MACHINE LEARNING
BARUCH COLLEGE
FALL 2013

MIGUEL A. CASTRO
T.A.: LIQUN ZHU

Homework Assignment 2

1. As in class, suppose that you corrupt a signal Y_i with additive Gaussian noise with zero mean and standard deviation equal to $n\sigma_Y$ where n is the noise strength, and σ_Y is the standard deviation of the original signal Y_i; *i.e.*, $\varepsilon_i \sim \mathcal{N}(0, n\sigma_Y)$. You then form a corrupted signal $Y_i^* = Y_i + \varepsilon_i$. Assuming that the original signal is Gaussian, prove that the maximum achievable R^2 by any model designed to extract the original signal Y_i is given by:

$$maxR^2 = 1 - \frac{n^2}{1+n^2}.$$

   Recall that the R^2 is obtained by regressing the network's output against the *corrupted* signal Y_i^*.

2. In real life applications, of course, we typically don't have the original signal available to us, and are given the noisy signal Y_i^* from which we extract an approximation $\hat{Y}_i$ to the original signal Y_i. We saw in class that the neural network models were able to achieve R^2s that were very close to the maximum achievable given by the equation you derived in 1. Discuss how it would be possible for a poorly designed model (*e.g.* overtrained or overfitted) to achieve a higher R^2 than the $maxR^2$ found above.

3. Suppose that Y_i^* represents the actual return of a security that you're trading, while Y_i (which is hidden from your model) represents the effectively predictable portion of Y_i^*. Suppose, further, that you're reasonably certain that $\hat{Y}_i$, your model's approximation of Y_i, was obtained optimally so that your model's measured R^2 is very close to $maxR^2$. Discuss how you can make use of this information to assess the risks you face by trading this security. How would your risk change if your measured R^2 overestimates (i.e., is higher) than $maxR^2$?

4. Show that a SATALINE with 2 *satlins* Activations in the Hidden Layer and a *purelin* Activation in the Output Layer can solve the XOR Problem by considering each output region of each of the two *satlins* Hidden Nodes and showing that the SATALINE is able to implement at least two class-separation hyperplanes. The effort this simple problem takes to analyze should illustrate to you the

power of the Delta Learning Rule, which can quickly converge on the solution in an *automated* fashion.

5. Discuss how it is possible for the Training Error to continue to decrease (as a function of training epochs), while the Validation Error can increase. Why would you expect this past a certain point? Why would you stop the training when the Validation Error starts to increase?

6. In the fair coin problem, let H be the bias factor so that $0 \leq H \leq 1$. If the coin is fair, for example, we assign a high belief to $H = 0.5$. Using Bayes' Theorem, the Posterior probability can be written as being proportional to the product of the likelihood and prior: $P(H|D,I) \propto P(D|H,I) \times P(H|I)$. In class, we assumed we had no prior knowledge of how biased the coin was, and assumed a uniform distribution for the prior. We also saw that the likelihood is proportional to a binomial distribution with R heads out of N tosses. Suppose that, upon inspecting the coin, we become fairly certain that the coin is fair, and reflect this with a prior having a normal distribution centered at 0.5, and with a standard deviation of 0.1.

    a. Write down the Posterior Probability using the proportionality version of BT.
    b. Compute and plot the pdf of H (unnormalized is OK) for both the normal prior and the uniform-distribution prior (as in class) for the following combination of Heads/Tosses: 0/0, 0/1, 1/1, 0/2, 1/2, 2/2, 5/10, 50/100, and 500/1000.
    c. Compare how the two sets of experimental processes converge to the "true" value of H. Does the fact that we used two different priors change the result much?

7. The Occam factor penalizes a model with an adjustable parameter inversely proportionally to the adjustable parameter's dispersion (distribution width, like a std. dev.). What is the intuition behind this?

8. One of the outputs of a Bayesian-Regularized FFNN is the effective number of degrees of freedom of the trained network. How could you use this information to improve the network's architecture?

9. What role does the temperature parameter play in the Simulated Annealing technique *vis à vis* convergence? Discuss.

10. In the House Values in Boston case study, how would you implement a sensitivity analysis to determine how impactful each of the 13 attributes is when determining Median House Values? Explain. Would it be advisable to find out if there are collinearities in the 13 attributes? How would you do that? If you found collinearities among the 13 attributes, what should you do and why?

11. Why do you think you would get faster learning convergence using two outputs (0,1) and (1,0), respectively, to represent classes A and B, than a single output (0) and (1), respectively? Why do you think you would get faster learning convergence using a thermometer scale to represent ordinal variables than a single output? Discuss.

12. In the ROC curve shown in class (shown also below), Classifier A was claimed to be better than both Classifiers B and C, while it was not clear whether B was better than C. This assumes symmetrical costs for Type I and Type II misclassification errors. However, if you introduce different costs for Type I and Type II errors (as is often the case in real life), would it be possible to change the order of preference of the classifiers? Discuss and give examples.