Ex.No.2

BUILD A SIMPLE NEURAL NETWORK WITH KERAS

AIM:

To build a simple neural network using Keras/TensorFlow

PROCEDURE:

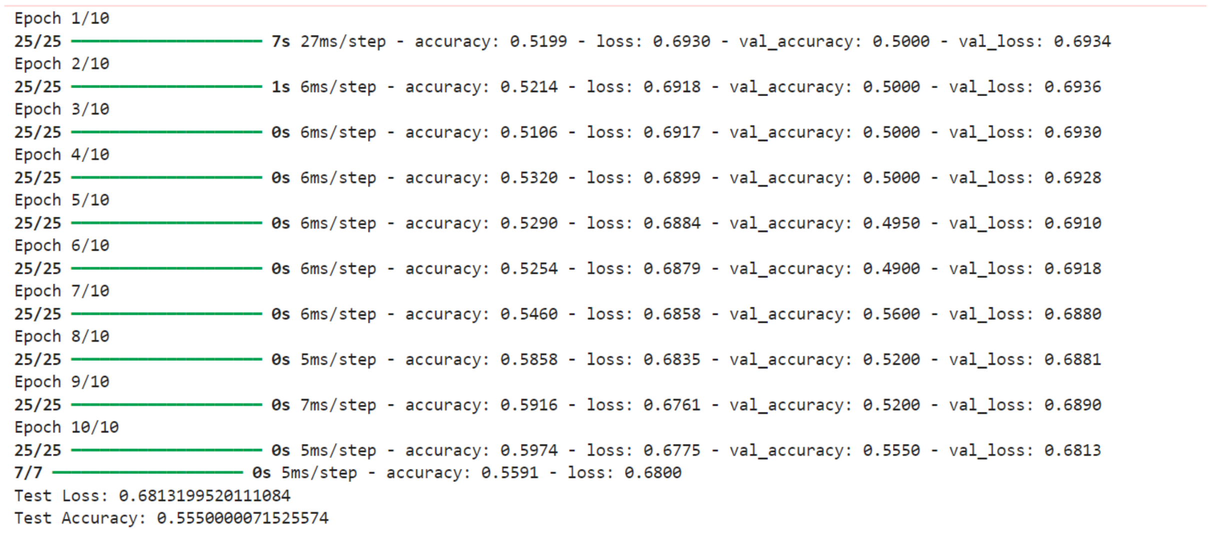
1. Create a dataset using rand() and randint() for the independent variable and dependent variable respectively.
2. Split the dataset into training data and test data.
3. Build a simple neural network model using Keras/TensorFlow.
4. Compile and fit the model on the training data.
5. Perform prediction with the test data.
6. Calculate performance metrics.

PROGRAM:

```
import numpy as np
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
from sklearn.model_selection import train_test_split

np.random.seed(42)

X = np.random.rand(1000, 10)
y = np.random.randint(0, 2, 1000)
```

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42)

model = Sequential()

model.add(Dense(64, input_shape=(10,), activation='relu'))
model.add(Dense(32, activation='relu'))
model.add(Dense(32, activation='relu'))
model.add(Dense(16, activation='relu'))
model.add(Dense(16, activation='relu'))
model.add(Dense(8, activation='relu'))
model.add(Dense(1, activation='sigmoid'))

model.compile(optimizer='adam',
loss='binary_crossentropy',
metrics=['accuracy'])

model.fit(X_train, y_train, epochs=10, batch_size=32, validation_data=(X_test,
y_test))

loss, accuracy = model.evaluate(X_test, y_test)
print(f"Test Loss: {loss}")
print(f"Test Accuracy: {accuracy}")
```

OUTPUT:



```
Epoch 1/10
25/25 ━━━━━━━━━━━━━━━━━━━━ 7s 27ms/step - accuracy: 0.5199 - loss: 0.6930 - val_accuracy: 0.5000 - val_loss: 0.6934
Epoch 2/10
25/25 ━━━━━━━━━━━━━━━━━━━━ 1s 6ms/step - accuracy: 0.5214 - loss: 0.6918 - val_accuracy: 0.5000 - val_loss: 0.6936
Epoch 3/10
25/25 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - accuracy: 0.5106 - loss: 0.6917 - val_accuracy: 0.5000 - val_loss: 0.6930
Epoch 4/10
25/25 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - accuracy: 0.5320 - loss: 0.6899 - val_accuracy: 0.5000 - val_loss: 0.6928
Epoch 5/10
25/25 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - accuracy: 0.5290 - loss: 0.6884 - val_accuracy: 0.4950 - val_loss: 0.6910
Epoch 6/10
25/25 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - accuracy: 0.5254 - loss: 0.6879 - val_accuracy: 0.4900 - val_loss: 0.6918
Epoch 7/10
25/25 ━━━━━━━━━━━━━━━━━━━━ 0s 6ms/step - accuracy: 0.5460 - loss: 0.6858 - val_accuracy: 0.5600 - val_loss: 0.6880
Epoch 8/10
25/25 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - accuracy: 0.5858 - loss: 0.6835 - val_accuracy: 0.5200 - val_loss: 0.6881
Epoch 9/10
25/25 ━━━━━━━━━━━━━━━━━━━━ 0s 7ms/step - accuracy: 0.5916 - loss: 0.6761 - val_accuracy: 0.5200 - val_loss: 0.6890
Epoch 10/10
25/25 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - accuracy: 0.5974 - loss: 0.6775 - val_accuracy: 0.5550 - val_loss: 0.6813
7/7 ━━━━━━━━━━━━━━━━━━━━ 0s 5ms/step - accuracy: 0.5591 - loss: 0.6800
Test Loss: 0.6813199520111084
Test Accuracy: 0.5550000071525574
```

RESULT:

Thus, a simple neural network using Keras/TensorFlow was built successfully.