Tehničko veleučilište u Zagrebu

Stručni studij računarstva

**PREPOZNAVANJE TEKSTA RAČUNALNIM VIDOM**

Seminarski rad

Metodologija stručnog i istraživačkog rada

Nositelj: dr. sc. Lidija Tepeš Golubić

Student: Stjepan Salopek

Zagreb, veljača 2021.

Sažetak

Seminarski rad metodološki će prikazati izradu aplikacije prepoznavanja teksta računalnim vidom. Pojam računalnog vida obraditi će se kroz tehnologiju umjetne inteligencije, točnije strojnog učenja prolaskom kroz potencijal područja te tržišnu vrijednost. Obraditi će se teorija vezana za kreiranje aplikacije strojnog učenja i navesti primjeri iz suvremenog svijeta u kojima se ono može koristiti.

Nadalje, kroz implementaciju će se proći kroz detaljniju teoriju potrebnu te opcije i odabir alata programskog jezika. Na kraju će se navesti rezultati i usporediti sa suvremenim aplikacijama.

Ključne riječi: strojno učenje, neuronske mreže, prepoznavanje teksta

SADRŽAJ

[Sažetak 1](#_Toc81347275)

[Sadržaj 2](#_Toc81347276)

[Popis tablica 2](#_Toc81347277)

[Popis slika 3](#_Toc81347278)

[1 UVOD 4](#_Toc81347279)

[2 O UMJETNOJ INTELIGENCIJI 5](#_Toc81347280)

[2.1 Osnovne definicije 5](#_Toc81347281)

[2.2 Strojno učenje na tržištu 5](#_Toc81347282)

[2.3 Potreba za strojnim učenjem 7](#_Toc81347283)

[2.4 Implementacija strojnog učenja neuronskim mrežama 8](#_Toc81347284)

[2.5 Vrste neuronskih mreža 10](#_Toc81347285)

[2.6 Primjeri iz prakse 11](#_Toc81347286)

[3 STVARANJE APLIKACIJE 13](#_Toc81347287)

[3.1 Odabir tehnologija 13](#_Toc81347288)

[3.2 Teorija i funkcionalnosti potrebni za izradu 14](#_Toc81347289)

[3.2.1 Tenzori 14](#_Toc81347290)

[3.2.2 Uobičajena propagacija neuronske mreže 14](#_Toc81347291)

[3.2.3 Unazadna propagacija neuronske mreže 15](#_Toc81347292)

[3.2.4 Stvaranje klase za strojno učenje 16](#_Toc81347293)

[3.2.5 Implementiranje klase za strojno učenje u Pythonu 17](#_Toc81347294)

[3.3 Rad sa kreiranom neuronskom mrežom 17](#_Toc81347295)

[3.4 Rad i analiza rezultata 18](#_Toc81347296)

[4 PRAKTIČNI DIO 19](#_Toc81347297)

[4 ZAKLJUČAK 20](#_Toc81347298)

[Popis literature 21](#_Toc81347299)

Popis tablica

[Tablica 1 Srednje prosječne plaće inženjera strojnog učenja i podataka po državama 5](#_Toc63094049)

Popis slika

[Slika 1 Odnos umjetne inteligencije i strojnog učenja 7](https://d.docs.live.net/00a34c8f5e266a1e/Dokumenti/Prepoznavanje%20teksta%20računalnim%20vidom.docx#_Toc63549710)

[Slika 2 Promjene u plaći inženjera strojnog učenja i podataka po državama uspoređeno sa istima prije tri godine [6] 9](#_Toc63549711)

[Slika 3 Primjer prepoznavanja brojeva neuronskom mrežom 11](#_Toc63549712)

[Slika 4 Primjer jednostavne *feedforward* neuronske mreže 12](https://d.docs.live.net/00a34c8f5e266a1e/Dokumenti/Prepoznavanje%20teksta%20računalnim%20vidom.docx#_Toc63549713)

[Slika 5 Primjer prepoznavanja lica strojnim učenjem 13](https://d.docs.live.net/00a34c8f5e266a1e/Dokumenti/Prepoznavanje%20teksta%20računalnim%20vidom.docx#_Toc63549714)

[Slika 6 Primjer strojnog učenja u Google tražilici 14](https://d.docs.live.net/00a34c8f5e266a1e/Dokumenti/Prepoznavanje%20teksta%20računalnim%20vidom.docx#_Toc63549715)

[Slika 7 Usporedba Google trendova za pojmove TensorFlow (crveno) te PyTorch (plavo) 15](https://d.docs.live.net/00a34c8f5e266a1e/Dokumenti/Prepoznavanje%20teksta%20računalnim%20vidom.docx#_Toc63549716)

[Slika 8 Primjer uobičajene propagacije mreže 16](#_Toc63549717)

[Slika 9 Ovisnost varijabli neuronske mreže 17](https://d.docs.live.net/00a34c8f5e266a1e/Dokumenti/Prepoznavanje%20teksta%20računalnim%20vidom.docx#_Toc63549718)

[Slika 10 Model za prepoznavanje teksta 18](https://d.docs.live.net/00a34c8f5e266a1e/Dokumenti/Prepoznavanje%20teksta%20računalnim%20vidom.docx#_Toc63549719)

[Slika 11 Pojednostavljenje klase za strojno učenje 19](https://d.docs.live.net/00a34c8f5e266a1e/Dokumenti/Prepoznavanje%20teksta%20računalnim%20vidom.docx#_Toc63549720)

[Slika 12 Implementacija funkcije za optimizaciju 19](https://d.docs.live.net/00a34c8f5e266a1e/Dokumenti/Prepoznavanje%20teksta%20računalnim%20vidom.docx#_Toc63549721)

[Slika 13 Pojednostavljeni prikaz treniranja neuronske mreže u programskom jeziku Python 20](https://d.docs.live.net/00a34c8f5e266a1e/Dokumenti/Prepoznavanje%20teksta%20računalnim%20vidom.docx#_Toc63549722)

[Slika 14 Pojednostavljeni primjer predviđanja u Python programskom jeziku 20](https://d.docs.live.net/00a34c8f5e266a1e/Dokumenti/Prepoznavanje%20teksta%20računalnim%20vidom.docx#_Toc63549723)

Popis prikaza programskog kôda

Prikaz programskog koda 1 - Primjer klase neuronske mreže 19

Prikaz programskog koda 2 – Primjer optimizacijske funkcije 19

Prikaz programskog koda 3 – Pojednostavljeni prolazak kroz mrežu 20

Prikaz programskog koda 4 - Računanje točnosti mreže 20

1 UVOD

U seminarskom radu obrađuje se računalno prepoznavanje teksta. Računalno ili strojno učenje je podijeljeno na više razina apstrakcije i svaka je razina do određene mjere obrađena.

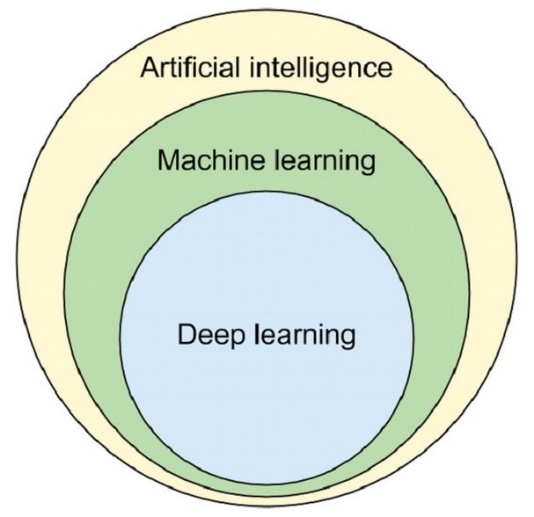
Uvod u područje strojnog učenja kao podvrste umjetne inteligencije započet je definiranjem pojmova korištenih kroz seminarski rad. Nakon navođenja osnovnih definicija, analizira se tržišna vrijednost strojnog učenja i budućeg potencijala u svijetu tako da se pokuša ukratko pojasniti razlog sve veće potražnje i interesa budućih inženjera unatoč velikoj količini.

Nakon toga strojno učenje predstavlja se kao rješenje na probleme iz prakse koje čovjek teoretski može riješiti, no oduzima previše vremena, i iz tog razloga pokušava poslove ranije nedostupne računalima učiniti efikasnijima. Nakon toga obrađuje se pojam neuronskih mreža sa svojim modelima kao način implementacije strojnog učenja. Objašnjen je način rada neuronskih mreža te su navedeni razni primjeri za koje se određeni model koristi. Prije prelaska na praktični dio seminarskog rada, obrađuju se primjeri iz svakodnevnog života u kojima se mogu uočiti aplikacije strojnog učenja.

Nakon analize stanja tržišta, definiranja pojmova i uvođenjem u neuronske mreže, rad prelazi na praktičniji dio, odnosno samu implementaciju jednostavne aplikacije za prepoznavanje teksta programskim jezikom Python koristeći teoriju navedenu kroz rad.

2 O UMJETNOJ INTELIGENCIJI

2.1 Osnovne definicije

S obzirom na to da je računarstvo područje koje se iznimno brzo mijenja, sami pojmovi umjetne inteligencije i strojnog učenja često se isprepliću i same definicije ovise o izvoru. Za svrhu seminarskog rada koristiti će se sljedeće općeprihvaćene definicije. Umjetna inteligencija (AI) je grana računarstva koja teži tome da dizajnira sisteme koji obrađuju informacije na sličan način na koji ljudi razmišljaju, a strojno učenje (ML) je jedno od područja istraživanja umjetne inteligencije koje pomoću specijaliziranih algoritamskih postupaka konstruiranih na temelju ljudskih misaonih, odnosno kognitivnih sposobnosti otkriva znanje. [1] [2] Nadalje, računalni vid je interdisciplinarno znanstveno područje koje se bavi načinom na koji računala mogu dobiti višu razinu razumijevanja digitalnih slika ili videozapisa, a neuronske mreže u računarstvu su skupovi logičkih neurona koji uzimaju informacije, obrađuju i zatim prosljeđuju daljnjim neuronima, čime se tvori mreža. [3] [4]

Slika Odnos umjetne inteligencije i strojnog učenja

Može se reći da je umjetna inteligencija, ili točnije strojno učenje zapravo način implementiranja računalnog vida, a implementira se korištenjem neuronskih mreža kroz određeni programski jezik.

2.2 Strojno učenje na tržištu

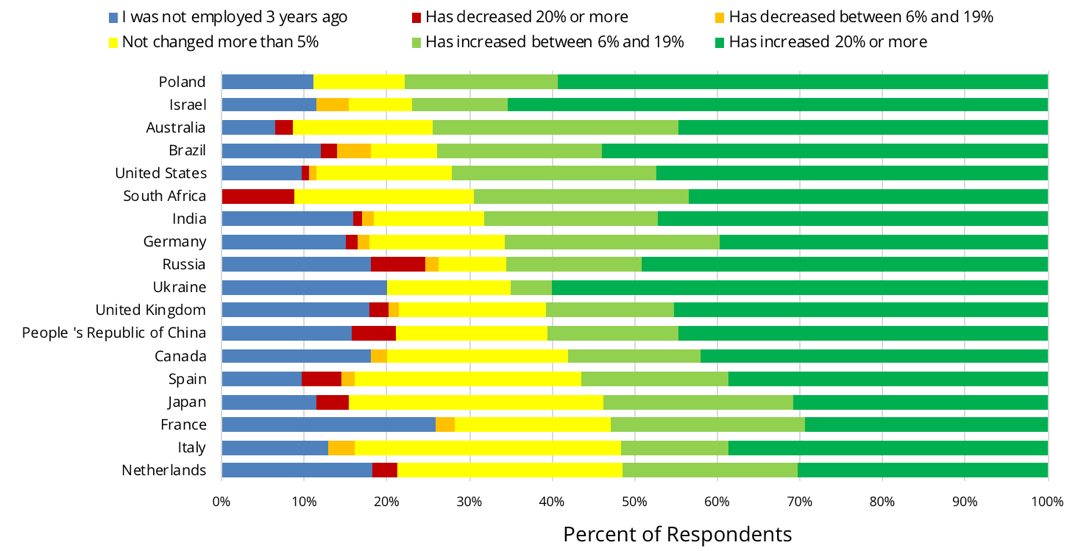
Strojno učenje još uvijek nije zaživjelo na tržištu poput mrežnih aplikacija, no i dalje je snažan alat koji se može koristiti u mnogim područjima. Prema podacima mrežne stranice *indeed*, prosječna plaća inženjera strojnog učenja u SADu je do sto pedeset tisuća dolara godišnje, što je preračunato sedamdeset i osam tisuća kuna mjesečno. [5]

|  |  |
| --- | --- |
| **Država** | **Godišnja plaća izražena u dolarima** |
| Sjedinjene Američke Države | 120.000 |
| Australija | 110.719 |
| Izrael | 87.500 |
| Kanada | 81.330 |
| Njemačka | 80.120 |
| Nizozemska | 75.337 |
| Japan | 70.132 |
| Ujedinjeno Kraljevstvo | 66.209 |
| Italija | 59.791 |
| Francuska | 55.008 |
| Južnoafrička Republika | 50.051 |
| Španjolska | 47.833 |
| Kina | 41.310 |
| Brazil | 35.349 |
| Poljska | 29.003 |

Tablica Srednje prosječne plaće inženjera strojnog učenja i podataka po državama prema podacima iz 2017. godine [6]

Kao što se može vidjeti na tablici jedan, vodeća tržišta strojnog učenja su razvijenije države poput Sjedinjenih Američkih Država, Kanade, Njemačke, Nizozemske i tako dalje. Razlog tome je taj što samo tržište strojnog učenja još uvijek nije rasprostranjeno poput nekih drugih tehnologija. Osim toga, inženjeri strojnog učenja skuplji su zbog relativne težine područja. Iz tih, ali i mnogih drugih razloga, strojno učenje sveprisutno je u većim tvrtkama poput Googlea i Facebooka, dok si manje tvrtke veću razinu strojnog učenja ne mogu priuštiti ili nije isplativa.

Osim trenutne tržišne vrijednosti, valja analizirati potencijal područja strojnog učenja za budućnost, kako u većim tako i u manjim državama. Grana strojnog učenja svojevrstan je spoj računarstva i matematike, što donekle sužava popis ljudi koji bi se njome mogli baviti. Stoga bi analiza potencijalnih inženjera mogla biti dobar poticaj na odluku bavljenja tim poslom.



Slika Promjene u plaći inženjera strojnog učenja i podataka po državama uspoređeno sa istima prije tri godine [6]

Osim podataka navedenih u drugoj tablici, zanimljiv podatak je da je najveći porast plaća, prikazan na drugoj slici, češći u srednje ili niže razvijenim državama negoli u razvijenijima. To je dokaz da unatoč dominiranju određenih država na tržištu svejedno postoji ogroman potencijal za inženjere strojnog učenja u budućnosti u većini država, što zbog primjenjivosti znanja inženjera unutar vlastite države, što zbog prirode cijele branše računarstva, odnosno mogućnosti rada na daljinu. To omogućava radnicima iz niže razvijenih država da rade za plaću koja bi bila prosječna, na primjer u Americi, a koja je iznimno dobra u njihovoj vlastitoj državi. Rad na daljinu mogao bi biti ostvaren kroz direktan rad za firme ili kroz sve popularniji način rada, a to je takozvani samostalni rad na daljinu (eng. *freelancing*).

2.3 Potreba za strojnim učenjem

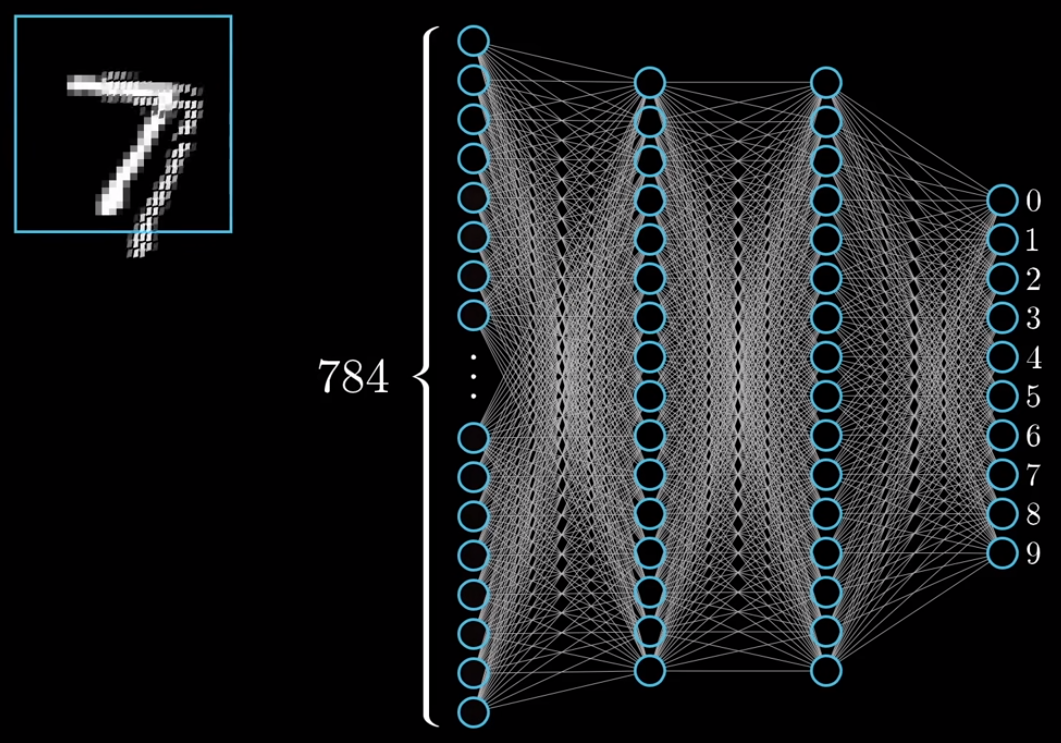
Ljudi obrađuju informacije iz kojih bi mogli izvući smisao. Na primjer, upravitelj tvrtke može kroz tablicu pročitati informacije o poslovanju firme u prethodnom periodu i zaključiti, ovisno o rezultatima, da firma dobro ili loše posluje. Vlada određene države može prema tablici nataliteta vidjeti kakvog učinka imaju određene mjere u povećanju broja stanovništva. No odakle dolaze takve tablice i informacije? U jednostavnim slučajevima mogu doći od osoba, no zbog ograničenosti ljudske sposobnosti, takvu vrstu informacije obično donosi računalo.

Svijet je prepun informacija. Te informacije mogu biti u formatu medija, na primjer slika, videozapisa i tako dalje, te mogu biti drugog oblika, poput skupa brojeva koji bez analize nemaju posebno značenje. Čitajući brojeve sa tablice bez naslova, opisa i oznaka ti brojevi mogu imati bilo koje značenje. Milijarde brojeva na jednom mjestu mogu biti stanje bankovnog računa svih osoba područja, brojevi životinja kroz određeni period i mnogo toga. Što je više takvih informacija, to je teže čovjeku uspješno dodijeliti značenje na jednostavan način pristupačan publici. Kao u mnogim situacijama, bilo bi idealno kada bi čovjek naporan i repetitivan posao mogao prebaciti na računalo. Upravo na to mjesto dolazi strojno učenje.

2.4 Implementacija strojnog učenja neuronskim mrežama

Strojno učenje može se promatrati kao skup alata i tehnologija koji se mogu koristiti da se pomoću određenih podataka dobiju odgovori na određena pitanja. Na primjer, podaci mogu biti milijuni brojeva, a pitanje može biti koliko efektivne su mjere poboljšanja nataliteta po kvartalima. Osim toga, važno svojstvo strojnog učenja i aplikacija koje ga implementiraju je to da one s vremenom postanu sve točnije. Pri stvaranju aplikacije potrebno je unesti mnogo podataka da računalo pokuša vidjeti inicijalni obrazac podataka i pokuša izmijeniti unutarnje brojke tako da na novim slikama uoči slične obrasce, no osim toga moguće je tokom samog rada aplikacije one vrijednosti koje se unesu na procjenu koristiti za dodatni ispravak unutarnjih brojki tako da se procjena budućih podataka poboljša naknadne procjene.

Da bi se izradila aplikacija strojnog učenja, mora se dizajnirati ili pronaći neuronska mreža koja će se naučiti prepoznavati određene obrasce. Kao što je ranije spomenuto, neuronska mreža je skup povezanih logičkih neurona ili čvorova koja obrađuje ulazne podatke i daje željeni izlaz. Na primjer, ulazni podaci mogu biti karakteristike nekretnine poput broja soba, prozora i kvadrature, a izlazni podatak može biti procjena cijene nekretnine. Postoji mnogo modela mreža, od kojih se dalje mogu dizajnirati različite mreže, a odabir uvelike ovisi o tome što inženjer želi da neuronska mreža nauči predviđati.

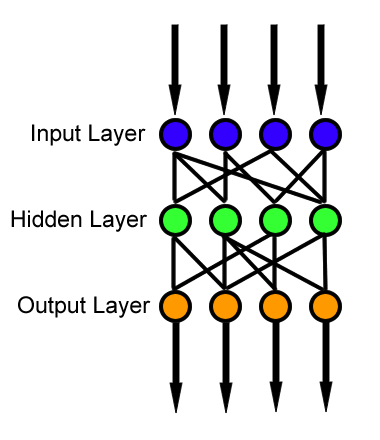


Slika Primjer prepoznavanja brojeva neuronskom mrežom

Neuronske mreže često se objašnjavaju na primjeru prepoznavanja brojeva. Koristeći navedene definicije neuronske mreže, može se uočiti da bi se mreža sa treće slike koristila za prepoznavanje brojeva, da su joj ulazni parametri crno-bijele slike sa različitim nijansama bijele i crne boje, da na ulazu ima broj neurona određen rezolucijom slike, a na izlazu određen i označen znamenkama.

Svi slojevi primaju vrijednost od ulaza ili drugih neurona, vrše neku funkciju i šalju drugim neuronima prema izlazu. Raniji slojevi mreže obično primaju temeljne informacije poput RGB vrijednosti piksela ili skupa piksela, dok kasniji slojevi rade sa apstraktnim informacijama poput linija, rubova, objekata i brojeva. Cilj tokom učenja neuronske mreže jest izmijeniti funkcije koje određeni neuron obavlja na način da na izlazu možemo pogledati vrijednosti neurona označenih znamenkama i vidjeti da je unesena slika broja sedam. Proces učenja uključuje sakupljanje velikog broja slika brojeva, raspodjelom na omjer 70:30 ili 80:20, što bi značilo da će se sedamdeset ili osamdeset posto slika koristiti za unos i izmjenu funkcija neurona, a ostale za provjeru točnosti mreže. Od svih ulaznih informacija unaprijed moramo znati koji broj je unesen tako da se može testirati točnost neuronske mreže.

2.5 Vrste neuronskih mreža

**Postoji mnogo vrsta neuronskih mreža koje služe različitim svrhama, od klasifikacije objekata, segmentacije, lociranja pa sve do procjena. [7] U mnogim modelima razlikuju se tri vrste slojeva neurona – jedan ulazni sloj, jedan ili više skrivenih slojeva te jedan izlazni sloj neurona. Neuronske se mreže ponašaju kao crna kutija te se skriveni ili središnji slojevi ne moraju se nužno analizirati nakon objave aplikacije. Njihove funkcije obrade mogu se programirano izmijeniti. Osim toga, podaci koji bi se izvukli iz njih ne bi imale veliko značenje za čovjeka jer korisnike zanima samo konačni rezultat na krajnjem sloju, a ne kako je računalo došlo do njega. Osim te zajedničke karakteristike svih neuronskih mreža, one se mogu podijeliti prema načinu obrade informacija. U sljedećem tekstu pri objašnjenjima koristiti će se engleski nazivi modela neuronskih mreža radi očuvanja semantičkog značenja.

Slika Primjer jednostavne *feedforward* neuronske mreže

*Feedforward* neuronska mreža najjednostavniji je model i onaj koji se najčešće uzima kao primjer neuronskih mreža. U takvoj mreži podaci putuju samo unaprijed, prema izlaznom sloju. Feedforward neuronske mreže često koriste mehanizam unazadne propagacije (eng. *back-propagation*) koji označava mijenjanje mreže na temelju izlaznih vrijednosti. Jedna vrsta te neuronske mreže je konvolucijska neuronska mreža, koja je također duboka neuronska mreža te je vrlo popularna i precizna u radu sa digitalnom grafikom pri detekciji, klasifikaciji i segmentaciji objekata.

*Radial basis function* (RBF) neuronske mreže uračunavaju udaljenost neurona od određene točke neuronske mreže, a može se koristiti u generiranju aproksimacije matematičke funkcije, predviđanju vremenskih funkcija, klasifikaciji te kontroli dinamičkih sistema. RBF neuronske mreže imaju tri sloja. Ulazni sloj ima po jedan neuron za svaki ulaz, srednji skriveni sloj ima unaprijed neodređen broj, a izlazni sloj neurona jedan je rezultat, a u slučaju klasifikacije je rezultata, odnosno neurona onoliko koliko je kategorija, a ti rezultati predstavljaju vjerojatnosti da ulaz spada pod određenu kategoriju.

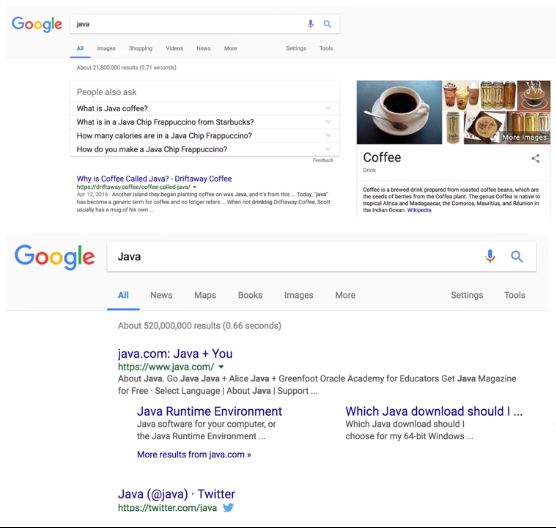
*Recurrent* neuronske mreže slične su prvoj navedenoj vrsti, no razlika je što u njima informacije mogu putovati u oba smjera, odnosno isti neuron i njegova memorija može se koristiti za obradu veće količine podataka. Zbog tog svojstva obično se primjenjuju u primjeni prepoznavanja rukopisa ili prepoznavanja govora.

Modularne neuronske mreže najsličnije su čovjekovoj neuronskoj mreži jer, poput čovjekove, implementirane su segmentacijom i modularizacijom težih zadataka u manje. [8] Na primjer, ljudski mozak vid paralelno raščlanjuje i obrađuje na boju i na kontrast u slojevima bočne genokulatne jezgre, a nakon toga se rezultati sažimaju u drugim dijelovima mozga. [9] Modularne neuronske mreže nisu popularne kao na primjer konvolucijske, no razvojem sklopovlja povećavati će se njihov potencijal u prostoru razumijevanja bioloških neuronskih mreža. Na primjer, moglo bi se razviti bolje razumijevanje toga kako primati raščlanjuju problem hvatanja objekata i zašto to rade vrlo efektivno u odnosu na druge životinje.

2.6 Primjeri iz prakse

Kao što je ranije spomenuto, strojno učenje češće je u velikim tvrtkama zbog cijene strojnih inženjera i težine implementacije, a usmjereno je na poboljšanju korisničkog iskustva i kvalitete usluge. Facebook već dugo vremena koristi tehnologiju strojnog učenja za detekciju lica, kojom se može omogućiti automatsko predlaganje osoba na fotografiji za označavanje. Osim toga, strojno učenje se koristi za predlaganje prijatelja na temelju informacija kao što su lokacija osobe, ostali prijatelji, interesi i sviđanja drugih stranica, filmova i tako dalje. [10]

Slika Primjer prepoznavanja lica strojnim učenjem

U Googleu strojno učenje koristi se za mnogo stvari, a najpopularnija implementacija je Google tražilica. Ona strojno učenje koristi za poboljšano rangiranje podataka tako da korisnik ranije pronađe željene podatke. Alat za to, osim samog upita upisanog u tražilicu, koristi i informacije koje može dobiti o korisniku iz njegove IP adrese, korisničkog računa, drugih pretraga korisnika njegovog područja u to doba i još mnogo toga. Na primjer, na šestoj slici može se uočiti da isti upit može dati potpuno različite rezultate. U prvoj polovici slike korisnik je pretragom izraza „java“ dobio vrstu kave, dok je na drugoj slici različit korisnik dobio za taj upit dobio programski jezik Java. Iz toga se može pretpostaviti da potonji često šalje upite vezane za računarstvo, dok prvi češće šalje upite vezano za pića.

Na ovom primjeru može se vidjeti koliko je strojno učenje korisno jer, u slučaju da je korisnik usluge znanstvenik koji traži relevantnu literaturu, mnogo vremena može se uštediti tako što će tražilica pokušati predvidjeti koje znanstvene radove ili članke on traži.

Slika Primjer strojnog učenja u Google tražilici

3 STVARANJE APLIKACIJE

3.1 Odabir tehnologija

3.1.1 Python programski jezik

Za stvaranje aplikacije strojnog učenja koristi se Python programski jezik koji je jedan od najpopularnijih kada je u pitanju umjetna inteligencija. Unutar jezika postoje mnogi paketi namijenjeni za strojno učenje. Ukratko će se navesti jedni od najpopularnijih paketa ili biblioteka - TensorFlow i PyTorch. Potonji je odabran za izradu seminarskog rada zbog jednostavnosti uporabe i intuitivne izvedbe.

3.1.2 Jupyter Bilježnice

Jupyter Bilježnice (eng. *Jupyter Notebooks*) su razvojna okruženja smještena na javnom ili lokalnom serveru koja omogućuju pisanje Python kôda, naredbi komandne linije (eng. *command-line*) te još mnogo toga. Imaju sposobnost prikaza formatiranog teksta, a često se koriste kao demonstracije funkcionalnosti ili pri obradi podataka, primjerice u strojnom učenju.

3.1.3 Colaboratory

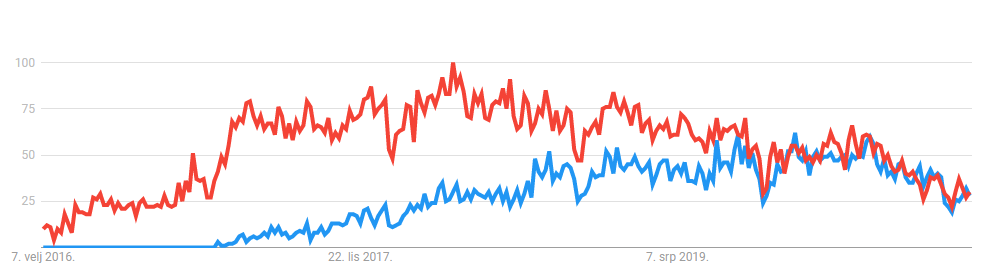
Colaboratory ili Google Colab proizvod je Googlea koji omogućuje udaljeno spajanje na Jupyter Bilježnice. Može se otvoriti u mrežnom pregledniku, a najčešće se koristi za demonstracije Python kôda te u svrhe strojnog učenja. Njegova najveća prednost je što, uz korištenje procesorske snage na Googleovim računalima, omogućuje i besplatan, no i vremenski ograničen pristup grafičkim karticama koje ubrzavaju treniranje modela neuronske mreže. Iz tog razloga odabran je za razvojno okruženje u kojemu je napisan praktični dio rada.

3.1.4 TensorFlow

TensorFlow je paket otvorenog kôda namijenjen za numeričke izračune visokih performansi korištenjem grafova protoka podataka (eng. *Data flow graph*). U pozadini TensorFlow je radni okvir koji izvodi računske operacije pomoću tenzora. Prednosti uključuju mnoge algoritme unutar paketa, ogromnu zajednicu, podršku rada kroz više grafičkih kartica i procesora. Nedostaci su vrlo niža brzina izvođenja u odnosu na druge radne te relativna težina učenja.

3.1.5 PyTorch

PyTorch je popularna biblioteka strojnog učenja za programski jezik Python, baziran na biblioteci Torch implementiranoj u programskim jezicima C i Lua. Izradio ga je Facebook, a koriste ga mnoge velike korporacije poput Twittera. Prednosti uključuju mnoge sposobnosti dubokog učenja i algoritama, korištenje grafičke kartice za ubrzanje računanja te mnogo istreniranih modela. Nedostaci su vezani za relativno mladu dob, a to su manjak resursa na internetu u odnosu na druge biblioteke.

Sedma slika prikazuje trendove pretrage. Može se uočiti da je TensorFlow dugo vremena bio popularniji na tržištu, no da je u zadnje vrijeme otprilike jednako popularan na Google tražilici kao što je to PyTorch.

Slika Usporedba Google trendova za pojmove TensorFlow (crveno) te PyTorch (plavo)

3.2 Teorija i funkcionalnosti potrebni za izradu

Ranije u radu ukratko je navedena relevantna teorija za seminarski rad. U sljedećim odlomcima detaljnije će se obraditi dijelovi koji su vezani za izradu aplikacije strojnog učenja. Obraditi će se podatkovna struktura tenzora, rad neuronske mreže kroz uobičajenu i unazadnu propagaciju te kreiranje klase koja predstavlja neuronsku mrežu. Rad koristi *feedforward* tip mreže.

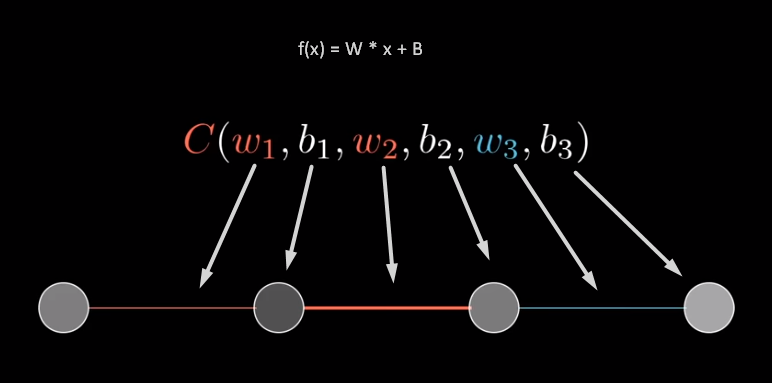
3.2.1 Tenzori

Tenzor je izraz posuđen iz matematike, a u strojnom učenju označava strukturu sličnu matricama više dimenzija. Tenzori u PyTorchu se sastoje od jednog tipa podatka, najčešće od realnog broja širine 16, 32 ili 64 bitova, a može i biti kompleksan broj širine do 128 bitova, *boolean* vrijednost ili cijeli broj širine do 64 bita.

3.2.2 Uobičajena propagacija neuronske mreže

Neuronske mreže ranije su opisane kao mreže u kojima se informacije kreću prema izlazu koji se može promatrati kao sloj jednog ili više neurona na desnom rubu. Neuroni su povezani standardnim linearnim funkcijama oblika .

Parametar „“ linearne funkcije naziva se težina (eng. *weight*), parametar „“ označava sklonost (eng. *bias*). Mreža bez ikakvih promjena ima konstantne težine i sklonosti, a mijenjaju se ulazne vrijednosti, odnosno parametar „“, koji može označavati RGB vrijednost piksela i slično.

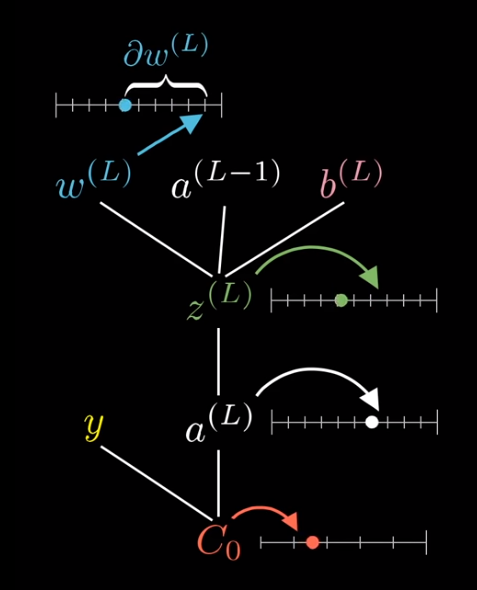


Slika Primjer uobičajene propagacije mreže

Na osmoj slici može se vidjeti prikaz jednostavne neuronske mreže sa jednom vezom između dva neurona. U toj mreži, težine predstavljaju linije između neurona, a sklonosti su konstante na čvorovima. Funkcija je jedan prolazak iz jednog u drugi čvor. Prema toj analogiji, prelazak iz ulaznog čvora, preko dva skrivena, pa sve do izlaznog, jest uzimanje ulaznog parametra „“ i slijedni prolazak kroz tri funkcije , od kojih svaka ima svoju težinu (, , ) i svoju sklonost (, , ).

3.2.3 Unazadna propagacija neuronske mreže

Uobičajena propagacija koristi se za predviđanje kada znamo koje težine i sklonosti treba postaviti na čvorove mreže koju smo dizajnirali. Na primjer, za mrežu sa dva čvora u kojoj je mreža istrenirana da izlaz bude vrijednost 2.5, iz formule možemo uočiti da je u toj mreži W postavljen na 0, a B na 2.5, što će za svaki „“ rezultirati u željenom izlazu .

Međutim, kada su u pitanju veće mreže, i kada želimo predvidjeti kompleksnije stvari, na primjer nalazi li se na slici mačka ili pas, tada se koristi metoda unazadne propagacije. Slično kao što se u prethodnom odlomku mreža istrenirala za izlaz 2.5 postavljajući težinu i sklonost, tako se unazadna propagacija koristi da se pokuša predvidjeti vrijednost težina i sklonosti u mnogo većim mrežama, i time dobio željeni izlaz.

Prema devetoj slici prikazati će se unazadna propagacija između dva neurona.

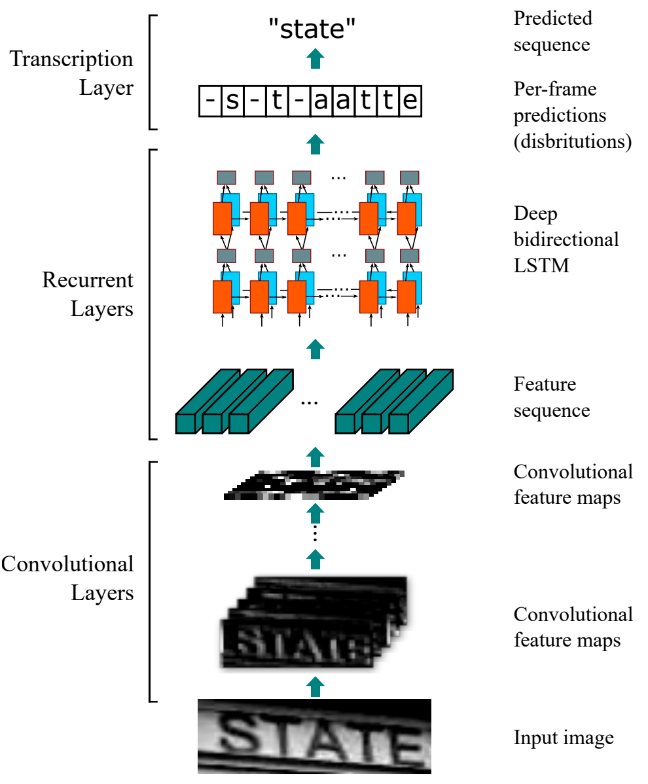
Slika Ovisnost varijabli neuronske mreže

Na slici je ulaz u prvi neuron, ili „“ označen sa . Težina prema drugom neuronu je označena sa , a sklonost sa . Slijedi da je ili rezultat desnog neurona sljedeće: . Dakle, izlazna vrijednost ovisi o ulaznoj. Da bismo dobili ovisnost izlazne vrijednosti o ulaznoj, moramo derivirati funkciju , čime se dobije gradijent tog čvora. Tada, ukoliko je gradijent malen, male promjene varijable „“ rezultirati će velikim izlaznim promjenama i tako dalje. Pomoću gradijenta dalje se mijenjanju konstante čvora i tako da se dobije željeni izlaz.

Prolaskom unazadne propagacije kroz cijelu mrežu paket PyTorch automatski sprema derivacije ili gradijente čvorova u njih same, te se na temelju toga može dobiti preciznije izlazna vrijednost.

Unazadna propagacija koristi se u kontekstu treniranja modela, a ne predviđanja. Naime, to je slučaj kada se zna koji „“ ulazi u funkciju i kada se mogu mijenjati neke karakteristike te funkcije, odnosno težina i sklonost. Nakon što je model istreniran, parametar „x“ smatra se nepoznatim jer kada je aplikacija objavljena, na primjer aplikacija za prepoznavanje tumora sa rendgenske slike, isprva se ne zna ima li pacijent tumor.

3.2.4 Stvaranje klase za strojno učenje

Za stvaranje klase korišten je model sa desete slike izvučen iz znanstvenog rada vezanog na temu prepoznavanja teksta. [11] Model kroz nekoliko koraka uzima sliku, primjenjuje na njoj mnoge konvolucije koje će naglasiti relevantne dijelove, a zatim se ti dijelovi dalje rastavljaju i u konačnici dobijemo predviđanje teksta.

Da bi se kreirala klasa, moramo u Python kôdu uvesti sljedeće pakete: torch, torchvision, torchvision.transforms, torch.nn i torch.nn.functional.

Slika Model za prepoznavanje teksta

3.2.5 Implementiranje klase za strojno učenje u Pythonu

class Net(nn.Module):

    def \_\_init\_\_(self):

        super(Net, self).\_\_init\_\_()

        self.konvolucija = nn.Conv2d(...)

        self.pool = nn.MaxPool2d(2, 2)

        self.fc1 = nn.Linear(16 \* 5 \* 5, 120)

    def forward(self, x):

        # Obrada kroz mrežu

        x = self.pool(F.relu(self.konvolucija(x)))

        x = torch.nn.functional.relu(self.fc1(x))

        return x

Prikaz programskog koda – Primjer klase neuronske mreže

Prikaz pojednostavljene klase na jedanaestoj slici pokazuje radnje potrebne za kreiranje klase. Pri kreiranju klase mora se naslijediti nn.Module koji sadrži mnoge dijelove ranije spomenute u radu koji se automatski odrađuju. Na primjer, ta klasa zadužena je za rad unazadne propagacije i podešavanje težina i sklonosti čvorova neuronske mreže.

U konstruktoru klase definiraju se radnje i konvolucije prikazane u modelu desete slike i mogu se postaviti određeni parametri koji će se koristiti u neuronskoj mreži.

Nadalje, mora se dizajnirati „forward“ funkcija koja će predstavljati jedan prolazak jedne ulazne vrijednosti kroz mrežu. Ta funkcija poziva se u pozadini kada se objektu klase pošalje objekt, u ovom slučaju slika koja sadrži tekst.

3.3 Rad sa kreiranom neuronskom mrežom

import torch.optim as optim

criterion = nn.CrossEntropyLoss()

optimizer = optim.SGD(net.parameters(), lr=0.001, momentum=0.9)

Prikaz programskog koda – Primjer optimizacijske funkcije

Da bi se dostigao učinak čitanja teksta iz grafike, potrebno je stvoriti objekt klase neuronske mreže i na njoj primijeniti sljedeće radnje: definirati funkciju koja će mijenjati težine i sklonosti. Implementacija je prikazana na dvanaestoj slici.

Funkcija korištena za optimizaciju težina i sklonosti naziva se Stohastični Spust Gradijenata (eng. *Stochastic Gradient Descent*), ili SGD funkcija. Navedena funkcija težinu mijenja pomoću njene trenutne vrijednosti, gradijenta dobivenog u unazadnoj propagaciji i konstante poslane koja određuje brzinu promjene:

for ponavljanje in range(2):

    for index, podaci in enumerate(spremnik\_podataka):

        # Nuliranje prethodnih gradijenata

        optimizer.zero\_grad()

        # Prolazak kroz mrežu i optimizacija

        izlaz = mreza(podaci)

        optimizer.step()

Prikaz programskog koda – Pojednostavljeni prolazak kroz mrežu

Nakon definiranja funkcije koja vrši optimizaciju, slijedi treniranje. Iz isječka koda prikazanog na trinaestoj slici, može se uočiti da treniranje uključuje prolazak po podacima, u ovom slučaju slika sa tekstom i slanjem tih podataka u objekt ranije kreirane koja predstavlja neuronsku mrežu. Proces se odvija dva puta radi povećanja preciznosti neuronske mreže.

3.4 Rad i analiza rezultata

tocni = 0

sveukupno = 0

for podaci in svi\_podaci:

    izlaz = mreza(podaci)

    # Usporedba...

    #tocni += 1

    sveukupno += 1

    print("Preciznost: %2d %%" %(tocni / sveukupno \* 100))

Prikaz programskog koda 4 – Računanje točnosti mreže

Pri usporedbi trinaeste i četrnaeste slike može se uvidjeti sličnost. Kada je u pitanju predviđanje, koristeći ulazne slike koje sadrže tekst prolazimo kroz petlju šaljući slike kroz mrežu. Da bi se rezultati mogli analizirati, koriste se slike čiji se tekst unaprijed zna. Nadalje, u dijelu označenim komentarom obrađuje se provjera i povećava brojač točnih rezultata za točne procjene. Na kraju izvedbe pomoću sveukupnog broja slika za predviđanje i broja točnih predviđanja dobiva se točnost neuronske mreže u predviđanju. Koristeći model sa desete slike dolazi se do preciznosti od oko 95%. Suvremene neuronske mreže sa većim brojem ulaznih podataka za testiranje i najjačim računalima mogu, ovisno o vrsti slike koja sadrži tekst, dostići vrijednosti između 95 i 100% točnosti.

4 PRAKTIČNI ZADATAK

4.1 Plan rada

5 ZAKLJUČAK

Kroz seminarski rad predstavljeno je područje strojnog učenja i umjetne inteligencije na više razina apstrakcije. Osim teorije potrebne za rad strojnog učenja i prepoznavanja teksta iz slika, analizirano je tržište i potencijal strojnog učenja u suvremenom svijetu. Osim toga, prikazani su i primjeri u kojima se koristi strojno učenje. Nakon toga, prikazano je implementiranje aplikacije za prepoznavanje teksta iz slika kroz programski jezik Python. Kratkom analizom popularnih alata za izradu i detaljnijom obradom relevantne teorije obuhvaćeni su svi dijelovi kreiranja aplikacije, od rada sa tenzorima do treniranja podataka i predviđanja. Osim toga, navedeni su dobiveni rezultati dizajnirane neuronske mreže, ali i rezultati suvremenih neuronskih mreža.

Popis literature

|  |  |
| --- | --- |
| [1] | Anderson, Dede, Fontana, Panikkar, Taylor and Waugh, World Book Encyclopedia, Chicago: World Book Inc., 2019. |
| [2] | Ž. Panian, Informatički Enciklopedijski Rječnik (M-Z), Zagreb: Europapress holding d.o.o, 2005. |
| [3] | C. M. B. Dana H. Ballard, Computer Vision, Englewood Cliffs: Prentice-Hall Inc., 1982. |
| [4] | J. J. Hopfield, »Neural networks and physical systems with emergent collective computational abilities,« *Proceedings of the National Academy of Sciences of the United States of America,* pp. 2554-2558, Travanj 1982. |
| [5] | »indeed,« [Mrežno]. Available: https://www.indeed.com/career/machine-learning-engineer/salaries?from=career. |
| [6] | B. Hayer, »Business Broadway,« 8 Travanj 2018. [Mrežno]. Available: https://businessoverbroadway.com/2018/04/08/salaries-of-data-scientists-and-machine-learning-engineers-from-around-the-world/. |
| [7] | »EL-PRO-CUS,« [Mrežno]. Available: https://www.elprocus.com/artificial-neural-networks-ann-and-their-types/. |
| [8] | »Design and evolution of modular neural network architectures,« *Neural Networks, Svezak 7, Izdanje 6-7,* pp. 985-1004, 1994. |
| [9] | P. Tahmasebi i H. Ardeshir, »Application of a Modular Feedforward Neural Network for Grade Estimation,« *Natural Resources Research,* pp. 25-32, 21 Siječanj 2011. |
| [10] | F. Careers, »Facebook,« 15 Siječanj 2020. [Mrežno]. Available: https://www.facebook.com/careers/life/machine-learning-at-facebook. |
| [11] | B. Shi, X. Bai i C. Yao, »An End-to-End Trainable Neural Network for Image-based Sequence Recognition and Its Application to Scene Text Recognition,« 21 Srpanj 2015. [Mrežno]. Available: https://arxiv.org/pdf/1507.05717.pdf. |