Technical Review

Diana Arita

Introduction to TensorFlow

TensorFlow is a powerful open source platform for machine learning. It contains tools, libraries, and community resources that lets researchers create state-of-the-art artificial intelligence technologies and developers build machine learning applications. [1] Originally developed to conduct machine learning and deep neural networks research, TensorFlow has the ability to be applied to a wide range of applications. In this technical review, TensorFlow's frameworks in Natural Language Processing (NLP) will be compared against PyTorch and Keras.

Non-Competitive Facts

Presented below are differences between the three technologies that briefly describe them. They are not differences that are meant to be used as a comparison. [2]

TensorFlow

- Created by Google
- version 1.0 in February, 2017

PyTorch

- Created by Facebook
- Version 1.0 in October, 2018
- Based on Torch, another deep learning framework based on Lua

Keras

- High level API to simplify the complexity of deep learning frameworks
- Runs on top of other deep learning APIs — TensorFlow, Theano and CNTK
- It is not a library on its own

Competitive differences

Recurrent Neural Networks (RNNs) are a popular architecture for deep learning in NLP. TensorFlow contains a tf.nn.rnn_cell module that has three classes for RNN. Cell level classes are used to define a single cell of RNN. MultiRNNCell classes are used to stach various cells to create deep RNNs. Lastly, DropoutWrapper class is used for dropout regularization. PyTorch has two classes for creating recurrent networks. Multi-layer classes are capable of representing deep bidirectional recurrent neural networks. Cell-level classes can represent only a single cell that can handle one timestep of the input data. Keras has layers that can be used for RNN implementations. SimpleRNN is fully connected RNN where the output is fed back to the input. Gated Recurrent Unit Layer and Long Short Term Memory layer are also Keras RNN layers. All three frameworks have the ability to create RNN architectures. Their differences are in their interfaces. Keras used on top of TensorFlow allows for easy usage. While PyTorch has a similar level of flexibility and a cleaner interface. [2]

TensorFlow doesn’t have the easiest to read API. It uses static graphs which means that you need to define a computation graph statically before a model is run. Communication with the outer world is only possible with tf.Session and tf.Placeholder objects which get substituted by external data at runtime. PyTorch is simpler to use and is more tightly integrated with Python. It’s more imperative and dynamic than TensorFlow.

Conclusion

The most mature and having the largest community, TensorFlow is currently the most popular of the three frameworks. However, PyTorch has a lot of potential and should not be overlooked.

References

[1] “Tensorflow: An Open Source Machine Learning Framework for Everyone.” *GitHub*, https://github.com/tensorflow/tensorflow.

[2] Montantes, James. “TensorFlow vs PyTorch vs Keras for NLP — Exxact.” Medium, Towards Data Science, 6 Aug. 2019, https://towardsdatascience.com/tensorflow-vs-pytorch-vs-keras-for-nlp-exxact-8e51dd13c3f5.