CS410 Technology Review
TensorFlow 2.0 and TensorFlow Text
Paul Zuradzki

Technology Review of TensorFlow 2.0 and TensorFlow Text

The following review will evaluate TensorFlow 2.0 and the relatively new TensorFlow Text library. Special attention will be given to TensorFlow's natural language processing features, the TensorFlow Text API, and applications in text retrieval and analysis.

What is TensorFlow

The two most popular deep learning libraries are TensorFlow by Google and PyTorch by Facebook. TensorFlow is older and more prevalent. Some find the PyTorch API to be more friendly for beginners. TensorFlow is strong at general numerical computation but it's especially suited for large-scale machine learning and artificial intelligence. TensorFlow was created by the Google Brain team and it is behind popular services including Google Search, Google Photos, and Google Cloud Speech. TensorFlow was released as open source in November 2015 and has grown to become the most popular Deep Learning library as measured by GitHub stars, paper citations, and industry usage. TensorFlow can be used for many machine learning tasks like image classification, natural language processing, recommender systems, and time series forecasting.

TensorFlow Core Features

TensorFlow has some key features that make it flexible, scalable, and production-ready. Other libraries tend to hit only a subset of these attributes. TensorFlow has NumPy-like functionality with GPU support. It supports distributed computing on multiple devices and servers. TensorFlows has a special just-in-time compiler that optimizes computations for speed and memory use. The computation graphs are portable in a way that allows one to train a model in one environment and run the model in another; e.g., one can train a model using Python on Linux and run the model using Java on Android. TensorFlow also implements auto differentiation (or autodiff). Autodiff is useful for implementing machine learning algorithms such as backpropagation for training neural networks. Autodiff helps take partial derivatives to implement gradient descent along with other optimization algorithms.

TensorFlow Additional Features and APIs

In addition to key features mentioned, TensorFlow includes the keras API, which provides a Python interface for artificial neural networks. TensorFlow also includes operations for data loading and preprocessing, image processing, signal processing (tf.data, tf.io, tf.image, tf.signal), and math and linear algebra modules (tf.math, tf.linalg, tf.rand, tf.bitwise).

TensorFlow Text

TensorFlow Text provides a collection of text processing classes and operations that integrate with TensorFlow 2.0. TensorFlow has a breadth of operations that help to build models from images and videos. However, many models begin with text. Language models also require preprocessing operations that are not in core TensorFlow. Preprocessing logic must be consistently coordinated with the TensorFlow computation graph at training and inference times to avoid problems of skewed data. TensorFlow Text helps manage this complexity and ensure that text pre-processing is consistent across model stages.

The most common operation in text models is tokenization: the process of breaking up a string into tokens like we have seen in MetaPy. TensorFlow Text offers three tokenizers. The whitespace tokenizer splits strings on whitespace characters; e.g., space, tab, new line. The Unicode Script tokenizer splits UTF-8 strings based on Unicode script boundaries. Lastly, the Wordpiece tokenizer splits tokens into subwords (prefixes and suffixes). Wordpiece is commonly used in BERT NLP pre-training models.

Beyond tokenization, TensorFlow Text also supports normalization, n-grams, and sequence constraints for labeling. Normalization makes sure that the same words from any source are recognized to be identical; e.g., by matching cases or transforming strings to common unicode representations. TensorFlow Text can make N-grams, which are sequential words given a sliding window size of n. The wordshape() utility method helps identify common features of sentences such as whether a token has title case, whether a token is upper case, whether a token has a punctuation or symbol, or if a token is numeric. This is especially useful in language understanding models; e.g., say, if one needs to recognize a break between sentences for question-answering.

Concluding Remarks

We can see TensorFlow is a powerful tool for general numeric processing. Integrated with the TensorFlow Text library, we also have a specialized software toolkit for processing text data. Users can use TensorFlow and TensorFlow Text together to create NLP models and text applications including chat bots, search, recommender systems, and more.

References and Resources

- TensorFlow
  - https://www.tensorflow.org
  - https://en.wikipedia.org/wiki/TensorFlow
  - https://en.wikipedia.org/wiki/Keras
  - *Hands-On Machine Learning with Scikit-Learn, Keras & TensorFlow* by Aurélien Géron. Book URL: https://www.amazon.com/dp/1492032646
- TensorFlow Text
  - https://www.tensorflow.org/text
  - https://blog.tensorflow.org/2019/06/introducing-tftext.html
  - https://github.com/tensorflow/text
  - https://www.tensorflow.org/text/guide/subwords_tokenizer#optional_the_algorithm
  - https://github.com/google-research/bert