Machine Learning and Pattern Recognition, Tutorial Sheet Number 4

School of Informatics, University of Edinburgh, Instructor: Chris Williams

1. Let $E(\boldsymbol{w})$ be a differentiable function. Consider the gradient descent procedure

$$\boldsymbol{w}^{t+1} \leftarrow \boldsymbol{w}^t - \eta \nabla_{\boldsymbol{w}} E$$

Let's say we initialize $\boldsymbol{w}^0 = \mathbf{0}$. True/false:

(a) Let $\boldsymbol{w}^1$ be the result of taking one gradient step. Then the error always improves, i.e., $E(\boldsymbol{w}^1) \leq E(\boldsymbol{w}^0)$.

(b) There exists some choice of the step size η such that $E(\boldsymbol{w}^1) < E(\boldsymbol{w}^0)$.

2. In the gradient descent procedure, a common programming mistake is to forget the minus sign, i.e., you unintentionally write a procedure that does

$$\boldsymbol{w}^{t+1} \leftarrow \boldsymbol{w}^t + \eta \nabla_{\boldsymbol{w}} E$$

If you make this mistake, what happens?

3. Consider the following classification problem. There are two real-valued features $x_1, x_2 \in \mathbb{R}$ and a binary class label. The class label is determined by

$$y = \begin{cases} 1 & \text{if } x_2 \geq |x_1| \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

(a) Can this be perfectly represented by a feedforward neural network without a hidden layer? Why or why not?

(b) Let's consider a simpler problem for a moment. Consider the classification problem.

$$y = \begin{cases} 1 & \text{if } x_2 \geq x_1 \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

Design a single neuron that solves this problem. Pick the weights by hand. For an activation function, use the hard threshold function

$$h(a) = \begin{cases} 1 & \text{if } a \geq 0 \\ 0 & \text{otherwise} \end{cases}$$

(c) Now go back to the classification problem at the beginning of this question. Design a two layer feedforward network (i.e., one hidden layer) that represents this function. Use the hard threshold activation function as in the previous question. *Hints:* Use two units in the hidden layer. The unit from the last question will be one of the units, and you will need to design one more. Your output unit will essentially perform a binary AND operation on the hidden units.

4. The following problem was introduced in the lectures. Work through the rest of the problem to obtain the solution.

You are an auditor of a firm. You receive details about the sales that a particular salesman is making. He attempts to make 4 sales a day to independent companies. You receive a list of the number of sales by this agent made on a number of days. Explain why you would expect the total number of sales to be binomially distributed.

If the agent was making the sales numbers up as part of a fraud, you might expect the agent (as he is a bit dim) to choose the number of sales at random from a uniform distribution. You are aware of the fraud possibility, and you understand there is something like a 1/5 chance this salesman is involved. Given daily sales counts of 1 2 2 4 1 4 3 2 4 1 3 3 2 4 3 3 2 3 3, do you think the salesman is lying?

From the lectures we had:

- $\mathcal{M} = 1$ - From $P_1(x|p)$ a binomial distribution Binomial$(4, p)$. Prior on p is uniform.
- $\mathcal{M} = 2$ - From $P_2(x)$ a uniform distribution Uniform(0,...,4).
- $P(\mathcal{M} = 1) = 0.8$.

$$P(\mathcal{D}|\mathcal{M} = 1) = \int dp\ P_1(\mathcal{D}|p)P(p)\ ,\ P(\mathcal{D}|\mathcal{M} = 2) = P_2(\mathcal{D})$$

$$P(\mathcal{M}|\mathcal{D}) = \frac{P(\mathcal{D}|\mathcal{M})P(\mathcal{M})}{P(\mathcal{D}|\mathcal{M} = 1)P(\mathcal{M} = 1) + P(\mathcal{D}|\mathcal{M} = 2)P(\mathcal{M} = 2)}$$

To get the solution, you should look at the form of the density of a beta distribution and note that it must integrate to 1. You also will need to be able to compute with Γ functions of big numbers. In matlab it is worth doing the computations in log space, and using the `gammaln` function. Then you can exponentiate it when you have done all the sums.