

Demo: INT8-based Quantization-aware Training (QAT) via Loss-aware Compensation (LAC)

Latest Version: ver0.8.1-demo, Edited By: ZHOU Qihua, 2021.03.08, Mon.

This demo is based on ver0.8.1

1. Demo Structure

| Folder | Description |
|---------------------------|--|
| <code>simplecnnlac</code> | 3-layer CNN (1CONV+2FC) based on QAT&LAC |
| <code>alexnet_lac</code> | 8-layerCNN (5CONV+3FC) based on QAT&LAC |
| <code>vgg11_lac</code> | 11-layerCNN (8CONV+3FC) based on QAT&LAC |
| <code>quantizer</code> | INT8 quantization module |
| <code>common</code> | Neural network common modules |
| <code>dataset</code> | MNIST dataset and data loader |

2. Core Files

For example, if we use MNIST dataset to train `simplecnn_lac` model and the core file is `simplecnn_lac/train_convnet.py`.

| Python File | Description |
|---|---|
| <code>simplecnn_lac/train_simplecnn_LAC_mnsit.py</code> | Main entrance of the training procedure |
| <code>simplecnn_lac/simplecnn_LAC_mnist.py</code> | Build the 3-layer CNN |
| <code>quantizer/KMQuantizer.py</code> | Quatization functions |
| <code>common/trainer</code> | Training handler |
| <code>common/layer</code> | Layers of the neural network |

3. Prerequisites

The following Python packages are required:

- Python 3.x (3.6 is recommended)
- NumPy
- Matplotlib

4. Run

4.1 Run in command-line interface

Shift into `simplecnn_lac` folder and excute Python files:

```
$ cd simplecnn_lac
$ python train_simplecnn_LAC_mnsit.py
```

4.2 Run in Pycharm (recommended)

Directly open the root folder, choose `train_simplecnn_LAC_mnsit.py` , click the `run` button.

Please configure the Python Interpreter correctly