



# Umjetne neuronske mreže

Zavod za elektroniku, mikroelektroniku,  
računalne i inteligentne sustave

---

v4.4.2, 27. svibnja 2008.

© Bojana Dalbelo Bašić

© Marko Čupić

© Jan Šnajder



- Automatiziranu obradu podataka danas uglavnom rade digitalna računala.
- Ipak, još je uvijek daleko više podataka čija obrada nije automatizirana. Te podatke obrađuju ...



... živčani sustavi  
živih organizama!



- Razvoj jedne grane računarstva motiviran je razmatranjem prevladavajućeg načina obrade podataka u svijetu u kojem živimo.
- Tražimo drugačiji koncept obrade podataka koji bi bio sličniji funkcioniranju biološkog mozga.
- **A.I.** - sustav koji uspješno oponaša rad mozga bio bi inteligentan.



## 1. Uvod u neuro-računarstvo

Povijesni pregled • Pravci AI • Definicija i vrste • Primjena • Učenje • Živčani sustav i biološki neuron • Model neurona i vrste • Arhitektura mreža

---

## 2. Procesni elementi

Perceptron • Pravilo perceptrona • Primjer • Učenje booleovih funkcija • Linearna odvojivost • XOR problem • Višeslojna mreža perceptrona • Linearna regresija • Srednja kvadratna pogreška • Gradijentni spust • LMS algoritam

---

## 3. Backpropagation algoritam

BACKPROPAGATION algoritam i interpretacija skrivenog sloja • Primjeri uporabe neuronskih mreža • Primjer sustava za raspoznavanje

---

# 1.

## Motivacija razvoja neuro-računarstva

- Istraživanja u neurofiziologiji i kognitivnoj znanosti upućuju: mozak se sastoji od **velikog broja neurona** koji rade **paralelno**.
- Najsličniji model: računalni sustav u kojem brojni procesni elementi podatke obrađuju paralelno.
- Paradigma:
  - umjetna neuronska mreža - UNM**  
[artificial neural network – ANN]
- Područje koje se bavi tim aspektom obrade:
  - neuro-računarstvo**
- Grane računarstva iz skupine tzv. '**mekog računarstva**' (engl. soft computing).

# 1.

## Usporedba ljudskog mozga i računala

- Arhitektura i organizacija mozga gotovo je potpuno različita od arhitekture konvencionalnih računala koja su danas u širokoj uporabi (von Neumannova računala):

atribut	mozak	računalo
gradbeni element	neuron (>100 vrsta)	logička vrata
brzina prijenosa	2 ms ciklus	ns ciklus
broj procesora	oko $10^{11}$	$\leq 32^*$
broj veza	$10^3 - 10^4$	$\leq 32^*$
način rada	serijski, paralelno	serijski
signali	analogni	digitalni
informacije	ispravne/neispravne	ispravne

# 1.

## ...mozga i super-računala?



- *Earth Simulator* (NEC - Kanazawa, Japan)
- 640 čvorova sa po 8 vektorskih procesora = 5120 procesora
- 5x brži od dosadašnjeg prvaka (IBM ASCI White, 8192 procesora)
- <http://www.top500.org> (koristi se LINPACK benchmark test)

# 1.

## Povijesni pregled neuro-računarstva

- **1943.**

McCulloch i Pitts (MIT) – matematički model neurona (Automata Theory). Procesna moć ondašnjih računala je šarmantno slaba - prva praktička ostvarenja tek u kasnim '70 (pojava LSI računala).

- **1948.**

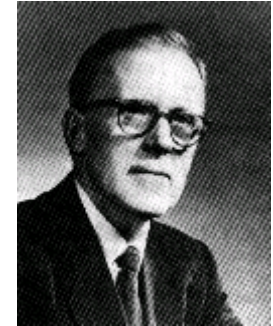
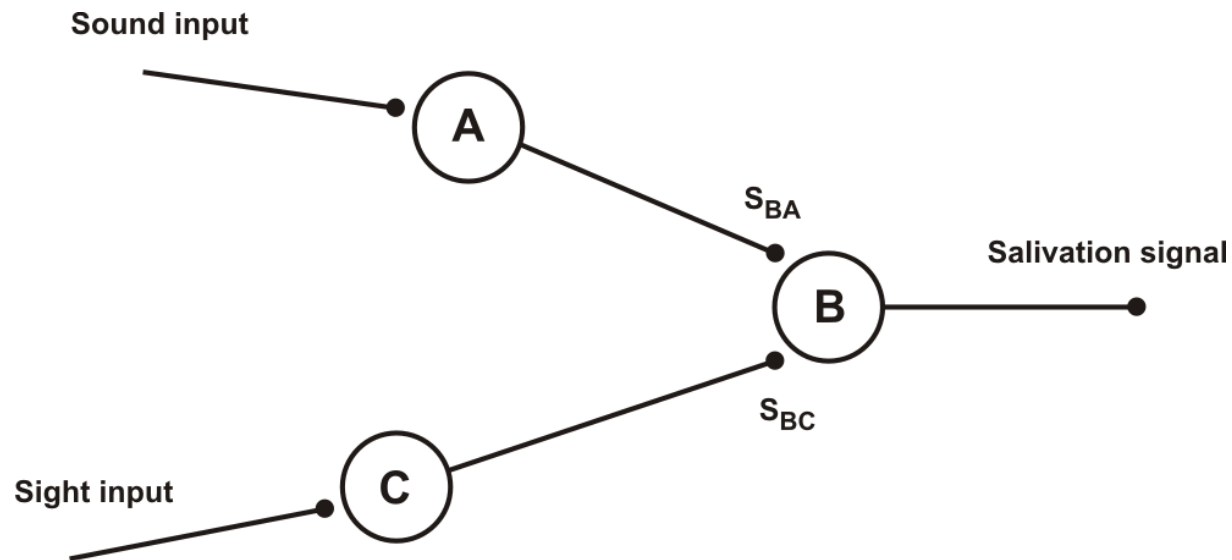
Wiener - kibernetika

- **1949.**

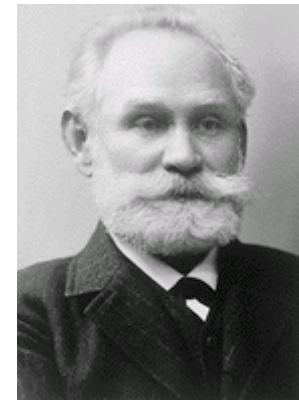
Hebb - učenje kao metabolička promjena u neuronima - temelje razvoja mehanizama učenja mreža (*Hebbovo pravilo*).

# 1.

## Hebbovo učenje



Donald Hebb



Ivan Pavlov

- Hebbovo pravilo izravno tumači Pavlovljev uvjetni refleks



# 1.

## Povijesni pregled neuro-računarstva

- **1958.**

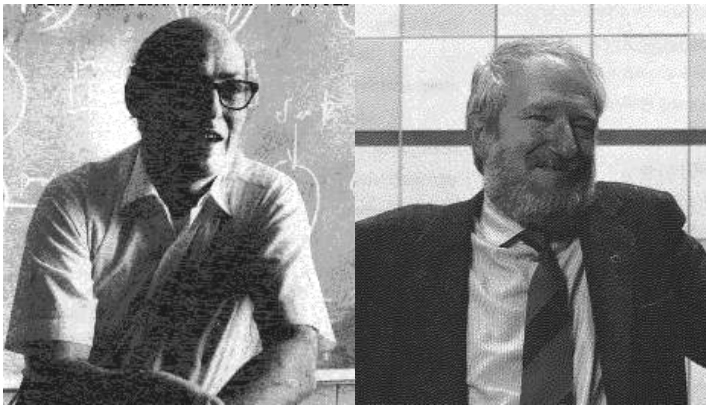
Rosenblatt – Perceptron i “Pravilo perceptrona”.

- **1969.**

Entuzijazma nestaje: *'Credit assignment problem'* ...

# 1.

## Povijesni pregled neuro-računarstva



### 'Credit assignment problem'

Kako odrediti koliko je svaki procesni element odgovoran za pogrešku mreže na izlazu?!

*Minsky i Papert: 'Perceptrons', 1969.*

- Zaključuju: nema mnogo nade da će se problem učenja višeslojne mreže ikada riješiti - **ne možemo riješiti probleme koji nisu linearno odvojivi.**
- ANN na 20-tak godina postaju *scientia non grata*
- Daljnja istraživanja u sjeni *mainstream AI* (Grossberg, Fukushima, Kohonen, Aleksander).

# 1.

## Povijesni pregled neuro-računarstva

- **1982.**

Renesansa ANN: Hopfield uvodi pojam **energije** mreže – fizičari zainteresirani (fizika materijala),

- **1986.**

Pronađeno rješenje za '*credit assignment problem*' – **back error propagation (BACKPROPAGATION)**.

- Više autora BP: otkriven (Werbos), ponovo otkriven (Parker), naposljetku još jednom ponovo otkriven i populariziran (Rumelhart, Hinton i Williams, 1986.).

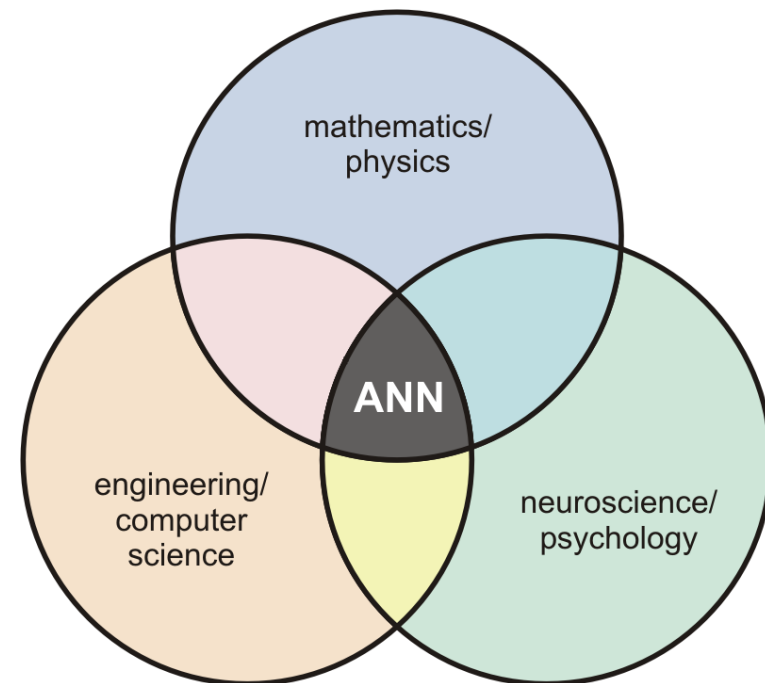
- **'90. -**

Porast istraživanja u tom području (daljnji razvoj matematike, razvoj informatičke tehnologije, 'conductive scientific Zeitgeist').

# 1.

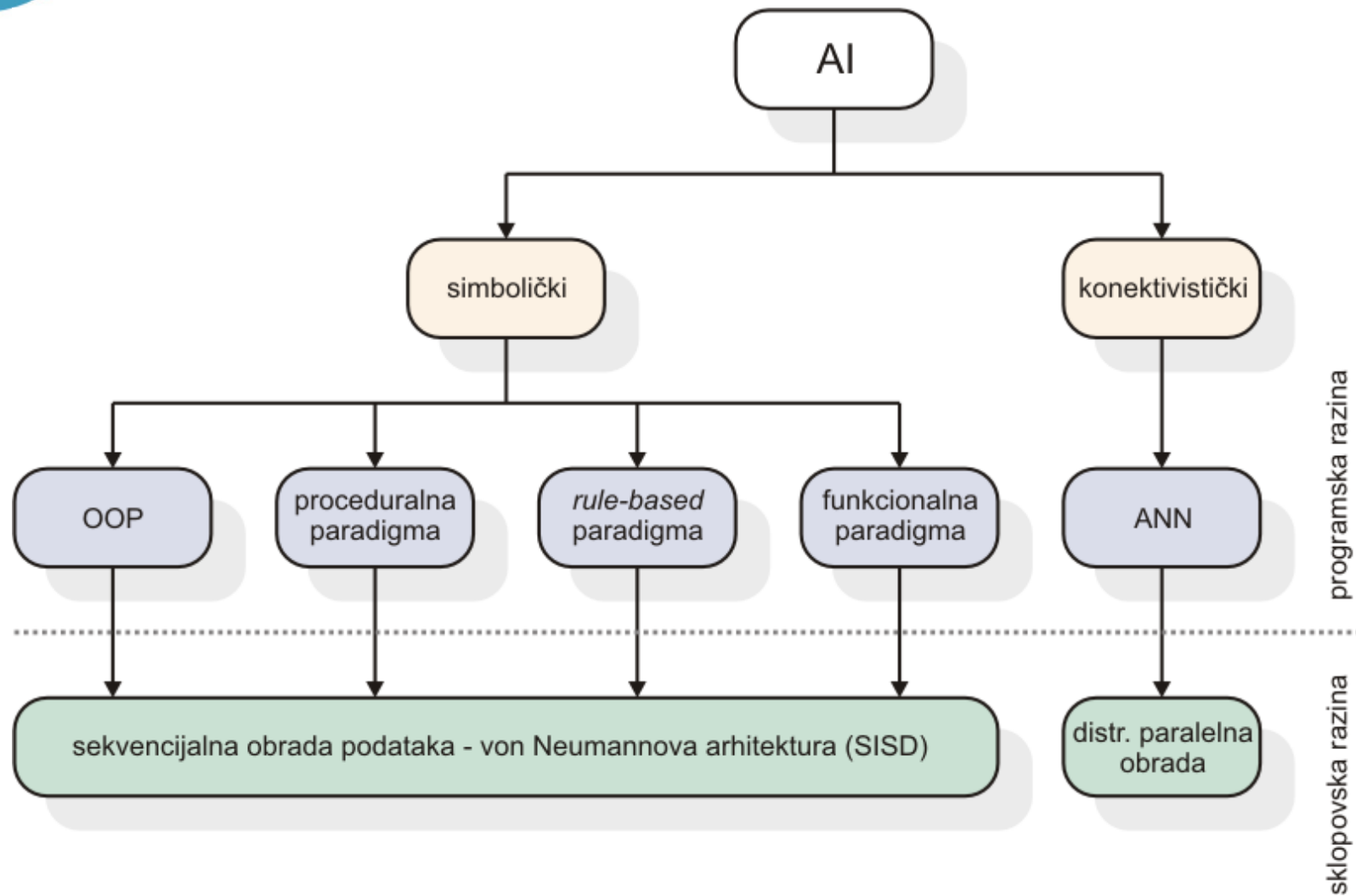
## Pristupi neuronskim mrežama

- Pojedini pristupi ANN imaju drugačije ciljeve istraživanja i dovode do različitih ideja njihove uporabe:
- **Psihologija i neurologija:** modeliranje mozga i biološkog postupka učenja,
- **Računarske znanosti:** paralelna distribuirana obrada u svrhu klasifikacije, asocijativna memorija i optimizacija,
- **Matematika i fizika:** proučavanje UNM u okviru statističke mehanike, teorije automata i nelinearnih dinamičkih sustava.



# 1.

## Pravci razvoja umjetne inteligencije



# 1.

## Pravci razvoja umjetne inteligencije

- Od prvih dana razvoja umjetne inteligencije (rane '50) postoje **dva pristupa** razvoju inteligentnih sustava:
  - prvim pristupom nastoji se znanje iz neke domene obuhvatiti skupom atomičkih semantičkih objekata (simbola) i zatim činiti manipulacija tih simbola pomoću algoritamskih pravila,
  - drugi pristup temelji se na izgradnji sustava arhitekture slične arhitekturi mozga koji, umjesto da ga se programira, uči samostalno na temelju iskustva.
- Različita područja zahtijevaju različite pristupe.
- Simbolički pristup je dobar u mnogim područjima (osobito isplativ postao je razvojem ekspertnih sustava), ali nije ispunio rana extravagantna obećanja.
- Neuspjeh leži u pogrešnoj pretpostavci da je svako znanje moguće formalizirati i da je mozak stroj koji podatke obrađuje formalnim pravilima.

# 1.

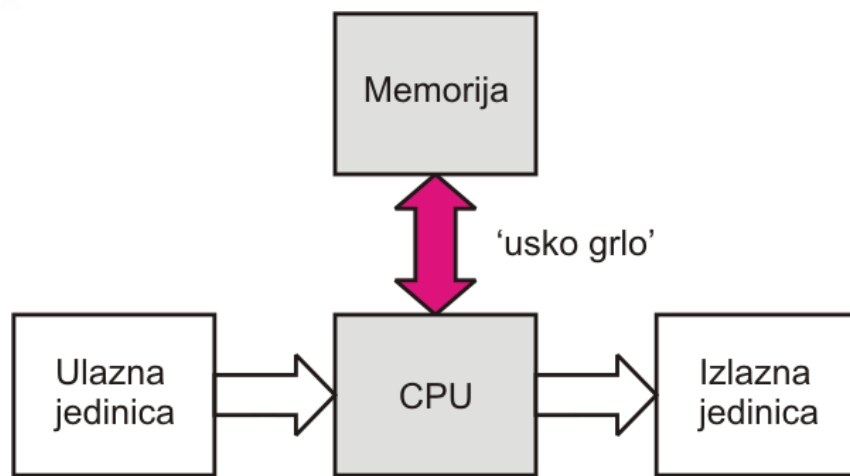
## Konektivistički pristup

- Smolensky ('88) eksplicitno razlučuje koje je znanje moguće formalizirati, a koje nije, čineći podjelu između:
  - kulturalnog (javnog) znanja i
  - privatnog (intuitivnog) znanja.
- Prema konektivističkom shvaćanju, intuitivno se znanje **ne može** obuhvatiti skupom formalnih pravila.
- Mnogi su svakodnevni zadaci previše složeni za simboličko predočavanje, npr. raspoznavanje uzoraka...
  - **Majku** možemo prepoznati u **0.1 s**
  - Neuroni u mozgu pale svake **ms**
  - U seriji, dakle, pali **samo 100** neurona
  - Očigledno **paralelna** obrada!



# 1.

## Von Neumannovo računalo



- 1946. von Neumann/Burks/Goldstine.
- Podaci i instrukcije programa pohranjeni su zajedno u jednoj memoriji i međusobno se ne razlikuju.
- Postoji samo jedan tok podataka i jedan tok instrukcija (SISD).
- Svi podaci i sve operacije prolaze tim tokom - von Neummanovo 'usko grlo' (Backus 1978.).
- Fizičko, ali i mentalno ograničenje.



# 1.

## Umjetne neuronske mreže vs. von Neumann

- Von Neumannova računala odlična su za simbolički pristup jer se problemi rješavaju algoritamski na sekvencijalnom stroju.
- Umjetne neuronske mreže su distribuirani i paralelni sustavi.
- Bitne karakteristične razlike dviju paradigmi/arhitektura:

von Neumann	ANN
Unaprijed detaljno opisujemo algoritam kroz korake	Uči samostalno ili s učiteljem
Samo se precizni podaci adekvatno obrađuju	Podaci mogu biti nejasni (šum) ili neizraziti
Funkcionalnost ovisi o svakom elementu	Obrada i rezultat ne ovisi mnogo o jednom elementu
Eksplisitna veza: sematički objekt - sklopovi računala	Implicitno znanje (teška interpretacija)

# 1.

## Implementacija umjetne neuronske mreže

- Von Neumannovo računalo možemo koristiti kao **emulator** neuronske mreže - samo programska implementacija UNM.
- Danas se istražuju druge pogodnije arhitekture za implementaciju ANN - peta generacija računala.

# 1.

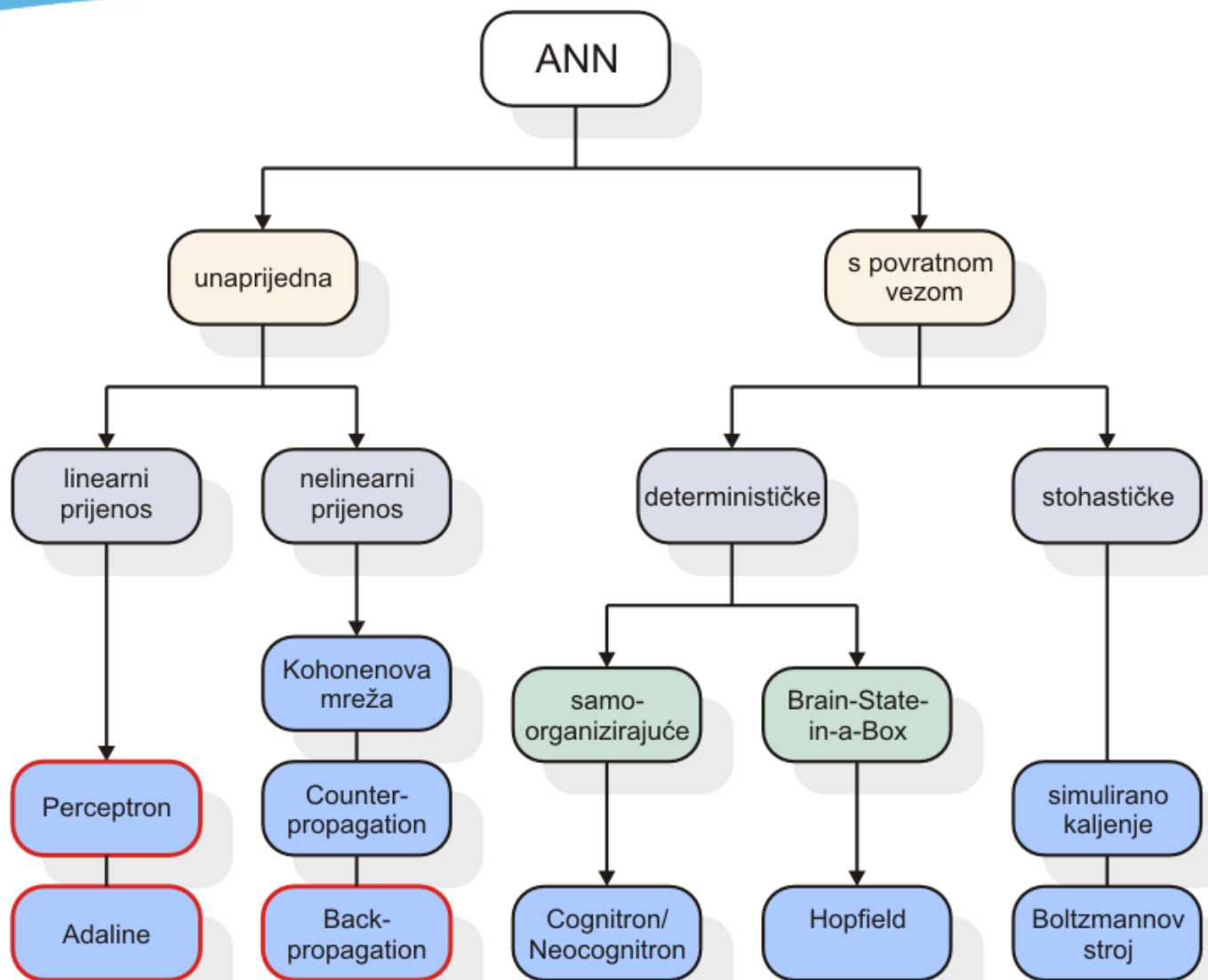
## Umjetna neuronska mreža - definicija

- U širem smislu: umjetna replika ljudskog mozga kojom se nastoji simulirati postupak učenja i obrade podataka.
- Zapravo dosta klimava analogija.
- **Neuronska mreža** ► skup međusobno povezanih jednostavnih procesnih elemenata (jedinica, čvorova) čija se funkcionalnost temelji na biološkom neuronu i koji služe **distribuiranoj paralelnoj** obradi podataka.



# 1.

## Vrste umjetnih neuronskih mreža



# 1.

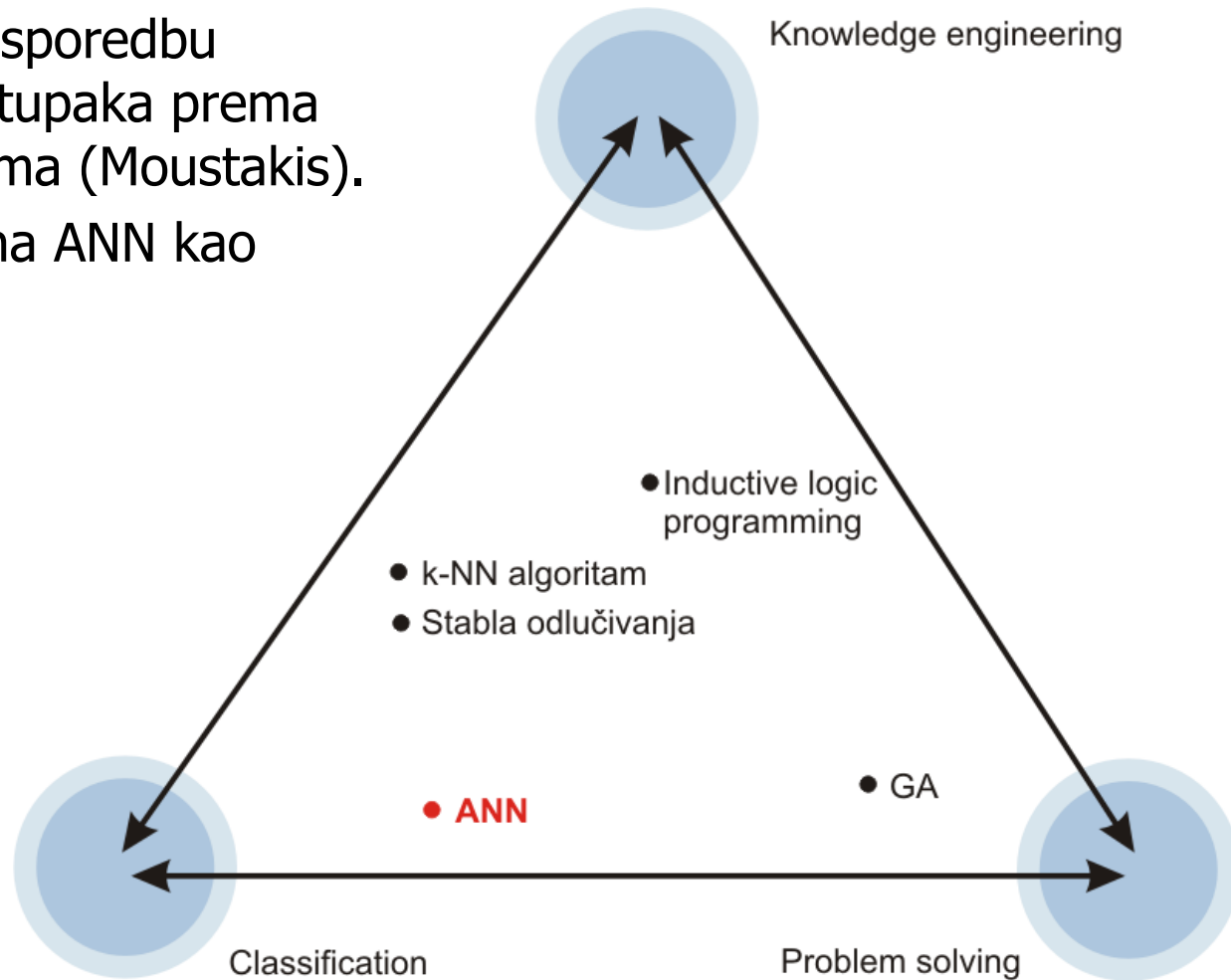
## Primjena umjetnih neuronskih mreža

- Odlično rješavaju probleme **klasifikacije i predviđanja** – sve probleme kod kojih postoji složena (nelinearna) veza ulaza i izlaza.
- Neke osobitosti:
  - Dobre u procjeni nelinearnosti. Mogu raditi s nejasnim ili manjkavim podacima (*sensor data*). Robusne na pogreške u podacima. Rade s velikim brojem varijabli i parametara. Prilagodljive okolini. Sposobne učiti. ...
- Najčešći zadaci:
  - Raspoznavanje uzoraka. Obrada slike i govora. Problemi optimizacije. Nelinearno upravljanje. Obrada nepreciznih i nepotpunih podataka. Simulacije. Prognoza vremenskih serija. ...

# 1.

## Primjena umjetnih neuronskih mreža

- Slika pokazuje usporedbu različitih ML postupaka prema različitim zadacima (Moustakis).
- Ističe se primjena ANN kao klasifikatora.



# 1.

## Učenje umjetne neuronske mreže

- Dvije faze rada s ANN:
  - Faza učenja (treniranja) i
  - Faza obrade podataka (iskorištavanja, eksploatacije).
- **Učenje** ► iterativan postupak predočavanja ulaznih primjera (uzoraka, iskustva) i eventualno očekivana izlaza pri čemu dolazi do postupnog prilagođavanja težina veza neurona
- Jedno predočavanje svih uzoraka naziva se epohom
- Razlikujemo:
  - Pojedinačno učenje (on-line)
    - za svaki primjer podešavamo faktore
  - Grupno učenje (batch)
    - cijela epoha u jednoj iteraciji
- Znanje o izlazu kao funkciji ulaza pohranjeno je implicitno u težinama veza neurona

# 1.

## Učenje umjetne neuronske mreže

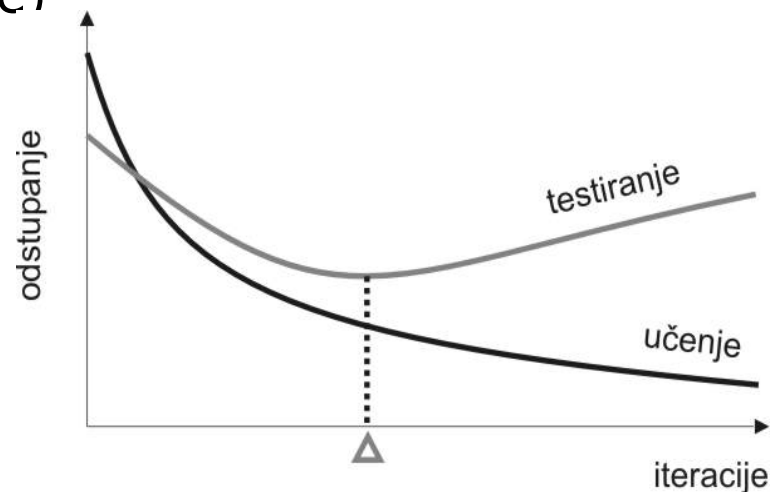
- Dva načina učenja:
  - Učenje s učiteljem (*supervised learning*) – postoje primjeri oblika (*ulaz, izlaz*)
  - Učenje bez učitelja (*unsupervised learning*) – izlaz je a priori nepoznat



# 1.

## Učenje umjetne neuronske mreže

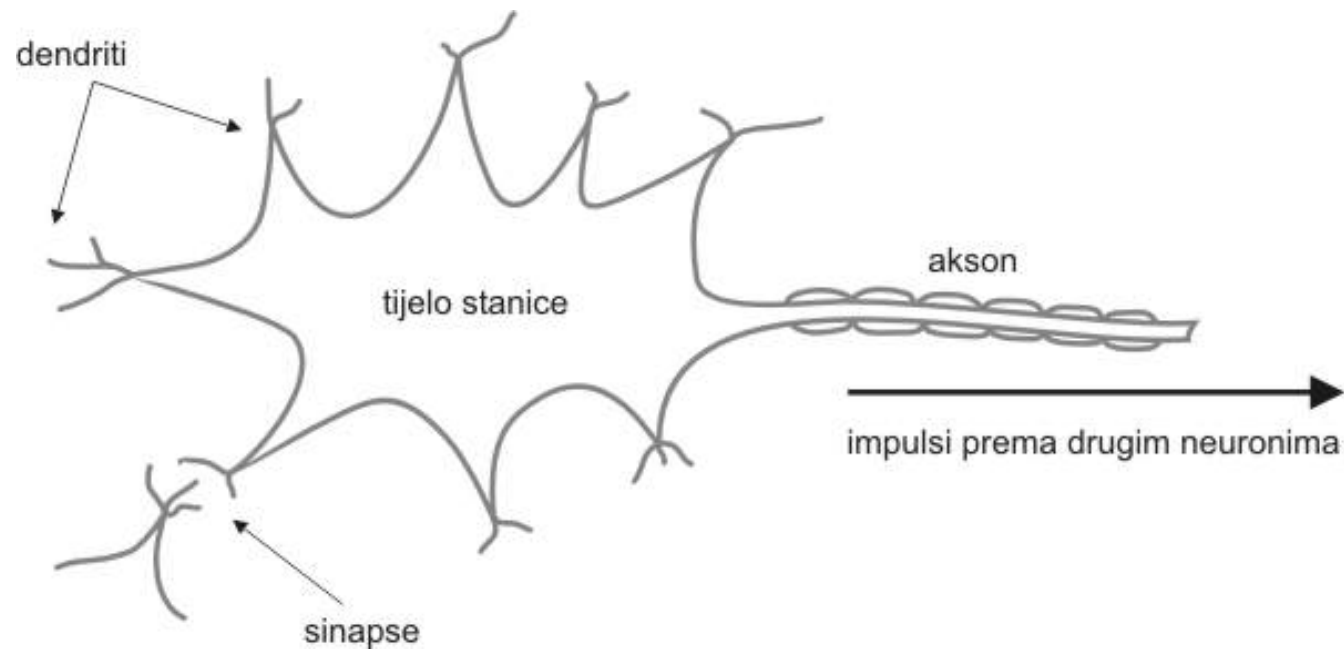
- Skup primjera za učenje često dijelimo na:
  - Skup za učenje – služi za iterativno podešavanje težina
  - Skup za testiranje – provjeravamo rad mreže
  - Skup za provjeru – konačna provjera
- Učenje se provodi dok mreža ne daje odgovarajuću točnost obrade podataka (uvodi se mjera pogreške)
- Pretreniranost ►  
ANN gubi poželjno svojstvo generalizacije i postaje stručnjak za podatke iz skupa za učenje (štreber)



# 1.

## Naš živčani sustav

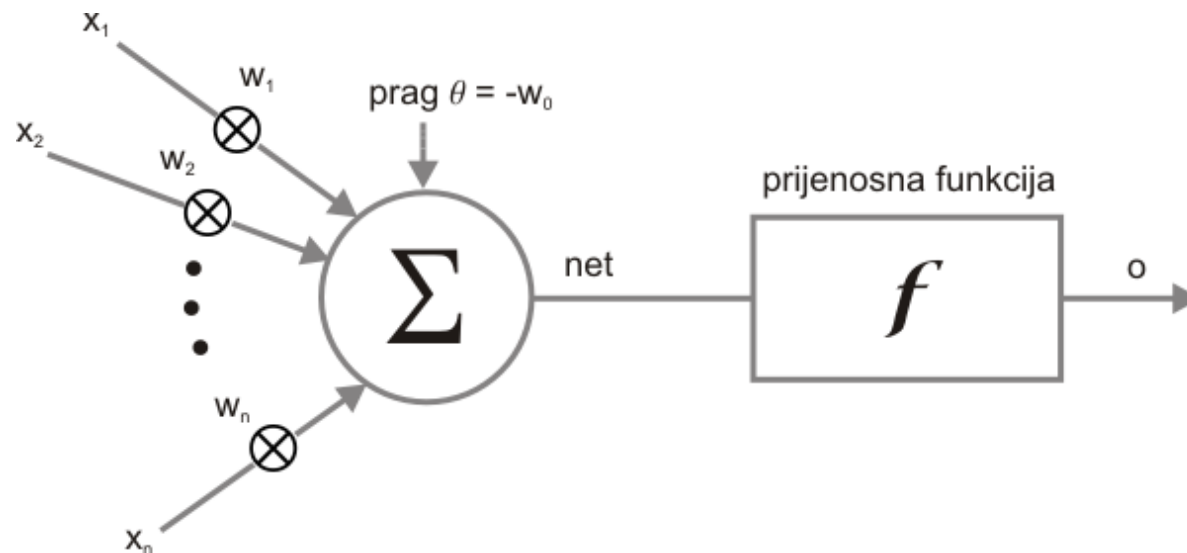
- $10^{11}$  neurona, 100 različitih vrsta, raspoređeni po definiranom rasporedu, svaki povezan s  $10^4$  drugih
- Dijelovi: soma, dendriti, akson, završni članci



# 1.

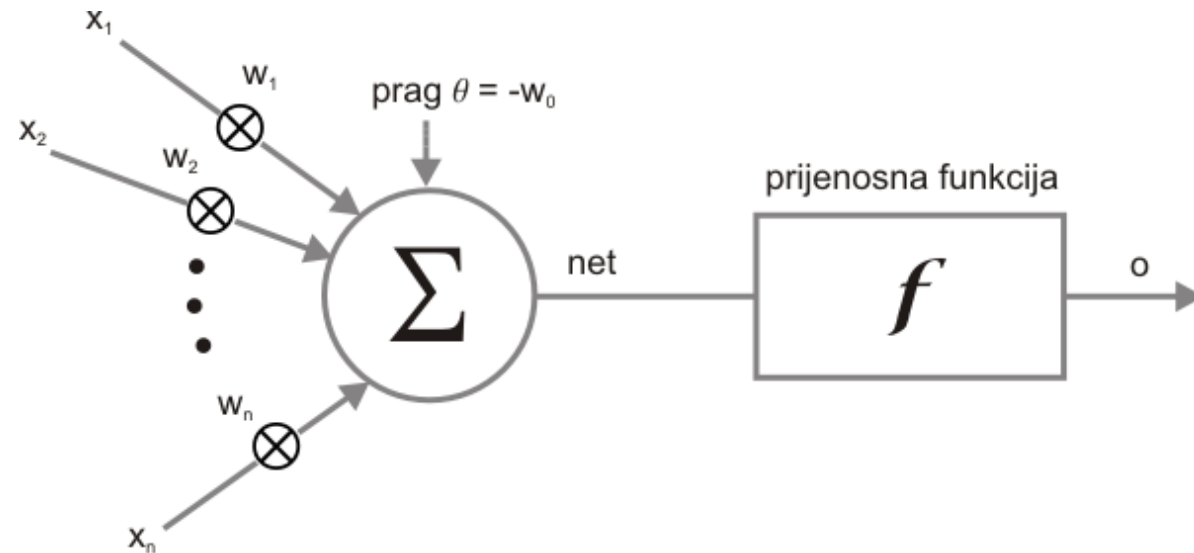
## Umjetni neuron

- McCulloch-Pitts model (1943.): *Threshold Logic Unit* (TLU)
- Analogija: signali su numeričke vrijednosti, jakost sinapse opisuje težinski faktor  $w$ , tijelo stanice je zbrajalo, akson je prijenosna (aktivacijska) funkcija  $f$



# 1.

## Umjetni neuron



$$net = \omega_1 x_1 + \omega_2 x_2 + \dots + \omega_n x_n - \theta$$

$$net = \omega_0 x_0 + \omega_1 x_1 + \omega_2 x_2 + \dots + \omega_n x_n = \sum_{i=0}^n \omega_i x_i$$

$X_0=1, w_0=-\text{theta}$   $o = f\left(\sum_{i=0}^n \omega_i x_i\right) = f(net)$

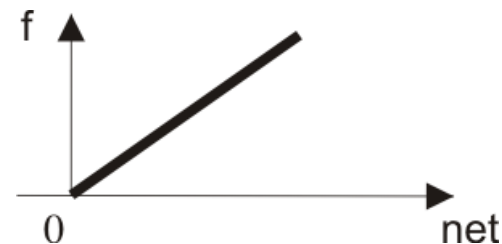
# 1.

## Umjetni neuron

- Različite funkcije dolaze u obzir kao prijenosne funkcije:

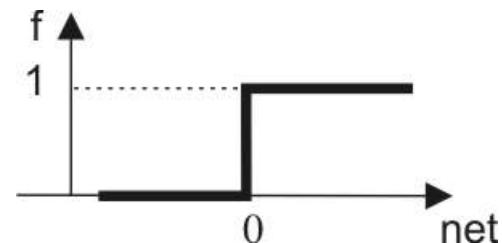
- ADALINE

$$f(net) = net$$



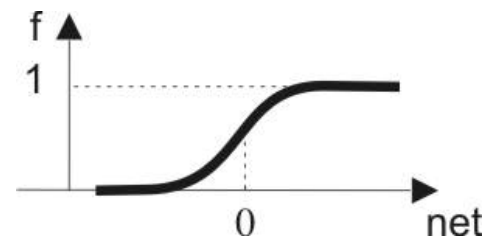
- TLU

$$f(net) = \begin{cases} 0 & \text{za } net < 0 \\ 1 & \text{inace} \end{cases}$$



- Sigmoidalna jedinica

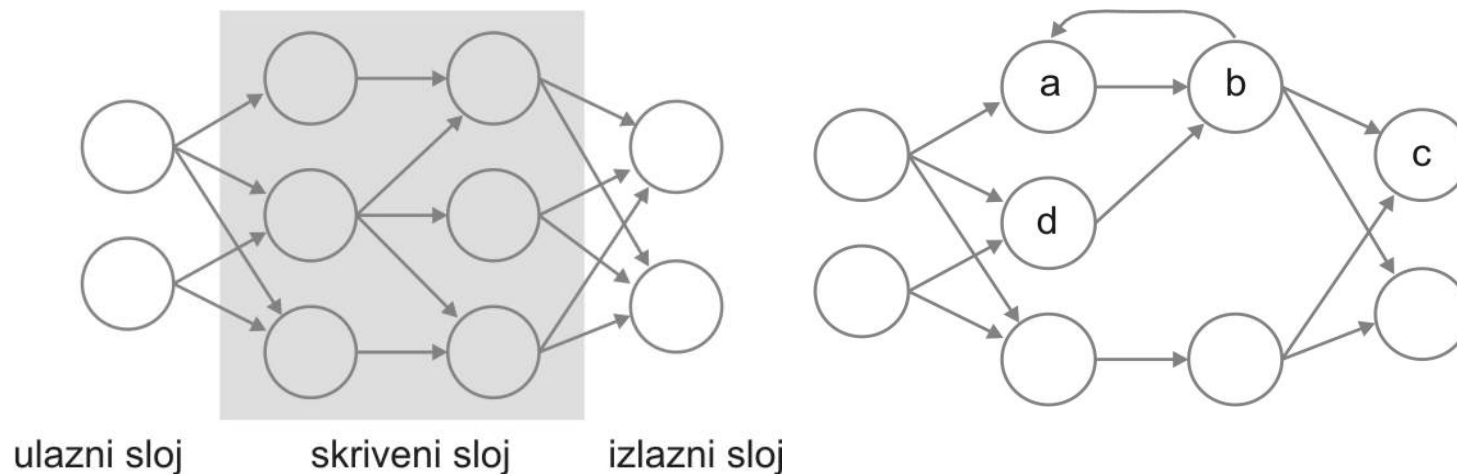
$$f(net) = \frac{1}{1 + e^{-a \cdot net}}$$



# 1.

## Arhitektura mreža

- Moguće arhitekture:

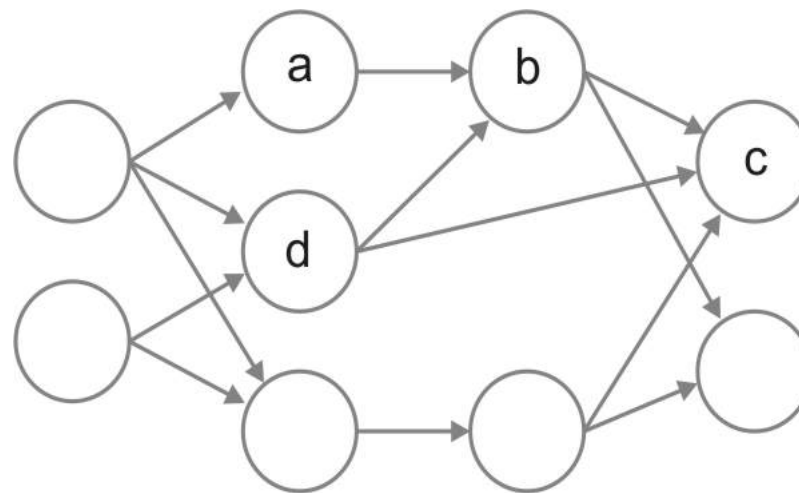


- Aciklička (*feedforward*)
- Mreža s povratnom vezom (*recurrent net*)
- Lateralno povezana mreža (*rešetkasta*)

# 1.

## Arhitektura mreža

- Podvrsta acikličke mreže je **slojevita aciklička mreža** - ne postoji skup od tri neurona  $B$ ,  $C$ ,  $D$  takav da je ulaz na  $C$  izlaz iz  $B$  i  $D$ , te da je istovremeno izlaz iz  $D$  spojen na ulaz neurona  $B$



► nije slojevita!

- Uniformno slojevita mreža
- Potpuno spojena mreža



## 1. Uvod u neuro-računarstvo

Povijesni pregled • Pravci AI • Definicija i vrste • Primjena • Učenje • Živčani sustav i biološki neuron • Model neurona i vrste • Arhitektura mreža

---

## 2. Procesni elementi

Perceptron • Pravilo perceptrona • Primjer • Učenje booleovih funkcija • Linearna odvojivost • XOR problem • Višeslojna mreža perceptrona • Linearna regresija • Srednja kvadratna pogreška • Gradijentni spust • LMS algoritam

---

## 3. Backpropagation algoritam

BACKPROPAGATION algoritam i interpretacija skrivenog sloja • Primjeri uporabe neuronskih mreža • Primjer sustava za raspoznavanje

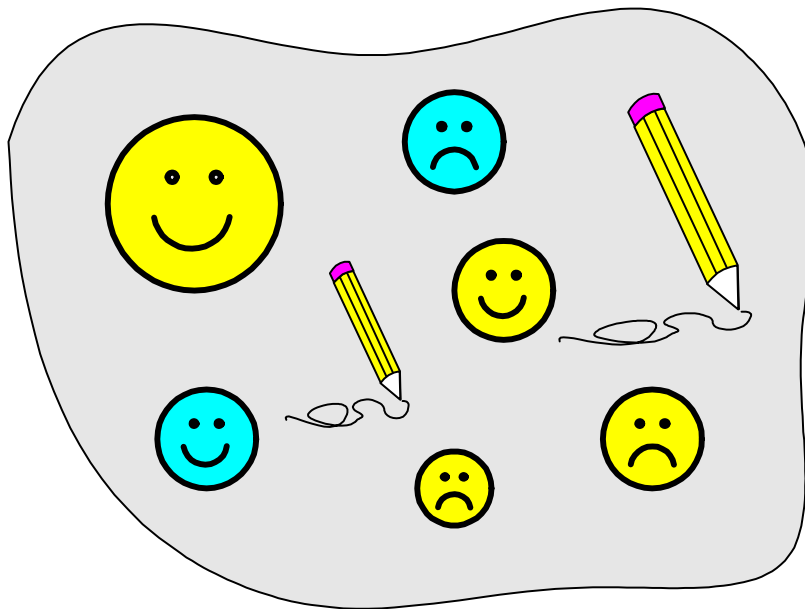
---



## 2.

## Klasifikacija

- Čovjek svakodnevno i neprestano obavlja – klasifikaciju!
- Postoji skup objekata/uzoraka koji imaju određen skup svojstava.
- Klasifikacija – pridjeljivanje naziva/oznaka svakom uzorku iz prostora uzoraka.

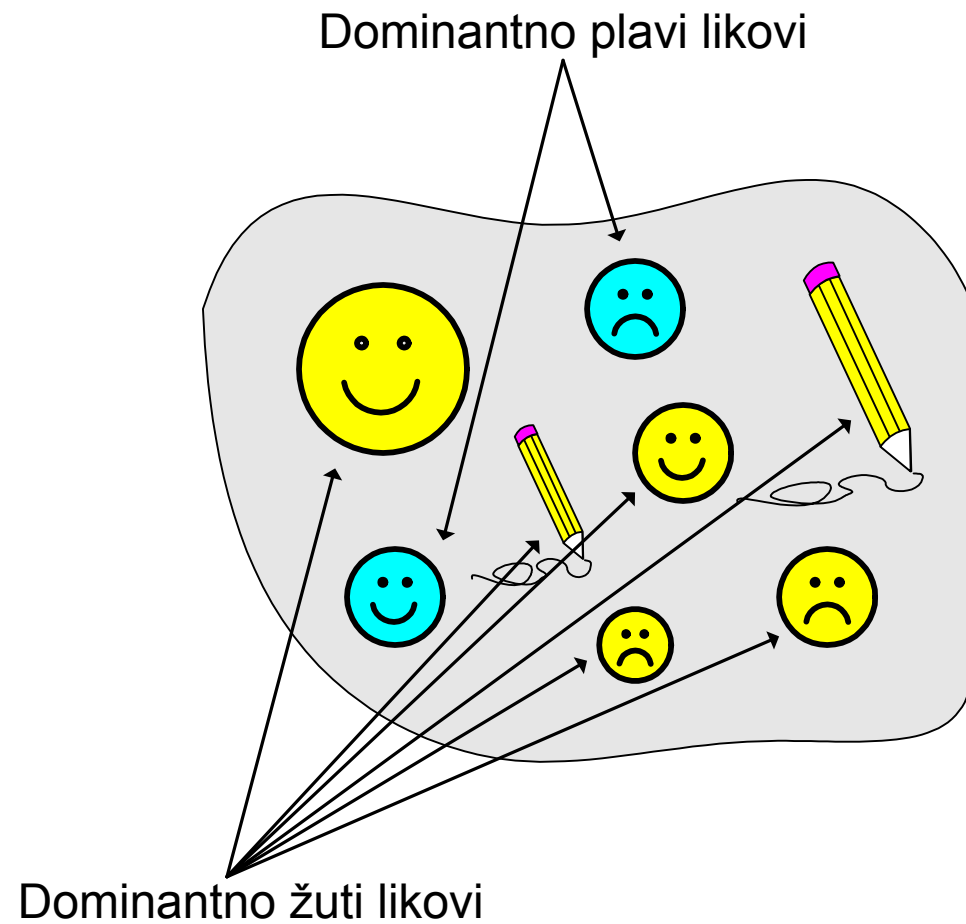


- Uzorci
- Prostor uzoraka

## 2.

## Klasifikacija

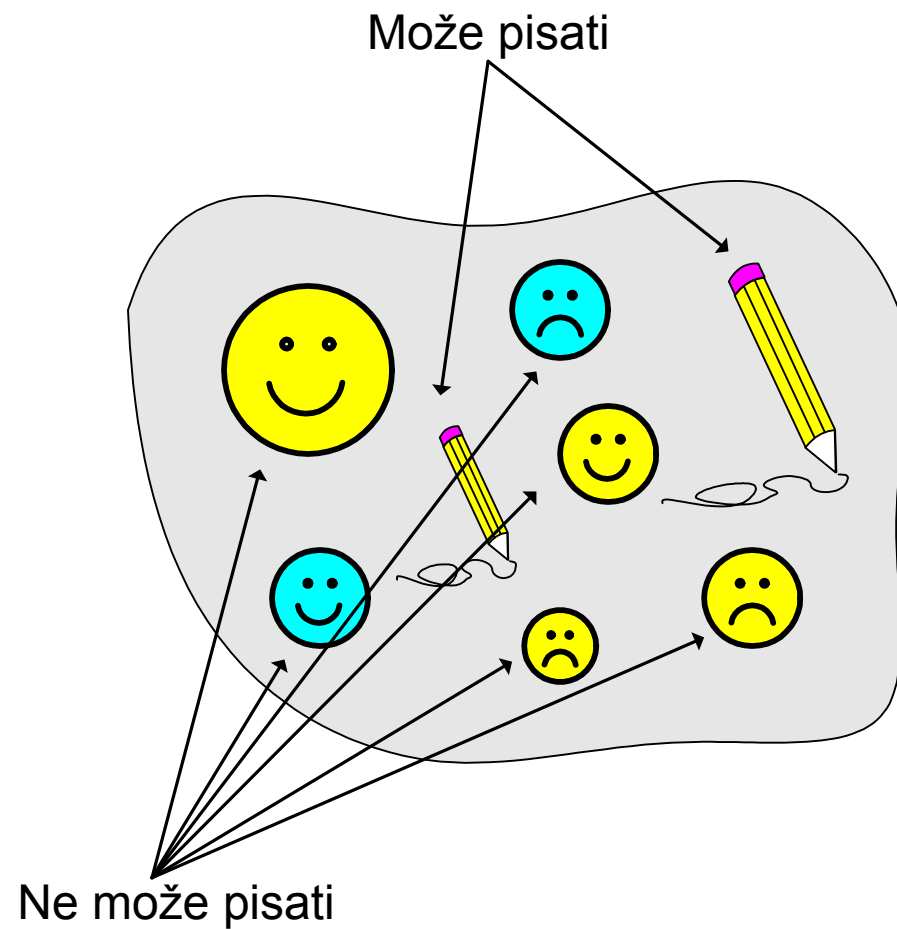
- Jedna moguća klasifikacija...



## 2.

## Klasifikacija

- Druga moguća klasifikacija...



- Svaki uzorak posjeduje određena svojstva (boja, oblik, dimenzije, ...).
- Klasifikaciju činimo na temelju nekih od svojstava.
- Uzoraka može biti beskonačno – ne možemo unaprijed generirati sve uzorke i pripadne klasifikacije → čovjek ipak dobro generalizira...
- Želimo postupak kojim bismo na temelju malog broja poznatih uzoraka mogli “naučiti” pravilno klasificirati neviđene primjere.

- Primjer koncepta “Dobra tajnica” – svaki objekt ima dva svojstva:
  - Svojstvo “Komunikativnost”
  - Svojstvo “Snalažljivost”
- Za potrebe računala svojstva treba kvantizirati – računalo radi s brojevima.
- Ocjenjujemo svojstva na skali od 1 do 5.
  - 1 je najlošija vrijednost.
  - 5 je najbolja vrijednost.
- Oznake “Dobra tajnica” i ne-“Dobra tajnica” kodiramo:
  - “Dobra tajnica” → 1
  - Ne-“Dobra tajnica” → -1

## 2.

## Klasifikacija

- Primjer "Dobra tajnica"

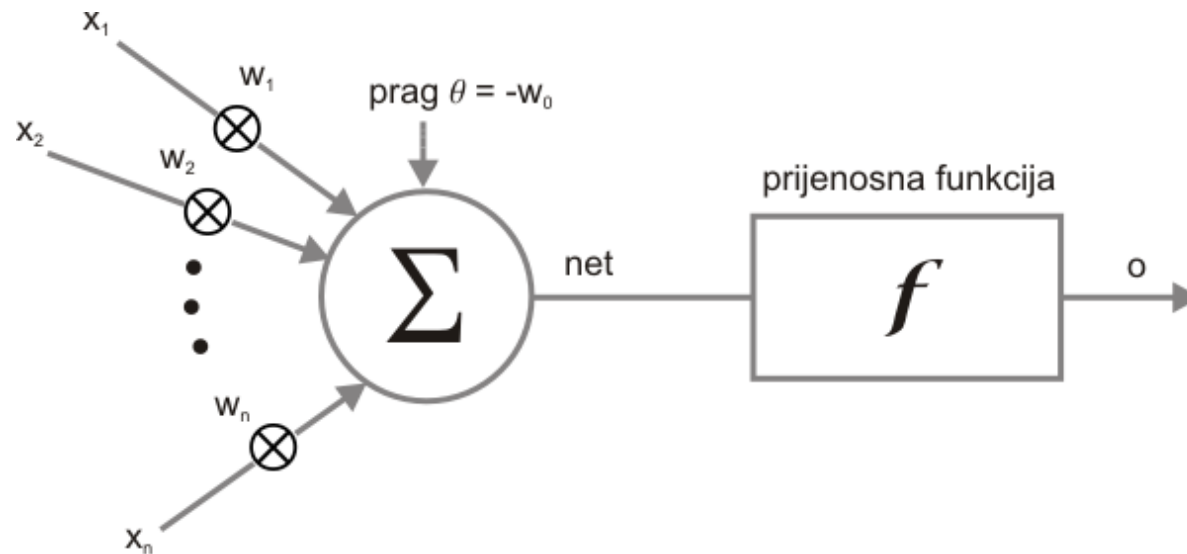
Komunikativnost	Snalažljivost	Oznaka	Kodirano
2	5	"Dobra tajnica"	1
5	2	"Dobra tajnica"	1
1	5	ne-"Dobra tajnica"	-1
5	1	ne-"Dobra tajnica"	-1

- Kada su *ulazi* i *izlazi* brojevi, za klasifikaciju možemo koristiti TLU perceptron.

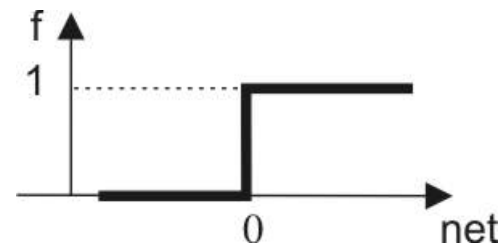
# 2.

## TLU perceptron

- 1943. McCulloch i Pitts – TLU Perceptron.



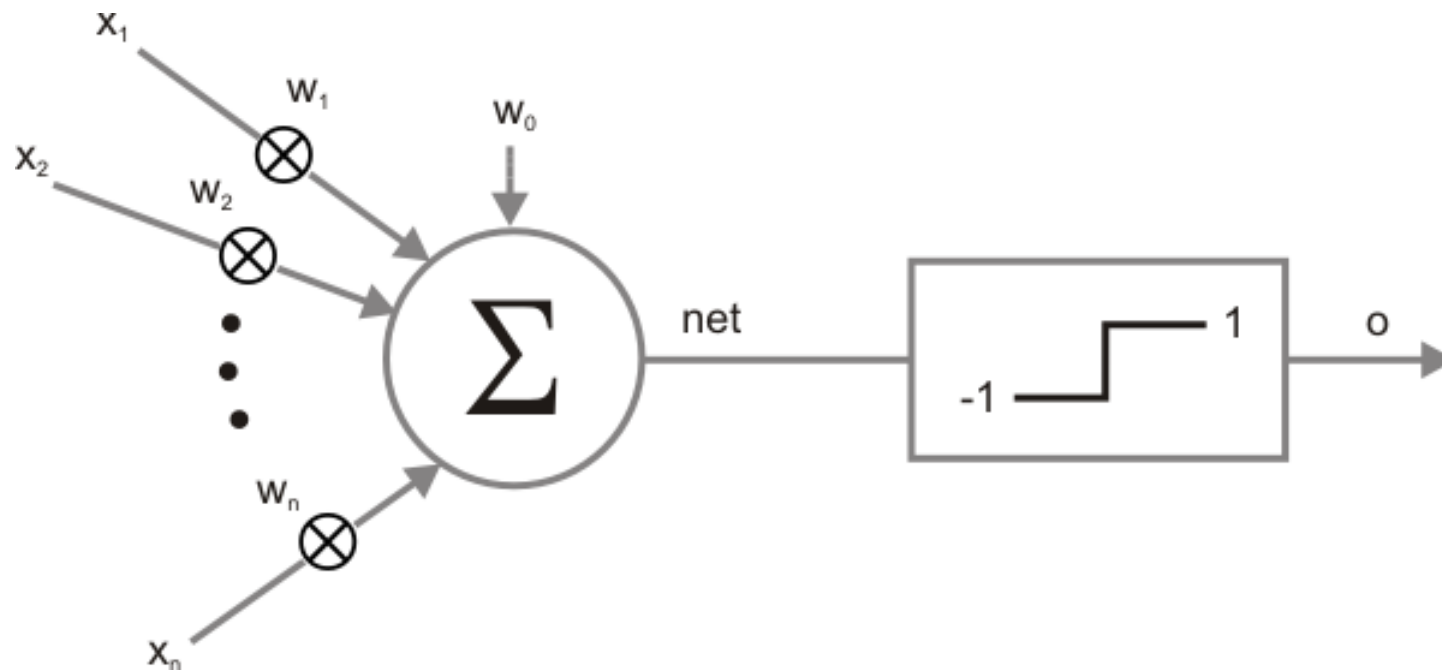
$$f(\text{net}) = \begin{cases} 0 & \text{za } \text{net} < 0 \\ 1 & \text{inace} \end{cases}$$



## 2.

## TLU perceptron

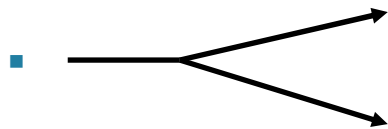
- 1943. McCulloch i Pitts – TLU Perceptron.





## 2.

## TLU perceptron

- 1943. McCulloch i Pitts – TLU Perceptron.
- Dvije definicije prijenosne funkcije:
  - $\text{Step}(x) = 1, x \geq 0$
  - 
    - $\text{Step}(x) = 0, x < 0$
    - $\text{Step}(x) = -1, x < 0$
- Oba oblika koriste se podjednako.
- U primjerima ćemo koristiti TLU:-1,1.

## 2.

## TLU perceptron

- Izlaz računamo u općem slučaju prema formuli:

$$o = \text{Step}(\text{net}) = \text{Step}(\vec{w}^T \cdot \vec{x}) = \text{Step} \left( \begin{bmatrix} w_n & \cdots & w_0 \end{bmatrix} \cdot \begin{bmatrix} x_n \\ \vdots \\ x_1 \\ 1 \end{bmatrix} \right)$$
$$= \text{Step}(w_n \cdot x_n + \dots + w_1 \cdot x_1 + w_0)$$

- EksPLICITNO se uvodi:  $x_0 = 1$

## 2.

## TLU perceptron

- Kada dolazi do promjene klasifikacije?

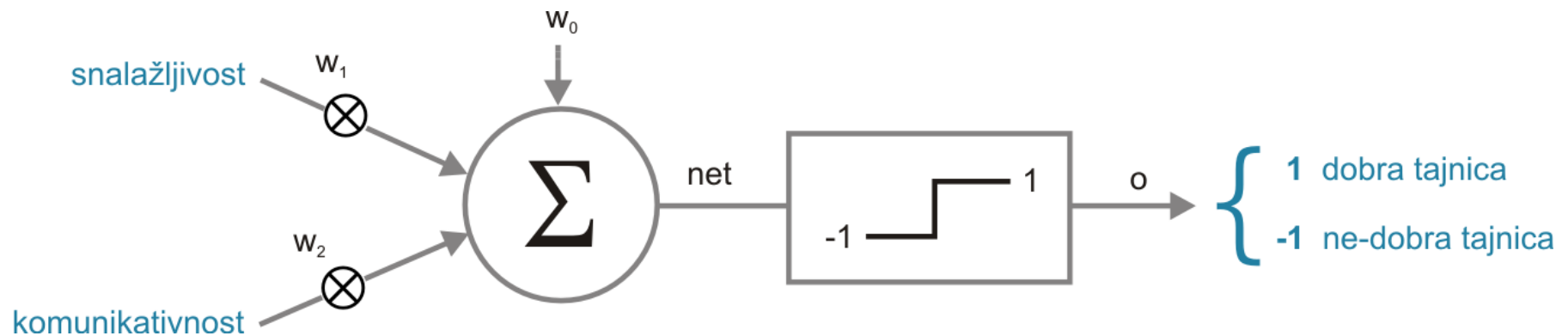
$$net = 0 \Rightarrow w_n \cdot x_n + \dots + w_1 \cdot x_1 + w_0 = 0$$

- Točke koje zadovoljavaju ovu jednadžbu čine granicu! (decizijska funkcija)
- Kada imamo dvije značajke (svojstva) granica je pravac.
- Više značajki rezultira hiperravninom.
- Bitno je uočiti – granica je **linearna**.

# 2.

## TLU perceptron

- U našem primjeru perceptron ima dva ulaza:



- Snalažljivost  $\equiv x_1$
- Komunikativnost  $\equiv x_2$

- Pretpostavimo:  $[w_2 \quad w_1 \quad w_0] = [1 \quad 1.3 \quad -5.85]$

# 2.

## TLU perceptron

- U našem primjeru izlaz računamo prema formuli:

$$o = \text{Step} \left( \begin{bmatrix} w_2 & w_1 & w_0 \end{bmatrix} \cdot \begin{bmatrix} x_2 \\ x_1 \\ 1 \end{bmatrix} \right) = \text{Step}(w_2 \cdot x_2 + w_1 \cdot x_1 + w_0)$$

Težinski faktori			Komun.	Snal.	Suma	Izlaz	Točan	Ispravno
$w_2$	$w_1$	$w_0$	$x_2$	$x_1$	net	$o=S(\text{net})$	t	
1	1.3	-5.85	2	5	2,65	1	1	<b>DA</b>
1	1.3	-5.85	5	2	1,75	1	1	<b>DA</b>
1	1.3	-5.85	1	5	1,65	1	-1	<b>NE</b>
1	1.3	-5.85	5	1	0,45	1	-1	<b>NE</b>

- 1949. Hebb – “*učiti znači mijenjati jakosti veza*”!
- Potrebno je mijenjati težinske faktore.
- 1958. Rosenblatt: spoj Hebbove ideje i McCulloch-Pitts modela
- **Pravilo Perceptrona**
  - Ukoliko se uzorak klasificira **ispravno** ▶ **ne radi korekciju**
  - Ukoliko se uzorak klasificira **neispravno** ▶ **primjeni korekciju**
  - Ciklički uzimaj sve uzorke redom, a postupak zaustavi kada su svi uzorci za redom klasificirani ispravno
- Korekcija: 
$$w_i(k+1) = w_i(k) + \eta \cdot (t - o) \cdot x(k)$$

## 2.

## TLU perceptron

- Učenje perceptrona uz:

- $\eta = 0.02$

- $[w_2 \quad w_1 \quad w_0] = [1 \quad 1.3 \quad -5.85]$

- Postupak završava s:

$$[w_2 \quad w_1 \quad w_0] = [0.92 \quad 0.94 \quad -5.93]$$

## 2.

## TLU perceptron

- Kako se klasificira novi uzorak [5 5]?

$$\text{Step}(0.92 \cdot 5 + 0.94 \cdot 5 - 5.93) = \text{Step}(3.37) = 1$$

- Kako se klasificira novi uzorak [1 1]?

$$\text{Step}(0.92 \cdot 1 + 0.94 \cdot 1 - 5.93) = \text{Step}(-4.07) = -1$$

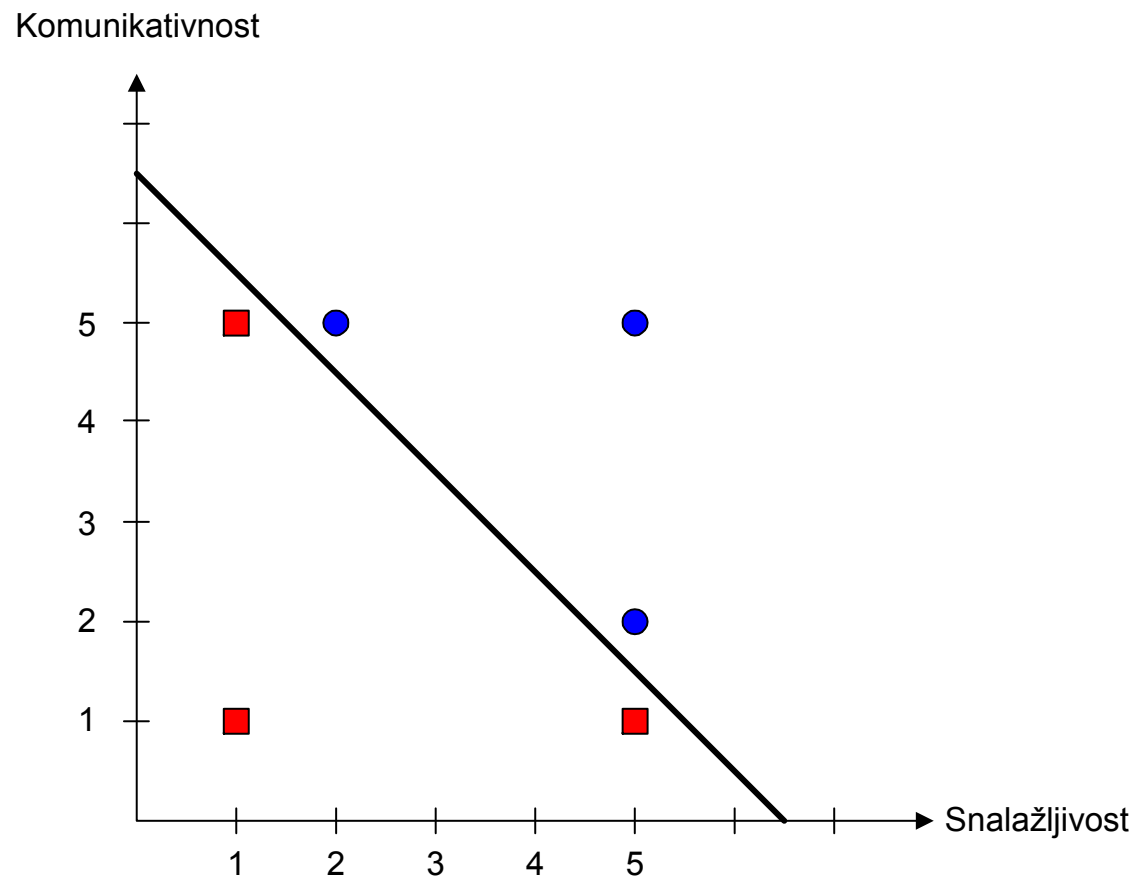
- Vidimo da je perceptron NAUČIO klasificirati uzorke koje prethodno nije vidio!



## 2.

## TLU perceptron

- Učenje je uspjelo jer su razredi bili LINEARNO RAZDVOJIVI!

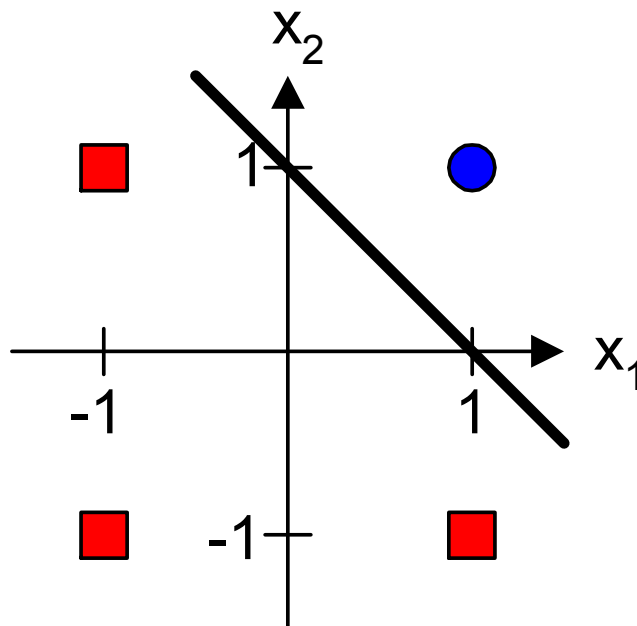


Primjer...

## 2.

## TLU perceptron

- TLU može naučiti neke logičke funkcije – I, ILI, NE, ...



- Logičko **I**
  - Istinu kodiramo s 1
  - Laž kodiramo s -1

$$w_1 = w_2 = 1$$

$$w_0 = -1$$

- Logičko **I** s M ulaza:

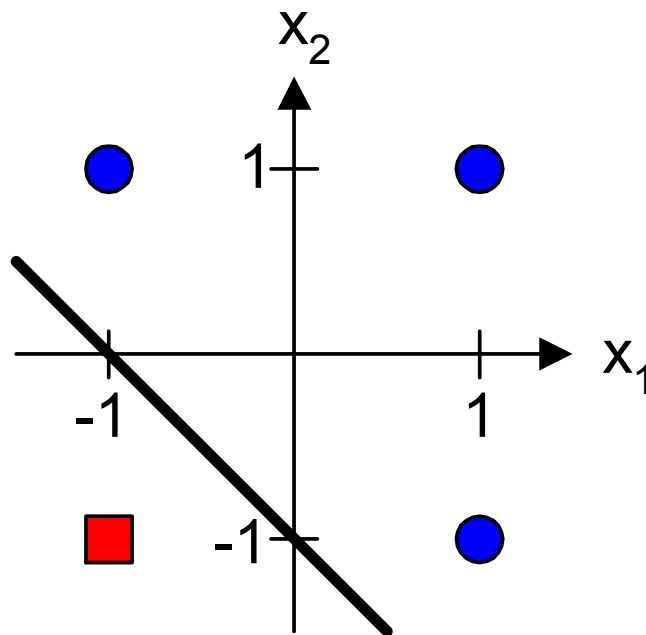
$$w_1 = \dots = w_M = 1$$

$$w_0 = -M \{+1\}$$

## 2.

## TLU perceptron

- TLU može naučiti neke logičke funkcije – I, ILI, NE, ...



- Logičko **ILI**

- Istinu kodiramo s 1
- Laž kodiramo s -1

$$w_1 = w_2 = 1$$

$$w_0 = 1$$

- Logičko **ILI** s M ulaza:

$$w_1 = \dots = w_M = 1$$

$$w_0 = M - 1$$

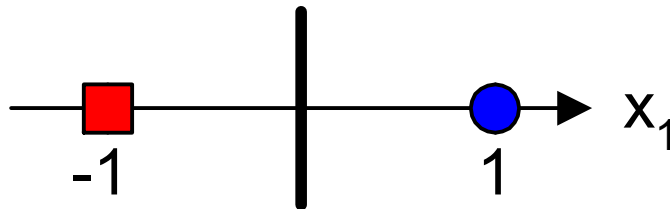
## 2.

## TLU perceptron

- TLU može naučiti neke logičke funkcije – I, ILI, NE, ...

- Logičko **NE**

- Istinu kodiramo s 1
- Laž kodiramo s -1



$$w_1 = -1$$

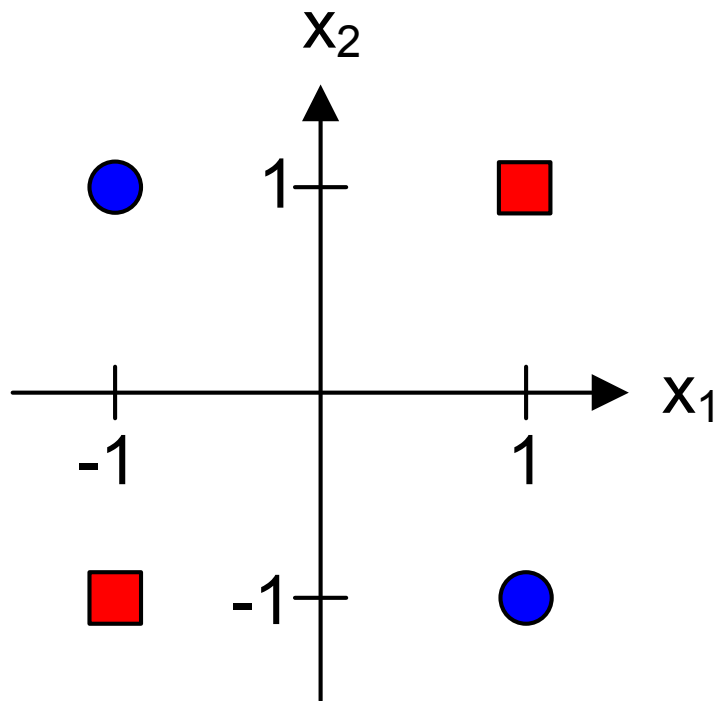
$$w_0 = 0$$

- Jednostavnije: negaciju ostvariti koristeći perceptron koji već obavlja neku drugu funkciju tako da težinskom faktoru koji dovodi tu "varijablu" promijenimo predznak!

## 2.

## XOR problem

- Logička funkcija **XOR**?

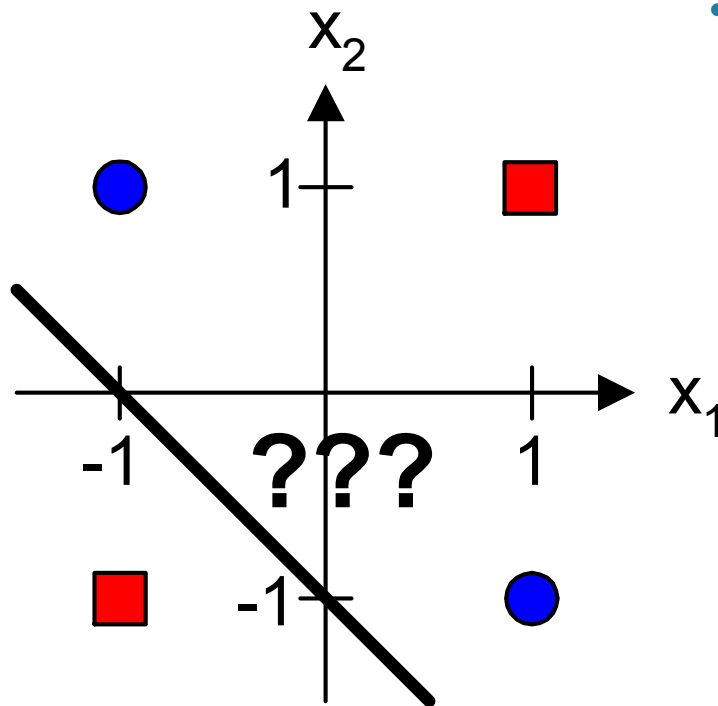


- Logičko **Ex-ILI**
  - Istinu kodiramo s 1
  - Laž kodiramo s -1
$$w_1 = ?$$
$$w_2 = ?$$
$$w_0 = ?$$

## 2.

## XOR problem

- Što je s logičkom funkcijom **XOR**?
- Nije linearno razdvojivo!

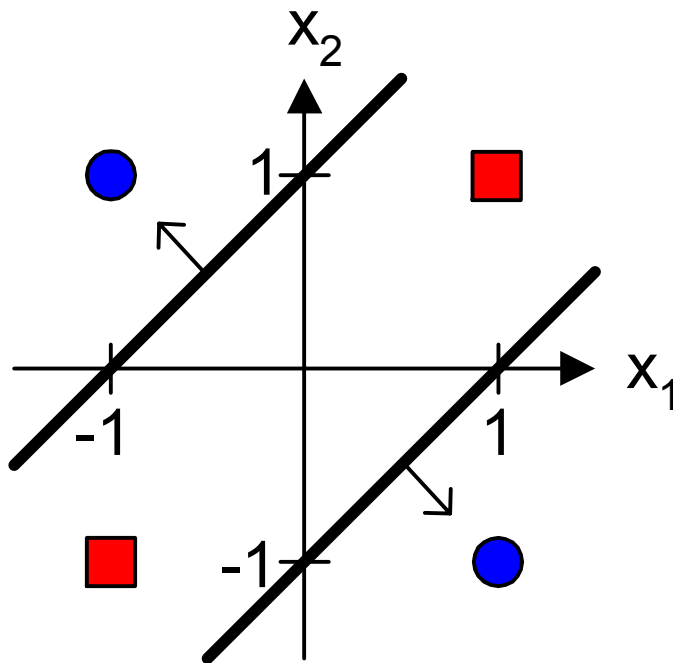


- 1969. Minsky i Papert – ‘Perceptrons’
  - Perceptron nije dobar kada ne može riješiti tako jednostavan problem poput XOR elementarne logičke funkcije!

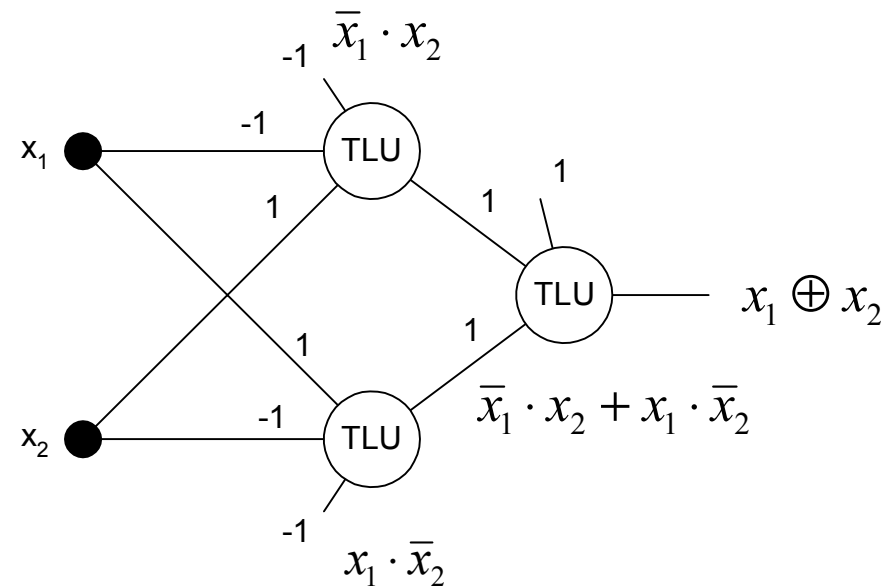
## 2.

## XOR problem

- Što je s logičkom funkcijom **XOR**? Treba konstruirati mrežu!



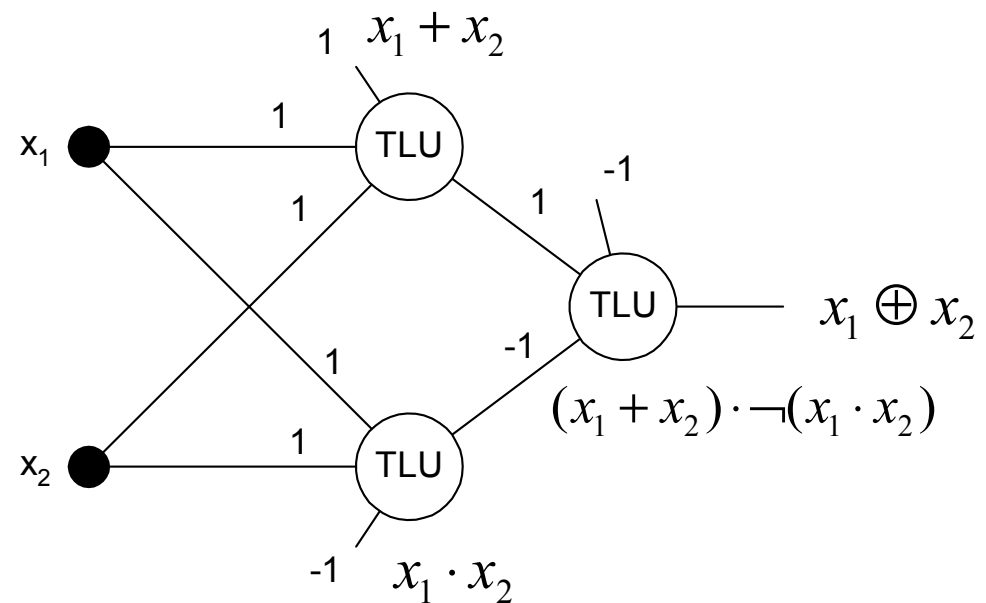
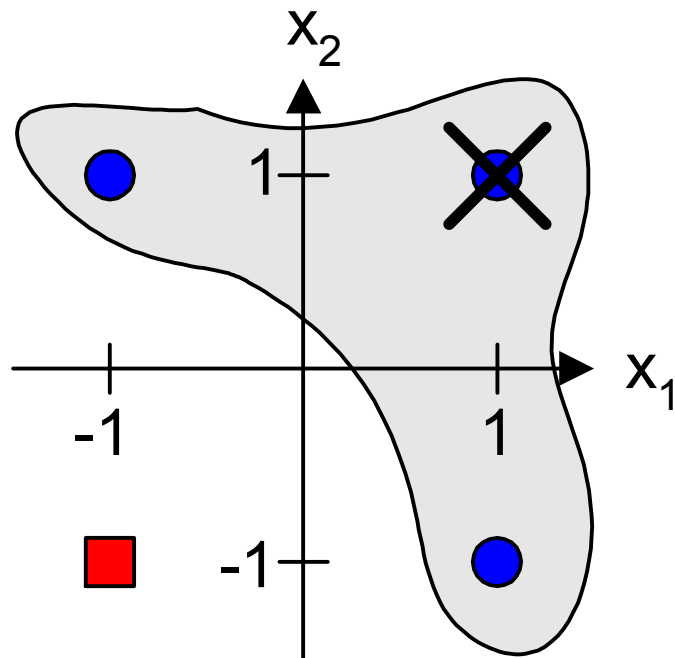
$$x_1 \oplus x_2 = \bar{x}_1 \cdot x_2 + x_1 \cdot \bar{x}_2$$



# 2.

## XOR problem

- Što je s logičkom funkcijom **XOR**? Treba konstruirati mrežu!
- Rješenje nije jednoznačno!
- ... Kada je **ILI** i nije **I**

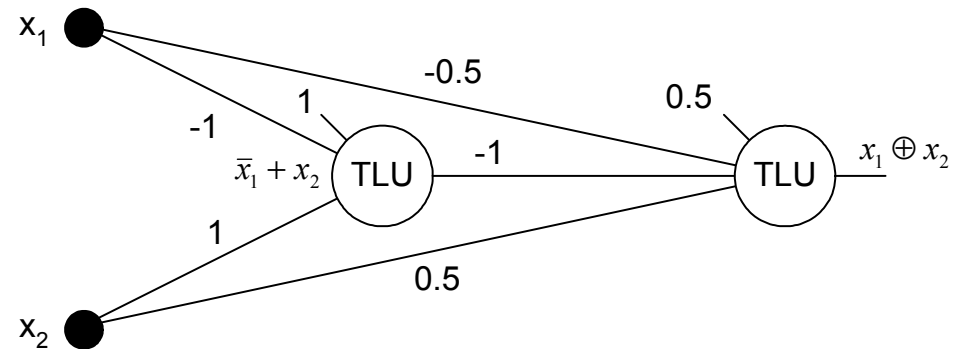
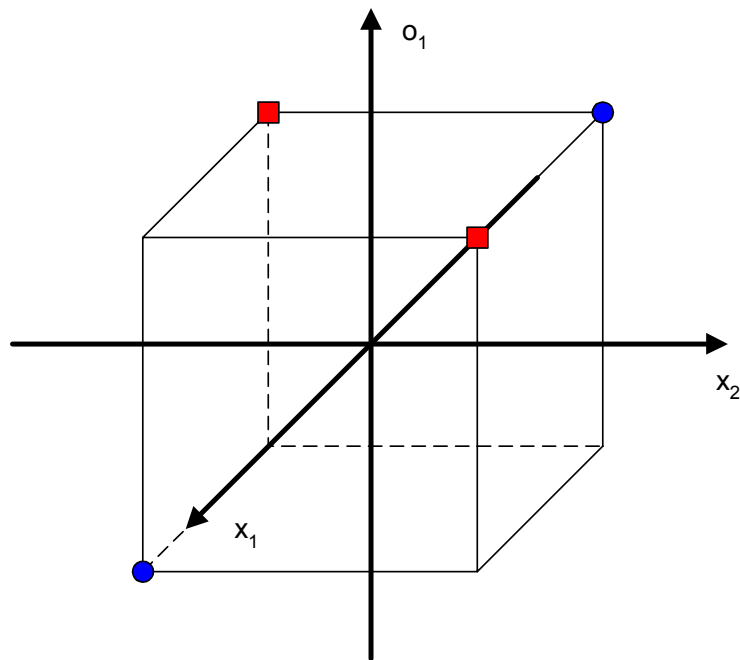




## 2.

## TLU perceptron

- Što je s logičkom funkcijom **XOR**? Treba konstruirati mrežu!
- Povećanjem dimenzionalnosti ulaza postići linearnu odvojivost → trik poznat iz *kernel machines* !

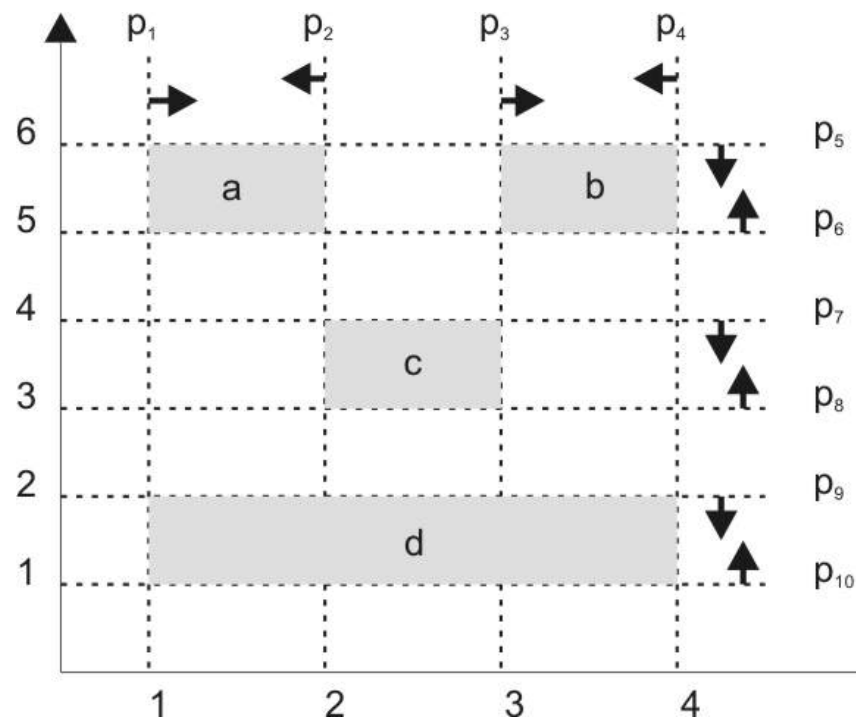


Primjer...

## 2.

## Višeslojna mreža perceptrona

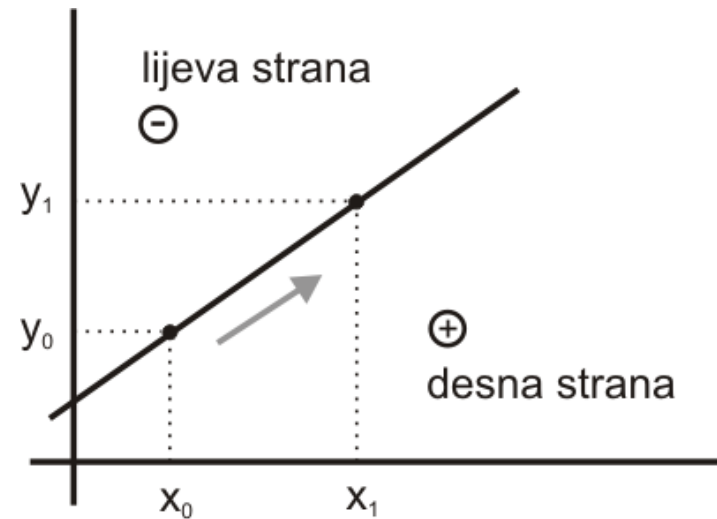
- Za predočavanje složenijih odnosa moramo koristiti mrežu od više međusobno povezanih perceptrona
- Primjer projektiranja ANN za klasifikaciju uzoraka unaprijed znane raspodjele



## 2.

## Višeslojna mreža perceptrona

$$y - y_0 = \frac{y_1 - y_0}{x_1 - x_0} (x - x_0)$$



$$y(x_0 - x_1) + x(y_1 - y_0) + y_0(x_1 - x_0) - x_0(y_1 - y_0) = 0$$

$$y \cdot w_2 + x \cdot w_1 + w_0 = 0$$

$$x_2 \cdot w_2 + x_1 \cdot w_1 + w_0 = 0$$

# 2.

## Višeslojna mreža perceptrona

$$p_1 \dots 1 \cdot x + 0 \cdot y - 1 = 0$$

$$p_2 \dots -1 \cdot x + 0 \cdot y + 2 = 0$$

$$p_3 \dots 1 \cdot x + 0 \cdot y - 3 = 0$$

$$p_4 \dots -1 \cdot x + 0 \cdot y + 4 = 0$$

$$p_5 \dots 0 \cdot x - 1 \cdot y + 6 = 0$$

$$p_6 \dots 0 \cdot x + 1 \cdot y - 5 = 0$$

$$p_7 \dots 0 \cdot x - 1 \cdot y + 4 = 0$$

$$p_8 \dots 0 \cdot x + 1 \cdot y - 3 = 0$$

$$p_9 \dots 0 \cdot x - 1 \cdot y + 2 = 0$$

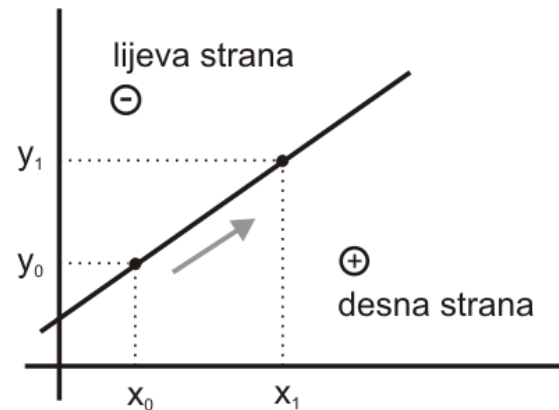
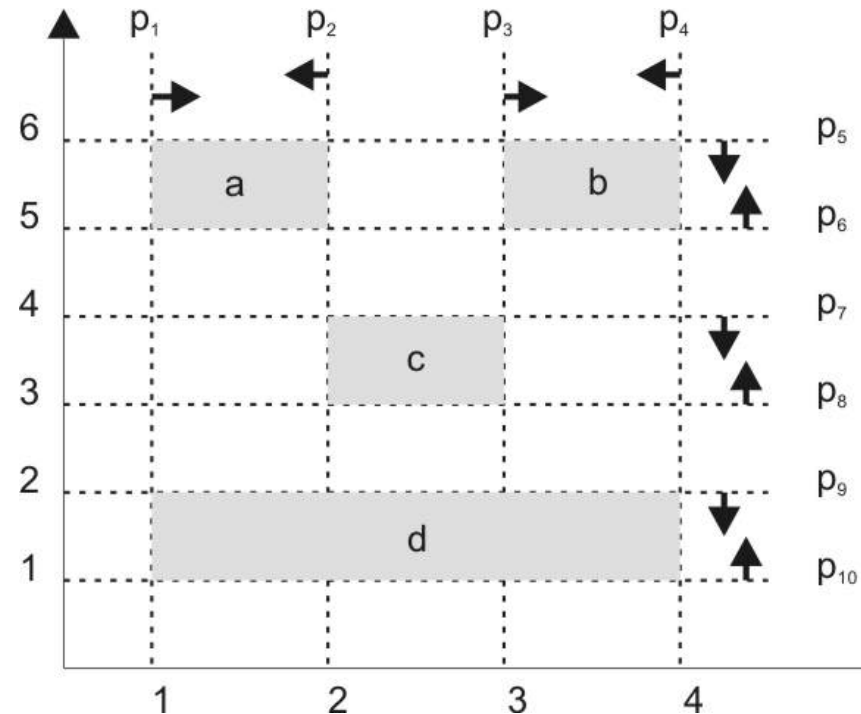
$$p_{10} \dots 0 \cdot x + 1 \cdot y - 1 = 0$$

$$a \equiv p_1 \cap p_2 \cap p_5 \cap p_6$$

$$b \equiv p_3 \cap p_4 \cap p_5 \cap p_6$$

$$c \equiv (-p_2) \cap (-p_3) \cap p_7 \cap p_8$$

$$d \equiv p_1 \cap p_4 \cap p_9 \cap p_{10}$$



# 2.

## Višeslojna mreža perceptrona

$$p_1 \dots 1 \cdot x + 0 \cdot y - 1 = 0$$

$$p_2 \dots -1 \cdot x + 0 \cdot y + 2 = 0$$

$$p_3 \dots 1 \cdot x + 0 \cdot y - 3 = 0$$

$$p_4 \dots -1 \cdot x + 0 \cdot y + 4 = 0$$

$$p_5 \dots 0 \cdot x - 1 \cdot y + 6 = 0$$

$$p_6 \dots 0 \cdot x + 1 \cdot y - 5 = 0$$

$$p_7 \dots 0 \cdot x - 1 \cdot y + 4 = 0$$

$$p_8 \dots 0 \cdot x + 1 \cdot y - 3 = 0$$

$$p_9 \dots 0 \cdot x - 1 \cdot y + 2 = 0$$

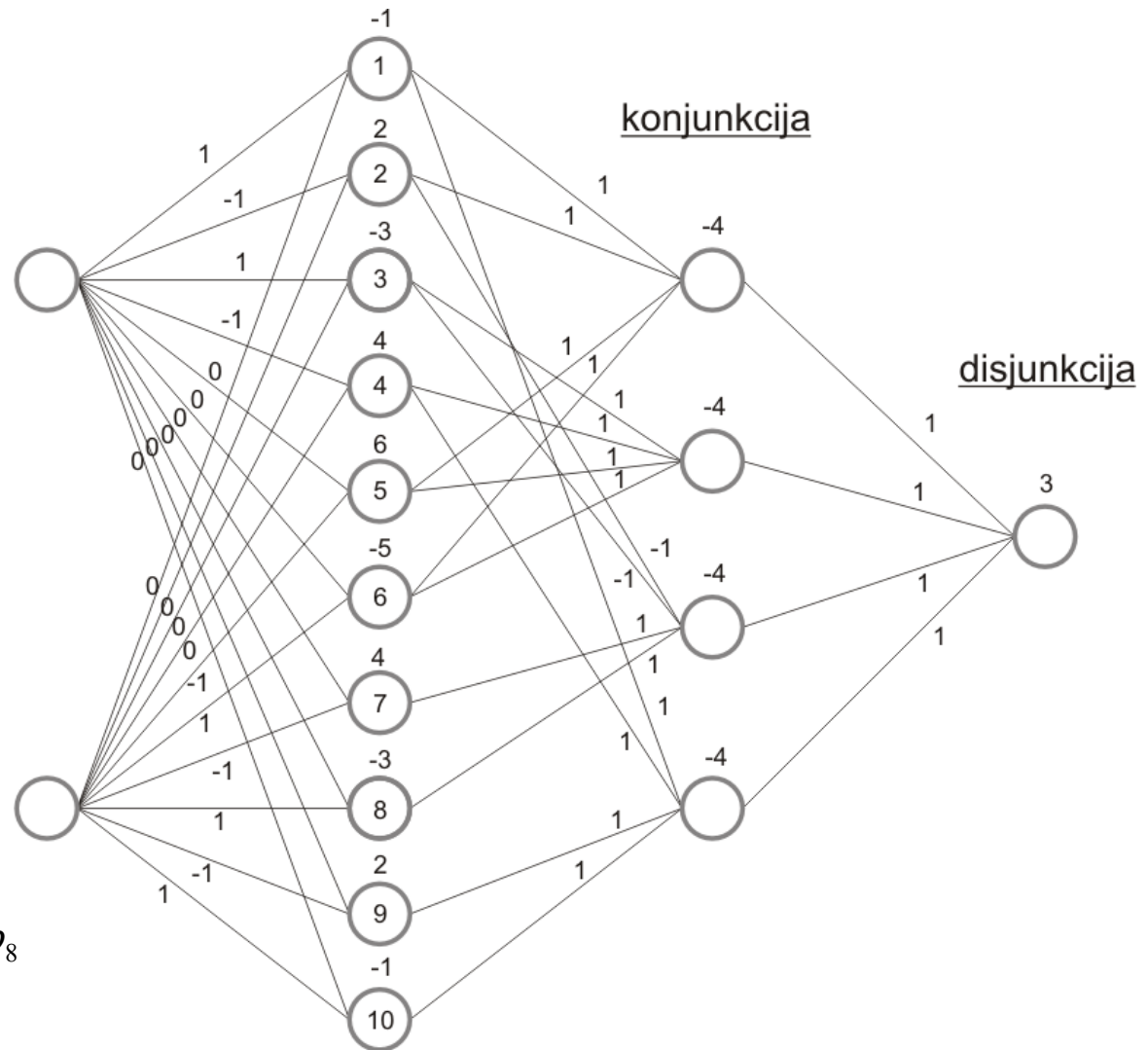
$$p_{10} \dots 0 \cdot x + 1 \cdot y - 1 = 0$$

$$a \equiv p_1 \cap p_2 \cap p_5 \cap p_6$$

$$b \equiv p_3 \cap p_4 \cap p_5 \cap p_6$$

$$c \equiv (-p_2) \cap (-p_3) \cap p_7 \cap p_8$$

$$d \equiv p_1 \cap p_4 \cap p_9 \cap p_{10}$$



## 2.

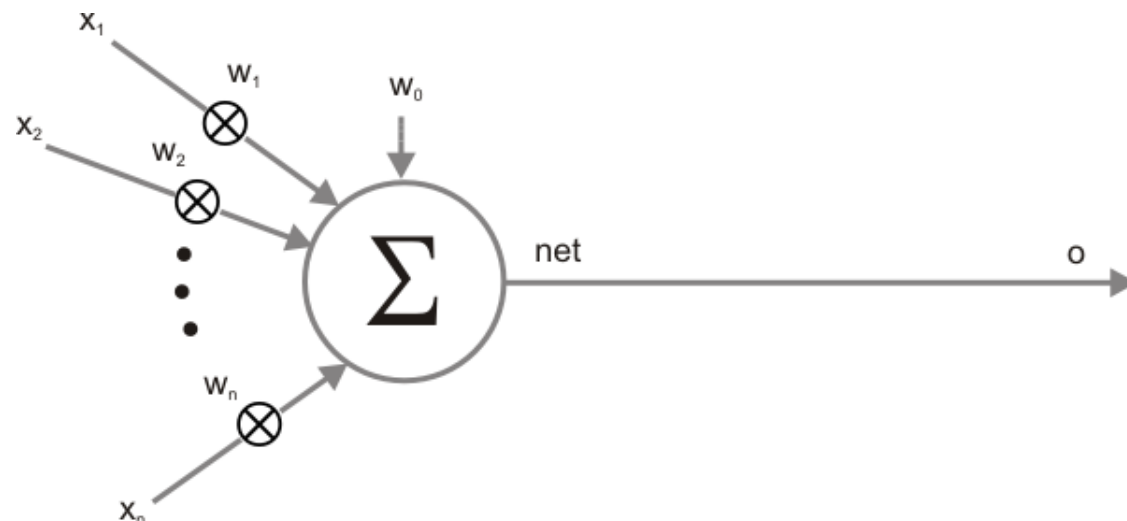
## TLU perceptron

- Problem s mrežama TLU perceptrona jest nemogućnost učenja konvencionalnim postupcima!
- Rješenje tražimo na drugom mjestu...

## 2.

# Aproksimacija funkcija

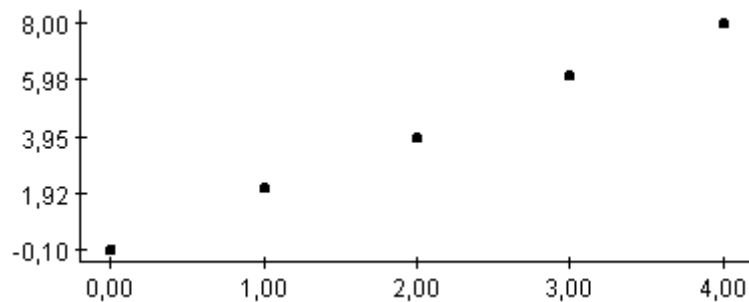
- Također vrlo težak problem – i vrlo koristan (ako se može riješiti)!
- TLU perceptron nije prikladan za aproksimaciju funkcija – izlaz ima samo dvije razine.
- Jedno moguće rješenje jest iz TLU-a izbaciti nelinearnost! Takav procesni element naziva se ADALINE.



## 2.

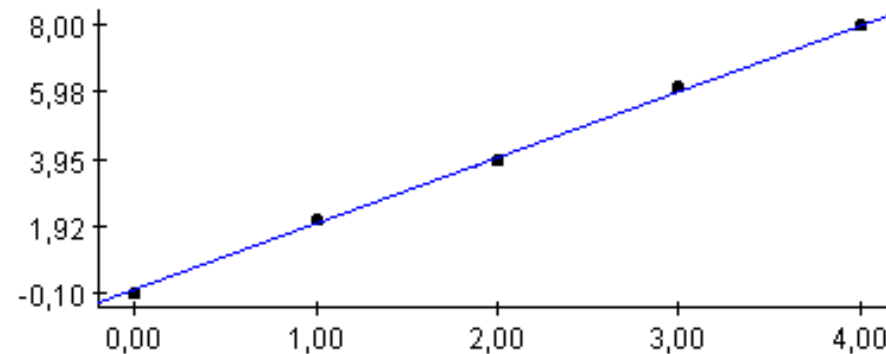
## Adaline

- Osnovno svojstvo ► linearna regresija



$$t_i \approx w \cdot x_i$$

???





## 2.

## Adaline

$$t_i \approx w \cdot x_i$$

$$t_i = w \cdot x_i + \mathcal{E}_i \quad \longrightarrow \quad \mathcal{E}_i = t_i - w \cdot x_i$$

½ MSE (engl. Mean Square Error)

$$E = \frac{1}{2N} \sum_{i=1}^N \mathcal{E}_i^2 = \frac{1}{2N} \sum_{i=1}^N (t_i - w \cdot x_i)^2$$

## 2.

## Adaline

$$\begin{aligned}\frac{dE}{dw} &= 0 \\ \Rightarrow \frac{d}{dw} \left( \frac{1}{2N} \sum_{i=1}^N (t_i - w \cdot x_i)^2 \right) &= 0 \\ \Rightarrow -\frac{1}{N} \sum_{i=1}^N (t_i - w \cdot x_i) \cdot x_i &= 0 \\ \Rightarrow w &= \frac{\sum_{i=1}^N t_i \cdot x_i}{\sum_{i=1}^N x_i^2}\end{aligned}$$

Ne može se uvijek rješavati analitički...

- Gradijentni spust – jedno moguće rješenje

$$w(k+1) = w(k) - \eta \cdot \frac{d}{dw} E(k)$$

- U općenitom slučaju  $\mathbf{w}$  je vektor i umjesto derivacije je  $\nabla E$
- Gradijent  $\frac{1}{2}$  MSE-a:

$$\nabla_w E = \frac{dE}{dw} = \frac{d}{dw} \left( \frac{1}{2N} \sum_{i=1}^N \epsilon_i^2 \right) = \frac{1}{N} \sum_{i=1}^N \epsilon_i \frac{d}{dw} \epsilon_i$$

➔ ovisi o SVIM uzorcima - nepoželjno

## 2.

## Adaline

- Widrow-Hoff (1960.) – aproksimacija gradijenta:

$$\nabla_w E \approx \varepsilon(k) \frac{d}{dw} \varepsilon(k) = -\varepsilon(k) \cdot x(k)$$

- Konačna formula ► **LMS pravilo**

$$w(k+1) = w(k) + \eta \cdot \varepsilon(k) \cdot x(k)$$

- 'on-line' izvedba - pojedinačno učenje

Primjer...

## 2.

## Adaline

- Potpuno funkcionalni ADALINE (D ulaza):

$$net = w_D \cdot x_D + w_{D-1} \cdot x_{D-1} + \dots + w_1 \cdot x_1 + w_0$$

$$\varepsilon_i = t_i - (w_D \cdot x_{i,D} + w_{D-1} \cdot x_{i,D-1} + \dots + w_1 \cdot x_{i,1} + w_0) =$$

$$= t_i - \sum_{k=0}^D w_k \cdot x_{i,k}$$

- Dalje je priča poznata...

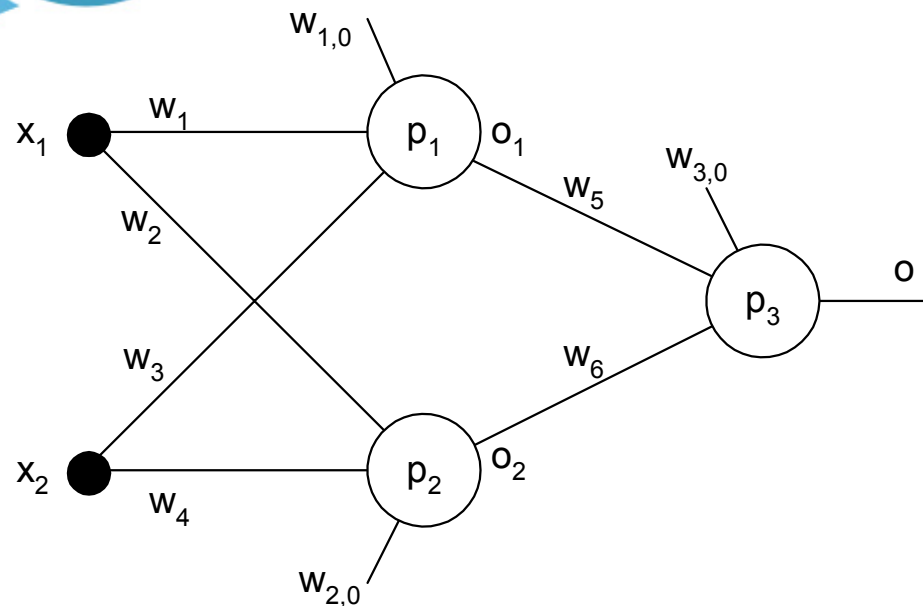
Umjesto skalara → vektori

LMS:

$$\vec{w}(k+1) = \vec{w}(k) - \eta \cdot \nabla E(k) = \vec{w}(k) + \eta \cdot \varepsilon(k) \cdot \vec{x}(k)$$

## 2.

## Adaline mreža



$$O = x_1 \cdot w_1 \cdot w_5 + x_2 \cdot w_3 \cdot w_5 + w_{1,0} \cdot w_5 \\ + x_1 \cdot w_2 \cdot w_6 + x_2 \cdot w_4 \cdot w_6 + w_{2,0} \cdot w_6 \\ + w_{3,0}$$

$$o_1 = x_1 \cdot w_1 + x_2 \cdot w_3 + w_{1,0}$$

$$o_2 = x_1 \cdot w_2 + x_2 \cdot w_4 + w_{2,0}$$

$$O = o_1 \cdot w_5 + o_2 \cdot w_6 + w_{3,0}$$

$$O = x_1 \cdot w_A + x_2 \cdot w_B + w_0$$

$$w_A = w_1 \cdot w_5 + w_2 \cdot w_6$$

$$w_B = w_3 \cdot w_5 + w_4 \cdot w_6$$

$$w_0 = w_{1,0} \cdot w_5 + w_{2,0} \cdot w_6 + w_{3,0}$$

Opet običan adaline...

## 2.

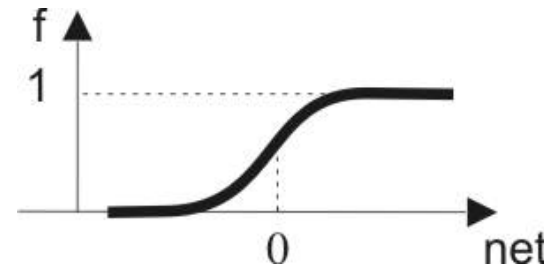
## Adaline mreža

Linearna kombinacija linearnih kombinacija opet je linearna kombinacija...

**Trebamo nelinearne prijenosne funkcije!**

→ Često se koristi sigmoidalna (logistička) prijenosna funkcija.

$$f(net) = \frac{1}{1 + e^{-a \cdot net}}$$



- Ovakve mreže su univerzalni aproksimatori – mogu aproksimirati proizvoljnu funkciju s proizvoljnom preciznošću.

## 2.

## Kolmogorovljev teorem

- Kolmogorov – egzistencijalni teorem (1957)
  - kontinuirana funkcija ***f*** od ***s*** varijabli može se prikazati u obliku konačne sume odgovarajućih kontinuiranih funkcija ***g<sub>q</sub>*** jedne varijable:

$$f(x_1, x_2, \dots, x_s) = \sum_{q=0}^{2s} g_q \left[ \sum_i \Psi_{qi}(x_i) \right]$$





## 1. Uvod u neuro-računarstvo

Povijesni pregled • Pravci AI • Definicija i vrste • Primjena • Učenje • Živčani sustav i biološki neuron • Model neurona i vrste • Arhitektura mreža

---

## 2. Procesni elementi

Perceptron • Pravilo perceptrona • Primjer • Učenje booleovih funkcija • Linearna odvojivost • XOR problem • Višeslojna mreža perceptrona • Linearna regresija • Srednja kvadratna pogreška • Gradijentni spust • LMS algoritam

---

## 3. Backpropagation algoritam

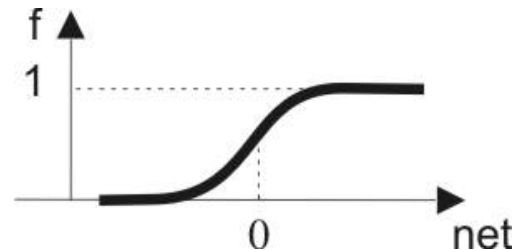
BACKPROPAGATION algoritam • Primjeri uporabe neuronskih mreža • Primjer sustava za raspoznavanje

### 3.

## BACKPROPAGATION algoritam

- ANN s perceptronima može predstaviti samo linearne odnose
- Kako bi mreža mogla predstaviti visoko nelinearne funkcije, prijenosna funkcija procesnih elemenata mora i sama biti **nelinearna funkcija**
- Radi gradijentne metode prijenosna funkcija mora biti i **derivabilna**
- Rješenje ► sigmoidalna funkcija

$$f(net) = \frac{1}{1 + e^{-a \cdot net}}$$

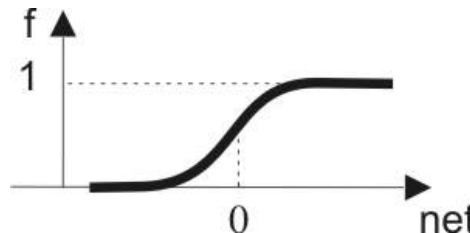


koja ima prvu derivaciju  $df(x)/dx = f(x) [1-f(x)]$

### 3.

## BACKPROPAGATION algoritam

- Koristimo neuron sa sigmoidalnom funkcijom



- Algoritam koristi metodu gradijentnog spusta kako bi minimizirao nastalu pogrešku  $E(\vec{w})$  na izlazu mreže nad skupom primjera za učenje  $D$

$$E(\vec{w}) = \frac{1}{2} \sum_{d \in D} (t_d - o_d)^2$$

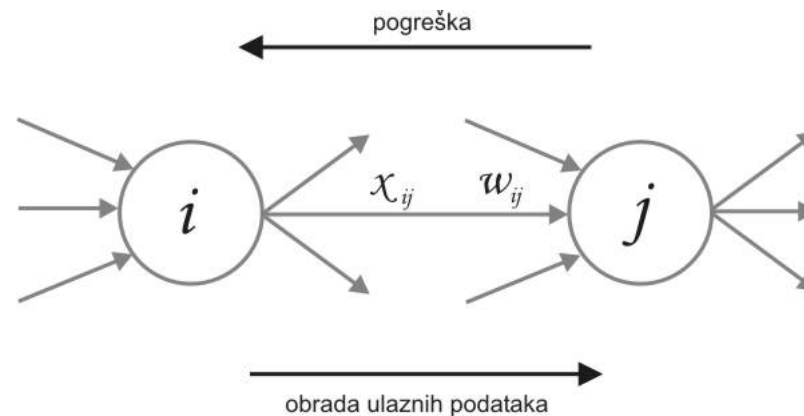
- Kod višeslojne mreže izlazni sloj može sačinjavati više neurona, pa definiramo pogrešku kao

$$E(\vec{w}) = \frac{1}{2} \sum_{d \in D} \sum_{k \in \text{outputs}} (t_{kd} - o_{kd})^2$$

# 3.

## BACKPROPAGATION algoritam

- Učenje se svodi na pretraživanje u  $n$ -dimenzionalnom prostoru hipoteza ( $n = \text{ukupan broj težina}$ )
- Površina pogreške  $E(\mathbf{w})$  nije parabolička kao kod jednog procesnog elementa, već sadržava brojne **lokalne minimume**
- Unatoč tome algoritam daje dobre rezultate (osobito *stohastička* varijanta)
- Notacija:



### 3.

## BACKPROPAGATION algoritam

**Inicijaliziraj** težinske faktore slu ajne vrijednosti

**Dok** nije ispunjen uvjet zaustavljanja **ini**

**Za** svaki  $(\mathbf{x}, \mathbf{t})$  iz  $D$  **ini**

Izra unaj izlaz  $o_u$  za svaku jedinicu  $u$

Za svaku **izlaznu** jedinicu  $k$  izra unaj pogrešku  $\delta_k$

$$\delta_k \leftarrow o_k(1-o_k)(t_k - o_k)$$

Za svaku **skrivenu** jedinicu izra unaj pogrešku

$$\delta_h \leftarrow o_h(1-o_h) \sum_{s \in \text{Downstream}(h)} \omega_{hs} \delta_s$$

Ugodi svaki težinski faktor  $w_{ij}$

$$\omega_{ij} \leftarrow \omega_{ij} + \Delta \omega_{ij}$$

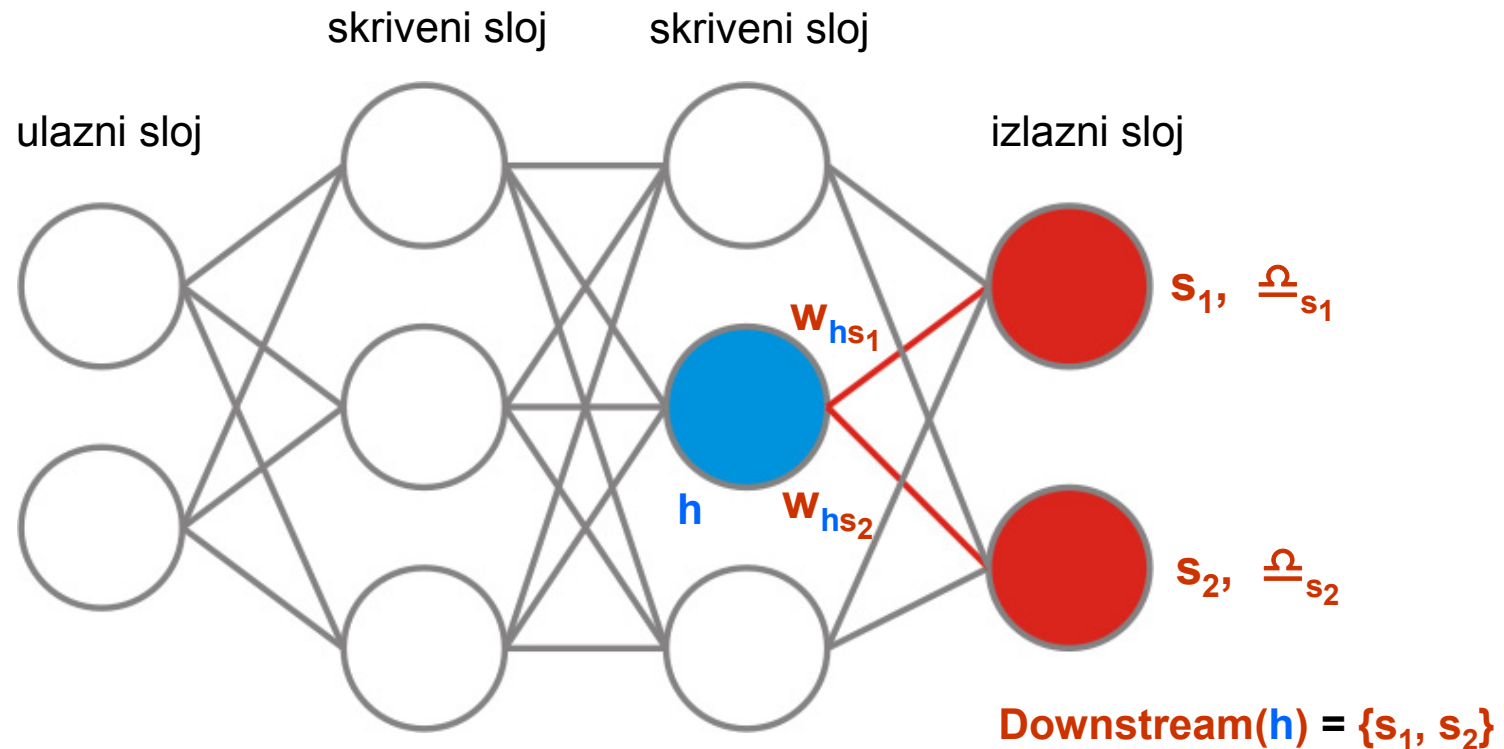
gdje je  $\Delta \omega_{ij} = \eta \delta_j x_{ij}$

**Kraj**

**Kraj**

# 3.

## BACKPROPAGATION algoritam



$$\delta_{s_1} = o_{s_1} (1 - o_{s_1}) (t_{s_1} - o_{s_1})$$

$$\delta_{s_2} = o_{s_2} (1 - o_{s_2}) (t_{s_2} - o_{s_2})$$

$$\delta_h = o_h (1 - o_h) (w_{hs_1} \delta_{s_1} + w_{hs_2} \delta_{s_2}) \quad \& \quad \delta_h \text{ će služiti za ugađanje ulaza u } h$$

### 3.

## Primjer: Raspoznavanje novčanica

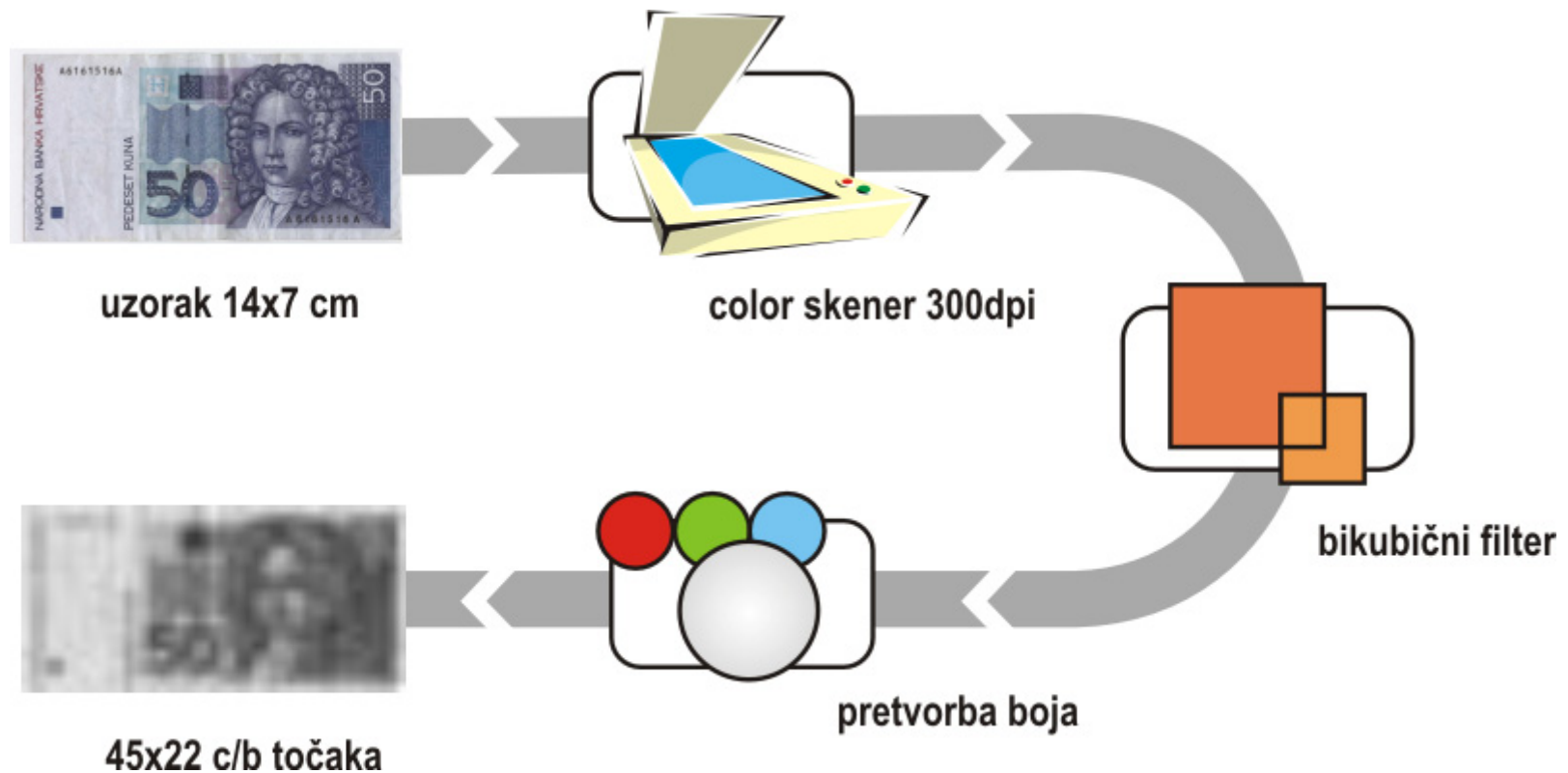
- **Zadatak** ► klasificiranje četiri vrste papirnatih novčanica neovisno o orijentaciji
- Skup primjera za učenje sačinjava **16** različitih uzoraka uzorkovanih s **45x22** slikovna elementa:



### 3.

## Primjer: Raspoznavanje novčanica

- Uzorci se prije dovođenja na ulaze ANN predprocesiraju:





### 3.

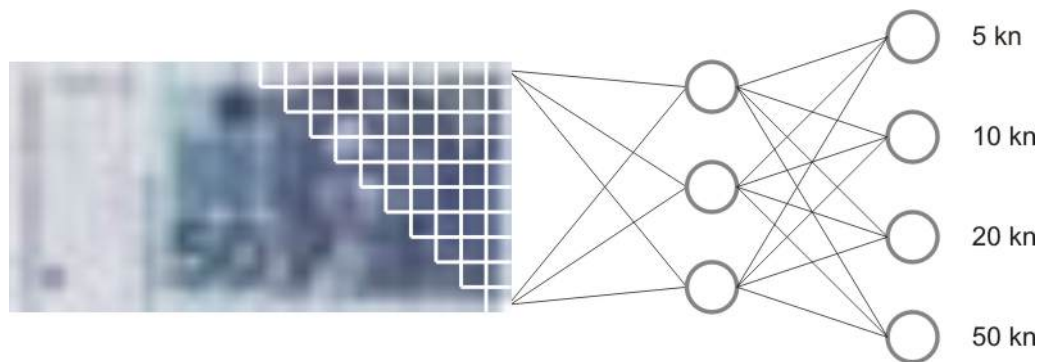
## Primjer: Raspoznavanje novčanica

- Uzorci su u stvari digitalizirane slike, pa će se međusobno razlikovati u intenzitetu odgovarajućih slikovnih elemenata (posljedica razlike u stupnju istrošenosti novčanice, izgužvanosti papira, oštećenosti).
- U primjeru je korišten **generator umjetnih uzoraka** koji generira uzorke različitog stupnja oštećenja u svrhu provjere rada mreže i ugađanja njezinih parametara.
- Prednosti:
  - dostupnost velikog broja uzoraka za provjeru,
  - mogućnost promatranja odziva mreže u ovisnosti o promjeni samo jednog parametra uzorka,
  - mogućnost definiranja praga tolerancije oštećenih uzoraka.
- Nedostaci:
  - generirani uzorci nisu u potpunosti stohastičke prirode,
  - generatorom se rijetko mogu obuhvatiti sva obilježja pravih uzoraka.

### 3.

## Primjer: Raspoznavanje novčanica

- **Parametri mreže** ▶ aciklička potpuno spojena uniformno-slojevita mreža strukture **990x3x4**, stopa učenja=**0.02**, moment=**0.02**, provjera nad skupom za testiranje svakih **2500** epoha



- Rezultat klasifikacije odgovara neuronu čiji je izlaz najveći
- Mreža nije učena s anti-primjerima, no međusoban odnos izlaznih vrijednosti može poslužiti kao **mjera pouzdanosti** klasifikacije
- Moguće je utvrditi neki prag tolerancije oštećenja uzorka



- T. M. Mitchell, *Machine Learning*. The McGraw-Hill Companies, Inc., 1997.
- R. S. Michalski, I. Bratko, M. Kubat, *Machine Learning And Data Mining*, John Wiley & Sons Ltd., 1998
- P. Picton, *Neural Networks*. PALGRAVE, 1994.
- B. Dalbelo Bašić, Bilješke s predavanja. Fakultet elektrotehnike i računarstva, Zagreb, 2001.
- K. Gurney, "Computers and Symbols versus Nets and Neurons". Dept. Human Sciences, Brunel University, Uxbridge, 2001.
- K. Gurney, "Drawing things together – some perspectives". Dept. Human Sciences, Brunel University, Uxbridge, 2001.
- D. Mišljenčević, I. Maršić, *Umjetna inteligencija*. Školska knjiga, Zagreb, 1991.
- *Automata Theory*. Encyclopaedia Britannica Inc., 2001 CD-ROM Edition.