International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 08 Issue: 08 | Aug 2021 www.irjet.net p-ISSN: 2395-0072

Overview of Decision Tree Pruning in Machine Learning

Kavisha Ghodasara

Dept. of Computer Engineering, SKNCOE, Pune, India, affiliated to Savitribai Phule Pune University, Pune

---

Abstract - *This document serves as an introduction to the Pruning of Decision Trees. Pruning, which serves to find a sparse subnetwork in a dense network that has the same overall accuracy, helps in reducing the space requirements and cost in operating the network. An introduction to the different approaches to pruning, types of pruning has been mentioned. The Lottery Ticket Hypothesis [1] has been mentioned in order to support the advantages of pruning, along with the strategies employed to do so. Alpha-beta Pruning, which often finds its use in multiplayer gaming to determine the next best moves to be made by a machine, has also been discussed. The merits and demerits of pruning so mentioned help in determining whether pruning would be a feasible option or not.*

Key Words: Pruning, Sparse Networks, Dense Neural Networks, Horizon Effect, Lottery Ticket Hypothesis, Alpha-Beta Pruning, Artificial Intelligence

1.INTRODUCTION

Sparse models have been shown to outperform dense models. In order to find an optimal set of connections in the decision tree that would perform with the same or increased accuracy as a dense model, pruning needs to be performed. Pruning results in a sparse network, which requires less space, and runs faster, reducing the computational cost required to train the networks.

Due to the advent of technology and increasing use of Artificial Intelligence in various domains, it becomes necessary to derive alternate ways to a successful model, that aims at improving one or more parameters of the original model. While the concept of pruning might sound tedious at first, as finding a subnetwork from such a dense network would entail several complicated methods and processes, there are many relevant and efficient ways to perform pruning, while maintaining the accuracy of the network.

2.PRUNING

Pruning is a concept in deep learning, which dates back to Yann LeCun's 1990 paper Optimal Brain Damage [2]. A major advantage of pruning is that it significantly increases the ability to deploy significantly smaller and faster models, while negligently affecting the network metrics like accuracy. In some instances, it is also seen to have improved the metrics.

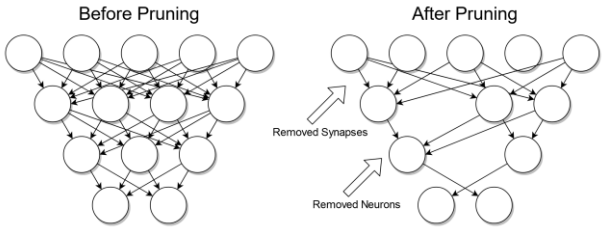

Fig -1: Before and after pruning

Pruning involves systematically removing the weight parameters in a neural network to increase the speed of the model and decrease model storage size. Typically, networks before pruning are large and dense, and simultaneously very accurate. The aim of pruning is to maintain or improve the accuracy of the network, but also decreasing the size of the network so as to make the model storage-efficient, and inexpensive as compared to the original model.

Pruning in machine learning is analogous to pruning in agriculture, which involves removing specific portions of a tree or shrub, that are of no use to the plant, such as dead and dying parts of it, or simply trimming the plant for healthy development and aesthetic purposes. The base idea of pruning in both domains involves removal of portions of the plant / network that are insignificant to the core functioning of the plant / network.

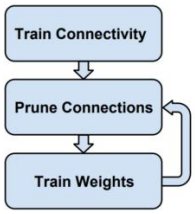

Fig -2: Three-step Training Pipeline

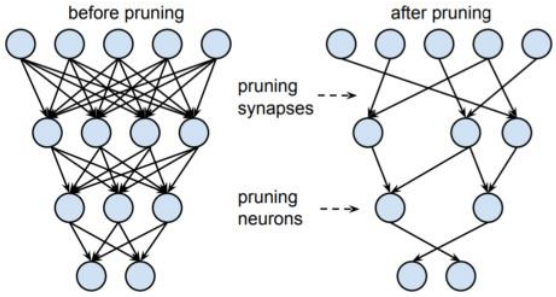

Fig -3: Synapses and neurons before and after pruning

There are two approaches to performing pruning on decision trees, which are based on the direction in which pruning occurs, namely Bottom Up Pruning, and Top Down Pruning.

2.1 Bottom Up Pruning

As the name signifies, bottom up pruning starts at the last node in the decision tree. Starting from the lowest point, it follows recursively upwards and determines the relevance of each node in the tree. The node is dropped, or replaced by a leaf node below it if the relevance for the classification is not given. A merit of this approach is that no relevant sub-trees can be lost or replaced with this method, as every node is traversed in a bottom-up manner. Methods such as Reduced Error Pruning (REP), Minimum Cost Complexity Pruning (MCCP), Minimum Error Pruning (MEP), employ this approach for pruning.

2.2 Top Down Pruning

Top down pruning starts at the root node (topmost node) of the decision tree, and traverses the child nodes from left to right. The relevance of each node is checked and it is determined whether the particular node is relevant for the classification of all the parameters or not. If the node is not relevant, it may so happen that the entire sub-tree (with the respective node as root) is dropped or replaced. This may result in relevant sub-trees being lost as the child nodes in the lost sub-tree might have a greater relevance, but since they were not traversed, they were lost solely because of non-relevance of the respective root node of the dropped sub-tree. Methods such as Pessimistic Error Pruning (PEP), employ this approach.

3. TECHNIQUES OF PRUNING

3.1 Pre-Pruning

As the root 'pre' in pre-pruning suggests, the pruning in this technique is done before the construction of the decision tree. An advantage of this is, it helps prevent overfitting in the decision tree by stopping the tree-building process early, before it produces leaf nodes with very small and irrelevant parameters. This heuristic is known as early stopping.

At each stage of splitting the tree, the cross-validation error is checked. If the error does not decrease notably, it signifies that the child nodes of the node might not have a significant impact on improving or maintaining the accuracy of the network. Hence, the tree-building can be stopped. A disadvantage of this is, early stopping may result in underfitting if stopped too early. [3]

3.2 Post-Pruning

Post-pruning involves cutting back the tree after the tree has been built. This is often used when the decision tree has a very large depth, and shows overfitting. The decision tree is generated, and then the non-significant nodes and subtrees are replaced with leaves to reduce complexity. Cross validation is performed at every step to check whether addition of a new branch results in increase or constancy of accuracy. If not, it is converted to a leaf node.

4. STRUCTURED AND UNSTRUCTURED PRUNING

Structured and unstructured pruning mainly differs on the basis of how the weights are removed. If the weights are removed together, such as channels, filters or layers, it is called structured pruning. Structured pruning has the effect of changing the input and output shapes of layers and weight matrices. [4] The disadvantages to this method are that due to the grouping, along with less relevant connections, the necessary connections in those respective channels will also need to be pruned, which might result in a loss of accuracy.

Unstructured pruning involves removal of individual weights by setting them to 0. It includes finding the less salient connections, and removing them. A disadvantage of this is that the resulting sparse networks do not have obvious efficiency gains when trained in Graphical Processing Units. Yet, this method is preferred due to the aforementioned disadvantages of structured pruning.

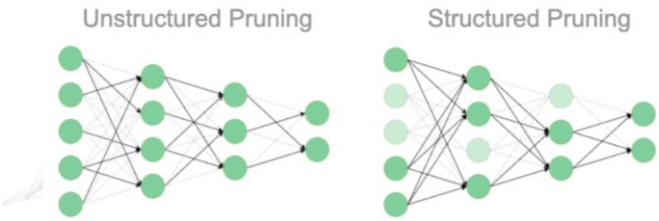

Fig -4: Unstructured and Structured Pruning

5. HORIZON EFFECT

The horizon effect is a problem in artificial intelligence where the number of possible states is huge and computers can only traverse through a small portion of them. This is problematic because if a computer traverses only through a small portion, it might so occur that it makes a detrimental move, but the effect of that move is not particularly visible because the computer does not traverse to the depth of the error. One of the disadvantages of pre-pruning is that it cannot be ascertained when a tree-building algorithm should stop, as it is difficult to tell if the addition of a one more extra node might decrease error. Hence, the optimal size of a tree cannot be determined. A feasible solution to this is to grow the tree until each node contains a small number of parameters, and then use pruning to remove nodes that are not relevant enough. [5]

6. THE LOTTERY TICKET HYPOTHESIS

The concept of pruning cannot be explained without a mention of the paper – "THE LOTTERY TICKET HYPOTHESIS: FINDING SPARSE, TRAINABLE NEURAL NETWORKS" by Jonathan Frankle and Michael Carbin. Using an analogy from the gambling world, the training of machine learning models if compared to winning the lottery by buying every possible ticket. The paper outlines the lottery ticket hypothesis as - "A randomly-initialized, dense neural network contains a subnetwork that is initialized such that—when trained in isolation—it can match the test accuracy of the original network after training for at most the same number of iterations." [1]. This concept toppled up the long-held belief that a greater number of parameters consequently results in a better performance of the network. [6]

Pruning is based on the process of removing unnecessary weights from trained, large neural networks, in order to reduce the size of the model without affecting its performance. This is the artificial intelligence equivalent of looking for and saving the winning tickets in a bag of lottery tickets, and discarding the rest of the tickets. Thus, the small network after pruning resembles a winning ticket, while the dense, large network is analogous to a bag containing all tickets. It is essential to determine the subnetwork that provides the same accuracy as the entire network.

Based on the aforementioned paragraph, some might think that it might be easier to train this smaller subnetwork rather than the bigger network, if it yields the same accuracy. But practical experiences on this topic further reveal that the subnetworks are harder to train from the start, and on training the subnetworks directly, results in a lower accuracy than the main network. Hence, the optimum way to go about it would be to train the network first, then prune it to obtain a subnetwork with similar accuracy.

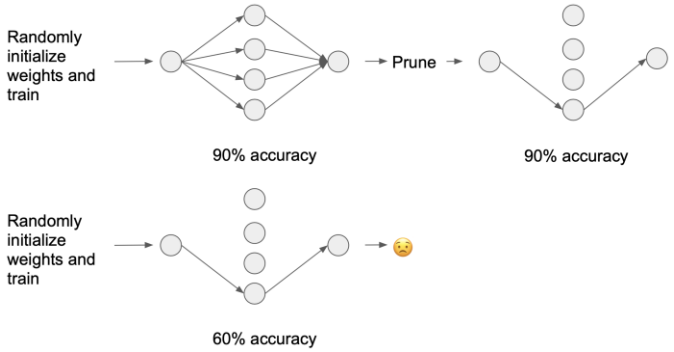

Fig -5: Training a network which has the topology of the pruned network using new random weight initialization

It was discovered that by preserving the original weight initializations from the initial dense network, a network with the topology of the pruned network can be obtained, which achieves the same or better test accuracy with the same number of iterations. [7]

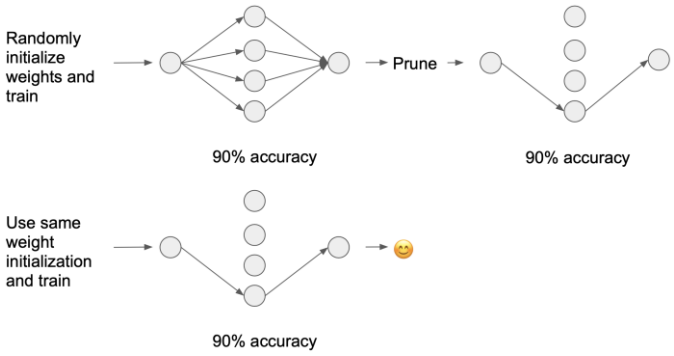

Fig -6: Training a network which has the topology of the pruned network using the original weight initialization

The process for pruning involves randomly initializing a neural network, training the network, and pruning a fraction (p%) of the network. The weights of the remaining portion of the network are reset to their initial values (the random initializations they received before the training began). [7]

According to the paper, the pruning approach can be one-shot or iterative. One-shot approach involves training the network only once, and resetting the rest of the surviving weights. If the iterative approach is used, the network is trained and pruned repeatedly, n times, with each round pruning p(1/n) % of the weights from the previous round. [1]

7. ALPHA-BETA PRUNING

Alpha-beta pruning is a type of pruning that aims to decrease the number of nodes evaluated by the minimax algorithm. This algorithm is generally used in machine playing of two-player games. [8]

The minimax algorithm is used to determine the next move in a game with n players, where n is usually two. Suppose the two players X and Y are playing a game, and the next move of X results in a direct win for X, the respective move is assigned value positive infinity. If it results in a win for Y, the move is assigned value negative infinity. Minimax is used to maximize the minimum possible gain, or minimize the maximum possible loss.

Alpha-beta pruning contains two values, alpha and beta, where alpha represents minimum score of maximizing player (let this player be 'X') and beta represents maximum score of minimizing player (let this player be 'Y'). When the minimum score of X is greater than the maximum score of Y, the subtree need not be further traversed and can be discarded. This results in decrease in traversal time and may enable the algorithm to traverse further deeper into the plausible subtrees, whose moves may prove to be beneficial for X.

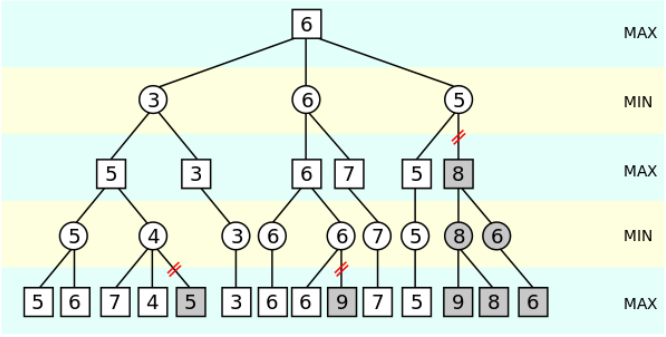

Fig -7: Illustration of alpha-beta Pruning

Fig -7 illustrates alpha-beta pruning. As shown in the image, the algorithm has determined up to next four possible moves. Starting from the bottom of the tree, at the leaf nodes, from left to right, the subtrees that will not yield a favorable result are discarded right when encountered. It will not affect the final result as discarding subtrees that yield a worse value will not negatively impact the final result.

8. TO PRUNE OR NOT TO PRUNE

8.1 Advantages of Pruning

In some cases, pruning results in faster inferences. It significantly leads to reduction in storage requirements. The sparse networks result in power savings, and reduced heat dissipation in wearable devices. It prevents the use of additional computational time, thus making the process more responsive and faster.

8.2 Disadvantages of Pruning

The impacts of pruning beyond overall accuracy cannot be accounted for. It might have undesirable or unwanted effects on the fairness, bias, robustness, and other such characteristics of the network. Pruning also requires a set depth limit, as in most cases it is not feasible to traverse the entire depth of the tree.

If the drawbacks and merits of pruning are taken into consideration, it can be observed that the merits outweigh the demerits, explaining why pruning is increasingly being used.

9. CONCLUSIONS

We have discussed about pruning, how it helps a network with lesser parameters to ensure the same accuracy. Various approaches to pruning, such as Bottom Up Pruning, Top Down Pruning have been touched upon. An introduction to The Lottery Ticket Hypothesis, how it works, and why it is beneficial has also been mentioned in this document, along with Alpha-Beta Pruning, which serves many real-life applications in todays tech-savvy world. The Horizon Effect, which is an issue in Artificial Intelligence, has also been mentioned to ponder upon ways on how it can be solved.

10. REFERENCES

[1] Jonathan Frankle, Michael Carbin - THE LOTTERY TICKET HYPOTHESIS: FINDING SPARSE, TRAINABLE NEURAL NETWORKS https://arxiv.org/pdf/1803.03635.pdf

[2] http://yann.lecun.com/exdb/publis/pdf/lecun-90b.pdf

[3] https://www.displayr.com/machine-learning-pruning-decision-trees/

[4] https://opendatascience.com/what-is-pruning-in-machine-learning/

[5] https://en.wikipedia.org/wiki/Decision_tree_pruning#Bottom-up_pruning

[6] https://www.clarifai.com/blog/neural-network-pruning-for-compression-understanding

[7] https://towardsdatascience.com/breaking-down-the-lottery-ticket-hypothesis-ca1c053b3e58

[8] https://en.wikipedia.org/wiki/Alpha%E2%80%93beta_pruning

[9] Davis Blalock, Jose Javier Gonzalez Ortiz, Jonathan Frankle, John Guttag - WHAT IS THE STATE OF NEURAL NETWORK PRUNING? https://arxiv.org/pdf/2003.03033.pdf