2018/2019 COMP1037 Coursework 2 – Machine Learning

FAI Coursework 2 – Neural Network Classification

1. Reports [3 marks]

Dataset structure:

Following code are the one used to create the dataset. As the original pixel being used to standardize the size of the picture is 50 * 50, the dataset X is 2500 * 3000, where 2500 stands for 50 * 50 image size and 3000 represent the total number of sample image.

```
%set a dataset X to put all data
X = [];
for i = 1 : length(list)
    image = imread(list(i).name);
    imgray = rgb2gray(image); %convert RGB images into grayscale images
    imnew = imresize(imgray,[50 50]); %resize the image to 50 * 50
    newmap = reshape(imnew, 1, []); %reshape the array from 2D to 1D
    normimage = im2double(newmap); % Normalize the image value.
    newmap = normimage';
    X(:, i) = newmap';
end
```

The statistics in dataset will like this:

2500x3000 double

	1	2	3	4	5	6	7	8	9	10
1	0.4863	0.3882	0.5922	0.1137	0.2157	0.0627	0.0510	0.2392	0.4039	0.41
2	0.4902	0.3294	0.6078	0.0745	0.1216	0.0588	0.0510	0.2196	0.3922	0.41
3	0.4902	0.3686	0.6314	0.0745	0.0588	0.0510	0.0510	0.2275	0.3843	0.44
4	0.5059	0.4039	0.5725	0.0706	0.0510	0.0510	0.0510	0.2392	0.4118	0.46
5	0.5098	0.4118	0.5255	0.0745	0.0549	0.0471	0.0549	0.2353	0.4275	0.39
6	0.5059	0.3765	0.5294	0.0784	0.0549	0.0431	0.0510	0.2353	0.4314	0.40
7	0.4980	0.5412	0.5804	0.0784	0.0588	0.0392	0.0549	0.2392	0.4235	0.39
8	0.4941	0.5765	0.6118	0.0745	0.0627	0.0392	0.0549	0.2353	0.4431	0.49
9	0.4784	0.5725	0.5098	0.0745	0.0588	0.0314	0.0549	0.2392	0.4471	0.38
10	0.4549	0.5765	0.5294	0.0706	0.0588	0.0275	0.0549	0.2392	0.4392	0.37
11	0.4353	0.5922	0.5255	0.0706	0.0588	0.0275	0.0431	0.2275	0.4667	0.33
12	0.4235	0.6000	0.5294	0.0706	0.0627	0.0275	0.0549	0.2039	0.5608	0.42
13	0.4118	0.4902	0.4902	0.0863	0.0667	0.0275	0.0549	0.1961	0.5843	0.37
14	0.4078	0.5216	0.5490	0.0863	0.0627	0.0314	0.0549	0.2000	0.6588	0.34
15	0.4314	0.3059	0.5020	0.0863	0.0510	0.0275	0.0549	0.1961	0.8157	0.38
16	0.4392	0.2275	0.5020	0.0824	0.0706	0.0235	0.0549	0.1961	0.6745	0.41
17	0.4549	0.2314	0.5569	0.0902	0.0784	0.0275	0.0588	0.2000	0.7882	0.43
18	0.4627	0.2275	0.8235	0.0902	0.0784	0.0275	0.0549	0.2000	0.8510	0.32
19	0.4745	0.2314	0.9333	0.0902	0.0902	0.0314	0.0549	0.1961	0.8667	0.26
20	0.4667	0.2314	0.9490	0.0980	0.0941	0.0314	0.0549	0.2000	0.8627	0.25
21	0.4549	0.2471	0.8980	0.1020	0.0941	0.0392	0.0549	0.2000	0.8706	0.39
22	0.4431	0.2314	0.5020	0.1020	0.1020	0.0392	0.0588	0.2039	0.8549	0.21
23	0.4275	0.2235	0.3333	0.1020	0.0824	0.0510	0.0588	0.2000	0.8588	0.40

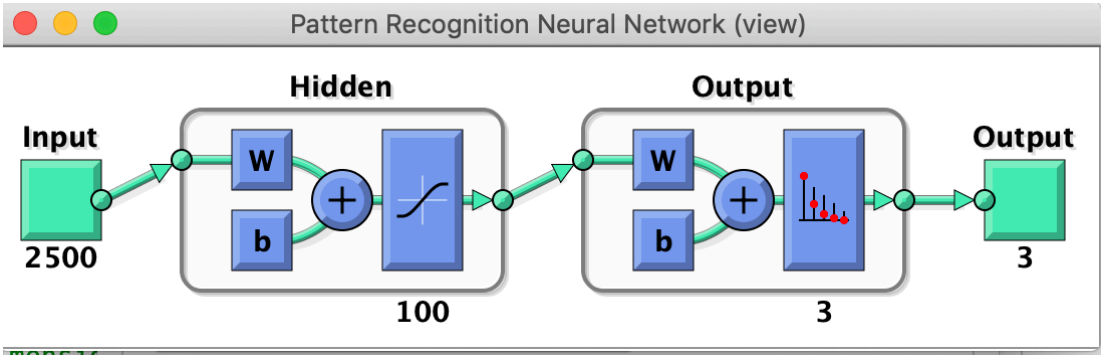
y is 3 * 3000, where 3 corresponds to vectorized label, 3000 corresponds to total number of images. The statistic of y will like this:

1	1	1	1	1	1	1	1	1	1
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

2018/2019 COMP1037 Coursework 2 – Machine Learning

Network Configuration:

The network being used is patternnet and following is the figure of the network configuration. As can be seen from the graph, the input of the network is 2500 and the hidden layer being used is 100. The output of the network is 3.



Best accuracy:

After several times of experiments, the best accuracy delivered from the test is 58.0. The pixel of the images has been changed into 50 * 50, the hidden layer is 100 and the learning rate of the network is 0.01. The trainRatio and testRatio are 0.8 and 0.2 respectively. Following is the confusion matrix of the best accuracy rate.

Meaning: The first three rows mean three different predicted classes, which is 1, 2 and 3 respectively. The first three columns represent the actual class of different pictures. For instance, the data in the

upper left corner means that for the pictures that actually belongs to class 1, only 113 pictures are predicted correctly, which stands for 18.8% of the total testing data, 63 of them are treated as class 2, and 19 of them are treated as class 3. The grid on the lower left corner means that the percentage of accurate prediction of class 1. And the one in the lower right corner stands for the total accuracy of the prediction, which is 58.0%.

Evaluation: From the confusion matrix, it can be seen that 58.0% of the prediction is correct, which means only about half of the prediction is accurate. It is still quite low after several modifications of different parameters and cannot meet the basic requirements of recognizing different categories. However, compared with class 1 and class 2, the correct prediction of class 3 is higher than other two labels. This may due to the noise in label 3 is less than label 1 and 2.

These are the parameters modified during the experiment:

	pixel	learning rate	hidden layer
DIFFERENT VALUE	50 * 50	0.05	25
	100 * 100	0.01	100
	200 * 200	none	150

Following are the accuracy acquired from the neural network based on different parameter (the accurate rate was the average value from ten times results):

ID	pixel	learning rate	hidden layer	accurate rate
1	50 * 50	none	25	51.9%
2	50 * 50	none	100	53.4%
3	50 * 50	none	150	52.7%
4	50 * 50	0.01	100	53.7%
5	50 * 50	0.05	100	53.1%
6	100 * 100	0.01	25	51.3%
7	200 * 200	0.01	25	50.6%

Conclusion: As can be seen from the statistics, setting a learning rate can increase the accuracy to some extent, however, sometimes, for size in 50 * 50 and the hidden layer is 100, the difference between learning rate in 0.01 and 0.05 is not that obvious. Compared with 0.05 and 0.01, 0.01 can provide a better accurate rate. Lower size of image may lead to higher accuracy. The disparity between different hidden layer is not that distinct as well. With more hidden layers, the accurate rate can increase to some extent, however, it doesn't represent that more hidden layers can gain higher accurate rate, it may depend on whether that number of hidden layer is best fitted for that network. We can see that after several changes of the value of different parameter, the accurate rate doesn't increase too much, this may due to the lack of enough sample to training or the limitation of traditional neural network.

Efficiency:

For image normalization, following code is used to derive the data. The function im2double is used to convert image to double precision.

```
normimage = im2double(newmap); % Normalize the image value.
newmap = normimage';
```

For the PCA, following code is used to apply PCA (however, as the pca may cause the decrease of the accurate rate, the code has been commented in the file):

```
X = X';
[coeff,score,latent] = pca(X);
a = cumsum(latent)./sum(latent);
dimension = 0;
for i = 1:size(a)
    if (a(i, 1) >= 0.95)
        dimension = i;
        break;
    end
end
tranMatrix = coeff(:,1:dimension);
X = X * tranMatrix;
X = X';
```

The input of the pca function should be a dataset X which is a n * p matrix, where n corresponds to the number of sample, and p corresponds to the dimension of each sample. The coeff means the principle component coefficient. The latent means the principal component variance. Using cumsum(latent) ./ sum(latent) to calculate the proportion of the variance. When the proportion is greater or equal to 0.95(95% of the original data), we can break the recursion. And reduce the coeff to only i-dimension, which can stand for almost 95% of the original data. When X = X * tranMatrix, the original X will be reduced to i-dimension.

CNN:

The CNN network being used in this coursework is a pretrained Convolutional Neural Network (CNN) called alexnet, which has been trained on approximately 1.2 million images. The network is 8 layers deep, the first five layers are convolutional and the last three are fully connected layer. The data being used in alexnet is quite different. imageDatestore is used to classify different classes and labels. The labels will be based on the subfolder name, and the class will be divided by folder as well. The image should be pre-handled and standardize to [227 227] and the image should have three channels. 75% of the data being treated as training data and 25% of them are test data. We have to modifying the layer of alexnet before using it, therefore, the last three layers has been changed to adapt to our cases. As we only have three classes, we have to change the class in fullyConnectedLayer to three. And add softmaxLayer, classificationLayer behind.