MODEL COMPRESSION PAPER REPORT

~Shreyas S (EP22B040)

Question 1. Provide a short (100-word) summary of the paper.

The paper proposes a 3-stage pipeline to compress vision neural networks using pruning, quantization, and Huffman coding. A key insight is that pruning and trained quantization do not interfere, enabling high compression. Semi-structured pruning produces a sparse model stored in CSR format, followed by K-means quantization to minimize within-cluster-sum-of-squares (WCSS). The choice of cluster initialization strongly affects compression quality, with linear initialization performing best. These methods yield significant compression and single-sample inference speedups. However, parallelism in uncompressed models diminishes speedup benefits when processing batches of test inputs.

Question 2. What did you learn from the paper?

- Pruning and trained quantization are able to compress the network without interfering each other, leading to a high compression rate.
- Pruned networks can be stored in CSR or CSC formats, requiring 2a+n+1 numbers to be stored.
- K-means clustering method: weights are quantized to k bins and hence each element in the weight matrix needs only $\log_2(k)$ bits to be stored. The k bins will still continue to store the original precision of each number that is stored. By choosing appropriately, significant compression can be made.
- Different initialization schemes of each of these bins significantly impact the final accuracy that can be achieved by the corresponding aggression of quantization.
- Huffman code – an optimal prefix code used for lossless data compression saves 20%-30% of network storage.
- Based on experiments,
  - CONV layers require more bits of precision than FC layers.
  - In the network used, FC layers did not have significant accuracy drop until 2 bits.
  - Linear initialization outperforms the density initialization and random initialization in most cases (except 3 bits). This can be attributed to the fact that linear initialization allows large weights a better chance to form a large centroid.

Question 3. What did you like about the paper?

- The paper demonstrates that pruning and trained quantization can be applied together without interfering, achieving very high compression rates while maintaining accuracy. The effects of both pruning and quantization are also individually studied and it is reported that the best compression is achieved by performing both together.
- The careful analysis of cluster initialization for K-means quantization and the use of Huffman coding show attention to implementation details that directly impact compression quality and inference efficiency.
- As highlighted in the introduction, the entire work is concerned more about reducing the model storage size more than the speedup achieved due to the compression.
    - The individual case where latency improvement can be observed is also identified, to find a practical use-case for such model compressions.

Question 4. What did you dislike about the paper?

- The paper does not provide a detailed explanation of Huffman encoding, nor does it quantify the compression benefit it contributes, making its inclusion in the pipeline less clear.
- CSR encoding is used to store the sparse matrices after pruning, but CSR is most effective at very high sparsity (>95%). Alternative methods like bitmap encoding could potentially achieve better compression for the sparsity levels observed.

Question 5. In Table 4 (AlexNet) and Table 5 (VGG-16), the authors show different compression rates for convolution and fully connected layers. If you had to decide which layer to compress most aggressively without significant accuracy loss, which one(s) would you choose and why? Justify your reasoning using the results in the paper, including the bit-width used for quantization and the effect of pruning on different layers.

The conclusions of Table 4 & Table 5 have also been described/extended to give a better visual interpretation in figures 6 & 7. It is very evident that the CONV layers require more bits of precision than FC layers.

For CONV layers, the accuracy drops significantly below 4 bits of quantization while FC layers are more robust – they are not affected until hitting 2 bits of quantization. This would also go in our favour as in the state-of-the-art object detection algorithms such a s R-CNN, upto 38% of computation time is consumed on FC layers on uncompressed models. Hence, such an aggressive quantization would not only decrease the memory footprint, but also result in a direct inference speedup.

Question 6. The proposed methodology in the paper describes pruning, quantization then Huffman coding in that order to reduce memory costs. How would you expect the results to be different if the order of operations was changed and why?

- Huffman coding relies on repeated values to achieve compression. If applied first, the network would still have all original weights, so the coding would compress less effectively.
- Quantizing all weights before pruning could obscure which connections are truly important, making pruning less accurate and potentially harming network accuracy.
  - Weight sharing may blur the significance of individual weights.
  - Hence, the seemingly less sensitive neuron maybe pruned which would impact the overall network.
- Pruning causes the weight matrix to become sparser – hence the number of clusters that would be required can also be reduced during the quantization stage.