2 Artificial Intelligence I (SBH)

We wish to solve a *supervised learning* problem using a *perceptron* computing the function

$$h(\mathbf{w};\mathbf{x}) = \sigma\left(\mathbf{w}^T\mathbf{x}\right)$$

where $\mathbf{w}$ is a vector of *weights*, $\mathbf{x}$ is a vector of *features* and $\sigma(z) = 1/(1+e^{-z})$. We have a set of m *labelled examples* $\mathbf{s} = ((\mathbf{x}_1, o_1), \ldots, (\mathbf{x}_m, o_m))$ where $o_i \in \{0, 1\}$.

(*a*) Derive the *gradient descent training algorithm* for training the perceptron by minimizing the *error function*

$$E(\mathbf{w}) = \sum_{i=1}^{m} \left(o_i - h(\mathbf{w};\mathbf{x}_i)\right)^2.$$

You may if you wish employ the result

$$\frac{d\sigma(z)}{dz} = \sigma(z)(1-\sigma(z)).$$

[7 marks]

(*b*) We are now told that some training examples are more important than others, and it is thus more important that, after training, there is only a small difference between o_i and $h(\mathbf{w};\mathbf{x}_i)$ for these examples. Derive a new version of the training algorithm that takes this modification into account. [6 marks]

(*c*) Having trained a classifier $h(\mathbf{w}_{\text{opt}};\mathbf{x})$ in part (*a*) using the training data available, a colleague presents you with a second classifier $h'(\mathbf{w}'_{\text{opt}};\mathbf{x}')$. Your colleague has trained this classifier using the same number of examples and the same labels, but a different collection of features, so for their classifier the training data was

$$\mathbf{s}' = ((\mathbf{x}'_1, o_1), \ldots, (\mathbf{x}'_m, o_m)).$$

Devise a way in which you might perform further training in order to *combine* the two classifiers h and h' into a single, possibly more powerful, classifier.

[7 marks]