A Deep Neural Network for Covid-19 Classification

Yuxiang Zhang

I. Methods

In this experiment, we try to build a deep neural network for Covid-19 CT image classification. Then, compare the results performed by different models. The selected baseline model is Resnet18. This section includes two parts, in the first part, we will briefly introduce the designed network architecture. In the second part, we will present the training strategies and tricks used in the network.

A. Architecture of Designed Deep Neural Network

Because it is a classification task for images, we choose to build a convolutional neural network(CNN). The details can be seen in the flow chart in Fig.1.

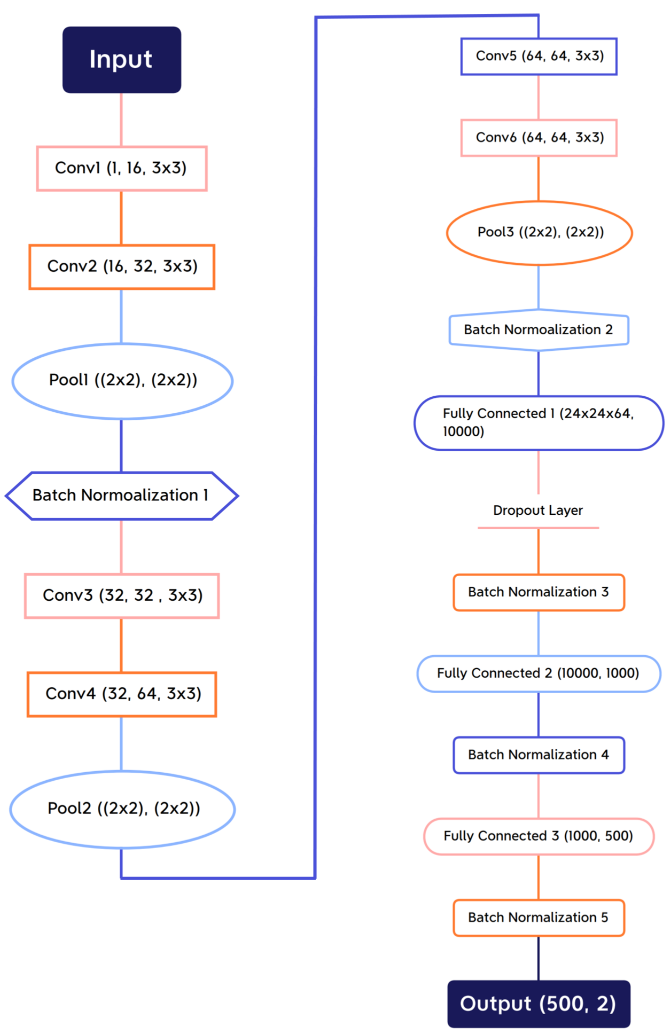

Fig. 1. Flow Chart of CNN Architecture.

The CNN model has 18 layers, including 6 convolution layers, 3 pooling layers, 4 fully connected layers and 5 batch normalization layer. The input data is 224*224 image with 1 channel. The network can be divided into 4 blocks, in the first block, we use 2 convolution layers transforming the image into 220*220*32 and maxpooling into 110*110*32, then, a batch normalization layer was added. The second and third block have a similar structure with the first block, they can transform the data into 53*53*64 and 24*24*64, respectively. The forth block contains 4 fully connected layers, which count for classify the CT images into 2 classes, also, batch normalization layer is added between each two layers.

B. Training and Optimization Strategies

1) Data Augmentation: Use this method generating more data from existing data to prevent overfitting due t little data in the dataset. In this model, we apply random crop and random horizontal flip. We also scale the values to [0, 1] for normalization and transform data to have a 0.5 mean and unit standard deviation.

2) ADAM: This method computes adaptive learning rates for each parameter at a high speed. Additionally, it can also keeps an exponentially decaying average of past gradients similar to momentum.

3) Batch Normalization: Normalize the data in order to prevent gradient explosion or disappearance while accelerating the convergence rate of the model.

4) Dropout: Randomly inactivating neurons at certain probability reduces computational complexity and prevents overfitting. In this experiment, we set the dropout rate at 0.5.

II. Experimental settings

The training dataset and test dataset come from COVID-CT-Dataset, which has 349 CT images containing clinical findings of COVID-19 from 216 patients. The dataset was splited into 3 subdatasets, training dataset, validation dataset and test dataset. We use the entire training dataset(contains 425 data) to train the designed network, while the entire test dataset(contains 203 data) for evaluating the model. Noticed that that the size of dataset is to small, we preprocess the data by data augmentation.

The detail parameter settings: After trying different batch size, we choose 100, which balance the computing efficiency and performance. Learning rate was set at 0.0001, which is an empirical setting. Dropout rate was set at 0.5 The number of epochs was set at 100.

In order to improve the training efficiency of model, we set the maximum number of epoch and the model will stop training when the parameters change to a certain extent. After comparing the prediction results of different optimization strategies , we choose the one with better performance for subsequent experiment. The environment we use is python 3.9, numpy 1.20.3, pandas 1.3.4, scikit-learn 0.21.2, torch 1.10.0 and torchvision 0.12.0.

III. Results & Discussion

A. Performance of the Designed CNN

First of all, we employ CNN designed by us to the test dataset with different epochs, using accuracy as the evaluation index. It is found that the effect is better when epoch at 50(see Fig.2). Then, we set different batch sizes according to the period of 10, finding that accuracy is highest at 100 batch size (see Fig.2). Finally, training and testing the model with batch size at 100 and epoch at 50.

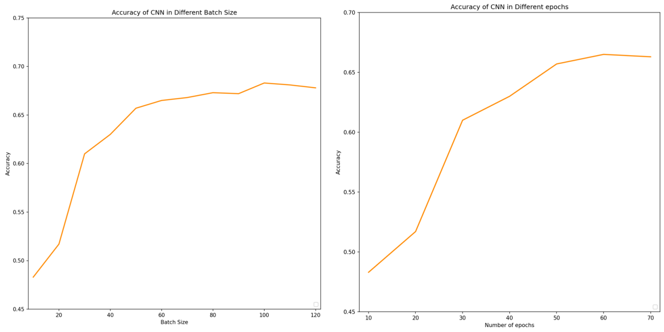

Fig. 2. Accuracy of CNN in Different epochs.

Generally, we find the parameter settings with the better performance on test dataset. However, we only gain a 74.5% acc when using these settings. We conclude that due to the increase of network layers and the small feature points of images, the features extracted by the model may be lost to some extent. Therefore, we employ Resnet18, which takes into account the shallow characteristics, on the same train and test dataset.

B. Performance of the Baseline Model Resnet18

The network structure of Resnet18 is shown in the Fig.3. It can be found that on the basis of ensuring a deeper network, Resnet18 also adds a shortcut to pay attention to the shallow characteristics. Then, Resnet18 is employed for the same

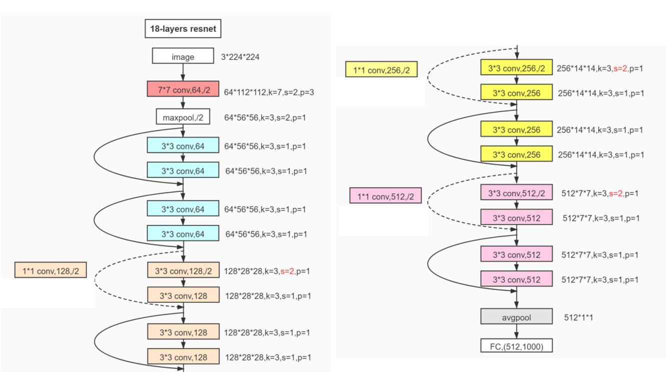

Fig. 3. Flow Chart of Resnet18 Architecture.

dataset. It can be found that the convergence speed of Resnet18 is very fast, and it can quickly achieve high accuracy (see Fig.4). We can conclude that shortcut does play an important role in reducing the feature disappearance problem when the convolution layer is deeper.

C. Comparison of Two Different Model

Firstly, we drow the ROC curve of two models, finding that the AUC of Resnet18 (0.86) is obviously higher than our CNN (0.79). Then, in order to compare whether there were significant

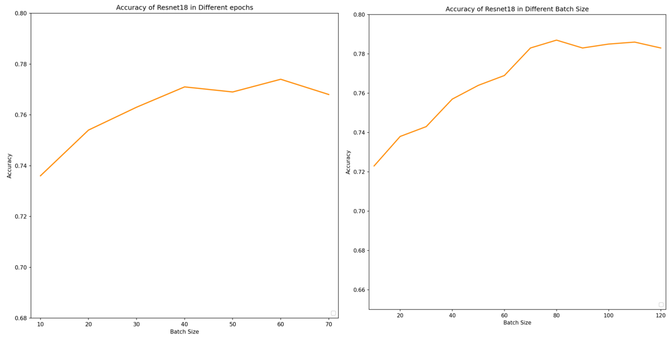

Fig. 4. Accuracy of Resnet18 in Different epochs and batch size.

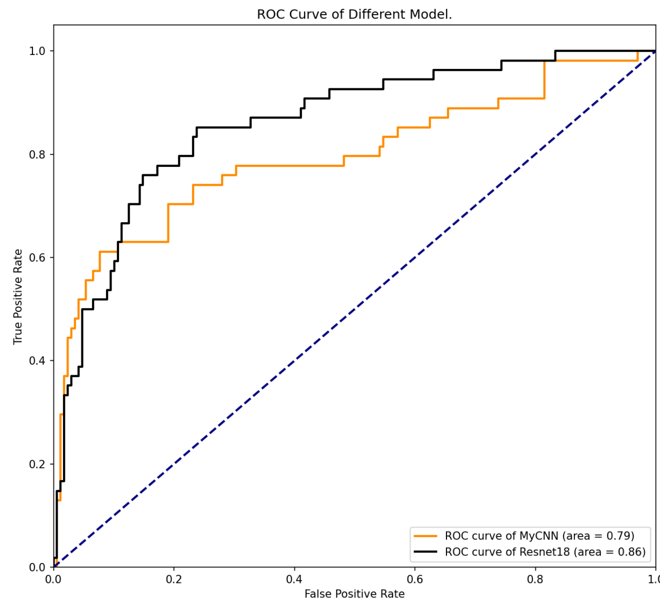

Fig. 5. ROC curve of Different models.

differences between the predictive abilities of different models, we performed statistical tests for the two models. Because of the use of the same dataset and the small number of models involved in the comparison, we used McNemar's test, which was tested with 95% confidence interval.

The null hypothesis H_0 is that "Two models have similar error rate on the test dataset", while the alternative hypothesis H_1 is that "Two models have different error rate on the test dataset". Using the 2×2 contingency table, the p-value of test between two models is 0.018, which is small than 0.05, so we have sufficient evidence to reject H_0, the predictive ability of Resnet18 is significantly better than our CNN in 95% confidence interval.

References

[1] Zhao, Jinyu and Zhang, Yichen and He, Xuehai and Xie, Pengtao,"COVID-CT-Dataset: a CT scan dataset about COVID-19", arXiv preprint arXiv:2003.13865,2020

[2] He, Xuehai and Yang, Xingyi and Zhang, Shanghang, and Zhao, Jinyu and Zhang, Yichen and Xing, Eric, and Xie, Pengtao, "Sample-Efficient Deep Learning for COVID-19 Diagnosis Based on CT Scans", medrxiv ,2020