# HW 6 Report

**1. Perceptron Convergence:**

Exercise 5(a):

Weight vector at the end of each epoch:

Epoch 1: [-1. -1.5 0. 1.5]

Epoch 2: [ 0. -1.5 1. 1.5]

Epoch 3: [ 0. -1.5 1. 1.5]

Epoch 4: [ 0. -1.5 1. 1.5]

Epoch 5: [ 0. -1.5 1. 1.5]

Epoch 6: [ 0. -1.5 1. 1.5]

Epoch 7: [ 0. -1.5 1. 1.5]

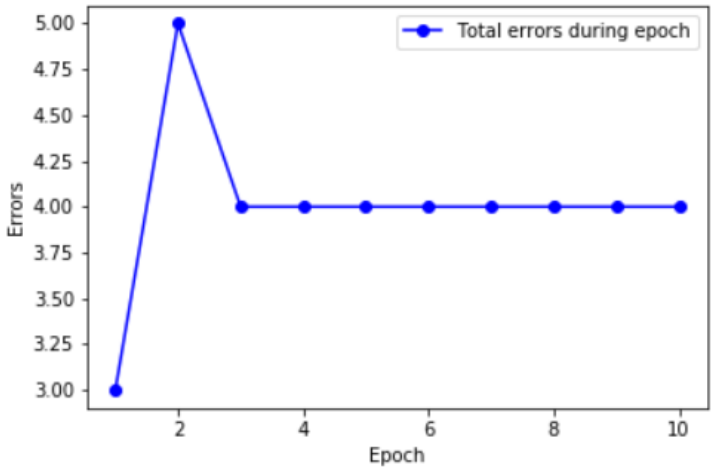
Epoch 8: [ 0. -1.5 1. 1.5]

Epoch 9: [ 0. -1.5 1. 1.5]

Epoch 10: [ 0. -1.5 1. 1.5]

The perceptron algorithm implementation was used to train the dataset from Exercise 5 of homework 5 for 10 epochs. As seen above, the weight vector returned by the algorithm from epoch 2 onwards is the same, proving that it will not converge.

From epoch 3 onwards, the number of errors during each epoch is the same, showing that there is no improvement and that it will run forever. The plot of errors during epoch is saved in '**perceptron_5a.png**' and is shown below:



Exercise 5(b):

Kernel perceptron converged after 5 epochs. This is expected as the larger feature map of the quadratic kernel used by the perceptron has enabled the data labels to be separable in its feature space.

Final values of dual parameters: [5. 1. 3. 0. 4. 0. 4. 0.]

Dual parameters at the end of each epoch:

Epoch 1: [1. 0. 0. 0. 1. 0. 1. 0.]

Epoch 2: [2. 1. 1. 0. 2. 0. 2. 0.]

Epoch 3: [3. 1. 2. 0. 3. 0. 3. 0.]

Epoch 4: [4. 1. 3. 0. 4. 0. 4. 0.]

Epoch 5: [5. 1. 3. 0. 4. 0. 4. 0.]

Epoch 6: [5. 1. 3. 0. 4. 0. 4. 0.]

As seen above, the dual parameters remain the same from epoch 5 to epoch 6, showing no further errors were made in epoch 6 and therefore the data converged. This is shown below.

Number of errors during each epoch:

Epoch 1: 3

Epoch 2: 5

Epoch 3: 4

Epoch 4: 4

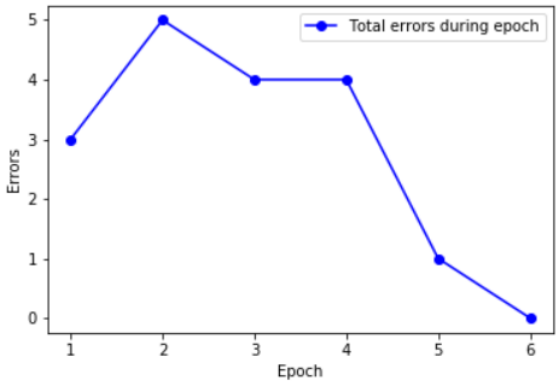
Epoch 5: 1

Epoch 6: 0

Total number of mistakes: 17

Total number of mistakes is given by the sum of the final dual parameter values. This is because we initialize each dual parameter to 0, and add 1 to it every time a mistake is made on the corresponding training example.

The plot of errors during epoch is saved in '**kperceptron_5b**.png' and is shown below:




**2. Spam vs. Non-Spam:**

Perceptron algorithm:

Converged after 18 epochs.

Number of mistakes during each epoch:

Epoch 1: 235

Epoch 2: 64

Epoch 3: 36

Epoch 4: 23

Epoch 5: 16

Epoch 6: 6

Epoch 7: 6

Epoch 8: 11

Epoch 9: 15

Epoch 10: 25

Epoch 11: 11

Epoch 12: 10

Epoch 13: 4

Epoch 14: 3

Epoch 15: 3

Epoch 16: 5

Epoch 17: 2

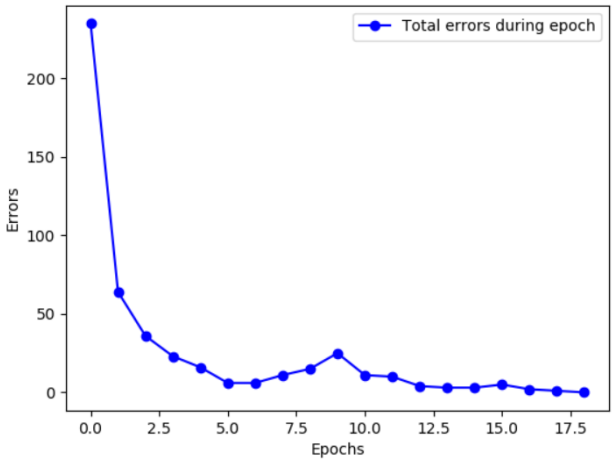
Epoch 18: 1

Epoch 19: 0

Total number of mistakes: 476

Accuracy on test data: 97.8%.

The plot of errors during epoch is saved in '**perceptron_train.png**' and is shown below.




Average Perceptron algorithm:

Converged after 18 epochs.

Number of mistakes during each epoch:

Epoch 1: 235

Epoch 2: 64

Epoch 3: 36

Epoch 4: 23

Epoch 5: 16

Epoch 6: 6

Epoch 7: 6

Epoch 8: 11

Epoch 9: 15

Epoch 10: 25

Epoch 11: 11

Epoch 12: 10

Epoch 13: 4

Epoch 14: 3

Epoch 15: 3

Epoch 16: 5

Epoch 17: 2

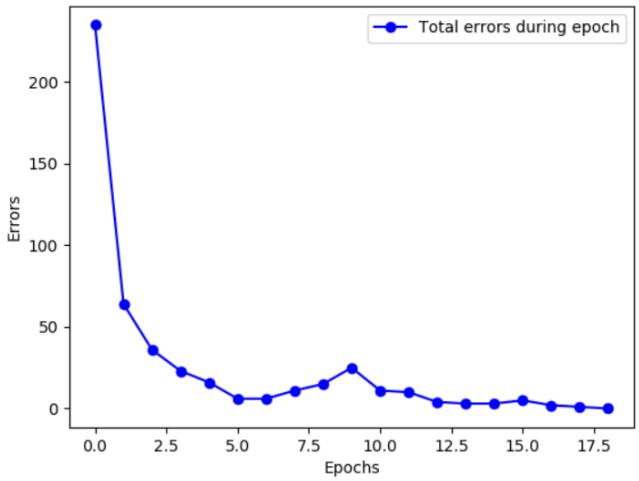
Epoch 18: 1

Epoch 19: 0

Total number of mistakes: 476

Accuracy on test data: 98.1%.

The plot of errors during epoch is saved in '**aperceptron_train.png**' and is shown below.




Both algorithms converge after the same number of epochs and have the same error statistics. This is expected, as the average perceptron only differs in that it returns the average of the weight vector updates computed by the perceptron. The average perceptron performs slightly better in terms of test accuracy because it improves generalization over all training examples.

**3. Atheism vs. Religion:**

Accuracy of perceptron on test dataset version 1: 59.474%.

Accuracy of perceptron on test dataset version 2: 59.298%.

Accuracy of average perceptron on test dataset version 1: 60.877%.

Accuracy of average perceptron on test dataset version 2: 61.579%.

Once again, we observe that the average perceptron performs bit better because it accounts for general performance over all training examples. Test performance on both versions of the test dataset is similar, which can indicate that the frequency with which tokens appear in a document may not be too important.

**4.** The perceptron algorithms do not converge on the 'Atheism vs. Religion' datasets, hence it does not perfectly classify the training examples and inductively performs much worse on the test dataset as compared to the test accuracies for the 'Spam vs. Non-Spam' dataset, for which the algorithms do converge. The training on the 'Spam vs. Non-Spam' dataset is also done on a greater number of examples as compared to the 'Atheism vs. Religion' datasets, which may also have helped improve its accuracy. However, Atheism and religion are closely related topics, hence it is likely that the same terms are used in both types of postings, which is why it can be difficult to distinguish between the two based on term usage. On the other hand, term usage in case of spam email vs. non-spam email is more clearly distinguishable. I believe this is ultimately the reason why the perceptron accuracy is much lower in case of the 'Atheism vs. Religion' datasets and why it does not converge even after 10000 epochs; as it attempts to classify examples based on terms (or tokens) used.