Report on Machine Learning Lab, Ex 3

Mostafa Mohamed, Omar Kassem

December 16, 2016

1 Introduction

This is a report about the deep learning lab, exercise 3. The general task of this assignment was to use Tensorflow/keras to implement a convolutional neural network to train it on solving a search problem and compare it to the A* search algorithm. As well as trying different configurations and architectures of the neural networks.

2 Search problem

The given problem was find a target position in a given maze. We were given a simulator that can visualize our algorithm or the given reference A* algorithm.

3 Architecture

- Input: There was a given script that generates the train/validation data according to the A* algorithm. It starts from a variety of random positions and tries to search for the target. The A* reaches the target optimally.
- Network: We used Keras to implement our neural network. We implemented a network that consists of 2 fully-connected layers followed by 3 convolutional layers. We tried different activation functions on some layers (like tanh), mostly the relu showed the best results.

4 Results

We tried different configurations for the running.

- Smaller local view: This possibility lead for an accuracy of 80%
- Different history length:
  - 10: 15/20 tests lead to 100% accuracy, 4/20 to 96% and 1/20 to 92% accuracies respectively.
  - 6: 19/20 tests lead to 100% accuracy and 1/20 to 96%.
  - 4: 15/20 tests lead to 100% accuracy and 5/20 to 96%.

- Changing the target after training: Lead to accuracy 45%, there was an interesting observation in this case, that the agent was smart enough to learn from it's own mistakes; in the sense that after it goes in a dead end path, it can go back and try different paths. Sometimes there was enough steps remaining to achieve the goal, sometimes not.
- Changing the map after training: lead to 0% accuracy, it got stuck in a lot of parts of the map because it was so unfamiliar.

5 Improvements

To improve the results we can try to train with different strategies.

- Train the agent on a several target locations, with a slightly longer history and/or larger view. We believe that this will improve the results on a given map, because the agent will get a better grasp of how the map looks like generally and how deadends look like also, and won't be fixed on a specific target point. However this will need much more training, because the agent can get confused easily if it was trained poorly on different targets, because the target's area isn't that relevant that much since it can change usually.
- We can have a slightly different strategy for storing history that can for example store the locations visited previously so it will learn the nogood areas (then we just store the nogood areas), while finding the target, hence accumulating failing experience, and try not to repeat bad areas to reach the target. This should show a potential improvement.