Experiments on Metric Learning Methods

Shaoheng Liang and Yerong Li

Introduction

Metric learning is the task of learning a distance function on pair of objects. We experiment on three metric learning methods based on deep learning, including Siamese Network [1], Triplet Network [2] and n-Tuple Network [3]. The three approaches can be viewed as frameworks that contain a set of networks sharing parameters and a loss function integrating the outputs of the networks. The basic idea is to force the network to give shorter distance between similar instances and longer distance between dissimilar instances.

We wrote a wrapper class for convolutional network to hide the low-level implementation, and focus on the design of the high-level frameworks and loss functions. In our experiment on MNIST dataset and a classic network structure, Siamese Network has a fair performance while Triplet Network produces better embeddings. We are not able to make the original n-Tuple Network work. However, we have tested a few modified versions of it, which keep the framework unchanged but use different loss functions.

Main Objectives

1. Implement a toolkit for experimenting deep metric learning methods.
2. Experiment on Siamese, Triplet and n-Tuple Network and compare the results.
3. Explore possible extensions to these networks.

Siamese Network

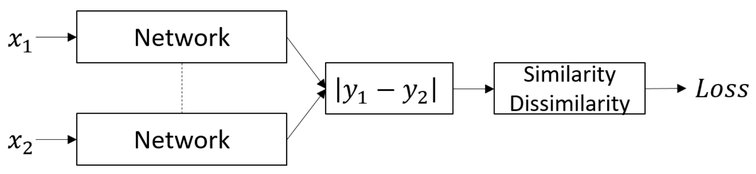

Figure 1: Structure of Siamese Network.

$$Loss = (1 - S)L_S(|y_1 - y_2|) + SL_D(|y_1 - y_2|) \tag{1}$$

$$L_S = \frac{1}{2}|y_1 - y_2|^2 \tag{2}$$

$$L_D = \frac{1}{2}\{max(0, m - |y_1 - y_2|)\}^2 \tag{3}$$

Triplet Network

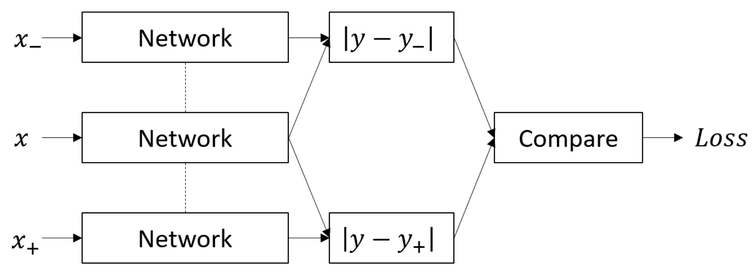

Figure 2: Structure of Triplet Network.

$$Loss = |(d_+, d_- - 1)|^2 = const \bullet d_+^2 \tag{4}$$

$$d_+ = \frac{\exp(|y - y_+|)}{\exp(|y - y_+|) + \exp(|y - y_-|)} \tag{5}$$

$$d_- = \frac{\exp(|y - y_-|)}{\exp(|y - y_+|) + \exp(|y - y_-|)} \tag{6}$$

n-Tuple Network

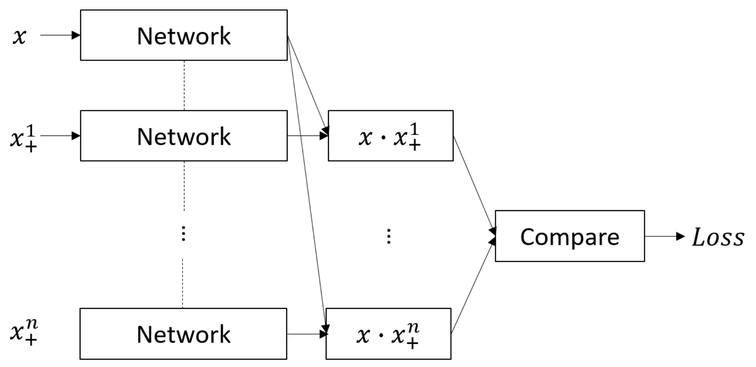

Figure 3: Structure of n-Tuple Network.

$$L(\{x_i, x_i^+\}_{i=1}^N) = \frac{1}{N}\log(\sum_{i=1}^{N}\exp(\sum_{j=1}^{N} y_i^T y_j^+ - y_i^T y_i^+)) \tag{7}$$

References

[1] Raia Hadsell, Sumit Chopra, and Yann LeCun. Dimensionality reduction by learning an invariant mapping. In *Computer vision and pattern recognition, 2006 IEEE computer society conference on*, volume 2, pages 1735–1742. IEEE, 2006.

[2] Elad Hoffer and Nir Ailon. Deep metric learning using triplet network. *CoRR*, abs/1412.6622, 2014.

[3] Kihyuk Sohn. Improved deep metric learning with multi-class n-pair loss objective. In *Advances in Neural Information Processing Systems*, pages 1857–1865, 2016.

Implementations

To facilitate our experiment, we write a wrapper class `Network` for convolutional network, which provides three member functions `add_conv()`, `add_fc()` and `connect()`. The implementation of a Triplet Network is as easy as follows.

```
net = Network(28, 28, 1, 2);
net.add_conv(5, 5, 2, 2, 32);
net.add_conv(5, 5, 2, 2, 64);
net.add_fc(1024);
net.add_fc(256);
mid_output = net.connect(mid_input);
pos_output = net.connect(pos_input);
neg_output = net.connect(neg_input);
```

However, the implementation of loss functions is still on case-by-case basis.

Results

The Siamese Network can separate the classes fairly. Expectable overlappings between digits “9”, “7” and “4” and between digits “2” and “3” is observed. In Triplet Network, these overlappings are further suppressed, which leads to a better separation.

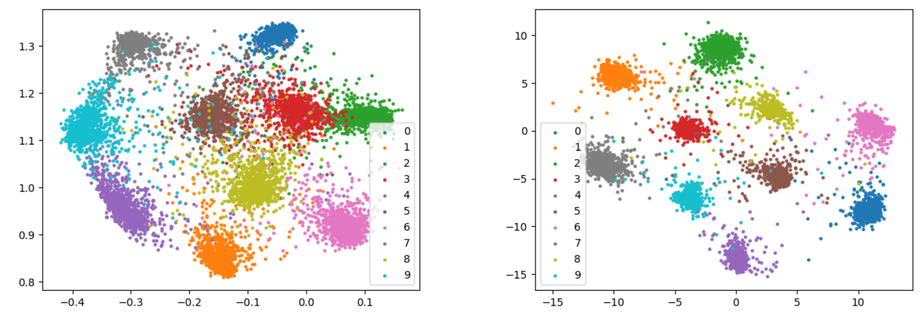

Figure 4: Left: Siamese Network; Right: Triplet Network

The original network is impaired by a trivial all zero solution for dot product. As a result, all points shrinks to a very small area. Modified loss function involves the L2-Norm, which resolves the shrinking problem, but does not lead to a better separation.

$$Loss(\{x_i, x_i^+\}_{i=1}^N) = \frac{1}{N}\log(\sum_{i=1}^{N}\exp(\sum_{j=1}^{N} y_i^T y_j^+ - y_i^T y_i^+)) + \frac{\lambda}{2N}\sum_{i=1}^{N}(\|y_i\|^2 + \|y_i^+\|^2)$$

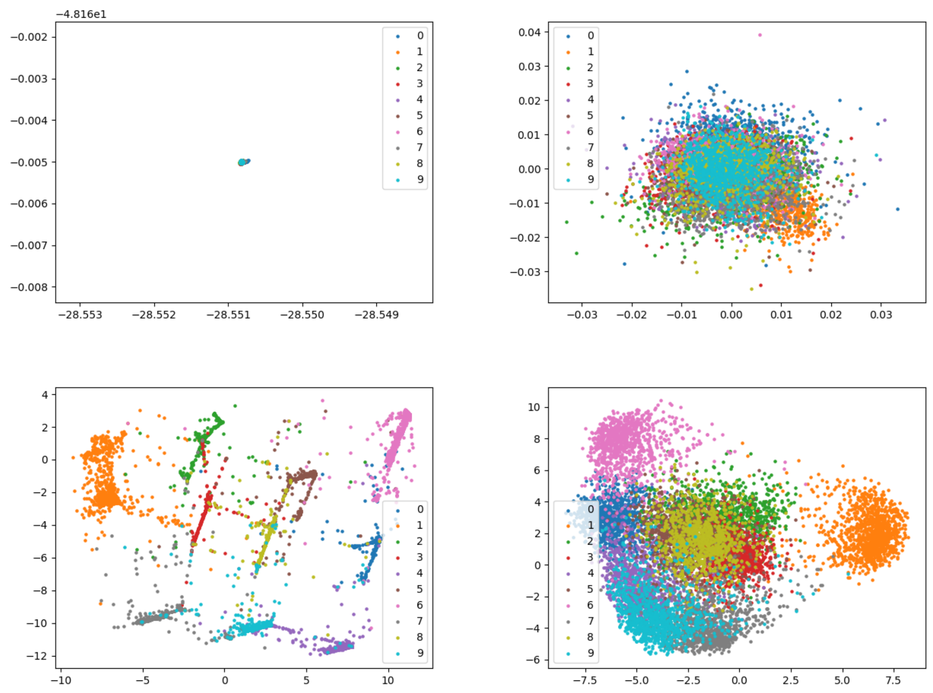

Figure 5: Top left: Original n-Tuple Network; Top right: n-Tuple Network with L2 Norm
Bottom left: n-Tuple Network with Triplet loss; Bottom right: n-Tuple Network with modified Triplet loss

We directly use Triplet loss function on n-Tuple framework, which gives 90 triplets per 20 images, and produces a well separated but badly scattered figure. The reason of the strange scattering may be the high dependency of the triplets. To solve this problem, we designed a new loss function resembles the Triplet Network loss function but combines all 10 classes into one expression. As a result, the scattering is better but the separation is worse.

$$Loss = \frac{\exp(|y - y_+|)}{\exp(|y - y_+|) + \sum_i \exp(|y - y_-^i|)} \tag{8}$$

Conclusions

- Our wrapper class gives a good basis for metric learning experiments.
- The Siamese Network and Triplet Network perform as expected. Triplet Network outperforms Siamese Network by reducing overlapping of digits that looks alike.
- We are not able to reproduce the original n-Tuple Network, but we have designed multiple loss functions as partial fixes.

Future Research

- Show quantitative comparison of the methods, although the difference performance is intuitive.
- Combine the loss functions for n-Tuple to balance the scattering and the separation, or find new loss functions to get a better performance.