Senior Thesis

Clement Lee

October 6, 2016

Motivation and Goal

Deep learning is an active area of research, but modern approaches have generally focused on a consistent improvement of accuracy and precision, at the cost of performance. Modern networks can take on the orders of weeks to train, even with specialized hardware and large clusters of servers.

Recently, increasing numbers of researchers are finding that it is possible to create far more efficient networks, using highly tuned methods and clever tricks. The aim of this project is to examine and develop better methods for deep networks to automatically optimize themselves, without needing what is an increasingly large corpus of expert knowledge to design. In particular, network architecture has typically been based solely on what has experimentally worked, and thus few people deviate from a few common designs. The space of potential structures is vast, and is deserving of additional exploration.

Problem Background and Related Work

A few modern results like Squeezenet [3] and Binarynet [1] have demonstrated that it is possible to engineer networks that rival modern networks in basic image classification problems. This means that current approaches are likely overdimensioned and can be reduced. However, a key limitation of such works is the amount of tweaking that is necessary to achieve such performance; the networks have to be precisely a certain structure in order to maintain such performance, without good understanding of what factors are truly contributed to increased accuracy.

Even extraordinarily basic measures to weight pruning have been explored, which can show immense compression of popular networks for essentially no effort [2]; many similar results exist in the literature to show that the vast majority of weights in trained networks are effectively zero and therefore inconsequential. At the same time, older methods like Optimal Brain Damage [4] have been

less explored in the modern literature except for small cases, often in extremely small networks designed to model trivial problems. A large scale application of improved weight-removal algorithms has not been developed.

Approach

This project approaches the problem through a different light: instead of attempting to simply prune a fully trained network through simple heuristics, I intend to examine better ways of understanding how different weights may influence the structure of the network. Simple weight removal can be further extended with more knowledge about the topology of the network, but additionally perhaps with a balance of more expensive-to-calculate algorithms combined with the faster heuristics.

In addition, all of these prior methods are focused on deletion. This project will aim not just to produce an improved pruning method, but also to produce a network that can build and scale itself to available hardware and improve its performance. This will potentially allow networks to find their own optimal structure, rather than requiring many manual tests by researchers to find what works best for a certain problem. If this process can be done continuously (i.e. slowly rather than in blocks at a time), this could produce significantly better results than constant tuning of hyperparameters.

Plan

This project will likely be built on common deep learning libraries, such as Torch or TensorFlow. Taking some prior knowledge of such libraries, each will have to be further examined to see which is most compatible with the modifications that this project will be aiming for. Access to the primitives that make up a network, and the ability to deeply hook into training functions will be crucial to building up an algorithm. As such, investigation of the library will be an important first step.

In parallel, understanding the ins and outs of backpropagation and error calculation will be crucical to developing a better algorithm for either weight deletion or construction. This will involve not just the basic ideas of the algorithms, but also learning about modern techniques to improve training, such as Stochastic Gradient Descent. By getting a better grasp of such algorithms, opportunities for better automation will become clearer. Further steps will follow as these are completed.

Evaluation

A number of general testing suites exist to test the efficacy of novel deep learning methodologies. In particular, image classification with datasets such as CIFAR-10 or CIFAR-100 are extremely widespread and are effectively a baseline benchmark for any general method. This also allows the project to rely on more common tools and libraries that have already been optimized to load such datasets and preprocess them.

However, as the methodology proposed is extremely general, it is also a plan to investigate the performance of the algorithm across different domains of learning. It has been observed that deep networks are able to identify patterns in data extremely well, but a variety of other structures are less explored as an input into such algorithms. I aim to also examine my algorithm's potential on atypical datasets (i.e. outside the realm of speech and image recognition), though the specifics of this will require additional thought.

References

[1] Courbariaux, M., and Bengio, Y. Binarynet: Training deep neural networks with weights and activations constrained to+ 1 or-1. *arXiv preprint arXiv:1602.02830* (2016).

[2] Hu, H., Peng, R., Tai, Y.-W., and Tang, C.-K. Network trimming: A data-driven neuron pruning approach towards efficient deep architectures. *arXiv preprint arXiv:1607.03250* (2016).

[3] Iandola, F. N., Moskewicz, M. W., Ashraf, K., Han, S., Dally, W. J., and Keutzer, K. Squeezenet: Alexnet-level accuracy with 50x fewer parameters and¡ 1mb model size. *arXiv preprint arXiv:1602.07360* (2016).

[4] LeCun, Y., Denker, J. S., Solla, S. A., Howard, R. E., and Jackel, L. D. Optimal brain damage. In *NIPs* (1989), vol. 2, pp. 598–605.