图灵奖大佬 Lecun 发表对比学习新作，比 SimCLR 更好用！

原创 Rukawa_Y 夕小瑶的卖萌屋 2021-11-17 12:05

文 | Rukawa_Y
编 | 智商掉了一地，Sheryc_王苏

> 比 SimCLR 更好用的 Self-Supervised Learning，一起来看看吧！

Self-Supervised Learning作为深度学习中的独孤九剑，当融汇贯通灵活应用之后，也能打败声名在外的武当太极剑。比如在NLP领域中，每当遇到文本分类的问题，BERT + funetuning的套路来应对，但是也正因为如此大家解决问题的思路出现固化。也正是这个原因，当本菜鸟第一次接触到Self-Supervised Learning这个概念时，就在项目中尝试应用Self-Supervised的SimCLR方法，但是却事与愿违，模型的预测效果并没有显著地提升，反而出现了一丢丢的下降，等厚着脸皮求助大佬后才明白，SimCLR对于模型效果的提升必须基于**大Batch Size**才会有效果。

而在近期，由 Yann Lecun 等人发表了一篇题为《Decouple Contrastive Learning》的论文，其中仔细分析了SimCLR和其他自监督学习模型所使用的InfoNCE损失函数，**仅仅对InfoNCE的表达式进行了一处修改，就大大缓解了InfoNCE对于大Batch Size的需求问题**，并在不同规模的Vision Benchmarks上均取得优于SimCLR的结果。

接下来就让我们跟随论文的思路，一起学习Decoupled Contrastive Learning吧。

论文标题

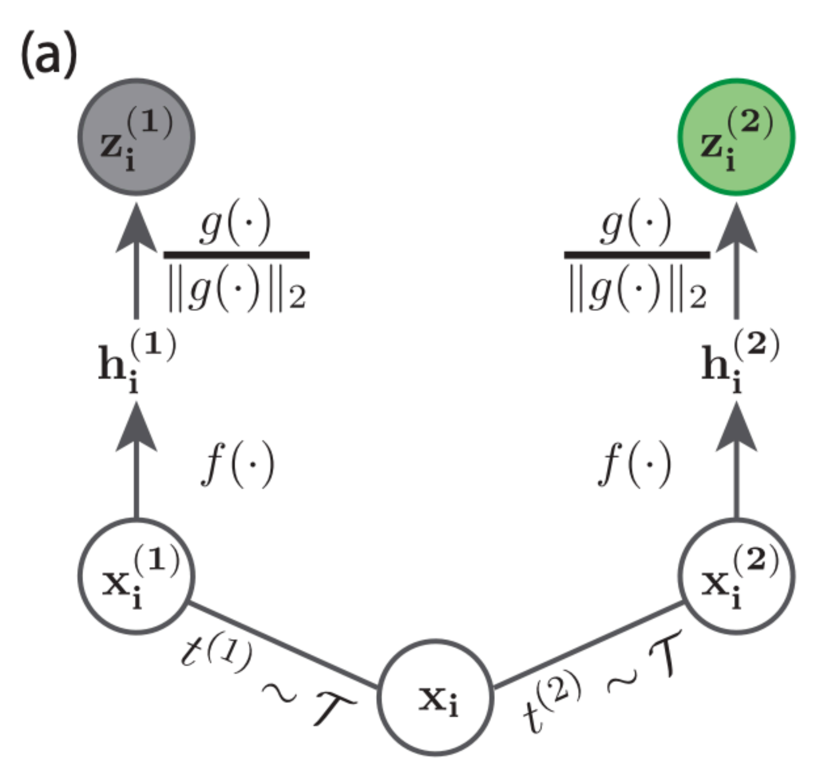
Decoupled Contrastive Learning

论文链接

https://arxiv.org/abs/2110.06848

1 对比学习中正负样本的解耦

作为本文的背景，我们先来介绍一下SimCLR的基本思想，它是对训练样本做数据增强（例如对于图像进行裁剪等），训练模型让同一图片增强后得到的表示相近，并互斥不同图片增强后的表示。



论文从SimCLR所使用的InfoNCE损失函数开始分析。InfoNCE对于其中一个样本x_i的增强数据$x_i^{(k)}$的InfoNCE损失函数$L_i^{(k)}$如下：

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

其中所使用的各个变量的意义分别为：

- $x_1, x_2, x_3, \cdots, x_N$为一个Batch中所使用的样本， N为Batch Size；
- $x_i^{(1)}, x_i^{(2)}$是样本x_i增强后的两个数据；
- $B = \{x_i^{(k)} | k \in \{1,2\}, i \in [1, N]\}$是对于Batch中所有样本增强后的数据集合；
- $h_i^{(k)} = f(x_i^{(k)})$是样本x_i的增强数据输入到Encode Layer中所对应的输出；
- $z_i^k = \frac{g(h_i^{(k)})}{\|g(h_i^{(k)})\|}$是归一化后的表示。

InfoNCE的损失函数分别求对于$z_i^{(1)}$，$z_i^{(2)}$和$z_j^{(l)}$的梯度：（这里作者对梯度进行了一定的变化，变化过程可参照论文附录的第一部分）

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

其中需要注意的是损失函数的梯度中均有一个**系数**$q_{B,i}^{(1)}$，这个系数导致**模型训练的梯度发生了放缩**。该系数的具体形式如下：

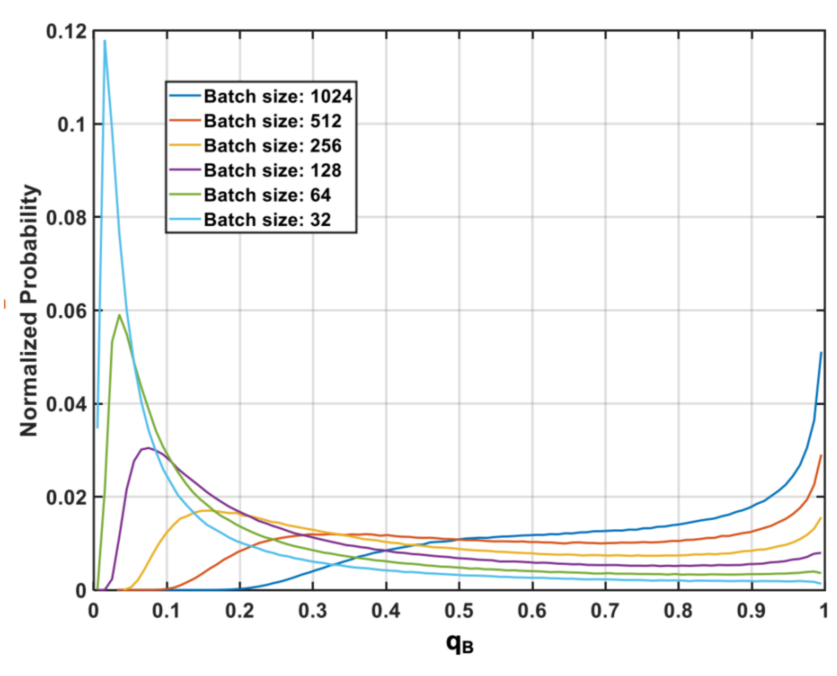
$$q_{B,i}^{(1)} = 1 - \frac{\exp(\langle \mathbf{z}_i^{(1)}, \mathbf{z}_i^{(2)} \rangle / \tau)}{\exp(\langle \mathbf{z}_i^{(1)}, \mathbf{z}_i^{(2)} \rangle / \tau) + \sum_{q \in \{1,2\}, j \in [\![1,N]\!], j \neq i} \exp(\langle \mathbf{z}_i^{(1)}, \mathbf{z}_j^{(q)} \rangle / \tau)}$$

作者将这个导致SimCLR模型梯度放缩的系数称为**Negative-Positive Coupling (NPC) Multiplier**，即**NPC乘数**。NPC乘数分子和分母上出现的$\exp(\langle z_i^{(1)}, z_i^{(2)} \rangle / \tau)$中衡量了正样本对的相似度，而分母上出现的$\exp(\langle z_i^{(1)}, z_j^{(q)} \rangle / \tau)$则衡量了负样本对的相似度。

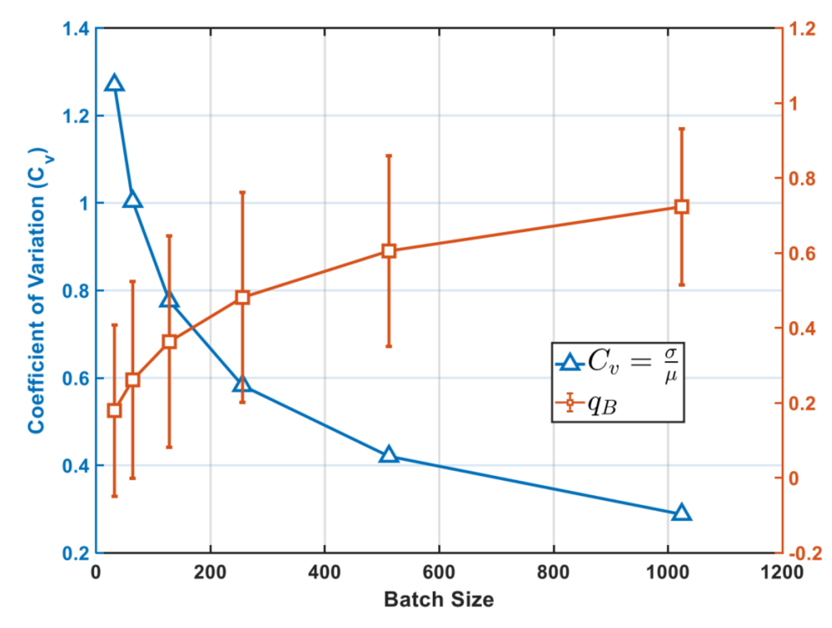
顾名思义，$q_{B,i}^{(1)}$**对训练的影响与正负样本的耦合有关**：当负样本较为分散时，正样本同样可能较为分散；反之，当正样本较为紧凑时，负样本同样可能较为紧凑。论文中对于不同情形下的NPC乘数进行了定性分析，总结如下：

- 当训练使用的正样本较分散时，负样本可能同样比较分散。此时**正样本为Hard Positive，负样本为Easy Negative**。这使得NPC乘数分子分母上的相似度同时减小，得到的小于1的NPC乘数会**减小Hard Positive带来的梯度幅度**。
- 当训练使用的负样本较紧凑时，正样本可能同样比较紧凑。此时**正样本为Easy Positive，负样本为Hard Negative**。这使得NPC乘数分子分母上的相似度同时增大，得到的小于1的NPC乘数会**减小Hard Negative带来的梯度幅度**。
- 当**Batch Size较小时**，分母上对Batch中负样本相似度的求和会受限于Batch Size，得到**更小的NPC乘数**，使得**梯度幅度进一步被减小**。

由此可见，**SimCLR对于大Batch Size的需求很可能来自于NPC乘数对于梯度的缩小。** Batch Size同NPC乘数q_B分布的具体的关系见下图：



从图中可以明显看出，Batch Size越小，q_B的分布越接近于$\delta(0)$；Batch Size越大，q_B的分布越接近于$\delta(1)$。同时作者还给出了q_B的均值和离散系数同Batch Size的关系，见下图：



可以看出，小的Batch Size使得q_B的均值减小，离散系数增大，从而使得训练过程中的梯度被大幅缩小。

综上所述，**SimCLR等自监督模型中对于大Batch Size的需求问题一定程度上来自于**q_B。论文的作者由此修改了InfoNCE的公式来消除q_B的影响，从而引出了本文的核心：Decoupled Contrastive Learning Loss。

2 Decoupled Contrastive Learning

既然NPC乘数的存在会使得梯度被缩小，那么**移除掉NPC乘数**不就能解决上面的问题了么？通过将导数中的NPC乘数移除，作者推导出了下面的损失函数。在这个损失函数中，**正负样本的耦合带来的梯度放缩被消去**，作者将该损失称为**Decoupled Contrastive Learning (DCL) Loss**，即**解耦对比损失函数**：

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

可见，Decoupled Constrive Learning中的损失直接**去掉了SimCLR损失函数分母中两个正样本对之间的相似度**，从而直接计算正样本对的相似度同所有负样本对相似度之和的比值。

Decoupled Contrastive Learning中所对应的梯度如下：

$$-\nabla_{z_i^{(1)}} L_i^{(1)} = \frac{1}{\tau}[z_i^{(2)} - \sum_{l \in \{1,2\}, j \in [1,N], j \neq i} \frac{\exp\langle z_i^{(1)}, z_i^{(2)} \rangle / \tau}{\sum_{q \in 1,2, j \in [1,N], j \neq i} \exp\langle z_i^{(1)}, z_j^{(q)} \rangle}]$$

$$-\nabla_{z_i^{(2)}} L_i^{(1)} = \frac{1}{\tau} \cdot z_i^{(1)}$$

$$-\nabla_{z_j^{(l)}} L_i^{(1)} = \frac{1}{\tau} \frac{exp(z_i^{(1)}, z_i^{(2)}) / \tau}{\sum_{q \in 1,2, j \in [1,N], j \neq i} exp(z_i^{(1)}, z_j^{(q)})} \cdot z_i^{(1)}$$

我们同样针对正负样本对耦合和Batch Size较小的情况，具体分析反向传播过程中的梯度：因为缺少了NPC这个系数的影响，当出现正负样本耦合的情况，正负样本比较分散（Hard Positive + Easy Negative）或者正负样本比较集中（Easy Positive + Hard Negative）反向传播过程中梯度幅度就不会减少，同时因为没有了NPC系数的存在，比较小的Batch Size也就不会使得梯度幅度变得很小。

综上所述，消去了NPC乘数的DCL损失函数能较SimCLR损失取得更好的效果，后面的实验结果也对此进行了证明。

同时论文的作者还提出了一种DCL损失的变形，即对DCL损失中衡量正样本对相似度的一项增加一个权重$w(z_i^{(1)}, z_i^{(2)})$。作者将其称为**DCLW损失**：

$$L_{DCW,i}^{(k)} = -w(\mathbf{z}_i^{(1)}, \mathbf{z}_i^{(2)})(\langle \mathbf{z}_i^{(1)}, \mathbf{z}_i^{(2)} \rangle / \tau) + \log \sum_{l \in \{1,2\}, j \in [\![1,N]\!], j \neq i} \exp(\langle \mathbf{z}_i^{(k)}, \mathbf{z}_j^{(l)} \rangle / \tau)$$

上式中，权重使用**负von Mises-Fisher权重函数**：

$$w(z_i^{(1)}, z_i^{(2)}) = 2 - \frac{\exp(\langle z_i^{(1)}, z_i^{(2)} \rangle / \sigma)}{\mathbb{E}\left[\exp(\langle z_i^{(1)}, z_i^{(2)} \rangle) / \sigma\right]}$$

且$\mathbb{E}[w] = 1$。σ为参数，在后续实验中取0.5。这一权重使得**在出现Hard Positive时能增大其提供的训练信号**。显然，L_{DC}是L_{DCW}的一个特殊情况。

总结来说，DCL损失仅在SimCLR所采用的损失函数基础上采取了一些小的改动，使得模型能够在训练过程中也不要求大Batch Size，同时对正负样本对进行解耦。

3 实验结果

论文作者首先比较在不同的Batch Size下，使用DCL损失和InfoNCE损失的SimCLR在ImageNet、STL10、CIFAR10和CIFAR100数据集上的表现：

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**

可以发现**在不同的Batch Size上，DCL损失的效果均优于SimCLR**。同时，**Batch Size越小，DCL损失提供的性能提升越大**，这与先前的理论推导一致。

作者又比较了在Batch Size固定为256，epoch固定为200时的DCL损失和加权重的DCLW损失，结果如下：

Dataset	CIFAR10	CIFAR100	ImageNet-100	ImageNet-1K
SimCLR	81.8	51.8	79.3	61.8
DCL	84.2 (+3.1)	54.6 (+2.8)	81.9 (+2.6)	65.9 (+4.1)
DCLW	**84.8** (+3.7)	**54.9** (+3.1)	**82.8** (+3.5)	**66.9** (+5.1)

可以看出，**DCLW损失相较于DCL损失能进一步提升模型效果**，甚至在ImageNet-1K上能够以256的Batch Size超越SimCLR使用8192 Batch Size的结果，66.2%。可见，DCL和DCLW损失能够通过较小的改动解决SimCLR对于大Batch Size的依赖。

4 文章小结

本篇论文针对自监督学习中的SimCLR方法为何要求较大Batch Size的原因开始分析，提出了一种可以让自监督学习在较小的Batch Size上取得很好效果的loss函数，大幅降低了自监督学习的计算成本，使得自监督学习可以有更广泛的应用。

除此之外,本篇论文还分析了SimCLR中使用的loss函数在反向传播梯度计算中的问题时，提出的一种名为**正负样本耦合（Negative-Positive Coupling）** 现象，同时也给予了我们一定的启发，如果是同SimCLR中所用的InfoNCE形式不相同的loss函数，在计算梯度的时候，是否也会有正负样本耦合现象，或者说不仅仅有正负样本耦合的现象，还有例如对于不同正样本的，在不同负样本之间的负负样本耦合的现象等，如果能够分析出Self-Supervised Learning中不同方法可能存在不同的耦合现象，那么我们是否可以进一步地提升自监督模型的效果，这些都是值得我们去思考和探索的。