**Shruti Karpe**

**2213182**

**AIA-1-A**

**Assignment No. 1**

**Aim:** Introduction to Keras and Tensorflow (Optional – Pytorch). Configure and use google colab and kaggle GPU

**Objectives:**

1. To configure anaconda and google colab, kaggle environment
2. To Explore TF/Keras/Pytorch libraries
3. To learn to use GPU/TPU
4. To learn and understand Git

**Theory:**

**Keras Configuration**

1. Setup Environment
2. pip install keras
3. to check whether it has installed properly: python>>import keras

Python>> print keras.\_\_version\_\_

**Tensorflow Configuration:**

1. pip install tensorflow==2.2.0
2. To verify installation: python>> import tensorflow as tf

If no error then the installation has been completed

**Colaboratory Configuration**

1. Setup the environment
2. Connect to the Drive
3. Upload the files using: from google.colab import files files.upload()

**Kaggle Configuration:**

1. Create a Kaggle Account
2. Create an Authorization token
3. Upload on Colab using the upload code
4. Make a folder and make the Json file executable
5. Get the dataset
6. Copy the API command and run on colab

**GitHub Configuration**

1. pip install git
2. Set up user profile by: global user.name “name”

Global user.email “mail”

**Dataset Attributes:**

1. Pregnancies
2. Glucose
3. BloodPressure
4. SkinThickness
5. Insulin
6. BMI
7. DiabetesPedigreeFunction
8. Age
9. Outcome

**Code:**

import pandas as pd import numpy as np import tensorflow as tf from tensorflow import keras from tensorflow.keras import layers,models from sklearn.model\_selection import train\_test\_split

from sklearn.preprocessing import StandardScaler

df=pd.read\_csv("diabetes.csv") x=df.iloc[:,0:8] y=df["Outcome"] obj=StandardScaler() x\_=obj.fit\_transform(x)

Xtrain,Xtest,Ytrain,Ytest=train\_test\_split(x\_,y,test\_size=0.1)

model=models.Sequential() model.add(layers.Dense(100,activation="relu")) #model.add(layers.Dense(75,activation="relu")) model.add(layers.Dense(50,activation="relu")) #model.add(layers.Dense(25,activation="relu")) model.add(layers.Dense(12,activation="relu")) model.add(layers.Dense(8,activation="relu"))

model.add(layers.Dense(1,activation="sigmoid"))

model.compile(optimizer="adam",loss="binary\_crossentropy",metrics=["accuracy"]

,) history=model.fit(Xtrain,Ytrain,epochs=50, validation\_data=(Xtest,Ytest)) result=model.evaluate(Xtest,Ytest)

import matplotlib.pyplot as plt plt.plot(history.history['loss'], label='loss')

#plt.plot(history.history['val\_accuracy'], label = 'val\_accuracy') plt.xlabel('Epoch') plt.ylabel('loss') plt.ylim([0, 0.8])

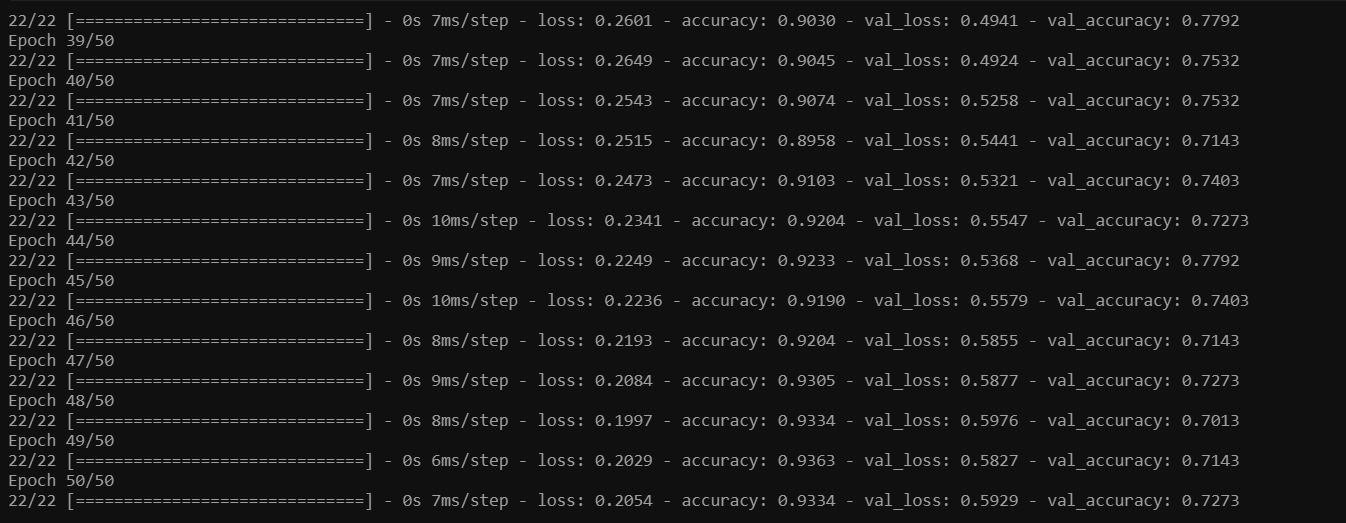
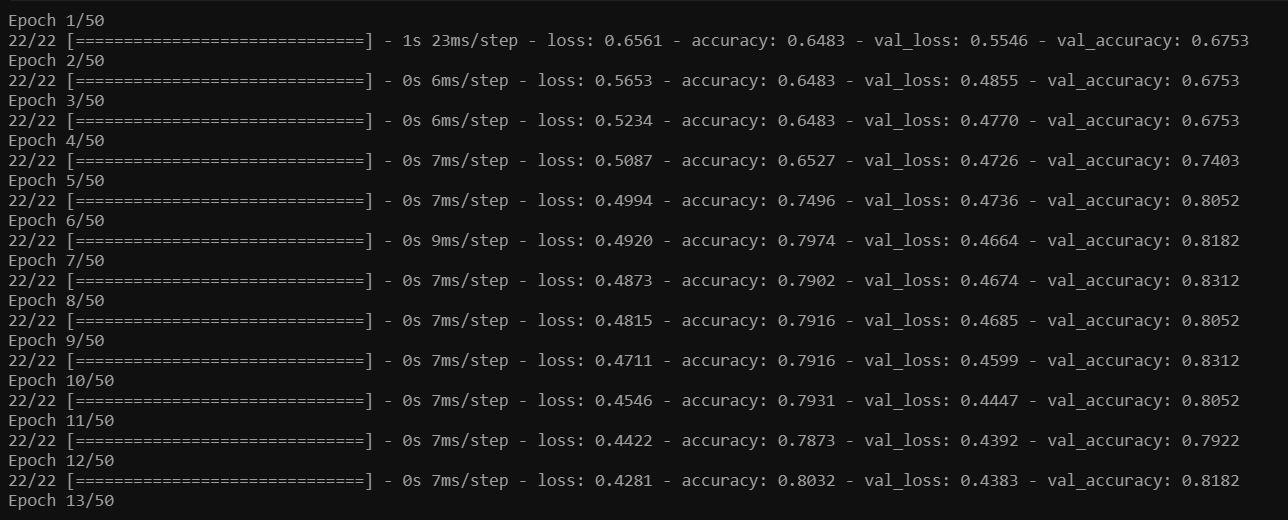
plt.legend(loc='lower right')

test\_loss, test\_acc = model.evaluate(Xtest, Ytest, verbose=2) plt.ylim([0.6,1])

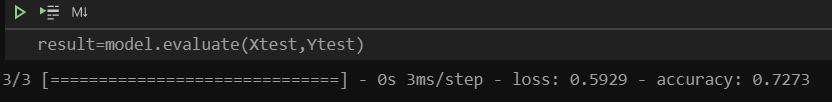
plt.plot(history.history['accuracy'], label = 'accuracy')

**Results:**

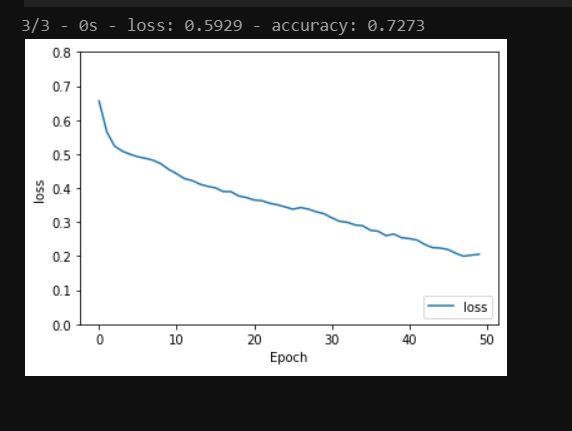
Training:



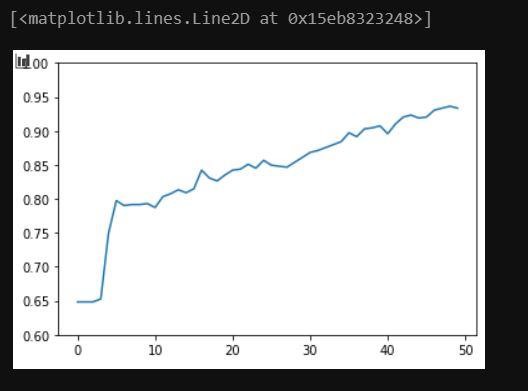
Evaluation



Loss function



Accuracy



**Conclusion:**

Thus we have understood the configuration steps of Google Colab, tensorflow, etc and learned to use tensorflow and create a model to predict the chance of diabetes or not.