2018/2019 COMP1037 Coursework 2 – Machine Learning

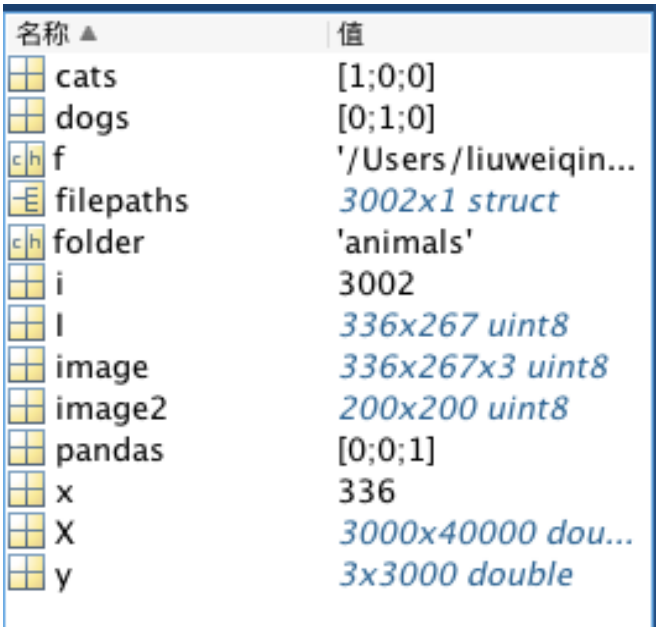
FAI Coursework 2 – Neural Network Classification

Reports

a. Show all your dataset structure, network configuration with figures or tables.

Answer:
The following picture is the work place with dataset after executing the file **"build_animals.m".**

- The **"cats"**, **"dogs"** and **"pandas"** are matrices for classification.
- **"f"** is the variable containing the file path.
- **"filepaths"** contains all paths of three thousand pictures.
- **"image"** is the numeric matrix of a RGB image.
- **"I"** is the gray picture of the former RGB image.
- **"image2"** is the adjusted picture of the former gray picture with size of 200x200.
- **"X"** contains matrices being reshaped of all pictures.
- **"y"** contains the information of classification.



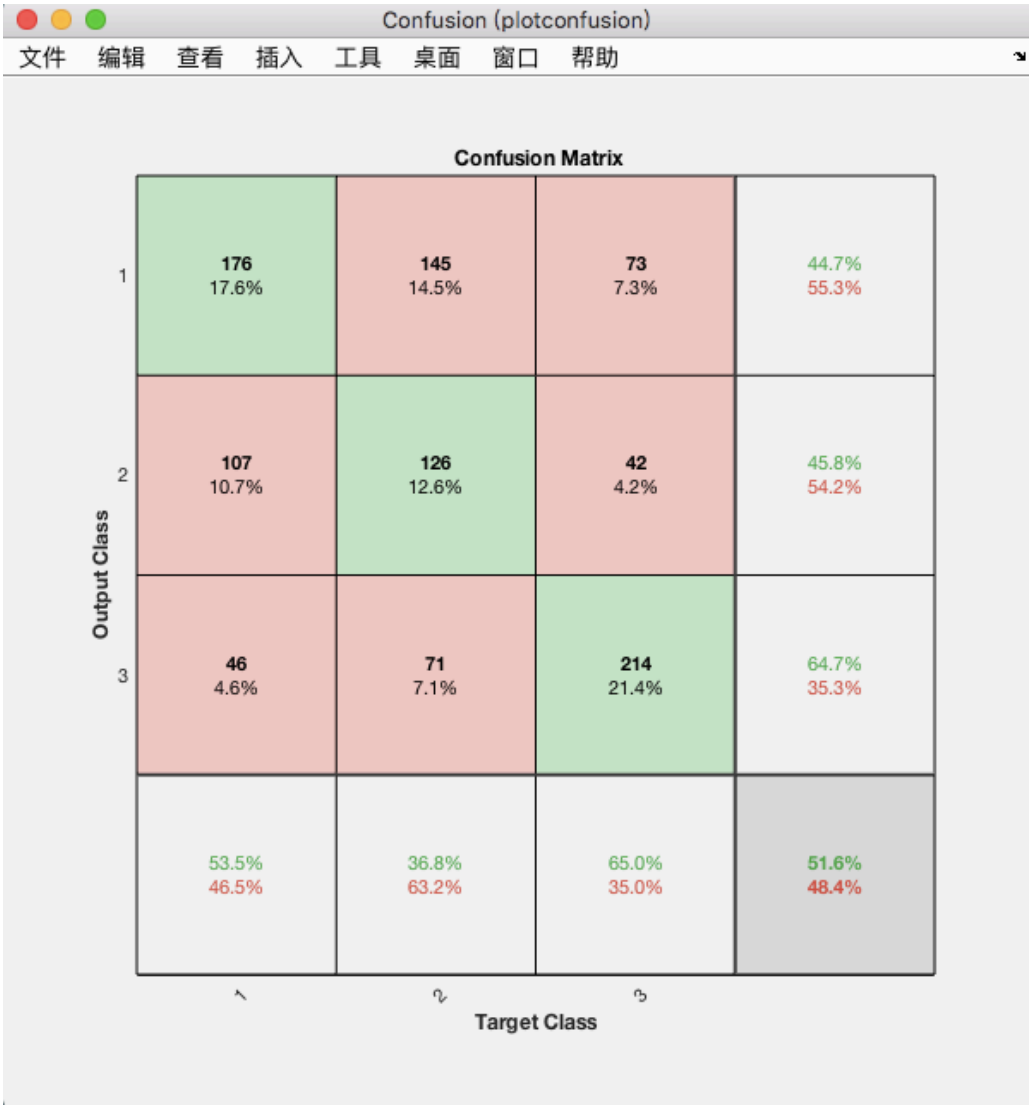
The following chart is the configuration of my neural network and parameters I have tried.

Function	Patternnet
The number of hidden layers	15/25/30/40/45/50/55/100
Train Ratio/Test Ratio	0.9/0.1 0.8/0.2 0.7/0.3 0.6/0.4 0.5/0.5
The number of Train data/Test data	2000/1000 1500/1500 1000/100

b. Include your best (or worth explaining or both) testing accuracy rate with confusion matrix. Explain the meaning of your confusion matrix (what is the row and what is the column). From the confusion matrix, evaluate the performance of your trained NN model.

Answer:
The following confusion matrix shows my best testing accuracy which is 51.6%.

2018/2019 COMP1037 Coursework 2 – Machine Learning



The meaning of the confusion matrix is that: The confusion matrix is used as a description of precision. Each row of the matrix represents the instances in a predicted class while each column represents the instances in an actual class. For the above confusion matrix, 176 pictures of class 1 which are cats are successfully predicted by neural network. For dogs and pandas, there are 126 and 214 pictures of them successfully predicted by neural network. The sum of each column is the total number of instances in an actual class. 329 cat pictures, 342 dog pictures and 329 panda pictures which totally add up to 1000 pictures are used as input. Each row shows the number of pictures which are successfully or unsuccessfully predicted. For example, the row 1 shows that 145 dog pictures and 73 panda pictures are wrongly predicted as cats. Moreover, the number percent in the fringe means the correct present of prediction.

c. Show all the different parameter settings you had try and their differences. Make a short conclusion or summary about it. Present your results in tabular format or figures if applicable.

Answer:
I have adjusted parameters, such as the number of hidden layers and training rate with testing rate. The following chart shows different data with different numbers of hidden layer when the training ratio and testing ratio are 0.8 and 0.2 respectively. 2000 pictures are used as training data and 1000 pictures are used as testing data.

The number of hidden layers	Accuracy%	Time cost	The number of iterations
15	48	7:53	358
25	49.4	9:35	181
30	48.3	8:20	146
40	51.4	11:30	121
45	49.9	8:35	112
50	46.9	9:42	126
55	50.4	10:28	109
100	51.2	10:30	116

I also change the learning rate and the following chart includes different data with different train/test ratio when the number of hidden layers is equal to 50. It can be concluded from the chart that the time cost gradually decreases with the decrease of training ratio. The accuracy seems to have a gradual fluctuation. Being similar to the accuracy, the number of iterations totally decrease a bit with a small fluctuation.

Training/Testing ratio	Accuracy%	Time cost	The number of iterations
0.9/0.1	50.8	12:22	134
0.8/0.2	46.9	9:42	126
0.7/0.3	49.9	8:58	109
0.6/0.4	47.9	8:51	117
0.5/0.5	50.5	7:50	112

Efficiency

In order to improve the efficiency of the neural network, PCA (Principle Component Analysis) could be used.

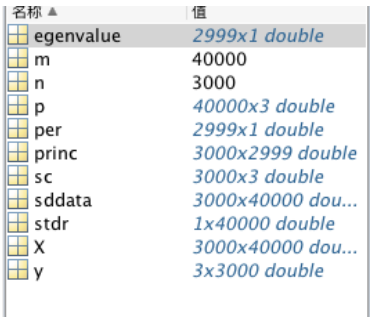
PCA could be used to achieve dimension reduction by grouping large number of data as different component. Data with similar features are contained in the same components which can decrease the input of neural network and increase the efficiency with less loss of precision.

The following codes could be used to achieve PCA in MATLAB and I have put the code in a single file called “PCA.m” The following picture is the screenshot of the code.

2018/2019 COMP1037 Coursework 2 – Machine Learning

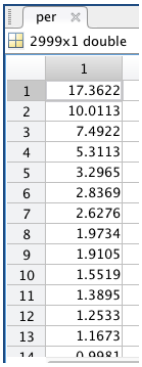
```
load('data.mat');
stdr = std(X);
[n,m]= size(X);
sddata = X./stdr(ones(m,1));
[p,princ,egenvalue]=pca(sddata);
p=p(:,1:3);
sc=princ(:,1:3);
egenvalue;
per=100*egenvalue/sum(egenvalue);
```

The following picture shows the workplace after executing the “PCA.m”



名称 ▲	值
egenvalue	2999x1 double
m	40000
n	3000
p	40000x3 double
per	2999x1 double
princ	3000x2999 double
sc	3000x3 double
sddata	3000x40000 dou...
stdr	1x40000 double
X	3000x40000 dou...
y	3x3000 double

“Per” shows different components of inputs.



per

2999x1 double

	1
1	17.3622
2	10.0113
3	7.4922
4	5.3113
5	3.2965
6	2.8369
7	2.6276
8	1.9734
9	1.9105
10	1.5519
11	1.3895
12	1.2533
13	1.1673
14	0.9981

Moreover, I did data normalization by using the function “mapminmax” and saved the handled data into another file called “data2.mat”. After doing normalization and PCA, the accuracy goes up a little bit. (from 48.3 to 49.0)

0.5769	0.5865	0.6010	0.6106	0.6010	0.5865	0.5865	0.6010	0.6010	0.5913	0.6010
0.5459	0.5240	0.4803	0.4017	0.2489	0.2664	0.2664	0.2533	0.2707	0.2445	0.2489
0.7052	0.6892	0.6494	0.6295	0.6494	0.6773	0.6932	0.6972	0.6972	0.6853	0.6574
0.2077	0.1111	0.0773	0.0918	0.0580	0.0725	0.0918	0.1014	0.0966	0.0725	0.0773
0.3263	0.2737	0.2474	0.1895	0.2053	0.1737	0.1105	0.1263	0.0684	0.0684	0.0579
0.0635	0.0635	0.0635	0.0635	0.0635	0.0556	0.0556	0.0556	0.0556	0.0516	0.0437
0.0202	0.0202	0.0202	0.0162	0.0162	0.0162	0.0202	0.0202	0.0202	0.0202	0.0202
0.2395	0.2335	0.2515	0.2335	0.2335	0.2216	0.2036	0.1916	0.2036	0.2275	0.2335
0.4048	0.3968	0.3929	0.3770	0.3889	0.3889	0.3651	0.3810	0.3611	0.3651	0.3770
0.2618	0.3047	0.3262	0.3133	0.2918	0.3047	0.3176	0.3004	0.3047	0.3262	0.3133
0.6374	0.6319	0.6209	0.6154	0.6154	0.6154	0.6209	0.6264	0.6264	0.6209	0.6099
0.8889	0.8929	0.8929	0.8849	0.8770	0.8690	0.8571	0.8373	0.8532	0.8532	0.8413
0.1857	0.1714	0.1857	0.1857	0.1714	0.1810	0.1952	0.1905	0.1810	0.1857	0.1952

2018/2019 COMP1037 Coursework 2 – Machine Learning