EECS 440: Machine Learning Fall 2020 Written Problems Week 6 due 10/16 11:59pm

General Instructions: Write or type your answers neatly and remember to show all relevant work. All questions are worth 10 points. Each answer should be a separate pdf, and you can turn in the pdfs on canvas in the appropriate assignment. Some questions may be very challenging; significant partial credit is available for reasonable attempts at solutions. Since each question is worth the same number of points, do not waste too much time on any one. Ask me or the TAs for help if stuck.

Some of the questions require you to write short programs to simulate things. You can use any language/software to do this, and you do not need to turn in your code.

Upload your answers to Canvas as a pdf file by 11:59pm on the due date specified after the question. You will receive a 10% bonus for a solution turned in a week or more in advance of the due date. You can use one late day each week (up to Saturday 11:59pm) with a penalty of 20%. Submissions after Saturday 11:59pm for any week will not be graded.

Each group must do their own work. Only one submission is needed from each group. Do not use any source other than the lecture notes, textbook(s) and readings on the class website to answer these questions. Only those who contributed equally to a submission should have their names and Case IDs on the submission. Those not listed as contributing will not receive points.

23. Using any software or language of your choice, plot the decision boundary for an ANN with two inputs, two hidden units and one output. All activation functions are sigmoids. Each layer is fully connected to the next. Assume the inputs range between −5 to 5 and fix all activation thresholds to 0. Plot the decision boundaries for the weights except the thresholds randomly chosen between (i) (−10,10), (ii) (−3,3), (iii) (−0.1,0.1) (one random set for each case is enough). Use your plots to show that weight decay can be used to control overfitting for ANNs.
24. Suggest modifications for backpropagation for *non-feedforward* neural network structures if edges are allowed between nodes in the same layer as well as between successive layers, but the graph is still directed acyclic. In other words, nodes in layer k, $x_{k1},x_{k2},...,x_{kn}$ can have edges between them as well as to the k+1 layer, as long as no cycle is created.

The Bayesian Candy Factory makes a Halloween Candy Box that contains a mix of yummy (Y) and crummy (C) candy. You know that each Box is one of three types: 1. 80% Y and 20% C, 2. 55% Y and 45% C and 3. 30% Y and 70% C. You open a Box and start munching candies. Let the i^{th} candy you munch be denoted by c_i. Answer the following questions using a program written in any language of your choice. Generate one Box with 100 candies for each type, and assume a fixed order of munching.

25. For each Box, plot $\Pr(T=i|c_1,\ldots,c_N)$ on a graph where T represents a type and N ranges from 1 to 100. (You should have three graphs and each graph will have three curves.)
26. For each Box, plot $\Pr(c_{N+1}=C|c_1,\ldots,c_N)$ where N ranges from 1 to 99.
27. Suppose this is 2020, so before opening a Box you believe that each Box has 70% crummy candies (type 3) with probability 0.8 and the probability of the other two types is 0.1 each. Replot $\Pr(T=i|c_1,\ldots,c_N)$ taking this belief into account for each of the 3 Boxes. Briefly explain the implications of your results.