Klasifikacija pulsara korištenjem metoda umjetne inteligencije

Stjepanović, Dean

Master's thesis / Diplomski rad

2022

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Rijeka, Faculty of Engineering / Sveučilište u Rijeci, Tehnički fakultet**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:190:821569

Rights / Prava: Attribution 4.0 International/Imenovanje 4.0 međunarodna

Download date / Datum preuzimanja: **2024-05-18**

Repository / Repozitorij:

Repository of the University of Rijeka, Faculty of Engineering

SVEUČILIŠTE U RIJECI

TEHNIČKI FAKULTET

Diplomski sveučilišni studij elektrotehnike

Diplomski rad

KLASIFIKACIJA PULSARA KORIŠTENJEM METODA UMJETNE INTELIGENCIJE

Rijeka, rujan 2022.

Dean Stjepanović

0069067611

SVEUČILIŠTE U RIJECI

TEHNIČKI FAKULTET

Diplomski sveučilišni studij elektrotehnike

Diplomski rad

KLASIFIKACIJA PULSARA KORIŠTENJEM METODA UMJETNE INTELIGENCIJE

Mentor: Prof. dr. sc. Zlatan Car

Rijeka, rujan 2022.

Dean Stjepanović

0069067611

SVEUČILIŠTE U RIJECI
TEHNIČKI FAKULTET
POVJERENSTVO ZA DIPLOMSKE ISPITE

Rijeka, 21. ožujka 2022.

Zavod: **Zavod za automatiku i elektroniku**
Predmet: **Primjena umjetne inteligencije**
Grana: **2.03.06 automatizacija i robotika**

ZADATAK ZA DIPLOMSKI RAD

Pristupnik: **Dean Stjepanović (0069067611)**
Studij: Diplomski sveučilišni studij elektrotehnike
Modul: Automatika

Zadatak: **Klasifikacija pulsara korištenjem metoda umjetne inteligencije/Pulsar Classification using artificial intelligence methods**

Opis zadatka:

Izraditi pregled literature za područje primjene umjetne inteligencije u astronomiji, posebice zvjezdanoj astronomiji. Statistički analizirati set podataka, te odrediti opisne statistike. Ispitati utjecaj ulaznih varijabli na izlazne varijable. Razviti klasifikacijski model umjetne inteligencije. Ispitati performanse modela i testirati utjecaj hiperparametara na iste. Testirati utjecaj balansiranosti seta podataka na performanse modela.

Rad mora biti napisan prema Uputama za pisanje diplomskih / završnih radova koje su objavljene na mrežnim stranicama studija.

Zadatak uručen pristupniku: 21. ožujka 2022.

Mentor:

Prof. dr. sc. Zlatan Car

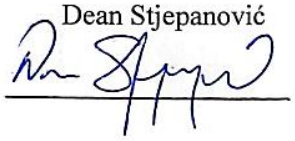
Predsjednik povjerenstva za diplomski ispit:

Prof. dr. sc. Viktor Sučić

IZJAVA

Sukladno članku 9. Pravilnika o diplomskom radu, diplomskom ispitu i završetku diplomskih sveučilišnih studija Tehničkog fakulteta Sveučilišta u Rijeci izjavljujem da sam izradio ovaj diplomski rad samostalno, koristeći vlastito znanje i navedenu literaturu, u razdoblju od datuma zadavanja zadatka do datuma predaje.

Dean Stjepanović



ZAHVALA

Zahvaljujem roditeljima na strpljenju, razumijevanju i podršci. Zahvala kolegama i prijateljima koje sam stekao tijekom studiranja na svim lijepim trenucima.
Posebne zahvale upućujem prof. dr. sc. Zlatanu Caru na prihvaćanju mentorstva za izradu rada i asist. Sandiju Baressi Šegoti, mag. ing. comp. na uputama i pomoći tijekom izrade.

-

Sadržaj

1. UVOD 1
2. PREGLED LITERATURE 2
2.1. Klasifikacija asteroida pomoću umjetne neuronske mreže 2
2.2. Pobošljavanje identifikacije zvijezda primjenom umjetne inteligencije 3
2.3. Identifikacija crnih rupa primjenom strojnog učenja 4
2.4. Usporedba metoda strojnog učenja za spektroskopiju gama zraka 5
2.5. Primjena umjetne inteligencije za simulacije formiranja galaksija 5
2.6. Određivanje istraživačkih prioriteta u području astronomije primjenom strojnog učenja ... 6
2.7. Klasifikacija boja planeta analognih planetu Zemlji primjenom umjetne inteligencije 7
2.8. SkyNet 8
2.9. Klasifikacija kvazara primjenom strojnog učenja 8
2.9.1. AdaBoost 9
3. METODOLOGIJA 10
3.1. Set podataka 10
3.1.1. Standardni opis podataka 11
3.1.2. Analiza osnovnih svojstava seta podataka 13
3.1.3. Deskriptivna statistika seta podataka 16
3.1.4. Korelacija između stupaca 17
3.1.5. Ujednačeni set podataka 18
3.2. Umjetne neuronske mreže 22
3.2.1. Umjetni neuron (Threshold logic unit, TLU) 24
3.2.1.1. Ulaz 25
3.2.1.2. Težinska funkcija 25
3.2.1.3. Sumacijska točka 25
3.2.1.4. Funkcija praga (aktivacijska funkcija) i izlaz 26
3.2.2. Mreža bez povratnih mreža 27
3.2.3. Jednoslojne mreže bez povratnih veza 27
3.2.4. Mreže s povratnim vezama 28
3.3. Višeslojni perceptron (MLP) 29
3.3.1. Programiranje višeslojnog perceptrona 32
3.4. Aktivacijske funkcije 34
3.4.1. Sigmoidalna aktivacijska funkcija 35
3.4.2. Tangens hiperbolna aktivacijska funkcija 37

3.4.3. ReLU aktivacijska funkcija 39
3.5. Usporedba zvijezda i pulsara 42
3.5.1. Formiranje i karaktetistike pulsara 43
4. TRENIRANJE MODELA I REZULTATI 45
5. ZAKLJUČAK 56
6. LITERATURA 57
7. POPIS SLIKA 59
8. POPIS TABLICA 60
9. SAŽETAK 61
10. SUMMARY 62
DODATAK A - PROGRAMSKI KOD ZA TRENIRANJE MODELA 63
DODATAK B – PROGRAMSKI KOD ZA VIŠESLOJNI PERCEPTRON 65

1. UVOD

Astronomske klasifikacije udaljenih objekata u svemiru teški su zadaci za astronome jer podaci kojima raspolažu često su neprecizni i kompleksni za izračun, ali neovisno o problemima, metode klasifikacija za proučavanje prirode svemira astronomi koriste više od stoljeća. Jedna od najpopularnijih klasifikacija u polju astronomije, pod koju teoretski spadaju i neutronske zvijezde, naziva se spektralna specifikacija zvijezda. Iako spadaju u skupinu zvijezda, pulsari su po astronomskim jedinicama značajno maleni objekti sa specifikacijama koji se ne mogu precizno definirati od strane stručnjaka. Najčešće nastaju nakon supernove gigantskih zvijezda te iako mogu biti i do nekoliko milijardi puta manje od zvijezda, njihova gravitacijska vrijednost graniči s gravitacijskim vrijednostima crne rupe.

Primarno zbog veličine objekta, a zatim i njihovog broja, klasifikacija pulsara predstavlja problem astronomima. Razvojem novih i moćnih tehnologija klasifikacije te enormnim povećanjem računalnih kapaciteta i razvoja novih generacija višeobjektnih digitalnih detektora, klasifikacija astronomskih objekata bitno je drukčija za razliku od prijašnjih godina kada su mogućnosti astronoma bile uvelike manje. Upravo iskorištavanjem omogućenih računalnih kapaciteta i metoda klasifikacije putem umjetne inteligencije, cilj je stvoriti klasifikaciju pulsara s visokim postotkom preciznosti.

Primjenom HTRU2 seta podataka koji opisuje uzorke kandidata pulsara (nastao istraživanjem svemira visokom vremenskom rezolucijom), zadatak je kreirati programski kod za treniranje modela u programskom paketu „Python“, čiji rezultati strojnog učenja predočuju uspješnost i preciznost klasifikacije pulsara. Programski kod u sebi sadrži višeslojni perceptron, jednu od najvažnijih komponenti koda, koji definira navedenu preciznost karakterizacije. Izmjenom broja skrivenih slojeva i aktivacijskih funkcija koje definiraju način strojnog učenja, poboljšavaju se (ili pogoršavaju) postoci preciznosti, čiji će se najpovoljniji rezultati prikazati kroz dijagrame i tablice.

2. PREGLED LITERATURE

Promatrajući veličinu i složenost podataka, klasifikacija astronomskih objekata doživljava eksponencijalno ubrzani porast. Zbog tog porasta, potiče se razvoj znanosti upravljanja podataka koji će biti predstavljen kao vrlo koristan suputnik već uobičajenoj paradigmi analize podataka koja upravlja modelima, u kojima astronomi i programeri razvijaju automatizirane alate za iskorištavanje skupova podataka i izdvajanje novih informacija koje se izvlače iz upravo tih skupova. Računalna obrada podataka u astronomiju uvedena je u kasnim šezdesetim godina 20. stoljeća. Od tog vremena postignut je golem napredak u samom polju astronomije zahvaljujući računalnoj obradi podataka, koji je uvelike pomagao u rješavanju problema koji bi u periodu prije uvođenja računala u astronomiji bili klasificirani kao neostvarivi. Ovaj proces simbioze računala i astronomije napreduje iz godine u godinu, prvenstveno zbog modernizacije računala, odnosno, iz razloga jer računala postaju sve moćnija i pristupačnija. Međutim, napredak u današnjem vremenu kategorizira se kao napredak stupnja, a ne principa , zbog kojeg se ponovno ispitalo područje računalne znanosti i njihove inovativne tehnike koje bi se mogle primijeniti u astronomiji i uvelike utjecati na sami rast astronomije. Jedna od tih tehnika koja je najviše obećavala bila je grana umjetne inteligencije. Posljednjih godina, algoritmi strojnog učenja postali su sve popularniji među astronomima te zbog njihovih povoljnih značajki, ovaj tip algoritama trenutno se koristi za široki spektar zadataka. U nastavku poglavlja obratit će se pozornost na pregled literature koji obrađuje umjetnu inteligenciju i strojno učenje u astronomiji.

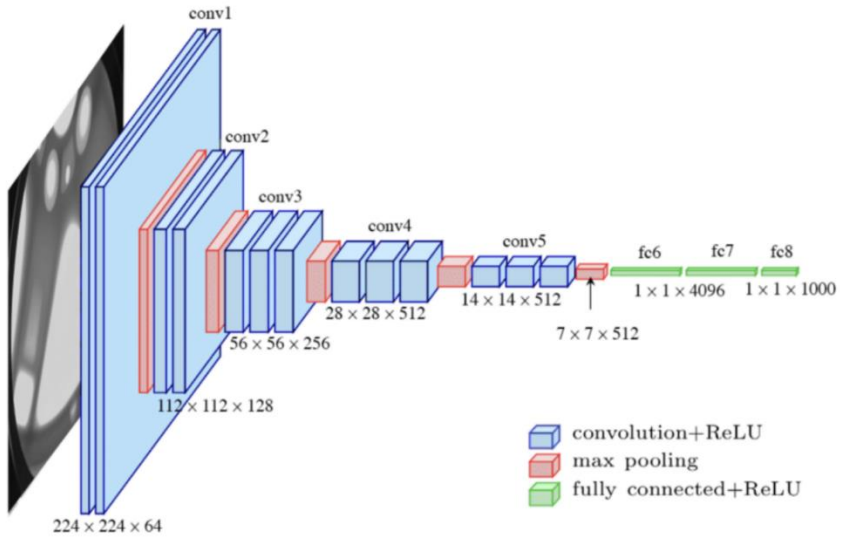
2.1. Klasifikacija asteroida pomoću umjetne neuronske mreže

U članku izvornog naziva "*Artificial Neural Network classification of asteroids in the M1:2 mean-motion resonance with Mars",* opisan je princip korištenja višeslojne neuronske mreže u svrhu automatskog prepoznavanja ponašanja asteroidnih orbita na koje utječe tzv. M1:2 rezonancija srednjeg kretanja s Marsom. Korišteni model bio je u stanju proizvesti znatno više od 85% razina za identifikaciju slika asteroidnih argumenata primjenjujući standardnu metriku kao što su točnost, preciznost i opoziv, čime je omogućena identifikacija tipa orbite svih numeriranih asteroida u skupini.

Također, korištenjem nadziranih metoda strojnog učenja koji su optimizirani putem genetskih algoritama, omogućeno je predviđanje orbitalnog statusa svih asteroida s više opozicija u samom području. Zaključno istraživanju, potvrđeno je djelovanje navedene rezonancije M1:2 na orbite skupina asteroida koji su bili promatrani.

2.2. Pobošljavanje identifikacije zvijezda primjenom umjetne inteligencije

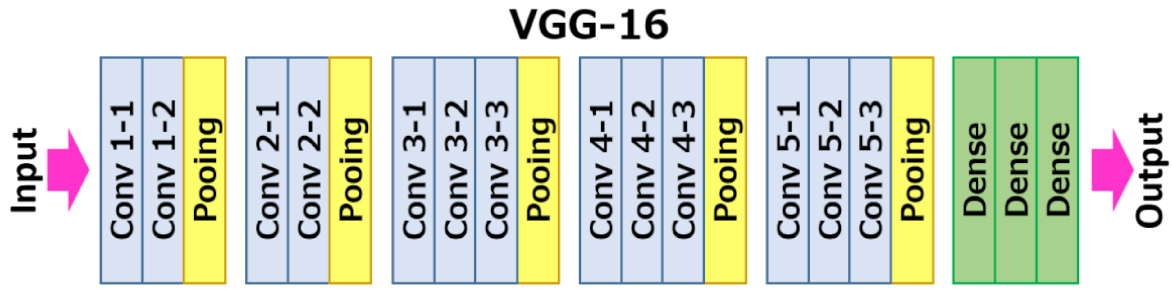
2019. godine, članovi tima državnog sveučilišta u New Yorku, izradili su članak u kojem su prezentirali opis i rezultate na temelju primjene algoritma koji se koristi za identifikaciju zvijezda. Algoritam koji je poboljšan modelom konvolucijske neuronske mreže Vgg16 analizira, identificira i klasificira slike zvijezda koje su locirane u tzv. "lost-inspace" području svemira. Set podataka za učenje konstruiran je za postizanje optimalne izvedbe mreža. Rezultati simulacije pokazali su da je predloženi algoritam vrlo otporan na razne vrste šuma, uključujući šum položaja, magnitudu šuma, lažne zvijezde i kutne brzine tragača. S dubokom konvolucijskom neuronskom mrežom, točnost identifikacije održava se na 96%, unatoč svim prekidima i šumovima, što je značajno poboljšanje u odnosu na tradicionalne algoritme mreže.



Slika 2.1.Grafički prikaz konvolucijske neuronske mreže Vgg16 [11]

Konvolucijska neuronska mreža Vgg16 model je neuronske mreže predložen od strane britanskih znanstvenika Karen Simonyan i Andrew Zissermana sa Sveučilišta Oxford u sklopu rada „Vrlo duboke konvolucijske mreže za prepoznavanje velikih slika".

Model postiže točnost od visokih 92.7% top-5 testova u skupu podataka „ImageNet“, koji sadrže preko 14 milijuna slika koje su grupirane kroz 1000 klasa. Princip rada Vgg 16 modela je zamjena filtara veličine jezgre sa više 3x3 filtara koji su poredani jedan iza drugoga. Ideja iza principa korištenja 3x3 filtara je da dva uzastopna 3x3 filtra osiguravaju učinkovito receptivno polje od 5x5. Slično tome principu, tri 3x3 filtra efektivno nadoknađuju receptivno polje od 7x7 filtara itd. Uz tri konvolucijska sloja, Vgg16 sadrži i tri nelinearna aktivacijska sloja (umjesto jednog), što čini funkcije odlučivanja konzervativnijima. Time se omogućuje brže konvergiranje mreže. Druga prednost ovog modela je značajno smanjivanje broja težinskih parametara u modelu. Vgg 16 je treniran tjednima i koristio je najmoćnije dostupne procesore.



Slika 2.2. Način poretka 3x3 filtara kroz konvolucijsku mrežu Vgg16 [12]

2.3. Identifikacija crnih rupa primjenom strojnog učenja

Sudionici u ovom istraživačkom radu naglašavaju povećanje popularnosti strojnog učenja u astronomiji i u astrofizici. Rad je prezentiran kao primjena strojnog učenja na više od tisuću modela kuglastih skupova (eng. globular cluster), koji su simulirani s projektom “MOCCA-Survey Database”. Model strojnog učenja zatim se primjenjuje na dostupne promatrane parametre za galaktičke kuglaste skupove kako bi se identificirao broj skupova koji najvjerojatnije sadrže značajan broj crnih rupa i otkrio uvid u svojstva koja dovode do formiranja podsustava crnih rupa. Ovim modelom, članovi tima uspjeli su napraviti uži izbor od 18 galaktičkih kuglastih skupova koji imaju najveću vjerojatnost za sadržavanja podsustava s crnim rupama. Isto tako, naglašeno je kako ss navedeni kandidati, M22, M10 i NGC 3201, podudaraju sa neovisnim simulacijama i prethodnim istraživanjima koji su ručno uspoređivali simulirane modele kuglastih skupova s promatranim svojstvima. Uz dobivene rezultate, kreiran je online alat koji omogućuje predviđanje vjerojatnosti nakon unosa vidljivih svojstava.

2.4. Usporedba metoda strojnog učenja za spektroskopiju gama zraka

U navedenom članku eksperimentira se sa različitim algoritmima iz područja umjetne inteligencije kako bi se postigao što optimalniji pristup automatizaciji spektroskopije gama zraka. Ukratko, gama zrake prikazane su kao elektromagnetski valovi koji nastaju kao posljedica kvantnih prijelaza jezgri atoma iz pobuđenih stanja. Za mapiranje i detekciju gama zraka iz svemira koristi se NASA-in svemirski teleskop pod nazivom "Fermi Gamma-ray", lansiran u lipnju 2008. godine. Od tog trenutka, teleskop je datirao 1873 objekta koji odašilju gama zrake.

Na osnovu tog popisa, utvrđeno je kako su algoritmi za prepoznavanje uzoraka, kao što su umjetne neuronske mreže i konvolucijske neuronske mreže, glavni kandidati za izvođenje automatizirane spektroskopije gama zraka. Način na koji ovi modeli treniraju i rade oponaša način rada obučenih spektroskopista koji identificiraju i analiziraju spektre. Primjenom navedenih algoritama ostvaruje se poboljšanje analize i identifikacije spektara gama zraka sa velikim kalibracijskim pomakom i nepoznatim poljima pozadinskog zračenja.

2.5. Primjena umjetne inteligencije za simulacije formiranja galaksija

Rad prezentira proces kozmoloških simulacija formiranja galaksija koje su još uvijek ograničene svojom prostornom rezolucijom te ne mogu iz prvih načela formirati neke od procesa kao što su npr. formiranje zvijezda, koji su ključni u pokretanju evolucije galaksija. Posljedica toga je oslanjanje na relativno neprecizne parametre koji pokušavaju uhvatiti razmjere i fizičke procese koji se ne mogu izravno riješiti u simulaciji. U navedenom eksperimentu prikazana je mogućnost korištenja tehnika strojnog učenja koje je primijenjeno na stvarne i simulirane slike galaksija kako bi se napravila razlika između različitih vrijednosti parametara korištenjem cjelovitog informacijskog sadržaja astronomske slike umjesto njegovog sažimanja u ograničeni skup vrijednosti poput veličine volumena plinskih masa. Članovi tima primjenjuju tzv. VLA-ANGST oblik promatranja (eng. Very Large Array – ACS Nearby Galaxy Survey Treasury") i mape obližnjih galaksija kako bi se testirala sposobnost različitih vrijednosti praga gustoće formiranja zvijezda za reprodukciju opaženih mapa. Rezultati studije pokazuju da je s primjenom strojnog učenja u potpunosti moguće u potpunosti iskoristiti informacijski sadržaj slika galaksija i usporediti simulacije promatranja na neparametarski i kvantitativan način.

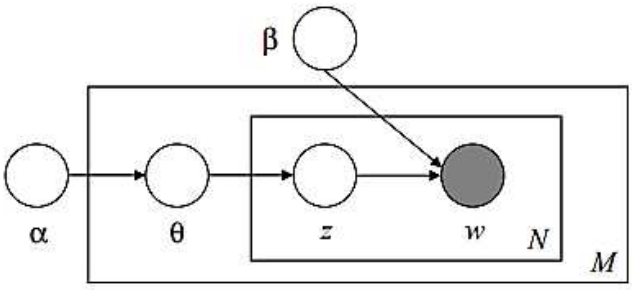
2.6. Određivanje istraživačkih prioriteta u području astronomije primjenom strojnog učenja

Rad se odnosi na ispitivanje mogućnosti utjecaja strojnog učenja na znanstveno strateško planiranje. Pretpostavlja se da pristup temeljen na latentnoj Dirichletovoj raspodjeli korištenjem sažetaka koji su izvučeni iz visokoutjecajnih astronomskih časopisa, može pružiti vodeći pokazatelj budućeg interesa za temu istraživanja.

Latentna Dirichletova alokacija (LDA), generativni je i probabilistički model korpusa. Osnovna ideja modela je da su dokumenti predstavljeni kao nasumične mješavine ponavljajućih latentnih tema, gdje je svaka tema okarakterizirana raspodjelom tzv. ponavljajućih riječi.

LDA pretpostavlja sljedeći generativni proces za svaki dokument *w* u korpusu *D:*

- Odabir N ~ Poisson(ξ)
- Odabir θ ~ Dir(α)
- Za svaki od N riječi wn:
    a) Odabir teme zn ~ Multinom (θ)
    b) Odabir riječi iz skupa p(wn|zn,β), multinomnih vjerojatnosti koje su uvjetovane temom zn.



Slika 2.3.Grafički model LDA [13]

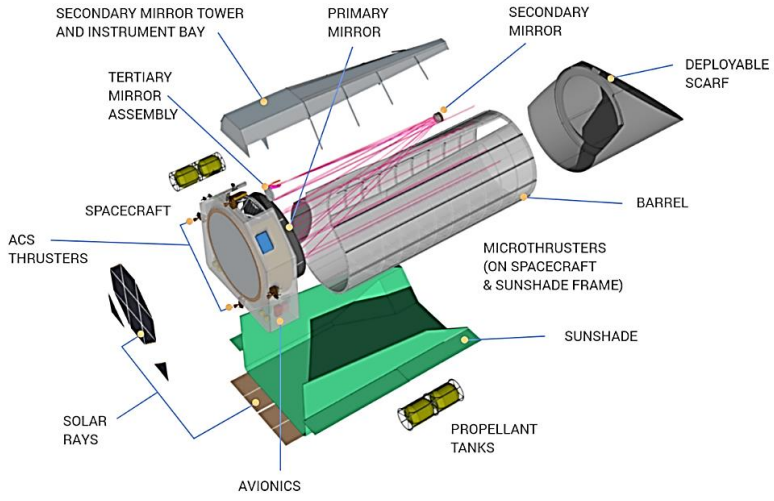
Na slici 2.3., prikazan je grafički model latentne Dirichletove alokacije. Pravokutnici predstavljaju replike cjelokupnog korpusa. Vanjska ploča predstavlja dokumente korpusa, dok unutarnja ploča predstavlja ponovljeni odabir teme i riječi unutar dokumenta. Prikazane su dvije tematske metrije koje su u dobroj korelaciji sa visokoprioritetnim istraživačkim područjima utvrđenim višegodišnjim istraživanjem astronomije i astrofizike državnih akademija iz 2010. godine.

Jedna metrika temelji se na zbroju frakcijskog doprinosa, a druga metrika je godišnja stopa rasta uzorkovanja. Rezultati zaključuju da višegodišnja analiza premalo naglašava brzorastuća istraživanja.

2.7. Klasifikacija boja planeta analognih planetu Zemlji primjenom umjetne inteligencije

Karakterizacija atmosfere direktno snimljenih egzoplaneta trenutno je ograničena na planete divove i tzv. „mini-Neptune“, međutim pri karakterizaciji atmosfere uskoro će se priključiti teleskopi i ostali svemirski koncepti kao što su Origins, HabEx I LUVOIR koji su dizajnirani specifično za karakterizaciju stjenovitih egzoplaneta. Problem spektroskopije planeta koji imaju karakteristike slične našem planetu je vremenski period analize. Implementacijom seta podataka koji se sastoji od 318780 refleksijskih spektara nominalnih egzoplaneta s jednim polumjerom Zemlje, jednom masenom vrijednošću Zemlje i moderne atmosfere, kreiraju se nadzorni algoritmi strojnog učenja za klasifikaciju planeta, koji također uzimaju u obzir i različite omjere signala i šuma.

Metode strojnog učenja sposobne su detekcije postojećih parametara koristeći samo fotometrijski tok reflektirane svjetlosti egzoplaneta sa preciznošću između 50 i 75 posto. Rezultati procjenjuju mogućnost korištenja fotometrijskog toka za početnu identifikaciju parametara na egzoplanetima i ujednačenost između dva kritična rezultata, a to su lažno pozitivne i lažno negativne stope.



Slika 2.4. Teleskop za promatranje zemljolikih planeta (eng. Habitable Exoplanet Observatory) [8]

2.8. SkyNet

SkyNet je učinkovit i robustan način za obuku neuronske mreže za strujno učenje u astronomiji. SkyNet može uvježbati velike i duboke neuralne mreže s naprednim prijenosom podataka uključujući autokodere koji se koriste u širokom rasponu nadziranih i nenadziranih metoda učenja, kao što su npr. regresija, klasifikacija procjena gustoće, grupiranje i smanjenje dimenzionalnosti. SkyNet koristi metodu "preliminarnog treniranja" za dobivanje skupa mrežnih parametara za koje se empirijski pokazalo da su relativno blizu prihvatljivog rješenja, nakon čega slijedi daljnja optimizacija korištenjem Newton-Raphsonove metode, gdje se razina regulacije određuje i prilagođava automatski. Za poboljšanje konvergencije, ali bez potrebe za procjenom ili pohranjivanjem pune Hessian matrice, korištenjem aproksimativne metode za izračunavanje Hessovih vektorskih proizvoda koristi se informacija izvedenice drugog reda. Ova kombinacija metoda omogućuje obuku kompliciranih neuronskih mreža koje je teško optimizirati korištenjem standardnih tehnika povratnog širenja.

SkyNet koristi pravila konvergencije koji prirodno blokiraju prekomjerno nakupljanje podataka te uključuje brzi algoritam za procjenu preciznosti mrežnih izlaza. Korisnost i fleksibilnost ove metode demonstrirana je primjenom na astronomske probleme koji se bave oporavkom strukture od zamućenih i šumnih slika, identifikaciju izbijanja gama zraka te kompresiju i uklanjanje šuma slika galaksija. SkyNet je u potpunosti implementiran u C/C++ programskom paketu i kompletno paraleliziran pomoću MPI sučelja („*message passing interface"*).

2.9. Klasifikacija kvazara primjenom strojnog učenja

Klasifikacija kvazar-zvijezdi u ovom istraživačkom radu izvodi se putem različitih metoda automatizirane klasifikacije koja se temelji na strojnom učenju preko seta podataka koji su implementirani iz "Sloan Digital Sky Survey" knjižnice, u kojem se primarno razlikuju obični tipovi zvijezda od kvazara. Cilj rada je istražiti prikladnost primjene pojedinih metoda strojnog učenja. Rezultati metoda pokazuju najveću učinkovitost asimetričnog AdaBoost algoritma za klasifikaciju fotometrijskih podataka. Rad predstavlja kritički osvrt na postojeću studiju i predlaže primjenu asimetričnog AdaBoost algoritma.

2.9.1. AdaBoost

AdaBoost je model algoritma koji se odnosi na određenu metodu treniranja pojačanog klasifikatora. Pojačani klasifikator je forma klasifikatora sa sljedećim izrazom:

$$F_T(x) = \sum_{t=1}^{T} f_t(x) \quad (2.1.)$$

Gdje je svaki f_t slabi učenik koji uzima objekt x kao ulaz i vraća vrijednost koji označava klasu objekta.

Svaki slabi učenik proizvodi izlaznu hipotezu *h(),* koja predočuje prognozu $h(x_i)$ za svaki uzorak u skupu za obuku. U svakoj iteraciji *t,* odabire se kandidat za učenje i dodjeljuje mu se koeficijent α_t, tako da ukupna greška uvježbavanja E_t od rezultirajućih razina pojačanog klasifikatora bude minimiziran.

$$E_t = \sum_i E[F_{t-1}(x_i) + \alpha_t h(x_i)] \quad (2.2.)$$

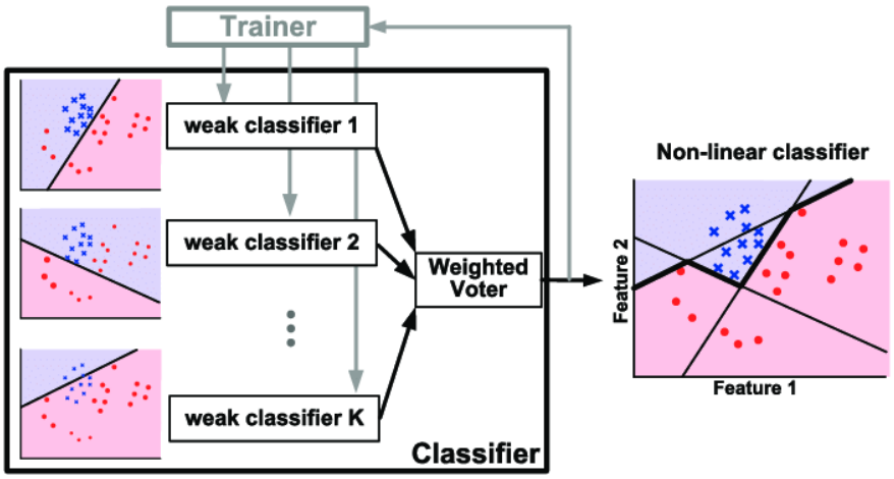
Gdje je:

E_t – Ukupna pogreška uvježbavanja

α_t – Koeficijent iteracije

$h(x_i)$ – Prognoza

$F_{t-1}(x_i)$ – Pojačani klasifikator koji je izgrađen do prethodne faze obuke



Slika 2.5. Prikaz AdaBoost algoritma za izradu pojačanog klasifikatora [14]

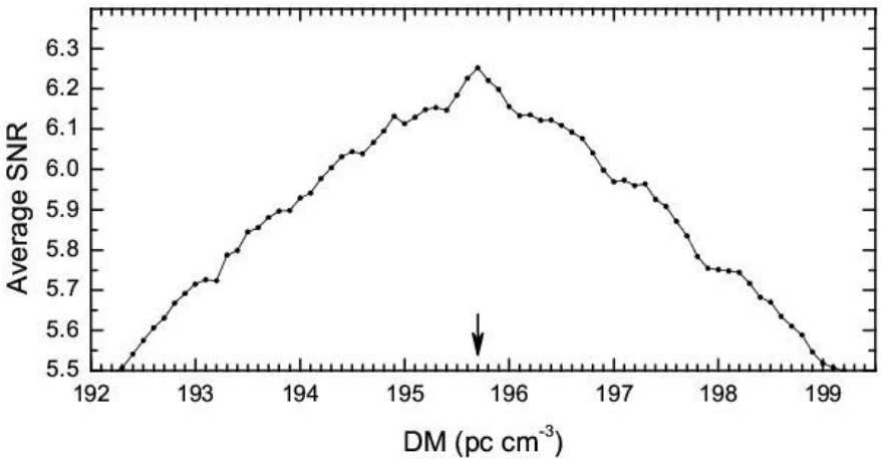
3. METODOLOGIJA

Poglavlje metodologije odnosi se na kategorizaciju pojedinih tema koji pobliže objašnjavaju terminologiju i pojmove koji opisuju proces i osnovu izrade programa za klasifikaciju pulsara. Set podataka koji se koristi za klasifikaciju javno je dostupan i predstavlja određeni broj radio valova koji su detektirani sa Zemlje i broj šumova i smetnji koji se koriste za strojno učenje. Set je analiziran metodom deskriptivne statistike i ujednačen po broju klasa i njegovi će se rezultati prikazati u idućem potpoglavlju. Što se tiče teorijskih pojmova, glavnina bitnih definicija objasnit će se u poglavlju umjetnih neurona koji se odnose na usporedbu umjetnog i biološkog neurona te opis bitnih pojmova koji preciznije definiraju cjelokupnu granu umjetne inteligencije i proces strojnog učenja. Za kraj poglavlja, definiran je nastanak pulsara i njegov način identifikacije te se obrađuje usporedba klasifikacije zvijezda i pulsara.

3.1.Set podataka

Kako bi se višeslojna neuronska mreža mogla primijeniti kod „machine learninga“ potrebno je odabrati pripadajući set podataka. Za potrebe strojnog učenja koristi se set podataka pod nazivom HTRU2, koji opisuje uzorke kandidata za pulsare prikupljene tijekom istraživanja svemira visokom vremenskom rezolucijom. Podaci koji predstavljaju uzorke pulsara odnose se na detektirane radio valove pulsara. Skup podataka sadrži 16259 „lažnih“ primjera signala koji su uzorkovani smetnjama ili šumom i 1639 potvrđenih signala koji su detektirani na Zemlji. Svaki kandidat opisan je sa 8 kontinuiranih varijabli i jednom pripadajućom klasom. Prva skupina varijabli prikazuje jednostavnu statistiku dobivenu iz integriranog profila impulsa. Preostale četiri varijable dobivene su na sličan način, ali iz DM-SNR („data mining streamlit application“) krivulje. Nazivi pojedinih stupaca su:

A. Prosječna vrijednost integriranog profila
B. Standardna devijacija integriranog profila
C. Višak kurtoze integriranog profila
D. Zakrivljenost integriranog profila
E. Srednja vrijednost DM-SNR krivulje
F. Standardna devijacija DM-SNR krivulje
G. Višak kurtoze DM-SNR krivulje
H. Zakrivljenost DM-SNR krivulje
I. Klasa signala



Slika 3.1. Primjer prikaza DM-SNR krivulje [15]

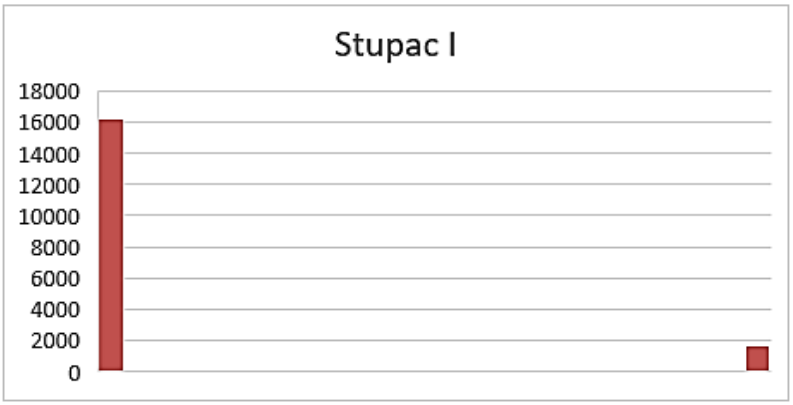
3.1.1. Standardni opis podataka

Standardni opis podataka uvozi izvorni set podataka bez sortiranja njegovih atributa. Dok vrijednosti prvih 8 stupaca variraju ovisno o njihovoj devijaciji i zakrivljenosti, posljednji stupac predstavlja klasu signala kroz binarnu vrijednost 0 ili 1. U nastavku je prikazan dio tablice korištenog seta podataka HTRU2.

	A	B	C	D	E	F	G	H	I
1	140.5625	55.68378	-0.23457	-0.69965	3.199833	19.11043	7.975532	74.24222	0
2	102.5078	58.88243	0.465318	-0.51509	1.677258	14.86015	10.57649	127.3936	0
3	103.0156	39.34165	0.323328	1.051164	3.121237	21.74467	7.735822	63.17191	0
4	136.75	57.17845	-0.06841	-0.63624	3.642977	20.95928	6.896499	53.59366	0
5	88.72656	40.67223	0.600866	1.123492	1.17893	11.46872	14.26957	252.5673	0
6	93.57031	46.69811	0.531905	0.416721	1.636288	14.54507	10.62175	131.394	0
7	119.4844	48.76506	0.03146	-0.11217	0.999164	9.279612	19.20623	479.7566	0
8	130.3828	39.84406	-0.15832	0.38954	1.220736	14.37894	13.53946	198.2365	0
9	107.25	52.62708	0.452688	0.170347	2.33194	14.48685	9.001004	107.9725	0
10	107.2578	39.49649	0.465882	1.162877	4.079431	24.98042	7.39708	57.78474	0
11	142.0781	45.28807	-0.32033	0.283953	5.376254	29.0099	6.076266	37.83139	0
12	133.2578	44.05824	-0.08106	0.115362	1.632107	12.00781	11.97207	195.5434	0
13	134.9609	49.55433	-0.1353	-0.08047	10.69649	41.34204	3.893934	14.13121	0
14	117.9453	45.50658	0.325438	0.661459	2.83612	23.11835	8.943212	82.47559	0
15	138.1797	51.52448	-0.03185	0.046797	6.330268	31.57635	5.15594	26.14331	0
16	114.3672	51.94572	-0.0945	-0.28798	2.738294	17.19189	9.050612	96.6119	0
17	109.6406	49.01765	0.137636	-0.2567	1.508361	12.0729	13.36793	223.4384	0
18	100.8516	51.74352	0.393837	-0.01124	2.841137	21.63578	8.302242	71.58437	0
19	136.0938	51.691	-0.04591	-0.27182	9.342809	38.0964	4.345438	18.67365	0
20	99.36719	41.5722	1.547197	4.154106	27.55518	61.71902	2.208808	3.66268	1
21	100.8906	51.89039	0.627487	-0.0265	3.883779	23.04527	6.953168	52.27944	0
22	105.4453	41.13997	0.142654	0.32042	3.551839	20.75502	7.739552	68.51977	0
23	95.86719	42.05992	0.326387	0.803502	1.832776	12.24897	11.24933	177.2308	0
24	117.3672	53.90861	0.257953	-0.40505	6.018395	24.76612	4.807783	25.52262	0
25	106.6484	56.36718	0.378355	-0.26637	2.436455	18.40537	9.37866	96.86023	0
26	112.7188	50.30127	0.279391	-0.12901	8.281773	37.81001	4.691827	21.27621	0
27	130.8516	52.43286	0.142597	0.018885	2.646321	15.65444	9.464164	115.6732	0
28	119.4375	52.87482	-0.00255	-0.46036	2.365385	16.49803	9.008352	94.75566	0

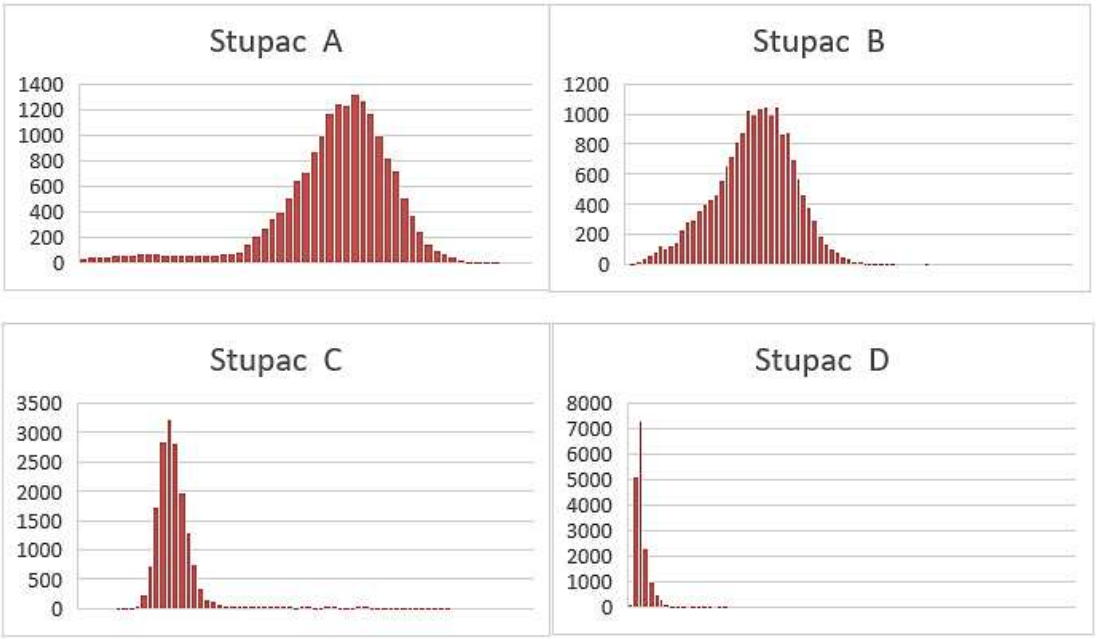
Slika 3.2. Primjer sadržaja HTRU2 seta podataka

Kako bi se olakšao prikaz pojedinih atributa od stupaca seta podataka, svaki stupac prikazan je kroz dijagram koji pobliže prikazuje varijabilne vrijednosti pojedinih redaka. U nastavku su prikazani dijagrami za svaki pojedini stupac.



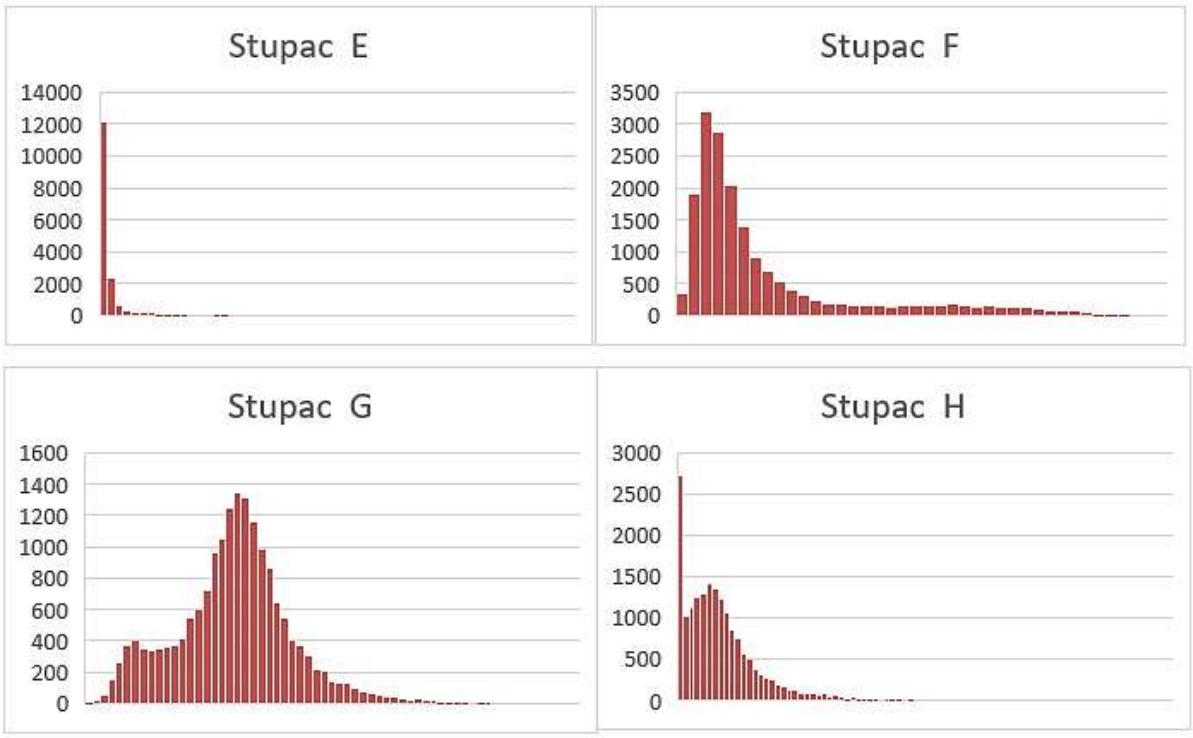
Slika 3.3. Dijagram vrijednosti seta podataka za stupac I

Slika 3.3. prikazuje posljednji stupac iz seta podataka. Podaci iz stupca „I“ daju najjednostavniji prikaz dijagrama (za razliku od ostalih stupaca) i determinira klasu signala po binarnoj vrijednosti 0 ili 1. Sortiranjem podatka potvrđene su prethodno navedene vrijednosti od 1639 signala detektiranih radio valova i 16259 signala koji su dobiveni šumom i smetnjama.



Slika 3.4. Dijagram vrijednosti seta podataka integriranih profila

Na slici 3.4. prikazana je grupa dijagrama koja se odnosi na vrijednosti integriranog profila. Osi ordinata na dijagramima prikazuju koliki broj ulaznih atributa postoji za određenu vrijednost za pojedine stupce. Određena vrijednost varira te će u nastavku poglavlja biti prikazana vrijednost minimalnog i maksimalnog atributa kroz osnovnu analizu.



Slika 3.5. Dijagram vrijednosti seta podataka DM-SNR krivulje

Slično kao i sa slike 3.4., slika 3.5. prikazuje grupu dijagrama čiji se u ovom slučaju podaci odnose na kategorije vrijednosti koje su očitane iz DM-SNR vrijednosti. Vrijednosti ne determiniraju precizno ponašanje detektiranih radio valova s obzirom da vrijednosti variraju u njihovim specifičnim domenama. Iz tog razloga, potrebno je sortirati i korigirati broj atributa, kako bi se dobio što ujednačeniji broj izlaznih atributa, što će biti učinjeno u nastavku poglavlja.

3.1.2. Analiza osnovnih svojstava seta podataka

Kako bi se bolje pojasnio korišteni set podataka, potrebno je analizirati pojedine atribute i njihove korelacije sa ostalim atributima korištenjem različitih funkcija koje se primijenjuju u programskom paketu „Microsoft Excel“. Sami set podataka pohranjen je kao .csv datoteka.

U ovom potpoglavlju prikazuje se analiza osnovnih svojstava seta podataka. U osnovnu analizu spadaju prikaz najveće i najmanje vrijednosti varijabli, prosječna vrijednost pojedinog stupca i njihova standardna devijacija.

Najveća vrijednost pojedinih stupaca pronalazi se na način da se u tzv. „formula bar“ upisuje funkcija =MAX(IME_STUPCA:IME_STUPCA). U nastavku su prikazani rezultati karakteristične funkcije za svaki stupac.

Tablica 3.1.Najveće vrijednosti za pojedine stupce

Najveće vrijednosti (=MAX)			
INTEGRIRANI PROFIL		DM-SNR KRIVULJA	
A	192.6712	**E**	223.3921
B	98.77891	**F**	110.6422
C	8.069522	**G**	34.53984
D	68.10162	**H**	1191.001
	KLASA SIGNALA		
	I	1	

Slično kao i kod pretraživanja najveće vrijednosti, način pronalaska najmanje vrijednosti je upisivanje funkcije =MIN(IME_STUPCA:IME_STUPCA). U nastavku su prikazani dobiveni rezultati.

Tablica 3.2. Najmanje vrijednosti za pojedine stupce

Najmanje vrijednosti (=MIN)			
INTEGRIRANI PROFIL		DM-SNR KRIVULJA	
A	5.8125	**E**	0.213211
B	24.77204	**F**	7.370432
C	-1.87601	**G**	-3.13927
D	-1.79189	**H**	-1.97698
	KLASA SIGNALA		
	I	0	

Treća funkcija koja se analizira u ovom potpoglavlju odnosi se na prosječnu vrijednost pojedinih stupaca.Ona se dobija na način da se kod unosa funkcije upiše =AVERAGE (IME_STUPCA:IME_STUPCA).

Tablica 3.3. Prosječne vrijednosti za pojedine stupce

Prosječne vrijednosti (=AVERAGE)			
INTEGRIRANI PROFIL		DM-SNR KRIVULJA	
A	111.08	**E**	12.6144
B	46.54953	**F**	26.32651
C	0.477857	**G**	8.303556
D	1.770279	**H**	104.8577
	KLASA SIGNALA		
	I	0.091	

Posljednja analiza osnovnih svojstava odnosi se na standardnu devijaciju. Standardna devijacija odnosi se na prosječno srednje kvadratno odstupanje numeričkih vrijednosti neke veličine od aritmetičke veličine. Ona se računa na način da se unosi funkcija =STDEV.S (IME_STUPCA :IME_STUPCA).

Tablica 3.4. Standardna devijacija za pojedine stupce

Standardna devijacija (=STDEV.S)			
INTEGRIRANI PROFIL		DM-SNR KRIVULJA	
A	25.65293536	**E**	29.47289715
B	6.84318941	**F**	19.47057233
C	1.064039716	**G**	4.506091859
D	6.167913248	**H**	106.5145395
	KLASA SIGNALA		
	I	0.288432385	

3.1.3. Deskriptivna statistika seta podataka

Deskriptivna statistika (eng. *„descriptive statistics"*) grana je statistike koje se odnosi na organizaciju učitanih podataka i njihov kratki opis. Organizacija podataka vrši se pomoću numeričkih i grafičkih prikaza. Prikaz deskriptivne statistike u „Microsoft Excelu" vrši se na idući način: U alatnoj traci odabire se opcija „*Data*". Potom se odabire opcija „*Data analysis*". Ova točka nudi razni spektar različitih analiza podataka, pa tako i deskriptivnu statistiku. Zadnji korak je odabiranje domene (u ovom slučaju cijeli stupac sloja). Ovim načinom dobiju se idući podaci za pojedine stupce:

Tablica 3.5. Deskriptivna statistika za stupce A,B,C i D

Stupac A		***Stupac B***		***Stupac C***		***Stupac D***	
Mean	111.08	Mean	46.54953	Mean	0.477857	Mean	1.770279
Standard Error	0.19175	Standard Error	0.051151	Standard Error	0.007953	Standard Error	0.046104
Median	115.0781	Median	46.94748	Median	0.22324	Median	0.19871
Mode	106.7109	Mode	49.50029	Mode	0.001934	Mode	#N/A
Standard Deviation	25.65294	Standard Deviation	6.843189	Standard Deviation	1.06404	Standard Deviation	6.167913
Sample Variance	658.0731	Sample Variance	46.82924	Sample Variance	1.132181	Sample Variance	38.04315
Kurtosis	2.972374	Kurtosis	1.689571	Kurtosis	14.63974	Kurtosis	30.16648
Skewness	-1.37519	Skewness	0.126641	Skewness	3.63841	Skewness	5.181293
Range	186.8047	Range	74.00687	Range	9.945533	Range	69.89351
Minimum	5.8125	Minimum	24.77204	Minimum	-1.87601	Minimum	-1.79189
Maximum	192.6172	Maximum	98.77891	Maximum	8.069522	Maximum	68.10162
Sum	1988109	Sum	833143.5	Sum	8552.689	Sum	31684.45
Count	17898	Count	17898	Count	17898	Count	17898

Deskriptivnom statistikom utvrđeni su detaljni podaci za svaki stupac integriranog profila (tablica 3.5.), koji otkrivaju preciznije raspone podataka, kao i njihove karakteristike kroz cijelu domenu podataka u stupcu. Za razliku od dijagrama vrijednosti, koje nisu u mogućnosti jasno predočiti os apcise i njihove podatke iz jednostavnog razloga što se u obzir uzimaju sve varijable stupaca tj, njih 17898, deskriptivnom statistikom izračunata je njihova domena i standardna pogreška, koja će pri izradi programa za strojno učenje dodatno olakšati i poboljšati postotak uspješnosti klasifikacije pulsara. Također, organizacijom podataka uz pomoć deskriptivne statistike, vidljivo je kako se vrijednosti minimalnih i maksimalnih vrijednosti, kao i njihove prosječne vrijednosti i standardne devijacije podudaraju sa analizom iz prethodnih tablica.

Tablica 3.6. Deskriptivna statistika za slojeve E,F,G i H

Stupac E		***Stupac F***		***Stupac G***		***Stupac H***	
Mean	12.6144	Mean	26.32651	Mean	8.303556	Mean	104.8577
Standard Error	0.220303	Standard Error	0.145538	Standard Error	0.033682	Standard Error	0.796172
Median	2.801839	Median	18.46132	Median	8.433515	Median	83.06456
Mode	2.332776	Mode	7.370432	Mode	34.53984	Mode	1191.001
Standard Deviation	29.4729	Standard Deviation	19.47057	Standard Deviation	4.506092	Standard Deviation	106.5145
Sample Variance	868.6517	Sample Variance	379.1032	Sample Variance	20.30486	Sample Variance	11345.35
Kurtosis	14.06472	Kurtosis	2.825997	Kurtosis	1.526209	Kurtosis	13.49411
Skewness	3.683302	Skewness	1.894254	Skewness	0.441501	Skewness	2.734514
Range	223.1789	Range	103.2718	Range	37.67911	Range	1192.978
Minimum	0.213211	Minimum	7.370432	Minimum	-3.13927	Minimum	-1.97698
Maximum	223.3921	Maximum	110.6422	Maximum	34.53984	Maximum	1191.001
Sum	225772.5	Sum	471192	Sum	148617	Sum	1876743
Count	17898	Count	17898	Count	17898	Count	17898

Tablica 3.6. prikazuje istu skupinu podataka kao i prethodna tablica, ali u ovom slučaju za skupinu seta podataka koji se odnose na vrijednosti učitane iz DM-SNR krivulje. Deskriptivna statistika za stupac „I“ nije potrebna zbog već poznatog raspona koji predstavlja klasu signala.

3.1.4. Korelacija između stupaca

U posljednjem segmentu analize seta podatka, važno je prikazati pripadajuću korelaciju između pojedinih stupaca. Svaka vrijednost u tablici prikazuje određenu korelaciju stupaca prema drugim stupcima. Način dobijanja korelacije izvršava se funkcijom =CORREL(A:A, $I:$I).

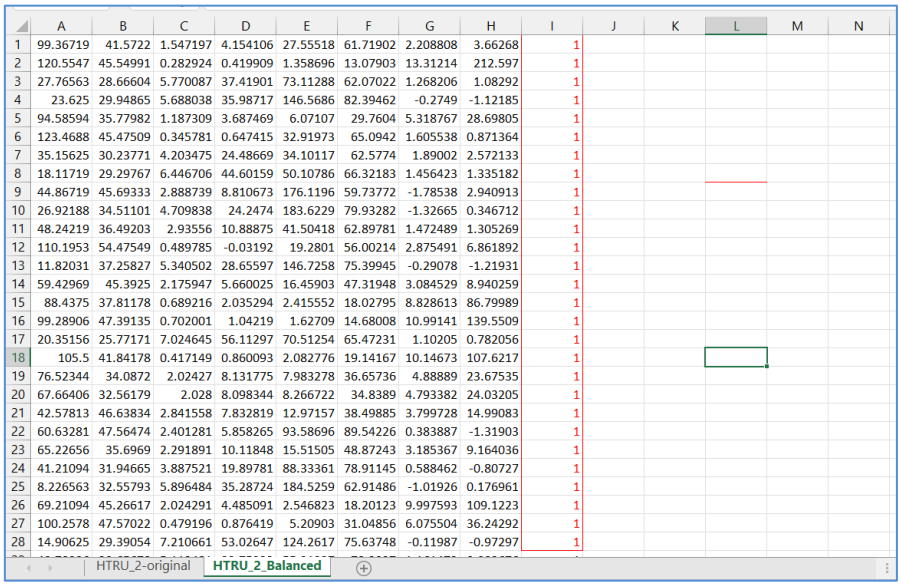
Tablica 3.7. Korelacija između pojedinih stupaca

Korelacija (=CORREL)			
INTEGRIRANI PROFIL		DM-SNR KRIVULJA	
A	0.67318	**E**	0.400876
B	-0.36371	**F**	0.491535
C	0.791591	**G**	-0.39082
D	0.709528	**H**	-0.25912
	KLASA SIGNALA		
	I	1	

Kako bi se ujednačio odnos varijabli iz posljednjeg stupca, koji u ovoj verziji seta podataka dominiraju klasom vrijednosti 0, odnosno vrijednostima signala šuma i smetnji, potrebno je organizirati set podataka na način kako bi broj izlaza sa vrijednosti 0 i broj izlaza sa vrijednosti 1 bio jednak. U idućem potpoglavlju obrađena je analiza seta podataka sa ujednačenim odnosom izlaznih atributa.

3.1.5. Ujednačeni set podataka

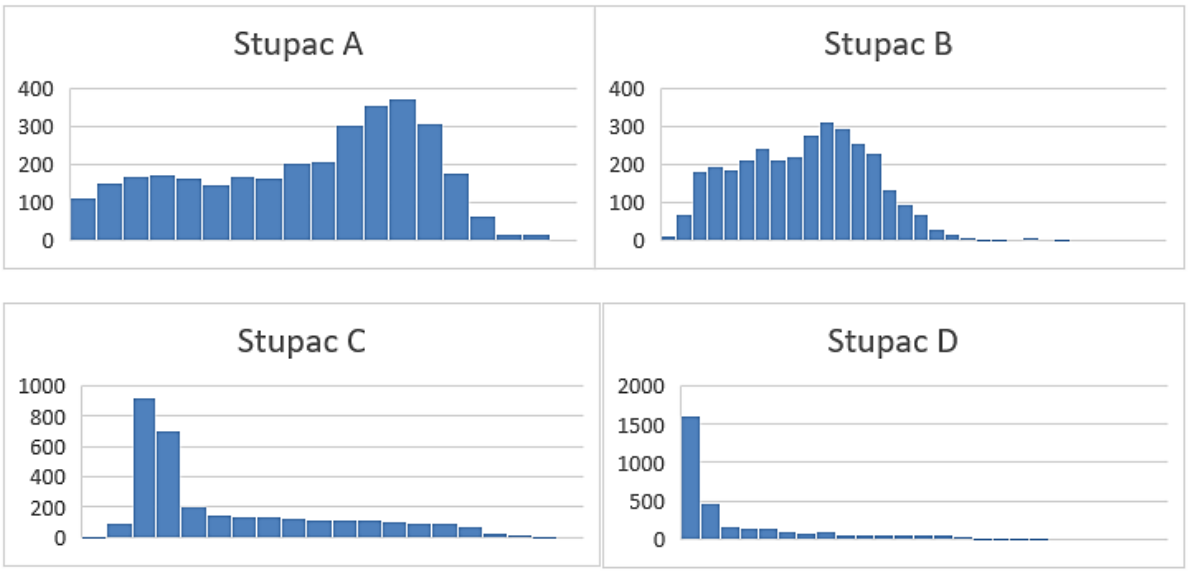
U ovoj verziji seta podataka značajno se izmijenio broj redaka korištenog seta podataka. Nakon sortiranja posljednjeg stupca „I“ , broj atributa smanjio se sa 17898 na 3280 atributa. Analiza ovog seta podataka biti će sukladna prethodnom setu podataka. U nastavku je prikazan ujednačeni set podataka, s naznakom na posljednji stupac u kojem su prikazane samo vrijednosti 1.



	A	B	C	D	E	F	G	H	I
1	99.36719	41.5722	1.547197	4.154106	27.55518	61.71902	2.208808	3.66268	1
2	120.5547	45.54991	0.282924	0.419909	1.358696	13.07903	13.31214	212.597	1
3	27.76563	28.66604	5.770087	37.41901	73.11288	62.07022	1.268206	1.08292	1
4	23.625	29.94865	5.688038	35.98717	146.5686	82.39462	-0.2749	-1.12185	1
5	94.58594	35.77982	1.187309	3.687469	6.07107	29.7604	5.318767	28.69805	1
6	123.4688	45.47509	0.345781	0.647415	32.91973	65.0942	1.605538	0.871364	1
7	35.15625	30.23771	4.203475	24.48669	34.10117	62.5774	1.89002	2.572133	1
8	18.11719	29.29767	6.446706	44.60159	50.10786	66.32183	1.456423	1.335182	1
9	44.86719	45.69333	2.888739	8.810673	176.1196	59.73772	-1.78538	2.940913	1
10	26.92188	34.51101	4.709838	24.2474	183.6229	79.93282	-1.32665	0.346712	1
11	48.24219	36.49203	2.93556	10.88875	41.50418	62.89781	1.472489	1.305269	1
12	110.1953	54.47549	0.489785	-0.03192	19.2801	56.00214	2.875491	6.861892	1
13	11.82031	37.25827	5.340502	28.65597	146.7258	75.39945	-0.29078	-1.21931	1
14	59.42969	45.3925	2.175947	5.660025	16.45903	47.31948	3.084529	8.940259	1
15	88.4375	37.81178	0.689216	2.035294	2.415552	18.02795	8.828613	86.79989	1
16	99.28906	47.39135	0.702001	1.04219	1.62709	14.68008	10.99141	139.5509	1
17	20.35156	25.77171	7.024645	56.11297	70.51254	65.47231	1.10205	0.782056	1
18	105.5	41.84178	0.417149	0.860093	2.082776	19.14167	10.14673	107.6217	1
19	76.52344	34.0872	2.02427	8.131775	7.983278	36.65736	4.88889	23.67535	1
20	67.66406	32.56179	2.028	8.098344	8.266722	34.8389	4.793382	24.03205	1
21	42.57813	46.63834	2.841558	7.832819	12.97157	38.49885	3.799728	14.99083	1
22	60.63281	47.56474	2.401281	5.858265	93.58696	89.54226	0.383887	-1.31903	1
23	65.22656	35.6969	2.291891	10.11848	15.51505	48.87243	3.185367	9.164036	1
24	41.21094	31.94665	3.887521	19.89781	88.33361	78.91145	0.588462	-0.80727	1
25	8.226563	32.55793	5.896484	35.28724	184.5259	62.91486	-1.01926	0.176961	1
26	69.21094	45.26617	2.024291	4.485091	2.546823	18.20123	9.997593	109.1223	1
27	100.2578	47.57022	0.479196	0.876419	5.20903	31.04856	6.075504	36.24292	1
28	14.90625	29.39054	7.210661	53.02647	124.2617	75.63748	-0.11987	-0.97297	1

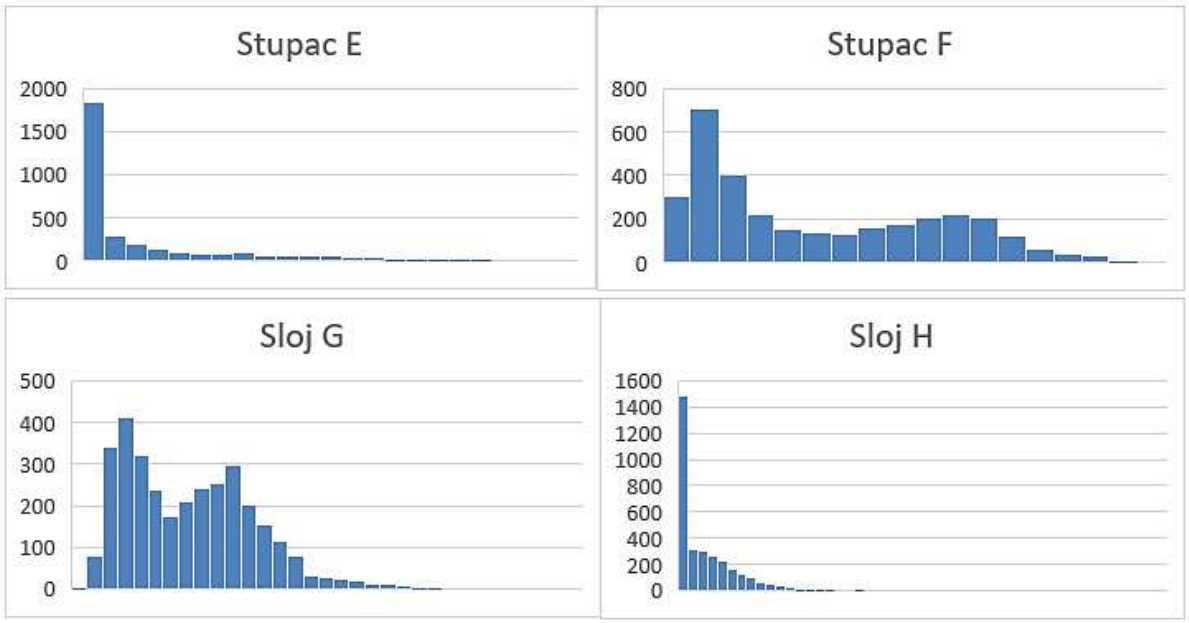
HTRU_2-original | HTRU_2_Balanced

Slika 3.6. Ujednačeni set podataka



Slika 3.7. Dijagram vrijednosti ujednačenog seta podataka integriranog profila

Usporedbom dijagrama standardnog i ujednačenog seta podataka integriranog profila, može se primijetiti kako se broj podataka drastično smanjio i kako su se prosječne vrijednosti, vrijednosti standardnih devijacija i viška kurtoze ujednačenog seta podataka smanjile za razliku od standardnog HTRU2 seta podataka.



Slika 3.8. Dijagram vrijednosti ujednačenog seta podataka učitanih iz DM-SNR krivulje

Slika 3.9. Dijagram vrijednosti ujednačenog seta podataka klase signala

Na slici 3.9., vidljivo je kako je sortiranje prethodnog seta podataka bio uspješan, te kako je odnos izlaznih atributa sa vrijednosti 0 i 1 ujednačen. U nastavku su prikazani rezultati analiza osnovnih svojstava te njihove deskriptivne statistike (karakteristične funkcije za pojedine tablice identične su onima iz prethodne analize).

Tablica 3.8. Osnovna svojstva podataka ujednačenog seta podataka

STUPAC	MAX	MIN	AVERAGE	ST.DEV
A	171.836	5.812	86.671	38.758
B	86.951	24.772	43.086	8.512
C	8.069	-1.137	1.668	1.984
D	68.101	-1.677	7.953	40.293
E	211.948	0.289	27.716	40.293
F	109.655	7.449	38.997	24.177
G	33.489	-2.557	5.909	4.743
H	1140.353	-1.875	66.299	96.333
I	1	0	0.499	0.500

Postupak analize osnovnih svojstava ujednačenog seta podataka identičan je kao i kod prethodne analize standardnog seta podataka HTRU2. Podaci sa osnovnim svojstvima pojedinih stupaca nalazi se u tablici 3.8. (Vrijednosti su zaokružene na tri decimale).

Slično kao i kod standardnog seta podataka, način izvršavanja deskriptivne statistike za ujednačeni set izvršava se preko opcije “Data analysis” čime se ispisuju opisne karakteristike zasebnih stupaca. Najveća razlika pri usporedbi deskriptivnih statistika je podatak o broju

varijabli pojedinih stupaca koji se smanjio na 3280, i time potvrdio ujednačenost klasa signala u ovoj verziji seta podataka.

Tablica 3.9. Deskriptivna statistika za pojedine stupce ujednačenog seta podataka

Stupac A		***Stupac B***		***Stupac C***		***Stupac D***	
Mean	86.67066	Mean	43.08647	Mean	1.66837	Mean	7.95302
Standard Error	0.676743	Standard Error	0.148634	Standard Error	0.034649	Standard Error	0.218106
Median	96.15625	Median	43.6039	Median	0.636421	Median	1.290728
Mode	29.6875	Mode	40.51519	Mode	#N/A	Mode	#N/A
Standard Deviation	38.75796	Standard Deviation	8.512454	Standard Deviation	1.984376	Standard Deviation	12.49119
Sample Variance	1502.179	Sample Variance	72.46187	Sample Variance	3.937748	Sample Variance	156.0299
Kurtosis	-0.97915	Kurtosis	0.277737	Kurtosis	0.005914	Kurtosis	2.655811
Skewness	-0.39129	Skewness	0.309197	Skewness	1.086174	Skewness	1.803411
Range	166.0234	Range	62.17935	Range	9.206535	Range	69.77835
Minimum	5.8125	Minimum	24.77204	Minimum	-1.13701	Minimum	-1.67672
Maximum	171.8359	Maximum	86.9514	Maximum	8.069522	Maximum	68.10162
Sum	284279.8	Sum	141323.6	Sum	5472.253	Sum	26085.9
Count	3280	Count	3280	Count	3280	Count	3280

Stupac E		***Stupac F***		***Stupac G***		***Stupac H***	
Mean	27.71623	Mean	38.99691	Mean	5.909953	Mean	66.29963
Standard Error	0.703545	Standard Error	0.42215	Standard Error	0.082827	Standard Error	1.682064
Median	6.123746	Median	30.86544	Median	5.460473	Median	30.85634
Mode	1.610368	Mode	#N/A	Mode	#N/A	Mode	#N/A
Standard Deviation	40.29292	Standard Deviation	24.17708	Standard Deviation	4.74361	Standard Deviation	96.33394
Sample Variance	1623.52	Sample Variance	584.531	Sample Variance	22.50183	Sample Variance	9280.229
Kurtosis	2.489726	Kurtosis	-1.04816	Kurtosis	1.092584	Kurtosis	18.33811
Skewness	1.811237	Skewness	0.535661	Skewness	0.848782	Skewness	3.28738
Range	211.6597	Range	102.2065	Range	36.04655	Range	1142.228
Minimum	0.289298	Minimum	7.448817	Minimum	-2.5568	Minimum	-1.87468
Maximum	211.949	Maximum	109.6553	Maximum	33.48975	Maximum	1140.353
Sum	90909.23	Sum	127909.9	Sum	19384.65	Sum	217462.8
Count	3280	Count	3280	Count	3280	Count	3280

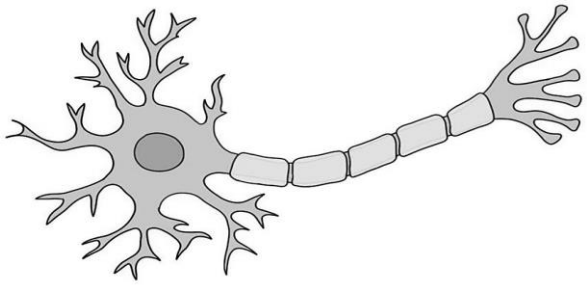
Za kraj analize ujednačenog seta podataka, prikazuje se korelacija zasebnih stupaca sa ostalim stupcima. Postupak dobivanja korelacije ostaje isti kao i kod korelacije standardnog seta podataka. Rezultati korelacije nalaze se u tablici 3.10.

Tablica 3.10. Korelacija između pojedinih stupaca ujednačenog seta podataka

Korelacija (=CORREL)			
INTEGRIRANI PROFIL		DM-SNR KRIVULJA	
A	-0.773166194	**E**	0.548475
B	-0.513819858	**F**	0.72234
C	0.736562639	**G**	-0.66436
D	0.608194951	**H**	-0.50186
	KLASA SIGNALA		
	I	1	

3.2. Umjetne neuronske mreže

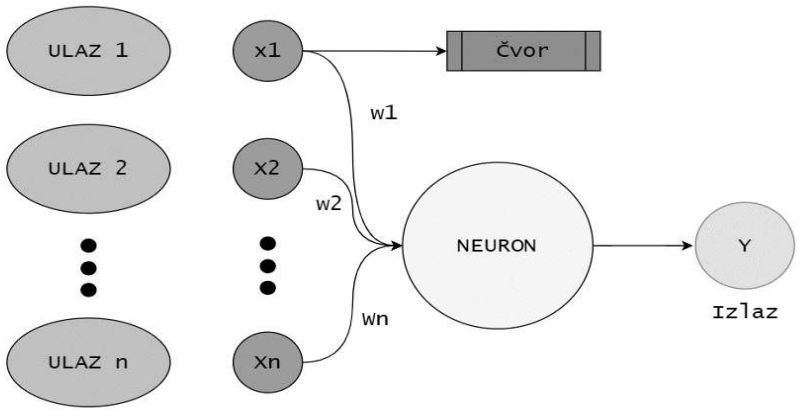
Pojam umjetnih neuronskih mreža izveden je iz bioloških neuronskih mreža koje čine strukturu ljudskog mozga. Slično principu ljudskog mozga koji se sastoji od neurona koji su međusobno povezani, umjetne neuronske mreže također adaptiraju i primjenjuju koncept povezanih neuronskih mreža u različitim slojevima mreže. Na slikama 3.10. i 3.11., prikazani su modeli biološke i umjetne neuronske mreže s njihovim dijelovima.



Slika 3.10. Model biološke neuronske mreže [15]

Glavni dijelovi biološke neuronske mreže su dendriti, aksoni, sinapse i stanična jezgra. Uspoređujući dijelove biološke neuronske mreže i umjetne neuronske mreže, može se determinirati poveznica između dvije mreže. Dendriti iz biološke mreže predstavljaju ulazne informacije koji se implementiraju na umjetnoj mreži, stanična jezgra koja u teoriji prenosi staničnu genetsku informaciju predstavlja se kao čvor, sinapse u umjetnoj neuronskoj mreži

prezentiraju se kao težinski faktori koji determiniraju jakost povezanosti umjetne mreže, a aksoni predstavljaju izlaze u mreži.



Slika 3.11. Model umjetne neuronske mreže

Cilj umjetne neuronske mreže u području umjetne inteligencije je pokušaj „oponašanja“ mreže neurona koji čine ljudski mozak tako da računala imaju svojstvo razumijevanja stvari i donošenja odluka na način kako to čine ljudska bića. Umjetna neuronska mreža dizajnirana je kreiranjem programa koji se implementira na računalo tako da računalo stekne mogućnost ponašanja poput povezanih moždanih stanica.

Ljudski mozak približno sadrži oko 1000 milijardi (trilijun) neuronskih stanica. Svaki neuron zasebno sadrži oko 1000 do 100 000 asocijacijskih točaka. U ljudskom mozgu podaci su pohranjeni na način da se oni distribuiraju, a mogućnost čovjeka je paralelno izdvajanje jednog dijela tih podataka iz memorije kada je god to potrebno.

Umjetna neuronska mreža može se prikazati preko primjera digitalnih logičkih vrata koji primaju i daju ulazne i izlazne varijable. Vrata „ILI“ (OR) primaju dva ulaza. Ako jedan ili oba ulaza poprimaju vrijednost 1 ili ON, tada se na izlazu prikazuje varijabla 1 (ON). Ako su oba ulaza isključena i poprimaju vrijednost 0, tada izlaz daje vrijednost 0 (OFF). Iz tog primjera može se zaključiti kako izlazna varijabla ovisi o ulaznoj varijabli. Taj slučaj ne odnosi se na ljudski mozak, čiji se odnos ulaza i izlaza konstantno mijenja, iz razloga što neuroni u mozgu kontinuirano „uče“.

Osobine umjetnih neuronskih mreža su:

- Svojsvo dobre procjene nelinearnosti
- Mogućnost rada sa nejasnim ili nepotpunim podacima

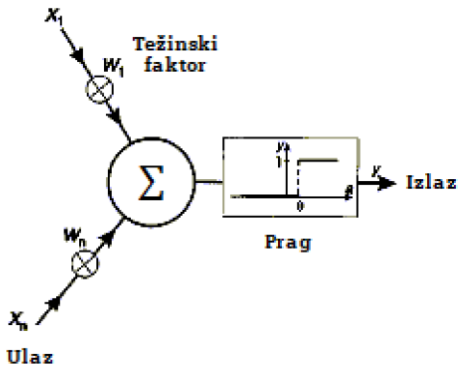
- Robusnost na pogreške u podacima
- Rad sa velikim brojem parametara i setova podataka
- Mogućnost prilagodnje na okolinu

Svojstva umjetnih neuronskih mreža su:

- Nelinearnost
- Preslikavanje ulaz-izlaz (učenje s nadzorom)
- Adaptivnost
- Tolerancija na pogreške
- VLSI implementacija

3.2.1. Umjetni neuron (Threshold logic unit, TLU)

Umjetni neuron osnova je umjetne neuronske mreže koji se najčešće prikazuje kao McCulloch-Pitts model, koji je prikazan na slici 3.12. Ovaj model je prvi računalni model neurona koji je predložen od strane neuroznanstvenika Warrena MuCullocha i logičara Waltera Pittsa 1943. godine.



Slika 3.12. McCulloch-Pitts model

Ulazne varijable skupljaju se u više ulaznih signala. Svaki ulazni signal množi se s težinskim faktorom za opskrbu težinskog signala. Nakon tog procesa, težinski se signal uvodi u sumacijsku točku, gdje se sve prethodne vrijednosti zbrajaju i gdje se formira signal koji će biti implementiran u ulaz aktivacijske funkcije. Logička jedinica praga uspoređuje ulaznu aktivacijsku funkciju sa vrijednošću praga. Ako je ta vrijednost prešla vrijednost praga, na

izlazu se generira vrijednost 1, a ako vrijednost ne prelazi vrijednost praga, generira se izlaz 0. Izlazne linije distribuiraju izlazni signal na više pozicija.

3.2.1.1. Ulaz

Svaki „threshold logic unit“ može poprimiti bilo koji broj ulaza. Svaki ulaz prikuplja signal ili izvan mreže ili iz drugih jedinice unutar preže preko sinapse. Obično je tip signala Booleov (Booleov signal najčešći je način pregledavanja učitanih podataka, omogućuju pregled logičke razine jednog kanala u bilo kojem trenutku u vremenu), i uzima samo vrijednosti 1, -1, ili 0.

Jedan ulaz odgovara dovodnoj sinapsi na dendritskom stablu biološkog neurona, a signal odgovara aktivaciji ili deaktivaciji sinapse unutar određenog vremenskog razdoblja. Broj ulaza definira dimenzionalnost prostora stanja unutar kojeg neuron operira (jedna dimenzija za svaki ulaz i jedna za determiniranu vrijednost praga). Vrijednosti ulaza definiraju točku unutar tog prostora.

3.2.1.2. Težinska funkcija

Umjetni neuron sadrži isti broj težinskih funkcija koliko ima ulaza. Između svake funkcije i ulaza vrijedi odnos 1:1. Funkcija težine uvećava (množi) ulazni signal s vrijednošću težinskog faktora prije vraćanja rezultata u sumacijsku točku. Vrijednost težine ulaza u jednostavnom umjetnom neuronu podudara se s elektrokemijskim svojstvima putanja u kristalnom stablu biološkog neurona. U biološkom neuronu ti putovi vode do pridružene ulazne sinapse do aksonskog brežuljka. Vrijednosti težine definiraju orijentaciju hiper ravnine koja raspolavlja prostor definiran ulazima.

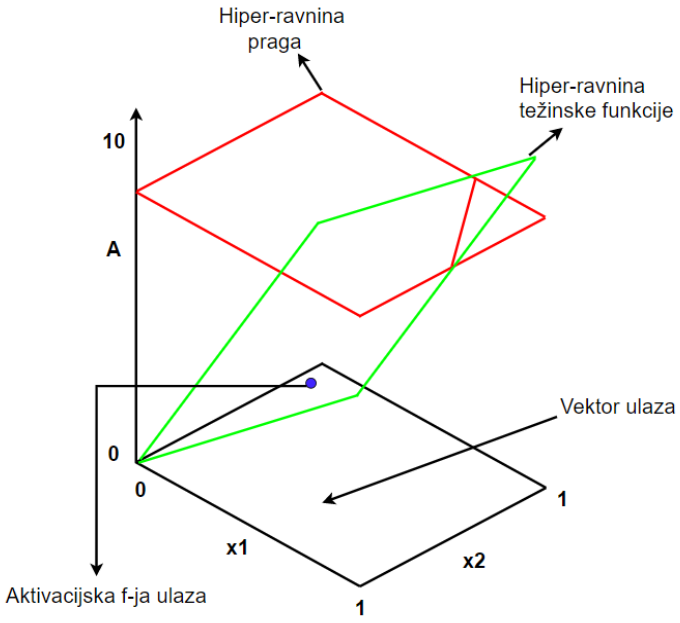
3.2.1.3.Sumacijska točka

Jedinka „threshold logic unit-a“ u sebi sadrži jednu sumacijsku točku koja ima funkciju zbrajanja svih težinskih funkcija prije nego što aktivacijska funkcija vrati izlaznu varijablu. Sumacijska točka odgovara konvergenciji kristalnog stabla u jedinki i integraciji zadanog signala tijekom vremena sve dok ne stigne do aksonskog brežuljka. Funkcije zbrajanja, zajedno sa težinskim funkcijama, projiciraju točku definiranu vrijednostima ulaza duž osi praga na hiper ravninu definiranu vrijednostima težine.

3.2.1.4.Funkcija praga (aktivacijska funkcija) i izlaz

Svaki umjetni neuron sadrži jednu aktivacijsku funkciju koja u „*threshold logic unit-u*“ predstavlja funkciju praga. Vrijednost zbroja uspoređuje se s vrijednošću praga. Ako je zbroj veći od praga, funkcija poprima vrijednost 1. U protivnom, kada zbroj ne dosegne vrijednost veću od praga, funkcija uzima vrijednost 0.

Aktivacijska funkcija odgovara skupljanju akcijskog potencijala na aksonskom brežuljku. Što je veći potencijal potreban za ispuštanje signala niz akson, to je prag neurona viši.



Slika 3.13. Usporedba hiper-ravnina u THU

Na slici 3.13. prikazana je usporedba pozicija hiper ravnina u „*threshold logic unit-u*“. Vrijednost funkije praga definira hiper ravninu koja je ortogonalna na os praga. Ova hiper ravnina praga može prepoloviti hiper ravninu koja je definirana vrijednostima težinske funkcije, tako da će jedan dio biti iznad praga, a drugi dio ispod. Funkcija praga ispituje ulaznu vrijednost koja je projicirana na hiper ravnini. Ako se ulaz projicira na dio iznad praga, aktivacijska vrijednost je 1, u suprotnom je 0 (što je i vidljivo na slici 3.13.).

„*Threshold logic unit*“ može imati proizvoljan broj izlaza. Svaki izlaz uzima uzima vrijednost aktivacijske funkcije i prosljeđuje je ili mrežnom izlazu ili drugim jedinicama unutar neuronske mreže.

3.2.2. Mreža bez povratnih mreža

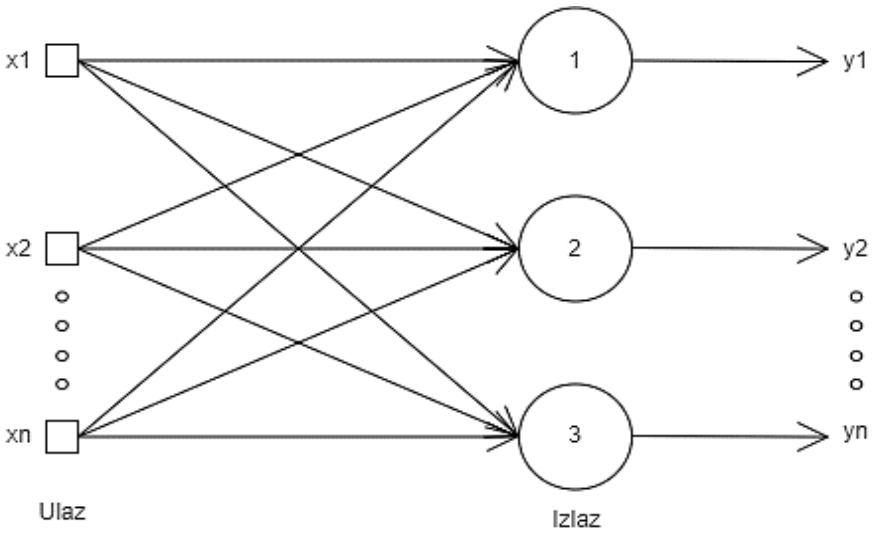
Mreža bez povratnih veza (eng. „*Feedforward neural network"*), izvedba je neuronskih mreža u kojoj poveznice između jedinica ne tvore nikakvu formu ciklusa. Ovaj tip neuronskih mreža bio je prva i najjednostavnija vrsta koja se primjenjivala u području umjetne inteligencije. Kod mreža bez povratnih veza informacije se kreću samo u jednom smjeru prema naprijed, od početnog ulaznog čvora koji se množenjem sa težinskim faktorom te kasnije kreće prema jednom ili više skrivenih slojeva, sve do izlaznog čvora. Postoje dva tipa neuronskih mreža bez povratnih veza, a to su:

- Jednoslojne mreže bez povratnih veza („*single-layer feedforward networks"*)
- Višeslojne mreže bez povratnih veza („*multi-layer feedforward networks*")

3.2.3. Jednoslojne mreže bez povratnih veza

Najjednostavnija vrsta neuronske mreže koja se može primjeniti u području dubokog učenja je jednoslojna perceptronska mreža, koja se sastoji od jednog sloja izlaznih čvorova. Princip rada mreže je da se ulazne vrijednosti vode izravno na izlaze preko težinskih faktora.

Jednoslojni perceptroni sposobni su učiti samo linearne uzorke. 1969. godine, Marvin Minsky i Seymour Papert dokazali su da jednoslojna perceptronska mreža ne može naučiti isključivo ILI (XOR) funkciju.



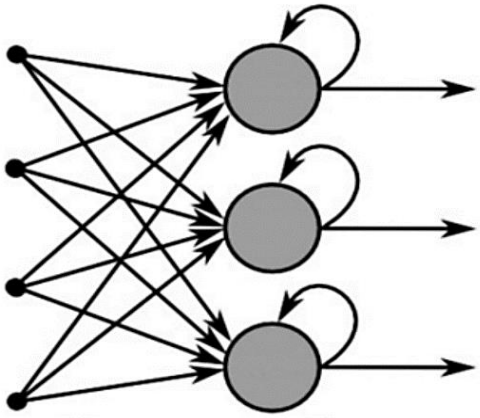
Slika 3.14. Jednoslojna neuronska mreža bez povratnih veza

Kao i kod osnovne neuronske mreže, izračunava se umnožak težinskog faktora s ulaznom vrijednošću mreže. Ako je vrijednost iznad zadanog praga, neuron se aktivira i prenosi aktiviranu vrijednost prema izlazu.

Jednoslojna neuronska mreža ima mogućnost izračunavanja kontinuiranog izlaza umjesto step funkcije. Najčešći izbor aktivacijske funkcije koja se primjenjuje kod korištenja jednoslojne mreže je logistička (sigmoidalna) aktivacijska funkcija.

3.2.4. Mreže s povratnim vezama

Mreža s povratnim vezama (eng. „*recurrent neural network – RNN*“), vrsta je neuronske mreže koja koristi sekvencijske podatke ili podatke iz vremenskih uzoraka. Ovaj tip algoritma dubokog učenja obično se koristi za klasične ili vremenske probleme, kao što su npr. prijevod jezika, prepoznavanje govora, opis slike itd. Ugrađeni su u moderne aplikacije za mobilne uređaje kao što su „*Siri*“, „*Google Translate*“ i njima slični. Slično kao i kod konvolucijskih neuronskih mreža, mreže s povratnim vezama koriste podatke za treniranje modela. Glavna razlika između konvolucijskih mreža i mreža s povratnim vezama je u njihovoj pohrani informacija (pamćenju), jer rekurentne mreže uzimaju informacije iz prethodnih ulaza kako bi utjecali i na trenutni ulaz te na izlaz.

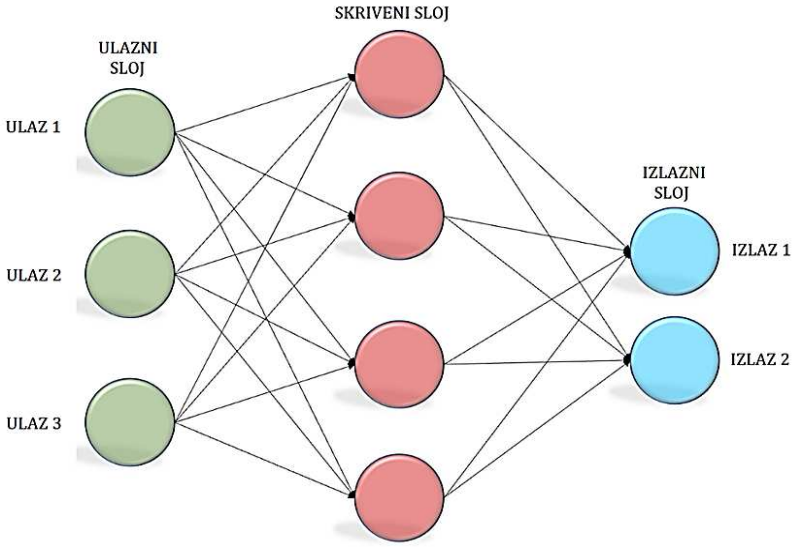


Slika 3.15. Shema mreže s povratnim vezama. [17]

Dok tradicionalne neuronske mreže pretpostavljaju da su ulazni i izlazni signali neovisni jedni o drugima, izlaz ponavljajućih neuronskih mreža ovisi o prethodnim elementima unutar niza. Dok bi budući događaji bili od pomoći u određivanju rezultata danog niza, jednosmjerne neuronske mreže s povratnim vezama mogu uzeti u obzir buduće događaje u svojim predviđanjima.

3.3. Višeslojni perceptron (MLP)

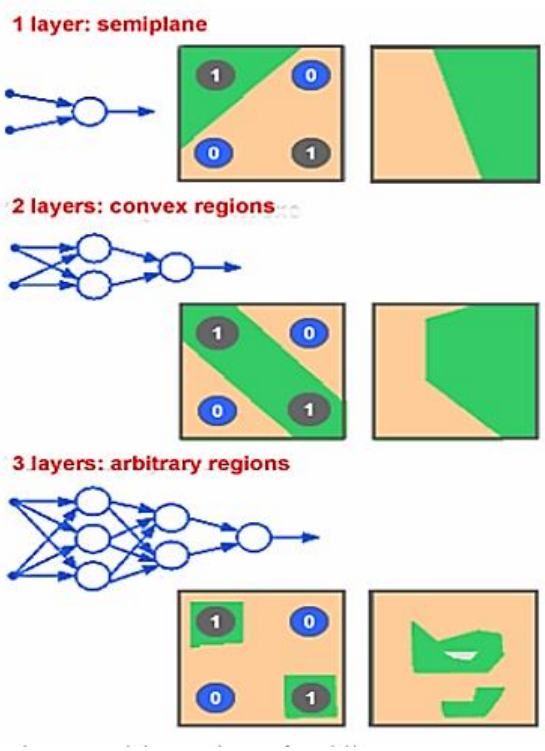
Višeslojni perceptron najpoznatija je i najčešća metoda implementacije umjetne neuronske mreže. Kao što je prethodno navedeno, u većini slučajeva princip rada neuronskih mreža odnosi se na prijenos signala jednosmjerno – od ulaza prema izlazu. U takvim situacijama nema petlje u mreži, pa izlaz neurona ne utječe na varijable neurona koje se prenose preko ulaznih i skrivenih slojeva. Slojevi koji nisu izravno povezani s okolinom nazivaju se skriveni slojevi. Dodavanjem skrivenih slojeva, svaki neuron djeluje kao standardni perceptron za izlaze neurona u sloju kojem on prednjači, što znači da izlaz mreže može procijeniti konveksna područja odlučivanja koja proizlaze iz sjecišta poluravnina koju stvaraju neuroni.



Slika 3.16. Višeslojni perceptron

Glavna značajka višeslojnih perceptrona povezanost je svakog neurona u svakom sloju sa svim neuronima iz idućeg sloja, kao što je i prikazano na slici 3.16. Preciznije rečeno, svaki neuron šalje više signala, od kojih jedan signal ide svakom idućem neuronu u sljedećem sloju. Za svaki signal, perceptron koristi različite težinske vrijednosti. Broj slojeva predstavlja dubinu, a broj perceptrona u neuronu predstavlja širinu neuronske mreže. Svaki sloj može imati veliki broj perceptrona, isto tako i više slojeva, čime sustav postaje složen.

Kada je broj slojeva manji, kao kod primjera sa slike koji predstavlja formu troslojnog perceptrona, mreža je plitka (nije duboka). Dodavanjem jednog ili više slojeva u mreži stvorila bi se forma duboke neuronske mreže.



Slika 3.17. Područja odlučivanja kod višeslojnih perceptrona

Razlika između MLP-a i bazne neuronske mreže je u tome što kod klasičnog perceptrona, funkcija odlučivanja ima vrijednost step funkcije ss binarnom vrijednošću izlaza, dok kod neuronskih mreža koje su se razvile iz MLP-a, koriste se aktivacijske funkcije koje rezultiraju vrijednostima stvarnih vrijednosti, obično u domeni između 0 i 1, ili između -1 i 1 (ovisno o aktivacijskog funkciji koja se primjenjuje). Time se omogućuju predviđanja na temelju vjerojatnosti ili klasifikacije podataka kroz više skupina. Sama moć višeslojnog perceptrona dolazi upravo iz nelinearnih aktivacijskih funkcija. Gotovo svaka nelinearna funkcija, osim polinomnih funkcija, može imati svrhu kod višeslojnog perceptrona.

Kako bi se determinirali izračuni koji se odvijaju na svakom zasebnom neuronu u izlaznom i skrivenom sloju primjenjuju se iduće formule:

$$o(x) = G(b(2) + W(2)h(x)) \qquad (3.1.)$$

$$h(x) = \phi(x) = s(b) + W(1)x) \qquad (3.2.)$$

Gdje su:

b(1), b(2) – vektori pristranosti

W(1), W(2) – matrica težine

G, s – aktivacijske funkcije

Set parametara za učenje prikazuje se na idući način:

$$\theta = \{W(1), b(1), W(2), b(2)\} \qquad (3.3.)$$

Kod primjene aktivacijskih funkcija u višeslojnim perceptronima, najčešći odabiri su tangens hiperbolna i sigmoidalna aktivacijska funkcija.

3.3.1. Programiranje višeslojnog perceptrona

Višeslojni perceptron najčešće se implementira u kod na način da se putem knjižnice „`sklearn.neural_network`“, poziva funkcija „`MLPClassifier`“. U slučaju da programski paket „Python“ nije u mogućnosti učitati „`sklearn`“ knjižnicu, u nastavku je prikazan drugi način kreiranja koda koji izrađuje višeslojni perceptron bez pozivanja potrebne knjižnice.

- Priprema funkcija i varijabli

```
import pandas
import numpy as np

def logistic(x):
    return 1.0/(1 + np.exp(-x))

def logistic_deriv(x):
    return logistic(x) * (1 - logistic(x))
```

Kod započinje pozivanjem najosnovnijih knjižnica koje se primjenjuju u programskom paketu „Python“, a to su „Numpy“ knjižnica koja se koristi pri mrežnim izračunima i primjeni matematičkih varijabli, i „Pandas“ knjižnica koja korisnica pruža mogućnost uvoza seta podataka o obuci stroja iz „Microsoft Excel“ datoteke.

U nastavku koda kreira se aktivacijska funkcija „logistic“ i njezina derivacija. Derivacija „logistic“ funkcije služi za povratno širenje (eng. *backpropagation*), dok sama logistička funkcija služi za izračun post-aktivacijskih vrijednosti.

```
Learning_Rate = #()

Ulazni_sloj = #()
Skriveni_sloj = #()
```

```
broj_epoha = #()

np.random.seed(1)
tezinski_faktor1 = np.random.uniform(-1, 1, (I_dim, H_dim))
tezinski_faktor1 = np.random.uniform(-1, 1, H_dim)

preActivation_H = np.zeros(H_dim)
postActivation_H = np.zeros(H_dim)
```

Drugi dio pripreme odnosi se na odabir tzv. stope učenja (eng. *Learning rate*), dimenzija ulaza i dimenzija skrivenih slojeva. Pri učenju stvarnih neuronskih mreža, odabir najpogodnijih vrijednosti za navedene varijable vrlo je bitna jer pravilnim odabirom omogućuje se maksimalni učinak strojnog učenja iz podataka o obuci. Svrha *np. random.seed(1)* funkcije je popunjavanje kreiranih matrica sa nasumičnim vrijednostima u domeni od -1 do 1.

Završni dio pripreme, stvaranje je praznih nizova za predaktivacijske i postaktivacijske vrijednosti u skrivenim slojevima.

- „*Feedforward*“ proces

```
for epoch in range(broj_epoha):
    for sample in range(broj_treniranja):
        for node in range(Hidden_dimension):
            preActivation_H[node] = np.dot(podaci_treniranja[sample,:],
weights_ItoH[:, node])
            postActivation_H[node] = logistic(preActivation_H[node])

        preActivation_O = np.dot(postActivation_H, weights_HtoO)
        postActivation_O = logistic(preActivation_O)

        FE = postActivation_O - target_output[sample]
```

Dio programskog koda koji proizvodi izlaznu vrijednost i u kojima se podaci pomiču slijeva prema desno u klasičnom dijagramu neuronske mreže, tvori „*feedforward*“ dio sustava.
Prva *for* petlja omogućuje računanje više perioda (epoha). Unutar svake epohe stvara se izračun izlazne vrijednosti za svaki uzorak. Svakim idućim uzorkom bilježi se iduća *for* petlja. U zadnjoj *for* petlji, pojedinačno se učitava svaki skriveni sloj, koristeći točkasti produkt za generiranje predaktivacijskog signala i aktivacijske funkcije za generiranje postaktivacijskog signala.

- „Backpropagation“

```
        for H_node in range(Hidden_dimension):
            S_greska = FE * logistic_deriv(preActivation_O)
            gradient_HtoO = S_error * postActivation_H[H_node]

            for I_node in range(I_dim):
                input_value = training_data[sample, I_node]
                gradient_ItoH = S_error * weights_HtoO[H_node] *
logistic_deriv(preActivation_H[H_node]) * input_value

                weights_ItoH[I_node, H_node] -= LR * gradient_ItoH

            weights_HtoO[H_node] -= LR * gradient_HtoO
```

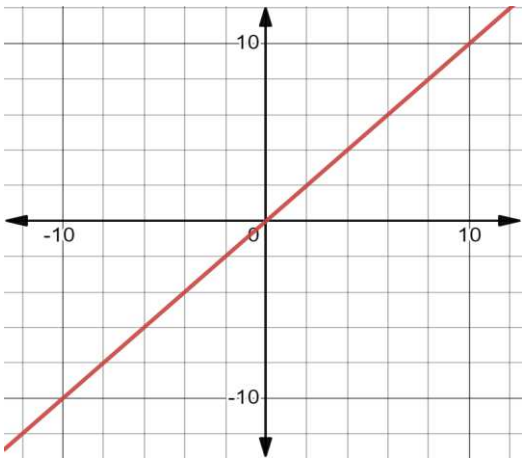
Nakon izvršenog proračuna za „*feedforward*“ proces, stvara se posljednji dio koda za višeslojni perceptron, a on se odnosi na povratno širenje procesa. Polazna točka je izlazni čvor, čiji se podaci „vraćaju“ natrag prema skrivenom čvoru te naposljetku prema ulaznom čvoru. Ovim procesom, postiže se prenošenje informacija o pogrešci koja se koristi u svrhu ispravljanja i učinkovitijeg treniranja neuronske mreže.

3.4. Aktivacijske funkcije

Aktivacijske funkcije ključni su dio dizajna neuronske mreže. Kod umjetnih neuronskih mreža, aktivacijska funkcija determinira izlaz čvora s obzirom na skup ulaza. Drugim riječima, standardni integrirani krug može se promatrati kao digitalna mreža aktivacijskih funkcija koje mogu poprimati vrijednosti „ON“ (1), ili „OFF“ (0), ovisno o skupu ulaznih vrijednosti. Odabir pogodne aktivacijske funkcije u skrivenom sloju kontrolirat će mogućnost koliko dobro će mrežni modeli učiti predodređeni skup podataka za obuku. Izborom aktivacijske funkcije u izlaznom sloju definirat će se vrsta predviđanja koje model može odrediti. Bazna podjela funkcije odvija se u dva dijela, a to su:

- Linearne aktivacijske funkcije
- Nelinearne aktivacijske funkcije

Na slici 3.18. prikazana je linearna *(„identity“)* aktivacijska funkcija f(x) = x. Vidljivo je kako izlaz funkcije nije ograničen ni u jednom rasponu (domena u skupu od -∞ do ∞). Linearna aktivacijska funkcija nije prikladna za primjenu kod skupa složenih parametara seta podataka koji se šalje neuronskim mrežama.



Slika 3.18. Linearna aktivacijska funkcija f(x) = x

Nelinearne aktivacijske funkcije najčešće su korištene aktivacijske funkcije. Prikladnije su za korištenje kod strojnog učenja jer olakšavaju modelu generalizaciju ili prilagodbu na različite podatke i devijaciju izlaza. Glavna terminologija koja se primjenjuje kod spominjanja nelinearnih aktivacijskih funkcija su:

- Derivacija ili diferencijal – Promjena kod osi ordinate i osi apcise, također poznata kao nagib
- Monotona funkcija – Funkcija koja je u svakom pogledu potpuno nerastuća ili neopadajuća

Nelinearne aktivacijske uglavnom se klasificiraju na temelju njihove domene ili tipa krivulje. Najčešće funkcije koje se primjenjuju u polju strojnog učenja su:

- ReLU – „Rectified Linear Activation“
- Sigmoidalna funkcija
- Tanh – Tangens hiperbolna aktivacijska funkcija

3.4.1. Sigmoidalna aktivacijska funkcija

Sigmoidalna (logistička) aktivacijska funkcija je funkcija koja se primjenjuje u algoritmu klasifikacije logističke regresije i spada u klasu nelinearnih aktivacijskih funkcija. Jedna od najvećih prednosti (i glavni razlog primjene) sigmoidalne funkcije je što funkcija ima vrijednosti strogo između 0 i 1. Način na koji sigmoidalna funkcija djeluje je da uzima bilo koju stvarnu vrijednost kao ulaz i daje izlazne vrijednosti u rasponu od 0 do 1.

Što je ulazna vrijednost veća (pozitivnija), to će izlazna vrijednost biti bliža vrijednosti 1, a što je manja to će vrijednost izlaza biti bliža nuli (ne dostiže vrijednost stvarne nule).

Sigmoidalna aktivacijska funkcija izračunava se na sljedeći način:

$$f(x) = \frac{1}{1+e^{-x}} \quad (3.4.)$$

Gdje je:

e – matematička konstanta, baza prirodnog logaritma

x – ulazna vrijednost aktivacijske funkcije

Način primjene sigmoidalne aktivacijske funkcije u programskom paketu „Python“ izvodi se na sljedeći način:

- Implementiranje „numpy“ knjižnice za potrebe korištenja prirodnog logaritma i definiranje sigmoidalne funkcije:

```
import numpy as np
import matplotlib.pyplot as plt

#Sigmoidalna aktivacijska funkcija
def Sigmoid(x):
    return(1/(1+np.exp(-x)))
```

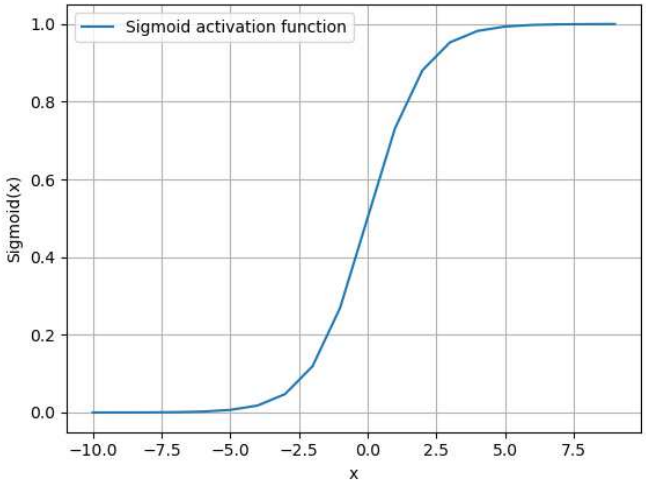
- Definiranje ulaznih i izlaznih vrijednosti koje se primjenjuju u sigmoidalnu funkciju:

```
ulaz = [x for x in range(-10,10)]

izlaz = [Sigmoid(x) for x in ulaz]
```

- Prikazivanje sigmoidalne aktivacijske funkcije:

```
plt.figure()
plt.plot(ulaz, izlaz, label = 'Sigmoid activation function'
plt.xlabel('x')
plt.ylabel('Sigmoid(x)')
plt.grid(True)
plt.legend()
plt.show()
```

Nakon pokretanja programskog koda, dobije se dijagram na slici 3.19. koji odgovara matematičkom prikazu sigmoidalne funkcije.



Slika 3.19. Rezultat programskog koda 'logistic' funkcije

3.4.2. Tangens hiperbolna aktivacijska funkcija

Tangens hiperbolna aktivacijska funkcija (ili kraće, Tanh funkcija), funkcija je koja uzima bilo koju stvarnu vrijednost na ulazu i daje izlazne vrijednosti u domeni od -1 do 1. Vrlo je slična sigmoidalnoj funkciji (poprima isti oblik krivulje kao i sigmoidalna funkcija), i na sličan se način programira u programskom paketu „Python“. Što je ulazna vrijednost veća (pozitivnija), izlaz će dati vrijednost bliži vrijednosti 1, a što je manja (negativnija), izlazna vrijednost težit će vrijednosti -1. Aktivacijska funkcija tanh izražava se na sljedeći način:

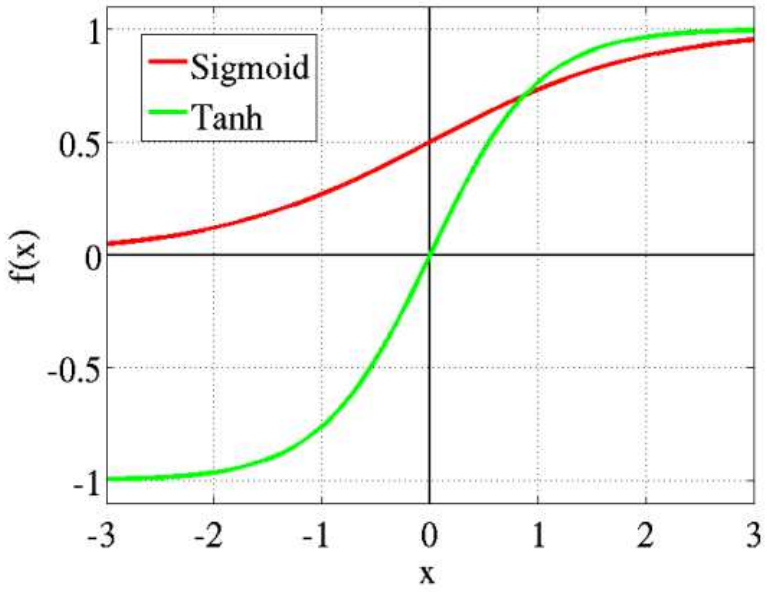
$$f(x) = \frac{e^{x}-e^{-x}}{e^{x}+e^{-x}} \qquad (3.5.)$$

$$f(x) = \frac{2}{1+e^{-2x}} - 1 \qquad (3.6.)$$

Gdje je:

e – matematička konstanta, baza prirodnog logaritma

x – ulazna vrijednost aktivacijske funkcije



Slika 3.20. Usporedba odziva za sigmoidalnu i tangens hiperbolnu aktivacijsku funkciju

U nastavku je prikazan programski kod za implementaciju tangens hiperbolne funkcije te rezultat programskog koda koji se implementira na skrivene slojeve neuronskih mreža.

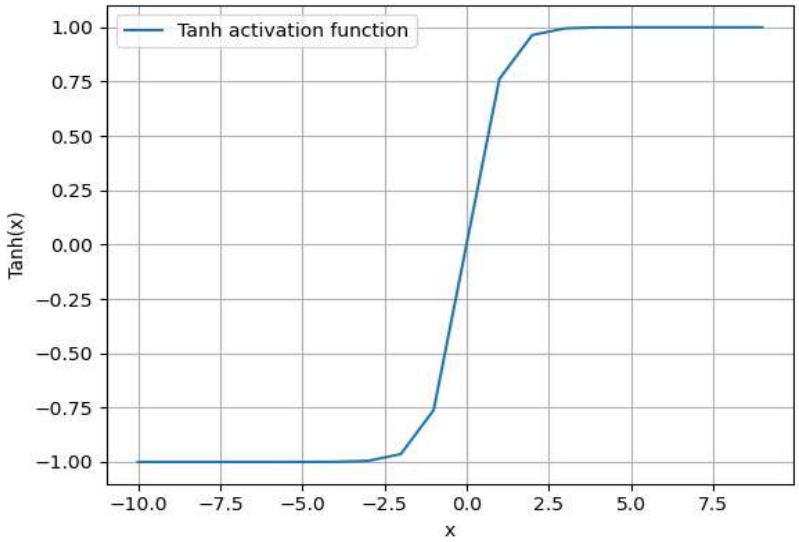
- Implementiranje „numpy“ knjižnice za potrebe korištenja prirodnog logaritma i definiranje tanh funkcije:

```
from math import exp
from matplotlib import pyplot

# tanh activation function
def tanh(x):
	return (exp(x) - exp(-x)) / (exp(x) + exp(-x))

#drugi nacin definiranja tanh funkcije

def tanh(x):
	return(np.exp(x) – np.exp(-x)) / (np.exp(x) + np.exp(-x))
```

- Definiranje ulaznih i izlaznih vrijednosti i prikaz dobivene funkcije:

```
# define input data
inputs = [x for x in range(-10, 10)]
# calculate outputs
outputs = [tanh(x) for x in inputs]
# plot inputs vs outputs
```

```
pyplot.plot(inputs, outputs)
pyplot.show()
```

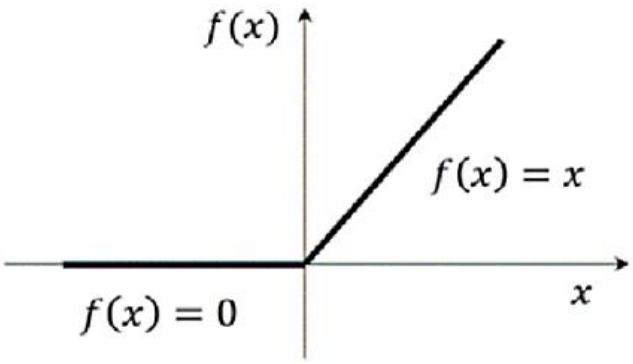


Slika 3.21. Rezultat programskog koda tanh funkcije

3.4.3. ReLU aktivacijska funkcija

Rektificirana linearna aktivacijska funkcija (eng. „*Rectified linear activation function*“), najčešće je korištena funkcija koja se koristi za skrivene slojeve. Uobičajena je iz razloga što je jednostavna za implementaciju te je učinkovita u prevladavanju limitacija drugih, prethodno korišenih aktivacijskih funkcija kao što su sigmoidalna i tanh funkcija. Preciznije rečeno, ReLU funkcija manje je osjetljiva na nestajanje gradijenata koji sprječavaju treniranje modela dubokog učenja. Iako ovaj tip aktivacijske funkcije sadrži veliki broj prednosti, model može patiti od problema kao što su zasićene ili tzv. mrtve jedinice. U nastavku je prikazan izraz za izračunavanje aktivacijske funkcije i graf funkcije (slika 3.22.)

$$f(x) = \begin{cases} 0, ako\ je\ x < 0 \\ x, ako\ je\ x \geq 0 \end{cases} \quad (3.7.)$$



Slika 3.22. ReLU aktivacijska funkcija

Princip računanja ReLU funkcije izračunava se na sljedeći način:

- Uvodi se varijabla `max`(0.0, x)

Čime se postavlja uvjet u kojem se vraća vrijednost 0.0, ako je ulazna vrijednost *x* negativna, a u protivnom se vraća vrijednost *x,* ako je ulazna vrijednost veća ili jednaka nuli.

Što se tiče implementacije aktivacijske funkcije ReLU u programski paket „Python“, ona se primjenjuje na sljedeći način:

- Uvođenje `matplotlib` knjižnice i definiranje ReLu aktivacijske funkcije

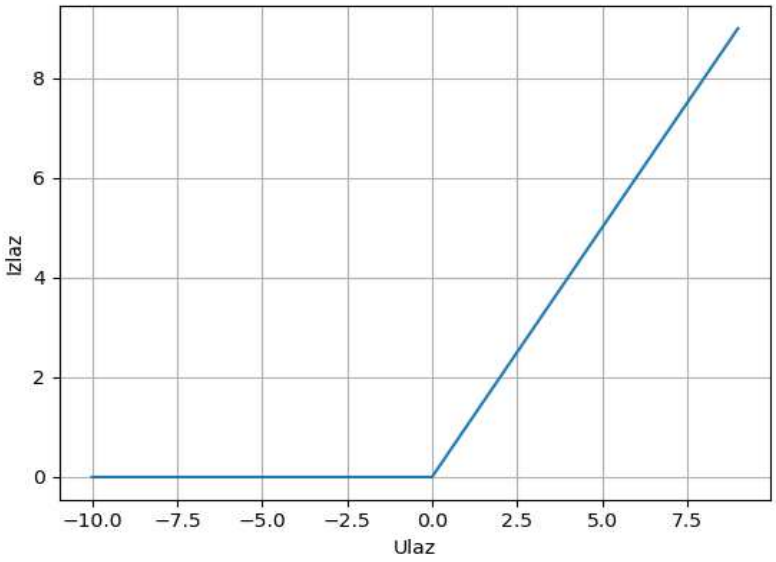
---

```
from matplotlib import pyplot as plt
def rectified(x):
      return max(0.0, x)
```

- Definiranje ulaznih i izlaznih parametara i prikaz dijagrama

---

```
ulaz = [x for x in range(-10,10)]

izlaz = [rectified(x) for x in ulaz]

plt.plot(ulaz,izlaz)
plt.xlabel('ulaz')
plt.ylabel('izlaz')
plt.legend
plt.grid(True)
plt.show()
```



Slika 3.23. Rezultati programskog koda za primjenu ReLU aktivacijske funkcije

Kod primjene ReLU aktivacijske funkcije na skrivene slojeve, često se u praksi primjenjuje funkcija „He Uniform“, čime se težinski faktori i ulazni podaci ograničavaju na rang između vrijednosti 0 i 1 (normalizacija).

ReLU funkcija najčešće se primjenjuje kod višeslojnih perceptrona (MLP) i konvolucijskih neuronskih mreža (CNN). Prednosti korištenja ReLU aktivacijske funkcije su:

- Računalna jednostavnost – Za razliku od sigmoidalne i tangens hiperbolne funkcije koje zahtijevaju upotrebu eksponencijalnih izračuna, implementacija ReLU aktivacijske funkcije jednostavna je za primijeniti na skrivene slojeve
- Reprezentativna rijetkost – Važna prednost ReLU-a je sposobnost ispisivanja stvarne nulte vrijednosti. Tanh i sigmoidalne aktivacijske funkcije uče aproksimirati nulti izlaz ili vrijednost koja je djelomično jednaka nuli, ali ne pravu nultu vrijednost.
- Linearno ponašanje – ReLU aktivacijska funkcija uglavnom izgleda i djeluje kao linearna aktivacijska funkcija (općenito, neuronsku mrežu lakše je optimizirati ako je njeno ponašanje linearno ili djelomično linearno).
- Treniranje dubokih mreža – Razvojem hardverskih računalnih specifikacija, implementiranje ReLU aktivacijska funkcija ima mogućnost iskorištavanja svih raspoloživih resursa i uspješnost u treniranju dubokih višeslojnih mreža s nelinearnom aktivacijskom mrežom metodom povratnog širenja.

3.5. Usporedba zvijezda i pulsara

Opće je poznato kako su zvijezde nebeska tijela koja se sastoje od enormne količine plinova, primarno vodika i helija. Kako bi se kvalitetno opisala usporedba pulsara i zvijezde, važno je napomenuti ključnu kategorizaciju "običnih" zvijezda. Glavna kategorizacija zvijezda vrši se po njihovoj masi i temperaturi te isto tako po njihovom spektru (elementu kojeg zvijezda apsorbira) i svjetlini (prividna magnituda). Na osnovu navedenih parametara, astronomi su zvijezde podijelili u 7 spektralnih tipova (kako bi se predočili stvarni radijus i masa zvijezde, oni će biti opisani kroz mjerilo Sunca):

Tablica 3.11. Kategorizacija zvijezda glavnog niza (eng. Main sequence stars)

Oznaka spektralnog tipa	Boja	Temperatura [K]	Radijus	Masa	Luminiscencija	Životni vijek [milijarde godina]
O	Plava	40 000 K	10 Sun	50 Sun	100 000 Sun	10
B	Plava	20 000 K	5 Sun	10 Sun	1000 Sun	100
A	Plava	8500 K	1.7 Sun	2.0 Sun	20 Sun	1
F	Plava/Bijela	6500 K	1.3 Sun	1.5 Sun	4.0 Sun	3
G	Bijela/Žuta	5700 K	1.0 Sun	1.0 Sun	1.0 Sun	10
K	Narančasta/Žuta	4500 K	0.8 Sun	0.7 Sun	0.2 Sun	50
M	Crvena	3200 K	0.3 Sun	0.2 Sun	0.01 Sun	200

Ovaj način kategorizacije poznat je i kao Morgan Keenan-ov sistem. MK sistem uveden je 1943. Godine i dan danas koristi se u modernoj astronomiji kao najpouzdanija klasifikacija zvijezda.

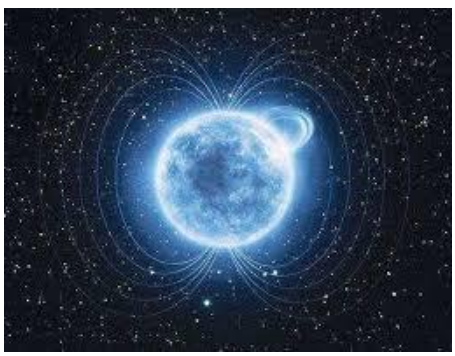
Pulsar ili neutronska zvijezda najgušći je objekt kojeg astronomi mogu promatrati izravno, te zbog njegove gustoće, jedan je od najekstremnijih objekata u svemiru. U većini slučajeva, dijametar pulsara iznosi nekoliko kilometara te bi mogao pokriti veći grad na zemlji. Iako je po radijusu pulsar prema astronomskim jedinicama mikroskopski malen objekt, njegova masa može doseći i mase zvijezda glavnog niza.

3.5.1. Formiranje i karaktetistike pulsara

Kako bi se pojasnilo formiranje pulsara, važno je objasniti odnos zvjezdane gravitacije i fuzije. Sama gravitacija privlači zvjezdanu plazmu prema samom centru zvijezde. Materijal u plazmi (primarno vodik), kompresira se sa silom dovoljnom da se jezgre vodika spajaju. Vodik se fuzijom pretvara u helij, čime se oslobađa dovoljna količina energije koja održava zvjezdanu ravnotežu između gravitacijske sile i fuzije (dok god postoji ova ravnoteža, zvijezde su stabilni objekti).

S vremenom, vodik u zvijezdi se potroši, nakon čega ostaje samo helij, koji nakon spajanja jezgri postaje ugljik i kisik. Kod zvijezda glavnog niza, proces izgaranja plina završava kod fuzije ugljika i kisika, čime ovaj tip zvijezda prelazi u tzv "white dwarf", ali kod sekvenci zvijezda koje su puno teže od npr. Sunca, dolazi do komplikacija kada se potroši sav helij u zvijezdi. U određenom trenutku, kada zvijezda potroši sav helij u sebi, odnos radijacije i gravitacijske sile više nije uravnotežen. Gravitacija privlači materijal zvijezde prema centru, zbog čega zvijezda postaje sve gušća. Jezgra zvijezde gori jače i brže, zbog čega nastaje fuzija "težih" elemenata (ugljik prelazi u neon, neon prelazi u kisik, kisik u silicij i na posljetku, silicij prelazi u željezo).

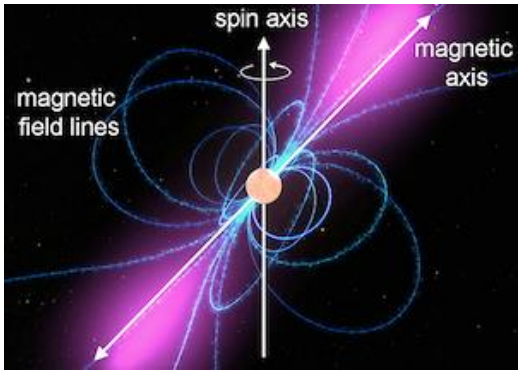
Nastankom željeza (koji je u kemiji prikazan kao nuklearni pepeo) u jezgri, zvijezda više nije u mogućnosti stvarati fuziju jer nema energiju za kreiranje fuzije. Gravitacija u potpunosti dominira zvijezdom te njenom silom privlači kompletnu masu prema željeznoj jezgri (pritisak postaje dovoljan da se elektroni i protoni spajaju u neutrone,kojim se zatim stlače dovoljnom snagom da se između atomskih jezgri izgubi sav prazan prostor). Nakon toga, cijela zvijezda implodira (brzinom od 25% brzine svjetlosti), i nastaje astronomski fenomen zvani supernova. Zaključno ovom procesu, nakon supernove, preostali materijal od zvijezde formira se i oblikuje pulsar.



Slika 3.24. Grafički prikaz pulsara [21]

Kao što je prethodno navedeno, pulsari su neutronske zvijezde koje se brzo okreću te zbog njihovih karakteristika, predstavljaju najekstremniji objekt u svemiru koji se može promatrati direktno. Uz crnu rupu, pulsari imaju najveću gravitaciju u svemiru (da se snaga gravitacijske sile poveća, pulsar bi postao crna rupa). Svjetlost je iskrivljena oko pulsara, što znači da se promatranjem pulsara može vidjeti prednja i dio zadnje strane u isto vrijeme.

Jedna od najznačajnijih karakteristika pulsara njihova je rotacija. Čim se neutronska zvijezda formira, počinje se rotirati oko svoje osi. Dok rotacija Sunca oko svoje osi traje 24 dana, pulsari se mogu vrtiti i do nekoliko milijuna puta u sekundi. Rotacijom se stvaraju pulsevi, jer magnetsko polje zvijezde rotacijom stvara radio valove. U našoj galaksiji, astronomi su trenutno pronašli približno 2000 radio pulsara. Važno je napomenuti činjenincu da je magnetsko polje pulsara najjače u svemiru. Usporedbe radi, magnetsko polje našeg planeta iznosi 0.00005 Tesli, dok magnetko polje prosječne neutronske zvijezde iznosi i preko 100 milijardi Tesli (pulsari koji imaju ovakvu snagu magnetskog polja nazivaju se magnetari)



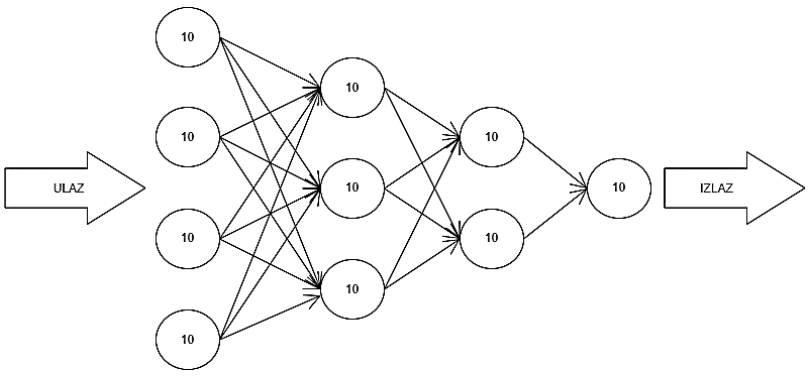
Slika 3.25. Rotacija neutronske zvijezde [21]

4. TRENIRANJE MODELA I REZULTATI

Kao što je napomenuto u prethodnom poglavlju, implementaciju višeslojnog perceptrona za strojno učenje puno je lakše izvesti uvozom programske funkcije '*MLPClassifier*' iz programske knjižnice '*sklearn.neural_network*'. Upravo je ta funkcija glavna značajka izrađenog programskog koda za treniranje modela i sami rezultati ovise o korekciji njenih parametara koji određuju uspješnost strojnog učenja. Cilj koji bi zadovoljio zahtjeve strojnog učenja je 88% preciznosti klasifikacije, s naglaskom na izmjenu atributa klasifikatora kako bi se dobili što je moguće bolji rezultati. U nastavku je prikazan primjer implementacije '*MLPClassifier*'*-a* s objašnjenjem pojedinih atributa koji čine sastavni dio klasifikatora.

```
MLPClassifier = [{'hidden_layer_sizes':  [(10,10,10,10),
                 (10,10,10),(10,10), (10)],
                 'activation': ['logistic','tanh','relu','identity'],
                 'solver': ['adam', 'lbfgs','sgd'],
                 'learning_rate':['constant','adaptive','invscaling'],
                 'learning_rate_init': [0.1,0.01,0.5, 0.00001],
                 'alpha': [0.01,0.1,0.001, 0.0001],
                 'max_iter': [1000]}]
```

- Prvi dio klasifikatora (`'hidden_layer_sizes'`) odnosi se na izgled neuronske mreže, odnosno broj skrivenih slojeva koje će neuronska mreža imati. Svaka zagrada sadrži određen broj redova skrivenih slojeva, a brojevi unutar zagrada predstavljaju broj neurona u skrivenom sloju. Primjer konfiguracije skrivenih slojeva neuronske mreže prikazan je na slici 4.1.



Slika 4.1. Primjer izgleda neuronske mreže

Povećavanjem skrivenih slojeva utječe se na uspješnost strojnog učenja uz napomenu kako veliki broj skrivenih slojeva zahtijeva visoke specifikacije računala. Optimizacijom parametara skrivenih slojeva bez da se opterećuje računalni sustav, postiže se maksimalna efikasnost treniranja modela.

- Funkcija `'activation'` je aktivacijska funkcija koja primjenjuje odabrane aktivacijske funkcije na skrivene slojeve. Kroz sve simulacije treniranja modela za klasifikaciju pulsara koristit će se sigmoidalna aktivacijska funkcija.
- `'solver'` je alat koji optimizira težinske faktore učenja modela. Za primjereno korištenje klasifikatora, važno je determinirati koji optimizator najbolje odgovara određenom treniranju modela.
    - `lbfgs` – optimizator koji koristi kvazi-Newtonove metode
    - `sgd` – odnosi se na stohastički gradijentni pad
    - `adam` – odnosi se na optimizator temeljen na stohastičkom gradijentu
- `'alpha'` alat determinira snagu člana regulacije L2. Izraz regulacije L2 dijeli se s veličinom uzorka kada se pridoda gubitku. Za potrebe klasifikacije pulsara, vrijednost `'alpha'` ostala je fiksna, s iznosom 0.01.
- `'learning_rate'{'constant', 'invscaling', 'adaptive'}` – Određuje način, odnosno raspored stope učenja za ažurirane težinske faktore. Nastavak navedenog dijela klasifikatora (`'learning_rate_init'),` kontrolira veličinu step funkcije pri ažuriranju težinskog faktora. Ovaj alat se koristi samo u slučajevima kada se za treniranje modela primjenjuje `'sgd'` ili `'adam'` optimizator
- `'max_iter'` određuje maksimalan broj iteracija treniranja modela. Za potrebe klasifikacije pulsara odredio se broj od maksimalno 1000 iteracija.

Druga važna značajka programskog koda odnosi se na implementaciju seta podataka. Za potrebe dobivanja što većeg postotka preciznosti treniranja modela, koristio se originalni set podataka HTRU2, kao i njegova varijanta u kojem je klasa signala detektiranih radio valova te signala šuma i smetnji ujednačena. Rezultati za implementaciju oba seta podataka prikazana su u nastavku.

.

Rezultati treniranja modela prikazani su kroz točnost, standarnu pogrešku i tzv. F1 rezultat. F1 rezultati zapravo najbolje opisuju uspješnost pojedinog treniranja modela iz razloga što F-rezultati (eng. *F-score*) predstavljaju mjeru točnosti modela na temelju implementiranog skupa podataka. F1-score koristi se za procjenu binarnih klasifikacijskih sustava, koji se temelje na klasificiranju primjera u „pozitivne primjere“ ili „negativne primjere“.

F1-score rezultat je načina kombinacije preciznosti i prisjećanja modela, a definiran je kao njihova harmonijska sredina. Standardni izraz za izračun F1-rezultata glasi:

$$F_1 = \frac{2}{\frac{1}{t_s}\cdot\frac{1}{t_a}} = 2 \cdot \frac{t_s \cdot t_a}{t_s + t_a} = \frac{tp}{tp + \frac{1}{2}(fp + fn)} \quad (4.1.)$$

Gdje je :

t_a – udio pravih pozitivnih primjera koje je model klasificirao

t_s – udio pravih pozitivnih rezultata podijeljen sa zbrojem pravih pozitivnih i lažno negativnih primjera

t_p – broj pravih pozitivnih primjera klasificiranih od strane modela

f_n – broj lažnih negativnih primjera klasificiranih od strane modela

f_p – broj lažnih pozitivnih primjera klasificiranih od strane modela

- Prva simulacija treniranja modela

a) Originalni set podataka

Tablica 4.1. Hiperparametri neurona prve simulacije (originalni set podataka)

Broj skrivenih slojeva neuronske mreže	(10,10,10,10),(10,10,10),(10,10),(10)
Aktivacijske funkcije	Sigmoidalna, tangens hiperbolna
Optimizatori	'adam' , 'lbfgs'
Alfa	0.1, 0.001, 0.0001
Stopa učenja	'constant', 'adaptive', 'invscaling'
Početna stopa učenja	0.01, 0.1, 0.5, 0.00001.
Najveći dopušteni broj iteracija	1000

Svaki hiperparametar za treniranje modela sadrži više kombinacija, kako bi simulacija bila u poziciji dati najbolje moguće rezultate za treniranje. Simulacija je dala 769 različtih rezultata, a najbolji rezultati simulacije prikazani su u tablici 4.2.

Tablica 4.2. Rezultati prve simulacije (originalni set podataka)

Preciznost	F1	Aktivacijska funkcija	Alfa	Broj skrivenih slojeva	Stopa učenja	Vrijednost stope učenja	Optimizator
0.985474	**0.902985**	**logistic**	**0.01**	**(10,10,10)**	**adaptive**	**0.00001**	**lbfgs**
0.985474	0.902621	tanh	0.001	(10,10,10)	invscaling	0.5	lbfgs
0.985196	0.900562	logistic	0.01	(10,10,10,10)	adaptive	0.1	lbfgs
0.985196	0.899810	logistic	0.01	(10,10)	constant	0.1	lbfgs
0.984637	0.897579	tanh	0.1	(10,10,10,10)	invscaling	0.01	lbfgs

Najbolji uzorak navedene simulacije daje uspješnost treniranja modela od 0.902985, uz korištenje sigmoidalne aktivacijske funkcije sa alfa faktorom od 0.01. Ostali rezultati nisu previše odstupali od prosječne vrijednosti f1-rezultata od oko 0.82. Kao najbolji optimizator za treniranje modela uz prethodno odabrane hiperparametre pokazao se 'lbfgs' optimizator.

b) Ujednačeni set podataka

Tablica 4.3. Hiperparametri neurona prve simulacije (ujednačeni set podataka)

Broj skrivenih slojeva neuronske mreže	(10,10,10,10),(10,10,10),(10,10),(10)
Aktivacijske funkcije	Sigmoidalna, tangens hiperbolna
Optimizatori	'adam' , 'lbfgs'
Alfa	0.1, 0.001, 0.0001
Stopa učenja	'constant', 'adaptive', 'invscaling'
Početna stopa učenja	0.01, 0.1, 0.5, 0.00001.
Najveći dopušteni broj iteracija	1000

Simulacija se ponovno pokreće, ovaj put uz uvoz ujednačenog seta podataka. Hiperparametri neurona ostali su nepromijenjeni te isto kao i kod prethodne simulacije, simulacija sa ujednačenim setom podataka nije premašila maksimalni broj dopuštenih iteracija i daje 769 različitih učenja modela.

Tablica 4.4. Rezultati prve simulacije (ujednačeni set podataka)

Preciznost	F1	Aktivacijska funkcija	Alfa	Broj skrivenih slojeva	Stopa učenja	Vrijednost stope učenja	Optimizator
0.949695	**0.948518**	**logistic**	**0.001**	**(10,10)**	**constant**	**0.5**	**adam**
0.948171	0.947368	tanh	0.0001	(10)	constant	0.5	adam
0.948171	0.947205	tanh	0.0001	(10)	adaptive	0.01	adam
0.948171	0.946708	logistic	0.01	(10,10,10)	adaptive	0.5	adam
0.946646	0.945736	tanh	0.0001	(10)	adaptive	0.1	adam

Simulacija sa ujednačenim setom podataka, za razliku od prethodne simulacije, daje konstantnije i bolje rezultate, kao i manji broj neuspješnih treniranja modela. Najbolji rezultati postigli su se 'adam' optimizatorom. Najveća točnost treniranja modela ima f1- rezultat od 0.948518.

- Druga simulacija treniranja modela

a) Originalni set podataka

Tablica 4.5. Hiperparametri neurona druge simulacije (originalni set podataka)

Broj skrivenih slojeva neuronske mreže	(12,12,12),(10,10,10),(8,8)
Aktivacijske funkcije	Sigmoidalna, tangens hiperbolna
Optimizatori	'sgd'
Alfa	0.1, 0.001, 0.0001, 0.2, 0.5
Stopa učenja	'constant', 'adaptive', 'invscaling'
Početna stopa učenja	0.1, 0.2, 0.5
Najveći dopušteni broj iteracija	700

Najveća izmjena hiperparametara, za razliku od prve simulacije, nalazi se u broju skrivenih slojeva neuronske mreže, koji za razliku od prve simulacije, ima veću varijaciju broja neurona u skrivenim slojevima. Simulacija daje 108 rezultata, a njeni najbolji rezultati prikazani su u tablici 4.5.

Tablica 4.6. Rezultati druge simulacije (originalni set podataka)

Preciznost	F1	Aktivacijska funkcija	Alfa	Broj skrivenih slojeva	Stopa učenja	Vrijednost stope učenja	Optimizator
0.980168	**0.863724**	**tanh**	**0.1**	**(8,8)**	**invscaling**	**0.5**	**sgd**
0.980726	0.863366	tanh	0.1	(10,10,10)	adaptive	0.2	sgd
0.979888	0.862069	tanh	0.0001	(12,12,12)	invscaling	0.2	sgd
0.97905	0.853229	tanh	0.1	(12,12,12)	adaptive	0.2	sgd
0.976257	0.841121	tanh	0.1	(8,8)	adaptive	0.1	sgd

Iako prikazani rezultati upućuju na prihvatljivost izmjene hiperparametara, za razliku od prve simulacije, puno je manja prosječna točnost treniranja modela. Promjenom optimizatora i broja skrivenih slojeva znatno je utjecalo na uspješnost simulacije, koja je uz manju točnost, imala i manji broj uspješnih treniranja modela.

b) Ujednačeni set podataka

Tablica 4.7. Hiperparametri neurona druge simulacije (ujednačeni set podataka)

Broj skrivenih slojeva neuronske mreže	(12,12,12),(10,10,10),(8,8)
Aktivacijske funkcije	Sigmoidalna, tangens hiperbolna
Optimizatori	'sgd'
Alfa	0.1, 0.001, 0.0001, 0.2, 0.5
Stopa učenja	'constant', 'adaptive', 'invscaling'
Početna stopa učenja	0.1, 0.2, 0.5
Najveći dopušteni broj iteracija	700

Tablica 4.8. Rezultati druge simulacije (ujednačeni set podataka)

Preciznost	F1	Aktivacijska funkcija	Alfa	Broj skrivenih slojeva	Stopa učenja	Vrijednost stope učenja	Optimizator
0.928354	**0.926217**	**tanh**	**0.0001**	**(12,12,12)**	**invscaling**	**0.1**	**sgd**
0.925305	0.923318	tanh	0.1	(10,10,10)	adaptive	0.2	sgd
0.922256	0.921659	tanh	0.0001	(10,10,10)	invscaling	0.5	sgd
0.919207	0.918587	tanh	0.1	(8,8)	adaptive	0.1	sgd
0.920732	0.917197	tanh	0.1	(10,10,10)	adaptive	0.2	sgd

Ujednačeni set podataka popravio je rezultate točnosti i poboljšao broj uspješnih treniranja modela. Iako je simulacija sa ujednačenim setom podataka povećala prosječnu preciznost i točnost učenja, rezultati simulacija imaju preveliku devijaciju između sebe i zaključuje se da izmjena hiperparametara u ovoj simulacije nije dala zadovoljavajuće rezultate.

- Treća simulacija treniranja modela

Tablica 4.9. Hiperparametri neurona treće simulacije (originalni set podataka)

Broj skrivenih slojeva neuronske mreže	(20,20,20,20,20), (20,20,20,20), (20,20,20), (20,20), (20), (5,5,5,5,5), (5,5,5), (5,5), (5)
Aktivacijske funkcije	Sigmoidalna, tangens hiperbolna
Optimizatori	'adam' , 'lbfgs'
Alfa	0.1, 0.001, 0.0001, 0.2, 0.5
Stopa učenja	'constant', 'adaptive', 'invscaling'
Početna stopa učenja	0.00001, 0.01, 0.1, 0.5
Najveći dopušteni broj iteracija	1000

Treća simulacija treniranja modela sastoji se od najsloženijeg skupa skrivenih slojeva neuronske mreže. Povećanjem broja skrivenih slojeva cilj je postići manju devijaciju dobivenih rezultata u njihovom međusobnom odnosu. Simulacija je, za razliku od prethodnih simulacija, dala najveći broj rezultata, njih 1728. Najbolji rezultati simulacije sa originalnim setom podataka prikazani su u tablici 4.9.

Tablica 4.10. Rezultati treće simulacije (originalni set podataka)

Preciznost	F1	Aktivacijska funkcija	Alfa	Broj skrivenih slojeva	Stopa učenja	Vrijednost stope učenja	Optimizator
0.985754	**0.90573**	**logistic**	**0.1**	**(20,20,20)**	**invscaling**	**0.01**	**lbfgs**
0.985475	0.903346	logistic	0.01	(20,20,20)	adaptive	0.1	lbfgs
0.984916	0.898496	logistic	0.01	(20,20,20)	adaptive	0.00001	lbfgs
0.984637	0.897579	logistic	0.01	(20,20)	constant	0.1	adam
0.984078	0.894249	logistic	0.01	(20,20,20)	invscaling	0.00001	lbfgs

Rezultati treće simulacije sa uvezenim originalnim setom podataka ne odstupaju previše od prosječne vrijednosti f1-rezultata. Iako su najbolji rezultati treniranja modela slični prvoj simulaciji koja je sadržavala manji broj skrivenih slojeva, treća simulacija također sadrži veći postotak uspješnosti treniranja modela.

- Četvrta simulacija treniranja modela

a) Originalni set podataka

Tablica 4.11. Hiperparametri neurona četvrte simulacije (originalni set podataka)

Broj skrivenih slojeva neuronske mreže	(18,18,18), (16,16,16), (15,15), (12)
Aktivacijske funkcije	sigmoidalna, tangens hiperbolna
Optimizatori	'adam', 'lbfgs'
Alfa	0.0001, 0.001, 0.01, 0.1
Stopa učenja	'invscaling', 'adaptive', 'constant'
Početna stopa učenja	0.00001, 0.01, 0.1, 0.5
Najveći dopušteni broj iteracija	1000

Tablica 4.12. Rezultati četvrte simulacije (originalni set podataka)

Preciznost	F1	Aktivacijska funkcija	Alfa	Broj skrivenih slojeva	Stopa učenja	Vrijednost stope učenja	Optimizator
0.985754	**0.903955**	**logistic**	**0.001**	**(16,16,16)**	**constant**	**0.00001**	**lbfgs**
0.985196	0.900563	logistic	0.1	(18,18,18)	adaptive	0.1	lbfgs
0.985196	0.900188	tanh	0.001	(12)	adaptive	0.1	lbfgs
0.984916	0.898876	tanh	0.0001	(12)	adaptive	0.1	adam
0.984637	0.897579	tanh	0.0001	(12)	invscaling	0.00001	lbfgs

Četvrta simulacija sa originalnim setom podataka daje 765 rezultata, od kojih su najbolji rezultati točnosti prikazani u tablici 4.11. Obzirom da f1-rezultati premašuju vrijednost zahtijevane točnosti za klasifikaciju pulsara, može se zaključiti da je simulacija uspješna.

Tablica 4.13. Rezultati četvrte simulacije (ujednačeni set podataka)

Preciznost	F1	Aktivacijska funkcija	Alfa	Broj skrivenih slojeva	Stopa učenja	Vrijednost stope učenja	Optimizator
0.949695	**0.948837**	**logistic**	**0.1**	**(15,15)**	**constant**	**0.00001**	**lbfgs**
0.943598	0.942636	logistic	0.1	(16,16,16)	invscaling	0.5	lbfgs
0.942073	0.94081	logistic	0.1	(15,15)	constant	0.1	adam
0.942073	0.94081	tanh	0.0001	12	constant	0.1	adam
0.942073	0.94081	tanh	0.0001	12	constant	0.01	adam

Tablica 4.12. prikazuje najbolje rezultate četvrte simulacije sa uvezenim ujednačenim setom podataka. Rezultati simulacije imaju najmanje odstupanje od prosječne vrijednosti, što je vidljivo i iz tablice, s obzirom na isti f1-rezultat točnosti treniranja modela posljednja tri rezultata.

- Peta simulacija treniranja modela

a) Originalni set podataka

Tablica 4.14. Hiperparametri neurona pete simulacije (originalni set podataka)

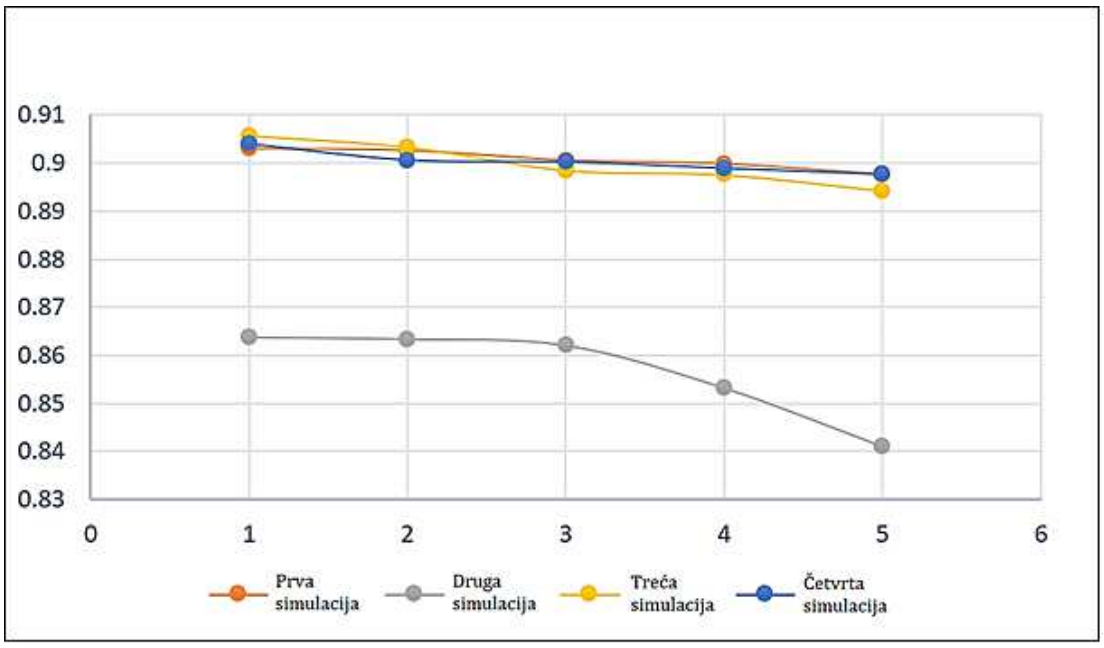
Broj skrivenih slojeva neuronske mreže	(15,15,15), (10,10), (5,5)
Aktivacijske funkcije	sigmoidalna
Optimizatori	'sgd'
Alfa	0.01, 0.1, 0.001, 0.0001
Stopa učenja	'constant', 'adaptive', 'invscaling'
Početna stopa učenja	0.01,0.05,0.1
Najveći dopušteni broj iteracija	700

Posljednja simulacija koja je trenirala model sa navedenim programskim kodom, sa svrhom eksperimenta, imala je znatno manji spektar hiperparametara kojima se izvodila klasifikacija. Ukupno je izradila 124 rezultata sa znatno manjim uspjehom treniranja modela za razliku od prethodnih simulacija. Rezultati simulacije prikazani su u tablici 4.15. Preciznosti treniranja poprilično su visoke i zadovoljavajuće, ali od ukupno 112 rezultata, 30 rezultata nije uspjelo završiti proces treniranja modela, što je retrospektivno gledano, daleko najveći broj neuspjelih treniranja za razliku od prethodnih simulacija.

Obzirom na veliki broj neuspješnih treniranja modela iz pete simulacije, nije potrebno odrađivati simulaciju sa ujednačenim setom podataka.

Tablica 4.15. Rezultati pete simulacije (originalni set podataka)

Preciznost	F1	Aktivacijska funkcija	Alfa	Broj skrivenih slojeva	Stopa učenja	Vrijednost stope učenja	Optimizator
0.98184	0.881226	**logistic**	**0.01**	**(15,15,15)**	**constant**	**0.05**	**sgd**
0.977376	0.875717	logistic	0.1	(15,15,15)	adaptive	0.01	sgd
0.977374	0.874045	logistic	0.1	(15,15,15)	adaptive	0.1	sgd
0.974022	0.872381	logistic	0.01	(10,10)	constant	0.01	sgd
0.96899	0.869070	logistic	0.01	(10,10)	adaptive	0.05	sgd



Slika 4.2. Usporedba najboljih rezultata pojedinih simulacija za originalni set podataka

Većina provedenih simulacija, koje su treniranje modela odradile sa originalnim HTRU_2 setom podataka, imale su zadovoljavuće rezultate, obzirom da je primarni zahtjev i očekivanje treniranja modela bilo iznad 0.88 točnosti. Druga simulacija ograničila je klasifikator na specifične postavke i obzirom na manji broj iteracija, relativno je zadovoljila zahtjev treniranja. Na slici 4.2. vidljivo je kako svaki pokušaj treniranja modela sa uvođenjem ujednačenog seta podataka, uvelike poboljšava točnost treniranja modela, primarno zbog jednakog broja klase signala pulsara, čime se postiže jednostavnije i točnije učenje modela. Razlike simulacija vidljive su promatranjem standardnih devijacija, koje su dosta velike kod rezultata čije se vrijednosti hiperparametara nisu dovoljno optimizirale za zadano strojno učenje.

5. ZAKLJUČAK

S obzirom na činjenicu da se svakim danom klasifikacije astronomskih objekata mijenjanju potencijalnim otkrićem objekata koji utječu na postojeće podatke, može se zaključiti da su astronomske klasifikacije konstantan zadatak za astronome. Veliku ulogu u obradi astronomskih podataka može odigrati umjetna inteligencija, koja svojim metodama može iskoristiti velike potencijale modernih računala i uvelike olakšati klasifikacije. Upravo je ta uloga definirana i dokazana kroz ovaj rad.

Za klasifikaciju neutronskih zvijezda (pulsara) odabrala se metoda korištenja višeslojnog perceptrona, koji se pokazao kao idealan kandidat za ovaj zadatak. Uz postojeći set podataka koji je javno dostupan, cilj implementacije višeslojnog perceptrona bio je postići treniranje modela uz zadovoljavajuće rezultate točnosti, koji su prije početka izrade programskog koda bili predviđeni da će iznositi oko 85-90%.

Nakon analize seta podataka pod nazivom HTRU_2 i utvrđivanja velike anomalije broja signala u klasama koji predstavljaju stvarni detektirani radio val pulsara i šum (smetnju), izvršila se izmjena podataka u setu pri kojoj se ujednačio broj signala iz svake klase kako bi se dodatno povećala točnost treniranja modela.

Programski kod temelji se na principu korištenja 'MLPClassifiera', koji predstavlja metodu višeslojne neuronske mreže s određenom grupom hiperparametara. Izmjenom navedenih hiperparametara, utječe se na rezultate simulacije koji daju točnost treniranja modela i do 91%, čime se ispunjuju uvjeti klasifikacije pulsara i nadmašuju prethodno određene prognoze. Implementacijom ujednačenog seta podataka, svaka iduća simulacija dosezala je i do 95% točnosti treniranja modela. Usporedbom dobivenih rezultata utvrđuje se kako svaka simulacija može zadovoljiti uvjete točnosti kroz određen broj rezultata ali potrebno je optimizirati hiperparametre kako bi se smanjilo odstupanje vrijednosti svih rezultata od prosječne vrijednosti.

Iako svaka simulacija daje povoljne rezultate treniranja modela, točnost treniranja za klasifikaciju pulsara dodatno se može poboljšati detaljnijom provjerom i izmjenom hiperparametara klasifikatora, primarno povećanjem skrivenih slojeva i smanjivanjem koraka stope učenja, za koje su potrebne puno jače računalne specifikacije koje će obrađivati simulaciju programskog koda u manjem vremenu.

6. LITERATURA

[1] Artificial neural network classification of asteroids in the M1:2 mean-motion resonance with Mars, *Monthly Notices of the Royal Astronomical Society, Volume 504, Issue 1, June 2021, Pages 692–700*, V. Carruba, S Aljbaae, R C Domingos, W Barletta, https://arxiv.org/pdf/2103.15586.pdf

[2] - Finding black holes with black boxes – using machine learning to identify globular clusters with black hole subsystems, Ammar Askar, Abbas Askar, Mario Pasquato, Mirek Giersz, https://doi.org/10.1093/mnras/stz628

[3] - An artificial intelligence enhanced star identification algorithm, *Frontiers of Information Technology & Electronic Engineering volume 21, pages1661–1670 (2020);* Hao Wang, Zhi-yuan Wang, Ben-dong Wang, Zhuo-qun Yu, Zhong-he Jin & John L. Crassidis; https://link.springer.com/article/10.1631/FITEE.1900590

[4] - A comparison of machine learning methods for automated gamma-ray spectroscopy; *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment, Volume 954, 21 February 2020, 161385;* Mark Kamuda, Jifu Zhao, Kathryn Huff; https://doi.org/10.1016/j.nima.2018.10.063

[5] Using artificial intelligence and real galaxy images to constrain parameters in galaxy formation simulations; *Monthly Notices of the Royal Astronomical Society, Volume 512, Issue 2, May 2022,* Andrea V Macciò, Mohamad Ali-Dib, Pavle Vulanovic, Hind Al Noori, Fabian Walter, Nico Krieger, Tobias Buck; https://doi.org/10.1093/mnras/stac482

[6] Determining Research Priorities for Astronomy Using Machine Learning; Brian Thomas1, Harley Thronson, Anthony Buonomo, and Louis Barbier; https://iopscience.iop.org/article/10.3847/2515-5172/ac4990/meta

[7] - Color classification of Earth-like planets with machine learning; *Monthly Notices of the Royal Astronomical Society, Volume 504, Issue 4, July 2021;* Dang Pham, Lisa Kaltenegger; https://doi.org/10.1093/mnras/stab1144

[8] - https://www.jpl.nasa.gov/habex/ Habitable Exoplanet Observatory

[9] - SKYNET: an efficient and robust neural network training tool for machine learning in astronomy; *Monthly Notices of the Royal Astronomical Society, Volume 441, Issue 2, 21 June 2014;* Philip Graff, Farhan Feroz, Michael P. Hobson, Anthony Lasenby; https://doi.org/10.1093/mnras/stu642

[10] - Machine Learning in Astronomy: A Case Study in Quasar-Star Classification; *Advances in Intelligent Systems and Computing*; Mohammed Viquar, Suryoday Basak, Ariruna Dasgupta,

Surbhi Agrawal & Snehanshu Saha; https://link.springer.com/chapter/10.1007/978-981-13-1501-5_72

[11] – S interneta: https://www.researchgate.net/figure/Fig-A1-The-standard-VGG-16-network-architecture-as-proposed-in-32-Note-that-only_fig3_322512435

[12] – S interneta: https://medium.com/analytics-vidhya/face-recognition-using-transfer-learning-and-vgg16-cf4de57b9154

[13] – Latentna Dirichletova alokacija, s interneta: https://brightminded.com/blog/intro-to-lda/

[14] – AdaBoost, s interneta: https://www.researchgate.net/figure/Illustration-of-AdaBoost-algorithm-for-creating-a-strong-classifier-based-on-multiple_fig9_288699540

[15] – Introduction to neural network;Avinash Navlani; Prosinac 2019.: https://www.datacamp.com/tutorial/neural-network-models-r

[16] - Variation of average SNR with DM - https://www.researchgate.net/figure/ariation-of-average-SNR-with-DM-The-average-SNR-that-has-a-peak-value-at-DM-1957-pc_fig8_51888058

[17] – Recurrent Neural Net; Deep Learning Library; Veljača 2019. : https://nerdthecoder.wordpress.com/2019/02/03/recurrent-neural-net/

[18] – HTRU2 Data Set; Machine Learning Repository; https://archive.ics.uci.edu/ml/datasets/HTRU2

[19] - Baressi Šegota, S., Lorencin, I., Musulin, J., Štifanić, D., & Car, Z. (2020). Frigate speed estimation using CODLAG propulsion system parameters and multilayer perceptron. NAŠE MORE: znanstveni časopis za more i pomorstvo, 67(2), 117-125.

[20] - Baressi Šegota, S., Lorencin, I., Anđelić, N., Mrzljak, V., & Car, Z. (2020). Improvement of marine steam turbine conventional exergy analysis by neural network application. Journal of Marine Science and Engineering, 8(11), 884.

[21] – S interneta: https://imagine.gsfc.nasa.gov/science/objects/neutron_stars1.html

7. POPIS SLIKA

Slika 2.1.Grafički prikaz konvolucijske neuronske mreže Vgg16 [11] 3
Slika 2.2. Način poretka 3x3 filtara kroz konvolucijsku mrežu Vgg16 [12] 4
Slika 2.3.Grafički model LDA [13] 6
Slika 2.4. Teleskop za promatranje zemljolikih planeta (eng. Habitable Exoplanet Observatory) [8] 7
Slika 2.5. Prikaz AdaBoost algoritma za izradu pojačanog klasifikatora [14] 9
Slika 3.1. Primjer prikaza DM-SNR krivulje [15] 11
Slika 3.2. Primjer sadržaja HTRU2 seta podataka 11
Slika 3.3. Dijagram vrijednosti seta podataka za stupac I 12
Slika 3.4. Dijagram vrijednosti seta podataka integriranih profila 12
Slika 3.5. Dijagram vrijednosti seta podataka DM-SNR krivulje 13
Slika 3.6. Ujednačeni set podataka 18
Slika 3.7. Dijagram vrijednosti ujednačenog seta podataka integriranog profila 19
Slika 3.8. Dijagram vrijednosti ujednačenog seta podataka učitanih iz DM-SNR krivulje 19
Slika 3.9. Dijagram vrijednosti ujednačenog seta podataka klase signala 20
Slika 3.10. Model biološke neuronske mreže [15] 22
Slika 3.11. Model umjetne neuronske mreže 23
Slika 3.12. McCulloch-Pitts model 24
Slika 3.13. Usporedba hiper-ravnina u THU 26
Slika 3.14. Jednoslojna neuronska mreža bez povratnih veza 28
Slika 3.15. Shema mreže s povratnim vezama. [17] 29
Slika 3.16. Višeslojni perceptron 30
Slika 3.17. Područja odlučivanja kod višeslojnih perceptrona 31
Slika 3.18. Linearna aktivacijska funkcija f(x) = x 35
Slika 3.19. Rezultat programskog koda 'logistic' funkcije 37
Slika 3.20. Usporedba odziva za sigmoidalnu i tangens hiperbolnu aktivacijsku funkciju 38
Slika 3.21. Rezultat programskog koda tanh funkcije 39
Slika 3.22. ReLU aktivacijska funkcija 40
Slika 3.23. Rezultati programskog koda za primjenu ReLU aktivacijske funkcije 41
Slika 3.24. Grafički prikaz pulsara [21] 43
Slika 3.25. Rotacija neutronske zvijezde [21] 44
Slika 4.1. Primjer izgleda neuronske mreže 45
Slika 4.2. Usporedba najboljih rezultata pojedinih simulacija za originalni set podataka 55

8. POPIS TABLICA

Tablica 3.1.Najveće vrijednosti za pojedine stupce 14
Tablica 3.2. Najmanje vrijednosti za pojedine stupce 14
Tablica 3.3. Prosječne vrijednosti za pojedine stupce 15
Tablica 3.4. Standardna devijacija za pojedine stupce 15
Tablica 3.5. Deskriptivna statistika za stupce A,B,C i D 16
Tablica 3.6. Deskriptivna statistika za slojeve E,F,G i H 17
Tablica 3.7. Korelacija između pojedinih stupaca 18
Tablica 3.8. Osnovna svojstva podataka ujednačenog seta podataka 20
Tablica 3.9. Deskriptivna statistika za pojedine stupce ujednačenog seta podataka 21
Tablica 3.10. Korelacija između pojedinih stupaca ujednačenog seta podataka 22
Tablica 3.11. Kategorizacija zvijezda glavnog niza (eng. Main sequence stars) 42
Tablica 4.1. Hiperparametri neurona prve simulacije (originalni set podataka) 47
Tablica 4.2. Rezultati prve simulacije (originalni set podataka) 48
Tablica 4.3. Hiperparametri neurona prve simulacije (ujednačeni set podataka) 48
Tablica 4.4. Rezultati prve simulacije (ujednačeni set podataka) 49
Tablica 4.5. Hiperparametri neurona druge simulacije (originalni set podataka) 49
Tablica 4.6. Rezultati druge simulacije (originalni set podataka) 50
Tablica 4.7. Hiperparametri neurona druge simulacije (ujednačeni set podataka) 50
Tablica 4.8. Rezultati druge simulacije (ujednačeni set podataka) 51
Tablica 4.9.Hiperparametri neurona treće simulacije (originalni set podataka) 51
Tablica 4.10. Rezultati treće simulacije (originalni set podataka) 52
Tablica 4.11. Hiperparametri neurona četvrte simulacije (originalni set podataka) 52
Tablica 4.12. Rezultati četvrte simulacije (originalni set podataka) 53
Tablica 4.13. Rezultati četvrte simulacije (ujednačeni set podataka) 53
Tablica 4.14. Hiperparametri neurona pete simulacije (originalni set podataka) 54
Tablica 4.15. Rezultati pete simulacije (originalni set podataka) 54

9. SAŽETAK

Primjenom višeslojnog perceptrona kao jedne od metoda umjetne inteligencije pri treniranju modela, cilj rada je klasifikacija pulsara korištenjem primjerenog seta podataka. Definiranjem umjetnih neuronskih mreža, prikazana je potencijalna uloga koja može uvelike poboljšati klasifikacije astronomskih objekata. Korišteni set podataka HTRU_2 , koji opisuje uzorke detektiranih signala pulsara, detaljno je analiziran i korigiran zbog velikog broja šuma i smetnji u samom setu podataka. Razlog korekcije mogućnost je dobivanja nezadovoljavajuće točnosti treniranja modela. Izradom programskog koda koji implementira klasifikator i set podataka pokreće se simulacija koja daje visoke rezultate točnosti treniranja modela, u granicama od 89% i 94%, ovisno o verziji seta podataka koji je implementiran u programski kod.

Ključne riječi: Umjetna inteligencija, pulsar, višeslojni perceptron, umjetne neuronske mreže, astronomske klasifikacije, treniranje modela

10. SUMMARY

By using a multilayer perceptron as one of the methods of artificial intelligence in model training, the aim of the thesis is the classification of pulsars while using an appropriate data set. Potential role and benefits of artifitial intelligence in astronomy were defined and explained throughout artificial neural networks. Used data set HTRU_2, which describes samples of detected pulsar signals, was analyzed in detail and corrected due to the large number of noise and disturbance in the data set itself. The reason for the correction is the possibility of obtaining unsatisfactory model training accuracy. By creating the program code that implements the classifier and data set, a simulation is started that gives high results of model training accuracy, within the limits of 89% and 94%, depending on the version of the data set implemented in the program code.

Keywords: Artificial intelligence, pulsar stars, multilayer perceptron, artificial neural networks, astronomical classifications, model training

DODATAK A - PROGRAMSKI KOD ZA TRENIRANJE MODELA

```
import numpy as np
import os
import warnings
os.environ['TF_CPP_MIN_LOG_LEVEL'] = '3'
np.seterr(all="ignore")
warnings.filterwarnings('ignore')


from sklearn.model_selection import KFold, GridSearchCV, ShuffleSplit
from sklearn.neural_network import MLPClassifier


import pickle
import uuid

TODO
CHANGE DATA
DATA = np.loadtxt("./HTRU_2_original.csv", skiprows=0, delimiter=',')
descriptor="original"

DATA = np.loadtxt("./HTRU_2.csv", skiprows=0, delimiter=',')
descriptor="balance"
X = DATA[:, 0:8]
Y = DATA[:, 8]
# TODO
params_dict = [{'hidden_layer_sizes':  [(10,10,10,10), (10,10,10), (10,10), (10)],
                'activation': ['logistic','tanh'],
                'solver': ['adam', 'lbfgs'],
                'learning_rate':['constant','adaptive','invscaling'],
                'learning_rate_init': [0.1,0.01,0.5, 0.00001],
                'alpha': [0.01,0.1,0.001, 0.0001],
                'max_iter': [1000]}]


scores = ['accuracy', 'f1']
model = GridSearchCV(MLPClassifier(), params_dict, cv=ShuffleSplit(test_size=0.20,
n_splits=1, random_state=0), n_jobs=8, scoring=scores, verbose=0, refit=False)
model.fit(X,Y)


print(50*"*"+"\n"+50*"*"+"\n"+50*"*"+"\n"+"Fitting
DONE\n"+50*"*"+"\n"+50*"*"+"\n"+50*"*"+"\n"


means1 = model.cv_results_['mean_test_accuracy']
stds1 = model.cv_results_['std_test_accuracy']
means2 = model.cv_results_['mean_test_f1']
stds2 = model.cv_results_['std_test_f1']


uuid_=uuid.uuid4()
file = open(descriptor+"-"+str(uuid_)+"-results.txt", 'w')
file.write("ACC; STD_ACC; F1; STD_F1; Hyperparameters \n")
```

```
for mean1, std1,mean2, std2, params in zip(means1, stds1,means2, stds2,
model.cv_results_['params']):
file.write("%0.20f; %0.020f; %0.20f; %0.020f; %r\n" % (mean1, std1 * 2, mean2,
std2 * 2, params))

file.close()
# model_name = descriptor+"-"+str(uuid_)+".pickle"

print(50*"*"+"\n"+50*"*"+"\n"+50*"*"+"\n"+"DONE\n"+50*"*"+"\n"+50*"*"+"\n"+50*"*"+
"\n")
```

DODATAK B – PROGRAMSKI KOD ZA VIŠESLOJNI PERCEPTRON

```
import pandas
import numpy as np

from math import exp
from matplotlib import pyplot

#tanh activation function
def tanh(x):

      return (exp(x) - exp(-x)) / (exp(x) + exp(-x))

ulaz = [x for x in range(-10, 10)]

izlaz = [tanh(x) for x in inputs]

pyplot.plot(inputs, outputs)
pyplot.show()

Learning_Rate = x

Ulazni_sloj = #()
Skriveni_sloj = #()

broj_epoha = #()

np.random.seed(1)
tezinski_faktor_1 = np.random.uniform(-1, 1, (I_dim, H_dim))
tezinski_faktor_0 = np.random.uniform(-1, 1, H_dim)

preActivation_H = np.zeros(H_dim)
postActivation_H = np.zeros(H_dim)

for epoch in range(broj_epoha):
    for sample in range(broj_treniranja):
        for node in range(skriveni_sloj):
            preActivation_H[node] = np.dot(treniranje[sample,:],
tezinski_faktor0[:, node])
            postActivation_H[node] = logistic(preActivation_H[node])

        preActivation_O = np.dot(postActivation_H, weights_HtoO)
        postActivation_O = logistic(preActivation_O)

FE = postActivation_O - target_output[sample]

        for H_node in range(skriveni_sloj):
            S_greska = FE * logistic_deriv(preActivation_O)
            gradient_HtoO = S_greska * postActivation_H[H_node]

            for I_node in range(I_dim):
                input_value = treniranje[sample, I_node]
                gradient_ItoH = S_greska * tezinski_faktor0[H_node] *
logistic_deriv(preActivation_H[H_node]) * input_value

                tezinski_faktor1[I_node, H_node] -= LR * gradient_ItoH

            tezinski_faktor0[H_node] -= LR * gradient_HtoO
```