2 Artificial Intelligence (sbh11)

(*a*) You wish to perform *linear regression* using a function

$$h_{\mathbf{w}}(\mathbf{x}) = w_0 + \sum_{i=1}^{n} w_i x_i$$

where $\mathbf{w}^T = \begin{bmatrix} w_0 & w_1 & \cdots & w_n \end{bmatrix}$ is a vector of weights including bias w_0, and $\mathbf{x}^T = \begin{bmatrix} x_1 & x_2 & \cdots & x_n \end{bmatrix}$ is a vector of features. You will do this by minimising the error function

$$E(\mathbf{w}) = \sum_{i=1}^{m} \left(h_{\mathbf{w}}(\mathbf{x}_i) - y_i\right)^2$$

for a training set $S = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_m, y_m)\}$. From an understanding of the underlying problem, you know that a given subset of the weights must be non-positive, and the rest must be non-negative. The bias is unconstrained.

(*i*) How can you influence the signs of the weights by modifying $E(\mathbf{w})$ only, by adding a term that is a function of $\mathbf{w}$ but not of the $\mathbf{x}_i$? In your answer it should be possible for the strength of the influence to be adjusted. [4 marks]

(*ii*) How can you influence the signs of the weights by modifying $h_{\mathbf{w}}(\mathbf{x})$ only? In this part you do not need to adjust the strength of the influence. Discuss which changes to the usual learning algorithm are needed. It is not necessary to derive the learning algorithm in full. [4 marks]

(*iii*) Which of your suggestions from Parts (*a*)(*i*) and (*a*)(*ii*) would you expect to be more effective in practice? [2 marks]

(*b*) You have found that linear regression is not appropriate for the problem at hand. You have tried unsuccessfully to use a standard multi-layer perceptron (MLP) with a single layer of p hidden nodes. Instead you want to try to solve it using the function

$$h_W(\mathbf{x}) = w_0 + \sum_{i=1}^{p} w_i \exp\left(-\frac{1}{\sigma_i^2}||\mathbf{x} - \mathbf{c}_i||^2\right).$$

It is now no longer necessary to influence the signs of the weights. Here, W is a set containing all the weights w_i, σ_i and $\mathbf{c}_i$. The error function remains the sum of squared differences, but now depends on all the weights in W.

(*i*) How does the standard MLP differ from this system? [2 marks]

(*ii*) Assuming that you have already completed backpropagation to compute the gradient of the error function for the MLP case, explain in detail how

the calculation of the gradient has to be modified for training the new system. [8 marks]