Data Preprocessing-Data Augmentation

Data Augmentation: Enhancing Your Dataset

Data augmentation is a popular technique in machine learning and deep learning used to artificially increase the size of your training dataset without collecting more data. The goal is to expose the model to diverse variations of the same input, thereby enhancing its robustness and reducing overfitting.

Types of Data Augmentation

There are several types of data augmentation techniques:

1. **Spatial Transforms**:
    - Rotation
    - Translation (movement)
    - Scaling
    - Flipping (horizontal and vertical)
2. **Color Transforms**:
    - Brightness adjustment
    - Contrast change
    - Saturation variation
3. **Noise Addition**:
    - Gaussian noise
    - Salt-and-pepper noise
4. **Time-based Transformations** (for time-series data):
    - Time warping
    - Temporal shifting

Example: Data Augmentation with Rotation and Flipping for Image Classification

Suppose we have a dataset of images with classes (e.g., cats, dogs). We want to use data augmentation to create more samples from our original dataset.

```
 import tensorflow as tf
from tensorflow import keras
from tensorflow.keras import layers
import matplotlib.pyplot as plt
import numpy as np

# Load your image dataset and preprocess it
(X_train, y_train), (X_test, y_test) = ...  # Load dataset here

# Define a function to apply data augmentation
def augment_data(image):
    # Randomly rotate the image between -30° and +30°
    rotation_angle = np.random.uniform(-0.5236, 0.5236)

    # Randomly flip the image horizontally
    if np.random.rand() < 0.5:
        image = tf.image.flip_left_right(image)

    return tf.image.rotate(image, angle=rotation_angle)

# Apply data augmentation to training images
augmented_X_train = []
for x in X_train:
    augmented_x = augment_data(x)
    augmented_X_train.append(augmented_x)

# Convert the list of tensors into a single tensor
augmented_X_train = tf.stack(augmented_X_train, axis=0)
```

By applying these transformations to your original dataset, you increase its size and diversity without collecting more data. This is particularly useful in computer vision tasks where image variations can be significant.

When implementing data augmentation, ensure that the transformed images are still recognizable by your model for each class.

Curated by Brajesh Kumar