The 33rd Annual Conference of the Japanese Society for Artificial Intelligence, 2019

One-shot Learning using Triplet Network with kNN classifier

Mu ZHOU[*1*2] Yusuke TANIMURA[*2*1] Hidemoto NAKADA[*2*1]

[*1]筑波大学
University of Tsukuba

[*2]産業技術総合研究所 人工知能研究センター
Artifical Intelligence Research Center, National Institute of Advanced Institute of Technology

We propose a triplet network with a kNN classifier for the problem of one-shot learning, in which we predict the query images by given single example of each class. Our triplet network learns a mapping from sample images to the Euclidean space. Then we apply kNN classifier on the embeddings generated by the triplet network to classify the query sample. Our method can improve the performance of one-shot classification with data augmentation by processing the images. Our experiments on different datasets which are based on MNIST dataset demonstrate that our approach provides a effective way for one-shot learning problems.

1. Introduction

Deep learning has shown great achievement in various tasks related to artificial intelligence such as object recognition [Girshick 15], image classification [Kaiming 15], and speech recognition [Yu 14]. However, huge amounts of labelled data is necessary for these deep neural network models to train on. In contrast, humans are capable of one-shot learning, which is to learn a concept from one or only a few training example, contrary to the normal practice of using a large amount of data. This is evident in the case of learning a new thing rapidly - humans have no problem recognizing the new category with one or a few direct observation. However, it is a challenging task for machine to solve the classification and recognition problem with very few labelled training data.

2. Related work

Several studies have investigated few-shot learning and one-shot learning, one special type neural network is Siamese Neworks [Koch 15]. The idea of the Siamese Network is based on distance metric learning which is to learn the distance metric from the input space of training data by a contrastive loss, then keep the samples belonging the same class close to each other and separate the dissimilar samples. The similar one is Triplet Network [Hoffer 15] which is composed of 3 parameter-shared convolutional neural networks.

Inspired by Siamese Networks and Triplet Networks, we improve the Triplet Network and use a triplet loss [Schroff 15] in our work. The loss function is to minimize the distance between the data with same label and maximize the distance between the data with different label. Before we get the embeddings trained on networks, we do data augmentation on the training set with only one sample. Then we make the prediction to the embedded query points by finding the nearest embedded support point by using k-Nearest Neighbor classifier. The procedure of the whole work is shown in Figure 1.

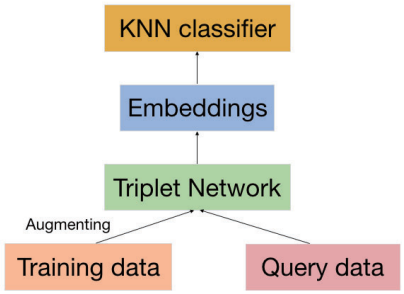

Figure 1: Prediction procedure.

3. Method

3.1 Triplet Network

In this research, we use the triplet network to learn the distance metric from inputs of triplet images. The triplet network is a horizontal concatenation triplet with 3 identical Convolutional Neural Networks (with shared parameters), these ConvNets are trained using triplets of inputs. The input triplet $(\vec{x_a}, \vec{x_p}, \vec{x_n})$ is composed of an anchor instance $\vec{x_a}$, a positive instance $\vec{x_p}$ (same class as the anchor), and a negative instance $\vec{x_n}$ (different class from the anchor). The network is then trained to learn an embedding function *f(x)* called triplet loss. The model architecture is shown in Figure 2.

3.1.1 Convolutional Networks

A series of breakthroughs in image classification came with the introduction of Convolutional Neural Networks (CNNs or ConvNets), where the image is input into a nested series of functions and convolved with filters, then output as feature vector. In our method, the ConvNet has 4 convolutional layers and is used as an embedding function. The ouput is passed through a fully connected layer resulting in a 128-dimensional embedding. In addition, we use ReLU as an activation function which is a common choice, especially for convolutional networks. The architecture of this ConvNet is as following:

- 1x{5x5-conv.layer (32 filters), 5x5-conv.layer (32 filters), batch normalization, max pool(2, 2), leaky relu,

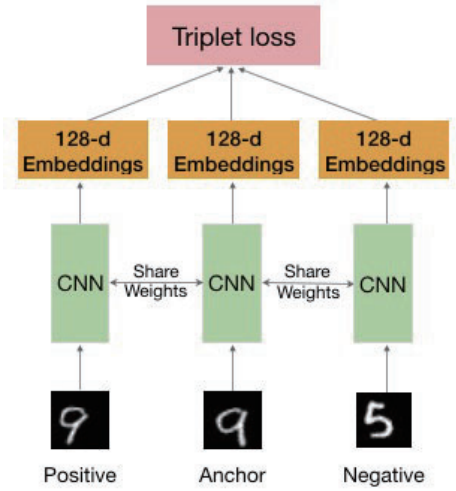

Figure 2: Triplet Network Model.

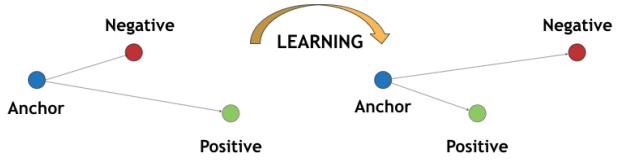

Figure 3: Triplet Loss Function.

dropout(0.25)},

- 1x{3x3-conv.layer (64 filters), 3x3-conv.layer (32 filters), batch normalization, max pool(2, 2), leaky relu, dropout(0.25)},
- 1x{fc-layer, batch normalization}.

3.1.2 Triplet Loss

Although we did not compare it to other loss function, we believe that the triplet loss is more suitable for this network, and triplet loss layer could improve the accuracy of ConvNets. A triplet loss is used to learn an embedding space for the images, such that embeddings of same class are close to each other, while those of different class are far away from each other. For the distance on the embedding space d, the loss of a triplet $(\vec{x_a}, \vec{x_p}, \vec{x_n})$ is:

$$L = max(d(x_a, x_p) - d(x_a, x_n) + \alpha, 0)$$

where α is a margin that is enforced between positive and negative pairs[Schroff 15]. In our research, the triplet loss minimizes the distance between the anchor and the positive, both of which have the same identity, and maximizes the distance between the anchor and a negative of a different identity, as shown in Figure 3.

3.2 kNN Classifier

The k-Nearest Neighbors algorithm is one of the simplest way to perform classification. Most kNN classifiers use Euclidean distances (also known as L2-norm distance) to measure the similarities between the instances which are represented as vector inputs. The L2-norm distance is as following:

$$d(\vec{x}, \vec{y}) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2}$$

In our research, after we trained the dataset (both train and test dataset) on the Triplet Network, we obtained the embeddings of the data, each of which is a 128-dimensional feature vector. Then we used PCA (Principal component analysis) to reduce the dimension of the feature vectors. Since these vector embeddings are represented in shared vector space, we can calculate the similarity between the vectors by using the vector distance. Finally we used kNN classifier to calculate the distance between the test point and all the training points by giving the feature vector of labelled training and unlabelled test data. We gained the best choice of k and choose the corresponding classification that appears most frequently as the predictive class.

3.3 Data Augmentation

Data augmentation is the most common solution for one-shot learning, since it can help to increase the amount of relevant data in the dataset and boost the performance of neural networks. In our research, we augmented the images in the training dataset. As a result, a large amount of training images was created, through different ways of processing or combination of multiple processing, such as random rotation, shifts and shear, etc.

4. Experiment

4.1 Dataset

MNIST database (Modified National Institute of Standards and Technology database) is a large database of handwritten digits that is commonly used for training various image processing systems. The MNIST database contains 60,000 images for training and 10,000 images for testing. Figure 4 presents some of the digits from MNIST dataset.

4.1.1 Initial Dataset

To setup the training dataset, we chose whole digit images with label 0 to 4, while we randomly selected simple digit image with the label 5 to 9 from the MNIST dataset. This initial dataset was used for our comparison experiment. The count of each label on initial training dataset is shown in Figure 5.

4.1.2 Augmented Dataset

In addition to the initial dataset, we generated another training dataset by the technique of data augmentation. In our experiment, we augmented the single image. Due to the limitation of some digit images, (i.e. digit 9 may be recognized as digit 6 after the 180-degree rotation,) we did the random rotation operation with only 30 degrees combined with random zoom and random shifts. To ensure similar appearance of the amount of each label, we enlarged the images several times with similar amount. The count of

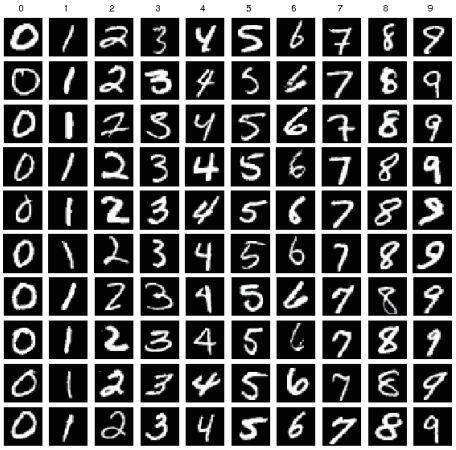

Figure 4: Samples from MNIST dataset.

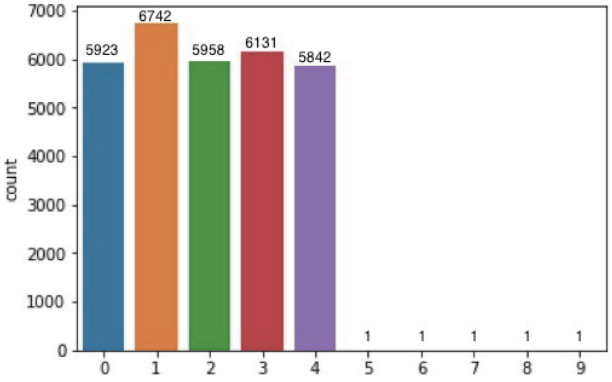

Figure 5: The initial dataset.

each label on augmented training dataset is shown in Figure 6.

4.2 Triplet Selection

Input triplets for Triplet Network were generated in two ways. One kind of triplets was produced by the augmented dataset, while another one was created by the initial dataset which was not augmented. For the first type, we randomly selected 1 sample (used as the anchor instance) from the dataset, then chose another one (used as the positive instance) from the same label. Then we randomly obtained the other sample (used as the negative instance) from any other label. Finally, we concatenated them as a triplet pair. However, for the other type created by initial dataset, we used the same image as the positive instance to overcome the limitation of lack of samples.

4.3 Results

We evaluated the performance of our model on above two datasets - initial dataset and augmented dataset, in order to judge the effectivity of data augmentation. To estimate the performance on Triplet Network in comparison to other model, we applied the CNN model on one-shot classification with the augmented dataset, as is mentioned above.

We obtained the embeddings of training points and test

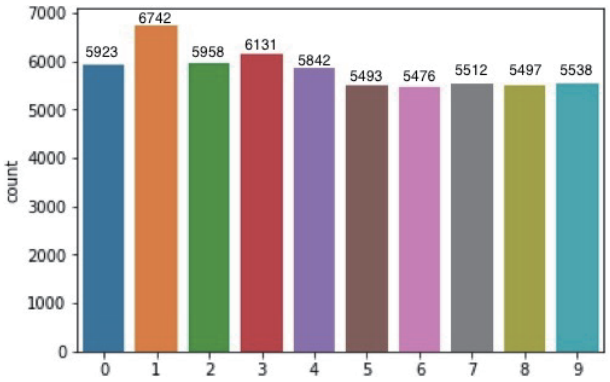

Figure 6: The augmented dataset.

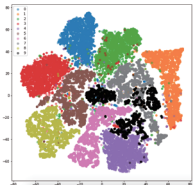

Figure 7: Embedding visualization of training points.

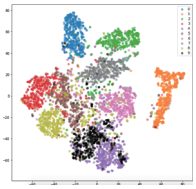

Figure 8: Embedding visualization of test points.

points from the Triplet Network, and we did visualization using t-SNE technique, as shown in Figure 7 and Figure 8. With these training points and test points, we evaluated the accuracy with different k ranging from 1 to 30, and selected the best choice of k. Figure 9 presents the accuracy of kNN classifier for different choice of k with augmented dataset in our Triplet Network model, and we get the best k (k=11) in this experiment. We predicted the label of test points with best k, and compared with the true label. The results are shown in Table 1, which present the accuracy of the test dataset with 1-shot classes (label 5 to label 9).

In our experiment on Triplet Network, the accuracy of the test dataset is 46.8% for 1-shot classes, while the accuracy

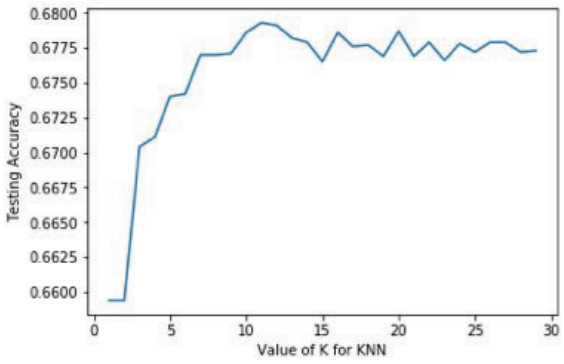

Figure 9: Accuracy of kNN classifer for different choices of k.

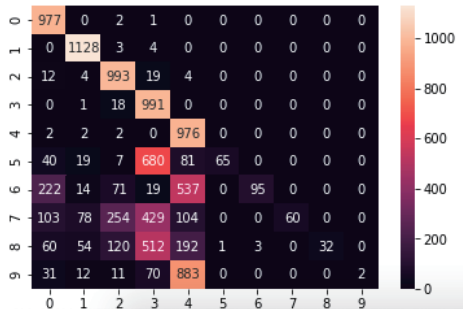

Figure 10: The result on TripletNN with initial dataset.

Figure 11: The result on TripletNN with augmented dataset.

is only 9.8% in the comparison experiment. This result suggests that data augmentation make sense and can obtain better prediction result than without data augmentation. In addition, the Triplet Network gives a better performance than the CNN model in this experiment.

Figure 10 and 11 show the results between actual labels and predicted labels in both datasets using Triplet Network. With regard to the accuracy of digit 9, it gets a low score since most of digit 9 are recognized as digit 4. This result implies that most written digit 9 are significantly similar to written digit 4, and the machine may not recognize them precisely with simple sample.

Method (dataset)	Accuracy					
	5	6	7	8	9	Average
TripletNN (not Agumented)	14%	18%	11%	6%	0%	9.8%
CNN (Augmented)	25%	26%	16%	24%	13%	20.8%
TripletNN (Augmented)	**42%**	**56%**	**66%**	**56%**	**14%**	**46.8%**

Table 1: Results of 1-shot classes.

5. Conclusion

In this work, we described how a Triplet Network model, inspired by the Siamese Network based on distance metric, can be used for one-shot learning. We used the embeddings of training points trained on kNN classifier and predict the label with the embedding of testing points by the classifier. We obtain significant improvement by the effectiveness of data augmentation. Of the 3 approaches tested, we achieved best results by augmenting the initial dataset with Triplet Network model. While in the contrast experiment on CNN model, data augmentation resulted accuracy of 20.8%. However, the experiment on Triplet model with initial dataset resulted accuracy of 9.8%, where almost all the data trained with 1 sample can not be recognized. This study therefore indicates that the benefits gained from data augmentation may work well on one-shot learning problem.

Although our experiment demonstrate a great improvement, the results are subject to certain limitations. For instance, since the differences between digit 9 and digit 4 are unable to be separated, most of digit 9 are recognized as digit 4 in the experiments. In addition, due to the computational constraint, our experiments were unable to explore how our approaches work on other much larger and complex datasets. Therefore, future work should focus on how to distinguish the difference between written digit 9 and digit 4 and how to enlarge the metric distance between digit 9 and 4. Furthermore, future studies need to be carried out in order to validate whether our approach does indeed help to solve the one-shot learning on other large and complex datasets, such as Fashion MNIST, Omniglot, Mini-Imagenet and e.t.

Acknowledgement

This paper is based on results obtained from a project commissioned by the New Energy and Industrial Technology Development Organization (NEDO). This work was supported by JSPS KAKENHI Grant Number JP16K00116.

References

[Girshick 15] Girshick, R.: Fast R-CNN, *IEEE International Conference on Computer Vision (ICCV) 2015* (2015)

[Hoffer 15] Hoffer, E. and Ailon, N.: Deep metric learning using triplet network, *International Workshop on Similarity-Based Pattern Recognition* (2015)

[Kaiming 15] Kaiming, H., Xiangyu, Z., Shaoqing, R., and Jian., S.: Delving deep into rectifiers: Surpassing human- level performance on imagenet classification, *arXiv preprint arXiv:1502.01852* (2015)

[Koch 15] Koch, G., Zemel, R., and Salakhutdinov, R.: Siamese neural networks for one-shot image recognition, *ICML Deep Learning Workshop* (2015)

[Schroff 15] Schroff, F., Kalenichenko, D., and Philbin., J.: Facenet: A unified embedding for face recognition and clustering, *Proceedings of the IEEE conference on computer vision and pattern recognition* (2015)

[Yu 14] Yu, D. and Deng, L.: *Automatic Speech Recognition: A Deep Learning Approach*, Springer Publishing Company (2014)