Problem Set 1 Solutions

This document serves as a guideline to find answers to the problem set.

1 Refer to Mitchell Section 6.5 and 6.5.1.

2 Consider the truth table for $Y = A \wedge \neg B$.

A	B	Y
0	0	0
0	1	0
1	0	1
1	1	0

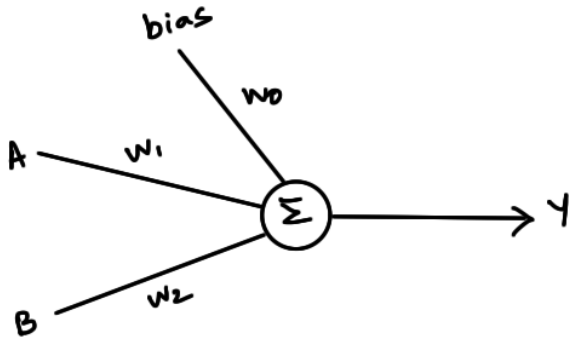

Figure 1: Linear Perceptron

Looking at the truth table and the perceptron, we can infer that,

$$\begin{aligned} w_0 &> 0 \\ w_1 &> w_0 \\ w_1 + w_2 &< w_0 \end{aligned}$$

We can clearly see that w_2 has to be -ve. Any combination of weights that satisfy these conditions will model Y.

For XOR, refer to this quiz.

3 Refer Mitchell Section 4.2. The perceptron training rule is given by

$$\Delta w_i = \eta(t - o)x_i \tag{1}$$

And the gradient descent training rule is given by

$$\Delta w_i = -\eta \frac{\delta E}{\delta w_i} \tag{2}$$

Perceptron training rule works when the data is linearly seperable. When it is not, we use gradient descent. It converges toward a best-fit approximation to the target concept.

As you can see for the current example the output o is given by $o = w_0 + w_1 x_1 + w_1 x_1^2 + + w_n x_n + w_n x_n^2$. The perceptron models a non-linear function. The gradient descent rule can find an optimal function.

4 When decision trees are used to perform regression, they are called regression trees.

Nodes with one data point cannot be split, and also creating a node for each data point is unlikely to generalize well. In decision tree classification we use criterions like Mutual Information and GINI Index to find out the right split of data points. In regression tasks we will use the sum of squared errors. The sum of squared errors for a tree is,

$$S = \sum_{c \in leaves(T)} \sum_{i \in c} (y_i - k_c)^2 \tag{3}$$

where k_c is a constant function and it is the best estimate all points in the leaf. We can show that this estimate is the mean when we use the sum of squared errors. Refer to this class video for proof. The basic regression algorithm will work as follows:

1. Start with a single node containing all points. Calculate the mean k_c and S.
2. Find the next best split, that will reduce S as much as possible. If you reach a stopping criterion, then stop.
3. For every new node, go to step 1.

Here stopping criterion could be specified by specifying a lower bound on the number of data points in a leaf or specifying an upper bound on the error in each leaf.

There are many implementations for regression trees available. Here is a good example using scikit-learn. You can use it to find a regression tree for the housing dataset.

5 Refer Lazy Decision Trees by Friedman et al.

6 Nearest neighbor. Decision trees would have to approximate the line with the best staircase-shaped decision surface that it could form. Nearest neighbor could represent the line with only two points, and hopefully would fnd a decision surface that is plausible given a reasonable sample of training data.

7

1. Let us start by considering two points. Two points can be labelled in 4 possible ways. We can always find a circle that shatters these two points. So the VC dimension is atleast 2.

   Let us now consider three points. In this case if the nearest and furthest point has the same label and the remaining point has the opposite label, you can't draw a circle to shatter this instance space.

   So the VC dimension is 2.

2. We use the same argument as above to prove that the VC dimension is 2.

   (Will our answers change if the cricle and sphere was not origin centered? Hint: Yes.)