UDACITY

‹ Return to Classroom

Deploying a Sentiment Analysis Model

REVIEW

CODE REVIEW

HISTORY

Meets Specifications

Dear Student,

well done ⭐ you have completed the assignment with flying colors.
I appreciate the effort you have made to training the model on AWS sagemaker and create the endpoint to deploy the model. As you have faced some issues to use sagemaker v1. we are currently working on shifting the assignment from V1 to V2. For the pandas issue, I think on that server non-compatible pandas also available already. As your training is done on the AWS server. if you have not constantly checked the notebook then after some time you may face connection time out. So because of that, your notebook didn't get any update further. if the notebook gets failed you can follow the link which explain that you can deploy endpoint using the trained model on the S3 bucket.

congratulation on completing the deep learning nano degree. Best of luck for your future endeavors and happy learning 📚

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 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.

Good 👏 all required files are available in the submission.
✅ noteook file
✅ train.py
✅ predict.py
✅ index.html

Preparing and Processing Data

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

Good 👏
you can also explain more about the following topics with an example:

- Remove HTML tags
- Convert to lower case
- Split string into words
- Remove stopwords
- stemming
  For reference, You can find the brief discussion for preprocessing of sentiment analysis here

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

Good 👏
There are some other techniques that can help you to create a dictionary for specific use cases.
link 1
link 2

Notebook displays the five most frequently appearing words.

Great work 👍
you are able to get the top five frequent words as follows:
`['movi', 'film', 'one', 'like', 'time']`

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

Good 👏 you have explained that preprocess should be applied to both datasets. you have also explained that words from the test data are not present in the dictionary but those are replaced by the INFREQ symbol. in the answer you can also add, the applied processes are correct because `word_dict` is generated by training dataset only so there won't be a leaking issue.
for more information about data leakage click here

Build and Train the PyTorch Model

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

Good network design 👏

```
self.embedding = nn.Embedding(vocab_size, embedding_dim, padding_idx=0)
self.lstm = nn.LSTM(embedding_dim, hidden_dim)
self.dense = nn.Linear(in_features=hidden_dim, out_features=1)
self.sig = nn.Sigmoid()
```

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

perfectly done 👍
I can see the logs which show that the model was trained properly.

```
2020-12-08 17:40:09,390 sagemaker-containers INFO Reporting training SUCCESS
```

Deploy the Model for Testing

The trained PyTorch model is successfully deployed.

it just one-liner code 😅 you remembered it very well

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**

In this situation, yes XGboost performs similarly compare to LSTM but LSTM has an advanced structure in deep learning so overall situation LSTM performs better.

There some points which I want to highlight:

- input features for XGBoost are Bags of words and for the LSTM model, it is Word embedding. so the difference between the two features should be explained.
- The architectural difference between XGBoost and LSTM.
- Limitation of XGBoost should be mentioned.
- LSTM model complexity should be explained.

for more detail about LSTM and XGboost click here

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].

Good 👏 you have used **convert_and_pad** and **review_to_words** correctly here.

```
test_data, test_length = convert_and_pad(word_dict, review_to_words(test_review), pad=500)
```

well done, you have also considered review length at the beginning of the testing.

```
data_pack = np.hstack((test_length, test_data))
data_pack = data_pack.reshape(1, -1)
```

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.

Good 👏 you are using the same method for sagemaker

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`).

Good! the link seems valid 👍
`https://84ek1m138f.execute-api.eu-west-1.amazonaws.com/prod`

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

Perfect 🔥
It would be nice if you have attached a screenshot of your testing.

DOWNLOAD PROJECT

RETURN TO PATH