AI3607 Task 3 Report

Weijiude 521030910418

May 14, 2023

1 Introduction

We use RNN to solve the classification problem on MNIST. We reload the training set where only 10% data with label 0~4 are reserved. To restore the accuracy over the label 0~4, we firstly made attempt with data augmentation like the former homework, but it turns out not working well on MNIST. Then we tried to adjust the loss weight to balance the accuracy of prediction over different labels and found it useful. We also compared the performance between CNN and RNN in solving MNIST classification task and discussed that CNN outperformance RNN practically due to their natural network layer properties.

2 Network Architecture

We slice the 28×28 image into 28 vectors with length 28 in the first step. The vectors are sent into one-dimensional, single-layered LSTMs with hidden layer 64. We take the last element of each LSTM to form a 28×1 feature.

Then we take the 28×1 feature into an MLP which constitutes of two successive fully-connected layers. The two fully-connected layers adjust the dimension of features from 28 to 512 to 10. ReLU layer are used after each of the two fully-connected layers as activate function.

3 Primal experiment

3.1 MNIST Baseline

We use learning rate 0.001 and weight decay 0.0001 to train our model on MNIST dataset for 50 epochs. The overall classification accuracy on testing set is 0.826. The accuracy over label 0~4 is about 0.848. The accuracy over label 5~9 is about 0.803. The confusion matrix of classification is shown in Figure 1.

3.2 Unbalanced MNIST

We reload MNIST training set where only 10% data with label 0~4 are reserved. We keep data with label 5~9 the same in training set.

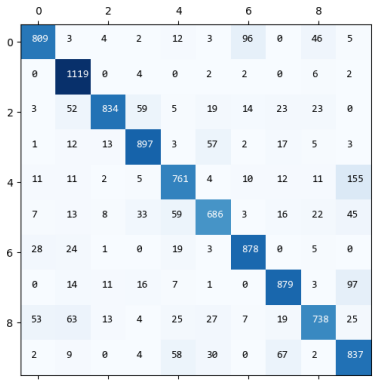

Figure 1: Confusion matrix of testing set. Trained with original training set.

We retrain our model on the unbalanced training set with same hyperparameters. The overall classification accuracy on testing set drops to 0.774. The accuracy over label 0~4 drops to 0.665 while the one over label 5~9 rises to 0.883. The change of accuracy from original training set to unbalanced training set is similar to the result given by CNN. The confusion matrix of classification is shown in Figure 2.

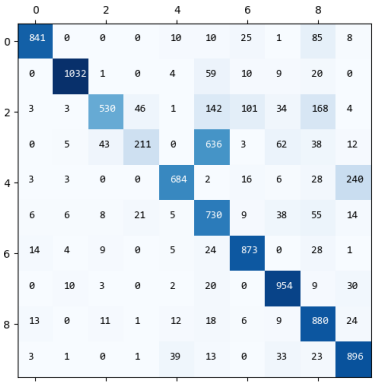

Figure 2: Confusion matrix of testing set. Trained with unbalanced training set.

4 Restore: Data augmentation

4.1 Data augmentation method

Given the well performance of data augmentation on CIFAR-10 in previous task, we natrually consider copying it on MNIST. However, some properties of MNIST limit the choice of data augmentation method.

	Original	Unbalanced 10%	Data Aug 40%	Data Aug 100%	Weighted Loss	Loss weight
0	0.926	**0.858**	0.809	0.844	0.946	6.0
1	0.986	**0.909**	0.826	0.873	0.940	7.0
2	0.808	0.514	0.574	**0.750**	0.824	7.0
3	0.888	0.209	**0.476**	0.427	0.711	7.0
4	0.775	**0.697**	0.589	0.625	0.794	5.0
5	0.769	0.818	0.930	0.874	0.889	
6	0.916	0.911	0.948	0.926	0.906	
7	0.855	0.928	0.923	0.885	0.932	1.0
8	0.758	0.903	0.894	0.842	0.759	
9	0.830	0.888	0.886	0.860	0.769	

Table 1: Test accuracy based on various training method.

MNIST consists of hand-written number data, horizontal and vertical flipping destroy the pattern of numbers. Only one channel exists in MNIST images. MNIST developers ensures that all patterns are at the center of images. Considering these properties, we determine our augmentation method as:

1. Add Gausian noises.
2. Rotate image with considerate angle.

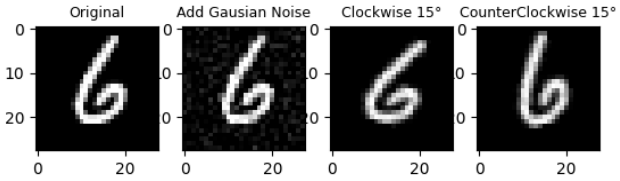

Figure 3: Sample of data augmentation cast on image number 6

4.2 Results

We enlarge the amount of images with label 0~4 in unbalanced training set by casting data augmentation on them. We firstly add Gausian noises and rotate images with ±15° separately and the number of images with label 0~4 has risen to 40% their original number (Data Aug 40% in Table 1). Then we combine the augmentation skills to further raise the number of such images to 100% their original number (Data Aug 100% in Table 1). As we enlarge the amount, the classification accuracy is constrained and shows no significant bounce. Also the accuracy over label 0~4 can not keep up with the accuracy over label 5~9.

4.3 Analysis

Most data augmentation methods that fits CIFAR-10 are not suitable for MNIST due to its properties. We cannot enhance the data diversity by simple and sparse data augmentation methods. Therefore data augmentation on MNIST cannot restore the accuracy over label 0~4 or rebalance the accuracy among labels.

5 Restore: Weighted loss

5.1 Weighted loss method

Another simple idea to restore the balance of test accuracy is to emphasize the critic of wrong prediction on data with label 0~4. We attempt to restore and rebalance the accuracy by respectively modifing the loss weight over the ten labels.

5.2 Results

We train our model based on unbalanced MNIST training set and same hyperparameters.

In the first step, we set the loss weight over label 5~9 as 1 and test accuracy with the loss weight over label 0~4 synchronously from 2 to 10. The accuracy of all labels with weighted loss and the mean accuracy of label 0~4 and label 5~9 are shown in Figure 4 and Figure 5. With the increase of loss weight of label 0~4, the average accuracy of label 5~9 slightly decreases; the average accuracy of label 0~4 reaches its peak at loss weight 7 and goes downwards after that. It indicates that weighted loss method is only effective in a limited range of weight.

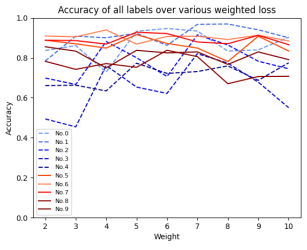

Figure 4: Accuracy of all labels on various weight

Figure 5: Mean accuracy of label 0~4 and 5~9

Moreover, we respectively set the loss weight over label 0~4 according to their best performance demonstrated in Figure 4. The result is shown in the second last column of Table 1 (Weighted Loss). The accuracy of

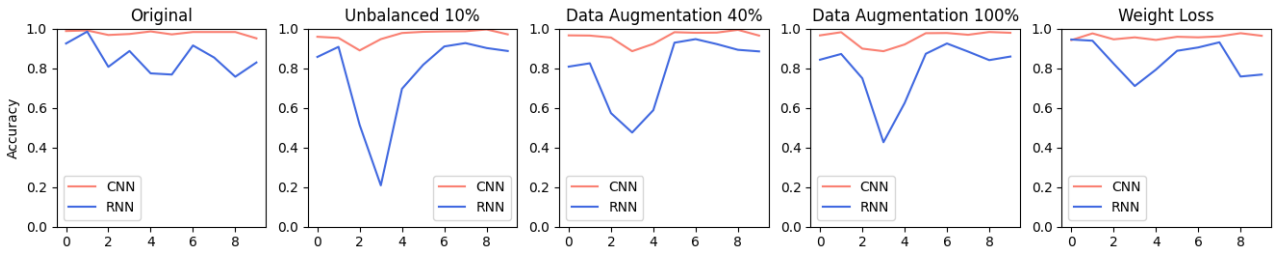

Figure 6: Comparation of accuracy between CNN and RNN based on various training method.

label 0~4 trained with weighted loss has overtaken the one trained without weighted loss (Unbalanced 10%). The classification accuracy of most label with WL has kept up with the accuracy trained with original data. Meanwhile, with weighted loss, the average accuracy of label 0~4 is 0.843; the average accuracy of label 5~9 is 0.851. Weighted loss effectively restored the balance of accuracy among labels.

5.3 Analysis

Based on the idea of multi-tasking learning, we view the classification task on MNIST as fitting ten individual distribution tasks. Our approach to combining multiple losses is to fit a weighted linear sum of the losses of each individual task.

$$\mathcal{L}_{\text{total}} = \sum_i w_i \mathcal{L}_i$$

Let $\boldsymbol{f^W(x)}$ be the output of a neural network with weight $\boldsymbol{W}$ given input $\boldsymbol{x}$ and ground truth $\boldsymbol{y}$. We are fitting

$$\min_{\boldsymbol{f^W}} \sum_i w_i \left\| \boldsymbol{y}_i - \boldsymbol{f^W(x)} \right\|^2$$

The benefit of weighted loss is that, the neural network focus more on difficult tasks and learns more from them.

With a reasonable combination of weights, the performance of neural network can be improved. In the other side, neural network may neglect some facts on less weighted tasks and hence reduce its performance on such tasks. An enormous weight may introduce huge gradient into descent process, hence underperformance globally.

6 Comparation with CNN

Compare the accuracy of MNIST classification between models constructed by CNN and RNN. With same hyperparameters (*e.g. learning rate, weight decay, iteration*), CNN outperformance RNN in all tests. (See Figure 6.)

Trained with unbalanced data and data augmented data, both accuracy of CNN and RNN drops compared with their respective original accuracy. Besides, both trend of accuracy are similar: the classification of No. 2, 3 and 4 are worst.

To explain why CNN outperformance RNN in MNIST classification, we are convinced that, CNN consists of convolution layers and pooling layers. Convolution layers are able to extract the features near a pixel. Pooling layers blur the features, which enable CNN to put emphasis on global information rather than concentrating on partial information. After these layers sense the features of an image, an MLP is used to complete classification task.

In contrast, RNN contains memory paths to extract features from series of pixels. RNN has advenatage in trajectory prediction tasks and less capable of extracting features from multi-dimensional data than CNN.

7 Conclusion

Throwing away most data with certain labels may severely influence theis classification accuracy. Due to some properties of MNIST, reasonable data augmentation methods are limited and hence we can not effectively restore or rebalance the accuracy. Meanwhile weighted loss is useful in this task. Moreover, we can respectively set the loss weight of each label to reach a better performance.

To extract the features of an image, CNN is more capable than RNN under all training methods. Even the best performance of RNN is weaker than CNN in above task. Therefore we should select a reasonable neural network in the first place to fit our model.