UDACITY

Return to Classroom

Deploying a Sentiment Analysis Model

REVIEW

CODE REVIEW

HISTORY

Meets Specifications

I think you've done a good job of implementing the sentiment analysis model. Your concepts and usage of AWS Sagemaker and sentiment analysis are quite clear and have been well implemented in the notebook.

Further Resource: Keras has a neat API for visualizing the architecture, which is very helpful while debugging your network. PyTorch-summary is a similar project(https://github.com/sksq96/pytorch-summary/) in PyTorch.

Further, AWS Sagemaker recently got a new update(https://aws.amazon.com/blogs/machine-learning/amazon-sagemaker-now-comes-with-new-capabilities-for-accelerating-machine-learning-experimentation/). It lets you find and evaluate the most relevant model training runs from the hundreds and thousands of your Amazon SageMaker model training jobs.

And Congratulations 🎉 once again, for successfully completing the project.

Files Submitted

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

Make sure your submission contains:

- **The `SageMaker Project.ipynb` file with fully functional code, all code cells executed and**

- The `SageMaker Project.ipynb` file with fully functional code, all code cells executed and displaying output, and all questions answered.
- An HTML or PDF export of the project notebook with the name `report.html` or `report.pdf`.
- The `train` folder with all provided files and the completed `train.py`.
- The `serve` folder with all provided files and the completed `predict.py`.
- The `website` folder with the edited `index.html` file.

Preparing and Processing Data

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

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.

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.

Make sure your answer includes:

- The comparison between the two models
- Which model is better for sentiment analysis

The test review has been processed correctly and stored in the `test_data` variable. The test_data should contain two variables: review_len and review[500].

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

- The predict script should include both the data processing and the prediction.
- The processing should produce two variables: data_X and data_len.

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`).

The answer includes a screenshot showing a sample review and the prediction.

DOWNLOAD PROJECT

RETURN TO PATH

Rate this review

START