*Elektronski fakultet u Nišu*

*Katedra za Računarstvo i informatiku*

Određivanje sentimenta komentara korišćenjem neuronske mreže

Soft kompjuting

*Student:* ***Uroš Janjić***, ***607***

*Mentor:* ***Aleksandar Milosavljević***

Sadržaj

[Uvod 3](#_Toc518656614)

[Neuronske mreže 3](#_Toc518656615)

[Određivanje sentimenta korišćenjem neuronske mreže 4](#_Toc518656616)

[Kreiranje modela neuronske mreže 4](#_Toc518656617)

[Colaboratory 7](#_Toc518656618)

[Upotreba 7](#_Toc518656619)

[Rezultat 9](#_Toc518656620)

[Zaključak i dalji rad 9](#_Toc518656621)

# Uvod

Klasifikacija teksta i analiza sentimenta predstavljaju jedan od problema oblasti koja se naziva procesuiranje prirodnog jezika. Ova oblast spaja nekoliko nauka poput računarstva, veštačke inteligencije i lingvistike. Problem koji rešava klasifikacija teksta je klasifikacija nepoznatih dokumenata u korisnički preddefinisane klase. Analiza sentimenta je specijalan slučaj klasifikacije teksta u kome postoje dve klase – pozitivna i negativna. Na ovaj način ovo predstavlja binarnu klasifikaciju. Međutim postojanje dve klase ne olakšava posao, jer se pri ovoj klasifikaciji radi sa ljudskim emocijama i osećanjima. Analiza sentimenta je široko primenjena u istraživanju mišljenja kupaca, odnosno njihovih kritika i odgovora u anketama, online i socijalnih medija, a ovakve informacije su izuzetno važne za apsolutno bilo koji oblik internet poslovanja.

Uopšteno govoreći, analiza sentimenta ima za cilj da utvrdi stav pojedinca u vezi sa nekim temama ili emocionalne reakcije na neki dokument, interakciju ili događaj. Stav može da bude sud ili ocena, emocionalno stanje (autora ili govornika) ili emocionalna komunikacija (emocionalne namere autora ili sagovornika). Poenta ovog posla je klasifikacija polariteta datog teksta, odnosno određivanje da li je mišljenje pozitivno ili negativno.

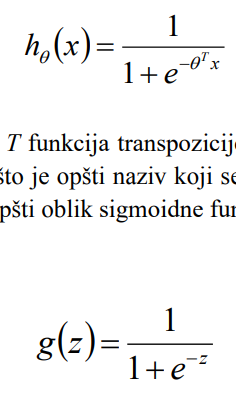
Zavisno od jezika za koji se analiza sentimenta radi, sam proces može biti manje ili više kompleksan, ali svakako inače podrazumeva pronalaženje zakonitosti u korišćenju prirodnog jezika i njihovo prepoznavanje u tekstu koji se analizira. Razvijanje sistema sa ovom namenom je uglavnom obiman posao koji zahteva saradnju ljudi iz različitih oblasti (lingvistika i računarstvo). Cilj ovog rada je da pokaže mogućnost upotrebe neuronskih mreža za analizu sentimenta teksta i time drastično pojednostavi proces razvijanja softvera ove namene. Dodatna predost ovog pristupa je mogućnost treniranja nad specifičnim skupom podataka, čime se dobija analizator za jednu užu oblast, što povećava preciznost u odnosu na univerzalne analizatore sentimenta.

# Neuronske mreže

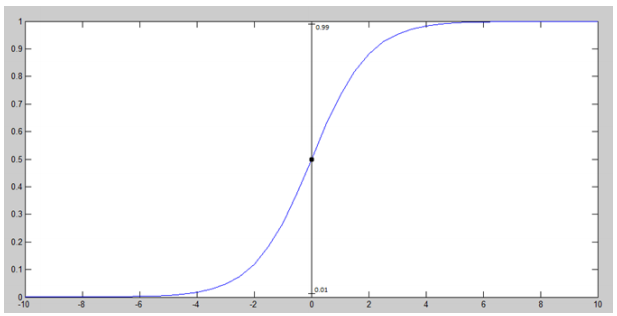
Algoritmi neuronskih mreža su nastali kao pokušaj da se oponaša način na koji ljudski mozak uči, tj. način na koji ljudski mozak klasifikuje i spoznaje stvari. Računske operacije koje obavljaju neuronske mreže su zahtevne, stoga je tek relativno skoro napredak tehnologije omogućio njihovo intenzivnije korišćenje. Oponašanje mozga je jako zahtevan proces, s obzirom na širok dijapazon mogućnosti i količine podataka koje mozak svakodnevno obrađuje.

Neuronske mreže su razvijene tako da simuliraju neurone, tj. mreže neurona mozga. Neuron je po svojoj suštini jedinica izračunavanja, koja dobija određen broj ulaza, obrađuje ih i izlaz prosleđuje dalje kako bi se izračunavanje nastavilo. Ovo je princip po kome se odvija ceo čovekov misaoni proces. Neuroni koji rade izračunavanja, prosleđuju poruke koje su rezultat ulaza koje su primili. Struktura mreže se razlikuje po broju slojeva. Prvi sloj se naziva ulazni, a poslednji izlazni, dok su slojevi između skriveni.

Ulazni podaci modela su dati vektorom x, a težinski faktori vektorom θ. Težinski faktori su od ključnog značaja za prilagođavanje modela podacima i generalno za ceo proces učenja. Obično se nasumično inicijalizuju i menjaju posle svake iteracije algoritma treniranja. Funkcija aktivacije perceptrona definiše njegov izlaz u zavisnosti od ulaza. Rezultat koji se formira na izlazu obeležava se sa hθ. Funkcija aktivacije koja se najčešće koristi je sigmoidna funkcija:



gde je x ulaz, θ težinski faktori, a T funkcija transpozicije. Grafik ove funkcije prikazan je na sledećoj slici.



Treniranje neuronske mreže podrazumeva određivanje težina potega između neurona. Logika je uporediti predviđeni izlaz neuronske mreže i tačan izlaz za uzorak podataka na kome se mreža trenira. Nakon izračunavanja razlike, prilagoditi težine potega tako da se minimizuje greška pri predikciji. Što je skup trening podataka veći i što se više pota ponovi algoritam treniranja, preciznost neuronske mreže biće veća, ali drastično se povećava i posao treniranja, tako da je neophodno napraviti balans.

# Određivanje sentimenta korišćenjem neuronske mreže

Ideja je kreirati neuronsku mrežu koja kao ulaz prima komentar korisnika na neki proizvod ili temu, a kao izlaz daje procenu da li je komentar bio pozitivan ili negativan.

## Kreiranje modela neuronske mreže

Za treniranje i validaciju iskorišćen je skup podataka u vidu komentara sa sajtova imdb, yelp i amazon, preuzet sa interneta i dostupan na linku:

<https://archive.ics.uci.edu/ml/datasets/Sentiment+Labelled+Sentences>

Podaci su smešteni u tekstualne datoteke i predstavljeni su kao niz komentara sa pomenutih sajtova, nakon kojih se nalazi binarna oznaka (0 ili 1) koja predstavlja manuelno određen sentiment komentara – jedinica za pozitivan, nula za negativan sentiment. Od celog skupa podataka, 90% je iskorišćeno u treniranju i 10% pri validaciji sistema.

It is very comfortable on the ear. 1

Good case, Excellent value. 1

Great for the jawbone. 1

I'm a bit disappointed. 0

Doesn't hold charge. 0

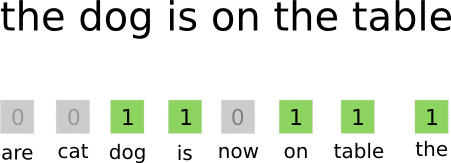
This is a simple little phone to use, but the breakage is unacceptible. 0

This product is ideal for people like me whose ears are very sensitive. 1

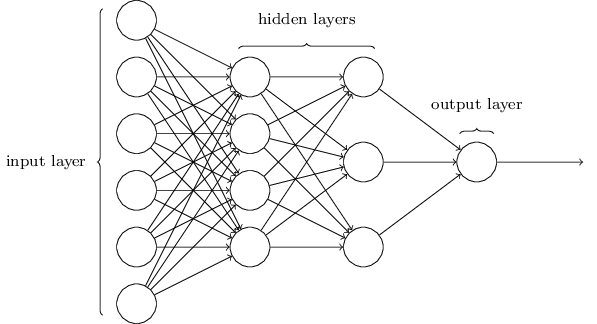
It is unusable in a moving car at freeway speed. 0

I have two more years left in this contract and I hate this phone. 0

Nad ovim skupom komentara vrši se tokenizacija, odnosno određuju se pojedinačne reči iz rečenica, a nakon toga se te reči svode na zajedničku osnovu kako bi se uklonili duplikati. Ove jedinstvene reči sistem koristi za predstavljanje komentara u obliku „bag of words“, odnosno niz bitova koji označavaju da se neka reč iz skupa svih prepoznatih reči nalazi ili ne nalazi u komentaru koji se analizira.



Ovaj postupak se ponavlja za svaku rečenicu iz trening skupa. Iz komentara se izdvajaju pojedinačne reči, dalje se na osnovu njih kreira „bag of words“ (binarni niz) za taj komentar i njemu se pridružuje očekivani izlaz, u slučaju ovog sistema 1 za pozitivan sentiment i 0 za negativan. Neuronska mreža koja se trenira i koja će biti korišćena za određivanje sentimenta komentara ima ulazni sloj, skriveni i izlazni sloj. Ulaznih čvorova je onoliko koliko reči podrazumeva skup svih pojedinačnih (dakle isključujući duplikate) reči u trening skupu. U skrivenom sloju je za ovaj primer konfigurisano 20 čvorova. Izlazni sloj sadrži samo čvorova koliko je klasa u koje se komentari razvrstavaju, u ovom slučaju to su klase „pozitivan“ i „negativan“ sentiment.



Sam proces treniranja modela neuronske mreže podrazumeva sledeće korake:

Inicijalizacija težina sinapsi u modelu nasumičnim vrednostima i postavljanje vrednosti srednje greške (mean error) na maksimalnu vrednost (vrednost srednje greške kreće se od 0.0 do 1.0).

Napomena: Vrednost srednje greške podrazumeva srednju razliku između predikcije sistema i tačnog izlaza. Ovu vrednost koristimo za prilagođavanje težina sinapsi u cilju dobijanja boljih rezultata (time što pokušavamo da smanjimo srednju grešku modela).

Postoji unapred definisan broj iteracija algoritma za treniranje i za svaku od tih iteracija potrebno je na ulaz modela dovesti niz „bag of words“ za sve komentare iz trening skupa (binarnu matricu veličine *broj različitih reči u trening skupu* x *broj komentara u trening skupu*). Odrediti vrednost čvorova u skrivenom sloju time što se za vrednost ulaznih čvorova i vrednost težina sinapsi koje povezuju ulazni i skriveni sloj izračuna vrednost sigmoidne funkcije. Na isti način odrediti vrednost čvorova izlaznog sloja pomoću sinapsi sa skrivenim slojem.

Srednju grešku odrediti kao srednju vrednost sume rezlika između rezultata predikcije i tačnih izlaza za komentare iz trening skupa. Backpropagation proces počinjemo određivanjem smera u kome je potrebno promeniti težine sinapsi, najpre između izlaznog i skrivenog sloja, pa onda unazad do sinapsi između skrivenog i ulaznog sloja. Kada se utvrdi smer promene, težine sinapsi se menjaju unapred predefinisanim korakom *alfa* koji za slučaj ovog rada ima vrednost 0.1.

Krajnji rezultat procesa treniranja je potpun model neuronske mreže, što podrazumeva kolekciju težina svih sinapsi. Uz neuronsku mrežu čuva se i skup svih pojedinačnih reči identifikovanih u trening skupu, kako bi se na osnovu njih kreirao „bag of words“. Korišćenjem ova dva dela moguće je vršiti analizu sentimenta rečenica.

## Colaboratory

Kako je sam proces treniranja neuronske mreže prilično obiman u smislu procesuiranja, jer podrazumeva ogroman broj množenja matrica velikih dimenzija, za obavljanje ovog zadatka potreban je ozbiljan hardver, a idealno i NVIDIA grafička kartica kako bi se GPGPU (General-purpose graphics processing units) pristupom izračunavanja obavila što brže. Kako dobar deo računara ne ispunjava ove uslove, Google Colaboratory može biti od pomoći.

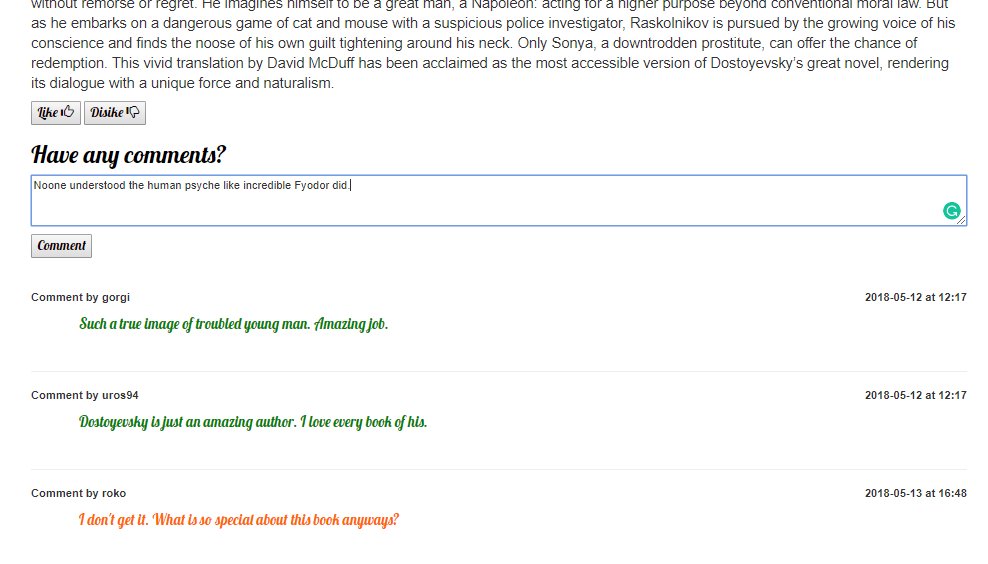
Colaboratory je Google istraživački projekat, stvoren da pomogne u širenju obrazovanja i istraživanja oblasti mašinskog učenja. To je okruženje Jupiter notebook-a koje ne zahteva nikakvo dodatno podešavanje za korišćenje i u potpunosti se izvršava u oblaku. Colaboratory podatke čuva na Google disku i moguće je te podatke deliti kao bilo koji Google dokument. Colaboratory je besplatan za korišćenje. Moguće je izvršavati kod napisan u Python 2 i Python 3 jeziku uz GPU akcelerator izvršavanja.

## Upotreba

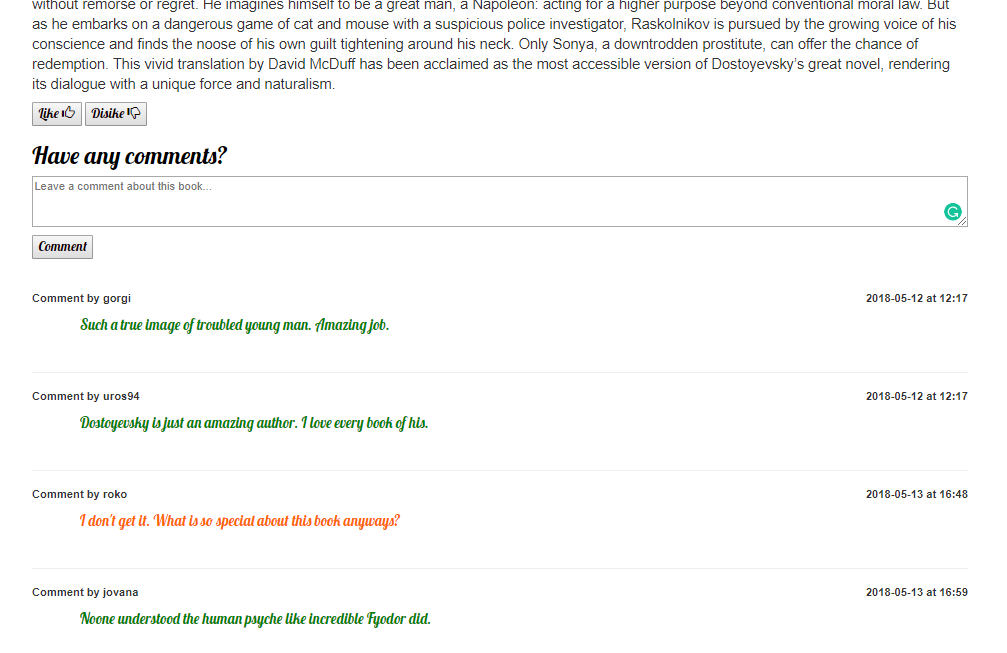
Na osnovu kreiranog modela neuronske mreže i skupa reči pomoću kog se kreira „bag of words“ moguće je funkcijama za analizu odrediti sentiment bilo kog komentara. Na Github-u je dostupan projekat <https://github.com/uros94/sentiment_analysis> koji sadrži već kreiran model zajedno sa funkcijma za analizu teksta, ali i skripte za treniranje i validaciju modela neuronske mreže na osnovu trening podataka koji bolje ilustruju budući način upotrebe analizatora. Odnosno, postoji gotov model treniran nad skupom podataka koji se takođe mogu naći u okviru pomenutog projekta na Github-u, ali moguće je i trenirati novi model tako da on bolje zadovoljava uslove svake specifične primene. U poslednjoj rečenici leži i odgovor na to koja je najveća prednost ovakvog pristupa semantičkoj analizi teksta – na ovaj način se lako dolazi do efikasnih analizatora za specifične primene.

Za korišćenje priloženog gotovog modela neuronske mreže potrebno je preuzeti fajlove *commentSemantics.py* i *commentSemanticsConstants.py*. Importovanjem ovih fajlova u bilo koji sistem moguće je obezbediti analizu sentimenta. *commentSemanticsConstants.py* sadrži podatke o težinama sinapsi, skupu reči i klasama i zapravo predstavlja podatke neophodne za reprezentaciju prethodno kreiranog modela neuronske mreže, a *commentSemantics.py* importuje pomenuti faj sa konstantama i sve pomoćne funkcije za analizu komentara. Za određivanje sentimenta bilo kog stringa, potrebno je samo pozvati funkciju *classify(comment)* iz *commentSemantics.py*. Ona ima jedan argument i to je tekst koji se analizira, a rezultat je niz čiji je prvi element vrednost 1 u slučaju pozitivnog i 0 u slučaju negativnog sentimenta, a drugi procena pouzdanosti predikcije na skali od 0.0 do 1.0.

Na sledeće dve slike je ilustracija upotrebe.



Uneti tekst se analizira i zavisno od sentimenta boji zeleno ili narandžasto.



Kako postoji velika razlika u smislu različitih reči u različitim kontekstima, za precizniju analizu je najbolje koristiti prilagođen analizator. Kreiranje prilagođenog modela neuronske mreže može se postići korišćenjem skripti networkTraining.py ili networkTraining&Validation.py. Prva skripta sve podatke iz fajlova priloženih kao trening set koristi za obučavanje mreže, a druga priložene podatke deli na skup za trening i skup za validaciju u odnosu 9:1. Dovoljno je u skripti izmeniti poziv funkcije *read\_data(path)* tako da *path* odgovara putanji do fajlova sa trening podacima i skripta će generisati model mreže. Po defaultnim podešavanjima algoritam treniranja se ponavlja 50000 puta sa korakom alfa=0.1. Kreirani model mreže biće zapamćen u json formatu kao zaseban fajl, a sadržaj tog json zapisa potrebno je kopirati u *commentSemanticsConstants.py* kako bi analizator koristio novokreirani model.

## Rezultat

Ovakav analizator zasnovan na određivanju sentimenta korišćenjem modela neuronske mreže dolazi do rešenja množenjem ulaznog vektora koji predstavlja rečenicu matricama težina sinapsi, tako da brzina dobijanja rezultata zavisi od veličina ovih matrica, ali za primer ovde opisan srednja brzina dolaženja do rešenja na prosečnom računaru je nešto ispod 0.1s.

Na primeru trening skupa podataka priloženih u okviru Github projekta, zavisno od načina podele podataka na trening i skup za validaciju, preciznost predikcije se kreće između 80% i 90%.

# Zaključak i dalji rad

Ovaj pristup je nezavisan od prirode teksta koji se analizira i čak nezavisan od jezika nad kojim se primanjuje sve dok postoji parser za taj jezik. Na ovaj način je moguće brzo i lako kreirati analizatore specifične namene, a pritom nije neophodno imati puno znanja iz oblasti lingvistike.

Smer potencijalnog daljeg rada bila bi primena ove metode za razvijanje analizatora teksta na srpskom jeziku. Srpski jezik pripada grupi jezika koji su jako zahtevni za analizu zbog svoje gramatike i nedostatka efikasnih parsera, ali uz povećanu redundantnost pojmova zbog različitih oblika istih reči verujem da je moguće ostvariti rezultate zadovoljavajuće za kreiranje upotrebljivog sistema za određivanje sentimenta.

Još jedna ideja za moguće unapređivanje sistema je dodavanje analizatora funkcija reči u rečenici (subjekat, objekat, predikat...). Za analizu sentimenta komentara je najvažniji predikat kao nosioc emocije ili stava koji korisnik izražava, a takođe i svi atributi u rečenici. Dodavanjem analizatora koji prepoznaje reči sa ovom funkcijom u rečenici, moguće je poboljšati performanse.