CMSC422 Machine Learning Homework 3 Winter 2018

Due Date: Wednesday, November 7th.

1. Suppose our hypothesis class is not one rectangle but a union of two (or $m > 1$) rectangles. What is the advantage of such a hypothesis class? Show that any class can be represented by such a hypothesis class with a large enough m.

2. VC-dimension.

   (a) Show that the VC dimension of the hypothesis class of triangles is 7 in two dimensions. Hint: for best separation, it is best to place the 7 points equidistant on a circle. Prove your claim.

   (b) Union of Intervals. What is the VC-dimension of subsets of the real number line formed by the union of k intervals? Prove your claim.

   (c) We define a set of concepts $H = \{\text{sign}(ax^2 + bx + c); a, b, c, \in R\}$, where $\text{sign}(\cdot)$ is 1 when the argument $\cdot$ is positive, and 0 otherwise. What is the VC dimension of H? Prove your claim.

3. Perceptrons.

   (a) Show/draw a two-input perceptron that implements the Boolean function $a \wedge \neg b$.

   (b) Show/draw a perceptron that calculates NAND of its two inputs.

   (c) Show/draw a perceptron that calculates parity its 5 inputs.

   (d) Consider training a two-input perceptron. Give an upper bound on the number of training examples sufficient to assure with 90% confidence that the learned perceptron will have true error of at most 5%. Does this bound seem realistic?

4. Consider a two-layer network function of the form shown below in which the hidden unit nonlinear activation functions $g(\cdot)$ are given by sigmoid functions of the form $\sigma(a) = (1 + e^a)^{-1}$.

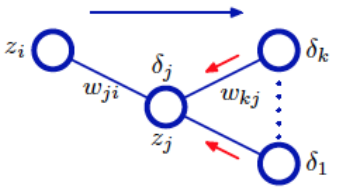
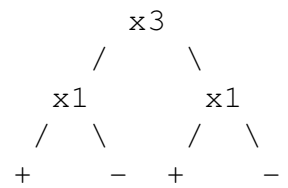


Illustration of the calculation of δ_j for hidden unit j by backpropagation of the δ's from those units k to which unit j sends connections. The blue arrow denotes the direction of information flow during forward propagation, and the red arrows indicate the backward propagation of error information.

Show that there exists an equivalent network, which computes exactly the same function, but with hidden unit activation functions given by $\tanh(a)$. Hint: first find the relation between $\sigma(a)$ and $\tanh(a)$, and then show that the parameters of the two networks differ by linear transformations.

5. The nearest-neighbor classifier assigns a new input vector $\vec{x}$ to the same class as that of the nearest input vector $\vec{x}_n$ from the training set, where in the simplest case, the distance is defined by the Euclidean metric $\| \vec{x} - \vec{x}_n \|^2$. By expressing this rule in terms of scalar products and then making use of kernel substitution, formulate the nearest-neighbor classifier for a general nonlinear kernel.

6. Consider the hypothesis class $H_{r\ d2}$ of "regular, depth-2 decision trees" over n Boolean variables. A "regular, depth-2 decision tree" is a depth-2 decision tree (a tree with four leaves, all distance 2 from the root) in which the left and right child of the root are *required to contain the same variable*. For instance, the following tree is a member of $H_{r\ d2}$:



   (a) As a function of n, how many syntactically distinct trees are there in $H_{r\ d2}$?

   (b) Give a tight upper bound for the number of examples needed in the PAC model to learn $H_{r\ d2}$ with error ϵ and confidence δ.

7. Consider an image classification problem. Suppose an algorithm first splits each image into $n = 4$ blocks (the blocks are non-overlapping and each block is at the same location and of constant size across all images) and computes some scalar feature value for each of the blocks (e.g., average intensity of the pixels within the block). Suppose that this feature is discrete and can take $m = 10$ values. The classification function classifies an image as 1 whenever each of the n feature values lies within some interval that is specific to this feature (i.e., the value of the first feature is between a_1 and b_1, the value of the second feature is between a_2 and b_2, and so on), and 0 otherwise. We would like to learn these intervals (a and b values for each interval) automatically based on a training set of images. All the other parameters such as locations and sizes of the blocks are not being learned. The following questions are helpful in understanding the requirements on the size of the training set.

   (a) What is the size of the hypothesis space H? Assume that only intervals with $a_i \leq b_i$ are considered for learning.

   (b) Assuming noiseless data and that the function we are trying to learn is capable of perfect classification, give an upper bound on the size of the training set required.

8. Kernels.

   (a) Prove or Disprove: The composition of two kernel functions is also a kernel function.

   (b) Given two examples $\vec{x} \in \Re^2$ and $\vec{z} \in \Re^2$, let
   $K(\vec{x}, \vec{z}) = (\vec{x}^T \vec{z})^3 + 49(\vec{x}^T \vec{z} + 4)^2 + 64\vec{x}^T \vec{z}$
   Prove that K as defined above is a valid kernel function.

9. SVMs. We have a set of six labeled examples D in the two-dimensional space, $D = \{\langle x^{(1)}, y^{(1)}\rangle, \langle x^{(2)}, y^{(2)}\rangle \ldots \langle x^{(6)}, y^{(6)}\rangle\}$, with $x^{(i)}, y^{(i)} \in \Re^2$ and $z^{(i)} \in \{1, -1\}$ for $i = 1 \ldots 6$ as listed below:

i	$x^{(i)}$	$y^{(i)}$	$z^{(i)}$
1	-2	0	1
2	-2.4	-1.6	1
3	1.3	2.6	-1
4	-0.3	-2.5	1
5	3	0.2	-1
6	0	2	-1

(a) Draw/sketch/graph the values.

(b) We want to find a linear classifier where examples $\vec{x}$ are positive if and only if $\vec{w} \cdot \vec{x} + \theta \geq 0$

i. Find an easy solution to $(\vec{w}, \theta)$ that can separate the positive and the negative examples given.

ii. Which of the six vectors would be the support vectors?

iii. Bonus: If you solved the optimization problem, what would the solution to $(\vec{w}, \theta)$ be?

10. **Ethics Question:** (probably requiring between one paragraph and one page to answer properly). Consider the well known philosophical conundrum of the Trolley Car Problem (`https://en.wikipedi` which goes like this:

    > You see a runaway trolley moving toward five tied-up (or otherwise incapacitated) people lying on the tracks. You are standing next to a lever that controls a switch. If you pull the lever, the trolley will be redirected onto a side track and the five people on the main track will be saved. However, there is a single person lying on the side track. You have two options:
    >
    > (a) Do nothing and allow the trolley to kill the five people on the main track.
    >
    > (b) Pull the lever, diverting the trolley onto the side track where it will kill one person.

    One could imagine various versions of this problem that flow from a system based on the learning algorithms we've studied to date. For example,

    (a) Most obviously, the sensor-actuation loop in an automated vehicle: what does the vehicle do if all sensors show critical obstacles? i.e., there is no "turn to avoid", as every action seems to hit a pedestrian.

    (b) In designing a neural network for automated analysis of medical records one must balance the need for unnecessary (and potentially risky) procedures vs doing nothing and missing a critical case.

    Other examples of variations on the Trolley Car Problem abound.

    As a software developer, your code (neural network, SVM or other machine learning technique) may be an element in a larger autonomous system (automated vehicle, package delivery drone, medical information system) that might, in the course of its natural operation, have to face an instance of the Trolley Car Problem. Comment on the following issues: What level of responsibly is borne by you, as the programmer? What can you do (if anything), at the level of your component, to manage this responsibly? Is an approach based on machine learning any different than a more traditional software engineered system? If so, in what way is it different? Feel free to comment on any other social, technological or philosophical issues you deem relevant to this question.