Compressing Deep Networks Using Fisher Score of Feature Maps

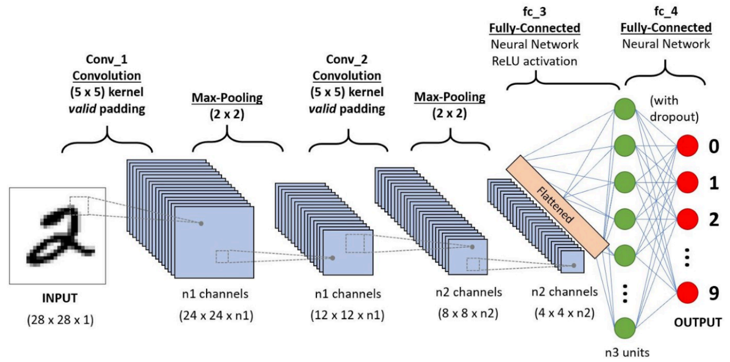
Mohammadreza Soltani

Duke University

Table of contents

Introduction

Deep Neural Networks



A CNN sequence to classify handwritten digits

- Source: google image.

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

- 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 phenomenon in more depth
- Studying two prominent examples of models with *skip-connection*: Resnet and DenseNet

DNNs with Skip-units

- 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.

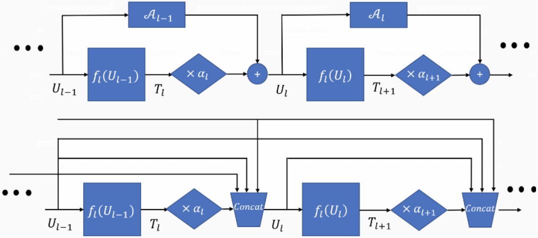


Figure 1: Two consecutive skip-units in a ResNet (top) and DesNet (bottom) family, respectively.

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

Shannon Mutual Information? Mutual Information is the measure of informativeness. However,

- Computationally challenging task to estimate Mutual Information in a high-dimensional feature space
- Proper assumption on the underlying probability distribution
- Using the gradient information instead of mutual information for measuring the information learned in the intermediate layers of a deep model
- The same properties of the Mutual Information; however, computationally more efficient

After computing the information of units:

- Clustering the units based on their gradient information
- Keeping only the cluster heads ($\alpha = 1$)
- Removing other units in each cluster ($\alpha = 0$)

Gradient Information

Fisher Divergence

- The gradient information is proposed based on the Hyvarinen loss with respect to an input x with density function p which is given by [Hyvarinen, 2005.]:

$$s_H(x, p) = \frac{1}{2} \left\| \nabla_x \log p(x) \right\|^2 + \Delta_x \log p(x),$$

  where ∇ denotes the gradient and Δ denotes the Laplacian.

- The expectation of the Hyvarien loss of a probability density $q(\cdot)$ w.r.t the another distribution $p(\cdot)$ can be reformulated as: $\mathbb{E}_p \left\{ s_H(x, q) \right\} = D_{\mathrm{F}}(p, q) - \frac{1}{2}\mathbb{E}_p \left\| \nabla_y \log q(x) \right\|^2$, where $D_{\mathrm{F}}(p, q)$ is the Fisher divergence given by:

$$D_{\mathrm{F}}(p, q) = \frac{1}{2} \int_{\mathbb{R}^d} \left\| \nabla_x \log q(x) - \nabla_x \log p(x) \right\|^2 p(x) dx.$$

Fisher Score

Definition (Gradient information [Ding et al., 2019.])

Consider continuous random variables T and Y with marginal density functions p_T and p_Y, respectively as well as the joint density p_{TY}. The information quantity is defined as $(T, Y) = D_F(p_{TY}, p_T p_Y)$.

Definition (Fisher score)

Given the random variables $T \in^d$ and $Y \in \mathcal{Y}$, with $\mathcal{Y} = \{1, 2, \cdots, p\}$, the Fisher score between T and Y is defined as

$$F(T, Y) = \max_{1 \leq i,j \leq p} D_F(p_i, p_j),$$

where p_i denotes the densities of T conditional on $Y = i$, and D_F denotes the gradient information in the above definition.

Proposed Pruning Algorithm

Algorithm 1

INPUT:

- DNN^0: Pre-trained Deep Neural Network
- S^0: The index set of skip-units in DNN^0
- T_l: Feature maps, $l = 1, 2, \dots, |S^0|$
- K^t: Cluster vector, $t = 0, 1, \dots, N-1$
- N: Number of stages

for $t = 0, 1, \dots, N-1$ **do**

 Compute Fisher scores, $F(T_l^t, Y)$, $l = 1, \dots, |S^t|$ using DNN^t

 Construct $\mathbf{F}^t = [F(T_1^t, Y), F(T_2^t, Y), \dots, F(T_{|S^t|}^t, Y)]$

 $\{Cluster_1^{K_1^t}, \dots, Cluster_{K_1^t}^{K_1^t}, \dots, Cluster_1^{K_{|K^t|}^t}, \dots, Cluster_{K_{|K^t|}^t}^{K_{|K^t|}^t}\} = \text{Clustering}(K^t, \mathbf{F}^t, S^t)$

 for k **in** K^t **do**

  for $j = 1, 2, \dots, k$ **do**

   $a_c = 1, \quad c = \text{cluster centroid index}$

   $a_u = 0, \quad \forall u \in Cluster_j^k \setminus c$

  end for

  Compute $Train_{err}^k$ for given k

 end for

 Select $k_*^t =_{k \in K^t} Train_{err}^k$

 Update S^t by keeping only k_*^t units and remove the rest of units

 Update DNN^t by re-training the model with $k_*^t = |S^t|$ units with weights initialized in stage t

end for

Return pruned model with $|S^{N-1}|$ active skip-units

Experimental Results

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
SVHN	73257	26032	$32 \times 32 \times 3$	10

2. Model architectures (based on CIFAR-10 data set):

Model	Units	Layers	Param. (M)	FLOPs (M)
ResNet-56	[9, 9, 9]	56	0.85	126.55
ResNet-110	[18, 18, 18]	164	1.73	254.00
DenseNet-100	[16, 16, 16]	100	0.77	296.50

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 (N=7)	0.9331	0.21	48.15	74.89
ResNet-110 (full)	0.9387	1.73	254.00	-
ResNet-110 (N=6)	0.9379	0.31	94.68	81.95
DenseNet-100-k12 (full)	0.9559	0.77	296.50	-
DenseNet100-k12 (N=7)	0.9470	0.36	181.72	52.74

Table 1: The results of pruning various DNNs on CIFAR-10 data set. Red (%) has been calculated in terms of number of parameters.

Experiment of DNNs on CIFAR-100 dataset

Model	Test Accuracy	Param. (M)	FLOPs (M)	Red.(%)
ResNet-56 (full)	0.9334	0.86	127.00	-
ResNet-56 (N=6)	0.7134	0.36	61.55	58.57
ResNet-110 (full)	0.7289	1.74	255.00	-
ResNet-110 (N=7)	0.7300	0.50	123.26	71.26

Table 2: The results of pruning various DNNs on CIFAR-100 data set. Red (%) has been calculated in terms of number of parameters.