

УНИВЕРЗИТЕТ У БЕОГРАДУ
ЕЛЕКТРОТЕХНИЧКИ ФАКУЛТЕТ



НЕУРАЛНЕ МРЕЖЕ

Први пројектни задатак

Ментор:

Др Горан Квашчев
Марија Новичић

Кандидати:

Петар Перин 0663/17
Никола Ристић 0661/17

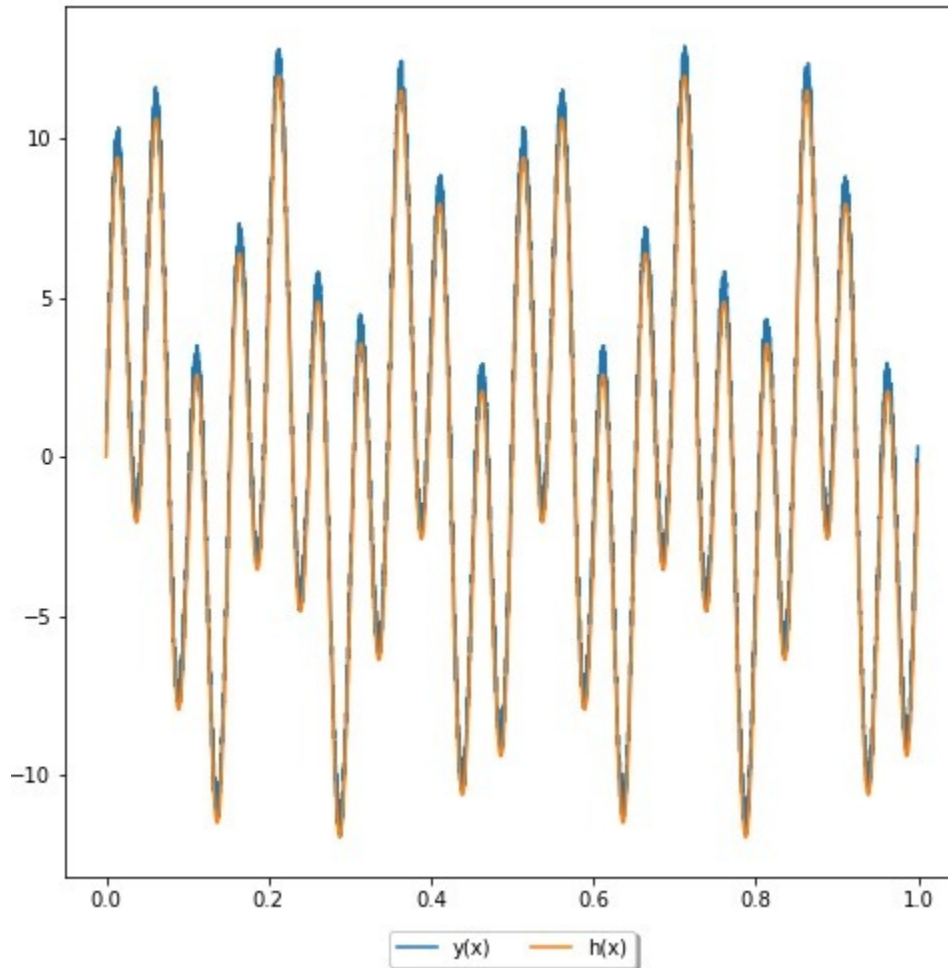
ПАРАМЕТРИ: $A = 7$, $B = 5$, $f_1 = 20$, $f_2 = 6$, $P = 2$, $Q = 5$

Zadatak 1

Kreiranje neuralne mreže za predikciju koja služi za predikciju zadate funkcije $h(x)$

Funkcija $h(x)$: $h(x) = A \cdot \sin(2 \cdot f_1 x) + B \cdot \sin(2 \cdot f_2 x)$

Funkcija $z(x)$: $y(x) = h(x) + s(x)$, $s(x)$ predstavlja slučajni šum standardne devijacije $\text{std} = 0.2 \cdot \min(A, B)$

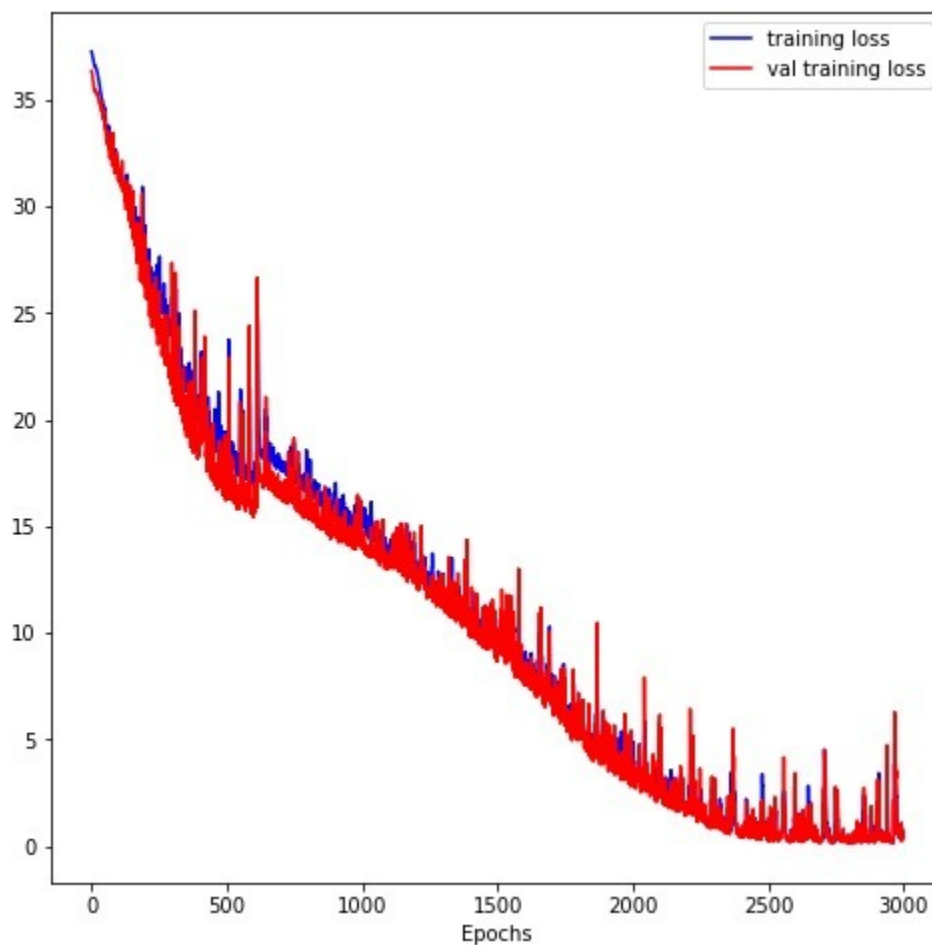


Struktura neuralne mreže koja vrši predikciju funkcije $h(x)$:

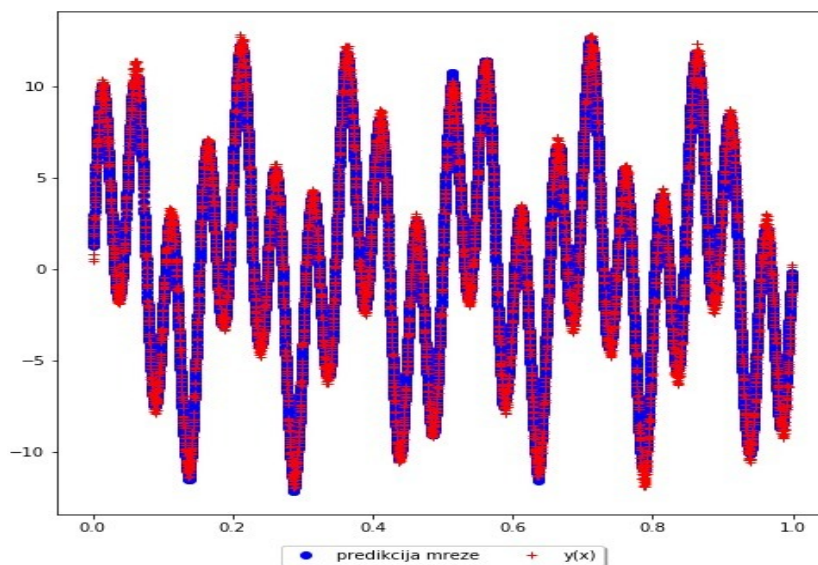
```
model = keras.Sequential()
model.add(keras.layers.Dense(32, input_shape = [1], activation = "relu"))
model.add(keras.layers.Dense(35, activation = "tanh"))
model.add(keras.layers.Dense(35, activation = "relu"))
model.add(keras.layers.Dense(45, activation = "tanh"))
model.add(keras.layers.Dense(45, activation = "relu"))
model.add(keras.layers.Dense(35, activation = "tanh"))
model.add(keras.layers.Dense(35, activation = "relu"))
model.add(keras.layers.Dense(20, activation = "tanh"))
model.add(keras.layers.Dense(32, activation = "relu"))

model.add(keras.layers.Dense(1))
opt = Adam(lr = 0.00107)
model.compile(optimizer = opt, loss='mse', metrics=['mae'])
history = model.fit(x_train, y_train, epochs = 3000, batch_size=900, validation_split = 0.15, verbose = 2)
```

Kriva performanse u zavisnosti od epohe treniranja je data na sledećem grafiku:

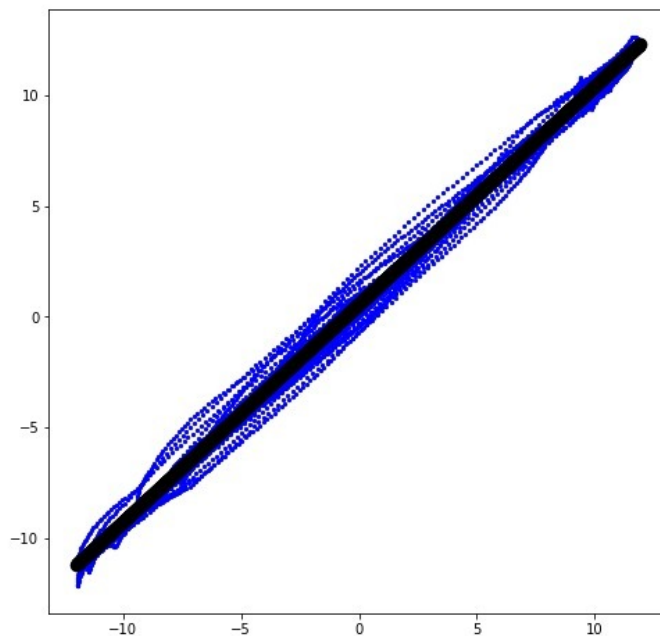


Predikcija neuralne mreže:



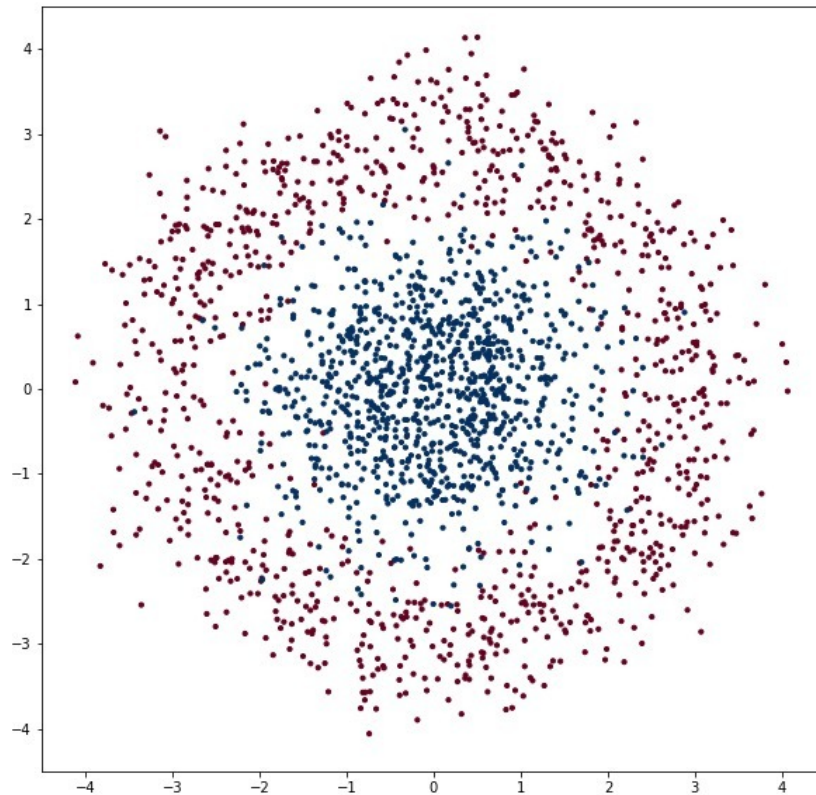
Neuralna mreža je je veoma dobro obučena. R^2 regresivne krive je 0.99. Za train podatke kao i za test podatke. Ponovna treniranja neuralne mreže nisu davale iste rezultate, ali R^2 se nalazio između 0,84 i 0,9. To svojstvo može se prisvojiti slučajnoj inicijalizaciji pojačanja. Opisani rezultat se može smatrati za “Best-case scenario”

Regresivna kriva:



Zadatak 2

Ulazni skup podataka raspoređen po klasama: klasa1- plavo, klasa0-crveno



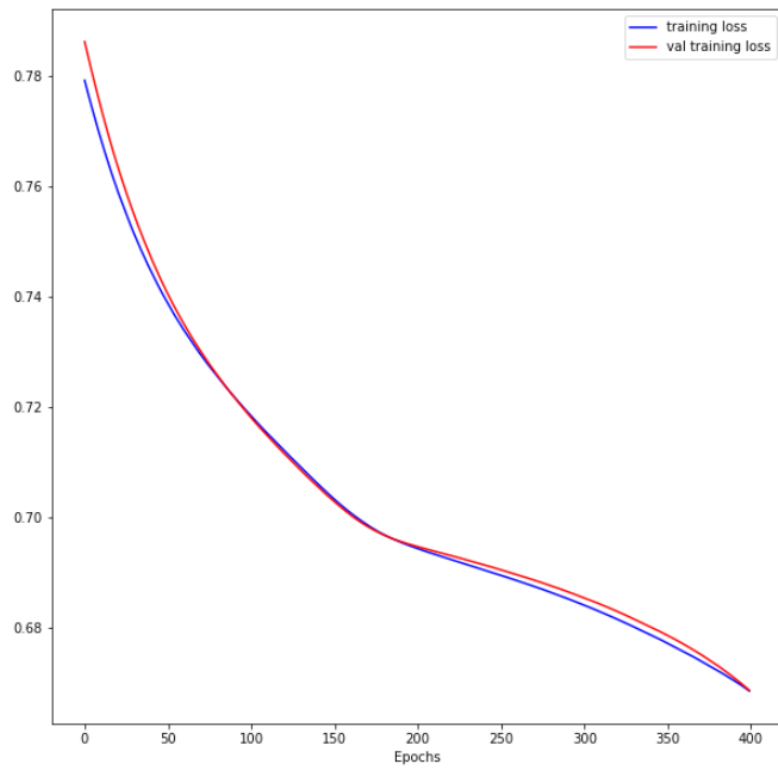
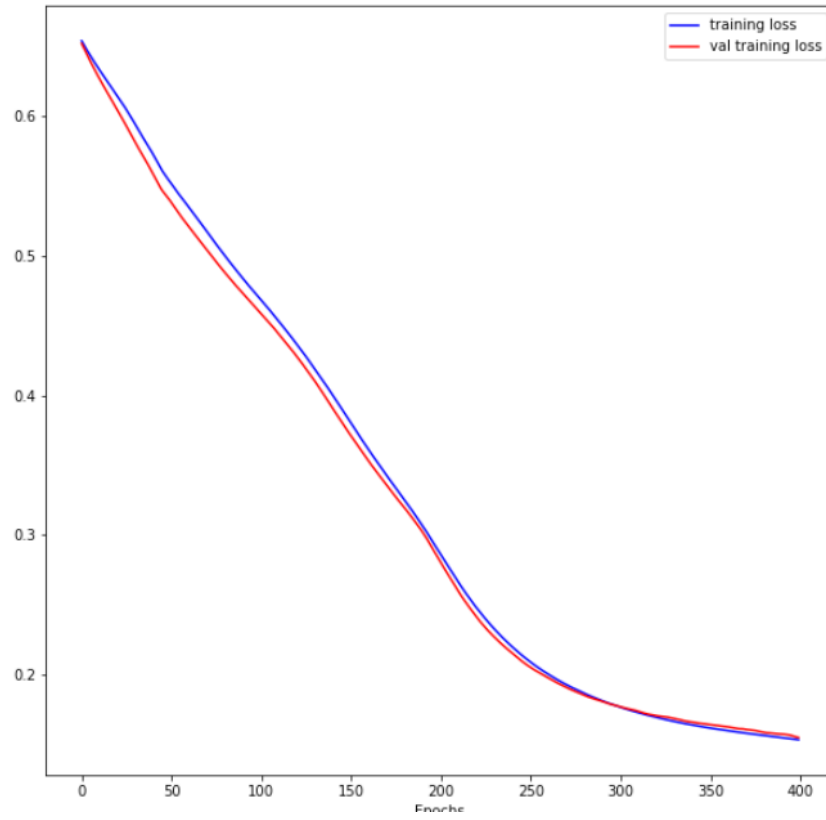
Podela je izvršena na sledeći način:

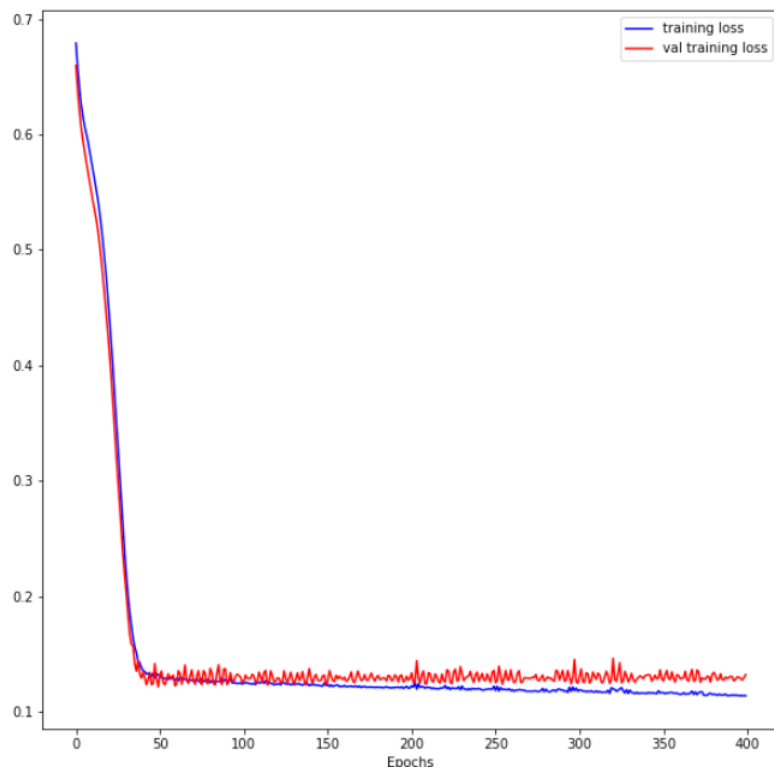
```
k_train, x_test, y_train, y_test = train_test_split(data, label, test_size = 0.2)
```

Veličina test skupa je 20%, dok je veličina trening skupa 80%, validation je 15% od trening seta.

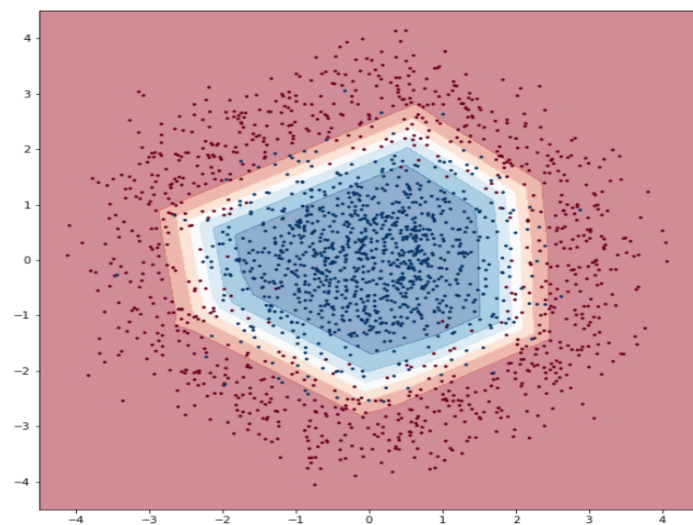
Podela je neophodna da bi se izmerile realne performanse mreže.

Rezultati treninga u zavisnosti od epohe standardne, underfitting i overfitting mreže, date su u nastavku, respektivno.

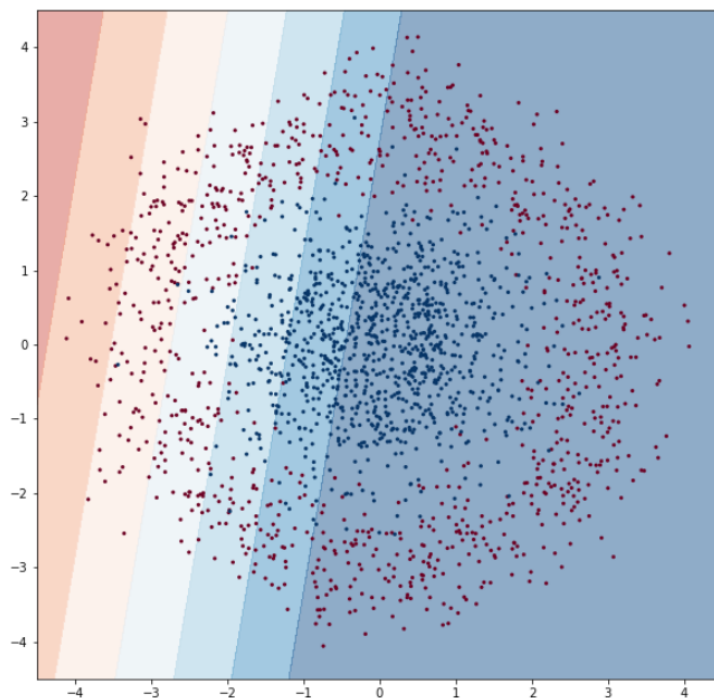




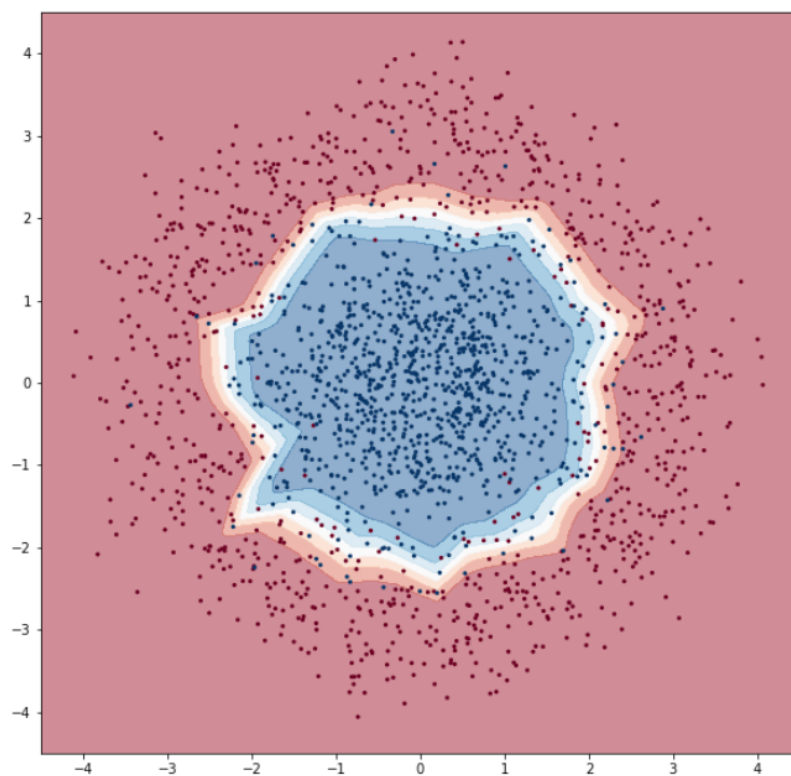
Granice odlučivanja standardne, underfitting i overfitting mreže, date su u nastavku, respektivno.



Standarda mreža ima procenat pogodaka oko 93%



Underfitting mreža je pokazala performanse od 55% tačnosti.



Overfitting ima 95% tačnosti, ali loše generalizuje. Mreža je preobučena za dati skup podataka.

Konfuzione matrice:

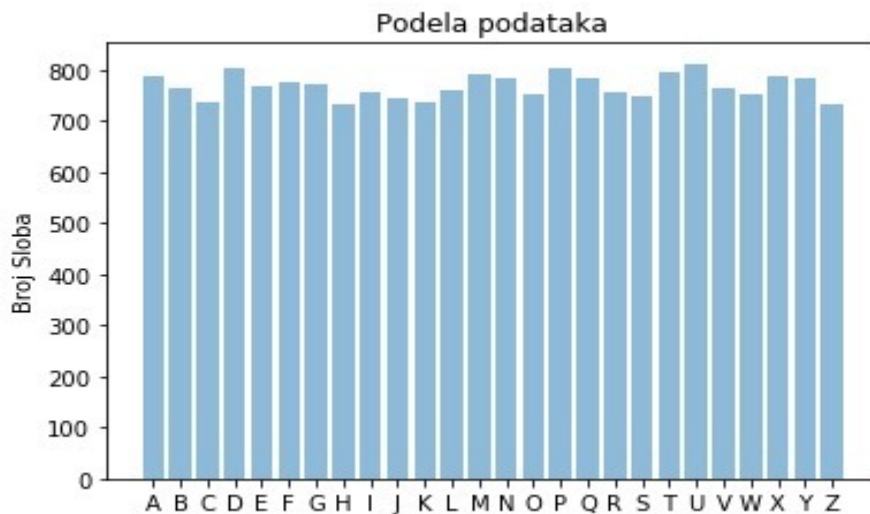
```
[[744  42]
 [ 52 762]] preciznost(93.61%) recall(94.78%)   za standradni.

[[309  477]
 [ 162 652]] preciznost(80.1%) recall(57.75%)   za underfiting.

[[746  40]
 [ 36 778]] preciznost(95.58%) recall(95.51%)   za overfiting.
```

Zadatak 3

Problem koji se rešava je prepoznavanje slova na osnovu zadatih parametara.
Određuju se hiperparametri mreže koja rešava ovaj problem.



Ulazni skup podataka

Sa grafika se može zaključiti da su svi podaci dobro balansirani.

Vrednosti hiperparametara koje su uzete u obzir su: struktura mreža, početna pojačanja, aktivacione funkcije, regularizacija.

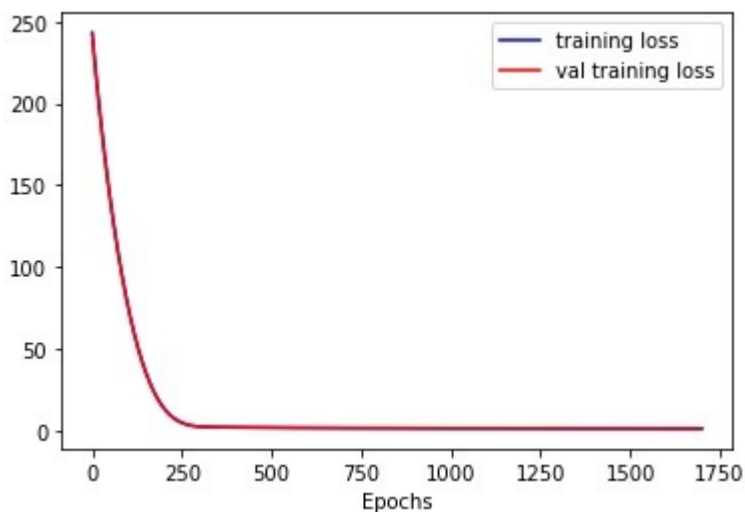
```
struktura = np.array([[10,15] , [20,50,10]])  
regularizacija = np.array([0.1 , 0.25, 0.5, 0.7])  
tezine = np.array([2, 3, 4])  
fje = np.array(['tanh' , 'relu' , 'sigmoid'])
```

Nakon isprobavanja svih mogućih kombinacija ovih hiperparamtera izabrana je mreža sa sledećim parametrima:

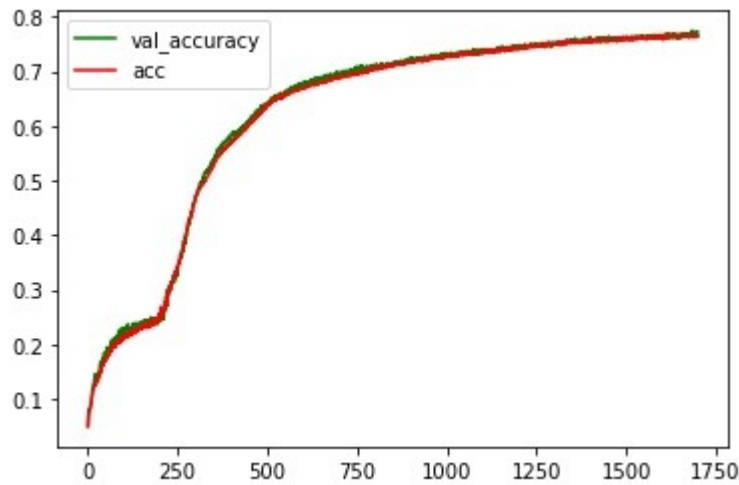
Broj Slojeva	Regularizacija	Težine	Funkcija Aktivacije
[10,15]	0.1	4	tanh

Upoređivanjem preciznosti mreže za test skup podataka izabrana je gore navedena struktura neuralne mreže (u tabeli).

Treniranje izabrane mreže po epohama:



Performanse mreže po epohama:



Za ulazni test skup mreža je imala preciznost od 76,3%

KONFUZIONA MATRICA ZA TRAIN:

[560	0	0	1	0	0	0	5	0	13	7	2	4	2	3	1	9	0	9	0	3	2	3	9	6	0]
[0	480	0	11	2	4	9	8	1	0	0	0	0	0	4	19	1	45	9	0	0	4	0	6	3	0]
[0	0	434	0	30	7	35	1	0	0	32	0	2	0	14	0	3	0	8	2	14	2	0	0	1	0]
[1	25	0	509	0	1	0	11	0	6	0	1	8	9	14	16	0	19	4	0	2	0	0	12	0	0]
[0	13	3	0	470	2	22	2	0	0	12	0	0	0	0	0	24	4	27	10	0	0	0	19	0	6]
[0	11	10	3	10	430	13	4	7	1	0	0	0	7	0	19	0	3	8	42	0	0	2	9	13	0]
[3	10	49	10	1	1	402	8	0	0	6	3	1	0	5	10	60	3	9	0	9	19	3	3	1	0]
[4	20	2	29	0	6	2	298	0	1	15	0	8	20	62	10	8	43	0	5	15	19	0	11	2	0]
[0	11	1	6	3	10	0	1	491	13	0	5	0	0	3	2	3	0	32	0	0	0	0	10	9	2]
[6	0	1	10	4	11	0	9	35	486	0	0	0	1	3	5	2	0	25	0	0	0	0	7	0	0]
[0	6	21	4	8	0	6	12	0	0	426	0	2	7	0	0	4	47	0	1	19	2	2	29	0	0]
[1	5	0	8	17	0	26	0	0	0	3	467	4	0	3	0	24	2	7	0	0	0	0	35	2	0]
[7	3	0	1	0	0	2	6	0	0	0	0	0	574	2	4	1	0	7	0	0	4	0	24	0	0]
[0	2	0	13	0	0	0	24	0	0	5	0	18	506	13	3	0	1	0	2	19	1	19	1	1	0]
[12	3	2	20	0	0	5	13	0	0	0	0	1	0	491	12	5	5	0	0	8	4	30	0	0	0]
[0	4	0	5	1	46	15	1	0	0	1	0	0	0	8	522	6	1	0	0	1	9	4	0	21	0]
[26	11	0	4	12	0	23	1	0	0	0	7	1	0	57	0	441	2	34	0	1	8	1	0	8	4]
[0	36	0	23	0	0	5	29	0	0	35	0	4	1	13	1	1	444	0	0	1	2	1	24	0	0]
[1	52	0	5	17	11	1	0	16	6	0	31	0	0	2	5	29	6	334	7	0	1	0	11	14	71]
[0	5	0	1	12	8	22	17	4	1	11	0	0	0	0	1	0	0	12	472	1	11	0	8	42	9]
[2	1	1	1	0	0	0	4	0	0	1	0	13	18	19	0	0	0	0	0	544	0	17	0	1	0]
[0	9	0	0	0	0	5	6	0	0	0	0	1	1	6	4	0	0	0	0	2	561	16	1	14	0]
[0	4	0	0	0	0	1	6	0	0	0	0	8	1	3	1	0	3	0	0	12	4	557	0	0	0]
[0	2	0	10	4	2	0	1	8	6	16	1	0	0	2	0	71	0	7	20	5	0	0	464	8	3]
[1	0	0	0	0	9	0	2	0	0	0	0	0	0	0	8	14	0	4	54	6	50	1	3	488	0]
[4	0	0	0	34	2	3	0	0	36	0	0	0	0	0	1	1	0	75	4	0	0	0	5	0	403]]

Precision: 87,6% Recall: 89,6% za izabranu klasu.