Optimizing Neural Networks for Mobile Devices

Erdi Çallı

Radboud University Nijmegen, Neurant

Abstract

Recent developments in Neural Networks(or Deep Learning) are promising. Some models are capable of accomplishing tasks as good as humans, or better. But we still lack the applications that are available to the public. The general opinion is, Neural Network models need expensive equipment. But that only applies to the training process, where the network learns from data. But using the trained model for inference is easier. There are also methods to reduce the computational complexity of a model. With these methods, we may be able to use cheap compute devices(i.e. Mobile Phones) for inference. Thus making some models available to public.

Optimizing Neural Networks for Mobile Devices

Introduction

Recent state of the art Deep Learning models are surpassing previous methods. Fields such as; computer vision, automatic speech recognition, natural language processing, speech recognition, and bioinformatics make use of these models. They use deep models consisting of many layers (e.g. 152 layers in He, Zhang, Ren, and Sun (2015)), many parameters in each layer, and as a result, a lot of Floating Point Operations to run Inference (e.g. 11.3×10^9 in He et al. (2015)). In contrast, mobile devices have limited processing power and memory. Also the best practice is to provide a fluent user experience with low response time. Thus, we should change these models to provide a good user experience. There is research on methods to define optimized models or optimize a given model. These methods consist; pruning unimportant parameters, using less bits to represent parameters, or using less parameters by using more optimized structures.

In this research we are going run experiments to answer;

1. Which models are running slow in Mobile Devices?

2. Why these models are running slow?

3. Which methods can we use to optimize these models?

4. What is the trade off of using these methods?

5. Why an optimization technique is working or not on a model?

6. Can we define a more optimized model for the same task?

7. How can we combine different optimization techniques?

8. Are these optimized models efficient enough to run in Mobile Devices?

Recent Studies

Artificial Neural Networks (ANN) have several parameters such as number of hidden layers, number of neurons in a layer, or the structure of a layer. Until now, we have seen different combinations for these parameters. For example, Simonyan and Zisserman (2014) introduces a model called VGGNet. VGGNet introduces more layers (16 to 19) than the previous models. They show how this parameter effects the accuracy. He et al. (2015) introduces the residual connections. This new connection between layers is capable of stacking more layers than before. Training up to 152 layers, they show superior accuracy. Zagoruyko and Komodakis (2016) compares having higher number of neurons in each layer to having more layers. Each combination resulting in a unique model with a different accuracy level. In contrast to all these, Szegedy et al. (2014) suggests something different. Having a good harmony within the network works better than having more parameters. Supporting that, Canziani, Paszke, and Culurciello (2016) does a detailed comparison of different models. They show that, increasing the number of hidden layers or the number of neurons in a layer does not necessarily increase the accuracy.

Following these, we think that, some models are over-parameterized. Meaning they contain parameters that they are not making use of. Therefore, they are making computations with these parameters that they are not making use of.

Methods

Activation Based Pruning

Introduction. Studies show that pruning unnecessary neurons in an ANN is effective in reducing the model complexity. Activation based pruning, sums up the neuron activation counts. Using that information, it determines the useless neurons, and deletes them. In theory, this method reduces the training time per cycle, and prioritizes important connections. We set up an experiment to see the efficiency of this method. We defined a problem with a simple optimal solution, summation of two inputs. We defined a network with redundant parameters and trained it in different

configurations. Our experiments show that this method is working to its expectations.

We can talk about how it is important in the nature, check Rust12, George, and Bolouri23 (1997)

References

Canziani, A., Paszke, A., & Culurciello, E. (2016, 05). An analysis of deep neural network models for practical applications. Retrieved from https://arxiv.org/abs/1605.07678

He, K., Zhang, X., Ren, S., & Sun, J. (2015, 12). Deep residual learning for image recognition. Retrieved from https://arxiv.org/abs/1512.03385

Rust12, A. G., George, S., & Bolouri23, H. (1997). Activity-based pruning in developmental artificial neural networks. In *Fourth european conference on artificial life* (p. 224).

Simonyan, K., & Zisserman, A. (2014, 09). Very deep convolutional networks for large-scale image recognition. Retrieved from https://arxiv.org/abs/1409.1556

Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., ... Rabinovich, A. (2014, 09). Going deeper with convolutions. Retrieved from https://arxiv.org/abs/1409.4842

Zagoruyko, S., & Komodakis, N. (2016, 05). Wide residual networks. Retrieved from https://arxiv.org/abs/1605.07146