Opt4ML Course Project Proposal

Ryan King, Zhale Nowroozilarki

October 2022

1 Literature Review

In SimCLR [Chen et al., 2020], a self-supervised training method is developed to pretrain a backbone architecture for a family of tasks related to the input data. During pretraining, a backbone model and a projector are trained to minimize the contrastive objective which maximizes the similarity of positive samples while minimizing the similarity of negative samples. To generate positive samples, a single image is transformed into multiple views utilizing image specify augmentations such as random cropping, random color distortion, and Gaussian blur. All other images in a batch are considered to be the negative samples. Using this setup, SimCLR minimizes the following objective function:

$$\ell_{i,j} = log\frac{exp(sim(\mathbf{z_i}, \mathbf{z_j})/\tau)}{\sum_{k=1}^{2N} \mathbb{1}_{[k\neq i]} exp(sim(\mathbf{z_i}, \mathbf{z_k})/\tau)} \tag{1}$$

Where sim is the cosine similarity function, $\mathbf{z}$ is the latent image representation, τ is a temperature parameter and $\mathbb{1}_{[k\neq i]}$ is the indicator function for negative pairs. While SimCLR is capable of pretraining models with high performance on downstream tasks, as evaluated by semi-supervised and linear evaluation experiments, the use of two random variables in the contrastive loss makes the method sensitive to batch sizes.

To overcome this issue, Global Contrastive Learning [Yuan et al., 2022a] proposes a method of decoupling the positive and negative samples using the following objective function:

$$\ell_{i,j} = log\frac{exp(E(\mathcal{A}(x_i))^\intercal E(\mathcal{A}(x_i))/\tau)}{g(w; z_i, \mathcal{A}, \mathcal{S}_i)} \tag{2}$$

where

$$g(w; z_i, \mathcal{A}, \mathcal{S}_i) = \sum_{z\in\mathcal{S}_i} exp(E(\mathcal{A}(x_i))^\intercal E(z_i)/\tau)) \tag{3}$$

where $\mathcal{A}$ is a random augmentation of the image x_i, $E(\cdot)$ is the network, and $\mathcal{S}_i$ is every negative image. While this provides a method for dealing with the second random variable from the negative pairs, it is not computationally efficient to compute.

Self-supervised learning approaches usually require large batches to achieve an acceptable result. To overcome this challenge, the aforementioned global objective was proposed in SogCLR [Yuan et al., 2022b]. The global contrastive learning objective is further combined with an efficient stochastic algorithm to reduce the required batch size in the optimization process of models based on self-supervised learning approaches. A stochastic gradient estimator is utilized to calculate the gradients of the negative samples in the Global Contrastive Objective. This is accomplished by first calculating a moving average of the negative sample term:

$$u_{i,t} = (1-\gamma)u_{i,t} + \gamma\frac{1}{2|\mathcal{B}|}(g(w_t;x_i,\mathcal{A},\mathcal{B}_i) + g(w_t;x_i,\mathcal{A}',\mathcal{B}_i)) \tag{4}$$

The stochastic gradient estimator is then calculated as follows:

$$\mathbf{m}_t = -\frac{1}{B}\sum_{\mathbf{x}_i\in\mathcal{B}}\nabla(E(\mathcal{A}(x_i))^\intercal E(\mathcal{A}'(x_i)))+\frac{p_{i,t}}{n}\sum_{x\in\mathcal{D}}(g(w_t;x_i,\mathcal{A},\mathcal{B}_i)+g(w_t;x_i,\mathcal{A}',\mathcal{B}_i)) \tag{5}$$

Where $p_{i,t} = \frac{\tau}{u_{i,t-1}} = \nabla f(u_{i,t-1})$.

Existing semi-supervised learning methods utilize similarity scores for positive and negative pairs. However, these metrics cannot capture the variation over different distributions in the data. In [Rezaei et al., 2021], a self-supervised learning method is proposed to use the contrastive loss between the pairs as well as the contrastive divergence between the distributions of the current mini-batches using Bregman divergence. The total loss is defined as shown below where λ is a tunable parameter.

$$L_{total} = \lambda * L_{contrastive} + L_{divergence} \tag{6}$$

In all methods mentioned above, the negative samples are treated with equal weights. While this provides an easy solution, there is no clear reason why each negative pair should be weighted equally. However, searching for hard examples can be a computationally expensive task. Distributionally Robust Optimization (DRO) [Rahimian and Mehrotra, 2022], is a method for learning to reweigh hard examples. The DRO objective is as follows:

$$\min_w \max_p \frac{1}{n}\sum_{i=1}^{n} p_i \ell(w_i) \quad p = \{p_1, p_2, ..., p_n\} \in \Delta \tag{7}$$

where ℓ is an objective function and p_i are elements of a simplex. By maximizing the value of p_i, this objective can assign weights to samples with high loss values.

2 Plan for Proposed Work

We propose a method of adaptively learning the difficulty of pairs of features by optimizing the DRO object GCL. In the following sections, we describe how

DRO and GCL are utilized to reweigh pairs based on how difficult they are to separate. We further extend our method by proposing a training scheme for focusing on hard pairs while skipping easier ones. Finally, we demonstrate the impact of using the sample-wise divergence as well as distribution-wise divergence in the objective function of the training schema.

2.1 DRO and GCL

Although, image datasets used in the self-supervised learning benchmarks are mostly equally distributed over all the classes, in many other domains the labels are imbalanced (e.g., medical domain). Randomly selecting negative pairs for each anchor point will result in the same distribution of the input data where the majority class has more representatives among negative pairs. This is not an ideal case where classes have different importance. Furthermore, existing semi-supervised models require large batch sizes to be adequately accurate. However, this will cause the model training and optimization to be computationally expensive and time-consuming. Therefore, in order to minimize the number of required negative examples for faster learning, one could use a weighted version where the weights are tuned based on the distribution of different classes or their importance accordingly. In this work, we propose a method based on the combination of DRO and GCL. Finally, data streams can have completely different distributions in online settings; hence, requiring us to tune the weights to better construct the pairs.

We start with the formulation of the DRO objective with the quadratic term:

$$\min_{w} \max_{p} \frac{1}{N} \sum_{i=1}^{N} p_i \ell_i(w_i) - \lambda D(p_i, 1/N) \tag{8}$$

where D is the distance between two probability distributions and λ is a coefficient that controls the strength of the quadratic term. In the above function, ℓ_i is the GCL averaged over all pairs of images in Equation 2. For p, we use a separate term for each positive pair.

The model parameters w will be minimized using the Layer-wise Adaptive Rate Scaling (LARS) [You et al., 2017] while the values of p will be optimized with Stochastic Gradient Ascent. The step for the minimization and maximization step will be determined by the gradient at the current step.

2.2 Training On Hard Samples Only

We consider the model to have good features for define the difference between pairs when the value of p at that point is smaller than some predefined threshold value. In contrast, we the value of p is larger than that threshold value, the model hasn't learned features needed to distinguish the two pairs. Since it is unclear what the distribution of the feature space is, we would like our model to learn features that are important for learning hard examples.

It is because of this, we allow our model skip "easy" examples and train only on harder pairs. To be specific, during pretraining, when values of $p_{i,j}$ falls below a certain predefined threshold values β, those values will be skipped during training. We believe that this will allow our model to focus on learning features that are more important while reducing the computational cost of training the model.

2.3 Distribution-Wise Training

As mentioned before, pair-wise similarity metrics cannot capture the comprehensive variation in input data distributions in all settings. Furthermore, learning the distribution of the data can help us in detecting abnormalities and out-of-distribution cases. As a result, there is a need for training contrastive learning both based on pair-wise similarity and distribution-wise convergence. To do so, we further expand the Global Contrastive Optimization for these two components of contrastive loss. The divergence loss of two distributions p and q over z_i and z_j, the divergence loss between $P(x_i)$ and $q(x_j)$ are defined as below [Rezaei et al., 2021].

$$l_{div}(p(x_i), q(x_j)) = -log(exp(\Psi_{i,j}) / \sum_{m=1}^{n} exp(\Psi_{i,m})) \tag{9}$$

where Gaussian kernel $\Psi_{i,j} = exp(-D/2\sigma^2)$ converts Bregman divergence to similarity D and σ is an adjustable parameter. The total loss will be then calculated based on 6 and optimized by using equations 4 and 5 accordingly.

References

[Chen et al., 2020] Chen, T., Kornblith, S., Norouzi, M., and Hinton, G. (2020). A simple framework for contrastive learning of visual representations. In *Proceedings of the 37th International Conference on Machine Learning*, ICML'20. JMLR.org.

[Rahimian and Mehrotra, 2022] Rahimian, H. and Mehrotra, S. (2022). Frameworks and results in distributionally robust optimization. *Open Journal of Mathematical Optimization*, 3:1–85.

[Rezaei et al., 2021] Rezaei, M., Soleymani, F., Bischl, B., and Azizi, S. (2021). Deep bregman divergence for contrastive learning of visual representations. *arXiv preprint arXiv:2109.07455*.

[You et al., 2017] You, Y., Gitman, I., and Ginsburg, B. (2017). Scaling SGD batch size to 32k for imagenet training. *CoRR*, abs/1708.03888.

[Yuan et al., 2022a] Yuan, Z., Wu, Y., Qiu, Z.-H., Du, X., Zhang, L., Zhou, D., and Yang, T. (2022a). Provable stochastic optimization for global contrastive learning: Small batch does not harm performance.

[Yuan et al., 2022b] Yuan, Z., Wu, Y., Qiu, Z.-H., Du, X., Zhang, L., Zhou, D., and Yang, T. (2022b). Provable stochastic optimization for global contrastive learning: Small batch does not harm performance. In *International Conference on Machine Learning*, pages 25760–25782. PMLR.