Vivekanand Education Society's
Institute of Technology
Approved by AICTE & Affiliated to University of Mumbai

Artificial Intelligence and Data Science Department

Subject/Odd Sem 2023-23/Experiment 1

Name : Prasad Jawale	**Class/Roll No. :** D16AD 20	**Grade :**

Title of Experiment : Explore Python Libraries TensorFlow and keras

Objective of Experiment : The objective of utilizing TensorFlow and Keras is to develop and train deep learning models to solve various machine learning tasks. These libraries provide tools and APIs that simplify the process of building, training, and evaluating neural networks. By leveraging these libraries, the aim is to achieve high-performance models that can make accurate predictions or classifications based on input data.

Outcome of Experiment : The outcome of using TensorFlow and Keras is the creation of well-performing machine learning models that can address specific problems. These models can be deployed in various applications, such as image classification, natural language processing, speech recognition, and more. The outcome also includes the ability to fine-tune pre-trained models for specific tasks, saving time and computational resources.

Problem Statement : To explore python libraries TensorFlow and Keras

Subject/Odd Sem 2023-23/Experiment 1

Description / Theory :

TensorFlow:

TensorFlow is an open-source machine learning framework developed by Google. It provides a comprehensive set of tools and libraries for building and training various types of machine learning models, with a focus on deep learning. TensorFlow is designed to be scalable and flexible, allowing you to work with neural networks and other machine learning algorithms efficiently.

Key features of TensorFlow:

Graph Computation: TensorFlow uses a symbolic graph representation of computations. This allows for efficient execution on CPUs, GPUs, and TPUs (Tensor Processing Units) without having to rewrite the code.

High-level APIs: TensorFlow provides both high-level and low-level APIs. High-level APIs, like Keras (which is now tightly integrated with TensorFlow), offer a more user-friendly interface for building neural networks quickly.

TensorBoard: A visualization tool that helps you monitor and analyze the training progress and performance of your models.

Distributed Computing: TensorFlow supports distributed computing, enabling you to train models on clusters of machines.

Flexibility: TensorFlow allows you to define custom operations and loss functions, giving you a high degree of flexibility when designing your models.

SavedModel Format: TensorFlow provides a standardized format for saving and exporting trained models, making it easier to deploy them in various environments.

Subject/Odd Sem 2023-23/Experiment 1

Keras:

Keras is an open-source high-level neural networks API written in Python. Originally developed as an independent library, Keras was later integrated as a part of TensorFlow's core library. It offers an intuitive and user-friendly interface for building and training neural networks. Keras abstracts away much of the complexity involved in designing neural networks, making it an excellent choice for beginners and rapid prototyping.

Key features of Keras:

Simple Interface: Keras provides a user-friendly, modular, and intuitive interface for building neural networks. You can define and customize layers, activations, loss functions, and optimizers easily.

Modularity: Keras allows you to create models through the sequential API (linear stack of layers) or the functional API (more flexible for creating complex architectures with multiple inputs/outputs).

Pre-trained Models: Keras offers pre-trained models for various tasks like image classification, object detection, and more. These models are ready to use and can be fine-tuned for specific tasks.

Easy Transfer Learning: Transfer learning is simplified in Keras, allowing you to leverage pre-trained models and adapt them to new tasks.

Integration with TensorFlow: Keras is now the official high-level API for TensorFlow, so you can use TensorFlow's powerful capabilities alongside Keras's simplicity.

Vivekanand Education Society's
Institute of Technology
Approved by AICTE & Affiliated to University of Mumbai

Artificial Intelligence and Data Science Department

Subject/Odd Sem 2023-23/Experiment 1

Program:

In [1]:
```
# Import necessary libraries
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.datasets import load_iris
import tensorflow as tf
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense
from tensorflow.keras.utils import to_categorical
```

In [2]:
```
# Load the Iris dataset
iris = load_iris()
X = iris.data
y = iris.target
```

In [3]:
```
# 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)

# Standardize the features
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)

# One-hot encode the target labels
y_train = to_categorical(y_train)
y_test = to_categorical(y_test)
```

In [4]:
```
# Create the DNN model
model = Sequential([
    Dense(units=8, activation='relu', input_shape=(4,)),  # 1st hidden layer
    Dense(units=3, activation='softmax')  # Output layer with 3 classes
])

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

In [7]:
```
# Train the model
model.fit(X_train, y_train, epochs=50, batch_size=10, validation_split=0.2)

# Evaluate the model
loss, accuracy = model.evaluate(X_test, y_test)
print(f"Test loss: {loss:.4f}, Test accuracy: {accuracy:.4f}")
```

```
10/10 [==============================] - 0s 6ms/step - loss: 0.1622 - accuracy: 0.9688 - val_loss: 0.3224 - val_accuracy: 0.9
583
Epoch 46/50
10/10 [==============================] - 0s 6ms/step - loss: 0.1611 - accuracy: 0.9688 - val_loss: 0.3192 - val_accuracy: 0.9
583
Epoch 47/50
10/10 [==============================] - 0s 7ms/step - loss: 0.1600 - accuracy: 0.9688 - val_loss: 0.3196 - val_accuracy: 0.9
583
Epoch 48/50
10/10 [==============================] - 0s 7ms/step - loss: 0.1586 - accuracy: 0.9688 - val_loss: 0.3174 - val_accuracy: 0.9
583
Epoch 49/50
10/10 [==============================] - 0s 7ms/step - loss: 0.1573 - accuracy: 0.9688 - val_loss: 0.3173 - val_accuracy: 0.9
583
Epoch 50/50
10/10 [==============================] - 0s 7ms/step - loss: 0.1562 - accuracy: 0.9688 - val_loss: 0.3160 - val_accuracy: 0.9
583
1/1 [==============================] - 0s 33ms/step - loss: 0.1418 - accuracy: 0.9667
Test loss: 0.1418, Test accuracy: 0.9667
```

Results and Discussions : In summary, TensorFlow and Keras are powerful tools for building and training neural networks and other machine learning models. TensorFlow provides a comprehensive ecosystem for deep learning, while Keras offers a more user-friendly interface for quickly prototyping and experimenting with neural network architectures. With their combined strengths, you can efficiently tackle a wide range of machine learning tasks.