Contrastive Learning on Point Clouds

January 3, 2023

Mirza Mujtaba Hussain

Abstract

In this project, I investigated the use of contrastive learning techniques on non-Euclidean data, specifically point clouds sampled from the ShapeNet10 dataset. I provided an overview of the main ideas and current state of the art contrastive learning techniques, and specifically used the SimCLRv2 technique in my experiments. I implemented this technique while comparing the encoders such as PointNet++ and Dynamic graph CNN (DGCNN), and was able to learn meaningful representations of data . However, the performance on downstream tasks, in this case classification, did not significantly improve the result. I also experimented with various augmentations sets for contrastive learning and evaluated their impact on the downstream task.The experiments conducted suggest that the choice of augmentations, encoder architecture, and encoder size used while training significantly impact the results on downstream tasks,therefore further experiments with different sets of augmentations and larger size of encoders might be able to yield better results.

1. Introduction

Contrastive learning is a self-supervised learning technique that aims to learn meaningful feature representations from unlabeled data. In contrastive learning, a single sample is taken from the training dataset and a transformed version of the sample is created by applying appropriate data augmentation techniques. During training, a contrastive loss is evaluated based on the feature representations of the original and transformed samples extracted from an encoder network. This loss function encourages the feature representations of the original and transformed samples to be similar, while simultaneously pushing dissimilar samples further apart.Some of the popular losses used in the recent works are Contrastive Loss, Triplet, Loss, InfoNCE, PointInfoNCE, NT-Xent loss.

Pretext tasks are tasks that use the inherent structure of the data to provide supervision during training. Pretext tasks can be used as a form of unsupervised pre-training to learn feature representations that can be fine-tuned on downstream tasks. Some examples of pretext tasks in computer vision include Geometric Transformation,image inpainting, colorizing grayscale images, jigsaw puzzles,video-frame prediction. While in NLP, some examples are Center and Neighbor Word Prediction, and Next and Neighbor Sentence Prediction used in Word2Vec and BERT respectively. Pretext tasks have proven to be effective for learning good feature representations but it is important to determine the right kind of pre-text task for a model to perform well with contrastive learning.

There are several approaches to contrastive learning, including end-to-end learning, which involves training two encoders to generate distinct representations of the same sample and using a contrastive loss to push the representations of the original sample and its augmented versions closer together and the representations of negative samples further apart. Another approach is using a memory bank, which involves storing the feature representations of a large number of samples and using these stored representations as negative samples during training. Clustering feature representations is another approach that partitions the feature representations into clusters and uses a contrastive loss function to push the feature representations within the same cluster closer together and the feature representations in different clusters further apart.

2. Related work

Contrastive learning has been a widely studied approach for learning representations from image data. Many methods have been proposed, which include an end-to-end model, SimCLR (Chen et al., 2020a), MoCo(He et al., 2019), and SwAV(Caron et al., 2020), which use memory bank and clustering feature respectively, showing promising results for learning representations that can be used in downstream tasks such as classification and object detection. However,

Email: Mirza Mujtaba Hussain <mirza.1989740@studenti.uniroma1.it>.

Deep Learning and Applied AI 2022, Sapienza University of Rome, 2nd semester a.y. 2021/2022.

the effectiveness of contrastive learning for learning from non-Euclidean data, such as point clouds, is not well explored. Some studies, such as PointContrast(Wang et al., 2020a) have explored the use of contrastive learning for point clouds,which use a unified U-Net architecture built with Minkowski Engine as the backbone network for their model and PointInfoNCE as loss function.

In this project, we examine the use of contrastive learning techniques on point clouds focusing on the SimCLRv2(Chen et al., 2020b) technique and compare using Pointnet++(Qi et al., 2017) and Dynamic graph CNN(DGCNN)(Wang et al., 2018) as encoders for downstream classification task.

3. Dataset

For this project, I used the ModelNet10 dataset which is a subset of the ModelNet40 dataset and contains 4,899 CAD-generated meshes in 10 categories. It is composed of 3,991 shapes for training and 908 shapes for testing. The corresponding point cloud data points are uniformly sampled from the mesh surfaces, and then further preprocessed by normalizing it.

4. Method

The method for this project consists of two main stages, pre-training and fine-tuning discussed below:

4.1. Pre Training

The SimCLR technique is used for pre-training which involves maximizing agreement between differently augmented views of the same data example via a contrastive loss in the latent space.More specifically, given a random mini-batch of point clouds, each sample is augmented twice with a set of augmentations such as rotations, translations,shearing, scaling operations and adding random noise creating two views of the same example.The two augmentations are encoded via an encoder network to generate a pair of representations which are then transformed again with a non-linear transformation network,a MLP projection head, before a contrastive loss (NT-Xent) is applied.Figure 1 demonstrates that samples from the same classes tend to be closer in the lower dimensions obtained after applying T-SNE to the representations obtined after pre-training.

Inspired from the SimCLRv2, I used a deeper encoder and a projection head with higher capacity with more layers.I also experimented with different sets of augmentation where augmentation-1 consists of adding random noise to each point, shearing,rotation and scaling. The augmentation-2 consists of adding random noise to each point,shearing and flipping along an axis.

4.2. Fine Tuning

Fine-tuning is a process that involves adapting a pre-trained model to a specific task by adjusting the model's parameters. In this project, I fine-tuned the pre-trained model from a middle layer of the projection head for the task of classification. In my experiments, I used 1 and 10 percent of overall labelled data available to. fine-tune the model.

5. Network Architecture

I employed small and large model architectures of PointNet++ and DGCNN.The small model of PointNet++ consists of two while as the larger model consists of three set abstraction levels with Ball query method using in grouping layer in both cases.Similarly in DGCNN, I applied Edge Convolution twice in small model and thrice in larger models with deeper layers.

6. Training

During training, I utilized the Adam optimization algorithm to update the weights of our models.The small models were trained without pre-training on all the data for 10 epochs.While as all of the other models were trained for 50 epochs on 10% labelled data and for 100 epochs on 1% labelled data.

7. Results

In table 1, I compare the results calculated on the test data set using the vanilla models with the models which have been pre-trained using two sets of different augmentation mentioned in pre-training section.You can also find the comparison of results when using larger models as compared to smaller models for pre-training in table 2.

8. Conclusions

In conclusion, the study demonstrated the effectiveness of SimCLRv2 technique on point clouds. While comparing the performance of PointNet++ and Dynamic Graph CNN on two sets of different augmentations, I found out using larger models for pre-training improves the results whereas the choice of augmentations and the model architecture also influence the results on downstream task.

Although the results when trained on just 1 percent labelled data are better for pre-trainined models,it was surprising to see the results of the vanilla model trained on just 10 percent of data were better with that of the pre-trainied models.This is a already known phenomenon with synthetic datasets like ShapeNet where pre-training, even in the supervised fashion, hampers downstream task learning as seen in PointContrast.

Table 1. Performance comparison of vanilla models with pre-trained models.

#factors	PointNet++ (Aug-1)	PointNet++ (Aug-2)	DGCNN (Aug-1)	DGCNN (Aug-2)	Vanilla PointNet++	Vanilla DGCNN
All labelled Data	-	-	-	-	**89.4%**	86.4%
10% labelled data	77.1%	**80.3%**	66%	68.5%	83.5%	80.7%
1% labelled data	**53.6%**	49%	37.6%	48%	27.6%	30.5%

Table 2. Comparison between larger and small models

#factors	PointNet++	PointNet++ (small)	DGCNN	DGCNN (small)
10% data	**80.3%**	74.4%	68.5%	61%
1% data	**49%**	44.1%	48%	42%

This project only explored with geometric transformations as pretext task, whereas there are other choices like Jigsaw(Sauder & Sievers, 2019) , OcCo(Wang et al., 2020b), and CM(Jing et al., 2020) which can also impact the final results on downstream tasks .

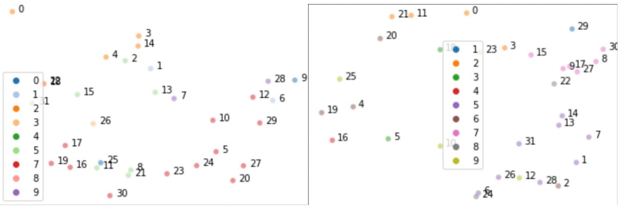

Figure 1. In the figure, the low-dimensional t-SNE embeddings of samples in a random batch are acquired from pre-trained PointNet++(left) and DGCNN(right) are plotted

Code. https://github.com/hussainmujtaba7/Contrastive-Learning-DL2021-22.

References

Caron, M., Misra, I., Mairal, J., Goyal, P., Bojanowski, P., and Joulin, A. Unsupervised learning of visual features by contrasting cluster assignments, 2020.

Chen, T., Kornblith, S., Norouzi, M., and Hinton, G. A simple framework for contrastive learning of visual representations, 2020a.

Chen, T., Kornblith, S., Swersky, K., Norouzi, M., and Hinton, G. Big self-supervised models are strong semi-supervised learners, 2020b.

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

Jing, L., Chen, Y., Zhang, L., He, M., and Tian, Y. Self-supervised feature learning by cross-modality and cross-view correspondences, 2020.

Qi, C. R., Yi, L., Su, H., and Guibas, L. J. Pointnet++: Deep hierarchical feature learning on point sets in a metric space, 2017.

Sauder, J. and Sievers, B. Self-supervised deep learning on point clouds by reconstructing space. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

Wang, A., Su, H., Qi, C. R., and Guibas, L. J. Pointcontrast: Unsupervised pre-training for 3d point cloud understanding, 2020a.

Wang, H., Liu, Q., Yue, X., Lasenby, J., and Kusner, M. J. Unsupervised point cloud pre-training via occlusion completion, 2020b.

Wang, Y., Sun, Y., Liu, Z., Sarma, S. E., Bronstein, M. M., and Solomon, J. M. Dynamic graph cnn for learning on point clouds, 2018.