Householder-Absolute Neural Layers For High Variability And Deep Trainability

This repository is the official implementation of [Householder-Absolute Neural Layers For High Variability And Deep Trainability].

Requirements

The codes have been developed in Python3.6, they require the following packages to run.

```
--pytorch >=1.8
--numpy
--pandas
--matplotlib
```

Demos on Checkerboard

We provide some GIFs about training HanNet and FCNet in

```
demo/
```

you can watch these animations, where HanNet and FCNet achieve the following testing accuracy:

Model name	Accuracy
HanNet	99.5%
FCNet	85.6%

Experiments on Checkerboard

Training on Checkerboard

To train HanNet in the paper on Checkerboard dataset, run this command:

```
cd checkerboard_experiments/
python main.py --model hannet --activation ABS --initial orth
```

or to train FCNet, by running the following command

```
python main.py --model fcnet --activation ReLU --initial kaiming
```

We also provide ResNet (FCNet with batch-normalization and skip-connection).
To train a 20-hidden-layer and 100-width ResNet, run this command

```
python main.py --model resnet --activation ReLU
```

Evaluation on Checkerboard

There are pre-trained models in

```
checkerboard_experiments/model/
```

To evaluate HanNet, FCNet and ResNet on Checkerboard, run:

```
python test.py --model hannet
python test.py --model fcnet
```

```
python test.py --model resnet
```

Experiments on Regression Datasets

To train HanNet in the paper on Elevators dataset, run this command:

```
cd regession_experiments/
python main.py --model hannet --prob elevators --rho 0.8
```

or train FCNet on Cal-housing dataset as follows

```
python main.py --model fcnet --prob calhousing --rho 0.2
```

where `rho` is training percentage.

Experiments on CIFAR-10

Download features and pre-trained models from https://drive.google.com/drive/folders/1F4UsbUM81iVvO9eX5bWoNZzR3hxfuwXy?usp=sharing

File name	Size
feature	469MB
model	347MB

Then put the downloaded directories into the directory `cifar10_experiments/`

Training on CIFAR-10

```
cd cifar10_experiments/
python main.py --device gpu --gpu 0
```

Testing on CIFAR-10

There are pre-trained models in

```
cifar10_experiments/model
```

By running the following command:

```
python test.py --device gpu --gpu 0
```

you should obtain the results:

Model name	Test err
LaNet	0.97%
FCNet	0.89%