Perceptron Quiz

Daniel Abraham, Ganesh Vurimi,
Matthew Bronars, Warren Deng, Rasika Iyer

October 2020

1 Questions

1. What assumptions do we make on the dataset when training a perceptron?
2. What's the point of the learning rate in the Perceptron Learning Algorithm?
3. How are the weights used in the perceptron?
4. Does the Perceptron Learning Algorithm always give the optimal weights for a given dataset?
5. How do we account for separating hyperplanes that do not cross the origin?
6. What is the point of the activation function?
7. Is convergence of the Perceptron Learning Algorithm guaranteed?
8. What needs to happen in the Perceptron Learning Algorithm for the algorithm to end?
9. What needs to change to extend the Perceptron Learning Algorithm to account for multiple classes?
10. What's the intuition behind the update step for weights?

2 Answers

1. We assume that the data is linearly separable.
2. The learning rate is a hyper parameter that allows us to control how much each misclassified training point affects the old weight. It is used during the update stage by scaling how much we add to the old weight.
3. There is one weight for every input (and possible an extra weight if we include the bias term). We put the weights into a vector and take the dot product of this weight vector with the input. This is our weighted sum. By using the activation function on this weighted sum, we can get the predicted output of our input.
4. No. There is no guarantee on the optimality of the solution. The only thing that is guaranteed is convergence of the Learning Algorithm if we have linearly separable data.
5. We amend a constant term to every input and add an extra weight variable to the weight vector. This way when we take the dot product of the weight vector and the input vector, we can learn an extra scalar which in 2-d represents where the decision boundary crosses the y-axis.

6. The activation function allows us to classify the weighted sum by sorting it into one of two classes. We can use various activation functions such as the step function, but in the note we use the sign function in our derivations.

7. Only when we have linearly separable data is the convergence guaranteed.

8. In order for the Perceptron Learning Algorithm to stop iterating, there needs to be one round where every training point is correctly classified.

9. To extend the algorithm, we add a weight vector for each class. Then, in order to classify some input, we take the argmax of the dot product of every weight vector and the input. We also change the update algorithm slightly by adding to the correct weight vector when we misclassify and subtracting from the incorrect ones.

10. We can view our problem as an optimization problem where we are trying to minimize the amount of misclassified points. Our update step is traveling in the negative gradient of the cost function of this optimization problem.