

PyTorch Models Implementation

This folder contains implementations of three popular convolutional neural network (CNN) architectures, namely ResNet50, AlexNet, and VGGNet, applied to the CIFAR-10 and CIFAR-100 datasets. Each model implementation has two files corresponding to the two datasets, resulting in a total of six files for each model.

File Structure:

- **ResNet50-CIFAR100.ipynb**: Implementation of ResNet50 on CIFAR-100 dataset.
- **ResNet50-CIFAR10.ipynb**: Implementation of ResNet50 on CIFAR-10 dataset.
- **VGGNet19-CIFAR100.ipynb**: Implementation of VGGNet19 on CIFAR-100 dataset.
- **VGGNet19-CIFAR10.ipynb**: Implementation of VGGNet19 on CIFAR-10 dataset.
- **AlexNet-CIFAR10.ipynb**: Implementation of AlexNet on CIFAR-10 dataset.
- **AlexNet-CIFAR100.ipynb**: Implementation of AlexNet on CIFAR-100 dataset.
- **AccuracyTemplate.ipynb**: A template file for checking the accuracy of each model.

Implementation Details:

These implementations were created based on the template provided in [this discussion](#) on the PyTorch forum. The code was developed using PyTorch and was executed on Google Colab, leveraging TPU v2 and TP4 GPUs for faster execution.

Usage:

To use these implementations, follow these steps:

1. Install PyTorch if not already installed.
2. Open the desired model file (e.g., **ResNet50-CIFAR10.ipynb**) in a Jupyter Notebook environment.
3. Adjust the code as needed for your specific use case or dataset.
4. Execute the code in a suitable environment, preferably on Google Colab with TPU or GPU acceleration.

Accuracy Testing:

For testing the accuracy of each model, refer to the **AccuracyTemplate.ipynb** file. This file provides a template for evaluating the accuracy of the models on the respective datasets. Adjust the code in this template according to the specific model and dataset being tested.