Week 9: Training RNNs to Identify Disaster-Related Tweets

Background and Motivation

This analysis was performed on a dataset of tweets made available via the "Natural Language Processing with Disaster Tweets Predict which Tweets are about real disasters and which ones are not" Kaggle competition (Kaggle 2022). The dataset contains 10,878 tweets, with 7,613 "training" tweets and 3,263 "test" tweets. Approximately 40% of the tweets describe natural disasters, such as earthquakes or hurricanes. The intent of this effort is to use natural language processing to identify which of the tweets relate to natural disasters and which do not. This semantic differentiation of intent is a critical building block to creating a machine-driven natural language, applicable to all sorts of voice-command systems, as well as to the study of automatic text parsing, e.g., for "robo-lawyers".

Approaches to Pre-Processing the Data

The dataset provided was also enriched with a location of tweet and a keyword, extracted from the tweet. Both of these fields could be null, with "location" being null for over half of observations. In modeling, both of these enrichments were ignored.

Raw tweet content is difficult for machine learning techniques to engage. Each tweet was cleansed for processing by removing URLs, removing emoji, hashtag symbols, and other special characters, cleaning up misspellings, expanding contractions, and clearing extraneous spaces. The intent of these steps is to create a corpus that generally aligns with existing NLP models. As a final step, each tweet was tokenized.

Modeling Approach

The focus this week was on developing recurrent neural networks, testing the impact of hyperparameter tuning on model output. I focused on testing the impact of learning rate and optimizer type, for a model built on top of an existing NLP model.

Other than the learning rate and optimizer differences, each of the three models were built identically, with a RoBERTa model transformer, two pooled output layers with dropout, and a dense layer predicting the dependent variable. Each model was trained for 50 epochs with early stopping of patience = 10. In practice, all three models stopped training within 15 epochs.

Results:

	Optimizer	**Learning Rate**	**Loss (Val)**	**F1 (test)**
Model 1	Adam	6.00E-05	0.4039	0.81949
Model 2	Adam	6e-5, LR reduce factor = 0.3	0.4297	0.81949
Model 3	SGD	6.00E-05	0.4237	0.81703

Implications

- The differences in learning rate and optimizer had no meaningful impact on the models' accuracy on an unseen test set. While there was some variability in performance on the validation set, it was surprisingly small
- Transfer learning provides a substantial lift in performance and speed. The RoBERTa transformer was trained on 160gb of data, and contains over 100mm parameters. This far exceeds the capacity of a weeklong analytics effort. Furthermore, the models trained quickly, with sub-minute epochs
- Hardware matters. Models were executed in Google Colab using GPUs. Not shown above, but Model 1 was also tested using CPUs and TPUs, with each epoch taking ~45 minutes, a substantial increase

References

"Natural Language Processing with Disaster Tweets Predict which Tweets are about real disasters and which ones are not." Kaggle. Accessed February 27, 2022. https://www.kaggle.com/c/nlp-getting-started/overview.