Exploring the Lottery Ticket Hypothesis with Explainability Methods: Insights into Sparse Network Performance

Explainability methods

The authors use two explainability methods to investigate the pruned networks:

Grad-CAM: A method for visualizing the importance of different pixels in an image for a given classification task.

Post-hoc concept bottleneck models (PCBMs): A method for identifying the high-level concepts that a neural network is learning.

Experiments

They prune networks to different percentages of their original size and evaluate the performance of the pruned networks on the test set.

They also use Grad-CAM and PCBMs to investigate the explainability of the pruned networks.

Results

The authors find that as more weights are pruned, the performance of the network degrades.

on the CIFAR-10 dataset, they found that pruning a network to 50% of its original size resulted in a drop in test accuracy from 93.2% to 88.4%. Pruning the network to 20% of its original size resulted in a drop in test accuracy to 73.6%.

Grad-CAM results showed that the pruned networks were focusing on different pixels in the images than the original networks.

PCBM results showed that the pruned networks were learning different high-level concepts than the original networks.

Grad-CAM:-
on the CIFAR-10 dataset, the original network was focusing on the entire image to classify an object. The pruned network, on the other hand, was focusing on a specific part of the image, such as the face of the object.

PCBM:-

on the CIFAR-10 dataset, the original network was learning concepts such as "dog", "cat", and "airplane". The pruned network, on the other hand, was learning concepts such as "face", "wheel", and "tail".

The authors' findings suggest that the LTH may not hold in all cases, especially when the network is pruned. This is because pruning can change the explainability of the network and lead to the network learning different concepts and focusing on different pixels than the original network.

This suggests that it is important to consider the explainability of the network when pruning. If the pruning process leads to the network learning different concepts and focusing on different pixels than the original network, then this could lead to a drop in performance.

What we can do

Use explainability to identify the most important weights in the network. Once we know which weights are most important, we can prune the less important weights without sacrificing too much performance.

Use explainability to monitor the network during pruning. We can use explainability to make sure that the pruning process is not leading to the network learning different concepts or focusing on different pixels than the original network. (what himakar discussed yesterday)

Use explainability to guide the pruning process.(what I proposed to himakar) We can use explainability to identify the weights that are not essential for the network to learn the target task. These weights can then be pruned without sacrificing performance.

Visualize the weights of the network at different stages of training.

Identify the weights that are not changing during training or that are becoming too specialized to the training data.

Prune these weights.

Continue training the network.

Repeat steps 2-4 until the network converges.