CSE 151B Project Final Report

Alexander Friend
apfriend@ucsd.edu

1 Task Description and Exploratory Analysis

1.1 Problem A [0.5 points]

Describe in your own words what the deep learning task is and why it is important. Provide some real-world examples where solving this task can have great impact on our daily life and the society.

1.2 Problem B [0.5 points]

Use Google Scholar or other internet resources to research on this task. What type of methods have been examined before? Include some references and discuss their ideas in a few sentences. You can use Bibtex to manage your bibliography.

1.3 Problem C [1 points]

Define the input and output in mathematical language and formu- late your prediction task. From the abstraction, do you think your model can potentially solve other tasks beyond this project? If so, list some examples and explain your rationale.

2 Exploratory Data Analysis

3 Deep Learning Model

3.1 Problem A [1 Points]

Thus far my best performing model is a PyTorch linear neural net running in an Anaconda environment. The platform I am currently working in is my local machine running Ubuntu 20.04 with a 4-core 4-thread Intel i7-7600k CPU running at 4.2 GHz, a GTX 1070 GPU with a max clock speed of 1721 MHz and 8 GB of GDDR5 memory, and 16 GB of 3000 MHz DDR4 memory.

For my initial baseline model I made a simple single layer linear neural network, using Adam as an optimizer And and mean square error as my loss function. I used a learning rate of 0.001, and trained my best model for 10 epochs. Each epoch took approximately 3 minutes to train.

I chose mean square error as it increases the penalty for worse predictions, compared to a optimizer like mean absolute error, which penalizes with a linear rate.

In order to get the correct outputs from my trained model, I sliced the output prediction tensors so as to only keep the first inner 2 elements of each tensor (the predicted x and y coordinates), and then selected the correct car based on each inputs `agent_id`.

Preprint. Under review.

3.2 Problem B [1 Points]

I experimented with multilayer linear models and an LSTM model as well, however these actually performed worse on the validation set once I uploaded them to Kaggle. Thus far, the best performing model is a linear one, which takes in the assumption that cars will be moving at about the same rate over the 5 seconds period of training and prediction.

4 Experiment Results and Future Work

4.1 Problem A [1 points]

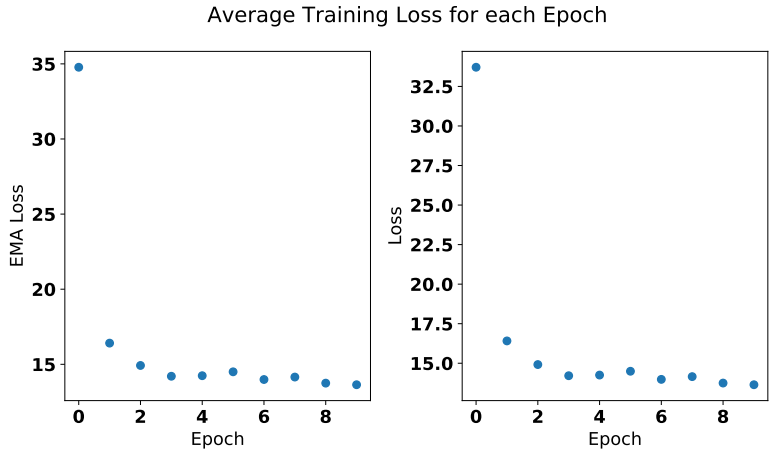

Figure 1: Plot of the average training mean square error for my best perfoming single layer model

This model however achieved a significantly lower loss on the training set compared to the final validation set on Kaggle, suggesting that my model has overfit the training data. This means that a more complex model with more variance will need to be implemented.

Based on my lack of success with an LSTM approach, I will continue to work on more complex linear models, as well as trying other architetures such as a RNN model, which may achieve a better result as it can take into account previous training examples.

My current rank on the Kaggle leaderboard is 40 out of 57.

A Appendix

`https://github.com/apfriend/cse151b-kaggle.git`