EECE 5136/6036: Homework 2

Given 2/21/2017, Due 3/7/2017

1. (100 points) A team of sociologists is doing a study of two-income couples living in Silicon Valley to determine a relationship between the incomes of the husband (H) and the wife (W) and their self-assessed level of stress. They have collected data from about 300 couples and classified each one into one of two classes: Stressed (S=1), and Not Stressed (S=0). They now want to come up with a classifier that can predict the stress class of new couples based on H and W.

You will try two different approaches to the task:

- A k-nearest neighbors classifier.
- A single perceptron.

You should write programs that implement each classifier and run them on the data set. *You should use the available data in a way that validates the ability of the classifiers to classify novel data*. Also, think about the numerical ranges of the data to see whether any prior scaling is needed.

For the k-NN classifier, you can try the basic version with different values of k, or try augmented versions with weighted distances or some other augmentation.

You should present your results in a brief report providing the following information:

- A brief description and justification of your overall strategy (i.e., how you use the data for training and testing).

- What k values and/or augmentation you chose for k-NN, and why.

- A graph showing the Hit-Rate performance of the perceptron before training and then every 50 epochs (1 epoch = one training run over all training data points). The graph will have “Epoch” as its x-axis and “Hit-Rate” as its y-axis. You should test performance on both the training set and the test set at each point (though the test set will not be used for training), and plot both curves on the same graph. Comment on the results shown in this graph.

- Two bar graphs, each with 4 bars – one showing the final Sensitivity, Specificity, PPV, and NPV values on the training set after training, and the other showing the same for the test set. If you wish, you can plot the two sets of bars on the same graph, with the two bars for each metric next to each other. Comment on the results shown in these graphs.

- Plots of the data in the H-W plane indicating the decision boundaries found by the classifiers. For the k-NN classifiers, this will involve sampling the H-W plane, and will only give an approximate boundary.

- Your opinion of the pros and cons of the classifiers *based on your experience on this project.*

- Your recommendation for one of the classifiers in each case, with a brief justification.

- Printouts of your program(s).

The data is available on BlackBoard as HW2 _data. If you cannot access it, please send me mail at Ali.Minai@uc.edu.

The report should be very brief – 1-2 printed pages plus the graphs and code.

2 (100 Points). A robot moves around in a 100 × 100 square environment with four walls. At any given time, the robot's sensors can accurately determine the distances *x* and *y* from the west and south walls, respectively. The robot has a head-mounted LED that is required to light up when the robot is in the 20 × 20 square region around the center point of the environment. The LED should be off when the robot is outside this region.

You will design two neural networks based on perceptrons, as follows:

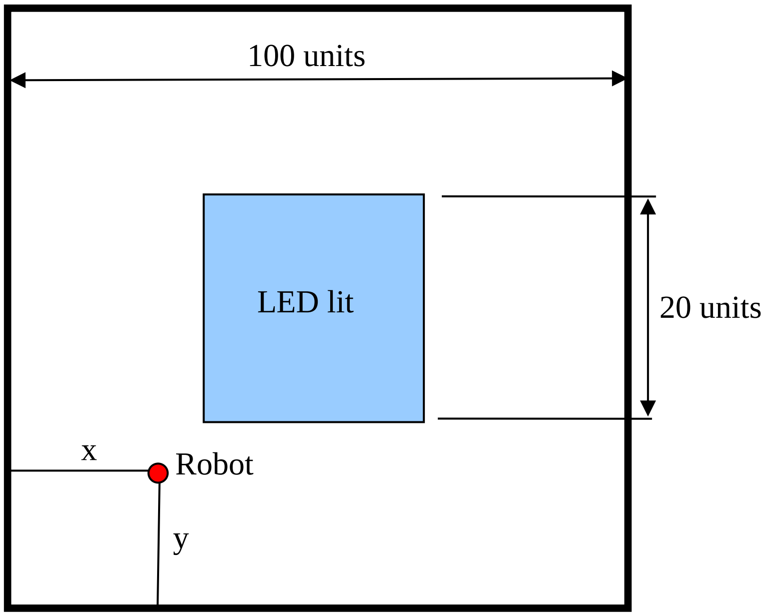
Designed Network: For this network, you should analyze the problem and come up with a network using perceptrons by designing an architecture and calculating the weights of each perceptron explicitly. The output perceptron should provide the on/off signal for the LED. [Hint: Think in terms of thresholds and Boolean functions]

Trained Network: For this case, you will first write a program to simulate random continuous trajectories of the robot moving through the environment and sample (*x*,*y*) and (*LED on*, *LED off*) data from these trajectories. Then, using this data set and the perceptron learning algorithm, you will train a network with the same architecture as the Designed Network. Note that if your network uses multiple layers of perceptrons, each layer will need to be trained separately, and you will need to extract the appropriate training data for each individual perceptron from the sampled dataset.

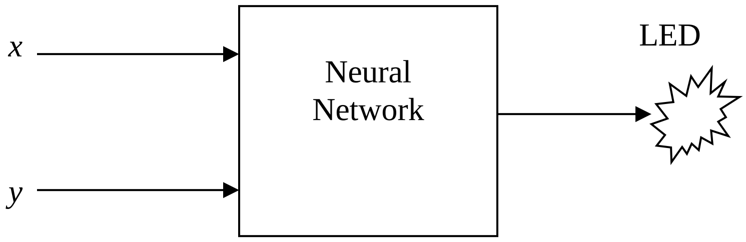
In both networks, each perceptron receiving input of the robot coordinates ***must*** receive input from both *x* and *y* coordinates, not only one. Of course, the weights from either coordinate can take any value, including 0.

This is actually a simple problem, but it should make you think in a neural networks idiom.

Write a brief report describing what you did, why you did it, and showing results to confirm that, after training, the robot indeed does what it is supposed to do. Make sure you do this for ***both*** networks. Compare the results and the weights obtained for the Trained Network with those for the Designed Network.



The system to be designed would have the following architecture:



Points on both questions will be awarded for: 1) Correctness (i.e., do the implementations work correctly?); 2) Clarity of description; 3) Quality of the strategy; and 4) Clarity of arguments and presentation. You may consult your colleagues for ideas, *but please write your own programs and come to your own conclusions.*