Author: Blake Edwards
References: Dr. Pooyan Jamshidi
Date: 08/27/19

Evolutionary Pruning and Knowledge Distillation for Effective Model Compression

Pruning has been widely used and has proven to be an effective methodology to achieve a massive reduction in the computational complexity of large neural networks. Stated simply, pruning is the process of determining which neural weights in a network are important and which are not. Those that fall below a certain rank of importance will be removed during pruning to reduce the number of overall neurons that need to be computed. This is done by looking at the activation and gradient outputs of logits within the target neural network and removing the weak neuronal connections that contribute the least to the networks performance.

Evolutionary Pruning and Knowledge Distillation (EPKD) is a novel approach to reducing the size and computational complexity of large neural networks through iterative compression. This method leverages state of the art techniques in pruning and knowledge distillation in order to achieve small, highly effective student models that perform like starting high accuracy teacher models.

The EPKD algorithm works as follows. A large, cumbersome teacher network is trained to perform with very high accuracy on a labeled training set of data. After training, a variety of pruning approaches are applied to the teacher network in order to reduce its size and computational complexity. These pruning approaches are applied

very carefully and specifically across each compression iteration in order to remove the same number of neuronal connections across student networks created using different pruning methods. This ensures that resulting student networks have similar numbers of trainable parameters, allowing them to be fairly compared against the performance of other students in their respective compression iteration. During each compression iteration, student networks are ranked based on a 1:3 weighted average of training and testing accuracy values. This weighted average ratio weighs the testing accuracy of each model to be more important because it is an indicator of the generalizability of the student network. The student network with the highest weighted accuracy score will then be fine tuned by training the top performing student network on a weighted combination of hard targets (one hot encoded label vectors) and soft targets (temperature dampened softmax output distributions of the teacher network) in order to preserve the semantic information contained and outputted by the parent network. After fine tuning the student model to further mimic the high accuracy teacher model through knowledge distillation, another compression iteration can be initiated. This process can be repeated until an arbitrarily small student network is created. However, it is unclear what performance can be achieved by student networks after many iterations of the EPKD compression technique. An aim of this empirical study is to understand the limits imposed on students produced via this compression method.

The key to our algorithms iterative model distillation performed on the pruned student networks lies in the variable temperature value that is used to soften the output distribution of the teacher model. The current best idea for temperature application acrost compression

iterations is to decrease the applied temperature value over generations of student models. The intuition behind this temperature decrease is that lower temperature values produce higher entropy in the output distribution of the teacher model. Increasing the entropy present in the output distribution increases the information it contains. Therefore, by decreasing the applied temperature after each compression iteration, the amount of information contained within the soft targets used to train student models progressively increases, increasing the assistance provided by the teacher model as the student network decreases in size, complexity, and representational capacity.

One important aspect of our application of model distillation is that the original parent network always acts as the teacher model for student networks during distillation. This allows student models reference constant, high accuracy soft target distributions when they are being fine tuned.

Narrowing the Empirical Domain

The datasets to be used in this empirical study are:

1. MNIST
2. CIFAR-10
3. ImageNet

The teacher networks to be consumed by this algorithm will be:

1. LeNet-5
2. AlexNet
3. VGG-16
4. Inception

5. ResNet

Student networks, depending on the starting teacher network, will contain convolutional and densely connected layers.

A fixed size decrease of 90% will be imposed upon each compression iteration. This fixed size decrease determines the thresholds for the pruning techniques applied to the teacher network. The reason this constraint is used is to allow student networks to be fairly compared against their companion students during each compression iteration. If this approach is not followed, student networks of each iteration will contain a variable amount of trainable parameters and will increasingly diverge from other student networks in terms of their number of trainable parameters. This will give some student networks high perceived performance in comparison to student networks in the same iterations that may use far less parameters to perform inference.

During iterative evolutions the following pruning approaches will be applied to the teacher network:

1. https://arxiv.org/abs/1608.08710 - Pruning Filters for Efficient ConvNets.
    a. In this method of pruning, entire filters are pruned. Filters to prune are chosen by taking the L1 norm of the weights in each filter. Filters are then ranked by their norm values and the M lowest ranking filters globally, among all the layers, are removed. The network is then re-trained and the process can be repeated again.
2. https://arxiv.org/pdf/1611.06440.pdf - Pruning Convolutional Neural Networks for Resource Efficient Inference.

a. With this method of pruning a subset of weights denoted as B, are removed from then network, such that the change is cost to the network will be minimal. This method is highly effective because the absolute difference of cost is used in order to prevent the pruned network from sacrificing performance.

$$\min_{W'} \left| C(\mathcal{D}|W') - C(\mathcal{D}|W) \right| \quad \text{s.t.} \quad ||W'||_0 \leq B$$

Absolute cost function used in this technique

3. Oracle Pruning - Nvidia Research Group.
    a. The approach they used first calculates the absolute cost difference between the network with and without every weight/filter. To realistically and efficiently calculate the network cost function for each filter, a Taylor expansion of the network cost function is used. Rankings of each layer are then normalized by the L2 norm of the ranks in that layer. The use of L2 normalization greatly affects the quality of the pruning.
    b. Oracle ranking is currently one of the best methods of determining network cost change when removing weights and filters from a network. It is a brute force approach where each filter is pruned and the change to the cost function when running the modified network on the training set is observed to rank filters for removal. To measure the effectiveness of their pruning method, Nvidia had the idea of computing the Spearman correlation with the oracle rankings. Unsurprisingly, the ranking method for effective pruning that Nvidia created, is highly

correlated with the oracle. This pruning method has outperformed other methods in terms of accuracy.

4. https://arxiv.org/pdf/1810.00299.pdf - Pruned and Structurally Sparse Neural Networks
    a. This is a highly effective pruning technique that successively removes neural connections in the student networks. Neural connections can be removed using a polynomial decay based sparsity pruning method. The difference between this technique and the others provided above is that this technique does not concern itself with the filters of convolutional neural networks. Instead, individual weights are removed from all layers, including convolutional layers and densely connected layers, creating sparse neural activations between layers. This has proven to be a very effective pruning method and has outperformed many other filter removal based pruning methods in terms of accuracy.

Further Experimental Details

Experiments with this approach will first perform 10 compression iterations.

The decrease in model size between compression iterations will be determined through testing. To start, the decrease in size between networks in each iteration will be 10%. Meaning, that after each iteration in the training process, student networks will be 90% the size of the network that precedes it.

The applied temperature in each iteration will start at a very high value (50) and linearly decreased as a function of compression iteration number:

$$Temperature = 50 - (iteration * 5)$$

Note: This function will likely change and is only relevant to the current explained implementation with 10 iterations

This function will be piecewise and will never reach a temperature value of 0. If the temperature is calculated to be zero, it will automatically be set to 1.

EPKD Logged Metrics

During compression iterations the following metrics will be recorded in order to decipher the quality of compression and navigate this highly configurable optimization space (many different hyperparameters that can be modified to achieve different performances):

1. Teacher model accuracy during each epoch of training for training and test sets
2. Pruned student model accuracy on training and test sets
3. Prune student model accuracy on training and test sets during each epoch of fine tuning via knowledge distillation
4. Pruned student total number of trainable parameters, model size (megabytes), average inference time
5. The ranking of student models by weighted accuracy during each compression iteration

Future Empirical Work

After implementing EPKD as stated above, exploring different configurations of hyperparameters is of great interest. For example,