Supervised and Reinforcement Learning of Neural Agent Controllers

Eirik Lid

January 2017

1 Simulation and visualization of Flatland + baseline agent

The whole project is programmed in Python 2.7. Tkinter was used for visualization. Map.py contains all visual and board classes. Agent.py contains the Agent class, which test, train, moves, etc. the agent.

The agent chooses tiles by type in the following order; food, open, poison, wall. If there is several tiles of the same type it chooses direction in this order: forward, right, left.

The baseline agent achieved an average score of 19.626 after 1000 trials

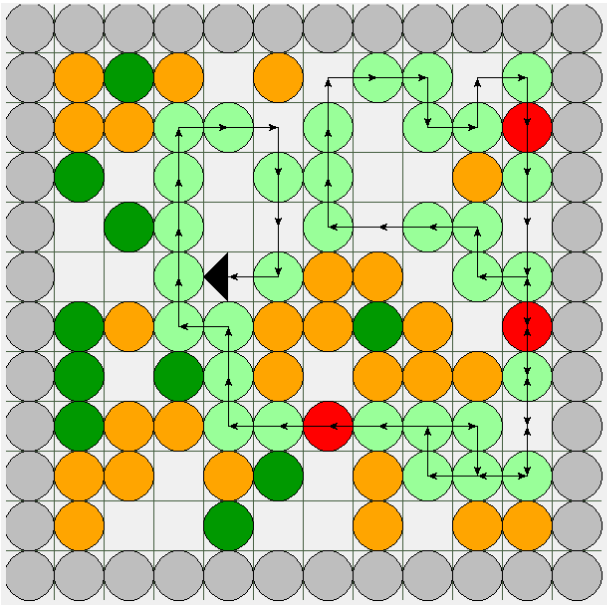

Figure 1: Visualization

2 Supervised learning of a neural agent controller

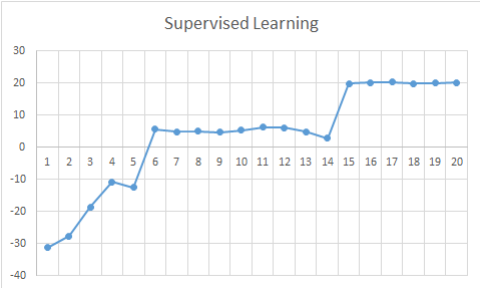

Figure 2: Scores for supervised learning

```
correct_choice = [1 if teacher_tile == neighbours[i] else 0 for i in range(3)]
exp_output = [math.exp(y_i) for y_i in output]
exp_output_sum = sum(exp_output)
delta = [correct_choice[i]-(exp_output[i]/exp_output_sum) for i in range(len(output))]
```

3 Reinforcement learning of a neural agent controller

```
Q_s_a = max(output)
neighbours = self.get_neighbours()

B_tile = self.choose_neural_action()
r = B_tile.val
B_dir = self.calc_dir(B_tile.x, B_tile.y)
s_marked = self.get_neural_input_at_tile(B_tile.x, B_tile.y,B_dir)
Q_s_a_marked = max(self.get_neural_output(s_marked))

delta = []

gamma = 0.9
for i in range(3):
    if (B_tile == neighbours[i]) :
        delta.append(r + gamma*Q_s_a_marked-Q_s_a)
    else:
        delta.append(0)
```

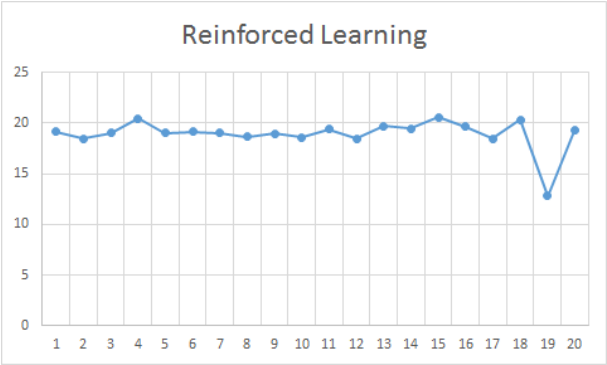

Figure 3: Scores for reinforced learning, with sensor range of 1

4 Extending the sensor range of the reinforcement agent

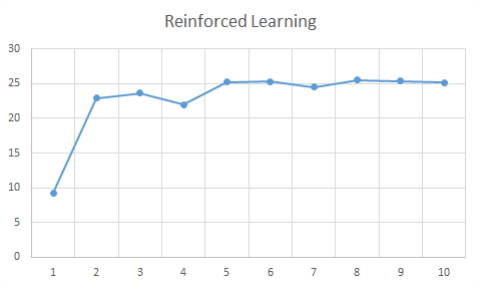

Figure 4: Scores for reinforced learning, with sensor range of 3

5 Analysis

Reinforced learning with sensor range at 1 gave the following weights:

	Front				Right				Left			
Output\Input	Open	Food	Poison	Wall	Open	Food	Poison	Wall	Open	Food	Poison	Wall
Front	2.316418	3.738222	-1.47658	-3.00408	0.582052	0.647126	0.690701	-0.34654	0.830622	-0.64907	0.821037	0.570846
Right	0.746151	-1.03901	0.809283	0.59282	2.166348	3.569357	-1.62426	-3.00177	0.611882	-0.9994	0.719459	0.776919
Left	0.823601	0.708633	0.852571	-0.26102	0.492557	0.503268	0.573514	0.555165	2.194531	3.612981	-1.68503	-1.99768

It can bee seen that the agent wants to move towards food, and if all tiles are open, it moves forward. The agent seems drawn to food to the left, but I can't explain why. Another thing to take notice of is action along walls. In both cases it moves away from the wall, rather than follow along the wall. Which makes sense since it gives 3 new tiles with a possibility of food, rather than 2.

(a) Food at all neighbours

Front	3.736
Right	1.531
Left	**4.825**

(b) Poison at all neighbours

Front	**0.035**
Right	-0.096
Left	-0.259

(c) Open at all neighbours

Front	**3.729**
Right	3.524
Left	3.511

(d) Poison in front, open at sides

Front	-0.064
Right	**3.588**
Left	3.540

(e) Open in front, food at sides

Front	2.314
Right	3.316
Left	**4.940**

(f) Wall to the left, rest is open

Front	3.469
Right	**3.689**
Left	-0.682

(g) Wall to the right, rest is open

Front	2.800
Right	-1.643735303
Left	**3.573**

Task 1,2 and 3 lay around 20 points. Task 4 got around 25 points, which corresponds to food every other tile. I didn't expect the reinforced agent to be a so much quicker learner than the supervised. It also has occurrences of dip in performance, which I suspect come from overfitting, perhaps weights that favor circling.