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

Mohammadreza Soltani

Duke University

Table of contents

Introduction

Motivation and Problem Statement

- Energy consumption for limited-resource embedded systems
- Very large Memory for saving the weights of a model
- Huge amount of computation for product operation
- For example, under 45nm CMOS technology,
  - A 32bit floating point add consumes 0.9pJ
  - A 32bit SRAM cache access requires 5pJ
  - A 32bit DRAM memory access takes 640pJ
- Running a 1 billion connection neural network, for example, at 20 fps needs almost 13W power just for DRAM!!!
- Need to compress model for deployment and fast inference running-time

An important observation

- Robustness of deep architectures with *skip-connection* against coarse pruning
  - Removing a random layer doesn't hurt the performance.
  - Removing the models without *skip-connection* drops the performance dramatically.
- Our focus is to investigate this phenomena in more depth
- Studying two prominent examples of models with *skip-connection*: Resnet and DenseNet

- A *skip-unit* is defined as a set of layers, and each layer consists of sequential operations including *Conv*, *Pooling*, *ReLU*, *BN*, *Dropout*, etc.
- A *skip-units* is mathematically defined as

$$U_\ell = \Psi(T_\ell, U_{1:\ell-1}, \alpha_\ell), \quad \ell = 1, 2, \ldots, L,$$

  - $U_{1:\ell-1}$, the input of ℓ-th unit
  - $T_\ell = f_\ell(U_{\ell-1})$, the output in the skip-unit
  - f_ℓ, the composition of aforementioned operations
- α_ℓ's are binary variables and Ψ denotes an operation that combines T_ℓ and $U_{1:\ell-1}$.

- ResNet architecture Ψ_{res} and DenseNet architecture Ψ_{den} are respectively given by:

$$U_\ell = \Psi_{res}(T_\ell, U_{\ell:\ell-1}, \alpha_\ell) = \alpha_\ell T_\ell + \mathcal{A}_{\ell-1} U_{\ell-1},$$
$$U_\ell = \Psi_{den}(T_\ell, U_{\ell:\ell-1}, \alpha_\ell) = Concat(\alpha_\ell T_\ell,\ U_{1:\ell-1}),$$

- *Concat* is the concatenation operation.
- $\mathcal{A}_{\ell-1}$ is an identity or a convolution operator.

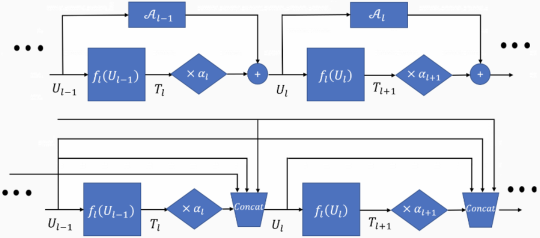

Figure 1: Two consecutive skip-units in a ResNet (top) and DesNet (bottom) family, respectively.

Compressing skip-units models: Pruning the model by removing the redundant skip-units based on their learned information.

1. How to study the learned features?
2. How to capture the information in the learned features?
3. How to quantify the redundant the skip-units?

Proposed Solution, Information Bottleneck

Mutual Information as a measure of information of skip-units:

- Measuring the mutual information between the skip-units and the output of the model
- 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

Upper Bound on the Mutual Information

- Following the method of Kolchinsky, et al., 2017, an upper bound is used for estimating the mutual information.
- Modeling the underlying probability distribution with a Gaussian mixture model with the number of components equals to the training samples.

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

- $P_k = \frac{n_k}{n}$ denotes the probability of class k and $n_k = \sum_{i=1}^{n} \mathbb{I}(y_i = k)$.
- μ_i is the mean of each Gaussian component and σ is set to a small number.

Proposed Pruning Algorithm

Algorithm 1

INPUT:

DNN^0: Pre-trained Deep Neural Network
S^0: The index set of units
T_l: Feature maps, $l = 1, 2, \ldots, |S^0|$
N: Number of stages
R^t: Resolution vector, $t = 0, 1, \ldots, N-1$
ϵ^t: Error threshold, $t = 0, 1, \ldots, N-1$

for $t = 0, 1, \ldots, N-1$ **do**
  Compute $I(T_l^t, Y)$, $l = 1, \ldots, |S^t|$ using **DNN**^{t}
  Construct $\mathbf{I}^t = [I(T_1^t; Y), I(T_2^t; Y), \ldots, I(T_{|S|}^t; Y)]$
  $\{Cluster_1^{policy_1}, \ldots, Cluster_{M^t_{policy_1}}^{policy_1}, \ldots, Cluster_{M^t_{policy_{|R^t|}}}^{policy_{|R^t|}}\} = \mathrm{Cluster}(R^t, \mathbf{I}^t, S^t)$
  for policy **in** R^t **do**
    for $j = 1, 2, \ldots, M^t_{policy}$ **do**
      $a_l = 1, \quad l = \min_{k \in Cluster_j^{policy}} k$
      $a_u = 0, \quad \forall u \in Cluster_j^{policy} \setminus l$
    end for
    Compute $Test^t_{error}$ for given policy
  end for
  Select one policy with $Test^t_{error} < \epsilon^t$
  $S^{t+1} \leftarrow S^t \setminus \{v : a_v = 0, v \in S\}$
  DNN^{t+1} $\leftarrow$ Re-train the resulted sub-network with the trained weights in stage t
end for
Return pruned model with $|S^{N-1}|$ units

Experimental Results

Datasets and Models

1. Datasets.

Dataset	Train data	Test data	Image Size	Classes
CIFAR-10	50000	10000	$32 \times 32 \times 3$	10
CIFAR-100	50000	10000	$32 \times 32 \times 3$	100
Tiny ImageNet	100000	10000	$64 \times 64 \times 3$	200

2. Model architectures:

Model	Units	Layers	Param. (M)	FLOPs (M)
ResNet-56	$[9, 9, 9]$	56	0.85	126.54
ResNet-164	$[18, 18, 18]$	164	1.70	254.94
DenseNet-100	$[6, 12, 24, 16]$	100	0.80	305.10

Experiment of pruning DNNs on CIFAR-10 dataset

Model	Test Accuracy	Param. (M)	FLOPs (M)	Red.(%)
ResNet-56 (full)	0.9334	0.85	126.55	-
ResNet-56	0.9128	0.23	42.30	72.72
ResNet-164 (full)	0.9569	1.70	254.94	-
ResNet164 (t=2)	0.9207	0.99	143.90	41.53
ResNet164 (t=7)	0.9173	0.47	112.50	72.02
DenseNet-100-k12 (full)	0.9531	0.77	293.55	-
DenseNet100-k12 (t=3)	0.9352	0.34	224.90	56.27
DenseNet100-k12 (t=13)	0.9437	0.29	173.20	62.66

Table 1: The results of pruning various DNNs on CIFAR-10 data.

Experiment of DNNson CIFAR-100 dataset

Model	Test Accuracy	Param. (M)	FLOPs (M)	Red.(%)
ResNet-164 (full)	0.7799	1.73	254.96	-
ResNet164	0.7459	0.94	130.97	45.30
DenseNet-100-k12 (full)	0.7793	0.80	304.10	-
DenseNet100-k12	0.7408	0.38	203.35	52.37

Table 2: The results of pruning various DNNs on CIFAR-100 data.