Supervised Learning 2004 — Assignment 1

Iain Murray <i.murray+ta@gatsby.ucl.ac.uk>

Due Monday 25th October 2004

Please hand in a print out of any code you write for this assignment. Each question carries equal marks. While discussion is always encouraged, you should do this assignment by yourself.

Question numbers and page numbers refer to *"Information Theory, Inference, and Learning Algorithms"*, David MacKay, CUP 2003. If necessary you can download the book or relevant pages: `http://www.inference.phy.cam.ac.uk/mackay/itila/book.html`

Q1 Binary Threshold Neuron

Do Exercise 40.6 p490. Explain your answers.

For this question assume (as in the chapter) that "binary threshold neurons" do *not* have bias weights and have the following activity rule:

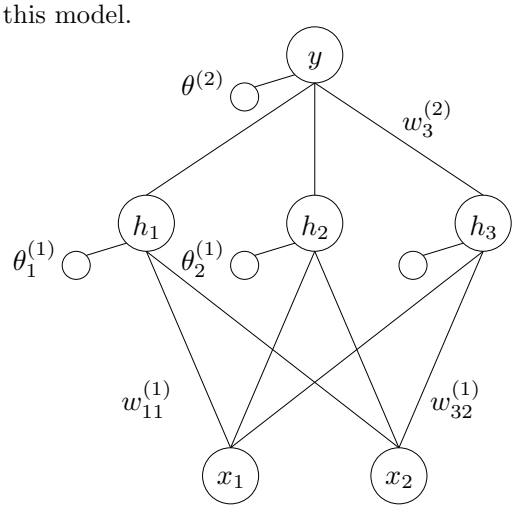
$$y(\mathbf{x}, \mathbf{w}) = \begin{cases} 1 & a > 0 \\ 0 & a \leq 0 \end{cases} \qquad \text{where} \qquad a = \mathbf{x}^\top \mathbf{w} = \mathbf{w}^\top \mathbf{x} = \sum_i w_i x_i$$

Q2 Training a single neuron

Do exercise 39.2 from p475. Make sure you understand why the answer to this question is useful.

Q3 A feed-forward neural network

Here we describe a neural network with 13 parameters. You will explore evaluating and learning in this model.



Hidden layer indexed by $j = 1 \ldots 3$

Input layer indexed by $l = 1 \ldots 2$

Activations

Hidden layer: $a_j^{(1)} = \sum_{l=1}^{2} w_{jl}^{(1)} x_l + \theta_j^{(1)}$

Output layer: $a^{(2)} = \sum_{j=1}^{3} w_j^{(2)} h_j + \theta^{(2)}$

Activity rules

$$h_j(a_j^{(1)}) = \sin(a_j^{(1)})$$

$$y(a^{(2)}) = \frac{1}{1+\exp(-a^{(2)})}$$

Parameter vector
We define $\mathbf{w} \equiv (\theta_1^{(1)}, w_{11}^{(1)}, w_{12}^{(1)}, \theta_2^{(1)}, w_{21}^{(1)}, w_{22}^{(1)}, \theta_3^{(1)}, w_{31}^{(1)}, w_{32}^{(1)}, \theta^{(2)}, w_1^{(2)}, w_2^{(2)}, w_3^{(2)})$

Error function
Let the error function E_D be the information measure used in Q2, reproduced here, using the observations shown on the right:

$$E_D(\mathbf{w}) \equiv G(\mathbf{w})$$
$$= -\sum_n \Big(t^{(n)} \log(y(\mathbf{x}^{(n)}, \mathbf{w})) + (1 - t^{(n)}) \log(1 - y(\mathbf{x}^{(n)}, \mathbf{w})) \Big)$$

n	$x_1^{(n)}$	$x_2^{(n)}$	$t^{(n)}$
1	-1	-1	0.9
2	-1	+1	0.1
3	+1	-1	0.1
4	+1	+1	0.9

a) Evaluating the error function

Write a program to compute $E_D(\mathbf{w})$ and evaluate $E_D(\mathbf{w}')$ for
$\mathbf{w}' = (0.1, 0.1, 0.1, 0.1, -0.2, -0.3, -0.1, 0.1, -0.2, 0.1, 0.2, 0.3, 0.4)$.

b) Perturbing the weight vector

Use your program to find for $k = 1 \ldots K$

$$\tilde{g}_k(\mathbf{w}') = \frac{E_D(\mathbf{w}' + \epsilon \boldsymbol{\eta}^{(k)}) - E_D(\mathbf{w}')}{\epsilon},$$

where $\epsilon = 10^{-6}$ and $\boldsymbol{\eta}^{(k)}$ is a vector of zeros with a one in position k:

$$\eta_j^{(k)} = \begin{cases} 1 & \text{if } j = k \\ 0 & \text{otherwise} \end{cases}$$

c) Learning the weight vector

Describe and explain a method for using $\tilde{g}_k(\mathbf{w})$ to find a setting of $\mathbf{w}$ that predicts the outputs $t^{(n)}$ from the inputs $\mathbf{x}^{(n)}$.

If you implement this (OPTIONAL), then include code and a solution for $\mathbf{w}$. Full credit may be obtained from a *good* description alone.

d) OPTIONAL: Backpropagation

Find $g_k(\mathbf{w}) \equiv \frac{\partial E_D(\mathbf{w})}{\partial \mathbf{w}_k}$ for $k = 1 \ldots K$ and learn $\mathbf{w}$ by backpropagation. This is discussed on p140 of *"Neural Networks for Pattern Recognition"*, C.M. Bishop. As discussed on p147 the method described in this assignment is less efficient, but should always be performed anyway to check your implementation. This part carries *no* credit and could be quite involved.