1. Design a linear neuron to perform the following mapping:

$\boldsymbol{x} = (x_1, x_2, x_2)$	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

Show one iteration of learning of the neuron with
(a) Stochastic gradient descent learning
(b) Gradient descent learning

Initialize the weights randomly and biases to 0.0 and use a leaning factor $\alpha = 0.01$.

Plot the learning curves (mean square error vs. epochs) until convergence and find the learned weights and biases and the predicted values of y by the neuron.

2. Design a perceptron to approximate the function y:

$$y = 0.5 + x_1 + 3{x_2}^2$$

for inputs $0 \leq x_1, x_2 \leq 1$.

(a) Divide the input space into a grid of squares of size 0.1×0.1 and use the grid points as training data.
(b) Use gradient descent learning to train the perceptron with a learning rate $\alpha = 0.01$. Plot the mean square error against epochs until convergence
(c) Plot the targets and the predicted data points, and estimate the mean square error of the prediction.
(d) Repeat above (a)-(c) with stochastic gradient descent learning and compare the performance.

3. Train a linear neuron to learn the following function ϕ:

$$\phi(x, y) = 1.5 + 3.3x - 2.5y + 0.2xy$$

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

(a) Sample 25 data points randomly from the input space for training.
(b) Use the gradient descent algorithm to train a linear neuron
(c) Compute the training error and plot the function approximated by the linear neuron.
(d) Repeat (b) and (c) for a perceptron and compare the results with those of the linear neuron.