Magnitude Pruning

Rishabh

**MNIST with Threshold-based Magnitude Pruning**

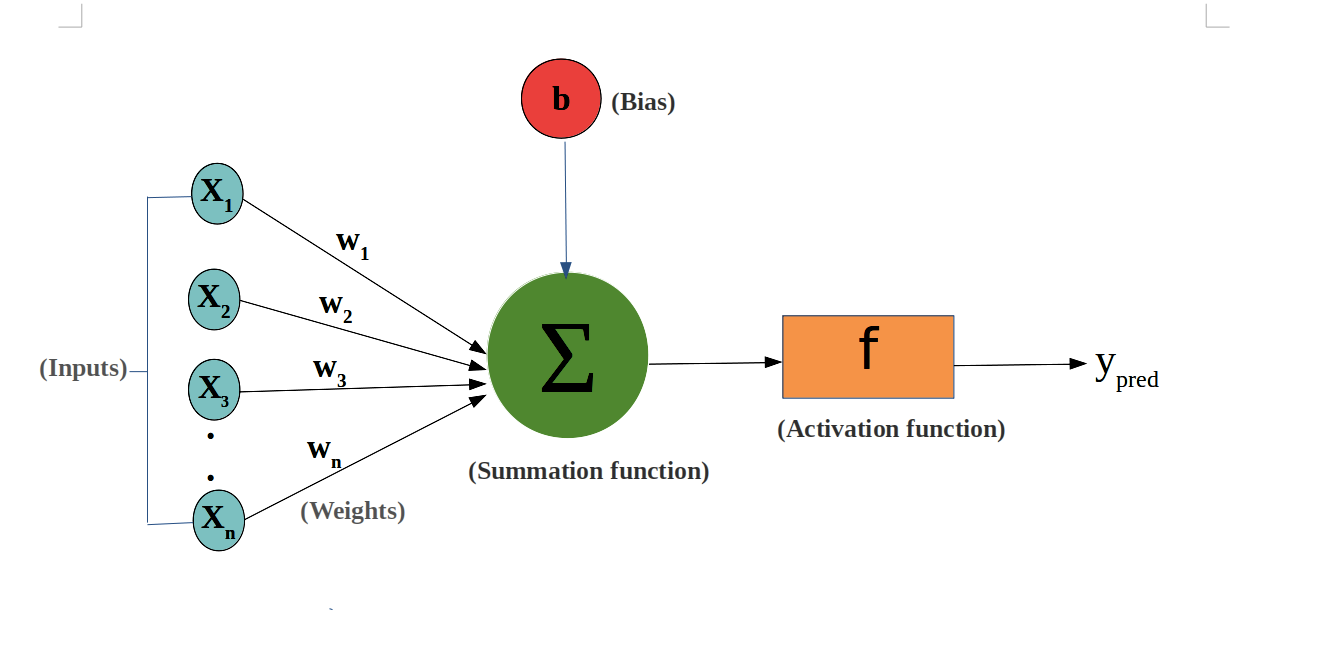
**Introduction**

This script trains a fully connected neural network using the MNIST or Fashion MNIST datasets and implements threshold-based magnitude pruning on the trained model. The script evaluates the pruned models for different threshold values and reports their accuracy as a function of the threshold value.

**Threshold based Magnitude Pruning:**

Magnitude-based pruning methods are a class of pruning techniques used in neural networks to reduce model complexity and improve efficiency. These methods involve selectively removing connections or weights from the network based on their magnitude or absolute value. The underlying principle is that small magnitude weights contribute less to the overall network performance and can be pruned without significant loss of accuracy.

One common approach in magnitude-based pruning is thresholding, where weights below a certain threshold value are pruned and set to zero. The threshold value determines the level of sparsity in the pruned model. Higher thresholds result in more aggressive pruning and higher sparsity, while lower thresholds preserve more weights and lead to lower sparsity.



For each layer, Wi=0 for |Wi|<threshold in the pruned model

Magnitude-based pruning methods offer several advantages. They are relatively simple to implement and can be applied to different layers or specific sets of weights in the network. Pruning based on magnitude also allows for weight pruning without introducing additional hyperparameters. Furthermore, magnitude-based pruning can be combined with fine-tuning techniques to regain some lost accuracy after pruning.

However, one challenge with magnitude-based pruning is the determination of an appropriate threshold value. Setting the threshold too high can result in excessive weight pruning and a significant drop in performance, while setting it too low may not lead to substantial model compression. Balancing sparsity and accuracy are a crucial aspect when using magnitude-based pruning methods.

Overall, magnitude-based pruning methods provide a practical and effective means to reduce model size, improve inference speed, and potentially reduce memory requirements, making them valuable tools for model compression and optimization in neural network applications.

**Datasets**:

- MNIST: Handwritten digits dataset with 60,000 training samples and 10,000 test samples.

- Fashion MNIST: Clothing images dataset with 60,000 training samples and 10,000 test samples.

**Network Architecture**:

- Input layer: 784 neurons (flattened 28x28 images)

- Hidden layer 1: 256 neurons with ReLU activation

- Hidden layer 2: 128 neurons with ReLU activation

- Output layer: 10 neurons with softmax activation (10 classes)

**Threshold values:**

[0.1, 0.2, 0.3, 0.5, 0.7, 1.0]

**Output:**

The script will print the accuracy of the pruned model as a function of the threshold value.

Fashion MNIST:

[['Threshold: 0.1, Accuracy: 0.7853999733924866'], ['Threshold: 0.2, Accuracy: 0.11190000176429749'], ['Threshold: 0.3, Accuracy: 0.10000000149011612'], ['Threshold: 0.5, Accuracy: 0.10000000149011612'], ['Threshold: 0.7, Accuracy: 0.10000000149011612'], ['Threshold: 1, Accuracy: 0.10000000149011612']]

MNIST:

[['Threshold: 0.1, Accuracy: 0.9413999915122986'], ['Threshold: 0.2, Accuracy: 0.11739999800920486'], ['Threshold: 0.3, Accuracy: 0.10320000350475311'], ['Threshold: 0.5, Accuracy: 0.0982000008225441'], ['Threshold: 0.7, Accuracy: 0.0982000008225441'], ['Threshold: 1, Accuracy: 0.0982000008225441']]

**Inferences:**

It can be observed from the outputs of the model that with increase in the threshold,

the accuracy of the model decreases from 94.13% at threshold of 0.1 to 11.73% at threshold of 0.2 for MNIST dataset. Note that, without pruning weights model had an accuracy of 95.77%. Setting the threshold too high in magnitude-based pruning can decrease the accuracy of the model because it leads to aggressive weight pruning, removing a large number of weights from the network. When weights with magnitudes below the threshold are pruned and set to zero, it effectively removes the corresponding connections in the neural network.

By removing connections, the pruned model loses the associated information flow and the ability to capture certain patterns or features in the data. This loss of information can result in a decrease in model performance and accuracy. The pruned model may struggle to generalize well to new, unseen data, leading to lower accuracy during inference.

Additionally, pruning a significant number of weights can disrupt the balance of the model. Neural networks are trained to find the optimal set of weights that minimize the loss function and maximize accuracy. Aggressively pruning weights by setting a high threshold can undermine this optimization process, resulting in suboptimal performance.

Hence, it is important to strike a balance when selecting the threshold value in magnitude-based pruning. A careful consideration of the trade-off between pruning magnitude and accuracy is required to ensure that the model remains accurate while achieving the desired level of compression and efficiency. Fine-tuning techniques can be applied after pruning to recover some of the lost accuracy and further refine the pruned model.