Neural Network Compression

Osman Zubair[1] and Ted Meeds[2]

[1]Cambridge University Engineering Department & [2]Microsoft Research Cambridge

Introduction

In search of greater accuracy, neural networks have exponentially increased in size. Larger models present significant drawbacks:

- **Slower** inference
- Increased **energy** consumption
- Increased **bandwidth** usage
- More **storage** required
- Unable to run inference on mobile devices
- Data transfer to cloud increases **privacy** concerns

Our project attempts to compress models with minimal effect on inference accuracy.

Method

1. Gaussian Mixture Prior on Parameters: Adding a prior over the weights will cluster the weights for pruning and quantization.

$$\mathcal{L}(y_{\hat{T},L}, y_{\hat{S},L}, \mathbf{w}, \{\mu_j, \sigma_j, \pi_j\}_{j=0}^{J}) = \underbrace{-\frac{1}{n}(\hat{y}_{S,L} - \hat{y}_{T,L})^2}_{\text{MSE Loss}} - \tau \underbrace{\sum_{i=1}^{I} \log \sum_{j=0}^{J} \pi_j \mathcal{N}(w_i|\mu_j, \sigma_j^2)}_{\text{Gaussian Mixture Prior Loss}} \quad (1)$$

- MSE loss ensures the retraining remains accurate
- Gaussian Mixture Prior on parameters forces weights to cluster
- 0-mean parameter clusters are pruned and the remaining quantized to their means
- Trade-off hyperparameter τ balances accuracy and compression

2. Teacher-Student Training: Use the predictions of a fully trained "teacher" network as output labels to train a "student" network can allow a smaller network to mimic a more powerful network.

$$\hat{y}_{Ti} = \frac{exp(z_i/T)}{\Sigma_j exp(z_j/T)} \quad (2)$$

- Temperature parameter T softens output softmax distribution
- Mean squared error used as loss function to match smoothed softmax distributions
- A smaller or less parametrized network can learn to mimic a larger network

3. Layer-wise Distillation: Each layer is trained separately and the teacher network mimicked layer-wise.

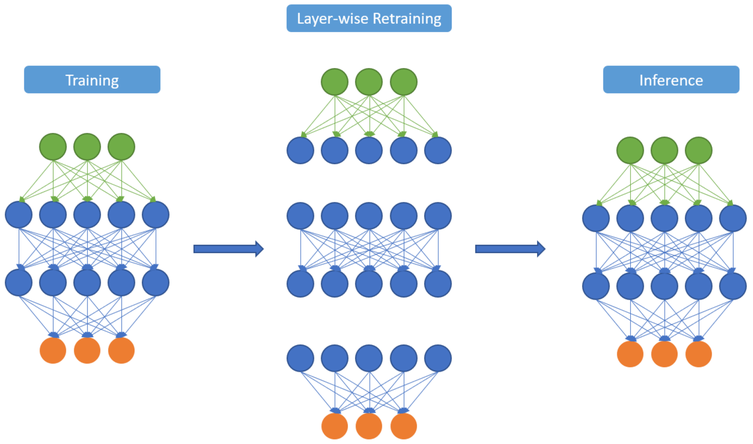


Figure 1: Layer-wise Training

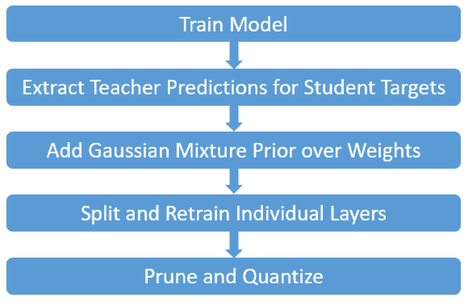


Figure 2: Compression Pipeline

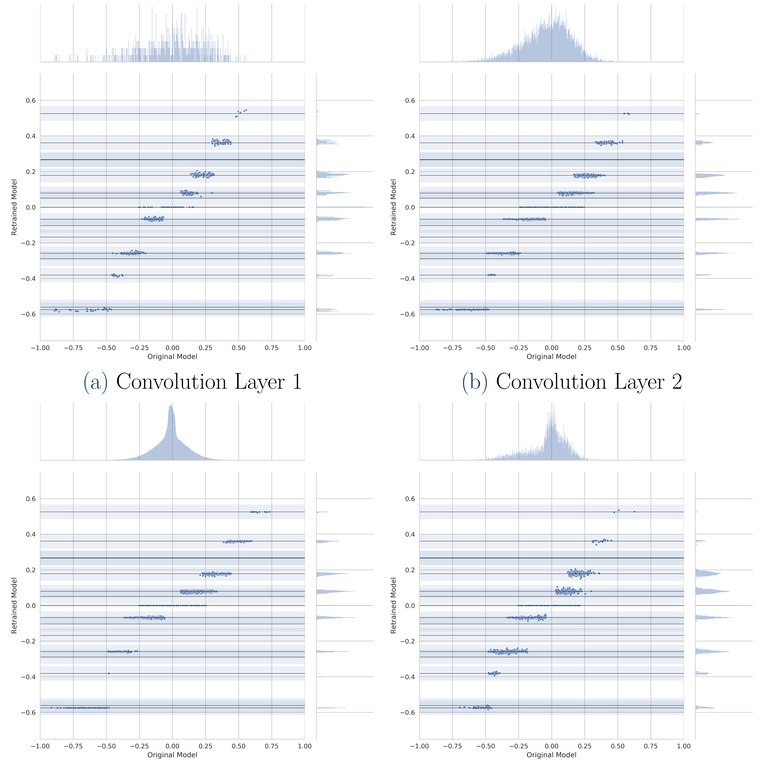
Results

Layer	Shape	Parameters	Sparsity
Convolution	(1, 25, 5, 5)	650	22.7%
Convolution	(25, 50, 3, 3)	11300	54.5%
Dense	(500, 1250)	625500	93.4%
Dense	(10, 500)	5010	60.7%
Total		642460	**92.3%**

Table 1: MNIST Classifier Sparsity

	Original	Retrained	Pruned
Accuracy	98.79%	98.59%	**98.14%**

Table 2: MNIST Classifier Accuracy



(a) Convolution Layer 1 (b) Convolution Layer 2

(c) Dense Layer 1 (d) Dense Layer 2

Figure 3: Retraining Clustering

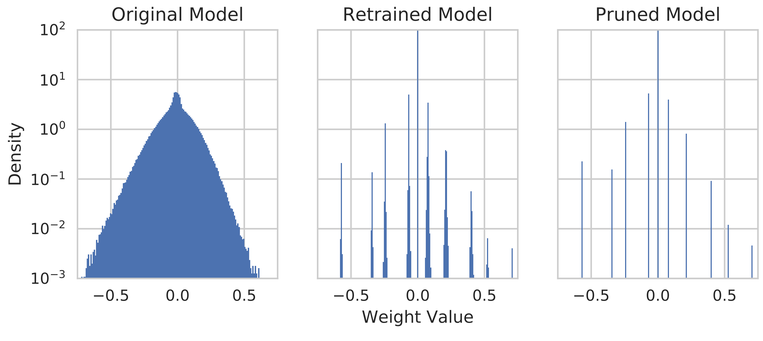


Figure 4: Model Visualization

Conclusion

- Model size can be substantially reduced without a significant impact on accuracy
- Implementing such methods will save energy, costs, time and potentially allow for new applications