idea INTERNATIONAL DIGITAL ECONOMY ACADEMY
粤港澳大湾区数字经济研究院

DisCo-CLIP: A Distributed Contrastive Loss for Memory Efficient CLIP Training

Yihao Chen, Xianbiao Qi, Jianan Wang, Lei Zhang

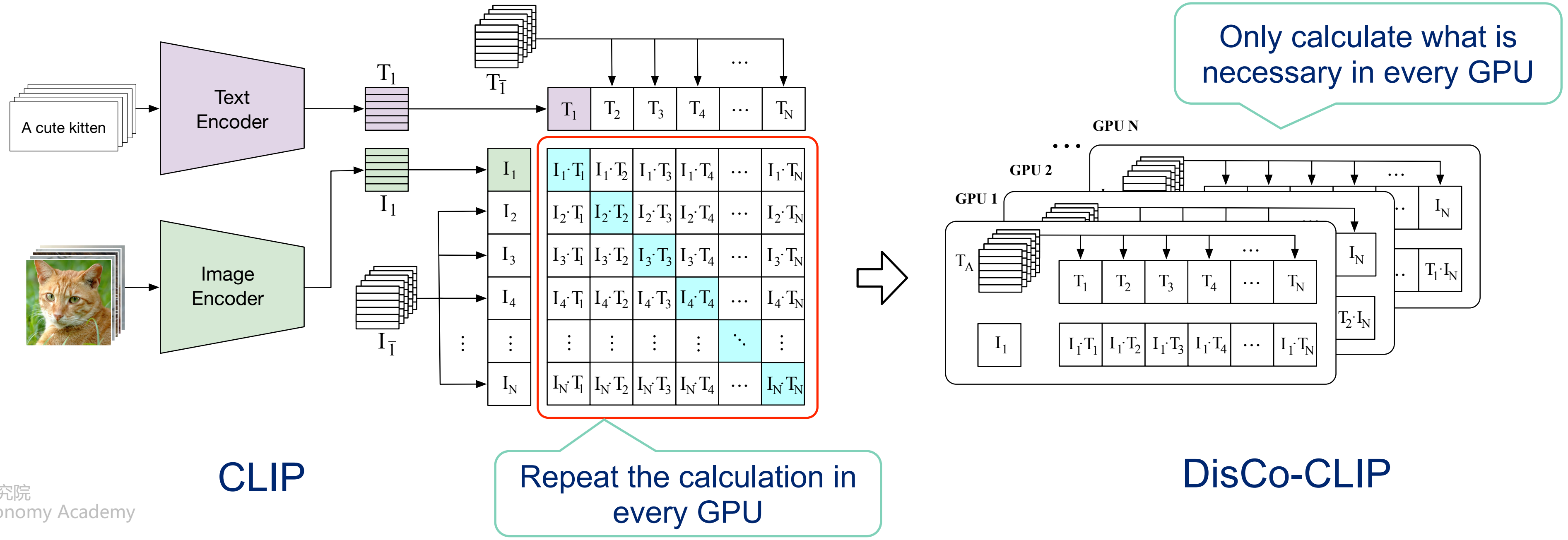
International Digital Economy Academy (IDEA)

Paper Tag: THU-PM-195

The world needs a few good IDEAs

Overview

- We decompose the Contrastive Loss and reduce unnecessary redundant computations. We effectively reducing the computational complexity from $O(N^2)$ to $O(N^2/n)$.
- Our work is mathematically equivalent to the original contrastive loss computation.
- We have observed that larger batch sizes are highly effective for contrastive learning.



Motivation

Train CLIP more efficiently !

- Enable training with large batch size for better performance.
- Enable training with limited GPU resources.

- The original CLIP contrastive loss function:

$$\mathcal{L}_d = \mathcal{L}_1(\mathrm{I}_A, \mathrm{T}_A) + \mathcal{L}_2(\mathrm{T}_A, \mathrm{I}_A)$$

Image-to-Text Contrastive Loss

Text-to-Image Contrastive Loss

Where I_A and T_A denote **all** image and text features.

➢ Each GPU calculates the full $\mathrm{I}_\mathrm{A} \times \mathrm{T}_\mathrm{A}$ similarity matrix for contrastive loss computation, causing huge computation waste.

➢ We split I_A and T_A as below:

$$\mathrm{I}_A = [\mathrm{I}_n, \ \mathrm{I}_{\overline{n}}],$$

$$\mathrm{T}_A = [\mathrm{T}_n, \ \mathrm{T}_{\overline{n}}]$$

Where I_n, T_n are image and text features on the n-th GPU, and $\mathrm{I}_{\overline{n}}$ and $\mathrm{T}_{\overline{n}}$ denote the collection of features on other GPUs.

➢ Then the $\mathcal{L}_d$ can be decomposed and rewritten as below:

$$\mathcal{L}_d = \mathcal{L}_1(\mathrm{I}_n, \mathrm{T}_A) + \mathcal{L}_1(\mathrm{I}_{\overline{n}}, \mathrm{T}_A) + \mathcal{L}_2(\mathrm{T}_n, \mathrm{I}_A) + \mathcal{L}_2(\mathrm{T}_{\overline{n}}, \mathrm{I}_A)$$

- It is clear that $\frac{\partial \mathcal{L}_1(\mathrm{I}_{\overline{n}}, \mathrm{T}_A)}{\partial \mathrm{I}_n}$ and $\frac{\partial \mathcal{L}_2(\mathrm{T}_{\overline{n}}, \mathrm{I}_A)}{\partial \mathrm{T}_n}$ do not include gradients.

- We can decompose the gradients $\frac{\partial \mathcal{L}_d}{\partial \mathrm{I}_n}$ and $\frac{\partial \mathcal{L}_d}{\partial \mathrm{T}_1}$ as below :

$$\frac{\partial \mathcal{L}_d}{\partial \mathrm{I}_n} = \frac{\partial \mathcal{L}_1(\mathrm{I}_n, \mathrm{T}_A)}{\partial \mathrm{I}_n} + \frac{\partial \mathcal{L}_2(\mathrm{T}_n, \mathrm{I}_A)}{\partial \mathrm{I}_n} + \frac{\partial \mathcal{L}_2(\mathrm{T}_{\overline{n}}, \mathrm{I}_A)}{\partial \mathrm{I}_n}$$

$$\frac{\partial \mathcal{L}_d}{\partial \mathrm{T}_n} = \frac{\partial \mathcal{L}_1(\mathrm{I}_n, \mathrm{T}_A)}{\partial \mathrm{T}_n} + \frac{\partial \mathcal{L}_2(\mathrm{T}_n, \mathrm{I}_A)}{\partial \mathrm{T}_n} + \frac{\partial \mathcal{L}_1(\mathrm{I}_{\overline{n}}, \mathrm{T}_A)}{\partial \mathrm{T}_n}$$

intra-GPU gradient **inter-GPU gradient**

- For **intra-GPU gradient** on the n-th GPU, we only need to calculate $\mathcal{L}_1(\mathrm{I}_n, \mathrm{T}_A)$ and $\mathcal{L}_2(\mathrm{T}_n, \mathrm{I}_A)$.

- For **inter-GPU gradient** on the n-th GPU, It would have already been calculated as **intra-GPU gradient** on other GPUs.

- In order to obtain $\frac{\partial \mathcal{L}_d}{\partial \mathrm{I}_n}$ and $\frac{\partial \mathcal{L}_d}{\partial \mathrm{T}_1}$, we use the *All_Reduce* operation to perform gradient communication and to collect all gradients with respect to I_n and T_n.

Pseudo-code

idea

Algorithm 1: Pseudo-code for DisCo-CLIP

```
# image_encoder:  ResNet or ViT
# text_encoder:  text Transformer
# img[b,H,W,C]: minibatch of images
# text[b,L]: minibatch of texts
# t:  temperature parameter
# D: feature dimension
# N: nums of GPU
# rank:  global rank

# extract feature representations
i_e = image_encoder(img) # [b,D]
t_e = text_encoder(text) # [b,D]

# gather features from all gpus
I_E = all_gather(i_e) # [N,b,D]
T_E = all_gather(t_e) # [N,b,D]

# scaled dot product similarities
logits_i = dot(I_E[rank], T_E)*t # [b,N*b]
logits_t = dot(T_E[rank], I_E)*t # [b,N*b]

# image-to-text and text-to-image losses
labels = arange(b) + b * rank
loss_i = cross_entropy_loss(logits_i, labels)
loss_t = cross_entropy_loss(logits_t, labels)

# loss backward
loss = (loss_i + loss_t) / 2
loss.backward()

# reduce gradients and losses from all gpus
I_E.grad = all_reduce(I_E.grad, op=AVG)
T_E.grad = all_reduce(T_E.grad, op=AVG)
Loss = all_reduce(loss, op=AVG)

# backbone backward
i_e.backward(I_E.grad[rank])
t_e.backward(T_E.grad[rank])
```

$\mathcal{L}_1(\mathrm{I}_n, \mathrm{T}_A)$

$\mathcal{L}_2(\mathrm{T}_n, \mathrm{I}_A)$

Calculate the **intra-GPU gradient** and the **inter-GPU gradient** of other GPUs.

Collect **intra-GPU gradient** and **inter-GPU gradient** from all GPUs

Experiments

Methods	GPUs	BS	Memory
CLIP	64× A100 40GB	32,768	27.4GB
CLIP	64× A100 40GB	65,536	OOM
DisCo-CLIP	64× A100 40GB	32,768	16.5GB
DisCo-CLIP	64× A100 40GB	**196,608**	38.1GB
DisCo-CLIP	**8**× A100 40GB	32,768	36.9GB

Table 3. Memory consumption of CLIP and DisCo-CLIP under different settings. "OOM" means out of memory.

Model	Data Sets	Epochs	Steps	Batch Size	INet [7]	INet-v2 [32]	INet-R [15]	INet-S [40]
CLIP [27]	CLIP WIT 400M	32	≈ 400 K	32,768	63.3	56.0	69.4	42.3
OpenCLIP [16, 34]	LAION-400M	32	≈ 400 K	32,768	62.9	55.1	73.4	49.4
DisCo-CLIP	LAION-400M*	32	≈ 400 K	32,768	63.2	55.2	73.4	50.6
DisCo-CLIP	LAION-400M*	32	≈ **200** K	**65,536**	**64.3**	**56.2**	**73.8**	**51.7**

Table 7. Comparison of DisCo-CLIP with vanilla CLIP and re-implemented CLIP by LAION group. We report top-1 zero-shot classification accuracy (%) on several data sets. All models are based on ViT-B/32. Our LAION-400M* is a 400M subset of LAION-2B. Training resource: 64 A100 40GB GPUs.

Conclusion

- Disco-CLIP is mathematically equivalent to the original contrastive loss computation.

- DisCo-CLIP can enable a much larger batch size for contrastive learning compared to CLIP, using the same GPU resource.

THANK YOU

中国深圳市福田区市花路5号
长富金茂大厦1号楼3901&57层，邮编518045

Room 3901 and 57/F , Building 1 ,
Chang Fu Jin Mao Tower , 5 Shihua Road ,
Futian District , Shenzhen , China , 518045

86-755-61610106 （3901）
86-755-83219017 （57楼）

www.idea.edu.cn