

Evolutionary Computation Theory and Application (ECTA) – Assignment 4: Pole Balancing

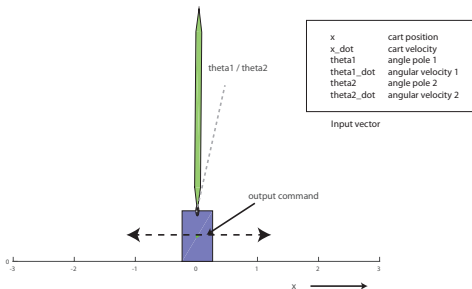
Alexander Asteroth, Adam Gaier, Alexander Hagg

Bonn Rhein Sieg University o.a.s., Department of Computer Science

Assignment 4: Pole Balancing

The Task

- Implement ESP and train an RNN to solve the double pole balancing problem
- Use the evaluation/simulation we offer you
- Inputs (see figure). **First** solve for all **6 inputs**
- **Second** solve for **4 inputs** (not the velocities)
- Stopping condition: target number of time steps (1000) — max. number of generations



Assignment 4: Pole Balancing

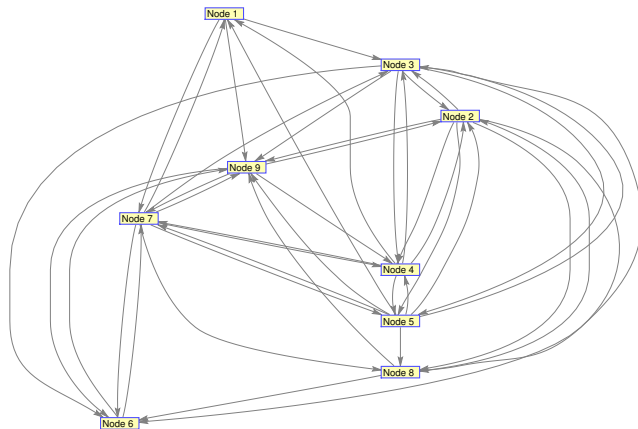
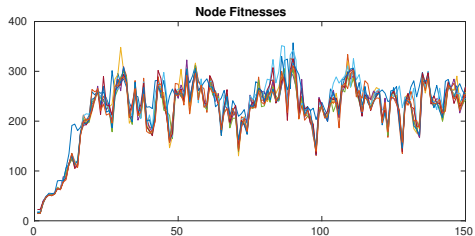
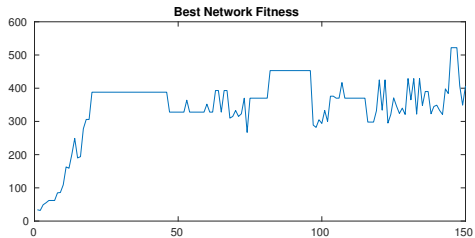


Figure : Resulting RNN, visualized with `view(biograph(wmat))`

Assignment 4: Pole Balancing



Assignment 4: Pole Balancing

bias	in1	in2	in3	in4	in5	in6	h1	h2	o1
0	1.03	0.30	-0.18	0	-0.39	1.64	-0.58	0	2.16
0.38	0.31	-0.38	-1.62	0	0	0	0	-1.70	-0.16
0	-0.47	0.17	-0.15	0.11	0.34	0	0	-0.28	0
0	-0.31	0	0	-0.23	-1.58	0	0	0	0
0	0.18	0	0	-0.57	0	-0.39	0	-0.14	0.16
-1.72	0	-0.20	0	0.39	-1.37	0	-1.20	-2.49	0
-0.40	0	0.16	-0.18	0	-0.35	-0.96	0	0	1.96
-0.40	0.11	-0.11	-0.24	-0.75	0	0	0	0	2.01
-0.37	0	-0.92	0	0	-0.44	-0.17	-0.21	0	0
-0.82	-0.90	0	0	0	0.10	0	0	0.35	-2.03

Table : Weight Matrix

Assignment 4: Pole Balancing

- Call the simulator
- Simulator needs function pointer to RNN activation
- Optionally use the simulator's builtin visualization

```
fitness = twoPole_test( wMat, @RNNNet, tFit );  
fitness = twoPole_test( wMat, @RNNNet, tFit, 'vis' );
```

- Used within the simulator
- Expected(!) function header of RNN activation

```
activation = RNNNet(wMat,[ bias input ], activation );
```

Assignment 4: Pole Balancing

- It might be a good idea to perform the evaluation of all individuals (does not work with visualization)
- Analyze the weight matrix and make sure it has the right shape
- Use a tanh activation function in the RNN
- Use permutations to select neurons from subpopulations (to combine them to networks)

Assignment 4: Pole Balancing

The Report

- Every .m file used should include comments describing the code
- Every .m file should be accompanied by an autogenerated report
- Use Sections (started with `%%`) to create headings out of comments within the code
- Report your results with illustrative figures and texts describing those figures
- You have two weeks to implement