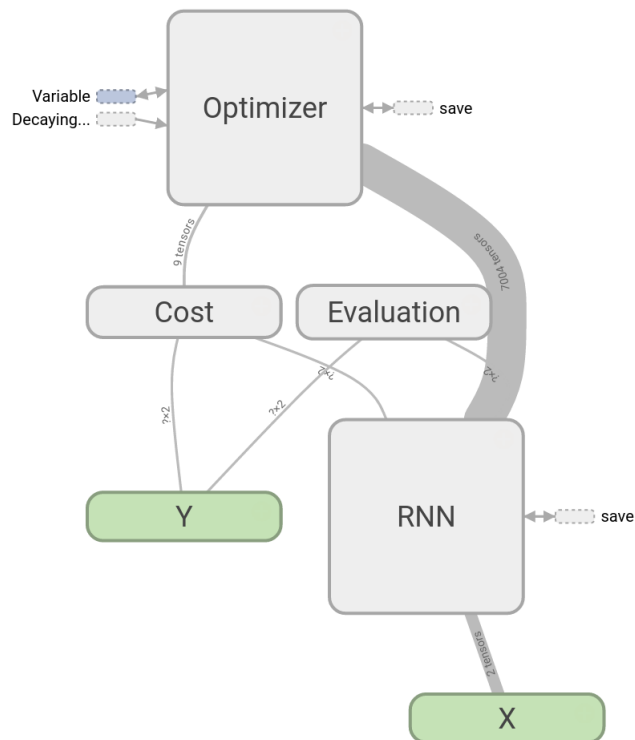
Postoje dva razloga zašto je razvoj neuronskih mreža stagnirao i zašto su *tek danas* popularizirane. Prvi je bio nedostatak postojanja algoritma za brzo treniranje dubokih mreža (algoritam unazadne propagacije, populariziran kasnije), a drugi razlog je bio nedostatak procesorske snage potrebne za računanje parametara velikih neuronskih mreža.

Danas neuronske mreže pokazuju ogroman potencijal te postoje cijela područja koja se bave proučavanjem samo određenih vrsta neuronskih mreža. U industriji imaju veliku primjenu u područjima prepoznavanja lica, govora, prirodnog jezika te, generalno, bilo kakvih uzoraka.

Unatoč uspjesima, unutarnji mehanizmi i ponašanje dubokih neuronskih mreža su i dalje nepoznati te se ulažu ogromni naponi kako bi se pojam neuronskih mreža i njihovog procesa učenja konačno formalizirao.

2.3. Relevantni koncepti

Neuronska mreža može se konceptualno podijeliti na nekoliko različitih komponenti, kako je prikazano na slici 2.2.



Slika 2.2: Neuronska mreža na visokoj razini apstrakcije (generirano u alatu TensorBoard)

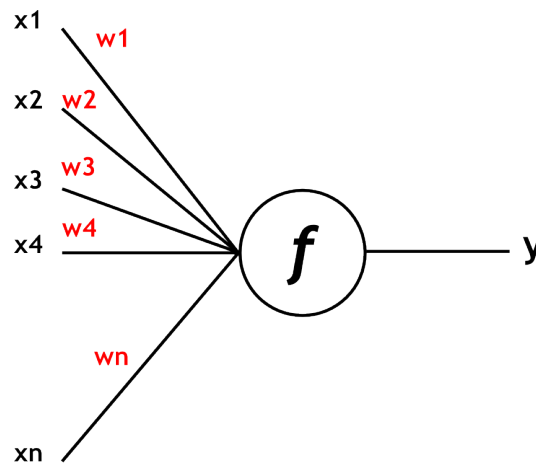
U neuronskoj mreži metoda optimizacije (*Optimizer*) pomoću funkcije gubitka (*Cost*) mijenja neuronsku mrežu (*RNN*) kako bi ona za određen ulaz (*X*) davala izlaz što sličniji stvarnom izlazu (*Y*). Na temelju specifične metode procjene (*Evaluation*) određuje se uspješnost mreže u transformiranju ulaznih podataka u izlazne.

U nastavku su detaljno opisane spomenute komponente i njihova međusobna interakcija.

2.3.1. Neuron i kompozicije neurona

Najjednostavnija građevna jedinica neuronske mreže je jedan neuron. Neuron je jednostavna funkcija koja transformira određen broj ulaznih varijabli u jednu izlaznu varijablu:

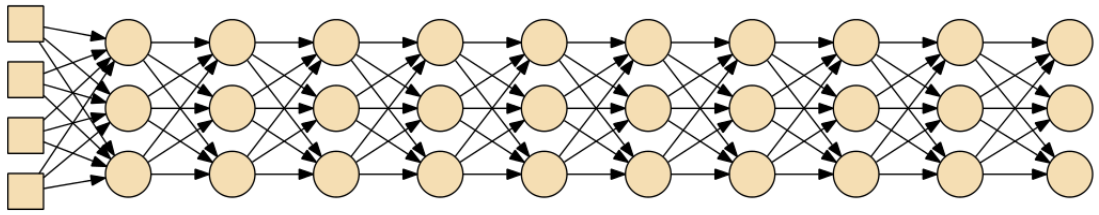
$$f(x_1, x_2, x_3, \dots) = y \quad (2.1)$$



Slika 2.3: Neuron kao funkcija ulaznih impulsa

Konkretna implementacija funkcije može varirati i bit će opisana u nastavku. Ono što čini model neurona moćnim je činjenica da se svaki izlaz iz neurona može smatrati novim ulazom u jedan drugi neuron, te da se cijela jedna skupina neurona može razmatrati modularno, kao *kompozicija funkcija* koje obavljaju različite zadatke.

Na slici 2.4 prikazana su tri neurona, od kojih svaki računa svoju funkciju i prosljeđuje rezultat zadnjem neuronu. Zadnji neuron ne zna da su njegovi ulazi zapravo funkcije "pravih" ulaza, nego ih smatra običnim ulazima, kao i ostali neuroni. Upravo to mu omogućava da modelira apstrakcije više razine nego u slučaju prva tri slučaja.



Slika 2.6: Duboka neuronska mreža³

različitih razina apstrakcija: jednostavniji oblici kao crte i krivulje te kompliciraniji oblici kao što su kotač ili nos.

2.3.2. Prijenosna funkcija

Postavlja se nekoliko pitanja, a to su: kako točno izgleda funkcija neurona? Po čemu se razlikuju neuroni koji modeliraju različite funkcije? Kako neuronska mreža uopće uči? Odgovor se sastoji od dvije komponente.

Prva komponenta je pridjeljivanje težina ulazima u neuron, odnosno, modeliranju neurona jednom težinskom funkcijom koja ima zapis:

$$g(x_1, x_2, x_3, \dots) = \sum_{i=0}^n x_i \cdot w_i = y, \quad (2.2)$$

Funkcija se može zapisati i u matričnom obliku:

$$g(\mathbf{x}) = \mathbf{w}^\top \mathbf{x} = y \quad (2.3)$$

gdje x i w u ovim slučajevima predstavljaju ulazni vektor i vektor težina.

Na jednu funkciju neurona se može gledati kao na *linearnu transformaciju* ulaza. U kontekstu dubokih neuronskih mreža, to znači da je cijela neuronska mreža zapravo kompozicija linearnih transformacija. S druge strane, kompozicija linearnih transformacija samo je još jedna linearna transformacija, a to bi značilo da se cijela duboka neuronska mreža može prikazati samo s jednim slojem. To je u kontradikciji s dosadašnjim poimanjem mreže i tu je dio gdje ulazi druga komponenta odgovora na pitanje s početka.

Druga komponenta je uvođenje tzv. *prijenosne funkcije* (engl. *activation function*), odnosno nelinearne komponente na izlazu svakog neurona. Osim što ima ulogu poništavanja spomenutog problema kompozicije linearnih funkcija, prijenosna funkcija omogućava normaliziranje vrijednosti brojeva koji "teku" kroz neuronsku mrežu. Postoji mnogo različitih prijenosnih funkcija, ali dvije su korištene najčešće. Prva je *lo-*

gistička funkcija:

$$\sigma(x) = \frac{1}{1 + e^{-x}}, \quad (2.4)$$

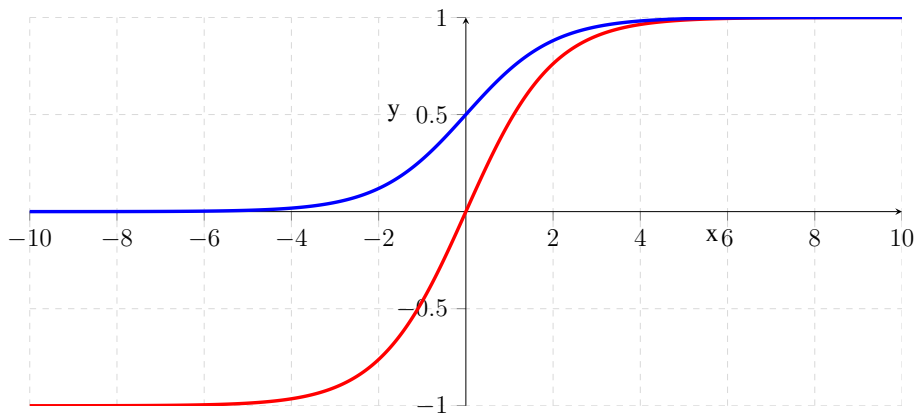
a druga je tangens hiperbolični:

$$\tanh(x) = 2 \cdot \sigma(x) - 1 = \frac{1 - e^{-2x}}{1 + e^{-2x}} \quad (2.5)$$

Jedina im je razlika u kodomeni:

$$\sigma : \mathbb{R} \rightarrow [0, 1]$$

$$\tanh : \mathbb{R} \rightarrow [-1, 1]$$



Slika 2.7: Logistička funkcija (plavo) i tangens hiperbolični (crveno)

Konačna funkcija jednog neurona bi, u slučaju primjene funkcije sigmoida, izgledala ovako:

$$f(\mathbf{x}) = \sigma[g(\mathbf{x})] = \frac{1}{1 + e^{-\mathbf{w}^\top \mathbf{x}}} \quad (2.6)$$

Posljedica korištenja ovakvih vrsta prijenosnih funkcija je činjenica da cijela neuronska mreža postaje diferencijabilna od početka do kraja. To će se pokazati kao ključna karakteristika koja omogućava proces učenja mreže.

2.3.3. Funkcija gubitka

Jedan od mogućih pogleda na težine koje ulaze u svaki neuron je onaj u kojem ih sve grupiramo u N matrica, gdje N predstavlja dubinu mreže:

$$W_0 = \begin{bmatrix} w_{00} & w_{01} & w_{02} & w_{03} \\ w_{10} & w_{11} & w_{12} & w_{13} \\ w_{20} & w_{21} & w_{22} & w_{23} \end{bmatrix}, \quad W_1 = \begin{bmatrix} w_{00} & w_{01} \\ w_{10} & w_{11} \\ w_{20} & w_{21} \\ w_{30} & w_{31} \end{bmatrix},$$

Razmotavanjem matrica težina u dugačke vektore i njihovim povezivanjem možemo dobiti *vektor težina*.

$$w = \begin{bmatrix} w_{00} & w_{01} & w_{10} & w_{11} & \dots & \dots & \dots & \dots & w_{x(y-1)} & w_{xy} \end{bmatrix}$$

Dimenzije dobivenog vektora će ovisiti o arhitekturi neuronske mreže, a skup njegovih vrijednosti označava sve moguće konfiguracije neuronske mreže s tom arhitekturom. Sa ovim opisom se može definirati proces učenja kao *promjena vrijednosti vektora težina* tako da težine što bolje zadovoljavaju određeni kriterij. Kriterij se zove *funkcija gubitka* (engl. *cost function*) i ukazuje na to koliko je trenutna konfiguracija težina udaljena od idealne konfiguracije u kojoj bi neuronska mreža davala točan izlaz za određene ulazne podatke. Odabir funkcije gubitka ovisi o konkretnom problemu koji se rješava. Najčešće korištene funkcije gubitka su *kvadratna funkcija gubitka*:

$$C(x, y, W) = \frac{1}{2} \sum_{j=0}^n (a_j - y_j)^2 \quad (2.7)$$

i *funkcija gubitka uzajamne entropije*:

$$C(x, y, W) = \sum_{j=0}^n \left[y_j \cdot \ln \frac{1}{a_j} + (1 - y_j) \cdot \ln \frac{1}{1 - a_j} \right] \quad (2.8)$$

u kojima x i y predstavljaju ulaz i izlaz, W predstavlja vektor težina, a a_j izlaz iz j -tog neurona zadnjeg sloja mreže.

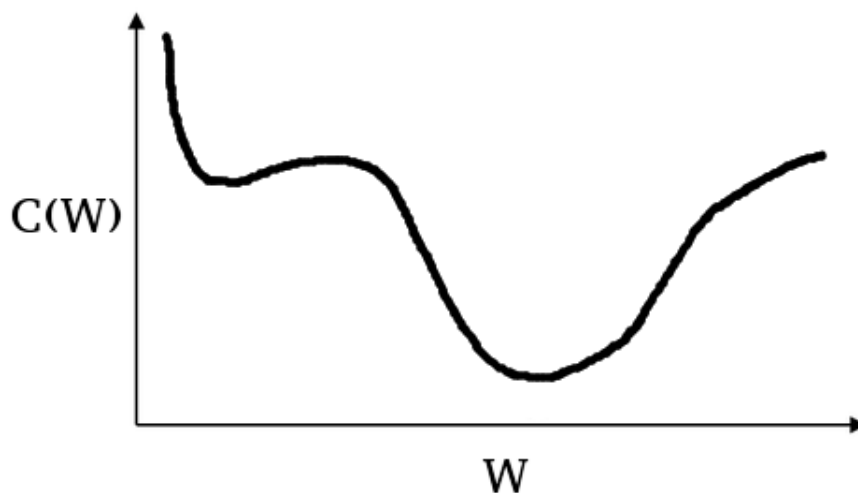
2.3.4. Euklidski prostor težina i metode optimizacije

Intuitivan način za razumijevanje problematike dolazi shvaćanjem da *prosjek* gubitka konvergira u jednu vrijednost kako skup ulaznih podataka za određen problem raste. Iz toga slijedi da se funkcija gubitka može zapisati kao funkcija samo od vektora težina:

$$C(W) : \mathbb{R}^n \rightarrow \mathbb{R} \quad (2.9)$$

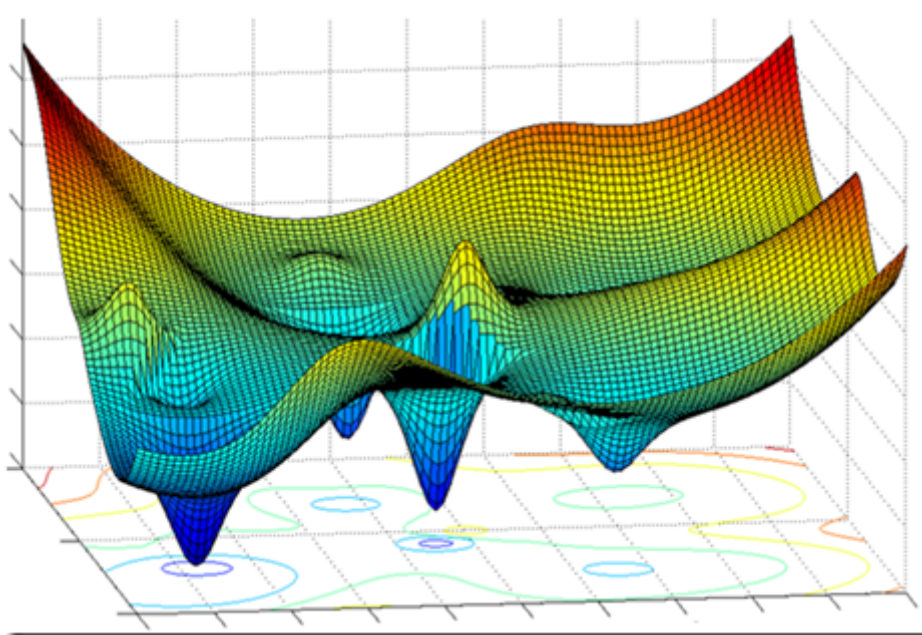
n -dimenzionalni prostor domene funkcije gubitka se dobije pridruživanjem elementa vektora težina odgovarajućoj dimenziji, gdje n predstavlja ukupan broj težina. Jedna točka u tom prostoru predstavlja jednu konfiguraciju težina.

U slučaju malog broja težina, dimenzionalnost ulaza je dovoljno mala da se prikaže vizualno.



Slika 2.8: Prikaz ovisnosti funkcije gubitka o jednoj težini

Funkcija gubitka je najčešće nekonveksna funkcija s mnogo lokalnih optimuma. Na slici 2.8 se vidi ovisnost funkcije gubitka o samo jednoj težini. U slučaju da prikazemo funkciju gubitka o dvije težine dobivamo graf mnogo veće kompleksnosti:



Slika 2.9: Prikaz ovisnosti funkcije gubitka o kombinacijama dvije težine

Kao što je spomenuto, proces učenja odgovara promjeni vrijednosti težina. Na slici se proces učenja može interpretirati kao pronalaženje globalnog minimuma *površine greške*. U $1-d$ ili $2-d$ slučaju površina se može prikazati, ali u većini slučajeva dimenzionalnost podataka će biti prevelika da bi se mogla vizualno interpretirati. Zbog toga je

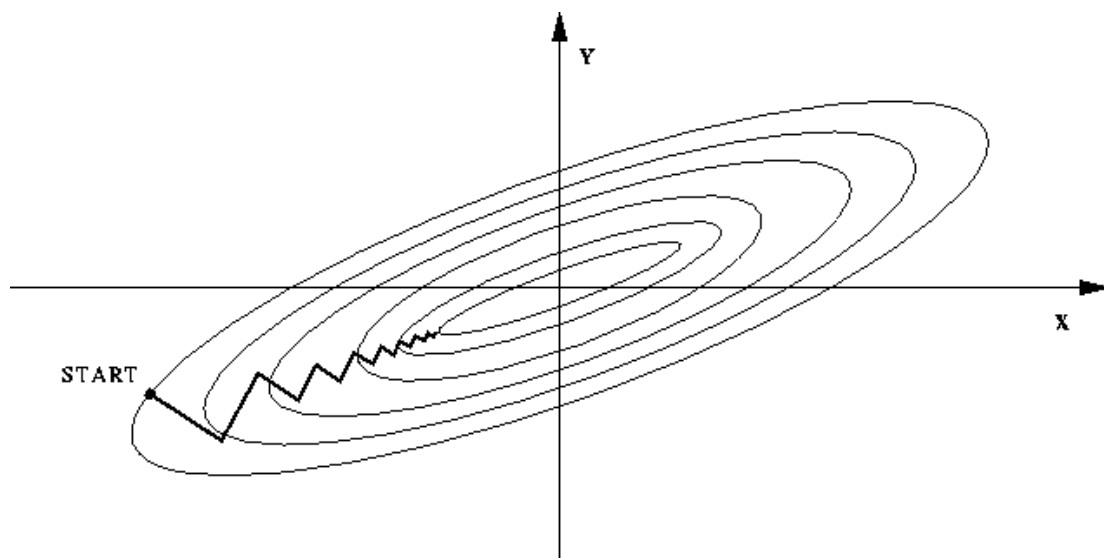
potrebno okrenuti se matematičkim metodama pronalaženja minimuma neke funkcije.

Metode optimizacije, odnosno pronalaženja minimuma funkcije više varijabli su iterativne i svode kretanje po hiperplohi funkcije gubitka dok se određen kriterij ne zadovolji (zadovoljavajući minimum, maksimalan broj iteracija). Iterativno se ponavljaju dva koraka:

1. Dobivanje informacije o lokalnoj strukturi hiperplohe oko odabrane točke;
2. Pronalazak sljedeće točke (na temelju dostupnih informacija strukturi hiperplohe) koja bi mogla dati najviše informacija o globalnom minimumu hiperplohe.

Ova perspektiva na optimizaciju je na dosta visokoj razini apstrakcije i većina je konkretnih implementacija jednostavnije implementirana.

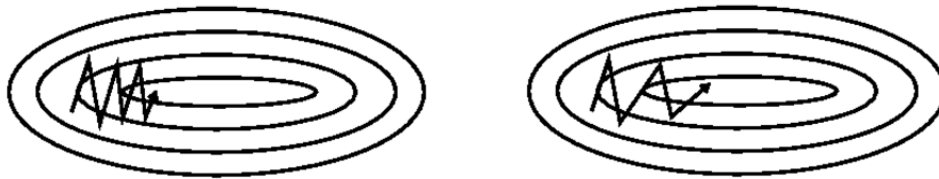
Najjednostavnija vrsta optimizacije je *gradijentni spust* (engl. *gradient descent*). Metoda evaluira gradijent u određenoj točki i kreće se u smjeru najvećeg pada.



Slika 2.10: Metoda gradijentnog spusta

Potrebno joj je zadati jedan hiperparametar: *stopu učenja* (engl. *learning rate*). To će se pokazati kao bitan parametar u procesu učenja.

Ostali modeli implementiraju složenije načine navigiranja po hiperplohi greške. Jedna od njih je *metoda momenta*, koja simulira kretanje po hiperplohi pod utjecajem sile koja vuče prema manjem gubitku (npr. gravitacija).



Slika 2.11: Usporedba metode gradijentnog spusta i metode momenta

Na slici 2.11 može se vidjeti da metoda momenta anulira okomito gibanje na pravac koji vodi do minimuma i time osigurava bržu konvergenciju.

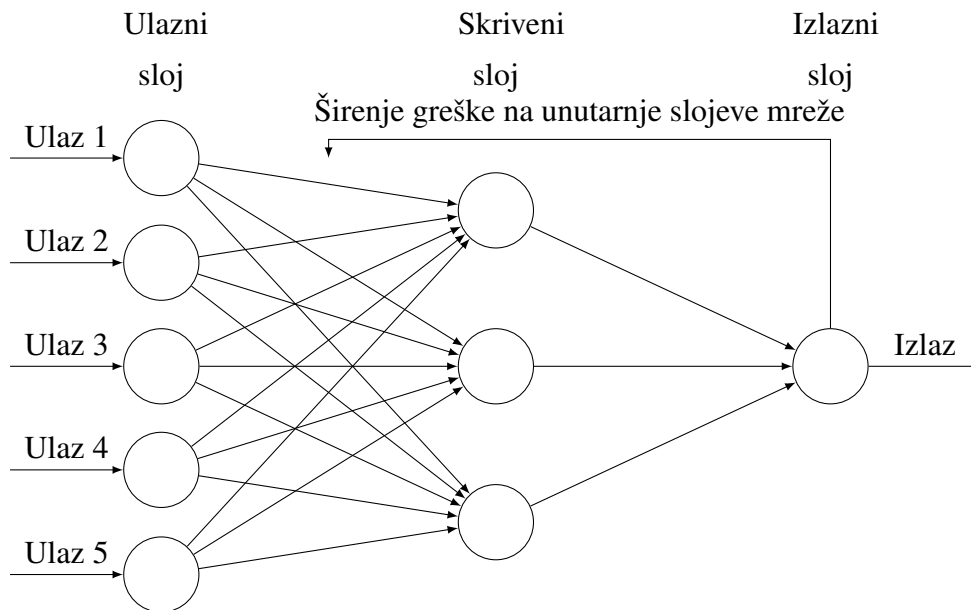
Postoje još razne metode optimiziranja: *Adagrad*, *Adadelta*, *RMSProp*, *Adam* i dr. Od spomenutih metoda se metoda *Adaptive Moment Estimation*, odnosno Adam [10] pokazala kao veoma efikasnom. Metoda efikasnije iskorištava informacije o lokalnoj strukturi hiperplohe i koristi napredne metode računanja smjera najvećeg pada gradijenta.

2.3.5. Algoritam unazadne propagacije

U poglavlju 2.3.3 spomenute su moguće implementacije funkcije gubitka u izlaznom sloju neuronske mreže, ali nije opisan proces računanja gubitka, tj. računanja grešaka u ostalim slojevima. Taj postupak se obavlja metodom *unazadne propagacije* (engl. *backpropagation*). Metoda, na obrnut način od prolaza prema naprijed (engl. *forward pass*) računa pogrešku predviđanja svakog neurona od izlaznog sloja prema ulaznom.

Algoritam unazadne propagacije povezuje sve već spomenute stvari (funkciju gubitka, prijenosnu funkciju, različite slojeve mreže) i lančano pravilo derivacije kako bi se efikasno izračunao rezultat. Metoda omogućava promjenu težina u mreži čime se postiže proces učenja. Cijeli postupak se može zapisati kao matrično množenje što ima znatan učinak na povećavanje brzine računanja, posebno na modernim grafičkim karticama.

Metoda unazadne propagacije postavlja jedan uvjet strukture neuronske mreže: to da je ona diferencijabilna. To znači da se se promjena izlaza može lančanim pravilom derivacije zapisati kao promjena svake pojedine težine. Derivabilnost mreže omogućava primjenjivanje informacija o strukturi hiperplohe greške kako bi se greška propagirala prema nazad i težine "pomakle" u određenom smjeru.



Slika 2.12: Prikaz širenja greške na unutarnje slojeve mreže

Propagiranje greške unazad donosi sa sobom jednu neželjenu posljedicu, a to je *problem nestajućeg gradijenta* (engl. *vanishing gradient problem*). Problem nestajućeg gradijenta odnosi se na smanjivanje propagirane greške kako se sloj propagacije udaljava od izlaza mreže. Posebno je istaknut na mrežama s velikim brojem slojeva. Posljedica je nemogućnost mreže da nauči više razine apstrakcije ili, u slučaju povratnih mreža (o kojima će biti riječi u poglavlju 3.2), nemogućnost mreže da zadržava informaciju duži vremenski period.

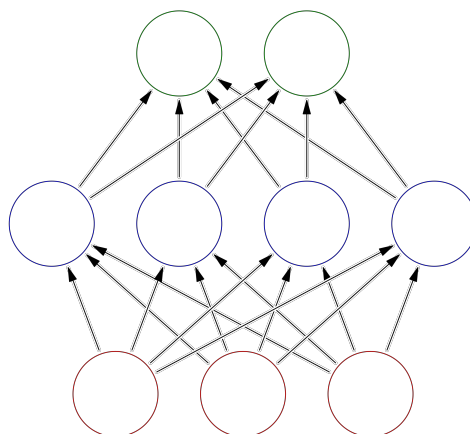
3. Vrste neuronskih mreža

Neuronske mreže se, unatoč svojoj složenosti, mogu prikazati kao obični usmjereni grafovi. Shodno tome, topologije neuronskih mreža se mogu podijeliti u nekoliko skupina, od kojih će svaka skupina dobro modelirati određenu vrstu problema.

U nastavku slijedi opis unaprijednih neuronskih mreža, povratnih neuronskih mreža i njihovih varijanti, ali osim njih postoje još i *konvolucijske* neuronske mreže, *mreže s radijalnom funkcijom* te mnoge druge.

3.1. Tradicionalne neuronske mreže

Unaprijedne neuronske mreže (engl. *feedforward networks*) su najjednostavnije i prve konstruirane umjetne neuronske mreže. U teoriji grafova, njihova topologija odgovara usmjerenom acikličnom grafu.



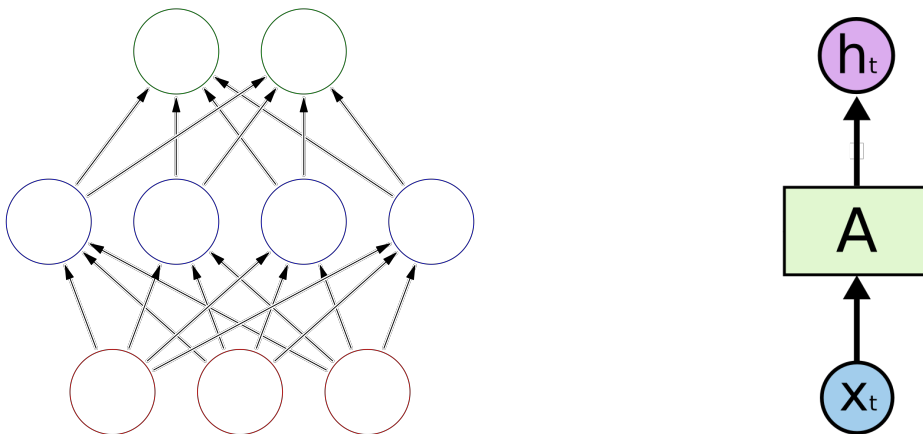
Slika 3.1: Prikaz topologije manje unaprijedne neuronske mreže

Iako po *univerzalnom teoremu aproksimacije* [12] sa samo jednim skrivenim slojem mogu aproksimirati bilo kakvu kontinuiranu funkciju, ovakve neuronske mreže imaju dosta velika ograničenja. Ne mogu primati vremenski ovisne podatke, tj. ne mogu biti naučene da, ovisno o podacima koji su prethodili, za iste ulazne podatke

daju različite izlaze. Drugi problem je što nisu invarijantne prema određenim transformacijama podataka. Unaprijedna neuronska mreža koja je naučila prepoznati znamenku 3 u sredini slike imat će potpuno drugačiju konfiguraciju težina od neuronske mreže koja je naučila prepoznati neku od geometrijskih transformacija znamenke 3 (rotiran, translaticiran ili povećan broj).

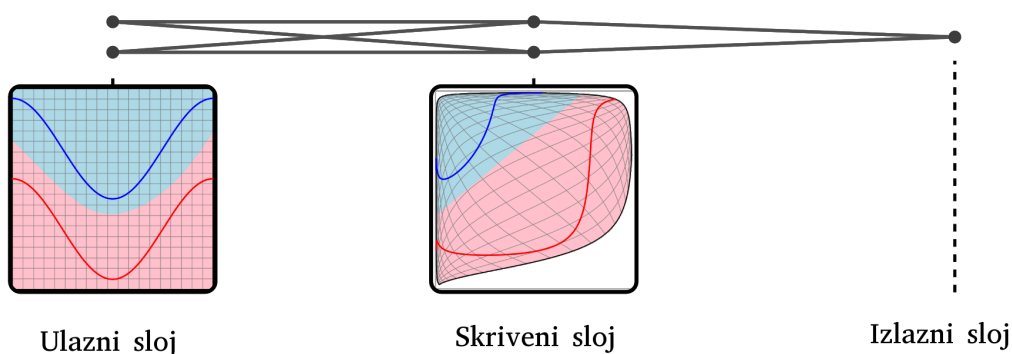
3.2. Povratne neuronske mreže

Unaprijedne neuronske mreže zapravo su podskup jednog šireg i moćnijeg skupa mreža, a to su povratne neuronske mreže (engl. *recurrent neural networks*). Njihova topologija odgovara usmjerenom *cikličkom* grafu i dobro rješava problem modeliranja vremenski ovisnih podataka. Dobivaju se proširenjem unaprijednih neuronskih mreža u dva koraka. Prvi korak je dizanje razine apstrakcije s individualnih ćelija neurona na ćelije *slojeva neurona*.



Slika 3.2: Unaprijedna neuronska mreža (lijevo) i njezina apstrakcija po slojevima (desno)

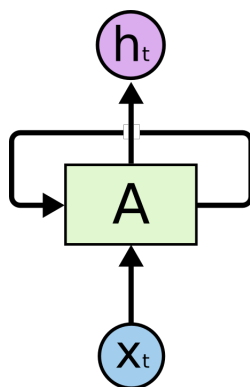
Ovakav pogled daje jednostavan pregled dubine mreže. Svaka ćelija (koja sada predstavlja cijeli sloj neurona) linearno transformira podatke na jedan određen način kako bi izlazni sloj, u slučaju klasifikacije, mogao lakše odijeliti različite klase podataka.



Slika 3.3: Linearne transformacije podataka slojeva mreže¹

Na slici 3.3 prikazana je troslojna neuronska mreža čiji je zadatak klasificirati točke ravnine kao dijelove jedne od dviju krivulja. Nakon treniranja, skriveni sloj mreže je naučio korisnu transformaciju ulaza.

Drugi korak je dodavanje povratne veze iz ćelije skrivenog sloja u samu sebe:



Slika 3.4: Povratna neuronska mreža²

Ova promjena omogućava vremenski ovisne izlaze i, iako se čini mala, ima velike posljedice na strukturu i performanse mreže, kao i na brzinu učenja. Omogućava ćeliji mreže da sama procesira svoj izlaz i, na neki način, prvi je korak prema *meta* učenju.

Ta promjena zahtijeva i pojašnjenje jer, kada se razina apstrakcije spusti niže, postavlja se pitanje kako implementirati povratnu vezu? Da li se informacije u uzastopnim ćelijama prenose isto matricama težina ili postoje još neke druge varijable koje opisuju stanje ćelije?

Odgovor je: prenose se kako god mi hoćemo. Osim osiguravanja svojstva da je mreža diferencijabilna od kraja do kraja, ne postoji ograničenje na strukturu podataka

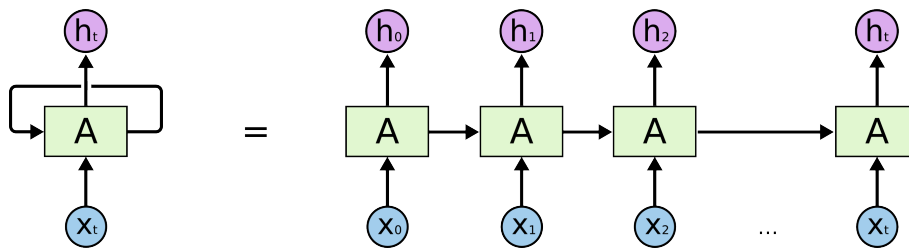
¹<http://colah.github.io/posts/2015-09-NN-Types-FP/>

²<http://colah.github.io/posts/2015-08-Understanding-LSTMs/>

kojom se prenosi informacija.

Najjednostavniji model povratnih neuronskih mreža implementira običnu matricu težina koja se samo proslijedi sljedećoj ćeliji, ali postoje modeli s puno kompleksnijom unutarnjom strukturom ćelije.

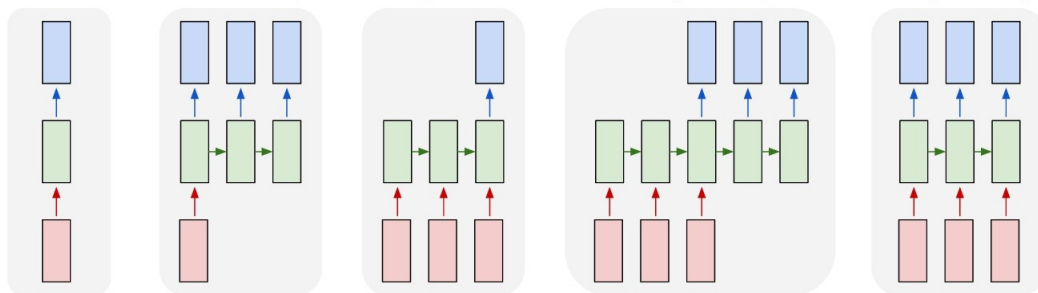
Intuitivan način za razumijevanje rada povratne neuronske mreže se može dobiti prikazivanjem vremenske dimenzije prostorno.



Slika 3.5: Razmotavanje povratne neuronske mreže ³

Na slici 3.5 prikazano je razmotavanje povratne neuronske mreže. Ona se može zamisliti kao više kopija iste neuronske mreže od kojih svaka prosljeđuje poruku sljedećoj. Lančana veza otkriva da su povratne neuronske mreže blisko povezane sa sekvencama i listama.

Na lijevoj strani slike 3.6 je prikazana unaprijedna neuronska mreža, dok prikazi desno od nje ilustriraju njezino proširenje, odnosno povratne neuronske mreže sa svojim različitim arhitekturama.

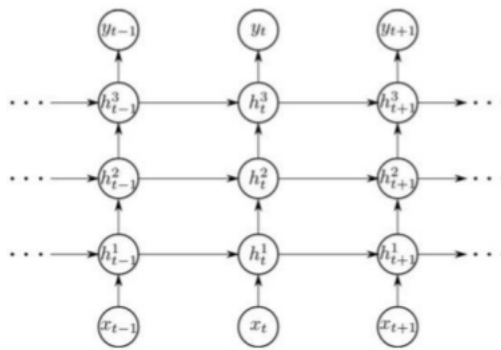


Slika 3.6: Različite arhitekture povratne neuronske mreže ⁴

Osim na prikazani način, različite arhitekture povratnih mreža mogu imati i *varijabilan broj skrivenih slojeva*. Osim što su odlične u modeliranju vremenskih sekvenci, ovakve mreže najčešće mogu modelirati više razine apstrakcije i najčešće brže konvergiraju.

³<http://colah.github.io/posts/2015-08-Understanding-LSTMs/>

⁴<http://karpathy.github.io/2015/05/21/rnn-effectiveness/>



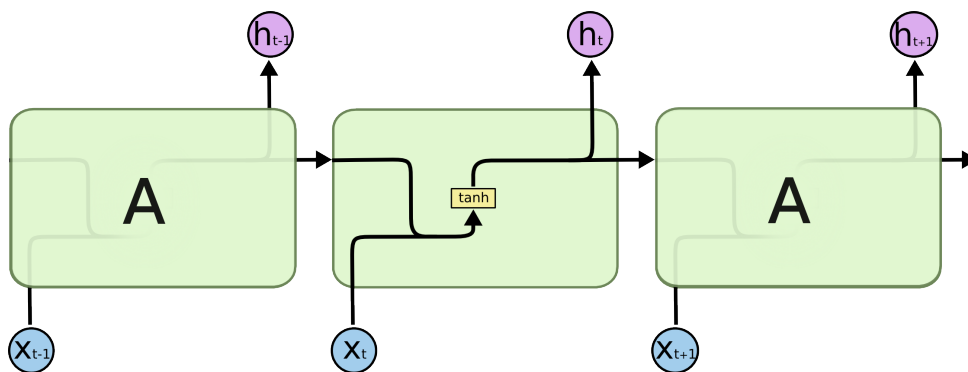
Slika 3.7: Povratna neuronska mreža s tri skrivena sloja

Iako bi se moglo reći da su povratne neuronske mreže prirodan način obrade vremenski ovisnih podataka, njihova osnovna verzija ima nekoliko nedostaka od kojih je najveći već spomenut problem nestajućeg gradijenta.

3.2.1. Mreže LSTM

Modeli neuronskih mreža s dugim kratkoročnim pamćenjem (engl. *Long short-term memory*, ili samo LSTM) posebna su vrsta arhitekture povratnih neuronskih mreža sposobne naučiti duže vremenske ovisnosti. Osmislili su ih Hochreiter i Schmidhuber [7] te su veoma popularne i rade iznimno dobro na raznim vrstama problema.

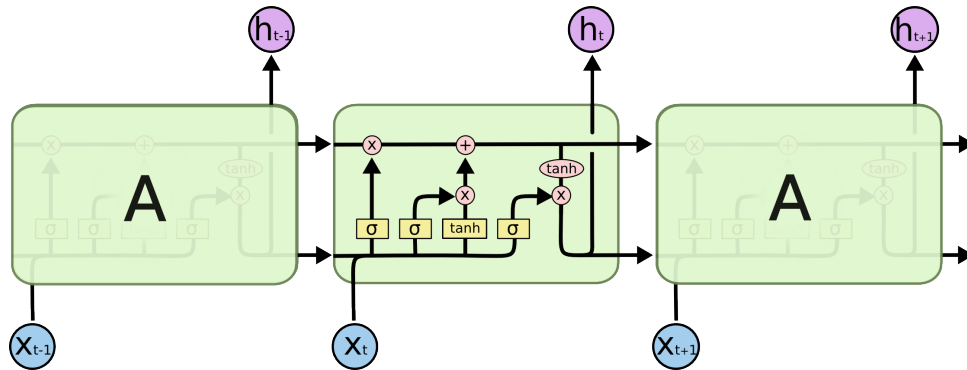
Eksplícitno su dizajnirane da izbjegnu problem nemogućnosti modeliranja dugih vremenskih ovisnosti. Cristoper Olah je sjajno opisao njihovu unutarnju strukturu te njihove mnogobrojne varijante u svom članku [14]. Razlika između ćelija obične povratne mreže i ćelija LSTM je u njihovom načinu procesiranja informacija.



Slika 3.8: Prikaz unutarnje strukture ćelije obične povratne mreže ⁵

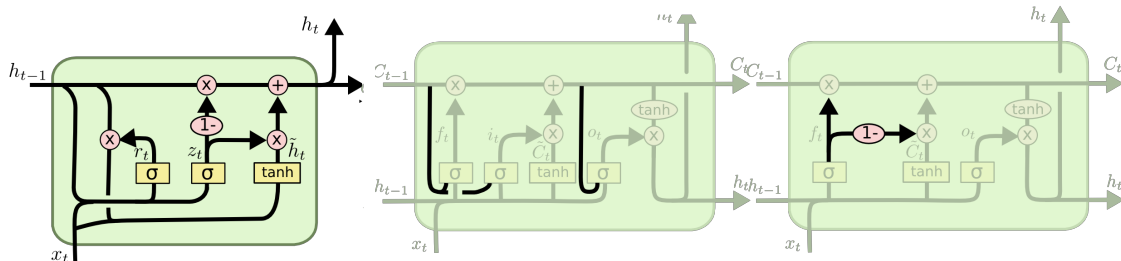
⁵<http://colah.github.io/posts/2015-08-Understanding-LSTMs/>

Dok povratna neuronska mreža radi jednostavan posao računanja linearne transformacije ulaznih podataka i prijašnjeg stanja ćelije (slika 3.8), LSTM ćelija radi mnogo složenije operacije, kako je prikazano slikom 3.9.



Slika 3.9: Prikaz unutarnje strukture ćelije LSTM ⁵

Svaka od crnih strelica predstavlja jedan cijeli vektor dok ljubičasti krugovi predstavljaju matematičke operacije između dva vektora. Ćelija LSTM ostvaruje složeniji način rada kombiniranjem različitih vrata (engl. *gate*) kojima propušta ili dodaje nove informacije u već postojeće vektore. Na slici 3.10 su prikazane različite kombinacije vrata u ćelijama LSTM:



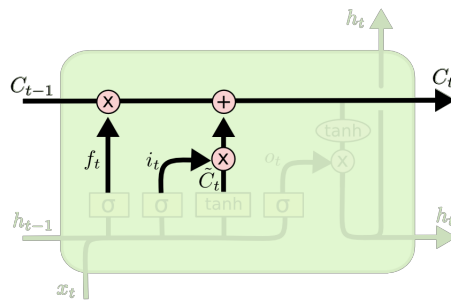
Slika 3.10: Prikaz unutarnjih struktura raznih varijanti ćelija LSTM ⁵

Može se primijetiti da ćelija LSTM prenosi informaciju o svom unutarnjem stanju kroz dva vektora neurona, za razliku od obične povratne ćelije.

Postoji puno varijanti ćelija LSTM, puno različitih načina kombiniranja vektora i puno različitih "sklopova" kojima se može opisati rad takvih ćelija. Postavljaju se pitanja: kako točno ti sklopovi mogu izgledati? Postoje li nekakve granice veličina tih sklopova? Kako izgled konfiguracije različitih vrata utječe na performans ćelije? Što je dobra, a što loša konfiguracija?

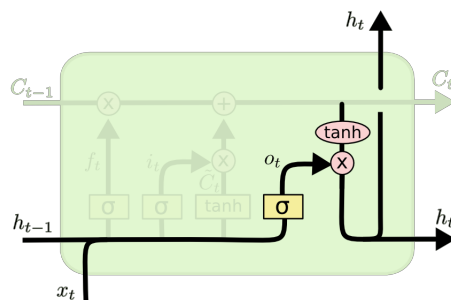
Empirijska analiza [6] ne pronalazi značajne razlike između različitih konfiguracija. Teorijsko razmatranje, doduše, može dovesti do nekih zanimljivih zaključaka (dodatak A).

Ćelija LSTM može se podijeliti na dva dijela: dio ćelije koji sadrži vektor njezinog unutarnjeg stanja (memorije) te dio ćelije koji ima mogućnosti *pisati* ili *čitati* iz te memorije. Pisanje se odvija pomoću *vrata zaborava* (engl. *forget gate*) i *vrata upisa* (engl. *input gate*). Prva vrata brišu određene informacije iz memorije, dok druga vrata upisuju neku informaciju u memoriju.



Slika 3.11: Upisivanje u memorijski sloj LSTM ćelije ⁶

Čitanje se odvija pomoću *izlaznih vrata* (engl. *output gate*), koji odabiru informacije koje će biti prosljeđene dalje.



Slika 3.12: Čitanje iz memorijskog sloja LSTM ćelije ⁶

⁶<http://colah.github.io/posts/2015-08-Understanding-LSTMs/>

4. Analiza sentimenata i korišteni alati

Duboke neuronske mreže, a pogotovo mreže LSTM, pokazuju zavidan uspjeh u području analize sentimenata. Postoje različiti načini i različite arhitekture koje možemo primjeniti te oni ovise o konkretnom problemu analize sentimenata koji rješavamo. U nastavku je opisan problem analize sentimenata na skupu podataka IMDb i dosadašnja postignuća nekih značajnijih radova.

4.1. Problem analize sentimenata

Analiza sentimenata (engl. *sentiment analysis*, *SA*) tipičan je zadatak u analizi prirodnog jezika. Analiza sentimenata ključna je za razumijevanje korisnički generiranog sadržaja na socijalnim mrežama ili u recenzijama proizvoda te u zadnje vrijeme dobiva puno pažnje i od akademske zajednice i od industrije [17]. Dijeli se u različite kategorije: analiza sentimenta na razini dokumenta, analiza sentimenta riječi, subjektivnih izraza ili subjektivnih rečenica. Analiza sentimenata na nižoj razini (riječi, rečenice) može dovesti do ukupnog poboljšavanja efektivnosti klasifikacije. Na problem klasifikacije sentimenata se mogu primjeniti razni pristupi.

Rani rad na preciznoj analizi sentimenata fokusira se na prepoznavanje subjektivnih fraza [19]. U radu [1] problem je formuliran kao označavanje sekvenci na razini *tokena*, dajući značajno bolje rezultate od prijašnjih referentnih vrijednosti. Proširivanje predviđanja sekvenci napravljeno je identificiranjem nositelja mišljenja i otkrivanjem polariteta i intenziteta zajedno sa sentimentom [4]. Vrlo su popularne metode dubokog učenja koje analiziraju rečenicu u obliku slijeda tokena i uspješno su primjenjene na problem analize sentimenata. Među njima su se *rekurzivne* neuronske mreže [16] pokazale iznimno uspješnima u modeliranju kompozicija različitih sentimenata [8].

4.2. Skup podataka

Skup podataka koji je korišten u analizi je *Large Movie Review Dataset v1.0* [11]. To je kolekcija korisničkih recenzija s internetske baze filmova (IMDb). Kolekcija služi kao mjerilo za kvalitetu analize sentimenata.

Kolekcija se sastoji od 50.000 recenzija na engleskom jeziku jednako podijeljenih u 25.000 primjera za učenje i 25.000 primjera za testiranje. Distribucija pozitivnih i negativnih ocjena je isto jednako podijeljena, s 25.000 pozitivnih i 25.000 negativnih primjera.

U cijeloj kolekciji nije dozvoljeno više od 30 recenzija za isti film zato jer recenzije za isti film znaju imati korelirane ocjene. Skupovi primjera za treniranje i za testiranje sadrže disjunktan skup filmova, tako da se ne može dobiti značajno poboljšanje performansi memoriziranjem pojmova vezanih za konkretan film i ocjena asociranih s njima.

Svaka recenzija ima asociranu ocjenu. Pozitivne recenzije klasificirane su one s ocjenom ≥ 7 od maksimalne ocjene 10. Negativne recenzije klasificirane su one s ocjenom ≤ 4 od 10. Neutralne recenzije nisu uključene.

4.3. Analiza i obrada podataka

U citatima ispod su navedeni konkretni primjeri korisničkih recenzija. Primjer pozitivne recenzije s ocjenom 10:

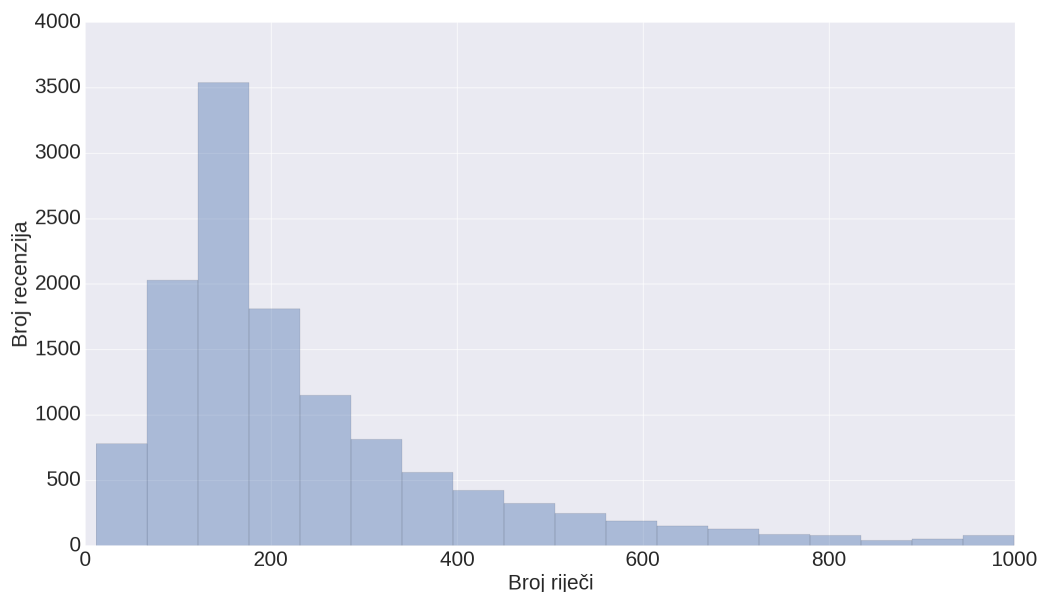
I think that this film was one of Kurt Russels good movies. Kurt russel is my favorite actor so I think that he is a good actor in any role he plays. But this movie had a lot of action in it and I know that it should have more then a 5.6 out of 10 on the meter but many people did not like this movie. Oh well I thought it was good so I think that every one should see this movie. If you see this movie and like it I think that you should see Back Draft also with Kurt Russel. I give Soldier *** 1/2 out of *****

Primjer negativne recenzije s ocjenom 1:

This film is a joke and Quinton should be ashamed of himself, trying to pass this off as a Modesty Blaise Film. If you are having trouble sleeping then all means rent this film. The stick figure they call a actress who is suppose to be Modesty Blaise has got to be the most boring person on this planet. Maybe she could be used as a hat stand in the back ground of a real film.seventy-five minutes of nothing thank you who ever invented the fast forward button. If you see this film if you can call it that coming your way RUN. I can't help but think what 3rd world country could of used the money wasted of this crap. this film is boring the actors are boring waste of colour a waste air they breath If you would like to see Mostey Blaise Film then watch the one they made in the 60's maybe that what the director should of done.

Može se primijetiti da recenzije u potpunosti iskorištavaju mogućnosti prirodnog jezika i na kolokvijalan način dočaravaju subjektivan dojam filma. Mnoge od njih sadrže i gramatičke greške, emotikone i nedovršene rečenice. Također, iz nekih od njih nije odmah jasno da li su pozitivne ili negativne, zbog korištenja sarkazama ili nekih drugih načina opisivanja koji su na semantički visokoj razini apstrakcije.

Distribucija duljine recenzije po broju riječi prikazana je na slici 4.1 na kojoj se može vidjeti da većina recenzija ima između 150 i 200 riječi. Prosječna duljina recenzije je 232 riječi.



Slika 4.1: Distribucija duljine recenzije po broju riječi

4.3.1. Vektorska reprezentacija riječi

Današnji sustavi za procesiranje slika i zvuka rade s bogatim, visoko dimenzionalnim skupovima podataka enkodiranim kao intenziteti piksela za slikovne podatke ili kao koeficijenti spektralne gustoće snage za zvukovne podatke. Obrada takvih vrsta podataka je, za razliku od obrade teksta, puno lakša zbog jedne bitne činjenice. Sve potrebne informacije za prepoznavanje zvuka ili govora se *već nalaze u podacima*, dok za tekst to nije istina. Ljudi nemaju problema s prepoznavanjem uzoraka u slikama ili zvuku s kojima se prije nikad nisu susreli, no imaju problema s prepoznavanjem teksta na nepoznatom jeziku.

Ključna razlika je u tome što je tekst *kodiran proizvoljno* i njegove "kodne riječi" ne pružaju nikakve korisne informacije o njihovim međusobnim odnosima, ako ti odnosi uopće postoje. Tekst može biti kodiran proizvoljnim atomima kao "mačka", "Id143" ili "aprendizaje profundo" i oni mogu imati različita značenja u ovisnosti o kontekstu. Drugi problem je da takva vrsta predstavljanja podataka vodi do oskudnosti podataka (engl. *data sparsity*) i to najčešće znači da je potrebno više podataka da bi se uspješno istrenirali statistički modeli.¹

Jedan način rješavanja navedenih problema je korištenje *vektorske reprezentacije riječi*. Dva najosnovnija načina predstavljanja riječi pomoću vektora su *one-hot encoding* i *distribuirano predstavljanje riječi*.

One hot encoding kodira riječ u N -dimenzionalni rijetki binarni vektor koji jednak veličini ukupnog vokabulara cijelog skupa podataka. Svaki element tog vektora predstavlja jednu riječ i samo jedan element može biti jednak jedan. To znači da je jedna riječ kodirana kao vektor kojim su svi elementi nula, osim jednog. Naprimjer, riječ "pas" može izgledati kao:

$$\begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 0 & \dots & 0 & 0 \end{bmatrix},$$

dok bi riječ "nebo" mogla izgledati ovako:

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & \dots & 1 & 0 \end{bmatrix}.$$

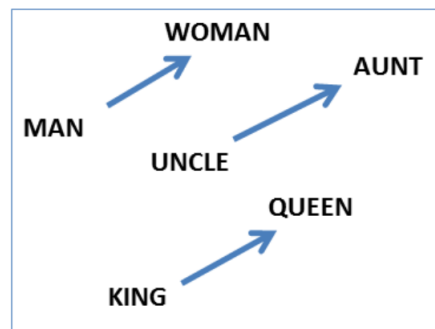
Prednost ovog načina kodiranja je uklanjanje linearne ovisnosti među različitim riječima, ali dolazi s cijenom dimenzionalnosti, koja najčešće bude reda 10^5 .

Drugi, mnogo moćniji i prirodniji način modeliranja ulaznih podataka je s tzv. Word2Vec modelom distribuiranog predstavljanja riječi. Taj model pridjeljuje svakoj

¹<https://www.tensorflow.org/versions/r0.9/tutorials/word2vec/index.html>

ulaznoj riječi jedan N -dimenzionalni gusti vektor. N je najčešće relativno malen broj, reda veličine 10^3 što mu je jedna od prednosti u odnosu na *one-hot* model.

Glavna prednost mu je to što modelira semantičku povezanost između određenih riječi. Svaki element tog vektora predstavlja određenu značajku, a neke od tih značajki se mogu vizualno interpretirati razmatranjem odnosa između riječi.



Slika 4.2: Modeliranje prirodnih veza među riječima ²

Na slici 4.2 se može vidjeti da se oduzimanjem vektora riječi "woman" od vektora riječi "man" dobije približno isti rezultat kao kad se oduzme vektor riječi "queen" od vektora riječi "king". Unatoč ovako zanimljivim rezultatima, velika većina značajki je neinterpretabilna. Ovaj model predstavlja velik korak naprijed u odnosu na *one-hot* model jer modelu strojnog učenja već predaje vektor koji ima nekakvo značenje.

4.3.2. Obrada recenzija

Obrada recenzija je uključivala dva koraka: ograničavanje vokabulara te ujednačavanje duljine recenzija.

Zbog tehničkih ograničenja (brzina učitavanja i procesiranja podataka) morao se ukupan vokabular svih recenzija smanjiti. Skup korištenih riječi se ograničio na 10.000 najčešćih riječi. To znači da su sve rijetke riječi (specifične za neki pojam relevantan filmu) bile izbačane.

Distribucija duljine recenzije nije konstantna nego varira (slika 4.1). Budući da povratne neuronske mreže rade jako dobro s ulaznim podacima koji imaju jednaku duljinu vremenske komponente, recenzije su se trebale obraditi i njihova duljina ujednačiti.

Korišten način bio je procesiranje samo prvih N riječi u recenziji, gdje se broj N naknadno odredio. To je značilo da neke recenzije nisu bile u potpunosti procesirane,

²<https://www.aclweb.org/anthology/N/N13/N13-1090.pdf>

ali i da su neke recenzije morale biti nadopunjene jer im je duljina manja od N . Odabran način nadopunjivanja je bio s jednim elementom koji predstavlja oznaku za kraj niza i drugim elementom koji je bio dodavan dok je god duljina recenzije bila manja od N .

4.4. Korišteni alati

Za razvijanje rješenja korišten je programski jezik Python 3.4 te razne pomoćne biblioteke. Najvažnija korištena biblioteka je *TensorFlow*.

TensorFlow je biblioteka za strojno učenje otvorenog koda (engl. *open source machine learning library*) stvorena u Googleu.³ Omogućava numeričko računanje korištenjem grafova tokova podataka. Fleksibilna arhitektura omogućava implementiranje računanja na jednom ili više procesora ili grafičkih kartica na stolnom računalu, serveru ili mobilnim uređajima. TensorFlow uključuje vrlo moćan alat za vizualizaciju unutarnje strukture mreže *TensorBoard* koji je u ovom radu korišten za prikaz neuronskih mreža. Ostale biblioteke uključuju *numpy*, *sklearn*, *matplotlib*, *seaborn*, *gensim* i dr.

Računala korištena za obradu podataka je prijenosno računalno sa integriranom grafičkom karticom, 4GB radne memorije i Intel Core i5 procesorom ($4 \times 1.60\text{GHz}$), kao i računala *Đurđa* i *Ljudmila* ustupljena od *Text Analysis and Knowledge Engineering Laba* (TakeLab) FER-a.

Đurđa je računalo karakteristično po velikoj količini radne memorije (64 GB) i procesorom s 32 jezgre. *Ljudmila* je računalo koje ima poznatu grafičku karticu *Titan X*, specijaliziranu za duboko učenje i to je računalo korišteno u većini eksperimenata.

³<https://www.tensorflow.org/>

5. Eksperimenti, evaluacija i analiza

Postoji nekoliko metoda evaluacije performansa neuronske mreže. Može se procjenjivati funkcija gubitka, preciznost, odziv (engl. *recall*) te njihove funkcije kao što su F-mjere. Za problem analize sentimenta dobro mjerilo performanse mreže je točnost izlaza. Točnost izlaza računa se kao omjer točnih rezultata naprama ukupnom broju svih rezultata.

Ono što razlikuje različite konfiguracije neuronskih mreža su njihovi *hiperparametri*. Razlika između hiperparametara i običnih parametara je u tome što neuronska mreža sama nauči podešavati svoje parametre, dok hiperparametri moraju biti ručno postavljeni. Provedeni eksperimenti se mogu podijeliti u nekoliko različitih kategorija, od kojih je svaka sljedeća kompleksnija i daje bolje rezultate. U tablici 5.1 prikazani su hiperparametri korišteni u treniranju LSTM mreže.

Tablica 5.1: Lista hiperparametara i odgovarajućih kratica

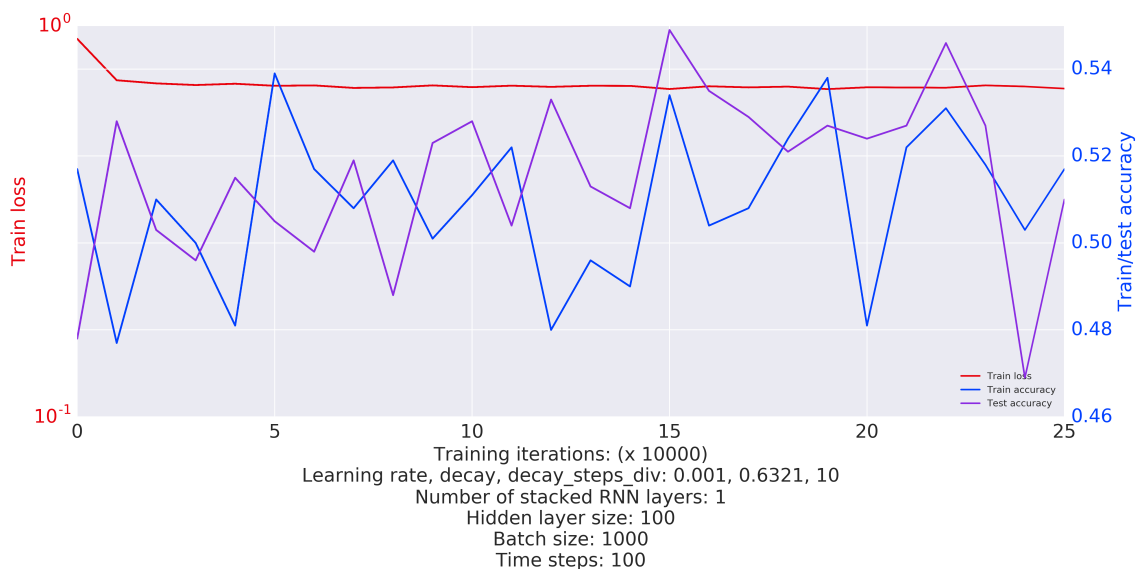
Hiperparametar	Korištena kratica
Veličina ulaznog sloja	n_input
Broj vremenskih koraka	n_steps
Broj slojeva povratne mreže	n_layers
Broj neurona u svakom skrivenom sloju	n_hidden
Stopa opadanja stope učenja	decay_rate
Period primjene stope opadanja	decay_steps_div
Broj iteracija treniranja	training_iters
Veličina gomile podataka	batch_size
Stopa učenja	learning_rate

Budući da je cilj klasificirati ulazne podatke u jednu od dvije klase, referentna vrijednost (engl. *baseline*) je 50%. Eksperimenti su pokrenuti na prijenosnom računalo, no bez rezultata zbog nedovoljne snage istog. Da bi se ostvarili ikakvi značajni rezultati treba se koristiti računalo s dovoljno radne memorije kako bi se skup podataka

mogao učitati u memoriju i računalo s dovoljno brzom grafičkom karticom koja u razumnom vremenu može istrenirati mrežu. U ostatku rada, svi opisani eksperimenti bili su pokrenuti na *Ljudmili*.

5.1. One-hot reprezentacija

Prva i najjednostavnija isprobana metoda je korištenje *one-hot encodinga* na jednoslojnoj povratnoj neuronskoj mreži s običnom ćelijom (ćelija *tanh* bez ikakvih vrata). Iako je za očekivati barem nekakav rezultat iznad referentne vrijednosti, niti jedna isprobana konfiguracija mreže nije davala konkretne rezultate.



Slika 5.1: Ovisnost točnosti i funkcije gubitka o iteraciji treniranja

Na slici 5.1 prikazan je proces učenja povratne neuronske mreže kroz nekoliko epoha. Na njoj se može vidjeti više informacija o performansu mreže: funkcija gubitka, točnost na skupu za treniranje i točnost na skupu za validaciju. Označe na x -osi predstavljaju broj s kojim se treba množiti parametar pod *training_iterations* kako bi se dobio točan broj primjera na kojima je mreža bila trenirana. Na Y -osi prikazane su funkcija gubitka na logaritamskoj skali i točnost na linearnoj skali. Može se zamijetiti da točnost stalno oscilira i efektivno ne prelazi 50%. To ukazuje na velika "skakanja" metode optimizacije po hiperplohi greške i često se može smanjiti smanjivanjem stope učenja. Ipak, pokušaj smanjivanja parametra *learning_rate* nije urodio plodom, kako je prikazano slikom 5.2.



Slika 5.2: Ovisnost točnosti i funkcije gubitka o iteraciji treniranja (smanjena stopa učenja)

Za treniranje mreže s *one-hot* ulazom potrebno je puno više vremena zbog velikih dimenzionalnosti matrica, što ograničava mogućnosti pretrage prostora hiperparametara. Veliki problem je što i *one-hot* ulaz ne modelira dobro semantiku podataka.

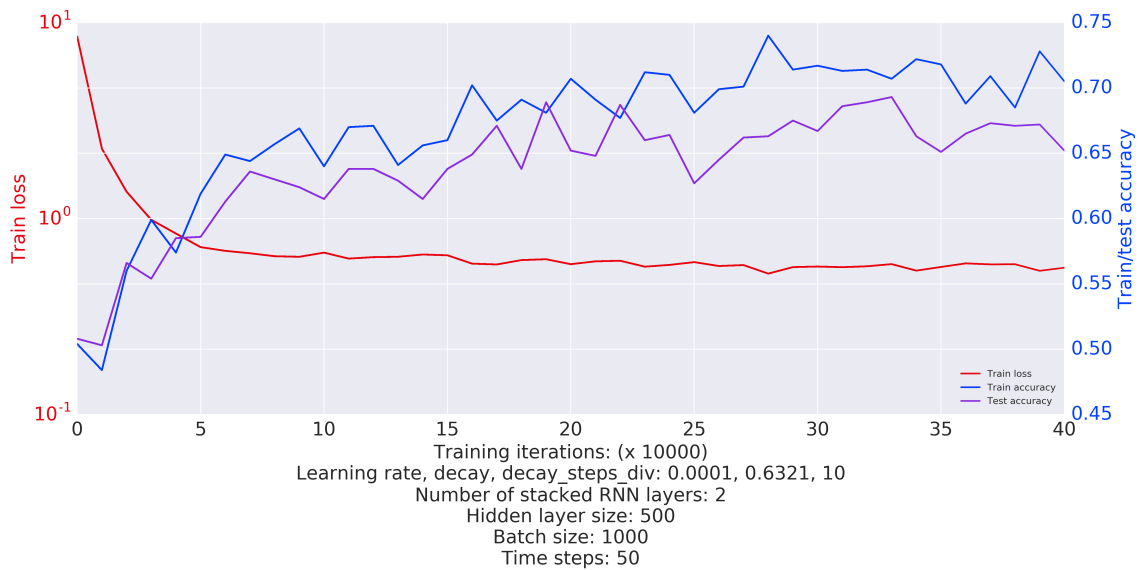
5.2. Distribuirano predstavljanje riječi

Distribuirano predstavljanje riječi izvanredno modelira semantiku podataka i kao takvo iz prvog pokretanja dalo je naznake da bolje modelira podatke (slika 5.3).



Slika 5.3: Ovisnost točnosti i funkcije gubitka o iteraciji treniranja (model Word2Vec)

Ostala pokretanja, s odokativno odabranim hiperparametrima također daju zanimljive rezultate, kao što se vidi na slici 5.3.

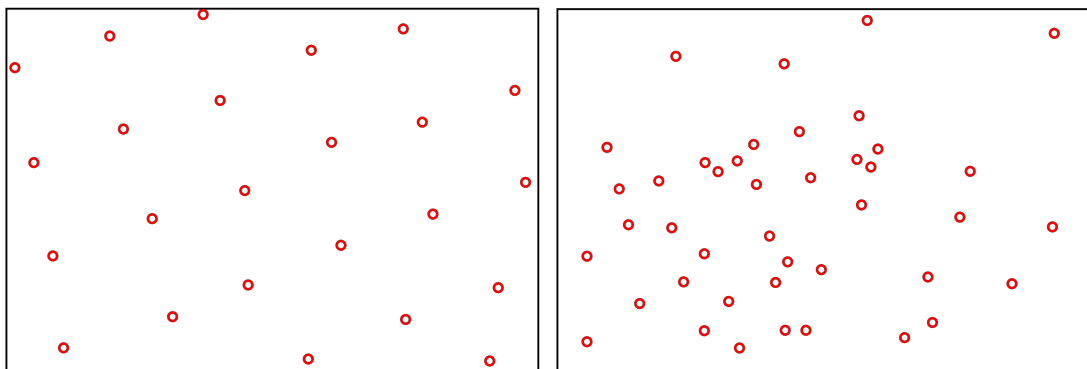


Slika 5.4: Ovisnost točnosti i funkcije gubitka o iteraciji treniranja (model Word2Vec)

5.2.1. Pretraživanje po rešetki

Iako velik broj konfiguracija koje koriste distribuirano predstavljanje riječi urađa plodom, nasumična i odokativna pretraga prostora hiperparametara ne daje puno informacija o njegovom eventualnom optimumu. Od raznih metoda pronalaženja najboljih hiperparametara, korištena je metoda pretraživanja po rešetki (engl. *grid search*).

Pretraživanje po rešetki se sastoji od ujednačenog uzorkovanja točaka iz prostora pretraživanja hiperparametara. Razlikuje se od nasumičnog pretraživanja po tome što je jednoliko raspoređeno u prostoru:



Slika 5.5: Usporedba pretraživanja po rešetki i nasumičnog pretraživanja

Pretraživanje po rešetki omogućava sistematsko pretraživanje prostora hiperparametara i, po potrebi, fokusiranje na jedan njegov podskup na koji se opet može primijeniti pretraživanje po rešetki.

U tablicama ispod prikazan je rezultat pretraživanja po rešetki u kojem su se mijenjali parametri *stopa učenja*, *veličina skrivenog sloja* i *broj vremenskih koraka*.

Tablica 5.2: Rezultati pretraživanja po rešetki s parametrom `time_steps = 60`

Točnost mreže		Veličina skrivenog sloja		
		100	500	1000
Stopa učenja	0,0003	76%	78%	78%
	0,001	75%	77%	76%
	0,01	62%	74%	71%

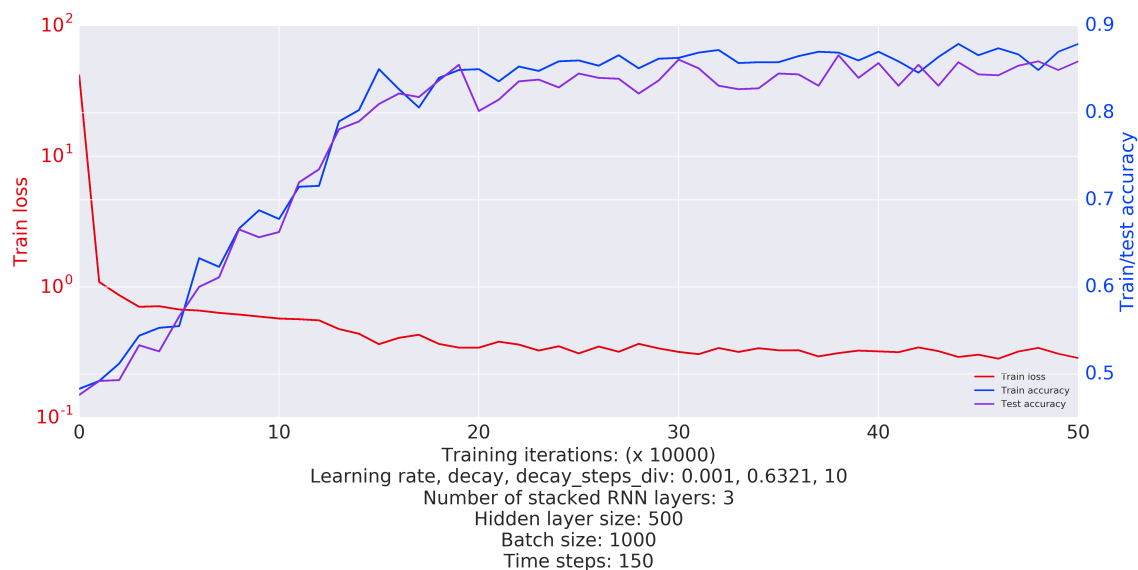
Tablica 5.3: Rezultati pretraživanja po rešetki s parametrom `time_steps = 30`

Točnost mreže		Veličina skrivenog sloja		
		100	500	1000
Stopa učenja	0,0003	70%	72%	63%
	0,001	71%	68%	71%
	0,01	66%	51%	50%

Iz gornjih tablica može se vidjeti da povećavanje broja vremenskih koraka veoma pozitivno utječe na dobivenu točnost. Efektivno se mreži daje više informacija s kojima ona može raditi. Povećavanje broja neurona u skrivenom sloju, odnosno, povećavanje kompleksnosti mreže, nužno ne poboljšava rezultat, a čak nekad može djelovati negativno na performans.

Nakon isprobavanja mnogih konfiguracija, najbolji dobiveni rezultat prikazan je na slici 5.6. Radi se o troslojnoj povratnoj neuronskoj mreži s ćelijom LSTM koja veoma brzo postiže točnost od 80%. Na kraju, konačna točnost na testnom skupu postiže vrijednost od 84%. To ujedno predstavlja i najbolji rezultat ovog rada.

Doduše, ovi se rezultati trebaju uzeti sa zrnom soli. Zbog vremenskih ograničenja, konačna evaluacija *nije napravljena na trećem skupu podataka*. Posljedica je da postoji mogućnost da konfiguracija mreže sa slike 5.6 daje previše optimistične rezultate.



Slika 5.6: Ovisnost točnosti i funkcije gubitka o iteraciji treniranja (model Word2Vec)

5.3. Analiza

Proces treniranja pokazuje znakove kaotičnog sustava (sustava osjetljivog na početne uvjete). Početne uvjete u ovom slučaju predstavlja inicijalna konfiguracija težina. Težine su na početku treniranja inicijalizirane nasumično i nerijetko se dogodi da za istu konfiguraciju hiperparametara mreža ostvari različite performanse.

Veliki dio isprobanih konfiguracija je kao konačnu točnost uspio dobiti rezultat između 70% i 80%. Maksimalna dobivena točnost je 84%. Nikakve isprobane promjene u mreži nisu uspjele podići točnost iznad 84%, što ukazuje na neku dublju vrstu problema u ovom pristupu rješavanja.

Također, mreže s više skrivenih slojeva ne uspijevaju postići veću točnost, ali puno brže konvergiraju prema rješenju.

5.3.1. Vizualizacija aktivacija skrivenog sloja

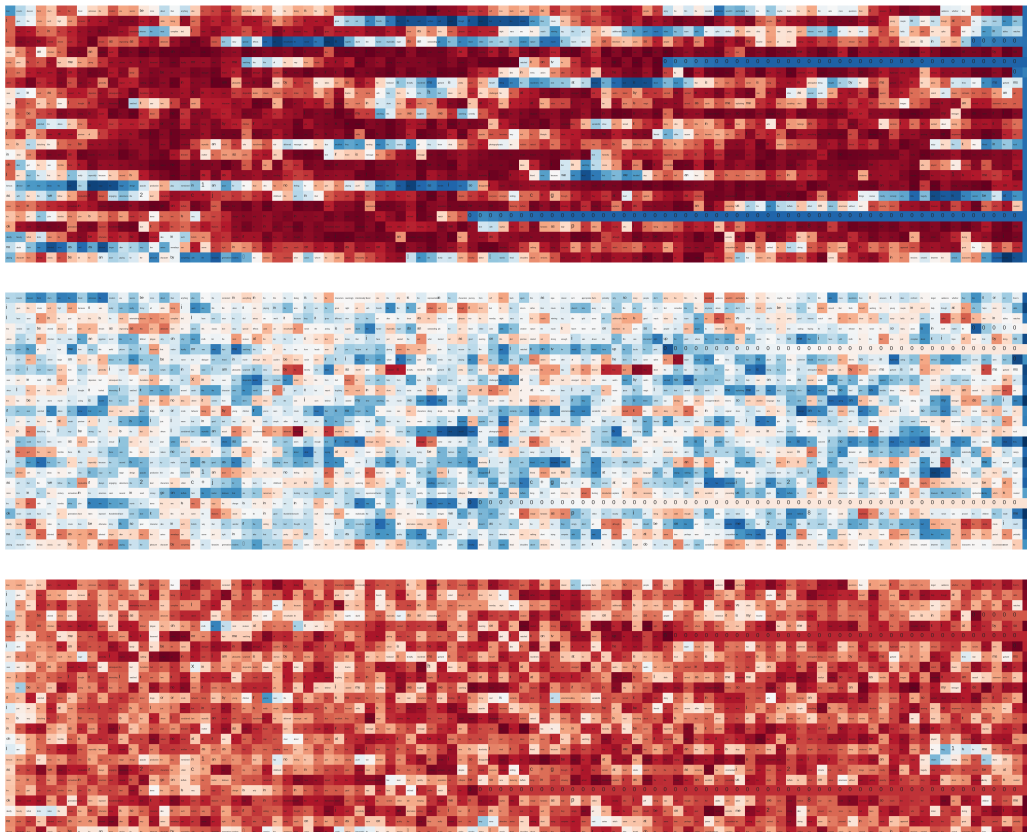
Jedan vrlo zanimljiv način analiziranja performansa povratne neuronske mreže i dobivanja intuicije o njezinim unutarnjim mehanizmima je vizualizacija aktivacija neurona skrivenog sloja [9].

Kao što je već objašnjeno, skriveni sloj sadrži određen broj neurona. Svaki od tih neurona tokom treniranja nauči prepoznavati nekakvu značajku sustava. Za slikovne podatke ta značajka je nama najčešće interpretabilna jer naš mozak sadrži veoma dobar sustav za prepoznavanje vizualnih uzoraka. To je najčešće crta, krivulja ili nekakav

jednostavniji geometrijski oblik i njegove razne rotacije. Veoma je lagano prikazati aktivacije tih neurona kao 2d-sliku na kojoj mi možemo prepoznati spomenute uzorke.

Za tekstne podatke situacija je *potpuno drugačija*. Osim što ne postoje lako interpretabilni uzorci analogni crtama i krivuljama (budući da je tekst kodiran u proizvoljno odabrane riječi), ne postoji niti lagan način da se vizualno prikažu! Ipak, postoji nekoliko metoda koje mogu pomoći u shvaćanju rada skrivenog sloja kada je u pitanju tekst.

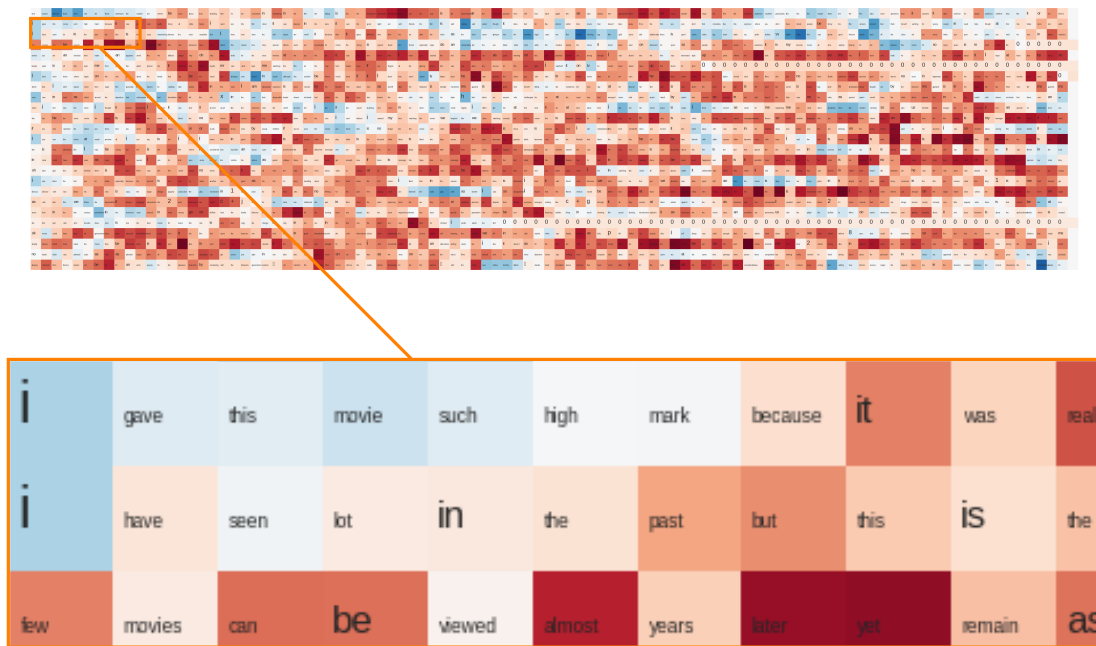
Jedna od metoda je prikazivanje *svih* aktivacija *određenog* neurona za *skupinu* ulaznih primjera za *svaki* trenutak u vremenu. Drugim riječima, metoda prikazuje kako neuron reagira na određeni ulaz. Iz zajedničkog prikaza više ulaznih primjera ponekad se može naslutiti generalno ponašanje. Konkretno, ako istrenirana neuronska mreža ima jedan skriveni sloj sa 100 neurona, to bi značilo da se može generirati 100 različitih 2d-prikaza aktivacija gdje X os označava trenutak u vremenu, Y os označava jedan određeni ulazni primjer (jednu recenziju), a svaki element tog grafa je obojan u skladu s vrijednosti njegove aktivacije. Na slici 5.7 su prikazane aktivacije nekih od 100 neurona mreže iz slike 5.6:



Slika 5.7: Prikaz aktivacija nekih od neurona istrenirane mreže

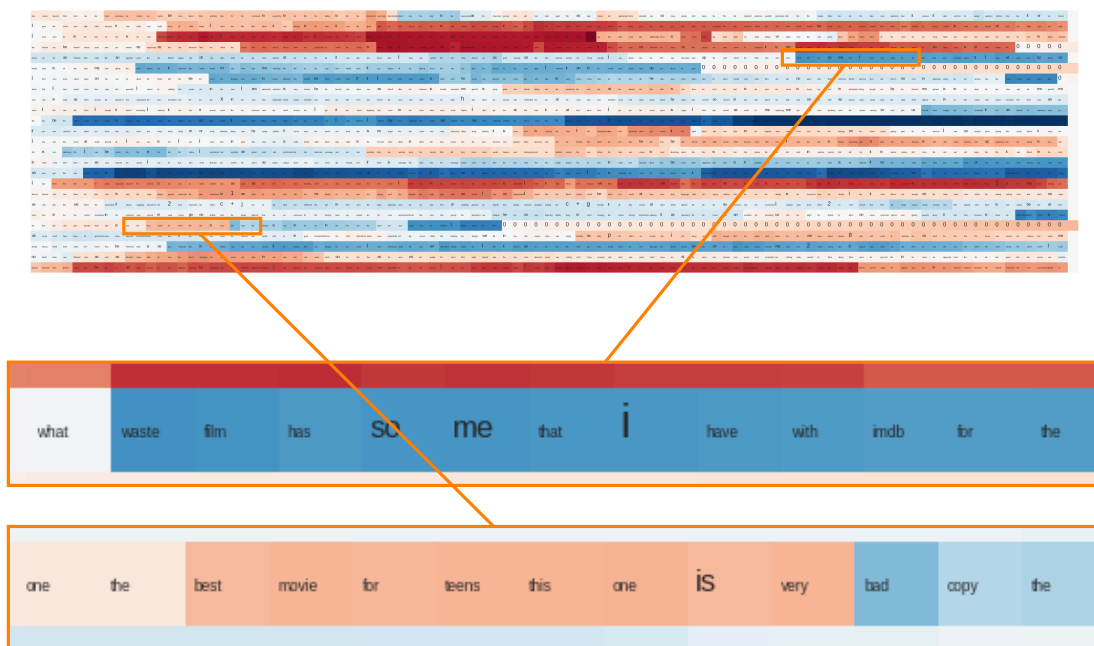
Svaki redak predstavlja jedan ulazni primjer, a svaki stupac predstavlja jednu riječ iz tog ulaznog primjera. Može se primjetiti da se prikazani neuroni ponašaju vrlo kaotično, bez nekakvih primjetnih uzoraka. Prvi i treći neuron imaju tendenciju reagirati na skoro svaku riječ pozitivno, s tim da je prvi neuron inertniji i nema tendenciju mijenjati svoje stanje.

Bližom inspekcijom (slika 5.8) može se vidjeti kako neuron reagira na točno određenu riječ.



Slika 5.8: Bliži pregled aktivacija jednog neurona

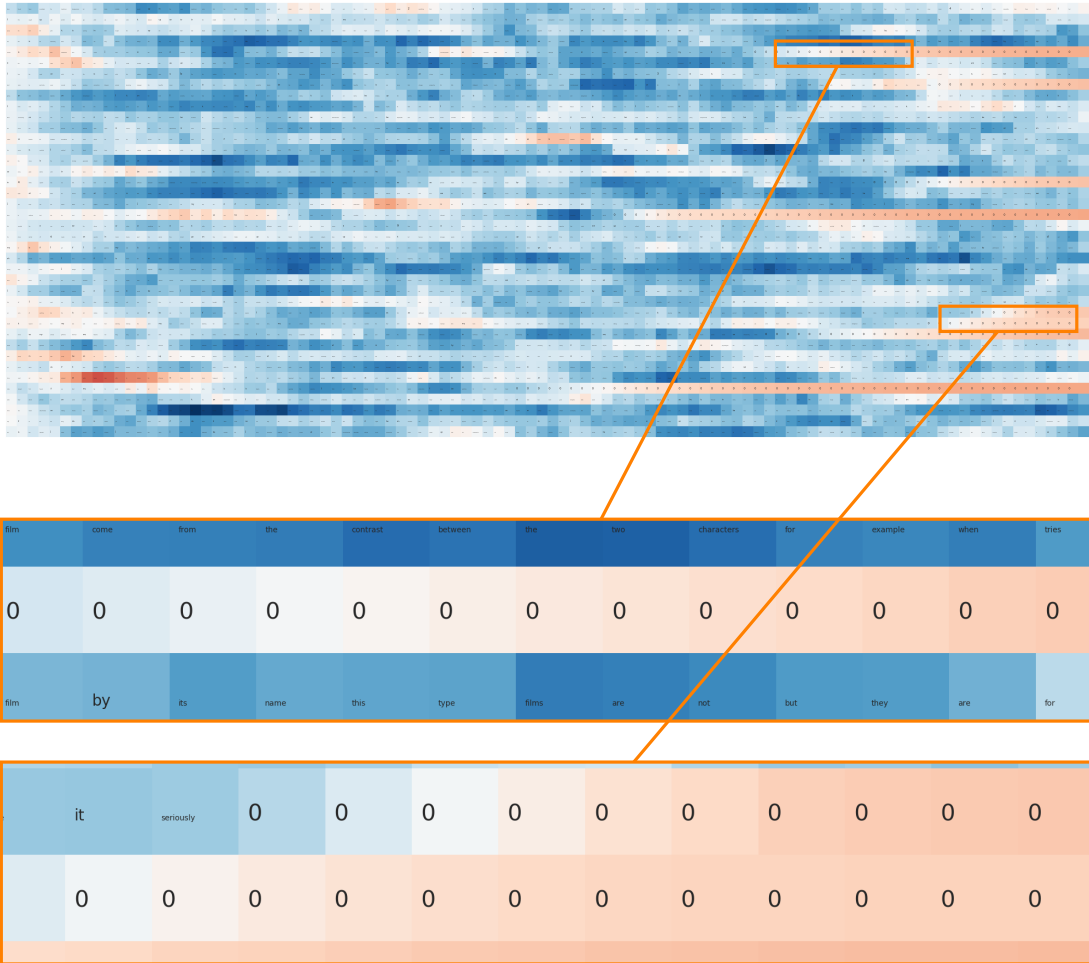
U spomenutoj mreži ima 100 neurona i praktički svi se ponašaju na ovakav kaotičan način što može biti začuđujuće s obzirom da je mreža uspjela ostvariti tako veliku točnost. Ipak, uz malo sreće, može se pronaći neuron iz čijih je aktivacija odmah očito da on radi nešto zanimljivo (slika 5.9).



Slika 5.9: Pregled aktivacije neurona koji je naučio prepoznavati interpretabilnu značajku

Prikazani neuron je naučio detektirati ključne riječi za izražavanje sentimenta. Naučio je pozitivno reagirati na riječi kao što su *liked*, *classic*, *finest*, *talented*... i negativno reagirati na riječi *horrible*, *boring*, *worst*, *bad*... te zadržavati svoje stanje do sljedeće relevantne riječi. Također, naučio ih je staviti na linearnu skalu tako za *classic* daje pozitivnu aktivaciju, ali manje pozitivnu od aktivacije za riječ *superb*. Ekvivalent vrijedi i za negativne sentimente.

Iako u nekim slučajevima značajka koju neuron modelira može biti interpretabilna, to ne znači da ona mora biti i korisna. Slika 5.10 prikazuje neuron koji *možda* modelira beskorisnu značajku. On je većinom u negativnom stanju, osim kada se počnu pojavljivati nule na kraju nekih recenzija, kada postaje sve pozitivniji kako je nula sve više. Drugim riječima, on je naučio brojati nule. Riječ *možda* ispred izjave da je to beskorisna značajka je tu s razlogom. Iako se to može činiti kao irelevantna značajka, moguće je da postoji nekakva korelacija između duljine recenzije i njihovog sentimenta koju je on uspio primijetiti.



Slika 5.10: Pregled aktivacije neurona koji je naučio brojati nule

6. Zaključak

U ovom radu cilj je bio napraviti analizu sentimenata na skupu podataka IMDb, kako bi se dobilo bolje razumijevanje različitih arhitektura dubokih neuronskih mreža. Analiza sentimenata odnosi se na korištenje obrade prirodnog jezika kako bi se identificirale i klasificirale subjektivne informacije u korisnički generiranim sadržajima. Razne metode strojnog učenja su se pokazale efektivnima u obradi prirodnog jezika. Nedavni rezultati ukazuju da duboko učenje i, općenito, neuronske mreže imaju veliku efikasnost u obradi prirodnog jezika.

Tokom izrade rada bilo je potrebno uložiti mnogo vremena na dobivanje teorijske podloge dubokog učenja koja je potrebna za konkretnu implementaciju u kodu. Na posljetku, kreiran je programski kod koji omogućava lagano evaluiranje raznih konfiguracija neuronske mreže. Kreirano je automatsko bilježenje performansa mreže i trenutnog modela, kao i kreiranje vizualizacije aktivacija skrivenog sloja. Kod je korišten za uspoređivanje različitih arhitektura povratnih neuronskih mreža, njihovih različitih ćelija i različitih modela ulaznih podataka. Zaključeno je da ćelija LSTM ima nadmoć nad običnom *tanh* povratnom ćelijom. Ulazni modeli distribuirane reprezentacije riječi daju daleko bolje rezultate od *one-hot* reprezentacije. Na skupu podataka IMDb dobivena je točnost od 84% s mrežom LSTM s dva skrivena sloja i ulaznim podacima dobivenim alatom Word2Vec.

LITERATURA

- [1] Eric Breck, Yejin Choi, i Claire Cardie. Identifying expressions of opinion in context. U *IJCAI*, svezak 7, stranice 2683–2688, 2007.
- [2] Sarath Chandar, Sungjin Ahn, Hugo Larochelle, Pascal Vincent, Gerald Tesaro, i Yoshua Bengio. Hierarchical memory networks. 2016. URL <https://arxiv.org/abs/1605.07427>.
- [3] An Mei Chen i R. Hecht-Nielsen. On the geometry of feedforward neural network weight spaces. U *Artificial Neural Networks, 1991., Second International Conference on*, stranice 1–4, Nov 1991.
- [4] Yejin Choi i Claire Cardie. Hierarchical sequential learning for extracting opinions and their attributes. U *Proceedings of the ACL 2010 conference short papers*, stranice 269–274. Association for Computational Linguistics, 2010.
- [5] Alex Graves, Greg Wayne, i Ivo Danihelka. Neural turing machines. *CoRR*, abs/1410.5401, 2014. URL <http://arxiv.org/abs/1410.5401>.
- [6] Klaus Greff, Rupesh Kumar Srivastava, Jan Koutník, Bas R. Steunebrink, i Jürgen Schmidhuber. LSTM: A search space odyssey. *CoRR*, abs/1503.04069, 2015. URL <http://arxiv.org/abs/1503.04069>.
- [7] Sepp Hochreiter i Jürgen Schmidhuber. Long short-term memory. *Neural Comput.*, 9(8):1735–1780, Studeni 1997. ISSN 0899-7667. doi: 10.1162/neco.1997.9.8.1735. URL <http://dx.doi.org/10.1162/neco.1997.9.8.1735>.
- [8] Ozan Irsoy i Claire Cardie. Opinion mining with deep recurrent neural networks. U *EMNLP*, stranice 720–728, 2014.

- [9] Andrej Karpathy. The unreasonable effectiveness of recurrent neural networks. URL <http://karpathy.github.io/2015/05/21/rnn-effectiveness/>.
- [10] Diederik P. Kingma i Jimmy Ba. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980, 2014. URL <http://arxiv.org/abs/1412.6980>.
- [11] Andrew Maas. Large movie review dataset. URL <http://ai.stanford.edu/~amaas/data/sentiment/>.
- [12] Michael Nielsen. Neural networks and deep learning. 2016. URL <http://neuralnetworksanddeeplearning.com/chap4.html>.
- [13] Cristopher Olah. Neural networks, types and functional programming, . URL <http://colah.github.io/posts/2015-09-NN-Types-FP/>.
- [14] Cristopher Olah. Understanding lstm networks, . URL <http://colah.github.io/posts/2015-08-Understanding-LSTMs/>.
- [15] Cristopher Olah. Visualizing representations, . URL <http://colah.github.io/posts/2015-01-Visualizing-Representations/>.
- [16] Richard Socher, Alex Perelygin, Jean Y Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, i Christopher Potts. Recursive deep models for semantic compositionality over a sentiment treebank. U *Proceedings of the conference on empirical methods in natural language processing (EMNLP)*, svezak 1631, stranica 1642. Citeseer, 2013.
- [17] Duyu Tang, Bing Qin, Xiaocheng Feng, i Ting Liu. Target-dependent sentiment classification with long short term memory. *CoRR*, abs/1512.01100, 2015. URL <http://arxiv.org/abs/1512.01100>.
- [18] Bret Victor. Up and down the ladder of abstraction. 2011. URL <http://worrydream.com/LadderOfAbstraction/>.
- [19] Theresa Wilson, Janyce Wiebe, i Paul Hoffmann. Recognizing contextual polarity in phrase-level sentiment analysis. U *Proceedings of the conference on human language technology and empirical methods in natural language processing*, stranice 347–354. Association for Computational Linguistics, 2005.

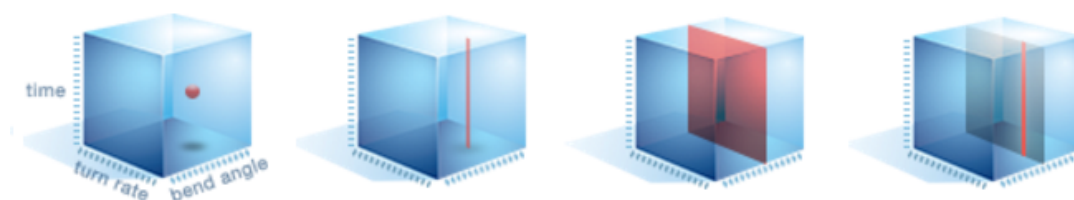
- [20] Jason Yosinski, Jeff Clune, Yoshua Bengio, i Hod Lipson. How transferable are features in deep neural networks? *CoRR*, abs/1411.1792, 2014. URL <http://arxiv.org/abs/1411.1792>.

Dodatak A

Daljni razvoj neuronskih mreža

Duboko učenje je, unatoč izvanrednom uspjehu, mlado područje. Postoje mnoge indikacije da su trenutni modeli samo slijepa nabadanja u mraku; kada i naiđu na neki koristan rezultat, ne zna se kako su došli do njega, niti kako bi mogli iskoristiti to znanje da dođu do drugih rezultata. Potrebno je rasvijetliti situaciju formalnijim opisom učenja.

Zanimljiv pogled na duboko učenje se može dobiti razmatranjem ljudskog procesa učenja. Srž ljudske inteligencije bi se mogla objasniti s dvije stvari: pronalaženjem uzoraka u podacima i apstrahiranjem istih (skrivanjem nebitnog). Skrivanje nebitnog je ključan pojam zbog naše ograničene moći procesiranja podataka. Omogućava nam da alociramo više resursa prema procesiranju bitnijih podataka i zanemarimo šum. Rekurzivnim primjenjivanjem ovih pravila dobije se veliko stablo različitih razina apstrakcija. To je ono s čime se i neki moderni pristupi strojnog učenja muče: kako generirati tu ogromnu hijerarhiju apstrakcija kojom mi tako lagano baratamo?



Slika A.1: Proces učenja kao funkcija iteriranja po presjecima podataka

Drugi zanimljiv pogled na proces učenja je rezultat generalizacija mehanizma rada mreža LSTM. Čelija LSTM se može interpretirati kao jedinica za procesiranje podataka koja ima svoju unutarnju *memoriju* koja je *izolirana* od ostatka procesiranja. Tu su dvije riječi istaknute: *memorija* i *izolirana*.

Memorija je bitan koncept jer omogućava ponovnu iskoristivost neke informacije.

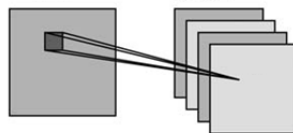
Svi današnji sustavi i programi koje smo kreirali su napisani da budu ponovno iskoristivi: napisani su *modularno*. Unaprijedne neuronske mreže i povratne neuronske mreže s ćelijom *tanh* nemaju tu modularnu karakteristiku. Ćelija LSTM bi bila postavljena malo više na toj "skali modularnosti", no niti ona ne omogućava lagano ponovno iskorištavanje. Moguće je kreirati višeslojnu povratnu mrežu, ali ne i formalno odrediti kada će se koja informacija/značajka iskoristiti (problem svih mreža, a time i mreže LSTM, je taj što su značajke neinterpretabilne, više o tome kasnije).

Druga istaknuta riječ je *izoliranost*. Svaki efikasan sustav za procesiranje podataka je dovoljno kompleksan da zahtijeva neku vrstu međuspremnik, odnosno memorije. Drugim sustavima, po definiciji, ta memorija međuspremnik nije potrebna. Time se ovaj proces zaključivanja može dovesti u vezu sa skrivanjem nebitnog jer je program samo razina apstrakcije koja, za ostale sustave, skriva nebitne detalje poput međuspremnik tog programa.

Iz spomenutih stvari se može naslutiti da ovaj proces zaključivanja vodi prema jednom poznatom konceptu u računarskoj znanosti: *objektno orijentiranoj paradigmi (OOP)*. Iako ne postoji formalno opisana arhitektura neuronske mreže koja implementira objektno orijentiranu paradigmu, mnoga istraživanja kreću u tom smjeru i mnoge nove arhitekture neuronskih mreža implicitno implementiraju neke dijelove filozofije OOP-a.

Najviše potencijala pokazuje koncept *Neuronskih Turingovih strojeva* (engl. *Neural Turing Machines*) [5]. Neuronski Turingov stroj opisuje povratnu neuronsku mrežu koja može čitati i pisati u određene blokove memorije. Efektivno modelira Turingov stroj koji je u potpunosti diferencijabilan. Postiže bolje rezultate od LSTM-a te čak uspijeva naučiti primitivne algoritme (kopiranje podataka, *prisjećanje* podataka, sortiranje podataka i sl.) Neuronski Turingov stroj, doduše, ne koristi nikakvu vrstu hijerarhija u svojoj implementaciji. Nedavni rezultati [2] počinju rješavati i taj problem.

Konvolucijske neuronske mreže djelomično pogađaju ideju objektno orijentiranog programiranja sa svojim slojevima neurona koji sukcesivno modeliraju više razine apstrakcije. Osim u nekim ograničenim slučajevima, [20], ne uspijevaju biti proširive.



Slika A.2: Sloj konvolucijske neuronske mreže

Proces mijenjanja razine apstrakcije je opisan u izvanrednom članku o *ljestvicama*

apstrakcije [18]. On bi se mogao opisati kao uzorkovanje $n - 1$ dimenzionalnog presjeka n dimenzionalne hiperkocke.

To je točno ono što je bilo napravljeno u prikazu aktivacija neurona skrivenog sloja. Gledali su se određeni presjeci tog *tenzora podataka* iz kojih su se prepoznali određeni uzorci. Gledanjem riječi po riječi jednog neurona jednog ulaznog primjera teško bi se prepoznali uzorci. Vizualiziranjem cijelog skupa podataka (što u ovom slučaju nije bilo moguće zbog dimenzionalnosti istih) bi dovelo do ogromnog šuma iz kojeg ne bi uspjeli prepoznati uzorke.

To se opet može povezati s konvolucijskim neuronskim mrežama koje uzorkuju $2d$ presjek podataka. Može li se taj proces konvolucije generalizirati na više dimenzija?

Problem efikasnog učenja neuronskih mreža se može gledati iz jedne potpuno druge perspektive. Nedokučivi izgled hiperplohe greške može se napasti sa strane matematike, odnosno *apstraktne algebre*. Poznato je da različite konfiguracije težina neuronskih mreža mogu davati približno jednake rezultate [15]. Postavlja se pitanje, koliko ima takvih konfiguracija težina? Mogu li se iz njihovih struktura zaključiti neke korisne informacije? Može li se suziti prostor pretrage?

Pitanja se mogu formalizirati: kakva je struktura hiperplohe greške? Prema kojim njezinim transformacijama je izlaz mreže invarijantan? Može li se eliminirati dio tog prostora u potrazi za minimumom? Trenutno popularni algoritmi optimizacije se bave iterativnim metodama koji na temelju *lokalne strukture hiperplohe* pokušavaju doći do informacije o minimumu. Prethodni način razmatra *globalnu strukturu hiperplohe*. Ne postoji previše radova na tu temu [3], no to nikako ne umanjuje vrijednost pitanja koja ona sadrži.

Ovo su samo neki od smjerova u kojima se može ići s daljnjim razmatranjima rada neuronskih mreža. Postoji ogroman skup pitanja iz različitih domena znanosti koja mogu dati uvid u unutarnje mehanizme ne samo neuronskih mreža, nego i ljudskog procesa učenja.

Primjena modela dubokog učenja na analizu sentimenata

Sažetak

Uzevši u obzir eksponencijalni rast dostupnih podataka generiranih u cijelom svijetu, postoji rastući interes u stvaranje modela sposobnih za analizu podataka koji imaju semantički nepoznat kontekst. Ovaj rad se fokusira na istraživanje i analizu teorijske osnove koja stoji iza modela zvanih *povratne neuronske mreže* (engl. *recurrent neural networks, RNN*) kroz konkretnu implementaciju na problemu semantičke analize. Poseban slučaj povratnih neuronskih mreža, *mreže s dugotrajnim kratkoročnim pamćenjem* (engl. *Long short-term memory networks, LSTM*) su implementirane na problemu klasifikacije sentimenata na skupu podataka recenzija filmova. Testirane su i uspoređene mnogobrojne arhitekture i konfiguracije hiperparametara spomenutih neuronskih mreža. Značajano poboljšanje performansi je primjećeno s dubokim višeslojnim LSTM mrežama, u za razliku od rezultata dobivenih s plitkim mrežama.

Ključne riječi: obrada prirodnog jezika, strojno učenje, neuronske mreže, duboko učenje, analiza sentimenata, LSTM, povratne neuronske mreže

Application of Deep Learning for Sentiment Analysis

Abstract

Given the exponential growth of amount of available data generated worldwide, there has been a rising interest in the creation of models capable of analyzing the data without knowing the semantic context. This paper focuses on exploration and analysis of the theoretical foundation that stands behind models called *recurrent neural networks* through concrete implementation on a problem of semantic analysis. Special case of recurrent neural networks, *long short-term memory networks* (LSTM networks) were implemented on a task of semantic classification on the IMDb dataset. Various architectures and hyperparameter configurations were tested and compared on the said neural networks. Significant performance boost was detected with deep multi-layer LSTM networks, compared to performance with shallow ones.

Keywords: natural language processing, machine learning, neural networks, deep learning, sentiment analysis, LSTM, recurrent neural networks