Final Project Report: Histology images query competition

Cheng-Han Tsai Sheng-Hsun Huang
National Taiwan University
Apartment of Biomedical Engineering
{b07508010, b07508020}@ntu.edu.tw

Abstract

To classify if two biopsy are the same cells, we use Simple Siamese (SimSiam) and SimTriplet method to solve this issue. In this report, we adjust the augmentation, the batch size, and the dataset. The final model we choose is the SimTriplet with the batch size 16 and the whole train dataset. The public score we get the c-index score of 0.82981 on Kaggle, which is higher than strong baseline.

1. Introduction

Nowadays, various diseases influence human's life and some of them need precise diagnosis. One method to get more information from patient is biopsy, which involves extraction of cells and tissue for examination. Usually, the samples getting from biopsy need to be labeled, and pathologists will spend a lot of time in this procedure. Therefore, it's worth to develop tools to accelerate the examination using deep learning.

One task for pathologists is classifying the kind of cells in the target views, and we can consider it as a classification task which we can solve with deep learning. In this project, we have to determine whether the kind of cells in two different images is the same or not. Almost all of the dataset is unlabeled. Due to the lack of label, this project should conduct with unsupervised learning methods, and we choose Simsiam and SimTriplet as our models. Both of them are self-supervised learning methods. In these two methods, the input of the models are the different augmentations of an image in our dataset, and the output is the feature of this image. We can distinguish if two images are the same by computing the cosine similarity of their features.

2. Method

In this section, we'll introduce the components involved in our project.

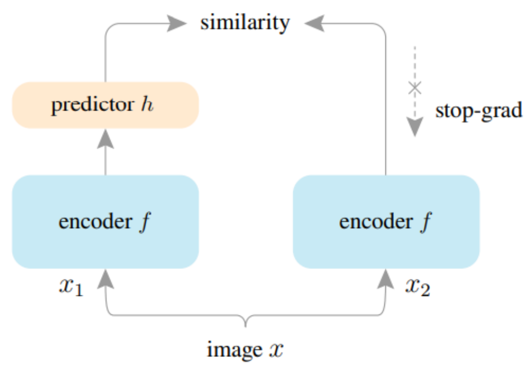

Figure 1. Simsiam Architecture

2.1. Data Processing

We use the Pytorch function for data augmentation. We use $RandomResizedCrop$, $RandomHorizontalFlip$, $RandomApply$, $RandomGrayscale$, $GaussianBlur$ and $Normalize$ in the following experiment. For the detailed information, please see the code on GitHub directly.

2.2. Simple Siamese Learning

Most of the images in this final project are unlabeled and our model should be able to proceed compare task. Therefore, we use Simple Siamese Learning(SimSiam) as our model which is unsupervised and based on Siamese networks, a general models for comparing entities. [1]

This architecture (Figure1) is proposed by Facebook AI Research in 2020. It transforms one image x to two augmented views x1 and x2 as its input. Next, they will be feed to the encoders respectively at the same time. It's worth noticing that these encoders network are the same. One branch of this architecture will cascade prediction MLP head. Denoting the two output vectors as $p_1 = h(f(x_1))$ and $z_2 = f(x_2)$. Loss function in this model is the symmetrized loss of negative cosine similarity D with stop-gradient as:

$$\pounds = \frac{1}{2}D(p_1, stopgrad(z_2)) + \frac{1}{2}D(p_2, stopgrad(z_1))$$

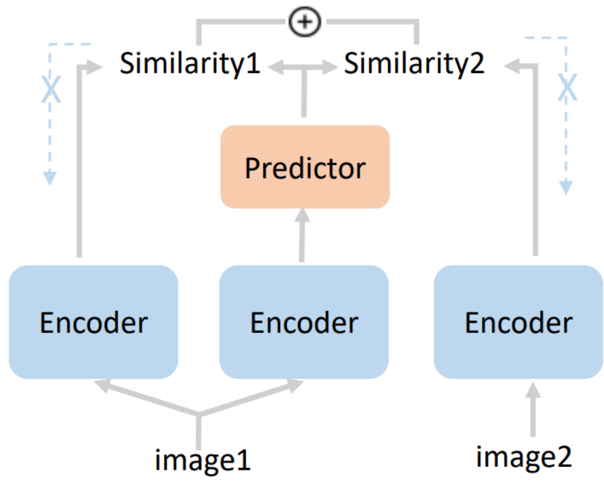

Figure 2. SimTriplet Architecture

2.3. SimTriplet

The SimTriplet [2] model is similar to SimSiam. In the original paper, the SimTriplet also uses one encoder and an MLP layer for the predictor. The differences between them are that the SimTriplet adds another image to caculate the intraloss. i.e., there are three images prepared for the input of the model, m1, m2, and m3. m1, m2 is the random augmentations of the image1, and m3 is the augmentation of image2 (Figure 2). The following is the loss function of the model, y_1, y_2 and y_3 are encoded from augmented views by encoder. z_1, z_2, and z_3 are the representation processed by the predictor. $C(x, y)$ is the cosine similarity between x and y.

$$\pounds_{Intrasample} = \frac{1}{2}C(y_1, z_2) + \frac{1}{2}C(y_2, z_1) \tag{1}$$

$$\pounds_{Intersample} = \frac{1}{2}C(y_2, z_3) + \frac{1}{2}C(y_3, z_2) \tag{2}$$

$$\pounds_{Total} = \pounds_{Intrasample} + \pounds_{Intersample} \tag{3}$$

In our experiment, we do not use two images to train the model. Instead, we use only one image and augment it to three images to train the model. Therefore the loss function is like:

$$\pounds_{Total} = \pounds_{Intersample} + \pounds_{Intersample} \tag{4}$$

2.4. Threshold

The output of the model is the feature of the image. To distinguish the class of two images in the testing phase, we need to set a threshold of the cosine similarity value. First, we get a random value in (0,1) or (0,0.5) interval. Then, using the ground truth data provided by TA, we test every threshold we get and choose the threshold with the best c-index score.

Guassianblur

- Batch size = 32

Method	ResNet34+Valid	ResNet34+Valid no Gaussian blur
C-index	0.722	0.694

Dataset

- Batch size = 32

Method	ResNet34+Valid	ResNet34	ResNet34+Alldataset
C-index	0.722	0.767	0.757

Figure 3. SimTriplet: Guassianblur and Dataset

Batch size

- Method: ResNet34

Batch size	8	16	32
C-index	0.789	***0.829***	0.767

Using ground truth for validation

- Batch size = 32

Method	ResNet34	ResNet34+Alldataset
C-index	0.789	0.761

Figure 4. SimTriplet: Batch size and Using ground truth for validation

3. Results

We experiment different batch sizes, models and the number of the training to test the effect of these variables. All of the results are scored on Kaggle competition. The score used on Kaggle is the c-index score.

3.1. Gaussian Blur

In the SimSiam paper [1], they train the model in the CIFAR-10 dataset without blur augmentation. Therefore, we have an experiment to know if the guassian blur is beneficial for our model. As a result, without another variables, the model with Gaussian blur get the better score than without one. (Figure 3)

3.2. Batch Size

We use various batch size in SimTriplet. In Figure4, it shows that model with batch size 8 gets 0.789 C-index, and with batch size 16 gets 0.829 as our best score in this project, and with batch size 32 gets 0.767.

3.3. Comparison

First, we compare SimSiam and SimTriplet in different batch size. The result of SimSiam shows the decreasing C-index when the batch size is increased. Next, we add ground truth for validation and get the result in Figure6. SimTriplet with batch size 16 achieves 0.77 C-index the same as Sim-

Batch size

- Method: ResNet34
- Dataset: All

Batch Size	SimTriplet	Simsiam
8	0.794	0.794
16	0.796	0.786
32	0.757	0.766

Figure 5. Comparison of SimSiam and SimTriplet in different Batch size

Using ground truth for validation

- Dataset: All
- Batch size = 32

Batch Size	SimTriplet	Simsiam
16	0.77	0.77
32	0.761	0.792

Figure 6. Comparison of SimSiam and SimTriplet with ground truth for validation

Siam. SimTriplet with batch size 32 gets 0.761 and SimSiam gets 0.792 respectively.

4. Discussion

In the discussion part, we want to propose 3 possible factors affecting the performance of the model.

4.1. Comparison of SimSiam and SimTriplet

In our report, it seems that there's no significant difference between the performance of SimSiam and SimTriplet. We think maybe the problem is we use the same image for training while the original method used two images.

4.2. Dataset

We totally try three different dataset to training the model: 1) all of data, which contain the train dataset and the test dataset 2) the whole train dataset 3) 80% dataset and 20% for validation. The best c-index score is the model we use the whole train dataset only. The reason is uncertain. More experiments are required to clarify this issue.

4.3. The overfitting problem

The overfitting problem on the self-supervised learing method is really hard to define and solve. In the supervised learning, we use the validation phase to prevent the overfitting. But in self-supervised learning or other method of unsupervised learning, we have no labels to check. It seems that there are many methods to validate the model. However, we cannot implement them in the limited time.

5. Conclusion

In this project, we find it's possible to use unsupervised learning to conduct competition task. It shows that deep learning has the potential to improve the examination of biopsy. We can expect one day this tool will be powerful and increase the quality of diagnosis.

References

[1] X. Chen and K. He. Exploring simple siamese representation learning. *arXiv preprint arXiv:2011.10566*, 2020. 1, 2

[2] Q. Liu, P. C. Louis, Y. Lu, A. Jha, M. Zhao, R. Deng, T. Yao, J. T. Roland, H. Yang, S. Zhao, L. E. Wheless, and Y. Huo. Simtriplet: Simple triplet representation learning with a single gpu, 2021. 2