**Неуралне мреже**

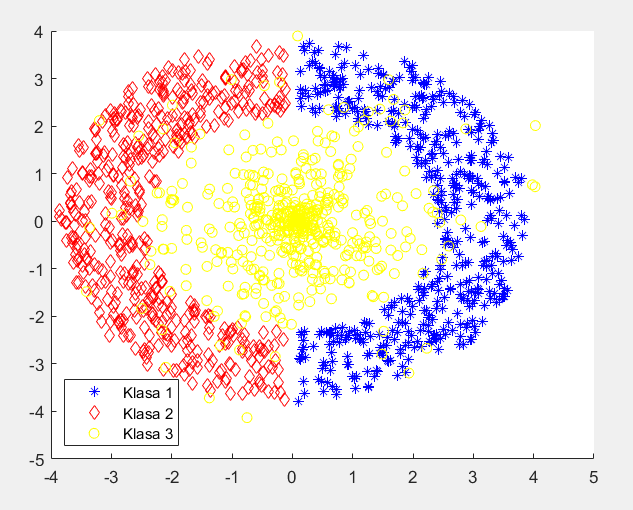
**Први пројектни задатак**

# **1. Решавање проблема класификације применом неуралне мреже**

На самом почетку израде нашег задатка учитавамо наше податке из табеле (наша табела је била dataset1) и смештамо у једну збирну матрицу која ће имати 3 врсте. У наставку задаткa ту матрицу делимо на две матрице, једну за улаз, а другу за излаз. Матрица улаза ће садржати прве две врсте збрине матрице, док ће излазна матрица садржати само трећу врсту збирне матрице. Обе матрице ће имати исти број врста као и збирна матрица.

Како се у излазној матрици налазе само 3 различите вредности, то значи да ми требамо да извршимо класификацију 3 различите класе. Тако да ћемо од улазне матрице направити 3 матрице класа (К1, К2, К3), где ће свака класна матрица задржати оне колоне улазне матрице где позиција у излазној матрици одговара тој класи.

На слици испод можемо да видимо расподелу по класама и како су њихови елементи распоређени по координатном систему. Плаве звездице нам означавају прву, црвени ромбоиди другу, жути кружићи трећу класу.



Након што смо извршили класификацију података, почињемо са тренирањем неуралне мреже. Наиме, улазне податке ћемо поделити на тренинг и тест скуп тј. на матрице за тренинг и тест. Тренинг скуп ће нам служити за обучавање мреже и на том скупу података ће мрежа учити и имаће 80% података улазног скупа, док ће нам тест скуп служити за проверу перформанси мреже и имаће 20% података улазног скупа. Ову поделу вршимо да бисмо установили валидност наше мреже, тј. да ли је дошло до преобучавања или недовољног обучавања. Ове матрице смо направили тако да буду све класе приближно једнако заступљене у њима (насумично одабрани индекси).

Започињемо са прављењем тражених мрежа. Узимамо исте параметре за све мреже:

**број епоха – 1500**

**максимална дозвољена грешка – 10-4**

**минимални дозвољени градијент – 10-5**

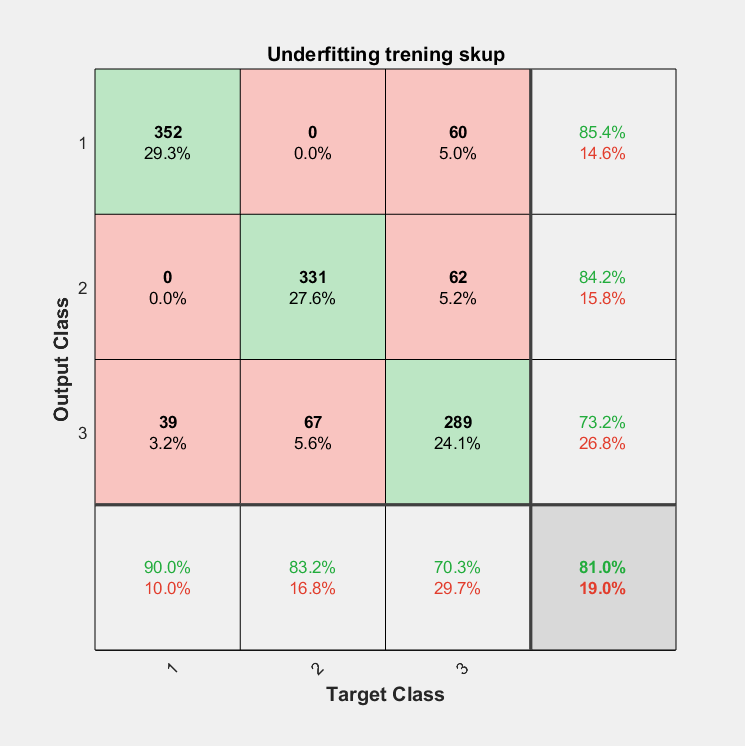
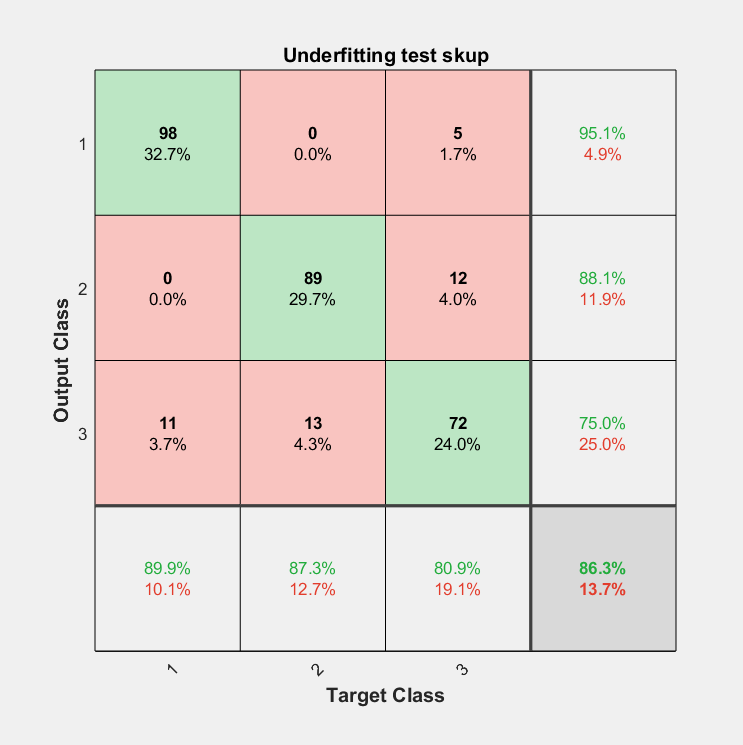
**активациона функција скривеног слоја – tansig**

**активациона функција излазног слоја - softmax**

Архитектуре:

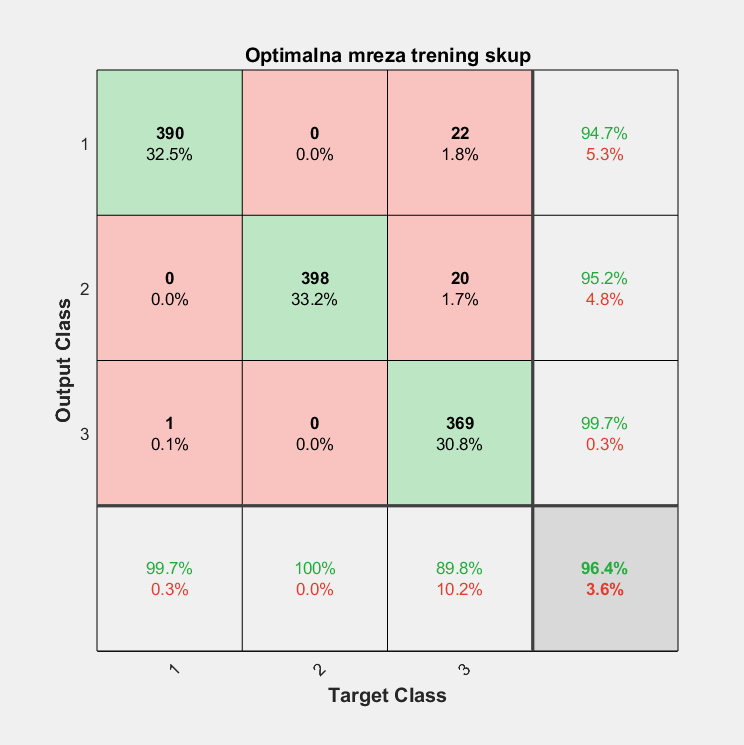
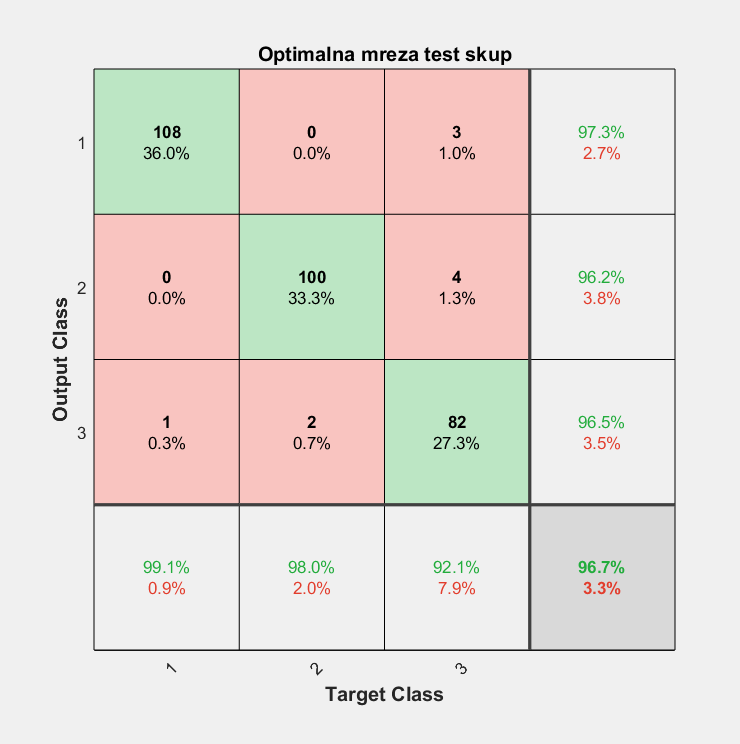
**1. UNDERFITTING**

Ову архитектуру морамо направити тако да нема добру класификацију података. То ћемо постићи тако што ћемо направити мрежу са малим бројем неурона тако да има **2** скривена слоја са по **2** и 1 неурона, респективно. На слици испод видимо да наша мрежа нема баш добар проценат успешности, тако да смо добро измоделоваили ову мрежу. Примећујемо и да се за тренинг и тест скуп проценти успешности знатно разликују.



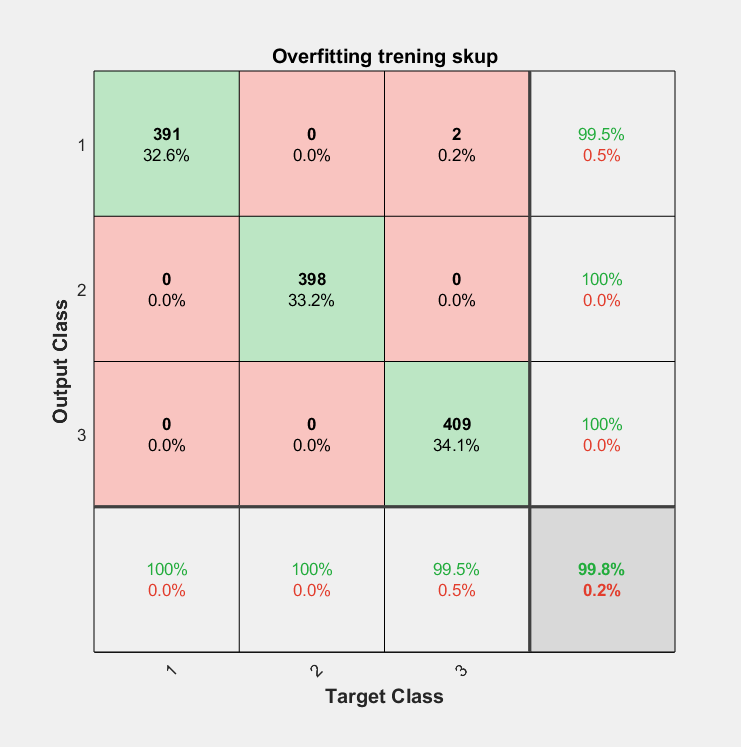
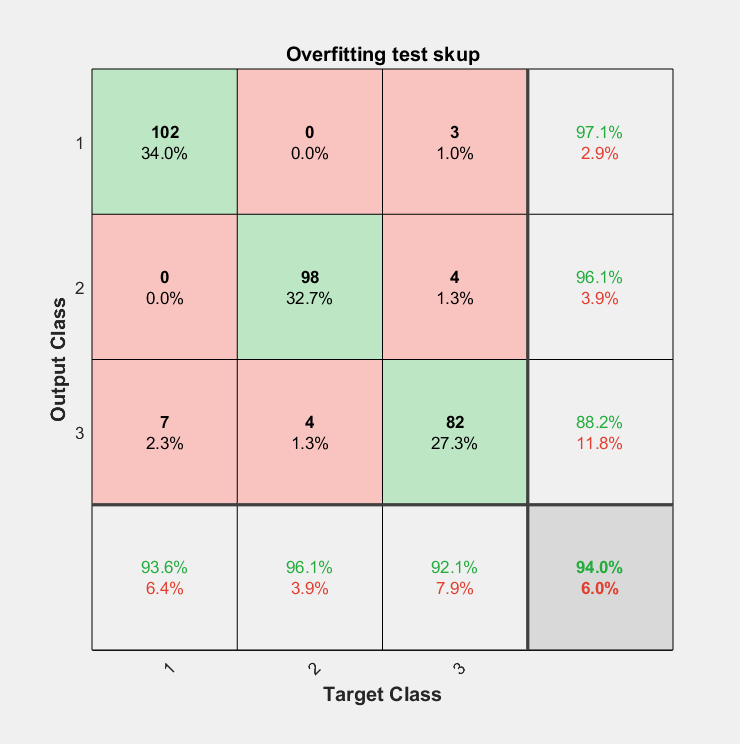
**2. ОПТИМАЛНА АРХИТЕКТУРА**

Ову архитектуру ћемо направити тако да има најбоље могуће перформансе тј. да вредност тест скупа буде скоро подједако добра као вредност тренинг скупа. То ћемо постићи тако што ћемо направити мрежу са оптималним бројем неурона тако да има **3** скривена слоја са по **4, 4, 3** неурона, респективно. На слици испод видимо да наша мрежа има добар проценат успешности и за тренинг и за тест скуп**)**, тако да смо добро измоделоваили ову мрежу.



**3. OVERFITTING**

Ову архитектуру ћемо направити тако да има много добре резултате на тренинг скупу, али лошије на скупу података за тест. То ћемо постићи тако што ћемо направити мрежу са великим бројем неурона тако да има **3** скривена слоја са по **30, 30, 20** неурона, респективно. На слици испод видимо да наша мрежа има одличан проценат успешности за тренинг, али прилично лошији за тест скуп.



**Закључак:**

На основу добијених резултата видимо да оптимална мрежа даје најбоље могуће карактеристике (као што смо и очекивали). Мрежа даје добре генералне резултате, за разлику од прве архитектуре и њен проценат успешности не зависи од скупа податакa на ком се обучавала тј. за сваки скуп даје приближно исте перформансе за разлику од overﬁtting мреже која даје добре резултате само на скупу на ком се обучавала, док за неки нови скуп нема толико добре перформансе. Можемо погледати и дијаграме одлучивања за горе наведене архитектуре и видимо како најправилније контуре разграничења има друга архитектура. Редом су дати примери за три врсте архитектура које су обрађене.

