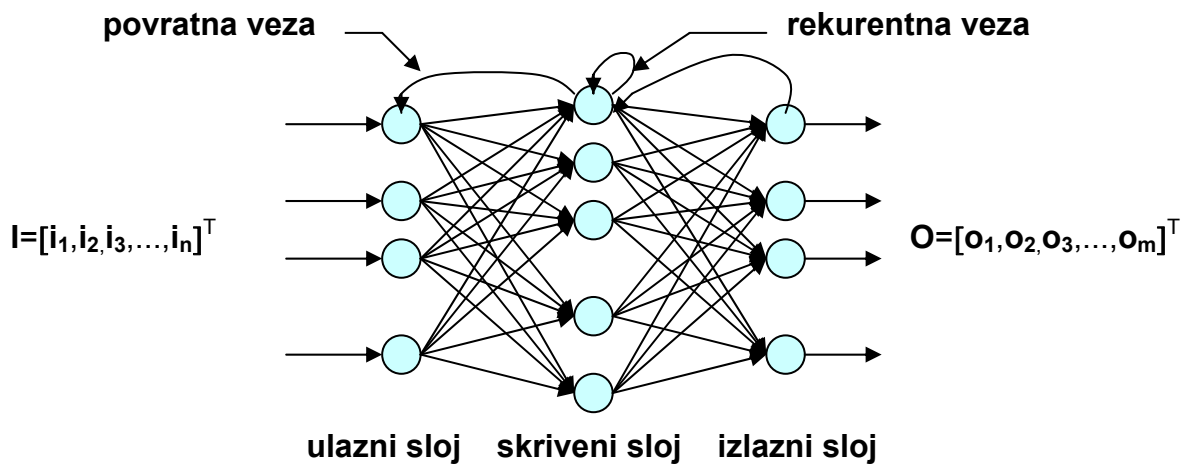


MAŠINSKO UČENJE I VEŠTAČKE NEURONSKE MREŽE

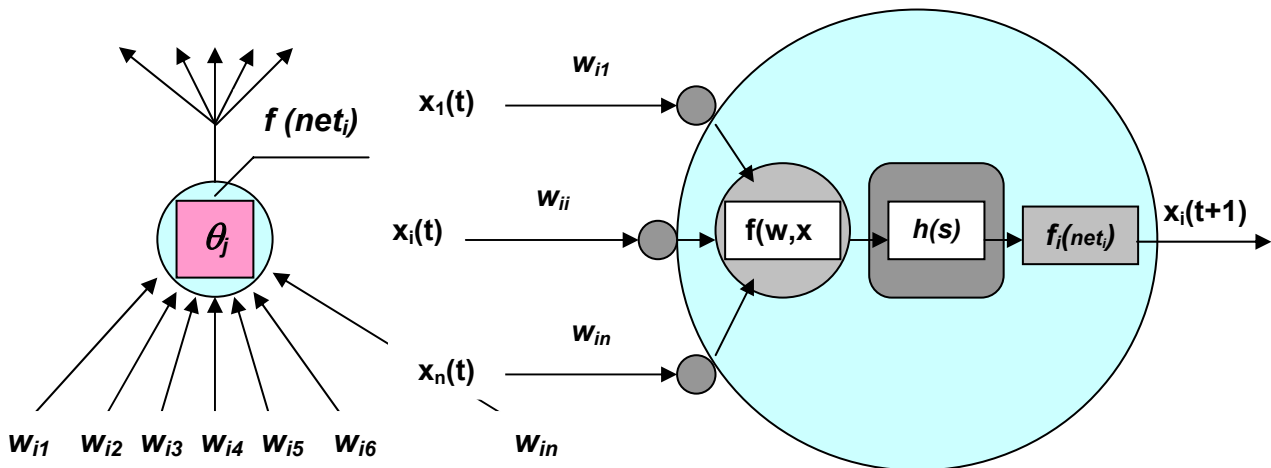
Veštačke neuronske mreže-osnovni koncepti

DARPA (Defense Advanced Research Projects Agency):

Neuronske mreže obezbeđuju značajne prednosti pri rešavanju problema procesiranja koji zahtevaju rad u realnom vremenu i interpretaciju međusobnih odnosa između promenljivih u višedimenzionalnim prostorima.



Opšta struktura veštačke neuronske mreže



Prikaz tipičnog procesirajućeg elementa - neurona

- **ulazni operator** $f(w,x)$, koji objedinjuje ulaze i težinske odnose međusobnih veza w i formira jedinstvenu vrednost s (odnosno $s=f(w,x)=\mathbf{w}^T \cdot \mathbf{x}$), koja je na taj način pripremljena za funkciju prenosa;
- **funkcija prenosa** $h(s)$, koja obrađuje izlaz iz neuronskog ulaznog operatora (vrši integraciju), formirajući potrebnu vrednost za aktivacionu funkciju;

- **aktivaciona funkcija** $f_i(net_i)$, koja obrađujući izlaz funkcije prenosa upravlja izlaznom vrednošću neurona.

Ulazni operatori, koji se najčešće koriste kod neuronskih modela, su dati u tabeli.

Matematička formulacija ulaznog operatora neurona

Tip ulaznog operatora	Matematički izraz	Napomena
Linearan	$net_j = \sum w_{ij} x_j$	<i>Linearna funkcija</i>
Težinska suma	$net_j = \sum w_{ij} x_j + \theta_j$	<i>Najčešće korišćen; u nekim slučajevima prag $\theta_j = 0$</i>
Povratna veza	$net_j = \alpha net_{staro} + \beta \sum w_{ij} x_j$	<i>α, β su težinske konstante</i>

Posle determinisanja stimulacije ulaza, ostvaruje se konvertovanje ulazne vrednosti u aktivacionu vrednost ili jednostavno aktivaciju neurona, što se može prikazati preko opšte jednačine date u:

$$a_i = F_i(a_i(t-1), net_i(t))$$

Ova jednakost ukazuje na to da je aktivacija neurona uvek eksplicitna funkcija ulaza mreže ka neuronu u vremenu t , odnosno simbolički napisano ($net_i(t)$), pri čemu treba voditi računa o činjenici da ona može takođe da zavisi i od aktivacije neurona u prethodnom vremenskom trenutku, što je dato kroz izraz $a_i(t-1)$. U mnogim slučajevima aktivacija neurona je ekvivalentna tekućem ulazu, odnosno:

$$a_i(t) = net_i(t)$$

Kada je aktivacija neurona u datom trenutku poznata, neuron daje **izlazni signal** koji je u relaciji sa njegovom aktivacijom preko **aktivacione funkcije**, što se matematički može izraziti na sledeći način:

$$o_i = x_i(t) = f_i(a_i(t)) = f_i(net_i(t))$$

Aktivacione funkcije igraju veoma važnu ulogu u različitim modelima veštačkih neuronskih mreža, tako da je potrebno opisati detalje koji su vezani za najpopularnije među njima. Aktivacione funkcije se mogu svrstati u nekoliko zajedničkih grupa kao što su: *linearne, binarne, sigmoidne, kompetitivne* i *Gausove*.

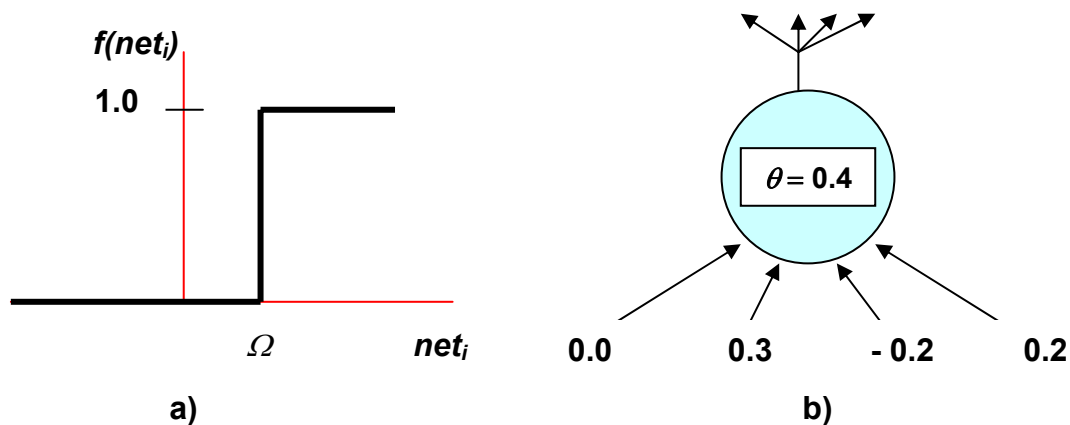
Linearna aktivaciona funkcija je najjednostavnija i praktično sumira sve ulaze u neuron propuštajući ih na izlazu iz neurona integrisane u koherentnom obliku, što se matematički može izraziti na sledeći način:

$$f_i^l(net_i(t)) = net_i(t)$$

Binarna aktivaciona funkcija, kao što sam naziv kaže, obezbeđuje neuronu dva stabilna stanja - aktivno i neaktivno, dato kroz sledeću jednačinu:

$$f_i^b(net_i(t)) = \begin{cases} 1 & \text{ako je } net_i(t) > \Omega \\ 0 & \text{ostalo} \end{cases}$$

gde Ω predstavlja tačku tranzicije između dva stanja, što se vidi na slici pod a. Kod mnogih paradigmi veštačkih neuronskih mreža Ω je jednako nuli, tako da je tada neuron u funkciji detektora polariteta. Ovu funkciju možemo smatrati odskočnom. Ako je ulazna aktivacija u neuron pozitivna i veća od nule, neuron "okida" i generiše aktivni izlaz, dok je u svim drugim slučajevima izlaz inhibitoran, što znači da stimulacija na ulazu neurona nije bila dovoljna da pređe prag θ . Za primer koji je prikazan na slici pod b, uočava se da će neuron imati aktivan izlaz samo kada su ulazni signali dva i četiri simultano aktivni, a signal tri neaktivan.



Binarna aktivaciona funkcija. a)Grafik funkcije. b)Moguća stanja ulaza.

Sigmoidna ili S-aktivaciona funkcija funkcioniše slično binarnoj, tako što generiše izlazni signal koji ima takođe dva stabilna stanja. Razlika između ove dve aktivacione funkcije je u tome što je sigmoidna matematički kontinualna i diferencijabilna. Takođe se može reći da je ona nelinearna i neopadajuća. Ima mnogo funkcija koje mogu da se koriste za modeliranje, ali najčešće su u upotrebi sledeće dve, date sledećim jednačinama:

$$f_i^s(net_i(t)) = \frac{1}{1 + e^{-(net_i(t) - \theta) / \tau}}$$

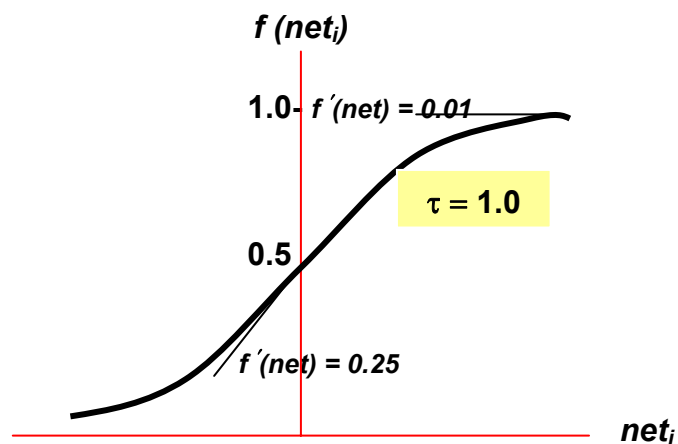
Jednačina predstavlja najopštiju formulaciju sigmoidne funkcije, jer se može odrediti i tačka tranzicije i oblik krive, variranjem vrednosti θ i τ , a u sledećoj jednačini na oblik krive utiče nezavisni parametar λ i koristi se kod specijalnih neuronskih mreža:

$$f_i^{s*}(net_i(t)) = \frac{1}{2} (1 + \tanh(\lambda net_i(t)))$$

Sličnost između sigmoidne i binarne aktivacione funkcije se uočava počevši od grafika (slika dole), i to:

- Obe aktivacione funkcije imaju dva domena, tako da značajna varijacija ulaza ne utiče drastično na promenu izlaza koji neuron ostvaruje.

- Širina regiona tranzicije kod obe funkcije je zanemarljiva, ako se poredi širina stabilnih regiona.
- Kod obe funkcije region tranzicije teži da egzistira oko nulte aktivacije, iako on može biti pomeren levo ili desno, modifikacijom veličine θ , date u jednačini.



Sigmoidna aktivaciona funkcija

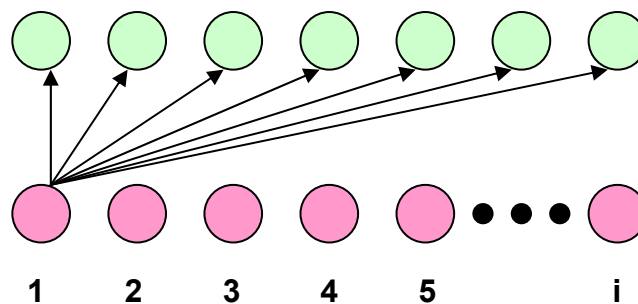
Aktivni izlaz iz neurona koji koristi sigmoidnu funkciju, zapravo predstavlja indikaciju da je prisutna specifična kombinacija karakteristika ulaznog vektora, dok neaktivni izlaz iz neurona implicira odsustvo tih karakteristika. Međutim, ne bi trebalo zaboraviti da kontinualnost sigmoidne funkcije ponekad uslovljava situaciju koja može da stvori konfuziju o tome šta neuron stvarno klasifikuje. Da bi se to ilustrovalo, neophodno je konstatovati da u primeru datom na slici gore pod b, pored dva moguća izlazna stanja neurona: aktivno i neaktivno, korišćenjem sigmoidne aktivacione funkcije ponekad se na izlazu pojavljuje treće stanje, a to je nedefinisano stanje između aktivnog i neaktivnog.

Takođe treba naglasiti i to da sigmoidna funkcija uslovljava šta će neuron na izlazu generisati. Moguće vrednosti se nalaze između minimalne (0.0) i maksimalne granice (1.0), tako da se, zbog prethodno nagoveštene nedefinisane situacije, uzima da su sve vrednosti preko 0.8 aktivni izlazi, a sve vrednosti ispod 0.2 su neaktivni izlazi. Izlazne vrednosti između 0.2 i 0.8 će biti razmatrane u tranziciji, a moguća je i promena efekta regiona tranzicije na ponašanje sigmoidnog neurona preko promene vrednosti τ (slika gore). Što se tiče neaktivnog ulaza (vrednost 0.0), on uslovljava to da takva nulta stimulacija, modulirana sa bilo kojom vrednošću težinskog odnosa, proizvodi nultu aktivaciju. Tako, nulta aktivacija sigmoidnog neurona prouzrokuje izlaz od 0.5, što znači polovinu između validnih izlaznih stanja. O ovoj značajnoj aktivacionoj funkciji biće reči u kasnijim razmatranjima koja su vezana za odgovarajući model veštačke neuronske mreže.

Kompetitivna aktivaciona funkcija, koja se koristi kod "takmičarskih" neurona, data je sledećim izrazom:

$$f_i^k(net_i(t)) = \begin{cases} 1 & \text{ako je } net_i(t) = \max\{net_j(t)\}, 1 \leq j \leq n \\ 0 & \text{ostalo} \end{cases}$$

Ova jednačina ukazuje na to da će, od svih n neurona kompetitivnog sloja, jedino kompetitivni neuron i "okidati" ("pobednik u takmičenju" sa ostalim), dok će svi ostali ostati neaktivni. Ovakvo ponašanje znači da će svaki ulazni vektor, koji je doveden do kompetitivnih neurona, uzrokovati da jedan od njih dominira nad ostalim i on će jedini generisati aktivan izlaz. Da bi se uočilo kako se ovo ponašanje manifestuje kod odgovarajuće strukture, potrebno je videti prikaz na slici dole. Vidi se da je neuron **1** "pobedio u takmičenju" i da je prosledio *memorisani uzorak* izlaznom sloju, a kada u narednim iteracijama pobednik bude neuron i izlaz iz mreže postaje *memorisani uzorak* koji je "uskladišten" u vezama između neurona i i izlaznog sloja.



Proces "takmičenja" kompetitivnih neurona

Ako se ponovo uzme u razmatranje mogućnost primene linearno aktiviranog neurona, treba konstatovati da je njegov izlaz suma svih moduliranih ulaza koje je on primio. Neka se neuroni u kompetitivnom sloju koriste kao opslužioc sloja sa linearnim neuronima. Tada, ako se samo jedan kompetitivni neuron aktivira za neki od ulaznih vektora koji su dovedeni do kompetitivnog sloja, svaki linearni neuron će primiti samo jedan aktivni ulazni signal – onaj koji je dobijen od pobedničkog neurona iz kompetitivnog sloja, a od bilo kog drugog kompetitivnog neurona neće primiti nikakvu aktivaciju. Na taj način linearni neuroni postaju *multiplekser uzorka*, pri čemu generišu izlazni uzorak koji je potpuno jednak uzorku koji je "uskladišten" u međusobnim težinskim odnosima između sloja linearnih neurona i pobedničkog kompetitivnog neurona. Tako se dobija *memorisani uzorak* koji može da bude n,m -dimenzionalan (n kompetitivnih neurona i m linearnih neurona).

Algoritmi učenja veštačkih neuronskih mreža

Modeli neuronskih mreža (neurona) su najčešće **dinamički**, jer se razvijaju u funkciji vremena, što se i vidi po prethodnim izrazima. Opšta diferencijalna jednačina,

$$\dot{x} = g_i(x_i, net_i)$$

predstavlja izlaz i -tog neurona, pri čemu je očigledno da se, zbog **dinamičkih neuronskih modela**, radi o diferenciranju po vremenu. Kako ulaz i -tog neurona net_i , zavisi od izlaza

mnogih neurona sa kojima je u vezi, praktično se razvija sistem spregnutih nelinearnih diferencijalnih jednačina.

Potrebno je i težinske odnose između neurona posmatrati kao **dinamički sistem**. To znači da imamo sistem diferencijalnih jednačina za težinske odnose,

$$\dot{w}_{ij} = A_i(w_{ij}, x_i, x_j, \dots)$$

gde A_i predstavlja **algoritam učenja**. Proces učenja, baziran na algoritmima učenja, određuje težinske odnose kroz iterativni postupak njihove modifikacije. Sistem veštačke neuronske mreže na taj način uči, tako da je neophodno pomenuti najčešće korišćene algoritme učenja (tabela dole).

Izbor odgovarajućeg algoritma učenja, shodno aplikaciji za koju se mreža koristi, predstavlja problem pri projektovanju mreže.

Pitanja na koja treba odgovoriti mogu biti:

- Koje performanse treba da ima sistem neuronske mreže u pogledu obučavanja?
- Da li je potrebno razviti čisto dinamički sistem neuronske mreže?
- Hoće li faza učenja biti realizovana u "on-line" ili "off-line" režimu?
- Da li je brzina konvergencije važna za učenje neuronske mreže?

Odgovori na ova pitanja su direktno vezani za algoritme učenja koji se danas koriste, a najznačajniji su dati i tabeli. Postoje varijacije i usavršavanja osnovnih algoritama učenja, u pogledu konvergencije, pa se u okviru ovog kursa posebno razmatra konvergencija prilikom primene generalisanog delta pravila za mrežu sa prostiranjem greške unazad ("back-propagation").

Pored ovih najznačajnijih algoritama učenja, koji se koriste za supervizorsko i nesupervizorsko učenje, postoje i neuronske mreže sa obučavanjem bez nadgledanja. One uče kroz proces dovođenja obučavajućeg vektora na njihove ulaze, a da pri tome vrednosti izlaznog vektora nisu poznate. U mrežu je ugrađena sposobnost klasifikacije ulaza u različite kategorije, na bazi samoorganizovanja i korišćenjem internih pravila. Jedna od paradigmi učenja bez nadgledanja je vezana za mrežu sa **kompetitivnim obučavanjem**. Osnovna snaga ove vrste mreže je u osposobljavanju mreže da vrši jednostavnu identifikaciju ulaznih varijanti, što je iskorišćeno i primenjeno u sprovedenim istraživanjima kod nas kroz efikasnu proceduru identifikacije objekata snimljenih kamerom.

Pregled najznačajnijih algoritama učenja

Algoritam učenja	Matematička formulacija	Napomene
- Hebb-ovo pravilo (1949)	$d\phi/dt = O(t)[I_i(t) - O(t)\phi_i(t)]$	$O(t)$ – izlaz neurona $I_i(t)$ – ulazi u neuron $\phi_i(t)$ – jačina sinaptičkih veza
- Widrow-Hoff-ovo pravilo (1960) (LMS-“least mean square” algoritam, tj. algoritam najmanje kvadratne razlike)	$w(t+1) = w(t) + 2\mu \varepsilon_k \mathbf{x}_k$	\mathbf{x}_k – k -ti ulazni vektor w – težinski vektor μ – pozitivna konstanta ε_k – trenutna razlika (greška) između zahtevane vrednosti izlaza (d_k) i aktuelne vrednosti (y_k)
- Generalisano delta pravilo (1986) (prostiranje greške unazad)	$w_{ji}(t+1) = w_{ji}(t) + \eta \delta_{pj} \mathbf{x}_i$	\mathbf{x}_i – i -ti ulazni vektor w_{ji} – vektor težinskih odnosa η – parametar učenja δ_{pj} – greška vezana za izlaz neurona $\delta_{pj} = (y_{pj} - o_{pj}) - y_{pj}$ je zahtevana vrednost izlaza, a o_{pj} je aktuelna vrednost p -tog obučavajućeg vektora za j -ti neuron

Modeli veštačkih neuronskih mreža

Primena sistema veštačkih neuronskih mreža je veoma široka. Tokom devedesetih godina 20. veka počele su da se koriste i u različitim oblastima proizvodnog mašinstva, pri čemu su se izdvojile tri osnovne kategorije upotrebe: *klasifikacija*, *predikcija* i *funkcionalna aproksimacija*.

Funkcionalna aproksimacija je predstavljena skupom nezavisnih promenljivih $\mathbf{x}=[x_1, x_2, \dots, x_m]^T$ i izlaznim signalom $y=f(x_1, x_2, \dots, x_m)$, koji su grupisani u parove $\{y, f(\mathbf{x})\}$.

Predikcija je predstavljena prethodnim vrednostima stanja sistema, $\mathbf{X}=[x(k), x(k-1), \dots, x(k-n)]^T$, kao i trenutnim i prethodnim vrednostima ulaza, $\mathbf{U}=[u(k), u(k-1), \dots, u(k-m)]^T$, koje sa budućim stanjima sistema, $x(k+1)=f(\mathbf{X}, \mathbf{U})$ formiraju parove $\{x(k+1), (\mathbf{X}, \mathbf{U})\}$.

Klasifikacija je predstavljena skupom ulaza $\{^c x_i, i=1, \dots, N_j\}$, koji sa specificiranom klasom C_j formira parove $\{C_j, ^c x_j\}$.

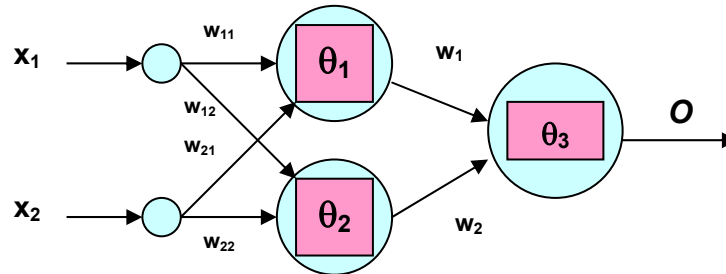
Najrasprostranjeniji *modeli veštačkih neuronskih mreža* su:

- Perceptron;
- **“Backpropagation” (BP) neuronska mreža;**
- Asocijativne neuronske mreže;
- *Hopfield*-ove neuronske mreže;
- **ART neuronske mreže (ART-1, ART-2, ART-3);**
- *Fuzzy* asocijativne neuronske mreže;
- Samoorganizujuće neuronske mreže.

Od svih nabrojanih modela veštačkih neuronskih mreža, u najzastupljenije se ubrajaju pre svih “backpropagation” (BP) mreža, a i ART-1 mreža ima široku primenu, tako da će se dati objašnjenja za ove dve neuronske mreže, a kako perceptron predstavlja preteču BP neuronske mreže objašnjenja će početi od ovog modela.

Perceptron

Perceptron predstavlja najraniji model veštačke neuronske mreže, a prvi pionirski radovi datiraju od *Warren McCulloch*-a i *Walter Pitts*-a iz 1943. godine, pa preko *Rosenblatt*-a (1958 i 1962) [138] do *Minsky*-og i *Papert*-a (1969). Arhitektura perceptrona je jednostavna, koristi prostiranje signala u jednom smeru ("feedforward"), ima jedan ili više slojeva neurona između ulaznih i izlaznih neurona (najčešće ceo perceptron ima dva sloja) i funkcioniše na bazi supervizorskog učenja. Šematski prikaz perceptrona je dat na slici.



Elementarna struktura perceptrona

Kod ove neuronske mreže, koja koristi linearni tip neurona sa pragom θ , izlazna vrednost je data sledećim izrazom:

$$o = f(\text{net}) = \begin{cases} 1 & \text{net} \geq \Theta \\ 0 & \text{net} < \Theta \end{cases}$$

pri čemu je $\text{net} = \sum w_{ij} x_j$.

Najreprezentativnije objašnjenje arhitekture perceptrona, kao sistema veštačke neuronske mreže, dao je *Rosenblatt*. Strukturu perceptrona čini skup ulaznih neurona ($\{s\}$ – neuroni), skup skrivenih neurona ($\{a\}$ – neuroni) i jedan izlazni neuron ($\{r\}$ – neuron). *Rosenblatt*-ova osnovna ideja je bila vezana za to, da bi jačinu veza između ulaznog sloja neurona i skrivenog sloja neurona trebalo slučajno izabrati po nekom zakonu verovatnoće i fiksirati njihove vrednosti tokom celog procesa učenja. Algoritam učenja bi zatim mogao da realizuje podešavanje težinskih odnosa između skrivenog sloja neurona i jedinog izlaznog neurona. Naime, on je smatrao da ako ima dovoljno različitih tipova neurona u skrivenom sloju koji reprezentuju različite logičke iskaze iz ulaznog sloja neurona, onda je proces učenja od neurona iz skrivenog sloja do izlaznog neurona u stanju da realizuje kompleksno logičko zaključivanje.

Algoritam za učenje perceptrona se može predstaviti kroz šest koraka. Konstatovano je da proces učenja uvek konvergira u konačnom broju iteracija prilikom korektne klasifikacije svih članova obučavajućeg skupa i da perceptronski algoritam učenja traži minimalnu vrednost aktivacione funkcije.

Algoritam učenja perceptrona

Korak 1: Inicijalizacija. Neka je $t=0$ i neka vektor inicijalnih težinskih odnosa $\mathbf{w}(t) \in \mathbb{R}^a$ definiše inicijalno preslikavanje od skrivenih neurona do izlaznog neurona.

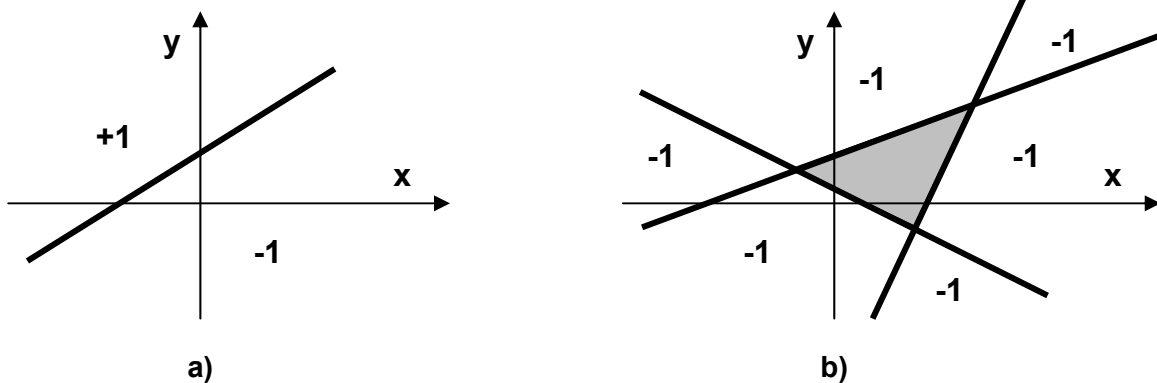
Korak 2: Izabrati stimulaciju perceptrona primenjujući uzorke iz obučavajućeg skupa \mathbf{x}^k na slučajan način, što se može iskazati pomoću $\mathbf{x}^k(t) = (\mathbf{s}^k(t), \mathbf{o}^k(t)) \in \{0, 1\}$, gde je \mathbf{o}^k ciljni vektor, koji predstavlja zahtevani odgovor na stimulaciju.

Korak 3: Izračunati aktivaciju neurona u skrivenom sloju h^k , na osnovu aktivacije ulaznih neurona, $h^k(t) = q(\mathbf{s}^k(t))$.

Korak 4: Izračunati aktivaciju izlaznog neurona mreže r^k , na osnovu aktivacije neurona iz skrivenog sloja, $r^k(t) = f(\text{net}) = f(\mathbf{w}(t)^T, h^k(t))$, pri čemu je $f(\text{net}) = 1$ ako je $\text{net} \geq 0$ i $f(\text{net}) = 0$ ako je $\text{net} < 0$.

Marvin Minsky i Seymour Papert su 1969. godine dali detaljnu analizu perceptrona u pogledu mogućnosti i ograničenja primene. Oni su konstatovali da je njegova glavna karakteristika vezana za linearnu diskriminacionu funkciju u prostoru uzoraka i to u konačnom broju iteracija. To znači da se korektna klasifikacija može sprovesti samo za linearno razdvajajuće klase uzoraka.

Na slici dole su prikazani regioni koje perceptron može lako da identifikuje. Za problem prikazan na slici pod a, samo jedan procesirajući element perceptrona je potreban, s obzirom da je cela površina razdeljena jednom linijom na dva odvojena regiona, koji su identifikovani sa +1 i -1. Međutim, region koji je osenčen (+1) na slici pod b, formiran je od mnogo linija, tako da je neophodno razviti više perceptrona koji će rešiti taj problem.



Identifikacija regiona

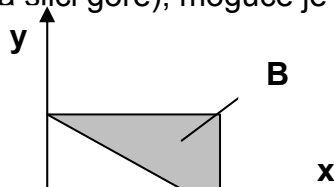
Da bi se ilustrovalo ponašanje perceptrona pri realizaciji logičkog **ILI**, polazi se od tabele koja sadrži istinite iskaze za regione prikazane na slici. Za realizaciju rezultata datih u tabeli dovoljna je jednostavna struktura troslojnog perceptrona koji je prikazan na slici gore, a koji ima sledeće vrednosti težinskih odnosa w_{ij} i praga neurona θ_1, θ_2 i θ_3 :

$$w_{11}=1, w_{12}=1, w_{21}=1, w_{22}=1,$$

$$w_1=1, w_2= -1,$$

$$\theta_1= 0.5, \theta_2 =1.5, \theta_3 = 0.5.$$

Prema datom izrazu, lako se može izračunati izlaz iz ovog troslojnog perceptrona za različite ulaze x_1 i x_2 , npr. ako je $x_1 = x_2 = 0$ (tabela), a prema datim vrednostima za w i θ , očigledno da je izlaz $o=f(net)=0$ jer je $net < \theta$. Na sličan način, pomoću troslojnog perceptrona sa tri neurona u srednjem sloju (modifikovana struktura u odnosu na perceptron prikazan na slici gore), moguće je izvršiti identifikaciju regiona **A** i **B** sa slike:



Logičko **ILI**

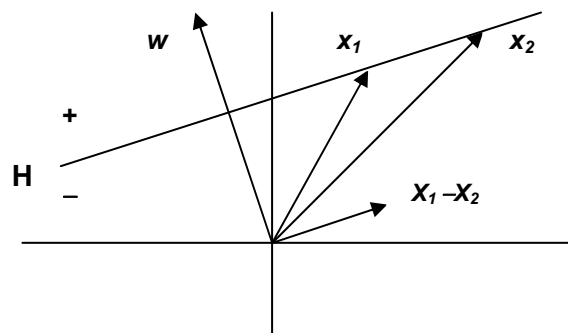
Tačka 1(x_1)		Tačka 2(x_2)		Izlaz
U reg. A	(0)	U reg. B	(0)	0
U reg. A	(0)	Van reg. B	(1)	1
Van reg. A	(1)	U reg. B	(0)	1
Van reg. A	(1)	Van reg. B	(1)	0

Primer za perceptron (logičko **ILI**)

Višeslojni perceptroni proširuju mogućnosti klasifikacije na šire skupove mogućih međuođnosa klasa, u zavisnosti od broja slojeva. Zato se uvode pojmovi *hiperravan* i *hiperprostor*, tako da su *hiperravni*, za n -dimenzionalni prostor (*hiperprostor*), određene sa $n-1$ dimenzijom. U n -dimenzionalnom ulaznom prostoru X , hiperravan H koja vrši separaciju, definisana je na sledeći način:

$$w_1x_1 + w_2x_2 + \dots + w_nx_n = \theta = -w_0$$

pri čemu je vektor težinskih odnosa $\mathbf{w} = [w_1, w_2, \dots, w_n]^T$ upravran na tu hiperravan (slika dole). Da bi se to pojasnilo treba posmatrati jednostavan primer sa dva n -dimenzionalna ulazna vektora, \mathbf{x}_1 i \mathbf{x}_2 , koji su locirani na hiperravni. Slika prikazuje dvodimenzionalan slučaj, gde se uočava da su vektor \mathbf{w} i hiperravan H međusobno upravni. Hiperravan H deli n -dimenzionalni ulazni prostor X na dva polu-prostora, region X^+ gde je $\sum w_i x_i > -w_0$ (izlaz je 1), odnosno region X^- gde je $\sum w_i x_i \leq -w_0$ (izlaz je 0). To znači da je u trodimenzionalnom prostoru ravan definisana dvema dimenzijama, tako da ona može da izvrši razdvajanje takvog prostora na dva odvojena regiona, a sa dve ravni na tri ili četiri regiona, itd. Na taj način se može, egzaktnom analizom, dokazati da je višeslojni perceptron sa jednim skrivenim slojem, u stanju da sa proizvoljnom tačnošću uniformno aproksimira bilo koju realnu kontinualnu funkciju na konačnoj realnoj osi, dok sa dva skrivena sloja može da uniformno aproksimira bilo koju realnu kontinualnu funkciju više argumenata.



Primer razdvajanja dvodimenzionalnog prostora

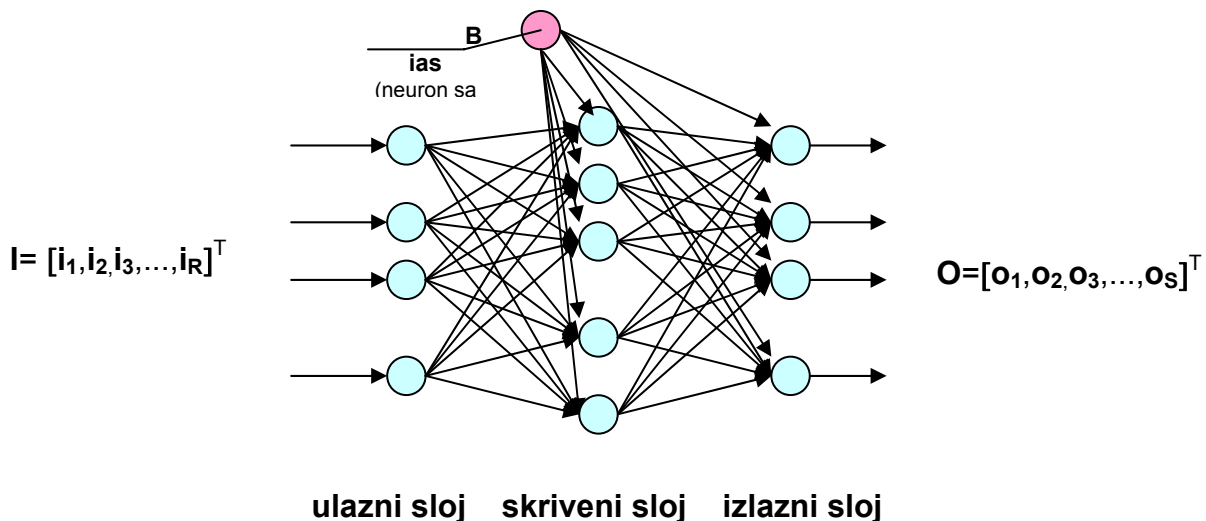
Perceptron, kao linearni separator, ima velika ograničenja u pogledu nelinearnih preslikavanja. To znači da je problem njegove primene vezan za nelinearno razdvajanje klasa, jer on koristi binarnu aktivacionu funkciju, koja uslovljava da neuronska mreža ima diskontinuitete.

"Backpropagation" (BP) neuronska mreža

"Backpropagation" (BP) neuronska mreža je razvijena sa ciljem da se problem nelinearnog preslikavanja iz ulaznog prostora u izlazni prostor uspešno reši, pri čemu se ostvaruje modifikacija težinskih odnosa i između ulaznog i skrivenog sloja neurona. BP je, kao i perceptron, neuronska mreža sa prostiranjem signala unapred, koja takođe realizuje supervizorski vid učenja, sa različitim aktivacionim funkcijama i algoritmima učenja. BP mreža koristi *gradijentni postupak* pri obučavanju, koji je analogan procesu minimizacije greške. To znači da uči preslikavanja iz ulaznog prostora uzorka u izlazni prostor, kroz *proces minimizacije greške* između aktuelnog izlaza koji je ostvarila mreža i zahtevanog izlaza, na osnovu skupa obučavajućih parova, odnosno primera. Proces učenja počinje sa prezentacijom ulaznog oblika uzorka BP mreži, koji prostiranjem kroz mrežu ostvaruje izlazni oblik. BP mreža zatim primenjuje *generalisano delta pravilo* da bi se utvrdila greška na izlazu, koju prostiranjem unazad preko skrivenog sloja, koristi za polako modifikovanje svakog težinskog odnosa između neurona, što se ponavlja za svaki novi uzorak. Generalisano delta pravilo obezbeđuje konvergenciju procesa učenja do zadanog nivoa tačnosti kroz iterativni proces adaptacije težinskih odnosa.

Sistem BP neuronske mreže koristi sigmoidnu aktivacionu funkciju. Kao što je rečeno ranije, ova aktivaciona funkcija je nelinearna, neprekidna i diferencijabilna, što obezbeđuje da se proces učenja BP mreže uspešno realizuje. Primenom *generalisanog delta pravila* utvrđena greška se koriguje, tako što se koristi prvi izvod sigmoidne funkcije $f'(net) = f'(S)$. Ukoliko sigmoidna funkcija $f(net)$ brže raste, onda treba izvršiti veću korekciju greške i obrnuto.

Elementarna arhitektura BP mreže ima tri sloja (slika dole), koji su potpuno povezani, s tim što se, ne tako retko, koristi i struktura sa više od jednog skrivenog sloja. Broj neurona u svakom sloju se razlikuje, u zavisnosti od oblasti primene, kao i broj skrivenih slojeva. Najčešće postoji samo jedan skriveni sloj, jer je BP mreža sa takvom strukturom u stanju da obezbedi reprodukovanje skupa zahtevanih izlaznih oblika za sve obučavajuće parove. Na slici se vidi da mreža može da poseduje i ekstra ulaz - neuron koji je uvek aktivan i ima izlazno stanje 1 (konstantna aktivacija), poznat kao "*bias*" neuron. On je povezan sa svim neuronima u skrivenom i izlaznom sloju. Ponaša se u mreži kao i svaki drugi neuron, pri čemu ima zadatak da participira u procesu učenja tako što težinski odnos između "*bias*" neurona i svakog neurona u sledećem sloju formira aktivaciju koja mora biti savladana ostatkom ulaza u svaki neuron, usled čega je i njihova aktivacija kontrolisana. "*Bias*" neuron obezbeđuje konstantan član u težinskim sumama neurona narednog sloja, što rezultira poboljšanjem konvergentne karakteristike BP mreže.



Elementarna arhitektura BP neuronske mreže

BP mreža koristi **generalisano delta pravilo** prilikom modifikacije težinskih odnosa između neurona, čime se ostvaruju osnovne karakteristike mreže a to je **generalizacija** i **nelinearno razdvajanje**. Da bi se shvatile prednosti ovih karakteristika u nastavku se opisuju generalisano delta pravilo i postupak obučavanja BP mreže.

Neka je M broj obučavajućih parova koji se prezentiraju mreži, R broj neurona u ulaznom sloju i S broj neurona u izlaznom sloju. Obučavajući parovi $(\mathbf{x}_1, \mathbf{y}_1) \dots (\mathbf{x}_M, \mathbf{y}_M)$, koji su dovedeni na ulaz mreže, obezbeđuju mreži mogućnost da realizuje nelinearno preslikavanje između \mathbf{x}_i i \mathbf{y}_i vektora, za $i=1, \dots, M$.

Neka je L broj slojeva mreže i neka je sa l označen svaki sloj:

$l = 0$ ulazni sloj,
 $l = 1, \dots, L - 1$ skriveni sloj(evi),
 $l = L$ izlazni sloj.

Neuronska mreža ima K_l neurona u svakom sloju, gde je $l = 1, \dots, L$. Ulazi u k -ti neuron l -tog sloja su dati preko I_{ij}^l , za $j=1, \dots, K_{l-1}$. Ovi ulazi su istovremeno i izlazi neurona prethodnog sloja, koji se mogu označiti sa O_{ij}^{l-1} . Zajedno sa pragom θ_k^l , za k -ti neuron u sloju l , sumirana ulazna vrednost je data sledećim izrazom:

$$S_{ik}^l = net_{ik}^l = \sum_{j=1}^{K_{l-1}} w_{kj}^l I_{ij}^l + \theta_k^l$$

gde su w_{kj}^l težinski odnosi između neurona j i k .

Izlaz iz k -tog neurona u l -tom sloju je ostvaren kao rezultat aktivacije neurona. Naime, propušta se ulaz S_{ik}^l preko funkcije prenosa $h_k^l(s)$, kako bi se korišćenjem aktivacione funkcije $f_k^l(x)$ generisao izlaz, dat preko izraza:

$$O_{ik}^l = f_{ik}^l(S_{ik}^l)$$

Greška na izlaznom sloju L , BP neuronske mreže, je predstavljena razlikom između izračunatih i zadatih izlaza:

$$E_{pi} = \frac{1}{2} \sum_{k=1}^{K_L} (y_{ik} - O_{ik}^L)^2$$

Ova greška se minimizira uz modifikaciju težinskih odnosa. Da bi se to ilustrovalo u nastavku se daje izraz za izračunavanje greške za mrežu sa dva skrivena sloja ($L=3$):

$$E_{pi} = \frac{1}{2} \sum_{k=1}^{K_3} (y_{ik} - O_{ik}^3)^2 = \frac{1}{2} \sum_{k=1}^{K_3} (y_{ik} - f_k^3(S_{ik}^3))^2 = \dots =$$

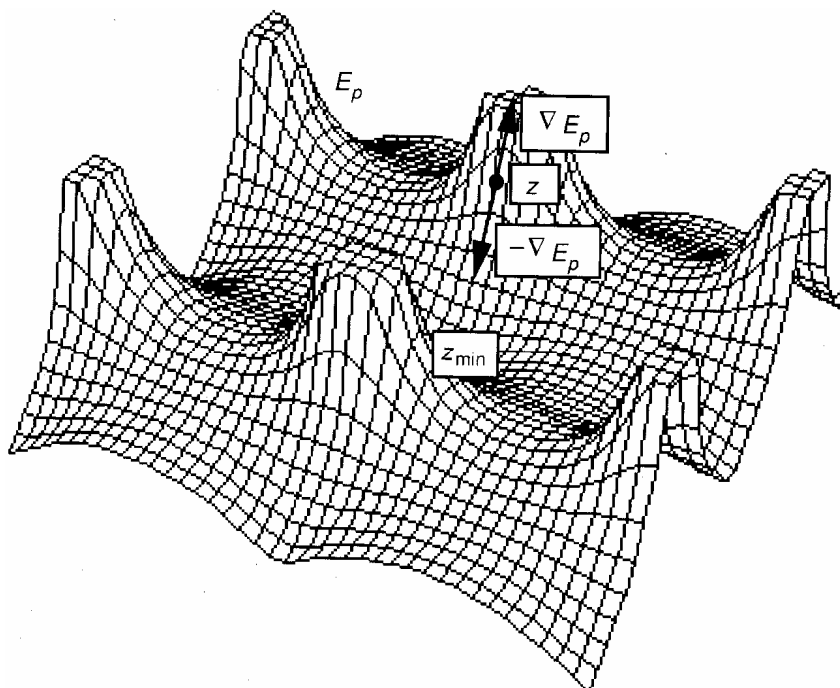
$$= \frac{1}{2} \sum_{k=1}^{K_3} \left(y_{ik} - f_k^3 \left(\sum_{j=1}^{K_2} w_{kj}^3 f_j^2 \left(\sum_{u=1}^K w_{ju}^2 f_u^1 \left(\sum_{v=1}^{K_0} w_{jv}^1 O_{iv}^0 + \theta_u^1 \right) + \theta_j^2 \right) + \theta_k^3 \right) \right)^2$$

gde je $O_{iv}^0 = I_{iv}^1 = x_{iv}$.

Na osnovu jednačine za E_p može se izračunati gradijent greške E_p , s obzirom na težinske odnose w_{kj}^l , tako da je za izlazni sloj ($l=3$):

$$\frac{\partial E_{pi}}{\partial w_{kj}^3} = -(y_{ik} - O_{ik}^3) \frac{\partial O_{ik}^3}{\partial S_{ik}^3} \frac{\partial S_{ik}^3}{\partial w_{kj}^3} = -(y_{ik} - O_{ik}^3) f_k^3 (S_{ik}^3) I_{ij}^3$$

Negativni predznak ukazuje na to da je pravac promene težinskih odnosa sa negativnim gradijentom ∇E_p , što je prikazano na slici dole za tačku z hipotetičke kompleksne površi greške E_p u prostoru težinskih odnosa. Težinski odnosi se iterativno menjaju dok greška E_p ne ostvari minimum u tački z_{min} .



Grafički prikaz pravca prostiranja gradijenta greške E_p

Inkrement promene težinskih odnosa, za zadati parametar učenja η , je:

$$\Delta_l w_{kj}^3 = \eta (y_{ik} - O_{ik}^3) f_k^3 (S_{ik}^3) I_{ij}^3 = \eta \delta_{ik}^3 I_{ij}^3$$

gde je $\delta_{ik}^3 = (y_{ik} - O_{ik}^3) f_k^3 (S_{ik}^3)$.

Parametar učenja η se najčešće bira u opsegu od 0.25 do 0.75. Ovaj parametar određuje brzinu obučavanja, tako što je za veće vrednosti obučavanje brže i obrnuto.

Suviše velike vrednosti ovog parametra utiču na stabilnost mreže i uzrokuju nezadovoljavajući proces obučavanja, dok s druge strane, ako je vrednost mnogo mala, obučavanje je veoma sporo. Ponekad se koristi i metoda promenljivog parametra, kod koje je u početku obučavanja parametar η veći, dok se tokom obučavanja on smanjuje, omogućavajući tako mreži bolje karakteristike učenja.

Slično prethodnom, za drugi skriveni sloj ($l=2$):

$$\frac{\partial E_{pi}}{\partial w_{ju}^2} = - \sum_{k=1}^{K_3} (y_{ik} - o_{ik}^3) \frac{\partial o_{ik}^3}{\partial s_{ik}^3} \frac{\partial s_{lik}^3}{\partial o_{ij}^2} \frac{\partial o_{ij}^2}{\partial s_{ij}^2} \frac{\partial s_{lij}^2}{\partial w_{ju}^2} = - \sum_{k=1}^{K_3} (y_{ik} - o_{ik}^3) f_k^3 (s_{ik}^3) w_{kj}^3 f_j^2 (s_{ij}^2) I_{iu}^2$$

$$\Delta_i w_{ju}^2 = \eta \delta_{ij}^2 I_{ij}^2, \quad \text{gde je } \delta_{ij}^2 = f_j^2 (s_{ij}^2) \sum_{k=1}^{K_3} \delta_{ik}^3 w_{kj}^3$$

Za prvi skriveni sloj ($l=1$):

$$\frac{\partial E_{pi}}{\partial w_{uv}^1} = - \sum_{k=1}^{K_3} (y_{ik} - o_{ik}^3) \frac{\partial o_{ik}^3}{\partial s_{ik}^3} \frac{\partial s_{lik}^3}{\partial o_{ij}^2} \frac{\partial o_{ij}^2}{\partial s_{ij}^2} \frac{\partial s_{ij}^2}{\partial o_{iu}^1} \frac{\partial o_{iu}^1}{\partial s_{iu}^1} \frac{\partial s_{iu}^1}{\partial w_{uv}^1}$$

$$= - \sum_{k=1}^{K_3} (y_{ik} - o_{ik}^3) f_k^3 (s_{ik}^3) w_{kj}^3 \sum_{j=1}^{K_2} f_j^2 (s_{ij}^2) w_{ju}^2 f_u^1 (s_{iu}^1) x_{iv}$$

$$\Delta_i w_{uv}^1 = \eta \delta_{iu}^1 x_{iv}, \quad \text{gde je } \delta_{iu}^1 = f_u^1 (s_{iu}^1) \sum_{j=1}^{K_2} \delta_{ij}^2 w_{ju}^2$$

Vrednosti $\Delta_i w_{kj}^3, \Delta_i w_{ju}^2, \Delta_i w_{uv}^1$ su korišćene za modifikaciju težinskih odnosa tokom procesa obučavanja, odnosno treniranja BP neuronske mreže, a obučavanje se završava kada su ove vrednosti bliske nuli. Algoritam učenja BP mreže se, na osnovu opisanog generalisanog delta pravila, može predstaviti kroz sedam koraka.

Algoritam učenja BP neuronske mreže

Korak 1: Inicijalizacija i filtriranje. Dovedi ulazni vektor $\mathbf{I}_{ij} = (x_{i1}, x_{i2}, \dots, x_{iR})$ do neurona u ulaznom sloju.

Korak 2: Izračunati izlazne vrednosti za svaki neuron u mreži korišćenjem izraza:

$$O_{ik}^l = f_{ik}^l (S_{ik}^l), \quad \text{gde je } S_{ik}^l = net_{ik}^l = \sum_{j=1}^{K_l} w_{kj}^l I_{ij}^l + \theta_k^l.$$

Korak 3: Izračunati grešku koju je mreža na izlazu generisala $E_{pi} = \frac{1}{2} \sum_{k=1}^{K_L} (y_{ik} - o_{ik}^L)^2$.

Korak 4: Za izlazni sloj, $l = L$, izračunati vrednosti promene težinskih odnosa pomoću izraza: $\Delta_i w_{kj}^L = \eta (y_{ik} - o_{ik}^L) f_k^L (s_{ik}^L) I_{ij}^L = \eta \delta_{ik}^L I_{ij}^L$

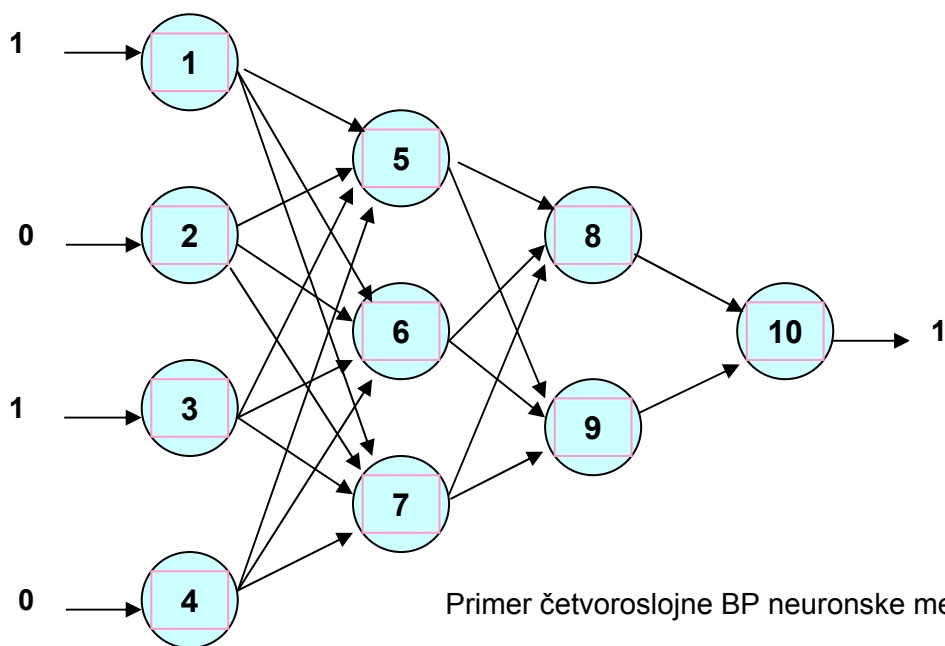
Korak 5: Za skrivene slojeve, $l = 1, \dots, L-1$, izračunati vrednosti promene težinskih odnosa

$$\Delta_i w_{ju}^l = \eta \delta_{ij}^l I_{ij}^l, \quad \text{gde je } \delta_{ij}^l = f_j^l (s_{ij}^l) \sum_{k=1}^{K_{l+1}} \delta_{ik}^{l+1} w_{kj}^{l+1}$$

Korak 6: Ažurirati (modifikovati) sve vrednosti inkrementalne promene težinskih odnosa $w_{ju}^l = w_{ju}^l + \Delta_i w_{ju}^l$, za $l = 1, \dots, L$.

Korak 7: Ponoviti sve prethodne korake dok greška koju generiše mreža ne bude bliska nuli ili dovoljno mala.

Da bi proces obučavanja bio jasno prezentiran, najbolje je ilustrovati ceo postupak jednostavnim primerom. Naime, četvoroslojna BP neuronska mreža, prikazana na slici, se trenira korišćenjem jednostavnih obučavajućih parova. Obučavajući par, (1010,1), je upotrebljen u ovom primeru za demonstraciju opisanog mehanizma ažuriranja težinskih odnosa, uz parametar učenja $\eta=0.5$. Različite sigmoidne aktivacione funkcije su korišćene za neurone u skrivenim slojevima i izlaznom sloju, a date su u tabeli dole. Prvi izvodi ovih funkcija su takođe dati u tabeli. Pre izračunavanja, binarni obučavajući parovi moraju biti promenjeni realnim brojevima, koji će se koristiti umesto 0 i 1. Tako će 0.1 predstavljati 0, a 0.9 će odgovarati 1. Neuroni ulaznog sloja prosleđuju ka sledećem sloju one vrednosti koje su primili od ulaznog vektora (1010).



Pregled korišćenih sigmoidnih aktivacionih funkcija

Neuron	Sigmoidna aktivaciona funkcija	Prvi izvod aktivacione funkcije
--------	--------------------------------	---------------------------------

5	$f(\text{net}) = f(S) = f(x) = (1 + e^{-3x})^{-1}$	$f'(\text{net}) = f'(S) = f'(x) = \frac{3e^{-3x}}{(1 + e^{-3x})^2}$
6	$f(\text{net}) = (1 + e^{-4x})^{-1}$	$f'(\text{net}) = \frac{4e^{-4x}}{(1 + e^{-4x})^2}$
7	$f(\text{net}) = (1 + e^{-5x})^{-1}$	$f'(\text{net}) = \frac{5e^{-5x}}{(1 + e^{-5x})^2}$
8	$f(\text{net}) = (1 + e^{-6x})^{-1}$	$f'(\text{net}) = \frac{6e^{-6x}}{(1 + e^{-6x})^2}$
9	$f(\text{net}) = (1 + e^{-7x})^{-1}$	$f'(\text{net}) = \frac{7e^{-7x}}{(1 + e^{-7x})^2}$
10	$f(\text{net}) = (1 + e^{-8x})^{-1}$	$f'(\text{net}) = \frac{8e^{-8x}}{(1 + e^{-8x})^2}$

Proces obučavanja se odvija shodno sledećim koracima:

1. Određuju se težinski odnosi w^l i prag neurona θ^l , korišćenjem vrednosti između -0.5 i 0.5 .

$$w^1 = \begin{bmatrix} 0.182 & -0.100 & -0.088 \\ 0.376 & -0.391 & -0.321 \\ 0.206 & 0.416 & 0.055 \\ -0.233 & 0.442 & -0.192 \end{bmatrix} \quad \theta^1 = [0.282 \quad 0.107 \quad -0.155]$$

$$w^2 = \begin{bmatrix} 0.058 & -0.249 \\ -0.281 & -0.332 \\ -0.352 & 0.471 \end{bmatrix} \quad \theta^2 = [-0.187 \quad 0.478]$$

$$w^3 = \begin{bmatrix} 0.231 \\ -0.399 \end{bmatrix} \quad \theta^3 = [0.399]$$

2. Uvodi se ulazni vektor u mrežu.

$$I = [0.9 \quad 0.1 \quad 0.9 \quad 0.1]^T$$

3. Izračunava se, prema prethodno datim izrazima, izlaz neurona u prvom skrivenom sloju ($l=1$),

$$O^1 = [0.874 \quad 0.831 \quad 0.234]$$

4. Izračunava se, prema prethodno datim izrazima, izlaz neurona u drugom skrivenom sloju ($l=2$),

$$O^2 = [0.062 \quad 0.661]$$

5. Izračunava se, prema prethodno datim izrazima, izlaz iz izlaznog sloja mreže ($l=3$),

$$O^3 = [0.768]$$

6. Izračunava se δ^3 , prema prethodno datim izrazima, za grešku koja se odnosi na razliku između zatevanog izlaza i izlaza koji je mreža generisala, pa s obzirom na prvi izvod sigmoidne aktivacione funkcije, dat u tabeli, u sloju $l=3$ biće:

$$\delta^3 = [0.188]$$

7. Izračunavaju se inkrementalne promene težinskih odnosa, Δw^3 , u sloju $l=3$,

$$\Delta w^3 = \begin{bmatrix} 0.0058 \\ 0.0621 \end{bmatrix}$$

8. Izračunava se δ^2 , korišćenjem w^3 i prvih izvoda sigmoidnih aktivacionih funkcija u sloju $l=2$,

$$\delta^2 = [0.0152 \quad -0.1179]$$

9. Izračunava se Δw^2 , promene težinskih odnosa, u sloju $l=2$,

$$\Delta w^2 = \begin{bmatrix} 0.0066 & -0.0515 \\ 0.0063 & -0.0489 \\ 0.0018 & -0.0139 \end{bmatrix}$$

10. Izračunava se δ^1 , korišćenjem w^2 i prvih izvoda sigmoidnih aktivacionih funkcija u sloju $l=1$,

$$\delta^1 = [0.0100 \quad 0.0197 \quad -0.0547]$$

11. Izračunava se Δw^1 , promene težinskih odnosa, u sloju $l=1$,

$$\Delta w^1 = \begin{bmatrix} 0.0045 & 0.0089 & -0.0246 \\ 0.0005 & 0.0010 & -0.0027 \\ 0.0045 & 0.0089 & -0.0246 \\ 0.0005 & 0.0010 & -0.0027 \end{bmatrix}$$

12. Ažuriraju (modifikuju) se sve matrice težinskih odnosa.

$$w^1 = \begin{bmatrix} 0.1865 & -0.0911 & -0.1126 \\ 0.3765 & -0.3900 & -0.3237 \\ 0.2105 & 0.4249 & 0.0796 \\ -0.235 & 0.4430 & -0.1947 \end{bmatrix}$$

$$w^2 = \begin{bmatrix} 0.0646 & -0.3005 \\ -0.2747 & -0.3809 \\ -0.3502 & 0.4571 \end{bmatrix}$$

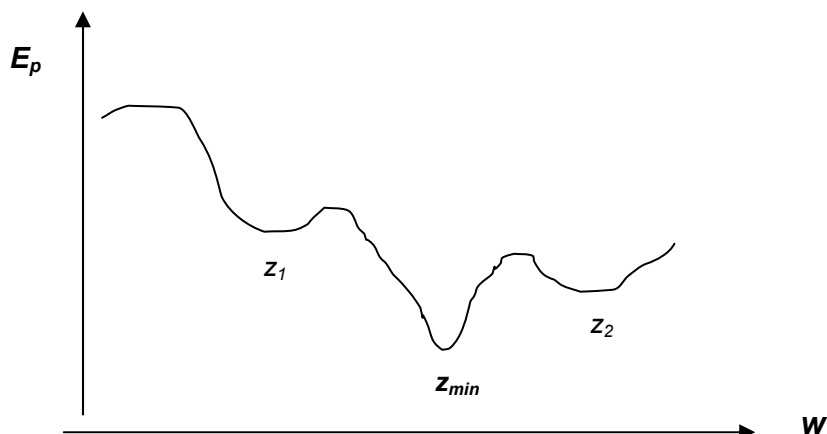
$$w^3 = \begin{bmatrix} 0.2368 \\ -0.3367 \end{bmatrix}$$

13. Dovodi se novi uzorak (ulazni vektor) u mrežu i izračunava se nova izlazna vrednost.

Ova procedura se nastavlja sve dok se ne dođe do zadovoljavajućih rezultata za sve uzorke iz obučavajućeg skupa. Do zadovoljavajućih rezultata BP neuronska mreža dolazi u trenutku kada je greška E_p dovoljno mala i kada mreža na izlazu daje željene vrednosti. Kada se u tom trenutku završi obučavanje mreže, težinski odnosi veza između neurona ostaju nepromenjeni. Tada se mreža može koristiti za ono za šta je obučavana!

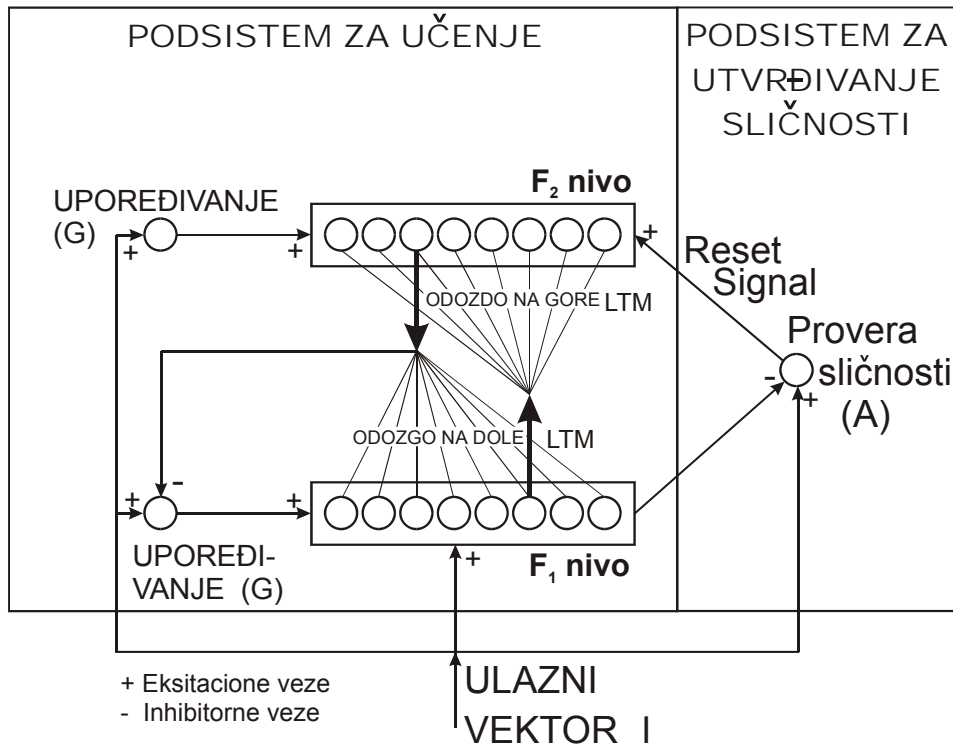
Uspešna primena BP veštačke neuronske mreže zavisi pre svega od njene strukture. Broj neurona u ulaznom i izlaznom sloju je uslovljen reprezentativnim ulaznim i izlaznim vektorom, shodno primeni za koju je struktura BP mreže projektovana. Što se broja skrivenih slojeva tiče, teško je proceniti da li će mreža sa više skrivenih slojeva "raditi" bolje. Takođe je otvoreno pitanje broja neurona u skrivenom sloju(evima). Vreme obučavanja BP mreže direktno zavisi od parametra učenja η . Dodatni problem se javlja i pri izboru parametra učenja η , tako da se do njegove optimalne vrednosti teško dolazi. Ovaj parametar je izuzetno bitan za konvergenciju mreže ka optimalnoj vrednosti greške E_p . Tako je moguće, da se u nekim slučajevima, proces obučavanja završi onog trenutka kada je maksimalna greška koju je mreža ostvarila, ipak manja od najmanje prihvatljive greške za problem koji rešava trenutna aplikacija mreže. Međutim, moguće je da BP neuronska mreža bude "uhvaćena u zamku", tako što će se naći u lokalnom minimumu prilikom konvergencije ka željenoj minimalnoj grešci.

Na slici je dat grafik koji predstavlja poprečni presek hipotetičke kompleksne površine greške E_p u prostoru težinskih odnosa. Tačka z_{min} je zapravo **globalni minimum**. Kao što se vidi, na slici ima i drugih tačaka, z_1 i z_2 , koje predstavljaju **lokalne minimume**, a gradijentni postupak minimizacije greške E_p ima za cilj da se umesto ovih lokalnih minimuma pronađe globalni minimum. Naime i pored ovih upozorenja, konvergencija je onoliko uspešna, koliko je BP neuronska mreža u stanju da brzo dođe do minimalno prihvatljive greške E_p , bez obzira da li se nalazi u lokalnom ili globalnom minimumu. Ponekad se obučavanje BP neuronske mreže zaustavlja u tački koja se nalazi negde na pravcu prostiranja gradijenta ∇E_p , pre nego što je stvarni minimum ostvaren, jer je ta vrednost greške E_p zadovoljavajuća. To upravo govori o kompleksnosti problema konvergencije.



Lokalni i globalni minimum greške E_p u prostoru težinskih odnosa w

"ART-1" veštačka neuronska mreža



Struktura ART-1 veštačke neuronske mreže

Sposobnost čoveka da uči mnoge nove stvari, bez nepoželjenog zaboravljanja onog što je u prošlosti naučeno, predstavlja izuzetnu osobinu ljudske memorije. ART-1 veštačka neuronska mreža (**A**daptive **R**esonance **T**heory) primenjuje ekstenzivnu teoriju adaptivnog učenja, a ona je iskorišćena kao osnova na kojoj je bazirana važna sposobnost ART-1 neuronske mreže da može da sačuva "stare" ulaze, odnosno oblike uzoraka, prilikom učenja novih oblika, što u mnogome podseća na pomenutu osobinu ljudske memorije.

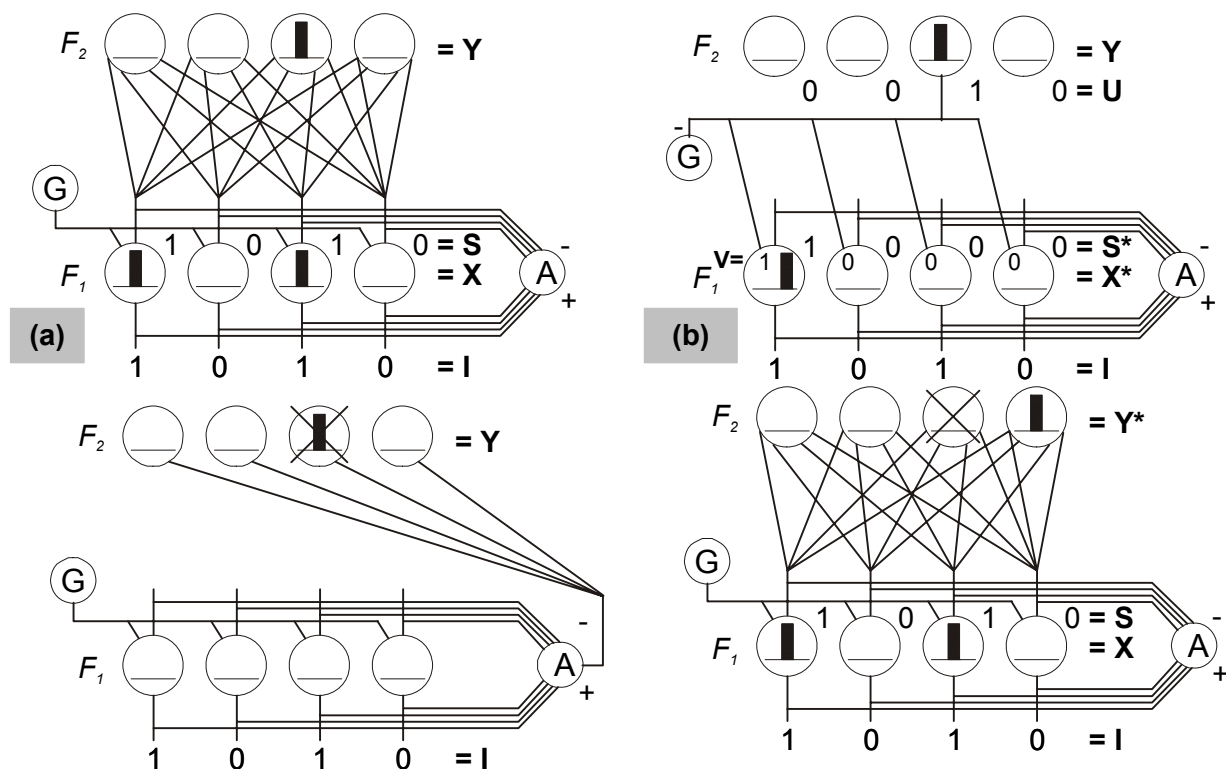
Pored ART-1 neuronske mreže, koja koristi binarni oblik ulaznog vektora, do sada je u svetu razvijena i ART-2 neuronska mreža koja procesira analogne ulazne oblike, što bi u pogledu procesiranja slike značilo da može da radi sa više nivoa sivih nijansi. Pominju se u poslednje vreme i pokušaji realizacije ART-3 neuronske mreže koja je hibridnog tipa.

Osnovu ART-1 neuronske mreže čini kompetitivno učenje, tako da se mehanizam sa povratnim spregama između kompetitivnog i ulaznog sloja (nivoa), koristi za "čuvanje" starih, pri učenju novih informacija. Praktično se informacije (ili signali), između ulaznog nivoa i takmičarskog nivoa, prosleđuju napred ili nazad sve dok mreža ne ostvari *stanje rezonance*, bez koga se proces učenja ne izvršava. Stanje rezonance je veoma značajno za ART-1 neuronsku mrežu u pogledu proteklog vremena u procesu učenja. Moguće su dve situacije u tom smislu: ako je ulazni oblik informacije prethodno naučen pomoću ART-

1 mreže, stanje rezonance se ostvaruje veoma brzo a memorisani oblik se uspešno ažurira; ako ulazni oblik nije odmah prepoznat, ostatak memorisanih oblika se brzo pretražuje i upoređuje sa njim, pa ako se ipak on ne prepozna, mreža odlazi u stanje rezonance kako bi memorisala taj oblik kao novi uzorak. Treba naglasiti da je proteklo vreme za dostizanje stanja rezonance mnogo manje od vremena koje je potrebno za ažuriranje težinskih odnosa između neurona ulaznog i takmičarskog (kompetitivnog) nivoa.

U opštem smislu, ART-1 neuronska mreža ima strukturu sa dva sloja ili nivoa (F_1 i F_2 nivo), čiji su neuroni međusobno povezani preko težinskih odnosa i to *odozdo na gore* i obrnuto *odozgo na dole*, što je prikazano na slici gore. U kompetitivnom sloju (F_2 nivo) neuroni se takmiče za šansu da odgovore na ulazni oblik. Pobednički neuron iz kompetitivnog sloja predstavlja klasifikacionu kategoriju oblika na ulazu. Takmičenje se odvija po algoritmu koji, poštujući određeni kriterijum odabira pobednika, ili, alternativno, kroz sprečavanje - inhibiciju neurona kompetitivnog sloja. U slučaju inhibicije, kompetitivni sloj se dovodi u stanje u kome je samo pobednički neuron aktivan.

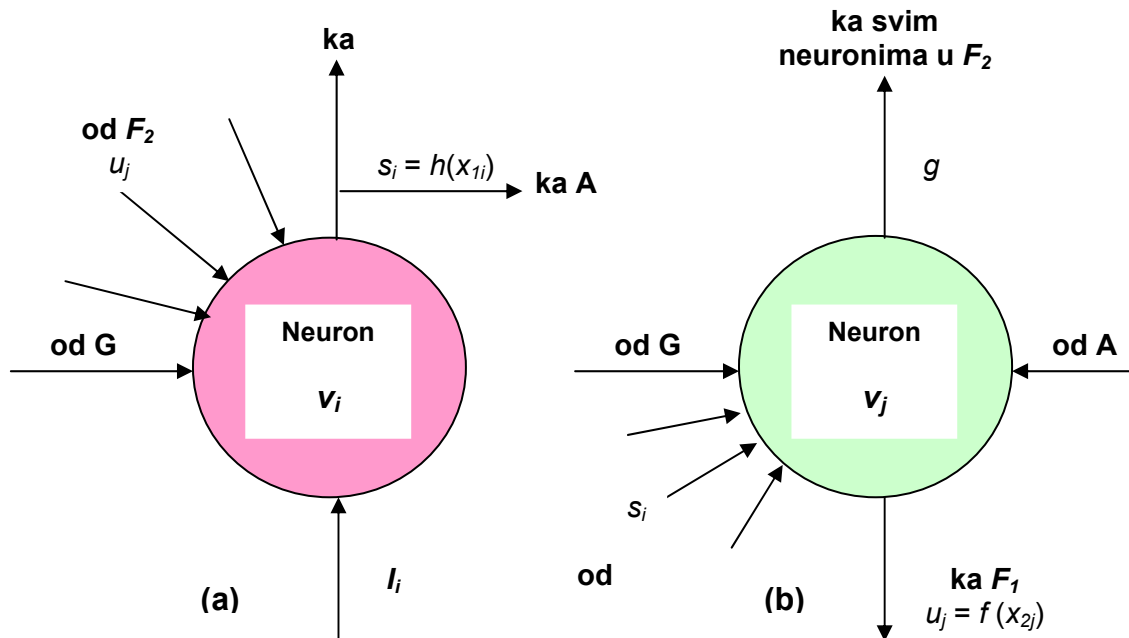
Sve aktivnosti koja su ostvarene preko neurona unutar mreže se zovu *vremenski kratkotrajna memorija* ili *STM (short-term memory)*, s obzirom da su realizovane samo u funkciji asocijacije sa svakim ulaznim oblikom, koji egzistira u kratkom periodu. Težinski odnosi međusobno povezanih neurona između F_1 i F_2 nivoa, su u asocijaciji sa informacijama koje ostaju deo mreže za neki sledeći period i zovu se *vremenski dugotrajna memorija* ili *LTM (long-term memory)*. Kao što se vidi na slici gore, ART-1 neuronska mreža ima dva podсистema: **podsystem za učenje** i **podsystem za utvrđivanje sličnosti**. *Podsystem za učenje* sadrži ulazni sloj i kompetitivni sloj neurona koji čine F_1 i F_2 nivo. Neuroni svakog nivoa su potpuno povezani sa neuronima drugog nivoa, a **podsystem za učenje ima zadatak da ostvari zaključak** da li je ulazni oblik vektora u asocijaciji sa prethodno memorisanim oblicima ili ne. To se realizuje preko veza ostvarenih pomoću *odozgo na dole LTM-a*, koji ima za cilj da, posle učenja, stabilizuje veze *odozdo na gore LTM-a*. *Podsystem za utvrđivanje sličnosti* je potreban ART-1 neuronskoj mreži da bi se odredila sličnost ulaznog vektora sa onim koji je poznat i uspešno prezentiran u okviru *LTM-a*, mada, ako je potrebno, za one oblike koji nisu naučeni, može da se formira i nova klasa uzoraka.



(c)

(d)

Postupak poređenja ulaznog oblika pomoću ART-1 neuronske mreže



Procesirajući elementi: (a) za F_1 (b) za F_2

Na slici pod a je prikazan ulazni oblik uzorka, predstavljen vektorom I , koji je doveden do neurona ulaznog sloja (F_1 nivo), pri čemu svaka komponenta tog ulaznog vektora odgovara po jednom od neurona. Vektor aktivacije X , koja je time prouzrokovana, određuje aktivaciju duž F_1 . Isti ulazni vektor pobuđuje i *podsystem za utvrđivanje sličnosti*, A, kao i deo ART-1 mreže koji vrši kontrolu, a zove se *regulator pojačanja*, G. Izlazni oblik uzorka iz F_1 , predstavljen vektorom S , prosleđuje inhibitorni signal, koji je takođe poslat do A. Mreža je tako strukturisana da ovaj inhibitorni signal potpuno otkazuje eksitacioni efekat signala koji dolazi od I , pa *podsystem za utvrđivanje sličnosti*, A, ostaje neaktivan. Treba uočiti da G dovodi jedan eksitacioni (pobuđivački) signal do F_1 , koji je primenjen na svaki neuron ulaznog sloja, a poznat je pod nazivom *nespecifičan signal*.

Pojava vektora aktivacije X unutar F_1 , rezultira izlaznim oblikom uzorka, S , koji se prosleđuje preko neuronskih veza do kompetitivnog sloja (F_2 nivo). Svaki neuron kompetitivnog sloja, F_2 , prima od ulaznog sloja, F_1 , ceo izlazni vektor S . Za neurone, koji čine F_2 , izračunavaju se ulazne vrednosti na uobičajen način, sumiranjem proizvoda ulaznih vrednosti aktivacije i težinskih odnosa. Kada su poznate sume za sve neurone kompetitivnog sloja, generiše se vektor aktivacije Y duž F_2 . Naime, nastupa takmičenje između neurona u kompetitivnom sloju, pri čemu "pobednički neuron uzima sve". Praktično, neuron koji ima najveću sumu težinskih odnosa se proglašava pobednikom. Njemu se dodeljuje novi aktivacioni nivo 1 , a svim ostalim neuronima 0 (slika pod b). U

slučaju da dva neurona imaju iste vrednosti suma težinskih odnosa, po konvenciji se bira neuron sa leve strane.

Na slici pod *b* se vidi da je usled vektora aktivacije \mathbf{Y} generisan izlazni vektor, \mathbf{U} , iz F_2 . Ovaj izlazni vektor se šalje, kao inhibitorni signal, ka G , pri čemu ako G primi bilo koji inhibitorni signal od F_2 , prestaje mu aktivnost. \mathbf{U} takođe postaje i drugi ulazni oblik za neurone u F_1 . Inače, izlazni vektor \mathbf{U} je transformisan pomoću *LTM*-a, preko *odozgo na dole* veza, od F_2 ka F_1 . Tako transformisan vektor se može označiti sa \mathbf{V} .

Treba naglasiti da sada postoje tri moguća ulaza ka F_1 , ali samo dva su aktuelna u svakom trenutku. Neuroni u F_1 , kao i u F_2 , su tako projektovani da postaju aktivni ako su dva od moguća tri ulaza aktivni. Ova karakteristika je poznata pod nazivom **2/3 pravilo** (dva od tri pravilo) i za ART mreže je od suštinske važnosti. Zbog 2/3 pravila, neuroni iz F_1 primaju signale od oba vektora \mathbf{I} i \mathbf{V} , tako da se u F_1 generiše novi vektor aktivacije \mathbf{X}^* (slika pod *b*). Kako je novi izlazni vektor iz F_1 , \mathbf{S}^* , različit od originalnog izlaznog vektora \mathbf{S} , inhibitorni signal ka A , ne zadugo, ukida eksitaciju koja dolazi od ulaznog vektora \mathbf{I} .

Na slici pod *c* se vidi da *podsystem za utvrđivanje sličnosti*, A , postaje aktivan u pogledu vektora koji dolaze od F_1 . Zatim, A šalje *reset signal* ka svim neuronima u kompetitivnom sloju. Neuroni u F_2 reaguju shodno njihovom trenutnom stanju. Ako su neaktivni, ne odgovaraju, a ako su aktivni postaju neaktivni i *ostaju u tom stanju u jednom dužem vremenskom periodu*. Ova "poduprta" inhibicija je neophodna zbog prevencije u smislu toga, da isti neuron ne bude pobednik u takmičenju koje nastupa u sledećem ciklusu poređenja. Kako \mathbf{Y} ne postoji za dugo, onda i inhibitorni signal, koji dolazi iz F_2 , a prostire se ka G , takođe nestaje.

Na slici pod *d* je prikazan postupak reaktivacije, koji se ostvaruje pomoću originalnog vektora aktivacije \mathbf{X} , ulaznog sloja F_1 , u okviru novog ciklusa, odnosno iteracije. Ovog puta novi vektor aktivacije \mathbf{Y}^* se pojavljuje u F_2 . Tako se iterativni postupak poređenja kontinuirano odvija dok se proces klasifikacije, odnosno identifikacije ne realizuje, čak i izlaženjem F_2 iz domena prethodno memorisanih uzoraka. Ako se ne realizuje uspešna klasifikacija (ili identifikacija), u kompetitivnom sloju se angažuju (aktiviraju) ponovo neuron(i) i počinje učenje novog ulaznog oblika.

Algoritam za brzo učenje ART-1 neuronske mreže

Korak 1: Određivanje veličine ulaznog sloja F_1 i kompetitivnog sloja F_2 . Neka je M broj neurona u sloju F_1 (za dati primer $M = 48$), a N broj neurona u sloju F_2 ($N = 4$).

Korak 2: Inicijalizacija konstantnih parametara:

$$A_1, C_1, D_1 \geq 0$$

$$\max\{D_1, 1\} < B_1 < D_1 + 1 \rightarrow A_1 = 1, B_1 = 1.5, C_1 = 5, D_1 = 0.9, L = 3$$

$$L > 1,$$

i određivanje veoma važnog parametra sličnosti $0 < \rho \leq 1 \rightarrow \rho = 0.9$.

Korak 3: Inicijalizacija *LTM*-a, prema sledećim izrazima za $i = 1, \dots, M$; $j = 1, \dots, N$:

$$\text{- odozgo na dole težinski odnosi: } z_{ij} > \frac{B_1 - 1}{D_1} = \frac{1.5 - 1}{0.9} = 0.55 \rightarrow z_{ij} = 0.2 + 0.55 = 0.75$$

$$\text{- odozdo na gore težinski odnosi: } 0 \leq z_{ji} \leq \frac{L}{L - 1 + M} = \frac{3}{3 - 1 + 48} = 0.06 \rightarrow z_{ji} = 0.05$$

Korak 4: Inicijalizacija aktivacije neurona u kompetitivnom sloju F_2 na vrednost 0.0, a u ulaznom sloju F_1 na vrednost:

$$x_{1i}(0) = \frac{-B_1}{1 + C_1} = \frac{-1.5}{1 + 5} = -0.25$$

Korak 5: Uvođenje ulaznog vektora \mathbf{I} u mrežu, preko ulaznog sloja F_1 , tako da će za 2D sliku objekta (prikazan na slici 3.14a) aktivacije neurona sada biti:

$$x_{1i} = \frac{I_i}{1 + A_1(I_i + B_1) + C_1} \rightarrow x_{1i}(I_i = 1) = 0.118 ; x_{1i}(I_i = 0) = 0$$

Dakle, za ulazni vektor $\mathbf{I}_1 = [0, 1, 0, 0, 1, 0, 0, 1, \dots, 1, 1, 0, 0]^T$, vektor aktivacije u F_1 je:

Korak 11: Određivanje novog izlaznog vektora za sloj F_1 , $\mathbf{S}^*=[0,1,\dots,1,0,0]^T$, po koraku 6.

Korak 12: Određivanje stepena podudarnosti (sličnosti) između ulaznog vektora i novo dobijenog oblika:

$$M = \frac{|S^*|}{|I|} = \frac{\sum_{i=1}^M S_i}{\sum_{i=1}^M I_i} \rightarrow \text{za dati objekat je } M = 1 > \rho = 0.9 \text{ (identifikovan je!!!)}$$

Korak 13: Poređenje M sa ρ . Ako je $M < \rho$, "markira" se pobednički neuron v_J , i tada on postaje neaktivan, što znači da su svi izlazi iz sloja F_2 "nulirani", pa je u ovom slučaju neophodno vratiti se na korak 5. Ako je $M \geq \rho$, ostvareno je stanje rezonance (nema reset signala od A – slika 4.18) i slede koraci 14 i 15.

Korak 14: Ažuriranje težinskih odnosa odozdo na gore, samo ka pobedničkom neuronu v_J u sloju F_2 :

$$z_{ji} = \begin{cases} \frac{L}{L-1+|S^*|} & \text{ako je neuron } v_i \text{ aktivan} \\ 0 & \text{ako je } v_i \text{ neaktivan} \end{cases}$$

Korak 15: Ažuriranje težinskih odnosa odozgo na dole, samo od pobedničkog neurona v_J ka svim neuronima u sloju F_1 :

$$z_{ij} = \begin{cases} 1 & \text{ako je neuron } v_i \text{ aktivan} \\ 0 & \text{ako je neuron } v_i \text{ neaktivan} \end{cases}$$

Korak 16: Resetovanje svih neaktivnih neurona u sloju F_2 , uvođenje novog ulaznog vektora i povratak na korak 5.

U ART-1 mrežu je ugrađena sposobnost klasifikacije ulaza u različite kategorije. Međutim, javljaju se i problemi ako skup ulaza nije dobro odabran, pa se može desiti da proces klasifikacije bude nestabilan. Inače, uspešnost procesa obučavanja mreže kod kompetitivnog učenja koji se primenjuje u ART-1 mreži, često zavisi i od izbora inicijalnih vrednosti težinskih odnosa, dok se izbor parametra učenja i broj iteracija najčešće određuje eksperimentalno. Tokom procesa obučavanja ART-1 mreže mogu se javiti brojni problemi, tako da performanse obučene mreže mogu biti narušene dovođenjem ulaznog oblika koji je potpuno drugačiji od svih prethodnih. Tada se primenjuje postupak podešavanja parametara, koji na žalost nije uvek produktivan. To je razlog za korigovanje algoritma učenja ART-1 mreže baš u pogledu fleksibilnosti izbora prilikom podešavanja parametara i konstanti. To je rezultiralo razvojem modifikovanog algoritma za brzo učenje ART-1 neuronske mreže (dat gore), koji se koristi za identifikaciju objekata snimljenih kamerom.