Convolutional network pruning with matrix factorization

Anonymous Author(s)
Affiliation
Address
email

Abstract

1 Introduction

Properties of convolutional networks (layers...)... convolutional layers take time, fully connected layers take space (importantly in test time). Pruning of convolutional network. Problems that can be solved with model: generalization, time and storage reduction, better interpretation. Introduction to our model, same-time matrix trifactorization on weights (on covolutional (time) and fully connected (size) layers separately). From almost keeping the performance of convolutional network to improvement of generalization.

2 Related work

For a typical convolutional neural network, about 90% of the model size is taken up by the dense connected layers and more than 90% of the running time is taken by the convolutional layers [19]. In article [4] they said that giving only a few weight values for each feature it is possible to accurately predict the remaining values while many of them do not need to be learned at all. They exploited the fact that the weights in learned networks tend to be sparse and structured. Because there is significant redundancy in the parametrization of networks, many researchers found solutions to compress them and fine-tune the compressed layers to recover the performance.

Running time complexity is depended from the computation which is dominated by convolution operations in the lower layers of the model. In contrast to model size compression, fewer approaches focused on reducing the time complexity. One of the earlier approaches of reducing the time complexity is FFT algorithm [11] which by computing the Fourier transforms of the matrices in each set efficiently performs convolutions as pairwise products. Main disadvantage of this approach is that in current implementation of the FFT algorithm, input images which are not a power of 2 must be padded to the next highest power. In newer researches they exploit the redundancy that exists between different feature channels and filters. In articles [9, 12] they use an intuition that CNN filter banks can be approximated using a low rank basis of filters that are separable in the spatial domain where in [9] substantial speedups can be achieved by also exploiting the cross-channel redundancy to perform low-rank decomposition in the channel dimension as well. Alternatively in article [5] they compressed each convolutional layer by finding an appropriate low-rank approximation with considering several elementary tensor decompositions based on singular value decompositions, as well as filter clustering methods to take advantage of similarities between learned features.

Compressing the parameters to reduce model size brings the focus upon how to compress the dense connected layers since the vast majority of weights reside in these layers which results in significant savings. Compressing the most storage demanding dense connected layers is possible by neural network pruning with low-rank matrix factorization methods [1, 14, 13]. Network pruning has been used both to reduce model size and to reduce over-fitting [7]. State-of-the-art approaches are Optimal Brain Damage [10] and Optimal Brain Surgeon [8].

Beside neural network pruning with matrix factorization other alternatives were presented where in [6], they used vector quantization methods for which they said have a clear gain over existing matrix factorization methods. Alternative is application of singular value decomposition (SVD) on the weight matrices [17]. A simple solution to reduce the model size and preserve the generalization ability is to train models that have a constant number of simpler neurons which was presented in article [3]. Replacing the fully connected layers of the network with an Adaptive Fastfood transform is introduced in article [18], and results in a deep fried convnet. The Fastfood transform allows for a theoretical reduction in computation also. However, the computation in convolutional neural networks is dominated by the convolutions, and hence the deep fried convnets are not necessarily faster in practice.

Removing all connections whose weight is lower than a threshold is introduced in [7]. There the first phase learns which connections are important and removes the unimportant ones using multiple iterations. Hashing is also an effective strategy for dimensionality reduction while preserving generalization performance [16, 15]. The strategy used on neural networks named HashedNets [2] uses a low-cost hash function to randomly group connection weights into hash buckets where all connection inside share a single and tuned parameter value.

3 Method description

Collecting the weight matrices from convolutional layers and from fully connected layers, perform pruning (description of pruning with same-time matrix trifactorization), setting the pruned weights to zero, tuning the parameters with iterations after pruning.

4 Experiments

Description and preparation of datasets, parameters, results, analysis. Datasets: mnist, imageNet, CIFAR Convnets: classical, alexnet, deep fried nets?

5 Discussion and conclusion

Acknowledgments

References

References

[1] Andrey Bondarenko and Arkady Borisov. Artificial neural network generalization and simplification via pruning. *Information Technology and Management Science*, 17(1):132–137, 2014.

[2] Wenlin Chen, James T Wilson, Stephen Tyree, Kilian Q Weinberger, and Yixin Chen. Compressing neural networks with the hashing trick. *arXiv preprint arXiv:1504.04788*, 2015.

[3] Maxwell D Collins and Pushmeet Kohli. Memory bounded deep convolutional networks. *arXiv preprint arXiv:1412.1442*, 2014.

[4] Misha Denil, Babak Shakibi, Laurent Dinh, Nando de Freitas, et al. Predicting parameters in deep learning. In *Advances in Neural Information Processing Systems*, pages 2148–2156, 2013.

[5] Emily L Denton, Wojciech Zaremba, Joan Bruna, Yann LeCun, and Rob Fergus. Exploiting linear structure within convolutional networks for efficient evaluation. In *Advances in Neural Information Processing Systems*, pages 1269–1277, 2014.

[6] Yunchao Gong, Liu Liu, Ming Yang, and Lubomir D. Bourdev. Compressing deep convolutional networks using vector quantization. *CoRR*, abs/1412.6115, 2014.

[7] Song Han, Jeff Pool, John Tran, and William J Dally. Learning both weights and connections for efficient neural networks. *arXiv preprint arXiv:1506.02626*, 2015.

[8] Babak Hassibi, David G Stork, and Gregory J Wolff. Optimal brain surgeon and general network pruning. In *Neural Networks, 1993., IEEE International Conference on*, pages 293–299. IEEE, 1993.

[9] Max Jaderberg, Andrea Vedaldi, and Andrew Zisserman. Speeding up convolutional neural networks with low rank expansions. *arXiv preprint arXiv:1405.3866*, 2014.

[10] Yann LeCun, John S Denker, Sara A Solla, Richard E Howard, and Lawrence D Jackel. Optimal brain damage. In *NIPs*, volume 89, 1989.

[11] Michael Mathieu, Mikael Henaff, and Yann LeCun. Fast training of convolutional networks through ffts. *arXiv preprint arXiv:1312.5851*, 2013.

[12] Roberto Rigamonti, Amos Sironi, Vincent Lepetit, and Pascal Fua. Learning separable filters. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 2754–2761. IEEE, 2013.

[13] Tara N Sainath, Brian Kingsbury, Vikas Sindhwani, Ebru Arisoy, and Bhuvana Ramabhadran. Low-rank matrix factorization for deep neural network training with high-dimensional output targets. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pages 6655–6659. IEEE, 2013.

[14] Jürgen Schmidhuber. Deep learning in neural networks: An overview. *Neural Networks*, 61:85–117, 2015.

[15] Qinfeng Shi, James Petterson, Gideon Dror, John Langford, Alex Smola, and SVN Vishwanathan. Hash kernels for structured data. *The Journal of Machine Learning Research*, 10:2615–2637, 2009.

[16] Kilian Weinberger, Anirban Dasgupta, John Langford, Alex Smola, and Josh Attenberg. Feature hashing for large scale multitask learning. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 1113–1120. ACM, 2009.

[17] Jian Xue, Jinyu Li, and Yifan Gong. Restructuring of deep neural network acoustic models with singular value decomposition. In *INTERSPEECH*, pages 2365–2369, 2013.

[18] Zichao Yang, Marcin Moczulski, Misha Denil, Nando de Freitas, Alex Smola, Le Song, and Ziyu Wang. Deep fried convnets. *arXiv preprint arXiv:1412.7149*, 2014.

[19] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *Computer Vision–ECCV 2014*, pages 818–833. Springer, 2014.