2018/2019 COMP1037 Coursework 2 – Machine Learning

FAI Coursework 2 – Neural Network Classification

Dataset Preparation [2 marks]

'build_animals.m' is provided as attachment.

NN Model Training [1 mark]

Training process is in my **'nn_animals.m'** file.

NN Model Testing [1 mark]

Code for displaying confusion matrix is also in **'nn_animals.m'** file, the confusion matrix of test data will be plotted after training ends.

Experiments [2 marks]

Optimize on parameters are in **'nn_animals.m'**.

Reports [3 marks]

a. b.

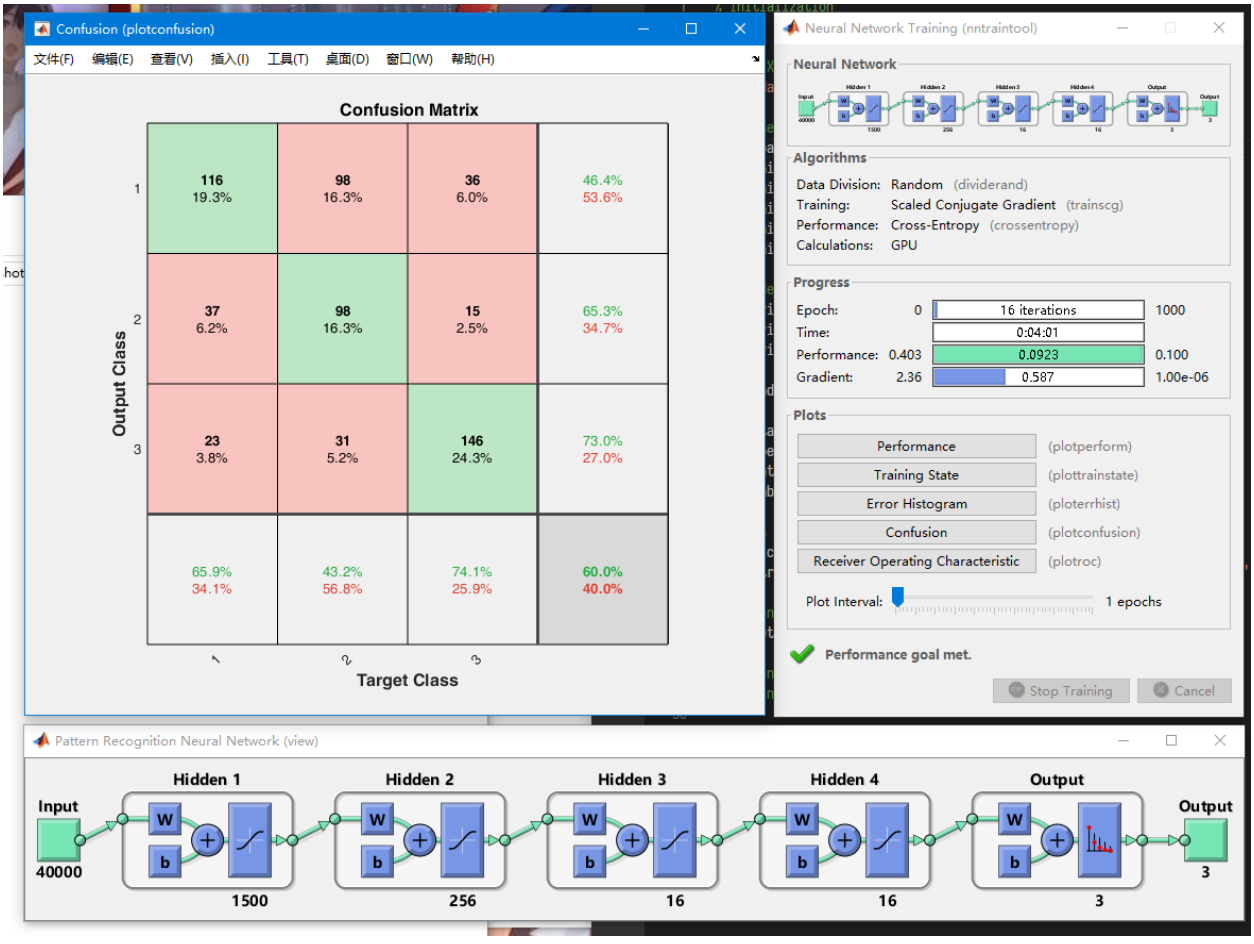

image size	200 × 200	**train goal**	0.1
network structure	1500 256 16 16	**learning rate**	0.05
train function	trainscg	**max epoch**	1000
transfer fucntion	tansig	**perform function**	crossentropy

Above are my confusion matrix and information about my model which have best accuracy for question a and b.

About confusion matrix, there are two part, in the inside square, for every column, number and percent of different output class for one target class are plotted, for every row, number and percent of different target class for one output class are plotted; value of percentage is number of one cell minus number of all test data; for percentage on fourth column, is the percentage from number of correct prediction minus number of all prediction for one output class, for percentage on fourth row, is the percentage from correct prediction minus number of all prediction for one target class. And the gray cell means total accuracy. In one word, the green number on gray cell better, the performance is better.

c. I tested the influence of performance function, image size, train goal, learning rate and network structure on performance of network.

1) Performance function

Basic Setting			
image size	200 × 200	**train goal**	0.1
network structure	256 64 16	**learning rate**	0.05
train function	trainscg	**max epoch**	1000
transfer function	tansig	**perform function**	

Performance Comparison			
net. performFcn	**gear**	**epoch**	**accuracy**
crossentropy	1st	22	54.5%
	2nd	25	58.7%
	3rd	23	55.5%
	average	23	56.2%
mse	1st	25	54.0%
	2nd	13	53.5%
	3rd	15	57.8%
	average	18	55.1%
mae	1st	39	55.2%
	2nd	32	56.2%
	3rd	40	56.3%
	average	37	55.9%

For the number of epochs need to reach the train goal, number of epochs on mae is greatest and the mse is the least one. As for accuracy, three algorithms are basically the same but crossentropy is better of 1% than mse.

Therefore, crossentropy is the best function for this task.

2) Image Size

Basic Setting			
image size		**train goal**	0.1
network structure	256 64 16	**learning rate**	0.05
train function	trainscg	**max epoch**	1000
transfer function	tansig	**perform function**	crossentropy

Performance Comparison			
Image Size	**gear**	**epoch**	**accuracy**
200*200	1st	22	54.5%
	2nd	25	58.7%
	3rd	23	55.5%
	average	23	56.2%
50*50	1st	24	56.0%
	2nd	26	55.0%
	3rd	27	55.2%
	average	26	55.4%
100*100	1st	23	58.8%
	2nd	25	58.3%
	3rd	22	56.7%
	average	23	57.9%

It seems that accuracy is best when image size is set to 100*100, it can be predicted that 200*200 is too large for current network to extract information but 50*50 is too small that there is a great loss of information.

3) Train Goal

Basic Setting			
image size	200*200	**train goal**	
network structure	256 64 16	**learning rate**	0.05
train function	trainscg	**max epoch**	1000
transfer function	tansig	**perform function**	crossentropy

Performance Comparison			
net.trainParam.goal	**gear**	**epoch**	**accuracy**
0.1	1st	22	54.5%
	2nd	25	58.7%
	3rd	23	55.5%
	average	23	56.2%
0.001	1st	59	53.5%
	2nd	60	56.5%
	3rd	68	56.8%
	average	62	55.6%
1	1st	1	43.7%
	2nd	0	33.2%
	3rd	0	32.7%
	average	0	36.52%

For training goal, 0.1 is the best choice for reasonable accuracy and epoch, number of epochs of 0.001 training goal is about 60 but get lower accuracy, it may be caused by overfit on train data.

As for 1 training goal, it is not trained actually and have bad accuracy on train set as well.

4) Learning Rate

Basic Setting			
image size	200*200	**train goal**	0.1
network structure	256 64 16	**learning rate**	
train function	trainscg	**max epoch**	1000
transfer function	tansig	**perform function**	crossentropy

Performance Comparison			
net.trainParam.lr	**gear**	**epoch**	**accuracy**
0.05	1st	22	54.5%
	2nd	25	58.7%
	3rd	23	55.5%
	average	23	56.2%
100	1st	23	58.0%
	2nd	19	56.5%
	3rd	20	57.7%
	average	21	57.4%
2	1st	24	59.0%
	2nd	23	57.2%
	3rd	26	55.8%
	average	24	57.3%

From the table, it can be concluded that in train function like “trainscg”, too large learning rate will not affect the performance, however, small learning rate will make training slower.

5) Network Structure (number of neurons in hidden layer)

Basic Setting			
image size	200*200	**train goal**	0.1
network structure	256 64 16	**learning rate**	0.05
train function	trainscg	**max epoch**	1000
transfer function	tansig	**perform function**	crossentropy

Performance Comparison			
patternnet()	**gear**	**epoch**	**accuracy**
[256 64 16]	1st	22	54.5%
	2nd	25	58.7%
	3rd	23	55.5%
	average	23	56.2%
[1536 256 64] (large number)	1st	23	59.7%
	2nd	19	56.5%
	3rd	15	57.5%
	average	19	57.9%
[256] (shallow)	1st	26	51.7%
	2nd	26	55.0%
	3rd	33	52.8%
	average	28.3	53.2%
[256 128 64 32 16] (deep)	1st	22	56.5%
	2nd	23	57.3%
	3rd	17	57.2%
	average	21	57.0%

From the table, it can be concluded that increase number of layer or number of the neurons can both get better performance, however, shallow net can have bad performance.

Efficiency [1 mark]

Normalization and PCA are implemented in **'nn_animals_efficiency.m'**.

PCA can compress the matrix to decrease the number of features, and the efficiency goes up, however, the accuracy is not good as 40000 input. The reason could be a loss of main information in images that leads to decrease on accuracy.

Bonus [1 mark]

My implementation of CNN in this coursework is **'cnn_animals.m'**.

CNN network in my coursework have three repeat of Convolution→ReLU→MaxPooling, the number of filters increases from 8, 16 to 32, and the a fully connected layer and softmax layer are on the end of network. The accuracy of CNN is pretty high, the reason is that Convolution→ReLU →MaxPooling can extract key information in images effectively.