ELEC0144
Machine Learning for Robotics

Assignment 2

Year 2023/2024

Dr. Chow Yin Lai & Dr. Yu Wu

Guidelines:

- **All deadlines are specified in Moodle**, under the assessment section. Penalties will be applied for late submissions in accordance with the guidelines: https://www.ucl.ac.uk/academic-manual/chapters/chapter-4-assessment-framework-taught-programmes/section-3-module-assessment#3.12

- Please also be aware of **UCL's Academic Misconduct policy**: https://www.ucl.ac.uk/academic-manual/chapters/chapter-6-student-casework-framework/section-9-student-academic-misconduct-procedure. Collaboration with other teams via exchange of ideas, sharing of codes, re-using portions of the reports etc. are not allowed and will be considered as collusion.

1 Assignment 2: Backpropagation in Multilayer Perceptron & Transfer Learning in Convolutional Neural Network

1.1 Objective Summary

In the "MLP and Backpropagation" lecture, two examples have been shown – One on regression and another on classification. However, both were achieved using Matlab's Neural Network Toolbox. In the first part of the assignment, you are required to code the Multilayer Perceptron "from scratch", to demonstrate your understanding of the theory of backpropagation and optimisation.

In the second part of the assignment, you are required to perform transfer learning to train a pre-trained CNN to classify new classes which were not in the original training set. This will be done using Matlab deep learning toolbox.

In the last part of the assignment, you are required to write a two-page literature review about latest development in image classification.

1.2 Task 1: Regression

The following data (Figure 1) is generated using the code in Figure 2.

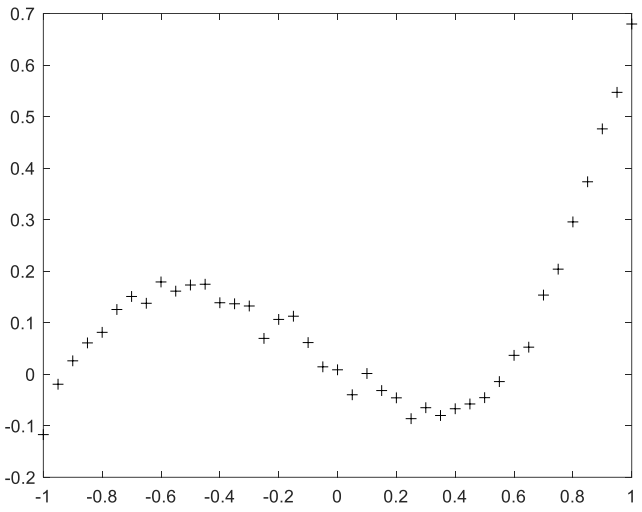

Figure 1: Training data for regression

```
%% the training data input x [-1, -0.95, -0.9, ..., 0.95, 1]

x=-1:0.05:1; % Note: need to put as row vector
len = length(x);

%% the training data output y, added with noise

d=0.8*x.^3 + 0.3 * x.^2 -0.4 * x + normrnd(0,0.02,[1,len]);
figure,plot(x,d,'k+')
```

Figure 2: Code for regression training data

a) Derive the backpropagation algorithm for a 1-3-1 network (1 input node, 3 nodes with tanh activation function in hidden layer, 1 linear output node) using **stochastic gradient descent**, and detail this in your report.

b) Write a Matlab code (.m preferred, or .mlx) from scratch, i.e. not using Matlab NN toolbox, to train the above network for regression of the data. After training, plot the network output for the unseen test data (Figure *3*). The expected output should be similar to that shown in Figure *4*. Discuss your results in the report.

```
xtest=-0.97:0.1:0.93;
```

Figure 3: Test data for regression

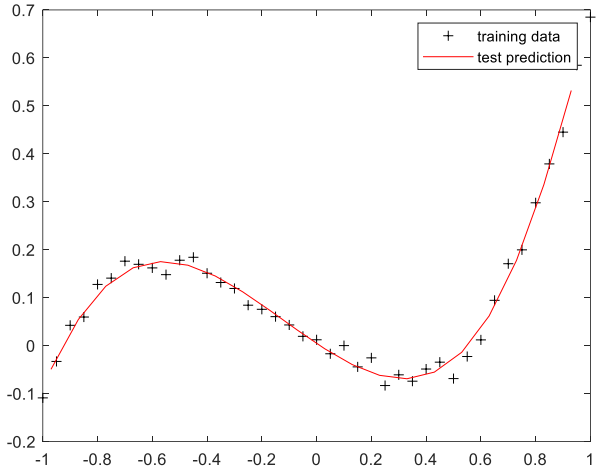

Figure 4: Expected outcome for regression

c) In a separate Matlab code, try using less or more hidden layer nodes, and compare the results with the network in part a) and b). Discuss the results in your report.

d) In a separate Matlab code, try using different activation functions for hidden and output layer, and compare the results with the network in part a) and b). You must try out ReLU in at least in one of your test cases. Discuss the results in your report. Also, please detail the changes in the local gradients when using these different types of activation functions.

e) In a separate Matlab code, try using different training algorithms (for e.g. ADAM instead of Stochastic Gradient Descent), and compare the results with the network in part a) and b). Discuss the results in your report.

f) Please use back to the original network (1-3tanh-1linear) network and by using stochastic gradient descent. However, this time, reduce the training data by 20% and 50% by removing 20%/50% samples randomly. Train the network and discuss the results.

g) In your report, sketch how you would use ReLU in the hidden layer to approximate the data.

1.3 Task 2: Classification

You are given the IrisData.txt file. Write your code to change the iris types into numbers, randomize the orders, then choose the first 70% of the data as training data and the remaining 30% as validation data.

Following that:

a) Derive the backpropagation algorithm for a 4-5-3-3 network (4 input node, 5 nodes in first hidden layer with tanh function, 3 nodes in second hidden layer with tanh function, and 3 output nodes) using **stochastic gradient descent**, and detail this in your report.

b) Write a Matlab code (.m preferred, or .mlx) from scratch, i.e. not using Matlab NN toolbox, to train the above network to classify the iris data into the correct classes. Discuss your results in the report.

c) In a separate Matlab code, try using less or more hidden layer nodes, and compare the results with the network in part a) and b). Discuss the results in your report.

d) In a separate Matlab code, try using different activation functions for hidden and output layer, and compare the results with the network in part a) and b). You must try out ReLU in at least in one of your test cases. Discuss the results in your report. Unlike in the regression task earlier, there is no need to detail the changes in the local gradients when using these different types of activation functions.

e) In a separate Matlab code, try using different training algorithms (for e.g. ADAM instead of Stochastic Gradient Descent), and compare the results with the network in part a) and b). Discuss the results in your report.

1.4 Task 3: Transfer Learning

Towards the end of the "CNN and Image Classification" lecture notes, an example code on performing transfer learning using Matlab has been provided. Another example which is similar can be found on this page:

https://uk.mathworks.com/help/deeplearning/ug/transfer-learning-using-alexnet.html

The "only" thing which has not been fully discussed is the Image Datastore, which is how images should be stored in Matlab. It is quite easy to set this up: Just create a folder with one subfolder for each label. For e.g.

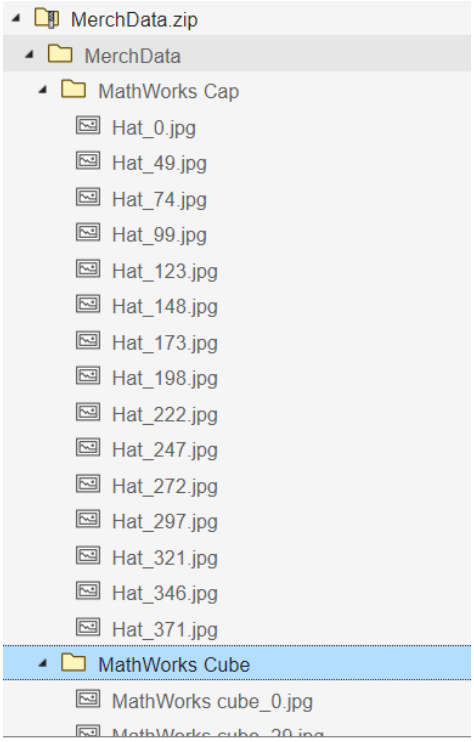

Figure 5: Image Datastore

And then run the command:

```
imds = imageDatastore('MerchData', ...
    'IncludeSubfolders',true, ...
    'LabelSource','foldernames');
```

Figure 6: Code to create Image Datastore

This assignment will allow you to understand the full process of performing transfer function on new image classes. Your tasks:

a) Download from the internet your own set of fruits data, comprising 5 different classes: Durian, papaya, kiwi fruits, mangosteen and mango. Collect 15 images for each category. You will need to resize them into 227 x 227 x 3 uint8. Then create the image datastore with meaningful names.

b) Perform transfer learning on the data, using AlexNet as the pre-trained network. Use 70% of the data for training and 30% of the data for validation. Discuss the results, and submit your code as well as your image folder.

c) Try varying the number of layers to be replaced and discuss the results. (Note: The last two layers will still be softmax and classification, so you will be varying the number of layers before the softmax layer).

d) Perform transfer learning on the data, using GoogLeNet as the pre-trained network. Use 70% of the data for training and 30% of the data for validation. Discuss the results, and submit your code as well as your image folder.

e) Perform transfer learning on the data, using GoogLeNet as the pre-trained network. Use 60% of the data for training and 40% of the data for validation. Discuss the results, and submit your code as well as your image folder.

f) For each of the subtasks above, please also try varying the learning parameters and discuss the effects on the outcomes.

1.5 Literature Review

In this task, you will write a **two-page** literature review about the latest development or algorithms for image classification. Please do not include any of the algorithms or network structure which have already been taught in the lecture. You may include one or two “older” algorithms for historical reasons, but emphasis should be given to newer algorithms since 2018.

1.6 What to Submit

- Your Matlab codes, with proper comments. The code will be tested! Please name your code in simple-to-follow manner, e.g. “Task1a.m”.
- A written report which details the implementation of your algorithms (e.g. how the derivatives of the activation function are implemented in the codes), results, comparisons, discussions, literature review etc.
- You should **put everything (Matlab codes AND your written report) into a zip folder**, then submit the zip folder onto the submission point on Moodle. Note: please do not submit .rar file – only .zip is allowed.
- Only one member per team needs to submit the zip folder.

1.7 More about the Report

The report should have a cover page clearly indicating the following details:

- Report title.
- Team number.
- Full name, student number and email address for each team member.
- Submission date.

The body of the report must be organized under the following section headings:

- Executive summary
- Regression
- Classification
- Transfer Learning
- Literature Review
- Teamwork – How the tasks have been split among the team members.
- Conclusion

The list of references should appear on separate pages. References should be formatted using the IEEE Citation Style. It is extremely important that all third party sources of information are properly credited and referenced in the correct manner. The inclusion of any text or diagrams from websites or documents must be clearly indicated and referenced.

Font size should be exactly 11 points. Recommended font type is Calibri or Arial. Text should be both left and right aligned (justified text). All figures should have captions, axes labels and legends where appropriate. Curves should be distinguishable even if printed in black and white.

1.8 Marking Criteria

This assignment contributes 40% to the overall score of the module. The marking criteria are described in the following table:

	Criteria	Mark Weight
Regression code	Code works properly, with good comments.	5%
Regression report	Details of the algorithm, derivation, discussions and comparisons clear and comprehensive.	6%
Classification code	Code works properly, with good comments.	5%
Classification report	Details of the algorithm, derivation, discussions and comparisons clear and comprehensive.	6%
Transfer learning code	Code works properly, with good comments.	5%
Transfer learning report	Details of the algorithm, derivation, discussions and comparisons clear and comprehensive.	6%
Literature Survey	Evidence of reading a wide range of literature, clear evidence of critical thinking, good organization.	5%
Report (Format)	English syntax and style, general organization and formatting, figure, table and equation presentation and use, literature citations are use all appropriate.	2%

1.9 Peer Review

Your group will receive a group mark for this assignment.

Individual students will then receive a different mark based on peer review. In the peer review (which will happen at the end of the term), you will be assessed by your team members on your attendance, effort, communication, contribution, respect, collaboration and standard of work.

The calculation is as follows:

- Group mark: Numerical score out of 40 (e.g. 30)
- Peer Review: Average percentage given by other students (e.g. 70%)
- Individual mark = Group mark x Peer Review (e.g. 30 x 70% = 21)

Note: Peer review submission is **compulsory**. Students who do not submit the peer review will have their individual peer review score capped at 70%.

You will be using the following peer review rubric:

Criteria	**No submission (0%)**	**Poor (40%)**	**Satisfactory (70%)**	**Good (100%)**
Attendance to meetings.	Never turns up to meetings.	Miss several meetings or late to several meetings, without notifying teammates.	Always attend meetings but late for several times. Teammates notified if cannot attend or late (with reasons provided).	Always attend meetings, always punctual. Teammates notified if cannot attend (with reasons provided).
Participation during meetings.	No participation during meetings.	Seldom actively providing suggestions, ideas, comments. Seldom participate in discussions. Seldom respectful to other students' ideas.	Actively providing suggestions, ideas, comments most of the time. Participate in discussions most of the time. Respectful to other students' ideas most of the time.	Always actively providing suggestions, ideas, comments. Always participate in discussions. Always respectful to other students' ideas.
Contribution to project and standard of work.	No contribution to the project.	Minimal contributions, frequently needs help from others.	Works independently but quality is not very high, needs some help with work.	Excellent quality of work, can work independently and able to help others if needed.

Communication	No communication at all.	Huge delay in replying messages / emails. Huge delay in seeking help, thereby delaying progress badly or create huge stress towards deadline.	Slight delay in replying messages / emails. Slight delay in seeking help, thereby delaying progress slightly or create some stress towards deadline.	Keep others up-to-date with the progress. Seek help early on (if needed) so as not to delay the project or create stress towards deadline. Fast in replying messages / emails.