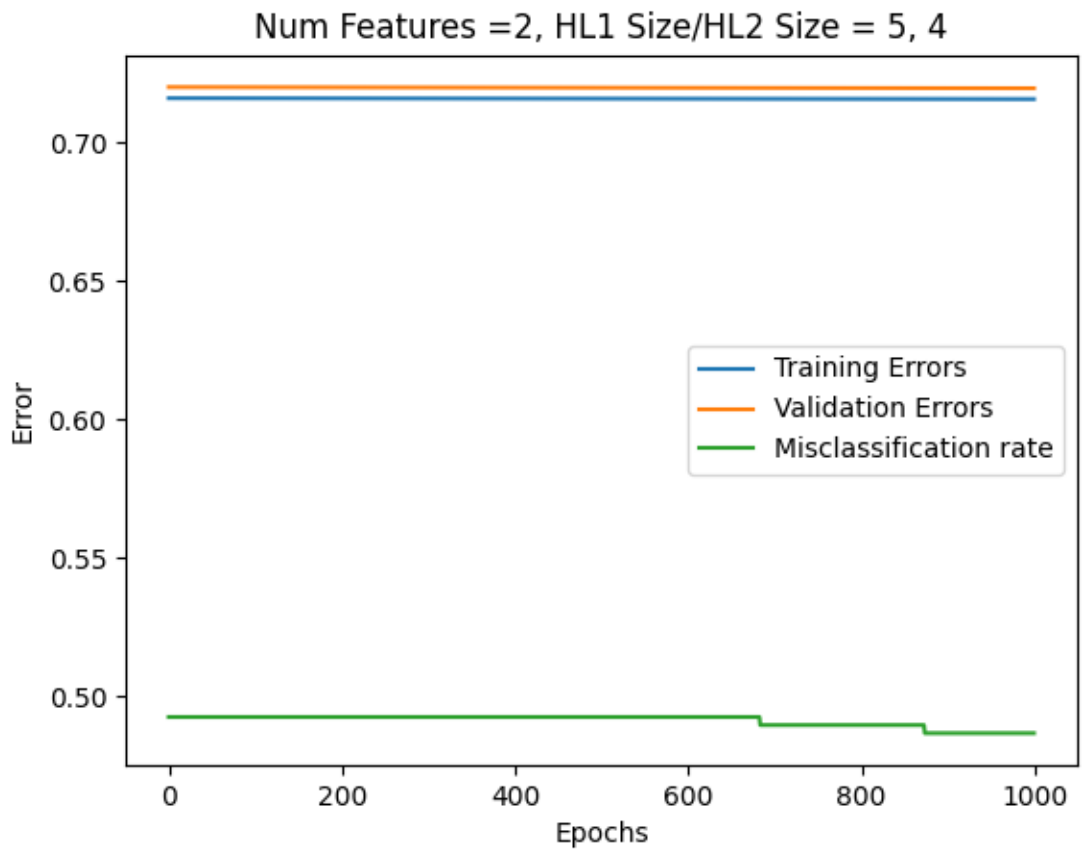
McMaster University

4SL3 Assignment 5

Uday Sharma - 400139248
12-6-2021

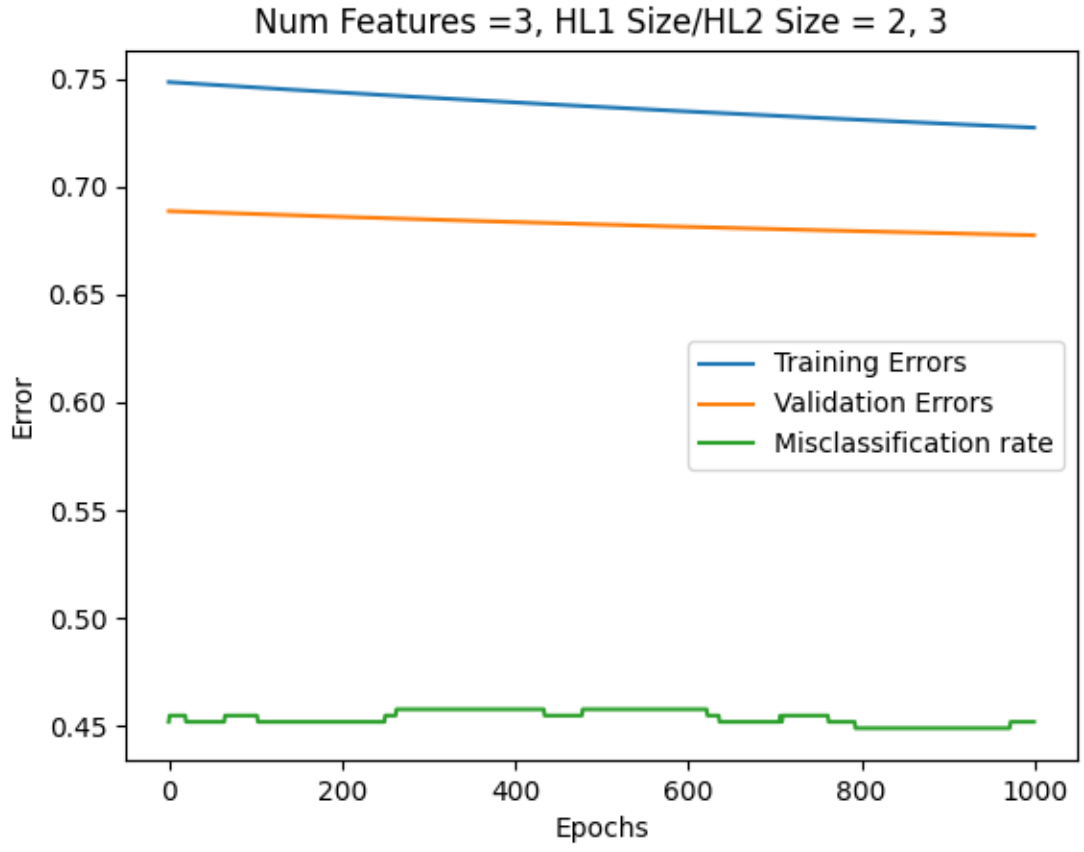
2 Features:

Hidden Layer 1 Size	Hidden Layer 2 Size	Cross Entropy Loss - Validation	Misclassification Rate
2	**2**	0.662718	0.199708
2	**3**	0.771308	0.421283
2	**4**	0.821665	0.421283
2	**5**	0.970792	0.421283
3	**2**	0.767103	0.769679
3	**3**	0.81334	0.80758
3	**4**	0.804767	0.718659
3	**5**	0.804703	0.443149
4	**2**	0.77711	0.41691
4	**3**	0.715666	0.521866
4	**4**	0.844168	0.702624
4	**5**	0.974159	0.421283
5	**2**	0.757394	0.421283
5	**3**	0.744632	0.421283
5	**4**	0.651971	0.460641
5	**5**	0.881916	0.400875



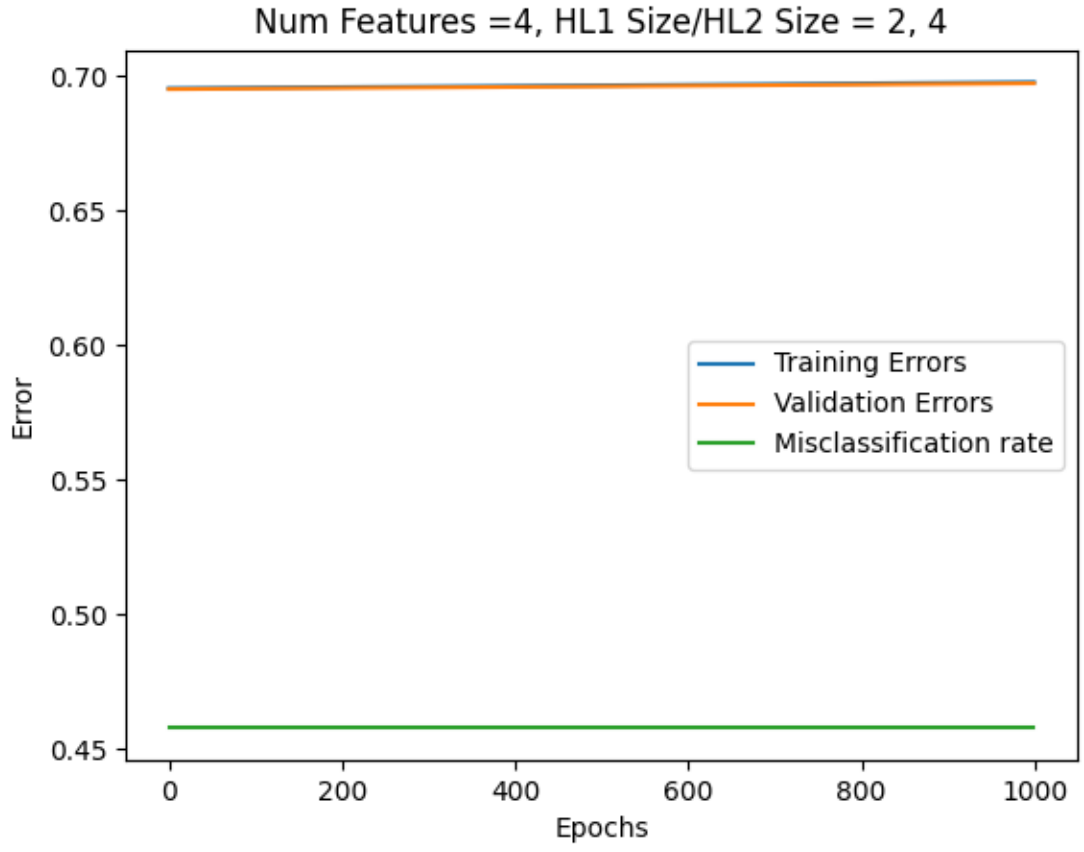
3 Features:

Hidden Layer 1 Size	Hidden Layer 2 Size	Cross Entropy Loss - Validation	Misclassification Rate
2	**2**	0.699915	0.421283
2	**3**	0.60752	0.450437
2	**4**	0.912838	0.734694
2	**5**	0.962499	0.51312
3	**2**	0.747659	0.365889
3	**3**	0.861111	0.421283
3	**4**	0.768055	0.670554
3	**5**	0.816243	0.421283
4	**2**	0.693034	0.421283
4	**3**	0.793724	0.421283
4	**4**	0.776555	0.421283
4	**5**	0.705697	0.712828
5	**2**	0.718986	0.421283
5	**3**	0.719174	0.427114
5	**4**	0.70249	0.430029
5	**5**	0.721727	0.397959



4 Features:

Hidden Layer 1 Size	Hidden Layer 2 Size	Cross Entropy Loss - Validation	Misclassification Rate
2	**2**	0.786837	0.421283
2	**3**	0.686279	0.25656
2	**4**	0.682235	0.421283
2	**5**	1.02011	0.696793
3	**2**	0.750255	0.421283
3	**3**	0.700073	0.421283
3	**4**	0.751312	0.653061
3	**5**	0.828053	0.736152
4	**2**	0.833818	0.574344
4	**3**	0.772343	0.319242
4	**4**	0.802469	0.616618
4	**5**	0.724618	0.593294
5	**2**	0.742466	0.421283
5	**3**	0.846551	0.421283
5	**4**	0.817041	0.373178
5	**5**	0.862061	0.696793



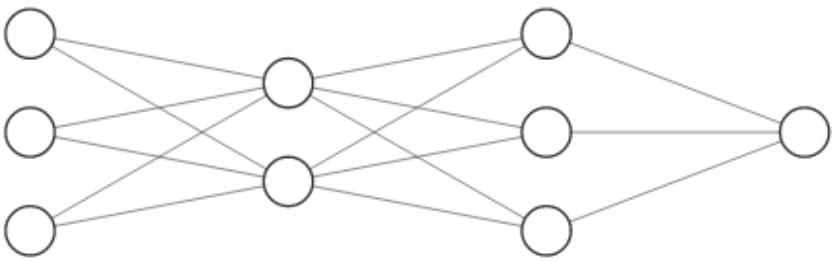
From the results above, the best parameters for the model were determined as:

Number of features: 3

Number of Neurons Hidden Layer 1: 2

Number of Neurons Hidden Layer 1: 3

Misclassification rate: 0.460641



Input Layer $\in \mathbb{R}^3$ Hidden Layer $\in \mathbb{R}^2$ Hidden Layer $\in \mathbb{R}^3$ Output Layer $\in \mathbb{R}^1$

The strategy for selecting pairs was to go across all combinations of numbers, in the range 2-5. Some time was spent experimenting with larger ranges, however, having too many neurons results in severe overfitting and degrading performance. These pairs were sufficient as when we start to increase the number of nodes, the optimal solution for each number of features doesn't change. Some interesting behaviour was noticed as well, at 2 and 4 features, we don't see any changes across the epochs. Furthermore, for 4 features, we see that there is a slight upward slope to the error lines, which indicates that the learning rate is too high. The learning rate choice was made with the assumption that the good performance of the current optimal model, is better than the worst-case performance of training a model with a learning rate too high. In other words, the optimal model we have chosen is much better than anything we had built with a lower learning rate, thus making this trade-off worth it. As for improving this model, if given more time, I believe that It would be beneficial to explore different activation functions as well as loss functions to better quantify the performance of different models.

References:

Loy, J. (2020, March 4). How to build your own neural network from scratch in Python. Medium. Retrieved December 8, 2021, from https://towardsdatascience.com/how-to-build-your-own-neural-network-from-scratch-in-python-68998a08e4f6.