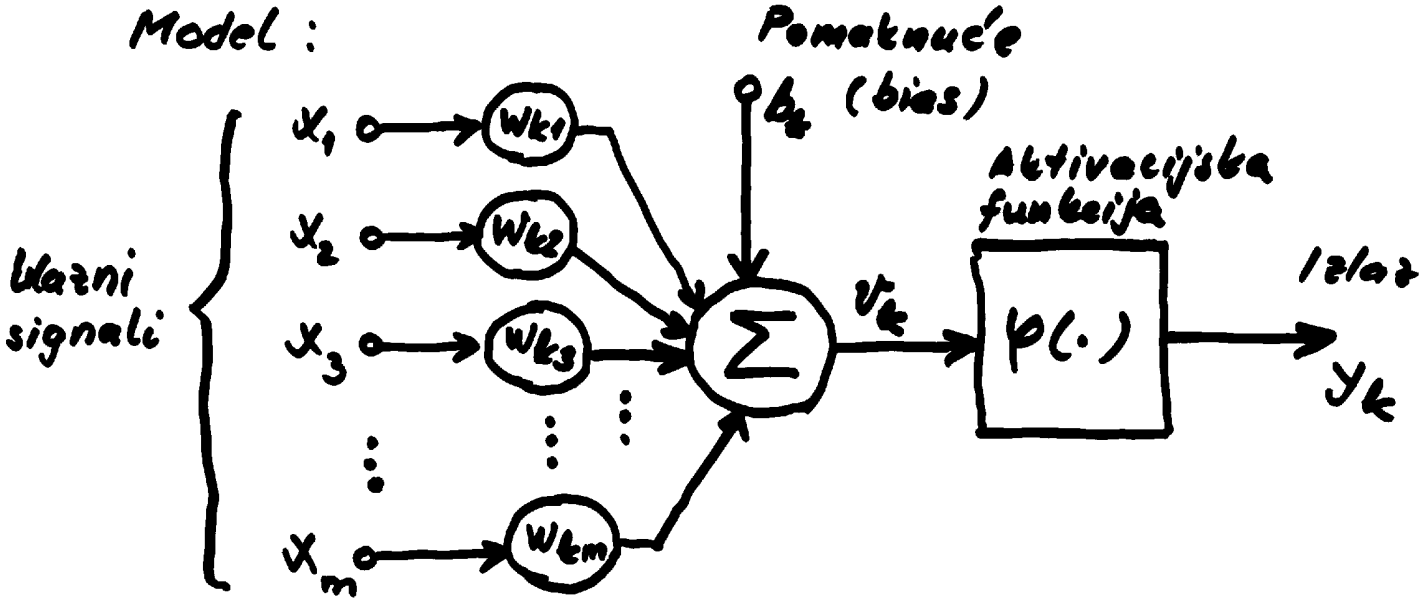
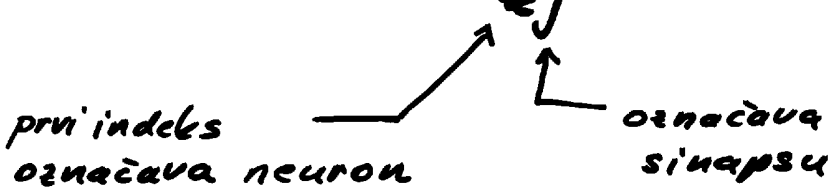
Modeli neurona

Neuron → informacijsko-procesna jedinica
temeljna građevna jedinica
neuronskih mreža

Model :



- tri osnovna elementa modela :
  - skup sinapsi ; svaka sinapsa karakterizirana je jačinom ili težinom ;
    signal x_j na ulazu sinapse j neurona k se množi sinaptičkom težinom w_{kj}.



  - zbrajalo (Σ) - zbraja ulazne signale koji su pomnoženi odgovarajućim sinaptičkim težinama
    (linear combiner)

- aktivacijska funkcija - za ograničavanje amplitude izlaza neurona
  /tipično, normalizirana amplituda [0,1] ili [-1,1]./

Pomaknuće (engl. bias b_k) ima zadatak povećanja ili umanjenja (tzv. net input) mrežnog ulaza u aktivacijsku funkciju

Neuron k može se opisati parom jednadžbi:

$$u_k = \sum_{j=1}^{m} w_{kj} \cdot x_j$$

i

$$y_k = \varphi(u_k + b_k), \text{ gdje su}$$

$x_1, x_2, \ldots, x_m$ - ulazni signali

$w_{k1}, w_{k2}, \ldots, w_{km}$ - sinaptičke težine

u_k - izlaz iz linearnog komb. sklopa

b_k - pomaknuće

$\varphi(\cdot)$ je aktivacijska funkcija

y_k - izlaz k-tog neurona

Uporabom pomaknuća b_k postiže se efekt primjene afine transformacije na u_k :

$$v_k = u_k + b_k$$

- pomaknuće b_k je vanjski parametar umjetnog neurona k; međutim, možemo formulirati model neurona i ovako:

$$v_k = \sum_{j=0}^{m} w_{kj} \cdot x_j$$

i

$$y_k = \varphi(v_k),$$

gdje je $x_0 = +1$ i $w_{k0} = b_k$

<u>Vrste aktivacijskih funkcija:</u>

1. Funkcija praga (Threshold Function)

$$\varphi(v) = \begin{cases} 1 & \text{if } v \geq 0 \\ 0 & \text{if } v < 0 \end{cases}$$

(Heaviside funkc.)

$$y_k = \begin{cases} 1 & \text{if } v_k \geq 0 \\ 0 & \text{if } v_k < 0, \end{cases}$$

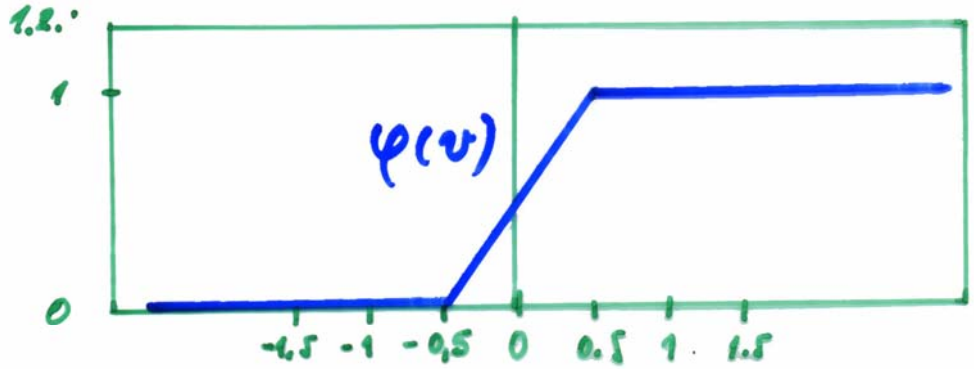
gdje je v_k inducirano lokalno polje:
(engl. <u>induced local field</u>)

$$v_k = \sum_{j=1}^{m} w_{kj} \cdot x_j + b_k$$

McCulloch-Pitts model neurona!
(1943. god.)

2. Linearna po odsječcima
(Piecewise - Linear Function)

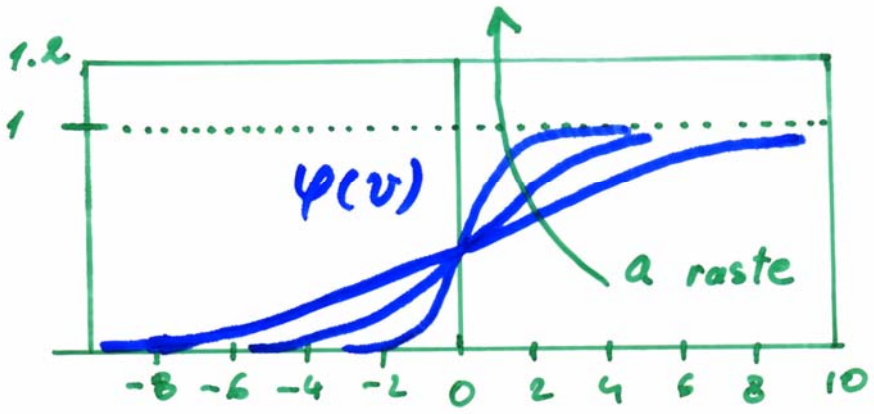
$$\varphi(v) = \begin{cases} 1 & v \geq +1/2 \\ v & +\frac{1}{2} > v > -\frac{1}{2} \\ 0 & v \leq -\frac{1}{2} \end{cases}$$



3. Sigmoidna funkcija (logistička funkcija)
- najčešće korištena aktivacijska funkcija u izgradnji umjetnih neuronskih mreža

$$\varphi(v) = \frac{1}{1 + \exp(-av)}$$

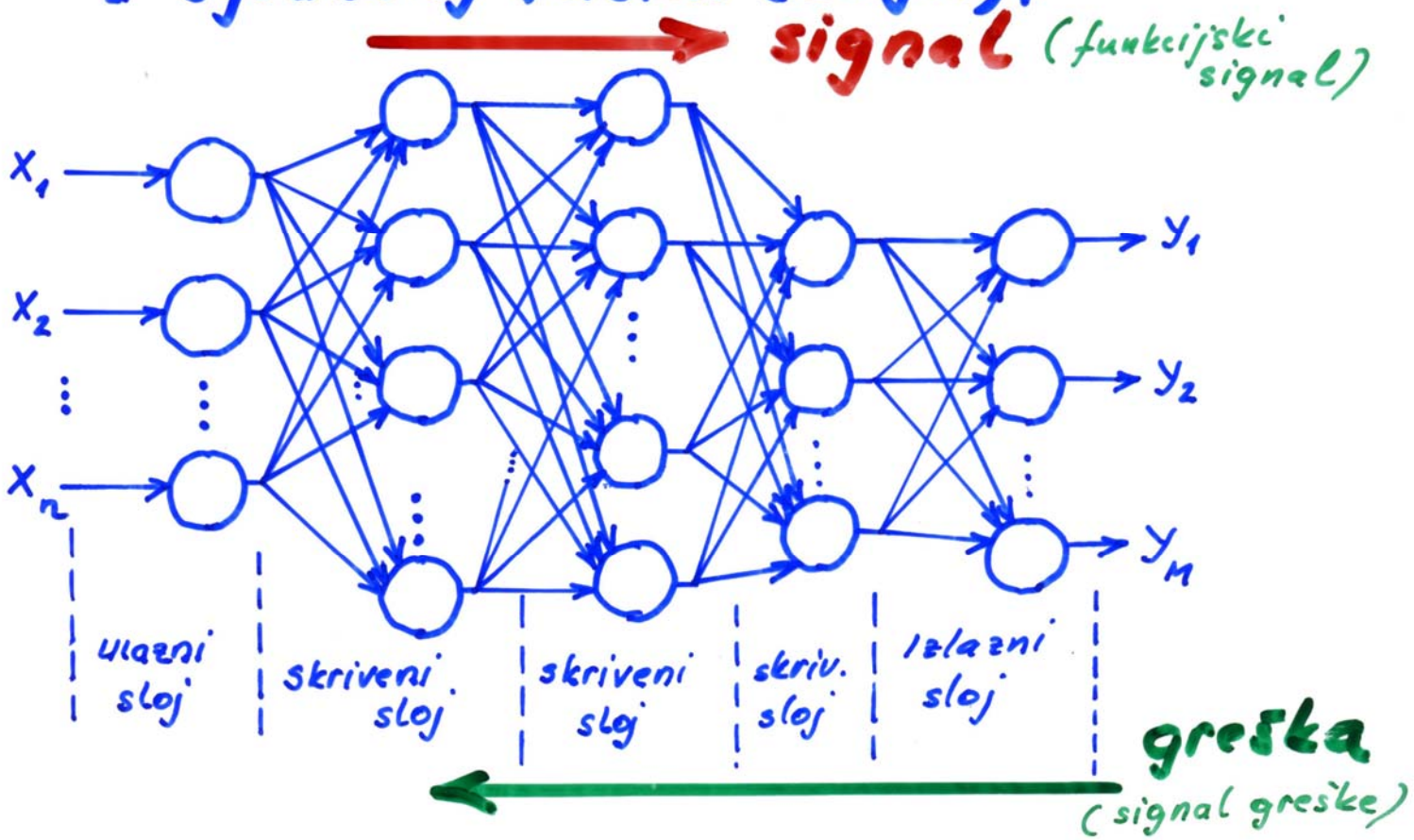
a - parametar nagiba sig. funkcije



Neuronske mreže s propagacijom signala naprijed
(Feedforward NN)

- FNN sastoji se od:
  - razine n ulaznih jedinica,
  - jednog ili više "skrivenih" slojeva
  - izlaznog sloja od M jedinica

Veze ili težine povezuju svaku jedinicu (procesni element) iz jednog sloja sa samo onim jedinicama u sljedećoj razini (sloju).



- ulazni sloj → jedinice samo "drže" ulazne vrijednosti i distribuiraju te vrijednosti svim jedinicama u sljedećem sloju;

- FNN ima mogućnost učenja preslikavanja uzorka (funkcija "pattern associator-a" pomoću učenja)

- Učenje se izvodi predstavljanjem uzoraka iz skupa za učenje i određivanjem (željenih) izlaza iz mreže.

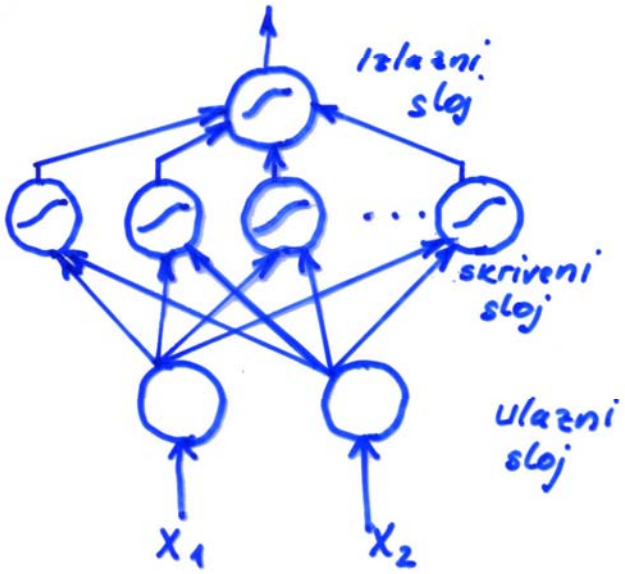
- Stvarni se izlaz mreže uspoređuje s željenim (ili ciljanim) izlazom i računa se pogreška

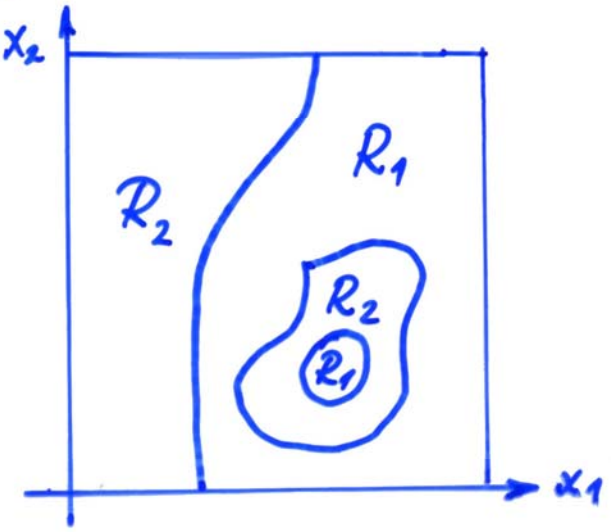
- Pogreška se tada propagira unatrag kroz mrežu i koristi se za određivanje promjena težinskih vrijednosti (<u>error backpropagation</u>)

Uloga skrivenog sloja?

- dvorazinske mreže → linearna odlučujuća funkcija

- tro-, četiri- i višeslojne NN mreže → proizvoljne odlučujuće funkcije





LMS (Least-Mean-Square) algoritam

$$E(\vec{w}) = \frac{1}{2} e^2(n)$$ ← trenutna vrijednost energije pogreške za neuron (n-označava prisutnost n-tog uzorka za učenje)

$\vec{w}$ - vektor težinskih faktora (vrijednosti)

$e(n)$ - signal greške za uzorak (n-ti) iz skupa za učenje

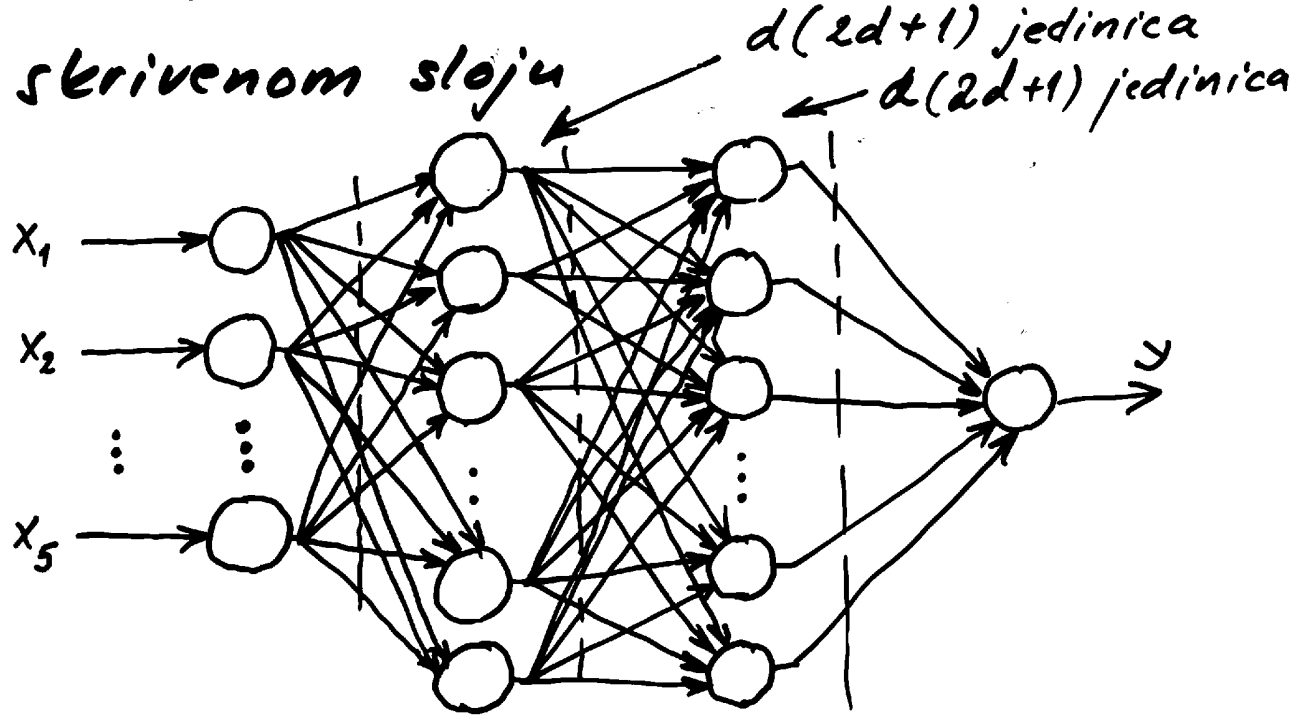
$$\frac{\partial E(\vec{w})}{\partial \vec{w}} = e(n) \frac{\partial e(n)}{\partial \vec{w}}$$

Broj neurona u skrivenom sloju ?

Kolmogorov teorem (1957)

Sprecher (1965), Kahane (1975)
Lorentz (1976)

Funkcija $y(\vec{x})$ koja preslikava
d ulaznih varijabli x_i u
jednu izlaznu varijablu y
može se predočiti 4-razinskim
perceptronom (4-slojnom NN)
koja ima $d(2d+1)$ jedinica
u prvom skrivenom sloju
i $d(2d+1)$ jedinica u drugom
skrivenom sloju



- Ako je neuron s linearnom aktivacijskom funkcijom onda se signal greške može izraziti:

$$e(n) = d(n) - y(n)$$

$$e(n) = d(n) - \vec{X}^T(n)\,\vec{w}(n)$$

$d(n)$ - željeni izlaz

$$\frac{\partial e(n)}{\partial \vec{w}(n)} = -\vec{x}(n)$$

$$\frac{\partial E(\vec{w})}{\partial \vec{w}} = -e(n)\cdot\vec{x}(n)$$

Izraz $-e(n)\cdot\vec{x}(n)$ predstavlja procjenu gradijenta $\vec{g}(n)$ koji se koristi u izrazu za promjenu težinskog vektora:

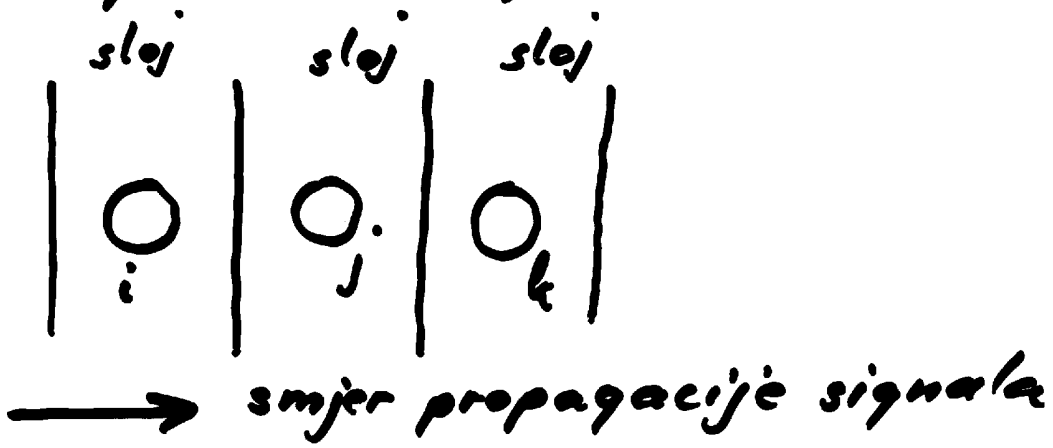
$$\vec{w}(n+1) = \vec{w}(n) - \eta\,\vec{g}(n)$$

$$\vec{w}(n+1) = \vec{w}(n) + \underbrace{\eta\, e(n)\,\vec{x}(n)}_{\Delta\vec{w}}$$

$$\boxed{\Delta\vec{w}(n+1) \propto \frac{\partial E}{\partial \vec{w}}}$$

Označavanje:

- i, j te k označavaju različite neurone u mreži; za propagaciju signala kroz mrežu s lijeva udesno, neuron j leži u sloju desno od neurona i, a neuron k leži u sloju desno od neurona j (kada je neuron j skrivena jedinica);

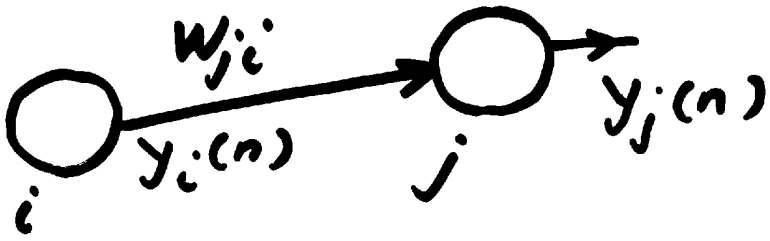


- U iteraciji (vremenskom koraku) n, n-ti se uzorak iz skupa za učenje predstavlja mreži;

- $E(n)$ označava trenutnu sumu kvadrata pogreške;

$$E(n) = \frac{1}{2} \sum_{j \in C} e_j^2(n)$$

- $e_j(n)$ signal pogreške (pogreška) za izlazni neuron j za iteraciju n
- $d_j(n)$ željeni odgovor za neuron j (koristi se za računanje $e_j(n)$);
- $y_j(n)$ signal koji se pojavljuje na izlazu neurona j za iteraciju n.
- $w_{ji}(n)$ sinaptička težina (veza) koja povezuje izlaz neurona i s ulazom neurona j pri iteraciji n



- $\Delta w_{ji}(n)$ korekcija sinaptičke težine pri iteraciji n
- $v_j(n)$ - lokalno inducirano polje neurona j pri iteraciji n

- $\varphi_j(\cdot)$ aktivacijska funkcija koja opisuje ulaz-izlaz odnos neurona j;

- w_{j0} - pomaknuće (engl. bias) b_j neurona j spojen s ulazom (čvoristim!) koji ima vrijednost +1. ← predočeno sinaptičkom težinom

- η korak (stopa) učenja;

- m_l označava veličinu - broj čvorova u sloju l;
  $l = 0, 1, 2, \ldots, L$
  L - "dubina" mreže
  m_0 - veličina ulaznog sloja;
  $m_L = M$ - veličina izlaznog sloja;

Postupak učenja propagacijom greške unatrag

- signal greške na izlazu neurona j u n-toj iteraciji:

$$e_j(n) = d_j(n) - y_j(n)$$

neuron j je čvor izlaznog sloja mreže!

Vrijednost greške neurona j je

$$\frac{1}{2} e_j^2(n)$$

$E(n)$ ukupna greška

$$E(n) = \frac{1}{2} \sum_{j \in C} e_j^2(n)$$

gdje je C skup neurona izlaznog sloja mreže.

N – broj uzoraka za učenje (veličina skupa za učenje)

Prosječna srednja kvadratna pogreška

$$E_{av} = \frac{1}{N} \sum_{n=1}^{N} E(n)$$

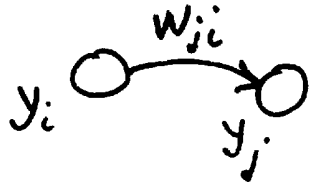
- $E(n)$ i E_{av} su funkcije svih slobodnih parametara mreže (težine, pomaknuća)

Zadatak učenja: prilagoditi slobodne parametre mreže da se postigne minimalna prosječna pogreška E_{av}.

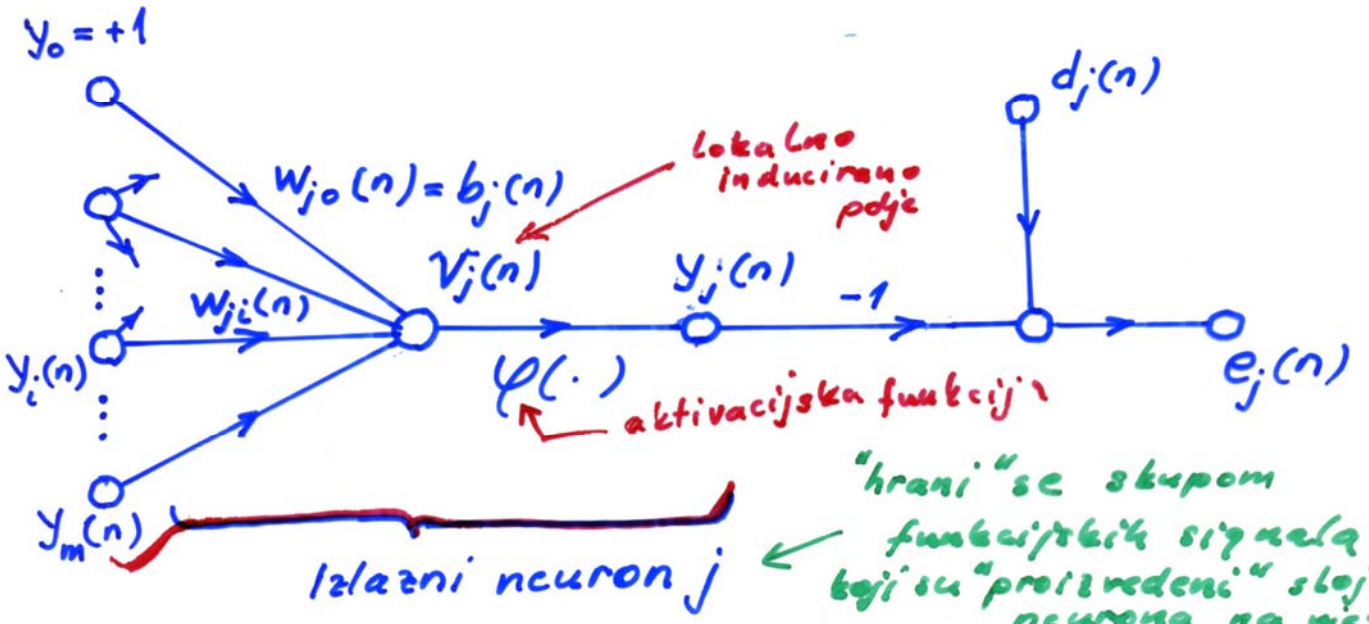
Metode učenja: prilagođavanje težina uzorkom (uzorak po uzorak) ili

prilagođavanje težina u jednoj epohi – dok mreži nisu predstavljeni svi uzorci iz skupa za učenje:

$$\Delta \bar{w}_{ji} = \sum_{p} \Delta^{p} w_{ji}$$



↑ ukupno akumulirana pogreška nakon jedne epohe



Graf toka signala (j je izlazni neuron)

Lokalno inducirano polje

$v_j(n)$ na ulazu u aktivacijsku funkciju pridruženu neuronu j računamo:

$$v_j(n) = \sum_{i=0}^{m} w_{ji}(n)\, y_i(n)$$

- $y_i(n)$ - funkcijski signal na <u>izlazu</u> neurona iz prethodnog sloja
- m - ukupni broj ulaza predstavljen neuronu j
- težina w_{j0} povezana je s $y_0 = +1$ — odgovara pomaknuću b_k

- Funkcijski signal $y_j(n)$ na izlazu neurona j u iteraciji n je:

$$y_j(n) = \varphi_j(v_j(n))$$

Imamo:

$$E(n) = \frac{1}{2}\sum_{j \in C} e_j^2(n)$$

$$e_j(n) = d_j(n) - y_j(n)$$

$$y_j(n) = \varphi_j(v_j(n))$$

$$v_j(n) = \sum_{i=0}^{m} w_{ji}(n)\, y_i(n)$$

Slično kao kod LMS algoritma, algoritam propagacije greške unatrag mijenja težinu $w_{ji}(n)$ za iznos korekcije $\Delta w_{ji}(n)$ koja je proporcionalna parcijalnoj derivaciji $\frac{\partial E(n)}{\partial w_{ji}(n)}$

Prema pravilu ulančavanja, gradijent je:

$$\frac{\partial E(n)}{\partial w_{ji}(n)} = \frac{\partial E(n)}{\partial e_j(n)} \cdot \frac{\partial e_j(n)}{\partial y_j(n)} \cdot \frac{\partial y_j(n)}{\partial v_j(n)} \cdot \frac{\partial v_j(n)}{\partial w_{ji}(n)}$$

↑ predstavlja faktor osjetljivosti koji određuje smjer pretraživanja u prostoru težina za $w_{ji}(n)$

$$\frac{\partial E(n)}{\partial w_{ji}(n)} = \underbrace{\frac{\partial E(n)}{\partial e_j(n)}}_{①} \underbrace{\frac{\partial e_j(n)}{\partial y_j(n)}}_{②} \underbrace{\frac{\partial y_j(n)}{\partial v_j(n)}}_{③} \cdot \underbrace{\frac{\partial v_j(n)}{\partial w_{ji}(n)}}_{④}$$

① $E(n) = \frac{1}{2}\sum e_j^2(n) \Big/ \frac{\partial}{\partial e_j(n)} \rightarrow \frac{\partial E(n)}{\partial e_j(n)} = e_j(n)$

② $e_j(n) = d_j(n) - y_j(n) \Big/ \frac{\partial}{\partial y_j(n)} \rightarrow \frac{\partial e_j(n)}{\partial y_j(n)} = -1$

③ $y_j(n) = \varphi_j(v_j(n)) \Big/ \frac{\partial}{\partial v_j(n)} \rightarrow \frac{\partial y_j(n)}{\partial v_j(n)} = \varphi'(v_j(n))$

④ $v_j(n) = \sum_{i=0}^{m} w_{ji}(n) y_i(n) \Big/ \frac{\partial}{\partial w_{ji}(n)}$

$$\hookrightarrow \frac{\partial v_j(n)}{\partial w_{ji}(n)} = y_i(n)$$

$$\boxed{\frac{\partial E(n)}{\partial w_{ji}(n)} = -e_j(n)\varphi_j'(v_j(n))y_i(n)}$$

Korekcija $\Delta w_{ji}(n)$ primijenjena na težinski faktor $w_{ji}(n)$:

$$\boxed{\Delta w_{ji}(n) = -\eta \frac{\partial E(n)}{\partial w_{ji}(n)}}$$

Delta pravilo!

- η parametar brzine učenja
- znak (−) upućuje na gradijentni spust u prostoru težina (traži se smjer promjene težine koja smanjuje vrijednost $E(n)$)

$$\Delta w_{ji}(n) = -\eta \frac{\partial E(n)}{\partial w_{ji}(n)}$$

$$\Delta w_{ji}(n) = \eta \underbrace{e_j(n)\varphi_j'(v_j(n))}_{\delta_j(n)\text{ - lokalni gradijent}} y_i(n)$$

- Lokalni gradijent je definiran sa:

$$\delta_j(n) = -\frac{\partial E(n)}{\partial v_j(n)} = -\frac{\partial E(n)}{\partial e_j(n)} \cdot \frac{\partial e_j(n)}{\partial y_j(n)} \cdot \frac{\partial y_j(n)}{\partial v_j(n)}$$

$$\boxed{\delta_j(n) = e_j(n)\varphi_j'(v_j(n))}$$

lokalni gradijent

Lokalni gradijent upućuje na potrebne promjene (sinaptičkih) težina neurona j.

$e_j(n)$ - signal greške

$\varphi_j'(v_j(n))$ - derivacija aktivacijske funkcije

Ključni element u računanju promjena težina $\Delta w_{ji}(n)$ je signal greške $e_j(n)$ na izlazu neurona j.

$$\Delta w_{ji}(n) = \eta \delta_j(n) \cdot y_i(n)$$

Dva slučaja:

- Neuron j je izlazni čvor.

(1) Ovaj slučaj je jednostavan jer se za svaki izlazni čvor definira željeni odziv te je lako izračunati odgovarajući signal greške.

- Neuron j je <u>čvor u skrivenom sloju</u>.

(2) Iako skrivenim čvorovima nije moguće pristupiti izravno, <u>oni dijele odgovornost u stvaranju greške na izlazu mreže</u>.

Pitanje: Kako znati i kako "kazniti" ili "nagraditi" skrivene neurone za njihov udio odgovornosti?

Problem: "credit-assignment problem" - rješenje

širenje signala greške unatrag kroz mrežu !!!

Slučaj ①

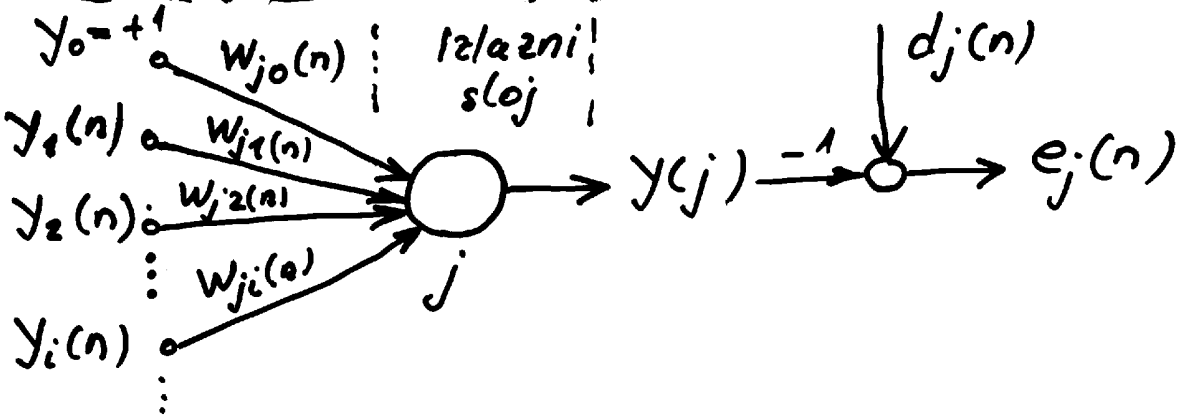
- neuron j je izlazni čvor

$$d_j(n) \; ; \quad y_j(n)$$

$$e_j(n) = d_j(n) - y_j(n)$$

$$\delta_j(n) = e_j(n) \cdot \varphi_j'(v_j(n))$$

$$\boxed{\Delta w_{ji}(n) = \eta \delta_j(n) y_i(n)}$$

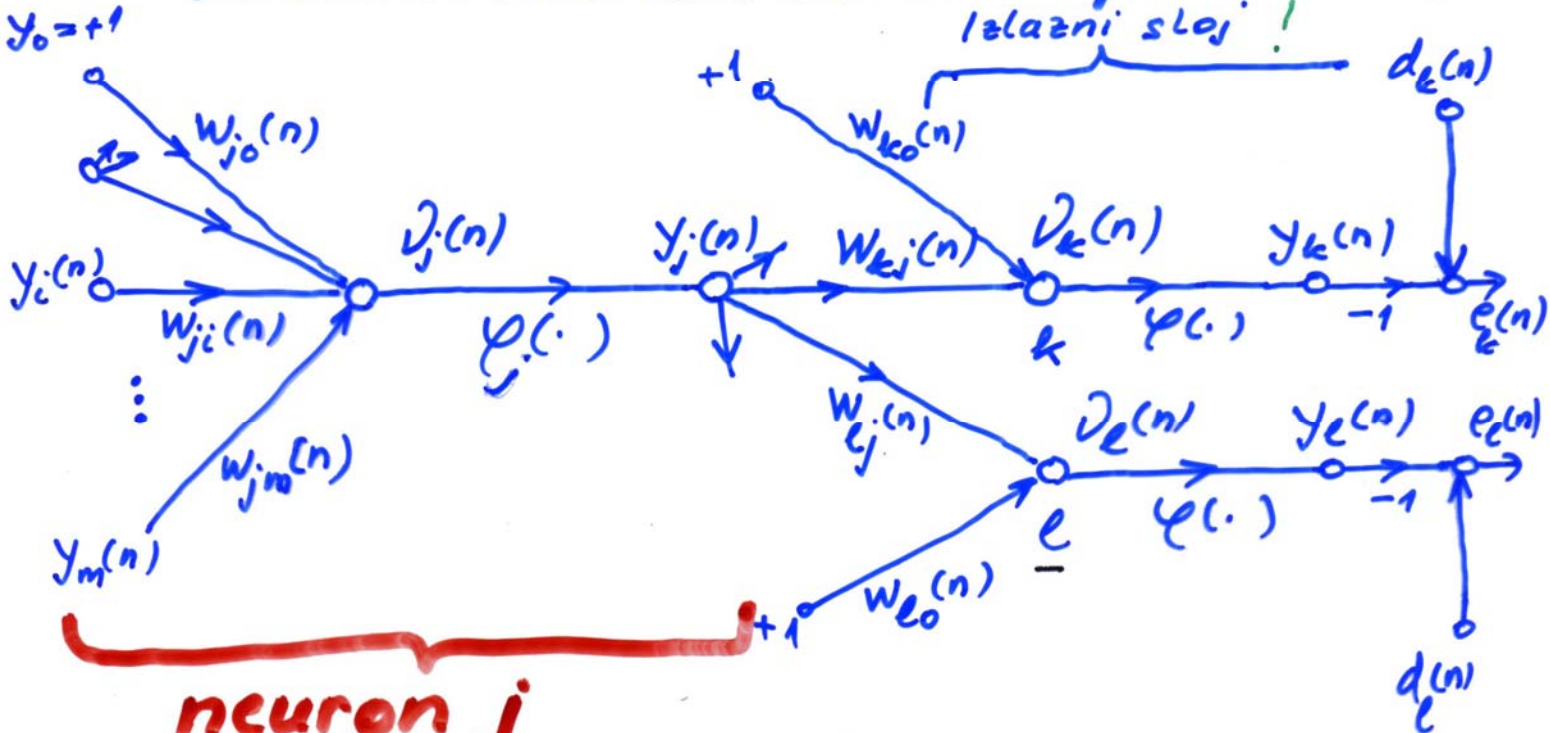


Slučaj ②: Neuron j je skriveni čvor

- Kada je neuron j skriveni čvor → ne postoji definirani željeni odziv tog neurona

Signal greške $e_j(n)$?

Signal greške skrivenog neurona mora se odrediti rekurzivno pomoću signala greški svih neurona s kojima je dotični skriveni neuron izravno povezan.



k-ti neuron u izlaznom sloju
l-ti neuron u izlaz.

- Za skrivenu jedinicu j želimo naći metodu za određivanje Δw_{ji};

Podsjetimo se:

$$\Delta w_{ji}(n) = -\eta \frac{\partial E(n)}{\partial w_{ji}(n)}$$

- sljedno tome želimo procijeniti $\frac{\partial E(n)}{\partial w_{ji}(n)}$ za <u>skrivenu jedinicu</u> j

Podsjetimo se:

E se računa na temelju uspoređivanja <u>izlaza izlaznih neurona</u> te <u>željenih izlaznih vrijednosti</u>.

- Npr. za 3-slojnu mrežu (ulazni sloj, skriveni sloj, izlazni sloj) možemo <u>promatrati skrivenu jedinicu j</u> i kako težine za njen izlaz $y_j(n)$ utječu na E:

1. Izlaz $y_j(n)$ aktivira neurone u izlaznom sloju (slika)
2. Izlaz $y_j(n)$ je funkcija ulaza jedinice j, težina i aktivacijske funkcije

Vrijedi:

$$\frac{\partial E(n)}{\partial w_{ji}} = \frac{\partial E(n)}{\partial y_j(n)} \cdot \frac{\partial y_j(n)}{\partial v_j(n)} \cdot \frac{\partial v_j(n)}{\partial w_{ji}(n)}$$

govori nam kako izlaz skrivene jedinice j utječe na grešku u svakoj od izlaznih jedinica

izlaz $y_j(n)$ je funkcija ulaza jedinice j, težina i aktivacijske funkcije

$$\frac{\partial E(n)}{\partial y_j(n)} = \frac{\partial}{\partial y_j(n)} \left[\frac{1}{2} \sum_{k=1}^{c} (d_k(n) - y_k(n))^2 \right]$$

$$= - \sum_{k=1}^{c} (d_k(n) - y_k(n)) \frac{\partial y_k(n)}{\partial y_j(n)}$$

$$= -\sum_{k=1}^{c} (d_k(n) - y_k(n)) \frac{\partial y_k(n)}{\partial v_k(n)} \cdot \frac{\partial v_k(n)}{\partial y_j(n)}$$

$$\frac{\partial E(n)}{\partial y_j(n)} = -\sum_{k=1}^{c} \underbrace{(d_k(n) - y_k(n))\, \varphi_k'(v_k)}_{\delta_k(n)} \cdot w_{kj}(n)$$

(vidi slučaj ①)

(str. NN 12)

$$\frac{\partial E(n)}{\partial w_{ji}(n)} = \left(-\sum_{k=1}^{c} \delta_k(n) \cdot w_{kj}(n) \right) \varphi_j'(v_j) \cdot y_i(n)$$

$$\frac{\partial E(n)}{\partial w_{ji}(n)} = \frac{\partial E(n)}{\partial y_j(n)} \cdot \frac{\partial y_j(n)}{\partial v_j(n)} \cdot \frac{\partial v_j(n)}{\partial w_{ji}(n)}$$

usporedimo

$$\Delta w_{ji} = \eta \delta_j \cdot y_i$$

$$\boxed{\delta_j(n) = \varphi_j'(v_j(n)) \sum_{k=1}^{c} w_{kj} \delta_k}$$

Lokalni gradijent skrivene jedinice j (osjetljivost) je suma pojedinih lokalnih gradijenta (osjetljivosti) izlaznih jedinica koje su "utežene" s w_{kj} i pomnožene s $\varphi_j'(v_j(n))$.

Pravilo učenja za težine (ulaz-skrivena težina) w_{ji}

$$\Delta w_{ji} = \eta \delta_j y_i$$

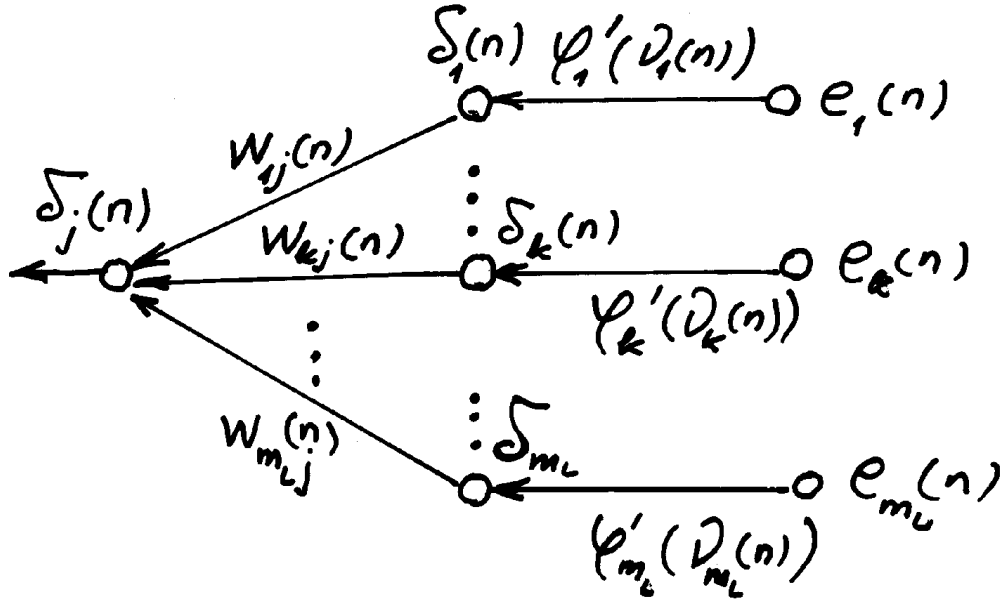
$$\Delta w_{ji} = \eta \underbrace{\left[\sum_{k=1}^{c} w_{kj} \cdot \delta_k\right] \varphi_j'(\vartheta_j(n))}_{\delta_j(n)} y_i(n)$$

Obnavljanje težina za veze (skriveni sloj - izlazni sloj):

$$\Delta w_{kj}(n) = \eta \delta_k y_j(n) = \eta e_k(n) \varphi'(\vartheta_k(n)) y_j$$

Učenje propagacijom greške unatrag (engl. "backpropagation of errors")

- za vrijeme učenja greška se mora propagirati od izlaznog sloja natrag prema skrivenom sloju.



$$\begin{bmatrix}\text{Korekcija težina} \\ \Delta w_{ji}(n)\end{bmatrix} = \begin{bmatrix}\text{Parametar brzine učenja} \\ \eta\end{bmatrix} \cdot \begin{bmatrix}\text{Lokalni gradijent} \\ \delta_j(n)\end{bmatrix} \cdot \begin{bmatrix}\text{Ulazni signal neuronaj} \\ y_i(n)\end{bmatrix}$$

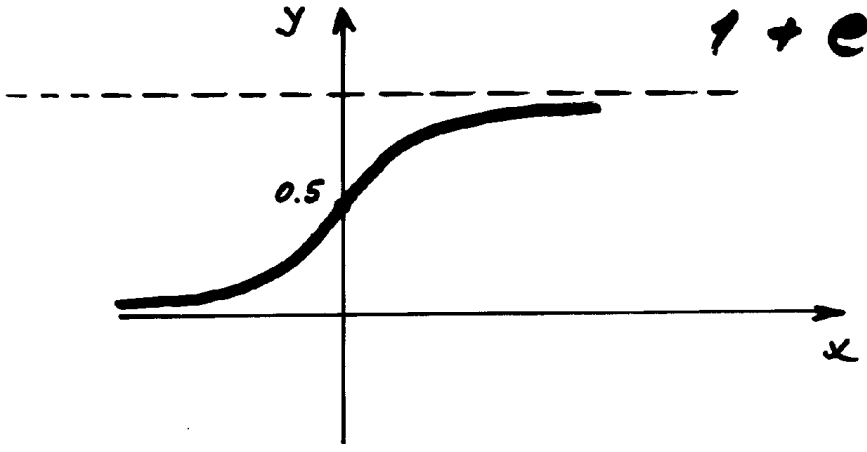
Lokalni gradijent $\delta_j(n)$ ovisi o tome da li je neuronj izlazni ili skriveni čvor:

(Generalizirano Delta pravilo)

1. Ako je neuron j izlazni čvor, $\delta_j(n)$ je jednak umnošku derivacije $\varphi_j'(v_j(n))$ i signala greške $e_j(n)$
2. Ako je neuron j skriveni čvor, $\delta_j(n)$ jednak je umnošku $\varphi_j'(v_j(n))$ i težinske sume faktora δ neurona u sljedećem skrivenom ili izlaznom sloju

Aktivacijska funkcija

- za računanje lokalnog gradijenta za svaki neuron višeslojnog perceptrona (MLP) zahtijeva se deriviranje aktivacijske funkcije $\varphi_j(\cdot)$ koja je pridružena tom neuronu (j).
- funkcija $\varphi_j(\cdot)$ kontinuirana i derivabilna
- Obično se uzima sigmoidna funkcija oblika $\frac{1}{1+e^{-x}}$



$$\varphi_j(v_j(n)) = \frac{1}{1+e^{-a\,v_j(n)}}$$

$$\varphi_j(v_j(n)) = \frac{1}{1 + e^{-a\, v_j(n)}}$$

$$\frac{d}{dx}\left(\frac{u}{v}\right) = \frac{u'v - u \cdot v'}{v^2} \qquad \frac{d}{dx}(e^{-x}) = -e^{-x}$$

$$\varphi_j'(v_j(n)) = \frac{a e^{-a\, v_j(n)}}{(1 + e^{-a\, v_j(n)})^2}$$

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

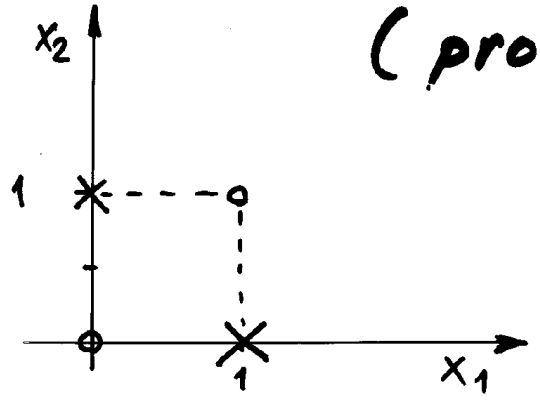
$$= \frac{-a + a\frac{1}{y}}{\frac{1}{y^2}}$$

$$= \frac{\frac{-ay + a}{y}}{\frac{1}{y^2}}$$

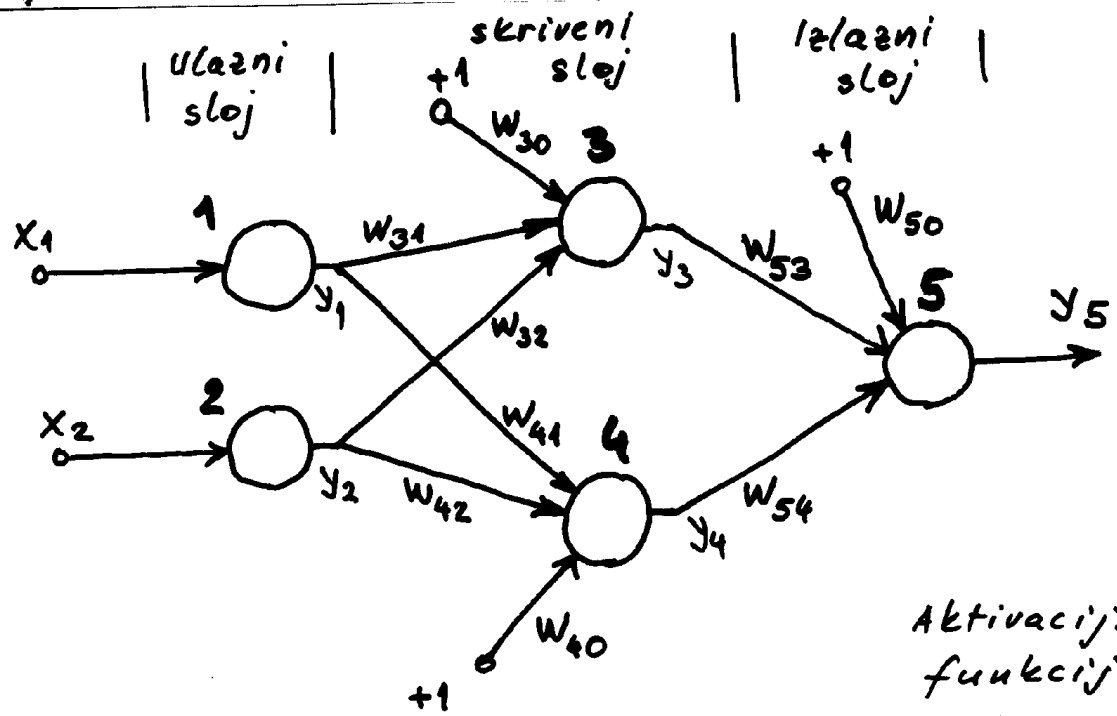
$$\boxed{\varphi_j'(v_j(n)) = a\, y_j(n)\,[1 - y_j(n)]}$$

$$y_j = \varphi_j(v_j(n))$$

Primjer: Troslojni perceptron (problem EXOR)



		željeni izlaz
$\vec{x}_1 = (0,0)^T$	$y = x_1 \oplus x_2$	0
$\vec{x}_2 = (1,1)^T$		0
$\vec{x}_3 = (0,1)^T$		1
$\vec{x}_4 = (1,0)^T$		1



Aktivacijska funkcija:

$$\varphi_j(v_j(n)) = \frac{1}{1+e^{-v_j(n)}}$$

$(a=1)$

Početne vrijednosti:

skriveni neuroni $j=3$ i $j=4$:

$w_{30} = 0.862518$ $w_{31} = -0.155797$

$w_{32} = 0.282885$

$w_{40} = 0.834986$ $w_{41} = -0.505997$ $w_{42} = -0.86449$

izl. neuron:

$w_{50} = 0.036498$ $w_{53} = -0.430437$ $w_{54} = 0.481210$

- uzimamo $\bar{x}_1 = (0,0)^T$

skriveni sloj

- neuron $j=3$

ulazi: +1 (pomaknuće)

$x_1 = 0$

$x_2 = 0$

računamo

$$v_3(1) = (+1)\cdot w_{30} + 0\cdot w_{31} + 0\cdot w_{32}$$

$$v_3(1) = (+1)\cdot 0.862518 = 0.862518$$

izlaz

$$y_3(1) = \frac{1}{1+e^{-0.862518}} = 0.70318$$

- neron $j=4$

ulazi: +1 (pomaknuće)

$x_1 = 0$

$x_2 = 0$

$$v_4(1) = (+1)\cdot w_{40} + 0\cdot w_{41} + 0\cdot w_{42}$$

$$v_4(1) = 0.834986$$

izlaz

$$y_4(1) = 0.697408$$

izlazni sloj

neuron $j = 5$

ulazi: $+1$ (pomaknuće)

(y_3) 0.570318

(y_4) 0.697408

$$v_5(1) = (+1) \cdot w_{50} + 0.570318 \cdot w_{53} + 0.697408 \cdot w_{54}$$

$$v_5(1) = 0.036498 + 0.570318 \cdot (-0.430437) + 0.697408 \cdot 0.481210$$

$$v_5(1) = 0.0694203$$

izlaz :

$$y_5(1) = \frac{1}{1 + e^{-0.0694203}}$$

$$y_5(1) = 0.517348$$

Sa zadanim početnim vrijednostima težina i ulazom $\vec{x}_1 = (0,0)^T$ dobivena vrijednost na izlazu troslojnog perceptrona je 0.517348

Željena vrijednost

$$d_5(1) = 0.0$$

Greška u izlaznom sloju
(neuron $j=5$)

$$e_5(1) = d_5(1) - y_5(1)$$

$$e_5(1) = 0.0 - 0.5173481$$

$$\underline{e_5(1) = -0.5173481}$$

Lokalni gradijent izlaznog neurona je:

$$\delta_j(n) = e_j(n) \cdot \varphi_j'(v_j(n))$$

$$\delta_5(1) = (-0.5173481) \cdot y_5[1 - y_5]$$

$$\delta_5(1) = -0.5173481 \cdot 0.5173481 \cdot 0.48265$$

$$\underline{\delta_5(1) = -0.1291813}$$

Lokalni gradijenti skrivenih neurona:

$$\delta_j(n) = \varphi_j'(v_j(n)) \sum_{k=1}^{c} w_{kj}\delta_k$$

$$\delta_3(1) = \varphi_3'(v_3(1)) \cdot w_{53}\delta_5(1)$$

$$\delta_3(1) = y_3(1)[1 - y_3(1)] w_{53}(1) \cdot \delta_5(1)$$

$$\delta_3(1) = 0.70318\,[1-0.70318]\cdot (-0.430437)\cdot(-0.1291813)$$

$$\delta_3(1) = 0.0116054$$

$$\delta_4(1) = \varphi_4'(v_4(1))\cdot w_{54}(1)\cdot \delta_5(1)$$

$$\delta_4(1) = 0.697408\,[1-0.697408]\cdot(0.481210)\cdot \cdot(-0.1291813)$$

$$\delta_4(1) = -0.013111$$

Za:

$$\eta = 0.5$$

slijedi korekcija težina:

- za izlazni neuron : $\Delta w_{ji}(n) = \eta\,\delta_j(n)\cdot y_i(n)$

$$\Delta w_{53}(1) = \eta\,\delta_5(1)\,y_3(1)$$

$$\Delta w_{53}(1) = 0.5\cdot(-0.1291813)\cdot 0.70318$$

$$\Delta w_{53}(1) = -0.0454192$$

$$\Delta w_{54}(1) = 0.5\cdot(-0.1291813)\cdot 0.697408$$

$$\Delta w_{54}(1) = -0.045046$$

$$\Delta w_{50}(1) = 0.5\cdot(-0.1291813)\cdot 1 = -0.0645906$$

Korekcije težina za neuron $j=3$

$$\Delta W_{30}(1) = \eta \delta_3(1) \cdot (+1)$$

$$\Delta W_{30}(1) = 0.5 \cdot 0.0116054 \cdot (+1)$$

$$\Delta W_{30}(1) = 0.0058027$$

$$\Delta W_{31}(1) = \eta \delta_3(1) \cdot y_1 \quad ; \; y_1 = 0$$

$$\Delta W_{31}(1) = 0$$

$$\Delta W_{32}(1) = \eta \delta_3(1) y_2 \quad ; \; y_2 = 0$$

$$\Delta W_{32}(1) = 0$$

- neuron $j = 4$

$$\Delta W_{40}(1) = \eta \delta_4(1) \cdot (+1)$$

$$\Delta W_{40}(1) = 0.5 \cdot (-0.01311)(+1)$$

$$\Delta W_{40}(1) = -0.0065592$$

$$\Delta W_{41}(1) = 0$$

$$\Delta W_{42}(1) = 0$$

Izračunajmo nove težinske vrijednosti:

$W_{50} = 0.036498 - 0.064706 = -0.028093$

$W_{53} = -0.430437 - 0.045492 = -0.475856$

$W_{54} = 0.481210 - 0.045046 = 0.436164$

$W_{40} = 0.834986 - 0.006588 = 0.828426$

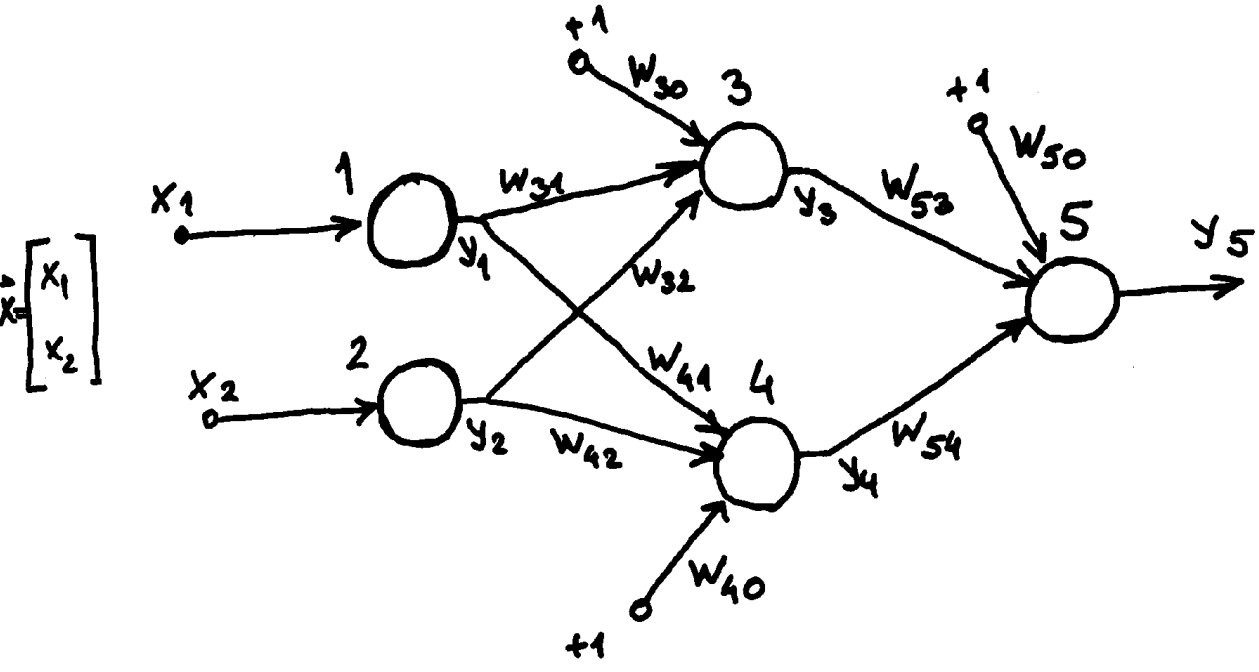
$W_{41} = -0.505997 + 0 = -0.505997$

$W_{42} = -0.86449 + 0 = -0.86449$

$W_{30} = 0.862518 + 0.000689 = 0.863207$

$W_{31} = -0.155797 + 0 = -0.155797$

$W_{32} = 0.282885 + 0 = 0.282885$



Mreži sada predstavimo novi uzorak npr. $\vec{x}(2) = (0,1)^T$ i cijeli postupak računanja promjena težina se ponavlja.

- Učenje se nastavlja sve dok greška između željenog i dobivenog izlaza mreže ne bude manje od unaprijed određene vrijednosti. Tada se postupak zaustavlja.

- Nakon nekoliko tisuća iteracija dobivaju se željeni izlazi (uz unaprijed dopuštenu grešku):

x_1	x_2	y
0	0	0.017622
0	1	0.981504
1	0	0.981491
1	1	0.022782