# MACHINE LEARNING

## Software Project 2, Component 1

Jose Luis Galiano Gómez

# PROBLEM DEFINITION

The task to solve in this second software project will be a simple supervised binary classification problem. Specifically, the task at hand will be to determine whether a given banknote is authentic given a number of measures taken from a photograph.

# PROBLEM SPECIFICATION

## Input data and preconditions

The input data we will use is a labeled dataset provided by the UC Irvine Machine Learning Repository at https://archive.ics.uci.edu/dataset/267/banknote+authentication. It consists of 1372 instances with 4 input features each, and a fifth class feature that can either be 0 (authentic banknote) or 1 (unauthentic). The dataset is simple, with a low number of features for each data instance, but sizable enough so that it can be used. Furthermore, there are no missing values.

## Output and postconditions

The output will be a trained model which will then be used on a test dataset (cross-validation). The model will output a classification for each of the instances within said dataset, either 0 or 1.

# SPECIFICATION OF THE LEARNING TASK

**Task:** To categorize each banknote, represented by 4 input features each, into either authentic or unauthentic.

**Performance measure:** Percentage of correctly classified instances within the tested dataset.

**Experience:** Training via gradient descent given a labeled dataset.

# ML TECHNIQUE: PERCEPTRON

The perceptron is a simple supervised machine learning algorithm and one of the earliest neural network architectures, introduced by Frank Rosenblatt in the late 1950s. A perceptron maps a set of training examples (of ***d***-dimensional input vectors) onto binary output values using a ***d*** - *1* dimensional hyperplane.

The perceptron is a very simple neural network. Given a dataset $\{(x_1,y_1), \ldots, (x_n,y_n)\}$ where $x_i$ is a ***d***-dimensional input vector $(x_{i1},\ldots,x_{id})$ of feature values for that data instance and $y_i$ the binary target variable, the perceptron has a real-valued weight vector $\boldsymbol{w} = (w_1,\ldots,w_d)$ and a real-valued bias ***b***.

The gradient descent algorithm used for training consists of the following steps:

1. Firstly, the weight vector and bias are initialized to all zeros.
2. A linear combination of the input features and weights is computed using a matrix ***X*** of (number of instances, number of features) proportions that holds all training examples through the following operation: $\boldsymbol{a = X \cdot w + b}$.
3. The Heaviside activation function is applied to the resulting vector from the previous step to convert the real values into binary values: $\boldsymbol{\bar{y}_i = 1 \text{ if } a_i >= 0, \text{ else } \bar{y}_i = 0}$.
4. The weights and bias updates are computed as follows: $\boldsymbol{\Delta w = \eta X^T \cdot (\bar{y} - y)}$, $\boldsymbol{\Delta b = \eta(\bar{y}-y)}$, where $\boldsymbol{\eta}$ is the learning rate, a scalar value that determines the magnitude of the update to the weights in each iteration.
5. The weights and biases are updated: $\boldsymbol{w = w + \Delta w, \; b = b + \Delta b.}$