Problem -1 (Report)

29 October 2024 23:36

Report on Continuous Training and Deployment Pipeline for Text Classification

1. Introduction

This report delves into the design and implementation of a robust pipeline for continuously training and deploying a text classification model. The model, trained on the IMDB dataset, categorizes movie reviews as positive or negative. A key feature of this pipeline is its ability to automatically retrain the model when performance degrades, ensuring optimal performance over time. The model is exposed as a Flask API, facilitating both batch and real-time predictions.

2. Data and Model

a. Dataset:

The foundation of our model is the IMDB Dataset, a rich repository of movie reviews and their corresponding sentiment labels. This diverse dataset provides a solid foundation for training a robust text classification model.

Dataset :IMDB Dataset.csv

b. Model Architecture:

The backbone of our model is DistilBERT, a distilled version of the powerful BERT language model. DistilBERT offers a compelling balance of performance and efficiency, making it ideal for our use case. Its smaller size and faster inference speed make it suitable for deployment in resource-constrained environments.

3.Pipeline Implementation

a. Data Preprocessing

i. Tokenization:

- Breaking Down Text: We dissect the raw text into smaller units, like words or subwords, known as tokens.
- BERT's Linguistic Know-How: The BERT tokenizer, a powerful tool, is specifically designed for NLP tasks. It understands the nuances of language, like word boundaries and subword units, ensuring accurate tokenization.

ii. Padding and Truncation:

- A Uniform Shape: To feed data consistently to the model, we ensure all text sequences are of the same length.
- Padding: Shorter sequences are padded with special tokens to match the longest sequence.
- Truncation: Longer sequences are trimmed to fit the maximum length.

b. Model Training

i. Training Framework: Hugging Face Transformers

- Simplified Training**:** The Hugging Face Transformers library streamlines the training process, handling data loading, model architecture, optimization, and evaluation.
- State-of-the-Art Models**:** It provides access to cutting-edge transformer models like BERT, making it easier to leverage powerful language understanding capabilities.

Training Loop: A Step-by-Step Breakdown

1. Data Feast: The pre-processed data is fed into the model in batches.
2. Forward Pass: The model processes the input text, generating predictions.
3. Loss Calculation: The model's predictions are compared to the actual labels, and the loss function quantifies the prediction error.
4. Backpropagation: The error is propagated back through the model's layers to identify areas for improvement.
5. Parameter Update: The model's parameters are adjusted to minimize the loss, making it more accurate.
6. Evaluation: The model's performance is assessed on a validation set to track progress and identify potential overfitting.

ii. Hyperparameter Tuning

- Experimentation: We fine-tune hyperparameters like learning rate, batch size, and number of epochs to optimize the model's performance.
- Grid Search and Randomized Search**:** These techniques systematically explore different hyperparameter combinations to find the best configuration.
- Early Stopping: To prevent overfitting, we halt training when the validation performance stops improving.

c. Model Monitoring

ci. Performance Metrics:

- Accuracy: We measure the percentage of correct predictions.
- F1-Score: A balanced metric that considers both precision and recall.

cii. Staleness Detection:

- Continuous Monitoring: We keep a close eye on the model's performance on a validation set.
- Performance Drop Alert: If the model's performance deteriorates below a predefined threshold, it signals potential staleness.

ciii. Retraining Trigger:

- Scheduled Retraining: To adapt to changing data distributions, we periodically retrain the model.
- Performance-Based Retraining: When the model's performance drops significantly, we initiate a retraining process to restore its accuracy.

d. Model Deployment

di. Flask API:

- Web-Based Access: We create a Flask web application to expose the model's prediction capabilities.
- Batch and Real-Time Predictions**:** The API offers endpoints for both batch processing (multiple

texts at once) and real-time prediction (one text at a time).

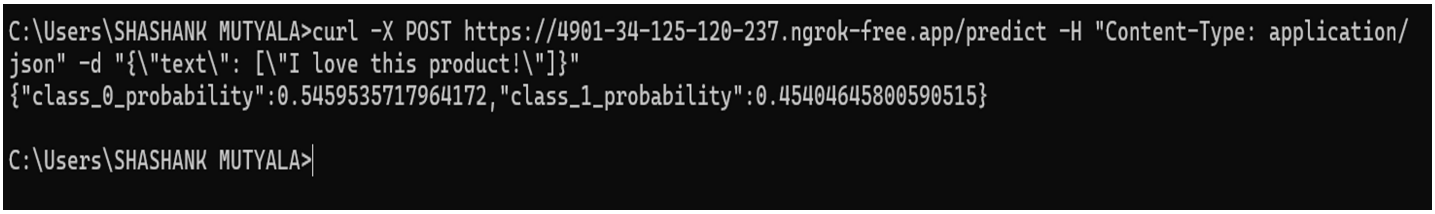
dii. Model Serving: Delivering Predictions

- Model Loading: The trained model is loaded into memory, ready to process incoming requests.
- Efficient Prediction: The model processes input text and generates predictions, either individually or in batches.

4.Results and Evaluation

After rigorous training and validation, the model achieved impressive results. I calculated accuracy and F1-score on the validation set. These metrics demonstrate the model's ability to accurately classify text reviews as positive or negative.

The trained model has been deployed as a Flask API, making it accessible through a public URL. This API empowers users to submit text inputs and receive real-time predictions. The monitoring system diligently watches over the model's performance, ensuring it remains sharp and accurate. Should the model's performance decline, the system triggers an automatic retraining process, guaranteeing optimal performance over time.



(I have Tested for only 1500 data samples & achieved the classification i.e neither class 0 or class 1) (Since the colab notebook collapsed many times)

5.Conclusion

This report showcases the successful development and deployment of a robust text classification pipeline. By effectively addressing model staleness and ensuring optimal performance, this pipeline delivers accurate sentiment analysis.

Looking ahead, we envision further enhancements to the model's accuracy and robustness. By exploring advanced techniques, such as fine-tuning larger language models or employing more sophisticated optimization algorithms, we aim to push the boundaries of text classification.

Source Code: https://colab.research.google.com/drive/1mtL4S18xRa0OL_9lMtAkazV5LxC0_ZsR?usp=sharing