|  |
| --- |
| Elektrotehnički fakultet |
| Neuralne mreže |
| Prvi projekat |
|  |
| **Jovana Jaćimović 2019/0593** |
| **Ana Radovanović 2019/0282** |

|  |
| --- |
| Januar 2021. |

Sadržaj

1. [Uvod 3](#_Toc94125555)
2. [Zadatak 1 4](#_Toc94125556)
3. [Zadatak 2 15](#_Toc94125557)

# Uvod

Neuralna mreža je sistem koji se sastoji od određenog broja međusobno povezanih procesnih elemenata koje nazivamo veštačkim neuronima. Neuralne mreže su ime dobile jer su pravljene po uzoru na ljudski mozak. Pokazuje se da su vrlo efikasne u zadacima kao što su mečovanje i klasifikacija oblika, aproksimacija funkcija, optimizacija, vektorska kvantizacija, klasterizacija podataka.

Arhitektura neuralne mreže predstavlja specifično povezivanje neurona u jednu celinu. Struktura neuronske mreže se razlikuje po broju slojeva. Prvi sloj se naziva ulazni, a poslednji izlazni, dok se slojevi između nazivaju skriveni slojevi. Slojevi komuniciraju tako što se izlaz svakog neurona iz prethodnog sloja povezuje sa ulazima svih neurona narednog sloja. Znači, svaki čvor ima nekoliko ulaza i jedan izlaz. Jačina veza kojom su neuroni povezani naziva se težinski faktor (weight).

Mi ćemo u ovo projektu obučiti dve neuralne mreže u različite svrhe. Prva mreža će imati zadatak da klasifikuje podatke u jednu od dve kategorije, dok druga mreža treba da napravi predikciju da li će padati kiša u odnosu na meterološke podatke od prethodnog dana.

# Zadatak 1

Zadatak : database3

Cilj: Klasifikacija podataka primenom feedforward neuralne mreže.

Dati skup podataka sastoji se iz dva obeležja i izlaza, koji predstavlja pripadnost jednoj od tri klase (K1,K2,K3). Tehnikom “one-hot encoding” pravimo matricu izlaza. Ova tehnika podrazumeva da izlaz iz neuralne mreže koji ima n kolona i jedan red svedemo na matricu gde će broj redova predstavljati broj različitih klasa, a broj kolona će ostati isti. Tako u našem primeru, pravimo matricu koja će imati 1500 kolona i 3 vrste. Izlaz će u svakoj koloni imati samo jednu vrednost koja će biti jedinica, dok će sve ostale vrednosti biti nule, pa će tako mesto jedinice i odrediti pripadnost određenoj klasi.

Pri vizuelizaciji, ulazne podatke smo podelili u 3 matrice, u zavisnoti od toga kojoj klasi pripadaju, a nakon toga uradili i njihov grafički prikaz na dijagramu.

Chart, scatter chart

Description automatically generated

SLIKA 1 Vizuelizacija podataka po klasama, gde su u klasu n, svrstani podaci čija je vrednost izlaza n

Nakon vizuelizacije izvršili smo podelu podataka na trening i test skup. Bilo je i neophodno transponovati matricu ulaza i izlaza, pošto dataset iz koga smo čitali podatke nije bio strukturno podešen za direktno korišćene, tačnije matrica podataka bila je transponovana . Za trening skup uzeli smo 80% ukupnih podataka, dok je za test skup uzeto preostalih 20%. Ova podela je neophodna, kako bi performanse neuralne mreže bile validne. Kako su podaci balansirano podeljeni, možemo izvršiti slučajan odabir podataka unutar jenog zajedničkog skupa, umesto tri pojedinačna skupa za svaku od klasa. Odabir podataka izvršen je slučajno, korišćenjem nasumične permutacije skupa indeksa svih ulaznih podataka, korišćenjem funkcije randperm. Bitno je da i test i trening podaci imaju odbirke svih klasa, kako se pri simulaciji ne bi grešilo zbog neadekvatnih ulaznih podataka.

Kada smo ulazne podatke podelili, sledeći korak je kreiranje neuralne mreže. U zadatku je traženo kreiranje tri neuralne mrže sa različitim arhitekturama i istim parammetrima za treniranje (broj epoha, maksimalna dozvoljena greška, minimalan gradijent).

Parametri treniranja:

* Broj epoha = 000
* Maksimalna dozvoljena greška = 10-3
* Minimalan gradijent = 10-4
* Aktivaciona funkcija skrivenih slojeva = poslin

Arhitekture:

* Optimalna neuralna mreža – optimal network: → 3 skrivena sloja sa redom 6,5,4 neurona
* Preobučena neuralna mreža – overfitting network: →3 skrivena sloja sa redom 130,100,80 neurona
* Neuralna mreža koja ne može ispratiti dinamiku podatala – unerfitting network →3 skrivena sloja sa redom 3,2 neurona

Takođe, po zahtevu zadataka potrebno je isključiti zaštitu od preobučavanja, pa je funkcija divideFcn postavljena na prazan string.

Sada, kada smo zavrsili sa definisanjem parametara, možemo izvršiti obučavanje neuralnih mreža.

Na datim slikama možemo videte krive perfomansi gore definisanih neurnih mreža.

* Optimalna neuralna mrežа

Graphical user interface

Description automatically generated with low confidence

SLIKA 2Kriva performanse optimalne neuralne mreže

* Neuralna mreža koja ne može da isprati dinamiku podataka

Graphical user interface

Description automatically generated

SLIKA 3 Kriva performanse neuralne mreže koja ne može ispratiti dinamiku podataka

* Preobučena neuralna mreža

Chart

Description automatically generated

SLIKA 4 Kriva performanse neuralne mreže koja dovodi do preobučavanja

Posle treniranja neuralnih mreža, izvršavamo i testiranje istih na osnovu trening skupa.

Na datim slikama prikazujemo konfuzione matrice za trening i test skup:

* Optimalna neuralna mreža

Table

Description automatically generated

SLIKA 5 Konfuizona matrica za optimalnu neuralnu mrežu za trening skup

Table

Description automatically generated

SLIKA 6 Konfuziona matrica za optimalnu mrežu za test skup

* Neuralna mreža koja ne može da isprati dinamiku podataka

Table

Description automatically generated

SLIKA 7 Konfuziona matrica koja ne može da isprati dinamiku podataka nm za trening skup

Table

Description automatically generated

SLIKA 8Konfuziona matrica koja ne može da isprati dinamiku podataka nm za test skup

* Preobučena neuralna mreža

Table

Description automatically generated

SLIKA 9 Konfuziona matrica za mrežu koja dovodi do preobučavanja za tening skup

Table

Description automatically generated

SLIKA 10Konfuziona matrica za mrežu koja dovodi do preobučavanja za test skup

Nakon dobijenih rezultata testiranja, radimo granicu odlučivanja za neuralne mreže, kako bi imali vizuelan prikaz performansi mreža u grupisanju.

Na datim slikama prikazujemo granice odlučivanja neuralnim mreža:

* + Optimalna neuralna mrežaGraphical user interface, chart

    Description automatically generated with medium confidence

SLIKA 11 Granica odlučivanja optimalne neuralne mreže

Možemo zaključiti da je optimalna neuralna mreža je uspela da se obuči veoma zadovoljavajuće. Odbirci klasa pretežno su dobro klasifikovani, što se može videti sa slike.

* Neuralna mreža koja ne može da isprati dinamiku podataka

Graphical user interface, chart, scatter chart

Description automatically generated

SLIKA 12 Granica odlučivanja neuralne mreže koja ne može ispratiti dinamiku podataka

Zbog premale arhitekture neuralne mreže, mreža nije mogla da isprati dinamiku podataka, pa samim tim nije uspela da se adekvatno obuči. Samim tim, klasifikacija podataka je takođe neuspešna, odnosno mreža je sve podatke test skupa priključila prvoj klasi, sto znaci da nije uspela da klasifikuje podatke druge i treće klase.

* Preobučena neuralna mreža

Graphical user interface, map

Description automatically generated

SLIKA 13 Granica odlučivanja neuralne mreže koja vodi do preobučavanja

Sa slike možemo videti da se neuralna mreža previše navikla na ulazne trening podatke i da zbog toga ne daje dovoljno dobre rezultate na test skupu podataka. Može se videti da je zbog dva ulazna odbirka klase 1, neuralna mreža obojila veću površinu pravom bojom.

Na osnovu dobijenih rezultata možemo zaključiti da feedforward mreže imaju osobinu da vrlo dobor rešavaju zahteve u kojima je potrebno izvršiti klasifikaciju linearno neseparabilnih klasa. Takođe, možemo uvideti da veći broj neurona u skrivenim slojevima ne predstavlja nužno i bolje performanse neuralne mreže, kao što bi se na prvu loptu očekivalo. Problem će nastati kada se neuralna mreža previše navikne na ulazne podatke i tako „napamet“ nauči kako da klasifikuje ulaze. Ovaj problem naziva se preobučavanje, a rešava se raznim tehnikama kao što su regularizacija, rano zaustavljanje...Takođe, možemo videti da arhitektura sa premalo neurona ipak nema mogućnost klasifikacije podataka. Performanse optimalne neuralne mreže mogle bi da se povećaju ako bi se broj ulaznih parametara povećao, a samim tim i broj podataka trening skupa.

# Zadatak 2

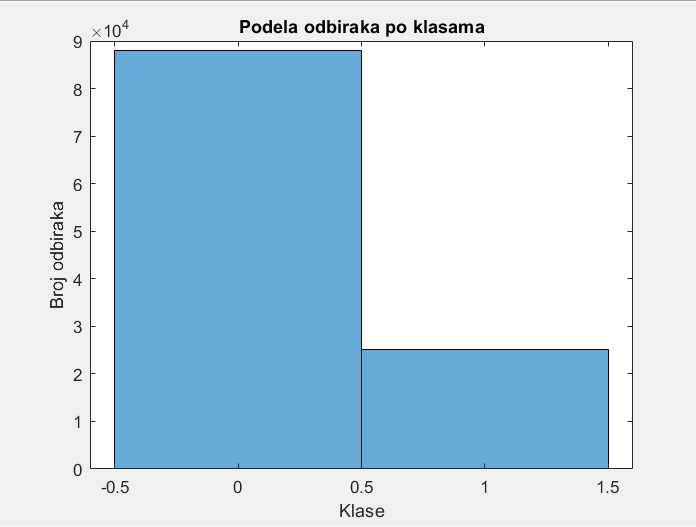
## Struktura mreže

U ovom zadatku se obrađuju podaci primenom feedforward neuralne mreže korišćenjem unakrsne validacije. U ulaznom sloju imaćemo 16 neurona jer imamo 16 ulaznih podataka koji nose informacije o temperaturi u toku dana, pravacu i brzini duvanja vetra, vlazi i pritisku vazduha. U izlaznom sloju ćemo imati 1 neuron jer nam je potrebna povratna informacija samo da li će naredni dan kiša da pada ili ne. Arhitekturu skrivenog sloja ćemo odabrati procesom krosvalidacije.

## Raspodela odbiraka po klasama

Prvo se podaci moraju učitati u dva niza, ulazni i izlazni tako da se diskretni tekstualni podaci pretvore u numeričke vrednosti. Zatim ćemo podeliti podatke u dve klase, jedna koja kao izlazni parametar ima vrednost „da“ i drugu za koju je izlazni parametar „ne“.

U ovom ulaznom skupu navedene klase nisu balansirane tj. ima znatno više podataka u drugoj klasi što se može videti na histogramu ispod.



## Izdvajanje podataka za treniranje i testiranje

S obzirom na nebalansiranost klasa podela podataka ne može da se izvrši nasumično. Bitno je da se podaci proporcionalno podele tako da i u skupu za treniranje i u skupu za testiranje bude srazmerno mnogo podataka iz obe klase. Kada ne bi obratili pažnju na ovo moglo bi da se desi da u skupu za trening nemamo podatke iz obe klase i da se mreža ne obuči dobro ili da se to desi u skupu za testiranje i da ne dobijemo stvarne performanse mreže.

Skup za trening se sastoji iz dva podskupa: onaj na kome se mreža stvarno obučava i skup za validaciju. Skup za validaciju koristimo pri metodi ranog zaustavljanja da se mreža ne bi preobučila. Prilikom odvajanja skupa za validaciju takođe treba paziti da podaci budu ravnomerno raspodeljeni između klasa.

Da bi bili sigurni da je podela pravilno odrađena sami ćemo podeliti obe klase na tri dela od po 70%, 15% i 15% respektivno za skup za trening, validaciju i testiranje. Zatim ćemo spojiti delove prve i druge klase za svaki od tri skupa i promeniti redosled podataka u svakom skupu slučajno tako da ne budu prvo svi podaci jedne klase da onda svi podaci druge.

## Unakrsna validacija

U procesu unakrsne validacije ili krosvalidacije menjaćemo 3 hiperparametra: arhitekturu mreže, konstantu obučavanja i koeficijent regularizacije.

Arhitektura mreže je bitan faktor. Treba odabrati optimalan broj slojeva i neurona u svakom sloju tako da mreža bude dovoljno velika da može da sadrži informaciju o tome kako da donese odluku, ali ne previše velika da ne bi na pamet naučila ceo skup za treniranje.

Konstanta obučavanja η označava koliko će jako nove promene biti primenjene na trenutne težine. Što je koeficijent veći algoritam brže konvergira i možemo povećavati η u toku obučavanja ako greška opada u više iteracija konzistentno. Prevelike vrednosti dovode do velikih promena težina što prouzrokuje nemogućnost da se dođe do lokalnog minimuma, dok premale vrednosti previše sporo konvergiraju ka minimumu.

Regularizacija je mehanizam uklanjanja ili umanjivanja pojedinačnih odstupanja na nivou skupa za obučavanje. Koeficijentom regularizacije 𝜆 se ograničava uticaj regularizacionog člana na gradijente. Suštinski, povećavanjem regularizacionog koeficijenta, smanjuje se odstupanje promena težina od proseka pri ažuriranju i na taj način promene težina se čuvaju bliskim nuli. Naravno, ukoliko je ova vrijednost prevelika, mreža može biti spriječena da uči bilo kakva obilježja. Ovaj koeficijent se bira tako da se mreža spreči u učenju sitnih detalja specifičnih za pojedinačne uzorke na ulazu, a da pri tom zadrži mogućnost učenja generalnijih obeležja.

Mreži ćemo proslediti parametre:

* Broj epoha : 1000
* Maksimalna dozvoljena greška: 10-4
* Minimalni gradijent: 10-5
* Funkcije aktivacije : tansig
* Arhitekture mreže: [15, 10, 8, 5], [17, 12, 7], [4, 5, 8, 6], [20, 15, 10]
* Koeficijent regularizacije: 0.2, 0.5, 0.8
* Konstanta obučavanja: 0.5, 0.1, 0.05

Budući da su ulazni podaci nebalansirani kao meru performanse koristimo:

, gde je P preciznost, a R osetljivost.

Optimalni hiperparametri dobijeni nakon obučavanja su:

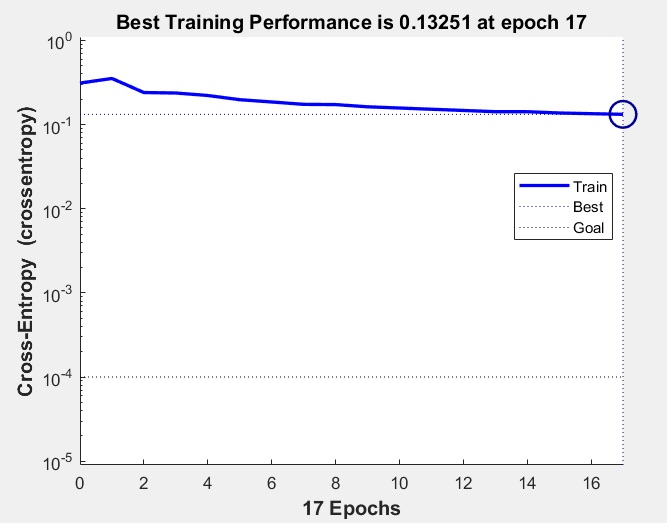
* Arhitekture mreže: [20, 15, 10]
* Koeficijent regularizacije: 0.8
* Konstanta obučavanja: 0.1

Broj epoha neophodan da se mreža obuči je 17.

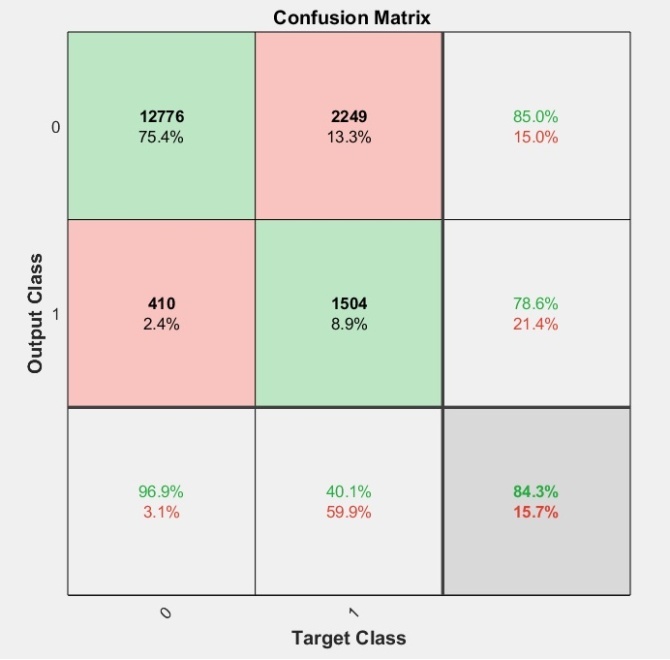
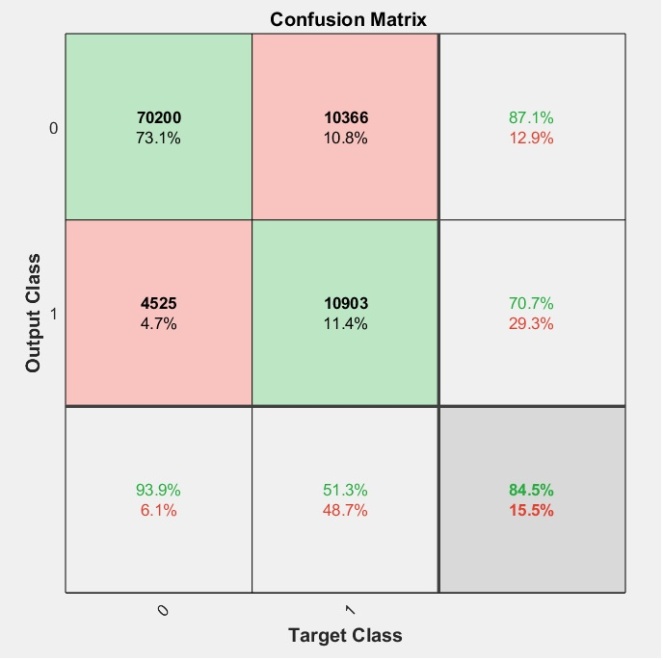
## Obučavanje mreže

Dalje treba mrežu obučiti sa hiperparametrima dobijenim unakrsnom validacijom.

Na grafiku ispod se vidi kriva perfomanse neuralne mreže u zavisnosti od broja epoha.



Da bi videli prave performance mreže moramo kroz nju pustiti skup podataka za testiranje i uprediti podatke sa skupom za treniranje. Na slikama ispod se vide matrice konfuzije za skup za treniranje(levo) i testiranje(desno). Vidimo da se podaci razlikuju za manje od 1%, dakle mreža je dobro obučena tj da nije došlo do preobučavanja.



Preciznost i osetljivost za mrežu su:

P = TP/TP+FP = 0,717

R = TP/TP+FN = 0,116