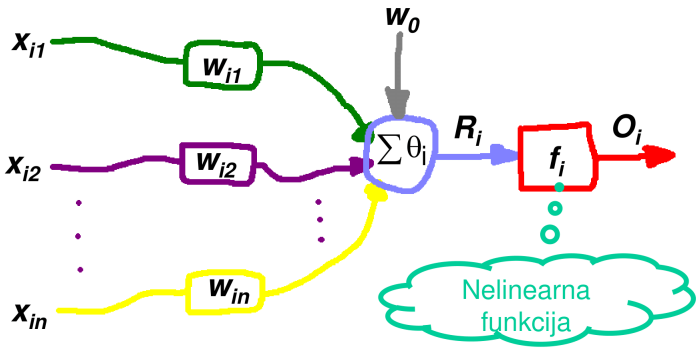
Osnovni koncepti vještačkih neuronskih mreža

- **Konekcionizam** - teorijsko stanovište koje želi objasniti ljudske intelektualne sposobnosti konstruisanjem vještačke neuronske mreže.
  - Čovjekov mozak sadrži nekoliko tipova ćelija, a sveukupno 10^{11} neurona koji ostvaruju 10^{15} konekcija.
  - Ćelije međusobno komuniciraju tako što prenose signal iz jedne ćelije u drugu, uz pomoć *neurotransmitera*.
- Cilj **vještačkih neuronskih mreža** je definisati osnovne principe bioloških neuronskih mreža matematičkim opisom.
- **Karakteristike neuronskih mreža**:
  - *Arhitektura*
    - Definiše strukturu *NM*, kao što je broj neurona i način njihove međusobne povezanosti.
    - *NM* se sastoje od određenog broja međusobno povezanih neurona, sa sličnim karakteristikama kao što su *ulazi, sinapse, sinaptičke težine, aktivacija, izlazi, bias,...*
      - *Sinapse* - veze od dendrita ka ćelijama. Postoji otpor u sinapsama, modeluje se određenim *težinama*.
      - *Aktivacija* - prag koliko se mora "nakupiti" signala da bi neuron dao izlaz.
      - *Bias* - otežanje u samom neuronu.
  - *Neurodinamika*
    - Definiše "ponašanje" *NM*, odnosno kako one uče, kako pozivaju naučeno, kako vrše grupisanje, kako upoređuju nove podatke sa već postojećim znanjem, kako klasifikuju nove informacije i kako razvijaju novu klasifikaciju.
- *VNM* vrše procesiranje informacija.
  - Takvo procesiranje je pogodno za *paralelizaciju* - više neurona istovremeno razlažu ulaznu informaciju, odrađuju je i davaju izlaz.
  - U poslednjih nekoliko desetina godina uložen značajan tehnološki trud kako bi se projektovala kola koja bi ličila na biološke *NM* sa svim karakteristikama.
  - Na taj način razvijeni su različiti modeli *NM* poznati pod nazivom **paradigme**.
  - Danas glavni zadatak jeste obrada slike, odnosno videa.
- Svi modeli *VNM* tradicionalno se opisuju *diferencijalnim* (*diferencnim*) *jednačinama*.
- Najbolje *VNM* idalje daleko inferiornije od najprimitivnijih oblika života.
- Da bi se razvio model *VMN* potrebno je razviti matematički model koji na najbolji mogući način opisuje funkcionalnost prirodnog sistema.

- Tada je moguće izvršiti računarsku simulaciju i ispitati u kojoj se mjeri slažu rezultati simulacije sa realnim ponašanjem biološke *NM*.
- Vrše se *korekcije*:
  - Ako su rezultati **loši** - promjena parametara i strukture (usložnjavanje modela) (problem *underfitting*-a).
  - Ako su rezultati **dobri** - uprošćavanje modela (problem *overfitting*-a).

Osnovni model neurona

- **Neuron** - sastavni element svake *NM*.
  - Pod ovim nazivom podrazumijeva se procesni element *VNM* u smislu svoje funkcije, odnosno načina djelovanja
  - Ima n ulaza označenih sa $x_j, j = 1, 2, \ldots, n$.
  - Svaki ulazni signal otežan prije nego što dođe do tijela procesnog elementa, odnosno neurona
    - Otežavanje se vrši množenjem vrijednosti ulaznog signala x_j sa njegovom odgovarajućom **težinom sinapse** w_j.
  - Takođe, postoji i slobodni član **bias** - w_0 i prag (**treshold**) - θ, koji određuju potrebni nivo signala za **aktivaciju** neurona.
  - Nelinearna funkcija f_j djeluje na pobudni signal R_j i formira izlaz neurona O_j.
    - Ovakva funkcija se naziva **aktivaciona funkcija**.
  - Izlaz neurona O_j predstavlja **ulaz** za neke druge neurone.
  - Često se neuron naziva i **čvor**.
    - Ako se *VNM* sastoji od više čvorova, dodaje se još jedan indeks koji služi da označi kom neuronu signal ili funkcija pripada, tako da je
      - x_{ij} - j-ti ulaz i-tog neurona.
      - w_{ij} - težina j-te sinapse i-tog neurona.

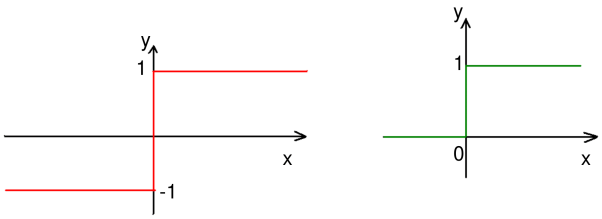


$$O_i = f_i\left(\sum_{j=1}^{n} w_{ij}x_{ij} + w_0\right)$$

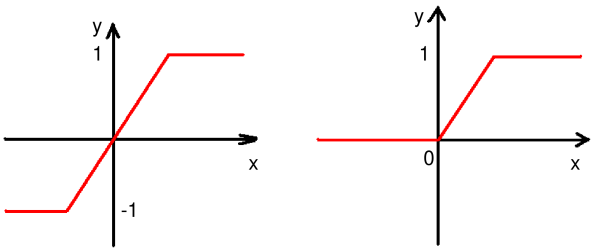
$$\sum_{j=1}^{n} w_{ij}x_{ij} + w_0 \geq \theta_i$$

- Nelinearna funkcija se uvodi radi **ograničavanja** nivoa izlaznog signala (nekad neuron može dati izlazni signal koji je reda veličine beskonačnosti, te je potrebno ograničenje - ništa u prirodi ne može dati signal beskonačne amplitude).
- Najčešći tipovi nelinearnih funkcija:
  - *Hard Limit*
  - *Rampa*
  - *Sigmoida* (veoma popularna jer je **monotona**, **ograničena** i **jednostavno joj se računa prvi izvod**)
  - *Gausijan*

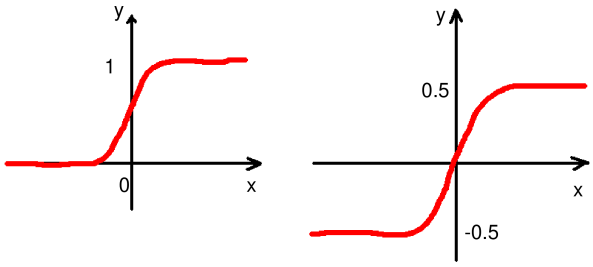
Hard limit



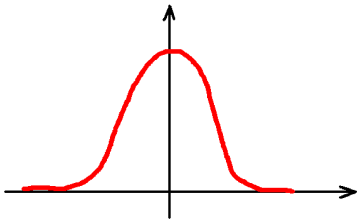
Rampa



Sigmoida



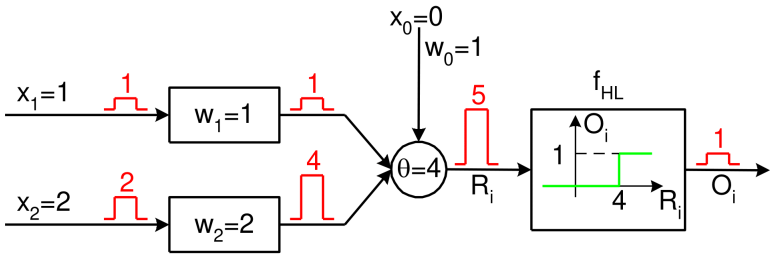
Gausijan



O analitičkim izrazima ovih funkcija će biti reči kasnije

- Primjer rada jednog neurona:

Primer rada jednog neurona



Učenje u *VNM*

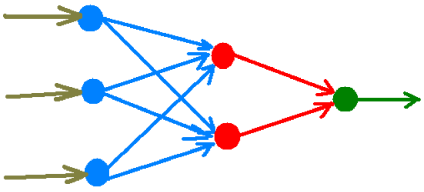
- Proces učenja u *VNM* je proces podešavanja **promjenljivih težina sinapsi** (w_{ij}) u cilju postizanja odgovarajućeg izlaza O_j za dati pobudni signal x_{ij}.
  - Kada postignemo takve rezultate, kažemo da je *VNM* **obučena**, odnosno da je "stekla znanje".
  - Obuka se vrši prema algoritmima koji se opisuju jednačinama obuke.
- Tipovi algoritama obuke:
  - *Učenje sa nadzorom* (*Supervised learning*)
  - *Učenje bez nadzora* (*Unsupervised learning*)
  - *Pojačano učenje* (*Reinforcement learning*)
  - *Kompetitivno učenje* (*Competitive learning*)
  - *Delta pravilo* (*Delta rule - LMS*)
  - *Metoda opadajućeg gradijenta* (*Gradient descent rule*)
  - *Hebeovo učenje* (*Hebbian learning*)

Osnovne karakteristike *VNM*

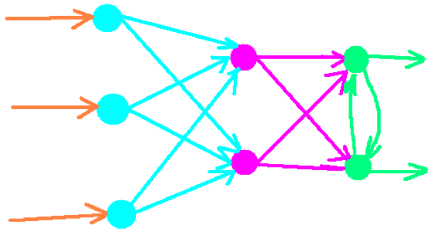
- **Kolektivna obrada podataka**
  - Program se izvršava kolektivno, sinergično, paralelno pa su time operacije decentralizovane.
- **Robusnost**
  - Operacije su neosjetljive na slučajne poremećaje i netačne ulaze.
- **Učenje**
  - *VNM* automatski uspostavlja preslikavanja (asocijacije), adaptira se "sa ili bez učitelja", ali svakako bez intervencije programera.
- **Asinhrono izvršavanje**
  - *VNM* zahtjevaju vremensko usklađivanje.
- **Performanse**
  - Govore u kojoj je mjeri *VNM* sposobna da u procesu eksploatacije reprodukuje rezultate dobijene na skupu za obuku.
  - 100% performansi je idealno, ali potrebno je da postoje granice.
- **Topologija *VNM***
  - Kod svih topologija, mogu se uočiti sledeći elementi
    - *Ulazni sloj*
    - *Izlazni sloj*
    - *Skriveni slojevi*

Karakteristika	Feed-forward NN	Recurrent NN
Smer toka signala	Samo napred	Dvosmerno
Uvođenje kašnjenja	Ne	Da
Kompeksnost	Niska	Visoka
Nezavisnost neurona u istom sloju	Da	Ne
Brzina	Velika (brze)	Mala (spore)
Primena	Prepoznavanje oblika, govora, karaktera, …	Prevod, konverzija govora u tekst, robotsko upravljanje, …

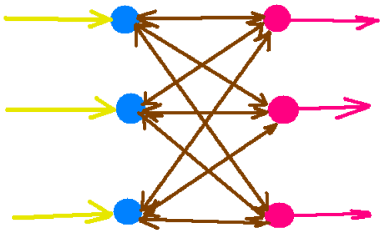
Višeslojni perceptron
Višeslojna VNM sa prostiranjem signala u napred
Multilayer Feed Forward ANN



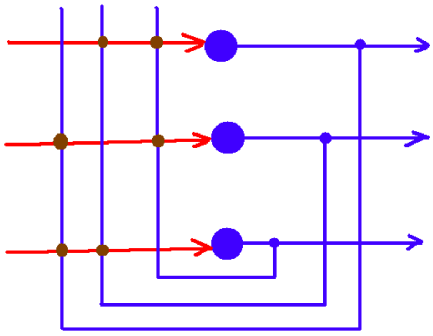
Višeslojna kooperativno/kompetetivna VNM
Multilayer cooperative/competitive ANN



Dvoslojna VNM sa prostiranjem signala napred/nazad
Bilayer feed forward/backward ANN



Jednoslojna VNM sa kombinovanom povratnom spregom
Monolayer hetero feedback ANN



- **Broj slojeva u *VNM***
- **Broj neurona po sloju**
- **Usvajanje algoritma obuke**
- **Broj iteracija po uzorku tokom treninga**
- **Brzina rada u eksploataciji**
- **Kapacitet *VNM***
  - Maksimalan broj uzoraka koji *VNM* može da nauči i kasnije pozove.
- **Stepen adaptivnosti**
  - U kom stepenu je *VNM* sposobna da se adaptira poslije okončanja procesa obuke.
- **Vrijednost *bias*-a**
  - Često unaprijed postavljeno.
- **Vrijednost praga (*threshold*)**
  - Često unaprijed postavljeno na neku fiksnu težinu.
- **Ograničenje težina sinapsi**
  - Za bolje performanse i otpornost prema šumu.
- **Izbor nelinearne aktivacione funkcije**
- **Otpornost *VNM* na šum**
  - Stepen u kojem šum, smetnja i anomalija na ulaznom signalu uzrokuje šum na izlaznom signalu.
- **Vrijednost težina u stacionarnom stanju**
  - Stanje *VNM* nakon obuke

Modelovanje *VNM*

- Matematičkom analizom može se doći do sledećih podataka o mreži
  - **Kompleksnost**
    - Koliko bi trebalo da *VNM* bude velika da bi izvršila zadatak.
  - **Kapacitet**

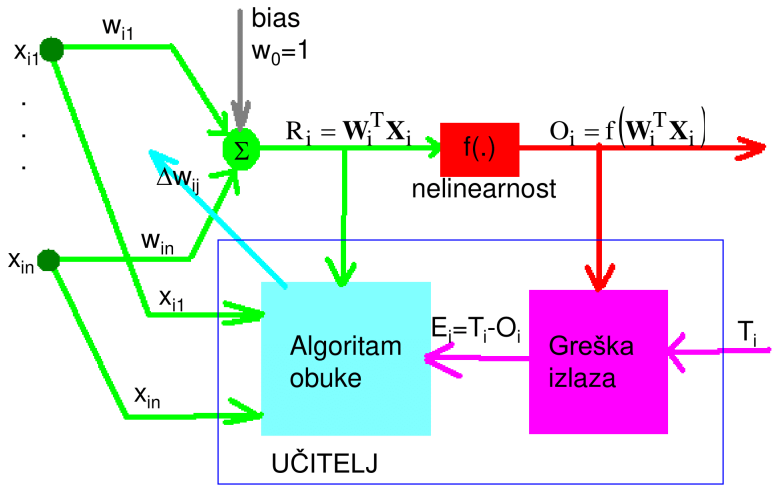
    - Koliko bita informacija se može pohraniti u jednoj *VNM*.
- **Izbor modela**
    - Koji tip *VNM* je najpogodniji za datu primjenu.
- **Performanse**
    - Koja *VNM* ima najbolje rezultate i performanse.
    - Koliko brzo uči.
    - Kolikom brzinom *VNM* daje odziv nakon dejstva signala na ulazu.
- **Pouzdanost**
    - Da li *VNM* daje uvijek isti odziv za istu pobudu.
- **Osjetljivost na šum**
    - Koliko tačno *VNM* reprodukuje željeni izlaz u prisustvu šuma.
- **Osjetljivost na otkaz**
    - Koliko tačno *VNM* radi ako jedan njen dio ne funkcioniše.

Obuka i programiranje *VNM*

- Matematički opis načina kako se mijenjaju težine sinapsi w_{ij} tokom procesa obuke *VNM* naziva se **algoritam obuke**.

Obuka sa nadzorom (*Supervised learning*)

- Iterativni postupak, zahtjeva više prolaza kroz skup za obuku.



Generalno: promena težina je proporcionalna signalu greške tokom obuke i stimulaciji (ulazu) neurona.

- Obuka i-tog neurona može se opisati izrazom

$$\frac{\partial w_{ij}}{\partial t} = \mu E_i(O_i, T_i) X_i(t)$$

  gdje je μ mala pozitivna konstanta poznata kao **korak obuke** (*learning rate*).
- Diskretna formulacija bi glasila

$$w_{ij}(k+1) = w_{ij}(k) + \underbrace{\mu E_i(O_i, T_i) X_i(k)}_{\text{iznos korekcije}}$$

Algoritmi obuke

- Matematički alat koji predstavlja metod kojim će se određenom brzinom uspješno doći do stacionarnog stanja parametara - težina i pragova *VNM*.
- Obuka počinje definisanjem **kriterijumske funkcije** (funkcije greške) koju je potrebno minimizirati.
  - Izražava se preko težina i pragova *VNM*, te se na taj način obezbjeđuje da kriterijumska funkcija bude u direktnoj vezi sa promjenljivim veličinama *VNM*.

Primer obuke VNM bez nadzora

T_i – vektor željenih izlaza, **ne postoji (nije poznat)**

$f = sgn(.)$ – nelinearnost je bipolarna funkcija (+1,-1)

$\boldsymbol{w}_{ij} = [w_{i1}\ w_{i2}\ ...\ w_{in}]$ – vektor težina ulaza *i*-tog neurona, na početku obuke male pozitivne i negativne vrednosti

$\boldsymbol{x}_{ij} = [x_{i1}\ x_{i2}\ ...\ x_{in}]$ – vektor ulaza *i*-tog neurona

μ – korak učenja; mali, pozitivan broj

k – broj iteracije

$$O_i(k) = f\left(\sum_{i=1}^{n} w_{ij}(k) x_{ij}\right)$$

$$\Delta w_i = \mu O_i(k) x_{ij}$$

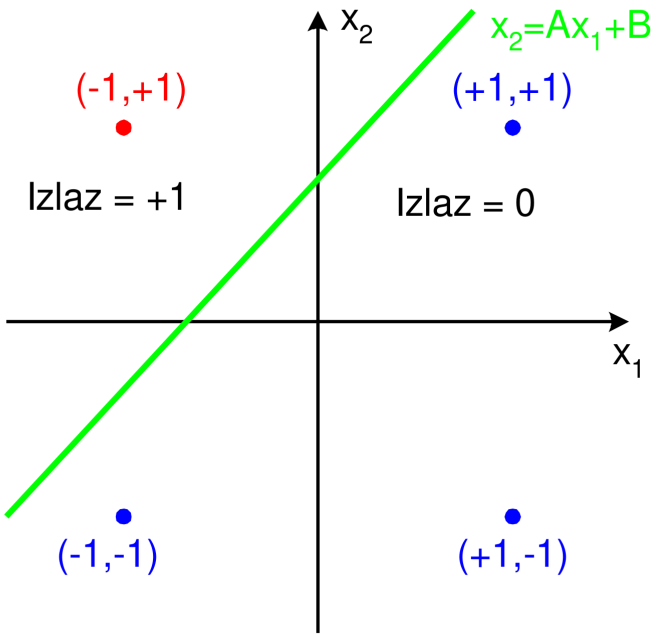
$$w_i(k+1) = w_i(k) + \mu O_i(k) x_{ij}$$

- Klasifikacioni problemi
  - Obučena *VNM* sposobna je prepoznati i klasifikovati različite vrste podataka.
  - *Sposobnost klasifikacije* - maksimalan broj tačaka koje *VNM* može da klasifikuje, odnosno tačno identifikuje, i za koje može da generiše jedinstveni izlaz.
  - Prema sposobnosti klasifikacije, *VNM* se dijele na
    - *Linearne*
    - *Multilinearne*
    - *Nelinearne*

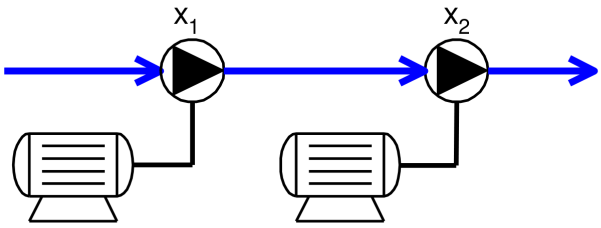
Linearni klasifikatori

VNM - linearni klasifikatori

x_1	x_2	Y
+1	+1	0
-1	+1	+1
+1	-1	0
-1	-1	0

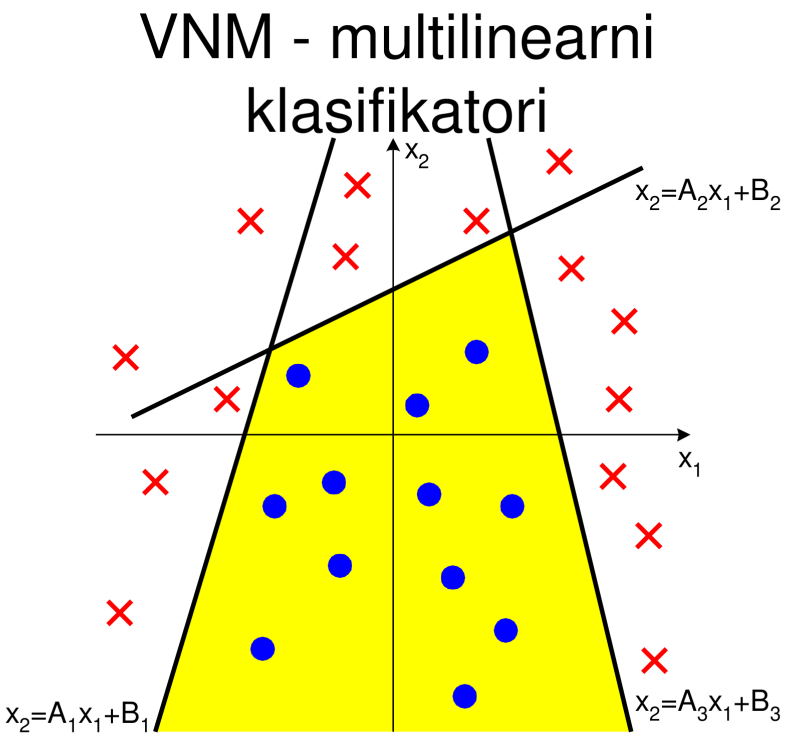


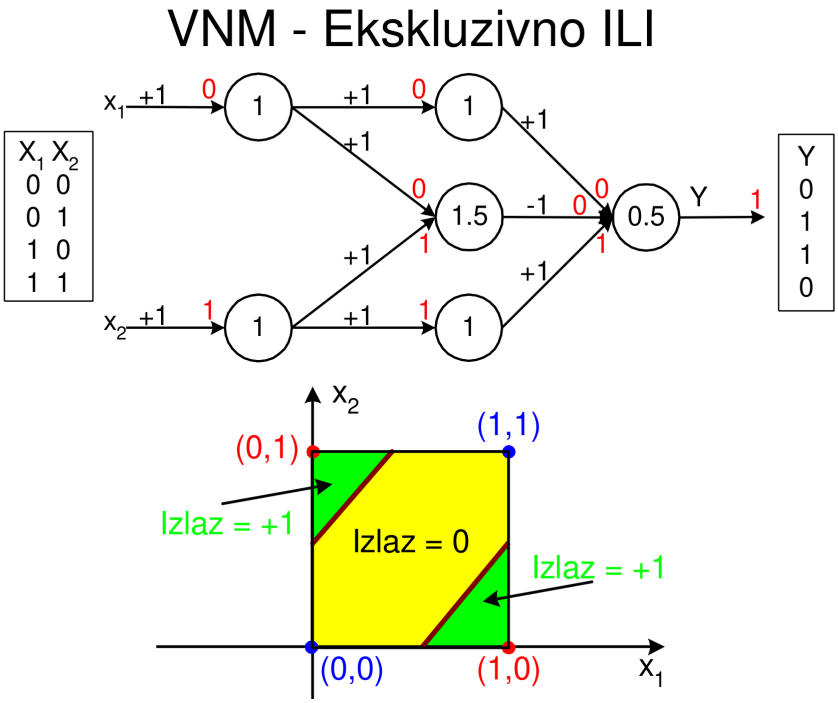
Kako se ovo može upotrebiti?



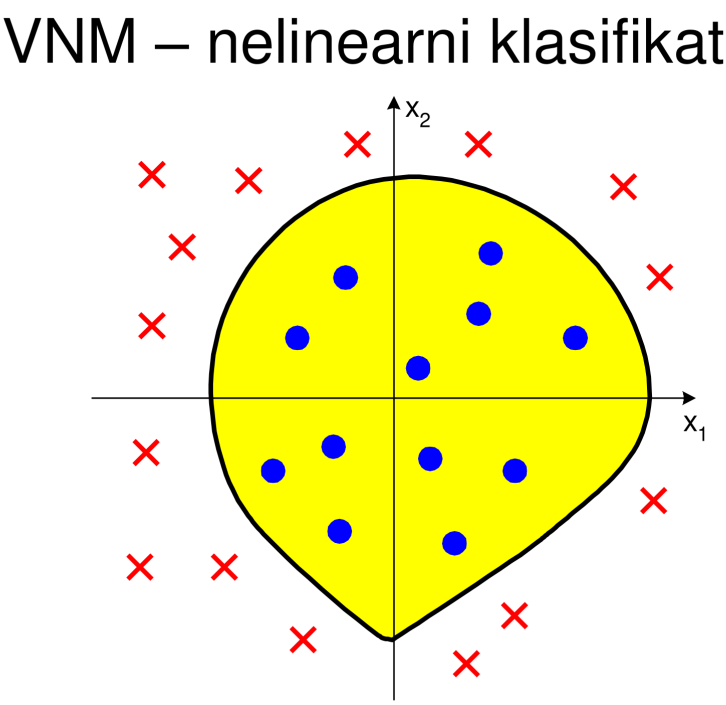
x_1	x_2	Y
+1 (ON)	+1 (ON)	0 (OK)
-1 (OFF)	+1 (ON)	+1 (Alarm)
+1 (ON)	-1 (OFF)	0 (OK)
-1 (OFF)	-1 (OFF)	0 (OK)

Multilinearni klasifikatori





Nelinearni klasifikatori



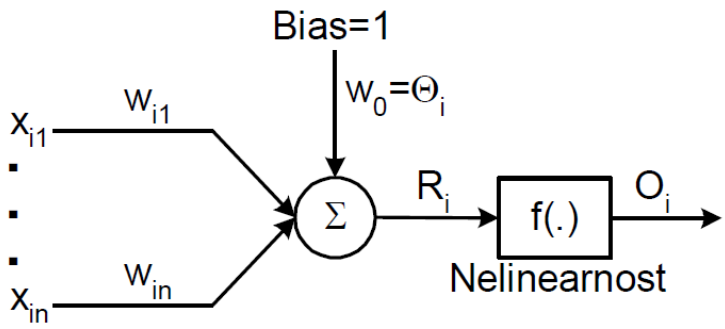
KRAJ

Perceptron

McCulloh - Pitt-ov model neurona

- Prva generacija, prvi pokušaj da se modeluje biološki neuron.
- Ne postoji obuka, kao ni adaptacija parametara.
- Izlaz definisan izrazom

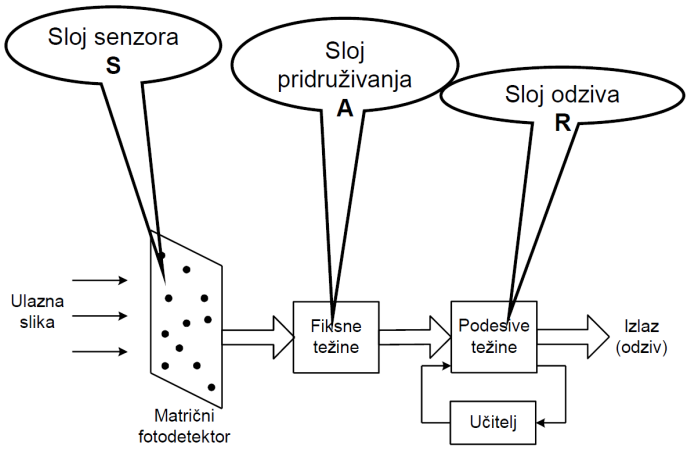
$$O_i = f\left(\sum_{j=1}^{n} x_{ij} w_{ij} - \Theta(i)\right)$$



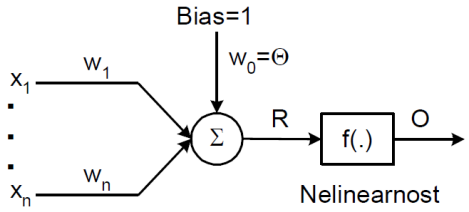
Model perceptrona

- Naredna generacija vještačkih neuronskih mreža, bazirana na *McCulloh-Pit*-ovom modelu uz dodavanje povratne sprege, odnosno mogućnosti *učenja* i *adaptacije*.
- Originalni perceptron zahtjeva učenje sa nadzorom.

- Razvijen je kao klasifikator uzoraka koji raspoznaje apstraktne i geometrijske oblike dovedene na optički ulaz.
- Sastoji se iz tri nivoa:
  - *Sloj senzora*
  - *Sloj pridruživanja*
  - *Sloj odziva*



Model pojedinačnog perceptrona



Greška izlaza perceptrona: $\boldsymbol{E = T - O}$

Korekcija težina sinapsi se računa prema izrazu:

$$\Delta \boldsymbol{w} = \mu[\boldsymbol{T} - f(\boldsymbol{w}(k)\boldsymbol{x})]\boldsymbol{x}$$

odnosno:

$$\boldsymbol{w}(k+1) = \boldsymbol{w}(k) + \mu[\boldsymbol{T} - f(\boldsymbol{w}(k)\boldsymbol{x})]\boldsymbol{x}$$

- Pripremni koraci za obuku neurona su:
  - Odrediti skup ulaznih vektora x
  - Odrediti skup željenih izlaza T, gdje jedan izlazni podatak odgovara jednom ulaznom
  - Izabrati mali pozitivan korak učenja μ i kriterijum po kojem će se taj parametar mijenjati tokom obuke
  - Izabrati nelinearnost (aktivacionu funkciju neurona) i ako je potebno, njene parametre
  - Odrediti kriterijum za završetak obuke (maksimalna dozvoljena greška izlaza, maksimalni broj iteracija,...)

- Obuka se vrši kroz nekoliko koraka:
  - Izabrati inicijalne vrijednosti za pragove Θ_i i težine w_{ij} kao male slučajne vrijednosti (najčešće u intervalu $[-1, 1]$)
  - Dovesti na ulaz uzorak x_p i odgovarajući izlaz T_p, gdje je p broj tekućeg uzorka
  - Izračunati tekući izlaz O, prema izrazu

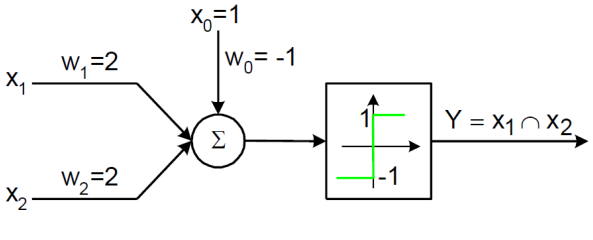
$$O(k) = f\left(\sum_{j=0}^{n} w_j(k)x_j\right) = f(w^T(k)x)$$

  - Izvršiti adaptaciju težina prema iterativnoj relaciji

$$w(k+1) = w(k) + \mu(T(k) - f(w^T(k)x)))x$$

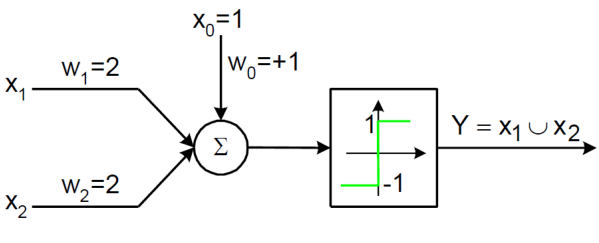
  - Provjeriti kriterijum zaustavljanja
- *VNM* sačinjena od jednog perceptrona (jednoslojni perceptron) može se smatrati klasifikatorom za rješavanje problema razgraničenja dvije klase podataka u okviru nekog uzorka. Takav perceptron je i *logička jedinica* i u zavisnosti od vrijednosti njegovih parametara, može implementirati različite logičke funkcije (ali ne sve).

Logička operacija "I"



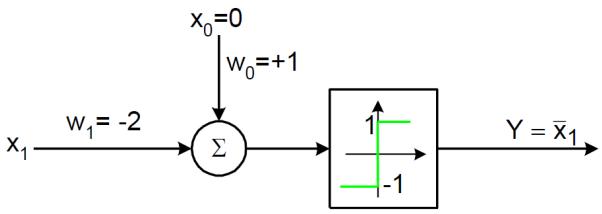
x_1	x_2	Y
-1	-1	-1
-1	+1	-1
+1	-1	-1
+1	+1	+1

Logička operacija "ILI"



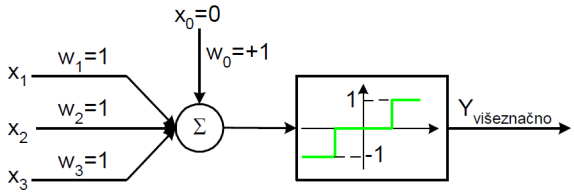
x_1	x_2	Y
-1	-1	-1
-1	+1	+1
+1	-1	+1
+1	+1	+1

Logička operacija “NE”



x_1	Y
-1	+1
+1	-1

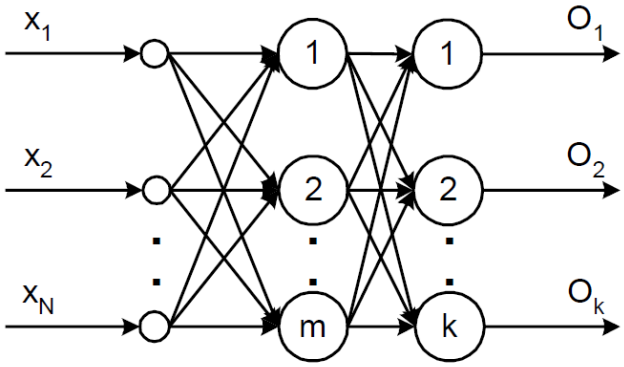
Logička operacija “PLURALNOST”



Bilo koji ulaz			Y
0	0	0	0
0	-1	+1	0
0	0	+1	+1
0	+1	+1	+1
+1	+1	+1	+1
-1	+1	+1	+1
0	0	-1	-1
0	-1	-1	-1
-1	-1	-1	-1
+1	-1	-1	-1

Višeslojni perceptron

- Predstavlja mrežu sačinjenu od većeg broja običnih perceptrona koji čine hijerarhijsku strukturu sa prostiranjem signala u unaprijed (*Multilayer Feed Forward ANN*).
- Osnovni perceptroni organizovani u slojeve, kojih između ulaznog i izlaznog može biti i više - *broj skrivenih slojeva nije fiksan*.
- Svaki perceptron ima naziv **neuron**.
- Svaki sloj može sadržati različit broj neurona u zavisnosti od namjene (ne postoji preporuka niti formalan način da se odredi optimalan broj neurona po sloju - *metoda probe i greške*).
  - Više neurona - rješavanje složenijih problema ali problem *overfit*-ovanja
- Uobičajeni postupci obuke:
  - *Delta pravilo*
  - *Prostiranje greške unazad* (*Back propagation*)
- Može se koristiti za
  - Implementaciju proizvoljne Bulove logičke funkcije koja vrši podjelu prostora uzoraka (dovoljna su dva sloja)
  - Klasifikaciju uzoraka (dovoljna su dva ili tri sloja)
  - Implementacija nelinearnih funkcija (dovoljna su dva ili tri sloja)



Sigmoidna funkcija

- Često korištena aktivaciona funkcija
- Oblik

$$f_S(R) = \frac{1}{1 + e^{-kR}}$$

- Osobina prvog izvoda jeste da je

$$f'_S(R) = \frac{ke^{-kR}}{(1 + e^{-kR})^2} = kf_S(R)(1 - f_S(R)))$$

Delta pravilo

- Bazira se na metodi *minimizacije kvadrata greške*.
- Cilj pravila jeste da minimizuje razliku između željenog i stvarnog ***izlaza*** preko ulaza i težina *VNM* (primjetiti da se ovdje koriguju samo greške u **izlaznim težinama**).
- Na početku, definiše se kvadrat greške (E) na osnovu razlike stvarnog (O) i željenog izlaza (T) *VNM*

$$E = \frac{1}{2}(T_i - O_i)^2 = \frac{1}{2}(T_i - f(w_i^T x_i))^2$$

  gdje je w_i matrična reprezentacija skupa težina i-tog neurona a x vektor ulaza i-tog neurona.
- Vektor gradijenta greške je

$$\nabla E = -(T_i - O_i) f'(w_i^T x_i) x_i$$

  a kako se traži minimum greške, korekcija težina će se vršiti na osnovu negativnog gradijenta, odnosno

$$\Delta w_i = -\mu \nabla E$$

- Na osnovu gornja dva izraza dobija se da je

$$\Delta w_i = \mu (T_i - O_i) f'(w_i^T x_i) x_i$$

te su korigovane vrijednosti težina u narednoj $(k+1)$ iteraciji

$$w_i(k+1) = w_i(k) + \mu(T_i - O_i)f'(w_i^T x_i)x_i$$

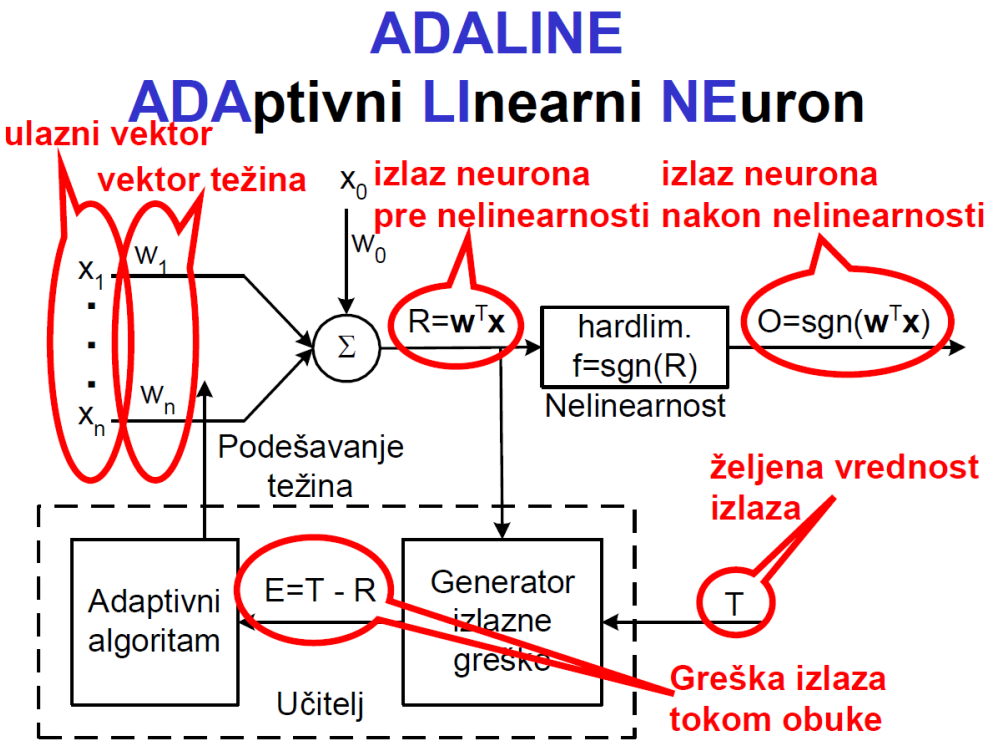
- Ako se usvoji da je aktivaciona funkcija *sigmoida* i parametar $k = 1$, dobijamo da je

$$w_i(k+1) = w_i(k) + \mu(T_i - O_i)(O_i - O_i^2)x_i$$

ADALINE (Adaptivni linearni neuron) i *MADALINE* (*Many ADALINE*)

- Aktivaciona funkcija *hard limit*.
- Izlaz binarni (1 ili -1).
- Greška se ne računa kao razlika željene i ostvarene vrijednosti izlaza već kao razlika željene vrijednosti i izlaza neurona prije nelinearnosti

$$E = T - R$$

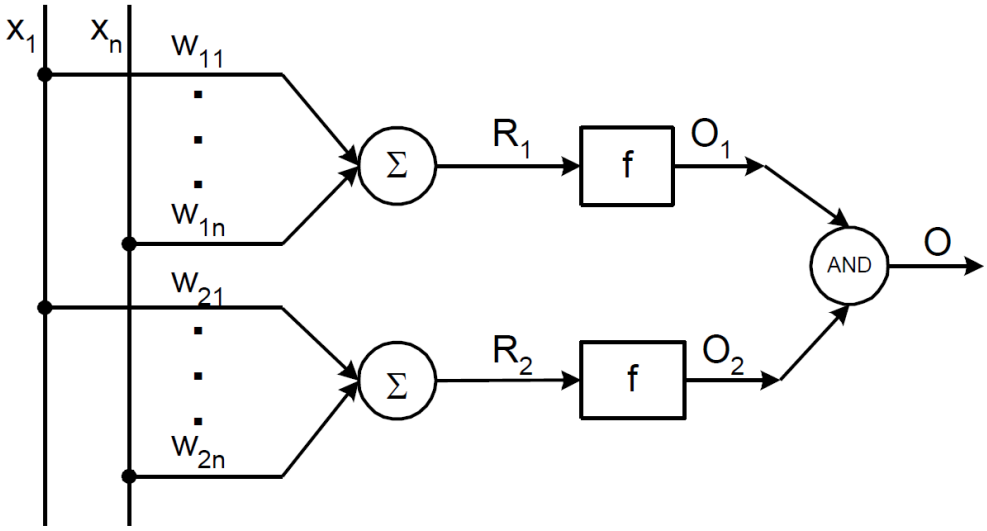


- Korekcija sinapsi (težina) vrši se prema izrazu

$$\Delta w_{ij} = \alpha(T - R)x_{ij}$$

gdje $0 < \alpha < 1$.

- ***MADALINE*** se formira povezivanjem u paralelni rad više *ADALINE*-a.
  - Svakoj od *ADALINE* na ulaz dovodi se isti signal.
  - *MADALINE* ima jedan izlaz koji se formira od izlaza svih izlaza *ADALINE*-a koristeći određena pravila. Ako se primjeni *logičko I*, tada izlaz *MADALINE* ima vrijednost 1 samo kada izlaz svih *ADALINE* perceptrona bude 1.



Algoritam prostiranja greške unazad (*Back Propagation Learning Algorithm*)

- Najčešće korišćen alogirtam obuke za višeslojne *VNM* sa prostiranjem signala unaprijed.
- *VNM* obučena ovim algoritmom sposobna je aproksimirati funkcije sa visokim stepenom nelinearnosti.
- Obukom se vrši korekcija težina sinapsi koje povezuju neurone sa svim slojevima, pa i skrivenim.
  - Korekcija se vrši na osnovu greške izlaza neurona koja se unazad propagira.
  - Iterativni postupak, jednostavan za primjenu na računaru.
  - Cilj obuke - odrediti skup težina sinapsi *VNM* za koji će greška izlaza biti minimalna.
  - Prije početka obuke odrediti *obučavajući skup*, *vrijednost koraka učenja*, *kriterijum zaustavljanja algoritma*, *način korekcije težina sinapsi*, *aktivacionu funkciju* (najčešće *sigmoida*) i *inicijalne vrijednosti težina sinapsi* (obično mali slučajni brojevi).

Matematička analiza procesa obuke

Matematička analiza procesa obuke

Posmatra se VNM sa prostiranjem signala u napred i sledećim parametrima:

VNM se sastoji od L slojeva i N_l neurona na sloju l;

w^l_{ij} – težina između i-tog neurona na sloju l-1 i j-tog neurona na sloju l;

$O^l_j(\boldsymbol{x}_p)$ – aktuelni izlaz j-tog neurona na sloju l za p-ti ulazni uzorak (nakon nelinearnosti, aktivacione funkcije);

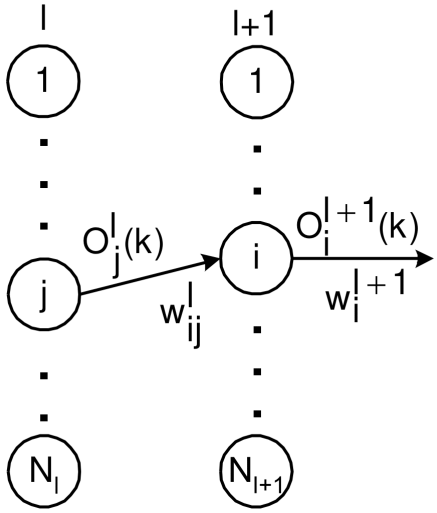
$T^L_j(\boldsymbol{x}_p)$ – željeni izlaz j-tog neurona na sloju L za p-ti ulazni uzorak;

$R^l_j(\boldsymbol{x}_p)$ – aktivacioni izlaz j-tog neurona na sloju l za p-ti ulazni uzorak (pre nelinearnosti, aktivacione funkcije);

$\boldsymbol{P}$ – skup za obuku;

$\boldsymbol{x}_p$ – p-ti obučavajući uzorak, p-ti elemenat skupa za obuku.

U cilju ilustracije BP algoritma posmatra se *i*-ti neuron na sloju *l*+1 koji prima signale od *j*-tog neurona sa sloja *l*, preko težine w^l_{ij}.



Izlaz *i*-tog neurona sa sloja *l*+1, za *k*-ti ulazni obučavajući vektor je opisan izrazom:

$$O_i^{l+1}(k) = f\left(\sum_{j=1}^{N_l} w_{ij}^l O_j^l(k) - \theta_i^{l+1}\right) = f\left(\sum_{j=1}^{N_l+1} w_{ij}^l O_j^l(k)\right)$$

Ako se za aktivacionu funkciju usvoji sigmoida, važe izrazi:

$$f(x) = \frac{1}{1+e^{-\beta x}} \qquad f'(x) = \beta f(x)(1-f(x))$$

Ukupna greška (*E*) mreže, za ceo obučavajući skup sa *K* uzoraka se definiše kao suma kvadrata greški svih neurona sa izlaznog sloja *L*:

$$E = \sum_{k=1}^{K} E_k = \sum_{k=1}^{K}\left(\frac{1}{2}\sum_{i=1}^{N_L}\left[T_i(k) - O_i^L(k)\right]^2\right)$$

Cilj je odrediti skup svih težina VNM koji minimizira *E*

Pravilo obuke definiše da je promena težina srazmerna negativnom gradijentu greške izlaza:

$$\Delta w_{nm}^{l} \approx -\frac{\partial E_k}{\partial w_{nm}^{l}}$$

Da bi se odredila vrednost prethodnog izraza primenjuje se pravilo ulančavanja izvoda:

$$\frac{\partial E_k}{\partial w_{nm}^{l}} = \frac{\partial E_k}{\partial O_i^L(k)} \frac{\partial O_i^L(k)}{\partial w_{nm}^{l}}$$

Što se može napisati u obliku:

$$-\frac{\partial E_k}{\partial w_{nm}^{l}} = \sum_{i=1}^{N_L} \left(T_i(k) - O_i^L(k) \right) \frac{\partial O_i^L(k)}{\partial w_{nm}^{l}}$$

Pošto je pretpostavljena sigmoidna aktivaciona funkcija, za $l=L$-1 (težine izlaznog sloja) može se pisati sledeći izraz:

Na osnovu prethodnog izraza se piše:

$$-\frac{\partial E_k}{\partial w_{nm}^{l}} = \left(T_n - O_n^L \right) \beta O_n^L \left(1 - O_n^L \right) O_m^{L-1}$$

pa je izraz za korekciju težina sinapsi neurona izlaznog sloja:

$$\Delta w_{nm}^{L} = \eta \left[\left(T_n - O_n^L \right) O_n^L \left(1 - O_n^L \right) \right] O_m^{L-1}$$

korak učenja

Ako je sada $l \neq L$-1, $O^{L-1}{}_m$ i dalje zavisi od $w^l{}_{nm}$, i zavisnost greške od težina se ponovo može odrediti, primenom pravila ulančavanja:

$$-\frac{\partial E_k}{\partial w_{nm}^{l}} = \sum_{i=1}^{N_L} \left(T_i - O_i^L \right) f'\left(O_i^L \right) \sum_{j=1}^{N_{L-1}+1} w_{ij}^{L-1} \frac{\partial O_j^{L-1}}{\partial w_{nm}^{l}}$$

Ako je l=L-2 (poslednji skriveni sloj), prethodna jednačina se može napisti u obliku:

$$-\frac{\partial E_k}{\partial w_{nm}^{l}} = \sum_{i=1}^{N_L}\left(T_i - O_i^L\right) f'\left(O_i^L\right) w_{in}^{L-1} f'\left(O_n^{L-1}\right) O_m^{L-2}$$

odnosno:

$$-\frac{\partial E_k}{\partial w_{nm}^{l}} = f'\left(O_n^{L-1}\right)\left[\sum_{i=1}^{N_L}\left(T_i - O_i^L\right) f'\left(O_i^L\right) w_{in}^{L-1}\right] O_m^{L-2}$$

pa je izraz za korekciju težina sinapsi neurona poslednjeg skrivenog sloja:

$$\Delta w_{nm}^{L-2} = \eta\left[f'\left(O_n^{L-1}\right)\sum_{i=1}^{N_L}\left(T_i - O_i^L\right) f'\left(O_i^L\right) w_{in}^{L-1}\right] O_m^{L-2}$$

Winner Takes All algoritam

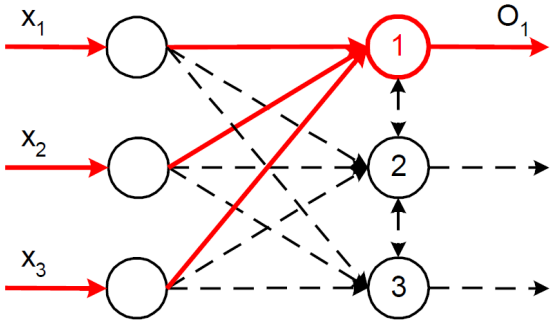
- Algoritam kompetitivne odluke bez nadzora.
- Pretpostavka je da *VNM* ima jedan sloj sa N neurona i da svakom neuronu odgovara vektor težina w_i.
  Svaki čvor (neuron) pobuđuje se istim setom ulaznih podataka tako da je odziv i-tog neurona

$$O_i = \sum_{j=1}^{m} w_{ij} x_j$$

  Čvor sa "najboljim" odzivom za dati ulazni vektor je pobjednik i korekcija njegove težine računa se prema izrazu

$$\Delta w_n = \alpha(k)(x - w_n)$$

  Korekcija težina je proporcionalna razlici (***x***-$\boldsymbol{w}_n$), tako da "pobeđuje" onaj vektor $\boldsymbol{w}_n$ koji je najbliži ulaznom vektoru ***x***.



- *VNM* koja koristi *Winner Takes All* algoritam sastoji se od dva sloja međusobno povezanih sinapsama sa adaptivnim težinama:
  - *Ulazni sloj* (adaptivni filteri, memorijski nivo)
  - *Izlazni sloj* (filter maksimalne vrijednosti, kodirajući nivo)

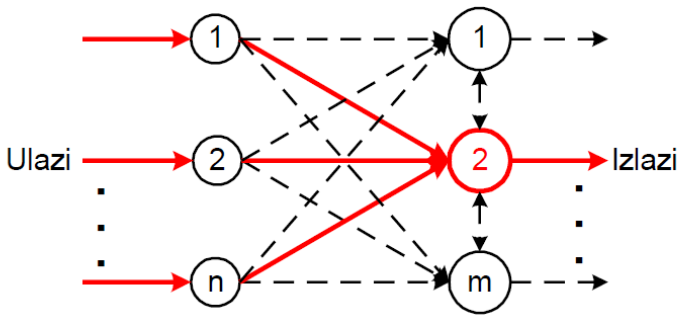
- Obuka:

Pravila i proces obuke

- Neuroni datog sloja su podeljeni u nepreklapajuće klastere;
- Svaki neuron u klasteru utiče na sve ostale. Neuroni u klasteru rade po “winner-takes-all” konceptu;
- Svaki element klastera prima ulazne signale na isti način. Za svaki ulaz se određuje najveći izlaz i “pobednički” neuron. Njegov izlaz je maksimalan (1), dok ostali neuroni u klasteru imaju minimalan izlaz (0);
- Obučava se samo “pobednički” neuron iz klastera;
- Ulazni uzorak x_j je binaran. Aktivni elemenat ima vrednost 1, a neaktivni 0;
- Svaki neuron ima **fiksiranu ukupnu vrednost težina** w_{ij} (**sve su pozitivne**) i one su raspodeljene po ulaznim sinapsama.
- Vrednost **sume svih težina** w_{ij} po neuronu je **jednaka 1**.
- Obuka se može vršiti sa i bez učitelja

Linear Vector Quantization

- Primjer klasifikatora.
- Obuka sa učiteljem.
- *VNM* ima dva sloja - ulazni i izlazni.
- Tokom obuke se vrši određivanje pripadnosti ulaznih podataka nekom od datih skupova (grupa, klastera).
- Koristi se *Winner Takes All* algoritam.
- Ovim algoritmom se vrši pomjeranje granica grupa ka njihovim optimalnim vrijednostima.
    - Primjena u okviru prepoznavanja štamparskih karaktera, konverzije govora u štampani tekst i sl.

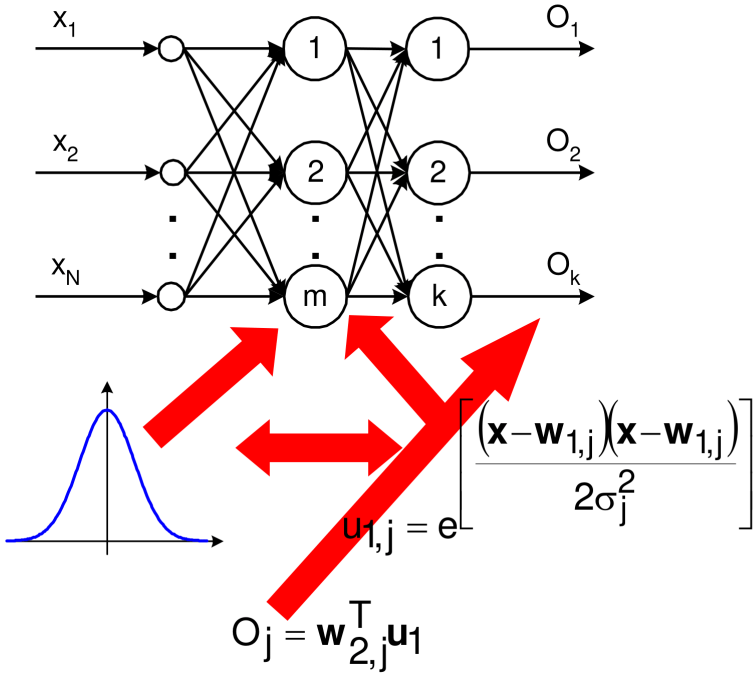


Self Organizing Feature Map

- Samoorganizujući algoritam za klastering.
- Obuka bez učitelja.
- Proizvodi niskodimenzionalnu (tipično $2D$) diskretnu reprezentaciju ulaznih podataka za obuku, koju zovemo mapa
- Za razliku od drugih *VNM*, *SOM* primjenjuju kompetitivnu obuku, a ne obuku minimizacije greške kao kod *Back Propagation* algoritma.
- Za razliku od *LVQ* algoritma (klasifikacioni problem), ovdje se set ulaznih uzoraka dijeli na unaprijed nepoznat skup klastera (predodređen skup vrijednosti, skup grupa).
- Vektor težina je iste dimenzije kao i vektor ulaznih podataka.
- Za određivanje neurona koji je najbliži datom uzorku koristi se metod određivanja udaljenosti između vektora (pobjeđuje neuron čije su težine najsličnije ulaznom vektoru).
- Cilj obuke je da različiti dijelovi mreže daju sličan odziv na određene oblike ulaznih podataka.
- Motivacija je proistekla iz toga kako se vizuelne, auditorne i druge senzorne informacije obrađuju u različitim dijelovima ljudskog mozga.
- Na kraju obuke, svaki izlazni neuron predstavlja jedan klaster.

Radial Basis Function

- Problemi klasifikacije i regresije (*pattern recognition*).
  - Da primjete koliko su ulazi udaljeni od željenih izlaza.
- Sastoje se od dva sloja.
- Na izlaznom sloju se formira linearna kombinacija funkcija izračunatih na skrivenom sloju.
- Bazna funkcija na skrivenom sloju daje značajan nenulti izlaz u slučaju kada pobudni signal uzima vrijednost iz određenog intervala.
- Najčešće pobudne funkcije na skrivenom sloju - *sigmoida* i *Gausijan*.
- Algoritam obuke:
  - Početi obuku skrivenog sloja metodom obuke bez nadzora.
  - Nastaviti sa obukom izlaznog sloja metodom obuke sa nadzorom.
  - Istovremeno primjeniti obučavanje sa nadzorom na skriveni i izlazni sloj radi finog podešavanja mreže (*backprop* algoritam za kraj).


x1
x2
xN
1
2
m
1
2
k
O1
O2
Ok
$u_{1,j} = e^{\left[\frac{(\mathbf{x}-\mathbf{w}_{1,j})(\mathbf{x}-\mathbf{w}_{1,j})}{2\sigma_j^2}\right]}$
$O_j = \mathbf{w}_{2,j}^T \mathbf{u}_1$