Neural Network Pruning via Relational Graphs

Andrei Rekesh 105343158
Sahil Bansal 905525442
Vaibhav Kumar 305710616
Vivek Arora 505526269

Abstract

Despite the wide use of neural networks, there is little understanding of the relationship between the graph structure of the neural network and its predictive performance. We try to find an efficient way to determine the correlation between the structure of neural network and its performance using pruning techniques on relational graph representation of a neural network.

1 Problem and Goal

Over the past decade, advances in neural network architecture have accelerated progress in computer vision, natural language processing, and geometric learning. Current understanding on design principles surrounding new architectures is still vague. In this project, we aim to systematically study what types of computation graphs lead to better performance in common machine learning tasks, and if any general design principles can be found through this analysis. One point of interest is the neural network weights after learning.

In this project we try to implement conversion of a neural network into a relational graph (You u.a. (2020)) and try pruning it through analyzing the magnitude of the weights. We try to find an efficient method to prune the this architecture before even training.

2 Data Plan

We would be using the CIFAR-10[1] and MNIST[2] datasets for the purpose of our project. The CIFAR-10 dataset consists of 32x32 colour images in 10 classes, with 6000 images per class. There are 50,000 training images and 10,000 test images. The MNIST dataset consists of 28x28 grayscale images of handwritten digits with a training set of 60,000 images, and a test set of 10,000 images.

We would be using these datasets as these are the standard datasets for exploring neural networks and considering the large number of candidate graphs that we would be training, these datasets would provide us an efficient way to benchmark our approach.

3 Solution Plan

In this project, we would first be representing a neural network architecture as a relational graph and then come up with a method to represent the norm of edge computation weights. We would then try training the model by pruning P% (Frankle u.a. (2018)) of the edges in the computation graph every K iterations.

[1] https://www.cs.toronto.edu/ kriz/cifar.html
[2] http://yann.lecun.com/exdb/mnist/

What we aim to achieve is to optimize the hyperparameters for K and P. One of the other bottlenecks while tackling this problem is that the considering norm representation method might not be the most optimal or won't just work to our expectations. In that case, we would have tune it to fit our expectation. Ideally, low value of P should not results in much degradation in the performance whereas a larger value of P should result in large performance degradation. The models that we plan on experimenting with in this project are fixed width multi-layer perceptron (MLP), variable width MLP and ResNet34* (He u.a. (2015)). The datasets that would be considered are CIFAR-10 and MNIST. We plan on performing detailed set of experiments to have a better understanding and insight into the significance of the norm of edge computation weights in a relational graph. Finally, we plan on formalizing out results in detailed report.

4 Evaluation Plan

In this paper, we will be using two graph measure to study the characteristic of the graphs. We will be using a global graph measure, average path length, and a local graph measure, clustering coefficient. These two measures are widely used in the field of network science. Here, the average path length measures the average shortest path distance between any pair of nodes. On the other hand, clustering coefficient measures the proportion of edges between the nodes within the neighborhood of a given node, divided by the total number of possible edges that could exist between them. It's obtained after averaging is done over all the nodes.

For the purpose of evaluation, we would be using accuracy (on the considered task), the number of parameters (% parameter reduction) and the FLOPs (Float Points Operations) to evaluate the model size and corresponding computational requirement. We further plan on considering (Chen u.a. (2021)) as one of the baseline methods for benchmarking the performance of our considered approach.

5 Schedule

We will follow the following schedule for our course project:

Week 3-4 We would start by doing a literature survey on existing methods to represent a neural network as a relational graph and existing pruning techniques. Then we would setup the baseline code and install all required packages and dependencies

Week 5 We would get the codebase running for different datasets/models and come up with method to represent norm of edge computation weights.We would start our experiments with the following models: Fixed width MLP, Variable width MLP, ResNet34*.

Week 6-7 We would try training the above architectures with pruning P% of edges in computation graph every K iterations and optimize the hyperparameters. We would be tuning norm of edge computation weights if our initial method doesn't work.

Week 8 We would work on getting the formal experimental results. Ideally, low P% should not result in much performance degradation whereas high P% should result in large performance degradation.

Week 9-10 In the last weeks we would we wrapping up on collating our results and work on our report writing and presentation.

Distribution amongst team members Our team would be working on each of the parts equally. We would all be doing the literature survey and discuss together the insights. We would be dividing the coding part once we move forward in the implementation part of the project.

References

Frankle, Jonathan / Carbin, Michael(2018): *The lottery ticket hypothesis: Finding sparse, trainable neural networks*.

You, Jiaxuan / Leskovec, Jure / He, Kaiming / Xie, Saining(2020): *Graph structure of neural networks*10881–10891.

Chen, Zhuangzhi / Xiang, Jingyang / Lu, Yao / Xuan, Qi(2021): *RGP: Neural Network Pruning through Its Regular Graph Structure*.

He, Kaiming / Zhang, Xiangyu / Ren, Shaoqing / Sun, Jian (2015): *Deep Residual Learning for Image Recognition*

.