Documentation for Image Data Augmentation

TensorFlow

Overview

To achieve data augmentation on images using TensorFlow, we used the `ImageDataGenerator`. It applies several transformations to generate new images from existing ones, enhancing the dataset for training machine learning models. The transformations include rotation, width shift, height shift, zoom, and horizontal flip. The augmented images are saved to a specified directory.

Requirements

- TensorFlow
- NumPy
- PIL (Python Imaging Library)
- os module

Steps

1. Import Necessary Libraries

First, import the required libraries.

2. Create Output Directory

Create a directory to save the augmented images.

3. Load and Preprocess Images

Load images from a specified folder and convert them to NumPy arrays.

4. Define ImageDataGenerator with Transformations

Set up the `ImageDataGenerator` with specified transformations.

5. Generate Augmented Images

Generate and save augmented images for each input image. In this script, five augmented images are generated per input image. At first the images augmented on are all gotten from memory. This is not very efficient as with large datasets, its impossible to load all the images in memory.

6. Process Each Image Individually (for Efficient Handling)

To handle images efficiently, especially with larger datasets, process each image one at a time, therefore only loading 1 at a time in memory

7. Process Images in Batches(also for Efficient Handling)

Another way to handle images efficiently is to load them in smaller batches, therefore having less images in memory, but still faster than 1 at a time.

Explanation of Transformations

- **rotation_range=40**: Randomly rotates images within the range of -40 to +40 degrees.
- **width_shift_range=0.2**: Shifts images horizontally by up to 20% of the width.
- **height_shift_range=0.2**: Shifts images vertically by up to 20% of the height.
- **zoom_range=0.2**: Zooms in on images by up to 20%.
- **horizontal_flip=True**: Randomly flips images horizontally.
- **fill_mode='nearest'**: Fills newly created pixels after transformations with the nearest pixel values from the boundary of the original image.

Conclusion

This script demonstrates how to efficiently augment image data using TensorFlow's `ImageDataGenerator`. It loads images from a directory, applies specified transformations, and saves multiple augmented versions of each image to a new directory. The process is designed to handle large datasets efficiently by processing one image at a time.

Use of Generative AI

Generative AI was used to generate this documentation. The code was copy and pasted into chat-gpt for which a layout to use was generated. Then the document was read through and was modified to ensure its accurate to the code. Some sections were also added.