Formatting instructions for NeurIPS 2018

Adriana Salcedo
Department of Medical Biophysics
University of Toronto
Toronto, ON M5S 1A1
a.salcedo@mail.utoronto.ca

Shashwat Sharma
Edward S. Rogers Sr. Department of Electrical & Computer Engineering
University of Toronto
Toronto, ON M5S 1A1
shash.sharma@mail.utoronto.ca

Abstract

The abstract paragraph should be indented ½ inch (3 picas) on both the left- and right-hand margins. Use 10 point type, with a vertical spacing (leading) of 11 points. The word **Abstract** must be centered, bold, and in point size 12. Two line spaces precede the abstract. The abstract must be limited to one paragraph.

1 Introduction

Modern neural networks tend to be immensely large with several millions of connections and weights. The memory and energy requirements for deploying such networks is significant, and constantly growing with the complexity of state-of-the-art networks. Although having a large number of trainable parameters is beneficial to network accuracy, it was shown by Han et al. (2015), LeCun et al. (1990) and Hassibi et al. (1993) that the impact of some parameters is negligible in comparison with others. Successful identification and removal ("pruning") of such parameters would lead to smaller and faster networks, which in turn would significantly reduce the computational cost at test time. The goal of most pruning algorithms is to achieve computational advantages without an appreciable impact on accuracy. The importance and relevance of network pruning was recently highlighted via MorphNet, proposed by Gordon et al. (2018) at Google to optimize existing networks through pruning.

It was argued by Molchanov et al. (2016) that pruning can also be applied in the context of transfer learning, for generalization of a trained network to related datasets. Improved ability of networks to generalize as a result of pruning was also achieved by Liu et al. (2019). However, these approaches target networks that have been trained on a single dataset. In contrast, the concept of learning the hyperparameters of a network by using multiple datasets, known as "meta-learning", has also been proposed by Finn et al. (2017). To the best of our knowledge, the meta-learning concept has not been applied in the context of network pruning.

The above points motivate the main idea proposed in this work: designing pruning methodologies that specifically target improved generalization of a given network, by leveraging multiple datasets during training. The goal is to optimize existing networks not only to be faster and smaller, but also to improve their ability to generalize. We first implement and study some simple and commonly-used pruning strategies and analyze their impact on network size, speed and accuracy. We then discuss and implement a simple meta-learning algorithm and study its performance. Finally, we combine the

CSC2516 S 2019: project report.

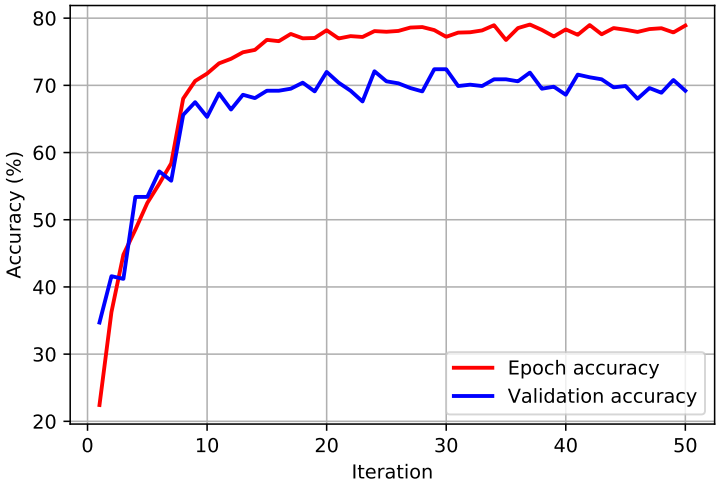

Figure 1 Baseline accuracy of VGG16 pre-trained on ImageNet. All layers were re-trained on 10% of CIFAR-10 data (4000 training images, 1000 validation images) for 50 epochs.

pruning strategies with the meta-learning algorithm to introduce a proof-of-concept "meta-pruning" methodology.

2 A Study on Simple Pruning Algorithms

2.1 Masking Classifier Layers

2.2 Masking Convolution Filters

2.3 Pruning Convolution Filters

2.3.1 Activation-Based Pruning

2.3.2 Weight-Based Pruning

********** Dumping figures here for now **********

********** Text below is copied from proposal **********

A variety of pruning algorithms have been proposed, many of which apply to convolutional neural networks. Some remove individual weights (leading to sparse weight matrices) while others remove entire filters, channels or layers. Units may be removed based on many criteria, including their magnitude, their contribution to the final loss (approximated through their gradients), and information criteria such as entropy Han et al. (2015); Molchanov et al. (2016); Luo and Wu (2017); Liu et al. (2017). Pruning may be applied towards a human-defined architecture (e.g. removing a

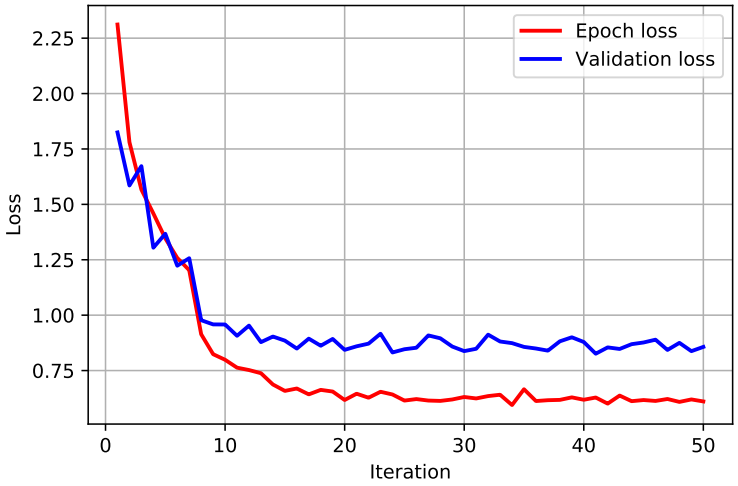

Figure 2 Baseline loss of VGG16 pre-trained on ImageNet. All layers were re-trained on 10% of CIFAR-10 data (4000 training images, 1000 validation images) for 50 epochs.

certain proportion of weights in each layer) Yu et al. (2017) or help determine the final architecture autonomously Luo et al. (2017).

3 Goals

After each dataset sample, only connections that survive pruning and thus contribute to the final loss will be retained. At the beginning of each dataset sample, weights will be initialized with the value of the updated weights in the previous sample. We hypothesize that this approach will result in weights that are important for a broad set of related tasks.

It has been suggested Liu et al. (2019) that pruning can also improve algorithm efficiency by virtue of the resulting architecture, rather than just the resulting weights. Therefore, we will also assess a modified meta-pruning algorithm that only retains the architecture, and not the weights, across datasets:

We hypothesize that this approach will result in architectures that are more generalizable and efficient than the full model. Comparing the two approaches in algorithms 1 and 2 will illuminate the relative contributions of the weights and the architecture to the value of pruning for generalizability.

4 Methods

We will first select a relatively simple pruning algorithm that can be adapted to the meta-pruning approach described above, as a proof of concept. Since pruning studies to date have largely focused on convolutional neural networks, we will develop our algorithm in the context of image recognition, initially working with the CIFAR-100 data. For the initial model, we will use VGGNet due to its simple architecture and expand to ResNet if time allows. To assess our algorithms' baseline performance, we will use test images which have the same categories as the training images. We will then assess our algorithms' capacity to generalize using a test set that contains a distinct set of categories from the training set. We will use cross entropy loss and assess the algorithms' performance by comparing its classification accuracy to the accuracy of the full model, as well as a reduced model with the same number of connections but where connections are randomly dropped. We will also compare the training time, memory requirements, and the number of retained parameters to assess the algorithms' efficiency and quantify the amount of compression achieved.

Algorithm 1 Meta-pruning for weights

$\theta \leftarrow$ *Random initialization*
for all D in collection of datasets $\mathcal{D}$ **do**
    $T \leftarrow$ *Sample tasks from dataset* D
    for all i in T **do**
        $G_i \leftarrow \nabla_{\theta,i}\mathcal{L}\left(f_{\theta,i}\right)$
        $\theta_i \leftarrow \theta_i - \beta G_i$
    $\theta \leftarrow$ *Pruned and fine-tuned* θ

Algorithm 2 Meta-pruning for architecture

for all D in collection of datasets $\mathcal{D}$ **do**
    $\theta \leftarrow$ *Random initialization*
    $T \leftarrow$ *Sample tasks from dataset* D
    for all i in T **do**
        $G_i \leftarrow \nabla_{\theta,i}\mathcal{L}\left(f_{\theta,i}\right)$
        $\theta_i \leftarrow \theta_i - \beta G_i$
    $\theta \leftarrow$ *Pruned* θ

An important consideration in the proposed algorithms is the consequence of pruning to backpropagation. Particularly in algorithm 1, depending on the order in which operations are applied, special care must be taken to ensure that gradients are computed correctly for the pruned model, at each outer iteration. One way to circumvent the issue is to recompute gradients every time the model is updated via pruning. However, it may be possible to do this more intelligently by applying advanced techniques for gradients of discrete problems. This will be an area to explore, if time permits.

5 Nice-to-haves

There are several additional analyses we can incorporate if time allows:

- To ensure our results are not specific to a particular pruning method, we can test meta-pruning using several pruning algorithms that span a range of approaches.
- Explore how to modify the meta-pruning approach to preserve connections that are retained in most, but not all training sets.
- Leverage the meta-learning framework to learn how to prune Han et al. (2015).
- Explicitly incorporate transfer learning into the meta-learning framework by assessing loss on a pseudo-test validation set with different categories Li et al. (2017).
- Apply meta-pruning to a more complex real-world dataset in a transfer learning context.

Acknowledgments

Use unnumbered third level headings for the acknowledgments. All acknowledgments go at the end of the paper. Do not include acknowledgments in the anonymized submission, only in the final paper.

References

Finn, C., Abbeel, P., and Levine, S. (2017). Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks. *CoRR*, abs/1703.03400.

Gordon, A., Eban, E., Nachum, O., Chen, B., Wu, H., Yang, T.-J., and Choi, E. (2018). MorphNet: Fast & Simple Resource-Constrained Structure Learning of Deep Networks.

Han, S., Pool, J., Tran, J., and Dally, W. J. (2015). Learning Both Weights and Connections for Efficient Neural Networks. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'15, pages 1135–1143, Cambridge, MA, USA. MIT Press.

Hassibi, B., Stork, D. G., and Wolff, G. J. (1993). Optimal Brain Surgeon and General Network Pruning. In *IEEE International Conference on Neural Networks*, pages 293–299 vol.1.

LeCun, Y., Denker, J. S., and Solla, S. A. (1990). Advances in Neural Information Processing Systems 2. chapter Optimal Brain Damage, pages 598–605. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

Li, D., Yang, Y., Song, Y., and Hospedales, T. M. (2017). Learning to Generalize: Meta-Learning for Domain Generalization. *CoRR*, abs/1710.03463.

Liu, Z., Li, J., Shen, Z., Huang, G., Yan, S., and Zhang, C. (2017). Learning Efficient Convolutional Networks through Network Slimming. *CoRR*, abs/1708.06519.

Liu, Z., Sun, M., Zhou, T., Huang, G., and Darrell, T. (2019). Rethinking the Value of Network Pruning. In *International Conference on Learning Representations*.

Luo, J. and Wu, J. (2017). An Entropy-based Pruning Method for CNN Compression. *CoRR*, abs/1706.05791.

Luo, J., Wu, J., and Lin, W. (2017). ThiNet: A Filter Level Pruning Method for Deep Neural Network Compression. *CoRR*, abs/1707.06342.

Molchanov, P., Tyree, S., Karras, T., Aila, T., and Kautz, J. (2016). Pruning Convolutional Neural Networks for Resource Efficient Transfer Learning. *CoRR*, abs/1611.06440.

Yu, R., Li, A., Chen, C., Lai, J., Morariu, V. I., Han, X., Gao, M., Lin, C., and Davis, L. S. (2017). NISP: Pruning Networks using Neuron Importance Score Propagation. *CoRR*, abs/1711.05908.