SVEUČILIŠTE U ZAGREBU
FAKULTET STROJARSTVA I BRODOGRADNJE

Završni rad

Joško Markučič

Zagreb, 2016.

SVEUČILIŠTE U ZAGREBU
FAKULTET STROJARSTVA I BRODOGRADNJE

Završni rad

Mentor:
Prof. dr. sc. Dubravko Majetić

Student:
Joško Markučič

Zagreb, 2016.

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 19-09-2016	Prilog
Klasa: 602-04/16-6/3	
Ur.broj: 15-1703-16-319	

ZAVRŠNI ZADATAK

Student: **Joško Markučič** Mat. br.: 0035185765

Naslov rada na hrvatskom jeziku: **Generalizacijsko svojstvo unaprijedne neuronske mreže**

Naslov rada na engleskom jeziku: **Generalisation property of feed forward neural network**

Opis zadatka:

Na primjeru učenja linearnog dinamičkog sustava prvog reda treba pokazati generalizacijsko svojstvo umjetne neuronske mreže s jednim skrivenim slojem neurona. Pri tome u strukturi skrivenog sloja treba koristiti više različitih aktivacijskih funkcija. Za ubrzanje algoritma učenja s povratnim prostiranjem pogreške odabrati metodu zamaha prvog i drugog reda. Učenje neuronske mreže treba obaviti principima učenja po skupu i učenja po uzorku.

U radu treba načiniti slijedeće:

1. Izvesti matematički model učenja po principu povratnog prostiranja pogreške za statičku neuronsku mrežu s jednim sakrivenim slojem neurona.
2. Radi ubrzanja procesa učenja matematički model učenja proširiti metodom zamaha (momentuma) prvog te drugog reda.
3. Za primjer učenja odabrati ponašanje nelinearnog i linearnog dinamičkog člana prvog reda.
4. U postupku analiziranja generalizacijskog svojstva naučene mreže koristiti test datoteke s različitim ulaznim pobudnim funkcijama dinamičkih članova.
5. Usporediti učenje i testiranje naučenih neuronskih mreža s različitim aktivacijskim funkcijama i različitim algoritmima učenja.
6. Programsku podršku načiniti u nekom od dostupnih matematičkih programskih paketa.
7. Izvesti zaključke rada.

Zadatak zadan:

25. studenog 2015.

Rok predaje rada:

1. rok: 25. veljače 2016.
2. rok (izvanredni): 20. lipnja 2016.
3. rok: 17. rujna 2016.

Predviđeni datumi obrane:

1. rok: 29.2., 02. i 03.03. 2016.
2. rok (izvanredni): 30. 06. 2016.
3. rok: 19., 20. i 21. 09. 2016.

Zadatak zadao:

Prof. dr. sc. Dubravko Majetić

Predsjednik Povjerenstva:

Prof. dr. sc. Zoran Kunica

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

Posebno se zahvaljujem voditelju rada prof. dr. sc. Dubravku Majetiću na prihvaćanju mentorstva, pružanju korisnih savjeta te stručne pomoći pri izradi rada.

Također, zahvaljujem obitelji i djevojci Andrei na pruženoj potpori tijekom studija.

Joško Markučič

Sadržaj

Popis slika

Popis tablica

Popis oznaka

$\mathbf{C}$	vektor parametara c_j gaussove aktivacijske funkcije
d_k	željena vrijednost k-og izlaza neuronske mreže
∇E	gradijent pogreške
E	funkcija cilja (suma kvadrata pogrešaka)
K_p	pojačanje sustava
MS	srednja kvadratna pogreška
$\mathbf{net_H}$	matrica izlaza sumatora skrivenoga sloja
net_{Hj}	vrijednost izlaza sumatora j-og neurona skrivenog sloja
$\mathbf{net_O}$	matrica izlaza sumatora izlaznoga sloja
net_{Ok}	vrijednost izlaza sumatora k-og neurona izlaznog sloja
$NRMS$	normalizirani korijen srednje kvadratne pogreške
O_k	k-ti izlaz neuronske mreže
O_k	dobivena vrijednost k-og izlaza neuronske mreže
RMS	korijen srednje kvadratne pogreške
$\mathbf{S}$	vektor parametara σ_j gaussove aktivacijske funkcije
$\mathbf{V}$	matrica težinskih koeficijenata skrivenoga sloja
v_{ij}	težinski koeficijent veze između j-og neurona skrivenoga sloja i i-og neurona ulaznoga sloja
$\mathbf{W}$	matrica težina
$\mathbf{W}$	matrica težinskih koeficijenata izlaznoga sloja
w_{kj}	težinski koeficijent veze između k-og neurona izlaznoga sloja i j-og neurona skrivenoga sloja
$\mathbf{x}$	ulazni vektor perceptrona
$\mathbf{Y}$	matrica izlaza neurona skrivenoga sloja
$\mathbf{y}$	izlazni vektor perceptrona
y	vrijednost izlaza neurona
y_j	vrijednost izlaza j-og neurona
$\mathbf{Z}$	matrica ulaza
Z_i	vrijednost i-og ulaza
α	koeificijent momentuma prvog reda
β	koeficijent momentuma drugog reda
δ	karakteristična vrijednost algoritma povratnog prostiranja pogreške
$\Delta\vartheta(n)$	veličina promjene parametara učenja
η	koeficijent brzine učenja

γ	aktivacijska funkcija neurona
γ'	derivacija aktivacijske funkcije neurona
σ_{d_n}	standardna devijacija željenih vrijednosti izlaza mreže
ϑ	parametar učenja (težinski koeficijent)
$\vartheta(n)$	trenutna vrijednost parametra učenja
$\vartheta(n+1)$	nova vrijednost parametra učenja

Sažetak

U ovom radu prikazan je postupak učenja umjetne neuronske mreže s povratnim rasprostiranjem pogreške u zadatku prikaza generalizacijskog svojstva neuronske mreže na primjeru ponašanja nelinearnog i linearnog dinamičkog člana prvog reda. Glavna značajka umjetnih neuronskih mreža je mogućnost estimacije funkcija koje ovise o velikom broju ulaza koji nisu općenito poznati. Uz sam postupak učenja u ovom radu je prikazano i testiranje mreže koje se provodi nakon samog postupka učenja. Programska podrška kojom je implementirana opisana umjetna neuronska mreža i postupak učenja u cijelosti je izrađena u programskom paketu MATLAB. Opis, učenje i glavne značajke umjetnih neuronskih mreža prikazane su i detaljno upisane u uvodu, a zatim je u nastavku prikazan način učenja i rada zadane mreže.

Ključne riječi

- neuronske mreže
- umjetna inteligencija
- umjetni neuron
- učenje neuronske mreže
- testiranje neuronske mreže
- težinski faktori
- zamah prvog reda
- zamah drugog reda
- povratno rasprostiranje pogreške
- generalizacijsko svojstvo
- dinamički član prvog reda

Summary

This paper shows the feed forward neural networks with error back-propagation generalisation property of nonlinear and linear first order dynamic. The main attribute of artificial neural networks is the ability to estimate or approximate functions that can depend on a large number of inputs that are generally unkown. Aside from the learning process, this paper also shows the neural network testing process which comes right after the learning stage. The program implementing the described artificial network and learning process has been entirely made in MATLAB. Description, main attributes and learning of artificial neural networks are shown and described in the introduction, after which the learning process is shown.

Keywords

- neural networks
- artificial intelligence
- artificial neuron
- artificial network learning
- artificial network testing
- neural network weights
- first order momentum
- second order momentum
- error back-propagation
- generalisation property
- first order dynamic term

1 Uvod

Način na koji ljudski mozak razmišlja i zaključuje značajno se razlikuje od načina na koji računala i računalni algoritmi funkcioniraju. Područje umjetnih neuronskih mreža, a i šire područje umjetne inteligencije, bavi se problematikom računalnog modeliranja načina na koji ljudi razmišljaju i zaključuju. Umjetne neuronske mreže pokušavaju modelirati biofiziologiju mozga te na taj način omogućiti sposobnost pohrane i obrade informacija slično ljudskome mozgu putem računala.

1.1 Umjetne neuronske mreže

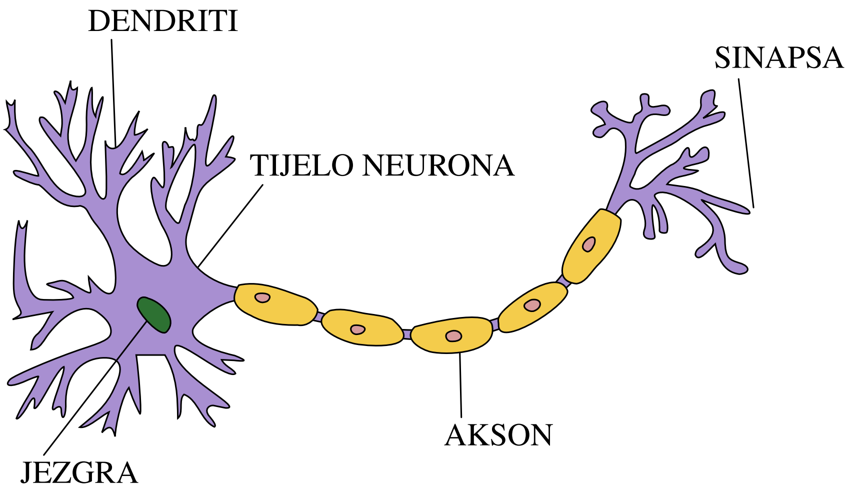
Pod pojmom umjetne neuronske mreže obično smatramo složen sustav sastavljen od osnovnih elemenata koje nazivamo neuronima. Svi elementi sustava (neuroni) u međusobnoj su interakciji te tako s okolinom sustava grade funkcionalnu cjelinu.

William James 1890. [1] iznosi tvrdnju: „Aktivnost bilo koje točke mozga čovjeka predstavlja zbroj tendencija svih ostalih točaka da se "prazne" u nju. Ove tendencije proporcionalne su: broju točaka (broju veza) koje djeluju na promatranu točku, intenzitetu tih uzbuda (težini veza) i odsutnosti rivalne točke koja nije u funkcionalnoj vezi s promatranom točkom, a u koju bi se "pražnjenje" ostalih točaka moglo skrenuti.”

Na temelju ove tvrdnje izgrađena je osnovna struktura umjetnog neuron. Zamijenimo li određenu točku u mozgu čovjeka neuronom, onda aktivnost tog neurona možemo modelirati kao zbroj **otežanih ulaza** neurona. Otežani ulazi su ulazi pomnoženi određenim faktorima koji se nazivaju **težinama** neurona. Stoga aktivnost neurona ovisi o broju ulaza iz okoline, intenzitetu tih veza, te o pragu osjetljivosti koju stanje neurona treba dosegnuti prije nego što "ispali impuls" preko svog izlaza u okolinu. Okruženje neurona čine ostali neuroni umjetne neuronske mreže i okruženje te mreže. Formalizaciju aktivnosti umjetnog neurona, osnovanog na tvrdnji Williama Jamesa, dali su McCulloch i Pitts 1943. godine [2], predloživši jednostavan model umjetnog neurona.

1.2 Biološki neuron

Kako bi mogli opisati umjetni neuron, potrebno je ukratko izložiti osnovnu strukturu i funkcije biološko neurona prikazanog ilustrativno na slici 1.1. U najosnovnijem obliku biološki neuron je stanica koja se sastoji se od tijela neurona, aksona i mnoštva dendrita koji okružuju tijelo neurona.
Akson je produljenje živčane stanice koji prenosi električne impulse nastale u tijelu stanice. Akson se svojim krajem spaja na dendrite drugih neurona ili na stanice mišića i žlijezdi putem sinapse. **Sinapsa** je mali razmak između završetka jednog neurona i dendrita ili tijela drugog neurona, na kojoj se signal prenosi kemijskim ili električnim putem.
Neuron šalje impuls kroz akson kada je doveden u stanje dovoljne uzbude koje je postignuto njegovim trenutnim stanjem te utjecajem signala drugih neurona preko njegovih dendrita ili tijela. Signali koji se prenose dendritima mogu biti smirujući ili uzbudni. Iz matematičkog aspekta takvi signali suprotnog su predznaka.

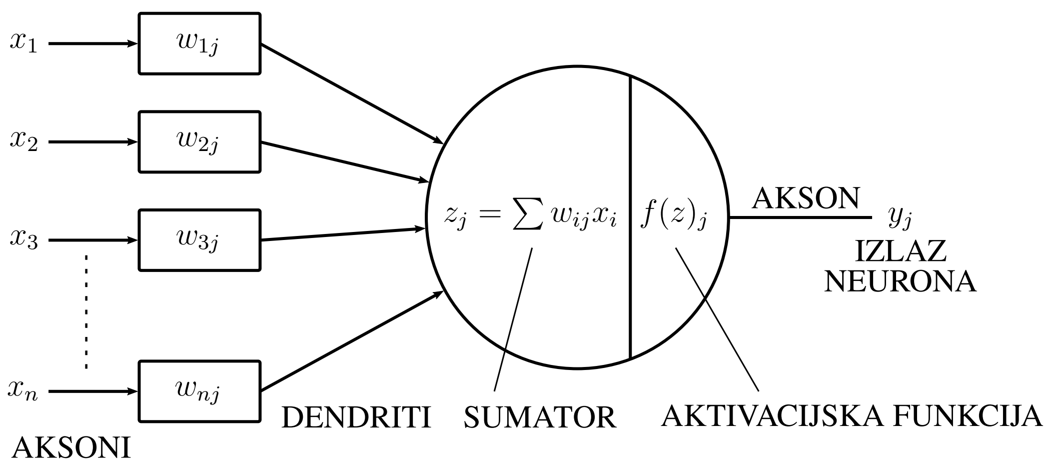


Slika 1.1: Struktura biološkog neurona [3]

Prema procjeni mozak čovjeka ima preko 100 milijardi neurona, koji su mnogo složeniji nego je ovdje prikazano. Ti neuroni komuniciraju kroz tijelo preko nuronskih vlakana koje grade preko 100 trilijuna veza. Upravo ta mreža neurona odgovorna je za našu mogućnost razmišljanja, učenja, emocija, spoznaja i percepcija kao i za izvođenje motorike i senzorike autonomnih funkcija. Kako točno radi biološka neuronska mreža još uvijek nije točno razjašnjeno.

1.3 Umjetni neuron

Temeljna ideja umjetnog neurona jest da oponaša osnovne funkcije biološkog neurona, slika 1.2. Kod umjetnog neurona ekvivalent tijela biološkog neurona je sumator, a ulogu dendrita preuzimaju ulazi u sumator. Izlaz sumatora predstavlja akson, a uloga praga osjetljivosti bioloških neurona preslikava se na *aktivacijske funkcije*. Funkciju sinaptičkih veza opisuju težinski faktori, preko koji se ostvaruje veza umjetnog neurona sa okolinom.



Slika 1.2: Struktura umjetnog neurona

Težinski faktori mogu biti pozitivni ili negativni, a kod modernih umjetnih neuronskih mreža i neka funkcija (varijabilni težinski faktor). Ukoliko je težinski faktor jednak nuli onda odgovarajuća veza s okolinom neurona ne postoji.
Izlaz sumatora povezan je na ulaz aktivacijske funkcije čiji izlaz je ujedno i izlaz umjetnog neurona. Aktivacijske funkcije mogu biti linearne i nelinearne. Linearne aktivacijske funkcije karakterizira množenje izlaza sumatora nekim faktorom, dok nelinearne aktivacijske funkcije mogu poprimiti različite oblike, ali se najčešće koriste funkcije praga osjetljivosti, sigmoidalne funkcije, hiperbolične i harmoničke funkcije.

1.4 Vrste i učenje umjetnih neuronskih mreža

Umjetne neuronske mreže moguće je kategorizirati po mnogim kriterijima. Često mreže razlikujemo prema broju slojeva. Paralelno složen skup neurona gradi jedan sloja neuronske mreže, a prema broju slojeva mreže mogu biti jednoslojne i višeslojne. Uobičajeno je da višeslojne mreže posjeduju ulazni i izlazni sloj između kojih se nalaze skriveni slojevi.

Ukoliko se slojevi mreže povežu tako da signali putuju samo u jednom smjeru, od ulaza do izlaza, onda se radi o unaprijednim neuronskim mrežama (engl. *feedforward neural networks*). Postojanjem povratnih petlji (suprotni smjer signala) u slojevima neuronske mreže onda govorimo o povratnim neuronskim mrežama (engl. *feedback/recurrent neural networks*). Klase problema koje mogu riješiti ovakve mreže međusobno su isključive.

Prema obliku učenja razlikujemo supervizorno učenje (uz nadzor) i nesupervizorno učenje (bez nadzora). Učenje uz nadzor zahtijeva vanjskog "učitelja" neuronske mreže, koji promatra ponašanje mreže te ju podešava dok se ne dobije željeno ponašanje mreže. Kod supervizornog učenja najprije se usvaja određena struktura mreže (broj ulaza, neurona, slojeva, izlaza te težina mreže). Usvajaju se početne vrijednosti težina neuronske mreže (obično putem generatora slučajnih brojeva). Zatim se na ulaz dovodi skup ulaznih varijabli čiji se dobiveni izlazni skup uspoređuje sa skupom željenih izlaznih vrijednosti. Razlika željenih i stvarnih izlaza neuronske mreže određuje pogrešku mreže, koja se koristi za promjenu težina mreže preko algoritma učenja. Cijeli postupak ponavlja se iteracijski dok pogreška mreže nije unutar traženih raspona. Nakon procesa učenja slijedi proces testiranja neuronske mreže. To se radi novim skupom ulaza mreže koji nije sadržan u ulaznom skupu korištenom tokom učenja mreže. Iznos pogreške mreže u procesu testiranja služi za ocjenu robusnosti, odnosno generalizacijskih svojstava mreže.
Kod nesupervizornog učenja neuronske mreže ne koristi se vanjski učitelj. Neuronska mreža se sama organizira, pa se mreže učene ovom metodom nazivaju samoorganizirajuće neuronske mreže. Na ulaz mreže dovodi se skup ulaznih varijabli, a mreža se samoorganizira podešavanjem svojih parametara prema definiranom algoritmu. Obzirom da željeni izlaz mreže nije specificiran za vrijeme učenja mreže, rezultat učenja nije predvidiv.

2 Statička unaprijedna neuronska mreža s povratnim prostiranjem pogreške

2.1 Perceptron

Perceptron je umjetni neuron kod kojeg se kao aktivacijska funkcija koristi binarna funkcija praga osjetljivosti, definirana matematičkim izrazom

$$y_j = f(z_j) = \begin{cases} 1, & \text{za } z_j > \text{ prag} \\ 0, & \text{za } z_j \leq \text{ prag} \end{cases} \tag{2.1}$$

Perceptron je binarni neuron kod kojeg izlaz može poprimiti dvije vrijednosti (0 ili 1). Izlazna vrijednost 1 označava promatrani neuron aktivnim, dok vrijednost 0 označava neaktivno stanje.
Veći broj perceptrona može se povezati u mrežu sa jednim ulaznim i jednim izlaznim slojem. Takav oblik mreže naziva se jednoslojna perceptronska mreža (engl. *single-layer perceptron*), a primjenjuje se kao klasifikator kod problema raspoznavanja uzoraka. Struktura mreže određena je problemom, broj neurona ulaznog sloja n jednak je broju značajki koje opisuju problem, dok broj neurona izlaznog sloja odgovara broju skupina za koje se provodi klasifikacija. Neuroni ulaznog sloja potpuno su povezani s neuronima izlaznog sloja, dok neuroni unutar istog sloja nisu međusobno povezani.
Ulazni sloj prenosi signal do izlaznog sloja, u kojem se ulazne vrijednosti množe pridruženim težinama w_{ij} i sumiraju. Težine izlaznog sloja w_{ij} podesivi su parametri perceptronskih mreža, te su predmet učenja. Učenje se provodi pomoću skupa za učenje, koji se sastoji od većeg broja uzoraka opisanih ulaznim vektorima (značajkama) i pridruženim im željenim izlaznim vrijednostima. Cilj učenja je podešavanje nepoznatih težina w_{ij} na taj način da binarni izlazni neuron koji pripada istoj skupini kao i promatrani uzorak skupa za učenje poprimi vrijednost 1, a svi ostali neuroni vrijednost 0.

2.1.1 Učenje perceptrona

Perceptronske mreže učimo iterativnim postupkom, prema sljedećem algoritmu:

1. Za ulazni vektor $\mathbf{x}$ uzorka iz skupa za učenje izračuna se izlazni vektor $\mathbf{y}$:

   $$\mathbf{y} = \mathbf{x} \cdot \mathbf{W} \tag{2.2}$$

   gdje je $\mathbf{W}$ matrica težina, čiji se elementi u prvom koraku najčešće inicijaliziraju kao slučajni brojevi

2. (a) Ako je izlaz točan, povrat na točku 1.
   (b) Ako je izlaz pogrešan i ako je 0 dodaju se vrijednosti ulaznog vektora pripadajućim težinama
   (c) Ako je pogrešan i ako je 1 od pripadajućih tezxina oduzimaju se vrijednosti ulaznog vektora

3. Povrat na točku 1.

Postupak učenja iterativno se provodi za sve uzorke skupa za učenje tako dok svi nisu pravilno razvrstani u pripadajuće im skupine. Dokazano je da postupak učenja uspješno konvergira u konačnom broju koraka ukoliko su uzorci linearno separabilni [4].

2.1.2 Delta pravilo

Delta pravilo je generalizacija postupka učenja perceptrona, za neurone s kontinuiranim ulaznim i izlaznim vrijednostima. Delta pravilo osnova je za učenje većine modela neuronskih mreža. Razlika (delta) između željene i izračunate izlazne vrijednosti neurona definirana je izrazom

$$\delta_j = T_j - A_j \tag{2.3}$$

Gdje je

T_j – željena izlazna vrijednost neurona j
A_j – izračunata izlazna vrijednost neurona j

Podešavanje težina w_{ij} provodi se prema sljedećim izrazima

$$\Delta_{ij} = \eta \cdot \delta_j \cdot x_i \tag{2.4}$$

$$w_{ij}(n+1) = w_{ij}(n) + \Delta_{ij} \tag{2.5}$$

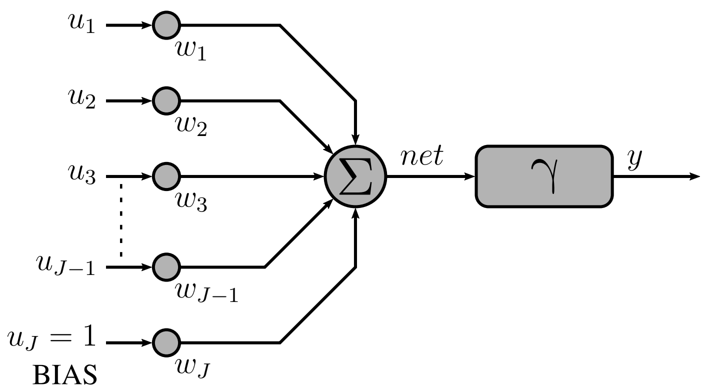
Gdje je

i – indeks neurona ulaznog sloja
j – indeks neurona izlaznog sloja
Δ_{ij} – faktor promjene težina
η – koeficijent brzine učenja
w_{ij} – težina prije podešavanja
$w_{ij}(n+1)$ – težina veze između neurona i ulaznog sloja i neurona j izlaznog sloja poslije podešavanja

2.2 Višeslojne neuronske mreže

2.2.1 Model statičkog neurona

Slika 2.1 jasno prikazuje da se statički neuron sastoji od više ulaza, a samo jednoga izlaza. Vrijedi uočiti kako ovakav oblik neurona sadrži dvije temeljne podfunkcije statičkog neurona od kojih je jedna *funkcija sume* Σ, a druga *aktivacijska funkcija* γ.
Radi mogućnosti odvijanja učenja svaki neuron također posjeduje poseban ulaz jedinične vrijednosti koji je u strukturi neuronske mreže realiziran vezom sa zasebnim neuronom oznake *Bias* konstantnog izlaza jednakog jedinici.



Slika 2.1: Model statičkog neurona

Funkcija sume koja predstavlja sumu umnožaka ulaza neurona i pripadajućih težinskih faktora veza daje vrijednost koju označavamo sa net. Aktivacijska funkcija γ preslikava vrijednosti dobivene funkcijom sume u izlaznu vrijednost neurona y.

$$net = \sum_{j=1}^{J} w_j \cdot u_j \tag{2.6}$$

$$y = \gamma(net) \tag{2.7}$$

Gdje je

net – rezultat funkcije sume
w_j – vrijednost j-te težine
u_j – vrijednost j-og ulaza
γ – aktivacijska funkcija neurona
y – izlazna vrijednost neurona

Aktivacijske funkcije su rastuće monotone funkcije sa zasićenjem, a odabir prigodne funkcije ovisi o skupu učenja i o skupu testiranja neuronske mreže. Često korištene aktivacijske funkcije su

- Bipolarna sigmoidalna funkcija (Slika 2.2)

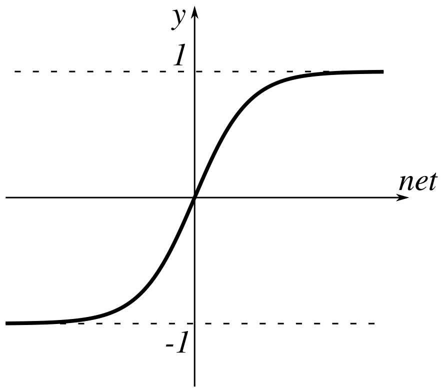
$$y = \frac{2}{1 + e^{-net}} - 1 \tag{2.8}$$

- Sinusna funkcija (Slika 2.3)

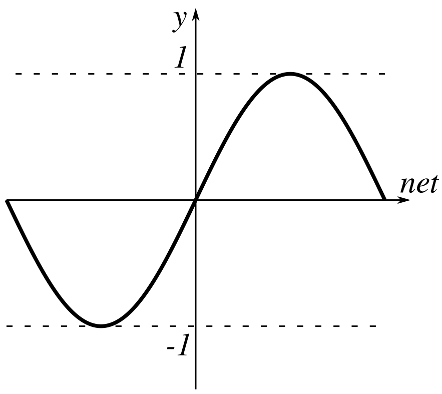
$$y = \sin(net) \tag{2.9}$$

- Gaussova funkcija (Slika 2.4)

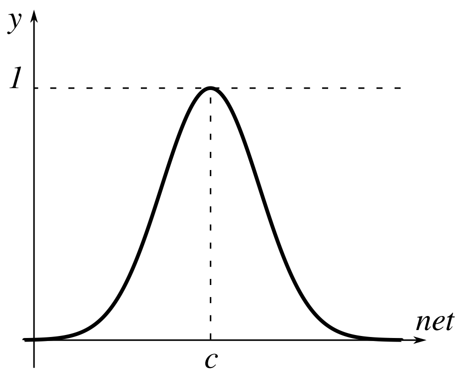
$$y = e^{-\frac{1}{2}\left(\frac{net-c}{\sigma}\right)^2} \tag{2.10}$$



Slika 2.2: Bipolarna sigmoidalna funkcja



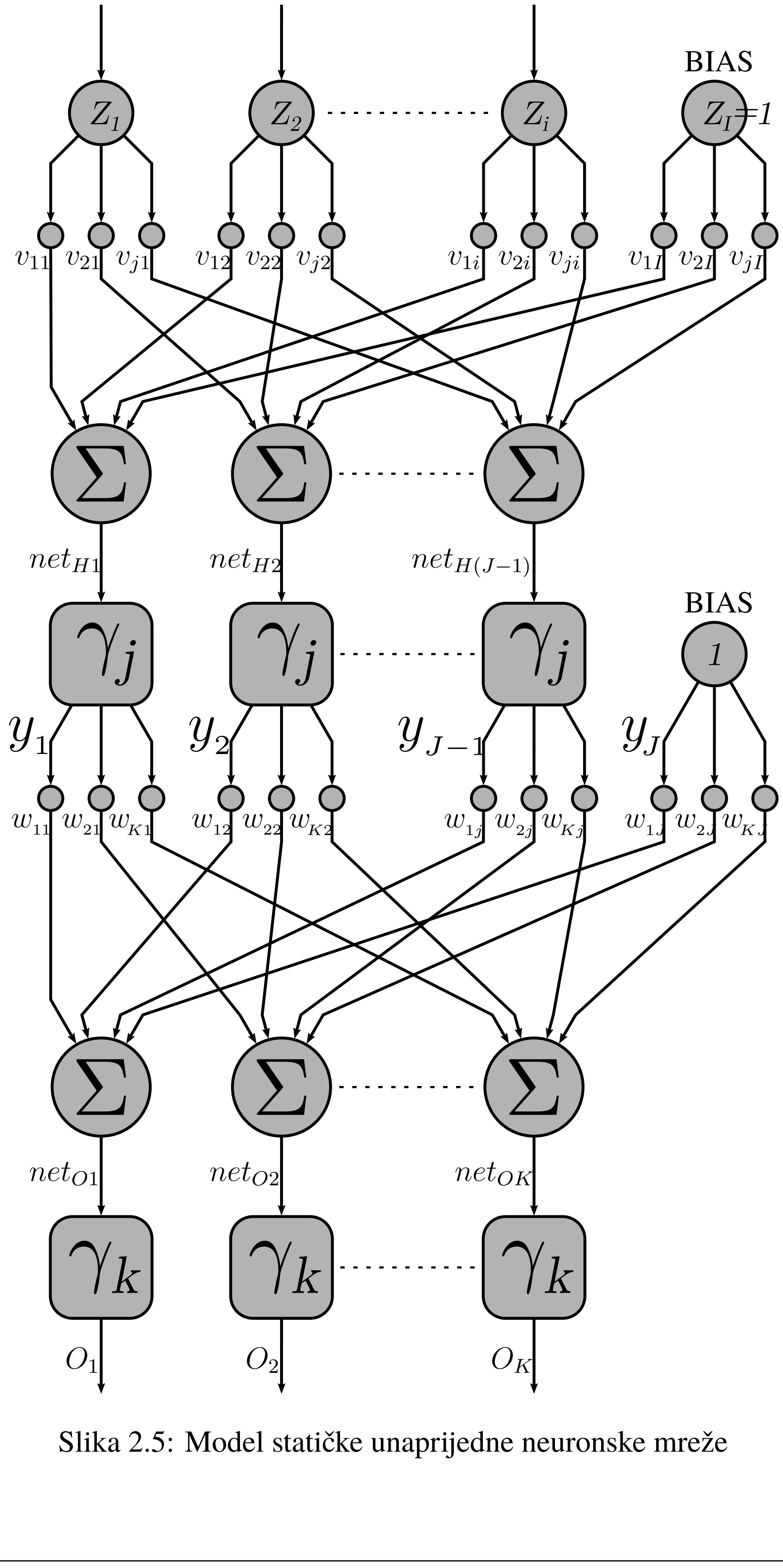
Slika 2.3: Sinusna funkcija



Slika 2.4: Gaussova funkcija

2.2.2 Model statičke neuronske mreže

Kako bi dobili strukturu (topologiju) neuronske mreže, potrebno je neurone organizirati u slojeve, a potom slojeve međusobno povezati vezama opterećenima težinskim koeficijentima. Razlikujemo tri tipa slojeva mreže: ulazni sloj, sakrivene slojeve, te izlazni sloj neurona. Ulazni i izlazni sloj u direktnoj su vezi s okolinom, dok sakriveni sloj ima interakciju samo unutar mreže. Najčešće upotrebljavani tip neuronske mreže je statička unaprijedna višeslojna neuronska mreža (engl. *SNN, static neural network*). U ovom radu korištena je statička unaprijedna neuronska mreža sa jednim sakrivenim slojem, a njen model prikazan je na slici 2.5.



Slika 2.5: Model statičke unaprijedne neuronske mreže

Iz sheme modela (Slika 2.5) vidljivo je da su ulazi neurona ulaznog sloja (Z_i) ujedno i ulazi u mrežu. Ulazni sloj potom je vezan sa sakrivenim slojem putem veza koje su opterećene težinama $v_j i$. Važno je uočiti kako su svi slojevi mreže potpuno umreženi, što znači da je svaki neuron promatranog sloja vezan sa svakim neuronom prethodnog sloja. Izuzetak su jedino neuroni imena *bias*, dakle neuroni stalne izlazne vrijednosti jednake jedinici.

2.3 Učenje povratnim rasprostiranjem greške

Postupke učenja možemo opisati pomoću tri različita karaktera: učenje s učiteljem, učenje bez učitelja i učenje koje je kombinacija prethodnih učenja.
Postupak učenja neuronske mreže ustvari je postupak podešavanja težinskih koeficijenata veza između slojeva mreže u svrhu približavanja izlaznih vrijednosti mreže željenim vrijednostima izlaza za zadane podatke iz skupa učenja. Stoga postupak učenja ne osigurava potpunu točnost izlaza mreže nego samo aproksimaciju željenih vrijednosti izlaza koja ovisi o više čimbenika kao što su zadatak učenja, topologija mreže te odabrani algoritam učenja.
Algoritmi se po načinu provođenja dijele na iterativne te na algoritme s jednim korakom učenja.
Postupak učenja u svakom koraku sastoji se od unaprijedne i povratne faze, dok se promjena parametara učenja može odvijati u *batch* proceduri ili *pattern* proceduri.
U *batch* proceduri parametri se mijenjaju jednom nakon prolaska čitavog ulaznog skupa učenja kroz mrežu, a promjena se odvija usrednjenom pogreškom, dok se kod *pattern* procedure parametri učenja mijenjaju za svaki ulazno-izlazni par skupa učenja.

U ovom radu prethodno prikazana statička unaprijedna neuronska mreža (Slika 2.5) učena je povratnim rasprostiranjem greške (engl. *error-back propagation*) korištenjem procedura po skupu (*batch*) i po uzorku (*pattern*).

2.3.1 Unaprijedna faza učenja statičke mreže

U ovoj fazi učenja za vrijednosti ulaza neuronske mreže zadanih u skupu učenja $\mathbf{Z}$ izračunavaju se izlazi mreže $\mathbf{O}$. Kako bi mogli izračunati izlaze mreže potrebno je prethodno odrediti početne vrijednosti težinskih koeficijenata skrivenoga ($\mathbf{v}$) i izlaznog sloja ($\mathbf{w}$). Obično se za početne vrijednosti postavljaju nasumično odabrani brojevi u rasponu vrijednosti ulaza iz skupa učenja. Za slučaj ulaza normiranih u rasponu $(-1 \mid 1)$ početne se vrijednosti odabiru u istom rasponu $(-1 \mid 1)$.

Sakriveni sloj
Prvo je potrebno izračunati net vrijednosti u skrivenome sloju.

$$net_{Hj} = \sum_{i=1}^{I} v_{ji} \cdot Z_i, \qquad j = 1, 2, ..., J - 1 \tag{2.11}$$

Gdje je

I – broj ulaznih neurona uključujući i bias (broj pravih ulaza je $I - 1$)
J – broj neurona u sakrivenom sloju uključujući i bias (broj sakrivenih neurona je $J - 1$)

Za izračunavanje izlaznih vrijednosti neurona y skrivenoga sloja potrebno je izračunati aktivacijsku funkciju za dobivene vrijednosti net_{Hj}. Vrijednosti izlaza sakrivenih neurona možemo računati nekom od prethodno navedenih aktivacijskih funkcija u izrazima (2.8), (2.9) ili (2.10). Za primjer ćemo odabrati bipolarnu sigmoidalnu funkciju (2.8) kao aktivacijsku funkciju.

$$y_j = \frac{2}{1 + e^{-net_{Hj}}} - 1, \qquad j = 1, 2, ..., J - 1 \tag{2.12}$$

$$y_J = 1 \tag{2.13}$$

Vrijednosti izlaza neurona sakrivenog sloja povezane su preko težinskih koeficijenata w_{kj} na ulaze neurona izlaznog sloja.

Izlazni sloj

$$net_{Ok} = \sum_{j=1}^{J} w_{kj} \cdot y_j, \qquad k = 1, 2, ..., K \tag{2.14}$$

Gdje je

K – broj neurona izlaznog sloja (broj izlaza)

Kao aktivacijsku funkciju izlaznog sloja dobro je odabrati linearnu funkciju radi mogućnosti postizanja vrijednosti izlaza mreže većeg od 1.

$$O_k = K_p \cdot net_{Ok}, \qquad k = 1, 2, ..., K \tag{2.15}$$

Gdje je

K_p – nagib linearne aktivacijske funkcije

2.3.2 Povratna faza učenja statičke mreže

Tokom povratne faze učenja vrši se korekcija težinskih koeficijenata veza između slojeva. Korekcija se osniva na pogrešci učenja koja ovisi o ostvarenim vrijednostima izlaza mreže i željenim vrijednostima izlaza mreže. Cijeli postupak se za svaki ulazno-izlazni par skupa učenja ponavlja dok se ne zadovolji uvjet minimalne pogreške (dozvoljeno odstupanje vrijednosti izlaza od željenih) koju zadaje učitelj.

Za funkciju cilja najčešće se koristi suma kvadrata pogreške kao mjera odstupanja vrijednosti izlaza od željenih.

$$E = \frac{1}{2} \sum_{n=1}^{N} (d_n - O_n)^2 \tag{2.16}$$

Gdje je

N – broj elemenata u skupu za učenje
$^1/_2$ – pojednostavljuje derivacije funkcije cilja
O_n – dobivena vrijednost izlaza
d_n – željena vrijednost izlaza

Osnovom odabrane funkcije cilja vrši se promjena koeficijenata težina primjenom nekog od algoritama nelinearnog programiranja. Izraz za računanje promijenjenog parametra učenja ϑ

$$\vartheta(n+1) = \vartheta(n) + \Delta\vartheta(n) \tag{2.17}$$

Gdje je

n – trenutni korak učenja
$\Delta\vartheta(n)$ – veličina promjene parametara učenja (za izlazni sloj $\vartheta = w$, a za skriveni sloj $\vartheta = v$)
$\vartheta(n+1)$ – nova vrijednost parametra učenja

Pogrešku $E(\vartheta)$ moguće je aproksimirati s prva dva člana taylorovog reda

$$E(\vartheta + \Delta\vartheta) \approx E(\vartheta) + \Delta E(\vartheta) \tag{2.18}$$

$$\Delta E(\vartheta) = \Delta\vartheta^T \, \nabla E(\vartheta) \tag{2.19}$$

$$\nabla E(\vartheta) = \frac{\partial E(\vartheta)}{\partial \vartheta} \tag{2.20}$$

Gdje je

$\nabla E(\vartheta)$ – gradijent pogreške

Kako bi se pogreška smanjivala maksimalnim mogućim iznosom potrebno je odrediti $\Delta\vartheta$ za koji promjena pogreške učenja $\Delta E(\vartheta)$ poprima najveći negativni iznos. To se postiže uvjetom

$$\Delta\vartheta = -\eta \nabla E(\vartheta) \tag{2.21}$$

Gdje je

η – koeficijent brzine učenja

Koeficijent brzine učenja određuje učitelj, a njegova vrijednost najčešće se nalazi između 10^-3 i 10 [6]. Uvrštavanjem (2.21) u (2.17) dobiva se izraz

$$\vartheta(n+1) = \vartheta(n) - \eta\nabla E(\vartheta(n)) \tag{2.22}$$

Ovim izrazom definiran je algoritam povratnog prostiranja pogreške (engl. *error back-propagation algorithm*). Najveći nedostatak ovakvog algoritma je veliki broj koraka učenja (iteracija), kako bi se ubrzao proces učenja te smanjio broj iteracija koristi se modificirani algoritam poznat pod nazivom *momentum* koji je dan izrazom (2.25).

Kod učenja po uzorku promjene parametara učenja ϑ iterativno se provode za svaki ulazno-izlazni par iz skupa učenja, dok se kod učenja po skupu promjene parametara učenja provode za cijeli skup učenja prema sljedećim izrazima

$$\Delta\vartheta = -\eta\sum_{i=1}^{n}\nabla E(\vartheta) \tag{2.23}$$

$$\vartheta(n+1) = \vartheta(n) - \eta\sum_{i=1}^{n}\nabla E(\vartheta) \tag{2.24}$$

2.4 Ubrzanje iterativnog učenja

S obzirom da algoritam učenja sa povratnim prostiranjem pogreške karakterizira velik broj potrebnih koraka učenja uvode se modifikacije algoritma danog (2.22) uvođenjem momentuma (zamaha) prvog i drugog reda.

2.4.1 Momentum prvog reda

Dodavanjem člana koji ovisi o promjeni parametara učenja iz prethodne iteracije u jednadžbu algoritma učenja sa povratnim prostiranjem pogreške (2.22) proizlazi izraz za momentum [6] prvog reda.

$$\Delta\vartheta(n) = -\eta\nabla E(\vartheta(n)) + \alpha\Delta\vartheta(n-1) \tag{2.25}$$

Gdje je:

n – trenutna promjena parametara učenja
$(n-1)$ – prethodna promjena parametara učenja
α – koeficijent momentuma

Vrijednost koeficijenta momentuma α određuje učitelj, a njegov iznos kreće se između 0.1 i 0.9 . Korištenjem momentuma prvog reda brzina učenja može se povećati i do 10 puta. Konačni oblik, uvrštavanjem (2.25)) u (2.17), za promjenu parametara učenja je

$$\vartheta(n+1) = \vartheta(n) - \eta\nabla E(\vartheta(n)) + \alpha\Delta\vartheta(n-1) \tag{2.26}$$

2.4.2 Momentum drugog reda

Analogno momentumu prvog reda moguće je i uvesti momentum drugog reda tj. uvesti član koji ovisi o promjeni parametara učenja i drugog prethodnog koraka $(n - 2)$. Momentum drugog reda rjeđe se koristi zbog mogućnosti usporavanja brzine učenja, stoga je dobro uzeti okvirnu vrijednost momentuma [5] drugog reda β u relaciji sa momentumom prvog reda.

$$\beta = \frac{\alpha - 1}{3} \tag{2.27}$$

Izraz za promjenu parametara učenja sa momentumom drugog reda (uključujući i momentum prvog reda) je

$$\Delta\vartheta(n+1) = -\eta\nabla E(\vartheta(n)) + \alpha\Delta\vartheta(n-1) + \beta\Delta\vartheta(n-2) \tag{2.28}$$

Dok je konačni oblik algoritma učenja

$$\vartheta(n+1) = \vartheta(n) - \eta\nabla E(\vartheta(n)) + \alpha\Delta\vartheta(n-1) + \beta\Delta\vartheta(n-2) \tag{2.29}$$

2.5 Promjena parametara učenja

Prilikom povratne faze učenja parametri učenja mijenjaju se prema algoritmu (2.22), a u slučaju korištenja algoritma učenja sa momentumom (2.26) ili (2.29). Kako bi mogli izračunati promjenu težina svakog sloja neuronske mreže potrebno je izvesti izraze sa svaki sloj.

2.5.1 Promjena težina izlaznog sloja

Kod povratne faze učenja promjena parametara učenja odvija se unazad, od izlaznog prema ulaznom sloju mreže. Promjena težinskih faktora veza ($\mathbf{w}$) između izlaznog i skrivenoga sloja provodi se prema (2.26) izrazom

$$\mathbf{w}_{kj}(n+1) = \mathbf{w}_{kj}(n) - \eta\nabla E(\mathbf{w}_{kj}(n)) + \alpha\Delta\mathbf{w}_{kj}(n-1) \tag{2.30}$$

Gradijent pogreške ∇E za težine $\mathbf{w}_{kj}$ računa se prema (2.20).

$$\nabla E(n) = \frac{\partial E(n)}{\partial \mathbf{w}_{kj}} \tag{2.31}$$

Problem određivanja pripadajućeg gradijenta pogreške lako se rješava uzastopnim parcijalnim derivacijama

$$\frac{\partial E(n)}{\partial \mathbf{w}_{kj}} = \frac{\partial E(n)}{\partial O_k}\frac{\partial O_k}{\partial net_{Ok}}\frac{\partial net_{Ok}}{\partial \mathbf{w}_{kj}} \tag{2.32}$$

Izrazi za pojedine parcijalne derivacije iz (2.32) su

$$\frac{\partial E(n)}{\partial O_k} = -(d_k - O_k) \tag{2.33}$$

$$\frac{\partial O_k}{\partial net_{Ok}} = \gamma_k' = 1 \tag{2.34}$$

Pomoću izraza (2.14) možemo odrediti posljednju parcijalnu derivaciju

$$\frac{\partial net_{Ok}}{\partial \mathbf{w}_{kj}} = y_j \tag{2.35}$$

Uzmemo li u obzir da je karakteristična vrijednost [6] algoritma povratnog prostiranja greške prema definiciji

$$\delta = -\frac{\partial E(N)}{\partial net_{Ok}} \tag{2.36}$$

onda možemo odrediti δ_{Ok} iz (2.33)

$$\delta_{Ok} = d_k - O_k \tag{2.37}$$

Uvrštavanjem (2.33), (2.34), (2.35) i (2.37) u (2.32) izraz za $\nabla E(n)$ na kraju izgleda

$$\nabla E(n) = \frac{\partial E(n)}{\partial \mathbf{w}_{kj}} = -(d_k - O_k) \cdot y_j = -\delta_{Ok} \cdot y_j \tag{2.38}$$

Uvrštavanjem (2.38) u izraz (2.30) dobiva se izraz za algoritam promjene težina izlaznog sloja

$$\mathbf{w}_{kj}(n+1) = \mathbf{w}_{kj}(n) + \eta\delta_{Ok}y_j + \alpha\Delta\mathbf{w}_{kj}(n-1) \tag{2.39}$$

2.5.2 Promjena težina skrivenoga sloja

Nakon promjene težina $\mathbf{w}_{kj}$ izlaznog sloja moguće je odrediti promjene težina skrivenoga sloja $\mathbf{v}_{ij}$. Isto kao i kod promjena težina izlaznog sloja, promjena težina skrivenoga sloja provodi se prema izrazu (2.26).

$$\mathbf{v}_{kj}(n+1) = \mathbf{v}_{kj}(n) - \eta\nabla E(\mathbf{v}_{kj}(n)) + \alpha\Delta\mathbf{v}_{kj}(n-1) \tag{2.40}$$

Analogno izrazu gradijenta greške u izlaznom sloju provode se uzastopne parcijalne derivacije za gradijent greške $\nabla E(n)$ u sakrivenom sloju.

$$\frac{\partial E(n)}{\partial \mathbf{v}_{ij}} = \frac{\partial E(n)}{\partial y_j}\frac{\partial y_j}{\partial net_{Hj}}\frac{\partial net_{Hj}}{\partial \mathbf{v}_{ij}} \tag{2.41}$$

Prema slici 2.5 vidljivo je da na svaku težinu skrivenoga sloja utječu svi neuroni izlaznog sloja.

$$\frac{\partial E(n)}{\partial y_j} = \sum_{k=1}^{K}\left(\frac{\partial E(n)}{\partial O_k}\frac{\partial O_k}{\partial net_{Ok}}\frac{\partial net_{Ok}}{\partial y_j}\right) \tag{2.42}$$

Pri čemu su pojedine parcijalne derivacije u izrazu (2.42)

$$\frac{\partial E(n)}{\partial O_k} = -(d_k - O_k) \tag{2.43}$$

$$\frac{\partial O_k}{\partial net_{Ok}} = 1 \tag{2.44}$$

$$\frac{\partial net_{Ok}}{\partial y_j} = \mathbf{w}_{kj}, \qquad k = 1, 2, ..., K;\ j = 1, 2, ..., J-1 \tag{2.45}$$

Uvrštavanjem izraza (2.43), (2.44), (2.45) u izraz (2.42) te daljnjim uvrštavanjem (2.37) dobiva se

$$\frac{\partial E(n)}{\partial y_j} = -\sum_{k=1}^{K}(d_k - O_k)\mathbf{w}_{kj} = -\sum_{k=1}^{K}\delta_{Ok}\mathbf{w}_{kj} \tag{2.46}$$

Druga i treća parcijalna derivacija iz izraza (2.42) određene su izrazima (2.11) i (2.12).

$$\frac{\partial y_j}{\partial net_{Hj}} = \gamma_j' = \frac{1}{2}\left(1 - y_j^2\right) \tag{2.47}$$

$$\frac{\partial net_{Hj}}{\partial \mathbf{v}_{ji}} = Z_i \tag{2.48}$$

Uvrštavanjem izraza (2.46), (2.47) i (2.48) nazad u izraz (2.42) dobivamo izraz za $\nabla E(n)$.

$$\nabla E(n) = -\frac{1}{2}\left(1 - y_j^2\right) Z_i \left(\sum_{k=1}^{K}\delta_{Ok}\mathbf{w}_{kj}\right) \tag{2.49}$$

Uvrštavanjem (2.49) u početni izraz (2.40) dobivamo konačni izraz za algoritam promjene težinskih koeficijenata sakrivenog sloja.

$$\mathbf{v}_{kj}(n+1) = \mathbf{v}_{kj}(n) + \frac{1}{2}\eta\left(1 - y_j^2\right) Z_i \left(\sum_{k=1}^{K}\delta_{Ok}\mathbf{w}_{kj}\right) + \alpha\Delta\mathbf{v}_{kj}(n-1) \tag{2.50}$$

U slučaju odabira neke druge aktivacijske funkcije izraz (2.47) će poprimiti drugačiji oblik. Za sinusnu aktivacijsku funkciju (2.9) izraz postaje

$$\frac{\partial y_j}{\partial net_{Hj}} = \gamma_j' = (\sin(net_{Hj}))' = \cos(net_{Hj}) \tag{2.51}$$

Odaberemo li gaussovu aktivacijsku funkciju (2.10) izraz postaje

$$\frac{\partial y_j}{\partial net_{Hj}} = \gamma_j' = \left(e^{-\frac{1}{2}\left(\frac{net - c_j}{\sigma_j}\right)^2}\right)' = e^{-\frac{1}{2}\left(\frac{net_{Hj} - c_j}{\sigma_j}\right)} \cdot \frac{c_j - net_{Hj}}{\sigma_j^2} \tag{2.52}$$

Obzirom da gaussova aktivacijska funkcija uvodi dodatne parametre učenja c i σ potrebno je izvesti izraze za njihovo učenje. Promjena parametara gaussove aktivacijske funkcije računa se prema istome izrazu (2.26) kao i promjena težinskih koeficijenata veza ($\mathbf{w}$ i $\mathbf{v}$). Izraz za ∇E pojedinog parametra dobiva se uzastopnim parcijalnim derivacijama.

$$\nabla E(n) = \frac{\partial E}{\partial c_j} = \frac{\partial E}{\partial y_j}\frac{\partial y_j}{\partial c_j} \tag{2.53}$$

$$\nabla E(n) = \frac{\partial E}{\partial \sigma_j} = \frac{\partial E}{\partial y_j}\frac{\partial y_j}{\partial \sigma_j} \tag{2.54}$$

(2.55)

$$\frac{\partial y_j}{\partial c_j} = e^{-\frac{1}{2}\left(\frac{net_{Hj} - c_j}{\sigma_j}\right)^2} \cdot \frac{net_{Hj} - c_j}{{\sigma_j}^2} \tag{2.56}$$

$$\frac{\partial y_j}{\partial \sigma_j} = e^{-\frac{1}{2}\left(\frac{net_{Hj} - c_j}{\sigma_j}\right)^2} \cdot \frac{(net_{Hj} - c_j)^2}{{\sigma_j}^3} \tag{2.57}$$

2.6 Ocjena točnosti algoritma učenja

Kako bi ocijenili točnost neuronske mreže tokom i nakon učenja za određeni zadatak potrebno je odrediti mjeru točnosti u usporedbi sa traženim vrijednostima izlaza mreže. Za mjeru točnosti moguće je koristiti srednju kvadratnu pogrešku, korijen srednje kvadratne pogreške te normalizirani korijen srednje kvadratne pogreške (NRMS).

Izraz za srednju kvadratnu pogrešku (MS) glasi

$$MS = \frac{\sum_{n=1}^{N} (d_n - O_n)^2}{N} \tag{2.58}$$

gdje je n broj pojedinog izlaza, a N ukupan broj izlaza.
Izraz za korijen srednje kvadratne pogreške glasi

$$RMS = \sqrt{MS} = \sqrt{\frac{\sum_{n=1}^{N} (d_n - O_n)^2}{N}} \tag{2.59}$$

Normalizirani korijen srednje kvadratne pogreške, NRMS (engl. *normalized root mean square error*), računa se prema izrazu

$$NRMS = \frac{RMS}{\sigma_{d_n}} = \frac{\sqrt{\frac{\sum_{n=1}^{N} (d_n - O_n)^2}{N}}}{\sigma_{d_n}} \tag{2.60}$$

gdje je σ_{d_n} standardno odstupanje koje se računa prema izrazu

$$\sigma_{d_n} = \sqrt{\frac{1}{N}\sum_{n=1}^{N} \left(d_n - \bar{d}\right)^2} \tag{2.61}$$

$$\bar{d} = \frac{\sum_{n=1}^{N} d_n}{N} \tag{2.62}$$

Normalizirani korijen srednje kvadratne pogreške karakterizira bezdimenzionalnost koja osigurava neovisnost mjere o dimenzijama učenih veličina te omogućuje usporedbu izvedenih algoritama učenja s drugim algoritmima.

3 Implementacija u MATLAB-u

Programski paket MATLAB, (**mat**rix **lab**oratory) **mat**rični **lab**oratorij, je programski jezik visoke razine te skup alata koji omogućavaju brzo i korisniku jednostavno računanje matematički složenih zadataka. Interaktivno okruženje programskog paketa MATLAB korisniku omogućuje numeričko i matrično računanje, programiranje te vizualizaciju dobivenih rezultata. Zbog brzog i jednostavnog rukovanja matricama te već ugrađenim grafičkim prikazom dobivenih rezultata MATLAB predstavlja izvrsno programsko rješenje za implementaciju umjetnih neuronskih mreža te algoritama za učenje istih.

3.1 Matrični oblik neuronskih mreža

Radi bolje implementacije algoritama učenja opisanih u poglavljima 2.3.1 i 2.3.2 potrebno je dobivene izraze prilagoditi matričnom zapisu koji je povoljan za korištenje u MATLAB-u.
Izraze (2.11) i (2.14) možemo zapisati kao umnožak matrica težina i ulaza neuronske mreže odnosno izlaza neurona sakrivenog sloja.

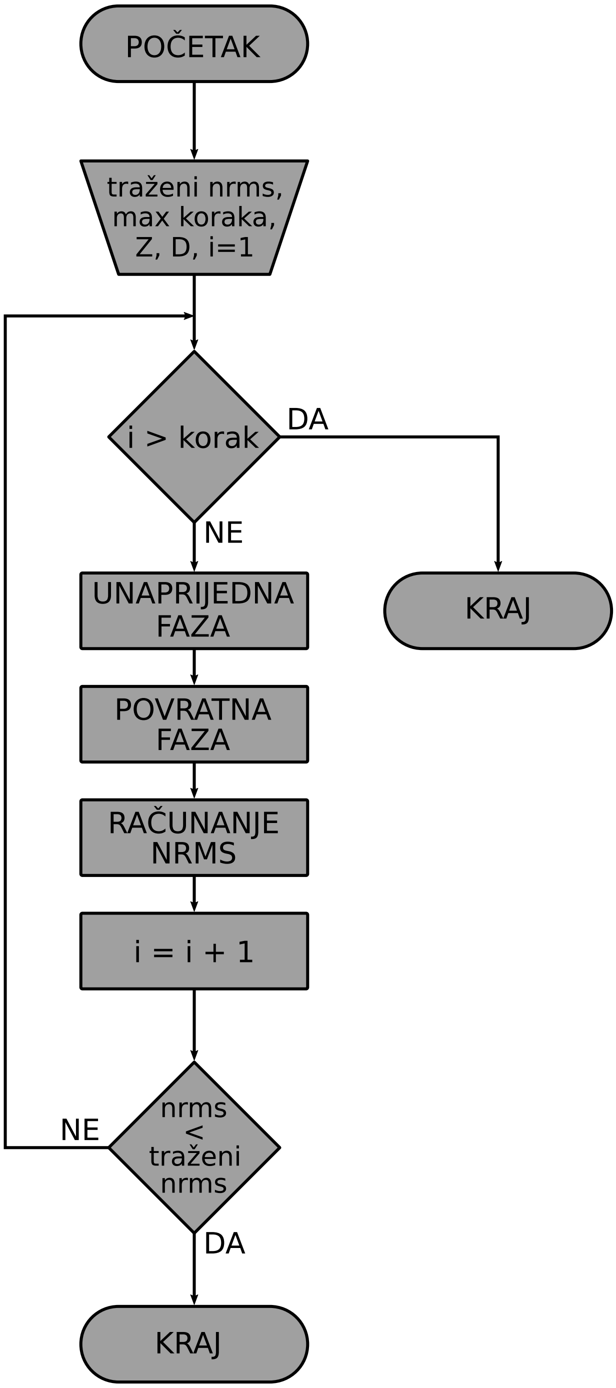
$$\begin{aligned} net_{Hj} &= \sum_{i=1}^{I} v_{ji} \cdot Z_i \\ \mathbf{net_H} &= \mathbf{V} \cdot \mathbf{Z}^T \end{aligned} \tag{3.1}$$

gdje je $\mathbf{V}$ matrica težinskih koeficijenata skrivenoga sloja dimenzija $[J-1 \times I]$, a $\mathbf{Z}$ matrica ulaza dimenzija $[N \times I]$ dok je matrica $\mathbf{net_H}$ dimenzija $[J \times N]$. N je broj uzoraka ulaznog skupa podataka, a u slučaju da je $N = 1$ matrice $\mathbf{Z}$ i $\mathbf{net_H}$ su vektori.

$$\begin{aligned} net_{Ok} &= \sum_{j=1}^{J} w_{kj} \cdot y_j \\ \mathbf{net_O} &= \mathbf{W} \cdot \mathbf{Y} \end{aligned} \tag{3.2}$$

gdje je $\mathbf{W}$ matrica težinskih koeficijenata izlaznog sloja dimenzija $[K \times J]$, a $\mathbf{Y}$ matrica izlaza neurona skrivenoga sloja dimenzija $[J \times N]$. Obzirom da se matrica $\mathbf{Y}$ formira primjenom aktivacijske funkcije γ_j na matricu $\mathbf{net_H}$ potrebno je osigurati da se funkcija γ_j primjenjuje po svakom elementu pojedinačno. Važno je pripaziti da se prilikom formiranja matrica $\mathbf{Z}$ i $\mathbf{Y}$ u njih uključi bias.

3.2 Algoritam učenja



Slika 3.1: Dijagram toka algoritma učenja

Slika 3.1 prikazuje pojednostavljeni dijagram toka algoritma učenja. Glavna petlja programa izvedena je s mogućnosti prekidanja izvođenja glavnog algoritma učenja ukoliko se postigne maksimalni željeni broj koraka ili postigne zadovoljavajuća vrijednost NRMS-a.

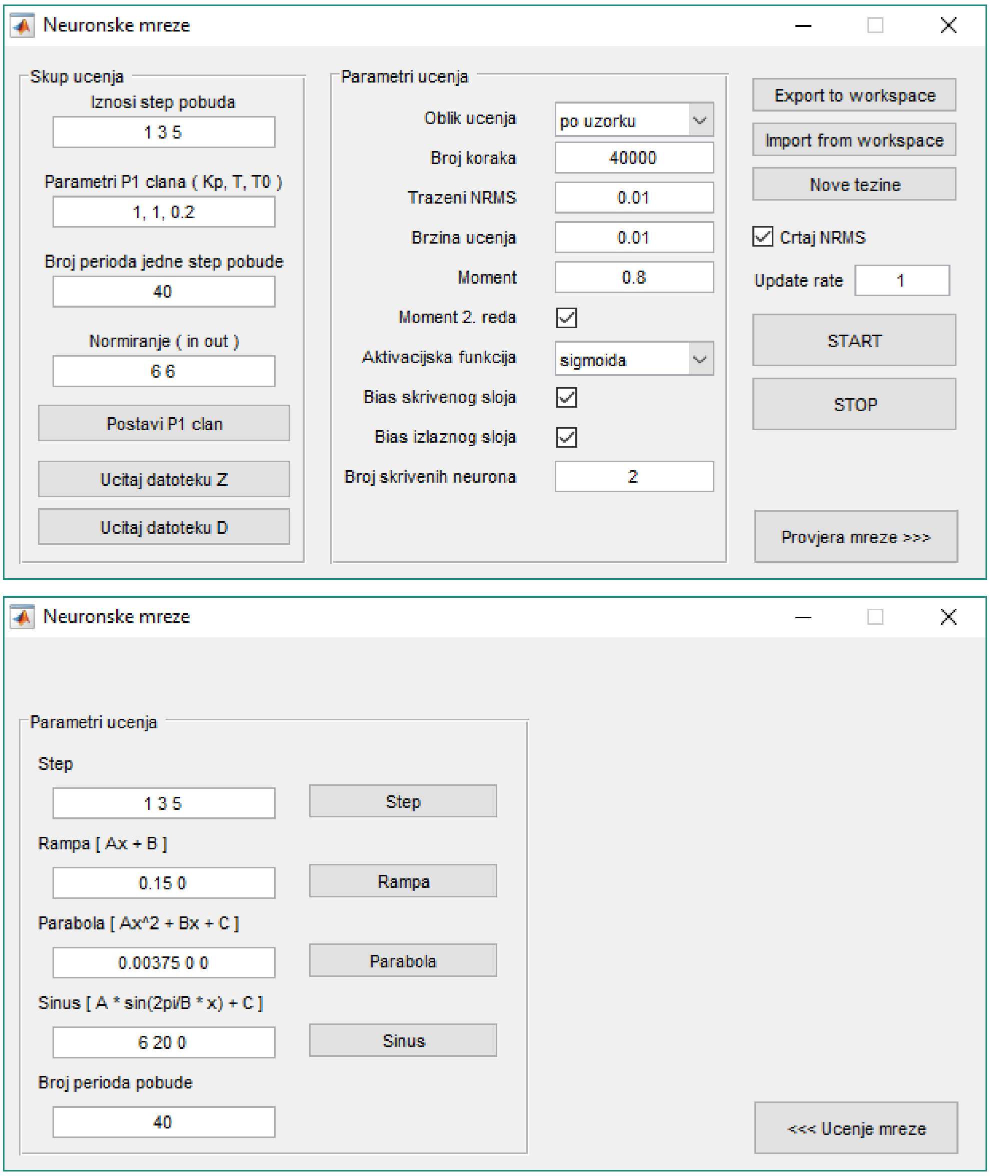
3.3 Grafičko sučelje

Radi jednostavnijeg korištenja programa u MATLAB-u je izrađeno grafičko sučelje koje je vidljivo na slici 3.2. Sučelje je podijeljeno u stupce koji odgovaraju pojedinim fazama postavljanja ulaznih parametara algoritma učenja.

Skup učenja
Omogućuje generiranje skupa učenja za zadani zadatak. Pritiskom gumba "Postavi P1 clan" programski se generira skup učenja za učenje mreže odziva diskretnog P1 člana, dok gumbima "Ucitaj datoteku Z" i "Ucitaj datoteku D" možemo učitati skupove za učenje iz datoteka.
Vrijednosti polja "Parametri P1 clana (Kp, T, T0)" odgovaraju slično imenovanim varijablama korištenim u jednadžbi (4.2) dok su njihove početne vrijednosti dane u (4.3).
Vrijednosti u polju "Iznosi step pobuda" određuju broj i iznose step pobuda kojima se pobuđuje diskretni P1 član zadan jednadžbom (4.2) nakon čega se dobivene vrijednosti postavljaju kao skup za učenje neuronske mreže. Polje "Broj perioda jedne step pobude" određuje broj perioda trajanja jedne step pobude dane u polju "Iznosi step pobuda".
Vrijednosti u polju "Normiranje (in out)" odgovaraju vrijednostima zadanima u (4.5).

Parametri učenja
Daje kontrolu nad parametrima algoritma učenja poput oblika učenja, maksimalnog broja koraka, traženog NRMS-a, koeficijenta brzine učenja η, koeficijenta momenta prvog reda α, primjene momenta drugog reda, aktivacijske funkcije te broja skrivenih neurona.

Testiranje
Omogućava brzu i jednostavnu realizaciju faze testiranja mreže nakon učenja. Testiranje mreže se provodi kako je prikazano slikom 4.5, dok pobudne funkcije rampe, parabole i sinusne funkcije odgovaraju izrazima navedenima u (4.11), (4.12) i (4.13). Polja "Step" i "Broj perioda pobude" analogna su poljima korištenima za postavljanje skupa učenja.



Slika 3.2: Grafičko sučelje

4 Učenje ponašanja nelinearnog i linearnog dinamičkog člana prvog reda

4.1 Identifikacija linearnog dinamičkog sustava

Na primjeru učenja ponašanja jednostavnog linearnog dinamičkog člana prvog reda (P1 član) treba pokazati generalizacijsko svojstvo unaprijednih neuronskih mreža te usporediti učenje i testiranje naučenih neuronskih mreža s različitim aktivacijskim funkcijama i različitim algoritmima učenja.

4.1.1 Opis problema

Dinamiku P1 člana opisuje diferencijalna jednadžba prvog reda

$$T\ddot{x}(t) + x(t) = K_p u(t) \tag{4.1}$$

gdje je T vremenska konstanta, a K_p konstanta pojačanja sustava. U danom obliku jednadžbu (4.1) teško je koristiti kao izvor informacija za učenje diskretne neuronske mreže, stoga je izvršena diskretizacija [7] [8].

$$x(n) = \frac{T_0}{T + T_0}\left[\frac{T}{T_0}x(n-1) + K_p u(n)\right] \tag{4.2}$$

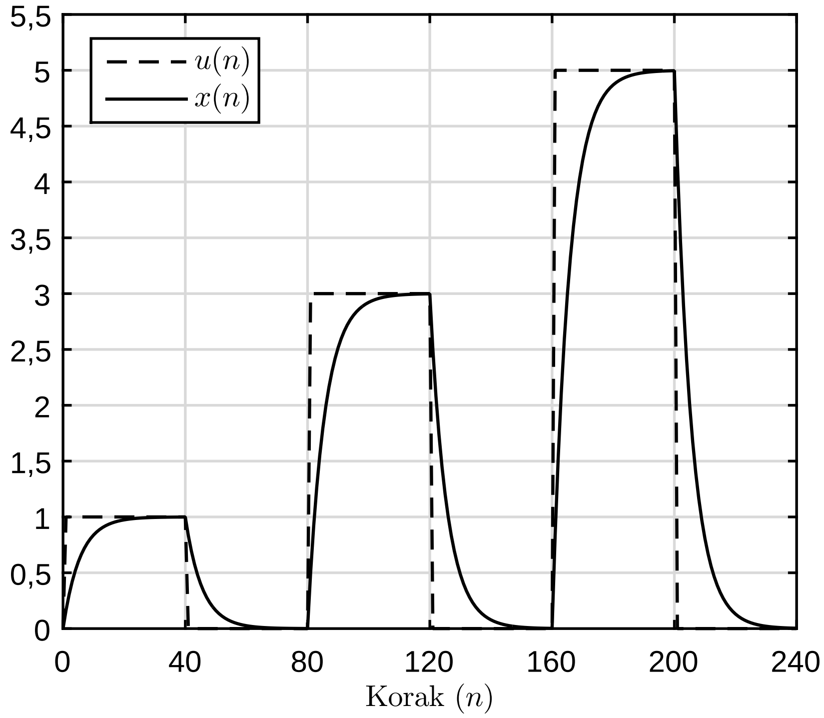
gdje $x(n)$ i $u(n)$ označavaju trenutnu vrijednost izlazne i ulazne vrijednosti, $x(n-1)$ je vrijednost prošle vrijednosti izlaza, a T_0 predstavlja vrijednost perioda uzorkovanja.
Simulacije kojima su dobivene vrijednosti za učenje i testiranje neuronskih mreža u ovom poglavlju obavljene su sa istim parametrima:

$$\begin{aligned} T &= 1s \\ T_0 &= 0.2s \\ K_p &= 1 \end{aligned} \tag{4.3}$$

Za pobudu P1 člana koristiti ćemo odskočnu funkciju sa iznosima 1, 3 i 5 dok će svaki pojedini step biti u trajanju od 40 uzoraka. Konačni skup za pobudu sadrži 240 točaka te prati sljedeći oblik:

$$\begin{aligned} u(n) &= 1, & 0 \leq n &\leq 39 \\ u(n) &= 0, & 40 \leq n &\leq 79 \\ u(n) &= 3, & 80 \leq n &\leq 119 \\ u(n) &= 0, & 120 \leq n &\leq 159 \\ u(n) &= 5, & 160 \leq n &\leq 199 \\ u(n) &= 0, & 200 \leq n &\leq 239 \end{aligned} \tag{4.4}$$

Odziv linearnog sustava (P1 član) na pobudnu funkciju zadanu u (4.4) grafički je prikazan na slici 4.1.

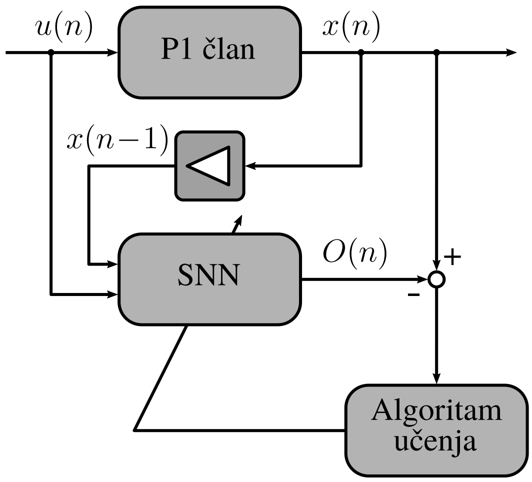


Slika 4.1: Skup učenja P1 člana prema (4.4)

4.1.2 Postupak učenja

Topologija

Obzirom da se radi o opisivanju ponašanja *dinamičkog* člana pomoću *statičke* neuronske mreže potrebno je dobro odabrati topologiju mreže te prilagoditi okolinu mreže kako bi učenje i testiranje bilo uspješno. Dodavanjem dodatnog broja ulaza neuronske mreže ovisno o redu sustava koji pokušavamo učiti te dodavanjem vanjske povratne veze prethodne vrijednosti izlaza $O(n-1)$ na ulaz mreže postižemo mogućnost opisivanja ponašanja dinamičkog člana putem statičke neuronske mreže. Na slici 4.2 grafički je prikazan proces učenja ponašanja dinamičkog člana prvog reda.



Slika 4.2: Učenje ponašanja P1 člana

Nakon što smo odredili da zadatak zahtijeva dva ulaza i jedan izlaz potrebno je još odrediti i broj neurona skrivenoga sloja. Radi usporedbe korištene su mreže s 2 i 4 neurona skrivenoga sloja, stoga su topologije učenih mreža 2-2-1 i 2-4-1.

Normiranje

Kako bi osigurali da su svi neuroni u mreži imaju jednak utjecaj na ishod učenja potrebno je normirati vrijednosti ulaza i izlaza prilikom učenja, ali i prilikom testiranja mreže. Kako bi normirali skup ulazno-izlaznih vrijednosti za učenje (matrice $\mathbf{Z}$ i $\mathbf{D}$) potrebno je vrijednosti u matricama podijeliti koeficijentom normiranja. Prema izboru normiranje je provedeno s vrijednosti $norm = 6$.

$$\begin{aligned} \mathbf{Z}^* &= \frac{1}{norm} \cdot \mathbf{Z} = \frac{1}{6} \cdot \mathbf{Z} \\ \mathbf{D}^* &= \frac{1}{norm} \cdot \mathbf{D} = \frac{1}{6} \cdot \mathbf{D} \end{aligned} \tag{4.5}$$

Učenje

Radi usporedbe brzine učenja, učenje je provedeno sa istim početnim težinama. Naravno početne težine za topologiju mreže 2-2-1 razlikuju se od onih za topologiju mreže 2-4-1 zato što dimenzije matrica $\mathbf{V}$ i $\mathbf{W}$ ovise o broju skrivenih neurona. Vektori $\mathbf{C}$ i $\mathbf{S}$ koji su definirani jednadžbama (2.56) i (2.57) koriste se samo u slučaju gaussove aktivacijske funkcije (2.10).

Početne težine za topologiju 2-2-1:

$$\mathbf{V} = \begin{bmatrix} 0{,}811\,063 & -0{,}691\,883 & 0{,}874\,183 \\ -0{,}165\,358 & 0{,}079\,999 & 0{,}321\,910 \end{bmatrix}$$

$$\mathbf{W} = \begin{bmatrix} -0{,}210\,685 & -0{,}482\,019 & 0{,}695\,824 \end{bmatrix} \tag{4.6}$$

$$\mathbf{C} = \begin{bmatrix} 0{,}890\,112 \\ -0{,}246\,000 \end{bmatrix} \qquad \mathbf{S} = \begin{bmatrix} -0{,}865\,440 \\ -0{,}636\,836 \end{bmatrix}$$

Početne težine za topologiju 2-4-1:

$$\mathbf{V} = \begin{bmatrix} -0{,}242\,346 & -0{,}464\,551 & -0{,}834\,617 \\ 0{,}356\,726 & 0{,}733\,821 & 0{,}477\,647 \\ 0{,}031\,850 & -0{,}252\,490 & -0{,}255\,864 \\ 0{,}956\,111 & -0{,}726\,739 & -0{,}238\,967 \end{bmatrix}$$

$$\mathbf{W} = \begin{bmatrix} -0{,}395\,652 & -0{,}410\,949 & -0{,}480\,373 & -0{,}073\,050 & 0{,}975\,705 \end{bmatrix} \tag{4.7}$$

$$\mathbf{C} = \begin{bmatrix} -0{,}229\,848 \\ 0{,}886\,468 \\ -0{,}665\,998 \\ 0{,}850\,008 \end{bmatrix} \qquad \mathbf{S} = \begin{bmatrix} 0{,}508\,675 \\ 0{,}713\,199 \\ 0{,}204\,773 \\ -0{,}785\,266 \end{bmatrix}$$

Tijekom učenja po uzorku korišten je koeficijent brzine učenja $\eta = 0{,}08$ dok je kod učenja po skupu $\eta = 0{,}007$. Koeficijenti brzine učenja odabrani su na način da se za zadane početne težine pokuša koristiti $\eta = 0{,}1$ koji se smanjuje sve dok se ne postigne konvergiranje pogreške NRMS. U oba dva slučaja, kod učenja po uzorku i skupu, korišten je koeficijent momentuma $\alpha = 0{,}8$ te je korišten i momentum drugog reda prema (2.27). Sva učenja su provedena za broj koraka $n = 40000$ dok je uvjet za NRMS postavljen na jako malu vrijednost reda 10^{-8}.

Rezultati učenja

Tablica 4.1 prikazuje rezultate konačne vrijednosti NRMS i vremena učenja neuronskih mreža prema zadanim parametrima nakon 40000 koraka učenja. Učenje je provedeno za sigmoidalnu (2.8), sinusnu (2.9) i gaussovu (2.10) aktivacijsku funkciju u dvije topologije.

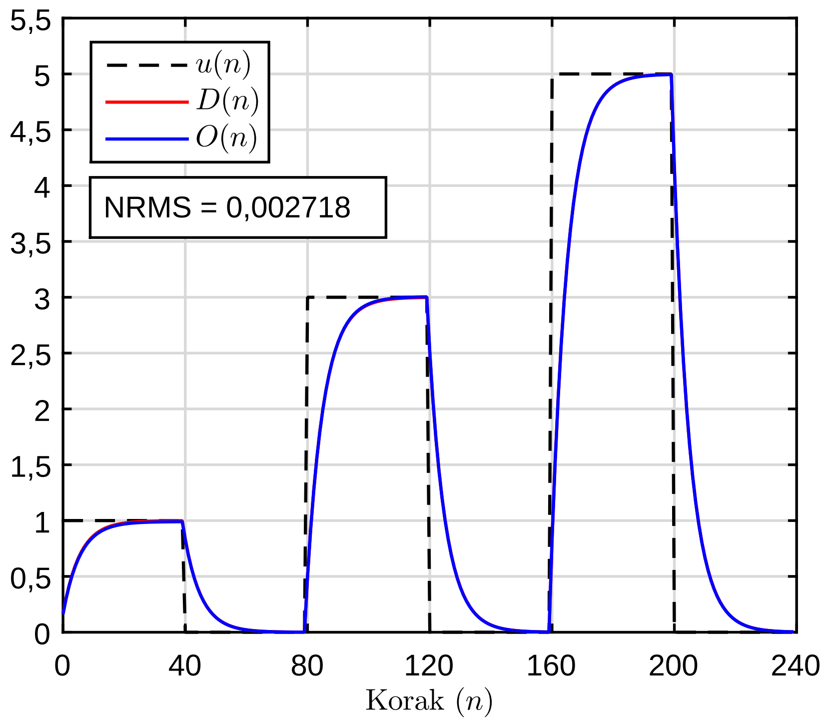
Tablica 4.1: Utjecaj procedure učenja na brzinu učenja

	Učenje po uzorku		Učenje po skupu	
	NRMS	Vrijeme [s]	NRMS	Vrijeme [s]
Topologija 2-2-1				
Sigmoida	0,004410	607,358371	0,003973	103,347406
Sinus	0,003982	596,128636	0,003679	103,535517
Gauss	0,007639	893,121753	0,003209	108,330245
Topologija 2-4-1				
Sigmoida	0,002718	610,576599	0,003568	104,660630
Sinus	0,003583	599,320946	0,017262	103,076595
Gauss	0,004780	904,653061	0,003863	113,779740

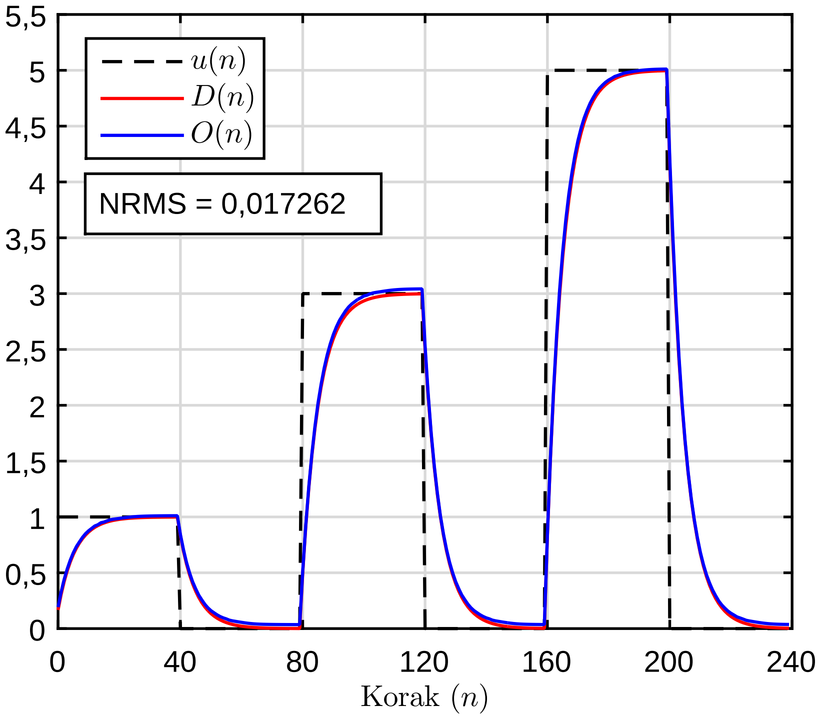
Važno je napomenuti da je vrijeme izvođenja mjereno MATLAB naredbama `tic` i `toc` koje mjere ukupno vrijeme izvođenja tako da izmjereno vrijeme sadrži razinu greške zbog odvijanja drugih procesa na računalu.

Iz tablice 4.1 vidljivo je da je vrijeme potrebno za učenje neuronskih mreža sa gaussovom aktivcijskom funkcijom (2.10) veće nego za druge aktivacijske funkcije što je posljedica dodatnih parametara učenja postavljenih putem matrica $\mathbf{C}$ i $\mathbf{S}$ za koje se učenje provodi prema (2.56) i (2.57). Također je vidljivo dulje vrijeme učenja potrebno kod učenja po uzorku nego kod učenja po skupu.

Slike 4.3 i 4.4 grafički prikazuju razliku između većih i manjih vrijednosti NRMS-a učenih neuronskih mreža. Crna isprekidana linija predstavlja ulaz $u(n)$, crvena linija predstavlja željene vrijednosti izlaza $D(n)$, a plava linija predstavlja dobiveni izlaz iz neuronske mreže $O(n)$. Slika 4.3 prikazuje odziv neuronske mreže sa sigmoidalnom aktivacijskom funkcijom topologije 2-4-1 učene po uzorku, dok slika 4.4 prikazuje odziv neuronske mreže sa sinusnom aktivacijskom funkcijom topologije 2-4-1 učene po skupu.



Slika 4.3: Mreža 2-4-1 (sigmoida) učena po uzorku



Slika 4.4: Mreža 2-4-1 (sinus) učena po skupu

Vrijednosti težina neuronske mreže prikazane na slici 4.3 nakon učenja su:

$$\mathbf{V} = \begin{bmatrix} 0{,}061\,742 & -0{,}697\,362 & -0{,}074\,414 \\ -0{,}097\,389 & 0{,}574\,562 & 0{,}373\,455 \\ 0{,}108\,236 & -0{,}864\,833 & 0{,}563\,281 \\ 1{,}193\,918 & 0{,}055\,100 & -0{,}516\,05 \end{bmatrix} \tag{4.8}$$

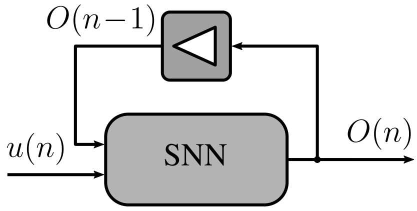
$$\mathbf{W} = \begin{bmatrix} -0{,}819\,875 & 0{,}410\,027 & -1{,}034\,251 & 0{,}451\,050 & 0{,}291\,513 \end{bmatrix}$$

Vrijednosti težina neuronske mreže prikazane na slici 4.4 nakon učenja su:

$$\mathbf{V} = \begin{bmatrix} -1072{,}692\,731 & -1040{,}662\,339 & -6087{,}535\,688 \\ 0{,}160\,316 & 0{,}798\,435 & -792{,}063\,796 \\ 1207{,}020\,908 & 1377{,}012\,541 & 8420{,}202\,154 \\ -935{,}814\,304 & -1173{,}961\,855 & -6204{,}992\,664 \end{bmatrix} \tag{4.9}$$

$$\mathbf{W} = \begin{bmatrix} 0{,}000\,310 & 1{,}067\,028 & -0{,}000\,107 & -0{,}000\,062 & 0{,}403\,949 \end{bmatrix}$$

4.1.3 Testiranje



Slika 4.5: Shema testiranja neuronske mreže

Postupak testiranja prethodno naučenih mreža grafički je prikazan slikom 4.5. Već je tijekom učenja bilo potrebno na jedan od ulaza neuronske mreže dovoditi vrijednost izlaza iz prethodnog koraka, naravno tijekom učenja ta vrijednost je dobivena iz prethodnog stanja izlaza samog P1 člana dok kod testiranja mreže ta vrijednost nije dostupna nego se koristi prethodno stanje izlaza same neuronske mreže. Obzirom da neuronska mreža na izlazu ne daje točno traženu vrijednost (onu zadanu tijekom učenja) nego aproksimaciju željenog izlaza za očekivati je da će generalizacija takve neuronske mreže biti skromnija, ipak u [8] je pokazano da greška koja nastaje opisanim postupkom nije aditivna.

Tablica 4.2 prikazuje vrijednosti NRMS pogreške za prethodno učene mreže u postavu pokazanom slikom 4.5. Korištena je ista pobuda kao i kod učenja (4.4).

Tablica 4.2: NRMS pogreška testiranja mreža pobudom (4.4)

Mreže:	učene po uzorku	učene po skupu
Topologija 2-2-1		
Sigmoida	0,021204	0,020082
Sinus	0,020955	0,018084
Gauss	0,040637	0,015950
Topologija 2-4-1		
Sigmoida	0,014797	0,016584
Sinus	0,019512	0,093192
Gauss	0,023289	0,020142

Kako bi dalje potvrdili generalizacijsko svojstvo neuronskih mreža potrebno je provesti testiranje neuronske mreže sa ulazima van skupa učenja. Za prvi skup ulaznih vrijednosti uzeta je step pobuda, ali sa drugačijim vrijednostima nego u (4.4)

$$\begin{aligned}
u(n) &= 0.5, & 0 \leq &n \leq 39 \\
u(n) &= 0, & 40 \leq &n \leq 79 \\
u(n) &= 2, & 80 \leq &n \leq 119 \\
u(n) &= 0, & 120 \leq &n \leq 159 \\
u(n) &= 4, & 160 \leq &n \leq 199 \\
u(n) &= 0, & 200 \leq &n \leq 239 \\
u(n) &= 6, & 240 \leq &n \leq 279 \\
u(n) &= 0, & 280 \leq &n \leq 319
\end{aligned} \tag{4.10}$$

Također testiranje je provedeno i sa sljedećim pobudama.
Pobudna funkcija rampe:

$$u(n) = A \cdot n + B \qquad A = 0{,}15;\ B = 0 \tag{4.11}$$

Pobudna funkcija parabole:

$$u(n) = A \cdot n^2 + B \cdot n + C \qquad A = 0{,}003\,75;\ B = 0;\ C = 0 \tag{4.12}$$

Sinusna pobudna funkcija:

$$u(n) = A \cdot \sin(\frac{2\pi}{B} \cdot n) + C \qquad A = 6;\ B = 20;\ C = 0 \tag{4.13}$$

Gdje je

$$n = 0, 1, 2, \ldots, 39 \tag{4.14}$$

Tablica 4.3 prikazuje NRMS pogrešku za mreže 2-2-1 topologije, dok tablica 4.4 prikazuje NRMS pogrežku za mreže 2-4-1 topologije.

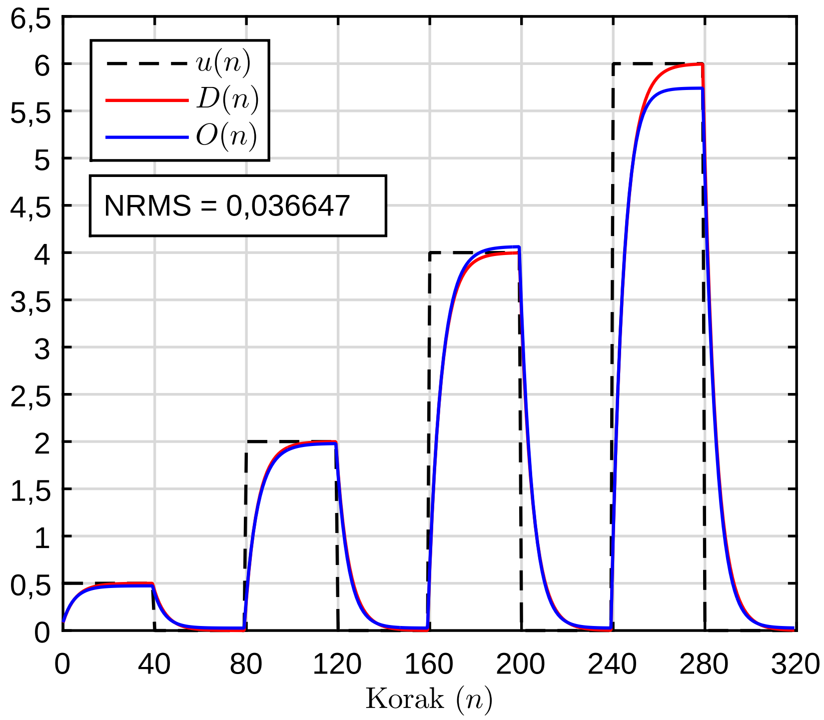
Tablica 4.3: NRMS pogreška testiranja mreža 2-2-1 topologije

Pobuda:	Step	Rampa	Parabola	Sinus
Mreže učene po uzorku				
Sigmoida	0,034115	0,018903	0,021734	0,140184
Sinus	0,039460	0,023977	0,026751	0,197111
Gauss	0,062709	0,045848	0,052490	0,358425
Mreže učene po skupu				
Sigmoida	0,043860	0,020809	0,023159	0,231111
Sinus	0,040966	0,018347	0,020817	0,234445
Gauss	0,033713	0,016248	0,014457	0,064323

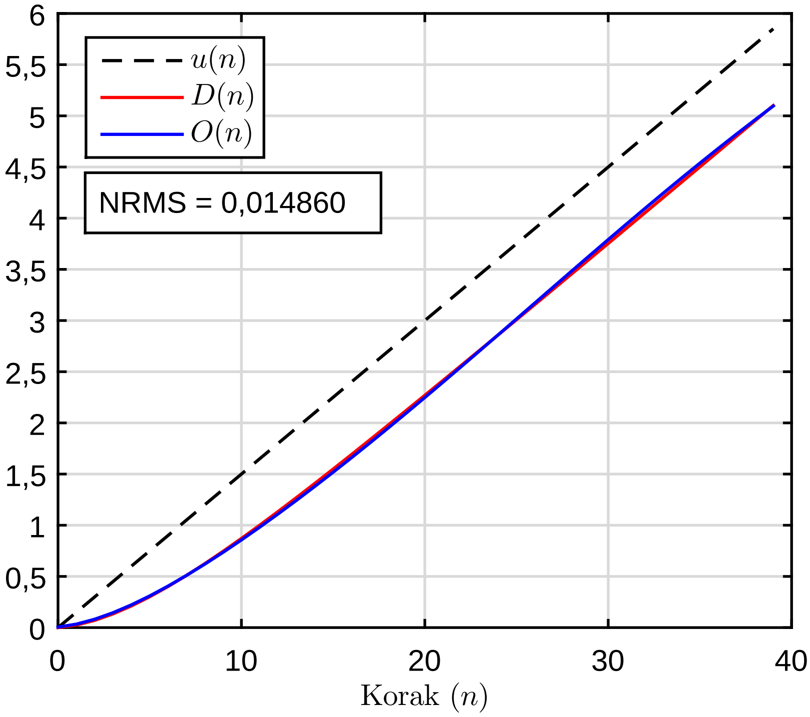
Tablica 4.4: NRMS pogreška testiranja mreža 2-4-1 topologije

Pobuda:	Step	Rampa	Parabola	Sinus
Mreže učene po uzorku				
Sigmoida	0,032643	0,016468	0,018414	0,303867
Sinus	0,044195	0,021886	0,022849	0,510774
Gauss	0,055187	0,030803	0,029254	0,131652
Mreže učene po skupu				
Sigmoida	0,036647	0,014860	0,014677	0,139989
Sinus	0,089128	0,102547	0,100387	0,271830
Gauss	0,039850	0,024796	0,026772	0,277082

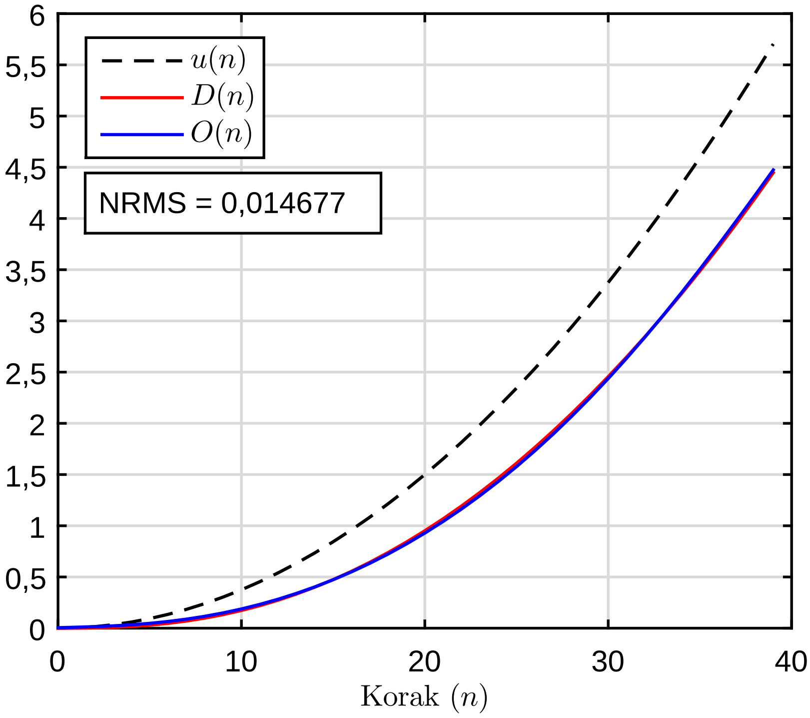
Slike 4.6, 4.7, 4.8 i 4.9 prikazuju odziv neuronske mreže sa sigmoidalnom aktivacijskom funkcijom topologije 2-4-1 koja je učena po skupu na prethodno opisane pobudne funkcije.



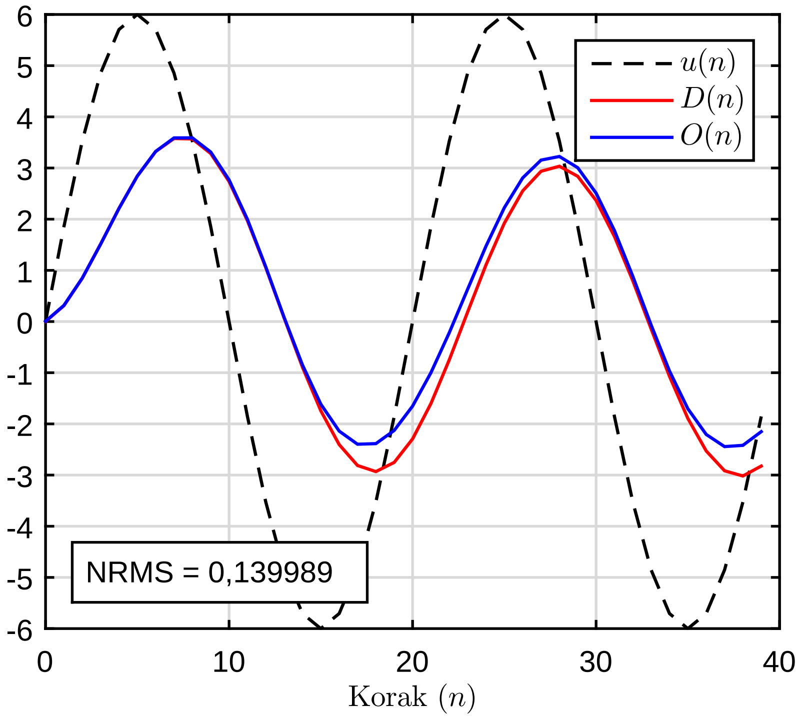
Slika 4.6: Odziv mreže na step



Slika 4.7: Odziv mreže na rampu



Slika 4.8: Odziv mreže na parabolu



Slika 4.9: Odziv mreže na sinusnu funkciju

4.2 Identifikacija nelinearnog dinamičkog sustava

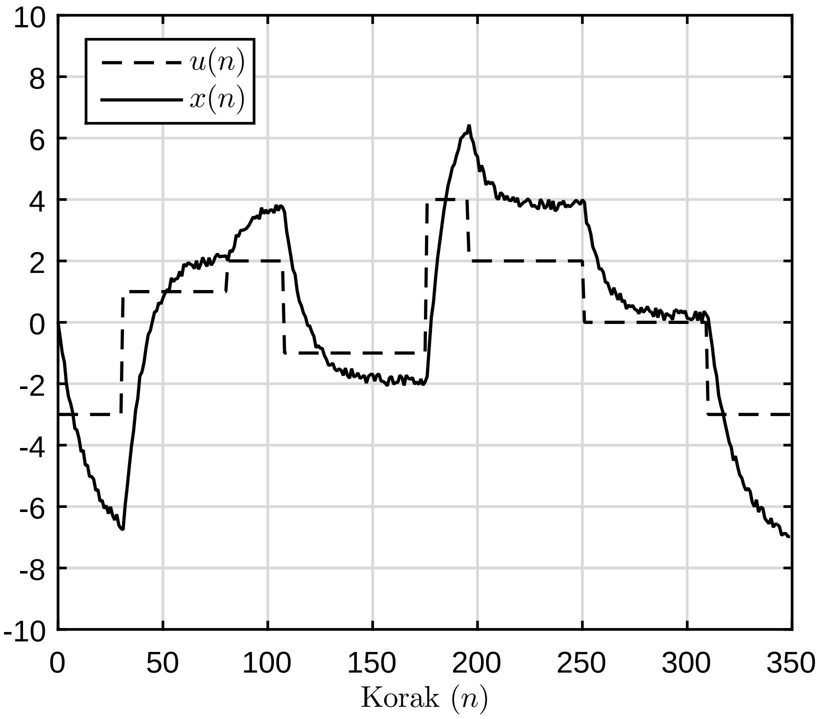
Na primjeru učenja ponašanja nelinearnog dinamičkog člana treba pokazati generalizacijsko svojstvo unaprijednih neuronskih mreža te usporediti učenje i testiranje naučenih neuronskih mreža s različitim aktivacijskim funkcijama i različitim algoritmima učenja.

4.2.1 Opis problema

Nelinearni dinamički sustav opisan je nelinearnom jednadžbom diferencije prvog reda uz period uzorkovanja od 1 sekunde, te vremensku konstantu sustava od 10 sekundi [7].

$$x(n+1) = (0.9 - 0.003 \cdot x(n)) \cdot x(n) + 0.2 \cdot u(n) \tag{4.15}$$

Kako bi mogli provesti identifikaciju ovakvog sustava potrebno je odrediti ulazni signal kojim ćemo pobuditi promatrani sustav. Ulazni signal treba biti takav da osigurava izlazni signal sustava koji sadrži dovoljno informacija o procesu. Kako bi to postigli ulazni signal je nasumičnog karaktera. Radi približenja realnim mjerenjima odziv sustava dodatno je opterećen šumom varijance $0.05 \cdot x_{max}$. Odziv nelinearnog sustava (4.15) na pobudnu funkciju grafički je prikazan na slici 4.10.



Slika 4.10: Skup učenja nelinearnog dinamičkog člana

4.2.2 Postupak učenja

Topologija

Analogno odabranoj topologiji u poglavlju 4.1 i u ovom slučaju je odabrana neuronska mreža s jednim izlazom te dva ulaza od kojih na jedan dovodimo prethodnu vrijednost izlaza sustava kao što je prikazano na slici 4.2. U ovom slučaju smo također koristili mreže s 2 i 4 neurona skrivenoga sloja, stoga su topologije učenih mreža 2-2-1 i 2-4-1.

Normiranje

Kako bi osigurali da su svi neuroni u mreži imaju jednak utjecaj na ishod učenja potrebno je normirati vrijednosti ulaza i izlaza prilikom učenja, ali i prilikom testiranja mreže. Kako bi normirali skup ulazno-izlaznih vrijednosti za učenje (matrice $\mathbf{Z}$ i $\mathbf{D}$) potrebno je vrijednosti u matricama podijeliti koeficijentom normiranja. Prema izboru normiranje je provedeno s vrijednosti $norm = 10$.

$$\begin{aligned} \mathbf{Z}^* &= \frac{1}{norm} \cdot \mathbf{Z} = \frac{1}{10} \cdot \mathbf{Z} \\ \mathbf{D}^* &= \frac{1}{norm} \cdot \mathbf{D} = \frac{1}{10} \cdot \mathbf{D} \end{aligned} \tag{4.16}$$

Učenje

Radi usporedbe brzine učenja, učenje je provedeno sa istim početnim težinama. Naravno početne težine za topologiju mreže 2-2-1 razlikuju se od onih za topologiju mreže 2-4-1 zato što dimenzije matrica $\mathbf{V}$ i $\mathbf{W}$ ovise o broju skrivenih neurona. Vektori $\mathbf{C}$ i $\mathbf{S}$ koji su definirani jednadžbama (2.56) i (2.57) koriste se samo u slučaju gaussove aktivacijske funkcije (2.10).

Početne težine za topologiju 2-2-1:

$$\begin{gathered} \mathbf{V} = \begin{bmatrix} -0{,}677\,541 & -0{,}013\,186 & 0{,}679\,562 \\ -0{,}941\,984 & -0{,}047\,017 & 0{,}737\,877 \end{bmatrix} \\ \mathbf{W} = \begin{bmatrix} -0{,}582\,953 & 0{,}491\,886 & 0{,}633\,854 \end{bmatrix} \\ \mathbf{C} = \begin{bmatrix} -0{,}635\,900 \\ 0{,}944\,801 \end{bmatrix} \qquad \mathbf{S} = \begin{bmatrix} 0{,}743\,344 - 0{,}387\,301 \end{bmatrix} \end{gathered} \tag{4.17}$$

Početne težine za topologiju 2-4-1:

$$\mathbf{V} = \begin{bmatrix} -0{,}714\,568 & -0{,}884\,113 & -0{,}357\,853 \\ -0{,}867\,082 & 0{,}838\,931 & -0{,}694\,838 \\ 0{,}793\,454 & 0{,}759\,671 & 0{,}859\,115 \\ 0{,}706\,898 & 0{,}789\,944 & 0{,}038\,367 \end{bmatrix}$$

$$\mathbf{W} = \begin{bmatrix} 0{,}669\,204 & 0{,}723\,568 & -0{,}101\,851 & -0{,}449\,079 & -0{,}056\,620 \end{bmatrix} \tag{4.18}$$

$$\mathbf{C} = \begin{bmatrix} -0{,}863\,404 \\ -0{,}641\,967 \\ -0{,}602\,513 \\ 0{,}517\,786 \end{bmatrix} \qquad \mathbf{S} = \begin{bmatrix} 0{,}749\,774 \\ -0{,}979\,751 \\ -0{,}407\,544 \\ 0{,}875\,352 \end{bmatrix}$$

Tijekom učenja po uzorku korišten je koeficijent brzine učenja $\eta = 0{,}09$ dok je kod učenja po skupu $\eta = 0{,}005$. Koeficijenti brzine učenja odabrani su na način da se za zadane početne težine pokuša koristiti $\eta = 0{,}1$ koji se smanjuje sve dok se ne postigne konvergiranje pogreške NRMS. U oba dva slučaja, kod učenja po uzorku i skupu, korišten je koeficijent momentuma $\alpha = 0{,}8$ te je korišten i momentum drugog reda prema (2.27). Sva učenja su provedena za broj koraka $n = 20000$ dok je uvjet za NRMS postavljen na jako malu vrijednost reda 10^{-8}.

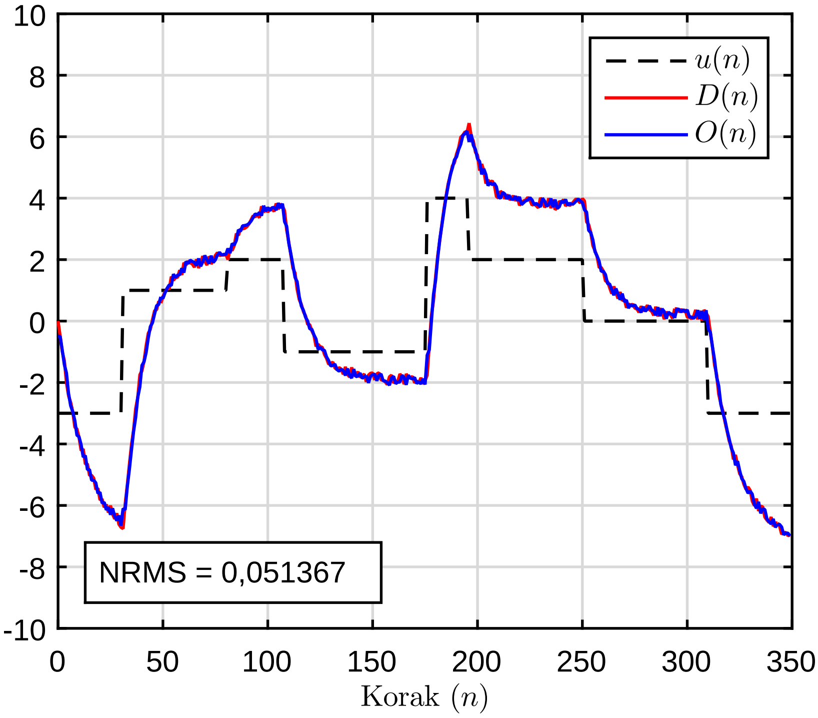
Rezultati učenja

Tablica 4.5 prikazuje rezultate konačne vrijednosti NRMS i vremena učenja neuronskih mreža prema zadanim parametrima nakon 20000 koraka učenja. Učenje je provedeno za sigmoidalnu (2.8), sinusnu (2.9) i gaussovu (2.10) aktivacijsku funkciju u dvije topologije.

Tablica 4.5: Utjecaj procedure učenja na brzinu učenja

	Učenje po uzorku		Učenje po skupu	
	NRMS	Vrijeme [s]	NRMS	Vrijeme [s]
Topologija 2-2-1				
Sigmoida	0,056823	464,483043	0,051367	57,967855
Sinus	0,062491	448,646478	0,645989	58,175317
Gauss	0,090001	691,945091	0,052068	61,863130
Topologija 2-4-1				
Sigmoida	0,061301	456,730207	0,051252	58,818536
Sinus	0,068642	449,552546	0,053920	59,456622
Gauss	0,122551	686,744970	0,068756	65,353249

Slika 4.11 grafički prikazuje odziv mreže topologije 2-2-1 sa sigmoidalnom aktivacijskom funkcijom učene po skupu. Crna isprekidana linija predstavlja ulaz $u(n)$, crvena linija predstavlja željene vrijednosti izlaza $D(n)$, a plava linija predstavlja dobiveni izlaz iz neuronske mreže $O(n)$.



Slika 4.11: Mreža 2-2-1 (sigmoida) učena po skupu

Vrijednosti težina neuronske mreže prikazane na slici 4.11 nakon učenja su:

$$
\begin{aligned}
\mathbf{V} &= \begin{bmatrix} 0{,}129\,730 & -0{,}849\,074 & -0{,}273\,873 \\ -0{,}926\,338 & -0{,}363\,123 & 0{,}092\,988 \end{bmatrix} \\
\mathbf{W} &= \begin{bmatrix} -1{,}947\,673 & -0{,}645\,668 & -0{,}233\,281 \end{bmatrix}
\end{aligned}
\tag{4.19}
$$

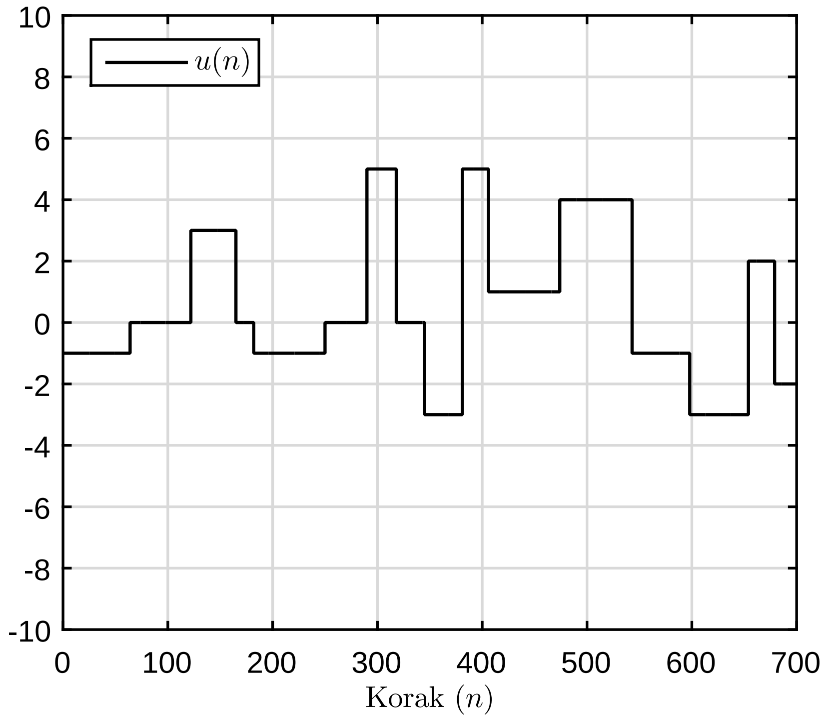
4.2.3 Testiranje

Analogno postupku testiranja opisanom u poglavlju 4.1, a grafički prikazanom slikom 4.5 treba provesti testiranje naučenih mreža. Tablica 4.6 prikazuje vrijednosti NRMS pogreške za prethodno učene mreže u postavu za testiranje. Mreža je pobuđena pobudom iz skupa učenja prikazanim na slici 4.10.

Tablica 4.6: NRMS pogreška testiranja mreža pobudom prema slici 4.10

Mreže:	učene po uzorku	učene po skupu
Topologija 2-2-1		
Sigmoida	0,221101	0,072517
Sinus	0,228334	0,752763
Gauss	0,632171	0,101778
Topologija 2-4-1		
Sigmoida	0,193332	0,068353
Sinus	0,212092	0,200114
Gauss	0,881710	8,418722

Kako bi dalje potvrdili generalizacijsko svojstvo neuronskih mreža naučenih ponašanju nelinearnog dinamičkog sustava potrebno je provesti testiranje mreže sa ulaznom pobudom drugačijom od skupa učenja. Mreža je pobuđena ulaznim signalom prikazanim na slici 4.12, a dobivene vrijednosti pogreške NRMS za naučene mreže su prikazane u tablici 4.7.

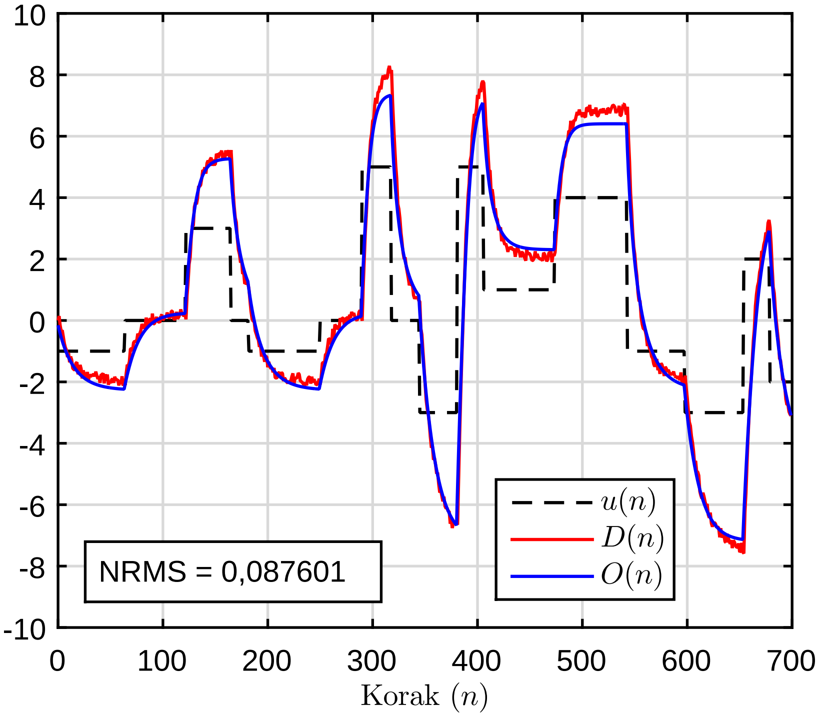


Slika 4.12: Ulazni signal za testiranje

Tablica 4.7: NRMS pogreška testiranja mreža pobudom prema slici 4.12

Mreže:	učene po uzorku	učene po skupu
Topologija 2-2-1		
Sigmoida	0,276925	0,096999
Sinus	0,278776	1,001752
Gauss	0,632410	0,149366
Topologija 2-4-1		
Sigmoida	0,455181	0,087601
Sinus	0,848250	0,178130
Gauss	0,826405	15,517935

Slika 4.13 prikazuje odziv mreže sa sigmoidalnom aktivacijskom funkcijom topologije 2-4-1 učene po skupu na ulazni signal zadan slikom 4.12.



Slika 4.13: Odziv mreže 2-4-1 (sigmoida) učene po skupu

5 Zaključak

Ovim zadatkom pobliže je objašnjeno što su neuronske mreže, njihov način rada i učenja te je uz pomoć programske podrške u MATLAB-u prikazano generalizacijsko svojstvo statičkih unaprijednih neuronskih mreža sa povratnim prostiranjem pogreške prilikom učenja ponašanja nelinearnog i linearnog dinamičkog člana prvog reda. Korištene su neuronske mreže topologija 2-2-1 te 2-4-1 sa aktivacijskim funkcijama navedenima u (2.8), (2.9) i (2.10), a učene su po skupu i uzorku kako je prikazano slikom 4.2. U svrhu ubrzanja samog učenja korišten je algoritam učenja sa momentumom prvog i drugog reda kako je prikazano u (2.29).

Tijekom učenja provedeno je mjerenje vremena učenja te je iz rezultata učenja ponašanja linearnog dinamičkog člana (tablica 4.1) i učenja ponašanja nelinearnog dinamičkog člana (tablica 4.5) vidljivo kako proceduru učenja po skupu koja je opisana u (2.24) karakterizira kraće vrijeme učenja nego kod učenja po uzorku. Također je uočeno dulje vrijeme učenja neuronskih mreža sa gaussovom aktivacijskom funkcijom što je posljedica većeg broja parametara učenja koji su opisani izrazima (2.56) i (2.57).

Prilikom testiranja (prikazano slikom 4.5) naučenih neuronskih mreža korištene su pobudne funkcije iz skupa učenja, ali i pobudne funkcije izvan skupa učenja u svrhu pokazivanja generalizacijskog svojstva neuronskih mreža. Kod testiranja neuronskih mreža naučenih odzivu linearnog dinamičkog člana korištene su pobudne funkcije odskoka, rampe, parabole i sinusa, dok je za testiranje neuronskih mreža naučenih odzivu nelinearnog dinamičkog člana generiran novi ulazni signal nasumičnog karaktera.

Iz dobivenih rezultata možemo zaključiti da neuronske mreže za zadane primjere brže uče primjenom procedure učenja po skupu nego učenjem po uzorku dok je postignuta vrijednost NRMS pogreške u većini slučajeva manja od NRMS pogreške postignute učenjem po uzorku. U svim slučajevima naučene neuronske mreže vjerno prate odzive naučenih sustava što ukazuje na dobra generalizacijska svojstva. Neuronske mreže topologije 2-4-1 pokazuju nešto bolja generalizacijska svojstva od neuronskih mreža topologije 2-2-1, dok neuronske mreže sa sigmoidalnom aktivacijskom funkcijom pokazuju najbolja generalizacijska svojstva.

6 Literatura

[1] W. James, *The Principles of Psychology*, 1890.

[2] W. McCulloch and W. Pitts, *A logical calculus of the ideas immanent in nervous activity*, Bulletin of Mathematical Biophysics 5, pp. 115-133, 1943.

[3] Wikipedia, `https://en.wikipedia.org/wiki/File:Neuron_Hand-tuned.svg`

[4] F. Rosenblatt, *Principles of neurodynamics*, Spartan Books, New York, 1962.

[5] Jondar Gibb , An Ieee , Clifford Lau; *Back propagation Family Album*; 1996.

[6] J. M. Zurada, *Artificial Neural Systems*, W.P. Company, USA, 1992.

[7] Branko Novaković, Dubravko Majetić, Mladen Široki; *Umjetne neuronske mreže*, FSB, Zagreb, 2011.

[8] D. Majetić, *Prilog primjeni neuronskih mreža u sistemima automatskog upravljanja*, Magistarski rad, FSB Zagreb, Croatia, 1992.