

Bias in Stochastic Gradient Methods

Chih-Jen Lin

National Taiwan Univ.



MBZUAI



Outline

- 1 Introduction
- 2 Stochastic gradient and the unbiasedness condition
- 3 A investigation on in-batch sampling, a non-uniform sampling scheme
- 4 Bias for the log-softmax loss
- 5 Efficient training for log-sum-exponential
- 6 Conclusions

Outline

- 1 Introduction
- 2 Stochastic gradient and the unbiasedness condition
- 3 A investigation on in-batch sampling, a non-uniform sampling scheme
- 4 Bias for the log-softmax loss
- 5 Efficient training for log-sum-exponential
- 6 Conclusions

Introduction

- Stochastic gradient methods are widely used in training deep neural networks.
- The design follows from the principle of

$$\mathbb{E} [\text{Stochastic Gradient}] = \text{Full Gradient}.$$

- However, in many recent applications, it is difficult to maintain this property.
- We call that a bias occurs.

Introduction (Cont'd)

- In this talk, we will give a general overview of this issue.
- We focus on similarity learning, for which the bias issue frequently occurs.

Similarity Learning

- Similarity learning finds a model to determine whether any two paired entities are similar or not.
- An example is multi-label classification.
 - The two entities are an instance (e.g., document) and a label (e.g., a tag).
 - The model predicts a score about the relationship between the instance and the label.

Similarity Learning (Cont'd)

- Assume there are m left entities and n right entities in the training set
- We use the following table for their similarity:

	C_1	C_2	$\dots$	$\dots$	C_n
R_1	1	0	0	1	0
$\vdots$	0	0	1	1	0
R_m	0	1	1	0	0

- We have that 1 is similar (i.e., positive) and 0 otherwise (i.e., negative).

Similarity Learning (Cont'd)

- If for multi-label classification, R_1 is an instance with two labels.
- After using the training set to obtain a model, we can predict the similarity of new pairs.
- For example, in multi-label classification, we use the model to predict which labels (from C_1 to C_n) are relevant to a new instance (e.g., R_{m+1}).

Similarity Learning (Cont'd)

- In practice, the set of labels may also change from training to prediction stages.
- Multi-label classification is a classic example. There are many new applications.
- An important one is contrastive learning.
- We have similar (text, image) pairs as positive instances.

Similarity Learning (Cont'd)

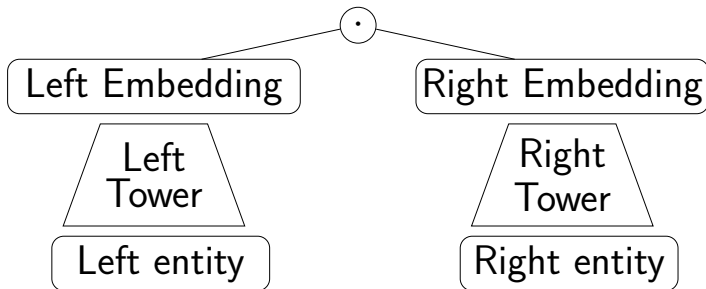
- It is like that in the label matrix,
 - diagonal entries are “1,” while
 - all others are “0.”

	C_1	C_2	$\dots$	$\dots$	$\dots$
R_1	1	0	0	0	0
R_2	0	1	0	0	0
$\vdots$	0	0	1	0	0

Two-Tower Model

- Similarity learning commonly involves a **two-tower model**.
- We have $\hat{Y}_{ij}$ as our estimated similarity of (i, j) pair.

$$\hat{Y}_{ij} = \text{Left Embedding} \odot \text{Right Embedding}$$



Two-Tower Model (Cont'd)

- The two towers independently encode two entities into embeddings.
- Similarity score $\hat{Y}_{ij}$ is computed from the resulting embeddings, typically through an inner product $\odot$.
- Higher scores indicate greater similarity.

Minimizing the Training Loss

- If Y_{ij} is the ground-truth label of the (i, j) pair, then we hope that our estimation $\hat{Y}_{ij}$ is close to Y_{ij} .
- To achieve this, we can design a loss function and minimize the training loss.
- For example, a loss function can encourage

$$\hat{Y}_{ij} \approx Y_{ij} \quad (1)$$

or

$$\hat{Y}_{ij} \gg \hat{Y}_{ik} \text{ if } Y_{ij} = 1 \text{ and } Y_{ik} = 0.$$

Minimizing the Training Loss (Cont'd)

- We end up with minimizing a function of all model weights.
- For example, from (1), we can minimize

$$L(\boldsymbol{\theta}) = \frac{1}{mn} \sum_{i=1}^m \sum_{j=1}^n \ell(Y_{ij}, \hat{Y}_{ij}). \quad (2)$$

Here $\hat{Y}_{ij}$ is a function

$$\hat{Y}_{ij}(\boldsymbol{\theta}; \text{input}),$$

and $\boldsymbol{\theta}$ includes parameters of the two towers.

Outline

- 1 Introduction
- 2 Stochastic gradient and the unbiasedness condition
- 3 A investigation on in-batch sampling, a non-uniform sampling scheme
- 4 Bias for the log-softmax loss
- 5 Efficient training for log-sum-exponential
- 6 Conclusions

Stochastic Gradient Descent

- Recall that the full objective $L(\theta)$ in (2) sums over all mn pairs:

$$[m] \times [n] = \{(i, j) : i \in [m], j \in [n]\},$$

where

$$[m] = \{1, \dots, m\} : \text{all row indices}$$

$$[n] = \{1, \dots, n\} : \text{all column indices}$$

- At each training step, we sample a batch of index pairs

$$\mathbb{B} \subseteq [m] \times [n].$$

Stochastic Gradient Descent (Cont'd)

- We then compute the stochastic gradient.

$$\frac{\partial}{\partial \boldsymbol{\theta}} \frac{1}{|\mathbb{B}|} \sum_{(i,j) \in \mathbb{B}} \ell_{ij}(Y_{ij}, \hat{Y}_{ij}). \quad (3)$$

- The stochastic gradient serves as an estimator of the full gradient and should satisfy **the unbiasedness condition**:

$$\mathbb{E} [\text{Stochastic Gradient}] = \text{Full Gradient},$$

Stochastic Gradient Descent (Cont'd)

- If the unbiasedness condition does not hold, the behavior of stochastic gradient methods becomes unclear.
- Sometimes using a biased stochastic gradient is beneficial, but in general we expect some issues.

Stochastic Gradient for Two-Tower Models

- We explain why biased stochastic gradients can easily arise in two-tower models
- As only a small portion of mn pairs are positive, easily the selected batch **may not have any positive pair**.
- Therefore, for training two-tower models, uniform sampling is not effective.
- So researchers proposed various non-uniform sampling strategies.

Stochastic Gradient for Two-Tower Models

- However, the non-uniform strategies change the data distribution
- Without appropriate correction, they causes that

$$\mathbb{E} [\text{Stochastic Gradient}] \neq \text{Full Gradient}.$$

- We briefly describe some of our recent research results on the bias issue.

Outline

- 1 Introduction
- 2 Stochastic gradient and the unbiasedness condition
- 3 A investigation on in-batch sampling, a non-uniform sampling scheme
- 4 Bias for the log-softmax loss
- 5 Efficient training for log-sum-exponential
- 6 Conclusions

Study on In-Batch Sampling

- In-batch sampling is a popular way to ensure positive pairs in a batch.

- Consider a 4-by-4 label matrix, where positive pairs are

$$\mathbb{O} = \{(R_1, C_1), (R_2, C_2), (R_2, C_4), (R_3, C_4), (R_4, C_1), (R_4, C_3), (R_4, C_4)\}$$

R_1	1	0	0	0
R_2	0	1	0	1
R_3	0	0	0	1
R_4	1	0	1	1
	C_1	C_2	C_3	C_4

Study on In-Batch Sampling (Cont'd)

- In-batch sampling involves two steps:
 - First, we uniformly sample a subset $\hat{\mathbb{O}}$ from the set $\mathbb{O}$ of all positive pairs. For example,

$$\hat{\mathbb{O}} = \{(R_2, C_2), (R_2, C_4), (R_4, C_3)\}.$$

- Then, we construct a batch using all $|\hat{\mathbb{O}}|^2$ combinations of the left and right entities appearing in $\hat{\mathbb{O}}$.

Study on In-Batch Sampling (Cont'd)

$$\hat{\mathbb{O}} \Rightarrow \begin{array}{c} R_2 \\ R_2 \\ R_4 \end{array} \begin{array}{|c|c|c|} \hline 1 & 0 & 0 \\ \hline 0 & 1 & 0 \\ \hline 0 & 0 & 1 \\ \hline \end{array} \begin{array}{c} \\ \\ C_2 C_4 C_3 \end{array}$$

- Pairs in $\hat{\mathbb{O}}$ are put in diagonal as positive, while all off-diagonal entries are treated as negative.
- The setting ensures positive pairs in a batch, but causes new issues.
- For example, there are the two (R_2, C_2) pairs in the batch; one is **positive** while the other is **negative**.

Study on In-Batch Sampling (Cont'd)

R_2	1	0	0
R_2	0	1	0
R_4	0	0	1
	C_2	C_4	C_3

- In-batch sampling **over-samples** pairs from rows/columns with more positives than others, though we do not get into details

Study on In-Batch Sampling (Cont'd)

- In an AISTATS 2026 paper (Liu et al., 2026), we study the in-batch sampling technique and have the following main results:
- Under the point-wise setting, i.e.,

$$L(\theta) = \frac{1}{mn} \sum_{i=1}^m \sum_{j=1}^n \ell(Y_{ij}, \hat{Y}_{ij}),$$

we proved that in-batch sampling causes a biased stochastic gradient.

Study on In-Batch Sampling (Cont'd)

- We derived a “modified” stochastic gradient, so that the expectation is the same as the full gradient.
- Experiments show that the bias causes worse performance than our new way.

Outline

- 1 Introduction
- 2 Stochastic gradient and the unbiasedness condition
- 3 A investigation on in-batch sampling, a non-uniform sampling scheme
- 4 Bias for the log-softmax loss**
- 5 Efficient training for log-sum-exponential
- 6 Conclusions

Softmax Function

- The softmax loss is also popular for two-tower models.

$$-\frac{1}{|\mathbb{O}|} \sum_{(i,j) \in \mathbb{O}} \log \frac{e^{\hat{Y}_{ij}}}{\sum_{j'=1}^n e^{\hat{Y}_{ij'}}}, \quad (4)$$

where $\mathbb{O}$ is the set of all positive pairs in the label matrix.

- Here

$$\frac{e^{\hat{Y}_{ij}}}{\sum_{j'=1}^n e^{\hat{Y}_{ij'}}} = \frac{1}{\sum_{j'=1}^n e^{\hat{Y}_{ij'} - \hat{Y}_{ij}}}$$

is called the softmax function.

Softmax Loss (Cont'd)

- The concept is that our estimation $\hat{Y}_{ij} \in (-\infty, \infty)$ should satisfy that,

$$\hat{Y}_{ij'} - \hat{Y}_{ij} \begin{cases} \rightarrow -\infty & \text{if } j' \neq j, \\ = 0 & \text{if } j' = j. \end{cases}$$

- Thus a successful prediction on a similar (i, j) pair gives the

$$\text{loss} \approx \log \frac{1}{e^0 + e^{-\infty} + \dots + e^{-\infty}} = \log 1 = 0.$$

Softmax Loss (Cont'd)

- To apply stochastic gradient methods, we select a batch $\mathbb{B} \subset \mathbb{O}$ and calculate the gradient of

$$-\frac{1}{|\mathbb{B}|} \sum_{(i,j) \in \mathbb{B}} \log \frac{e^{\hat{Y}_{ij}}}{\sum_{j'=1}^n e^{\hat{Y}_{ij'}}}. \quad (5)$$

- As n is large, calculating the denominator is expensive.
- Note that for each $\hat{Y}_{ij'}$ we must go through one tower of complicated network.

Softmax Loss (Cont'd)

- To reduce cost, we replace

$$\sum_{j'=1}^n e^{\hat{Y}_{ij'}}$$

with a value involving

$$\hat{Y}_{ij}, \hat{Y}_{i\tau_1}, \dots, \hat{Y}_{i\tau_k}, \quad (6)$$

where $\tau_1, \dots, \tau_k$ are randomly sampled from $\{1, \dots, n\} \setminus \{j\}$.

Softmax Loss (Cont'd)

- Let us use a function

$$s(\hat{Y}_{ij}, \hat{Y}_{i\tau_1}, \dots, \hat{Y}_{i\tau_k})$$

to represent the value in the denominator.

Bias of Softmax Function

- We ask if it is possible to have a function $s()$ so that

$$E_{\tau_1, \dots, \tau_k} \left[\frac{e^{\hat{Y}_{ij}}}{s(\hat{Y}_{ij}, \hat{Y}_{i\tau_1}, \dots, \hat{Y}_{i\tau_k})} \right] = \frac{e^{\hat{Y}_{ij}}}{e^{\hat{Y}_{ij}} + \sum_{j' \neq j} e^{\hat{Y}_{ij'}}}.$$

- In a NeurIPS 2025 paper (Lin et al., 2025), we proved that this is not possible.

Bias of Softmax Function (Cont'd)

- Therefore, we cannot get an unbiased estimate for the log-softmax loss.
- This is different from the earlier situation of point-wise losses.
- Our results suggest that future work should focus on **bias reduction** rather than **bias elimination**.

Outline

- 1 Introduction
- 2 Stochastic gradient and the unbiasedness condition
- 3 A investigation on in-batch sampling, a non-uniform sampling scheme
- 4 Bias for the log-softmax loss
- 5 Efficient training for log-sum-exponential
- 6 Conclusions

The Need of Efficient Algorithms

- An easy remedy to reduce bias is to **sample more elements** – a practice done in industry as they have more resources.
- However, for some large applications this way is not viable.
- So we still need to develop efficient algorithms to address the bias issue.

Log-sum-exponential

- Let's rewrite the objective of using the log-softmax loss as

$$\begin{aligned}
 & -\frac{1}{|\mathbb{O}|} \sum_{(i,j) \in \mathbb{O}} \log \frac{1}{\sum_{j'=1}^n e^{\hat{Y}_{ij'} - \hat{Y}_{ij}}}, \\
 & = \frac{1}{|\mathbb{O}|} \sum_{(i,j) \in \mathbb{O}} \log \sum_{j'=1}^n e^{\hat{Y}_{ij'} - \hat{Y}_{ij}},
 \end{aligned}$$

which is in a form of **log-sum-exponential**.

- Many problems including contrastive learning can be written in this form.

Log-sum-exponential (Cont'd)

- Now let's consider a more general form

$$\min_{\mathbf{w}} \frac{1}{n} \sum_{i=1}^n \log \left(\sum_{k=1}^K \exp(s_{ik}(\mathbf{w})) \right),$$

where $s_{ik}(\cdot)$ is a function reflecting i versus k .

- Through **conjugate transformation**, it is equivalent to

$$\min_{\mathbf{w}, \boldsymbol{\nu}} \frac{1}{n} \sum_{i=1}^n \left(\sum_{k=1}^K \exp(s_{ik}(\mathbf{w}) - \nu_i) + \nu_i \right).$$

- A nice thing is that because we no longer have **log**, stochastic gradient is **unbiased**.

Log-sum-exponential (Cont'd)

- This problem has **two blocks of variables w and v** .
- We can consider alternating block minimization.
- However, the problem has certain structures/properties, so we cannot just do simple block minimization (details not discussed)
- In ICLR 2026 (Wei et al., 2026a) and ICML 2026 (Wei et al., 2026b), we developed effective techniques.
- More research is still ongoing.

Outline

- 1 Introduction
- 2 Stochastic gradient and the unbiasedness condition
- 3 A investigation on in-batch sampling, a non-uniform sampling scheme
- 4 Bias for the log-softmax loss
- 5 Efficient training for log-sum-exponential
- 6 Conclusions

Conclusions

- Through this presentation we see that biased stochastic gradient occurs in many situations.
- The seriousness varies according to applications.
- We still need deep investigation on this subject.