Deep Learning Course

Assignment 6

ASL Alphabet Dataset

Introduction

The American Sign Language (ASL) Alphabet dataset, a collection of images representing hand gestures corresponding to the 26 letters of the English alphabet, from A to Z, is not just a dataset for machine learning. It's a powerful tool for developing algorithms that can recognize gestures and translate sign language. In this assignment, you will harness the potential of the ASL Alphabet dataset to train a Convolutional Neural Network (CNN) using PyTorch. The goal is to accurately classify these hand gestures into their respective letters, a task that has significant real-world applications in sign language translation and communication.

Objective

The primary goal of this assignment is to familiarize students with:

- The ASL Alphabet dataset.
- Implementing a Convolutional Neural Network (CNN) using PyTorch.
- Training and optimizing the CNN model to achieve higher accuracy on the ASL Alphabet dataset.

Dataset Description

The ASL Alphabet dataset comprises images representing hand gestures corresponding to the 26 English alphabet letters in American Sign Language (ASL). The dataset comprises 26 classes, each representing a unique letter from A to Z.

Tasks

1. Data Preparation

- Prepare the ASL Alphabet dataset in Kaggle to start work with it.
- Preprocess the dataset (e.g., normalization, resizing).
- Split the dataset into training and testing sets.

2. Model Architecture

- Design a Convolutional Neural Network (CNN) architecture using PyTorch.
- Experiment with different architectures, including variations in the number of convolutional layers, pooling layers, and fully connected layers. Additionally, test the following pre-existing CNN architectures on the ASL Alphabet dataset:
  - **AlexNet**
  - **ZFNet**
  - **VGG**
  - **ResNet**
  - **GoogLeNet**
- Compare each model's performance with others regarding loss and accuracy metrics.

3. Training

- Train the CNN models using the training dataset.
- Experiment with different optimization algorithms (e.g., SGD, Adam) and learning rates.
- Monitor the training process, including loss and accuracy metrics.

4. Evaluation

- Evaluate the trained model using the testing dataset.
- Calculate the accuracy of the model on the testing dataset.
- Analyze the performance of the model and identify areas for improvement.

5. Hyperparameter Tuning

- Perform hyperparameter tuning to optimize the performance of the model.
- Experiment with different hyperparameters such as batch size, number of epochs, and learning rate.

6. Visualization

- Visualize the training process (e.g., loss curve, accuracy curve).
- Visualize sample predictions and compare them with the ground truth labels.

Note

Ensure proper code documentation and comments for better understanding. Clear and thorough documentation is not just a requirement for this assignment but a valuable skill in any professional setting. So, as you work on this assignment, practice communicating your ideas and processes clearly through your code. Refer to PyTorch documentation and tutorials for guidance on implementing CNN models. Experimentation is encouraged to improve model accuracy, but document all changes and observations thoroughly.

References

PyTorch Documentation
ASL Alphabet Dataset