Practice Problems

1. Consider a training set of five data points $\{(\mathbf{x}_i, t_i)\}_{i=1}^{5}$, where each input $\mathbf{x}_i$ is 2-dimensional and its corresponding target t_i is a real valued scalar. We wish to learn a suitable regression function, that can perfectly pass through all these points. To this goal, consider adopting a Radial Basis Function (RBF) neural network.

   (a) Draw a schematic representation of the neural network for this regression problem. Clearly specify:

      i. the number of nodes in the input layer

      ii. the number of basis functions in the hidden layer

      iii. the number of nodes in the output layer.

   (b) What is the training error for this RBF network. Justify your answer.

   (c) How many weights are to be learnt for this regression task. Give your reasoning.

2. Consider an HMM representation of a coin-tossing experiment. You are given a three state model (corresponding to three different coins) with probabilities

	State 1	State 2	State 3
$P(H)$	0.5	0.6	0.4
$P(T)$	0.5	0.4	0.6

H: Head T: Tail

and with all state transition probabilities set to $\frac{1}{3}$. The initial probabilities for state 1, state 2 and state 3 are 0.5, 0.2 and 0.3 respectively. Given the observation sequence $\mathbf{O} = HHTT$, of length four,

   (a) What is the probability that this sequence came from state 1 entirely?

   (b) Using the Viterbi algorithm, what is the optimal sequence of states that can be assigned to $\mathbf{O}$?

3. Consider a multi-layer perceptron with the following specifications:

   - d nodes in the input layer
   - sigmoidal activation functions at the M nodes in the hidden layer.
   - linear activation functions at the c nodes in the output layer.

The weights of the network are to be learnt using the error function

$$E = \frac{1}{6}\sum_{k=1}^{c}(t_k - y_k)^6$$

where t_k and y_k denote the target and predicted values at the k^{th} output node. Derive expressions for the following:

(a) Predicted value y_k in terms of the training sample $\mathbf{x} = (x_1, x_2.....x_d)^T$, activation functions and network weights. (You may ignore the bias terms in the derivation.)

(b) Updation of weight w_{kj}, connecting the j^{th} hidden node to the k^{th} output node.

(c) Updation of weight w_{ji}, connecting the i^{th} input node to the j^{th} hidden node. **Hint**: Back propagation!

Note: For the sigmoidal function $g(x) = \frac{1}{1+e^{-x}}$, the derivative is $g(x)(1-g(x))$.