Rethinking the Value of Network Pruning

Rethinking the Value of Network Pruning

Why we need pruning?
As most of the experiments show that most of the weights in a deep learning model are reductant and so they can be removed to get a similar result with less time.

Most of the papers follow a general pruning procedure of three layer pipeline,

`Training` -> `Pruning` -> `Fine-Tuning`

Here a large over-parametrized model is trained and then its pruned to get an architecture which does not perform too well but on fine-tuning it gives comparable or better results. Previously, the researchers used to consider that the weights obtained after pruning more important than the architecture. But in this paper, the author proves that the architecture is more important as the architecture obtained after pruning can be retrained with random initialisation to obtain result somewhat better than the previous model and that too in less time.

There are two types of pruning:

1. Structured Pruning: In this method, whole channel or block is pruned. Removing whole block or channel reduces their depth and hence their computation complexity.
2. Unstructured Pruning: In this method, some of the feature maps are removed and so here the number of parameters do not decrease as in the case of structured one. So here the complexity remains as such and there is no significant speed up.

Most of the time structured pruning is preferred due to reduction in complexity but there are specific cases of unstructured also, where it performs better than its counter-part.

There are two types of structurede pruning:

1. Predefined Structured Pruning: Here the pruning ratio is predetermined
2. Automatic Structured Pruning: Here the pruning ratio is determined automatically using different methods.

- Network Slimming: Imposes L1-sparsity on channel-wise scaling factors from Batch Normalisation layers during training, and prunes channels with lower scaling factors sfterward.
- Sparse Structure Scaling: Generalisation of Network Slimming, can be used to rpune blocks, etc.

The method, that is randomly initialising the pruned network and then training, was used before also but that time researchers didn't got any promising result as they were using same number of epochs as they had used while training the previous model|(Scratch-B), while here in this paper author trained the randomly initialised model on twice the number of epochs used previously(Scratch-E), which gave better results than the non-pruned networks.

Network Pruning As Architcture Search:

As proved in this paper the architecture obtained after pruning the trained model gves much beter result than the later model. So this method can be used for architecture search which is much faster than the present method that uses Reinforcement Learning.

In the last part of the paper, the author also proposed a transfer learning approach of the pruned models where using the sparsity patters obtained from a pruned VGG-16 on CIFAR-10 to design a network for VGG-19 on CIFAR-100.