INF552 Machine Learning for Data Informatics HW5

1. Contribution:

a. Shih-Yu Lai (40%): (Monday)

Email: shihyula@usc.edu

USC-ID: 7183984563

Implement the Back Propagation Algorithm code (NeuralNetwork.py).

b. Shiuan-Chin Huang (30%): (Monday)

Email: shiuanch@usc.edu

USC-ID: 9815633745

Write the ReadMe.

c. Danhui Wu (30%): (Tuesday)

Email: danhuiwu@usc.edu

USC-ID: 8038800437

Implement the sklearn_NeuralNetwork.py.

2. Steps of the algorithm and data structure

Step 1: Load the images from training data txt and convert their pixels into pixels with gray scale ranging from 0 to 1. Then label them with 1 if the word “down” is in the image, label them with 0 otherwise.

Step2: Initialized the Neural Network with 0.1 learning rate, hidden layer of size 100, set the sigmoid function 1.0 / (1.0 + np.exp(-x)), set 1000 epochs, and store the training data and training label we got from the previous step, and hidden layer as numpy arrays.

Step3: Start training. First assign random weights for forward propagation network. We calculate the random weights with samples from normal distribution.

Step4: We get a random training data and pass it to forward propagation and calculate the outputs for each neuron.

Step5: Use the output from forward propagation for back propagation to calculate the error to update the weights. w’ = w – learning rate* (∂ error/ ∂ w). Repeat Step3 to Step5 for 1000 times.

Step6: predict testing data and print the outcome.

3. Our Outcome

NeuralNetwork.py

The accuracy is slightly different in each compile because of the random data for training step.

1. Accuracy: 85.54%

```
Predict wrong : test-image is gestures/A/A_down_1.pgm and output is [0.]
Predict wrong : test-image is gestures/A/A_down_2.pgm and output is [0.]
Predict correct : test-image is gestures/A/A_hold_1.pgm and output is [0.]
Predict correct : test-image is gestures/A/A_hold_10.pgm and output is [0.]
Predict correct : test-image is gestures/A/A_stop_1.pgm and output is [0.]
Predict correct : test-image is gestures/A/A_stop_4.pgm and output is [0.]
Predict correct : test-image is gestures/A/A_up_1.pgm and output is [0.]
Predict correct : test-image is gestures/A/A_up_10.pgm and output is [0.]
Predict correct : test-image is gestures/B/B_down_1.pgm and output is [1.]
Predict wrong : test-image is gestures/B/B_down_2.pgm and output is [0.]
Predict correct : test-image is gestures/B/B_hold_1.pgm and output is [0.]
Predict correct : test-image is gestures/B/B_hold_2.pgm and output is [0.]
Predict correct : test-image is gestures/B/B_stop_1.pgm and output is [0.]
Predict correct : test-image is gestures/B/B_stop_2.pgm and output is [0.]
Predict correct : test-image is gestures/B/B_up_1.pgm and output is [0.]
Predict correct : test-image is gestures/B/B_up_4.pgm and output is [0.]
Predict correct : test-image is gestures/C/C_down_1.pgm and output is [1.]
Predict correct : test-image is gestures/C/C_down_2.pgm and output is [1.]
Predict correct : test-image is gestures/C/C_hold_1.pgm and output is [0.]
Predict correct : test-image is gestures/C/C_hold_2.pgm and output is [0.]
Predict correct : test-image is gestures/C/C_stop_2.pgm and output is [0.]
Predict correct : test-image is gestures/C/C_stop_3.pgm and output is [0.]
Predict correct : test-image is gestures/C/C_up_1.pgm and output is [0.]
Predict wrong : test-image is gestures/D/D_down_1.pgm and output is [0.]
Predict wrong : test-image is gestures/D/D_down_2.pgm and output is [0.]
Predict correct : test-image is gestures/D/D_hold_1.pgm and output is [0.]
Predict correct : test-image is gestures/D/D_hold_2.pgm and output is [0.]
Predict correct : test-image is gestures/D/D_hold_6.pgm and output is [0.]
Predict correct : test-image is gestures/D/D_stop_1.pgm and output is [0.]
Predict correct : test-image is gestures/D/D_stop_2.pgm and output is [0.]
Predict correct : test-image is gestures/D/D_up_1.pgm and output is [0.]
Predict correct : test-image is gestures/D/D_up_3.pgm and output is [0.]
Predict wrong : test-image is gestures/E/E_down_1.pgm and output is [0.]
Predict correct : test-image is gestures/E/E_hold_1.pgm and output is [0.]
Predict correct : test-image is gestures/E/E_hold_5.pgm and output is [0.]
Predict correct : test-image is gestures/E/E_stop_1.pgm and output is [0.]
Predict correct : test-image is gestures/E/E_stop_2.pgm and output is [0.]
Predict correct : test-image is gestures/E/E_up_1.pgm and output is [0.]
Predict correct : test-image is gestures/E/E_up_2.pgm and output is [0.]
Predict wrong : test-image is gestures/F/F_down_1.pgm and output is [0.]
Predict wrong : test-image is gestures/F/F_down_4.pgm and output is [0.]
Predict correct : test-image is gestures/F/F_hold_1.pgm and output is [0.]
Predict correct : test-image is gestures/F/F_hold_2.pgm and output is [0.]
Predict correct : test-image is gestures/F/F_stop_2.pgm and output is [0.]
Predict correct : test-image is gestures/F/F_stop_5.pgm and output is [0.]
Predict wrong : test-image is gestures/G/G_down_2.pgm and output is [0.]
Predict wrong : test-image is gestures/G/G_down_3.pgm and output is [0.]
Predict correct : test-image is gestures/G/G_hold_4.pgm and output is [0.]
Predict correct : test-image is gestures/G/G_stop_2.pgm and output is [0.]
Predict correct : test-image is gestures/G/G_stop_5.pgm and output is [0.]
Predict correct : test-image is gestures/G/G_up_2.pgm and output is [0.]
Predict correct : test-image is gestures/G/G_up_5.pgm and output is [0.]
Predict wrong : test-image is gestures/H/H_down_2.pgm and output is [0.]
Predict correct : test-image is gestures/H/H_hold_10.pgm and output is [0.]
Predict correct : test-image is gestures/H/H_hold_2.pgm and output is [0.]
Predict correct : test-image is gestures/H/H_hold_5.pgm and output is [0.]
```

```
Predict correct : test-image is gestures/H/H_hold_5.pgm and output is [0.]
Predict correct : test-image is gestures/H/H_stop_5.pgm and output is [0.]
Predict correct : test-image is gestures/H/H_stop_6.pgm and output is [0.]
Predict correct : test-image is gestures/H/H_up_5.pgm and output is [0.]
Predict correct : test-image is gestures/I/I_hold_2.pgm and output is [0.]
Predict correct : test-image is gestures/I/I_hold_5.pgm and output is [0.]
Predict correct : test-image is gestures/I/I_down_3.pgm and output is [1.]
Predict correct : test-image is gestures/I/I_stop_5.pgm and output is [0.]
Predict correct : test-image is gestures/I/I_stop_6.pgm and output is [0.]
Predict correct : test-image is gestures/I/I_up_2.pgm and output is [0.]
Predict correct : test-image is gestures/I/I_up_3.pgm and output is [0.]
Predict correct : test-image is gestures/J/J_down_5.pgm and output is [1.]
Predict wrong : test-image is gestures/J/J_down_6.pgm and output is [0.]
Predict correct : test-image is gestures/J/J_hold_2.pgm and output is [0.]
Predict correct : test-image is gestures/J/J_hold_3.pgm and output is [0.]
Predict correct : test-image is gestures/J/J_stop_7.pgm and output is [0.]
Predict correct : test-image is gestures/J/J_stop_8.pgm and output is [0.]
Predict correct : test-image is gestures/J/J_up_1.pgm and output is [0.]
Predict correct : test-image is gestures/J/J_up_2.pgm and output is [0.]
Predict correct : test-image is gestures/K/K_down_2.pgm and output is [1.]
Predict correct : test-image is gestures/K/K_down_3.pgm and output is [1.]
Predict correct : test-image is gestures/K/K_hold_1.pgm and output is [0.]
Predict correct : test-image is gestures/K/K_hold_2.pgm and output is [0.]
Predict correct : test-image is gestures/K/K_hold_3.pgm and output is [0.]
Predict correct : test-image is gestures/K/K_stop_1.pgm and output is [0.]
Predict correct : test-image is gestures/K/K_stop_2.pgm and output is [0.]
Predict correct : test-image is gestures/K/K_stop_1.pgm and output is [0.]
Predict correct : test-image is gestures/K/K_stop_2.pgm and output is [0.]
Accuracy: 85.54216867469879%
```

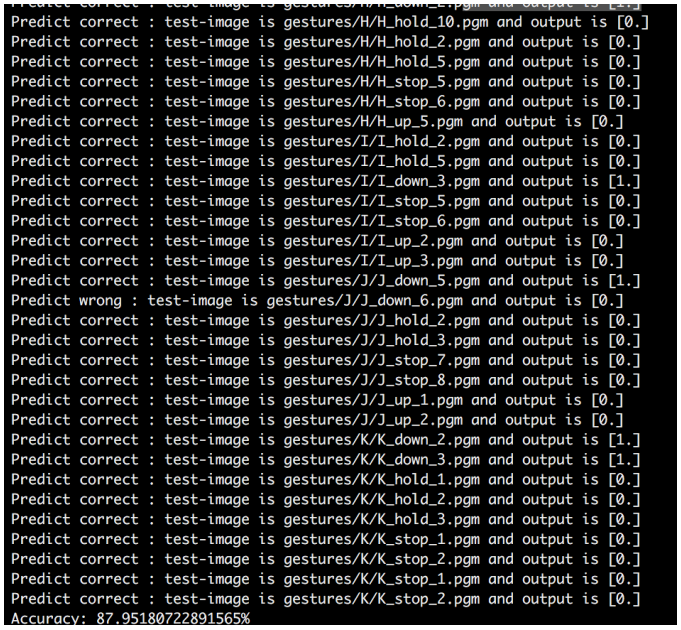
2. Accuracy: 81.93%

```
Predict correct : test-image is gestures/A/A_down_1.pgm and output is [1.]
Predict correct : test-image is gestures/A/A_down_2.pgm and output is [1.]
Predict correct : test-image is gestures/A/A_hold_1.pgm and output is [0.]
Predict wrong : test-image is gestures/A/A_hold_10.pgm and output is [1.]
Predict correct : test-image is gestures/A/A_stop_1.pgm and output is [0.]
Predict correct : test-image is gestures/A/A_stop_4.pgm and output is [0.]
Predict correct : test-image is gestures/A/A_up_1.pgm and output is [0.]
Predict correct : test-image is gestures/A/A_up_10.pgm and output is [0.]
Predict correct : test-image is gestures/B/B_down_1.pgm and output is [1.]
Predict correct : test-image is gestures/B/B_down_2.pgm and output is [1.]
Predict correct : test-image is gestures/B/B_hold_1.pgm and output is [0.]
Predict correct : test-image is gestures/B/B_hold_2.pgm and output is [0.]
Predict correct : test-image is gestures/B/B_stop_1.pgm and output is [0.]
Predict correct : test-image is gestures/B/B_stop_2.pgm and output is [0.]
Predict correct : test-image is gestures/B/B_up_1.pgm and output is [0.]
Predict correct : test-image is gestures/B/B_up_4.pgm and output is [0.]
Predict correct : test-image is gestures/C/C_down_1.pgm and output is [1.]
Predict correct : test-image is gestures/C/C_down_2.pgm and output is [1.]
Predict wrong : test-image is gestures/C/C_hold_1.pgm and output is [1.]
Predict wrong : test-image is gestures/C/C_hold_2.pgm and output is [1.]
Predict wrong : test-image is gestures/C/C_stop_2.pgm and output is [1.]
Predict correct : test-image is gestures/C/C_stop_3.pgm and output is [0.]
Predict wrong : test-image is gestures/C/C_up_1.pgm and output is [1.]
Predict wrong : test-image is gestures/D/D_down_1.pgm and output is [0.]
Predict wrong : test-image is gestures/D/D_down_2.pgm and output is [0.]
Predict correct : test-image is gestures/D/D_hold_1.pgm and output is [0.]
Predict correct : test-image is gestures/D/D_hold_2.pgm and output is [0.]
Predict correct : test-image is gestures/D/D_hold_6.pgm and output is [0.]
Predict correct : test-image is gestures/D/D_stop_1.pgm and output is [0.]
Predict correct : test-image is gestures/D/D_stop_2.pgm and output is [0.]
Predict correct : test-image is gestures/D/D_up_1.pgm and output is [0.]
Predict correct : test-image is gestures/D/D_up_3.pgm and output is [0.]
Predict wrong : test-image is gestures/E/E_down_1.pgm and output is [0.]
Predict wrong : test-image is gestures/E/E_hold_1.pgm and output is [1.]
Predict correct : test-image is gestures/E/E_hold_5.pgm and output is [0.]
Predict correct : test-image is gestures/E/E_stop_1.pgm and output is [0.]
Predict correct : test-image is gestures/E/E_stop_2.pgm and output is [0.]
Predict correct : test-image is gestures/E/E_up_1.pgm and output is [0.]
Predict correct : test-image is gestures/E/E_up_2.pgm and output is [0.]
Predict wrong : test-image is gestures/F/F_down_1.pgm and output is [0.]
Predict correct : test-image is gestures/F/F_down_4.pgm and output is [1.]
Predict correct : test-image is gestures/F/F_hold_1.pgm and output is [0.]
Predict wrong : test-image is gestures/F/F_hold_2.pgm and output is [1.]
Predict correct : test-image is gestures/F/F_stop_2.pgm and output is [0.]
Predict correct : test-image is gestures/F/F_stop_5.pgm and output is [0.]
Predict correct : test-image is gestures/G/G_down_2.pgm and output is [1.]
Predict wrong : test-image is gestures/G/G_down_3.pgm and output is [0.]
Predict correct : test-image is gestures/G/G_hold_4.pgm and output is [0.]
Predict correct : test-image is gestures/G/G_stop_2.pgm and output is [0.]
Predict correct : test-image is gestures/G/G_stop_5.pgm and output is [0.]
Predict wrong : test-image is gestures/G/G_up_2.pgm and output is [1.]
Predict correct : test-image is gestures/G/G_up_5.pgm and output is [0.]
```

```
Predict correct : test-image is gestures/G/G_up_5.pgm and output is [0.]
Predict wrong : test-image is gestures/H/H_down_2.pgm and output is [0.]
Predict correct : test-image is gestures/H/H_hold_10.pgm and output is [0.]
Predict correct : test-image is gestures/H/H_hold_2.pgm and output is [0.]
Predict correct : test-image is gestures/H/H_hold_5.pgm and output is [0.]
Predict correct : test-image is gestures/H/H_stop_5.pgm and output is [0.]
Predict correct : test-image is gestures/H/H_stop_6.pgm and output is [0.]
Predict correct : test-image is gestures/H/H_up_5.pgm and output is [0.]
Predict correct : test-image is gestures/I/I_hold_2.pgm and output is [0.]
Predict correct : test-image is gestures/I/I_hold_5.pgm and output is [0.]
Predict correct : test-image is gestures/I/I_down_3.pgm and output is [1.]
Predict correct : test-image is gestures/I/I_stop_5.pgm and output is [0.]
Predict correct : test-image is gestures/I/I_stop_6.pgm and output is [0.]
Predict correct : test-image is gestures/I/I_up_2.pgm and output is [0.]
Predict correct : test-image is gestures/I/I_up_3.pgm and output is [0.]
Predict correct : test-image is gestures/J/J_down_5.pgm and output is [1.]
Predict wrong : test-image is gestures/J/J_down_6.pgm and output is [0.]
Predict correct : test-image is gestures/J/J_hold_2.pgm and output is [0.]
Predict correct : test-image is gestures/J/J_hold_3.pgm and output is [0.]
Predict correct : test-image is gestures/J/J_stop_7.pgm and output is [0.]
Predict correct : test-image is gestures/J/J_stop_8.pgm and output is [0.]
Predict correct : test-image is gestures/J/J_up_1.pgm and output is [0.]
Predict correct : test-image is gestures/J/J_up_2.pgm and output is [0.]
Predict correct : test-image is gestures/K/K_down_2.pgm and output is [1.]
Predict correct : test-image is gestures/K/K_down_3.pgm and output is [1.]
Predict correct : test-image is gestures/K/K_hold_1.pgm and output is [0.]
Predict correct : test-image is gestures/K/K_hold_2.pgm and output is [0.]
Predict correct : test-image is gestures/K/K_hold_3.pgm and output is [0.]
Predict correct : test-image is gestures/K/K_stop_1.pgm and output is [0.]
Predict correct : test-image is gestures/K/K_stop_2.pgm and output is [0.]
Predict correct : test-image is gestures/K/K_stop_1.pgm and output is [0.]
Predict correct : test-image is gestures/K/K_stop_2.pgm and output is [0.]
Accuracy: 81.92771084337349%
```

3. Accuracy: 87.95%

```
Predict correct : test-image is gestures/A/A_down_1.pgm and output is [1.]
Predict wrong : test-image is gestures/A/A_down_2.pgm and output is [0.]
Predict correct : test-image is gestures/A/A_hold_1.pgm and output is [0.]
Predict wrong : test-image is gestures/A/A_hold_10.pgm and output is [1.]
Predict correct : test-image is gestures/A/A_stop_1.pgm and output is [0.]
Predict correct : test-image is gestures/A/A_stop_4.pgm and output is [0.]
Predict correct : test-image is gestures/A/A_up_1.pgm and output is [0.]
Predict correct : test-image is gestures/A/A_up_10.pgm and output is [0.]
Predict correct : test-image is gestures/B/B_down_1.pgm and output is [1.]
Predict wrong : test-image is gestures/B/B_down_2.pgm and output is [0.]
Predict correct : test-image is gestures/B/B_hold_1.pgm and output is [0.]
Predict correct : test-image is gestures/B/B_hold_2.pgm and output is [0.]
Predict correct : test-image is gestures/B/B_stop_1.pgm and output is [0.]
Predict correct : test-image is gestures/B/B_stop_2.pgm and output is [0.]
Predict correct : test-image is gestures/B/B_up_1.pgm and output is [0.]
Predict correct : test-image is gestures/B/B_up_4.pgm and output is [0.]
Predict correct : test-image is gestures/C/C_down_1.pgm and output is [1.]
Predict correct : test-image is gestures/C/C_down_2.pgm and output is [1.]
Predict correct : test-image is gestures/C/C_hold_1.pgm and output is [0.]
Predict correct : test-image is gestures/C/C_hold_2.pgm and output is [0.]
Predict correct : test-image is gestures/C/C_stop_2.pgm and output is [0.]
Predict correct : test-image is gestures/C/C_stop_3.pgm and output is [0.]
Predict correct : test-image is gestures/C/C_up_1.pgm and output is [0.]
Predict wrong : test-image is gestures/D/D_down_1.pgm and output is [0.]
Predict wrong : test-image is gestures/D/D_down_2.pgm and output is [0.]
Predict correct : test-image is gestures/D/D_hold_1.pgm and output is [0.]
Predict correct : test-image is gestures/D/D_hold_2.pgm and output is [0.]
Predict correct : test-image is gestures/D/D_hold_6.pgm and output is [0.]
Predict correct : test-image is gestures/D/D_stop_1.pgm and output is [0.]
Predict correct : test-image is gestures/D/D_stop_2.pgm and output is [0.]
Predict correct : test-image is gestures/D/D_up_1.pgm and output is [0.]
Predict correct : test-image is gestures/D/D_up_3.pgm and output is [0.]
Predict wrong : test-image is gestures/E/E_down_1.pgm and output is [0.]
Predict correct : test-image is gestures/E/E_hold_1.pgm and output is [0.]
Predict correct : test-image is gestures/E/E_hold_5.pgm and output is [0.]
Predict correct : test-image is gestures/E/E_stop_1.pgm and output is [0.]
Predict correct : test-image is gestures/E/E_stop_2.pgm and output is [0.]
Predict correct : test-image is gestures/E/E_up_1.pgm and output is [0.]
Predict correct : test-image is gestures/E/E_up_2.pgm and output is [0.]
Predict wrong : test-image is gestures/F/F_down_1.pgm and output is [0.]
Predict wrong : test-image is gestures/F/F_down_4.pgm and output is [0.]
Predict correct : test-image is gestures/F/F_hold_1.pgm and output is [0.]
Predict correct : test-image is gestures/F/F_hold_2.pgm and output is [0.]
Predict correct : test-image is gestures/F/F_stop_2.pgm and output is [0.]
Predict correct : test-image is gestures/F/F_stop_5.pgm and output is [0.]
Predict correct : test-image is gestures/G/G_down_2.pgm and output is [1.]
Predict wrong : test-image is gestures/G/G_down_3.pgm and output is [0.]
Predict correct : test-image is gestures/G/G_hold_4.pgm and output is [0.]
Predict correct : test-image is gestures/G/G_stop_2.pgm and output is [0.]
Predict correct : test-image is gestures/G/G_stop_5.pgm and output is [0.]
Predict correct : test-image is gestures/G/G_up_2.pgm and output is [0.]
Predict correct : test-image is gestures/G/G_up_5.pgm and output is [0.]
Predict correct : test-image is gestures/H/H_down_2.pgm and output is [1.]
```



4. Challenge

The pgm format is easy for coding but because of the large amount of input data and complex mathematics formulas, it takes us some time to think about what data structure to use. For instance, in the back propagation part, we need to convert the output of pixels of forward propagation and the error partial derivative results into 2D arrays for further multiplications, so that it becomes easier to read and to do further steps.

Also, the concepts for forward propagation and back propagation are not difficult, however, the mathematics formulas include lots of mathematics symbols and partial derivatives, and a neuron output might be affected by multiple neurons. Thus, it takes us lots of time to understand these formulas and especially converting them into coding.

5. Optimization

To select random weights in the forward propagation step, we tried many different approaches, we found that weights ranging from -0.01 to 1 if better than weights ranging from -0.01 to 0.01 (which is stated in HW5 description) Thus, we use the following approach to calculate our random weights:

```
w1 = -0.01, w2 = 1
```

```
((w2) - (w1)) * np.random.normal(size = (network_size[l - 1],network_size[l])) + (w1)
```

In our implementation, we use gradient descent method to optimize the weights. However, from the result of scikit learn implementation, the optimizer based on quasi-Newton methods performed better. Therefore, we can improve our code by using quasi-Newton methods such as DFP(Davidon-Fletcher-Powell) algorithm or

BFGS(Broyden-Fletcher-Goldfard-Shano) algorithm to update weights. Also, we can try implement the update of learning rate in each iteration to be invscaling or adaptive.

6. Software Familiarization

sklearn.neural_network.MLPClassifier:

This library saves people lots of time of struggling converting the data structures of those mathematical, we simply store our training data and training label as list and call the fit() method. There are lots of attributes which we can adjust conveniently when we initialized the MLPClassifier. In addition to **hidden_layer_sizes** and **max_iter**, some more special attributes in the library are as below.

activation:

There are four kinds of activation function you can choose: (identity, logistic, tanh, relu), default=relu. Here we use logistic, that is, sigmoid function 1 / (1 + exp(-x)).

solver:

The solver for weight optimization. The default is 'adam', but we tried different solver 'sgd' and 'lbfgs', the result of accuracy are very different. By 'lbfgs' solver, the accuracy is the highest, which is around 93% while by the 'adm' solver the accuracy is the lowest, which is around 77%.

learning_rate:

The learning rates for weight updates. We can choose: (constant, invscaling, adaptive) default=constant. Here we use constant, that is, our learning rate is always the same as the attribute learning_rate_init in each iteration.

learning_rate_init:

The initial learning rate used. It controls the step-size in updating the weights. In our case, it is 0.1.

Result under different solver:

-Lbfgs solver: an optimizer in the family of quasi-Newton methods.

```
Accuracy: 93.975903614457 83%
```

-Sgd solver: refers to stochastic gradient descent.

```
Accuracy: 89.15662650602 41%
```

-Adam solver: refers to a stochastic gradient-based optimizer.

```
Accuracy: 77.108433734939 77%
```

7. Applications

1. Speech Reconstruction:

   Converting sound waves into 2D arrays based on the sound wave amplitudes at millisecond to feed in the Neural Networks.

2. Forecasting Profits:

   Using the previous stock price data to find the relationships and predict the future stock market.

3. Image Recognition:

   Converting the pixels of images into 2D array and feed them into the Neural Networks.

4. Image Reconstruction:

   Calculating linear approximation for images to do the training.