Neuralne mreže

Izveštaj o projektnom zadatku

Dušan Galić 0092/2014

Jovana Trifunović 0589/2014

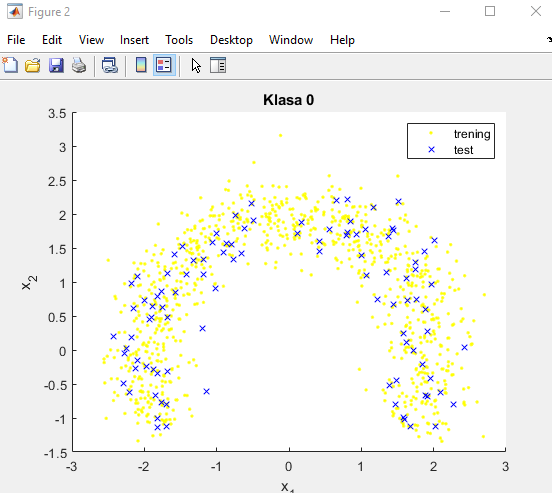
# Zadatak 1

## Vizualizacija podataka po klasama

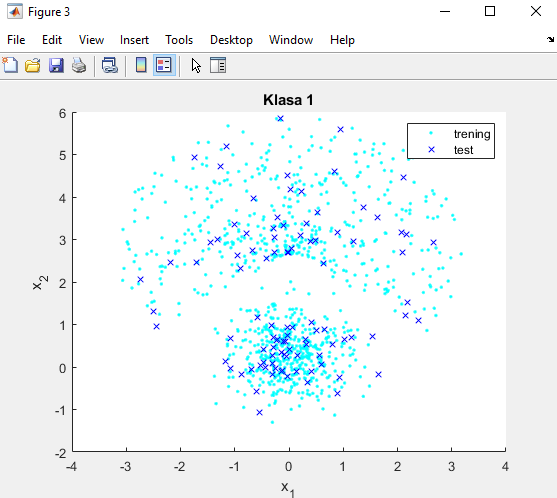
## 

1.Ulazni podaci, obe klase

## Podela podataka



2.Ulazni podaci, klasa 0



3.Ulazni podaci, klasa 1

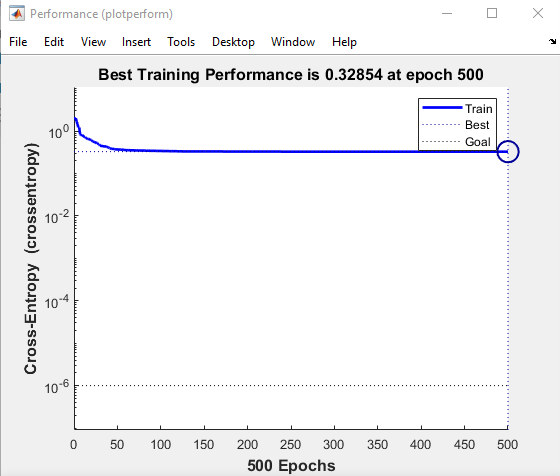
Podela podataka na trening I test skup nam je potrebna da bismo mogli da se uverimo da će mreža davati dobre rezultate na nezavisnom skupu, jer nam ne treba mreza koja će davati dobre rezultate samo na skupu na kom se obučila. Kako pravimo test skup je isto bitno, ne mozemo nasumično da uzimamo iz celog skupa trening podataka, nego uzimamo od obe klase po 10% podataka. Na ovaj način dobijamo istu proporciju podataka jedne I druge klase u test skupu kao što je I u početnom.

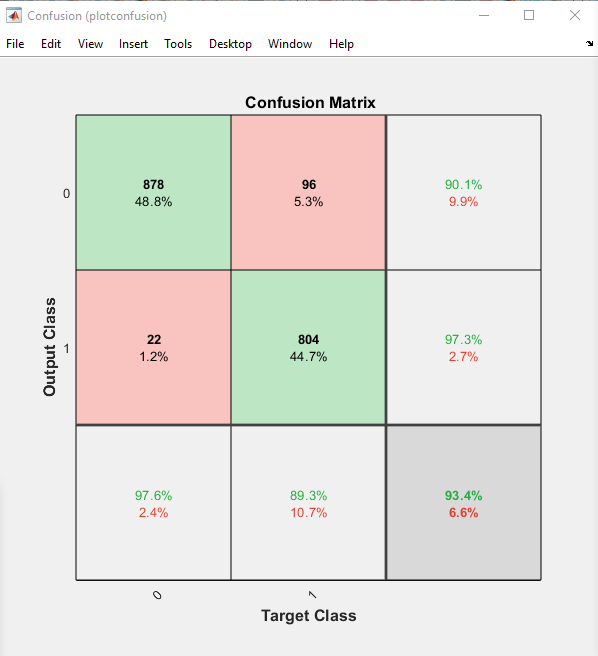
## Formiranje i obučavanje neuralne mreže

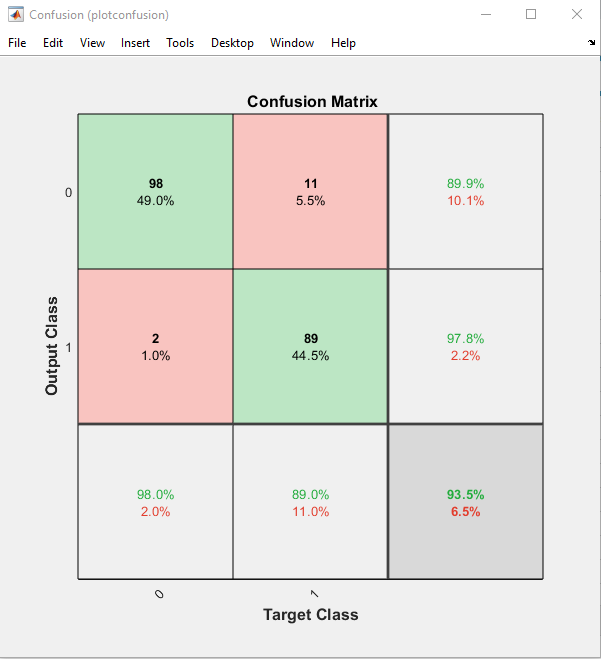
Pošto su podaci balansirani, nećemo koristiti težine za klase pri treniranju. Takođe, pošto je rečeno da ne koristimo metode zaštite od preobučavanja, postavljamo valRatio na nulu jer ne koristimo rano zaustavljanje. Uzimamo tansig kao aktivacionu funkciju za skriveni sloj.

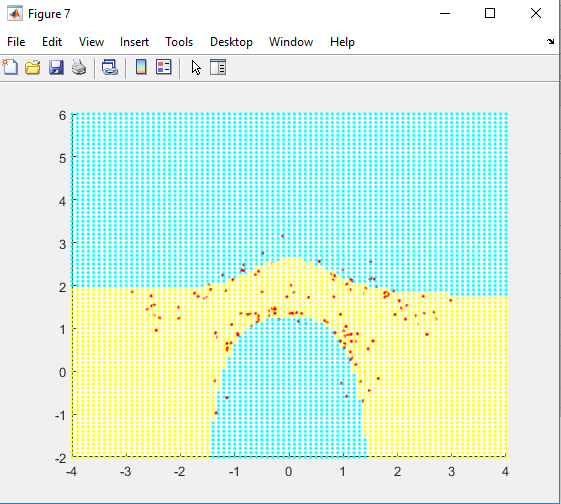
## Arhitektura neuralne mreže

Način na koji smo mi došli do najbolje arhitekture je postavljanje na početnu arhitekturu sa većim brojem neurona i postepeno uklanjanje neurona (tzv. pruning), gledajući da accuracy ostane visok. Uklanjanjem neurona accuracy ostaje visok sve do [4 4] skrivenog sloja. Nakon toga vec [3 3] nije u stanju da obradi mrezu (underfitted). Za demonstraciju smo uzeli skrivene slojeve [5, 5], [2] i [30 30 30 30] kao primere za najbolju, underfitovanu i overfitovanu mrežu, respektivno.  
 Precision i recall za obe klase mozemo procitati iz konfuzione matrice (treca kolona i treci red, respektivno) za sve slucajeve. Precision predstavlja odnos tacno pogodjenih instanci klase i netacno pogodjenih (instance drugih klasa koje su greskom deklarisane kao ta.). Recall predstavlja odnos tacno pogodjenih instanci klase i nepogodjenih (instance te klase koje su greskom deklarisane kao druge.)



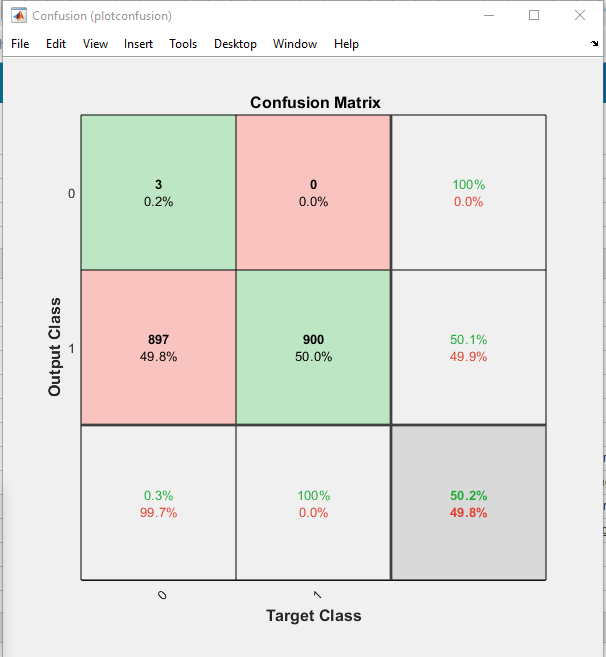


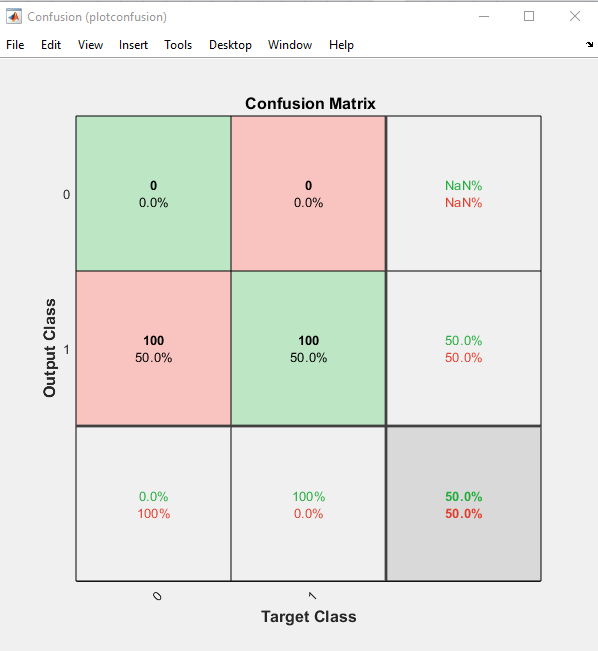


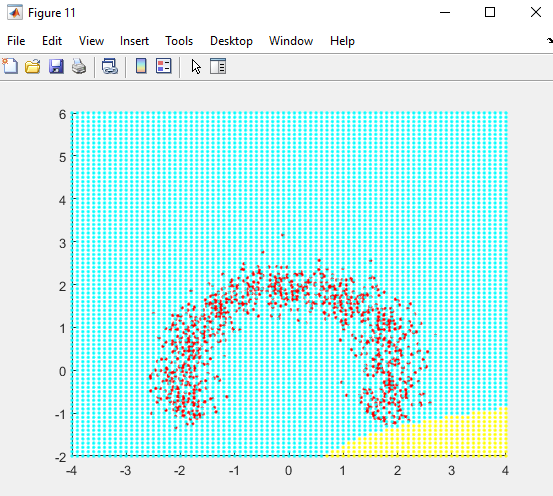


3.Grafici za najbolju arhitekturu [5,5]



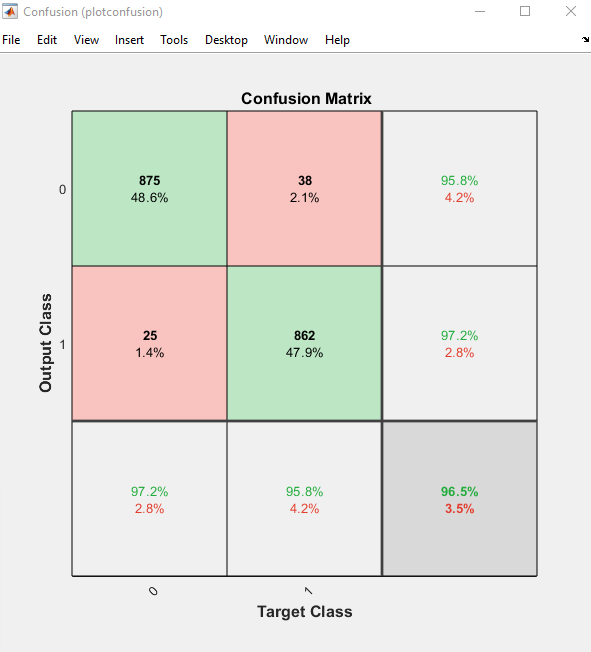


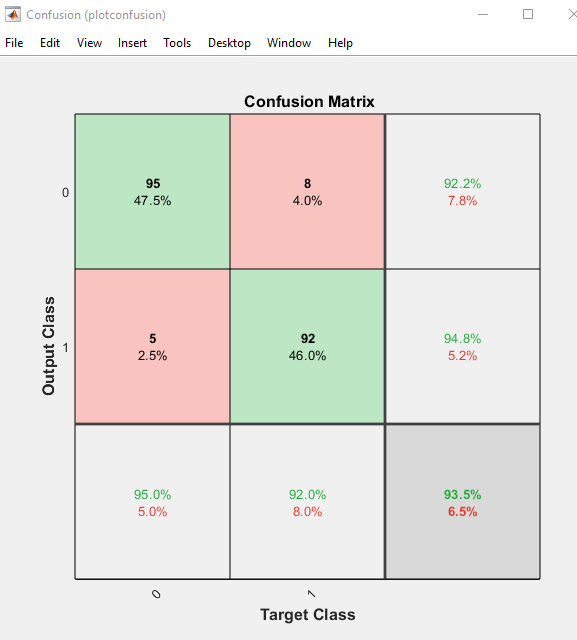


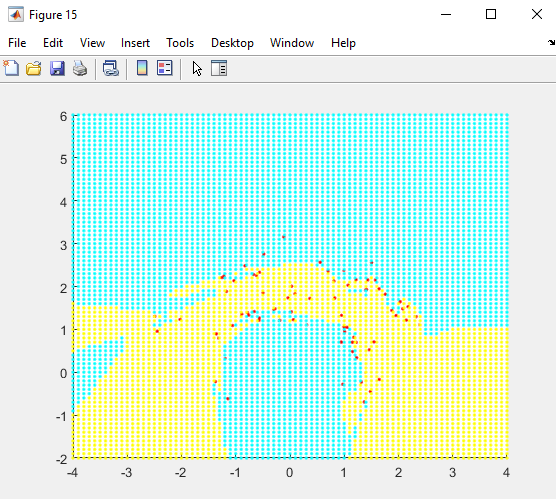


3.Grafici za underfitovanu arhitekturu [2]









3.Grafici za overfitovanu arhitekturu [30 30 30 30]

# Zadatak 2

Za drugi zadatak nam je po formuli dodeljen broj 3 (zadatak Connect-4).

## Opis problema

Kao ulazne podatke dobijamo bazu u kojoj su zapamćena stanja table za igru Connect-4, pri kojima igra nije završena, a idući potez nije iznuđen. Izlazne klase su mogući ishod za prvog igrača, pri čemu su nam mogućnosti *win, loss, i draw.* Izlaz je kodiran sa tri neurona, za svaku od klasa po jedan. Ulazi su stanja table – potezi prvog i drugog igrača, kao I prazna polja na tabli su kodirani sa (1,2,3), respektivno.. Izlazi su kodirani sa 100 za  *win,* 010 za *loss*, i 001 za *draw.*

Usvojili smo pretpostavku da nam je klasa od značaja *win*, odnosno pobeda za prvog igrača.

## Odabrani tip mreže

Pomoću *feedforward* mreže za klasifikaciju čiji su izlazi oblika (100,010,001) unakrsnom validacijom određujemo tačan oblik mreže, odnosno skriveni sloj i ostale hiperparametre.

Za aktivacione funkcije smo odabrali *tansig* i *logsig*, budući da su pogodne za diferenciranje.

## Odabir ulaznih podataka i test podataka

## Nakon što iz ulaznih podataka izdvojimo *X* i *Y*, podelimo ih na x\_win, x\_loss, i x\_draw. Potom iz svake od klasa odaberemo 10% ulaznih podataka kao test podatke. Biramo po 10% iz svake klase umesto iz celog uzorka kako bismo imali istu proporciju različitih klasa u test skupu kao što imamo i u početnom skupu. (ako bismo uzeli 10% nasumice iz početnog skupa može se desiti da uzmemo sve pripadnike jedne klase što nam ne odgovara za testiranje).

## Formiranje i obučavanje višeslojne neuralne mreže za klasifikaciju podatka.

Nakon što postavimo najbolje početne hiperparametre, sa njima pomoću *patternet*-a formiramo neuralnu mrežu za klasifikaciju.

## Implementacija metoda unakrsne validacije za pronalaženje najboljih hiperparametara modela

Budući da su podaci nebalansirani, svakoj klasi pridružujemo težinu koja je otprilike obrnuto proporcionalna njenoj zastupljenost u ulaznom skupu podataka.

Zbog pretpostavke da je klasa koja nas najviše interesuje *win*, računaćemo najbolje hiperparametre kao one koji pružaju najbolji *fscore* (kombinaciju dobre preciznosti I dobre osetjivosti u vidu objedinjene metrike fscore) za tu klasu.

Slede matrice konfuzije I kriva obučavanja za mrežu koja ima skriveni sloj [30, 30].

