## PARTHIB RANJAN RAY

## RA1911003010664

**ARTIFICIAL INTELLIGENCE LAB EXPERIMENT N0: 12**

**IMPLEMENTATION OF DEEP LEARNING – KERAS-MODEL**

***Working Principle:***

Keras is a deep learning algorithm toll that wraps the efficient numerical computation libraries [Theano](https://machinelearningmastery.com/introduction-python-deep-learning-library-theano/) and [TensorFlow](https://machinelearningmastery.com/tensorflow-tutorial-deep-learning-with-tf-keras/) and allows you to define andtrain neural network models in just a few lines of code.

The steps to be followed are:

1. Load Data.
2. Define Keras Model.
3. Compile Keras Model.
4. Fit Keras Model.
5. Evaluate Keras Model.
6. Tie It All Together.
7. Make Predictions

# Source code:

# first neural network with keras make predictions from numpy import loadtxt

from keras.models import Sequential from keras.layers import Dense

# load the dataset

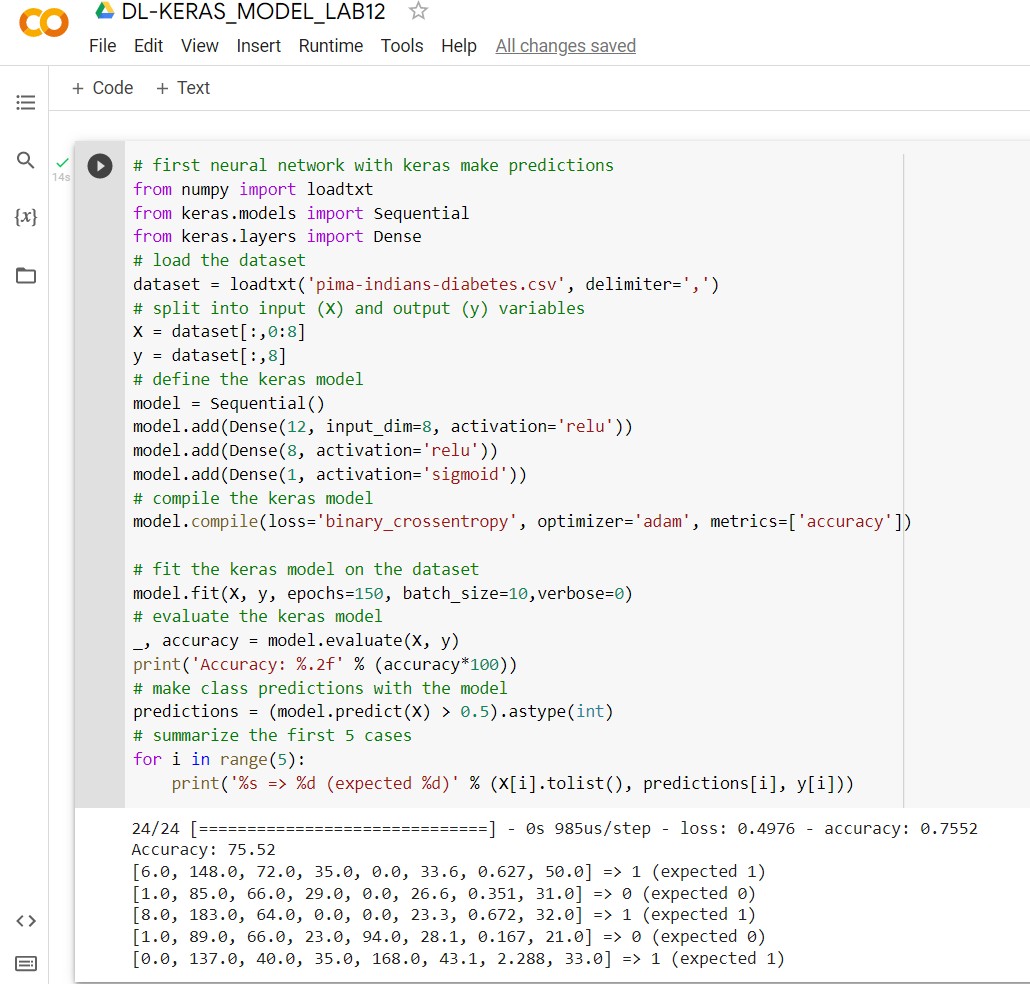
dataset = loadtxt('pima-indians-diabetes.csv', delimiter=',') # split into input (X) and output (y) variables

X = dataset[:,0:8] y = dataset[:,8]

# define the keras model model = Sequential()

model.add(Dense(12, input\_dim=8, activation='relu'))

model.add(Dense(8, activation='relu')) model.add(Dense(1, activation='sigmoid')) # compile the keras model



model.compile(loss='binary\_crossentropy', optimizer='adam', metrics=['accurac y'])

# fit the keras model on the dataset

model.fit(X, y, epochs=150, batch\_size=10,verbose=0) # evaluate the keras model

\_, accuracy = model.evaluate(X, y) print('Accuracy: %.2f' % (accuracy\*100)) # make class predictions with the model

predictions = (model.predict(X) > 0.5).astype(int) # summarize the first 5 cases

for i in range(5):

print('%s => %d (expected %d)' % (X[i].tolist(), predictions[i], y[i]))

# Output:

***Result:***

Hence, the Implementation of Deep Learning for Keras Model is done successfully.