Machine Learning Personal Project

Jiang Shuyang
Email: jiangshuyang@sjtu.edu.cn

June 1, 2022

1 Introduction

In this project, I finish the basic task, i.e., image classification on MNIST dataset. Besieds, I also finish image generation with VAE and sentiment analysis with LSTM. I will in detail illustrate the model architectures in each task, final result on test set and corresponding visualization results.

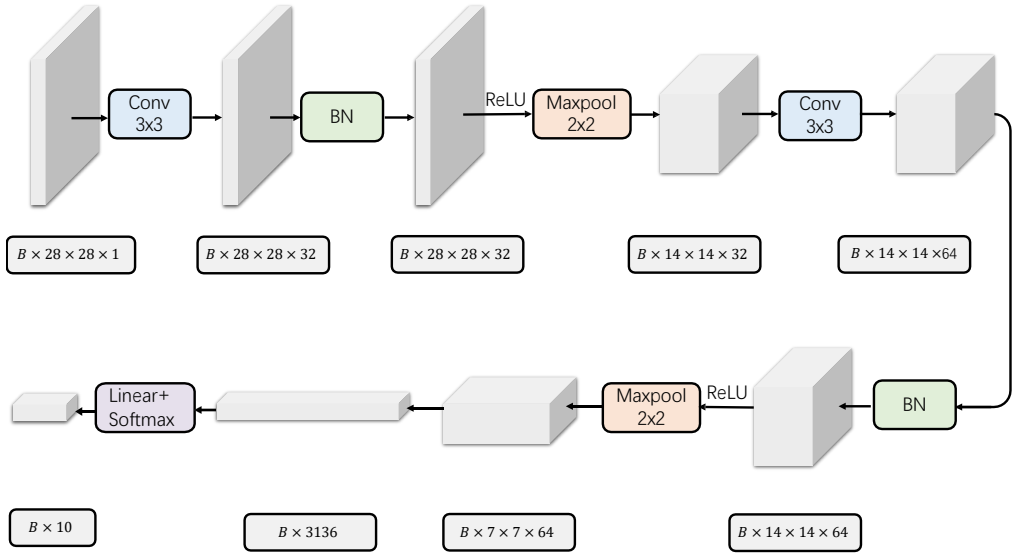

Figure 1: Model diagram of self-designed CNN network for MNIST image classification task. The whole model contains 12 neural layers including the output layer.

2 Image Classification

2.1 Model & Hyperparameters

I design a 2-layer CNN models with an additional linear and softmax layer. The architecture can be interpreted in Fig. 1. I set the batch size to 256 so as to maintain the high capability of batch normalization layer. The learning rate is set to 0.001 and the dropout rate is set to 0.1. I train the whole model for 30 epochs and select the checkpoint with best binary cross entropy loss for evaluation on test set.

2.2 Experiment Results

The selected checkpoint can achieve 99.39% accuracy in test set. The loss and accuracy change figures of self-designed CNN model are depicted in Fig. 2. It can be seen that the change is smooth, which proves that the hyperparameters are well-set and does not require more tuning steps.

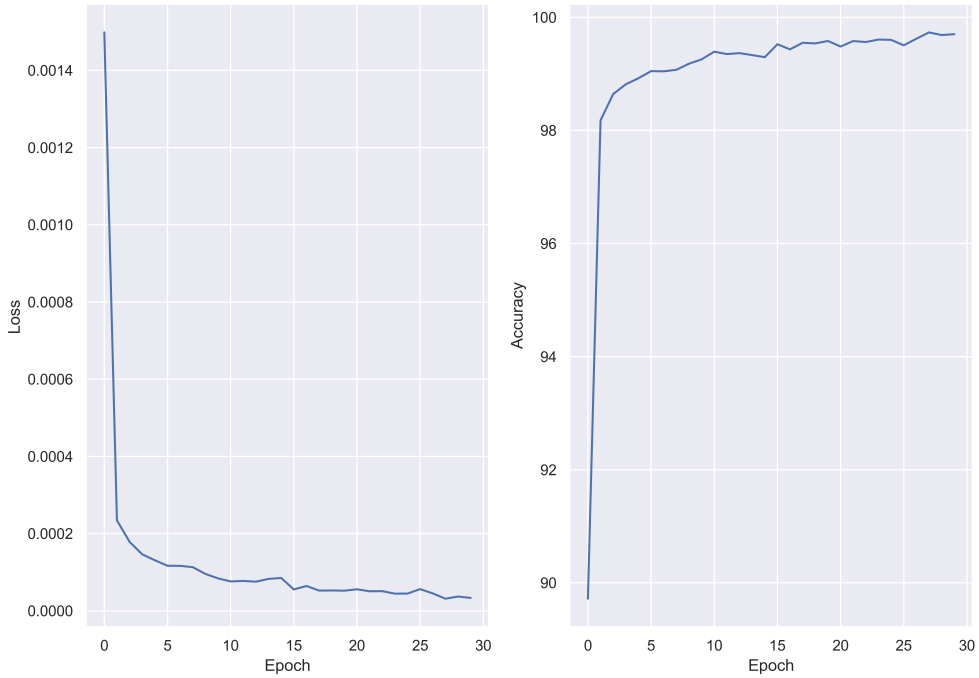

Figure 2: Training loss and accuracy change figures. The left figure corresponds to the change of loss and the right one corresponds to the change of accuracy.

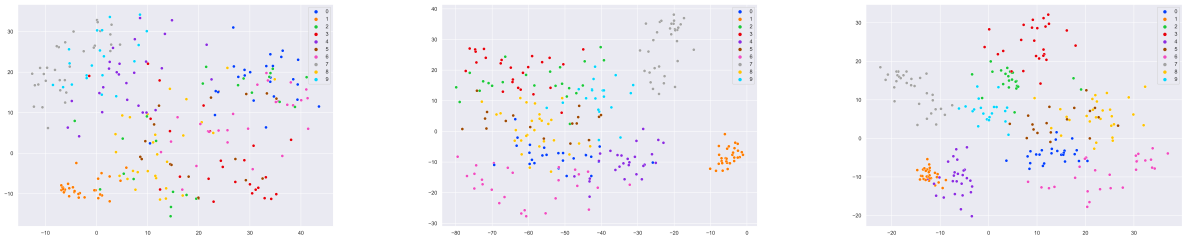

Figure 3: Visualization images of intermediate output tensors with PCA method. Left: Tensors after the first max pooling layer. Middle: Tensors after the second max pooling layer. Right: Tensors after the output linear layer.

Figure 4: Visualization images of intermediate output tensors with t-SNE method. Left: Tensors after the first max pooling layer. Middle: Tensors after the second max pooling layer. Right: Tensors after the output linear layer.

Besides, I output the tensor after the first pooling layer, the tensor after the second pooling layer and the tensor after the output linear layer. And then I use PCA and t-SNE to respectively visualize them in order to see what my model has learned during training. Their corresponding results are depicted in Fig. 3 and Fig. 4, respectively. It can be seen that the first tensor cannot distinguish well each images but after the second CNN layer, my model has learned to distinguish each image with high accuracy.