EX. NO : 11

220801179

DATE : 17/05/2024

SIMPLE PROGRAM IN ARTIFICIAL NEURAL NETWORK

AIM:

To Create a simple Artificial Neural Network in Python.

ALGORITHM:

1. Import the necessary libraries.
2. Load and preprocess the data.
3. Build the ANN model.
4. Compile the model.
5. Train the model.
6. Evaluate the model.

PROGRAM:

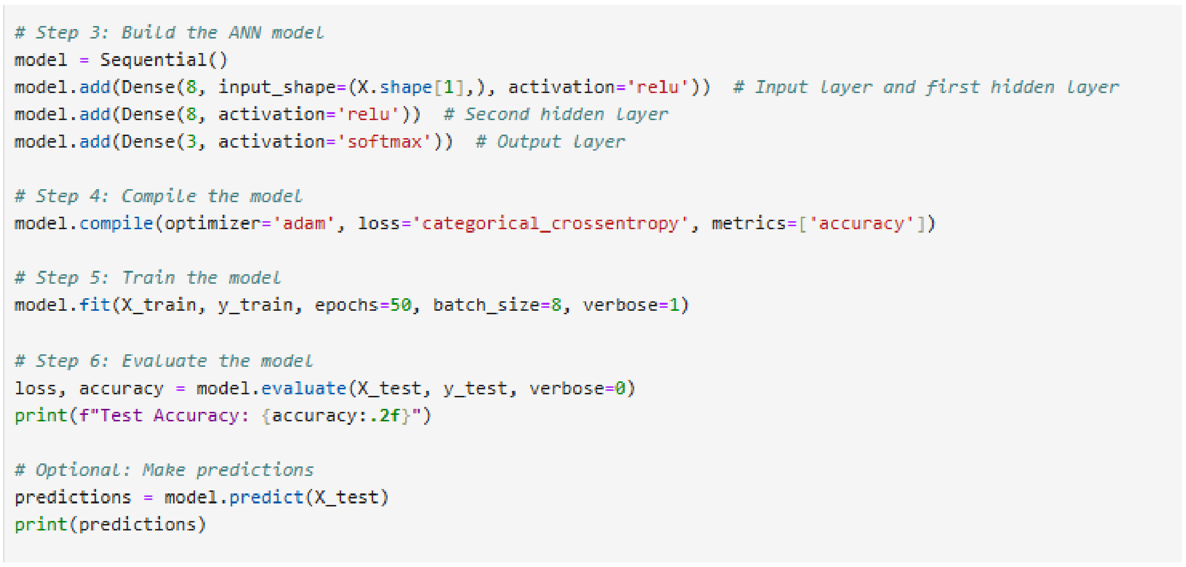
```python
#Step 1: Import the necessary libraries
import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import OneHotEncoder, StandardScaler

# Step 2: Load and preprocess the data
iris = load_iris()
X = iris.data
y = iris.target.reshape(-1, 1)

# One-hot encode the target variable
encoder = OneHotEncoder()
y = encoder.fit_transform(y)

# Standardize the features
scaler = StandardScaler()
X = scaler.fit_transform(X)

# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```



```python
# Step 3: Build the ANN model
model = Sequential()
model.add(Dense(8, input_shape=(X.shape[1],), activation='relu'))  # Input layer and first hidden layer
model.add(Dense(8, activation='relu'))  # Second hidden layer
model.add(Dense(3, activation='softmax'))  # Output layer

# Step 4: Compile the model
model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])

# Step 5: Train the model
model.fit(X_train, y_train, epochs=50, batch_size=8, verbose=1)

# Step 6: Evaluate the model
loss, accuracy = model.evaluate(X_test, y_test, verbose=0)
print(f"Test Accuracy: {accuracy:.2f}")

# Optional: Make predictions
predictions = model.predict(X_test)
print(predictions)
```

RESULT:

This program is executed successfully.