

SVEUČILIŠTE JOSIPA JURJA STROSSMAYERA U OSIJEKU ELEKTROTEHNIČKI
FAKULTET

Diplomski studij računarstva

Laboratorijska vježba 4

Aproksimiranje kontinuirane funkcije neuronskom mrežom

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'	11
Aktivacijska funkcija skrivenog sloja – 'tanh'	14
Aktivacijska funkcija skrivenog sloja – 'relu'.....	17
ZAKLJUČAK	20

UVOD

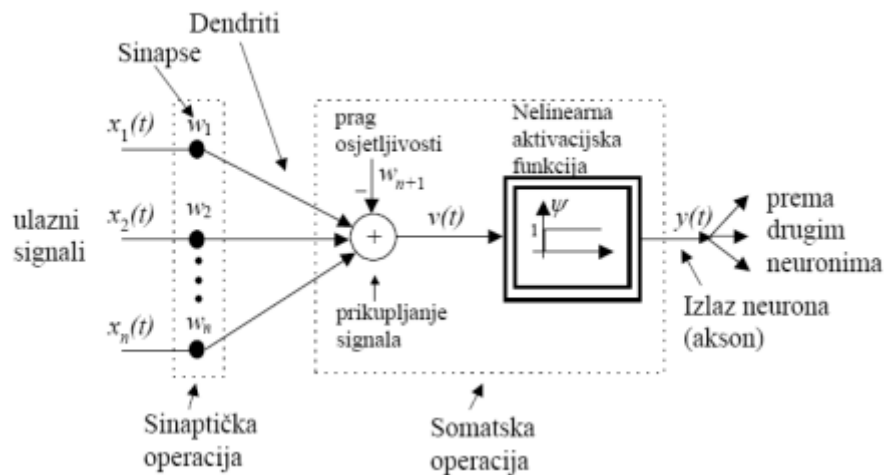
U četvrtoj laboratorijskoj vježbi potrebno je projektirati i ispitati neuronske mreže koja aproksimiraju sljedeću funkciju: $y = \frac{x^2}{15} - \frac{x}{1.5} + 1.7 * \sin(\frac{x}{1.15})$ na temelju uzoraka ulazne i izlazne veličine. Parametre zadatka mijenjati na slijedeće vrijednosti:

- Broj uzoraka funkcije : 10, 30, 60.
- 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'.

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.

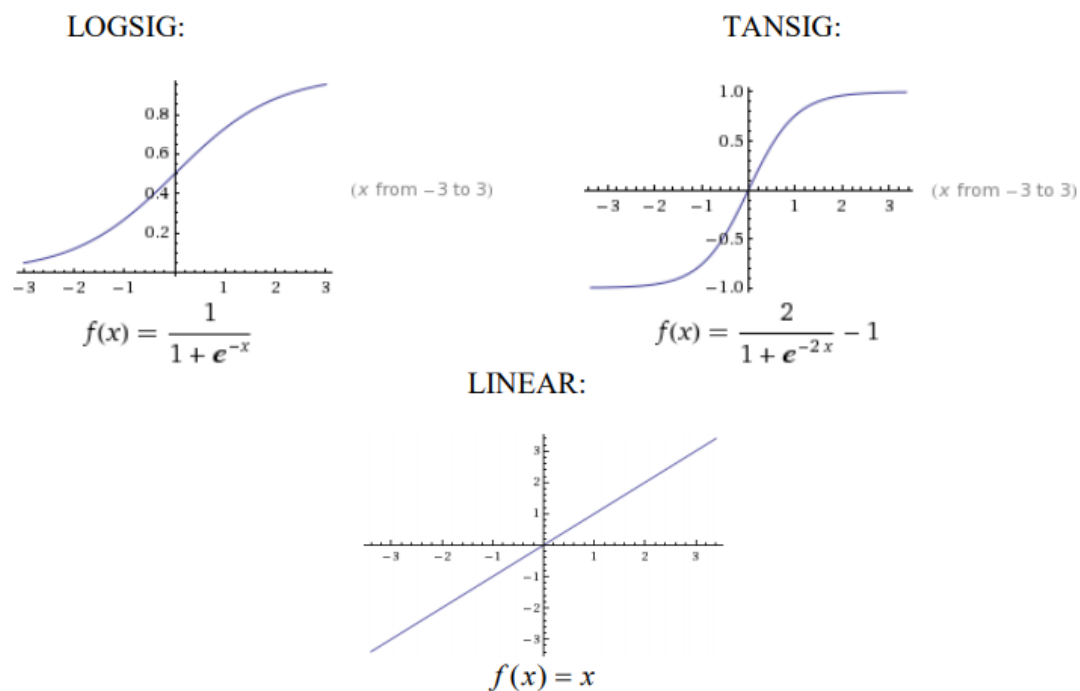


Slika 1. Shematski prikaz perceptrona

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

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

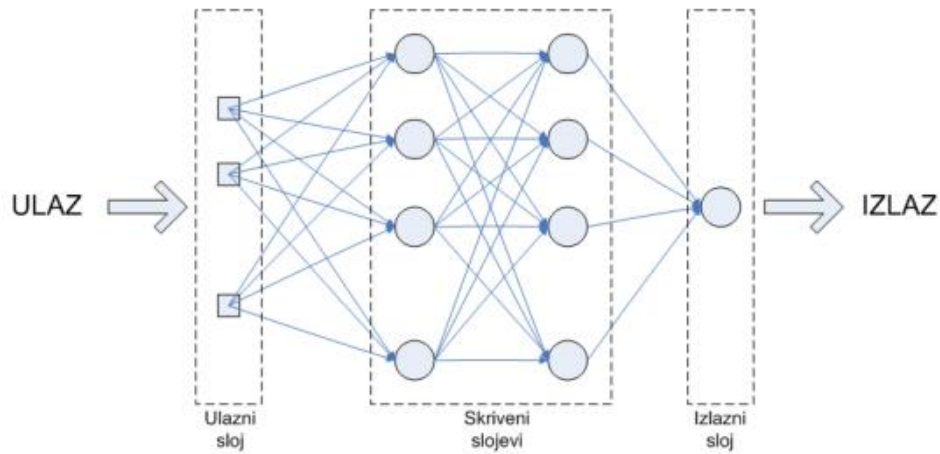
određene povratne veze i promjenjive aktivacijske funkcije. Kao što je ranije rečeno aktivacijske funkcije najčešće su sigmoide kao što vidimo na slijedećim slikama.



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:

1. Dovođenje na ulaz neuronske mreže niz slučajeva (uzoraka) koje želimo naučiti raspoznavati
2. Odrediti pogrešku između dobivenog izlaza i željenog izlaza
3. 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*. 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 koji se mijenjaju su dodani u kod kao globalne varijable te se pomoću *for* petlji izmjenjuju. Na taj način je cijeli proces automatiziran.

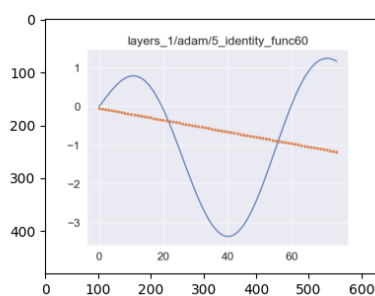
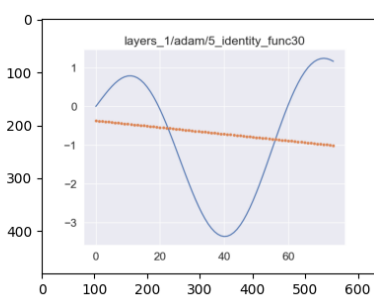
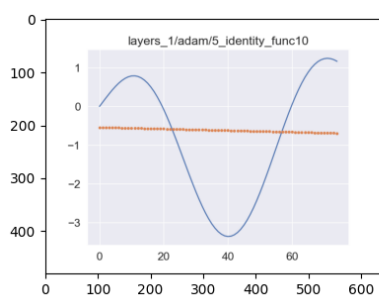
REZULTATI

Parametri zadatka mijenjani su na sljedeće vrijednosti:

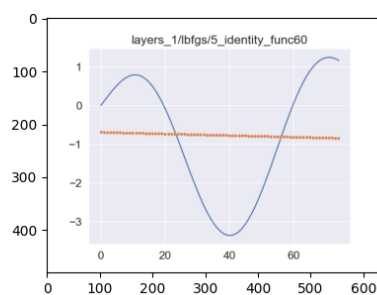
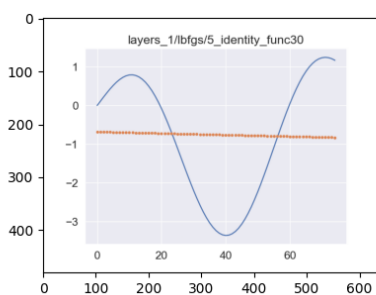
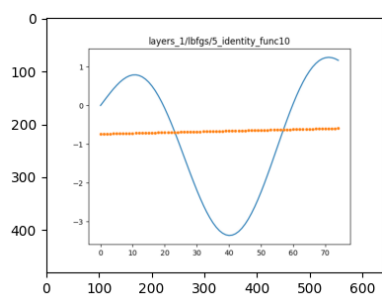
- Broj uzoraka funkcije : 10, 30, 60
- Broj neurona po skrivenom sloju : 5, 10, 30
- Broj skrivenih slojeva : 1, 2, 3
- Aktivacijska funkcija skrivenog sloja : 'identity', 'logistic', 'tahn', 'relu'
- Algoritam učenja : 'lbfgs', 'sgd', 'adam'

Aktivacijska funkcija skrivenog sloja – 'identity'

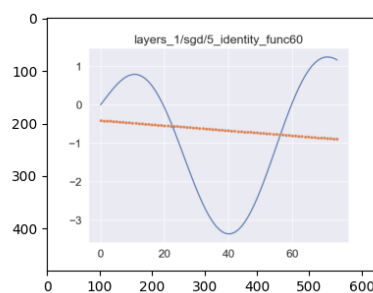
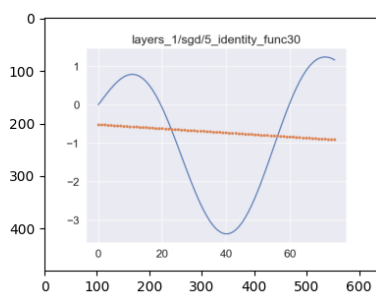
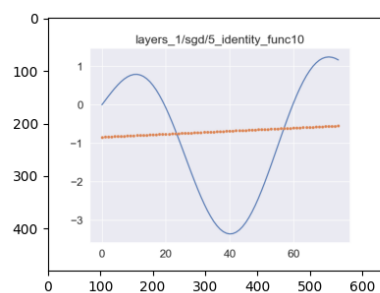
Broj skrivenih slojeva	1		
Broj neurona po skrivenom sloju	5		
Algoritam učenja	adam		
Broj uzoraka funkcije	10	30	60
MSE	2.5029	2.5051	2.5831



Broj skrivenih slojeva	1		
Broj neurona po skrivenom sloju	5		
Algoritam učenja	lbfgs		
Broj uzoraka funkcije	10	30	60
MSE	2.4980	2.4790	2.4787

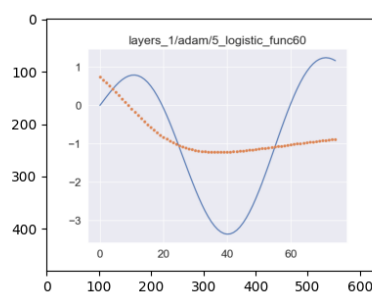
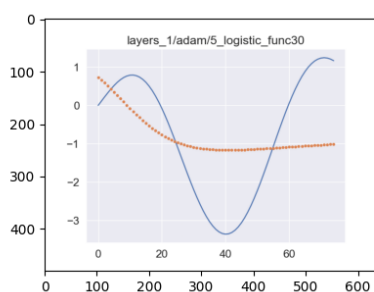
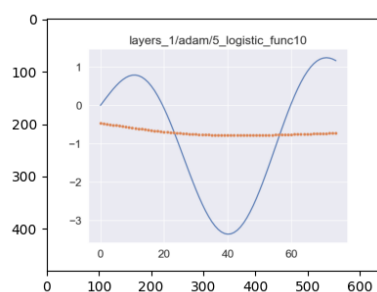


Broj skrivenih slojeva	1		
Broj neurona po skrivenom sloju	5		
Algoritam učenja	sgd		
Broj uzoraka funkcije	10	30	60
MSE	2.502	2.4868	2.5016

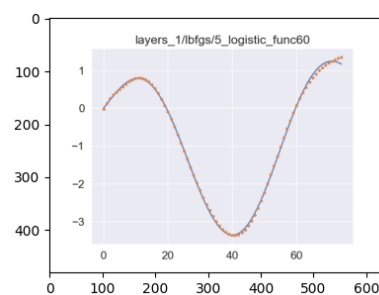
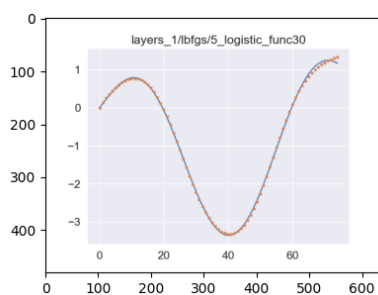
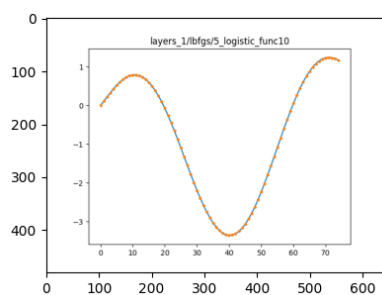


Aktivacijska funkcija skrivenog sloja – 'logistic'

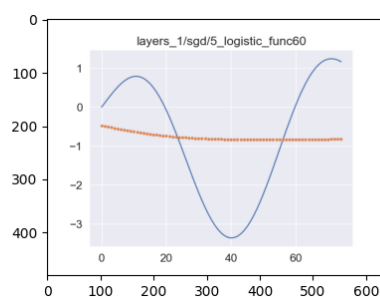
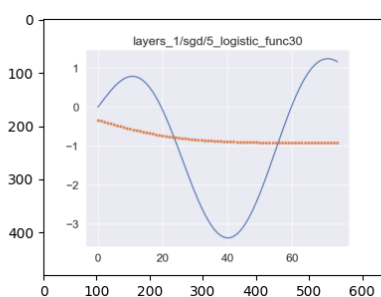
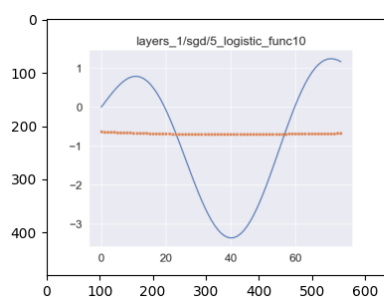
Broj skrivenih slojeva	1		
Broj neurona po skrivenom sloju	5		
Algoritam učenja	adam		
Broj uzoraka funkcije	10	30	60
MSE	2.3348	1.9411	1.825



Broj skrivenih slojeva	1		
Broj neurona po skrivenom sloju	5		
Algoritam učenja	lbfgs		
Broj uzoraka funkcije	10	30	60
MSE	0.00004	0.00157	0.00197

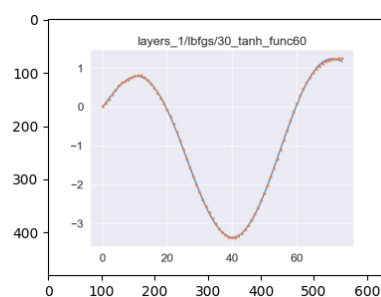
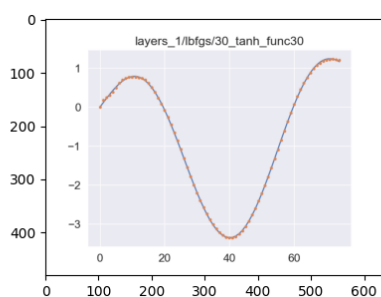
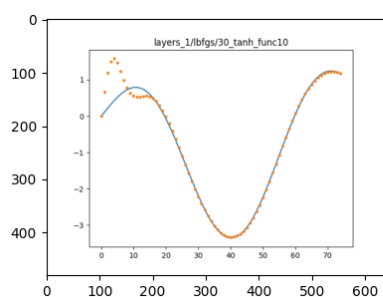


Broj skrivenih slojeva	1		
Broj neurona po skrivenom sloju	5		
Algoritam učenja	sgd		
Broj uzoraka funkcije	10	30	60
MSE	2.4529	2.3027	2.3529

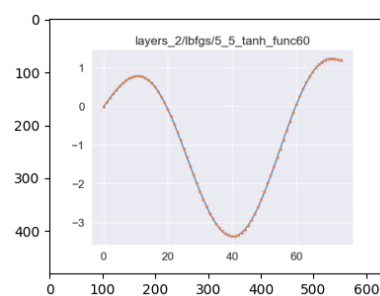
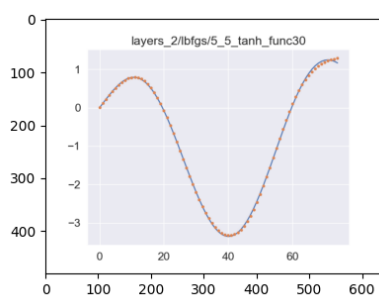
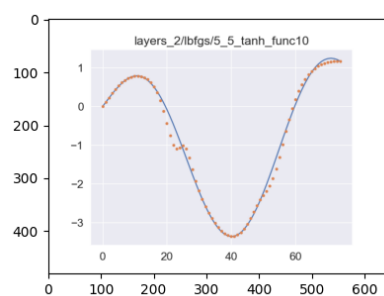


Aktivacijska funkcija skrivenog sloja – 'tanh'

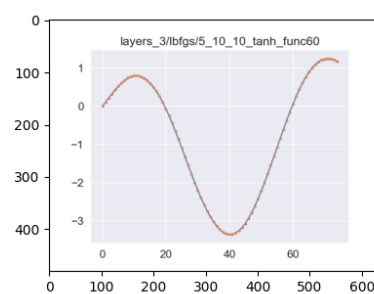
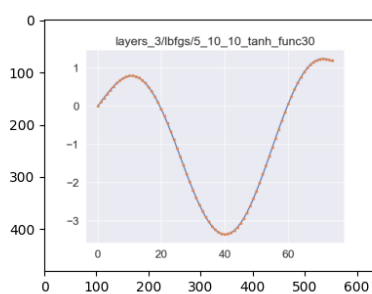
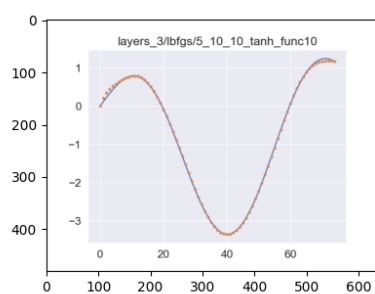
Broj skrivenih slojeva	1		
Broj neurona po skrivenom sloju	30		
Algoritam učenja	lbfgs		
Broj uzoraka funkcije	10	30	60
MSE	0.0757	0.00029	0.0004



Broj skrivenih slojeva	2		
Broj neurona po skrivenom sloju	5,5		
Algoritam učenja	lbfgs		
Broj uzoraka funkcije	10	30	60
MSE	0.0205	0.0009	0.00013

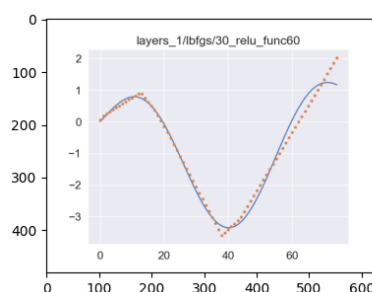
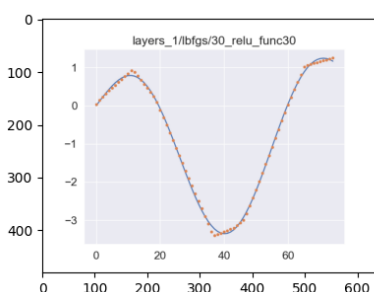
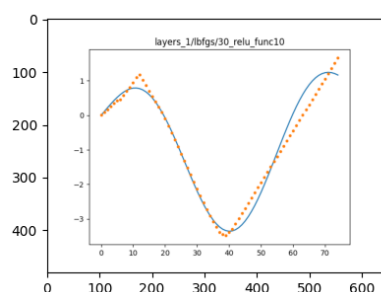


Broj skrivenih slojeva	3		
Broj neurona po skrivenom sloju	5,10,10		
Algoritam učenja	lbfgs		
Broj uzoraka funkcije	10	30	60
MSE	0.0010	$2.7366 \cdot 10^{-5}$	$3.6146 \cdot 10^{-5}$

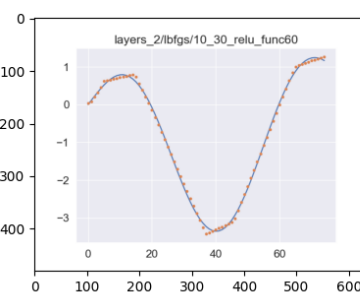
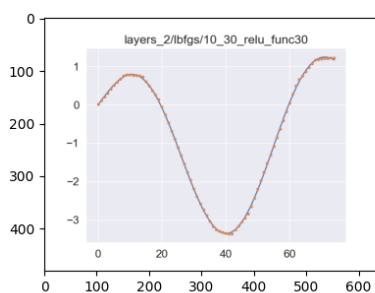
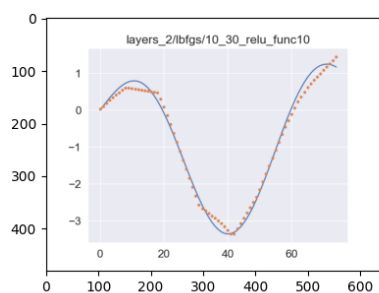


Aktivacijska funkcija skrivenog sloja – 'relu'

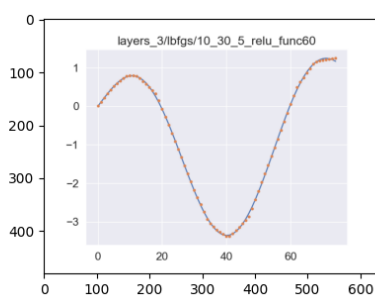
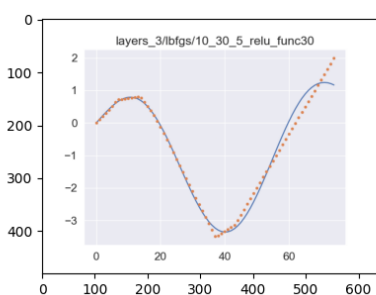
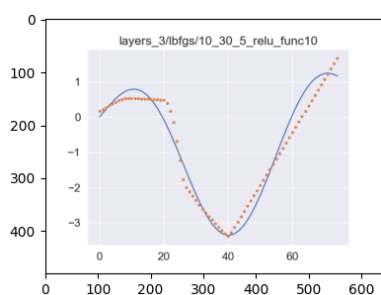
Broj skrivenih slojeva	1		
Broj neurona po skrivenom sloju	30		
Algoritam učenja	lbfgs		
Broj uzoraka funkcije	10	30	60
MSE	0.0682	0.0032	0.0497



Broj skrivenih slojeva	2		
Broj neurona po skrivenom sloju	10,30		
Algoritam učenja	lbfgs		
Broj uzoraka funkcije	10	30	60
MSE	0.0214	0.0005	0.0041



Broj skrivenih slojeva	3		
Broj neurona po skrivenom sloju	10,30,5		
Algoritam učenja	lbfgs		
Broj uzoraka funkcije	10	30	60
MSE	0.0987	0.04767	0.0007



ZAKLJUČAK

Najbolji rezultati su dobiveni korištenjem L-BFGS algoritma u kombinaciji sa 'tahn' i 'logistic' aktivacijskim funkcijama u skrivenom sloju. Zadatak je da se aproksimira nelinearna funkcija stoga 'identity' funkcija u kombinaciji sa bilo kojim algoritmom ne daje dobre rezultate. Ona predstavlja linearnu funkciju. Aktivacijska funkcija 'relu' uz kombinaciju bilo kojeg algoritma za učenje također ne daje dobre rezultate. Razlog tome je što je i ona dosta jednostavna i slična linearnoj funkciji.

Povećanjem slojeva mreže dobivaju se bolji rezultati, ali pod cijenu korištenja više resursa računala. Zbog jednostavnosti zadatka postignuti su dobri rezultati i sa samo jednim skrivenim slojem.

Povećanjem broja uzoraka funkcije dobivaju se bolji rezultati jer mreža ima više podataka za učenje. Podaci za treniranje moraju biti što raznolikiji kako ne bi došlo do overfittinga odnosno kako bi mreža radila na što raznolikijem inputu. Premali broj podataka za treniranje dovodi do loših rezultata predviđanja.