ME Project:
Low Bit Width Neural Networks

Sean Fanning (UCD: 13360951)
Monthly Report 1, October 16, 2017

Table of Contents

1 Work plan

Over the past month, I have worked mostly on state-of-the-art research and online courses to develop a better understanding of neural networks. I have been progressing through the Coursera course Machine Learning https://www.coursera.org/learn/machine-learning/ and I am continuing UCDs machine learning course which similarly covers the theory of machine learning and neural networks. I also plan on completing other courses which focus on using tools such as Tensorflow to develop neural networks. I already have some experience coding in Python, however I will need to work on learning the machine learning specific tools. I have installed both Tensorflow and Theano, as well as Keras on my computer, and I intend to familiarise myself with these software packages in the next few weeks.

Before my next monthly report, I intend to complete the online courses, and have a good understanding of the software and technologies that will be in use for the project. I hope to take an in depth look at some of the open source projects which implement low bit values in neural networks. I have already tried BinaryNet https://github.com/MatthieuCourbariaux/BinaryNet which was developed by Matthieu Courbariaux, however the training was taking too long on my laptop as I do not have a compatible gpu. I hope to get started using the TitanX machine so that I can begin experimenting with different implementations. Other projects I intend on looking at are SqueeseNet https://github.com/DeepScale/SqueezeNet and DoReFa-Net https://github.com/ppwwyyxx/tensorpack/tree/master/examples/DoReFa-Net

By December, I hope to have a good enough understanding of Tensorflow, and deep learning theory so that I can develop a conventional neural network for handwriting recognition on the MNIST dataset. I also hope to have enough experience to begin experimenting with my own implementations.

As of now, I am not entirely sure of the amount of time that will be required for some of the later steps. My rough plan is from January onwards to be working on a binarized implementation of the neural network and benchmarking it against the conventional network. After this, I will develop ternary and quinary networks also compare them. I hope to benchmark all of the networks on different datasets, as some implementations will likely have advantages for some, and disadvantages for others. If possible, I would also like to experiment using embedded or mobile devices, as the advantages of low bit width neural networks would suit devices such as these, where memory and power requirements are limited.

2 Git & Backups

I have created a git repository for my project which is available here: https://github.com/SeanFanning/Low-Bit-Width-Neural-Networks. It will contain code, documentation and the reports as I progress in the project. I have also created an Overleaf project for the monthly reports and other documents using LaTeX.

3 State of the Art

While the concept of low bit width neural networks is not new, there has been significant development in this area in recent years. There have been several new developments in recent years, and even in

the past few weeks new papers have been published, so I plan to keep up to date with the ongoing development in the area.

Many of the recent developments in binarized neural networks have stated similar advantages in terms of memory consumption, time complexity and energy consumption over conventional neural networks. For certain datasets, they achieve near state-of-the-art results in accuracy, however in some cases they are less effective and see a reduction in performance. It is clear that there are some workloads that would suit a binarized neural network, and ternary and quinary networks may provide a reasonable compromise for some datasets. Another area which would suit low bit width neural networks is in mobile or embedded devices where power consumption is priority.

The paper Binarized Neural Networks [1] introduces a method to train binarized neural networks for runtime by using binary weights and activations during train time. This results in a significant reduction in memory size and accesses, which is generally one of the limiting features in neural networks and accounts for a significant portion of the time and power improvements observed in this paper. By using binary values, it is possible to use bit-wise operations instead of arithmetic operations. This has clear advantages in terms of time complexity, and also power consumption, especially on smaller embedded devices.

Tests were done on both Deterministic Binarization and Stochastic Binarization. A GPU kernel was developed, which could reduce time complexity by 60%, with comparable results to 32bit counterparts. For small datasets such as MNIST, state-of-the-art accuracy was observed, however larger datesets such as ImageNet resulted in a degradation in performance. It was noted that by using more than 1 bit, state-of-the-art results can be achieved. This is covered in another paper by the same authors Quantized Neural Networks [2]

The paper Quantized Neural Networks [2] was written by the same authors as Binarized Neural Networks, and uses a similar method of extremely low precision weights and activations at run time. During the forward pass, the quantized neural network also drastically reduces the memory size and accesses, and the arithmetic operations are still able to be replaced by bit-wise operations. By using 1-bit weights, 2-bit activations, 6-bit gradients, the gradients can be calculated using only bit-wise operations. As a result, the quantized neural network sees similar advantages over a conventional neural network in terms of time, memory and power consumption. For more complex datasets such as ImageNet, the quantized neural network resulted in a noticable improvement over previous attempts at low bit neural networks, however they noted that the accuracy was still below conventional 32 bit counterparts.

Training Low bitwidth Convolutional Neural Networks with Low Bit-width Gradients [3] is another paper which covers stochastically quantised parameter gradients. The network DoReFa-Net (`https://github.com/ppwwyyxx/tensorpack/tree/master/examples/DoReFa-Net`) was developed which is derived from AlexNet.

The paper Embedded Binarized Neural Networks [**?**] covers allowing current binarized Neural Networks to perform feedforward inference efficiently on small embedded devices. It focuses on minimizing the required memory footprint of neural networks, as embedded devices are often limited to very low amounts of memory. By using low bit binary values instead of floating point values to store weights, the memory used by temporary intermediate values is significantly reduced by up to 32x. This embedded binarized neural network was tested on an Intel Curie with only 15kB memory, and was observed to have an accuracy of 95% on the MNIST dataset, with runtime of under 50ms per sample.

The paper Efficient Inference Engine on Compressed Deep Neural Network [4] also notes the ad-

vantages of reducing the memory consumption of neural networks. It focuses on deep compression, by having multiple connections share the same weight. Fetching weights from DRAM is 2 orders of magnitude more expensive than ALU operations. Therefore, if a deep neural network can be stored entirely in on-chip SRAM, it would have a great improvement in speed.

It will be interesting to benchmark low bit width neural networks on different datasets. Since binary networks may prove to be optimum for certain datasets such as MNIST, while quinary networks may be more suited to larger datasets such as ImageNet. It would also be interesting to test low bit width neural networks on embedded and mobile devices, as the advantages such as low memory and power usage would certainly be beneficial for these kind of devices.

References

[1] I. Hubara, M. Courbariaux, D. Soudry, R. El-Yaniv, and Y. Bengio, "Binarized neural networks," in *Advances in Neural Information Processing Systems 29*, D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, Eds. Curran Associates, Inc., 2016, pp. 4107–4115. [Online]. Available: http://papers.nips.cc/paper/6573-binarized-neural-networks.pdf

[2] I. Hubara, M. Courbariaux, D. Soudry, R. El-Yaniv, and Y. Bengio, "Quantized neural networks: Training neural networks with low precision weights and activations," *CoRR*, vol. abs/1609.07061, 2016. [Online]. Available: http://arxiv.org/abs/1609.07061

[3] S. Zhou, Z. Ni, X. Zhou, H. Wen, Y. Wu, and Y. Zou, "Dorefa-net: Training low bitwidth convolutional neural networks with low bitwidth gradients," *CoRR*, vol. abs/1606.06160, 2016. [Online]. Available: http://arxiv.org/abs/1606.06160

[4] S. Han, X. Liu, H. Mao, J. Pu, A. Pedram, M. A. Horowitz, and W. J. Dally, "Eie: Efficient inference engine on compressed deep neural network," in *Proceedings of the 43rd International Symposium on Computer Architecture*, ser. ISCA '16. Piscataway, NJ, USA: IEEE Press, 2016, pp. 243–254. [Online]. Available: https://doi.org/10.1109/ISCA.2016.30