2018/2019 COMP1037 Coursework 2 – Machine Learning

Neural Network Classification Report

FAI Coursework 2

A. Data instruction and training preparation

The folder '**animals**' contains 3000 images, which are divided into three classes: dogs, cats, and pandas with 1,000 images each. Each image is stored with the class of animal followed by an index number in the format of 'type_num.jpg'. By reading the images with the relevant MATLAB command, the data is generated in **build_animals.m** and is saved in **data.mat**. A NN model is trained in **'nn_animals.m'** and is saved in '**nn_model.mat**'.

B. Experiments with different parameter settings

Considering that the training result may be influenced by various factors, the following different parameter settings are attempted and results are compared. The results of each parameter setting are collected for 5 times in order to obtain the average and maximum levels of performance for better evaluation. This is primarily applied to decrease the influence of randomness.

a) number of neurons in one hidden layer

For this part, the 3000 images of 200 * 200 size are read one by one. X is used to store a matrix of 40000 * 3000 data (3000 samples in 3000 rows and 200 * 200 number of features in each row) and y represents a matrix of 3 * 3000 (3 classes and 3000 corresponding labels with 0 or 1). Here, 'patternnet' function is used to build NN models with default training function ('trainscg') and default learning rate (0.01). The training goal is set to 0.05. The NN model contains 3 layers, with input layer of 40000 inputs, a single hidden layer and an output layer of 3 outputs. Different number of neurons in the hidden layer are attempted.

# of neurons	accuracy1	accuracy2	accuracy3	accuracy4	accuracy5	average	max
5	49.83	52.00	49.17	51.17	50.83	50.60	52.00
10	50.17	51.50	50.50	50.83	51.00	50.80	51.50
15	49.00	52.33	49.50	52.17	53.33	51.27	53.33
20	51.33	54.50	53.30	54.50	53.67	53.46	54.50
25	53.50	54.50	53.83	54.00	52.00	53.57	54.50
50	55.17	55.00	52.83	55.33	55.83	54.83	55.83
64	56.00	55.00	58.00	53.83	56.17	55.80	58.00
75	52.17	56.83	52.67	49.33	56.50	53.50	56.83
100	50.67	50.33	49.17	51.83	54.33	51.27	54.33
200	51.50	50.17	49.00	54.00	51.50	51.23	54.00

Summary: According to the table, it is obtained that increasing the number of neurons, the performance seems to improve in the beginning and reach the highest performance in the case of 64 neurons, with the average accuracy is 55.80% and maximum accuracy is 58.00%. Then the performance falls with the increase of number of neurons. The NN model becomes more and more complex, which is initially helpful to perform well but will end up overfitting the training data and make a bad prediction.

2018/2019 COMP1037 Coursework 2 – Machine Learning

b) learning rate

For this part, same dataset structure is used and the network configuration is also similar with that of part a. The NN model contains 3 layers, with input layer of 40000 inputs, a hidden layer of 64 neurons and an output layer of 3 outputs. Different learning rate are attempted.

learning rate	accuracy1	accuracy2	accuracy3	accuracy4	accuracy5	average	max
0.0003	54.50	52.17	55.83	54.83	51.50	53.77	55.83
0.001	52.50	56.00	52.83	54.33	51.50	53.43	56.00
0.003	54.33	54.83	53.67	52.67	53.50	53.80	54.83
0.01	51.17	54.50	58.00	54.00	54.83	54.50	58.00
0.03	51.50	55.83	55.83	54.00	53.50	54.13	55.83
0.1	53.67	55.50	52.67	53.50	53.00	53.67	55.50
0.3	54.33	52.83	51.50	55.83	52.50	53.40	55.83
1	54.50	55.83	52.67	52.00	52.50	53.50	55.83

Summary: According to the table, it is obtained that increasing the learning rate by 3 times, the performance seems to improve in the beginning and reach the highest performance in the case of the learning rate of 0.01(default learning rate), with the average accuracy is 54.50% and maximum accuracy is 58.00%. Then the performance falls with the increase of learning rate. The low learning rate may helpful to obtain a better performance, but it takes long time to perform and may risk of overfitting. The high learning rate takes less time, but it is likely to miss the best case and results in underfitting.

c) number of hidden layers with the same number of neurons

For this part, same dataset structure is used and the different network configurations are implemented, where the number of hidden layers will be a factor to consider. Different number of hidden layers are applied to the NN model. (Having looked through the internet, same number of neurons in each hidden layer are recommended.)

# of neurons	accuracy1	accuracy2	accuracy3	accuracy4	accuracy5	average	max
64	56.00	55.00	58.00	53.83	56.17	55.80	58.00
64,64	55.33	56.17	56.33	55.00	56.5	55.87	56.50
64,64,64	57.50	57.67	55.83	55.00	54.67	56.13	57.67
64,64,64,64	56.50	53.17	57.00	56.5	56.83	56.00	57.00
64,64,64,64,64	55.17	54.33	57.33	56.33	53.83	55.40	57.33
64,64,64,64,64,64	56.83	53.67	54.50	54.50	55.83	55.07	56.83

Summary: According to the table, it is obtained that increasing the number of hidden layers, the performance seems to improve in the beginning and reach the highest performance in the case of 3 hidden layers with 64 neurons each, where the average accuracy is 56.13% and maximum accuracy is 57.67%. Then the performance falls with the increase of number of hidden layers. The NN model becomes more and more complex, which is initially helpful to perform well but will end up overfitting the training data.

d) train function

For this part, the dataset structure is used and the network configuration is same as that of part b. Here, 'patternnet' function is used to build NN models with and default learning rate (0.01). Several training functions are attempted.

function	accuracy1	accuracy2	accuracy3	accuracy4	accuracy5	average	max
trainscg	56.00	55.00	58.00	53.83	56.17	55.80	58.00
traincgf	56.80	55.00	54.67	55.67	52.67	54.96	56.80
trainoss	55.00	54.33	55.00	54.00	55.83	54.83	55.83
traingdx	55.17	56.50	52.33	52.67	52.67	53.87	56.50
traincgb	51.50	53.50	53.17	51.67	51.50	52.27	53.50
traincgp	54.17	53.67	54.83	55.17	53.17	54.20	55.17

Summary: According to the table, it is obtained that the best performance occurs when default training function 'trainscg' is used, where the average accuracy is 55.80% and maximum accuracy is 58.00%.

e) size of images

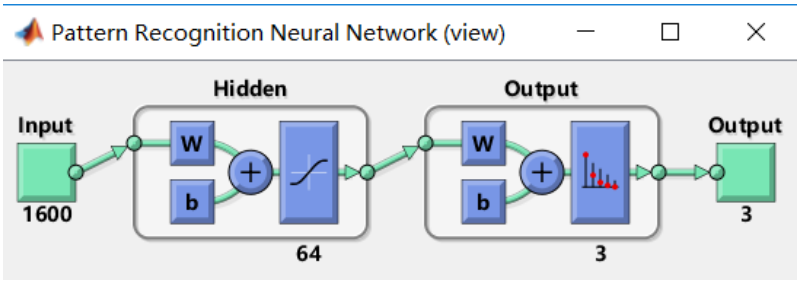
For this part, different dataset structures are used and the network configuration is similar with that of part b. The images are compressed into different resolutions.

image_size	accuracy1	accuracy2	accuracy3	accuracy4	accuracy5	average	max
40*40	52.17	57.50	55.00	60.17	58.00	56.57	60.17
50*50	57.50	56.17	57.33	56.33	55.00	56.47	57.50
100*100	53.83	55.83	55.83	55.17	56.83	55.50	56.83
200*200	56.00	53.50	55.50	53.83	56.17	55.00	56.17
400*400	54.17	52.50	56.67	55.33	54.17	54.57	56.67

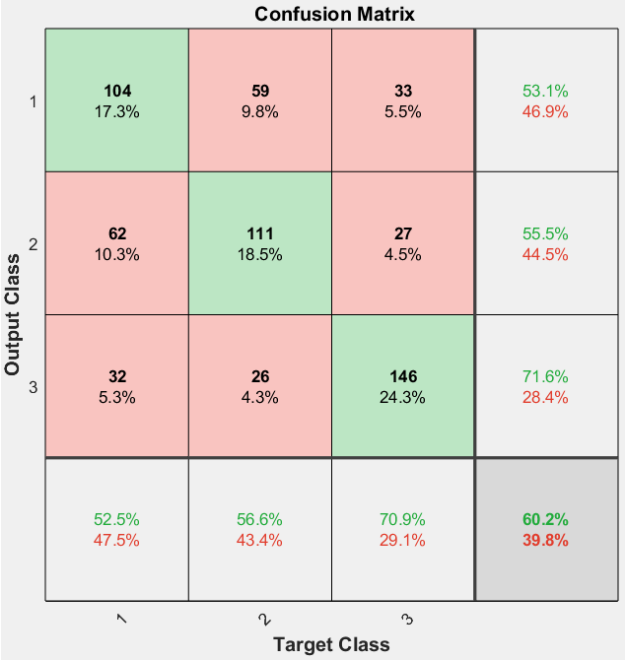
Summary: According to the table, it is obtained that the best performance occurs when the resolution is 40 * 40, with the average accuracy is 57.37% and maximum accuracy is 60.17%. Because the number of samples is only 3000, too many features (high resolution) will cause the training process overfitting. Reducing some number of features appropriately will improve the performance. However, if the resolution is very low, the process cannot catch the essential features, resulting in underfitting.

C. Best Performance

Up to now, the best performance occurs when the size of images is 40 * 40, which shows the average accuracy is 57.37% and maximum accuracy is 60.17%. The NN model contains 3 layers, with input layer of 1600 inputs, a single hidden layer of 64 neurons and an output layer of 3 outputs. The 'patternnet' function is used to train the model with default learning rate and default training function.



2018/2019 COMP1037 Coursework 2 – Machine Learning



In the figure, the row represents how many test samples are predicted to be that class; the column represents how many test samples are labeled in that class. For instance, 104 test samples (in green grid) are well predicted in the class 1 and 62 test samples (in pink grid) are predicted wrongly to be in the class 2 but they are labeled as class 1. The bottom row represents the ratio that how well the output data are predicted right in each labeled class. The rightmost column represents the ratio that how well the output data are predicted right in each output class. In the bottom right corner, it indicates the average performance of the prediction. In this figure, 60.2% output data are predicted right and 39.8% were predicted wrongly.

D. Efficiency improving steps

a) image normalization

For this part, max-min normalization methods are applied in **nn_animals_normalization.m** file. The basic idea is to normalize the image value from [0, 255] to the range of [0.0, 1.0]. The network configuration and dataset are same as that of part b in the previous section.

image_size	accuracy1	accuracy2	accuracy3	accuracy4	accuracy5	average	max
max_min	51.20	54.50	49.67	52.17	54.83	52.47	54.83

Summary: This method seems to perform the similar level as before. Sometimes the performance is even a bit worse. The time for training seems shorter, because the compressed data is convenient to compute but the information is lost.

b) image dimension reduction method

Principal Component Analysis (PCA) is applied to build a new dataset before training NN model in **nn_animals_pca.m** file. The dataset and network configuration are same as that of part b in the previous section. Here, k = 1000.

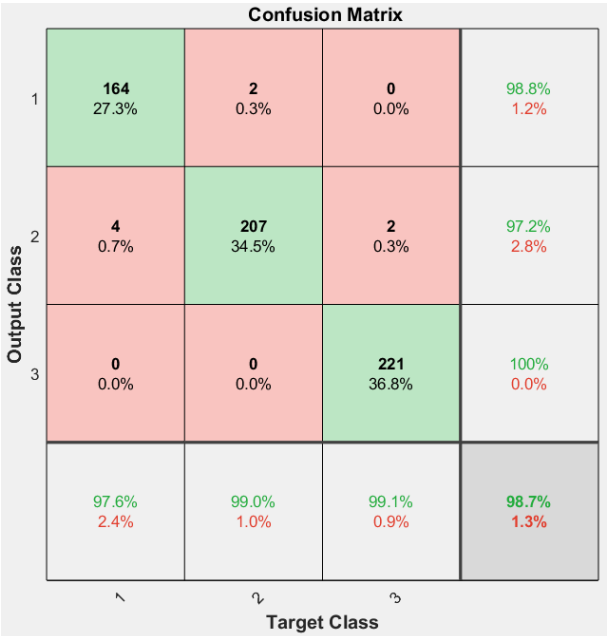
dimension	accuracy1	accuracy2	accuracy3	accuracy4	accuracy5	average	max
1000	39.83	33.50	34.83	36.83	35.67	36.13	39.83

Summary: PCA is a way to compress data and reduce memory needed to store data, resulting in speeding up learning algorithm. It projects n-dimension data into k-dimension data (n > k). In other words, it removes the noise and irrelevant information and extracts k most important features. However, it is likely to remove the useful features. According to the table, the performance of PCA is not good as expected when internal PCA function are used. The time for training is shorter than before because less data is computed

2018/2019 COMP1037 Coursework 2 – Machine Learning

.E. Bonus part: Convolutional Neural Networks

Convolutional Neural Networks (CNNs) is applied in **nn_animals_CNN.m** file. Before building the model, the dataset should be adjusted to the 4-Dimension dataset (200, 200, 1, 3000) and label set should be adjusted to a set with the size of 3000 * 1, where the 'cat', 'dog', 'panda' are classified as class 1, class 2 and class 3. The dataset is divided into three parts: 60% training data, 20% validation data, 20% test data. A CNN is composed of a series of layers, where each layer defines a specific computation. In the network, 9 layers are defined: an image input layer with [200 200 1], 2 convolution layers, 2 batch normalization layers, 2 pooling layers, 2 ReLU layers, 1 fully connected layer, 1 softmax layer, 1 classification layer.



According to the experiments, the accuracy is very high, has the average of 98.7%. In CNN, each layer applies different filters, combines the results and feeds the output into the next layer in the network. The convolutional layers define sets of filter weights, which are updated during network training. The ReLU layer adds non-linearity to the network, which allow the network to approximate non-linear functions that map image pixels to the semantic content of the image. The pooling layers down-sample data as it flows through the network. The three layers are the most important part in the network. CNN learns the values of these filters throughout the training process automatically and makes a prediction.