

Umjetne neuronske mreže kao metoda umjetne inteligencije

Vulin, Mate

Undergraduate thesis / Završni rad

2020

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Zagreb, Faculty of Organization and Informatics / Sveučilište u Zagrebu, Fakultet organizacije i informatike**

Permanent link / Trajna poveznica: <https://urn.nsk.hr/urn:nbn:hr:211:803651>

Rights / Prava: [Attribution-NonCommercial 3.0 Unported / Imenovanje-Nekomercijalno 3.0](#)

Download date / Datum preuzimanja: **2025-03-13**



Repository / Repozitorij:

[Faculty of Organization and Informatics - Digital Repository](#)



**SVEUČILIŠTE U ZAGREBU
FAKULTET ORGANIZACIJE I INFORMATIKE
VARAŽDIN**

Mate Vulin

**UMJETNE NEURONSKE MREŽE KAO
METODA UMJETNE INTELIGENCIJE**

ZAVRŠNI RAD

Varaždin, 2020.

SVEUČILIŠTE U ZAGREBU
FAKULTET ORGANIZACIJE I INFORMATIKE
V A R A Ź D I N

Mate Vulin

Matični broj: 44944/16–R

Studij: Informacijski sustavi

**UMJETNE NEURONSKE MREŽE KAO METODA UMJETNE
INTELIGENCIJE**

ZAVRŠNI RAD

Mentor :

Dr. sc. Okreša Đurić Bogdan

Varaždin, rujan 2020.

Mate Vulin

Izjava o izvornosti

Izjavljujem da je moj završni rad izvorni rezultat mojeg rada te da se u izradi istoga nisam koristio drugim izvorima osim onima koji su u njemu navedeni. Za izradu rada su korištene etički prikladne i prihvatljive metode i tehnike rada.

Autor potvrdio prihvaćanjem odredbi u sustavu FOI-radovi

Sažetak

Rad obrađuje koncept umjetnih neuronskih mreža kao metode umjetne inteligencije te postupak njihovog treniranja. Uz kratki pregled metoda umjetne inteligencije, rad se usredotočuje na jednu metodu - metodu umjetne neuronske mreže. U radu je prikazana struktura umjetne neuronske mreže u kojim su prikazni njezini gradivni elementi te slojevi umjetne neuronske mreže i njihove funkcionalnosti. Prema raznim kriterijima su predstavljeni tipovi umjetnih neuronskih mreža. U radu su dani primjeri korištenja umjetne neuronske mreže u stvarnom svijetu. Praktični primjer je implementiran u jednom od popularnih programskih jezika za implementaciju ovog segmenta umjetne inteligencije, implementira primjer umjetne neuronske mreže i njeno treniranje, uz razna objašnjenja s popratnom dokumentacijom.

Ključne riječi: Ključne riječi: umjetna inteligencija, umjetne neuronske mreže, treniranje umjetne neuronske mreže, obrada podataka, umjetni neuron, struktura neuronske mreže, ulazni sloj, skriveni sloj, izlazni sloj, algoritam propagacije pogrešaka unatrag, strojno učenje .

Sadržaj

1. Uvod	1
2. Metode i tehnike rada	2
3. Umjetna inteligencija	3
3.1. Metode umjetne inteligencije	3
3.1.1. AIoT (<i>engl. Artificial intelligence of things</i>)	4
3.1.2. Inteligentno planiranje (<i>engl. AI planning</i>)	4
3.1.3. Višeagentni sustav (<i>engl. Multi agent system</i>)	4
3.1.4. Računalni vid (<i>engl. Computer vision</i>)	4
3.1.5. Strojno učenje (<i>engl. Machine learning</i>)	5
3.1.5.1. Nenadzirano učenje (<i>engl. Unsupervised learning</i>)	5
3.1.5.2. Nadzirano učenje (<i>engl. Supervised learning</i>)	5
3.1.5.3. Ojačano učenje (<i>engl. Reinforcement learning</i>)	6
3.1.5.4. Samostalno nadzirano učenje (<i>engl. Self-supervised learning</i>)	6
3.1.5.5. Slojevit prikaz povezanosti	6
3.2. Metode strojnog učenja	7
3.2.1. Linearna regresija	7
3.2.2. Logistička regresija	8
3.2.3. Stablo odluke (<i>engl. Decision trees</i>)	8
3.2.4. Šume nasumičnih odluka (<i>engl. Random Decision Forests or Bagging</i>)	9
3.2.5. Naive Bayes	9
3.2.6. Strojevi za podršku vektora (<i>engl. Support vector machines</i>)	10
4. Umjetne neuronske mreže	11
4.1. Definicija	12
4.2. Umjetni neuron	13
4.3. Struktura umjetne neuronske mreže	15
4.4. Treniranje i učenje umjetne neuronske mreže	16
4.4.1. Unaprijedna operacija (<i>engl. Forward operation</i>)	16
4.4.2. Algoritam propagacije pogrešaka unatrag (<i>engl. Backpropagation algorithm</i>)	16
4.5. Topologija umjetne neuronske mreže	17
4.5.1. Perceptron	18
4.5.2. Duboke unaprijedne umjetne neuronske mreže	18
4.5.3. Rekurzivne ili ponavljajuće umjetne neuronske mreže	19

4.5.4.	Umjetne neuronske mreže sa zatvorenim ponavljajućim neuronima	20
4.5.5.	Hopfieldove umjetne neuronske mreže	20
4.5.6.	Duboko vjerujuće umjetne neuronske mreže (<i>engl. Deep belief network</i>) .	21
4.5.7.	Duboko konvolucijske umjetne neuronske mreže (<i>engl. Deep convoulti- onal network</i>)	21
4.5.8.	Duboko konvolucijske grafički obrnute umjetne neuronske mreže (<i>engl. Deep convolutional inverse graphic network</i>)	22
5.	Primjena umjetne neuronske mreže	23
5.1.	Realne situacije	23
5.2.	Praktični primjer	23
5.2.1.	Programski kod	24
5.2.2.	Prikaz korištenja umjetne neuronske mreže	25
6.	Zaključak	27
	Popis literature	30
	Popis slika	32

1. Uvod

Rad obrađuje koncept umjetnih neuronskih mreža kao metode umjetne inteligencije te postupak njihovog treniranja. Motivacija prilikom odabira tema je bila u tome što me zanimalo područje umjetne inteligencije te sam zato odabrao jednu od metoda umjetne inteligencije pod nazivom umjetne neuronske mreže. U metodi umjetne neuronske mreže se razvija umjetna inteligencija računala. Razvoj umjetne inteligencije kod računala je značajan zato što smatram da računala mogu pomoći ljudima kod donošenja bitnih zaključaka u daljnjem razvijanju ljudskog znanja. U današnjem svijetu već postoje razni programi u kojem računalo nadilazi ljudske sposobnosti donošenja brzih zaključaka primjerice kod igranja šaha te popularne računalne igre Dota 2.

Nakon uvodnog dijela o motivaciji i značaju teme bit će dana struktura rada u samom uvodu. U početnom dijelu rada nalaze se metode i tehnike rada koje su korištene prilikom razrade teme također su dani i svi programski alati i aplikacije koje su korištene. U daljnjem dijelu rada razrađena je tema završnog rada. Prvo su objašnjeni ključni pojmovi umjetne inteligencije i kratki pregled metoda u kojima ona nastaje. Nakon kratkog pregleda metoda rad će se sasvim usredotočiti na metodu neuronskih mreža. U metodi neuronskih mreža je prikazana struktura umjetne neuronske mreže u kojim su prikazni njezini gradivni elementi te slojevi umjetne neuronske mreže i njihove funkcionalnosti. Prema raznim kriterijima su predstavljeni tipovi umjetnih neuronskih mreža. U radu su također dani i primjeri korištenja umjetne neuronske mreže u stvarnom svijetu. Nakon teorijskog objašnjenja rada prikazan je praktični primjer koji je implementiran u jednom od popularnih programskih jezika za implementaciju ovog segmenta umjetne inteligencije, implementira primjer umjetne neuronske mreže i njeno treniranje, uz razna objašnjenja s popratnom dokumentacijom. U završnom dijelu rada će biti dan zaključak kao osvrt na samu temu.

2. Metode i tehnike rada

Istraživačke aktivnosti su provedene internetskim putem u kojima su pronađene mnoge knjige, članci i video materijali vezani uz temu umjetne neuronske mreže. Tema je napisana preko online LaTeX uređivača Overleaf.

Overleaf je jako poznat LaTeX tekstualni uređivač kojeg koriste mnogi studenti diljem svijeta. Nije potrebno neko početno znanje da bi se njime koristilo. Jednostavno se može naučiti koristiti s Overleafovim uređivačem.[1] Sliku loga Overleafa se može vidjeti odmah ispod.



Slika 1: Overleaf ([2])

Za izradu programske aplikacije korištena je programski jezik Python i web stranica pod nazivom google Colab.

Google Colaboratory ili skraćeno Colab je Googlova internetska stranica koja omogućuje pisanje i izvršavanje Python-a u pregledniku. Pruža prednosti lakog prenošenja koda koji ima pristup obradi preko grafičke kartice u kojem nisu potrebne nikakve konfiguracije.[3]

Tensorflow je open source platforma koja se koristi za strojno učenje. Tensorflow u sebi sadrži mnoštvo alata, biblioteka i resursa koji omogućuju nastanak snažnih aplikacija koje se zasnivaju na strojnom učenju. Google je jedna od kompanija koja koristi Tensorflow.[4]

Keras je API od TensorFlow-a 2.0 koje je lako dostupan te se bavi rješavanjem problema strojnog učenja modernim postupcima dubokog učenja.[5]

3. Umjetna inteligencija

Umjetna inteligencija je sposobnost računala da obavi zadatak na način koji je često povezan s inteligentnim bićima. U procesu rješavanja zadataka računalo poprima karakteristike razvijanja inteligentnog procesa razmišljanja na primjer donosi nove zaključke, zaključuje na prijašnjim iskustvima i generalizira neku vrstu znanja.[6]

Pojam umjetne inteligencije se koristi i za označavanja svakog neživog sustava koji pokazuje inteligenciju, a takav sustav nazivamo inteligentnim sustavom. Inteligentni sustav je svaki sustav koji se može prilagoditi, učiti na prijašnjim iskustvima, zaključivati, prikupljati i obrađivati znanje itd. [7]

Umjetna inteligencija se može kategorizirati na jaku i slabu umjetnu inteligenciju. Slaba umjetna inteligencija (*engl. weak or narrow AI*) je dizajnirana i trenirana da obavlja točno određene specifične zadatke. Jaka umjetna inteligencija (*engl.strong or general AI*) se smatra onom koja može replicirati kognitivne sposobnosti ljudskog mozga. Jaka umjetna inteligencija rješava nepoznate zadatke korištenjem neizravne logike odnosno iskorištavanjem znanja iz jedne domene u drugu. [8]

Nažalost aplikacije i programi (umjetni inteligentni sustavi) koji su do danas napravljeni se ne mogu mjeriti sa fleksibilnošću i kreativnošću ljudskog uma. Najčešće programi i aplikacije pripadaju slabom tipu umjetne inteligencije odnosno one su specificirane na točno određeno područje znanja na primjer predviđanje kad se isplati kupovati i prodavati dionice u određenom trenutku.

3.1. Metode umjetne inteligencije

Umjetna inteligencija je primjenjiva na mnoge načine, te zato postoje puno metoda ili tehnika u kojima se koristi umjetna inteligencija. U ovome radu su opisane odabrane poznate metode umjetne inteligencije, a to su: inteligentno planiranje, višeagentni sustavi, računalni vid, AIoT i strojno učenje.

3.1.1. AIoT (engl. *Artificial intelligence of things*)

AIoT je kratica koja dolazi iz engleskog pojma artificial intelligence of things. AIoT je metoda koja kombinira umjetnu inteligenciju s tehnologijama IoT-a (Internet of things) u kojima se poboljšava komunikacija između čovjeka i mnogih uređaja. Obije tehnologije dobivaju na vrijednosti njihovom međusobnom kombinacijom. Umjetna inteligencija transformira podatke koji se generiraju od strane IoT-a koji postaju korisni kod donošenja odluka. Primjenu AIoT tehnologije se može vidjeti primjerice kod pametnih kuća. [9]

3.1.2. Inteligentno planiranje (engl. *AI planning*)

Inteligentno planiranje poput običnog planiranja pokušava naći niz akcija kojim će neki sustav doći do nekakvog cilja ili rješenja. U takvom planiranju se optimizira način kojim se dolazi do rješenja odnosno optimizira se rad samog sustava u svakom koraku sustava. Inteligentno planiranje također prikazuje objašnjenje zašto je pojedina akcija poduzeta odnosno prikazuje akcije koje narušavaju optimizaciju sustava. Primjena inteligentnog planiranja se može vidjeti kod aplikacija koji se najčešće koriste u industriji na primjer autonomni sustavi. [10]

3.1.3. Višeagentni sustav (engl. *Multi agent system*)

Višeagentni sustav je metoda umjetne inteligencije koja razvija umjetnu inteligenciju pomoću elemenata koji se nazivaju agenti. Agent je element višeagentnog sustava koji može samostalno donositi odluke ili zaključke. Naravno agenti razmatraju ili predviđaju odluke drugih agenata te tako dolaze do svoje optimalne odluke ili zaključka. [11]

3.1.4. Računalni vid (engl. *Computer vision*)

Računalni vid je metoda umjetne inteligencije koja pokušava replicirati funkcionalnosti sustava ljudskog vida. Replikacijom funkcionalnost vidnog sustava računalni vid omogućuje računalima da identificiraju odnosno procesiraju objekte preko slika i videa poput ljudi. Zbog razvoja dubokog učenja i umjetnih neuronskih mreža računalni vid čak i nadilazi sposobnosti ljudskog vida u nekim segmentima. [12]

3.1.5. Strojno učenje (*engl. Machine learning*)

Strojno učenje je najpopularnija i najučinkovitija metoda umjetne inteligencije koja treniranjem razvija umjetnu inteligenciju. Učenje strojnog učenja se nikako ne zasniva na eksplicitno isprogramiranim rješenjima(programima), nego se zasniva na samostalnom pronalasku rješenja kroz mnoge uzorke koji su povezani sa zadatkom ili problemom koje treba riješiti. Strojno učenje je usko povezano s naprednim metodama iz matematike i statistike. [13, str. 5-6]

Strojno učenje se može podijeliti na 4 glavna tipa ili grane učenja, a to su: nenadzirano učenje, nadzirano učenje, ojačano učenje i samostalno nadzirano učenje. Razlika u tipovima se vidi u samim postupcima i ciljevima učenja. U sljedećim ulomcima su objašnjeni tipovi strojnog učenja. [13, str. 94]

3.1.5.1. Nenadzirano učenje (*engl. Unsupervised learning*)

Ovaj tip strojnog učenja se sastoji od pronalaska zanimljivih transformacija unesenih podataka u kojem cilj učenja nije jasan. Cilj nenadziranog učenja je bolja vizualizacija i kompresija unesenih podataka odnosno poboljšanje shvaćanje odnosa između unesenih podataka. Za nenadzirano učenje se može reći da jako korisno u području analize podataka i da se često koristi kao priprema podataka koji se koriste prilikom nadziranog učenja. [13, str. 94]

3.1.5.2. Nadzirano učenje (*engl. Supervised learning*)

Ovaj tip strojnog učenja se sastoji od raznih koraka u kojem se uče mapirati uneseni podaci za određene poznate ciljeve ili rješenja određenog problema u kojima se daju ljudski primjerci rješenja određenog problema. Poznate primjene nadziranog učenja su optičko prepoznavanje simbola, prepoznavanje zvuka, klasifikacija slika i prevođenje teksta u druge jezike. [13, str. 94]

3.1.5.3. Ojačano učenje (engl. Reinforcement learning)

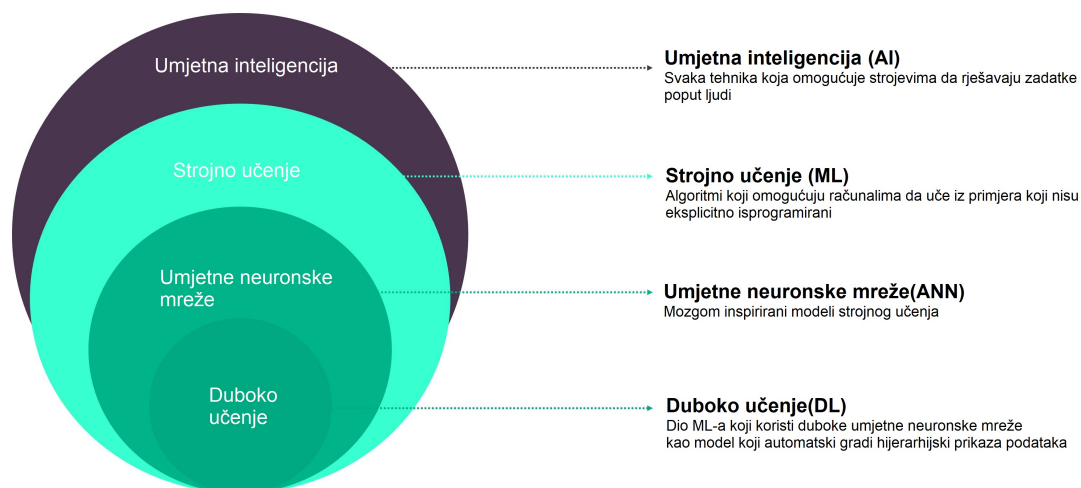
Ovaj tip strojnog učenja koristi agente za dohvaćanje podataka o okruženju u kojima se uči izbor „puteva“ kojim se dolazi do rješenja u kojem postoji sustav ocjenjivanja ili bodovanja. Ojačano učenje je nedavno dobilo veliku popularnost prilikom savladavanje popularne komplicirane tradicionalne kineske igre Go. Nažalost Ojačano učenje još nema svoju praktičnu primjeru te se smatra da će u budućnosti imati primjenu prilikom automatske vožnje auta, u robotici i edukaciji. [13, str. 95]

3.1.5.4. Samostalno nadzirano učenje (engl. Self-supervised learning)

Ovaj tip strojnog učenja se može smatrati tipom nadziranog učenja, ali ga mnogi autori vole izdvojiti u zaseban tip ili granu strojnog učenja. Razliku se uviđa u tome što više nije potreban ljudski faktor u učenju. [13, str. 94-95]

3.1.5.5. Slojevit prikaz povezanosti

Iz prethodno napisanog možemo naslutiti da umjetna inteligencija pokriva doista veliko znanstveno područje. Prema slici 3 koja se nalazi odmah ispod umjetnu inteligenciju možemo podijeliti na slojeve koji će dodatno pojasniti temu završnoga rada. Strojno učenje (Machine learning) je jedan od slojeva umjetne inteligencije koji proučava algoritme koji omogućuju učenje računala iz primjera te to učenje nije eksplicitno isprogramirano. Umjetne neuronske mreže su samo dio strojnog učenja koji se zasniva na principima učenja kao i kod samih ljudi.



Slika 2: Slojeviti prikaz AI ([14]), slobodan prijevod

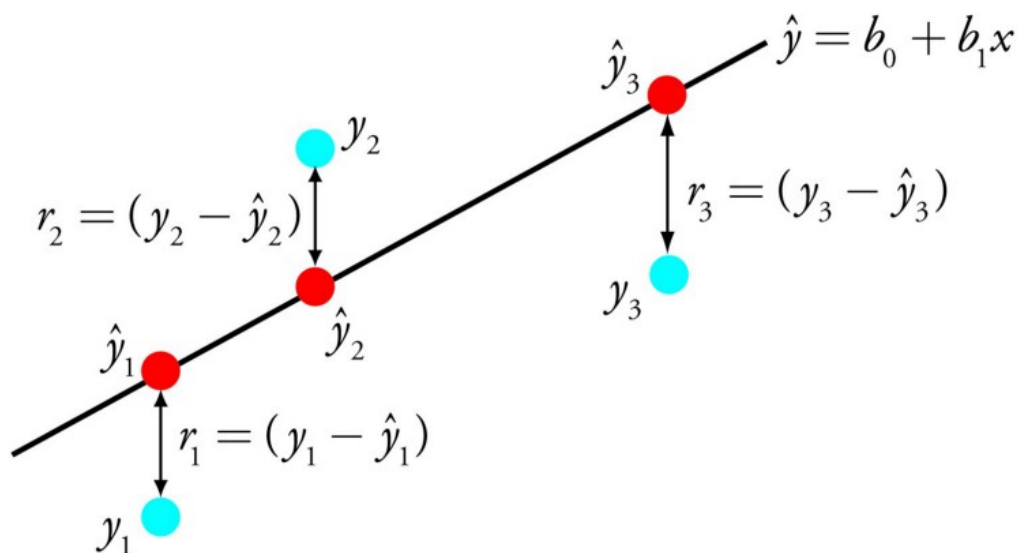
Umjetne neuronske mreže pripadaju u nenadzirano učenje, nadzirano učenje i usko su povezane s ojačanim učenjem. Primjer nenadziranog učenja su Kohonenove samo organizirajuće mape (engl. self-organizing map)[15, str. 147]. Primjer nadziranog učenja su unaprijedne umjetne neuronske mreže koje su dodatno objašnjene u sekciji Topologija umjetnih neuronskih mreža[15, str. 69, 84].Usko su povezane s ojačanim učenjem primjerice kod igranja računalnih igara kod kojih se iskorištava struktura umjetnih neuronskih mreža.[15, str. 207]

3.2. Metode strojnog učenja

Metode ili tehnike koje se koriste u razvoju umjetne inteligencije iz strojnog učenja najčešće potječu iz naprednih područja matematike i statistike. Matematičke i statističke metode su na domišljati način umetnute u programske aplikacije koje omogućuju sam trening umjetne inteligencije. U ovom radu kratko su opisane poznate odabrane metode, a to su: linearna regresija, logistička regresija, stablo odluke, šume nasumičnih odluka, naive Bayes, strojevi za podršku vektora te umjetne neuronske mreže koje su posebno obrađene kao glavna tema ovog rada. Opisane metode su dodatno obogaćene slikovitim sadržajem.

3.2.1. Linearna regresija

Linearna regresija je metoda koja je najpoznatija u području statistike. Smisao metode je pronalazak vrijednosti koeficijenta (B) koji pridonosi najjači utjecaj na preciznost funkcije f koju pokušavamo odrediti trenirajući umjetnu inteligenciju. Najjednostavniji primjer funkcije je pravac $y = B_0 + B_1 \times x$

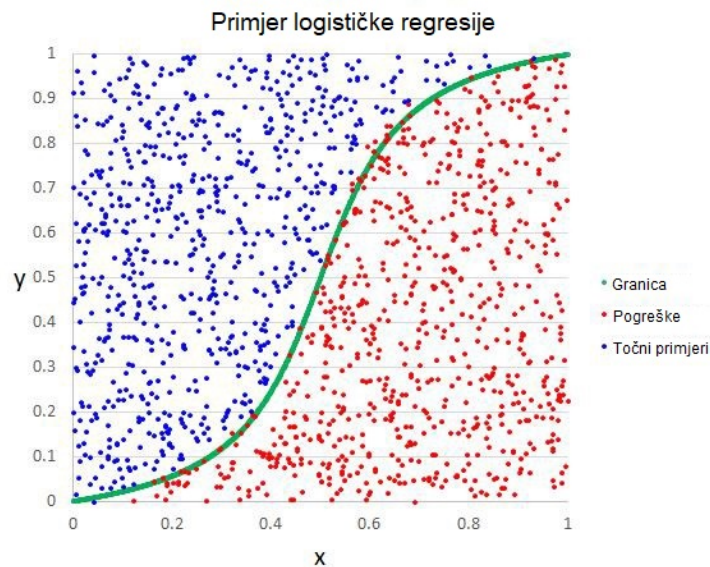


Slika 3: Linearna regresija ([16])

Mijenjanjem težinske vrijednosti koeficijenata dobivaju se različiti rezultati treniranja umjetne inteligencije. Metoda se smatra uspješnom kada se dobivaju što precizniji rezultati uklanjanjem sličnih unosnih vrijednosti koje daju slične rezultate. Metoda kao prije spomenuta se najviše koristi kod statičkih analiza primjerice financija. [16]

3.2.2. Logistička regresija

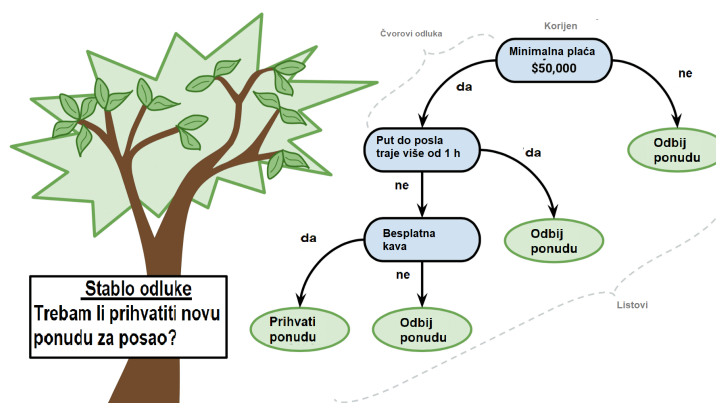
Logistička regresija je metoda koja daje kao rezultat binarno rješenje. To znači da može predvidjeti rezultat i specificirati vrijednost y u jednu od dvije klase. Funkcionira slično kao i linearna regresija mijenjanjem težinskih vrijednosti algoritma dobivaju se različiti rezultati treniranja umjetne inteligencije. Razlika je u tome što se kao rezultat dobiva nelinearna funkcija. Metoda linearna diskriminatorska analiza je metoda koja je jako slična logističkoj regresiji samo što može kao rezultat predvidjeti više od dvije klase. Logistička regresija je jako korisna kod klasifikacije podataka. [16]



Slika 4: Logistička regresija ([16]), slobodan prijevod

3.2.3. Stablo odluke (engl. Decision trees)

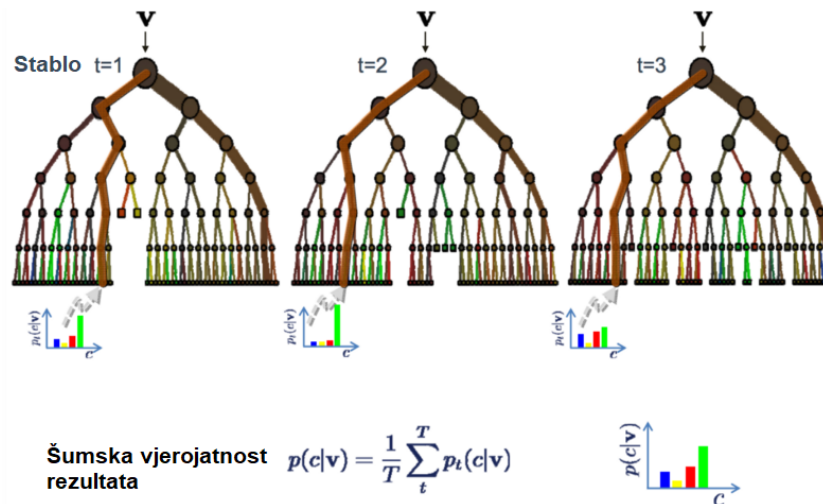
Metoda stabla odluke se bazira na klasičnom binarnom stablu. Rezultat metode se nalazi u nekim od listovima stabala odnosno djetetu koje nema potomka. Stablom odluke se mogu rješavati razni problemi kao na primjer odabir posla prema slici dole.[16]



Slika 5: Stablo odluke ([16]), slobodan prijevod

3.2.4. Šume nasumičnih odluka (*engl. Random Decision Forests or Bagging*)

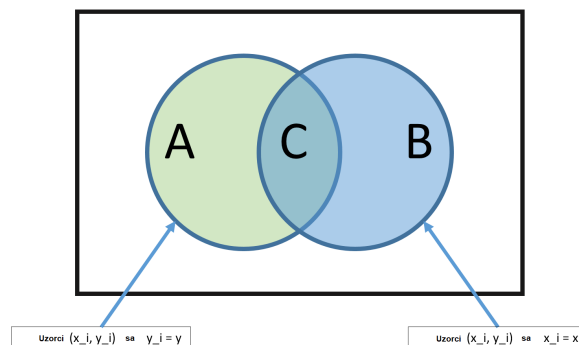
Šume nasumičnih odluka se formiraju pomoću stabala odluka. Kroz šumu se provode podaci te se rezultati sakupljaju kako bi se pronašla što točnija vrijednost rezultata. Šuma nasumičnih odluka ne traži rješenje kroz optimalnu rutu nego ona kroz više manje optimalnijih ruta dolazi do preciznog rješenja. Kada stablo odluke pronađe rješenje problema ostala stabla se prilagođavaju problemu kako bi se pronašao što bolji rezultat.[16]



Slika 6: Šuma nasumičnih odluka ([16]), slobodan prijevod

3.2.5. Naive Bayes

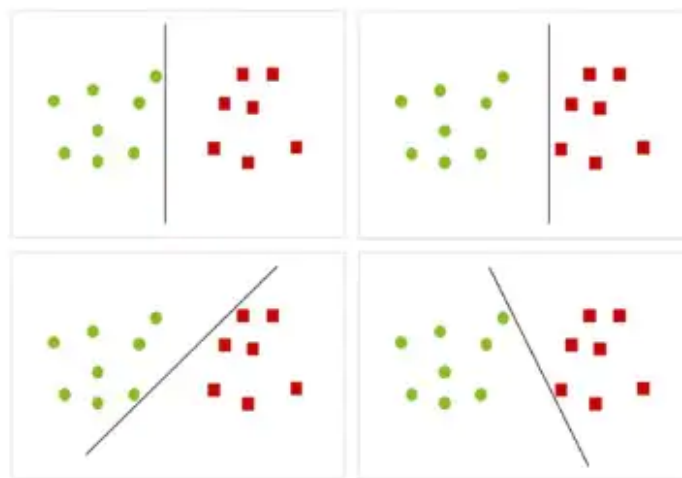
Naivna Bayesova (*engl. Naive Bayes*) metoda je jako jednostavan način rješavanja raznih kompleksnih rješenja. Metoda može izračunati dvije vrste vjerojatnosti. Može izračunati vjerojatnost pojavljivanja klase te može izračunati vrijednost pojavljivanja samostalne klase s obzirom na to da postoji dodatni x modifikator. Metoda se zove naivnom zbog toga što metoda pretpostavlja da su ulazni podaci nevezani jedni za druge. Ova metoda jedino funkcionira kod normaliziranih podataka te kod takvih podataka predviđa precizne podatke.[16]



Slika 7: Naive Bayes ([16]), slobodan prijevod

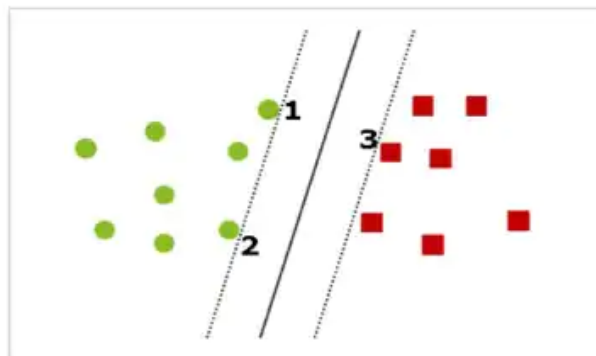
3.2.6. Strojevi za podršku vektora (*engl. Support vector machines*)

Strojevi za podršku vektora primjer je metode koji može riješiti problem s klasifikacijom podataka. Kod ovih vrsta problema cilj je utvrditi pripada li određeni podatak određenoj klasi ili ne pripada. Prvi korak metode zasniva se na treniranjem umjetne inteligencije na poznatoj klasi te se nakon toga može trenirati na novoj nepoznatoj klasi. Glavna ideja metode je da se pokušava pronaći granična linija koja razdvaja dvije klase na takav način da granična linija stvara maksimalan razmak između dvije klase. U primjeru prema slikama dolje postoje različito obojeni likovi koji mogu predstavljati dvije različite klase (npr crveni voće, a zeleni povrće). Svaka crta koja odvaja zelene krugove s lijeve strane i crvene kvadrate s desne strane smatra se valjanom graničnom linijom za problem s klasifikacijom. Kao što se može vidjeti na slici postoji beskonačan broj linija koje se mogu nacrtati između dvije klase.[17] Kao što je prije



Slika 8: Strojevi za podršku vektora 1 ([17])

spomenuto ova metoda pronalazi liniju koja maksimalizira razdvajanje dviju klasa. Na ovom primjeru postoje dvije isprekidane crte koje su međusobno paralelne te čine najveći razmak između klasa. Stvarna linija granice klasifikacije bit će linija koja se nalazi u sredini i međusobno je paralelna s druge dvije isprekidane crte.[17]



Slika 9: Strojevi za podršku vektora 2([17])

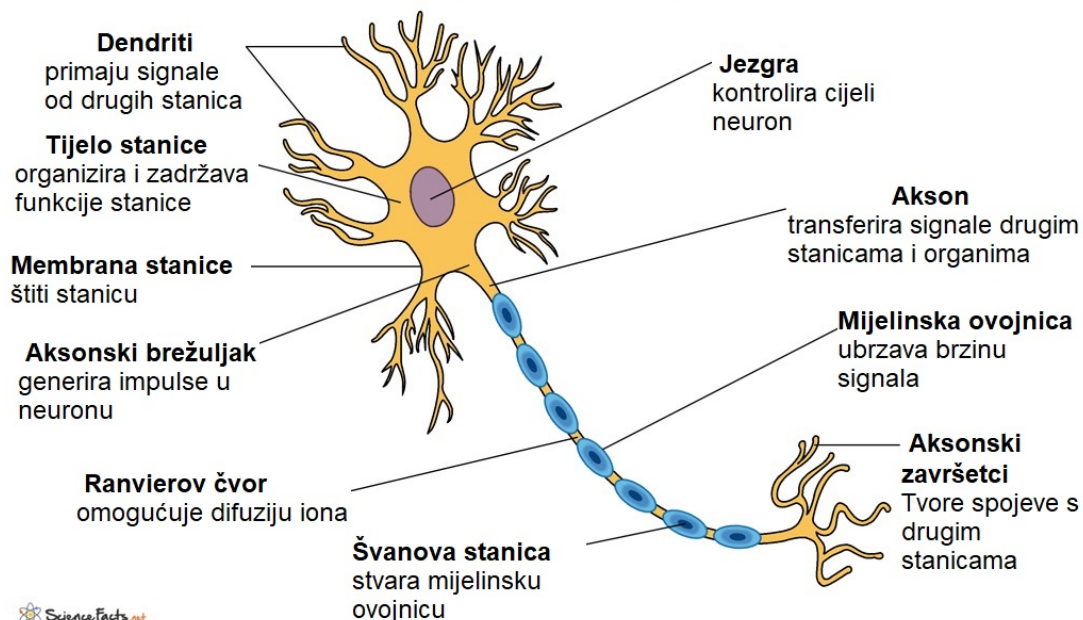
4. Umjetne neuronske mreže

Kao što je prije spomenuto umjetne neuronske mreže su metoda ili tehnika kojom se razvija umjetna inteligencija. Umjetne neuronske mreže pripadaju algoritmima strojnog učenja odnosno algoritmima koji uče iz primjera, te njihovo učenje nije eksplicitno. Radi lakšeg shvaćanja pojma umjetne neuronske mreže iz područja umjetne inteligencije u sljedećem ulomku rada je opisan neuron ili živčana stanica po kojoj je i sama metoda dobila ime.

Neuron ili živčana stanica čini osnovni gradivni element živčanog sustava. Živčani sustav sadrži oko 86 milijardi neurona koji međusobnim radom omogućuju funkcionalnosti živčanoga sustava. Grupa neurona čini živac. Neuroni su specijalizirane stanice kojim je uloga slanje i procesiranje informacija mozgu.[18]

Prema prethodno napisanom ulomku može se zaključiti da su neuroni izrazito vrijedne stanice koje se isplate replicirati u informatički svijet. Njihova funkcionalnost je zaista impresivna te se gradivni dijelovi neurona s njihovim funkcionalnostima mogu vidjeti na slici 10. Glavni dijelovi neurona su dendriti, tijelo i axon. Dendriti se ponašaju kao primači signala. Tijelo obrađuje signale te ih šalje preko odabranih aksioma do drugih neurona odnosno do drugih stanica. Funkcionalnosti primanja, obrade i slanja informacija su jako bitne u informatičkom svijetu te su se zato funkcionalnosti i struktura neurona kopirane u informatički svijet pod nazivom umjetne neuronske mreže.[18]

Djelovi neurona sa funkcijama



Slika 10: Neuron ([18]), slobodan prijevod

4.1. Definicija

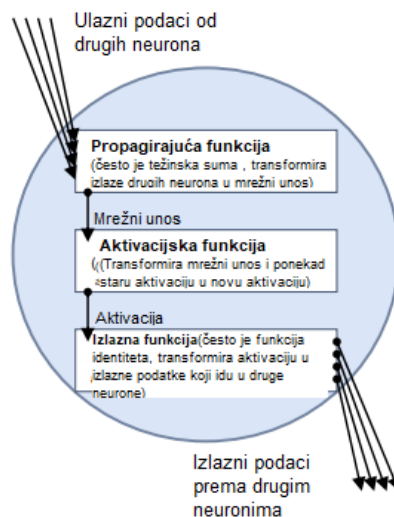
Neuronska mreža je skup algoritama koji nastoje prepoznati temeljne odnose u skupu podataka kroz proces koji oponaša način rada ljudskog mozga. Zbog toga se umjetne neuronske mreže odnose na organske ili umjetne sustave neurona. Umjetne neuronske mreže mogu se prilagoditi promjenama ulaznih podataka, te zbog toga umjetna neuronska mreža može generirati najbolji mogući rezultat bez potrebe za redizajnim izlaznih kriterija.[19]

Umjetne neuronske mreže mogu se smatrati pojednostavljenim modelima mreža od neurona koji se javlja u mozgu od živih bića. Gledajući s biološkog stajališta glavna zadaća neuronske mreže je obrada i procesiranje podataka.[20, str. 19]

Umjetna neuronska mreža je uređena trojka (N, V, w) koju čine 2 skupa N, V i funkcija w . N skup je skup neurona, a V skup $\{(i, j) | i, j \in N\}$ je skup veza između neurona i i neurona j . Funkcija w definira težinu (snagu) veze. $W(i, j)$ je težina veze između neurona i i neurona j koju skraćeno možemo zapisati kao $w_{i,j}$. Težina ili snaga veze se implementira u kvadratnu matricu W ili u težinski vektor W gdje redak predstavlja gdje započinje veza, a stupac završetak veze odnosno ciljani neuron.[15, str. 34]

4.2. Umjetni neuron

Umjetni neuron je zapravo sastavni ili gradivni element umjetne neuronske mreže. Replikacijom prave žive neuronske stanice se dobivaju njegove funkcionalnosti, a to su primanje, obrada i slanje informacija. U sljedećem ulomku objašnjene su funkcionalnosti neurona prema slici 11.



Slika 11: Umjetni neuron ([15, str. 35]), slobodan prijevod

Prema slici 11 se može vidjeti da su funkcionalnosti umjetnog neurona podijeljene u 3 funkcionalnosti, a to su:

- Propagirajuća funkcija (*engl. Propagation function*)
- Aktivacijska funkcija ili prijenosna funkcija (*engl. Activation function or transfer function*)
- Izlazna funkcija (*engl. Output function*)

Propagirajuća funkcija je funkcija koja za neuron j prima unose od o_{i_1}, \dots, o_{i_n} od povezanih neurona i_1, \dots, i_n te na te unose multiplicira težine w_{i_j} . Rezultat propagirajuće funkcije se naziva mrežnim unosom koji se može izračunati kao suma umnožaka svakog unosa povezanih neurona sa njihovim težinama w_{i_j} . Funkcija mrežnog unosa se može zapisati kao [15, str. 35]

$$net_j = \sum(o_j \times w_{i_j})$$

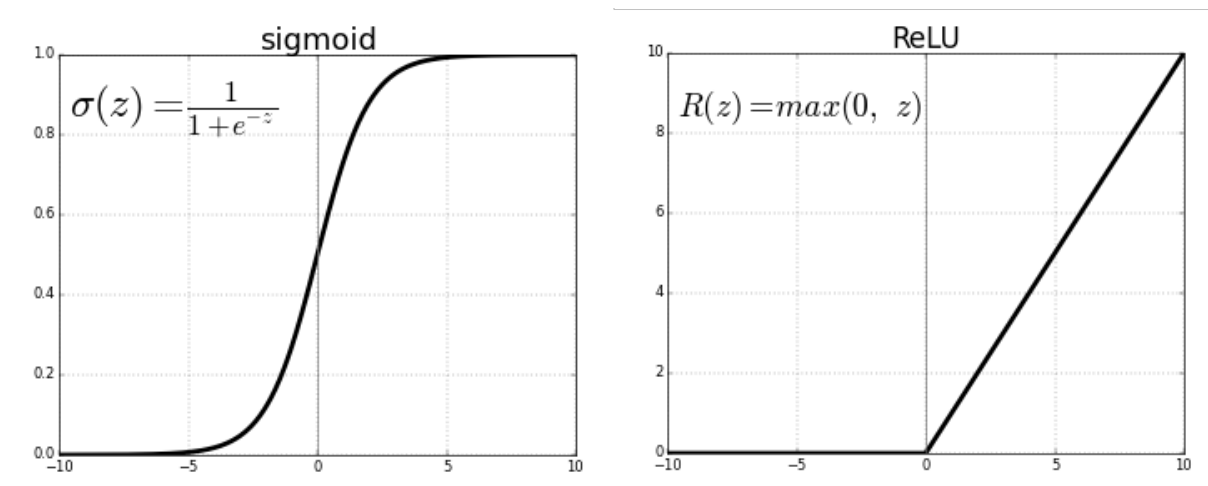
Aktivacijska funkcija ili prijenosna funkcija je funkcija koja omogućuje aktivaciju umjetnog neurona. Aktivni umjetni neuron poput biološkog omogućuje obavljanje određenih aktivnosti ili funkcionalnosti (prijenos vrijednosti) za razliku od ne aktivnog ili mirujućeg neurona. Biološki neuron postaje aktivan kada njegov električni naboj pređe određenu vrijednost. Zamisao vrijednosti koja nam govori o aktivnosti samog neurona je kopirana kod umjetnih neuronskih mreža pod nazivom granična vrijednost (*engl.threshold value*) koju označujemo s grčkim slovom theta. Kada se pređe određena granična vrijednost naravno umjetni neuron postaje aktivan u suprotnom je on mirujući ili ne aktivan. Vrijednost granične vrijednosti neurona se konstantno mijenja kroz procese učenja te svaki neuron može imati drukčiju graničnu vrijednost u umjetnoj neuronskoj mreži. Prema prethodno napisanom može se zaključiti da aktivacijska funkcija ovisi o graničnoj vrijednosti umjetnog neurona. Za aktivacijsku funkciju se može reći da ona transformira mrežni unos net_j (rezultat propagirajuće funkcije), te prijašnje aktivno stanje $a_j(t - 1)$ u novo aktivacijsko stanje $a_j(t)$. Ovisnosti aktivacijske funkcije za neuron j možemo zapisati kao [15, str. 36][20, str. 49]

$$a_j(t) = f_{act}(net_j(t), a_j(t - 1), \Theta_j)$$

Aktivacijske funkcije se mogu podijeliti na linearne i nelinearne aktivacijske funkcije. Linearne funkcije ne mogu poprimiti kompleksne parametre kojim se uobičajno opskrbljuju umjetne neuronske mreže. Nelinearne aktivacijske funkcije se mogu na lakši način prilagoditi kompleksnijim problemima te se zato i one upravo koriste prilikom aktivacije umjetnih neurona kod umjetnih neuronskih mreža. U sljedećem odlomcima su opisane dvije najpoznatije nelinearne aktivacijske funkcije, a to su sigmoidne funkcije i ReLu funkcije.[21]

Sigmoidna funkcija je funkcija kojoj krivulja poprima S oblik. Glavni razlog korištenja sigmoidne funkcije zato što samo poprima vrijednosti između 0 i 1 i zbog tog razloga je lako odrediti granične vrijednosti prilikom koje neuron postaje aktivan. Nažalost prilikom treniranja umjetne neuronske mreže često se javljaju zastoji koji usporavaju učenje umjetne neuronske mreže. ReLU (Rectified Linear Unit) funkcija je funkcija koja se danas najčešće koristi prilikom aktivacije umjetnih neurona. Prikaze funkcija s pripadajućim formulama se mogu vidjeti na slici 12 koja se nalazi odmah ispod.[21]

Izlazna funkcija je funkcija koja računa vrijednosti koje se prenose neuronima. Preciznije rečeno izlazna funkcija za neuron j računa vrijednost o_j iz njezinog aktivacijskog stanja a_j . Izlazna funkcija se može zapisati kao $f_{out}(a_j) = a_j$. Često se vrijednost aktivacijske funkcije a_j direktno prenose u povezani neuron. [15, str. 38]



Slika 12: Sigmoidna i ReLU funkcija([21])

4.3. Struktura umjetne neuronske mreže

Umjetna neuronska mreža se sastoji od umjetnih neurona ili jedinica koje su povezane vezama. Svaka veza ima svoju težinu ili snagu veze w dok svaki neuron ima propagirajuću i aktivacijsku funkciju kojom se dobiva rezultat samog rada neurona. [22, str. 728]

Različitim povezivanjem umjetnih neurona ili jedinica se dobivaju različiti dizajnovi strukture ili topologije umjetne neuronske mreže. Različite topologije umjetne neuronske mreže omogućuju rješavanje i prilagođavanje rješenja određenog problema. Naravno za jednostavne probleme topologija umjetne neuronske mreže izgleda jednostavnije i sastoji se od brojčano manje umjetnih neurona i veza između njih, dok za kompleksnije probleme topologija neuronske mreže izgleda kompleksnije. Standardizacijom određenih topologija umjetnih neuronskih mreža za koje se zna da daju izvrsna rješenja za određene probleme topologija umjetne neuronske mreže dobiva nazivlje. Slikovita nazivlja topologija umjetnih neuronskih mreža se mogu vidjeti na slici 13.

Kao što je i prije spomenuto umjetni neuroni se mogu povezivati na razne načine. Način povezivanja možemo podjeliti na :

- acikličke ili unaprijedne (*engl. forward*)
- prečice (*engl. shortcut connections*)
- rekurzivne ili ponavljajuće (*engl. recurrence connections*)
- lateralno ponavljajuće (*engl. lateral recurrence*)

Prema načinu povezivanja određene umjetne neuronske mreže dobivaju nazivlje kao naprimjer acikličke ili unaprijedne umjetne neuronske mreže (Feedforward network), unaprijedne umjetne neuronske mreže s prečicama (Feedforward network with shortcut connections), mreže s povratnom ili rekurzivnom vezom (recurrent networks), mreže s lateralnim ponavljanjima (laterally recurrent network). [15, str. 39-42]

4.4. Treniranje i učenje umjetne neuronske mreže

Treniranje je postupak u kojem umjetna neuronska mreža uči rješavati određeni problem. U sljedećim ulomcima je opisan način funkcioniranja unaprijedne umjetne neuronske mreže koja je jedna od najjednostavnijih tipova umjetnih neuronskih mreža (slika 15).

4.4.1. Unaprijedna operacija (*engl. Forward operation*)

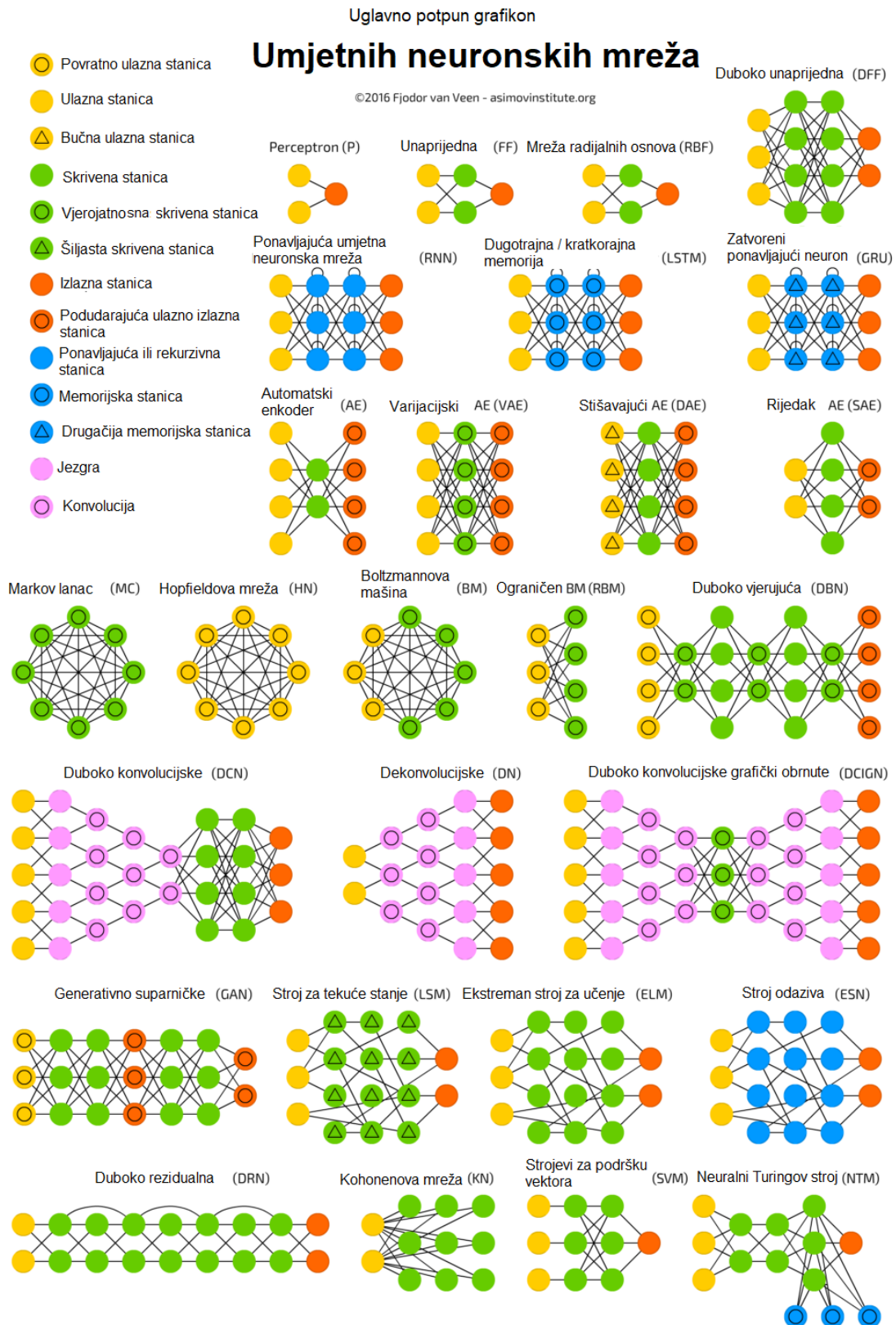
Unaprijedna operacija je postupak u kojem umjetna neuronska mreža prima ulazne podatke preko ulaznog sloja (ne sadrži neurone) te ih širi preko slojeva umjetne neuronske mreže unaprijed sve do izlaznog sloja u kojem neuroni daju rješenja ili izlaze. Važno je napomenuti da širenje informacija o kojem ovisi samo rješenje ovisi o aktivaciji neurona unutar umjetne neuronske mreže. [23][24, str. 61-62]

4.4.2. Algoritam propagacije pogrešaka unatrag (*engl. Backpropagation algorithm*)

Algoritam propagacije pogrešaka unatrag je jedan od najčešće upotrebljivanih algoritama koji je zaslužan za učenje odnosno treniranje umjetne neuronske mreže. Algoritam se temelji na naprednoj matematičkoj metodi gradijentnog spusta. Ovaj algoritam kalkulira težinske vrijednosti veza neuronske mreže na način u kojem se smanjuju gubitci ili pogreške pri pronalasku rješenja. Algoritam djeluje prvo na izlazni sloj te se širi prema uzlaznom sloju. Gubitci ili greške koje se stvaraju prilikom forward operacije su u principu odstupanja stvarnih rješenja od željenih ili dobrih rješenja. [24, str. 63][23]

Treniranje backpropagation algoritmom se može provesti preko dvije metode, a to su batch i non-batch metoda. U non-batch metodi težinske vrijednosti veza se mijenja nakon svakog unesenog uzorka dok se u batch metodi težinske vrijednosti veza mijenjaju nakon svih unesenih uzoraka. [24, str. 64]

4.5. Topologija umjetne neuronske mreže

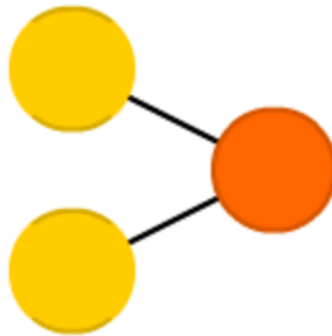


Slika 13: Topologija umjetne neuronske mreže ([25]), slobodan prijevod

Kao što je i prije spomenuto različitim povezivanjem umjetnih neurona ili jedinica se dobivaju različiti dizajnovi strukture ili topologije umjetne neuronske mreže. Na slici 13 pod nazivom topologija umjetnih neuronskih mreža nalaze se poznatiji primjeri topologija umjetnih neuronskih mreža od kojih su neki objašnjeni u sljedećim ulomcima.

4.5.1. Perceptron

Perceptron je najstariji i najjednostavniji oblik umjetne neuronske mreže koju je osmislio Frank Rosenblatt 1958 godine. Perceptron je jednoslojna neuronska mreža koja služi kao linearni klasifikator. Perceptron nažalost nije u mogućnosti riješiti složenije probleme. Slika perceptrona se nalazi odmah ispod.[26]



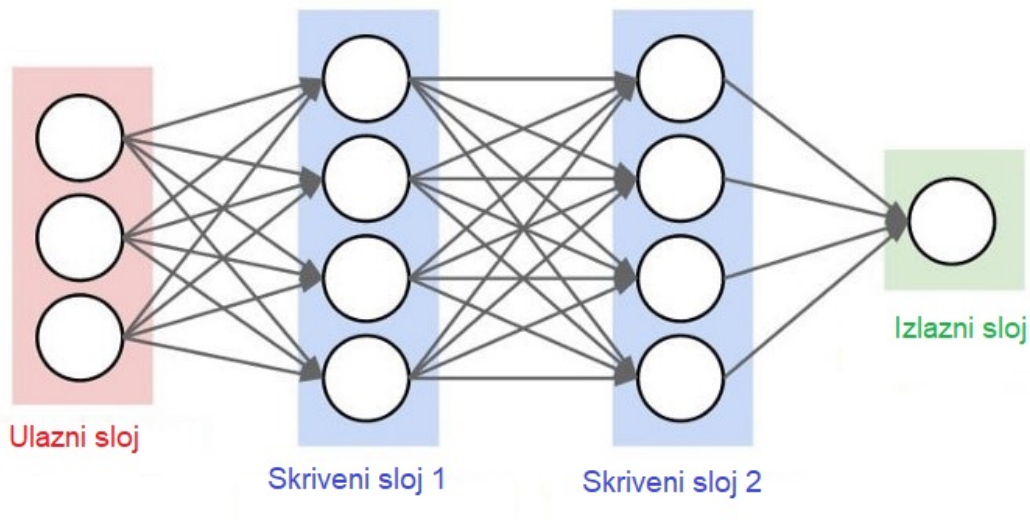
Slika 14: Perceptron ([25])

Ova topologija sadrži dva različita tipa stanica ili neurona. Žuto obojene stanice se nazivaju ulaznim stanicama (*engl. input cell*), one su zaslužne za primanje podataka. Narandžasto obojena stanica se naziva izlazna stanica (*engl. output cell*) koje daju izlaze odnosno rezultate.[27]

4.5.2. Duboke unaprijedne umjetne neuronske mreže

Duboke unaprijedne umjetne neuronske mreže (*engl. Deep feed forward neural network*) pripadaju unaprijednim umjetnim neuronskim mrežama (*engl. Feed forward neural networks*) koje imaju više od jednog skrivenog sloja. Postale su izrazito popularne u 21. stoljeću.[25]

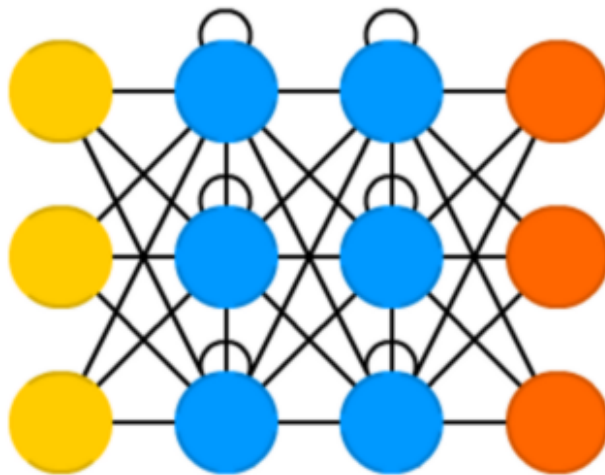
Unaprijedne umjetne neuronske mreže su slojevito građene te se sastoje od unosnog sloja (*engl. input layer*), skrivenog sloja (*engl. hidden layer*) i izlaznog sloja (*engl. output layer*). Nazivaju se unaprijedne zato što su sve veze usmjerene prema naprijed. Unaprijedne umjetne neuronske mreže su također potpuno povezane što znači da je u principu svaki neuron jednog sloja povezan sa svim neuronima idućeg sloja. Slikoviti prikaz duboke unaprijedne umjetne neuronske mreže se može vidjeti na slici ispod.[24, str. 61]



Slika 15: Duboke unaprijedne umjetne neuronske mreže ([26]), slobodan prijevod

4.5.3. Rekurzivne ili ponavljajuće umjetne neuronske mreže

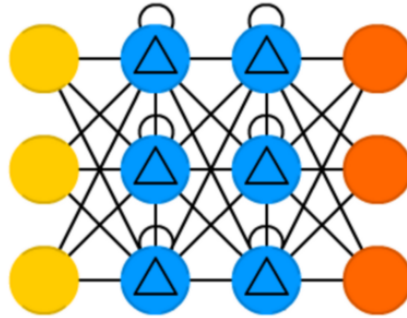
Rekurzivne ili ponavljajuće umjetne neuronske mreže (*engl. Recurrent neural network*) su umjetne neuronske mreže u kojima se pojavljuju neuroni ili stanice koje sami sebi šalju svoje rezultate. Takve stanice se nazivaju ponavljajućim ili rekurzivnim stanicama (*engl. recurrent cell*). Prvi takav tip neuronske mreže je poznat pod nazivom Jordanova mreža. Rekurzivno svojstvo neurona može biti jako korisno osobito u različitim analizama. [[25]]



Slika 16: Rekurzivne ili ponavljajuće umjetne neuronske mreže ([25])

4.5.4. Umjetne neuronske mreže sa zatvorenim ponavljajućim neuronima

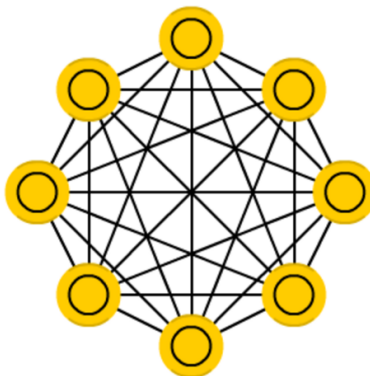
Umjetne neuronske mreže sa zatvorenim ponavljajućim neuronima (*engl. gated recurrent unit*) imaju poseban tip jedinica ili neurona koje imaju mogućnosti pamćenja koje svoju primjenu najčešće pronalaze u zvuku ili muzici. Sliku umjetne neuronske mreže sa zatvorenim ponavljajućim neuronima se može vidjeti na slici 17 koja se nalazi odmah ispod. [25]



Slika 17: Umjetne neuronske mreže sa zatvorenim ponavljajućim neuronima ([25])

4.5.5. Hopfieldove umjetne neuronske mreže

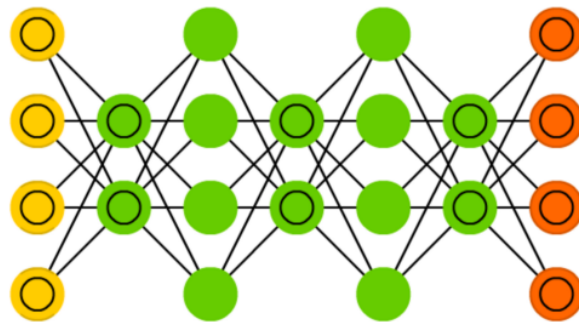
Hopfieldove umjetne neuronske mreže su posebna vrsta umjetnih neuronskih mreža koje se treniraju na ograničenim uzorcima i iz tog razloga na uzorak reagiraju s poznatim uzorak koji je isti. Njezina svaka stanica ili neuron služi kao unos prije treninga, kao skrivena stanica tijekom treninga i kao rezultatna stanica kada se treba koristiti. Takve stanice se nazivaju povratno ulaznim stanicama (*engl. back-fed input cell*) Slikoviti prikaz Hopfieldove umjetne neuronske mreže se može vidjeti na slici 18 koja se nalazi odmah ispod. [25]



Slika 18: Hopfieldove umjetne neuronske mreže ([25])

4.5.6. Duboko vjerujuće umjetne neuronske mreže (*engl. Deep belief network*)

Duboko vjerujuće umjetne neuronske mreže su nakupine Boltzmannovih mašina koje su jako slične Hopfieldovim umjetnim neuronskim mrežama. Povezivanje više takvih neuronskih mreža omogućuje treniranje jedne neuronske mreže s drugom te one mogu generirati podatke po već poznatim uzorcima. Slikoviti prikaz duboko vjerujuće mreže se može vidjeti na slici 19 koja se nalazi odmah ispod. [25]

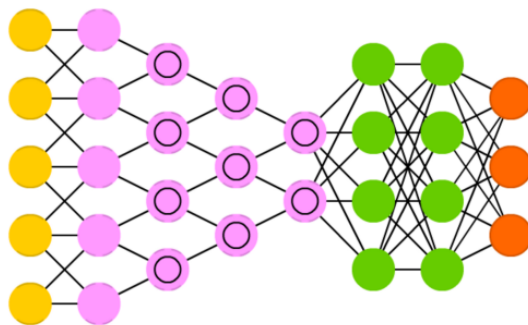


Slika 19: Duboko vjerujuće umjetne neuronske mreže ([25])

Ova topologija sadrži dvije nove vrste stanica. Zelene stanice s kružnicom se nazivaju vjerojatnosne skrivene stanice (*engl. probabilistic hidden cell*) koje primjenjuju radijalnu osnovnu funkciju koja računa vjerojatnost između testnog slučaja i vrijednosti stanice. Narandčasto obojene stanice s kružnicom se nazivaju podudarajuće ulazno izlazne stanice (*engl. match input output cell*), te se one najčešće koriste kod automatskih enkodera. Podudarajuće ulazno izlazne stanice pokušavaju približiti ili shvatiti povezanost između unesenih i izlaznih podataka. [28]

4.5.7. Duboko konvolucijske umjetne neuronske mreže (*engl. Deep convolutional network*)

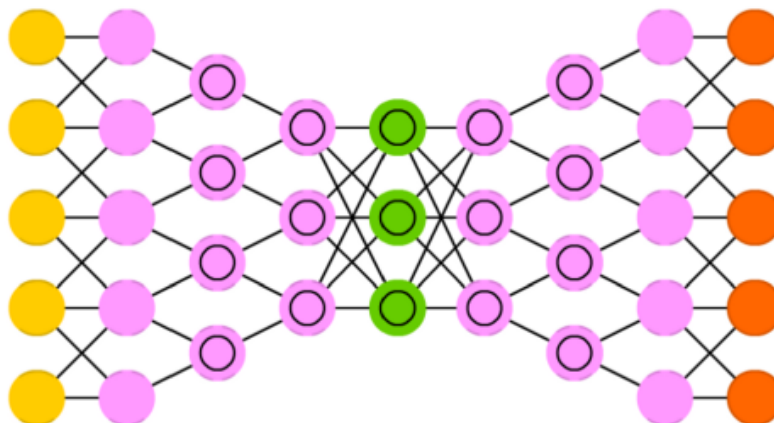
Duboko konvolucijske umjetne neuronske mreže su jako popularna vrsta umjetnih neuronskih mreža. Takve umjetne neuronske mreže sadrže posebne vrste neurona ili jedinica koje procesiraju ulazne podatke te se takve vrste neurona ili stanica nazivaju jezgrama (*engl. kernel*). Konvolucijski sloj se sastoji od konvolucijskih neurona (*engl. convolution*) koji pojednostavljaju podatke prijašnjeg sloja neurona. Duboko konvolucijske umjetne neuronske mreže se najčešće koriste za prepoznavanje slikovitih sadržaja. Slikoviti prikaz duboko konvolucijske umjetne neuronske mreže se može vidjeti na slici 20 koja se nalazi odmah ispod. [25]



Slika 20: Duboko konvolucijske umjetne neuronske mreže ([25])

4.5.8. Duboko konvolucijske grafički obrnute umjetne neuronske mreže (*engl. Deep convolutional inverse graphic network*)

Duboko konvolucijske grafički obrnute umjetne neuronske mreže se najviše koriste kod grafičkog procesiranja. Takve umjetne neuronske mreže omogućuju procesiranje slika za koje nisu posebno istrenirane. Primjerice mogu maknuti detalje sa slike, obojati sliku na drugačiji načini ili primjerice mogu zamijeniti objekte na slici kao naprimjer konja i zebrou. Slikoviti prikaz duboko konvolucijske grafički obrnute umjetne neuronske mreže se može vidjeti na slici 21 koja se nalazi odmah ispod. [25]



Slika 21: Duboko konvolucijske grafički obrnute umjetne neuronske mreže ([25])

5. Primjena umjetne neuronske mreže

Primjena umjetne neuronske mreže je prikazana u realnim situacijama, te je dodatno predstavljen i praktični primjer upotrebe umjetne neuronske mreže s pripadajućim programskim kodom.

5.1. Realne situacije

Skoro svaki čovjek koji se koristi modernim tehnologijama koristi umjetne neuronske mreže. U sljedećim primjerima su prikazani odabrani primjeri u kojima ljudi koriste umjetne neuronske mreže.

Umjetne neuronske mreže se često koriste prilikom upotrebe društvenih mreža. Facebook koristi umjetne neuronske mreže prilikom prepoznavanja lica. Kompanija Face.com je jedna od mnogih kompanija koja je pomogla Facebook-u u prepoznavanju ljudskih lica te je zaradila veliki novčani iznos radi svog programskog proizvoda. Od 2016. godine Facebook koristi DeepText koji koristi duboku umjetnu neuronsku mrežu koja na neki način omogućuje shvaćanje teksta poput ljudi. Instagram koristi duboku umjetnu neuronsku mrežu prilikom generiranja sugestija u kojima predlaže mijenjanje nekih riječi s emotikonima. Umjetne neuronske mreže se također koriste prilikom pretraživanja na internetu. Amazon nam primjerice daje sugestije na proizvode koje smo pretraživali u povijesti. Umjetne neuronske mreže se koriste na mobilnim uređajima primjerice s pritiskom na gumb i izricanjem glasovne komande "Ok Google". Opće poznati pametni asistenti Siri i Alexa također koriste umjetne neuronske mreže prilikom izricanja glasovnih komandi.[29]

5.2. Praktični primjer

Primjena neuronske mreže je napravljena na praktičnom primjeru u kojem umjetna neuronska mreža prepoznaje rukom napisane brojke. Primjer je napisan u google colab-u pomoću popularnog programskog jezika python. Primjer je obrađen i napravljen uz pomoć YouTube poučka([30]). Prikaz ručno napisanih brojki se može vidjeti na slici ispod. U primjeru su također korišteni TensorFlow i Keras.



Slika 22: Brojke ([31])

Primjer se zasniva na modelu unaprijednih neuronskih mreža koje se sastoje od ulaznog, skrivenog i izlaznog sloja. Ulazni sloj je "izravan" zbog efikasnijeg načina učenja neuronske mreže. Skriveni sloj se aktivira preko ReLU funkcije te sadrži 128 neurona. Izlazni sloj sadrži 10 neurona (zbog brojki 0-9) koji se aktiviraju sa softmax funkcijom. Prilikom treniranja umjetne neuronske mreže prate se 3 parametra, a to su parametri: optimizacije, gubitka i preciznosti. Kroz ulazne slojeve se šalju podaci u matricama koje su dohvaćene iz mnist baze koja sadrži rukom napisane brojke. Matrice od brojki se prvo normaliziraju u smislu da se sve brojke u matrici pretvaraju u vrijednosti od 0-1. Nakon što su pripremljeni ulazni podaci kreira se model umjetnih neuronskih mreža koji je prethodno opisan. Nakon kreacije i pripreme ulaznih podataka model umjetnih neuronskih mreža se trenira. Nakon treniranja modela umjetnih neuronskih mreža model se može spremiti. Kada je model spremljen model umjetnih neuronskih mreža može prikazati numeričku vrijednost brojke kojom predviđa vrijednost koja se nalazi na slici iz mnist baze. [30]

5.2.1. Programski kod

Kao što je i već prije spomenuto programski kod je napravljen prema YouTube poučku([30]) koji je dodatno zakomentiran s hrvatskim objašnjenjima.

```
import tensorflow as tf
mnist = tf.keras.datasets.mnist # 28x28 pixelne slike od brojki koje su rukom
    nacrtane
(x_train, y_train), (x_test, y_test)=mnist.load_data()#brojke u matrice

x_train=tf.keras.utils.normalize(x_train,axis=1)#normalizira brojke u matrici od 0-1
x_test=tf.keras.utils.normalize(x_test,axis=1)#žpomae pri čuenju

model = tf.keras.models.Sequential()#unaprijedni model umjetnih neuronskih mreža
model.add(tf.keras.layers.Flatten())#ulazni sloj (input layer)
model.add(tf.keras.layers.Dense(128, activation=tf.nn.relu))#skriveni sloj sa 128
    neurona i relu aktivacijom
model.add(tf.keras.layers.Dense(10, activation=tf.nn.softmax))#izlazni sloj sa 10
    izlaza zbog 10 brojki softmax je bolja funkcija

model.compile(optimizer='adam',
              loss='sparse_categorical_crossentropy',
              metrics=['accuracy']
              )#parametri za treniranje žnajvaniji je loss zato što se po njemu
              optimizira, optimizer treba znati dobro matematiku
model.fit(x_train, y_train, epochs=3)#treniranje NN epoch =iteracija po ulaznom
    parametru
val_loss, val_acc = model.evaluate(x_test, y_test)#procjeni gubitke i preciznost
print(val_loss, val_acc)
model.save('epic_num_reader.model')#sprema model

new_model=tf.keras.models.load_model('epic_num_reader.model')

predictions= new_model([x_test])
print(predictions)
```

```
import numpy as np
print(np.argmax(predictions[1])) #umjesto matrice prikazuje brojku

plt.imshow(x_test[1], cmap=plt.cm.binary) #brojke 1 parametar, a drugi za crno bijelo
plt.show() # naredba za prikazivanje
```

[30]

5.2.2. Prikaz korištenja umjetne neuronske mreže

Prikaz treniranja umjetnih neuronskih mreža u kojem se mogu vidjeti gubici i preciznosti koje nastaju prilikom treniranja umjetnih neuronskih mreža. Treniranje uz parametre nastaje sa sljedećim naredbama

```
model.compile(optimizer='adam',
              loss='sparse_categorical_crossentropy',
              metrics=['accuracy']
              ) #parametri za treniranje žnaujvaniji je loss zato što se po njemu
              optimizira, optimizer treba znati dobro matematiku
model.fit(x_train, y_train, epochs=3) #treniranje NN epoch =iteracija po ulaznom
              parametru
```

[30]

```
Epoch 1/3
1875/1875 [=====] - 3s 2ms/step - loss: 0.3129 - accuracy: 0.9132
Epoch 2/3
1875/1875 [=====] - 3s 2ms/step - loss: 0.1427 - accuracy: 0.9585
Epoch 3/3
1875/1875 [=====] - 3s 2ms/step - loss: 0.0982 - accuracy: 0.9710
<tensorflow.python.keras.callbacks.History at 0x7fcf88b24c18>
```

Slika 23: Treniranje (autor)

Prikaz parametara u kojima se mogu pratiti preciznosti i gubitci nastali tijekom treniranja modela umjetnih neuronskih mreža mogu se dobiti sa sljedećim naredbama

```
val_loss, val_acc = model.evaluate(x_test, y_test) #procjeni gubitke i preciznost
print(val_loss, val_acc)
```

[30]

```
313/313 [=====] - 0s 1ms/step - loss: 0.1033 - accuracy: 0.9688
0.10326087474822998 0.9688000082969666
```

Slika 24: Parametri (autor)

Prikaz matičnog rezultata ili izlaza istreniranog spremljenog modela umjetnih neuronskih mreža se mogu prikazati uz sljedeće naredbe

```
model.save('epic_num_reader.model')
new_model=tf.keras.models.load_model('epic_num_reader.model')
predictions= new_model([x_test])
print(predictions)
```

[30]

```
tf.Tensor(
[[[2.5153190e-07 1.2687776e-08 3.0202027e-05 ... 9.9842370e-01
  4.6189160e-07 4.3768841e-06]
 [1.4426335e-07 7.7214204e-03 9.9205106e-01 ... 3.1384589e-11
  3.7443360e-06 1.9027937e-12]
 [4.0856025e-06 9.9892825e-01 7.2025818e-05 ... 5.9411246e-05
  7.4187224e-04 3.8246694e-06]
 ...
 [1.4588989e-08 5.0511492e-08 5.4807224e-08 ... 8.0425882e-05
  2.5143722e-04 1.1646181e-03]
 [1.4611331e-05 5.0252997e-06 2.0681586e-07 ... 1.8108892e-06
  4.5835865e-03 2.8630140e-07]
 [5.6712975e-06 2.0702568e-07 8.4725332e-05 ... 1.2063005e-09
  1.9602790e-06 1.8780165e-08]], shape=(10000, 10), dtype=float32)
```

Slika 25: Matični izlaz (autor)

Prikaz brojanog rezultata ili izlaza istreniranog spremljenog modela umjetnih neuronskih mreža se mogu prikazati uz sljedeće naredbe

```
import numpy as np
print(np.argmax(predictions[1])) #umjesto matrice prikazuje brojku
```

[30]

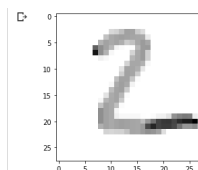
↳ 2

Slika 26: Brojčani izlaz (autor)

Prikaz slikovitog rezultata ili izlaza se pokazuje uz sljedeće naredbe

```
import matplotlib.pyplot as plt
plt.imshow(x_test[1], cmap=plt.cm.binary) #brojke 1 parametar, a drugi za crno bijelo
plt.show() #naredba za prikazivanje
```

[30]



Slika 27: Slikoviti izlaz (autor)

6. Zaključak

Umjetne neuronske mreže su stvarno impresivna metoda u kojim se razvija umjetna inteligencija. One potječu iz područja strojnog učenja koje je samo djelić umjetne inteligencije. Umjetne neuronske mreže na neki način oponašaju funkcionalnosti živčane stanice ili neurona koje su zaslužne za slanje i procesiranje informacija u mozgu. Različitim topologijama odnosno dizajnom strukture umjetne neuronske mreže dobivaju se mogućnosti rješavanja različitih problema. Umjetne neuronske mreže se najčešće specijaliziraju za rješenje određenog problema te nažalost se ne mogu mjeriti s kreativnošću ljudskog uma. Za specijalizirane probleme čak i u nekim segmentima nadilaze ljudske sposobnosti. Umjetne neuronske mreže se svakodnevno koriste prilikom pretraživanja na internetu ili korištenjem modernih tehnologija. Smatram da su umjetne neuronske mreže još u "razvoju" te da će njihovo korištenje biti učestalije u bližnjoj budućnosti.

Popis literature

- [1] overleaf.com. (2020). LaTeX, Evolved The easy to use, online, collaborative LaTeX editor, adresa: <https://www.overleaf.com/> (pogledano 15. 8. 2020).
- [2] startupbeat.com. (2014). Overleaf-by-writelatex-logo-300dpi, adresa: <https://startupbeat.com/writelatex-qa-id3915/13514/overleaf-by-writelatex-logo-300dpi1/> (pogledano 15. 8. 2020).
- [3] google.com. (2020). What is Colaboratory?, adresa: <https://colab.research.google.com/notebooks/intro.ipynb> (pogledano 15. 8. 2020).
- [4] tensorflow.org. (2020). An end-to-end open source machine learning platform, adresa: <https://www.tensorflow.org/> (pogledano 15. 8. 2020).
- [5] keras.io. (2020). About Keras, adresa: <https://keras.io/about/> (pogledano 15. 8. 2020).
- [6] B. Copeland. (2020). Artificial intelligence, adresa: <https://www.britannica.com/technology/artificial-intelligence> (pogledano 15. 8. 2020).
- [7] encklipedija.hr. (2020). Umjetna inteligencija, adresa: <https://www.enciklopedija.hr/natuknica.aspx?ID=63150> (pogledano 15. 8. 2020).
- [8] M. Rouse. (2020). Artificial intelligence, adresa: <https://searchenterpriseai.techtarget.com/definition/AI-Artificial-Intelligence#> (pogledano 15. 8. 2020).
- [9] MargaretRouse. (2019). Artificial Intelligence of Things (AIoT), adresa: <https://internetofthings.techtarget.com/definition/Artificial-Intelligence-of-Things-AIoT> (pogledano 15. 8. 2020).
- [10] IBM. (2020). AI Planning, adresa: https://researcher.watson.ibm.com/researcher/view_group.php?id=8432 (pogledano 15. 8. 2020).
- [11] C. Shih-Fen. (2019). Course Introductory - Multi Agent Systems, adresa: <https://www.youtube.com/watch?v=8GOQjDVdMJM> (pogledano 15. 8. 2020).
- [12] I. Mihajlovic. (2019). Everything You Ever Wanted To Know About Computer Vision, adresa: <https://towardsdatascience.com/everything-you-ever-wanted-to-know-about-computer-vision-heres-a-look-why-it-s-so-awesome-e8a58dfb641e> (pogledano 15. 8. 2020).
- [13] F. Chollet, *Deep Learning with Python*. Manning shelter island, 2018.

- [14] Matab. (2019). Artificial Intelligence, Enough of the hype! What is it?, adresa: <https://community.hpe.com/t5/hpe-blog-uk-ireland/artificial-intelligence-enough-of-the-hype-what-is-it/ba-p/7046672> (pogledano 15. 8. 2020).
- [15] D. Kriesel, *A Brief Introduction to Neural Networks*. David Kriesel, 2007.
- [16] V. Fedak. (2018). Top 10 Most Popular AI Models, adresa: <https://dzone.com/articles/top-10-most-popular-ai-models> (pogledano 15. 8. 2020).
- [17] T. S. t. E. i. H. e. B. Guido Diepen. (2017). Part 2: Artificial Intelligence Techniques Explained, adresa: <https://www2.deloitte.com/nl/nl/pages/data-analytics/articles/part-2-artificial-intelligence-techniques-explained.html> (pogledano 15. 8. 2020).
- [18] sciencefacts.net. (2020). Parts of a Neuron and Their Function, adresa: <https://www.sciencefacts.net/parts-of-a-neuron.html> (pogledano 15. 8. 2020).
- [19] J. Chen. (2020). Neural Network, adresa: <https://www.investopedia.com/terms/n/neuralnetwork.asp> (pogledano 15. 8. 2020).
- [20] K. Gurney, *An Introduction to Neural network*. TylorFrancis, 2004.
- [21] S. Shmarma. (2017). Activation Functions in Neural Networks, adresa: <https://towardsdatascience.com/activation-functions-neural-networks-1cbd9f8d91d6> (pogledano 15. 8. 2020).
- [22] S. Russell i P. Norvig, *Artificial Intelligence: A Modern Approach*. Prentice Hall, 2010.
- [23] 3blue1brown.com. (2020), adresa: <https://www.3blue1brown.com/neural-networks> (pogledano 15. 8. 2020).
- [24] G. N. Yannakakis i J. Togelius, *Artificial Intelligence and Games*. Springer International Publishing, 2018.
- [25] A. Tch. (2017). The mostly complete chart of Neural Networks, explained, adresa: <https://towardsdatascience.com/the-mostly-complete-chart-of-neural-networks-explained-3fb6f2367464> (pogledano 15. 8. 2020).
- [26] Shreyak. (2020). A to Z about Artificial Neural Networks (ANN) (Theory N Hands-on), adresa: <https://medium.com/analytics-vidhya/a-to-z-about-artificial-neural-networks-ann-theory-n-hands-on-713c12f3351e> (pogledano 15. 8. 2020).
- [27] F. V. Veen. (2017). Neural Network Zoo Prequel: Cells and Layers, adresa: <https://www.asimovinstitute.org/author/fjodorvanveen/> (pogledano 9. 9. 2020).
- [28] stackexchange.com. (2019). What is the definition of each of these neural network cell types?, adresa: <https://ai.stackexchange.com/questions/5898/what-is-the-definition-of-each-of-these-neural-network-cell-types> (pogledano 9. 9. 2020).
- [29] R. Chandra. (2018). Neural Networks: Applications in the Real World, adresa: <https://www.upgrad.com/blog/neural-networks-applications-in-the-real-world/> (pogledano 9. 9. 2020).

- [30] sentdex. (2018). Deep Learning with Python, TensorFlow, and Keras tutorial, adresa: <https://www.youtube.com/watch?v=wQ8BIBpya2k> (pogledano 15. 8. 2020).
- [31] ashukumar27.io. (2018). Handwritten digit recognition in MNIST, adresa: <http://www.ashukumar27.io/mnist-cnn-keras/#content> (pogledano 15. 8. 2020).

Popis slika

1.	Overleaf ([2])	2
2.	Slojeviti prikaz AI ([14]), slobodan prijevod	6
3.	Linearna regresija ([16])	7
4.	Logistička regresija ([16]), slobodan prijevod	8
5.	Stablo odluke ([16]), slobodan prijevod	8
6.	Šuma nasumičnih odluka ([16]), slobodan prijevod	9
7.	Naive Bayes ([16]), slobodan prijevod	9
8.	Strojevi za podršku vektora 1 ([17])	10
9.	Strojevi za podršku vektora 2([17])	10
10.	Neuron ([18]), slobodan prijevod	11
11.	Umjetni neuron ([15, str. 35]), slobodan prijevod	13
12.	Sigmoidna i ReLU funkcija([21])	15
13.	Topologija umjetne neuronske mreže ([25]), slobodan prijevod	17
14.	Perceptron ([25])	18
15.	Duboke unaprijedne umjetne neuronske mreže ([26]), slobodan prijevod	19
16.	Rekurzivne ili ponavljajuće umjetne neuronske mreže ([25])	19
17.	Umjetne neuronske mreže sa zatvorenim ponavljajućim neuronima ([25])	20
18.	Hopfieldove umjetne neuronske mreže ([25])	20
19.	Duboko vjerujuće umjetne neuronske mreže ([25])	21
20.	Duboko konvolucijske umjetne neuronske mreže ([25])	22
21.	Duboko konvolucijske grafički obrnute umjetne neuronske mreže ([25])	22
22.	Brojke ([31])	23
23.	Treniranje (autor)	25

24. Parametri (autor)	25
25. Matrični izlaz (autor)	26
26. Brojčani izlaz (autor)	26
27. Slikoviti izlaz (autor)	26