

Primjena evolucijskog algoritma za učenje neuronske mreže

Medovka, Krešimir

Master's thesis / Diplomski rad

2022

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:235:466283>

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

Download date / Datum preuzimanja: **2025-01-12**

Repository / Repozitorij:

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



SVEUČILIŠTE U ZAGREBU
FAKULTET STROJARSTVA I BRODOGRADNJE

DIPLOMSKI RAD

Krešimir Medovka, bacc. ing. mech.

Zagreb, 2022.

SVEUČILIŠTE U ZAGREBU
FAKULTET STROJARSTVA I BRODOGRADNJE

DIPLOMSKI RAD

Mentor:

Student:

Izv. prof. dr. sc. Petar Čurković, dipl. ing.

Krešimir Medovka, bacc. ing. mech.

Zagreb, 2022.

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

Zahvaljujem mentoru izv. prof. dr. sc. Petru Ćurkoviću što mi je pružio priliku raditi na ovako zanimljivom diplomskom radu te na njegovoj strpljivosti i podršci.

Zahvaljujem svojoj obitelji na svim savjetima i podršci tijekom studija te svim prijateljima.

Na kraju zahvaljujem svim članovima obitelji i prijateljima koji više nisu tu, bilo fizički ili psihički, a koji su me inspirirali u raznim područjima života.

Krešimir Medovka, bacc. ing. mech.

“Prepreka djelu osnažuje djelo. Što stoji na putu postaje put.“

- Marcus Aurelius



SVEUČILIŠTE U ZAGREBU
FAKULTET STROJARSTVA I BRODOGRADNJE



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

Sveučilište u Zagrebu Fakultet strojarstva i brodogradnje	
Datum:	Prilog:
Klasa:	602-14/22-6/1
Ur. broj:	15-1703-22-

DIPLOMSKI ZADATAK

Student: **KREŠIMIR MEDOVKA** Mat. br.: 0035203080

Naslov rada na hrvatskom jeziku: **Primjena evolucijskog algoritma za učenje neuronske mreže**

Naslov rada na engleskom jeziku: **Application of an evolutionary algorithm for neural network learning**

Opis zadatka:

Povećanjem kompleksnosti neuronskih mreža, učinkovitost standardnih metoda učenja postaje nedostatna. Dodatno, predmet učenja ne moraju nužno biti samo težinske vrijednosti sinapsi, već i sama topologija mreže, kao i prijenosne funkcije.

Evolucijski algoritam omogućuje generiranje populacije neuronskih mreža različitih struktura koje se međusobno natječu u rješavanju zadatka. Procesom kompleksifikacije, krećući od jednostavnijih struktura, razvija se evolucijskim algoritmom populacija mreža odgovarajuće kompleksnosti za rješavanje zadanog problema. Pri tome se procesom selekcije odabire mreža koja je najučinkovitija u trenutnoj generaciji, a postupkom križanja i mutacije tvore se nove strukture mreža.

U radu je potrebno napraviti sljedeće:

- upoznati se s područjem neuroevolucije
- proučiti teorijski koncept NEAT (NeuroEvolution of Augmenting Topologies)
- implementirati NEAT u programskom okružju Matlab
- implementirano rješenje upotrijebiti za rješavanje XOR problema i problema inverznog njihala

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

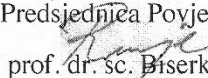
Zadatak zadan:
5. svibnja 2022.

Rok predaje rada:
7. srpnja 2022.

Predviđeni datum obrane:
18. srpnja do 22. srpnja 2022.

Zadatak zadao:

izv. prof. dr. sc. Petar Čurković

Predsjednica Povjerenstva:

prof. dr. sc. Biserka Runje

SADRŽAJ

SADRŽAJ	I
POPIS SLIKA	II
POPIS TABLICA.....	V
POPIS OZNAKA	VI
SAŽETAK.....	VIII
SUMMARY	IX
1. UVOD	1
2. NEURONSKE MREŽE.....	2
3. EVOLUCIJSKI ALGORITMI.....	6
4. O NEUROEVOLUCIJI, PROMJENJIVIM TOPOLOGIJAMA I TEŽINAMA NEURONSKIH MREŽA	8
5. NEUROEVOLUCIJA PROMJENJIVIH TOPOLOGIJA (eng. <i>NeuroEvolution of Augmenting Topologies</i> – NEAT).....	14
5.1 GENETSKO KODIRANJE.....	14
5.2 PRAĆENJE GENA KROZ POVIJESNE OZNAKE	15
5.3 ZAŠTITA INOVACIJE KROZ SPECIACIJU.....	18
5.4 MINIMIZIRANJE DIMENZIONALNOSTI KROZ INKREMENTALNI RAST IZ MINIMALNE STRUKTURE	19
6 IMPLEMENTACIJA NEAT ALGORITMA	20
6.1 XOR PROBLEM.....	20
6.2 OBRNUTO NJIHALO	47
7 ZAKLJUČAK	81
LITERATURA.....	82

POPIS SLIKA

Slika 1. Općenita struktura neuronske mreže [1]	2
Slika 2. Slojevi neuronske mreže: ulazni, sakriveni, izlazni [1]	3
Slika 3. Općenita shema evolucijskih algoritama [5].....	6
Slika 4. Problem konkurentskih konvencija [6]	11
Slika 5. Primjer mapiranja genotipa u fenotip [6].....	14
Slika 6. Dva tipa strukturne mutacije u NEAT-u: mutacija dodavanja veze i mutacija dodavanja čvora [6]	15
Slika 7. Usklađivanje genoma za različite mrežne topologije korištenjem inovacijskih brojeva [6]	17
Slika 8. Početni fenotip i optimalni XOR [6].....	21
Slika 9. Mreža rješenja XOR problema za prvi pokušaj	23
Slika 10. Grafovi fitnesa i razvoja topologije mreže XOR problema za prvi pokušaj.....	24
Slika 11. Mreža rješenja XOR problema za drugi pokušaj	25
Slika 12. Grafovi fitnesa i razvoja topologije mreže XOR problema za drugi pokušaj.....	26
Slika 13. Mreža rješenja XOR problema za treći pokušaj	27
Slika 14. Grafovi fitnesa i razvoja topologije mreže XOR problema za treći pokušaj	28
Slika 15. Mreža rješenja XOR problema za četvrti pokušaj	29
Slika 16. Grafovi fitnesa i razvoja topologije mreže XOR problema za četvrti pokušaj.....	30
Slika 17. Mreža rješenja XOR problema za 5. pokušaj	31
Slika 18. Grafovi fitnesa i razvoja topologije mreže XOR problema za 5. pokušaj	32
Slika 19. Mreža rješenja XOR problema za 6. pokušaj	33
Slika 20. Grafovi fitnesa i razvoja topologije mreže XOR problema za 6. pokušaj	34
Slika 21. Mreža rješenja XOR problema za 7. pokušaj	35
Slika 22. Grafovi fitnesa i razvoja topologije mreže XOR problema za 7. pokušaj	36
Slika 23. Mreža rješenja XOR problema za 8. pokušaj	37
Slika 24. Grafovi fitnesa i razvoja topologije mreže XOR problema za 8. pokušaj	38
Slika 25. Mreža rješenja XOR problema za 9. pokušaj	39
Slika 26. Grafovi fitnesa i razvoja topologije mreže XOR problema za 9. pokušaj	40
Slika 27. Mreža rješenja XOR problema za 10. pokušaj	41
Slika 28. Grafovi fitnesa i razvoja topologije mreže XOR problema za 10. pokušaj	42
Slika 29. Mreža rješenja XOR problema za 11. pokušaj	43
Slika 30. Grafovi fitnesa i razvoja topologije mreže XOR problema za 11. pokušaj	44
Slika 31. Mreža rješenja XOR problema za 12. pokušaj	45

Slika 32. Grafovi fitnesa i razvoja topologije mreže XOR problema za 12. pokušaj	46
Slika 33. Obrnuto njihalo na kolicima	47
Slika 34. Mreža rješenja problema stabilizacije obrnutog njihala za prvi pokušaj	51
Slika 35. Grafovi fitnesa i razvoja topologije mreže problema stabilizacije obrnutog njihala za prvi pokušaj	52
Slika 36. Grafovi varijabli stanja sustava obrnutog njihala za prvi pokušaj	53
Slika 37. Mreža rješenja problema stabilizacije obrnutog njihala za drugi pokušaj	54
Slika 38. Grafovi fitnesa i razvoja topologije mreže problema stabilizacije obrnutog njihala za drugi pokušaj	55
Slika 39. Grafovi varijabli stanja sustava obrnutog njihala za drugi pokušaj	56
Slika 40. Mreža rješenja problema stabilizacije obrnutog njihala za treći pokušaj	57
Slika 41. Grafovi fitnesa i razvoja topologije mreže problema stabilizacije obrnutog njihala za treći pokušaj	58
Slika 42. Grafovi varijabli stanja sustava obrnutog njihala za treći pokušaj	59
Slika 43. Mreža rješenja problema stabilizacije obrnutog njihala za četvrti pokušaj	60
Slika 44. Grafovi fitnesa i razvoja topologije mreže problema stabilizacije obrnutog njihala za četvrti pokušaj	61
Slika 45. Grafovi varijabli stanja sustava obrnutog njihala za četvrti pokušaj	62
Slika 46. Mreža rješenja problema stabilizacije obrnutog njihala za 5. pokušaj.....	63
Slika 47. Grafovi fitnesa i razvoja topologije mreže problema stabilizacije obrnutog njihala za 5. pokušaj	64
Slika 48. Grafovi varijabli stanja sustava obrnutog njihala za 5. pokušaj	65
Slika 49. Mreža rješenja problema stabilizacije obrnutog njihala za 6. pokušaj.....	66
Slika 50. Grafovi fitnesa i razvoja topologije mreže problema stabilizacije obrnutog njihala za 6. pokušaj	67
Slika 51. Grafovi varijabli stanja sustava obrnutog njihala za 6. pokušaj	68
Slika 52. Mreža rješenja problema stabilizacije obrnutog njihala za 7. pokušaj.....	69
Slika 53. Grafovi fitnesa i razvoja topologije mreže problema obrnutog njihala za 7. pokušaj	70
Slika 54. Grafovi varijabli stanja sustava obrnutog njihala za 7. pokušaj	71
Slika 55. Mreža rješenja problema stabilizacije obrnutog njihala za 8. pokušaj.....	72
Slika 56. Grafovi fitnesa i razvoja topologije mreže problema stabilizacije obrnutog njihala za 8. pokušaj	73
Slika 57. Grafovi varijabli stanja sustava obrnutog njihala za 8. pokušaj	74
Slika 58. Mreža rješenja problema stabilizacije obrnutog njihala za 9. pokušaj.....	75
Slika 59. Grafovi fitnesa i razvoja topologije mreže problema stabilizacije obrnutog njihala za 9. pokušaj	76
Slika 60. Grafovi varijabli stanja sustava obrnutog njihala za 9. pokušaj	77

Slika 61. Mreža rješenja problema stabilizacije obrnutog njihala za 10. pokušaj.....	78
Slika 62. Grafovi fitnesa i razvoja topologije mreže problema stabilizacije obrnutog njihala za 10. pokušaj	79
Slika 63. Grafovi varijabli stanja sustava obrnutog njihala za 10. pokušaj	80

POPIS TABLICA

Tablica 1. Tablica istinosti za XOR problem	20
Tablica 2. Rezultati eksperimenta pronalaženja strukture za XOR problem	22
Tablica 3. Rezultati eksperimenta pronalaženja strukture za problem obrnutog njihala	50

POPIS OZNAKA

Oznaka	Mjerna jedinica	Opis oznake
c_1, c_2, c_3	-	Koeficijenti prilagođavanja važnosti
D	-	Broj nepovezanih gena
E	-	Broj suvišnih gena
f_i	-	Fitnes funkcija i -tog organizma
f'_i	-	Modificirana fitnes funkcija i -tog organizma
F	N	Sila
g	m/s^2	Ubrzanje sile teže
i	-	Broj trenutnog organizma u populaciji
j	-	Broj svakog drugog organizma u populaciji
J_2	kgm^2	Inercija njihala
K	J	Ukupna kinetička energija sustava obrnutog njihala na kolicima
K_1	J	Kinetička energija kolica
K_2	J	Kinetička energija njihala
l	m	Duljina kraka
L	J	Lagrangeova funkcija
m_1	kg	Masa kolica
m_2	kg	Masa njihala
n	-	Maksimalni broj organizama u populaciji
N	-	Broj gena u genomu
P	J	Ukupna potencijalna energija sustava obrnutog njihala na kolicima
P_1	J	Potencijalna energija kolica
P_2	J	Potencijalna energija njihala

Oznaka	Mjerna jedinica	Opis oznake
q	-	Vektor stupac varijabli stanja sustava
\bar{W}	-	Prosjek razlike težina gena
x	m	pomak kolica
\dot{x}	m/s	brzina kolica
\ddot{x}	m/s ²	ubrzanje kolica
X_1, X_2	-	Ulazi XOR funkcije
Y	-	Izlaz XOR funkcije
δ	-	Udaljenost kompatibilnosti gena
δ_t	-	Prag udaljenosti kompatibilnosti gena
θ	rad	kut zakreta njihala
$\dot{\theta}$	rad/s	kutna brzina njihala
$\ddot{\theta}$	rad/s ²	Kutno ubrzanje njihala
τ	-	Vektor stupac ulaznih (upravljajčkih) varijabli sustava

SAŽETAK

Tema ovog diplomskog rada bila je Primjena evolucijskog algoritma za učenje neuronske mreže.

Opisane su osnove neuronskih mreža i njihove karakteristike kao što su čvorovi i slojevi, te vrste neuronskih mreža.

Također opisani su evolucijski algoritmi, inspiracija, struktura algoritma i potrebni koraci.

Dan je opis i pregled područja NeuroEvolucije s naglaskom na Neuronske mreže evoluirajućih topologija i težina (eng. *Topology and Weight Evolving Artificial Neural Networks - TWEANN*) te su postavljena neka vrlo bitna pitanja koje će NEAT algoritam pokušati riješiti.

Proučen je teorijski koncept NeuroEvolucije promjenjivih topologija (eng. *NeuroEvolution of Augmenting Topologies - NEAT*).

NEAT algoritam je implementiran u programskom okruženju MATLAB te je implementacija ispitana na dva problema: XOR problem i problem stabilizacije inverznog njihala.

Ključne riječi: NEAT, neuronske mreže, evolucijski algoritmi

SUMMARY

The topic of this thesis was Application of an evolutionary algorithm for neural network learning.

The basics of neural networks and their characteristics such as nodes and layers, and types of neural networks are described.

Evolutionary algorithms, inspiration, algorithm structure and necessary steps are also described.

A description and overview of the field of NeuroEvolution with an emphasis on Topology and Weight Evolving Artificial Neural Networks (TWEANN) was given, and some very important questions were asked that the NEAT algorithm will try to solve.

The theoretical concept of NeuroEvolution of Augmenting Topologies (NEAT) was studied.

The NEAT algorithm was implemented in the MATLAB programming environment, and the implementation was tested on two problems: the XOR problem and the inverse pendulum stabilization problem.

Key words: NEAT, neural networks, evolutionary algorithms

1. UVOD

Povećanjem kompleksnosti neuronskih mreža, učinkovitost standardnih metoda učenja postaje nedostatna. Dodatno, predmet učenja ne moraju nužno biti samo težinske vrijednosti sinapsi, već i sama topologija mreže, kao i projenosne funkcije.

Evolucijski algoritam omogućuje generiranje populacije neuronskih mreža različitih struktura koje se međusobno natječu u rješavanju zadatka. Procesom kompleksifikacije, krećući od jednostavnijih struktura, razvija se evolucijskim algoritmom populacija mreža odgovarajuće kompleksnosti za rješavanje zadanog problema. Pri tome se procesom selekcije odabire mreža koja je najučinkovitija u trenutnoj generaciji, a postupkom križanja i mutacije tvore se nove strukture mreža.

Tema ovog diplomskog rada je Primjena evolucijskog algoritma za učenje neuronske mreže.

Prvo će biti opisane osnove neuronskih mreža i njihove karakteristike kao što su čvorovi i slojevi, te vrste neuronskih mreža.

Drugo biti će opisani evolucijski algoritmi, inspiracija, struktura algoritma i potrebni koraci.

Treće biti će dan opis i pregled područja NeuroEvolucije s naglaskom na Neuronske mreže evoluirajućih topologija i težina (eng. *Topology and Weight Evolving Artificial Neural Networks* - TWEANN).

Četvrto proučiti će se teorijski koncept NeuroEvolucije promjenjivih topologija (eng. *NeuroEvolution of Augmenting Topologies* - NEAT).

Peto, NEAT algoritam će biti implementiran u programskom okruženju MATLAB te će implementacija biti ispittana na dva problema: XOR problem i problem stabilizacije inverznog njihala.

2. NEURONSKE MREŽE

Neuronska mreža je sustav međusobno povezanih procesnih jedinica zvanih neuroni.

Umjetne neuronske mreže (eng. *Artificial Neural Networks*) sastavni su dio umjetne inteligencije i temelj dubokog učenja. ANN je računalna arhitektura koja se sastoji od neurona koji matematički predstavljaju način na koji biološka neuronska mreža djeluje kako bi identificirala i prepoznala odnose unutar podataka.

U osnovi neuronske mreže su nelinearni modeli strojnog učenja, koji se mogu koristiti i za nadzirano i za nenadzirano učenje. Neuronske mreže se također vide kao skup algoritama, koji se približno modeliraju na temelju ljudskog mozga i izgrađeni su za prepoznavanje uzoraka.

Umjetna neuronska mreža je računalni sustav dizajniran da simulira kako ljudski mozak analizira i obrađuje informacije. To je temelj umjetne inteligencije i rješava probleme koji bi se pokazali nemogućim ili teškim prema ljudskim ili statističkim standardima.

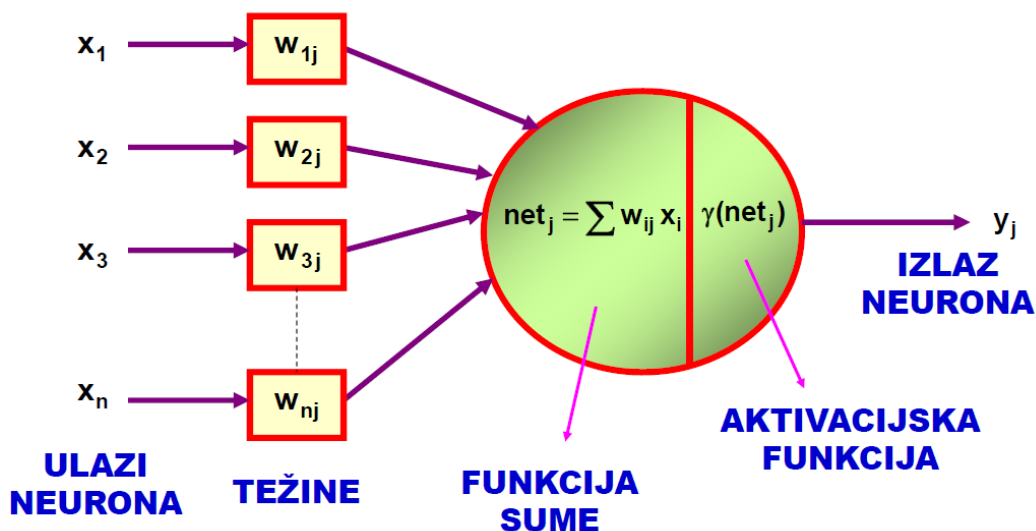
Umjetne neuronske mreže prvenstveno su dizajnirane da oponašaju i simuliraju funkcioniranje ljudskog mozga. Koristeći matematičku strukturu, ANN je konstruirana da replicira biološke neurone.

Ljudski mozak ima proces donošenja odluka: on vidi ili biva izložen informacijama kroz pet osjetilnih organa; te se informacije pohranjuju, povezuju registriranu informaciju s prethodnim saznanjima i u skladu s tim donose određene odluke.

Koncept umjetne neuronske mreže slijedi isti proces kao i prirodne neuronske mreže. Cilj ANN-a je naučiti strojeve ili sustave da shvate kako ljudski mozak donosi odluku i potom poduzima akciju. Inspirirani ljudskim mozgom, osnove neuronskih mreža povezane su neuronima ili čvorovima.

Struktura neuronske mreže ovisi o problemu, a konfigurira se prema primjeni.

Na slici 1. prikazana je općenita struktura neuronske mreže preuzeta iz [1].

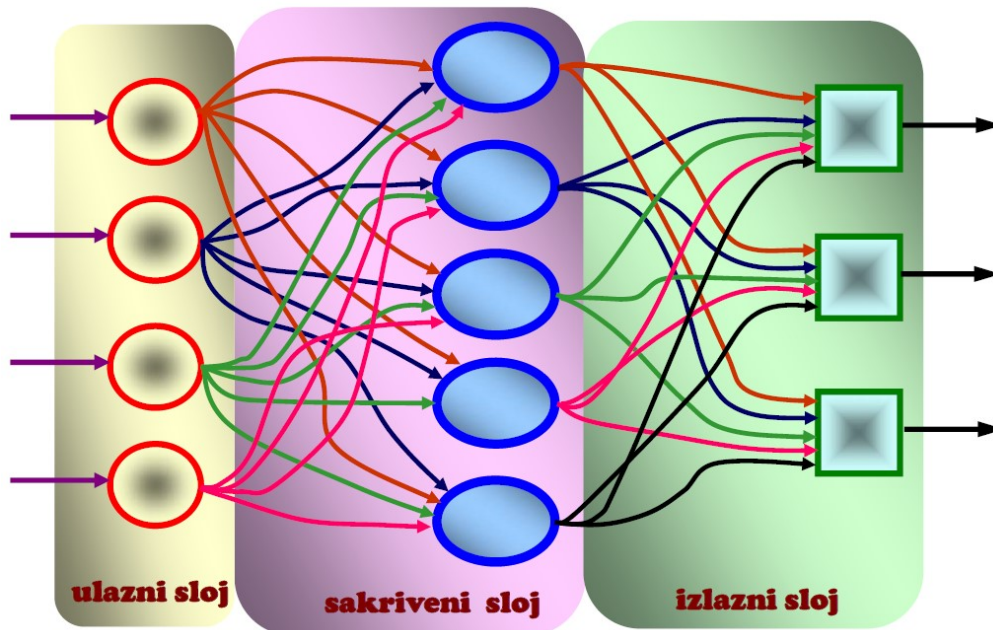


Slika 1. Općenita struktura neuronske mreže [1]

Neuronska mreža ima sljedeću strukturu, a komponente umjetnih neuronskih mreža koje čine osnove neuronskih mreža su:

- Slojevi: ulazni, skriveni i izlazni slojevi
- Neuron
- Funkcija aktivacije
- Težine i *bias*

Na slici 2. prikazani su slojevi neuronske mreže, slika je preuzeta iz [1].



Slika 2. Slojevi neuronske mreže: ulazni, sakriveni, izlazni [1]

U neuronskoj mreži postoje tri sloja: ulazni sloj, skriveni slojevi i izlazni sloj.

Ulazni sloj se sastoji od ulaza ili nezavisne X varijable poznate kao prediktor. Ti se ulazi prikupljaju iz vanjskih izvora kao što su tekstualni podaci, slike, audio ili video datoteke. U prirodnoj mreži, te su varijable informacija koja se percipira iz osjetilnih organa.

Izlazni sloj je rezultat neuronske mreže; to može biti numerička vrijednost u problemu regresije ili, binarna ili višeslojna klasa za problem klasifikacije. Izlaz također može biti prepoznavanje rukopisa, ili glasa, ili klasificirane slike ili teksta u kategorijama.

Osim ulaznog i izlaznog sloja, postoji još jedan sloj u neuronskim mrežama, nazvan skriveni sloj, koji izvodi značajke za model. Neuronska mreža s jednim skrivenim slojem naziva se jednoslojni perceptron, a neuronska mreža s više od jednog skrivenog sloja i gdje je svaki od slojeva povezan naziva se višeslojni perceptron.

Neuroni su primarna i osnovna procesna jedinica u neuronskoj mreži. Prima informacije ili podatke, izvodi jednostavne izračune, a zatim ih prosljeđuje dalje. Neuroni su u skrivenom i izlaznom sloju. Ulazni sloj nema neurone; krugovi u ulaznom sloju predstavljaju nezavisne varijable ili X -ove.

Broj čvorova u izlaznom sloju ovisi o vrsti problema:

- Regresijski: u slučaju da se predviđaju cijene dionica ili drugim riječima kada je priroda izlaza kontinuirana, tada će postojati jedan čvor u izlaznom sloju
- Klasifikacijski: u slučaju klasifikacijskog problema, čvorovi u izlaznom sloju jednaki su broju klasa ili kategorija. Za binarnu klasifikaciju, može postojati jedan ili dva čvora u izlaznom sloju.

Broj neurona u skrivenim slojevima podlozan je korisniku. Arhitektura neuronske mreže konfigurira se na temelju problema i određuje je korisnik. Ulazni sloj je uvijek unaprijed definiran, a izlaz je cilj mreže koji je također unaprijed određen.

Broj skrivenih slojeva i broj neurona čine dio hiperparametara. To je zato jer ovi parametri stvaraju značajke, a male permutacije u njima mogu značajno utjecati na izlaz.

Ulazi iz ulaznog sloja povezani su sa svakim neuronom prvog skrivenog sloja. Slično, neuroni ovog skrivenog sloja povezani su s neuronima sljedećih slojeva. Ukratko, izlaz jednog sloja postaje ulaz drugog sloja. Bilo koji neuron može imati mnogo odnosa s višestrukim ulaznim i izlaznim vezama. Težine i *bias* primjenjuju se na svaku od veza tijekom prijenosa čvorova između slojeva.

U biološkoj neuronskoj mreži sinapse ukazuju na snagu neurona. Na isti način u umjetnim neuronskim mrežama, težine kontroliraju snagu veza između neurona.

Ove težine predstavljaju relativnu važnost neuronske mreže, a značenje pokazuje koliki će prioritet imati ulaz *X* ili koliki će prioritet naknadno izvedeni neuroni imati na izlaz. To će prirodno dovesti do toga da će neuroni s većom težinom imati veći utjecaj u sljedećem sloju, a u konačnici će neuroni koji nemaju značajnu težinu ispasti. Stoga težine ili veze usmjeravaju neuronsku mrežu.

Bias je dodatni ulaz za svaki sloj počevši od ulaznog sloja. *Bias* nije ovisan niti pod utjecajem prethodnog sloja. Jednostavnim riječima, *bias* je pojam presretanja i konstantan je. To implicira da čak i ako nema ulaza ili nezavisnih varijabli, model će biti aktiviran sa zadanom vrijednošću *bias-a*.

Težine i *bias* su parametri modela koji se mogu naučiti. U prvoj iteraciji ili procesu inicijalizacije, težine se nasumično postavljaju ili dodjeljuju i optimiziraju kako bi se smanjili gubitak ili pogreška.

U gornjem odjeljku vidjeli smo da težine kontroliraju prijenos neurona, a dodavanje *bias-a* čini čvorove linearnom kombinacijom težina i *bias-a*.

Trenutačno je jednadžba za neuronsku mrežu modela linearne regresije gdje će svaki neuron izvesti samo linearnu transformaciju za kombinacije težina i *bias-a*. Ovo je previše pojednostavljena neuronska mreža i neće izvesti nikakve složene izračune niti otkriti bilo kakve obrasce. Da bi model bio pripremljen za takvu složenost, potrebna je transformacija, gdje u priču ulaze aktivacijske funkcije.

Aktivacijske funkcije sastavni su dio neuronskih mreža. Svrha aktivacijske funkcije je unijeti nelinearnost u mrežu primjenom transformacije. Aktivacijske funkcije poznate su i kao "funkcija gnječenja".

Postoje različite aktivacijske funkcije ovisno o njihovoj funkcionalnosti i sloju na koji se primjenjuju.

Neki od najčešće korištenih modela umjetnih neuronskih mreža i njihove ključne primjene su:

- Unaprijedna neuronska mreža (eng. *Feed-forward Neural Network*):
- Konvolucijska neuronska mreža (eng. *Convolutional Neural Network*)
- Povratna neuronska mreža (eng. *Recurrent Neural Network*)
- Generativna suparnička mreža (eng. *Generative Adversarial Network*)
- Autoenkoderi

Najjednostavniji tip ANN-a je unaprijedna neuronska mreža. U FNN-u informacije putuju samo u jednom smjeru od ulaznog do izlaznog sloja. Svaki od slojeva je u potpunosti povezan sa svim čvorovima u sloju prije i poslije. U ovoj mreži mogu postojati ili ne moraju postojati skriveni slojevi. Mreža ovdje ne čini petlju. FNN se koristi za klasifikaciju, prepoznavanje lica, prepoznavanje govora, računalni vid.

Konvolucijska neuronska mreža je unaprijedna neuronska mreža, varijacija višeslojnih perceptrona za korištenje minimalne količine predprocesiranja. Uzima ulazne podatke u serijama poput filtera, dodjeljuje relativnu važnost težinama i *bias*-u različitim aspektima ili objektima na slici i razlikuje jedan od drugog. Mreža može zapamtiti slike u dijelovima i izračunati operacije.

CNN ima jedinstvenu arhitekturu koja se fokusira na izdvajanje složenih značajki iz podataka na svakom sloju kako bi se odredio izlaz. Izlaz CNN-a je jedan vektor sa ocjenama vjerojatnosti, a klasa s najvećom vjerojatnošću je predviđanje neuronske mreže.

CNN se obično koristi za izdvajanje informacija iz nestrukturiranih podataka kao što su slike i video podaci. Koristi se za video, signal, prepoznavanje slika, analizu vizualnih prikaza i sustave preporuka.

Povratna neuronska mreža pohranjuje izlaz sloja i vraća ga ulaznom sloju kako bi predvidjela ishod sloja. Ova mreža koristi interno stanje (ili memoriju) za obradu nizova ulaza promjenjive duljine i stoga može rukovati proizvoljnim ulaznim ili izlaznim duljinama. Neuronima ove višeslojne neuronske mreže imaju petlje u svojim vezama.

RNN se primjenjuje za obavljanje složenih zadataka, uključujući pružanje natpisa slika, predviđanje vremenskih serija, učenje rukopisa, prepoznavanje jezika, *chatbot*-ove, otkrivanje prijevare, otkrivanje anomalija i najprikladniji su za obradu nizova unosa.

Generativne suparničke mreže su algoritamske arhitekture koje koriste dvije neuronske mreže: generator i diskriminator, za generiranje sadržaja. Obje ove mreže se treniraju istovremeno, a svaka se mreža natječe jedna protiv druge, s ciljem stvaranja novih, sintetičkih instanci podataka koji se mogu uzeti kao stvarni podaci. GAN-ovi se naširoko koriste u generiranju slika, videa i glasa i rješavaju probleme prijevoda slike u sliku i progresije dobi.

Autoenkoderi uče kodiranje podataka bez nadzora. Cilj autoenkodera je naučiti prikaz ili kodiranje za skup podataka. Oni su uglavnom primjenjivi za smanjenje dimenzionalnosti obučavanjem mreže da ignorira signalni "šum" i učenjem generativnih modela podataka. Primjenjiv je u rekonstrukciji slike i bojanju slike.

3. EVOLUCIJSKI ALGORITMI

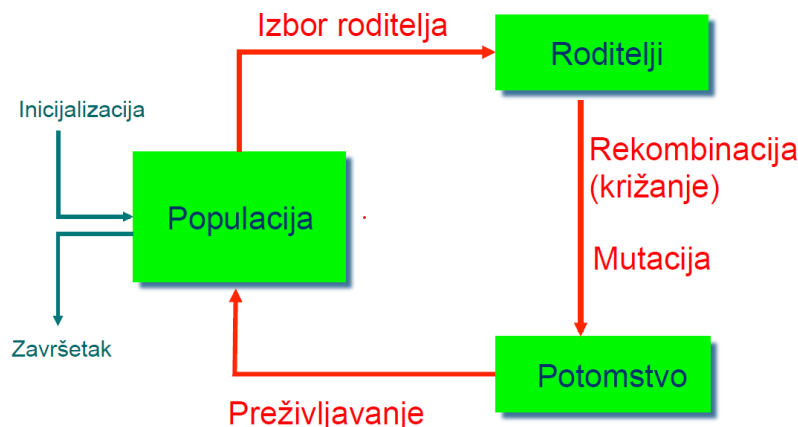
Evolucijski algoritam je heuristika pretraživanja koja je inspirirana teorijom prirodne evolucije Charlesa Darwina. Ovaj algoritam odražava proces prirodne selekcije gdje se za reprodukciju odabiru najsposobniji pojedinci kako bi se proizvelo potomstvo sljedeće generacije.

Evolucijski algoritmi uspješno se koriste između ostalog za optimiranje topologije rešetkastih struktura [2], te kod planiranja trajektorije robota [3], a u kombinaciji s neuronskim mrežama, omogućuje se efikasno planiranje trajektorija mobilnih robota [4].

Proces prirodne selekcije počinje odabirom najsposobnijih jedinki iz populacije. Oni proizvode potomstvo koje nasljeđuje karakteristike roditelja i koji će biti dodani sljedećoj generaciji. Ako roditelji imaju bolju fitnes funkciju, njihovo će potomstvo biti bolje od roditelja i imat će veće šanse za preživljavanje. Ovaj proces se nastavlja kroz određen broj iteracija i na kraju će se pronaći generacija s pojedincima koji imaju najbolju fitnes funkciju.

Ovaj se pojam može primijeniti na problem pretraživanja. Razmatramo skup rješenja za problem i od njih odabiremo skup najboljih.

Na slici 3. je prikazana općenita shema evolucijskih algoritama preuzeta iz [5]



Slika 3. Općenita shema evolucijskih algoritama [5]

U genetskom algoritmu razmatra se pet faza:

1. Početna populacija
2. Fitnes Funkcija
3. Odabir
4. Križanje
5. Mutacija

Proces počinje skupom pojedinaca koji se naziva populacija. Svaki pojedinac je rješenje za problem koji se želi riješiti. Pojedinaca karakterizira skup parametara (varijabli) poznatih kao geni. Geni se spajaju u niz kako bi se formirao kromosom (rješenje).

U genetskom algoritmu, skup gena pojedinca je predstavljen pomoću niza, u terminima abecede. Obično se koriste binarne vrijednosti (niz jedinica i nula). Kažemo da kodiramo gene u kromosomu.

Fitnes funkcija određuje koliko je pojedinac u formi (sposobnost pojedinca da se natječe s drugim pojedincima). Daje ocjenu fitnesa svakom pojedincu. Vjerojatnost da će pojedinac biti odabran za reprodukciju temelji se na ocjeni njegovog fitnesa.

Ideja faze selekcije je odabrati najsposobnije pojedince i pustiti ih da svoje gene prenesu na sljedeću generaciju.

Dva para pojedinaca (roditelja) biraju se na temelju njihovih rezultata fitnesa. Pojedinci s visokim fitnesom imaju više šanse da budu odabrani za reprodukciju.

Križanje je najznačajnija faza u genetskom algoritmu. Za svaki par roditelja koji će se pariti, točka križanja se bira nasumično unutar gena. Potomstvo se stvara razmjenom gena roditelja među sobom dok se ne postigne točka križanja.

U određenim novo formiranim potomcima, neki od njihovih gena mogu biti podvrgnuti mutaciji s malom slučajnom vjerojatnošću. To implicira da se neki od bitova u nizu bitova mogu preokrenuti.

Mutacija se događa kako bi se održala raznolikost unutar populacije i spriječila prerana konvergencija.

Algoritam završava ako je populacija konvergirala (ne proizvodi potomke koji se značajno razlikuju od prethodne generacije). Tada se može reći da je genetski algoritam dao skup rješenja za zadani problem.

4. O NEUROEVOLUCIJI, PROMJENJIVIM TOPOLOGIJAMA I TEŽINAMA NEURONSKIH MREŽA

Neuroevolucija (eng. *NeuroEvolution*), umjetna evolucija neuronskih mreža korištenjem genetskih algoritama, pokazuje veliko obećanje u složenim zadacima podržanog učenja (eng. *reinforcement learning*) (Gomez and Miikkulainen, 1999; Gruau et al., 1996; Moriarty and Miikkulainen, 1997; Potter et al., 1995; Whitley et al., 1993). Neuroevolucija kroz prostor ponašanja traži mrežu koja dobro izvršava zadani zadatak. Ovaj pristup rješavanju problema složenog upravljanja predstavlja alternativu statističkim metodama koje pokušavaju procijeniti korisnost pojedinih radnji u pojedinim stanjima (Kaelbling et al., 1996). NE je obećavajući pristup rješavanju problema podržanog učenja iz nekoliko razloga. Neke studije su pokazale da je NE brža i učinkovitija od metoda podržanog učenja kao što su *Adaptive Heuristic Critic* i *Q-Learning* kod problema inverznog njihala i upravljanja robotskom rukom (Moriarty and Miikkulainen, 1996; Moriarty, 1997). Kako NE traži ponašanje umjesto funkcije vrijednosti, učinkovita je u problemima s kontinuiranim i visokodimenzionalni prostorima stanja. Osim toga, pamćenje je lako predstavljeno kroz povratne veze u neuronskim mrežama, čineći NE prirodnim izborom za učenje ne-Markovljevskih zadataka (Gomez and Miikkulainen, 1999, 2002).

U tradicionalnim NE pristupima, topologija je odabrana za evoluirajuće mreže prije nego što eksperiment počinje. Obično je topologija mreže jedan skriveni sloj neurona, pri čemu je svaki skriveni neuron povezan sa svakim mrežnim ulazom i svakim mrežnim izlazom. Evolucija pretražuje prostor veza težina ove potpuno povezane topologije dopuštajući mrežama visokih performansi reprodukciju. Prostor težina se istražuje kroz križanje vektora težine mreže i kroz mutaciju težina pojedinačnih mreža. Dakle, cilj NE fiksne topologije je optimizacija veza težina koje određuju funkcionalnost mreže.

Međutim, težine veza nisu jedini aspekt neuronskih mreža koji doprinose njihovom ponašanju. Topologija, odnosno struktura, neuronskih mreža također utječe na njihovu funkcionalnost. Modificiranje strukture mreže pokazalo se učinkovitim kao dio nadziranog učenja (Chen et al., 1993). Također vlada veliki interes u razvoju mrežnih topologija, kao i težina (Angeline et al., 1993; Branke, 1995; Gruau et al., 1996; Yao, 1999). Osnovno pitanje, međutim, ostaje: Može li razvoj topologije zajedno s težinama pružiti prednost nad evoluirajućim težinama fiksne topologije? Potpuno povezana mreža u načelu može aproksimirati bilo koju kontinuiranu funkciju (Cybenko, 1989). Pa zašto gubiti vrijedan trud permutirajući kroz različite topologije?

Dosadašnji odgovori su neuvjerljivi. Neki su tvrdili da složenost mreže može utjecati na brzinu i točnost učenja (Zhang and Muhlenbein, 1993). Iako je ova tvrdnja istinita za algoritam širenja unatrag, nije jasno primjenjuje li se ta tvrdnja kada se težine optimiziraju evolucijom, a ne propagacijom unatrag.

Uvjerljiv argument za evoluciju topologije i težina iznesen je od strane Gruau i sur. (1996.), koji je tvrdio da evoluirajuća struktura štedi vrijeme potrošeno od strane ljudi koji pokušavaju odlučiti o topologiji mreža za određeni NE problem. Iako gotovo svi NE sustavi fiksne topologije koriste potpuno povezan skriveni sloj, odlučivanje koliko je skrivenih čvorova potrebno je proces pokušaja i pogreške. Gruau i sur. potkrijepili su svoj argument razvijanjem topologije i težina umjetne neuronske mreže koja je riješila najteži referentni problem inverznog njihala do tada. Međutim, kasniji rezultati sugeriraju da struktura nije potrebna za rješavanje problema. Metoda fiksne topologije nazvana *Enforced Subpopulations* (ESP) (Gomez and Miikkulainen, 1999) uspjela je riješiti isti problem 5 puta brže jednostavnim ponovnim pokretanjem s nasumičnim brojem skrivenih neurona kad god bi se zaglavila.

Cilj [6] bio je demonstrirati suprotan zaključak: ako se napravi kako treba, evoluiranje strukture zajedno s vezama težina može značajno poboljšati performanse NE. Predstavili su novu NE metodu pod nazivom *NeuroEvolution of Augmenting Topologies* (NEAT) koja je dizajnirana da iskoristi prednost strukture kao načina minimiziranja dimenzionalnosti prostora za pretraživanje veza težina. Ako je struktura evoluirana tako da se topologije minimiziraju i postupno povećavaju, rezultat je značajan dobitak u brzini učenja. Poboljšana učinkovitost je rezultat minimiziranja topologija kroz evoluciju, a ne samo na samom kraju.

Evoluiranje strukture postepenim povećavanjem predstavlja nekoliko tehničkih izazova:

- (1) Postoji li genetski prikaz koji omogućuje da se različite topologije križaju na smislen način?
- (2) Kako topološka inovacija, kojoj je potrebno nekoliko generacija da bi bila optimizirana, može biti zaštićena da prerano ne nestane iz populacije?
- (3) Kako se topologije mogu minimizirati kroz evoluciju bez potrebe za posebno izmišljenom fitness funkcijom koja mjeri složenost?

NEAT metoda se sastoji od rješenja za svaki od ovih problema kako će biti opisano u poglavlju 5. Metodu su potvrdili na problemu inverznog njihala, gdje NEAT izvodi 25 puta brže od *Cellular Encoding* i 5 puta brže od ESP-a. Rezultati pokazuju da je struktura moćan resurs u NE kada se pravilno koristi. NEAT je jedinstvena metoda jer strukture postaju sve složenije kako postaju sve više optimalne, jačajući analogiju između GA i prirodne evolucije.

Tijekom vremena razvijeni su mnogi sustavi koji razvijaju i topologiju neuronske mreže i težine (Angeline et al., 1993; Braun and Weisbrod, 1993; Dasgupta and McGregor, 1992; Fullmer and Miikkulainen, 1992; Gruau et al., 1996; Krishnan and Ciesielski, 1994; Lee and Kim, 1996; Mandischer, 1993; Maniezzo, 1994; Opitz and Shavlik, 1997; Pujol and Poli, 1998; Yao and Liu, 1996; Zhang and Muhlenbein, 1993). Ove metode obuhvaćaju niz ideja o tome kako bi se Neuronske mreže evoluirajućih topologija i težina (eng. *Topology and Weight Evolving Artificial Neural Networks*) trebale biti implementirane. U ovom odjeljku adresirane su neke od ideja i pretpostavki o dizajnu TWEANN-a i ponuđena rješenja nekih neriješenih problema. Cilj je otkriti kako metoda neuroevolucije može koristiti evoluciju topologije kako bi povećala svoju učinkovitost.

Pitanje kako kodirati mreže koristeći učinkovitu genetsku reprezentaciju mora biti adresiran od strane svih TWEANN-a.

TWEANN-ovi se mogu podijeliti između onih koji koriste izravno kodiranje i onih koji koriste neizravno. Sheme izravnog kodiranja, koje koristi većina TWEANN-a, specificira u genomu svaku vezu i čvor koji će se pojaviti u fenotipu (Angeline et al., 1993; Braun and Weisbrod, 1993; Dasgupta and McGregor, 1992; Fullmer and Miikkulainen, 1992; Krishnan and Ciesielski, 1994; Lee and Kim, 1996; Maniezzo, 1994; Opitz and Shavlik, 1997; Pujol and Poli, 1998; Yao and Liu, 1996; Zhang and Muhlenbein, 1993). Nasuprot tome, metode neizravnog kodiranja obično samo specificiraju pravila za konstruiranje fenotipa (Gruau, 1993; Mandischer, 1993). Ova pravila mogu biti specifikacije slojeva ili pravila rasta kroz diobu ćelija. Neizravno kodiranje omogućuje kompaktniji prikaz nego izravno kodiranje, jer svaka veza i čvor nisu specificirani u genomu, iako se iz njega mogu izvesti.

Izravna kodiranja obično zahtijevaju jednostavnije implementacije od neizravnih kodiranja. Najjednostavnija implementacija temelji se na tradicionalno korištenom prikazu niza bitova od strane GA-a. Na primjer, Dasgupta i McGregor (1992) koriste takvo kodiranje u svojoj metodi, nazvanoj *Structured Genetic Algoritam* (sGA), gdje niz bitova predstavlja matricu veza mreže. sGA je prepoznatljiv po svojoj jednostavnosti, što mu omogućuje rad gotovo kao standardni GA. Međutim, postoji i nekoliko ograničenja. Prvo, veličina matrice veza je kvadrat broja

čvorova. Dakle, reprezentacija se raspada za veliki broj čvorova. Drugo, zato što veličina niza bitova mora biti ista za sve organizme, maksimalni broj čvorova (a time i veza također) mora odabrati čovjek koji upravlja sustavom, a ako maksimum nije dovoljan, pokus se mora ponoviti. Treće, korištenje linearnog niza bitova za predstavljanje grafičke strukture otežava osiguravanje korisnih kombinacija koje rezultiraju križanjem.

Kako nizovi bitova nisu najprirodniji prikaz za mreže, većina TWEANN-ova koriste kodiranja koja eksplicitnije predstavljaju grafičke strukture. Pujol i Poli (1997) koriste shemu dvojnog predstavljanja kako bi se omogućile različite vrste križanja u njihovom sustavu *Parallel Distributed Genetic Programming* (PDGP). Prva reprezentacija je grafička struktura. Druga je linearni genom definicija čvorova koji specificiraju dolazne i odlazne veze. Ideja je da su različiti prikazi prikladni za različite vrste operatera. Križanja s zamjenom podgraфа i topološke mutacije koriste mrežu, dok križanja točaka i mutacije parametara veza koriste linearni prikaz.

Kao i u sGA, PDGP ima ograničenje broja čvorova u mreži, što odgovara broju čvorova u dvodimenzionalnoj mreži koja predstavlja grafičku verziju genoma. PDGP koristi grafičko kodiranje tako da podgrafovi mogu biti zamijenjeni u križanju. Zamjena podgraфа je reprezentativna za prevladavajuću filozofiju u TWEANN-ovima da su podgrafovi funkcionalne jedinice i stoga njihova zamjena ima smisla jer čuva strukturu funkcionalnih komponenti. Međutim, ne može se sa sigurnošću reći jesu li određeni podgrafovi koji se kombiniraju u PDGP-u ispravni za stvaranje funkcionalnog potomstva.

Budući da križanje mreža s različitim topologijama često može dovesti do gubitka funkcionalnosti, neki istraživači su u potpunosti odustali od križanja u tzv. Evolucijskom programiranju (Yao i Liu, 1996.). Angeline i sur. (1993) implementirali su sustav pod nazivom *Generalized Acquisition of Recurrent Links* (GNARL), komentirajući da "Izgledi za razvojem konekcionističkih mreža s križanjem općenito izgledaju ograničeni." Iako GNARL koristi grafičko kodiranje, ono se bitno razlikuje od PDGP u tome što u potpunosti zaobilazi pitanje križanja. GNARL pokazuje da TWEANN ne treba križanje da bi funkcionirala, ostavljajući problem demonstriranja prednosti križanja drugim metodama.

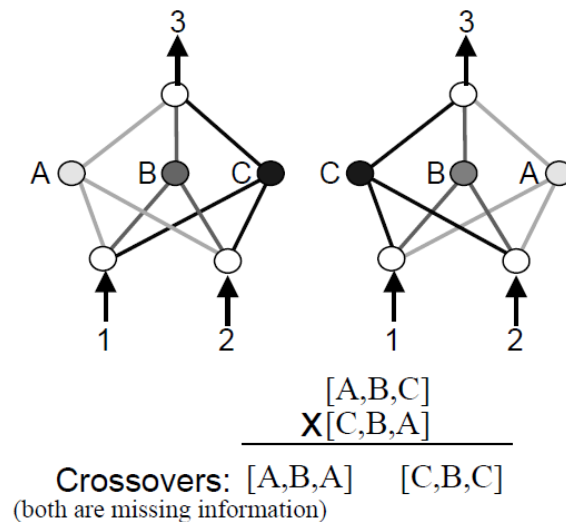
Gruauova (1993.) metoda *Cellular Encoding* (CE) primjer je sustava koji koristi neizravno kodiranje mrežnih struktura. U CE, genomi su programi napisani u specijaliziranom jeziku za transformaciju grafova. Transformacije su motivirane prirodom u tome što određuju diobe stanica. Različite vrste povezivosti mogu rezultirati iz razdiobe, pa je moguće nekoliko vrsta staničnih dioba. Velika prednost CE je da su genetski prikazi kompaktni. Geni u CE mogu se upotrijebiti više puta tijekom razvoja mreže, svaki put zahtijevajući podjelu stanice na drugo mjesto. CE pokazuje da stanične diobe mogu kodirati razvoj mreže iz jedne stanice, kao što organizmi u prirodi počinju kao jedna stanica koja se razlikuje dok se dijeli na više stanica.

Iako CE pokazuje da je moguće evoluirati razvojne sustave, u [6] koriste izravno kodiranje za NEAT jer riječima Brauna i Weisbroda (1993) neizravno kodiranje „zahtijeva detaljnije poznavanje genetskih i neuronskih mehanizama“. Drugim riječima, zato što se neizravnim kodiranjem ne mapiraju izravno na njihove fenotipove, pretraživanje može postati pristrano na nepredvidive načine. Da bi se neizravno kodiranje dobro iskoristilo, potrebno ga je najprije dovoljno dobro razumjeti kako bi se osiguralo da se ne usredotočuje na traženje nekih suboptimalnih klasa topologija.

Jedan od glavnih problema za NE je problem konkurentskih konvencija (Montana i Davis, 1989.; Schaffer et al., 1992.), također poznat kao problem permutacija (Radcliffe, 1993.). Konkurentne konvencije znači imati više od jednog načina izražavanja rješenja problema

optimizacije težina s neuronskom mrežom. Kada genomi koji predstavljaju isto rješenje nemaju isto kodiranje, križanjem će nastati oštećeno potomstvo.

Na slici 4., preuzetoj iz [6], je prikazan problem konkurentskih konvencija. Dvije mreže računaju istu egzaktну funkciju iako se njihove skrivene jedinice pojavljuju drugačijim redoslijedom i predstavljene su različitim kromosomima, što ih čini nekompatibilnima za križanje. Slika pokazuje da objema rekombinacijama u jednoj točki nedostaje jedna od 3 glavne komponente svakog rješenja. Prikazane mreže su samo dvije od šest mogućih permutacija poretka skrivenih jedinica.



Slika 4. Problem konkurentskih konvencija [6]

Slika 4., preuzeta iz [6], prikazuje problem za jednostavnu mrežu s 3 skrivene jedinice. Tri skrivena neurona A, B i C, mogu predstavljati isto opće rješenje u $3! = 6$ različitih permutacija. Kada se jedna od ovih permutacija križa s drugom, kritične informacije vjerojatno će se izgubiti. Na primjer, križanje $[A, B, C]$ i $[C, B, A]$ može rezultirati u $[C, B, C]$, prikaz koji je izgubio jednu trećinu informacija koje su oba roditelji imali. Općenito, za n skrivenih jedinica postoji $n!$ funkcionalno ekvivalentnih rješenja. Problem se može dodatno zakomplicirati različitim konvencijama, tj. $[A, B, C]$ i $[D, B, E]$, koji dijele funkcionalnu međuovisnost o B.

Još teži oblik konkurentskih konvencija prisutan je u TWEANN-ovima, jer TWEANN mogu predstavljati slična rješenja koristeći potpuno različite topologije, ili čak genome različitih veličina. Budući da TWEANN ne zadovoljavaju stroga ograničenja na vrste topologija koje proizvode, predložena rješenja za problem konkurentnih konvencija za mreže fiksne ili ograničene topologije, kao što je neredundantno genetsko kodiranje (Thierens, 1996), se ne primjenjuju. Radcliffe (1993) ide tako daleko nazivajući integriranu shemu koja kombinira povezanost i težine "Sveti gral u ovom području". Iako su neki TWEANN-ovi poput PDGP-a pokušali riješiti problem uz pretpostavku da podmreže predstavljaju funkcionalne jedinice koje se mogu rekombinirati, različite topologije se možda uopće ne temelje na istim podmrežama, pa u tom slučaju ne postoji smisljena kombinacija podstruktura.

Glavna intuicija iza NEAT-a proizlazi iz temeljnog problema sa predstavljanjem različitih struktura: njihovi se prikazi neće nužno podudarati. Ponekad genomi mogu biti različitih veličina. U drugim slučajevima, geni u potpuno istom položaju na različitim kromosomima mogu izražavati potpuno različite osobine. Također, geni koji izražavaju istu osobinu mogu se pojaviti na različitim pozicijama na različitim kromosomima. Kako se te komplikacije mogu riješiti?

Priroda se suočava sa sličnim problemom s usklađivanjem gena u seksualnoj reprodukciji. Genomi u prirodi također nisu fiksne duljine. Negdje u evoluciji iz pojedinačnih stanica u složenije organizme, novi geni su dodani genomima u procesu koji se naziva amplifikacija gena (Darnell and Doolittle, 1986; Watson et al., 1987). Ako bi se novi geni mogli samo nasumično umetnuti u položaje na genomu bez bilo kakve naznake koji je gen koji, život vjerojatno ne bi uspio, jer bi problem konkurentskih konvencija desetkovao ogroman dio potomaka. Trebao je postojati neki način da se križanje održi uredno, kako bi se pravi geni mogli križati s pravim genima.

Priroda kao rješenje koristi homologiju: dva gena su homologna ako su aleli iste osobine. Stvarna homologija između neuronske mreže ne može se lako utvrditi izravnom strukturnom analizom (dakle, problem konkurentskih konvencija). Glavni uvid u NEAT-u je da je povijesno podrijetlo dva gena izravan dokaz homologije ako geni dijele isto podrijetlo. Stoga, NEAT izvršava umjetnu sinapsu na temelju povijesnih oznaka, dopuštajući joj dodavanje novih struktura bez gubljenja traga koji je gen koji tijekom simulacije.

U TWEANN-ovima, inovacija se odvija dodavanjem novih struktura mrežama kroz mutaciju. Često dodavanjem nove strukture u početku uzrokuje smanjenje vrijednosti fitnes funkcije mreže. Na primjer, dodavanje novog čvora uvodi nelinearnost gdje je prije nije bilo; dodavanje nove veze može smanjiti vrijednost fitnes funkcije prije nego što njezina težina ima priliku za optimizaciju. Malo je vjerojatno da će novi čvor ili veza imati korisnu funkciju čim se uvede. Za optimizaciju je potrebno nekoliko generacija novu strukturu kako bi se mogla iskoristiti. Nažalost, zbog početnog gubitka vrijednosti fitnes funkcije uzrokovane novom strukturom, malo je vjerojatno da će inovacija preživjeti u populaciji dovoljno dugo da bi bila optimizirana. Stoga je potrebno nekako zaštititi mreže sa strukturnim inovacijama kako bi imale priliku iskoristiti svoju novu strukturu.

Sustav GNARL rješava problem zaštite inovacija dodavanjem nefunkcionalne strukture. Čvor se dodaje genomu bez ikakvih veza, u nadi se da će se u budućnosti razviti neke korisne veze. Međutim, nefunkcionalne strukture možda se nikada neće povezati sa funkcionalnom mrežom, dodajući vanjske parametre za pretragu.

U prirodi se različite strukture obično nalaze u različitim vrstama koje se natječu u različitim nišama. Dakle, inovacija je implicitno zaštićena unutar niše. Slično, ako bi mreže s inovativnim strukturama mogle biti izolirane unutar vlastite vrste, imale bi priliku optimizirati svoje strukture prije nego što bi se morale natjecati s populacijom u cjelini.

Specijacija, također poznata kao *niching*, proučavana je u GA, ali se obično ne primjenjuje na neuroevoluciju. Specijacija se najčešće primjenjuje na multimodalnu optimizaciju funkcija (Mahfoud, 1995), gdje funkcija ima više optimuma, a GA s nekoliko vrsta se koristi za pronalaženje tih optimuma. Specijacija je također primijenjena u kooperativnoj koevoluciji modularnih sustava sa više rješenja (Darwen and Yao, 1996; Potter and De Jong, 1995)..

Specijacija zahtijeva funkciju kompatibilnosti kako bi se utvrdilo trebaju li dva genoma biti u istoj vrsti ili ne. Teško je formulirati takvu funkciju kompatibilnosti između mreža različitih topologija, što može biti razlog specijacija nije uvedena u TWEANN-ove. Problem konkurentskih konvencija čini mjerenje kompatibilnosti posebno problematično jer mreže koje računaju istu funkciju mogu biti vrlo različite.

Međutim, budući da NEAT ima rješenje za problem konkurentskih konvencija, koristeći povijesne informacije o genima, specijacija populacije u NEAT-u lako je provediva. Koristi se eksplicitno dijeljenje fitnessa, što prisiljava pojedince sa sličnim genomima da podijele svoju kondicijsku isplatu (Goldberg and Richardson, 1987). Izvorna Hollandova (1975) verzija implicitnog dijeljenja fitnessa grupirala je pojedince prema sličnosti izvedbe, a ne genetskoj

sličnosti. Eksplicitna verzija je prikladna za TWEANN-ove jer omogućuje grupiranje mreža na temelju topologije i konfiguracije težina. Rezultat dijeljenja fitnessa je veliki broj mreža koje mogu postojati u populaciji na jednom vrhuncu fitnessa, a broj je ograničen je veličinom vrha. Stoga, populacija se dijeli na brojne vrste, svaka na različitom vrhu, bez prijetnje preuzimanja bilo koje vrste. Eksplicitno dijeljenje fitnessa jako je prikladno za NEAT, jer se sličnost može lako izmjeriti na temelju povijesnih informacija u genima. Tako su inovacije u NEAT-u zaštićene u svojoj vrsti.

U mnogim TWEANN sustavima, početna populacija je zbirka nasumičnih topologija. Takva populacija osigurava topološku raznolikost od samog početka. Međutim, ispostavilo se da nasumične početne populacije stvaraju mnoge probleme za TWEANN. Na primjer, u mnogim shemama izravnog kodiranja, postoji šansa da mreža neće imati put od svakog ulaza do svakog izlaza. Za takve nepraktične mreže potrebno je vrijeme da bi se iskorijenile iz stanovništva.

Međutim, postoji suptilniji i ozbiljniji problem s nasumičnim pokretanjem. Poželjno je razvijati minimalna rješenja; na taj se način smanjuje broj parametara koje se mora pretraživati. Početak sa slučajnim topologijama ne dovodi do pronalaženja minimalnih rješenja, budući da populacija počinje s mnogo već prisutnih nepotrebnih čvorova i veza. Nijedan od tih čvorova ili veza nisu morali izdržati niti jednu evaluaciju, što znači da nema opravdanja za njihovu konfiguraciju. Bilo kakvo minimiziranje mreža bi moralo biti utrošeno na rješavanje varijabli koje uopće nisu trebale biti tamo, i ništa u procesu rekombinacije različitih topologija ne kreće prema takvoj minimizaciji. Budući da u stvaranju većih mreža nema troškova fitnessa, one će dominirati sve dok imaju visoku kondiciju.

Jedan od načina da se nametnu minimalne topologije je uključivanje veličine mreže u fitness funkciju, a neki TWEANN-ovi to zapravo rade (Zhang and Muhlenbein, 1993). U takvim metodama, većim mrežama fitness je kažnjen. Iako promjena fitness funkcije na ovaj način može potaknuti manje mreže, teško je znati kolika bi kazna trebala biti za bilo koju veličinu mreže, posebno zato što različiti problemi mogu imati značajno različite topološke zahtjeve. Mijenjanje funkcija fitnessa je ad hoc i može uzrokovati da evolucija djeluje drugačije od one koju je dizajner izvorne nemodificirane fitness funkcije namijenio.

Alternativno rješenje je da sama metoda neuroevolucije teži prema minimalnosti. Ako populacija počinje bez skrivenih čvorova i struktura raste samo kada to koristi rješenju, nema potrebe za ad hoc modifikacijom kondicije kako bi se mreže minimizirale. Stoga, počevši s minimalnom populacijom i rastućom strukturom je princip dizajna NEAT-a.

Počevši minimalno, NEAT osigurava da sustav traži rješenje u prostoru najniže moguće dimenzije težina tijekom svih generacija. Dakle, cilj nije minimizirati samo konačni proizvod, već također sve usputne međumreže. Ova ideja je ključna za stjecanje prednosti od evolucije topologije: omogućuje nam minimiziranje prostora pretraživanja, što rezultira dramatičnim dobitcima u performansama. Jedan od razloga zašto sadašnji TWEANN-i ne započinju minimalno je taj da bez prisutne topološke raznolikosti u početnoj populaciji, topološke inovacije ne bi preživile. Ovim metodama se ne rješava problem zaštite inovacija, tako da se mreže s velikim strukturnim dodacima vjerojatno neće reproducirati. Stoga, specijacija populacije omogućuje minimalno pokretanje kod NEAT-a.

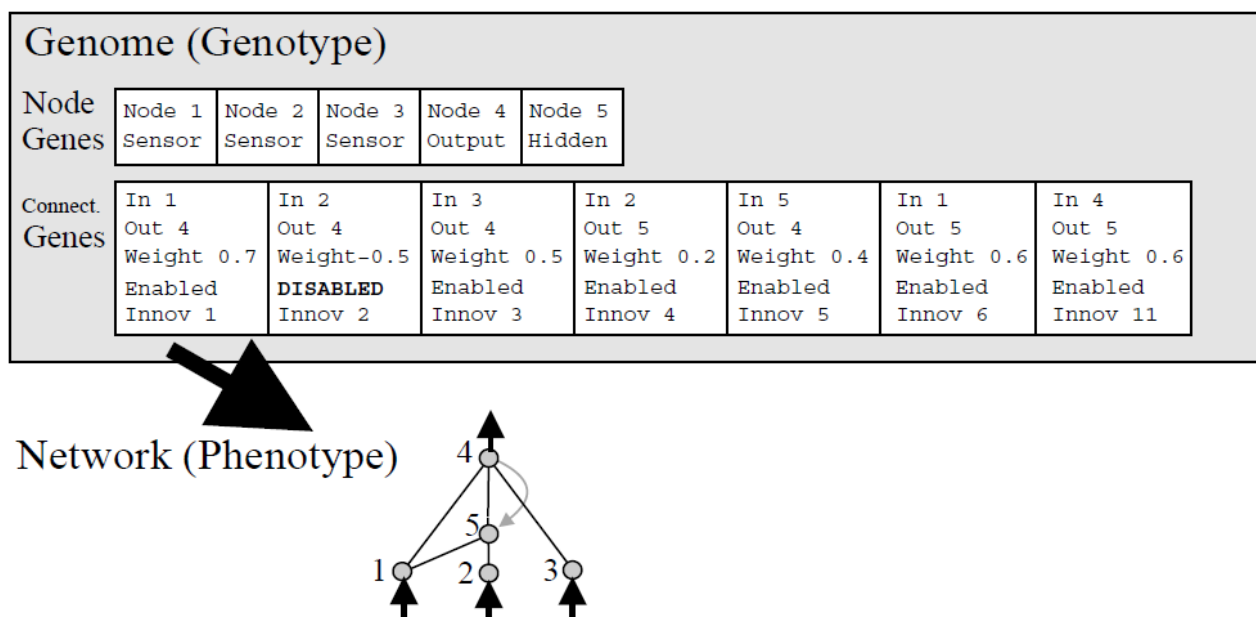
5. NEUROEVOLUCIJA PROMJENJIVIH TOPOLOGIJA (eng. *NeuroEvolution of Augmenting Topologies – NEAT*)

NEAT metoda, kako je detaljno opisano u ovom odjeljku, sastoji se od sastavljanja ideja razvijenih u prethodnom odjeljku u jedan sustav. Na početku je objašnjenje genetsko kodiranje korišteno u NEAT-u, a zatim su opisane komponente koje rješavaju svaki od tri problema TWEANN-a.

5.1 GENETSKO KODIRANJE

Shema genetskog kodiranja NEAT-a osmišljena je kako bi se omogućilo odgovarajućim genima jednostavno kako bi se jednostavno postrojili kada se dva genoma ukrste tijekom parenja. Genomi su linearni prikazi mrežne povezivosti (slika 5.). Svaki genom uključuje popis gena veza, od kojih se svaki odnosi na dva gena čvora koji su povezani. Geni čvora pružaju popis ulaza, skrivenih čvorova i izlaza koji se mogu spojiti. Svaki gen veza specificira ulazni čvor, izlazni čvor, težinu veze, bez obzira na to je li gen veze izražen ili ne je (omogućeni bit), i inovacijski broj, koji dopušta pronalaženje odgovarajućih gena (kao što će biti objašnjeno u nastavku).

Na slici 5., preuzetoj iz [6], prikazan je primjer mapiranja genotipa u fenotip. Prikazani genotip proizvodi prikazani fenotip. Postoje tri ulazna čvora, jedan skriveni i jedan izlazni čvor, i sedam definicija veze, od kojih je jedna povratna. Drugi gen je onemogućen, tako da veza koju specificira (između čvorova 2 i 4) nije izražena u fenotip.

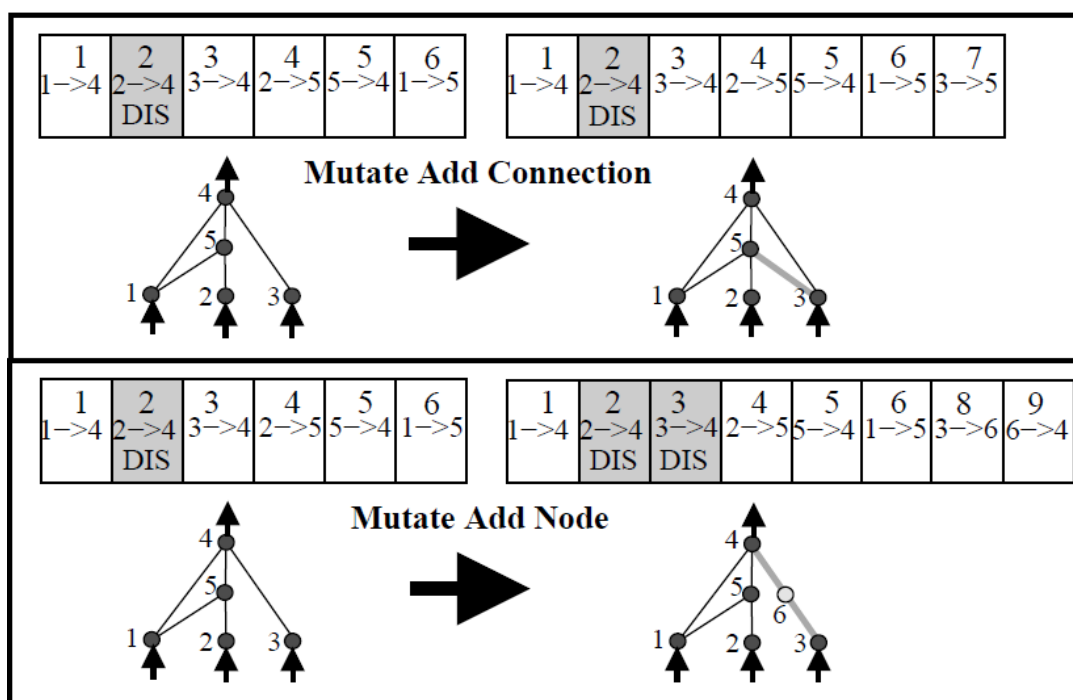


Slika 5. Primjer mapiranja genotipa u fenotip [6]

Mutacija u NEAT-u može promijeniti i težine veze i mrežne strukture. Težine veze mutiraju kao u bilo kojem NE sustavu, pri čemu je svaka veza u svakoj generaciji ili poremećena ili nije. Strukturne mutacije se javljaju na dva načina (slika 6). Svaka mutacija proširuje veličinu genoma dodavanjem jednog ili više gena. U mutaciji gdje se dodaje veza, dodaje se jedan novi gen veza sa nasumičnom težinom koji povezuje dva prethodno nepovezana čvora. U mutaciji gdje se dodaje čvor, postojeća veza se dijeli, a novi čvor se postavlja tamo gdje je bila stara veza. Stara veza je onemogućena i genomu se dodaju dvije nove veze. Nova veza koja vodi u novi čvor dobiva težinu jedan, a nova veza koja vodi van dobiva istu težinu kao i stara veza.

Ova metoda dodavanja čvorova odabrana je kako bi se minimizirao početni učinak mutacije. Nova nelinearnost u vezi neznatno mijenja funkciju, ali se novi čvorovi mogu odmah integrirati u mrežu, za razliku od dodavanja vanjske strukture koja bi morala biti kasnije evoluirana u mrežu. Na ovaj način, zbog specijacije, mreža će imati vremena za optimizaciju i korištenje nove strukture.

Na slici 6., preuzetoj iz [6], prikazana su dva tipa strukturne mutacije u NEAT-u: mutacija dodavanja veze i mutacija dodavanja čvora. Objе vrste, dodavanje veze i dodavanje čvora, ilustrirani su prikazanim genima veza mreže iznad njihovih fenotipova. Najveći broj u svakom genomu je broj inovacije tog gena. Brojevi inovacija povijesni su biljezi koji identificiraju izvornog povijesnog pretka svakog gena. Novim genima se pripisuju novi, sve veći brojevi. Prilikom dodavanja veze, jedan novi gen veze dodaje se na kraj genoma pripisuje mu se sljedeći dostupni broj inovacije. Prilikom dodavanja novog čvora, gen veze koji se dijeli je onemogućen, a dva nova gena veze dodaju se na kraj genoma. Novi čvor je između dvije nove veze. Novi gen čvora (nije prikazan) koji predstavlja ovaj novi čvor također je dodan u genom.



Slika 6. Dva tipa strukturne mutacije u NEAT-u: mutacija dodavanja veze i mutacija dodavanja čvora [6]

Kroz mutaciju, genomi u NEAT-u će se postupno povećavati. Genomi različitih veličina će nastati, ponekad s različitim vezama na istim pozicijama. Najsloženiji oblik problema konkurentskih konvencija, s brojnim različitim topologijama i kombinacijama težina, neizbježan je rezultat dopuštanja genomima da rastu neograničeno. Kako NE može križati genome različitih veličina na razuman način? Sljedeći odjeljak objašnjava kako NEAT rješava ovaj problem.

5.2 PRAĆENJE GENA KROZ POVIJESNE OZNAKE

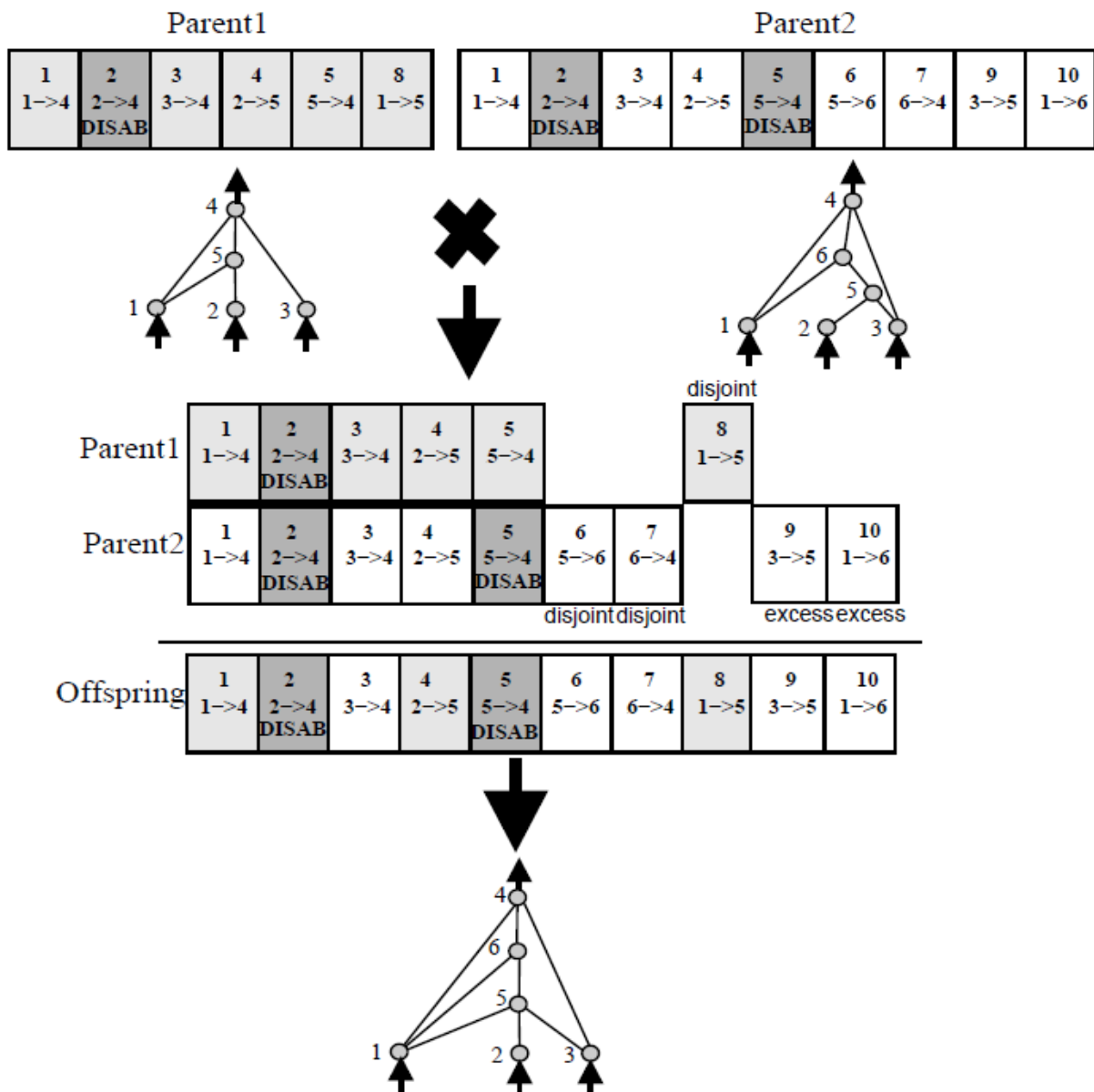
U evoluciji postoje neiskorištene informacije koje nam govore koji se geni točno podudaraju s kojim genima između bilo kojih pojedinaca u topološki raznolikoj populaciji. Ta informacija je povijesno podrijetlo svakog gena. Dva gena s istim povijesnim podrijetlom mora predstavljati istu strukturu (iako moguće s različitim težinama), budući da su oba izvedena od istog gena pretka u nekom trenutku u prošlosti. Dakle, sve što sustav treba učiniti kako bi znao koji se geni slažu s kojima je pratiti povijesno podrijetlo svakog gena u sustavu.

Praćenje povijesnog podrijetla zahtijeva vrlo malo računanja. Kad god se novi gen pojavi (putem strukturne mutacije), povećava se globalni broj inovacije i dodijeljen tom genu. Brojevi inovacija stoga predstavljaju kronologiju pojave svakog gena u sustavu. Kao primjer, recimo da su se dvije mutacije sa slike 6 javljale jedna za drugom u sustavu. Novi gen veze stvoren u prvoj mutaciji je dodijeljen broj 7, a dodana su dva nova gena veza tijekom nove mutacije čvora dodjeljuju se brojevi 8 i 9. U budućnosti, kad god se ovi genomi pare, potomci će naslijediti isti inovacijski broj svakog gena; brojevi inovacija se nikada ne mijenjaju. Dakle, povijesno podrijetlo svakog gena u sustavu poznat je tijekom evolucije.

Mogući problem je da će ista strukturna inovacija dobiti različite inovacijske brojeve u istoj generaciji ako se slučajno dogodi više puta. Međutim, vođenjem popisa inovacija koje su se dogodile u trenutnoj generaciji, moguće je osigurati da kada ista struktura nastane više puta kroz neovisne mutacije u istoj generaciji, svakoj identičnoj mutaciji se dodjeljuje isti inovacijski broj. Dakle, nema rezultatne eksplozije inovacijskih brojeva.

Povijesne oznake daju NEAT-u moćnu novu sposobnost. Sustav sada zna točno koji se geni podudaraju s kojima (slika 7). Prilikom križanja, geni u oba genoma s istim inovacijskim brojem su postrojani. Ovi geni nazivaju se podudarni geni. Geni koji se ne podudaraju su ili nepovezani ili suvišni, ovisno o tome da li se pojavljuju unutar ili izvan raspona inovacijskog broja drugog roditelja. Oni predstavljaju strukturu koja nije prisutna u drugom genomu. U stvaranju potomstva, geni su nasumično odabrani od bilo kojeg roditelja u podudarnim genima, dok su svi suvišni ili nepovezani geni uvijek uključeni od roditelja s boljim fitnessom. Na ovaj način, povijesne oznake omogućuju NEAT-u da izvede križanje koristeći linearne genome bez potrebe za skupom topološkom analizom.

Na slici 7., preuzetoj iz [6], prikazano je usklađivanje genoma za različite mrežne topologije korištenjem inovacijskih brojeva. Iako Roditelj 1 i Roditelj 2 izgledaju drugačije, njihovi inovacijski brojevi (prikazano na vrhu svakog gena) govore nam koji se geni podudaraju s kojim. Čak i bez topološke analiza, može se stvoriti nova struktura, koja kombinira podudarne dijelove oba roditelja kao i njihove različite dijelove. Podudarni geni se nasljeđuju nasumično, dok se nepovezani geni (oni koji se ne podudaraju u sredini) i suvišni geni (oni koji se na kraju ne podudaraju) nasljeđuju od roditelja s boljim fitnessom. U ovom slučaju pretpostavlja se jednaki fitness, pa se nepovezani i suvišni geni također nasljeđuju nasumično. Onemogućeni geni mogli bi ponovno postati omogućeni u budućim generacijama: postoji unaprijed postavljena šansa da je naslijeđeni gen onemogućen ako je onemogućen u bilo kojem roditelju.



Slika 7. Usklađivanje genoma za različite mrežne topologije korištenjem inovacijskih brojeva [6]

Dodavanjem novih gena populaciji i razumnim parenjem genoma koji predstavljaju različite strukture, sustav može formirati populaciju različitih topologija. Međutim, ispada da takva populacija sama po sebi ne može održavati topološke inovacije. Budući da se manje strukture optimiziraju brže od većih struktura, a kako dodavanje čvorova i veza obično u početku smanjuje fitnes mreže, nedavno povećane strukture imaju malo nade da će preživjeti više od jedne generacije iako bi inovacije koje predstavljaju mogle biti ključne za dugoročno rješavanje zadatka. Rješenje je zaštititi inovaciju specijacijom populacije, kao što je objašnjeno u sljedećem odjeljku.

5.3 ZAŠTITA INOVACIJE KROZ SPECIACIJU

Specijacija populacije omogućuje organizmima da se natječu prvenstveno unutar svoje niše umjesto sa populacijom u cjelini. Na taj su način topološke inovacije zaštićene u novoj niši gdje imaju vremena optimizirati svoju strukturu kroz natjecanje unutar niše. Ideja je podijeliti populaciju na takve vrste da su slične topologije u istoj vrsti. Čini se da je ovaj zadatak problem topološkog podudaranja. Međutim, opet se pokazuje da povijesne oznake nude učinkovito rješenje.

Broj suvišnih i nepovezanih gena između para genoma je prirodna mjera njihove udaljenosti kompatibilnosti. Što su dva genoma više nepovezana, to je manje evolucijske povijesti koju dijele, stoga su manje kompatibilni. Stoga, moguće je izmjeriti udaljenost kompatibilnosti δ različitih struktura u NEAT-u kao jednostavnu linearnu kombinaciju broja suvišnih gena E i nepovezanih gena D , kao i prosjek razlike težina odgovarajućih gena \bar{W} , uključujući onesposobljene gene:

$$\delta = \frac{c_1 E}{N} + \frac{c_2 D}{N} + c_3 \cdot \bar{W} \quad (1)$$

Koeficijenti c_1 , c_2 i c_3 omogućuju nam da prilagodimo važnost tri čimbenika, i faktor N , broj gena u većem genomu, normalizira se za veličinu genoma (N se može postaviti na 1 ako su oba genoma mala, i.e. sastoje se od manje od 20 gena).

Mjera udaljenosti δ omogućuje nam specijaciju pomoću praga kompatibilnosti δ_t . Održava se uređeni popis vrsta. U svakoj generaciji genomi su sekvencijalno stavljeni u vrste. Svaka postojeća vrsta predstavljena je slučajnim genomom unutar vrsta iz prethodne generacije. Neki genom g u trenutnoj generaciji je smješten u prvu vrstu u kojoj je g kompatibilan s reprezentativnim genomom te vrste. Na taj se način vrste ne preklapaju. Ako g nije kompatibilan s bilo kojom postojećom vrstom, stvara se nova vrsta s g kao predstavnikom. Također je moguće odrediti kompatibilnost genoma g s vrstom s pomoću prosjeka kompatibilnost g sa svakim genomom u vrsti s , ali u praksi, samo usporedba s prvim genomom u s je dovoljna i potrebno je konstantno vrijeme.

Kao mehanizam reprodukcije za NEAT, koristi se eksplicitno dijeljenje fitnessa (Goldberg and Richardson, 1987), pri čemu organizmi u istoj vrsti moraju dijeliti fitness njihove niše. Stoga si vrsta ne može priuštiti da postane prevelika čak i ako mnogi njezini organizmi rade dobro. Stoga je malo vjerojatno da će bilo koja vrsta preuzeti cijelu populaciju, što je presudno za specijacijsku evoluciju. Prilagođena fitness funkcija f'_i za organizam i izračunava se prema njegovoj udaljenosti δ od svakog drugog organizma j u populaciji:

$$f'_i = \frac{f_i}{\sum_{j=1}^n \sinh(\delta(i,j))} \quad (2)$$

Funkcija dijeljenja sh je postavljena na nulu kada je udaljenost $\delta(i, j)$ iznad praga δ_t ; inače, $sh(\delta(i, j))$ je postavljeno na jedan. Dakle, $\sum_{j=1}^n \sinh(\delta(i, j))$ se svodi na broj organizama u istoj vrsti kao organizam i . Ovo smanjenje je prirodno jer vrste su već grupirane prema kompatibilnosti pomoću praga δ_t . Svakoj vrsti je dodijeljen potencijalno drugačiji broj potomaka proporcionalno zbroju prilagođenih fitnessa f'_i članova organizma. Vrste se potom razmnožavaju eliminacijom članova s najslabijim rezultatima iz populacije. Cijela populacija je tada zamjenjena potomcima preostalih organizama u svakoj vrsti. U rijetkim slučajevima kada se fitness cjelokupne populacije ne poboljša više od 20 generacija, samo dvije najbolje vrste se smiju razmnožavati, preusmjeravajući potragu u najperspektivnije prostore.

Neto željeni učinak specijacije populacije je zaštita topološke inovacije. Konačni cilj sustava je, dakle, izvršiti traženje rješenja što učinkovitije. Ovaj cilj se postiže minimiziranjem dimenzionalnosti prostora za pretraživanje.

5.4 MINIMIZIRANJE DIMENZIONALNOSTI KROZ INKREMENTALNI RAST IZ MINIMALNE STRUKTURE

TWEANN-ovi obično počinju s početnom populacijom od nasumičnih topologija kako bi se uvela raznolikost od samog početka. Suprotno tome, NEAT koristi pristranost pretraživanja prema prostorima minimalnih dimenzija počevši s uniformnom populacijom mreža s nula skrivenih čvorova (i.e. svi ulazi se povezuju izravno s izlazima). Nova struktura se uvodi postepeno kako dolazi do strukturnih mutacija, i opstaju samo one strukture za koje se utvrdi da su korisne kroz procjenu fitnesa. Drugim riječima, strukturne razrade koje se javljaju u NEAT-u uvijek su opravdane. Kako populacija počinje minimalno, dimenzionalnost prostora za pretraživanje je minimizirana, a NEAT uvijek pretražuje kroz manje dimenzija od ostalih TWEANN-a i NE sustava fiksne topologije. Minimiziranje dimenzionalnosti daje NEAT-u prednost u performansama u odnosu na druge pristupe.

6 IMPLEMENTACIJA NEAT ALGORITMA

NEAT algoritam implementiran je u MATLAB-u, kod je u biti modificirani kod iz [7], vrlo elegantan, a sada i malo kompaktniji, doduše malo sporiji, uz potpuno izmjenjen vizualizacijski dio te kozmetičke i funkcionalne dodatke. Kod je dostupan na GitHub profilu Adama Gaiera.

Parametri za kod su uzeti iz originalnog NEAT članka [6] uz manje modifikacije radi poboljšanja performansi i ubrzanje izvedbe.

Kod će biti ispitan na XOR problemu te problemu zanjihavanja i stabilizacije obrnutog njihala na kolicima.

6.1 XOR PROBLEM

Isključivo ili (XOR) je problem klasifikacije koji se često koristi za testiranje neuronskih mreža. Mreža ima za cilj proizvesti izlaze Y za ulaze X_1, X_2 , a tablica 1 prikazuje tablicu istinitosti.

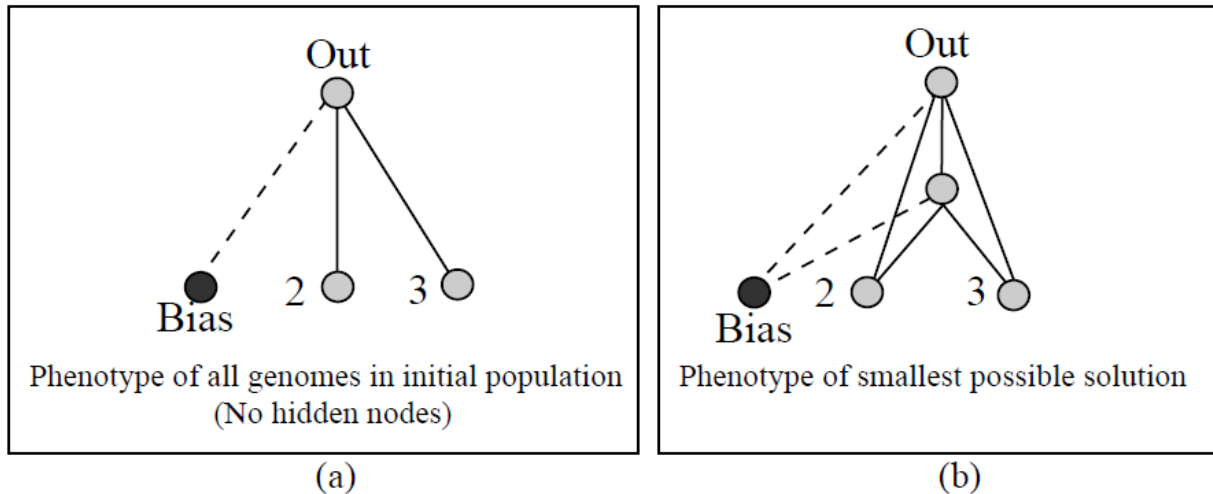
Tablica 1. Tablica istinosti za XOR problem

X_1	X_2	Y
0	0	0
0	1	1
1	0	1
1	1	0

Budući da XOR nije linearno odvojiv, neuronska mreža zahtijeva skrivene jedinice za rješavanje problema. Dva ulaza moraju se kombinirati na nekoj skrivenoj jedinici, za razliku od samo na izlaznom čvoru, jer ne postoji funkcija nad linearnom kombinacijom ulaza koja može razdvojiti ulaze u odgovarajuće klase. Ovi strukturni zahtjevi čine XOR pogodnim za testiranje NEAT-ove sposobnosti da evoluiraju strukturu. Na primjer, NEAT-ova metoda za dodavanje novih čvorova mogla bi biti previše destruktivna da bi omogućila novim čvorovima da uđu u populaciju. Ili, može pronaći lokalnog prvaka s krivom vrstom povezivosti koji dominira populacijom toliko da sustavi ne uspijevaju razviti odgovarajuću povezanost. Treće, možda promjena strukture učini prošle vrijednosti težina veza zastarjelima. U tom bi slučaju algoritam imao problema s povećanjem topologija koje su već uvelike specijalizirane. Ovaj eksperiment ima za cilj pokazati da NEAT nije spriječen takvim potencijalnim preprekama, nego može učinkovito i dosljedno rasti strukturu kada je to potrebno.

Početna generacija sastoji se od mreža bez skrivenih jedinica (slika 8(a)). Mreže su imaju 2 ulaza, 1 *bias* i 1 izlaz. *Bias* je ulaz koji je uvijek postavljeno na 1. U početku su u svakom genomu tri gena za povezivanje populacije. Dva su gena povezuju ulaze s izlazom, a jedan je povezuje *bias* s izlazom. Svaki vezni gen dobio je nasumične težine veze.

Na slici 8., preuzetoj iz [6], prikazan je početni fenotip i optimalni XOR. Slika (a) prikazuje dati fenotip cjelokupnoj početnoj populaciji. Treba primjetiti da nema skrivenih čvorova. U NEAT-u, *bias* je čvor koji se može spojiti na bilo koji čvor osim ulaza. Slika (b) prikazuje optimalno rješenje sa samo jednim skrivenim čvorom. (mreža bez skrivenih čvorova ne može izračunati XOR.) Veze *bias*-a nisu uvijek potrebne ovisno o rješenju; Sve ostale veze su neophodne.



Slika 8. Početni fenotip i optimalni XOR [6]

U [6], na 100 pokušaja, eksperiment pokazuje da NEAT sustav pronalazi strukturu za XOR u prosjeku unutar 32 generacije (4.755 procijenjenih mreža). U prosjeku mreža rješenja imala je 2,35 skrivenih čvorova i 7,48 gena omogućenih veza. Broj čvorova i veza je blizu optimalnog s obzirom da najmanja moguća mreža ima jednu skrivenu jedinicu (slika 8(b)). Optimalno (jedan skriveni čvor) rješenje je pronađeno u 22 od 100 pokušaja (22%). NEAT je vrlo dosljedan u pronalaženju rješenja. Nije podbacio niti jednom u 100 simulacija. Najlošija izvedba imala je 13.459 evaluacija, odnosno oko 90 generacija (u odnosu na 32 generacije u prosjeku). NEAT vrlo dosljedno koristi jedan ili dva skrivena čvora za izgradnju XOR mreže. U zaključku, NEAT rješava XOR problem bez problema i pritom zadržava topologiju malom.

U ovom diplomskom radu uzorak je puno manji, 12 pokušaja, pa autor očekuje da će se, s obzirom na gore spomenute rezultate, optimalno rješenje pojaviti dva puta. Očekivani broj generacija potrebnih za pronalaženje rješenja je između 15 i 140. Potreban broj evaluacija koji se očekuje je između 1500 i 20 000. Pretpostavka je da će prosječno trebati dva do tri skrivena čvora za izgradnju mreže. Također, procjena je da će u prosjeku biti potrebno 7 i 9 omogućenih veza.

Rezultati 10 pokušaja pronalaženja strukture za XOR problem prikazani su u tablici 2.

Tablica 2. Rezultati eksperimenta pronalaženja strukture za XOR problem

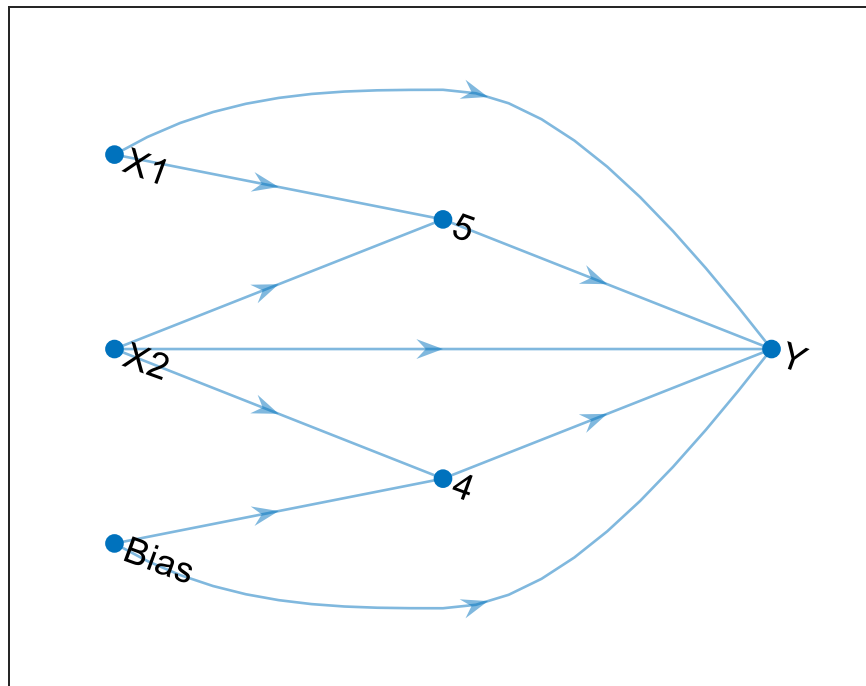
Broj pokušaja	Optimalno rješenje	Broj generacija potrebnih za pronalazak rješenja	Broj procijenjenih mreža potrebnih za pronalazak rješenja	Broj skrivenih čvorova	Broj omogućenih veza
1	Ne	11	1800	2	9
2	Ne	87	13200	13	44
3	Ne	27	4200	2	11
4	Ne	49	7500	5	24
5	Ne	19	3000	2	10
6	Ne	26	4050	3	16
7	Ne	19	3000	4	12
8	Ne	17	2700	2	9
9	Ne	25	3900	3	16
10	Da	7	1200	1	6
11	Ne	11	1800	2	9
12	Da	17	2700	1	7

Kao što se iz Tablice 2. može vidjeti, optimalno rješenje se pojavilo dva puta, kao što je predviđeno. Broj minimalnih potrebnih generacija za pronalaženje rješenja je 7, prema očekivanih 15, što je manje od očekivanog raspona. Maksimalni broj generacija za pronalaženje rješenja je 87, prema očekivanih 140, što je unutar očekivanog raspona. Prosječan broj generacija za pronalaženje rješenja je 26,25. Broj minimalnih procijenjenih mreža potrebnih za pronalaženje rješenja je 1200, prema očekivanih 1500, što je ispod očekivanog raspona. Broj maksimalnih procijenjenih mreža potrebnih za pronalaženje rješenja je 13200, prema očekivanih 20000, što je unutar očekivanog raspona. Prosječan broj procijenjenih mreža za pronalaženje rješenja je 4712,5. Broj potrebnih skrivenih čvorova je u prosjeku 3,3 pa se pretpostavka o dva ili tri čvora pokazala netočnom. U prosjeku je 14,42 potrebnih omogućenih veza te je tako pretpostavka o 7 ili 8 potpuno promašena.

Svi ovi rezultati pokazuju ono što je ljepota evolucijskih algoritama: Optimizacija rješenja problema na načine kojih se ljudski um ne bi nikada mogao dosjetiti.

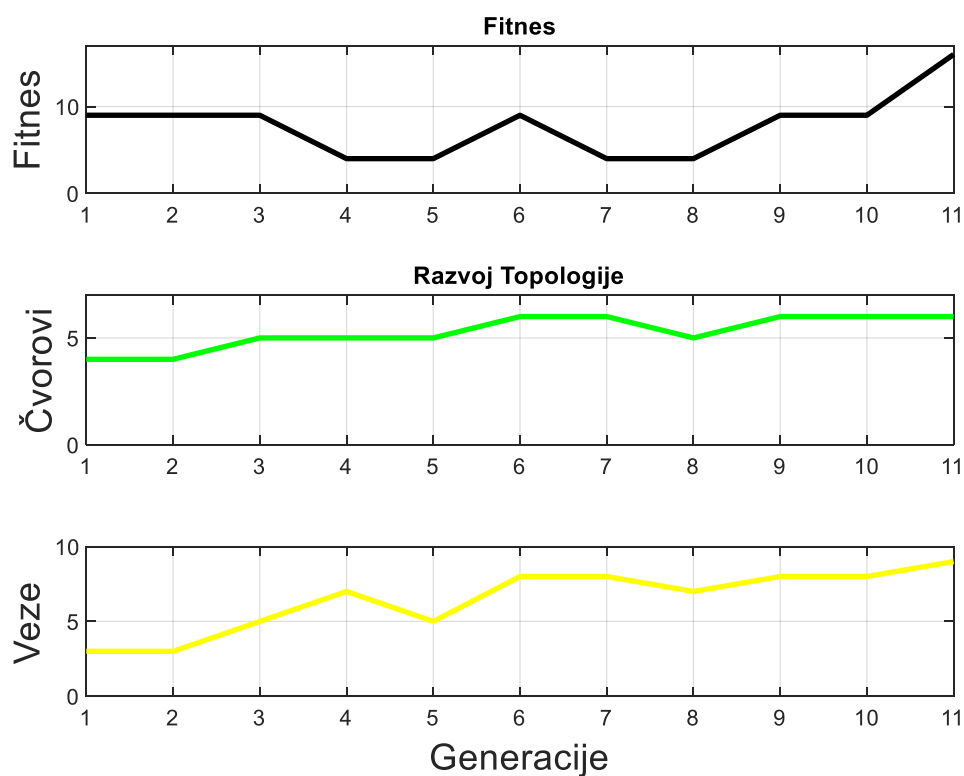
Sada će biti prikazani grafički prikazi mreža rješenja, grafovi razvoja topologije, odnosno grafovi broja skrivenih čvorova te broja omogućenih veza i grafovi fitnesa tijekom generacija potrebnih za rješavanje XOR problema, nastalih nakon 12 pokušaja, na slikama 9. do 32..

Grafički prikaz neuronske mreže



Slika 9. Mreža rješenja XOR problema za prvi pokušaj

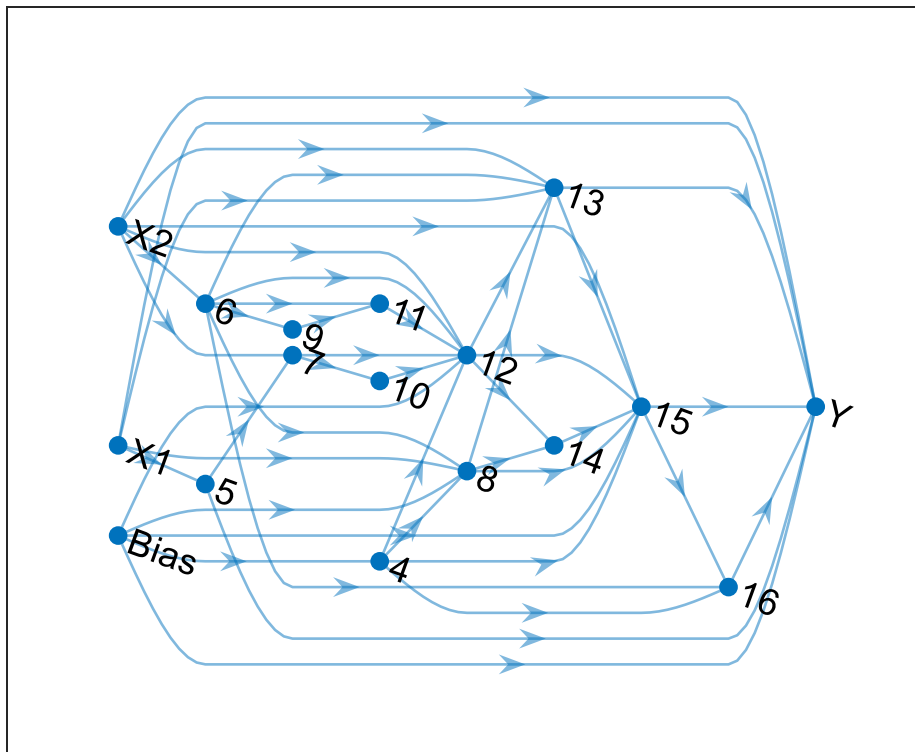
Na slici 9. može se vidjeti mreža koja je u prvom pokušaju riješila XOR problem. Mreža nije optimalna, ali je na dobrom putu. Mreža je nastala nakon 11 generacija i 1800 procijenjenih mreža. Mreža ima ukupno 6 čvorova, od kojih su dva ulazna, *bias*, jedan izlazni i dva skrivena čvora te 9 omogućenih veza. Prvi ulaz X_1 direktno je vezan sa izlazom Y te je također vezan sa skrivenim čvorom 5. Drugi ulaz X_2 direktno je vezan sa izlazom Y te je također vezan sa skrivenim čvorovima 4 i 5. *Bias* je direktno vezan sa izlazom Y te je također vezan sa skrivenim čvorom 4. Skriveni čvor 5 povezan je na svom ulazu sa ulazima X_1 i X_2 , a na izlazu je povezan sa izlazom Y . Skriveni čvor 4 povezan je na svom ulazu sa ulazima X_2 i *Bias*-om, a na izlazu je povezan sa izlazom Y . Izlaz Y direktno je povezan sa ulazima X_1 i X_2 , *Bias*-om te skrivenim čvorovima 4 i 5. Zanimljivo je da je ova mreža iz prvog pokušaja ista kao i mreža iz 8. pokušaja, dakle pojavila se dvaput, iako je u 8. pokušaju bilo potrebno 17 generacija i 2700 procijenjenih mreža, a vjerojatnost da se to dogodi je vjerojatno jako mala. Mreža je također vizualno vrlo privlačna i podsjeća na svemirski brod.



Slika 10. Grafovi fitnessa i razvoja topologije mreže XOR problema za prvi pokušaj

Na slici 10. vidi se graf fitnessa i razvoj topologije mreže u prvom pokušaju. Konačni fitness je 16, konačni broj čvorova je 6, a konačni broj omogućenih veza je 9. Broj čvorova se povećava i smanjuje kako generacije prolaze, odnosno kako se izvršava križanje te mutacija čvorova. Broj omogućenih veza se povećava i smanjuje kako generacije prolaze, odnosno kako se izvršava križanje te mutacija veza. Dodavanje čvorova i veza obično u početku smanjuje fitness mreže, a oduzimanje čvorova i veza obično u početku povećava fitness mreže.

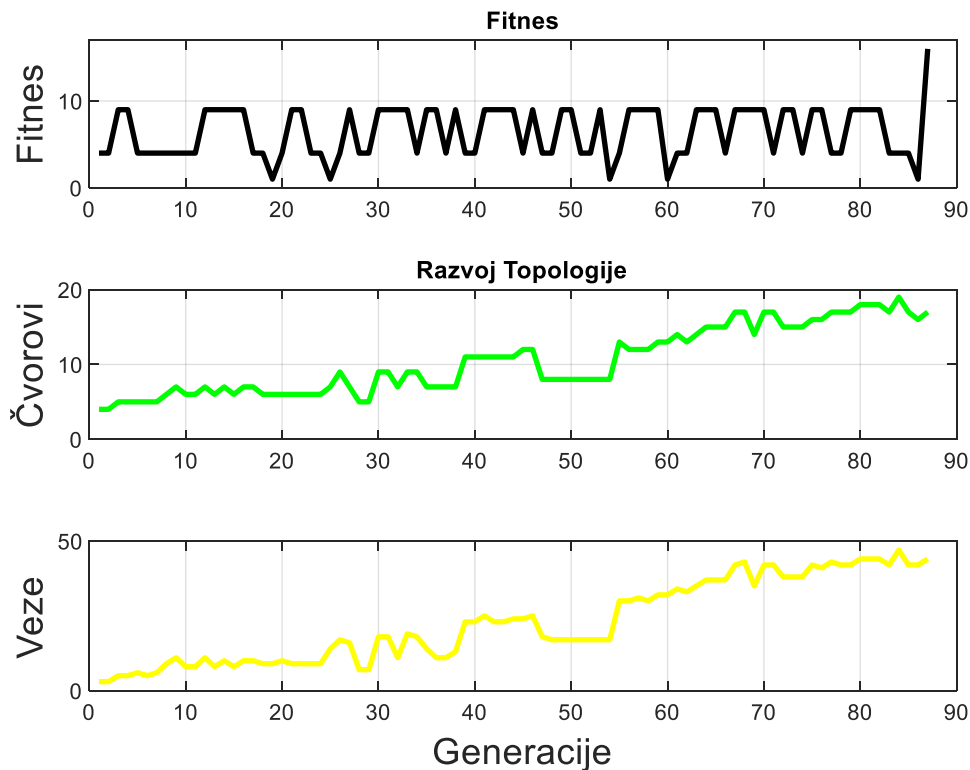
Grafički prikaz neuronske mreže



Slika 11. Mreža rješenja XOR problema za drugi pokušaj

Na slici 11. može se vidjeti mreža koja je u drugom pokušaju riješila XOR problem. Mreža nije niti približno optimalna, vrlo je kompleksna, ali rješava problem. Mreža je nastala nakon 87 generacija i 13200 procijenjenih mreža. Mreža ima ukupno 17 čvorova, od kojih su dva ulazna, *bias*, jedan izlazni i 13 skrivenih čvorova te 44 omogućenih veza. Prvi ulaz X_1 direktno je vezan sa izlazom Y te je također vezan sa skrivenim čvorovima 5, 8 i 13. Drugi ulaz X_2 direktno je vezan sa izlazom Y te je također vezan sa skrivenim čvorovima 6, 7, 12, 13 i 15. *Bias* je direktno vezan sa izlazom Y te je također vezan sa skrivenim čvorovima 4, 8, 12 i 15. Skriveni čvor 6 povezan je na svom ulazu sa ulazima X_1 i X_2 , a na izlazu je povezan sa skrivenim čvorovima 8, 9, 11, 12 i 16. Skriveni čvor 5 povezan je na svom ulazu sa ulazima X_1 , a na izlazu je povezan sa izlazom Y te skrivenim čvorom 7. Skriveni čvor 9 povezan je na svom ulazu sa skrivenim čvorom 6, a na izlazu je povezan sa skrivenim čvorom 14. Skriveni čvor 7 povezan je na svom ulazu sa ulazom X_2 te skrivenim čvorom 5, a na izlazu je povezan sa skrivenim čvorovima 10 i 12. Skriveni čvor 11 povezan je na svom ulazu sa skrivenim čvorovima 6 i 9, a na izlazu je povezan sa skrivenim čvorom 12. Skriveni čvor 10 povezan je na svom ulazu sa skrivenim čvorom 7, a na izlazu je povezan sa skrivenim čvorom 12. Skriveni čvor 4 povezan je na svom ulazu sa *Bias*-om, a na izlazu je povezan sa skrivenim čvorovima 8, 13 i 16. Skriveni čvor 12 povezan je na svom ulazu sa ulazom X_2 i *Bias*-om te skrivenim čvorovima 4, 6, 7, 10 i 11, a na izlazu je povezan sa skrivenim čvorovima 13, 14 i 15. Skriveni čvor 8 povezan je na svom ulazu sa ulazom X_1 i *Bias*-om te skrivenim čvorovima 4 i 6, a na izlazu je povezan sa skrivenim čvorovima 13, 14 i 15. Skriveni čvor 13 povezan je na svom ulazu sa ulazom X_1 i X_2 te skrivenim čvorovima 6, 8 i 12, a na izlazu je povezan sa izlazom Y te skrivenim čvorom 15. Skriveni čvor 14 povezan je na svom ulazu sa čvorovima 8 i 12, a na izlazu je povezan sa skrivenim čvorom 15. Skriveni čvor 15 povezan je na svom ulazu sa ulazom X_2 i *Bias*-om te skrivenim čvorovima 4, 8, 12, 13 i 14, a na izlazu je povezan sa sa izlazom Y te skrivenim čvorom 16. Skriveni čvor

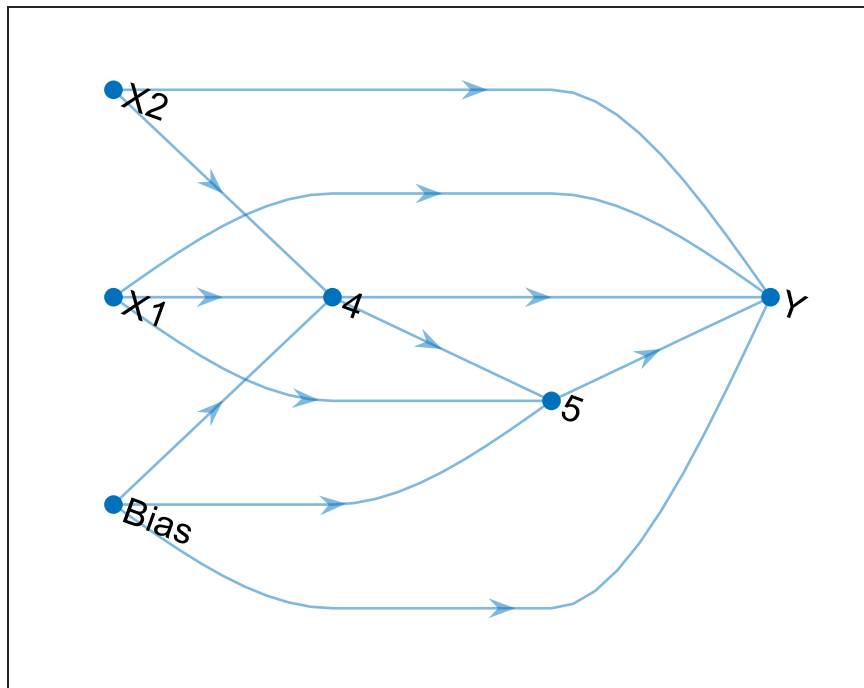
16 povezan je na svom ulazu sa skrivenim čvorovima 4, 6 i 15, a na izlazu je povezan sa izlazom Y . Izlaz Y direktno je povezan sa ulazima X_1 i X_2 , Bias-om te skrivenim čvorovima 5, 13, 15 i 16.



Slika 12. Grafovi fitnessa i razvoja topologije mreže XOR problema za drugi pokušaj

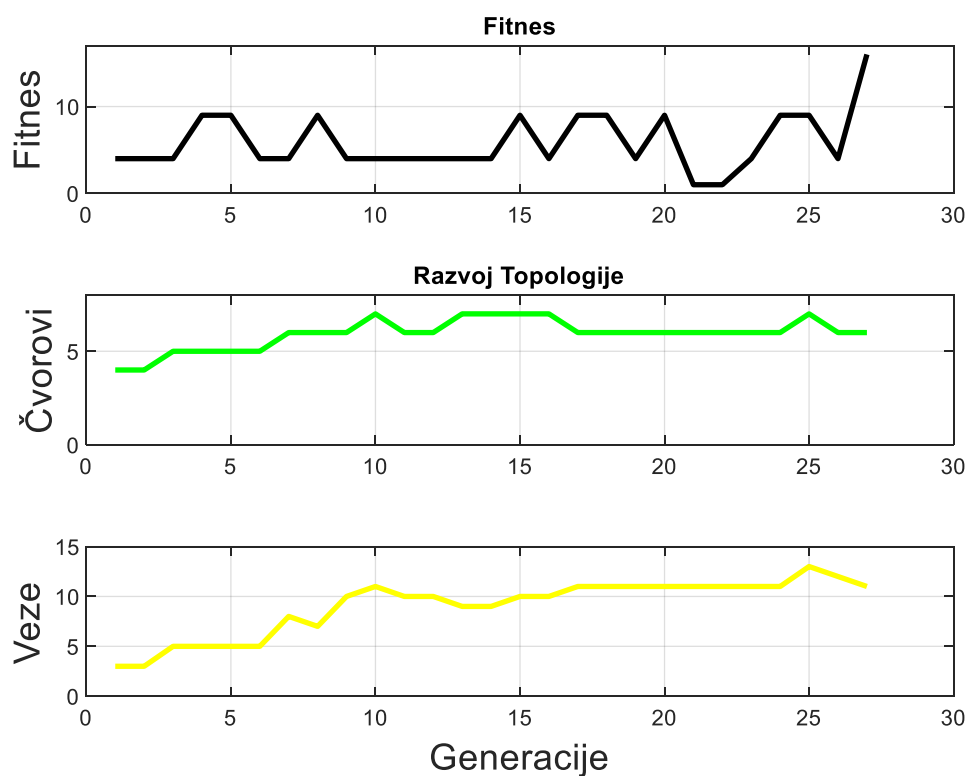
Na slici 12. vidi se graf fitnessa i razvoj topologije mreže u prvom pokušaju. Konačni fitness je 16, konačni broj čvorova je 17, a konačni broj omogućenih veza je 44. Broj čvorova se povećava i smanjuje kako generacije prolaze, odnosno kako se izvršava križanje te mutacija čvorova. Broj omogućenih veza se povećava i smanjuje kako generacije prolaze, odnosno kako se izvršava križanje te mutacija veza. Dodavanje čvorova i veza obično u početku smanjuje fitness mreže, a oduzimanje čvorova i veza obično u početku povećava fitness mreže.

Grafički prikaz neuronske mreže



Slika 13. Mreža rješenja XOR problema za treći pokušaj

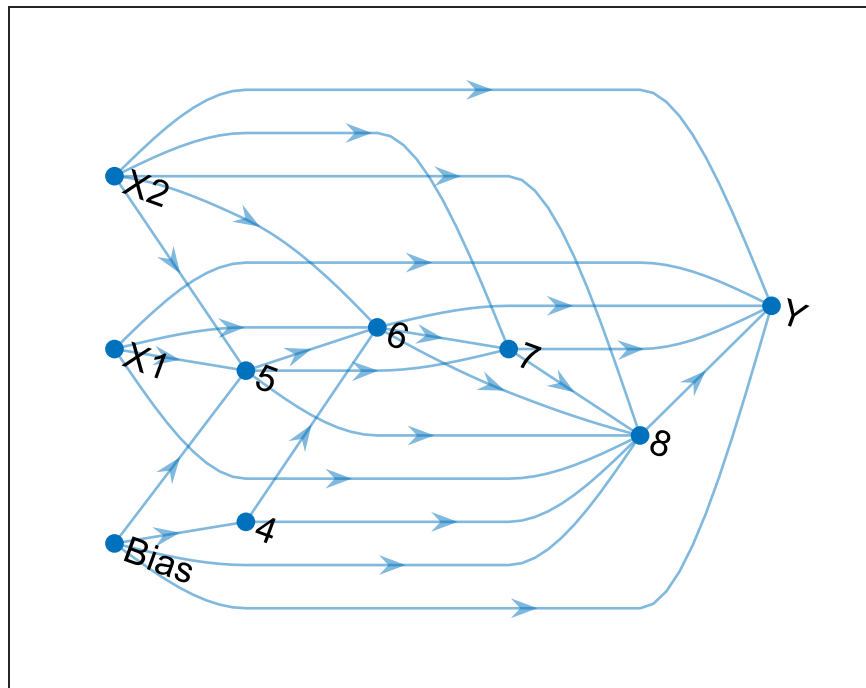
Na slici 13. može se vidjeti mreža koja je u trećem pokušaju riješila XOR problem. Mreža nije optimalna, ali je na dobrom putu. Mreža je nastala nakon 27 generacija i 4200 procijenjenih mreža. Mreža ima ukupno 6 čvorova, od kojih su dva ulazna, *bias*, jedan izlazni i dva skrivena čvora te 11 omogućenih veza. Prvi ulaz X_1 direktno je vezan sa izlazom Y te je također vezan sa skrivenim čvorovima 4 i 5. Drugi ulaz X_2 direktno je vezan sa izlazom Y te je također vezan sa skrivenim čvorom 4. *Bias* je direktno vezan sa izlazom Y te je također vezan sa skrivenim čvorovima 4 i 5. Skriveni čvor 4 povezan je na svom ulazu sa ulazima X_1 i X_2 te *Bias*-om, a na izlazu je povezan sa izlazom Y i skrivenim čvorom 5. Skriveni čvor 5 povezan je na svom ulazu sa ulazom X_1 i *Bias*-om, a na izlazu je povezan sa izlazom Y . Izlaz Y direktno je povezan sa ulazima X_1 i X_2 , *Bias*-om te skrivenim čvorovima 4 i 5.



Slika 14. Grafovi fitnessa i razvoja topologije mreže XOR problema za treći pokušaj

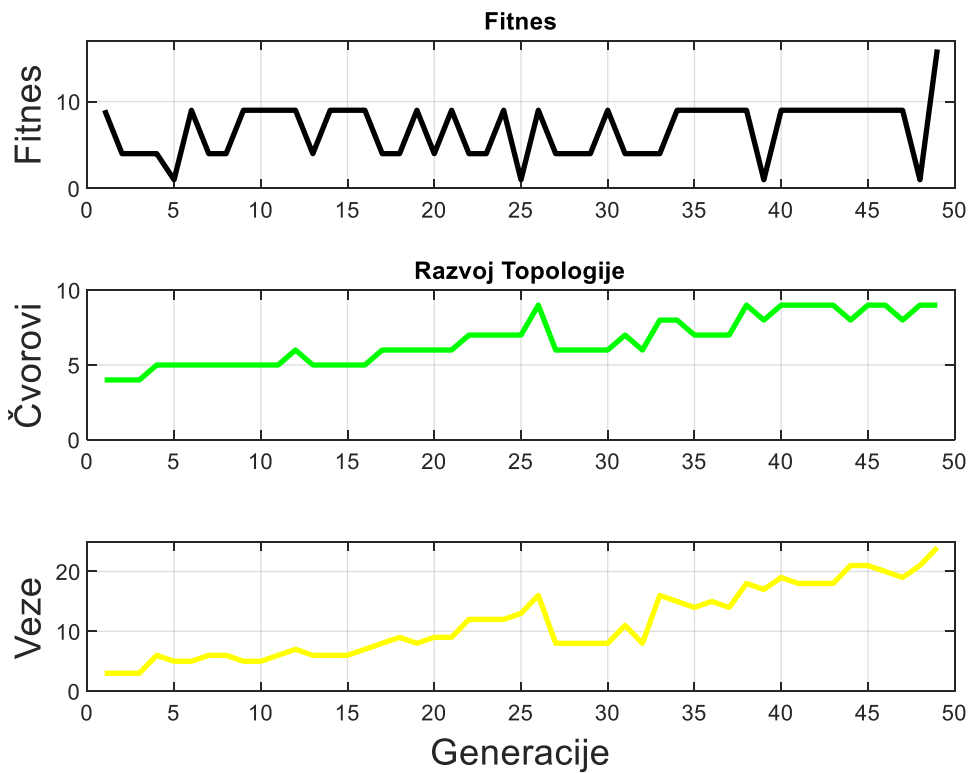
Na slici 14. vidi se graf fitnessa i razvoj topologije mreže u prvom pokušaju. Konačni fitness je 16, konačni broj čvorova je 6, a konačni broj omogućenih veza je 11. Broj čvorova se povećava i smanjuje kako generacije prolaze, odnosno kako se izvršava križanje te mutacija čvorova. Broj omogućenih veza se povećava i smanjuje kako generacije prolaze, odnosno kako se izvršava križanje te mutacija veza. Dodavanje čvorova i veza obično u početku smanjuje fitness mreže, a oduzimanje čvorova i veza obično u početku povećava fitness mreže.

Grafički prikaz neuronske mreže



Slika 15. Mreža rješenja XOR problema za četvrti pokušaj

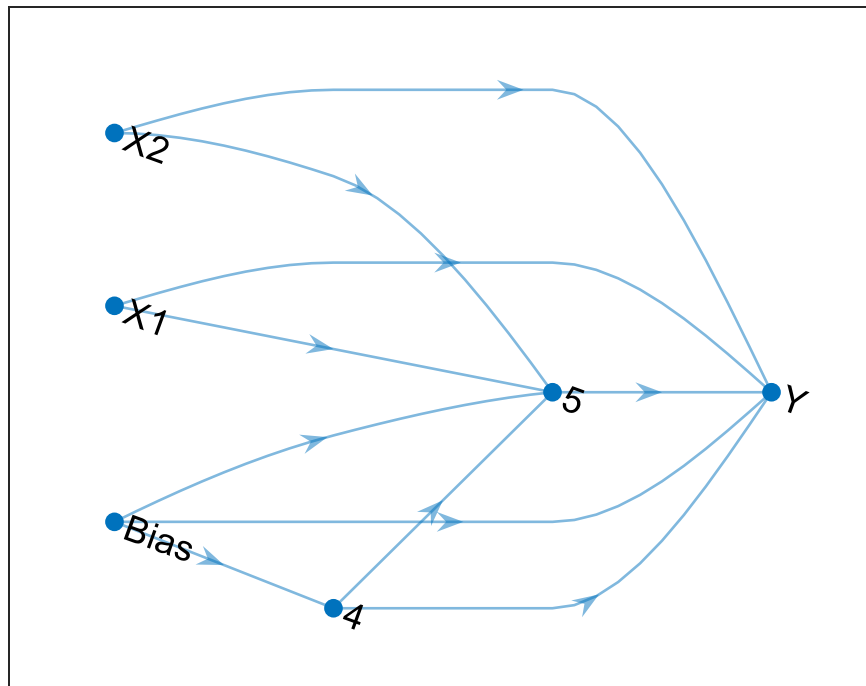
Na slici 15. može se vidjeti mreža koja je u četvrtom pokušaju riješila XOR problem. Mreža nije optimalna ali je dovoljno jednostavna i učinkovita. Mreža je nastala nakon 49 generacija i 7500 procijenjenih mreža. Mreža ima ukupno 9 čvorova, od kojih su dva ulazna, *bias*, jedan izlazni i 5 skrivenih čvorova te 24 omogućenih veza. Prvi ulaz X_1 direktno je vezan sa izlazom Y te je također vezan sa skrivenim čvorovima 5, 6 i 8. Drugi ulaz X_2 direktno je vezan sa izlazom Y te je također vezan sa skrivenim čvorovima 5, 6, 7 i 8. *Bias* je direktno vezan sa izlazom Y te je također vezan sa skrivenim čvorovima 4, 5 i 8. Skriveni čvor 5 povezan je na svom ulazu sa ulazima X_1 i X_2 te *Bias*-om, a na izlazu je povezan sa skrivenim čvorovima 6, 7 i 8. Skriveni čvor 4 povezan je na svom ulazu sa *Bias*-om, a na izlazu je povezan sa skrivenim čvorovima 6 i 8. Skriveni čvor 6 povezan je na svom ulazu sa ulazima X_1 i X_2 te skrivenim čvorovima 4 i 5, a na izlazu je povezan sa izlazom Y te skrivenim čvorovima 7 i 8. Skriveni čvor 7 povezan je na svom ulazu sa ulazom X_2 te skrivenim čvorovima 5 i 6, a na izlazu je povezan sa izlazom Y te skrivenim čvorom 8. Skriveni čvor 8 povezan je na svom ulazu sa ulazima X_1 i X_2 , *Bias*-om te skrivenim čvorovima 4, 5, 6 i 7, a na izlazu je povezan sa izlazom Y . Izlaz Y direktno je povezan sa ulazima X_1 i X_2 , *Bias*-om te skrivenim čvorovima 6, 7 i 8.



Slika 16. Grafovi fitnessa i razvoja topologije mreže XOR problema za četvrti pokušaj

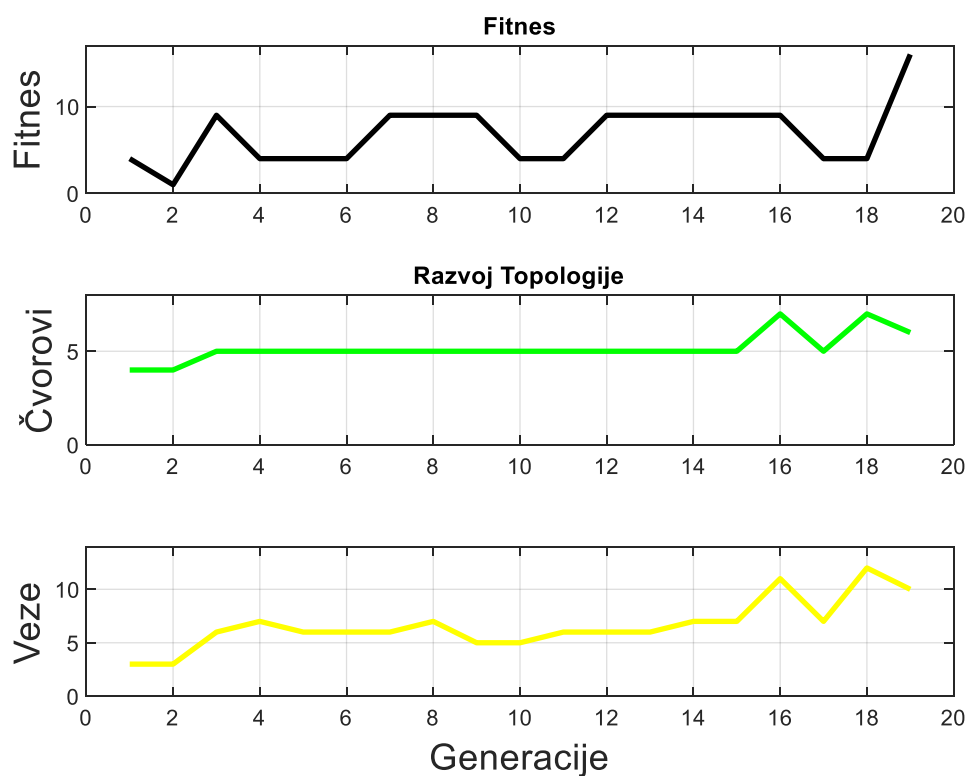
Na slici 16. vidi se graf fitnessa i razvoj topologije mreže u prvom pokušaju. Konačni fitness je 16, konačni broj čvorova je 9, a konačni broj omogućenih veza je 24. Broj čvorova se povećava i smanjuje kako generacije prolaze, odnosno kako se izvršava križanje te mutacija čvorova. Broj omogućenih veza se povećava i smanjuje kako generacije prolaze, odnosno kako se izvršava križanje te mutacija veza. Dodavanje čvorova i veza obično u početku smanjuje fitness mreže, a oduzimanje čvorova i veza obično u početku povećava fitness mreže.

Grafički prikaz neuronske mreže



Slika 17. Mreža rješenja XOR problema za 5. pokušaj

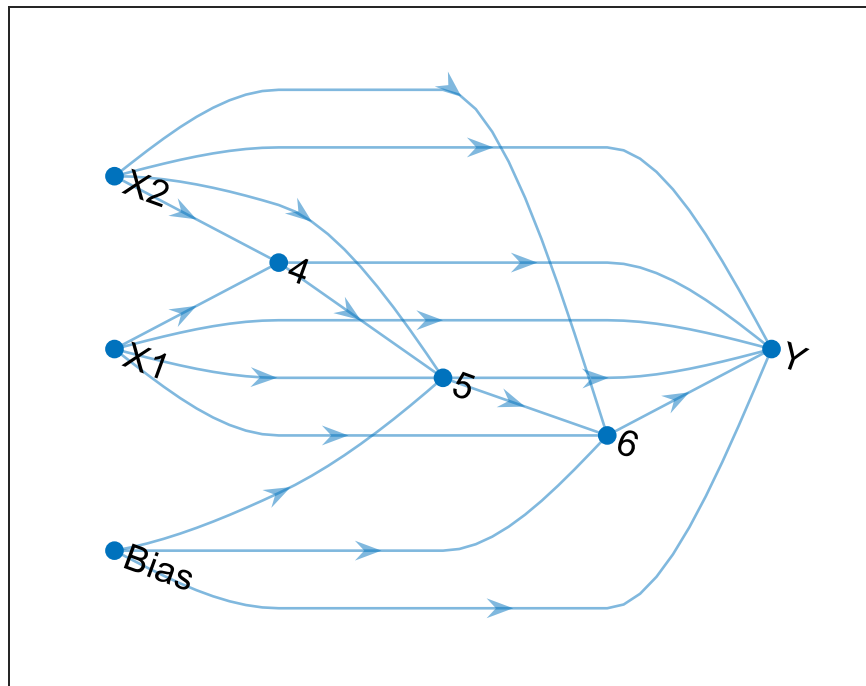
Na slici 17. može se vidjeti mreža koja je u 5. pokušaju riješila XOR problem. Mreža nije optimalna, ali je na dobrom putu. Mreža je nastala nakon 19 generacija i 3000 procijenjenih mreža. Mreža ima ukupno 6 čvorova, od kojih su dva ulazna, *bias*., jedan izlazni i dva skrivena čvora te 10 omogućenih veza. Prvi ulaz X_1 direktno je vezan sa izlazom Y te je također vezan sa skrivenim čvorom 5. Drugi ulaz X_2 direktno je vezan sa izlazom Y te je također vezan sa skrivenim čvorom 5. *Bias* je direktno vezan sa izlazom Y te je također vezan sa skrivenim čvorovima 4 i 5. Skriveni čvor 4 povezan je na svom ulazu sa *Bias*-om, a na izlazu je povezan sa izlazom Y i skrivenim čvorom 5. Skriveni čvor 5 povezan je na svom ulazu sa ulazima X_1 i X_2 , *Bias*-om, te skrivenim čvorom 4, a na izlazu je povezan sa izlazom Y . Izlaz Y direktno je povezan sa ulazima X_1 i X_2 , *Bias*-om te skrivenim čvorovima 4 i 5. Zanimljivo je kod ove mreže što je nastala u 19 generacija i 3000 mreža je procijenjeno, isto kao i mreža iz 7. pokušaja, ali ta mreža ima četiri skrivena čvora i 12 omogućenih veza, dok ova mreža ima dva skrivena čvora i 10 omogućenih veza, što pokazuje da se u istom broju generacija i sa istim brojem procijenjenih mreža može dobiti mreža potpuno drugačije topologije, što je nevjerojatno fascinantno te pokazuje raznovrsnost, moć, ali i nepredvidljivost evolucije.



Slika 18. Grafovi fitnessa i razvoja topologije mreže XOR problema za 5. pokušaj

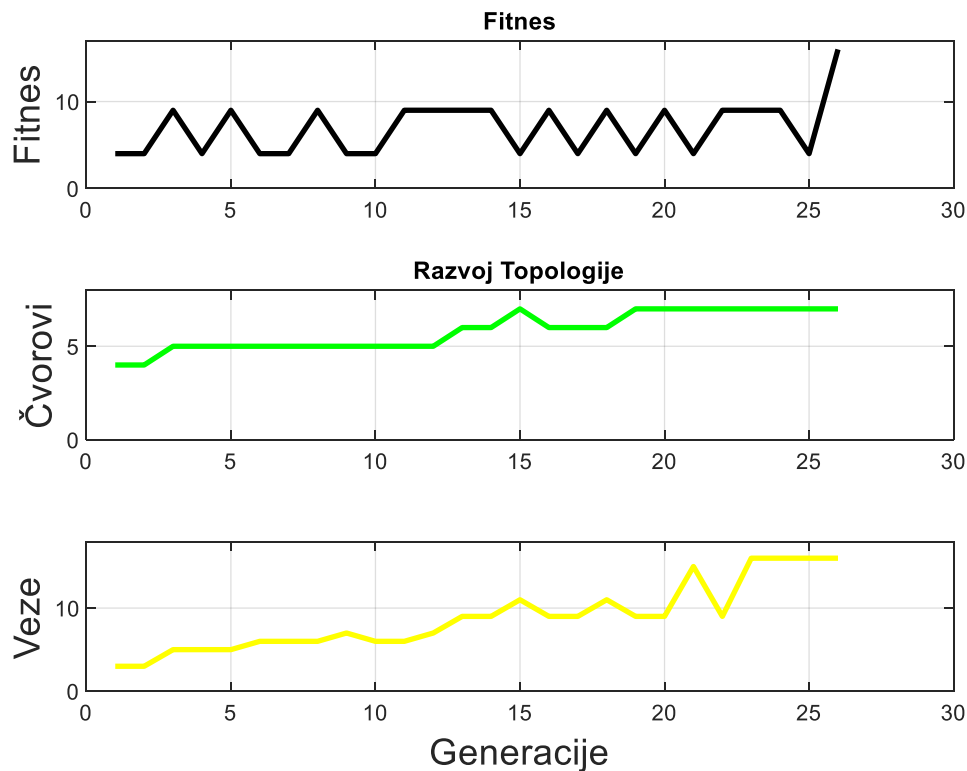
Na slici 18. vidi se graf fitnessa i razvoj topologije mreže u prvom pokušaju. Konačni fitness je 16, konačni broj čvorova je 6, a konačni broj omogućenih veza je 10. Broj čvorova se povećava i smanjuje kako generacije prolaze, odnosno kako se izvršava križanje te mutacija čvorova. Broj omogućenih veza se povećava i smanjuje kako generacije prolaze, odnosno kako se izvršava križanje te mutacija veza. Dodavanje čvorova i veza obično u početku smanjuje fitness mreže, a oduzimanje čvorova i veza obično u početku povećava fitness mreže.

Grafički prikaz neuronske mreže



Slika 19. Mreža rješenja XOR problema za 6. pokušaj

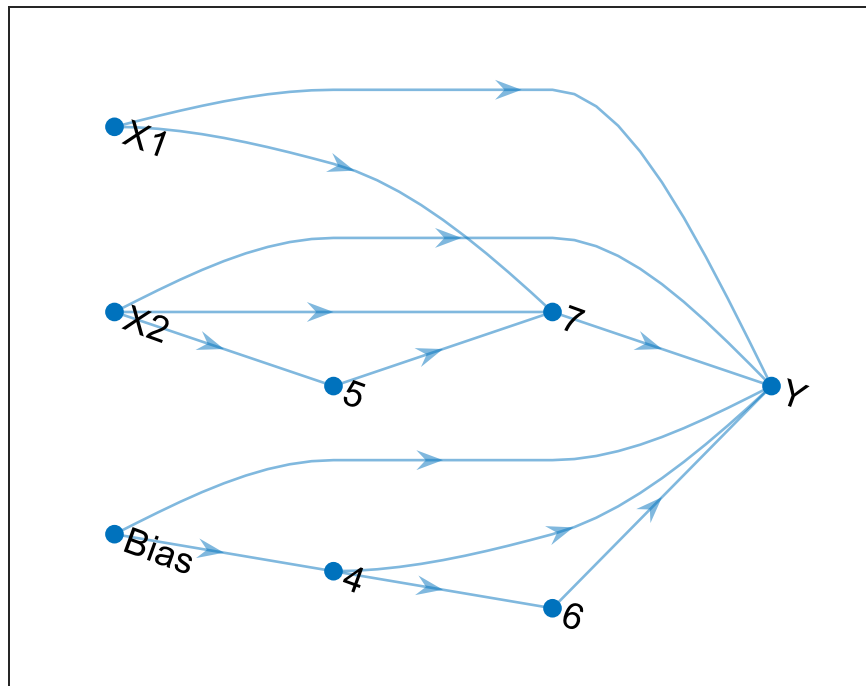
Na slici 19. može se vidjeti mreža koja je u 6. pokušaju riješila XOR problem. Mreža nije optimalna, ali je na dobrom putu. Mreža je nastala nakon 26 generacija i 4050 procijenjenih mreža. Mreža ima ukupno 7 čvorova, od kojih su dva ulazna, *bias*, jedan izlazni i tri skrivena čvora te 16 omogućenih veza. Prvi ulaz X_1 direktno je vezan sa izlazom Y te je također vezan sa skrivenim čvorovima 4, 5 i 6. Drugi ulaz X_2 direktno je vezan sa izlazom Y te je također vezan sa skrivenim čvorovima 4, 5 i 6. *Bias* je direktno vezan sa izlazom Y te je također vezan sa skrivenim čvorovima 5 i 6. Skriveni čvor 4 povezan je na svom ulazu sa ulazima X_1 i X_2 , a na izlazu je povezan sa izlazom Y te skrivenim čvorom 5. Skriveni čvor 5 povezan je na svom ulazu sa ulazima X_1 i X_2 , *Bias*-om, te skrivenim čvorom 4, a na izlazu je povezan sa izlazom Y te skrivenim čvorom 6. Skriveni čvor 6 povezan je na svom ulazu sa ulazima X_1 i X_2 , *Bias*-om, te skrivenim čvorom 5, a na izlazu je povezan sa izlazom Y . Izlaz Y direktno je povezan sa ulazima X_1 i X_2 , *Bias*-om te skrivenim čvorovima 4, 5 i 6.



Slika 20. Grafovi fitnessa i razvoja topologije mreže XOR problema za 6. pokušaj

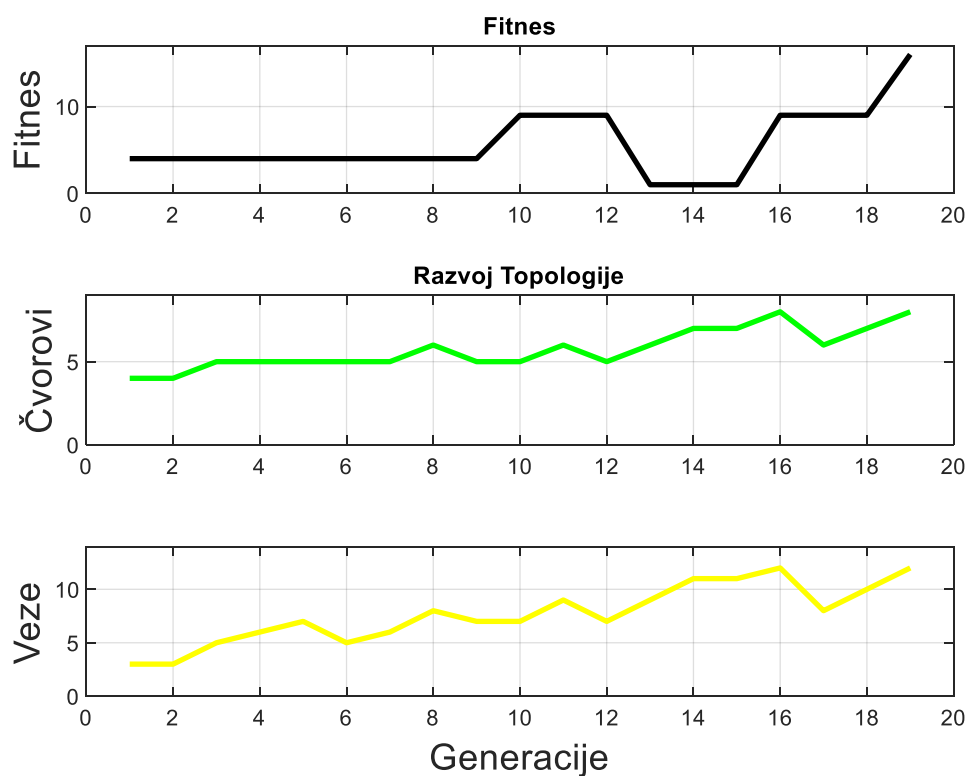
Na slici 20. vidi se graf fitnessa i razvoj topologije mreže u prvom pokušaju. Konačni fitness je 16, konačni broj čvorova je 7, a konačni broj omogućenih veza je 16. Broj čvorova se povećava i smanjuje kako generacije prolaze, odnosno kako se izvršava križanje te mutacija čvorova. Broj omogućenih veza se povećava i smanjuje kako generacije prolaze, odnosno kako se izvršava križanje te mutacija veza. Dodavanje čvorova i veza obično u početku smanjuje fitness mreže, a oduzimanje čvorova i veza obično u početku povećava fitness mreže.

Grafički prikaz neuronske mreže



Slika 21. Mreža rješenja XOR problema za 7. pokušaj

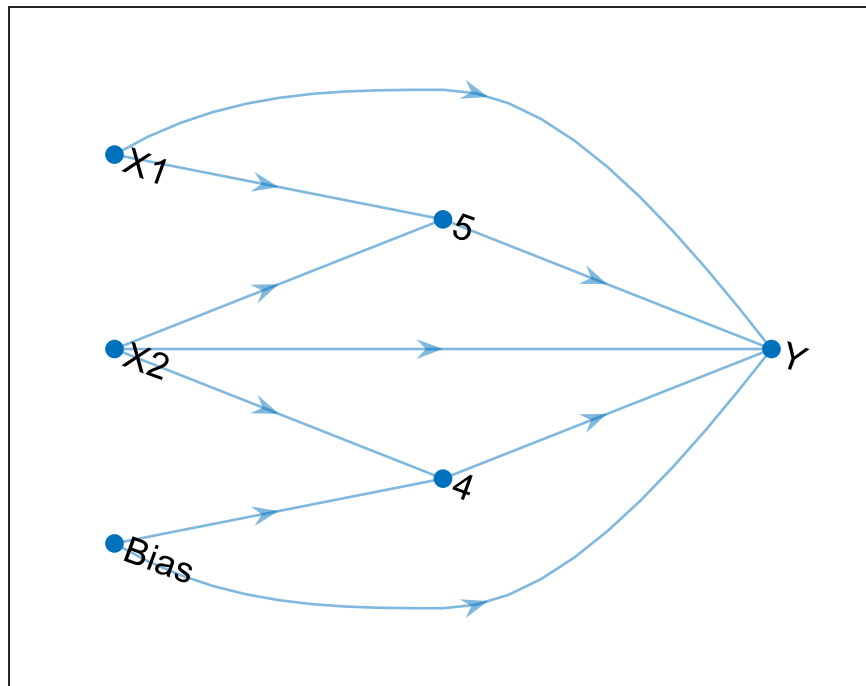
Na slici 21. može se vidjeti mreža koja je u 7. pokušaju riješila XOR problem. Mreža nije optimalna, ali je dovoljno jednostavna i učinkovita. Mreža je nastala nakon 19 generacija i 3000 procijenjenih mreža. Mreža ima ukupno 8 čvorova, od kojih su dva ulazna, *bias*, jedan izlazni i četiri skrivena čvora te 12 omogućenih veza. Prvi ulaz X_1 direktno je vezan sa izlazom Y te je također vezan sa skrivenim čvorom 7. Drugi ulaz X_2 direktno je vezan sa izlazom Y te je također vezan sa skrivenim čvorovima 5 i 7. *Bias* je direktno vezan sa izlazom Y te je također vezan sa skrivenim čvorom 4. Skriveni čvor 5 povezan je na svom ulazu sa ulazom X_2 , a na izlazu je povezan sa skrivenim čvorom 7. Skriveni čvor 4 povezan je na svom ulazu sa *Bias*-om, a na izlazu je povezan sa izlazom Y i skrivenim čvorom 6. Skriveni čvor 7 povezan je na svom ulazu sa ulazima X_1 i X_2 te skrivenim čvorom 5, a na izlazu je povezan sa izlazom Y . Skriveni čvor 6 povezan je na svom ulazu sa skrivenim čvorom 4, a na izlazu je povezan sa izlazom Y . Izlaz Y direktno je povezan sa ulazima X_1 i X_2 , *Bias*-om te skrivenim čvorovima 4, 6 i 7. Zanimljivo je kod ove mreže, kako je već opisano, što je nastala u 19 generacija i 3000 mreža je procijenjeno, isto kao i mreža iz 5. pokušaja, ali ta mreža ima dva skrivena čvora i 10 omogućenih veza, dok ova mreža ima četiri skrivena čvora i 12 omogućenih veza, što pokazuje da se u istom broju generacija i sa istim brojem procijenjenih mreža može dobiti mreža potpuno drugačije topologije, što je nevjerojatno fascinantno te pokazuje raznovrsnost, moć, ali i nepredvidljivost evolucije.



Slika 22. Grafovi fitnessa i razvoja topologije mreže XOR problema za 7. pokušaj

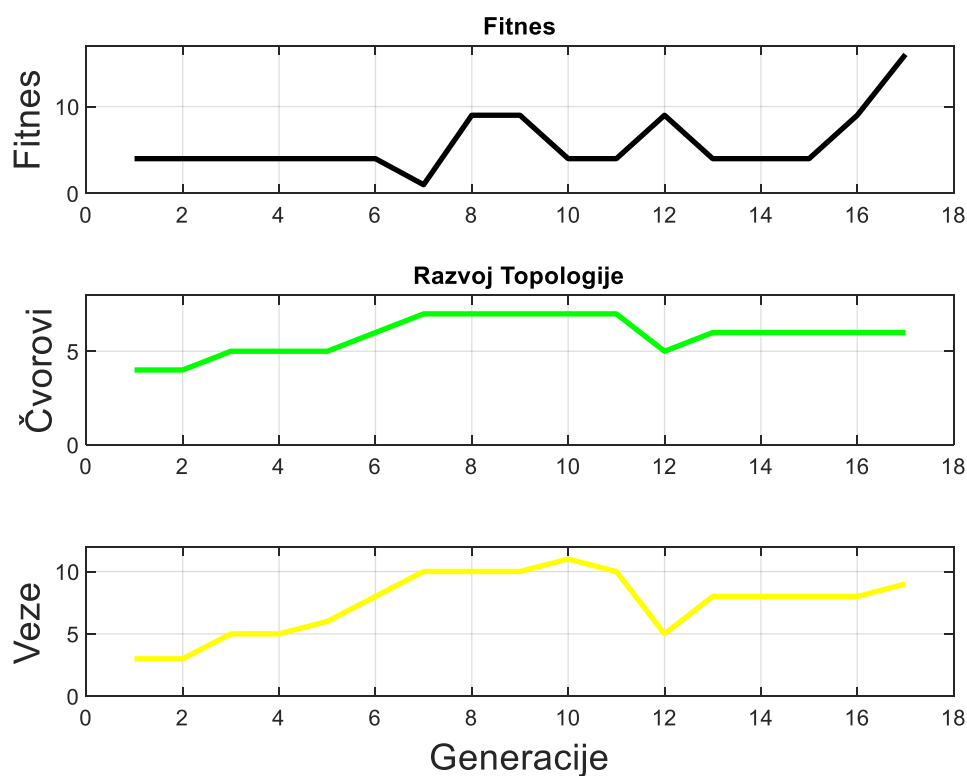
Na slici 22. vidi se graf fitnessa i razvoj topologije mreže u prvom pokušaju. Konačni fitness je 16, konačni broj čvorova je 8, a konačni broj omogućenih veza je 12. Broj čvorova se povećava i smanjuje kako generacije prolaze, odnosno kako se izvršava križanje te mutacija čvorova. Broj omogućenih veza se povećava i smanjuje kako generacije prolaze, odnosno kako se izvršava križanje te mutacija veza. Dodavanje čvorova i veza obično u početku smanjuje fitness mreže, a oduzimanje čvorova i veza obično u početku povećava fitness mreže.

Grafički prikaz neuronske mreže



Slika 23. Mreža rješenja XOR problema za 8. pokušaj

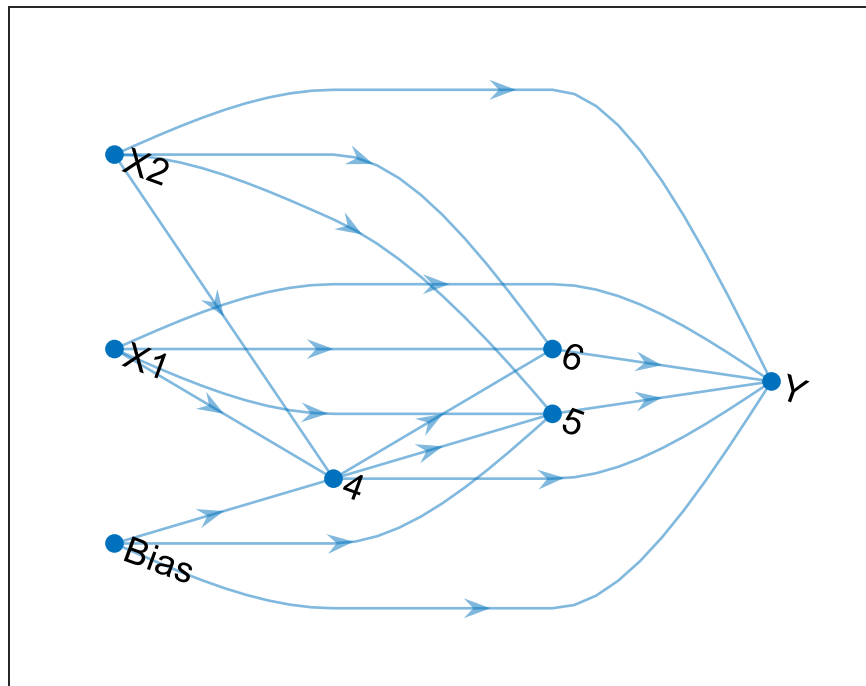
Na slici 23. može se vidjeti mreža koja je u 8. pokušaju riješila XOR problem. Mreža nije optimalna, ali je na dobrom putu. Mreža je nastala nakon 17 generacija i 2700 procijenjenih mreža. Mreža ima ukupno 6 čvorova, od kojih su dva ulazna, *bias*, jedan izlazni i dva skrivena čvora te 9 omogućenih veza. Prvi ulaz X_1 direktno je vezan sa izlazom Y te je također vezan sa skrivenim čvorom 5. Drugi ulaz X_2 direktno je vezan sa izlazom Y te je također vezan sa skrivenim čvorovima 4 i 5. *Bias* je direktno vezan sa izlazom Y te je također vezan sa skrivenim čvorom 4. Skriveni čvor 5 povezan je na svom ulazu sa ulazima X_1 i X_2 , a na izlazu je povezan sa izlazom Y . Skriveni čvor 4 povezan je na svom ulazu sa ulazima X_2 i *Bias*-om, a na izlazu je povezan sa izlazom Y . Izlaz Y direktno je povezan sa ulazima X_1 i X_2 , *Bias*-om te skrivenim čvorovima 4 i 5. Zanimljivo je da je ova mreža iz 8. pokušaja, kako je već spomenuto, ista kao i mreža iz prvog pokušaja, dakle pojavila se dvaput, iako je u prvom pokušaju bilo potrebno 11 generacija i 1800 procijenjenih mreža, a vjerojatnost da se to dogodi je vjerojatno jako mala. Mreža je također vizualno vrlo privlačna i podsjeća na svemirski brod.



Slika 24. Grafovi fitnessa i razvoja topologije mreže XOR problema za 8. pokušaj

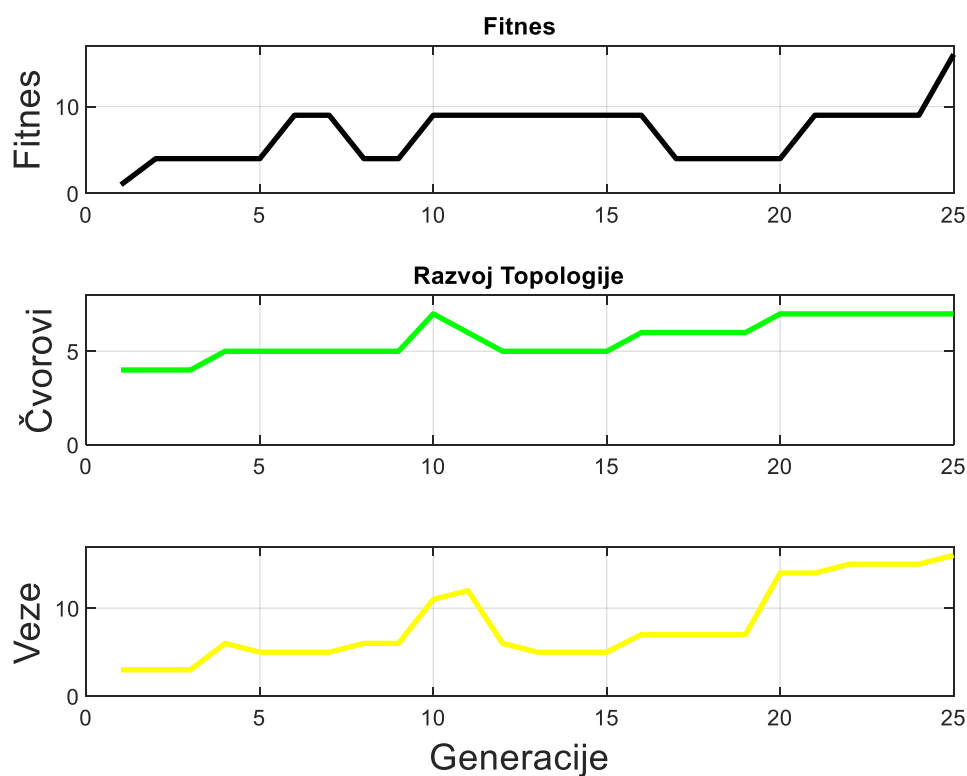
Na slici 24. vidi se graf fitnessa i razvoj topologije mreže u prvom pokušaju. Konačni fitness je 16, konačni broj čvorova je 6, a konačni broj omogućenih veza je 9. Broj čvorova se povećava i smanjuje kako generacije prolaze, odnosno kako se izvršava križanje te mutacija čvorova. Broj omogućenih veza se povećava i smanjuje kako generacije prolaze, odnosno kako se izvršava križanje te mutacija veza. Dodavanje čvorova i veza obično u početku smanjuje fitness mreže, a oduzimanje čvorova i veza obično u početku povećava fitness mreže.

Grafički prikaz neuronske mreže



Slika 25. Mreža rješenja XOR problema za 9. pokušaj

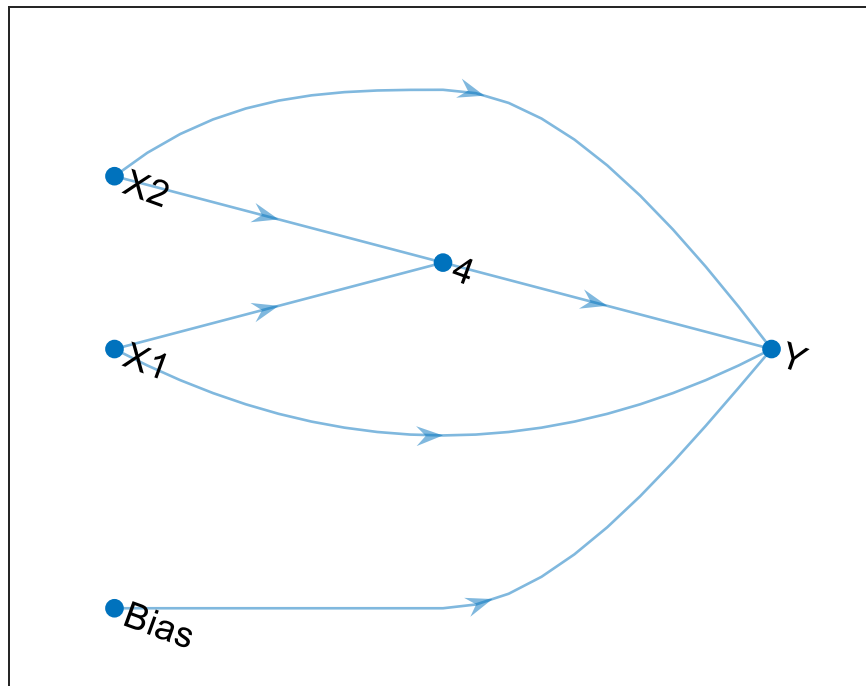
Na slici 25. može se vidjeti mreža koja je u 9. pokušaju riješila XOR problem. Mreža nije optimalna, ali je na dobrom putu. Mreža je nastala nakon 25 generacija i 3900 procijenjenih mreža. Mreža ima ukupno 7 čvorova, od kojih su dva ulazna, *bias*, jedan izlazni i tri skrivena čvora te 16 omogućenih veza. Prvi ulaz X_1 direktno je vezan sa izlazom Y te je također vezan sa skrivenim čvorovima 4, 5 i 6. Drugi ulaz X_2 direktno je vezan sa izlazom Y te je također vezan sa skrivenim čvorovima 4, 5 i 6. *Bias* je direktno vezan sa izlazom Y te je također vezan sa skrivenim čvorovima 4 i 5. Skriveni čvor 4 povezan je na svom ulazu sa ulazima X_1 i X_2 i *Bias*-om, a na izlazu je povezan sa izlazom Y te skrivenim čvorovima 5 i 6. Skriveni čvor 6 povezan je na svom ulazu sa ulazima X_1 i X_2 te skrivenim čvorom 4, a na izlazu je povezan sa izlazom Y . Skriveni čvor 5 povezan je na svom ulazu sa ulazima X_1 i X_2 , *Bias*-om, te skrivenim čvorom 4, a na izlazu je povezan sa izlazom Y . Izlaz Y direktno je povezan sa ulazima X_1 i X_2 , *Bias*-om te skrivenim čvorovima 4, 5 i 6.



Slika 26. Grafovi fitnessa i razvoja topologije mreže XOR problema za 9. pokušaj

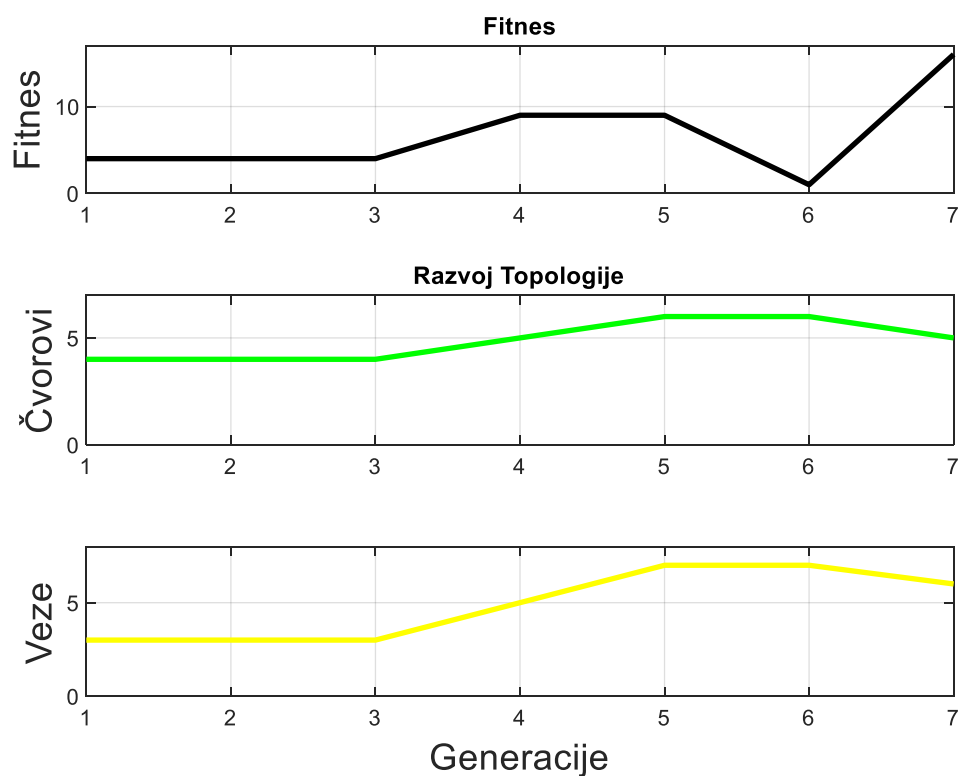
Na slici 26. vidi se graf fitnessa i razvoj topologije mreže u prvom pokušaju. Konačni fitness je 16, konačni broj čvorova je 7, a konačni broj omogućenih veza je 16. Broj čvorova se povećava i smanjuje kako generacije prolaze, odnosno kako se izvršava križanje te mutacija čvorova. Broj omogućenih veza se povećava i smanjuje kako generacije prolaze, odnosno kako se izvršava križanje te mutacija veza. Dodavanje čvorova i veza obično u početku smanjuje fitness mreže, a oduzimanje čvorova i veza obično u početku povećava fitness mreže.

Grafički prikaz neuronske mreže



Slika 27. Mreža rješenja XOR problema za 10. pokušaj

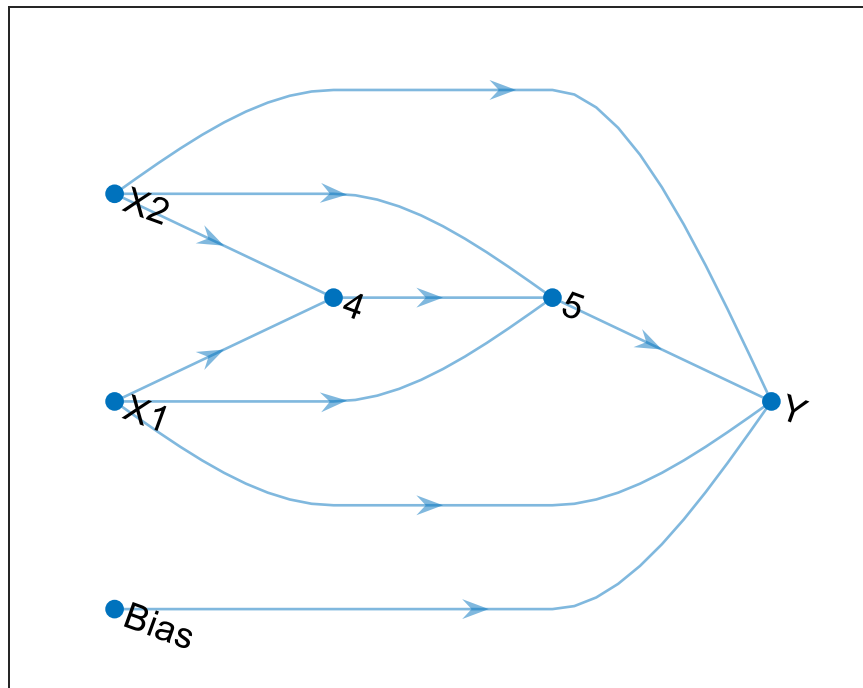
Na slici 27. može se vidjeti mreža koja je u 10. pokušaju riješila XOR problem. Mreža je optimalna. Mreža je nastala nakon 7 generacija i 1200 procijenjenih mreža. Mreža ima ukupno 5 čvorova, od kojih su dva ulazna, *bias*, jedan izlazni i jedan skriveni čvor te 6 omogućenih veza. Prvi ulaz X_1 direktno je vezan sa izlazom Y te je također vezan sa skrivenim čvorom 4. Drugi ulaz X_2 direktno je vezan sa izlazom Y te je također vezan sa skrivenim čvorovima 4. *Bias* je direktno vezan sa izlazom Y . Skriveni čvor 4 povezan je na svom ulazu sa ulazima X_1 i X_2 , a na izlazu je povezan sa izlazom Y . Izlaz Y direktno je povezan sa ulazima X_1 i X_2 , *Bias*-om te skrivenim čvorom 4.



Slika 28. Grafovi fitnessa i razvoja topologije mreže XOR problema za 10. pokušaj

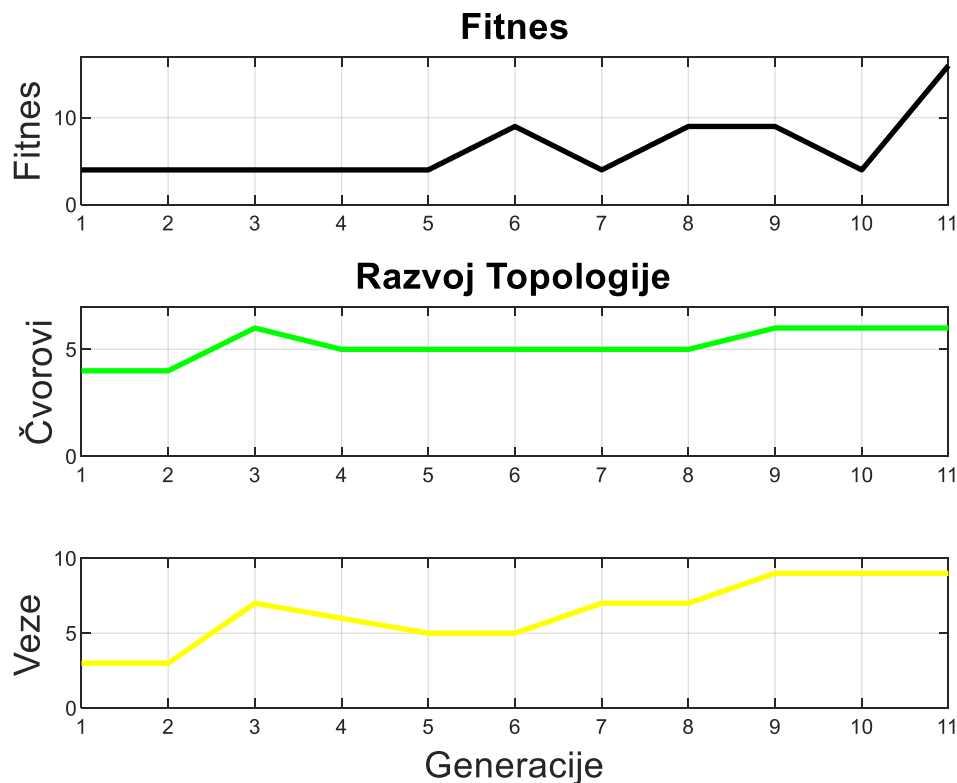
Na slici 28. vidi se graf fitnessa i razvoj topologije mreže u prvom pokušaju. Konačni fitness je 16, konačni broj čvorova je 5, a konačni broj omogućenih veza je 6. Broj čvorova se povećava i smanjuje kako generacije prolaze, odnosno kako se izvršava križanje te mutacija čvorova. Broj omogućenih veza se povećava i smanjuje kako generacije prolaze, odnosno kako se izvršava križanje te mutacija veza. Dodavanje čvorova i veza obično u početku smanjuje fitness mreže, a oduzimanje čvorova i veza obično u početku povećava fitness mreže.

Grafički prikaz neuronske mreže



Slika 29. Mreža rješenja XOR problema za 11. pokušaj

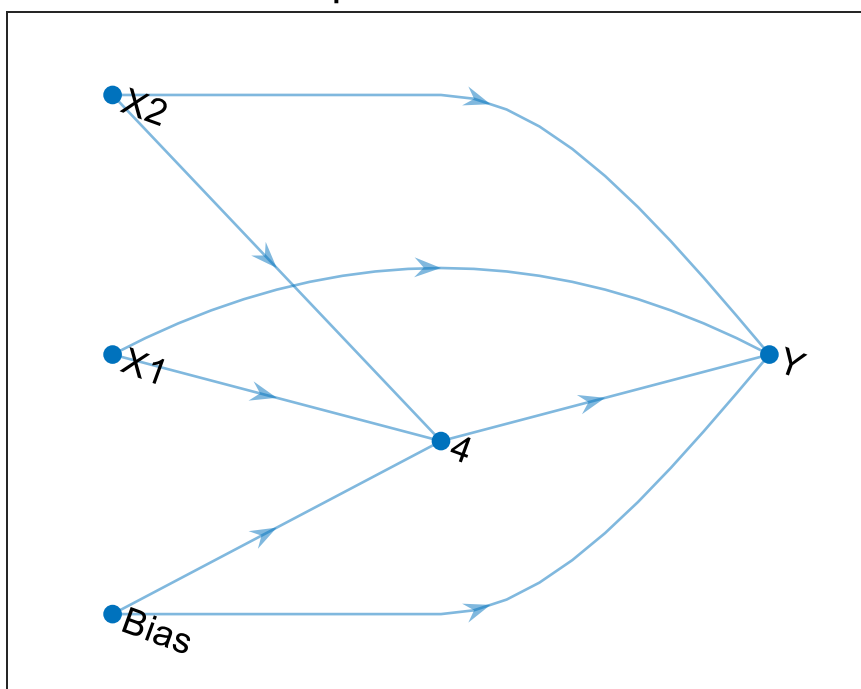
Na slici 29. može se vidjeti mreža koja je u 11. pokušaju riješila XOR problem. Mreža nije optimalna, ali je na jako dobrom putu. Mreža je nastala nakon 11 generacija i 1800 procijenjenih mreža. Mreža ima ukupno 6 čvorova, od kojih su dva ulazna, *bias*, jedan izlazni i dva skrivena čvora te 9 omogućenih veza. Prvi ulaz X_1 direktno je vezan sa izlazom Y te je također vezan sa skrivenim čvorovima 4 i 5. Drugi ulaz X_2 direktno je vezan sa izlazom Y te je također vezan sa skrivenim čvorovima 4 i 5. *Bias* je direktno vezan sa izlazom Y . Skriveni čvor 4 povezan je na svom ulazu sa ulazima X_1 i X_2 , a na izlazu je povezan sa skrivenim čvorom 5. Skriveni čvor 5 povezan je na svom ulazu sa ulazima X_1 i X_2 te skrivenim čvorom 4, a na izlazu je povezan sa izlazom Y . Izlaz Y direktno je povezan sa ulazima X_1 i X_2 , *Bias*-om te skrivenim čvorom 5. Mreža je također vizualno vrlo privlačna.



Slika 30. Grafovi fitnessa i razvoja topologije mreže XOR problema za 11. pokušaj

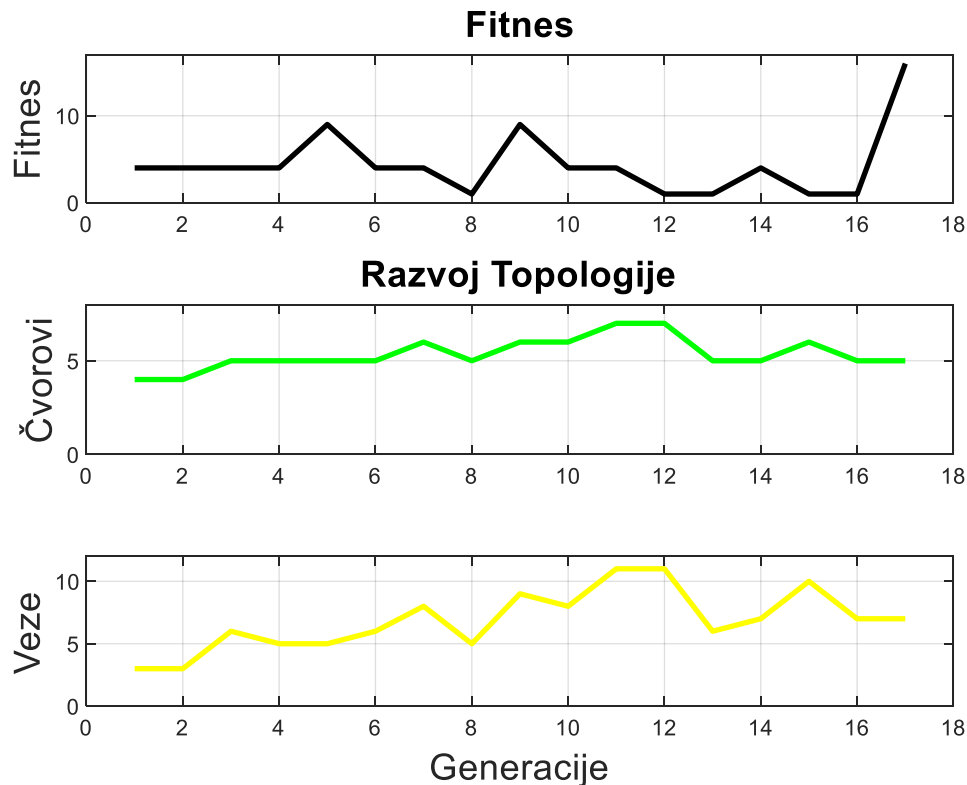
Na slici 30. vidi se graf fitnessa i razvoj topologije mreže u prvom pokušaju. Konačni fitness je 16, konačni broj čvorova je 6, a konačni broj omogućenih veza je 9. Broj čvorova se povećava i smanjuje kako generacije prolaze, odnosno kako se izvršava križanje te mutacija čvorova. Broj omogućenih veza se povećava i smanjuje kako generacije prolaze, odnosno kako se izvršava križanje te mutacija veza. Dodavanje čvorova i veza obično u početku smanjuje fitness mreže, a oduzimanje čvorova i veza obično u početku povećava fitness mreže.

Grafički prikaz neuronske mreže



Slika 31. Mreža rješenja XOR problema za 12. pokušaj

Na slici 31. može se vidjeti mreža koja je u 12. pokušaju riješila XOR problem. Mreža je optimalna. Mreža je nastala nakon 17 generacija i 2700 procijenjenih mreža. Mreža ima ukupno 5 čvorova, od kojih su dva ulazna, *bias*, jedan izlazni i jedan skriveni čvor te 7 omogućenih veza. Prvi ulaz X_1 direktno je vezan sa izlazom Y te je također vezan sa skrivenim čvorom 4. Drugi ulaz X_2 direktno je vezan sa izlazom Y te je također vezan sa skrivenim čvorom 4. *Bias* je direktno vezan sa izlazom Y te je također vezan sa skrivenim čvorom 4. Skriveni čvor 4 povezan je na svom ulazu sa ulazima X_1 i X_2 te *Bias*-om, a na izlazu je povezan sa izlazom Y . Izlaz Y direktno je povezan sa ulazima X_1 i X_2 , *Bias*-om te skrivenim čvorom 4.



Slika 32. Grafovi fitnesa i razvoja topologije mreže XOR problema za 12. pokušaj

Na slici 32. vidi se graf fitnesa i razvoj topologije mreže u prvom pokušaju. Konačni fitnes je 16, konačni broj čvorova je 5, a konačni broj omogućenih veza je 7. Broj čvorova se povećava i smanjuje kako generacije prolaze, odnosno kako se izvršava križanje te mutacija čvorova. Broj omogućenih veza se povećava i smanjuje kako generacije prolaze, odnosno kako se izvršava križanje te mutacija veza. Dodavanje čvorova i veza obično u početku smanjuje fitnes mreže, a oduzimanje čvorova i veza obično u početku povećava fitnes mreže.

Prema viđenim podacima i slikama, može se zaključiti da će NEAT algoritam generirati četiri vrste rješenja: optimalno rješenje (10. i 12. pokušaj), približno optimalno rješenje (prvi, treći, 5., 6., 8., 9., i 11. pokušaj), rješenje koje nije optimalno ali je dovoljno jednostavno i učinkovito (četvrti i 7. pokušaj) te rješenje koje nije niti približno optimalno (drugi pokušaj). Kao što se može zaključiti, najčešće će generirati približno optimalno rješenje.

Naravno da će sva navedena rješenja riješiti problem, ali potrebno je uzeti u obzir da kompleksnija rješenja zauzimaju puno više memorije i procesorske snage od optimalnih ili približno optimalnih. Također, bilo kakve greške u sustavu je puno jednostavnije naći u optimalnim i jednostavnijim mrežama, dok to u kompleksnijim mrežama može bit prilično izazovno.

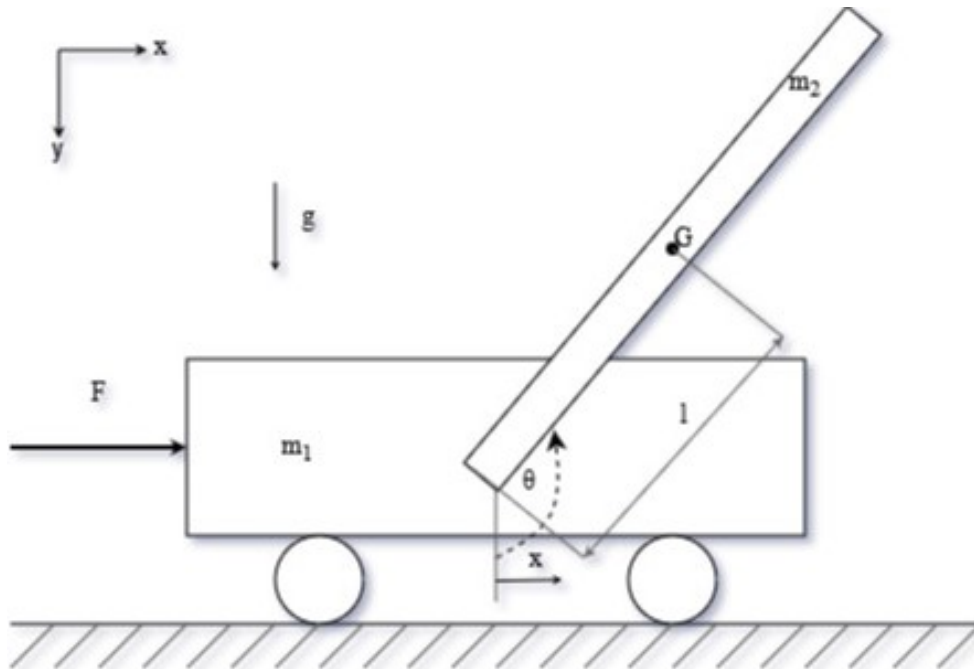
6.2 OBRNUTO NJIHALO

Postoji mnogo zadataka iz teorije upravljanja za učenje gdje tehnike korištene u NEAT-u mogu napraviti razliku. Mnoge od ovih potencijalnih primjena, poput robotske navigacije ili igranje računalnih igara, predstavljaju probleme bez poznatih rješenja. Domena balansiranja polova koristi se za dokaz jer je poznato mjerilo u literaturi, zbog čega je moguće pokazati učinkovitost NEAT-a. Također je dobra zamjena za stvarne probleme, djelomično zato što je balansiranje greda zapravo stvaran zadatak, a i zato što se težina može prilagoditi.

Obrnuto njihalo jedan je od najpopularnijih laboratorijskih pokusa koji se koristi za ilustraciju tehnika nelinearnog upravljanja. Ovaj sustav je inspiriran primjenama kao što su upravljanje raketama i antiseizmička kontrola zgrada.

Obrnuto njihalo na kolicima sastoji se od grede čija je točka zakretanja postavljena na kolica, koja su pokretna platforma te sustav ima dva stupnja slobode. Njihalo se slobodno njiše oko svoje točke zakretanja i nije izravno upravljano. Kolica se mogu kretati vodoravno, okomito na os rotacije njihala, i upravljana su silom koja se na njih primjenjuje u istom smjeru. Cilj upravljanja je dovesti njihalo iz donjeg stabilnog ravnotežnog položaja do gornjeg nestabilnog ravnotežnog položaja pomicanjem kolica po horizontalnoj ravnini. Podupravljeni sustav je sustav koji ima manje izravno upravljivih ulaza nego stupnjeva slobode. Budući da se kutnim ubrzanjem njihala ne može izravno upravljati, obrnuto njihalo je podupravljeni mehanički sustav. Stoga se tehnike razvijene za potpuno upravljive mehaničke sustave ne mogu koristiti za upravljanje obrnutim njihalom.

Obično se za upravljanje obrnutim njihalom na kolicima koriste neke od metoda iz klasične teorije upravljanja kao npr. LQR regulator. U ovom diplomskom radu obrnuto njihalo biti će upravljano pomoću NEAT algoritma. Kako bi to bilo moguće potrebno je izvesti dinamičke jednadžbe stanja sustava.



Slika 33. Obrnuto njihalo na kolicima

Na slici 33. prikazano je obrnuto njihalo na kolicima te su prikazani svi parametri sustava.

Parametri su:

m_1 – masa kolica,

m_2 – masa njihala,

J_2 – inercija njihala,

l – duljina kraka,

g – ubrzanje sile teže,

x – pomak kolica,

θ – kut zakreta njihala,

F – sila (parametar upravljanja)

Prvo je potrebno izvesti matematički model obrnutog njihala na kolicima. Za izvod modela biti će korištene Euler-Lagrangeove jednadžbe. Euler-Lagrange metoda ovisi o bilanci energije. Objekti koji se kreću posjeduju određenu kinetičku energiju, a također ovisno o razlici u položaju unutar gravitacijskog polja posjeduju određenu potencijalnu energiju. Vrijede standardne pretpostavke, tj. nema trenja, nema disipativnih sila itd.

Kinetička energija kolica K_1 iznosi:

$$K_1 = \frac{1}{2} m_1 \dot{x}^2 \quad (3).$$

Potencijalna energija kolica P_1 iznosi nula jer gravitacijske sile ne djeluju na kolica.

Kinetička energija njihala K_2 iznosi:

$$K_2 = \frac{1}{2} m_1 \left[\frac{d}{dt} (x + l \sin \theta) \right]^2 + \frac{1}{2} m_1 \left[\frac{d}{dt} (l \cos \theta) \right]^2 + \frac{1}{2} J_2 \dot{\theta}^2 \quad (4).$$

Potencijalna energija njihala P_2 iznosi:

$$P_2 = m_2 g l (1 - \cos \theta) \quad (5).$$

Ukupna kinetička energija K iznosi:

$$K = K_1 + K_2 \quad (6),$$

$$K = \frac{1}{2} (m_1 + m_2) \dot{x}^2 + m l \dot{x} \dot{\theta} \cos \theta + \frac{1}{2} (J_2 + m l^2) \dot{\theta}^2 \quad (7).$$

Ukupna potencijalna energija P iznosi:

$$P = P_1 + P_2 \quad (8),$$

$$P = m_2 g l (1 - \cos \theta) \quad (9).$$

Lagrangeova jednadžba L iznosi:

$$L = K - P \quad (10),$$

$$L = \frac{1}{2} (m_1 + m_2) \dot{x}^2 + m l \dot{x} \dot{\theta} \cos \theta + \frac{1}{2} (J_2 + m l^2) \dot{\theta}^2 - m_2 g l (1 - \cos \theta) \quad (11).$$

Koristi se slijedeći oblik Euler-Lagrange jednadžbe:

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{q}} \right) - \frac{\partial L}{\partial q} = \tau \quad (12),$$

Gdje su:

$\mathbf{q} = [q_1 \ q_2 \ \dots \ q_n]^T$ – generalizirani vektor varijabli, jedna za svaki stupanj slobode, za obrnuto njihalo na kolicima vrijedi $\mathbf{q} = [x \ \theta]^T$

$\boldsymbol{\tau} = [\tau_1 \ \tau_2 \ \dots \ \tau_n]^T$ – generalizirani vektor vanjskih sila koje djeluju na sustav, za obrnuto njihalo na kolicima vrijedi $\boldsymbol{\tau} = [F \ 0]^T$

Slijedi:

$$\left(\frac{\partial L}{\partial \dot{x}}\right) = (m_1 + m_2)\dot{x} + m_2 l \dot{\theta} \cos \theta \quad (13),$$

$$\left(\frac{\partial L}{\partial \dot{\theta}}\right) = 0 \quad (14),$$

$$\left(\frac{\partial L}{\partial \theta}\right) = m_2 l \dot{x} \cos \theta + (J_2 + m_2 l^2) \dot{\theta} \quad (15),$$

$$\left(\frac{\partial L}{\partial \theta}\right) = m_2 l \dot{x} \dot{\theta} \sin \theta - m_2 g l \sin \theta \quad (16).$$

Kombiniranjem jednadžbi (11) do (16) dobije se sustav dviju diferencijalnih jednadžbi sustava:

$$(m_1 + m_2)\ddot{x} + m_2 l (\cos \theta) \ddot{\theta} - m_2 l (\sin \theta) \dot{\theta}^2 = F \quad (17),$$

$$m_2 l^2 \ddot{\theta} = m_2 l \ddot{x} \cos \theta - m_2 g l \sin \theta \quad (18).$$

Izlučivanjem \ddot{x} iz (17), uvrštavanjem u (18) i sređivanjem dobiju se jednadžbe potrebne za uspješno implementiranje u NEAT algoritam:

$$\ddot{x} = \frac{1}{m_1 + m_2 \sin^2 \theta} [F + m_2 \sin \theta (l \dot{\theta}^2 + g \cos \theta)] \quad (19),$$

$$\ddot{\theta} = \frac{1}{l(m_1 + m_2 \sin^2 \theta)} [-F \cos \theta - m_2 l \dot{\theta}^2 \sin \theta \cos \theta - (m_1 + m_2) g \sin \theta] \quad (20).$$

Dobivanjem dinamičkih jednadžbi stanja sustava, moguće je implementirati sustav obrnutog njihala na kolicima u NEAT algoritam u MATLAB-u.

Parametri za simulaciju njihala proizvoljno su odabrani i iznose kako slijedi: masa kolica $m_1=1$ kg, masa njihala $m_2=0,25$ kg, duljina kraka $l=0,25$ m, duljina staze je 4,8 m, upravljačka sila F ograničena je u rasponu od -10 N do 10 N, a početna vrijednost sile F je nula. Početni položaj obrnutog njihala na kolicima je donji stabilni ravnotežni položaj, što znači da njihalo slobodno visi tj. kut zakreta θ je jednak nuli kao i kutna brzina $\dot{\theta}$ i kutno ubrzanje $\ddot{\theta}$. Također to znači da su pomak kolica x , brzina kolica \dot{x} i ubrzanje kolica \ddot{x} također jednaki nuli. Vrijeme simulacije je 5 sekundi koje je podjeljeno na vremenske korake od 0,025 s, a simulacijski koraci iznose 200, odnosno rezultat su dijeljenja vremena simulacije i vremenskih koraka.

Simulacija je uspješna ako neka od vrsta NEAT algoritma uspije zanjhati njihalo i stabilizirati ga 100 simulacijskih koraka. Zbog inherentne nestabilnosti obrnutog njihala postoji vjerojatnost da svaka simulacija neće biti uspješna, za razliku od XOR problema. Eksperimentalno je ustanovljeno da ako je NEAT algoritmu potrebno više od 350 generacija za uspješno stabiliziranje obrnutog njihala, simulacija će sigurno biti neuspješna.

Zbog gore navedenih razloga nije moguće, kao kod XOR problema, predvidjeti potreban prosječan broj generacija, broj procijenjenih mreža, broj skrivenih čvorova i broj omogućenih veza, niti njihove minimalne i maksimalne iznose, kao niti optimalnu strukturu mreže.

Broj potrebnih uspješnih simulacija je 10, a biti će dan i postotak uspješnosti stabilizacije kao omjer broja uspješnih simulacija i ukupnog broja simulacija pomnožen sa 100.

Rezultati 10 pokušaja pronalaženja strukture za problem zanjihavanja i stabilizacije obrnutog njihala na kolicima prikazani su u tablici 3.

Tablica 3. Rezultati eksperimenta pronalaženja strukture za problem obrnutog njihala

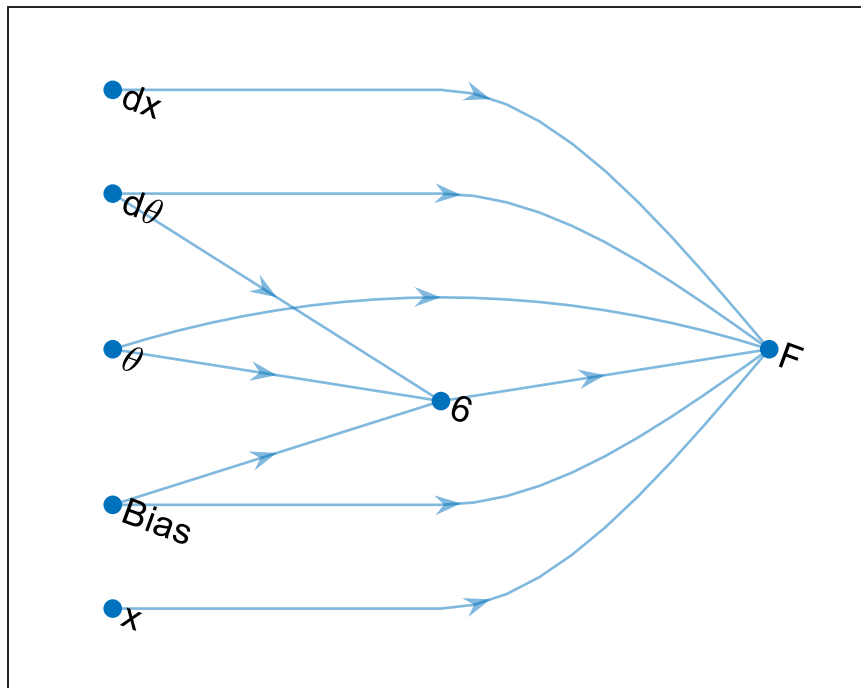
Broj uspješnog pokušaja	Broj generacija potrebnih za pronalazak rješenja	Broj procijenjenih mreža potrebnih za pronalazak rješenja	Broj skrivenih čvorova	Broj omogućenih veza
1	36	5550	1	9
2	108	16350	9	38
3	157	23700	10	45
4	25	3900	6	23
5	18	2850	4	16
6	111	16800	11	41
7	112	16950	6	29
8	153	23100	14	52
9	25	3900	2	10
10	67	10200	3	17

Ukupan broj simulacija je 25, a broj uspješnih simulacija je 10 pa je postotak uspješnosti 40%, što je dosta nisko, ali s obzirom da je stabilizacija podupravljenih sustava vrlo kompleksan problem to je očekivano.

Kao što se iz Tablice 3. može vidjeti, broj minimalnih potrebnih generacija za pronalaženje rješenja je 18. Maksimalni broj generacija za pronalaženje rješenja je 157. Prosječan broj generacija za pronalaženje rješenja je 81,2. Broj minimalnih procijenjenih mreža potrebnih za pronalaženje rješenja je 2850. Broj maksimalnih procijenjenih mreža potrebnih za pronalaženje rješenja je 23700. Prosječan broj procijenjenih mreža za pronalaženje rješenja je 12330. Broj potrebnih skrivenih čvorova je u prosjeku 6,6. U prosjeku je 28 potrebnih omogućenih veza.

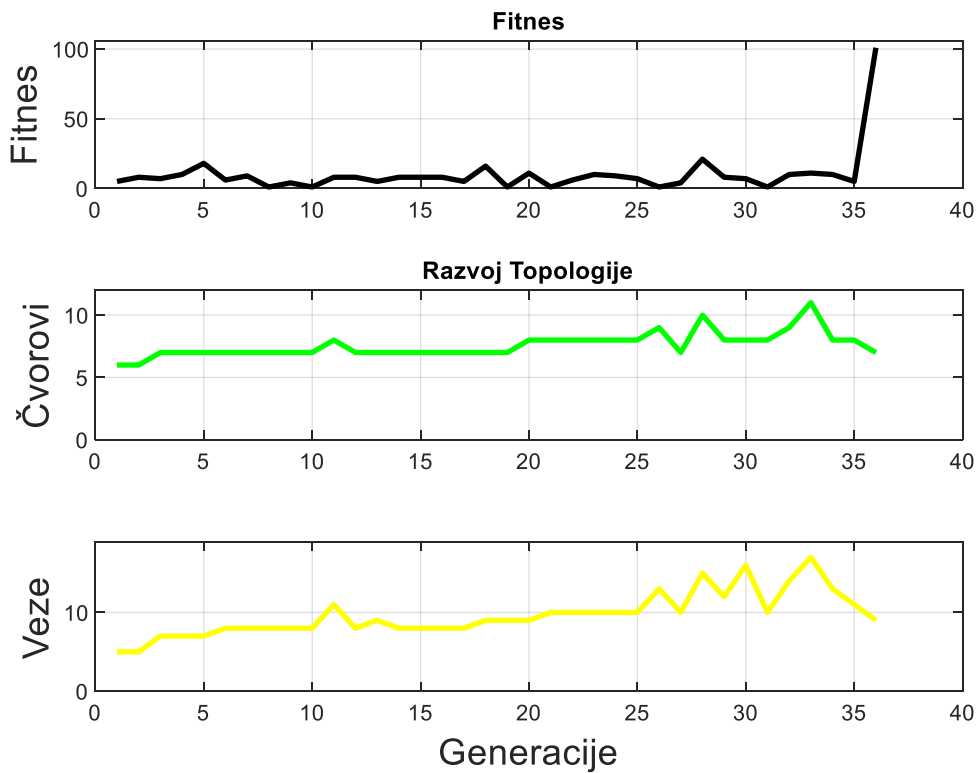
Sada će biti prikazani grafički prikazi mreža rješenja, grafovi razvoja topologije, odnosno grafovi broja skrivenih čvorova te broja omogućenih veza, grafovi fitnesa tijekom generacija potrebnih za rješavanje problema te također grafovi varijabli stanja sustava nastalih nakon 10 uspješnih pokušaja, na slikama 34. do 63.

Grafički prikaz neuronske mreže



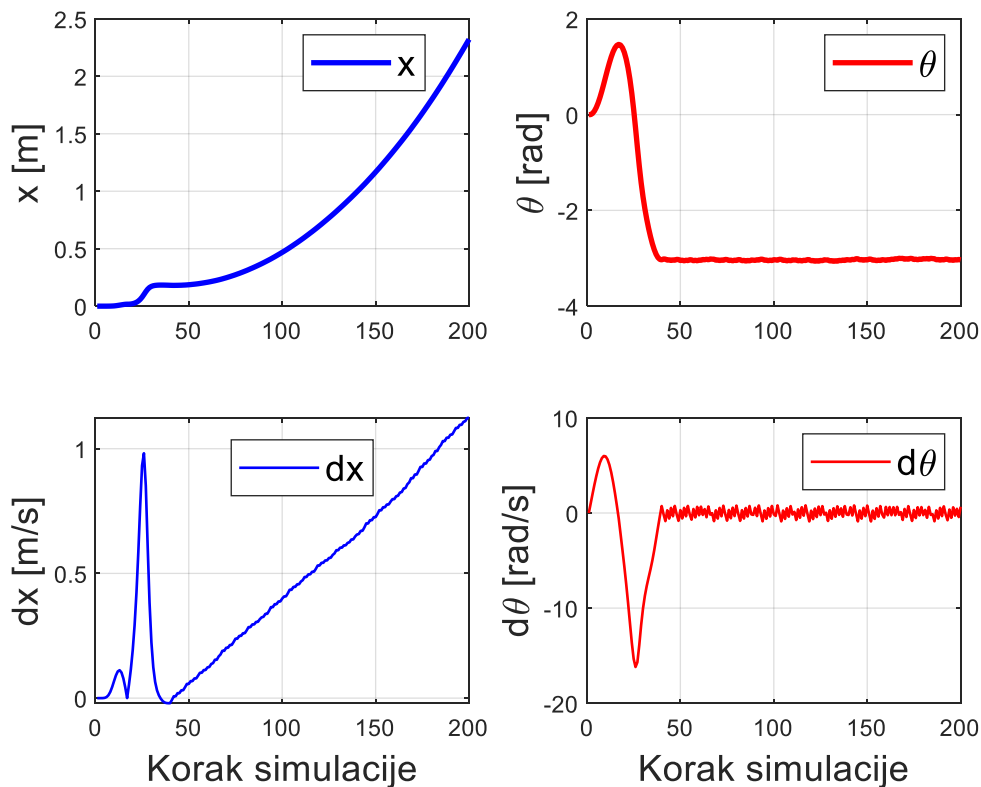
Slika 34. Mreža rješenja problema stabilizacije obrnutog njihala za prvi pokušaj

Na slici 34. može se vidjeti mreža koja je u prvom pokušaju riješila problem stabilizacije obrnutog njihala. Mreža je nastala nakon 36 generacija i 5550 procijenjenih mreža. Mreža ima ukupno 7 čvorova, od kojih su četiri ulazna, *bias*, jedan izlazni i 1 skriveni čvor te 9 omogućenih veza. Prvi ulaz x direktno je vezan sa izlazom F . Drugi ulaz θ direktno je vezan sa izlazom F te je također vezan sa skrivenim čvorom 6. Treći ulaz dx direktno je vezan sa izlazom F . Četvrti ulaz $d\theta$ direktno je vezan sa izlazom F te je također vezan sa skrivenim čvorom 6. *Bias* je direktno vezan sa izlazom F te je također vezan sa skrivenim čvorom 6. Skriveni čvor 6 povezan je na svom ulazu sa ulazima θ i $d\theta$ te *Bias*-om, a na izlazu je povezan sa izlazom F . Izlaz F direktno je povezan sa ulazima x , θ , dx i $d\theta$, *Bias*-om te skrivenim čvorom 6.



Slika 35. Grafovi fitnessa i razvoja topologije mreže problema stabilizacije obrnutog njihala za prvi pokušaj

Na slici 35. vidi se graf fitnessa i razvoj topologije mreže u prvom pokušaju. Konačni fitness je 101, konačni broj čvorova je 7, a konačni broj omogućenih veza je 9. Broj čvorova se povećava i smanjuje kako generacije prolaze, odnosno kako se izvršava križanje te mutacija čvorova. Broj omogućenih veza se povećava i smanjuje kako generacije prolaze, odnosno kako se izvršava križanje te mutacija veza. Dodavanje čvorova i veza obično u početku smanjuje fitness mreže, a oduzimanje čvorova i veza obično u početku povećava fitness mreže.



Slika 36. Grafovi varijabli stanja sustava obrnutog njihala za prvi pokušaj

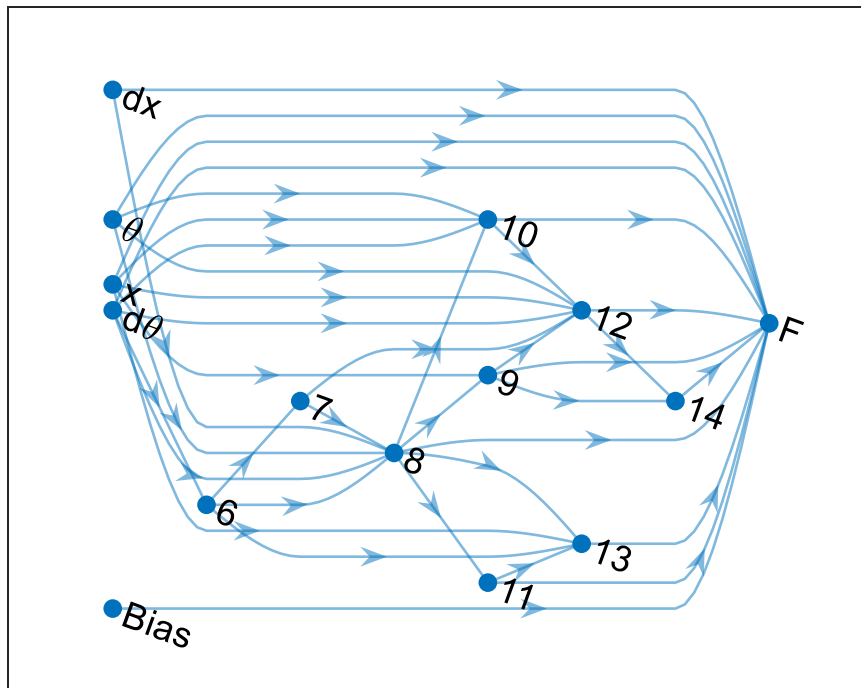
Na slici 36. prikazani su grafovi varijabli stanja sustava obrnutog njihala za prvi pokušaj.

U početnom trenutku obrnuto njihalo nalazi se u donjem stabilnom ravnotežnom položaju, što znači da njihalo slobodno visi tj. kut zakreta θ je jednak nuli kao i kutna brzina $\dot{\theta}$ i kutno ubrzanje $\ddot{\theta}$. Također to znači da su pomak kolica x , brzina kolica \dot{x} i ubrzanje kolica \ddot{x} također jednaki nuli.

U ovom pokušaju njihalo je prvo zanjihano iz donjeg stabilnog ravnotežnog položaja $\theta = 0$ do pozitivnog (desnog) horizontalnog položaja $\theta = \frac{\pi}{2}$, pada natrag u donji stabilni ravnotežni položaj, a zatim je zanjihano do negativnog (lijevog) gornjeg nestabilnog položaja $\theta = \pi$ gdje je stabilizirano. Kolica se isprva ne gibaju, pokreću se tek kada se njihalo vrati u donji stabilni ravnotežni položaj, te se nastavljaju kontinuirano gibati po stazi kako bi se njihalo dovelo do gornjeg nestabilnog ravnotežnog položaja, a zatim kako bi se stabiliziralo. Kada bi kolica došla do kraja staze njihalo bi palo natrag u donji stabilni ravnotežni položaj.

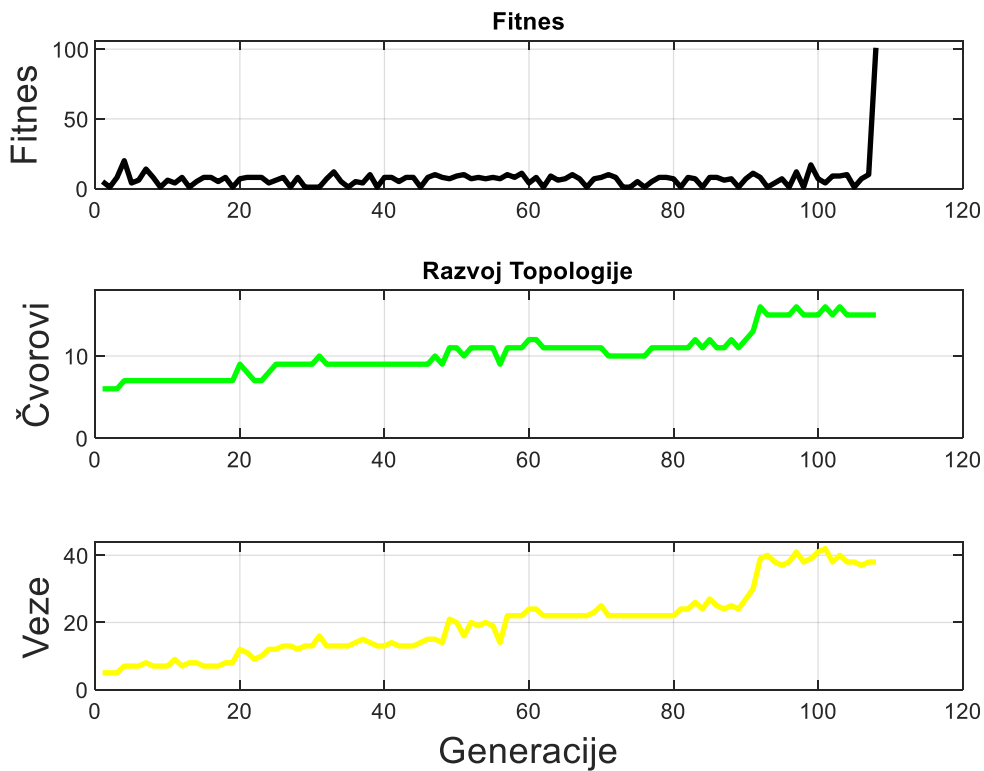
Zanimljivo je da je isti način stabilizacije primjenjen i u pokušaju 7, što pokazuje da različite mreže mogu uspješno doći do iste metode.

Grafički prikaz neuronske mreže



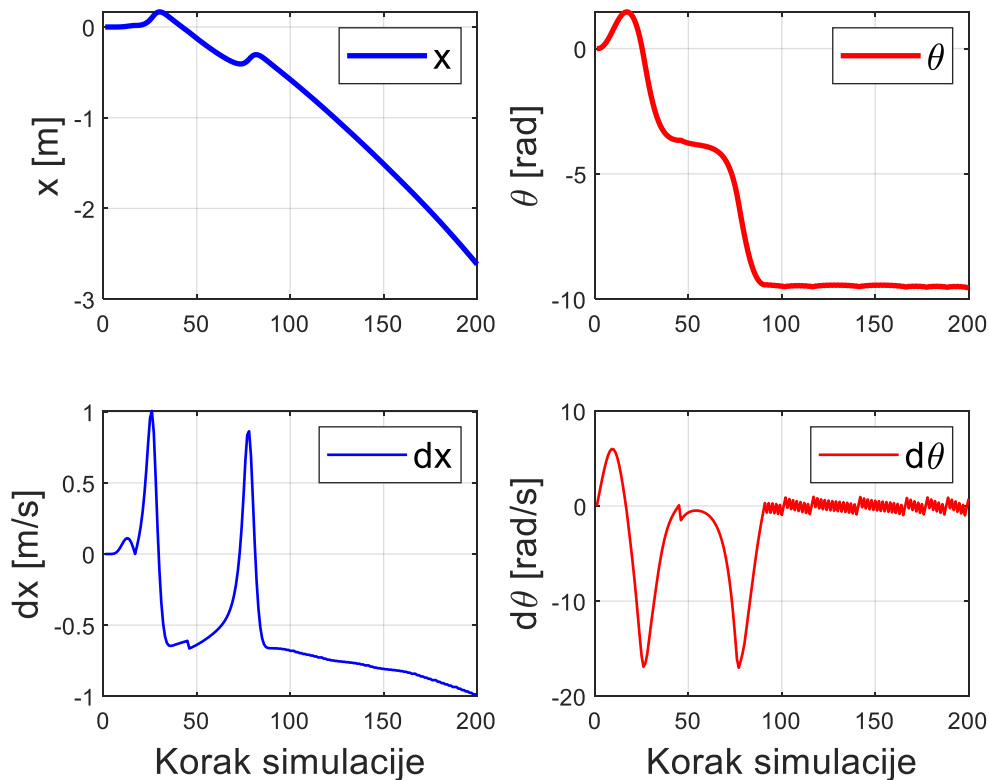
Slika 37. Mreža rješenja problema stabilizacije obrnutog njihala za drugi pokušaj

Na slici 37. može se vidjeti mreža koja je u prvom pokušaju riješila problem stabilizacije obrnutog njihala. Mreža je nastala nakon 108 generacija i 16350 procijenjenih mreža. Mreža ima ukupno 15 čvorova, od kojih su četiri ulazna, *bias*, jedan izlazni i 9 skrivenih čvorova te 38 omogućenih veza. Prvi ulaz x direktno je vezan sa izlazom F , te skrivenim čvorovima 9, 10, 12 i 13. Drugi ulaz θ direktno je vezan sa izlazom F te je također vezan sa skrivenim čvorovima 8, 10 i 12. Treći ulaz dx direktno je vezan sa izlazom F te je također vezan sa skrivenim čvorom 8. Četvrti ulaz $d\theta$ direktno je vezan sa izlazom F te je također vezan sa skrivenim čvorovima 6, 8, 10 i 12. *Bias* je direktno vezan sa izlazom F . Skriveni čvor 6 povezan je na svom ulazu sa ulazom $d\theta$, a na izlazu je povezan sa skrivenim čvorovima 7, 8 i 13. Skriveni čvor 7 povezan je na svom ulazu sa skrivenim čvorom 6, a na izlazu je povezan sa skrivenim čvorovima 8 i 12. Skriveni čvor 8 povezan je na svom ulazu sa ulazima θ , dx i $d\theta$ te je također vezan sa skrivenim čvorom 6, a na izlazu je povezan sa izlazom F te skrivenim čvorovima 9, 10, 11 i 13. Skriveni čvor 10 povezan je na svom ulazu sa ulazima x , θ , i $d\theta$ te je također vezan sa skrivenim čvorom 8, a na izlazu je povezan sa izlazom F te skrivenim čvorom 12. Skriveni čvor 9 povezan je na svom ulazu sa ulazom x te je također vezan sa skrivenim čvorom 8, a na izlazu je povezan sa izlazom F te skrivenim čvorovima 12 i 14. Skriveni čvor 11 povezan je na svom ulazu sa skrivenim čvorom 8, a na izlazu je povezan sa izlazom F te skrivenim čvorom 13. Skriveni čvor 12 povezan je na svom ulazu sa ulazima x , θ , i $d\theta$ te je također vezan sa skrivenim čvorovima 9 i 10, a na izlazu je povezan sa izlazom F te skrivenim čvorom 14. Skriveni čvor 13 povezan je na svom ulazu sa ulazom x te je također vezan sa skrivenim čvorovima 6, 8 i 11, a na izlazu je povezan sa izlazom F . Skriveni čvor 14 povezan je na svom ulazu sa skrivenim čvorovima 9 i 12, a na izlazu je povezan sa izlazom F . Izlaz F direktno je povezan sa ulazima x , θ , dx i $d\theta$, *Bias*-om te skrivenim čvorovima 8, 9, 10, 11, 12, 13 i 14.



Slika 38. Grafovi fitnessa i razvoja topologije mreže problema stabilizacije obrnutog njihala za drugi pokušaj

Na slici 38. vidi se graf fitnessa i razvoj topologije mreže u prvom pokušaju. Konačni fitness je 101, konačni broj čvorova je 15, a konačni broj omogućenih veza je 38. Broj čvorova se povećava i smanjuje kako generacije prolaze, odnosno kako se izvršava križanje te mutacija čvorova. Broj omogućenih veza se povećava i smanjuje kako generacije prolaze, odnosno kako se izvršava križanje te mutacija veza. Dodavanje čvorova i veza obično u početku smanjuje fitness mreže, a oduzimanje čvorova i veza obično u početku povećava fitness mreže.



Slika 39. Grafovi varijabli stanja sustava obrnutog njihala za drugi pokušaj

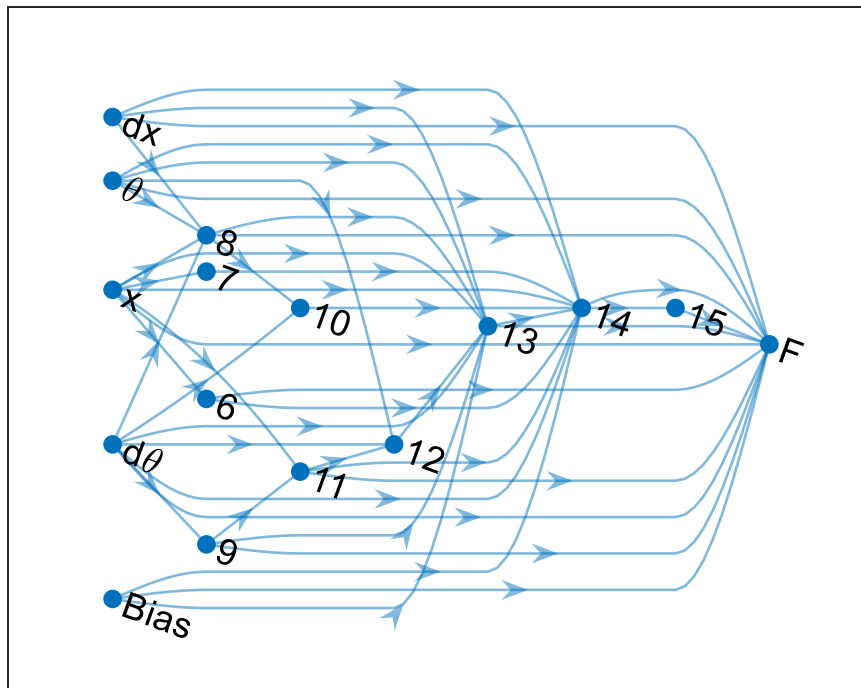
Na slici 39. prikazani su grafovi varijabli stanja sustava obrnutog njihala za drugi pokušaj.

U početnom trenutku obrnuto njihalo nalazi se u donjem stabilnom ravnotežnom položaju, što znači da njihalo slobodno visi tj. kut zakreta θ je jednak nuli kao i kutna brzina $\dot{\theta}$ i kutno ubrzanje $\ddot{\theta}$. Također to znači da su pomak kolica x , brzina kolica \dot{x} i ubrzanje kolica \ddot{x} također jednaki nuli.

U ovom pokušaju njihalo je prvo zanjihano iz donjeg stabilnog ravnotežnog položaja $\theta = 0$ do pozitivnog (desnog) horizontalnog položaja $\theta = \frac{\pi}{2}$, pada natrag u donji stabilni ravnotežni položaj, a zatim je zanjihano do negativnog (lijevog) gornjeg nestabilnog položaja $\theta = -\pi$ gdje se kratkotrajno stabilizira, ali nedovoljno dugo te je zatim zanjihano na vrlo pametan način da napravi puni krug u smjeru kazaljke na satu (-2π), vrati se u gornji nestabilni položaj $\theta = -3\pi$ gdje se ovaj put uspješno stabilizira. Kolica se isprva ne gibaju, pokreću se tek kada se njihalo vrati u donji stabilni ravnotežni položaj, a zatim se gibaju u suprotnom smjeru kako bi se pokušalo stabilizirati njihalo, neuspješno, pomiče se lagano unatrag, kako bi se osigurala rotacija njihala, te se nastavlja kontinuirano gibati, što rezultira uspješnom stabilizacijom njihala u gornjem nestabilnom ravnotežnom položaju. Kada bi kolica došla do kraja staze njihalo bi palo natrag u donji stabilni ravnotežni položaj.

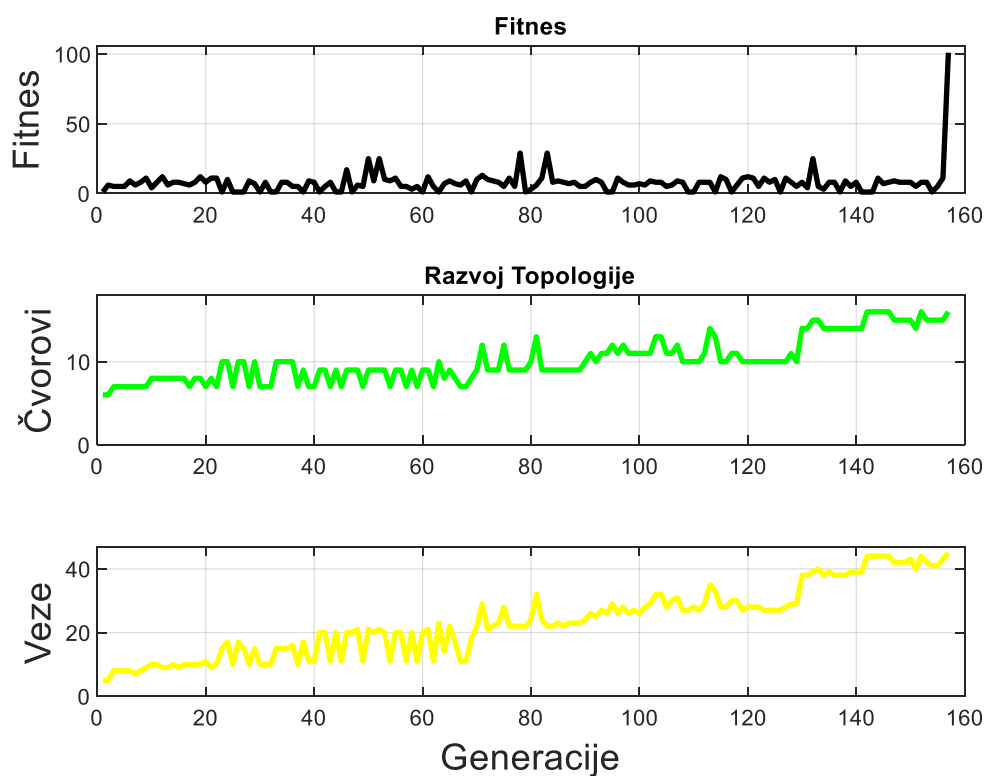
Zanimljivo je da je sličan način stabilizacije primjenjen i u pokušaju 10, što pokazuje da različite mreže mogu uspješno doći do iste metode.

Grafički prikaz neuronske mreže



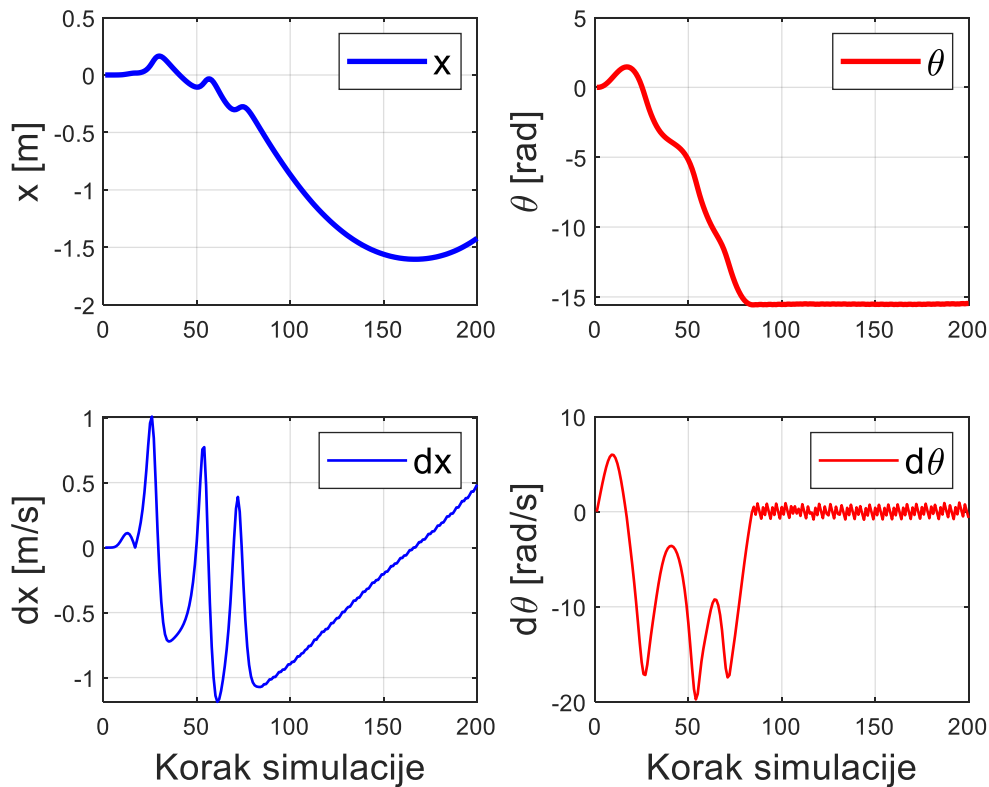
Slika 40. Mreža rješenja problema stabilizacije obrnutog njihala za treći pokušaj

Na slici 40. može se vidjeti mreža koja je u prvom pokušaju riješila problem stabilizacije obrnutog njihala. Mreža je nastala nakon 157 generacija i 23700 procijenjenih mreža. Mreža ima ukupno 16 čvorova, od kojih su četiri ulazna, *bias*, jedan izlazni i 10 skrivenih čvorova te 45 omogućenih veza. Prvi ulaz x direktno je vezan sa izlazom F , te skrivenim čvorovima 6, 7, 8, 11, 13 i 14. Drugi ulaz θ direktno je vezan sa izlazom F te je također vezan sa skrivenim čvorovima 8, 12, 13 i 14. Treći ulaz dx direktno je vezan sa izlazom F te je također vezan sa skrivenim čvorovima 8, 13 i 14. Četvrti ulaz $d\theta$ direktno je vezan sa izlazom F te je također vezan sa skrivenim čvorovima 8, 9, 10, 12, 13 i 14. *Bias* je direktno vezan sa izlazom F te je također vezan sa skrivenim čvorovima 13 i 14. Skriveni čvor 8 povezan je na svom ulazu sa ulazima x , θ , dx , $d\theta$, a na izlazu je povezan sa izlazom F te skrivenim čvorovima 10 i 13. Skriveni čvor 7 povezan je na svom ulazu sa ulazom x , a na izlazu je povezan sa skrivenim čvorom 14. Skriveni čvor 6 povezan je na svom ulazu sa ulazom x , a na izlazu je povezan sa izlazom F te skrivenim čvorom 14. Skriveni čvor 9 povezan je na svom ulazu sa ulazom $d\theta$, a na izlazu je povezan sa izlazom F te skrivenim čvorom 13. Skriveni čvor 10 povezan je na svom ulazu sa ulazom $d\theta$ te je također vezan sa skrivenim čvorom 8, a na izlazu je povezan sa skrivenim čvorom 14. Skriveni čvor 11 povezan je na svom ulazu sa ulazom x skrivenim čvorom 9, a na izlazu je povezan sa izlazom F te skrivenim čvorovima 12 i 14. Skriveni čvor 12 povezan je na svom ulazu sa ulazima θ i $d\theta$ te je također vezan sa skrivenim čvorom 11, a na izlazu je povezan sa skrivenim čvorom 13. Skriveni čvor 13 povezan je na svom ulazu sa ulazima x , θ , dx i $d\theta$, *Bias*-om te skrivenim čvorovima 8, 9 i 12, a na izlazu je povezan sa izlazom F te skrivenim čvorom 14. Skriveni čvor 14 povezan je na svom ulazu sa ulazima x , θ , dx i $d\theta$, *Bias*-om te skrivenim čvorovima 6, 7, 10, 11 i 13, a na izlazu je povezan sa izlazom F te skrivenim čvorom 15. Izlaz F direktno je povezan sa ulazima x , θ , dx i $d\theta$, *Bias*-om te skrivenim čvorovima 6, 8, 9, 11, 13, 14 i 15.



Slika 41. Grafovi fitnessa i razvoja topologije mreže problema stabilizacije obrnutog njihala za treći pokušaj

Na slici 41. vidi se graf fitnessa i razvoj topologije mreže u prvom pokušaju. Konačni fitness je 101, konačni broj čvorova je 16, a konačni broj omogućenih veza je 45. Broj čvorova se povećava i smanjuje kako generacije prolaze, odnosno kako se izvršava križanje te mutacija čvorova. Broj omogućenih veza se povećava i smanjuje kako generacije prolaze, odnosno kako se izvršava križanje te mutacija veza. Dodavanje čvorova i veza obično u početku smanjuje fitness mreže, a oduzimanje čvorova i veza obično u početku povećava fitness mreže.



Slika 42. Grafovi varijabli stanja sustava obrnutog njihala za treći pokušaj

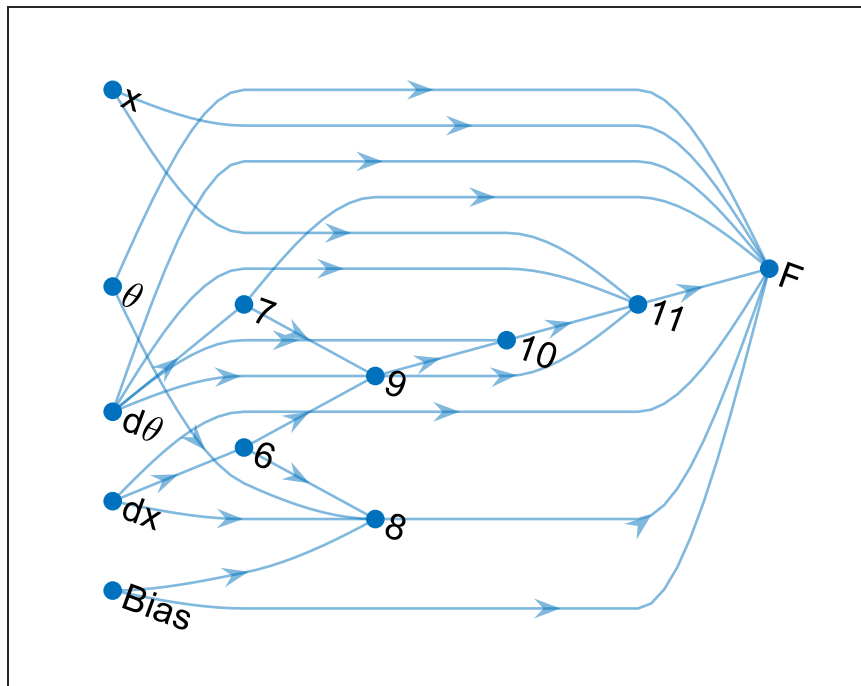
Na slici 42. prikazani su grafovi varijabli stanja sustava obrnutog njihala za treći pokušaj.

U početnom trenutku obrnuto njihalo nalazi se u donjem stabilnom ravnotežnom položaju, što znači da njihalo slobodno visi tj. kut zakreta θ je jednak nuli kao i kutna brzina $\dot{\theta}$ i kutno ubrzanje $\ddot{\theta}$. Također to znači da su pomak kolica x , brzina kolica \dot{x} i ubrzanje kolica \ddot{x} također jednaki nuli.

U ovom pokušaju njihalo je prvo zanjihano iz donjeg stabilnog ravnotežnog položaja $\theta = 0$ do pozitivnog (desnog) horizontalnog položaja $\theta = \frac{\pi}{2}$, pada natrag u donji stabilni ravnotežni položaj, a zatim je zanjihano do negativnog (lijevog) gornjeg nestabilnog ravnotežnog položaja $\theta = -\pi$ gdje se kratkotrajno stabilizira, ali nedovoljno dugo te je zatim zanjihano na vrlo pametan način da napravi dva puna kruga u smjeru kazaljke na satu (-4π), vrati se u gornji nestabilni položaj $\theta = -5\pi$ gdje se ovaj put uspješno stabilizira. Kolica se isprva ne gibaju, pokreću se tek kada se njihalo vrati u donji stabilni ravnotežni položaj, a zatim se gibaju u suprotnom smjeru kako bi se pokušalo stabilizirati njihalo, neuspješno, pomiče se lagano unatrag, kako bi se osigurala rotacija njihala, te se nastavlja kontinuirano gibati, ponavlja prethodni postupak, što rezultira uspješnom stabilizacijom njihala u gornjem nestabilnom ravnotežnom položaju. Kako bi se stabilizacija održala nastavlja se kontinuirano gibati, a zatim se počinje kretati unatrag.

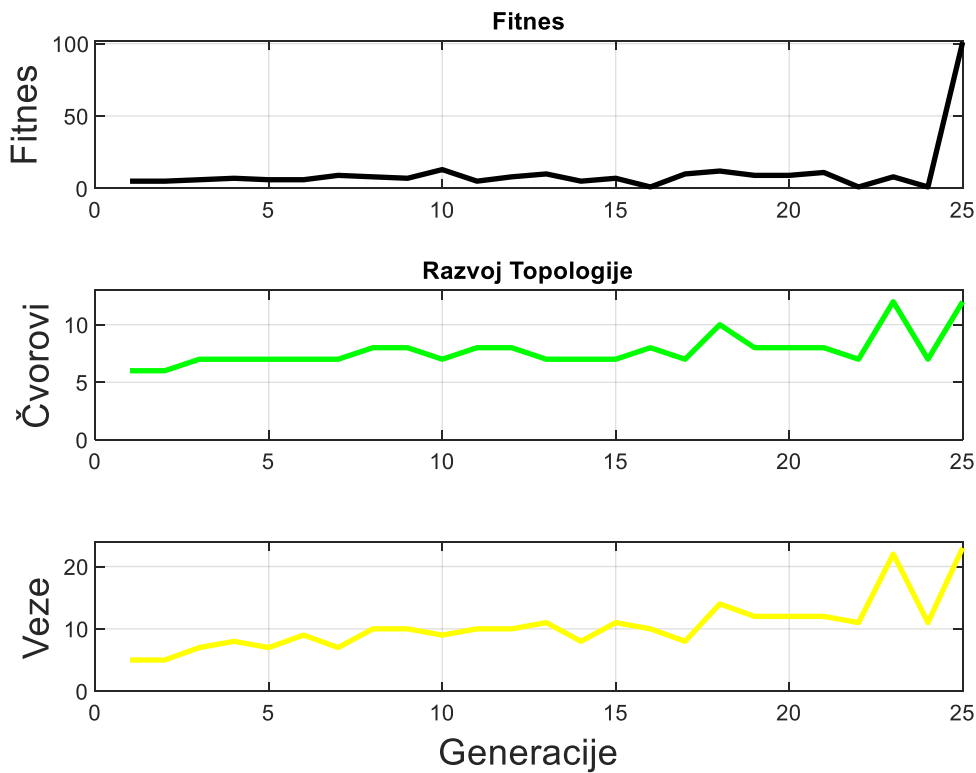
Zanimljivo je da je isti način stabilizacije primjenjen i u pokušaju četiri, što pokazuje da različite mreže mogu uspješno doći do iste metode.

Grafički prikaz neuronske mreže



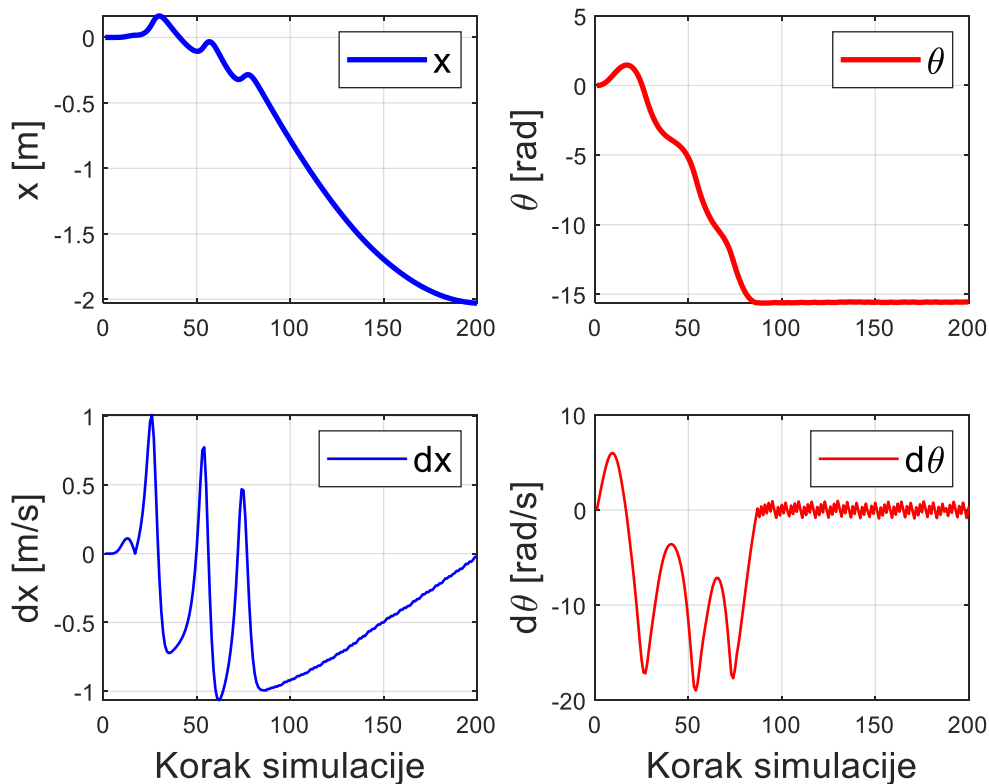
Slika 43. Mreža rješenja problema stabilizacije obrnutog njihala za četvrti pokušaj

Na slici 43. može se vidjeti mreža koja je u prvom pokušaju riješila problem stabilizacije obrnutog njihala. Mreža je nastala nakon 25 generacija i 3900 procijenjenih mreža. Mreža ima ukupno 12 čvorova, od kojih su četiri ulazna, *bias*, jedan izlazni i 6 skrivenih čvorova te 23 omogućenih veza. Prvi ulaz x direktno je vezan sa izlazom F , te skrivenim čvorom 11. Drugi ulaz θ direktno je vezan sa izlazom F te je također vezan sa skrivenim čvorom 8. Treći ulaz dx direktno je vezan sa izlazom F te je također vezan sa skrivenim čvorovima 6 i 8. Četvrti ulaz $d\theta$ direktno je vezan sa izlazom F te je također vezan sa skrivenim čvorovima 7, 9, 10 i 11. *Bias* je direktno vezan sa izlazom F te je također vezan sa skrivenim čvorom 8. Skriveni čvor 7 povezan je na svom ulazu sa ulazom $d\theta$, a na izlazu je povezan sa izlazom F te skrivenim čvorom 9. Skriveni čvor 6 povezan je na svom ulazu sa ulazom dx , a na izlazu je povezan sa skrivenim čvorovima 8 i 9. Skriveni čvor 9 povezan je na svom ulazu sa ulazom $d\theta$ te je također vezan sa skrivenim čvorovima 6 i 7, a na izlazu je povezan sa skrivenim čvorovima 10 i 11. Skriveni čvor 8 povezan je na svom ulazu sa ulazima θ i dx , *Bias*-om te je također vezan sa skrivenim čvorom 6, a na izlazu je povezan sa izlazom F . Skriveni čvor 10 povezan je na svom ulazu sa ulazom $d\theta$ te je također vezan sa skrivenim čvorom 9, a na izlazu je povezan sa skrivenim čvorom 11. Skriveni čvor 11 povezan je na svom ulazu sa ulazima x i $d\theta$ te skrivenim čvorovima 9 i 10, a na izlazu je povezan sa izlazom F . Izlaz F direktno je povezan sa ulazima x , θ , dx i $d\theta$, *Bias*-om te skrivenim čvorovima 7, 8 i 11.



Slika 44. Grafovi fitnessa i razvoja topologije mreže problema stabilizacije obrnutog njihala za četvrti pokušaj

Na slici 44. vidi se graf fitnessa i razvoj topologije mreže u prvom pokušaju. Konačni fitness je 101, konačni broj čvorova je 12, a konačni broj omogućenih veza je 23. Broj čvorova se povećava i smanjuje kako generacije prolaze, odnosno kako se izvršava križanje te mutacija čvorova. Broj omogućenih veza se povećava i smanjuje kako generacije prolaze, odnosno kako se izvršava križanje te mutacija veza. Dodavanje čvorova i veza obično u početku smanjuje fitness mreže, a oduzimanje čvorova i veza obično u početku povećava fitness mreže.



Slika 45. Grafovi varijabli stanja sustava obrnutog njihala za četvrti pokušaj

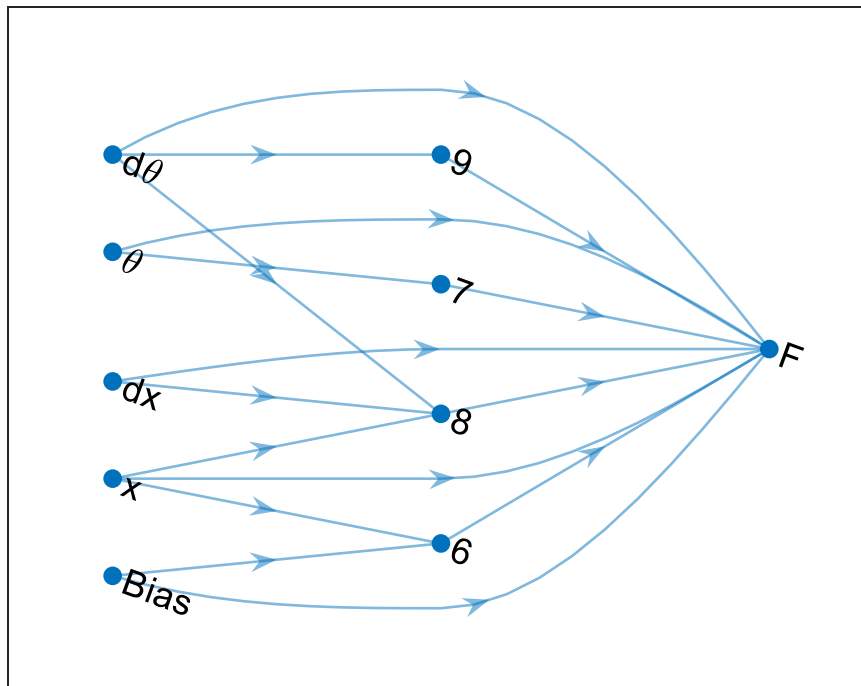
Na slici 45. prikazani su grafovi varijabli stanja sustava obrnutog njihala za četvrti pokušaj.

U početnom trenutku obrnuto njihalo nalazi se u donjem stabilnom ravnotežnom položaju, što znači da njihalo slobodno visi tj. kut zakreta θ je jednak nuli kao i kutna brzina $\dot{\theta}$ i kutno ubrzanje $\ddot{\theta}$. Također to znači da su pomak kolica x , brzina kolica \dot{x} i ubrzanje kolica \ddot{x} također jednaki nuli.

U ovom pokušaju njihalo je prvo zanjihano iz donjeg stabilnog ravnotežnog položaja $\theta = 0$ do pozitivnog (desnog) horizontalnog položaja $\theta = \frac{\pi}{2}$, pada natrag u donji stabilni ravnotežni položaj, a zatim je zanjihano do negativnog (lijevog) gornjeg nestabilnog ravnotežnog položaja $\theta = -\pi$ gdje se kratkotrajno stabilizira, ali nedovoljno dugo te je zatim zanjihano na vrlo pametan način da napravi dva puna kruga u smjeru kazaljke na satu (-4π), vrati se u gornji nestabilni položaj $\theta = -5\pi$ gdje se ovaj put uspješno stabilizira. Kolica se isprva ne gibaju, pokreću se tek kada se njihalo vrati u donji stabilni ravnotežni položaj, a zatim se gibaju u suprotnom smjeru kako bi se pokušalo stabilizirati njihalo, neuspješno, pomiče se lagano unatrag, kako bi se osigurala rotacija njihala, te se nastavlja kontinuirano gibati, ponavlja prethodni postupak, što rezultira uspješnom stabilizacijom njihala u gornjem nestabilnom ravnotežnom položaju. Kako bi se stabilizacija održala nastavlja se kontinuirano gibati, a zatim se počinje kretati unatrag.

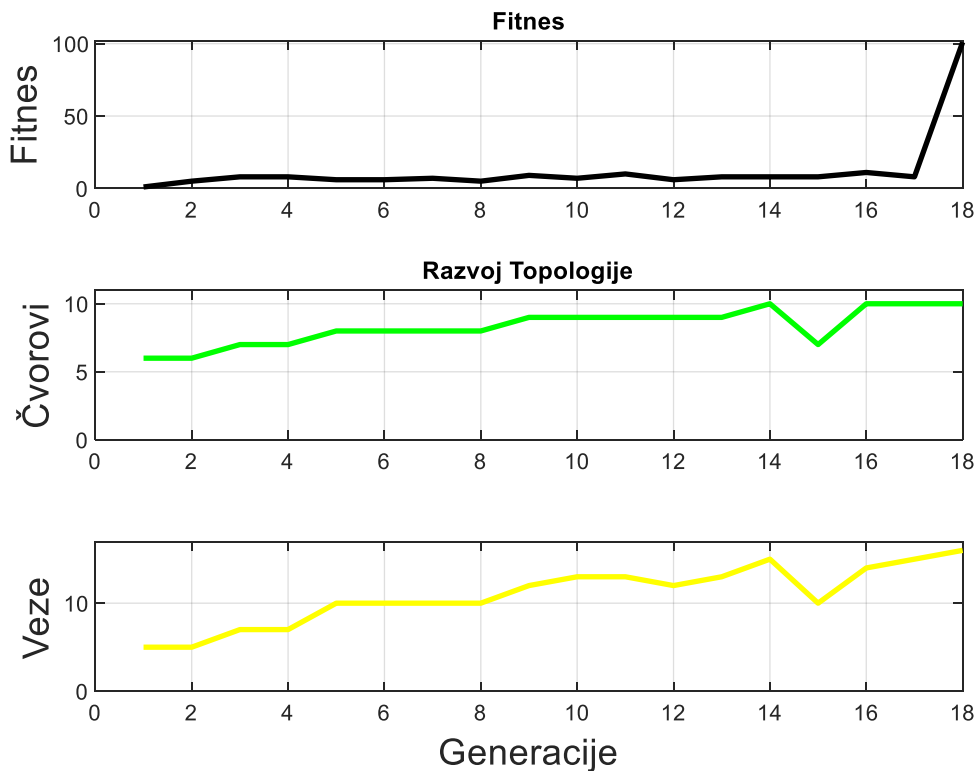
Zanimljivo je da je isti način stabilizacije primjenjen i u pokušaju tri, što pokazuje da različite mreže mogu uspješno doći do iste metode.

Grafički prikaz neuronske mreže



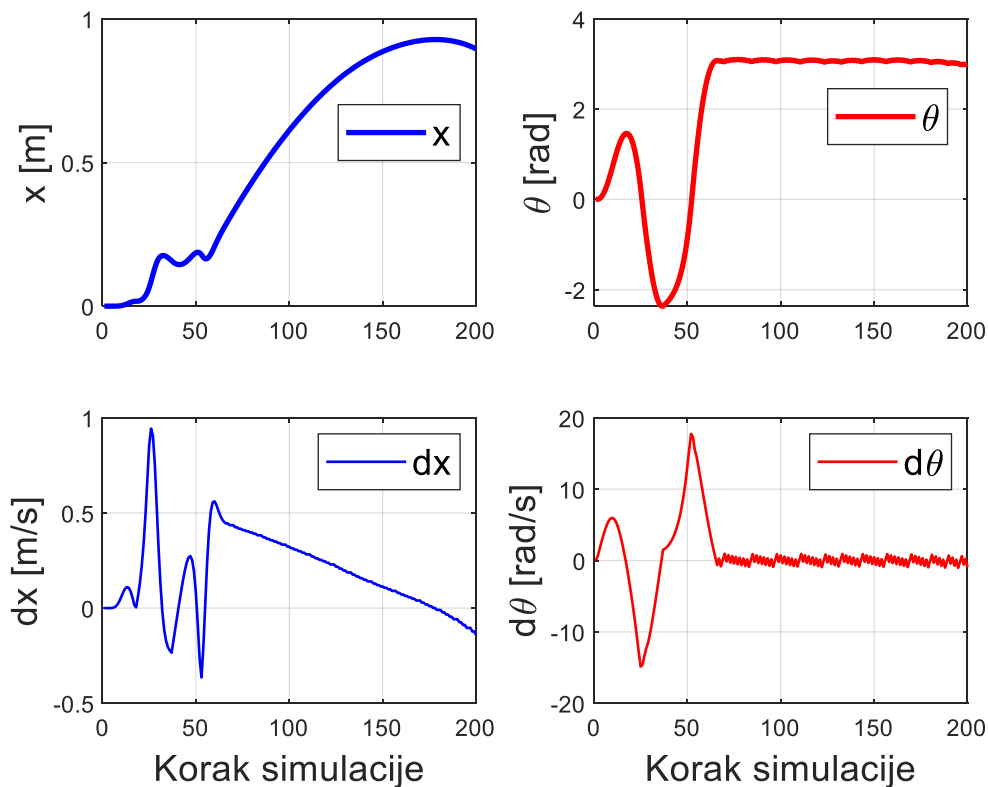
Slika 46. Mreža rješenja problema stabilizacije obrnutog njihala za 5. pokušaj

Na slici 46. može se vidjeti mreža koja je u prvom pokušaju riješila problem stabilizacije obrnutog njihala. Mreža je nastala nakon 18 generacija i 2850 procijenjenih mreža. Mreža ima ukupno 10 čvorova, od kojih su četiri ulazna, *bias*, jedan izlazni i četiri skrivena čvora te 16 omogućenih veza. Prvi ulaz x direktno je vezan sa izlazom F , te skrivenim čvorovima 6 i 8. Drugi ulaz θ direktno je vezan sa izlazom F te je također vezan sa skrivenim čvorom 7. Treći ulaz dx direktno je vezan sa izlazom F te je također vezan sa skrivenim čvorom 8. Četvrti ulaz $d\theta$ direktno je vezan sa izlazom F te je također vezan sa skrivenim čvorovima 8 i 9. *Bias* je direktno vezan sa izlazom F te je također vezan sa skrivenim čvorom 9. Skriveni čvor 9 povezan je na svom ulazu sa ulazom $d\theta$, a na izlazu je povezan sa izlazom F . Skriveni čvor 7 povezan je na svom ulazu sa ulazom θ , a na izlazu je povezan sa izlazom F . Skriveni čvor 8 povezan je na svom ulazu sa ulazima x , dx i $d\theta$, a na izlazu je povezan sa izlazom F . Skriveni čvor 6 povezan je na svom ulazu sa ulazom x i *Bias*-om, a na izlazu je povezan sa izlazom F . Izlaz F direktno je povezan sa ulazima x , θ , dx i $d\theta$, *Bias*-om te skrivenim čvorovima 6, 7, 8 i 9.



Slika 47. Grafovi fitnessa i razvoja topologije mreže problema stabilizacije obrnutog njihala za 5. pokušaj

Na slici 47. vidi se graf fitnessa i razvoj topologije mreže u prvom pokušaju. Konačni fitness je 16, konačni broj čvorova je 10, a konačni broj omogućenih veza je 16. Broj čvorova se povećava i smanjuje kako generacije prolaze, odnosno kako se izvršava križanje te mutacija čvorova. Broj omogućenih veza se povećava i smanjuje kako generacije prolaze, odnosno kako se izvršava križanje te mutacija veza. Dodavanje čvorova i veza obično u početku smanjuje fitness mreže, a oduzimanje čvorova i veza obično u početku povećava fitness mreže.



Slika 48. Grafovi varijabli stanja sustava obrnutog njihala za 5. pokušaj

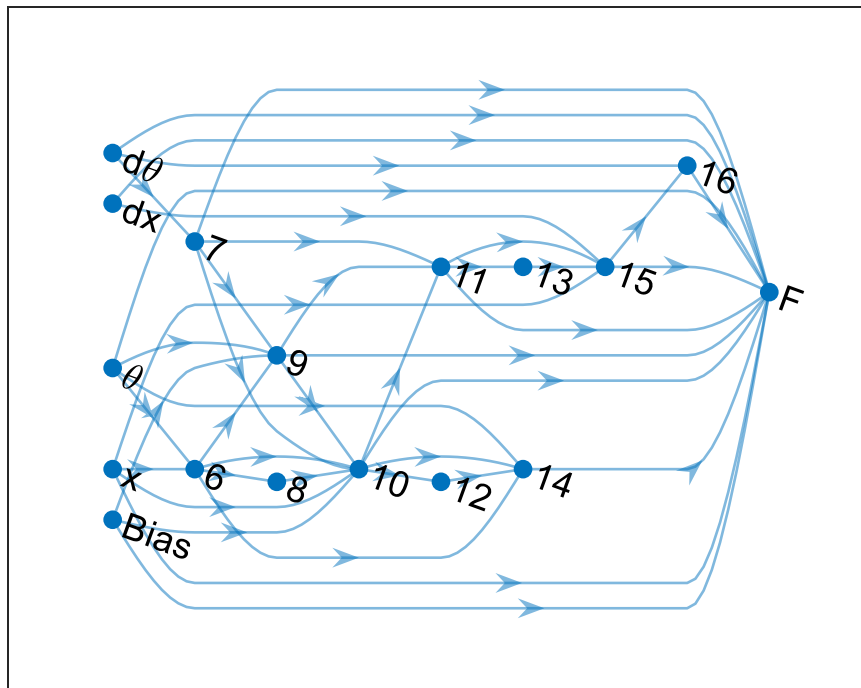
Na slici 48. prikazani su grafovi varijabli stanja sustava obrnutog njihala za 5. pokušaj.

U početnom trenutku obrnuto njihalo nalazi se u donjem stabilnom ravnotežnom položaju, što znači da njihalo slobodno visi tj. kut zakreta θ je jednak nuli kao i kutna brzina $\dot{\theta}$ i kutno ubrzanje $\ddot{\theta}$. Također to znači da su pomak kolica x , brzina kolica \dot{x} i ubrzanje kolica \ddot{x} također jednaki nuli.

U ovom pokušaju njihalo je prvo zanjihano iz donjeg stabilnog ravnotežnog položaja $\theta = 0$ do pozitivnog (desnog) horizontalnog položaja $\theta = \frac{\pi}{2}$, pada natrag u donji stabilni ravnotežni položaj, zatim je zanjihano u negativnom (lijevom) smjeru do otprilike $\theta = -\frac{3\pi}{4}$, pada natrag u donji stabilni ravnotežni položaj, a zatim je zanjihano do pozitivnog (desnog) gornjeg nestabilnog ravnotežnog položaja $\theta = \pi$ gdje je stabilizirano. Kolica se isprva ne gibaju, pokreću se tek kada se njihalo vrati u donji stabilni ravnotežni položaj, a zatim se gibaju na takav način da se osigura njihanje u negativnom, a zatim u pozitivnom smjeru kako bi se njihalo stabiliziralo. Kako bi se stabilizacija održala kolica se počinju gibati u suprotnom smjeru.

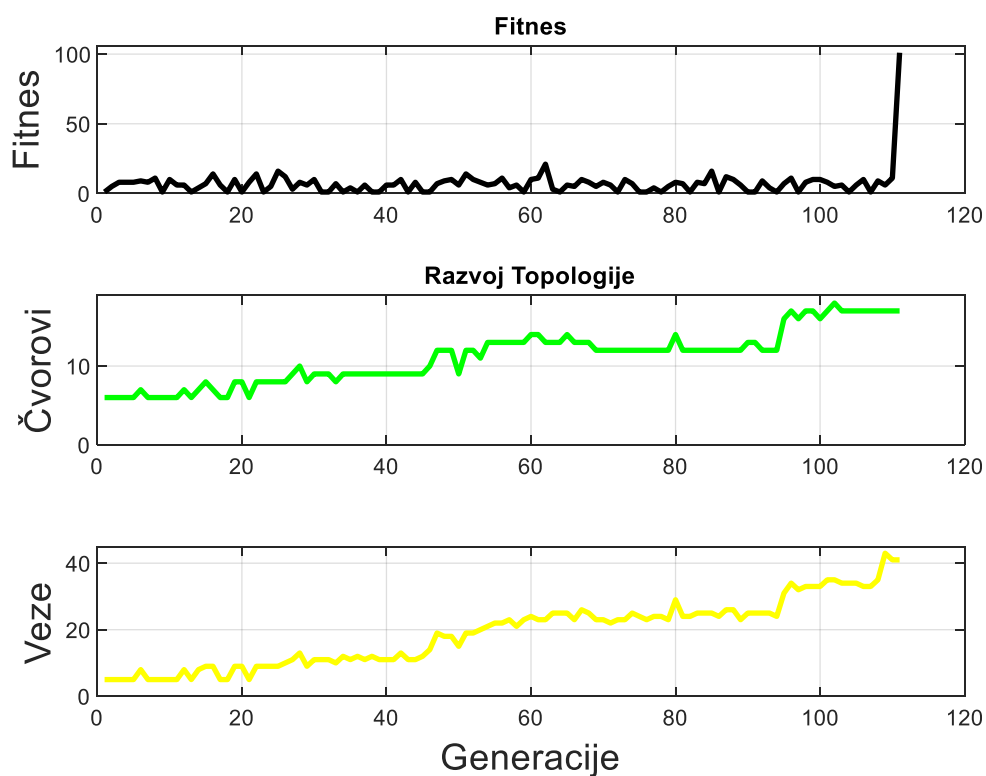
Zanimljivo je da je isti način stabilizacije primjenjen i u pokušaju 8, što pokazuje da različite mreže mogu uspješno doći do iste metode.

Grafički prikaz neuronske mreže



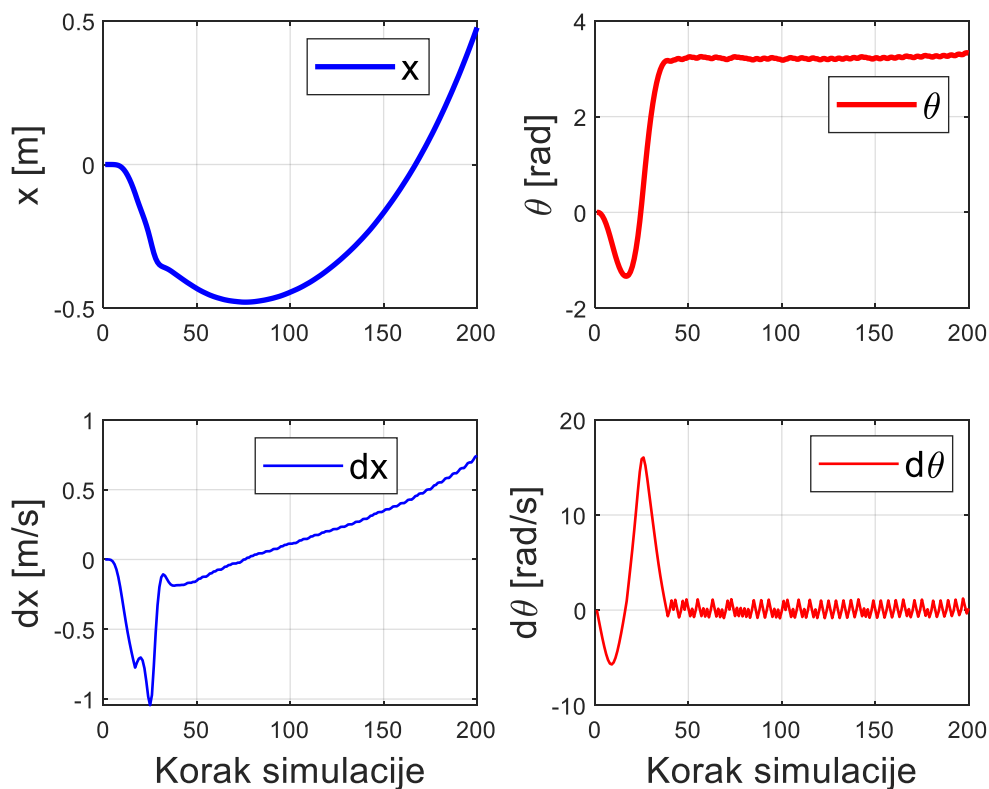
Slika 49. Mreža rješenja problema stabilizacije obrnutog njihala za 6. pokušaj

Na slici 49. može se vidjeti mreža koja je u prvom pokušaju riješila problem stabilizacije obrnutog njihala. Mreža je nastala nakon 111 generacija i 16800 procijenjenih mreža. Mreža ima ukupno 17 čvorova, od kojih su četiri ulazna, *bias*, jedan izlazni i 11 skrivenih čvorova te 41 omogućenu vezu. Prvi ulaz x direktno je vezan sa izlazom F , te skrivenim čvorovima 6, 10 i 15. Drugi ulaz θ direktno je vezan sa izlazom F te je također vezan sa skrivenim čvorovima 6, 9 i 14. Treći ulaz dx direktno je vezan sa izlazom F te je također vezan sa skrivenim čvorom 15. Četvrti ulaz $d\theta$ direktno je vezan sa izlazom F te je također vezan sa skrivenim čvorovima 7 i 16. *Bias* je direktno vezan sa izlazom F te je također vezan sa skrivenim čvorovima 9 i 10. Skriveni čvor 7 povezan je na svom ulazu sa ulazom $d\theta$, a na izlazu je povezan sa izlazom F te skrivenim čvorovima 9, 10 i 11. Skriveni čvor 6 povezan je na svom ulazu sa ulazom x , a na izlazu je povezan sa skrivenim čvorom 14. Skriveni čvor 9 povezan je na svom ulazu sa ulazom θ i *Bias*-om te je također vezan sa skrivenim čvorovima 6 i 7, a na izlazu je povezan sa izlazom F te skrivenim čvorovima 10 i 11. Skriveni čvor 8 povezan je na svom ulazu sa skrivenim čvorom 6, a na izlazu je povezan sa skrivenim čvorom 10. Skriveni čvor 10 povezan je na svom ulazu sa ulazom x i *Bias*-om te je također vezan sa skrivenim čvorovima 6, 7 i 8, a na izlazu je povezan sa izlazom F te skrivenim čvorovima 11, 12 i 14. Skriveni čvor 11 povezan je na svom ulazu sa skrivenim čvorovima 7, 9 i 10, a na izlazu je povezan sa izlazom F te skrivenim čvorovima 13 i 15. Skriveni čvor 12 povezan je na svom ulazu sa skrivenim čvorom 10, a na izlazu je povezan sa skrivenim čvorom 14. Skriveni čvor 13 povezan je na svom ulazu sa skrivenim čvorom 11, a na izlazu je povezan sa skrivenim čvorom 15. Skriveni čvor 14 povezan je na svom ulazu sa ulazom θ te skrivenim čvorovima 6, 10 i 12, a na izlazu je povezan sa izlazom F . Skriveni čvor 15 povezan je na svom ulazu sa ulazima x i dx te je također vezan sa skrivenim čvorovima 11 i 13, a na izlazu je povezan sa izlazom F te skrivenim čvorom 16. Skriveni čvor 16 povezan je na svom ulazu sa ulazom $d\theta$ te je također vezan sa skrivenim čvorom 15, a na izlazu je povezan sa izlazom F . Izlaz F direktno je povezan sa ulazima x , θ , dx i $d\theta$, *Bias*-om te skrivenim čvorovima 7, 9, 10, 11, 14, 15 i 16.



Slika 50. Grafovi fitnessa i razvoja topologije mreže problema stabilizacije obrnutog njihala za 6. pokušaj

Na slici 50. vidi se graf fitnessa i razvoj topologije mreže u prvom pokušaju. Konačni fitness je 101, konačni broj čvorova je 17, a konačni broj omogućenih veza je 41. Broj čvorova se povećava i smanjuje kako generacije prolaze, odnosno kako se izvršava križanje te mutacija čvorova. Broj omogućenih veza se povećava i smanjuje kako generacije prolaze, odnosno kako se izvršava križanje te mutacija veza. Dodavanje čvorova i veza obično u početku smanjuje fitness mreže, a oduzimanje čvorova i veza obično u početku povećava fitness mreže.



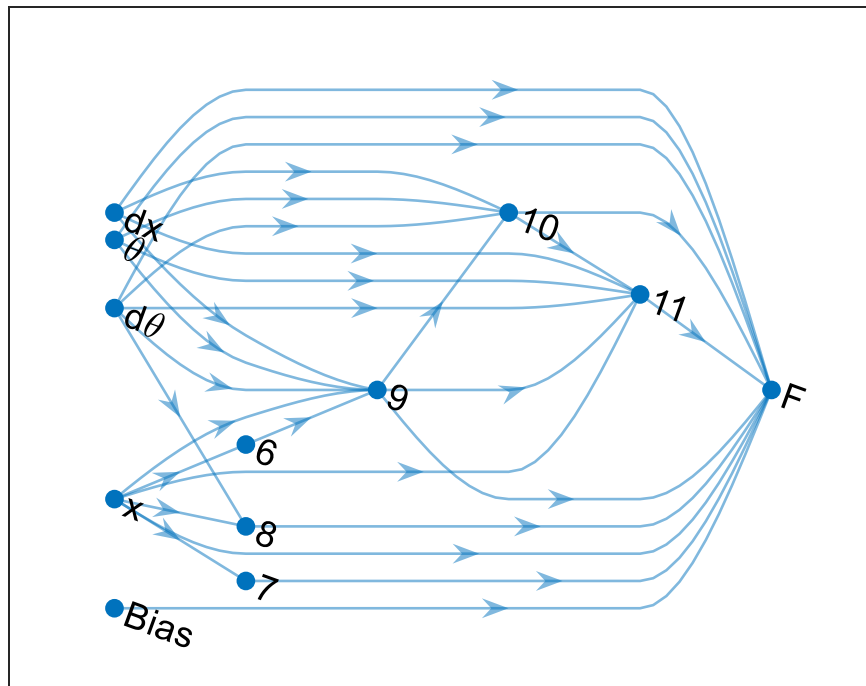
Slika 51. Grafovi varijabli stanja sustava obrnutog njihala za 6. pokušaj

Na slici 48. prikazani su grafovi varijabli stanja sustava obrnutog njihala za 6. pokušaj.

U početnom trenutku obrnuto njihalo nalazi se u donjem stabilnom ravnotežnom položaju, što znači da njihalo slobodno visi tj. kut zakreta θ je jednak nuli kao i kutna brzina $\dot{\theta}$ i kutno ubrzanje $\ddot{\theta}$. Također to znači da su pomak kolica x , brzina kolica \dot{x} i ubrzanje kolica \ddot{x} također jednaki nuli.

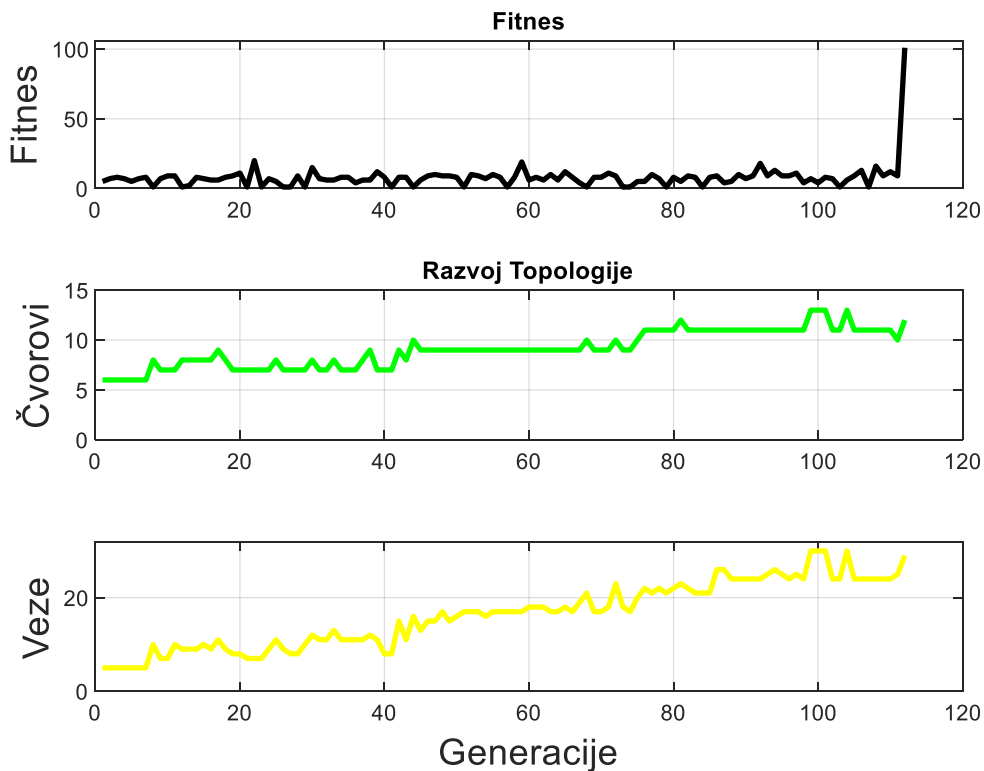
U ovom pokušaju njihalo je prvo zanjihano iz donjeg stabilnog ravnotežnog položaja $\theta = 0$ do pozitivnog (desnog) horizontalnog položaja $\theta = \frac{\pi}{2}$, pada natrag u donji stabilni ravnotežni položaj, zatim je zanjihano u negativnom (lijevom) smjeru do otprilike $\theta = -\frac{3\pi}{4}$, pada natrag u donji stabilni ravnotežni položaj, a zatim je zanjihano do pozitivnog (desnog) gornjeg nestabilnog ravnotežnog položaja $\theta = \pi$ gdje je stabilizirano. Kolica se isprva ne gibaju, pokreću se tek kada se njihalo vrati u donji stabilni ravnotežni položaj, a zatim se gibaju na takav način da se osigura njihanje u negativnom, a zatim u pozitivnom smjeru kako bi se njihalo stabiliziralo. Kako bi se stabilizacija održala kolica se počinju gibati u suprotnom smjeru.

Grafički prikaz neuronske mreže



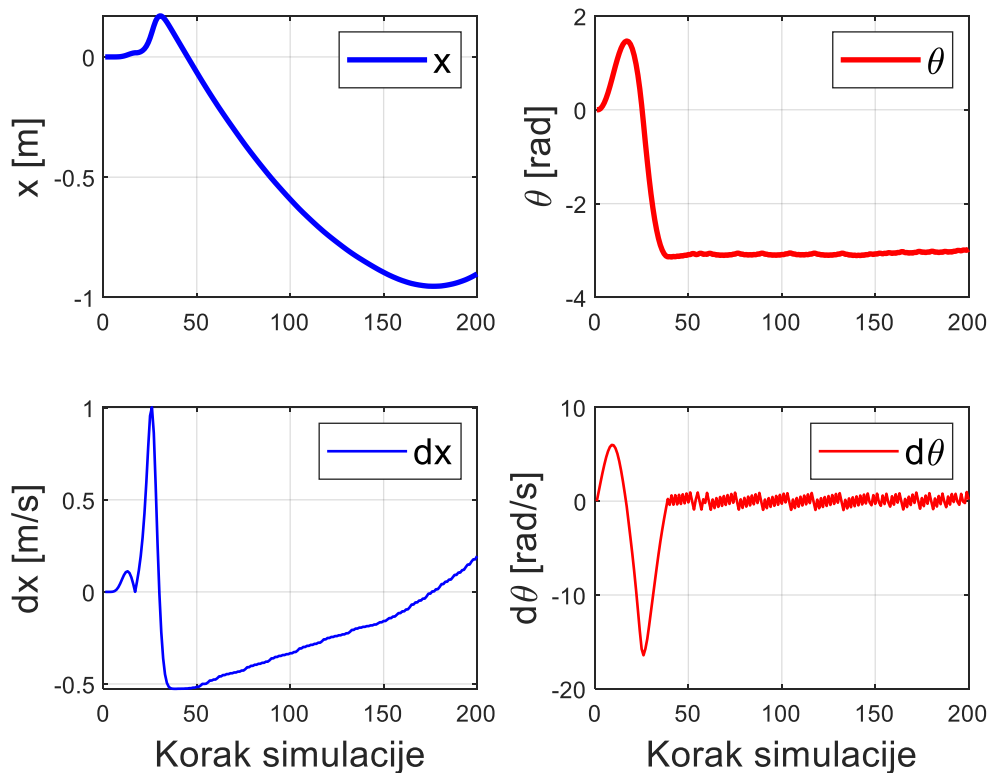
Slika 52. Mreža rješenja problema stabilizacije obrnutog njihala za 7. pokušaj

Na slici 52. može se vidjeti mreža koja je u prvom pokušaju riješila problem stabilizacije obrnutog njihala. Mreža je nastala nakon 112 generacija i 16950 procijenjenih mreža. Mreža ima ukupno 12 čvorova, od kojih su četiri ulazna, *bias*, jedan izlazni i 6 skrivenih čvorova te 29 omogućenih veza. Prvi ulaz x direktno je vezan sa izlazom F , te skrivenim čvorovima 6, 7, 8, 9 i 11. Drugi ulaz θ direktno je vezan sa izlazom F te je također vezan sa skrivenim čvorovima 9, 10 i 11. Treći ulaz dx direktno je vezan sa izlazom F te je također vezan sa skrivenim čvorovima 9, 10 i 11. Četvrti ulaz $d\theta$ direktno je vezan sa izlazom F te je također vezan sa skrivenim čvorovima 8, 9 i 10. *Bias* je direktno vezan sa izlazom F . Skriveni čvor 6 povezan je na svom ulazu sa ulazom x , a na izlazu je povezan sa skrivenim čvorom 9. Skriveni čvor 8 povezan je na svom ulazu sa ulazima x i $d\theta$, a na izlazu je povezan sa izlazom F . Skriveni čvor 7 povezan je na svom ulazu sa ulazom x , a na izlazu je povezan sa izlazom F . Skriveni čvor 9 povezan je na svom ulazu sa ulazima x , θ , dx i $d\theta$ te je također vezan sa skrivenim čvorom 6, a na izlazu je povezan sa izlazom F te je također vezan sa skrivenim čvorovima 10 i 11. Skriveni čvor 10 povezan je na svom ulazu sa ulazima θ , dx i $d\theta$ te je također vezan sa skrivenim čvorom 9, a na izlazu je povezan sa izlazom F te je također vezan sa skrivenim čvorom 11. Skriveni čvor 11 povezan je na svom ulazu sa ulazima x , θ , dx i $d\theta$ te skrivenim čvorovima 9 i 10, a na izlazu je povezan sa izlazom F . Izlaz F direktno je povezan sa ulazima x , θ , dx i $d\theta$, *Bias*-om te skrivenim čvorovima 7, 8, 9, 10 i 11.



Slika 53. Grafovi fitnessa i razvoja topologije mreže problema obrnutog njihala za 7. pokušaj

Na slici 53. vidi se graf fitnessa i razvoj topologije mreže u prvom pokušaju. Konačni fitness je 101, konačni broj čvorova je 12, a konačni broj omogućenih veza je 29. Broj čvorova se povećava i smanjuje kako generacije prolaze, odnosno kako se izvršava križanje te mutacija čvorova. Broj omogućenih veza se povećava i smanjuje kako generacije prolaze, odnosno kako se izvršava križanje te mutacija veza. Dodavanje čvorova i veza obično u početku smanjuje fitness mreže, a oduzimanje čvorova i veza obično u početku povećava fitness mreže.



Slika 54. Grafovi varijabli stanja sustava obrnutog njihala za 7. pokušaj

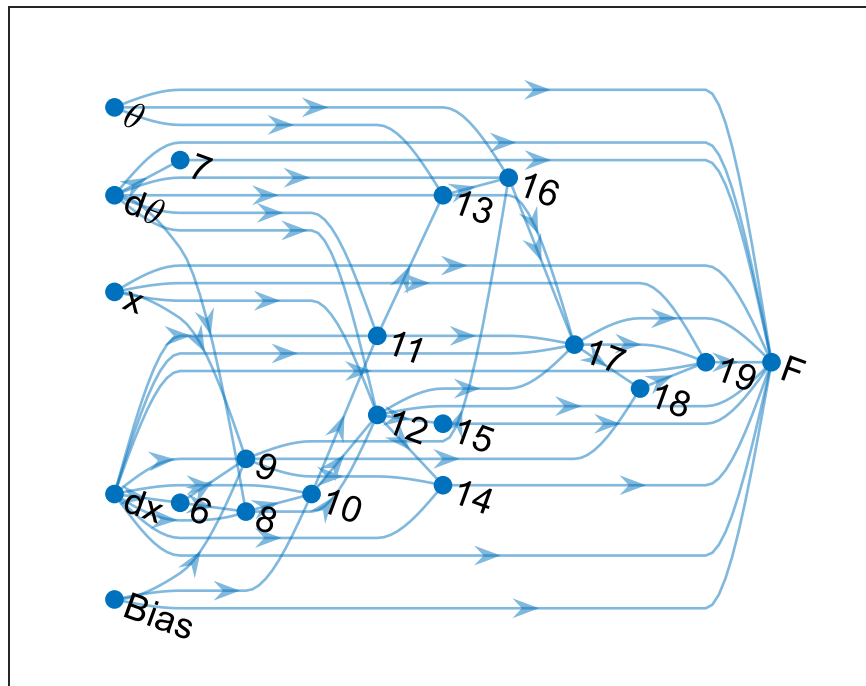
Na slici 54. prikazani su grafovi varijabli stanja sustava obrnutog njihala za 7. pokušaj.

U početnom trenutku obrnuto njihalo nalazi se u donjem stabilnom ravnotežnom položaju, što znači da njihalo slobodno visi tj. kut zakreta θ je jednak nuli kao i kutna brzina $\dot{\theta}$ i kutno ubrzanje $\ddot{\theta}$. Također to znači da su pomak kolica x , brzina kolica \dot{x} i ubrzanje kolica \ddot{x} također jednaki nuli.

U ovom pokušaju njihalo je prvo zanjihano iz donjeg stabilnog ravnotežnog položaja $\theta = 0$ do pozitivnog (desnog) horizontalnog položaja $\theta = \frac{\pi}{2}$, pada natrag u donji stabilni ravnotežni položaj, a zatim je zanjihano do negativnog (lijevog) gornjeg nestabilnog položaja $\theta = \pi$ gdje je stabilizirano. Kolica se isprva ne gibaju, pokreću se tek kada se njihalo vrati u donji stabilni ravnotežni položaj, te se nastavljaju kontinuirano gibati po stazi kako bi se njihalo dovelo do gornjeg nestabilnog ravnotežnog položaja, a zatim kako bi se stabiliziralo. Kada bi kolica došla do kraja staze njihalo bi palo natrag u donji stabilni ravnotežni položaj. Kako bi se stabilizacija održala nastavlja se kontinuirano gibati, a zatim se počinje kretati unatrag.

Zanimljivo je da je sličan način stabilizacije primjenjen i u pokušaju jedan, što pokazuje da različite mreže mogu uspješno doći do iste metode.

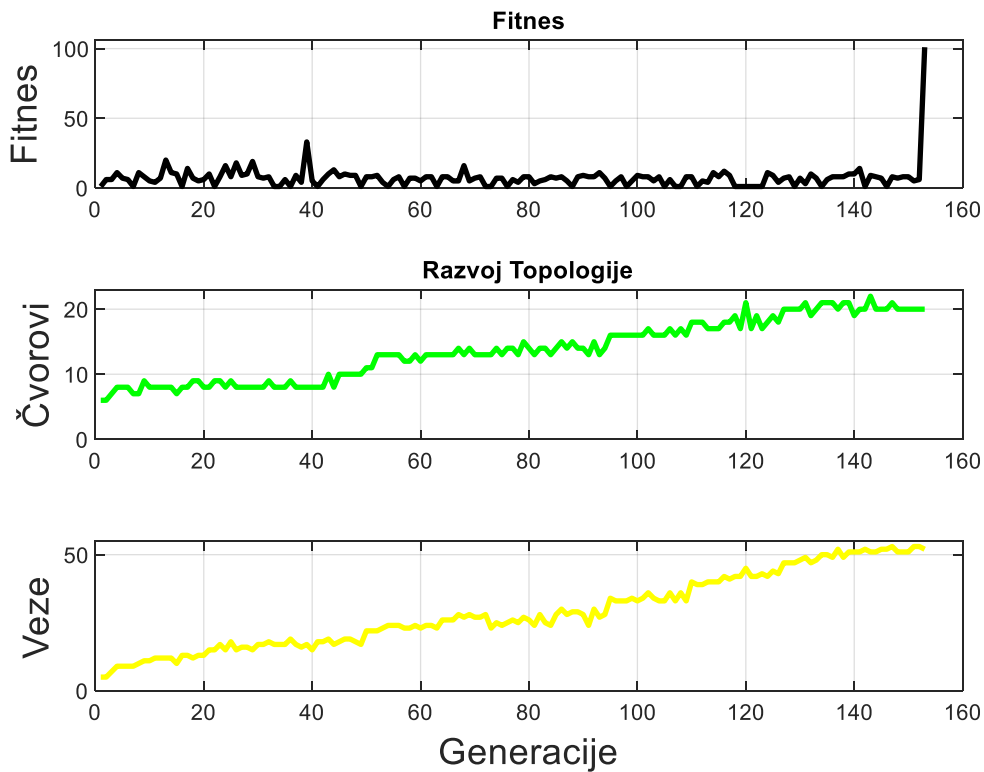
Grafički prikaz neuronske mreže



Slika 55. Mreža rješenja problema stabilizacije obrnutog njihala za 8. pokušaj

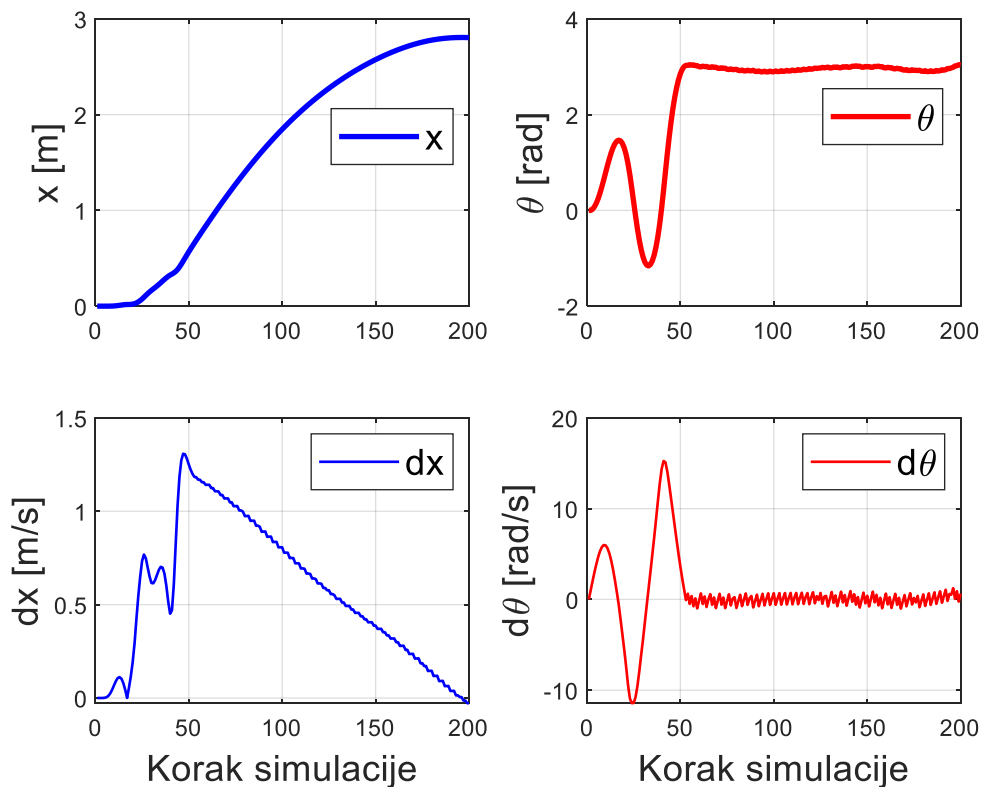
Na slici 55. može se vidjeti mreža koja je u prvom pokušaju riješila problem stabilizacije obrnutog njihala. Mreža je nastala nakon 153 generacija i 23100 procijenjenih mreža. Mreža ima ukupno 20 čvorova, od kojih su četiri ulazna, *bias*, jedan izlazni i 14 skrivenih čvorova te 52 omogućene veze. Prvi ulaz x direktno je vezan sa izlazom F , te skrivenim čvorovima 9, 12 i 19. Drugi ulaz θ direktno je vezan sa izlazom F te je također vezan sa skrivenim čvorovima 13 i 16. Treći ulaz dx direktno je vezan sa izlazom F te je također vezan sa skrivenim čvorovima 6, 8, 9, 10, 11, 14, 17 i 19. Četvrti ulaz $d\theta$ direktno je vezan sa izlazom F te je također vezan sa skrivenim čvorovima 7, 8, 11, 12, 13 i 16. *Bias* je direktno vezan sa izlazom F te je također vezan sa skrivenim čvorovima 9 i 10. Skriveni čvor 7 povezan je na svom ulazu sa ulazom $d\theta$, a na izlazu je povezan sa izlazom F . Skriveni čvor 6 povezan je na svom ulazu sa ulazom dx , a na izlazu je povezan sa skrivenim čvorovima 8 i 9. Skriveni čvor 9 povezan je na svom ulazu sa ulazima x i dx , *Bias*-om te je također vezan sa skrivenim čvorom 6, a na izlazu je povezan sa skrivenim čvorovima 14, 16 i 18. Skriveni čvor 8 povezan je na svom ulazu ulazima dx i $d\theta$ te je također vezan sa skrivenim čvorom 6, a na izlazu je povezan sa skrivenim čvorovima 10 i 12. Skriveni čvor 10 povezan je na svom ulazu sa ulazom dx te je također vezan sa skrivenim čvorom 8, a na izlazu je povezan sa skrivenim čvorom 12. Skriveni čvor 11 povezan je na svom ulazu sa ulazima dx i $d\theta$, *Bias*-om, a na izlazu je povezan sa skrivenim čvorovima 13 i 17. Skriveni čvor 12 povezan je na svom ulazu sa x i $d\theta$ te je također vezan sa skrivenim čvorovima 8 i 10, a na izlazu je povezan sa izlazom F te skrivenim čvorovima 14, 15 i 17. Skriveni čvor 13 povezan je na svom ulazu sa ulazom θ i $d\theta$ te skrivenim čvorom 11, a na izlazu je povezan sa skrivenim čvorovima 16 i 17. Skriveni čvor 15 povezan je na svom ulazu sa skrivenim čvorom 12, a na izlazu je povezan sa izlazom F . Skriveni čvor 14 povezan je na svom ulazu sa ulazom dx te je također vezan sa skrivenim čvorovima 9 i 12, a na izlazu je povezan sa izlazom F . Skriveni čvor 16 povezan je na svom ulazu sa ulazima θ i $d\theta$ te je također vezan sa skrivenim čvorovima 9 i 13, a na izlazu je povezan sa skrivenim čvorom 17. Skriveni čvor 17 povezan je na svom ulazu sa ulazom dx te je također vezan sa skrivenim čvorovima 11, 12, 13 i 16, a na

izlazu je povezan sa izlazom F te skrivenim čvorovima 18 i 19. Skriveni čvor 18 povezan je na svom ulazu sa skrivenim čvorovima 9 i 17, a na izlazu je povezan sa skrivenim čvorom 19. Skriveni čvor 19 povezan je na svom ulazu sa ulazima x i dx te je također vezan sa skrivenim čvorovima 17 i 18, a na izlazu je povezan sa izlazom F . Izlaz F direktno je povezan sa ulazima x , θ , dx i $d\theta$, Bias-om te skrivenim čvorovima 7, 12, 14, 15, 17 i 19.



Slika 56. Grafovi fitnesa i razvoja topologije mreže problema stabilizacije obrnutog njihala za 8. pokušaj

Na slici 56. vidi se graf fitnesa i razvoj topologije mreže u prvom pokušaju. Konačni fitnes je 101, konačni broj čvorova je 20, a konačni broj omogućenih veza je 52. Broj čvorova se povećava i smanjuje kako generacije prolaze, odnosno kako se izvršava križanje te mutacija čvorova. Broj omogućenih veza se povećava i smanjuje kako generacije prolaze, odnosno kako se izvršava križanje te mutacija veza. Dodavanje čvorova i veza obično u početku smanjuje fitnes mreže, a oduzimanje čvorova i veza obično u početku povećava fitnes mreže.



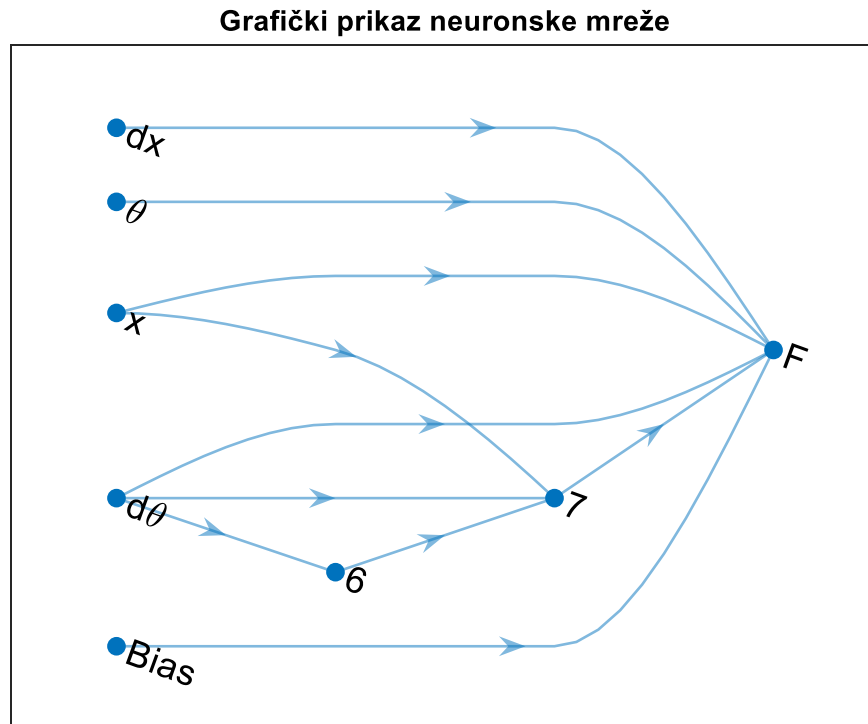
Slika 57. Grafovi varijabli stanja sustava obrnutog njihala za 8. pokušaj

Na slici 57. prikazani su grafovi varijabli stanja sustava obrnutog njihala za 8. pokušaj.

U početnom trenutku obrnuto njihalo nalazi se u donjem stabilnom ravnotežnom položaju, što znači da njihalo slobodno visi tj. kut zakreta θ je jednak nuli kao i kutna brzina $\dot{\theta}$ i kutno ubrzanje $\ddot{\theta}$. Također to znači da su pomak kolica x , brzina kolica \dot{x} i ubrzanje kolica \ddot{x} također jednaki nuli.

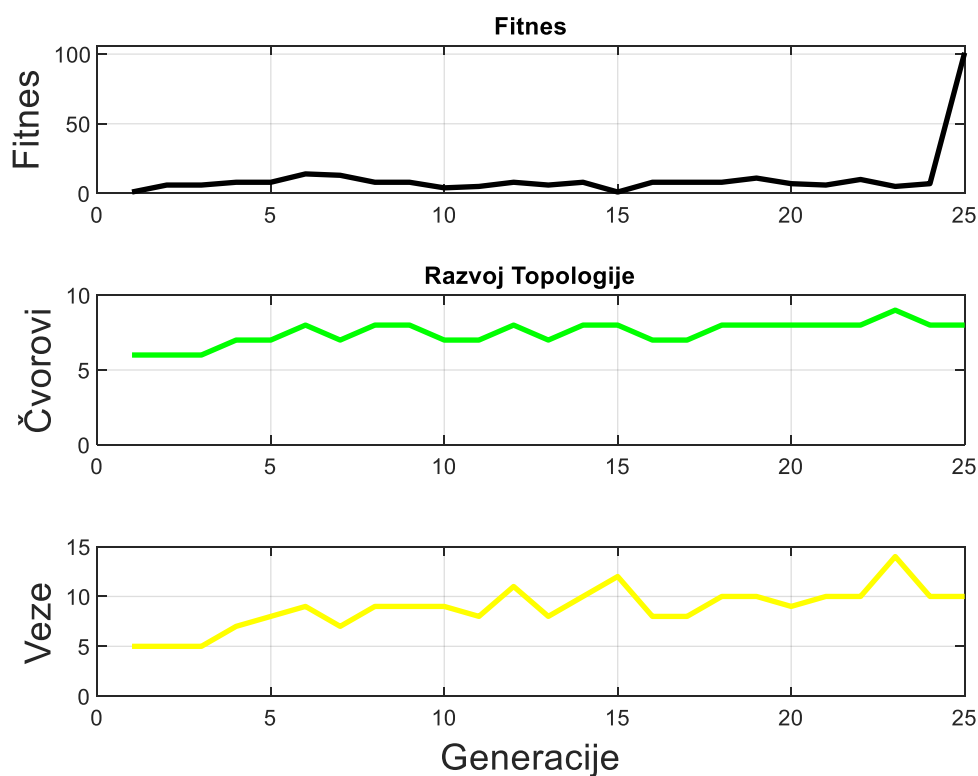
U ovom pokušaju njihalo je prvo zanjihano iz donjeg stabilnog ravnotežnog položaja $\theta = 0$ do pozitivnog (desnog) horizontalnog položaja $\theta = \frac{\pi}{2}$, pada natrag u donji stabilni ravnotežni položaj, zatim je zanjihano do negativnog (lijevog) horizontalnog položaja $\theta = -\frac{\pi}{2}$, pada natrag u donji stabilni ravnotežni položaj, a zatim je zanjihano do pozitivnog (desnog) gornjeg nestabilnog ravnotežnog položaja $\theta = \pi$ gdje je stabilizirano. Kolica se isprva ne gibaju, pokreću se tek kada se njihalo vrati u donji stabilni ravnotežni položaj, a zatim se gibaju na takav način da se osigura njihanje u negativnom, a zatim u pozitivnom smjeru kako bi se njihalo stabiliziralo. Kako bi se stabilizacija održala kolica se počinju gibati u suprotnom smjeru.

Zanimljivo je da je isti način stabilizacije primjenjen i u pokušaju 5, što pokazuje da različite mreže mogu uspješno doći do iste metode.



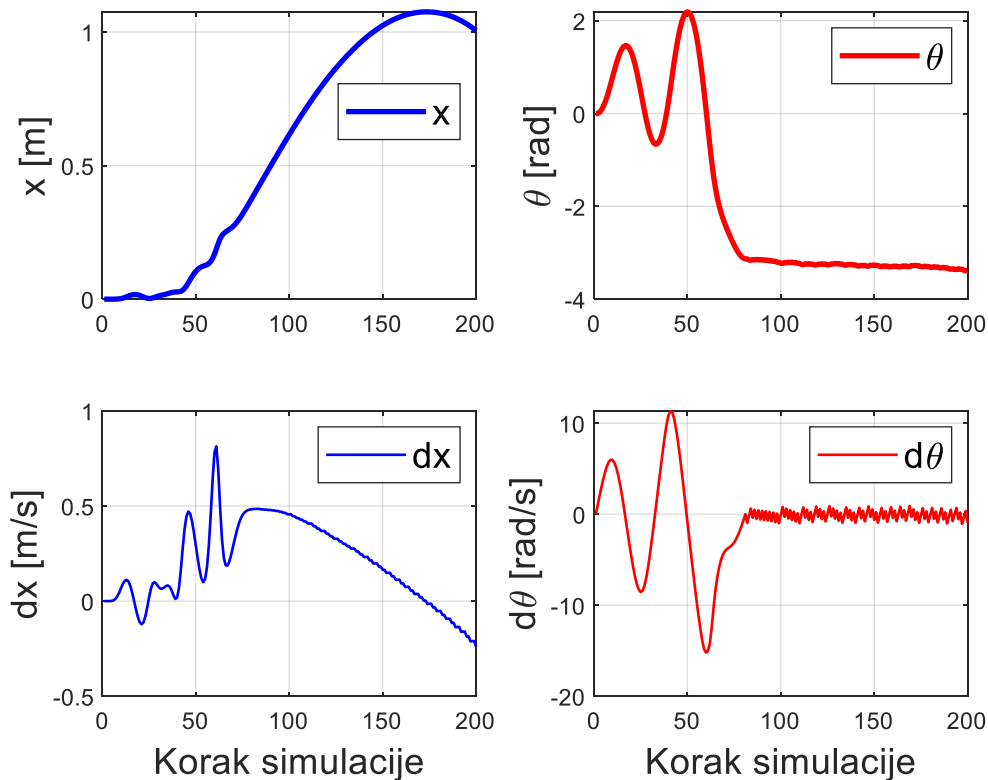
Slika 58. Mreža rješenja problema stabilizacije obrnutog njihala za 9. pokušaj

Na slici 58. može se vidjeti mreža koja je u prvom pokušaju riješila problem stabilizacije obrnutog njihala. Mreža je nastala nakon 25 generacija i 3900 procijenjenih mreža. Mreža ima ukupno 8 čvorova, od kojih su četiri ulazna, *bias*, jedan izlazni i 2 skrivenih čvorova te 10 omogućenih veza. Prvi ulaz x direktno je vezan sa izlazom F te je također vezan sa skrivenim čvorom 7. Drugi ulaz θ direktno je vezan sa izlazom F . Treći ulaz dx direktno je vezan sa izlazom F . Četvrti ulaz $d\theta$ direktno je vezan sa izlazom F te je također vezan sa skrivenim čvorovima 6 i 7. *Bias* je direktno vezan sa izlazom F . Skriveni čvor 6 povezan je na svom ulazu sa ulazom $d\theta$, a na izlazu je povezan sa skrivenim čvorom 7. Skriveni čvor 7 povezan je na svom ulazu sa ulazima x i $d\theta$ te je također vezan sa skrivenim čvorom 6, a na izlazu je povezan sa izlazom F . Izlaz F direktno je povezan sa ulazima x , θ , dx i $d\theta$, *Bias*-om te skrivenim čvorom 7.



Slika 59. Grafovi fitnessa i razvoja topologije mreže problema stabilizacije obrnutog njihala za 9. pokušaj

Na slici 59. vidi se graf fitnessa i razvoj topologije mreže u prvom pokušaju. Konačni fitness je 101, konačni broj čvorova je 8, a konačni broj omogućenih veza je 10. Broj čvorova se povećava i smanjuje kako generacije prolaze, odnosno kako se izvršava križanje te mutacija čvorova. Broj omogućenih veza se povećava i smanjuje kako generacije prolaze, odnosno kako se izvršava križanje te mutacija veza. Dodavanje čvorova i veza obično u početku smanjuje fitness mreže, a oduzimanje čvorova i veza obično u početku povećava fitness mreže.



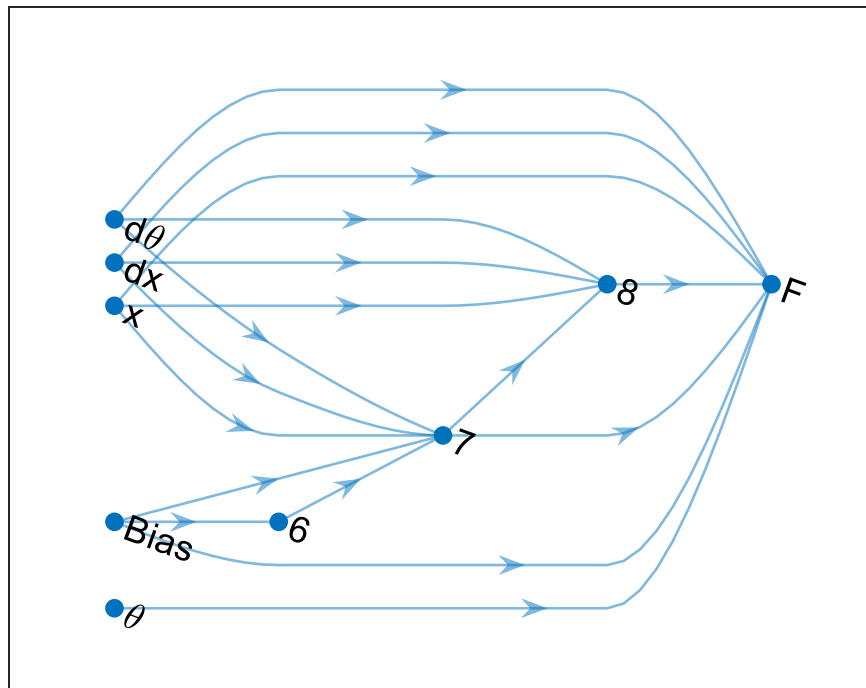
Slika 60. Grafovi varijabli stanja sustava obrnutog njihala za 9. pokušaj

Na slici 60. prikazani su grafovi varijabli stanja sustava obrnutog njihala za 9. pokušaj.

U početnom trenutku obrnuto njihalo nalazi se u donjem stabilnom ravnotežnom položaju, što znači da njihalo slobodno visi tj. kut zakreta θ je jednak nuli kao i kutna brzina $\dot{\theta}$ i kutno ubrzanje $\ddot{\theta}$. Također to znači da su pomak kolica x , brzina kolica \dot{x} i ubrzanje kolica \ddot{x} također jednaki nuli.

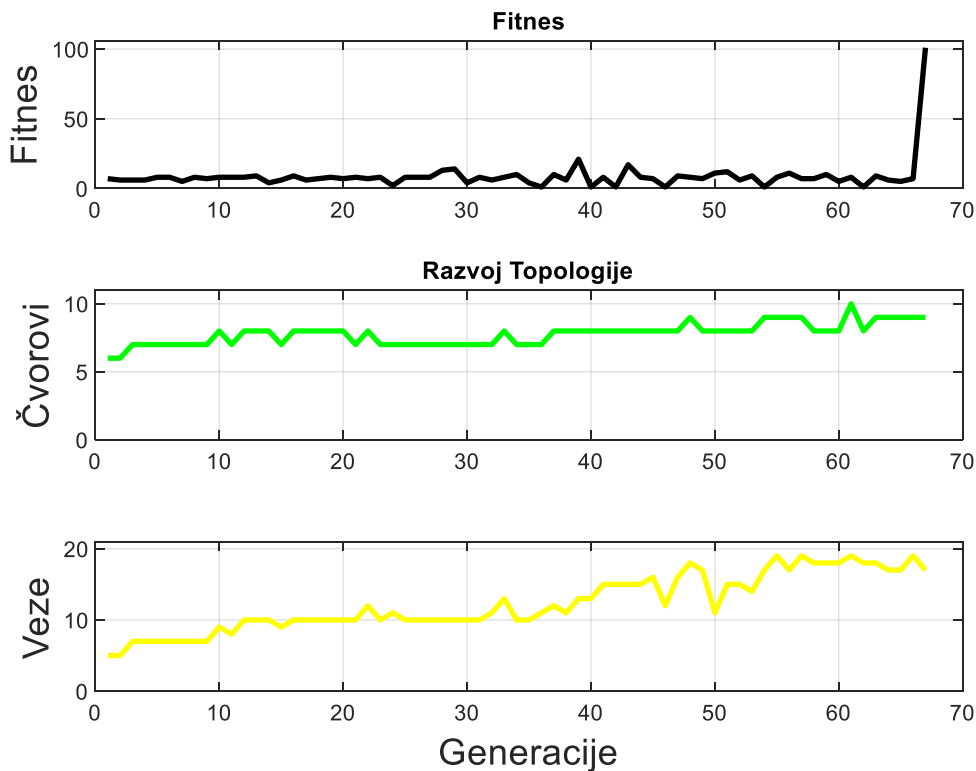
U ovom pokušaju njihalo je prvo zanjihano iz donjeg stabilnog ravnotežnog položaja $\theta = 0$ do pozitivnog (desnog) horizontalnog položaja $\theta = \frac{\pi}{2}$, pada natrag u donji stabilni ravnotežni položaj, zatim je zanjihano otprilike do negativnog (lijevog) položaja $\theta = -\frac{\pi}{4}$, pada natrag u donji stabilni ravnotežni položaj, potom je zanjihano otprilike do pozitivnog (desnog) položaja $\theta = \frac{3\pi}{4}$, pada natrag u donji stabilni ravnotežni položaj, a zatim je zanjihano do negativnog (lijevog) gornjeg nestabilnog ravnotežnog položaja $\theta = -\pi$ gdje je stabilizirano. Kolica se isprva ne gibaju, pokreću se tek kada se njihalo vrati u donji stabilni ravnotežni položaj, a zatim se gibaju na takav način da se osigura stabilizacija njihala. Kako bi se stabilizacija održala kolica se počinju gibati u suprotnom smjeru.

Grafički prikaz neuronske mreže



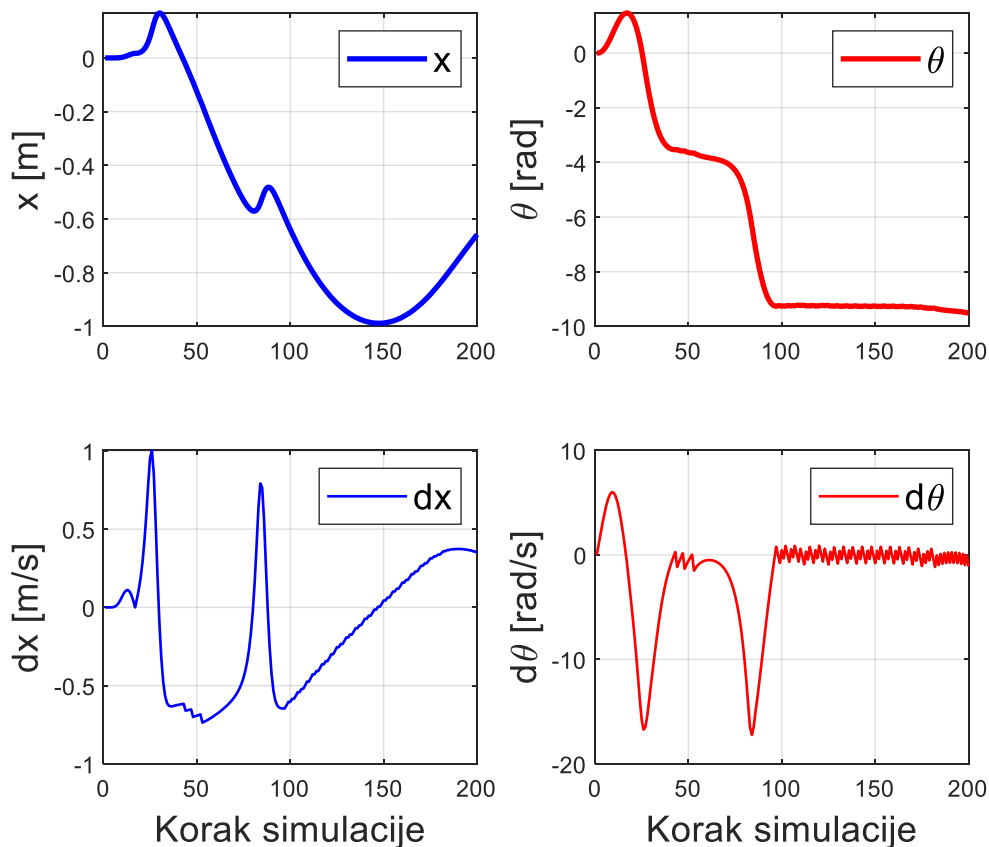
Slika 61. Mreža rješenja problema stabilizacije obrnutog njihala za 10. pokušaj

Na slici 61. može se vidjeti mreža koja je u prvom pokušaju riješila problem stabilizacije obrnutog njihala. Mreža je nastala nakon 67 generacija i 10200 procijenjenih mreža. Mreža ima ukupno 9 čvorova, od kojih su četiri ulazna, *bias*, jedan izlazni i tri skrivenih čvorova te 17 omogućenih veza. Prvi ulaz x direktno je vezan sa izlazom F te je također vezan sa skrivenim čvorovima 7 i 8. Drugi ulaz θ direktno je vezan sa izlazom F . Treći ulaz dx direktno je vezan sa izlazom F te je također vezan sa skrivenim čvorovima 7 i 8. Četvrti ulaz $d\theta$ direktno je vezan sa izlazom F te je također vezan sa skrivenim čvorovima 7 i 8. *Bias* je direktno vezan sa izlazom F te je također vezan sa skrivenim čvorovima 6 i 7. Skriveni čvor 6 povezan je na svom ulazu sa *Bias*-om, a na izlazu je povezan sa skrivenim čvorom 7. Skriveni čvor 7 povezan je na svom ulazu sa ulazima x , dx , $d\theta$ i *Bias*-om te je također vezan sa skrivenim čvorom 6, a na izlazu je povezan sa izlazom F te je također vezan sa skrivenim čvorom 8. Skriveni čvor 8 povezan je na svom ulazu sa ulazima x , dx , $d\theta$ i *Bias*-om te je također vezan sa skrivenim čvorom 7, a na izlazu je povezan sa izlazom F . Izlaz F direktno je povezan sa ulazima x , θ , dx i $d\theta$, *Bias*-om te skrivenim čvorovima 7 i 9.



Slika 62. Grafovi fitnessa i razvoja topologije mreže problema stabilizacije obrnutog njihala za 10. pokušaj

Na slici 62. vidi se graf fitnessa i razvoj topologije mreže u prvom pokušaju. Konačni fitness je 101, konačni broj čvorova je 9, a konačni broj omogućenih veza je 17. Broj čvorova se povećava i smanjuje kako generacije prolaze, odnosno kako se izvršava križanje te mutacija čvorova. Broj omogućenih veza se povećava i smanjuje kako generacije prolaze, odnosno kako se izvršava križanje te mutacija veza. Dodavanje čvorova i veza obično u početku smanjuje fitness mreže, a oduzimanje čvorova i veza obično u početku povećava fitness mreže.



Slika 63. Grafovi varijabli stanja sustava obrnutog njihala za 10. pokušaj

Na slici 63. prikazani su grafovi varijabli stanja sustava obrnutog njihala za 10. pokušaj.

U početnom trenutku obrnuto njihalo nalazi se u donjem stabilnom ravnotežnom položaju, što znači da njihalo slobodno visi tj. kut zakreta θ je jednak nuli kao i kutna brzina $\dot{\theta}$ i kutno ubrzanje $\ddot{\theta}$. Također to znači da su pomak kolica x , brzina kolica \dot{x} i ubrzanje kolica \ddot{x} također jednaki nuli.

U ovom pokušaju njihalo je prvo zanjihano iz donjeg stabilnog ravnotežnog položaja $\theta = 0$ do pozitivnog (desnog) horizontalnog položaja $\theta = \frac{\pi}{2}$, pada natrag u donji stabilni ravnotežni položaj, a zatim je zanjihano do negativnog (lijevog) gornjeg nestabilnog položaja $\theta = -\pi$ gdje se kratkotrajno stabilizira, ali nedovoljno dugo te je zatim zanjihano na vrlo pametan način da napravi puni krug u smjeru kazaljke na satu (-2π), vrati se u gornji nestabilni položaj $\theta = -3\pi$ gdje se ovaj put uspješno stabilizira. Kolica se isprva ne gibaju, pokreću se tek kada se njihalo vrati u donji stabilni ravnotežni položaj, a zatim se gibaju u suprotnom smjeru kako bi se pokušalo stabilizirati njihalo, neuspješno, pomiče se lagano unatrag, kako bi se osigurala rotacija njihala, te se nastavlja kontinuirano gibati, što rezultira uspješnom stabilizacijom njihala u gornjem nestabilnom ravnotežnom položaju. Kada bi kolica došla do kraja staze njihalo bi palo natrag u donji stabilni ravnotežni položaj. Kako bi se stabilizacija održala nastavlja se kontinuirano gibati, a zatim se počinje kretati unatrag.

Zanimljivo je da je sličan način stabilizacije primjenjen i u pokušaju dva, što pokazuje da različite mreže mogu uspješno doći do iste metode.

Kao što se moglo primjetiti, mreže različitih topologija i kompleksnosti sposobne su generirati ista ili slična rješenja na vrlo inteligentan način.

7 ZAKLJUČAK

Tema ovog diplomskog rada bila je Primjena evolucijskog algoritma za učenje neuronske mreže.

Opisane su osnove neuronskih mreža i njihove karakteristike kao što su čvorovi i slojevi, te vrste neuronskih mreža.

Također opisani su evolucijski algoritmi, inspiracija, struktura algoritma i potrebni koraci.

Dan je opis i pregled područja NeuroEvolucije s naglaskom na Neuronske mreže evoluirajućih topologija i težina (eng. *Topology and Weight Evolving Artificial Neural Networks - TWEANN*) te su postavljena neka vrlo bitna pitanja koje će NEAT algoritam pokušati riješiti.

Proučen je teorijski koncept NeuroEvolucije promjenjivih topologija (eng. *NeuroEvolution of Augmenting Topologies - NEAT*).

NEAT algoritam je implementiran u programskom okruženju MATLAB te je implementacija ispitana na dva problema: XOR problem i problem stabilizacije inverznog njihala.

LITERATURA

- [1] Majetić D., Predavanja iz kolegija Neuronske mreže, FSB, 2017.
- [2] Čurković, Petar. Optimization of Generatively Encoded Multi-Material Lattice Structures for Desired Deformation Behavior // *Symmetry*, 13 (2021), 2; 293, 14 doi:10.3390/sym13020293
- [3] Čurković, Petar; Čehulić, Lovro .Diversity maintenance for efficient robot path planning // *Applied Sciences-Basel*, 10 (2020), 5; 10051721, 15 doi:10.3390/app10051721
- [4] Čurković, Petar; Jerbić, Bojan; Stipančić, Tomislav. Hybridization of adaptive genetic algorithm and ART 1 neural architecture for efficient path planning of a mobile robot // *Transactions of FAMENA*, 32 (2008), 2; 11-21
- [5] Čurković P., Predavanja iz kolegija Umjetna inteligencija, FSB, 2017.
- [6] Stanley K.O., Miikkulainen R. (2002), *Evolving Neural Networks through Augmenting Topologies*, *Evolutionary Computation*. 10 (2): 99–127.
- [7] Gaier A., Asteroth A., Mouret J.-B. (2018), *Data-efficient neuroevolution with kernel-based surrogate models*, GECCO '18: Proceedings of the Genetic and Evolutionary Computation Conference, July 2018, Pages 85–92, <https://doi.org/10.1145/3205455.3205510>
- [8] Anderson, C. W. (1989). Learning to control an inverted pendulum using neural networks. *IEEE Control Systems Magazine*, 9:31–37.
- [9] Angeline, P. J., Saunders, G. M., and Pollack, J. B. (1993). An evolutionary algorithm that constructs recurrent neural networks. *IEEE Transactions on Neural Networks*, 5:54–65.
- [10] Branke, J. (1995). Evolutionary algorithms for neural network design and training. In Alander, J. T., editor, *Proceedings First Nordic Workshop on Genetic Algorithms and their Applications*, pages 145–163, University of Vaasa Press, Vaasa, Finland.
- [11] Braun, H. and Weisbrod, J. (1993). Evolving feedforward neural networks. In Albrecht, R. F., Reeves, C. R., and Steele, N. C., editors, *Proceedings of ANNGA93, International Conference on Artificial Neural Networks and Genetic Algorithms*, pages 25–32, Springer-Verlag, Innsbruck.
- [12] Chen, D. et al. (1993). Constructive learning of recurrent neural networks. In Petsche, T., Judd, S., and Hanson, S., editors, *Computational Learning Theory and Natural Learning Systems III*. MIT Press, Cambridge, Massachusetts.
- [13] Cybenko, G. (1989). Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals, and Systems*, 2(4):303–314.
- [14] Darnell, J. E. and Doolittle, W. F. (1986). Speculations on the early course of evolution. *Proceedings of the National Academy of Sciences, USA*, 83:1271–1275.
- [15] Darwen, P. J. (1996). *Co-Evolutionary Learning by Automatic Modularisation with Speciation*. Ph.D. thesis, School of Computer Science, University College, University of New South Wales, Sydney, Australia.

- [16] Darwen, P. and Yao, X. (1996). Automatic modularization by speciation. In *Proceedings of the 1996 IEEE International Conference on Evolutionary Computation (ICEC '96)*, pages 88–93, Nagoya, IEEE Press, Piscataway, New Jersey.
- [17] Dasgupta, D. and McGregor, D. (1992). Designing application-specific neural networks using the structured genetic algorithm. In Whitley, D. and Schaffer, J. D., editors, *Proceedings of the International Conference on Combinations of Genetic Algorithms and Neural Networks*, pages 87–96, IEEE Press, Piscataway, New Jersey.
- [18] Fullmer, B. and Miikkulainen, R. (1992). Using marker-based genetic encoding of neural networks to evolve finite-state behaviour. In Varela, F. J. and Bourguine, P., editors, *Toward a Practice of Autonomous Systems: Proceedings of the First European Conference on Artificial Life*, pages 255–262, MIT Press, Cambridge, Massachusetts.
- [19] Goldberg, D. E. (1989). *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley, Reading, Massachusetts.
- [20] Goldberg, D. E. and Richardson, J. (1987). Genetic algorithms with sharing for multimodal function optimization. In Grefenstette, J. J., editor, *Proceedings of the Second International Conference on Genetic Algorithms*, pages 148–154, Morgan Kaufmann, San Francisco, California.
- [21] Gomez, F. and Miikkulainen, R. (1997). Incremental evolution of complex general behavior. *Adaptive Behavior*, 5:317–342.
- [22] Gomez, F. and Miikkulainen, R. (1999). Solving non-Markovian control tasks with neuroevolution. In Dean, T., editor, *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence*, pages 1356–1361, Morgan Kaufmann, San Francisco, California.
- [23] Gomez, F. and Miikkulainen, R. (2002). Learning robust nonlinear control with neuroevolution. Technical Report AI02-292, Department of Computer Sciences, The University of Texas at Austin, Austin, Texas.
- [24] Gruau, F. (1993). Genetic synthesis of modular neural networks. In Forrest, S., editor, *Proceedings of the Fifth International Conference on Genetic Algorithms*, pages 318–325, Morgan Kaufmann, San Francisco, California.
- [25] Gruau, F., Whitley, D., and Pyeatt, L. (1996). A comparison between cellular encoding and direct encoding for genetic neural networks. In Koza, J. R. et al., editors, *Genetic Programming 1996: Proceedings of the First Annual Conference*, pages 81–89, MIT Press, Cambridge, Massachusetts.
- [26] Harvey, I. (1993). *The Artificial Evolution of Adaptive Behavior* Ph.D. thesis, School of Cognitive and Computing Sciences, University of Sussex, Sussex, UK. Holland, J. H. (1975). *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence*. University of Michigan Press, Ann Arbor, Michigan.
- [27] Kaelbling, L. P., Littman, M., and Moore, A.W. (1996). Reinforcement learning: A survey. *Journal of Artificial Intelligence*, 4:237–285.
- [28] Krishnan, R. and Ciesielski, V. B. (1994). Delta-gann: A new approach to training neural networks using genetic algorithms. In Tsoi, A. C. and Downs, T., editors, *Proceedings of the Australian Conference on Neural Networks*, pages 194–197, University of Queensland, Brisbane, Australia.

- [29] Lee, C.-H. and Kim, J.-H. (1996). Evolutionary ordered neural network with a linked-list encoding scheme. In *Proceedings of the 1996 IEEE International Conference on Evolutionary Computation*, pages 665–669, IEEE Press, Piscataway, New Jersey.
- [30] Mahfoud, S.W. (1995). *Niching Methods for Genetic Algorithms* Ph.D. thesis, Department of General Engineering, IlliGAL Report 95001, University of Illinois at Urbana-Champaign, Urbana, Illinois.
- [31] Mandischer, M. (1993). Representation and evolution of neural networks. In Albrecht, R. F.,
- [32] Reeves, C. R., and Steele, U. C., editors, *Artificial Neural Nets and Genetic Algorithms*, pages 643–649, Springer Verlag, New York, New York.
- [33] Maniezzo, V. (1994). Genetic evolution of the topology and weight distribution of neural networks. *IEEE Transactions on Neural Networks*, 5(1):39–53.
- [34] Montana, D. J. and Davis, L. (1989). Training feedforward neural networks using genetic algorithms. In Sridharan, S., editor, *Proceedings of the Eleventh International Joint Conference on Artificial Intelligence*, pages 762–767, Morgan Kaufmann, San Francisco, California.
- [35] Moriarty, D. E. (1997). *Symbiotic Evolution of Neural Networks in Sequential Decision Tasks*. Ph.D. thesis, Department of Computer Sciences, The University of Texas at Austin. Technical Report UT-AI97-257.
- [36] Moriarty, D. E. and Miikkulainen, R. (1996). Efficient reinforcement learning through symbiotic evolution. *Machine Learning*, 22:11–32.
- [37] Moriarty, D. E. and Miikkulainen, R. (1997). Forming neural networks through efficient and adaptive co-evolution *Evolutionary Computation*, 5(4):373–399.
- [38] Opitz, D. W. and Shavlik, J. W. (1997). Connectionist theory refinement: Genetically searching the space of network topologies. *Journal of Artificial Intelligence Research*, 6:177–209.
- [39] Pendrith, M. (1994). On reinforcement learning of control actions in noisy and non-Markovian domains. Technical Report UNSW-CSE-TR-9410, School of Computer Science and Engineering, The University of New South Wales, Sydney, Australia.
- [40] Potter, M. A. and De Jong, K. A. (1995). Evolving neural networks with collaborative species. In Oren, T. I. and Birta, L. G., editors, *Proceedings of the 1995 Summer Computer Simulation Conference*, pages 340–345, Society for Computer Simulation, San Diego, California..
- [41] Potter, M. A., De Jong, K. A., and Grefenstette, J. J. (1995). A coevolutionary approach to learning sequential decision rules. In Eshelman, L. J., editor, *Proceedings of the Sixth International Conference on Genetic Algorithms*, pages 366–372, Morgan Kaufmann, San Francisco, California.
- [42] Pujol, J. C. F. and Poli, R. (1998). Evolving the topology and the weights of neural networks using a dual representation. *Special Issue on Evolutionary Learning of the Applied Intelligence Journal*, 8(1):73–84.
- [43] Radcliffe, N. J. (1993). Genetic set recombination and its application to neural network topology optimisation. *Neural Computing and Applications*, 1(1):67–90.
- [44] Rosin, C. D. and Belew, R. K. (1997). New methods for competitive evolution. *Evolutionary Computation*, 5(1):1–29.

- [45] Saravanan, N. and Fogel, D. B. (1995). Evolving neural control systems. *IEEE Expert*, 10(3):23–27.
- [46] Schaffer, J. D., Whitley, D., and Eshelman, L. J. (1992). Combinations of genetic algorithms and neural networks: A survey of the state of the art. In Whitley, D. and Schaffer, J., editors, *Proceedings of the International Workshop on Combinations of Genetic Algorithms and Neural Networks (COGANN-92)*, pages 1–37, IEEE Press, Piscataway, New Jersey.
- [47] Spears, W. (1995). Speciation using tag bits. In *Handbook of Evolutionary Computation*. IOP Publishing Ltd. and Oxford University Press, Oxford, UK.
- [48] Thierens, D. (1996). Non-redundant genetic coding of neural networks. In *Proceedings of the IEEE International Conference on Evolutionary Computation*, pages 571–575, IEEE Press, Piscataway, New Jersey.
- [49] Watson, J. D. et al. (1987). *Molecular Biology of the Gene Fourth Edition*. The Benjamin Cummings Publishing Company, Inc., Menlo Park, California.
- [50] Whitley, D. et al. (1993). Genetic reinforcement learning for neurocontrol problems. *Machine Learning*, 13:259–284. Wieland, A. (1991). Evolving neural network controllers for unstable systems. In *Proceedings of the International Joint Conference on Neural Networks*, pages 667–673, IEEE Press, Piscataway, New Jersey.
- [51] Yao, X. (1999). Evolving artificial neural networks. *Proceedings of the IEEE*, 87(9):1423–1447.
- [52] Yao, X. and Liu, Y. (1996). Towards designing artificial neural networks by evolution. *Applied Mathematics and Computation*, 91(1):83–90.
- [53] Yao, X. and Shi, Y. (1995). A preliminary study on designing artificial neural networks using coevolution. In *Proceedings of the IEEE Singapore International Conference on Intelligent Control and Instrumentation*, pages 149–154, IEEE Singapore Section.
- [54] Zhang, B.-T. and Muhlenbein, H. (1993). Evolving optimal neural networks using genetic algorithms with Occam's razor. *Complex Systems*, 7:199–220.