

IMAE for Noise-Robust Learning: Mean Absolute Error Does Not Treat Examples Equally and Gradient Magnitude's Variance Matters

Downloading Link

https://www.dropbox.com/sh/vb48gq7rugeugtm/AAACFbUSqAc8XNmVr_PniUena?dl=0

(https://www.dropbox.com/sh/vb48gq7rugeugtm/AAACFbUSqAc8XNmVr_PniUena?dl=0)

Command for Visualising the Repository/Directory Tree Structure

```
cd directory_name
tree
```

Dependencies

The core functions are implemented in the [caffe](https://github.com/BVLC/caffe) (<https://github.com/BVLC/caffe>) framework. We use matlab interfaces [matcaffe](#) for data preparation.

- [CaffeMex_v2](https://github.com/sciencefans/CaffeMex_v2/tree/9bab8d2aaa2dbc448fd7123c98d225c680b066e4) (https://github.com/sciencefans/CaffeMex_v2/tree/9bab8d2aaa2dbc448fd7123c98d225c680b066e4)
- [MATLAB 2017b](https://uk.mathworks.com/products/new_products/release2017b.html) (https://uk.mathworks.com/products/new_products/release2017b.html)

Setup

- Install dependencies on Ubuntu 16.04 (<http://caffe.berkeleyvision.org/install Apt.html>)

```
sudo apt-get install libprotobuf-dev libleveldb-dev libsnappy-dev libopencv-dev libhdf5-serial-dev protobuf-compiler
sudo apt-get install --no-install-recommends libboost-all-dev
sudo apt-get install libopenblas-dev
sudo apt-get install python-dev
sudo apt-get install libgflags-dev libgoogle-glog-dev liblmdb-dev
```

- Install [MATLAB 2017b](https://uk.mathworks.com/products/new_products/release2017b.html) (https://uk.mathworks.com/products/new_products/release2017b.html)

Download and Run the install binary file

```
./install
```

- Compile Caffe and matlab interface

Note you may need to change some paths in Makefile.config according your system environment and MATLAB path

```
cd ../CaffeMex_UnifiedWeight_V01
make -j8 && make matcaffe

cd ../CaffeMex_MAE_V00
make -j8 && make matcaffe
```

Usage

Examples for reproducing our results on [CIFAR-100](https://www.cs.toronto.edu/~kriz/cifar.html) (<https://www.cs.toronto.edu/~kriz/cifar.html>) are given.

- Data preparation for CIFAR-100

- Prepare testing data:

```
cd CIFAR100_Data_Toolkit
matlab -nodisplay -nosplash -nodesktop -r "run('test_data_preparation.m');exit;" | tail -n +11
```

- Prepare training data (symmetric noise rate: 0.0, 0.2, 0.4, 0.6):

```
matlab -nodisplay -nosplash -nodesktop -r "run('train_data_preparationV2_noise_0_0.m');exit;" | tail -n +11
matlab -nodisplay -nosplash -nodesktop -r "run('train_data_preparationV2_noise_0_2.m');exit;" | tail -n +11
matlab -nodisplay -nosplash -nodesktop -r "run('train_data_preparationV2_noise_0_4.m');exit;" | tail -n +11
matlab -nodisplay -nosplash -nodesktop -r "run('train_data_preparationV2_noise_0_6.m');exit;" | tail -n +11
```

- Copy data

```
cd ..
echo CIFAR100_ResNet44_V03_lambda1/pre_pro_process | xargs -n 1 cp CIFAR100_Data_Toolkit/TestImageDataCell.mat
echo CIFAR100_ResNet44_V03_lambda1/pre_pro_process | xargs -n 1 cp CIFAR100_Data_Toolkit/TrainImageDataCell0.0.mat
echo CIFAR100_ResNet44_V03_lambda1/pre_pro_process | xargs -n 1 cp CIFAR100_Data_Toolkit/TrainImageDataCell0.2.mat
echo CIFAR100_ResNet44_V03_lambda1/pre_pro_process | xargs -n 1 cp CIFAR100_Data_Toolkit/TrainImageDataCell0.4.mat
echo CIFAR100_ResNet44_V03_lambda1/pre_pro_process | xargs -n 1 cp CIFAR100_Data_Toolkit/TrainImageDataCell0.6.mat
```

- Train & Test

Run the training and testing scripts in the training folder of a specific setting defined by its corresponding prototxt folder.

For example,

```
cd CIFAR100_ResNet44_V03_lambda1/train_Res44_USW_Beta02_lambda1_0.6
matlab -nodisplay -nosplash -nodesktop -r "run('train.m');exit;" | tail -n +11
matlab -nodisplay -nosplash -nodesktop -r "run('test.m');exit;" | tail -n +11
```

Our trained results

- Our trained results are stored in corresponding folders. For example, in Folder **CIFAR100_ResNet44_V03_lambda1/train_Res44_USW_Beta02_lambda1_0.6**, there are:
 - accuracy.txt
 - accuracy_curve.png
- Without changing the random seed (123), you are supposed to obtain exactly the same results.

Acknowledgements

Our implementation benefits from:

- Caffe library: <https://caffe.berkeleyvision.org/>
- CaffeMex_v2 library: https://github.com/sciencefans/CaffeMex_v2/tree/9bab8d2aaa2dbc448fd7123c98d225c680b066e4

