|  |
| --- |
| **Technical Note: Differences between Tensorflow and Pytorch** |
| **Junyeob Kim** |

1. **Introduction**

Two tools, Tensorflow and Pytorch, are examples of deep learning frameworks for python. Both are commonly used for many purposes, including research and prototypes. These tools enable easy and fast development of deep learning networks. In this technical note, differences between two popular tools will be discussed, after brief description of each framework.

1. **Description of two frameworks**

Tensorflow is an open source, deep learning framework which is created and released by

Google in 2015. Its new version, Tensorflow 2.0, has been released on 30th of September 2019.

On the other hand, Pytorch is created and released by Facebook in 2017.

1. **Comparison**

One of the main differences between Tensorflow and Pytorch is how they define their computational graphs. Tensorflow uses static computational graph. This type of computation first statically defines the graph and executes the required computation. On the other hand, Pytorch uses dynamic computational graph. The graphs define the structure and executes it as the program runs. Therefore, with Pytorch, it is possible to change the structure of the Deep Learning Model during the process.

|  |  |
| --- | --- |
| A screenshot of a cell phone  Description automatically generated | A screenshot of a cell phone  Description automatically generated |
| Figure 1 – Simple multiplication of two float values with (reft) Tensorflow tensor and (right) Pytorch tensor. | |

Also, the way each framework deals with data types is different.

As shown in fig.1, the data types of two multiplication results are float32. However, tensorflow tensor has precision up to 7 decimal places, while pytorch tensor only has 4 decimal places. With huge set of computation, the result will vary significantly, if float values are used as inputs.

1. **Change in models**

Batch normalisation is added to all convolution layers on all models (VGG16, GoogleNet, Alexnet) that does not have it as default (in original paper) -> To improve results with models that does not learn much with current hyperparameter settings.

As result, the accuracy of VGG16 has improved by approximately 60%.

AlexNet - performance reduced by 0 ~ 10%

GoogleNet – performance before batchnorm - 37% ~ 74%, after – 77%

1. **References**

[1] M. Abadi et al., “TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems**”,** [arXiv:1603.04467](https://arxiv.org/abs/1603.04467).

[2] A. Paszke et al., “PyTorch: An Imperative Style, High-Performance Deep Learning Library”, [arXiv:1912.01703](https://arxiv.org/abs/1912.01703)**.**