Exam Report

Kaiqi Yu

Introduction

This is a report of the mid-term exam for the Machine Learning II. In this exam, only the Multi-Layer Perceptron (MLP) is used to classify blood cells in 4 categories: “red blood cell”, “ring”, “schizont” and “trophozoite”. During the seven days of exam, I adjusted the model in different ways, hoping that it could have better classification ability. The specific work will be described in detail below.

Daily work

Day 01 Load data and simple test

First day, I tried to load the data and perform simple model building and testing by simply modifying the teacher’s code. I held the learning rate and batch size, changed the number of neurons to 400, and epochs to 1000. When testing on the local test set, I got an 0.6936 as mean of F1-score and Cohen’s Kappa Score. But because I started too late, the deadline was already past when I got that result and I didn’t submit the code and model.

Parameters				**Scores**			
LR	N_NEURONS	N_EPOCHS	BATCH_SIZE	F1-score (macro)	Cohen's Kappa Score	Accuracy	Mean Score
1e-3	400	1000	512	0.6644	0.7228	91.86%	0.6936

Day 02 Change model parameters

On the basis of the first day of work, I tried to increase the number of layers of the neural network, expand it to 2 layers, and use BatchNormalization layer after each neuron layer. And I got 0.7875 as mean of F1-score and Cohen’s Kappa Score. But in the predict script, I wrote the wrong path, which made it impossible to test. Therefore, I go no grade at that day.

Parameters				**Scores**			
LR	N_NEURONS	N_EPOCHS	BATCH_SIZE	F1-score (macro)	Cohen's Kappa Score	Accuracy	Mean Score
1e-3	(400, 400)	1000	512	0.7466	0.8285	94.59%	0.7875

Day 03 Modify predict script

Third day, the predict script was modified and the model was uploaded at the same time, and no other changes were made. However, the results obtained are quite different from the local test results. Then after analysis, I thought the reason is that I did not use EarlyStopping, which caused the model to run through all epochs and overfit.

Day 04 Modify preprocessing code

On the fourth day, I tried to modify the preprocessing code. When reading the image data, I tried to change it into a grayscale image for reading. But the result was not ideal. I guess it may be because the three-channel data contains more information than the single-channel data, which affects the fitting and prediction of the model. Because I didn’t get an ideal result -- the average score is only between 0.4-0.5, so I did not upload the file that day.

Day 05 Modify network

On the fifth day, I gave up using grayscale images for training and re-used color images. And changed the BGR type picture data read by cv2 to RGB type. At the same time, changed the size of the resize parameter when reading to (28, 28), and increase the number of neural network layers to 3 layers, with 200, 400, and 200 neurons respectively, changed the epochs to 300 and added EarlyStopping to makes it stop fitting early. I didn't get the ideal result either. Although it was improved compared to the previous day, it did not exceed the score obtained on the second day, so I also chose not to upload it.

Day 06 Adjust the network again

On the sixth day, the number of neurons in each layer of the network was changed to 50, 500, 50, and the batch size was changed to 128. I got a worse result than before. At the same time, the model and script were not submitted due to other events that day.

Day 07 Final change (Data loading, data expansion, neural network)

On the last day I chose to set the parameters of the neural network to:

- Learning rate 1e-4,
- Number of neurons 50, 500, 50
- Number of epochs 300
- Batch size 64

At the same time, I changed the size of the RESIZETO parameter when loading data to 64. Besides, I found that data set was unbalanced, the number of normal red blood cells was much larger than the other three types. So, I searched for relevant information on the Internet and used the over_sampler in the imblearn library to expand the data. After trying various expansion methods, I finally chose RandomOverSampler.

In the end I got an average score of about 0.75, but because I forgot to use EarlyStopping, the model may be over-fitting, and the final test result on the private test set was not very good.

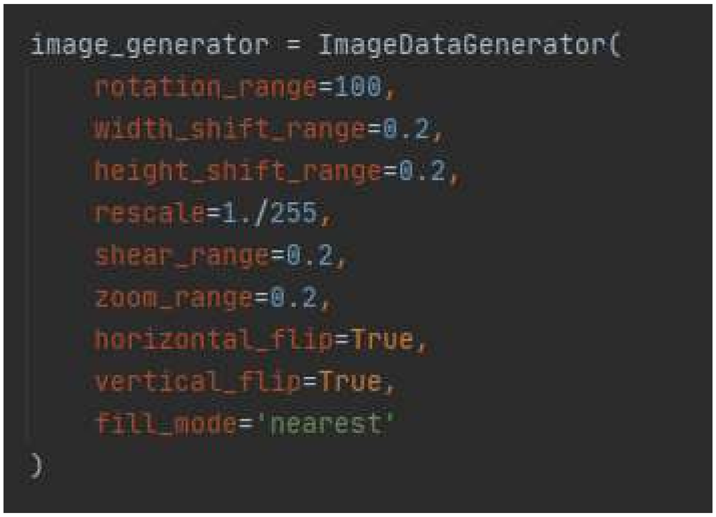
The day after day 07 (A new thing I found)

After the seventh day, I found a new way to solve the imbalanced data set on the Internet. That is, using the ImageDataGenerator method of keras library to generate new image data. Then manually add to the data set or directly flow. I simply tried it, tested it with the previous neural network and got result as below.

Parameters				**Scores**			
LR	N_NEURONS	N_EPOCHS	BATCH_SIZE	F1-score (macro)	Cohen's Kappa Score	Accuracy	Mean Score

1e-4	(1000, 500, 50)	1000	64	0.7926	0.7149	78.65%	0.7537

The parameters of ImageDataGenerator (just for simple test):



```
image_generator = ImageDataGenerator(
    rotation_range=100,
    width_shift_range=0.2,
    height_shift_range=0.2,
    rescale=1./255,
    shear_range=0.2,
    zoom_range=0.2,
    horizontal_flip=True,
    vertical_flip=True,
    fill_mode='nearest'
)
```

Conclusion and Future Work

During these seven days, I tried to modify the preprocessing method, neural network, and data set. The final model result may not be the best model. This shows that my knowledge is not enough, and the improvement and adjustment of the model is not ideal. But I still learned a lot of new things, and at the same time I also realized the pros and cons of each method/improvement. In the future work, I will learn more methods and make more improvements. At the same time, I hope I can get a balance between the methods to make the model's scores are at an excellent level.