

Unaprijedna troslojna perceptronska neuronska mreža s adaptivnom ReLu aktivacijskom funkcijom

Penava, Luka

Undergraduate thesis / Završni rad

2025

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:235:913047>

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

Download date / Datum preuzimanja: **2025-02-27**

Repository / Repozitorij:

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



SVEUČILIŠTE U ZAGREBU
FAKULTET STROJARSTVA I BRODOGRADNJE

ZAVRŠNI RAD

Luka Penava

Zagreb, 2024.

SVEUČILIŠTE U ZAGREBU
FAKULTET STROJARSTVA I BRODOGRADNJE

ZAVRŠNI RAD

Mentor:

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

Student:

Luka Penava

Zagreb, 2024.

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

Najiskrenije zahvaljujem svom mentoru prof. dr. sc. Danku Brezaku na nesebičnom trudu, pomoći i susretljivosti, kao i na vrijednim savjetima i strpljivosti u prenošenju znanja tijekom izrade ovog rada.

Posebno zahvaljujem svojoj obitelji i prijateljima, osobito svojem djedu Dušanu, koji su mi pružali neizmjernu podršku i razumijevanje tokom studiranja.

Luka Penava



SVEUČILIŠTE U ZAGREBU
FAKULTET STROJARSTVA I BRODOGRADNJE



Središnje povjerenstvo za završne i diplomске ispite
Povjerenstvo za završne i diplomске 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 / 25 - 06 / 1	
Ur.broj: 15 - 25 -	

ZAVRŠNI ZADATAK

Student: **Luka Penava** JMBAG: **0035225814**

Naslov rada na hrvatskom jeziku: **Unaprijedna troslojna perceptronska neuronska mreža s adaptivnom ReLU aktivacijskom funkcijom**

Naslov rada na engleskom jeziku: **A feedforward three-layered perceptron neural network with the adaptive ReLU activation function**

Opis zadatka:

Najčešće korištena aktivacijska funkcija dubokih neuronskih mreža je ReLU (*Rectified Linear Unit*) funkcija. Navedena funkcija omogućila je brzo učenje i dobra generalizacijska svojstva i kod mreža s jednim sakrivenim slojem. Pritom je u tim analizama korištena ReLU aktivacijska funkcija bez mogućnosti učenja ili adaptacije njezinih parametara. Kako biološki neuroni imaju različite pragove aktivacije, logično je i umjetne neurone modelirati pomoću aktivacijskih funkcija s istim svojstvom, tj. s mogućnošću adaptacije njihovih parametara. Stoga je u ovom radu potrebno provesti usporednu analizu klasične ReLU aktivacijske funkcije i njezinih adaptivnih varijanti primjenom nekoliko standardnih usporednih testova.

U radu je potrebno:

- Definirati matematički model i izraditi programsku podršku unaprijedne perceptronske neuronske mreže učene EBP metodom učenja, čiji neuroni sakrivenog sloja sadrže standardnu i adaptivnu varijantu ReLU aktivacijske funkcije.
- Generirati odzive za sve analizirane strukture neuronske mreže primjenom nekoliko standardnih usporednih testova.
- Usporediti dobivene rezultate različitih struktura mreže.
- Izvesti zaključke rada.

U radu je potrebno navesti korištenu literaturu i eventualno dobivenu pomoć.

Zadatak zadan:

30. 11. 2024.

Zadatak zadao:

Prof. dr. sc. Danko Brezak

Datum predaje rada:

1. rok: 20. i 21. 2. 2025.
2. rok: 10. i 11. 7. 2025.
3. rok: 18. i 19. 9. 2025.

Predvideni datumi obrane:

1. rok: 24. 2. - 28. 2. 2025.
2. rok: 15. 7. - 18. 7. 2025.
3. rok: 22. 9. - 26. 9. 2025.

Predsjednik Povjerenstva:

Prof. dr. sc. Damir Godec

SADRŽAJ

SADRŽAJ	I
POPIS SLIKA	III
POPIS TABLICA	V
POPIS KRATICA	VI
SAŽETAK	VII
SUMMARY	VIII
1. UVOD	1
1.1. Neuronske mreže	1
1.2. Struktura neuronskih mreža.....	1
1.3. Učenje neuronskih mreža	1
1.4. Funkcija cilja	1
1.5. Unaprijedna faza učenja	2
1.6. Povratna faza učenja	3
1.7. Topologija mreže	6
2. Usporedba aktivacijskih funkcija i programska podrška za učenje	8
2.1. Uloga aktivacijskih funkcija.....	8
2.2. Usporedba aktivacijskih funkcija	8
2.3. Adaptacije ReLU funkcije.....	9
2.4. Promjena parametara ReLU aktivacijske funkcije	10
2.5. Realizacija algoritma neuronske mreže	12
3. Opis provedenih testova	13
3.1. Identifikacija linearnog sustava	13
3.2. Identifikacija nelinearnog sustava	14
3.3. Predviđanje kaotičnih vremenskih serija	15
3.4. Standardizacija parametara	17
3.5. Mjera točnosti algoritma učenja	17
4. Rezultati testova.....	19
4.1. Rezultati identifikacije linearnog sustava	19
4.1.1. ReLU aktivacijska funkcija	19
4.1.2. ReLU s prilagodljivim koeficijentom nagiba	22
4.1.3. ReLU s prilagodljivim pragom aktivacije	26
4.1.4. Tablični prikaz NRMS-a	29
4.2. Rezultati identifikacije nelinearnog sustava.....	29
4.2.1. ReLU aktivacijska funkcija	30
4.2.2. ReLU s prilagodljivim koeficijentom nagiba	33
4.2.3. ReLU s prilagodljivim pragom aktivacije	36
4.2.4. Tablični prikaz NRMS-a	40
4.3. Rezultati predviđanja kaotičnog sustava.....	40
4.3.1. ReLU aktivacijska funkcija	40
4.3.2. ReLU s prilagodljivim koeficijentom nagiba	43
4.3.3. ReLU s prilagodljivim pragom aktivacije	47

4.3.4. Tablični prikaz NRMS-a	50
5. Zaključak	51
LITERATURA.....	52

POPIS SLIKA

Slika 1.1.	Opća struktura mreže.....	7
Slika 2.1.	Grafički prikaz aktivacijskih funkcija	9
Slika 3.1.	Odziv linearnog sustava.....	14
Slika 3.2.	Odziv nelinearnog sustava	15
Slika 3.3.	Odziv kaotičnog sustava	16
Slika 4.2.	Odziv mreže nakon učenju skupa s NRMS parametrom u konfiguraciji 2-5-1.....	19
Slika 4.1.	Odziv mreže nakon učenju skupa s NRMS parametrom u konfiguraciji 2-2-1.....	19
Slika 4.3.	Odziv mreže nakon učenju skupa s NRMS parametrom u konfiguraciji 2-10-1.....	20
Slika 4.4.	Odzivi mreže P1 člana na testnim primjerima pri konfiguraciji a) 2-2-1 b) 2-5-1	21
Slika 4.5.	Odzivi mreže P1 člana na testnim primjerima pri konfiguraciji 2-10-1.....	22
Slika 4.6.	Odziv mreže nakon učenja skupa s NRMS parametrom u konfiguraciji 2-2-1	23
Slika 4.7.	Odziv mreže nakon učenju skupa s NRMS parametrom u konfiguraciji 2-5-1.....	23
Slika 4.8.	Odziv mreže nakon učenju skupa s NRMS parametrom u konfiguraciji 2-10-1.....	23
Slika 4.9.	Odzivi mreže P1 člana na testnim primjerima pri konfiguraciji a) 2-2-1 b) 2-5-1	24
Slika 4.10.	Odzivi mreže P1 člana na testnim primjerima pri konfiguraciji 2-10-1.....	25
Slika 4.11.	Odziv mreže nakon učenju skupa s NRMS parametrom u konfiguraciji 2-2-1	26
Slika 4.12.	Odziv mreže nakon učenju skupa s NRMS parametrom u konfiguraciji 2-5-1	26
Slika 4.13.	Odziv mreže nakon učenju skupa s NRMS parametrom u konfiguraciji 2-10-1....	26
Slika 4.14.	Odzivi mreže P1 člana na testnim primjerima pri konfiguraciji a) 2-2-1 b) 2-5-1 ..	27
Slika 4.15.	Odzivi mreže P1 člana na testnim primjerima pri konfiguraciji 2-10-1.....	28
Slika 4.16.	Odziv mreže nakon učenju skupa s NRMS parametrom u konfiguraciji 2-2-1.....	30
Slika 4.17.	Odziv mreže nakon učenju skupa s NRMS parametrom u konfiguraciji 2-5-1.....	30
Slika 4.18.	Odziv mreže nakon učenju skupa s NRMS parametrom u konfiguraciji 2-10-1....	31
Slika 4.19.	Odzivi mreže nelinearnog sustava na testnim primjerima pri konfiguraciji a) 2-2-1 b) 2-5-1	32
Slika 4.20.	Odzivi mreže nelinearnog sustava na testnim primjerima pri konfiguraciji 2-10-1 .	33
Slika 4.21.	Odziv mreže nakon učenju skupa s NRMS parametrom u konfiguraciji 2-2-1.....	34
Slika 4.22.	Odziv mreže nakon učenju skupa s NRMS parametrom u konfiguraciji 2-5-1.....	34
Slika 4.23.	Odziv mreže nakon učenju skupa s NRMS parametrom u konfiguraciji 2-10-1.....	34
Slika 4.24.	Odzivi mreže nelinearnog sustava na testnim primjerima pri konfiguraciji a) 2-2-1 b) 2-5-1	35
Slika 4.25.	Odzivi mreže nelinearnog sustava na testnim primjerima pri konfiguraciji 2-10-1 .	36
Slika 4.26.	Odziv mreže nakon učenju skupa s NRMS parametrom u konfiguraciji 2-2-1.....	37
Slika 4.27.	Odziv mreže nakon učenju skupa s NRMS parametrom u konfiguraciji 2-5-1.....	37
Slika 4.28.	Odziv mreže nakon učenju skupa s NRMS parametrom u konfiguraciji 2-10-1.....	37
Slika 4.29.	Odzivi mreže nelinearnog sustava na testnim primjerima pri konfiguraciji a) 2-2-1 b) 2-5-1	38
Slika 4.30.	Odzivi mreže nelinearnog sustava na testnim primjerima pri konfiguraciji 2-10-1 .	39
Slika 4.31.	Odziv mreže nakon učenju skupa s NRMS parametrom u konfiguraciji 2-2-1.....	41
Slika 4.32.	Odziv mreže nakon učenju skupa s NRMS parametrom u konfiguraciji 2-5-1.....	41
Slika 4.33.	Odziv mreže nakon učenju skupa s NRMS parametrom u konfiguraciji 2-10-1.....	41
Slika 4.34.	Odzivi mreže kaotičnih vremenskih serija na testnim primjerima pri konfiguraciji a) 2-2-1 b) 2-5-1	42
Slika 4.35.	Odzivi mreže kaotičnih vremenskih serija na testnim primjerima pri konfiguraciji 2-10-1.....	43
Slika 4.36.	Odziv mreže nakon učenju skupa s NRMS parametrom u konfiguraciji 2-2-1.....	44
Slika 4.38.	Odziv mreže nakon učenju skupa s NRMS parametrom u konfiguraciji 2-10-1.....	44
Slika 4.37.	Odziv mreže nakon učenju skupa s NRMS parametrom u konfiguraciji 2-5-1.....	44

Slika 4.39. Odzivi mreže kaotičnih vremenskih serija na testnim primjerima pri konfiguraciji a) 2-2-1 b) 2-5-1	45
Slika 4.40. Odzivi mreže kaotičnih vremenskih serija na testnim primjerima pri konfiguraciji 2-10-1.....	46
Slika 4.41. Odziv mreže nakon učenju skupa s NRMS parametrom u konfiguraciji 2-2-1.....	47
Slika 4.42. Odziv mreže nakon učenju skupa s NRMS parametrom u konfiguraciji 2-5-1.....	47
Slika 4.43. Odziv mreže nakon učenju skupa s NRMS parametrom u konfiguraciji 2-10-1.....	48
Slika 4.44. Odzivi mreže kaotičnih vremenskih serija na testnim primjerima pri konfiguraciji a) 2-2-1 b) 2-5-1	49
Slika 4.45. Odzivi mreže kaotičnih vremenskih serija na testnim primjerima pri konfiguraciji 2-10-1.....	50

POPIS TABLICA

Tablica 1	Krajnje vrijednosti NRMS-a za P1 član	29
Tablica 2	Krajnje vrijednosti NRMS-a za nelinearan sustav	40
Tablica 3	Krajnje vrijednosti NRMS-a za kaotični sustav.....	50

POPIS KRATICA

MSE eng. *Mean Square Error*

ReLU eng. *Rectified Linear Unit*

Tanh Tanges-hiperbolna

APRBS eng. *Amplitude modulated Pseudo-Random Binary Signal*

NRMS eng. *Normalized Root Mean Square error*

MAE eng. *Mean Absolute Error*

RMS eng. *Root Mean Square*

SAŽETAK

U višeslojnim perceptronskim mrežama najčešće korištene aktivacijske funkcije su sigmoidalna i tangens-hiperbolna. Za razliku od toga, u dubokim mrežama dominira ReLU funkcija uvedena radi ublažavanja problema nestajućeg gradijenta, prisutnog kod sigmoidalnih i tangens-hiperbolnih funkcija. Iako ReLU funkcija rješava taj problem, njena primjena donosi i nova ograničenja poput pojave neaktivnih neurona. Stoga se, u ovom radu, istražuju učinkovitost primjene ReLU funkcije, konkretno ReLU s prilagodljivim nagibom i ReLU s prilagodljivim pragom aktivacije. Cilj rada je analizirati svojstva adaptacija, fokusirajući se na identifikaciju dinamičkih sustava i predviđanje kaotičnih vremenskih serija. Performanse različitih aktivacijskih funkcija se uspoređuju analizom mjere točnosti algoritma učenja (NRMS) s ciljem identificiranja optimalnih rješenja za navedene probleme.

Usporedbom svih triju testova perceptronske neuronske mreže, može se zaključiti da ReLU aktivacijska funkcija s prilagodljivim pragom aktivacije pokazuje najbolja svojstva aproksimacije i generalizacije. Njena jednostavnost implementacije čini ju idealnom čak i za kompleksnije sustave. Osim toga njezina uporaba ne zahtjeva preveliku računalnu snagu, memoriju, niti dugo vrijeme treniranja.

Ključne riječi: Neuronske mreže; ReLU; Adaptive ReLU; Identifikacija dinamičkih sustava; Identifikacija kaotičnih sustava

SUMMARY

In multilayer perceptron networks, the most commonly used activation functions are sigmoidal and hyperbolic tangent. In contrast, deep networks are dominated by the ReLU function, which was introduced to mitigate the vanishing gradient problem present in sigmoidal and hyperbolic tangent functions. Although the ReLU function solves this problem, its application also introduces new limitations, such as the appearance of inactive neurons. Therefore, this paper investigates the effectiveness of applying ReLU function adaptations, specifically ReLU with adaptive slope and ReLU with adaptive activation threshold. The aim of the work is to analyze the properties of these adaptations, focusing on the identification of dynamic systems and the prediction of chaotic time series. The performance of different activation functions is compared by analyzing the measure of accuracy of the learning algorithm (NRMS), with the aim of identifying optimal solutions for the stated problems.

By comparing all three tests of the perceptron neural network, it can be concluded that the ReLU activation function with adaptive activation threshold exhibits the best approximation and generalization properties. Its simplicity of implementation makes it ideal even for more complex systems. Furthermore, its use does not require excessive computing power, memory, or long training times.

Key words: Neural Networks; ReLU; Adaptive ReLU; Dynamic Systems Identification; Chaotic Systems Identification

1. UVOD

1.1. Neuronske mreže

Neuronske mreže modeli su strojnog učenja inspirirani ljudskim mozgom. Sastoje se od povezanih čvorova (neurona), strukturiranih unutar slojeva koji obrađuju podatke i prilagođavaju svoje težine kako bi naučili složene odnose između ulaza i izlaza. Koriste se u rješavanju niza kompleksnih problema klasifikacije, regresije i predikcije paralelnim procesiranjem podataka.

1.2. Struktura neuronskih mreža

Struktura neuronskih mreža može biti jednoslojna i višeslojna. Jednoslojne mreže sastoje se od neurona ulaznog i izlaznog sloja dok višeslojne mreže imaju jedan ili više sakrivenih slojeva koji omogućuju modeliranje složenih nelinearnih sustava. To je najlakše predočiti tako da svaki sloj ima svoju zadaću. Ulazni sloj prima podatke i prosljeđuje ih mreži, sakriveni sloj obrađuju informaciju putem povezanih neurona te otkriva obrasce ponašanja, dok izlazni sloj generira konačan rezultat. Aktivacijske funkcije, integrirane unutar sakrivenog sloja, uvode nelinearnost unutar neuronske mreže, omogućujući joj učenje složenih obrazaca i odnosa među podacima. Višeslojne mreže uče kompleksne probleme, a njihovi parametri se podešavaju algoritmima poput povratnog rasprostiranja pogreške (eng. Gradient Descent Algorithm).

1.3. Učenje neuronskih mreža

Učenje parametara vrši se primjenom različitih metoda učenja, a sve u cilju dobivanja zadovoljavajućeg odziva mreže. Tri su glavne vrste učenja neuronskih mreža:

- učenje s učiteljem ili nadzirano učenje (eng. Supervised learning),
- učenje bez učitelja ili nenadzirano učenje (eng. Unsupervised learning),
- učenje s pojačanjem (eng. Reinforcement learning).

Unutar ovog rada korišteno je samo učenje s učiteljem (eng. Supervised learning). Treniranje mreže vrši se unutar dvije faze:

- unaprijedna faza,
- povratna faza.

1.4. Funkcija cilja

Funkcija cilja ključna je komponenta u procesu učenja ovakvih tipova sustava. Njena uloga je izračun razlike između stvarnih i željenih vrijednosti izlaza pomoću matematičkih izraza

specifičnih za dani problem. Tim procesom funkcija cilja omogućuje mreži da prilagođavanjem svojih težinskih parametara smanji grešku i poboljšava performanse. S obzirom na vrstu problema koji mreža treba riješiti možemo ju podijeliti na dva tipa. Prvi je vezan uz regresijske probleme, a drugi je tip vezan uz klasifikacijske probleme. Za regresijske zadatke, kojima je cilj predviđanje kontinuirane vrijednosti, najčešće korištene funkcije su:

- MSE (Mean Squared Error),
- MAE (Mean Absolute Error),
- Huber Loss (kombinacija gore spomenutih).

Za klasifikacijske zadatke, u kojima mreža predviđa pripadnost određenoj kategoriji, koriste se:

- Cross-Entropy Loss,
- Categorical Cross-Entropy i sl.

U ovom radu zbog svoje široke primjene i lakog deriviranja, upotrijebljena je MSE funkcija. Njena sposobnost da kvadriranjem razlike naglasi velike pogreške unutar sustava čini je idealnom pri ubrzavanju učenja. Formula za MSE (s dodatnim faktorom $\frac{1}{2}$ koji se često koristi u kontekstu neuronskih mreža) izgleda:

$$E = \frac{1}{2} \sum_{n=1}^N (d_n - O_n)^2, \quad (1.1)$$

gdje d_n predstavlja željene vrijednosti, a O_n predstavlja stvarne, odnosno dobivene rezultate.

1.5. Unaprijedna faza učenja

U unaprijednoj fazi učenja neuronske mreže prikupljaju se podaci s ulaza mreže Z , koji onda služe za izračun izlaznih vrijednosti O . Ovaj proces prolazi kroz sve slojeve mreže, od ulaznog do izlaznog sloja. Za sam proces izračuna potrebno je inicirati početne težine sakrivenog sloja V i izlaznog sloja W , kako bi u prvom prolazu mreža mogla dati izlazne vrijednosti. Početne težine je najčešće preporučljivo inicijalizirati slučajnim odabirom unutar intervala od -0.01 do 0.01 . U slučaju loše (niske) preciznosti, dodatno se smanjuje red veličine.

Ulazni i sakriveni sloj:

Ulazni podaci se pripremaju za prolazak kroz skrivene slojeve. Unose se u formi matrice (vektora) Z , dimenzije $n \times m$, gdje je “ n “ broj značajki u ulaznim podacima, a “ m “ broj ulaza. Svaki ulazni neuron povezan je s neuronima sljedećeg sloja pomoću težina. Težine su organizirane u matrici V

dimenzija $n \times m_1$, gdje je m_1 broj neurona unutar prvog skrivenog sloja. Svaka težina v_{ij} povezuje i -ti neuron prethodnog sloja s j -tim neuronom u idućeg.

$$\mathit{net}H_j = \sum_{i=1}^I v_{ij}Z_i; \quad j = 1, 2, \dots, J - 1; \quad i = 1, 2, \dots, I, \quad (1.2)$$

gdje je i označuje ulazne neurone, a j označuje neurone skrivenog sloja (njihov je broj uvećan za jedan, odnosno Bias neuron).

Aktivacijska funkcija neurona skrivenog sloja primijenjena u ovom radu je ReLU funkcija, prikazana slikom 2.1.c. Shodno tome izlaz skrivenog sloja računamo prema:

$$y_j = \begin{cases} \mathit{net}H_j, & \mathit{net}H_j > 0 \\ 0, & \mathit{net}H_j \leq 0 \end{cases}. \quad (1.3)$$

Izlazni sloj:

Izlazni sloj neuronske mreže predstavlja završni korak u procesu izračuna odziva mreže. Na temelju obrade svih prethodnih slojeva izračunava se u formi varijable $\mathit{net}O_k$ kao suma umnožaka izlaznih težina i vrijednosti neurona iz prethodnog sloja:

$$\mathit{net}O_k = \sum_{j=1}^J w_{kj}y_j; \quad k = 1, 2, \dots, K, \quad (1.4)$$

gdje K predstavlja broj neurona izlaznog sloja.

Navedena varijabla predstavlja argument aktivacijske funkcije neurona izlaznog sloja, a koja je u ovom radu izostavljena. Time se konačni odziv mreže može opisati kao:

$$O_k = \mathit{net}O_k. \quad (1.5)$$

1.6. Povratna faza učenja

Povratna faza učenja realizirana je primjenom algoritma povratnog rasprostiranja pogreške (eng. Error Back-Propagation). Ona se temelji na izračunu greške učenja svakog ulazno-izlaznog para prema stvarnoj izlaznoj vrijednosti O i željenom izlazu D . Ažuriranje težinskih faktora kao i ostalih parametara vrši se na osnovu izračunate pogreške. Ovdje je upotrebljavana već spomenuta funkcija cilja pod nazivom MSE. Cilj podešavanja težinskih (i ostalih) faktora je smanjenje vrijednosti funkcije cilja E . Težinski koeficijenti ϑ se adaptiraju prema izrazu:

$$\vartheta(n + 1) = \vartheta(n) + \Delta\vartheta(n), \quad (1.6)$$

gdje n predstavlja trenutni korak učenja, $\Delta\vartheta(n)$ veličinu promjene parametara učenja, a $\vartheta(n+1)$ novu vrijednost parametara učenja.

Kako bi pogrešku smanjivali najvećim mogućim iznosom, potrebno je izračunati gradijent pogreške prema:

$$\Delta E(\vartheta) = \frac{\partial E(\vartheta)}{\partial \vartheta}. \quad (1.7)$$

Budući da pomak u smjeru gradijentnog pada često nije optimalan i može dovesti do preskakanja optimalnog rješenja, uvodimo koeficijent učenja η . On omogućuje prilagodbu brzine učenja i poboljšava točnost modela.

Kako bi se dodatno ubrzao proces učenja, odnosno smanjio broj potrebnih koraka za postizanje iste razine pogreške, uvodi se parametar α poznat kao momentum. Promjena parametara tada se računa kao:

$$\Delta E(n) = -\eta \nabla E(\vartheta(n)) + \alpha \Delta \vartheta(n-1), \quad (1.8)$$

gdje (n) označava promjenu parametara učenja unutar razmatranog koraka, a $(n-1)$ promjenu unutar prethodnog koraka. Vrijednosti koeficijenta η i α odabire učitelj, najčešće unutar predloženih intervala. Konačno, dolazimo do završnog oblika izraza za promjenu parametara učenja:

$$\vartheta(n+1) = \Delta \vartheta(n) - \partial \nabla E(\vartheta(n)) + \alpha \Delta(n-1). \quad (1.9)$$

Promjena težina izlaznog sloja

Promjena težina između izlaznog i sakrivenog sloja odvija se prema 1.9 na sljedeći način:

$$w_{kj}(n+1) = w_{kj}(n) - \eta \nabla E(n) + \alpha \Delta w_{kj}(n-1), \quad (1.10)$$

gdje se gradijent pogreške računa prema izrazu:

$$\nabla E(n) = \frac{\partial E(n)}{\partial w_{kj}}, \quad (1.11)$$

te se može lako odrediti primjenom uzastopnih parcijalnih derivacija:

$$\frac{\partial E(n)}{\partial w_{kj}} = \frac{\partial E(n)}{\partial O_k} \frac{\partial O_k}{\partial net_{Ok}} \frac{\partial net_{Ok}}{\partial w_{kj}}. \quad (1.12)$$

Svaki član uzastopne parcijalne derivacije možemo jednostavno izračunati:

$$\frac{\partial E(n)}{\partial o_k} = -(d_k - O_k), \quad (1.13)$$

$$\frac{\partial O_k}{\partial net_{Ok}} = 1, \quad (1.14)$$

$$\delta O_k = d_k - O_k, \quad (1.15)$$

$$\frac{\partial O_k}{\partial w_{kj}} = y_j. \quad (1.16)$$

Uvrštavanjem gore navedenih izraza 1.13, 1.14, 1.15 i 1.16 u izraz 1.12 dobivamo:

$$\nabla E(n) = \frac{\partial E(n)}{\partial w_{kj}} = -(d_k - O_k)y_j = \delta_{Ok}y_j, \quad (1.17)$$

Ako se jednakost 1.17 pridruži izrazu 1.10, dobivamo konačan izraz koji opisuje promjenu težina izlaznog sloja:

$$w_{kj}(n+1) = w_{kj}(n) - \eta \delta_{Ok}y_j + \alpha \Delta w_{kj}(n-1). \quad (1.18)$$

Promjena težina skrivenog sloja

Izraz za izračun težinskih faktora koji povezuju ulazni sa skrivenom slojem glasi:

$$v_{ji}(n+1) = v_{ji}(n) - \eta \nabla E(n) + \alpha \Delta v_{ji}(n-1). \quad (1.19)$$

Vodeći se prema istoj logici kao u izlaznom sloju, gradijent pogreške može se lako raspisati:

$$\frac{\partial E(n)}{\partial v_{ji}} = \frac{\partial E(n)}{\partial y_j} \frac{\partial y_j}{\partial net_{Hj}} \frac{\partial net_{Hj}}{\partial v_{ji}}, \quad (1.20)$$

a pojedini članovi niza računaju se kao:

$$\frac{\partial E(n)}{\partial y_j} = - \sum_{k=1}^K \delta_{Ok} \omega_{kj}. \quad (1.21)$$

Derivacija aktivacijske funkcije ovisi o izabranoj funkciji. Za običnu ReLU funkciju izraz za derivaciju glasi:

$$\frac{\partial y_j}{\partial net_{Hj}} = \begin{cases} 1, & net_{Hj} > 0 \\ 0, & net_{Hj} \leq 0 \end{cases}. \quad (1.22)$$

Derivacija skrivenog sloja određuje se iz izraza 1.20 kao:

$$\frac{\partial net_{Hj}}{\partial v_{ji}} = Z_i. \quad (1.23)$$

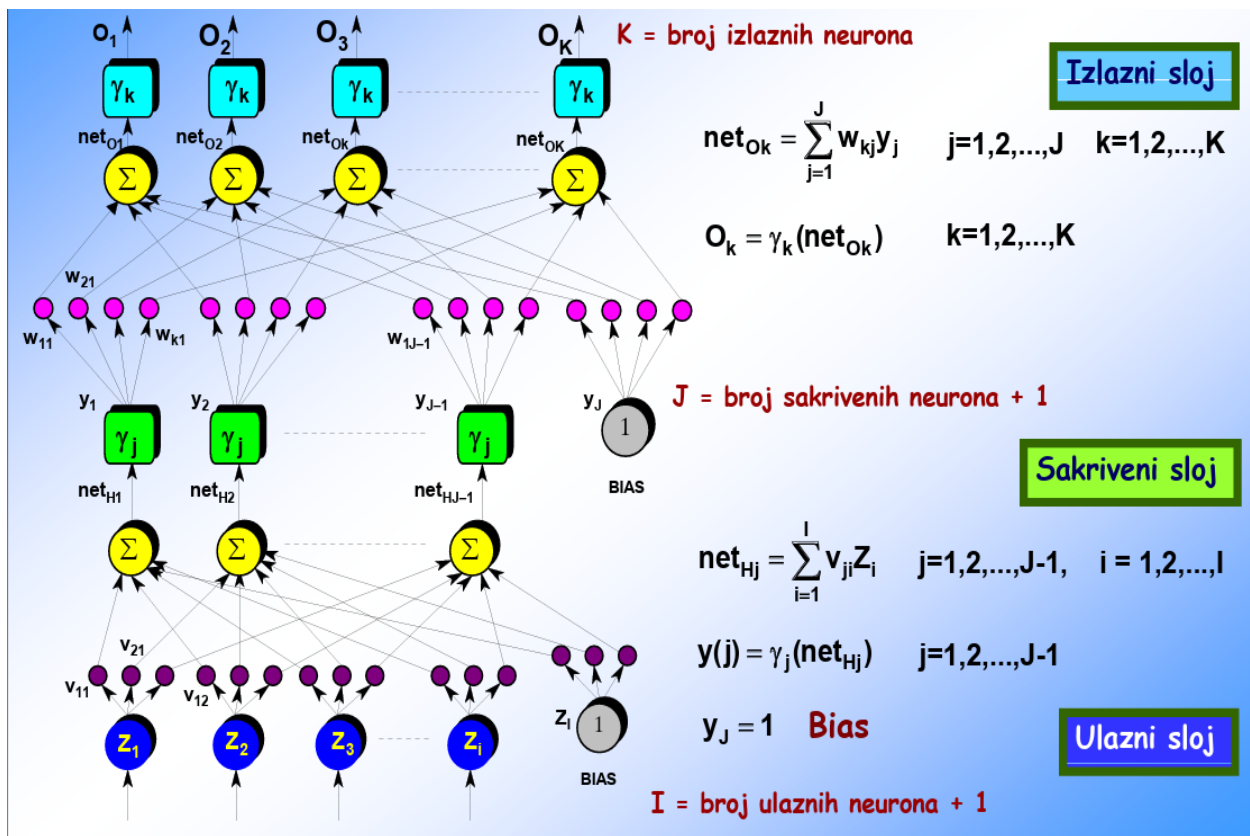
Uvrštavanjem gore spomenutih izraza 1.20, 1.21, 1.22, 1.23 u izraz 1.19 dobiven je konačni izraz za promjenu težinskih faktora skrivenog sloja:

$$v_{ji}(n+1) = v_{ji}(n) + \eta \frac{\partial y}{\partial net_{Hj}} Z_i \sum_{k=1}^K \delta_{Ok} \omega_{kj} + \alpha \Delta v_{ji}(n-1). \quad (1.24)$$

Za ostale faktore unutar višeslojne mreže numerički postupak ostaje isti kao za prvi skriveni sloj, ali se u uzastopnim parcijalnim derivacijama javljaju dodatni članovi. Promjenom aktivacijske funkcije mijenja se samo pripadajući član derivacije, dok ostali članovi ostaju nepromijenjeni.

1.7. Topologija mreže

Promjena arhitekture mreže ima nekoliko važnih uloga. Različiti problemi zahtijevaju različite kapacitete mreže. Ako se mreža sastoji od nedovoljnog broja neurona, neće imati dovoljan kapacitet da nauči složenije obrasce što može dovesti do pojave podtreniranosti (eng. “underfitting-a”). S druge strane prevelik broj neurona može dovesti do pretreniranosti (eng. “overfitting-a”) pri čemu mreža, osim relevantnih obrazaca, uči i potencijalne šumove unutar podataka. Uz prethodno navedene pojave promjenom strukture mreže traži se ravnoteža između preciznost i efikasnosti koju mreža može postići. Povećanjem broja neurona povećava se broj parametara koje je potrebno trenirati, što ujedno produžuje vrijeme potrebno za treniranje mreže. Osim toga povećava se i potreba za većim alociranjem memorije, kako bi se navedeni parametri pohranili. Unutar ovog rada korištene su strukture s 2, 5 i 10 neurona sakrivenog sloja referirajući se na provedene testove opisane u poglavlju 3. Osim promjene kapaciteta mreže, kod nekih testova dolazi i do promjene broja ulaznih neurona. Na slici 1.1. prikazana je struktura mreže u općoj formi u kojoj I označava broj ulaznih neurona, K broj izlaznih neurona, a J predstavlja broj neurona unutar sakrivenog sloja. Također na slici se nalazi i Bias koji služi povećanju fleksibilnosti sustava i poboljšanju njegova učenja. Njegovim uvođenjem omogućujemo krajnjoj funkciji pomak, a samim time i bolju aproksimaciju podataka.



Slika 1.1. Opća struktura mreže

U praksi se odabiru strukture mreže ovisno o problem koji ona uči. U ovom se radu izbor arhitekture mreže vrši putem dimenzija matrica težina ($n \times m$) sadržanih unutar tekstualnih datoteka ("V" i "W"). Povećavanjem broja redova matrice "V" (težine ulaznog sloja) i broja stupaca matrice "W" (težine izlaznog sloja) povećava se broj neurona unutar zadane strukture. Također dodavanjem dodatnih stupaca u matricu "V" omogućava se povećanje broja ulaza, čime se zadovoljava potreba za većim brojem ulaznih podataka.

2. Usporedba aktivacijskih funkcija i programska podrška za učenje

2.1. Uloga aktivacijskih funkcija

Kao što je već izneseno u uvodu, uvođenjem aktivacijskih funkcija omogućeno je učenje složenijih nelinearnih odnosa između podataka što je u praksi čest slučaj. Također osim nelinearnosti, primjenom aktivacijske funkcije možemo regulirati vrijednosti izlaza neurona unutar željenog spektra kao i utjecati na brzinu i kvalitetu treniranja eliminacijom neželjenih stanja (nestajući gradijent).

2.2. Usporedba aktivacijskih funkcija

U slučaju perceptronskih mreža, najčešće korištene aktivacijske funkcije su sigmoidna, hiperbolička tangens (Tanh), Gaussova, sinusna i druge. Iako češće korištena kod dubinskih neuronskih mreža s više sakrivenih slojeva, u ovom je radu analizirana ReLU (eng. Rectified Linear Unit) aktivacijska funkcija. Dok sigmoidna i tanges-hiperbolna funkcije pate od pojave saturacije i problema nestajućeg gradijenta, ReLU funkcija pokazuje prednost u takvim situacijama. S druge strane jedna od glavnih mana ReLU funkcije proizlazi iz izostanka aktivnosti neurona u slučaju da mu je odziv negativna vrijednost. Dodatno, treba napomenuti da je sigmoidna funkcija, sa svojim binarnim svojstvima, karakterističnija za klasifikaciju, dok je ReLU često prikladnija za proces regresije čime se različite funkcije optimalno primjenjuju ovisno o analitičkoj tehnici i problemu.

Izrazi za pojedine aktivacijske funkcije:

- Sigmoidna funkcija

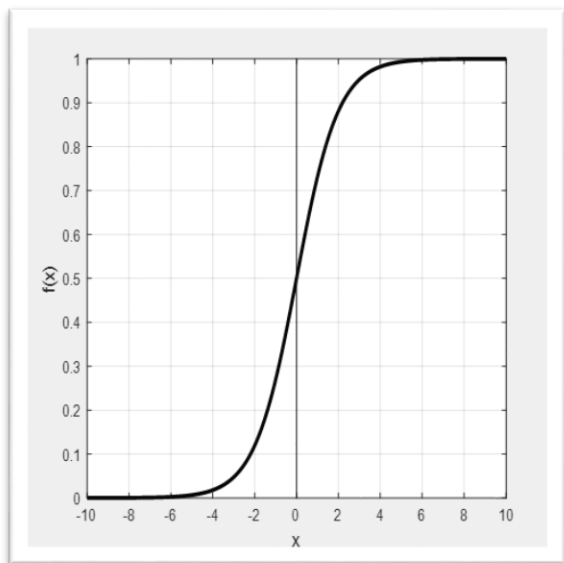
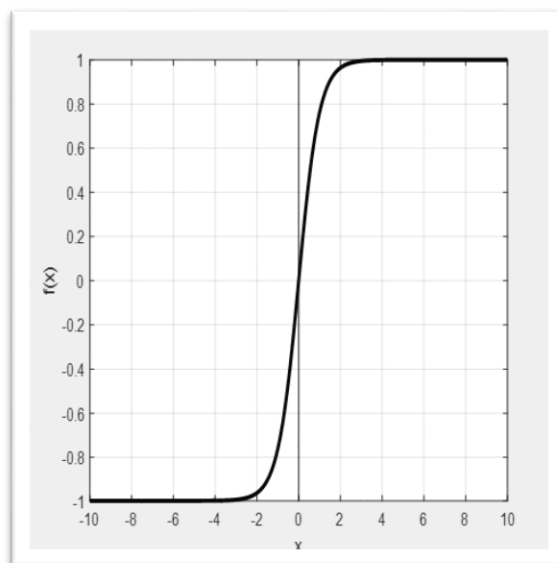
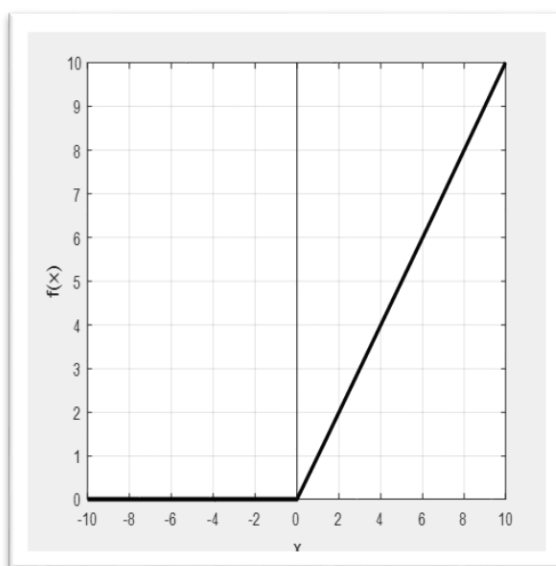
$$f(x) = \frac{1}{1 + e^{-x}}$$

- Tangens hiperbolna funkcija

$$f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$

- ReLU funkcija

$$f(x) = \begin{cases} netH_j, & net H_j > 0 \\ 0, & netH_j \leq 0 \end{cases}$$

**(c) Sigmoidna funkcija****(c) Tangens hiperbolna funkcija****(c) ReLU funkcija****Slika 2.1. Grafički prikaz aktivacijskih funkcija**

2.3. Adaptacije ReLU funkcije

Zahvaljujući svojim mnogobrojnim prednostima u odnosu na ostale funkcije ReLU aktivacijska funkcija osobito je popularna u konvolucijskim mrežama. Neke od njezinih prednosti navedenih u prethodnom odlomku uključuju rješavanje problema nestajućeg gradijenta i bolju konvergenciju tijekom treniranja. Osim toga važna je i jednostavnost i učinkovitost pri implementaciji, kao i minimalni računalni resursi. No uz sve navedene benefite tradicionalna

ReLU funkcija nosi i jedno bitno ograničenje u vidu takozvanih “mrtvih neurona“. Ovaj problem nastaje kada neuroni postanu “trajno neaktivni“ i prestanu doprinositi modelu učenja, što se događa kada ReLU funkcija prima negativne ulazne vrijednosti te kao rezultat postavlja izlaz na nulu. Takav problem značajno smanjuje kapacitet modela za učenje i otežava prepoznavanje složenih uzoraka. Adaptacijom osnovne ReLU funkcije omogućena je primjena prilagodljivih pragova aktivacije i učenje koeficijenta nagiba. Time je ostvaren inovativan pristup u kojem se različiti neuroni aktiviraju na različite načine što bolje oponaša ponašanje biološkog neurona i osigurava robusniji rad mreže. U daljnjem dijelu rada cilj je eksperimentalno ispitati učinkovitost ovakvih funkcija na standardiziranim testovima, primjenom različitih struktura mreže.

2.4. Promjena parametara ReLU aktivacijske funkcije

Budući da je ovaj rad baziran na testiranju adaptivnih ReLU funkcija, potrebno je predstaviti i matematiku iza učenja navedenih koeficijenata. U radu se javljaju dvije adaptivne funkcije koje ćemo dalje zvati “ReLU s prilagodljivim pragom aktivacije“ i “ReLU s prilagodljivim koeficijentom nagiba“. Za svaku od navedenih funkcija temeljna jednadžba adaptacije ne razlikuje se od one za učenje težinskih koeficijenata

ReLU s prilagodljivim koeficijentom nagiba

$$\beta_j(n+1) = \beta(n) - \eta \nabla E(n) + \alpha \Delta \beta_j(n-1), \quad (2.1)$$

gdje $\beta_j(n)$ označava trenutni koeficijent nagiba ReLU funkcije.

Gradijent raspisujemo kao:

$$\frac{\partial E(n)}{\partial \beta_j} = \frac{\partial E(n)}{\partial O_k} \frac{\partial O_k}{\partial net_{Ok}} \frac{\partial net_{Ok}}{\partial y_j} \frac{\partial y_j}{\partial \beta_j}, \quad (2.2)$$

gdje je derivacija pogreške po y_j izračunata i prikazana u promjeni težine skrivenog sloja.

Derivaciju y_j po nagibu (β_j) možemo raspisati kao:

$$\frac{\partial y_j}{\partial \beta_j} = \begin{cases} netH_j, & netH_j > 0 \\ 0, & netH_j \leq 0 \end{cases}, \quad (2.3)$$

što je vrlo intuitivno jer je izgled same funkcije opisan jednadžbom:

$$y_j = \begin{cases} \beta_j netH_j, & netH_j > 0 \\ 0, & netH_j \leq 0 \end{cases}. \quad (2.4)$$

Nakon uvrštavanja, izgled jednadžbe adaptacije parametra nagiba izgleda ovako:

$$\beta_j(n+1) = \beta(n) + \eta \text{net}H_j \left(\sum_{k=1}^K \delta_{Ok} \omega_{kj} \right) + \alpha \Delta \beta_j(n-1), \quad (2.5)$$

no, kako je opisano jednadžbom 2.3, vrijednost derivacije može poprimiti i nulu te se, u tom slučaju, kompletan drugi član ove funkcije može zanemariti.

ReLU s prilagodljivim pragom aktivacije

$$\theta_j(n+1) = \theta(n) - \eta \nabla E(n) + \alpha \Delta \theta_j(n-1), \quad (2.6)$$

gdje $\theta_j(n)$ označava trenutni prag funkcije.

Gradijent pogreške raspisuje se kao:

$$\frac{\partial E(n)}{\partial \theta_j} = \frac{\partial E(n)}{\partial O_k} \frac{\partial O_k}{\partial \text{net}_{Ok}} \frac{\partial \text{net}_{Ok}}{\partial y_j} \frac{\partial y_j}{\partial \theta_j}, \quad (2.7)$$

gdje se derivacija pogreške po y_j ponovo javlja te ju možemo iščitati iz promjene težine skrivenog sloja.

Derivaciju y_j po θ_j može se raspisati:

$$\frac{\partial y_j}{\partial \theta_j} = \begin{cases} 0, & \text{net}H_j > \theta_j \\ 1, & \text{net}H_j \leq \theta_j \end{cases}, \quad (2.8)$$

što je, kada se pogleda izraz same funkcije, i očekivano:

$$y_j = \begin{cases} \text{net}H_j, & \text{net}H_j > \theta_j \\ 0, & \text{net}H_j \leq \theta_j \end{cases}. \quad (2.9)$$

Napokon, dolazimo do izraza za adaptaciju parametra θ_j koji, nakon uvrštavanja, izgleda ovako:

$$\theta_j(n+1) = \theta_j(n) + \eta \left(\sum_{k=1}^K \delta_{Ok} \omega_{kj} \right) + \alpha \Delta \theta_j(n-1), \quad (2.10)$$

gdje ponovo moramo paziti o rasponu sume unutar kojeg se funkcija nalazi. U slučaju da je raspon manji od zadanog pomaka (θ_j), drugi član gornje jednadžbe ponovno se isključuje jer vrijednost derivacije prelaze u 0.

Uvođenjem navedenih dodatnih parametara za adaptaciju osnovne varijante ReLU funkcije postižu se uvjeti za efikasnije učenje složenijih nelinearnih odnosa među podacima što može značajno poboljšati performanse u neuronskim mrežama. Njihova sposobnost prilagodbe

različitim ulaznim podacima čini ih vrijednim alatom u razvoju naprednih sustava temeljenih na umjetnoj inteligenciji.

2.5. Realizacija algoritma neuronske mreže

Algoritam za učenje i testiranje analiziranih varijanti neuronskih mreža, odnosno aktivacijskih funkcija neurona skrivenog sloja, napisan je u programskom paketu MATLAB. Riječ je o višenamjenskom programskom jeziku koji pruža interaktivno okruženje, prvenstveno dizajnirano za matematičke i znanstvene izračune. Svoju popularnost unutar raznih disciplina pronalazi upravo zbog svoje svestranosti te moćnih alata za analizu podataka, simulaciju i vizualizaciju.

Sam kod obrađuje niz tekstualnih datoteka koje sadrže podatke potrebne za treniranje i testiranje neuronskih mreža. Ključne datoteke primijenjene u rad programa su: **D.txt**, **Z.txt**, **V.txt**, **Z.txt** i dodatno generirane datoteke za spremanje rezultata poput **korak_NRMS.txt** i **NRMS_test.txt**. Učitavanje ovih datoteka obavlja se pomoću funkcije **importdata**, no prije samog učitavanja provjerava se i postojanost iste pomoću funkcije **exist**. Datoteke je potrebno pronaći putem radnog direktorija s panela prikaza mape (gore). Na ovaj način osigurava se integritet podataka prije pokretanja algoritma. Same datoteke se sastoje od niza brojeva koje su formirane u redovima i stupcima, a koje Matlab učitava i pohranjuje u obliku matrica. Pri izmjeni datoteka mora se paziti na razmak između brojčanih vrijednosti, kao i na simbole koji se upotrebljavaju kako bi se ispravno učitali. Rad s tekstualnim datotekama u ovom kodu ima ključnu ulogu u upravljanju s ulaznim i izlaznim podacima kao i sa samom strukturom mreže (spomenutom unutar topologije mreže). Također se osigurava i spremanje rezultata u standardiziranom formatu koji je kasnije lako obradiv.

3. Opis provedenih testova

U prethodnim poglavljima prikazana je struktura statičke neuronske mreže s povratnim rasprostiranjem pogreške, kao i njezine varijacije aktivacijskih funkcija. Navedene strukture neuronskih mreža analizirat će se korištenjem tri vrste testova. Zadaća provedenih ispitivanja bit će usporedba učinkovitosti adaptacije različitih parametara aktivacijske funkcije. Za prvi test uzet je problem identifikacije jednostavnog dinamičkog člana prvog reda, poznatog kao “proporcionalni član s kašnjenjem – P1“. Njegovo ponašanje dobit ćemo simuliranjem diferencijalne jednadžbe na računalo, a rezultat ćemo iskoristiti za učenje mreže. Po istom principu za sljedeći je test uzet problem identifikacije nelinearnog dinamičkog sustava. Njegovo ponašanje također dobivamo računalnom simulacijom njegove diferencijalne jednadžbe. Zadnji test sastojati će se od predviđanja ponašanja jednog tipičnog nelinearnog kaotičnog sustava pod nazivom Glass-Mackeyeva sustav. Ovakav sustav služi za analizu kapaciteta mreže u predviđanju složenih, nelinearnih i dijelom kaotičnih sustava. Navedeni testovi predstavljaju standardne referentne testove (eng. benchmark test) za određivanje i usporedbu performansi algoritama neuronskih mreža.

3.1. Identifikacija linearnog sustava

Linearni sustav opisan je dinamikom člana P1. Ovu dinamiku određuje poznata diferencijalna jednadžba prvog reda:

$$Tx(t) + x(t) = K_p u(t), \quad (3.1)$$

gdje T predstavlja vremensku konstantu, a K_p konstantu pojačanja sustava.

Zbog otežane primjene kontinuirane jednadžbe kao izvora informacija za učenje, spomenuta je jednadžba diskretizira se na sljedeći način:

$$x(n) = \frac{T_0}{T + T_0} \left[\frac{T}{T_0} x(n-1) + K_p u(n) \right], \quad (3.2)$$

gdje $x(n-1)$ označava prošlu izlaznu vrijednost, $x(n)$ i $u(n)$ prikazuju trenutne izlazne i ulazne vrijednosti, a T_0 ukazuje na vrijednost perioda uzrokovanja. Parametri simulacije, korišteni za generiranje vrijednosti namijenjenih učenju i testiranju, postavljeni su na:

- $T = 1$ s,
- $T_0 = 0.2$ s,
- $K_p = 1$.

Član se pobuđuje odskočnom pobudnom funkcijom, čime dobivamo sljedeći odziv sustava:

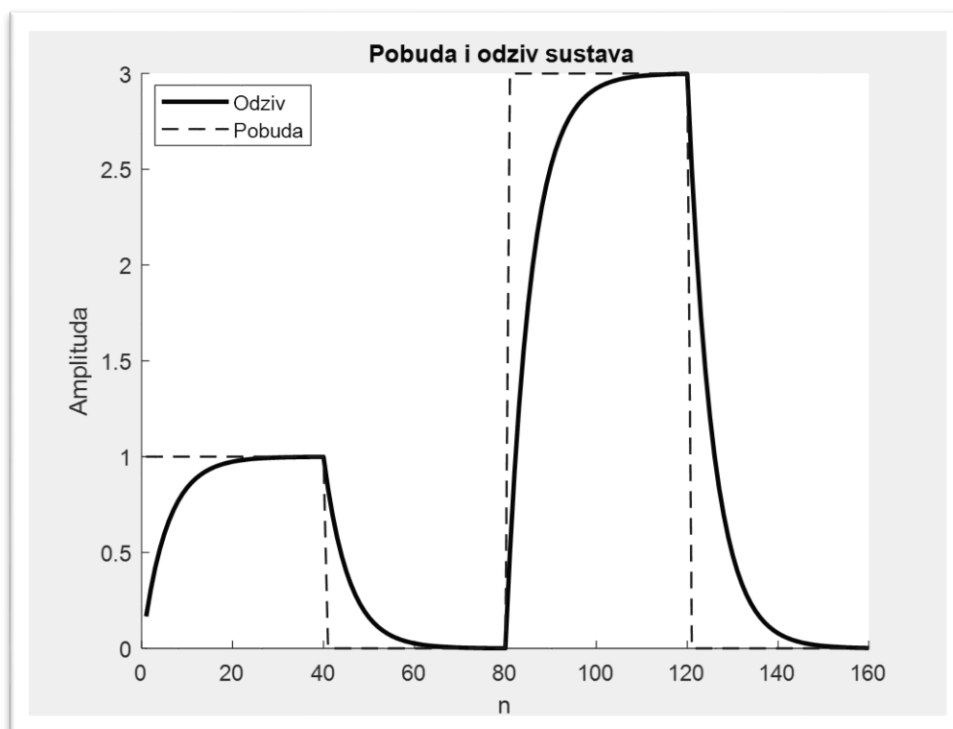
$$u(n) = 1, \quad n = 0, 2, \dots, 39,$$

$$u(n) = 0, \quad n = 40, 41, \dots, 79,$$

$$u(n) = 3, \quad n = 80, 81, \dots, 119,$$

$$u(n) = 0, \quad n = 120, 121, \dots, 159,$$

Grafički prikaz odziva linearnog sustava (P1 člana) na pobudu u gore danoj funkciji, prikazan je slikom 3.1. Ona ujedno i predstavlja skup podataka za učenje.



Slika 3.1. Odziv linearnog sustava

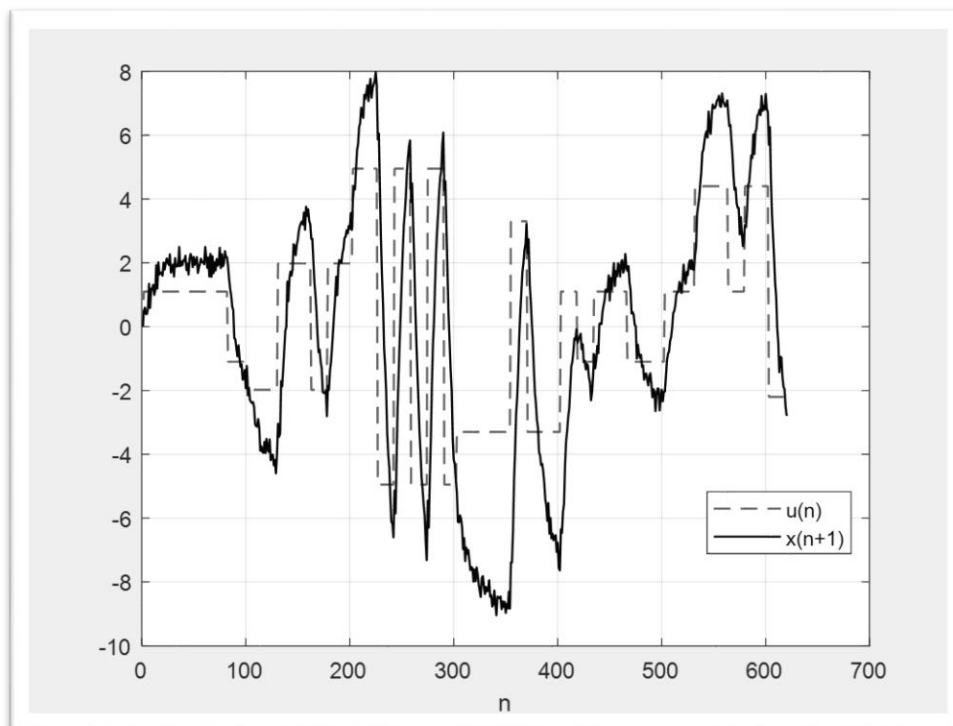
3.2. Identifikacija nelinearnog sustava

Idući model za ispitivanje mreže primjer je nelinearnog dinamičkog sustava čije se ponašanje kroz vrijeme opisuje diferencijalnom jednačbom prvog reda. Uz period uzrokovanja od 1 sekunde, te vremensku konstantu sustava od 10 sekundi diskretizirana diferencijalna jednačba ovakvog tipa modela glasi:

$$x(n + 1) = (0.9 - 0.003x(n))x(n) + 0.2u(n). \quad (3.3)$$

U cilju postizanja adekvatnog odziva ovakvog sustava, a na temelju kojeg će mreža biti u stanju identificirati njegovu dinamiku, za pobudnu funkciju korišten je pseudo-binarni signal slučajne frekvencije i amplitude (APRBS - engl. Amplitude modulated Pseudo-Random Binary Signal).

Ulazno-izlazni podaci potrebni za učenje mreže generirani su simulacijom jednačbe 3.3 na uzorku od 620 točaka. Grafički prikaz podataka prikazan je slikom 3.2.:



Slika 3.2. Odziv nelinearnog sustava

Iz grafa se vidi da veliki skokovi ulaznog signala uzrokuju izraženu nelinearnost. Kod manjih skokova ponekad se javlja linearan rast, a ponekad ponovno nelinearnost. Također odziv na negativne skokove značajno se razlikuje od odziva na pozitivne. Testiranje naučenih mreža izvedeno je novim APRBS signalima generirani na uzorku od 400 točaka.

3.3. Predviđanje kaotičnih vremenskih serija

Jedan od danas najčešće korištenih problema za testiranje neuronskih mreža problem je predviđanja ponašanja kaotičnog sustava. Poznato je da su ovakvi sustavi često, pa čak i uz današnju primjenu raznih matematičkih modela teško predvidljivi zbog svoje složene, često nedeterminističke dinamike.

Jedan od najčešće korištenih primjera nelinearnog kaotičnog sustava je Glass-Mackeyev nelinearni kaotični sustav. Opisan je nelinearnom diferencijalnom jednačinom (s kašnjenjem τ) te određen izrazom:

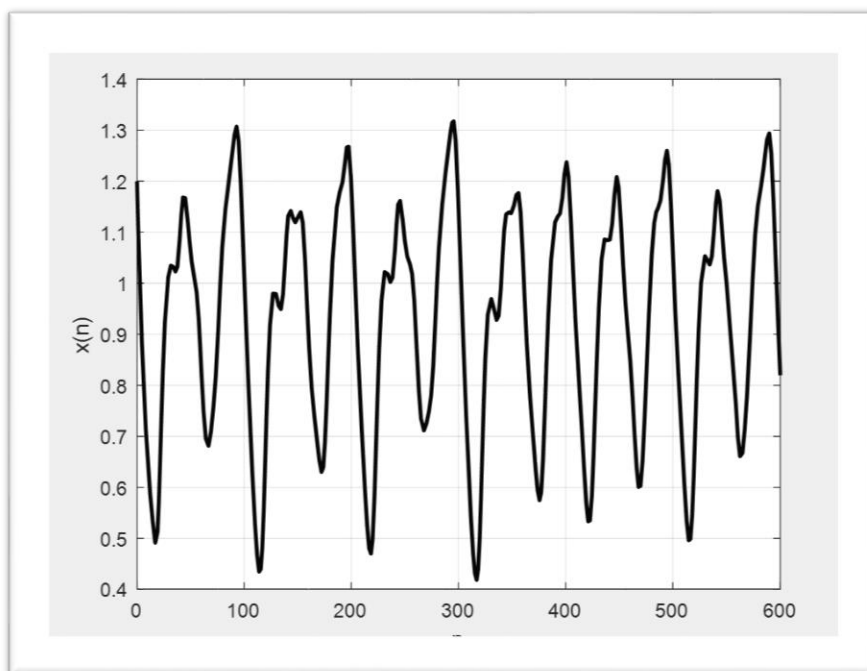
$$\dot{x} = \frac{\alpha x(t - \tau)}{1 + x^{10}(t - \tau)} - bx(t), \quad (3.4)$$

gdje α i b predstavljaju parametre sustava, a t oznaku vremena.

Kako bismo iz ove jednadžbe dobili podatke za učenje te kasnije testiranje mreže, potrebno ju je diskretizirati. Diskretizaciju vršimo postavljanjem parametara $\alpha = 0.2$, $b = 0.1$ i $\tau = 30$ uz početna stanja $x(0) = 0.1$, $T_0 = 1$ s. Dobivena jednadžba tada glasi:

$$x(n) = \frac{1}{1+b} \left[x(n-1) + \frac{\alpha x(n-\tau)}{1+x^{10}(n-\tau)} \right]. \quad (3.5)$$

Grafički prikaz Glass-Mackeyove vremenske serije možemo vidjeti na slici 3.3.:



Slika 3.3. Odziv kaotičnog sustava

Predviđanje ponašanja nelinearnog kaotičnog sustava temelji se na upotrebi prošlih vrijednosti vremenske serije do određene točke za predikciju budućih. Standardna metoda za ovakav tip predikcije je:

$$x(n+P) = f(x(n), x(n-\Delta), x(n-2\Delta), \dots, x(n-m\Delta)), \quad (3.6)$$

gdje je P broj točaka vremena za predviđanje u budućnost, Δ je kašnjenje signala, a m cjelobrojna konstanta.

Odabirom $P = \Delta$ omogućuje se predviđanje ponašanja sustava P koraka unaprijed pri čemu se izlazne vrijednosti sustava u određenim intervalima ponovno vraćaju na ulaz. U ovom radu odabrano je $P = \Delta = 6$.

3.4. Standardizacija parametara

Kako bi rezultati testova bili što vjerodostojniji, potrebno je prije svakog treniranja standardizirati početne parametre i način prikaza rezultata. Svako ispitivanje sprovedeno je primjenom triju različitih topologija s dva, pet ili deset neurona unutar sakrivenog sloja. Na početku učenja, vrijednosti težina za sve prisutne probleme jednake su i nasumično generirane pri čemu se iznosi kreću unutar intervala od 0.1 do -0.1. Koeficijent brzine učenja i moment također su strogo definirani i ostaju nepromijenjeni kroz sva tri primjera. Jedina iznimka je treći primjer u kojem se brzina učenja smanjuje za jedan red veličine radi postizanja veće točnosti, što će biti prikazano u idućem poglavlju. Također osim standardizacije osnovnih parametara potrebno je i ujednačeno prikazivanje rezultata. Izlazne vrijednosti mreže prikazivat će se grafički radi bolje vizualizacije, a ocjena uspješnosti učenja bit će opisana putem normaliziranog korijena srednje kvadratne pogreške (NRMS - engl. Normalized Root Mean Square error).

3.5. Mjera točnosti algoritma učenja

Mjera točnosti algoritma učenja ključna je vrijednost za procjenu kvalitete i učinkovitosti sustava. Ovakav tip mjere temelji se na razlici između željenog i stvarnog odziva mreže. Nužno je korištenje iste metode izračuna kako bi se osigurala usporedivost dobivenih rezultata. Različite mjere koriste se za izračunavanje uspješnosti ovakvih sustava. Neke od najčešćih metoda su:

- Srednja apsolutna pogreška (MAE)
- Srednja kvadratna pogreška (MSE),
- Korijen srednje kvadratne pogreške (RMS),
- Normalizirani korijen srednje kvadratne pogreške (NRMS).

U ovom radu korištena je NRMS metoda za ocjenu uspješnosti učenja. Njena glavna prednost leži u bezdimenzijskoj neovisnosti što omogućuje neovisnost mjera o dimenzijama učenih veličina. Time se olakšava usporedba različitih algoritama učenja, bez obzira na korišteni program. Metoda izračuna NRMS-a prilično je jednostavna i prikazuje se sljedećim formulama:

$$\bar{d} = \frac{1}{N} \sum_{n=1}^N d_n, \quad (3.7)$$

gdje N označava ukupan broj, a suma zbraja sve željene izlaze sustava.

$$\sigma_{d_n} = \sqrt{\frac{1}{N} \sum_{n=1}^N (d_n - \bar{d})^2}, \quad (3.8)$$

računanje standardne devijacije podataka σ_{d_n} putem srednje vrijednosti skupa \bar{d} .

$$NRMS = \frac{RMS}{\sigma_{d_n}} = \frac{\sqrt{\frac{\sum_{n=1}^N (d_n - O_n)^2}{N}}}{\sigma_{d_n}}, \quad (3.9)$$

izračun NRMS-a preko stvarnih vrijednosti mreže O_n .

Vrijednosti NRMS-a pohranjene su za svaki korak učenja te će biti prikazane grafički radi lakšeg praćenja dinamike učenja. Rezultati testiranja također su pohranjeni te će biti obrađeni u nastavku.

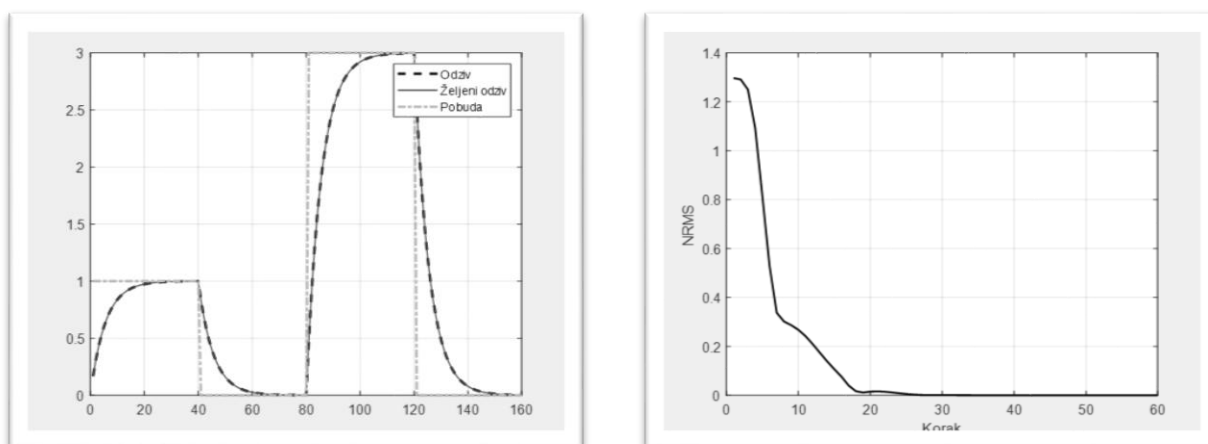
4. Rezultati testova

4.1. Rezultati identifikacije linearnog sustava

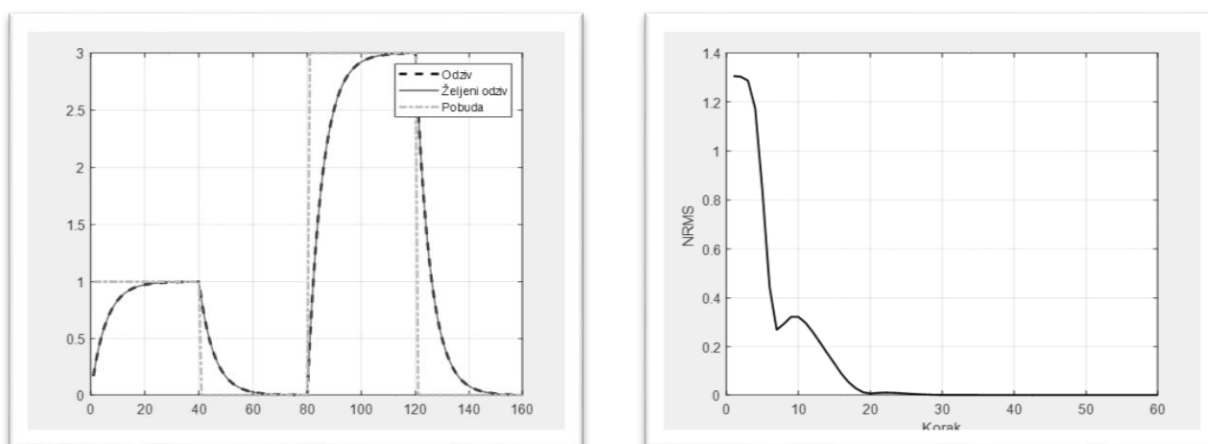
Učenje i testiranje naučene mreže provodi se pomoću skokovityh funkcija koje su označene linijom crta-točka-crta. Sve pobudne funkcije diskretizirane su na 80 točaka osim onih korištenih za učenje koje sadrže 160 točaka. Testne pobude nalaze se unutar raspona od 0.5 do 3.0. Odziv mreže na grafovima je prikazan isprekidanom linijom, dok je željeni odziv prikazan punom linijom. Tijekom faze učenja moment je isključen, a koeficijent učenja iznosi 0.1. Učenje je provedeno kroz 3000 koraka, ali su grafovi NRMS-a ciljano suženi kako bi prikazali bitnije informacije. Krajnje vrijednost NRMS-a su pohranjene unutar Tablice 1.

4.1.1. ReLU aktivacijska funkcija

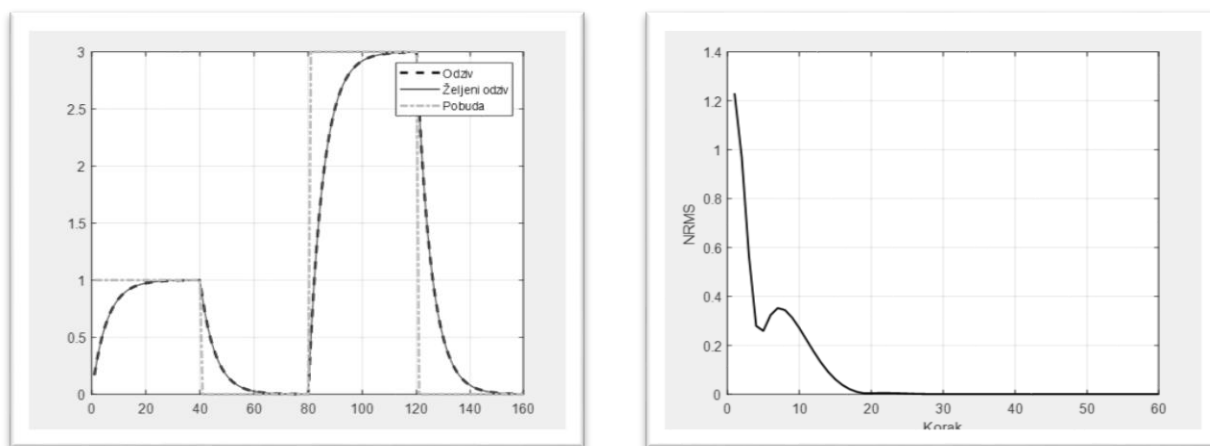
Prema slikama 4.1. 4.2. 4.3. i Tablici 1 vidljivo je da već obična ReLU aktivacijska funkcija u svim strukturama odlično aproksimira željeni odziv P1 člana. NRMS izrazito brzo konvergira i nakon faze učenja on se kreće oko vrijednosti 1.78×10^{-5} .



Slika 4.2. Odziv mreže nakon učenja skupa s NRMS parametrom u konfiguraciji 2-2-1



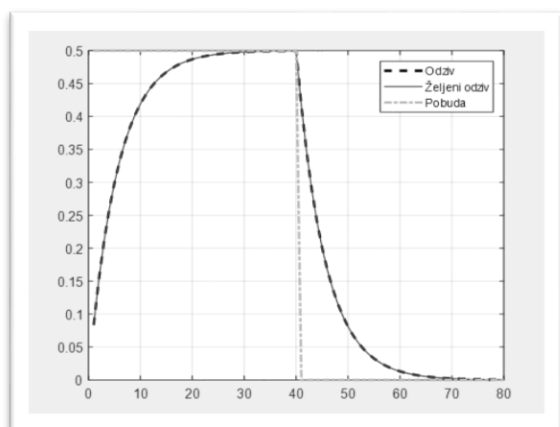
Slika 4.1. Odziv mreže nakon učenja skupa s NRMS parametrom u konfiguraciji 2-5-1



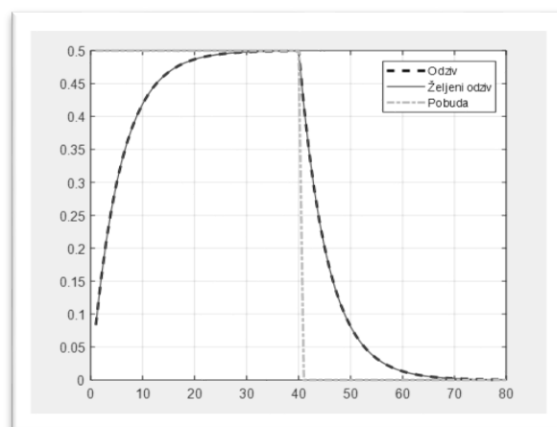
Slika 4.3. Odziv mreže nakon učenja skupa s NRMS parametrom u konfiguraciji 2-10-1

Usporedbom grafova 4.1., 4.2. i 4.3. može se uočiti da povećanje broja neurona u skrivenom sloju ne utječe značajno na preciznost niti na brzinu konvergiranja NRMS-a budući da se konačne vrijednosti postižu već unutar 20 koraka.

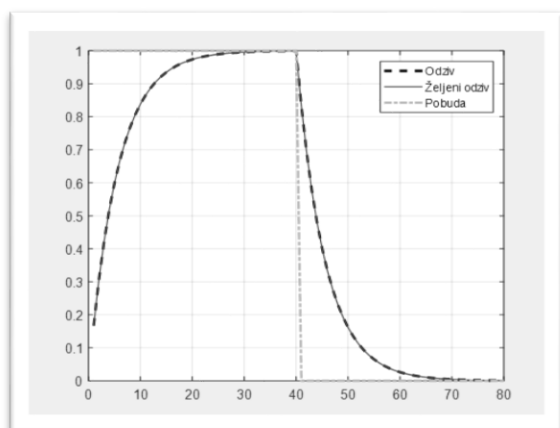
U fazi testiranja (prema slikama 4.4. i 4.5. te Tablici 1) uočljiva su izvrsna generalizacijska i aproksimacijska svojstva mreže. Također se slika se ne mogu iščitati značajne razlike između željenog i stvarnog odziva mreže niti pojave pretreniranosti ili podtreniranosti. Vrijednosti NRMS-a su izrazito niske te se razlike javljaju tek na petoj decimali.



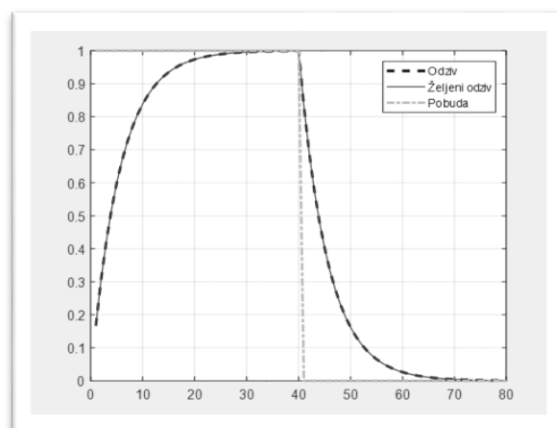
Test 0.5



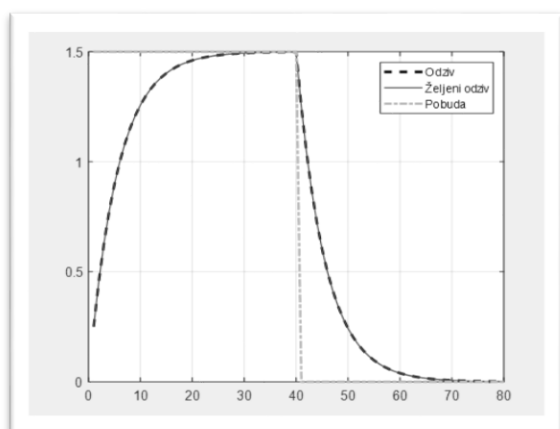
Test 0.5



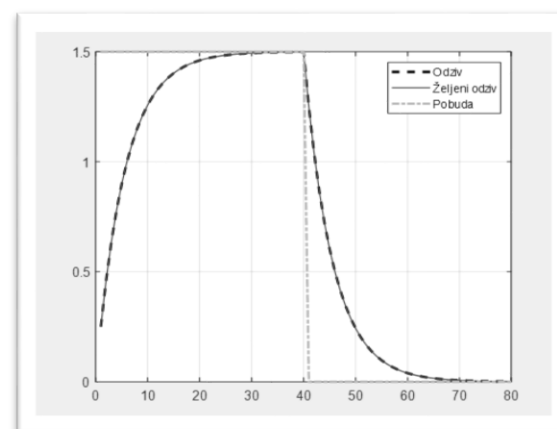
Test 1.0



Test 1.0



Test 1.5



Test 1.5

(a)

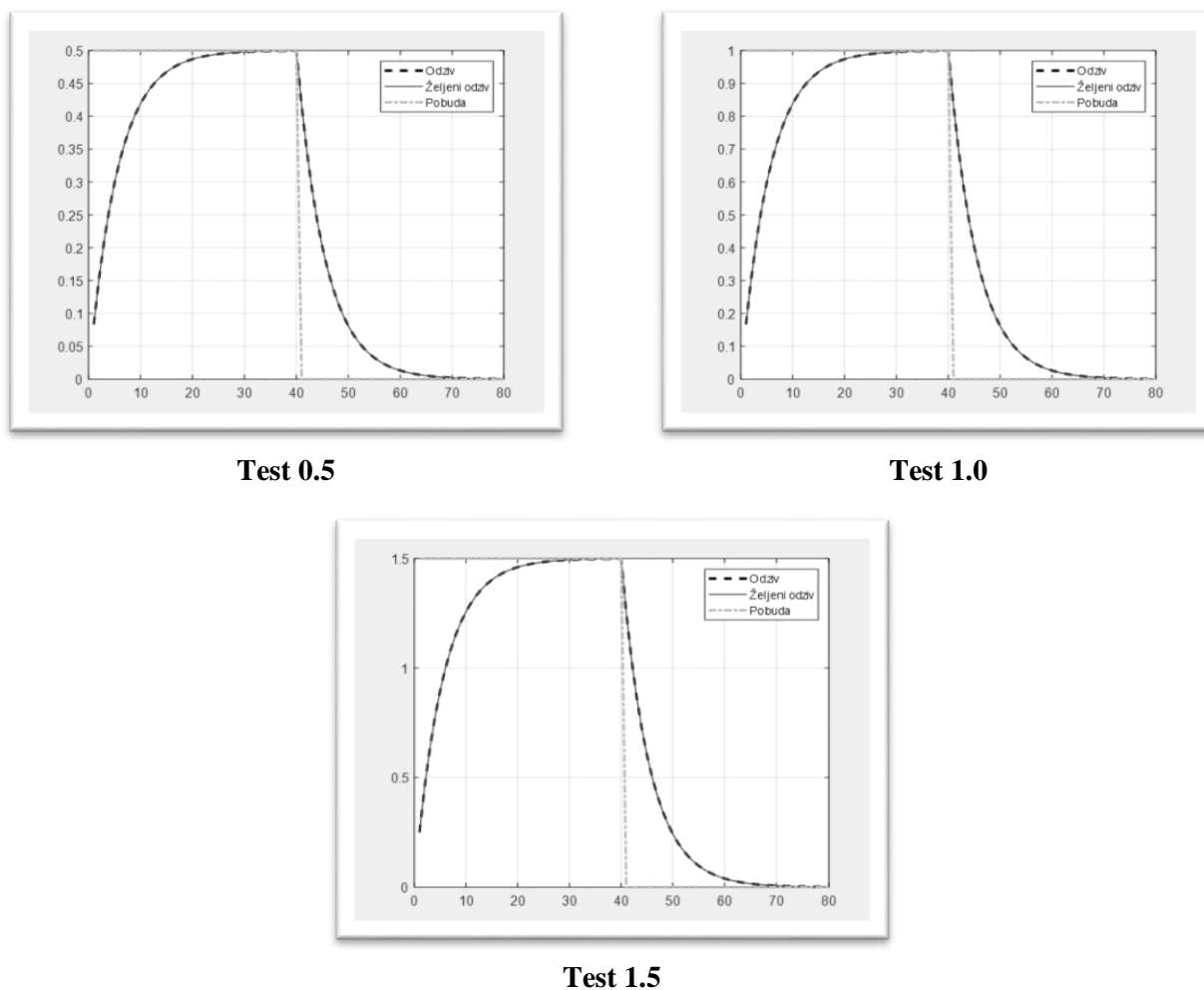
(b)

Slika 4.4. Odzivi mreže P1 člana na testnim primjerima pri konfiguraciji

a) 2-2-1

b) 2-5-1

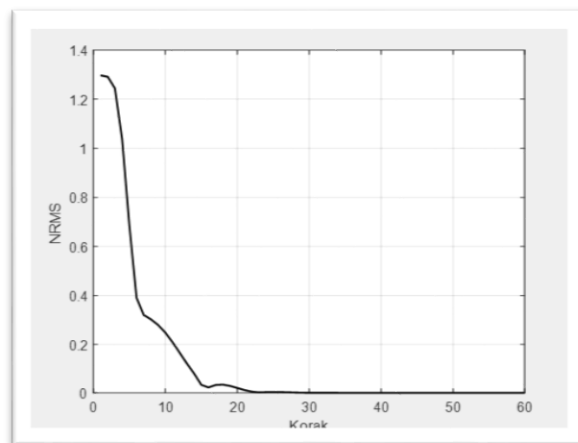
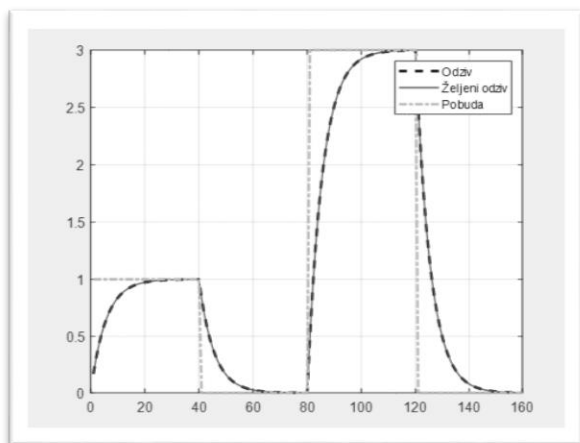
Iz slike 4.5. vidljivo je da daljnje povećanje broja neurona unutar skrivenog sloja ne utječe značajno na preciznost, budući da se ona već nalazi u idealnim granicama. Greška ove konfiguracije mreže ostaje izrazito niska (Tablica 1), s istim promjenama koje se javljaju tek na 5. decimali.



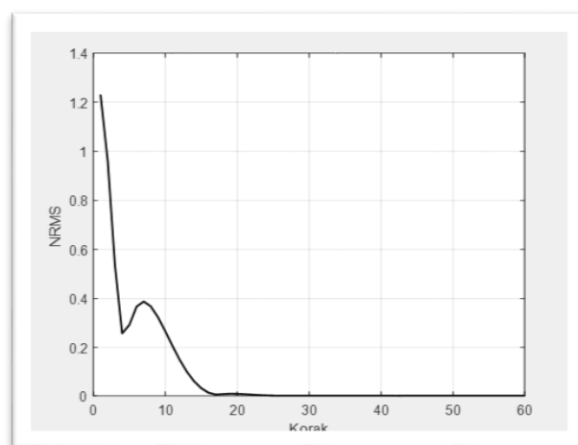
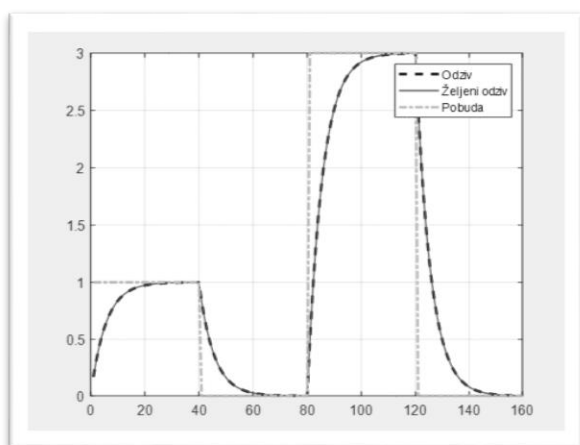
Slika 4.5. Odzivi mreže P1 člana na testnim primjerima pri konfiguraciji 2-10-1

4.1.2. ReLU s prilagodljivim koeficijentom nagiba

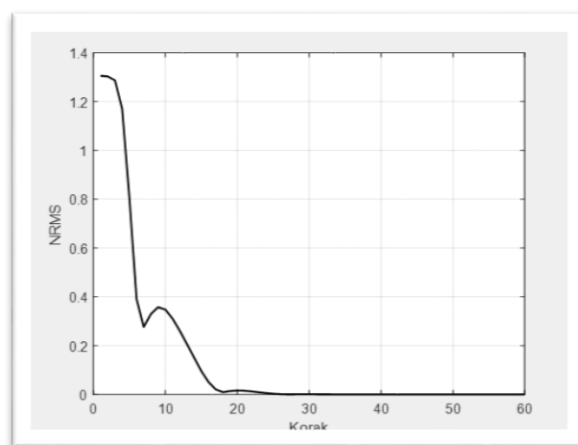
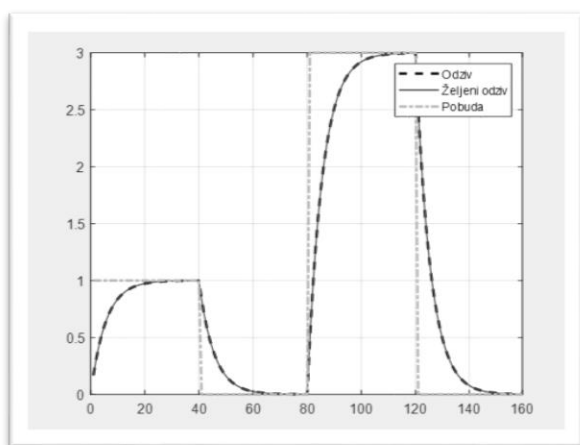
Promjenom aktivacijske funkcije, obične ReLU na ReLU s prilagodljivim koeficijentom nagiba, ne dolazi do ozbiljnijih odstupanja. Vrijednosti NRMS-a za sve tri strukture (prema slikama 4.6., 4.7. i 4.8. te Tablici 1) opet izrazito brzo padaju te već na 17. koraku postižu vrijednosti oko 1.77×10^{-5} .



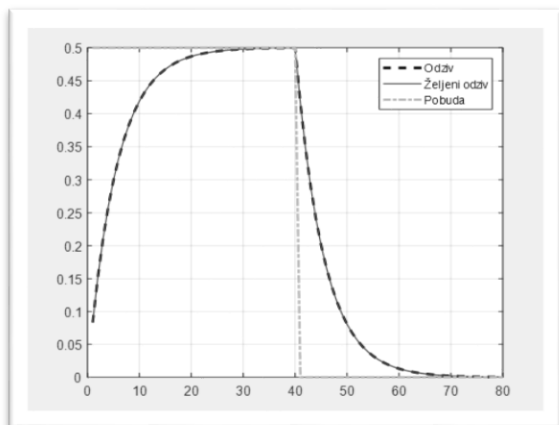
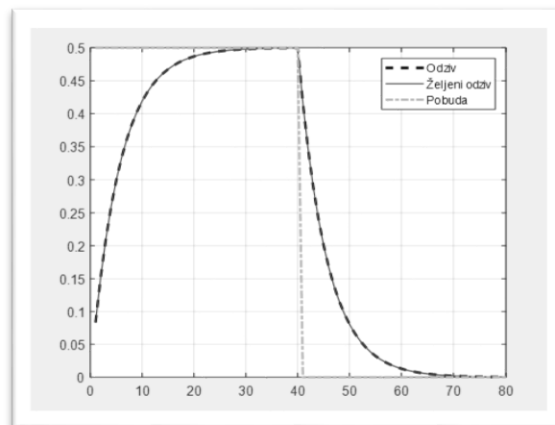
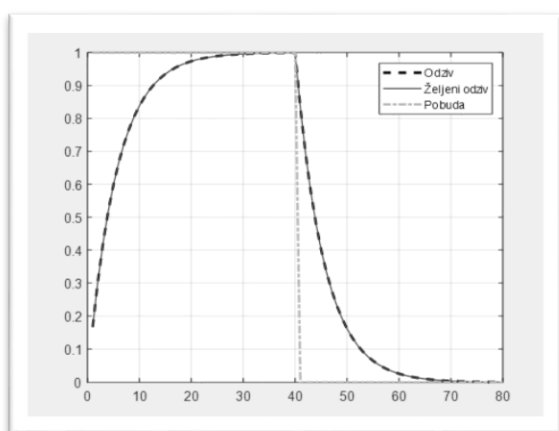
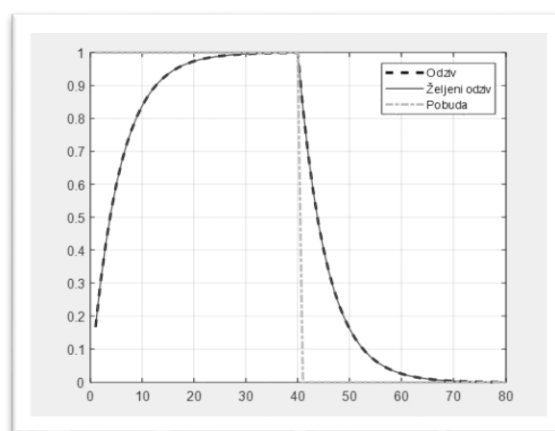
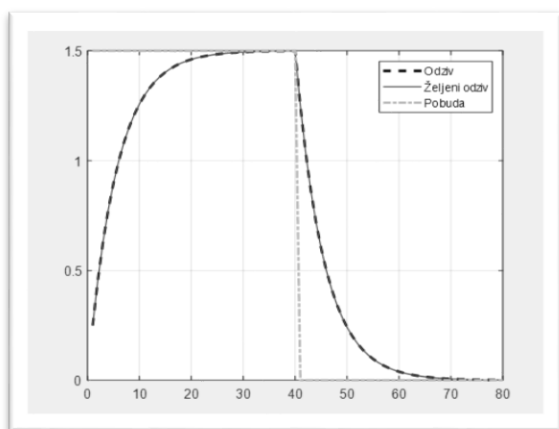
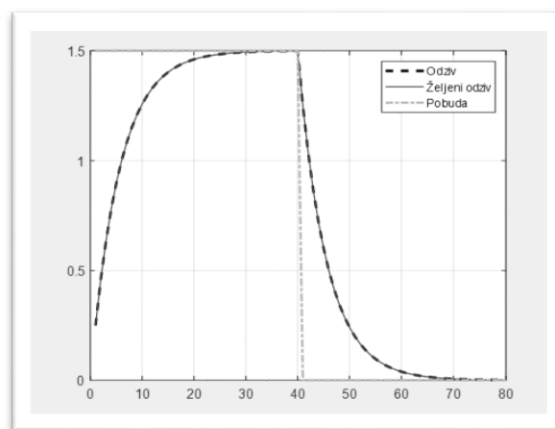
Slika 4.6. Odziv mreže nakon učenja skupa s NRMS parametrom u konfiguraciji 2-2-1



Slika 4.7. Odziv mreže nakon učenju skupa s NRMS parametrom u konfiguraciji 2-5-1



Slika 4.8. Odziv mreže nakon učenju skupa s NRMS parametrom u konfiguraciji 2-10-1

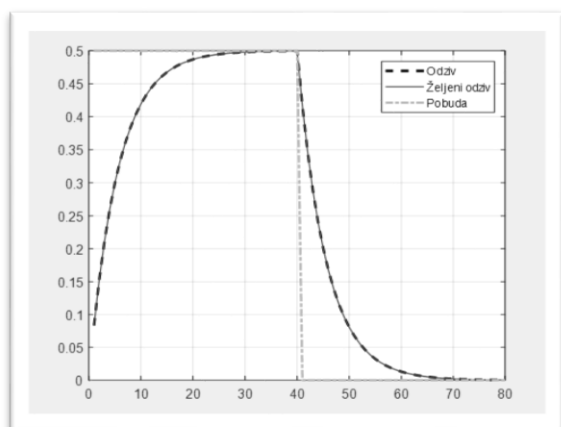
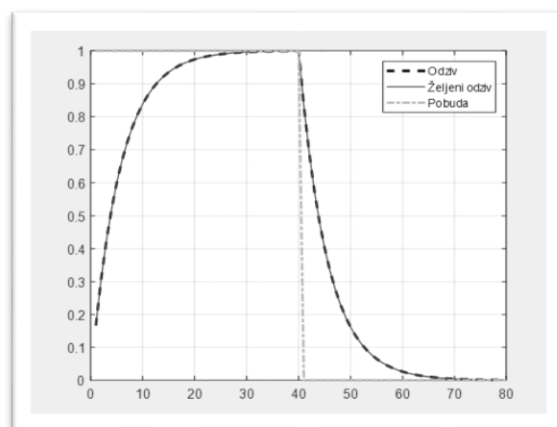
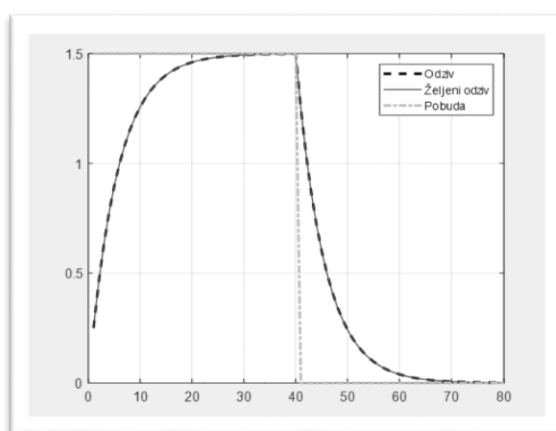
**Test 0.5****Test 0.5****Test 1.0****Test 1.0****Test 1.5****Test 1.5**

(a)

(b)

Slika 4.9. Odzivi mreže P1 člana na testnim primjerima pri konfiguraciji**a) 2-2-1****b) 2-5-1**

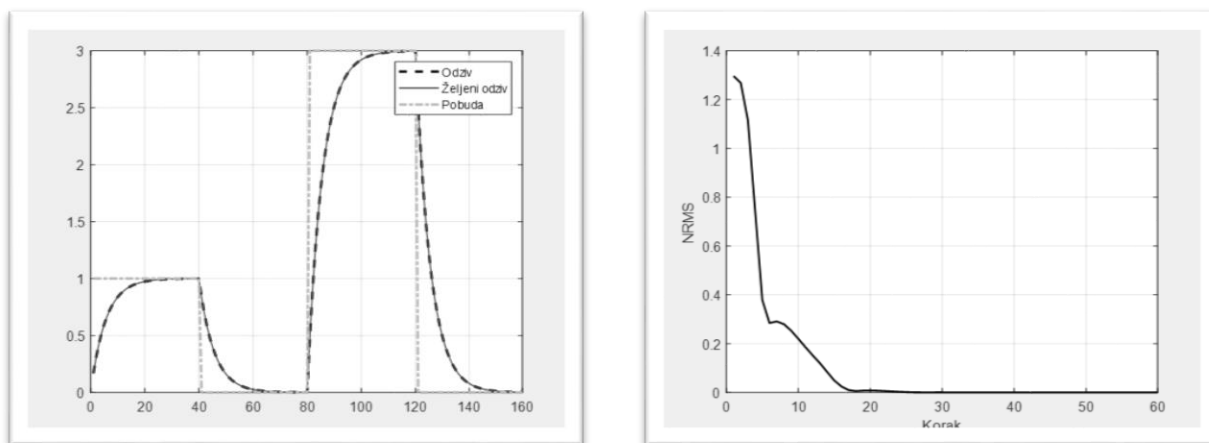
Kao što ozbiljnija odstupanja, u odnosu na prethodnu aktivacijsku funkciju, nisu vidljiva tijekom faze učenja, tako nisu vidljiva ni tijekom faze testiranja. Mreža i s ovom aktivacijskom funkcijom pruža izrazito dobra aproksimacijska i generalizacijska svojstva što se jasno da iščitati sa slika 4.9. i 4.10. Također ne može se uočiti niti pojave ni podtreniranosti ni pretreniranosti mreže, a vrijednosti NRMS-a se (prema Tablici 1) nalaze ponovo unutar pete decimalne.

**Test 0.5****Test 1.0****Test 1.5**

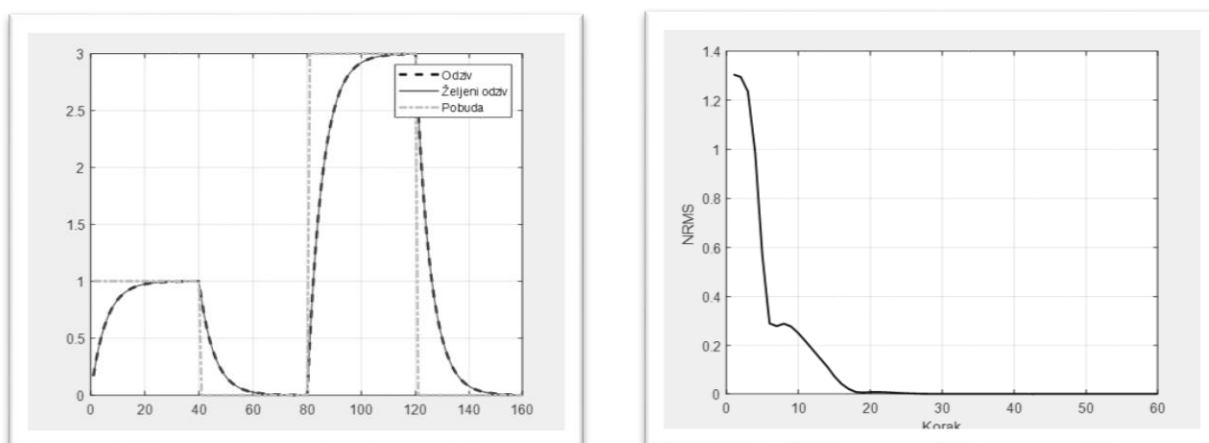
Slika 4.10. Odzivi mreže P1 člana na testnim primjerima pri konfiguraciji 2-10-1

4.1.3. ReLU s prilagodljivim pragom aktivacije

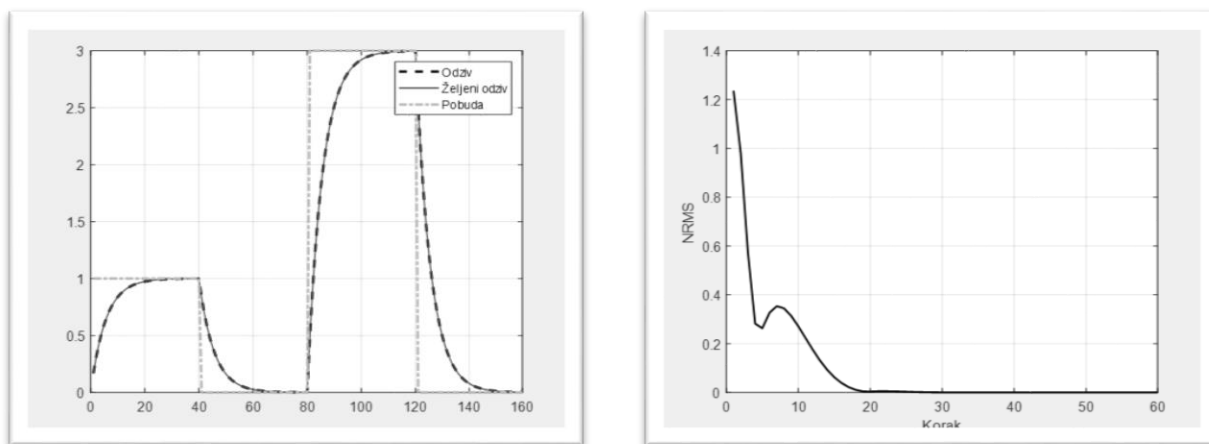
Uvođenjem ReLU aktivacijske funkcije s prilagodljivim pragom aktivacije, isto se ne događaju nikakve vidljive promjene u radu mreže. Aproksimacijska pogreška i dalje ostaje mala, dok vrijednosti NRMS-a ponovno konvergiraju u vrlo kratkom intervalu. Nakon faze učenja, NRMS za sve tri strukture doseže vrijednost od približno 1.78×10^{-5} , što je vidljivo iz Tablice 1.



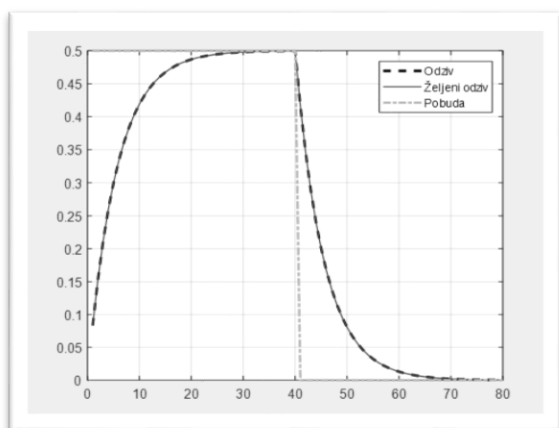
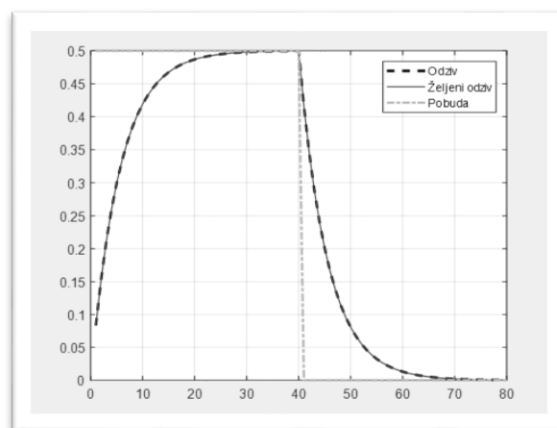
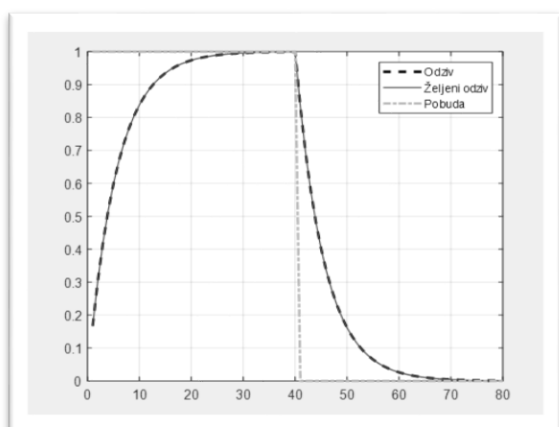
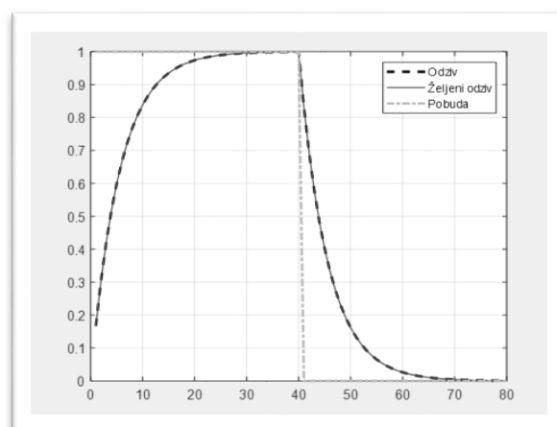
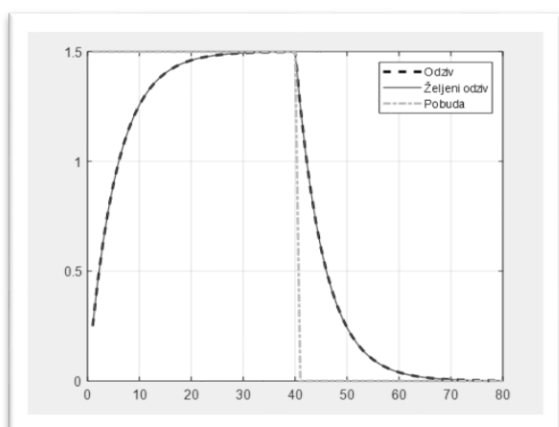
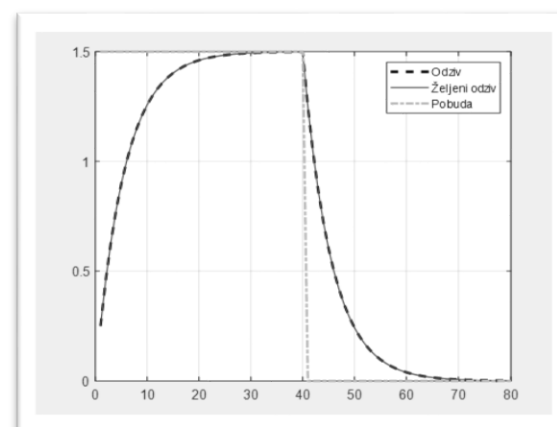
Slika 4.11. Odziv mreže nakon učenja skupa s NRMS parametrom u konfiguraciji 2-2-1



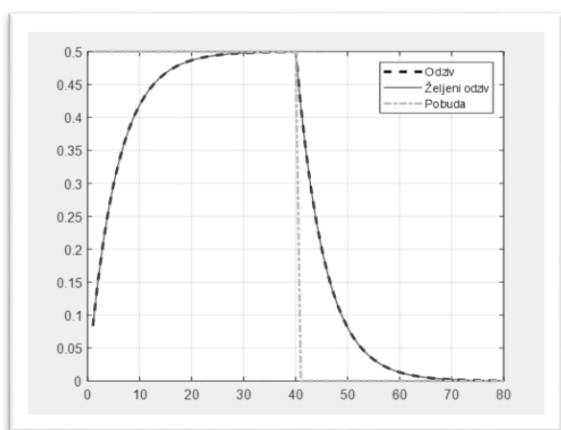
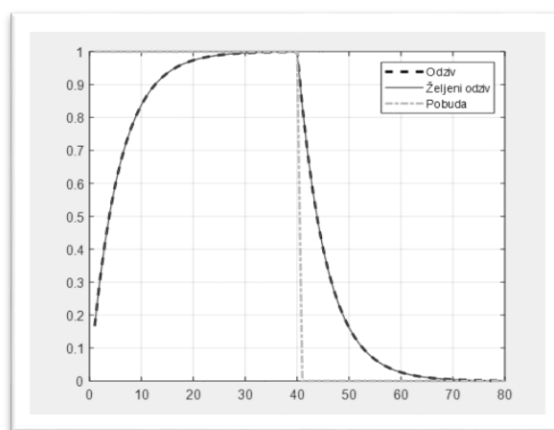
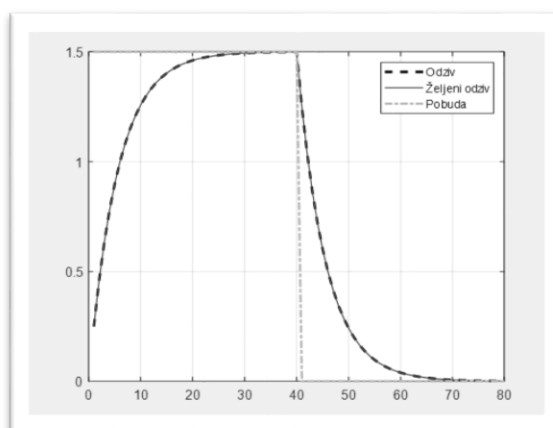
Slika 4.12. Odziv mreže nakon učenja skupa s NRMS parametrom u konfiguraciji 2-5-1



Slika 4.13. Odziv mreže nakon učenja skupa s NRMS parametrom u konfiguraciji 2-10-1

**Test 0.5****Test 0.5****Test 1.0****Test 1.0****Test 1.5****Test 1.5****(a)****(b)****Slika 4.14. Odzivi mreže P1 člana na testnim primjerima pri konfiguraciji****a) 2-2-1****b) 2-5-1**

Iz slika 4.14. i 4.15. te Tablice 1 također se ne mogu iščitati značajne razlike između željenog i stvarnog odziva. Vrijednosti NRMS-a ostaju iznimno niske, gdje se promjene dešavaju tek na 5. decimali. Aproximacijska i generalizacijska svojstva, sudeći prema grafovima, i dalje su odlična. Usporedbom rezultata s drugim aktivacijskim funkcijama ne uočavaju se značajni pomaci u preciznosti.

**Test 0.5****Test 1.0****Test 1.5**

Slika 4.15. Odzivi mreže P1 člana na testnim primjerima pri konfiguraciji 2-10-1

4.1.4. Tablični prikaz NRMS-a

Tablica 1 Krajnje vrijednosti NRMS-a za P1 član

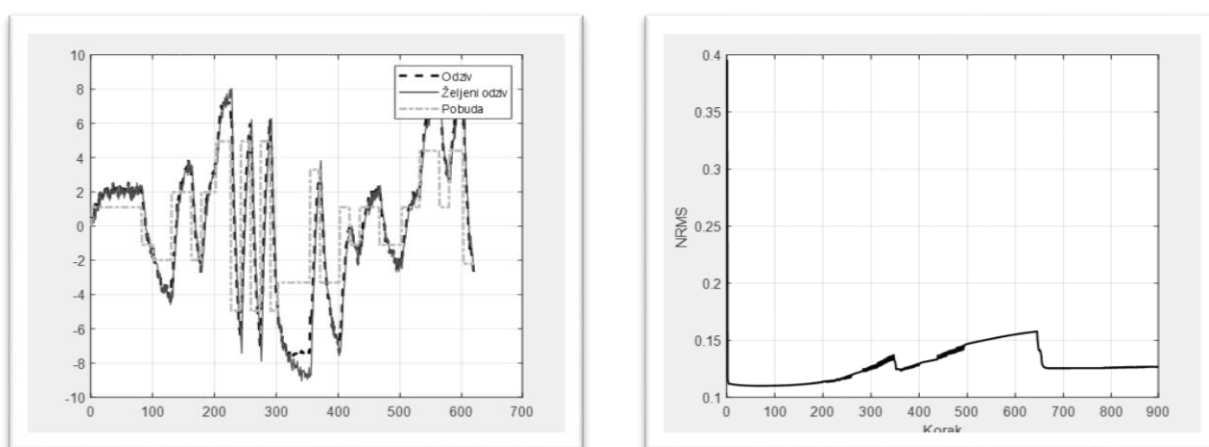
<i>Linearan sustav</i>		2-2-1	2-5-1	2-10-1
<i>ReLU aktivacijska funkcija</i>	UČENJE	1.7837x10 ⁻⁵	1.7810 x10 ⁻⁵	1.7749 x10 ⁻⁵
	Test 0.5	1.5155x10 ⁻⁵	1.5099 x10 ⁻⁵	1.5134 x10 ⁻⁵
	Test 1.0	8.1440x10 ⁻⁶	8.0375 x10 ⁻⁶	8.1699 x10 ⁻⁶
	Test 1.5	4.6083x10 ⁻⁵	4.6071 x10 ⁻⁵	4.6113x10 ⁻⁵
	Test 2.0	3.0643x10 ⁻⁵	3.0609 x10 ⁻⁵	3.0624 x10 ⁻⁵
	Test 2.5	2.5140x10 ⁻⁵	2.5115 x10 ⁻⁵	2.5160 x10 ⁻⁵
	Test 3.0	2.1054x10 ⁻⁵	2.1027 x10 ⁻⁵	2.0948 x10 ⁻⁵
<i>ReLU s prilagodljivim koeficijentom nagiba</i>	UČENJE	1.7778x10 ⁻⁵	1.7770 x10 ⁻⁵	1.9786 x10 ⁻⁵
	Test 0.5	1.5043x10 ⁻⁵	1.5013 x10 ⁻⁵	1.6034 x10 ⁻⁵
	Test 1.0	7.9139x10 ⁻⁶	7.8631 x10 ⁻⁶	9.8798 x10 ⁻⁶
	Test 1.5	4.6057x10 ⁻⁵	4.6052 x10 ⁻⁵	4.6459 x10 ⁻⁵
	Test 2.0	3.0547x10 ⁻⁵	3.0526 x10 ⁻⁵	3.1160 x10 ⁻⁵
	Test 2.5	2.5088x10 ⁻⁵	2.5078 x10 ⁻⁵	2.5745 x10 ⁻⁵
	Test 3.0	2.0993x10 ⁻⁵	2.0985 x10 ⁻⁵	2.3316 x10 ⁻⁵
<i>ReLU s prilagodljivim pragom aktivacije</i>	UČENJE	1.7832x10 ⁻⁵	1.7826 x10 ⁻⁵	1.7811 x10 ⁻⁵
	Test 0.5	1.5142 x10 ⁻⁵	1.5130 x10 ⁻⁵	2.3430 x10 ⁻³
	Test 1.0	8.1236 x10 ⁻⁶	8.0999 x10 ⁻⁶	8.0336 x10 ⁻⁶
	Test 1.5	4.6081 x10 ⁻⁵	4.6078 x10 ⁻⁵	4.6071 x10 ⁻⁵
	Test 2.0	3.0635 x10 ⁻⁵	3.0626 x10 ⁻⁵	3.0600 x10 ⁻⁵
	Test 2.5	2.5136 x10 ⁻⁵	2.51291 x10 ⁻⁵	2.5115 x10 ⁻⁵
	Test 3.0	2.1049 x10 ⁻⁵	2.1043 x10 ⁻⁵	2.1028 x10 ⁻⁵

4.2. Rezultati identifikacije nelinearnog sustava

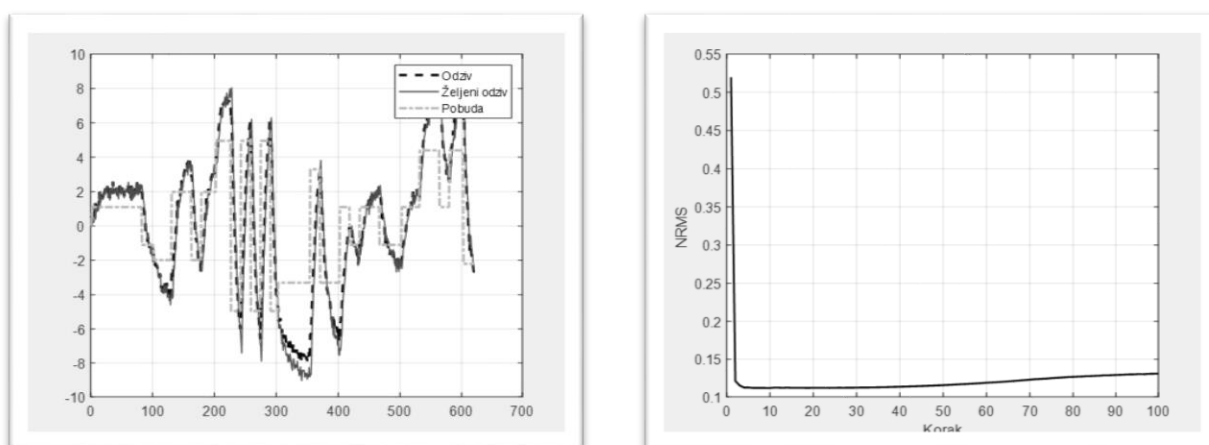
Učenje i testiranje naučene mreže provodi se pomoću APRBS signala opisanog u prethodnom poglavlju. Na grafu je taj signal prikazan crta-točka-crta linijom, dok je željeni odziv prikazan punom linijom. Signali su diskretizirani na 600 točaka, osim u fazi testiranja, gdje su generirani unutar raspona od 400 točaka. Odziv mreže na grafovima prikazan je isprekidanom linijom. Također parametri mreže tijekom učenja u ovom testu ostaju isti kao u prethodnoj cjelini “Rezultati identifikacije linearnog sustava“. Krajnje vrijednosti NRMS-a su pohranjene unutar Tablice 2.

4.2.1. ReLU aktivacijska funkcija

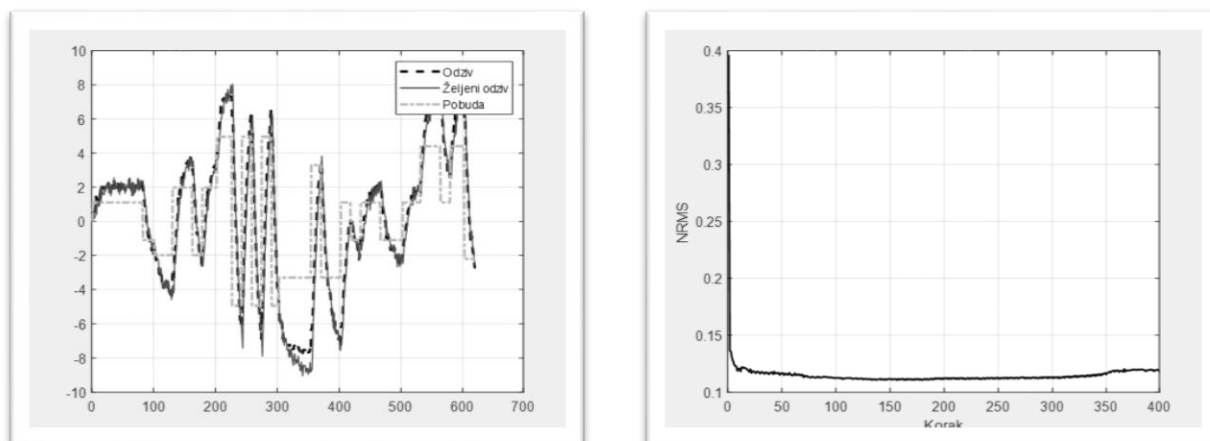
Na slikama 4.16, 4.17 i 4.18, za razliku od linearnog sustava, uočena su veća odstupanja između stvarnog i željenog odziva mreže pri primjeni standardne ReLU funkcije. U sva tri slučaja, promjena broja neurona u skrivenom sloju, ne uzrokuje značajnu razliku u aproksimaciji željenih vrijednosti. Sama odstupanja su najizraženija u ekstremima, odnosno pri nagloj promjeni predznaka signala. U tim trenucima, unutar svake strukture dolazi do lošeg i sporog praćenja, što se očituje kao kašnjenje signala. Vrijednosti NRMS-a (prema Tablici 2) također se ustaljuju na “visokim” vrijednostima između 0.12 i 0.13. gdje se na grafu 4.16., pri strukturi s dva neurona unutar sakrivenog sloja, mogu uočiti i određene nepravilnosti.



Slika 4.16. Odziv mreže nakon učenja skupa s NRMS parametrom u konfiguraciji 2-2-1

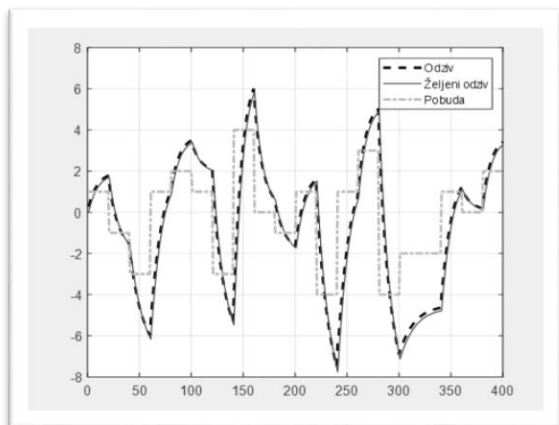


Slika 4.17. Odziv mreže nakon učenja skupa s NRMS parametrom u konfiguraciji 2-5-1

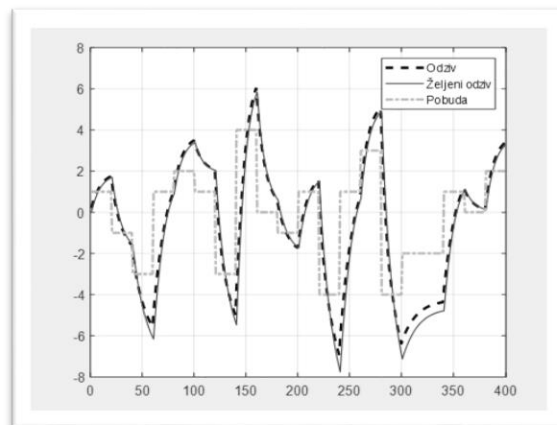


Slika 4.18. Odziv mreže nakon učenja skupa s NRMS parametrom u konfiguraciji 2-10-1

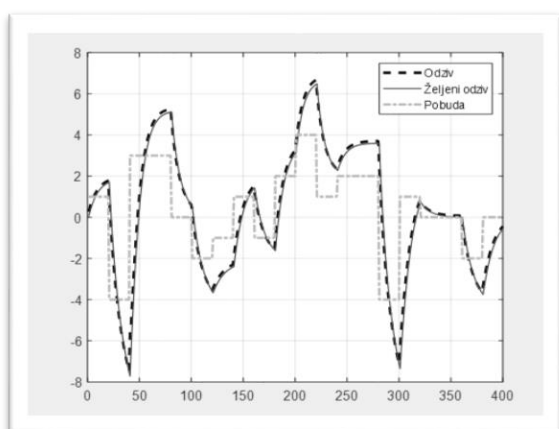
Na slikama 4.19. i 4.20. usporedbom tri različite strukture, uočavaju se zanimljiva odstupanja. Mreže s dva i deset neurona u skrivenom sloju pokazuju relativno dobra generalizacijska svojstva, dok mreža s pet neurona pokazuje značajnije razlike u ekstremnim vrijednostima. To rezultira vidljivim odstupanjima između željenog i stvarnog odziva pri čemu su, za tu strukturu, vrijednosti NRMS-a neznatno veće (Tablica 2). Povećanje broja neurona također ne dovodi do pretreniranosti mreže.



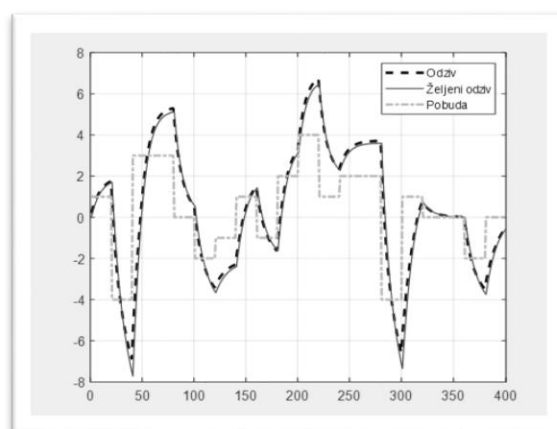
Test 1



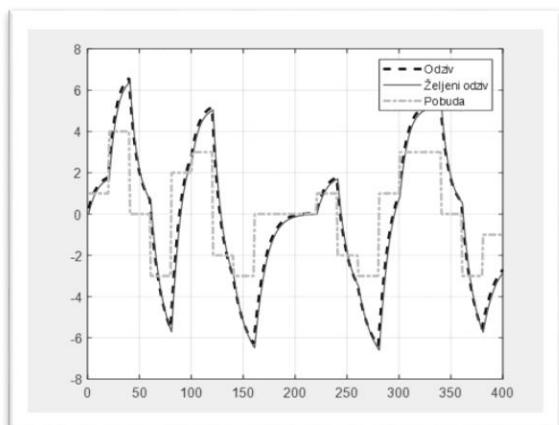
Test 1



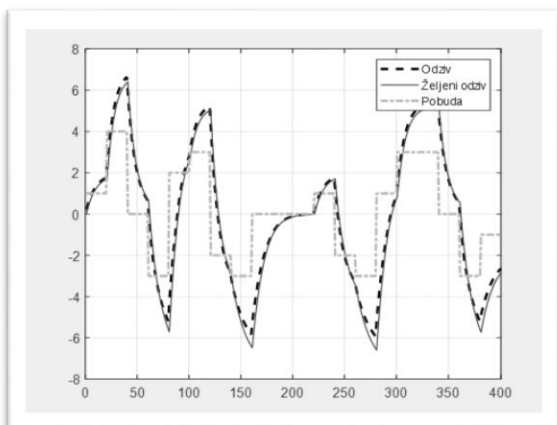
Test 2



Test 2



Test 3



Test 3

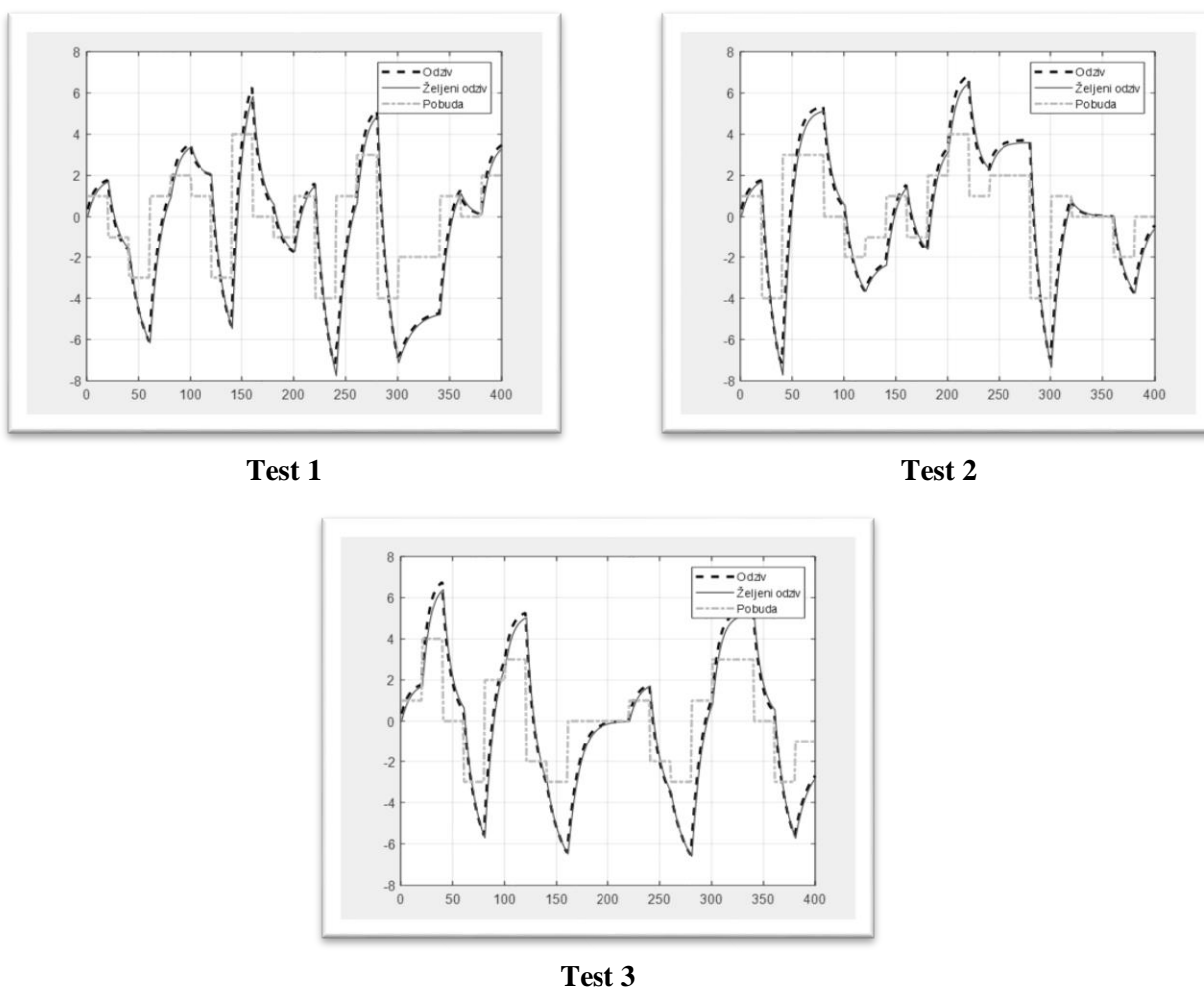
(a)

(b)

Slika 4.19. Odzivi mreže nelinearnog sustava na testnim primjerima pri konfiguraciji

a) 2-2-1

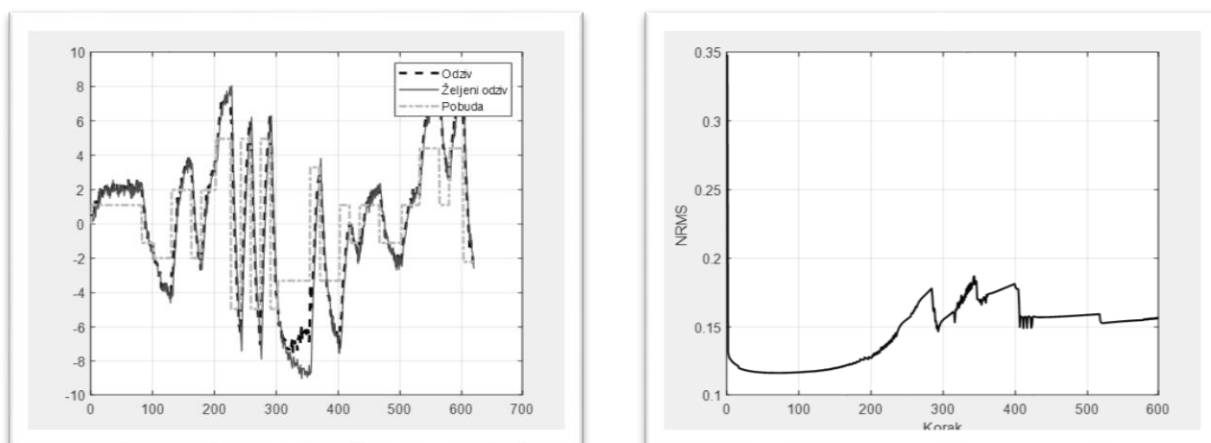
b) 2-5-1



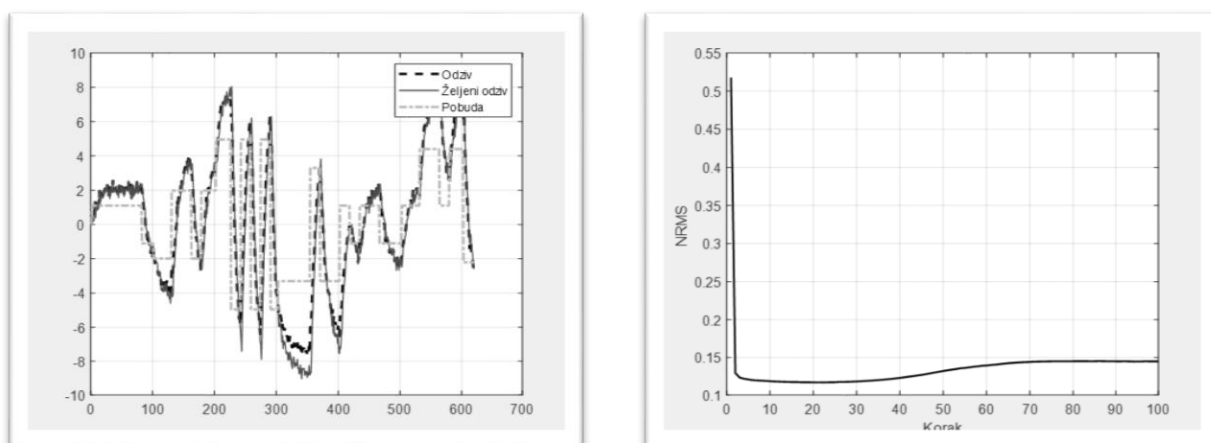
Slika 4.20. Odzivi mreže nelinearnog sustava na testnim primjerima pri konfiguraciji 2-10-1

4.2.2. *ReLU s prilagodljivim koeficijentom nagiba*

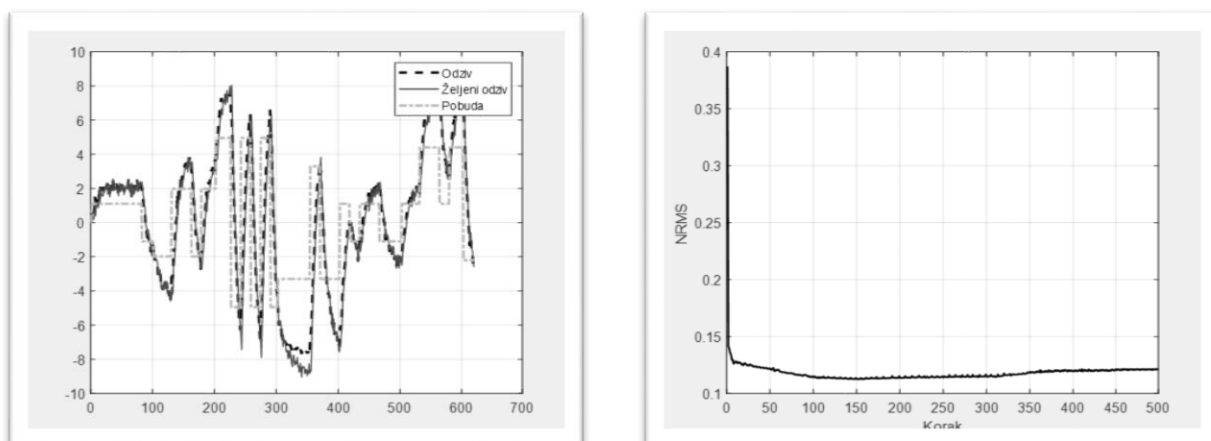
Promjenom aktivacijske funkcije na ReLU s prilagodljivim koeficijentom nagiba možemo uočiti nove zanimljive pojave. Na slikama 4.21., 4.22. i 4.23. vidljiva su odstupanja vrijednosti na kojima konvergiraju pojedine strukture mreže. Kod mreže s 2 neurona (koja je ujedno i najlošija od prikazanih) primjećuju se skokovi tijekom koraka učenja, što ukazuje na poteškoće u adaptaciji parametara. Također se može primijetiti da ReLU funkcija s prilagodljivim koeficijentom nagiba znatno lošije prati nelinearni sustav, u usporedbi s običnom ReLU aktivacijskom funkcijom. Pri učenju najbolji rezultat ove aktivacijske funkcije postiže mreža s 10 neurona, gdje NRMS vrijednost konvergira na 0.13.



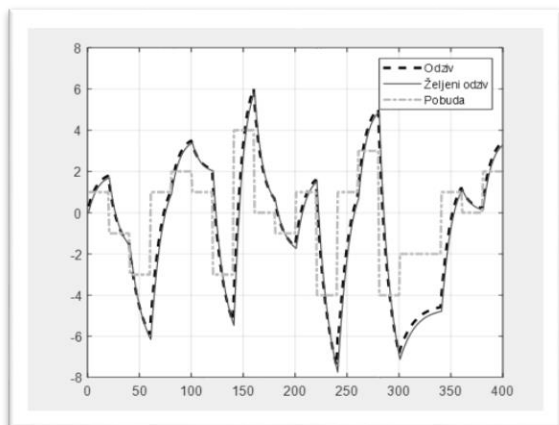
Slika 4.21. Odziv mreže nakon učenja skupa s NRMS parametrom u konfiguraciji 2-2-1



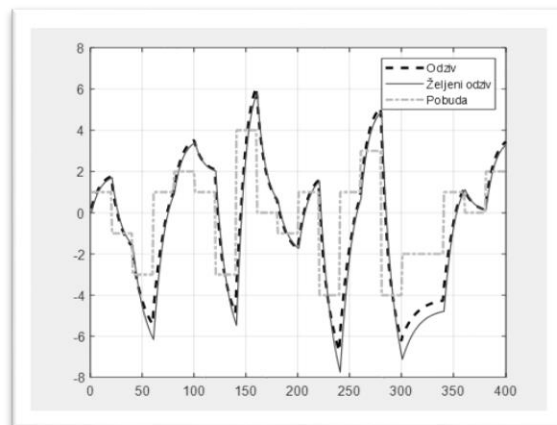
Slika 4.22. Odziv mreže nakon učenja skupa s NRMS parametrom u konfiguraciji 2-5-1



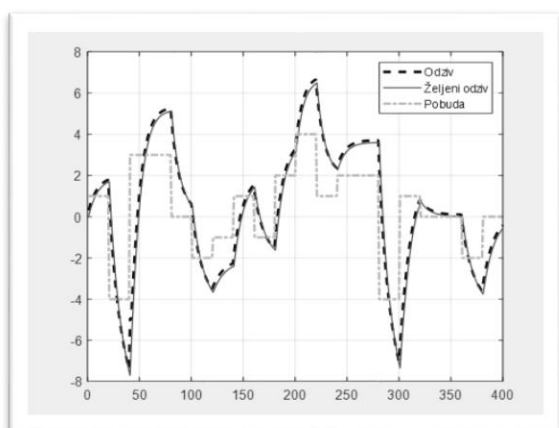
Slika 4.23. Odziv mreže nakon učenja skupa s NRMS parametrom u konfiguraciji 2-10-1



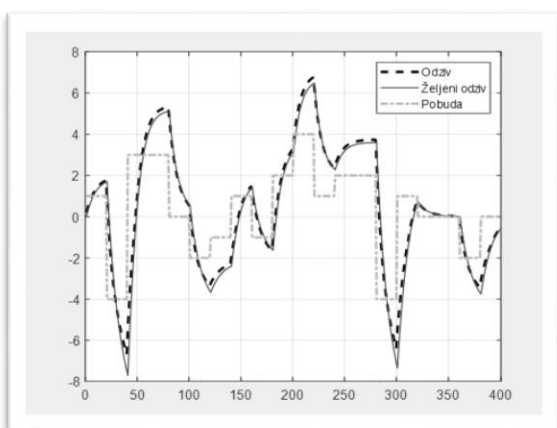
Test 1



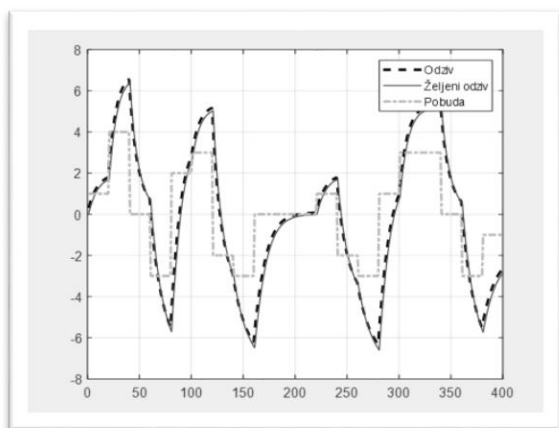
Test 1



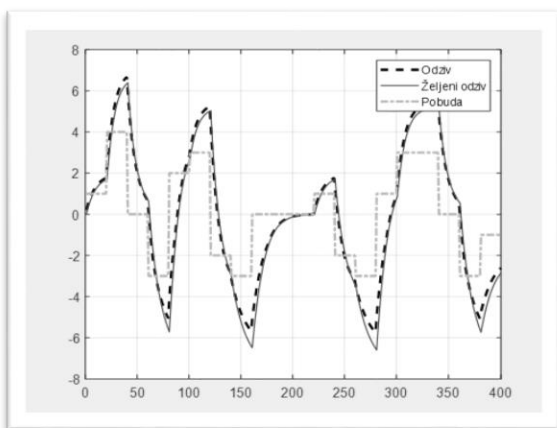
Test 2



Test 2



Test 3

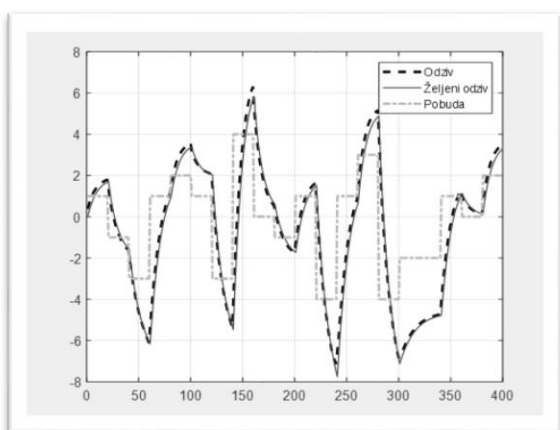


Test 3

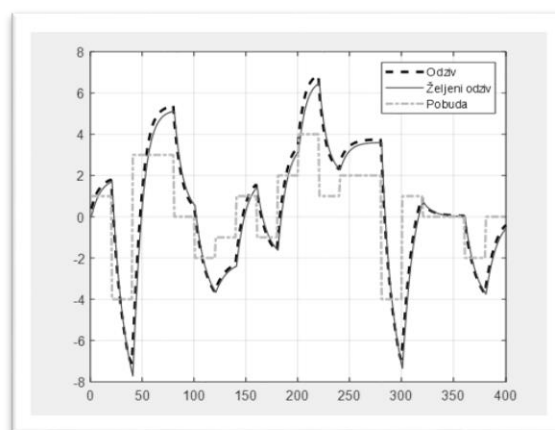
(a)

(b)

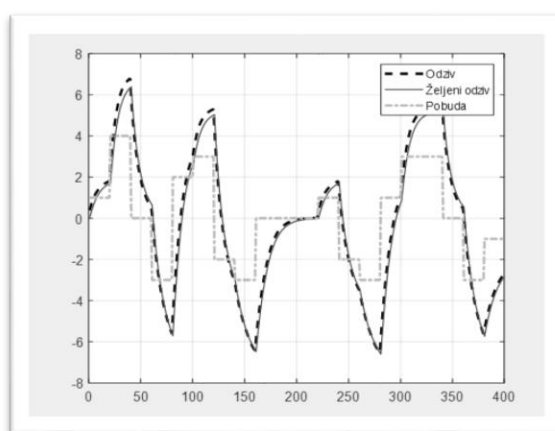
Slika 4.24. Odzivi mreže nelinearnog sustava na testnim primjerima pri konfiguraciji
 a) 2-2-1
 b) 2-5-1



Test 1



Test 2



Test 3

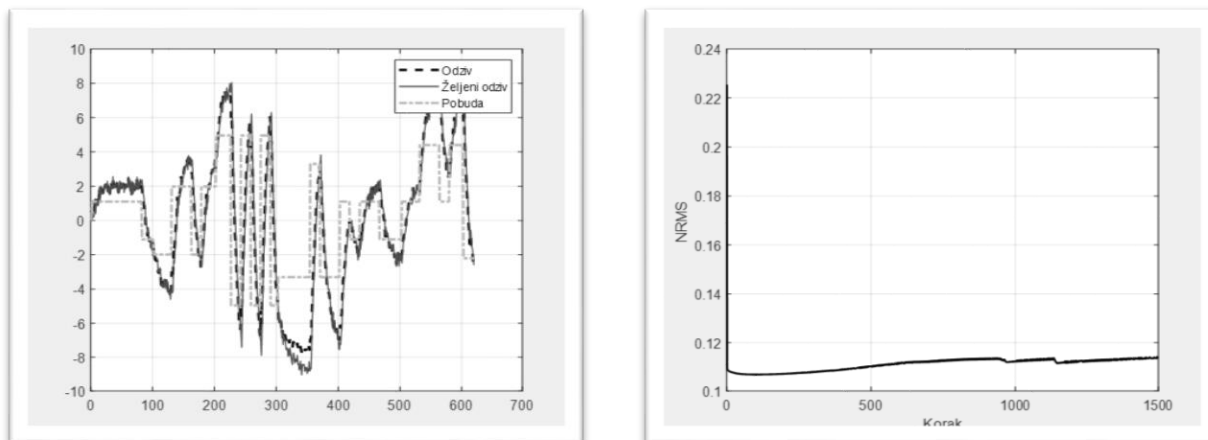
Slika 4.25. Odzivi mreže nelinearnog sustava na testnim primjerima pri konfiguraciji 2-10-1

Pri analizi testova sa slika 4.24. i 4.25. može se uočiti da se svojstva lošije aproksimacije ne prenose i na svojstva generalizacije. Provlačenjem novih signala kroz mrežu uočava se da izlazi mreže s 2 i 10 neurona dobro prate željene odzive sustava, dok se za mrežu s 5 neurona to ne može jasno zaključiti. Pažljivijim promatranjem i usporedbom s prethodnom funkcijom, jasno je da i ova funkcija dijeli slične karakteristike kao i “klasična ReLU”.

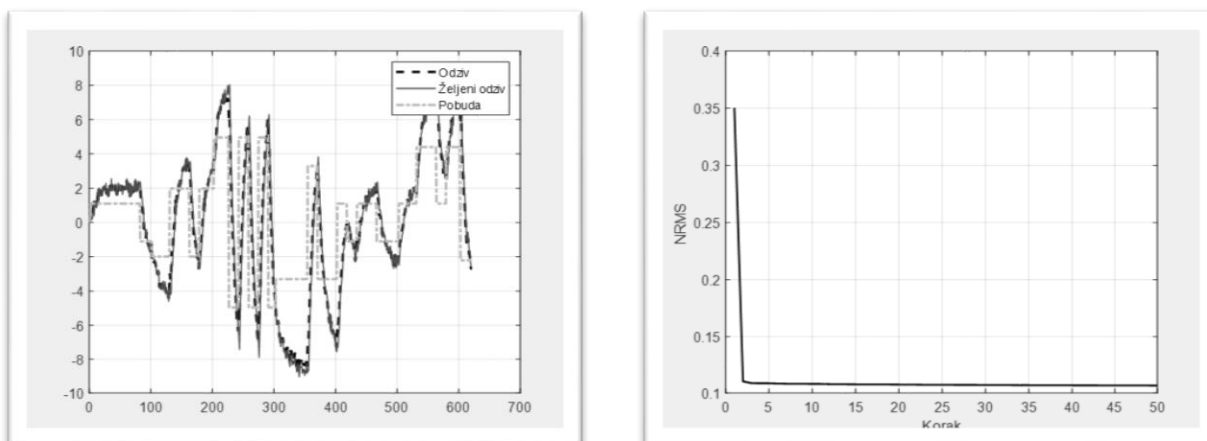
4.2.3. *ReLU s prilagodljivim pragom aktivacije*

Za razliku od prethodnih funkcija, uvođenjem aktivacijske funkcije pod nazivom ReLU s prilagodljivim pragom aktivacije, sa slika 4.26., 4.27. i 4.28. vidljivo je da mreža u ovoj konfiguraciji daje najbolje rezultate. Pri strukturi s pet i deset neurona NRMS mreže pada ispod prethodno dobivenih 0.11. Također ovdje možemo uočiti i očekivani manjak kapaciteta u konfiguraciji mreže s dva neurona skrivenog sloja. Kao i u prethodne dvije funkcije, najveća

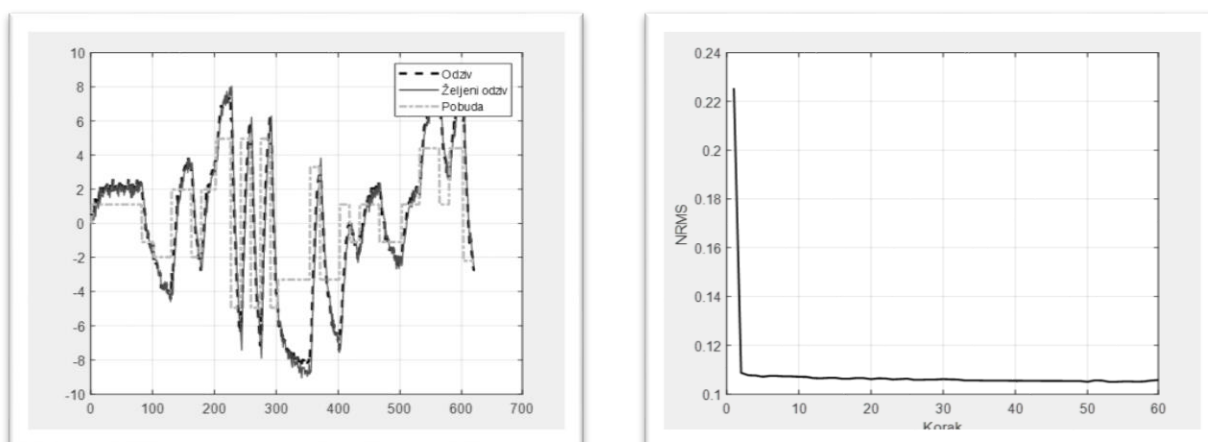
odstupanja se događaju pri pojavi ekstremnih skokova pobude. Pri analizi grafa NRMS-a za sve tri strukture možemo još uočiti i pojavu blage anomalije u strukturi s dva neurona skrivenog sloja.



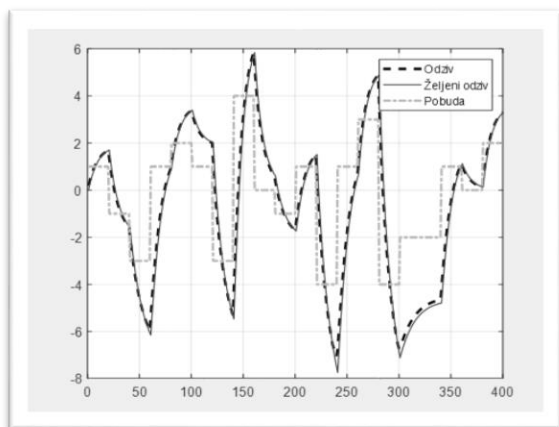
Slika 4.26. Odziv mreže nakon učenja skupa s NRMS parametrom u konfiguraciji 2-2-1



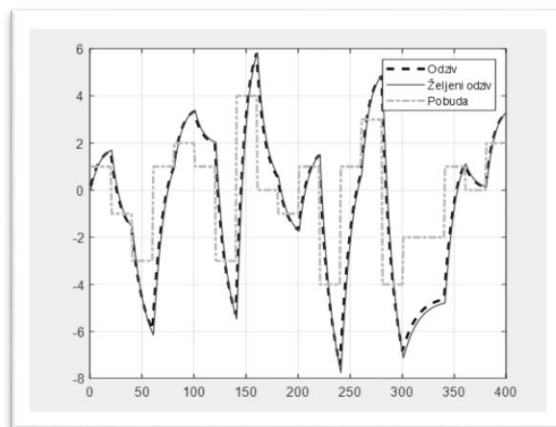
Slika 4.27. Odziv mreže nakon učenja skupa s NRMS parametrom u konfiguraciji 2-5-1



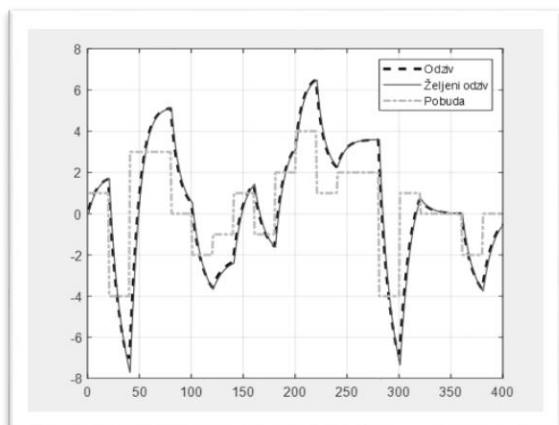
Slika 4.28. Odziv mreže nakon učenja skupa s NRMS parametrom u konfiguraciji 2-10-1



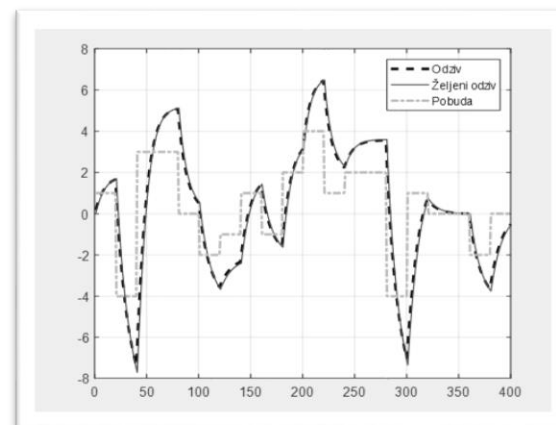
Test 1



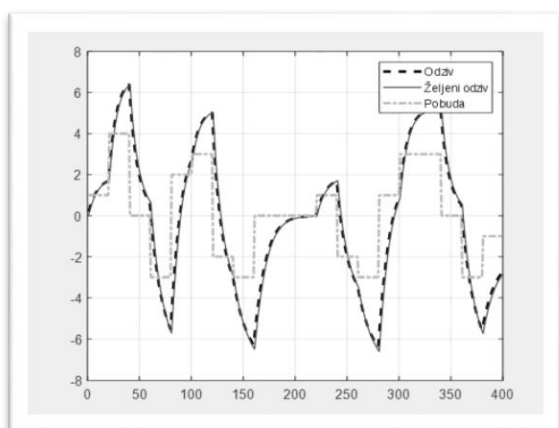
Test 1



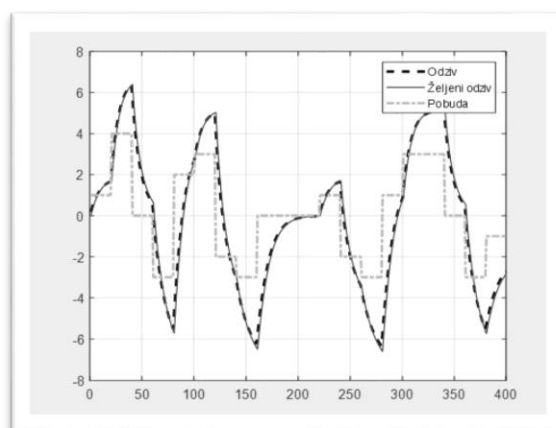
Test 2



Test 2



Test 3



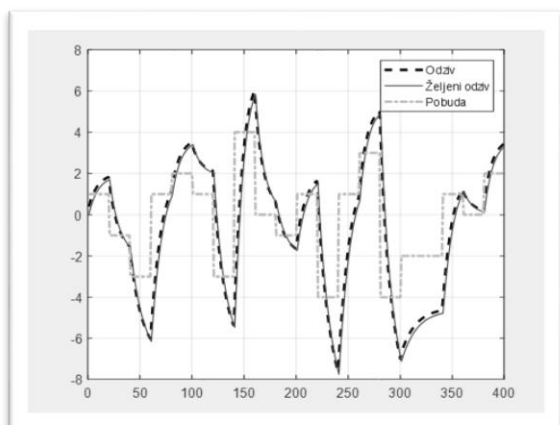
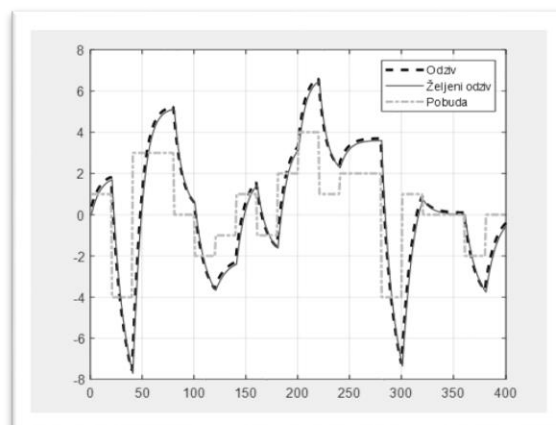
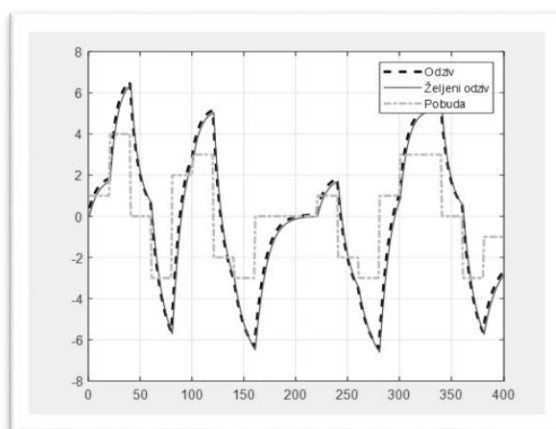
Test 3

(a)

(b)

Slika 4.29. Odzivi mreže nelinearnog sustava na testnim primjerima pri konfiguraciji
a) 2-2-1
b) 2-5-1

Iz slika 4.29. i 4.30. te pogledom u Tablicu 2 vidljivo je da ovaj oblik funkcije nastavlja pružati izvrsna svojstva generalizacije. Štoviše, usporedbom odziva mreže uočavamo da sve tri strukture vrlo precizno prate željeni odziv, što potvrđuju i NRMS vrijednosti koja iznosi oko 0.3 za sva tri testa.

**Test 1****Test 2****Test 3**

Slika 4.30. Odzivi mreže nelinearnog sustava na testnim primjerima pri konfiguraciji 2-10-1

4.2.4. Tablični prikaz NRMS-a

Tablica 2 Krajnje vrijednosti NRMS-a za nelinearan sustav

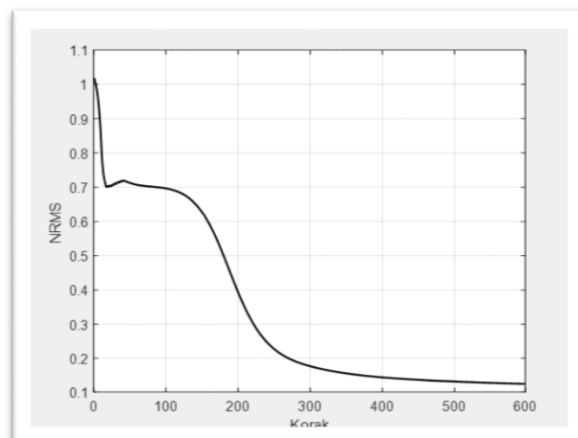
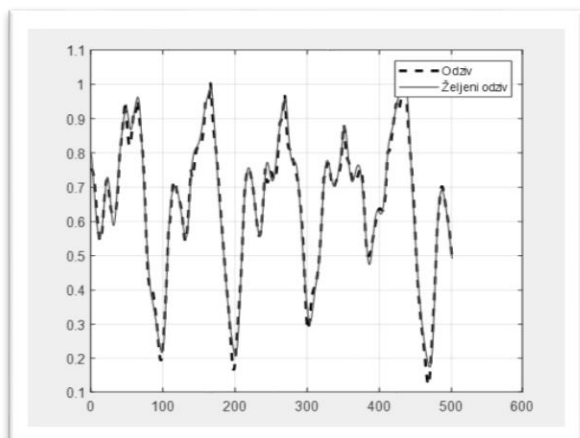
<i>Nelinearan sustav</i>		2-2-1	2-5-1	2-10-1
<i>ReLU aktivacijska funkcija</i>	UČENJE	0.12683	0.13016	0.12257
	Test 1	0.03640	0.08319	0.05499
	Test 2	0.03724	0.06406	0.05819
	Test 3	0.03689	0.07080	0.05195
<i>ReLU s prilagodljivim koeficijentom nagiba</i>	UČENJE	0.16317	0.14429	0.12480
	Test 1	0.04977	0.10411	0.05710
	Test 2	0.04968	0.07937	0.06191
	Test 3	0.04480	0.08695	0.05605
<i>ReLU s prilagodljivim pragom aktivacije</i>	UČENJE	0.11847	0.10685	0.10569
	Test 1	0.03186	0.03228	0.04248
	Test 2	0.02760	0.02550	0.03985
	Test 3	0.02846	0.03182	0.04098

4.3. Rezultati predviđanja kaotičnog sustava

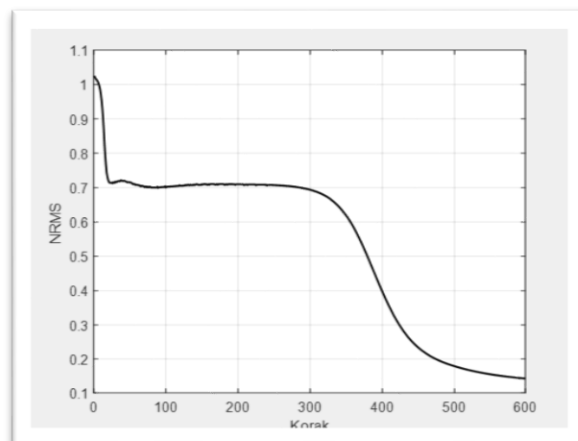
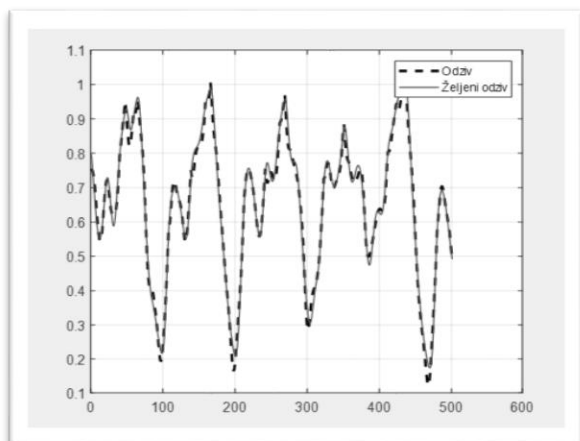
Kako bi se mreža testirala na nelinearan kaotični sustav, generirano je 500 novih točaka korištenjem Mackey-Glass vremenske serije. Mreža je trenirana na već generiranom skupu pri čemu je željeni odziv sustava prikazan na slici 3.3. Također koeficijent brzine učenja smanjen je za jedan red veličine kako bi se povećala preciznost te sada iznosi 0.01. Vrijednost momenta, skupa s ostalim parametrima, ostaje ista kao i za prethodne sustave.

4.3.1. ReLU aktivacijska funkcija

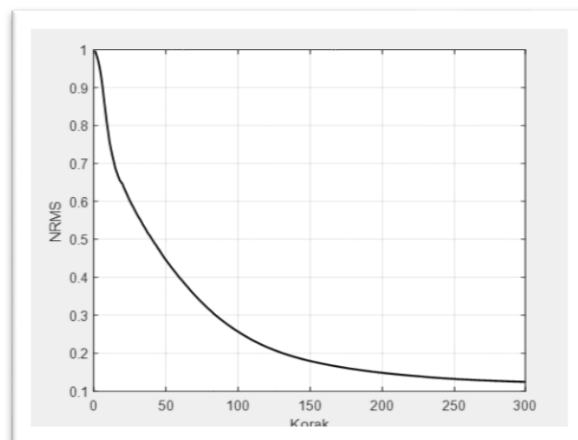
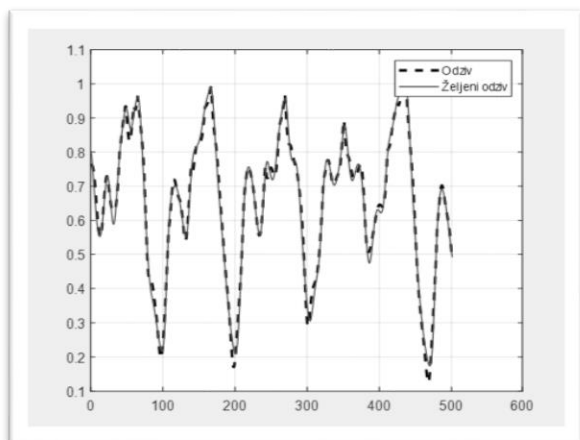
Sa slika 4.31. 4.32. i 4.33. vidljivo je da obična ReLU aktivacijska funkcija prati vremensku seriju s određenom greškom koja se posebno ističe pri pojavi ekstremiteta. Također korištenjem ovakvog tipa testa može se uočiti potreba za većim brojem koraka prije stagnacije vrijednosti NRMS-a za sve tri strukture mreže, osim za strukturu s 10 neurona koja konvergira u nešto manjem broju koraka. Vrijednosti NRMS-a svih triju struktura kreću se oko 0.11, što se da iščitati iz Tablice 3.



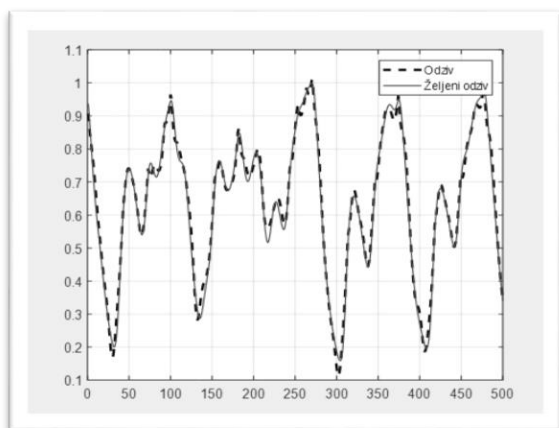
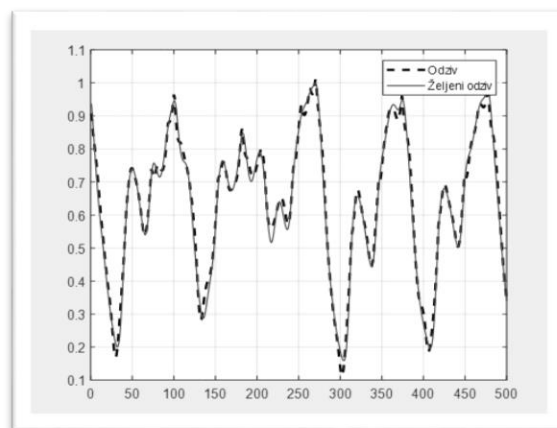
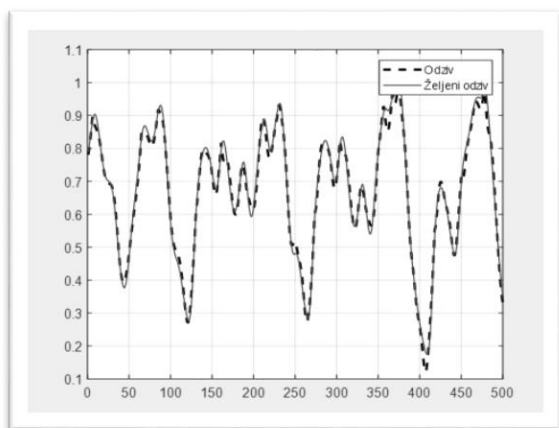
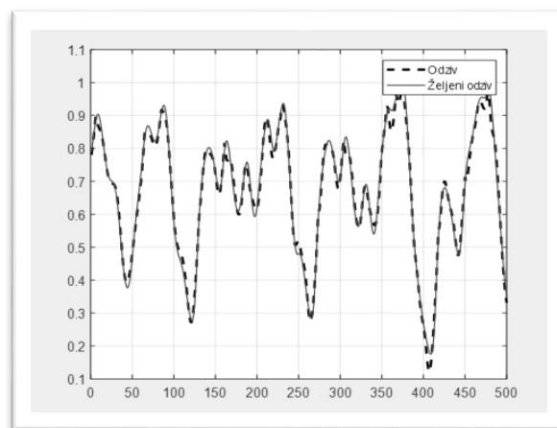
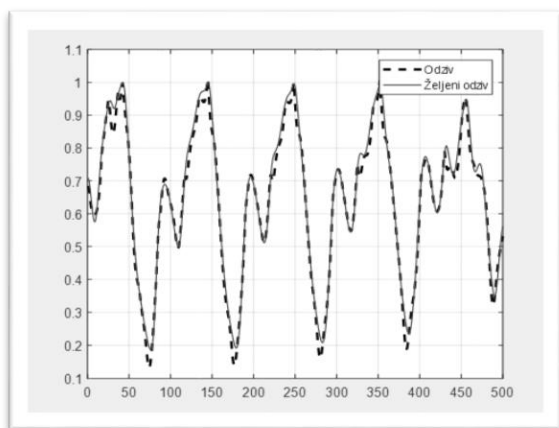
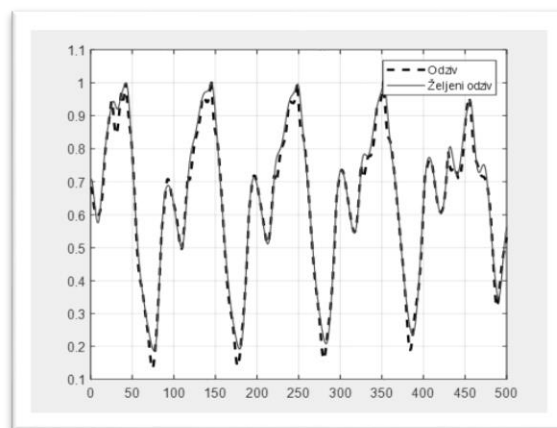
Slika 4.31. Odziv mreže nakon učenja skupa s NRMS parametrom u konfiguraciji 2-2-1



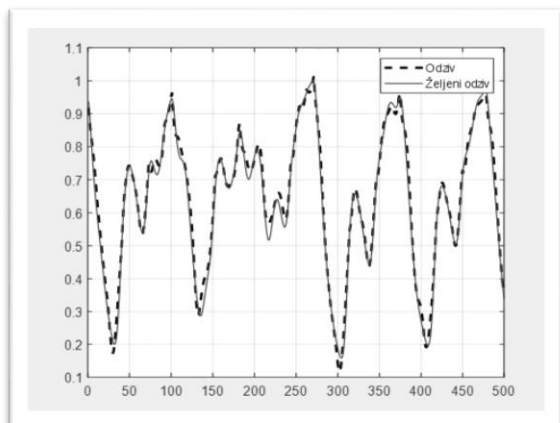
Slika 4.32. Odziv mreže nakon učenja skupa s NRMS parametrom u konfiguraciji 2-5-1



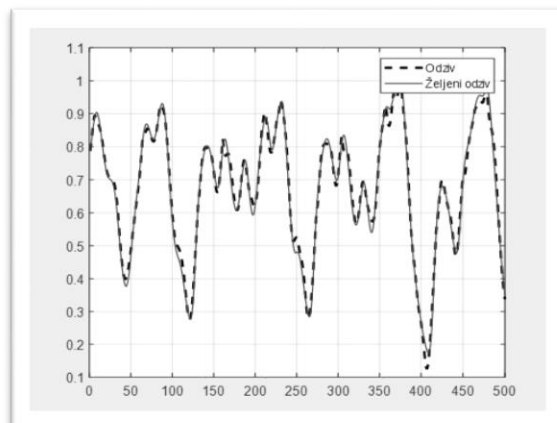
Slika 4.33. Odziv mreže nakon učenja skupa s NRMS parametrom u konfiguraciji 2-10-1

**Test 1****Test 1****Test 2****Test 2****Test 3****(a)****Test 3****(b)**

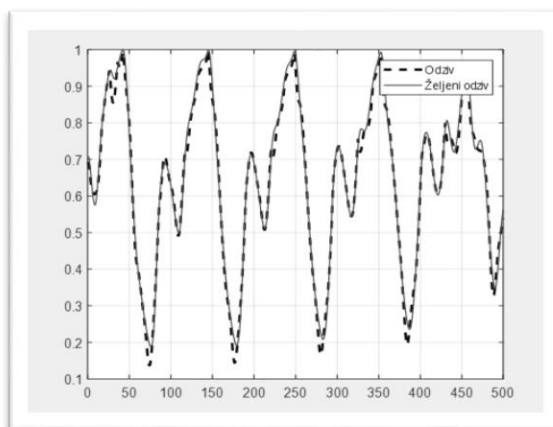
Slika 4.34. Odzivi mreže kaotičnih vremenskih serija na testnim primjerima pri konfiguraciji
a) 2-2-1
b) 2-5-1



Test 1



Test 2



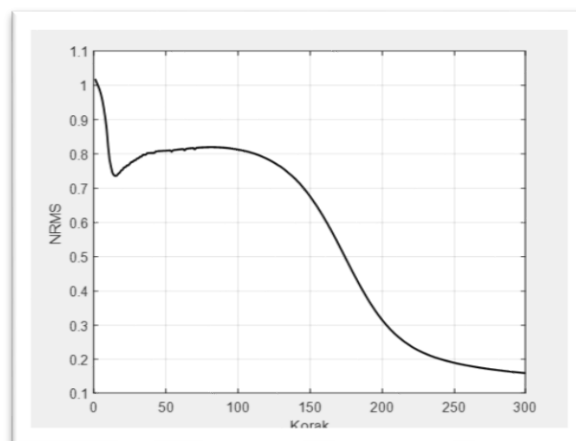
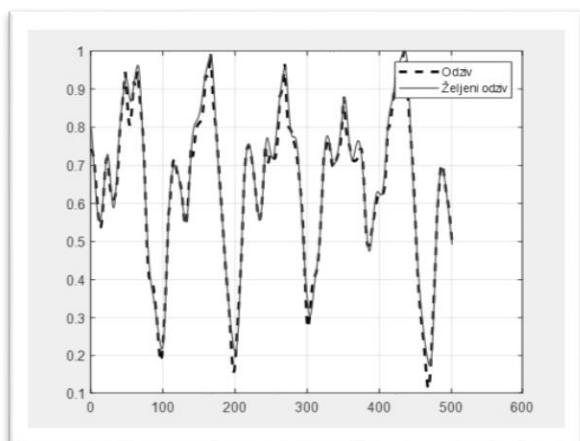
Test 3

Slika 4.35. Odzivi mreže kaotičnih vremenskih serija na testnim primjerima pri konfiguraciji 2-10-1

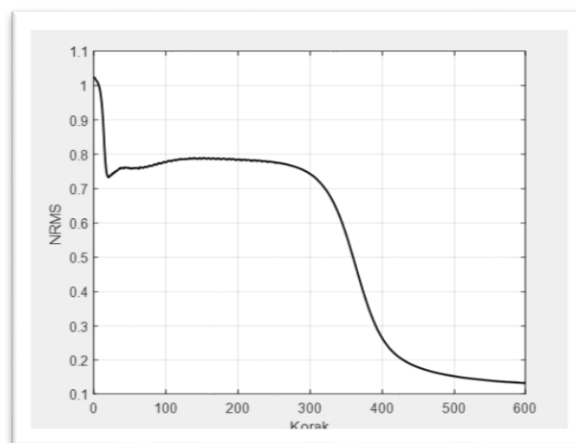
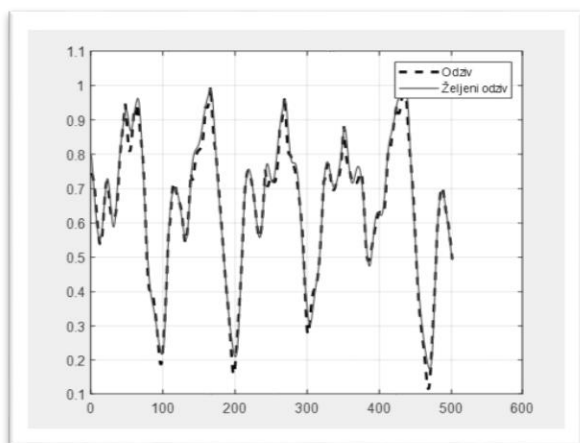
Na temelju slika 4.35. i 4.36. može se iščitati da, neovisno o strukturi, mreža lošije prati željeni signal i ima slabija svojstva generalizacije. NRMS vrijednosti svih triju testova se kreću oko vrijednosti 0.11, s najboljim rezultatom u Testu 2 s deset neurona (NRMS = 0.100277).

4.3.2. ReLU s prilagodljivim koeficijentom nagiba

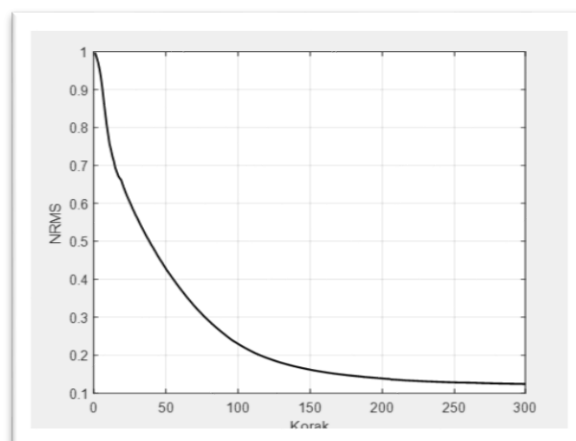
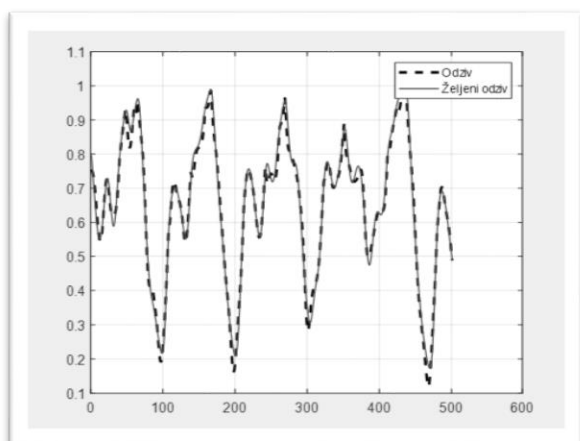
Upotrebom ReLU aktivacijske funkcije s prilagodljivim nagibom, prema slikama 4.36. 4.37. i 4.38. te Tablici 3, ne postižu se bitnije razlike u preciznosti sustava. Naime vrijednosti NRMS parametra i dalje iznose približno 0.11 te se ne razlikuju bitno od razina NRMS-a za prethodnu funkciju. Također brzina konvergiranja ostaje približno ista te se ne javljaju dodatne anomalije. Možemo slobodno reći da upotrebom ovakve funkcije aproksimacijsko svojstvo ostaje isto.



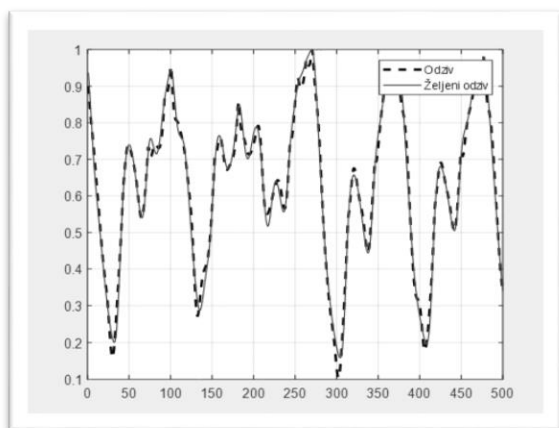
Slika 4.36. Odziv mreže nakon učenju skupa s NRMS parametrom u konfiguraciji 2-2-1



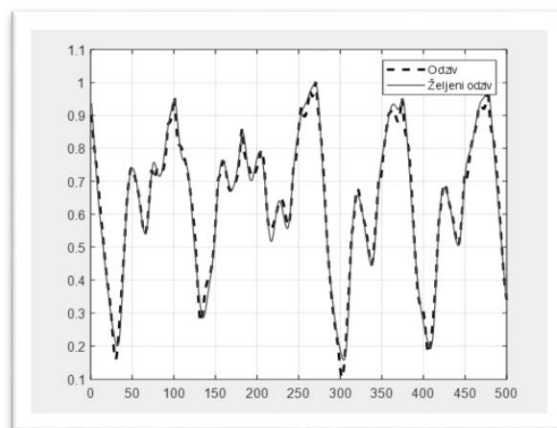
Slika 4.38. Odziv mreže nakon učenju skupa s NRMS parametrom u konfiguraciji 2-5-1



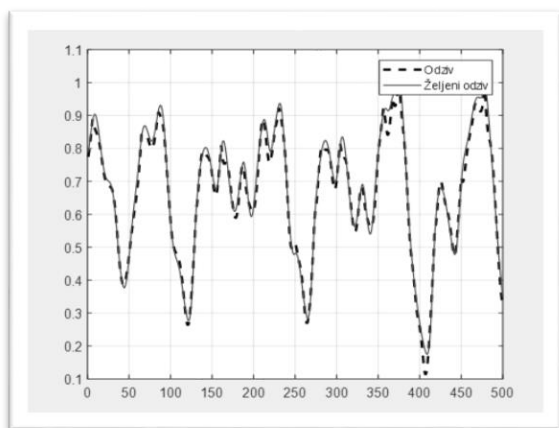
Slika 4.37. Odziv mreže nakon učenju skupa s NRMS parametrom u konfiguraciji 2-10-1



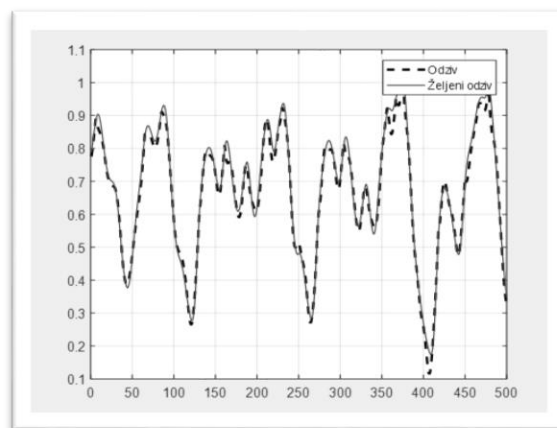
Test 1



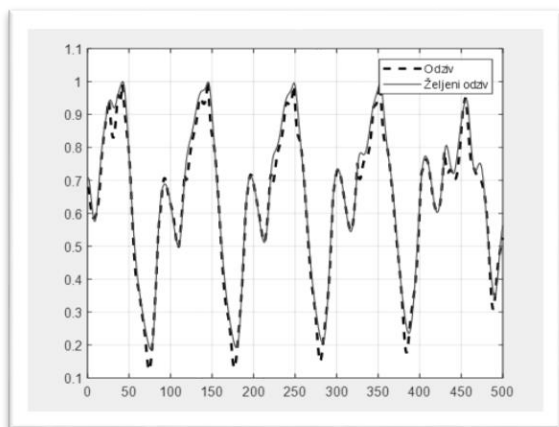
Test 1



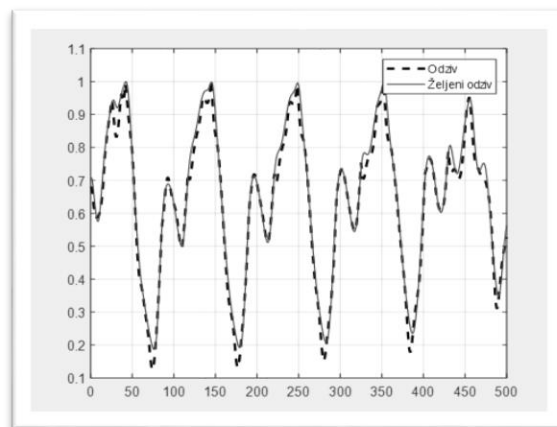
Test 2



Test 2



Test 3



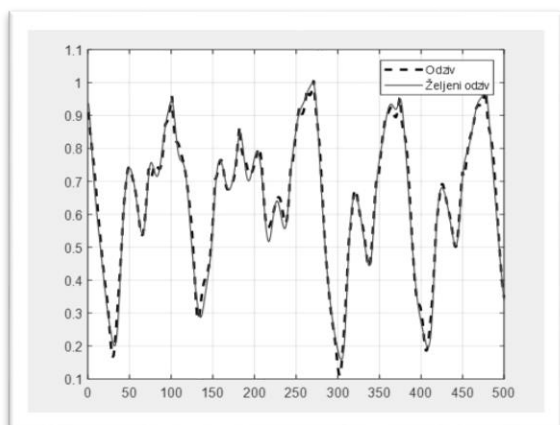
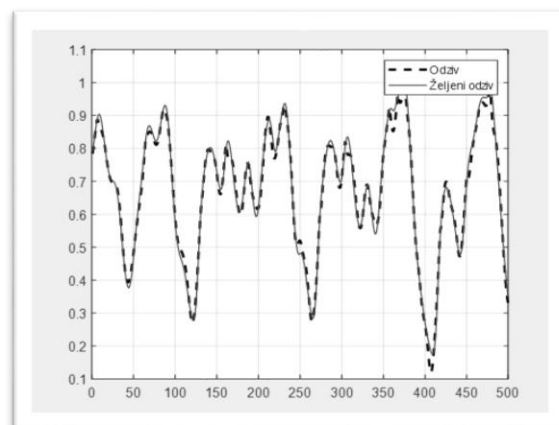
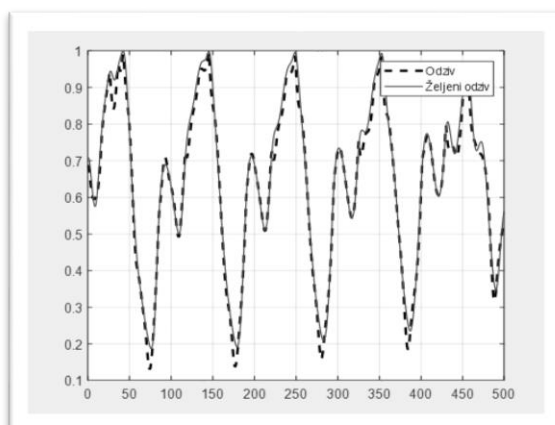
Test 3

(a)

(b)

Slika 4.39. Odzivi mreže kaotičnih vremenskih serija na testnim primjerima pri konfiguraciji
 a) 2-2-1
 b) 2-5-1

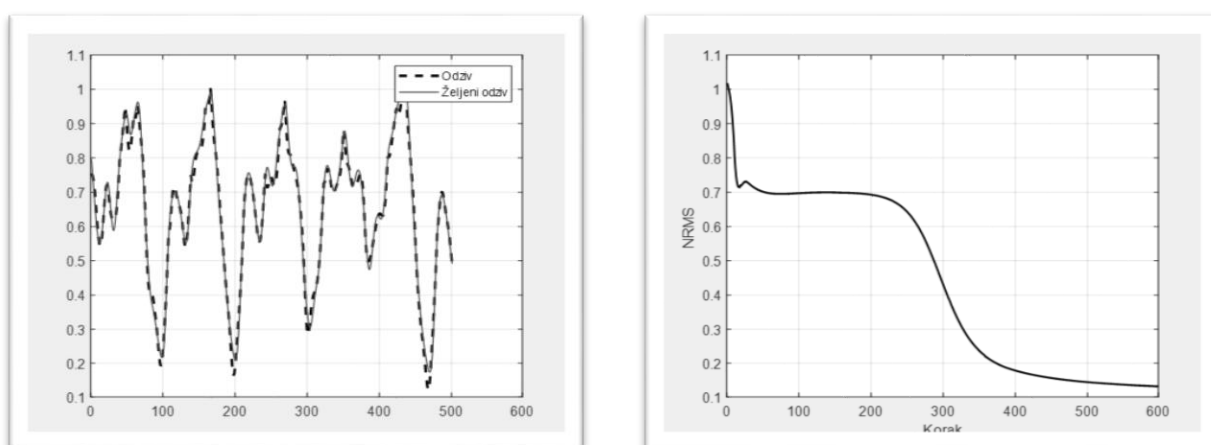
Iz slika 4.39. i 4.40. te slika 4.34. i 4.35. jasno je vidljivo da se grafovi unutar ovih funkcija ne razlikuju previše te se može zaključiti da pružaju podjednaka svojstva pri generiranju izlaznih vrijednosti ovakvog tipa sustava. Štoviše, detaljnijom usporedbom vrijednosti NRMS-a ove aktivacijske funkcije s klasičnom ReLU mogu se primijetiti lošije ocijene u nekim testovima (Tablica 2 i Tablica 3).

**Test 1****Test 2****Test 3**

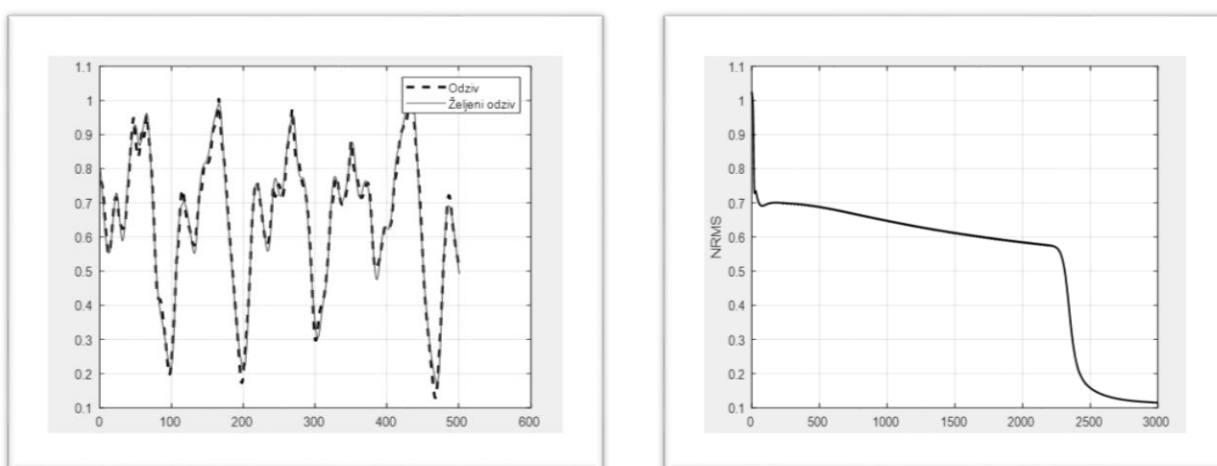
Slika 4.40. Odzivi mreže kaotičnih vremenskih serija na testnim primjerima pri konfiguraciji 2-10-1

4.3.3. ReLU s prilagodljivim pragom aktivacije

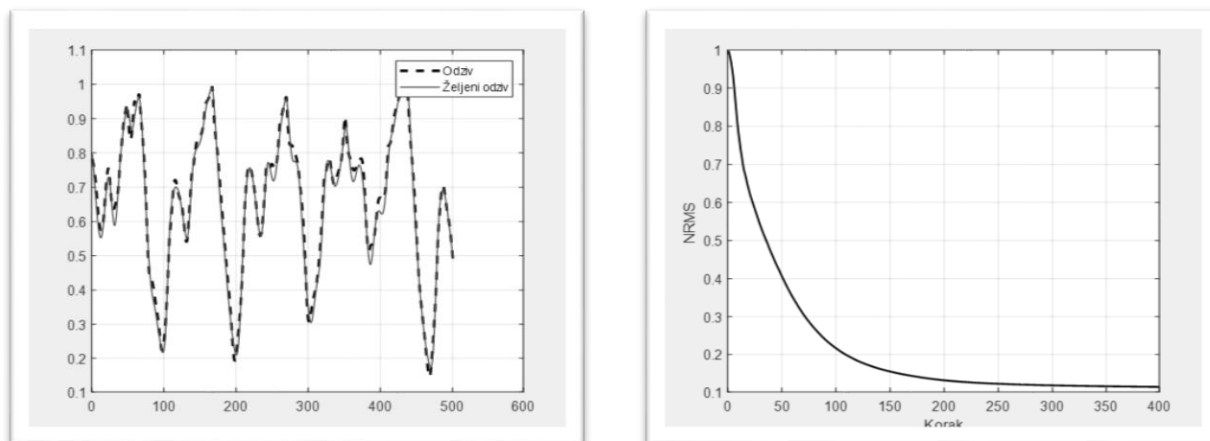
Tek korištenjem ReLU aktivacijske funkcije s prilagodljivim pragom aktivacije (slike 4.41. 4.42. i 4.43.) mogu se uočiti promjene u preciznosti mreže. Naime primjenom ovakvog tipa aktivacijske funkcije u strukturi s 10 neurona unutar skrivenog sloja, dolazi do relativno brzog konvergiranja NRMS-a. Mjera odstupanja za ovaj primjer učenja (Tablica 3) iznosi oko 0.11, što je u skladu s prethodnim rezultatima. Unatoč tome mreža s ovom aktivacijskom funkcijom pokazuje najbolja generalizacijska svojstva, što će biti spomenuto u nastavku.



Slika 4.41. Odziv mreže nakon učenja skupa s NRMS parametrom u konfiguraciji 2-2-1

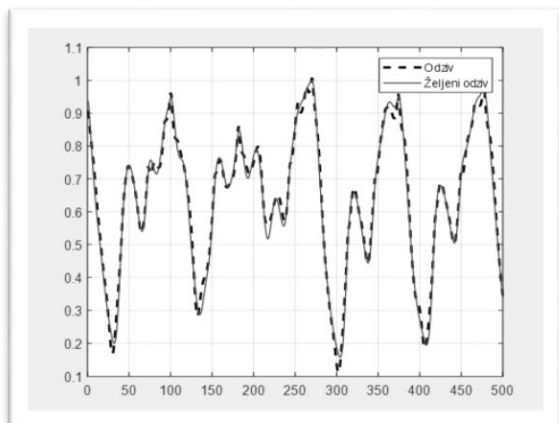


Slika 4.42. Odziv mreže nakon učenja skupa s NRMS parametrom u konfiguraciji 2-5-1

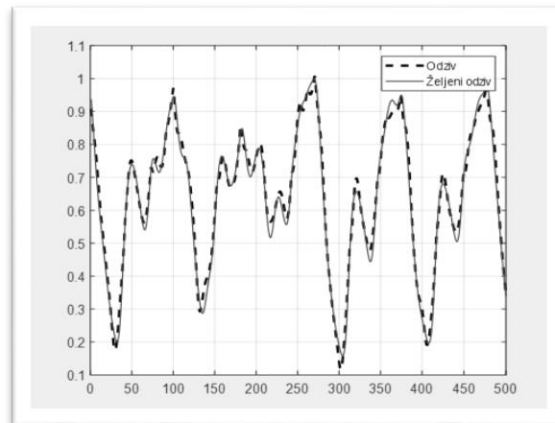


Slika 4.43. Odziv mreže nakon učenju skupa s NRMS parametrom u konfiguraciji 2-10-1

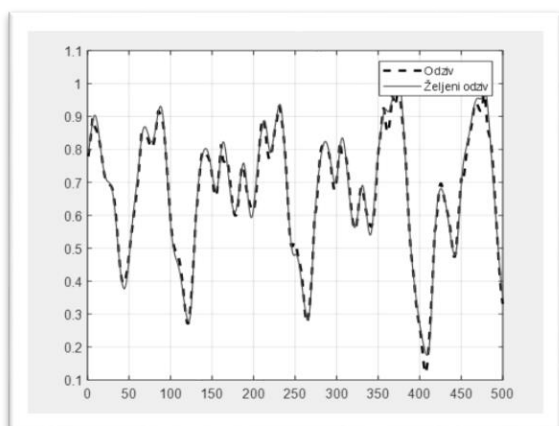
Analizom grafova sa slika 4.44. i 4.45. provedenih na prethodno spomenutoj funkciji, dolazi se do zaključka da mreža s prilagodljivim pragom aktivacije u konfiguraciji s 10 neurona u skrivenom sloju (slika 4.45.) pokazuje najbolje rezultate među svim prikazanim strukturama. To se može uočiti i na temelju odziva Testa 3, gdje je odstupanje između stvarnog i željenog odziva jedva vidljivo. Stoga s obzirom na jedva primjetno odstupanje u Testu 3 (NRMS = 0.0871), može se zaključiti da ova mreža ima najbolja svojstva generalizacije.



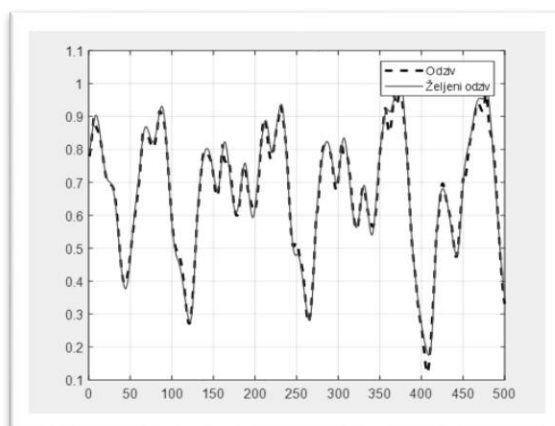
Test 1



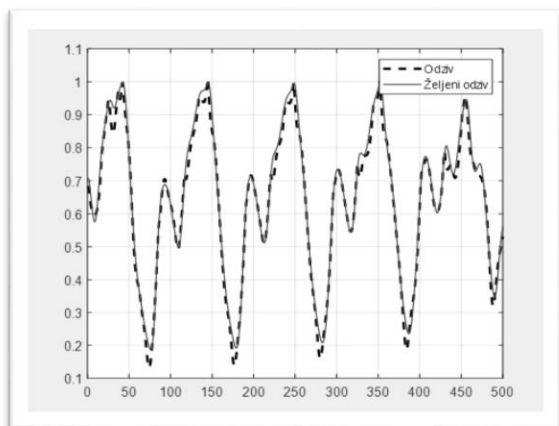
Test 1



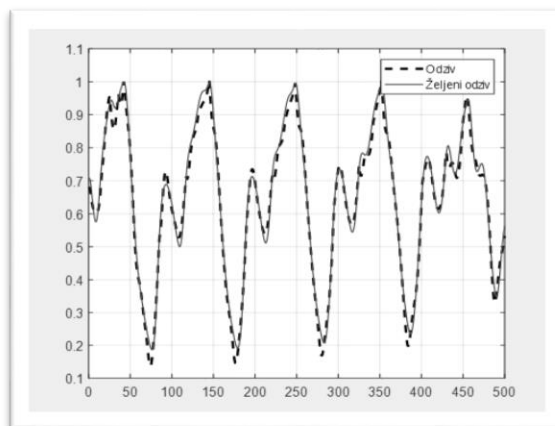
Test 2



Test 2



Test 3



Test 3

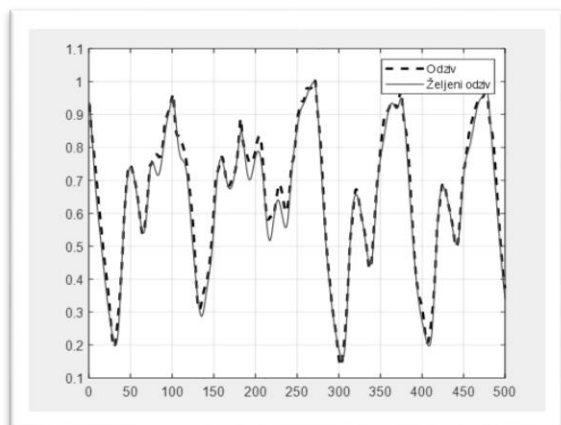
(a)

(b)

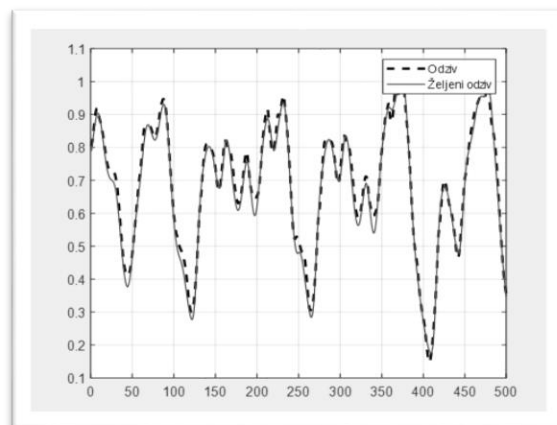
Slika 4.44. Odzivi mreže kaotičnih vremenskih serija na testnim primjerima pri konfiguraciji

a) 2-2-1

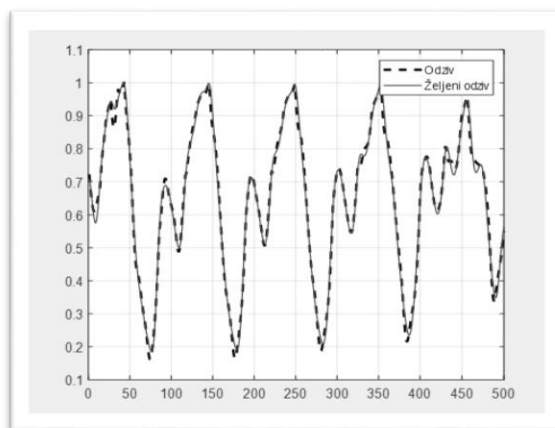
b) 2-5-1



Test 1



Test 2



Test 3

Slika 4.45. Odzivi mreže kaotičnih vremenskih serija na testnim primjerima pri konfiguraciji 2-10-1

4.3.4. Tablični prikaz NRMS-a

Tablica 3 Krajnje vrijednosti NRMS-a za kaotični sustav

<i>Kaotični sustav</i>		<i>5-2-1</i>	<i>5-5-1</i>	<i>5-10-1</i>
<i>ReLU aktivacijska funkcija</i>	UČENJE	0.1121	0.1117	0.1067
	Test 1	0.1084	0.1085	0.1166
	Test 2	0.1051	0.1048	0.1003
	Test 3	0.1264	0.1256	0.1104
<i>ReLU s prilagodljivim koeficijentom nagiba</i>	UČENJE	0.1284	0.1245	0.1124
	Test 1	0.1093	0.1085	0.1090
	Test 2	0.1211	0.1169	0.1030
	Test 3	0.1494	0.1443	0.1262
<i>ReLU s prilagodljivim pragom aktivacije</i>	UČENJE	0.1126	0.1148	0.1096
	Test 1	0.1082	0.1192	0.1363
	Test 2	0.1060	0.1153	0.1147
	Test 3	0.1273	0.1239	0.0872

5. Zaključak

Neuronske mreže danas se primjenjuju u različitim znanstvenim i tehnološkim područjima, a očekuje se da će u budućnosti postati sveprisutne i integrirane u svakodnevni život. Zbog toga postoji stalna potreba za njihovim razvojem i unaprjeđenjem karakteristika. U skladu s tim u ovome je radu, korištenjem troslojne perceptronske neuronske mreže, provedena usporedna analiza klasične ReLU aktivacijske funkcije neurona skrivenog sloja i njezine dvije adaptivne varijante. Uz klasičnu ReLU aktivacijsku funkciju analizirana je i primjena ReLU funkcije s prilagodljivim koeficijentom nagiba i funkciju s prilagodljivim pragom aktivacije.

Programska podrška razvijena u okviru ovog rada omogućuje korisniku jednostavan odabir aktivacijske funkcije i različitih parametara učenja, dok se topologija mreže može lako podesiti korištenjem datoteka s podacima za učenje i početnim vrijednostima težinskih faktora. U svrhu navedene analize korištena su tri različita testa: identifikacija linearnog proporcionalnog člana prvog reda, identifikacija nelinearnog dinamičkog sustava i predviđanje Mackey-Glass vremenske serije.

Prilikom identifikacije linearnog dinamičkog člana, testiranjem sve tri aktivacijske funkcije i sve tri strukture mreže, nisu zabilježena značajna odstupanja od željenog odziva. Također nisu primijećene niti značajne razlike u brzini učenja jer je greška učenja u svim slučajevima konvergirala u sličnom broju koraka. Na temelju toga može se zaključiti da sve tri aktivacijske funkcije pokazuju jednaku učinkovitost u identifikaciji ovakvog tipa modela.

Kod identifikacije nelinearnog dinamičkog sustava mreža s prilagodljivim pragom aktivacije postiže najbolje rezultate. Ova aktivacijska funkcija, primijenjena u strukturi s 10 neurona u skrivenom sloju, pokazuje odstupanja koja su gotovo idealna. Zanimljivo je primijetiti da ReLU aktivacijska funkcija s prilagodljivim koeficijentom nagiba daje rezultate koji su jednaki ili čak lošiji u odnosu na običnu ReLU aktivacijsku funkciju.

Najbolji odziv u slučaju problema predviđanja Mackey-Glass vremenske serije ponovno pokazuje ReLU aktivacijska funkcija s prilagodljivim pragom aktivacije. Ova funkcija, primijenjena u strukturi s 10 neurona u skrivenom sloju, postiže zanemariva odstupanja. Usporedbom ostale dvije aktivacijske funkcije nisu uočene značajne razlike u odzivima mreže.

Usporedbom sva tri testa perceptronske neuronske mreže može se zaključiti da ReLU aktivacijska funkcija s prilagodljivim pragom aktivacije pokazuje najbolja svojstva aproksimacije i generalizacije. Njena jednostavnost u implementaciji i visoka točnost odziva čine je idealnim izborom za modeliranje kako jednostavnih, tako i složenih dinamičkih sustava. Osim toga njezina uporaba ne zahtijeva preveliku računalnu snagu, memoriju niti dugo vrijeme treniranja.

LITERATURA

- [1] Branko Novaković, Dubravko Majetić, Mladen Široki. Umjetne neuronske mreže, 1998.
- [2] Hongyang Gao, Lei Cai, Shuiwang Ji. Adaptive Convolutional ReLUs
- [3] Patrick H. Winston. Lecture 12A: Neural Nets <https://ocw.mit.edu/courses/6-034-artificial-intelligence-fall-2010/resources/lecture-12a-neural-nets/>
- [4] Sparsh Gupta. 7 Common Loss Functions in Machine Learning <https://builtin.com/machine-learning/common-loss-functions>
- [5] GeeksforGeeks. Activation functions in Neural Networks <https://www.geeksforgeeks.org/activation-functions-neural-networks/>
- [6] Michael Nielsen. Neural Networks and Deep Learning <http://neuralnetworksanddeeplearning.com/index.html>
- [7] Danke Brezak. Neuronske mreže u proizvodnji <https://e-ucenje.fsb.hr/>