ELEC0144

Machine Learning for Robotics

Assignment 2

Year 2025/2026

Dr. Yu Wu & Dr Roselina Arelhi

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 and Reinforcement Learning

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. You will then use MATLAB’s Neural Network Toolbox only to further explore the effects of hyperparameters on the final results.

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 the MATLAB Deep learning toolbox.

In the last part of the assignment, you will be required to write your own code to implement tabular Q-Learning to find the optimal path to the goal with the highest reward.

1.2 Task 1: Regression

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

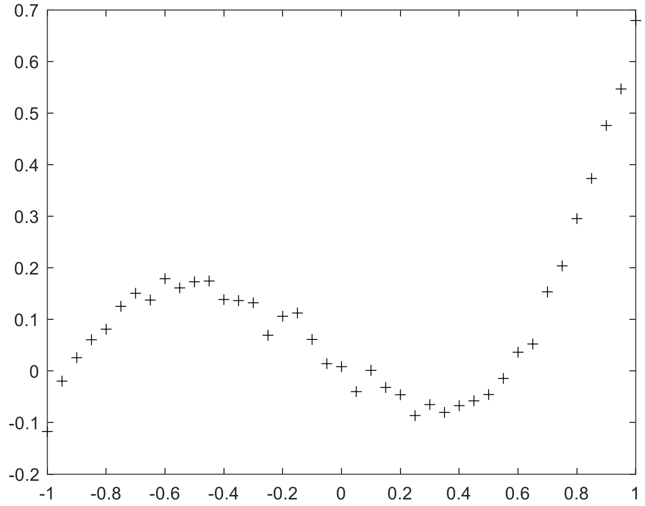
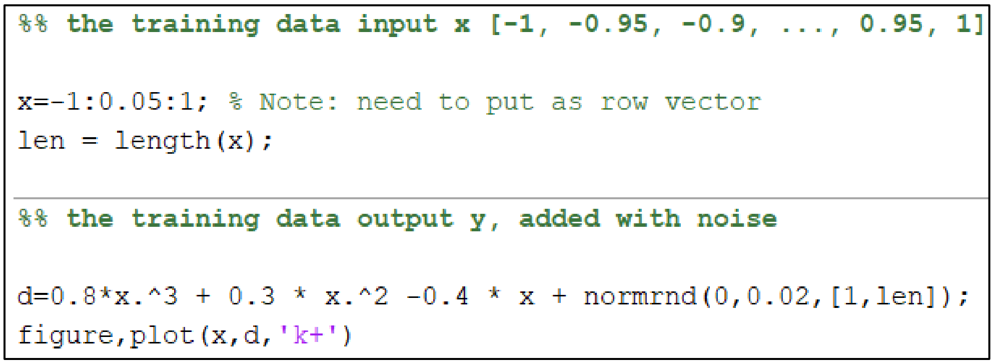


Figure 1: Training data for regression

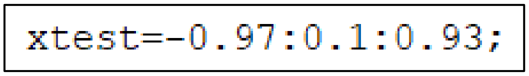


```
%% the training data input x [-1, -0.95, -0.9, ..., 0.95, 1]

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 the hidden layer, 1 linear output node) using **stochastic gradient descent**, and detail this in your report. You should provide the details of your calculation for one weight per layer, and then generalize this for other weights in the layer (if there are more than one weights).

b) Write a MATLAB code (.m preferred, or .mlx) **from scratch**, i.e. not using the 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

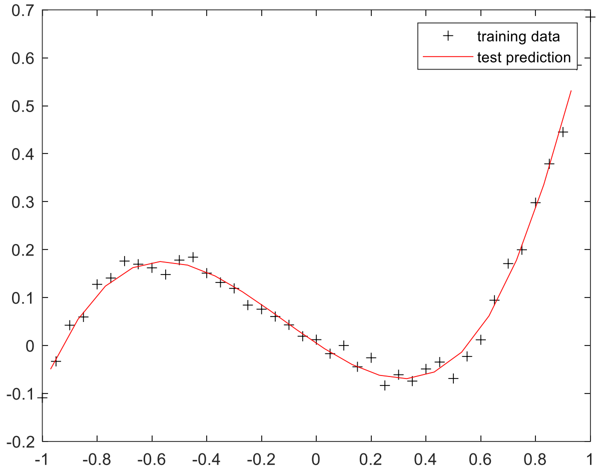


Figure 4: Expected outcome for regression

Important note for task b): There is no need to write a code which is too complex or advanced. For e.g. there is no need to make the code automatically adaptable when you change the number of nodes or layers. I only require a simple code, which demonstrates your understanding of the backpropagation algorithm for the 1-3-1 network.

c) 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.

d) Use MATLAB's Neural Network toolbox to explore the following:

- Using less or more hidden layer and / or nodes.
- Using different activation functions for hidden and output layer. You must try out ReLU in at least one of your test cases.

Then compare the results with the network in part a) and b). Discuss the results in your report.

e) In your report, sketch how you would use ReLU in the hidden layer to approximate the data. (See slides 45 and 46 in week 2 notes).

Note: You may also use Python **if your whole group agrees to it**. However, for b) and c) you may only use basic package including panda, numpy and matplotlib.

No advanced packages are required for this task.

For d), you may use Keras.

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 nodes, 5 nodes in the first hidden layer with tanh function, 3 nodes in the second hidden layer with tanh function, and 3 output nodes) using **stochastic gradient descent**, and detail this in your report. You should provide the details of your calculation for one weight per layer, and then generalize this for other weights in the layer.

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.

   Important note for task b): There is no need to write a code which is too complex or advanced. For e.g. there is no need to make the code automatically adaptable when you change the number of nodes or layers. I only require a simple code, which demonstrates your understanding of the backpropagation algorithm for the 4-5-3-3 network.

c) 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.

d) Use MATLAB's Neural Network toolbox to explore the following:

   - Using less or more hidden layer and / or nodes.
   - Using different activation functions for hidden and output layer. You must try out ReLU in at least one of your test cases.

   Then compare the results with the network in part a) and b). Discuss the results in your report.

Note: You may also use Python **if your whole group agrees to it.** However, for b) and c) you may only use basic package including panda, numpy and matplotlib.

No advanced packages are required for this task.

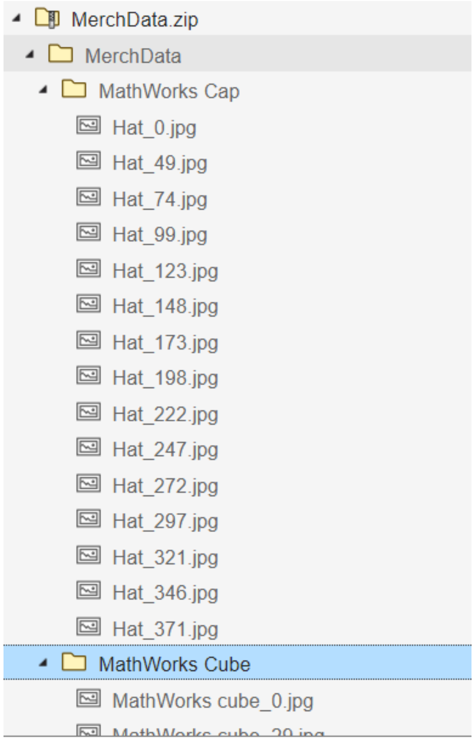
For d), you may use Keras.

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

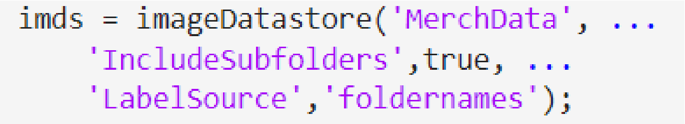
The "only" thing which has not been fully discussed is the Image Datastore, which is how images should be stored in MATLAB. It is quite easy to set this up: Just create a folder with one subfolder for each label. For e.g.



```
MerchData.zip
  MerchData
    MathWorks Cap
      Hat_0.jpg
      Hat_49.jpg
      Hat_74.jpg
      Hat_99.jpg
      Hat_123.jpg
      Hat_148.jpg
      Hat_173.jpg
      Hat_198.jpg
      Hat_222.jpg
      Hat_247.jpg
      Hat_272.jpg
      Hat_297.jpg
      Hat_321.jpg
      Hat_346.jpg
      Hat_371.jpg
    MathWorks Cube
      MathWorks cube_0.jpg
      MathWorks cube_20.jpg
```

Figure 5: Image Datastore

And then run the command:



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

Figure 6: Code to create Image Datastore

ELEC0144 – Assignment 2

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) or (224 x 224 x 3 unit8). 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) 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.

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

Note: You may also use Python **if your whole group agrees to it**. For this task, you are allowed to use KERAS or Pytorch.

1.5 Task 4: Tabular Q-Learning

You are given a grid-world as shown in Figure 7:

9	10	11	12 (+10)
5	6	7	8 (-10)
1	2	3	4

Figure 7: Grid world

There is a positive reward of +10 if the agent enters cell number 12, and a negative reward of -10 if the agent enters cell number 8. Cell number 6 is an obstacle, which the agent cannot enter. The agent will also be refrained from exiting the grid-world. For e.g. if the agent is in cell number 2, and the action is down, then the agent will still stay in cell number 2.

The agent can only perform 4 actions: Up, Right, Down and Left. It is assumed that there is no slip, thus the agent will execute the command perfectly.

There is a living reward of -1 for each step taken, even if the agent tries (but fails) to exit the grid world or enter the obstacle (cell number 6).

Based on your understanding of the tabular Q-Learning algorithm taught in the lectures, write your own code to implement the algorithm. The results should be the Q-values of all the state-action pair, as well as the best action for each cell.

In your report, please explain the updates of the Q-values for the first three iterations (similar to lecture slides) of the first episode, second episode and third episode.

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”. The comments should be clear so that the assessor can “match” the code with derivations / equations from the report.
- A written report which details the derivations, results, comparisons, discussions, etc. You should **NOT** add snippets of codes into your report to explain how these are implemented, as this would make the report very long. This is why the codes and comments should be self-explanatory.
- 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
- Q-Learning
- 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.

Page limit is 30 pages (including text and images) using single spacing and 11 points Arial font. 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 70% 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.	10
Regression report	Details of the algorithm, derivation, discussions and comparisons clear and comprehensive.	10
Classification code	Code works properly, with good comments.	10
Classification report	Details of the algorithm, derivation, discussions and comparisons clear and comprehensive.	10
Transfer learning code	Code works properly, with good comments.	2
Transfer learning report	Details of the algorithm, derivation, discussions and comparisons clear and comprehensive.	4
Q learning code	Code works properly, with good comments.	10
Q learning report	Details of the algorithm, derivation, discussions and comparisons clear and comprehensive.	10
Report (Format)	English syntax and style, general organization and formatting, figure, table and equation presentation and use, literature citations are use all appropriate.	4

For each of the categories above, the marking rubric is as follows:

	0-39%	40-49%	50-59%	60-69%	70-79%	80-100%
Codes	Program does not work, or has major flaws that prevent its intended use. Program is very difficult to read. Code contains lines that do not work or are out of order.	Program mostly work, but has major flaws. Program is difficult to read.	Program works in the way the student intended, but has minor flaws. Program is slightly difficult to read.	Program works in the way the student intended. Program is well organized, easy to read and understand.	Program is functional and refined. Program is well organized, makes good use of white space and comments.	Program is functional and exceptionally refined, with extra features that exceed the requirements. Program is extremely well organized, makes good use of white space and comments. Variables have helpful names.

	0-39%	40-49%	50-59%	60-69%	70-79%	80-100%
Report	Missing or very poor written-report.	Details and examples are not organized, are hard to follow and understand. Unable to find specific details.	Information is scattered and needs further development. Details are somewhat sketchy.	Information is logically ordered with paragraphs and transitions. Some details don't support the report topic.	Information is presented in effective order. Good structure of paragraphs and transitions enhances readability and comprehension. Supporting details are specific to topic and provide the necessary information. Good discussions which demonstrates critical thinking and analytical skills.	Information is presented in effective order, and is of award or publication quality. Excellent structure of paragraphs and transitions enhances readability and comprehension. Supporting details are specific to topic and provide the necessary information. Exceptional discussions which demonstrates excellent critical thinking and analytical skills.
Format	A (very) large number of language errors. Inappropriate language style. Significant proportion of the report(very) difficult to follow. Literature missing or incorrectly cited, unsuitable formatting, or not accessible, e.g. wrong or broken links, non-English language.	Frequent language errors and/or inconsistent use of language style distract from the content. Citations accessible, but significant formatting errors distract from the content.	Report is generally readable, but with significant number of language errors. Language style requires some reworking. Mostly correct citations and formatting following guidelines, with some errors or inconsistencies.	Report generally well written, using scientific/technical language, but with some language errors. Correct citations and formatting with minor errors or inconsistencies.	Report well written throughout, always using of scientific/technical language. No or only minor language errors. Correct citations and formatting without errors or inconsistencies.	Error free Report, with excellent use of scientific/technical language. Correct citations and formatting without errors or inconsistencies. Citations might be hyperlinked to online version where possible.

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.