

Predviđanje vremenskih serija primjenom umjetnih neuronskih mreža

Kukurin, Luka

Undergraduate thesis / Završni rad

2020

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:235:457762>

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

Download date / Datum preuzimanja: **2024-05-13**

Repository / Repozitorij:

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



SVEUČILIŠTE U ZAGREBU
FAKULTET STROJARSTVA I BRODOGRADNJE

ZAVRŠNI RAD

Luka Kukurin

Zagreb, 2020.

SVEUČILIŠTE U ZAGREBU
FAKULTET STROJARSTVA I BRODOGRADNJE

ZAVRŠNI RAD

Mentori:

Izv. prof. dr. sc. Danko Brezak, dipl. ing.

Student:

Luka Kukurin

Zagreb, 2020.

Izjavljujem da sam ovaj rad izradio samostalno koristeći znanja stečena tijekom studija i navedenu literaturu.

Zahvaljujem se mentoru Danku Brezaku na podršci.

Luka Kukurin



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: 602 - 04 / 20 - 6 / 3	
Ur. broj: 15 - 1703 - 20 -	

ZAVRŠNI ZADATAK

Student: **Luka Kukurin** Mat. br.: 0035204686

Naslov rada na hrvatskom jeziku: **Predviđanje vremenskih serija primjenom umjetnih neuronskih mreža**

Naslov rada na engleskom jeziku: **Time series forecasting using artificial neural networks**

Opis zadatka:

Predviđanja vremenskih serija imaju za cilj procjenu budućih vrijednosti serija na temelju prošlih vrijednosti. Procesi kod kojih se promjene u vrijednostima parametara mogu opisati dinamikom vremenskih serija često su izrazito nelinearne prirode i vrlo zahtjevni za modeliranje. U tom se smislu umjetne neuronske mreže nameću kao logičan alat za primjenu u rješavanju navedenih vrsta problema. Stoga je u ovom radu potrebno usporediti dvije vrste neuronskih mreža, statičke i dinamičke mreže, na problemu predviđanja Mackey-Glass vremenske serije.

U radu je potrebno:

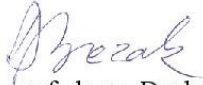
1. Izraditi programsku podršku za statičku (eng. *feed-forward*) neuronsku mrežu.
2. Izraditi programsku podršku za dinamičku (eng. *recurrent*) neuronsku mrežu.
3. Generirati skupove podataka za učenje i testiranje analiziranih algoritama mreža.
4. Usporediti i analizirati dobivene odzive neuronskih mreža.
5. Izvesti zaključke rada.

Zadatak zadan:
15. svibnja 2020.

Datum predaje rada:
2. rok (izvanredni): 1. srpnja 2020.
3. rok: 17. rujna 2020.

Predviđeni datumi obrane:
2. rok (izvanredni): 3.7.2020.
3. rok: 21.9. - 25.9.2020.

Zadatak zadao:


Izv. prof. dr. sc. Danko Brezak

Predsjednik Povjerenstva:

Prof. dr. sc. Branko Bauer

SADRŽAJ

SADRŽAJ	I
POPIS SLIKA	III
POPIS TABLICA	IV
POPIS OZNAKA	V
SAŽETAK	VII
SUMMARY	VIII
1. UVOD	1
2. Statička neuronska mreža	2
2.1. Model statičkog neurona	2
2.2. Model statičke neuronske mreže	3
2.3. Unaprijedna faza učenja statičke mreže	4
2.3.1. Sakriveni sloj	4
2.3.2. Izlazni sloj	5
2.4. Povratna faza učenja statičke mreže	5
2.4.1. Promjena težina izlaznog sloja	6
2.4.2. Promjena težina sakrivenog sloja	7
3. Dinamička neuronska mreža	9
3.1. Model dinamičkog neurona	9
3.2. Model dinamičke neuronske mreže	10
3.3. Postupak učenja dinamičke mreže	11
3.3.1. Unaprijedna faza učenja dinamičke mreže	12
3.3.1.1. Sakriveni sloj	13
3.3.1.2. Izlazni sloj	13
3.3.2. Povratna faza učenja dinamičke neuronske mreže	13
3.3.2.1. Promjena parametara učenja izlaznog sloja	14

3.3.2.2. Promjena parametara učenja sakrivenog sloja.....	14
4. Ocjena uspješnosti algoritma učenja.....	18
5. Predviđanje ponašanja nelinearnog kaotičnog sustava	19
5.1. Glass-mackey vremenska serija.....	19
5.2. Predviđanje Glass-Mackeyeve vremenske serije pomoću statičke i dinamičke neuronske mreže	20
6. ZAKLJUČAK.....	28

POPIS SLIKA

Slika 1. Model statičkog neurona	2
Slika 2. Sigmoidalna funkcija	3
Slika 3. Model statičke unparijedne neuronske mreže	4
Slika 4. Model dinamičkog neurona.....	9
Slika 5. Model dinamičke neuronske mreže	11
Slika 6. Shematski prikaz postupka učenja dinamičke neuronske mreže.....	12
Slika 7. Odziv SNM na test 1.....	21
Slika 8. Odziv SNM na test 2.....	22
Slika 9. Odziv SNM na test 3.....	23
Slika 10. Odziv DNM na test 1	24
Slika 11. Odziv DNM na test 2	25
Slika 12. Odziv DNM na test 3	26

POPIS TABLICA

Tablica 1. Usporedba rezultata testova nakon prvog učenja.....	26
Tablica 2. Usporedba rezultata testova nakon drugog učenja.....	27

POPIS OZNAKA

Oznaka	Opis
A_j	Nazivnik prijenosne funkcije dinamičkog člana dinamičkog neurona diskretne forme
a_{ij}	i-ti koeficijent nazivnika prijenosne funkcije dinamičkog člana j-og dinamičkog neurona
B_j	brojnik prijenosne funkcije dinamičkog člana dinamičkog neurona diskretne forme
b_{ij}	i-ti koeficijent brojnika prijenosne funkcije dinamičkog člana j-og dinamičkog neurona
BIAS	neuron bez ulaza s konstantnom izlaznom vrijednosti jednakom 1
DNM	dinamička neuronska mreža
d_k	željena vrijednost k-og izlaza neuronske mreže
E	funkcija cilja (suma kvadrata pogrešaka)
I	broj neurona ulaznog sloja mreže
J	broj neurona sakrivenog sloja mreže
K	broj neurona izlaznog sloja mreže
MS	srednja kvadratna pogreška
N	broj parova ulazno-izlaznih vrijednosti seta učenja
$NRMS$	normalizirani korijen srednje kvadratne pogreške
net	vrijednost funkcije sume
net_{Hj}	vrijednost funkcije sume j-og neurona sakrivenog sloja
net_{Ok}	vrijednost funkcije sume k-og neurona izlaznog sloja
RMS	korijen srednje kvadratne pogreške
SNM	statička neuronska mreža
O_k	k-ti izlaz neuronske mreže
u_j	vrijednost j-og ulaza neurona
v_{ji}	težinski koeficijent veze između j-og neurona sakrivenog sloja i i-tog neurona ulaznog sloja
w_j	težinski koeficijent j-og ulaza neurona

w_{kj}	težinski koeficijent veze između k-og neurona izlaznog sloja i j-tog neurona sakrivenog sloja
y	vrijednost izlaza neurona
y	izlaz dinamičkog člana dinamičkog neurona
$y(n-i)$	izlaz dinamičkog člana dinamičkog neurona u $(n-i)$ koraku
$\bar{y}$	ulaz aktivacijske funkcije dinamičkog neurona
y_j	vrijednost izlaza j-og neurona sakrivenog sloja
Z_i	i-ti ulaz neuronske mreže
$z^{-1}[net(n)]$	operator linearnog pomaka
ϑ	parametar učenja
$\vartheta(n)$	trenutna vrijednost parametra učenja
$\vartheta(n+1)$	nova vrijednost paramtera učenja
$\Delta \vartheta(n)$	tekuća promjena paramtera učenja
α	vrijednost momentuma
Δ	kašnjenje signala
Δ	iznos promjene težina u jednom koraku
δ	parametar algoritma povratnog prostiranja pogreške
δ_{ok}	parametar algoritma povratnog prostiranja pogreške izlaznog sloja
γ	aktivacijska funkcija neurona
γ'	derivacije aktivacijske funkcije neurona
η	koeficijent brzine učenja
τ	parametar pomaka Glass-Mackey kaotične jedandžbe
ΔE	promjena pogreške
∇E	gradijent pogreške

SAŽETAK

Neuronske mreže moćan su statistički alat, utoliko da ih je možda i pogrešno tako nazvati. Pokazuju svoj puni sjaj tamo gdje tradicionalne metode statistike nailaze na zapreke. Jedan takav primjer su kaotični sustavi. U ovom radu istražujemo koliko dobro neuronske mreže mogu predvidjeti Glass-Mackey kaotični sustav koristeći dva modela neuronskih mreža – statičku i dinamičku neuronsku mrežu, te uspoređujemo njihove rezultate.

Ključne riječi: Neuronske mreže, statička neuronska mreža, dinamička neuronska mreža, kaotični sustavi, Glass-Mackey jednadžba

SUMMARY

Neural networks are a powerful statistical tool, so much that it might be wrong to call them a statistical tool. They shine where traditional statistics hit a brick wall. One of such cases are chaotic systems. In this paper we investigate how well can neural networks predict the Glass-Mackey equation using two different models of neural networks – a static neural network and a dynamic neural network, and we compare their results.

Key words: Neural networks, dynamic neural network, static neural networks, chaotic systems, Glass-Mackey equation.

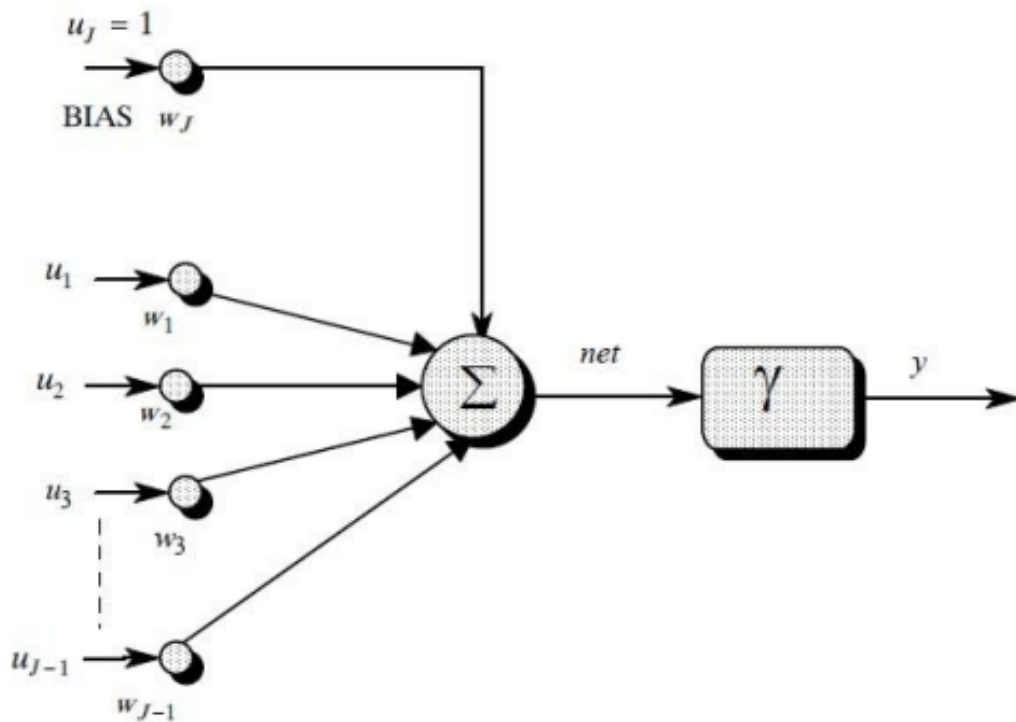
1. UVOD

Ljudi su oduvijek pokušavali oponašati stvari koje vide u prirodi. Tu je neizbježno spomenuti kako je Leonardo Da Vinci istraživao ptice te pomoću tih saznanja pokušao konstruirati avion. Moderniji primjer bile bi razne hvataljke robotskih ruku koje mogu biti nalik kandžama, zatim sferni zglobovi koji su nalik zglobovima sisavaca.... Moderna znanost omogućila nam je da istražimo i stvari koje nisu vidljive golim okom. U ovom slučaju to je način funkcioniranja ljudskog mozga. U mozgu postoji ogroman broj međusobno povezanih neurona na kojima se zasniva ideja umjetnih neuronskih mreža.

2. Statička neuronska mreža

2.1. Model statičkog neurona

Model statičkog neurona dan je slikom 1.



Slika 1. Model statičkog neurona

Statički neuron posjeduje samo jedan izlaz, dok broj ulaza ovisi o broju neurona prethodnog sloja. Osim njih, kao ulazna varijabla pojavljuje se bias neuron, čija je vrijednost najčešće postavljena na jedinicu. Međuvrijednost sakrivenog neurona računa se sumom umnožaka ulaznih vrijednosti i težinskih faktora prema izrazu:

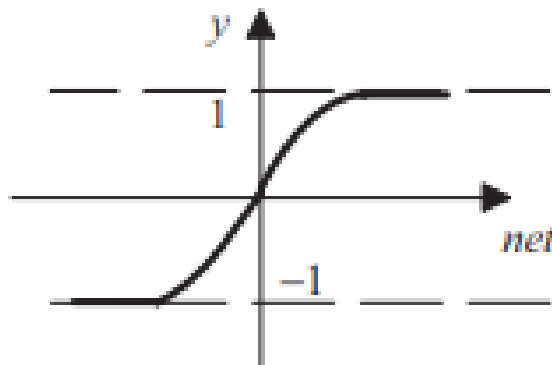
$$net = \sum_{j=1}^J w_j u_j \quad , \quad (2.1)$$

gdje je u_j vrijednost j-og ulaza neurona a w_j težinski koeficijent j-og ulaza neurona.

Ta se suma provuče kroz aktivacijsku funkciju – najčešće neka monotono rastuća funkcija sa zasićenjem. Ovdje je korištena sigmoidalna funkcija sa vrijednostima između -1 i 1 čija je funkcija dana izrazom:

$$y = \frac{2}{1 + e^{-net}} - 1, \quad (2.2)$$

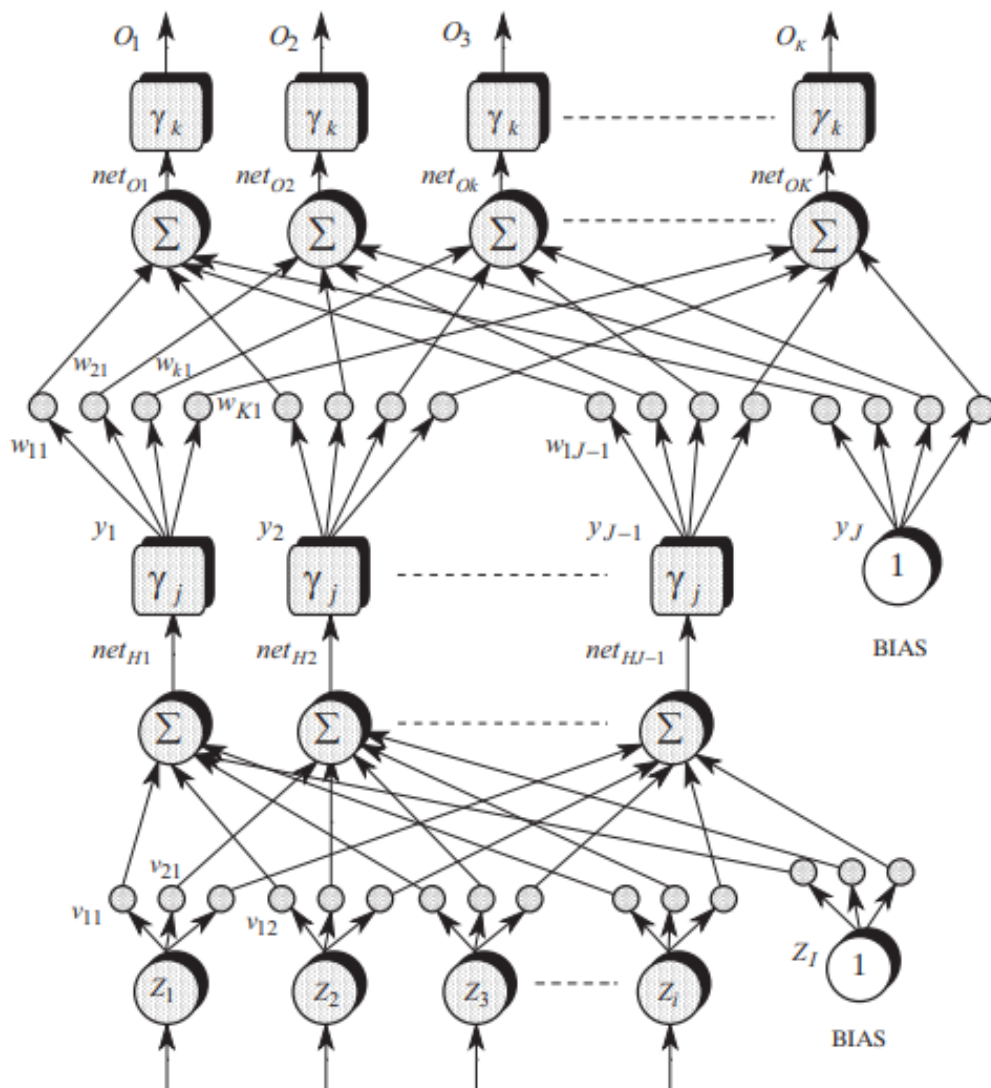
gdje je *net* vrijednost funkcije sume.



Slika 2. Sigmoidalna funkcija

2.2. Model statičke neuronske mreže

Da bi mreža mogla funkcionirati, neurone treba postaviti u strukturu koja se sastoji od više slojeva. Prvi sloj naziva se ulazni te vrijednosti neurona u tom sloja definira korisnik. Svi slojevi između prvog i posljednjeg nazivaju se sakriveni slojevi. Njih može biti više, ali može postojati i samo jedan, što je u velikom broju slučajeva dovoljno. Posljedni sloj naziva se izlazni sloj.



Slika 3. Model statičke uniparijedne neuronske mreže

2.3. Unaprijedna faza učenja statičke mreže

2.3.1. Sakriveni sloj

U unaprijednoj fazi učenja, neuronska mreža na osnovu ulaznih podataka i pomoću težinskih faktora računa izlaznu vrijednost mreže, koja će u prvom koraku vrlo vjerojatno u velikoj mjeri odstupati od željene. Težinski faktori slučajno su generirani brojevi između -1 i 1. Međuvrijednosti net neurona u sakrivenom sloju računaju se prema izrazu:

$$net_{H_j} = \sum_{i=1}^I v_{ji} Z_i, \quad j=1,2,\dots,J-1, \quad i=1,2,\dots,I, \quad (2.3)$$

gdje I predstavlja broj ulaznih neurona +1, a J broj neurona u sakrivenom sloju uvećan za jedan (Bias). Sa v_{ji} je označen težinski koeficijent veze između j -og neurona sakrivenog sloja i i -tog neurona ulaznog sloja. Z_i označava i -ti ulaz neuronske mreže. Međuvrijednosti net se provlače kroz aktivacijsku funkciju koja je u ovom radu sigmoidalna funkcija, te je izlaz neurona sakrivenog sloja jednak:

$$y_j = \frac{2}{1 + e^{-net_{H_j}}} - 1, \quad j=1,2,\dots,J-1, \quad (2.4)$$

$$y_J = 1, \text{ Bias.} \quad (2.5)$$

2.3.2. Izlazni sloj

Izlazne vrijednosti prethodnog (sakrivenog) sloja predstavljaju ulaze u neurone izlaznog sloja. Suma net izlaznog neurona dobiva se na analogan način:

$$net_{Ok} = \sum_{j=1}^J w_{kj} y_j, \quad k=1,2,\dots,K, \quad (2.6)$$

gdje je K broj neurona izlaznog sloja. Težinski koeficijent veze između k -og neurona izlaznog sloja i j -tog neurona sakrivenog sloja označen je s w_{kj} . Kao aktivacijsku funkciju može se uzeti linearna funkcija kako bismo mogli dobiti izlaze veće od 1, tako da je konačna vrijednost izlaza mreže:

$$O_k = K_p net_{Ok}, \quad k=1,2,\dots,K, \quad (2.7)$$

gdje je K_p nagib linearne aktivacijske funkcije koji je u ovom radu jednak jedan.

2.4. Povratna faza učenja statičke mreže

U povratnoj se fazi na osnovu razlike između izlazne vrijednosti koju je izračunala mreža i vrijednosti koja je zadana kao ispravna ažuriraju težinski faktori. Za funkciju cilja mogu se odabrati razne funkcije i time se usmjeruje rad mreže. U ovom radu odabrana je najčešće korištena funkcija cilja:

$$E = \frac{1}{2} \sum_{n=1}^N (d_n - O_n)^2 \quad . \quad (2.8)$$

N je broj elemenata u skupu za učenje, d_n je željena vrijednost n -tog izlaza mreže a O_n je n -ti izlaz mreže. Postupak učenja svodi se na minimizaciju funkcije cilja. Promjena težinskih koeficijenata vrši se prema izrazu:

$$\mathcal{G}(n+1) = \mathcal{G}(n) + \Delta \mathcal{G}(n) \quad . \quad (2.9)$$

Trenutni korak učenja označen je s n dok $n+1$ predstavlja vrijednosti za sljedeći korak. Gradijent pogreške može se izračunati pomoću Taylorovog reda. Pomoću njega dolazimo do izraza za veličinu promjene parametara učenja koji osigurava da se greška smanjuje najvećim mogućim iznosom:

$$\Delta \mathcal{G}(n) = -\eta \nabla E(\mathcal{G}) \quad , \quad (2.10)$$

gdje je η proizvoljno odabrana veličina koja se naziva koeficijent brzine učenja, te je njena uobičajena vrijednosti između 0.0001 i 10. Uvrštavanjem izraza (2.10) u (2.9) dobivamo izraz za promjenu parametara mreže:

$$\mathcal{G}(n+1) = \mathcal{G}(n) - \eta \nabla E(\mathcal{G}(n)) \quad . \quad (2.11)$$

Ovaj izraz karakterizira *algoritam povratnog prostiranja pogreške* koji je najčešće korišteni algoritam učenja neuronskih mreža. Njegov najveći nedostatak je potreban broj iteracija odnosno koraka učenja, koji se može smanjiti primjenom momentuma, koji promjenu parametara dodatno povećava ili smanjuje u ovisnosti o iznosu njihove promjene iz prethodnog koraka. Ako koristimo momentum tada je izraz za izračun novih parametara jednak:

$$\mathcal{G}(n+1) = \mathcal{G}(n) - \eta \nabla E(\mathcal{G}(n)) + \alpha \Delta \mathcal{G}(n-1) \quad . \quad (2.12)$$

2.4.1. Promjena težina izlaznog sloja

Promjena težina odvija se od izlaznog sloja prema ulaznom sloju. Nove težine izlaznog sloja računaju se prema izrazu:

$$\mathbf{w}_{kj}(n+1) = \mathbf{w}_{kj}(n) - \eta \nabla E(n) + \alpha \Delta \mathbf{w}_{kj}(n-1) \quad . \quad (2.13)$$

Vidimo da je osnova čitavog postupka pronalazak gradijenta pogreške, koji se lako nalazi lančanom derivacijom:

$$\frac{\partial E(n)}{\partial w_{kj}} = \frac{\partial E(n)}{\partial O_k} \frac{\partial O_k}{\partial net_{Ok}} \frac{\partial net_{Ok}}{\partial w_{kj}} . \quad (2.14)$$

Članovi desne strane jednadžbe su redom:

$$\frac{\partial E(n)}{\partial O_k} = -(d_k - O_k) , \quad (2.15)$$

$$\frac{\partial O_k}{\partial net_{Ok}} = \gamma'_k = 1 , \quad (2.16)$$

$$\frac{\partial net_{Ok}}{\partial w_{kj}} = y_j , \quad (2.17)$$

što daje konačni izraz za ažuriranje težina vanjskog sloja:

$$w_{kj}(n+1) = w_{kj}(n) + \eta \delta_{Ok} y_j + \alpha \Delta w_{kj}(n-1) , \quad (2.18)$$

gdje je:

$$\delta_{Ok} = d_k - O_k . \quad (2.19)$$

2.4.2. Promjena težina sakrivenog sloja

Promjena težina sakrivenog sloja vrši se po istom principu kao i promjena težina vanjskog sloja. Temeljna jednadžba glasi:

$$v_{ji}(n+1) = v_{ji}(n) - \eta \nabla E(n) + \alpha \Delta v_{ji}(n-1) . \quad (2.20)$$

Gradijent greške možemo raspisati kao umnožak parcijalnih derivacija:

$$\frac{\partial E(n)}{\partial v_{ji}} = \frac{\partial E(n)}{\partial y_j} \frac{\partial y_j}{\partial net_{Hj}} \frac{\partial net_{Hj}}{\partial v_{ji}} . \quad (2.21)$$

Na promjenu svake težine sakrivenog sloja utječu svi neuroni izlaznog sloja, pa prvi izraz jednadžbe (2.21) ima oblik:

$$\begin{aligned}
\frac{\partial E(n)}{\partial y_j} &= \frac{\partial E(n)}{\partial O_1} \frac{\partial O_1}{\partial net_{O1}} \frac{\partial net_{O1}}{\partial y_j} + \\
&+ \frac{\partial E(n)}{\partial O_2} \frac{\partial O_2}{\partial net_{O2}} \frac{\partial net_{O2}}{\partial y_j} + \\
&+ \dots + \\
&+ \frac{\partial E(n)}{\partial O_K} \frac{\partial O_K}{\partial net_{OK}} \frac{\partial net_{OK}}{\partial y_j} .
\end{aligned} \tag{2.22}$$

Pri tome je:

$$\frac{\partial E(n)}{\partial O_k} = -(d_k - O_k) \quad , \quad k=1,2,\dots,K, \tag{2.23}$$

$$\frac{\partial O_k}{\partial net_{Ok}} = 1 \quad , \quad k=1,2,\dots,K, \tag{2.24}$$

$$\frac{\partial net_{Ok}}{\partial y_j} = w_{kj} \quad , \quad k=1,2,\dots,K, \quad j=1,2,\dots,J-1 . \tag{2.25}$$

Sada su članovi desne strane jednadžbe (2.21) jednaki:

$$\frac{\partial E(n)}{\partial y_j} = -\sum_{k=1}^K (d_k - O_k) w_{kj} \quad , \tag{2.26}$$

$$\frac{\partial y_j}{\partial net_{Hj}} = \gamma'_{ij} = \frac{1}{2} (1 - y_j^2) \quad , \tag{2.27}$$

$$\frac{\partial net_{Hj}}{\partial v_{ji}} = Z_i . \tag{2.28}$$

U izraz (2.22) možemo uvrstiti izraz (2.19) čime dobivamo:

$$\frac{\partial E(n)}{\partial y_j} = -\sum_{k=1}^K \delta_{Ok} w_{kj} \quad , \tag{2.29}$$

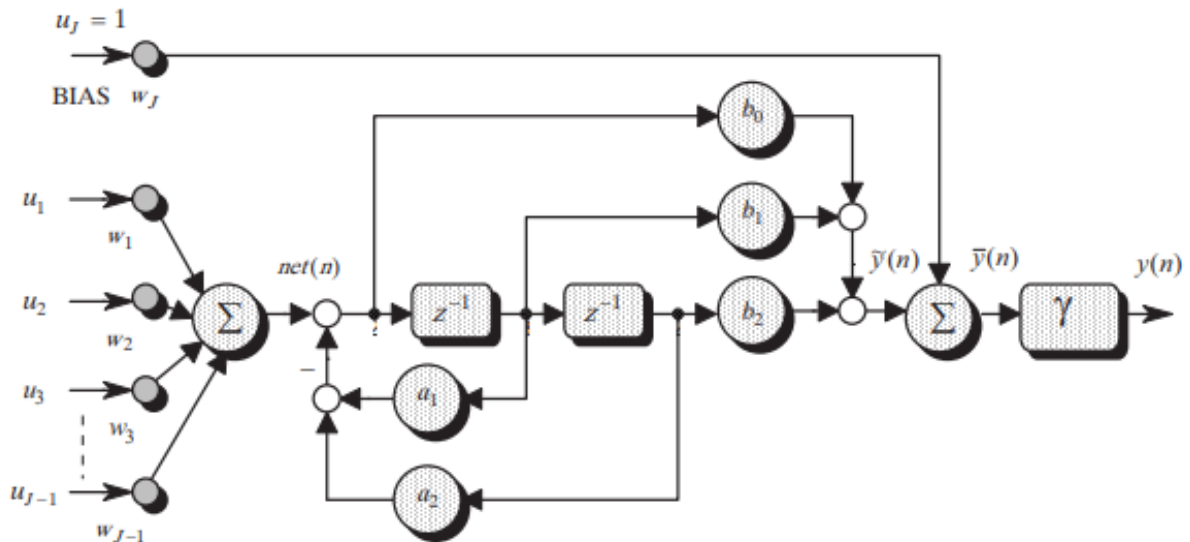
pa konačni izraz za promjenu težina sakrivenog sloja glasi:

$$v_{ji}(n+1) = v_{ji}(n) + \frac{1}{2} \eta (1 - y_j^2) Z_i \left(\sum_{k=1}^K \delta_{Ok} w_{kj} \right) + \alpha \Delta v_{ji}(n-1) . \tag{2.30}$$

3. Dinamička neuronska mreža

3.1. Model dinamičkog neurona

U modelu dinamičkog neurona pojavljuje se povratna veza tako da aktivnost neurona bude u funkciji stanja neurona. Ugradnja poznatog modela ARMA-filtera (engl. Auto Regressive Moving Average) ispred aktivacijske funkcije neurona omogućuje djelovanje dinamičkog neurona na osnovu ulaznih signala i prethodno dobivenih vrijednosti izlazne veličine dinamičkog člana dinamičkog neurona.



Slika 4. Model dinamičkog neurona

Funkcija sume dinamičkog neurona gotovo je jednaka funkciji sume statičkog neurona, jedino što ovdje ne uračunavamo bias neuron u sumu, već se on dodaje kasnije kao što je vidljivo na slici 4.

$$net(n) = \sum_{j=1}^{J-1} w_j u_j \quad . \quad (3.1)$$

Izlaz dinamičkog člana dobiva se korištenjem vremenskih pomaka signala ulaza i izlaza:

$$y(n) = b_0 net(n) + b_1 net(n-1) + b_2 net(n-2) - a_1 y(n-1) - a_2 y(n-2) \quad , \quad (3.2)$$

gdje su b_i i a_i parametri dinamičkog neurona.

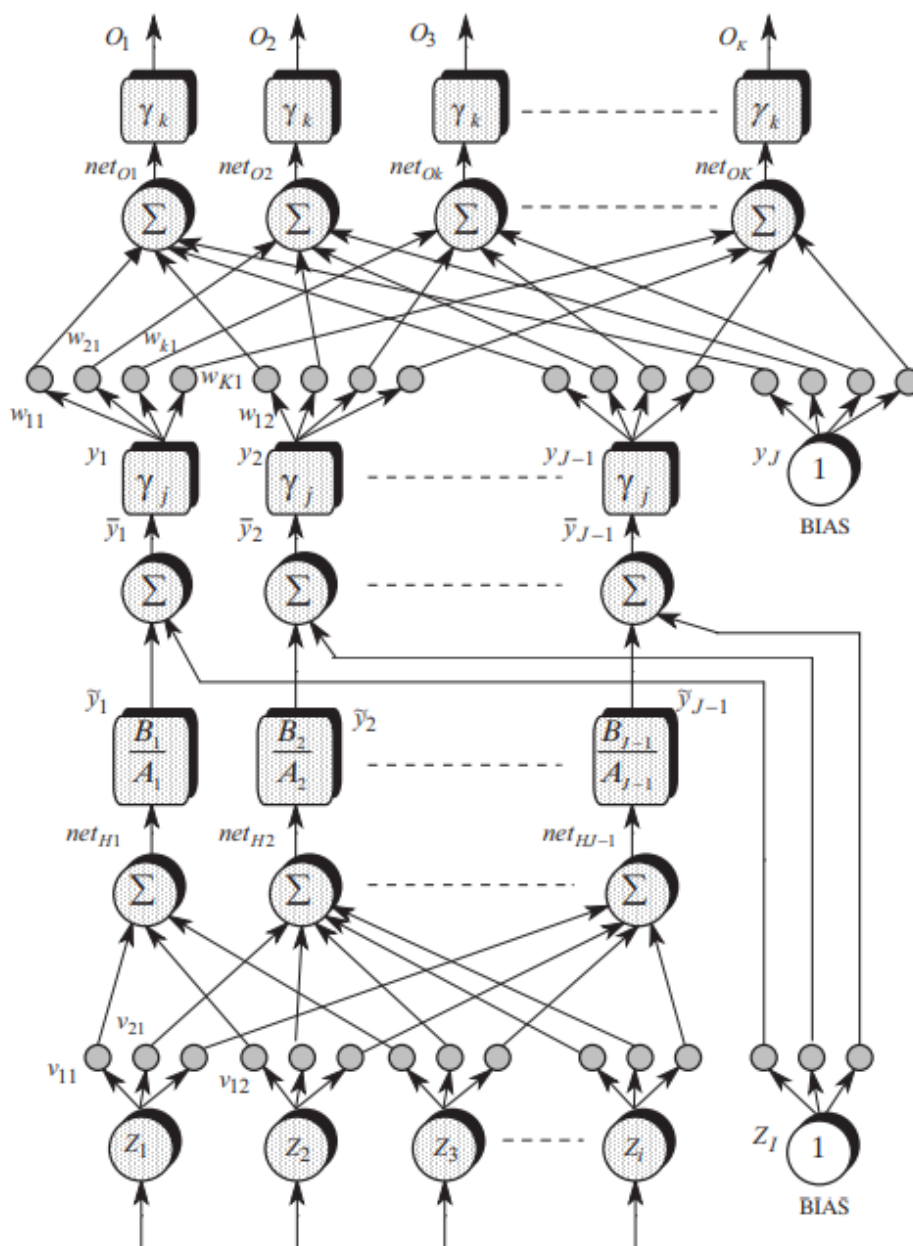
Toj vrijednosti još trebamo dodati bias neuron, te rezultirajuću vrijednost provući kroz sigmoidalnu aktivacijsku funkciju:

$$\bar{y}(n) = y(n) + w_J, \quad (3.3)$$

$$y(n) = \gamma(\bar{y}(n)) = \frac{2}{1 + e^{-\bar{y}(n)}} - 1. \quad (3.4)$$

3.2. Model dinamičke neuronske mreže

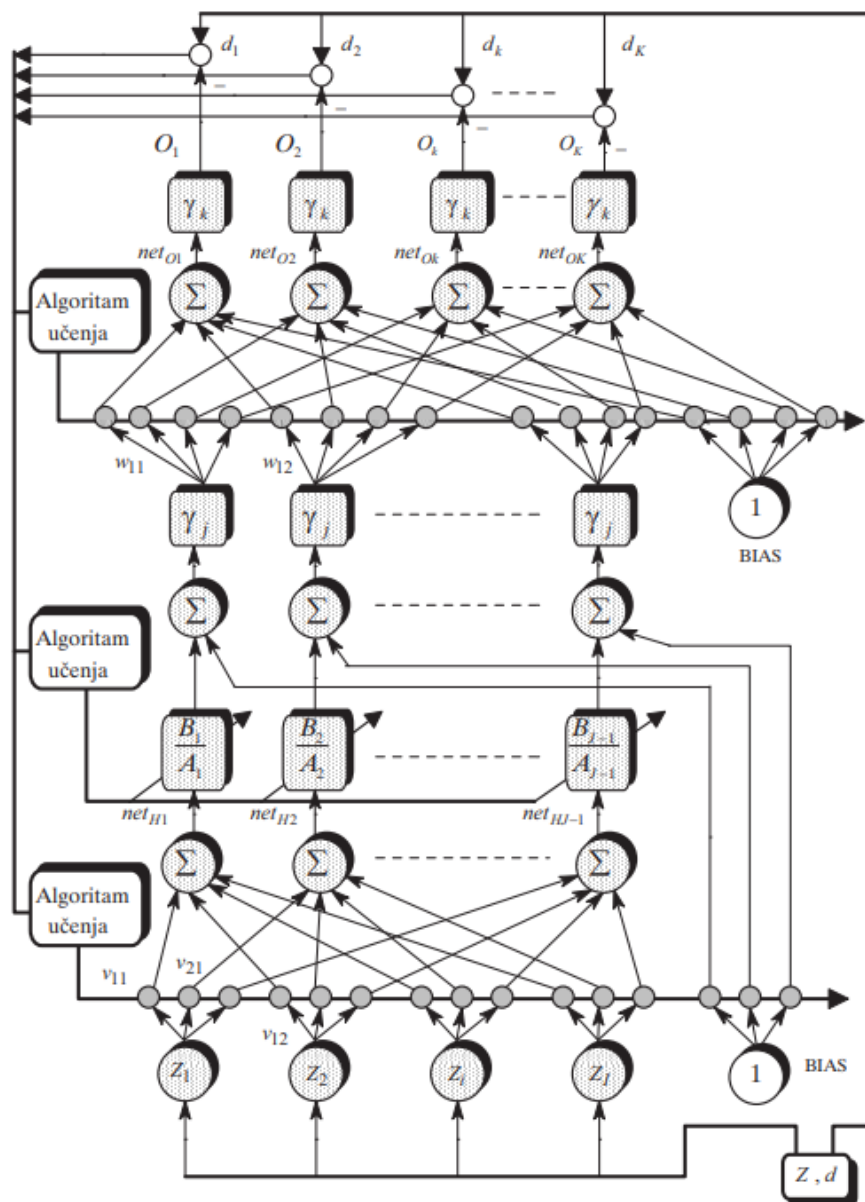
Model same neuronske mreže jednak je modelu statičke neuronske mreže, jedino što ovdje koristimo dinamičke neurone.



Slika 5. Model dinamičke neuronske mreže

3.3. Postupak učenja dinamičke mreže

Ostvaren je klasični algoritam učenja povratnim prostiranjem pogreške baš kao i kod statičke mreže. Uz težinske faktore, ova mreže uči i parametre a i b koji se odnose na dinamičke komponente neurona. Učenje je shematski prikazano na slici 6.



Slika 6. Shematski prikaz postupka učenja dinamičke neuronske mreže

3.3.1. Unaprijedna faza učenja dinamičke mreže

U unaprijednoj se fazi na osnovi početnih vrijednosti parametara učenja računa izlaz mreže. Težinski koeficijenti dobiveni su generatorom slučajnih brojeva, dok su koeficijenti filtra postavljeni na nulu, osim koeficijenta b_{0j} koji je postavljen na jedinicu. Početne vrijednosti veličina koje se odnose na prethodne korake postavljaju se na nul-vrijednosti.

3.3.1.1. Sakriveni sloj

Funkcija sume glasi:

$$net_{Hj} = \sum_{i=1}^{I-1} v_{ji} Z_i \quad , j = 1, 2, \dots, J-1, \quad i = 1, 2, \dots, I-1 \quad . \quad (3.5)$$

Izlazi dinamičkih članova računaju se prema:

$$y(n) = b_{0j}(n)net_{Hj}(n) + b_{1j}(n)net_{Hj}(n-1) + b_{2j}(n)net_{Hj}(n-2) - \\ - a_{1j}y_j(n-1) - a_{2j}(n)y(n-2) \quad . \quad (3.6)$$

Dalje imamo funkciju koja je ulazna veličina aktivacijskoj funkciji:

$$\bar{y}_j(n) = y_j(n) + v_{jI}(n)Z_I \quad . \quad (3.7)$$

Uz $Z_I = 1$ (Bias), gornja jednadžba poprima oblik:

$$\bar{y}_j(n) = y_j(n) + v_{jI}(n) \quad . \quad (3.8)$$

Na kraju dobivamo izlaz dinamičkog neurona:

$$y_j(n) = \frac{2}{1 + e^{-\bar{y}_j(n)}} - 1 \quad , j = 1, 2, \dots, J-1 \quad , \quad (3.9)$$

$$y_J = 1 \quad \text{Bias.} \quad (3.10)$$

3.3.1.2. Izlazni sloj

Funkcija sume glasi:

$$net_{Ok}(n) = \sum_{j=1}^J y_j(n)w_{kj}(n) \quad . \quad (3.11)$$

Primjenom linearne aktivacijske funkcije dobivaju se vrijednosti izlaza neuronske mreže:

$$O_k(n) = net_{Ok}(n) \quad , k = 1, 2, \dots, K \quad , \quad (3.12)$$

gdje K označava broj izlaza mreže.

3.3.2. Povratna faza učenja dinamičke neuronske mreže

Funkcija cilja identična je onoj kod statičke mreže:

$$E = \frac{1}{2} \sum_{n=1}^N (d_n - O_n)^2 \quad . \quad (3.13)$$

Promjena parametara mreže (težinski koeficijenti i koeficijenti dinamičkih članova neurona sakrivenog sloja) odvija se po već poznatom algoritmu:

$$\mathcal{G}(n+1) = \mathcal{G}(n) - \eta \nabla E(\mathcal{G}(n)) + \alpha \Delta \mathcal{G}(n-1) \quad . \quad (3.14)$$

Promjena parametara odvija se od izlaznog prema ulaznom sloju.

3.3.2.1. Promjena parametara učenja izlaznog sloja

Promjena parametara učenja izlaznog sloja odvija se prema sljedećem izrazu:

$$w_{kj}(n+1) = w_{kj}(n) - \eta \nabla E(n) + \alpha \Delta w_{kj}(n-1) \quad . \quad (3.15)$$

Gradijent pogreške izračunava se prema prije pokazanim izrazima (2.14) do (2.17). Konačna forma promjene težina w_{kj} dana je izrazom (2.18).

3.3.2.2. Promjena parametara učenja sakrivenog sloja

Nakon promjene težina izlaznog sloja slijedi promjena koeficijenata dinamičkih članova sakrivenog sloja. Zbog pojednostavljenja potrebnih derivacija definiramo linearni operator vremenskog pomaka:

$$\left[y(n) \right] = \frac{B(z)}{A(z)} [net(n)], \quad z^{-i} [net(n)] = net(n-i) \quad , \quad (3.16)$$

gdje su:

$$A(z) \left[y(z) \right] = y(n) + a_1 y(n-1) + a_2 y(n-2) \quad , \quad (3.17)$$

$$B(z) [net(n)] = b_0 net(n) + b_1 net(n-1) + b_2 net(n-2) \quad . \quad (3.18)$$

Formula za promjenu parametara izražena pomoću dinamičkog parametra b_{0j} glasi:

$$b_{0j}(n+1) = b_{0j}(n) - \eta \nabla E(b_{0j}(n)) + \alpha \Delta b_{0j}(n-1) \quad . \quad (3.19)$$

Gradijent možemo zapisati pomoću parcijalnih derivacija:

$$\nabla E(n) = \frac{\partial E(n)}{\partial b_{0j}} = \frac{\partial E}{\partial y_j} \frac{\partial y_j}{\partial \bar{y}_j} \frac{\partial \bar{y}_j}{\partial y_j} \frac{\partial y_j}{\partial b_{0j}} \quad . \quad (3.20)$$

Prvi član desne strane jednadžbe već smo izračunali u prethodnom poglavlju, jednadžba (2.29). Drugi član računa se na slijedeći način:

$$\frac{\partial \bar{y}_j}{\partial y_j} = \gamma'_{ij} = \frac{1}{2}(1 - y_j^2) \quad . \quad (3.21)$$

S obzirom da smo bias postavili na jedinicu, vrijednost treće derivacije također je jedinica:

$$\frac{\partial \bar{y}_j}{\partial y_j} = 1 \quad . \quad (3.22)$$

Primjenom operatora vremenskog pomaka izvodi se:

$$\left. \frac{\partial [y(n)]}{\partial \theta} \right|_{\theta=b_{0j}} = [D_{b_{0j}}(n)] = \frac{z^{-i}}{A(z)} [net(n)], \quad i = 0 \quad . \quad (3.23)$$

Imamo sve što je potrebno za zapisati konačni izraz za promjenu paramtera b_0 :

$$b_{0j}(n+1) = b_{0j}(n) + \frac{1}{2} \eta (1 - y_j^2) D_{b_{0j}}(n) \left(\sum_{k=1}^K \delta_{Ok} w_{kj} \right) + \alpha \Delta b_{0j}(n-1) \quad . \quad (3.24)$$

Promjena parametara b_1 i b_2 odvija se na jednak način, pa možemo zapisati opći oblik formule za promjenu parametara b :

$$b_{ij}(n+1) = b_{ij}(n) + \frac{1}{2} \eta (1 - y_j^2) D_{b_{ij}}(n) \left(\sum_{k=1}^K \delta_{Ok} w_{kj} \right) + \alpha \Delta b_{ij}(n-1), \quad i = 0, 1, 2 \quad , \quad (3.25)$$

$$[D_{b_{ij}}(n)] = \frac{z^{-i}}{A(z)} [net(n)], \quad i = 0, 1, 2 \quad . \quad (3.26)$$

Promjena parametra a_{1j} odvija se prema:

$$a_{1j}(n+1) = a_{1j}(n) - \eta \nabla E(a_{1j}(n)) + \alpha \Delta a_{1j}(n-1) \quad , \quad (3.27)$$

$$\nabla E(n) = \frac{\partial E(n)}{\partial a_{1j}} = \frac{\partial E}{\partial y_j} \frac{\partial y_j}{\partial \bar{y}_j} \frac{\partial \bar{y}_j}{\partial y_j} \frac{\partial y_j}{\partial a_{1j}} \quad . \quad (3.28)$$

Prva tri člana desne strane jednadžbe već smo izračunali kod algoritma za promjenu parametra b te su dani jednadžbama (2.29), (3.21) i (3.22). Posljednji razlomak primjenom vremenskog operatora ima slijedeći oblik:

$$\left. \frac{\partial [y(n)]}{\partial \vartheta} \right|_{\vartheta=a_{1j}} = [D_{a_{1j}}(n)] = \frac{-z^{-i}}{A(z)} [y(n)], \quad i=1, \quad (3.29)$$

a konačni izraz za promjenu parametra a_1 glasi:

$$a_{1j}(n+1) = a_{1j}(n) + \frac{1}{2} \eta (1 - y_j^2) D_{a_{1j}}(n) \left(\sum_{k=1}^K \delta_{Ok} \mathbf{w}_{kj} \right) + \alpha \Delta a_{1j}(n-1) \quad (3.30)$$

Kao i kod parametra b , možemo dati opću formu promjene parametra a :

$$a_{ij}(n+1) = a_{ij}(n) + \frac{1}{2} \eta (1 - y_j^2) D_{a_{ij}}(n) \left(\sum_{k=1}^K \delta_{Ok} \mathbf{w}_{kj} \right) + \alpha \Delta a_{ij}(n-1), \quad i=1, 2, \quad (3.31)$$

gdje je:

$$[D_{a_{ij}}(n)] = \frac{-z^{-i}}{A(z)} [y(n)], \quad i=1, 2 \quad (3.32)$$

Na redu su promjene težinskih koeficijenata koji povezuju ulazni i sakriveni sloj neurona.

Formula glasi:

$$\mathbf{v}_{ji}(n+1) = \mathbf{v}_{ji}(n) - \eta \nabla E(n) + \alpha \Delta \mathbf{v}_{ji}(n-1) \quad (3.33)$$

$$\nabla E(n) = \frac{\partial E(n)}{\partial \mathbf{v}_{ji}} = \frac{\partial E}{\partial y_j} \frac{\partial y_j}{\partial \bar{y}_j} \frac{\partial \bar{y}_j}{\partial y_j} \frac{\partial y_j}{\partial \mathbf{v}_{ji}} \quad (3.34)$$

Prva tri razlomka već imamo. Posljednji razlomak dobiva se pomoću operatora vremenskog pomaka:

$$\left. \frac{\partial [y(n)]}{\partial \vartheta} \right|_{\vartheta=\mathbf{v}_{ji}} = [D_{\mathbf{v}_{ji}}(n)] = \frac{B(z)}{A(z)} [Z_i(n)] \quad (3.35)$$

a konačni izraz za promjenu težinskih koeficijenata glasi:

$$\mathbf{v}_{ji}(n+1) = \mathbf{v}_{ji}(n) + \frac{1}{2}\eta(1-y_j^2)D_{\mathbf{v}_{ji}}(n)\left(\sum_{k=1}^K\delta_{Ok}\mathbf{w}_{kj}\right) + \alpha\Delta\mathbf{v}_{ji}(n-1) \quad , \quad (3.36)$$

gdje je:

$$\left[D_{\mathbf{v}_{ji}}(n)\right] = \frac{B(z)}{A(z)}[Z_i(n)] \quad . \quad (3.37)$$

Ne smijemo zaboraviti na bias neuron i njemu pridružen težinski koeficijent:

$$\mathbf{v}_{jI}(n+1) = \mathbf{v}_{jI}(n) - \eta\nabla E(n) + \alpha\Delta\mathbf{v}_{jI}(n-1) \quad , \quad (3.38)$$

$$\nabla E(n) = \frac{\partial E(n)}{\partial \mathbf{v}_{jI}} = \frac{\partial E}{\partial y_j} \frac{\partial y_j}{\partial \bar{y}_j} \frac{\partial \bar{y}_j}{\partial \mathbf{v}_{jI}} \quad . \quad (3.39)$$

Prva dva razlomka već su dana dok posljednji glasi:

$$\frac{\partial \bar{y}_j}{\partial \mathbf{v}_{jI}} = Z_I = 1 \quad , \quad (3.40)$$

pa imamo konačni izraz za promjenu težine bias neurona:

$$\mathbf{v}_{jI}(n+1) = \mathbf{v}_{jI}(n) + \frac{1}{2}\eta(1-y_j^2)\left(\sum_{k=1}^K\delta_{Ok}\mathbf{w}_{kj}\right) + \alpha\Delta\mathbf{v}_{jI}(n-1) \quad . \quad (3.41)$$

4. Ocjena uspješnosti algoritma učenja

Kako bismo mogli ocjeniti uspješnost algoritma te ga objektivno usporediti s drugim algoritmima, potrebno je imati valjan način za ocjenu točnosti. Danas su u uporabi najčešće slijedeće tri mjere točnosti algoritma učenja:

1. Srednja kvadratna pogreška, MS (engl. Mean square error):

$$MS = \frac{\sum_{n=1}^N (d_n - O_n)^2}{N}, \quad (4.1)$$

2. Korijen srednje kvadratne pogreške, RMS (engl. Root mean square error):

$$RMS = \sqrt{MS} = \sqrt{\frac{\sum_{n=1}^N (d_n - O_n)^2}{N}}, \quad (4.2)$$

3. Normalizirani korijen srednje kvadratne pogreške, $NRMS$ (engl. Normalized root mean square error):

$$NRMS = \frac{RMS}{\sigma_{d_n}} = \frac{\sqrt{\frac{\sum_{n=1}^N (d_n - O_n)^2}{N}}}{\sigma_{d_n}}, \quad (4.3)$$

gdje su:

$$\sigma_{d_n} = \sqrt{\frac{1}{N} \sum_{n=1}^N (d_n - \bar{d})^2}, \quad (4.4)$$

$$\bar{d} = \frac{1}{N} \sum_{n=1}^N d_n. \quad (4.5)$$

U ovom će se radu koristiti posljednje mjera točnosti ($NRMS$) pri usporedbi uspješnosti mreža.

5. Predviđanje ponašanja nelinearnog kaotičnog sustava

Neke od tipičnih primjena neuronskih mreža uključuju obradu signala i previđanje ponašanja dinamičkih sustava, primjerice predviđanje vremena ili cijena pojedinih proizvoda na tržištu. Tradicionalne statističke metode daju slabe rezultate u predikciji dinamičkih sustava, posebno onih s nelinearnostima, vremenski promjenjivim parametrima i značajkama kaosa. U ovom radu odabrano je predviđanje tipičnog nelinearnog kaotičnog sustava za šest vremenskih koraka unaprijed, u literaturi poznatom pod nazivom Glass-Mackeyeva vremenska serija.

5.1. Glass-mackey vremenska serija

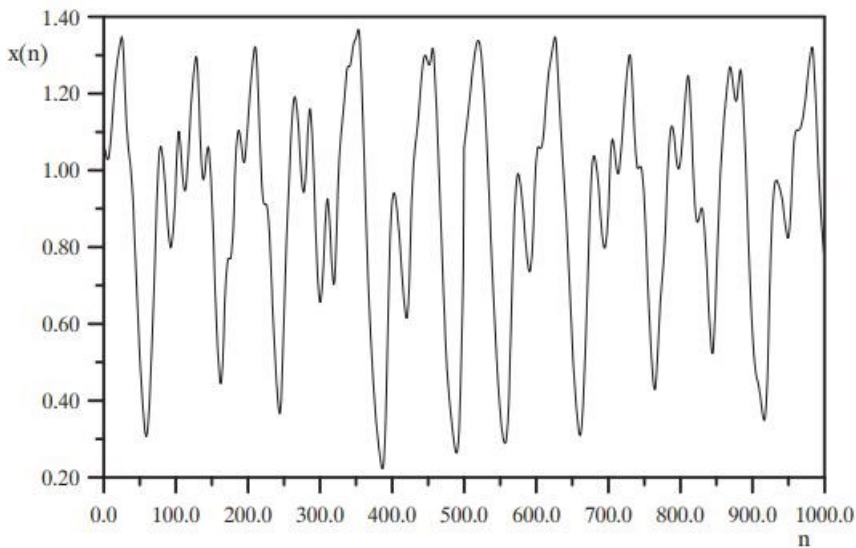
Lapedes i Farber predlažu Glass-Mackeyev nelinearni kaotični sustav za testiranje algoritama učenja neuronskih mreža jer ima jednostavnu definiciju, a njegovo ponašanje teško se predviđa. Matematički izraz jednadžbe glasi:

$$x' = \frac{ax(t-\tau)}{1+x^{10}(t-\tau)} - bx(t) \quad , \quad (5.1)$$

gdje su a i b parametri sustava, a t oznaka vremena. Diskretiziramo li jednadžbu dobiti ćemo slijedeći izraz:

$$x(n) = \frac{1}{1+b} \left[x(n-1) + \frac{ax(n-\tau)}{1+x^{10}(n-\tau)} \right] \quad , \quad (5.2)$$

a grafički prikaz iste sa 1000 točaka izgleda ovako:



5.2. Predviđanje Glass-Mackeyeve vremenske serije pomoću statičke i dinamičke neuronske mreže

Osnovni zadatak predviđanja ponašanja nelinearnog kaotičkog sustava svodi se na uporabu prošlih vrijednosti vremenskih serija signala $x(n)$ do neke vremenske točke, za predikciju buduće vrijednosti $x(n+P)$, za P točaka u budućnosti:

$$x(n+P) = f(x(n), x(n-\Delta), x(n-2\Delta), \dots, x(n-m\Delta)) \quad . \quad (5.3)$$

U ovom je radu odabrano $P = \Delta = 6$ i $m=4$. Sada gornja jednadžba ima slijedeću formu:

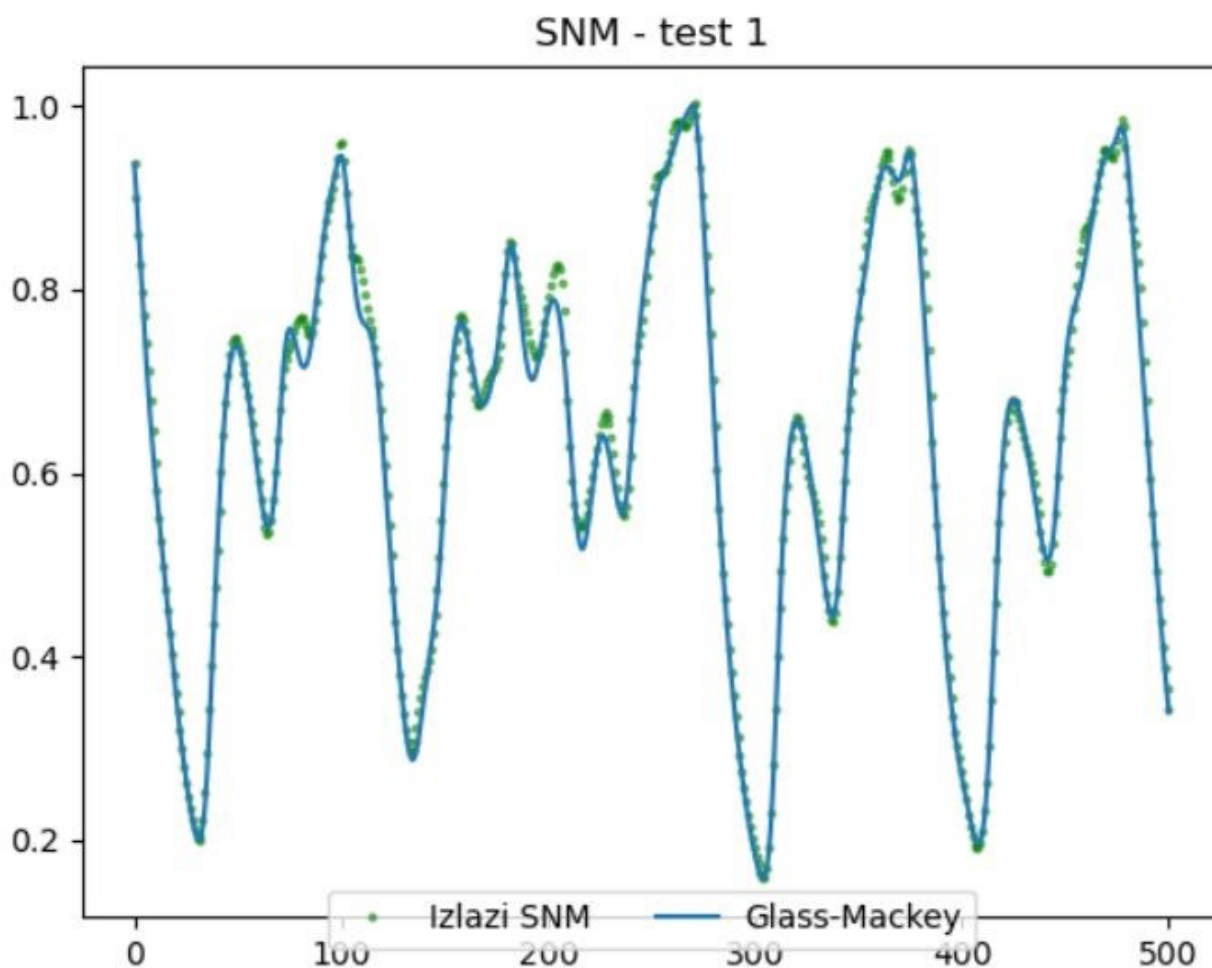
$$x(n+6) = f(x(n), x(n-6), x(n-12), x(n-18), x(n-24)) \quad . \quad (5.4)$$

Drugim riječima, zadatak neuronske mreže je da na osnovu mjerenih veličina izlaza sustava $x(n)$, $x(n-6)$, $x(n-12)$, $x(n-18)$, $x(n-24)$ predvidi vrijednost izlaza sustava 6 koraka unaprijed ($x(n+6)$). Imamo dakle 5 ulaznih neurona (plus bias) i 1 izlazni neuron. Koristiti ćemo samo jedan sakriveni sloj koji će u prvom učenju imati 8 neurona (plus bias), a u drugom 10 neurona (plus bias).

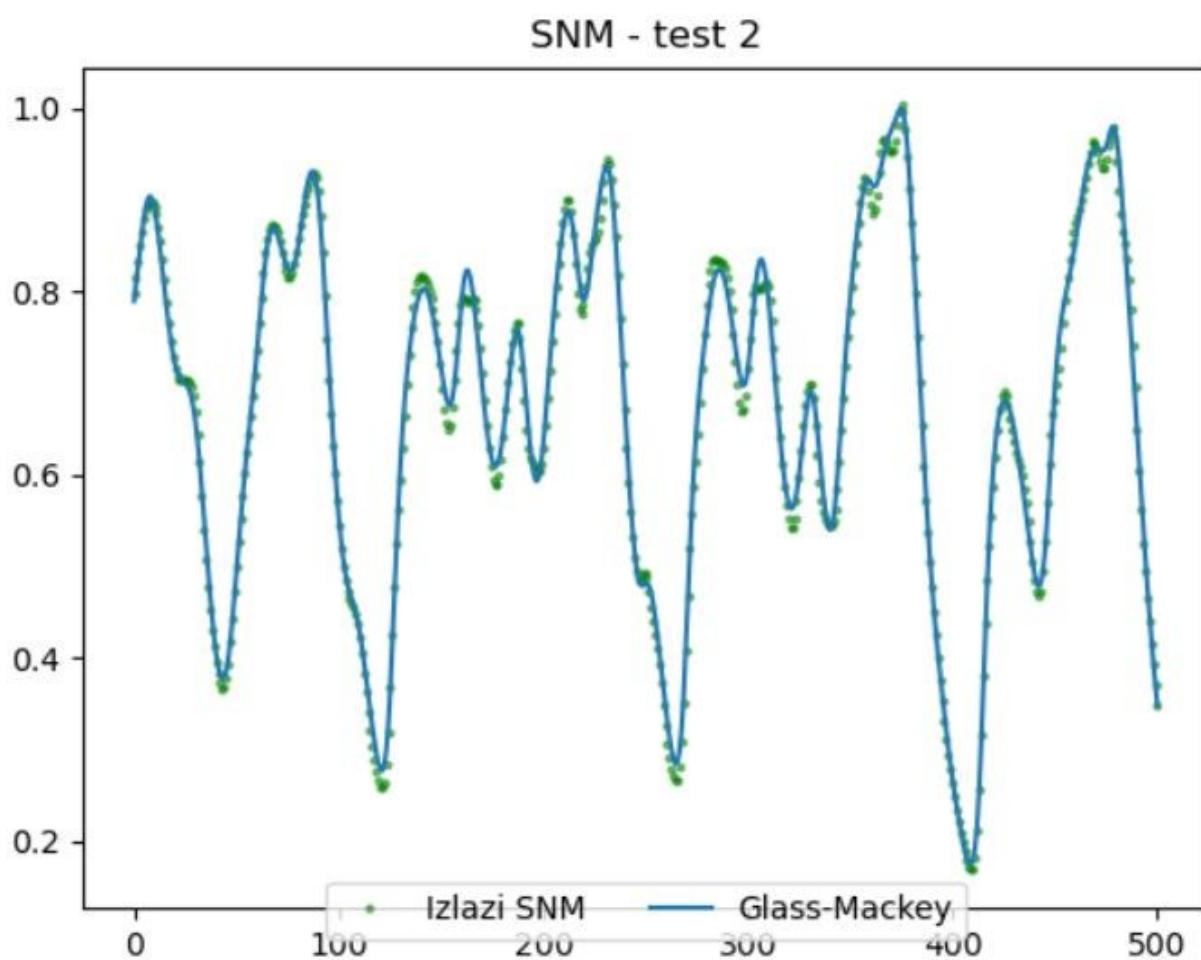
Predviđanje se provelo tako da se najprije generiralo 5 skupova podataka koji su normalizirani između 0 i 1. Jedan služi u učenju mreže, drugi u validaciji, a ostali su podaci za testiranje mreže. Uobičajeno je da se tijekom učenja svakih n koraka provede samo unaprijedna faza učenja na skupu podataka za validaciju korištenjem aktualnih parametara

učenja, te se izračuna NRMS. Konačne vrijednosti parametara učenja koje ćemo koristiti u fazi testiranja mreže nisu nužno vrijednosti koje je mreža generirala u zadnjem koraku učenja, već vrijednosti koji su dale najmanji *NRMS* na podacima za validaciju. Cilj ovakvog postupka je da se spriječi pretreniranost, odnosno loša generalizacija mreže što bi rezultiralo dobrim rezultatom na podacima za učenje, ali lošim rezultatom na testnim podacima.

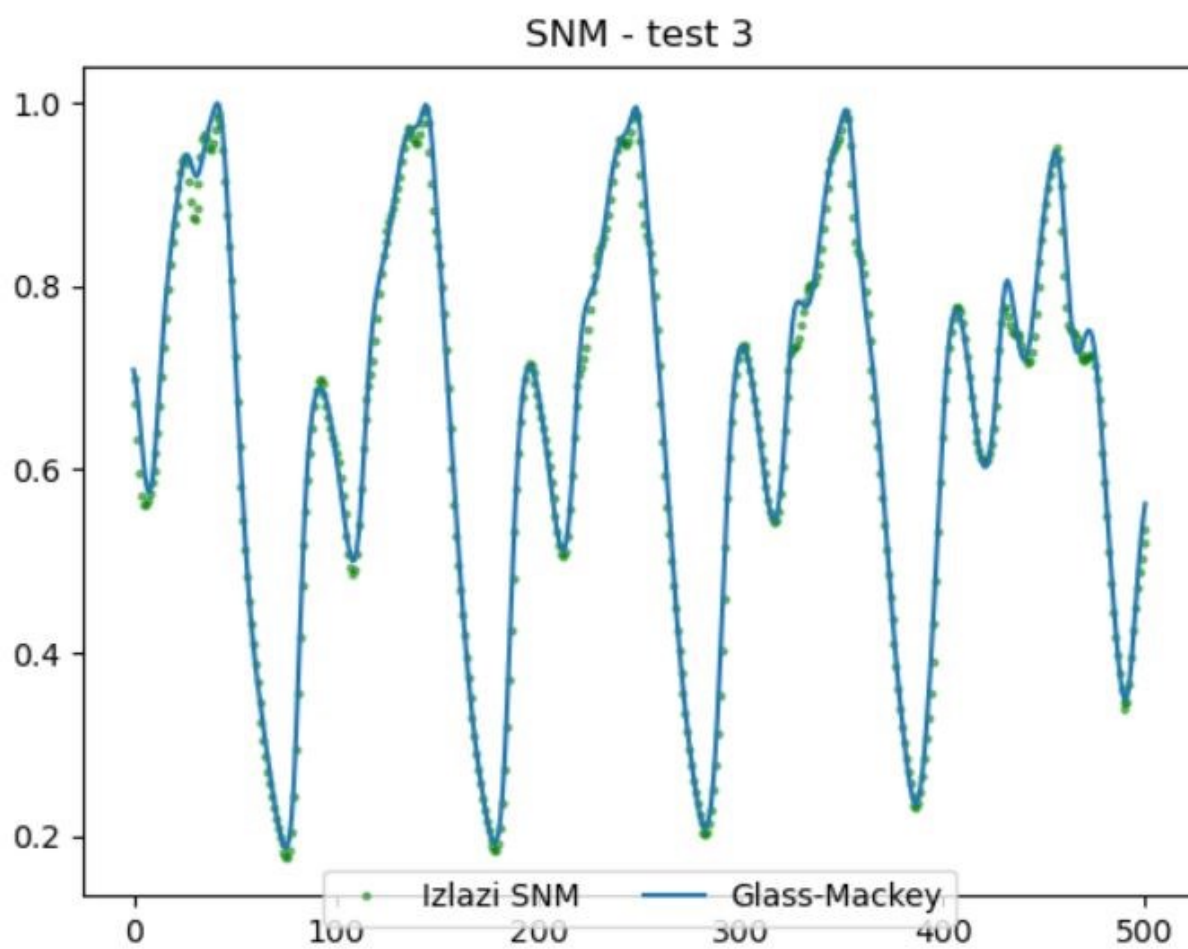
Pri prvom učenju su korišteni parametri preuzeti iz literature [1]. Korak učenja η postavljen je na 0.01, dok je momentum α jednak 0.8. Broj koraka učenja je pedeset tisuća, a broj skrivenih neurona osam (plus bias). Težinski koeficijenti su slučajno generirani brojevi između -0.1 i 0.1. Rezultati su u nastavku prikazani grafički i tablično



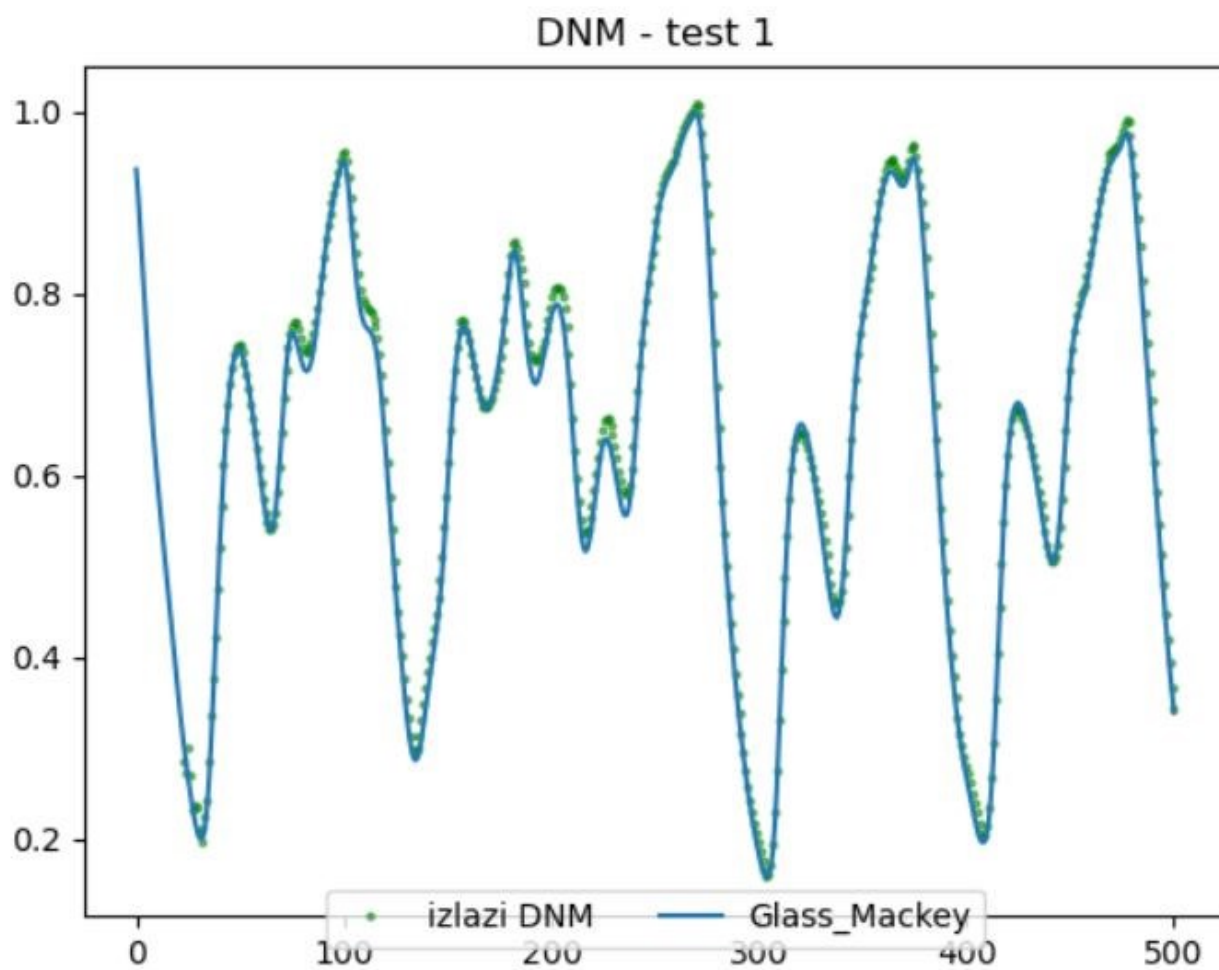
Slika 7. Odziv SNM na test 1



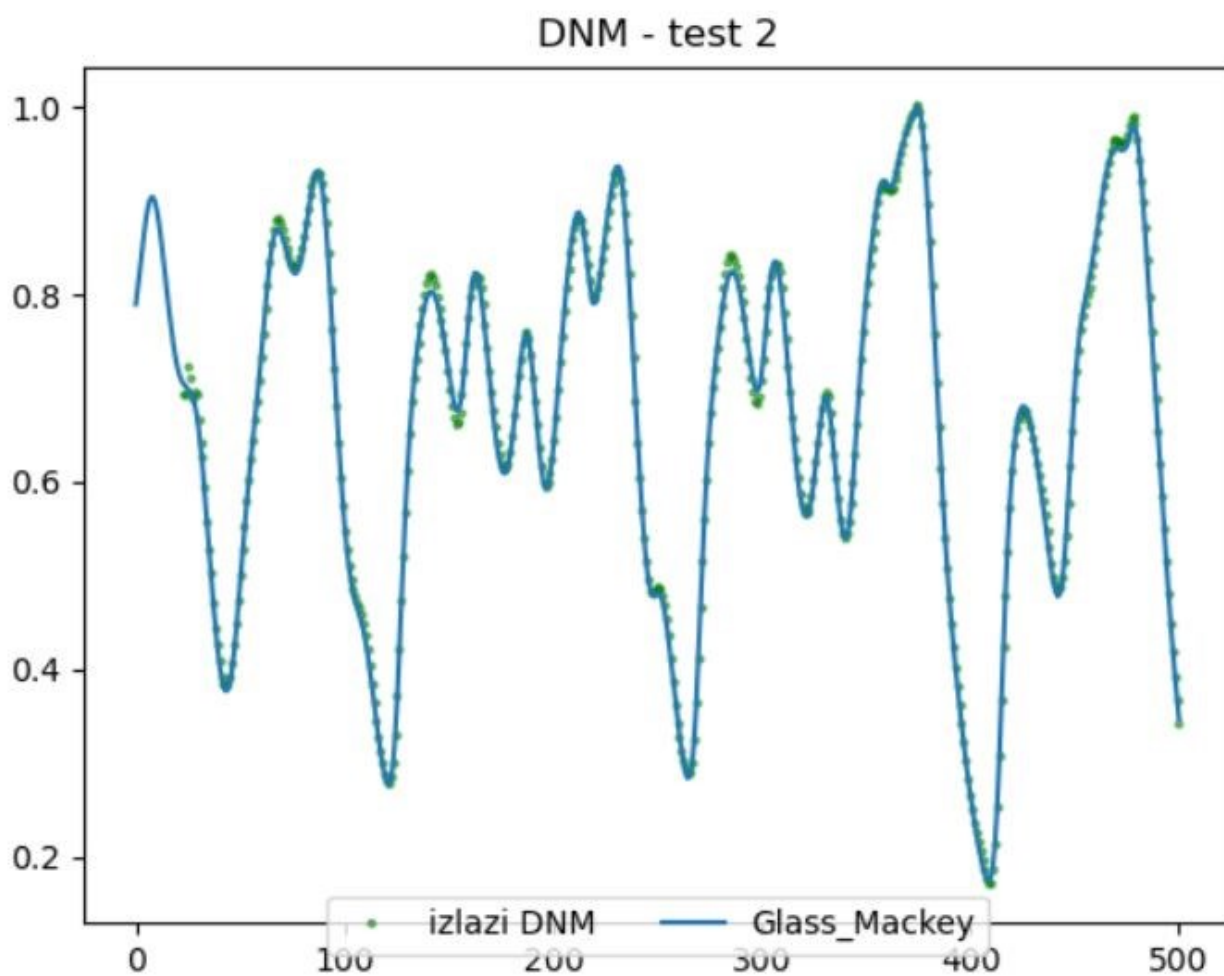
Slika 8. Odziv SNM na test 2



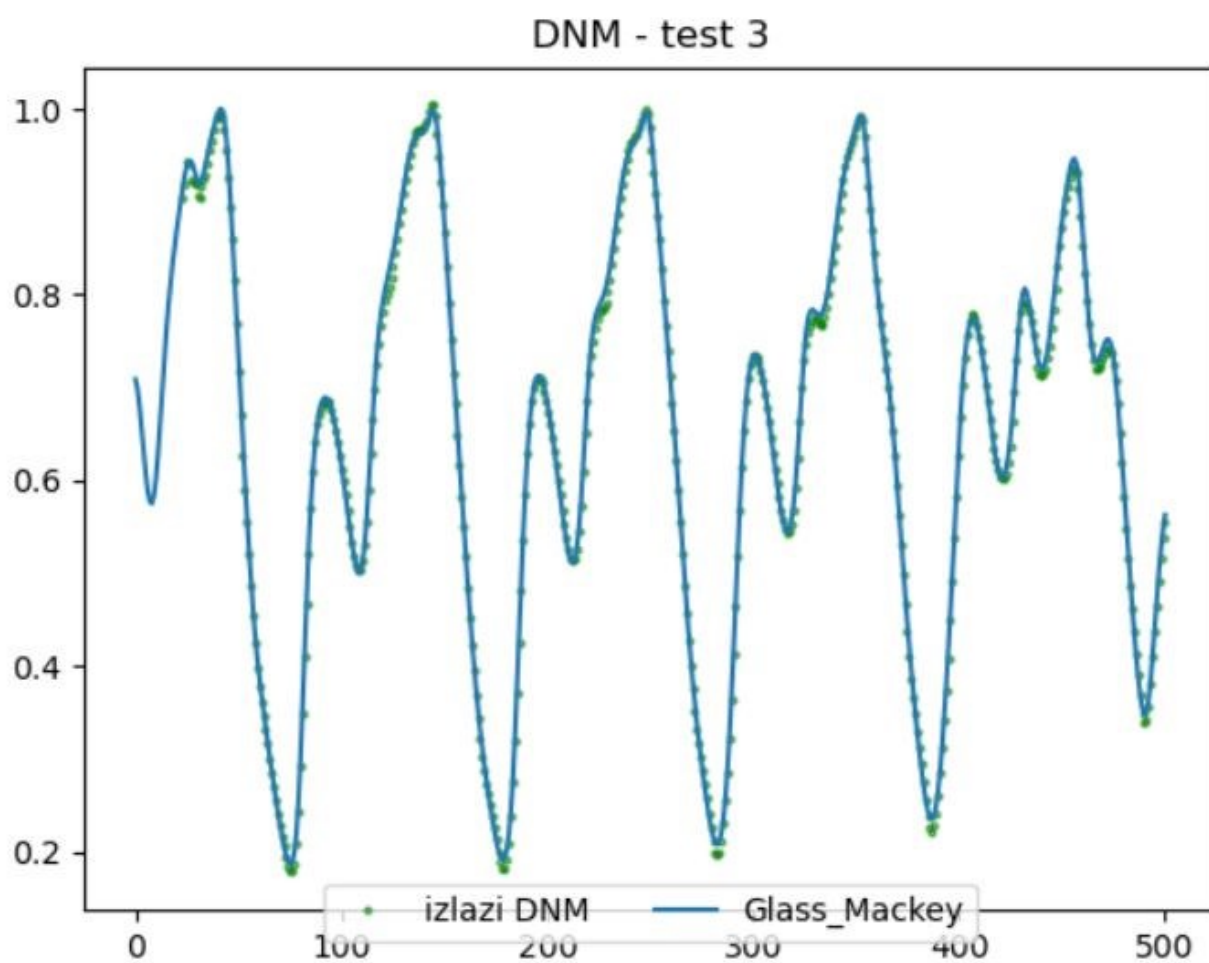
Slika 9. Odziv SNM na test 3



Slika 10. Odziv DNM na test 1



Slika 11. Odziv DNM na test 2



Slika 12. Odziv DNM na test 3

Tablica 1. Usporedba rezultata testova nakon prvog učenja

	Statička mreža	Dinamička mreža
Test 1 - NRMS	0.0778	0.06497
Test 2 - NRMS	0.0749	0.0308
Test 3 - NRMS	0.0796	0.03321

Osim s parametrima učenja iz literature, učenje je provedeno još i s vlastitim parametrima. Korak učenja η postavljen je na 0.007 a momentum α na 0.3. Broj neurona sakrivenog sloja je 10 (plus bias). Broj koraka učenja i dalje je pedeset tisuća. Težinski koeficijenti su slučajno generirani brojevi između -0.1 i 0.1. Rezultati su dani u tablici 2.

Tablica 2. Usporedba rezultata testova nakon drugog učenja

	Statička mreža	Dinamička mreža
Test 1 - NRMS	0.11143	0.08550
Test 2 - NRMS	0.09070	0.06863
Test 3 - NRMS	0.08635	0.06029

6. ZAKLJUČAK

Testirane su dvije konfiguracije statičke i dinamičke mreže s različitim parametrima i brojem sakrivenih neurona. Dinamička mreža se pokazala boljom na svakom testu, te možemo zaključiti da daje bolje rezultate kod predviđanja vremenskih serija i kaotičnih sustava u odnosu na statičku mrežu, iako i naizgled jednostavna statička mreža daje zadovoljavajuće rezultate.

LITERATURA

- [1] Novaković B., Majetić D., Široki M., Umjetna neuronske mreže, Fakultet strojarstva i brodogradnje, Zagreb, 1998.
- [2] Michael Nielsen, Neural networks and deep learning
- [3] Mandan M. Gupta, Liang Jin, Noriyashu Homma, Static and dynamic neural networks