Momentum Contrastive Learning with Pseudo Labels

Gyanesh Gupta [*1] **Rohith Mukku** [*1] **Xinyue Chen** [*1]

Abstract

Self Supervised Learning (SSL) *is the paradigm of learning representations of unlabeled data, and then fine tune them over downstream tasks such as classification. The importance is paramount, as most of the real world data we have is unlabeled, and only a fraction is labeled.* ***Contrastive Learning*** *is one of the most widely used approaches in Self-Supervised Learning, where we try to teach a model differences between the points using some defined metric. Since the last few years, various techniques have been developed. On the dataset of this competition, We specifically focus on* ***Momentum Contrast (MoCo)****, and add our pseudo label heuristic to it. Additional, we apply some active learning methods to optimally select unlabeled images from a human supervisor. Finally we test our output to given validation and test set.*

1. Introduction

1.1. Self-Supervised Learning

Supervised Learning is the methodology where a machine learning model learns from explicit label given by a human supervisor. The aim of the model is to convert the input, which can be any image or text, into a matrix and using mathematical transformations, arrive at a prediction as close to the ground truth as possible. This is done via using a set training examples, from which it learns patterns and only those patterns which can be generalized into any unseen data. On the other hand, an unsupervised learning method involves learning representations in the absence of human supervision. Without any human intervention, the model finds patterns like similarity, dissimilarity, frequency, etc. A more recent paradigm, self supervised learning, has emerged due to the presence of mixed data, i.e. large amounts of unsupervised data, along with small supervised set of the same distribution. The idea is to learn as much as possible about the features from the unlabeled dataset, and then try to use the model to perform downstream tasks on the smaller supervised set. There may or may not be minor changes in the model while using the supervised data (fine-tuning). SSL initially was applied predominantly in Natural Language tasks as shown by Devlin et al. (2019) in **BERT**, but now have become ubiquitous in Computer Vision. From the various recent works, particularly in the field of computer vision, we see that such techniques can compete in terms of accuracy with their supervised counterparts in various tasks.

1.2. Contrastive Learning

As succinctly stated by Le-Khac et al. (2020), contrastive learning is a methodology whereby models learn representations by comparing different samples. Compared to discriminative models, where 'labels' are used, or in the case of generative models such as autoencoders, as summarized by Bank et al. (2021) where data is reconstructed, in contrastive learning, the notion of how similar or dissimilar the inputs are is considered. A distribution of positive and negative examples ($p^+(.|x)$ and $p^-(.|x)$ is to be learnt from where the required examples can be sampled. The goal can be summarized as pushing similar examples (perhaps of the same class) closer and dissimilar ones apart in the representation space.

1.3. Problem Statement

The above self supervised learning problem statement involved classifying images into 800 given classes. The training data consisted of *512,000* unlabeled images, known to be of these classes, and *25,600* labeled images, with each class having 32 images. We were also given a validation set of *25,600* labeled images. Additionally, there is an unknown test set on which the model will also be tested upon.

Besides this, we were required to submit a request for the labels of *12,800* images out of the total unsupervised image dataset. Our aim is to select those images whose labels are the best ones to have during the supervised training. [1]

[*]Equal contribution [1]New York University. Correspondence to: Gyanesh Gupta <gg2501@nyu.edu>, Rohith Mukku <rm5708@nyu.edu>, Xinyue Chen <xc1305@nyu.edu>.

Proceedings of the 38th International Conference on Machine Learning, PMLR 139, 2021. Copyright 2021 by the author(s).

[1] Our full code is in the following repo `https://github.com/gyaneshg96/DL_finalcode`.

2. Related Work

There have been various methods introduced to solve this task. Some of the recent works are:-

- **Contrastive Based Learning**
  Some of the methods in this category are **SimCLR, MoCo, PIRL**. The general concept behind these methods is that an image is cropped and the correlation or similarity between the cropped images coming from same image is maximised. The similarity between cropped images of different data points is decreased through a loss function. This strategy needs huge amounts of data and huge batch size as it has to establish relation between every data point. Therefore, this takes a lot of time to converge and get good results.

- **Clustering Based Learning**
  DeepCluster, SeLA, SwAV are some of the methods in this category. Images are cropped and based on the embeddings of these cropped images, they are assigned a group our prototype. Generally, soft assignments is practiced and these assignments of cropped images from same data point are then trained to have similar representations. These methods generally converge faster and require lesser time, data as they are non-contrastive methods.

- **Distillation Based Learning**
  Some of the popular methods in this category are BYOL, SimSiam. These methods involve two models, one being the teacher and one being the student. Both these models may or may not share the weights and the learning happens through evaluating teacher model and updating student model.

- **Semi Supervised Based Learning**
  These methods are a bit different from self supervised learning methods where the training or learning happens simultaneously with the supervised labeled data. Some of the popular methods are **FixMatch**, **EnAET**, **MixMatch**, etc.

Current state-of-the-art models perform differently on different datasets. For example, on Cifar dataset, SimCLR and SwAV seem to be performing the best under self-supervised learning. With semi-supervised learning, MixMatch ((Berthelot et al., 2019)) seems to be the best performing. If we consider Imagenet dataset, then SwAV and MoCo seem to be the best performers. For STL dataset, semi supervised techniques like FixMatch ((Sohn et al., 2020)) and EnAET produce the best results as given in . There's an newer version of SimCLR (Chen et al., 2020a) which combines both self-supervised and semi-supervised techniques.

3. Momentum Contrast (MoCo)

3.1. Key Idea

The Momentum Contrast (MoCo) algorithm for representation learning is introduced by He et al. (2020). It has 2 networks: the Query encoder f_{q} and the Key encoder f_{k}. To summarize, there are 3 key ideas in the learning process of MoCo.

Firstly, contrastive learning can be alternatively viewed as a dictionary look-up task, and MoCo maintain its dictionary as a queue. Specifically, the samples in the dictionary are progressively replaced. The current mini-batch is enqueued to the dictionary, and the oldest mini-batch is dequeued. The advantage of this approach is that the encoded keys are less outdated and more consistent with the new ones.

MoCo uses the InfoNCE loss as the unsupervised learning objective:

$$\mathcal{L}_q = -\log \frac{\exp\left(q \cdot k_+ / \tau\right)}{\sum_{i=0}^{K} \exp\left(q \cdot k_i / \tau\right)}$$

which is a sum over 1 positive and K negative samples. Intuitively, this loss trains a $(K+1)$ -way softmax-based classifier that tries to classify q as k_+. The query representation is obtained via $q = f_{\mathrm{q}}\left(x^q\right)$, where the input x^q is a query sample. Likewise, we have $q = f_{\mathrm{k}}\left(x^k\right)$. Then, parameters of the query encoder θ^q are updated by back-propagation through the loss.

However, MoCo choose not to update θ^k in the same way as θ^q. This is because the gradient would propagate to all samples in the queue, which is computationally intractable. Moreover,if we just copy the parameters of query encoder to the key encoder, it still yields poor performance. A possible reason is that rapidly changing encoder reduces the key representations' consistency. To address this issue, MoCo proposes momentum update:

$$\theta_{\mathrm{k}} \leftarrow m\theta_{\mathrm{k}} + (1-m)\theta_{\mathrm{q}}.$$

Here $m \in [0, 1)$ is a momentum coefficient. This approach makes θ_k evolve more smoothly than θ_q. As a result, the keys in the queue are encoded by different encoders but of small differences.

Furthermore, an illustration on how MoCo differs from other contrastive methods is presented in figure 1.

3.2. Version 2

In practice, we used MoCo-V2, which is an enhanced version of MoCo. MoCo-V2 inherits the key idea from above, and further includes the designs of 2 MLP projection heads and stronger data augmentation. These designs are introduced the other unsupervised learning approach SimCLR.

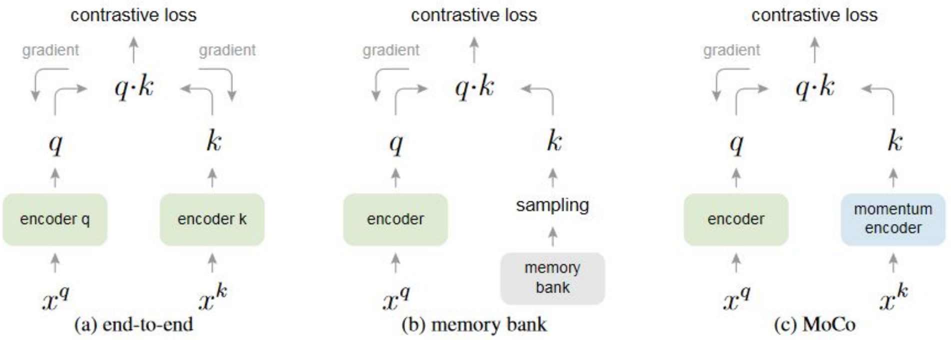

Figure 1. **Conceptual comparison of three contrastive loss mechanisms**. Here we illustrate one pair of query and key. The three mechanisms differ in how the keys are maintained and how the key encoder is updated. (a): The encoders for computing the query and key representations are updated end-to-end by back-propagation (the two encoders can be different). (b): The key representations are sampled from a memory bank. (c): MoCo encodes the new keys on-the-fly by a momentum-updated encoder, and maintains a queue of keys. This figure is adapted from figure 2 in He et al. (2020).

When used with MoCo, they are leading to better image classification and object detection transfer learning results (Chen et al., 2020b).

3.3. Pseudo Labelling heuristic

Pseudo labelling has been explored in a lot of approaches in semi-supervised learning with some success. Theoretically, any model can be improved using extra labels, and these labels are selected from the unsupervised data. Even though our model was mediocre to begin with, by setting a simple threshold we can get a good amount of extra labels.

A caveat that needs to be pointed out is that pseudo labelling should not work if the same model was being used to generate them. This is because, those data points are being assumed to predict correctly (with good confidence), so the model gets positive examples only, and learns nothing new. To combat this, we run pseudo labelling using the SimCLR model, under the assumption that these two models will predict correctly in different distributions. Thus, we are more likely to get a better training set, which will have both positive and negative examples.

4. Active Learning

There are situations in which unlabeled data is abundant but manual labeling is expensive. In such a scenario, learning algorithms can actively query the user/teacher for labels. This type of iterative supervised learning is called active learning. Since the learner chooses the examples, the number of examples to learn a concept can often be much lower than the number required in normal supervised learning. Naturally, we have to develop a strategy that only novel examples are selected, else our learning algorithm will only receive uninformative examples, which do not significantly train it for generalization.

There are two basic approaches of sampling data for labels, **uncertainty based sampling** and **diversity based sampling**. In the former, we select examples based on how uncertain our model(s) is/are while making a prediction on the data. A good candidate for active learning will be one that the model is finding difficult to make a guess, which can be measured based on the probability outputs of the classification task. Diversity based methods ensure that the out the set of data points being supplied are as diverse, possibly having an egalitarian distribution among all classes. This is to ensure that we do not get a restrictive sample, which is not representative of the entire set. Usually hybrid methods which incorporate both approaches are employed.

4.1. Approaches Used

Uncertainty based sampling is easier to implement and work with, which is why we have adopted this approach. We decided to use three criteria for sampling, summarized below. In each of these, we have set a threshold by trial and error, so that we get roughly the same number of examples. Having done that, we take the intersection of the examples so that we get the most uncertain examples, whose label we need. The values in the brackets are the threshold values used.

- **Least confidence** : The maximum probability should be less than the threshold (**0.1**)
- **Margin of confidence** : The difference of the max 2 probabilities should be less than the threshold (**0.01**)

- **Entropy** : The entropy of prediction, i.e. $\sum p \log p$ should be greater than the threshold (**0.74**)

5. Our Experiment

5.1. Unsuccessful methods

On our first attempt, we had tried with a number of approaches, some of which have been mentioned previously. We started with FixMatch and Autoencoders, and both were giving undesired results. For FixMatch, our intuition was that it was among the top methods in the STL dataset by Adam Coates (2011), which has the same image size as our dataset. However, that failed spectacularly due to less number of classes and small batch size.

We observed that the problem posed similarities to the **ImageNet** (Russakovsky et al., 2015) dataset. So, we next tried SimCLR (v1), given that it is one of the most studied approaches and it performs well on ImageNet. Our initial trial was using a backbone of *ResNet18*, and it was giving better results. We decided to proceed with both *ResNet34* and *ResNet50* architectures with batchsize **1024**. We trained for around **300** epochs in total. For every 100th epoch, we further trained the model with a single layer of classifier over the labelled data. Unfortunately, we could not get anything beyond the **25%** mark. Saturation of accuracy was taking place, and the model was grossly overfitting, based on the increase in training accuracy. We deduced that the representations themselves were not learnt properly, by directly observing the misclassified examples. On reason behind this was the restriction of batch-size over just 2 GPUs. The authors of SimCLR have mentioned the dependency of batch size over final accuracies, and our GPU pair was insufficient to deal with a larger batchsize.

5.2. Final Method

We decided to switch to a batchsize agnostic model. MoCo was a natural choice, again for its popularity and results on ImageNet. We decided to first train both versions of MoCo for 20 epochs, and seeing that the MoCo-V2 was showing better performance, decided to go on with the full training. We trained on a batchsize of **512**, and *Stochastic Gradient Descent* with learning rate **0.06** and momentum **0.9**, following the linear scaling mentioned by Goyal et al. (2018). We trained for 300 epochs, again checking every epochs the classification performance. For the final classification, we decided to adopt a learning rate of **30** as mentioned in the paper, and went with a *Cosine LR Scheduler* as used in Loshchilov & Hutter (2017), with a *warm-up* of **10** epochs, as it was showing better results.

6. Results and Observations

6.1. Our Results

Table 1 and 2 shows the results we achieved during the course of this project. FixMatch and Relation Net seemed to take a lot of time to converge. Denoising Autoencoder experiment was an out-of-box trial and the results weren't encouraging as it produced just 2.7% accuracy after 20 epochs and they were consuming a lot of time. We then focused on SimCLR where we got better results and we tried to explore other contrastive based learning methods.

Table 1. Experimented methods

Method Used	Batch Size	Epochs Trained	Acc. (%)
FixMatch	256	20	1.1
Denoise Autoencoder	128	20	2.7
Relation Net	64	20	7.5
SimCLR (ResNet18)	256	20	13.5
SimCLR (ResNet34)	1024	300	22.5
SimCLR (ResNet50)	1024	300	21.7
MoCo-V2	512	300	35.4

Table 2. Accuracy after extra labels

Method Used	Acc. (%)
SimCLR (ResNet34)	23.4
SimCLR (ResNet50)	22.1
MoCo ResNet	36.7

6.2. Our Observations

The following observations are with respect to the MoCo (ResNet) model:

- Figure 2 show some of the images (given with their classes/labels) that the MoCo model failed to classify. As we can see, there are not much distinctive features on which the model could have trained. The images contain generic shapes and colors with no useful information that can separate the object from background content.
- On the other hand in Figure 3, these images are well-defined, well placed in the centre. Therefore, the model seems to do well on these sort of images.

7. Improvements and Future Work

We have recognized a number of areas where we can improve our validation accuracy. Based on the work of our

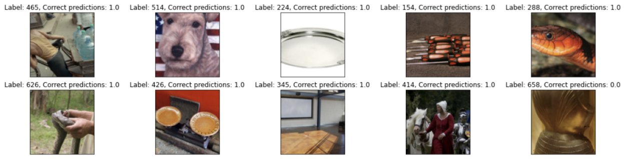
Label: 465, Correct predictions: 1.0
Label: 514, Correct predictions: 1.0
Label: 224, Correct predictions: 1.0
Label: 154, Correct predictions: 1.0
Label: 288, Correct predictions: 1.0
Label: 626, Correct predictions: 1.0
Label: 426, Correct predictions: 1.0
Label: 345, Correct predictions: 1.0
Label: 414, Correct predictions: 1.0
Label: 658, Correct predictions: 0.0

Figure 2. Some of the image classes which the model fails to classify correctly

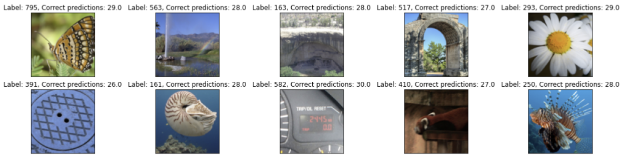
Label: 795, Correct predictions: 29.0
Label: 563, Correct predictions: 28.0
Label: 163, Correct predictions: 28.0
Label: 517, Correct predictions: 27.0
Label: 293, Correct predictions: 29.0
Label: 391, Correct predictions: 26.0
Label: 161, Correct predictions: 28.0
Label: 582, Correct predictions: 30.0
Label: 410, Correct predictions: 27.0
Label: 250, Correct predictions: 28.0

Figure 3. Some of the image classes which the model classifies correctly

peers we could have tried various other SSL models such as SimSiam, Meta-Pseudo Labels Barlow's Twins or VAE based approaches. Additionally, we could have improved our pretraining approach, by using some boosting method. For instance, there were examples whose representations could not learnt effectively, and increasing their weights could remedy this. An example of such method is proposed by Jiang & Zhang (2012). Finally, there are a number of diversity based and hybrid approaches in active learning, such as using clustering algorithms to ensure diversity.

8. Acknowledgements

We would like to acknowledge the instructor of the course CSCI-GA.2572-001 Deep Learning, Prof. Yann LeCun, Prof. Alfredo Canziani, for this extraordinary opportunity to explore Self-Supervised techniques. We also thank Jiachen Zhu and Vlad Sobol for their continuous guidance.

References

Adam Coates, Honglak Lee, A. Y. N. An analysis of single layer networks in unsupervised feature learning, 2011.

Bank, D., Koenigstein, N., and Giryes, R. Autoencoders, 2021.

Berthelot, D., Carlini, N., Goodfellow, I. J., Papernot, N., Oliver, A., and Raffel, C. Mixmatch: A holistic approach to semi-supervised learning. *CoRR*, abs/1905.02249, 2019. URL http://arxiv.org/abs/1905.02249.

Chen, T., Kornblith, S., Swersky, K., Norouzi, M., and Hinton, G. E. Big self-supervised models are strong semi-supervised learners. *CoRR*, abs/2006.10029, 2020a. URL https://arxiv.org/abs/2006.10029.

Chen, X., Fan, H., Girshick, R., and He, K. Improved baselines with momentum contrastive learning. *arXiv preprint arXiv:2003.04297*, 2020b.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. Bert: Pre-training of deep bidirectional transformers for language understanding, 2019.

Goyal, P., Dollár, P., Girshick, R., Noordhuis, P., Wesolowski, L., Kyrola, A., Tulloch, A., Jia, Y., and He, K. Accurate, large minibatch sgd: Training imagenet in 1 hour, 2018.

He, K., Fan, H., Wu, Y., Xie, S., and Girshick, R. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9729–9738, 2020.

Jiang, Z. and Zhang, S. A semi-supervised ensemble learning algorithm. In *2012 IEEE 2nd International Conference on Cloud Computing and Intelligence Systems*, volume 02, pp. 913–918, 2012. doi: 10.1109/CCIS.2012.6664309.

Le-Khac, P. H., Healy, G., and Smeaton, A. F. Contrastive representation learning: A framework and review. *IEEE Access*, 8:193907–193934, 2020. ISSN 2169-3536. doi: 10.1109/access.2020.3031549. URL http://dx.doi.org/10.1109/ACCESS.2020.3031549.

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

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., and Fei-Fei, L. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015. doi: 10.1007/s11263-015-0816-y.

Sohn, K., Berthelot, D., Li, C., Zhang, Z., Carlini, N., Cubuk, E. D., Kurakin, A., Zhang, H., and Raffel, C. Fixmatch: Simplifying semi-supervised learning with consistency and confidence. *CoRR*, abs/2001.07685, 2020. URL https://arxiv.org/abs/2001.07685.