CSE 417T: Homework 6

Due: December 1 (Friday), 2017 at 11:59PM

Notes:

- Note that the due time is **11:59PM** for this homework. Please plan your schedule accordingly. The reason is (given the two-day rule for late days) that we can safely give you the solution sketches and discuss the homework in the review sessions on December 4/5.
- Please check the submission instructions for Gradescope provided on the course website. You must follow those instructions exactly.
- You have to submit your written solutions tp Gradescope. **Write-ups submitted via SVN will not be accepted or graded.**
- Homework is due **by 11:59 PM on the due date.** Remember that you may not use more than 2 late days on any one homework, and you only have a budget of 5 in total.
- Please keep in mind the collaboration policy as specified in the course syllabus. If you discuss questions with others you **must** write their names on your submission, and if you use any outside resources you **must** reference them. **Do not look at each others' writeups, including code.**
- There are 5 problems on 2 pages in this homework.

Problems:

1. (20 points) Suppose your input data consists of the following (x, y) pairs:

   $(3, 5); (5, 6); (7, 9); (2, 11); (3, 8)$

   What value of y would you predict for a test example where $x = 3.2$ using the 3-nearest neighbors average?

2. (20 points) (From Russell & Norvig) Construct a support vector machine that computes the XOR function. Use values of $+1$ and -1 (instead of 1 and 0) for both inputs and outputs, so that an example looks like $([-1, 1], 1)$ or $([-1, -1], -1)$. Map the input $[x_1, x_2]$ into a space consisting of x_1 and x_1x_2. Draw the four input points in this space, and the maximal margin separator. What is the margin? Now draw the separating line back in the original Euclidean input space.

3. (20 points) The key point of the so-called "kernel trick" in SVMs is to learn a classifier that effectively separates the training data in a higher dimensional space without having to explicitly compute the representation $\Phi(\mathbf{x})$ of every point $\mathbf{x}$ in the original input space. Instead,

all the work is done through the kernel function that computes dot products $K(\mathbf{x}_i, \mathbf{x}_j) = \Phi(\mathbf{x}_i)\Phi_(\mathbf{x}_j)$.

Show how to compute the squared Euclidean distance in the projected space between any two points $\mathbf{x}_i$, $\mathbf{x}_j$ in the original space without explicitly computing the Φ mapping, instead using the kernel function K.

4. (20 points) Create a neural network with only one hidden layer (of any number of units) that implements XOR(AND(x_1, x_2), $\quad x_3 \quad$). Draw your network, and show all weights of each unit.

5. (20 points) (From Russell & Norvig) Suppose that a training set contains only a single example, repeated 100 times. In 80 of the 100 cases, the single output value is 1; in the other 20, it is 0. What will a back- propagation network predict for this example, assuming that it has been trained and reaches a global optimum? (Hint: to find the global optimum, differentiate the error function and set it to zero.)