

SVEUČILIŠTE U ZAGREBU  
PRIRODOSLOVNO – MATEMATIČKI FAKULTET  
FIZIČKI ODSJEK

SMJER: profesor fizike i informatike

**Kenan Mačkić**

Diplomski rad

Primjena neuronskih mreža u fizici i  
informatici

Voditelj diplomskog rada: Prof. dr. sc. Mirko Planinić

Ocjena diplomskog rada: \_\_\_\_\_

Povjerenstvo: 1. \_\_\_\_\_

2. \_\_\_\_\_

3. \_\_\_\_\_

Datum polaganja: \_\_\_\_\_

Zagreb, 2009.

*Zahvaljujem se svom mentoru, prof. dr. sc. Mirku Planiniću, na strpljenju i savjetima koji su mi pomogli da što lakše i kvalitetnije napravim diplomski rad. Zahvaljujem se kolegi i prijatelju Ivanu Zubčiću na pomoći i razumijevanju oko izrade rada.*

*Želio bih se zahvaliti mojim roditeljima i bratu koji su bili uz mene tijekom čitavog studija i bili mi potpora u svemu što sam radio. Hvala vam na tome. Vama posvećujem ovaj diplomski rad.*

*Kenan Mačkić*

## **SADRŽAJ**

1. UVOD .....	4
2. POVIJESNI RAZVOJ NEURONSKIH MREŽA.....	5
3. USPOREDBA BIOLOŠKE NEURONSKE MREŽE I UMJETNE NEURONSKE MREŽE .	7
3.1    BIOLOŠKE NEURONSKE MREŽE .....	7
3.2    UMJETNE NEURONSKE MREŽE .....	8
4. PREPOZNAVANJE STATISTIČKOG UZORKA.....	10
4.1    PREPOZNAVANJE UZORKA.....	10
4.2    DODATAK .....	17
5. IZRADA KRIVULJE POMOĆU POLINOMA.....	22
5.1    GENERALIZACIJA .....	26
6. PRIMJENA NEURONSKIH MREŽA U FIZICI .....	30
7. NEURONSKE MREŽE U NASTAVI.....	32
8. ZAKLJUČAK.....	35
9. LITERATURA .....	37

## 1. UVOD

Što su neuronske mreže?

Neuronske mreže su računalni modeli za obradu informacija koji funkcioniraju na sličan način kao ljudski mozak ili neke druge biološke neuronske mreže. Sastoje se od velikog broja jednostavnih procesora (nelinearnih signalnih procesora koje zovemo neuronima ili jedinicama), pri čemu svaki od njih može imati malu lokalnu memoriju. Ključni element ovog modela je struktura sustava za obradu podataka. Ona se sastoji od velikog broja međusobno povezanih elemenata (neurona) koji skladno rade u svrhu što boljeg rješavanja konkretnog problema. Umjetne neuronske mreže imaju sposobnost poput ljudi učiti na primjerima, skupovima podataka te na taj način vrlo brzo izlaze na kraj s problemima koji se teško rješavaju tradicionalnim pristupom. Svaka mreža je konfigurirana za određenu primjenu, kao što je prepoznavanje uzorka, govora ili podataka klasifikacije kroz proces učenja. Učenje u biološkim sustavima podrazumijeva prilagođavanje na sinaptičkim vezama koje postoje među neuronima, a to pravilo vrijedi i za umjetne neuronske mreže.

## 2. POVIJESNI RAZVOJ NEURONSKIH MREŽA

Pedesetih godina prošlog stoljeća na konferenciji *Dartmouth Summer Research Project*, kao jedna od grana umjetne inteligencije, profiliralo se neuronsko računarstvo. Na toj konferenciji je najavljeno ostvarivanje vizije računalnog modela koji bi mogao u nekim temeljnim principima oponašati funkcioniranje mozga. Pojavljivanje neuronskog računarstva je odgovor Von Neumannovim računalima, nastojeći ostvariti ili simulirati usporednu obradu informacija koju koristi ljudski mozak dok se sjeća, razmišlja, rješava probleme itd.

Neki vrlo važni događaji za razvoj neuronskih mreža:

- 1943. Pitts i Mulock su prvi dokazali da neuroni mogu imati dva stanja (umirujuće i pobuđujuće) te da njihova aktivnost ovisi o određenoj vrijednosnoj granici. Ti dokazi su bili temelj za razvoj neuronskih mreža.
- 1949. Hebb je dao prijedlog za pravilo kojim se opisuje proces učenja (Hebbovo pravilo).
- 1956. Rochester i skupina autora predstavljaju prvu simulaciju Hebbovog modela na *Dartmouth Summer Conference*, koji je kolijevka modela neuronskih mreža.
- 1958. prva neuronska mreža „Perceptron“, koju je razvio Frank Rosenblatt, u kojoj se učenje razvija u dva sloja, nije mogla rješavati probleme klasifikacije koji nisu bili linearno djeljivi (npr. XOR problem).
- 1969. Minsky i Papert objavljuju rad u kojem oštro kritiziraju nedostatke „Perceptrona“ što dovodi do prekida ulaganja u razvoj neuronskih mreža.
- 1974. unatoč slabim ulaganjima Paul Werbos je razvio višeslojnu Perceptron mrežu – MLP, prvu verziju Backpropagation mreže koja prevladava nedostatak Perceptrona uvođenjem učenja u „skrivenom sloju“.

- 1986. Rumelhart, Hinton i Williams usavršavaju „Backpropagation“ mrežu koja vraća ugled neuronskim mrežama jer omogućuje aproksimiranje gotovo svih funkcija i rješavanje praktičnih problema.

Usavršenom „Backpropagation“ mrežom dolazi do potpunog znanstvenog i komercijalnog svjetskog interesa za neuronske mreže čija je primjena i danas mnogostruka te se koristi u mnogim područjima ljudskog djelovanja i raznim industrijama. Postoje i drugi algoritmi razvijeni za neuronske mreže.

### 3. USPOREDBA BIOLOŠKE NEURONSKE MREŽE I UMJETNE NEURONSKE MREŽE

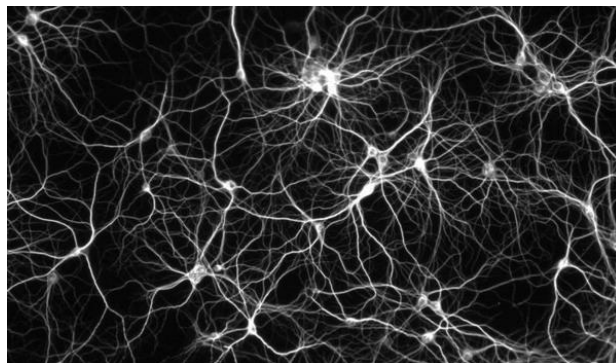
Iako su neuronske mreže nastale po uzoru na biološke, ova se dva modela u nekim karakteristikama međusobno razlikuju.

#### 3.1 BIOLOŠKE NEURONSKE MREŽE

Biološka neuronska mreža je vrlo složena. Sastoji se od **tijela, aksona i mnoštva dendrita** koji okružuju tijelo neurona. Akson je tanka cjevčica koja s jedne strane sadrži tijelo neurona, a s druge se strane dijeli na niz grana. Na krajevima tih grana nalaze se zadebljanja koja dodiruju dendrite, a ponekad i tijelo drugog neurona. Sinapsa je razmak između završetka aksona prethodnog neurona i dendrita ili tijela sljedećeg neurona. Dendriti su produžeci staničnog tijela, vrlo su tanki u odnosu na veličinu tijela stanice i funkcija im je prikupljanje informacija.

Biološke neuronske mreže sastavni su dio čovjeka čiji se mozak sastoji od 100 milijardi neurona, a zaslužne su za izvođenje vrlo važnih funkcija kao što su razmišljanje, učenje, emocije, precepcija, motorika itd.

Iako se funkcije neuronskih mreža istražuju već dugi niz godina, brojni procesi, kao i njihov način rada, ljudskom umu još uvijek nisu u potpunosti razumljivi.



Slika 1 Prikaz biološkog neurona

## 3.2 UMJETNE NEURONSKE MREŽE

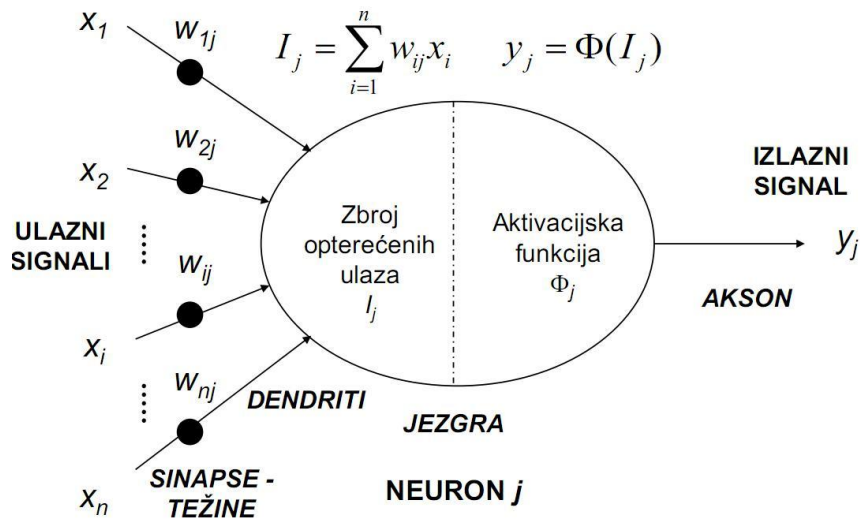
Umjetne neuronske mreže dizajnirane su tako da oponašaju osnovne funkcije bioloških neuronskih mreža. Tijelo biološkog neurona zamjenjuje se sumatorom, ulogu dendrita preuzimaju ulazi (ulazne varijable) u sumator, izlaz (izlazna varijabla) sumatora je akson umjetnog neurona, a uloga praga osjetljivosti bioloških neurona preslikava se iz tzv. aktivacijske funkcije (slika 2). Veza umjetnog neurona s okolinom ostvaruje se pomoću funkcijske sinaptičke veze biološkog neurona. Težinski faktori mogu biti pozitivan ili negativan broj, a imaju istu funkciju kao i sinapse kod biološkog neurona: povezuju izlaze (izlazne varijable) iz okoline neurona tj. izlaze drugih neurona (aksone) s ulazima sumatora (dendriti). Intenzitet te veze ovisi o iznosu (modulu), a vrsta veze o predznaku težinskog faktora. Aktivacijske funkcije dijele se na: ***linearne i nelinearne***.

- **Linearne aktivacijske funkcije** – izlaz sumatora množi se s nekim faktorom (pojačanjem) i tako dobiva izlaz neurona.
- **Nelinearne aktivacijske funkcije** – mogu poprimiti različite oblike, a najrasprostranjenije su : funkcije praga osjetljivosti, sigmoidne, hiperbolične i harmoničke funkcije. Nelinearne aktivacijske funkcije prevode izlaz sumatora na izlaz neurona preko nelinearnog pojačanja. Na taj način funkcija praga osjetljivosti daje na izlazu neurona 1, ako je izlaz sumatora veći od zadanog broja (prag osjetljivosti), što odgovara ispaljivanju impulsa kod biološkog neurona. Ukoliko neuron nije aktivan, onda je na izlazu neurona 0.

Iz priloženog se vidi da umjetni neuron funkcionira slično kao biološki neuron. Izlazi iz drugih neurona u okruženja promatranog neurona, koji se upućuju promatranom neuronu množe se težinskim faktorima i dovode do sumatora. U sumatoru se dobiveni produkti sumiraju, a njihova suma se odvodi na ulaz



aktivacijske funkcije, koja će na svom ulazu dati izlaz neurona. Na primjer, ako je izlaz sumatora označen s  $x$ , izlaz neurona s  $y$ , a aktivacijska funkcija je oblika sinusa vrijedit će izraz  $y = \sin(x)$ . Umjetni neuron na slici 2 naziva se po svojim autorima McCulloch – Pittsov neuron. Kada aktivacijska funkcija ima oblik praga osjetljivosti (binary threshold function) onda se opisani neuron naziva **Perceptron**. Njegova vrlo jednostavna struktura zanemaruje mnoge karakteristike biološkog neurona (npr. ne vodi računa o kašnjenju signala jer ulaz neurona producira trenutačno njegov izlaz, što se odražava na dinamiku sustava).



Slika 2 Model umjetnog neurona

## 4. PREPOZNAVANJE STATISTIČKOG UZORKA

Pojam prepoznavanja uzorka obuhvaća veliki broj informacija koji se odnose na probleme koji su značajni u praksi, od prepoznavanja i klasifikacije govora do pismenih sposobnosti i otkrivanja nedostataka u mehaničkoj i medicinskoj dijagnozi. Ljudi često takve probleme rješavaju prividno jednostavno.

No, takvi problemi su se u većini slučajeva pokazali „velikim“. Upravo zato pri rješavanju ovih problema nije dovoljno orijentirati se samo na praktična objašnjenja jer zbog svoje složenosti oni zahtjevaju detaljno poznavanje i primjenu teorije.

Okosnica rješavanja problema prepoznavanja statističkog uzorka je statistika koja prepoznaje vjerojatnost svih informacija koje tražimo u procesu i pruža najbolji način dobivanja rezultata. Tehnike statističkog uzorka vrlo su važne i svakako ih treba uzeti u obzir prilikom istraživanja.

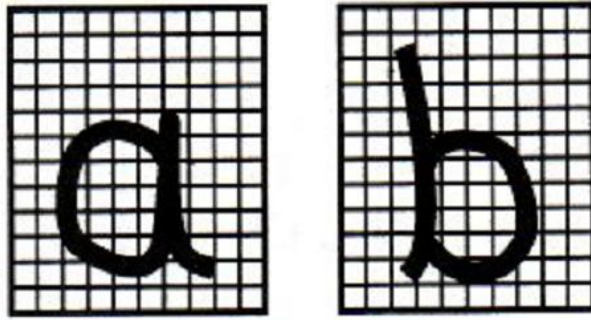
### 4.1 PREPOZNAVANJE UZORKA

Postoje brojni temeljni koncepti prepoznavanja statističkog uzorka. Pri prepoznavanju važno je uzeti u obzir hipotetički problem razlikovanja pismene verzije znakova „a“ i „b“. Slike znakova mogu biti zabilježene televizijskom kamerom i pohranjene u računalo. Potom tražimo algoritam koji najpouzdanije razlikuje dva znaka. Znak je prikazan nizom piksela kao što je prikazano na slici 3. Svaki piksel nosi odgovarajuću vrijednost koju označavamo s  $x_i$  (index „i“ označava pojedini piksel). Vrijednost  $x_i$  može se rasporediti od 0 (označava potpuno bijeli piksel) do 1 (označava crni piksel).

Često je najpraktičnije objediniti varijable  $x_i$  i označiti ih s pojedinačnim vektorom  $x = (x_1, \dots, x_d)^T$ , gdje „d“ označava ukupan broj varijabli, a eksponent T označava promjenjivi red. Promatrajući ovaj primjer zanemarit ćemo broj detaljnih, praktičnih značenja koja bi trebala biti adresirana u pravu implementaciju i fokusirati se na potcrtane rezultate. Cilj ove klasifikacije je razviti algoritam koji će pripisati svaku sliku (prikazanu pomoću vektora  $x$ ) jednom od dva skupa koji ćemo označiti s  $C_k$ , gdje  $k = 1, 2$ , tako da skup  $C_1$  korespondira (podudara se) sa znakom „a“, a skup  $C_2$  sa znakom „b“.

Pretpostavimo da imamo velik broj primjera slika koje korespondiraju i sa znakom „a“ i sa znakom „b“ koje su ljudi već klasificirali. Ovakva kolekcija odnosi se na skup podataka koji se u statistici naziva **uzorak**. Očiti problem s kojim se susrećemo proizlazi iz velike dimenzionalnosti podataka koje sakupljamo. Za sliku tipičnih dimenzija 256 x 256 piksela, svaka slika može se prikazati kao točka u  $d$  – dimenzionalnom prostoru, gdje je  $d = 65\,536$ . Osi ovog prostora predstavljaju **stupanj zatamnjenja** vrijednosti odgovarajućih piksela, koji u ovom primjeru mogu biti prikazani 8 – bitnim brojevima.

U biti, mogli bismo pohraniti svaku moguću sliku zajedno s njenom odgovarajućom rednom oznakom. U praksi je to, naravno, potpuno nepraktično zbog prevelikog broja mogućih slika: za sliku 256 x 256 sa 8 – bitnim vrijednostima postojalo bi  $2^{8 \times 256 \times 256} \approx 10^{158000}$  različitih slika. No, moguće je da postoji nekoliko tisuća primjeraka u trenirajućem skupu. Prema tome, potpuno je jasno da sustav klasifikacije mora biti organiziran tako da prethodno nepoznate vektore (slika) točno klasificira.



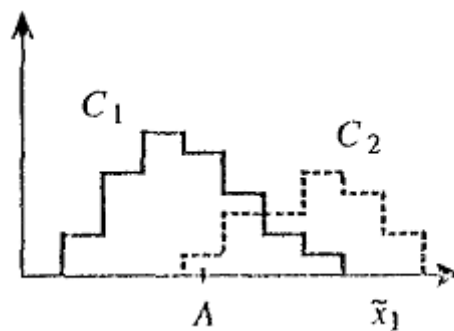
Slika 3

Ilustracija za dvije slike koje predstavljaju hipotetičnu verziju znakova „a“ i „b“ pisanih rukom. Svaka slika je opisana nizom piksela koji imaju vrijednost  $x_i$  koje se kreću od 0 do 1 prema kvadratu piksela označenih crnom bojom.

S druge strane, mogli bismo imati nekoliko tisuća primjera u našem treniranom skupu. Potpuno je jasno da u tom slučaju, sustav klasifikacije (po skupovima) mora biti dizajniran tako da može bez problema vrlo precizno klasificirati prijašnji, njemu nepoznat vektor slike (image vector). Postojanje velikog broja ulaznih varijabli može uzrokovati ozbiljne probleme za sustave prepoznavanja uzorka. Jedna od tehnika koja umanjuje ove probleme je kombinirati ulazne varijable zajedno, kako bi se dobio manji broj novih varijabli tzv. **Features**.

U ovom primjeru, mogli bismo vrednovati omjer visine i širine znakova, što ćemo označiti s  $\tilde{x}_1$ , budući da očekujemo da će znakovi iz skupa  $C_2$  (koji korespondiraju sa „b“) imati veću vrijednost  $\tilde{x}_1$ , nego znakovi iz skupa  $C_1$  (koji korespondiraju sa „a“). Tada se možemo nadati da će vrijednost od  $\tilde{x}_1$  omogućiti novim slikama da budu dodjeljeni pravoj klasi. Pretpostavimo da mjerimo vrijednost od  $\tilde{x}_1$  za svaku od slika u našem setu podatka i njihove vrijednosti prikažemo grafički za svaki skupinu posebno. Slika 4 pokazuje izgled tog grafa. Iz navedenog grafa vidimo da tipičan primjer znaka „b“ ima veću vrijednost  $\tilde{x}_1$ , nego primjer znaka „a“. Također vidimo da se ova dva prikaza preklapaju tako da se ponekad može dogoditi da se „susretnemo“ s primjerom „b“ koji ima manju vrijednost  $\tilde{x}_1$ , nego primjer znaka „a“. Upravo zato ne možemo precizno razlikovati dva skupa koristeći samo vrijednost od  $\tilde{x}_1$ . Za novu sliku koja ima vrijednost  $\tilde{x}_1$  određena točkom A (kao što je prikazano na slici 4) možemo zaključiti da slika pripada skupu  $C_1$ , a ne skupu  $C_2$ . Upravo se zato nastoji napraviti klasifikacijski sustav. Takav bi sustav jednostavno koristio prag za vrijednost  $\tilde{x}_1$ , a koji u  $C_2$  razvrstava svaku sliku za koju  $\tilde{x}_1$  prelazi granicu. Sve ostale slike razvrstava u skup  $C_1$ . Možemo očekivati da će prema ovom pristupu broj pogrešnih klasifikacija biti sveden na najmanju mjeru, ako odaberemo da prag bude u točki u kojoj se histogrami preklapaju.

Proces klasifikacije koji smo do sada opisali temelji se na vrednovanju  $\tilde{x}_1$  i usporedbi njegove vrijednosti s pragom. Za očekivati je da će ovaj proces rezultirati određenim stupnjem diskriminacije između dva skupa. Kao što prikazuje slika 4 u ovom procesu postoji veliki nedostatak, a to je značajno preklapanje dva histograma. Prema tome, možemo očekivati da će novi znakovi, koje ćemo testirati, biti pogrešno razvrstani.

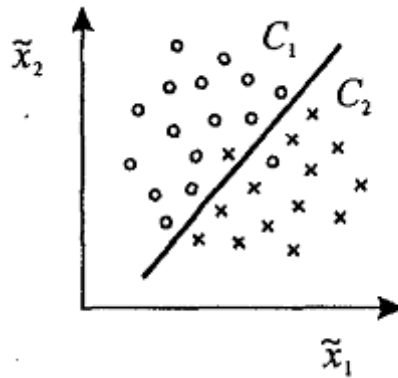


Slika 4

Shematski prikaz histograma varijabli  $\tilde{x}_1$  prikazan omjerom visine i širine znaka za podatkovni skup slika koje sadrže primjere iz skupova  $C_1 \equiv „a“$  i  $C_2 \equiv „b“$ . Primjećujemo da znakovi iz skupa  $C_2$  teže prema većim vrijednostima  $\tilde{x}_1$ , nego znakovi iz skupa  $C_1$ , ali ima značajno preklapanje između dva histograma. Ako promatramo sliku s vrijednosti  $\tilde{x}_1$ , koja je prikazana točkom A, možemo očekivati da slika više pripada skupu  $C_1$ , nego skupu  $C_2$ .

Jedan od načina da pokušamo poboljšati situaciju jest da razmotrimo drugu karakteristiku  $\tilde{x}_2$  (čiju stvarnu definiciju nije potrebno razmotriti) i da pokušamo klasificirati nove slike na osnovu vrijednosti  $\tilde{x}_1$  i  $\tilde{x}_2$  razmatrajući ih zajedno. Razlog zbog čega bi nam ovo moglo koristiti je naznačen u slici 5. Ovdje ćemo vidjeti primjere obrazaca iz dvije kategorije iscrtane u prostoru  $(\tilde{x}_1, \tilde{x}_2)$ .

Moguće je povući liniju u ovom prostoru, poznatom kao **decision boundary** (granica odluke), što nam daje dobro razdvajanje dvaju skupova. Novi uzorci koji leže iznad granice odluke se klasificiraju kao pripadnici skupa  $C_1$ , dok se uzorci pada ispod granice odluke klasificiraju kao pripadnici skupa  $C_2$ . Nekoliko primjera je i dalje krivo klasificirano, ali odvajanje od uzoraka je puno bolje, nego da je karakteristika bila gledana pojedinačno, kao što se može vidjeti s obzirom na sve ove podatke projicirane kao histogrami na jednu ili drugu os.



Slika 5

Hipotetički problem klasifikacije uključuje dvije varijable  $\tilde{x}_1$  i  $\tilde{x}_2$ . Kružići označavaju uzorke iz skupa  $C_1$ , a križići označavaju uzorke iz skupa  $C_2$ . Granica odluke (prikazana linijom) je u mogućnosti predvidjeti dobru separaciju između dva skupa, iako još uvijek postoji nekoliko uzoraka koji će biti neispravno podijeljeni prema ovoj granici. Imajmo na umu da ako je vrijednost bilo koje od dviju značajki smatrana svaka za sebe (odgovaraju prikazu podataka na jednoj ili drugoj osi) tada bi bilo znatno veće preklapanje u dva skupa.

Možemo i dalje razmatrati sve veći broj (nezavisnih) karakteristika i nadati se neodređenom poboljšanju performansi. Što više karakteristika dodajemo, dobivamo sve lošije rezultate. Kod stvarnih aplikacija prepoznavanja uzoraka preklapanja među distribucijskim klasama ponekad su neizbježna. To stvara problem kod klasifikacije uzorka zbog čestog podudaranja jedne klase s drugom.

Kod znakova pisanih rukom, na primjer, postoji znatna promjenjivost u načinu na koji su znakovi nacrtani.

Prisiljeni smo tretirati mjerene varijable kao slučajne vrijednosti i prihvatiti da savršeno klasificiranje novih primjera možda neće uvijek biti moguće. Umjesto toga možemo težiti gradnji „klasifikatora“ koji ima najmanju vjerojatnost pogreške.



## 4.2 DODATAK

Koristeći program MATLAB 7.5.0 (R2007b) izradio sam program koji služi za prepoznavanje slova engleske abecede. Program radi tako da se kreira neuronska mreža koja se zatim trenira (uči) na dobivenim podacima s greškom (šumom). Program testiramo tako da mreži predamo jedno slovo sa unaprijed određenim stupnjem deformacije, nakon čega neuronska mreža pokušava pogoditi o kojem je slovu riječ.

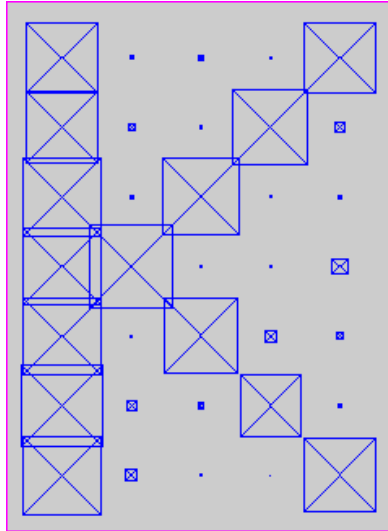
Program:

```
[alphabet,targets] = prprob;
net = newff(alphabet,targets,10,{'logsig','logsig'}, 'trainscg');
P = alphabet;
T = targets;
net.performFcn = 'sse';
net.trainParam.goal = 0.1;
net.trainParam.show = 25;
net.trainParam.epochs = 500;
net.trainParam.mc = 0.95;
net.trainParam.min_grad = 1.0000e-100;
[net] = train(net,P,T);
netn = net;
netn.trainParam.goal = 0.1;
netn.trainParam.show = 100;
netn.trainParam.epochs = 500;
netn.trainParam.min_grad = 1.0000e-100;
T = [targets targets targets targets];
P1 = P;
[R,Q] = size(P);
for pass = 1:10
    P = [(P1 + randn(R,Q)*0.1), (P1 + randn(R,Q)*0.2), (P1 + randn(R,Q)*0.3), (P1
+ randn(R,Q)*0.4)];
    [netn] = train(netn,P,T);
end

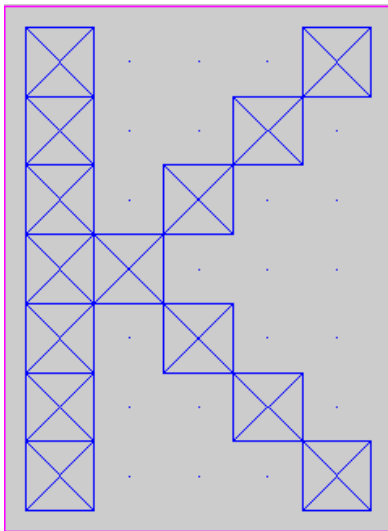
P = alphabet;
T = targets;
netn.performFcn = 'sse';
netn.trainParam.goal = 0.1;
netn.trainParam.show = 25;
netn.trainParam.epochs = 500;
netn.trainParam.mc = 0.95;
netn.trainParam.min_grad = 1.0000e-100;
[netn,tr] = train(netn,P,T);
noisyChar = alphabet(:,11)+randn(35,1) * 0.2;
plotchar(noisyChar);
pause(6);
A2 = sim(netn,noisyChar);
A2 = compet(A2);
answer = find(compet(A2) == 1);
plotchar(alphabet(:,answer));
```

Slučajevi:

a) stupanj deformacije 0,1



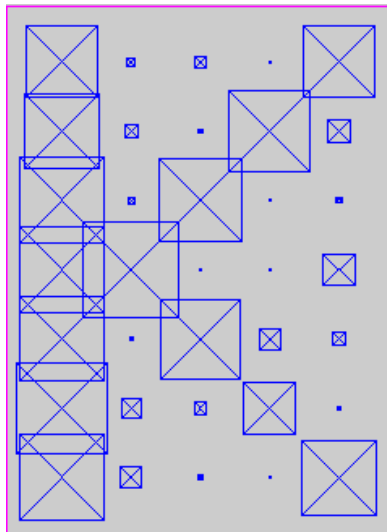
Slika 5.1 Slovo s greškom (šumom)



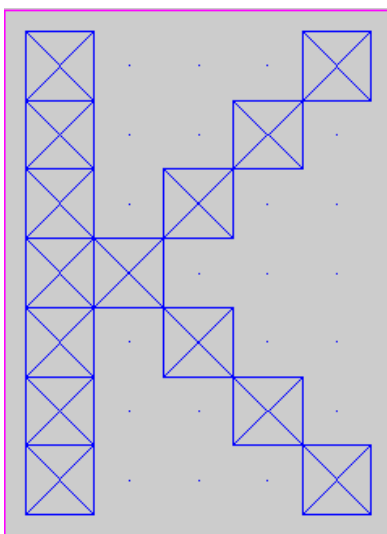
Slika 5.2 Slovo koje je mreža prepoznala

Vidimo da sa stupnjem deformacije od 0,1 mreža nema problema sa prepoznavanjem slova.

b) stupanj deformacije 0,2



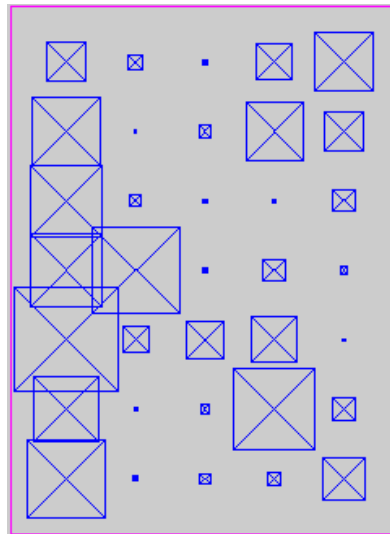
Slika 5.3 Slovo s greškom stupnja 0,2



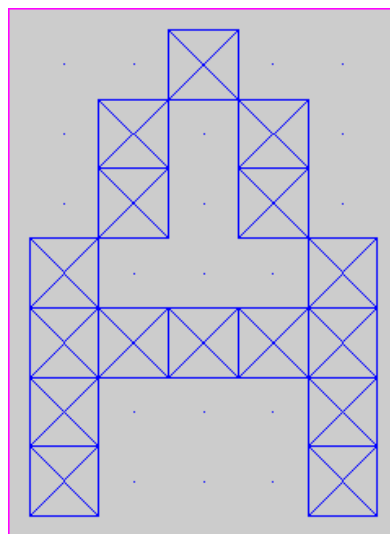
Slika 5.4 Slovo koje je mreža prepoznala

Na slici 5.3 vidi se da je stupanj deformacije veći nego na slici 5.1. Unatoč povećanom stupnju deformacije mreža je uspjela prepoznati o kojem slovu se radi.

c) stupanj deformacije 0,3



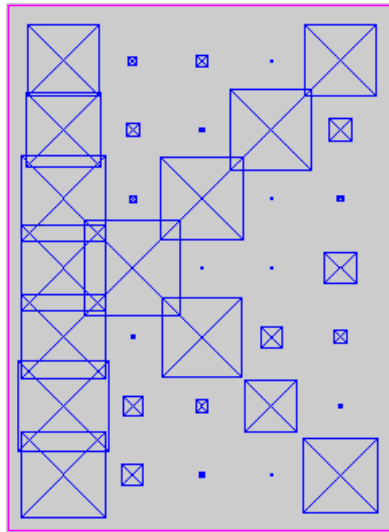
Slika 5.5 slovo s deformacijom stupnja 0,3



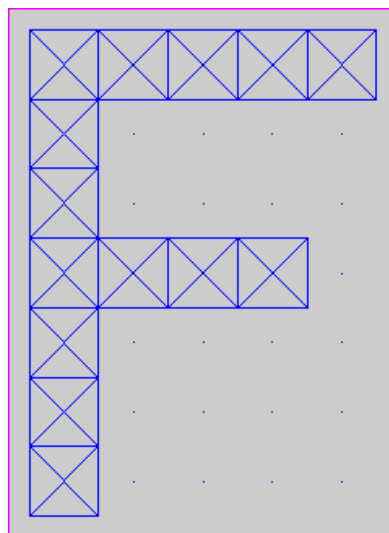
Slika 5.6 Slovo koje je mreža prepoznala

Pri stupnju deformacije 0,3 (slika 5.5), vjerojatnost mreže da pogodi o kojem slovu se radi je manja nego u slučajevima prije, stoga se događa da ponekad i ne pogodi o kojem se slovu radi.

d) slučaj pretrenirane mreže



Slika 5.7 Slovo s deformacijom



Slika 5.8 Slovo koje je pretrenirana mreža prepoznala

Sada samo dopustili da se neuronska mreža pretrenira, a to znači da se mreža adaptirala na šumove i da više nema mogućnost generalizacije → znači da proizvodi nasumične izlazne varijable. Mreža trenira na slovima sa šumom (slova koja nisu pravilno napisana). Matrica je previše istrenirana što dovodi do toga da se mreža adaptirala na sami šum tako da gubi mogućnost prepoznavanja slova.

## 5. IZRADA KRIVULJE POMOĆU POLINOMA

Bit upotrebe neuronskih mreža može se prikazati vrlo jednostavno pomoću polinomske izrade krivulje. Problem je uskladiti (podesiti) funkciju polinoma sa skupinom od  $N$  podataka koristeći tehniku minimiziranja funkcije pogreške.

Promotrimo sljedeći polinom  $M$ -tog reda :

$$y(x) = w_0 + w_1 x + \dots + w_M x^M = \sum_{j=0}^M w_j x^j \quad (1)$$

Problem se može promatrati kao nelinearni prikaz u kojem je varijabla  $x$  prikazana kao ulazna varijabla (input), a  $y$  kao izlazna varijabla (output). Točna forma funkcije  $y(x)$  je određena vrijednostima parametara  $w_0 \dots w_m$  koje su slične (analogne) težinama u neuronskim mrežama. Skup parametara ( $w_0 \dots w_m$ ) označava se vektorom  $w$ . Tako se polinom može prikazati u formi  $y = y(x;w)$ . Označit ćemo podatak s indeksom  $n = 1, \dots, N$ , tako da se svaki pojedini podatak sastoji od vrijednosti  $x$ , označen  $x^n$ ; odgovarajuće vrijednosti za izlaznu varijablu  $y$  koju ćemo označiti sa  $t^n$ . Te izlazne varijable se u neuralnim mrežama nazivaju **ciljne vrijednosti**. Kako bi se pronašle odgovarajuće vrijednosti za koeficijente u polinomu potrebno je uzeti u obzir grešku između očekivane izlazne varijable  $t^n$  (za određenu ulaznu varijablu  $x^n$ ) i odgovarajućih procjenjenih vrijednosti na temelju funkcije polinoma određene prema  $y(x^n;w)$ . Standardna procedura izrade krivulje uključuje metodu najmanje kvadratne pogreške u obliku (izraz 2)

$$E = \frac{1}{2} \sum_{n=1}^N \{y(x^n; w) - t^n\}^2 \quad (2)$$

Varijablu E možemo promatrati kao varijablu koja je funkcija od  $w$ , tako da ovaj polinom može odgovarati podacima na način da se odabere vrijednost za  $w$  koju ćemo označiti sa  $w^*$ , a to smanjuje varijablu E. Primijetimo da je ovaj polinom u izrazu (1) linearna funkcija parametra  $w$  pa je tako izraz (2) kvadrat funkcije  $w$ . To znači da se minimum varijable E može pronaći iz linearne algebarske jednadžbe funkcije koje linearno ovise o parametrima, a to su tzv. **linearni modeli**, iako mogu biti nelinearne funkcije.

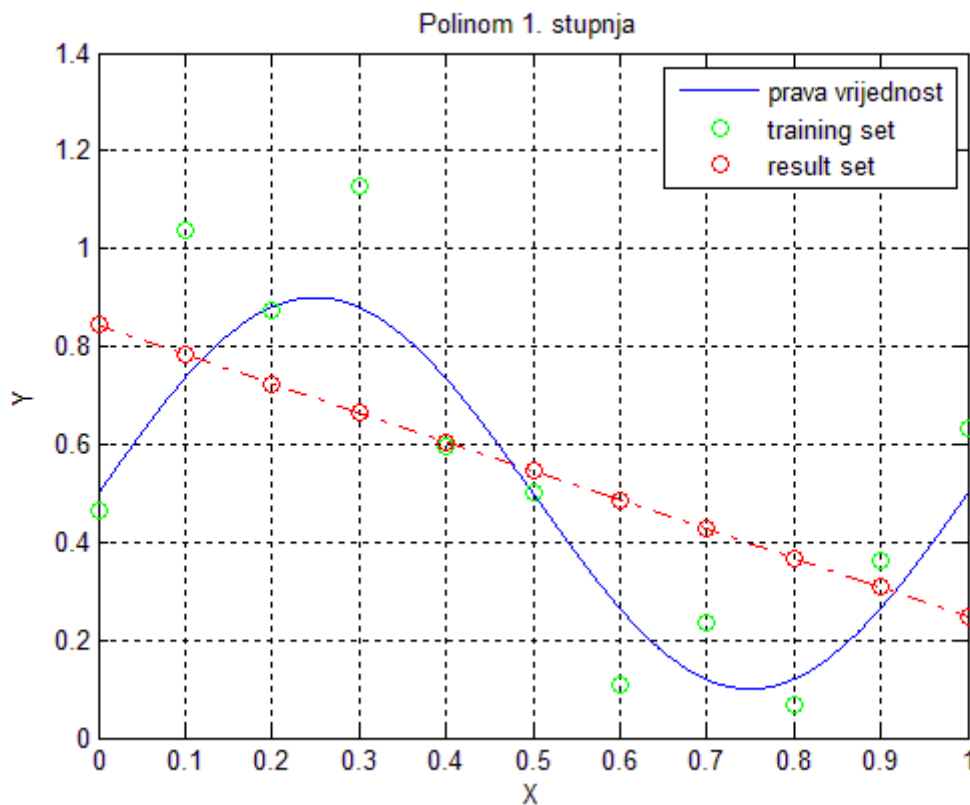
Minimiziranje funkcije pogreške kao u izrazu (2) koji uključuje ciljne vrijednosti za mrežne izlazne varijable naziva se **kontrolirano učenje** (supervised learning), budući da je za svaki uzorak ulazne varijable vrijednost očekivane izlazne varijable specificirana. Drugi način učenja u neuronskim mrežama je tzv. **nekontrolirano učenje** koje ne uključuje korištenje ciljnih podataka. Ovdje je cilj modelirati distribuciju vjerojatnosti ulazne varijable ili otkriti druge strukture u podacima. Treći oblik učenja je tzv. **pojačano učenje** u kojem su informacije zastupljene bez obzira jesu li izlazne varijable mreže dobre ili loše, ali opet kao i u drugom obliku ne postoje tražene vrijednosti. Tehnika izrade krivulje pomoću polinoma može se ilustrirati generirajući sintetičke podatke na način da se pokuša reproducirati neke od osnovnih osobina skupova realnih podataka kao kod problema prepoznavanja uzorka. Konkretno generiramo podatke na način da ih „treniramo“ iz sljedeće funkcije, izraz (3)

$$h(x) = 0.5 + 0.4 \sin(2\pi x) \quad (3)$$

koristeći funkciju  $h(x)$  u jednakim intervalima od  $x$ , dodavajući slučajni šum  $s$  Gaussovom raspodjelom koja ima standardnu devijaciju  $\delta = 0.05$ .

Osnovni cilj prepoznavanja uzorka je stvoriti sustav koji će precizno predvidjeti nove podatke. Kako bi izmjerili mogućnosti polinoma, stvorili smo drugi skup podataka tzv. **test set**, koji je stvoren na isti način kao i skup podataka za treniranje, ali s novim vrijednostima za komponentu šuma. To odražava osnovnu pretpostavku da su podaci na kojima želimo primijeniti postupak prepoznavanja uzorka proizvedeni istim mehanizmom kao i podaci za treniranje. Najbolji način generaliziranja novih podataka dobit ćemo, kada preslikavanje (mapping) predstavlja osnovni aspekt podataka, umjesto promatranja određenog detalja (npr. noise contribution) iz skupa za treniranje. Stoga ćemo težiti promatranju koliko je polinom  $y(x)$  blizu funkciji  $h(x)$ . Slika 6 prikazuje 11 točaka iz skupa za treniranje kao i samu funkciju  $h(x)$  s izrazom (3) zajedno s rezultatom podešavanja linearnog polinoma danog s izrazom (1) uz  $M = 1$ .





Slika 6

Na slici vidimo primjer skupa od 11 točaka podataka dobivenih funkcijom  $h(x)$ , definirane izrazom (3) u jednakim intervalima od  $x$  i dodane komponente slučajnog šuma. Isprekidana krivulja prikazuje funkciju  $h(x)$ , dok puna krivulja prikazuje približno lošu aproksimaciju dobivenu linearnim polinomom (izraz 1) uz  $M = 1$ .

Kao što vidimo taj polinom daje loš prikaz funkcije  $h(x)$ , kao njegova posljedica ograničene fleksibilnosti. Možemo dobiti bolje slaganje povećavajući red polinoma obzirom da na taj način povećavamo broj stupnjeva slobode (tj. broj slobodnih parametara) u funkciji, što nam daje fleksibilnost.

Slika 7 prikazuje rezultat podešavanja (fitting) polinoma trećeg stupnja ( $M=3$ ), koji daje puno bolju aproksimacije funkcije  $h(x)$ . Međutim, ako previše povećamo red polinoma onda aproksimacija funkcije postaje lošija, u smislu lošijeg predviđanja točaka i sve većeg odstupanja od zadane funkcije.

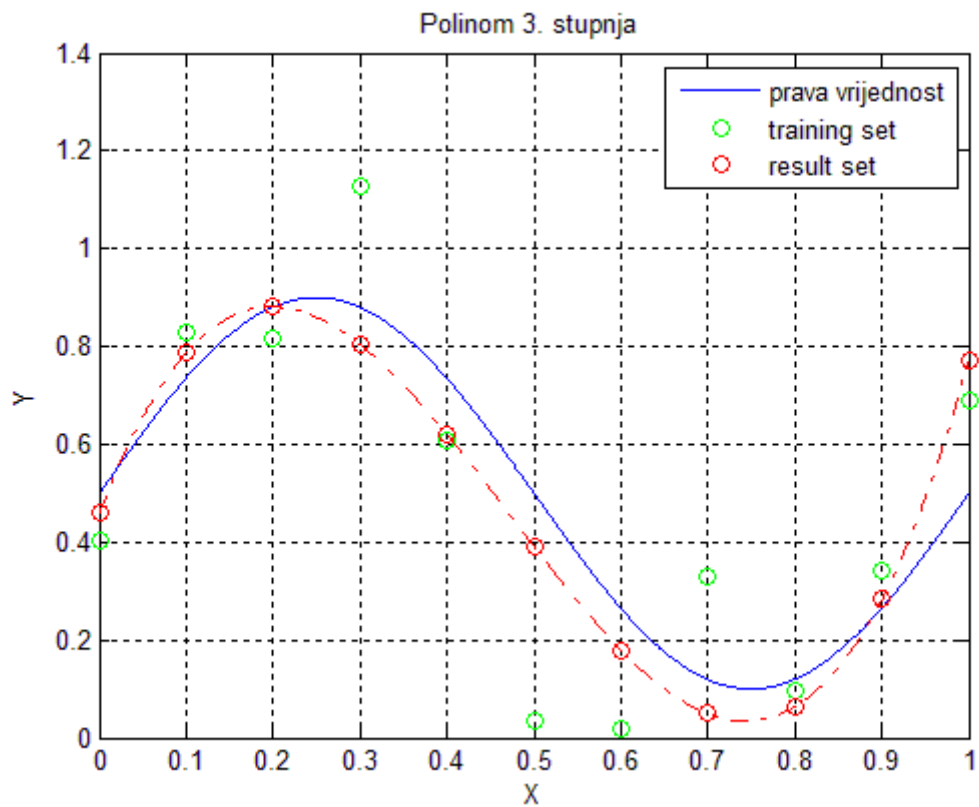
Slika 8 prikazuje rezultate koje smo dobili podešavanjem (fitting) polinoma 10. reda ( $M=10$ ). Ovime je moguće ostvariti savršeno pristajanje (perfect fit) trenirajućem setu, budući da polinom desetog reda ima 11 slobodnih parametara, a time i 11 podatkovnih točaka. Međutim, polinom podešava podatke razvijajući dramatične oscilacije. Za takve funkcije kažemo da su **pretrenirane**. Kao posljedica toga pretrenirana funkcija onemogućava dobro predviđanje funkcije  $h(x)$  za zadani  $x$ .

## 5.1 GENERALIZACIJA

Kako bismo ocijenili sposobnost polinoma da generalizira nove podatke, uobičajeno je razmotriti srednju kvadratnu grešku koja je dana izrazom :

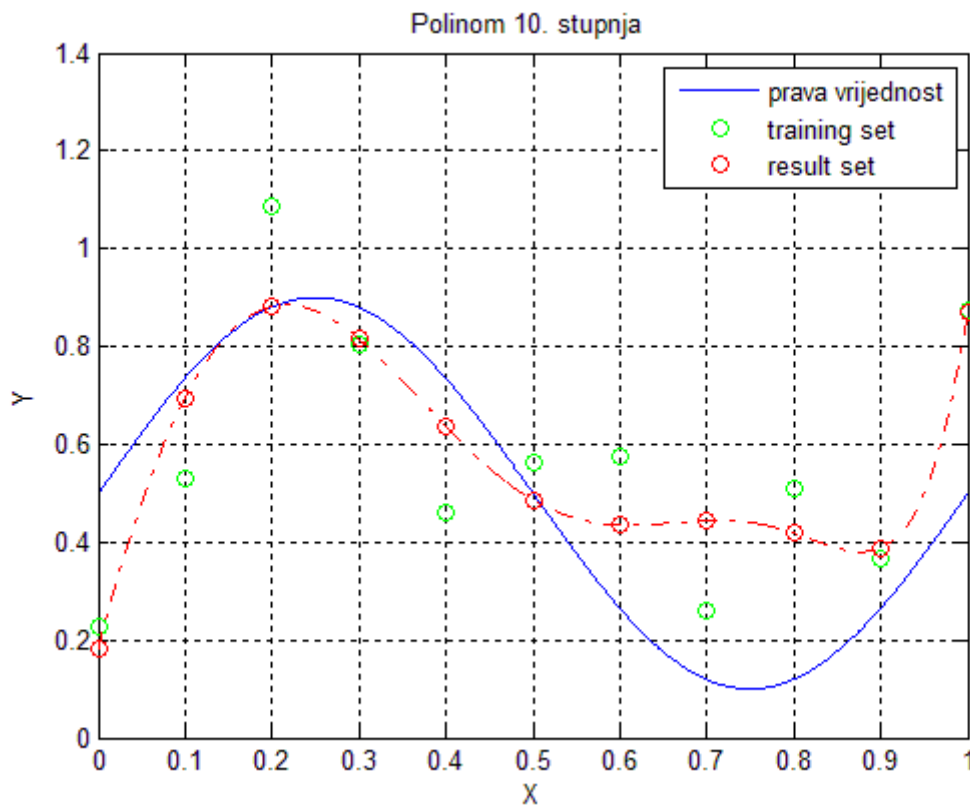
$$E^{RMS} = \sqrt{\frac{1}{N} \sum_{n=1}^N \{y(x^n; w^*) - t^n\}^2} \quad (4)$$

gdje  $w^*$  predstavlja vektor koeficijenata koji odgovaraju minimumu funkciji greške, tako da  $y(x, w^*)$  predstavlja podešeni polinom. Za svrhu procjene efikasnosti polinoma kod predviđanja novih podataka, ova je veličina puno prikladnija od izvorne sume kvadrata (izraz 2), budući da je uklonjena jaka ovisnost o broju podataka.



Slika 7

Na slici 7 je prikazan isti skup podataka kao i na slici 6 ali ovaj put je podešen polinom 3. stupnja ( $M = 3$ ) prikazujući bitno poboljšanu srednju aproksimaciju funkcije  $h(x)$ , koja je ostvarena fleksibilnijom funkcijom.



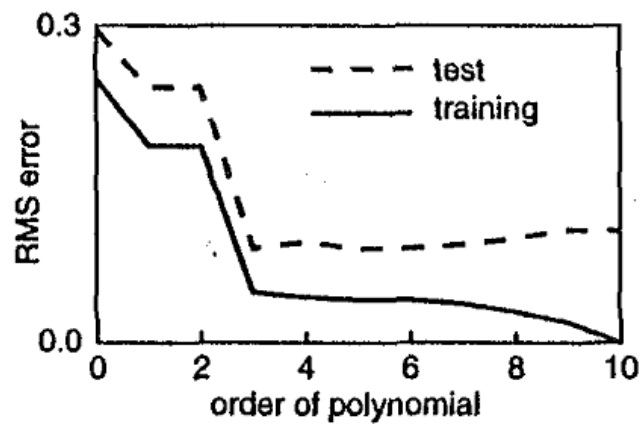
Slika 8

Slika 8 grafički prikazuje skup podataka koristeći polinom desetog stupnja ( $M = 10$ , prikazan crvenom linijom) koji naizgled savršeno odgovara zadanoj funkciji. No zbog prevelikih oscilacija funkcije sa polinomom desetog stupnja vidljivo je da se ona ne podudara sa pravom vrijednošću kao što se podudara funkcija sa polinomom trećeg stupnja.

Slika 9 prikazuje graf  $E^{\text{RMS}}$  i za trenirajući skup podataka i za testni skup podataka kao funkciju reda polinoma  $M$ . Vidimo da greška kod trenirajućeg skupa blago opada kako red polinoma  $M$  raste. Greška testiranog skupa, međutim, doseže minimum u  $M = 3$ , a nakon toga raste s porastom reda polinoma.

Sposobnost polinoma da generalizira nove podatke (tj. testni skup) stoga doseže optimalnu vrijednost za polinom određenog stupnja kompleksnosti. Linearni polinom sa slike 6 je primjer modela sa premalo fleksibilnosti zbog velikog nagiba (posljedica loše aproksimacije). Primjer polinoma desetog reda (slika 8) je model koji ima previše fleksibilnosti zbog velikog odstupanja.

Svrha što bolje generalizacije određena je balansiranjem između ovih dvaju konkurentnih karakteristika, a nastaje kada je broj stupnjeva slobode u modelu relativno mali u odnosu na veličinu skupa podataka (4 slobodna parametra za  $M = 3$ , u usporedbi sa 11 točaka podataka iz ovog primjera).

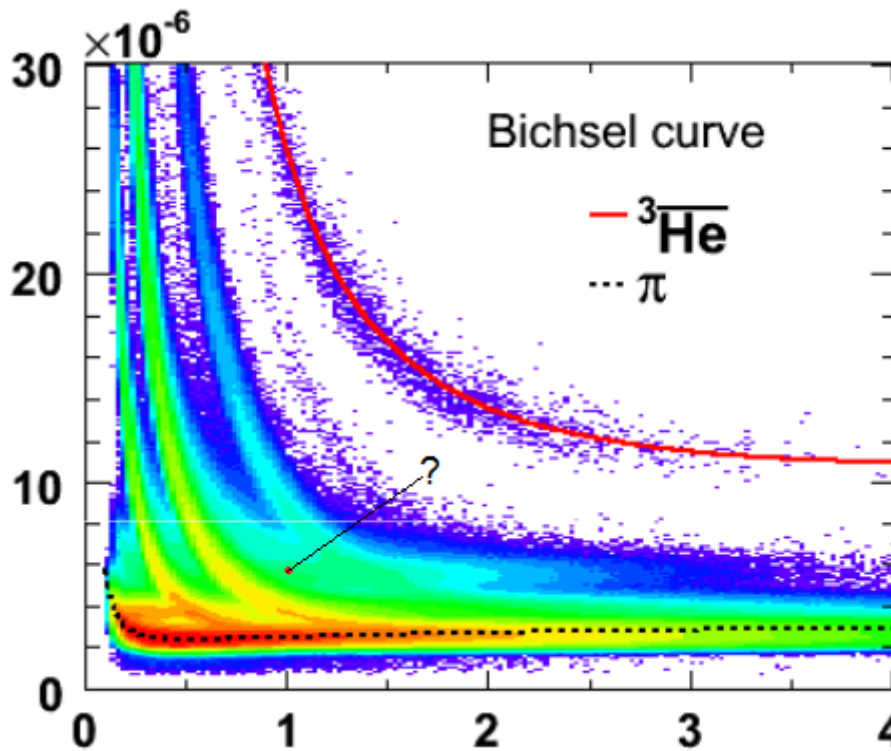


Slika 9

Iscrtana RMS pogreška (izraz 4) kao funkciju reda polinoma za oboje - i pokusni trening i test postavku - npr. problem razmatran u tri prethodna prikaza. Greška: s uvažavanjem trening (pokusne) postavke određuje monotono povećavanje funkcije prema  $M$ , dok greška u predviđanjima za nove podatke izmjerenih u posebnim postavkama pokazuje minimum na  $M = 3$ .

## 6. PRIMJENA NEURONSKIH MREŽA U FIZICI

Neuronske mreže u fizici imaju široku primjenu. Jedan od primjera gdje je možemo upotrijebiti je prepoznavanje čestica u  $\frac{dE}{dx}$  krivulji prikazanoj na slici 10.



Slika 10  $dE/dx$  krivulja

Na slici se vidi mnoštvo raznih čestica sa odgovarajućim krivuljama za pojedinu česticu. Na crvenoj krivulji možemo sa velikom preciznošću odrediti o kojoj se čestici radi. Što ako nas zanima koja je vrste čestica u zelenoj „smjesi“ svih čestica (označena crvenom točkom)?

U tom će nam slučaju dobro doći neuronske mreže, jer nam one mogu reći kolika je vjerojatnost da je to elektron, proton, pion ili neka druga čestica.

Slika 10 predstavlja opis Bethe – Blocheove formule (izraz 5). Ona opisuje gubitak energije elektrona ili neke druge čestice koja putuje velikom brzinom u materijalu po pređenom putu.

Koliko će energije određena čestica izgubiti ovisi o brojnim faktorima kao što su gustoća materijala, dužina puta, vrsta čestice itd. Najvažnija ovisnost Bethe – Blocheove formule je ovisnost o naboju i masi čestice.

$$-\frac{dE}{dx} = \frac{4\pi}{m_e c^2} \cdot \frac{n z^2}{\beta^2} \cdot \left(\frac{e^2}{4\pi\epsilon_0}\right)^2 \cdot \left[ \ln\left(\frac{2m_e c^2 \beta^2}{I \cdot (1 - \beta^2)}\right) - \beta^2 \right] \quad (5)$$

Kazalo pojmova:

$$\beta = \frac{v}{c}$$

$v$  - brzina čestice

$E$  – energija čestice

$x$  - pređena udaljenost čestice

$c$  – brzina svjetlosti

$z$  - naboj čestice

$e$  – naboj elektrona

$m_e$  – masa elektrona

$n$  – gustoća elektrona

$I$  - srednji potencijal pobuđenja

## 7. NEURONSKE MREŽE U NASTAVI

Neuronske mreže su jako složene i treba dosta znanja kako bi se koristile. Svrstavaju se u dio naprednog programiranja koje se uči na fakultetu. Tema bi se mogla predstaviti učenicima srednje škole. Problem bi mogao nastati zbog nedostatka predznanja o programiranju, stoga je učenicima treba predstaviti i približiti na što jednostavniji način kroz mnoštvo primjera.

Na početku sata, učenike se uvodi u novu nastavnu jedinicu te im se postavljaju slijedeća pitanja:

- *Jeste li ikada čuli za neuronske mreže?*
- *Što su neuronske mreže?*
- *Gdje ih susrećemo?*
- *Čemu služe?*
- *Može li itko navesti primjer?*

Nakon kratkog uvoda, slijedi objašnjenje novih pojmova (biološki neuron, akson, dendriti, živčani sustav itd.) te obrada glavne teme.

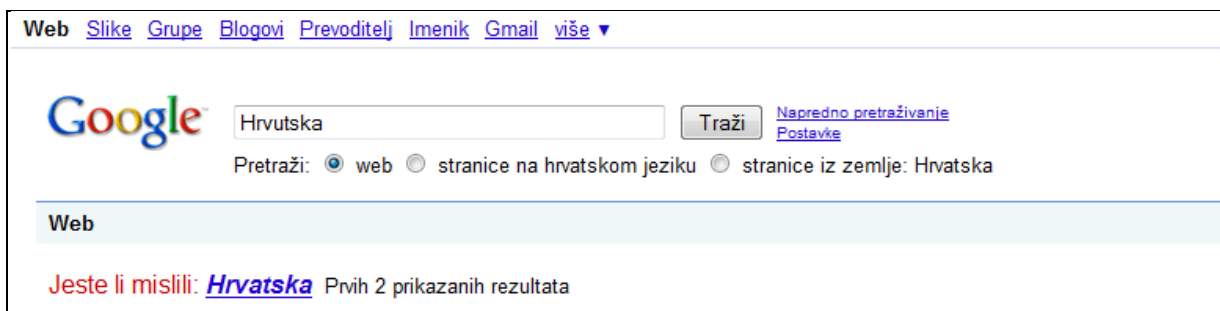
Kako bi što bolje objasnili neuronske mreže, potrebno je usporediti biološke i umjetne neuronske mreže te istaknuti njihove sličnosti. Npr. tijelo biološkog neurona zamjenjuje se sumatorom, ulogu dendrita preuzimaju ulazi (ulazne varijable) u sumator, izlaz (izlazna varijabla) sumatora je akson umjetnog neurona, a uloga praga osjetljivosti bioloških neurona preslikava se iz tzv. aktivacijske funkcije.

Učenicima je potrebno objasniti gdje će primjeniti neuronske mreže i zašto nam je njihovo poznavanje važno. Na taj ćemo način stvoriti motivaciju i zainteresiranost učenika za sat. Činjenica je da će im nastavni sat biti zanimljiviji kada znaju gdje se to gradivo može primjeniti i na koji način.



Zahvaljujući primjerima koji su nam dostupni na internetu ili na samom računalu, učenicima možemo kvalitetno približiti umjetne neuronske mreže.

Najbolji i najjednostavniji primjeri za objašnjavanje principa rada neuronskih mreža su oni koje se u praksi svakodnevno koriste, a koji su učenicima već dostupni i jako dobro poznati. Jedan od primjera koji svakako treba istaknuti je Google. To je pretraživač pomoću kojeg se učenicima vrlo jednostavno, u kratkom vremenu može demonstrirati rad neuronskih mreža i to tako da učitelj u tražilicu unese određeni pojam koji je neispravno napisan (npr. Hrvatska), a sličan je pojmu koji zaista tražimo. Zahvaljujući radu neuronske mreže, ovaj će pretraživač na ekranu „izbaciti“ točan pojam (Hrvatska). Ova metoda učenicima omogućuje da, u trenu, izravno vide rezultate rada neuronskih mreža.



Slika 11 Primjer rada neuronskih mreža na pretraživaču

Osim Google-a, većina drugih pretraživača funkcionira na sličnom principu. Postoji još nekoliko primjera koji se mogu koristiti za objašnjavanje neuronskih mreža. Navedimo neke: razne simulacije koje se najčešće koriste u izradi računalnih igara, kompjuterski programi koji služe za identifikaciju putem otiska prsta ili zjenice oka, meteorološke prognoze (predviđanje potresa, temperature), sve su područja koja bez neuronskih mreža ne bi mogla funkcionirati.

U završnom djelu sata važno je izvršiti evaluaciju pomoću koje učitelj dobiva povratnu informaciju jesu li učenici razumjeli nastavnu jedinicu. Jedan od načina provjere je da učitelj zada učenicima zadatak, tako da na komad papira napišu odgovore na sljedeća pitanja:

- *Napišite jedan primjer neuronskih mreža?*
- *Objasnite zašto mislite da su to neuronske mreže?*

## 8. ZAKLJUČAK

Primjena neuronskih mreža u današnjem svijetu tehnologije neminovna je. Zbog sve većeg broja korisnika koji pristupaju internetu kao i sve zahtjevnijih programa u poslovnom svijetu, gotovo je nemoguće pristupiti računalu, a da se pritom ne aktiviraju neuronske mreže bez kojih bi rad mnogih aplikacija bio nezamisliv. Najjednostavniji primjer rada neuronskih mreža je Google pretraživač. Ukoliko tražimo neki pojam, a nismo sigurni kako se on točno piše, tu nam pomaže pretraživač (slika 11) koji radi kao jedna velika optimalno istrenirana neuronska mreža.

Njihova sposobnost da uče na skupu podataka, čini ih vrlo fleksibilnima i moćnima. Nema potrebe izmišljati algoritam kako bi se obavio određeni zadatak, tj. nema potrebe da razumijemo interne mehanizme za određeni zadatak. One su nam također pogodne za sustave realnog vremena zbog brzog odaziva i kratkog računalnog puta, zahvaljujući njihovoj paralelnoj strukturi. Neuronske mreže pridonose i mnogim znanostima kao što su neurologija, psihologija (svakodnevno se koriste na dijelovima modela koji su slični živom organizmu, pokušavajući istražiti i oponašati „mehanizam“ ljudskog mozga), ekonomija (proizvodnja, trgovina, marketing itd), medicina (medicinska dijagnostika), fizika (prepoznavanje, klasifikacija čestica) itd.

Zahvaljući mogućnosti neuronskih mreža da prepoznaju uzorke (znakovi pisani rukom, detekcija lica, glasa, otpornost materijala itd) omogućen je napredak i olakšan je rad i u mnogim drugim područjima. Možda najzanimljiviji aspekt neuronske mreže jest mogućnost da se u bliskoj budućnosti proizvedu „mreže savjesti“ koje bi u potpunosti oponašale ljudsku vrstu. Pitanje je, hoće li ikada sustav umjetnih neuronskih mreža moći doseći ljudskom umu još uvijek nedokučivo savršenstvo biološke neuronske mreže.

Može li nas umjetna mreža savjesti uistinu u potpunosti zamjeniti. Jesmo li svjesni da svojom težnjom za napretkom tehnologije, stvaranjem umjetne inteligencije i svih drugih izuma koji nas mogu zamjeniti zapravo štetimo sebi, pitanja su koja ćemo si vrlo vjerojatno još dugo postavljati. U jedno možemo biti sigurni – odgovor leži u nama samima.

## 9. LITERATURA

1. Branko Novaković, Dubravko Majetić, Mladen Široki: *Umjetne neuronske mreže*, Fakultet strojarstva i brodogradnje. Zagreb 1998.
2. Christopher M. Bishop: *Neural Networks for Pattern Recognition*, Clarendon Press. Oxford 1995.
3. Miroslav Kiš; *Informatički rječnik*, Naklada Ljevak. Zagreb 2002.
4. <http://www.wikipedia.org/>
5. [http://www.doc.ic.ac.uk/~nd/surprise\\_96/journal/vol4/cs11/report.html#Introduction%20to%20neural%20networks](http://www.doc.ic.ac.uk/~nd/surprise_96/journal/vol4/cs11/report.html#Introduction%20to%20neural%20networks)
6. [http://ict.ewi.tudelft.nl/~duin/papers/ascii\\_00\\_NNReview.pdf](http://ict.ewi.tudelft.nl/~duin/papers/ascii_00_NNReview.pdf)
7. <http://www.tsrb.hr/meha/index.php>
8. <http://www.mojdoktor.hr>
9. <http://www.google.com>