COMP7703 – Homework Task 6
Joel Thomas 44793203

1.

For the case of a single neuron with a linear activation function, without a bias term, the output after activation must pass through the origin whenever the linear combination of the inputs and the weights is passed into the activation function. Using bias weights allows us to shift the linear activation function left or right (equivalent to controlling the "y-axis intercept") to better fit the data.

Generalising this to a larger network, having bias weights can shift an activation function left or right for each neuron in each layer of the network. This may be critical for successful learning in order for the MLP model to be able to better fit the data.

2.

The discriminant function in Figure 1 will incorrectly classify a number of the outer class samples as belonging to the inner class (see bottom portion of the outer circle) whereas the discriminant functions appearing in Figures 2 and 3 appear to be much better fits for the type of the underlying data (donut dataset).

It is evident that the trained model in Figure 1 became stuck in a local optimum as seen by the large difference between the train and test error diagrams after model convergence (decreases in error become smaller and smaller) – this may potentially explain the inaccurate discriminant function. Conversely, the trained models in Figures 2 and 3 may be very close to/are the global optimum for this MLP model (parameters, dataset) as is evident by both their train and test error diagrams displaying the desired convergence curve as well as both the final errors being very close to 0 in each trained model.

Using the reasons explained above, the more samples there are of the outer class located towards the bottom, the higher the error for the trained model in Figure 1 as they are guaranteed to be misclassified as belonging to the inner class. Thus, we expect poor generalisation performance for this model. In contrast, the trained models in Figures 2 and 3 are expected to have good generalisation performance as they do well on both the training and validation datasets (train and test error diagrams) as well as have desirable discriminant functions for this type of dataset.

3.

The rectified linear (ReLU) is a very sharp non-linear activation function in that for a given input < 0, the output after activation is 0 but for a given input ≥ 0, the output after activation is the same as the input. Due to this non-linear combination, it is differentiable everywhere except at the origin where the input $= 0$ (left-side derivative $= 0$, right-side derivative $= 1 \neq 0$). This type of non-linearity is the underlying cause for the discriminant functions in Figures 1-3 appearing much sharper around the boundary edges.

Conversely, the hyperbolic tangent (Tanh) is also a non-linear activation function but note that it is much smoother and is also bounded as the input $\rightarrow \pm\infty$ (usually ± 1 but modifiable through appropriate scaling). It is also a differentiable function as it is differentiable everywhere. Thus, this explains why the discriminant function in Figure 4 appears much smoother around the boundary edges.

4.

$$y_i = \frac{\exp(o_i)}{\sum_k \exp(o_k)}, \qquad o_2 = o_3 = 0.2$$

$$\therefore y_1 = \frac{\exp(o_1)}{\sum_{k=1}^{3} \exp(o_k)}$$

$$y_1 = \frac{\exp(o_1)}{\exp(o_1) + \exp(0.2) + \exp(0.2)} = \frac{\exp(o_1)}{\exp(o_1) + 2\exp(0.2)}$$

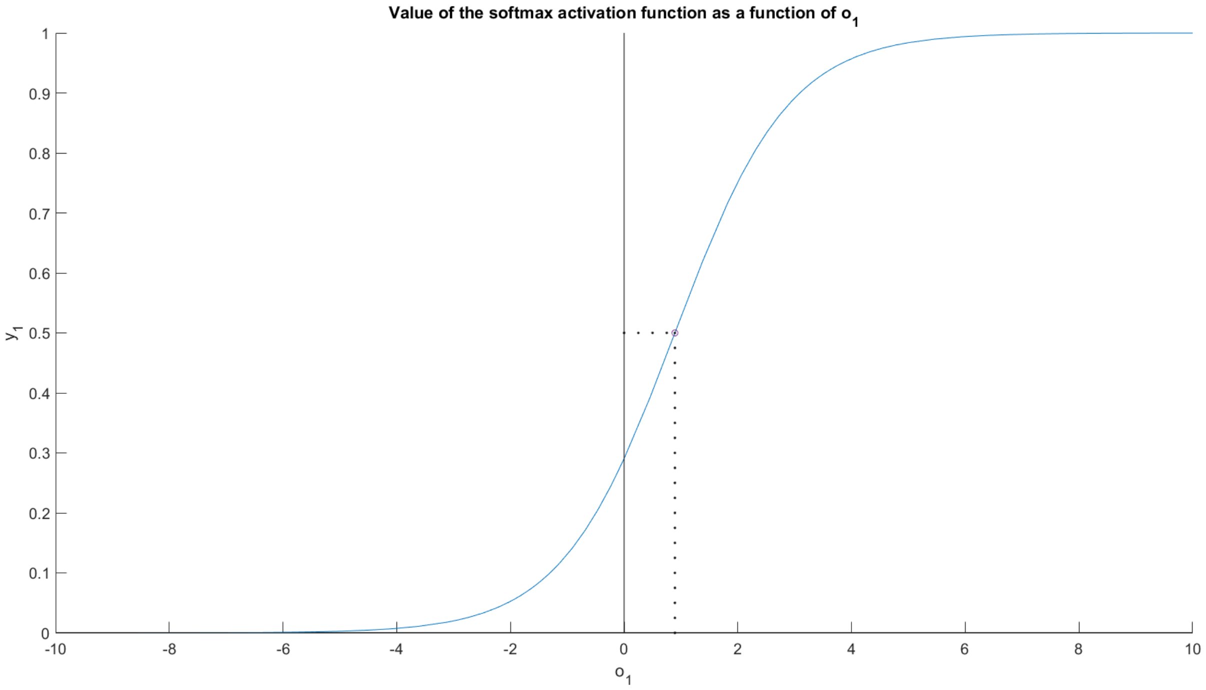

Figure 1: Plot of the softmax activation function as a function of o_1