Practice #4

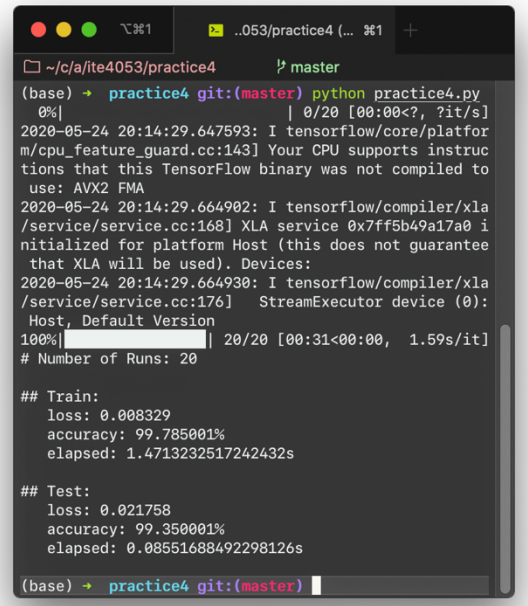
2016025305 Jihun Kim

jihunkim@hanyang.ac.kr

May 26, 2020.

Development Environment

OS	MacOS 10.15.4 (19E287)
HW	MacBook Pro (13-inch, 2017) 3.5 GHz Dual-Core Intel Core i7 16 GB 2133 MHz LPDDR3
Language	Python 3.6.8 |Anaconda, Inc.|
Libraries	NumPy 1.18.4 TensorFlow 2.2.0 Keras 2.3.0-tf tqdm 4.36.1



Run

```
$ cd /path/to/repo/practice4
$ python practice4.py
```

Experimental Setup

Used Adam with `learning_rate=0.5` as optimizer for TensorFlow-implemented version.

Results

		NumPy (Practice 3, Model 3)	TensorFlow (Local, CPU)	TensorFlow (Colab, CPU)	TensorFlow (Colab, GPU)
Accuracy	Train	99.24%	99.78%	99.76%	99.78%
	Test	99.00%	99.35%	99.30%	99.25%
Loss	Train	0.02	0.008329	0.008506	0.008517
	Test	0.02	0.021758	0.021491	0.023119
Elapsed Time	Train	124.59ms	1.47s	1.88s	3.05s
	Test	0.08ms	85ms	111ms	92ms

Results above are average value of 20 runs per model. Models implemented with TensorFlow performed slightly better than NumPy-implemented models but showed much poor speed. It seems that this difference is mainly because of different optimizer (Adam versus GD). And there is somewhat weird thing that training TensorFlow-implemented model with Colab GPU were much slower than others. As I think, the core reason for this is because of the CPU-GPU data transfer time, which could be larger than the time gain from usage of GPU (instead of CPU).