Izgradnja radijalnih mreza za klasifikaciju pomoću grupiranja podataka

Falak, Antonio

Master's thesis / Diplomski rad

2018

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **Josip Juraj Strossmayer University of Osijek, Faculty of Electrical Engineering, Computer Science and Information Technology Osijek / Sveučilište Josipa Jurja Strossmayera u Osijeku, Fakultet elektrotehnike, računarstva i informacijskih tehnologija Osijek**

Permanent link / Trajna poveznica: https://urn.nsk.hr/urn:nbn:hr:200:288429

Rights / Prava: In copyright / Zaštićeno autorskim pravom.

Download date / Datum preuzimanja: **2024-03-28**

Repository / Repozitorij:

Faculty of Electrical Engineering, Computer Science and Information Technology Osijek

zir.nsk.hr

Image not found or type unknown

SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU

FAKULTET ELEKTROTEHNIKE, RAČUNARSTVA I INFORMACIJSKIH TEHNOLOGIJA

Sveučilišni diplomski studij

Informacijske i podatkovne znanosti

IZGRADNJA RADIJALNIH MREŽA ZA KLASIFIKACIJU POMOĆU GRUPIRANJA PODATAKA

Diplomski rad

Antonio Falak

Osijek, 2018.

Sadržaj

1. UVOD 1

2. RADIJALNE MREŽE ZA KLASIFIKACIJU 2

2.1 Kratki uvod u problem klasifikacije 2

2.2 Radijalne mreže 3

3. IZGRADNJA RADIJALNIH MREŽA ZA KLASIFIKACIJU 9

3.1 Određivanje parametara radijalnih mreža 9

3.1.1 Kratki uvod u grupiranje podataka i algoritam *k*-means 9

3.1.2 Određivanje centara radijalnih funkcija 11

3.1.3 Određivanje širina radijalnih funkcija 12

3.1.4 Određivanje težina veza neurona skrivenog i izlaznog sloja 13

3.2 Vrednovanje kvalitete mreže 14

3.3 Određivanje veličine radijalne mreže 17

4. OSTVARENO PROGRAMSKO RJEŠENJE 19

4.1 Način rada programskog rješenja 19

4.1.1 Pozadinski dio - radijalna neuronska mreža 19

4.1.2 Grafički radni okvir 22

4.2 Prikaz i način uporabe programskog rješenja 24

5. EKSPERIMENTALNA ANALIZA 27

5.1 Postavke eksperimenta 28

5.2 Rezultati 29

6. ZAKLJUČAK 45

Literatura

Sažetak

Životopis

Prilozi

1. UVOD

Zbog lakšeg preživljavanja i snalaženja u svijetu, ljudski je mozak razvio sposobnost klasificiranja bića, stvari i pojava u određene kategorije tj. klase. Klasifikacija je proces koji se primjenjuje i u računarstvu, a obuhvaća razdvajanje zapažanja, odnosno podataka u diskretni broj klasa na osnovi njihovih karakteristika. Klasifikacija posjeduje zamjetnu primjenu u mnogim područjima kao što su medicina, statistika i strojno učenje. Postoje različiti algoritmi koji se upotrebljavaju za problem klasifikacije, od kojih svaki algoritam ima svoje prednosti i nedostatke, a njihova učinkovitost ovisi o podacima za koje se primjenjuju. U diplomskom radu riješen je problem klasifikacije pristupom umjetne neuronske mreže (engl. *Artificial Neural Network*, ANN). Korištena je vrsta neuronske mreže nazvana radijalna neuronska mreža (engl. *Radial Basis Function Network, RBFN*), koja svojom jednostavnom troslojnom arhitekturom vrši klasifikaciju predstavljajući ulazni prostor s radijalnim funkcijama. Radijalne funkcije (engl. *Radial Basis Function*, RBF) su aktivacijske funkcije neurona skrivenog sloja na temelju kojih neuronska mreža odlučuje pripadnost podatka pojedinoj klasi. Izazovi izgradnje radijalne neuronske mreže obuhvaćaju određivanje prikladne veličine mreže tj. broj neurona u skrivenom sloju mreže, kako bi se prikladno predstavio ulazni prostor podataka. Također je potrebno odrediti parametre aktivacijske funkcije s težinama koje povezuju skriveni i izlazni sloj. Razvijeno je mnoštvo metoda ili pristupa za određivanje parametara koji opisuju mrežu. Važnu ulogu u određivanju parametara ima i grupiranje podataka (engl. *clustering*) koje pronalazi grupe sličnih podatka u svrhu prikladnijeg postavljanja radijalnih funkcija, a samim time i potencijalnog povećanja učinkovitosti klasifikatora.

Poglavlje 2 diplomskog rada obuhvaća detaljniji opis radijalnih neuronskih mreža, njihovu arhitekturu i njihovu primjenu. Specificirane su sličnosti i razlike s drugom vrstom umjetne neuronske mreže zvanom višeslojni perceptron (engl. *Multi-Layer Perceptron*, MLP). Poglavlje 3 sadrži teorijsku podlogu za izgradnju radijalnih neuronskih mreža za klasifikaciju, što uključuje pronalaženje svih potrebnih parametara mreže te je predstavljen način na koji je određen prikladan broj neurona u skrivenom sloju. Također su opisani načini vrednovanja kvalitete mreže i različite mjere učinkovitosti izgrađenog klasifikatora. Poglavlje 4 opisuje ostvareno programsko rješenje, a to obuhvaća radijalnu neuronsku mrežu i grafički radni okvir. U poglavlju 5 dana je eksperimentalna analiza kojom je testirana neuronska mreža na pet skupova podataka u smislu pokazivanja utjecaja na klasifikaciju kada se koriste različiti pristupi određivanja parametara neuronske mreže.

2. RADIJALNE MREŽE ZA KLASIFIKACIJU

Radijalna neuronska mreža je troslojni mrežni model koji rješava problem klasifikacije, a naziv je dobila po vrsti aktivacijske funkcije koju mreža koristi. RBFN pristupa problemu klasifikacije tako što nastoji prikladno opisati skup podataka aktivacijskim funkcijama, a klasificira nove podatke tako što izračunava sličnost između parametra aktivacijske funkcije i novog podatka. Osim što se koristi za klasifikaciju, RBFN ima zamjetnu primjenu u obradi slike, prepoznavanju govora, medicinskoj dijagnostici, lokalizaciji izvora kod radara, analizi stohastičkih signal i drugom.

2.1 Kratki uvod u problem klasifikacije

Bishop [1] predstavlja problem klasifikacije kao zadatak kojemu je cilj dodijeliti nove podatke jednoj od diskretnih klasa ili kategorija. Cilj klasifikacije je dodijeliti ulazni vektor, odnosno podatak x_p jednoj od K diskretnih klasa. U pravilu, klase su disjunktne (engl. *disjoint*), tako da je svaki ulaz dodijeljen samo jednoj klasi. Klasifikator se može predstaviti funkcijom $f(x)$ koja preslikava s ulaza *x* na izlaze *y* (kao što je prikazano na slici 2.1), gdje je $y \in \{y_1, \ldots, y_K\}$, a K predstavlja broj klasa. Ako je K=2, to se naziva binarna klasifikacija, a ako je K>2, to se naziva višeklasna klasifikacija. Zadatak klasifikacije je donijeti predviđanja na novim ulaznim podacima, što znači na podacima neviđenim do sada (to se naziva generalizacija), budući da je lako predvidjeti odziv na skupu podataka za treniranje (jednostavno se provjeri rješenje) [2].



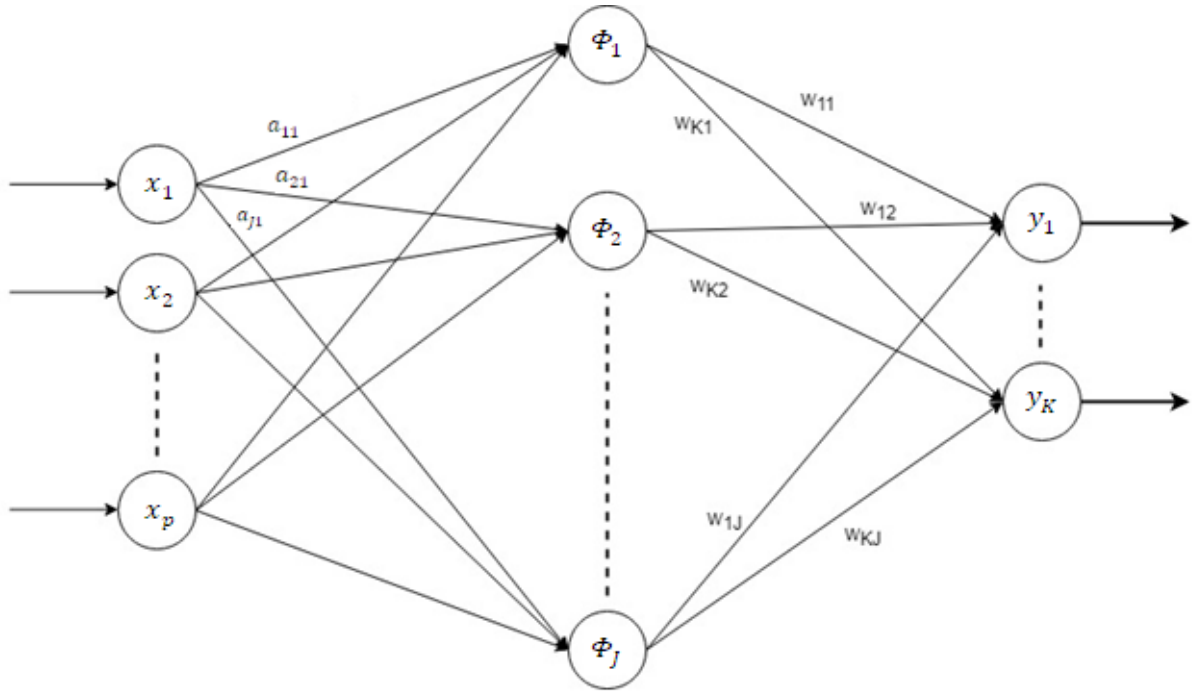
Sl. 2.1: Skica klasifikatora

Slijedi nekoliko zanimljivih primjera za ilustriranje rasprostranjene primjene klasifikacije [3]:

- detekcija neželjene pošte − klasifikacija pridošle pošte u dvije klase koje predstavljaju normalnu poštu i neželjenu poštu (engl. *spam*)
- prepoznavanje rukom napisanih brojki − ulazne podatke predstavlja skup slika gdje su rukom napisane brojke (npr. od 1 do 9), a cilj klasifikacije je odrediti kojem broju nova slika pripada
- segmentacija slike−svrstavanje dijelova slike u predefinirane klase zbog transformiranja slike za lakšu analizu
- raspoznavanje govora − višeklasni klasifikacijski problem gdje je cilj prepoznati kojoj riječi ulazni podatak pripada

2.2 Radijalne mreže

Arhitektura RBFN (kao na slici 2.2) se sastoji od ulaznog sloja, sloja neurona u skrivenom sloju (ili RBF neurona) i izlaznog sloja s jednim neuronom po kategoriji (klasi) podataka. Ulazni i skriveni sloj povezuje prvi sloj težina koje posjeduju jediničnu vrijednost, dok skriveni i izlazni sloj povezuje drugi sloj težina. Ulazni sloj prenosi parametre ulaznog vektora (podatak u RBFN) prema svakom neuronu skrivenog sloja. Svaka jedinica u skrivenom sloju zatim proizvodi aktivaciju zasnovanu na pridruženoj radijalnoj funkciji tj. svaki skriveni neuron izračunava razinu sličnosti između parametra aktivacijske funkcije i ulaznog vektora. Konačno, svaka jedinica u izlaznom sloju izračunava linearnu kombinaciju aktivacija skrivenih neurona.



Sl. 2.2: Arhitektura RBFN

Radijalna neuronska mreža je neuronska mreža s propagacijom prema naprijed, gdje skrivene jedinice koriste radijalne funkcije kao aktivacijske funkcije. Propagacija prema naprijed znači da se podaci kreću od ulaznog sloja prema skrivenom sloju, a zatim iz skrivenog prema izlaznom sloju mreže. Ulazni sloj je podatak koji u RBFN predstavlja n-dimenzionalni vektor koji se pokušava klasificirati, gdje n predstavlja broj atributa koji opisuju podatke unutar skupa. Cijeli je ulazni vektor prikazan svakom od RBF neurona. U [4] je navedeno da svaki RBF neuron posjeduje radijalnu funkciju koju određuju dva parametra, a to su centar i širina radijalne funkcije. Centar radijalne funkcije je n-dimenzionalni vektor smješten u ulaznom prostoru podataka, a može biti predstavljen određenim ulaznim vektorom, dok je širina skalar koji opisuje djelokrug aktivacijske funkcije oko centra. Svaki RBF neuron uspoređuje ulazni vektor s centrom radijalne funkcije te donosi vrijednost između nula i jedan, što predstavlja mjeru sličnosti. Ako je ulaz jednak centru, izlaz RBF neurona će biti jedan. Kako udaljenost između ulaznog vektora i centra raste, odziv eksponencijalno pada prema nuli. Oblik aktivacijske funkcije RBF neurona predstavljen je Gaussovom krivuljom, gdje je centar radijalne funkcije (μ_j) u središtu Gaussove krivulje, a širina Gaussove krivulje predstavlja parametar širine (σ_j) radijalne funkcije.

Svaki RBF neuron tj. neuron skrivenog sloja posjeduje svoju radijalnu funkciju. Postoje različiti mogući izbori radijalnih funkcija, ali najčešće korištena je ona temeljena na Gaussovoj krivulji. Gaussova aktivacijska funkcija RBF neurona se obično zapisuje kao:

$$\Phi_j(x) = e^{\frac{-||x_p-\mu_j||_2^2}{2\sigma_j^2}} \tag{2-1}$$

Oznaka x_p predstavlja ulazni vektor koji je doveden na RBF neuron. Oznaka μ_j predstavlja srednju vrijednost razdiobe tj. centar radijalne funkcije, a σ_j predstavlja standardnu devijaciju ili širinu radijalne funkcije. Zapis s dvostrukom linijom u aktivacijskoj funkciji označava da se uzima euklidska udaljenost između x_p i μ_j. Važno je zapaziti da je ovdje osnovni mjerni podatak za vrednovanje sličnosti između ulaznog vektora i centra euklidska udaljenost između ta dva vektora. Eksponencijalni pad aktivacijske funkcije znači da će neuroni čiji je centar daleko od ulaznog vektora malo doprinijeti rezultatu [4].

Engelbrecht je u [5] ponudio brojne druge radijalne funkcije:

• linearna funkcija,

$$\Phi(||x_p - \mu_j||_2) = ||x_p - \mu_j||_2 \tag{2-2}$$

• kubna funkcija,

$$\Phi(||x_p - \mu_j||_2) = ||x_p - \mu_j||_2^3 \tag{2-3}$$

• *thin-plate-spline* funkcija,

$$\Phi(||x_p - \mu_j||_2) = ||x_p - \mu_j||_2^2 ln||x_p - \mu_j||_2 \tag{2-4}$$

• multikvadratna funkcija,

$$\Phi(||x_p - \mu_j||_2, \sigma_j) = \sqrt{||x_p - \mu_j||_2^2 + \sigma_j^2} \tag{2-5}$$

• inverzna multikvadratna funkcija,

$$\Phi(||x_p - \mu_j||_2, \sigma_j) = \frac{1}{\sqrt{||x_p-\mu_j||_2^2+\sigma_j^2}} \tag{2-6}$$

• Gaussova funkcija,

$$\Phi(||x_p - \mu_j||_2, \sigma_j) = e^{-||x_p-\mu_j||_2^2/(2\sigma_j^2)} \tag{2-7}$$

• logistička funkcija,

$$\Phi(||x_p - \mu_j||_2, \sigma_j) = \frac{1}{1 + e^{||x_p-\mu_j||_2^2/\sigma_j^2-\theta_j}} \tag{2-8}$$

gdje je θ_j pristranost (engl. *bias*).

Izlaz mreže sastoji se od skupa neurona, jednog po kategoriji koja se pokušava klasificirati. Svaki izlazni neuron izračunava svojevrsni rezultat za pripadajuću kategoriju. Obično se odluka klasifikacije donosi dodjeljivanjem ulaznog vektora kategoriji s najvećim rezultatom. Neuroni izlaznog sloja za svaku klasu uzimaju linearnu kombinaciju svih RBF neurona u mreži – drugim riječima, svaki će neuron u mreži imati utjecaj na odluku klasifikacije. Linearna kombinacija znači da izlazni neuron pridružuje vrijednost težine sa svakim RBF neuronom te množi izlaz RBF neurona s tom težinom prije zbrajanja s ukupnim rezultatom [4]. Izlaz iz neurona izlaznog sloja mreže je linearna kombinacija radijalnih funkcija [5]:

$$y_{k,p} = \sum_{j=1}^{J+1} w_{kj} \Phi_{j,p}\,, \tag{2-9}$$

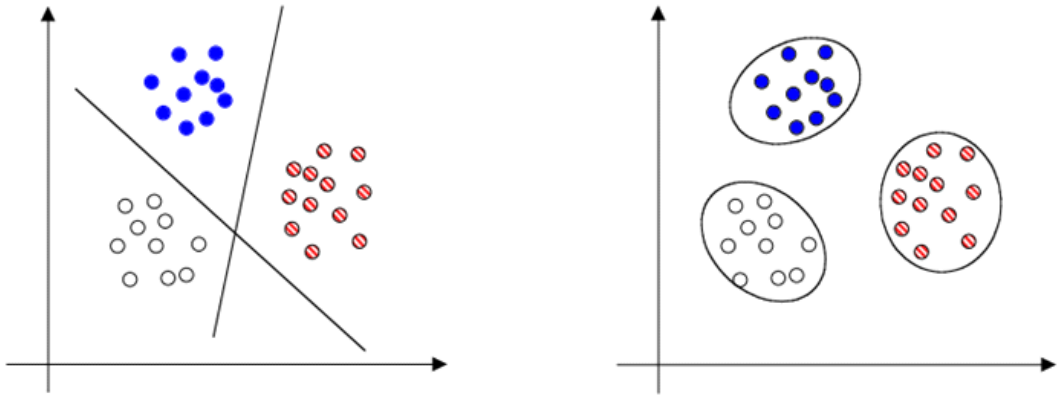
gdje oznaka $y_{k,p}$ predstavlja vrijednosti izlaznih neurona koji ovise o težinama između skrivenog i izlaznog sloja w_{kj} te o izlazima iz RBF neurona $\Phi_{j,p}$. Naspram prvog sloja težina koje povezuju ulazni i skriveni sloj neuronske mreže, drugi sloj težina ne posjeduje jedinične vrijednosti. Težine između skrivenog i izlaznog sloja su parametar radijalne neuronske mreže koji se izračunava metodama linearne algebre.

Kao što je kod drugih neuronskih mreža s propagacijom prema naprijed, pokazano je da su i RBFN univerzalni aproksimatori [5], a na čiju učinkovitost utječu:

- broj korištenih radijalnih funkcija. Što se više radijalnih funkcija koristi, bolja će biti aproksimacija ciljane funkcije. Ipak, nepotrebne radijalne funkcije povećavaju složenost izračunavanja.
- lokacija radijalne funkcije, definirana centrom, μ_j, za svaku radijalnu funkciju. Radijalne funkcije bi trebale biti ravnomjerno raspoređene kako bi pokrivale cijeli ulazni prostor podataka.
- za neke funkcije, širina receptivnog polja, σ_j . Što je veća širina σ_j , više je ulaznog prostora zastupljeno tom radijalnom funkcijom.

Treniranje RBFN bi stoga trebalo razmotriti metode za pronalazak najbolje vrijednosti za ove parametre.

Višeslojni perceptron, druga vrsta umjetne neuronske mreže, razdvaja klase hiper-ravninama u ulaznom prostoru (kao što je prikazano lijevo na slici 2.3). Alternativni bi pristup bio oblikovati ulazni prostor lokaliziranim radijalnim funkcijama (kao što je desno na slici 2.3) [6].



Sl. 2.3: Razlika u razdvajanju klasa između MLP i RBFN

Kao dvije vrste umjetnih neuronskih mreža, radijalna mreža i višeslojni perceptron dijele određene sličnosti, a prema [6] to su:

- obje su nelinearne mreže s propagacijom prema naprijed
- obje su univerzalni aproksimatori
- upotrebljavaju se u sličnim područjima primjene

S obzirom na te sličnosti, nije iznenađujuće da uvijek postoji RBFN sposobna precizno oponašati određen MLP ili obrnuto. Unatoč tomu, Bullinaria je u [6] opisao međusobnu razliku tih dviju vrsta umjetnih neuronskih mreža u brojnim važnim pogledima:

- RBFN ima jedan skriveni sloj, dok ih MLP mogu imati više.
- RBFN obično su potpuno spojene, dok su MLP često djelomično spojeni.
- u MLP neuroni u različitim slojevima mreže mogu posjedovati različite aktivacijske funkcije. U RBFN skriveni neuroni imaju različitu funkciju i svrhu od izlaznih neurona.
- u RBFN, argument svake aktivacijske funkcije skrivenog neurona je udaljenost između ulaznog vektora i centra, dok je u višeslojnim mrežama to skalarni produkt ulaznog vektora i težina
- MLP se uglavnom trenira s jednim globalnim nadziranim (engl. *supervised*) algoritmom, dok se kod RBFN obično trenira sloj po sloj, s tim da je prvi sloj nenadziran (engl. *unsupervised*).

Aktivacijska funkcija skrivenog neurona i njezini parametri utvrđuju "veličinu" djelokruga neurona određivanjem težine utjecaja ulaznog vektora s obzirom na njegovu udaljenost od centra. Izlazni sloj radijalne mreže služi za kombinaciju aktivacija skrivenih neurona u izlaz mreže, slično djelovanju višeslojnih perceptrona. Međutim, aktivacijska funkcija izlaznih neurona u radijalnoj mreži je linearna funkcija, dana jednadžbom (2-9) [8].

3. IZGRADNJA RADIJALNIH MREŽA ZA KLASIFIKACIJU

Kod radijalnih neuronskih mreža u pravilu se razmatraju tri sloja mreže: ulazni, skriveni i izlazni sloj. Prije klasifikacije je potrebno izvršiti predobradu podataka tj. izbaciti stupce koji nisu potrebni za klasifikaciju, ukloniti nepotpune redove i izvršiti normalizaciju podataka. Kod modeliranja radijalne neuronske mreže potrebno je odrediti niz parametara kako bi klasifikator postigao što bolje rezultate. Potrebno je odrediti parametre aktivacijskih funkcija u neuronima skrivenog sloja, težine koje povezuju skriveni i izlazni sloj, kao i odgovarajuću veličinu mreže. Određivanje centara radijalnih funkcija može se izvršiti na više načina, a objašnjena su dva pristupa, od kojih je jedan nasumičan odabir centara iz skupa podataka za treniranje mreže. Drugi način odabiranja prikladnih centara zasniva se na algoritmu za grupiranje podataka zvanom *k*-means.

3.1 Određivanje parametara radijalnih mreža

Za izgradnju radijalne neuronske mreže za klasifikaciju potrebno je odrediti sljedeće parametre:

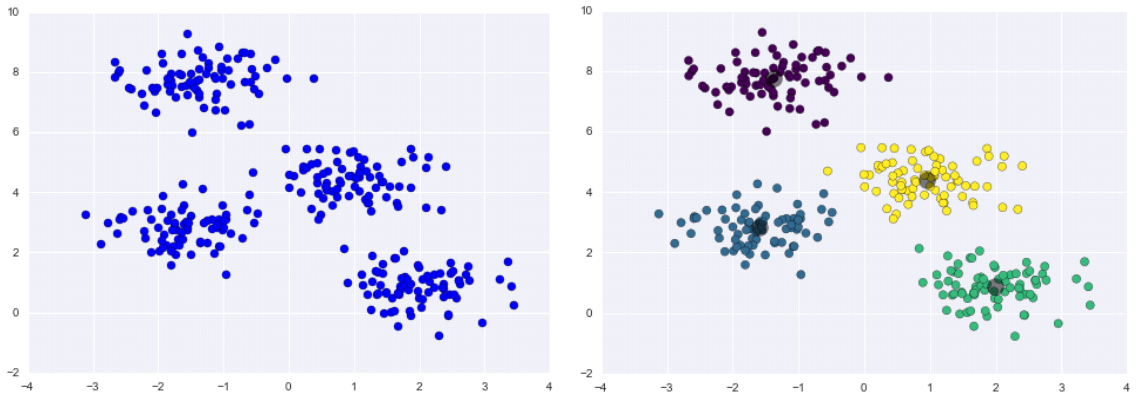
- prikladnu veličinu mreže
- centre radijalnih funkcija
- širine radijalnih funkcija
- težine veza između skrivenog i izlaznog sloja

3.1.1 Kratki uvod u grupiranje podataka i algoritam *k*-means

Grupiranje podataka je proces razdvajanja podataka u grupe, tako da svaka grupa sadrži međusobno slične podatke. Xu i Wunsch [7] opisuju da se kod grupiranja podataka prvobitna grupa podataka dijeli u manje homogene grupe po načelu odabrane mjere sličnosti. Mjera sličnosti se subjektivno odabire kako bi se mogle opisati "zanimljive" grupe podataka. Štoviše, drugačiji odabir parametara za isti algoritam grupiranja može rezultirati različitim rezultatima grupiranja. Primjerice, ljudska bića se mogu klasificirati na osnovi dobi, regije, obrazovanja, visine, težine i drugim osnovama. Sličnost podataka se najčešće određuje korištenjem funkcija udaljenosti.

Algoritam *k*-means [7, 8, 9, 10] pripada nenadziranim algoritmima učenja. U nenadziranom grupiranju, oznake klasa ne postoje na raspolaganju. Cilj grupiranja je odrediti grupe podataka gdje podaci u svakoj grupi dijele slične karakteristike, a oznaka *k* u nazivu algoritma predstavlja broj takvih grupa. Algoritam *k*-means iterativno svrstava svaki podatak u jednu od *k* grupa.

Algoritam je jednostavan. U početku se *k* broj centara grupa odabere nasumično, a tada se podaci podijele u *k* grupa dodjeljujući svakom centru grupe sve podatke koji su bliži danoj grupi nego bilo kojem centru druge grupe. U drugom se koraku izračunavaju novi centri grupa pronalazeći srednju vrijednost (engl. *mean*, po čemu algoritam nosi naziv) formirane grupe podataka [8]. Na slici 3.1 lijevo je predstavljen ulazni prostor neoznačenih podataka, a desno se vidi podjela podataka korištenjem algoritma *k*-means gdje je *k*=4.

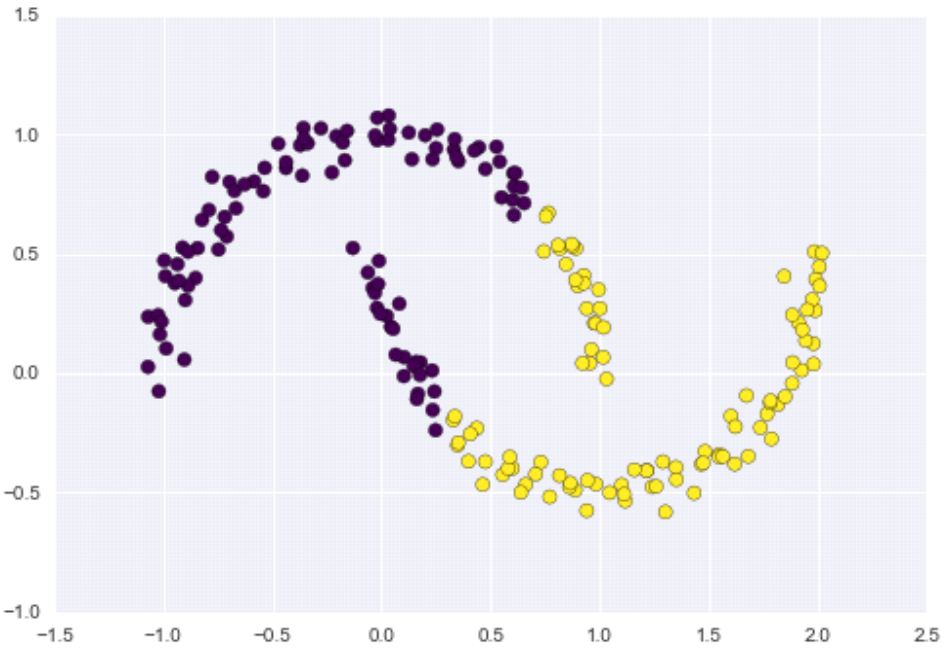


Sl. 3.1: Primjer grupiranja podataka pomoću algoritma *k*-means [9]

Uobičajena mjera sličnosti je euklidska udaljenost u čijem se slučaju može pokazati da će se nenegativna funkcija koja se nastoji minimizirati,

$$\sum_{p=1}^{N}\left(\min ||x_p - c_j||_2^2\right) , \tag{3-1}$$

smanjivati prilikom svakog preseljenja srednjih vrijednosti, stoga je konvergencija zajamčena u konačnom broju iteracija [10]. Jednadžba (3-1) predstavlja kriterij kojeg se nastoji minimizirati, a služi za procjenu kvalitete grupiranja podataka. Pretpostavka *k*-means algoritma da će točke biti bliže svom centru grupe nego ostalim centrima znači da će algoritam često biti bezuspješan ako grupe posjeduju kompleksne geometrije kao na slici 3.2.



Sl. 3.2: Podbacivanje algoritma *k*-means kod kompleksnih oblika [9]

Algoritam je također osjetljiv na prisustvo stršećih podataka (engl. *outliers*), stoga uklanjanje stršećih podataka može znatno pomoći. Naknadna obrada rezultata za, primjerice, elimininiranje malih grupa ili spajanje bliskih grupa u veliku grupu je također poželjna [10]. Svaka iteracija algoritma *k*-means mora pristupiti svakom podatku u skupu, stoga algoritam može biti relativno spor kod velikih skupova podataka.

Wu et al. [10] naznačuju da unatoč svojim nedostacima, *k*-means ostaje najčešće korišten algoritam za grupiranje u praksi. Algoritam je jednostavan, lako razumljiv i razumno skalabilan te se lako može modificirati za tok (engl. *stream*) podataka. Stalna poboljšanja i generaliziranje osnovnog algoritma osigurali su trajnu značajnost te postupno povećavanje efektivnosti algoritma.

3.1.2 Određivanje centara radijalnih funkcija

Osim parametra o broju centara radijalnih funkcija, važno je odrediti poziciju navedenih centara kako bi radijalne funkcije prikladno opisale ulazni prostor podataka. Razmatraju se dva pristupa pri pozicioniranju centara, od kojih je jedan nasumično odabiranje J ulaznih vektora iz skupa za treniranje, gdje J predstavlja broj centara radijalnih funkcija. Jednostavna procedura odabiranja centara je postavljanje nasumičnog podskupa ulaznih vektora iz skupa za treniranje za centre

radijalnih funkcija. Bishop [11] nalaže da ovo nije optimalni pristup što se tiče opisivanja ulaznog prostora podataka, a osim toga može zahtijevati i nepotrebno velik broj radijalnih funkcija kako bi se postigle prihvatljive performanse pri treniranju. Ova metoda ima veliku standardnu devijaciju rezultata zato što ovisi o kvaliteti odabira ulaznih vektora. Primjerice, ako se nasumično odaberu centri radijalnih funkcija koji ne opisuju dobro podatkovni skup, učinkovitost će biti izrazito niska.

Pogodniji bi način za odabiranje centara radijalnih funkcija bio pomoću algoritama za grupiranje, zato što nasumičan odabir nije ujedno i jamstvo da će centri opisivati podatkovnu raspodjelu ulaznih podataka dovoljno dobro da bi prikladno aproksimirali željene izlaze [8]. Drugi pristup se odnosi na korištenje algoritma *k*-means za odabir centara kako bi se preciznije opisala raspodjela podataka u skupu. Algoritam traži najbolje pozicije za centre radijalnih funkcija nad skupom podataka za treniranje koji ne sadrže informaciju kojoj klasi svaki ulazni vektor pripada. Algoritam za grupiranje podataka poput algoritma *k*-means predstavlja poboljšanje učinkovitosti radijalne mreže jer ne ovisi o slučajnom odabiru centara poput prethodne metode.

3.1.3 Određivanje širina radijalnih funkcija

Širina radijalne funkcije ili širina receptivnog polja σ_j predstavlja radijalno polje oko svakog centra radijalne funkcije. Što je σ_j veći, više ulaznog prostora je obuhvaćeno svakom radijalnom funkcijom. U radu su obuhvaćena dva načina izračuna širina radijalnih funkcija. Prvi način koristi najveću euklidsku udaljenost između centara kako bi se izračunala univerzalna širina za svaki centar, a prema [12] to je:

$$\sigma_j = \sigma = \frac{d_{max}}{\sqrt{2J}}, j = 1, \dots, J \tag{3-2}$$

gdje J predstavlja broj centara radijalnih funkcija (ili neurona skrivenog sloja), a d_{max} je najveća euklidska udaljenost između bilo kojeg para centara. Prema Engelbrechtu [5], ovaj izbor bi bio blizu optimalnog rješenja da su podaci uniformno raspoređeni u ulaznom prostoru.

Benoudjit i Verleysen [13] tvrde da, ako udaljenosti između centara nisu jednake, bilo bi bolje dodijeliti zasebnu širinu svakom centru. Primjerice, razumno je dodijeliti veće širine centrima koji su međusobno udaljeniji, a manje širine onima koji su međusobno bliži. Drugi algoritam

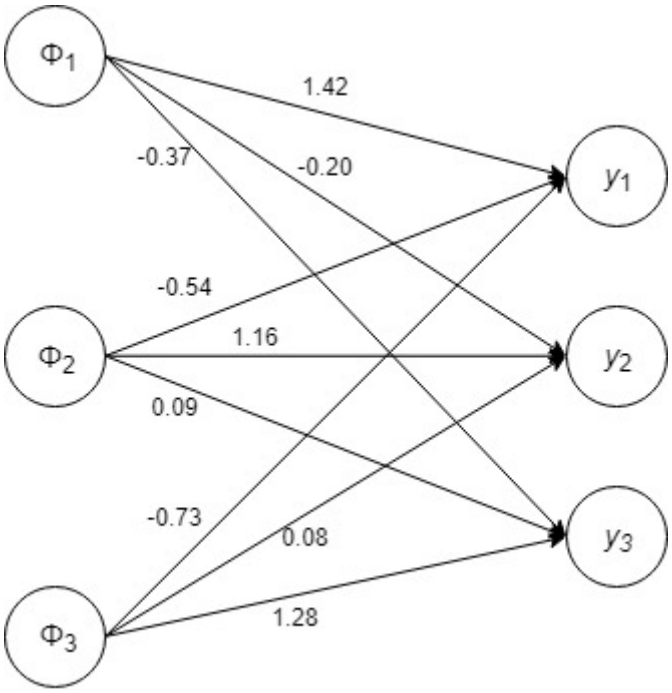
prema [13] pronalazi širinu za svaki centar posebno pomoću pravila p-najbližih susjeda (engl. *p-nearest neighbours*):

$$\sigma_j = \frac{1}{p}(\sum_{i=1}^{p} ||c_j - \mu_i||^2)^{\frac{1}{2}} \tag{3-3}$$

gdje je c_j centar za kojeg se traži p-najbližih susjeda μ_i, dok je predložena vrijednost za p jednaka dva.

3.1.4 Određivanje težina veza neurona skrivenog i izlaznog sloja

Izlazni sloj se sastoji od K neurona, gdje K određuje skup podataka s kojim se radi, a predstavlja broj klasa u skupu. Skriveni i izlazni sloj povezuju težine koje nemaju jediničnu vrijednost kao u prvom sloju težina. Rezultat za svaki neuron izlaznog sloja se dobiva sumom težina sa svakim izlazom iz skrivenog sloja. Bishop [1] navodi da se radijalna neuronska mreža obično trenira u dvije faze, s radijalnim funkcijama koje se određuje nenadziranim tehnikama koristeći podatke za treniranje, a zatim se težine između skrivenog i izlaznog sloja pronalaze brzim linearnim nadziranim metodama. Zato što svaki izlazni neuron računa rezultat za drugu klasu, svaki izlazni neuron ima svoj skup težina. Izlazni neuron obično dodjeljuje pozitivnu težinu RBF neuronima koji pripadaju njegovoj klasi, a negativnu težinu drugim RBF neuronima [4]. Na slici 3.3 je primjer izračunatih težina u mreži s tri neurona u skrivenom sloju i tri neurona u izlaznom sloju (koji predstavljaju tri klase). Može se uočiti da težine koje povezuju prvi neuron izlaznog sloja daju najveći koeficijent prvoj klasi, dok druge dvije klase obično imaju koeficijent blizu nuli ili negativan koeficijent.



Sl. 3.3: Primjer izračunatih težina koje povezuju skriveni i izlazni sloj

Prema Broomheadu i Loweu [14], ulazna vrijednost svih neurona izlaznog sloja je linearna kombinacija svih izlaza neurona skrivenog sloja, gdje je vrijednost veze iz skrivenog neurona *j* prema izlaznom neuronu *k* označena s w_{kj}. Problem određivanja preciznih vrijednosti težina svodi se na linearnu optimizaciju najmanjih kvadrata koja posjeduje "zagarantiran algoritam učenja" putem metode pseudo-inverza [14].

Kruse et al. [8] opisuju da izlaze iz neurona skrivenog sloja uobičajeno predstavlja nekvadratna matrica na kojoj se izračunava tzv. Moore-Penrose pseudo-inverz matrice:

$$A^{+} = (A^{T}A)^{-1}A^{T} \tag{3-4}$$

Težine između skrivenog i izlaznog sloja se računaju pomoću jednadžbe:

$$w_{kj} = A^{+} \cdot y_k = (A^{T}A)^{-1}A^{T} \cdot y_k \tag{3-5}$$

3.2 Vrednovanje kvalitete mreže

Radijalna neuronska mreža prolazi kroz dva procesa evaluacije. Prvi proces uključuje samo podatke za treniranje te se koristi za određivanje prikladne veličine mreže. Drugi proces evaluacije se provodi pomoću skupa podataka za testiranje, a služi za vrednovanje klasifikatora. Generalizacija je veoma važan aspekt učenja neuronske mreže. Generalizacija je mjera koliko

dobro mreža vrši klasifikaciju nad podacima koji nisu korišteni pri treniranju mreže, a konačni cilj učenja neuronske mreže je proizvesti model s niskom pogreškom generalizacije. Stoga, cilj neuronske mreže je dobro naučiti podatke iz skupa za treniranje, ali još uvijek omogućavati dobru generalizaciju nad podacima izvan skupa za treniranje. Međutim, postoji mogućnost da neuronska mreža posjeduje manje odstupanje prilikom treniranja, ali loše svojstvo generalizacije zbog pretjerane prilagođenosti podacima za treniranje [5].

Prema [5], Engelbrecht navodi da je najčešća mjera odstupanja od točnih vrijednosti srednja kvadratna pogreška (engl. *mean squared error, MSE*) kod koje se pogreška treniranja izračunava poput:

$$MSE = \frac{\sum_{p=1}^{P_T} \sum_{k=1}^{K} \left(t_{k,p} - o_{k,p}\right)^2}{P_T K} \tag{3-6}$$

gdje je P_T ukupni broj podataka za treniranje u skupu D_T, a K je broj izlaznih neurona. Vrijednost $t_{k,p}$ predstavlja očekivani rezultat izlaznog neurona, a vrijednost $o_{k,p}$ predstavlja stvarnu vrijednost neurona. Umjesto MSE, može se koristiti zbroj kvadratnih pogreški (engl. *sum squared error, SSE*):

$$SSE = \sum_{p=1}^{P} \sum_{k=1}^{K} \left(t_{k,p} - o_{k,p}\right)^2 \tag{3-7}$$

gdje je P ukupni broj podataka u promatranom skupu. Međutim, SSE nije dobra mjera kad se uspoređuju performanse nad skupovima različitih veličina [5].

Kod problema klasifikacije, postotak točno klasificiranih (ili netočno klasificiranih) podataka se koristi kao mjera točnosti. Razlog zašto MSE nije dobra mjera, navodi Engelbrecht [5], je taj što mreža može imati visoku točnost u broju točnih klasifikacija, ali pritom imati velik MSE. Ako se samo MSE koristi za indikaciju prestanka treniranja mreže, to može rezultirati predugim treniranjem mreže kako bi se postigao nizak MSE, dakle, uzaludno trošenje vremena i povećavanje pretjerane prilagođenosti podacima za treniranje.

U [15] je predstavljen način za prikazivanje rezultata predviđanja pomoću matrice zbunjenosti (engl. *confusion matrix* ili *contingency table*). Za binarni problem klasifikacije matrica ima dva reda i dva stupca. Stupci predstavljaju predviđene klase, dok redovi predstavljaju stvarne klase. Svaki element matrice predstavlja broj predviđenih podataka klasifikatora koji pripadaju toj

kategoriji. Na slici 3.4 je dan općeniti oblik matrice zbunjenosti predstavljen kao tablica istinitosti u slučaju kad postoje dvije klase podataka.

	Pozitivno	Negativno
Pozitivno	Pravi pozitiv	Lažni pozitiv
Negativno	Lažni negativ	Pravi negativ

Sl. 3.4: Matrica zbunjenosti za dvije klase

Postoje razni načini za vrednovanje performansi klasifikatora. Matrica zbunjenosti $A = [A(i,j)]$ je definirana tako da je element $A(i,j)$ broj koji ukazuje broj podataka čija je prava klasa i te su klasificirani u klasu j. Iz matrice A, moguće je direktno izračunati opoziv (engl. *recall*), preciznost (engl. *precision*) i F1 mjeru (engl. *F1 score* ili *F1 measure*) za svaku klasu, s ukupnom točnošću na način [16]:

- Opoziv (R_i) – podaci sa stvarnom klasom i koji su točno klasificirani u tu klasu tj. točnost klasifikatora u označavanju pravih pozitiva. Nizak opoziv ukazuje na puno lažnih negativa, a izračunava se kao:

$$R_i = \frac{Pravi\ pozitiv}{Pravi\ pozitiv + Lažni\ negativ} \tag{3-8}$$

- Preciznost (P_i) – podaci klasificiranih u klasu i čija je stvarna klasa uistinu i tj. koliko ima točnih pozitiva u skupu svih pozitiva. Niska preciznost ukazuje na velik broj lažnih pozitiva. Izračunava se kao:

$$P_i = \frac{Pravi\ pozitiv}{Pravi\ pozitiv + Lažni\ pozitiv} \tag{3-9}$$

- F1 mjera – izračunava se kao

$$F1 = \frac{2}{\frac{1}{R_i} + \frac{1}{P_i}} = 2\left(\frac{P_i * R_i}{P_i + R_i}\right) \tag{3-10}$$

  Drugim riječima, F1 mjera prikazuje ravnotežu između preciznosti i opoziva.

- Ukupna točnost (A_C) – postotak podataka koji su točno klasificirani. Za dani K-klasni problem, A_C se izračunava iz matrice zbunjenosti prema jednadžbi:

$$A_C = \frac{\sum A(i,i)}{N} \tag{3-11}$$

gdje je *N* ukupan broj podataka u skupu za testiranje.

U [15] se objašnjava da točnost klasifikacije može navoditi na krivi zaključak. Primjerice, u slučaju da postoji velika neravnoteža među klasama u ulaznim vektorima, klasifikator može predvidjeti vrijednost za većinsku klasu u svim predviđanjima te time postići visoku točnost klasifikacije, a da pritom posjeduje nisku točnost za manjinske klase. To se naziva paradoks točnosti. Za probleme poput ovoga potrebne su dodatne mjere za vrednovanje klasifikatora, kao što su preciznost, opoziv ili F1 mjera.

3.3 Određivanje veličine radijalne mreže

Pod pojmom veličina radijalne mreže misli se na broj neurona u skrivenom sloju. Broj ulaznih i izlaznih neurona za isti podatkovni skup se ne mijenja, ali količina neurona u skrivenom sloju može se mijenjati u svrhu postizanja veće učinkovitosti. U pravilu, optimalni broj neurona nije unaprijed poznat, stoga je poželjno provesti slijedno povećavanje broja neurona u skrivenom sloju. Potrebno je koristiti neki kriterij za određivanje odstupanja od stvarnih vrijednosti, poput srednje kvadratne pogreške (MSE).

Postoje dva efekta koja se pojavljuju u vezi skrivenih neurona [17]:

- pretjerana prilagođenost podacima za treniranje (engl. *overfitting*) - nastaje kada postoji prevelik broj redundantnih skrivenih neurona u mreži zbog kojih mreža "pamti" ulazne vektore i nije u stanju generalizirati kod skupa za testiranje
- podprilagođenost podacima za treniranje (engl. *underfitting*) - nastaje kad je broj skrivenih neurona manji od kompleksnosti problema te neuronska mreža nije u stanju odgovarajuće klasificirati podatke

Postoje razni pristupi kod određivanja količine skrivenih neurona, a po [17] neki od pristupa su metoda pokušaja i pogreške, praćenje generalnih smjernica, metoda s dvije faze i druge. Metodu

pokušaja i pogreške karakterizira ponavljanje i različiti pokušaji koji se izvršavaju sve do uspjeha ili dok korisnik ne prestane s ispitivanjem. Ovu metodu opisuju dva pristupa:

- pristup prema naprijed (engl. *forward approach*) - pristup započinje odabiranjem malog broja početnih neurona, često s dva neurona. Nakon toga se trenira i testira mreža, a zatim se poveća broj skrivenih neurona. Proces se ponavlja sve dok rezultati ne budu zadovoljavajući.
- pristup unazad (engl. *backward approach*) - u ovom se pristupu započinje s velikim brojem skrivenih neurona. Mreža se trenira i testira te se postupno smanjuje broj neurona sve dok se rezultati ne poboljšaju.

Neke od generalnih smjernica [17] za određivanje prikladnog broja skrivenih neurona su:

- broj skrivenih neurona treba biti u intervalu između broja ulaznih neurona i broja izlaznih neurona
- broj skrivenih neurona treba biti 2/3 veličine ulaznog broja neurona zbrojeno s brojem neurona izlaznog sloja

Metoda korištena u radu uključuje slijedno povećavanje broja neurona, a za svaki broj skrivenih neurona vršeno je popratno treniranje mreže i iz rezultata je izračunata srednja kvadratna pogreška između točnog rješenja i izračunatih vrijednosti. Određivanje veličine radijalne mreže započinje s dva neurona (što predstavlja najmanji mogući broj klasa) te se inkrementalno za jedan neuron povećava radijalna mreža sve do 20 neurona. Algoritam za svaki broj skrivenih neurona vrši treniranje mreže, nakon čega se vrši podjela skupa podataka na deset podjednakih skupova ili rezova (engl. *folds*) te se deset puta vrši izračunavanje izlaza mreže. Svaki se put kod izračunavanja izlaza izostavi određena desetina skupa, a na dobivenim izlazima se izračuna srednja kvadratna pogreška s obzirom na točna rješenja. Nakon deset puta, algoritam izračunava srednju vrijednost MSE te dobiveni broj predstavlja srednji MSE za navedeni broj skrivenih neurona. Nakon izračuna srednje vrijednosti MSE za svaku veličinu mreže, moguće je grafički prikazati ovisnost srednje kvadratne pogreške o veličini radijalne mreže.

4. OSTVARENO PROGRAMSKO RJEŠENJE

Ostvareno programsko rješenje obuhvaća radijalnu neuronsku mrežu za klasifikaciju koja je ostvarena u programskom jeziku Python (verzija 3.5.2) te grafički radni okvir (engl. *Graphical User Interface*, GUI) koje je ostvareno pomoću Electron tehnologije. Python je interpretirani jezik visokog nivoa za koji postoji bogat skup biblioteka kao podrška širokom spektru programskih problema. Electron je radni okvir otvorenog koda (engl. *open-source framework*) koje omogućava izradu GUI aplikacija koristeći komponente i jezike prvobitno osmišljene za razvoj mrežnih stranica. Tehnologije koje tvore Electron su Node.js i Chromium, a aplikacije se izrađuju pomoću HTML, CSS i JavaScript tehnologija.

4.1 Način rada programskog rješenja

Programsko rješenje se koristi tako što se pokrene program koji otvara grafički radni okvir (GUI) putem kojeg korisnik otvara proizvoljnu datoteku, odabire parametre te sve odabrane opcije prenosi radijalnoj neuronskoj mreži koja tada vrši klasifikaciju i vraća rezultate grafičkom radnom okviru na uvid korisniku. Programsko rješenje je podijeljeno u dva logička dijela: neuronska mreža kao pozadinski dio i grafički radni okvir.

4.1.1 Pozadinski dio - radijalna neuronska mreža

Radijalna neuronska mreža je jedinstvena Python datoteka koja kao ulaze prima datoteku sa skupom podataka (engl. *dataset*) i odabrane parametre od strane korisnika kao *bool* vrijednosti. Korišteno je nekoliko biblioteka u programu, primjerice `numpy` za matematičke operacije nad matricama, `csv` za čitanje datoteka, `sklearn` za određivanje kvalitete klasifikatora i `matplotlib` za iscrtavanje grafova. Program vrši predobradu skupa podataka što uključuje pretvaranje cijelog skupa u `numpy` matricu za buduće algebarske operacije nad skupom. Nakon toga slijedi normalizacija svih stupaca osim posljednjeg stupca koji označava klasu svakog retka u podatkovnom skupu tj. linearno skaliranje vrijednosti unutar intervala [0,1] kao što je opisano formulom (4-2). Engelbrecht [5] predlaže normalizaciju tj. provođenje skaliranja nad atributima podataka zbog poboljšanja performansi. Korišteno je linearno "Min-max" skaliranje kako bi se podaci skalirali između nove minimalne i maksimalne vrijednosti,

$$t_s = \frac{t_u - t_{u,min}}{t_{u,max} - t_{u,min}} \left(t_{s,max} - t_{s,min}\right) + t_{s,min} \quad (4\text{-}1)$$

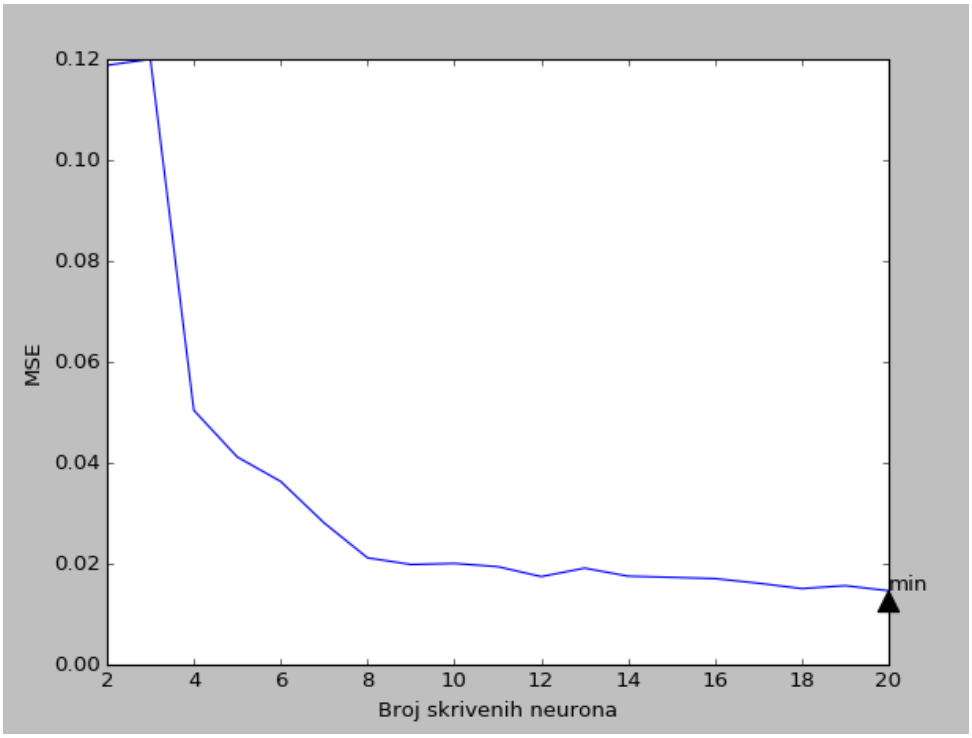
gdje su $t_{u,max}$ i $t_{u,min}$ maksimalna i minimalna vrijednost neskaliranog stupca, a $t_{s,max}$ i $t_{s,min}$ predstavljaju novu maksimalnu i minimalnu vrijednost skaliranog stupca. Stoga ova formula linearno preslikava interval $[t_{u,min}, t_{u,max}]$ u interval $[t_{s,min}, t_{s,max}]$. U slučaju kad se koristi linearno skaliranje između nula i jedan, jednadžba se pojednostavljuje u:

$$t_s = \frac{t_u - t_{u,min}}{t_{u,max} - t_{u,min}} \quad (4\text{-}2)$$

Posljednji stupac koji označava pripadnost pojedinoj kategoriji se ne normalizira, već se samo provjerava broj različitih kategorija, a taj se broj predaje kao parametar instanci klase radijalne neuronske mreže. Neuronsku mrežu u programskom kodu predstavlja klasa za čiju inicijalizaciju je potrebno predati podatke za treniranje, podatke za testiranje, broj kategorija skupa podataka te broj neurona u skrivenom sloju. Metode koje pruža klasa neuronske mreže su:

- ***kmeans*** metoda - predstavlja algoritam za grupiranje podataka *k*-means koji služi za određivanje prikladnih centara radijalnih funkcija
- ***pickDatapoints*** metoda - koristi se za nasumičan odabir redova iz skupa za treniranje koji bi predstavljali centre radijalnih funkcija
- ***sigma*** metoda - koristi se za pronalaženje jednakih širina za radijalne funkcije kako je opisano jednadžbom (3-2)
- ***std_dev*** metoda - koristi se za određivanje pojedinačnih širina za svaku radijalnu funkciju po pravilu p-najbližih susjeda opisano jednadžbom (3-3)
- ***train*** metoda - koristi se za treniranje mreže tj. određivanja pozicije za svaki centar radijalne funkcije, njezinu širinu kao i težine između skrivenog i izlaznog sloja opisano jednadžbom (3-5)
- ***test*** metoda - služi za testiranje neuronske mreže tj. uspoređivanje predviđenih izlaza (na dosad neviđenim podacima) sa stvarnim vrijednostima
- ***hiddenSize*** metoda - koristi se za određivanje veličine neuronske mreže računanjem srednje kvadratne pogreške za svaki broj centara između 2 i 20
- ***conf_matrix*** metoda - služi za izračunavanje matrice zbunjenosti
- ***MeanSquaredError*** metoda - služi za izračunavanje srednje kvadratne pogreške
- ***ClassificationReport*** metoda - koristi se za izračun F1 mjere, točnosti i preciznosti

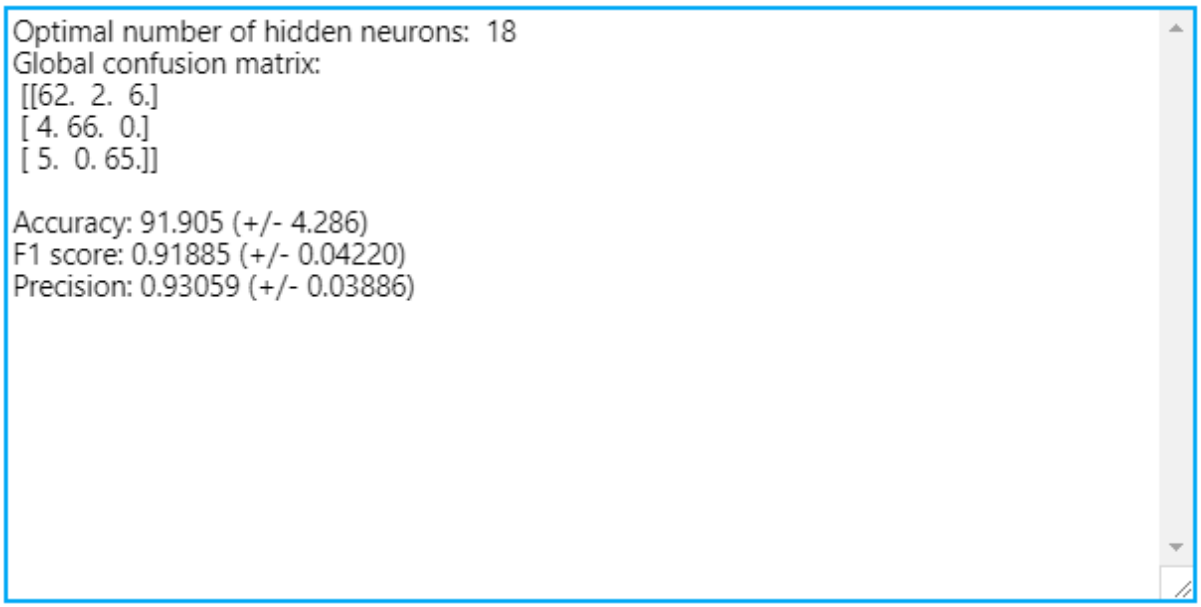
Metoda ***hiddenSize*** za svaki broj skrivenih neurona od 2 do 20 neurona vrši treniranje mreže, zatim dijeli skup podataka na deset jednakih dijelova te se stvara petlja koja svaki dio izostavi točno jednom, a $\frac{9}{10}$ podataka testira i izračunava srednju kvadratnu pogrešku. Kad se petlja ponovi deset puta, dobije se deset vrijednosti za srednju kvadratnu pogrešku. Od njih se tada izračuna srednja vrijednost koja se koristi kao reprezentativna vrijednost srednje kvadratne pogreške za *j*-ti broj neurona skrivenog sloja. Na kraju izračuna se dobiva MSE za svaki broj skrivenih neurona te se iscrtava graf kao na slici 4.1, a broj skrivenog neurona koji posjeduje najmanji izračunati MSE se uzima kao prikladna veličina neuronske mreže.



Sl. 4.1: Primjer prikazanog grafa ovisnosti MSE o broju skrivenih neurona

Nakon što je određena veličina neuronske mreže, slijedi testiranje mreže radi izračuna ukupne matrice zbunjenosti kao i ukupne točnosti, preciznosti i F1 mjere. Skup podataka se opet dijeli na deset jednakih dijelova te se vrši treniranje i testiranje neuronske mreže gdje $\frac{9}{10}$ podataka predstavlja skup za treniranje, a $\frac{1}{10}$ podataka predstavlja skup za testiranje. Nakon svake iteracije petlje (ukupno deset) dobiva se matrica zbunjenosti za danu iteraciju, preciznost, točnost i F1 mjera. Nakon odrađenih deset iteracija, matrice zbunjenosti se zbrajaju kako bi se dobila globalna matrica zbunjenosti za sve iteracije petlje. Od dobivenih deset vrijednosti, točnost, preciznost i F1 mjera su predstavljeni izračunavanjem srednjih vrijednosti i standardne devijacije

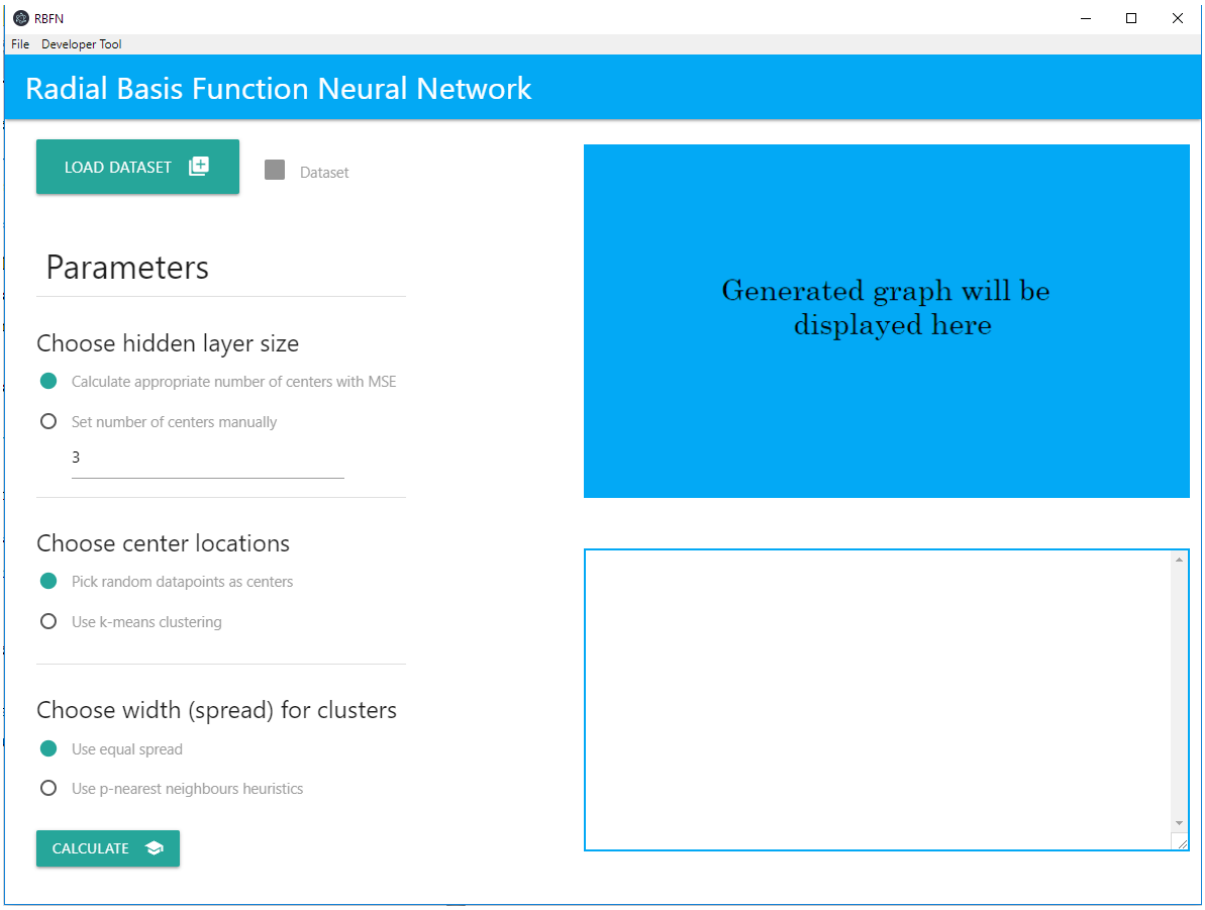
tih deset vrijednosti. Na slici 4.2 predstavljen je konačni rezultat koji korisnik dobiva na uvid u tekstualnom polju prilikom završetka programa.



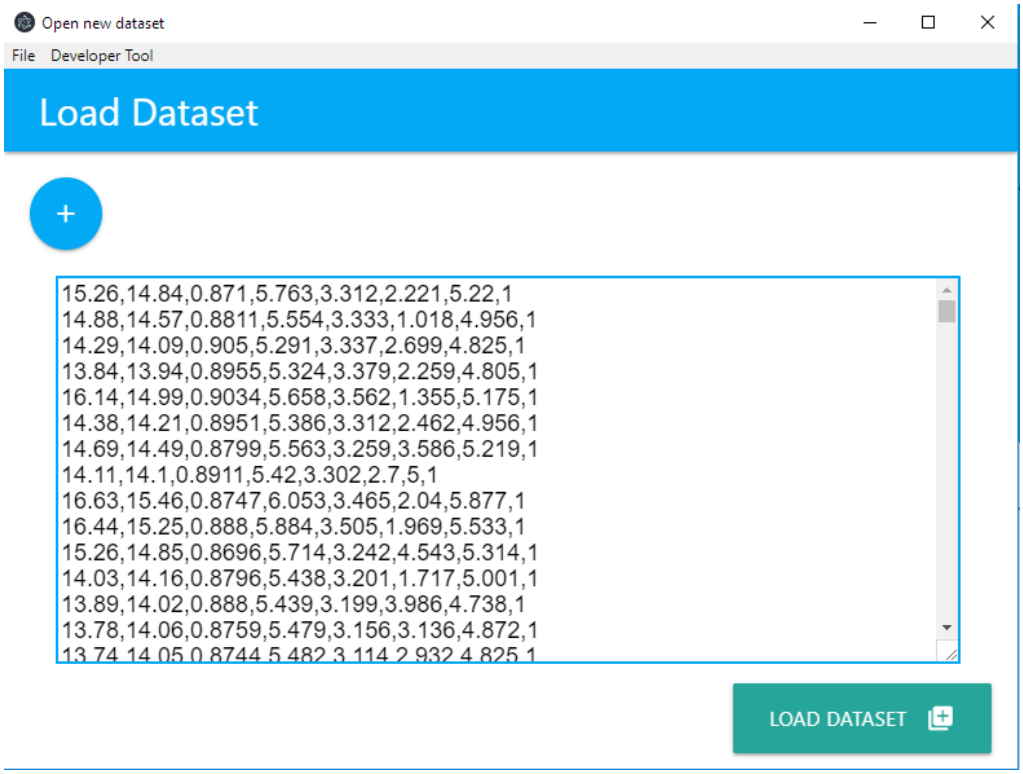
Sl. 4.2: Primjer rezultata neuronske mreže

4.1.2 Grafički radni okvir

Programsko rješenje sadrži grafički radni okvir koji olakšava korisnicima interakciju s neuronskom mrežom. Nakon pokretanja radnog okvira, korisnik ima mogućnosti poput učitavanja datoteka zapisanih u csv formatu, odabiranja parametara i dobivanja izračunatih rezultata od pozadinskog dijela. Prilikom pokretanja aplikacije otvara se prozor radnog okvira koji izgleda kao na slici 4.3. Korisnik tada otvara proizvoljnu datoteku u csv formatu kako bi mogao trenirati i testirati neuronsku mrežu s odabranim parametrima. Parametri se nalaze s lijeve strane radnog okvira, a korisnik odabire jednu opciju za svaku od triju kategorija (veličina mreže, centri i širine). Na desnoj strani radnog okvira nalazi se mjesto za prikazivanje generiranog grafa kod određivanja veličine neuronske mreže, ispod kojeg je tekstualno polje u koje se ispisuju rezultati testiranja radijalne neuronske mreže. Pritiskom na tipku "LOAD DATASET" otvara se drugi prozor kao na slici 4.4, putem kojeg korisnik odabire i učitava datoteku u programsko rješenje.



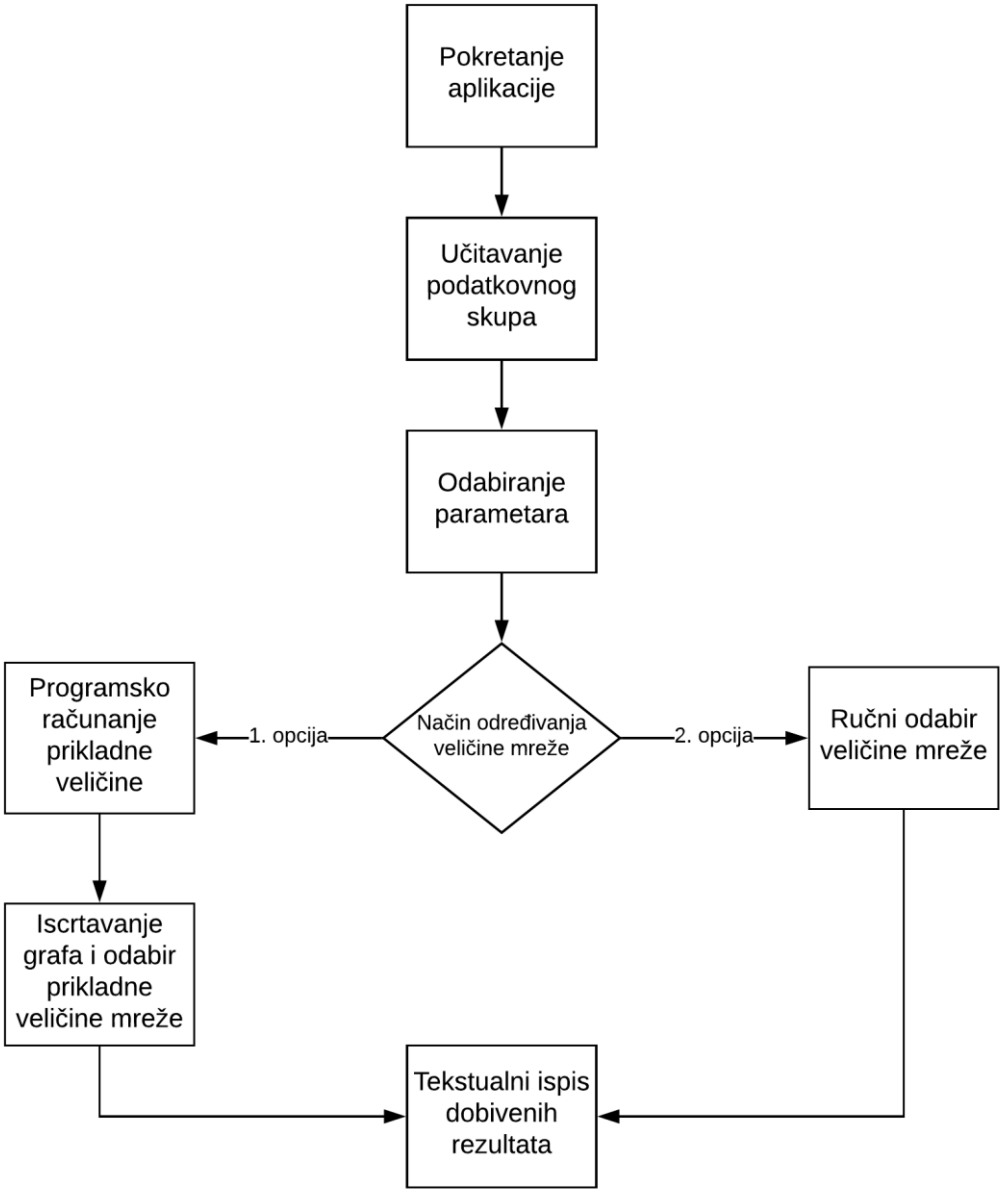
Sl. 4.3: Početni prozor programskog rješenja



Sl. 4.4: Prozor za učitavanje datoteke u programskom rješenju

4.2 Prikaz i način uporabe programskog rješenja

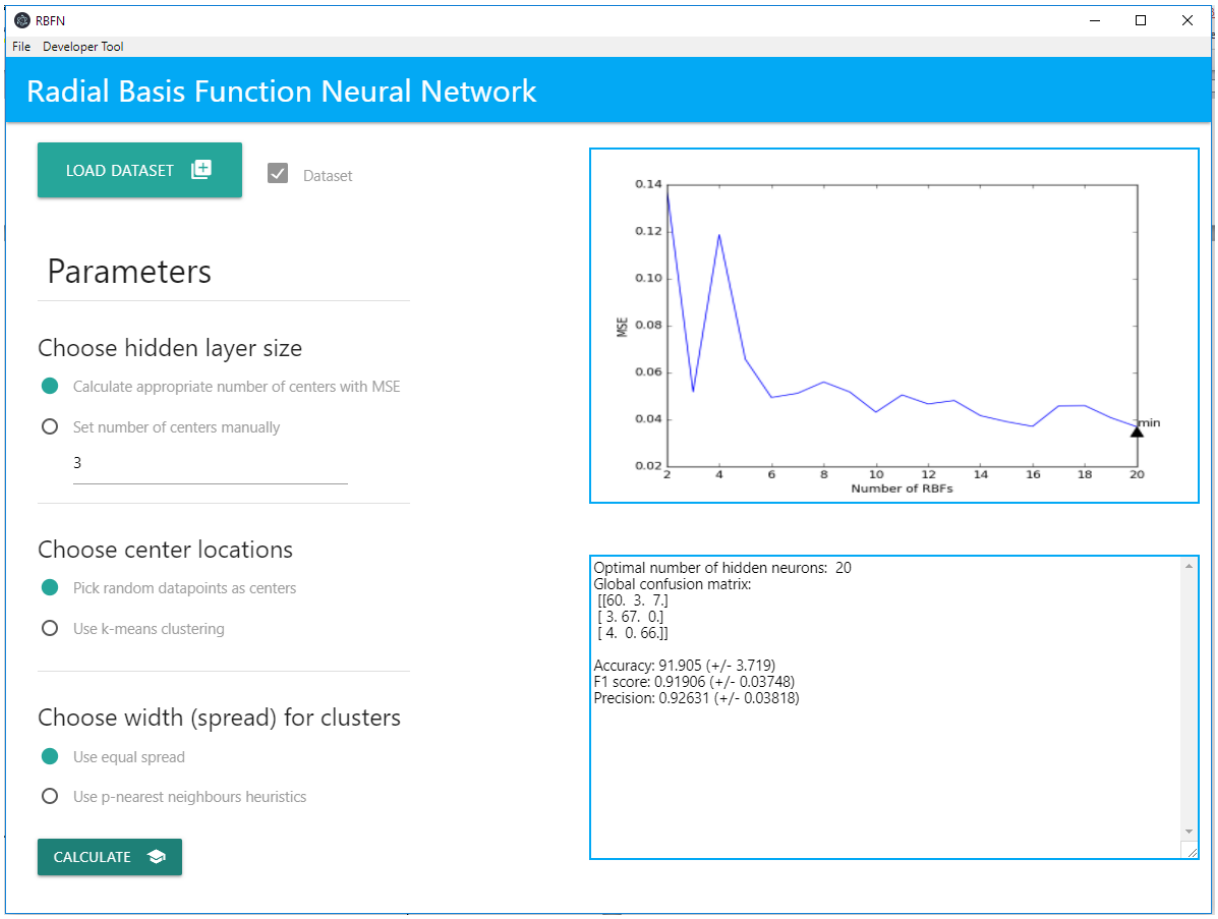
Način korištenja programskog rješenja se svodi na učitavanje datoteke s podacima, odabiranje parametara neuronske mreže te razmatranje dobivenih rezultata, kao što se vidi iz dijagrama toka na slici 4.5. Nakon pokretanja programskog rješenja s radne površine i otvaranja grafičkog radnog okvira, potrebno je otvoriti dodatni prozor (slika 4.4) za učitavanje skupa podataka. U novom prozoru je potrebno pritisnuti na tipku "+", a zatim u pretraživaču odabrati proizvoljnu datoteku csv formata. U tekstualnom polju sa slike 4.4 se ispiše sadržaj odabrane datoteke, a korisnik te podatke predaje programu pritiskom na tipku "LOAD DATASET". Nakon učitavanja skupa podataka korisnik odabire parametre neuronske mreže. Tipka "CALCULATE" predaje pozadinskom dijelu (radijalnoj neuronskoj mreži) učitane podatke iz datoteke s odabranim parametrima mreže. U slučaju da korisnik pritisne navedenu tipku bez učitavanja skupa podataka, u tekstualnom se polju ispiše poruka da podaci nisu predani aplikaciji, što korisnik može zaključiti iz neoznačene kućice pored "LOAD DATASET" tipke na slici 4.3.



Sl. 4.5: Dijagram toka korištenja programskog rješenja

Daljnji tijek izvođenja programa ovisi o odabiru načina određivanja veličine mreže. Korisnik ima mogućnost ručno odabrati broj skrivenih neurona (2. opcija na slici 4.5) tj. radijalnih funkcija pomoću kojih želi provesti analizu. Na taj način korisnik samo dobiva uvid u rezultate u tekstualnom polju, a izračun prikladnog broja radijalnih funkcija, kao i iscrtavanje grafa nisu raspoloživi. U slučaju da korisnik odabere prvu opciju s dijagrama toka na slici 4.5, na radnom okviru prikaže se dobiveni graf preko kojeg neuronska mreža odabire prikladnu veličinu skrivenog sloja, a zatim se dobije uvid u dobivene rezultate.

Na slici 4.6 se može vidjeti iscrtani graf s dobivenim rezultatima neuronske mreže (desna strana radnog okvira). Rezultati se sastoje od podatka o prikladnom broju skrivenih neurona (što je označeno i na grafu), ukupnoj matrici zbunjenosti te klasifikacijskim mjerama poput točnosti, preciznosti i F1 mjere. Učitani skup podataka ostaje i nakon izračuna u programu, stoga korisnik može odabrati drukčije parametre te izvršiti analizu iznova. Ako korisnik želi izvršiti analizu na drugom skupu podataka, potrebno je iznova učitati novu datoteku sa željenim podacima.



Sl. 4.6: Grafički radni okvir s dobivenim rezultatima

Korisnik ima mogućnost provesti treniranje i testiranje mreže proizvoljan broj puta, pritom mijenjajući parametre mreže po izboru. Svaki put kada korisnik pritisne na tipku "CALCULATE", uklanja se graf te se postavlja početna slika kao na slici 4.3. Također se rezultati u tekstualnom polju uklone te se ispiše poruka o ponovnoj analizi.

5. EKSPERIMENTALNA ANALIZA

Eksperimentalnom analizom prvenstveno je utvrđeno postoji li razlika u efikasnosti korištenja algoritma grupiranja u problemu klasifikacije koristeći radijalnu neuronsku mrežu. Uspoređena je kvaliteta klasifikacije u slučaju da je korišten algoritam za grupiranje podataka *k*-means ili su nasumično odabrani podaci kao centri radijalnih funkcija. Također je pokazana efikasnost neuronske mreže u ovisnosti korištenja dvaju različitih načina određivanja širina radijalnih funkcija. Prvi se način određivanja širina zalaže za jednaku širinu za svaki centar radijalne funkcije, dok se drugi način zalaže za određivanje širine svakog centra radijalne funkcije posebno. U sklopu eksperimentalne analize provedeno je testiranje neuronske mreže za sve četiri kombinacije određivanja položaja centara i širina radijalne funkcije. Osim te prvenstvene svrhe, eksperimentalnom analizom prikazana je i kvaliteta određivanja prikladne veličine neuronske mreže. To je pokazano predloženim grafom koji je popraćen analizom mreže za svaki broj skrivenih neurona.

Eksperimentalna analiza provedena je na pet skupova podataka. Skupovi podataka su "Seeds", "Iris", "Breast Cancer Wisconsin (Diagnostic)", "Blood Transfusion Service Center" te "Glass Identification" preuzeti s [18]. U nastavku je dan sažeti opis navedenih skupova podataka:

- "Seeds" (u nastavku *Seeds* skup podataka) se sastoji od mjerenja geometrijskog svojstva jezgre tri različita tipa pšenice: Kama, Rosa i Canadian. Skup podataka sadrži 70 podataka svake klase pšenice, a svaki podatak opisuje sedam atributa.
- "Iris" (u nastavku *Iris* skup podataka) jedan je od najpoznatijih skupova podataka korištenih u strojnom učenju. Skup podataka sadrži po 50 podataka za svaku od tri klase, gdje svaka klasa predstavlja vrstu cvijeta perunike (lat. *Iris*). Klase nose imena Setosa, Versicolor i Virginica, a karakteriziraju ih četiri atributa.
- "Breast Cancer Wisconsin (Diagnostic)" (u nastavku *Cancer* skup podataka) se sastoji od 569 podataka s deset atributa. Međutim, prvi atribut predstavlja identifikacijski kod (ID) koji nema matematičku ovisnost s klasama, stoga ga je potrebno ukloniti prije eksperimentalne analize. Klasifikacija u ovom slučaju je dvoklasna, a opisuje maligni i benigni tumor.
- "Blood Transfusion Service Center" (u nastavku *Transfusion* skup podataka) je skup čiji su podaci prikupljeni od usluga transfuzije krvi u Tajvanu. Postoji 748 podataka, a četiri atributa koja ih opisuju su: broj mjeseci od zadnje donacije, ukupni broj donacija, ukupna

količina donirane krvi te vrijeme od prve donacije. Dvije klase predstavljaju informaciju je li osoba donirala krv u ožujku 2007. godine ili nije donirala.

- "Glass Identification" (u nastavku *Glass* skup podataka) koji se sastoji od 214 podataka opisanih s deset atributa. Kao i u *Cancer* skupu podataka, važno je ukloniti prvi stupac koji služi kao identifikacijski kod. Prijašnji skupovi podataka posjeduju samo dvije ili tri klase u koje se klasificiraju podaci. Stoga je *Glass* skup podataka odabran kako bi se izvršilo testiranje na skupu koji posjeduje šest mogućih klasa.

5.1 Postavke eksperimenta

Eksperiment je proveden na pet skupova podataka opisanih na početku poglavlja. Svaki od skupova podataka je podvrgnut testiranju koristeći radijalnu neuronsku mrežu s četiri moguće kombinacije parametara (u daljnjem tekstu se koristi naziv četiri "Varijante"). Skup podataka je testiran za svaku od četiri Varijante uz inkrementalno povećavanje neuronske mreže, a to znači da se broj neurona u skrivenom sloju povećava od 2 neurona na početku eksperimenta pa sve do 20 neurona na kraju eksperimenta. Za svaki od pet skupova podataka su dane dvije tablice, prva se odnosi na rezultate testiranja F1 mjere, a druga na rezultate testiranja preciznosti.

Nakon svake tablice je također iscrtan stupčasti graf koji predstavlja srednju vrijednost F1 mjere ili preciznosti, a uzimajući u obzir sve korištene veličine mreže u analizi. Svaki od četiri stupca u grafu predstavlja jednu od varijanti izgradnje neuronske mreže. Svaki stupac ujedno sadrži vertikalnu dužinu koja predstavlja srednju vrijednost standardne devijacije za danu varijantu testiranja. Nakon tablica i stupčastih grafova također su generirani grafovi određivanja prikladne veličine mreže pomoću MSE, iz kojih se u kombinaciji s prethodnim tablicama može uvidjeti kvaliteta odluke o veličini neuronske mreže.

Kao što je navedeno u prethodnom tekstu, moguće su četiri Varijante izgradnje mreže, a svaka Varijanta predstavlja stupac u tablici i stupčastom grafu:

- Varijanta 1 − korištenje algoritma *k*-means za određivanje centara i pravila p-najbližih susjeda za određivanje širina
- Varijanta 2 − korištenje algoritma *k*-means za određivanje centara i korištenje jednakih širina za radijalne funkcije
- Varijanta 3 − korištenje nasumičnih ulaznih vektora za određivanje centara i pravila p-najbližih susjeda za određivanje širina

- Varijanta 4 − korištenje nasumičnih ulaznih vektora za određivanje centara i korištenje jednakih širina za radijalne funkcije

Metoda testiranja nad podatkovnim skupom je sljedeća: podatkovni je skup algoritmom K-rez (engl. *K-fold*) podijeljen na deset jednakih rezova (engl. 10-*folds*), od kojih skup za treniranje čini devet rezova, a skup za testiranje jedan rez. Za svih deset ponavljanja izračunata je F1 mjera i preciznost. Nakon izvršenih deset treniranja i testiranja, izračunata je srednja vrijednost F1 mjere i preciznosti s njihovim standardnim devijacijama. Cijeli postupak je ponovljen za sve zadane veličine mreže od 2 do 20 skrivena neurona. U tablicama su vrijednosti prikazane na način: srednja vrijednost +/- standardna devijacija.

5.2 Rezultati

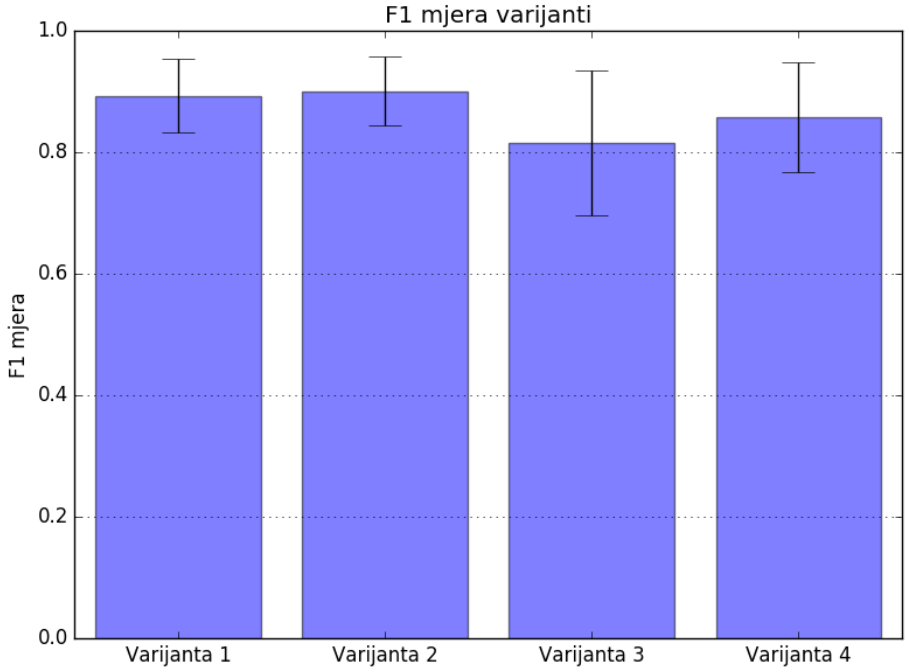
U radu su obrađene četiri mjere klasifikacije: točnost, preciznost, opoziv i F1 mjera. Za rezultate analize korištene su dvije mjere, a to su F1 mjera i preciznost. Točnost klasifikacije nije korištena zbog moguće pojave paradoksa točnosti, a mjerenje opoziva je izbjegnuto zbog toga što je opoziv moguće izračunati iz danih vrijednosti preciznosti i F1 mjere.

Tablicom 5.1 i slikom 5.1 numerički i grafički su opisani rezultati F1 mjere testiranja izvršenog na *Seeds* skupu podataka. Sve četiri Varijante prelaze vrijednost F1 mjere od 0.9 na određenoj veličini mreže, međutim, prve dvije Varijante koje koriste algoritam za grupiranje podataka *k*-means pokazuju bolje rezultate. Za svaki stupac iz tablice 5.1 izračunata je srednja vrijednost podataka te navedena srednja vrijednost predstavlja stupac Varijante u stupčastom grafu kao na slici 5.1. Isti postupak je proveden i nad podacima standardne devijacije. Stoga, u svakom su stupcu u grafu obuhvaćene sve veličine mreže za svaku varijantu.

Tab. 5.1: F1 mjera za *Seeds* skup podataka

F1 mjera	Varijanta 1	Varijanta 2	Varijanta 3	Varijanta 4
Veličina mreže	Srednja vrijednost +/- standardna devijacija	Srednja vrijednost +/- standardna devijacija	Srednja vrijednost +/- standardna devijacija	Srednja vrijednost +/- standardna devijacija
2	0.5988 +/- 0.1298	0.5750 +/- 0.0924	0.3761 +/- 0.2105	0.3292 +/- 0.1558
3	0.9126 +/- 0.0465	0.9002 +/- 0.0692	0.4642 +/- 0.1687	0.6064 +/- 0.2324
4	0.9242 +/- 0.0315	0.9096 +/- 0.0691	0.8212 +/- 0.1839	0.7590 +/- 0.172
5	0.9165 +/- 0.0535	0.9246 +/- 0.0478	0.6758 +/- 0.2648	0.8291 +/- 0.1588
6	0.9139 +/- 0.0569	0.9257 +/- 0.0555	0.8427 +/- 0.137	0.8765 +/- 0.1147
7	0.9024 +/- 0.0627	0.9264 +/- 0.04	0.8649 +/- 0.1465	0.9049 +/- 0.0516

8	0.9121 +/- 0.0483	0.9156 +/- 0.0787	0.8734 +/- 0.0822	0.9287 +/- 0.0721
9	0.9093 +/- 0.039	0.9193 +/- 0.0314	0.8875 +/- 0.0784	0.8407 +/- 0.1578
10	0.8989 +/- 0.0663	0.9096 +/- 0.0625	0.8455 +/- 0.1011	0.9242 +/- 0.0616
11	0.9108 +/- 0.0805	0.9189 +/- 0.0522	0.8617 +/- 0.0941	0.9330 +/- 0.0608
12	0.8942 +/- 0.0693	0.9088 +/- 0.08	0.8096 +/- 0.139	0.9095 +/- 0.0455
13	0.9173 +/- 0.0486	0.9146 +/- 0.0633	0.9023 +/- 0.0598	0.9048 +/- 0.0806
14	0.8884 +/- 0.0574	0.9281 +/- 0.049	0.8939 +/- 0.0513	0.9379 +/- 0.0308
15	0.9241 +/- 0.0486	0.9292 +/- 0.0482	0.8984 +/- 0.0921	0.9216 +/- 0.0662
16	0.9149 +/- 0.0544	0.9187 +/- 0.0317	0.9157 +/- 0.0639	0.9472 +/- 0.0402
17	0.9071 +/- 0.0647	0.9097 +/- 0.027	0.8406 +/- 0.1869	0.9282 +/- 0.0575
18	0.8961 +/- 0.0788	0.9289 +/- 0.0634	0.9042 +/- 0.0528	0.9391 +/- 0.0425
19	0.9134 +/- 0.0709	0.9275 +/- 0.0563	0.8995 +/- 0.079	0.9382 +/- 0.0477
20	0.9132 +/- 0.0431	0.9191 +/- 0.0569	0.9095 +/- 0.0654	0.9306 +/- 0.0576

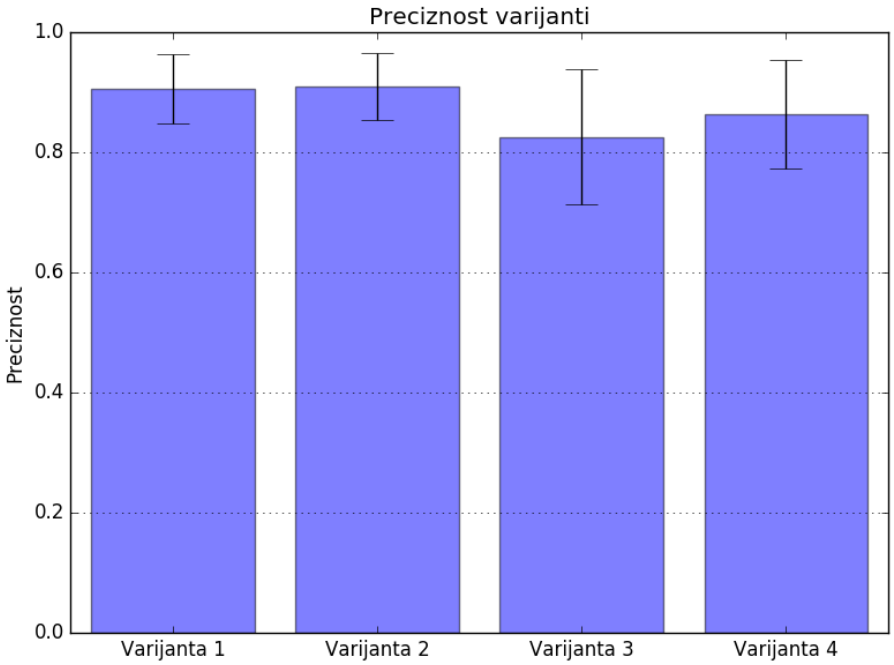


Sl. 5.1: Graf F1 mjere za *Seeds* skup podataka

Rezultati mjerenja preciznosti u tablici 5.2 također pokazuju slične rezultate. Uočava se da su kod prve dvije Varijante dovoljna samo tri skrivena neurona za postizanje približno najveće učinkovitosti, što znači da razlika u preciznosti između korištenja 3 i 20 skrivenih neurona nije velika. Za razliku od prve dvije Varijante, treća i četvrta Varijanta zahtijevaju preko deset radijalnih funkcija za istu razinu preciznosti. Također je uočljiva znatno veća standardna devijacija kod treće i četvrte Varijante na slici 5.2.

Tab. 5.2: Preciznost za *Seeds* skup podataka

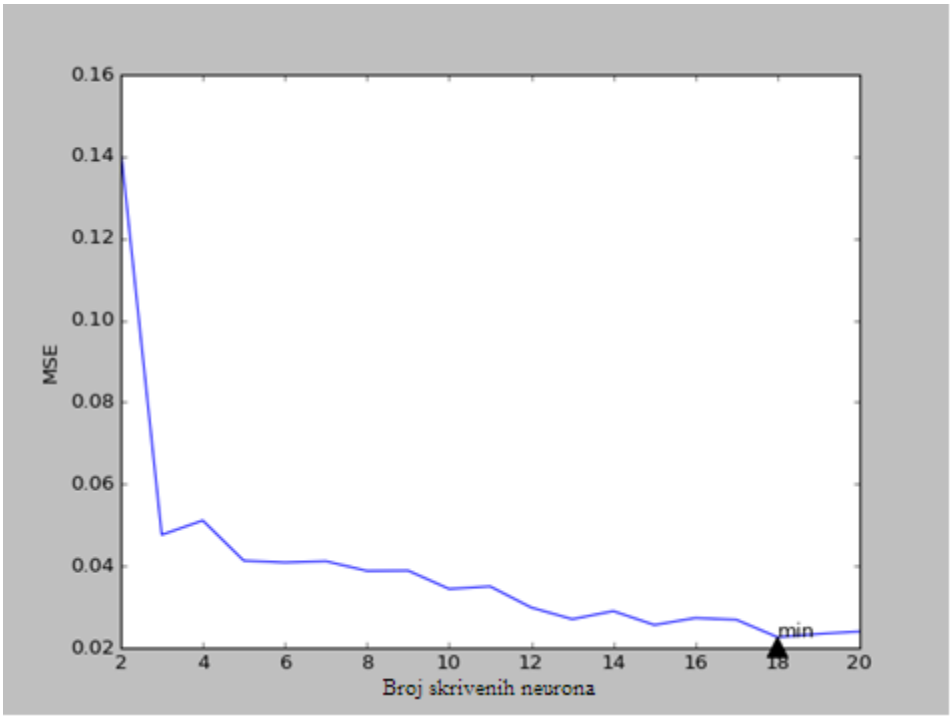
Preciznost	Varijanta 1	Varijanta 2	Varijanta 3	Varijanta 4
Veličina mreže	Srednja vrijednost +/- standardna devijacija	Srednja vrijednost +/- standardna devijacija	Srednja vrijednost +/- standardna devijacija	Srednja vrijednost +/- standardna devijacija
2	0.6028 +/- 0.2061	0.5625 +/- 0.1513	0.3211 +/- 0.2093	0.2894 +/- 0.1422
3	0.9271 +/- 0.0384	0.9121 +/- 0.0637	0.4339 +/- 0.1945	0.6164 +/- 0.2649
4	0.9296 +/- 0.0318	0.9199 +/- 0.0586	0.8184 +/- 0.2012	0.7478 +/- 0.1905
5	0.9302 +/- 0.0455	0.9329 +/- 0.0434	0.6662 +/- 0.2963	0.8445 +/- 0.1642
6	0.9279 +/- 0.0431	0.9389 +/- 0.0473	0.8801 +/- 0.0841	0.8816 +/- 0.1349
7	0.9152 +/- 0.0533	0.9340 +/- 0.0392	0.8694 +/- 0.1556	0.9126 +/- 0.0507
8	0.9264 +/- 0.0445	0.9294 +/- 0.0714	0.8840 +/- 0.0838	0.9461 +/- 0.0473
9	0.9200 +/- 0.0369	0.9285 +/- 0.0317	0.9078 +/- 0.0649	0.8466 +/- 0.1769
10	0.9122 +/- 0.0572	0.9218 +/- 0.0548	0.8927 +/- 0.045	0.9362 +/- 0.0575
11	0.9251 +/- 0.0648	0.9261 +/- 0.05	0.8949 +/- 0.0515	0.9377 +/- 0.0588
12	0.9077 +/- 0.0673	0.9182 +/- 0.0779	0.8322 +/- 0.1623	0.9216 +/- 0.0408
13	0.9272 +/- 0.0465	0.9253 +/- 0.0529	0.9234 +/- 0.0437	0.9173 +/- 0.0742
14	0.9009 +/- 0.0529	0.9323 +/- 0.049	0.9208 +/- 0.0382	0.9468 +/- 0.0267
15	0.9310 +/- 0.0439	0.9345 +/- 0.046	0.9207 +/- 0.0658	0.9342 +/- 0.0624
16	0.9274 +/- 0.0497	0.9328 +/- 0.0233	0.9299 +/- 0.0534	0.9534 +/- 0.0361
17	0.9213 +/- 0.0618	0.9222 +/- 0.0295	0.8388 +/- 0.2038	0.9395 +/- 0.0488
18	0.9126 +/- 0.0639	0.9407 +/- 0.0505	0.9192 +/- 0.0479	0.9482 +/- 0.0397
19	0.9271 +/- 0.0614	0.9341 +/- 0.0537	0.9124 +/- 0.0702	0.9422 +/- 0.0458
20	0.9315 +/- 0.0389	0.9333 +/- 0.0523	0.9211 +/- 0.0598	0.9425 +/- 0.0543



Sl. 5.2: Graf preciznosti za *Seeds* skup podataka

Iz danih grafova se uočava da Varijanta 1 i 2 pokazuju najbolje rezultate za zadani skup podataka. Na slici 5.3 je predstavljen graf određivanja prikladne veličine mreže za Varijantu 2

kako bi se mogao usporediti s tablicama. Ako se usporedi graf sa slike 5.3 i tablica 5.2, zamjećuje se odlična korelacija između predložene veličine mreže i dobivenih podataka. Za dva skrivena neurona postoji veliki MSE zato što dvije radijalne funkcije nisu u stanju dovoljno dobro opisati ulazni prostor za tri klase.



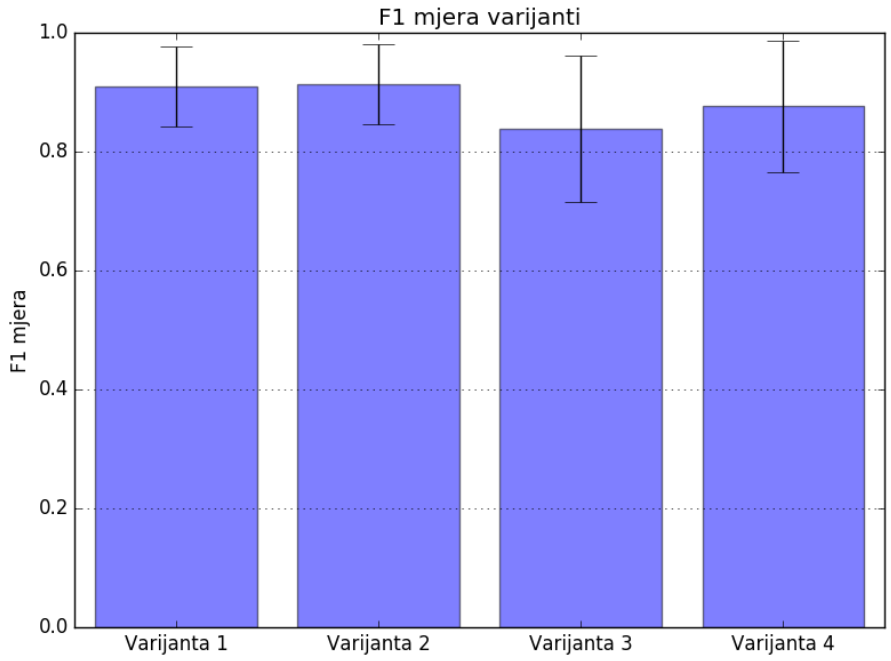
Sl. 5.3: Graf prikladnog broja neurona kod varijante 2 za *Seeds* skup podataka

Iz tablice 5.3 se uočava da *Iris* skup podataka nije dovoljno dobro opisan sa samo tri radijalne funkcije (kao što je *Seeds* skup podataka), već se rezultati postupno poboljšavaju povećavanjem broja radijalnih funkcija. Na slici 5.4 je predstavljen stupčasti graf F1 mjere za svaku varijantu kojom su obuhvaćene sve veličine mreže. Tablicom 5.4 su prikazani rezultati preciznosti za *Iris* skup podataka, dok su na slici 5.5 grafički prikazani dobiveni rezultati stupčastim grafom.

Tab. 5.3: F1 mjera za *Iris* skup podataka

F1 mjera	Varijanta 1	Varijanta 2	Varijanta 3	Varijanta 4
Veličina mreže	Srednja vrijednost +/- standardna devijacija	Srednja vrijednost +/- standardna devijacija	Srednja vrijednost +/- standardna devijacija	Srednja vrijednost +/- standardna devijacija
2	0.5064 +/- 0.1688	0.4969 +/- 0.1785	0.4846 +/- 0.178	0.4083 +/- 0.202
3	0.8759 +/- 0.0754	0.7848 +/- 0.1677	0.6415 +/- 0.1979	0.6730 +/- 0.3051
4	0.8420 +/- 0.0829	0.8820 +/- 0.0838	0.6644 +/- 0.2296	0.7726 +/- 0.1488
5	0.8850 +/- 0.1072	0.8882 +/- 0.1028	0.6672 +/- 0.2497	0.8031 +/- 0.1919

6	0.9070 +/- 0.0528	0.9048 +/- 0.0829	0.8869 +/- 0.1281	0.9112 +/- 0.1335
7	0.9333 +/- 0.0734	0.9397 +/- 0.0554	0.8856 +/- 0.1452	0.8778 +/- 0.1532
8	0.9199 +/- 0.0777	0.9603 +/- 0.0607	0.8762 +/- 0.1273	0.9339 +/- 0.0517
9	0.9416 +/- 0.0537	0.9668 +/- 0.0332	0.8330 +/- 0.1946	0.9060 +/- 0.1558
10	0.9321 +/- 0.077	0.9669 +/- 0.0331	0.8453 +/- 0.1114	0.9001 +/- 0.1693
11	0.9595 +/- 0.0445	0.9596 +/- 0.0444	0.9226 +/- 0.0924	0.8918 +/- 0.1829
12	0.9519 +/- 0.0633	0.9390 +/- 0.0707	0.8357 +/- 0.139	0.9537 +/- 0.0425
13	0.9535 +/- 0.0517	0.9539 +/- 0.0303	0.9062 +/- 0.079	0.9529 +/- 0.0444
14	0.9530 +/- 0.0428	0.9665 +/- 0.0335	0.8795 +/- 0.1033	0.9676 +/- 0.0441
15	0.9601 +/- 0.0533	0.9660 +/- 0.0341	0.9396 +/- 0.0637	0.9329 +/- 0.0665
16	0.9465 +/- 0.0717	0.9579 +/- 0.0671	0.9591 +/- 0.0334	0.9592 +/- 0.0334
17	0.9465 +/- 0.0489	0.9527 +/- 0.0524	0.9118 +/- 0.0683	0.9593 +/- 0.0333
18	0.9539 +/- 0.0304	0.9466 +/- 0.0499	0.9281 +/- 0.0684	0.9667 +/- 0.0333
19	0.9543 +/- 0.0661	0.9538 +/- 0.0783	0.9342 +/- 0.0782	0.9274 +/- 0.0709
20	0.9664 +/- 0.0455	0.9662 +/- 0.0339	0.9234 +/- 0.0597	0.9526 +/- 0.0432

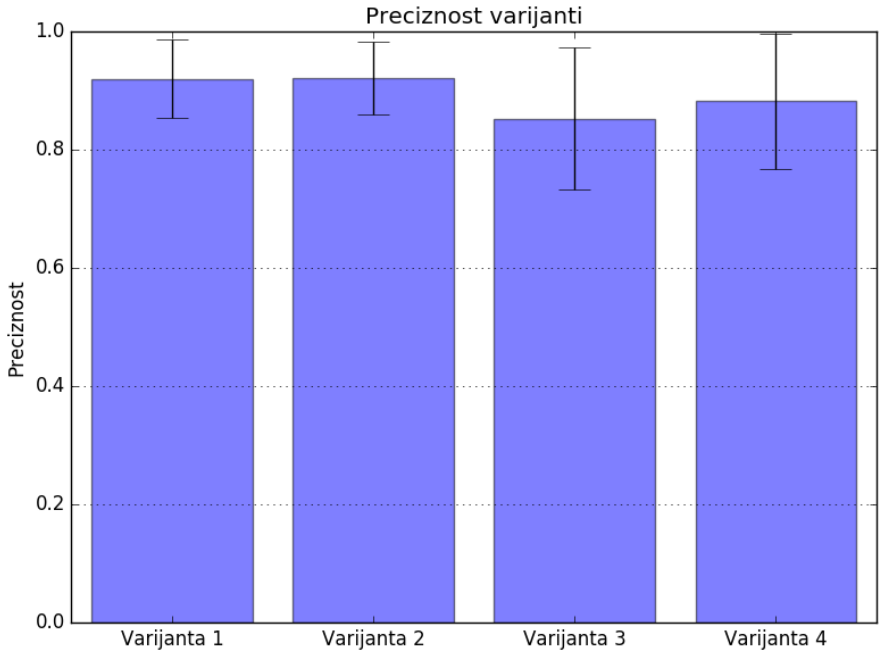


Sl. 5.4: Graf F1 mjere za *Iris* skup podataka

Tab. 5.4: Preciznost za *Iris* skup podataka

Preciznost	Varijanta 1	Varijanta 2	Varijanta 3	Varijanta 4
Veličina mreže	Srednja vrijednost +/- standardna devijacija	Srednja vrijednost +/- standardna devijacija	Srednja vrijednost +/- standardna devijacija	Srednja vrijednost +/- standardna devijacija
2	0.4965 +/- 0.2266	0.4602 +/- 0.2013	0.4855 +/- 0.1637	0.3554 +/- 0.2068
3	0.8928 +/- 0.0794	0.7916 +/- 0.1804	0.6385 +/- 0.2333	0.6585 +/- 0.3367
4	0.8770 +/- 0.0852	0.9123 +/- 0.0726	0.6891 +/- 0.2531	0.7922 +/- 0.1795
5	0.9025 +/- 0.0936	0.9091 +/- 0.0918	0.6836 +/- 0.2531	0.8252 +/- 0.217
6	0.9229 +/- 0.0495	0.9347 +/- 0.0502	0.8921 +/- 0.1492	0.9115 +/- 0.1464
7	0.9443 +/- 0.0659	0.9575 +/- 0.031	0.8873 +/- 0.1568	0.8843 +/- 0.175
8	0.9284 +/- 0.0724	0.9658 +/- 0.0565	0.8825 +/- 0.146	0.9390 +/- 0.0521

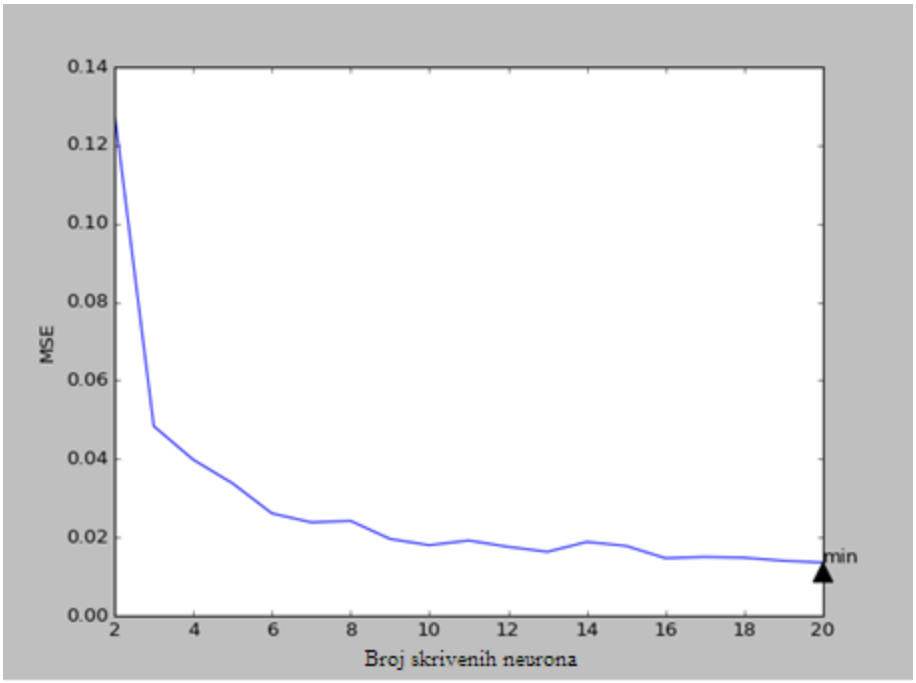
9	0.9547 +/- 0.0432	0.9736 +/- 0.0264	0.8723 +/- 0.187	0.9116 +/- 0.1557
10	0.9424 +/- 0.0719	0.9738 +/- 0.0263	0.8619 +/- 0.117	0.9048 +/- 0.1832
11	0.9650 +/- 0.0417	0.9639 +/- 0.0421	0.9407 +/- 0.0683	0.8942 +/- 0.1882
12	0.9561 +/- 0.0627	0.9438 +/- 0.0679	0.8532 +/- 0.151	0.9605 +/- 0.0399
13	0.9589 +/- 0.0481	0.9637 +/- 0.0241	0.9227 +/- 0.0706	0.9649 +/- 0.0307
14	0.9591 +/- 0.0403	0.9724 +/- 0.0276	0.8949 +/- 0.0897	0.9722 +/- 0.0416
15	0.9632 +/- 0.0522	0.9715 +/- 0.0285	0.9580 +/- 0.0382	0.9543 +/- 0.0417
16	0.9528 +/- 0.0653	0.9707 +/- 0.0421	0.9667 +/- 0.0273	0.9667 +/- 0.0273
17	0.9601 +/- 0.0349	0.9611 +/- 0.0452	0.9322 +/- 0.0503	0.9668 +/- 0.0272
18	0.9644 +/- 0.0242	0.9514 +/- 0.049	0.9421 +/- 0.0599	0.9726 +/- 0.0274
19	0.9632 +/- 0.0587	0.9572 +/- 0.0737	0.9568 +/- 0.0442	0.9466 +/- 0.0529
20	0.9740 +/- 0.0344	0.9729 +/- 0.0273	0.9435 +/- 0.0392	0.9707 +/- 0.0304



Sl. 5.5: Graf preciznosti za *Iris* skup podataka

Iz grafova na slikama 5.4 i 5.5 može se zaključiti da metoda određivanja širina radijalnih funkcija za prve dvije Varijante nema utjecaja na ishode rezultata, tj. ako se koristi algoritam za grupiranje podataka, tada korištena metoda za širine nije važna. Za razliku od prve dvije Varijante, kod Varijanti gdje nije korišten algoritam za grupiranje podataka, metoda za određivanje širina ima znatno veći utjecaj na rezultate. U ovom slučaju, srednja vrijednost preciznosti i F1 mjere je viša kod metode korištenja jednakih širina nego kod pravila p-najbližih susjeda. Također, standardna devijacija je čak dva puta veća kod treće i četvrte Varijante gdje nije korišten algoritam za grupiranje podataka. Na slici 5.6 je prikazan graf određivanja prikladne veličine mreže za odabranu varijantu 2. Iz grafa je vidljiva korelacija s prethodnim

tablicama preciznosti i F1 mjere za *Iris* skup podataka, iz koje je razvidno da se rezultati postupno poboljšavaju svakim dodanim neuronom u skrivenom sloju.

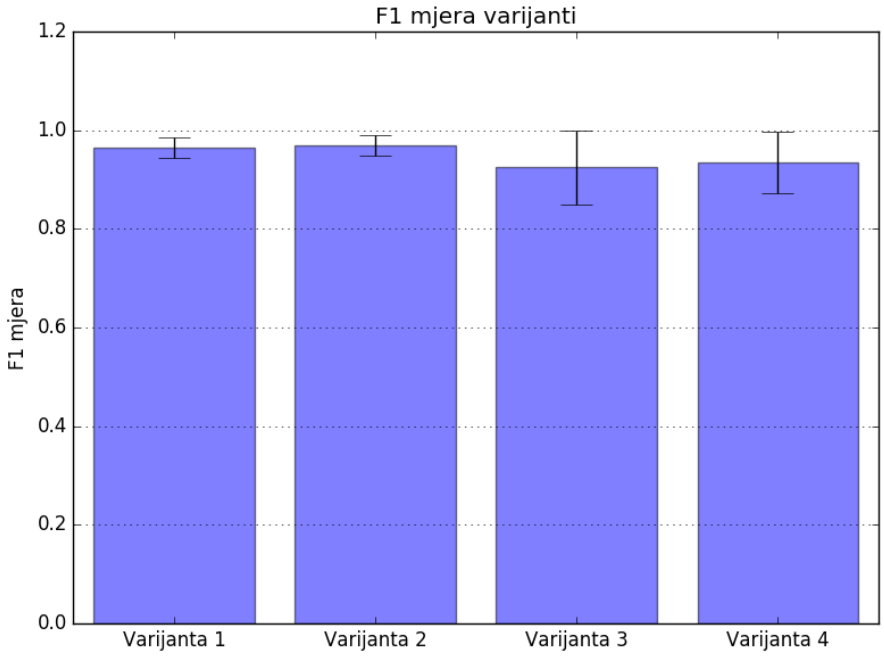


Sl. 5.6: Graf prikladnog broja neurona kod Varijante 2 za *Iris* skup podataka

Cancer skup podataka se sastoji od samo dvije klase, a iz tablica 5.5 i 5.6 se vidi da su podaci iz dviju klasa međusobno veoma različiti. Zato samo dva neurona u skrivenom sloju mogu postići približne rezultate kao i 20 skrivenih neurona. Prve dvije varijante gdje je korišten algoritam za grupiranje podataka pokazuju izričito dobre rezultate čak i kod minimalnog broja skrivenih neurona, a uz to je i standardna devijacija vrlo niska. Druge dvije varijante također pokazuju odlične rezultate, ali s mnogo većom standardnom devijacijom. Takvi rezultati su očekivani, pogotovo kod malog broja skrivenih neurona. Iz danih tablica je vidljivo da postoji velika standardna devijacija u slučajevima gdje ima manje od pet neurona u skrivenom sloju. Na slici 5.7 i slici 5.8 predstavljeni su stupčasti grafovi za preciznost i F1 mjeru. Iz grafova je uočljiva približno jednaka učinkovitost klasifikatora s vrlo niskom standardnom devijacijom za prve dvije varijante. Druge dvije varijante pokazuju nižu vrijednost mjera klasifikacije s nešto većom standardnom devijacijom.

Tab. 5.5: F1 mjera za *Cancer* skup podataka

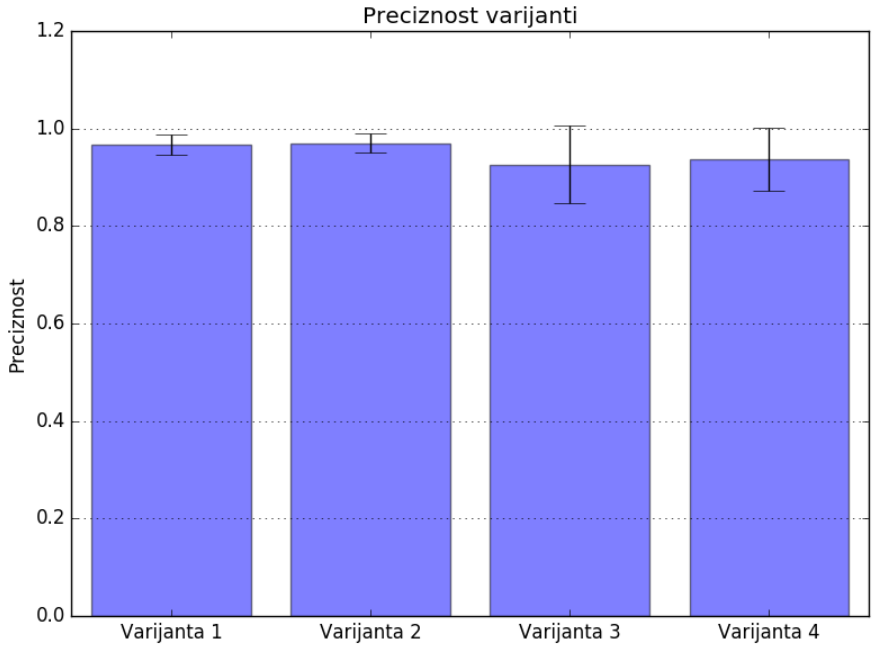
F1 mjera	Varijanta 1	Varijanta 2	Varijanta 3	Varijanta 4
Veličina mreže	Srednja vrijednost +/- standardna devijacija	Srednja vrijednost +/- standardna devijacija	Srednja vrijednost +/- standardna devijacija	Srednja vrijednost +/- standardna devijacija
2	0.9649 +/- 0.0163	0.9648 +/- 0.0136	0.6899 +/- 0.3015	0.7996 +/- 0.257
3	0.9678 +/- 0.0241	0.9693 +/- 0.0267	0.8335 +/- 0.2198	0.7846 +/- 0.2558
4	0.9648 +/- 0.0118	0.9662 +/- 0.0188	0.8627 +/- 0.1987	0.7807 +/- 0.2119
5	0.9649 +/- 0.0238	0.9664 +/- 0.0148	0.8679 +/- 0.162	0.9706 +/- 0.0149
6	0.9677 +/- 0.0227	0.9707 +/- 0.0197	0.9171 +/- 0.0848	0.8696 +/- 0.1813
7	0.9574 +/- 0.0222	0.9707 +/- 0.0147	0.9122 +/- 0.1307	0.9620 +/- 0.0229
8	0.9723 +/- 0.0189	0.9691 +/- 0.0277	0.9512 +/- 0.0416	0.9692 +/- 0.0213
9	0.9614 +/- 0.0271	0.9679 +/- 0.0156	0.9564 +/- 0.0223	0.9723 +/- 0.0177
10	0.9649 +/- 0.0163	0.9708 +/- 0.0285	0.9635 +/- 0.0209	0.9678 +/- 0.0156
11	0.9652 +/- 0.0267	0.9663 +/- 0.0187	0.9624 +/- 0.0256	0.9708 +/- 0.0173
12	0.9678 +/- 0.0158	0.9737 +/- 0.0158	0.9609 +/- 0.027	0.9693 +/- 0.0265
13	0.9649 +/- 0.0162	0.9751 +/- 0.0209	0.9593 +/- 0.0193	0.9694 +/- 0.021
14	0.9616 +/- 0.0309	0.9679 +/- 0.0276	0.9563 +/- 0.0216	0.9662 +/- 0.0209
15	0.9589 +/- 0.0235	0.9693 +/- 0.0179	0.9596 +/- 0.0226	0.9707 +/- 0.0173
16	0.9622 +/- 0.0294	0.9722 +/- 0.0138	0.9592 +/- 0.0315	0.9679 +/- 0.0258
17	0.9634 +/- 0.0209	0.9693 +/- 0.0165	0.9576 +/- 0.0302	0.9680 +/- 0.0141
18	0.9695 +/- 0.0249	0.9692 +/- 0.0223	0.9637 +/- 0.0312	0.9736 +/- 0.0144
19	0.9692 +/- 0.0167	0.9650 +/- 0.0284	0.9666 +/- 0.0215	0.9678 +/- 0.011
20	0.9695 +/- 0.0099	0.9707 +/- 0.0197	0.9635 +/- 0.0239	0.9678 +/- 0.0183



Sl. 5.7: Graf F1 mjere za *Cancer* skup podataka

Tab. 5.6: Preciznost za *Cancer* skup podataka

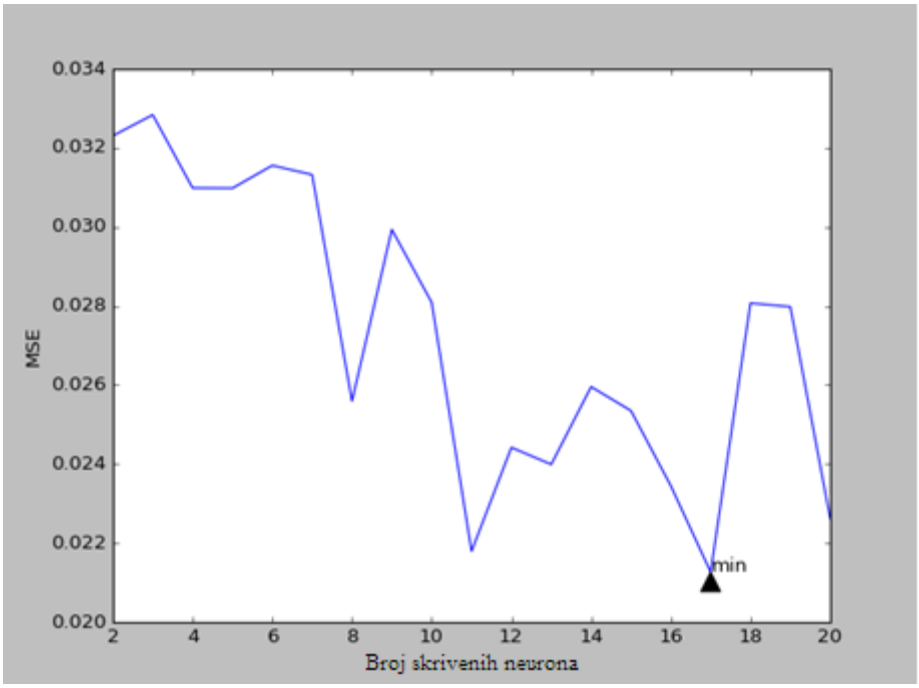
Preciznost	Varijanta 1	Varijanta 2	Varijanta 3	Varijanta 4
Veličina mreže	Srednja vrijednost +/- standardna devijacija	Srednja vrijednost +/- standardna devijacija	Srednja vrijednost +/- standardna devijacija	Srednja vrijednost +/- standardna devijacija
2	0.9657 +/- 0.0159	0.9656 +/- 0.0132	0.7335 +/- 0.2918	0.7759 +/- 0.2951
3	0.9697 +/- 0.022	0.9700 +/- 0.0262	0.8072 +/- 0.2614	0.8030 +/- 0.2709
4	0.9655 +/- 0.0117	0.9668 +/- 0.019	0.8467 +/- 0.2355	0.7816 +/- 0.2302
5	0.9655 +/- 0.0239	0.9673 +/- 0.0151	0.8568 +/- 0.1969	0.9710 +/- 0.0149
6	0.9698 +/- 0.0215	0.9713 +/- 0.0195	0.9351 +/- 0.0524	0.8889 +/- 0.1624
7	0.9597 +/- 0.0209	0.9710 +/- 0.0146	0.9071 +/- 0.1577	0.9626 +/- 0.0228
8	0.9731 +/- 0.0182	0.9706 +/- 0.0266	0.9586 +/- 0.031	0.9695 +/- 0.0212
9	0.9634 +/- 0.0256	0.9690 +/- 0.0154	0.9591 +/- 0.0211	0.9735 +/- 0.0174
10	0.9675 +/- 0.0139	0.9711 +/- 0.0283	0.9649 +/- 0.0202	0.9686 +/- 0.0152
11	0.9674 +/- 0.0249	0.9669 +/- 0.0188	0.9649 +/- 0.0227	0.9717 +/- 0.0166
12	0.9686 +/- 0.0158	0.9745 +/- 0.0158	0.9640 +/- 0.0221	0.9702 +/- 0.026
13	0.9653 +/- 0.0162	0.9756 +/- 0.0207	0.9618 +/- 0.0173	0.9709 +/- 0.0199
14	0.9635 +/- 0.029	0.9695 +/- 0.0269	0.9584 +/- 0.0207	0.9673 +/- 0.0201
15	0.9601 +/- 0.0228	0.9700 +/- 0.0178	0.9633 +/- 0.0193	0.9716 +/- 0.017
16	0.9648 +/- 0.0255	0.9727 +/- 0.0138	0.9618 +/- 0.0307	0.9691 +/- 0.0248
17	0.9655 +/- 0.0193	0.9703 +/- 0.0162	0.9597 +/- 0.0289	0.9691 +/- 0.0135
18	0.9710 +/- 0.0242	0.9701 +/- 0.0221	0.9651 +/- 0.0299	0.9746 +/- 0.0139
19	0.9704 +/- 0.0163	0.9670 +/- 0.0266	0.9685 +/- 0.0198	0.9682 +/- 0.0111
20	0.9710 +/- 0.0094	0.9711 +/- 0.0198	0.9646 +/- 0.024	0.9691 +/- 0.018



Sl. 5.8: Graf preciznosti za *Cancer* skup podataka

Na slici 5.9 je dan graf za određivanje prikladne veličine mreže za Varijantu 2. Prikazani graf ne posjeduje postepeno opadajuću karakteristiku kao kod prethodna dva skupa podataka. Može se

pretpostaviti da vrijednost ovisnosti MSE o veličini mreže teži smanjivanju prilikom povećavanja mreže, ali točke funkcije su više nasumične nego što prate silaznu putanju. Uistinu, ako je korišten algoritam za grupiranje podataka, RBFN daje iste rezultate testiranja neovisno o veličini mreže. Problem može nastati jedino ako su podaci nasumično odabrani za centre radijalnih funkcija te je korišten mali broj skrivenih neurona.



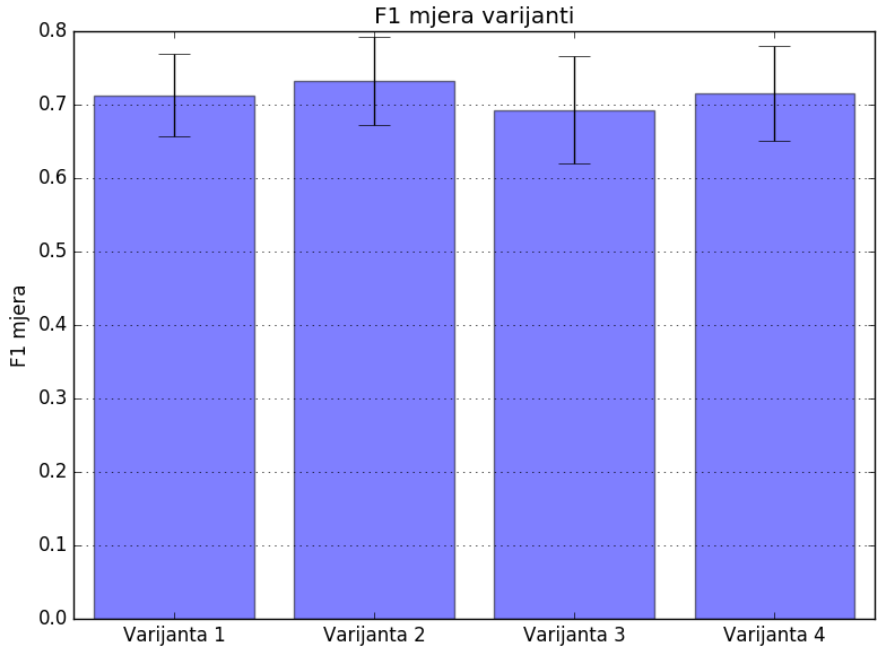
Sl. 5.9: Graf prikladnog broja neurona kod Varijante 2 za *Cancer* skup podataka

Tablica 5.7 i tablica 5.8 prikazuju rezultate analize F1 mjere i preciznosti za *Transfusion* skup podataka. Poslije svake tablice slijedi stupčasti graf (kao na slikama 5.10 i 5.11) kojim je opisana prethodna tablica uzevši u obzir sve veličine mreže. Uočljivo je postepeno poboljšavanje mjera klasifikacije s povećavanjem neuronske mreže.

Tab. 5.7: F1 mjera za *Transfusion* skup podataka

F1 mjera	Varijanta 1	Varijanta 2	Varijanta 3	Varijanta 4
Veličina mreže	Srednja vrijednost +/- standardna devijacija	Srednja vrijednost +/- standardna devijacija	Srednja vrijednost +/- standardna devijacija	Srednja vrijednost +/- standardna devijacija
2	0.6595 +/- 0.0421	0.6601 +/- 0.07	0.5804 +/- 0.1938	0.6209 +/- 0.0824
3	0.6598 +/- 0.0527	0.6601 +/- 0.0643	0.6662 +/- 0.0573	0.6571 +/- 0.0739
4	0.6619 +/- 0.0779	0.6598 +/- 0.0542	0.6641 +/- 0.0361	0.6737 +/- 0.0829
5	0.7000 +/- 0.0603	0.7115 +/- 0.1039	0.6703 +/- 0.0545	0.6992 +/- 0.047
6	0.7097 +/- 0.062	0.7218 +/- 0.0419	0.6804 +/- 0.0635	0.6980 +/- 0.0838
7	0.6933 +/- 0.0963	0.7191 +/- 0.0307	0.6818 +/- 0.0896	0.7108 +/- 0.0656
8	0.7079 +/- 0.046	0.7264 +/- 0.0636	0.6907 +/- 0.0774	0.7140 +/- 0.0667

9	0.6953 +/- 0.0471	0.7321 +/- 0.061	0.6981 +/- 0.083	0.7141 +/- 0.0597
10	0.7141 +/- 0.0549	0.7448 +/- 0.0549	0.7129 +/- 0.0648	0.6960 +/- 0.0728
11	0.7135 +/- 0.0899	0.7485 +/- 0.0545	0.7127 +/- 0.0732	0.7378 +/- 0.0717
12	0.7242 +/- 0.0578	0.7518 +/- 0.0467	0.7144 +/- 0.0667	0.7287 +/- 0.054
13	0.7465 +/- 0.0466	0.7500 +/- 0.0654	0.6867 +/- 0.0585	0.7518 +/- 0.0637
14	0.7374 +/- 0.0396	0.7491 +/- 0.0825	0.7044 +/- 0.0726	0.7378 +/- 0.04
15	0.7396 +/- 0.0603	0.7513 +/- 0.0791	0.7208 +/- 0.0777	0.7522 +/- 0.0774
16	0.7435 +/- 0.0545	0.7676 +/- 0.0553	0.7068 +/- 0.0744	0.7442 +/- 0.0448
17	0.7197 +/- 0.0279	0.7597 +/- 0.047	0.7068 +/- 0.0876	0.7515 +/- 0.0529
18	0.7312 +/- 0.0518	0.7652 +/- 0.0315	0.7238 +/- 0.0632	0.7383 +/- 0.0656
19	0.7310 +/- 0.0417	0.7655 +/- 0.0616	0.7198 +/- 0.0706	0.7333 +/- 0.0697
20	0.7517 +/- 0.0608	0.7701 +/- 0.0653	0.7257 +/- 0.0366	0.7425 +/- 0.0535

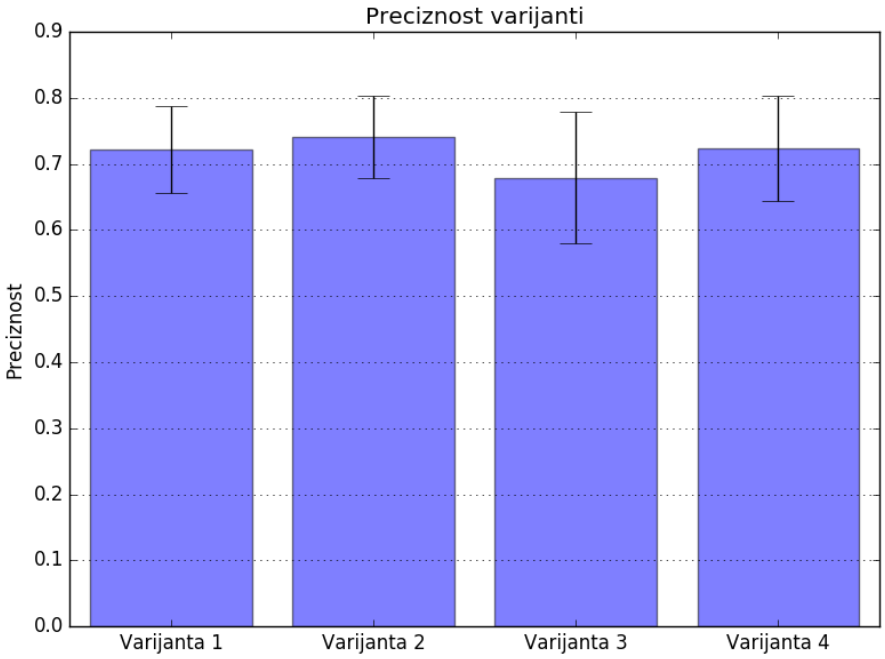


Sl. 5.10: Graf F1 mjere za *Transfusion* skup podataka

Tab. 5.8: Preciznost za *Transfusion* skup podataka

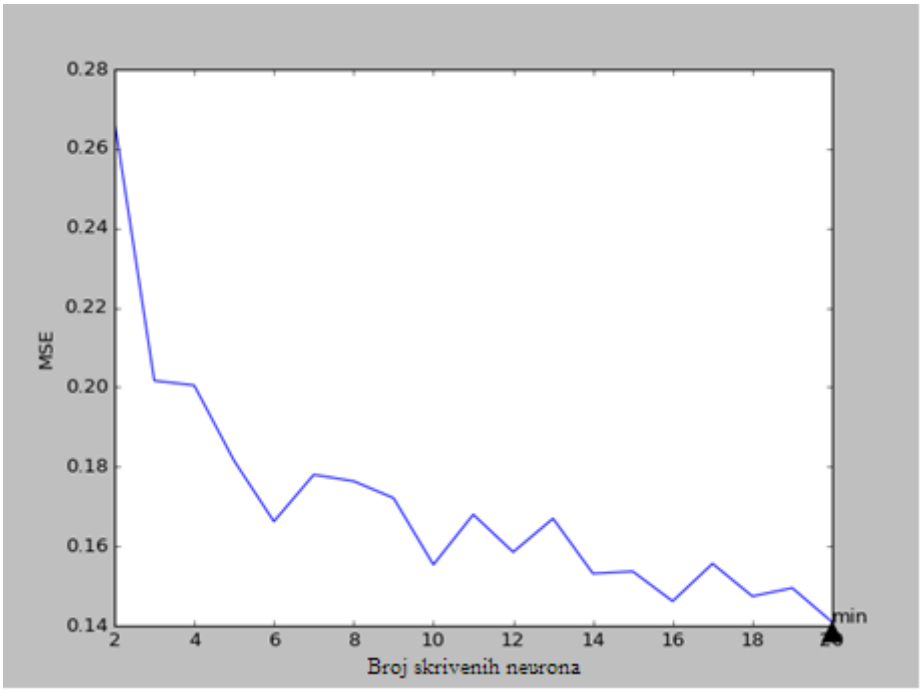
Preciznost	Varijanta 1	Varijanta 2	Varijanta 3	Varijanta 4
Veličina mreže	Srednja vrijednost +/- standardna devijacija	Srednja vrijednost +/- standardna devijacija	Srednja vrijednost +/- standardna devijacija	Srednja vrijednost +/- standardna devijacija
2	0.5816 +/- 0.0472	0.5834 +/- 0.0784	0.5483 +/- 0.2	0.6122 +/- 0.0828
3	0.5823 +/- 0.0589	0.5831 +/- 0.0713	0.5988 +/- 0.0768	0.6334 +/- 0.1111
4	0.5952 +/- 0.0966	0.5824 +/- 0.0599	0.6009 +/- 0.0487	0.6861 +/- 0.1142
5	0.7432 +/- 0.0909	0.7378 +/- 0.0927	0.6374 +/- 0.0849	0.6796 +/- 0.0769
6	0.7445 +/- 0.0787	0.7599 +/- 0.0584	0.6483 +/- 0.1094	0.6653 +/- 0.1252
7	0.7170 +/- 0.1227	0.7661 +/- 0.0574	0.6517 +/- 0.1154	0.7232 +/- 0.0781
8	0.7432 +/- 0.0475	0.7607 +/- 0.0604	0.6702 +/- 0.1244	0.7436 +/- 0.1024
9	0.7158 +/- 0.0756	0.7566 +/- 0.0651	0.6944 +/- 0.1191	0.7262 +/- 0.0806
10	0.7546 +/- 0.0727	0.7883 +/- 0.0502	0.6950 +/- 0.0909	0.6862 +/- 0.1118
11	0.7611 +/- 0.0735	0.7720 +/- 0.0688	0.7152 +/- 0.107	0.7472 +/- 0.0813

12	0.7563 +/- 0.0747	0.7732 +/- 0.0516	0.7102 +/- 0.1118	0.7407 +/- 0.0742
13	0.7685 +/- 0.0651	0.7727 +/- 0.0617	0.6667 +/- 0.101	0.7690 +/- 0.0635
14	0.7435 +/- 0.0376	0.7799 +/- 0.0737	0.7191 +/- 0.0682	0.7419 +/- 0.0488
15	0.7738 +/- 0.0486	0.7679 +/- 0.0571	0.7214 +/- 0.0933	0.7766 +/- 0.068
16	0.7662 +/- 0.0475	0.7913 +/- 0.0579	0.6873 +/- 0.1094	0.7697 +/- 0.0519
17	0.7391 +/- 0.0442	0.7691 +/- 0.0518	0.7030 +/- 0.1223	0.7818 +/- 0.0411
18	0.7394 +/- 0.057	0.7747 +/- 0.0315	0.7247 +/- 0.0738	0.7520 +/- 0.0654
19	0.7352 +/- 0.0427	0.7825 +/- 0.0585	0.7523 +/- 0.0816	0.7388 +/- 0.0882
20	0.7606 +/- 0.0637	0.7725 +/- 0.0728	0.7623 +/- 0.0656	0.7705 +/- 0.0494



Sl. 5.11: Graf preciznosti za *Transfusion* skup podataka

U *Transfusion* skupu podataka postoji uočljiva razlika kod korištenja različitih metoda za određivanje širina radijalnih funkcija. U Varijantama 2 i 4, gdje se koriste jednake širine, ostvareni su bolji rezultati nego u Varijantama 1 i 3. Usporedbom Varijanti 1 i 4, uviđa se da ostvaruju jednake srednje vrijednosti s nešto većom standardnom devijacijom kod Varijante 4. Međutim, kod Varijante 1 je korišten algoritam za grupiranje podataka. Na slici 5.12 je dan graf određivanja prikladne veličine mreže za odabranu Varijantu 2 jer dana Varijanta pokazuje najbolje rezultate preciznosti i F1 mjere. Primjetna je postepena silazna putanja grafa, što navodi da je veći broj neurona pogodniji za klasifikaciju *Transfusion* skupa podataka.



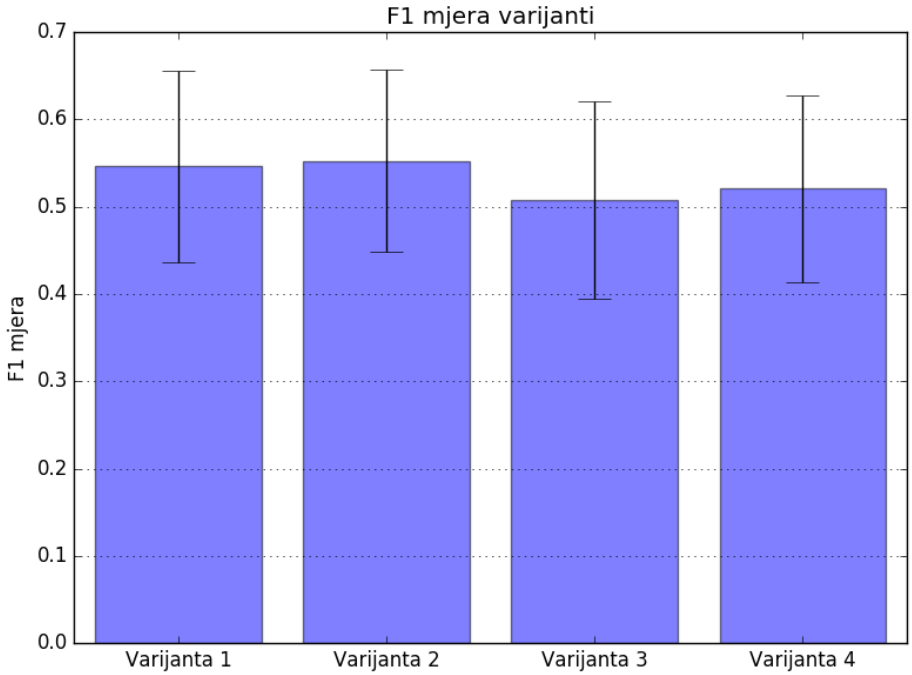
Sl. 5.12: Graf prikladnog broja neurona kod Varijante 2 za *Transfusion* skup podataka

Iz tablica 5.9 i 5.10 za *Glass* skup podataka se vide lošiji rezultati od prijašnjih skupova podataka. Zbog lošijih rezultata je potrebno razmotriti nekoliko činjenica. *Glass* skup podataka se sastoji od šest klasa, od kojih su prve dvije klase većinski zastupljene, dok treća i četvrta, a posebno peta klasa nisu toliko zastupljene tj. podataka tih klasa ima vrlo malo. U analizi je provedena težinska (engl. *weighted*) srednja vrijednost preciznosti i F1 mjere, što znači da je izračunata srednja vrijednost preciznosti od svih klasa uzimajući u obzir koliko podataka se pojavilo u mjerenju za svaku klasu. Ako se promotre mjere klasifikacije za jedan rez (tablica 5.11), uviđa se navedeni problem. Kada se pojave podaci iz treće, četvrte ili pete klase u skupu za testiranje, veoma rijetko budu točno klasificirani. Također, *Glass* skup podataka opisuje mnogo stršećih podataka. Njihovim uklanjanjem u predobradi mogu se postići bolji rezultati klasifikacije.

Tab. 5.9: F1 mjera za *Glass* skup podataka

F1 mjera	Varijanta 1	Varijanta 2	Varijanta 3	Varijanta 4
Veličina mreže	Srednja vrijednost +/- standardna devijacija	Srednja vrijednost +/- standardna devijacija	Srednja vrijednost +/- standardna devijacija	Srednja vrijednost +/- standardna devijacija
2	0.3039 +/- 0.0973	0.2757 +/- 0.0948	0.2439 +/- 0.0966	0.2608 +/- 0.0869
3	0.4040 +/- 0.1239	0.3399 +/- 0.076	0.3227 +/- 0.1187	0.2810 +/- 0.158
4	0.3905 +/- 0.1038	0.4632 +/- 0.1058	0.4473 +/- 0.1328	0.3282 +/- 0.1683
5	0.4598 +/- 0.1388	0.4397 +/- 0.0978	0.3873 +/- 0.1301	0.4670 +/- 0.1342

6	0.4861 +/- 0.0896	0.4902 +/- 0.0683	0.4573 +/- 0.134	0.4352 +/- 0.0522
7	0.5387 +/- 0.0965	0.5070 +/- 0.117	0.5273 +/- 0.1436	0.4999 +/- 0.1024
8	0.5118 +/- 0.0931	0.5560 +/- 0.0829	0.5223 +/- 0.0934	0.5533 +/- 0.092
9	0.5609 +/- 0.1535	0.5436 +/- 0.1001	0.5440 +/- 0.1138	0.5749 +/- 0.1138
10	0.5787 +/- 0.1142	0.5793 +/- 0.1439	0.5257 +/- 0.1181	0.5767 +/- 0.1107
11	0.5819 +/- 0.1062	0.5886 +/- 0.1421	0.5428 +/- 0.0806	0.5538 +/- 0.0826
12	0.6160 +/- 0.0884	0.5951 +/- 0.1069	0.5340 +/- 0.0973	0.5893 +/- 0.1104
13	0.6162 +/- 0.0997	0.6098 +/- 0.119	0.5661 +/- 0.0906	0.5806 +/- 0.0845
14	0.6125 +/- 0.1159	0.6349 +/- 0.1354	0.6214 +/- 0.1634	0.5910 +/- 0.1377
15	0.6239 +/- 0.1046	0.6404 +/- 0.0827	0.5273 +/- 0.119	0.6075 +/- 0.1042
16	0.6281 +/- 0.081	0.6234 +/- 0.093	0.5427 +/- 0.0977	0.6239 +/- 0.0992
17	0.5969 +/- 0.1445	0.6498 +/- 0.118	0.5703 +/- 0.1044	0.6311 +/- 0.1081
18	0.6232 +/- 0.1038	0.6709 +/- 0.1271	0.5856 +/- 0.125	0.5197 +/- 0.1121
19	0.6386 +/- 0.1094	0.6495 +/- 0.1054	0.5866 +/- 0.0954	0.6311 +/- 0.1111
20	0.6005 +/- 0.1108	0.6332 +/- 0.0653	0.5789 +/- 0.0932	0.5831 +/- 0.0688

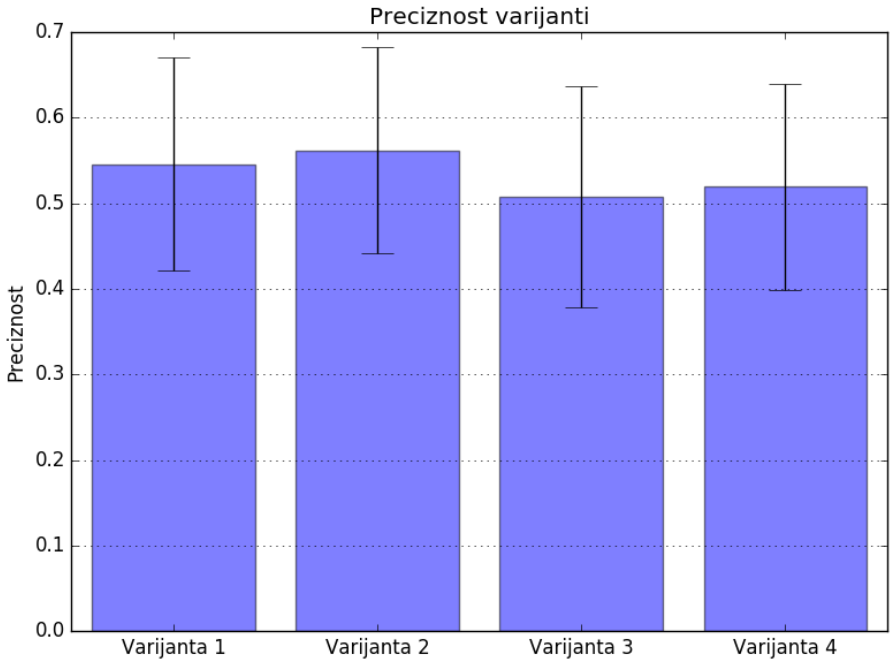


Sl. 5.13: Graf F1 mjere za *Glass* skup podataka

Tab. 5.10: Preciznost za *Glass* skup podataka

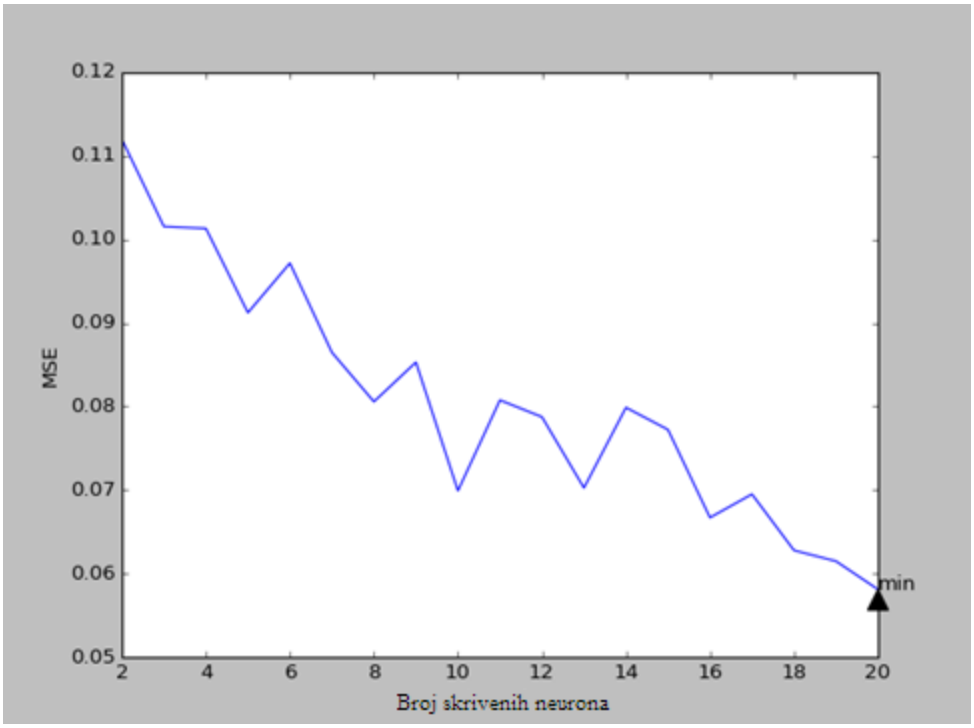
Preciznost	Varijanta 1	Varijanta 2	Varijanta 3	Varijanta 4
Veličina mreže	Srednja vrijednost +/- standardna devijacija	Srednja vrijednost +/- standardna devijacija	Srednja vrijednost +/- standardna devijacija	Srednja vrijednost +/- standardna devijacija
2	0.2615 +/- 0.1234	0.2457 +/- 0.1317	0.2122 +/- 0.0899	0.2126 +/- 0.0924
3	0.4062 +/- 0.1633	0.4040 +/- 0.1127	0.3250 +/- 0.165	0.2547 +/- 0.174
4	0.4260 +/- 0.1756	0.4559 +/- 0.13	0.4774 +/- 0.1663	0.3360 +/- 0.1826
5	0.4451 +/- 0.1545	0.4438 +/- 0.1022	0.3965 +/- 0.1547	0.4676 +/- 0.174
6	0.4824 +/- 0.0906	0.5026 +/- 0.1002	0.4568 +/- 0.1268	0.4176 +/- 0.0745

7	0.5611 +/- 0.1056	0.5424 +/- 0.1241	0.5083 +/- 0.1471	0.5057 +/- 0.1457
8	0.5187 +/- 0.1097	0.5553 +/- 0.1092	0.4964 +/- 0.1044	0.5396 +/- 0.1074
9	0.5652 +/- 0.1616	0.5410 +/- 0.1041	0.5394 +/- 0.114	0.5801 +/- 0.1027
10	0.5712 +/- 0.1257	0.5819 +/- 0.1521	0.5119 +/- 0.1345	0.5967 +/- 0.1471
11	0.5767 +/- 0.0924	0.5671 +/- 0.1566	0.5378 +/- 0.0884	0.5342 +/- 0.0826
12	0.6106 +/- 0.0867	0.6195 +/- 0.1284	0.5431 +/- 0.1278	0.5906 +/- 0.1238
13	0.6241 +/- 0.1166	0.6211 +/- 0.147	0.5632 +/- 0.1089	0.5777 +/- 0.0805
14	0.6179 +/- 0.1402	0.6438 +/- 0.1336	0.6445 +/- 0.1806	0.6051 +/- 0.1367
15	0.6050 +/- 0.1163	0.6606 +/- 0.1081	0.5310 +/- 0.136	0.6258 +/- 0.1342
16	0.6211 +/- 0.0966	0.6399 +/- 0.1208	0.5253 +/- 0.1029	0.6290 +/- 0.0965
17	0.6089 +/- 0.1494	0.6569 +/- 0.132	0.5732 +/- 0.1309	0.6313 +/- 0.1269
18	0.6264 +/- 0.1124	0.7019 +/- 0.1174	0.5940 +/- 0.1353	0.5223 +/- 0.1422
19	0.6459 +/- 0.1373	0.6537 +/- 0.1241	0.6074 +/- 0.1188	0.6532 +/- 0.0798
20	0.5962 +/- 0.116	0.6402 +/- 0.0619	0.5986 +/- 0.1128	0.5898 +/- 0.0817



Sl. 5.14: Graf preciznosti za *Glass* skup podataka

Na slici 5.15 je dan graf prikladne veličine mreže na kojem funkcija prati silaznu putanju pri povećavanju broja skrivenih neurona. Kao i kod prethodnih skupova podataka, algoritmom *k*-means ostvareni su bolji rezultati nego odabirom nasumičnih podataka za centre radijalnih funkcija. Dok korištenje grupiranja pokazuje znatno poboljšanje preciznosti i F1 mjere, u nekim skupovima podataka (poput *Glass* skupa) metoda koja implementira jednake širine rezultira blagom prednošću naspram pravila p-najbližih susjeda.



Sl. 5.15: Graf prikladnog broja neurona kod Varijante 2 za *Glass* skup podataka

Tab. 5.11: Primjer neravnoteže između raspodjele podataka među klasama za jedan rez

Klasa	Preciznost	Opoziv	F1 mjera	Pojavljivanje
1	0.50	0.80	0.62	5
2	0.75	0.60	0.67	10
3	0.00	0.00	0.00	1
4	0.00	0.00	0.00	0
5	0.00	0.00	0.00	2
6	0.60	0.75	0.67	4
Težinska srednja vrijednost/ukupni broj podataka	0.56	0.59	0.56	22

Iz tablice 5.11 se vidi broj pojavljivanja podataka za svaku klasu i izračunate težinske srednje vrijednosti. Često se pojavi slučaj u nekom rezu skupa da gotovo svi podaci koji pripadaju manjinskoj klasi završe u skupu za testiranje (ili se uopće ne pojave u skupu za testiranje). Stoga je vrlo malo takvih podataka u fazi treniranja mreže te neuronska mreža nije sposobna klasificirati određenu manjinsku klasu. U *Glass* skupu podataka, mreža najčešće klasificira nove podatke u prvu, drugu ili šestu klasu, dok preostale tri klase ne budu zastupljene nakon klasifikacije.

6. ZAKLJUČAK

Diplomskim je radom opisan problem klasifikacije kao i radijalne neuronske mreže za potrebe njegovog rješavanja, izrađeno programsko rješenje za korisnike te izvršena eksperimentalna analiza na pet skupova podataka koristeći različite parametre neuronske mreže. U praktičnom dijelu rada ostvareno je programsko rješenje koje obuhvaća radijalnu neuronsku mrežu s grafičkim radnim okvirom. Programsko rješenje omogućuje korisnicima učitavanje vlastitih skupova podataka, podešavanje parametara te treniranje i testiranje neuronske mreže. Nakon izračuna, korisnici imaju uvid u prikladnu veličinu radijalne neuronske mreže pomoću grafa, kao i dobivenu matricu zbunjenosti s izračunatim mjerama klasifikacije poput točnosti, preciznosti i F1 mjere. Izvršeno je eksperimentalno testiranje i vrednovanje klasifikatora na pet različitih skupova podataka. Analiza ukazuje na važnost korištenja algoritma za grupiranje podataka kod određivanja lokacija radijalnih funkcija. Varijante analize u kojima je korišten algoritam *k*-means pokazuju znatno bolje rezultate od odabiranja nasumičnih podataka za centre. Korištenje različitih metoda za određivanje širina nema znatan utjecaj na rezultate. Međutim, ako nije korišten algoritam za grupiranje podataka, metoda jednakih širina, neočekivano, pokazuje bolje rezultate od pravila p-najbližih susjeda. Budući rad može obuhvatiti implementiranje dodatnih metoda za određivanje parametara aktivacijske funkcije. Također je moguće proširiti funkcionalnost programskog rješenja kako bi se mogle učitati datoteke skupova podataka zapisane i u drugim često korištenim formatima.

Literatura

[1] C. M. Bishop, *Neural Networks for Pattern Recognition*. Oxford University Press, Inc., Birmingham, UK, 1995.

[2] Kevin P. Murphy, *Machine Learning: A Probabilistic Perspective*. The MIT Press, Cambridge, Massachusetts, London, England, 2012.

[3] STAT 897D , URL: https://newonlinecourses.science.psu.edu/stat857/node/147/ (2.9.2018.)

[4] Chris McCormick, URL: http://mccormickml.com/2013/08/15/radial-basis-function-network-rbfn-tutorial/ (2.9.2018.)

[5] A. P. Engelbrecht, *Computational Intelligence: An Introduction, Second Edition*. John Wiley & Sons Ltd, England, 2007.

[6] Radial Basis Function Networks: Applications, Introduction to Neural Networks: Lecture 14, John A. Bullinaria, 2004

[7] R. Xu and D. C. Wunsch, *Clustering*, John Wiley & Sons Inc., Hoboken, New Jersey 2007.

[8] R. Kruse, C. Borgelt, F. Klawonn, C. Moewes, M. Steinbrecher, and P. Held, *Computational Intelligence: A Methodological Introduction*. Springer-Verlag, London, 2013.

[9] Python Data Science Handbook, URL: https://jakevdp.github.io/PythonDataScienceHandbook/05.11-k-means.html (2.9.2018.)

[10] X. Wu, V. Kumar, J. Ross Quinlan, J. Ghosh, Q. Yang, H. Motoda, G. J. McLachlan, A. Ng, B. Liu, P. S. Yu, Z.-H. Zhou, M. Steinbach, D. J. Hand, and D. Steinberg, Top 10 algorithms in data mining, *Knowledge and Information Systems*, br. 14, sv. 1, str. 1–37, 4. prosinca 2007.

[11] C. M. Bishop, *Pattern Recognition and Machine Learning*, Springer Science + Business Media LLC, Cambridge CB3 0FB, UK, 2006.

[12] N. Benoudjit, C. Archambeau, A. Lendasse, J. Lee, M. Verleysen, Width optimization of the Gaussian kernels in Radial Basis Function Networks, d-side publi., str. 425-432, Bruges, Belgium, 24.-26. travnja 2002.

[13] N. Benoudjit and M. Verleysen, On the Kernel Widths in Radial-Basis Function Networks, *Neural Processing Letters*, br. 2, sv. 18, str. 139–154, 2003.

[14] D. S. Broomhead and D. Lowe, Multivariable functional interpolation and adaptive networks, *Complex Systems*, br. 3, sv. 2, str. 321-355, 1988.

[15] Machine Learning Mastery, URL: https://machinelearningmastery.com/classification-accuracy-is-not-enough-more-performance-measures-you-can-use/ (2.9.2018.)

[16] S. Theodoridis and K. Koutroumbas, *Pattern Recognition, Fourth Edition*. Academic Press, Elsevier Inc., Burlington, Massachusetts, San Diego, California, London, UK, 2009.

[17] Foram S. Panchal and Mahesh Panchal, Review on Methods of Selecting Number of Hidden Nodes in Artificial Neural Network, *International Journal of Computer Science and Mobile Computing*, br. 11,sv. 3, str. 455-464, 2014.

[18] A. Asuncion and D. Newman, UCI machine learning repository, 2007. URL: http://mlr.cs.umass.edu/ml/index.html (22.9.2018.)

Sažetak

U diplomskom radu razmotren je problem klasifikacije kojem je pristupljeno metodom umjetne neuronske mreže, preciznije, korištena je radijalna neuronska mreža. Radijalne neuronske mreže posjeduju nekoliko parametara koje je potrebno odrediti, a za aktivacijske funkcije koriste radijalne funkcije. Neuroni skrivenog sloja implementiraju algoritam za grupiranje podataka zvan *k*-means. Izrađeno je programsko rješenje koje korisnicima omogućuje korištenje neuronske mreže putem intuitivnog grafičkog radnog okvira. Provedena je eksperimentalna analiza nad pet skupova podataka koristeći različite metode za određivanje parametara radijalne neuronske mreže.

Ključne riječi: aktivacijska funkcija, grupiranje podataka, klasifikacija, radijalna neuronska mreža, umjetna neuronska mreža

Abstract

Radial basis function network design for classification using data clustering

Graduation thesis deals with a problem of classification by an artificial neural network approach, more precisely, by using a radial basis neural network. Radial basis neural networks have several parameters to determine, and radial basis functions are used for activation functions. The hidden layer neurons implement data clustering algorithm called *k*-means. A desktop application that allows users to use a neural network through an intuitive graphical interface was developed. An experimental analysis of five data sets was performed by using different methods for determining radial basis neural network parameters.

Key words: activation function, data clustering, classification, radial basis neural network, artificial neural network

Životopis

Antonio Falak rođen je 1. travnja 1994. godine u Osijeku, gdje trenutno živi i završava fakultetsko obrazovanje. Pohađao je OŠ "Tin Ujević" u Osijeku u razdoblju od 2001. do 2009. godine. Svoje daljnje obrazovanje nastavlja u srednjoj školi Elektrotehnička i prometna škola Osijek, u kojoj 2013. godine stječe zvanje elektrotehničar. Završetkom srednje škole, iste godine uspješno upisuje Fakultet elektrotehnike, računarstva i informacijskih tehnologija u Osijeku gdje nakon tri godine završava preddiplomski sveučilišni studij računarstva.

Potpis:

_____

Prilozi /na CD-u/

- Diplomski rad u .doc, .docx i PDF formatu
- Projekt programskog rješenja
- Korišteni skupovi podataka

GitHub repozitorij programskog rješenja: https://github.com/afalak94/Radial-Basis-Function-Neural-Network