Sentiment Analysis Using SageMaker And Pytorch

Mazi Essoloani Aleza, Dr Ravinder Kaur

Chandigarh University

Abstract.

Machine learning engineers have a lot of data from many online and offline sources. IMDB which stands for internet movie database is one of the available online databases that contain various sets of movie reviews from millions of reviewers. The IMDB dataset comes with labelled and unlabelled reviews; the labels are either positive or negative, and a positive label means that the review is positive and a negative label means that the review is negative. Learning patterns from these millions of online reviews is a challenge as data comes with a lot of incomplete and unstructured texts, and is sometimes meaningless as the reviewers use some unfamiliar abbreviations. This paper focused on building a machine learning model based on Long-Short Term Memory (LSTM) and Recurrent Neural Networks. Accuracy is the performance metric used to measure the robustness of the model. This paper explains the most suitable technique for detecting if sentiment is positive or negative, it also explains a new technique to train, test and deploy a sentiment analyser using Amazon SageMaker. The words in the reviews are converted into an array of vectors, then fed to the model. After training the model, accuracy is 85%.

Keywords. Sentiment Analysis, SageMaker, Pytorch.

1. INTRODUCTION

The need to make a computer understand what text means is more and more needed in today's society, the increasing number of online shopping applications, where the users will review product and a machine algorithm has to rate if it is a positive comment or not. The increasing amount of data available online gives the opportunity to machine learning enthusiasts to learn patterns from the data and come up with a model that can understand at machine level what a text means or if text has a positive or negative sentiment. The need to make computer understand text is needed in various fields such as classifying an email if it is spam or not, getting information from web, text summarization, text translation, and many more. In email classification and sentiment classification, there are some predefined classes but in text translation and summarization there is no predefined class making it more difficult to the approach to finding a good model that could translate well a text. This paper aims to develop a sentiment analyser using Amazon SageMaker, and develop as machine learning algorithm on IMDB dataset to predict the sentiment of a text. The paper also covers the processing of data, the way the data is uploaded on Amazon Web service (AWS) on the S3 bucket, the model building using SageMaker, training, testing, and deploying the model using AWS, and building a website to interfere with the model, the website will take reviews from the users and feed it to the model on the cloud, the model returns the sentiment of the reviews and that sentiment will be displayed on the web application.

2. Related Work

Artificial intelligence (AI) is one of the most active research areas, researchers got their interest in artificial intelligence due to the fact that it has a lot of undiscovered potentials. Both deep learning and machine learning coming under AI have seen an increasing number of researchers trying to understand and discover the underlying potential of these fields. Many researchers worked on developing and deploying models using Amazon Web services (AWS), in this direction Singh Himanshu wrote a book where he explained three useful cases where AWS is used to develop and deploy machine learning models with few numbers of code reducing the work for developers. Zakari et al. wrote a paper where they classified toxic comments, they presented a new method to analyse social media comments using AWS and S3 bucket, they found that LSTM perform better on sentiment analysis with AWS, the accuracy of their model is 70%. Hudgeon et al. wrote a book where they explained how AWS can be used efficiently to develop a machine learning model that can help businesses [3]. Kumar et al. worked on the IMDB dataset; they used a hybrid feature classification using a concatenation of machine learning features with lexicon features [4], their model performed well over many other models, this model is complex and difficult for new researchers to understand. Qaisar and Saeed Mian published an article in 2020 where they used the LSTM model to classify the IMDB dataset, the found an accuracy of 89.9% [8], their model is based on a recurrent neural network (RNN) model, they measured the performance of their model on the basis of accuracy. Md. Rakibul Haque, Salma Akter Lima, and Sadia Zaman Mishu published a paper in 2019 where they used CNN, LSTM, and a combination of CNN and LSTM (LSTM-CNN) to classify IMDB data, they used F-Score as an evaluation metric, their experimental work showed that CNN performed well over LSTM, LSTM-CNN, the CNN model gave an F-Score of 91% [2]. Mathapati et al. developed a classifier for sentiment analysis, they used the IMBD dataset, they also used deep learning techniques to gain higher accuracy and reduce loss and computational time [6]. Rehman et al. developed a model based on the word to vector applied to CNN and LSTM [10], the word to vector translates word into an array of integers, their model outperformed conventional classifier in term of accuracy, precision, recall, and F-Score; they applied their method on Amazon movie review and they got satisfying results.

3. Data

This paper used IMDB for training and testing. IMDB is a large dataset obtained from various internet movie reviewers. It is for binary classification [1], the dataset has two classes either positive or negative, it has 50,000 records of very popular movie reviews split into two sets such as 35,000 for training and the other 15000 for testing making a ration of 70% of data is used for training and 30% for testing, the data has been randomly shuffled before splitting into training and testing. The dataset comes with other records with no class, these unlabelled records can be used for reinforcement learning. They can be downloaded from the Standford University website, each movie has a huge number of reviews, the dataset has taken only 30 reviews from any movie, this is because some movies have more reviews than others, so it is convenient to take the same number of reviews so that we have a same dimensional data for computing.

4. PROPOSED WORK

4.1. Processing the Data

The data comes in a raw file and contains a lot of things that should be cleaned such as punctuation, to have train and test data, this paper divided the data into two sets, one for training and another for testing. The downloaded data contains 25000 training records, 12,500 of which are positive reviews and the remaining 12,500 are negative reviews. The paper combined and randomly shuffled the positive and negative reviews after reading the positive and negative reviews. After shuffling the data, it is the right time to separate it into training and testing. Four variables are created: train X, test X, train y, test y represent respectively the training reviews, test reviews, training labels and test labels. Each variable has a length of 25,000. The data is downloaded from the internet so some HTML tags come with the data and these tags should be removed. The input should be tokenized in a way that words such as exited and exiting are considered as same in the sentiment analysis. To do so this paper used the stopwords from nltk (natural language Toolkit) corpus and stemmer porter and BeautifulSoup A method review to words is then created that uses BeautifulSoup to remove any HTML tag that appears in the data, to tokenize the reviews, the nltk is used. This operation is expensive in time so the method review to words caches the results, removes all punctuation, and convert all word to lower case. While running this method for the second time, it will not take time, it will just load the data save in the cache.

4.2. Upload Data to S3

The training will be done using Amazon SageMaker, so the data should be uploaded to Amazon S3 so that the training code can access it. To begin, the data is saved locally then it is uploaded to S3.

4.2.1. Amazon SageMaker:

SageMaker is a service provided by Amazon which helps every data scientist or any developer to build, train and deploy machine learning models in a rapid way. Amazon SageMaker is a cloud-based service that covers the entire machine learning lifecycle, from data labelling and preparation to selecting an algorithm, training the model, tuning and optimising it for deployment, and making predictions.

4.2.2. AWS- IAM account:

AWS stands for Amazon Web services, IAM stands for Identity and Access Management, IAM is a service that help to access with security AWS services, in the case, a web application has to access the SageMaker endpoints, an IAM role will be assigned to authorize this access.

4.2.3. S3:

The training data is saved locally into a comma separated value file called ‘train.csv’, this format is very important as it will be used while writing training code. It has a label, length, and an array of length 500 containing an array of integers obtained from the words in the review. The data is then uploaded into the SageMaker default S3 bucket, saving the data in this bucket help to access it in the training phase of the model.

4.3. Introduction to SageMaker

4.3.1. Building the Model:

The model is composed of three things such as the artifacts of the model, training code, and testing code, they interact with each other, Amazon provides containers and gives the option to add a new code. In the training folder, there is the model file model.py. In the building process of the model, the torch.nn is imported then a python class is created and inherits the module of the torch.nn, the class init of the class takes parameters such as self ‘embedding dim’, ‘hidden dim’ and ‘vocab size’. The ‘self.embedding’ is assigned to be an instance of the Pytorch neural network embedding module having the vocab size, embedding dimension and padding set to zero as parameters. The LSTM is inherited from the torch neural network module and takes parameters such as embedding dim and hidden dim. As it is a long short-term classifier, dense and sigmoid layers are important so a dense layer is created from the linear model of torch neural network. A sigmoid activation function is then added, a forward function is very important, it takes the self and x as parameters and returns the sigmoid output, it has an output range of 0 to 1. Before it returns a value, it transposes the variable x to get the lengths of the reviews, it embeds the reviews and feeds them into the LSTM defined in the initialization part of the class, the LSTM output is then passed to the dense layer and the function finally returns the sigmoid activation of the output of the dense layer. To improve the performance of the model, the embedding dimension, hidden dimension, and vocab size should be tuned well, these parameters will be modifiable in the code without modifying the training script. A small portion of the training set is taken to see if the training script is running well, error-free, and produces the desired output.

4.3.2. Training the Model:

The model is a PyTorch model builtin Amazon SageMaker, there is a need to specify an entry point, it is the python file that will be run when in the training phase of the model, it contains all necessary code to train a model, SageMaker passes hyperparameters to a training script, these hyperparameters are then passed to and used by the training code. Some steps of the training in SageMaker are explained in the following: Starting the training job, launching requested ML instances, Downloading the input data, Downloading the training image, uploading generated training model, completing the training. BCELoss (Binary Cross Entropy Loss) is used to measure the loss. BCELoss measures the Binary Cross Entropy between the target and the output.

4.4. Test the model

Testing is an important part of any project, a machine learning model can perform well on training data and give worse results on live data that is why testing is an important step in building and deploying a machine learning model. This model will be tested by deploying and sending the test data on the endpoint that is already deployed, this is also a way to check if the deployed endpoint is working well.

4.4.1. Deploy the Model for Testing*:*

The model is trained and it should be tested to see how it performs on unseen reviews, the model takes two variables as input: the review length and an array containing a sequence of integers of length, the array has a size of 500, these integers represent the word in the review, reviews are encoded to integers using the word dict method. There should be a function that loads the saved model, this function is called model fn(), and it takes one parameter: the directory of the saved model, this function is in the python script that is used for the entry point.

4.4.2. Test the Deployed Model:

Once the model is deployed, it can be used for testing, test data is then set and the model gives the output and accuracy of 85%, a precision of 83% and recall of 81%. After deploying the model for testing and doing the testing, it is good to delete the endpoint otherwise it will keep running.

4.5. Deploy the Trained Model for Web App

Now that the model is deployed and tested, it is time to write some inference code so it can receive a review, process it, and give tell if the review is positive or negative. The estimator need to use the entry script and the directory of the saved script, the input of the model will be a string so the words in the string should be processed and converted into integers of length 500 by the method review to word and convert and pad functions. the following function: model fn, input fn, output fn, and predict fn should be deployed into the SageMaker as the container need to use them, the model function loads the model, the input function takes the input and has the role of sending the input to the model endpoint, The output function returns the output to the caller of the model endpoint, the predict function is where the prediction is done. To deploy the model, a new PyTorchModel object is needed and it points to the model artifacts created during the training and to the inference code that will be used, after this a deploy function is provided by Amazon SageMaker, it should be called so it can launch the deployment container.

4.6. Use the Deployed Model for Web App

Now that is model is saved and is working well on Sage-Maker, it is time to develop a website where a user can type a review and receive the sentiment of that review, to achieve this, the web application should access the SageMaker endpoint, to access the endpoint it needs to be authenticated with AWS using an IAM role that contains the access to the endpoint of the SageMaker, this can be done in an easier way using more AWS services. The next diagram shows the interaction between different services from the left, the web application sends and receives data from the API. It collects a user review of a movie, sends it to the API, and expects a positive or negative sentiment from the API. On the right side, there is the model, the trained and deployed model. The API and the lambda function are in the middle. The lambda function [7] is a Python function that executes whenever a given event occurs; this function has permission to send and receive data from the SageMaker endpoint to execute the Lambda function, an API Gateway is needed to create a new endpoint, the new endpoint is the URL used to listen to the data that will be sent. Once the API Gateway has the data, it forwards it to the Lambda function, the Lambda function passes the data to the model, the model returns the sentiment to the Lambda function which returns it to the API, and finally, the API sends the sentiment to the web application and the web application will display the sentiment to the reviewer.

4.6.1. Setting Up a Lambda Function*:*

The Lambda function executes every time the public API sent data, it processes the data and sends it to the SageMaker endpoint.

Creation of an IAM Role for the Lambda function: The Lambda function is set up to call the API so it should have the necessary permissions to do so; to achieve this a role shou ld be created and later given this role to the Lambda function. The role will be created using AWS Console, in the IAM page there is an option for Roles, then Create role, this should be under the type of trusted entity and then choose the service as Lambda, this will be used in the role then click next, it will navigate to the permissions. A search box appears, type SageMaker and check the box which is next to AmazonSageMakerFullAccess policy, click next, and review. A name is assigned to the role, and that name will be necessary in the next processes, in this example the name of the role is LambdaSageMakerRole, then click Create role, and the role is then successfully created.

Creation of a Lambda function: In the AWS console, there is an option in the navigation called AWS Lambda, create a function in this section and name the function, this project used the Python 3.6 version, Choose the role created by the previous part, click on Create Function, the Lambda function is then successfully created; In the next page, all the information about the Lambda function is available and an editor is available to write the code that will be executed when the Lambda function is called. The editor will contain custom code and save the endpoint.

4.6.2. Setting Up API Gateway*:*

On the AWS console, there is an option to create an API gateway, once clicked on that option it will be easily created, a method is also created from the same console to trigger the lambda function created previously. From the AWS console, the API gateway can be deployed and can access the SageMaker model.

5. DEPLOYING THE WEB APP AND RESULTS

Now that the model is deployed on AWS, it is time to build a simple website that will take user review and returns the sentiment of that particular review. The SageMaker endpoint must be launched and running for the web app to connect with it. This implies that every time the endpoint runs, AWS will charge. The endpoint is left running when the web app is in use but turned off when the web application is not used, else it will end up with a relatively huge AWS bill. A website is then built, it has a question asking if a review is positive or negative, then a paragraph saying to “Enter your review below and click submit to find out...” then there is a text box where a review can be typed, after submitting the review, the sentiment will display. If the sentiment is positive it will display “Your sentiment is positive” otherwise it will display “your sentiment is negative” The web app that interferes with the model on the cloud, the website work well and gives the review but an error occurs when the endpoint is deleted because the web app will not be able to access the model if there is no endpoint. Using SageMaker saves a lot of time for developers as it is easy to setup, train test and deploy a model.

6. CONCLUSION

This paper presented how to build, train, test, and deploy a sentiment analyser using a cloud service, it uses AWS, and, Using AWS saves a lot of time for the programmer and reduces code, SageMaker is a robust machine learning model builder and having the model on the AWS cloud reduce the risks of the model being hacked , model on local servers may suffer from unresponsiveness, the model will not be able to give output due to the server being too busy, locals servers may crash or be down several times due to technical problem, using AWS has resolved all these issues. In future work, this paper will be built using another cloud provided and compare the billing, the effectiveness and the robustness of each cloud to find which one is better.

7. REFERENCES

[1] Nehal Mohamed Ali, Marwa Mostafa Abd El Hamid, and Aliaa Youssif. Sentiment analysis for movies reviews dataset using deep learning models. International Journal of Data Mining & Knowledge Management Process (IJDKP) Vol, 9, 2019.

[2] Md Rakibul Haque, Salma Akter Lima, and Sadia Zaman Mishu. Performance analysis of different neural networks for sentiment analysis on imdb movie reviews. In 2019 3rd International Conference on Electrical, Computer & Telecommunication Engineering (ICECTE), pages 161– 164. IEEE, 2019.

[3] Doug Hudgeon and Richard Nichol. Machine learning for business: using Amazon SageMaker and Jupyter. Simon and Schuster, 2019.

[4] HM Kumar, BS Harish, and HK Darshan. Sentiment analysis on imdb movie reviews using hybrid feature extraction method. International Journal of Interactive Multimedia & Artificial Intelligence, 5(5), 2019.

[5] Edo Liberty, Zohar Karnin, Bing Xiang, Laurence Rouesnel, Baris Coskun, Ramesh Nallapati, Julio Delgado, Amir Sadoughi, Yury Astashonok, Piali Das, et al. Elastic machine learning algorithms in amazon sagemaker. In Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data, pages 731–737, 2020.

[6] Savitha Mathapati, Amulya K Adur, R Tanuja, SH Manjula, and KR Venugopal. Collaborative deep learning techniques for sentiment analysis on imdb dataset. In 2018 Tenth International Conference on Advanced Computing (ICoAC), pages 361–366. IEEE, 2018.

[7] Danilo Poccia. AWS Lambda in Action: Event-driven serverless applications. Simon and Schuster, 2016.

[8] Saeed Mian Qaisar. Sentiment analysis of imdb movie reviews using long short-term memory. In 2020 2nd International Conference on Computer and Information Sciences (ICCIS), pages 1–4. IEEE, 2020.

[9] Nisha Rathee, Nikita Joshi, and Jaspreet Kaur. Sentiment analysis using machine learning techniques on python. In 2018 Second International Conference on Intelligent Computing and Control Systems (ICICCS), pages 779–785. IEEE, 2018.

[10] Anwar Ur Rehman, Ahmad Kamran Malik, Basit Raza, and Waqar Ali. A hybrid cnn-lstm model for improving accuracy of movie reviews sentiment analysis. Multimedia Tools and Applications, 78(18):26597– 26613, 2019.

Biographies

Mazi Essoloani Aleza received the bachelor's degree in computer engineering from Alakh Prakash Goyal Shimla University in 2021. He is currently pursuing his master's degree in computer engineering from Chandigarh University.

Dr. Ravinder Kaur is an Assistant Professor in the Department of Computer Science and Engineering (CSE) at University Institute of Engineering, Chandigarh University, Punjab, India. She received her PhD from UIET Panjab University, Chandigarh India. She received her PhD from UIET Panjab University, Chandigarh India. She received her Undergraduate Degree with Distinction in 2011, received her Postgraduate Degree (ME in Information Security) with Distinction in 2013 from PEC University of Technology, Chandigarh. She has published numerous papers in refereed international journals and conference proceedings with a good number of citations in Google Scholar.