Artificial Intelligence I

A perceptron takes inputs \( x^T = (x_1, x_2, \cdots, x_n) \in \mathbb{R}^n \) and computes its output

\[
h(x; w) = w_0 + \sum_{i=1}^{n} w_i x_i
\]

using weight vector \( w^T = (w_0, w_1, w_2, \cdots, w_n) \in \mathbb{R}^{n+1} \). We aim to use it to solve a regression problem using a training set \( s^T = ( (x_1, y_1), (x_2, y_2), \cdots, (x_m, y_m) ) \) with \( y_i \in \mathbb{R} \). The approach will be to minimise the error function

\[
E(w) = \sum_{i=1}^{m} (y_i - h(x_i, w))^2
\]

by gradient descent.

(a) Derive the gradient descent learning algorithm for this problem. [5 marks]

(b) The application dictates that the learning process sets as many weights as possible to zero, with the possible side effect that \( E \) is increased. It has been suggested that the error function used above might be modified by adding a further term

\[
\lambda \sum_{i=0}^{n} f(w_i, \theta)
\]

to \( E \) where

\[
f(w, \theta) = \begin{cases} 
1 & \text{if } |w| > \theta \\
0 & \text{if } |w| \leq \theta
\end{cases}
\]

(i) Explain the purpose of the parameters \( \lambda \) and \( \theta \) in the extra term. [4 marks]

(ii) Assuming we continue to use a gradient descent approach, explain why this term might be inappropriate. [1 mark]

(c) Suggest a function that is appropriate for a gradient descent approach, having a shape similar to that of \( f \), and derive the associated gradient descent learning algorithm. [10 marks]