Project 3 IT3708:

Evolving Neural Networks for a Flatland Agent

a) EA choices

Scenario	static
Scenario set size	1
Population size	800
Adult selection	Over-production
Number of children	1600
Parent selection	Tournament selection
e	0.66
K	10
Hidden layers	0
Neurons per hidden layer	0
Weight accuracy	8
Direct copies (elitism)	1
Crossover ratio	0.58
Mutation ratio	0.6

With some trial and error, the selection methods where set for the static scenario with 5 different maps. Then the parameters for first K and e in tournament selection where optimized. The population size and number of children where increased until the results stopped improving. Last the crossover and mutation ratios where optimized.

The fitness function gives every set of weights 60 timesteps to solve a map. The collected food and poison fields are counted and the fitness is calculated as

$$f=\frac{\sum_{i=0}^{m}\frac{\sum_{c_i}1-\sum_{p_i}2}{n_i}}{m}$$

m is the number of maps in the scenario, n_i the number of food fields in the map i , c_i the number of collected food fields in map i and p_i the number of poison fields collected in map i . So the agent gains a point for every food field he clears, but looses two points for eating poison. His score is then divided by the number of food fields he could have collected on that map. This score is averaged over the maps that he was tested on.

b) Implementation

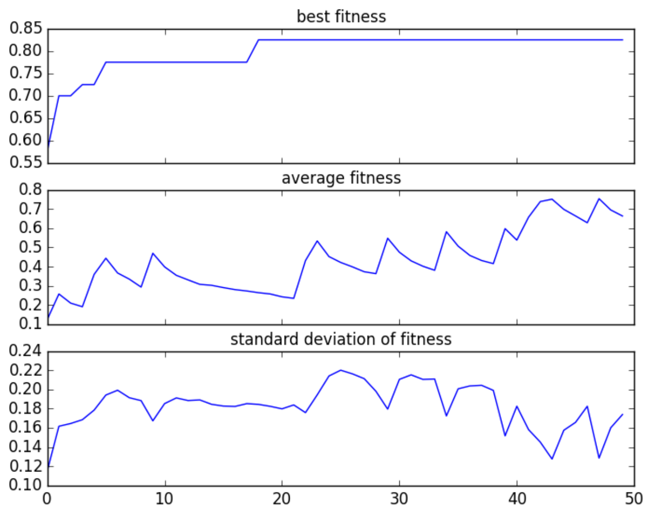
Long trials with different neural network sizes showed that no hidden layer is needed. In fact the results got worse using any number of hidden layers. To prevent the agent from doing nothing one bias neuron is added to the input layer. Since all input nodes are directly connected to the output nodes, no activation function is needed. Since the agend can only walk in one direction per timestep, the direction with the highest output value is chosen. The range of possible weights, that was most successfull was 0 to 256.

One can imagine a neural network that behaves reasonable without any hidden layers as follows. The connections of the food senses to the respective directions are weighted heavy. The connections of the poison senses are heavy weighted to the remaining directions. The bias neuron can for example just make the agent walk forward if there is no food or poison.

c) Performance

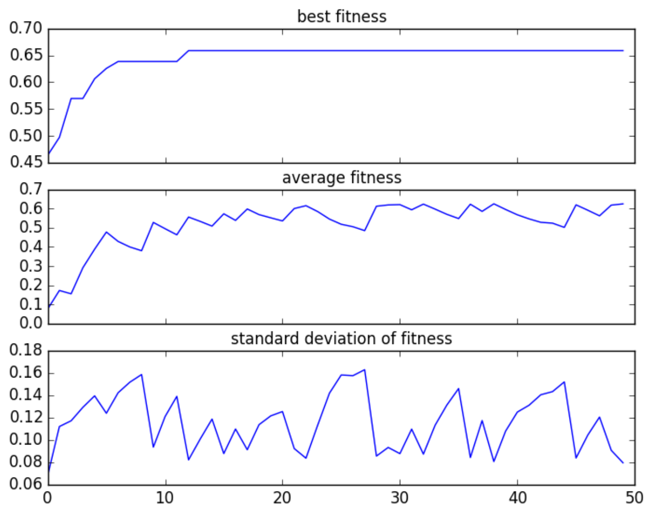
Static – single scenario

The EA created a neural network, that reached a fitness of 0.825 on the one static scenario. The agent eats most food on the map, but turns away from a food field one time. On randomly generated scenarios, the agent sometimes turns away from food fields and seldomly eats poison. He sometimes move in circles.



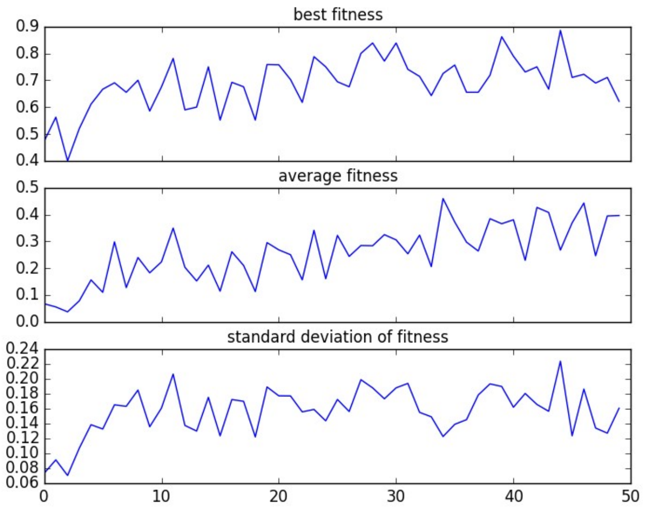
Static – five scenarios

The best fitness was 0.659. The agent behaves reasonable but skips some food fields and eats two posion fields in the five scenarios. On random maps, he behaves likely but tends to walk in circles.



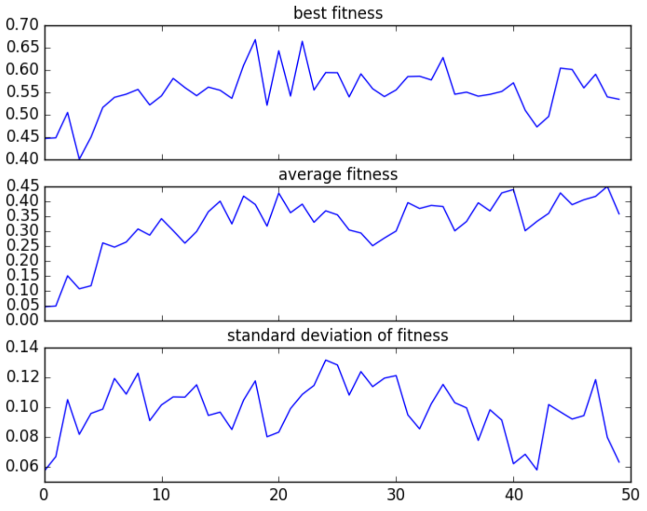
Dynamic – single scenario

The best fitness values fluctuate very strong during the run and the best agent in the final generation has a fitness of 0.622. He behaves only partly reasonable, skips some food and eats some poison.



Dynamic – five scenarios

The fitness values fluctuate less in this run, but they don't get much better. The best in the final generation is 0.534. This agent behaves most reasonable of all the tests in random maps, but he is not perfect. He skips some food and sometimes walks in circles.



In general, the runs with more than one scenario produce much better agents. Especially the dynamic run that tests each agent on only one map can not really generate and keep good agents, because an agent might be good on one map but then fail badly on another one. The run with five static scenarious performs almost as good as the dynamic one, which gives the best results. In the plots it can be seen, that the single dynamic runs fitness values fluctuate strongly, while in the dynamic – five scenario run, a general increase in both best and average fitness can be seen.