Deep Learning homework 5 completed by Eri Kim, Aidan Horn, and Adeline Evans

Problem 1.

```
860/860 [==============================] - 4s 5ms/step - loss: 0.1888 - accuracy: 0.9295 - val_loss: 0.1969 - val_accuracy: 0.9300
Epoch 10/20
855/860 [============================>.] - ETA: 0s - loss: 0.1843 - accuracy: 0.9298
Epoch 10: val_loss did not improve from 0.19545
860/860 [==============================] - 4s 5ms/step - loss: 0.1844 - accuracy: 0.9296 - val_loss: 0.1956 - val_accuracy: 0.9310
Epoch 11/20
854/860 [============================>.] - ETA: 0s - loss: 0.1857 - accuracy: 0.9289
Epoch 11: val_loss did not improve from 0.19545
860/860 [==============================] - 4s 5ms/step - loss: 0.1858 - accuracy: 0.9289 - val_loss: 0.1979 - val_accuracy: 0.9284
Epoch 12/20
854/860 [============================>.] - ETA: 0s - loss: 0.1828 - accuracy: 0.9307
Epoch 12: val_loss did not improve from 0.19545
860/860 [==============================] - 4s 5ms/step - loss: 0.1825 - accuracy: 0.9308 - val_loss: 0.2059 - val_accuracy: 0.9272
Epoch 13/20
853/860 [============================>.] - ETA: 0s - loss: 0.1838 - accuracy: 0.9306
Epoch 13: val_loss did not improve from 0.19545
860/860 [==============================] - 4s 5ms/step - loss: 0.1838 - accuracy: 0.9306 - val_loss: 0.2024 - val_accuracy: 0.9276
Epoch 14/20
858/860 [============================>.] - ETA: 0s - loss: 0.1803 - accuracy: 0.9315
Epoch 14: val_loss did not improve from 0.19545
860/860 [==============================] - 4s 5ms/step - loss: 0.1802 - accuracy: 0.9316 - val_loss: 0.2017 - val_accuracy: 0.9302
Epoch 15/20
851/860 [============================>.] - ETA: 0s - loss: 0.1774 - accuracy: 0.9330
Epoch 15: val_loss did not improve from 0.19545
860/860 [==============================] - 4s 5ms/step - loss: 0.1773 - accuracy: 0.9330 - val_loss: 0.1985 - val_accuracy: 0.9284
Epoch 16/20
855/860 [============================>.] - ETA: 0s - loss: 0.1790 - accuracy: 0.9310
Epoch 16: val_loss did not improve from 0.19545
860/860 [==============================] - 4s 5ms/step - loss: 0.1790 - accuracy: 0.9310 - val_loss: 0.1993 - val_accuracy: 0.9316
Epoch 17/20
857/860 [============================>.] - ETA: 0s - loss: 0.1794 - accuracy: 0.9327
Epoch 17: val_loss did not improve from 0.19545
860/860 [==============================] - 4s 5ms/step - loss: 0.1794 - accuracy: 0.9327 - val_loss: 0.1960 - val_accuracy: 0.9298
Epoch 18/20
854/860 [============================>.] - ETA: 0s - loss: 0.1730 - accuracy: 0.9347
Epoch 18: val_loss did not improve from 0.19545
860/860 [==============================] - 4s 5ms/step - loss: 0.1729 - accuracy: 0.9347 - val_loss: 0.1998 - val_accuracy: 0.9284
Epoch 19/20
857/860 [============================>.] - ETA: 0s - loss: 0.1765 - accuracy: 0.9334
Epoch 19: val_loss did not improve from 0.19545
860/860 [==============================] - 4s 5ms/step - loss: 0.1766 - accuracy: 0.9334 - val_loss: 0.1987 - val_accuracy: 0.9316
Epoch 20/20
854/860 [============================>.] - ETA: 0s - loss: 0.1724 - accuracy: 0.9355
Epoch 20: val_loss did not improve from 0.19545
860/860 [==============================] - 4s 5ms/step - loss: 0.1721 - accuracy: 0.9356 - val_loss: 0.2060 - val_accuracy: 0.9276
<keras.callbacks.History at 0x7fcb4d290b10>
```

Fig 2: Screenshot showing gradient updates on the training set.

Test Accuracy

```
# Evaluate the model on test set
score = model.evaluate(x_test, y_test, verbose=0)

# Print test accuracy
print('\n', 'Test accuracy:', score[1])
```

```
 Test accuracy: 0.9225999712944031
```

Fig 1: Screenshot showing test accuracy of at least 92%.

Problem 2.

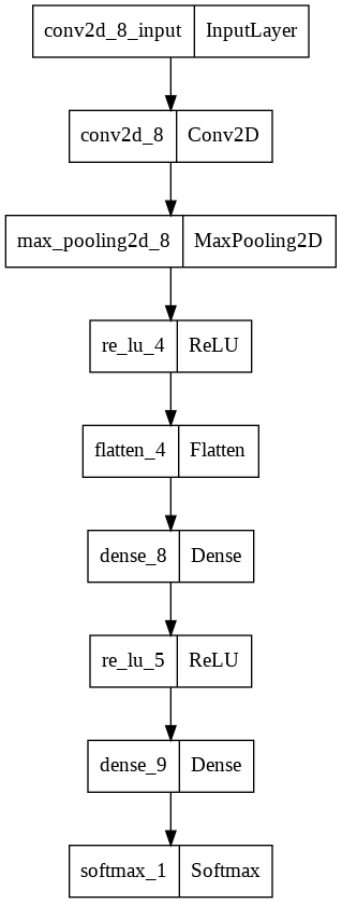

Fig 3: output of tf.keras.utils.plot_model

```
-----TensorFlow Convolutional Neural Network prediction-----
[8.6958230e-10 3.4364154e-11 3.1793821e-08 1.3371185e-10 1.1007579e-08
 9.5289181e-09 2.7218139e-09 1.4777918e-06 4.9786246e-09 9.9999845e-01]
-----Fully-connected Neural Network prediciton-----
[8.69581026e-10 3.43641141e-11 3.17938122e-08 1.33711544e-10
 1.10075868e-08 9.52893705e-09 2.72181437e-09 1.47779202e-06
 4.97863013e-09 9.99998461e-01]
```

Fig 4: comparison of predictive distributions between CNN and a Fully-Connected NN