**Classification – Bank Notes Workshop**

**Objective:** To practice training a neural network for **classification** using Keras library (<https://keras.io/>)

**Overview**

In this workshop, we train a neural network to classify a given banknote as either Genuine or Counterfeit.

**Data**

The dataset provided is called ‘banknotes.csv’ and consists of physical dimensions of genuine and counterfeit Swiss banknotes. There are 200 samples in the dataset; 100 are samples of Genuine banknotes and 100 are samples of Counterfeit banknotes. Source of dataset - <http://www.statistics4u.com/fundstat_eng/data_fluriedw.html>

**Your Tasks**

1. Use Pandas to read in ‘banknotes.csv’.
2. Select the following columns as **features**:
   * Length
   * Left
   * Right
   * Bottom
   * Top
   * Diagonal
3. Select the column ‘Genuine’ as your **labels**. A value of 0 denotes a Counterfeit sample, while a value of 1 denotes a Genuine sample.

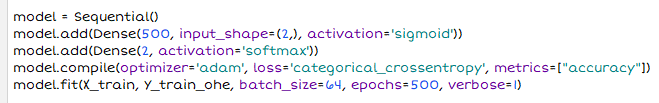
For example, the sample BN100 denotes a Genuine banknote, while BN101 denotes a Counterfeit banknote.



1. Out of the 200 samples, use only 90 Genuine and 90 Counterfeit samples for training. In other words, your training features consists of 180 rows of data (half are Genuine, the other half Counterfeit). Naturally, your training labels would then come from the selected 180 rows of data.

The remaining 20 samples (10 Genuine; 10 Counterfeit) is for testing the accuracy of your neural network.

1. Construct your neural network with Keras and feed the 180 training samples to train it. Use the XOR demo as reference. For ease of reference, part of the XOR demo code is shown below:



* Sequential() returns an empty neural network model
* Dense() allows us to add a dense (or fully-connected) hidden layer.
* The **first** Dense(), that is added to our network, has to specify our **input shape** (i.e. how many **columns/features** does your input data have)
* The **last** Dense() must specify the number of possible labels/classes for your dataset (e.g. in our XOR demo, we only have two possible labels/classes, which is 0 or 1)
* Compile your model, which is basically specifying which optimization technique (e.g. ‘adam’, ‘sgd’) to use for backpropagation, the equation of the loss function, and the accuracy metrics.
* Fitting the model means we want the training of the model to begin.

1. Once your neural network is trained (accuracy more than 0.9 or 90%), feed your test data and print out the predictions from your neural network.

Here is an example:

