Xiaotong Sun XSEDE Username: xs018 Nov 04, 2021

HW 4
Machine Learning
PCA-MLP

HW 4 Problem Statement (provided by Dr. Han Hu, MEEG 491V/591V-028, Fall 2021)
Problem 4-1 Classification using PCA-MLP:
HW-4 uses the same data set as Problem 1-2 of HW-1.

Re-do Problem 1-2 using PCA-MLP. Run SVD or PCA to obtain the PCs of the images. Feed the PCs to an MLP neural network to classify the regime of the boiling images.

Code Running Instructions
The environment is default only with scikit-image download.

Coding explanation
(Data loading) The very first step is to load all the images data along with their corresponding labels. The way to do that is to first store the data path and label of each image into a csv file. Then I looped over each image paths to read the image matrix and their corresponding label. Subsequently, we fed all the images into a single matrix with each row is the flattened pixel intensities of a single image. Finally, the data matrix is standardized by its mean and standard deviation and the principal components (100 principal components I was using) were extracted and stored in a 2-dimensional matrix. While the labels were stored in a separate matrix. The reason why we stored the matrix is that loading the images one-by-one and compute the principal components are I/O-computation intense procedures, which should be avoided whenever we fed the principal components into the Multiple Linear Perceptron. Once the input data was stored in matrices, only the computation over MLP was involved, which is well-paralleled by GPU (~1s/epoch). Also note that the dimension of the matrix we stored equals to the NxC, where N is the number of images and C is the number of principal components we are using.

(MLP architecture) The rest of the step is straight-forward which is similar to what we did in homework 1. The code for MLP is adapted from homework but the number of layers and the number of neurons each layer were a bit different. In this assignment, I used four layers with 64, 32, 16, 2 neurons, respectively. Also note that the whole dataset was spited into training (72%), validation (18%) and testing set (10%).

(evaluation) The result for the classification was evaluated mainly by the confusion matrix, ROC curve, all of which were adapted from homework 1.

Results

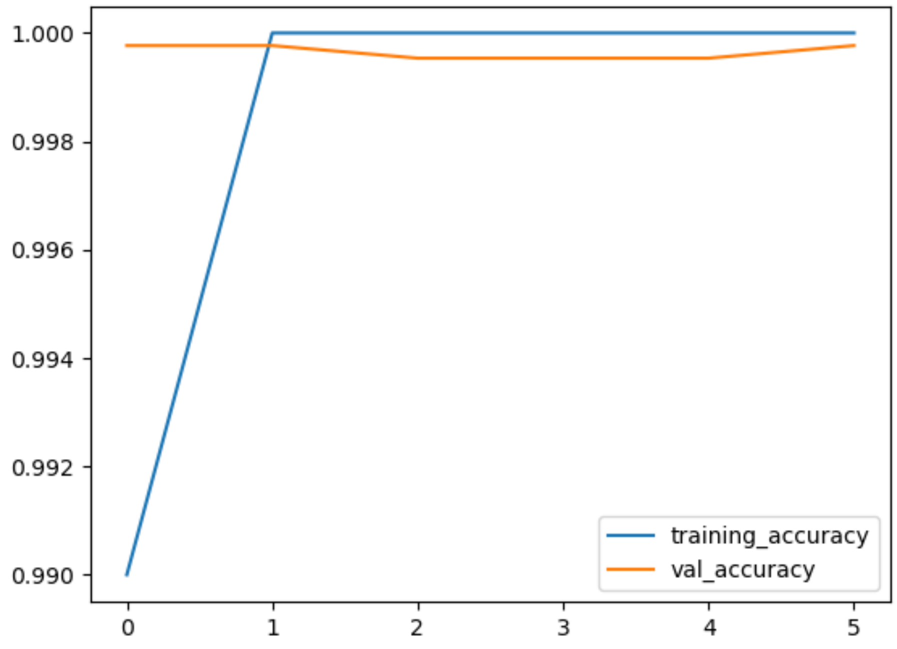

Fig 1. Training accuracy and validation accuracy for 50 epochs.

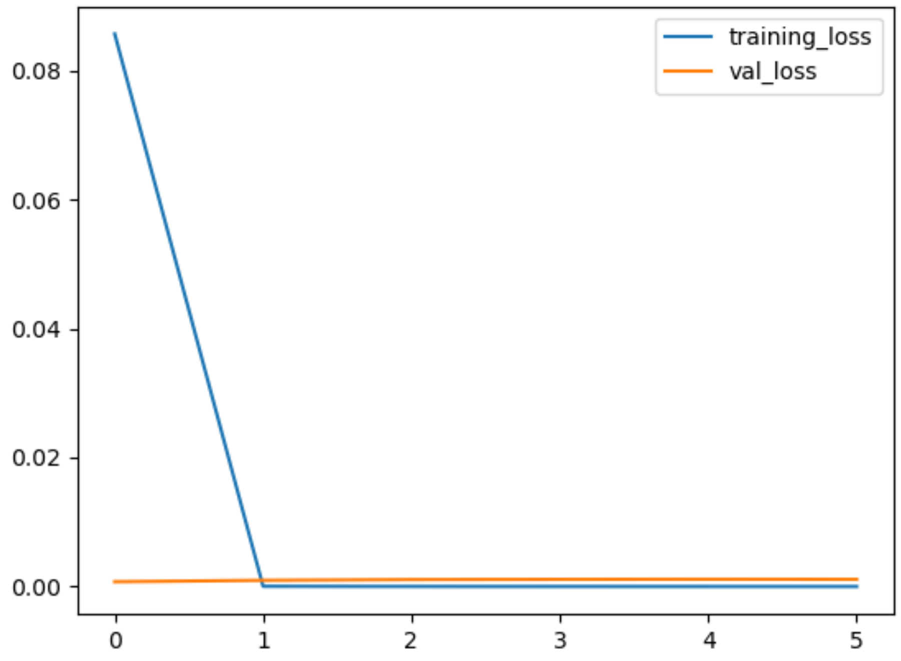

Fig 2. Training loss and validation loss

Figure 1 is the graph for the training and validation accuracy for 50 epochs. Figure 2 is the graph of training and validation loss. Since the early stopping criteria is used, it stopped at epoch 6. The result of each epoch is shown as follows:

```
Train on 17200 samples, validate on 4301 samples

Epoch 1/50

17200/17200 - 2s - loss: 0.0857 - accuracy: 0.9900 - val_loss: 7.4459e-04 - val_accuracy: 0.9998

Epoch 2/50

17200/17200 - 1s - loss: 4.0743e-05 - accuracy: 1.0000 - val_loss: 9.6638e-04 - val_accuracy: 0.9998

Epoch 3/50

17200/17200 - 1s - loss: 1.5345e-05 - accuracy: 1.0000 - val_loss: 0.0011 - val_accuracy: 0.9995

Epoch 4/50

17200/17200 - 1s - loss: 9.0905e-06 - accuracy: 1.0000 - val_loss: 0.0011 - val_accuracy: 0.9995

Epoch 5/50

17200/17200 - 1s - loss: 6.0865e-06 - accuracy: 1.0000 - val_loss: 0.0011 - val_accuracy: 0.9995

Epoch 6/50

Restoring model weights from the end of the best epoch.

17200/17200 - 1s - loss: 4.3450e-06 - accuracy: 1.0000 - val_loss: 0.0011 - val_accuracy: 0.9998

Epoch 00006: early stopping.
```

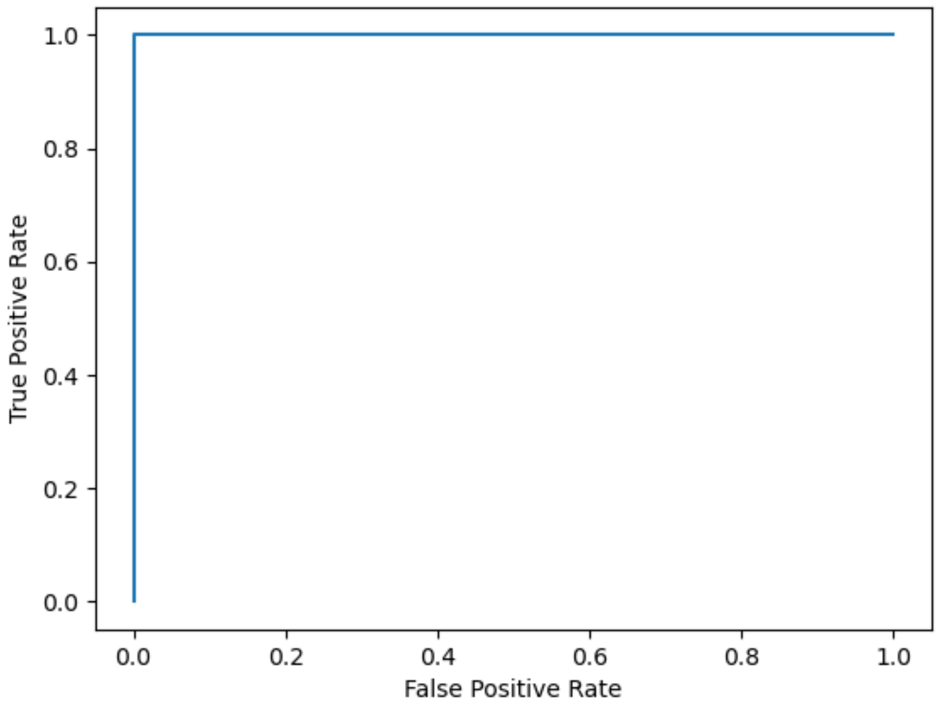

Fig 3. Ture and false positive rate

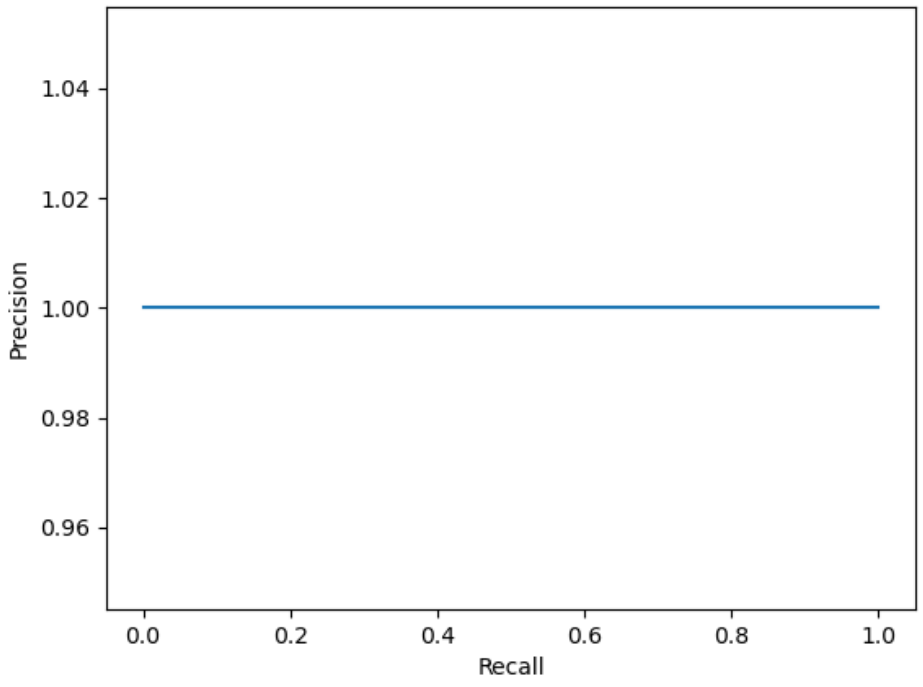

Fig 4. Precision of the classification

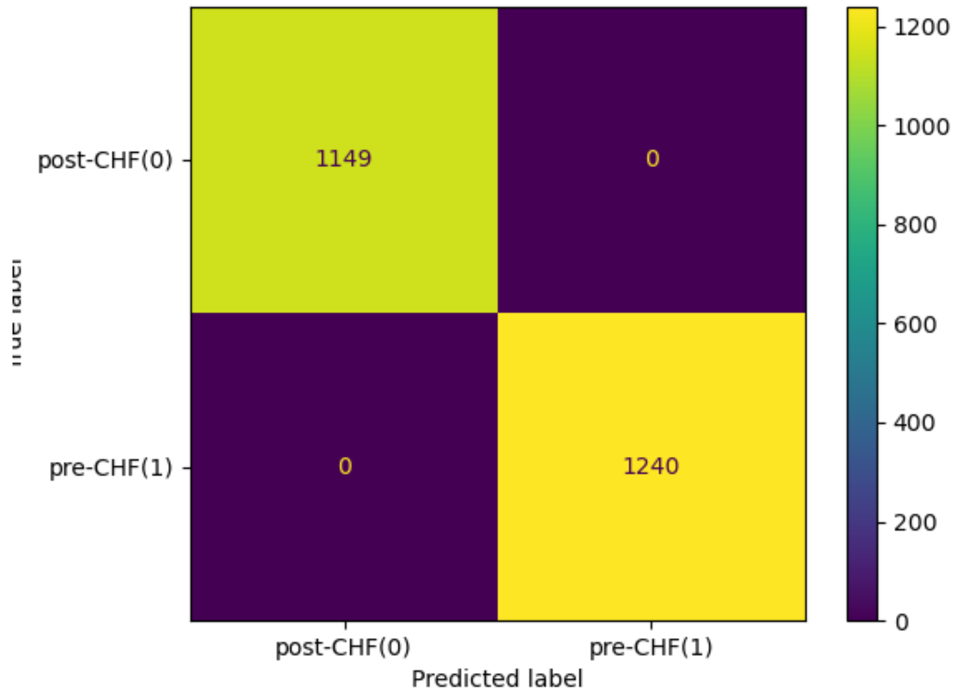

Fig 5. Confusion matrix graph

The result is as follows:

True Negative: 1149, False Positive: 0, False Negative: 0, True Postive: 1240

Area Under Curve: 1.0

Accuracy: 1.0

Precision: 1.0

Recall: 1.0

F1 Score: 1.0

Challenges

The main challenge is that when I feed the MLP with the result of PCA, there is some errors which costs me lots of time to figure it out. For example, when the number of PCs is smaller than the number of my sample size (I test the code with small number of images), it said the number of PCs should between 0 to min sample size. It takes me some time to find it out, that is because when the number of PCs bigger than the sample size, there will be some problems for the matrix calculation. The other challenge I met is that I usually use VS code to write and test my code remotely, but it didn't allow me to login from Nov 3. I tested in different laptop, it still didn't work. I feel it could be the problem of the connection between server and VS code. So I changed to FileZilla which I used that in last semester, but it is not as convenient as VS code because I need to use terminal and download the result to my laptop.

References

[1] Facial Recognition using PCA and MLP in Python - PDF.co.

[2] https://stackoverflow.com/questions/67957035/pca-for-face-recognition-in-python