AI Application Lecture 9

Neural Network Compression and Distance Measures between Probabilistic Language Models

SUZUKI, Atsushi
Jing WANG

Outline

Introduction

1.1 Review of the Previous Lecture

In the previous lecture, we rigorously defined and calculated frameworks for the **automatic and quantitative evaluation** of a **single probabilistic language model**.

1.1 Review of the Previous Lecture

In the previous lecture, we rigorously defined and calculated frameworks for the **automatic and quantitative evaluation** of a **single probabilistic language model**.

We covered:

- Perplexity
- Accuracy of the most likely option in multiple-choice questions
- Various metrics for string output (EM/F1, BLEU, ROUGE, chrF, BERTScore, numerical Accuracy)

1.2 Learning Outcomes for This Lecture

By the end of this lecture, you should be able to:

- Explain the motivation and methods to **reduce the scale (model compression)** of a **neural network** while maintaining its properties as a **function**.

1.2 Learning Outcomes for This Lecture

By the end of this lecture, you should be able to:

- Explain the motivation and methods to **reduce the scale (model compression)** of a **neural network** while maintaining its properties as a **function**.
- When a **probabilistic language model** is modified, evaluate the **amount of change from the original model** in a **mathematically rigorous** and **quantitative** manner.

Preliminaries: Mathematical Notations

2. Preliminaries: Mathematical Notations

Set:

- Sets: $\mathcal{A}$
- Membership: $x \in \mathcal{A}$
- Empty set: $\{\}$
- Roster notation: $\{a, b, c\}$
- Set-builder: $\{x \in \mathcal{A} | P(x)\}$
- Cardinality: $|\mathcal{A}|$
- Real numbers: $\mathbb{R}, \mathbb{R}_{>0}, \mathbb{R}_{\geq 0}$
- Integers: $\mathbb{Z}, \mathbb{Z}_{>0}, \mathbb{Z}_{\geq 0}$
- Integer range: $[1, k]_{\mathbb{Z}} := \{1, \ldots, k\}$

Function:

- $f : \mathcal{X} \to \mathcal{Y}$: f maps from $\mathcal{X}$ to $\mathcal{Y}$.
- $y = f(x)$: The output of f for input x.

Definition:

- $(\mathrm{LHS}) := (\mathrm{RHS})$: Left side is defined by the right side.

Sequence: Denoted by $\boldsymbol{a} = (a_1, a_2, \dots)$.

- A function $\boldsymbol{a} : [1, n]_{\mathbb{Z}} \to \mathcal{A}$.
- Length is denoted by $|\boldsymbol{a}|$.

Vector: Denoted by $\boldsymbol{v}$.

- A column of numbers, $\boldsymbol{v} \in \mathbb{R}^n$.
- i-th element is v_i.

Matrix: Denoted by $\boldsymbol{A}$.

- $m \times n$ matrix: $\boldsymbol{A} \in \mathbb{R}^{m,n}$.
- (i, j)-th element is $a_{i,j}$.
- Transpose: $\boldsymbol{A}^\top$.

Tensor: Denoted by $\underline{\boldsymbol{A}}$.

- Simply a multi-dimensional array.
- Vector $\to$ 1st-order, Matrix $\to$ 2nd-order.

Neural Network Compression (Model Compression)

3.1 Revisiting the Formulation of AI as Function Learning

The primary goal of AI was to **learn an appropriate relationship (function) between inputs and outputs**.

3.1 Revisiting the Formulation of AI as Function Learning

The primary goal of AI was to **learn an appropriate relationship (function) between inputs and outputs**.

However, even after a good function has been obtained, there is a strong practical motivation to modify it to reduce **implementation resources** (memory, computational complexity, latency) while **preserving the input-output relationship of that function as much as possible**.

3.1 Revisiting the Formulation of AI as Function Learning

The primary goal of AI was to **learn an appropriate relationship (function) between inputs and outputs**.

However, even after a good function has been obtained, there is a strong practical motivation to modify it to reduce **implementation resources** (memory, computational complexity, latency) while **preserving the input-output relationship of that function as much as possible**.

This process is what we call model **compression.**

3.2 Motivation and Overview of Compression

Even when a good function f has already been obtained, there is motivation to make the **model's representation (parameterization)** more lightweight while preserving the **function values (input-output relationship)**.

3.2 Motivation and Overview of Compression

Even when a good function f has already been obtained, there is motivation to make the **model's representation (parameterization)** more lightweight while preserving the **function values (input-output relationship)**.

Representative methods include:

- **Low-precision floating point**
- **Quantization**
- **Model distillation** [9, 6, 5, 4, 2]

3.3 Rigorous Definition of Low-precision floating point

Definition (Generalized Low-Precision Real Number System and Rounding Operator)

Let $d_{\mathrm{param}} \in \mathbb{Z}_{>0}$. For each index $i \in \{1, \ldots, d_{\mathrm{param}}\}$, we are given a **finite set** $\mathbb{F}_i \subset \mathbb{R}$ as a **floating-point format** and a **rounding operator** $R_i : \mathbb{R} \to \mathbb{F}_i$.

Definition (Generalized Low-Precision Real Number System and Rounding Operator)

Let $d_{\text{param}} \in \mathbb{Z}_{>0}$. For each index $i \in \{1, \ldots, d_{\text{param}}\}$, we are given a **finite set** $\mathbb{F}_i \subset \mathbb{R}$ as a **floating-point format** and a **rounding operator** $R_i : \mathbb{R} \to \mathbb{F}_i$.

The post-low-precision format for each component is $\mathbb{F}'_i \subset \mathbb{R}$ with rounding operator $R'_i : \mathbb{R} \to \mathbb{F}'_i$.

Definition (Generalized Low-Precision Real Number System and Rounding Operator)

Let $d_{\mathrm{param}} \in \mathbb{Z}_{>0}$. For each index $i \in \{1, \ldots, d_{\mathrm{param}}\}$, we are given a **finite set** $\mathbb{F}_i \subset \mathbb{R}$ as a **floating-point format** and a **rounding operator** $R_i : \mathbb{R} \to \mathbb{F}_i$.

The post-low-precision format for each component is $\mathbb{F}'_i \subset \mathbb{R}$ with rounding operator $R'_i : \mathbb{R} \to \mathbb{F}'_i$.

The **generalized low-precision mapping** $\Phi_{\mathrm{fp}} : \mathbb{R}^{d_{\mathrm{param}}} \to \mathcal{F}'$ is defined as

$$\Phi_{\mathrm{fp}}(\boldsymbol{\theta}) := \left(R'_1(\theta_1),\, R'_2(\theta_2),\, \ldots,\, R'_{d_{\mathrm{param}}}(\theta_{d_{\mathrm{param}}})\right) \tag{1}$$

The method of replacing $\boldsymbol{\theta}$ with $\Phi_{\mathrm{fp}}(\boldsymbol{\theta})$ is called **low-precision conversion**.

Remark

Equations allow for a mixed type where the format differs for each **element (per-parameter).**

3.3 Rigorous Definition of Low-precision floating point

Remark

Equations allow for a mixed type where the format differs for each **element (per-parameter)**.

In **mixed-precision training**, different formats $\mathbb{F}_i$ are assigned to gradients, gradient accumulations, weights, and activations to reduce computational resources [9].

3.4 Rigorous Definition of Quantization

Definition (General Form of Integer Quantization)

Given a parameter vector $\boldsymbol{\theta}$, we partition its indices. For each partition, we define a **quantization mapping** Q_j and a **dequantization mapping** $\widetilde{Q}_j$.

3.4 Rigorous Definition of Quantization

Definition (General Form of Integer Quantization)

Given a parameter vector $\boldsymbol{\theta}$, we partition its indices. For each partition, we define a **quantization mapping** Q_j and a **dequantization mapping** $\widetilde{Q}_j$.

$$Q_j(x) := \mathrm{clip}_{[-M_j,\, M_j]}\Big(\mathrm{round}\big(\tfrac{x}{s_j}\big) + b_j\Big), \tag{2}$$

$$\widetilde{Q}_j(n) := s_j\,\big(n - b_j\big) \qquad (n \in \mathbb{Z}) \tag{3}$$

where s_j is a scale, b_j is a zero-point, and M_j is a clipping value.

3.4 Rigorous Definition of Quantization

Definition (General Form of Integer Quantization)

Given a parameter vector $\boldsymbol{\theta}$, we partition its indices. For each partition, we define a **quantization mapping** Q_j and a **dequantization mapping** $\widetilde{Q}_j$.

$$Q_j(x) \coloneqq \mathrm{clip}_{[-M_j,\, M_j]}\Big(\mathrm{round}\big(\tfrac{x}{s_j}\big) + b_j\Big), \tag{2}$$

$$\widetilde{Q}_j(n) \coloneqq s_j\,(n - b_j) \qquad (n \in \mathbb{Z}) \tag{3}$$

where s_j is a scale, b_j is a zero-point, and M_j is a clipping value.

The goal is, for a given function $f_{\boldsymbol{\theta}}$, to choose appropriate quantization parameters to achieve

$$f_{\mathsf{Quantize}(\boldsymbol{\theta}')} \approx f_{\boldsymbol{\theta}} \tag{4}$$

Remark

Determining quantization parameters $(\boldsymbol{s}, \boldsymbol{b})$ based on data statistics after training is called **post-training quantization** (PTQ). Optimizing them during the training process is called **quantization-aware training** (QAT) [6, 5].

Remark

Determining quantization parameters $(\boldsymbol{s}, \boldsymbol{b})$ based on data statistics <u>after</u> training is called **post-training quantization** (PTQ). Optimizing them <u>during</u> the training process is called **quantization-aware training** (QAT) [6, 5].

Remark

In practice, not just parameters but also **intermediate outputs (activations)** are often quantized. This allows arithmetic to be performed using faster, more efficient integer operations [6].

Example (Manual Calculation Example of Quantize and Dequantize)

Given: $d_{\mathrm{param}} = 4$, $\boldsymbol{\theta}' = (1.20,\ -0.55,\ 0.07,\ 2.31)$, and a partition $(\mathcal{I}_1, \mathcal{I}_2) = (\{1,2\},\{3,4\})$.

Parameters for partition 1: $M_1 = 127,\ s_1 = 0.01,\ b_1 = 0$.

Parameters for partition 2: $M_2 = 7,\ s_2 = 0.1,\ b_2 = 1$.

Let's find the quantized vector $\boldsymbol{q}$ and the dequantized vector $\widetilde{\boldsymbol{\theta}}$.

3.4 Rigorous Definition of Quantization

Component 1 ($i = 1 \in \mathcal{I}_1$):

$$\begin{aligned} q_1 &= \mathrm{clip}_{[-127,127]}\big(\mathrm{round}(1.20/0.01) + 0\big) = \mathrm{clip}(120) = 120 \\ \widetilde{\theta}_1 &= 0.01 \cdot (120 - 0) = 1.20 \end{aligned}$$

Component 1 ($i = 1 \in \mathcal{I}_1$):

$$\begin{aligned} q_1 &= \mathrm{clip}_{[-127,127]}\big(\mathrm{round}(1.20/0.01) + 0\big) = \mathrm{clip}(120) = 120 \\ \widetilde{\theta}_1 &= 0.01 \cdot (120 - 0) = 1.20 \end{aligned}$$

Component 2 ($i = 2 \in \mathcal{I}_1$):

$$\begin{aligned} q_2 &= \mathrm{clip}_{[-127,127]}\big(\mathrm{round}(-0.55/0.01)\big) = \mathrm{clip}(-55) = -55 \\ \widetilde{\theta}_2 &= 0.01 \cdot (-55 - 0) = -0.55 \end{aligned}$$

3.4 Rigorous Definition of Quantization

Component 3 ($i = 3 \in \mathcal{I}_2$):

$$
\begin{aligned}
q_3 &= \mathrm{clip}_{[-7,7]}\big(\mathrm{round}(0.07/0.1) + 1\big) = \mathrm{clip}(1 + 1) = 2 \\
\widetilde{\theta}_3 &= 0.1 \cdot (2 - 1) = 0.1
\end{aligned}
$$

3.4 Rigorous Definition of Quantization

Component 3 ($i = 3 \in \mathcal{I}_2$):

$$q_3 = \text{clip}_{[-7,7]}\big(\text{round}(0.07/0.1) + 1\big) = \text{clip}(1+1) = 2$$
$$\widetilde{\theta}_3 = 0.1 \cdot (2-1) = 0.1$$

Component 4 ($i = 4 \in \mathcal{I}_2$):

$$q_4 = \text{clip}_{[-7,7]}\big(\text{round}(2.31/0.1) + 1\big) = \text{clip}(23+1) = \text{clip}(24) = 7$$
$$\widetilde{\theta}_4 = 0.1 \cdot (7-1) = 0.6$$

3.4 Rigorous Definition of Quantization

From the calculations, the final vectors are:

Quantized vector:

$$\boldsymbol{q} = (120,\ -55,\ 2,\ 7)$$

Dequantized vector:

$$\widetilde{\boldsymbol{\theta}} = (1.20,\ -0.55,\ 0.1,\ 0.6)$$

It can be seen that the large value 2.31 was saturated by the clipping value M_2, increasing the error.

Exercise (Quantize/Dequantize Exercise)

Let $d_{\text{param}} = 3$, $\boldsymbol{\theta}' = (0.34,\ -1.26,\ 0.51)$. Partition: $(\mathcal{I}_1, \mathcal{I}_2) = (\{1,3\}, \{2\})$.

- For $\mathcal{I}_1$: $M_1 = 15,\ s_1 = 0.02,\ b_1 = 2$.
- For $\mathcal{I}_2$: $M_2 = 127,\ s_2 = 0.01,\ b_2 = 0$.

Find $\boldsymbol{q}$ and $\widetilde{\boldsymbol{\theta}}$.

3.4 Rigorous Definition of Quantization

Answer

Component 1 ($i = 1 \in \mathcal{I}_1$): $q_1 = \text{clip}_{[-15,15]}(\text{round}(0.34/0.02) + 2) = 15$. $\widetilde{\theta}_1 = 0.02 \cdot (15 - 2) = 0.26$.

Answer

Component 1 ($i = 1 \in \mathcal{I}_1$): $q_1 = \mathrm{clip}_{[-15,15]}(\mathrm{round}(0.34/0.02) + 2) = 15$.
$\widetilde{\theta}_1 = 0.02 \cdot (15 - 2) = 0.26$.

Component 2 ($i = 2 \in \mathcal{I}_2$): $q_2 = \mathrm{clip}_{[-127,127]}(\mathrm{round}(-1.26/0.01) + 0) = -126$.
$\widetilde{\theta}_2 = 0.01 \cdot (-126 - 0) = -1.26$.

Answer

Component 1 ($i = 1 \in \mathcal{I}_1$): $q_1 = \mathrm{clip}_{[-15,15]}(\mathrm{round}(0.34/0.02) + 2) = 15$.
$\widetilde{\theta}_1 = 0.02 \cdot (15 - 2) = 0.26$.

Component 2 ($i = 2 \in \mathcal{I}_2$): $q_2 = \mathrm{clip}_{[-127,127]}(\mathrm{round}(-1.26/0.01) + 0) = -126$.
$\widetilde{\theta}_2 = 0.01 \cdot (-126 - 0) = -1.26$.

Component 3 ($i = 3 \in \mathcal{I}_1$): $q_3 = \mathrm{clip}_{[-15,15]}(\mathrm{round}(0.51/0.02) + 2) = 15$.
$\widetilde{\theta}_3 = 0.02 \cdot (15 - 2) = 0.26$.

Answer

Component 1 ($i = 1 \in \mathcal{I}_1$): $q_1 = \mathrm{clip}_{[-15,15]}(\mathrm{round}(0.34/0.02) + 2) = 15$.
$\widetilde{\theta}_1 = 0.02 \cdot (15 - 2) = 0.26$.

Component 2 ($i = 2 \in \mathcal{I}_2$): $q_2 = \mathrm{clip}_{[-127,127]}(\mathrm{round}(-1.26/0.01) + 0) = -126$.
$\widetilde{\theta}_2 = 0.01 \cdot (-126 - 0) = -1.26$.

Component 3 ($i = 3 \in \mathcal{I}_1$): $q_3 = \mathrm{clip}_{[-15,15]}(\mathrm{round}(0.51/0.02) + 2) = 15$.
$\widetilde{\theta}_3 = 0.02 \cdot (15 - 2) = 0.26$.

Final vectors:

$$\boldsymbol{q} = (15,\ -126,\ 15) \quad \text{and} \quad \widetilde{\boldsymbol{\theta}} = (0.26,\ -1.26,\ 0.26)$$

The error is large because the 1st and 3rd components were saturated by $M_1 = 15$.

3.5 Rigorous Definition of Knowledge Distillation

Definition (Model Distillation)

Suppose we are given a function $f_{\boldsymbol{\theta}}$ (the "teacher" model).

3.5 Rigorous Definition of Knowledge Distillation

Definition (Model Distillation)

Suppose we are given a function $f_{\boldsymbol{\theta}}$ (the "teacher" model).

For the purpose of approximating $f_{\boldsymbol{\theta}}$ at a lower cost, we use another parametric function $g_{(\cdot)}$ with a lower-dimensional parameter space (the "student" model).

3.5 Rigorous Definition of Knowledge Distillation

Definition (Model Distillation)

Suppose we are given a function $f_{\boldsymbol{\theta}}$ (the "teacher" model).

For the purpose of approximating $f_{\boldsymbol{\theta}}$ at a lower cost, we use another parametric function $g_{(\cdot)}$ with a lower-dimensional parameter space (the "student" model).

We then find a suitable parameter vector γ such that g_{γ} becomes a good approximation of $f_{\boldsymbol{\theta}}$. This process is called **model distillation**.

Definition (Model Distillation)

Suppose we are given a function $f_{\boldsymbol{\theta}}$ (the "teacher" model).

For the purpose of approximating $f_{\boldsymbol{\theta}}$ at a lower cost, we use another parametric function $g_{(\cdot)}$ with a lower-dimensional parameter space (the "student" model).

We then find a suitable parameter vector γ such that g_{γ} becomes a good approximation of $f_{\boldsymbol{\theta}}$. This process is called **model distillation**.

Remark

This is a form of **model distillation**. Another form is **dataset distillation**, where the goal is to synthesize a small dataset that captures the learning effect of a large original dataset.

Divergence between Probabilistic Language Models

4. Divergence between Probabilistic Language Models

The core of large language models is a probabilistic language model.

4. Divergence between Probabilistic Language Models

The core of large language models is a probabilistic language model.

When this is compressed, one becomes concerned about how much it differs from the original.

4. Divergence between Probabilistic Language Models

The core of large language models is a probabilistic language model.

When this is compressed, one becomes concerned about how much it differs from the original.

This chapter deals with metrics that express how much another probabilistic language model has diverged when there is a reference probabilistic language model.

4.1 A Simple Method: Differences in Individual Evaluations

A simple method is to measure the performance of two models on the same task (e.g., multiple-choice questions) and compare the **difference.**

4.1 A Simple Method: Differences in Individual Evaluations

A simple method is to measure the performance of two models on the same task (e.g., multiple-choice questions) and compare the **difference.**

However, if the evaluation is based solely on the answers, it is possible to overlook cases where the inference processes are significantly different even if the final answers are the same [1].

4.1 A Simple Method: Differences in Individual Evaluations

A simple method is to measure the performance of two models on the same task (e.g., multiple-choice questions) and compare the **difference**.

However, if the evaluation is based solely on the answers, it is possible to overlook cases where the inference processes are significantly different even if the final answers are the same [1].

Therefore, one might consider directly comparing the probability distributions constituted by the probabilistic language models.

Definition (Vocabulary and Token Sequence)

The set of possible values a token can take is called the **vocabulary**, denoted by $\mathcal{V}$[1]. We identify $\mathcal{V}$ with $\{1, 2, \ldots, D\}$.

[1] In previous lectures, the set of nodes in a neural network was also denoted by $\mathcal{V}$, but since this lecture does not explicitly describe the graph structure of neural networks, $\mathcal{V}$ should always be taken to refer to the vocabulary.

Definition (Vocabulary and Token Sequence)

The set of possible values a token can take is called the **vocabulary**, denoted by $\mathcal{V}$[1]. We identify $\mathcal{V}$ with $\{1, 2, \ldots, D\}$.

- The set of token sequences of length n is $\mathcal{V}^n$.
- The set of all token sequences of finite length is $\mathcal{V}^* = \mathcal{V}^0 \cup \mathcal{V}^1 \cup \mathcal{V}^2 \cup \cdots$.
- For a sequence $\boldsymbol{t} = (t_1, \ldots, t_n)$, we use notations like $\boldsymbol{t}_{<i}$ for the prefix $(t_1, \ldots, t_{i-1})$.

[1] In previous lectures, the set of nodes in a neural network was also denoted by $\mathcal{V}$, but since this lecture does not explicitly describe the graph structure of neural networks, $\mathcal{V}$ should always be taken to refer to the vocabulary.

Definition (Probabilistic Language Model (most general form))

A **probabilistic language model** is a function $P(\cdot|\cdot)$ that, given any finite-length token sequence $\boldsymbol{t} \in \mathcal{V}^*$, returns the probability mass function of the next token, conditioned on it.

Definition (Probabilistic Language Model (most general form))

A **probabilistic language model** is a function $P(\cdot|\cdot)$ that, given any finite-length token sequence $\boldsymbol{t} \in \mathcal{V}^*$, returns the probability mass function of the next token, conditioned on it.

More formally, for any $\boldsymbol{t} \in \mathcal{V}^*$, the following must hold:

$$\sum_{v \in \mathcal{V}} P(v \mid \boldsymbol{t}) = 1 \tag{5}$$

Definition (Probabilistic Language Model (most general form))

A **probabilistic language model** is a function $P(\cdot|\cdot)$ that, given any finite-length token sequence $\boldsymbol{t} \in \mathcal{V}^*$, returns the probability mass function of the next token, conditioned on it.

More formally, for any $\boldsymbol{t} \in \mathcal{V}^*$, the following must hold:

$$\sum_{v \in \mathcal{V}} P(v \mid \boldsymbol{t}) = 1 \tag{5}$$

Since a probabilistic language model is a conditional probability mass function, the problem reduces to quantifying the divergence between general probability mass functions.

4.3 How to Quantify the Divergence of Probability Mass Functions

Given a reference probability mass function P and another one Q, we want to measure how much Q diverges from P.

4.3 How to Quantify the Divergence of Probability Mass Functions

Given a reference probability mass function P and another one Q, we want to measure how much Q diverges from P.

If Q is close to P, then for an outcome z where $P(z)$ is large, $Q(z)$ should also be large.

4.3 How to Quantify the Divergence of Probability Mass Functions

Given a reference probability mass function P and another one Q, we want to measure how much Q diverges from P.

If Q is close to P, then for an outcome z where $P(z)$ is large, $Q(z)$ should also be large.

This suggests a divergence criterion of the form:

$$\mathbb{E}_{Z\sim P}\big[\phi\big(Q(Z)\big)\big] - C \tag{6}$$

using a monotonically decreasing function ϕ to penalize small probability masses.

Definition (Axiomatization of Divergence Functional)

For $\Delta_\phi(P \parallel Q) \coloneqq \mathbb{E}_{Z\sim P}\big[\phi\big(Q(Z)\big)\big] - C$, we impose:

(A1) **Reflexivity**: $\Delta_\phi(P \parallel P) = 0$.

(A2) **Non-negativity**: $\Delta_\phi(P \parallel Q) \geq 0$, with equality iff $P = Q$.

(A3) **Continuity**: $\Delta_\phi(P \parallel Q)$ is continuous in Q.

(A4) **Additivity over independent products**: For product distributions $P_1 \otimes P_2$ and $Q_1 \otimes Q_2$,

$$\Delta_\phi(P_1 \otimes P_2 \parallel Q_1 \otimes Q_2) = \Delta_\phi(P_1 \parallel Q_1) + \Delta_\phi(P_2 \parallel Q_2)$$

Theorem (Uniqueness of KL Divergence)

For a monotonically decreasing continuous function ϕ satisfying axioms (A1) – (A4), there exist constants $c > 0$ and B such that

$$\phi(u) = -c \log u + B \qquad (u \in (0, 1]) \tag{7}$$

4.3 How to Quantify the Divergence of Probability Mass Functions

Theorem (Uniqueness of KL Divergence)

For a monotonically decreasing continuous function ϕ satisfying axioms (A1) – (A4), there exist constants $c > 0$ and B such that

$$\phi(u) = -c \log u + B \qquad (u \in (0,1]) \tag{7}$$

This implies that the divergence must be proportional to the KL divergence:

$$\Delta_\phi(P \parallel Q) = c\, \mathbb{E}_P\left[\log \frac{P(Z)}{Q(Z)}\right] = c\, D_{\mathrm{KL}}(P \parallel Q) \tag{8}$$

The proof is insightful:

- **Step 1:** The **additivity** axiom (A4) forces ϕ to satisfy a functional equation: $\phi(uv) - \phi(u) - \phi(v) =$ constant.

The proof is insightful:

- **Step 1:** The **additivity** axiom (A4) forces ϕ to satisfy a functional equation: $\phi(uv) - \phi(u) - \phi(v) =$ constant.
- **Step 2:** This functional equation, combined with the **continuity** axiom (A3), implies that ϕ must be a logarithmic function. This is a classic result related to Cauchy's functional equation.

The proof is insightful:

- **Step 1:** The **additivity** axiom (A4) forces ϕ to satisfy a functional equation: $\phi(uv) - \phi(u) - \phi(v) =$ constant.
- **Step 2:** This functional equation, combined with the **continuity** axiom (A3), implies that ϕ must be a logarithmic function. This is a classic result related to Cauchy's functional equation.
- **Step 3:** The **reflexivity** (A1) and **non-negativity** (A2) axioms fix the constants and ensure the result is proportional to the standard KL divergence, with a positive coefficient.

Definition (KL Divergence (for pmfs))

Let P, Q be two probability mass functions on a finite set $\mathcal{S}$. The **KL divergence (relative entropy)** is defined as

$$D_{\mathrm{KL}}(P \parallel Q) := \sum_{z \in \mathcal{S}} P(z) \, \log\left(\frac{P(z)}{Q(z)}\right) \in [0, \infty] \tag{9}$$

It is always non-negative, and $D_{\mathrm{KL}}(P \parallel Q) = 0$ if and only if $P = Q$ [7, 3].

Example (Complete Numerical Example of KL)

Consider the probability distributions on $\{a, b, c\}$:

$$P = (0.5,\ 0.3,\ 0.2)$$
$$Q = (0.4,\ 0.4,\ 0.2)$$

Example (Complete Numerical Example of KL)

Consider the probability distributions on $\{a, b, c\}$:

$$P = (0.5,\ 0.3,\ 0.2)$$
$$Q = (0.4,\ 0.4,\ 0.2)$$

From the definition:

$$\begin{aligned} D_{\mathrm{KL}}(P \parallel Q) &= \sum_{x \in \{a,b,c\}} P(x)\ \log \frac{P(x)}{Q(x)} \\ &= 0.5 \log \frac{0.5}{0.4} + 0.3 \log \frac{0.3}{0.4} + 0.2 \log \frac{0.2}{0.2} \end{aligned} \tag{10}$$

The last term is $0.2 \log(1) = 0$.

The last term is $0.2\log(1) = 0$.

Using the natural logarithm:

$$0.5\log(1.25) \approx 0.5 \times 0.22314 \approx 0.11157$$
$$0.3\log(0.75) \approx 0.3 \times (-0.28768) \approx -0.08630$$

The last term is $0.2\log(1) = 0$.

Using the natural logarithm:

$$0.5\log(1.25) \approx 0.5 \times 0.22314 \approx 0.11157$$
$$0.3\log(0.75) \approx 0.3 \times (-0.28768) \approx -0.08630$$

Summing them up:

$$D_{\mathrm{KL}}(P \parallel Q) \approx 0.11157 - 0.08630 = 0.02527 \text{ [nats]} \tag{11}$$

Exercise (Numerical Calculation of KL)

On the vocabulary $\{x_1, x_2, x_3\}$, let

$$P = (0.2,\ 0.5,\ 0.3), \qquad Q = (0.1,\ 0.7,\ 0.2)$$

Calculate $D_{\mathrm{KL}}(P \parallel Q)$ using the natural logarithm.

Answer

$$
\begin{aligned}
D_{\mathrm{KL}}(P \parallel Q) &= 0.2 \log \frac{0.2}{0.1} + 0.5 \log \frac{0.5}{0.7} + 0.3 \log \frac{0.3}{0.2} \\
&= 0.2 \log 2 + 0.5 \log(5/7) + 0.3 \log(3/2)
\end{aligned}
$$

Numerically, this is:

$$
\begin{aligned}
&\approx 0.2 \times (0.69315) + 0.5 \times (-0.33647) + 0.3 \times (0.40547) \\
&\approx 0.13863 - 0.16824 + 0.12164 \approx 0.09203 \text{ [nats]}
\end{aligned}
$$

4.5 Extension of KL to Language Models (Conditional Distributions)

How do we apply KL divergence to probabilistic language models, which are conditional distributions?

4.5 Extension of KL to Language Models (Conditional Distributions)

How do we apply KL divergence to probabilistic language models, which are conditional distributions? There are two main, virtually equivalent ways.

4.5 Extension of KL to Language Models (Conditional Distributions)

Definition (A. KL based on Joint Distribution)

For a fixed length n, we can define the **joint distributions** over sequences of length n induced by the language models P and Q.

$$P^{(n)}(\boldsymbol{t}_{1:n}) := \prod_{i=1}^{n} P(t_i \mid \boldsymbol{t}_{<i}) \tag{12}$$

(and similarly for $Q^{(n)}$).

Definition (A. KL based on Joint Distribution)

For a fixed length n, we can define the **joint distributions** over sequences of length n induced by the language models P and Q.

$$P^{(n)}(\boldsymbol{t}_{1:n}) := \prod_{i=1}^{n} P(t_i \mid \boldsymbol{t}_{<i}) \tag{12}$$

(and similarly for $Q^{(n)}$).

Then we define the KL divergence between these two joint distributions:

$$D_{\mathrm{KL}}\big(P^{(n)} \parallel Q^{(n)}\big) = \sum_{\boldsymbol{t}_{1:n}} P^{(n)}(\boldsymbol{t}_{1:n}) \log \frac{P^{(n)}(\boldsymbol{t}_{1:n})}{Q^{(n)}(\boldsymbol{t}_{1:n})} \tag{13}$$

4.5 Extension of KL to Language Models (Conditional Distributions)

This joint KL can be decomposed using a chain rule.

This joint KL can be decomposed using a chain rule.

Proposition (Chain Rule for KL Divergence)

For any $n \in \mathbb{Z}_{>0}$,

$$D_{\mathrm{KL}}\big(P^{(n)} \parallel Q^{(n)}\big) = \sum_{i=1}^{n} \mathbb{E}_{\boldsymbol{t}_{<i} \sim P^{(i-1)}} \big[D_{\mathrm{KL}}\big(P(\cdot \mid \boldsymbol{t}_{<i}) \parallel Q(\cdot \mid \boldsymbol{t}_{<i})\big) \big] \tag{14}$$

This is the sum of expected conditional KL divergences at each step.

Definition (B. KL based on a Dataset)

A more practical approach is to compute the average KL divergence over a validation dataset $\mathcal{D} = \{\boldsymbol{t}^{(j)}\}_{j=1}^{N}$.

Definition (B. KL based on a Dataset)

A more practical approach is to compute the average KL divergence over a validation dataset $\mathcal{D} = \{\boldsymbol{t}^{(j)}\}_{j=1}^{N}$.

We average the KL divergence of the next-token predictions over all positions in all sequences in the dataset:

$$\widehat{D}_{\mathrm{KL}}^{\mathcal{D}}(P \parallel Q) := \frac{1}{|\mathcal{D}|_{\text{tokens}}} \sum_{j=1}^{N} \sum_{i=1}^{|\boldsymbol{y}^{(j)}|} D_{\mathrm{KL}}\Big(P(\cdot \mid \text{context}) \parallel Q(\cdot \mid \text{context})\Big). \tag{15}$$

4.5 Extension of KL to Language Models (Conditional Distributions)

These two definitions are closely related.

These two definitions are closely related.

Proposition (Virtual Equivalence of A and B)

If the dataset $\mathcal{D}$ is generated i.i.d. according to the model P, then the expected value of the dataset-based KL divergence (B) is equal to the per-token joint KL divergence (A).

$$\mathbb{E}_{\mathcal{D}\sim(P^{(n)})^{\otimes N}}\left[\widehat{D}_{\mathrm{KL}}^{\mathcal{D}}(P \parallel Q)\right] = \frac{1}{n}\, D_{\mathrm{KL}}\big(P^{(n)} \parallel Q^{(n)}\big). \tag{16}$$

4.5 Extension of KL to Language Models (Conditional Distributions)

Example (Numerical Example of Sequential KL (length 2))

Let $\mathcal{V} = \{A, B\}$, $n = 2$. The conditional distributions are:

$$P(A \mid ()) = 0.6, P(A \mid A) = 0.7, P(A \mid B) = 0.2$$
$$Q(A \mid ()) = 0.5, Q(A \mid A) = 0.6, Q(A \mid B) = 0.3$$

(The probabilities for B are just 1 minus these values). Let's calculate $D_{\mathrm{KL}}(P^{(2)} \parallel Q^{(2)})$.

Step 1: KL for the first token ($i = 1$)

The context is the empty sequence $()$.

$$
\begin{aligned}
D_{\mathrm{KL}}\big(P(\cdot \mid ()) \parallel Q(\cdot \mid ())\big) &= 0.6 \log \frac{0.6}{0.5} + 0.4 \log \frac{0.4}{0.5} \\
&\approx 0.0204
\end{aligned}
$$

Step 2: Expected KL for the second token ($i = 2$)

We need to average the conditional KL over the first token, drawn from $P(\cdot \mid ())$.

$$\begin{aligned}
&\mathbb{E}_{t_1 \sim P}\big[D_{\mathrm{KL}}\big(P(\cdot \mid t_1) \parallel Q(\cdot \mid t_1)\big)\big] \\
&\quad = P(A \mid ()) \cdot D_{\mathrm{KL}}\big(P(\cdot \mid A) \parallel Q(\cdot \mid A)\big) \\
&\quad\ + P(B \mid ()) \cdot D_{\mathrm{KL}}\big(P(\cdot \mid B) \parallel Q(\cdot \mid B)\big)
\end{aligned}$$

4.5 Extension of KL to Language Models (Conditional Distributions)

Step 2: Expected KL for the second token ($i = 2$)

We need to average the conditional KL over the first token, drawn from $P(\cdot \mid ())$.

$$\begin{aligned}
&\mathbb{E}_{t_1 \sim P}\big[D_{\mathrm{KL}}\big(P(\cdot \mid t_1) \parallel Q(\cdot \mid t_1)\big)\big] \\
&\quad = P(A \mid ()) \cdot D_{\mathrm{KL}}\big(P(\cdot \mid A) \parallel Q(\cdot \mid A)\big) \\
&\quad\ + P(B \mid ()) \cdot D_{\mathrm{KL}}\big(P(\cdot \mid B) \parallel Q(\cdot \mid B)\big)
\end{aligned}$$

$$\begin{aligned}
&= 0.6 \cdot \left(0.7 \log \frac{0.7}{0.6} + 0.3 \log \frac{0.3}{0.4}\right) \\
&+ 0.4 \cdot \left(0.2 \log \frac{0.2}{0.3} + 0.8 \log \frac{0.8}{0.7}\right) \approx 0.0125
\end{aligned}$$

Step 3: Total KL Divergence

Summing the results from both steps:

$$
\begin{aligned}
D_{\mathrm{KL}}(P^{(2)} \parallel Q^{(2)}) &= (\text{KL at } i = 1) + (\text{Expected KL at } i = 2) \\
&\approx 0.0204 + 0.0125 \\
&= 0.0329
\end{aligned}
$$

4.5 Extension of KL to Language Models (Conditional Distributions)

Exercise (Sequential KL Calculation Practice)

Let $\mathcal{V} = \{0, 1\}$, $n = 2$.

$$P(1 \mid ()) = 0.3,\ P(1 \mid 1) = 0.6,\ P(1 \mid 0) = 0.2$$
$$Q(1 \mid ()) = 0.4,\ Q(1 \mid 1) = 0.5,\ Q(1 \mid 0) = 0.3$$

Find $D_{\mathrm{KL}}(P^{(2)} \parallel Q^{(2)})$.

Answer

First token ($i = 1$):

$$D_{\mathrm{KL}}(P(\cdot|()) \parallel Q(\cdot|())) = 0.3 \log \frac{0.3}{0.4} + 0.7 \log \frac{0.7}{0.6} \approx 0.0224$$

4.5 Extension of KL to Language Models (Conditional Distributions)

Answer

First token ($i = 1$):

$$D_{\mathrm{KL}}(P(\cdot|()) \parallel Q(\cdot|())) = 0.3 \log \frac{0.3}{0.4} + 0.7 \log \frac{0.7}{0.6} \approx 0.0224$$

Expected KL for second token ($i = 2$):

$$\begin{aligned}\mathbb{E}_{t_1 \sim P} &= 0.3 \cdot \left[0.6 \log \frac{0.6}{0.5} + 0.4 \log \frac{0.4}{0.5}\right] \\ &\quad + 0.7 \cdot \left[0.2 \log \frac{0.2}{0.3} + 0.8 \log \frac{0.8}{0.7}\right] \\ &\approx 0.3 \cdot (0.0201) + 0.7 \cdot (-0.0033) \approx 0.0037\end{aligned}$$

4.5 Extension of KL to Language Models (Conditional Distributions)

Answer

First token ($i = 1$):

$$D_{\mathrm{KL}}(P(\cdot|()) \parallel Q(\cdot|())) = 0.3 \log \frac{0.3}{0.4} + 0.7 \log \frac{0.7}{0.6} \approx 0.0224$$

Expected KL for second token ($i = 2$):

$$\begin{aligned}\mathbb{E}_{t_1 \sim P} &= 0.3 \cdot \left[0.6 \log \frac{0.6}{0.5} + 0.4 \log \frac{0.4}{0.5}\right] \\ &+ 0.7 \cdot \left[0.2 \log \frac{0.2}{0.3} + 0.8 \log \frac{0.8}{0.7}\right] \\ &\approx 0.3 \cdot (0.0201) + 0.7 \cdot (-0.0033) \approx 0.0037\end{aligned}$$

Total KL:

$$D_{\mathrm{KL}}(P^{(2)} \parallel Q^{(2)}) \approx 0.0224 + 0.0037 = 0.0261 \text{ [nats]}$$

(Note: previous lecture note answer 0.019 was a calculation error).

4.6 Definition of Jensen – Shannon (JS) divergence

KL divergence is not symmetric. A symmetric version is the JS divergence.

4.6 Definition of Jensen – Shannon (JS) divergence

KL divergence is not symmetric. A symmetric version is the JS divergence.

Definition (JS Divergence)

For the **mixture distribution** $M := \frac{1}{2}(P + Q)$ of P, Q,

$$D_{\mathrm{JS}}(P \parallel Q) := \frac{1}{2} D_{\mathrm{KL}}(P \parallel M) + \frac{1}{2} D_{\mathrm{KL}}(Q \parallel M) \tag{17}$$

is called the **Jensen – Shannon divergence.** D_{JS} is symmetric and bounded $[0, \log 2]$ [8].

4.6 Definition of Jensen – Shannon (JS) divergence

KL divergence is not symmetric. A symmetric version is the JS divergence.

Definition (JS Divergence)

For the **mixture distribution** $M := \frac{1}{2}(P + Q)$ of P, Q,

$$D_{\mathrm{JS}}(P \parallel Q) := \frac{1}{2} D_{\mathrm{KL}}(P \parallel M) + \frac{1}{2} D_{\mathrm{KL}}(Q \parallel M) \tag{17}$$

is called the **Jensen – Shannon divergence.** D_{JS} is symmetric and bounded $[0, \log 2]$ [8].

Remark

The square root of the JS divergence, $\sqrt{D_{\mathrm{JS}}}$, satisfies the axioms of a **metric**, including the triangle inequality [8].

Example (Complete Numerical Example of JS)

Using P, Q from the KL example:

$$P = (0.5,\ 0.3,\ 0.2)$$
$$Q = (0.4,\ 0.4,\ 0.2)$$

First, find the mixture distribution $M = \frac{1}{2}(P + Q)$.

Example (Complete Numerical Example of JS)

Using P, Q from the KL example:

$$P = (0.5,\ 0.3,\ 0.2)$$

$$Q = (0.4,\ 0.4,\ 0.2)$$

First, find the mixture distribution $M = \frac{1}{2}(P + Q)$.

$$M = (0.45,\ 0.35,\ 0.2)$$

4.6 Definition of Jensen – Shannon (JS) divergence

Example (Complete Numerical Example of JS)

Using P, Q from the KL example:

$$P = (0.5,\ 0.3,\ 0.2)$$
$$Q = (0.4,\ 0.4,\ 0.2)$$

First, find the mixture distribution $M = \frac{1}{2}(P + Q)$.

$$M = (0.45,\ 0.35,\ 0.2)$$

Now, calculate $D_{\mathrm{JS}}(P \parallel Q) = \frac{1}{2}D_{\mathrm{KL}}(P \parallel M) + \frac{1}{2}D_{\mathrm{KL}}(Q \parallel M)$.

4.6 Definition of Jensen – Shannon (JS) divergence

Step 1: Calculate $D_{\mathrm{KL}}(P \parallel M)$:

$$D_{\mathrm{KL}}(P \parallel M) = 0.5 \log \frac{0.5}{0.45} + 0.3 \log \frac{0.3}{0.35} + 0.2 \log \frac{0.2}{0.2}$$
$$\approx 0.0527 - 0.0463 + 0 = 0.0064$$

4.6 Definition of Jensen – Shannon (JS) divergence

Step 1: Calculate $D_{\mathrm{KL}}(P \parallel M)$:

$$\begin{aligned} D_{\mathrm{KL}}(P \parallel M) &= 0.5 \log \frac{0.5}{0.45} + 0.3 \log \frac{0.3}{0.35} + 0.2 \log \frac{0.2}{0.2} \\ &\approx 0.0527 - 0.0463 + 0 = 0.0064 \end{aligned}$$

Step 2: Calculate $D_{\mathrm{KL}}(Q \parallel M)$:

$$\begin{aligned} D_{\mathrm{KL}}(Q \parallel M) &= 0.4 \log \frac{0.4}{0.45} + 0.4 \log \frac{0.4}{0.35} + 0.2 \log \frac{0.2}{0.2} \\ &\approx -0.0472 + 0.0536 + 0 = 0.0064 \end{aligned}$$

4.6 Definition of Jensen – Shannon (JS) divergence

Step 1: Calculate $D_{\mathrm{KL}}(P \parallel M)$:

$$D_{\mathrm{KL}}(P \parallel M) = 0.5 \log \frac{0.5}{0.45} + 0.3 \log \frac{0.3}{0.35} + 0.2 \log \frac{0.2}{0.2}$$
$$\approx 0.0527 - 0.0463 + 0 = 0.0064$$

Step 2: Calculate $D_{\mathrm{KL}}(Q \parallel M)$:

$$D_{\mathrm{KL}}(Q \parallel M) = 0.4 \log \frac{0.4}{0.45} + 0.4 \log \frac{0.4}{0.35} + 0.2 \log \frac{0.2}{0.2}$$
$$\approx -0.0472 + 0.0536 + 0 = 0.0064$$

Step 3: Average them:

$$D_{\mathrm{JS}}(P \parallel Q) = \frac{1}{2}(0.0064 + 0.0064) = 0.0064 \text{ [nats]}$$

Exercise (Numerical Calculation of JS)

For $P = (0.2, 0.5, 0.3)$ and $Q = (0.1, 0.7, 0.2)$ from the KL exercise, find $D_{\mathrm{JS}}(P \parallel Q)$.

4.6 Definition of Jensen – Shannon (JS) divergence

Answer

Mixture: $M = \frac{1}{2}(P + Q) = (0.15, 0.6, 0.25)$.

Answer

Mixture: $M = \frac{1}{2}(P + Q) = (0.15, 0.6, 0.25)$.

KLs:

$$\begin{aligned} D_{\mathrm{KL}}(P \parallel M) &= 0.2 \log \frac{0.2}{0.15} + 0.5 \log \frac{0.5}{0.6} + 0.3 \log \frac{0.3}{0.25} \\ &\approx 0.0575 - 0.0912 + 0.0547 = 0.021 \\ D_{\mathrm{KL}}(Q \parallel M) &= 0.1 \log \frac{0.1}{0.15} + 0.7 \log \frac{0.7}{0.6} + 0.2 \log \frac{0.2}{0.25} \\ &\approx -0.0405 + 0.1079 - 0.0446 = 0.0228 \end{aligned}$$

Answer

Mixture: $M = \frac{1}{2}(P + Q) = (0.15, 0.6, 0.25)$.

KLs:

$$\begin{aligned}
D_{\mathrm{KL}}(P \parallel M) &= 0.2 \log \frac{0.2}{0.15} + 0.5 \log \frac{0.5}{0.6} + 0.3 \log \frac{0.3}{0.25} \\
&\approx 0.0575 - 0.0912 + 0.0547 = 0.021 \\
D_{\mathrm{KL}}(Q \parallel M) &= 0.1 \log \frac{0.1}{0.15} + 0.7 \log \frac{0.7}{0.6} + 0.2 \log \frac{0.2}{0.25} \\
&\approx -0.0405 + 0.1079 - 0.0446 = 0.0228
\end{aligned}$$

JS Divergence:

$$D_{\mathrm{JS}}(P \parallel Q) = \frac{1}{2}(0.021 + 0.0228) \approx 0.0219 \text{ [nats]}$$

4.6 Definition of Jensen – Shannon (JS) divergence

Similar to KL, JS divergence can be extended to language models by considering either the **joint distribution** over sequences or by averaging over a **dataset.**

4.6 Definition of Jensen – Shannon (JS) divergence

Similar to KL, JS divergence can be extended to language models by considering either the **joint distribution** over sequences or by averaging over a **dataset**.

A similar **chain rule** also holds, decomposing the total JS divergence into a sum of expected conditional JS divergences at each position.

$$D_{\mathrm{JS}}\big(P^{(n)} \parallel Q^{(n)}\big) = \sum_{i=1}^{n} \Big\{ \tfrac{1}{2}\, \mathbb{E}_{P^{(i-1)}}\big[D_{\mathrm{KL}}(P|\text{prefix}) \parallel M|\text{prefix})\big] + \tfrac{1}{2}\, \mathbb{E}_{Q^{(i-1)}}\big[D_{\mathrm{KL}}(Q|\text{prefix}) \parallel M|\text{prefix})\big] \Big\}$$

Summary

5. Summary

Let's summarize the key takeaways from today's lecture.

- We organized the motivation for **model compression**: reducing the scale of a model while preserving its input-output relationship as a function. We looked at methods like low-precision arithmetic, quantization, and distillation.

5. Summary

Let's summarize the key takeaways from today's lecture.

- We organized the motivation for **model compression**: reducing the scale of a model while preserving its input-output relationship as a function. We looked at methods like low-precision arithmetic, quantization, and distillation.
- We quantified the **difference between probabilistic language models** before and after modification using information-theoretic divergences.

5. Summary

Let's summarize the key takeaways from today's lecture.

- We organized the motivation for **model compression**: reducing the scale of a model while preserving its input-output relationship as a function. We looked at methods like low-precision arithmetic, quantization, and distillation.
- We quantified the **difference between probabilistic language models** before and after modification using information-theoretic divergences.
- We showed that from a few natural axioms, the **KL divergence** is uniquely derived as the measure of divergence. We also introduced its symmetrized version, the **JS divergence**.

References i

[1] Rishiraj Acharya.
Why maybe we're measuring llm compression wrong.
`https://huggingface.co/blog/rishiraj/kld-guided-quantization`, 2025.
Blog article.

[2] Cristian Buciluă, Rich Caruana, and Alexandru Niculescu-Mizil.
Model compression.
In Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD), pages 535–541, 2006.

[3] Thomas M. Cover and Joy A. Thomas.
Elements of Information Theory.
Wiley-Interscience, 2 edition, 2006.

[4] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean.
Distilling the knowledge in a neural network.
In NIPS Deep Learning and Representation Learning Workshop, 2015.
arXiv:1503.02531.

[5] Itay Hubara, Matthieu Courbariaux, Daniel Soudry, Ran El-Yaniv, and Yoshua Bengio.
Quantized neural networks: Training neural networks with low precision weights and activations.
Journal of Machine Learning Research, 18(187):1–30, 2018.
Earlier version: arXiv:1609.07061 (2016).

[6] Benoit Jacob, Skirmantas Kligys, Bo Chen, et al.
Quantization and training of neural networks for efficient integer-arithmetic-only inference.
In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pages 2704–2713, 2018.

[7] Solomon Kullback and Richard A. Leibler.
On information and sufficiency.
Annals of Mathematical Statistics, 22(1):79–86, 1951.

[8] Jianhua Lin.
Divergence measures based on the shannon entropy.
IEEE Transactions on Information Theory, 37(1):145–151, 1991.

[9] Paulius Micikevicius, Sharan Narang, Jonah Alben, et al.
Mixed precision training.
In International Conference on Learning Representations (ICLR) Workshop, 2018.
arXiv:1710.03740.