# 2. IDENTIFIKACIJA I APROKSIMACIJA

*Velika klasa problema optimizacije javlja se u zadacima identifikacije i aproksimacije. Ovi zadaci imaju mnogo zajedničkih svojstava i zato ih je teško razlikovati. Osnovni pristupi merenju kvaliteta identifikacije i aproksimacije u suštini su isti.*

*Pojam identifikacije široko se koristi u teoriji sistema upravljanja. Pod njim se podrazumeva proces izgradnje matematičog modela dinamičkog sistema na osnovu snimljenih podataka, obično na osnovu merenja ulaznih i izlaznih signala sistema. Primenjuje se kada posmatrani sistem nije dovoljno proučen i kada za njega ne postoji poznata matematička funkcija koja povezuje izlaze sa ulazom. Često se pod identifikacijom podrazumeva samo proces određivanja parametara izabranog matematičkog modela razmatranog dinamičkog sistem.*

*Tipičan zadatak u teoriji aproksimacije jeste određivanje kombinacije funkcija date klase tako da dobijena aproksimativna kriva na posmatranom intervalu odstupa što manje od neke zadate krive. Druga velika klasa zadataka koji se razmatraju u teoriji aproksimacije jeste određivanje aproksimativne krive na osnovu poznatih vrednosti promenljivih i funkcije u konačnom broju diskretnih tačaka.*

*Jedan moderan pristup rešavanju problema aproksimacije jesu veštačke neuronske mreže. Ta oblast savremene matematike i inženjerstva razvijena je na idejama i analogijama koje su potekle iz proučavanja nervnih sistema živih bića. Zbog prvobitne inspiracije preuzet je deo biološke teminologije, prevashodno radi jednostavnosti izražavanja i objašnjavanja osnovnog koncepta neuronske mreže. Pored primene u rešavanju problema aproksimacije, neuronske mreže se efikasno primenjuju i u rešavanju problema klasifikacije podataka, kategortizacije ili klasterovanja, predviđanja, optimizacije i upravljanja. U svim tim primerima se u nekoj fazi razvoja i/ili primene neuronske mreže javljaju i rešavaju problemi određivanja optimalne strukture i/ili optimalnih parametara mreže.*

*U ovoj glavi se postavljaju zadaci identifikacije i aproksimacije i izlažu osnovni pristupi njihovom rešavanju. Zatim se daju osnovni pojmovi iz neuronskih mreža i opisuje način njihove primene na rešavanju zadataka aprokasimacije.*

# 2. IDENTIFIKACIJA I APROKSIMACIJA

## 2.1. Problemi identifikacije

Identifikacija je oblast u teoriji sistema u kojoj se razmatraju problemi izgradnje pogodnih matematičih modela dinamičkih sistema, po pravilu objekta upravljanja, na osnovu eksperimentalnih podataka i analize tih podataka. Promenljive koje opisuju sistem najpre se podele na ulazne i izlazne promenljive. Merenjem vrednosti ulaznih i izlaznih promenljivih dobijaju se podaci na osnovu kojih treba identifikovati sistem, tj. razviti odgovarajući matematički model koji opisuje dinamiku sistema i odrediti njegove parametre.

*Primer* 2.1: Opisati neki konkretni sistem koji treba identifikovati. Naznačiti koje se njegove veličine mere. Prikazati oblike izmerenih signala. ♦

### 2.1.1. *Proces identifikacije*

U opštem slučaju proces identifikacije obuhvata rešavanje sledećih osnovnih zadataka:

- izbor klase matematičkih modela
- izbor klase ulaznih signala
- izbor kriterijuma za ocenu koliko model odgovara sistemu
- razvoj i primena algoritma za rešavanje.

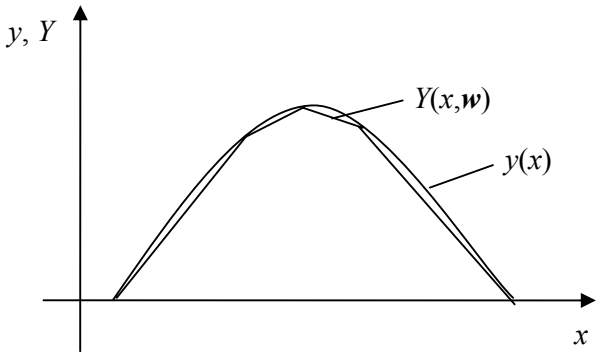
U užem smislu se pod identifikacijom podrazumeva rešavanje zadatka određivanja optimalnih parametara izabranog matematičkog modela. Ovde se pretpostavlja da su prethodno usvojeni klasa modela i kriterijum za ocenu valjanosti aproksimacije sistema modelom.

## 2.2. Problemi aproksimacije

U teoriji *aproksimacije* razvijaju se metode aproksimacije ili interpolacije neprekidne skalarne funkcije $y(\boldsymbol{x})$ vektorskog argumenta $\boldsymbol{x}$ pomoću aproksimativne funkcije $Y(\boldsymbol{w},\boldsymbol{x})$ koja ima fiksirani broj parametara $\boldsymbol{w}$ koji pripadaju nekom skupu $P$. Ovde su $\boldsymbol{x}$ i $\boldsymbol{w}$ realni vektori $\boldsymbol{x} = (x_1,\ldots,x_n)$ i $\boldsymbol{w} = (w_1,\ldots,w_m)$. Za izabranu funkciju $Y$ problem je naći vektor parametara $\boldsymbol{w}$ koji daje najbolju moguću aproksimaciju funkcije $y$ na posmatranom skupu.

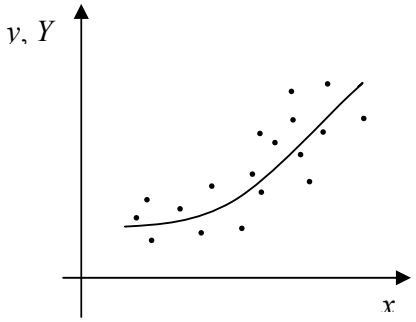
Postoje dva tipa zadataka aproksimacije. Prvi se odnosi na slučajeve kada zadatu krivu $y(\boldsymbol{x})$ treba aproksimirati funkcijama date klase ili njihovim kombinacijama, tako da dobijena aproksimativna kriva $Y(\boldsymbol{w},\boldsymbol{x})$ na posmatranom intervalu odstupa od zadate što je moguće manje. U klasičnoj teoriji aproksimacije kao aproksimativne funkcije se uobičajeno koriste kombinacije linearnih funkcija, sinusnih funkcija Čebiševljevih polinoma, funkcija oblika gustine normalne raspodele, itd.

Na slici 2.1. ilustrovana je aproksimacija jedne nelinearne krive pomoću nekoliko linearnih funkcija. Lako se uočava da kvalitet aproksimacije date nelinearne funkcije zavisi od broja aproksimirajućih linearnih fukcija.



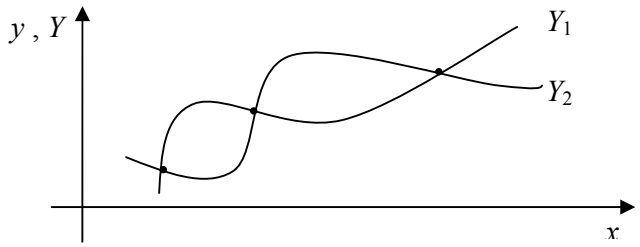

*Slika* 2.1. *Aproksimacija nelinearne krive linearnim odsečcima*

Drugi tip zadataka aproksimacije odnosi se na slučajeve kada su poznate realizacije posmatranih promenljivih u određenim diskretnim tačkama ($\boldsymbol{x}^i$, $y(\boldsymbol{x}^i)$) slika 2.2. Te se vrednosti obično dobijaju merenjem realnih veličina razmatranog sistema. Tada treba odrediti aproksimativnu krivu $Y(\boldsymbol{w},\boldsymbol{x})$ i njene parametre tako da zbir rastojanja aproksimiranih vrednosti $Y(\boldsymbol{w},\boldsymbol{x})$ od realizacija $y(\boldsymbol{x}^i)$ u posmatranim tačkama bude minimalan.




*Slika* 2.2. *Aproksimacija na osnovu uzoraka*

Treba imati na umu da je zadatak aproksimacije glatke funkcije na osnovu uzoraka, tj. na osnovu skupa diskretnih vrednosti, matematički slabo postavljen problem jer ne postoji dovoljno podataka da se rekonstruiše preslikavanje u oblastima za koje ne postoje podaci. Drugim rečima, može se konstruisati više međusobno različitih aproksimacija koje su po ukupnoj greški jednako dobre za podatke kojima se raspolaže. Razlike se naročito ispoljavaju u oblastima za koje se ne poznaju originalni podaci, slika 2.3. Pored toga, podaci po pravilu nisu apsolutno tačni već sadrže neku vrstu statističke greške. To stvara dodatne probleme u aproksimaciji i potrebu da se na početku postave neke apriorne pretpostavke (uslovi regularnosti) da bi zadatak aproksimacije bio dobro postavljen.




*Slika* 2.3. *Dve aproksimacije "jednake" ukupne greške*

Na zadacima identifikacije i aproksimacije lepo se uočavaju važnost i uloga izbora klase matematičkog modela i aproksimativnih funkcija kao i izbor kriterijuma kojim se ocenjuje valjanost rešenja.

Nema potrebe posebno objašnjavati koliko je važno izabrati aproksimativnu funkciju $Y$ koja može predstaviti $y$ najbolje što je moguće. Bilo bi skoro uzaludno tražiti optimalne vrednosti parametara aproksimativne krive $Y(\boldsymbol{w},\boldsymbol{x})$ u slučaju kada ona može da ostvari samo siromašnu predstavu funkcije $y(\boldsymbol{x})$, npr. ako se jako nelinerna funkcija pokušava da aproksimira linearnom funkcijom. Zato u rešavanju zadataka aproksimacije treba razlikovati sledeće glavne probleme:

1) Problem koju aproksimaciju koristiti, tj. koje klase funkcija $y(\boldsymbol{x})$ mogu da se efektivno aproksimiraju kojim aproksimativnim funkcijama $Y(\boldsymbol{w},\boldsymbol{x})$. Ovo je problem predstavljanja (reprezentacije).
2) Problem koji algoritam koristiti za nalaženje optimalnih vrednosti parametara $\boldsymbol{w}$ za dati izbor $Y$, uključujući u ovaj problem i izbor kriterijuma za ocenu valjanosti rešenja.

## 2.3. Merenje kvaliteta aproksimacije

Da bi se merio kvalitet aproksimacije u opštem slučaju se uvodi *funkcija rastojanja* $J$ koja utvrđuje rastojanje $J[Y(\boldsymbol{w},\boldsymbol{x}), y(\boldsymbol{x})]$ aproksimacije $Y(\boldsymbol{w},\boldsymbol{x})$ od $y(\boldsymbol{x})$. Rastojanje između $y(\boldsymbol{x})$ i aproksimativne krive $Y(\boldsymbol{w},\boldsymbol{x})$ naziva se *greška aproksimacije*. Rastojanje se obično uvodi nekom *normom*, npr. standardnom $L_2$ normom. Problem aproksimacije se može tada postaviti kao zadatak određivanja parametara aproksimativne krive kojim se minimizira rastojanje, odnosno greška aproksimacije. Taj zadatak ima sledeći oblik:

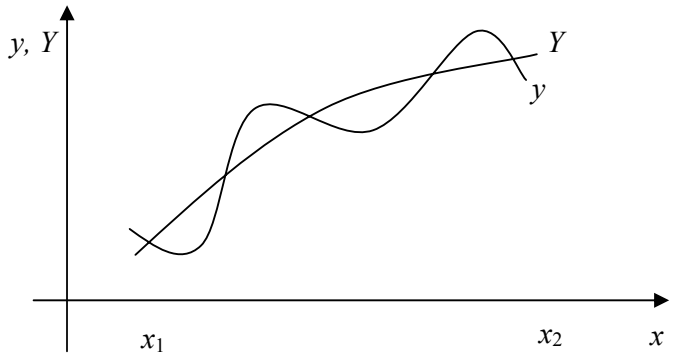
Treba odrediti parametre $\boldsymbol{w}^*$ aproksimativne funkcije $Y(\boldsymbol{w},\boldsymbol{x})$ tako da je

$$J[\, Y(\boldsymbol{w}^*,\boldsymbol{x}), y(\boldsymbol{x})] \le J[\, Y(\boldsymbol{w},\boldsymbol{x}), y(\boldsymbol{x})]$$

za svako $\boldsymbol{w}\in$P pri čemu je funkcija $y(\boldsymbol{x})$ neprekidna na posmatranom skupu promenljive $\boldsymbol{x}$, a aproksimativna funkcija $Y(\boldsymbol{w},\boldsymbol{x})$ neprekidna i po $\boldsymbol{w}\in P$ i po $\boldsymbol{x}$.

Rešenje ovog problema, ako postoji, naziva se *najbolja aproksimacija*. Od klase funkcija kojoj pripada $Y(\boldsymbol{w},\boldsymbol{x})$ zavisi da li postoji najbolja aproksimacija $Y(\boldsymbol{w},\boldsymbol{x})$.

Na primeru funkcije jedne promenljive $y(x)$ koja se aproksimira funkcijom $Y(\boldsymbol{w},x)$ na intervalu $[x_1,x_2]$, slika 2.4., ilustrovaćemo nekoliko mogućih načina za računanje kriterijuma za ocenu valjanosti aproksimacije na osnovu greške. Uopštavanje za funkcije više promenljivih je po pravilu jednostavno. Treba primetiti da parametri aproksimativne funkcije $Y(\boldsymbol{w},x)$ u opšem slučaju zavise od izabranog kriterijuma




*Slika* 2.4. *Aproksimacija glatke krive*

1. Jedan pristup je računanje apsolutne vrednosti površine između originalne i aproksimativne krive

$$J_1 = \int_{x_1}^{x_2} |y(x) - Y(\boldsymbol{w}, x)| \, dx .$$

2. Sledeći pristup je računanje kvadrata odstupanja na intervalu ($x_1$, $x_2$)

$$J_2 = \int_{x_1}^{x_2} [y(x) - Y(\boldsymbol{w}, x)]^2 \, dx .$$

3. Ako se želi da se greški u izvesnim delovima intervala $[x_1,x_2]$ da relativno veći značaj, onda se prethodna dva slučaja mogu modifikovati uvođenjem težinske funkcije $\omega(x)$, npr.

$$J_3 = \int_{x_1}^{x_2} \omega(x) [y(x) - Y(\boldsymbol{w}, x)]^2 \, dx .$$

4. Često se kao kriterijum postavlja maksimalno odstupanje u posmatranom intervalu

$$J_4 = \max_{[x_1,x_2]} | y(x) - Y(\boldsymbol{w},x) | .$$

Navedeni kriterijumi su dati za prvi tip zadataka aproksimacije kada je zadata neprekidna funkcija $y(x)$. Za drugi tip problema aproksimacije kada su date vrednosti funkcije $y(x)$ samo u određenim tačkama $x^i$ analogni kriterijumi su:

$$J_1 = \sum_{i=1}^{I} \left| y(x^i) - Y(\boldsymbol{w}, x^i) \right|$$

$$J_2 = \sum_{i=1}^{I} \left[ y(x^i) - Y(\boldsymbol{w}, x^i) \right]^2$$

$$J_3 = \sum_{i=1}^{I} \omega_i \left[ y(x^i) - Y(\boldsymbol{w}, x^i) \right]^2$$

$$J_4 = \max_i \left| y(x^i) - Y(\boldsymbol{w}, x^i) \right|$$

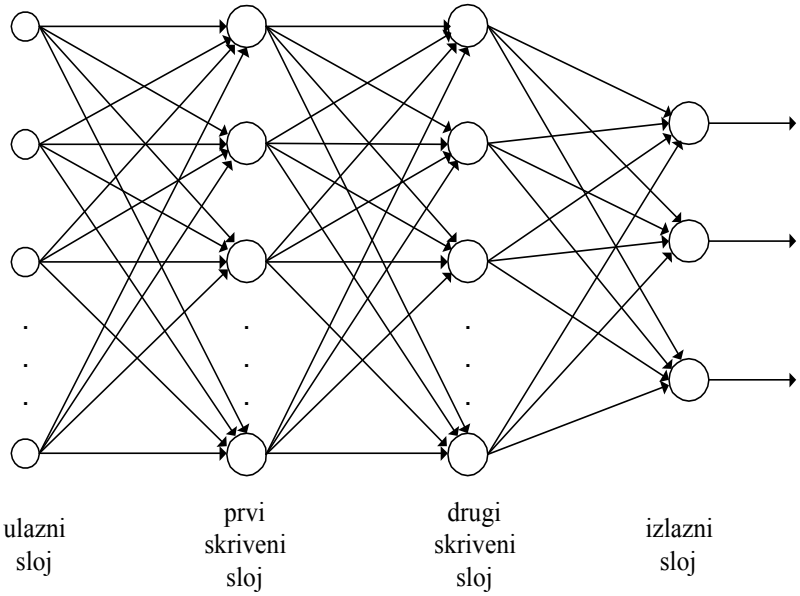
**Primer 2.2:** Dvostruka nelinearna regresija. Dati podaci o tražnji i cenama ($y^i$, $p_1^i$, $p_2^i$). Odrediti $a,b,c$.

$y = a p_1^b p_2^c$. ♦

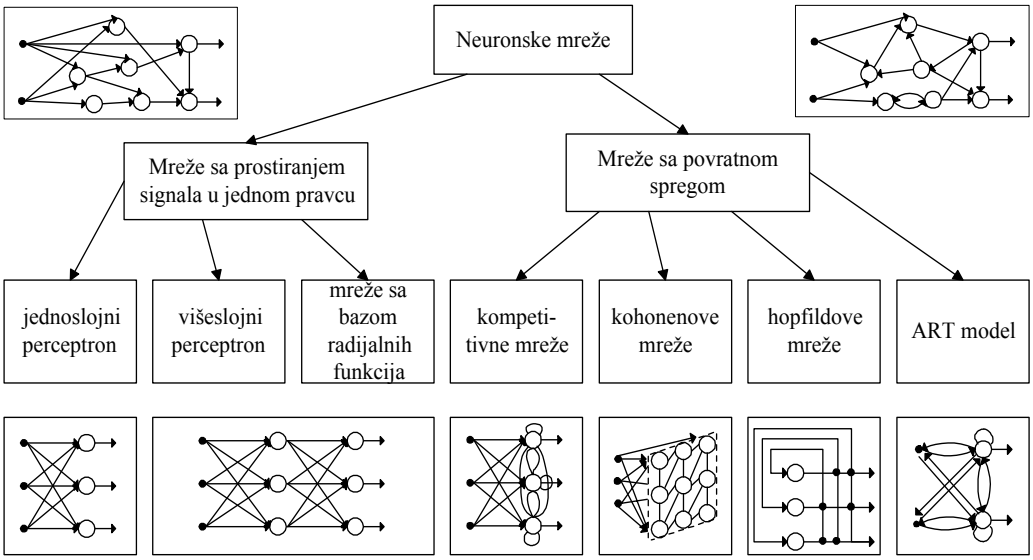
## 2.4. Neuronske mreže

Jedan pristup rešavanju problema aproksimacije zasniva se na korišćenju *veštačkih neuronskih mreža* koje se u matematičkoj i tehničkoj literaturi skraćeno nazivaju neuronske mreže, ali se koriste i drugi nazivi kao što su: mreže za paralelnu obradu, mreže distribuirane paralelne obrade, konekcionističke mreže, paralelno procesiranje i slično. Zbog analogije sa realnim biološkim sistemima preuzet je deo biološke terminologije koji pojednostavljuje objašnjenje karakterističnih pojmova i pojava u neuronskim mrežama. Međutim, treba imati na umu da obrade signala u čvorovima veštačkih neuronskih mreža i način njihovog povezivanja mogu ali u opštem slučaju ne moraju da imitiraju procese u realnim neuronima i načine njihovih povezivanja u biološkim sistemima. Zato je potrebna opreznost u korišćenju terminologije, tumačenjima i analogijama veštačkih i stvarnih bioloških neuronskih mreža. U daljem tekstu pojmovi koji su preuzeti iz biologije koristiće se isključivo u onom smislu kako su definisani u oblasti veštačkih neuronskih mreža. Da bi istakli razliku između veštačkih i bioloških neuronskih mreža, jedan broj domaćih autora za ove prve koristi termin *neuralne* mreže.

Neuronska mreža se može posmatrati kao jedan ulazno izlazni sistem, slika 2.5. Za ovaj sistem je karakteristično da njegova struktura odgovara mreži međusobno povezanih elementarnih čvorova, *neurona*. Neuroni su delovi mreže koji vrše elementarne obrade signala koji u njih ulaze. Na osnovu tipa obrade koji se obavlja u čvoru (tj. na osnovu vrste neurona) i na osnovu međusobne povezanosti čvorova (neurona) moguće su različite klasifikacije neuronskih mreža. Jedna takva klasifikacija prikazana je na slici 2.6.




*Slika* 2.5. *Graf višeslojnog perceptrona sa dva skrivena sloja*

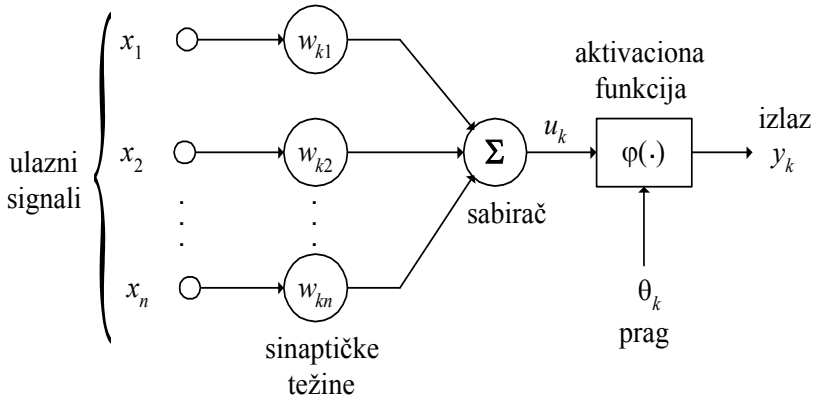



*Slika* 2.6. *Jedna klasifikacija neuronskih mreža*

Ovde se pažnja ograničava na mreže u kojima je moguć tok signala samo u jednom pravcu (*feedforward*) i ne razmatraju se mreže sa povratnom spregom (*feedback*). Mreža sa prostiranjem signala u jednom smeru naziva se *perceptron*. Grafovski prikaz perceptrona dat je na slici 2.5. Čvorovi mreže mogu da se grupišu u slojeve tako da izeđu čvorova istog sloja ne postoje veze; drugim rečima u takvim mrežama ostvaruju se veze samo između čvorova susednih slojeva kao što je prikazano na slici 2.5. Postoje ulazni, izlazni i skriveni slojevi.

### 2.4.1. *Model neurona*

Neuron je osnovni elemenat neuronske mreže. To je jedinica koja obrađuje informacije koje dobija na ulazu i kao rezultat daje jedan izlaz. Na slici 2.7. prikazan je osnovni model neurona.




*Slika* 2.7. *Model neurona*

Za opis neurona bitna su sledeća tri elementa:

1. Skup povezujućih linija od kojih je svaka karakterisana svojom težinom ili jačinom. Ove linije se nazivaju *sinapse* a

odgovarajući težinski koeficijenti su *sinaptičke težine* ili težine. Signal $x_j$ na ulazu sinapse $j$ povezan je sa neuronom $k$ na taj način što je njegova vrednost pomnožena težinom $w_{kj}$. Težina $w_{kj}$ može biti pozitivna ili negativna: u prvom slučaju se kaže da je sinapsa *pobuđujuća* (ekscitatorna) a u drugom da je *kočeća* (inhibitorna).

2. *Sabirač* (sumator) je elemenat koji sabira ulazne signale koji su prethodno otežani sinapsama neurona tako da se na izlazu sabirača dobija linearna kombinacija ulaza. Sinaptičke težine i sabirač formiraju linearni kombinator.
3. *Aktivaciona funkcija* daje izlaz iz neurona na osnovu takozvanog *neto ulaza* koji se dobija na izlazu sabirača pri čemu je dodatno izvršena modifikacija ulaza *pragom* koji ima efekat snižavanja vrednosti ulaza. Aktivaciona funkcija po pravilu ograničava vrednost izlaznog signala iz neurona na neke konačne vrednosti. Tipično je da se normalizovani opseg amplituda izlaza piše kao zatvoreni interval [0, 1] ili [-1, 1].

Model prikazan na slici 2.7. obuhvata i eksterno zadati *prag* ili *pomeraj* koji ima efekat snižavanja neto ulaza aktivacione funkcije. Neto ulaz aktivacione funkcije može biti i povećan korišćenjem *pristrastnosti* $b_k$ umesto praga; u stvari, pristrasnost se može posmatrati kao negativan prag.

Matematički, neuron $k$ se opisuje sledećim parom jednačina

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

$$y_k = \varphi(u_k - \theta_k)$$

gde su $x_1, x_2, \ldots, x_n$ ulazni signali, $w_{k1}, w_{k2}, \ldots, w_{kn}$ su sinaptičke težine za neuron $k$ , $u_k$ je linearna kombinacija ulaza, $\theta_k$ je prag, $\varphi(\cdot)$ je aktivaciona funkcija, i $y_k$ je izlazni signal neurona. Korišćenje praga $\theta_k$ ima efekat primenjivanja afine transformacije na izlaz $u_k$ iz sabirača u modelu sa slike 2.7, tj.

$$v_k = u_k - \theta_k .$$

Zavisno od toga da li je prag $\theta_k$ pozitivan ili negativan odnos između efektivnog internog nivoa aktivacije $v_k$ ili *aktivacionog potencijala* neurona $k$ i izlaza $u_k$ iz sabirača (linearnog kombinatora ulaza) modifikuje se na način prikazan na slici 2.8. Treba uočiti da kao rezultat afine transformacije grafik $v_k$ u odnosu na $u_k$ ne prolazi kroz koordinatni početak.

Prag $\theta_k$ je eksterni parametar neurona $k$ i treba računati sa njegovim prisustvom iako se model neurona može uopštiti uvođenjem dodatnog jediničnog ulaza čija je sinaptička težina jednaka vrednosti praga, odnosno pristrasnosti, na sledeći način

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

$$y_k = \varphi(v_k)$$

$$x_0 = -1$$

$$w_{k0} = \theta_k$$

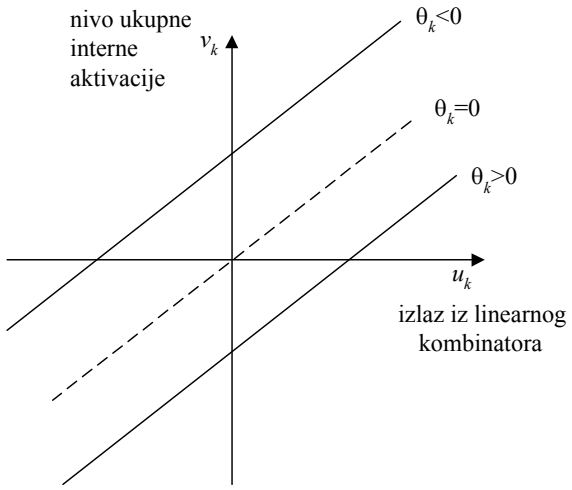
ili

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

$$y_k = \varphi(v_k)$$

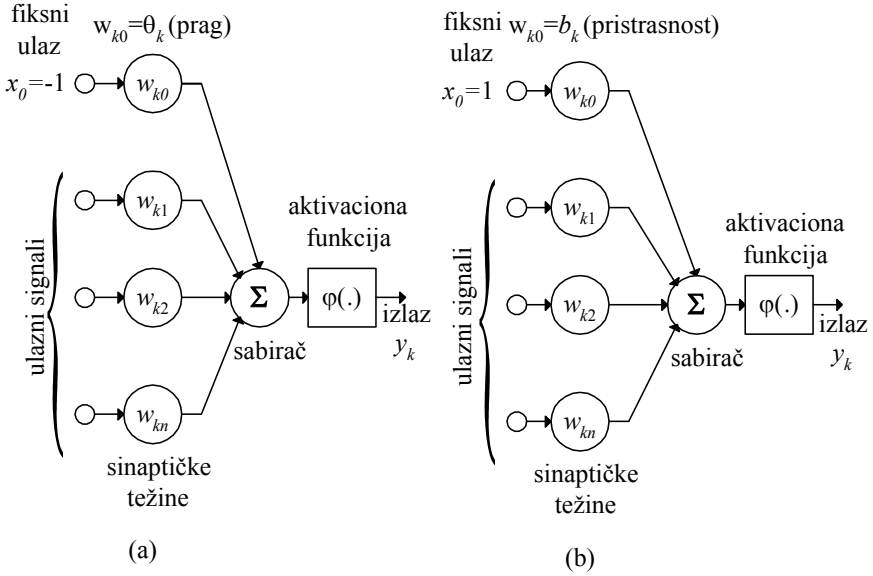
$$x_0 = 1$$

$$w_{k0} = b_k..$$




*Slika* 2.8. *Uticaj praga na interni nivo aktivacije*

Navedne modifikacije su ilustrovane slikom 2.9.




*Slika* 2.9. *Modifikacija polaznog modela neurona*

### 2.4.2. *Tipovi aktivacione funkcije*

Aktivaciona funkcija φ(.) definiše izlaz neurona u zavisnosti od nivoa aktivacije na njegovom ulazu. Navešćemo tri osnovna tipa aktivacione funkcije.

1. *Odskočna funkcija* (slika 2.10). Ovaj tip aktivacione funkcije opisan je sa

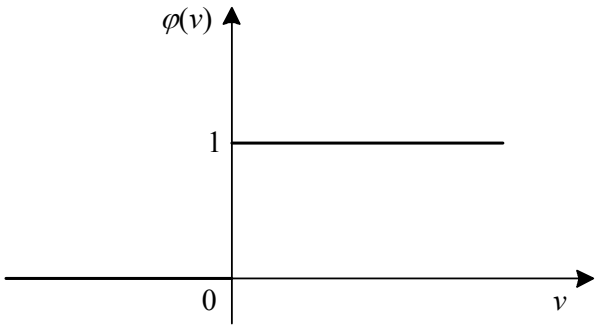
$$\varphi(v)=\begin{cases}0, & v<0\\ 1, & v\geq 0\end{cases}$$

odnosno, izlaz iz neurona *k* koji koristi takvu funkciju je

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

gde je $v_k$ interni nivo aktivacije

$$v_k = \sum_{j=1}^{n} w_{kj} x_j - \theta_k$$




*Slika* 2.10. *Odskočna funkcija*

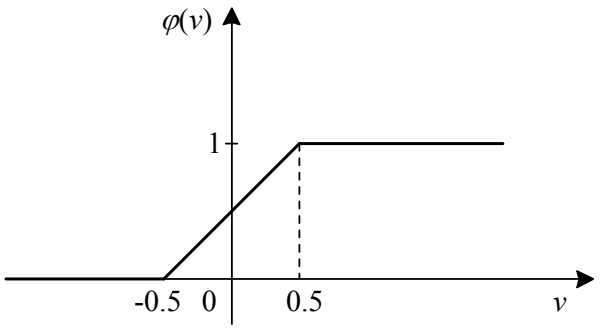
Ovakav neuron se naziva Mekkaloč-Pitov (McCulloch, Pitt) model kao znak poštovanja njihovog pionirskog rada. Izlaz iz modela je 1 ako je interni nivo aktivacije nenegativan a u suprotnom je 0. Takvo ponašanje opisuje osobinu *sve ili ništa* koju poseduje ovaj model.

2. *U delovima linearna funkcija* (slika 2.11). Ovaj tip aktivacione funkcije opisan je sa

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

pri čemu je pretpostavljeno da je faktor pojačanja u linearnom delu jednak jedinici što u opštem slučaju ne mora da bude. Ovaj oblik aktivacione krive može se posmatrati kao aproksimacija nelinearnog pojačavača. Sledeća dva slučaja mogu da se posmatraju kao specijalni oblici u delovima linearne funkcije:

a) Linearni kombinator nastaje ako se oblast linearne transformacije održava bez ulaska u zasićenje.
b) U delovima linearna funkcija se redukuje u odskočnu funkciju ako se faktor pojačanja učini beskonačno velikim u linearnoj oblasti.




*Slika* 2.11. *U delovima linearna funkcija*

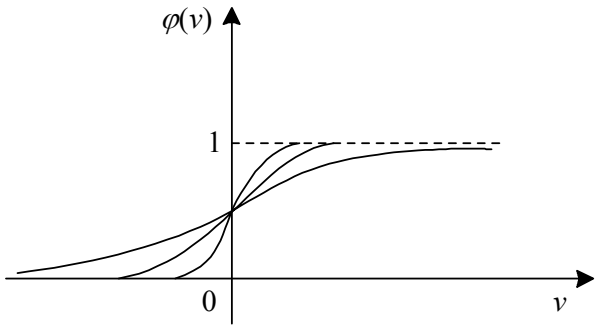
3. *Sigmoidna funkcija* (slika 2.12). Sigmoidna funkcija je oblik aktivacione funkcije koji se najviše koristi u neuronskim mrežema. Ona se definiše kao striktno rastuća funkcija koja ima osobine glatkosti i asimptote. Primer sigmoidne funkcije je logistička funkcija definisana sa

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

gde je $a$ parametar nagiba sigmoidne funkcije. Menjajući vrednost parametra $a$ dobijaju se različiti oblici, slika 2.12. Nagib funkcije za $v$=0 je $a$/4 a ako $a$ teži beskonačnosti, onda sigmoidna funkcija prelazi u odskočnu. Međutim, odskočna funkcija je prekidna i ima vrednost 0 ili 1, a sigmoidna funkcija je neprekidna i uzima sve vrednosti između 0 i 1. Sigmoidna funkcija je diferencijabilna, što je veoma važno za primenu u neuronskim mrežama, dok odskočna funkcija nije.

Pored navedenih koriste se i drugi tipovi aktivacionih funkcija. Nekada je poželjno da se se umesto između 0 i 1 izlaz iz aktivacione funkcije menja između –1 i 1. Tada se odskočna funkcija redefiniše na sledeći način

$$\varphi(v) = \begin{cases} -1, & v < 0 \\ 0, & v = 0 \\ 1, & v > 0 \end{cases}$$




*Slika* 2.12. *Sigmoidna funkcija*

što je takozvana *signum funkcija*. Kao sigmoida koristi se funkcija tangensa hiperboličnog

$$\varphi(v) = \tanh\frac{v}{2} = \frac{1-e^{-v}}{1+e^{-v}}.$$

### 2.4.3. *Rešavanje problema aproksimacije*

Vratimo se sada problemu aproksimacije za čije se rešavanje koriste višeslojni perceptroni. U skladu sa ranijim razmatranjima, u rešavanju konkretnog problema najpre bi trebalo izabrati arhitekturu mreže i tip neurona a potom odrediti parametre mreže. Ovde se daje nekoliko primera aproksimativnih funkcija $Y(\boldsymbol{w},\boldsymbol{x}) : R^n - R$ koje se mogu predstaviti višeslojnim mrežama.

1. Najjednostavnija struktura za aproksimaciju obuhvata *linearnu kombinaciju ulaza*

$$Y(\boldsymbol{w},\boldsymbol{x}) = \boldsymbol{w}\,\boldsymbol{x} = \sum_{j=1}^{n} w_j x_j$$

gde $\boldsymbol{w}$ i $\boldsymbol{x}$ predstavljaju $n$-dimenzione vektore; ovom slučaju odgovara mreža bez skrivenih slojeva.

2. Klasična shema aproksimacije je linearna *kombinacija funkcija pogodne baze* $\{\phi_i\}_{i=1}^{m}$ gde su funkcije $\phi_i$ funkcije originalnih ulaza $\boldsymbol{x}$ tako da je

$$Y(\boldsymbol{w},\boldsymbol{x}) = \sum_{i=1}^{m} w_i \phi_i(\boldsymbol{x})$$

Ova shema odgovara mreži sa jednim skrivenim slojem. Splajn interpolacija i mnoge druge aproksimativne sheme kao što je razvoj u red ortogonalnih polinoma obuhvaćeni su ovom predstavom. Kada su funkcije $\phi_i$ proizvodi i stepeni ulaznih komponenata, funkcija $Y$ je polinom.

3. Za neuronske mreže je posebno zanimljiva shema *ugneždenih sigmoida* koja se koristi zajedno sa algoritmom prostiranja greške unazad, koji će kasnije biti opisan. Ova shema se piše na sledeći način

$$Y(\boldsymbol{w},\boldsymbol{x}) = \sigma(\Sigma_n w_n \sigma(\Sigma_i v_i \sigma(\ldots \sigma(\Sigma_j u_j x_j) \ldots)))$$

gde je $\sigma$ sigmoidna funkcija. Ona odgovara višeslojnoj mreži čvorova koji sumiraju svoje ulaze pomnožene "težinama" $W = \{\boldsymbol{w}_n, \boldsymbol{v}_i, \boldsymbol{u}_j, \ldots\}$ a onda izvršavaju sigmoidnu transformaciju ove sume. Ovakva shema (ugneždene nelinearne funkcije) nije korišćena u klasičnoj teoriji aproksimacije neprekidnih funkcija. Motivacija za korišćenje ugneždenih funkcija potiče od sledećeg razmatranja. Ako je $\sigma$ odskočna funkcija, tada

$$Y(\boldsymbol{w},\boldsymbol{x}) = \sigma(\Sigma_n w_n \sigma(\Sigma_j u_j x_j))$$

može predstaviti sve Bulove funkcije. (Bilo koje preslikavanje $S : \{0,1\}^n \rightarrow \{0,1\}$ može da se napiše kao disjunkcija sastavljena od konjukcija, /disjunktivna normalna forma/, koja pomoću odskočnih funkcija dobija gornji izraz, pri čemu je dozvoljeno korišćenje fiktivnih ulaza i pristrasnosti). Mreže ovog tipa, sa jednim skrivenim slojem čvorova, mogu proizvoljno dobro da aproksimiraju bilo koju neprekidnu funkciju više promenljivih. Isti rezultat je dokazan i za mreže sa dva skrivena sloja.

Kao što je već rečeno, u rešavanju problema aproksimacije najpre treba izabrati klasu ili tip aproksimativne funkcije. U skladu sa poslednjim stavom, za aproksimaciju neprekidnih funkcija po pravilu se koriste perceptroni sa jednim ili dva skrivena sloja čvorova koji obavljaju sigmoidnu transformaciju. Dalje razmatranje se ograničava upravo na tu vrstu neuronskih mreža.

Ne postoji konačan odgovor na pitanje koliko čvorova treba koristiti u kojem sloju, odnosno koja je struktura mreže optimalna. To se rešava na osnovu karakteristika konkretnog zadatka aproksimacije, iskustva i eksperimenata.

Dakle, kada se izabere vrsta mreže (npr. višeslojni perceptron sa sigmoidnim aktivacionim funkcijama), broj slojeva (jedan ili dva) i broj čvorova u slojevima, tj. kada se odredi struktura mreže, postavlja se zadatak određivanja parametara mreže, sinaptičkih težina, kojima se minimizira ukupna greška aproksimacije. Kao kriterijum se najčešće koristi ukupna srednja kvadratna greška, ranije definisan kriterijum $J_2$.

### 2.4.4. *Proces obučavanja*

Za određivanje parametara mreže koristi se iterativni postupak koji se naziva se *proces obučavanja* ili *proces učenja* neuronske mreže. Proces obučavanja je proces rešavanja optimizacionog zadatka /problem aproksimacije iz prethodnog odeljka/ primenom pogodnog algoritma. U opštem slučaju, zadatak koji treba rešiti pripada problemima globalne optimizacije.

Razlikuju se dva načina obučavanja mreže: *obučavanje sa nadgledanjem* ili obučavanje sa učiteljem i *obučavanje bez nadgledanja* ili obučavanje bez učitelja. U prvom slučaju se pretpostavlja da postoji skup uzoraka na kojima se mreža obučava; u drugom slučaju se parametri mreže podešavaju u toku rada. Sposobnost nekih mrežnih struktura da im se u toku primene parametri mogu dinamički da podešavaju bez skupa uzoraka za obučavanje učinili su neuronske mreže jako atraktivnim u oblasti izučavanja veštačke inteligencije odakle je potekao termin da mreže mogu da uče i da se samoobučavaju.

Za rešavanje razmatranog problema aproksimacije ovde su od interesa procesi obučavanja sa nadgledanjem. Uzorak je definisan parom ($\boldsymbol{x}^i$, $\boldsymbol{y}^i$) gde je $\boldsymbol{x}^i$ vrednost ulaza a $\boldsymbol{y}^i = \boldsymbol{y}(\boldsymbol{x}^i)$ odgovarajuća vrednost izlaza, pri čemu su i $\boldsymbol{x}$ i $\boldsymbol{y}$ u opštem slučaju vektori. Pretpostavlja se da postoji skup za obučavanje od $n$ parova ($\boldsymbol{x}^i$, $\boldsymbol{y}^i$). Osnovni postupak procesa učenja sa nadgledanjem sastoji se u sledećem:

1. Zadaju se početni parametri (sinaptičke težine) mreže. Obično se to radi tako što se vrednosti ovih parametra postave na slučajan način ali tako da nijedan ne bude jednak nuli.
2. Na ulaz neuronske mreže dovode se ulazni signali iz skupa za obučavanje i računaju odgovarajući izlazi iz mreže.
3. Izračuna se ukupna greška na izlazu i zatim poboljša vrednost parametara.

   Koraci 2. i 3. ponavljaju se dok se ne stabilišu vrednosti težinskih koeficijenata.

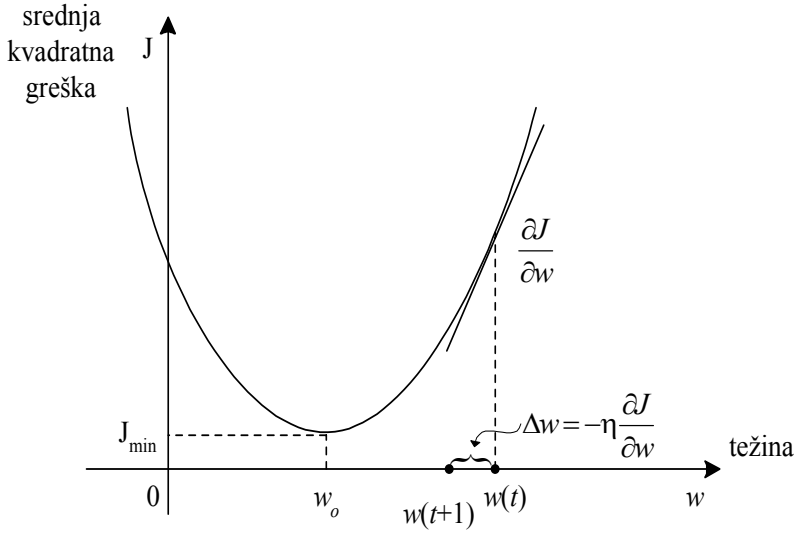
U navedenom opštem postupku za obučavanje neuronske mreže ključni deo je poboljšanje vrednosti parametara mreže. U tu svrhu se koriste različiti algoritmi. Za određivanje težina višeslojnog perceptrona koristi se *algoritam sa prostiranjem (širenjem) unazad* (*backpropagation algorithm*). Ovde će se dati samo osnovne ideje ovog algoritma.

### 2.4.5. *Algoritam sa prostiranjem unazad*

Kao kriterijum za ocenu valjanosti aproksimacije koristi se srednja kvadratna greška koja se računa po svim izlaznim čvorovima i po svim uzorcima za obučavanje. Traženi parametri mreže određuju se modifikacijom poznatog algoritma najmanjeg kvadrata. Modifikacija je neophodna zbog strukture mreže, odnosno zbog karakterističnog ugneždenog oblika aproksimativne funkcije. Algoritam najmanjeg kvadrata spada u grupe gradijentnih algoritama za čiju primenu kriterijumska funkcija

treba da je diferencijabilna. U slučaju kada se koristi sigmoidna aktivaciona funkcija taj uslov je ispunjen.

U osnovi gradijentnih algoritama je princip korišćenja najveće strmine ili najbržeg spusta. U numeričkom postupku iterativnog tipa polazi se od neke zadate početne vrednosti nepoznate težine i progresivno kreće ka optimalnoj vrednosti pri čemu se pravac kretanja određuje na osnovu informacije o pravcu opadanja kriterijumske funkcije. Za najjednostavniji slučaj kada bi se koristila linearna aproksimacija, zavisnost greške od težine je kvadratna funkcija pa se može primeniti postupak ilustrovan na slici 2.13.




*Slika* 2.13. *Ilustracija metode najmanjeg kvadrata*

Kada u nekoj iteraciji $t$ posmatrana težina ima vrednost $w_k(t)$ radi približavanja optimalnoj vrednosti $w_{ko}$ u sledećoj iteraciji vrednost težine treba da se promeni u pravcu suprotnom od pravca u kojem kriterijumska funkcija raste, slika #.11, tj. u pravcu suprotnom od pravca gradijenta kriterijumske funkcije u toj tački

$$w_k\,(t+1) = w_{ij}(t) + \Delta w_k(t)$$

gde je

$$\Delta w_k(t) = -\,\eta\,\nabla_{w_k} J(t)$$

pri čemu je $\eta$ pozitivna konstanta koja se naziva faktor pojačanja ili parametar brzine učenja.

U algoritmu sa prostiranjem unazad koristi se opisani gradijentni postupak tako što se u iterativnom postupku najpre primeni na težine čvorova u izlaznom sloju. Potom se pređe na sloj koji prethodi izlaznom i u njemu izvrši podešavanje težina. Postupak se nastavlja u pravcu od izlaza ka ulazu, odnosno u pravcu koji je suprotan pravcu prostiranja signala. Pokazano je da ovaj postupak konvergira.

Algoritam sa prostiranjem unazad nije jednostavno precizno izložiti jer zahteva veliki broj indeksa. Njegovi osnovni koraci su sledeći:

1. Inicijalizacija. Sve težine i pragove (pomeraje) staviti na male slučajne vrednosti,
2. Dati ulaze i željene izlaze. Na ulaz dovoditi ulazne vektore $\boldsymbol{x}^i$ i za svaki od njih specificirati željene izlaze $\boldsymbol{y}^i$. Skup za obučavanje može u svakom novom pokušaju biti nov, a može biti uvek jedan isti dok se ne stabilizuju parametri mreže.

3. Izračunati stvarne izlaze. Koristeći kao aktivacione funkcije za svaki neuron odgovarajuću sigmoidnu funkciju, na osnovu datih formula i strukture mreže izračunati izlaze iz mreže $\boldsymbol{Y}^i = \boldsymbol{Y}(\boldsymbol{w}, \boldsymbol{x}^i)$.
4. Adaptirati parametre. Koristiti rekurzivni algoritam koji počinje od izlaznih čvorova i radi unazad prema prvom skrivenom sloju. Podešavanje težina vrši se po sledećem opštem obrascu

   $w_{ij}(t+1) = w_{ij}(t) + \eta\, \delta_j\, x_i'$

   U ovoj jednačini $w_{ij}(t)$ je težina iz skrivenog sloja $i$ ili od ulaza (za prvi skriveni sloj) do čvora $j$ u iteraciji (trenutku) $t$, $x_i'$ je ili izlaz čvora $i$ ili ulaz, $\eta$ je faktor pojačanja, a $\delta_j$ je izraz za grešku za čvor $j$. Ako je $j$ izlazni čvor, onda je (za sigmoidnu aktivacionu funkciju)

   $\delta_j = Y_j\,(1 - Y_j)(y_j - Y_j)$

   gde je $y_j$ željeni (zadati) izlaz čvora $j$, a $Y_j$ je stvarni izračunati izlaz mreže.

   Ako je čvor $j$ unutrašnji skriveni čvor, onda je

   $\delta_j = x_j'\,(1 - x_j')\,\Sigma_k\, \delta_k w_{jk}$

   pri čemu $k$ ide preko svih čvorova u sloju koji prethodi čvoru $j$. Unutrašnji pragovi se adaptiraju na sličan način jer se modeliraju kao težine na pomoćnim vezama sa konstantnim vrednostima ulaza. Konvergencija algoritma je nekada brža ako se u izraz za adaptaciju težina doda faktor izglađivanja

   $w_{ij}(t+1) = w_{ij}(t) + \eta\, \delta_j\, x_i' + a(w_{ij}(t) - w_{ij}(t-1))$

   gde je $0 < a < 1$.
5. Vratiti se na korak 2.

Eksperimentalni rezutati pokazuju da se algoritam sa prostiranjem unazad efikasno primenjuje u podešavanju parametara višeslojnog perceptrona sa aktivacionim funkcijama tipa sigmoide.

---

*Primer* 2.3. Navesti primer primene neuronske mreže i prikazati rezultate. Koristiti postojeći softver. ♦

---

**Literatura**


[1] Hammerstrom D., (1993), "Neural networks at work", *IEEE Spectrum,* June1993, pp. 26-32.

[2] Hammerstrom D., (1993), "Working with neural networks", *IEEE Spectrum,* July1993, pp. 43-56.

[3] Jain A.K., Mao J., Mohiuddin K.M., (1996), "Artificial neural networks: A tutorial", *IEEE computer*, March 1996, pp. 31-44.

[4] Pal S.K., Srimani P. K., (1996), "Neurocomputing - Motivation, models and hibridization", *IEEE computer*, March 1996, pp. 24-28.

[5] Setiono R., Liu H., (1996), "Symbolic representation of neural networks", *IEEE computer*, March 1996, pp. 71-77.

[6] Shang Y., Wah B.W., (1996), "Global optimization for neural network training", *IEEE computer*, March 1996, pp. 45-54.

[7] Šerbedžija N.B. (1996), "Simulating artificial neural networks on paralel architectures", *IEEE computer*, March 1996, pp. 56-63.

[8] Tan C.L., Quah T.S., Teh H.H., (1996), " An artificial neural network that models human decision making", *IEEE computer*, March 1996, pp. 64-70.

[9] Wawrzynek J., Asanovi} K., Kingsbury B., Johnson D., Beck J., Morgan N., (1996), "Spert-II: A vector microprocessor system", *IEEE computer*, March 1996, pp. 79-87.