

README

notes

- environment:
 - python 3.6.8
 - numpy 1.14.7
 - tensorflow 2.0.0
 - keras 2.3.1
 - matplotlib 3.1.2
 - graphviz 2.42.2 (need to install from official website;graphviz is used to plot the model structure)
 - pydot 1.4.1
- There are 9 programs in the folder 'codes'.
- If you want to test you own data in my program, please change the variable '**filename**' in the program you want to use and put the data file in the folder 'codes'.
the variable '**filename**' is at the top of each program.
- If you have any questions, please contact me in canvas before you grade me. Thanks.
- In **promble 2 and 3**, you can input data with all labels, but my program will **only train and test the data with label 1 and 5**.

programs:

other program

- [DataPreprocess.py](#)
 - This program is essential, which any other programs depened on
 - This program encluds method :LoadData(fileName), DivideData(data), GetOneFive(fileName), which are used in other programs.
 - This program is not for run. It won't get you any output.

promble1

- [Problem1-train.py](#)

- Input: This problem get the train data which has two features and one label.(You can change the filename to load other data, which should have the same dimensions of the data in 'features.train.txt')
- Output: 11 figures.
 - 1-10. Draw all the data in the picture, use two colors to divided them.
 - 11. Draw all the data with different colors accroding to their labels.
- The output pictures are shown in the report.

problem2

- [Problem2-train.py](#)
 - Input: 3 values data:(label,intensity,symmetry), like 'feature.train.txt'
 - Output:
 - 3 different models' in sample loss, test set loss,in sample accuracy and test set loss in 3-fold cross-validation.
 - 3 model information files named 'P2-1.h5','P2-2.h5','P2-3.h5', which contain the information of those three models after training with all the train data.
 - This program build three different models with 3,6,10 units in hidden layer. Use 3-fold cross-validation to compare them. Use all the train data to train those models and save the models for later use.
- [Problem2-test.py](#)
 - Input: The data you want to test in my models, it should be 3 values data: (label,intensity,symmetry), like 'feature.test.txt'
 - Output : loss and accuracy in each model.
 - This program build the models same with the train program.
And load the models' weights I trained in the train program to predict the label of the test data and output the loss and accuray.

Problem3

- [Problem3-1-train.py](#)
 - Input: 257 values for one data:(label, 256 raw features),like file 'zip.train.txt'
 - Output:
 - 2 different models' in sample loss, test set loss,in sample accuracy and test set loss in 3-fold cross-validation.
 - 2 model information files named 'P3-1.h5','P3-2.h5', which contain the information of those three models after training with all the train data.
 - This program build two different models with structure: [256, 6, 2, 1], [256, 3, 2, 1]. Use 3-fold cross-validation to compare them. Use all the train data to train those models and save

the models for later use.

- [Problem3-2-train.py](#)
 - Input: both train data and test data with raw features
 - Output: a picture show the change of the in-sample error and test-set error for each iteration.
 - The picture is shown in the report.
- [Problem3-test.py](#)
 - Input: The data you want to test in my models, it should be 257 values data:(label,256 raw features), like 'zip.test.txt'
 - Output: loss and accuracy in each model.
 - This program build the models same with the train program.
And load the models' weights I trained in the train program to predict the label of the test data and output the loss and accuray.

Problem4

- [Problem4-train.py](#)
 - Input: 257 values for one data:(label, 256 raw features),like file 'zip.train.txt'
 - Output: loss and accuracy in each fold.
 - This program build a model to classify
- [Problem4-test.py](#)
 - Input: 257 values for one data:(label, 256 raw features),like file 'zip.test.txt'
 - Output: loss and accuracy
 - This program build the model same with the train program.
And load the models' weights I trained in the train program to predict the label of the test data and output the loss and accuray.