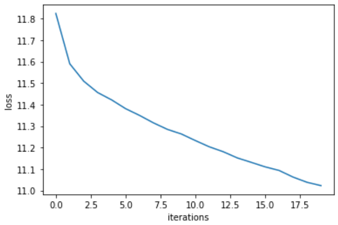
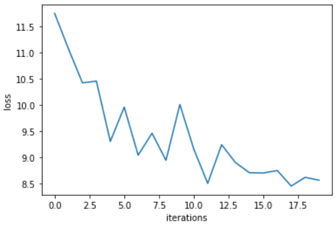
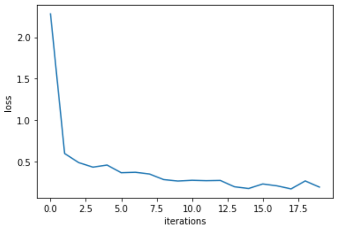
Report of HW4 - Convolutional Neural Network

Student ID: 109062211 Name: 張惇媛

I. Do a simple experiment to find out whether your convolutional neural network exceeds your linear model (under the same number of epochs).

	Linear Model	**My CNN Model**	**Tensorflow CNN Model**
Epochs	20	20	20
Training time	584 ms	17 min 58 s	28.8 s
Accuracy (training/validation)	0.719/0.550	0.798/0.750	0.9587/0.8667
Number of parameters	32,833	7,153	96,494,217
Training loss curve			
Others	batch_size=32 fixed learning rate	batch_size=32 learning rate changes	batch_size=32 pre_trained_model +(Fully) connected_layer

II. For the advanced part: Describe how you design or choose your own model architecture, and how you choose loss function and optimizer.

- Model – "Xception + (Fully) Connected Layer"

  I choose my model from some popular pre-trained models, InceptionV3, ResNet and EfficientNet, evaluating the performance of each model by the accuracy rate. Finally, choose the best model – Xception, which is the improved version of InceptionV3.

  At the end, I flatten the last layer of the Xception model and add 3 Connected layers. One of them(the first one) is not a fully connected layer, which is to avoid overfitting, and the others are fully connected layers. Also, for the first 2 connected layers, I use advanced activation function – LeakyReLU & PReLU, and the last layer used sigmoid function to get the output of 1 classes for the prediction answer.

- Loss function – "binary_crossentropy"

  Since we implement binary classifier, I use binary cross entropy as the loss function, which is the same as my CNN model.

- Optimizer – "Adam"

  I select from the optimizers in keras, like SGD, RMSprop, Adam. Since the update for SGD is noisy, we consider RMSprop and Adam as the optimizer. Compared methods between RMSprop and Adam, both have adaptive learning rate methods. However, Adam is the most popular method because of the addition of momentum and bias correction, which cause Adam usually has better performance than RMSprop. Therefore, I think Adam is a good choice to be the optimizer for my model. In the end, I finetune the parameter of learning_rate to get the final model of pulmonary disease prediction.

ADDITIONAL NOTE:

I import the package of "ipython-autotime" in this assignment for keeping track of the execution time, which is used for part I. the row of training time in the report. This package will print the execution time of each block as the output.

I import the package of "cv2" in the advanced part of this assignment for resizing the images and normalization. Since I use the pre-trained model of Xception, the smallest size for this model is (71, 71, 3). Therefore, I use cv2.resize to scale the image into the required smallest size. Also, I use cv2.normalize to simplify the action of min-max scaling since there are 3 channels.

For the advanced part, I use some function in Tensorflow: "ImageDataGenerator" to do data argumentation, "Callback" to prevent overfitting, and the others functions for pre-trained model & optimizer. In addition, because I use sigmoid function in the last layer, the output of my model is 1 class. To satisfy the final prediction of my model, I change the function of np.argmax into np.rint to get the answer of output["advanced_pred_test"].