Adam Kiehl

5/11/2023

Homework 4 (Ibotta)

The goal of this modeling exercise was to predict the class of a product given a product name and brand. This dataset comprises 9,999 observations (8,000 for training and 1,999 for testing) that each belong to one of seven classes (dairy, produce, etc.). The only preprocessing performed on the text was to combine the brand and product names fields into a single textual field. A variety of neural networks were used for this assignment and all models were evaluated using validation F1 score.

Several methods were employed to vectorize the textual data and all model frameworks were tried in conjunction with each vectorization method. The first method encoded each token to an integer according to a vocabulary index; each token sequence was padded to the maximum sequence length (50). The second method one-hot encoded each token sequence with respect to the data's full vocabulary (4,880). The third method performed TF-IDF encoding on each token sequence, again with respect to the data's full vocabulary (4,880). Versions of the one-hot and TF-IDF encodings were also generated only using the top 1,000 vocabulary tokens.

Beginning with simple models, each training data set was used on several MLP models with ranging numbers of neurons and hidden layers. The integer vector models performed poorly across the board, but both the one-hot and TF-IDF models were very strong. A model with four hidden layers (512, 256, 128, and 64 neurons) achieved a validation F1 score of 0.915 with the one-hot encoded data and 0.911 with the TF-IDF encoded data. Another intriguing model with six hidden layers, penalized regularization, and dropout achieved a validation F1 score of 0.905 with the TF-IDF encoded data. The 1,000-feature training sets performed robustly but not as strongly as their complete counterparts.

The next class of models tested included an embeddings layer followed by a flattening and a small MLP. A range of embedding lengths from 8 to 128 were explored. The integer vector models performed reasonably well but not to a competitive degree. The one-hot and TF-IDF models at times performed impressively and other times performed poorly. Most notably, a model with a 64-length embedding and two hidden layers (128 and 64 neurons) achieved a validation F1 score of 0.9122 with the one-hot encoded data and 0.917 with TF-IDF.

Various architectures that included RNN layers in conjunction with embedding were implemented but all performed distinctly poorly and were nowhere near as effective as the simpler models tried. Only the integer vector RNN models were able to score above 0.5.

Five models were chosen to make final submissions: the simple MLP models with one-hot and TF-IDF encoding, the deeper regularized MLP with TF-IDF encoding, and the embeddings models with one-hot and TF-IDF encoding. The first of these (four hidden layers and one-hot encoding) performed the best on the Kaggle test set with a test F1 score of 0.908, compared to 0.881 and 0.865 for the other MLP models. The embeddings model with one-hot encoding achieved an impressive test F1 score 0.893 but the version with TF-IDF encoding performed strangely poorly with a score of only 0.240. For this case, it seems that the simplest version of text vectorization combined with the simplest version of deep learning is the best approach. For future research, I would continue to explore RNN structures and try to better understand why those kinds of models did not perform well in this scenario.