ECE/CS 559 - Fall 2018 - Homework #2

Montagna Marco

September 24, 2018

Exercise 2. Try to comment the obtained results:

(f) As it is possible to see from the printed results, from running steps (d) and (f) with $n = 50$, $\eta = 1$ and $\epsilon = 0$ we obtain the hoped results: step (d) terminates with 0 errors eventually. Regarding the percentage of misclassification errors (over all 1000 test samples) the computed result in this example is 45.6%. It is possible to find the results and the plot of this example at the end of the code. There is a discrepancy between the percentages of errors obtained through the training and test samples (the first one is 0% while the second one is around 45%). This is probably due to the fact that the network is not completely trained, but only with a sub-set of the possible samples. This bring to a big errors during testing phase.

(g) In this step it asked to perform the same passages with the following parameters: $n = 1000$, $\eta = 1$ and $\epsilon = 0$. Again step (d) terminates with 0 errors eventually. Increasing the parameter 'n' it is possible to observe that the the percentage of misclassification errors decrease to a value equal to 17.7% (for the tested example). Talking about the difference with the step (f) I assume that working with a greater number of elements in the training section brings to a better trained network; this leads to a minor percentage of misclassification errors. In this situation the sub-set of samples is bigger than the step (f) so the error decrease. It means that the network has been trained with a sub-set of 1000 samples, instead that n = 50, therefore the network will be more able to recognize the samples during the test phase.

(h) In this part it is requested to run the algorithm with n = 60000 (all the available samples in the file). The code ran until the epoch number was equal to 6125, than I decided to stop, guessing that the algorithm would not converge. I have made this hypothesis also because of the fact that the values in the error(epoch) continued to oscillate, never decreasing. The value n and η have been left unchanged and ϵ have been changed several times until the convergence situation was reached: it has been computed that, imposing $\epsilon = 0.12$, after 431 epoch the algorithm converges with a percentage of misclassification errors = 16.29 (for the tested example). Sometime, setting $n = 60000$, $\eta = 1$ and $\epsilon = 0.12$ seems that the algorithm could not converge (the number of epochs keep increasing and the number of misclassifications stop decreasing). Using an $\eta < 0.1125$ seems that the algorithm could not reach the convergence at all. The number of epochs in which the algorithm converge depends also on the initial random weights W.

(i) In this last step is requested to run three time the code keeping the same parameters and comment the results. I set $n = 60000$, $\eta = 1$ and $\epsilon = 0.125$ and the following results where obtained:

1. Percentage of misclassification errors = 15.7200
2. Percentage of misclassification errors = 15.6100
3. Percentage of misclassification errors = 15.0200

In this last step all the parameters n, $\eta and \epsilon$ are the same, the only thing that is changing is the vector of initial weights W that is initialized in a random way. This means that the little differences on the percentage of misclassification errors depends on the initial vector W. Using $\epsilon = 0.125$ means that the error during the the test could not go under the 12.5%. As already mentioned, the number of final epochs depends also on the initial values of W, that have been chosen randomly.