University Institute of Engineering

Department of Electronics & Communication Engineering

Experiment No. :- 4

Student Name: Priyanshu Mathur	**UID:** 20BEC1073
Branch: Electronics and Communication	**Section/Group:** A
Semester: 7th	**Date of Performance:** 12/09/23
Subject Name: AIML	**Subject Code:** 20ECA-445

1. Aim of the practical: Write a program for medical diagnosis using ANN.

2. Tool Used: Google Colab

3. Theory: Keras is an open-source high-level neural networks API written in Python that is capable of running on top of other popular deep learning frameworks, such as TensorFlow, Theano, and Microsoft Cognitive Toolkit (CNTK). It is used to train the ANN model. Keras is known for its simplicity and ease of use, making it a popular choice among researchers and developers for building and prototyping deep learning models. Keras simplifies the process of building, training, and evaluating deep learning models, making it a valuable tool for both beginners and experienced deep learning practitioners. Its modular and high-level API, along with its integration with popular deep learning frameworks, has contributed to its widespread adoption in the machine learning community.

4. Steps for experiment/practical:

Step 1: - Open Google Collab

Step 2: - Create a new notebook

Step 3: - Write the code given below and run it.

5. Program Code and Simulation Output:

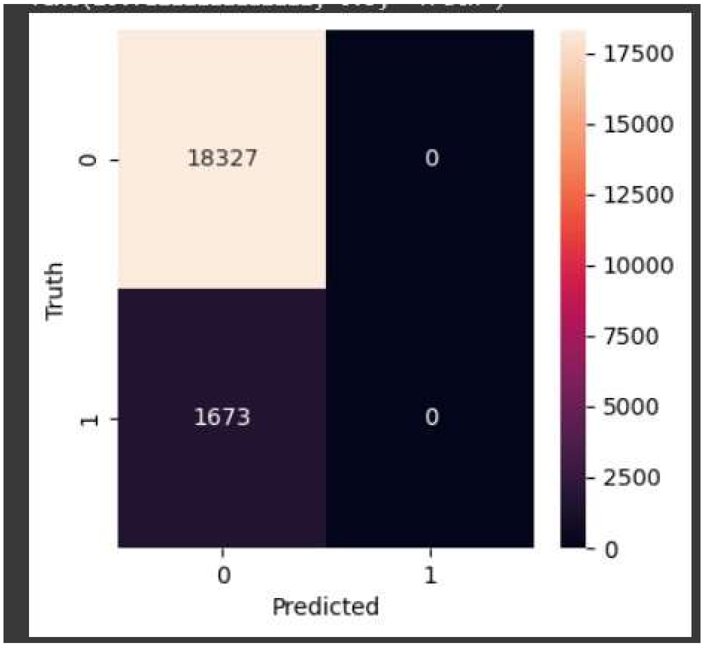
Code:-

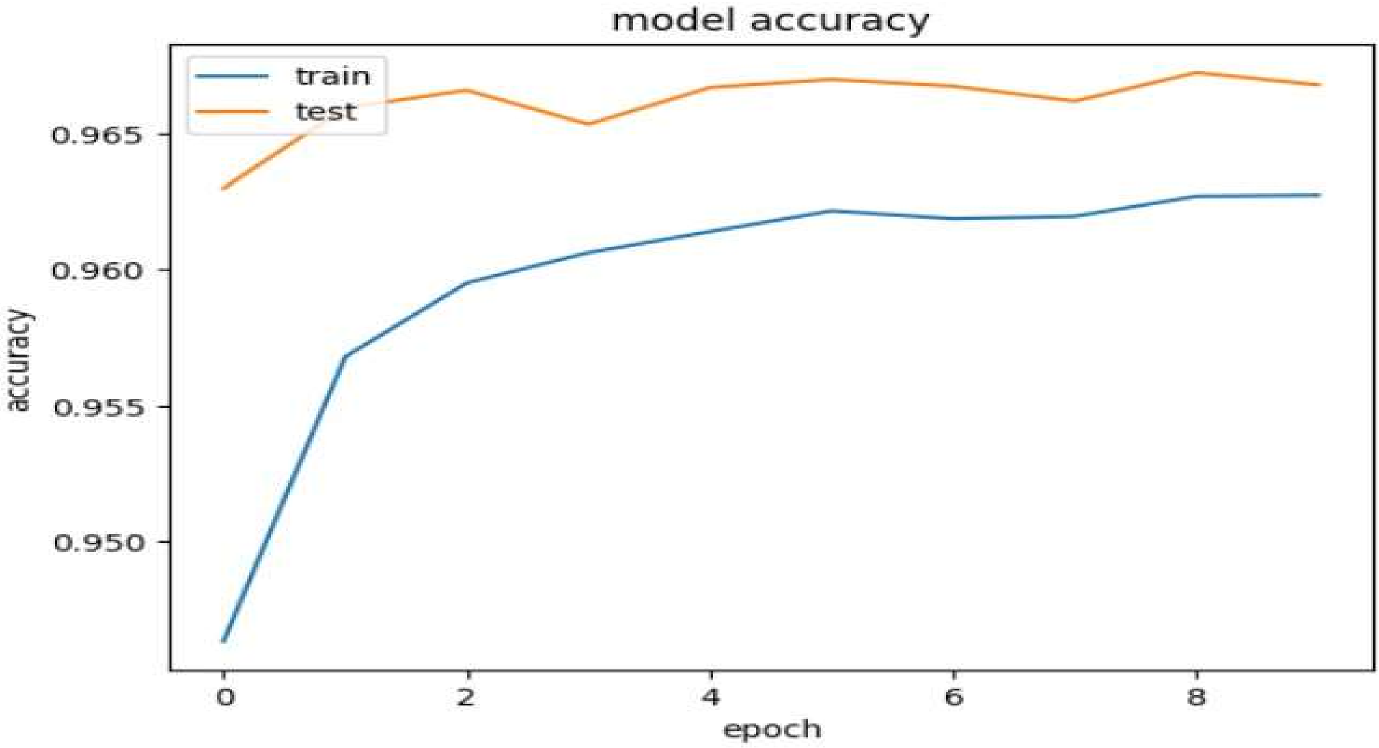
```
//Importing the libraries

import pandas as pd;
df = pd.read_csv('diabetes_prediction_dataset.csv'); df.tail(6)

//Created Depended and Independent variables y
= df['diabetes']
X = df[['age' ,'hypertension' , 'heart_disease' , 'bmi' , 'HbA1c_level' , 'blood_glucose_level']]

//Splitting Data as Training Set and Testing Set from
sklearn.model_selection import train_test_split
X_train , X_test , y_train , y_test = train_test_split(X , y , test_size = 0.2 , random_state = 100)


//Model Creating and Training import
keras
from keras.models import Sequential
from keras.layers import Dense, ReLU, Dropout, BatchNormalization from
sklearn.metrics import accuracy_score

model = Sequential()

model.add(BatchNormalization())
model.add(Dense(32 , 'swish' , 6))
model.add(Dropout(0.2))
model.add(BatchNormalization())
model.add(Dense(64 , 'swish'))
model.add(Dropout(0.2)) #model.add(Dense(128
, 'relu')) model.add(Dense(1 , 'sigmoid'))
#Complie the Model
model.compile('adam' , 'binary_crossentropy' , ['accuracy'])
#Fitting the model
model_histroy = model.fit(X_train , y_train , epochs = 10 , batch_size =  60 , validation_data =
(X_test , y_test))
```

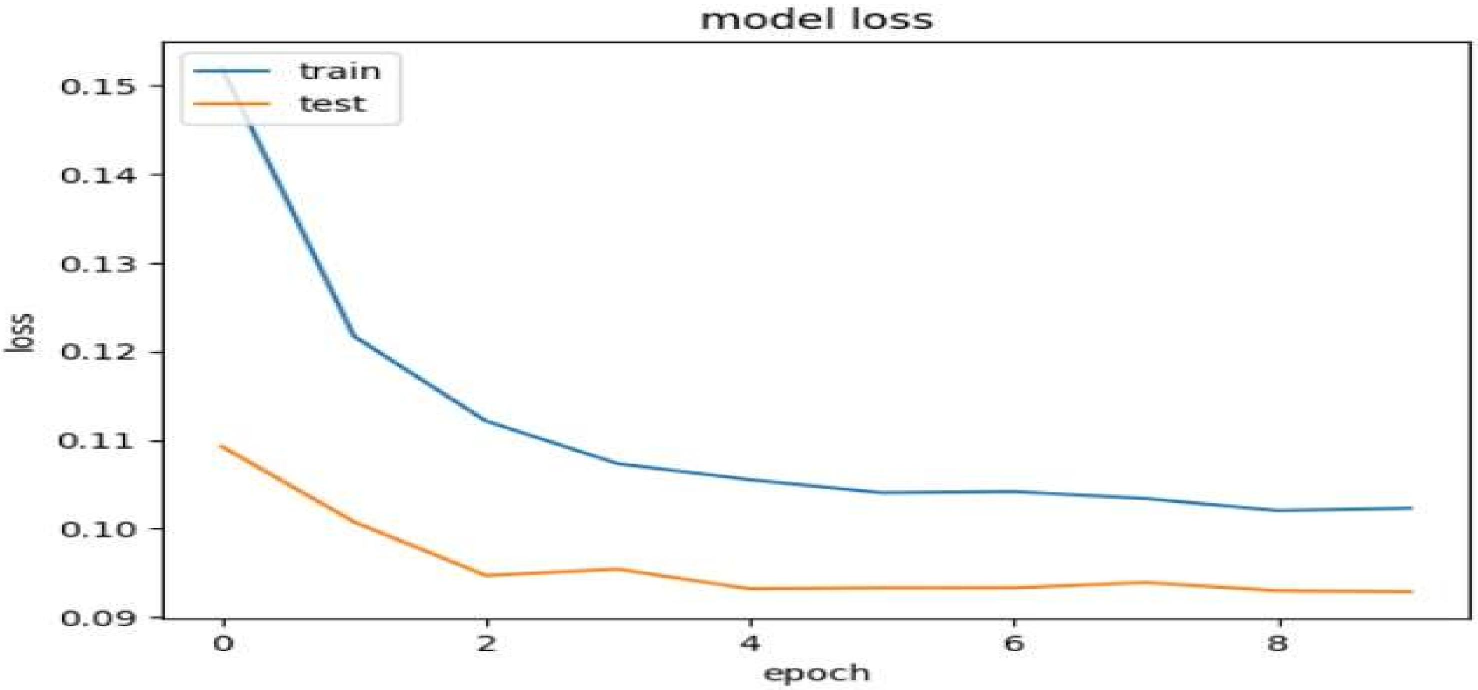
//Plotting the test and train data accuracy and loss

```
import matplotlib.pyplot as plt
plt.plot(model_history.history['accuracy'])
plt.plot(model_history.history['val_accuracy'])
plt.title('model accuracy') plt.ylabel('accuracy')
plt.xlabel('epoch')
plt.legend(['train', 'test'], loc='upper left')
plt.show()

# summarize history for loss plt.plot(model_history.history['loss'])
plt.plot(model_history.history['val_loss'])
plt.title('model loss')
plt.ylabel('loss') plt.xlabel('epoch')
plt.legend(['train', 'test'], loc='upper left')
plt.show()
```

Plot




model accuracy
train
test
0.965
0.960
0.955
0.950
accuracy
0
2
4
6
8
epoch


model loss
train
test
0.15
0.14
0.13
0.12
0.11
0.10
0.09
loss
0
2
4
6
8
epoch

Result and Discussion: - In this experiment we took a diabetic patient CSV file and created an ANN model using Keras and train the model with that particular CSV file. We created an ANN system using many layers using different number of preceprtons.We divided the CSV file into two parts train data and test data. We train the model with train data and test the model with test data from which we get our model accuracy and precision.

Learning outcomes (What I have learnt):

- Learnt about Keras and it’s different functions.
- Learn about how to create model and train it.
- Learn about different metrics and confusion matrix.