MLPR Assignment Part 1 Feedback

Below are comments on some recurring issues I came across when marking Part 1: The Next Pixel Prediction Task of the MLPR assignment. Please note that the individual feedback for Part 1 consists of the combination of the printed feedback form attached to the front of your script and the written comments throughout the script.

1 General comments

1.1 Answer everything you're asked to

A number of people missed out on marks by not answering certain parts of questions (e.g. describing the data set based on the histogram in Q1.a or commenting on the structure of the three-dimensional plot in Q2.a). Often, these required observations were quite simple (it was enough to state that most patches in the data set were flat (Q1.a) and note that most points on the plot clustered around the line from the origin to (1, 1, 1), concluding that the pixels in question largely had similar values (Q2.a)). However, many students also did not provide any comments or comparisons of their results when they were specifically asked to do so (e.g. Q3.b, Q4, Q5.a).

It is important to make sure to always answer every part of the question, in order not to miss on an odd mark here and there. Comments and comparisons are even more crucial: often, most marks for a question are awarded for the interpretation of results, which allows to evaluate one's understanding, not for a (meaningless outside of the context) value one reports.

1.2 Using vectorised MATLAB expressions

A number of students made use of for-loops (e.g. for populating the feature matrix X in linear regression) instead of employing MATLAB's vectorisation. It is generally advisable to make yourself comfortable with vectorising your code for a number of reasons: MATLAB (and other similar technical computing languages) are optimised for vectorised code, which makes it run much faster than the corresponding code containing loops; vectorised code appears more readable and, consequently, is often less error prone. Further, some students missed bugs in their code (e.g. defining a matrix to be of wrong dimensions) due to using loops; whereas an attempt to run the corresponding vectorised expression would have immediately alerted them to a dimension mismatch error. No penalty was incurred on using non-vectorised code.

1.3 Regression versus classification

A number of people, when asked to evaluate regressor's performance in Q2.c, described the test data points as being 'linearly separable'. Linear separability refers to the property of a (binary) classification data, whereby a hyperplane can be placed between the multidimensional data points such that all of one class label lie on one side of it and all of the other – on the opposite side. Note that this definition relies on class labels being associated with each data point.

For the data set you were given, the outputs were continuous valued, not class labels, so it did not make sense to refer to the data points as being linearly separable.

1.4 Labelling plots

A worrying number of people did not correctly label plots they produced. At the very least, a plot should have the quantities being plotted on each axis labelled and if multiple data sets are plotted on the same axes, a legend given to identify how each data set is represented. Ideally all figures should also have captions (on top of any description in the main text), which succinctly describe their contents, so that the reader can quickly identify what they are showing without having to dig through the text for references to them.

1.5 Role of training and test data sets

Some people seemed confused about the different roles of training and test data sets. As the name suggests, the training data set is for 'training' i.e. fitting the parameters of a model such that some objective is achieved. The test data is a distinct set of data points, which are not used during training and are instead used to test whether the trained model's performance generalises to data it did not see before. Therefore, it does not make sense to refit the model's parameters using the test set before evaluating its performance on that test set (as your test data would effectively just become an independent set of training data). Unfortunately, this was quite a common mistake in implementing linear regression in Q4 or training the radial basis function (RBF) network in Q3.b.

1.6 Reporting significant figures

Many students reported an unreasonably large number of significant figures for root mean squared errors (RMSEs). It is recommended to critically evaluate the usefulness of the information that you report; don't report anything that does not add to your answer. The same applies to overly long-winded answers that could be phrased more succinctly. However, this should not be taken at the expense of the informativeness and completeness of the answer, e.g. such as that expected from the discussion in Q6.

1.7 Commenting on RMSEs

In this assignment, RMSEs served as the main tool for evaluating and comparing performance of different models. Therefore, it was desirable for a good answer to explain what these values (computed on training and test data sets) measure and how they and the difference between them relate to the model's performance. It is not enough to simply state than one RMSE is smaller than the other, concluding improved performance. This argument extends to a more general one: you should avoid making non-trivial statements without providing a reasonable justification.

1.8 General comment on overfitting

In Q4, many students noticed potential overfitting of the linear regression model on all pixels due to a larger difference between its RMSEs, and proceeded to claim it with various degrees of confidence. This behaviour was not an integral part of that question and was initially attributed to numerical precision problems. The general advice for statements regarding overfitting follows below.

Before confidently stating overfitting, a good answer should reflect on the number of model parameters and the number of training data points. If their difference leaves some room for doubt (and even if not), further diagnostic measures should be employed to investigate this. These include but are not limited to: computing standard errors on the RMSEs for training/test sets to evaluate how different they really are; checking the condition number of the covariance matrix to diagnose numerical errors associated with matrix inversion; permuting the data sets and altering the training/test set splits to see if an unfortunate data shuffle is causing the observed behaviour; performing regularisation by adding a constant to the diagonal of the covariance matrix to see if this alleviates the effect.

It is, however, fair to state that one models overfits to the training data more than the other, upon examination of their RMSEs (and ideally, the standard errors on them). This was the case in Q5.a, where it was fine to state that the well-trained neural network (NN) showed a worse case of ovefitting compared to linear regression on all pixels. This could be further supported by appreciating that the NN had roughly 10 times more parameters than linear regression, compared to about 17000 data points in the training set.

2 Question-specific comments

2.1 Q1.b: Misinterpretation

Many students failed to correctly read the question and interpreted it as "provide a simple way to predict if a patch is flat". While the phrasing of the question can suggest this interpretation at a short glance, it is important to read the question carefully. Further, in the next paragraph, the assignment introduces a threshold for the standard deviation to divide the data set in flat and non-flat patches. Assignments very rarely give you an answer right after asking a question; this should have made those who made this mistake reconsider their interpretation.

Additionally, many students provided a number of ways to predict the target pixel in a flat patch: the answers ranged from the mean (mode, median etc.) to the value of the neighbouring pixel. All of these methods would pretty much be the same for a flat patch, so providing one "simple way" would have been more than enough. (Exceptions include those who related their answer to computational efficiency – in these cases, discussing alternative methods made sense.)

2.2 Q2.a: Downsampling

Another point of confusion was caused by a suggestion to downsample the training data to avoid overprinting in a 3D plot in Q2.a. Some students continued to

use the subset of the data they plotted (often 10000 points, as instructed in the assignment) for computing the linear regression weights in Q2.c. A few scripts also went on to test their model on an even smaller (5000 points) subset of the test data. Notably, these answers did not explain the purpose of doing so. This underlines the importance of always providing a commentary to your answer (what exactly you did and why): not only does it illustrate your understanding, but it can also help you notice problems with your answer. If you don't understand why you are doing something, it will probably become clear to your marker too.

2.3 Q2.c (and Q4): Left division operator

It is generally advised to use matrix left division instead of using the $inv()$ operator. The following is taken from the MathWorks documentation:

"If A is a square matrix, A\B is roughly equal to $inv(A) * B$, but MATLAB processes A\B differently and more robustly."

2.4 Q3.a: Report all of your results

Most students noticed that the RMSEs were not stable across multiple runs. However, only a small number decided to explain what experiments they performed and which number of RBFs they chose in the end. A good answer would report the number of times the script was run and summarise the results in a plot with error bars (instead of plotting some illustrative results from a single run, as many students did). This formal and informative presentation of results would immediately provide strong justification for the choice of the number of RBFs.

Most students have utilised function handles in their code in order to implement cross-validation. It would be beneficial to provide at least a short explanation of this data type. This applies to all concepts not explicitly mentioned in the assignment that you choose to discuss in your report: acronyms should be expanded, error measures or scores should be defined, etc.

2.5 Q4: More data does not always mean better performance

A very common interpretation of the improved performance of linear regression on all pixels, compared to linear regression and RBF network on adjacent pixels, was that we have more data (more features). Note how this explanation is not enough on its own: having more unrelated data would not improve the predictive power of the model. The conclusion you should have made was that more pixels in the patch convey useful information about the target pixel than just its two neighbours that we considered. This observation could be further evaluated in the context of the question: does this make sense for natural images? In general, a good answer would try and link every result and its interpretation back to the original problem; only a few students discussed image compression in Q6, which was a very good idea.

2.6 Q5.a: Misinterpretation

Another question that many people misread was Q5.a, in which you were supposed to use the provided well-trained neural network, in order to predict pixel values for the training and test sets. Many people instead took the code from Part B and trained a new network with it, consequently (and incorrectly) declaring its performance as being worse than that of linear regression, without mentioning the insufficient training time. Q5.b (which would have been virtually the same question as Part A then) should have alerted you to the wrong interpretation of that question.

2.7 Q6: Discussion

From a technical coursework of this level, it is generally expected to provide a more informative evaluation of model performance than simply resorting to words such as "good", "normal", or "excellent". The same applies to comparing different models, which was a large part of this assignment. Many students claimed either "similar" or "better" performance of the RBF network in comparison to linear regression on adjacent pixels. In both cases, no justification was provided apart from reporting the RMSEs themselves. A small number of students reported standard errors on the RMSEs, supporting their conclusions. However, there were also students who reported standard errors but failed to make any use of them.

Note that a standard error gives an indication of the uncertainty in our estimate of some quantity (in our case, RMSE) due to computing it with a finite subset of the assumed infinite possible set of test examples. Therefore, two models can be claimed to demonstrate similar performance by showing that the difference between their RMSEs is a lot smaller in magnitude than the standard errors on those RMSEs.

It is important to note that computing a standard error on an RMSE does not equal to taking a squared root of the standard error on the MSE, as squared root is not a linear transformation. Instead, one could use Taylor expansion to arrive at:

$$f(x \pm \delta) \approx f(x) \pm \delta f'(x),$$

with

$$f(x) = \sqrt{x}.$$

Quantification becomes even more important when doing a comparative analysis between various models. An exceptional answer could quantitatively compare a model's performance, generalisation abilities, and runtime. Further, one could utilise big O notation to rigorously contrast computational complexity of different models.

Finally, a discussion is supposed to be a detailed summary of your findings throughout the assignment, with added comparisons between different models. Some people answered Q6 by simply ranking the models by their RMSEs in a single sentence, whereas others went into full depth of describing further experiments they have actually performed. While the latter is much preferred over the former, neither of these approaches are ideal. It is great to include additional investigations in your script (if done right, they can only gain marks!). However, this should not be done at the expense of answering the question.