Assignment 1

1 `perform` uses the chosen network's default performance function, in this case `net1.performFcn`, i.e., the network's error function. In this case, the network is using `mse`, or the 'Mean squared error'. To calculate the MSE, the output for each input pattern is subtracted from the target output. These differences are then squared, added up, and then divided by the total number of outputs to get the MSE. The MSE is one way of measuring how different the actual output is from the target output.

2 MSE = 0.0479

3 MAE (mean absolute error) = 0

4 MAE = 0; MSE = 0

5 The perceptron classified all the words correctly.

6 The perceptron performed 3 iterations.

7 MAE=MSE=0.1667

8 **'dolphin'**: the only input it has in the 16 selected features, 'beh_-_eats', was weighted 0 (which makes sense, since all animals eat, and therefore its distribution should not be correlated with being a bird or mammal), so its $\Sigma x_i w_i=0$ (where x=input and w=weight).
'platypus': the only feature it has in the table is 'has a beak', which is normally associated with birds, so it makes sense for it to be grouped with birds rather than mammals, as usually this is a good distinctive feature for identifying birds.
'raccoon' and **'beaver'**: Both have one feature weighted 0 ('is_small' and 'is_brown' respectively), which gives rise to the same problem as 'beh_-_eats' for 'dolphin'. The non-zero weighted feature they share is 'has_a_tail' , and as all the animals in the input have a tail, this should not be correlated with either birds or mammals.
'bat': the two rated features are 'has_fur' and 'beh_-_flies'. The 'has_fur' is negatively weighted and 'beh_-_flies' positively weighted (since it is a feature that is very correlated with birds when comparing birds and mammals), but the product of the input and weight for 'beh_-_flies' is greater than the product of the input and weight of 'has_fur' is small.

9 MSE(training) = 0.018609
MSE(test) = 0.086455
The neural network miscategorised no training words and only 'platypus' and 'bat' in the test words, performing better than the perceptron. While the MSE for the training data was greater than for the perceptron (where it was 0), the neural network still classified all the training data correctly. With the test data, the NN only miscategorised two animals (less than the perceptron), which were also the hardest two (as explained in **8**: both 'has_a_beak' and 'beh_-_flies' are common characteristics of birds).

10 Unlike the perceptron, the results are different every time, and the process by which the final weights were reached is different, as can be evidenced by the corresponding graphs having a variety of patterns (they all shared a negative exponential-shaped curve, but some were more steep than others, and some had bumps). In a few iterations, the NN got up to two of the training inputs wrong (once 'peacock and 'turkey'; once 'bison' and once just 'turkey'). The performance on test data varied between 2 and 5 wrong (usually including 'platypus' and 'bat'). MSE does not seem to be a good measurement of correct categorisation, since, as mentioned above, there are cases where the MSE is non-zero and yet everything is categorised correctly, and, for example, in one iteration it had an MSE of 0.086455 and miscategorised two animals, while in another it had an MSE of 0.078141 an miscategorised four. This shows that for a NN, the "error" is not always correlated with mistakes in categorisation, which is unintuitive.

11 Changing the parameters individually varied in amount of effect: changing the **hidden layers and units** did little for the most part (but changing the number of units per layer to a low number, e.g. 1, made the NN perform worse no matter how many hidden layers there were, and a high number, e.g. ≥100, slightly improved it, especially in terms of the training examples); changing the **learning rate** mostly influenced the initial learning speed (though for very low values, i.e. <0.01, the 1000 iterations is not enough to learn enough to categorise properly); changing the **stopping criterion** made the NN more or less accurate with its categorisations (for high values, many miscategorisations, and below a certain level they performed as if it had been set to 0).

A high LR and SC (stopping criterion) leads to very unpredictable results with a lot of miscategorisation, and a high LR makes any network with hidden units less effective at categorisation.

Best Paramaters: A high number of units and/or layers does make it less likely for the training input to be miscategorised (as long as the LR is ≤0.1, and as mentioned above, the SC should be 0), but it does not seem like the categorisation of test input can be improved much compared to the default, though with the aforementioned parameters, there may be a slight improvement.
Running a network with `lr = 0, hidLayerSpec = [100]` and `goal = 0` produced the best results, never miscategorising more than 3 animals (including 'bat' and 'platypus').