Self-Supervised Learning of Visual representations

Souhaiel BenSalem
ENS Paris Sacaly
souhaiel.ben_salem@ens-paris-saclay.fr

Abstract

Self-supervised learning (SSL) is a sub-field of machine learning in which a model is trained to learn representations of input data without the need for explicit supervision in the form of labeled data. During the past few years, SSL has shown great success in the field of computer vision with the emergence of several powerful contrastive (SimCLR [3] , MoCo [5], CPC [8]) and non-contrastive methods (Auto-encoding, Generative Pre-training, Clustering and VICreg). The goal of this project is In this project, we investigate VICReg, the novel self-supervised learning method presented in [1] by pre-training a visual model on smaller datasets than what is suggested in the original paper, and evaluating the performance of an ImageNet pre-trained model on additional downstream tasks.

1. Introduction

Self-supervised representation learning has seen significant advancements in recent years, with many methods achieving performance comparable to supervised baselines on various downstream tasks. A popular approach among these methods is the use of joint embedding architectures, where two networks are trained to produce similar embeddings for different views of the same image. A well-known example of this architecture is the Siamese network, which uses the same weights for both networks. One of the main challenges with joint embedding architectures is preventing collapse, which occurs when the two branches of the network produce constant or non-informative vectors, ignoring the input. Two main strategies to prevent collapse are contrastive methods and information maximization methods. However contrastive methods can still be the subject of a dimentional collapse where the embeddings only span a lower-dimensional subspace instead of the entire available embedding space [6]
The studied method, VICReg (Variance-Invariance-Covariance Regularization), is an information maximization method that was developed to explicitly avoid the collapse problem by applying two regularization terms to the embeddings: (1) maintaining the variance of each embedding dimension above a threshold, and (2) decorrelating each pair of variables. Moreover, unlike other similar methods, VICReg does not require techniques such as weight sharing, batch normalization, feature-wise normalization, output quantization, stop gradient, memory banks, etc. and performs similarly to state of the art methods on several downstream tasks.

2. VICreg

2.1. The method

VICreg is based on using a loss function that has three components: Invariance, Variance, and Covariance.

- **Invariance** : makes the embedding from different image view closer to each other.

$$s(Z, Z') = \frac{1}{n} \sum \|z_i - z'_i\|_2^2$$

- **Variance**: uses a hinge loss to maintain the standard deviation of each variable of the embedding above a given threshold, this encourages the embedding vectors of samples within a batch to be different.

$$v(Z) = \frac{1}{d} \sum_{j=1}^{d} \max(0, \gamma - \sqrt{Var(x) + \epsilon})$$

- **Covariance**: decorrelates the variables of each embedding and prevents informational collapse in which the variables would vary together or be highly correlated.

$$c(Z) = \frac{1}{d} \sum_{i \neq j} [C(Z)]_{i,j}^2$$

where $C(z)$ is the covariance matrix as defined for the Barlow Twins [10] method.

The overall loss function is a weighted sum of the three terms:

$$l(Z, Z0) = \lambda s(Z, Z_0) + \mu[v(Z) + v(Z_0)] + \nu[c(Z) + c(Z_0)] \quad (1)$$

Where λ, μ, and ν are parameters that determine the relative importance of each term.

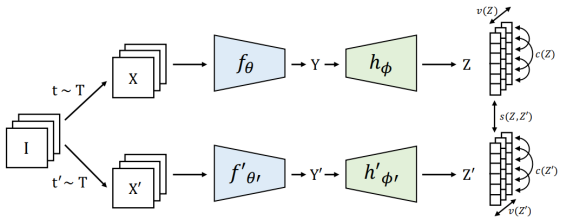

Figure 1. VICreg architecture [1]

2.2. Our approach

For this project, we made some key changes in the model we used compared to that used in the paper. throughout the project, we use ResNet-34 backbones as encoders with output dimension 512 (representation space) and the expander consist of 3 layers that are fully connected and have an output size of 2048 (embedding space).

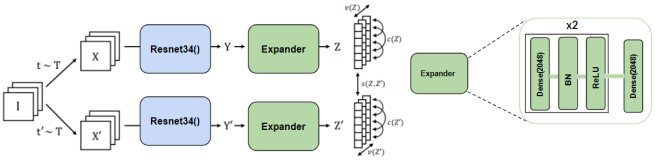

Figure 2. our model

3. Implementation & Experimental Results

3.1. Datasets:

All of experiments i.e pretraining and linear evaluation were conducted on the CIFAR-10 and CIFAR-100 datasets. CIFAR-10 consists of 60,000 32x32x3 images distributed evenly in 10 classes. 50,000 of these images are intended for training and the remaining 10,000 are for testing, whereas CIFAR-100 has 100 classes containing 600 images each: 500 training images and 100 testing images per class.

3.2. Pre-training on CIFAR-10 & CIFAR100

We re-implemented the paper in a simpler way to gain a deeper understanding of the theoretical concepts and to ensure that we fully grasp the practical applications of the method.

3.2.1 Implementation Details

The details for using VICReg for pretraining visual backbones on the 10-class CIFAR and 100-class CIFAR datasets in a self-supervised manner are as follows. We pre-trained our models for 100 epochs using the LARS optimizer as outlined in [9]. of 10^{-5} and a learning rate calculated as $lr = batch\_size/256 * base\_lr$, where the default batch size is 128 and $base\_lr$ is set to 0.2. The learning rate follows a cosine decay schedule, as described in [7], starting at 0 with 10 warmup epochs and ending at 0.002. We also used the same augmentation pipeline described in the paper with the only difference being the change of the RandomResizedCrop method's scale's lower bound form 0.08 to 0.2 since we are dealing with 32x32 images.

3.2.2 Experimental Results

Inspecting VICreg Loss

We visualize the different components of VICreg's loss as well as the total loss.

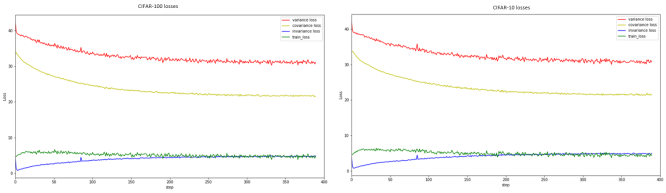

Figure 3. Pretraining losses on CIFAR-100 (left) and CIFAR-10 (right) after 100 epochs

Interestingly, for the choice of $\lambda = \mu = 25$ and $\nu = 1$, we see that in both cases, the Invariance and Covarinace losses converge to the same scale, which can be indicative of a collapse problem.

Pre-evaluation

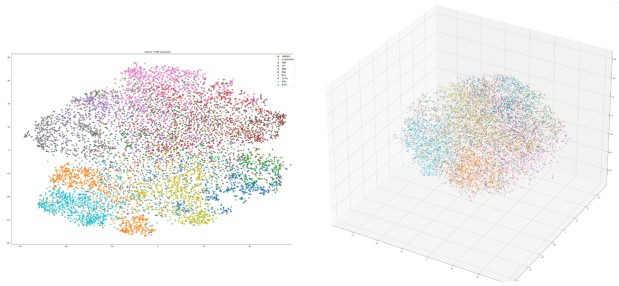

Figure 4. T-SNE projection of the CIFAR-10 test set embeddings in 2D and 3D

To get a sense of our backbones performance before moving to the downstream tasks, we project the output of our model's projector i.e embeddings for the test set of CIFAR-10 and (respectively a subset of CIFAR-100) using the T-SNE method to visualize the ability of our model to separate classes from each other.
We can see that our model is capable of separating the 10 classes for the most part. We also notice that classes with similar visual features such as trucks and automobiles are projected closer to each other.

3.3. Linear Evaluation

We follow the same standard linear evaluation process as described in the paper. We train a linear classifier on top of the frozen representations of our pre-trained ResNet-34 backbones for 100 epochs using SGD as an optimized and a learning rate of 0.01 that follows a cosine decay. We use the exact augmentation pipeline as described in the paper.

3.3.1 Experimental results

The linear evaluation results show that the linear classifier trained on top of our CIFAR-10-pre-trained backbone achieves an accuracy of **76,67%** whereas the classifier trained on top of our CIFAR-100-pre-trained backbone achieves only an accuracy of **51.22%**. The performance diffence is exoected since the first backbone was only tasked with learning the visual representation of 10 classes whereas the other backbone was trained to learn 100 different visual representations.

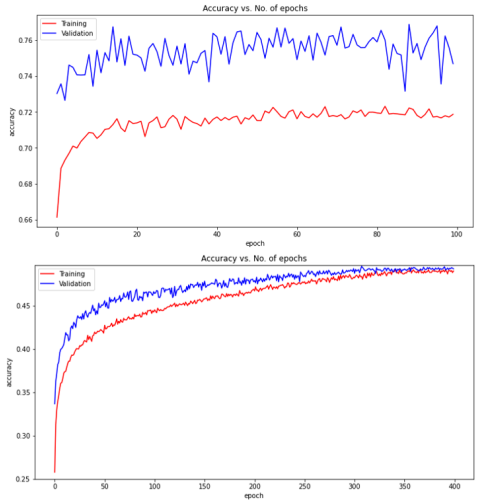

Figure 5. Evaluation of the accuracy for the downstream classification task on CIFAR-10 (top) and CIFAR-100 (bottom).

Before performing our final linear evaluation after 100 epochs of pre-training, we also performed a pre-mature linear evaluation after only 50 epochs. We noticed that more pre-training epochs led to an improvement of the accuracy for the linear classification task by **2%**. We also pre-trained a ResNet-18 backbone to test the effect of the size of the pre-traing batch, since ResNet-18 allows us to use a batch size of 256 a noticed that the overall accuracy for the the linear classification did not improve. However, the training process is more stable.

3.4. Generalization evaluation

To evaluate the generalization capabilities of backbones trained with VICreg, we performed a linear evaluation of the ResNet-50 backbone used in the paper and originally tained on ImageNet-1000, on CIFAR-10 and CIFAR-100.

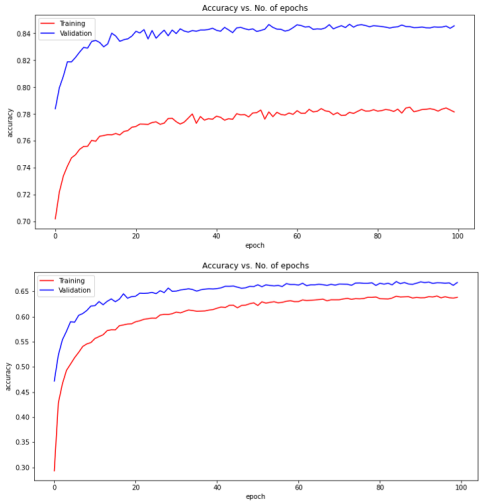

Figure 6. Evaluation of the accuracy of the pre-trained ResNet-50 for the downstream classification task on CIFAR-10 (top) and CIFAR-100 (bottom) after 100 epochs.

We can notice that not only that the model generalizes well on the CIFAR-10 and CIFAR-100 dataset and achieves better accuracies, but also it is more stable during training.
The evaluation of the representations obtained with the ResNet-50 backbone pretrained with VICReg on ImageNet-1000, our ResNet-34 backbone pretrained with VICreg on CIFAR-10 and our ResNet-34 backbone pretrained on CIFAR-100 for the linear classification task is summarized in the following table.

Linear Evaluation of the three backbones

Backone	CIFAR-10 (top-1)	CIFAR-10 (top-1)
ResNet-50 [1]	84.67	66.75
ResNet-34 (Ours)	76.67	-
ResNet-34 (Ours)	-	51.22

4. Conclusion

During this project, we explored VICReg, a powerful self-supervised learning method that introduces a novel objective to learn representations that are invariant to different views, preserve variation in the data, and contain maximum information. Empirically, VicReg performs better than contrastive techniques. Its performance is also on par the other non-contrastive techniques (BYOL [4], SwAV [2]), but it is more interesting and has greater potential due to its simplicity and theoretical transparency.
Our next goal will be to dive deep into the details and try various combinations of hyperparameters and try to compare VICreg to other non-contrastive methods.

References

[1] Adrien Bardes, Jean Ponce, and Yann LeCun. Vicreg: Variance-invariance-covariance regularization for self-supervised learning, 2021. 1, 2, 3

[2] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments, 2020. 4

[3] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations, 2020. 1

[4] Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre H. Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Daniel Guo, Mohammad Gheshlaghi Azar, Bilal Piot, Koray Kavukcuoglu, Rémi Munos, and Michal Valko. Bootstrap your own latent: A new approach to self-supervised learning, 2020. 4

[5] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning, 2019. 1

[6] Li Jing, Pascal Vincent, Yann LeCun, and Yuandong Tian. Understanding dimensional collapse in contrastive self-supervised learning. 2021. 1

[7] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts, 2016. 2

[8] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding, 2018. 1

[9] Yang You, Igor Gitman, and Boris Ginsburg. Large batch training of convolutional networks, 2017. 2

[10] Jure Zbontar, Li Jing, Ishan Misra, Yann LeCun, and Stéphane Deny. Barlow twins: Self-supervised learning via redundancy reduction, 2021. 1