Ex.No: 1 Roll.No: 210701182

# PYTHON PROGRAM TO BUILD A SIMPLE NEURAL NETWORK WITH KERAS

Aim:

To implement a simple neural network with keras using python language,

# Procedure:

1. Import NumPy and necessary Keras modules for building the model.
2. Generate random dummy training data with 1000 samples and 10 features each.
3. Create random binary labels (0 or 1) for the training data.
4. Initialize a Sequential model for a simple feedforward neural network.
5. Add a Dense layer with 10 units and ReLU activation for the input.
6. Add another Dense layer with 1 unit and sigmoid activation for binary classiﬁcation.
7. Compile the model using Adam optimizer and binary cross-entropy loss.
8. Train the model for 20 epochs with a batch size of 10 using the training data.
9. Generate random dummy test data with 100 samples and binary labels.
10. Evaluate the model on the test data and print the loss and accuracy values.

# Code:

import numpy as np

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

# Generate some dummy data for training x\_train\_data = np.random.random((1000, 10)) y\_train\_data = np.random.randint(2, size=(1000, 1))

# Building the model model = Sequential()

model.add(Dense(10, activation='relu', input\_dim=10)) model.add(Dense(1, activation='sigmoid'))

# Compiling the model

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

# Train the model

model.ﬁt(x\_train\_data, y\_train\_data, epochs=20, batch\_size=10)

# Generate some dummy test data x\_test\_data = np.random.random((100, 10))

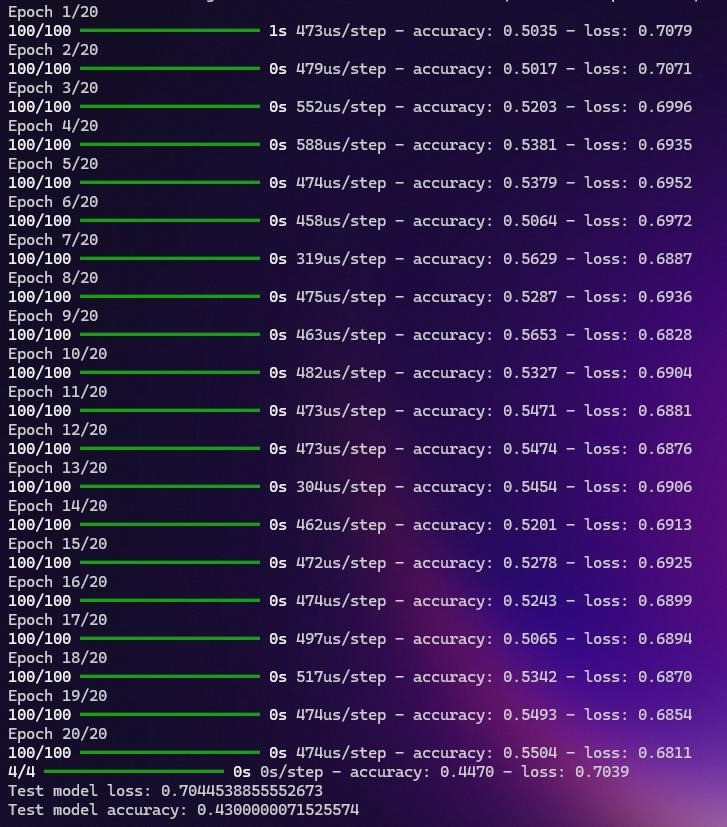
y\_test\_data = np.random.randint(2, size=(100, 1))

# Evaluating the model on the test data

loss, accuracy = model.evaluate(x\_test\_data, y\_test\_data) print('Test model loss:', loss)

print('Test model accuracy:', accuracy)

# Output:



Result:

Thus, to implement a simple neural networks using Keras in Python has been completed successfully.