EECS708P Assignment 1 Part 2 Report

Name: Bheki Maenetja
Student Number: 230382466
Assignment: No.1 Part 2

Classification

Question 1

Plot the normalized and raw training sets; what do you observe?

Looking at the 3D scatter plots of both the raw and normalised training sets we observe that the distribution of data points has roughly the same "shape" for both training sets, with differing scales on the axes of course. Both training sets provide us with useful information. The plot for the raw training set shows us the natural scales and distribution of the features whereas the plot for the normalised set is more effective in outlining the structure of the data without one feature dominating another. This difference is most clearly observed in the colour of the data points (which represent the values for petal width). For the raw training set the data points are mostly light in colour, ranging from light purple to bright yellow. The plot for the normalised set has a more uniform colour distribution and thus gives a better picture of how petal width correlates with the rest of the features.

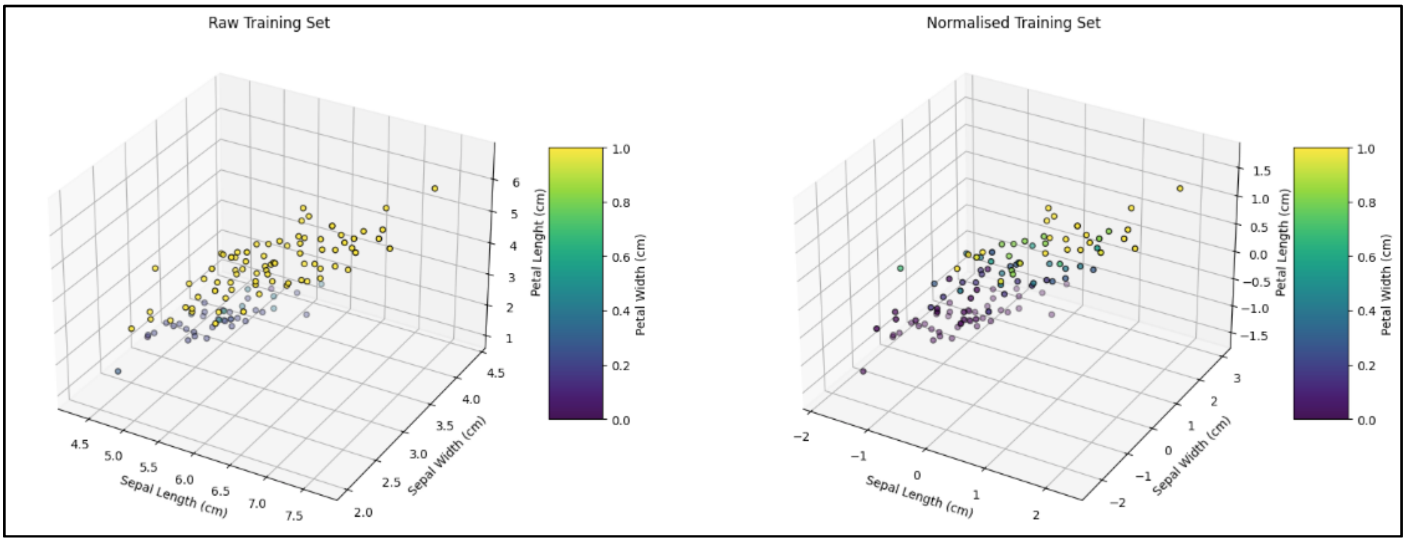

Figure 1 3D Scatter Plots for both the Raw and Normalised Training Sets

Question 5

Draw the decision boundary on the test set using the learned parameters. Is this decision boundary separating the classes? Does this match our expectations?

Looking at the decision boundary that has been derived from the learned parameters, it's clear to see that the decision boundary is not doing an adequate job of separating the target class (setosa) from the rest of the classes. This goes against our expectations.

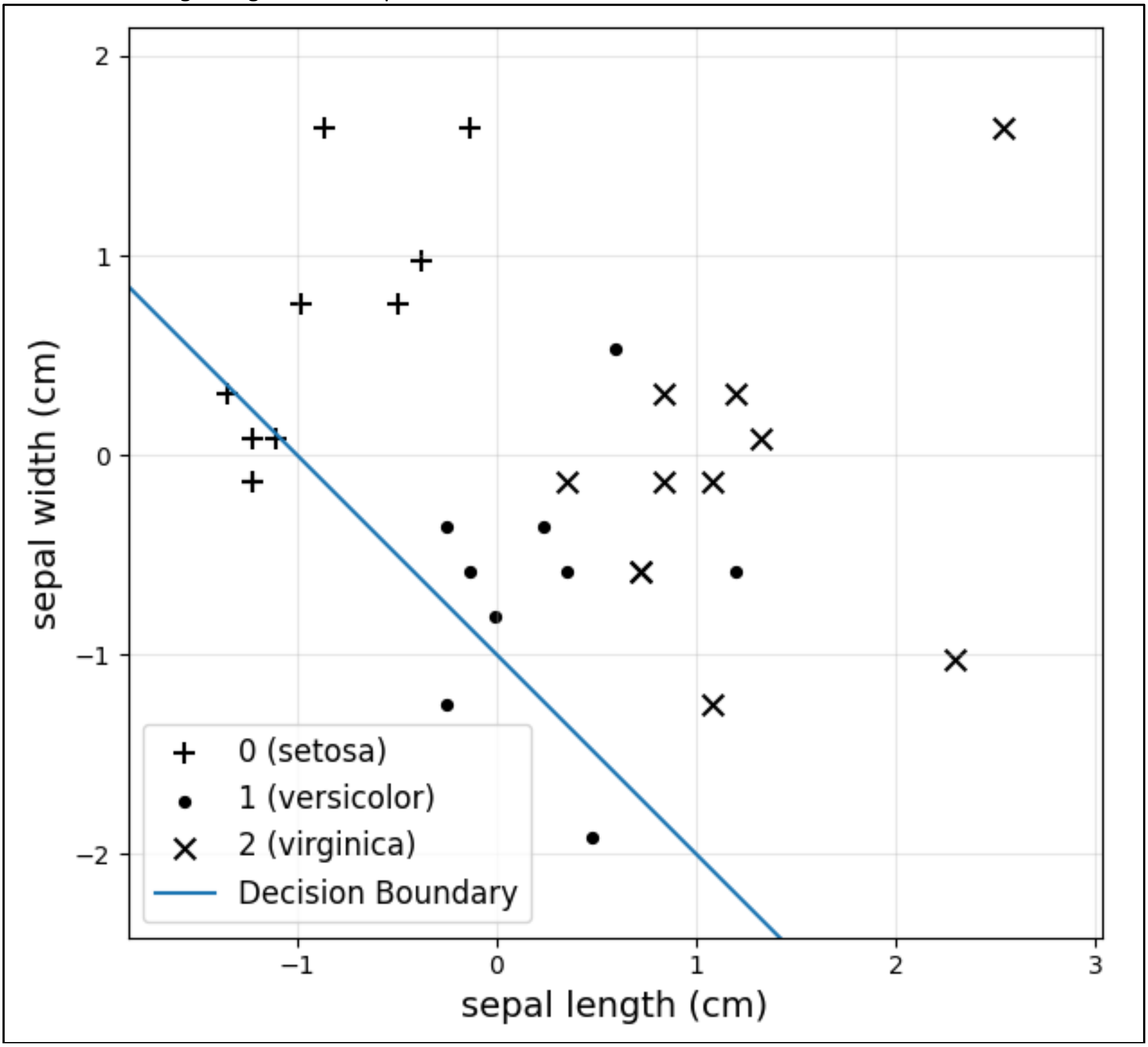

Figure 2 Decision boundary plotted against features and classes of the test set.

Question 6

Using the 3 classifiers, predict the classes of the samples in the test set and show the predictions in a table. Do you observe anything interesting?

Looking at the results, we observe that all of the 3 classifiers appear to have a high level of accuracy with a substantial majority of the predictions matching the actual class. We also observe that the classifiers for each class have been well-tuned to their particular class, giving very high probabilities when they predict the correct class. This suggests that the classifiers are able to clearly distinguish their class from the others and provide a good means of separation between the classes. The setosa classifier, in particular, is very adept at distinguishing its class from the others, assigning very high values for positive predictions and very low values for negative ones. This aligns with the fact that the setosa class is much more linearly separable from the other classes in terms of sepal width and length. Conversely, there is considerably more overlap between the virginica and versicolour classes; and as we see in the table, predictions for these 2 classes are made with less confidence.

	Sepal Length	Sepal Width	Petal Length	Petal Width	Setosa Classifier	Versicolor Classifier	Virginica Classifier	Actual Class	Predicted Class
0	0.353037	-0.582617	0.555446	0.022155	0.034584	0.730531	0.462601	Versicolor	Versicolor
1	-0.132515	1.643944	-1.156546	-1.174194	0.999960	0.173763	0.106643	Setosa	Setosa
2	2.295244	-1.027930	1.810907	1.484359	0.000004	0.843587	0.944460	Virginica	Virginica
3	0.231649	-0.359961	0.441313	0.420938	0.037899	0.588261	0.708477	Versicolor	Virginica
4	1.202753	-0.582617	0.612513	0.288010	0.008923	0.771613	0.519965	Versicolor	Versicolor
5	-0.496679	0.753320	-1.270679	-1.041267	0.999682	0.330506	0.076097	Setosa	Setosa
6	-0.253903	-0.359961	-0.072284	0.155082	0.285290	0.554678	0.526268	Versicolor	Versicolor
7	1.324141	0.085351	0.783712	1.484359	0.002132	0.405783	0.967232	Virginica	Virginica
8	0.474425	-1.918554	0.441313	0.420938	0.000526	0.921228	0.368889	Versicolor	Versicolor
9	-0.011127	-0.805273	0.098915	0.022155	0.080298	0.736139	0.368329	Versicolor	Versicolor
10	0.838589	0.308008	0.783712	1.085576	0.013318	0.351382	0.948750	Virginica	Virginica
11	-1.225006	-0.137305	-1.327745	-1.440050	0.999310	0.589052	0.021290	Setosa	Setosa
12	-0.375291	0.975976	-1.384811	-1.307122	0.999921	0.316538	0.045047	Setosa	Setosa
13	-1.103618	0.085351	-1.270679	-1.440050	0.999513	0.535696	0.025276	Setosa	Setosa
14	-0.860843	1.643944	-1.270679	-1.174194	0.999983	0.127946	0.133594	Setosa	Setosa
15	0.595813	0.530664	0.555446	0.553865	0.148834	0.335801	0.864182	Versicolor	Virginica
16	0.838589	-0.137305	1.183177	1.351432	0.000796	0.461435	0.969724	Virginica	Virginica
17	-0.253903	-1.250585	0.098915	-0.110773	0.041452	0.828408	0.251235	Versicolor	Versicolor
18	-0.132515	-0.582617	0.441313	0.155082	0.049313	0.658656	0.570762	Versicolor	Versicolor
19	0.717201	-0.582617	1.069044	1.351432	0.000362	0.583191	0.955076	Virginica	Virginica
20	-1.346395	0.308008	-1.213612	-1.307122	0.999648	0.416779	0.047170	Setosa	Setosa
21	0.353037	-0.137305	0.669579	0.819721	0.014380	0.473474	0.890168	Virginica	Virginica
22	-0.982230	0.753320	-1.213612	-1.041267	0.999743	0.284803	0.098344	Setosa	Setosa
23	0.717201	-0.582617	1.069044	1.218504	0.000487	0.605554	0.940540	Virginica	Virginica
24	2.538020	1.643944	1.525575	1.085576	0.017973	0.199684	0.978878	Virginica	Virginica
25	1.081364	-0.137305	0.840778	1.484359	0.001220	0.450353	0.966500	Virginica	Virginica
26	1.081364	-1.250585	1.183177	0.819721	0.000113	0.852011	0.774290	Virginica	Versicolor
27	1.202753	0.308008	1.240243	1.484359	0.001260	0.346206	0.982157	Virginica	Virginica
28	-1.225006	-0.137305	-1.327745	-1.174194	0.998752	0.543513	0.037810	Setosa	Setosa
29	-1.225006	0.085351	-1.213612	-1.307122	0.999304	0.501156	0.037292	Setosa	Setosa

Table 1 Predictions of the 3 classifiers against the actual class.

Question 8

Looking at the datapoints below, can we draw a decision boundary using Logistic Regression? Why? What are the specific issues or logistic regression with regards to XOR?

A core assumption that is made when using logistic regression is that the classes can be separated by a single linear boundary. As we can see in this case, that simply cannot be done; therefore, we cannot draw a decision boundary using logistic regression. The XOR pattern requires a non-linear decision boundary, typically involving higher-dimensional feature space to learn the parameters. Logistic regression is a purely linear function that is unable to tackle the XOR problem.

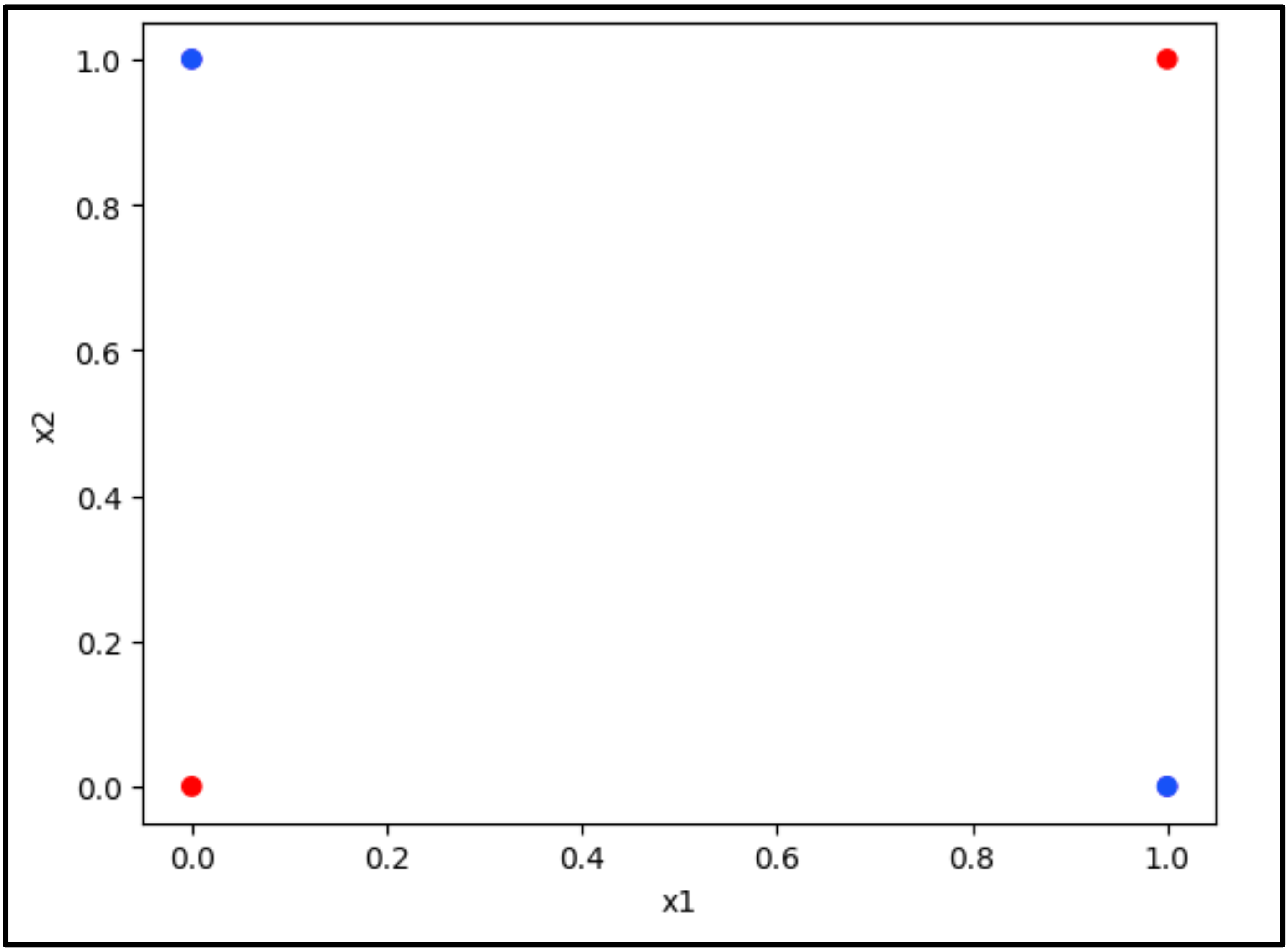

Figure 3 The XOR pattern

Neural Networks

Question 1

Why is it important to use a random set of initial weights rather than initializing all weights as zero in a Neural Network?

Randomising a neural network's initial parameters ensures that the neurons will learn different features differently to each. Were all the parameters set to zero or one, then each neuron in a layer would effectively be the same, as they would all be adjusted by the same gradient. Randomised weights allow each neuron to start at a different point in the search space, and learn different patterns in the data; thereby allowing the network to deal with complex inputs.

Question 2

How does a NN solve the XOR problem?

ANNs solve the XOR problem by using non-linear activation functions, in a hidden layer of nodes, to create a feature space where the points of the XOR pattern are linearly separable; either by line, plane or hyperplane. This enables the neural network to learn the XOR pattern and produce accurate outputs.

Question 3

Explain the performance of the different networks on the training and test sets.

Looking at the performance of the different multilayer perceptron networks (refer to code block 11 in the 2nd notebook), we observe a clear correlation between the number of hidden units and the cost (for both the training and test sets). As the number of hidden nodes increases, the training and test costs go down precipitously. The MLP with 1 hidden unit has a training cost of 1.096 and test set cost of 1.098 whilst the MLP with 32 hidden units has costs of 0.67 and 0.68 respectively. It should also be noted that for all MLP models the gap between the training set cost and test set cost is negligible, indicating that all the models generalise well to new data.