|  |
| --- |
| **Machine Learning Report of Pro2** |

**Haorui Dong**

**UBIT number: 50291149**

Department of Computer Science and Engineering

University at Buffalo

[*haoruido@buffalo.edu*](mailto:haoruido@buffalo.edu)

**Abstract**

In project2, we get 6 .csv files. We use them to do a linear regression to find similarity between the handwritten samples of the known and the questioned writer. It’s the first time that I use a lot of pandas.DataFrame. So I do lots of works with these 6 csv files. After that I modify code of linear regression in project1.2 to train my data.

1. Process the origin data.
   1. introduce the pandas and DataFrame

When I want to read data from csv files, I found that there is a method in pandas called pandas.read\_csv(file).

This function could read the data from csv files and make a DataFrame which is very flexible and easy to use in data processing. When we print a DataFrame, it looks like an excel form.

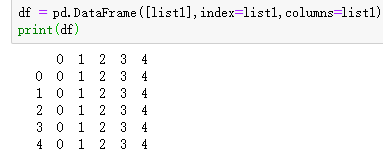


Figure1:initial a dataframe

It has a raw index and columns name. We can use df.loc or df.iloc to get each element we want. Or we can easily change the DataFrame to numpy.ndarray by df.values.

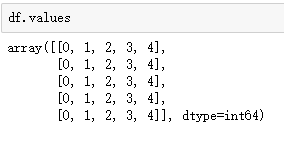


Figure2:change type from dataframe to np.array

Dataframe.iloc(i,j) can get the element which index is i and columns is j.

* 1. processing the csv data

I use read\_csv to get 6 dataframes.

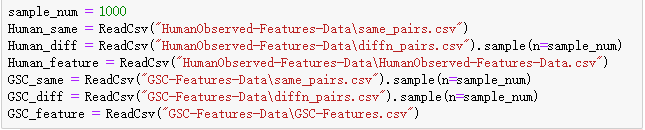


Figure3: 6 dataframes of origin data

Dataframe.sample could do sampling. N is sample number. That’s really convenient.

We will use Human\_same and Human\_diff pairs img\_id to get feature of each id. And make them do concatenation and subtraction. Then we let GSC\_same and GSC\_diff do the same process. And we set target that same has 1 target and diff has 0 target. Then we can get 4 dataframe. They are:

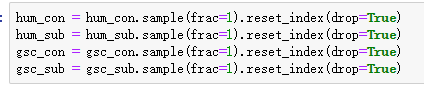


Figure4: sampling

We use df.sample to mess up the order. Frac = 1 means do not modify the size of matrix. And reset\_index will make the new index instead of old index.

Then we use slice to get test validation and train matrix. And we get the target array of each matrix. After this we are ready to do linear regression.

1. Linear regression

First, we define many variables at beginning. It will make some changes more convenient.

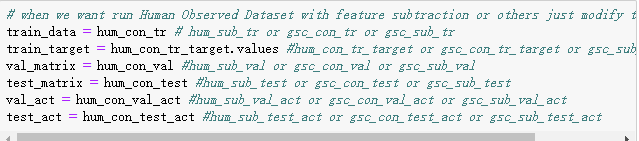


Figure5: modify variables to different groups of training

When we want to change the matrix to do another train. Just change these variables and be sure that change the same group of variables.

Then I use the code in project1.2, and there are lots of problems if we do not modify some code.

I do some transpose of my matrix. And finally, I got my close form result and the gradient descent for linear regression.

1. result
   1. human concatenation matrix

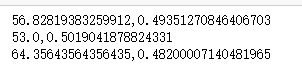


Figure6: close form result of human concatenation

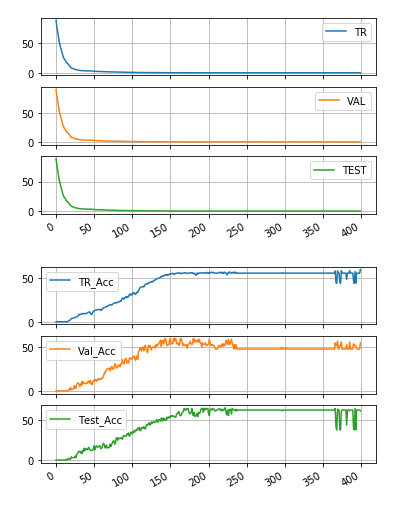


Figure7: gradient descent result

* 1. human subtraction matrix

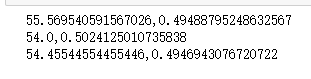


Figure8: close form result of human concatenation

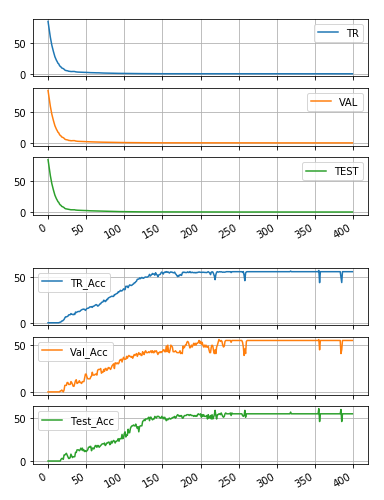


Figure9: gradient descent result

* 1. gsc concatenation matrix

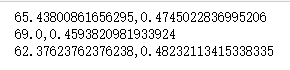


Figure10: close form result of human concatenation

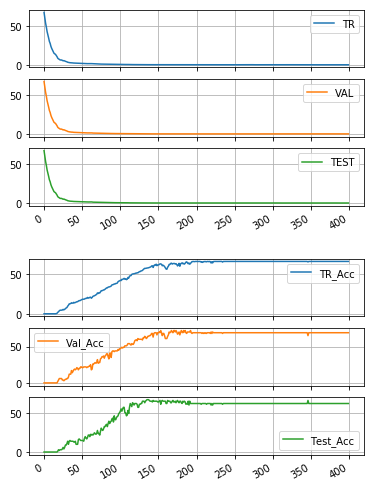


Figure11: close form result of human concatenation

* 1. gsc subtraction matrix

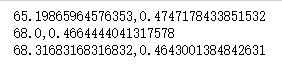
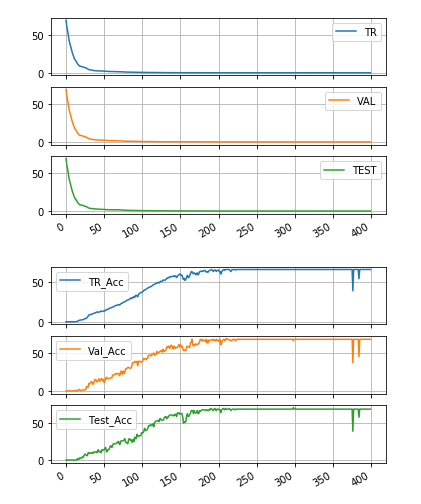


Figure12: close form result of human concatenation



Figur13: close form result of human concatenation