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
Nowadays, neural networks become more and more powerful.
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
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?
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?
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

Given a pretrained model $f_w(x)$, how well can we compress the model, given certain ratio?
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?
We bring the tool of *rate distortion theory* from information theory.
We bring the tool of *rate distortion theory* from information theory.

**Rate**: average number of bits to represent parameters.
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

\[ d(w, \hat{w}) = \mathbb{E}_{X} \left[ \| f_w(X) - f_{\hat{w}}(X) \|_2^2 \right] \]

For classification

\[ d(w, \hat{w}) = \mathbb{E}_{X} \left[ \text{KL}(f_{\hat{w}}(X) || f_w(X)) \right] \]
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

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.
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

Weihao Gao (UIUC)
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
Consider linear regression model $f_w(x) = w^T x$
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 \( \mathbb{E}[X_i^2] = \lambda_{x,i} \), \( \mathbb{E}[X_i X_j] = 0 \).

The lower bound is tight for linear regression.

\( \text{Theorem: the rate distortion function is lower bounded by:} \)

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

where \( D_i = \begin{cases} \frac{\mu}{\lambda_{x,i}} & \text{if } \mu < \lambda_{x,i} \\ \mathbb{E}[W_i^2] & \text{if } \mu \geq \lambda_{x,i} \end{cases} \).
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:
  \[
  R(D) \geq R(D) = \frac{1}{2} \log \det(\Sigma_W) - \sum_{i=1}^{m} \frac{1}{2} \log(D_i),
  \]
  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 $\mu$ is chosen that $\sum_{i=1}^{m} \lambda_{x,i} D_i = D$.
- The lower bound is **tight** for linear regression.
Two “golden rules” of the optimal compressor

1. Orthogonality: $E_{W,\hat{W}}[\hat{W}^T\Sigma_X(W - \hat{W})] = 0$

2. Minimization: $E_{W,\hat{W}}[(W - \hat{W})^T\Sigma_X(W - \hat{W})]$ should be minimized, given certain rate.

For regression, $I_w = E_X[\nabla_w f_w(X)(\nabla_w f_w(X))^T]$.

For classification, $I_w = E_X[\nabla_w f_w(X)\text{diag}[f - 1_w(X)]\nabla_w f_w(X)^T]$. 

Weihao Gao (UIUC)
From theory to practice

- **Two “golden rules” of the optimal compressor**
  1. Orthogonality: \( \mathbb{E}_{\hat{W}, \hat{W}} [\hat{W}^T \Sigma_X (W - \hat{W})] = 0 \)
  2. Minimization: \( \mathbb{E}_{\hat{W}, \hat{W}} [(W - \hat{W})^T \Sigma_X (W - \hat{W})] \) 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.

\( \Sigma_X \) is the empirical covariance, \( I_w \) is the weight importance matrix. For regression, \( I_w = \mathbb{E}(\nabla_{w} f_w(X) (\nabla_{w} f_w(X))^T) \). For classification, \( I_w = \mathbb{E}(\nabla_{w} f_w(X) \text{diag}(f - 1_w(X)) (\nabla_{w} f_w(X))^T) \).
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, \( I_w = \mathbb{E}_X \left[ (\nabla_w f_w(X)) \text{diag}[f_w^{-1}(X)] (\nabla_w f_w(X))^T \right] \)
Optimality of “golden rules”

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

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.

\[ I_w = E_X \left[ (\nabla w f(w(X))) \text{diag} \left( f - 1 w(X) \right) (\nabla w f(w(X)))^T \right] \]

We drop the off-diagonal terms of \( I_w \)

Compare with baseline: \( I_w = \text{identity} \).

<table>
<thead>
<tr>
<th>Name</th>
<th>Minimizing objective</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>( \sum_{m=1}^{m} (w_i - \hat{w}_i)^2 )</td>
</tr>
<tr>
<td>Proposed</td>
<td>( \sum_{m=1}^{m} E_X \left[ (\nabla w f(w(X)))^2 f(w(X)) (w_i - \hat{w}_i)^2 \right] )</td>
</tr>
</tbody>
</table>

Table 1: Comparison of unsupervised compression objectives.
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,

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

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

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,

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

- We drop the off-diagonal terms of \( I_w \)
- Compare with baseline: \( I_w = \text{identity.} \)

<table>
<thead>
<tr>
<th>Name</th>
<th>Minimizing objective</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>( \sum_{i=1}^{m} (w_i - \hat{w}_i)^2 )</td>
</tr>
<tr>
<td>Proposed</td>
<td>( \sum_{i=1}^{m} \mathbb{E}_X \left[ \frac{(\nabla_w f_w(X))^2}{f_w(X)} \right](w_i - \hat{w}_i)^2 )</td>
</tr>
</tbody>
</table>

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

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

<table>
<thead>
<tr>
<th>Name</th>
<th>Minimizing objective</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseline</td>
<td>$\sum_{i=1}^{m} (w_i - \hat{w}_i)^2$</td>
</tr>
<tr>
<td>Gradient (1st approx. of $\mathcal{L}$)</td>
<td>$\sum_{i=1}^{m} \mathbb{E}[(\nabla_{w_i} \mathcal{L}_w(X, Y))^2](w_i - \hat{w}_i)^2$</td>
</tr>
<tr>
<td>Hessian ([LeCun 90'])</td>
<td>$\sum_{i=1}^{m} \mathbb{E}[\nabla^2_{w_i} \mathcal{L}_w(X, Y)](w_i - \hat{w}_i)^2$</td>
</tr>
<tr>
<td>Gradient+Hessian (2nd approx. of $\mathcal{L}$)</td>
<td>$\sum_{i=1}^{m} \mathbb{E}[(\nabla_{w_i} \mathcal{L}<em>w(X, Y))^2](w_i - \hat{w}<em>i)^2$ $+ \frac{1}{4} \sum</em>{i=1}^{m} \mathbb{E}[(\nabla^2</em>{w_i} \mathcal{L}_w(X, Y))^2](w_i - \hat{w}_i)^4$</td>
</tr>
</tbody>
</table>

Table 2: Comparison of supervised compression objectives.
Real data experiment

Figure 3: Result for supervised pruning experiment. Left: pruning. Right: quantization.
Thank you for your attention!
Our poster #169 tonight.