DeepTactics

Neat i Super Mario

By: Ludvig Øvrevik, Brage Kvamme, Christian Johnsen, Håkon Støren, Vetle Støren, Kristian Carnelius og Kacper Pawlowski

AGENDA

01

Om Neat

02

Implementasjon

03

Resultater

04

Demo og Spørsmål

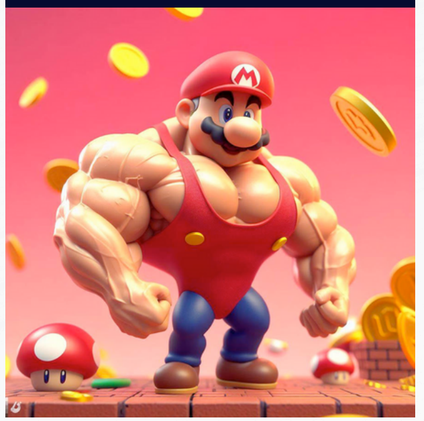
MÅL

Realistiske mål

- Lage en evolusjonær algoritme fra bunn
- Få Mario til å klare første bane

- Få verdensrekord i Super Mario
- Kunne spille alle typer spill, med litt trening

Ambisiøse mål



HVA ER NEAT?

NeuroEvolution of Augmenting Topologies

- Etterligner evolusjonsprosessen
- Survival of the fittest
- Muterer og utvikler seg over flere generasjoner
- Hensikt: Løsning med minimalt nettverk

Fig. 1. **A genotype to phenotype mapping example.** The third gene is disabled, so the connection that it specifies (between nodes 2 and 5) is not expressed in the phenotype.

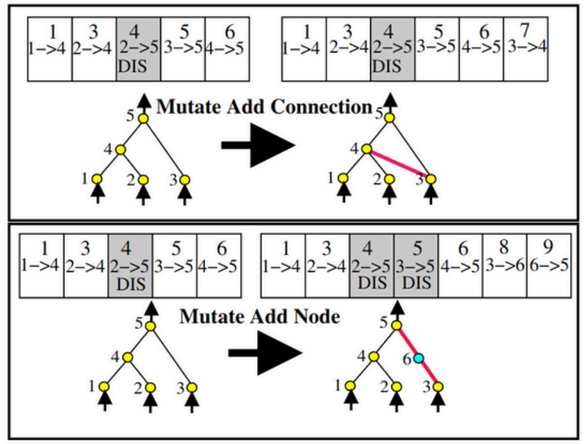
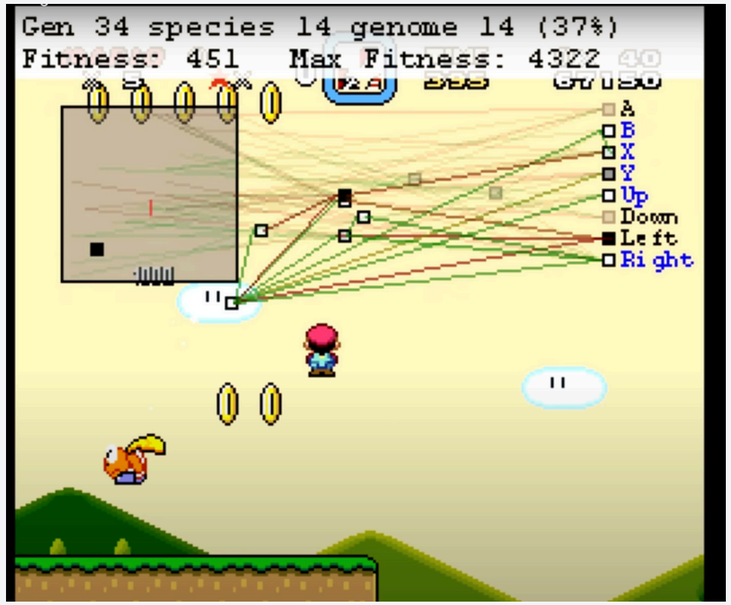
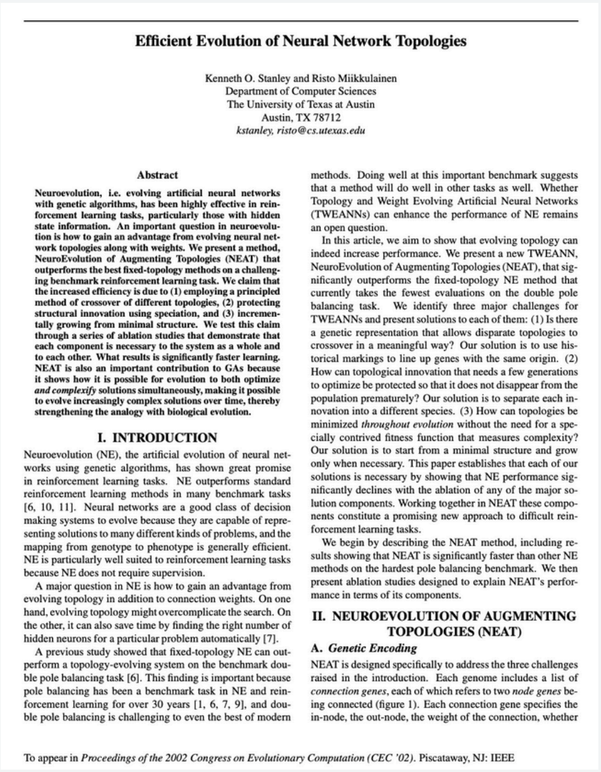


Fig. 2. **The two types of structural mutation in NEAT.** Both types, adding a connection and adding a node, are illustrated with the genes above their phenotypes. The top number in each genome is the *innovation number* of that gene. These numbers identify the original historical ancestor of each gene, making it possible to find matching genes during crossover. New genes are assigned new increasingly higher numbers.

Hva vi tok inspirasjon fra



Sethbling



Efficient Evolution of Neural Network Topologies

Kenneth O. Stanley and Risto Miikkulainen
Department of Computer Sciences
The University of Texas at Austin
Austin, TX 78712
kstanley, risto@cs.utexas.edu

Abstract

Neuroevolution, i.e. evolving artificial neural networks with genetic algorithms, has been highly effective in reinforcement learning tasks, particularly those with hidden state information. An important question in neuroevolution is how to gain an advantage from evolving neural network topologies along with weights. We present a method, NeuroEvolution of Augmenting Topologies (NEAT) that outperforms the best fixed-topology methods on a challenging benchmark reinforcement learning task. We claim that the increased efficiency is due to (1) employing a principled method of crossover of different topologies, (2) protecting structural innovation using speciation, and (3) incrementally growing from minimal structure. We test this claim through a series of ablation studies that demonstrate that each component is necessary to the system as a whole and to each other. What results is significantly faster learning. NEAT is also an important contribution to GAs because it shows how it is possible for evolution to both optimize *and complexify* solutions simultaneously, making it possible to evolve increasingly complex solutions over time, thereby strengthening the analogy with biological evolution.

I. INTRODUCTION

Neuroevolution (NE), the artificial evolution of neural networks using genetic algorithms, has shown great promise in reinforcement learning tasks. NE outperforms standard reinforcement learning methods in many benchmark tasks [6, 10, 11]. Neural networks are a good class of decision making systems to evolve because they are capable of representing solutions to many different kinds of problems, and the mapping from genotype to phenotype is generally efficient. NE is particularly well suited to reinforcement learning tasks because NE does not require supervision.

A major question in NE is how to gain an advantage from evolving topology in addition to connection weights. On one hand, evolving topology might overcomplicate the search. On the other, it can also save time by finding the right number of hidden neurons for a particular problem automatically [7].

A previous study showed that fixed-topology NE can outperform a topology-evolving system on the benchmark double pole balancing task [6]. This finding is important because pole balancing has been a benchmark task in NE and reinforcement learning for over 30 years [1, 6, 7, 9], and double pole balancing is challenging to even the best of modern methods. Doing well at this important benchmark suggests that a method will do well in other tasks as well. Whether Topology and Weight Evolving Artificial Neural Networks (TWEANNs) can enhance the performance of NE remains an open question.

In this article, we aim to show that evolving topology can indeed increase performance. We present a new TWEANN, NeuroEvolution of Augmenting Topologies (NEAT), that significantly outperforms the fixed-topology NE method that currently takes the fewest evaluations on the double pole balancing task. We identify three major challenges for TWEANNs and present solutions to each of them: (1) Is there a genetic representation that allows disparate topologies to crossover in a meaningful way? Our solution is to use historical markings to line up genes with the same origin. (2) How can topological innovation that needs a few generations to optimize be protected so that it does not disappear from the population prematurely? Our solution is to separate each innovation into a different species. (3) How can topologies be minimized *throughout evolution* without the need for a specially contrived fitness function that measures complexity? Our solution is to start from a minimal structure and grow only when necessary. This paper establishes that each of our solutions is necessary by showing that NE performance significantly declines with the ablation of any of the major solution components. Working together in NEAT these components constitute a promising new approach to difficult reinforcement learning tasks.

We begin by describing the NEAT method, including results showing that NEAT is significantly faster than other NE methods on the hardest pole balancing benchmark. We then present ablation studies designed to explain NEAT's performance in terms of its components.

II. NEUROEVOLUTION OF AUGMENTING TOPOLOGIES (NEAT)

A. *Genetic Encoding*

NEAT is designed specifically to address the three challenges raised in the introduction. Each genome includes a list of *connection genes*, each of which refers to two *node genes* being connected (figure 1). Each connection gene specifies the in-node, the out-node, the weight of the connection, whether

To appear in *Proceedings of the 2002 Congress on Evolutionary Computation (CEC '02)*. Piscataway, NJ: IEEE

Originalt paper fra 2002 om NEAT

NEAT Paper

DÅRLIG!

- Var ikke lett å forstå hvordan de selv hadde implementert algoritmen
- Måtte gjøre mange antagelser og finne egne løsninger på problemer

Implementasjon

- Input: Piksel fra spillet - 10x20 grayscale | 20x40 RGB
- Output: Action (Jump, move left, move right...)

- Ikke brukt pytorch eller tensorflow.
- Alt fra genomes, til noder og connections er klasser

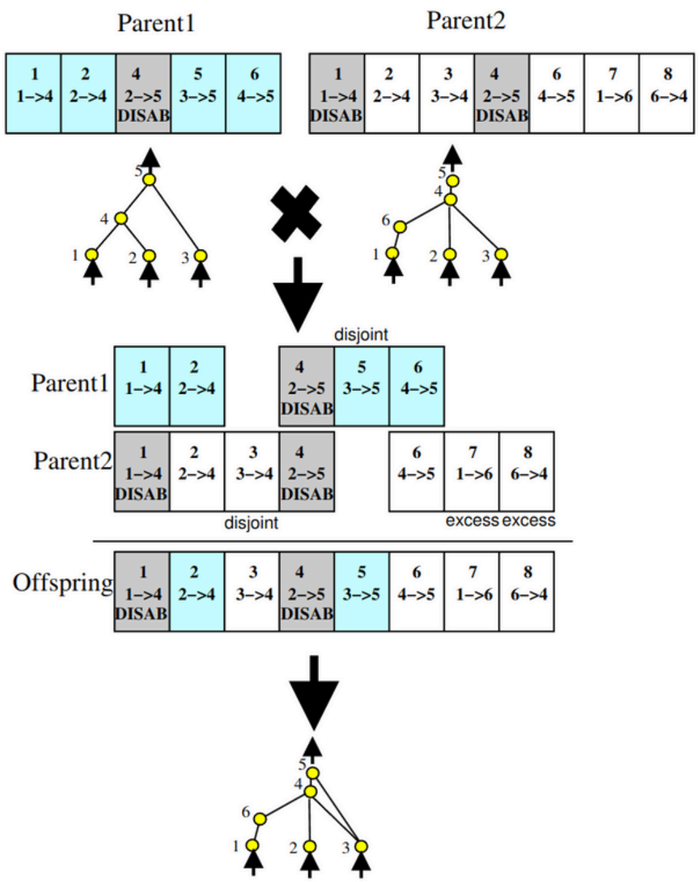
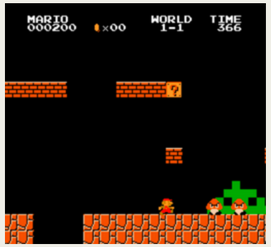
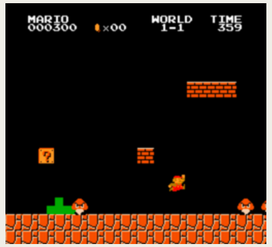


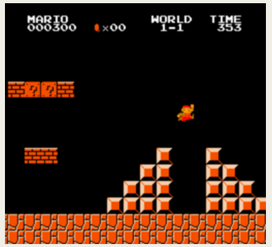
Fig. 3. **Matching up genomes for different network topologies using innovation numbers.** Although Parent 1 and Parent 2 look different, their innovation numbers (shown at the top of each gene) tell us which genes match up with which without the need for topological analysis.

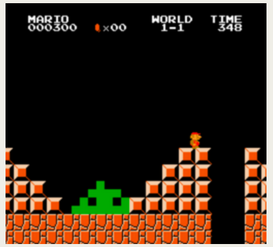
For hver generasjon:

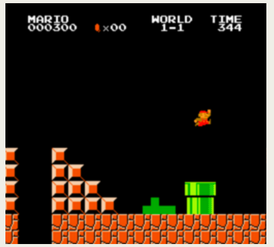
Starter med en populasjon

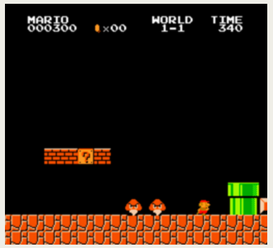


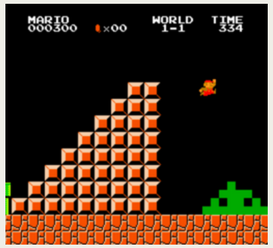


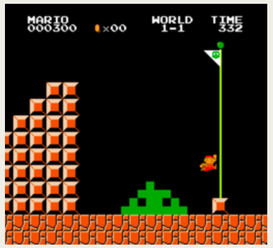


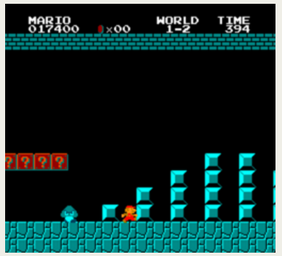


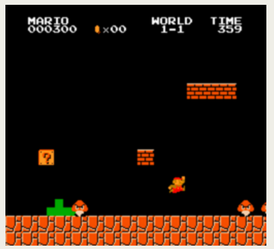




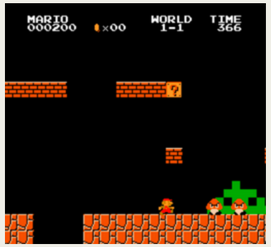




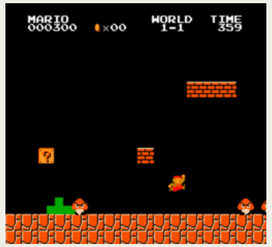




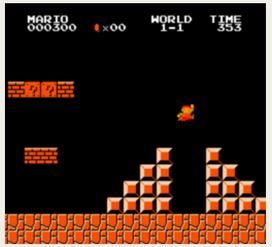
Spill mario og få fitness



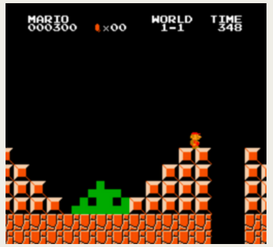
Fitness: 69



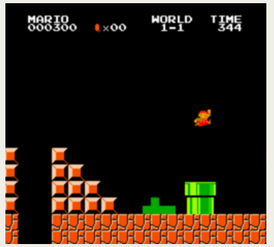
Fitness: 150



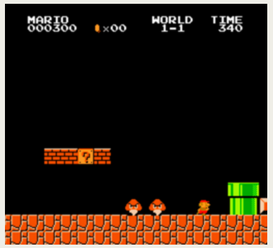
Fitness: 190



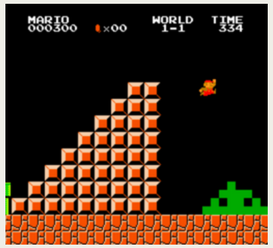
Fitness: 12



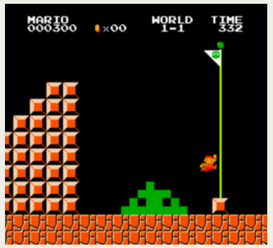
Fitness: 42



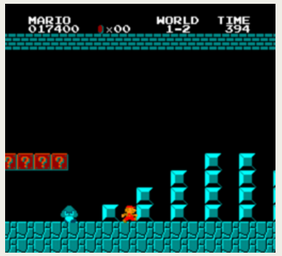
Fitness: 74



Fitness:260



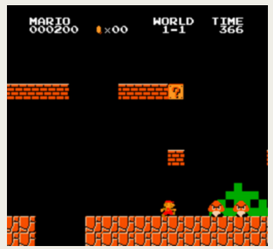
Fitness: 300



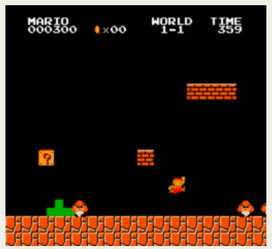
Fitness: 500

Fitness: 95

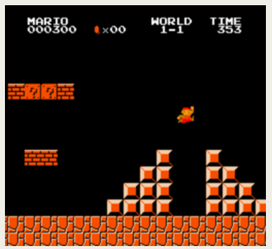
Del inn i arter



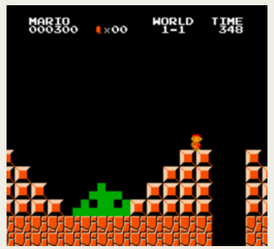
Fitness: 69



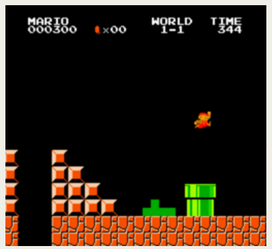
Fitness: 150



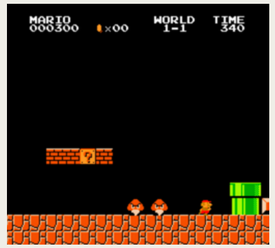
Fitness: 190



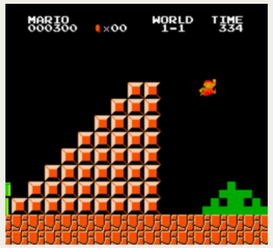
Fitness: 12



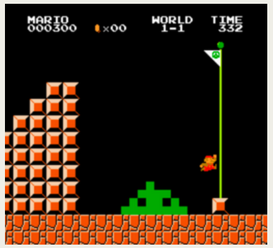
Fitness: 42



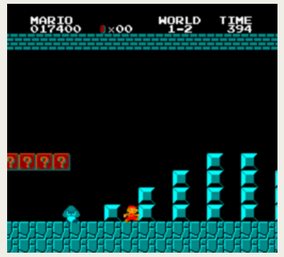
Fitness: 74



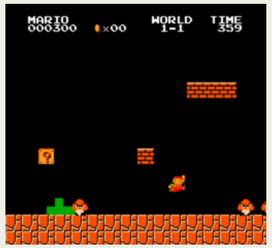
Fitness:260



Fitness: 300



Fitness: 500



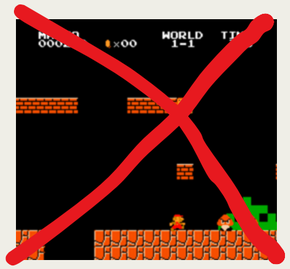
Fitness: 95

Art nummer 1

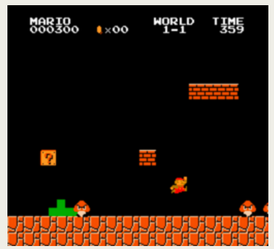
Art nummer 2

Art nummer 3

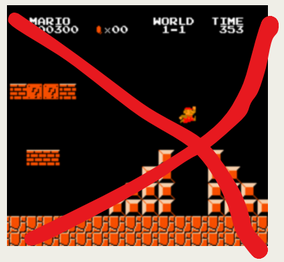
DREP DE DÅRLIGSTE



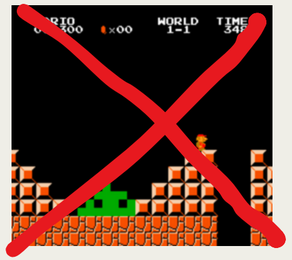
Fitness: 69



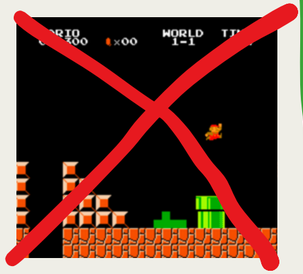
Fitness: 420



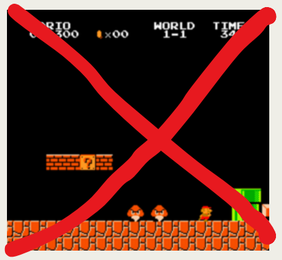
Fitness: 3



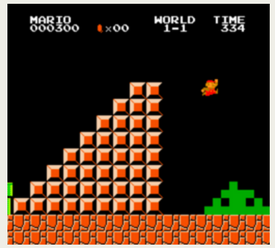
Fitness: 12



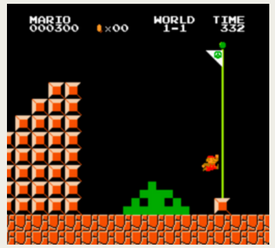
Fitness: 42



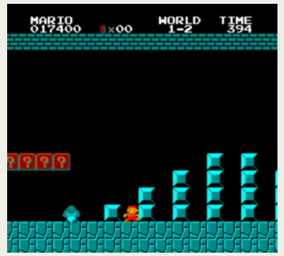
Fitness: 74



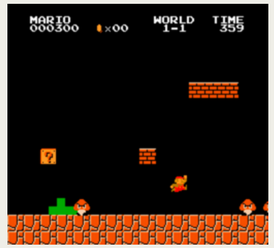
Fitness:79



Fitness: 300



Fitness: 500



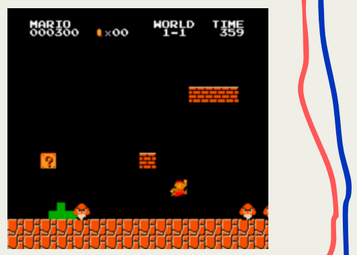
Fitness: 95

Art nummer 1

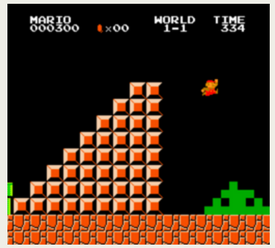
Art nummer 2

Art nummer 3

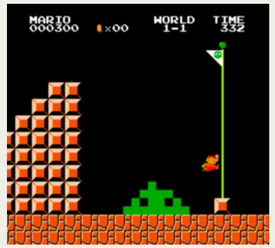
De som overlever parrer seg



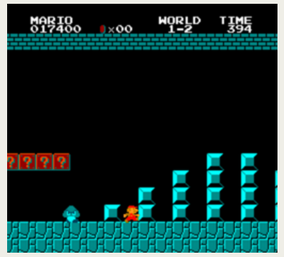
Fitness: 420



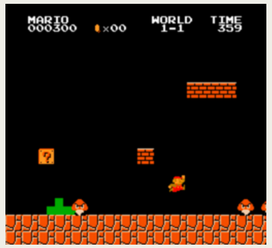
Fitness:79



Fitness: 300



Fitness: 500



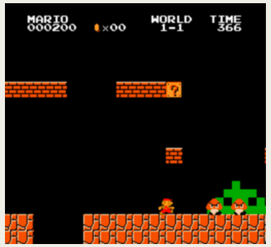
Fitness: 95

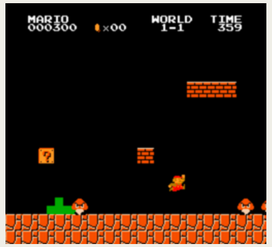
Art nummer 1

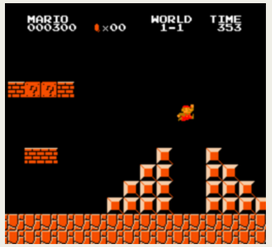
Art nummer 2

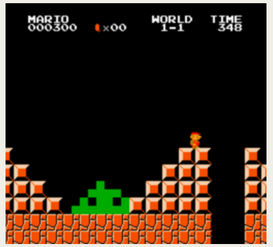
Art nummer 3

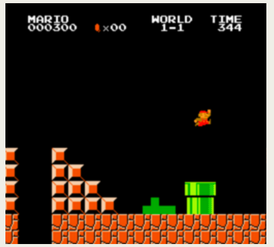
Ny generasjon blir laget

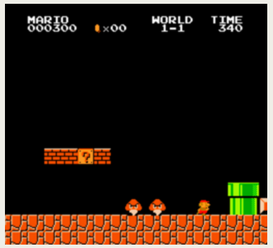


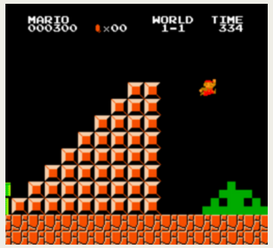


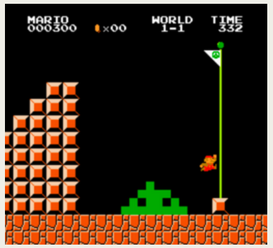


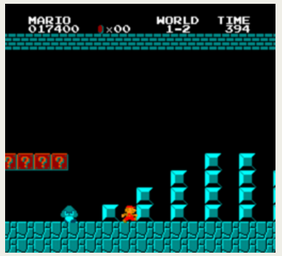




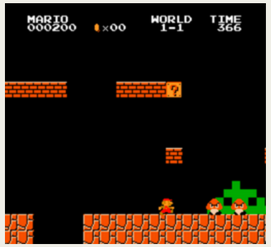


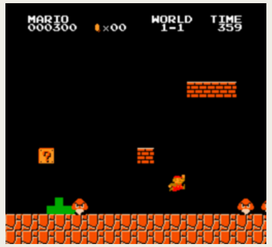


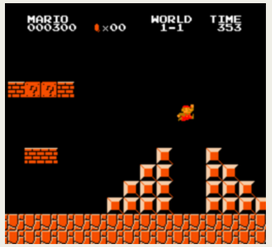


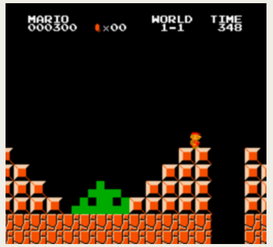


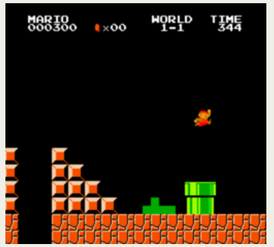
Muter den nye generasjonen

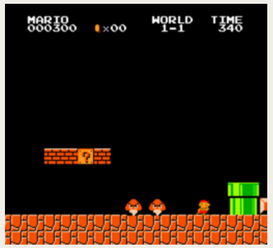


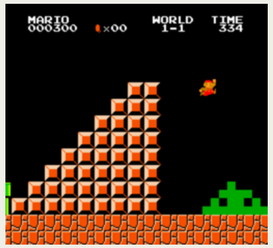


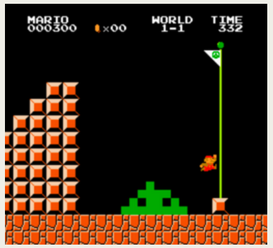


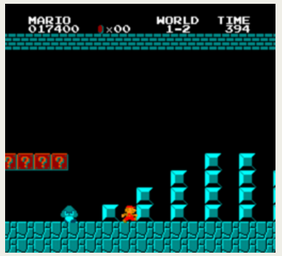




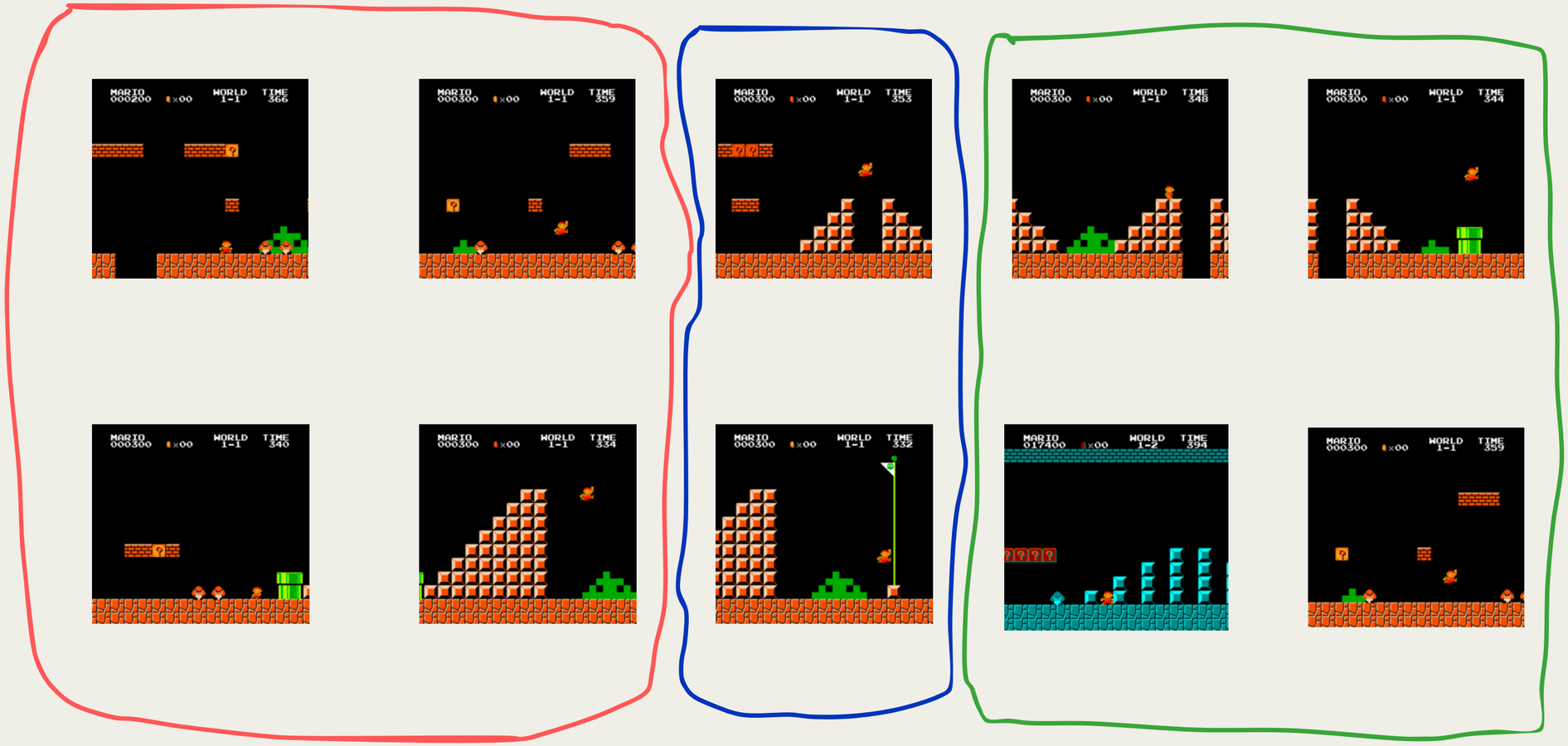








Del inn i nye arter



Art nummer 1

Art nummer 2

Art nummer 3

Formel for å regne genetisk distanse mellom to nettverk

$$\delta = \frac{c_1 E}{N} + \frac{c_2 D}{N} + c_3 \cdot \overline{W}.$$

Formel for å regne ut hvor mange individer hver art skal ha i neste generasjon.

$$N_j' = \frac{\sum_{i=1}^{N_j} f_{ij}}{\overline{f}},$$

Filosofiske og etiske avveiinger

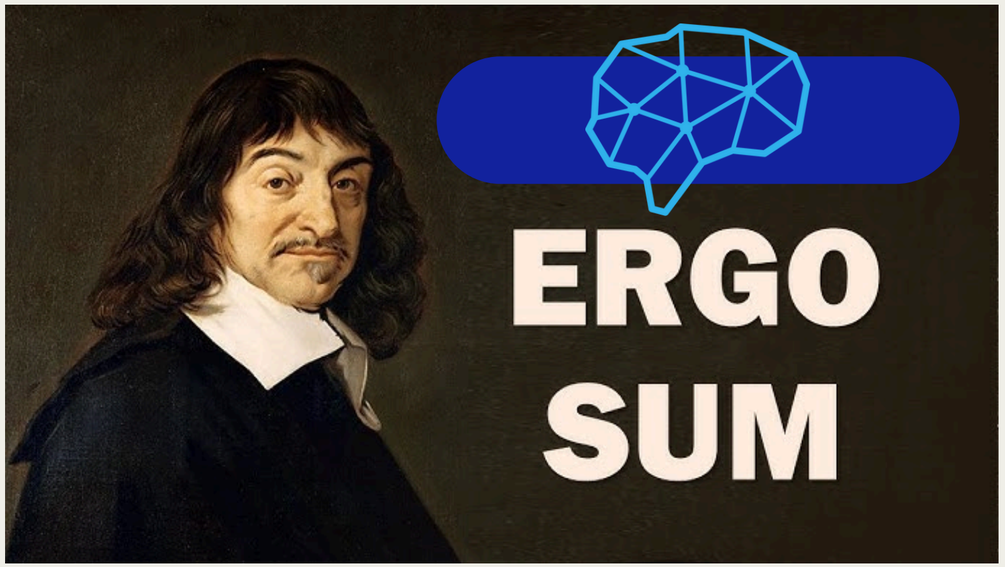
Sosialdarwinisme

Dominant og undertrykket nettverk

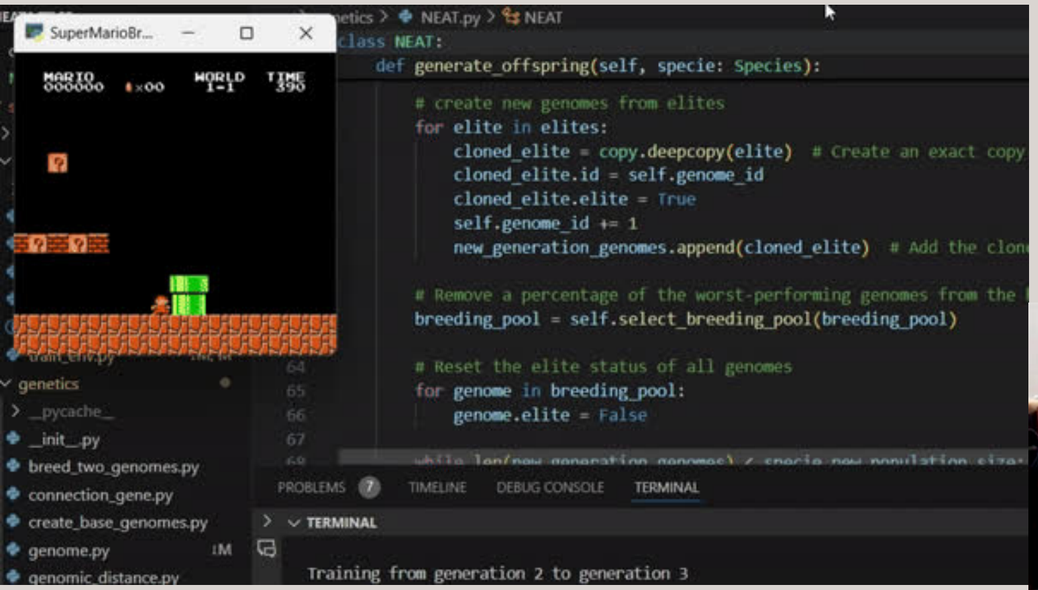
Dele inn i arter og segregere dem

Drep svakeste barn

Genetisk determinisme

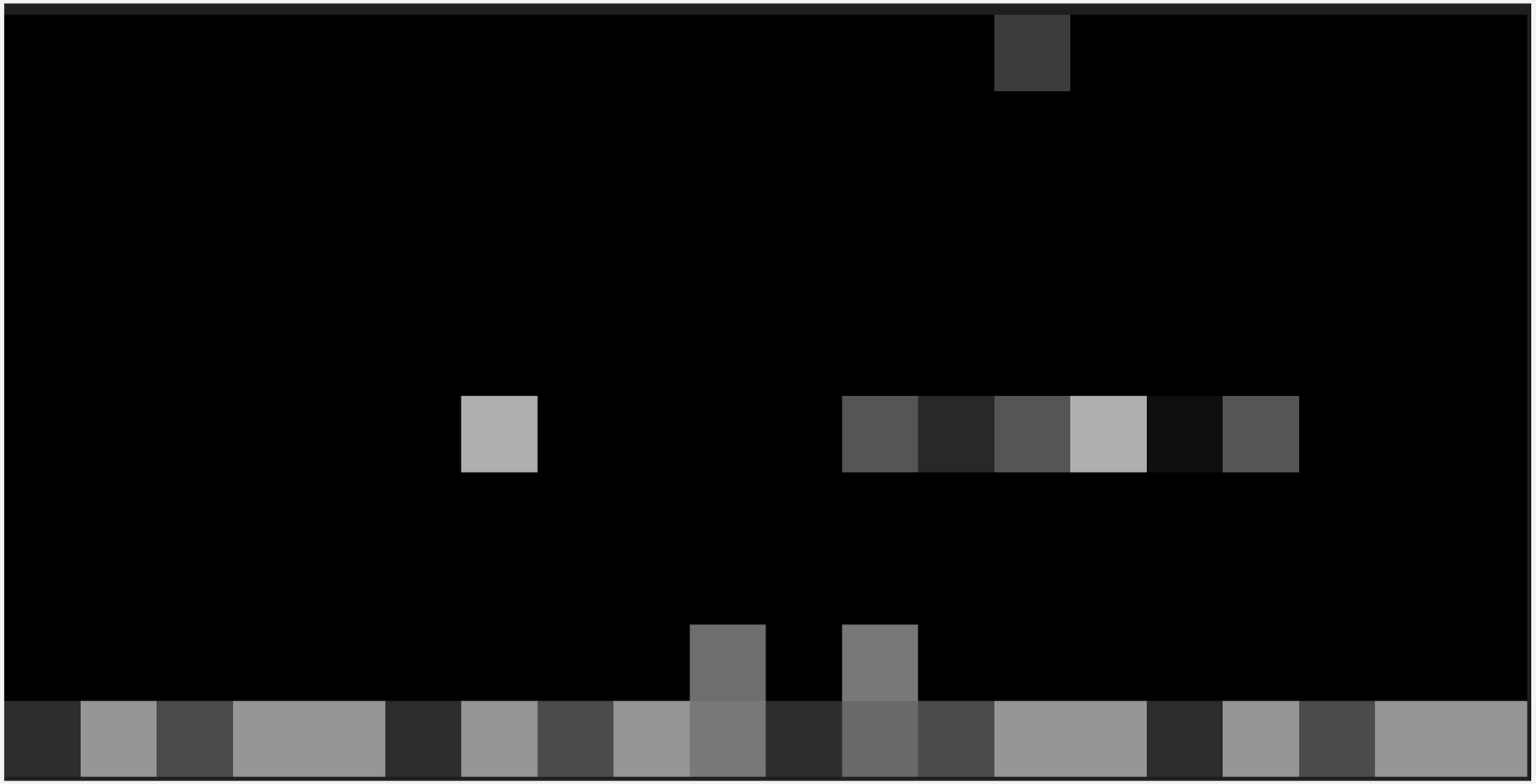


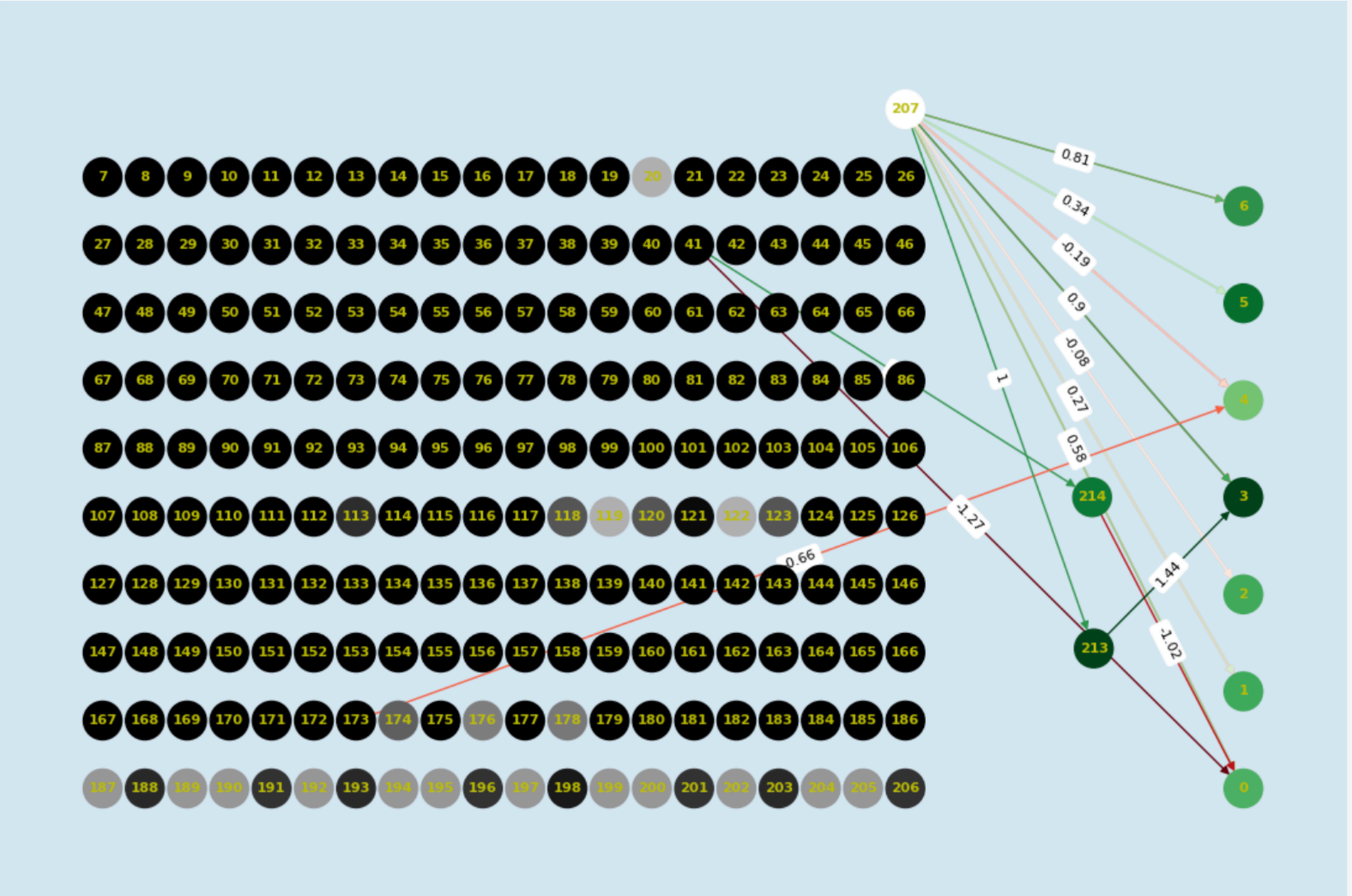
Så var det Trening

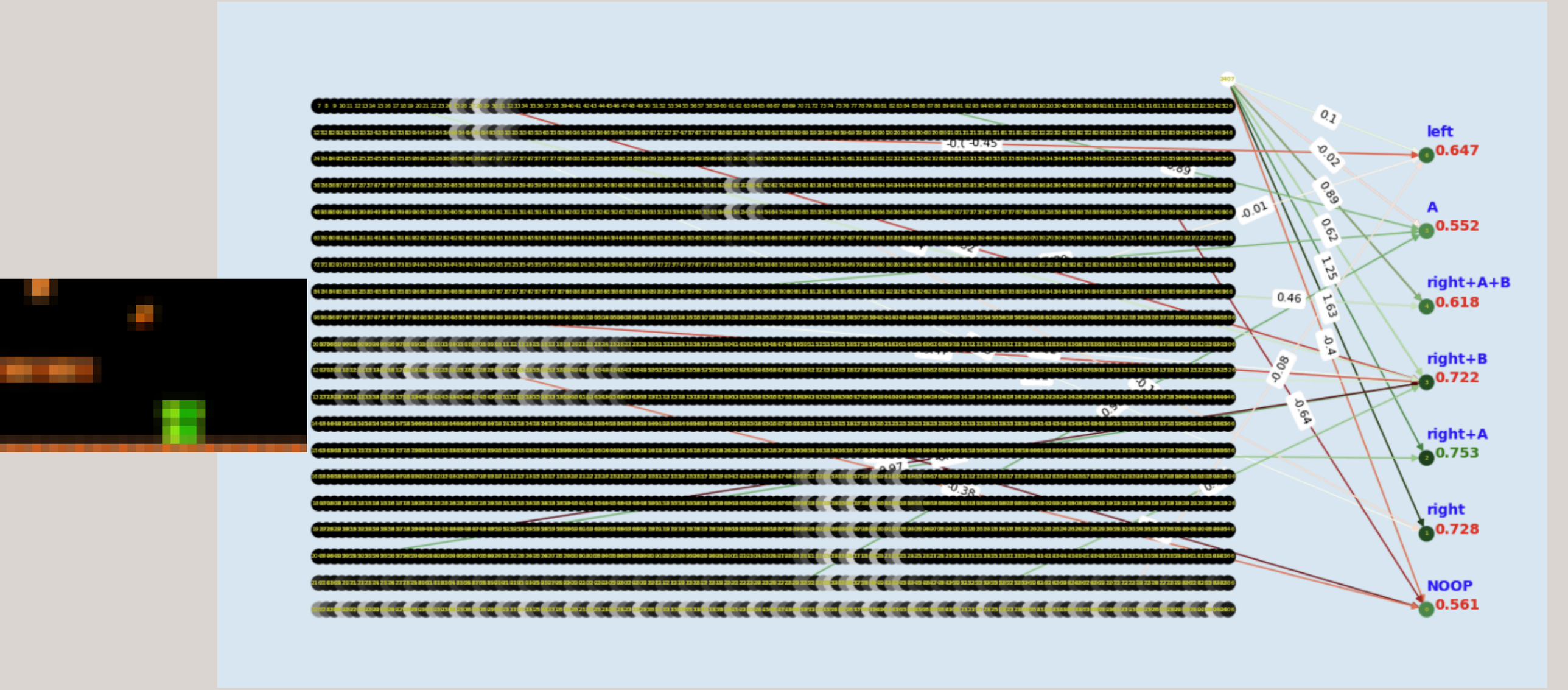



```
20879714      GPUQ cost_eff jcrogers PD       0:00      1 (Priority)
20879712      GPUQ cost_eff jcrogers PD       0:00      1 (Priority)
20879710      GPUQ cost_eff jcrogers PD       0:00      1 (Priority)
20879708      GPUQ cost_eff jcrogers PD       0:00      1 (Priority)
20879706      GPUQ cost_eff jcrogers PD       0:00      1 (Priority)
20879704      GPUQ cost_eff jcrogers PD       0:00      1 (Priority)
20879702      GPUQ cost_eff jcrogers PD       0:00      1 (Priority)
20879700      GPUQ cost_eff jcrogers PD       0:00      1 (Priority)
20879698      GPUQ cost_eff jcrogers PD       0:00      1 (Priority)
20879756      GPUQ carbon_e jcrogers PD       0:00      1 (Priority)
20879754      GPUQ carbon_e jcrogers PD       0:00      1 (Priority)
20879752      GPUQ carbon_e jcrogers PD       0:00      1 (Priority)
20879750      GPUQ carbon_e jcrogers PD       0:00      1 (Priority)
20879748      GPUQ carbon_e jcrogers PD       0:00      1 (Priority)
20879746      GPUQ carbon_e jcrogers PD       0:00      1 (Priority)
20879864      GPUQ carbon_e jcrogers PD       0:00      1 (Priority)
20879862      GPUQ carbon_e jcrogers PD       0:00      1 (Priority)
20879860      GPUQ carbon_e jcrogers PD       0:00      1 (Priority)
20879858      GPUQ carbon_e jcrogers PD       0:00      1 (Priority)
20879856      GPUQ carbon_e jcrogers PD       0:00      1 (Priority)
20879854      GPUQ carbon_e jcrogers PD       0:00      1 (Priority)
20879852      GPUQ carbon_e jcrogers PD       0:00      1 (Priority)
20879850      GPUQ carbon_e jcrogers PD       0:00      1 (Priority)
20879848      GPUQ carbon_e jcrogers PD       0:00      1 (Priority)
20879846      GPUQ carbon_e jcrogers PD       0:00      1 (Priority)
20879844      GPUQ carbon_e jcrogers PD       0:00      1 (Priority)
20879842      GPUQ carbon_e jcrogers PD       0:00      1 (Priority)
20879816      GPUQ carbon_e jcrogers PD       0:00      1 (Priority)
20879814      GPUQ carbon_e jcrogers PD       0:00      1 (Priority)
20879812      GPUQ carbon_e jcrogers PD       0:00      1 (Priority)
20879810      GPUQ carbon_e jcrogers PD       0:00      1 (Priority)
20879808      GPUQ carbon_e jcrogers PD       0:00      1 (Priority)
20879806      GPUQ carbon_e jcrogers PD       0:00      1 (Priority)
20879804      GPUQ carbon_e jcrogers PD       0:00      1 (Priority)
20879802      GPUQ carbon_e jcrogers PD       0:00      1 (Priority)
20879800      GPUQ carbon_e jcrogers PD       0:00      1 (Priority)
20879798      GPUQ carbon_e jcrogers PD       0:00      1 (Priority)
20879796      GPUQ carbon_e jcrogers PD       0:00      1 (Priority)
20879794      GPUQ carbon_e jcrogers PD       0:00      1 (Priority)
20879768      GPUQ carbon_e jcrogers PD       0:00      1 (Priority)
20879766      GPUQ carbon_e jcrogers PD       0:00      1 (Priority)
20879764      GPUQ carbon_e jcrogers PD       0:00      1 (Priority)
20879762      GPUQ carbon_e jcrogers PD       0:00      1 (Priority)
20879760      GPUQ carbon_e jcrogers PD       0:00      1 (Priority)
20879758      GPUQ carbon_e jcrogers PD       0:00      1 (Priority)
20879912      GPUQ carbon_e jcrogers PD       0:00      1 (Priority)
20879910      GPUQ carbon_e jcrogers PD       0:00      1 (Priority)
20879908      GPUQ carbon_e jcrogers PD       0:00      1 (Priority)
20879906      GPUQ carbon_e jcrogers PD       0:00      1 (Priority)
20879904      GPUQ carbon_e jcrogers PD       0:00      1 (Priority)
20879902      GPUQ carbon_e jcrogers PD       0:00      1 (Priority)
20879900      GPUQ carbon_e jcrogers PD       0:00      1 (Priority)
20879898      GPUQ carbon_e jcrogers PD       0:00      1 (Priority)
20879896      GPUQ carbon_e jcrogers PD       0:00      1 (Priority)
20879894      GPUQ carbon_e jcrogers PD       0:00      1 (Priority)
20879892      GPUQ carbon_e jcrogers PD       0:00      1 (Priority)
```

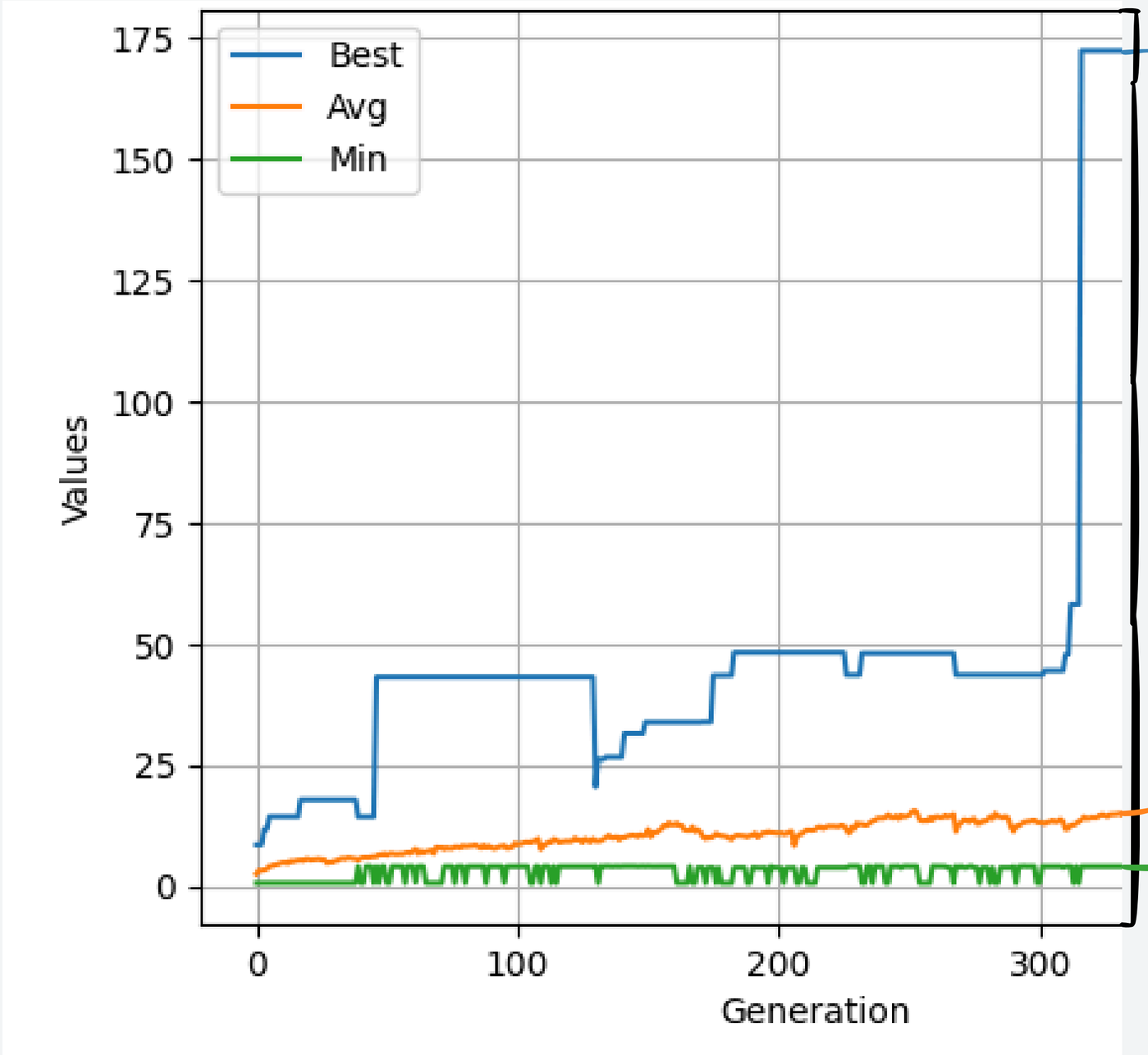
Hva nettverket ser:




7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26
27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46
47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66
67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86
87 88 89 90 91 92 93 94 95 96 97 98 99 100 101 102 103 104 105 106
107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126
127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144 145 146
147 148 149 150 151 152 153 154 155 156 157 158 159 160 161 162 163 164 165 166
167 168 169 170 171 172 173 174 175 176 177 178 179 180 181 182 183 184 185 186
187 188 189 190 191 192 193 194 195 196 197 198 199 200 201 202 203 204 205 206
207
0.81
0.34
-0.19
0.9
-0.08
0.27
0.58
1
-1.27
0.66
214
213
1.44
-1.02
6
5
4
3
2
1
0


left
0.647
A
0.552
right+A+B
0.618
right+B
0.722
right+A
0.753
right
0.728
NOOP
0.561
0.1
-0.02
0.89
0.62
1.25
1.63
-0.4
-0.64
0.46
-0.08
-0.01
-0.45
-0.38

Fitness


Best
Avg
Min
175
150
125
100
75
50
25
0
Values
0
100
200
300
Generation

Problemer og ting vi kunne gjort annerledes


NEAT klarer
ikke å vinne
Deep
tactics


Mer input


Cutting
edge
algoritmer
Super-
computer
IDUN
gym
environment


Klare bane
1 på super
mario
NEAT


Klare bane
1 på super
mario
Den siste
trappen
NEAT


Når du er en KI som blir bedt om å
runne super mario, og blir kjeftet på
Men du gjør bare ditt beste ut av det du har fått


NEAT
SOM
SPILLER
SUPER MARIO
10*20
PIXLER,
GRAYSCALE,
DISCO

Carlenius Consulting

Org nr **931 467 220**

Telefon **954 83 129** Adresse **Huldreveien 19, 1388 Borgen**

Sammenlign Overvåk **Finn lignende** **Del profil**

Oversikt Regnskap Nøkkeltall Roller og Eiere Organisasjon Kunngjøringer

Kontaktopplysninger

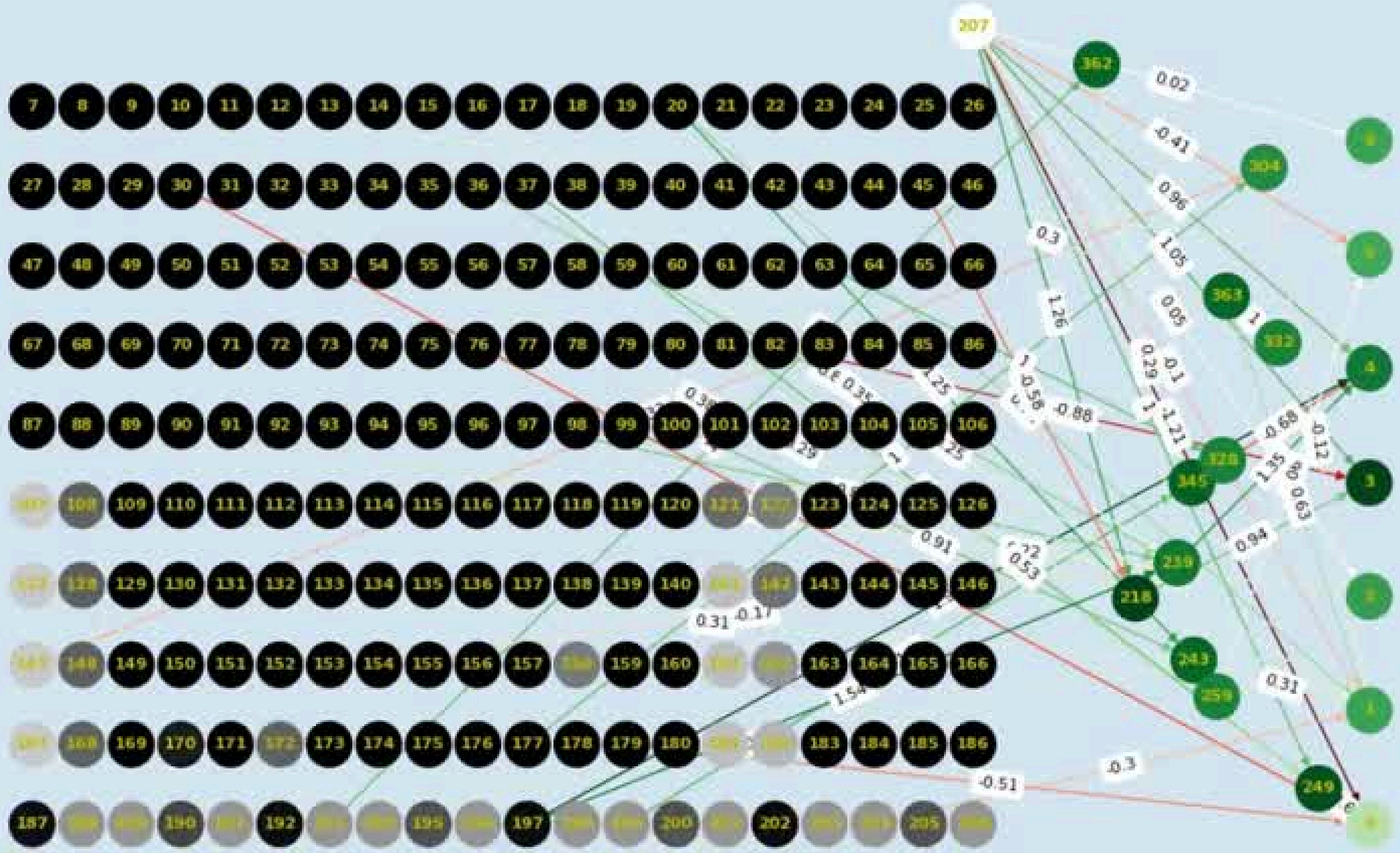
Telefon	954 83 129	Adresse	Huldreveien 19, 1388 Borgen
		Postadresse	Huldreveien 19, 1388 Borgen

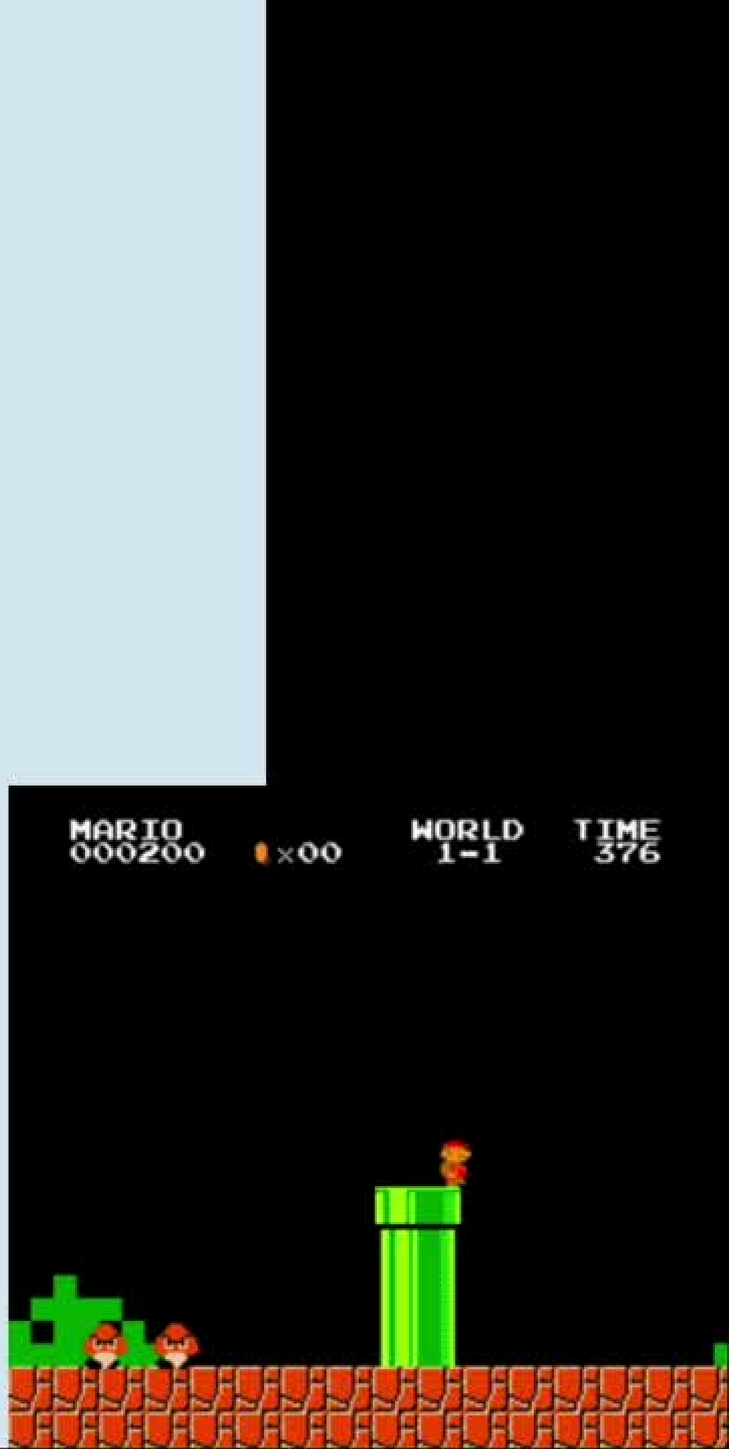
Offisiell foretaksinformasjon

Juridisk navn	CARLENIUS CONSULTING	Adresse	c/o Signy Gautefall Huldreveien 19, 1388 Borgen
Org nr	931 467 220	Postadresse	c/o Signy Gautefall Huldreveien 19, 1388 Borgen
Registreringsdato	30.05.2023	Registrert i	⊗ MVA
Selskapsform	Enkeltpersonforetak		⊗ Foretaksregisteret
NACE-bransje	62.010 Programmeringstjenester		⊗ NAV aa-registeret
Innehaver	Kristian Gautefall Carlenius		⊗ Frivillighetsreg

Kilde: Brønnøysundregistrene

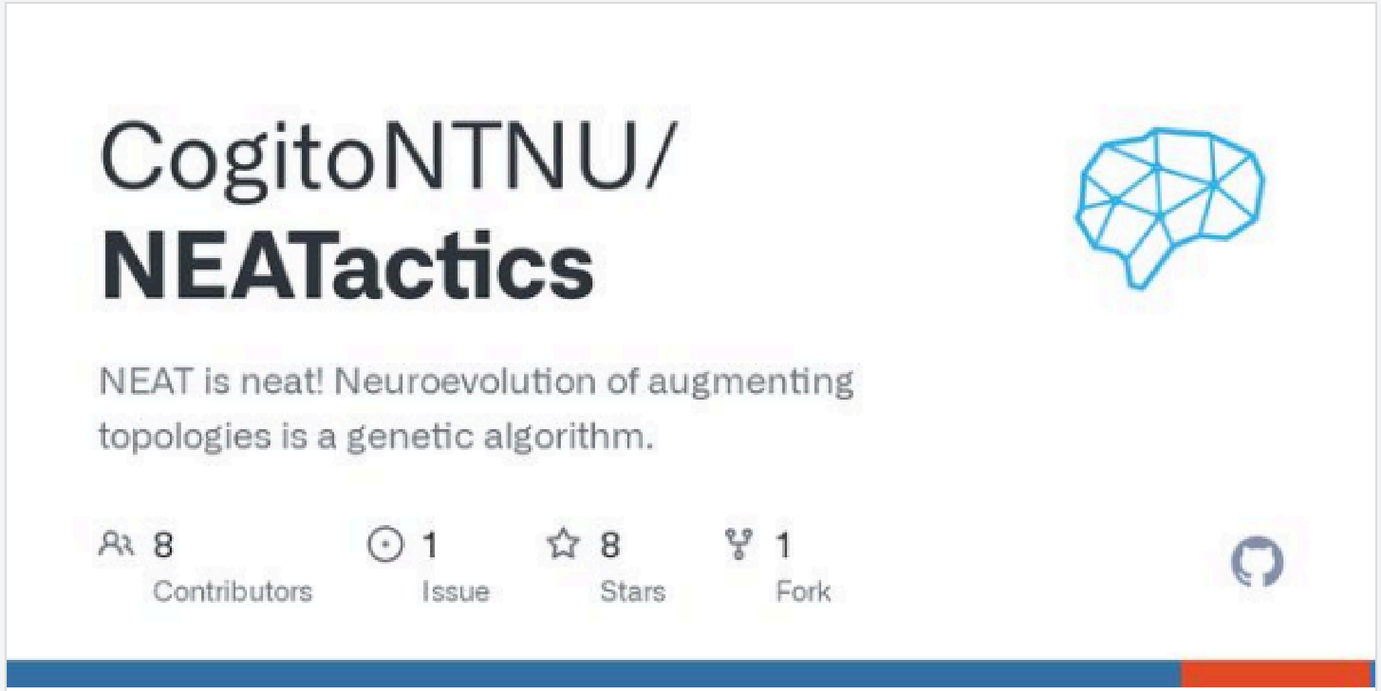
EXAMPLE NETWORK





Demo

Spørsmål?


CogitoNTNU/
NEATactics
NEAT is neat! Neuroevolution of augmenting topologies is a genetic algorithm.
8
Contributors
1
Issue
8
Stars
1
Fork
CogitoNTNU/NEATactics: NEAT is neat! Neuroevolution of augmenting topologies is a genetic algorithm.
NEAT is neat! Neuroevolution of augmenting topologies is a genetic algorithm. - CogitoNTNU/NEATactics
GitHub