Neural Network Compression
Osman Zubair¹ and Ted Meeds²
¹Cambridge University Engineering Department & ²Microsoft 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.

\[
\mathcal{L}(\hat{y}_T, L, \hat{y}_S, L, \{\mu_j, \sigma_j, \pi_j\}_{j=0}^J) = \frac{1}{n} (\hat{y}_S - \hat{y}_T)^2 + \tau \sum_{j=0}^J \log \sum_{j=0}^J \pi_j \mathcal{N}(w_i | \mu_j, \sigma_j^2)
\]

- 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 \( \tau \) 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.

\[
\hat{y}_T = \frac{\exp(z_i/T)}{\sum_j \exp(z_j/T)}
\]

- 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.

Results

<table>
<thead>
<tr>
<th>Layer</th>
<th>Shape</th>
<th>Parameters Sparsity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Convolution</td>
<td>(1, 25, 5, 5)</td>
<td>650</td>
</tr>
<tr>
<td>Convolution</td>
<td>(25, 50, 3, 3)</td>
<td>11300</td>
</tr>
<tr>
<td>Dense</td>
<td>(500, 1250)</td>
<td>625500</td>
</tr>
<tr>
<td>Dense</td>
<td>(10, 500)</td>
<td>5010</td>
</tr>
<tr>
<td>Total</td>
<td></td>
<td>642460</td>
</tr>
</tbody>
</table>

Table 1: MNIST Classifier Sparsity

![Figure 1: Layer-wise Training](image1.png)

![Figure 2: Compression Pipeline](image2.png)

![Figure 3: Retraining Clustering](image3.png)

![Figure 4: Model Visualization](image4.png)

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