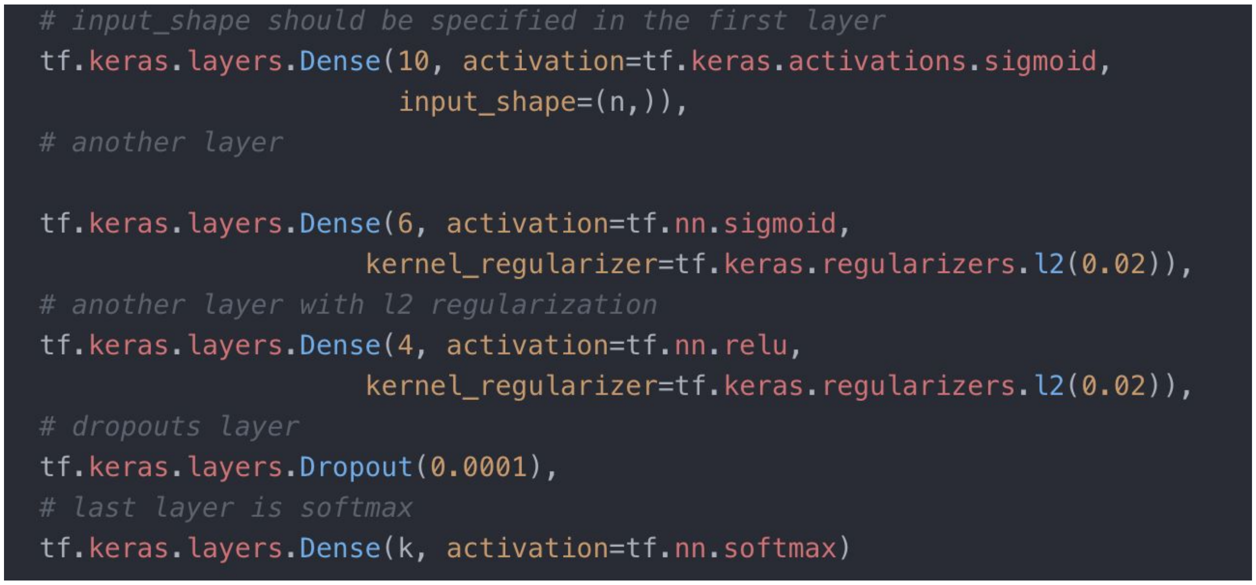
Graded homework 1 report

Feng Mi 2021274537

1.Describe the network:

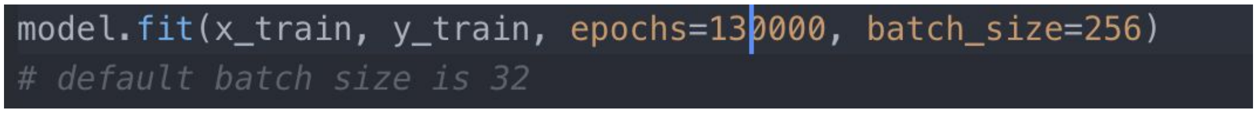


```
# input_shape should be specified in the first layer
tf.keras.layers.Dense(10, activation=tf.keras.activations.sigmoid,
                      input_shape=(n,)),
# another layer

tf.keras.layers.Dense(6, activation=tf.nn.sigmoid,
                    kernel_regularizer=tf.keras.regularizers.l2(0.02)),
# another layer with l2 regularization
tf.keras.layers.Dense(4, activation=tf.nn.relu,
                    kernel_regularizer=tf.keras.regularizers.l2(0.02)),
# dropouts layer
tf.keras.layers.Dropout(0.0001),
# last layer is softmax
tf.keras.layers.Dense(k, activation=tf.nn.softmax)
```

Layers from bottom to up.

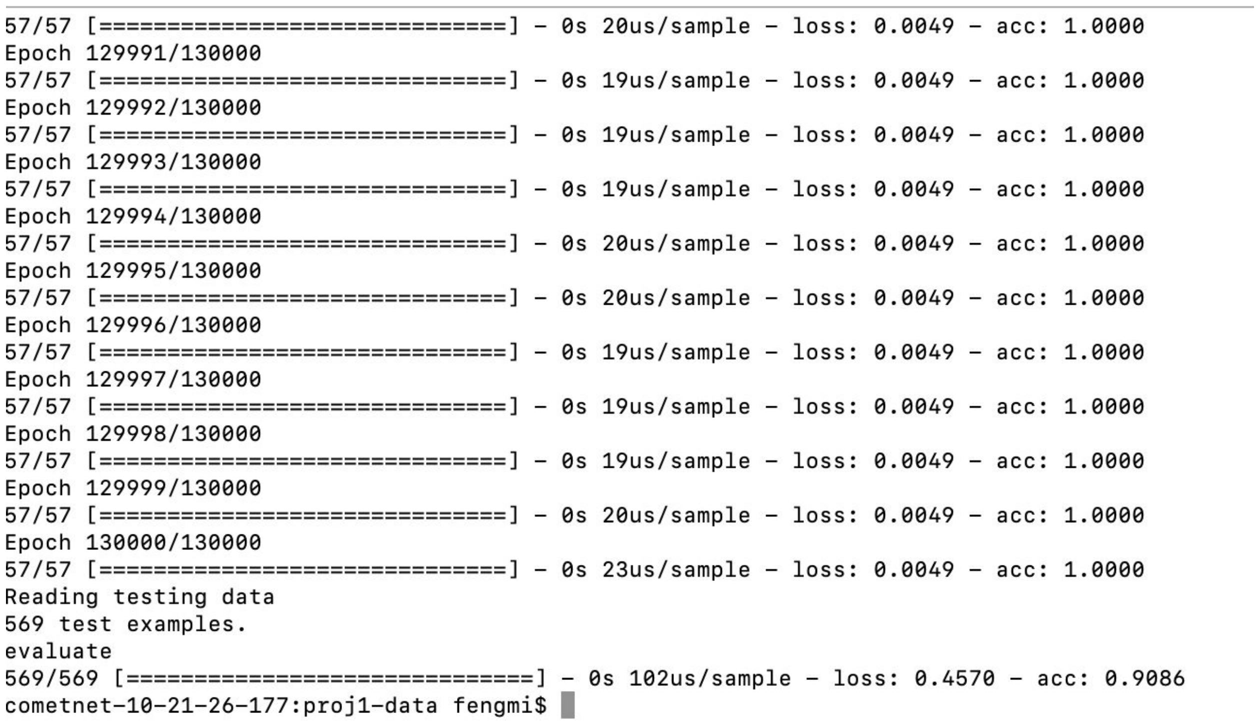
1. It has 10 nodes and perceptron using sigmoid.
2. It has 6 nodes and perceptron using relu. The parameter set to be 0.02.
   Explanation: Regularizers allow applying penalties on layer parameters or layer activity during optimization. These penalties are incorporated in the loss function that the network optimizes.
3. It has 4 nodes and perceptron using relu. The parameter set to be 0.02.
4. This is a dropouts layer. The parameter set to be 0.001.
   Explanation: View training as if multiple models are being trained separately to produce the same output. The final output is produced as an average of these models. Random models that overfit are expected to overfit in a different way. Therefore, averaging them is expected to reduce overfitting.
5. This is a softmax layer.



```
model.fit(x_train, y_train, epochs=130000, batch_size=256)
# default batch size is 32
```

6. 6. The epochs I set it to be 130000 and the batch size I set it to be 256.
   Explanation:Epoch: an arbitrary cutoff, generally defined as "one pass over the entire dataset", used to separate training into distinct phases, which is useful for logging and periodic evaluation.So, in other words, a number of epochs means how many times you go through your training set.The model is updated each time a batch is processed, which means that it can be updated multiple times during one epoch.
7.

Results of experiments/accuracy:



```
57/57 [==============================] - 0s 20us/sample - loss: 0.0049 - acc: 1.0000
Epoch 129991/130000
57/57 [==============================] - 0s 19us/sample - loss: 0.0049 - acc: 1.0000
Epoch 129992/130000
57/57 [==============================] - 0s 19us/sample - loss: 0.0049 - acc: 1.0000
Epoch 129993/130000
57/57 [==============================] - 0s 19us/sample - loss: 0.0049 - acc: 1.0000
Epoch 129994/130000
57/57 [==============================] - 0s 20us/sample - loss: 0.0049 - acc: 1.0000
Epoch 129995/130000
57/57 [==============================] - 0s 20us/sample - loss: 0.0049 - acc: 1.0000
Epoch 129996/130000
57/57 [==============================] - 0s 19us/sample - loss: 0.0049 - acc: 1.0000
Epoch 129997/130000
57/57 [==============================] - 0s 19us/sample - loss: 0.0049 - acc: 1.0000
Epoch 129998/130000
57/57 [==============================] - 0s 19us/sample - loss: 0.0049 - acc: 1.0000
Epoch 129999/130000
57/57 [==============================] - 0s 20us/sample - loss: 0.0049 - acc: 1.0000
Epoch 130000/130000
57/57 [==============================] - 0s 23us/sample - loss: 0.0049 - acc: 1.0000
Reading testing data
569 test examples.
evaluate
569/569 [==============================] - 0s 102us/sample - loss: 0.4570 - acc: 0.9086
cometnet-10-21-26-177:proj1-data fengmi$
```

As it shows in the screenshot, the loss value is 0.4570 and the accuracy value is: 0.9086

Project Description and achievements:
In this project, it needs to run experiments on the Wisconsin Breast Cancer dataset. There are 569 examples, each labeled as 0 or 1. Classical approaches achieve accuracy of over 98%, using cross-validation while I am achieving 0.9086 with fewer data. I think I have achieved a very good result. Additionally, I try to use many other the functionality of tensorflow, not only the parts that were described in class.

How to run the code:
python3 fraction_xy.py x_test.csv y_test.csv 0.1 7
python3 proj1.py.