José Tristani | jtristani3

Charles Isbell

CS-7641: Machine Learning

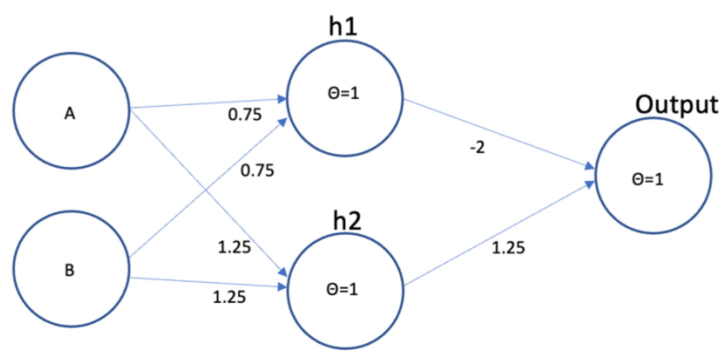
September 22, 2021

Problem Set #1:

1. Design a two-input perceptron that implements the Boolean function $A \wedge \neg B$. Design a two-layer network of perceptrons that implements $A \oplus B$ (where $\oplus$ is XOR).

   With this 2 layer network, we can achieve a XOR logic using specifically selected weights to get the desired output with thresholds of 1 on all perceptrons. The weights can be slightly shifted moving the boundary layer for the 2 line functions that the networks represents, as long as the 0,1 and 1,0 points are above the Outputs threshold.

A=0|B=0
H1=(0*.75+0*0.75)=0
H2=(0*1.25+0*1.25)=0
Output=(0*-2+0*1.25)=0
A=0|B=1
H1=(0*.75+1*0.75)=0.75
H2=(0*1.25+1*1.25)=1.25
Output=(0*-2+1*1.25)=1.25
A=1|B=1
H1=(1*.75+1*0.75)=1.50
H2=(1*1.25+1*1.25)=2.50
Output=(1*-2+1*1.25)=-0.75
A=1|B=0
H1=(1*.75+1*0.75)=1.50
H2=(1*1.25+1*1.25)=2.50
Output=(1*-2+1*1.25)=-0.75



2. Suggest a lazy version of the eager decision tree learning algorithm ID3. What are the advantages and disadvantages of your lazy algorithm compared to the original eager algorithm?

   One way to create a lazy version would be to calculate the standard deviation of the data and using this create the tree until the edges are within a set std that we choose, this could be a hyperparameter on how lazy we want the built tree to be. This would work better for regression, in my head its similar to discretizing values.

3. Imagine you had a learning problem with an instance space of points on the plane and a target function that you knew took the form of a line on the plane where all points on one side of the line are positive and all those on the other are negative. If you were constrained to only use decision tree or nearest-neighbor learning, which would you use? Why?

   Decision Trees would be the better classifier, why? Mostly because a decision tree can split the data using a function, in the instance of an Oblique Decision Tree, although more expensive to compute, the boundary would represent the line on the plane to split the data perfectly. Another case for not using K-NN would be that the boundaries for KNN are not as precise as a line on a plane, points close to the boundary could be mislabeled. Even classical decision trees would make a better classifier.

4. . Give the VC dimension of these hypothesis spaces, briefly explaining your answers:
   a. An origin-centered circle (2D)
      The VC dimension is 2, because with sets of 3 points, there is no way to label 2 points without labeling the remaining point.
   b. An origin-centered sphere (3D)
      The same applies; the VC dimension is 2, because with sets of 3 points, there is no way to label 2 points without labeling the remaining point.