Homework:

(1) A single layer perceptron is given

$$y(\mathbf{x}, \mathbf{w}) = h\Big(\sum_{i=0}^{2} w_i x_i\Big)$$

where $\mathbf{x} \in [x_1, x_2]^{\mathrm{T}}$ and $x_0 = 1$, $\mathbf{w} = [w_0, w_1, w_2]^{\mathrm{T}}$. $h(v) = \frac{1}{1+\exp(-50v)}$. Calculate the network output for each of the data input $\mathbf{x} = [1, 2]^{\mathrm{T}}$, $\mathbf{x} = [2, 1]^{\mathrm{T}}$, $\mathbf{x} = [2, 2]^{\mathrm{T}}$, in the cases that the network weights are $\mathbf{w} = [1, 0.5, -0.5]^{\mathrm{T}}$, $\mathbf{w} = [-1, 0.5, -0.5]^{\mathrm{T}}$ and $\mathbf{w} = [0, 0.5, -0.5]^{\mathrm{T}}$ respectively. Sketch the network diagram.

(2) The online gradient descent algorithm is used to train a single layer perceptron given by

$$y(\mathbf{x}, \mathbf{w}) = h\Big(\sum_{i=0}^{2} w_i x_i\Big)$$

where $\mathbf{x} \in [x_1, x_2]^{\mathrm{T}}$ and $x_0 = 1$, $\mathbf{w} = [w_0, w_1, w_2]^{\mathrm{T}}$. $h(v) = \tanh(v) = \frac{\exp(v)-\exp(-v)}{\exp(v)+\exp(-v)}$. (Note: $h'(v) = (1 - h(v)^2)$). Assume that the current weight vector is $\mathbf{w} = [1, 0.3, 0.4]^{\mathrm{T}}$, Calculate the new weight updated from a new training datum $[\mathbf{x}, t] = [2, 1, 0.2]^{\mathrm{T}}$, using the learning rate $\eta = 0.02$.

(3) The mathematical form for a two layer MLP is

$$y(\mathbf{x}, \mathbf{w}) = \sigma\Big(\sum_{j=0}^{M} w_j^{(2)} h\Big(\sum_{i=0}^{D} w_{ji}^{(1)} x_i\Big)\Big)$$

where $\mathbf{x} \in \Re^D$. $x_0 = 1$. The superscript $^{(1)}$ and $^{(2)}$ indicate the corresponding weights are in *first* or *second* layer. $h(.)$ and $\sigma(.)$ are chosen activation functions. Sketch the diagram of MLP for $M = 2$ and $D = 3$, specifying the weights on the paths of the diagram.

(4) The mathematical form for a two layer MLP is

$$y(\mathbf{x}, \mathbf{w}) = \sigma\Big(\sum_{j=0}^{2} w_j^{(2)} h\Big(\sum_{i=0}^{2} w_{ji}^{(1)} x_i\Big)\Big)$$

where $x_0 = 1$, $\mathbf{x} \in \Re^D$. The superscript $^{(1)}$ and $^{(2)}$ indicate the corresponding weights are in *first* or *second* layer. For

$$\sigma(v) = h(v) = \begin{cases} 1 & \text{if } v \geq 0 \\ 0 & \text{if } v < 0 \end{cases}$$

Calculate the network outputs for each of the data input $\mathbf{x} = [-1, 3]^{\mathrm{T}}$, $\mathbf{x} = [-3, 1]^{\mathrm{T}}$, $\mathbf{x} = [-2, 2]^{\mathrm{T}}$, in the cases that the network weights are $w_{ji}^{(1)} = 1$, $\forall i, j$, $\mathbf{w}^{(2)} = [0, 0.5, -0.5]^{\mathrm{T}}$, $\mathbf{w}^{(2)} = [0, 1, 5, -0.5]^{\mathrm{T}}$ and $\mathbf{w}^{(2)} = [0, 0.5, -1.5]^{\mathrm{T}}$ respectively.

(5) Explain the training step of the error backpropogation algorithm for the two layer perceptron in (4). If the previous network weights are $w_{ji}^{(1)} = 1$, $\forall i, j$,

$\mathbf{w}^{(2)} = [0, 0.5, -0.5]^{\mathrm{T}}$, what are the weights after applying one pass of the error backpropogation algorithm with the training sample $[\mathbf{x}, t] = [2, 1, 3]^{\mathrm{T}}$?

(6) Run XORMLP.m and discuss the results. Then modify the program line 5 to eta=0.05; Explain the effects on the learning. Then modify line 22 to $y = 1/(1 + exp(-50 * temp))$; Explain the effects on the network output surface.

(7) A radial basis function has the form of

$$y(\mathbf{x}) = \sum_{i=1}^{3} w_i \exp\left(-\frac{(\|\mathbf{x} - \mathbf{c}_i\|)^2}{2}\right)$$

The centers are $\mathbf{c}_1 = [-1, 3]^{\mathrm{T}}$, $\mathbf{c}_2 = [-3, 1]^{\mathrm{T}}$, $\mathbf{c}_3 = [-2, 2]^{\mathrm{T}}$. Calculate the network outputs for input $\mathbf{x}$ equals to each center $\mathbf{c}_1$, $\mathbf{c}_2$ and $\mathbf{c}_3$, respectly in the cases that the network weights as $\mathbf{w} = [1/3, 1/3, 1/3]^{\mathrm{T}}$,$\mathbf{w} = [2, 1, -1]^{\mathrm{T}}$,$\mathbf{w} = [1, 0, 0]^{\mathrm{T}}$.

(8) Run sinEX.m and discuss the result. Modify sinEX.m to model a RBF network to approximate Cosine function using five centers $c_1 = 0$, $c_2 = 0.2$, $c_3 = 0.5$, $c_4 = 0.8$, $c_5 = 1$.