loSSLess: A Novel Approach For Using Large Amounts of Unlabeled Data To Improve Image Classification Performance

Vidit Bhargava [1] **Nandhitha Raghuram** [1] **Vasudev Awatramani** [1]

Abstract

One of the main challenges in the field of Deep Learning is the access to accurate labeled data. The process of gathering such data can be expensive and cumbersome. Thus, it is important for models to gain the ability to learn with the least amount of supervision. In this paper, we explore a procedure which exploits a large number of unlabeled images to learn features and improve accuracy on a image classification downstream task. We combine the Barlow Twins and Pseudo-labeling to construct a model which is greater than the sum of its parts.

1. Introduction

Self-Supervised visual representation learning aims to learn useful features without relying on manual annotations. Recent state-of-the-art methods can be broadly categorized under *contrastive learning*, *clustering* and *distillation* paradigms, however the underlying theme in each of them is to learn representations that are invariant under different distortions. A typical strategy to achieve this notion is by using variants of Siamese Networks along with a mechanism to avoid a collapsed solution (such as constant representation). This results in maximizing the similarity of representations produced from various transformed versions of the sample.

Contrastive Learning approaches such as SIMCLR (Chen et al., 2020) define *positive* and *negative* pairs and rely on contrastive losses to remove the notion of instance classes such that directly compare image features. Such methods require selective sampling over a large number of negative samples, which can become computationally infeasible very quickly. Clustering methods like DEEPCLUSTER(Caron et al., 2018) avoid pairwise computations and aim to cluster the data while enforcing consistency between cluster assignments produced for distorted versions of the same image. However, such methods can suffer from collapsed solutions in the form of empty clusters (or all assignments correspond to a single cluster) and storing features when the number of clusters is much larger than the batch size. Distillation methods such as BOOTSTRAP YOUR OWN LATENT (BYOL)(Grill et al., 2020) involve use of asymmetric network architecture or asymmetric parametric updates to avoid trivial solutions.

The recently published paper, BARLOW TWINS employs the *redundancy reduction* principle to self-supervised learning. It utilizes the representations obtained by feeding two random augmentations of an image to a Siamese network to compute a cross-correlation matrix. The objective is to make this matrix as close to an identity matrix as possible. As described in further sections, BARLOW TWINS is conceptually simple and compared to previously described methods, does not require very large batch sizes, asymmetric mechanisms, momentum encoders, or non-differentiable operators.

In this paper, we propose a pipeline that combines BARLOW TWINS with a semi-supervised learning technique to performance on an image classification dataset leveraging large amount of unlabelled data. Overall, the pipeline consists of the following steps:

- Pre-training a custom ResNet-34 architecture using Barlow Twins.
- Utilizing the backbone obtained from Barlow and adding two fully-connected layers before training the model on the labeled training set
- Fine-tuning the model using additional samples obtained by most confident pseudo-labels.

2. Literature Review

Handcrafted pretext tasks: Many of the self-supervised learning methods were initially inspired by artificial pretext tasks that aimed at manipulating the input data to extract a supervised signal. Such tasks include relative patch prediction (Doersch et al., 2015), solving jigsaw puzzles (Noroozi & Favaro, 2016), and colorization (Zhang et al., 2016). (Kolesnikov et al., 2019) indicates that bigger network can generalize. However, pretext tasks rely on ad-hoc heuristics which restricts the generalization of learned features. Recent methods such as SWAV (Caron et al., 2021) apply distortions like `multi-crop` while

training which is similar to tasks like Jigsaw puzzle (Misra & van der Maaten, 2019).

Instance and Contrastive Learning: Instance-level classification considers each image in a dataset as its class. (Dosovitskiy et al., 2014) proposed to treat each instance as a class represented by a feature vector (in a parametric form). The use of a memory bank to store the instance class representation vector was proposed by (Wu et al., 2018), which was later adopted by MoCo (He et al., 2019) However, SimCLR (Chen et al., 2020) claimed that the memory bank can be entirely replaced with the elements from the same batch if the batch is large enough. SimCLR is a relatively simple method that employs a ResNet encoder network with non-linear projector heads trained using normalized temperature-scaled Cross Entropy Loss and composite data augmentations. The method shows that contrastive learning benefits not only from larger batch sizes but also from deeper and wider networks.

Clustering: DeepCluster (Caron et al., 2018) employs k-means assignments as pseudo-labels to learn visual representations such that the sample and its distorted version would belong to the same cluster. Another method, SeLa(Asano et al., 2020) casts the pseudo-label assignment problem as an instance of the optimal transport problem. SwAV(Caron et al., 2021) is based on a similar formulation such that the objective is to map representations to prototype vectors using soft assignments produced by the Sinkhorn-Knopp algorithm to hold equipartition constraint. This is followed by a swapped-prediction mechanism where the assignment code of a view is predicted from the representation of another view. Compared to contrastive methods, clustering methods such as SwAV are more memory efficient and do not require large batch sizes.

Asymmetric Mechanisms: These method also maximize the similarity between the sample and its distortions but differ in the way they circumvent trivial solutions. Methods like BYOL(Grill et al., 2020) employ asymmetric network architecture where the *online* network has an extra predictor head in comparison to *target* network. Moreover, gradients are propagated only through the online network whereas the target network is updated with a slow-moving average of the online network.

Redundancy Reduction: Barlow Twins involves a joint embedding architecture that takes distorted views of all samples of a batch as inputs. The obtained representations are normalized around the batch dimensions to produce a cross-correlation matrix. The aim is to make the matrix as close to identity such as the identical networks produce similar representations of distorted versions of a given sample. Barlow Twins imposes the redundancy reduction principle through its loss function:

$$\mathcal{L}_{\mathcal{BT}} = \underbrace{\sum_i (1 - \mathcal{C}_{ii})^2}_{\text{invariance term}} + \lambda \underbrace{\sum_i \sum_{j \neq i} \mathcal{C}_{ij}^2}_{\text{redudancy reduction term}}$$

$\lambda (= 5 \times 10^{-3})$ is a positive trading off constant and $\mathcal{C}$ represents the cross correlation matrix with dimension along network's output.

3. Methodology

3.1. Dataset

The dataset consists of 96x96 colored images which can be classified into 800 different classes. It is divided into three parts; a training set of 25,600 images, a validation set of 25,600 images and an unlabeled set with 512,000 images. The test set is kept as a secret and is used to evaluate the performance of the models.

3.2. Backbone

The Backbone used is a custom Resnet architecture, we modified the architecture so that the first Convolution layer has a kernel size of 3x3 and a stride of 1 instead of the original 7x7 (kernel size) and 2 (stride). This was done to cater to the small size of the images in the dataset and ensure a 6x6 version of image reaches the Average Pooling layer.

3.3. Self-Supervised Learning: Barlow Twins

The backbone was followed by three linear layers each with 1024 output units. The optimizer we used was LARS and we trained for 1000 epochs with a batch size of 1024. (You et al., 2017) We experimented multiple data augmentations which included random cropping, resizing to 224x224, horizontal flipping, color jittering, converting to grayscale, Gaussian blurring and solarization. The learning rate starts from 0 and is linearly increased to 0.2 in the first 10 epochs of training, and then decreased to 0.002 using cosine decay schedule (Loshchilov & Hutter, 2017) weight decay of 1e-6.

3.4. Building the Classifier

To build the classifier, we utilized the backbone trained using the Barlow Twins technique and appended two fully connected layers with ReLU between them, so as to project the embedding to a higher dimension before classifying. The classifier was trained using the Stochastic Gradient Descent optimizer with a learning rate of 0.01 for the two FC layers and 0.0008 for the backbone. We used the Cosine Annealing learning rate scheduler to improve performance. To avoid over-fitting we included a dropout layer between the two layers and included weight decay of 1e-5 in the optimizer. In addition, in case of any technical failure we

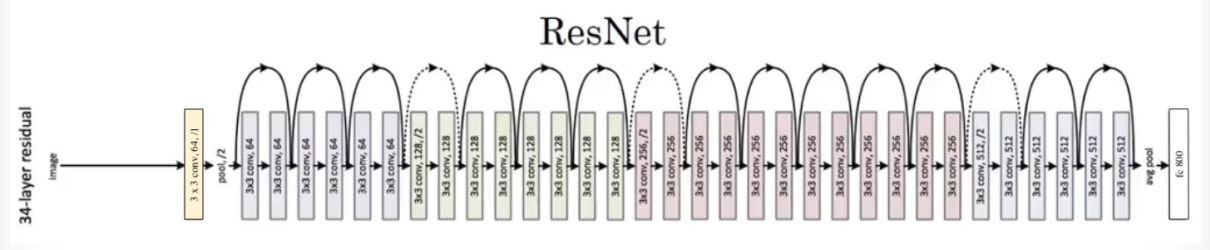

Figure 1. Custom Resnet34

also saved the model after each epoch. Hence, we could still resume as we had the last epoch saved and the model can resume training from that point. Further, in situations when the performance begins to degrade we also saved the best version of the model, i.e, the model with the least loss. We also used Pytorch Lightning that helped us train faster and automate the training loop.

3.5. Fine-tuning: Pseudo-Labelling

To further improve the models performance, we used a pseudo-labelling approach. The classifier was used to predict the labels of images in the unlabeled set. We calculated the softmax of the output of the classifier and stored the index of the maximum value as the prediction and its value as the confidence score. We, then, sorted the images and their labels according to the descending order of confidence scores and skimmed off top 3% of the highest confidence score. These predictions were assumed to be true and subsequently mixed in with the training set. This extra set was used to train the classifier further using the Adam optimizer albeit with a smaller learning rate of 0.0001. This process was repeated for 3 iteration and each time the accuracy stopped improving while training, the learning rate was reduced by a factor of 10.

3.6. Active Learning: Intelligently Selecting a Subset Of Unlabeled Images

To select the 12,800 images out of the 512,000 images from the unlabeled set, we used the difference between confidence score of the top two predicted classes. The lesser this difference the more the model is confused between the top two classes. This results in a better solution as compared to just taking the least confident images as it avoids outliers. Thus, we sorted the images based on the ascending order of this margin and took the top 70,000 images. These 70,000 images were fed into our backbone to obtain their embeddings and then clustered using these embeddings into 800 different classes. We took the top 16 images from each class to ensure an even distribution. We repeated this method with classifiers obtained from different self-supervised learning methods such as Barlow Twins, BYOL and SWaV. For the final result, we took the intersection of images from each of these models.

4. Experiments

In order to decide on which architecture would be best for our dataset, we experimented with various popular architectures like ResNet and VGG. However, the standard version of these models are catered to images of size 224x224, hence we modified these to the dataset we had. In the Resnet architectures, the kernel of the first convolutional layer was changed from 7 to 3 and the stride from 2 to 1. Similarly in VGG-16, we dropped the last CNN layer and included only the last max pooling layer, in addition to the image being resized to 84 x 84. The performance between Resnet and VGG were comparable, and hence decided to move forward with Resnet since it was used by most SSL methods.

Smaller architectures capped at lower accuracy as well as showed the tendency for Catastrophic forgetting while in the pseudo-labeling phase. For example, with a smaller network, each iteration in the pseudo labeling phase showed a drop in the accuracy on the validation set. We could not scale up to Resnet-50 as the GPUs couldn't fit both the the model and the 1024 batch size required for Barlow Twins in memory.

In attempt to achieve a model, with a good performance, a number of self-supervised methods were tried and tested. We started with SimCLR, however, despite training it for a large number of epochs, the performance reached a certain limit and then began to degrade. In order to understand general representations of the image, we used BYOL, which instead of taking negative samples, tries to learn from different augmentations of the same image. However, after training for 80 epochs it reached an upper limit accuracy of 22.4% and had very little improvements after that. Similar results were observed with SWaV, which saturated after 80 epochs to fine-tuning accuracy of 25.7% despite experiment with hyper-parameters such as weight decay and number of prototypes.
Finally, we tried take advantage of the different models we trained, and developed a new network that would ensem-

Figure 2. Visualization of Convolution Filters of Barlow Twins ResNet-34

ble the three models - Barlow Twins, BYOL and SWaV to see if a combination of the three of these would give a higher accuracy. Based on the validation set, we could see that even though Barlow had a significantly higher accuracy, the images it got right were not a superset of the images correctly predicted by other models. We saw an overlap of 3,000 correctly labeled images between the three models however, there were around 1,000 images for both BYOL and SWaV which weren't in common with Barlow. The first approach to ensemble the three models was taking an average of the outputs of each model. This lead to an accuracy of 34% which was lower then Barlow Twins individually. While BYOL and SWaV had 22.5% and 25.7% accuracy respectively, Barlow Twins had an accuracy of around 40%. Hence, we developed a second approach to ensemble the three models, which was a weighted average among the three models. We assigned a higher weight to Barlow Twins and equal weights to BYOL and SWaV. The accuracy for this approach was 39% with a less number of epochs. This method can give a higher accuracy if trained further. Although, this lead to an improvement compared to the previous ensemble approach, this was still lower than the model implemented using Barlow Twins. Hence, we decided to move forward with Barlow Twins and trained it further.

5. Results

We evaluated our model based on accuracy. The performance of the various methods we tried can be seem in Table 1. As we see Barlow Twins with Resnet-34 performed the best among all the other methods. The other models were reached an upper limit in their accuracy. Hence we picked Barlow Twins and trained it for a 1000 epochs and then further trained the model with extra labels generated from pseudo-labeling.

The performance on our validation set is 43.46% and test set is 43.34%. By including the extra labels while training we got a significant increase in accuracy in both validation, 47.66% and test set, 47.63%.

Table 1. Performance of various SSL methods

NETWORK	SSL	ACCURACY
RESNET18	SIMCLR	17.45%
RESNET34	BYOL	19%
RESNET18	BYOL	22.5%
RESNET18	SWAV	25%
RESNET34	ENSEMBLE (AVERAGE)	34%
RESNET34	ENSEMBLE (WT AVERAGE)	39%
RESNET34	BARLOW TWINS	43.46%

References

Asano, Y. M., Rupprecht, C., and Vedaldi, A. Self-labelling via simultaneous clustering and representation learning. In *International Conference on Learning Representations (ICLR)*, 2020.

Caron, M., Bojanowski, P., Joulin, A., and Douze, M. Deep clustering for unsupervised learning of visual features. In *European Conference on Computer Vision*, 2018.

Caron, M., Misra, I., Mairal, J., Goyal, P., Bojanowski, P., and Joulin, A. Unsupervised learning of visual features by contrasting cluster assignments, 2021.

Chen, T., Kornblith, S., Norouzi, M., and Hinton, G. A simple framework for contrastive learning of visual representations. *arXiv preprint arXiv:2002.05709*, 2020.

Doersch, C., Gupta, A., and Efros, A. A. Unsupervised visual representation learning by context prediction. *2015 IEEE International Conference on Computer Vision (ICCV)*, pp. 1422–1430, 2015.

Dosovitskiy, A., Springenberg, J. T., Riedmiller, M., and Brox, T. Discriminative unsupervised feature learning with convolutional neural networks. In *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'14, pp. 766–774, Cambridge, MA, USA, 2014. MIT Press.

Grill, J.-B., Strub, F., Altché, F., Tallec, C., Richemond, P. H., Buchatskaya, E., Doersch, C., Pires, B. A., Guo,

Z. D., Azar, M. G., Piot, B., Kavukcuoglu, K., Munos, R., and Valko, M. Bootstrap your own latent: A new approach to self-supervised learning, 2020.

He, K., Fan, H., Wu, Y., Xie, S., and Girshick, R. Momentum contrast for unsupervised visual representation learning. *arXiv preprint arXiv:1911.05722*, 2019.

Kolesnikov, A., Zhai, X., and Beyer, L. Revisiting self-supervised visual representation learning. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1920–1929, 2019. doi: 10.1109/CVPR.2019.00202.

Langley, P. Crafting papers on machine learning. In Langley, P. (ed.), *Proceedings of the 17th International Conference on Machine Learning (ICML 2000)*, pp. 1207–1216, Stanford, CA, 2000. Morgan Kaufmann.

Loshchilov, I. and Hutter, F. Sgdr: Stochastic gradient descent with warm restarts. *arXiv: Learning*, 2017.

Misra, I. and van der Maaten, L. Self-supervised learning of pretext-invariant representations, 2019.

Noroozi, M. and Favaro, P. Unsupervised learning of visual representations by solving jigsaw puzzles. In *ECCV*, 2016.

Wu, Z., Xiong, Y., Yu, S., and Lin, D. Unsupervised feature learning via non-parametric instance discrimination. *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 3733–3742, 2018.

You, Y., Gitman, I., and Ginsburg, B. Large batch training of convolutional networks, 2017. URL http://arxiv.org/abs/1708.03888. cite arxiv:1708.03888.

Zhang, R., Isola, P., and Efros, A. A. Colorful image colorization. In *ECCV*, 2016.