**CS 678 Machine Learning**
**Programming Assignment #4 Part A**
**(Graded – mandatory)**

# Learning a Linear Function via Perceptron

## Specification

The idea is to write a simple program that uses a Perceptron to "learn" certain Boolean functions. Basically, implement the In-class Exercise and Homework #9.

## Background

A single perceptron is a bio-inspired model that typically uses a Sum_of_Products (*S*) operation together with a "step" function (*y*) for activation:

$$S = \sum_{i=1}^{d} w_i x_i + w_{bias} \qquad\qquad y = \begin{cases} 1, & if\ S > 0 \\ 0, & otherwise \end{cases}$$

where *w* are parameters (or weights), *x* are the inputs and *d* is the dimension.

Perceptrons are capable of learning linear functions by using gradient descent to minimize error (*E*) and find class boundaries:

$$\nabla_w E = \left[\frac{\partial E}{\partial w_1}, \frac{\partial E}{\partial w_2}, \cdots \frac{\partial E}{w_d}\right]^T$$

This involves a process of iterative parameter update using the "Delta" rule:

$$\Delta w_{i,j} = \eta(t_j - y_j)x_i$$

where *t* is the target and η is the learning rate.

## Assignment

Implement a Perceptron capable of learning any separable two-input Boolean function (e.g. AND, OR, NAND, NOR).

## Requirements

- Your program should display diagnostic output (as demonstrated in class) that clearly shows progress towards learning the function.
- The same program must be able to learn any linearly-learnable Boolean function, simply by changing the data presented to it.
- Submit a written report (single PDF):
    - Include complete documentation, source code and program output.