Neural Network Assignment 1: (3 Parts), Prof. Q.J. Zhang, Carleton University

Part 1: MLP

(i). Consider a 3-layer perceptron neural network consisting of 1 input neuron, 2 hidden neurons with sigmoid activation functions, and 1 output neuron with linear activation function. Assuming v_1 and v_2 represent the connection weights from the hidden neurons to the output neuron, v_0 represents the bias for the output neuron, w_{11} and w_{21} represent the connection weights from the input neuron to the hidden neurons, and w_{10} and w_{20} represent the biases for the 2 hidden neurons. Let x represent the input to the neural network and y represent the output from the neural network.

Suppose the vector of neural network weights is:

$$\boldsymbol{w} = [v_0 \;\; v_1 \;\; v_2 \;\; w_{10} \;\; w_{11} \;\; w_{20} \;\; w_{21}]^T = [0 \quad 0.1 \quad 0.5 \quad 0 \quad -1 \quad 1 \quad 0.2]^T.$$

Calculate the output y if input $x = 0$. Calculate y again for $x = 1$.

(ii). For the same neural network as in (1), calculate the derivative of y with respect the neural network weight parameters v_0, v_1, v_2, w_{10}, w_{11}, w_{20}, and w_{21} for $x = 0$. Calculate the derivatives again for $x = 1$.

Neural Network Assignment 1:

Part 2: RBF

Consider a RBF network of *n* inputs $\boldsymbol{x} = [x_1, x_2, \ldots, x_n]^T$ and *m* outputs $\boldsymbol{y} = [y_1, y_2, \ldots, y_m]^T$. Suppose the outputs are weighted sum of RBF functions:

$$y_j = \sum_{i=1}^{H} c_{ji} R_i(\boldsymbol{x}), \quad j = 1, 2, \ldots, m$$

$$R_i(\boldsymbol{x}) = exp\left(-\frac{(\boldsymbol{x} - \boldsymbol{u}_i)^T (\boldsymbol{x} - \boldsymbol{u}_i)}{2\sigma_i^2} \right)$$

where $\boldsymbol{u}_i$ is a vector of dimension *n*, and *H* is the number of RFB hidden neurons.

Derive the gradient of training error with respect to necessary variables in the RBF network in order to use gradient-based training algorithms, assuming the training error per sample of data as

$$E = \frac{1}{2} \sum_{j=1}^{m} (y_j(\boldsymbol{x}) - \hat{y}_j)^2$$

where $\hat{y}_j$ represents one sample of the training data for the *j*th output.

Neural Network Assignment 1:

Part 3: Function Approximation by Neural Networks

(i) For the x-y function in Figure 1, approximate the function using a 3-layer MLP with sigmoid activation functions in the hidden layer and linear functions in the output layer. Show the neural network structure graphically, and indicate the approximate values of neural network internal biases and connection weights.

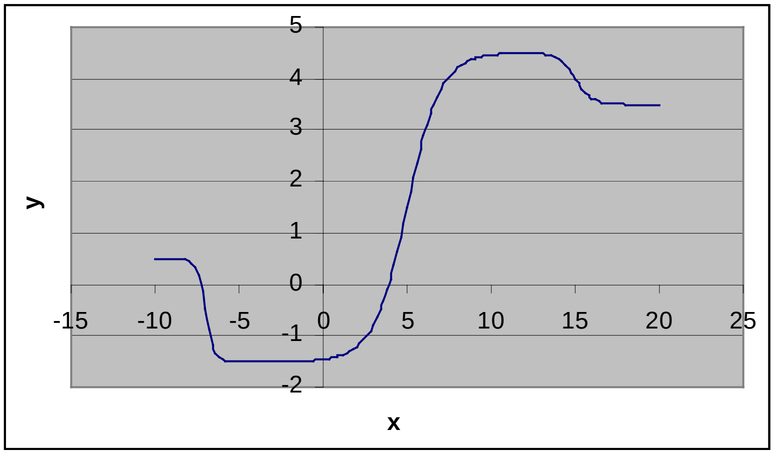

Figure 1

(ii). For the x-y function in Figure 2, approximate the function using a RBF network with Gaussian activation functions in the hidden layer and linear functions in the output layer. Show the RBF neural network structure graphically, and indicate the approximate values of neural network internal parameters including parameters for the Gaussian functions and the neural network connection weights.

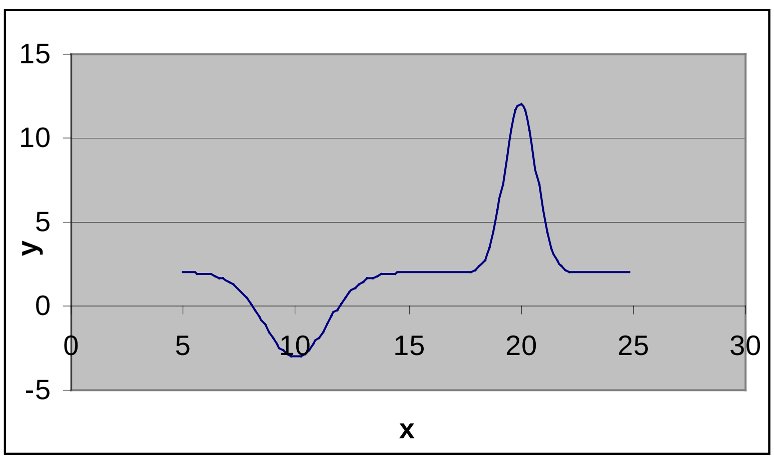

Figure 2