On the Information of Feature Maps and Pruning of Deep Neural Networks

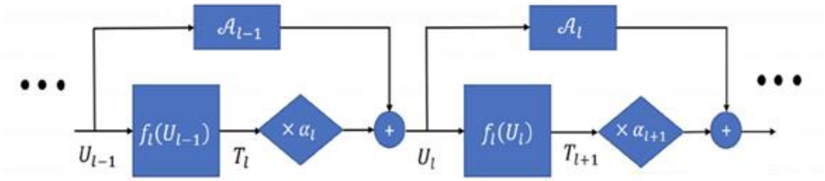
Mohammadreza Soltani

Duke University

Compressing of Deep Neural Networks:

- Deep Neural Networks (DNNs) are intensive in terms of **computation** and **memory requirement.**
- Deploy DNNs to embedded systems with limited hardware resources is challenging.
- One needs to compress a DNN by pruning the weights or the neurons of a deep model.
- We focus on ResNet-type architectures.
- ResNet-type architectures are the core of many modern deep models.
- Our pruning strategy is to remove the redundant residual units instead of individual neurons or weights.

ResNet Architecture



Two consecutive residual units in ResNet architecture.

$$U_l = \Psi(T_l, U_{1:l-1}, \alpha_l) = \alpha_l T_l + \mathcal{A}_{l-1} U_{l-1}, l = 1, .. \text{L}$$

- T_l denotes a set of operations (convolution, pooling, etc) on U_{l-1}.
- $\alpha_l \in \{0,1\}$
- $\mathcal{A}_{l-1}$ is an identity or a convolution operator

Pruning Less Important Residual Units:

- Pruning a model by removing the redundant residual units based on their learned information
- Need to measure the information between the residual units and the output of the model
  - ✓ Mutual Information as a natural choice
- Clustering the units based on their mutual information
- Keeping only the cluster heads ($\alpha = 1$)
- Removing the other units in each cluster from the graph of the model ($\alpha = 0$)

Estimating the Mutual Information Using GMM:

$$\begin{aligned} I(T;Y) &= H(T) - H(T|Y) \\ &\leq -\frac{1}{n}\sum_{i=1}^{n}\ln\frac{1}{n}\sum_{j=1}^{n}\exp\Big(-\frac{\|\mu_j-\mu_i\|_2^2}{2\sigma^2}\Big) \\ &-\sum_{k=0}^{l-1}p_k\Big(-\frac{1}{n_k}\sum_{\substack{i=1\\ y_i=k}}^{n}\ln\frac{1}{n_k}\sum_{\substack{j=1\\ y_j=k}}^{n_k}\exp\Big(-\frac{1}{4}\frac{\|\mu_j-\mu_i\|_2^2}{2\sigma^2}\Big)\Big), \end{aligned}$$

Multi-Stage Pruning with Information Clustering:

1. Train a network
2. Measure the energy of the residual units
3. Cluster the units with similar energy
4. Keep the cluster head and remove the other units
5. Retrain the network with the weights from the previous stage
6. Repeat this process for multiple stages

Some Results (Classification of CIFAR-10) Dataset

DenseNet-100 (full)	Test Accuracy (0.9531)	Param. (M) (0.77)
CondenseNet	0.9496	0.52
Ours	**0.9437**	**0.29**