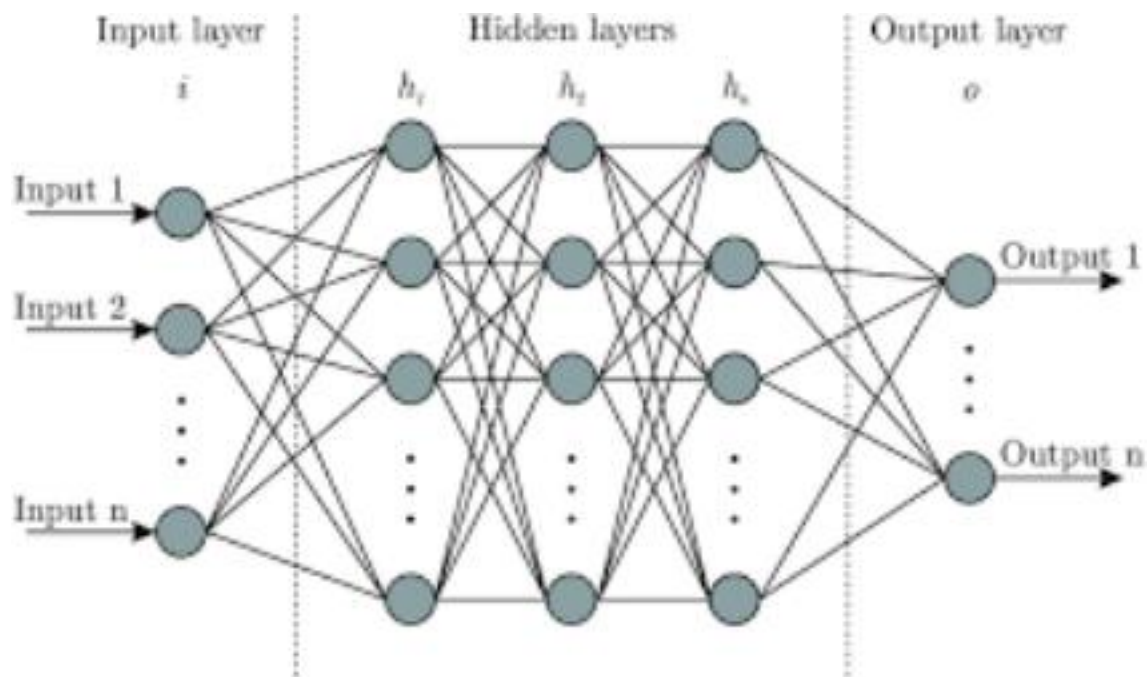


UNIVERZITET U BIHAĆU  
TEHNIČKI FAKULTET BIHAĆ  
ELEKTROTEHNIKA/INFORMATIKA



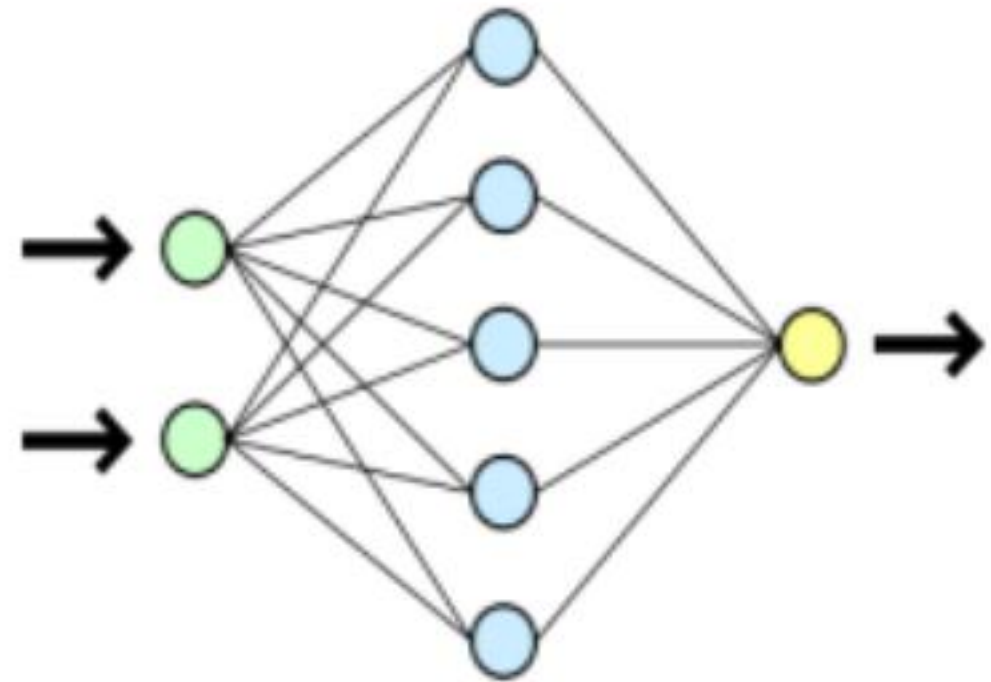
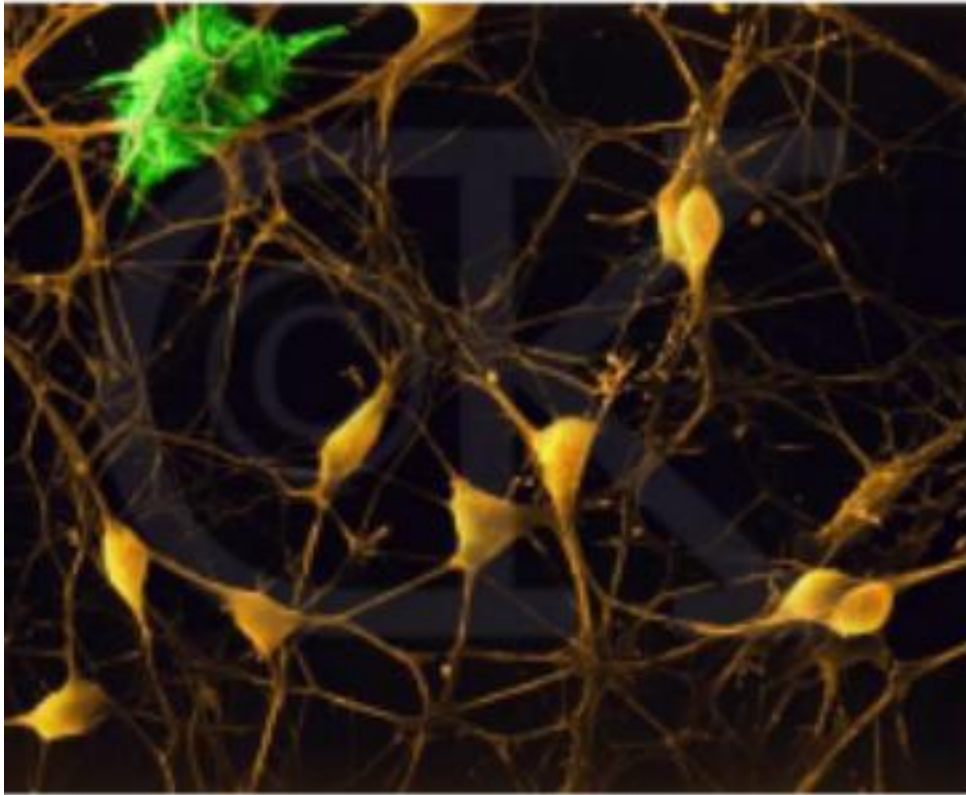
**AUDITORNE VJEŽBE**

**PREDMET:** *INTELIGENTNI SISTEMI*

**OBLAST:** *UVOD U NEURONSKE MREŽE*

*Asistent: Una Drakulić, bach.el.*

# BIOLOŠKA I VJEŠTAČKA NEURONSKA MREŽA



# BIOLOŠKA I VJEŠTAČKA NEURONSKA MREŽA

- **Mozak:**
  - $10^{10}$  neurona
  - $10^{13}$  veza između neurona
  - Brzina rada na nivou milisek
  - Potpuno paralelni rad
- **VNM:**
  - Do 20 000 neurona
  - Brzina rada na nivou nanosek
  - Simulacija paralelnog rada

# DEFINICIJA: NEURONSKA MREŽA

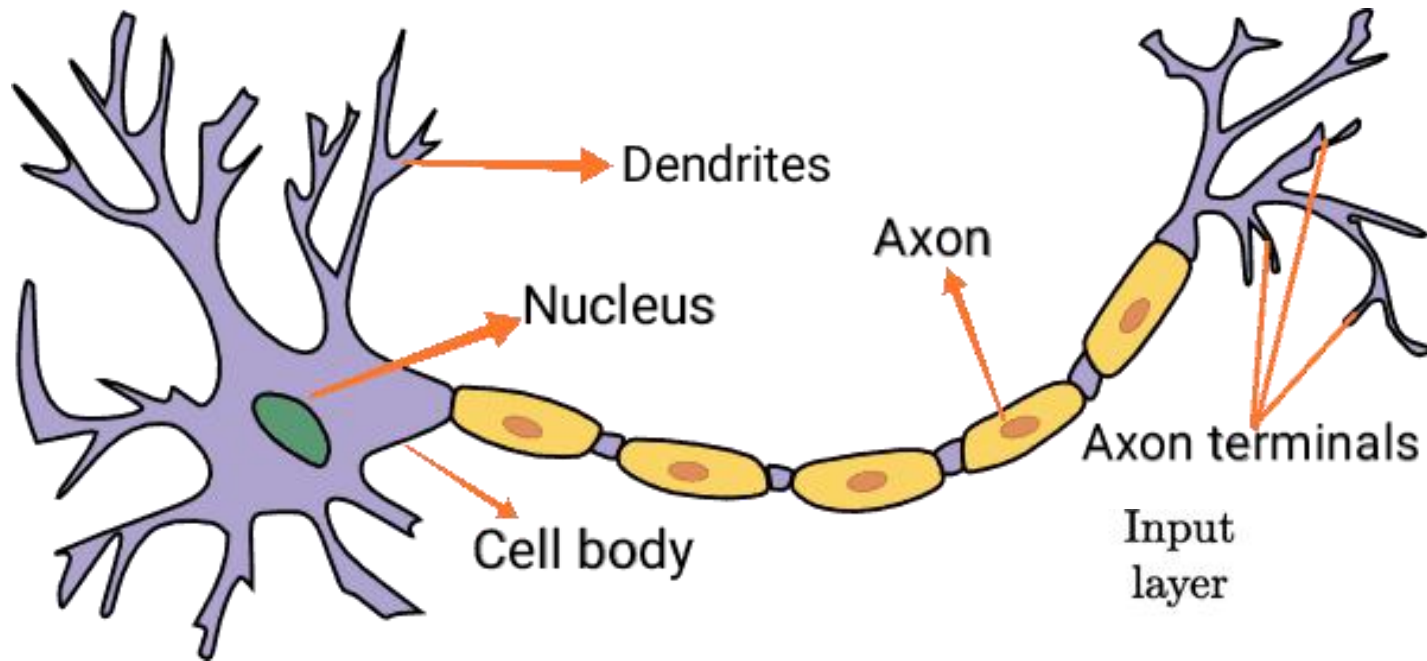
**DARPA:** Neuronska mreža je sistem koji se sastoji od velikog broja međusobno povezanih, jednostavnih elemenata procesiranja koji rade paralelno. Funkcija NM je određena strukturom mreže, težinom veza, i obradom u elementima procesiranja.

**Haykin:** Neuronska mreža je paralelni distribuirani procesor koji ima prirodnu sposobnost čuvanja i korišćenja iskustvenog znanja. Sličnost sa mozgom se ogleda kroz dve osobine:

- mreža stiče znanje kroz proces učenja
- znanje se čuva u vezama između neurona (sinaptičkim težinama)

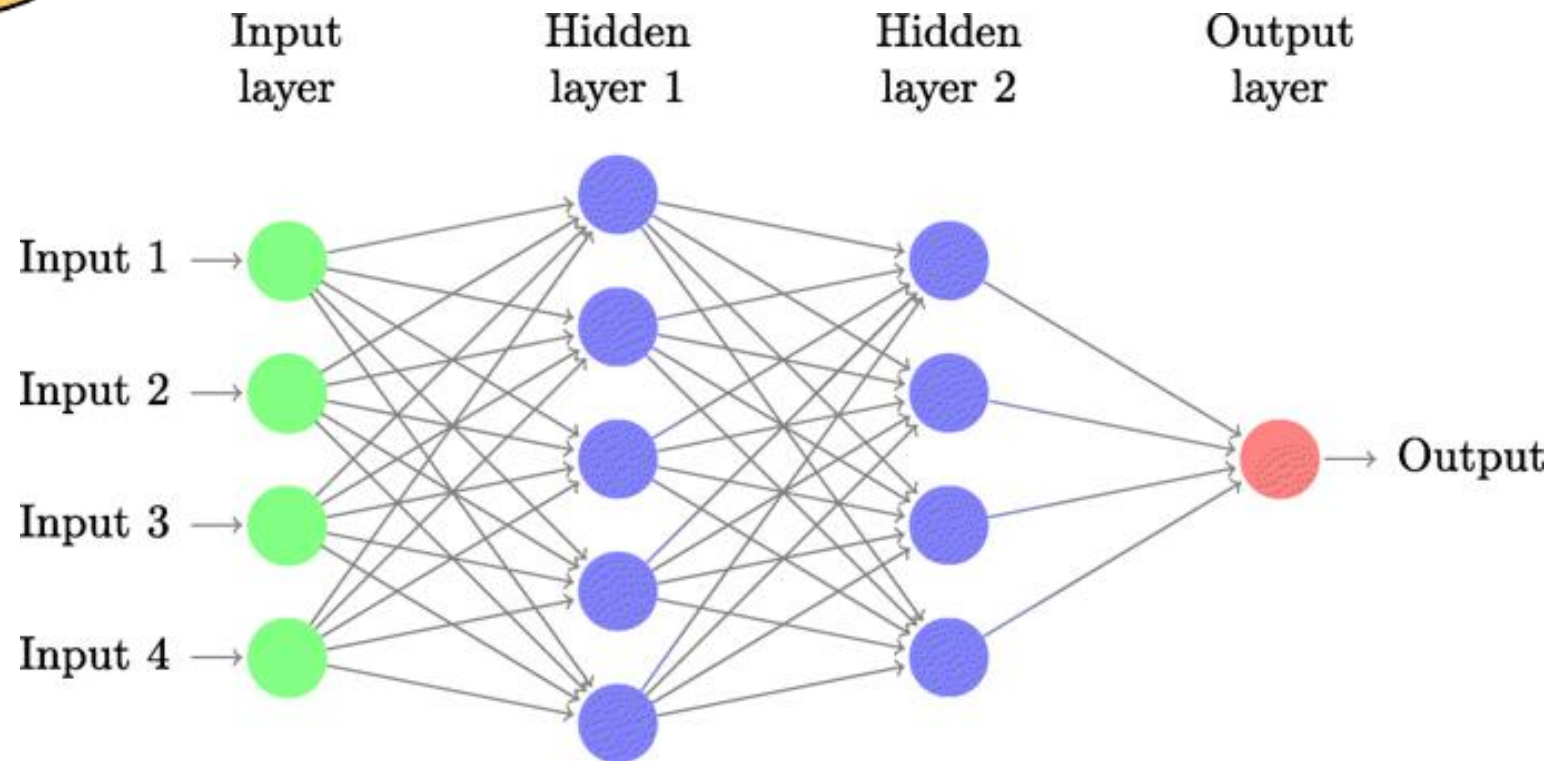
**Zurada:** Veštački neuro sistemi ili neuronske mreže, su ćelijski sistemi koji mogu da stiču, čuvaju i koriste iskustveno znanje.

# BIOLOŠKI I VJEŠTAČKI NEURON



Osnovni dijelovi:

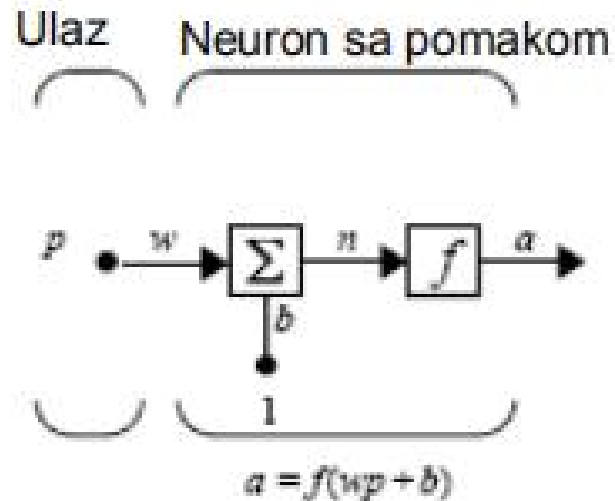
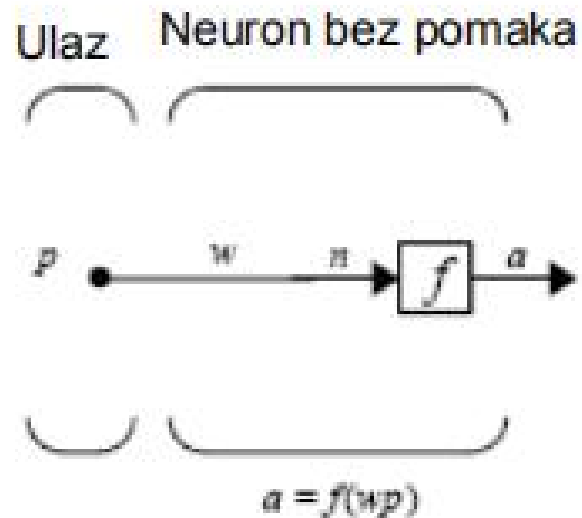
- tijelo (soma),
- dendriti (ulazi),
- akson (izlaz),
- sinapse (spojevi)



# MODEL NEURONA

## NEURON S JEDNIM ULAZOM

Neuron s jednim ulazom i bez pomaka prikazan je na slici. Skalarni ulaz  $p$  prenosi se kroz model i pri tome se množi sa skalarnom težinom  $w$  i formira produkt  $wp$ , ponovo skalar. U ovom slučaju otežani ulaz  $wp$  predstavlja jedini argument prijenosne funkcije  $F$ , koja generiše skalarni izlaz  $a$ .



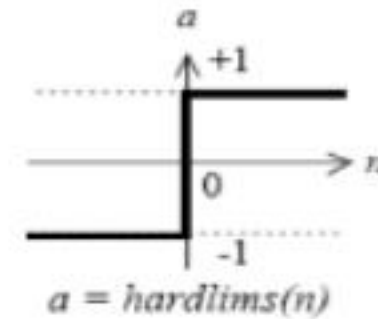
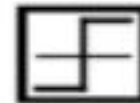
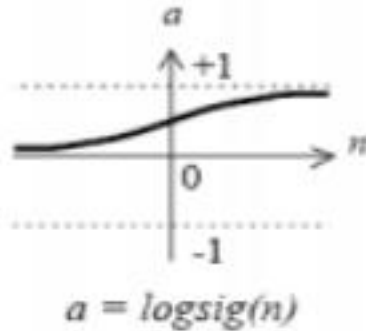
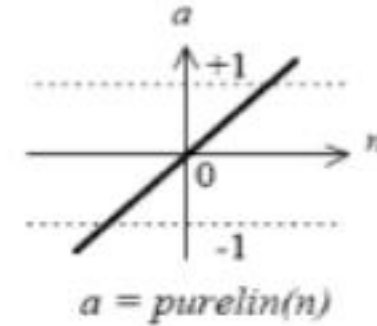
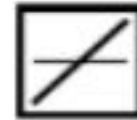
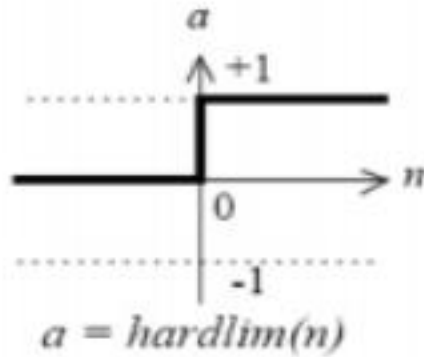
Neuron na b) ima skalarni pomak (engl. bias, offset)  $b$ . Pomak  $b$  se može promatrati kao težina, s tim da uvijek ima konstantan ulaz 1. Ulazni argument  $n$  prijenosne funkcije  $F$  je, također, skalar i jednak je sumi otežanog ulaza  $w_p$  i pomaka  $b$ .  $F$  je prijenosna (aktivacijska) funkcija koja, na osnovu ulaznog argumenta  $n$ , generiše izlaz  $a$ .

## **PRIJENOSNE (AKTIVACIJSKE) FUNKCIJE**

Pri sintezi neuronskih mreža koristi se veliki broj prijenosnih funkcija. Mnoge od njih su implementirane u MATLAB-u u okviru Neural Network Toolbox-a. Zbog otvorene arhitekture MATLAB-ovog razvojnog okruženja, ostavljena je mogućnost proširenja postojećeg skupa prijenosnih funkcija vlastitim.



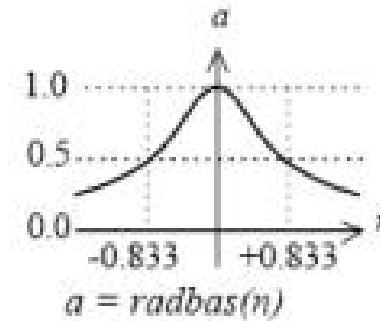
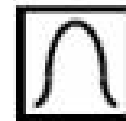
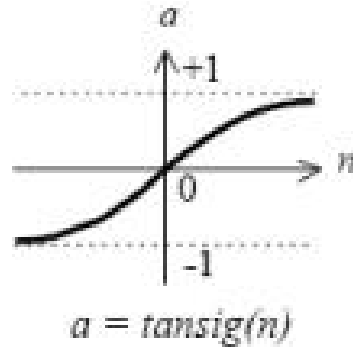
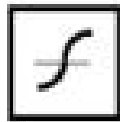
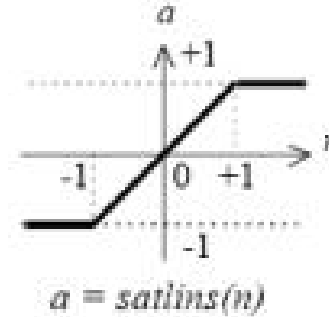
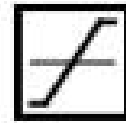
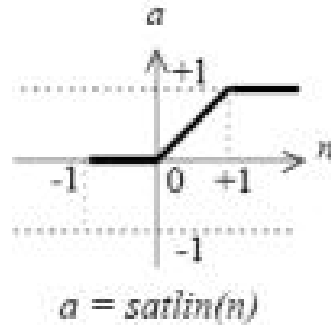
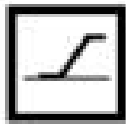
# ***PRIJENOSNE (AKTIVACIJSKE) FUNKCIJE***



*Grafički prikaz prenosnih funkcija Hard Limit, Linear, Log-Sigmoid i Symmetric Hard Limit*



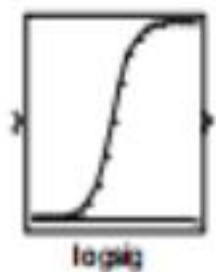
# ***PRIJENOSNE (AKTIVACIJSKE) FUNKCIJE***



*Grafički prikaz prenosnih funkcija Satlin, Satlins, Tan-Sigmoid i Radial Basis*

# LOGSIG

Logsig (llogaritam sigmoid) funkcija (Sl.) uzima ulaz (koji može biti bilo koja realna vrijednost iz intervala  $(-\infty, +\infty)$ ) i obavlja zbijanje (zgušnjavanje) izlaza u interval  $(0, 1)$ . Izlaz  $a$  neurona određen je izrazom :



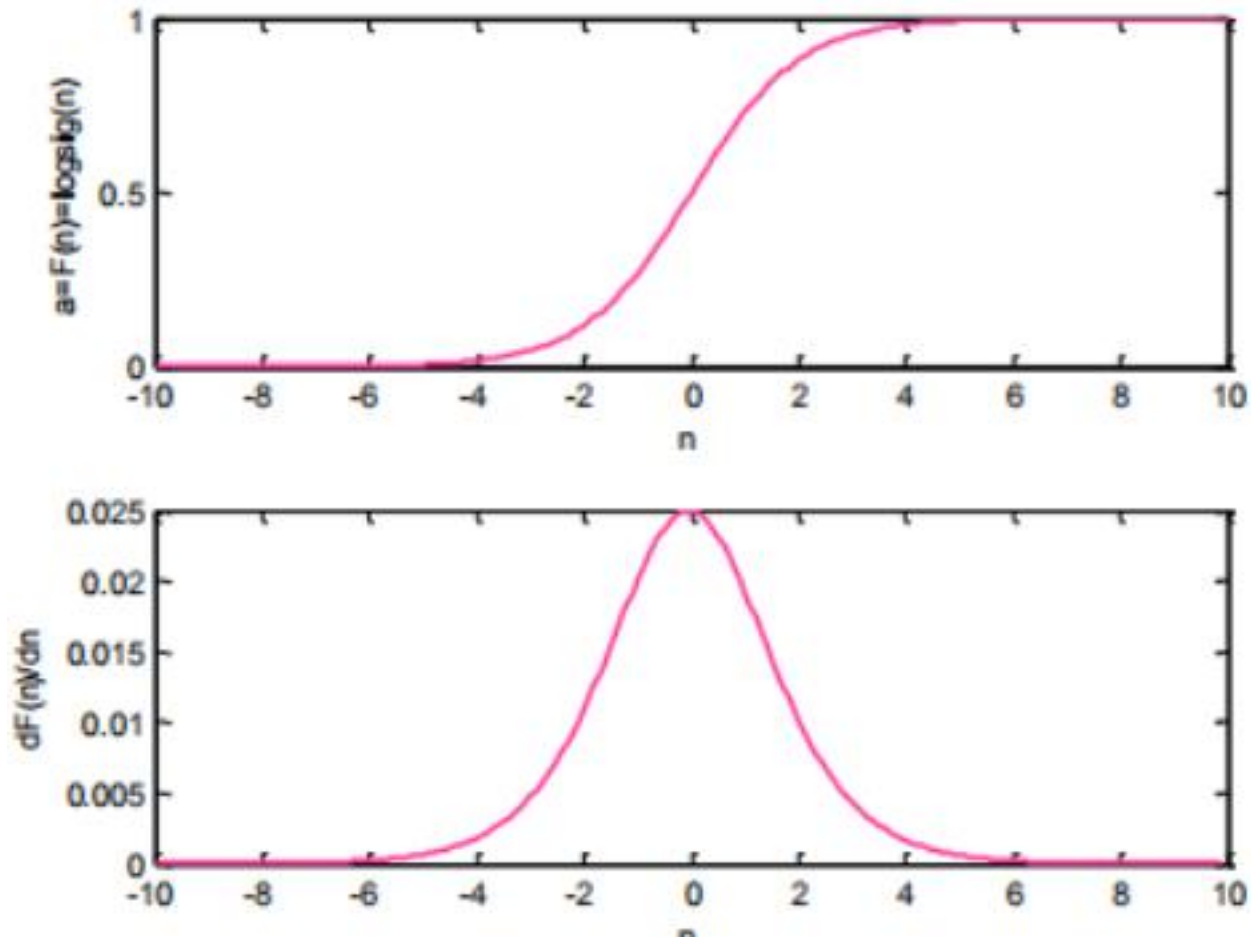
$$a = F(n) = 1/(1+e^{-n})$$

Dok je derivacija od prijenosne funkcije:

$$\frac{da}{dn} = \frac{dF(n)}{dn} = \frac{e^{-a}}{[1 + e^{-a}]^2} = a(1-a)$$

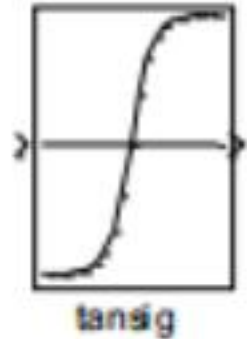
Ova prijenosna funkcija se često koristi u neuronskim mrežama sa povratnim prostiranjem izlazne pogreške (eng. Backpropagation Neural Networks - BPNN), budući da je posvuda derivabilna, a derivacija se jednostavno računa.

# LOGSIG



Logsig prijenosna funkcija (a) i  
njena derivacija (b)

# TANSIG

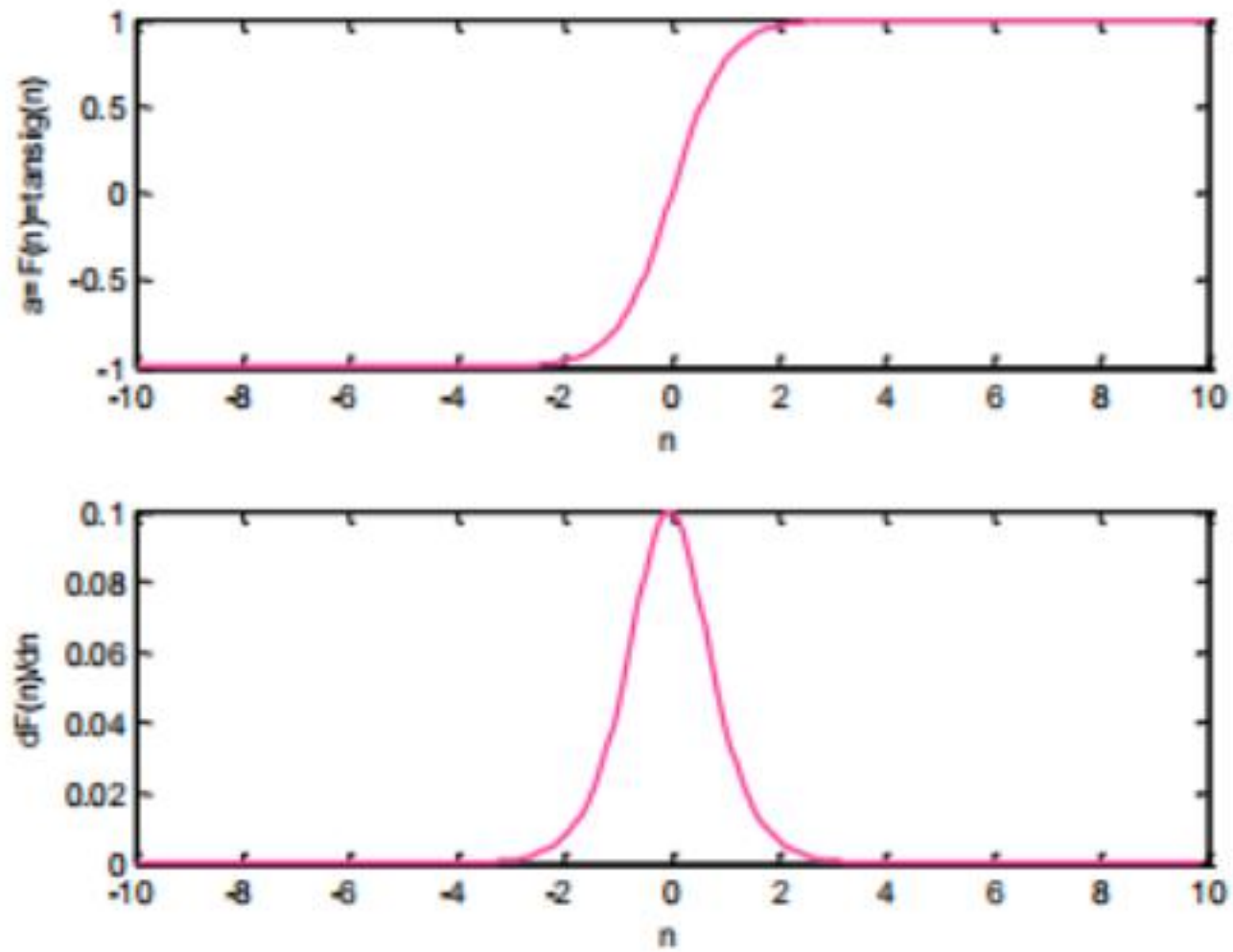


b. **Tansig** (tangent sigmoid) funkcija ( Sl. ) uzima ulaz (koji može biti bilo koja realna vrijednost iz intervala  $(-\infty, +\infty)$  ) i vrši zbijanje (zgušnjavanje) izlaza u simetričan interval  $(-1, 1)$ . Izlaz  $a$  neurona sa *Tansig* prijenosnom funkcijom određen je izrazom:

$$a = F(n) = \frac{2}{1 + e^{-2n}} - 1$$

$$\frac{da}{dn} = \frac{dF(n)}{dn} = \frac{4e^{-2n}}{[1 + e^{-2n}]^2} = 1 - a^2$$

# TANSIG

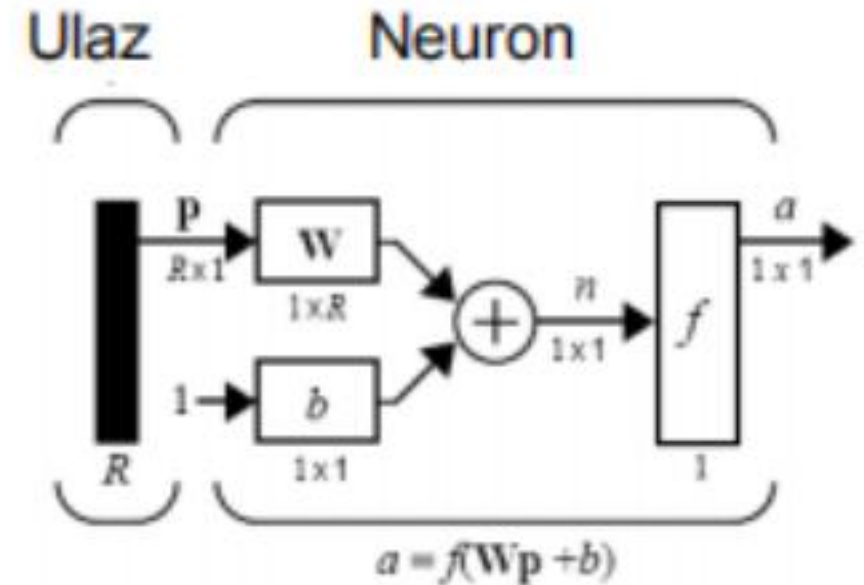
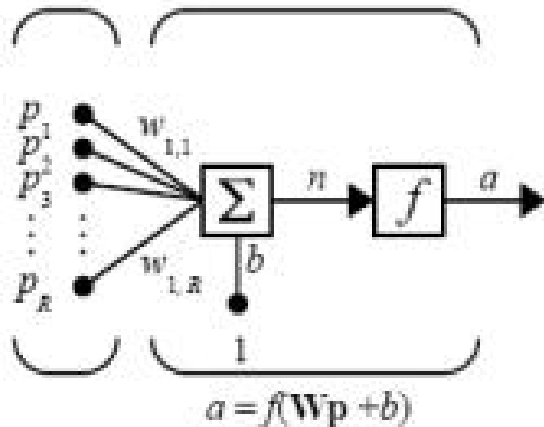


# NEURON SA VIŠE ULAZA

Neuron sa R ulaza prikazan je na slici. Pojedini skalarni ulazi  $p(1), p(2), \dots, p(R)$  množe se (otežavaju se) sa skalarnim težinama  $w(1, 1), w(1, 2), \dots, w(1, R)$  i vode na mjesto sumiranja. Njihova suma je, u stvari, skalarni proizvod  $\mathbf{w} \mathbf{p}$  vektora retka  $1 \times R$  i vektora stupca  $R \times 1$ . Neuron ima pomak  $b$ , koji se sabira sa otežanim ulazima i formira ukupni net ulaz  $n$  (ulaz u prijenosnu funkciju  $F$ ):

$$\mathbf{W} = [w(1, 1) \ w(1, 2) \ \dots \ w(1, R)] \quad \cdot \quad \mathbf{P} = \begin{bmatrix} p(1) \\ p(2) \\ \vdots \\ p(R) \end{bmatrix}$$

$$n = \mathbf{w} \mathbf{p} + b = \sum_{i=1}^R w(1, i) p(i) + b$$

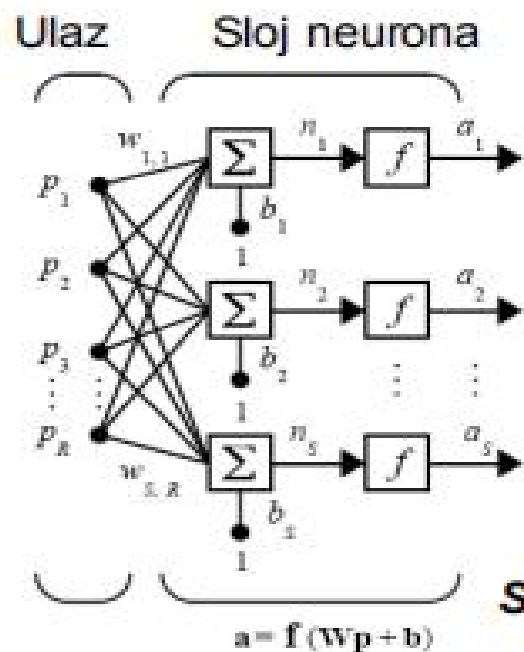


Slika: Model neurona sa više ulaza

# ARHITEKTURA NEURONSKE MREŽE

Dva ili više neurona mogu se kombinovati da tvore sloj. Neuronska mreža može se sastojati od više takvih slojeva međusobno integrisanih u cjelovit sistem. Najprije će se razmatrati osnovna struktura koja se sastoji od jednog sloja neurona. Nakon toga promatrat će se višeslojna mreža.

## JEDNOSLOJNA STRUKTURA - OSNOVNA STRUKTURA NEURONSKE MREŽE



**Slika** Jednoslojna struktura sa  $R$  ulaza i  $S$  neurona



# ARHITEKTURA NEURONSKE MREŽE

U ovoj strukturi svaki element ulaznog vektora  $p$  povezan je sa svakim neuronom preko težinske matrice  $W$ .  $i$ -ti neuron ima sumator koji zbraja otežane ulaze i pomak u taj neuron i formira vlastiti skalarni ukupni ulaz  $n(i)$  po formuli:

$$n(i) = + b(i), i = 1, \dots, S$$

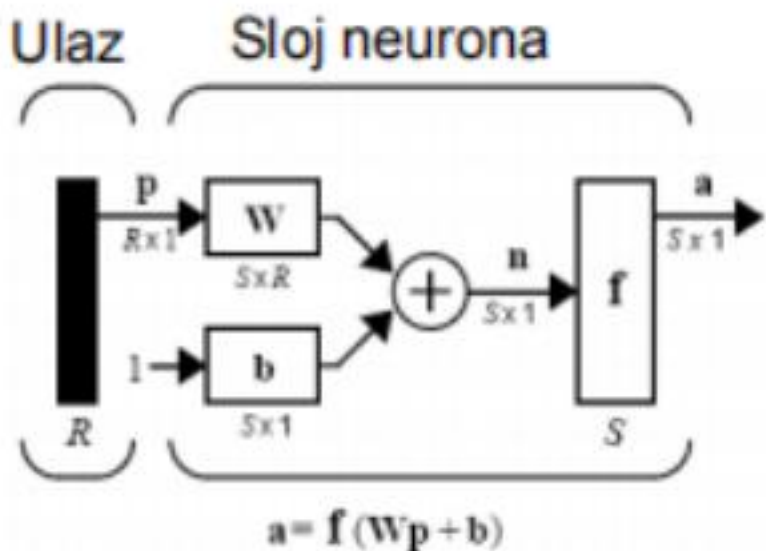
Pojedini ukupni ulazi  $n(i)$ ,  $i = 1, 2, \dots, S$ , uzeti zajedno, formiraju vektor  $n_{S \times 1}$ . Izlazi pojedinih neurona  $a(i)$  formiraju vektor  $a_{S \times 1}$ . Potrebno je primijetiti da je u opštem slučaju broj ulaza u sloj različit od broja izlaza ( $R \neq S$ ).

Težinska matrica  $W$  ima oblik

$$W = \begin{bmatrix} w(1,1) & w(1,2) & \dots & w(1,R) \\ w(2,1) & w(2,2) & \dots & w(2,R) \\ \vdots & \vdots & \dots & \vdots \\ w(S,1) & w(S,2) & \dots & w(S,R) \end{bmatrix}$$

# ARHITEKTURA NEURONSKE MREŽE

Element  $w(i,j)$  predstavlja težinu kojom se množi  $j$ -ti ulaz  $i$ -tog neurona. Jednoslojna struktura s  $R$  ulaza i  $S$  neurona može se u kompaktnoj formi prikazati kao na slici.

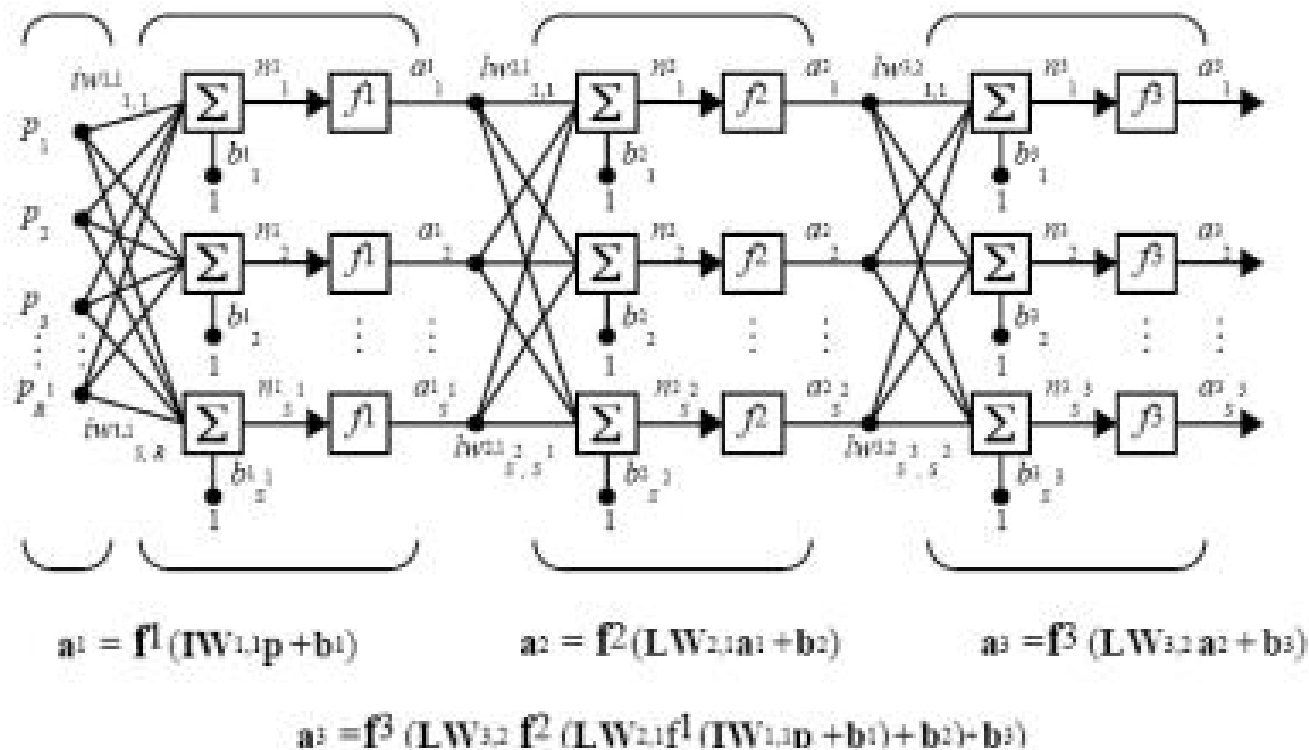


**Slika.** Kompaktno označavanje za predstavljanje  $S$  neurona sa  $R$  ulaza

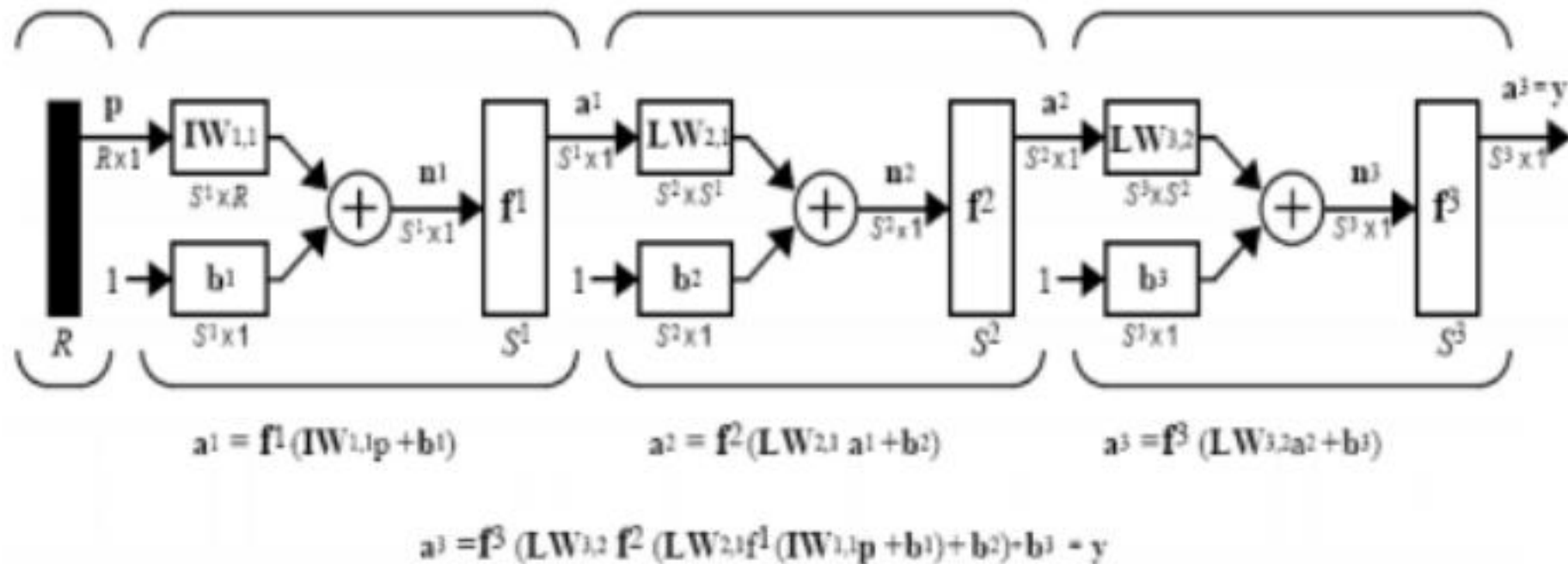
# VIŠESLOJNA NEURONSKA MREŽA

Upotrebljiva neuronska mreža mora imati više od jednog sloja. Svaki sloj ima težinsku matricu  $W$ , vektor pomaka  $b$  i izlazni vektor  $a$ . Da bismo razlikovali težinske matrice, izlazne vektore itd. za svaki sloj, dodaje se oznaci vektora ili matrice indeks (broj) pripadajućeg sloja. Na primjer, MATLAB izraz za težinsku matricu drugog sloja je  $W2$ . Upotreba ove notacije može se vidjeti na Sl., gdje je prikazana troslojna mreža.

Slika. Troslojna neuronska mreža sa  $R$  ulaza,  $S1$  neuronu u prvom sloju,  $S2$  neurona u drugom sloju i  $S3$  neurona u trećem sloju



# VIŠESLOJNA NEURONSKA MREŽA



**Slika.** Kompaktno označavanje za prikaz troslojne neuronske mreže sa  $R$  ulaza,  $S^1$  neurona u prvom sloju,  $S^2$  neurona u drugom sloju i  $S^3$  neurona u trećem sloju

# PROGRAMIRANJE NEURONSKE MREŽE

## OSNOVNI PODACI ZA TRENIRANJE NEURONSKE MREŽE U MATLAB-U

### STATIČKE NEURONSKE MREŽE

- nemaju povratne veze (nema povratka u više slojeve)
- kreiranje neuronske mreže se vrši naredbom:

**net=newff(PR, [S1, S2, ..., SN], {'TF1','TF2',..'TFN'},'BFT');**

gdje je:

PR – min i max vrijednost funkcije

S – velicina sloja

TF – aktivacijska funkcija sloja

BTF – algoritam učenja, metoda učenja (treniranje)

# PROGRAMIRANJE NEURONSKE MREŽE

## OSNOVNI PODACI ZA TRENIRANJE NEURONSKE MREŽE U MATLAB-U

Naredba za treniranje mreže:

```
net1=train(net,x,y);
```

gdje je:

net- kreirana mreža

x – ulaz

y – izlaz

Nakon toga treba prikazati istreniranu mrežu metodom simulacije:

```
ytest=sim(net1,x);
```

gdje je:

net1 – istrenirana mreža

x - ulaz

*Potrebno je otvoriti Matlab, te kreirati novi m.file. Ovaj kod prepisati u kreirani m.file te pokrenuti.*

**PRIMJER 1: Kreirati funkciju humps,  $y=(1/((x-3)^2+0.01) + 1/((x-9)^2+0.04)) - 6$**

```
x=0:0.05:2; //vremeska osa za crtanje zdane funkcije
```

```
y=humps(x); // ova je funkcijaugrađena u matlab pod ovim nazivom zato se ne mora pisati cijela vec samo pozvati
```

```
P=x; // precrtavanje x i y za P i T koji ce se koristiti za neuronsku mrežu
```

```
T=y;
```

```
plot(P,T,'x'); // iscrtavanje
```

```
grid ON // postavljanje mreže
```

```
%kreiranje neuronske mreže
```

```
net=newff([0 2],[5,1],{'tansig','purelin'},'traingd');
```

```
%definiranje parametara mreže
```

```
net.trainParam.show=50;
```

```
net.trainParam.lr=0.05;
```

```
net.trainParam.epochs=1000;
```

```
net.trainParam.goal=1e-3; %greška tolerancije
```

```
%treniranje neuronske mreže
```

```
net1=train(net,P,T);
```

```
%pohrana vrijednosti - simulacija
```

```
a=sim(net1,P);
```

```
%iscrtavanje rezultata i poredenje
```

```
plot(P,a,'r',P,T,'g');
```

```
legend('neuronska mreža','humps funkcija');
```



# PRIMJER 2

Zadatak glasi: Zadana je funkcija  $x = \sin(2 \cdot \pi \cdot t^2 + 1)$ . Istrenirati zadanu funkciju.

```
t=0:0.01:1; //vrijeme za treniranje, ide od 0 do 1, sa korakom 0.01
```

```
x=sin(2*pi*t.^2+1); //zadana funkcija
```

```
plot(t,x); //iscravanje zadane funkcije nedba plot (ono sto crtamo, ono po cemu crtamo)
```

```
hold on // zadrzi stanje, tj. ostavi iscrtani graf za daljnji postupak,
```

```
P=[ 0 0.1000 0.2000 0.3000 0.4000 0.5000 0.6000 0.7000 0.8000 0.9000 1.0000]; // slojevi u mrezi zadane funkcije, ove vrijednosti dobijete kada pokrenete prethodni dio koda ispisan u komandnom prostoru, i samo se kopira ovdje
```

```
T=[0.8415 0.8737 0.9494 1.0000 0.9071 0.5403 -0.1201 -0.8059 -0.9527 -0.1926 0.8415]; // isto
```

```
net=newff([0 1], [10 1], {'tansig', 'purelin'}); // naredba za kreiranje nove neuronske mreže
```

```
Y=sim(net, P); // naredba za kreiranje simulink modela zadane funkcije
```

```
plot(P,Z,P,Y,'-o'); // iscrtavanje mreže prije i poslije treniranja
```

```
hold on
```

```
pause
```

```
net.trainParam.epochs=500; // broj epoha za treniranje NM
```

```
net=train(net,P,T); // treniranje
```

```
figure // iscrtavanje
```

```
Y=sim(net,P);
```

```
plot (P,T,P,Y,'*'); // ponovno crtanje istrenirane mreže
```

```
gensim(net) // simulink model istrenirane mreže
```

# Kada porenete porgram trebate dobiti sljedeci odziv

