

## Practical – 11

**Name:** Shivam Tawari

**Roll no:** A-58

**Aim:** Building a basic deep network using keras

**Theory:**

**Keras:**

Keras is a high-level library that's built on top of Theano or TensorFlow. It provides a scikit-learn type API (written in Python) for building Neural Networks. Developers can use Keras to quickly build neural networks without worrying about the mathematical aspects of tensor algebra, numerical techniques, and optimization methods.

The key idea behind the development of Keras is to facilitate experimentations by fast prototyping. The ability to go from an idea to result with the least possible delay is key to good research.

This offers a huge advantage for scientists and beginner developers alike because they can dive right into Deep Learning without getting their hands dirty with low-level computations. The rise in the demand for Deep Learning has resulted in the rise in demand for people skilled in Deep Learning.

**Keras features:**

- Keras is a high-level interface and uses Theano or Tensorflow for its backend.
- It runs smoothly on both CPU and GPU.
- Keras supports almost all the models of a neural network – fully connected, convolutional, pooling, recurrent, embedding, etc.

Furthermore, these models can be combined to build more complex models.

- Keras, being modular in nature, is incredibly expressive, flexible, and apt for innovative research.
- Keras is a completely Python-based framework, which makes it easy to debug and explore.

## Code:

The image displays two screenshots of a Jupyter Notebook titled "Skills Practical 11.ipynb". The interface includes a file explorer on the left showing a folder named "sample\_data" containing a file "pima-indians-diabetes.csv". The top toolbar shows options for File, Edit, View, Insert, Runtime, Tools, and Help, with a status bar indicating "All changes saved".

The first screenshot shows the initial setup of the Keras model. It includes three code cells:

- Cell [8]: Imports necessary libraries: `from numpy import loadtxt`, `from keras.models import Sequential`, and `from keras.layers import Dense`.
- Cell [9]: Loads the dataset from "pima-indians-diabetes.csv", splits it into input (X) and output (y) variables, and displays the shapes: `X = dataset[:,0:8]` and `y = dataset[:,8]`.
- Cell [10]: Defines the Keras model as a `Sequential` stack with three layers: `Dense(12, input_dim=8, activation='relu')`, `Dense(8, activation='relu')`, and `Dense(1, activation='sigmoid')`.

The second screenshot continues the implementation:

- Cell [10] (continued): Shows the model definition code.
- Cell [11]: Compiles the model with `loss='binary_crossentropy'`, `optimizer='adam'`, and `metrics=['accuracy']`. It then fits the model to the dataset: `model.fit(X, y, epochs=150, batch_size=10, verbose=0)`. The output shows the training history: `<keras.callbacks.History at 0x7f2ae7485d50>`.
- Cell [12]: Makes class predictions with the model: `predictions = (model.predict(X) > 0.5).astype(int)`. It summarizes the first 5 cases in a loop: `for i in range(5): print('%s => %d (expected %d)' % (X[i].tolist(), predictions[i], y[i]))`. The output shows the first 5 cases and their predicted vs. expected values:   
[6.0, 148.0, 72.0, 35.0, 0.0, 33.6, 0.627, 50.0] => 1 (expected 1)  
[1.0, 85.0, 66.0, 29.0, 0.0, 26.6, 0.351, 31.0] => 0 (expected 0)  
[8.0, 183.0, 64.0, 0.0, 0.0, 23.3, 0.672, 32.0] => 1 (expected 1)  
[1.0, 89.0, 66.0, 23.0, 94.0, 28.1, 0.167, 21.0] => 0 (expected 0)  
[0.0, 137.0, 40.0, 35.0, 168.0, 43.1, 2.288, 33.0] => 1 (expected 1)

**Conclusion:** Hence, successfully performed Building a basic deep network using Keras.