16ML Perceptron Quiz

1. What assumptions do we make on the dataset when training a perceptron?
   (a) Data is linearly separable
   (b) Output labels are binary
   (c) There are more input data points than outputs
   (d) Input data is in two dimensions
2. Suppose we are training a binary perceptron with a learning rate of $r = 1$. How is our update algorithm affected by changing r to $r = 0.5$?
3. Once a perceptron is trained, how are the weights used to classify new points?
4. What can we say about the optimally of the weights updated by the Perceptron Learning Algorithm?
5. While looking at examples of perceptrons, Nicholas found that in one instance all inputs had an extra 1 appended to them.

$$\mathbf{x}_i = \begin{bmatrix} x_1 \\ x_2 \\ \dots \\ x_n \\ 1 \end{bmatrix}$$

   What is the significance of this?
6. What is the point of the activation function? Is it unique for the Binary Perceptron?
7. When does the Perceptron Learning Algorithm halt? Does it always converge?
8. Explain the intuition behind the update step from the perspective of gradient descent
9. Suppose we trained a binary perceptron on a set of training data and received a weight vector

$$\mathbf{w} = \begin{bmatrix} 1.3 \\ 2.2 \\ -2.8 \end{bmatrix}$$

If we have some point

$$\mathbf{x}_i = \begin{bmatrix} -1.2 \\ 0.63 \\ 0.09 \end{bmatrix}$$

what label does our perceptron classify it as?

10. What needs to change to extend the Perceptron Learning Algorithm to account for multiple classes?

11. We will walk through training a multi-class perceptron by hand. Assume we have input vectors $\mathbf{x} \epsilon R^2$ and four labels $1, 2, 3, 4$. We randomly initialize the weights

$$\mathbf{w}_1 = \begin{bmatrix} 0.30 \\ 0.22 \end{bmatrix}, \mathbf{w}_2 = \begin{bmatrix} 0.71 \\ -0.48 \end{bmatrix}, \mathbf{w}_3 = \begin{bmatrix} -0.54 \\ 0.13 \end{bmatrix}, \mathbf{w}_4 = \begin{bmatrix} .83 \\ 0.35 \end{bmatrix}$$

Remember that we have a weight vector for each possible label. Additionally, the shape of the weight vector is the same as the weight of the input because we want to compute their dot product. Assume a learning rate of $r = 1$

(a) We receive a new input

$$\mathbf{x}_i = \begin{bmatrix} 2.1 \\ 0.3 \end{bmatrix}$$

with label $y_i = 3$. Calculate the dot product between each weight vector and the input $\mathbf{x}_i^T \mathbf{w}_j$

(b) What class does the current weight scheme classify the point as? Is it the correct label?

(c) Report the updated weights based using the Multiclass Perceptron Learning Algorithm

(d) We receive another input

$$\mathbf{x}_i = \begin{bmatrix} -0.3 \\ -2 \end{bmatrix}$$

with label $y_i = 2$. Using the updated weights from the previous part, calculate the dot product between each weight vector and the input $\mathbf{x}_i^T \mathbf{w}_j$

(e) What class does the current weight scheme classify the point as? Is it the correct label?

(f) Report the updated weights based using the Multiclass Perceptron Learning Algorithm