COMS 4030A

Adaptive Computation and Machine Learning

LAB EXERCISE 1

Code up the PERCEPTRON TRAINING ALGORITHM for an m-input perceptron.
Your method should take an array X of inputs, of dimension $N \times m$, where N is the number of data points, and an array T, of length N, with target values for the data points.
In your algorithm, you should randomly sort the inputs to the algorithm before each epoch.
You should also code up the loss function and compute the loss after each epoch in testing.

You can create your own test datasets as follows.
Choose a linear function, say $f(x_1, x_2) = 2x_1 + 3x_2 - 1$.
Choose two random values in $(0, 1)$, say r_1 and r_2, and add (r_1, r_2) to your dataset.
If $f(r_1, r_2) \leq 0$ give it a target of 0, otherwise give it a target of 1.
Add as many points as you want to the dataset.
Try randomly changing a few target values and see what happens.

For a small test dataset, try the dataset with $X = \begin{bmatrix} 0 & 0 \\ 1 & 1 \\ 1 & 0 \\ 1 & 1 \end{bmatrix}$ $T = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 0 \end{bmatrix}$

Once you have learned the weights and threshold of the perceptron, draw a sketch of its linear discriminant and see if it separates the data points as it should. Now change the first entry in T to 0, and see what happens.

For the random sorting of the inputs, create an array I with integer entries from 0 to $N - 1$.
At the start of every epoch, randomly shuffle the elements of I.
Then, if the main **for** loop runs over i from 0 to $N - 1$, apply the update rule to $X[I[i]]$ and its corresponding target $T[I[i]]$.

(You don't need to submit anything for this exercise.)