CZ4042 Tutorial 4 **Neuron Layers**

1. Design a softmax layer of neurons to perform the following classification, given the inputs $x = (x_1, x_2)$ and target class labels d:

(x_1, x_2)	(0 4)	(-1 3)	(2 3)	(-2 2)	(0 2)	(1 2)	(-1 2)	(-3 1)	(-1 1)
d	A	A	A	B	B	A	B	B	B

(x_1, x_2)	(2 1)	(4 1)	(-2 0)	(1 0)	(3 0)	(-3 -1)	(-2 -1)	(2 -1)	(4 -1)
d	C	C	B	C	C	B	B	C	C

(a) Show one iteration of gradient descent learning at a learning factor 0.05.
(b) Find the weights and biases at convergence of learning
(c) Indicate the probabilities that the network predicts the classes of trained patterns.

2. Use mini-batch gradient decent learning to train a softmax layer to classify Iris dataset (https://archive.ics.uci.edu/ml/datasets/Iris). The dataset contains 150 data points. Use 120 data points for training the classifier and test on the remaining 30 data points.
Set learning rate = 0.01 and batch size = 16.

You can use the following python commands to load Iris data:

```
from sklearn import datasets
iris = datasets.load_iris()
```

Repeat the classification with batch sizes = 2, 4, 8, 16, 24, 32, 48, and 64, and compare the accuracies and times taken for a weight update.

3. Design a perceptron layer to perform the following mapping:

Inputs	Outputs
(0.50 0.23)	(0.16 0.74)
(0.20 0.76)	(0.49 0.97)
(0.17 0.09)	(0.01 0.26)
(0.69 0.95)	(1.19 1.70)
(0.00 0.51)	(0.13 0.52)
(0.81 0.61)	(0.77 1.48)
(0.72 0.29)	(0.40 1.04)
(0.92 0.72)	(1.14 1.7)

Train the perceptron layer with (a) GD and (b) SGD. Show one iteration of learning and plot learning curves and predicted and target outputs. Set the learning factor $\alpha = 0.05$.