

EXPERIMENT-8

Aim: Develop a small application with keras

Description:

Developing a small application with Keras involves utilizing the Keras library, which is a high-level API built on top of TensorFlow or other backend frameworks like Theano or Microsoft Cognitive Toolkit (CNTK). Begin by importing the necessary modules from Keras, such as `Sequential` for creating a linear stack of layers or `Model` for more complex architectures. Define your model's architecture by adding layers such as dense (fully connected), convolutional, recurrent, or other specialized layers, along with activation functions and input/output configurations. Compile the model with an optimizer (e.g., Adam, SGD), loss function (e.g., categorical cross-entropy for classification, mean squared error for regression), and optional metrics (e.g., accuracy). Prepare your data, including feature scaling, splitting into training and testing sets, and encoding labels if necessary. Train the model using the `fit` method on your training data, specifying the number of epochs and batch size. Evaluate the model's performance on the test set using the `evaluate` method. Finally, use the trained model to make predictions on new data or integrate it into an application for tasks like image classification, sentiment analysis, or recommendation systems. Keras provides an intuitive and user-friendly interface for building and deploying neural networks, making it ideal for rapid prototyping and development of machine learning applications.

Source Program:

```
pip install tensorflow keras pillow
import tensorflow as tf
from tensorflow import keras
from tensorflow.keras import datasets, layers, models
import matplotlib.pyplot as plt
from google.colab import files
from IPython.display import Image
from PIL import Image
import numpy as np

model = keras.applications.VGG16(weights='imagenet')

(train_images, train_labels), (test_images, test_labels) = datasets.cifar10.load_data()
train_images, test_images = train_images/255.0, test_images/255.0

img = Image.open('dog.jpg')
img = img.resize((224, 224))
img_array = keras.preprocessing.image.img_to_array(img)
img_array = np.expand_dims(img_array, axis=0)
img_array = keras.applications.vgg16.preprocess_input(img_array)
predictions = model.predict(img_array)
decoded_predictions = keras.applications.vgg16.decode_predictions(predictions, top=3)[0]
for class_id, class_name, probability in decoded_predictions:
    print(f'{class_name} ({class_id}): {probability * 100:.2f}%')
```

Output:

```
golden_retriever (n02099601): 51.63%
beagle (n02088364): 10.98%
Brittany_spaniel (n02101388): 10.88
```