IMPROVED PRUNING FOR CONVOLUTIONAL NEURAL NETWORK

Dhaval Parmar

Artificial Intelligence
Indian Institute of Science

ABSTRACT

Pruning Convolutional Neural Network (CNN) is important for network compression. I have analysed the filters across layers for trained CNN model and pruned network using method proposed by Hao li et al.[1]. I have also explored cosine similarity based approach to further prune this network and also compared results with model trained from scratch.

Index Terms— Network Pruning, CNN

1. INTRODUCTION

Recently Convolutional neural network is performing very well in various tasks. But high computation and memory requirement for such model make it difficult to deploy on end devices. Han et al.[2] proposed Deep compression model that can prune the model to reduce parameter without significantly affecting model performance. In 2017, Hao li et al.[1] proposed technique for pruning filters for efficient convolution networks. I have considered VGG-16 [3] network trained on CIFAR-10 [4] data set to analyse redundancy in network. Accordingly I have pruned layers with various percentage of filters using minimum L_1 norm. I have compared results with pruned model trained from scratch. To further compress the model, I have removed filter using similarity measure.

2. TECHNICAL DETAILS

Proposed approach uses sum of absolute value $\sum |\mathbf{F}_{i,j}|$ of filter or L_1 norm to find out less important filter. As we are removing whole filter from $(i+1)^{th}$ layer it will reduce $n_i k^2 h_{i+1} w_{i+1}$ number of computations. Here n_i is number of feature map in previous layer, k is kernel size, h_{i+1} and w_{i+1} are height and width of feature map in current layer. The kernels that apply on the removed feature maps from the filters of the next convolutional layer are also removed, which saves an additional $n_{i+2} k^2 h_{i+2} w_{i+2}$ operations. For further pruning filters cosine similarity between all combination of filters can be consider to find redundant knowledge. If value is greater than threshold value of 0.8 one of the filter is pruned. Due to $O(n^2)$ computation applying cosine similarity directly on trained model is computationally heavy.

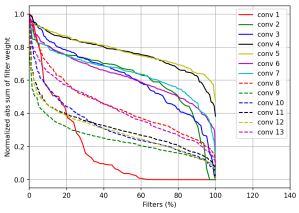

Fig. 1. Trained

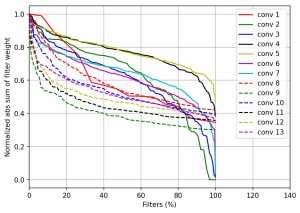

Fig. 2. Prunned

	Accuracy	Parameters	% Pruned
Trained	92.30	149919946	-
Pruned (Scratch)	91.25	3471530	76.84
Pruned	92.37	3471530	76.84
Pruned (cosine)	92.42	2694357	82.02

Table 1. Comparison Results

3. RESULTS

Figure 1 shows normalised sum of absolute value of filter weights with respect to maximum value for trained VGG network. According to graph, I have pruned conv-8 and conv-13 by 50%. Conv-1 and conv-9 to 12 was pruned by 75%. I am pruning all layers together and training them to regain accuracy. As we can see from figure 2 weights are well distributed after pruning. We can see from Table 1 that pruned model is giving comparable result with trained model and better compare to pruned model trained from scratch. So training big model and than pruning it can help in capturing better information. We can further prune upto 82.02% using cosine similarity with accuracy 92.42%.

4. CONTRIBUTIONS

I have studied vGG-16 network trained on CIFAR-10 data set. I have pruned 76.84% parameters by pruning filter with minimum sum of absolute weight without affecting accuracy. Compared performance with pruned model trained from scratch. Further I have extended pruning by removing redundant filter using cosine similarity. So I was be able to remove total 82.02% parameters without affecting results.

5. RESOURCES

- My code : https://github.com/DhavalParmar61/Prunning-CNN.git

6. REFERENCES

[1] Durdanovic I. Samet H.and Graf H.P. Li H., Kadav A., "Pruning filters for efficient convnets," ICLR, 2017.

[2] and Dally W. J. Han S., Mao H., "Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding," ICLR, 2016.

[3] Zagoruyko S., "92.45 on cifar-10 in torch," http://torch.ch/blog/2015/07/30/ cifar.html, 2015.

[4] Krizhevsky A., "Learning multiple layers of features from tiny images," 2009.