Project Report

Name : Jhao-Han Chen

Data:12/1/2019

- **Problem Statement**

Millions of people suffer from diabetic retinopathy, the leading cause of blindness among working aged adults. Aravind Eye Hospital in India hopes to detect and prevent this disease among people living in rural areas where medical screening is difficult to conduct. So they set up a competition on kaggle.com for competitors to build model which can help ophthalmologists to identify potential patients.

- **Evaluate metrix**

Maximize the quadratic weighted kappa, which measures the agreement between true labels and predicted results. This metric typically varies from 0 (random agreement between raters) to 1 (complete agreement between raters).

- **Data Understanding**

Training data contain 3662 samples, each sample contains one image and it's severity scale. Ophthalmologists rate each clinician for the severity of diabetic retinopathy on a scale of 0 to 4. "0"represent no decease, "1" is mild, "2" is moderate, "3"is severe, and "5" has proliferative diabetic retinopathy.

- **Training images preprocessing**

First, I load the image into python, and I crop the dark area which is useless while analyzing. I then try gray scale and feel understand better for some pictures, as color distraction is gone. For example, we can see the blood clearer. Next, I resize all the image to (300,300,3). At last, I add some Gaussian noise to increase robustness. After doing these steps, I get my final training data.

- **Data augmentation**

There are only 3662 training images, which are a very small training set, but the deep learning model may contain 100000 or more parameters, we no doubt will face overfitting problem while training. To deal with the problem, I generate data by changing angles, flipping, and randomly cropping the training images to increase my training samples. Also, I split the training data into 80% train and 20% validation data to check if the model has overfitting problem.

- Define callback function

The main callback function is maximize the Cohan-Kappa score, and I set an early-stopping with patient=5 and also reduce learning rate with patient=3 by 0.000006 per epoch.

- Model 1 – simple CNN

I start with hand-built CNN model. Even though the model just contain 4 convolutional layers and 2 full-connected layers, it has 175k parameters to train. I add some dropout layers and batch normalization layers to avoid overfitting problem and increase accuracy. The accuracy of the model is 0.73 and the validate accuracy is 0.7.

- Model 2 – Resnet50

Then, in order to go deeper to the model, I try Resnet model which contain 49 convolutional layers and I add two full-connected layers as top layers. It increase accuracy by using skip connection to deal with vanish gradient problem. First, I pretrain the top layer of the Resnet model and then fine tune the complete model. The accuracy of the model is 0.87 and the validate accuracy is 0.82 by training 17 epochs. The validation Cohen-Kappa score is 0.9.

- Model 3 – Efficientnet B3

At last, same as model 2, I pretrain a large dataset first and fine tune the complete model. This time I use efficientnet model, which is a brand new model release by Google AI in June 2019. It introduces a systematic way to scale CNN in a nearly optimal way. The accuracy of the model is 0.9 and the validate accuracy is 0.79 by training 15 epochs. The validation Cohen-Kappa score is 0.89.

- ## Evaluate the model

At last, I predict the test image using these three models. The last two model get higher accuracy. Therefore, it’s better to pretrain the large dataset first before training the complete model.