EE 634/734 Intro to Neural Networks
Assignment #4: Data Augmentation

Due Date: Monday, November 6, 2023

Objective: The aim of this assignment is to understand the nuances of Convolutional Neural Networks (CNNs) and Dense networks using the CIFAR-10 dataset and TensorFlow.

Dataset Description: The CIFAR-10 dataset consists of 60,000 32x32 color images in 10 classes, with 6,000 images per class. The classes include airplane, automobile, bird, cat, deer, dog, frog, horse, ship, truck. To access the dataset, you can use the following instructions:

```
# Load the CIFAR-10 dataset
from tensorflow.keras.datasets import cifar10
(train_images, train_labels), (test_images, test_labels) = cifar10.load_data()
```

Objective:

In this assignment, you will build Convolutional Neural Network (CNN) models to classify images from the CIFAR-10 dataset. You will explore the impact of data augmentation on model performance and discuss how it helps mitigate overfitting.

1. Data Loading and Preprocessing:
    - Download the CIFAR-10 dataset if not already available.
    - Normalize the pixel values to be between 0 and 1.

2. Build a baseline CNN model with the following architecture:
    - Convolutional Layer 1: 32 filters, (3x3) kernel, ReLU activation
    - MaxPooling Layer 1: (2x2) pool size
    - Convolutional Layer 2: 64 filters, (3x3) kernel, ReLU activation
    - MaxPooling Layer 2: (2x2) pool size
    - Flatten Layer
    - Dense Layer: 128 units, ReLU activation
    - Output Layer: 10 units (for 10 classes), softmax activation
    - Compile the baseline model with appropriate loss and metrics.

3. Implement data augmentation using the Keras ImageDataGenerator:
    - Random horizontal flipping
    - Random rotation within a specified range
    - Random width and height shifts

4. Augmented CNN Model:
    - Build a CNN model like the baseline model but include data augmentation using the ImageDataGenerator for the training data.
    - Compile the augmented model with the same loss and metrics as the baseline model.
    - Implement data loading and preprocessing in two ways: a) Reading the image data from the disk (using ImageDataGenerator with flow_from_directory), and b) Loading the entire dataset into memory.

5. Training and Evaluation:
    - Train both the baseline and augmented models on the training data. Monitor and plot the training and validation accuracy and loss for each model.
    - Evaluate both models on the testing data and report the test accuracy for each.

6. Overfitting Analysis:
    - Compare the training and validation curves for the baseline and augmented models. Discuss any signs of overfitting observed during training.
    - Explain how data augmentation helped mitigate overfitting in the augmented model.

7. Data Augmentation Visualization:
    - Show a few examples of original images and their augmented counterparts.

8. Discussion:
    - Summarize your findings and compare the performance of the baseline and augmented models.
    - Reflect on the benefits and challenges of using data augmentation in training CNN models.

9. Submission:
    - Prepare a report that includes code, visualizations, and explanations for each step.
    - Include comments in your code to explain your implementation.

To set up the CIFAR-10 dataset on your disk:

Create a root directory where you will store the CIFAR-10 dataset. Inside this directory, create subdirectories for the training and testing datasets, as well as subdirectories for each class label. The structure should look like this:

- cifar-10-data/
  - train/
    - airplane/

  - automobile/
  - bird/
  - cat/
  - deer/
  - dog/
  - frog/
  - horse/
  - ship/
  - truck/
- test/
  - airplane/
  - automobile/
  - bird/
  - cat/
  - deer/
  - dog/
  - frog/
  - horse/
  - ship/
  - truck/