

SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU ELEKTROTEHNIČKI  
FAKULTET

Diplomski studij računarstva

Laboratorijska vježba 5

Klasificiranje vrste cvijeta irisa pomoću neuronskih mreža

Ivan Budoš, DRB

Osijek, 2022.

# SADRŽAJ

UVOD .....	3
NEURONSKE MREŽE .....	4
Programski kod .....	7
REZULTATI.....	8
Aktivacijska funkcija skrivenog sloja – 'identity' .....	8
Aktivacijska funkcija skrivenog sloja – 'logistic' .....	9
Aktivacijska funkcija skrivenog sloja – 'relu' .....	11
Aktivacijska funkcija skrivenog sloja – 'tanh' .....	13
ZAKLJUČAK .....	15

## UVOD

U petoj laboratorijskoj vježbi potrebno je projektirati i ispitati neuronsku mrežu koja određuje vrstu cvijeta irisa (perunika) na osnovu njihovih mjerenih karakteristika. Korištena baza je sadržava mjerenja fizičkih parametara latica i čašićnih listića tri vrste cvjetova irisa: setosa, virginica, versicolor. Prilikom rješavanja parametre zadatka je potrebno mijenjati na sljedeće vrijednosti:

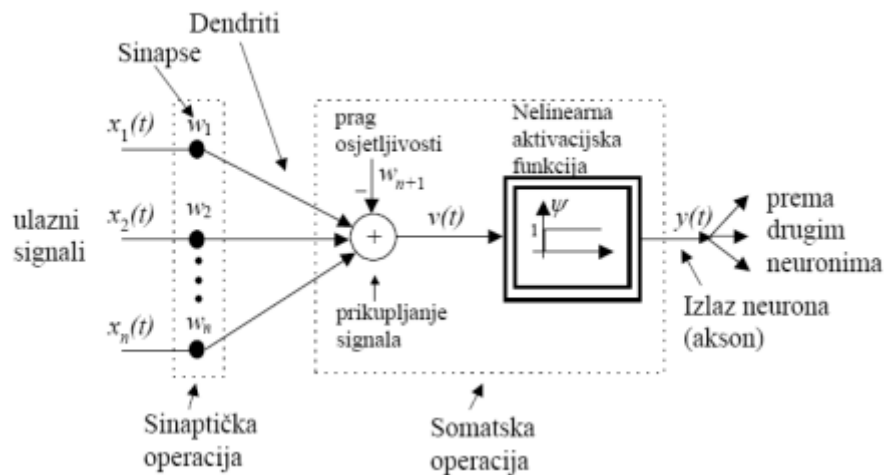
- Broj neurona po skrivenom sloju: 5, 10, 30.
- Broj skrivenih slojeva: 1, 2, 3.
- Aktivacijska funkcija skrivenog sloja: 'identity', 'logistic', 'tanh', 'relu'.
- Algoritam učenja: 'lbfgs', 'sgd', 'adam'.

Potrebno je proučiti kako koji parametar utječe na mrežu i na dobivene rezultate.

# NEURONSKE MREŽE

Neuronske mreže su sustavi za procesuiranje informacija koji su inspirirani biološkim živčanim sustavom kao što je mozak. Sastoji se od uvjetno rečeno velikog broja međusobno povezanih procesnih elemenata tzv. neurona. Svaki neuron je u biti sumirajući element povezan sa aktivacijskom funkcijom. Biološki neuroni primaju elektrokemijske pobude preko dendrita, a ako je težinska suma svih ulaza na dendritima veća od praga, neuron okida i šalje impuls na aksion preko kojeg je spojen s ulazima ostalih neurona.

Prvi model neurona koji je bio osmišljen još 1943. od strane McCullocha i Pittsa se zvao „perceptron“ i njegova jedina razlika od modernijih modela neurona je bila ta što je kao prijenosnu funkciju koristio diskontinuiranu step funkciju. Dok su se kasnije puno logičnije počele koristiti kontinuirane funkcije, bile one linearne ili sigmoidalne (logsig, tansig). Jer u slučaju kontinuirane funkcije imamo mogućnosti puno preciznije aproksimacije modela problema. Ali je najveća revolucija započela uvođenjem višeslojnih mreža i njihovih algoritama za učenje. Shematski prikaz perceptrona je vidljiv na slici 1.

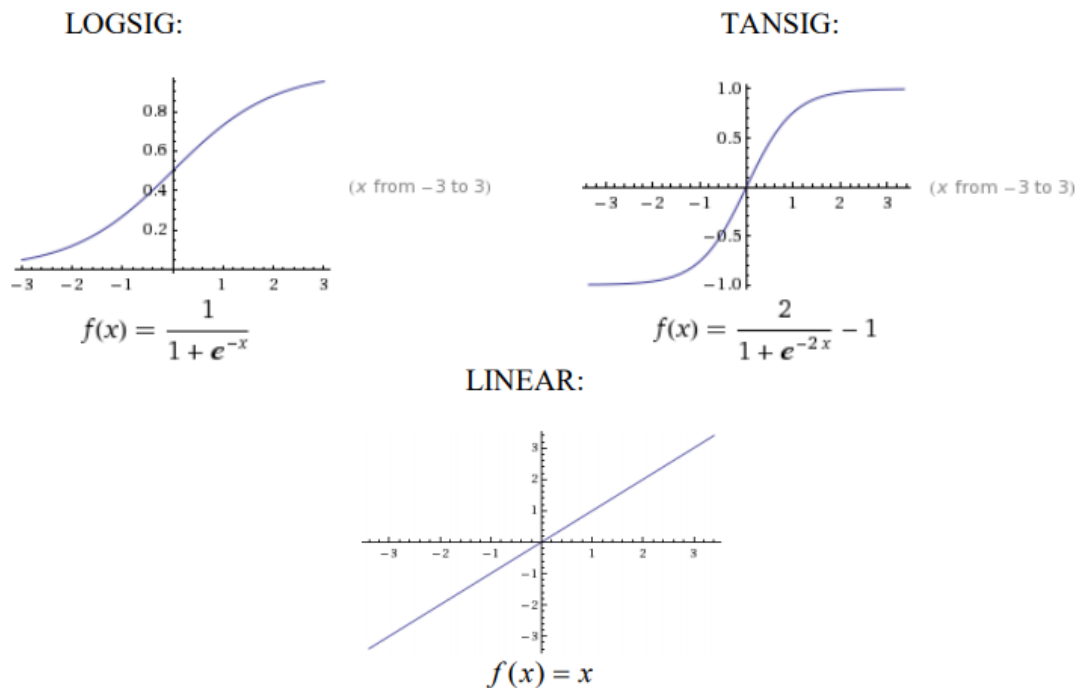


Slika 1. Shematski prikaz perceptrona

$$v(t) = \sum_{i=1}^n w_i(t) * x_i(t) - w_{n+1}, y(t) = \psi(v), y(t) = w^T(t) * x(t)$$

Kao što se vidi na slici 1, svaki neuron prikuplja signale od prethodnog sloja (pomnožene sa težinama), te uz dodatak praga osjetljivosti dolazi do prijenosne funkcije odnosno nelinearne aktivacijske funkcije. Izlaz iz te funkcije potom odlazi do svakog neurona u idućem sloju gdje se proces ponavlja. Neuroni se najčešće dijele na statičke i dinamičke, gdje statički neuroni ovise isključivo o trenutnim vrijednostima signala i težina, dok kod dinamičkih postoje

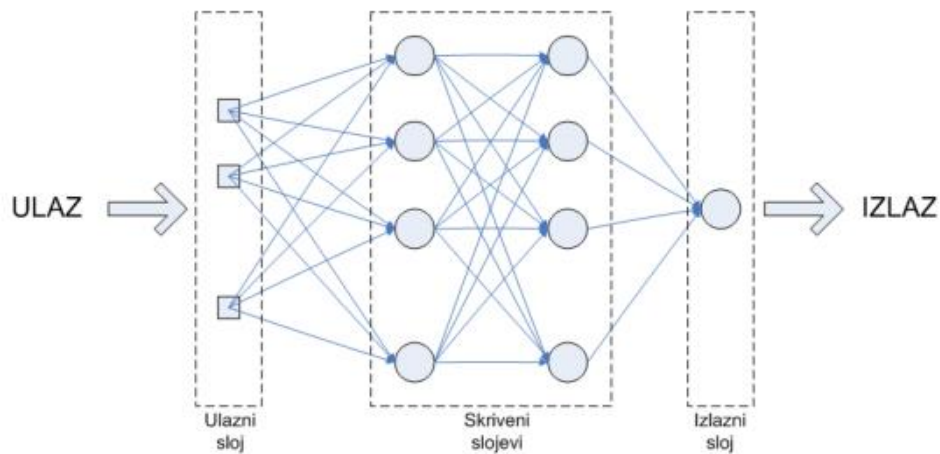
određene povratne veze i promjenjive aktivacijske funkcije. Kao što je ranije rečeno aktivacijske funkcije najčešće su sigmoide kao što vidimo na slijedećim slikama.



Slika 2. Aktivacijske funkcije

Da bi se neuronska mreža definirala, pored osnovnih parametara koji opisuju oblik i tip mreže, odnosno arhitekturu, potrebno je odrediti i algoritam učenja. Proces učenja je u biti proces optimizacije pomoću algoritma gdje se pronalaze težine između neurona koje najbolje opisuju rješenje odnosno aproksimaciju problema. Proces učenja najčešće uključuje slijedeće korake:

- Dovođenje na ulaz neuronske mreže niz slučajeva (uzoraka) koje želimo naučiti raspoznavati
- Odrediti pogrešku između dobivenog izlaza i željenog izlaza
- Promijeniti težine da bi se izlaz bolje aproksimirao



Slika 3. Višeslojna neuronska mreža

Na slici 3 vidimo primjer višeslojne neuronske mreže sa ulaznim i izlaznim slojem, te dva skrivena sloja. Iako broj neurona nije ograničen do sada se je u praktičnim primjenama koristilo do najviše par stotina neurona i to u jako kompleksnim primjenama. Jer uvođenjem dodatnih neurona (i slojeva) višestruko povećava problem učenja, a može se dogoditi da u slučaju pretjeranog broja neurona (za pojedini problem) može unijeti smetnje u izlaz neuronske mreže zbog neskladnog rada. Imamo 3 osnovna tipa učenja neuronskih mreža:

**Nadzirano učenje** – učenje na temelju poznatih uzoraka i rezultata

**Učenje pojačavanjem** – uključuje povratnu vezu iz okoline

**Nenadzirano učenje** – učenje iz pravilnosti ulaznih podataka

Najčešće se koristi nadzirano učenje, a najčešće korišteni algoritam učenja je sa povratnom propagacijom pogreške (eng. backpropagation). Neuronske mreže zbog svoje sposobnosti učenja i aproksimacije se najčešće koristi za slijedeće primjene:

- Raspoznavanje znakova teksta (i analiza slika)
- Prepoznavanje govora
- Adaptivno uklanjanje šuma
- Predviđanje cijena dionica (financije)
- Medicinska dijagnostika

## Programski kod

Uz priloženi kod potrebno je implementirati kreiranje neuronske mreže pomoću klase *MLPRegressor*. Postupak je isti kao i u laboratorijskoj vježbi četiri. Kreirana su dva polja: *activationKey* i *learningAlgorithm* koja sadržavaju moguće aktivacije funkcija skrivenog sloja i algoritme učenja. Oni se predaju konstruktoru ovisno o odabranom indeksu. Kreiranje neuronske mreže prikazano je na slici 4.

```
mlp = MLPRegressor(hidden_layer_sizes=N_PER_LAYER,
                    activation=activationKey[ACTIVATION_F], solver=learningAlgorithm[SOLVER],
                    alpha=ALPHA, max_iter=MAX_ITER)
```

Slika 4. Kreiranje neuronske mreže

Parametri koje prima konstruktor su sljedeći:

*hidden\_layer\_sizes* – broj neurona u skrivenim slojevima neuronske mreže

*activation* – aktivacijska funkcija neurona koja može biti jedna od sljedećih:

- „identity“ -  $f(x) = x$ .
- „logistic“ -  $f(x) = 1 / (1 + \exp(-x))$ .
- „tanh“ -  $f(x) = \tanh(x)$ .
- „relu“ -  $f(x) = \max(0, x)$ .

*solver* – metoda za optimizaciju težina neuronske mreže koja može biti jedna od navedenih:

- - ‘lbfgs’, metoda iz porodice kvazi-Newton metoda.
- - ‘sgd’, engl. Stochastic Gradient Descent.
- - ‘adam’ novija metoda 'sgd'-a

- *alpha* – L2 regularizacijski parametar

- *max\_iter* – maksimalni broj iteracija učenja neuronske mreže

Parametri koji se mijenjaju su dodani u kod kao globalne varijable te se pomoću *for* petlji izmjenjuju. Na taj način je cijeli proces automatiziran.

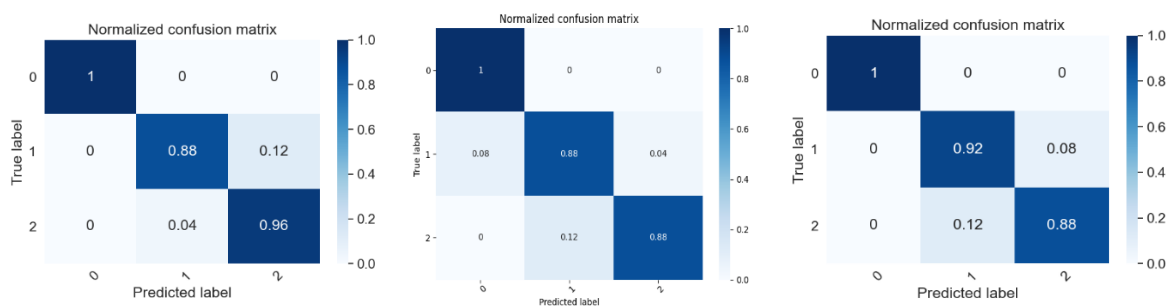
## REZULTATI

Parametri zadatka mijenjani su na sljedeće vrijednosti:

- Broj neurona po skrivenom sloju : 5, 10, 30
- Broj skrivenih slojeva : 1, 2, 3
- Aktivacijska funkcija skrivenog sloja : 'identity', 'logistic', 'tahn', 'relu'
- Algoritam učenja : 'lbfgs', 'sgd', 'adam'

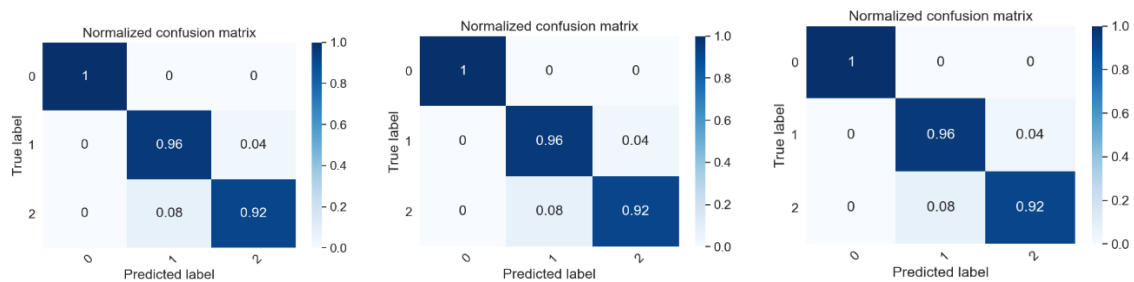
### Aktivacijska funkcija skrivenog sloja – 'identity'

Broj skrivenih slojeva	1	1	1
Broj neurona po skrivenom sloju	5	5	5
Algoritam učenja	adam	lbfgs	sgd
Točnost	0.947	0.92	0.93

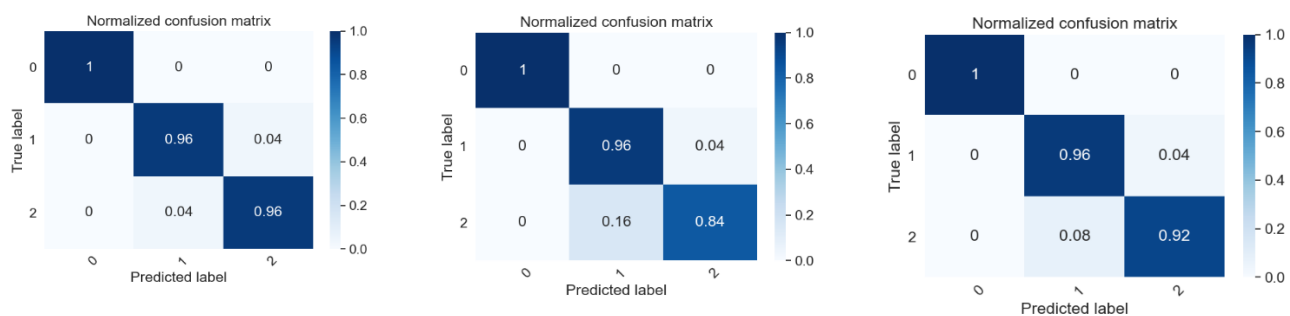


Broj skrivenih slojeva	2	2	2
Broj neurona po skrivenom sloju	30, 5	30, 5	30, 5
Algoritam učenja	adam	lbfgs	sgd
Točnost	0.96	0.96	0.96



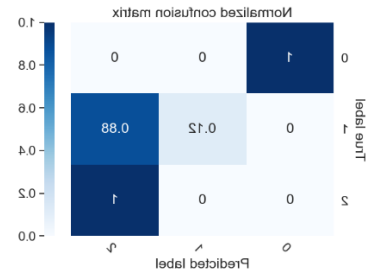
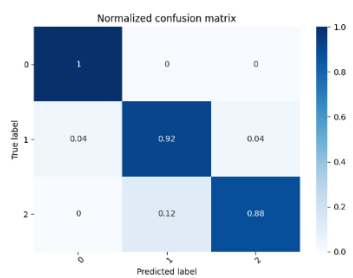
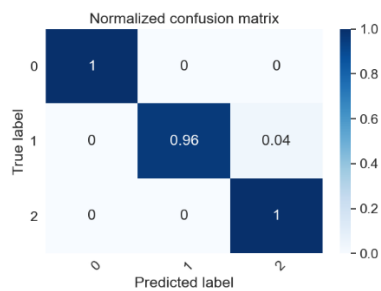


Broj skrivenih slojeva	3	3	3
Broj neurona po skrivenom sloju	5,10,5	5,10,5	5,10,5
Algoritam učenja	adam	lbfgs	sgd
Točnost	0.973	0.93	0.96

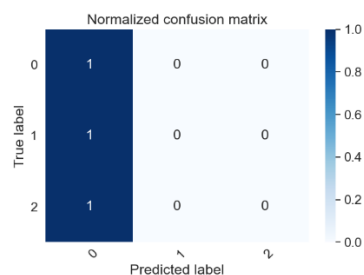
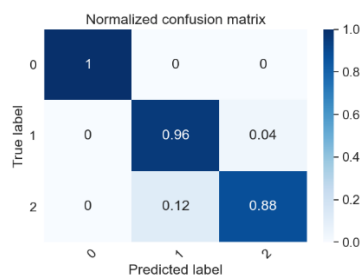
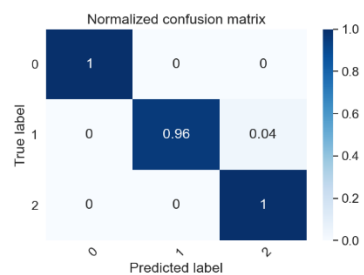


### Aktivacijska funkcija skrivenog sloja – 'logistic'

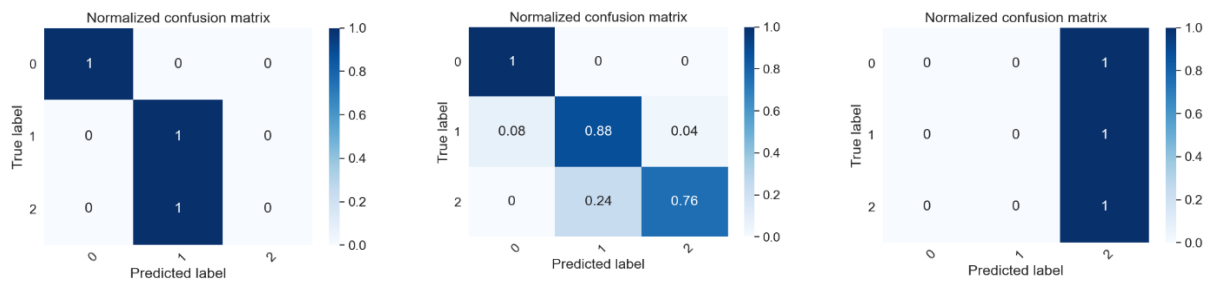
Broj skrivenih slojeva	1	1	1
Broj neurona po skrivenom sloju	5	5	5
Algoritam učenja	adam	lbfgs	sgd
Točnost	0.987	0.93	0.706



Broj skrivenih slojeva	2	2	2
Broj neurona po skrivenom sloju	30, 5	30, 5	30, 5
Algoritam učenja	adam	lbfgs	sgd
Točnost	0.986	0.946	0.33

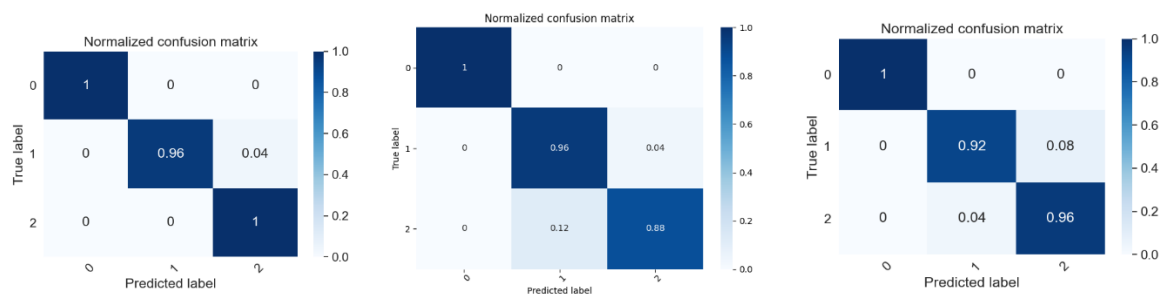


Broj skrivenih slojeva	3	3	3
Broj neurona po skrivenom sloju	5,10,5	5,10,5	5,10,5
Algoritam učenja	adam	lbfgs	sgd
Točnost	0.66	0.88	0.33

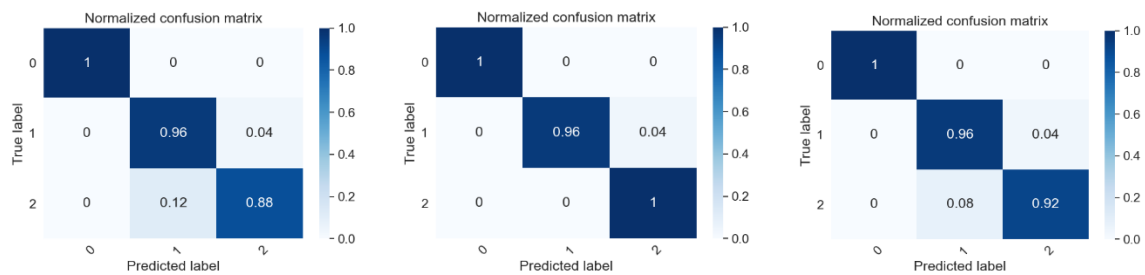


## Aktivacijska funkcija skrivenog sloja – 'relu'

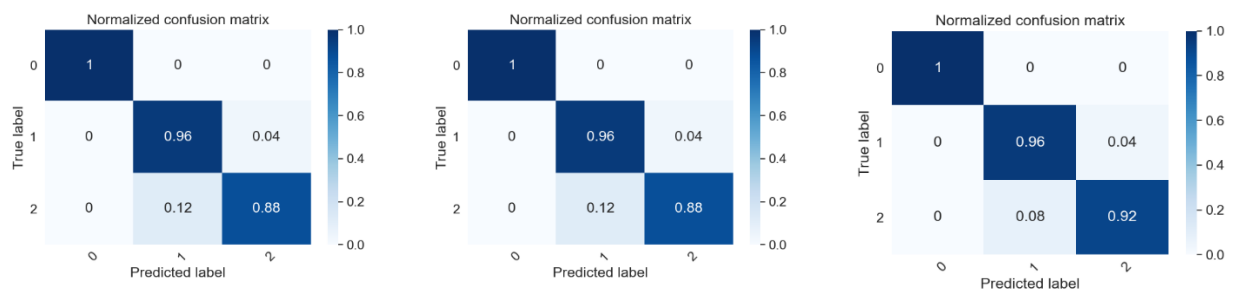
Broj skrivenih slojeva	1	1	1
Broj neurona po skrivenom sloju	5	5	5
Algoritam učenja	adam	lbfgs	sgd
Točnost	0.987	0.947	0.96



Broj skrivenih slojeva	2	2	2
Broj neurona po skrivenom sloju	30, 5	30, 5	30, 5
Algoritam učenja	adam	lbfgs	sgd
Točnost	0.947	0.987	0.96

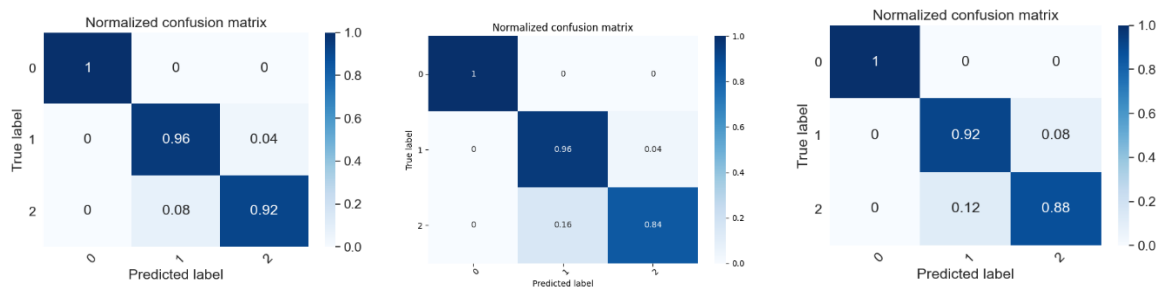


Broj skrivenih slojeva	3	3	3
Broj neurona po skrivenom sloju	5,10,5	5,10,5	5,10,5
Algoritam učenja	adam	lbfgs	sgd
Točnost	0.947	0.947	0.96

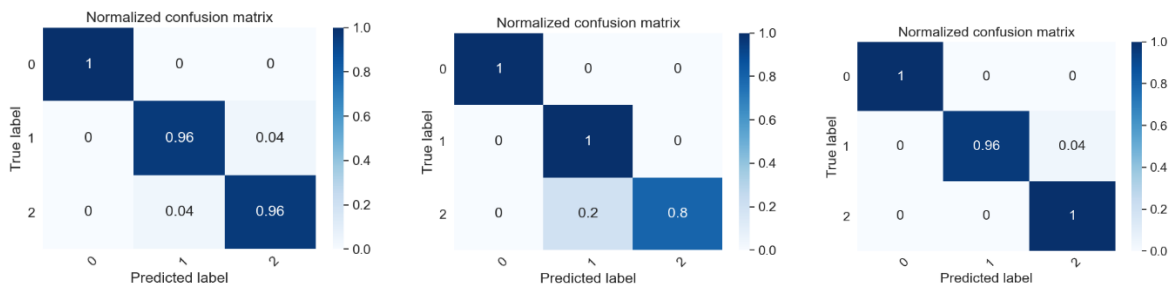


## Aktivacijska funkcija skrivenog sloja – 'tanh'

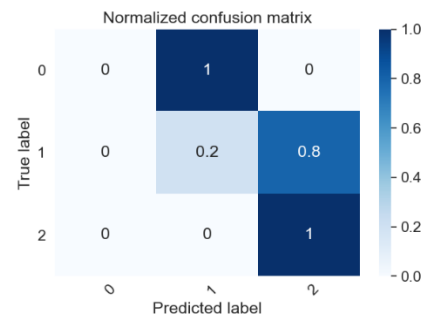
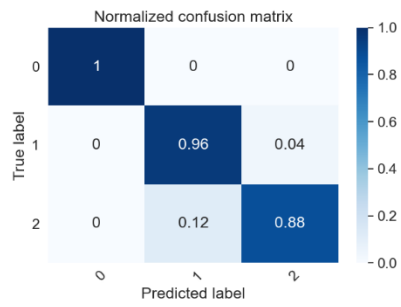
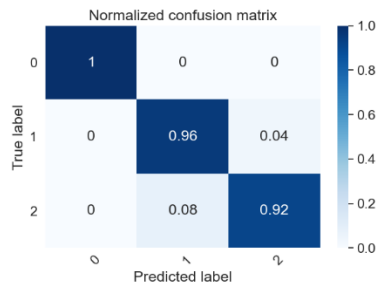
Broj skrivenih slojeva	1	1	1
Broj neurona po skrivenom sloju	5	5	5
Algoritam učenja	adam	lbfgs	sgd
Točnost	0.96	0.93	0.93



Broj skrivenih slojeva	2	2	2
Broj neurona po skrivenom sloju	30, 5	30, 5	30, 5
Algoritam učenja	adam	lbfgs	sgd
Točnost	0.973	0.93	0.987



Broj skrivenih slojeva	3	3	3
Broj neurona po skrivenom sloju	5,10,5	5,10,5	5,10,5
Algoritam učenja	adam	lbfgs	sgd
Točnost	0.96	0.947	0.4



## ZAKLJUČAK

Projektirana neuronska mreža služi za prepoznavanje tri vrste cvijeta irisa. Korištena je baza koja se sastoji od sedamdeset i šest uzoraka mjerenja karakteristika latica i čašićnih listića te se ta baza koristila za treniranje i testiranje modela. Vrste cvijeta su setosa, virginica i versicolor i svaka se razlikuje po karakteristikama. Mreža je vrednovana preko konfuzijske matrice. Izmjenom parametara mreže uočeno je da neovisno o broju neurona aktivacijska funkcija 'adam' uglavnom ima najbolju točnost, a nakon nje aktivacijska funkcije 'lbfgs' dok 'sgd' aktivacijska funkcija daje nešto lošije rezultate. Povećanjem slojeva mreže dobivaju se bolji rezultati, ali pod cijenu korištenja više resursa računala.