UDACITY

‹ Return to "Deep Learning" in the classroom

Deploying a Sentiment Analysis Model

REVIEW

CODE REVIEW

HISTORY

Meets Specifications

Congratulations on completing the project🖐🌌 This is a very well done project! Your code has been written very well in a modular manner. Your concepts and usage of AWS Sagemaker and sentiment analysis are quite clear and have been well implemented in the notebook.

I really hope you enjoyed studying Deep Learning, the hottest topic in AI right now, here with Udacity 😄

Until next time! Have an amazing time working with neural nets.

Files Submitted

The submission includes all required files, including notebook, python scripts and html files.

Preparing and Processing Data

Answer describes what the pre-processing method does to a review.

Great job, you've identified key details in the pre-processing method. Here are all the 5 steps:

- Get text only out of the input review by removing html format (if any).
- Lower the text and then replace the pattern specified with space, that is, replace all characters that are not alphabets nor digits with space.
- Split each word of the text into a list (separated by a space).
- Remove all stopwords in the list.
- Stemming each word in the list. (convert entitled/entitling -> entitl, builds/building -> build, ...)

The `build_dict` method is implemented and constructs a valid word dictionary.

Notebook displays the five most frequently appearing words.

Answer describes how the processing methods are applied to the training and test data sets and what, if any, issues there may be.

Also, there is no issue of any data leakage doing this because: `preprocess_data` is applied per record and so there is no issue there, and the reason that `convert_and_pad_data` doesn't cause an issue is that `word_dict` is constructed using only the training data. So, this just to makes sure that processing data does not accidentally introduce leakage.

Build and Train the PyTorch Model

The train method is implemented and can be used to train the PyTorch model.

The RNN is trained using SageMaker's supported PyTorch functionality.

Deploy the Model for Testing

The trained PyTorch model is successfully deployed.

Use the Model for Testing

Answer describes the differences between the RNN model and the XGBoost model and how they perform on the IMDB data.

The test review has been processed correctly and stored in the `test_data` variable.

The `predict_fn()` method in `serve/predict.py` has been implemented.

Deploying the Web App

The model is deployed and the Lambda / API Gateway integration is complete so that the web app works (make sure to include your modified `index.html`).

Answer gives a sample review and the resulting predicted sentiment.

DOWNLOAD PROJECT

RETURN TO PATH

Rate this review