Implementation of the SimSiam algorithm

Mathias Mellemstuen

October 17, 2022

Introduction

This report will explore the SimSiam [1] algorithm. An implementation of SimSiam will be done and a SimSiam network will be trained. The network will be trained with and without a stop-gradient operation and the results will be compared with the results in [1].

Siamese Neural Networks

Siamese neural networks is popular becuase of it's ability to make predictions based on a small training dataset. A Siamese network contains two or more identical sub-networks, meaning the two instances of the network are sharing parameters and weights. When updating parameters, the update should happen across both networks.

An undesired outcome of Siamese networks are collapsing of outputs. There are many different solutions to prevent collapsing, like the ones used in other methods such as contrastive learning, clustering or BYOL. Simple Siamese Networks are using a simple method for preventing collapsing which works well without utilizing any of the mentioned methods [1].

SimSiam

SimSiam [1] is a new method of Siamese Networks which prevents collapsing of the Siamese network in a simple way. SimSiam is maximizing the similarity of two views in one image, without using one of the other methods mentioned above.

Collapsing solutions do exists in SiamSiam, and therefore a stop-gradient operation is implemented in the algorithm to prevent collapsing.

The SimSiam algorithm was tested with and without the stop gradient operation. It was shown that without the stop graident operation the network would collapse. This would happen just after a few epochs. It did not collapse when running with the stop gradient operation. An empirical study was then done to examine if any other part like the predictor, batch size and batch normalization could add to the contribution of preventing collapse. This emperical study concluded that stop-gradient is the part of the algorithm that prevents collapsing. The other parts affected the accuracy, but did not show any tendency to affect collapsing.

The SimSiam algorithm was compared against other state-of-the-art frameworks. This comparison showed that SimSiam has competitive results, where the accuracy of SimSiam is the highest when doing under 100 epochs of pre-training. Other frameworks showed a higher accuracy when training longer. When comparing SimSiam to SimCLR, SimSiam had better results in all cases.

The paper used the full 1000-class *ImageNet* dataset without labels to train the network. The paper used a series of augmentations like resize and cropping, horizontal flip, color jitter, grayscale and blurring. These augmentations were done with random parameters inside a defined range for each augmentation. Two augmented views of the same image was then created and used further in the algorithm.

Implementation

This implementation will only implement data loading, augmentation of the data and the SimSiam network. Then the SimSiam network will be trained with and without the stop-gradient part, to see if the results in figure 2 in [1] will be replicated.

The dataset used in this implementation is the *Tiny ImageNet (Stanford CS23N)* from *ImageNet*. This dataset was chosen instead of the full dataset from *ImageNet* which was used in [1]. This was done because there was a need to cut down on computation time.

The augmentations performed are:

- Resize and cropping
- Horizontal flip
- Color jitter
- Grayscale

- Blurring

The augmentations were done with the same parameters as in [1]. These augmentations are done twice on each image. This is important for getting two different views of the same image, which will be compared and used in the Siamese neural network. The neural network is created with the same parameters as explained in [1].

Discussion

The figure below shows the training loss when training the model for 50 epochs.

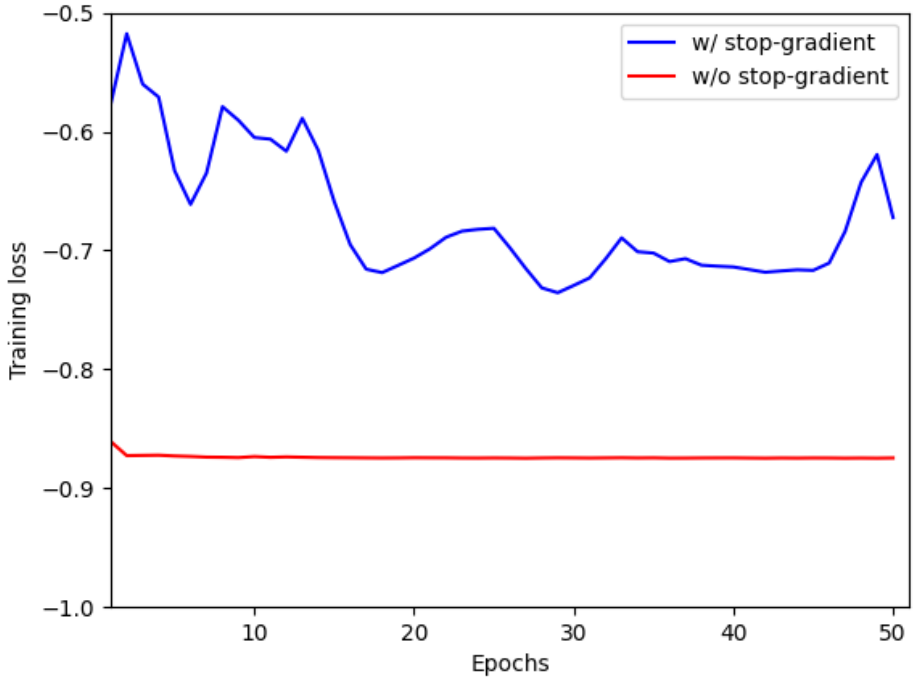

Figure 1: Visualizing training loss over 50 epochs of training, both with and without stop-gradient.

Figure 1 shows that after just one or two epochs, the training loss (w/o stop-gradient) will converge to about -0.87. This is different from the result in [1] where the training loss would converge to -1.0 which is the minimum possible loss value.

The loss with stop-gradient has some slight dissimilarities from the results in [1], but they both seem to be in the same area of around -0.7. One potential reason for the dissimilarities in the results between figure 1 and figure 2 in [1] is that the dataset in this implementation contains less data. The augmentations of the data is also random, meaning there will always be a slight difference in the results because the input data is different.

It can further be seen that the stop gradient operation is helping with stopping the collapsing in this implementation. Without the stop gradient operation, we can see that the training loss is converging to a loss of -0.87 after just a few epochs. This is indicating that the stop-grad is an absolute necessary component of the algorithm to stop collapsing.

Conclusion

After implementing the SimSiam algorithm and training the model for 50 epochs, the results showed some similarities with the results in [1] with some dissimilarities which should be expected. A difference from this implementation and the implementation in [1] was probably expected, since less data was used and the augmentations are random. This implementation indicates that the stop-gradient operation is needed to stop collapsing.

References

[1] Xinlei Chen and Kaiming He. Exploring simple siamese representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15750–15758, 2021.

A Code

The code for this project can be found here.