

# Učenje neuronske mreže po uzorku i skupu za učenje

---

Rožić, Igor

Undergraduate thesis / Završni rad

2009

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://um.nsk.hr/um:nbn:hr:235:043095>

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

Download date / Datum preuzimanja: **2024-09-12**

Repository / Repozitorij:

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



SVEUČILIŠTE U ZAGREBU  
FAKULTET STROJARSTVA I BRODOGRADNJE

# ZAVRŠNI RAD

Igor Rožić

Zagreb, 2009.

SVEUČILIŠTE U ZAGREBU  
FAKULTET STROJARSTVA I BRODOGRADNJE

# ZAVRŠNI RAD

Voditelj rada:  
Prof. dr. sc. Dubravko Majetić

Igor Rožić

Zagreb, 2009.

Izjavljujem da sam samostalno izradio završni rad koristeći se stečenim znanjem i iskustvom, te informacijama prikupljenim iz literature i sa Interneta.

Zahvaljujem se prof. dr. sc. Dubravku Majetiću, mentoru ovoga rada, na stručnoj pomoći pri izradi ovog rada.

# SADRŽAJ

POPIS SLIKA .....	III
POPIS TABLICA.....	V
POPIS OZNAKA .....	VI
Sažetak .....	VII
1. Uvod.....	1
1.1. Biološki neuron.....	1
1.2. Umjetni neuron .....	2
1.3. Podjela umjetnih neuronskih mreža .....	3
2. Statičke neuronske mreže koje uče po principu povratnog prostiranja pogreške .....	5
2.1. Model statičkog neurona i statičke neuronske mreže .....	5
2.2. Algoritam učenja povratnim prostiranjem pogreške .....	7
2.3. Matematički model statičke neuronske mreže s učenjem po principu povratnog prostiranja pogreške.....	8
2.3.1. Unaprijedna faza učenja .....	8
2.3.2. Povratna faza učenja.....	10
2.4. Matematički model povratne faze učenja za učenje po uzorku skupa za učenje .....	11
2.5. Matematički model povratne faze učenja za učenje po skupu za učenje .....	14
2.6. Ocjena uspješnosti algoritma učenja .....	15
3. Usporedba učenja po uzorku i učenja po skupu za učenje za klasifikacijski problem .....	17
3.1. Početni parametri učenja .....	18
3.2. Usporedba rezultata učenja za učenje po uzorku i učenje po skupu za učenje.....	19
4. Usporedba učenja po uzorku i učenja po skupu za učenje za regresijski problem .....	24
4.1. Početni parametri učenja .....	25
4.2. Usporedba rezultata učenja za učenje po uzorku i učenje po skupu za učenje.....	26
4.3. Nova krivulja za učenje mreže .....	29
4.4. Početni parametri učenja (nova krivulja).....	29
4.5. Usporedba rezultata učenja za učenje po uzorku i učenje po skupu za učenje.....	31
5. Zaključak.....	40
Literatura .....	41

## POPIS SLIKA

<b>Slika 1.1.</b> Pojednostavljena struktura biološkog neurona .....	1
<b>Slika 1.2.</b> Prijenos signala u sinapsi.....	2
<b>Slika 1.3.</b> Struktura umjetnog neurona .....	3
<b>Slika 2.1.</b> Model statičkog neurona .....	5
<b>Slika 2.2.</b> Nelinearna bipolarna sigmoidalna funkcija.....	6
<b>Slika 2.3.</b> Model statičke neuronske mreže .....	7
<b>Slika 3.1.</b> Dijagram ovisnosti NRMS-a i koraka učenja za učenje po uzorku (prvi slučaj) .....	19
<b>Slika 3.2.</b> Dijagram ovisnosti NRMS-a i koraka učenja za učenje po skupu (prvi slučaj).....	20
<b>Slika 3.3.</b> Dijagram ovisnosti NRMS-a i koraka učenja za učenje po uzorku za nove vrijednosti početnih parametara (prvi slučaj).....	21
<b>Slika 3.4.</b> Dijagram ovisnosti NRMS-a i koraka učenja za učenje po skupu za nove vrijednosti početnih parametara (prvi slučaj).....	22
<b>Slika 3.5.</b> Dijagram ovisnosti NRMS-a i koraka učenja za učenje po uzorku i učenje po skupu (drugi slučaj) .....	23
<b>Slika 4.1.</b> Sinusoidna krivulja za učenje .....	24
<b>Slika 4.2.</b> Zadane točke učenja za prvi slučaj .....	25
<b>Slika 4.3.</b> Usporedba dobivene krivulje i tražene krivulje za učenje po uzorku.....	26
<b>Slika 4.4.</b> Usporedba dobivene krivulje i tražene krivulje za učenje po skupu .....	27
<b>Slika 4.5.</b> Dijagram ovisnosti NRMS-a i koraka učenja za učenje po uzorku (regresijski problem) .....	27
<b>Slika 4.6.</b> Dijagram ovisnosti NRMS-a i koraka učenja za učenje po skupu (regresijski problem)..	28
<b>Slika 4.7.</b> Prikaz točaka sa istom vrijednošću varijable $y$ za različite vrijednosti varijable $x$ .....	28
<b>Slika 4.8.</b> Nova krivulja za učenje .....	29
<b>Slika 4.9.</b> Zadane točke učenja nove krivulje za prvi slučaj.....	30
<b>Slika 4.10.</b> Zadane točke učenja nove krivulje za drugi slučaj.....	31
<b>Slika 4.11.</b> Dijagram ovisnosti NRMS-a i koraka učenja za učenje po uzorku i učenje po skupu (prvi slučaj – bez momentuma).....	32
<b>Slika 4.12.</b> Usporedba dobivene i tražene krivulje za učenje po uzorku (prvi slučaj – bez momentuma).....	32
<b>Slika 4.13.</b> Usporedba dobivene i tražene krivulje za učenje po skupu (prvi slučaj – bez momentuma).....	33
<b>Slika 4.14.</b> Dijagram ovisnosti NRMS-a i koraka učenja za učenje po uzorku i učenje po skupu (prvi slučaj – sa momentumom).....	34

<b>Slika 4.15.</b> Usporedba dobivene i tražene krivulje za učenje po uzorku (prvi slučaj – sa momentumom) .....	34
<b>Slika 4.16.</b> Usporedba dobivene i tražene krivulje za učenje po skupu (prvi slučaj – sa momentumom) .....	35
<b>Slika 4.17.</b> Dijagram ovisnosti NRMS-a i koraka učenja za učenje po uzorku i učenje po skupu (drugi slučaj – bez momentuma).....	36
<b>Slika 4.18.</b> Usporedba dobivene i tražene krivulje za učenje po uzorku (drugi slučaj – bez momentuma).....	36
<b>Slika 4.19.</b> Usporedba dobivene i tražene krivulje za učenje po skupu (drugi slučaj – bez momentuma).....	37
<b>Slika 4.20.</b> Dijagram ovisnosti NRMS-a i koraka učenja za učenje po uzorku i učenje po skupu (drugi slučaj – sa momentumom).....	38
<b>Slika 4.21.</b> Usporedba dobivene i tražene krivulje za učenje po uzorku (drugi slučaj – sa momentumom) .....	38
<b>Slika 4.22.</b> Usporedba dobivene i tražene krivulje za učenje po skupu (drugi slučaj – sa momentumom) .....	39

## POPIS TABLICA

<b>Tablica 3.1.</b> XOR problem .....	17
<b>Tablica 3.2.</b> Početni parametri učenja mreže za prvi slučaj .....	18
<b>Tablica 3.3.</b> Početni parametri učenja mreže za drugi slučaj .....	18
<b>Tablica 3.4.</b> Usporedba rezultata učenja za prvi slučaj klasifikacijskog problema.....	19
<b>Tablica 3.5.</b> Nove vrijednosti početnih parametara za prvi slučaj .....	20
<b>Tablica 3.6.</b> Usporedba rezultata učenja s novim vrijednostima početnih parametara za prvi slučaj klasifikacijskog problema .....	21
<b>Tablica 3.7.</b> Usporedba rezultata učenja za drugi slučaj klasifikacijskog problema.....	22
<b>Tablica 4.1.</b> Početni parametri učenja mreže .....	25
<b>Tablica 4.2.</b> Usporedba rezultata učenja za regresijski problem.....	26
<b>Tablica 4.3.</b> Početni parametri učenja mreže (nova krivulja).....	30
<b>Tablica 4.4.</b> Usporedba rezultata učenja za prvi slučaj regresijskog problema (bez momentuma) ..	31
<b>Tablica 4.5.</b> Usporedba rezultata učenja za prvi slučaj regresijskog problema (sa momentumom) .	33
<b>Tablica 4.6.</b> Usporedba rezultata učenja za drugi slučaj regresijskog problema (bez momentuma)	35
<b>Tablica 4.7.</b> Usporedba rezultata učenja za drugi slučaj regresijskog problema (sa momentumom)	37



## POPIS OZNAKA

BIAS	neuron bez ulaza s konstantnom izlaznom vrijednosti jednakom 1
$D_k$	željena vrijednost k-tog izlaza neuronske mreže
$E$	funkcija cilja
$I$	broj neurona ulaznog sloja mreže
$J$	broj neurona sakrivenog sloja mreže
$K$	broj neurona izlaznog sloja mreže
$n$	broj koraka učenja
$net$	vrijednost funkcije sume
$net_{Hj}$	vrijednost funkcije sume j-tog neurona sakrivenog sloja
$net_{Ok}$	vrijednost funkcije sume k-tog neurona izlaznog sloja
NRMS	mjera točnosti – normalizirani korjen srednje kvadratne pogreške
$N_u$	broj uzoraka u skupu za učenje
$O_k$	k-ti izlaz neuronske mreže
RMS	korjen srednje kvadratne pogreške
$u_j$	vrijednost j-og ulaza neurona
$v_{ji}$	težinski faktor veze između j-og neurona sakrivenog sloja i i-tog neurona ulaznog sloja
$w_j$	težinski faktor j-og ulaza neurona
$w_{kj}$	težinski faktor veze između k-og neurona izlaznog sloja i j-tog neurona sakrivenog sloja
$y$	vrijednost izlaza neurona
$y_j$	vrijednost izlaza j-tog neurona sakrivenog sloja
$Z_i$	i-ti ulaz neuronske mreže
$\alpha$	vrijednost momentuma
$\Delta$	iznos promjene težinskih faktora u jednom koraku
$\gamma$	aktivacijska funkcija neurona
$\eta$	koeficijent brzine učenja
$\sigma_{dn}$	standardna devijacija željenih vrijednosti izlaza mreže
$\mathcal{G}$	parametar učenja (težinski faktor)
$\mathcal{G}(n)$	trenutna vrijednost parametra učenja
$\mathcal{G}(n+1)$	nova vrijednost parametra učenja
$\Delta\mathcal{G}(n)$	tekuća promjena parametra učenja
$\nabla E$	gradijent pogreške
$\nabla E_{ukupni}$	gradijent pogreške za cijeli skup učenja

Ostale oznake, kao i navedene oznake s drugim značenjem, objašnjene su u tekstu.

## Sažetak

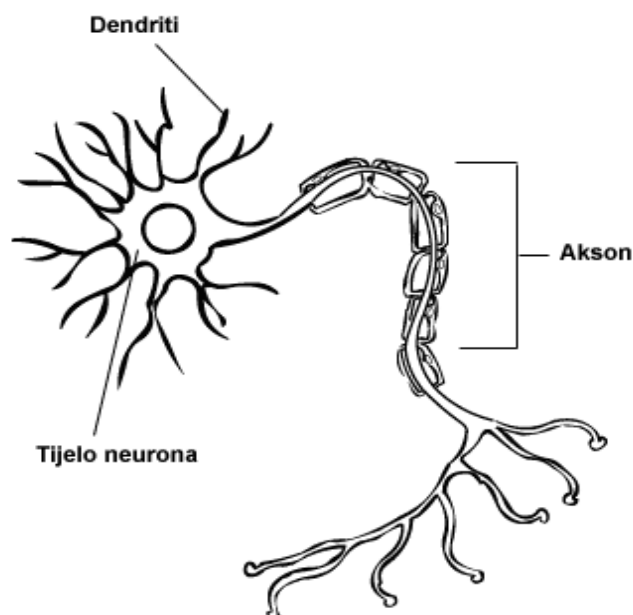
Završni rad se sastoji od pet poglavlja. U prvom, uvodnom poglavlju daje se kratki osvrt na osnovne značajke umjetnih neuronskih mreža i njihovu povezanost i sličnost sa biološkim neuronskim mrežama. U drugom poglavlju govori se o neuronskim mrežama koje su i tema ovog završnog rada – statičke neuronske mreže sa povratnim prostiranjem pogreške. Izveden je i matematički model za dvije procedure učenja ovih neuronskih mreža. Prva procedura je učenje po uzorku skupa za učenje, a druga je učenje po skupu za učenje. Ovdje je još opisana i ocjena uspješnosti algoritma učenja odnosno mjera točnosti. U trećem poglavlju uspoređeni su rezultati učenja spomenutih dvaju procedura za klasifikacijski problem. Uspoređivani su sljedeći parametri: dobivena mjera točnosti, potrebno vrijeme učenja, odziv naučene mreže, te potreban broj koraka učenja kada je zadana tražena mjera točnosti. U četvrtom poglavlju uspoređeni su rezultati učenja istih dvaju procedura, ali za regresijski problem. Uspoređivani su isti parametri. Eksperimenti iz trećeg i četvrtog poglavlja rađeni su u programskom paketu Matlab. U zadnjem, petom poglavlju dan je zaključak na temelju rezultata dobivenih u prethodna dva poglavlja.

## 1. Uvod

Umjetne neuronske mreže kreirane su po uzoru na rad ljudskog mozga odnosno biološke neuronske mreže. Iz toga proizlazi da je glavna zadaća umjetnih neuronskih mreža procesiranje (prihvatanje, obrada, generiranje, pohranjivanje i prijenos) informacija. Da bi se lakše shvatio način na koji radi umjetna neuronska mreža odnosno umjetni neuron u ovom uvodu daje se i kratak prikaz biološkog neurona i njegovih funkcija.

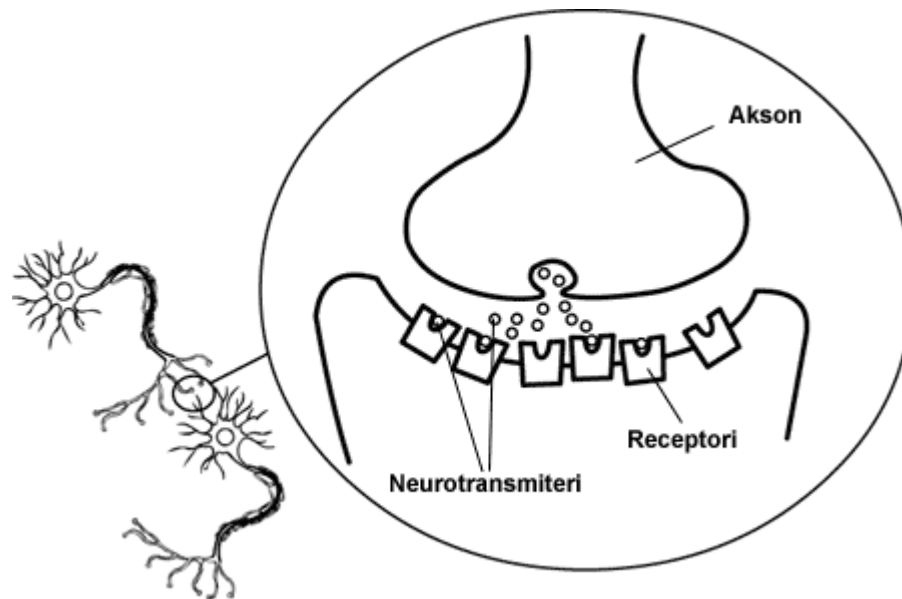
### 1.1. Biološki neuron

Sa slike 1.1. je vidljivo da se biološki neuron sastoji iz tijela, aksona i mnoštva dendrita koji okružuju tijelo neurona.



**Slika 1.1.** Pojednostavljena struktura biološkog neurona

Akson je dugačko vlakno čiji je jedan kraj povezan sa tijelom neurona, a drugi kraj se dijeli na niz grana koje završavaju malim zadebljanjima. Na ta se zadebljanja dalje nastavljaju dendriti sljedećeg neurona (češće) ili tijelo sljedećeg neurona (rjeđe). Ipak, postoji mali razmak između krajeva aksona prethodnog neurona i dendrita ili tijela sljedećeg neurona. Taj razmak se naziva sinapsa i prikazan je na slici 1.2. Akson jednog neurona formira sinaptičke veze s mnoštvom drugih neurona.

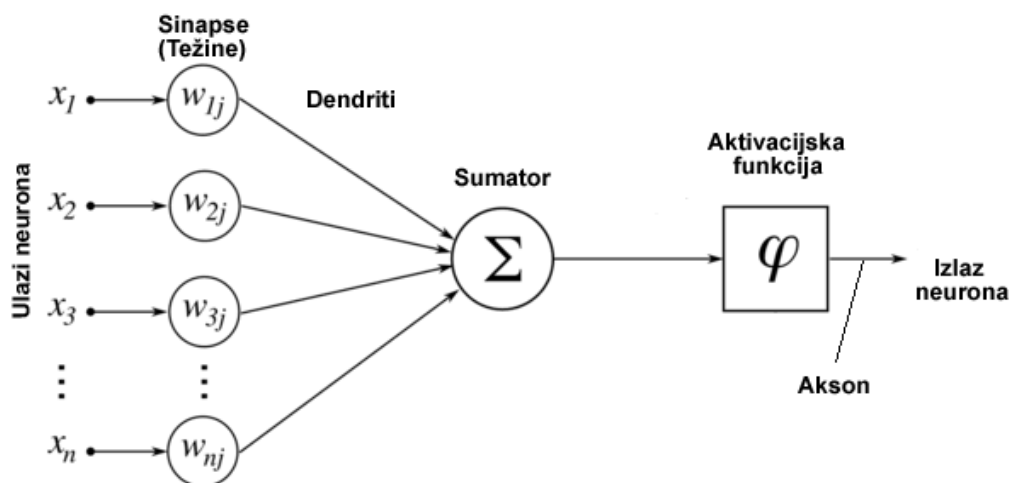


**Slika 1.2.** Prijenos signala u sinapsi

Impulsi koje šalje tijelo neurona putuju preko aksona do dendrita ili tijela sljedećeg neurona. Prijenos signala s neurona na neuron se zbiva otpuštanjem kemijskih tvari u sinapsu. Te kemijske tvari se nazivaju neurotransmiteri [1]. Na krajevima dendrita nalaze se specijalne molekule, nazvane receptori, koje primaju i procesiraju signal. Signali koje prenose dendriti mogu biti ili smirujući ili uzbudni. Ti signali su, matematički gledano, suprotnog predznaka. Neuron će poslati impuls kroz svoj akson ako razlika između uzbude i smirujućeg utjecaja dosegne kritični iznos, koji predstavlja prag osjetljivosti neurona.

## **1.2. Umjetni neuron**

Karakteristika umjetnog neurona je oponašanje osnovne funkcije (karakteristike prvog reda) biološkog neurona. U usporedbi sa biološkim neuronom, umjetni neuron, prikazan na slici 1.3., se sastoji od sumatora (zamjena za tijelo biološkog neurona), ulaza u sumator (zamjena za dendrite), izlaza sumatora (zamjena za akson biološkog neurona) i aktivacijske funkcije (zamjena za prag osjetljivosti biološkog neurona). Veza umjetnog neurona sa okolinom se ostvaruje preko težinskih faktora. [2]



**Slika 1.3.** Struktura umjetnog neurona

Težinski faktor može biti pozitivan ili negativan broj, a kod suvremenih umjetnih neuronskih mreža i neka funkcija. Ako je težinski faktor jednak nuli onda odgovarajuća veza s okolinom neurona ne postoji. Što zapravo težinski faktori rade? U usporedbi sa biološkim neuronom, vidi se da težinski faktori kod umjetnog neurona rade ono što i sinapse kod biološkog neurona: povezuju izlaze drugih neurona (aksone) s ulazima sumatora (dendriti). Pritom, o iznosu težinskog faktora ovisi intenzitet te veze, dok karakter veze (smirujuća ili uzbudna) ovisi o predznaku težinskog faktora. Izlaz sumatora povezuje se na aktivacijsku funkciju, a aktivacijska funkcija na svom izlazu producira konačni izlaz umjetnog neurona. Aktivacijske funkcije mogu biti linearne i nelinearne. Karakteristika linearnih aktivacijskih funkcija je da se izlaz neurona dobiva množenjem izlaza sumatora sa nekim faktorom (pojačanjem). Od nelinearnih aktivacijskih funkcija najčešće se koriste: funkcije praga osjetljivosti, sigmoidalne, hiperbolične i harmonične funkcije. Svaka od navedenih aktivacijskih funkcija ima svoje područje primjene, gdje posjeduje određene prednosti.

### 1.3. Podjela umjetnih neuronskih mreža

Umjetne neuronske mreže se sastoje od slojeva, a jedan sloj neuronske mreže je izgrađen od paralelno složenog skupa neurona. Ako se podjela umjetnih neuronskih mreža vrši prema broju slojeva, tada se može reći da postoje jednoslojne i višeslojne umjetne neuronske mreže. Višeslojne mreže se sastoje od ulaznog i izlaznog sloja, a između njih se nalaze skriveni slojevi. Skriveni slojevi nemaju direktnu interakciju s okolinom. Ako se slojevi neuronske mreže povežu tako da signali putuju jednosmjerno od ulaza prema izlazu, tada se govori o unaprijednim (statičkim) neuronskim mrežama (engl. Feed-forward Neural Networks). Ukoliko postoji bar jedna povratna petlja unutar neuronske mreže, tada se govori o povratnim (dinamičkim) neuronskim mrežama

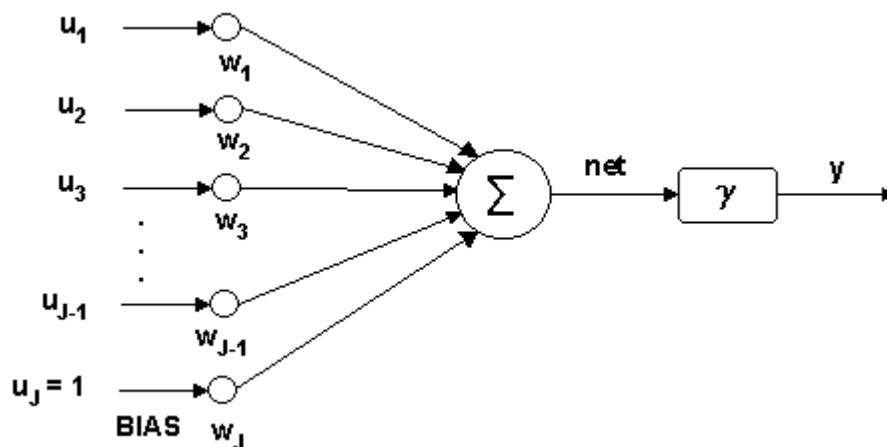
(engl. Feedback or Recurrent Neural Networks). Ako se podjela vrši prema učitelju, tada se neuronske mreže dijele na one s učiteljem (supervizorne) i one bez učitelja (nesupervizorne). Supervizorne neuronske mreže zahtijevaju vanjskog “učitelja” koji promatra ponašanje mreže te radi korekcije na istoj dok se ne dobije željeno ponašanje mreže. Nesupervizorne neuronske mreže se same organiziraju, pa se još nazivaju i samoorganizirajuće neuronske mreže. Dijeljenjem neuronskih mreža prema načinu učenja razlikuju se mreže koje uče iterativno, mreže koje uče u jednom koraku i mreže koje uče kombinacijom ovih dvaju algoritama učenja. Kao primjer mreža s kombiniranim učenjem mogu se navesti sve popularnije neuronske mreže s radijalnim baznim funkcijama (engl. RBF – Radial Basis Function). Još jedna važna podjela neuronskih mreža je podjela prema pravilu odnosno algoritmu učenja. Toj grupi pripadaju Kohonenove neuronske mreže, Hopfieldove neuronske mreže, Hebbove neuronske mreže, MADALINE neuronske mreže, neuronske mreže s učenjem po principu povratnog prostiranja pogreške (engl. Error-Back Propagation), i dr. Ovaj rad bavi se unaprijednim (statičkim) neuronskim mrežama koje uče po principu povratnog prostiranja pogreške te će o njima biti više rečeno u sljedećim poglavljima.

## 2. Statičke neuronske mreže koje uče po principu povratnog prostiranja pogreške

Ove neuronske mreže su najpoznatiji i najčešće upotrebljavani tip neuronskih mreža, a algoritam učenja koji te mreže koriste potaknuo je široku upotrebu neuronskih mreža kao popularne metode u raznim područjima. Prvi kreator ovog algoritma učenja bio je Paul Werbos 1974. godine, ali pravu primjenu ovaj algoritam je doživio 1986. godine kada su David E. Rumelhart, Geoffrey E. Hinton i Ronald J. Williams prepoznali njegovu vrijednost. [3]

### 2.1. Model statičkog neurona i statičke neuronske mreže

Na slici 2.1. prikazan je standardni model statičkog neurona. Bitno je uočiti dvije temeljne podfunkcije statičkog neurona – funkciju sume  $\Sigma$  i aktivacijsku funkciju  $\gamma$ , te osnovnu značajku svakog neurona – posjedovanje više ulaza i samo jednog izlaza. Ono što se još može primjetiti je poseban ulaz koji se naziva BIAS i čija je vrijednost izlaza konstantna i jednaka jedinici. Bez tog posebnog ulaza unutar svakog neurona učenje se ne bi moglo odvijati.



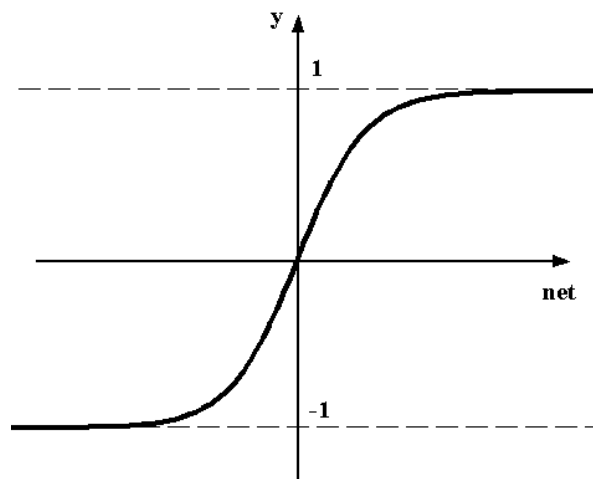
Slika 2.1. Model statičkog neurona

Izrazom (2-1) definirana je funkcija sume statičkog neurona koja predstavlja sumu umnožaka ulaza neurona i pripadajućih težinskih faktora veza. Rezultat funkcije sume je vrijednost  $net$ . Pomoću aktivacijske funkcije  $\gamma$  vrši se preslikavanje vrijednosti  $net$  u izlaznu vrijednost neurona  $y$ , što je prikazano izrazom (2-2).

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

$$y = \gamma(\text{net}). \quad (2-2)$$

U uvodnom poglavlju su navedene najčešće korištene aktivacijske funkcije, a ovdje će se spomenuti ona koja se koristi u ovom radu i koja je ustvari najčešće korištena u usporedbi sa svim aktivacijskim funkcijama neurona. To je nelinearna bipolarna sigmoidalna funkcija i ona je prikazana na slici 2.2.



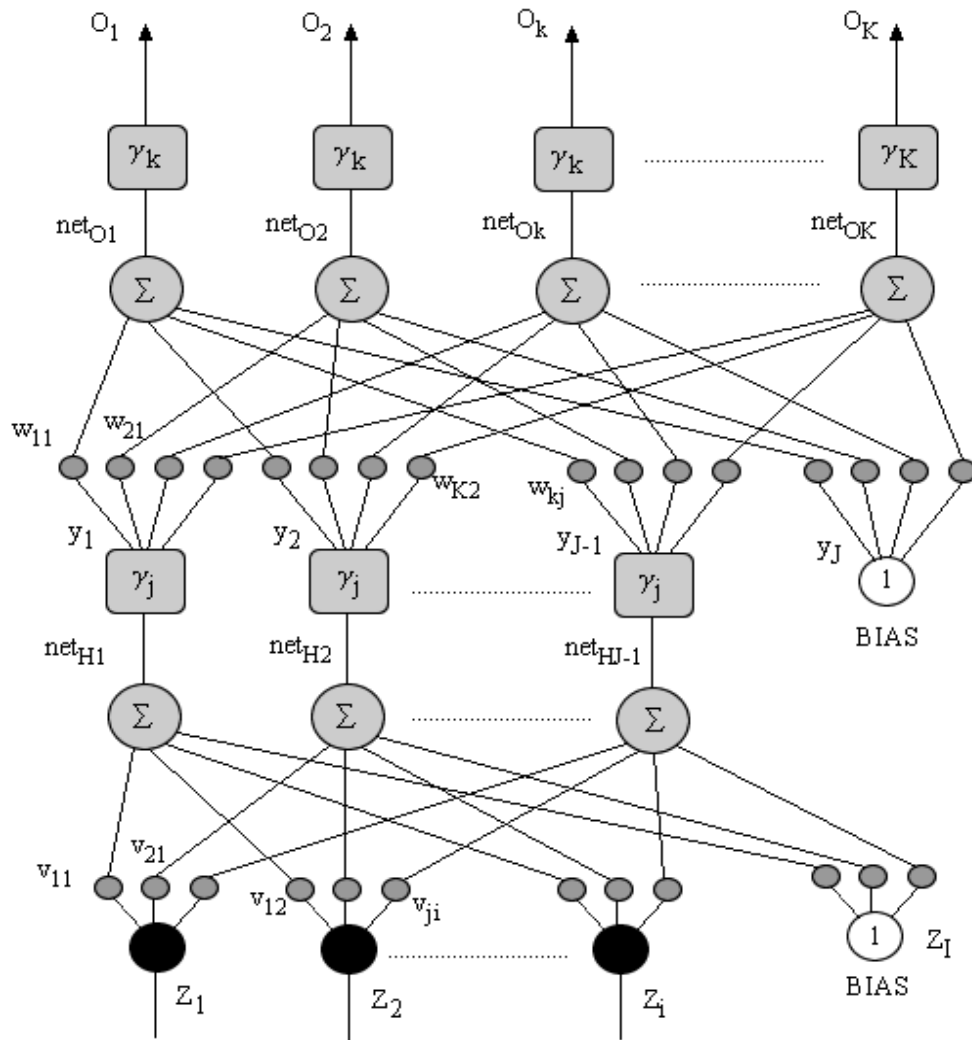
**Slika 2.2.** Nelinearna bipolarna sigmoidalna funkcija

Sa slike se vidi da ova aktivacijska funkcija normira vrijednosti izlaza neurona na vrijednosti između -1 i 1, a to pridonosi numeričkoj stabilnosti algoritma učenja. Također, ova aktivacijska funkcija je globalnog karaktera te pokazuje dobre rezultate i prilikom učenja dinamičkih sustava. Kao što je već rečeno, u ovom radu se koristi upravo ova aktivacijska funkcija, stoga izraz (2-2) poprima sljedeći oblik:

$$y = \frac{2}{1 + e^{-\text{net}}} - 1. \quad (2-3)$$

Da bi se dobila struktura neuronske mreže, neuroni se organiziraju u slojeve, a slojevi se međusobno povezuju vezama koje su opterećene težinskim koeficijentima. Postoje tri tipa slojeva mreže: ulazni sloj, sakriveni sloj i izlazni sloj. Već je rečeno da su ulazni i izlazni sloj u direktnoj interakciji s okolinom, dok sakriveni sloj nije. Potrebno je znati i da je broj sakrivenih slojeva proizvoljan. Na slici 2.3. je prikazan model statičke neuronske mreže sa jednim sakrivenim slojem.





Slika 2.3. Model statičke neuronske mreže

Sa slike 2.3. se može vidjeti da su svi slojevi mreže potpuno umreženi odnosno svaki neuron promatranog sloja povezan je sa svakim neuronom prethodnog sloja. Izuzetak su neuroni označeni sa BIAS. Također se može vidjeti da se težinski faktori između ulaznog i sakrivenog sloja označavaju sa  $v_{ji}$ , dok se težinski faktori između sakrivenog i izlaznog sloja označavaju sa  $w_{kj}$ . Izlazi neuronske mreže su označeni sa  $O_k$ .

## 2.2. Algoritam učenja povratnim prostiranjem pogreške

Kako zapravo neuronske mreže uče? Neuronske mreže uče tako da podešavaju vrijednosti težinskih faktora između slojeva mreže kako bi se dobiveni izlazi mreže što više približili traženim vrijednostima izlaza mreže. Već je spomenuto da postoje razni algoritmi učenja neuronskih mreža, a ovdje će se opisati algoritam učenja povratnim prostiranjem pogreške. Prva faza mreže koja uči po

tom algoritmu je da zadane ulaze sprovede od ulaznog do izlaznog sloja. Ta faza se zove unaprijedna faza. Kada se za te ulaze dobiju određene vrijednosti izlaza, tada se te vrijednosti uspoređuju sa traženim izlazima. Na taj način se dobije greška učenja. U sljedećoj fazi greška učenja se koristi za promjenu težinskih faktora od izlaznog sloja mreže prema ulaznom sloju mreže. Ovo se zove povratna faza. Jedan korak učenja je prolazak kroz cijeli skup za učenje (zapisnik učenja). Postoje dva načina na koja se može odvijati promjena parametara učenja. Jedan način je učenje po uzorku skupa za učenje (tzv. “pattern” procedura), a drugi način je učenje po skupu za učenje (tzv. “batch” procedura). U sljedećem poglavlju će biti detaljnije razrađene obje procedure.

### **2.3. Matematički model statičke neuronske mreže s učenjem po principu povratnog prostiranja pogreške**

Kao što je već rečeno, algoritam učenja se sastoji od dvije faze – unaprijedna faza i povratna faza. U nastavku će biti razrađeni matematički modeli obje faze po redoslijedu u kojem se pojavljuju.

#### **2.3.1. Unaprijedna faza učenja**

U unaprijednoj fazi učenja iz skupa za učenje se uzimaju vrijednosti svih ulaza mreže  $Z$ , te se s njima prolazi kroz cijelu mrežu da bi se dobile vrijednosti izlaza  $O$ . Prije prolaska ulaza kroz mrežu potrebno je odrediti početne vrijednosti težinskih faktora  $v$  i  $w$ . Početne vrijednosti mogu biti ili zadane ili generirane generatorom slučajnih brojeva. U nastavku je dan matematički model unaprijedne faze učenja od ulaznog prema izlaznom sloju.

#### ***Sakriveni sloj***

U sakrivenom sloju je potrebno sa vrijednostima ulaza mreže i vrijednostima težinskih faktora dobiti funkciju sume  $net$  neurona sakrivenog sloja  $H$ . Prvi indeks u oznaci funkcije sume  $net$  neurona je oznaka sakrivenog sloja  $H$ , a drugi indeks funkcija sume dobiva za svaki  $j$ -ti neuron. Funkcija sume  $net$  neurona se računa na sljedeći način:

$$net_{Hj} = \sum_{i=1}^I v_{ji} Z_i, \quad j = 1, 2, \dots, J-1, \quad i = 1, 2, \dots, I, \quad (2-4)$$

gdje  $I$  predstavlja broj ulaznih neurona uvećan za jedan (BIAS), a  $J$  predstavlja broj neurona sakrivenog sloja uvećan za jedan (BIAS).

Za aktivacijsku funkciju je odabrana nelinearna bipolarna sigmoidalna funkcija prikazana slikom 2.2., pa se stoga izlazi neurona sakrivenog sloja računaju prema:

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

Pošto sakriveni sloj sadrži  $J$  neurona, izraz (2-5) nije potpun. Da bi izraz (2-5) bio potpun potrebno je definirati izlaz  $J$ -tog neurona sakrivenog sloja. Sa slike 2.3. je vidljivo da je  $J$ -ti neuron u sakrivenom sloju BIAS, pa je stoga izlaz  $J$ -tog neurona sakrivenog sloja definiran kao:

$$y_j = 1. \quad (2-6)$$

Vrijednosti izlaza neurona sakrivenog sloja spojene su na ulaz svakog neurona izlaznog sloja preko težinskih faktora  $w_{kj}$ .

### ***Izlazni sloj***

Funkcija sume  $net$  neurona izlaznog sloja  $O$  je definirana slično kao i funkcija sume  $net$  neurona sakrivenog sloja u izrazu (2-4). Prvi indeks u oznaci funkcije sume  $net$  neurona izlaznog sloja pripada izlaznom sloju  $O$ , dok drugi indeks pripada svakom neuronu izlaznog sloja. Funkcija sume  $net$  neurona se izračunava izrazom:

$$net_{Ok} = \sum_{j=1}^J w_{kj} y_j, \quad k = 1, 2, \dots, K, \quad (2-7)$$

gdje  $K$  predstavlja broj neurona izlaznog sloja odnosno broj izlaza neuronske mreže.

Ako se za aktivacijsku funkciju izlaznog sloja odabere linearna funkcija tada izraz za izlaz mreže glasi:

$$O_k = K_p net_{Ok}, \quad k = 1, 2, \dots, K, \quad (2-8)$$

gdje je  $K_p$  nagib linearne aktivacijske funkcije. Ovime je moguće ostvariti vrijednosti izlaza mreže i veće od 1. Ipak, ovdje je korišteno  $K_p=1$ , pa stoga izraz (2-8) poprima ovaj oblik:

$$O_k = net_{Ok}, \quad k = 1, 2, \dots, K. \quad (2-9)$$

Ovime završava unaprijedna faza učenja i počinje povratna faza učenja.

### 2.3.2. Povratna faza učenja

U povratnoj fazi učenja se preko ostvarenog izlaza mreže i željenog izlaza mreže izračunava pogreška učenja. Na osnovu pogreške učenja vrši se podešavanje vrijednosti težinskih faktora. Ukoliko je od strane učitelja zadana pogreška učenja, tada se ovaj postupak ponavlja sve dok se ne postigne pogreška manja ili jednaka zadanoj pogrešci. Postupak podešavanja težinskih faktora se može opisati i kao takvo podešavanje težinskih faktora koje minimizira funkciju cilja  $E$ , što znači da je potrebno odabrati funkciju cilja. Najčešće korištena funkcija cilja je suma kvadrata pogreške, a dana je izrazom:

$$E = \frac{1}{2} \sum_{n=1}^N (D_n - O_n)^2 , \quad (2-10)$$

gdje je  $N$  broj elemenata u skupu za učenje odnosno broj podataka učenja iz zapisnika učenja. Promjena težinskih faktora vrši se primjenom nekog od algoritama nelinearnog optimiranja. Odabrana forma promjene težinskih faktora (parametara učenja)  $\mathcal{G}$  je dana izrazom (2-11):

$$\mathcal{G}(n+1) = \mathcal{G}(n) + \Delta\mathcal{G}(n) , \quad (2-11)$$

gdje je  $n$  trenutni korak učenja,  $\Delta\mathcal{G}(n)$  veličina promjene parametara učenja ( $\mathcal{G} = w$  za izlazni sloj, odnosno  $\mathcal{G} = v$  za sakriveni sloj), a  $\mathcal{G}(n+1)$  je nova vrijednost parametra učenja. Da bi se pogreška smanjivala najvećim mogućim iznosom potrebno je koristiti uvjet:

$$\Delta\mathcal{G} = -\eta\nabla E(\mathcal{G}) , \quad (2-12)$$

gdje je  $\eta$  mjera promjene pogreške učenja odnosno koeficijent brzine učenja, a  $\nabla E(\mathcal{G})$  gradijent pogreške koji je definiran izrazom:

$$\nabla E(\mathcal{G}) = \frac{\partial E(\mathcal{G})}{\partial \mathcal{G}} . \quad (2-13)$$

Koeficijent brzine učenja određuje učitelj, a njegova vrijednost se najčešće kreće između  $10^{-3}$  i 10. Ipak, bilo koja vrijednost koeficijenta brzine učenja ne osigurava uvijek dobru konvergenciju prema traženoj pogrešci ili traženim izlazima, već je potrebno eksperimentom utvrditi sa kojim vrijednostima se dobivaju najbolji rezultati. Ukoliko se izraz (2-12) uvrsti u izraz (2-11) tada se dobiva:

$$\mathcal{G}(n+1) = \mathcal{G}(n) - \eta\nabla E(\mathcal{G}(n)) . \quad (2-14)$$

Algoritam dan izrazom (2-14) je poznat i pod nazivom algoritam povratnog prostiranja pogreške te je upravo taj algoritam jedan od najčešće primjenivanih načina promjene parametara učenja. Najveći nedostatak ovog algoritma je veliki broj potrebnih iteracija odnosno koraka učenja. Upravo zato je ovaj algoritam modificiran – izrazu (2-12) dodana je modifikacija koja se naziva momentum odnosno količina gibanja:

$$\Delta \mathcal{G}(n) = -\eta \nabla E(\mathcal{G}(n)) + \alpha \Delta \mathcal{G}(n-1) , \quad (2-15)$$

gdje  $n$  označava trenutni korak učenja,  $n-1$  prethodni korak učenja, a  $\alpha$  momentum. Vrijednost momentuma određuje učitelj, a najčešće se nalazi u intervalu između 0.1 i 0.9 . Izrazom (2-15) brzina algoritma se u odnosu na klasični algoritam povećava i do 10 puta. Ukoliko se izraz (2-15) uvrsti u izraz (2-11) tada on poprima konačni oblik:

$$\mathcal{G}(n+1) = \mathcal{G}(n) - \eta \nabla E(\mathcal{G}(n)) + \alpha \Delta \mathcal{G}(n-1) . \quad (2-16)$$

Nakon što je izveden matematički model za povratnu fazu učenja koji vrijedi i za sakriveni i za izlazni sloj potrebno ga je primjeniti na svakom od ovih slojeva kako bi se dobile jednadžbe za promjenu težinskih faktora ovih slojeva. No, upravo se u ovom dijelu razlikuje učenje po uzorku skupa za učenje i učenje po skupu za učenje. Stoga će u sljedeća dva poglavlja biti izvedeni matematički modeli za svaku od procedura učenja.

## **2.4. Matematički model povratne faze učenja za učenje po uzorku skupa za učenje**

Kod povratne faze učenja za učenje po uzorku, promjena težinskih faktora odvija se za svaki ulazno-izlazni par (uzorak) iz skupa za učenje. To znači da će se za  $N_u$  uzoraka skupa za učenje, promjena težinskih faktora dogoditi  $N_u$  puta u jednom koraku učenja. Postupak promjene težinskih faktora polazi od izlaznog sloja prema ulaznom sloju, što znači da se u povratnoj fazi prvo mijenjaju težinski faktori izlaznog sloja, a onda težinski faktori sakrivenog sloja.

### ***Promjena težinskih faktora izlaznog sloja***

Promjena težinskih faktora izlaznog sloja odvija se prema izrazu (2-16) na sljedeći način:

$$w_{kj}(n+1) = w_{kj}(n) - \eta \nabla E(n) + \alpha \Delta w_{kj}(n-1) . \quad (2-17)$$

Gradijent pogreške za težine  $w_{kj}$  računa se prema izrazu (2-13):

$$\nabla E(n) = \frac{\partial E(n)}{\partial w_{kj}} \quad . \quad (2-18)$$

Osnovni zadatak čitavog postupka učenja jest određivanje pripadajućeg gradijenta pogreške danog prethodnim izrazom. Da se olakša rješavanje ovog problema primjenjuju se uzastopne parcijalne derivacije:

$$\frac{\partial E(n)}{\partial w_{kj}} = \frac{\partial E(n)}{\partial O_k} \frac{\partial O_k}{\partial net_{ok}} \frac{\partial net_{ok}}{\partial w_{kj}} \quad . \quad (2-19)$$

Vrijednosti pojedinih članova iz izraza (2-19) se mogu lako dobiti primjenom prethodno definiranih izraza. Tako se iz izraza (2-10) vidi da je:

$$\frac{\partial E(n)}{\partial O_k} = -(D_k - O_k) \quad . \quad (2-20)$$

Vrijednost drugog člana izraza (2-19) se dobiva iz izraza (2-9):

$$\frac{\partial O_k}{\partial net_{ok}} = 1 \quad . \quad (2-21)$$

Na osnovu izraza (2-7) može se odrediti i posljednja parcijalna derivacija izraza (2-19):

$$\frac{\partial net_{ok}}{\partial w_{kj}} = y_j \quad . \quad (2-22)$$

Uvrštavanjem izraza (2-20), (2-21) i (2-22) u izraz (2-19) odnosno izraz (2-18) dobiva se:

$$\nabla E(n) = \frac{\partial E(n)}{\partial w_{kj}} = -(D_k - O_k)y_j \quad . \quad (2-23)$$

Konačno, uvrštavanjem izraza (2-23) u izraz (2-17) dobiva se konačni algoritam promjene težinskih faktora izlaznog sloja:

$$w_{kj}(n+1) = w_{kj}(n) + \eta(D_k - O_k)y_j + \alpha \Delta w_{kj}(n-1) \quad . \quad (2-24)$$

### ***Promjena težinskih faktora sakrivenog sloja***

Kao što je već spomenuto, nakon promjene težinskih faktora  $w_{kj}$  izlaznog sloja slijedi promjena težinskih faktora  $v_{ji}$  sakrivenog sloja. Osnovna ideja ostaje ista, pa je temeljni izraz prema kojem se vrši podešavanje težinskih faktora sakrivenog sloja ponovno izraz (2-16):

$$v_{ji}(n+1) = v_{ji}(n) - \eta \nabla E(n) + \alpha \Delta v_{ji}(n-1) \quad . \quad (2-25)$$

Ponovno, jedini problem je izračunati gradijent pogreške  $\nabla E(n)$ . Primjenjujući parcijalno deriviranje prema slici 2.3. vidi se:

$$\frac{\partial E(n)}{\partial v_{ji}} = \frac{\partial E(n)}{\partial y_j} \frac{\partial y_j}{\partial net_{Hj}} \frac{\partial net_{Hj}}{\partial v_{ji}} . \quad (2-26)$$

Također, slika 2.3. pokazuje da na promjenu svakog težinskog faktora sakrivenog sloja utječe svaki neuron izlaznog sloja, pa prvi član u izrazu (2-26) poprima sljedeći oblik:

$$\begin{aligned} \frac{\partial E(n)}{\partial y_j} &= \frac{\partial E(n)}{\partial O_1} \frac{\partial O_1}{\partial net_{O1}} \frac{\partial net_{O1}}{\partial y_j} + \\ &+ \frac{\partial E(n)}{\partial O_2} \frac{\partial O_2}{\partial net_{O2}} \frac{\partial net_{O2}}{\partial y_j} + \\ &\vdots \quad \vdots \quad \vdots \quad \vdots \quad \vdots \\ &+ \frac{\partial E(n)}{\partial O_K} \frac{\partial O_K}{\partial net_{OK}} \frac{\partial net_{OK}}{\partial y_j} , \end{aligned} \quad (2-27)$$

gdje je:

$$\frac{\partial E(n)}{\partial O_k} = -(D_k - O_k) , \quad k = 1, 2, \dots, K , \quad (2-28)$$

$$\frac{\partial O_k}{\partial net_{Ok}} = 1 , \quad k = 1, 2, \dots, K , \quad (2-29)$$

$$\frac{\partial net_{Ok}}{\partial y_j} = w_{kj} , \quad k = 1, 2, \dots, K , \quad j = 1, 2, \dots, J - 1 . \quad (2-30)$$

Uvrštavanjem izraza (2-28), (2-29) i (2-30) u izraz (2-26) dobiva se konačni oblik prvog člana na desnoj strani izraza (2-26):

$$\frac{\partial E(n)}{\partial y_j} = - \sum_{k=1}^K (D_k - O_k) w_{kj} . \quad (2-31)$$

Drugi i treći član s desne strane izraza (2-26) određuju se iz izraza (2-5) i (2-4) kako slijedi:

$$\frac{\partial y_j}{\partial net_{Hj}} = \frac{1}{2} (1 - y_j^2) , \quad (2-32)$$

$$\frac{\partial net_{Hj}}{\partial v_{ji}} = Z_i . \quad (2-33)$$

Konačni oblik algoritma promjene težinskih faktora sakrivenog sloja dobiva se uvrštavanjem izraza (2-31), (2-32) i (2-33) u izraz (2-26):

$$v_{ji}(n+1) = v_{ji}(n) + \frac{1}{2} \eta (1 - y_j^2) Z_i \left( \sum_{k=1}^K (D_k - O_k) w_{kj} \right) + \alpha \Delta v_{ji}(n-1) . \quad (2-34)$$

Nakon što se dobiju nove vrijednosti težinskih faktora za izlazni i sakriveni sloj, te vrijednosti se proglašavaju važećima i sa njima se ide u unaprijednu fazu novog uzorka. Ovako izgleda prolazak kroz unaprijednu i povratnu fazu za jedan uzorak iz skupa učenja. Ako bi postojalo više uzoraka u skupu za učenje, tada bi slijedilo uzimanje ulaznih podataka novog uzorka, prolazak kroz unaprijednu fazu s tim ulaznim podacima i ponovno podešavanje težinskih faktora u povratnoj fazi. Prvi korak učenja bi završio onda kada bi svi uzorci prošli kroz unaprijednu i povratnu fazu učenja.

## 2.5. Matematički model povratne faze učenja za učenje po skupu za učenje

Kod učenja po skupu za učenje, promjena težinskih faktora se odvija nakon prolaska kroz cijeli skup za učenje. To znači da za bilo koji broj uzoraka  $N_u$  skupa za učenje, promjena težinskih faktora će se dogoditi samo jednom za jedan korak učenja. Da bi to bilo izvedivo potrebno je odrediti na koji način će se računati gradijent pogreške. Gradijent pogreške koji će se koristiti za promjenu težinskih faktora će biti suma gradijenata pogreške svih uzoraka iz skupa za učenje. Naravno, gradijenti pogreške izlaznog sloja se neće miješati sa gradijentima pogreške sakrivenog sloja.

### *Promjena težinskih faktora izlaznog sloja*

Izrazi za promjenu težinskih faktora izlaznog sloja kod učenja po uzorku skupa za učenje vrijede i ovdje, uz malu modifikaciju izraza (2-17). Već je rečeno da će se za gradijent pogreške koristiti suma gradijenata pogreške svih uzoraka iz skupa za učenje, pa izraz (2-17) poprima sljedeći oblik:

$$w_{kj}(n+1) = w_{kj}(n) - \eta \nabla E_{ukupni}(n) + \alpha \Delta w_{kj}(n-1) , \quad (2-35)$$

gdje  $\nabla E_{ukupni}(n)$  predstavlja gradijent pogreške za cijeli skup učenja, a dan je izrazom:

$$\nabla E_{ukupni}(n) = \sum_{n_u=1}^{N_u} \nabla E_{n_u}(n) , \quad (2-36)$$



gdje  $N_u$  predstavlja broj uzoraka u skupu za učenje.

Ako se izraz (2-36) uvrsti u izraz (2-35) dobiva se konačni algoritam promjene težinskih faktora izlaznog sloja za učenje po skupu za učenje:

$$w_{kj}(n+1) = w_{kj}(n) - \eta \sum_{n_u=1}^{N_u} \nabla E_{n_u}(n) + \alpha \Delta w_{kj}(n-1) . \quad (2-37)$$

### ***Promjena težinskih faktora sakrivenog sloja***

Kao i u prethodnom slučaju, i ovdje se radi o modifikaciji temeljne jednadžbe (2-25). Za učenje po skupu za učenje ona izgleda ovako:

$$v_{ji}(n+1) = v_{ji}(n) - \eta \nabla E_{ukupni}(n) + \alpha \Delta v_{ji}(n-1) . \quad (2-38)$$

Gradijent pogreške za cijeli skup učenja  $\nabla E_{ukupni}(n)$  već je dan izrazom (2-36), stoga preostaje prikazati konačni algoritam promjene težinskih faktora sakrivenog sloja:

$$v_{ji}(n+1) = v_{ji}(n) - \eta \sum_{n_u=1}^{N_u} \nabla E_{n_u}(n) + \alpha \Delta v_{ji}(n-1) . \quad (2-39)$$

## **2.6. Ocjena uspješnosti algoritma učenja**

Da bi se mogli uspoređivati načini učenja neuronskih mreža potrebno je definirati mjeru točnosti. Također, mjera točnosti služi i za procjenu uspješnosti algoritma učenja u rješavanju nekog postavljenog zadatka. U ovom radu, za usporedbu dvaju algoritama učenja, odabran je normalizirani korijen srednje kvadratne pogreške (engl. NRMS – Normalized Root Mean Square error) kao mjera točnosti:

$$NRMS = \frac{RMS}{\sigma_{d_n}} = \frac{\sqrt{\frac{\sum_{n=1}^N (D_n - O_n)^2}{N}}}{\sigma_{d_n}} , \quad (2-40)$$

gdje je  $\sigma_{d_n}$  izražena kao:

$$\sigma_{d_n} = \sqrt{\frac{1}{N} \sum_{n=1}^N (D_n - \bar{D})^2} , \quad (2-41)$$

$$\bar{D} = \frac{1}{N} \sum_{n=1}^N D_n , \quad (2-42)$$

gdje  $N$  predstavlja veličinu skupa za učenje.

Prednost ove mjere je njena bezdimenzionalnost koja osigurava neovisnost mjere o dimenzijama učenih veličina i koja omogućuje usporedbu izvedenih algoritama učenja s drugim algoritmima, neovisno o korištenoj programskoj podršci. Izračun NRMS mjere točnosti se u algoritmu radi nakon što mreža prođe kroz cijeli skup za učenje i sa novim vrijednostima težinskih faktora napravi jedan korak unaprijed i tako dobije vrijednosti izlaza sa novim težinskim faktorima. Ukoliko je vrijednost NRMS mjere točnosti manja ili jednaka traženoj vrijednosti, tada učenje završava.

### 3. Usporedba učenja po uzorku i učenja po skupu za učenje za klasifikacijski problem

U ovom poglavlju prikazat će se usporedba učenja po uzorku skupa za učenje i učenja po skupu za učenje za klasifikacijski problem. Usporedit će se rezultati učenja, brzina učenja te broj koraka i vrijeme učenja za zadanu grešku učenja. Kao primjer klasifikacijskog problema odabran je poznati logički problem Booleove algebre – ekskluzivni ILI (engl. XOR) problem koji je prikazan tablicom 3.1. Ovaj problem se sastoji od dva ulaza i jednog izlaza. I ulazi i izlaz su binarne varijable što znači da mogu poprimiti samo vrijednost nula ili jedan. Izlaz će poprimiti vrijednost jedan samo ako je jedan od ulaza jednak jedan. Ako ulazne varijable imaju istu vrijednost (nula ili jedan), tada će izlazna varijabla poprimiti vrijednost nula.

**Tablica 3.1.** XOR problem

Ulaz 1	Ulaz 2	Izlaz
0	0	0
0	1	1
1	0	1
1	1	0

Skup za učenje prikazan tablicom 3.1. može se definirati matricom ulaznih vrijednosti  $Z$  i vektorom željenih izlaznih vrijednosti  $D$ :

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

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

S matricom ulaznih vrijednosti  $Z$  proći će se kroz unaprijedne faze dvaju algoritama učenja, te će se dobiveni izlazi  $O$  usporediti sa traženim izlazima  $D$ . Zatim će se s razlikom traženih i dobivenih izlaza proći kroz povratnu fazu i izvršiti podešavanje težinskih faktora.

### 3.1. Početni parametri učenja

Da bi mreža mogla početi učiti potrebno je odrediti početne parametre – iznos broja koraka, vrijednost koeficijenta brzine učenja ( $\eta$ ), vrijednost momentuma ( $\alpha$ ) te vrijednost traženog NRMS-a. Pretpostavljeno je da su početni težinski faktori sakrivenog i izlaznog sloja zadani. Ispitivanje učenja neuronske mreže vršit će se za dva slučaja. U prvom slučaju vrijednost traženog NRMS-a neće biti zadana odnosno biti će nula, dok će za drugi slučaj vrijednost NRMS-a biti zadana.

#### *Prvi slučaj*

Tablicom 3.2. prikazani su početni parametri učenja mreže za klasifikacijski problem.

**Tablica 3.2.** Početni parametri učenja mreže za prvi slučaj

Broj koraka	5000
Koeficijent brzine učenja ( $\eta$ )	0,1
Momentum ( $\alpha$ )	0,8
Traženi NRMS	0

U ovom slučaju učenje mreže će se izvršiti za svih 5000 koraka, te će se kao rezultat učenja vidjeti vrijednost dobivenog NRMS-a.

#### *Drugi slučaj*

Za drugi slučaj sve vrijednosti tablice 3.2. ostaju iste osim vrijednosti NRMS-a koja će u ovom slučaju biti zadana, te vrijednosti momentuma koja će iznositi 0,7. Početni parametri učenja mreže za drugi slučaj prikazani su tablicom 3.3.

**Tablica 3.3.** Početni parametri učenja mreže za drugi slučaj

Broj koraka	5000
Koeficijent brzine učenja ( $\eta$ )	0,1
Momentum ( $\alpha$ )	0,7
Traženi NRMS	0,001

U ovom slučaju učenje mreže će se izvršiti za svih 5000 koraka ukoliko dobiveni NRMS bude veći od traženog NRMS-a. Ukoliko dobiveni NRMS u bilo kojem trenutku učenja mreže bude manji ili jednak traženom NRMS-u mreža će stati sa učenjem u tom koraku učenja. Na ovaj način će se vidjeti koji od ova dva algoritma za zadanu vrijednost NRMS-a daje manji broj koraka učenja.

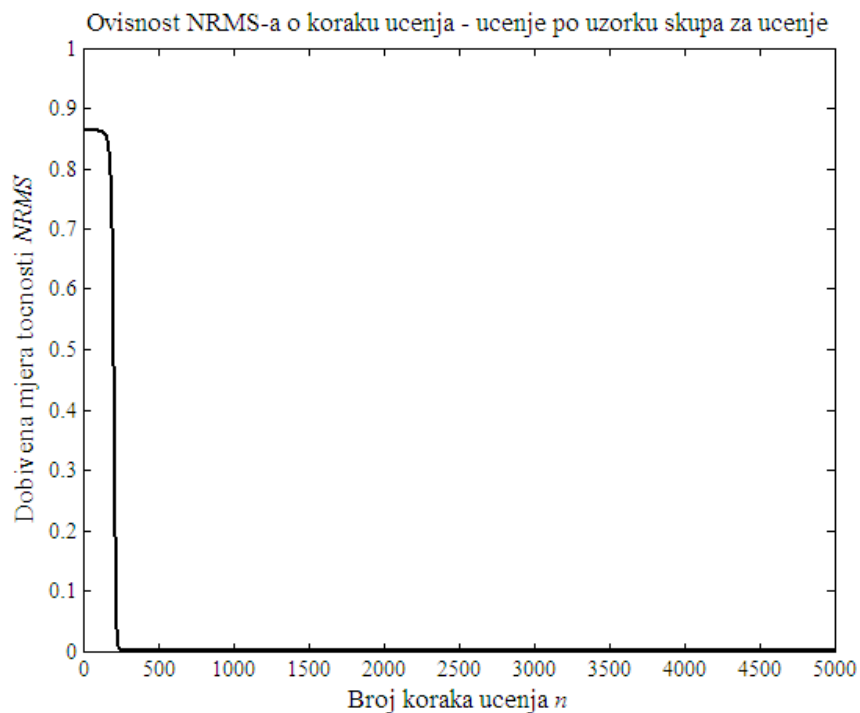
### 3.2. Usporedba rezultata učenja za učenje po uzorku i učenje po skupu za učenje

#### *Prvi slučaj*

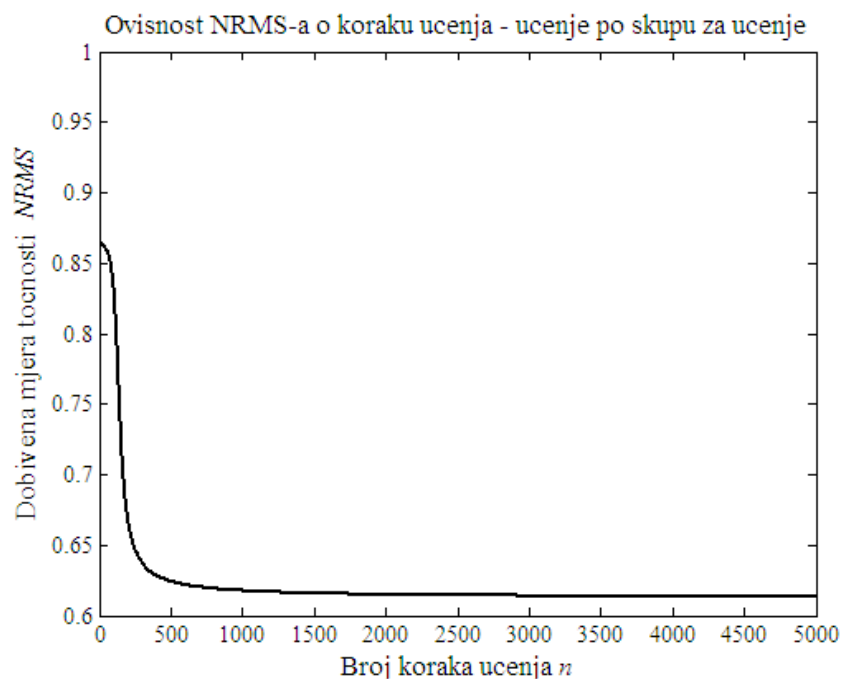
U tablici 3.4. dani su rezultati učenja za obje procedure, a slikama 3.1. i 3.2. prikazan je dijagram ovisnosti NRMS-a o koraku učenja za učenje po uzorku odnosno učenje po skupu.

**Tablica 3.4.** Usporedba rezultata učenja za prvi slučaj klasifikacijskog problema

Prvi slučaj	Učenje po uzorku	Učenje po skupu
Dobiveni NRMS	$1,1417 \cdot 10^{-15}$	0,6133
Vrijeme učenja mreže, s	2,094	2,016
Odziv naučene mreže, $O$	$\begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 0,0009 \\ 0,4992 \\ 0,9991 \\ 0,5008 \end{bmatrix}$



**Slika 3.1.** Dijagram ovisnosti NRMS-a i koraka učenja za učenje po uzorku (prvi slučaj)



**Slika 3.2.** Dijagram ovisnosti NRMS-a i koraka učenja za učenje po skupu (prvi slučaj)

Iz tablice 3.4. se vidi da za učenje po uzorku mreža vrlo dobro uči jer je greška učenja vrlo mala vrijednost. Isto tako, sa slike 3.1. se vidi da mreža do vrlo dobrih rezultata učenja dolazi već oko 200-og koraka učenja za učenje po uzorku. Gledajući rezultate učenja za učenje po skupu vidi se da mreža ne uči dobro zbog prevelike vrijednosti dobivenog NRMS-a ( $NRMS \approx 0,6$ ) te razlike odziva naučene mreže  $O$  i traženih odziva mreže  $D$ . Ovakav slučaj lošeg učenja mreže se pojavljuje zbog problema s lokalnim minimumima. Problem lokalnih minimuma događa se zbog loše odabranih vrijednosti koeficijenta brzine učenja i momentuma. Rješenje ovog problema dobiva se eksperimentalnim putem - ponovnim odabirom vrijednosti koeficijenta brzine učenja i momentuma. U tablici 3.5. prikazane su nove vrijednosti početnih parametara za prvi slučaj učenja mreže.

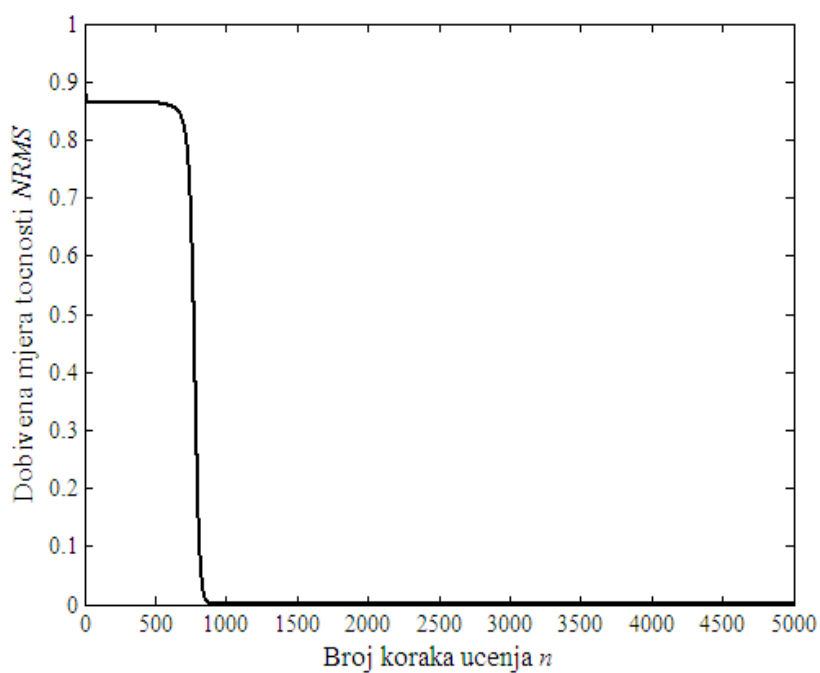
**Tablica 3.5.** Nove vrijednosti početnih parametara za prvi slučaj

Broj koraka	5000
Koeficijent brzine učenja ( $\eta$ )	0,1
Momentum ( $\alpha$ )	0,5
Traženi NRMS	0

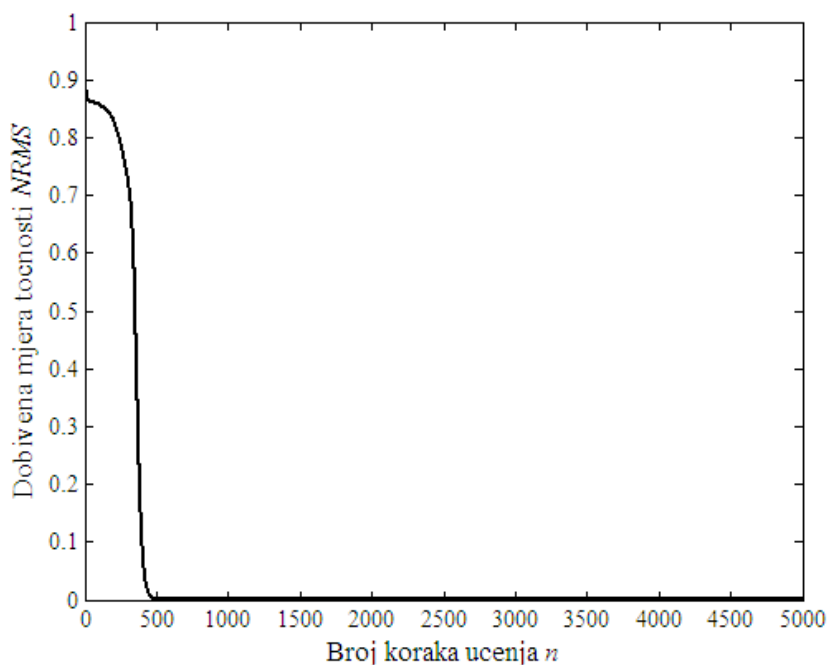
Nakon učenja mreže sa novim vrijednostima početnih parametara dobiveni su rezultati prikazani tablicom 3.6., te slikama 3.3. i 3.4.

**Tablica 3.6.** Usporedba rezultata učenja s novim vrijednostima početnih parametara za prvi slučaj klasifikacijskog problema

Prvi slučaj	Učenje po uzorku	Učenje po skupu
Dobiveni NRMS	$1,8369 \cdot 10^{-15}$	$1,7226 \cdot 10^{-15}$
Vrijeme učenja mreže, s	2,11	2
Odziv naučene mreže, $O$	$\begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}$	$\begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \end{bmatrix}$



**Slika 3.3.** Dijagram ovisnosti NRMS-a i koraka učenja za učenje po uzorku za nove vrijednosti početnih parametara (prvi slučaj)



**Slika 3.4.** Dijagram ovisnosti NRMS-a i koraka učenja za učenje po skupu za nove vrijednosti početnih parametara (prvi slučaj)

Iz novih rezultata učenja mreže vidi se da je učenje po skupu malo učinkovitije od učenja po uzorku što se tiče vremena učenja te dobivene greške učenja (NRMS). Također, vidi se da je primjenom učenja po skupu iznos broja koraka kod kojeg se dobiva prihvatljiva greška učenja (NRMS) puno manji (oko 500) nego kod učenja po uzorku (oko 1000).

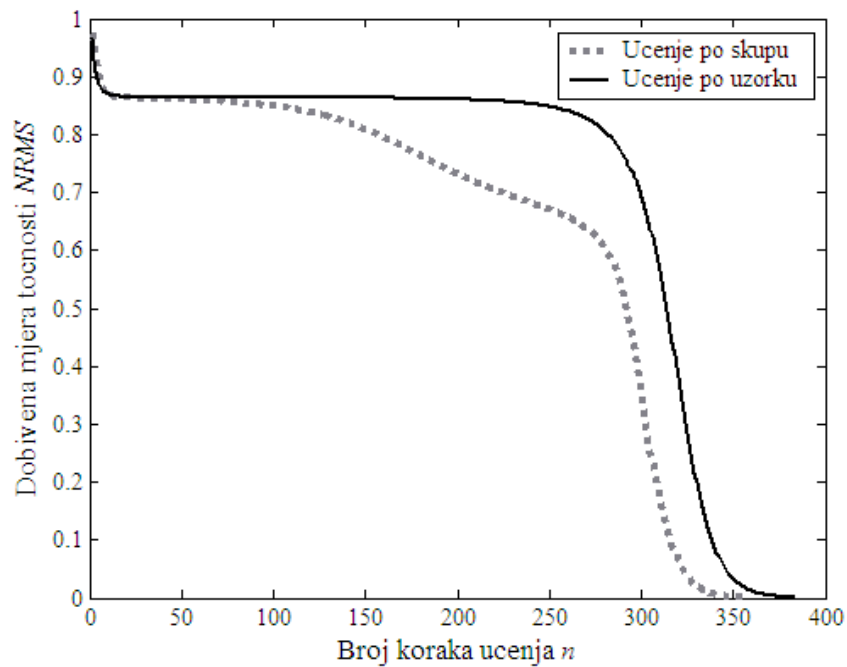
### ***Drugi slučaj***

Rezultati učenja mreže za obje procedure dani su u tablici 3.7., a slikom 3.5. prikazan je dijagram ovisnosti NRMS-a o koraku učenja za obje procedure učenja.

**Tablica 3.7.** Usporedba rezultata učenja za drugi slučaj klasifikacijskog problema

<b>Drugi slučaj</b>	Učenje po uzorku	Učenje po skupu
Dobiveni NRMS	$9,5821 \cdot 10^{-4}$	$8,9288 \cdot 10^{-4}$
Vrijeme učenja mreže, s	0,156	0,141
Potreban broj koraka učenja $n$	384	356
Odziv naučene mreže, $O$	$\begin{bmatrix} 0,0007 \\ 0,9994 \\ 0,9995 \\ 0,0004 \end{bmatrix}$	$\begin{bmatrix} 0,0006 \\ 0,9993 \\ 0,9996 \\ 0,0003 \end{bmatrix}$





**Slika 3.5.** Dijagram ovisnosti NRMS-a i koraka učenja za učenje po uzorku i učenje po skupu (drugi slučaj)

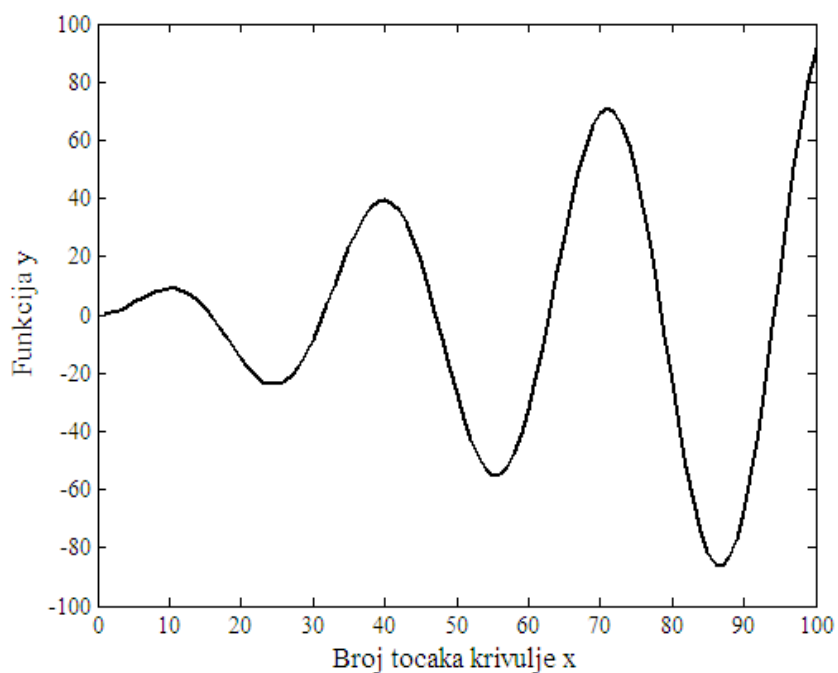
Iz ovih rezultata je vidljivo da će učenje mreže biti i kvalitetnije i brže ukoliko mreža uči po principu učenja po skupu za učenje. Princip učenja po skupu za učenje će dati kraće vrijeme učenja te manji broj potrebnih koraka učenja za traženu grešku učenja (NRMS). Također, ono što je još vidljivo iz tablice 3.7., jesu bolji rezultati odziva naučene mreže u korist učenja po skupu za učenje. Sa slike 3.5. se vidi i da u intervalu tražene greške učenja od 0,7 do 0,85 učenje po skupu za učenje daje posebno dobre rezultate odnosno puno manji potreban broj koraka učenja (do 200 koraka manje).

#### 4. Usporedba učenja po uzorku i učenja po skupu za učenje za regresijski problem

Regresijski problem učenja neuronske mreže je definiran kao provlačenje krivulje kroz neke zadane točke. U ovom radu krivulja za učenje je sinusoidna krivulja, te je zadana sa sto točaka, a mreži će na učenje biti dana svaka četvrta, pa svaka šesta točka. Također, mreži će biti zadana i vrijednost traženog NRMS-a. Krivulja koju će mreža učiti zadana je izrazom (4-1):

$$y = x \cdot \sin\left(\frac{x}{5}\right), \quad (4-1)$$

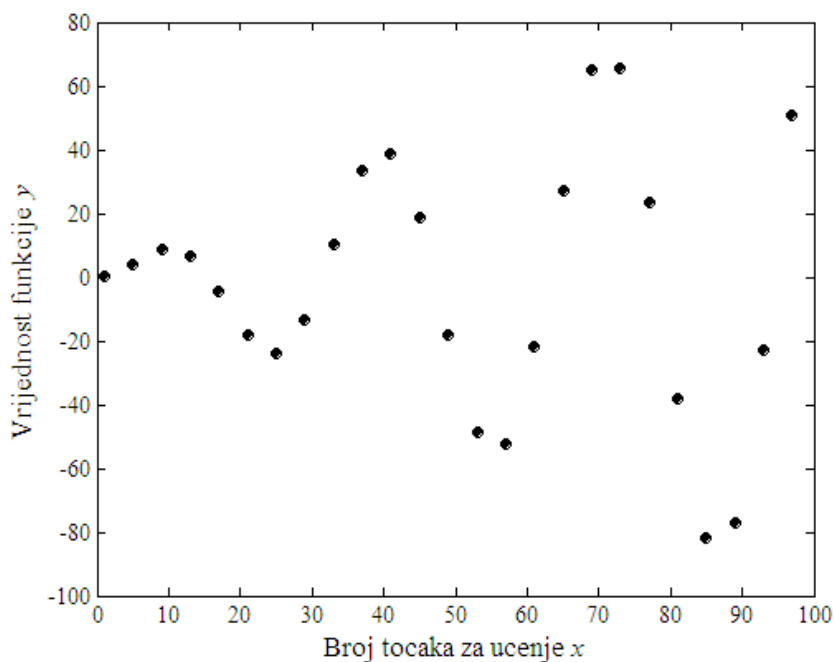
gdje  $x$  predstavlja broj točaka krivulje ( $x=1,2,\dots,100$ ). Na slici 4.1. prikazan je izgled krivulje.



**Slika 4.1.** Sinusoidna krivulja za učenje

## 4.1. Početni parametri učenja

Za prvi slučaj učenja mreže za regresijski problem uzeta je svaka četvrta točka zadane krivulje, a vrijednosti varijable  $x$  i pripadajućih varijabli  $y$  spremljene su u matricu učenja. Vrijednosti iz matrice učenja su zatim normirane odnosno podijeljene sa najvećim apsolutnim iznosom (za svaku varijablu zasebno) kako bi varijable poprimile vrijednosti u intervalu od -1 do 1, jer mreža uči samo s normiranim vrijednostima. Na slici 4.2. prikazane su točke dane za prvi slučaj učenja mreže.



Slika 4.2. Zadane točke učenja za prvi slučaj

Početni parametri (broj koraka učenja, koeficijent brzine učenja, momentum i NRMS) dani su tablicom 4.1.

Tablica 4.1. Početni parametri učenja mreže

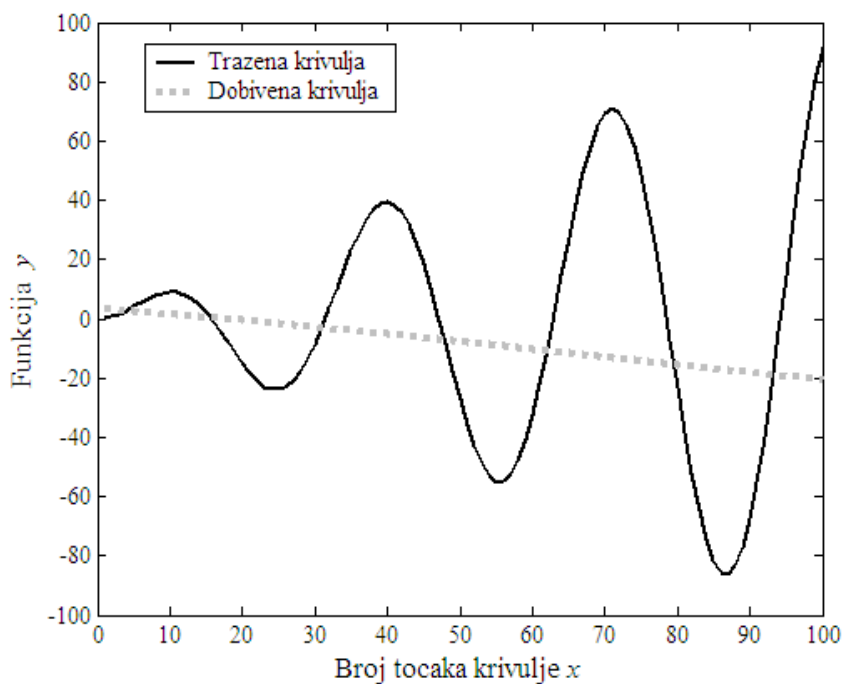
Broj koraka	10000
Koeficijent brzine učenja ( $\eta$ )	0,1
Momentum ( $\alpha$ )	0,5
Traženi NRMS	0,015

## 4.2. Usporedba rezultata učenja za učenje po uzorku i učenje po skupu za učenje

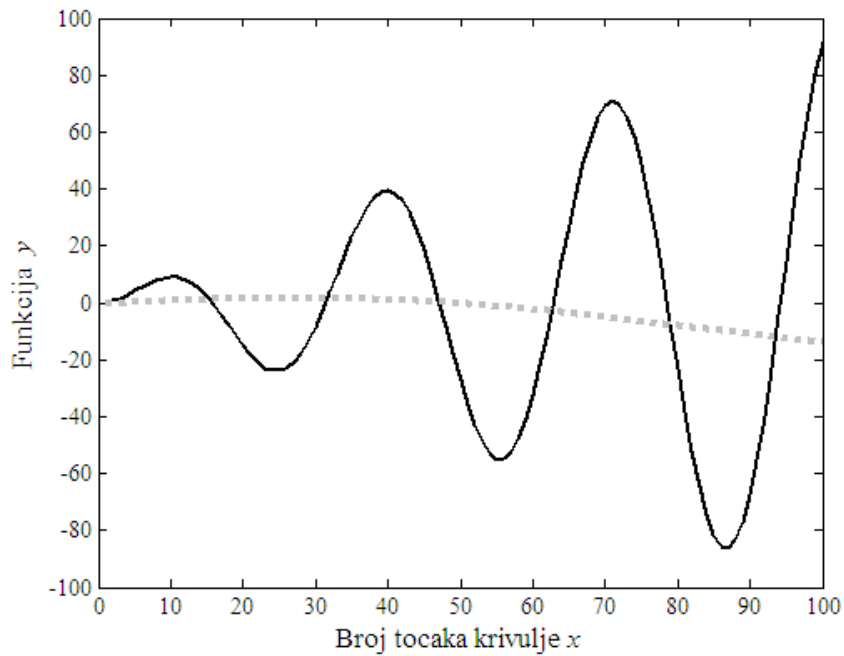
U tablici 4.2. dani su rezultati učenja za obje procedure, a na slikama 4.3. i 4.4. prikazana je krivulja dobivena učenjem i tražena krivulja.

**Tablica 4.2.** Usporedba rezultata učenja za regresijski problem

	Učenje po uzorku	Učenje po skupu
Dobiveni NRMS	0,9849	1,4602
Vrijeme učenja mreže, s	24,422	22,391
Potreban broj koraka učenja $n$	10000	10000

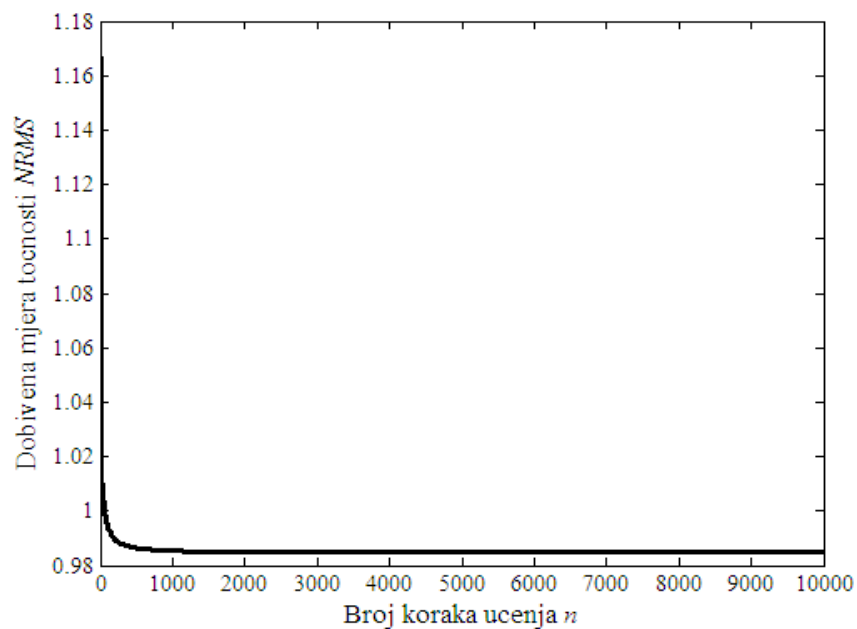


**Slika 4.3.** Usporedba dobivene krivulje i tražene krivulje za učenje po uzorku

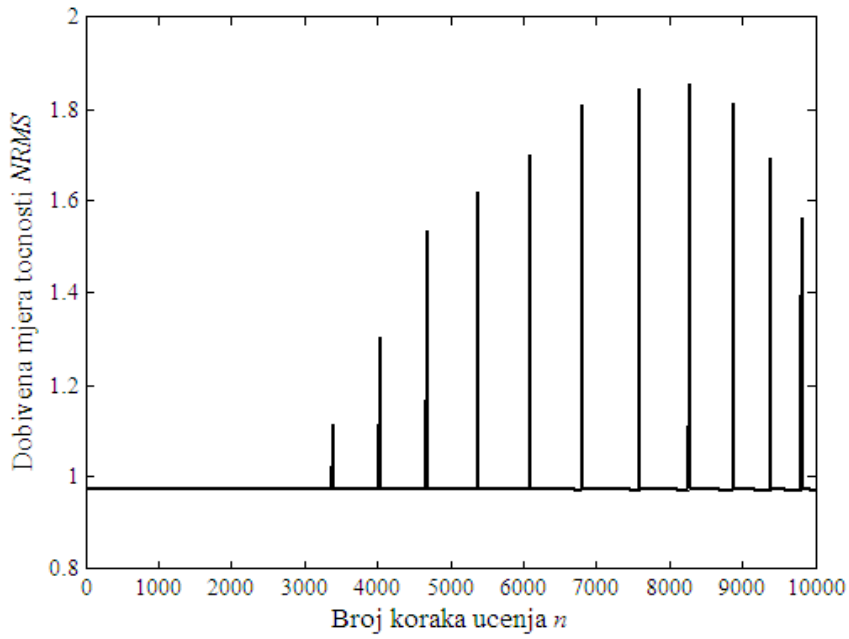


**Slika 4.4.** Usporedba dobivene krivulje i tražene krivulje za učenje po skupu

Ukoliko se pogledaju rezultati dobivenih vrijednosti NRMS-a na slikama 4.5. i 4.6., može se zaključiti da mreža loše uči i za učenje po skupu i za učenje po uzorku. Vidljivo je da bi i za veći broj koraka učenja vrijednost NRMS-a ostala ista. Ovaj problem se ne može okarakterizirati kao problem lokalnih minimuma jer i mijenjanjem vrijednosti koeficijenta brzine učenja i momentuma, vrijednost NRMS-a i dalje ostaje neprihvatljiva.

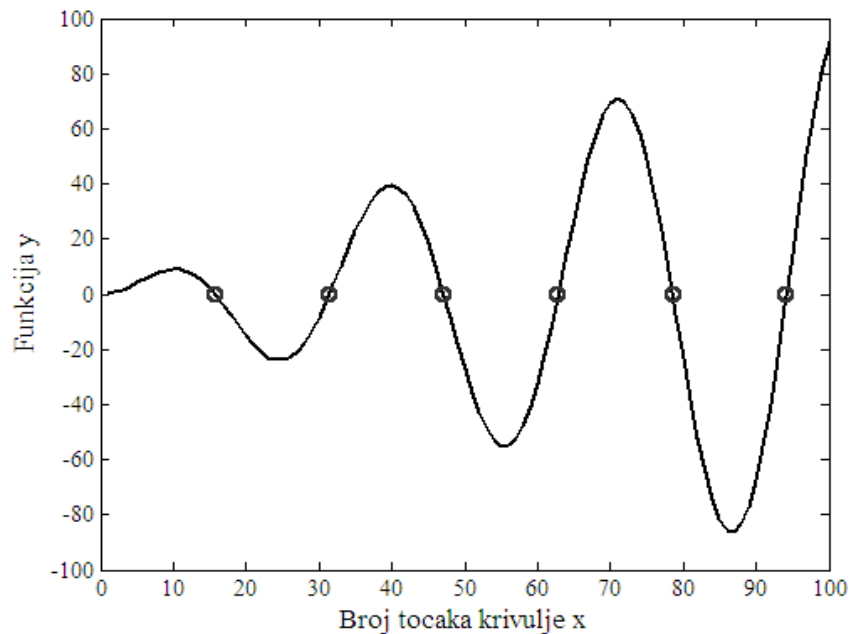


**Slika 4.5.** Dijagram ovisnosti NRMS-a i koraka učenja za učenje po uzorku (regresijski problem)



**Slika 4.6.** Dijagram ovisnosti NRMS-a i koraka učenja za učenje po skupu (regresijski problem)

Postavlja se pitanje zašto mreža loše uči? Razlog lošeg učenja mreže jest odabrana krivulja za učenje. Iz izraza (4-1) vidi se da će krivulja za različite vrijednosti varijable  $x$  dati istu vrijednost varijable  $y$ , a to je prikazano na slici 4.7. Upravo to ponavljanje vrijednosti varijable  $y$  za različite vrijednosti varijable  $x$  stvara problem ovakvoj vrsti mreže pri učenju, pa čak ni povećanje broja koraka ne rješava problem. Da bi se uspio riješiti ovaj problem, potrebno je odabrati prikladniju vrstu mreže za učenje.



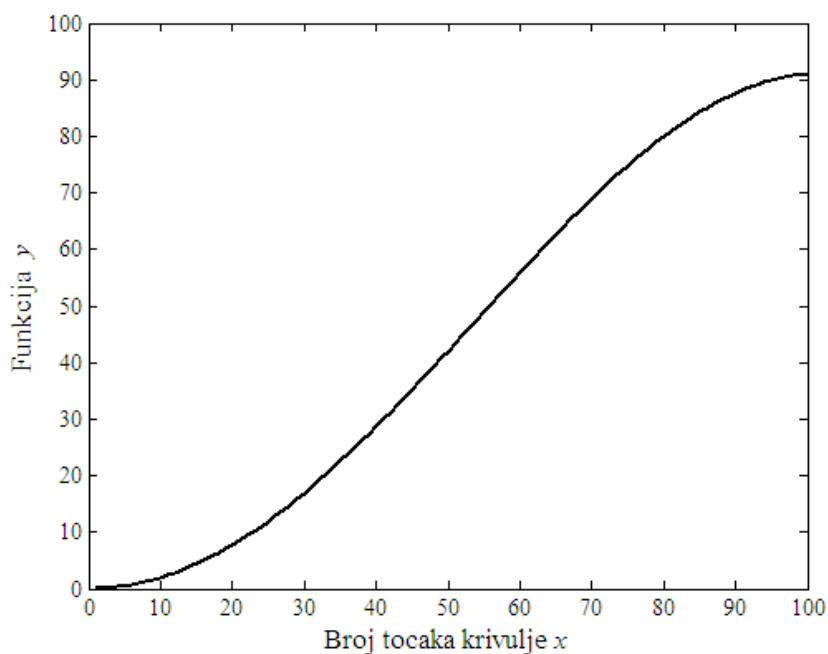
**Slika 4.7.** Prikaz točaka sa istom vrijednošću varijable  $y$  za različite vrijednosti varijable  $x$

### 4.3. Nova krivulja za učenje mreže

Zbog razloga navedenih u prethodnom potpoglavlju, odabrat će se nova krivulja za učenje koja će biti jednostavnija od prethodno zadane. Nova krivulja je definirana izrazom (4-2) i uvjetom (4-3) i prikazana na slici 4.8.:

$$y = x \cdot \sin\left(\frac{x}{50}\right), \quad (4-2)$$

$$x = 1, 2, \dots, 100. \quad (4-3)$$



Slika 4.8. Nova krivulja za učenje

### 4.4. Početni parametri učenja (nova krivulja)

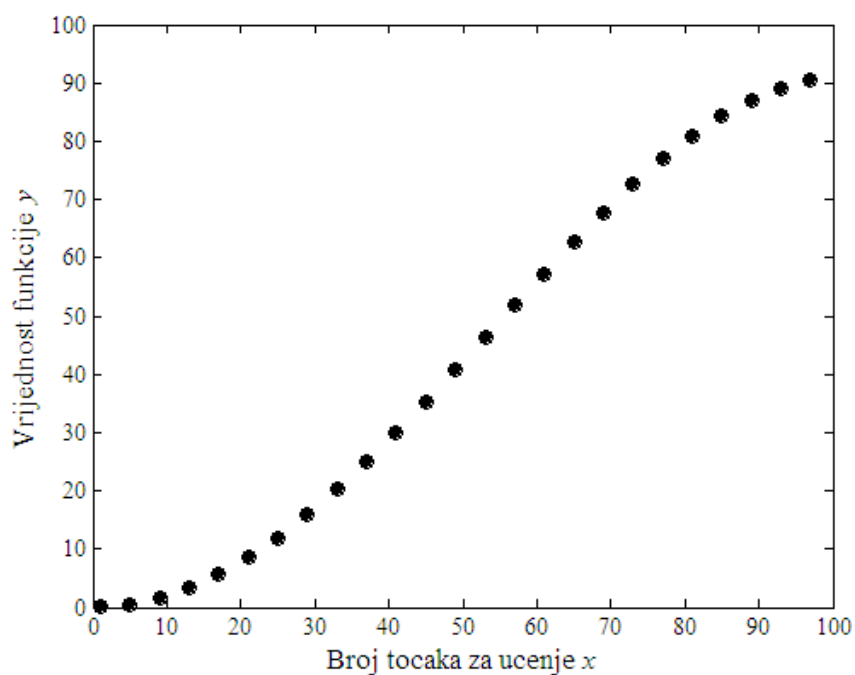
Učenje mreže će se provoditi na dva slučaja. U prvom slučaju mreži će za učenje biti dana svaka četvrta točka krivulje, a u drugom slučaju svaka šesta točka krivulje. Za oba slučaja koristit će se početni parametri zadani tablicom 4.3. – jednom bez zadane vrijednosti momentuma, a drugi put sa zadanom vrijednosti momentuma. Nakon učenja, mreža će biti testirana sa dobivenim vrijednostima težinskih faktora.

**Tablica 4.3.** Početni parametri učenja mreže (nova krivulja)

	Bez momentuma	Sa momentumom
Broj koraka	30000	30000
Koeficijent brzine učenja ( $\eta$ )	0,05	0,05
Momentum ( $\alpha$ )	0	0,8
Traženi NRMS	0,015	0,015

***Prvi slučaj (učenje svake četvrte točke krivulje)***

Na slici 4.9. prikazane su zadane točke krivulje za učenje mreže.

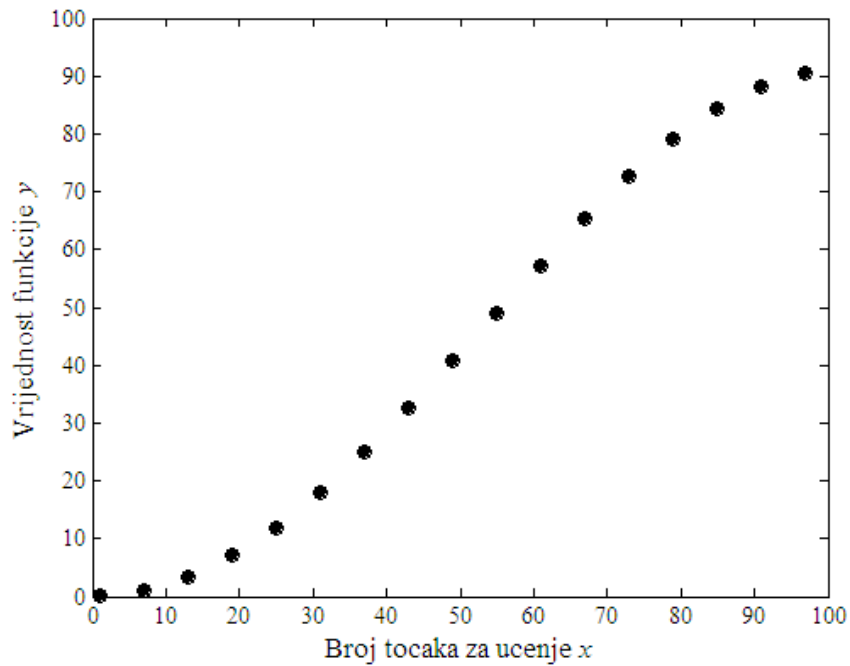


**Slika 4.9.** Zadane točke učenja nove krivulje za prvi slučaj

***Drugi slučaj (učenje svake šeste točke krivulje)***

Na slici 4.10. prikazane su zadane točke krivulje za učenje mreže. Vrijednosti ostalih početnih parametara iste su kao i kod prethodnog slučaja i prikazane su tablicom 4.3.





**Slika 4.10.** Zadane točke učenja nove krivulje za drugi slučaj

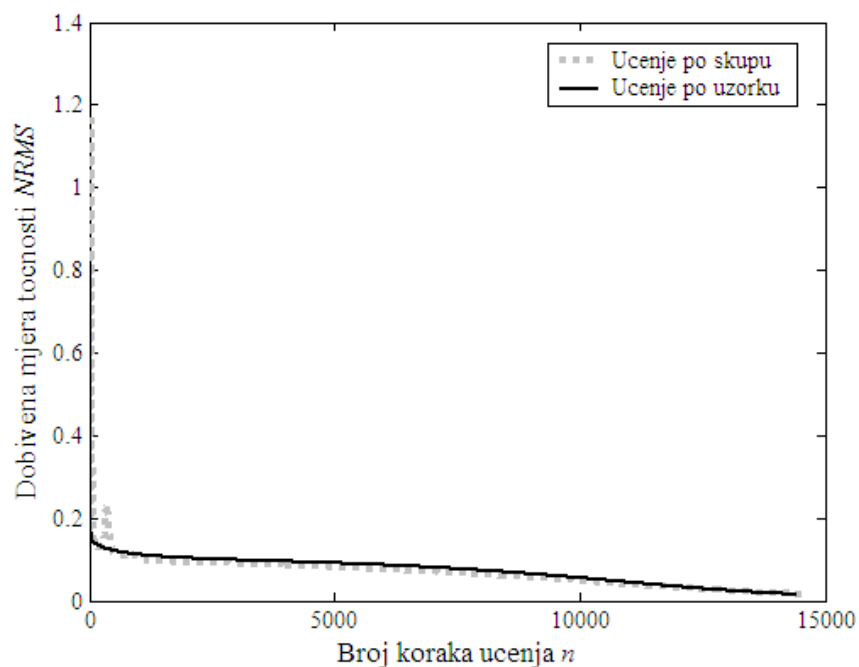
#### 4.5. Usporedba rezultata učenja za učenje po uzorku i učenje po skupu za učenje

##### *Prvi slučaj (učenje svake četvrte točke krivulje) – bez momentuma*

U tablici 4.4. dani su rezultati učenja za obje procedure kada vrijednost momentuma nije zadana, a slikom 4.11. prikazana je usporedba između dobivenog NRMS-a za učenje po uzorku i NRMS-a za učenje po skupu za učenje.

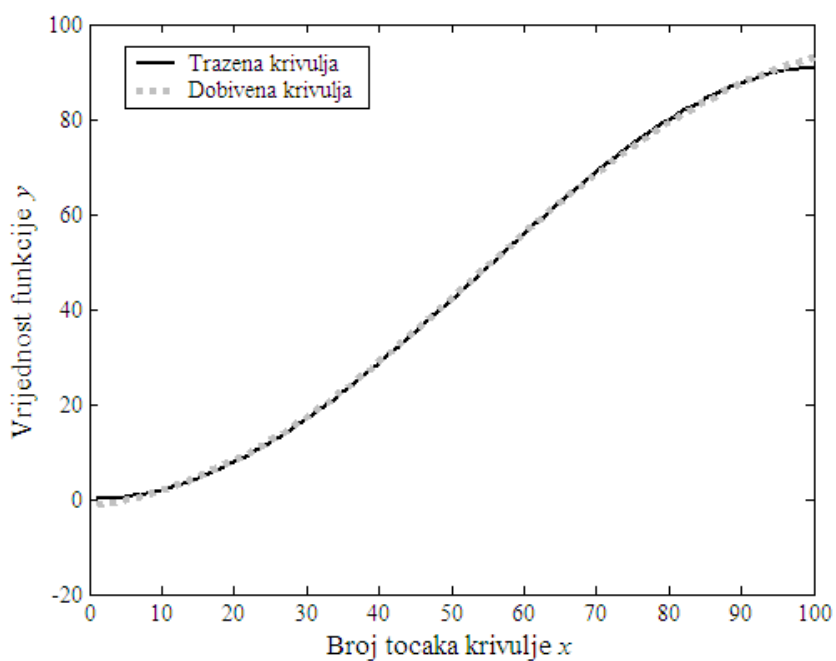
**Tablica 4.4.** Usporedba rezultata učenja za prvi slučaj regresijskog problema (bez momentuma)

<b>Prvi slučaj – bez momentuma</b>	Učenje po uzorku	Učenje po skupu
Dobiveni NRMS	0,0150	0,0150
Vrijeme učenja mreže, s	35,765	32,953
Potreban broj koraka učenja $n$	14423	14642

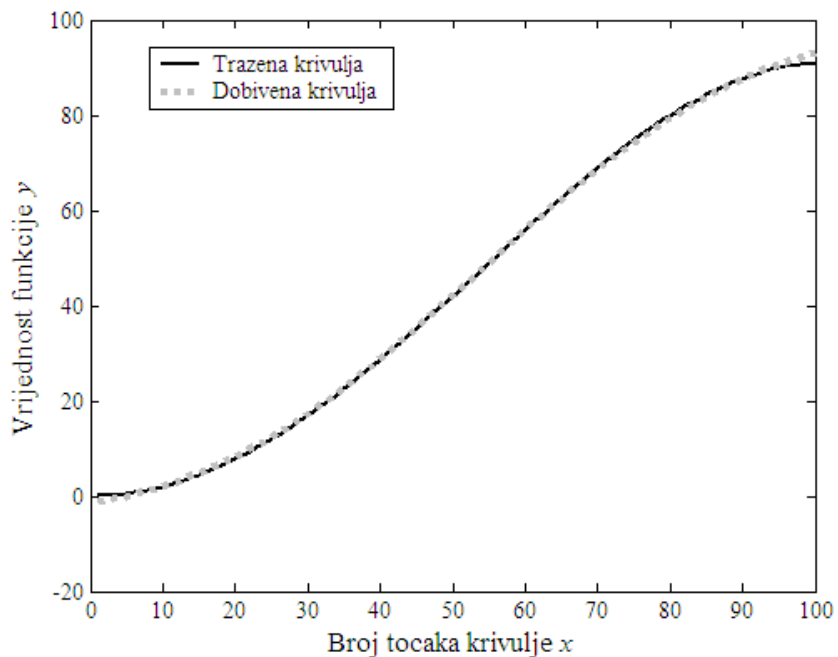


**Slika 4.11.** Dijagram ovisnosti NRMS-a i koraka učenja za učenje po uzorku i učenje po skupu (prvi slučaj – bez momentuma)

Iz ovih je rezultata vidljivo da učenje po uzorku za regresijski problem daje bolje rezultate za zadane početne parametre. Unatoč nešto većem potrošenom vremenu učenja, učenje po uzorku koristi manji broj koraka učenja za ostvarivanje traženog NRMS-a. Rezultati testiranja prikazani slikama 4.12. i 4.13. pokazuju da su obje procedure izvršile učenje za zadani NRMS.



**Slika 4.12.** Usporedba dobivene i tražene krivulje za učenje po uzorku (prvi slučaj – bez momentuma)



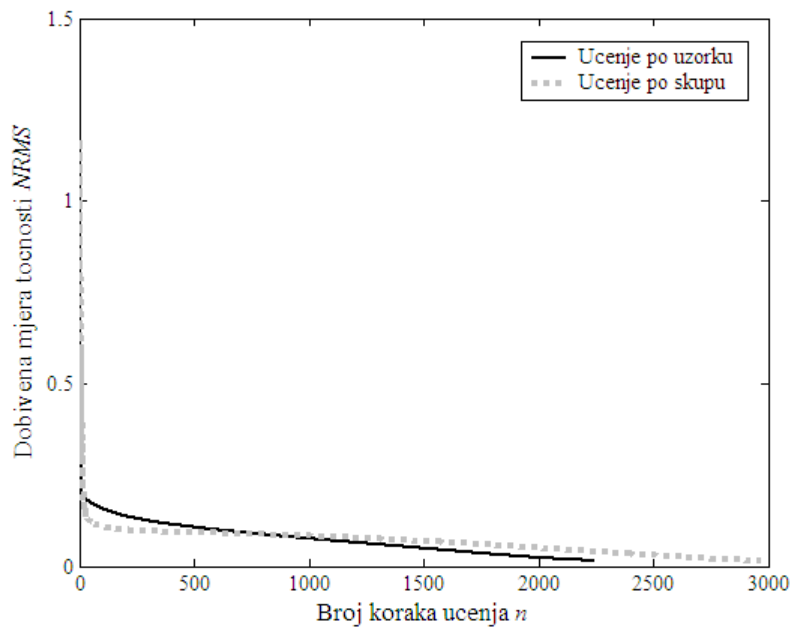
**Slika 4.13.** Usporedba dobivene i tražene krivulje za učenje po skupu (prvi slučaj – bez momentuma)

***Prvi slučaj (učenje svake četvrte točke krivulje) – sa momentumom***

U tablici 4.5. dani su rezultati učenja za obje procedure kada je vrijednost momentuma zadana, a slikom 4.14. prikazana je usporedba između dobivenog NRMS-a za učenje po uzorku i NRMS-a za učenje po skupu za učenje.

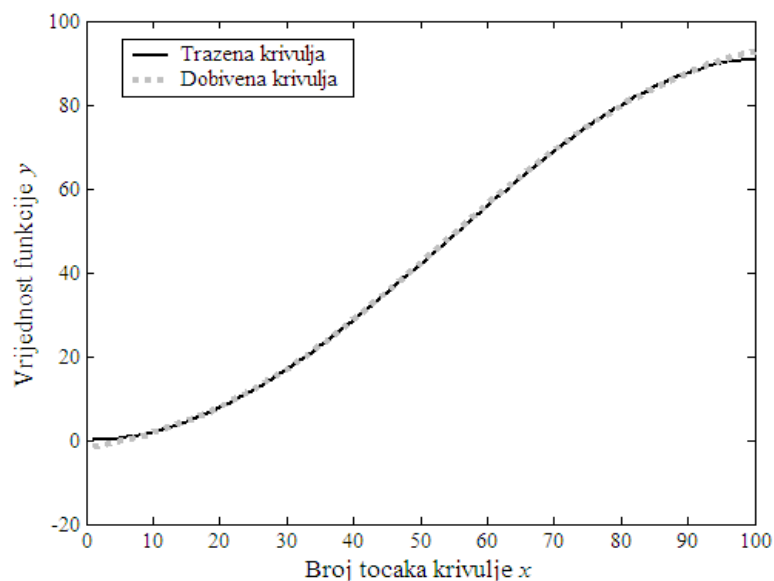
**Tablica 4.5.** Usporedba rezultata učenja za prvi slučaj regresijskog problema (sa momentumom)

<b>Prvi slučaj – sa momentumom</b>	Učenje po uzorku	Učenje po skupu
Dobiveni NRMS	0,0150	0,0150
Vrijeme učenja mreže, s	5,485	6,672
Potreban broj koraka učenja $n$	2244	2972

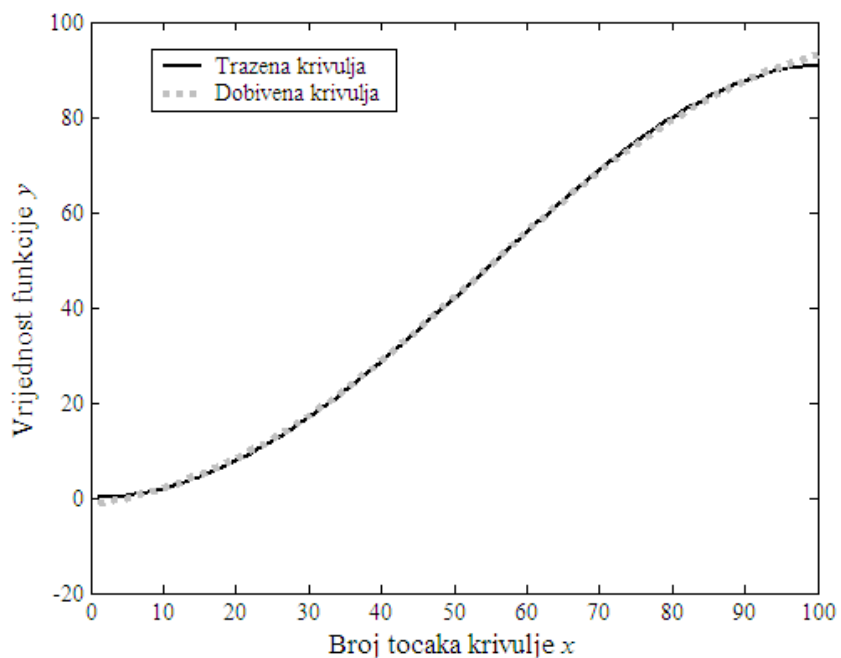


**Slika 4.14.** Dijagram ovisnosti NRMS-a i koraka učenja za učenje po uzorku i učenje po skupu (prvi slučaj – sa momentumom)

Promatrajući dobivene rezultate, mogu se izvesti dva zaključka. Prvi zaključak je da ako se u proceduri učenja koristi momentum, tada mreža osjetno brže uči nego kada se momentum ne koristi. Drugi zaključak se odnosi na usporedbu dvaju procedura učenja. Kada je zadana vrijednost momentuma, još više dolazi do izražaja superiornost učenja po uzorku. Vidljivo je da ova procedura u ovom slučaju koristi manji broj koraka učenja i još k tome kraće procesorsko vrijeme. Na slikama 4.15. i 4.16. pokazano je da obje procedure dobro uče za zadani NRMS.



**Slika 4.15.** Usporedba dobivene i tražene krivulje za učenje po uzorku (prvi slučaj – sa momentumom)



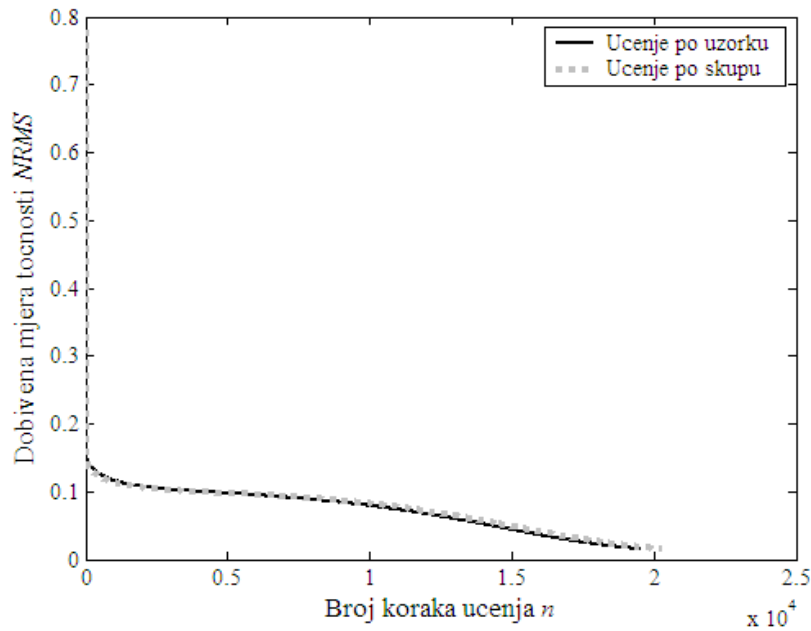
**Slika 4.16.** Usporedba dobivene i tražene krivulje za učenje po skupu (prvi slučaj – sa momentumom)

***Drugi slučaj (učenje svake šeste točke krivulje) – bez momentuma***

U tablici 4.6. dani su rezultati učenja za obje procedure kada momentum nije zadan, a slikom 4.17. prikazana je usporedba između dobivenog NRMS-a za učenje po uzorku i NRMS-a za učenje po skupu za učenje.

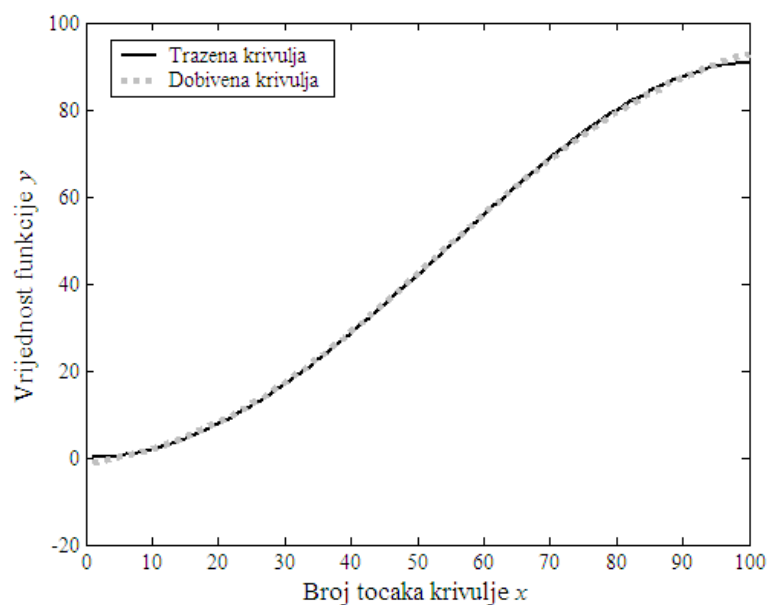
**Tablica 4.6.** Usporedba rezultata učenja za drugi slučaj regresijskog problema (bez momentuma)

<b>Drugi slučaj – bez momentuma</b>	Učenje po uzorku	Učenje po skupu
Dobiveni NRMS	0,015	0,015
Vrijeme učenja mreže, s	33,594	32,406
Potreban broj koraka učenja $n$	19533	20328

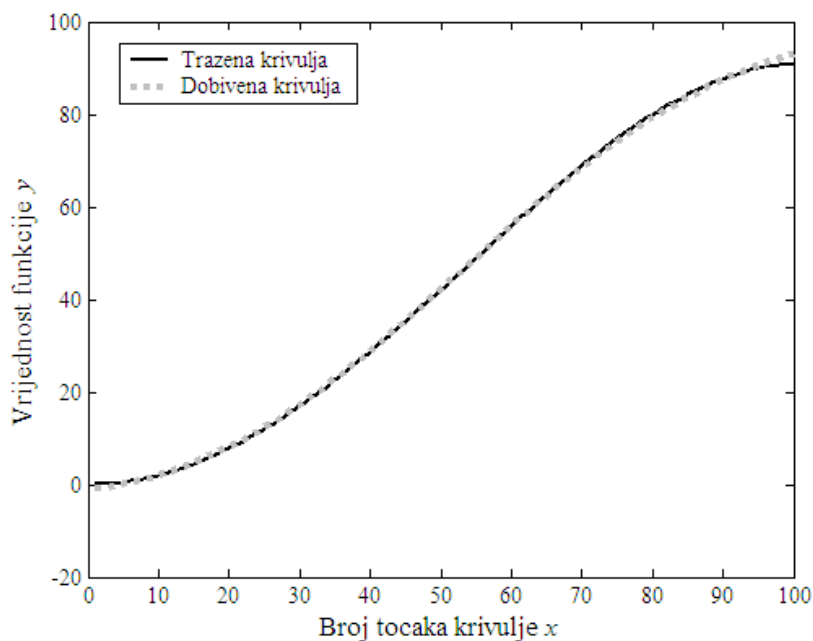


**Slika 4.17.** Dijagram ovisnosti NRMS-a i koraka učenja za učenje po uzorku i učenje po skupu (drugi slučaj – bez momentuma)

Ponovno se može primjetiti da učenje po uzorku troši nešto više vremena za učenje ali zato koristi puno manji broj koraka da dostigne traženu vrijednost NRMS-a. Ako se rezultati drugog slučaja usporede sa rezultatima prvog slučaja, vidi se da je vrijeme učenja mreže manje kod drugog slučaja (zbog manjeg broja ulaznih podataka – svaka šesta točka se koristi za ulazni podatak), dok je broj potrebnih koraka da bi se dobio traženi NRMS veći za drugi slučaj. Rezultati testiranja prikazani slikama 4.18. i 4.19. opet potvrđuju činjenicu da obje procedure dobro uče.



**Slika 4.18.** Usporedba dobivene i tražene krivulje za učenje po uzorku (drugi slučaj – bez momentuma)



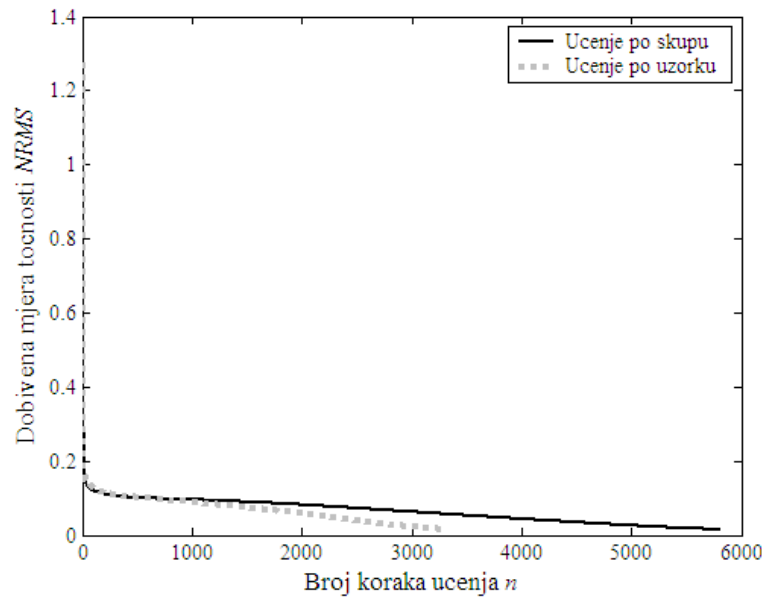
**Slika 4.19.** Usporedba dobivene i tražene krivulje za učenje po skupu (drugi slučaj – bez momentuma)

***Drugi slučaj (učenje svake šeste točke krivulje) – sa momentumom***

U tablici 4.7. dani su rezultati učenja za obje procedure kada je momentum zadan, a slikom 4.20. prikazana je usporedba između dobivenog NRMS-a za učenje po uzorku i NRMS-a za učenje po skupu za učenje.

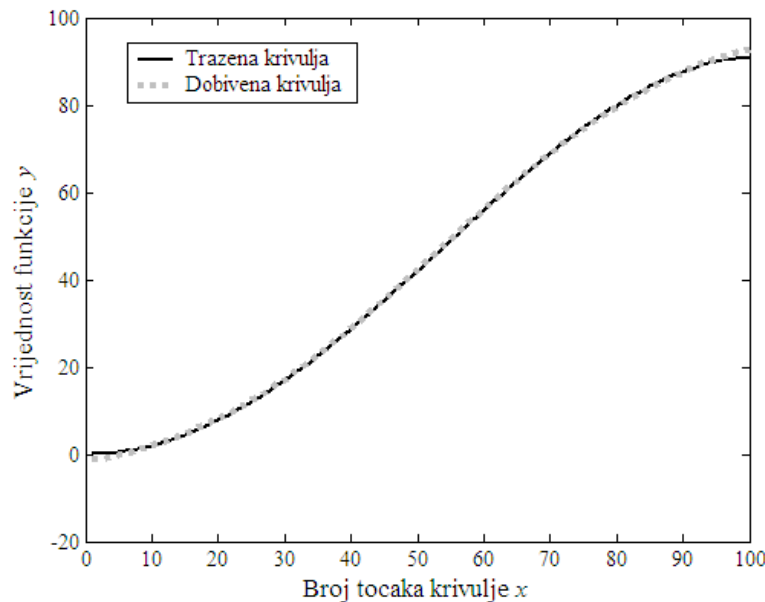
**Tablica 4.7.** Usporedba rezultata učenja za drugi slučaj regresijskog problema (sa momentumom)

<b>Drugi slučaj – sa momentumom</b>	Učenje po uzorku	Učenje po skupu
Dobiveni NRMS	0,015	0,015
Vrijeme učenja mreže, s	5,546	9,000
Potreban broj koraka učenja $n$	3320	5809



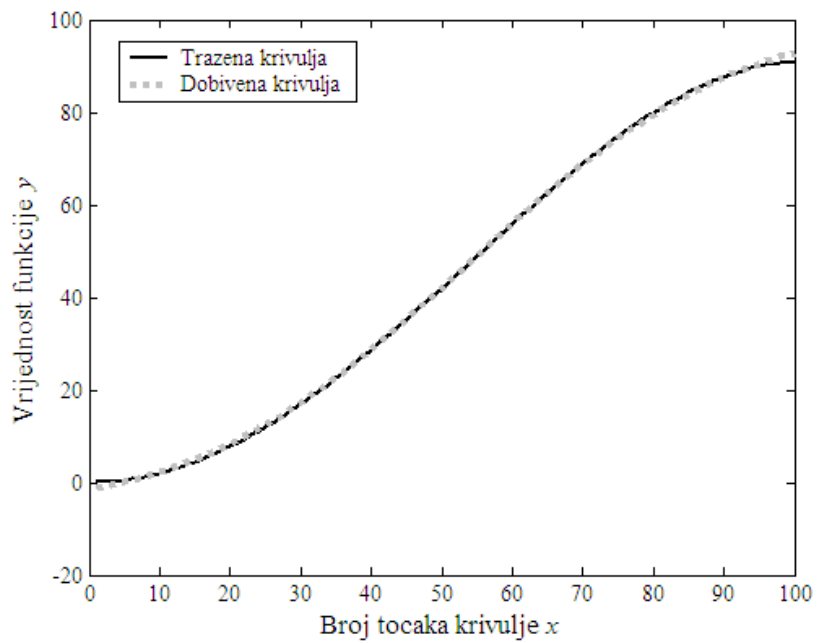
**Slika 4.20.** Dijagram ovisnosti NRMS-a i koraka učenja za učenje po uzorku i učenje po skupu (drugi slučaj – sa momentumom)

Ovim rezultatima je samo potvrđena činjenica da korištenje momentuma znatno ubrzava proces učenja mreže za obje procedure. Također, opet su potvrđeni zaključci iz prvog slučaja kod učenja svake četvrte točke krivulje – procedura učenja po uzorku daje puno bolje rezultate (manje utrošeno vrijeme učenja te manji broj koraka učenja – čak 2500 koraka manje!) od procedure učenja po skupu. Na slikama 4.21. i 4.22. prikazano je da su obje procedure učenja u stanju naučiti mrežu zadanom problemu, ali, kao što je već rečeno, procedura učenja po uzorku će to učiniti kvalitetnije.



**Slika 4.21.** Usporedba dobivene i tražene krivulje za učenje po uzorku (drugi slučaj – sa momentumom)





**Slika 4.22.** Usporedba dobivene i tražene krivulje za učenje po skupu (drugi slučaj – sa momentumom)

## 5. Zaključak

U ovom radu pokazane su različitosti dvaju procedura učenja statičke neuronske mreže sa povratnim prostiranjem pogreške – učenja po uzorku skupa za učenje i učenja po skupu za učenje. Promatrajući rezultate klasifikacijskog problema, vidi se da bolje rezultate za sve ispitivane parametre daje učenje po skupu za učenje (tzv. “batch” procedura). S druge strane, promatrajući rezultate regresijskog problema, učenje mreže po skupu za učenje ipak ne ostvaruje bolje rezultate od učenja mreže po uzorku. Štoviše, dobiveni rezultati učenja za regresijski problem pokazuju da je procedura učenja mreže po uzorku puno kvalitetnija što se tiče broja koraka i vremena učenja. Ono što još ide u korist proceduri učenja po uzorku jest da daje bolje rezultate za relativno složeniji problem. U ovom radu potvrđena je još jedna činjenica – korištenje momentuma u algoritmu učenja znatno ubrzava proces učenja. Iz ovih saznanja može se zaključiti da je procedura učenja po uzorku kvalitetnija i bolja od procedure učenja po skupu. Ipak, to treba uzeti sa zadržkom, jer za neke jednostavnije probleme učenje po skupu će možda dati bolje rezultate kao što je to vidljivo iz rezultata klasifikacijskog problema. Znači, za određeni problem potrebno je pravilno odabrati proceduru učenja kako bi se dobili što optimalniji rezultati. Kada se to tako učini, umjetne neuronske mreže mogu postati vrlo moćan alat za rješavanje bilo kakvog problema.

## Literatura

- [1] NIDA: *Sending and Receiving Messages*, [www.drugabuse.gov/JSP/MOD3/page3.html](http://www.drugabuse.gov/JSP/MOD3/page3.html), 22.02.2009.
- [2] Novaković, B., Majetić, D., Široki, M.: *Umjetne neuronske mreže*, Fakultet strojarstva i brodogradnje, Sveučilišta u Zagrebu, Zagreb, 1998.
- [3] ERIS: *Mreža "širenje unatrag"*, [eris.foi.hr/11neuronske/nn-predavanje5.html](http://eris.foi.hr/11neuronske/nn-predavanje5.html), 22.02.2009.