Decoupled Contrastive Learning & Debiased Contrastive Learning

presenter: Shen Yuan

中国人民大學 RENMIN UNIVERSITY OF CHINA

高瓴人工智能学院 Gaoling School of Artificial Intelligence

- Understanding Contrastive Learning
- Decoupled Contrastive Learning
- Debiased Contrastive Learning
- Summary

What is Contrastive Learning?

Contrastive learning is a machine learning technique used to learn the general features of a dataset **without labels** by teaching the model which data points are similar or different.

What is Contrastive Learning?

Contrastive learning is a machine learning technique used to learn the general features of a dataset **without labels** by teaching the model which data points are similar or different.

What is Contrastive Learning?

The aim of Contrastive Learning is to **learn an encoder** $f(\cdot)$ such that:

similarity(f(),f())

>

similarity(f(),f())

How does Contrastive Learning Work?

In this part, we will focus on **SimCLR**, one of the contrastive learning approaches proposed by the Google Brain Team.

How does Contrastive Learning Work?

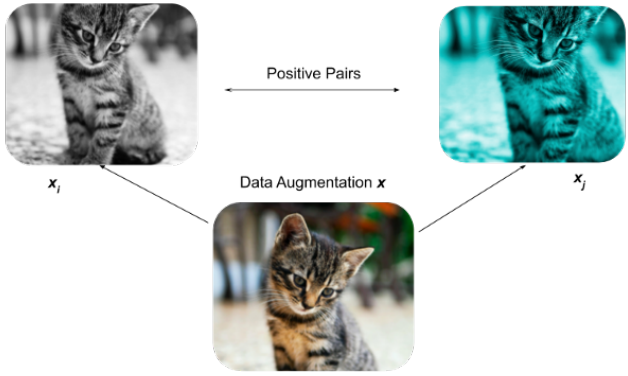
In this part, we will focus on **SimCLR**, one of the contrastive learning approaches proposed by the Google Brain Team.

This framework consists of three basic steps:

- data augmentaion
- encoding
- Loss minimization of representations

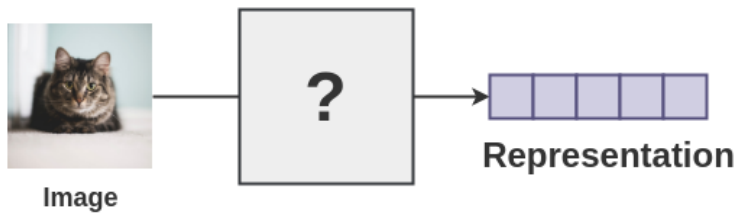
SimCLR - Data Augmentation

For each image in dataset, we can perform augmentation operations (i.e. crop, resize, recolor, etc.) to get **two** augmented views.
We want the model to learn that these two images are “similar” because they are actually different versions (or views) of the same image.



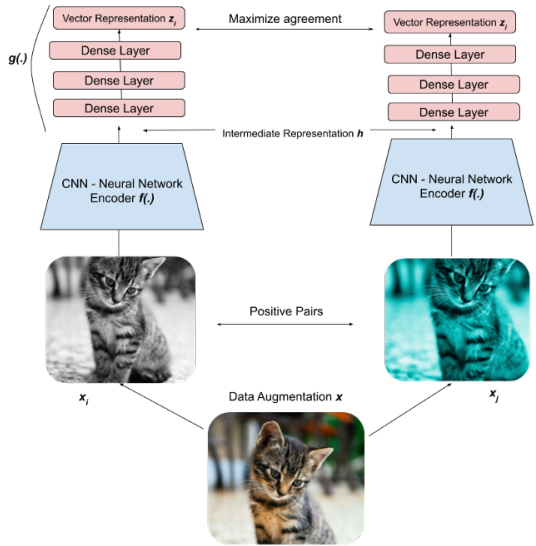
SimCLR - Encoding

Then, we can feed these two images into some deep learning models (i.e. encoder $f(\cdot)$) to extract **vector representations** for each image.



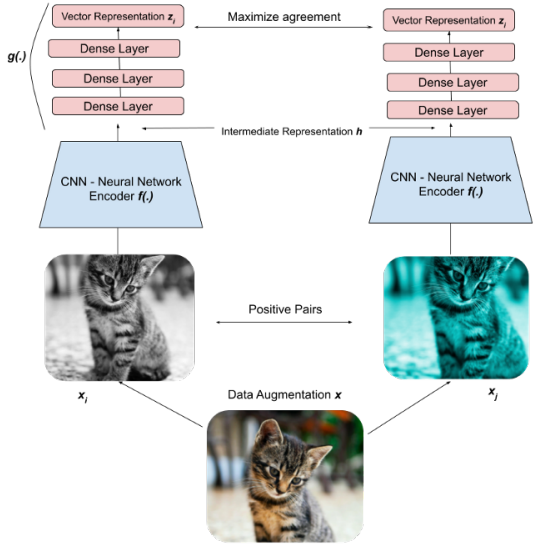
SimCLR - Encoding

- The framework SimCLR uses **ResNet** as the encoder $f(\cdot)$ to encode two images as vector representations (i.e., $\boldsymbol{h} = \boldsymbol{f}(\boldsymbol{x})$).



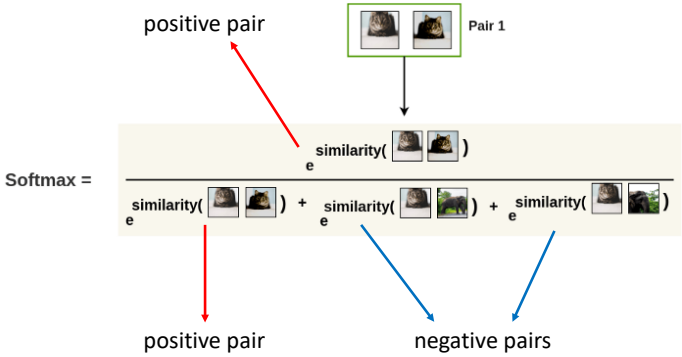
SimCLR - Encoding

- The framework SimCLR uses **ResNet** as the encoder $f(\cdot)$ to encode two images as vector representations (i.e., $\boldsymbol{h} = \boldsymbol{f}(\boldsymbol{x})$).
- The output of the CNN is then inputted to a set of Dense Layers called the **projection head**, $\boldsymbol{z} = \boldsymbol{g}(\boldsymbol{h}) / \|\boldsymbol{g}(\boldsymbol{h})\|_2$ to transform the data into another space. This extra step is empirically shown to improve performance.



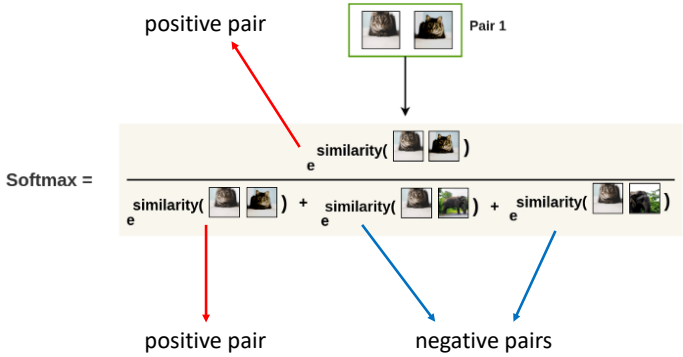
SimCLR - Loss Minimization of Representations

We can compute the probability that the two augmented images are similar by **softmax** function.



SimCLR - Loss Minimization of Representations

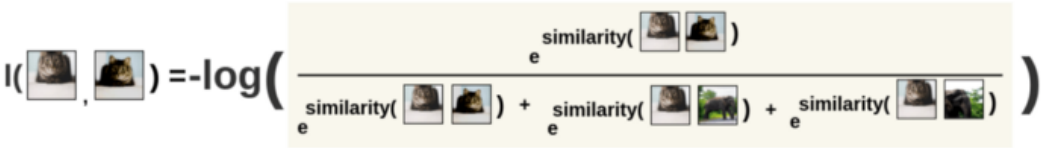
We can compute the probability that the two augmented images are similar by **softmax** function.



Negative pairs are created by combining our augmented images with **all** of the other images in our batch.

SimCLR - Loss Minimization of Representations

Finally, the loss function of the two augmented images can be illustrated as follows



Similarity

Generally, we compute the similarity between two vectors by **cosine similarity**,

$$\text{similarity}(\boldsymbol{z_i}, \boldsymbol{z_j}) = \frac{\langle \boldsymbol{z_i}, \boldsymbol{z_j} \rangle}{\|\boldsymbol{z_i}\| \|\boldsymbol{z_j}\|}$$

SimCLR - More Mathematical Notations!

Reformulate the loss function in a more mathematical form:

SimCLR - More Mathematical Notations!

Reformulate the loss function in a more mathematical form:

- given a batch of N samples (e.g. images), $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$

SimCLR - More Mathematical Notations!

Reformulate the loss function in a more mathematical form:

- given a batch of N samples (e.g. images), $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$
- let $\boldsymbol{x}_i^{(1)}, \boldsymbol{x}_i^{(2)}$ be two augmented views of the sample $\boldsymbol{x}_i$ and B be the set of all of the augmented views in the batch, i.e. $B = \{\boldsymbol{x}_i^{(k)} | k \in \{1, 2\}, i = 1, \ldots, N\}$

SimCLR - More Mathematical Notations!

Reformulate the loss function in a more mathematical form:

- given a batch of N samples (e.g. images), $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$
- let $\boldsymbol{x}_i^{(1)}, \boldsymbol{x}_i^{(2)}$ be two augmented views of the sample $\boldsymbol{x}_i$ and B be the set of all of the augmented views in the batch, i.e. $B = \{\boldsymbol{x}_i^{(k)} | k \in \{1, 2\}, i = 1, \ldots, N\}$
- each of the views $\boldsymbol{x}_i^{(k)}$ is sent into the same encoder network f and the output $\boldsymbol{h}_i^{(k)} = f(\boldsymbol{x}_i^{(k)})$ is then projected by a normalized MLP projector that $\boldsymbol{z}_i^{(k)} = g(\boldsymbol{h}_i^{(k)}) / \|g(\boldsymbol{h}_i^{(k)})\|$

SimCLR - More Mathematical Notations!

Reformulate the loss function in a more mathematical form:

- given a batch of N samples (e.g. images), $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N\}$
- let $\boldsymbol{x}_i^{(1)}, \boldsymbol{x}_i^{(2)}$ be two augmented views of the sample $\boldsymbol{x}_i$ and B be the set of all of the augmented views in the batch, i.e. $B = \{\boldsymbol{x}_i^{(k)} | k \in \{1, 2\}, i = 1, \ldots, N\}$
- each of the views $\boldsymbol{x}_i^{(k)}$ is sent into the same encoder network f and the output $\boldsymbol{h}_i^{(k)} = f(\boldsymbol{x}_i^{(k)})$ is then projected by a normalized MLP projector that $\boldsymbol{z}_i^{(k)} = g(\boldsymbol{h}_i^{(k)}) / \|g(\boldsymbol{h}_i^{(k)})\|$
- For each augmented view $\boldsymbol{x}_i^{(k)}$, SimCLR solves a classification problem by using all the rest of views in B as targets, and assigns the only positive label to $\boldsymbol{x}_i^{(u)}$, where $u \neq k$.

SimCLR - More Mathematical Notations!

SimCLR creates a cross-entropy loss function $L_i^{(k)}$ for each view $\boldsymbol{x}_i^{(k)}$, and the overall loss function is $L = \sum_{k\in\{1,2\},i=1,\ldots,N} L_i^{(k)}$

$$L_i^{(k)} = -\log \frac{\exp(\langle \boldsymbol{z}_i^{(1)}, \boldsymbol{z}_i^{(2)} \rangle / \tau)}{\exp(\langle \boldsymbol{z}_i^{(1)}, \boldsymbol{z}_i^{(2)} \rangle / \tau) + \sum_{k\in\{1,2\},\, j=1,\ldots,N,\, j\neq i} \exp(\langle \boldsymbol{z}_i^{(k)}, \boldsymbol{z}_j^{(l)} \rangle / \tau)} \tag{1}$$

- Understanding Contrastive Learning
- Decoupled Contrastive Learning
- Debiased Contrastive Learning
- Summary

Decoupled Contrastive Learning - Motivation

There exists a **negative-positive coupling (NPC)** multiplier $q_{B,i}^{(1)}$ in the gradient of $L_i^{(1)}$:

$$
\begin{cases}
-\nabla_{\boldsymbol{z}_i^{(1)}} L_i^{(1)} = \dfrac{q_{B,i}^{(1)}}{\tau}\left[\boldsymbol{z}_i^{(2)} - \sum_{l\in\{1,2\},\, j=1,\ldots,N,\, j\neq i} \dfrac{\exp(\langle \boldsymbol{z}_i^{(1)}, \boldsymbol{z}_j^{(l)}\rangle/\tau)}{\sum_{q\in\{1,2\},\, j=1,\ldots,N,\, j\neq i}\exp(\langle \boldsymbol{z}_i^{(1)}, \boldsymbol{z}_j^{(q)}\rangle/\tau)} \cdot \boldsymbol{z}_j^{(l)}\right] \\
-\nabla_{\boldsymbol{z}_i^{(2)}} L_i^{(1)} = \dfrac{q_{B,i}^{(1)}}{\tau} \cdot \boldsymbol{z}_i^{(1)} \\
-\nabla_{\boldsymbol{z}_j^{(l)}} L_i^{(1)} = -\dfrac{q_{B,i}^{(1)}}{\tau} \dfrac{\exp(\langle \boldsymbol{z}_i^{(1)}, \boldsymbol{z}_j^{(l)}\rangle/\tau)}{\sum_{q\in\{1,2\},\, j=1,\ldots,N,\, j\neq i}\exp(\langle \boldsymbol{z}_i^{(1)}, \boldsymbol{z}_j^{(q)}\rangle/\tau)} \cdot \boldsymbol{z}_i^{(1)}
\end{cases}
\quad (2
$$

Decoupled Contrastive Learning - Motivation

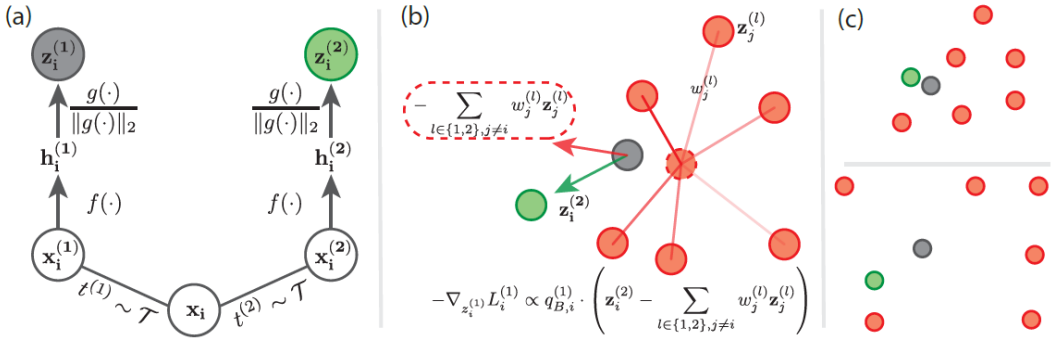
There exists a **negative-positive coupling (NPC)** multiplier $q_{B,i}^{(1)}$ in the gradient of $L_i^{(1)}$:

$$\begin{cases} -\nabla_{\boldsymbol{z}_i^{(1)}} L_i^{(1)} = \frac{q_{B,i}^{(1)}}{\tau}\left[\boldsymbol{z}_i^{(2)} - \sum_{l\in\{1,2\},\, j=1,\ldots,N,\, j\neq i} \frac{\exp(\langle \boldsymbol{z}_i^{(1)}, \boldsymbol{z}_j^{(l)}\rangle/\tau)}{\sum_{q\in\{1,2\},\, j=1,\ldots,N,\, j\neq i}\exp(\langle \boldsymbol{z}_i^{(1)}, \boldsymbol{z}_j^{(q)}\rangle/\tau)} \cdot \boldsymbol{z}_j^{(l)}\right] \\ -\nabla_{\boldsymbol{z}_i^{(2)}} L_i^{(1)} = \frac{q_{B,i}^{(1)}}{\tau} \cdot \boldsymbol{z}_i^{(1)} \\ -\nabla_{\boldsymbol{z}_j^{(l)}} L_i^{(1)} = -\frac{q_{B,i}^{(1)}}{\tau} \frac{\exp(\langle \boldsymbol{z}_i^{(1)}, \boldsymbol{z}_j^{(l)}\rangle/\tau)}{\sum_{q\in\{1,2\},\, j=1,\ldots,N,\, j\neq i}\exp(\langle \boldsymbol{z}_i^{(1)}, \boldsymbol{z}_j^{(q)}\rangle/\tau)} \cdot \boldsymbol{z}_i^{(1)} \end{cases} \quad (2$$

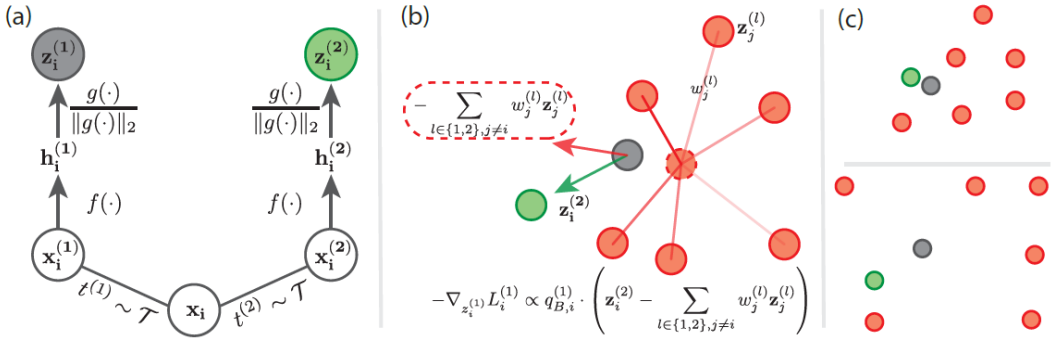
where the NPC multiplier $q_{B,i}^{(1)}$ is:

$$q_{B,i}^{(1)} = 1 - \frac{\exp(\langle \boldsymbol{z}_i^{(1)}, \boldsymbol{z}_i^{(2)}\rangle/\tau)}{\exp(\langle \boldsymbol{z}_i^{(1)}, \boldsymbol{z}_i^{(2)}\rangle/\tau) + \sum_{q\in\{1,2\},\, j=1,\ldots,N,\, j\neq i}\exp(\langle \boldsymbol{z}_i^{(1)}, \boldsymbol{z}_j^{(q)}\rangle/\tau)} \quad (3)$$

Negative-positive Coupling (NPC) reduces efficiency



Negative-positive Coupling (NPC) reduces efficiency



$$q_{B,i}^{(1)} = 1 - \frac{\exp(\langle \boldsymbol{z}_i^{(1)}, \boldsymbol{z}_i^{(2)} \rangle / \tau)}{\exp(\langle \boldsymbol{z}_i^{(1)}, \boldsymbol{z}_i^{(2)} \rangle / \tau) + \sum_{q \in \{1,2\},\, j=1,\ldots,N,\, j \neq i} \exp(\langle \boldsymbol{z}_i^{(k)}, \boldsymbol{z}_j^{(q)} \rangle / \tau)}$$

Decoupled contrastive learning loss

If we remove the NPC multiplier $q_{B,i}^{(k)}$, we reach a decoupled contrastive learning loss $L_{DC} = \sum_{k\in\{1,2\},\ i=1,\dots,N} L_{DC,i}^{(k)}$, where $L_{DC,i}^{(k)}$ is :

$$
\begin{aligned}
L_{DC,i}^{(k)} &= -\log \frac{\exp(\langle \boldsymbol{z}_i^{(1)}, \boldsymbol{z}_i^{(2)} \rangle / \tau)}{\cancel{\exp(\langle \boldsymbol{z}_i^{(1)}, \boldsymbol{z}_i^{(2)} \rangle / \tau)} + \sum_{k\in\{1,2\},\, j=1,\dots,N,\, j\neq i} \exp(\langle \boldsymbol{z}_i^{(k)}, \boldsymbol{z}_j^{(l)} \rangle / \tau)} \\
&= -\langle \boldsymbol{z}_i^{(1)}, \boldsymbol{z}_i^{(2)} \rangle / \tau + \log \sum_{k\in\{1,2\},\, j=1,\dots,N,\, j\neq i} \exp(\langle \boldsymbol{z}_i^{(k)}, \boldsymbol{z}_j^{(l)} \rangle / \tau)
\end{aligned}
\tag{4}
$$

Experiments

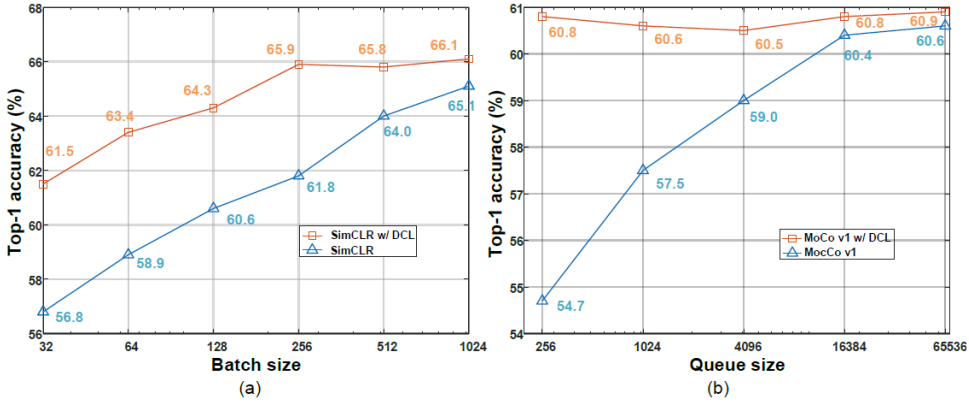


Figure 3: Comparisons on ImageNet-1K with/without DCL under different numbers of (a): batch sizes for SimCLR and (b): queues for MoCo. Without DCL, the top-1 accuracy significantly drops when batch size (SimCLR) or queues (MoCo) becomes very small. Note that the temperature τ of SimCLR is 0.1, and the temperature τ of MoCo is 0.07 in the comparison.

Experiments

Table 1: Comparisons with/without DCL under different numbers of batch sizes from 32 to 512. Results show the effectness of DCL on four widely used benchmarks. The performance of DCL keeps steadier than the SimCLR baseline while the batch size is varied.

Architecture@epoch	ResNet-18@200 epoch									
Dataset	ImageNet-100 (linear)					STL10 (kNN)				
Batch Size	32	64	128	256	512	32	64	128	256	512
SimCLR	74.2	77.6	79.3	80.7	81.3	74.1	77.6	79.3	80.7	81.3
SimCLR w/ DCL	**80.8**	**82.0**	**81.9**	**83.1**	**82.8**	**82.0**	**82.8**	**81.8**	**81.2**	**81.0**
Dataset	CIFAR10 (kNN)					CIFAR100 (kNN)				
Batch Size	32	64	128	256	512	32	64	128	256	512
SimCLR	78.9	80.4	81.1	81.4	81.3	49.4	50.3	51.8	52.0	52.4
SimCLR w/ DCL	**83.7**	**84.4**	**84.4**	**84.2**	**83.5**	**51.1**	**54.3**	**54.6**	**54.9**	**55.0**
Architecture@epoch	ResNet-50@500 epoch									
SimCLR	82.2	-	88.5	-	89.1	49.8	-	59.9	-	61.1
SimCLR w/ DCL	**86.1**	-	**89.9**	-	**90.3**	**54.3**	-	**61.6**	-	**62.2**

- Understanding Contrastive Learning
- Decoupled Contrastive Learning
- Debiased Contrastive Learning
- Summary

Debiased Contrastive Learning - Motivation

When we create negative pairs (x, x^-) after sampling positive pair (x, x^+), we actually made an assumption that **all the rest of images** are considered as negative samples **NOT** "similar" with the given view x.

Debiased Contrastive Learning - Motivation

When we create negative pairs (x, x^-) after sampling positive pair (x, x^+), we actually made an assumption that **all the rest of images** are considered as negative samples **NOT** "similar" with the given view x.

However, it's possible that x^- is actually similar to x.

Debiased Contrastive Learning - Motivation

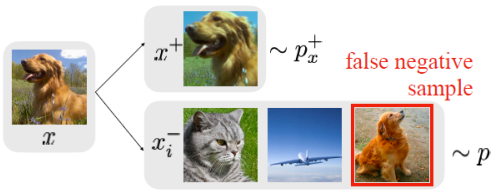


Figure 1: **"Sampling bias":** The common practice of drawing negative examples x_i^- from the data distribution $p(x)$ may result in x_i^- that are actually similar to x.

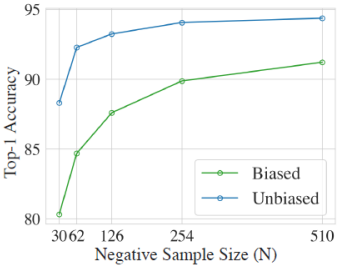


Figure 2: **Sampling bias leads to performance drop:** Results on CIFAR-10 for drawing x_i^- from $p(x)$ (biased) and from data with different labels, i.e., truly semantically different data (unbiased).

Preliminary

- Assume an **underlying** set of discrete latent classes $\mathcal{C}$ that represent semantic content, i.e. similar pairs (x, x^+) have the same latent class.

Preliminary

- Assume an **underlying** set of discrete latent classes $\mathcal{C}$ that represent semantic content, i.e. similar pairs (x, x^+) have the same latent class.
- Denoting the distribution over classes by $\rho(c)$, and the class probabilities $\rho(c) = \tau^+$ are **uniform**(i.e. $\rho(c) = 1/\|\mathcal{C}\|, c \in \mathcal{C}$), and let $\tau^- = 1 - \tau^+$ be the probability of observing any different class.

Preliminary

- Assume an **underlying** set of discrete latent classes $\mathcal{C}$ that represent semantic content, i.e. similar pairs (x, x^+) have the same latent class.
- Denoting the distribution over classes by $\rho(c)$, and the class probabilities $\rho(c) = \tau^+$ are **uniform**(i.e. $\rho(c) = 1/\|\mathcal{C}\|, c \in \mathcal{C}$), and let $\tau^- = 1 - \tau^+$ be the probability of observing any different class.
- $p_x^+(x') = \tau^+$ is the probability of observing x' as a positive example for x and $p_x^-(x') = \tau^-$ is the probability of a negative example.

Unbiased Contrastive Loss

For contrastive learning, the ideal loss to optimize would be

$$L^N_{\text{Unbiased}}(f) = \mathbb{E}_{\substack{x\sim p,\ x^+\sim p^+_x,\\ x^-_i\sim p^-_x}} \left[-\log \frac{e^{f(x)^\top f(x^+)}}{e^{f(x)^\top f(x^+)} + \sum_{i=1}^N e^{f(x)^\top f(x^-_i)}} \right] \tag{5}$$

Unbiased Contrastive Loss

For contrastive learning, the ideal loss to optimize would be

$$L^N_{\text{Unbiased}}(f) = \mathbb{E}_{\substack{x\sim p,\ x^+\sim p^+_x,\\ x^-_i\sim p^-_x}} \left[-\log \frac{e^{f(x)^\top f(x^+)}}{e^{f(x)^\top f(x^+)} + \sum_{i=1}^N e^{f(x)^\top f(x^-_i)}} \right] \tag{5}$$

However, $x^-_i \sim p^-_x$ is not accessible.

$$L^N_{\text{biased}}(f) = \mathbb{E}_{\substack{x\sim p,\ x^+\sim p^+_x,\\ x^-_i\sim p}} \left[-\log \frac{e^{f(x)^\top f(x^+)}}{e^{f(x)^\top f(x^+)} + \sum_{i=1}^N e^{f(x)^\top f(x^-_i)}} \right] \tag{6}$$

Debiased Contrastive Loss

Next, we derive a loss that is closer to the ideal L^N_{Unbiased}.

Debiased Contrastive Loss

Next, we derive a loss that is closer to the ideal L^N_{Unbiased}.

We can decompose the data distribution as

$$p(x') = \tau^+ p^+_x(x') + \tau^- p^-_x(x')$$

Debiased Contrastive Loss

Next, we derive a loss that is closer to the ideal L^N_{Unbiased}.

We can decompose the data distribution as

$$
\begin{aligned}
p(x') &= \tau^+ p^+_x(x') + \tau^- p^-_x(x') \\
p^-_x(x') &= (p(x') - \tau^+ p^+_x(x'))/\tau^- \\
p^-_x(x') &= \frac{1}{\tau^-} p(x') - \frac{\tau^+}{\tau^-} p^+_x(x')
\end{aligned}
\tag{7}
$$

Debiased Contrastive Loss

Next, we derive a loss that is closer to the ideal L^N_{Unbiased}.

We can decompose the data distribution as

$$\begin{aligned} p(x') &= \tau^+ p_x^+(x') + \tau^- p_x^-(x') \\ p_x^-(x') &= (p(x') - \tau^+ p_x^+(x'))/\tau^- \\ p_x^-(x') &= \frac{1}{\tau^-} p(x') - \frac{\tau^+}{\tau^-} p_x^+(x') \end{aligned} \tag{7}$$

Hence, $x_i^- \sim p_x^-$ can be replaced with that $x_i^- \sim p$ with probability $\frac{1}{\tau^-}$ and $x_i^- \sim p_x^+$ with probability $\frac{\tau^+}{\tau^-}$.

Debiased Contrastive Loss

Hence, $x_i^- \sim p_x^-$ can be replaced with that $x_i^- \sim p$ with probability $\frac{1}{\tau^-}$ and $x_i^- \sim p_x^+$ with probability $\frac{\tau^+}{\tau^-}$.

$$L_{\text{Unbiased}}^N(f) = \mathbb{E}_{\substack{x\sim p,\ x^+\sim p_x^+,\\ x_i^-\sim p_x^-}} \left[-\log \frac{e^{f(x)^\top f(x^+)}}{e^{f(x)^\top f(x^+)} + \sum_{i=1}^N e^{f(x)^\top f(x_i^-)}} \right] \tag{8}$$

Debiased Contrastive Loss

Hence, $x_i^- \sim p_x^-$ can be replaced with that $x_i^- \sim p$ with probability $\frac{1}{\tau^-}$ and $x_i^- \sim p_x^+$ with probability $\frac{\tau^+}{\tau^-}$.

$$L_{\text{Unbiased}}^N(f) = \mathbb{E}_{\substack{x\sim p,\ x^+\sim p_x^+,\\ x_i^-\sim p_x^-}} \left[-\log \frac{e^{f(x)^\top f(x^+)}}{e^{f(x)^\top f(x^+)} + \sum_{i=1}^N e^{f(x)^\top f(x_i^-)}} \right] \tag{8}$$

Leveraging Bernoulli distribution,

$$\frac{1}{(\tau^-)^N} \sum_{k=0}^N \binom{N}{k} (-\tau^+)^k \mathbb{E}_{\substack{x\sim p,\ x^+\sim p_x^+,\\ \{x_i^-\}_{i=1}^k\sim p_x^+,\\ \{x_i^-\}_{i=k+1}^N\sim p}} \left[-\log \frac{e^{f(x)^\top f(x^+)}}{e^{f(x)^\top f(x^+)} + \sum_{i=1}^N e^{f(x)^\top f(x_i^-)}} \right] \tag{9}$$

Debiased Contrastive Loss

We can use the empirical estimate $\tilde{L}^N_{\text{Debiased}}$ to approximate the real loss L^N_{Debiased}:

$$
\begin{aligned}
L^N_{\text{Debiased}} &= \mathbb{E}_{\substack{x\sim p,\ x^+\sim p^+_x,\\ \{x^-_i\}_{i=1}^N\sim p^-_x}}\left[-\log\frac{e^{f(x)^\top f(x^+)}}{e^{f(x)^\top f(x^+)}+\sum_{i=1}^N e^{f(x)^\top f(x^-_i)}}\right] \\
\tilde{L}^N_{\text{Debiased}} &= \mathbb{E}_{\substack{x\sim p,\\ x^+\sim p^+_x}}\left[-\log\frac{e^{f(x)^\top f(x^+)}}{e^{f(x)^\top f(x^+)}+N\left(\frac{1}{\tau^-}\mathbb{E}_{x^-\sim p}\left[e^{f(x)^\top f(x^-)}\right]-\frac{\tau^+}{\tau^-}\mathbb{E}_{v\sim p^+_x}\left[e^{f(x)^\top f(v)}\right]\right)}\right]
\end{aligned}
\tag{10}
$$

Debiased Contrastive Loss

With N samples $\{u_i\}_{i=1}^N$ from p and M samples $\{v_i\}_{i=1}^M$ from p_x^+, we estimate the expectation of the second term in the denominator as

$$g(x, \{u_i\}_{i=1}^N, \{v_i\}_{i=1}^M) = \max\left\{\frac{1}{\tau^-}\left(\frac{1}{N}\sum_{i=1}^N e^{f(x)^\top f(u_i)} - \tau^+ \frac{1}{M}\sum_{i=1}^M e^{f(x)^\top f(v_i)},\ e^{-1/t}\right)\right\} \quad (11)$$

Debiased Contrastive Loss

With N samples $\{u_i\}_{i=1}^N$ from p and M samples $\{v_i\}_{i=1}^M$ from p_x^+, we estimate the expectation of the second term in the denominator as

$$g(x, \{u_i\}_{i=1}^N, \{v_i\}_{i=1}^M) = \max\left\{\frac{1}{\tau^-}\left(\frac{1}{N}\sum_{i=1}^N e^{f(x)^\top f(u_i)} - \tau^+ \frac{1}{M}\sum_{i=1}^M e^{f(x)^\top f(v_i)},\ e^{-1/t}\right)\right\} \quad (11)$$

We constrain the estimator g to be greater than its theoretical minimum $e^{-1/t} \leq e^{f(x)^\top f(x_i^-)}$ to prevent calculating the logarithm of a negative number.

Debiased Contrastive Loss

With N samples $\{u_i\}_{i=1}^N$ from p and M samples $\{v_i\}_{i=1}^M$ from p_x^+, we estimate the expectation of the second term in the denominator as

$$g(x, \{u_i\}_{i=1}^N, \{v_i\}_{i=1}^M) = \max\left\{\frac{1}{\tau^-}\left(\frac{1}{N}\sum_{i=1}^N e^{f(x)^\top f(u_i)} - \tau^+ \frac{1}{M}\sum_{i=1}^M e^{f(x)^\top f(v_i)},\ e^{-1/t}\right)\right\} \quad (11)$$

We constrain the estimator g to be greater than its theoretical minimum $e^{-1/t} \leq e^{f(x)^\top f(x_i^-)}$ to prevent calculating the logarithm of a negative number.

$$L_{\text{Debiased}}^{N,M}(f) = \mathbb{E}_{\substack{x\sim p,\ x^+\sim p_x^+,\\ \{u_i\}_{i=1}^N\sim p,\\ \{v_i\}_{i=1}^M\sim p_x^+}}\left[-\log\frac{e^{f(x)^\top f(x^+)}}{e^{f(x)^\top f(x^+)} + Ng(x, \{u_i\}_{i=1}^N, \{v_i\}_{i=1}^M)}\right] \quad (12)$$

Error bound

$$\left|\tilde{L}^{N}_{\text{Debiased}}(f) - L^{N,M}_{\text{Debiased}}(f)\right| \leq \frac{e^{3/2}}{\tau^{-}}\sqrt{\frac{\pi}{2N}} + \frac{e^{3/2}\tau^{+}}{\tau^{-}}\sqrt{\frac{\pi}{2M}} \tag{13}$$

This inequality bounds the error due to finite N and M as decreasing with rate $\mathcal{O}(N^{-1/2} + M^{-1/2})$.

Experiments - Image Classification

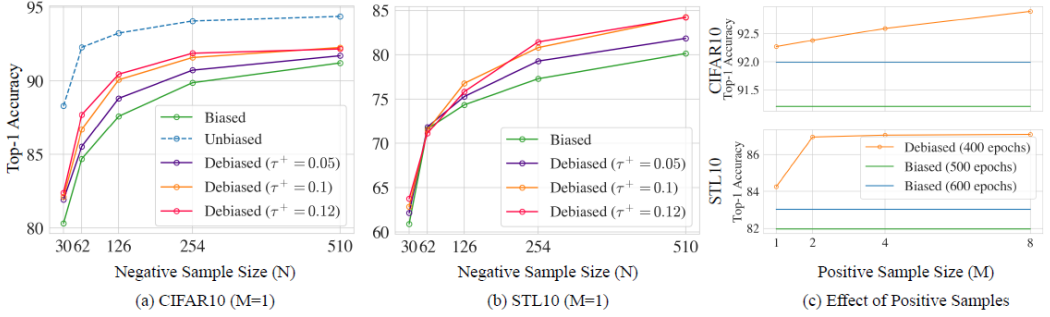


Figure 4: **Classification accuracy on CIFAR10 and STL10.** (a,b) Biased and Debiased ($M = 1$) SimCLR with different negative sample size N where $N = 2(BatchSize - 1)$. (c) Comparison with biased SimCLR with 50% more training epochs (600 epochs) while fixing the training epoch for Debiased ($M \geq 1$) SimCLR to 400 epochs.

Experiments - Reinforcement Learning

Objective	Finger Spin	Cartpole Swingup	Reacher Easy	Cheetah Run	Walker Walk	Ball in Cup Catch
Biased (CURL)	310±33	850±20	918±96	266±41	623±120	928±47
Debiased Objective with $M = 1$						
Debiased ($\tau^+ = 0.01$)	324±34	843±30	**927±99**	**310±12**	**626±82**	937±9
Debiased ($\tau^+ = 0.05$)	308±57	**866±7**	916±114	284±20	613±22	945±13
Debiased ($\tau^+ = 0.1$)	**364±36**	860±4	868±177	302±29	594±33	**951±11**
Debiased Objective with $M = 2$						
Debiased ($\tau^+ = 0.01$)	330±10	858±10	754±179	286±20	**746±93**	949±5
Debiased ($\tau^+ = 0.1$)	**381±24**	864±6	904±117	303±5	671±75	**957±5**

Table 3: **Scores achieved by biased and debiased objectives.** Our debiased objective outperforms the biased baseline (CURL) in all the environments, and often has smaller variance.

Summary

Although the motivations of these two paper are different, they have similar solution.

$$L^N(f) = \mathbb{E}_{\substack{x\sim p,\ x^+\sim p_x^+,\\ x_i^-\sim p}}\left[-\log \frac{e^{f(x)^\top f(x^+)}}{e^{f(x)^\top f(x^+)} + \sum_{i=1}^N e^{f(x)^\top f(x_i^-)}}\right] \tag{14}$$