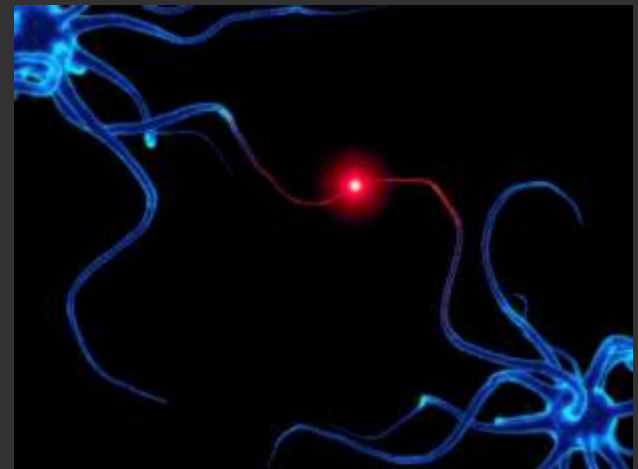


VEŠTAČKA INTELIGENCIJA 2020/21

VEŠTAČKE NEURONSKE MREŽE

Sadržaj

- Pojam i biološka osnova
- Perceptron
- Nuronske mreže
- Učenje



Slika preuzeta sa: <http://ldletkeman.com/Leon/Blog.html>

Uvodne informacije

- Razvoj neuronskih mreža (veštačke neuronske mreže/artificial neural network-ANN) počinje ranih 1940ih;
- Najveću popularnost stekle kasnih 1980ih, kao posledica novih tehnika i razvoja računarskog hardvera.
- Popularnost **danas** je posledica dostupnosti velike količine podataka (Big Data) i potrebe za njihovom analizom (Machine Learning, Deep Learning)
- Neke ANN u modeli bioloških neuronskih mreža, neke nisu, ali istorijski inspiracija za celu oblast potiče od želje da se proizvedu veštački sistemi sposobni za **inteligentne** proračune slične onima koje rutinski obavlja ljudski mozak.

Neuronske mreže i konvencionalno računarstvo

- Računare eksplicitno programirate (pojednostavljeno):
 - ▣ Analiza problema kojeg treba rešiti.
 - ▣ Pisanje koda u nekom programskom jeziku.
- ANN uče na osnovu primera
 - ▣ Nema eksplicitnog opisa problema.
 - ▣ Nema potrebe za programerom.
 - ▣ *Neural computer* može da sam sebe adaptira tokom perioda treninga, na osnovu primera sličnih problema čak i bez postojanja željenog rešenja za svaki problem. Nakon uspešnog treninga, *neural computer* je sposoban poveže podatke o problemu sa rešenjem, ulaze sa izlazima, kao i to da je sposoban da ponudi rešenje za potpuno novi problem.
 - ▣ Sposobnost da generalizuje ili da rukuje sa nekompletnim podacima.

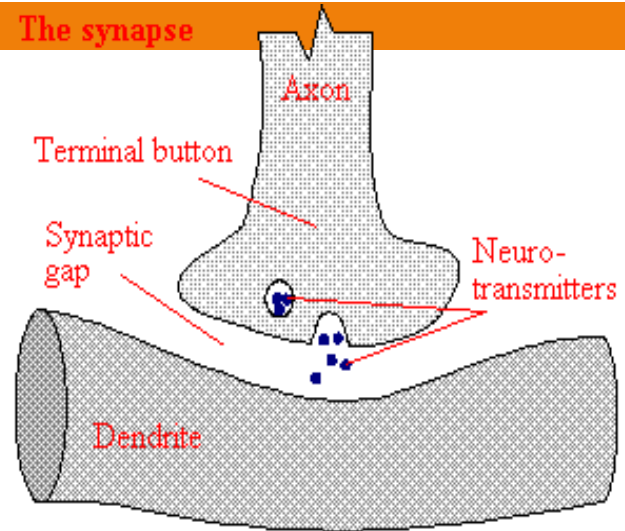
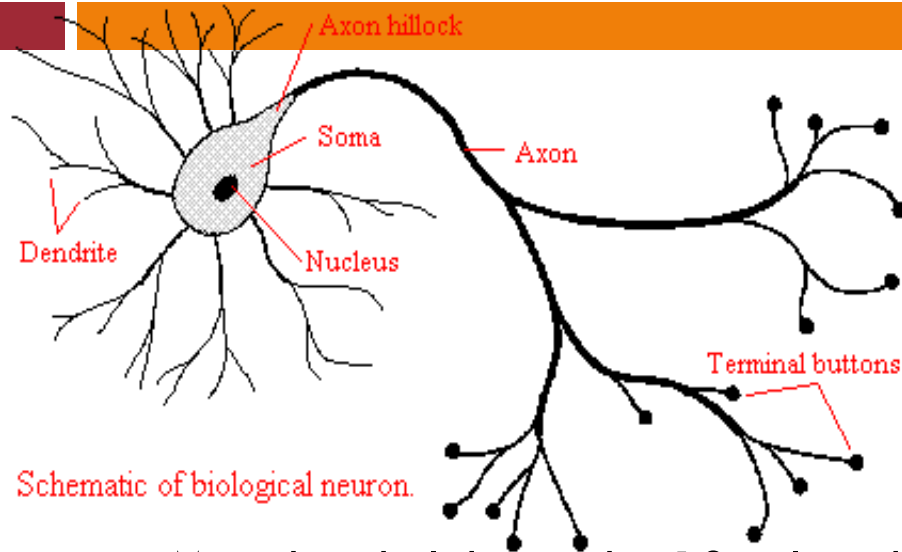
Šta ANN mogu a šta ne?

- ANN mogu da izračunaju bilo koju „izračunljivu“ funkciju, što može i bilo koji računar.
- U praksi, ANN su posebno korisne za **klasifikaciju** kao i ta probleme **aproksimacije funkcija** obično kada ekspertni sistemi ne mogu jednostavno da budu primenjeni.
- ANN se teško mogu uspešno primeniti za probleme koji se odnose na manipulaciju simbolima. Ne postoje metode za trening ANN koje magično mogu da kreiraju informacije koje nisu sastavni deo trening podataka.

Osobine ANN

- Veštačka neuronska mreža ima osobinu da **prepoznaje obrasce** u zadatoj kolekciji podataka i da **izgrađuje model** za te podatke.
- Oponašaju prirodne nervne sisteme životinja i ljudi na dva načina:
 - ▣ Akvizicija znanja se ostvaruje u procesu **obučavanja** neuronske mreže.
 - ▣ **Težine veza** između neurona u mreži (sinapse) se koriste za skladištenje znanja.

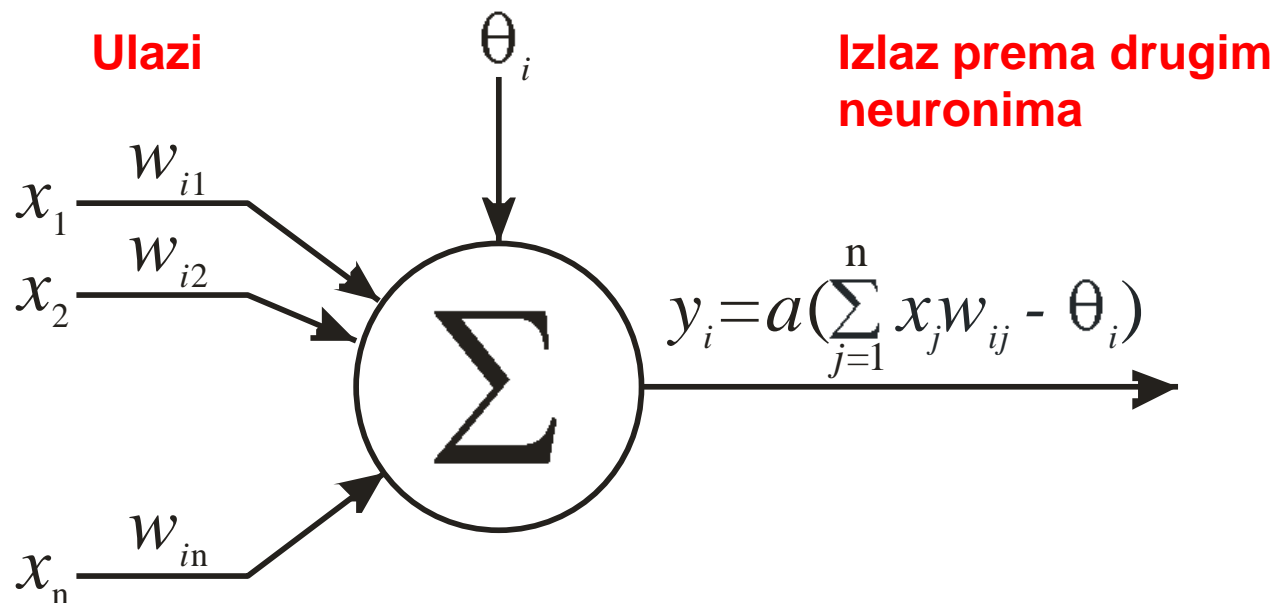
Biloški Neuron



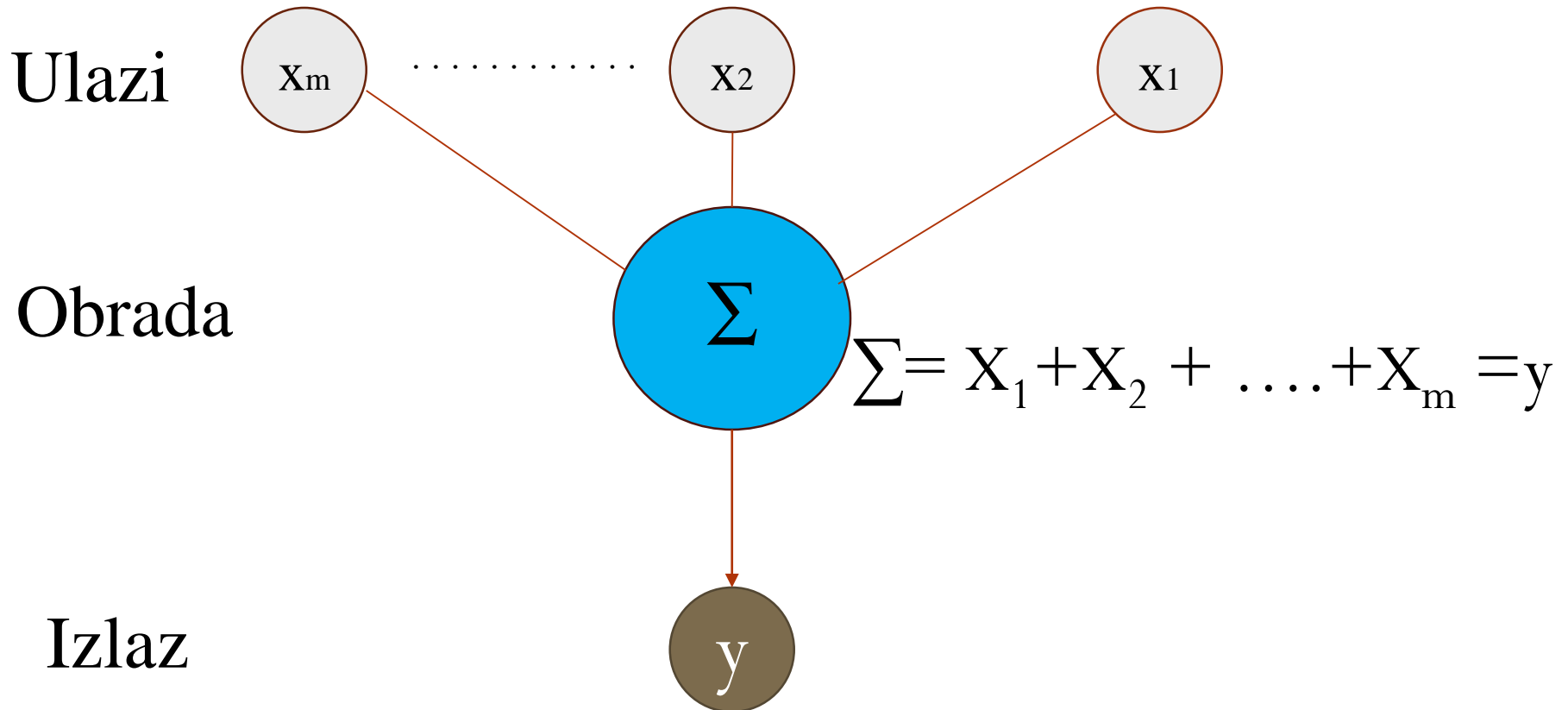
- Mozak je kolekcija oko 10 milijardi povezanih neurona. Svaki neuron je ćelija koja koristi biohemijske reakcije da prima, obrađuje i prosleđuje informacije.
- Svaka od terminalnih tačaka je povezana sa drugim neuronima preko malog međuprostora koji se zove sinapsa.
- Neuronsko dendritsko drvo je povezano sa hiljadama susednih neurona. Kada se neki do tih neurona aktivira, pozitivni ili negativni naboj stiže preko nekog od dendrita. Jačina svih dobijenih nabojja se dobija na osnovu vremenskog i prostornog sumiranja.

Model veštačkog neurona

- Veštački neuron predstavlja jednostavni procesni element koji obavlja relativno jednostavnu matematičku funkciju.
- Predstavlja imitaciju biološkog neurona.

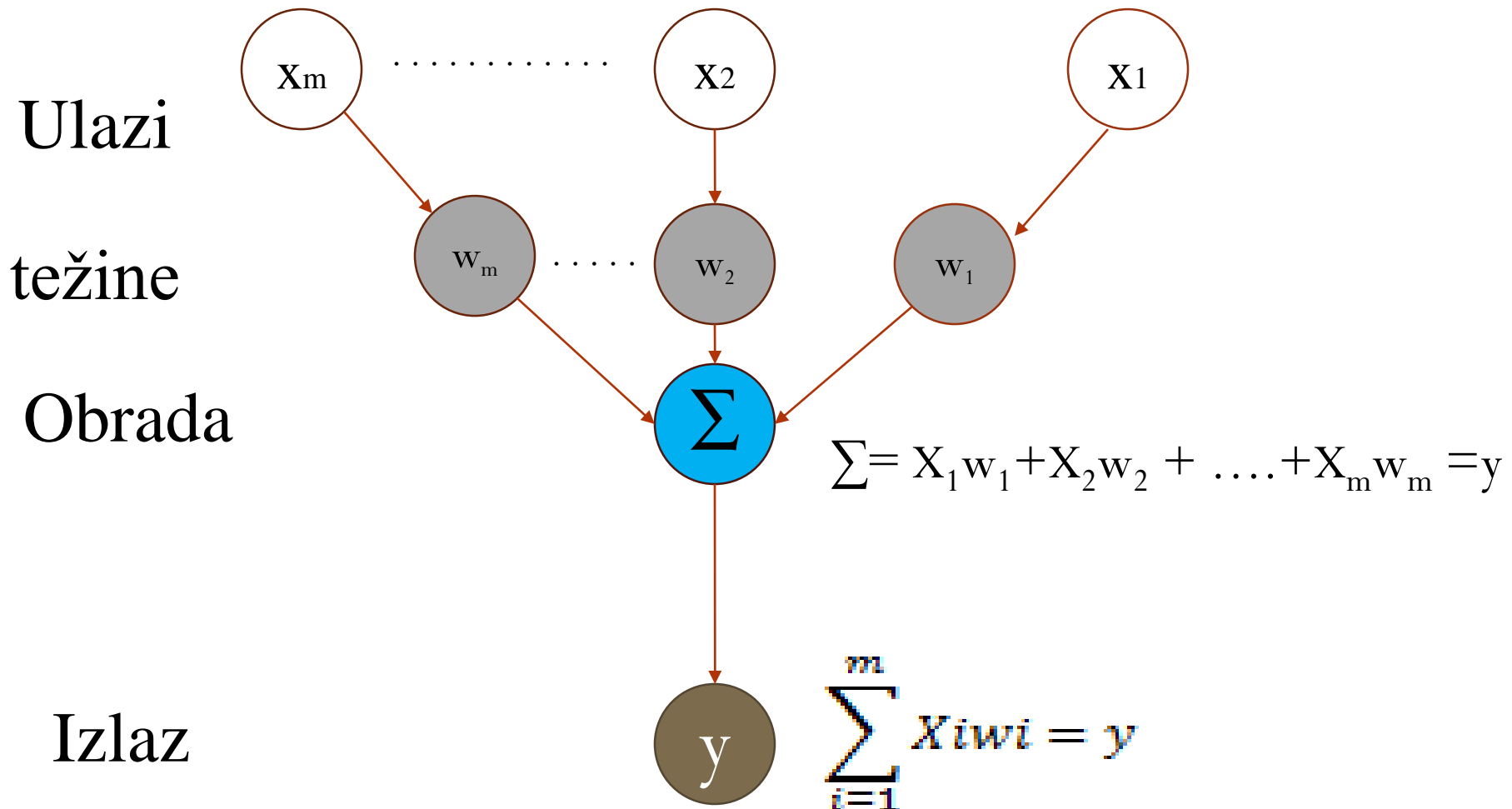


Kako neuron radi?



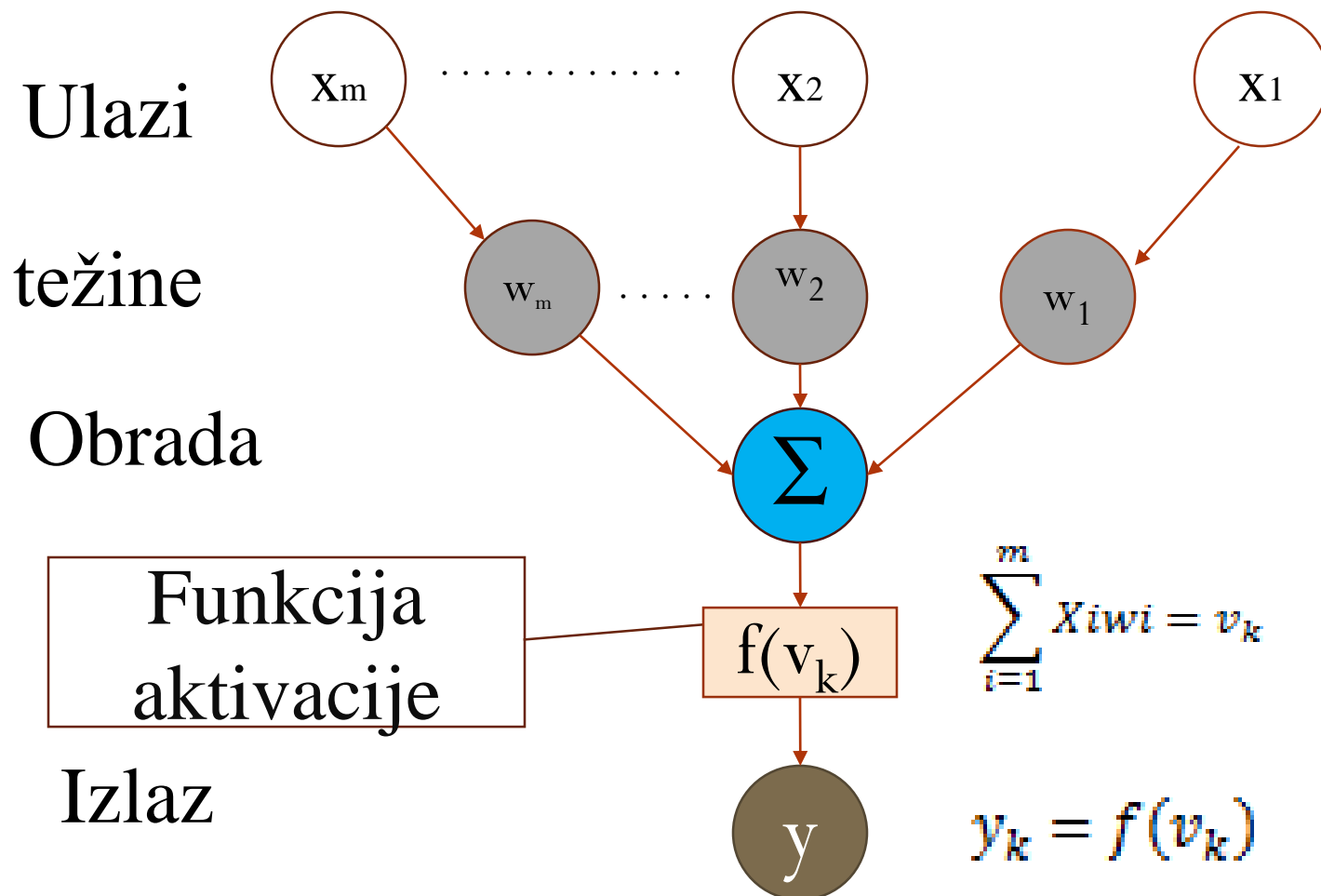
Kako neuron radi?

Nisu svi ulazi jednaki



Kako neuron radi?

Signal se ne prosleđuje sledećem neuronu doslovno



Primer rada neurona - Perceptron

□ Neuron sa funkcijom praga (Rosenblatt, 1957)

□ Pretpostavka:

▣ dve ulazne vrednosti, x i y za perceptron P .

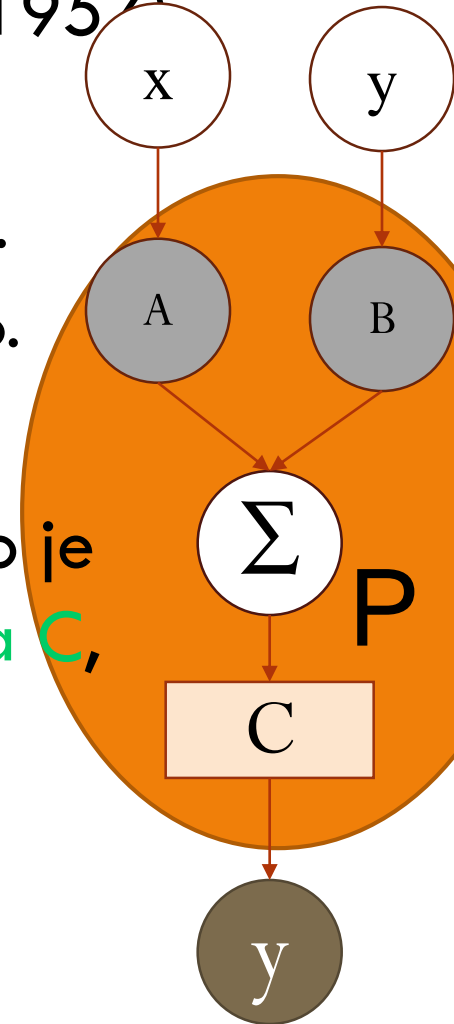
▣ Težine za x i y neka budu A i B respektivno.

□ Posledica: zbir na izlazu je $Ax + By$.

□ Pošto perceptron daje non-zero izlaz ako je izračunata suma veća od zadatog praga C ,

izlaz se može napisati kao:

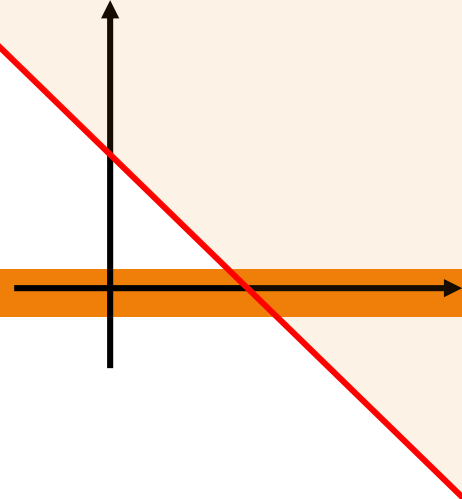
$$\text{Izlaz za } P = \begin{cases} 1 & \text{ako } Ax + By > C \\ 0 & \text{ako } Ax + By \leq C \end{cases}$$



Perceptron – primer rada

$$\text{Izlaz za } P = \begin{cases} 1 & \text{ako } Ax + By > C \\ 0 & \text{ako } Ax + By \leq C \end{cases}$$

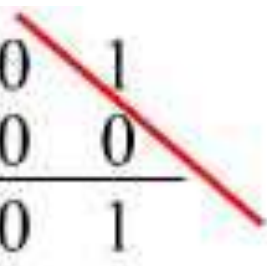
- $Ax + By > C$ i $Ax + By < C$ su dva regiona u xy ravni podeljeni linijom $Ax + By + C = 0$.
- Ako razmatramo ulaz (x, y) kao tačku u ravni, tada perceptron u stvari kaže u kom se regionu nalazi ta tačka.
- Takvi region, podeljeni jednom linijom su *linearно separabilni regioni*.



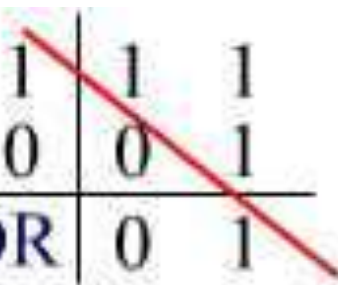
Perceptron – primer rada

- Ovakav rezultat je koristan!
- Neke logičke funkcije kao što su AND, OR i NOT su linearno separabilne odnosno mogu se izvršiti jednim perceptronom.
- Ilustracija (za 2D prostor):

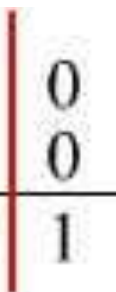
1	0	1
0	0	0
AND	0	1



1	1	1
0	0	1
OR	0	1

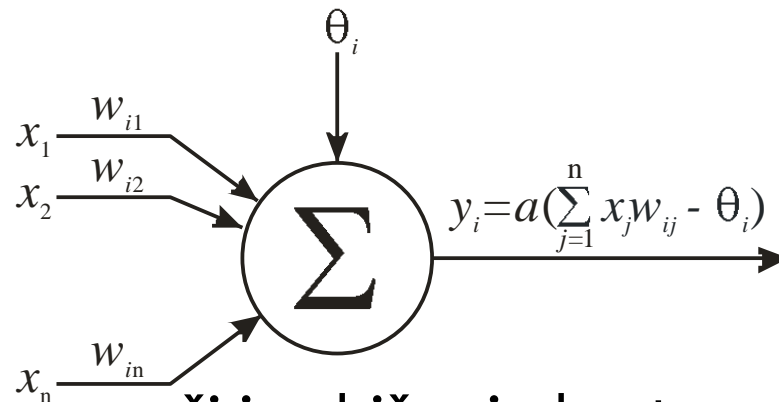


1	1	0
0	1	0
NOT	0	1



Ključni elementi za ANN

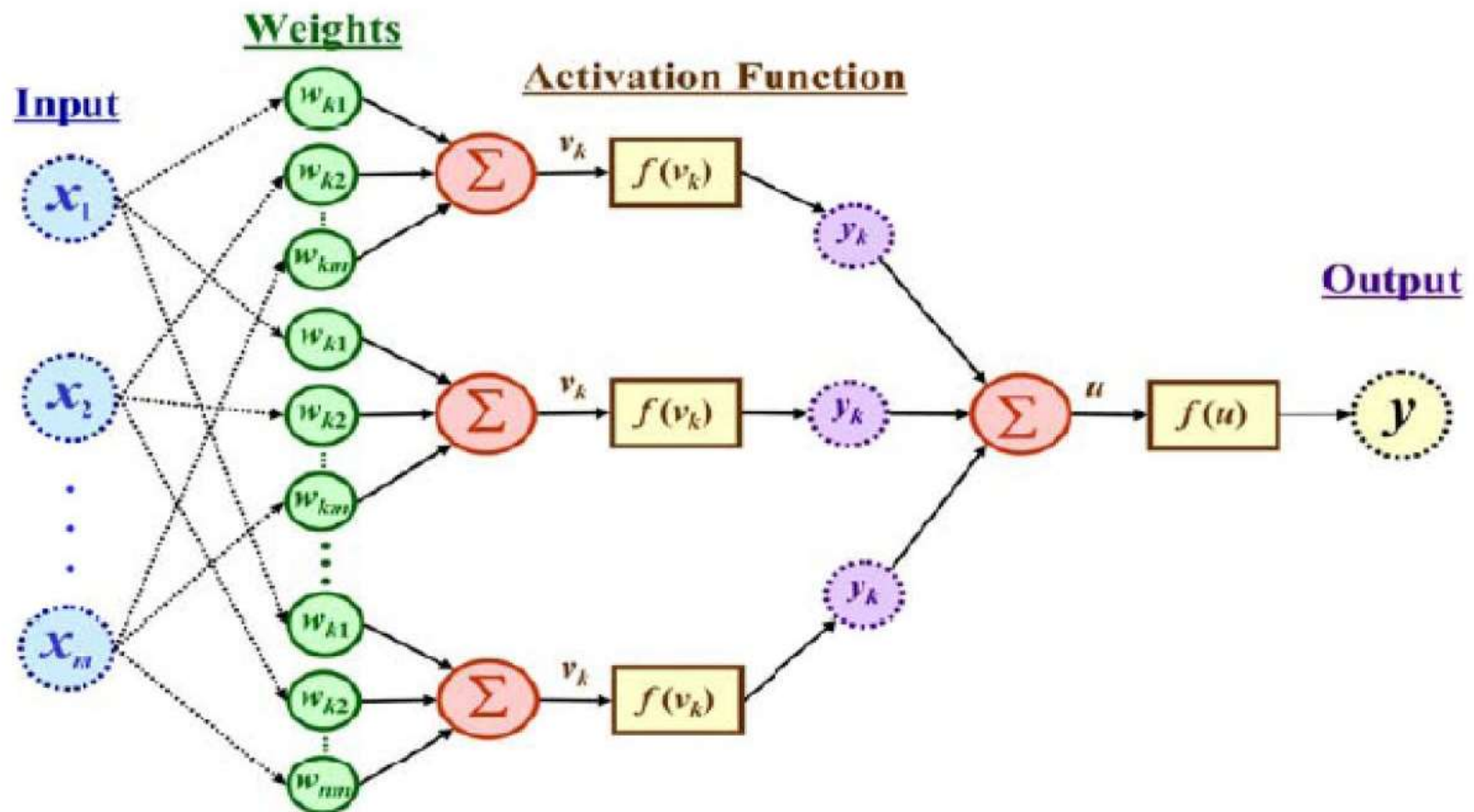
- *Neural computing* zahteva određeni broj veštačkih **neurona**, povezanih međusobno u **neuronsku mrežu**. Neuroni su organizovani u nivoe.



- Svaki neuron u mreži je obično jednostavna procesna jedinica koja dobija jedan ili više ulaza i produkuje izlaznu vrednost.
- Kod svakog neurona, svaki od ulaza ima asociranu **težinu** koja modifikuje jačinu svakog od ulaza.
- Neuron jednostavno računa zbir ulaza i izračunava izlaz na osnovu praga (bias) i aktivacione funkcije.

Neuronska mreža

- Izlaz je funkcija ulaza, uz uticaj težina i funkcija aktivacija.



Kako ANN radi?

- Postavi inicijalne vrednosti težina slučajno.
- Ulaz: tabela istinitosti za XOR
- **Do**
 - Čitaj ulaz (napr. 0, i 0)
 - Izračunaj izlaz (napr. 0.60543)
 - Uporedi ga sa očekivanim izlazom. (Diff= 0.60543)
 - Modifikuj težine u skladu s rezultatom.
- **Loop until** ispunjenje uslova
 - Uslov: određeni broj iteracija
 - Uslov: prag greške

Projektovanje ANN

– problemi koje treba rešiti

- Inicijalne težine (male slučajne vrednosti $\in [-1, 1]$)
- Funkcija aktivacije (kako se ulazi i težine kombinuju da produkuju izlaz?)
- Estimacija greške
- Podešavanje težina
- Broj neurona
- Reprezentacija podataka
- Veličina trening skupa

Aktivacione funkcije neurona

- Aktivacione funkcije generalno su nelinearne. Linearne funkcije su ograničene pošto je izlaz jednostavno samo proporcionalan ulazu.

- Linearna

$$a(x) = x$$

- Odskočna $a(x) = \begin{cases} 1, & x \geq 0 \\ 0, & x < 0 \end{cases}$

- Linearna odsečena (korak po korak)

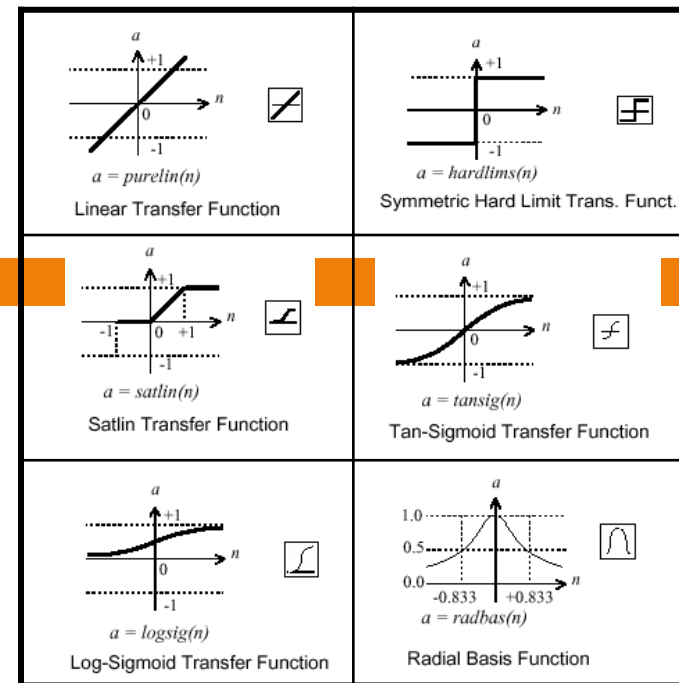
$$a(x) = \begin{cases} -1, & x \leq -1/k \\ kx, & -1/k < x < 1/k \\ 1, & x \geq 1/k \end{cases}$$

- Sigmoidalna (logistička)

$$a(x) = \frac{1}{1 + \exp(-kx)}$$

- Tangens hiperboločki

$$a(x) = \tanh(kx) = \frac{\exp(kx) - \exp(-kx)}{\exp(kx) + \exp(-kx)}$$



Estimacija greške

- Često se koristi srednje kvadratna greška (**root mean square error**) kao mera razlike između vrednosti

Podešavanje težina

- Nakon svake iteracije, podešavaju se težine da bi se minimizovala greška
 - Sve moguće težine
 - Back propagation

Back propagation

- Back-propagation je primer nadgledanog učenja
- Koristi se na svakom nivou da bi se minimizovala greška između odgovora nivoa i aktuelnih podataka
- Greška u svakom skrivenom sloju je srednja vrednost procenjene greške
- Skriveni nivoi mreže se treniraju na taj način

Back propagation

- N je neuron.
- N_w je jedna od N -ovih ulaznih težina
- N_{out} je N -ov izlaz.
- $N_w = N_w + \Delta N_w$
- $\Delta N_w = N_{out} * (1 - N_{out}) * N_{ErrorFactor}$
- $N_{ErrorFactor} = N_{ExpectedOutput} - N_{ActualOutput}$
- Ovo radi samo za poslednji nivo, pošto znamo aktuelni izlaz kao i očekivani izlaz.

Broj neurona

- Mnogo neurona:
 - Veća tačnost
 - Sporije
 - Rizik od over-fitting-a
 - Memorisanje, umesto razumevanja
 - Mreža je beskorisna za nove probleme.
- Malo neurona:
 - Niska tačnost
 - Nemogućnost da uče
- Optimalni broj!?
- **Arhitekture ANN!!**

Reprezentacija podataka

- Obično ulazni/izlazni podaci zahtevaju *pre-processing*
- Slike
 - Intenzitet piksela
- Tekst:
 - Uzorak

Veličina trening skupa

- Nema *one-fits-all* formule
- *Over-fitting* može da se desi ako se ne izabere “dobar” trening skup
- Šta čini “dobar” trening skup?
 - Primeri moraju da reprezentuju generalnu populaciju.
 - Primeri moraju da sadrže članove svih klasa.
 - Primeri u svakoj klasi moraju da sadrže širok skup varijacija ili efekata šuma.
- Veličina trening skupa je povezana sa brojem skrivenih neurona.

Arhitekture neuronskih mreža

- Neuronska mreža se formira od većeg broja neurona povezujući ulaze jednih za izlaze drugih.
- Sve neurone u mreži možemo razvrstati na:
 - ulazne,
 - izlazne i
 - skrivene.

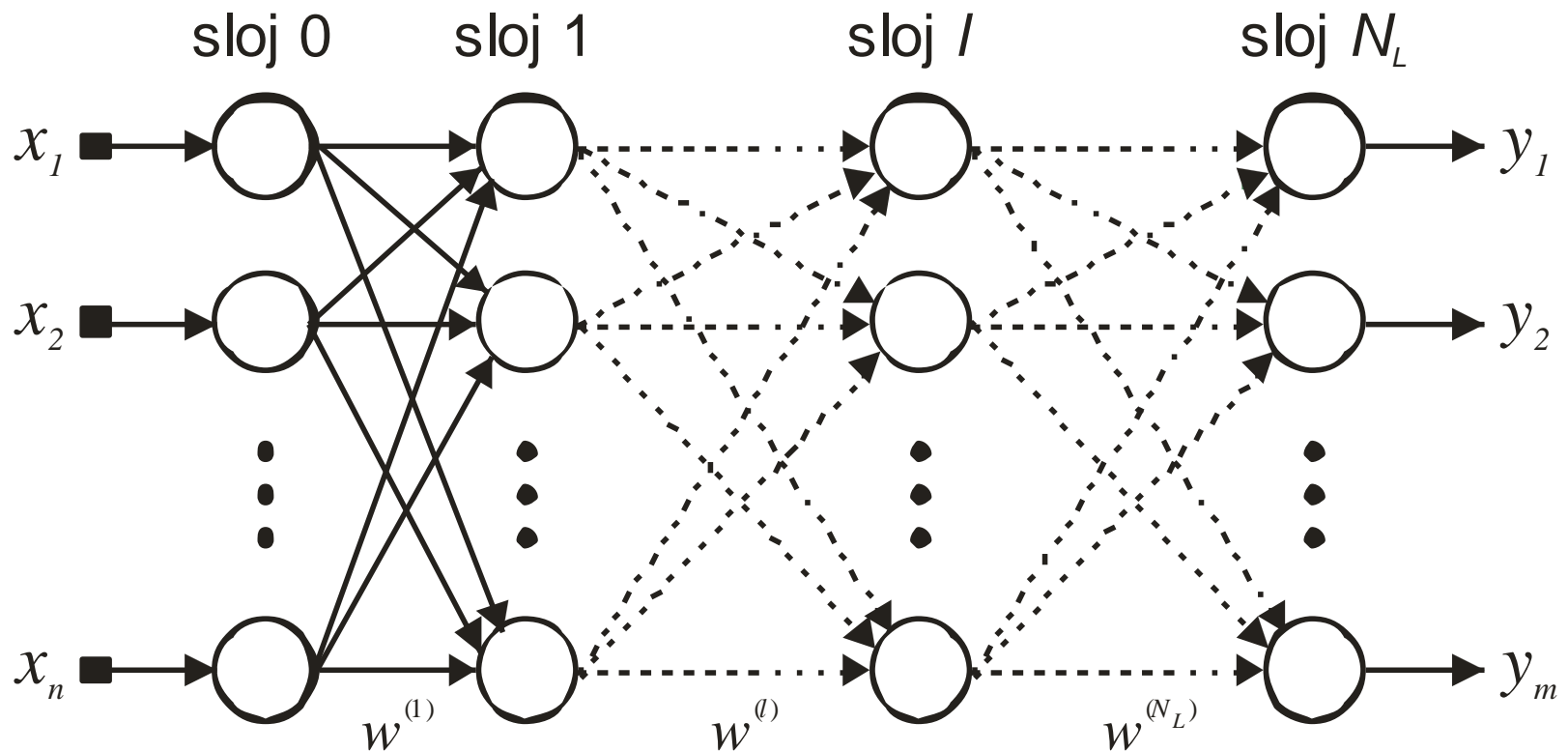
Arhitekture neuronskih mreža

- Po načinu povezivanja postoje dve klase neuronskih mreža:
 - ▣ sa povratnim vezama (rekurentne) i
 - ▣ bez povratnih veza (eng. *feed-forward*)
- Najpoznatije arhitekture rekurentnih mreža su:
 - ▣ potpuno povezane neuronske mreže i
 - ▣ ćelijske neuronske mreže.
- Najpoznatija arhitektura *feed-forward* mreža je:
 - ▣ slojevita neuronska mreža.

Slojevite neuronske mreže

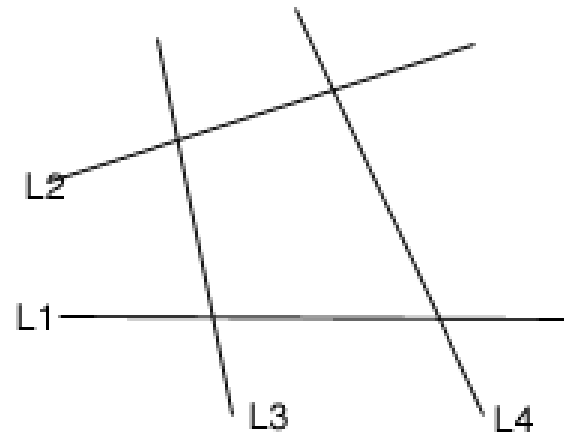
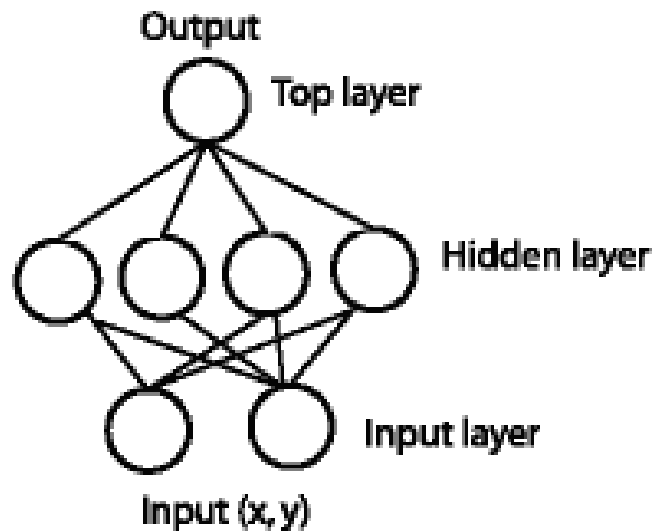
- Neuronu su organizovani u slojeve.
- Na ulaz jednog neurona iz sloja dovode se izlazi svih neurona iz prethodnog sloja, a njegov izlaz se vodi na sve neurone narednog sloja.
- Neuronu u prvom, ulaznom, sloju imaju samo po jedan ulaz - ulazi u mrežu.
- Izlazi neurona iz poslednjeg, izlaznog, sloja predstavljaju izlaze mreže.
- Ostali slojevi neurona se nazivaju skriveni slojevi.

Slojevite neuronske mreže



Primer primene slojevite ANN

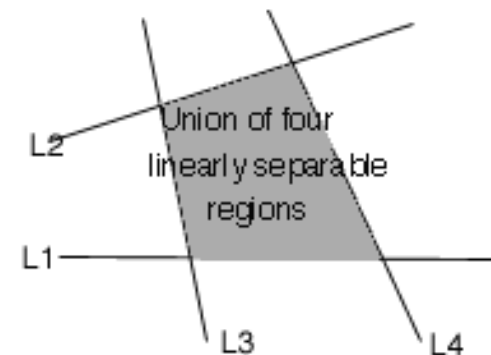
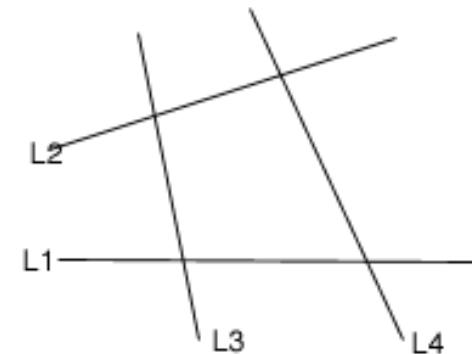
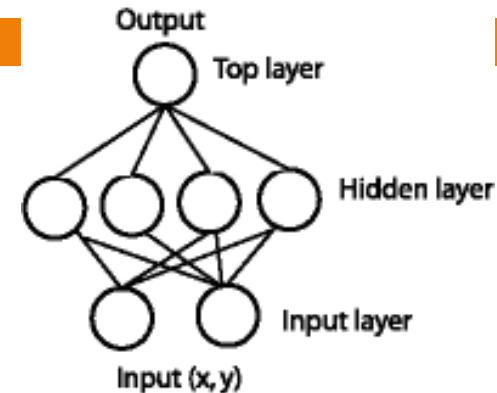
- Podsetnik: perceptron i linearno separabilnost



- ANN sa jednim skrivenim slojem sa 4 perceptrona
- Ulaz (x, y) se klasifikuje u 4 para linearno separabilnih regiona, svaki od njih ima svoju liniju koja razdvaja regione
- **Feed-forward ANN**

Primer primene slojevite ANN

- Top perceptron izvršava logičku funkciju koja na osnovu ulaza iz skrivenog sloja klasifikuje ulaze u mrežu u dva regiona koja mogu da budu linearno neseparabilni!
- Variranjem broja čvorova u skrivenom sloju, broja nivoa, broja ulaznih i izlaznih čvorova, može se vršiti klasifikacija tačaka iz prostora proizvoljnih dimenzija u proizvoljan broj grupa.
- Posledica: *feed-forward* mreže se uglavnom koriste za klasifikaciju.
- (slika, AND operacija, tj unija regiona)



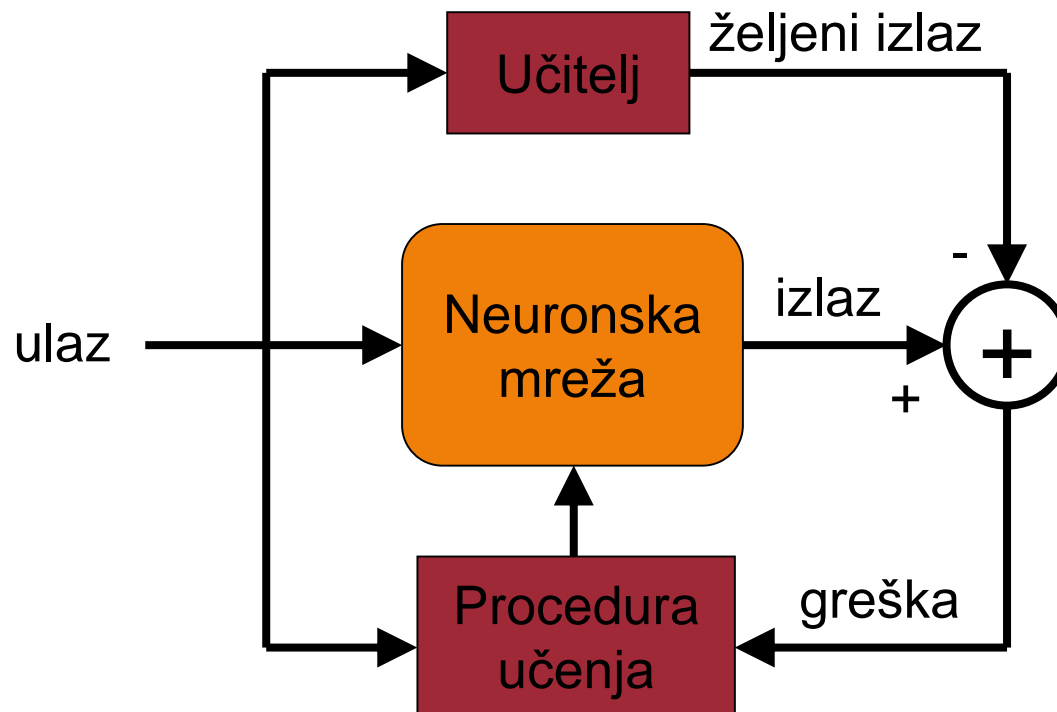
Obučavanje neuronskih mreža

- Ponašanje neuronske mreže zavisi od:
 - arhitekture mreže,
 - broja neurona i
 - težina veza između neurona.
- Obučavanje neuronske mreže se najčešće svodi na promenu težina veza između neurona.

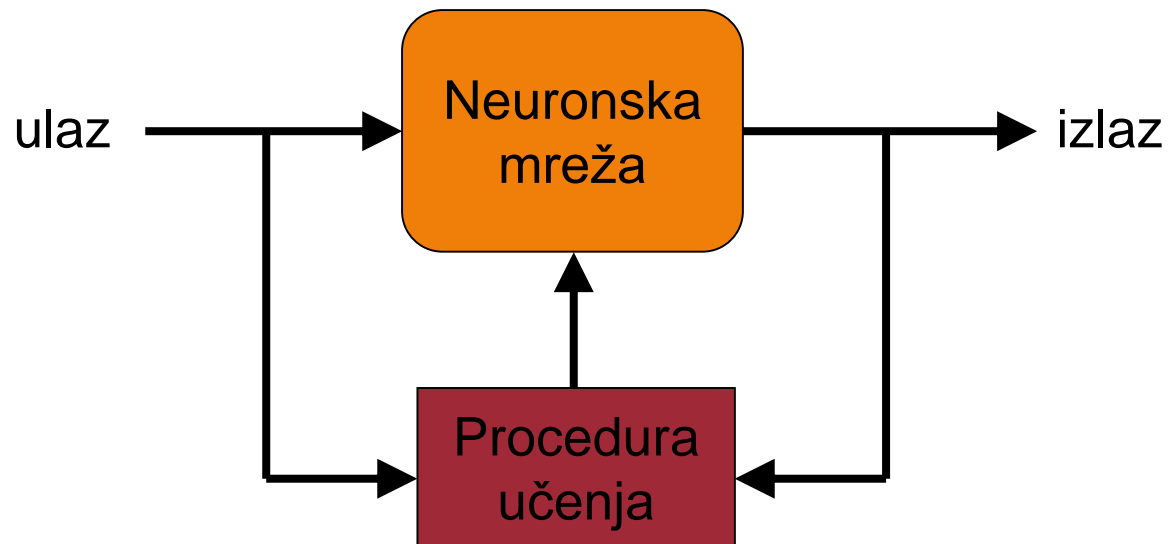
Klasifikacija načina za obučavanje neuronskih mreža

- U zavisnosti od toga da li su poznati očekivani izlazi neuronske mreže imamo:
 - ▣ nadgledano (eng. *supervised*) i
 - ▣ nenadgledano (eng. *unsupervised*) obučavanje.
- U zavisnosti od toga da li se pri obučavanju menjaju i neki arhitekturni elementi mreže (broj neurona) imamo:
 - ▣ statičko i
 - ▣ dinamičko obučavanje.

Nadgledano obučavanje



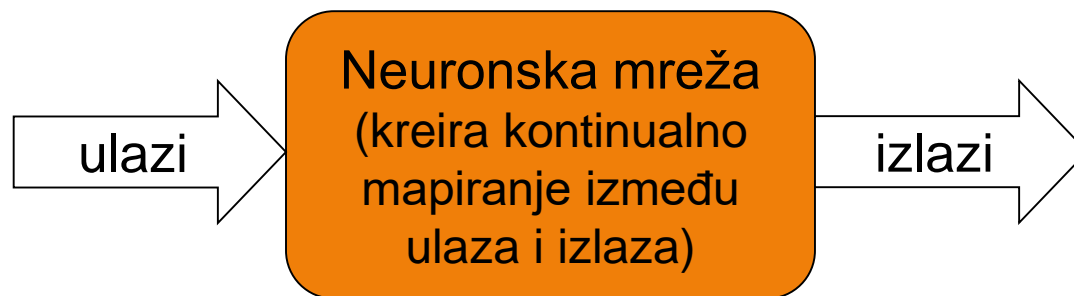
Nenadgledano obučavanje



Tipične primene neuronskih mreža

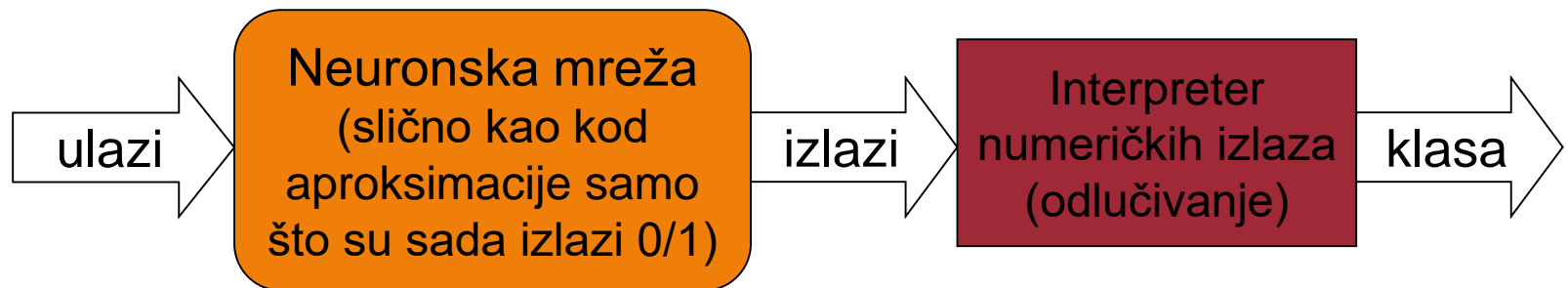
- Klasifikacija
 - ▣ dijagnostika, prepoznavanje uzoraka, prepoznavanje govora
- Aproksimacija funkcija
 - ▣ modelovanje procesa, upravljanje
- Predikcija vremenskih serija
 - ▣ finansijska previđanja, modelovanje dinamičkih sistema
- Predprocesiranje
 - ▣ klasterizacija, vizuelizacija podataka, ekstrakcija podataka

Aproksimacija funkcija



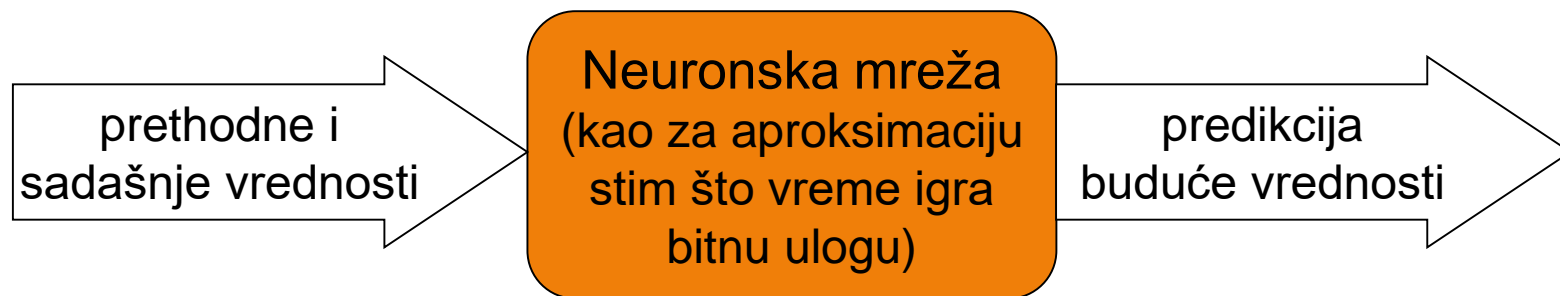
- $Y = f(X)$, gde je:
 - ▣ X skup kontinualnih numeričkih ulaza,
 - ▣ Y skup kontinualnih numeričkih izlaza i
 - ▣ $f()$ nepoznata zavisnost između ulaza i izlaza.
- Neuronska mreža mora da aproksimira $f()$ kako bi davala odgovarajuće izlaze za svaki ulazni skup.

Klasifikacija



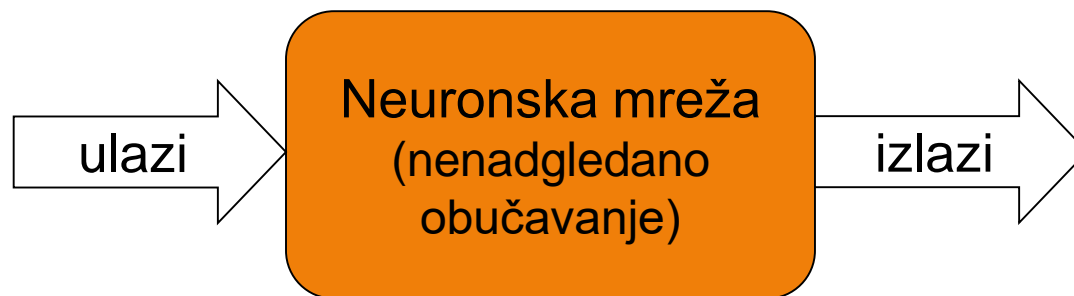
- Svodi se na aproksimaciju diskretnih funkcija.
- Interpreter određuje klasu za zadati uzorak na osnovu numeričkih vrednosti pripadnosti uzorka svakoj od klasa.

Predikcija vremenskih serija



- Predikcija vremenskih serija predstavlja dinamičku aproksimaciju funkcija.
- Kao ulazi se koriste prethodne vrednosti i tekuća vrednost, a kao izlaz se dobija predikcija buduće vrednosti.

Predprocesiranje (*Data Mining*)



- Nisu poznati očekivani izlazi mreže, već se želi ekstrakcija podataka iz ulaza. Koristi se za:
 - ▣ Klasterizaciju
 - ▣ Kompresiju
 - ▣ PCA (Principal Component Analysis)

PITANJA?

Dileme?

Komentari?

