MNIST shuffle

Albert Gregus

January 2022

1 Introduction

The digits in the test set of the MNIST dataset got cut in half vertically and shuffled around. The aim of this task is to make an algorithm to pair together the matching tops and bottoms from the shuffled images in MNIST test set.

2 Structure

2.1 Time complexity

The main difficulty of the task is the characteristics of MNIST, where several tops and bottoms could fit together and the time complexity of the pairings. If we naively pair all the tops to all the bottoms, then we have $\mathcal{O}(n^2)$ time complexity, which means if we calculate with 0.5s for an epoch, then we need more than an hour to run on the whole test set. Instead I decided to build an algorithm based on two networks, which transform the tops and the bottoms to a latent space, where I check their similarity (with their dot product), similarly to siamese[1] or contrastive[2] learning. This requires only $\mathcal{O}(2n)$ time complexity.

2.2 Algorithm structure

After running the network on the test set and collecting all the tops and bottoms latent vectors I calculate the distance of all top-bottom pairs and gather the best k tops for all the bottoms, and the best k bottoms for all the

tops. These include most of the right pairings besides hardly seperable hard negatives. Calculating their intersection I get most of the right pairings, but can filter out around half of the negatives. Finally I run a bipartite matching algorithm to get the best pairings.

2.3 Training structure

The training consists of two parts. First I am training on positives (good pairings) and random negatives (wrong pairings). During the second stage I repetitively do the following:

1. train for a few epochs on random positives and negatives to not forget easy examples
2. run the best k gathering on the train set, and collect the hardest negative examples
3. train for a few epochs on positives and hard negatives

3 Results

3.1 Numerical results

The inference on a set of 10000 images with 28x28 resolution (the val or the test set) finishes under 15 seconds, and has 55% accuracy on the val and 54% on the test set.
Note: The method could probably be further improved with ensembling.

3.2 Visual results

It can be seen from the results, that the model could find right pairings, and even for the wrong pairings produce decent numbers.

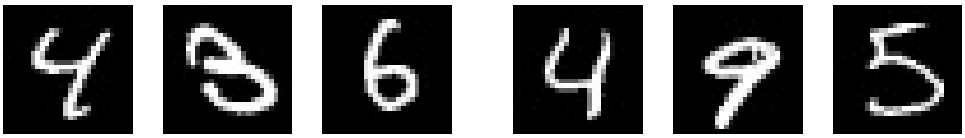

Figure 1: Original images of 3 good and 3 wrong predictions

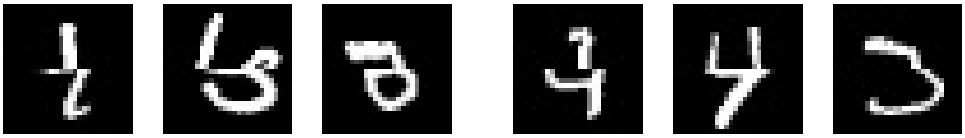

Figure 2: Shuffled inputs of 3 good and 3 wrong predictions

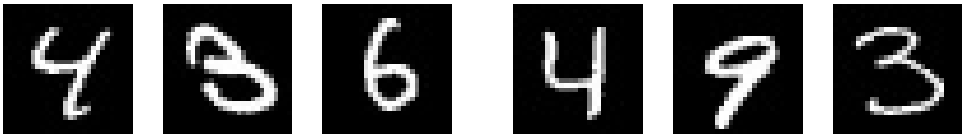

Figure 3: Reconstruction of 3 good and 3 wrong predictions

References

[1] Gregory R. Koch. "Siamese Neural Networks for One-Shot Image Recognition". In: 2015.

[2] S. Chopra, R. Hadsell, and Y. LeCun. "Learning a similarity metric discriminatively, with application to face verification". In: *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*. Vol. 1. 2005, 539–546 vol. 1. DOI: 10.1109/CVPR.2005.202.