TWTA-Net

Features:

- This supplemental material gives a reproduction function of ANN training, testing experiments for **TWTA-Net** (TW: ternary weight, TA: ternary activation).

File overview:

- `README.md` - this readme file.
- `requirements.txt` - installation file.
- `MNIST` - the workspace folder for `LeNet` on MNIST.
- `CIFAR10` - the workspace folder for `VGGNet-13` and `group_V2` on CIFAR10.
- `PDF` - pdf version of readme file

Requirements

Dependencies and Libraries:

- python 3.5 (https://www.python.org/ or https://www.anaconda.com/)
- tensorflow_gpu 1.2.1 (https://github.com/tensorflow)
- tensorlayer 1.8.5 (https://github.com/tensorlayer)
- CPU: Intel(R) Xeon(R) CPU E5-2620 v4 @ 2.10GHz
- GPU: Tesla V100

Installation:

To install requirements,

```
pip install -r requirements.txt
```

Datasets:

- MNIST: dataset, preprocessing
- CIFAR10: dataset, preprocessing

ANN Training

Before running:

- Please installing the required package Tensorflow and Tensorlayer (using our modified version is ok)
- Please note your default dataset folder will be `workspace/data`, such as `TWTA/CIFAR10/VGGNet-13/data`
- Select the index of GPU in the training scripts (0 by default, if you have)

Run the code:

for example (training, VGGNet-13, CIFAR10):

```
$ cd CIFAR10
$ python TWTA_3_3_CIFAR10_VGG13.py --resume False --learning_rate 0.001 --mode
'training'
```

ANN Inference

Run the code:

for example (inference, *k=0*, CNN1, CIFAR10):

```
$ python TWTA_3_3_CIFAR10_VGG13.py --resume True  --mode 'inference'
```

Others

- We do not consider the ternary quantization of input encoding layer and the last classification layer in ANNs.
- More for BWBA-Net, TWBA-Net and ST-conversion on MNIST and CIFAR-10, please refer to our previous works[1][2]. 参见之前的论文实验
- In future, **可训练(at present)**, 固定值(经验值), 运行中决定, BN融合, 池化层, 卷积核大小, to be completed.

Results

Our proposed methods achieve the following performances on MNIST and CIFAR10 dataset:

MNIST:

Quantization Level	Network Size	Epochs	Accuracy	Notes
FP32-5*5	32C5-2P2-64C5-2P2-512	150	--	全精度32位 FP_5_5_MNIST.py
TWTA-2*2	64C2-2P2-64C2-2P2-64C2-2P2-64C2-512	150	very low	TWTA_2_2_MNIST.py
TWTA-4*4	32C4-2P2-64C4-2P2-64C4-512	150	99.38%	TWTA_4_4_MNIST.py
TWTA-5*5	32C5-2P2-64C5-2P2-512	150	99.32%	TWTA_5_5_MNIST.py

CIFAR10:

Quantization Level	Network Size	Epochs	Accuracy	Notes

Quantization Level	Network Size	Epochs	Accuracy	Notes
FP32-3*3-VGGNet-13	64C3*3-2P2-128C3*2-2P2-256C3*2-2P2-512C3*2	200	--	全精度32位--
TWTA-3*3-VGGNet-13	64C3*3-2P2-128C3*2-2P2-256C3*2-2P2-512C3*2	200	--	`TWTA_3_3_CIFAR10_VGG13.py`
TWTA-4*4-VGGNet-13	64C3-64C4*2-2P2-128C4*2-2P2-256C4*2-4P2-512N1	200	90.1%	`TWTA_4_4_CIFAR10_VGG13.py`
TWTA-3*3-Group_V2	32C3-64C3-4P2/4-512C3/16-2P2/8-2048C3/32-1024N1/8-2P2/16-1024C3/16-512N1	200	88.1%	`TWTA_3_3_CIFAR10_Group_V2.py`
TWTA-4*4-Group_V2	32C3-64C4-4P2/4-512C4/16-2P2/8-2048C4/32-1024N1/8-4P2/16-1024C2/16-512N1	200	--	`TWTA_4_4_CIFAR10_Group_V2.py`

More question:

- There might be a little difference of results for multiple training repetitions, because of the randomization.
- Please feel free to reach out here or email: 2829008362@qq.com, if you have any questions or difficulties. I'm happy to help guide you.