Doubts and Clarifications (Perceptron Algorithm):

Here are some of my observations while practicing the Perceptron method for A.I. lab, and replies by Chandra Shekhar Lakshminarayanan Sir:

Q1. We cannot initialize Weights matrix(W) as a zero matrix, as it always gives the classifier line near to the origin(which doesn't shifts), irrespective of hyper-parameter settings(iterations, learning rate).

Ans: The algorithm is meant for linearly separable data and centered around the origin. So, we really don't want shifts. In order to understand what happens when w is not initialized to origin, say w was initialized to (-100,0) and there is only one example (1,1) which belongs to +1. Now it might take around 50 steps for the algorithm to converge even for this simple example. Moral of the story: if w is not initialized to 0 (origin), the proof that Perceptron converges in less than R_2/γ_2 iterations does not hold anymore.

Q2. We should use a learning rate (alpha, which is less than 1), while changing Weight(W) in Perceptron equation, otherwise the transitions/rotations in binary classifier lines could be drastically big.

W=W+alpha*X[j]*Y[j] for jth example

Ans: We actually want to make big changes and learn as fast as possible.

My reply: But sometimes it happens to be too big that, the classifier passes the complete data in a very few iterations and keeps on having a sort of oscillatory motion.

His reply: Its good to have that thought but we don’t have any proper proof say that. Moreover according to our current equations, now the convergence will take place in $(R^2)/(alpha*y^2)$, which is actually more than the original no. of iterations required as alpha<1.

Q3. For a change in data, there could be cases when the classifier which performed well for previous data-set might work very bad for new one, even after retaining same hyper-parameters (no. of iterations, alpha).

Ans: If the data set changes, we have to learn from scratch. Please note that unseen data of one problem and data set corresponding to a different problem are not equivalent.

Q4. My proposal according to my observations: Instead of using a constant learning rate for all the iterations we could have:

alpha[i+1]=alpha[i]/(1+i), where i is index corresponding to the ith iteration

This is to, reduce the transition/rotation rate as the iterations increases, because ideally some of the points should have been classified correctly in previous iterations and we should try to avoid bigger transitions in further iterations which could disturbed those prior classified points.

This sometimes even solves problem in point 3

Ans: Using 1/(i+1) learning rate is a big killer. After say 100 iterations, any change is only reflected on the second decimal point. This is equivalent to saying all the information in the data set is available in the first 100 training examples, and later on we don't have to learn anything. Here 100 is just an arbitrary choice, we can repeat the same with 1000 point and the changes are reflected only on the third decimal etc. Moral of the story: diminishing learning rate implies we don't learn as we get more and more data.

My reply: What I actually meant was, not to take this consideration by running/iterating through the dataset, but considering the iterations over the complete data in one go and then changing alpha.

His reply: Its ok to have these manipulations, but we cannot judge its correctness. Moreover you need to check if this also works when our data has some merging examples.