

Predviđanje količine dolaznog i odlaznog prometa komunikacijskih mreža pomoću neuronske mreže

Tomljenović, Andrija

Master's thesis / Diplomski rad

2020

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Zagreb, Faculty of Transport and Traffic Sciences / Sveučilište u Zagrebu, Fakultet prometnih znanosti**

Permanent link / Trajna poveznica: <https://urn.nsk.hr/urn:nbn:hr:119:160199>

Rights / Prava: [In copyright](#)/[Zaštićeno autorskim pravom.](#)

Download date / Datum preuzimanja: **2024-11-27**



Repository / Repozitorij:

[Faculty of Transport and Traffic Sciences -
Institutional Repository](#)



SVEUČILIŠTE U ZAGREBU
FAKULTET PROMETNIH ZNANOSTI

Andrija Tomljenović

**PREDVIĐANJE KOLIČINE DOLAZNOG I ODLAZNOG PROMETA
KOMUNIKACIJSKIH MREŽA POMOĆU NEURONSKE MREŽE**

DIPLOMSKI RAD

Zagreb, 2020.

Zagreb, 8. travnja 2020.

Zavod: **Zavod za inteligentne transportne sustave**
Predmet: **Umjetna inteligencija**

DIPLOMSKI ZADATAK br. 5897


Pristupnik: **Andrija Tomljenović (0135243327)**
Studij: **Promet**
Smjer: **Informacijsko-komunikacijski promet**

Zadatak: **Predviđanje količine dolaznog i odlaznog prometa komunikacijskih mreža pomoću neuronske mreže**

Opis zadatka:

Danas postoji velika potreba za brzom razmjenom podataka primjenom komunikacijskih mreža. Za optimalno korištenje postojeće komunikacijske infrastrukture sve se više koriste napredne metode iz područja umjetne inteligencije. Takve metode zahtijevaju kvalitetne podatke o trenutnom i budućem predviđenom stanju komunikacijske mreže. U radu je potrebno opisati probleme u predviđanju količine prometa komunikacijskih mreža, analizirati mogućnost primjene neuronskih mreža za predviđanje količine prometa, predložiti strukturu neuronske mreže te ju nakon učenja provjeriti na odabranom skupu podataka količine prometa komunikacijske mreže.

Mentor:



izv. prof. dr. sc. Edouard Ivanjko

Predsjednik povjerenstva za
diplomski ispit:



Sveučilište u Zagrebu
Fakultet prometnih znanosti

DIPLOMSKI RAD

**PREDVIĐANJE KOLIČINE DOLAZNOG I ODLAZNOG PROMETA
KOMUNIKACIJSKIH MREŽA POMOĆU NEURONSKE MREŽE**

**PREDICTION OF UPLINK AND DOWNLINK TRAFFIC OF COMMUNICATION
NETWORKS USING NEURAL NETWORKS**

Mentor: Izv. prof. dr. sc. Eduard Ivanjko

Student: Andrija Tomljenović

JMBAG: 0135243327

Zagreb, rujan 2020.

SAŽETAK

U ovom diplomskom radu cilj je predvidjeti količina prometa na dolaznom i odlaznom linku koristeći neuronske mreže. Kreirano je nekoliko mreža koje su se na temelju prikupljenih podataka naučile karakteristikama prometnog toka. Korišteni podatkovni skup prikupljan je unutar perioda od jedne godine u petominutnim koracima. Rezultati rada pokazuju da neuronska mreža ima vrlo dobru sposobnost naučiti proces prema predstavljenim podacima, ali i da rezultati koje mreža daje uvelike ovise o karakteristikama podatkovnog skupa na temelju kojega se uči.

KLJUČNE RIJEČI: neuronske mreže, umjetna inteligencija, predviđanje prometa, telekomunikacijske mreže.

SUMMARY

TITLE: Prediction of Uplink and Downlink Traffic of Communication Networks using Neural Networks

Goal of this thesis is predicting amount of uplink and downlink traffic using neural networks. Several networks were created which, based on the collected data, learned the characteristics of the traffic flow. Used data set was collected over period of one year in five-minute steps. The results of this work show that neural network has a very good ability to learn the process according to the presented data, but also that the results that the network gives largely depend on the characteristics of the data set on which network has been learning.

KEY WORDS: neural networks, artificial intelligence, traffic prediction, telecommunication networks.

Sadržaj:

| | |
|---|----|
| 1. UVOD | 1 |
| 2. Predviđanje prometa komunikacijskih mreža | 3 |
| 2.1. Naivne metode | 4 |
| 2.2. Parametarski modeli | 5 |
| 2.3. Neparametarski modeli | 6 |
| 3. Primjena neuronskih mreža u predviđanju komunikacijskog prometa | 8 |
| 3.1. Neuronske mreže | 8 |
| 3.1.1. Procesne jedinice - neuroni | 9 |
| 3.1.2. Aktivacijske funkcije | 11 |
| 3.1.3. Svojstva i struktura neuronske mreže | 14 |
| 3.1.4. Učenje neuronske mreže | 16 |
| 3.2. Primjena neuronskih mreža u telekomunikacijskom prometu | 19 |
| 4. Određivanje strukture neuronske mreže | 23 |
| 4.1. MATLAB programski paket | 23 |
| 4.2. Definiranje strukture neuronske mreže | 26 |
| 5. Podatkovni skup korišten za učenje neuronske mreže | 28 |
| 5.1. Opis podatkovnog skupa | 28 |
| 5.2. Učenje neuronske mreže | 30 |
| 5.2.1. Obrada podataka i priprema ulaznih vektora | 30 |
| 5.2.2. Učenje neuronske mreže | 32 |
| 6. Rezultati učenja i predviđanja | 35 |
| 6.1. Analiza rezultata za predviđanje odlaznog prometa | 36 |
| 6.2. Analiza rezultata za predviđanje dolaznog prometa | 39 |
| 7. Zaključak | 43 |
| POPIS LITERATURE | 44 |

| | |
|--------------------------------|-----------|
| POPIS ILUSTRACIJA | 46 |
| POPIS TABLICA | 47 |

1. UVOD

Razvitak tehnologije dao je veliki doprinos porastu prometa koji se odvija današnjim telekomunikacijskim mrežama. Aplikacije postaju sve gladnije u pogledu prometa koji stvaraju i resursa koje zauzimaju. Postojeća infrastruktura postaje sve opterećenija stoga je potrebno primijeniti nove tehnologije i u mrežnom dijelu. Jedna od takvih tehnologija, a koja se može iskoristiti u svrhu poboljšanja performansi telekomunikacijske mreže, jesu neuronske mreže. Iako matematički modeli postoje već desetljećima tek su u skorije vrijeme razvitkom računala pronašle svoju primjenu.

U ovome diplomskom radu cilj je predvidjeti količinu prometa kroz link u odlaznom i dolaznom smjeru koristeći tehnologiju neuronskih mreža. Na temelju korištenih podataka biti će ispitano nekoliko arhitektura i vrsta neuronske mreže te predstaviti one s najboljim rezultatima. Sirove podatke je potrebno prvo obraditi kako bi bili povoljniji za učenje neuronske mreže te kreirati ulazne vektore.

Rad je podijeljen u sedam cjelina:

1. Uvod
2. Predviđanje prometa komunikacijskih mreža
3. Primjena neuronskih mreža u predviđanju komunikacijskog prometa
4. Određivanje strukture neuronske mreže
5. Podatkovni skup korišten za učenje neuronske mreže
6. Rezultati učenja i predviđanja
7. Zaključak

U drugom poglavlju opisani su modeli predviđanja prometa. Navedena je podjela modela koji se koriste pri predviđanju prometa te je navedena i podjela predviđanja.

U trećem poglavlju opisane su neuronske mreže, njihov način rada i vrste. Također, navedeno je i pojašnjeno nekoliko primjena neuronskih mreža u telekomunikacijskim sustavima.

U četvrtom poglavlju ukratko se opisuje programski paket korišten za izradu i učenje neuronskih mreža. Nakon toga se opisuje struktura neuronskih mreža koje su korištene za predviđanje količine prometa.

U petom poglavlju prvo se opisuje podatkovni skup koji se koristio za učenje neuronske mreže. Zatim se opisuje postupak pripreme podataka za učenje neuronske mreže i ukratko sam postupak učenja.

Šesto poglavlje sastoji se od dva dijela. U prvome potpoglavlju opisuju se rezultati predviđanja prometa odlaznog linka, a u drugome dolaznog linka.

Rad završava zaključkom i opisom potencijalnog nastavka rada.

2. Predviđanje prometa komunikacijskih mreža

Predviđanje prometa u telekomunikacijskim mrežama važan je dio kao i alat pri učinkovitom upravljanju mrežom i mrežnim resursima. Analiza i predviđanje količine mrežnog prometa predstavljaju značajne tehnike za uspješne sheme preventivne kontrole zagušenja. Ovakav pristup predstavlja proaktivno rješenje za razliku od reaktivnog odnosno omogućuje sprječavanje određenih problema prije nego se oni i pojave. Osim poboljšanja performansi mreže u smislu smanjenja zagušenja i poboljšanja kvalitete usluge predviđanje količine prometa može se iskoristiti i u svrhe poboljšanja sigurnosti mreže. Usporedbom stvarnih vrijednosti i predviđenih mogu se uočiti anomalije to jest velika odstupanja te tako detektirati sigurnosni problemi. Konkretno predviđanjem prometa pozitivno se utječe na dinamičko dodjeljivanje pojasne širine, sigurnosti mreže, planiranju mreže kao i već spomenuto prediktivno sprječavanje zagušenja mreže, [1].

Podjela predviđanja može se napraviti prema vremenskom dometu odnosno duljini vremenskog intervala za kojeg se predviđaju vrijednosti. Ugrubo se predviđanja mogu podijeliti na dugoročna i kratkoročna, a prema [2] navodi se nešto detaljnija podjela se četiri vrste predviđanja u odnosu na vremenski aspekt:

- Dugoročna predviđanja: unaprijed se predviđa interval od nekoliko mjeseci pa sve do nekoliko godina. Ovakva predviđanja koriste se za planiranje i donošenje strateških odluka;
- Srednje duga predviđanja: predviđa se nekoliko dana unaprijed, a sa svrhom planiranja resursa;
- Kratkoročna predviđanja: predviđaju se vrijednosti do nekoliko sati, te se koriste u svrhu kontrole i detekcije anomalija;
- Stvarno vremena predviđanja: predviđanja do nekoliko minuta, ovakva predviđanja zahtijevaju ugrađene sustave koji imaju stvarno remeni pristup podacima.

S obzirom na to da se praktični dio ovoga rada odnosi na kratkoročno predviđanje takvim modelima će se pridodati više pažnje.

Informacije o vrijednostima parametara prometnog toka imaju potencijal poboljšati rad mreže. No kako bi se ti efekti uspješno realizirali potrebno je imati točne i pouzdane informacije. Stoga je moguće zaključiti da kratkoročna predviđanja predstavljaju uvjet za

napredak i poboljšanje telekomunikacijskih mreža. S obzirom na to da skup podataka o prometu komunikacijske mreže nelinearan skup predviđanje takvih vrijednosti čini kompleksni nelinearni problem, [3].

Modele za kratkoročno predviđanje moguće je podijeliti u tri potkategorije:

1. Naivne metode;
2. Parametarski modeli;
3. Neparametarski modeli.

2.1. Naivne metode

Naivne metode su vrlo raširene jer iziskuju vrlo malo računalne snage i lako se implementiraju. To su zapravo samo metode odnosno ne temelje se na modelima. Također, karakterizira ih vrlo mala točnost te u većini slučajeva bilo koja parametarska ili ne parametarska metoda daje bolje rezultate. Prema [3] navedene su sljedeće metode:

- Trenutna metoda;
- Povijesni prosjeci;
- Grupiranje.

Trenutna metoda pretpostavlja da su parametri prometa koji se predviđaju konstantne prirode. Shodno tome i ne zahtijevaju ikakvo računanje. S obzirom na to da su prometne vrijednosti komunikacijske mreže nelinearne karakteristike i nikako nisu konstante ova metoda daje vrlo loše rezultate

Metoda povijesnih prosjeka računa prosjek na osnovi prethodnih vrijednosti za pojedinu varijablu. Ovu metodu moguće je kombinirati s trenutnom metodom kako bi se poboljšala točnost predviđanja.

Metode grupiranja koriste prosječne vrijednosti promatrane varijable unutar specificirane grupe dana, a na temelju sličnih obrazaca vrijednosti. Mogu se koristiti za pred obradu ulaznih podataka. Ove metode su točnije od metode povijesnih prosjeka pa čak i u nekim slučajevima od metode linearne regresije, ali su grube te se zbog toga rijetko koriste.

2.2. Parametarski modeli

U parametarskim modelima potrebno je pronaći parametre prema kojima će model davati zadovoljavajuće rezultate. Model sam po sebi odnosno njegova struktura je predodređena te, kako je već navedeno, je potrebno odrediti parametre za ulazne podatke. Znanje o prometnom procesu je moguće implementirati u ovakve modele te se posebni slučajevi mogu modelirati. Također, potrebna je manja količina podataka za predviđanje u odnosu na neparametarske modele. Prema [3] navode se sljedeći modeli:

- Makroskopski, mikroskopski i mezoskopski;
- Vremenske serije;
- Linearna regresija;
- ARIMA (engl. *Autoregressive Integrated Moving Average*);
- Kalmanov filter;
- SEATAR (engl. *The Self-Exciting Threshold Auto Regressive*).

Makroskopski, mikroskopski i mezoskopski modeli se razlikuju prema razini detalja koji promatraju. Pa tako makroskopski model promatra mrežu globalno te uzima samo globalne parametre, ne ulazi u dubinu mreže odnosno ne uzima parametre zasebnih dijelova mreže ili pojedinih čvorova. Ovaj model karakteriziraju jednostavnost i brz izračun rezultata. U mikroskopskom modelu promatraju se pojedine prometne jedinice, ako se govori o Internet mreži to će biti pojedini paket ili se može promatrati i pojedina sesija koja se uspostavlja. Mezoskopski modeli kombiniraju makroskopske i mikroskopske modele, na temelju makroskopskih varijabli modeliraju se mikroskopski modeli.

Model vremenske serije pokušava modelirati promatranu varijablu kao funkciju na temelju njenih prošlih vrijednosti i na temelju pogreške. Kako bi model vremenske serije dao validne rezultate potrebno je da promatrani proces bude stacionaran. S obzirom na to da komunikacijski promet nije stacionaran proces i pojavljuje se sezonalnost (engl. *Seasonality*) pa je potrebno i to uvrstiti u model. Pojam sezonalnosti označava da se u pojedinim periodima pojavljuju veće fluktuacije u podacima, npr. povećano opterećenje mreže prilikom praznika, vikenda i slično.

U modelu linearne regresije pretpostavlja se da je funkcija predviđanja linearna kombinacija svojih kovarijanci, a parametri određuju koliko koja kovarijanca doprinosi izlazu. Ovaj model

odlikuje jednostavnost, a posljedično tome i brzo se dolazi do rezultata te su istraživanja pokazala da može dati dobre rezultate.

ARIMA odnosno autoregresivni integrirani pomični prosjek je statistička tehnika koja se može koristiti kao model za predviđanje. ARIMA model je zapravo generalizirani ARMA (engl. *Autoregressive Moving Average*), a koristi se za predviđanje vremenskih serija. Postoji mnoštvo varijanti ovog modela od kojih su neke: SARIMA (engl. *Seasonal Autoregressive Integrated Moving Average*), podskup ARIMA, Kohonen ARIMA i ostale.

Kalmanov filter funkcionira tako da na temelju trenutne i prve prethodne vrijednosti pretpostavlja sljedeću vrijednost. Kalmanov filter može dati dobre rezultate ali ponekad su ostali modeli točniji.

SETAR model na temelju linearne kombinacije trenutnog mjerenja i jednog mjerenja iz prošlosti nastoji predvidjeti buduću vrijednost tako da trenutno mjerenje ima veću težinu. Iako brz ovaj model ima malu točnost, [3].

2.3. Neparametarski modeli

Neparametarski modeli, iako bi se po imenu zaključilo suprotno, također imaju parametre ali su znatno fleksibilniji nego oni u parametarskim modelima. Broj parametara nije unaprijed određen te se točan model kao i parametar moraju odrediti na temelju podataka nad kojima se radi predviđanje. Prednost neparametarskih modela nad parametarskim je da se mogu predviđati i nelinearni, dinamički kao i teški procesi. Ovakvi modeli zahtijevaju velike količine podataka kako bi dali bolje rezultate. Kvaliteta izlaza ovisi i o kakvoći podataka odnosno ako se neke sezonske karakteristike ne nalaze u podacima na temelju koji se predviđa model neće moći predvidjeti takve fluktuacije. Također, prednost neparametarskih modela je što ne zahtijevaju poznavanje procesa čija stanja se predviđaju. Važniji neparametarski modeli jesu, [3]:

- K-najbližih susjeda;
- Lokalna regresija s težinskim faktorima (engl. *Locally Weighted Regression*);
- Neizrazita logika;
- Bayesove mreže (engl. *Bayesian networks*);
- Neuronske mreže.

Model k -najbližih susjeda je algoritam koji pretpostavlja da se slične stvari, u ovome slučaju vrijednosti nalaze u blizini. Parametar k koji se nalazi u nazivu ovoga algoritma označava broj susjeda koji će se uzeti u obzir. Algoritam izračunava udaljenost odnosno razliku ili pogrešku trenutnog stanja mreže od svih poznatih vrijednosti te kao izlaz daje prosjek k najbližih susjeda, [21].

Lokalna regresija s težinskim faktorima koristi modele lokalne regresije. Koristi se vrijednosti prethodnih mjerenja uzimajući u obzir udaljenost od trenutnog mjerenja, ova metoda daje dobre rezultate u kratkom vremenu.

Neizrazita logika (engl. *Fuzzy logic*) se koristi neizrazitim skupovima odnosno promatra se pripadnost vrijednosti parametra pojedinim pravilima. Pravila su strukturirana u obliku ako-onda pravila zaključivanja. Na temelju tih pravila i razine pripadnosti određuje se predviđana vrijednost.

Bayesova mreža je usmjereni grafički model, a predstavlja ovisnosti između nasumičnih varijabli. U ovome modelu podaci sa susjednih linkova uzimaju se kako bi se saznali parametri promatranog linka.

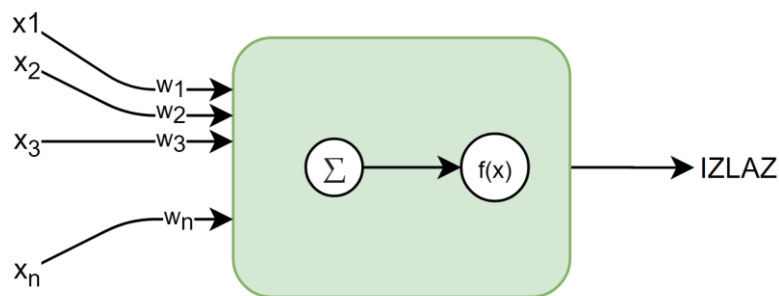
Neuronske mreže su modeli koji se temelje na strukturi ljudskog mozga i njegovih procesnih elemenata odnosno neurona. Modeli neuronskih mreža pogodni su za predviđanje prometa jer su sposobni dobro se prilagoditi nelinearnim i dinamičkim procesima. Zbog toga su često korišteni u predviđanju prometa. Model neuronske mreže detaljno će se opisati u jednom od sljedećih poglavlja, [3].

3. Primjena neuronskih mreža u predviđanju komunikacijskog prometa

S obzirom na svoje sposobnosti koje će biti pobliže opisane u sljedećim poglavljima neuronske mreže imaju širok raspon primjena u svim granama znanosti. Pa se tako pronalazi i mnoštvo primjena u telekomunikacijskim sustavima. U sljedećem poglavlju biti će definirana neuronska mreža i opisane njene karakteristike. Također, navesti će se nekoliko konkretnih primjena umjetnih neuronskih mreža u telekomunikacijskim sustavima.

3.1. Neuronske mreže

Umjetne neuronske mreže nastoje imitirati strukturu ljudskog mozga. Prvi model neurona predložen je 1943. godine, a predložili su ga McCulloch i Pitts. Svoj model opisali su kao mrežu s promjenjivim sinapsama. Neuron u predloženoj mreži mogao je imati dva stanja odnosno 1 ili 0, a ovisno o zbroju ulaznih sinapsi. 1950. godine prvi puta je implementiran umjetni neuron koristeći računalnu simulaciju. Sljedeće važno otkriće je model perceptrona čiji je model definiran 8 godina kasnije od prve implementacije umjetnog neurona u računalnoj simulaciji. Perceptron je zapravo jednoslojna neuronska mreža u današnjem smislu i karakteristika mu je da je aktivacijska funkcija koraka, a sastoji se od ulaznih vrijednosti, težina, stalnog odmaka (engl. *bias*), sume težina i aktivacijske funkcije. Danas se koristi u sklopu nadziranog učenja, a model perceptrona prikazan je na slici 1, [4], [5].



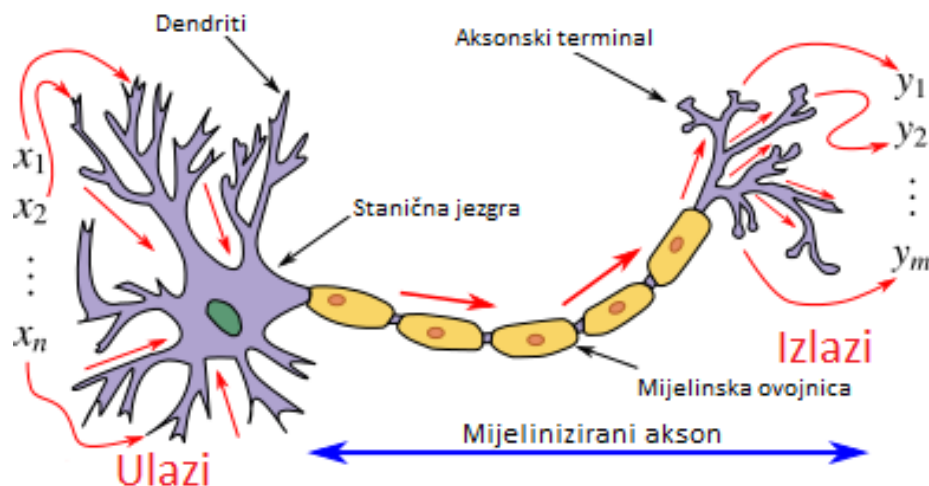
Slika 1. Model perceptrona
Izvor: [5]

ADALINE (engl. *Adaptive Linear Neuron*) je model koji je zapravo unaprjeđenje perceptrona a definiran je 1960. godine. Nakon ADALINE modela na dulji period istraživanje neuronskih mreža prestaje jer se sumnjalo u njihove sposobnosti. Godine 1986. definirani su

višeslojni perceptron i model povratnog rasprostiranja pogreške. Nedugo nakon toga umjesto modela povratnog rasprostiranja predložen je RBFNN (engl. *Radial Basis Function Neural Networks*), to su mreže koje kao svoju aktivacijsku funkciju koriste funkciju radijalne osnove. 1995. godine razvijen je SVM (engl. *Support Vector Machines*) model neuronske mreže koji je davao dobre rezultate u klasifikacijama i analizi regresije. Godine 2006. razvijen je model dubokih neuronskih mreža sa teorijom da što je mreža dublja može naučiti više, a nastale su na temelju modela višeslojnih perceptrona odnosno kao njegova nadogradnja. U to vrijeme pojavila se i ideja o evolucijskim neuronskim mrežama na temelju teorije evolucije, [4].

3.1.1. Procesne jedinice - neuroni

Kako joj i samo ime govori glavni element neuronske mreže je neuron. Procesni elementi to jest neuroni koji se nalaze u umjetnim neuronskim mrežama zapravo predstavljaju matematičke modele bioloških neurona koje možemo pronaći u kori ljudskog mozga. Na slici 2 prikazan je biološki neuron i on čini osnovnu građevnu jedinicu živčanog sustava. Kada se govori o biološkim neuronima njima se prenose odnosno obrađuju električni signali, a u umjetnim neuronima obrađuju se brojevi koristeći aritmetičke operacije i složene funkcije.



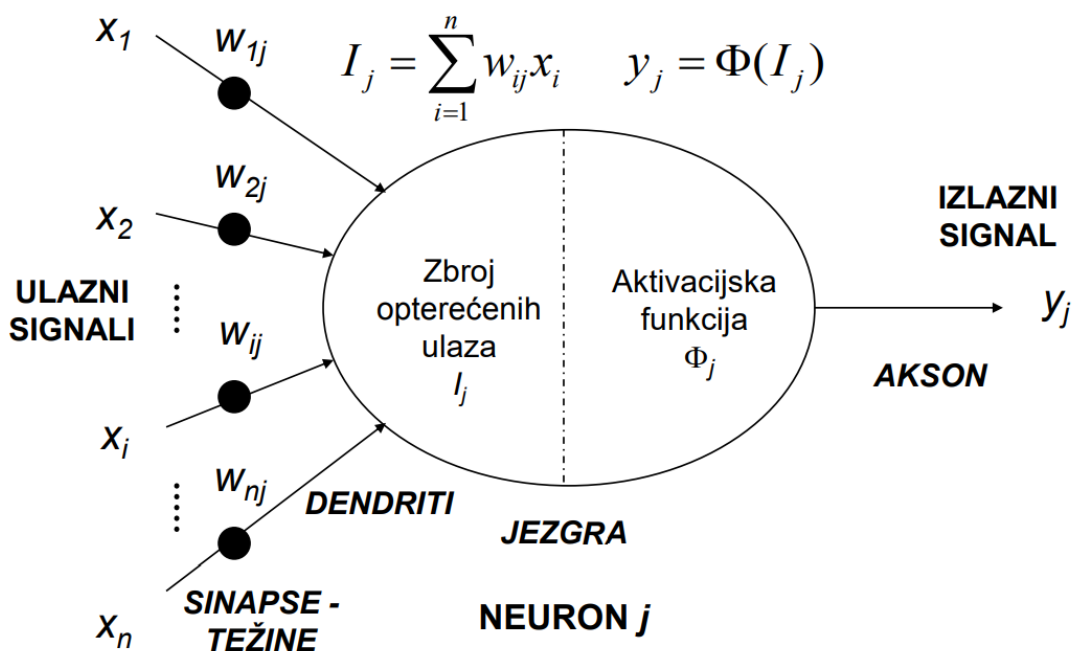
Slika 2. Prikaz biološkog neurona
Izvor: [7]

Kako je prikazano na slici 2 neuron se sastoji od, [6]:

- Dendrita;
- Aksona;

- Sinapse;
- Stanične jezgre.

Dendriti predstavljaju ulaz u neuron, odnosno tu su povezani sa drugim neuronima od kojih primaju signale. Akson predstavlja izlaz neurona i služi za prosljeđivanje izlaznog signala drugim neuronima nakon što je on obrađen u staničnoj jezgri. Sinapsa je međuprostor između dendrita jednog neurona i aksona drugoga. Sinapsa je ispunjena tekućinom u kojoj se nalaze prijenosnici signala, ovisno o potrebi za podražajem prijenosnici signala mogu pojačati ili smanjiti prijenos signala iz jednog u drugi neuron. Na temelju sinapse odnosno ubrzavanja ili usporavanja prijenosa signala neuronska mreža uči i pamti. Stanična jezgra služi obradi signala, u biološkom neuronu to je biološki proces dok se u umjetnom neuronu to svodi na matematičku funkciju, [6], [7].



Slika 3. Prikaz umjetnog neurona
Izvor [6]

Slika 3 prikazuje model umjetnog neurona. Na lijevoj strani nalaze se ulazi u neuron, ulazne vrijednosti mogu biti izlazi drugih neurona prethodnog sloja ili ako se radi o ulaznom sloju onda su to početne vrijednosti to jest vrijednosti na temelju kojih umjetna neuronska mreža treba doći do određenog zaključka. Svaka ulazna vrijednost ima svoju težinu kojom se određuje njen utjecaj na konačni rezultat. Računa se zbroj svih ulaza uzimajući u obzir njihove težine,

w_{ij} predstavljaju težinu vezanu za pojedini ulaz, a x_i se odnosi na samu vrijednost ulaza pa dobivamo jednadžbu:

$$I_j = w_1x_1 + w_2x_2 + \dots + w_nx_n, \quad (1)$$

odnosno:

$$I_j = \sum_{i=1}^n w_ix_i. \quad (2)$$

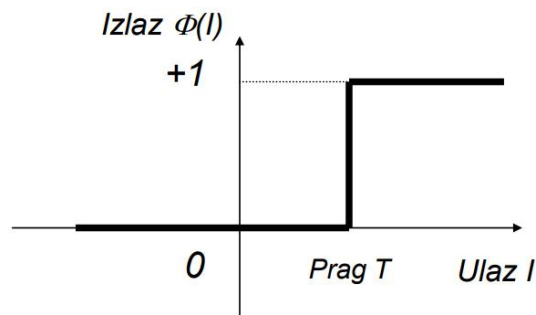
Navedena suma se zatim predaje aktivacijskoj funkciji Φ_j te ovisno o pobudi aktivacijske funkcije dobiva se odgovarajuća izlazna vrijednost:

$$y_j = \Phi_j(\sum_{i=1}^n w_ix_i) = \Phi_j(I_j). \quad (3)$$

3.1.2. Aktivacijske funkcije

Aktivacijska funkcija je zapravo ono što procesira podatke u umjetnom neuronu. Zahvaljujući njoj neuron se aktivira ili ostaje neaktivan odnosno daje veću ili manju vrijednost kao izlaz. Najjednostavnija aktivacijska funkcija je funkcija praga, a naziva se još i funkcija skoka ili binarna funkcija (engl. *Binary Step Function*) te ona kao svoj izlaz daje Booleovu vrijednost. Na slici 4 prikazana je funkcija praga. Kao što joj i samo ime govori, funkcija praga, kada ulazna vrijednost prijeđe određeni prag neuron se aktivira i prosljeđuje signal na svoje izlaze. Neuroni s ovom aktivacijskom funkcijom nazivaju se i perceptroni. Ova funkcija nije pogodna za složene nelinearne probleme.

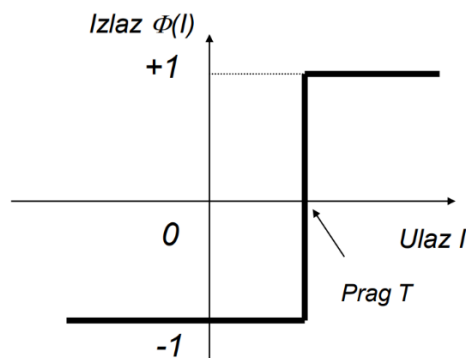
Funkcija skoka, praga $\Phi(I) = \begin{cases} 1, & I > T \\ 0, & I \leq T \end{cases}$



Slika 4. Prikaz funkcije praga
Izvor: [6]

Još jedna osnovna, a slična funkcija prethodno spomenutoj funkciji praga jest funkcija predznaka. Ona se razlikuje od funkcije praga samo po tome što umjesto 0 ili 1 kao izlaznih vrijednosti daje +1 ili -1, a otud joj i naziv. Na slici 5 prikazan je graf funkcije predznaka i njena jednadžba.

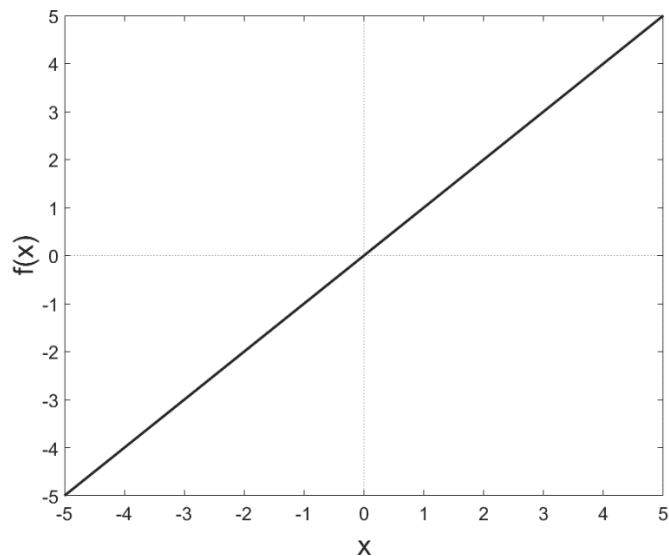
Funkcija predznaka $\Phi(I) = \begin{cases} +1, & I > T \\ -1, & I \leq T \end{cases}$



Slika 5. Prikaz funkcije predznaka
Izvor: [6]

Linearne funkcije na svojem izlazu daju vrijednost proporcionalnu sumi ulaza i težina. Njihova prednost nad funkcijom koraka i funkcijom predznaka je što kao izlaz mogu dati više vrijednosti za razliku od prethodno navedenih funkcija čiji izlaz je binarne prirode. Slika 6 prikazuje linearnu funkciju. Mana linearne funkcije je što se ne može koristiti metoda povratnog rasprostiranja pogreške pri treniranju mreže. Ako se cijela umjetna neuronska mreža sastoji od

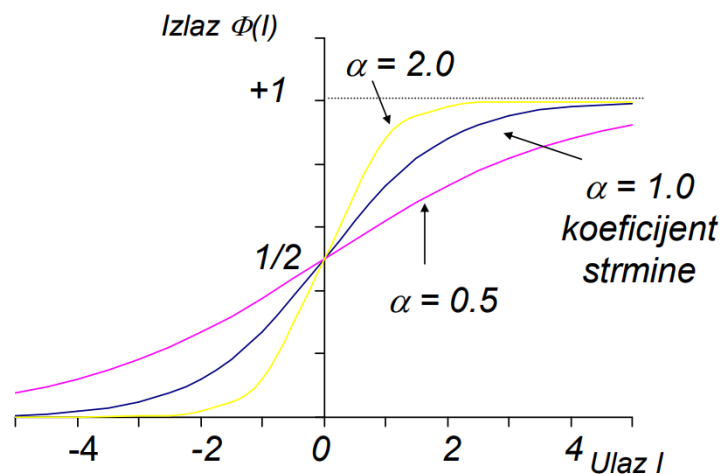
neurona s linearnom aktivacijskom funkcijom to znatno ograničava mrežu i onemogućuje joj da uči na temelju kompleksnih podataka, [8].



Slika 6. Prikaz linearne funkcije
Izvor: izradio autor

Logistička ili sigmoidna funkcija je najčešće korištena aktivacijska funkcija. Velika prednost nad prethodno navedenim funkcijama je što je derivabilna što omogućuje korištenje povratnog rasprostiranja i kreiranje dubokih neuronskih mreža, [8]. Na slici 7 prikazana je logistička funkcija i njeni parametri te promjene ovisno o vrijednostima parametara.

Logistička funkcija (sigmoid) $\Phi(I) = \frac{1}{1 + e^{-\alpha I}}$



Slika 7. Prikaz logističke funkcije
Izvor: [6]

3.1.3. Svojstva i struktura neuronske mreže

U prethodnim poglavljima opisani su osnovni elementi neuronske mreže to jest neuroni. Neuron sam za sebe nema veliku procesnu moć ali kada se više neurona posloži u strukturu odnosno mrežu pojavljuje se mogućnost učenja složenih procesa. Iako su umjetne neuronske mreže kompleksni sustavi i daju dobre rezultate u raznim primjenama zapravo su vrlo primitivne u odnosu na biološke neuronske mreže. Umjetnu neuronsku mrežu moguće je definirati kao masivno paralelni distribuirani procesor čija je odlika pamćenje iskustvenog znanja, [9], [10]. Prema [9] neuronsku mrežu, naspram nekih drugih metoda obrade podataka, odlikuju:

- Mogućnost rada s velikim brojem parametara i varijabli;
- Sposobne su raditi s nejasnim ili manjkavim podacima te u takvim skupovima prepoznati uzorke. Ovakvi podaci tipični su za različita osjetila;
- Robusne su na pogreške u skupovima podataka;
- Imaju vrlo dobru sposobnost procjene nelinearnih odnosa uzoraka;
- Stvaraju vlastite odnose između podataka koji nisu eksplicitan simbolički način;
- Same formiraju znanje na temelju prikazanih primjera to jest na temelju iskustva;
- Prilagođuju se okolini;
- Mogućnost jednostavne VLSI (engl. *Very large scale integration*) implemetacije;
- Imaju sposobnost naučiti složene procese.

Kako se navodi u [7] neuronsku mrežu smatra se univerzalnim aproksimatorom funkcija (engl. *Universal Function Approximator*) što znači da ima sposobnost naučiti i računati bilo koju funkciju. S obzirom na svoje sposobnosti neuronske mreže se mogu primijeniti gotovo u bilo kojem polju znanosti, a neke od najčešćih primjena jesu, [9]:

- Raspoznavanje uzoraka;
- Obrada slika;
- Obrada govora;
- Problemi optimizacije;
- Predviđanje stanja složenih procesa;
- Nelinearno upravljanje;
- Obrada nepreciznih i nekompletnih podataka;

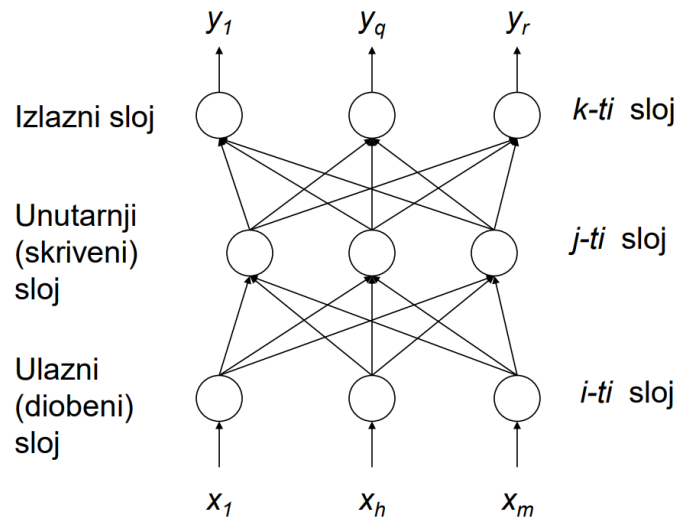
- Simulacije.

Arhitektura mreže određuje na koji način su neuroni povezani te kako međusobno utječu jedni na druge. Razlikuju se četiri osnovne topologije, [10]:

- Jednoslojne mreže bez povratnih veza (engl. *Single-layer feedforward networks*);
- Višeslojne mreže bez povratnih veza (engl. *multi-layer feedforward networks*);
- Mreže s povratnim vezama (engl. *Recurrent networks*);
- Ljestvičaste mreže (engl. *Lattice structures*).

Jednoslojne neuronske mreže sastoje se samo od jednog sloja neurona. Taj sloj može imati više neurona. Ulazi u mrežu spojeni su direktno na ulaze neurona, a izlazi neurona predstavljaju i izlaz mreže. U jednoslojnim neuronskim mrežama nema povratnih veza.

Višeslojne neuronske mreže sastoje se od ulaznog, izlaznog i jednog ili više skrivenih slojeva. Na slici 8 prikazana je topologija višeslojne neuronske mreže s jednim skrivenim slojem. Izlazi n -tog sloja predstavljaju ulaze $n+1$ -tog sloja osim ako se, naravno, ne radi o izlaznom sloju. Višeslojne mreže mogu se podijeliti na potpuno povezani i djelomično povezane. Potpuno povezana mreža je ona čiji su svi neuroni n -tog sloja povezani sa svim neuronima $n+1$ -tog sloja. Neuroni djelomično povezane mreže imaju konekcije samo na neke neurone u sljedećem sloju, [10].



Slika 8. Prikaz topologije višeslojne neuronske mreže bez povratnih veza
Izvor: [6]

U mrežama s povratnom vezom signal ne putuje kroz mrežom prolazeći samo jedanput kroz pojedini neuron već kroz neke dijelove mreže prolazi više puta. U takvim mrežama postoji

poveznica između barem jednoga izlaz neurona u $n+1$ sloju na ulaz n -tog sloja. Dodavanjem povratnih veza u neuronsku mrežu povećava se njena složenost te je teže napraviti analizu rada takve mreže. Osim što se povećava složenost povećava se i kvaliteta mreže pa u kombinaciji s elementima za kašnjenje dobivaju se nelinearni dinamički sustavi koji poboljšavaju sposobnost memoriranja u mreži.

Ljestvičaste mreže se sastoje od 1D, 2D ili višedimenzionalnog polja neurona te skupa čvorova koji daju ulazne signale. Svaki ulaz ima vezu prema svakom od neurona koji se nalaze u polju, [10].

3.1.4. Učenje neuronske mreže

Cilj učenja neuronske mreže je da za svaki ulazni vektor na svojem izlazu izađe odgovarajući ciljni vektor. Pojednostavljeno rečeno ako se mreži kao ulaz preda slika jabuke željeni rezultat je da mreža uistinu zaključi da je to jabuka. Konkretno promjene koje se dešavaju u mreži, a uslijed postupka učenja, jesu prilagodbe težina spojnica neurona. Teoretski gledano, nakon što se mreži prikaže niz ulazno izlaznih parova te mreža za svaki sljedeći ulazni vektor na svojem izlazu da odgovarajući ciljni vektor smatra se da je mreža naučena. U praksi, kada se govori o složenim nelinearnim sustavima, je gotovo nemoguće da izlazni vektor bude jednak ciljnom vektoru. Pa se učenje umjetne neuronske mreže zapravo svodi na postizanje što bližeg izlaza ciljnom vektoru to jest dobiti što manju pogrešku. Skup podataka namijenjen učenju mreže često se dijele na više skupova to jest na: skup za učenje, skup za treniranje i skup za validaciju. Skup za učenje čini najveći dio dok su skup za treniranje i validaciju manji, npr. podaci za učenje se dijele u omjeru 70% za skup za učenje, 15% skup za treniranje i 15% skup za validaciju. Prvi i najveći skup koristi se za učenje u užem smislu to jest za podešavanje težina između neurona. Pomoću drugog skupa algoritam provjerava trenutne performanse mreže te ukoliko se primijeti degradacija performansi treniranje prestaje i uzimaju se vrijednosti koje su dale najbolji rezultat. Treći skup se koristi kao provjera nakon što je mreža prošla postupak učenja. Također, važno je napomenuti da se mreža može i previše istrenirati odnosno mreža postaje ekspert za podatke u skupu za treniranje ali za bilo koje druge daje loše rezultate, [9].

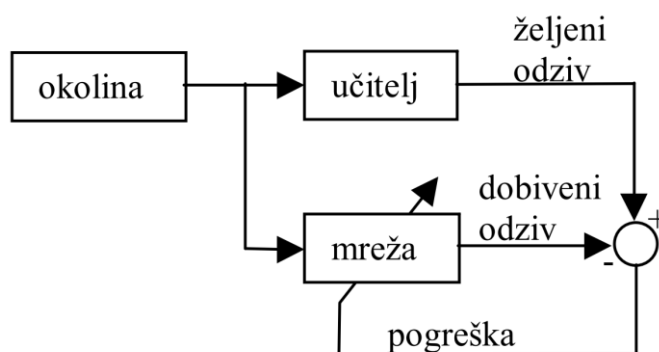
Pojmove koje je važno definirati kada se govori o učenju jesu iteracija i epoha. Iteracija predstavlja korak u algoritmu postupka za učenje u kojem se odvija promjena težina, a epoha

se odnosi na jedno predstavljanje cjelokupnog skupa za učenje. Ovisno o tome koliko se primjera mreži predoči u jednoj iteraciji razlikuju se, [9]:

- **Pojedinačno učenje (engl. *On-line training*)** – u jednoj iteraciji mreži se pokazuje samo jedan primjer to jest za svaki primjer zasebno se podešavaju težine;
- **Grupno učenje (engl. *Batch training*)** – u jednoj iteraciji mreži se prikazuje cijeli skup za učenje pa se epohe podudaraju s iteracijama.

Kada se govori o učenju neuronske mreže razlikuju se algoritmi učenja i paradigme učenja. Algoritam učenja predstavlja način odnosno skup pravila kojima se izračunavaju nove vrijednosti težina na spojnicama neurona. Paradigme učenja jesu zapravo odnos neuronske mreže i njene okoline to jest na koji način okolina utječe na učenje neuronske mreže. Metode učenja mogu se podijeliti prema, [10]:

- Paradigmi strojnog učenja:
 - **Učenje pod nadzorom** – osnovni princip nadziranog učenja je da se mreža uči prikazivanjem serija ulazno izlaznih parova, može se reći da mrežu nadzire učitelj i pokazuje joj ispravne odgovore. Kao povratna informacija mreži se u obliku razlike ciljnog i izlaznog vektora vraća pogreška. Na slici 9 prikazana je shema učenja pod nadzorom.;
 - **Učenje podrškom** – u ovome obliku učenja također postoji učitelj, ali za razliku od nadziranog učenja ne prikazuje mreži ciljne vrijednosti i samim time ne postoji funkcija pogreške. Mreža kao povratnu informaciju dobiva ocjenu koja je skalarna vrijednost. Na taj način se omogućuje da mreža nauči odnosno bude sprema i na pojave koje nisu u skupu za učenje;
 - **Učenje bez nadzora** – u ovoj paradigmi učitelja nema, odnosno mreža uči bez poznavanja izlaza.
- Algoritmu učenja:
 - Učenje korekcijom pogreške;
 - Kompetitivno učenje;
 - Hebbovo učenje;
 - Thorndikeovo učenje;
 - Boltzmannovo učenje;
 - Povratno rasprostiranje pogreške (engl. *Backpropagation*).



Slika 9. Shema učenja pod nadzorom
Izvor: [10]

Kod učenja korekcijom pogreške cilj je smanjiti funkciju pogreške. Pogreška se računa kao razlika između izlaznog i ciljnog vektora. Kao funkcija pogreške najčešće se koristi srednja kvadratna pogreška. S obzirom na to da se mreži daju ciljne vrijednosti na uvid radi se o paradigmi učenja pod nadzorom. Kompetitivno učenje je učenje bez nadzora. Neuroni se natječu za priliku da se aktiviraju, samo jedan neuron može biti aktiviran u isto vrijeme, [10].

Algoritam povratnog rasprostiranja pogreške je najčešće korištena metoda učenja višeslojne neuronske mreže. Kako bi mreža bila u mogućnosti obrađivati nelinearne skupove potrebno je da i njeni mrežni elementi imaju nelinearne prijenosne funkcije. Kao što je ranije navedeno to omogućava sigmoidalna funkcija. Konkretno, nelinearna funkcija potrebna je radi primjene gradijentne metode koja zahtijeva derivabilnu funkciju težinskih faktora. Algoritam povratnog rasprostiranja pogreške koristi metodu gradijentnog spusta kako bi minimizirao pogrešku. Cijelo učenje svodi se na pretraživanje u n -dimenzionalnom prostoru hipoteza, gdje n predstavlja ukupan broj težinskih faktora u mreži. kada se pogreška u takvom prostoru vizualizira ona čini hiper-površinu, a ona može sadržavati više lokalnih minimuma. To stvara mogućnost da algoritam prilikom učenja zaglavi u lokalnom minimumu. Ako algoritam zaglavi ne može postići bolji rezultat i učenje završava. Postoje tehnike kojima se pokušava izbjeći ta situacija. Jedna od tih tehnika je uključivanje momenta inercije. Unatoč tome u praksi se pokazalo da algoritam daje vrlo dobre rezultate. Kao parametre algoritam uzima skup za učenje, stopu učenja, broj čvorova ulaznog sloja, broj čvorova izlaznog sloja i broj čvorova skrivenog sloja. Algoritam povratnog rasprostiranja spada u paradigmu učenja pod nadzorom pa se mreži predočavaju primjeru u obliku parova ulaznih i ciljnih vektora.

Prvo se postavljaju inicijalne težine kao male nasumične vrijednosti. Nakon toga ponavlja se petlja u kojoj se mreži predstavljaju svi primjeri dokle god nije zadovoljen uvjet

zaustavljanja. Uvjeta za zaustavljanje učenja može biti više. To može biti maksimalni dozvoljeni iznos pogreške, broj iteracija ili epoha i slično. Ključno je pravovremeno zaustaviti učenje jer se može dogoditi da se mreža neće naučiti dovoljno ili da će se previše naučiti u oba slučaja mreža prilikom obrade podataka neće davati dobre rezultate.

Algoritam za svaki ulazno izlazni par računa izlaz tako što signali prolaze od ulaza prema izlazima, a računaju se i izlazi svakog pojedinog čvora. Na temelju razlike između izlazne i ciljne vrijednosti računa se pogreška. Ako se pretpostavi da je n neuron u skrivenom sloju mreže koja se uči algoritmom povratnog rasprostiranja pogreške. Pogreška od n računa se kao zbroj svih pogrešaka neurona na koje n utječe uz množenje težinskim faktorom. Težinski faktor govori koliko je neuron n utjecao na pogrešku nekog neurona koji slijedi nakon njega. Pogrešku pojedinog neurona moguće je zapisati kao:

$$\delta_n = o_n(1 - o_n) \sum_{s \in \text{Downstream}(n)} \omega_{ns} \delta_s, \quad (4)$$

gdje je:

δ_n – pogreška na izlazu čvora n ;

o_n – izlaz čvora n ;

ω_{ns} – težina s čvora n na čvor s ;

δ_s – pogreška na čvoru s ;

$s \in \text{Downstream}(n)$ – označava da je s „nizvodno“ od n to jest da je s čvor na koji n utječe.

Kako algoritam računa pogrešku svakog neurona tako ju rasprostire od izlaza prema ulazu i od tuda naziv algoritma. Prema pogreškama se postepeno prilagođuju težine između neurona i mreža korak po korak uči, [9].

3.2. Primjena neuronskih mreža u telekomunikacijskom prometu

Količina prometa koja protječe kroz Internet mrežu se znatno povećala u posljednjem desetljeću kao rezultat napretka tehnologije, industrije i aplikacija. Samo promet mobilnim Internet mrežama je porastao za 70% između 2016. i 2017. godine. IoT (engl. *Internet of Things*) – Internet stvari, računalstvo u oblaku i aplikacije koje se koriste podatkovnim centrima

se sve više koriste, a time i promet koji one generiraju. Mreže se suočavaju sa sve većim prometnim zahtjevima i moraju osigurati odgovarajući kvalitetu usluznosti QoS (engl. *Quality of Service*). Stoga je izrazito važno postojeće mrežne resurse iskoristiti što je bolje moguće. Izgradnja nove infrastrukture vrlo je skupa i često neisplativa stoga se traže načini kako iskoristiti postojeću na što efektivniji način. Točno predviđanje količine prometa pomaže u efikasnom iskorištavanju postojeće infrastrukture.

Jedna od primjena predviđanja prometa je štednja energije. Točnije, predviđanje prometa koristi se u jezgrenim usmjerivačima te se time smanjuje potrošnja energije. Kao što je ranije rečeno količina prometa koja se odvija mrežom sve je veća. Sav promet koji prolazi mrežom potrebno je i obraditi odnosno usmjeriti, provući kroz razne protokole i *gateway-e*, za sve to je potrebna procesna snaga koja će te podatke i obraditi. Snažni procesori sami troše velike količine energije i zahtijevaju snažnije rashladne sisteme koji opet troše puno energije. Ako se unaprijed poznaje količina prometa u pojedinom trenutku moguće je višak procesora ugaziti i tako uštediti na potrošnji energije. Osim što se time smanjuju troškovi i negativni ekološki utjecaj jedna od posljedica je poboljšanje QoS-a u energetski osjetljivim mrežama. Mreže Interneta stvari su primjer takvih mreža, a sve se češće koriste. Velik broj senzora koji rade na vlastitom napajanju imaju velike koristi ako znaju kada će biti potrebno slati podatke, a kada se mogu isključiti u stanje pripravnosti, [11].

Prema [15] određene karakteristike se navode kao prednosti i područja primjene neuronskih mreža u sklopu telekomunikacijske mreže, a prikazane su u tablici 1.

Tablica 1. Prednosti neuronskih mreža i područje njihove primjene u okviru telekomunikacija

| Prednosti | Područje primjene |
|---|---|
| <ul style="list-style-type: none"> ➤ Mogućnost rješavanja podatkovno zahtjevnih problema; ➤ Vrlo brza izrada prototipa; ➤ Mogućnost prilagodbe i učenje; ➤ Skalabilnost; ➤ Nelinearnost. | <p>Tamo gdje konvencionalni procesi nisu povoljni, nije ih moguće definirati ili nisu u mogućnosti u potpunosti uhvatiti kompleksnost podataka.</p> <p>Kada je stohastičko ponašanje važno, a nije potrebno poznavati objašnjenje odluke mreže.</p> |

Izvor: [15]

Osim prethodno navedenih prednosti koje pružaju neuronske mreže važno je i spomenuti mane neuronskih mreža. Iako pružaju rješenje za mnoštvo složenih problema koji se ne mogu riješiti konvencionalnim metodama predstavljaju izrazito složene sustave, a time se pojavljuju određeni problemi. Neuronska mreža je crna kutija, a to znači da nije moguće objasniti njene odluke. Kada neuronska mreža nauči neki složeni proces to ne omogućava da se taj proces i razumije već je samo moguće na određene podražaje dobiti odgovor sustava iako on skoro nikada nije u potpunosti točan. Često se kao važno svojstvo neuronskih mreža spominje mogućnost obrade velikih količina podataka. U današnje doba računala sustavi generiraju velike količine podataka sve te podatke potrebno je i pohraniti. Često je potrebno podatke i obraditi kako bi se mogli predati neuronskoj mreži što pak zahtjeva dodatnu procesorsku moć. Kod kreiranja strukture neuronske mreže ne postoji točno određeni postupak. Za pojedini problem jedini način za pronaći odgovarajuću strukturu neuronske mreže je testiranjem. Mreža kreirana za određeni problem će davati dobre rezultate samo za taj problem, nije moguće prenamijeniti mrežu. Kao i kod kreiranja strukture neuronske mreže odabir metode učenja nije određen već je potrebno ispitati koja od metoda daje najbolje rezultate. Praksa je pokazala da kreiranje funkcionalnog sustava primjenjivog u realnom okruženju na temelju prototipa je podosta težak postupak, [16].

U [17] istražuje se primjena neuronskih mreža u sklopu bežičnih mreža. Kao način uporabe neuronskih mreža u bežičnim komunikacijama navode se:

- Bežična komunikacija s bespilotnim letjelicama;
- Bežične mreže za virtualnu stvarnost;
- Rubno predmemoriranje i računanje za mobilne mreže (engl. *mobile edge caching and computing*);
- Pristupne tehnologije za radio mreže;
- Internet stvari.

Spominju se i dva općenita načina primjene neuronskih mreža:

1. u svrhu predviđanja, zaključivanja i analize velikih skupova podataka (engl. *big data*);
2. korištenje u svrhu rada samoorganizirajućih mreža.

Primjena umjetne inteligencije u obliku neuronskih mreža bežičnim sustavima omogućuje učenje na temelju korisnikova ponašanja, okruženja i povezanih uređaja. Primjerice neuronske mreže mogu naučiti uzorke kretanja korisnika i sadržaje koje korisnici preuzimaju te time omogućiti baznim postajama pravovremeno predmemoriranje traženih sadržaja. Ovime se

postiže smanjenje opterećenje mreže i bolja kvaliteta usluge. Ovakav način rada mreže je temelj koncepta proaktivnog bežičnog umrežavanja, gdje mreže predviđaju i tako se prilagođavaju ponašanju korisnika. Također, ova tehnologija omogućiti će IoT i paradigme pametnih gradova. Gdje će se masivne količine podataka prikupljene iz senzora koristiti za optimizaciju korištenja resursa, razumijevanje rada mreže, nadgledanje kvarova i pogrešaka te pružanje pametnih usluga, [17].

U istraživanju [18] istražuje se primjena umjetnih neuronskih mreža u prevenciji tzv. čovjek u sredini (engl. *man in the middle*) napada na MANET (engl. *mobile ad hoc network*) mreže. MANET mreže omogućuju povezivanje različitih mobilnih uređaja, koji imaju tu mogućnost, a ne zahtijevaju izgrađenu infrastrukturu odnosno uređaji koji koriste mrežu su ujedno i čvorovi. Problem kod takvih mreža, a zapravo i njihova prednost, je raznolikost uređaja i aplikacija koji pristupaju i šalju podatke kroz mrežu. U navedenom istraživanju se predstavlja rješenje za navedene napade koristeći AODV protokol i umjetne neuronske mreže, [18].

4. Određivanje strukture neuronske mreže

Matematički model neuronske mreže potrebno je realizirati u programskom kodu. Za izradu praktičnog djela ovog seminara korišten je programski jezik MATLAB. Kratica MATLAB proizlazi iz MATrix LABoratory odnosno prevedeno na hrvatski jezik znači matrični laboratorij. Njegovo ime je zapravo i glavna osobina to jest MATLAB je okruženje specijalizirano za matrice, gotovo sve varijable zapisuju se u matričnom obliku. Više o MATLAB-u biti će napisano u sljedećem potpoglavlju.

Za svaki skup podataka koji se obrađuje neuronskim mrežama potrebno je definirati odgovarajuću strukturu neuronske mreže. Ne postoje točno definirana pravila prema kojima se određuje najpovoljnija struktura, već je potrebno testiranjem pronaći strukturu koja će dati optimalne rezultate.

4.1. MATLAB programski paket

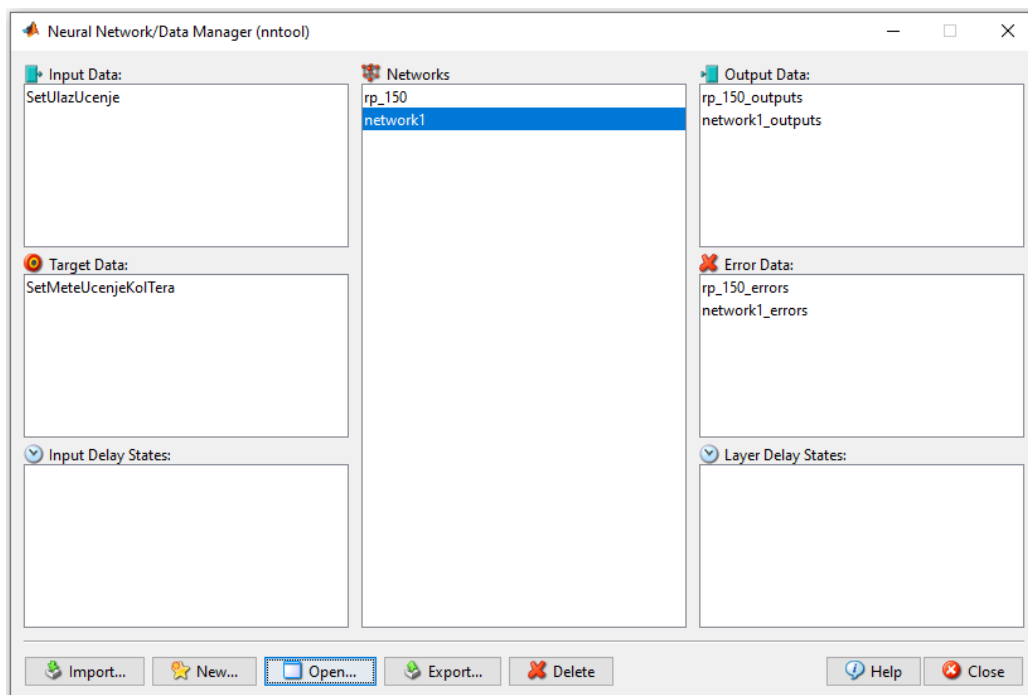
MATLAB (engl. *MATrix LABoratory*) je matrično orijentirani programski jezik što ga čini vrlo povoljnim za izražavanje matematičkih izraza i za rad s istim. Ugrađena grafika omogućuje jednostavno iscrtavanje funkcija, podataka i varijabli. Rad s velikim skupovima podataka jednostavan je zahvaljujući redovima i matricama. MATLAB pruža pregršt funkcija namijenjenim različitim granama znanosti. Sve funkcije vezane za određenu svrhu skupljene su u paket pod nazivom *Toolbox*. Neki paketi zahtijevaju instalaciju drugih paketa čije su im funkcije potrebne za rad. Također, korisnici mogu kreirati i vlastite pakete te ih dijeliti s drugim korisnicima. Neki od značajnijih paketa u okviru ovoga rada jesu:

- *Deep Learning Toolbox*;
- *Statistics and Machine Learning Toolbox*;
- *Curve Fitting Toolbox*;
- *Financial Toolbox*.

Osim samog programskog jezika koji pruža mnoštvo mogućnosti vrlo važan dio MATLAB programskog paketa je Simulink. On predstavlja grafički alat koji koristi matematičku ljusku MATLAB-a za simulaciju sustava. Važno svojstvo je što se izrada simulacijskih modela na vrlo

jednostavan način obavlja koristeći grafičke blokove. Osim postojeće biblioteke gotovih blokova moguće je izraditi vlastite blokove koristeći MATLAB-ove m-funkcije ili funkcije napisane u programskom jeziku C/C++. Sheme kreirane u Simulinku sastoje se od blokova i linija odnosno signala kojima se povezuju pojedini blokovi i time realiziraju jednadžbe koje opisuju sustav, [14].

U sklopu *Deep Learning Toolbox-a* postoje funkcije kojima se mogu jednostavno kreirati, trenirati i simulirati neuronske mreže. Neuronsku mrežu moguće je kreirati ručno odnosno koristeći naredbe u naredbenom prozoru ili kroz grafičko sučelje. Na slici 10 prikazan je početni prozor alata za neuronske mreže koje se pokreće naredbom „nntool“. Ovo je zapravo prozor za upravljanje podacima. Kroz njega je moguće učitati podatke koji će se koristiti za učenje i simuliranje neuronske mreže. Također, služi za izvoz rezultata i ostalih podataka koji rezultiraju simuliranjem mreže.

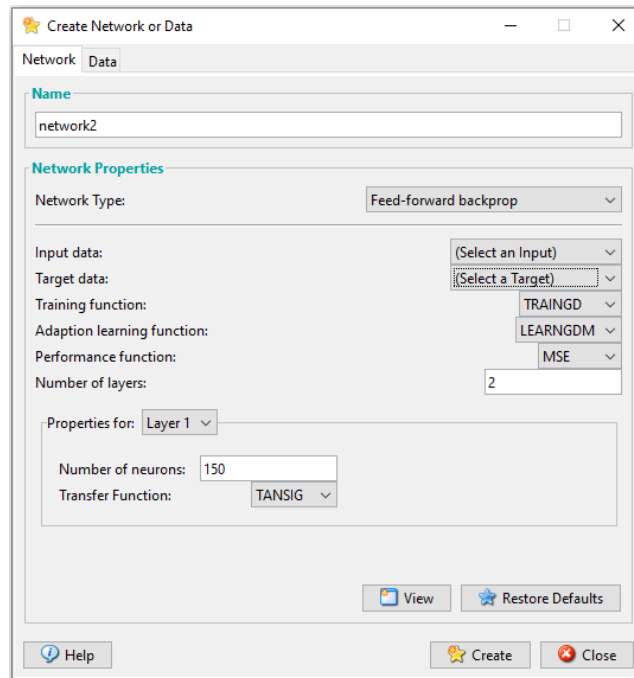


Slika 10. Prikaz prozora za upravljanje podacima ujedno i početni prozor alata za neuronske mreže

Izvor: izradio autor

Raspored prozora je posložen tako da se s lijeve strane nalaze podaci koji se predaju mreži (ulazi u mrežu i ciljni podaci koji će služiti učenju ili izračunu pogreške), u sredini se nalaze kreirane neuronske mreže i s desne strane izlazni podaci (izlaz neuronske mreže i pogreške). Podatke je moguće uvesti iz radnog prostora MATLAB-a ili prethodno pohranjene m datoteke. Isto tako rezultati se mogu izvesti u radni prostor ili m datoteku koju je kasnije moguće

jednostavno učitati. Na slici 11 prikazano je sučelje za kreiranje neuronske mreže koje se otvara klikom na gumb „New...“.



Slika 11. Sučelje za kreiranje neuronske mreže
Izvor: izradio autor

U ovom prozoru odabire se:

- Vrsta mreže;
- Ulazni podaci;
- Ciljne vrijednosti;
- Algoritam učenja;
- Funkciju prilagodbe;
- Funkciju pogreške;
- Broj slojeva;
- Broj neurona i prijenosnu funkciju za svaki skriveni sloj.

Kada se kreira mreža sa željenom arhitekturom moguće je na jednostavan način odabrati uvezene podatke za treniranje, a potom i definirati parametre mreže. Ovisno o algoritmu treniranja mogu se ugađati različiti parametri, a značajniji parametri jesu:

- Broj epoha/iteracija – kada prođe kroz definirani broj epoha učenje se prekida;
- Maksimalno vrijeme učenja – vremenska granica učenja (zadana i najčešća vrijednost je beskonačno);

- Ciljna vrijednost pogreške – kada pogreška dosegne ovu vrijednost prestaje učenje (zadana vrijednost je 0);
- Maksimalni broj pogrešaka – ovaj parametar služi kako se mreža ne bi previše naučila, a funkcionira tako da se učenje zaustavlja kada se performanse na skupu za validaciju ne poboljšavaju zadani broj epoha.

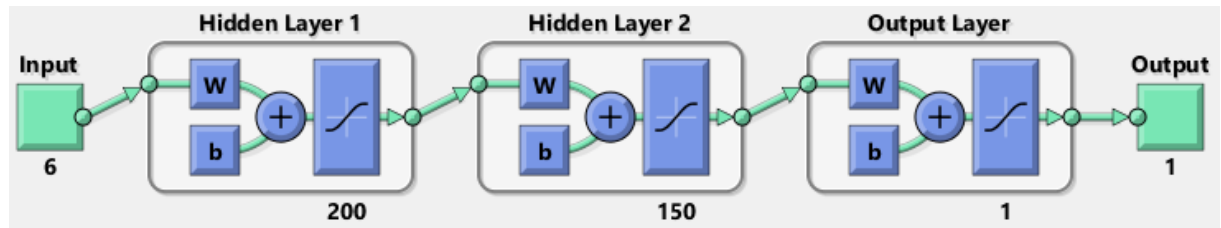
Tijekom procesa učenja postoji mogućnost iscrtavanja pogreške, regresije i stanja treniranja u stvarnom vremenu.

Osim alata za neuronske mreže postoji i niz alata koji pružaju sučelje pomoću kojih se kreiraju neuronske mreže za specificirane probleme. Osnovna razlika je što koristeći alata za neuronske mreže moguće definirati sve parametre mreže, a koristeći specificirane alate korisnika se vodi kroz sučelje i tako da za vlastitu mrežu može odabrati dio parametara. Naredbom „nstart“ otvara se početno sučelje gdje se odabire željeni alat. Moguće je i svaki alat pozvati zasebnom funkcijom. Alat koji je relevantan za ovaj rad je alat za vremenske serije. Ovim alatom se na vrlo jednostavan način mogu učitati podaci, odrediti raspodjela podataka, količina neurona i dobiti rezultate. Nisu potrebne ulazne serije već se samo odaberu ciljne vrijednosti, a dalje sustav sam sve odrađuje.

4.2. Definiranje strukture neuronske mreže

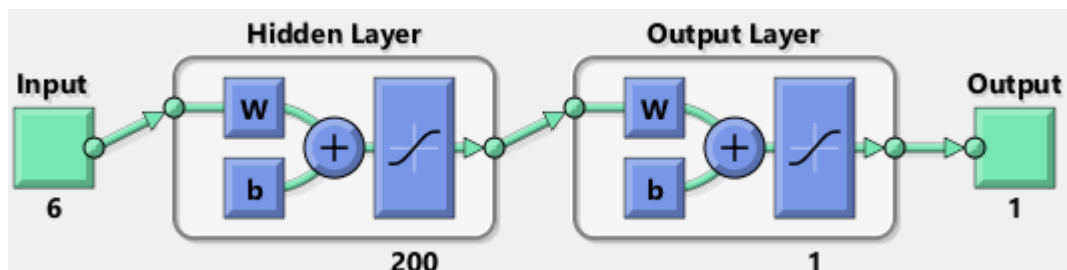
Struktura mreže određivana je vodeći se Kolmogorovim teoremom. Prema navedenom teoremu postoji troslojna neuronska mreža s m neurona u ulaznom, n neurona na izlaznom i $2m+1$ neurona u unutarnjem sloju mreže koja ostvaruje navedeno preslikavanje, to jest rješava nelinearno odvojive zadaće. Jednostavnija mreža ne može izvesti navedeno preslikavanje. Dakle cilj je pronaći najjednostavniju mrežu koja daje što bolje rezultate. Kao što je već rečeno učenje neuronske mreže je ugađanje težina na temelju prikazanih ulazno izlaznih parova. Za predviđanje količine prometa u ovome radu koristile su se jednoslojna i višeslojna aciklička mreža. Aciklička mreža je ona koja nema povratne veze pa svi signali prolaze kroz mrežu do izlaza odnosno propagacija je jednosmjerna. Također, ovakve mreže imaju ulazni sloj, skriveni sloj i izlazni sloj, a karakteristično je što u ulaznom sloju nema neurona nego se zapravo radi o ulaznim vrijednostima. Takve mreže se još nazivaju i unaprijedne (engl. *feedforward*) neuronske mreže, [9].

Dakle, cilj je bio pronaći što jednostavniju mrežu a da daje prihvatljive rezultate. Za svaki skup podataka kreirane su dvije mreže jedna sa 2 skrivena sloja, a druga sa 1 skrivenim slojem. Ispitano je više mreža a odabrane su one s najboljim rezultatima to jest s najmanjom pogreškom i najvećim koeficijentom korelacije. S obzirom na karakteristike podatkovnog skupa korištenog za učenje mreže bilo je potrebno koristiti nešto složenije mreže sa većim brojem neurona. za oba skupa podataka korištene su iste mreže, ali su učene na različitim primjerima zbog različitosti podataka.



Slika 12. Prikaz strukture neuronske mreže s dva skrivena sloja
Izvor: izradio autor

Na slici 12 prikazana je arhitektura neuronske mreže s dva skrivena sloja. Mreža se sastoji od ulaznog sloja koji prima 6 argumenata, prvog skrivenog sloja koji sadrži 200 neurona, drugog skrivenog sloja koji sadrži 150 neurona i izlaznog sloja. Neuroni u svim slojevima mreže za aktivacijsku funkciju koriste hiperboličko-tangentnu sigmoidnu funkciju. Ona je jedna od često korištenih jer dobro preslikava nelinearne probleme kao što je i predmet ovoga rada. Testiranjem se pokazalo da najbolje rezultate daje mreža u kojoj svi neuroni imaju navedenu aktivacijsku funkciju. Druga korištena mreža sa jednim skrivenim slojem prikazana je na slici 13 Glavna razlika je što ova mreža ima samo jedan skriveni sloj sa 200 neurona. Pokazalo se da obje mreže daju slične rezultate, ali jednostavnija mreža dala bolje rezultate prilikom učenja, a složenija kod predviđanja.



Slika 13. Prikaz arhitekture jednostavnije neuronske mreže s dva sloja
Izvor: izradio autor

5. Podatkovni skup korišten za učenje neuronske mreže

Svrha ovoga rada je primijeniti neuronske mreže kao alat za predviđanje telekomunikacijskog prometa. Kako bi neuronska mreža mogla predvidjeti količinu prometa potrebni su joj podaci o količini telekomunikacijskog prometa iz prošlosti. Naravno, kada se neuronska mreža uči na temelju podataka iz pojedine telekomunikacijske mreže ona će moći predviđati podatke samo za tu mrežu, naučiti će njene karakteristike.

5.1. Opis podatkovnog skupa

Podatkovni skup korišten u ovome radu je dobiven je na korištenje u svrhu izrade rada od Hrvatske akademske i istraživačke mreže - CARNET (engl. *Croatian Academic and Research Network*). CARNET je javna ustanova koja djeluje u sklopu Ministarstva znanosti i obrazovanja u području informacijsko-komunikacijskih tehnologija i primjene istih u obrazovanju. CARNET mreža povezuje 2.600 ustanova na 3.800 lokacija širom Hrvatske od kojih su većina znanstvene i istraživačke zajednice Republike Hrvatske te institucije u sklopu osnovnog i srednjoškolskog obrazovnog sustava. Podatkovni skup predstavlja prosječne brzine na poveznici prema GEANT (engl. *Multi-Gigabit European Academic Network*) mreži. GEANT mreža je Europska akademska mreža namijenjena znanstvenoj izvrsnosti, istraživanjima, edukaciji i inovaciji. Povezana je s 39 nacionalnih istraživačkih i obrazovnih mreža (NREN, engl. *National Research and Education Network*), a jedna od NREN-a je i CARNET. Jezgrena mreža radi na brzinama do 500Gbps te doseže preko 100 nacionalnih mreža širom svijeta, [12], [13].

Ustupljeni su podaci dolazne i odlaznog linka između CARNET i GEANT mreže. Prvih nekoliko redaka iz skupa podataka prikazano je na slici 14 S lijeva na desno nalazi se naziv podatka, vrijeme kada je podatak spremljen i na kraju vrijednost mjerenog parametra u tom trenutku. S obzirom na to da su dobiveni podaci za mjerenja prosječnih brzina ostvarenih u petominutnom razdoblju mjerna jedinica je bajt po sekundi odnosno skraćeno B/s. Vremenske značke su u *Unix Timestamp* formatu. Taj format zapisa vremena koji se često koristi u računalnim mrežama zbog jednostavnog zapisa se zasniva na broju proteklih sekundi od 1. siječnja 1970. godine.

```

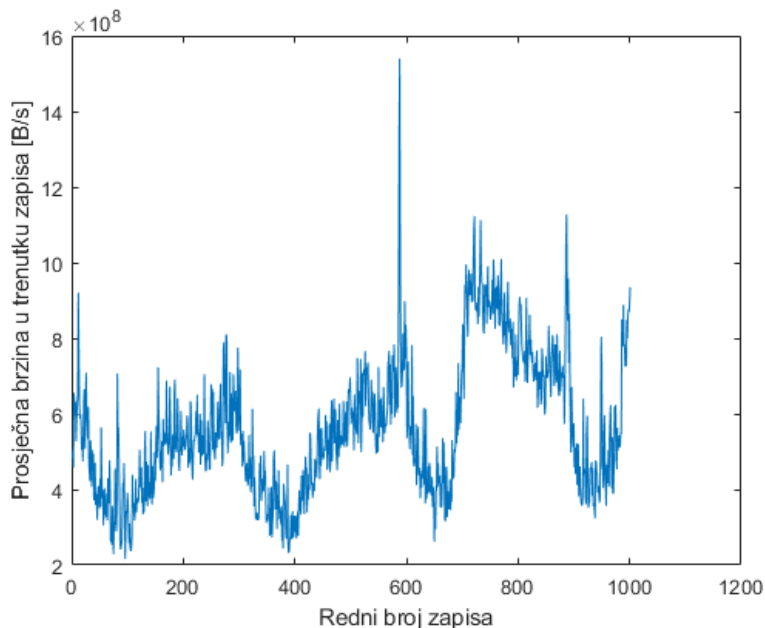
inteOctInOut,1561452274000000000,879181584.72
inteOctInOut,1561452574000000000,873842360.96
inteOctInOut,1561452874000000000,894202305.12
inteOctInOut,1561453128000000000,995208142.4566929
inteOctInOut,1561453428000000000,1016955500.3466667
inteOctInOut,1561453768000000000,1017636861.482353
inteOctInOut,1561454068000000000,868167221.6
inteOctInOut,1561454368000000000,932698048.24
inteOctInOut,1561454668000000000,1019260017.2266667
inteOctInOut,1561454968000000000,870097742.6666666
inteOctInOut,1561455268000000000,562715164.4533334
inteOctInOut,1561455568000000000,482469170.4
inteOctInOut,1561455868000000000,542950664
inteOctInOut,1561456168000000000,526852069.6
inteOctInOut,1561456468000000000,510816491.06666666
inteOctInOut,1561456768000000000,540129025.84
inteOctInOut,1561457068000000000,556043630.6133333

```

Slika 14. Prikaz sirovih podataka

Izvor: izradio autor

Podaci obuhvaćaju period od približno jedne godine, točnije od 25. lipnja 2019. do 24. lipnja 2020. godine što čini skup od 104784 zapisa. Na slici 15 prikazan je grafički prikaz sirovih podataka, izdvojen je skup od 1000 zapisa radi boljeg prikaza. Moguće je primijetiti da postoje jasni uzorci u podacima. Također, oko 600. zapisa moguće je primijetiti fluktuaciju odnosno vrijednost koja znatno odskače od ostalih vrijednosti. Takvih fluktuacija ima velik broj te postoji i nekoliko vrijednosti koje odskaču od okolnih vrijednosti i za dvije decimale, takve vrijednosti su uklonjene kako bi se odmah eliminirao dio pogreške u ulaznim podacima.



Slika 15. Grafički prikaz sirovih podataka

Izvor: izradio autor

5.2. Učenje neuronske mreže

Kao što je već objašnjeno učenje neuronske mreže je prilagođavanje težina između neurona kako bi se na određeni podražaj ostvario odgovarajući izlaz. Vrlo često dio pogreške leži i u samim podacima pomoću kojih se uči mreža. Stoga je potrebno podatke prilagoditi i napraviti neke promjene koje će omogućiti mreži da lakše nauči. U sljedećim potpoglavljima biti će objašnjeno kako su podaci obrađeni, kako su prezentirani mreži i kako je mreža na temelju njih naučila karakteristike procesa.

5.2.1. Obrada podataka i priprema ulaznih vektora

Dobiveni podaci su u formatu zarezom odvojenih vrijednosti (CSV, engl. *Comma Separated Values*). Takav oblik povoljan je za daljnju obradu jer je vrijednosti vrlo lako učitati u bilo koji programski jezik. Za inicijalnu obradu podataka korišten je programski jezik Python. U prilogu 1 nalazi se kod napisan u Pythonu kojim je izvršena inicijalna obrada podataka. Rad koda moguće je opisati sljedećim koracima:

1. Uvoze se potrebni moduli: csv, datetime, holidays;
2. Otvaraju se pisari koji će nakon obrade potrebne podatke ispisati u tekstualnom formatu;
3. Kreira se lista u koju se smješta popis datuma koji su praznici u godini 2019. i 2020.
4. Otvara se csv datoteka s izvornim podacima te se kroz podatke prolazi redak po redak. Za svaki redak se čita vremenska značka te se u zasebne tekstualne datoteke zapisuju odgovarajući podaci o pojedinom zapisu (petominutni dio sata (1-12), polovica sata (1 ili 2) sat u danu, dan u tjednu, dan u mjesecu i blagdan). Za svaki redak to jest podatak se provjerava jeli uzet na dan kada je blagdan.
5. Zatvaraju se pisari.

Ulaz u mrežu moraju biti podaci koji opisuju izlaznu vrijednost, odnosno oni parametri koje ćemo znati i u budućnosti te ćemo ih moći predati mreži kako bi dobili izlaz koji će predstavljati predviđenu vrijednost. Skup podataka čine zapisi u razdoblju od jedne godine. Kada mreži pokazujemo primjere to jest ulazno izlazne parove uobičajeno vrijedi da što više primjera istovrsnih podataka ima to će mreža bolje naučiti. Pojam istovrsnih primjera najbolje je

objasniti primjerom, pa ako se želi postići da mreža predvidi količinu prometa za neki ponedjeljak u 12 sati mreži je potrebno prikazati što više ponedjeljaka s popratnim meta podacima. Odnosno mreži ćemo predati o kojem ponedjeljku u mjesecu se radi, jeli pojedini ponedjeljak blagdan i slične parametre koji su nam poznati a mogu opisati specifični dan.

Kao ulaz u mrežu koristile su se klasifikacije prema:

- Dijelu sata – ako se govori o petominutnom koraku koji od 12 dijelova je pojedini zapis isto vrijedi i za polusatni korak samo što su 2 dijela za odrediti;
- Satu u kojemu je podatak zabilježen;
- Dan u tjednu;
- Dan u mjesecu;
- Dali je podatak zabilježen na blagdan.

Koristeći tekstualne datoteke kreirane pomoću Python skripte podaci su uvezeni u MATLAB radni prostor i pomoću MATLAB funkcija pretvoreni u jednu matricu čiji stupci predstavljaju ulazne vektore. Na slici 16 prikazana je transponirana ulazna matrica radi lakšeg prikaza pa su ulazni vektori poredani u retke. Krajnji lijevi stupac predstavlja oznaku blagdana, drugi stupac predstavlja dan u mjesecu, u trećem stupcu nalazi se oznaka dana u tjednu (0 predstavlja ponedjeljak pa redom do 6 odnosno nedjelje), četvrti stupac označava sat u danu i posljednji stupac označava o kojoj polovici sata se radi.

| | | | | |
|---|----|---|----|---|
| 1 | 25 | 1 | 21 | 2 |
| 1 | 25 | 1 | 22 | 1 |
| 1 | 25 | 1 | 22 | 2 |
| 1 | 25 | 1 | 23 | 1 |
| 1 | 25 | 1 | 23 | 2 |
| 0 | 26 | 2 | 0 | 1 |
| 0 | 26 | 2 | 0 | 2 |
| 0 | 26 | 2 | 1 | 1 |
| 0 | 26 | 2 | 1 | 2 |
| 0 | 26 | 2 | 2 | 1 |
| 0 | 26 | 2 | 2 | 2 |
| 0 | 26 | 2 | 3 | 1 |
| 0 | 26 | 2 | 3 | 2 |
| 0 | 26 | 2 | 4 | 1 |
| 0 | 26 | 2 | 4 | 2 |
| 0 | 26 | 2 | 5 | 1 |
| 0 | 26 | 2 | 5 | 2 |
| 0 | 26 | 2 | 6 | 1 |
| 0 | 26 | 2 | 6 | 2 |

Slika 16. Prikaz jedne od ulaznih matrica za učenje mreže
Izvor: izradio autor

S obzirom na to da je tema seminara predviđanje količine prometa, a dobiveni su podaci o brzini potrebno je i njih pretvoriti u odgovarajući oblik. Mjerna jedinica sirovih podataka je B/s a mjerna jedinica u koju ih se želi pretvoriti je bit. Svaka vrijednost pomnožena je sa 8 kako bi se bajti pretvorili u bite, a zatim sa 300 jer su mjerene prosječne brzine u petominutnom razdoblju pa kako bi se dobila količina bitova prenesena u 5 minuta. Podaci na slici 16 predstavljaju ulaze za polusatni promet. Izlazni vektori su dobiveni tako da se računala prosječna vrijednost za svakih pola sata. To je između ostaloga učinjeno kako bi se smanjio šum u podacima.

S obzirom na karakteristike podataka bilo ih je potrebno razdvojiti na više skupova i napraviti dodatnu obradu. Osnovni problem je bio što su podaci šumoviti, to je moguće uočiti i na slici 13 gdje, iako je uzorak uočljiv, ima mnoštvo nečistoća to jest krivulja nije „glatka“. Kao što je već spomenuto u prethodnom odlomku dio tog šuma je otklonjen pretvaranjem podataka u satne korake. U MATLAB-u postoji funkcija koja računa pomični prosjek i njome se vrlo učinkovito smanjuje šum a zadržava karakteristika podataka. Svi skupovi podataka odlaznog prometa korišteni za učenje mreže prošli su kroz funkciju pomičnog prosjeka s korakom 15. Time se dodatno zagladila krivulja podataka i smanjile razlike uzastopnih zapisa. Također, iz podataka su uklonjeni zapisi s velikim odstupanjima jer su utjecali na učenje mreže, odnosno mreža je predviđala znatno više vrijednosti za pojedine ulaze. Odstupanja su bila za nekoliko redova veća od ostalih podataka. Zbog već navedene sezonalnosti podataka neke karakteristike se nisu pojavljivale u skupu za učenje. Kako bi mreža što uspješnije predviđela količinu prometa nužno joj je prilikom učenja pokazati sve karakteristične dijelove. Isto tako zbog postojanja istih bilo je potrebno mreži „reći“ da se radi o različitim dijelovima. Stoga je uveden šesta ulazna varijabla koja označava različita razdoblja.

5.2.2. Učenje neuronske mreže

Pojam učenja umjetne neuronske mreže već je definiran, no podesiti težine na sinapsama neurona tako da mreža preda zadovoljavajući izlaz nije jednostavno. Prilikom učenja mreže određuje se niz parametara koji će utjecati na učenje mreže i u konačnici na njene performanse. Za učenje mreže korištena je funkcija otpornog povratnog rasprostiranja pogreške (engl. *Resilient backpropagation*). Ova funkcija smatra se brzom metodom učenja neuronske mreže, a karakterizira ju manja osjetljivost na velike vrijednosti. Kod ostalih algoritama kada su u

pitanju sigmoidne aktivacijske funkcije može se dogoditi da na velike ulazne vrijednosti desi mala promjena težina što uzorkuje zaustavljanje učenja iako je mreža daleko od optimalnih težina za predstavljeni problem, [19]. Parametri koji se ugađaju za ovu metodu učenja, a ujedno i vrijednosti parametara korištene prilikom treniranja prikazane su na slici 17.

| | |
|----------|--------|
| delta0 | 0.0007 |
| delt_inc | 1.2 |
| delt_dec | 0.5 |
| deltamax | 50 |

Slika 17. Prikaz parametara koji se mogu ugađati kod metode otpornog rasprostiranja pogreške
Izvor: izradio autor

Parametri redosljedom kao i na slici jesu, [6]:

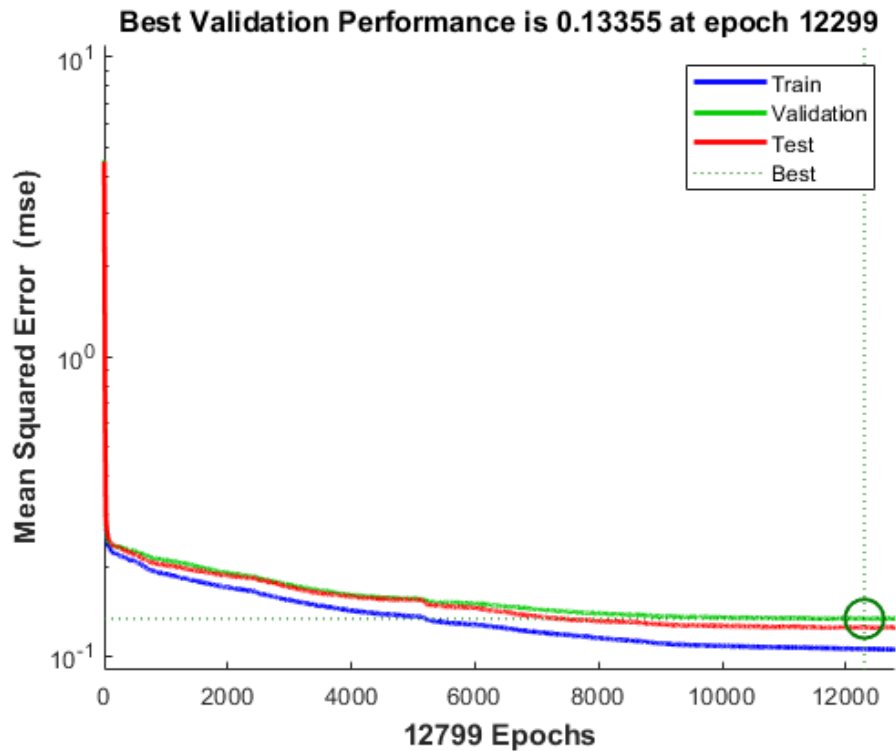
- Inicijalna vrijednost za promjenu težina;
- Povećanje za promjenu težina;
- Smanjenje za promjenu težina;
- Maksimalna promjena težina.

Pri učenju mreža ključnu ulogu igra postavljanje uvjeta zaustavljanja učenja. Isto kao i kod određivanja strukture mreže ne postoji pravilo kako namjestiti parametre već su oni specifični za svaki problem i skup podataka koji se prezentira mreži. Kao maksimalan broj iteracija postavljena je vrijednost 15.000 jer, a maksimalan broj pogrešaka na 500. Obje mreže su dosegle granicu od 500 pogrešaka te time završile svoj postupak učenja. Kao što je opisan taj parametara, to znači da mreža nije postigla svoj rezultat 500 uzastopnih iteracija. Na temelju skupova za validaciju i testiranje određuje se pogreška kojom se tumači uspješnost mreže. Najčešće greške koje se računaju jesu, [20]:

- Srednja kvadratna pogreška, MSE (engl. *Mean Square Error*) - računa se prosjek kvadrata pogreške odnosno razlika izlaznog i ciljnog vektora;
- Srednja apsolutna pogreška, MAE (engl. *Mean Absolute Error*) – veličina koja se upotrebljava kako bi se doznalo koliko je predviđanje blizu eventualnim rezultatima;
- Koeficijent korelacije – pokazuje koliko je jedna varijabla u ovisnosti promjene druge varijable.

U svrhu opisa kvalitete mreže u ovom radu koristila se srednja apsolutna pogreška (MAE) i koeficijent korelacije. Na slici 18 može se vidjeti srednja kvadratna greška tijekom učenja

neuronske mreže, radi se o jednoslojnoj mreži za predviđanje odlaznog prometa. Na x osi nalaze se epohe, a na y osi prikazana je pogreška u pojedinoj epohi. Učenje za navedenu mrežu prestalo je 12.299. epohi sa srednjom kvadratnom pogreškom od 0,13355. Također, na grafu postoje tri krivulje plava označava podatke za treniranje, zelena podatke za validaciju i crvena podatke za testiranje.



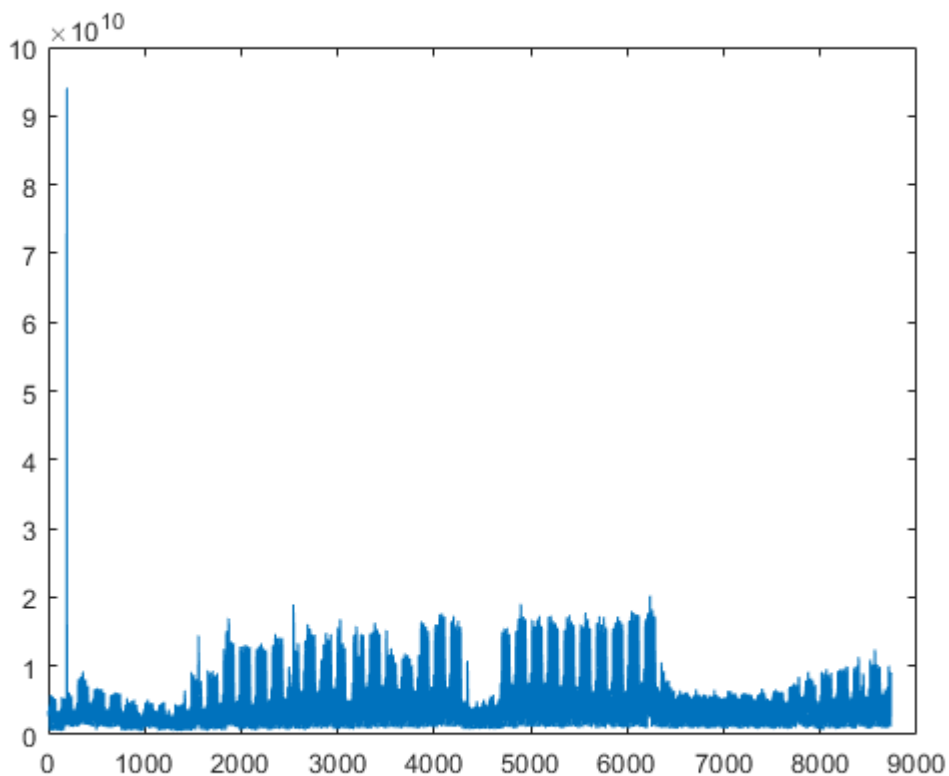
Slika 18. Prikaz pogreške mreže prilikom procesa učenja
Izvor: izradio autor

Od ukupnog skupa podataka 85% korišteno je za učenje mreže, a preostalih 15% kao provjera za točnost predviđanja. Time je neuronskoj mreži dana dovoljna količina podataka za učenje, a rezultati predviđanja su se mogli usporediti sa stvarnim vrijednostima.

6. Rezultati učenja i predviđanja

S obzirom na karakteristiku podataka može se reći da je neuronska mreža dobro naučila kretanje količine prometa na linkovima. Problem je što se u mreži pojavljuje sezonalnost u netipičnim periodima što se odrazilo i na rezultate neuronske mreže. S obzirom na to da je mreža učena na podacima perioda od jedne godine nije mogla naučiti sezonske fluktuacije koje karakteriziraju skup podataka. Specifične periode bilo je potrebno ručno unijeti odnosno predati mreži kako bi ih razlikovala.

Karakteristike prometa odlaznog i dolaznog prometa za dani set podataka su vrlo slične. Pojavljuje se isti trend no kod dolaznog linka se pojavljuju znatno veće vrijednosti. Stoga je mreže bilo potrebno učiti zasebno za svaki link. Na slici 19 prikazan je cijeli skup. Može se uočiti kako podaci imaju periode povećanog prometa i periode smanjenog prometa te upravo ta razdoblja predstavljaju problem mreži.



Slika 19. Prikaz svih podataka u satnoj rezoluciji
Izvor: izradio autor

Kao podatkovni skup za učenje korišten je polusatni period. Ulazni vektor sastojao se od 6 varijabli. Na slici 20 prikazana su dva ulazna vektora koji čine jedan sat. Ulazni vektor sastoji se od varijabli koje označavaju: blagdan, dan u mjesecu, dan u tjednu, sat u danu, polusatni dio i oznaku perioda sezonalnosti.

| | | | | | |
|---|---|---|---|---|---|
| 0 | 9 | 0 | 2 | 1 | 1 |
| 0 | 9 | 0 | 2 | 2 | 1 |

Slika 20. Prikaz dva ulazna vektora koji čine jedan sat
Izvor: izradio autor

Ulazni vektor na slici 20 govori da pojedini zapis nije blagdan, zapis je zabilježen 9. dan u mjesecu, a taj dan je ponedjeljak, prvi zapis je zabilježen u 2 sata ujutro dok je drugi u 2:30 sati ujutro, također radi se o 2. periodu sezonalnosti. Podaci o danu u tjednu započinju s 0 što označava ponedjeljak, a 6 nedjelju. Skup je podijeljen na tri različita perioda sezonalnosti, a oznake su od 0 do 2.

6.1. Analiza rezultata za predviđanje odlaznog prometa

U svrhu predviđanja količine prometa kreirane su dvije arhitekture neuronske mreže, a ispitano ih je mnogo više. Odabrane su dvije različite s najboljim rezultatima, iako su u većini slučajeva krajnji rezultati bili slični.

Tablica 2. Prikaz rezultata učenja neuronskih mreža korištenih za predviđanje odlazne količine prometa

| Broj skrivenih slojeva (broj neurona u pojedinom sloju) | Broj uzoraka za treniranje | Srednja kvadratna pogreška (MSE) | Koeficijent korelacije |
|---|----------------------------|----------------------------------|------------------------|
| 1 (200) | 14.843 | 0,0424 | 0,96 |
| 2 (200, 150) | 14.843 | 0,0626 | 0,94 |

Izvor: izradio autor

U tablici 2 prikazane su karakteristike mreže nakon učenja. Moguće je uočiti da je jednostavnija mreža dala malo bolje rezultate. No ovi podaci su vezani za skup na kojemu se

mreža učila. Nakon simulacije mreža pokazalo se da mreža sa složenijom arhitekturom ima bolje performanse kod predviđanja vrijednosti. Ovi podaci prikazani su u tablici 3 Iako je srednja kvadratna pogreška nešto veća koeficijent korelacije pokazuje veliku povezanost ulazno izlaznih vektora.

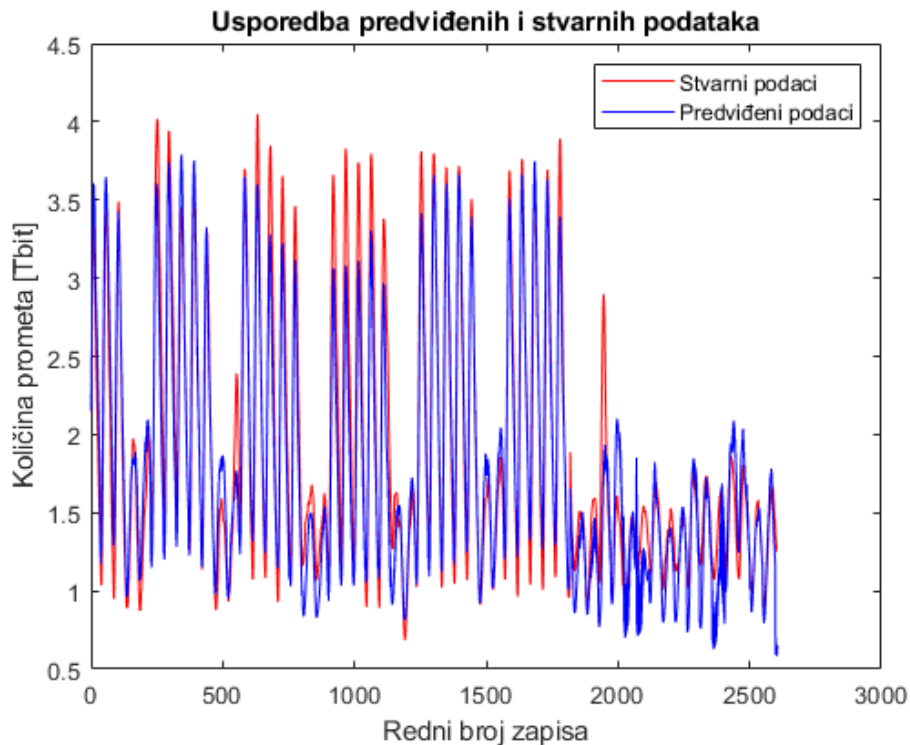
Tablica 3. Prikaz rezultata neuronske mreže pri predviđanju količine prometa odlaznog linka

| Broj skrivenih slojeva (broj neurona u pojedinom sloju) | Broj uzoraka za predviđanje | Srednja kvadratna pogreška (MSE) | Koeficijent korelacije |
|---|-----------------------------|----------------------------------|------------------------|
| 1 (200) | 2.605 | 0,0765 | 0,94 |
| 2 (200, 150) | 2.605 | 0,0542 | 0,96 |

Izvor: izradio autor

Na slici 21 prikazana je grafička usporedba predviđenih podataka i stvarnih podataka za jednoslojnu mrežu. Moguće je primijetiti da je mreža vrlo dobro naučila trend podataka no svejedno postoje odstupanja od stvarnih vrijednosti. To se pojavljuje, najvećim djelom, zbog već spomenute ne konzistencije podataka i premalo podataka za učenje mreže.

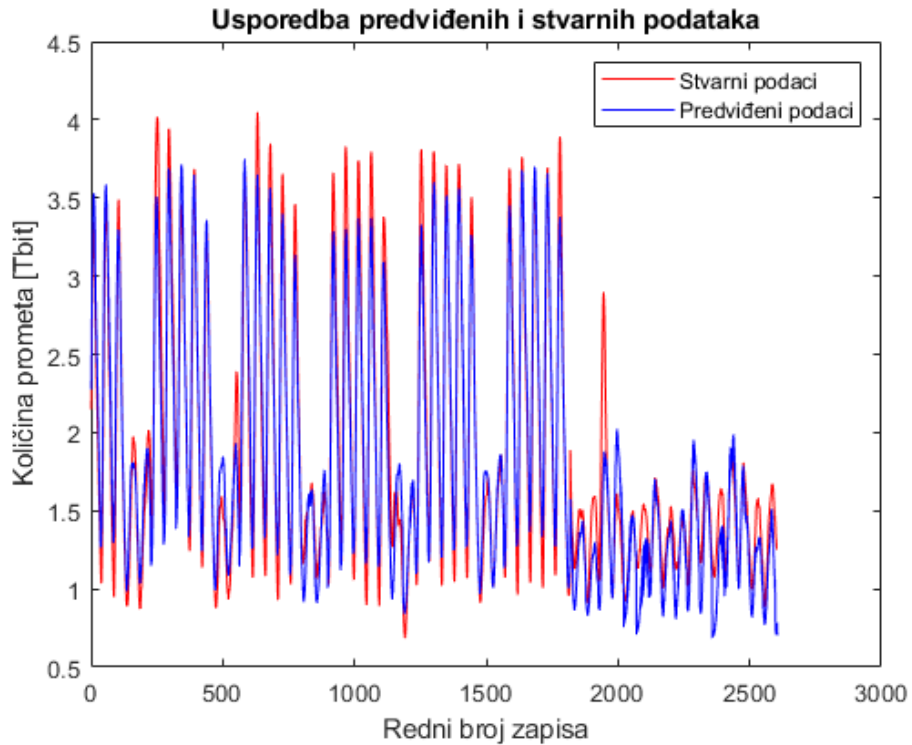
Kod mreža koje su učene odlaznim podacima dogodilo se da su mreže previše naučene. To je uzrokovalo vrlo dobre rezultate na setu podataka za učenja, ali vrlo loše rezultate kada se radi o skupu za predviđanje. Pa je bilo potrebno naći odgovarajući broj iteracija koje će mreža proći kako bi rezultati bili što bolji. Nizom testiranja s različitim brojevima iteracija pokazalo se da za složeniju mrežu povoljan broj iteracija 250, a za jednostavniju 3088.



Slika 21. Usporedba stvarnih i predviđenih podataka odlaznog linka koristeći jednoslojnu mrežu

Izvor: izradio autor

Slika 22 prikazuje istu usporedbu, ali za složeniju mrežu s dva sloja. Moguće je primijetiti da su rezultati vrlo slični, no iz tablice 3. vidljivo je da su rezultati složenije mreže bolji. Ovakav ishod najvećim dijelom uzrokovan je karakteristikom podataka. Na slici 19 moguće je uočiti kako se podatkovni skup sastoji od nekoliko sezona odnosno razdoblja koja imaju specifične trendove. S obzirom na to da se skup podijelio u dva dijela, a ako se podaci podijele kronološki mreža prilikom učenja neće vidjeti neke sezonske karakteristike, mreža ih neće moći naučiti. Stoga su podaci pomiješani tako da mreža ima uvid u sve sezonske karakteristike, a time ih može i naučiti. Ovaj problem bih se mogao riješiti prikupljanjem većeg skupa podataka tako da mreža može naučiti sve trendove i kada se dešavaju tijekom godine. Naravno, moguće je poboljšati rezultate samo na ovom skupu daljnjim ispitivanjem struktura mreže kao i proširivanjem ulaznog skupa s više informacija o ulaznim podacima, ali to nije najpovoljnije rješenje.



Slika 22. Usporedba stvarnih i predviđenih podataka odlaznog linka koristeći dvoslojnu mrežu
Izvor: izradio autor

6.2. Analiza rezultata za predviđanje dolaznog prometa

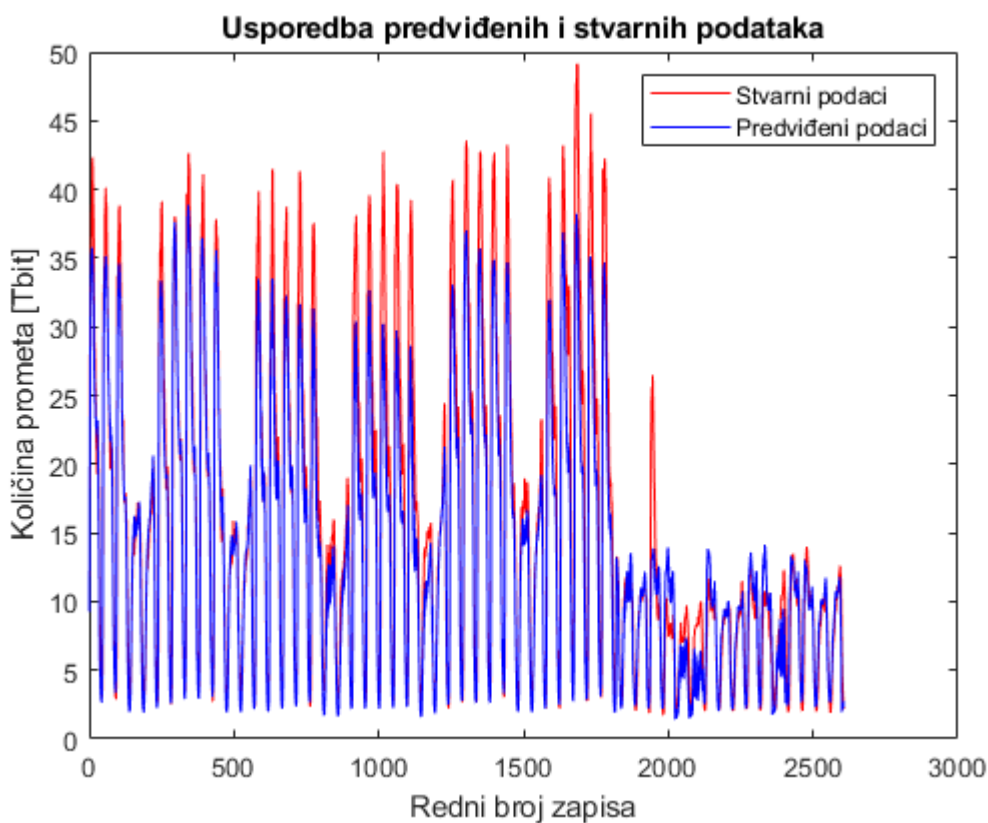
Podaci dolaznog prometa čine isti trendovi kao i kod podataka odlaznog prometa. Što će uglavnom rezultirati sličnim rezultatima prilikom predviđanja količine prometa. Jedina bitnija razlika je što su podaci odlaznog linka znatno manje šumoviti pa nije bilo potrebe primjenjivati pomični prosjek. Tablica 4 prikazuje vrijednosti performansi umjetne neuronske mreže prilikom treniranja na temelju podataka dolaznog linka.

Tablica 4. Prikaz rezultata učenja neuronskih mreža korištenih za predviđanje dolazne količine prometa

| Broj skrivenih slojeva (broj neurona u pojedinom sloju) | Broj uzoraka za predviđanje | Srednja kvadratna pogreška (MSE) | Koeficijent korelacije |
|---|-----------------------------|----------------------------------|------------------------|
| 1 (200) | 14.843 | 6,516 | 0,96 |
| 2 (200, 150) | 14.843 | 6,8174 | 0,95 |

Izvor: izradio autor

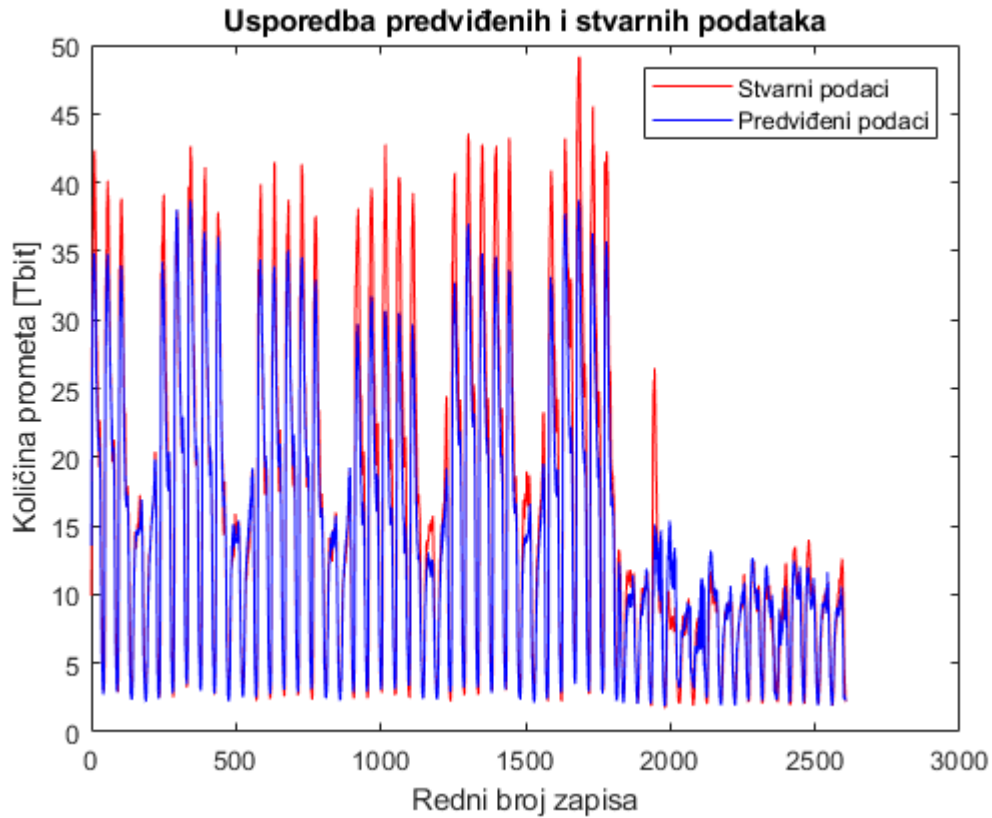
Vrijednosti podataka dolaznog prometa su znatno veće od vrijednosti odlaznog prometa. To je moguće vidjeti usporedbom grafova predviđanja odlaznog i dolaznog prometa. Veće greške koje se mogu primijetiti u tablicama 4 i 5 su također rezultat većih vrijednosti. To se odnosi na parametar Srednje kvadratne pogreške jer on ovisi i o redu vrijednosti s kojima neuronska mreža radi, dok je koeficijent korelacije relativni kazatelj performansi mreže.



Slika 23. Usporedba stvarnih i predviđenih podataka dolaznog linka koristeći jednoslojnu mrežu

Izvor: izradio autor

Na slici 23 prikazano je predviđanje količine prometa u usporedbi sa stvarnim vrijednostima predviđano jednostavnom mrežom. Kao i kod odlaznog prometa mreža je vrlo dobro naučila trend no na prikazu postoji vidljivo odstupanje od stvarnih vrijednosti.



Slika 24. Usporedba stvarnih i predviđenih podataka dolaznog linka koristeći dvoslojnu mrežu
Izvor: izradio autor

Slika 24 prikazuje iste podatke kao i slika 23 samo za složeniju mrežu odnosno mrežu sa dva skrivena sloja. Također, u tablici 5 prikazane su performanse umjetne neuronske mreže prilikom predviđanja količine dolaznog prometa. Na 23. i 24. slici vidljivo je da se količina prometa dolaznog linka kreće u rasponu od 2,5 do 50 terabita u razdoblju od pola sata dok se kod odlaznog linka radi o znatno manjim vrijednostima odnosno od 0,5 do 4 terabita u pola sata.

Tablica 1. Prikaz rezultata neuronske mreže pri predviđanju količine prometa dolaznog linka

| Broj skrivenih slojeva (broj neurona u pojedinom sloju) | Broj uzoraka za predviđanje | Srednja kvadratna pogreška (MSE) | Koeficijent korelacije |
|---|-----------------------------|----------------------------------|------------------------|
| 1 (200) | 2.606 | 13,90 | 0,96 |
| 2 (200, 150) | 2.606 | 11,93 | 0,97 |

Izvor: izradio autor

7. Zaključak

U ovom diplomskom radu korištene su umjetne neuronske mreže u svrhu predviđanja količine prometa na odlaznom i dolaznom linku. Predviđanje se vršilo na temelju dva skupa podataka, jedni podaci su predstavljali dolazni link, a drugi odlazni. Za učenje su se koristile acikličke neuronske mreže sa jednim ili dva skrivena sloja. Kvaliteta predviđanja mreže uvelike ovisi, između ostaloga, o podacima na temelju kojih mreža uči. Što je vidljivo i u rezultatima ovog rada. S obzirom na karakteristiku i nekonzistentnost podataka može se reći da je neuronska mreža relativno dobro naučila karakteristike skupa.

Neuronske mreže zbog svojih karakteristika pronalaze primjenu gotovo u svim poljima znanosti. Za to je odgovorna sposobnost rada s linearnim i nelinearnim skupovima. U korist im ide i razvitak računala te sve veća računalna snaga koja omogućuje brže učenje i složenije strukture. Pa tako pronalaze primjenu u telekomunikacijskim sustavima koji generiraju enormne količine podataka, a pomoću neuronskih mreža ti podaci se mogu na jednostavan način iskoristiti kao unaprjeđenje performansi mreže. Primjena se pronalazi u detektiranju malicioznih paketa, napadača, predviđanju zagušenja, predviđanju opterećenja i ponašanja korisnika mreže. Naravno to su samo neki od primjera u kojima neuronske mreže pokazuju svoju moć.

Međutim, neuronske mreže su još uvijek tehnologija koja se razvija. Iako uče vrlo složene procese problem predstavlja što za svaki od tih procesa potrebno kreirati umjetnu neuronsku mrežu sa strukturom koja će odgovarati tom problemu. I ta struktura će davati dobre rezultate samo za taj problem. To se pokazalo i u rezultatima ovoga rada, iako su podaci imali iste karakteristike, ali različite redove vrijednosti količina prometa na dolaznom linku se nije mogla predviđati neuronskom mrežom koja je učena s podacima odlaznog linka. Također, vrlo je važno imati dovoljnu količinu podataka na kojima će mreža učiti, inače mreža nauči samo dio ponašanja procesa.

Prikupljanjem podataka za dulja razdoblja i poznavanjem događaja ili perioda koji mogu biti značajni u smislu generiranja prometa mogle bi se poboljšati informacije koje se predaju mreži kao ulaz, a time i postići bolje rezultate neuronske mreže. Time bi se osiguralo bolje shvaćanje procesa koji se mrežu pokušava naučiti. Također, ispitivanjem složenijih arhitektura, aktivacijskih funkcija i algoritama učenja omogućiti će postizanje manje pogreške, a time i točnija predviđanja.

POPIS LITERATURE

- [1] Manish R. Joshi, Theyazn Hassn Hadi, A Review of Network Traffic Analysis and Prediction Techniques, 2015.
- [2] Paulo Cortez, Miguel Rio, Miguel Rocha, and Pedro Sousa, Internet Traffic Forecasting using Neural Networks, International Joint Conference on Neural Networks, 2006.
- [3] C.P.IJ. van Hinsbergen, J.W.C. van Lint, F.M. Sanders, Short Term Traffic Prediction Models, ITS World Congress, Beijing, China, October 2007
- [4] Ahmad Jobran Al-Mahasneh, Sreenatha G. Anavatti, and Matthew A. Garratt, The Development of Neural Networks applications from Perceptron to Deep Learning, School of Engineering and Information Technology, 2017
- [5] <https://www.neuraldesigner.com/blog/perceptron-the-main-component-of-neural-networks> (kolovoz 2020.)
- [6] Ivanjko E., Gold H.: Predavanja za kolegij: Umjetna inteligencija. Fakultet prometnih znanosti, Zagreb, 2019.
- [7] <https://towardsdatascience.com/introduction-to-artificial-neural-networks-ann-1aea15775ef9> (kolovoz 2020.)
- [8] <https://missinglink.ai/guides/neural-network-concepts/7-types-neural-network-activation-functions-right/> (kolovoz 2020.)
- [9] Dalbelo Bašić B., Čupić M., Šnajder J.: Umjetne neuronske mreže [Internet]. Fakultet elektrotehnike i računarstva, 2008.
- [10] Lončarić S. Predavanja za kolegij: Neuronske mreže, Fakultet elektrotehnike i računarstva, Zagreb, 2012.
- [11] Muhammad Faisal Iqbal , Muhammad Zahid, Durdana Habib, Lizy Kurian John, Efficient Prediction of Network Traffic for Real-Time Applications, Journal of Computer Networks and Communications Volume 2019, Article ID 4067135, 2019.
- [12] <https://www.carnet.hr/o-carnet-u/> (kolovoz 2020.)
- [13] <https://www.geant.org/About> (kolovoz 2020.)
- [14] Laboratorij i vještine – MATLAB, Uvod u SIMULINK, Fakultet elektrotehnike i računarstva, Zagreb, 2006.
- [15] Trevor Clarkson, Applications of Neural Networks in Telecommunications, King's College London, 2001.

- [16] Corina Botoca, Georgeta Budura, Neural Networks Intelligent Tools For Telecommunications Problems, 2003.
- [17] Mingzhe Chen, Ursula Challita, Walid Saad, Changchuan Yin, and M erouane Debbah, Machine Learning for Wireless Networks with Artificial Intelligence: A Tutorial on Neural Networks, 2017.
- [18] Robert A. Sowah , Kwadwo B. Ofori-Amanfo, Godfrey A. Mills, and Koudjo M. Koumadi, Detection and Prevention of Man-in-the-Middle Spoofing Attacks in MANETs Using Predictive Techniques in Artificial Neural Networks (ANN), Journal of Computer Networks and Communications, Volume 2019, Article ID 4683982, 2019.
- [19] <https://www.mathworks.com/help/deeplearning/ref/trainrp.html> (kolovoz 2020.)
- [20] Matea Ravnjak, Predviđanje količine prometa zasnovano na neuronskoj mreži, diplomski rad, Fakultet prometnih znanosti, 2015.
- [21] <https://towardsdatascience.com/machine-learning-basics-with-the-k-nearest-neighbors-algorithm-6a6e71d01761> (rujan 2020.)

POPIS ILUSTRACIJA

| | |
|--|----|
| Slika 1. Model perceptrona, izvor: izradio autor prema | 8 |
| Slika 2. Prikaz biološkog neurona | 9 |
| Slika 3. Prikaz umjetnog neurona | 10 |
| Slika 4. Prikaz funkcije praga | 12 |
| Slika 5. Prikaz funkcije predznaka | 12 |
| Slika 6. Prikaz linearne funkcije | 13 |
| Slika 7. Prikaz logističke funkcije | 13 |
| Slika 8. Prikaz topologije višeslojne neuronske mreže bez povratnih veza | 15 |
| Slika 9. Shema učenja pod nadzorom | 18 |
| Slika 10. Prikaz prozora za upravljanje podacima ujedno i početni prozor alata za neuronske mreže | 24 |
| Slika 11. Sučelje za kreiranje neuronske mreže | 25 |
| Slika 12. Prikaz strukture neuronske mreže s dva skrivena sloja | 27 |
| Slika 13. Prikaz arhitekture jednostavnije neuronske mreže s dva sloja | 27 |
| Slika 14. Prikaz sirovih podataka | 29 |
| Slika 15. Grafički prikaz sirovih podataka | 29 |
| Slika 16. Prikaz jedne od ulaznih matrica za učenje mreže | 31 |
| Slika 17. Prikaz parametara koji se mogu ugađati kod metode otpornog rasprostiranja pogreške | 33 |
| Slika 18. Prikaz pogreške mreže prilikom procesa učenja | 34 |
| Slika 19. Prikaz svih podataka u satnoj rezoluciji | 35 |
| Slika 20. Prikaz dva ulazna vektora koji čine jedan sat | 36 |
| Slika 21. Usporedba stvarnih i predviđenih podataka odlaznog linka koristeći jednoslojnu mrežu | 38 |
| Slika 22. Usporedba stvarnih i predviđenih podataka odlaznog linka koristeći dvoslojnu mrežu | 39 |
| Slika 23. Usporedba stvarnih i predviđenih podataka dolaznog linka koristeći jednoslojnu mrežu | 40 |
| Slika 24. Usporedba stvarnih i predviđenih podataka dolaznog linka koristeći dvoslojnu mrežu | 41 |

POPIS TABLICA

| | |
|---|----|
| Tablica 1. Prednosti neuronskih mreža i područje njihove primjene u okviru telekomunikacija | 20 |
| Tablica 2. Prikaz rezultata učenja neuronskih mreža korištenih za predviđanje odlazne količine prometa | 36 |
| Tablica 3. Prikaz rezultata neuronske mreže pri predviđanju količine prometa odlaznog linka | 37 |
| Tablica 4. Prikaz rezultata učenja neuronskih mreža korištenih za predviđanje dolazne količine prometa | 40 |
| Tablica 1. Prikaz rezultata neuronske mreže pri predviđanju količine prometa dolaznog linka | 42 |

Prilog 1. : Kod korišten za kreiranje ulaznih varijabli, napisan u Python programskom jeziku

```
# Uvoz potrebnih modula
import csv, datetime, holidays

# Definiranje putanje datoteke
filename = "export_GEANT_OUT.csv"

# Otvaranje pisaa i stvaranje tekstualnih datoteka u koje ce se zapisivati podaci
IspisSatUDanu=open("Ispis_csvCitac/SatUDanuSve.txt","w")
#IspisVrstaDana=open("VrstaDana.txt","w")
IspisDanUTjednu=open("Ispis_csvCitac/DanUTjednuSve.txt","w")
IspisDanUMjesecu=open("Ispis_csvCitac/DanUMjesecuSve.txt","w")
IspisBlagdan=open("Ispis_csvCitac/BlagdanSve.txt","w")
IspisPetominutniDio=open("Ispis_csvCitac/PetominutniDio.txt","w")
IspisPolusatniDio=open("Ispis_csvCitac/PolusatniDio.txt","w")

# Inicijalizacija brojača x
x=0

#uvođenje liste s blagdanima i unos blagdana u listu
Praznici=[]

for date in holidays.Croatia(years=2019).items():
    Praznici.append(date[0])

for date in holidays.Croatia(years=2020).items():
    Praznici.append(date[0])

# Čitanje csv datoteke
```

```

with open(filename, 'r') as csvfile:
    # Stvaranje čitača csv datoteke
    csvreader = csv.reader(csvfile)

    # Preskakanje prvog retka s imenima stupaca
    next(csvreader)

    # čitanje podataka redak po redak
    for row in csvreader:
        x+=1
        timestamp=int(row[1])/1000000000 # dijeljenje vremenske značke kako bi se dobile
vrijednosti bez nula
        vrijeme=datetime.datetime.utcfromtimestamp(timestamp)

        if vrijeme.date() in Praznici:           #Provjera je li pojedini datum praznik
            IspisBlagdan.write('1'+'\n')
        else:
            IspisBlagdan.write('0'+'\n')

    #petlja za određivanje petominutnog dijela (0-11)
    minuta=vrijeme.minute
    dio=0

    if minuta<5:
        dio=0
    elif minuta<10:
        dio=1
    elif minuta<15:
        dio=2
    elif minuta<20:
        dio=3
    elif minuta<25:

```



```
    dio=4
elif minuta<30:
    dio=5
elif minuta<35:
    dio=6
elif minuta<40:
    dio=7
elif minuta<45:
    dio=8
elif minuta<50:
    dio=9
elif minuta<55:
    dio=10
else:
    dio=11
```

```
#petlja za određivanje polusatnog dijela (0-2)
```

```
if minuta<30:
    plusatnidio=1
else:
    plusatnidio=2
```

```
IspisSatUDanu.write(str(vrijeme.hour)+'\n')
IspisDanUTjednu.write(str(vrijeme.weekday())+'\n')
IspisDanUMjesecu.write(str(vrijeme.day)+'\n')
IspisPetominutniDio.write(str(dio)+'\n')
IspisPolusatniDio.write(str(plusatnidio)+'\n')
```

```
#if x==20000: #broj zapisa koji se vadi iz csv-a
    # break
```

```
#Zatvaranje pisaca
```

IspisSatUDanu.close()

IspisDanUTjednu.close()

IspisDanUMjesecu.close()

IspisBlagdan.close()

IspisPetominutniDio.close()

IspisPolusatniDio.close()

Prilog 2. : Kod korišten u MATLAB-u za kreiranje ulaznih vektora i daljnju obradu podataka

```
plot_
```

```
podaci=readmatrix('export_GEANT_OUT.csv');
```

```
SatUDanu=load('PodaciPhyton/SatUDanuSve.txt');
```

```
DanUTjednu=load('PodaciPhyton/DanUTjednuSve.txt'); % 0= ponedjeljak,1= utorak...
```

```
DanUMjesecu=load('PodaciPhyton/DanUMjesecuSve.txt');
```

```
Blagdan=load('PodaciPhyton/BlagdanSve.txt');
```

```
PolusatniDio=load('PodaciPhyton/PolusatniDio.txt');
```

```
PetominutniDio=load('PodaciPhyton/PetominutniDio.txt');
```

```
Podaci=podaci(:,3);
```

```
s1=size(Podaci,1);
```

```
%računanje do kojeg člana uzeti podatke da svi uzorci budu puni(da i zadnji ima 12)
```

```
m=s1-mod(s1,12);
```

```
m2=s1-mod(s1,6);
```

```
%stupci koji čine jedan sat i pola sata zapisa
```

```
PodaciPoSatu=reshape(Podaci(1:m),12,[]);
```

```
SatUDanuProsjek=reshape(SatUDanu(1:m),12,[]);
```

```
DanUTjednuProsjek=reshape(DanUTjednu(1:m),12,[]);
```

```
DanUMjesecuProsjek=reshape(DanUMjesecu(1:m),12,[]);
```

```
BlagdanProsjek=reshape(Blagdan(1:m),12,[]);
```

```
PodaciPolaSata=reshape(Podaci(1:m2),6,[]);
```

```
SatUDanuProsjekp=reshape(SatUDanu(1:m2),6,[]);
```

```
DanUTjednuProsjekp=reshape(DanUTjednu(1:m2),6,[]);
```

```
DanUMjesecuProsjekp=reshape(DanUMjesecu(1:m2),6,[]);
```

```
BlagdanProsje kp=reshape(Blagdan(1:m2),6,[]);  
PolusatniDiop=reshape(PolusatniDio(1:m2),6,[]);
```

```
%računanje prosjeka brzine u jednom satu i pola sata  
PodaciProsje k=transpose(sum(PodaciPoSatu,1)/12);  
PodaciProsje k=PodaciProsje k.';
```

```
PodaciProsje kp=transpose(sum(PodaciPolaSata,1)/6);  
PodaciProsje kp=PodaciProsje kp.';
```

```
%stvaranje po jedan zapis za svaki sat i za svakih pola sata  
SatUDanuProsje k=SatUDanuProsje k(1,:);  
DanUTjednuProsje k=DanUTjednuProsje k(1,:);  
DanUMjesecuProsje k=DanUMjesecuProsje k(1,:);  
BlagdanProsje k=BlagdanProsje k(1,:);
```

```
SatUDanuProsje kp=SatUDanuProsje kp(1,:);  
DanUTjednuProsje kp=DanUTjednuProsje kp(1,:);  
DanUMjesecuProsje kp=DanUMjesecuProsje kp(1,:);  
BlagdanProsje kp=BlagdanProsje kp(1,:);  
PolusatniDiop=PolusatniDiop(1,:);
```

```
%transponiranje  
DanUTjednuProsje k=DanUTjednuProsje k.';  
SatUDanuProsje k=SatUDanuProsje k.';  
DanUMjesecuProsje k=DanUMjesecuProsje k.';  
BlagdanProsje k=BlagdanProsje k.';
```

```
DanUTjednuProsje kp=DanUTjednuProsje kp.';  
SatUDanuProsje kp=SatUDanuProsje kp.';  
DanUMjesecuProsje kp=DanUMjesecuProsje kp.';  
BlagdanProsje kp=BlagdanProsje kp.';
```

```

PolusatniDiop=PolusatniDiop.';

% stvaranje ulaznog vektora
UlazSve=[BlagdanProsjek,DanUMjesecuProsjek,DanUTjednuProsjek,SatUDanuProsjek];
UlazSve=UlazSve.';

% raspodjela na dio za učenje i dio za predviđanje
SetMeteUcenje=PodaciProsjek(1:6100);
SetMetePredvidanje=PodaciProsjek(6101:end);

SetUlazUcenje=UlazSve(:,1:6100);
SetUlazPredvidanje=UlazSve(:,6101:end);

% pretvaranje u bitove
SetMeteUcenje=SetMeteUcenje*8;
SetMetePredvidanje=SetMetePredvidanje*8;

% pretvaranje u količinu, mjeri se svakih pet min stoga je prosječni promet
% u pet minuta
SetMeteUcenjeKol=SetMeteUcenje*300;
SetMetePredvidanjeKol=SetMetePredvidanje*300;

% smanjivanje reda na terabite
SetMeteUcenjeKolTera=SetMeteUcenjeKol/1000000000000;
SetMetePredvidanjeKolTera=SetMetePredvidanjeKol/1000000000000;

% izbacivanje podataka sa velikim odstupanjem
SetMeteUcenjeKolTera(193:195)=[];
SetUlazUcenje(:,193:195)=[];

% primjenjivanje pomičnog prosjeka
SetMeteUcenjeKolTera= movmean(SetMeteUcenjeKolTera,15);

```

```

SetMetePredvidanjeKolTera=movmean(SetMetePredvidanjeKolTera,15);

%Set1: satna rezolucija, svi podaci
UlazniSet1_ucenje=SetUlazUcenje;
CiljniSet1_ucenje=SetMeteUcenjeKolTera;

UlazniSet1_predvidanje=SetUlazPredvidanje;
CiljniSet1_predvidanje=SetMetePredvidanjeKolTera;

%Set2: petominutna rezolucija, svi podaci
UlazniSet2_sve=[Blagdan,DanUMjesecu,DanUTjednu,SatUDanu,PolusatniDio,Petominut
niDio];
CiljniSet2_sve=Podaci;

UlazniSet2_sve=UlazniSet2_sve.';
CiljniSet2_sve=CiljniSet2_sve.';

CiljniSet2_sve=CiljniSet2_sve*300;
CiljniSet2_sve=CiljniSet2_sve*8;
CiljniSet2_sve=CiljniSet2_sve/1000000000000;
CiljniSet2_sve=movmean(CiljniSet2_sve,15);

CiljniSet2_sve(2294:2350)=[];
UlazniSet2_sve(:,2294:2350)=[];

UlazniSet2_ucenje=UlazniSet2_sve(:,1:89017);
UlazniSet2_predvidanje=UlazniSet2_sve(:,89018:end);

CiljniSet2_ucenje=CiljniSet2_sve(1:89017);
CiljniSet2_predvidanje=CiljniSet2_sve(89018:end);

%Set3: polusatna rezolucija

```

```
UlazniSet3_sve=[BlagdanProsje kp,DanUMjesecuProsje kp,DanUTjednuProsje kp,SatUDa  
nuProsje kp,PolusatniDiop];
```

```
UlazniSet3_sve=UlazniSet3_sve.;
```

```
CiljniSet3_sve=PodaciProsje kp;
```

```
CiljniSet3_sve=CiljniSet3_sve*300;
```

```
CiljniSet3_sve=CiljniSet3_sve*8;
```

```
CiljniSet3_sve=CiljniSet3_sve/1000000000000;
```

```
CiljniSet3_sve=movmean(CiljniSet3_sve,15);
```

```
CiljniSet3_sve(380:394)=[];
```

```
UlazniSet3_sve(:,380:394)=[];
```

```
UlazniSet3_sve=[UlazniSet3_sve; a];
```

```
UlazniSet3_ucenje=UlazniSet3_sve(:,1:14843);
```

```
UlazniSet3_predvidanje=UlazniSet3_sve(:,14844:end);
```

```
CiljniSet3_ucenje=CiljniSet3_sve(1:14843);
```

```
CiljniSet3_predvidanje=CiljniSet3_sve(14844:end);
```

```
clearvars -except a UlazniSet1_ucenje CiljniSet1_ucenje UlazniSet1_predvidanje  
CiljniSet1_predvidanje UlazniSet2_sve CiljniSet2_sve UlazniSet3_sve CiljniSet3_sve  
UlazniSet2_ucenje UlazniSet2_predvidanje CiljniSet2_ucenje CiljniSet2_predvidanje  
CiljniSet2_predvidanje UlazniSet3_predvidanje CiljniSet3_ucenje CiljniSet3_predvidanje  
UlazniSet3_ucenje
```

```
%clear SatUDanu DanUTjednu DanUMjesecu Blagdan podaci m
```

```
%-----
```

```
%Dio za određivanje dijelova i miješanje podataka
```

```
%podjela sezonalnosti
```

```
UlazniSet3_sve(6,1:2918)=0;
```

```
UlazniSet3_sve(6,2919:3617)=2;
```

```
UlazniSet3_sve(6,3618:8546)=1;
UlazniSet3_sve(6,8547:9333)=0;
UlazniSet3_sve(6,9334:12590)=1;
UlazniSet3_sve(6,12591:15940)=0;
UlazniSet3_sve(6,15941:end)=2;
```

```
%podijela na dijelove
```

```
a=CiljniSet3_sve(1:2918); %1
b=CiljniSet3_sve(2919:3617); %2
c=CiljniSet3_sve(3618:8546); %3...
d=CiljniSet3_sve(8547:9333);
e=CiljniSet3_sve(9334:12590);
f=CiljniSet3_sve(12591:15940);
g=CiljniSet3_sve(15941:end); %7
```

```
% provjera=(size(a,2)+size(b,2)+size(c,2)+size(d,2)+size(e,2)+size(f,2)+size(g,2));
```

```
CiljniSet3_sve_mix=[ c g f b a e d];
```

```
%ulazniSet
```

```
au=UlazniSet3_sve(:,1:2918);
bu=UlazniSet3_sve(:,2919:3617);
cu=UlazniSet3_sve(:,3618:8546);
du=UlazniSet3_sve(:,8547:9333);
eu=UlazniSet3_sve(:,9334:12590);
fu=UlazniSet3_sve(:,12591:15940);
gu=UlazniSet3_sve(:,15941:end);
```

```
%
provjera2=(size(au,2)+size(bu,2)+size(cu,2)+size(du,2)+size(eu,2)+size(fu,2)+size(gu,2));
```



```
UlazniSet3_sve_mix=[ cu gu fu bu au eu du];
```

```
%kreiranje ulaznih i ciljnih setova
```

```
UlazniSet3_ucenje_mix=UlazniSet3_sve_mix(:,1:14843);
```

```
UlazniSet3_predvidanje_mix=UlazniSet3_sve_mix(:,14844:end);
```

```
CiljniSet3_ucenje_mix=CiljniSet3_sve_mix(1:14843);
```

```
CiljniSet3_predvidanje_mix=CiljniSet3_sve_mix(14844:end);
```

```
clear a au b bu c cu d du e eu f fu g gu
```



Sveučilište u Zagrebu
Fakultet prometnih znanosti
10000 Zagreb
Vukelićeva 4

IZJAVA O AKADEMSKOJ ČESTITOSTI I SUGLASNOST

Izjavljujem i svojim potpisom potvrđujem kako je ovaj _____ diplomski rad

isključivo rezultat mog vlastitog rada koji se temelji na mojim istraživanjima i oslanja se na objavljenu literaturu što pokazuju korištene bilješke i bibliografija.

Izjavljujem kako nijedan dio rada nije napisan na nedozvoljen način, niti je prepisan iz necitiranog rada, te nijedan dio rada ne krši bilo čija autorska prava.

Izjavljujem također, kako nijedan dio rada nije iskorišten za bilo koji drugi rad u bilo kojoj drugoj visokoškolskoj, znanstvenoj ili obrazovnoj ustanovi.

Svojim potpisom potvrđujem i dajem suglasnost za javnu objavu _____ diplomskog rada

pod naslovom **Predviđanje količine dolaznog i odlaznog prometa komunikacijskih mreža pomoću neuronske mreže**

na internetskim stranicama i repozitoriju Fakulteta prometnih znanosti, Digitalnom akademskom repozitoriju (DAR) pri Nacionalnoj i sveučilišnoj knjižnici u Zagrebu.

Student/ica:

U Zagrebu, _____ 17.9.2020 _____

(potpis)