

Ex.No: 1

Roll.No: 210701157

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 1ms/step - accuracy: 0.5213 - loss: 0.7028
Epoch 2/20
100/100 ----- 0s 2ms/step - accuracy: 0.4836 - loss: 0.6988
Epoch 3/20
100/100 ----- 0s 1ms/step - accuracy: 0.4801 - loss: 0.6978
Epoch 4/20
100/100 ----- 0s 806us/step - accuracy: 0.4912 - loss: 0.6946
Epoch 5/20
100/100 ----- 0s 1ms/step - accuracy: 0.4618 - loss: 0.6950
Epoch 6/20
100/100 ----- 0s 1ms/step - accuracy: 0.4927 - loss: 0.6923
Epoch 7/20
100/100 ----- 0s 782us/step - accuracy: 0.4859 - loss: 0.6935
Epoch 8/20
100/100 ----- 0s 622us/step - accuracy: 0.4640 - loss: 0.6944
Epoch 9/20
100/100 ----- 0s 627us/step - accuracy: 0.4866 - loss: 0.6946
Epoch 10/20
100/100 ----- 0s 639us/step - accuracy: 0.4931 - loss: 0.6912
Epoch 11/20
100/100 ----- 0s 630us/step - accuracy: 0.5185 - loss: 0.6919
Epoch 12/20
100/100 ----- 0s 624us/step - accuracy: 0.5148 - loss: 0.6928
Epoch 13/20
100/100 ----- 0s 632us/step - accuracy: 0.5652 - loss: 0.6912
Epoch 14/20
100/100 ----- 0s 593us/step - accuracy: 0.5125 - loss: 0.6920
Epoch 15/20
100/100 ----- 0s 542us/step - accuracy: 0.5681 - loss: 0.6899
Epoch 16/20
100/100 ----- 0s 552us/step - accuracy: 0.5297 - loss: 0.6909
Epoch 17/20
100/100 ----- 0s 505us/step - accuracy: 0.5706 - loss: 0.6880
Epoch 18/20
100/100 ----- 0s 910us/step - accuracy: 0.5579 - loss: 0.6893
Epoch 19/20
100/100 ----- 0s 620us/step - accuracy: 0.5019 - loss: 0.6917
Epoch 20/20
100/100 ----- 0s 573us/step - accuracy: 0.5223 - loss: 0.6889
4/4 ----- 0s 2ms/step - accuracy: 0.4170 - loss: 0.7022
Test model loss: 0.7003750801086426
Test model accuracy: 0.41999998688697815
```

Result:

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