

Comparison of Python NLP Frameworks: PyTorch and Keras

Introduction

Deep learning is one of the most significant research areas in data science. The practical applications using deep learning are quite expansive and growing day by day. Appropriately, there are numerous tools and frameworks developed for deep learning. This technology review is focused on two specific NLP (Natural Language Processing) frameworks, PyTorch ^[1] and Keras ^[2]. These are considered to be the most commonly used NLP frameworks.

About PyTorch and Keras

As mentioned earlier, PyTorch and Keras have been the most popular NLP frameworks because of their ease of use and availability of online learning resources. PyTorch is relatively new, developed in 2016 by Facebook (now Meta) and is an open-source framework. Because of its open-source implementation, PyTorch is more commonly used in research settings. Number of popular deep learning applications such as Tesla Autopilot ^[3], Uber's Pyro ^[4] are using PyTorch. Keras was developed by Alphabet in 2015. Keras is considered to be more common in deployment and production settings.

Syntactic Variations

Keras is a higher-level framework which provides easier and concise implementation options. The most simplest model is using the sequential model (`tf.keras.Sequential`). This provides a linear stack of layers and is most commonly used for basic NLP architectures. For more complex architectures or to create more flexible models, Keras offers functional API. This functional API offers the ability to handle models with non-linear topology, shared layers and multiple inputs or outputs. Keras is also considered to be good for beginners for learning and understanding NLP concepts.

PyTorch on the other hand offers implementation in a lower-level environment that is good for flexibility and experimentation. This provides the ability to write custom layers and develop complex architectures. This increased flexibility and support of complex architectures comes at the cost of number of lines of code required to implement this framework.

In general, for someone with Python skills, Keras is considered to be simple, concise and readable whereas PyTorch is complex and requires some time to get familiar with the framework.

Choice of Framework based on Performance in NLP

Based on online resources ^[5] and example performance comparison available on GitHub ^[6], most of the out-of-the-box implementations seem to highlight that PyTorch has better performance than Keras. The general consensus is that if the dataset size is not large and there is a need for rapid experimentation with easy model training, then Keras is preferred framework. On the other hand, if the dataset size is large, involved high-performance models requiring further optimization and training speed reduction, PyTorch is the preferred framework.

Ease of Debugging

Keras in order to offer simple implementation, wraps computational chunks in abstraction and therefore makes it harder for debugging the exact line of code that causes the error. The advantage of this approach however is that because of dealing with fewer lines of code, the chances of encountering errors is lesser.

PyTorch because of the flexibility, offers specific lines of code for that provides the ability to access all objects in the framework. The user is able to introduce interim statements which enable easy debugging methods regardless of the model complexity. However, both Keras and PyTorch, because of their popularity, provide vast repository of working code, tutorials and discussion groups.

Conclusion

Keras is the preferred framework when implementing NLP frameworks for the first time because of its simplicity and ease of use. PyTorch is preferred when there are specific customization and performance requirements. On the performance front, both PyTorch and Keras pipelines can always be better optimized. However, when using out-of-the-box parameters, PyTorch seems to offer the best performance. The right framework depends on the use case and needs. The user needs to decide the right NLP framework based on which feature or performance metric matters the most.

References

- [1] <https://pytorch.org/>
- [2] <https://keras.io/>
- [3] How Tesla Uses PyTorch, <https://analyticsindiamag.com/tesla-pytorch-self-driving-computer-vision-karpathy-elon-musk-ai/>
- [4] Uber Has Been Quietly Assembling One of the Most Impressive Open Source Deep Learning Stacks in the Market, <https://www.kdnuggets.com/2020/01/uber-quietly-assembling-impressive-open-source-deep-learning.html>
- [5] Deep Learning Frameworks Speed Comparison, <https://wrosinski.github.io/deep-learning-frameworks/>
- [6] Deep Learning Framework Examples, <https://github.com/ilkarman/DeepLearningFrameworks/>