Ex.No: 1 Roll.No: 210701702

PYTHON PROGRAM TO BUILD A SIMPLE NEURAL NETWORK WITH KERAS

Aim:

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

Procedure:

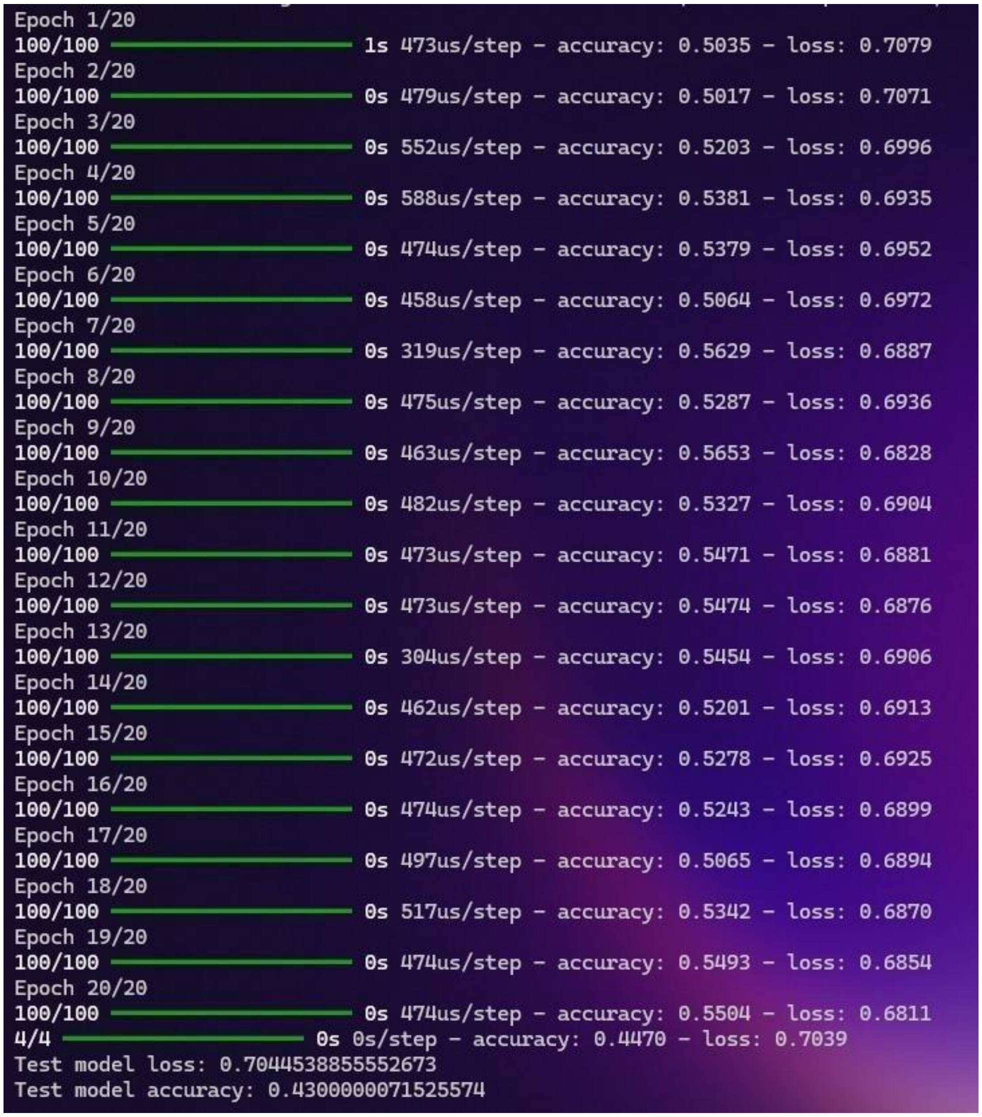
1. Import NumPy and necessary Keras modules for building the model.
2. Generate random dummy training data with 1000 samples and 10 features each.
3. Create random binary labels (0 or 1) for the training data.
4. Initialize a Sequential model for a simple feedforward neural network.
5. Add a Dense layer 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:



```
Epoch 1/20
100/100 ━━━━━━━━━━━━━━━━━━━━ 1s 473us/step - accuracy: 0.5035 - loss: 0.7079
Epoch 2/20
100/100 ━━━━━━━━━━━━━━━━━━━━ 0s 479us/step - accuracy: 0.5017 - loss: 0.7071
Epoch 3/20
100/100 ━━━━━━━━━━━━━━━━━━━━ 0s 552us/step - accuracy: 0.5203 - loss: 0.6996
Epoch 4/20
100/100 ━━━━━━━━━━━━━━━━━━━━ 0s 588us/step - accuracy: 0.5381 - loss: 0.6935
Epoch 5/20
100/100 ━━━━━━━━━━━━━━━━━━━━ 0s 474us/step - accuracy: 0.5379 - loss: 0.6952
Epoch 6/20
100/100 ━━━━━━━━━━━━━━━━━━━━ 0s 458us/step - accuracy: 0.5064 - loss: 0.6972
Epoch 7/20
100/100 ━━━━━━━━━━━━━━━━━━━━ 0s 319us/step - accuracy: 0.5629 - loss: 0.6887
Epoch 8/20
100/100 ━━━━━━━━━━━━━━━━━━━━ 0s 475us/step - accuracy: 0.5287 - loss: 0.6936
Epoch 9/20
100/100 ━━━━━━━━━━━━━━━━━━━━ 0s 463us/step - accuracy: 0.5653 - loss: 0.6828
Epoch 10/20
100/100 ━━━━━━━━━━━━━━━━━━━━ 0s 482us/step - accuracy: 0.5327 - loss: 0.6904
Epoch 11/20
100/100 ━━━━━━━━━━━━━━━━━━━━ 0s 473us/step - accuracy: 0.5471 - loss: 0.6881
Epoch 12/20
100/100 ━━━━━━━━━━━━━━━━━━━━ 0s 473us/step - accuracy: 0.5474 - loss: 0.6876
Epoch 13/20
100/100 ━━━━━━━━━━━━━━━━━━━━ 0s 304us/step - accuracy: 0.5454 - loss: 0.6906
Epoch 14/20
100/100 ━━━━━━━━━━━━━━━━━━━━ 0s 462us/step - accuracy: 0.5201 - loss: 0.6913
Epoch 15/20
100/100 ━━━━━━━━━━━━━━━━━━━━ 0s 472us/step - accuracy: 0.5278 - loss: 0.6925
Epoch 16/20
100/100 ━━━━━━━━━━━━━━━━━━━━ 0s 474us/step - accuracy: 0.5243 - loss: 0.6899
Epoch 17/20
100/100 ━━━━━━━━━━━━━━━━━━━━ 0s 497us/step - accuracy: 0.5065 - loss: 0.6894
Epoch 18/20
100/100 ━━━━━━━━━━━━━━━━━━━━ 0s 517us/step - accuracy: 0.5342 - loss: 0.6870
Epoch 19/20
100/100 ━━━━━━━━━━━━━━━━━━━━ 0s 474us/step - accuracy: 0.5493 - loss: 0.6854
Epoch 20/20
100/100 ━━━━━━━━━━━━━━━━━━━━ 0s 474us/step - accuracy: 0.5504 - loss: 0.6811
4/4 ━━━━━━━━━━━━━━━━━━━━ 0s 0s/step - accuracy: 0.4470 - loss: 0.7039
Test model loss: 0.7044538855552673
Test model accuracy: 0.4300000071525574
```

Result:

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