EECS 440: Machine Learning Fall 2020 Written Problems Week 5 due 10/9 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.

19. Consider the primal linear program (LP): min $c^T x$ s.t. $Ax \geq b$, $x \geq 0$ and its dual: max $b^T u$ s.t. $A^T u \leq c$, $u \geq 0$. Carefully prove that for any feasible (x,u) (i.e. x and u satisfying the constraints of the two LPs), $b^T u \leq c^T x$.
20. Derive the backpropagation weight updates for hidden-to-output and input-to-hidden weights when the loss function is cross entropy with a weight decay term. Cross entropy is defined as $L(\mathbf{w}) = -\sum y_i \log(o_i) + (1-y_i)\log(1-o_i)$, where y_i is the true label (assumed 0/1) and o_i is the estimated label for the i^{th} example.
21. Draw an artificial neural network structure which can perfectly classify the examples shown in the table below. Treat attributes as continuous. Show all of the weights on the edges. For this problem, assume that the activation functions are sign functions instead of sigmoids. Propagate each example through your network and show that the classification is indeed correct.

x_1	x_2	Class
−4	−4	−
−1	−1	+
1	1	+
4	4	−

22. When learning the weights for the perceptron, we dropped the *sign()* activation function to make the objective smooth. Show that the same strategy does not work for an arbitrary ANN. (Hint: consider the shape of the decision boundary if we did this.)