

Cognitive Robotics Lab Report

Sean Parker
34212-Lab-S-Report

April 15, 2020

1 Introduction

The aim of this report is to describe the current problem in computer vision for solving image recognition, which is very important in the area of robotic vision, to describe the current state-of-the-art network architectures and finally to describe my approach for solving this problem. This project used the CIFAR-10 dataset which consists of 60000 32x32 colour images, grouped into 10 classes. Although the dataset has low quality images, and for a real-world robot it's visual acuity would be much greater (for example iCub which use 2x Dragonfly2 camera, each with a resolution of 648x488 pixels), solving image recognition on a simple dataset provides a proof of concept for a system that could be upscaled in order to perform image recognition on higher resolution images.

Firstly, the network architecture was chosen before evaluation of the network with a selection of hyperparameters. In order to find a good architecture, I researched the current state of the art (see Section 2) in order to determine the baseline performance. Using a very deep CNN architecture, one can achieve an error rate of only 1.0%[\[1\]](#), however this architecture has over 500-million parameters which is far too large for the available computer power for this project.

As such, I looked for simpler network architectures that still achieved good results on the CIFAR dataset. After some research in modern techniques to improve training networks, I decided to include layers using Batch Normalisation and Dropout layers which have been shown to improve the accuracy of networks[\[2, 3\]](#).

Lastly, in terms of hyperparameter choice, there were some main parameters that had to be refined for the network topology. The main hyperparameters that had to be defined were the learning rate of the optimiser, and weight decay for batch norm layers. Additionally, the network topology is based on blocks of (CONV + ReLU + Batch Norm)¹. More details of the network architecture can be found in Section 3.

2 State of the art

3 Implementation

4 Conclusion

References

- [1] Yanping Huang, Youlong Cheng, Ankur Bapna, Orhan Firat, Mia Xu Chen, Dehao Chen, Hyoungho Lee, Jiquan Ngiam, Quoc V. Le, Yonghui Wu, and Zhifeng Chen. Gpipe: Efficient training of giant neural networks using pipeline parallelism, 2018.
- [2] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(56):1929–1958, 2014.
- [3] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift, 2015.

¹In this case, “+” refers to a combination of ideas, rather than the plus operator in mathematics.