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
`simplecnn_lac`	3-layer CNN (1CONV+2FC) based on QAT&LAC
`alexnet_lac`	8-layerCNN (5CONV+3FC) based on QAT&LAC
`vgg11_lac`	11-layerCNN (8CONV+3FC) based on QAT&LAC
`quantizer`	INT8 quantization module
`common`	Neural network common modules
`dataset`	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
`simplecnn_lac/train_simplecnn_LAC_mnsit.py`	Main entrance of the training procedure
`simplecnn_lac/simplecnn_LAC_mnist.py`	Build the 3-layer CNN
`quantizer/KMQuantizer.py`	Quatization functions
`common/trainer`	Training handler
`common/layer`	Layers of the neural network

3. Prerequisites

The following Python packages are required:

- Python 3.x (3.6 is recommanded)
- 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 (recommanded)

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

> **Please configure the Python Interpreter correctly**

5. Relevant Publications

- "Octo: INT8 Training with Loss-aware Compensation and Backward Quantization for Tiny On-device Learning", In Proc. of USENIX ATC 2021.