Introduction

The main goal of this coursework is to introduce the process of machine learning and give you first-hand practice experience of using machine learning and data analysis tools provided by Matlab.

Marks

Coursework 2 accounts for 10% of the FAI marks. Report should be written using this file as template (keeping the same format of margin, font size, spacing etc.). Longer report doesn’t lead to higher marks.

Plagiarism vs. Group Discussions

As you should know, there is no tolerance of plagiarism, and any breach of which will be dealt with according to the university standard policies. Please be very careful not to cross the boundary into plagiarism while having general discussions regarding the coursework to promote the generation of new ideas and to enhance the learning experience. The important part is that when you sit down to do the actual work and write the answers, you do it individually. If you do this, and you truly understand what you have written, you will not be guilty of plagiarism. DO NOT, under any circumstances, share code or share figures, graphs or charts, etc.

Deadline and Submission Procedure

The submission deadline is 12pm on the 3rd May 2019 via Moodle. Late submission results into 5% reduction of your coursework mark for each weekday. Any work handed in after the 11th May will receive zero marks.

Name your submission file: **FAIcw2-*XXX*.zip** (incl. your report **FAIcw2-*XXX*.pdf**, your code file called ‘**build\_animals.m**’, ‘**nn\_animals.m**’ and your model ‘**nn\_model.mat**’, where ***XXX*** should be your student ID number, and submit a single compressed file via Moodle.

If you can’t submit your coursework on time due to Extenuating Circumstances, please contact your personal tutor first. I am only granting an extension of submission based on his/her recommendation.

Please remove the above text while writing your coursework report.

FAI Coursework 2 – Neural Network Classification

Data instruction

The folder ‘**animals**’ consists of 3000 images, of three classes: dogs, cats, and pandas with 1,000 images each. In this coursework, you will be using these 3,000 samples. Each image is stored with the class of animal followed by an index number in the format of ‘type\_num.jpg’ (e.g., ‘cats\_00123.jpg’).



Fig. 1. Animals dataset example.

Coursework

Use Matlab Neural Network tool to train a NN to recognize cats, dogs and pandas.

1. Dataset Preparation [2 marks]
2. First, build up the required dataset to use later for NN model training. Read images from the folder into the workspace as the network input. You can name this filename as **‘build\_animals.m’**.

Hint: you can use functions including **‘fullfile’**, **‘dir’** and **‘imread’** to build full file name from parts, get the folder content list and read in image file one at a time.

1. After you read in any image, the image is in true color format with structure of width × height × 3. The 3rd dimension 3 indicated RGB colors with red, green and blue colors respectively. Training NN with images of RGB colors will take 2-3 times longer. Thus, it is better to convert all RGB images into grayscale images.

Hint: you can use the function **‘rgb2gray’**.

1. Notice that the given size of each image is different, so you need to standardize the size of image, e.g., 200 × 200 (width × height). You can choose other preference image size to use in this coursework.
2. Now, reshape the image from 2D into 1D array. Then, concatenate the image into a 2D array, let’s name it as ***X***. Your ***X*** should have the dimension of 40000 × 3000 (40000 corresponds to 200 × 200 image size, and 3000 corresponds to total number of images).

Hint: you can use the **‘reshape’** function to change array from 2D to 1D.

1. Next, vectorized the labels data into 1D, with cats, dogs, and pandas as class ‘1’, ‘2’, and ‘3’. Let’s name the labels data as ***y***. At the end of this part, your ***y*** should have the dimension of 3 × 3000 (3 corresponds to vectorized label, and 3000 corresponds to total number of images).

Hint: class 2 is expressed as 010.

1. Finally, save your ***X*** and ***y*** into **‘data.mat’**. Kindly verify your ***X*** and ***y*** are in the correct format and size.
2. NN Model Training [1 mark]
3. Create a new file, called it **‘nn\_animals.m’**. You can use this file to load the dataset that you had build in the previous section and used it to train a NN model.
4. Now, separate the data into training and testing set. You can use the NN toolbox to help you to separate the data into training, validation and testing sets while constructing net. Usually, the dataset can be separated into training set and testing set with ratio of 0.8:0.2 (general practice), but you are free feel to try another ratio.

At the end of the data preparation, you should have four data variables named as ‘trainData’, ‘trainLabels’, ‘testData’ and ‘testLabels’.

1. You can use Matlab NN toolbox ‘**feedfowardnet’** or ‘**patternnet’** to configure your neural network. Then, save your NN model as ‘**nn\_model.mat**’.
2. NN Model Testing [1 mark]
3. Use your trained net (or model) to test the performance of your net using the testing data you created in previous section. Then, plot out the confusion matrix to observe the performance.

Hint: you can use either the function ‘**net**’ or ‘**sim**’ to get your prediction results.

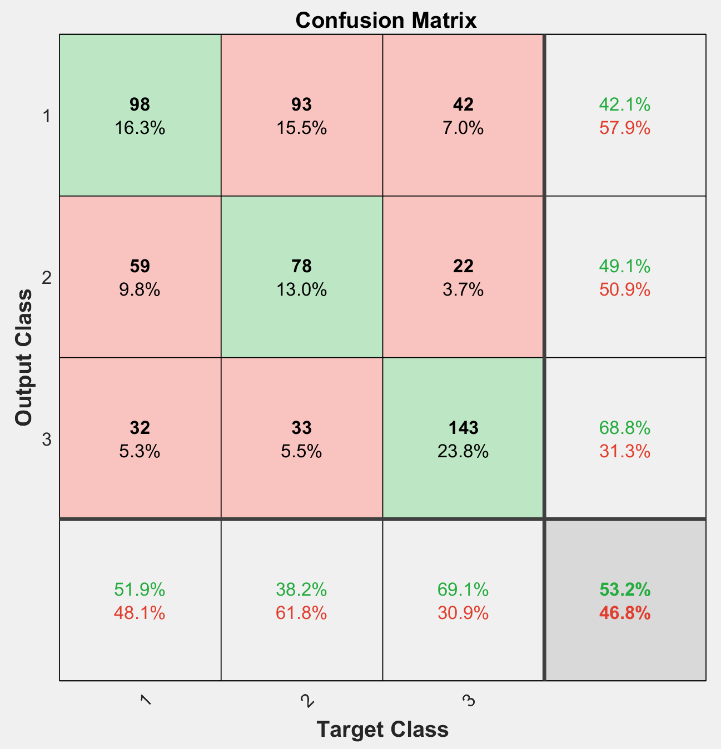


Fig. 2. Confusion matrix.

1. Experiments [2 marks]
2. Try to adjust different parameters to obtain your best accuracy, such as number of hidden layers, number of neurons in each hidden layer, learning rate, performance function, gradient function or other related parameters that you think that might affect the performance of your trained NN model.
3. You can also consider trying with different size of images (40 × 40, 100 × 100, 500 × 500 etc.) for training NN model and observe any differences if any.
4. You can also suggest or try alternative ways that you can do (or think) to improve the performance of the neural network. (e.g., learning rate, training function, etc.)
5. Reports [3 marks]

In your report (maximum 5 pages), please include the following content:

1. Show all your dataset structure, network configuration with figures or tables.
2. 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.
3. 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.
4. Efficiency [1 mark]
5. When you train with large number of images (probably 100,000 images) with complex designed NN, you will find that it takes days or weeks to train the NN model. This is partially due to the large dimension and large quantity of input images.
6. As such, you can apply (1) image normalization or (2) image dimension reduction method to improve the efficiency.
7. In image processing, normalization is the process that changes the range of pixel intensity values. Applications such as photographs with poor contrast due to glare, for example. Normalization is sometimes called contrast stretching or histogram stretching. In this case, the basic idea is to normalize the image value from [0, 255] to the range of [0.0, 1.0]. There are quite a few of normalization methods you can found online and feel free to try out several to observe the differences.
8. Principal Component Analysis (PCA) is a widely adopted data dimension reduction method. In this coursework, you can try to apply PCA when you build your dataset before training your NN model. Kindly provide appropriate explanation in the report to show your understanding of the PCA if applicable.
9. Bonus [1 mark]
10. Until now, you should have noticed that the current neural network is limited in classifying images that have high similarity. Thus, deep learning was introduced and the common type of NN in deep learning is ***Convolutional Neural Networks (CNNs)***.
11. The different of CNN with traditional NN is that each layer (hidden and neuron) applies different filters, typically hundreds or thousands of them, combines the results and feeds the output into the next layer in the network. CNN automatically learns the values of these filters throughout the training process.
12. You can try out this bonus section in this coursework by applying CNN in this coursework. It is to remind that CNN takes longer time (approximately 4 or 5 times longer to train the network, but faster with GPU). More information can be found out in the link as follow.

<https://ww2.mathworks.cn/en/solutions/deep-learning/convolutional-neural-network.html>

The CNN is not covered in this course as CNN is another topic in deep learning. This bonus section is basically just an introduction to you about the existing of CNN used in various computer vision applications.

All the best and wish you good luck! May the force be with you 😊