

# Ubrzanje algoritma učenja neuronske mreže s povratnim

---

**Turnar, Robi**

**Undergraduate thesis / Završni rad**

**2010**

*Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj:* **University of Zagreb, Faculty of Mechanical Engineering and Naval Architecture / Sveučilište u Zagrebu, Fakultet strojarstva i brodogradnje**

*Permanent link / Trajna poveznica:* <https://urn.nsk.hr/urn:nbn:hr:235:994558>

*Rights / Prava:* [In copyright](#)/[Zaštićeno autorskim pravom.](#)

*Download date / Datum preuzimanja:* **2024-11-28**

*Repository / Repozitorij:*

[Repository of Faculty of Mechanical Engineering and Naval Architecture University of Zagreb](#)



Sveučilište u Zagrebu  
**Fakultet Strojarsstva i brodogradnje**

# **ZAVRŠNI RAD**

Robi Turnar

Zagreb, 2010.

Sveučilište u Zagrebu  
**Fakultet Strojarsstva i brodogradnje**

# **ZAVRŠNI RAD**

Voditelj rada:  
Prof. dr. sc. Dubravko Majetić

Robi Turnar

Zagreb, 2010.



SVEUČILIŠTE U ZAGREBU  
FAKULTET STROJARSTVA I BRODOGRADNJE



Središnje povjerenstvo za završne i diplomske ispite  
Povjerenstvo za završne ispite studija strojarstva za smjerove:  
proizvodno inženjerstvo, računalno inženjerstvo, industrijsko inženjerstvo i menadžment, inženjerstvo  
materijala i mehatronika i robotika

Sveučilište u Zagrebu	
Fakultet strojarstva i brodogradnje	
Datum	Prilog
Klasa:	
Ur.broj:	

## ZAVRŠNI ZADATAK

Student: **ROBI TURNAR**

Mat. br.: 0035166014

Naslov: **UBRZANJE ALGORITMA UČENJA NEURONSKE MREŽE S  
POVRATNIM RASPROSTIRANJEM POGREŠKE**

Opis zadatka:

Temeljni problem klasičnog algoritma učenja umjetne neuronske mreže s povratnim rasprostiranjem pogreške (EBP) je dugotrajan proces učenja. Zato se u cilju ubrzanja procesa učenja pojavljuju brojne modifikacije tog algoritma učenja. U radu treba pokazati i međusobno usporediti kvalitetu učenja klasičnog EBP algoritma s dvije metode njegovog unapređenja, metodom zamaha i metodom adaptacije parametra brzine učenja koju predlažu autori Silva i Almeida.

U radu je potrebno načiniti slijedeće:

1. Izvesti algoritam učenja statičke unaprijedne neuronske mreže s povratnim prostiranjem pogreške (EBP), pri čemu valja koristiti različite tipove aktivacijske funkcije skrivenog sloja neurona.
2. U klasični EBP algoritam uvesti metodu zamaha (momentuma).
3. U klasični EBP algoritam uvesti metodu adaptacije parametra brzine učenja prema Silvi i Almeidi.
4. Testirati učenje i generalizaciju izvedenih algoritama na primjerima klasifikacije i regresije.
5. Za pisanje programske podrške koristiti programski paket Matlab.
6. Izvesti zaključke rada.

Zadatak zadan:

11. prosinca 2009.

Zadatak zadao:

Prof. dr. sc. Dubravko Majetić

Rok predaje rada:

Prosinac 2010.

Predsjednik Povjerenstva:

Prof. dr. sc. Dubravko Majetić

## Sažetak

U ovom završnom radu opisana je umjetna neuronska mreža sa povratnim rasprostiranjem pogreške (EBP) te modifikacije klasičnog algoritma učenja s ciljem ubrzanja učenja. U prvom poglavlju ukratko je opisana cijela ideja neuronskih mreža i povezanost sa ljudskim neuronom. Drugo poglavlje opisuje strukturu statičke neuronske mreže, njen način funkcioniranja i klasični algoritam učenja. U svrhu uspoređivanja različitih aktivacijskih funkcija izvedeni su algoritmi učenja za četiri osnovne vrste aktivacijskih funkcija. Radi ubrzanja učenja dodane su dvije modifikacije algoritma učenja, sa momentumom prvog reda te sa metodom adaptacije brzine učenja prema Silvi i Almeidi. U trećem poglavlju opisana su tri primjera primjene statičkih neuronskih mreža, primjer klasifikacije XOR problema te identifikacija linearnog i nelinearnog dinamičkog sustava. Radi usporedbe testirane su sve aktivacijske funkcije te algoritmi učenja. Sve simulacije i testiranja provedeni su u programskom paketu Matlab.

# Sadržaj

Sažetak .....	4
Sadržaj .....	5
Popis slika .....	6
Popis tablica .....	7
Popis oznaka.....	8
Izjava. ....	9
1. Uvod .....	10
1.1. Biološki neuron .....	10
1.2. Umjetni neuron.....	11
1.3. Umjetne neuronske mreže .....	11
2. Statičke unaprijedne neuronske mreže .....	12
2.1. Statički neuron i statička neuronska mreža .....	12
2.2. Učenje statičke neuronske mreže .....	14
2.2.1. Unaprijedna faza učenja statičke neuronske mreže.....	16
2.2.2. Povratna faza učenja statičke neuronske mreže .....	17
2.3. Modifikacije algoritma učenja statičke neuronske mreže .....	22
2.3.1. Momentum (zamah) prvog reda .....	22
2.3.2. Metoda Silva i Almeida.....	23
3. Primjena statičkih neuronskih mreža .....	25
3.1. Klasifikacija .....	25
3.2. Identifikacija linearnog dinamičkog sustava.....	28
3.3. Identifikacija nelinearnog dinamičkog sustava .....	31
4. Zaključak .....	37
5. Literatura .....	38

## Popis slika

Slika 1.1. Biološki neuron .....	10
Slika 1.2. Struktura umjetnog neurona.....	11
Slika 2.1. Statički neuron .....	12
Slika 2.2. Najčešće korištene aktivacijske funkcije .....	13
Slika 2.3. Statička unaprijedna neuronska mreža.....	14
Slika 2.4. Učenje statičke neuronske mreže .....	15
Slika 3.1. Smanjivanje pogreške tijekom učenja XOR problema .....	27
Slika 3.2. Skup uzoraka za učenje P1 člana .....	29
Slika 3.3. Odziv nelinearnog dinamičkog sustava .....	31
Slika 3.4. Odziv naučene mreže sa aktivacijskom funkcijom skrivenog sloja linear .....	33
Slika 3.5. Odziv naučene mreže sa aktivacijskom funkcijom skrivenog sloja sigmoid.....	33
Slika 3.6. Odziv naučene mreže sa aktivacijskom funkcijom skrivenog sloja sinus .....	34
Slika 3.7. Odziv naučene mreže sa aktivacijskom funkcijom skrivenog sloja gauss.....	34
Slika 3.8. Odziv naučene mreže na neučeni signal sa funkcijom skrivenog sloja linear .....	35
Slika 3.9. Odziv naučene mreže na neučeni signal sa funkcijom skrivenog sloja sigmoid .....	35

## Popis tablica

Tablica 3.1. Parametri učenja za XOR problem.....	25
Tablica 3.2 Rezultati učenja za XOR problem.....	27
Tablica 3.3. Parametri učenja za linearni dinamički sustav .....	29
Tablica 3.4. Rezultati učenja za linearni dinamički sustav .....	30
Tablica 3.5. Parametri učenja za nelinearni dinamički proces .....	32
Tablica 3.6. Rezultati učenja za nelinearni dinamički sustav.....	32



## Popis oznaka

BIAS	dodatni neuron konstantnog izlaza jednakog jedinici
$c$	centar gaussove aktivacijske funkcije
$d$	koeficijent promjene brzine prema Silvi i Almeidi
$\mathbf{d}$	matrica izlaznih vrijednosti skupa za učenje
$d_k$	traženi izlaz k-tog neurona izlaznog sloja
$E$	suma kvadrata pogreške
EBP	algoritam povratnog rasprostiranja greške
$J$	broj neurona ulaznog sloja uvećan za jedan
$K_p$	konstanta pojačanja dinamičkog sustava
$k_p$	koeficijent smjera linearne aktivacijske funkcije
$O_k$	izlaz k-tog neurona izlaznog sloja
$N$	broj uzoraka iz skupa za učenje
NRMS	normalizirani korijen srednje kvadratne pogreške
$n$	korak učenja
net	suma statičkog neurona
$\text{net}H_j$	suma j-tog neurona skrivenog sloja
$\text{net}O_k$	suma k-tog neurona izlaznog sloja
$T$	vremenska konstanta dinamičkog sustava
$T_0$	vrijeme uzorkovanja dinamičkog sustava
$u$	koeficijent promjene brzine prema Silvi i Almeidi
$v_{ji}$	težina j-tog neurona skrivenog sloja i i-tog neurona ulaznog sloja
$w_{kj}$	težina k-tog neurona izlaznog sloja i j-tog neurona skrivenog sloja
$w_j$	težina j-tog ulaza statičkog neurona
$w_n$	težina n-tog ulaza umjetnog neurona
$x_j$	j-ti ulaz statičkog neurona
$x_n$	n-ti ulaz umjetnog neurona
$y$	izlaz statičkog neurona
$y_j$	izlaz j-tog neurona skrivenog sloja
$\mathbf{Z}$	matrica ulaznih vrijednosti skupa za učenje
$Z_i$	i-ti neuron ulaznog sloja
$\alpha$	koeficijent momentuma prvog reda
$\gamma$	aktivacijska funkcija statičkog neurona
$\gamma_j$	aktivacijska funkcija j-tog neurona skrivenog sloja
$\gamma_k$	aktivacijska funkcija k-tog neurona izlaznog sloja
$\eta$	koeficijent brzine učenja
$\vartheta$	težinski faktor neurona
$\sigma$	standardna devijacija gaussove aktivacijske funkcije

Izjavljujem da sam završni rad izradio samostalno stečenim znanjem i iskustvom uz navedenu literaturu i konzultacije sa voditeljem rada.

Robi Turnar

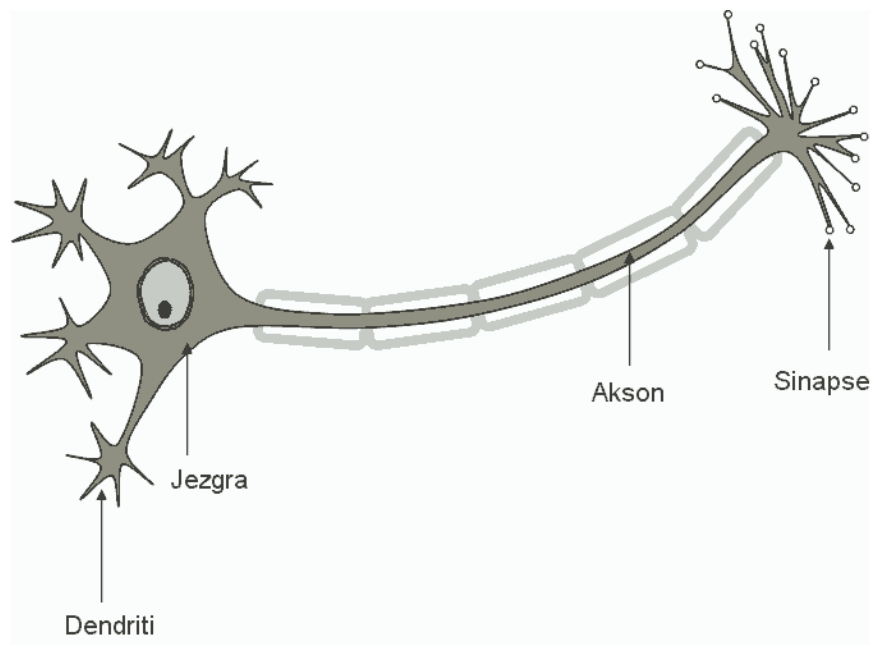
Zahvaljujem svima koji su na ikoji način doprinijeli mom obrazovanju.

# 1. Uvod

Čovjek odavno želi stvoriti neki umjetni dio sebe kako bi time otvorio neke nove mogućnosti te sebi olakšao posao. To je mnoge potaknulo na razmišljanje o umjetnom mozgu koji bi mogao uvelike nadmašiti ljudski. U tom se području u današnje vrijeme dosta intenzivno radi, ali umjetni mozak nije još ni približno dostigao ljudski potencijal. U nekim se pak područjima današnje neuronske mreže dosta uspješno koriste te pokazuju kako njihov razvoj ima dobru budućnost. Neuronske mreže rađene su na uzoru na ljudski mozak te se radi lakšeg razumijevanja isti opisuje ovdje.

## 1.1. Biološki neuron

Ljudski mozak sastoji se od mnogo neurona, procjenjenog broja preko 100 milijardi. Približan opis strukture biološkog neurona vidljiv je na slici 1.1:

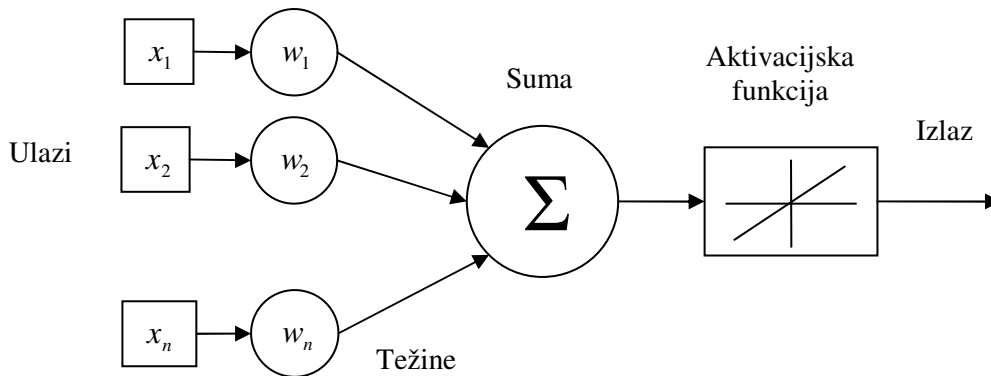


Slika 1.1. Biološki neuron

Biološki neuron sastoji se od tijela u kojem se nalazi jezgra koja upravlja procesima, dendrita koji primaju informacije od drugih neurona i aksona kroz kojeg električni impulsi putuju do završetka neurona. Tu neuron drugim neuronima šalje impulse na dendrite drugih neurona preko veza koje zovemo sinapse. Kao što se vidi na slici 1.1, neuron može imati puno dendrita i sinapsi kojima se povezuje s drugim neuronima. Jačina tih veza odlučuje koji će na njega susjedni neuroni utjecati više a koji manje te kojom će jačinom on utjecati na druge neurone.

## 1.2. Umjetni neuron

Umjetni neuron nalik je na biološki neuron, kao što se vidi na slici 1.2:



Slika 1.2. Struktura umjetnog neurona

Dendriti biološkog neurona zamijenjeni su ulazima koji preko težina ulaze u tijelo neurona koje se sastoji od sumatora i aktivacijske funkcije. Težine služe za postavljanje jačine utjecaja pojedinih ulaza na tijelo neurona, a time i na sumu. U sumatoru zbrojeni ulazi prolaze kroz aktivacijsku funkciju koja zapravo zamjenjuje akson kod biološkog neurona. Aktivacijska funkcija prema zbroju iz sumatora prosljeđuje odgovarajuću jačinu signala na izlaz.

## 1.3. Umjetne neuronske mreže

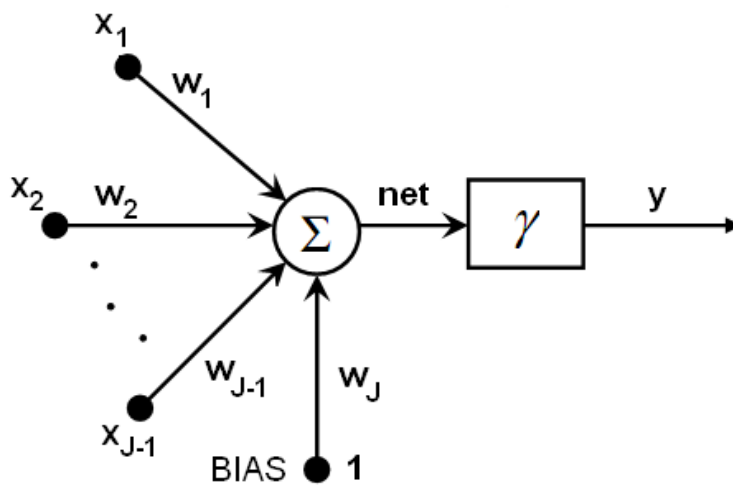
Kao što se ljudski mozak sastoji od bioloških neurona, tako se i umjetna neuronska mreža sastoji od umjetnih neurona. Neuronske mreže sastavljene su od paralelnih slojeva neurona, pri čemu su jednoslojne mreže sastavljene samo od ulaznog sloja i izlaznog sloja, koji se zapravo ne smatra slojem. Višeslojne mreže dodatno imaju skrivene slojeve koji se nalaze između ulaznog i izlaznog sloja i te nisu direktno povezani s okolinom. Signali koji putuju kroz neuronsku mrežu dijele neuronske mreže na unaprijedne (eng. feedforward neural networks) i povratne (eng. feedback or recurrent neural networks). Kroz unaprijedne neuronske mreže signali putuju samo u jednom smjeru i ne vraćaju se. Na toj se podjeli uvelike dijele problemi koje pojedina neuronska mreža može riješiti. Kako neuronska mreža zapravo uči svoje težine ovisi o algoritmu učenja. Postoji mnogo algoritama učenja od kojih svaki služi za određeni tip problema. Najčešće se koristi povratno propagiranje sa generaliziranim delta pravilom koje se može koristiti kod unaprijednih i povratnih mreža.

## 2. Statičke unaprijedne neuronske mreže

Statičke unaprijedne neuronske mreže najkorištenije su mreže koje mogu opisivati statičke te nekim dijelom i dinamičke probleme. Budući da nemaju povratnih petlji njihova izvedba i algoritam puno su jednostavniji od povratnih mreža.

### 2.1. Statički neuron i statička neuronska mreža

Statički neuron sličan je umjetnom neuronu i prikazan je na slici 2.1:



Slika 2.1. Statički neuron

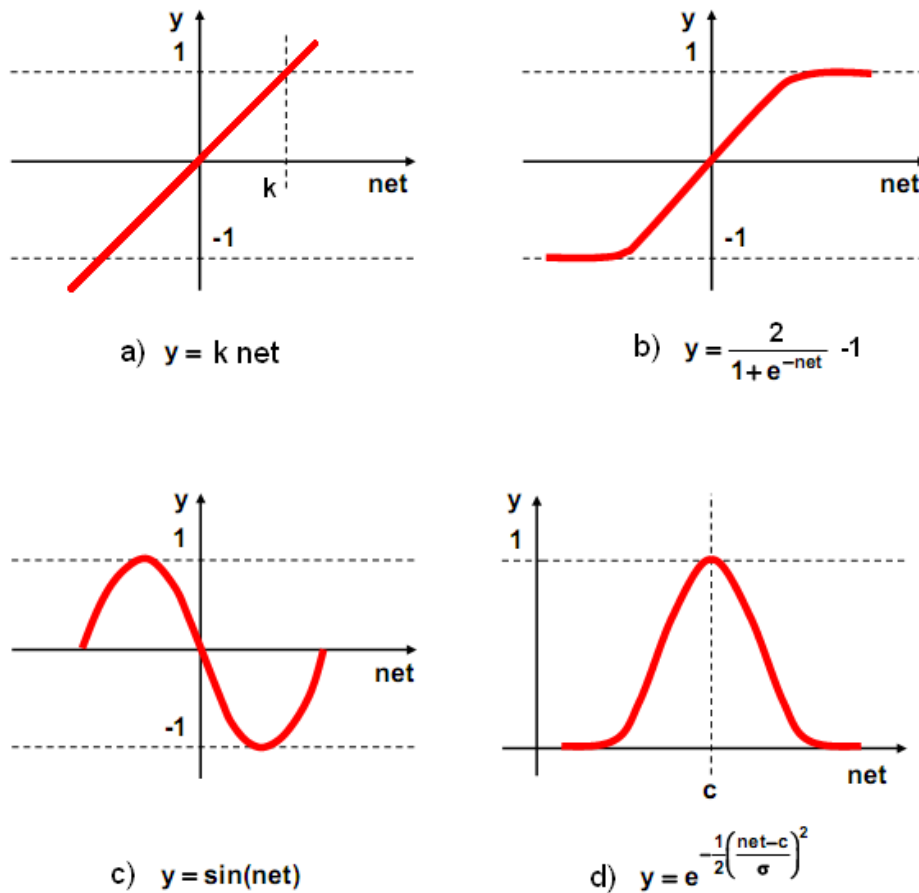
Sastoji se od više ulaza, funkcije sume, aktivacijske funkcije te jednog izlaza. Pritom je dodan još jedan jedinični neuron označen sa BIAS uvijek iste vrijednosti jednake jedinici, koji je zapravo potreban kako bi mreža uopće mogla učiti. Funkcija sume predstavlja zbroj *net* svih umnožaka vrijednosti ulaznih neurona i pripadajućih težina, dano izrazom (2-1).

$$net = \sum_{j=1}^J x_j w_j \quad (2-1)$$

Aktivacijska funkcija  $\gamma$  preslikava vrijednost zbroja *net* na izlaz *y* prema izrazu (2-2).

$$y = \gamma(net) \quad (2-2)$$

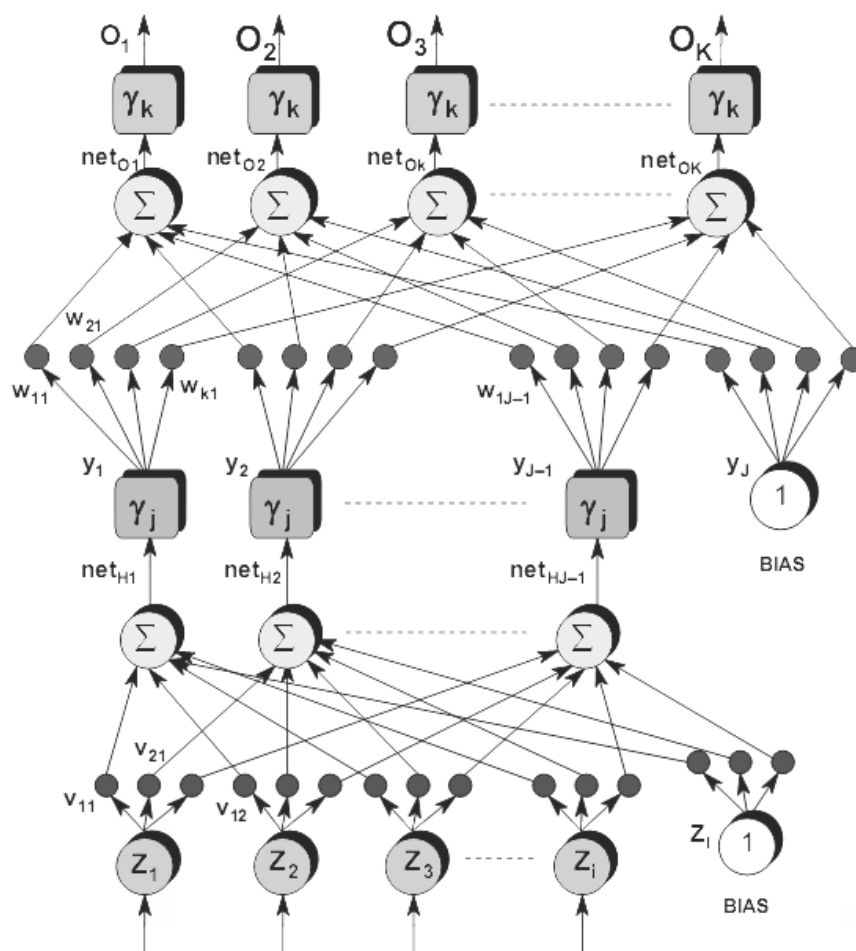
Aktivacijske funkcije koje se najčešće koriste [1] prikazane su na slici 2.2:



Slika 2.2. Najčešće korištene aktivacijske funkcije

Monotono rastuće funkcije prikazane su na slikama 2.2.a. i 2.2.b. Na slici 2.2.a. prikazana je obična linearna funkcija nagiba pravca  $k$  dok je na slici 2.2.b. prikazana bipolarna sigmoidalna aktivacijska funkcija koja je najčešće u upotrebi. Primjer nemonotonih rastućih funkcija su sinusna funkcija na slici 2.2.c. i gaussova na slici 2.2.d., koje se rjeđe koriste. Kao što se vidi slici 2.2. sve funkcije imaju ograničenu vrijednost između -1 i 1, pa je zbog toga preporučljivo ulazne i izlazne podatke normirati unutar tih vrijednosti.

Za dobivanje statičke neuronske mreže, statički neuroni slažu se u slojeve. Najčešći oblik mreže je sa tri sloja, ulaznim, skrivenim i izlaznim. U ulaznom sloju broj neurona jednak je broju ulaza uvećan za BIAS. Broj neurona izlaznog sloja jednak je broju izlaza, dok se u skrivenom sloju nalaze neuroni koji nemaju direktne veze s okolinom, pa njihov broj nije uvjetovan okolinom. Skrivenih slojeva može biti više i svaki od njih ima dodan BIAS. Model statičke neuronske mreže prikazan je na slici 2.3.



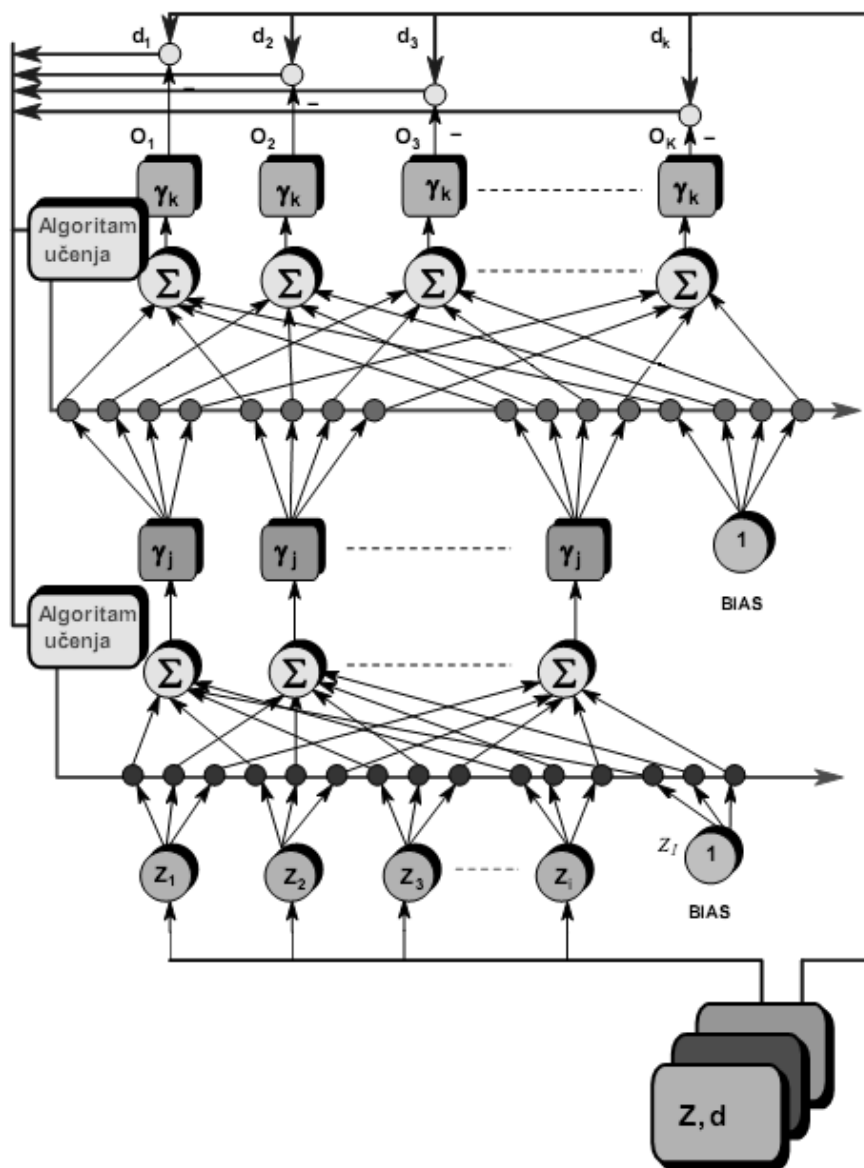
Slika 2.3. Statička unaprijedna neuronska mreža

Statička neuronska mreža na slici 2.3. naziva se statička unaprijedna neuronska mreža budući da svi signali putuju od ulaza prema izlazu. Neuroni ulaznog sloja ( $Z_i$ ) zapravo su ulazi sa dodanim BIAS-om ( $Z_i$ ) koji preko pripadajućih težina skrivenog sloja ( $v_{ji}$ ) ulaze u sumator skrivenog sloja. Zbrojevi iz sumatora ( $net_{Hj-1}$ ) ulaze u aktivacijske funkcije ( $\gamma_j$ ) koje daju vrijednosti na izlazima skrivenog sloja ( $y_{j-1}$ ). Nakon toga preko težina izlaznog sloja ( $w_{kj}$ ) signali putuju u sumatore izlaznog sloja i preko aktivacijskih funkcija izlaznog sloja ( $\gamma_k$ ) dolaze na izlaze ( $O_k$ ).

## 2.2. Učenje statičke neuronske mreže

Učenje statičke neuronske mreže svodi se zapravo na podešavanje težina skrivenog i izlaznog sloja. To se postiže tako da se na ulaze mreže dovode ulazne vrijednosti ( $Z_i$ ) a na izlaze željeni rezultati ( $d_k$ ). Tijekom učenja mreže dobiveni rezultati uspoređuju se sa

željenima te se na temelju odabranog algoritma učenja težine skrivenog i izlaznog sloja podešavaju tako dugo dok ne zadovoljavaju rezultate unutar neke granice greške. Budući da se na ulaze i izlaze dovodi više različitih uzoraka vrijednosti, algoritam učenja mijenja težine na dva načina. Prvi način je učenje po uzorku gdje se nakon svakog uzorka računa greška i na temelju te greške podešavaju težine. Drugi način je učenje po skupu koji za razliku od učenja po uzorku podešava težine tek nakon prolaska kroz sve uzorke i zbrajanja svih grešaka u jednu preko koje se nakon jednog koraka učenja podešavaju težine. To znači da za jedan korak učenja po skupu učenje po uzorku izvrši za broj uzoraka puta više koraka. Prikaz učenja pokazan je na slici 2.4:



Slika 2.4. Učenje statičke neuronske mreže



## 2.2.1. Unaprijedna faza učenja statičke neuronske mreže

U unaprijednoj fazi učenja potrebno je na temelju zadanih ulaza izračunati izlaze. Kako bi to bilo moguće, potrebno je na početku podesiti sve težine zadane neuronske mreže. U većini slučajeva najbolje je početne težine odabrati nasumično pomoću generatora nasumičnih brojeva i to u rasponu od -1 do 1. Nakon toga pristupa se računanju izlaza. Najprije se iz skupa učenja odabire uzorak (ulazno-izlazni par  $Z, d$ ) te se njegove ulazne vrijednosti ( $Z$ ) postavljaju na neurone ulaznog sloja. Nakon toga slijedi računanje izlaza skrivenog sloja.

### Skriveni sloj

Funkcija sume skrivenog sloja označava se sa  $net_H$  i dobiva pripadajući indeks za svaki skriveni  $j$ -ti neuron te se računa prema izrazu (2-3):

$$net_{Hj} = \sum_{i=1}^I v_{ji} Z_i, \quad j=1,2,\dots,J-1, \quad i=1,2,\dots,I, \quad (2-3)$$

gdje je  $J$  broj skrivenih neurona uvećan za jedan (BIAS), a  $I$  jednak broju ulaznih neurona. Nakon funkcije sume slijedi aktivacijska funkcija koja može biti jedna od četiri navedene funkcije na slici 2.2 ili neka druga. U ovom slučaju biti će prikazane navedene četiri. Izrazi za računanje izlaza neurona skrivenog sloja su: za linearnu (2-4), za sigmoidalnu (2-5), za sinusnu (2-6) i za gaussovu (2-7):

$$y_j = k_p \cdot net_{Hj}, \quad (2-4)$$

$$y_j = \frac{2}{1 + e^{-net_{Hj}}} - 1, \quad (2-5)$$

$$y_j = \sin(net_{Hj}), \quad (2-6)$$

$$y_j = e^{-\frac{1}{2} \left( \frac{net_{Hj} - c}{\sigma} \right)^2}, \quad j=1,2,\dots,J-1, \quad (2-7)$$

$$y_J = 1, \text{ BIAS}. \quad (2-8)$$

Na kraju se prema izrazu (2-8) dodaje i BIAS na kraj skrivenog sloja.

### Izlazni sloj

Izračunati izlazi skrivenog sloja u ovom dijelu preko težina izlaznog sloja ( $w_{kj}$ ) dolaze u sumator izlaznog sloja. Funkcija sume izlaznog sloja sada dobiva oznaku i indeks  $net_o$  i drugi indeks izlaza na koji se funkcija sume odnosi, te se računa prema izrazu (2-9):

$$net_{ok} = \sum_{j=1}^J v_{kj} y_j \quad , k=1,2,\dots,K \quad , \quad (2-9)$$

gdje je  $K$  broj izlaza. Kao i kod skrivenog sloja, i ovdje će biti prikazane četiri navedene aktivacijske funkcije. Izrazi su potpuno analogni, za linearnu (2-10), za sigmoidalnu (2-11), za sinusnu (2-12) i za gaussovu (2-13):

$$O_k = k_p \cdot net_{ok} \quad , \quad (2-10)$$

$$O_k = \frac{2}{1 + e^{-net_{ok}}} - 1, \quad (2-11)$$

$$O_k = \sin(net_{ok}) \quad , \quad (2-12)$$

$$O_k = e^{-\frac{1}{2} \left( \frac{net_{ok} - c}{\sigma} \right)^2} \quad , \quad j=1,2,\dots,K \quad . \quad (2-13)$$

### 2.2.2. Povratna faza učenja statičke neuronske mreže

Nakon unaprijedne faze slijedi povratna faza u kojoj se podešavaju težine slojeva. Kako bi se došlo do pogreške učenja, vrijednosti izlaza dobivene unaprijednom fazom ( $O$ ) uspoređuju se sa željenim vrijednostima ( $d$ ) iz uzorka iz kojeg su uzeti ulazni podaci. Za podatak o odstupanju najčešće se koristi suma kvadrata pogreške [1] definirana izrazom:

$$E = \frac{1}{2} \sum_{n=1}^N (d_n - O_n)^2 \quad , \quad (2-14)$$

pri čemu je  $N$  broj uzoraka iz skupa za učenje. Ta funkcija je zapravo funkcija cilja koju algoritam učenja tijekom učenja mora minimizirati do zadane vrijednosti mijenjanjem težina.

Promjena parametara učenja  $\vartheta$  (težina) dana je izrazom (2-15):

$$\vartheta(n+1) = \vartheta(n) + \Delta\vartheta(n) , \quad (2-15)$$

pri čemu je  $\vartheta$  za skriveni sloj jednaka  $v$ , a za izlazni sloj jednaka  $w$ .  $\Delta\vartheta$  je promjena parametara učenja dok je  $n$  trenutni korak učenja. Pogreška  $E(\vartheta)$  može se u blizini točke  $\vartheta$  aproksimirati s prva dva člana Taylorovog reda [1] prema sljedećim izrazima:

$$E(\vartheta + \Delta\vartheta) \approx E(\vartheta) + \Delta E(\vartheta) , \quad (2-16)$$

$$\Delta E(\vartheta) = \Delta\vartheta^T \nabla E(\vartheta) , \quad (2-17)$$

$$\nabla E(\vartheta) = \frac{\partial E(\vartheta)}{\partial \vartheta} , \quad (2-18)$$

pri čemu se izraz (2-18) naziva gradijent pogreške. Za smanjivanje greške najvećim mogućim iznosom  $\Delta\vartheta$  prema izrazu (2-19) treba biti jednaka:

$$\Delta\vartheta = -\eta \nabla E(\vartheta) , \quad (2-19)$$

pri čemu je  $\eta$  koeficijent brzine učenja koji se korisnički odabire i kreće između 0.001 do 10 [1]. Uvrštavanjem izraza (2-19) u izraz (2-15) dobiva se konačan izraz (2-20) za promjenu parametara učenja:

$$\vartheta(n+1) = \vartheta(n) - \eta \nabla E(\vartheta(n)) . \quad (2-20)$$

Taj algoritam učenja poznat je pod nazivom *algoritam povratnog prostiranja greške*, skraćeno EBP (eng. Error Back-Propagation Algorithm) te se najčešće primjenjuje kod promjena težinskih koeficijenata.

## Promjena težina izlaznog sloja

Povratna faza učenja mreže za razliku od unaprijedne faze počinje od izlaznog sloja prema ulaznom. Težine izlaznog sloja prema izrazu (2-20) mijenjaju se preko sljedećeg izraza:

$$w_{kj}(n+1) = w_{kj}(n) - \eta \nabla E(n) , \quad (2-21)$$

pri čemu se gradijent pogreške za izlazne težine  $w_{kj}$  računa prema izrazu (2-18):

$$\nabla E(n) = \frac{\partial E(n)}{\partial w_{kj}} . \quad (2-22)$$

Kako bi se zadani gradijent mogao izračunati, parcijalna derivacija rastavlja se na tri uzastopne parcijalne derivacije prema sljedećem izrazu (2-23):

$$\frac{\partial E(n)}{\partial w_{kj}} = \frac{\partial E(n)}{\partial O_k} \frac{\partial O_k}{\partial net_{Ok}} \frac{\partial net_{Ok}}{\partial w_{kj}} . \quad (2-23)$$

Prvi i treći član izraza (2-23) određuju iz izraza (2-14) i (2-9) i oni su jednaki za sve aktivacijske funkcije:

$$\frac{\partial E(n)}{\partial O_k} = -(d_k - O_k) , \quad (2-24)$$

$$\frac{\partial net_{Ok}}{\partial w_{kj}} = y_j . \quad (2-25)$$

Drugi član ovisi o tipu aktivacijske funkcije i određuje se za sve četiri navedene aktivacijske funkcije iz izraza (2-10), (2-11), (2-12) i (2-13):

$$\frac{\partial O_k}{\partial net_{Ok}} = k_p , \quad (2-26)$$

$$\frac{\partial O_k}{\partial net_{Ok}} = \frac{1}{2}(1 - O_k^2) , \quad (2-27)$$

$$\frac{\partial O_k}{\partial net_{Ok}} = \cos(net_{Ok}) , \quad (2-28)$$

$$\frac{\partial O_k}{\partial net_{Ok}} = \frac{O_k}{s^2}(c - net_{Ok}) . \quad (2-29)$$

Konačni gradijent pogreške za sve vrste aktivacijskih funkcija računa se prema izrazu (2-23):

$$\nabla E(n) = -(d_k - O_k) \frac{\partial O_k}{\partial net_{Ok}} y_j, \quad (2-30)$$

Konačni izraz za promjenu težina izlaznog sloja prema izrazu (2-21) glasi:

$$w_{kj}(n+1) = w_{kj}(n) + \eta (d_k - O_k) y_j \frac{\partial O_k}{\partial net_{Ok}}, \quad (2-31)$$

pri čemu zadnji član ovisi o aktivacijskoj funkciji izlaznog sloja te se računa iz izraza (2-26) za linearnu, (2-27) za sigmoidalnu, (2-28) za sinusnu i (2-29) za gaussovu funkciju.

### Promjena težina skrivenog sloja

Nakon promjena težina izlaznog sloja slijedi promjena težina skrivenog sloja. Izraz za promjenu težina skrivenog sloja analogan je izrazu (2-21):

$$v_{ji}(n+1) = v_{ji}(n) - \eta \nabla E(n). \quad (2-32)$$

Kao i kod izlaznog sloja potrebno je izračunati gradijent greške, pa se analogno i ovdje rastavlja na tri uzastopne parcijalne derivacije:

$$\nabla E(n) = \frac{\partial E(n)}{\partial y_j} \frac{\partial y_j}{\partial net_{Hj}} \frac{\partial net_{Hj}}{\partial v_{ji}}, \quad (2-33)$$

Svaka težina skrivenog sloja povezana je sa svim neuronima izlaznog sloja pa je prva parcijalna derivacija izraza (2-33) jednaka:

$$\frac{\partial E(n)}{\partial y_j} = \sum_{k=1}^K \frac{\partial E(n)}{\partial O_k} \frac{\partial O_k}{\partial net_{Ok}} \frac{\partial net_{Ok}}{\partial y_j}, \quad (2-34)$$

pri čemu su prvi i treći član desne strane izraza (2-34) dobiveni iz izraza (2-9) i (2-14) te jednaki:

$$\frac{\partial E(n)}{\partial O_k} = -(d_k - O_k), \quad (2-35)$$

$$\frac{\partial net_{Ok}}{\partial y_j} = w_{kj}, \quad (2-36)$$

dok drugi član ovisi o vrsti aktivacijske funkcije te se odabire za navedene funkcije na slici 2.2 iz izraza (2-26) za linearnu, (2-27) za sigmoidalnu, (2-28) za sinusnu i (2-29) za gaussovu. Drugi član uzastopnih parcijalnih derivacija iz izraza (2-33) ovisi o vrsti aktivacijske funkcije skrivenog sloja, analogno kao i za izlazni sloj. Za odabrane četiri aktivacijske funkcije drugi član se dobiva iz izraza (2-4), (2-5), (2-6) i (2-7) jednak je:

$$\frac{\partial y_j}{\partial net_{Hj}} = k_p, \quad (2-37)$$

$$\frac{\partial y_j}{\partial net_{Hj}} = \frac{1}{2}(1 - y_j^2), \quad (2-38)$$

$$\frac{\partial y_j}{\partial net_{Hj}} = \cos(net_{Hj}), \quad (2-39)$$

$$\frac{\partial y_j}{\partial net_{Hj}} = \frac{y_j}{s^2}(c - net_{Hj}). \quad (2-40)$$

Treći član analogan je članu kod izlaznog sloja, a dobiva se iz izraza (2-3) i jednak je:

$$\frac{\partial net_{Hj}}{\partial v_{ji}} = Z_i. \quad (2-41)$$

Konačan izraz za promjenu težina skrivenog sloja prema izrazu (2-32) glasi:

$$v_{ji}(n+1) = v_{ji}(n) + \eta \left[ \sum_{k=1}^K (d_k - O_k) \frac{\partial O_k}{\partial net_{Ok}} w_{kj} \right] \frac{\partial y_j}{\partial net_{Hj}} Z_i, \quad (2-42)$$

pri čemu se  $\frac{\partial O_k}{\partial net_{Ok}}$  odabire kao i kod izraza (2-31) prema aktivacijskoj funkciji izlaznog sloja

dok se  $\frac{\partial y_j}{\partial net_{Hj}}$  odabire prema aktivacijskoj funkciji skrivenog sloja iz izraza (2-37) za

linearnu, (2-38) za sigmoidalnu, (2-39) za sinusnu te (2-40) za gaussovu funkciju.

## 2.3. Modifikacije algoritma učenja statičke neuronske mreže

Algoritam učenja s povratnim prostiranjem greške (EBP) u svom klasičnom obliku zahtjeva puno koraka učenja. Za ubrzanje učenja klasični EBP algoritam može se modificirati na više načina, u ovom slučaju bit će objašnjena modifikacija sa momentumom prvog reda te metoda adaptacije brzine učenja prema Silvi i Almeidi.

### 2.3.1. Momentum (zamah) prvog reda

Budući da je odabrana brzina učenja tijekom učenja stalna, kako bi se ubrzao proces koristi se momentum. Njegovim ubacivanjem izraz (2-19) proširuje se još jednim članom te se dobiva:

$$\Delta \vartheta = -\eta \nabla E(\vartheta) + \alpha \Delta \vartheta(n-1) , \quad (2-43)$$

pri čemu  $\alpha$  označava koeficijent momentuma koji se također korisnički određuje i kreće se od 0.1 do 0.9 [1] dok je  $\Delta \vartheta(n-1)$  promjena težina u prethodnom koraku koju algoritam tijekom učenja treba pamtit. Konačan izraz za promjenu težina glasi:

$$\vartheta(n+1) = \vartheta(n) - \eta \nabla E(\vartheta(n)) + \alpha \Delta \vartheta(n-1) . \quad (2-44)$$

Ovom modifikacijom EBP algoritam zadržava svoj postupak računanja s jedinom razlikom što izrazi (2-21) i (2-32) dobivaju novi član te glase:

$$w_{kj}(n+1) = w_{kj}(n) - \eta \nabla E(n) + \alpha \Delta \vartheta(n-1) , \quad (2-45)$$

$$v_{ji}(n+1) = v_{ji}(n) - \eta \nabla E(n) + \alpha \Delta \vartheta(n-1) . \quad (2-46)$$

### 2.3.2. Metoda Silva i Almeida

Za razliku od modifikacije momentumom, kod metode Silva i Almeida ne dodaje se nikakav član nego se brzina učenja dinamički mijenja za svaku težinu posebno prema [2]:

$$\eta_{ij}(n) = \left\{ \begin{array}{ll} \eta_{ij}(n-1) * u & \text{if } \frac{\partial E}{\partial v_{ij}}(n) * \frac{\partial E}{\partial v_{ij}}(n-1) \geq 0 \\ \eta_{ij}(n-1) * d & \text{else} \end{array} \right\}, \quad (2-47)$$

pri čemu su  $u$  i  $d$  koeficijenti približnog odnosa  $u \approx 1/d$ . Vrijednost  $u$  tipično je između 1.1 i 1.3 dok je vrijednost  $d$  između 0.7 i 0.9. Prema navedenom izrazu (2-47) sve dok gradijent pogreške ne mijenja predznak, brzina učenja se povećava. Ako se nakon učenja greška povećala, sve brzine učenja se raspolavljaju, čime se ograničava eksponencijalni rast brzina. Pritom treba naglasiti kako ova metoda može biti uvedena u klasični EBP algoritam kao i u modificirani EBP algoritam sa momentumom. Ako se metoda Silva i Almeida uvede u EBP algoritam sa momentumom, izrazi za promjenu težina izlaznog sloja prema izrazu (2-45) glasi:

$$w_{kj}(n+1) = w_{kj}(n) - \eta_{kj}(n) \nabla E(n) + \alpha \Delta v(n-1), \quad (2-48)$$

pri čemu se brzina učenja ( $\eta_{kj}$ ) pojedine težine izlaznog sloja mijenja prema izrazu (2-47):

$$\eta_{kj}(n) = \left\{ \begin{array}{ll} \eta_{kj}(n-1) * u & \text{if } \frac{\partial E}{\partial w_{kj}}(n) * \frac{\partial E}{\partial w_{kj}}(n-1) \geq 0 \\ \eta_{kj}(n-1) * d & \text{else} \end{array} \right\}, \quad (2-49)$$

Za EBP algoritam bez momentuma jednostavno se izostavlja treći član izraza (2-48) ili se postavi da je momentum jednak nuli ( $\alpha = 0$ ). Izrazi za promjenu težina skrivenog sloja dodavanjem metode promjene brzine učenja Silva i Almeida prema izrazu (2-46) glase:

$$v_{ji}(n+1) = v_{ji}(n) - \eta_{ji}(n) \nabla E(n) + \alpha \Delta v(n-1), \quad (2-50)$$



pri čemu se brzina učenja ( $\eta_{ji}$ ) pojedine težine skrivenog sloja mijenja prema izrazu (2-47):

$$\eta_{ji}(n) = \left\{ \begin{array}{ll} \eta_{ji}(n-1) * u & \text{if } \frac{\partial E}{\partial v_{ji}}(n) * \frac{\partial E}{\partial v_{ji}}(n-1) \geq 0 \\ \eta_{ji}(n-1) * d & \text{else} \end{array} \right\}. \quad (2-51)$$

Kao kod izraza za promjenu težina izlaznog sloja tako se i kod skrivenog sloja momentum može ili ne mora koristiti.

Radi usporedbe navedenih algoritama učenja, u sljedećem poglavlju biti će opisane neke primjene statičkih neuronskih mreža.

### 3. Primjena statičkih neuronskih mreža

Statičke neuronske mreže najčešće se primjenjuju kod statičkih procesa, ali se uz određena podešavanja mogu koristiti i za dinamičke linearne i nelinearne procese. Navedeni primjeri pokazat će kako statička neuronska mreža može poslužiti za klasifikaciju i identifikaciju određenih problema. Svi primjeri rađeni su u Matlabu u batch proceduri te imaju normirani skup podataka za učenje.

#### 3.1. Klasifikacija

Za problem klasifikacije odabran je ekskluzivni ILI (eng. XOR) problem. Ulazni i izlazni skupovi podataka navedenog problema su sljedeći:

$$Z = \begin{bmatrix} 0 & 0 \\ 0 & 1 \\ 1 & 0 \\ 1 & 1 \end{bmatrix}, \quad (3-1)$$

$$d = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}. \quad (3-2)$$

Testiranje je provedeno sa tri tipa učenja i tri vrste aktivacijske funkcije skrivenog sloja. Aktivacijska funkcija izlaznog sloja za sva učenja je linearna sa koeficijentom smjera jednakim jedan. Početne težine odabrane su nasumično između -1 i 1, dok je za algoritam učenja Silva i Almeida brzina učenja pet puta manja. Parametri učenja prikazani su u tablici:

Algoritam/parametri	Brzina	Momentum	Koeficijent $u$	Težine $v, w$
EBP	0,2	0	N/A	-1 : 1
EBP+momentum	0,2	0,7	N/A	-1 : 1
EBP+Silva i Almeida	0,04	0	1,3	-1 : 1

Tablica 3.1. Parametri učenja za XOR problem

Razlog zašto je početna brzina učenja po Silva i Almeida metoda pet puta manja je u eksponencijalnom rastu brzine nakon svakog koraka učenja za koeficijent  $u$ . Odabere li se prevelika brzina, učenje može krenuti u krivom smjeru i za njegov proces može biti potrebno čak i više koraka nego kod smanjenih parametara. Točnost algoritama učenja može se mjeriti na više načina, u ovom slučaju u svim navedenim primjerima mjeri se preko normaliziranog korijena srednje kvadratne pogreške (eng. NRMS):

$$NRMS = \frac{RMS}{\sigma_{d_n}} = \frac{\sqrt{\frac{\sum_{n=1}^N (d_n - O_n)^2}{N}}}{\sigma_{d_n}}, \quad (3-3)$$

pri čemu su:

$$\sigma_{d_n} = \sqrt{\frac{1}{N} \sum_{n=1}^N (d_n - \bar{d})^2}, \quad (3-4)$$

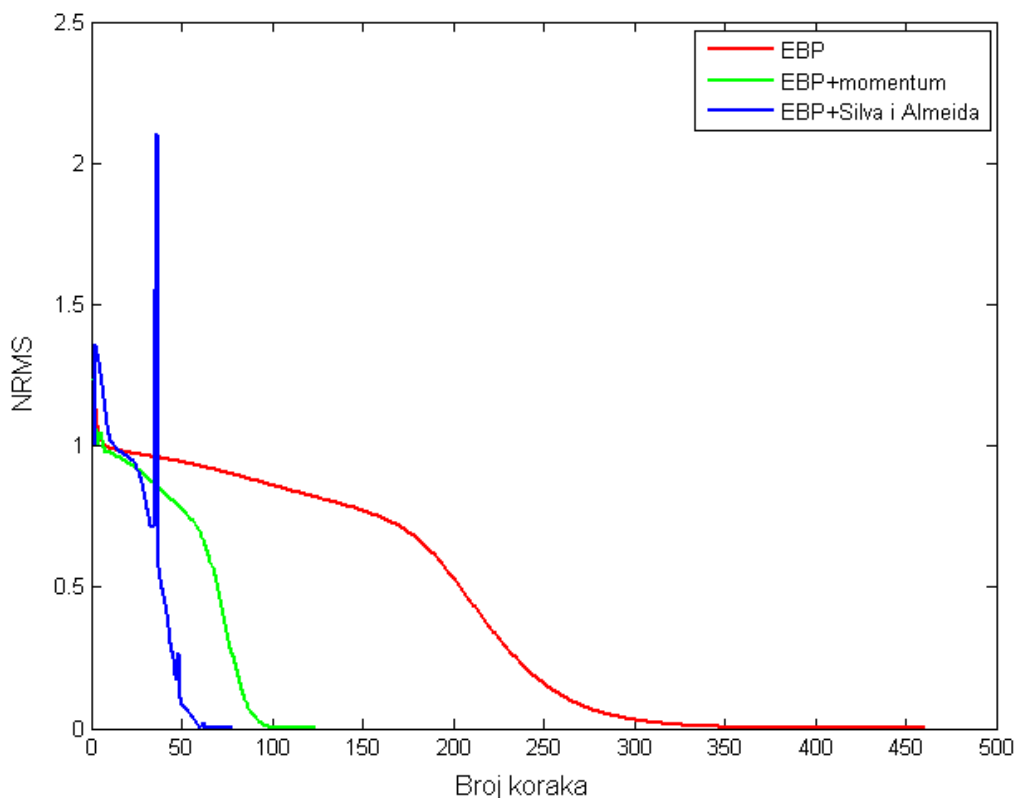
$$\bar{d} = \frac{1}{N} \sum_{n=1}^N d_n. \quad (3-5)$$

Učenje se provodi po batch proceduri (učenje po skupu) sa dva skrivena neurona i konačne pogreške NRMS-a od 0,0001. Linearna aktivacijska funkcija u skrivenom sloju se ne koristi jer se njome dokazano testiranjima ne dolazi do rješenja, što i ne čudi jer je problem linearno neseparabilan. Nakon provedenih učenja, rezultati su prikazani u tablici 3.2. Kao što se vidi u tablici, najmanji broj koraka potreban za učenje do minimalne greške od 0,0001 postiže algoritam sa Silva i Almeida metodom. EBP algoritam sa momentumom prvog reda približno je dvostruko sporiji dok je klasični EBP algoritam sporiji za tri do deset puta. Sa stajališta aktivacijskih funkcija, sigmoidalna i sinusna funkcija daju najbrže učenje i pri tome čak smanjuju kašnjenje sporijih algoritama učenja. Gaussova funkcija predstavlja tvrđi orah za sve algoritme učenja, posebno za sporije. Trajanje svih učenja zapravo ovisi o početnim težinama; postavte li se težine na početku učenja vrlo blizu rješenja, rezultat će biti mali broj potrebnih koraka, kao u primjeru drugog učenja sinus funkcije iz tablice 3.2.

Akt. funkcija	Algoritam	Broj koraka				Prosječni broj koraka
		Učenje 1	Učenje 2	Učenje 3	Učenje 4	
Sigmoid	EBP	323	414	412	461	402
	EBP+M	77	110	95	124	101
	EBP+SA	40	43	58	77	54
Sinus	EBP	347	70	82	195	173
	EBP+M	130	53	56	75	78
	EBP+SA	84	28	42	71	56
Gauss	EBP	826	732	604	650	703
	EBP+M	258	217	183	185	211
	EBP+SA	77	82	89	68	79

**Tablica 3.2** Rezultati učenja za XOR problem

Na slici 3.1 prikazano je četvrto učenje sigmoidalne funkcije. Kao što se može primijetiti, klasični EBP algoritam i EBP sa momentumom u početku postižu manju pogrešku ali uslijed eksponencijalnog povećanja brzine pogreška algoritma sa Silva i Almeida metodom počinje naglo padati te se sa slike može vidjeti kako je pri tom padu algoritam naišao na dva veća lokalna minimuma (dva skoka pogreške). Sva ostala učenja približno odgovaraju prikazanom.



**Slika 3.1.** Smanjivanje pogreške tijekom učenja XOR problema

### 3.2. Identifikacija linearnog dinamičkog sustava

Za primjer jednostavnog linearnog dinamičkog sustava odabran je proporcionalan član prvog reda P1 čija je dinamika opisana diferencijalnom jednačinom prvog reda:

$$T\dot{x}(t) + x(t) = K_p u(t) , \quad (3-6)$$

pri čemu je  $T$  vremenska konstanta a  $K_p$  konstanta pojačanja sustava. Za potrebe učenja neuronske mreže potrebna je njegova diskretna forma:

$$x(n) = \frac{T_0}{T + T_0} \left[ \frac{T}{T_0} x(n-1) + K_p u(n) \right] , \quad (3-7)$$

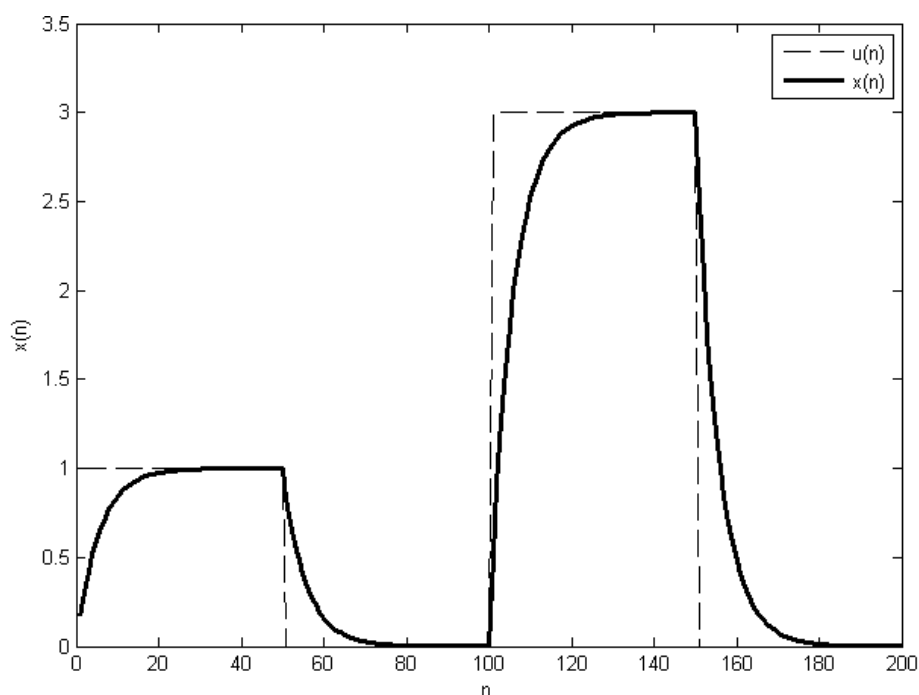
pri čemu su  $x(n)$  i  $u(n)$  trenutne vrijednosti izlazne i ulazne varijable,  $x(n-1)$  je prethodna vrijednost izlazne veličine, a  $T_0$  vrijeme uzorkovanja. Prema [1] odabrane vrijednosti parametara su:

$$\begin{aligned} T &= 1 \text{ s} , \\ T_0 &= 0,2 \text{ s} , \\ K_p &= 1 . \end{aligned} \quad (3-8)$$

Za učenje ovog problema odabrano je dvjesto točaka ( $n=200$ ) i ulaznom varijablom određenih vrijednosti u odabranim intervalima:

$$\begin{aligned} u(n) &= 1, & n &= 1, 2, \dots, 50 \\ u(n) &= 0, & n &= 51, 52, \dots, 100 \\ u(n) &= 3, & n &= 101, 102, \dots, 150 \\ u(n) &= 0, & n &= 151, 152, \dots, 200 \end{aligned} \quad (3-9)$$

Odziv sustava na odabranu pobudnu funkciju iz (3-9) prikazan je na slici 3.2. Tih dvjesto točaka predstavlja skup ulaznih i izlaznih podataka učenja.



**Slika 3.2.** Skup uzoraka za učenje P1 člana

Početni parametri podešeni su slično kao i kod XOR problema. Neke vrijednosti mreža povećane se kako bi se dobila manja pogreška NRMS-a za odabrani broj koraka. Kao poseban primjer izdvajaju se početne težine algoritma sa Silva i Almeida metodom koje su pet puta manje kao i brzina učenja, s svrhom da algoritam ne dođe brzo lokalni minimum. Parametri su prikazani u sljedećoj tablici:

Algoritam/parametri	Brzina	Momentum	Koeficijent $u$	Težine $v, w$
EBP	0,3	0	N/A	-1 : 1
EBP+momentum	0,3	0,7	N/A	-1 : 1
EBP+Silva i Almeida	0,06	0	1,25	-0,2 : 0,2

**Tablica 3.3.** Parametri učenja za linearni dinamički sustav

Kod ovih učenja korištene su četiri vrste aktivacijskih funkcija skrivenog sloja sa dva neurona i linearna aktivacijska funkcija izlaznog sloja. Sva učenja bit će ograničena na 3000 koraka i pogrešku NRMS-a od 0,0001. Rezultati učenja prikazani su u tablici 3.4.

Akt. funkcija	Algoritam	Učenje 1		Učenje 2	
		Broj koraka	NRMS	Broj koraka	NRMS
Linear	EBP	302	0,000098	351	0,000098
	EBP+M	60	0,000084	118	0,000095
	EBP+SA	85	0,000074	99	0,000095
Sigmoid	EBP	3000	0,095	3000	0,016
	EBP+M	3000	0,017	3000	0,013
	EBP+SA	3000	0,0111	3000	0,012
Sinus	EBP	3000	0,0188	3000	0,013
	EBP+M	3000	0,0106	3000	0,01
	EBP+SA	3000	0,0128	3000	0,012
Gauss	EBP	3000	0,028	3000	0,028
	EBP+M	3000	0,0194	3000	0,018
	EBP+SA	3000	0,0185	3000	0,021

**Tablica 3.4.** Rezultati učenja za linearni dinamički sustav

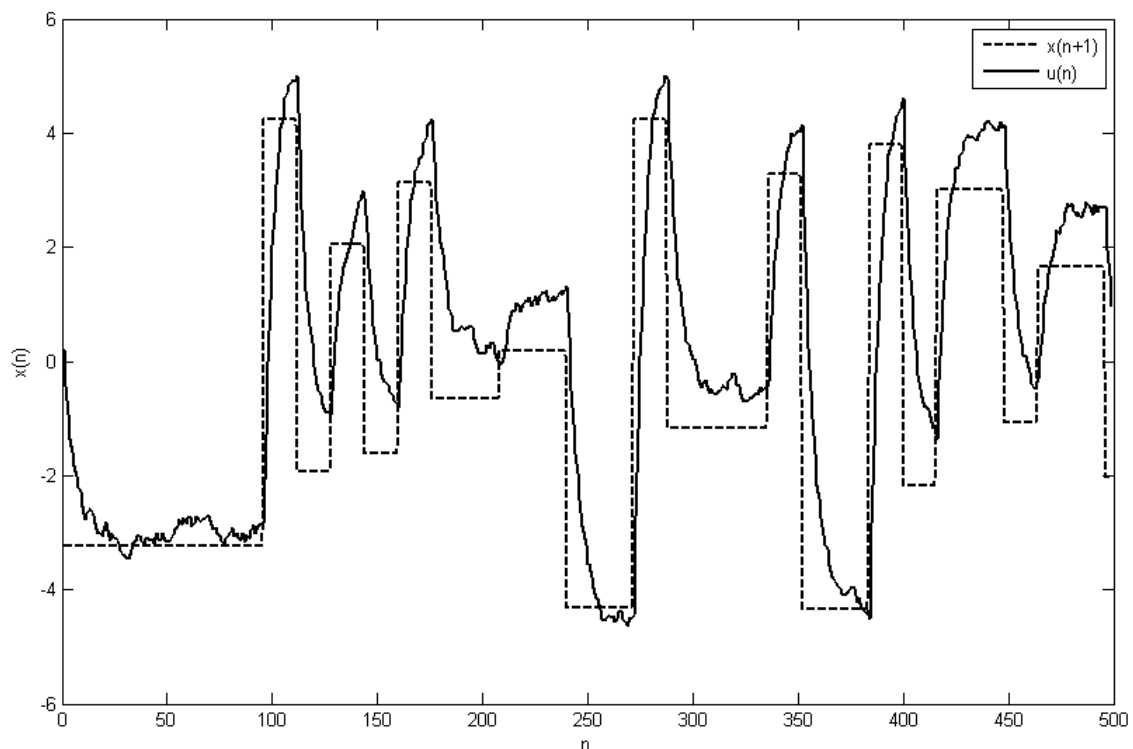
Gledanjem rezultata odmah se može zaključiti kako je prema testiranim mrežama najbolja ona kojoj su aktivacijske funkcije skrivenog i izlaznog sloja linearne. Druge testirane kombinacije ne mogu ni blizu opisati zadani proces, što nije iznenađujuće, budući da je sustav linearnog karaktera. Sa stajališta algoritama učenja može se zaključiti prema rezultatima testiranja kako su najbrže funkcije modificirane funkcije. Modifikacija sa Silva i Almeida metodom još je jednom pokazala kako dobro ubrzava učenje mreže. Također se mora spomenuti kako je taj algoritam korišten bez momentuma pa zato algoritam sa momentumom nekad postiže bolje rezultate. Testiranje naučene mreže u ovom se slučaju mora izvoditi na drugačiji način, zato jer je jedan ulaz u sustav prethodna vrijednost izlaza. To se postiže pamćenjem izlaza te postavljanjem na ulaz u sljedećem koraku. Budući da naučena mreža sa linearnim aktivacijskim funkcijama gotovo potpuno uklanja grešku, testiranja s promijenjenim amplitudama pokazuju kako naučena mreža dobro generalizira zadani proces. Prikazivanje navedenih testova nema smisla jer se iz slika gotovo i ne može razlučiti koja je krivulja prava a koja je rezultat naučene mreže.

### 3.3. Identifikacija nelinearnog dinamičkog sustava

Sljedeći primjer identifikacije je nelinearni dinamički sustav opisan nelinearnom jednačbom diferencije prvog reda uz period uzorkovanja jedne sekunde i vremenske konstante od deset sekundi [1]:

$$x(n+1) = (0.83 - 0.003x(n))x(n) + 0,2u(n) . \quad (3-10)$$

Da bi se sustav mogao dobro identificirati potreban je kvalitetan ulazni signal. U svrhu toga za ulazni signal biti će odabran pseudo-binarni signal slučajne frekvencije i slučajne amplitude unutar granica od -5 do 5. Kako bi se dobio realniji odziv na izlazni signal pridodan je slučajan šum amplitude od -0.3 do 0.3. Prvih 500 točaka simuliranog sustava prikazano je na slici 3.3:



Slika 3.3. Odziv nelinearnog dinamičkog sustava

Za učenje je upotrijebljeno prvih 400 točaka prikazanog skupa. Ostale točke biti će iskorištene kod testiranja. Učenje je provedeno za sve vrste aktivacijskih funkcija skrivenog sloja te u batch proceduri. Parametri učenja koji najbolje odgovaraju pojedinom algoritmu učenja dani su u tablici 3.5:



Algoritam/parametri	Brzina	Momentum	Koeficijent $u$	Težine $v, w$
EBP	0,3	0	N/A	-1 : 1
EBP+momentum	0,3	0,8	N/A	-1 : 1
EBP+Silva i Almeida	0,08	0	1,2	-1 : 1

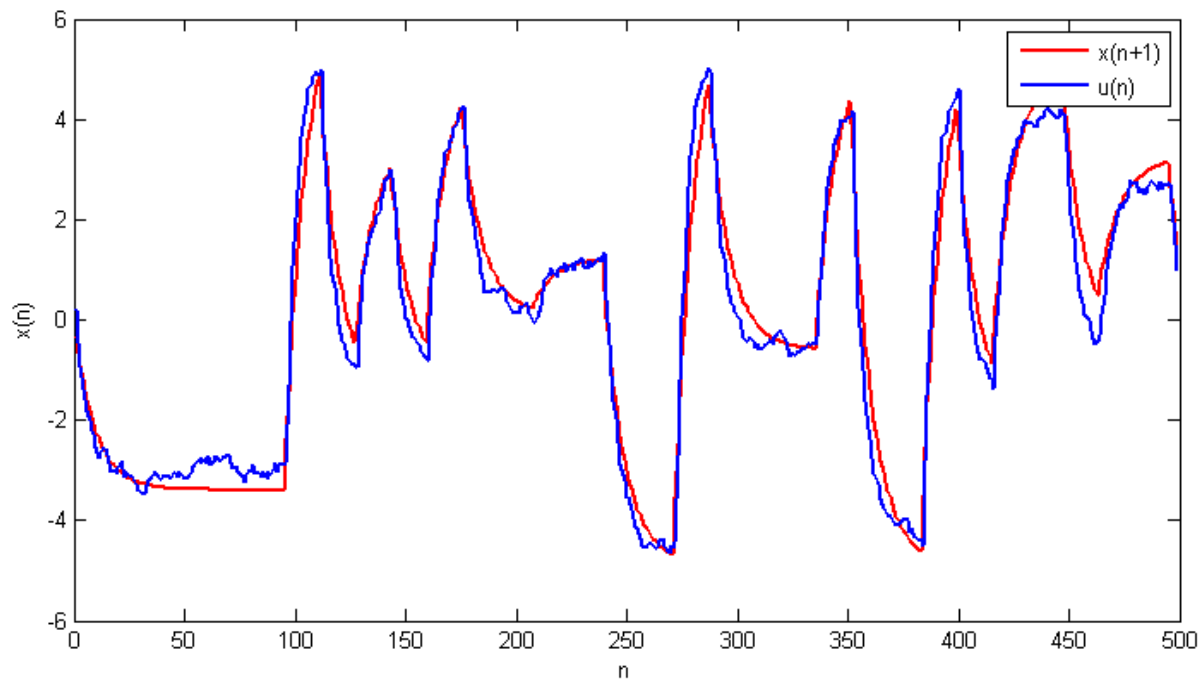
**Tablica 3.5.** Parametri učenja za nelinearni dinamički proces

Budući da se radi o složenijem procesu, broj skrivenih neurona biti će povećan na osam, čime bi naučena mreža trebala postići manju grešku. Početne težine birane su nasumično te su jednake za sve aktivacijske funkcije i algoritme učenja u pojedinom učenju. Ograničeni broj koraka učenja je 5000. Rezultati učenja prikazani su u tablici 3.6:

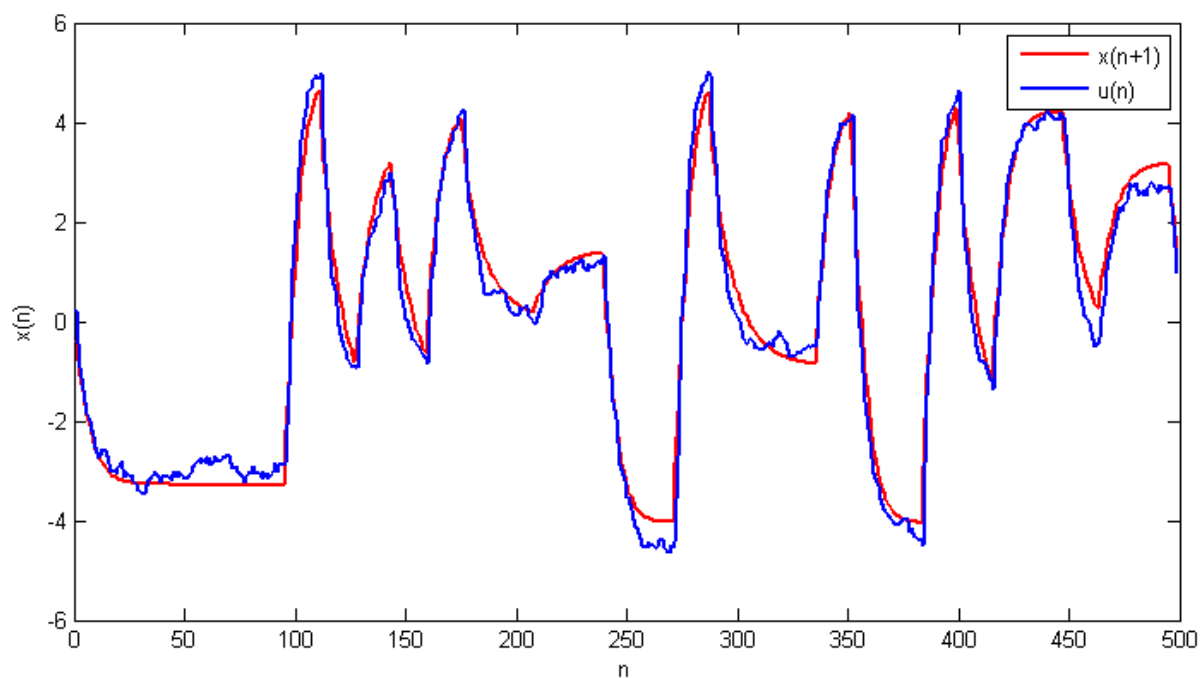
Akt. funkcija	Algoritam	NRMS	
		Učenje 1	Učenje 2
Linear	EBP	0,08	0,08
	EBP+M	0,08	0,08
	EBP+SA	0,08	0,08
Sigmoid	EBP	0,0826	0,084
	EBP+M	0,08	0,081
	EBP+SA	0,081	0,087
Sinus	EBP	0,08	0,081
	EBP+M	0,077	0,078
	EBP+SA	0,081	0,083
Gauss	EBP	0,081	0,086
	EBP+M	0,081	0,082
	EBP+SA	0,076	0,081

**Tablica 3.6.** Rezultati učenja za nelinearni dinamički sustav

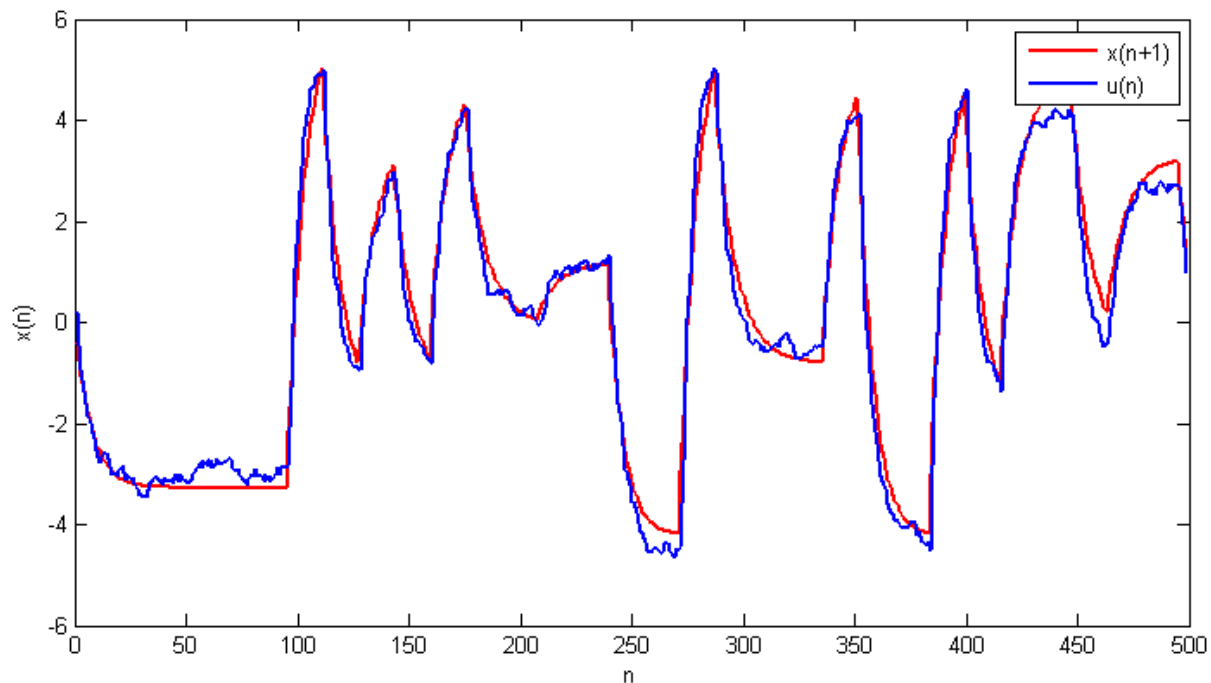
Kao što je vidljivo iz rezultata, svi algoritmi učenja imaju približnu istu pogrešku nakon učenja. Malo brži su ipak modificirani algoritmi, posebno na početku, ali se nakon 5000 koraka ne može očekivati velika razlika budući da svi algoritmi vrlo brzo dolaze do nekog minimuma. S gledišta aktivacijskih funkcija, pogreška učenja približno je jednaka za sve tipove funkcija. Za provjeru generalizacije pojedine aktivacijske funkcije, provedena su četiri testiranja naučenih mreža iz izvornog skupa za učenje:



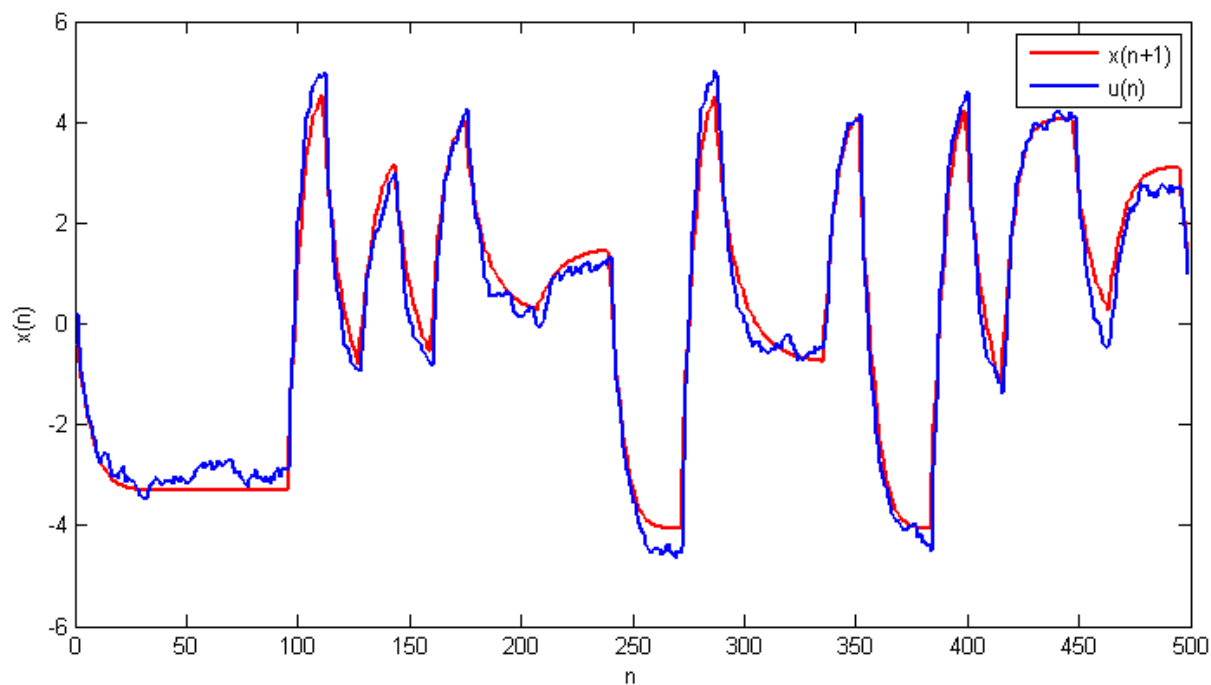
**Slika 3.4.** Odziv naučene mreže sa aktivacijskom funkcijom skrivenog sloja linear



**Slika 3.5.** Odziv naučene mreže sa aktivacijskom funkcijom skrivenog sloja sigmoid

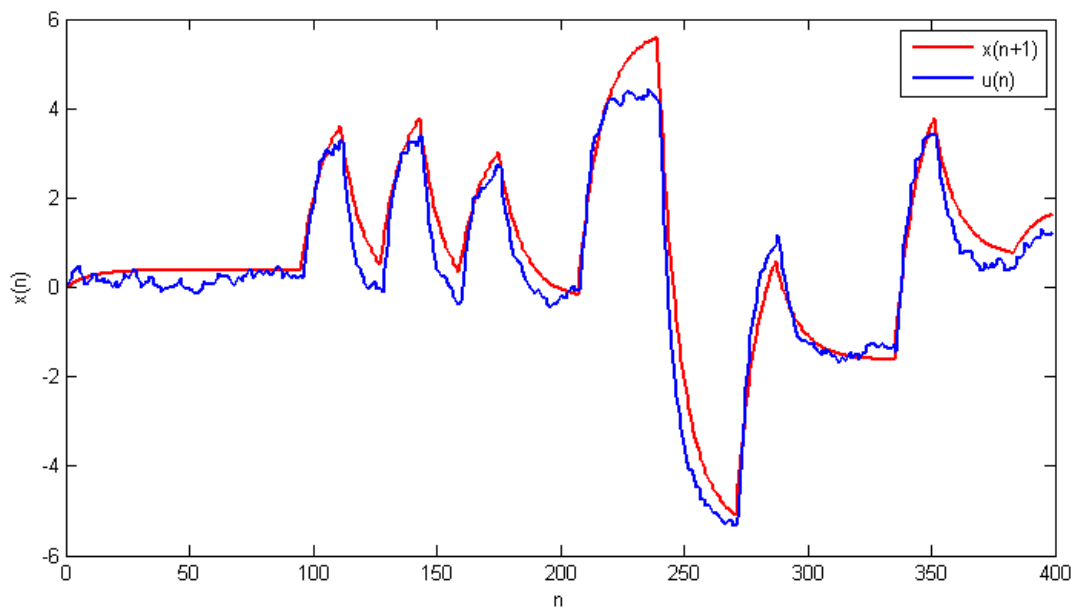


**Slika 3.6.** Odziv naučene mreže sa aktivacijskom funkcijom skrivenog sloja sinus

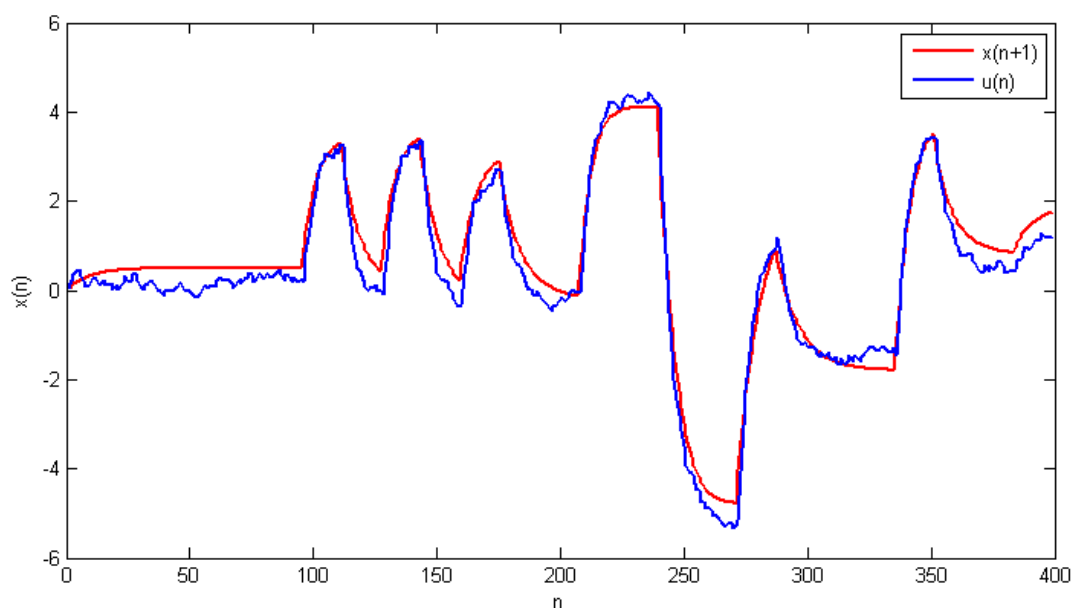


**Slika 3.7.** Odziv naučene mreže sa aktivacijskom funkcijom skrivenog sloja gauss

Na slici 3.4 vidi se da linearna funkcija s obzirom na svoju jednostavnost dobro opisuje zadani sustav. Odziv neuronskih mreža sa sigmoidnom, sinusnom i gaussovom aktivacijskom funkcijom sa slika 3.5, 3.6 i 3.7 vrlo je sličan što i ne čudi s obzirom na njihov sličan oblik, pri čemu i one s obzirom na pogrešku NRMS-a od 0,08 približno dobro opisuju sustav. Da bi se dobio potpuni uvid u generalizaciju, naučene mreže sa linearnom i sigmoidalnom aktivacijskom funkcijom skrivenog sloja biti će dodatno testirane na neučeni signal. Testiranja su prikazana na sljedećim slikama:



**Slika 3.8.** Odziv naučene mreže na neučeni signal sa aktivacijskom funkcijom skrivenog sloja linear



**Slika 3.9.** Odziv naučene mreže na neučeni signal sa aktivacijskom funkcijom skrivenog sloja sigmoid

Kao što se vidi na slikama 3.8 i 3.9, pri odzivu na neučeni signal naučena mreža sa sigmoidnom aktivacijskom funkcijom ima zaobljenije vrhove a time i bolju generalizaciju od prikazane mreže sa linearnom funkcijom. Ostala odstupanja približno su identična.

Kao što se može zaključiti iz navedenih testiranja, statička neuronska mreža puno teže rješava problem dinamičke nelinearne mreže od dinamičke linearne. Pritom ne pomaže ni odabir većeg broja neurona jer se pogreška NRMS-a još uvijek zadržava na približni vrijednosti od 0,08. Iako se ne postiže prevelika točnost, statička mreža ipak dobro generalizira zadani nelinearni dinamički proces, što je zapravo i cilj ovog testiranja.

## 4. Zaključak

U ovom radu prikazan je izvod i primjena jednostavne statičke neuronske mreže s povratnim rasprostiranjem greške. Uzimajući u obzir jednostavnost strukture i učenja navedene neuronske mreže, naučene mreže postigle su dobar rezultat u navedenim primjerima. Posebno dobro ispadaju prva dva primjera, primjer klasifikacije XOR problemom i identifikacije linearnog dinamičkog sustava, gdje je postignuta velika točnost i vrlo dobra generalizacija. Modificirani algoritmi učenja pokazali su dobro ubrzanje učenja i dokazali kako se s njihovim uključivanjem mogu postići mnogo veće brzine, što pak ovisi o vrsti problema, kao što je ovdje i pokazano. Usporedba različitih aktivacijskih funkcija pokazala je kako u nekim primjerima neke funkcije mogu savršeno opisati proces (linearna kod linearnog dinamičkog procesa) dok u drugim primjerima ne daju nikakve rezultate (linearna kod XOR problema). Svaka aktivacijska funkcija ima neka svoja dobra svojstva te se može zaključiti kako je od posebne važnosti odabir aktivacijskih funkcija pojedinih slojeva, budući da izbor utječe na konačni rezultat.

## 5. Literatura

- [1] Novaković, B., Majetić, D., Široki, M.: *Umjetne neuronske mreže*, Fakultet strojarstva i brodogradnje, Zagreb, 1998
- [2] Schiffmann, W., Joost, M., Werner, R.: *Optimization of the Backpropagation Algorithm for Training Multilayer Perceptrons*, University of Koblenz, Koblenz, 1994