SC4001 Tutorial 2 **Regression and classification**

1. Train a linear neuron to learng the mapping from input $\boldsymbol{x} \in \boldsymbol{R}^3$ to output y from the following examples:

$\boldsymbol{x} = (x_1, x_2, x_2)$	y
$(0.09 \quad -0.44 \quad -0.15)$	-2.57
$(0.69 \quad -0.99 \quad -0.76)$	-2.97
$(0.34 \quad 0.65 \quad -0.73)$	0.96
$(0.15 \quad 0.78 \quad -0.58)$	1.04
$(-0.63 \quad -0.78 \quad -0.56)$	-3.21
$(0.96 \quad 0.62 \quad -0.66)$	1.05
$(0.63 \quad -0.45 \quad -0.14)$	-2.39
$(0.88 \quad 0.64 \quad -0.33)$	0.66

(a) Show one iteration of learning of the neuron with

i. Stochastic gradient descent learning

ii. Gradient descent learning

Initialize the weights as $\begin{pmatrix} 0.77 \\ 0.02 \\ 0.63 \end{pmatrix}$ and biases to 0.0, and use a leaning factor $\alpha = 0.01$.

(b) Plot the learning curves (mean square error vs. epochs) until convergence. Determine the learned weights and biases.

(c) Find the predicted values y of training inputs after the training.

2. Two-dimensional training patterns (inputs) to design a dichotomizer are given as:

$$\mathbf{x}_1 = \begin{bmatrix} 5 \\ 1 \end{bmatrix}; \quad \mathbf{x}_2 = \begin{bmatrix} 7 \\ 3 \end{bmatrix}; \quad \mathbf{x}_3 = \begin{bmatrix} 3 \\ 2 \end{bmatrix}; \quad \mathbf{x}_4 = \begin{bmatrix} 5 \\ 4 \end{bmatrix}; \qquad \textit{Class } 1$$

$$\mathbf{x}_5 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}; \quad \mathbf{x}_6 = \begin{bmatrix} -1 \\ -3 \end{bmatrix}; \quad \mathbf{x}_7 = \begin{bmatrix} -2 \\ 3 \end{bmatrix}; \quad \mathbf{x}_8 = \begin{bmatrix} -3 \\ 0 \end{bmatrix}; \qquad \textit{Class } 2$$

(a) Determine whether the two classes of patterns are linearly separable.

(b) Find the center of gravity of patterns in each class. Show that a linear decision boundary passing perpendicularly through the middle point of the line joining the two centroids is given by:

$$6.5x_1 + 2.5x_2 - 14.5 = 0$$

(c) Design a discrete perceptron having the decision boundary as in part (b) for the classification and show that the perceptron separates the points perfectly.

(d) Determine the classes identified by the neuron for following input patterns:

$$\begin{pmatrix}4\\2\end{pmatrix}, \begin{pmatrix}0\\5\end{pmatrix}, \begin{pmatrix}36/_{13}\\0\end{pmatrix}$$

3. Use ‘make_blobs’ function from **sklearn**.datasets to create 100 samples of two Gaussian distributed classes for 3-dimensional inputs:

   from **sklearn**.datasets import make_blobs

   Assume each class has a standard deviation = 5.0.

   Use torch.nn.BCELoss() and torch.autograd() functions to train a logistic neuron to separate the two classes.
   Find the classification error at convergence and plot the decision boundary.

4. Train a perceptron to learn the following function ϕ:

$$\phi(x, y) = 1.5 + 3.3x - 2.5y + 1.2xy$$

   for $0 \leq x, y \leq 1$.

   (a) Sample 100 data points randomly from the input space to create a training dataset.
   (b) Use the gradient descent algorithm to train the perceptron
   (c) Compute the training error and plot the function approximated by the perceptron.
   (d) Show how a linear neuron can be used to predict the above function
   (e) Compare the results of approximations above by the linear neuron and the perceptron