CS 4700: Foundations of Artificial Intelligence
Spring 2020
Quiz 4 Review Questions

1. True/False: A perceptron is guaranteed to learn any set of training data given a suitable learning rate.

2. Consider a supervised learning problem with N examples where each is a point in d-dimensional space. Imagine we apply the perceptron learning algorithm on the data and it converges (finds a perceptron that separates the data) after t iterations. What is the runtime of this algorithm as a function of N, d, and t?

3. Consider a supervised learning problem with only 2 examples where each is a point in 5-dimensional space. The first example is (1,5,2 7,9) and has a label of 1. The second is (-3,-8,2,4,6) and has a label of 0. Give two distinct perceptrons that would correctly label these two points. (When I say to give a perceptron that means give the list of weights that define that perceptron.)

4. There are 2^(2^n) distinct Boolean functions over n inputs. Thus there are 16 distinct Boolean functions over 2 inputs. How many of these are representable by a perceptron?

5. Imagine you trained a perceptron and after learning was completed you discovered that the data were all labeled backwards: Every example that was labeled 1 should have been a 0, and vice versa. Sadly, you no longer have the data and have no ability to change the code that uses the perceptron to flip its answers. All you have access to are the weights of the perceptron. How would you change the weights in order to flip all the answers? You may assume that there are no data points that fall exactly on the boundary between the two categories.

6. Consider a learning problem where each example has n attributes x_1, …, x_n with each x_i taking on the value of either 0 or 1. Give a Perceptron that returns 1 if more of the x_i's have value 0 than value 1 and otherwise returns 0.

Solutions

1. True/False: A perceptron is guaranteed to learn any set of training data given a suitable learning rate.
   False. If the data are not separable then it can't be learned regardless of the learning rate.

2. Consider a supervised learning problem with N examples where each is a point in d-dimensional space. Imagine we apply the perceptron learning algorithm on the data and it converges (finds a perceptron that separates the data) after t iterations. What is the runtime of this algorithm as a function of N, d, and t?
   It's N(d+1)t – it's not n but d+1 because of the extra x_0 that's added to each example.

3. Consider a supervised learning problem with only 2 examples where each is a point in 5-dimensional space. The first example is (1,5,2 7,9) and has a label of 1. The second is (-3,-8,2,4,6) and has a label of 0. Give two distinct perceptrons that would correctly label these two points. (When I say to give a perceptron that means give the list of weights that define that perceptron.)
   There are an infinite number of such perceptrons. Each correct answer has the following property: the dot product of the weights w and (1,1,5,2,7,9) is ≥ 0 and the dot product of w and (1,-3,-8,2,4,6) is <0.
   Note that this is a problem that did not ask you to run the perceptron learning algorithm. You just needed to get weights that work. Thus, for example, $-x_0+x_1 \geq 0$ (where w = (-1,1,0,0,0,0)) and $-x_0+x_2 \geq 0$ (where w = (-1,0,1,0,0,0)). These are simply saying you can separate them by asking is $x_1 \geq 1$ and is $x_2 \geq 1$.

4. There are 2^(2^n) distinct Boolean functions over n inputs. Thus there are 16 distinct Boolean functions over 2 inputs. How many of these are representable by a perceptron?
   There are 14. There are four combinations of inputs, {(0,0), (0,1), (1,0), (1,1)}. For each of the four there is a perceptron that labels only that point as 1 and labels the others 0. There is similarly one that labels just that item as 0 and the others 1. This totals 8. Next, there's a perceptron that ignores input 2 and labels as 1 both points for which input 1 equals 1 and labels the other two points as 0, and there is also one that labels those points in reverse. That's 2 more. Finally there's an identical pair of perceptrons that ignore input 1 and does the same as the previous case, only this time using input 2. Finally, there are also the "trivial" cases of all data points being labeled 1, or all being labeled 0, which are also representable as linear thresholds for which all points are on the same desired side of a linear surface.

5. Imagine you trained a perceptron and after learning was completed you discovered that the data were all labeled backwards: Every example that was labeled 1 should have been a 0, and vice versa. Sadly, you no longer have the data and have no ability to change the code that uses the perceptron to flip its answers. All you have access to are the weights of the perceptron. How would you change the weights in order to flip all the answers? You may assume that there are no data points that fall exactly on the boundary between the two categories.
   Here are two different approaches:
   - You could negate all the weights. That would flip the labels of everything (except points that are on the decision surface, which you were told you can ignore).

- You could use the old perceptron's weights to label a set of randomly generated points, flip all the labels for those points, learn a new perceptron on this new data, then plug its weights into the original perceptron.

6. Consider a learning problem where each example has n attributes $x_1, \ldots, x_n$ with each x_i taking on the value of either 0 or 1. Give a Perceptron that returns 1 if more of the x_i's have value 0 than value 1 and otherwise returns 0.
   We want to return 1 if
   $$\sum_{i=1}^{n} x_i \leq \frac{n-1}{2}$$
   Doing some minor algebra, this becomes:
   $$0 \leq \frac{n-1}{2} - \sum_{i=1}^{n} x_i = \frac{n-1}{2} + \sum_{i=1}^{n} (-1) \cdot x_i$$
   This is the form of a Perceptron where $w_1 = \cdots = w_n = -1$ and $w_0 = \frac{n-1}{2}$.