Rate Distortion for Model Compression: From Theory To Practice

Weihao Gao[*], Yu-Han Liu[†], Chong Wang[‡] and Sewoong Oh[§]

[*]UIUC, [†]Google, [‡]Bytedance, [§]Univ of Washington

June 10, 2019

Motivation

- Nowadays, neural networks become more and more powerful

Motivation

- Nowadays, neural networks become more and more powerful
- Also, neural networks become larger and larger
  - LeNet 40K, AlexNet 62M, BERT 110M(base)/340M(large)

Motivation

- Nowadays, neural networks become more and more powerful
- Also, neural networks become larger and larger
    - LeNet 40K, AlexNet 62M, BERT 110M(base)/340M(large)
- Compression of models are necessary for saving
    - training and inference time
    - storing space, e.g., for mobile Apps

Motivation

- Nowadays, neural networks become more and more powerful
- Also, neural networks become larger and larger
  - LeNet 40K, AlexNet 62M, BERT 110M(base)/340M(large)
- Compression of models are necessary for saving
  - training and inference time
  - storing space, e.g., for mobile Apps

Two fundamental questions about model compression

Motivation

- Nowadays, neural networks become more and more powerful
- Also, neural networks become larger and larger
  - LeNet 40K, AlexNet 62M, BERT 110M(base)/340M(large)
- Compression of models are necessary for saving
  - training and inference time
  - storing space, e.g., for mobile Apps

Two fundamental questions about model compression

1. Is there any *theoretical* understanding of the *fundamental limit* of model compression algorithms?

Motivation

- Nowadays, neural networks become more and more powerful
- Also, neural networks become larger and larger
  - LeNet 40K, AlexNet 62M, BERT 110M(base)/340M(large)
- Compression of models are necessary for saving
  - training and inference time
  - storing space, e.g., for mobile Apps

Two fundamental questions about model compression

1. Is there any *theoretical* understanding of the *fundamental limit* of model compression algorithms?
2. How can theoretical understanding help us to improve *practical* compression algorithms?

Fundamental limit for model compression

- Trade-off between *compression ratio* and *quality* of compressed model

Fundamental limit for model compression

- Trade-off between *compression ratio* and *quality* of compressed model

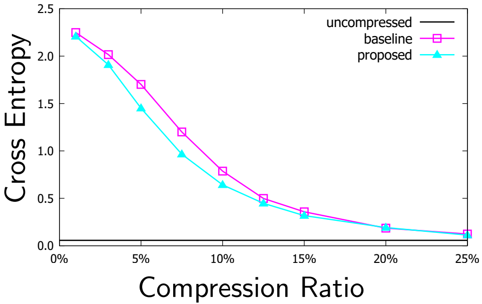

Figure 1: Trade-off between compression ratio and cross entropy loss

Fundamental limit for model compression

- Trade-off between *compression ratio* and *quality* of compressed model

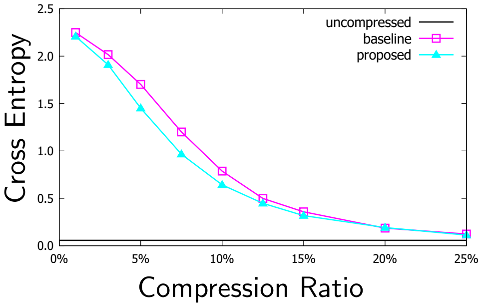

Figure 1: Trade-off between compression ratio and cross entropy loss

- **Fundamental question:** Given a pretrained model $f_w(x)$, how well can we compress the model, given certain ratio?

Rate distortion for model compression

- We bring the tool of *rate distortion theory* from information theory

Rate distortion for model compression

- We bring the tool of *rate distortion theory* from information theory
- **Rate**: average number of bits to represent parameters

Rate distortion for model compression

- We bring the tool of *rate distortion theory* from information theory
- **Rate**: average number of bits to represent parameters
- **Distortion:** difference between compressed model and original model

Rate distortion for model compression

- We bring the tool of *rate distortion theory* from information theory
- **Rate**: average number of bits to represent parameters
- **Distortion:** difference between compressed model and original model
  - For regression $d(w, \hat{w}) = \mathbb{E}_X[\|f_w(X) - f_{\hat{w}}(X)\|^2]$
  - For classification $d(w, \hat{w}) = \mathbb{E}_X[D_{KL}(f_{\hat{w}}(X)||f_w(X))]$

Rate distortion for model compression

- We bring the tool of *rate distortion theory* from information theory
- **Rate**: average number of bits to represent parameters
- **Distortion:** difference between compressed model and original model
  - For regression $d(w, \hat{w}) = \mathbb{E}_X[\|f_w(X) - f_{\hat{w}}(X)\|^2]$
  - For classification $d(w, \hat{w}) = \mathbb{E}_X[D_{KL}(f_{\hat{w}}(X)||f_w(X))]$
- Rate-distortion theorem for model compression

$$R(D) = \min_{P_{\hat{W}|W}:\mathbb{E}[d(W,\hat{W})]\leq D} I(W; \hat{W})$$

Our contributions

- Generally, it is intractable to evaluate $R(D)$ due to
  - High dimensionality of parameters
  - Complicated non-linearity

Our contributions

- Generally, it is intractable to evaluate $R(D)$ due to
  - High dimensionality of parameters
  - Complicated non-linearity
- In this talk, our contributions are

Our contributions

- Generally, it is intractable to evaluate $R(D)$ due to
  - High dimensionality of parameters
  - Complicated non-linearity
- In this talk, our contributions are
  - For linear regression model, we give a lower bound of $R(D)$ and give an algorithm achieving the lower bound

Our contributions

- Generally, it is intractable to evaluate $R(D)$ due to
  - High dimensionality of parameters
  - Complicated non-linearity
- In this talk, our contributions are
  - For linear regression model, we give a lower bound of $R(D)$ and give an algorithm achieving the lower bound
  - Inspired by the optimal algorithm, we propose two “golden rules” for model compression

Our contributions

- Generally, it is intractable to evaluate $R(D)$ due to
  - High dimensionality of parameters
  - Complicated non-linearity
- In this talk, our contributions are
  - For linear regression model, we give a lower bound of $R(D)$ and give an algorithm achieving the lower bound
  - Inspired by the optimal algorithm, we propose two "golden rules" for model compression
  - We prove the optimality of proposed "golden rules" for one layer ReLU network

Our contributions

- Generally, it is intractable to evaluate $R(D)$ due to
  - High dimensionality of parameters
  - Complicated non-linearity
- In this talk, our contributions are
  - For linear regression model, we give a lower bound of $R(D)$ and give an algorithm achieving the lower bound
  - Inspired by the optimal algorithm, we propose two "golden rules" for model compression
  - We prove the optimality of proposed "golden rules" for one layer ReLU network
  - We show that the algorithm following "golden rules" performs better in real models

Linear regression

- Consider linear regression model $f_w(x) = w^T x$

Linear regression

- Consider linear regression model $f_w(x) = w^T x$ and the following assumptions
  - Weights W are drawn from $\mathcal{N}(0, \Sigma_W)$
  - Data X has zero mean and $\mathbb{E}[X_i^2] = \lambda_{x,i}$, $\mathbb{E}[X_i X_j] = 0$.

Linear regression

- Consider linear regression model $f_w(x) = w^T x$ and the following assumptions
  - Weights W are drawn from $\mathcal{N}(0, \Sigma_W)$
  - Data X has zero mean and $\mathbb{E}[X_i^2] = \lambda_{x,i}$, $\mathbb{E}[X_i X_j] = 0$.
- Theorem: the rate distortion function is lower bounded by:

$$R(D) \geq \underline{R}(D) = \frac{1}{2} \log \det(\Sigma_W) - \sum_{i=1}^{m} \frac{1}{2} \log(D_i),$$

where

$$D_i = \begin{cases} \mu / \lambda_{x,i} & \text{if} \mu < \lambda_{x,i} \mathbb{E}_W[W_i^2] \,, \\ \mathbb{E}_W[W_i^2] & \text{if} \mu \geq \lambda_{x,i} \mathbb{E}_W[W_i^2] \,, \end{cases}$$

where μ is chosen that $\sum_{i=1}^{m} \lambda_{x,i} D_i = D$.

- The lower bound is **tight** for linear regression.

From theory to practice

- Two "golden rules" of the optimal compressor
  1. Orthogonality: $\mathbb{E}_{W,\hat{W}}[\hat{W}^T \Sigma_X (W - \hat{W})] = 0$
  2. Minimization: $\mathbb{E}_{W,\hat{W}}[(W - \hat{W})^T \Sigma_X (W - \hat{W})]$ should be minimized, given certain rate.

From theory to practice

- Two "golden rules" of the optimal compressor
  1. Orthogonality: $\mathbb{E}_{W,\hat{W}}[\hat{W}^T \Sigma_X (W - \hat{W})] = 0$
  2. Minimization: $\mathbb{E}_{W,\hat{W}}[(W - \hat{W})^T \Sigma_X (W - \hat{W})]$ should be minimized, given certain rate.
- Modified "golden rules" for practice
  1. Orthogonality: $\hat{w}^T I_w (w - \hat{w}) = 0$,
  2. Minimization: $(w - \hat{w})^T I_w (w - \hat{w})$ is minimized given certain constraints.

From theory to practice

- Two "golden rules" of the optimal compressor
  1. Orthogonality: $\mathbb{E}_{W,\hat{W}}[\hat{W}^T\Sigma_X(W-\hat{W})]=0$
  2. Minimization: $\mathbb{E}_{W,\hat{W}}[(W-\hat{W})^T\Sigma_X(W-\hat{W})]$ should be minimized, given certain rate.
- Modified "golden rules" for practice
  1. Orthogonality: $\hat{w}^T I_w(w-\hat{w})=0$,
  2. Minimization: $(w-\hat{w})^T I_w(w-\hat{w})$ is minimized given certain constraints.

here I_w is the **weight importance matrix**

- For regression, $I_w=\mathbb{E}_X\left[\nabla_w f_w(X)(\nabla_w f_w(X))^T\right]$
- For classification, $I_w=\mathbb{E}_X\left[(\nabla_w f_w(X))\mathrm{diag}[f_w^{-1}(X)](\nabla_w f_w(X))^T\right]$

Optimality of "golden rules"

- One-layer ReLU model $f_w(x) = ReLU(w^T x)$.
  - Data X has zero mean and $\mathbb{E}[X_i^2] = \lambda_{x,i}$, $\mathbb{E}[X_i X_j] = 0$

Optimality of "golden rules"

- One-layer ReLU model $f_w(x) = ReLU(w^T x)$.
  - Data X has zero mean and $\mathbb{E}[X_i^2] = \lambda_{x,i}$, $\mathbb{E}[X_i X_j] = 0$
- For **pruning** and **quantization** algorithm, if a compressor minimizes $(w - \hat{w})^T I_w (w - \hat{w})$, it *automatically* satisfies orthogonality: $\hat{w}^T I_w (\hat{w} - w) = 0$.

Optimality of "golden rules"

- One-layer ReLU model $f_w(x) = ReLU(w^T x)$.
  - Data X has zero mean and $\mathbb{E}[X_i^2] = \lambda_{x,i}$, $\mathbb{E}[X_i X_j] = 0$
- For **pruning** and **quantization** algorithm, if a compressor minimizes $(w - \hat{w})^T I_w (w - \hat{w})$, it *automatically* satisfies orthogonality: $\hat{w}^T I_w (\hat{w} - w) = 0$.
- Hence, for pruning and quantization, minimizing the objective $(w - \hat{w})^T I_w (w - \hat{w})$ is equivalent to minimizing MSE loss.

Optimality of "golden rules"

- One-layer ReLU model $f_w(x) = ReLU(w^T x)$.
  - Data X has zero mean and $\mathbb{E}[X_i^2] = \lambda_{x,i}$, $\mathbb{E}[X_i X_j] = 0$
- For **pruning** and **quantization** algorithm, if a compressor minimizes $(w - \hat{w})^T I_w (w - \hat{w})$, it *automatically* satisfies orthogonality: $\hat{w}^T I_w (\hat{w} - w) = 0$.
- Hence, for pruning and quantization, minimizing the objective $(w - \hat{w})^T I_w (w - \hat{w})$ is equivalent to minimizing MSE loss.
- For practical models, we test the objective on real data.

Real data experiment

- CIFAR10 with 5 conv layers + 3 fc layers (More experiments in full paper)

Real data experiment

- CIFAR10 with 5 conv layers + 3 fc layers (More experiments in full paper)
- Algorithms
  - Pruning: same prune ratio for all conv and fc layers
  - Quantization: same number of clusters for all conv and fc layers.

Real data experiment

- CIFAR10 with 5 conv layers + 3 fc layers (More experiments in full paper)
- Algorithms
  - Pruning: same prune ratio for all conv and fc layers
  - Quantization: same number of clusters for all conv and fc layers.
- Recall that for classification problem,

$$I_w = \mathbb{E}_X \left[(\nabla_w f_w(X)) \mathrm{diag}[f_w^{-1}(X)] (\nabla_w f_w(X))^T \right]$$

Real data experiment

- CIFAR10 with 5 conv layers + 3 fc layers (More experiments in full paper)
- Algorithms
  - Pruning: same prune ratio for all conv and fc layers
  - Quantization: same number of clusters for all conv and fc layers.
- Recall that for classification problem,

$$I_w = \mathbb{E}_X \left[(\nabla_w f_w(X)) \mathrm{diag}[f_w^{-1}(X)] (\nabla_w f_w(X))^T \right]$$

- We drop the off-diagonal terms of I_w

Real data experiment

- CIFAR10 with 5 conv layers + 3 fc layers (More experiments in full paper)
- Algorithms
  - Pruning: same prune ratio for all conv and fc layers
  - Quantization: same number of clusters for all conv and fc layers.
- Recall that for classification problem,

$$I_w = \mathbb{E}_X \left[(\nabla_w f_w(X)) \mathrm{diag}[f_w^{-1}(X)] (\nabla_w f_w(X))^T \right]$$

- We drop the off-diagonal terms of I_w
- Compare with baseline: $I_w =$ identity.

Name	Minimizing objective
Baseline	$\sum_{i=1}^m (w_i - \hat{w}_i)^2$
Proposed	$\sum_{i=1}^m \mathbb{E}_X[\frac{(\nabla_{w_i} f_w(X))^2}{f_w(X)}](w_i - \hat{w}_i)^2$

Table 1: Comparison of unsupervised compression objectives.

Real data experiment

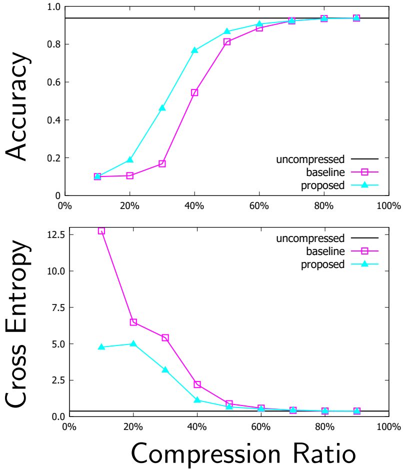

Figure 2: Result for unsupervised experiment. Left: pruning. Right: quantization.

Real data experiment

- In the previous experiments, we didn't use the **training labels**

Real data experiment

- In the previous experiments, we didn't use the **training labels**
- To use training label, treat the loss function $\mathcal{L}_w(x, y) = \mathcal{L}(f_w(x), y)$ as a function to be compressed and define

$$I_w \;=\; \mathbb{E}\left[\nabla_w \mathcal{L}_w(X, Y)(\nabla_w \mathcal{L}_w(X, Y))^T\right]$$

- In the previous experiments, we didn't use the **training labels**
- To use training label, treat the loss function $\mathcal{L}_w(x, y) = \mathcal{L}(f_w(x), y)$ as a function to be compressed and define

$$I_w = \mathbb{E}\left[\nabla_w \mathcal{L}_w(X, Y)(\nabla_w \mathcal{L}_w(X, Y))^T\right]$$

- By first and second order approximation of $\mathcal{L}$, we propose

Name	Minimizing objective
Baseline	$\sum_{i=1}^m (w_i - \hat{w}_i)^2$
Gradient (1st approx. of $\mathcal{L}$)	$\sum_{i=1}^m \mathbb{E}[(\nabla_{w_i} \mathcal{L}_w(X, Y))^2](w_i - \hat{w}_i)^2$
Hessian ([LeCun 90'])	$\sum_{i=1}^m \mathbb{E}[\nabla_{w_i}^2 \mathcal{L}_w(X, Y)](w_i - \hat{w}_i)^2$
Gradient+Hessian (2nd approx. of $\mathcal{L}$)	$\sum_{i=1}^m \mathbb{E}[(\nabla_{w_i} \mathcal{L}_w(X, Y))^2](w_i - \hat{w}_i)^2 + \frac{1}{4}\sum_{i=1}^m \mathbb{E}[(\nabla_{w_i}^2 \mathcal{L}_w(X, Y))^2](w_i - \hat{w}_i)^4$

Table 2: Comparison of supervised compression objectives.

Real data experiment

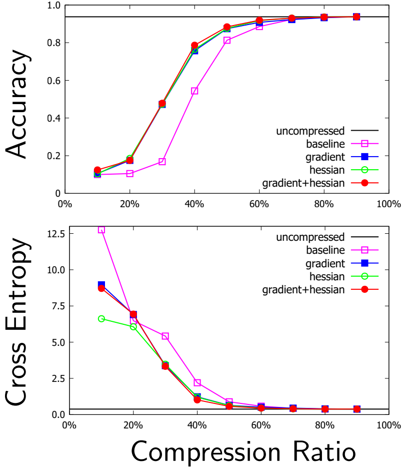

Figure 3: Result for supervised pruning experiment. Left: pruning. Right: quantization.

Thanks

Thank you for your attention!
Our poster **#169** tonight.