Directed Research on the Performance of Exotic Neural Networks on an End-to-End Self Driving Car Model

Siddartha Yeliyur Shivakumara Swamy

December 10, 2018

1 Introduction

This directed research involved, implementation and comparison of the performances of various new and existing neural networks on an existing End to End self driving car model, which gives the angle of the steering wheel as output[1].

The neural networks used to replace the vanilla neural network were the VGGNet[3] and the Capsule Neural Network[4][2].

Firstly, I proceeded with replacing the existing standard convolutional neural network, with a non-standard VGGNet. Non-standard as all of the max pooling layers were removed and only the convolutional layers retained. This was then trained on the comma.ai self-driving dataset.

Upon the successful implementation of the VGGNet, I proceeded to implement the Capsule Neural Networks[4][2] in place of the standard convolutional neural network. A capsule neural network was chosen as it is one of the latest development in the field of deep learning that can retain spatial information between the different parts of an image. This was then trained on the comma.ai self-driving dataset.

All the code is in my github: https://github.com/Puttu1994/DR-Directed-Research-on-the-Performance-of-Exotic-Neural-Networks-on-an-End-to-End-Self-Driving-Car-.git

2 Theory

2.1 VGGNet

VGGNet is a neural network that performed very well in the Image Net Large Scale Visual Recognition Challenge (ILSVRC) in 2014. It scored first place on the image localization task and second place on the image classification task.

This architecture is from VGG group, Oxford. It has advantage of using multiple 3X3 kernel-sized filters one after another. With a given receptive field(the effective area size of input image on which output depends), multiple stacked smaller size kernel is better than the one with a larger size kernel because multiple non-linear layers increases the depth of the network which enables it to learn more complex features, and that too at a lower cost.

2.2 Capsule Neural Networks

The Capsule Network is a new type of neural network architecture conceptualized by Geoffrey Hinton[4], the motivation behind Capsule Networks is to address some of the short comings of Convolutional Neural Networks, such as:

1. Convolutional Networks are Translational Invariant
2. Convolutional Networks require a lot of data to generalize

Capsule Neural Networks aim to solve this by adding structures called capsules to a Convolutional Network, and to reuse output from seeral of those capsules to form more stable representation for higher order capsules. The output is a vector consisting of the probability of an obseravation and a pose for that observation.

3 Implementation

3.1 VGGNet

As the problem involves predicting the angle of steering wheel as output, the standard VGGNet required modifications. Firstly, as VGGNet has a number of max pooling layers, it could result in loss of information, and every bit of information is crucial as the safety of others is at risk when using a self-driving car. Hence, the maxpooling layers were all removed and the convolutional layers were connected back to back. Secondly, as the VGGNet was built for classification task, the softmax layer at the end of the network would not help in a regression task, and as such the softmax layer was replaced with a fully connected layer with a single neuron(single output). The network was implemented using TensorFlow and a batch size of 200 was used for 20,000 iterations.

3.2 Capsule Neural Networks

The first 5 layers of my implementation of capsule neural network contain 5 convolutional layers, of difference filter sizes. The kernel size is either a 5x5 or 3x3. This was done as performing convolution on capsules once they were created is computationally very and expensive and I was not able to run it on my local machine. Hence I resorted to performing convolution at the beginning, since it would extract much of the information. Following this, I used the output of the 5 convolutional layers and reshaped them into capsules, while performing an additional convolutional operation on them. The output were reshaped into capsules each of vector size 8x1. This is the primary capsule layer. To extract more information I performed another additional convolutional operation on these capsules with a filter size of 32. The capsules were then flattened to form a fully connected capsule layer, called the dense layer, with a single output, as this is a regression task. The network was implemented using TensorFlow and a batch size of 1 was used for 30,000 iteration.

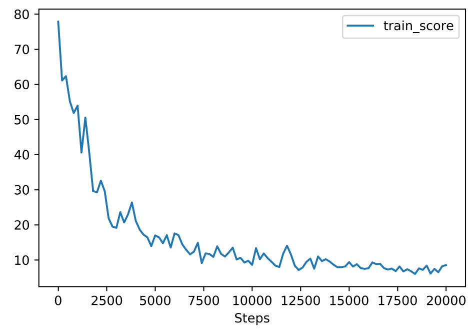

Figure 1: Train Error for the CNN implemented in the original Paper.

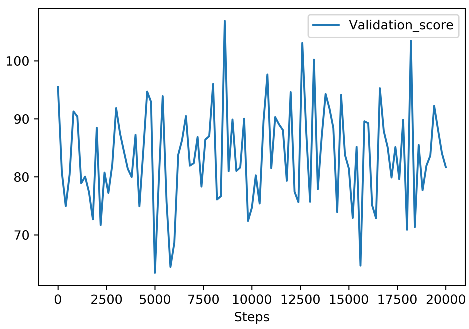

Figure 2: Validation Error for the CNN implemented in the original Paper. The validation error greatly reduces and almost equivalent to train error (4-6%) before it reaches step number 30,000

4 Results and interpretation

The results of using VGGNet and Capsule Neural Networks to predict the angle of a steering wheel is as follows:

4.1 VGGNet:

Upon implementing and training on the comma.ai dataset using a modified non-standard VGGNet, I find that the VGGNet performs badly. This is probably due the fact that the VGGNet is a very deep network, containing 19 layers, which results in a loss of generalization.

The figures 3 and 4 show the training and validation error for 20000 iterations.

4.2 Capsule Neural Networks

Upon implementing and training capsule neural network[2] on the comma.ai dataset, I find that the network is highly volatile. The networks performance is highly variant. Sometimes it performs really well for 3-4 iterations, resulting in

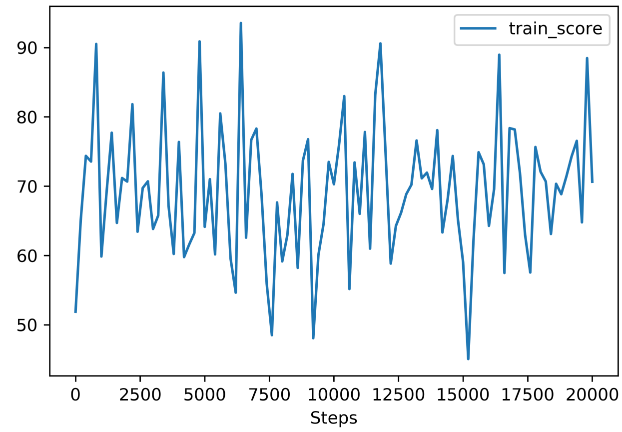

Figure 3: Train Error for the implemented non-standard VGGNet.

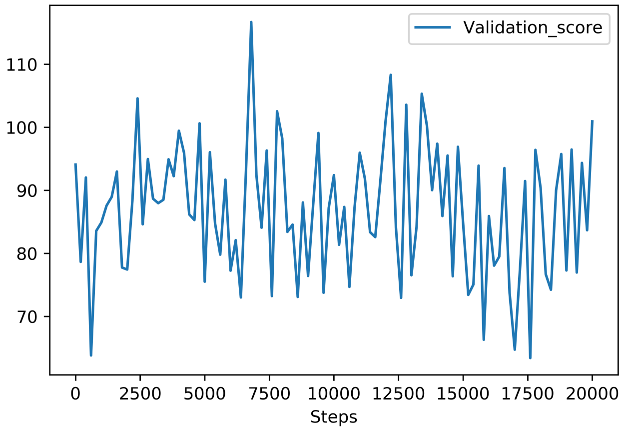

Figure 4: Validation Error for the implemented non-standard VGGNet.

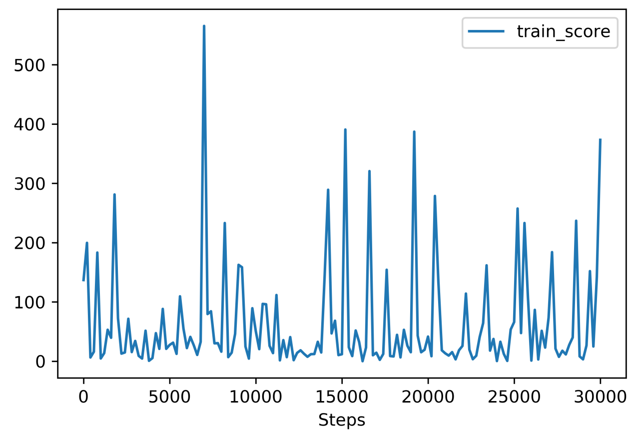

Figure 5: Train Error for the implemented Capsule Neural Network.

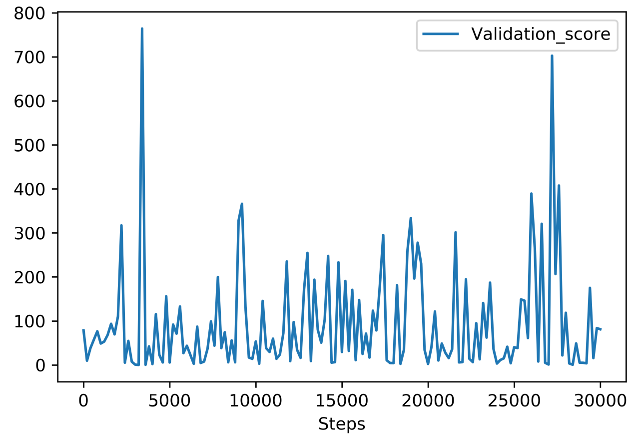

Figure 6: Validation Error for the implemented Capsule Neural Netowrk.

a training and validation error of 1-5%, while in other cases it results in a loss of 100-700%. This can be attributed to the fact that the batch size is 1. This is the least possible batch size to train with and hence the network can readily overfit. The reason for such a low batch size is the lack of computational power available at my end.

I tried to solve this problem by using the concept of dropout. This did not show any improvement in performance.

The figures 5 and 6 show the training and validation error for 30000 iterations.

5 Dataset

The dataset used for this directed research was the comma.ai driving dataset available publicly.

The dataset is fairly large, containing 7 and a quarter hours of largely highway driving.The size of each image(frame) in the dataset is 160x320x3.

The dataset is available at http://research.comma.ai/. This particular dataset was the solely used dataset as this enabled me to have a reference to compare the performance of the implemented non-standard VGGNet and Capsule Neural Networks.

6 Discussion

In this Directed Research, I have gained knowledge about neural networks, learned about 2 different deep learning architectures, namely, VGGNet and Capsule Neural Networks and how to successfully implement them to find a solution to a problem.

Unfortunately, both implementations did not give good results. While VGGNet does not perform well, Capsule Networks shows some promise with good results occurring intermittently, which may improve when trained with computationally powerful system.

For further research I would like to understand and implement the Clockwork RNN in place of the present LSTM in the attention model of the system.

References

[1] John F. Canny, Jinkyu Kim, *Interpretable Learning for Self-Driving Cars by Visualizing Causal Attention.* University of California, Berkeley, arXiv:1703.10631v1 [cs.CV]

[2] Huadong Liao, Jiawei He, *CapsLayer: An advanced library for capsule theory*, GitHub, GitHub Project, http://naturomics.com/CapsLayer

[3] Karen Simonyan, Andrew Zisserman, *Very Deep Convolutional Networks for Large-Scale Image Recognition*, Visual Geometry Group, Department of Engineering Science, University of Oxford, arXiv:1409.1556v6 [cs.CV]

[4] Sara Sabour, Nicholas Frosst, Geoffrey E. Hinton, *Dynamic Routing Between Capsules*, Google Brain, Toronto, arXiv:1710.09829v2 [cs.CV]