Google Colaboratory : Tool for Deep Learning and Machine Learning Applications

Praveen Gujjar J.[1] and *Naveen Kumar V.*[2]

Abstract

Google Colaboratory is a cloud based service which is also known as Colab. Google Colab is based on Jupyter Notebook where machine learning and deep learning concepts can be implemented. Google Colab provides free access to Graphical Processing Unit (GPU) which is very much required to disseminate deep learning concepts. This paper presents a tool for performing a deep learning application which is in Google Colaboratory and also discusses the performance of transfer learning model xlm and MobileNetV2 in Google colab. This GPU may help many researchers to carry out their research work with high end infrastructure to implement the concept of machine learning and deep learning concept.

Keywords : Deep Learning, Google Colab, MobileNetV2, Transfer learning, Xlm-roberta

I. INTRODUCTION

Data is the need of the hour. Deep learning and machine learning help in analysing data and automating many applications in the real world scenario. To implement any deep learning and machine learning concept, researchers may require higher end infrastructure to carry out their work. Some of the examples of deep learning and machine learning applications are natural language recognition, sentiment analysis, e-commerce suggestions/recommendations, and social network recommendations etc. [1]. Many deep learning and machine learning applications rely on heavy computations on massive datasets. GPU is an acronym for graphical processing units which may help in parallel task processing [2]. Most of the applications in deep learning and machine learning use NVIDIA GPUs [3] [4]. Google Colab will help in providing the necessary infrastructure to carry out the task free of cost by default Google Colab with 12 GB RAM and 32 GB of disk space. If any researcher wants to use more, Google Cloud provides necessary infrastructure in a pay-by-hour manner to use the hardware with a fully configured GPU for deep learning applications [5] [6]. The paper is organized into four parts. Part I is the introduction ; Part II Background and Related works ; Part III MobileNetV2 and xlm ; Part IV presents the results and discussion ; Part V presents conclusions.

II. BACKGROUND AND RELATED WORKS

In this paper Deep learning and machine learning applications such as MobileNetV2 and XLM-Roberta models have been discussed. MobileNetV2 and xlm both are trained models which are also called ***Transfer Learning***. MobileNetV2 and xlm model has been implemented in Google Colab. MobileNetV2 is used for image classification and image prediction. Xlm is used for natural language processing. Xlm is mainly used to classify text based on the certain categories. Amazon and Microsoft both will provide high performance computing

Manuscript Received : May 5, 2021 ; Revised : June 8, 2021 ; Accepted : July 10, 2021. Date of Publication : August 5, 2021

[1] *P. Gujjar J.,* Associate Professor ; Email : dr.praveengujjar@cms.ac.in ; ORCID iD : https://orcid.org/0000-0003-0240-7827

[2] *N. Kumar. V.,* Associate Professor ; Email : Naveenkumar_v@cms.ac.in ; ORCID iD : https://orcid.org/0000-0002-2618-9978

[1, 2] CMS Business School, Jain (Deemed-to-be University), No. 17, Sheshadri Road, Gandhi Nagar, Bengaluru – 560 009, Karnataka.

DOI : https://doi.org/10.17010/ijcs/2021/v6/i3-4/165408

hardware and the necessary infrastructure to carry out deep learning and machine learning applications. Relatively Google Colab is cost effective [7] [8]. Expósito et al. [9] showed that Amazon EC2 is having the performance bottlenecks in application scalability, especially with high performance computing infrastructure. A convolution neural network is used for image classification and prediction [10]. Transfer learning is nothing but a reuse of the trained model for classification and prediction [11]. Sentiment analysis and opinion mining can be done using the textblob library using the tool Google colab [12] [13].

III. MOBILENetV2 AND XLM

Convolutional neural network convolution layer, pooling layers, and fully connected layer. An additional layer may be used for the complex set of models. Matrix vector multiplication is at the heart of how data and weights are represented. XLM-Roberta trained models are used in categorizing the text based on certain categories. MobileNetV2 and xlm both are trained models which are also known as transfer learning. MobileNetV2 and xlm model has been implemented in Google Colab. MobileNetV2 is used for image classification and image prediction. Xlm is used for natural language processing. Xlm is mainly used to classify the text based on certain categories. MobileNetV2 builds upon the ideas from MobileNet, using depthwise separable convolution as efficient building blocks. MobilNetV2 introduces two important features : (a) shortcut connection between the bottlenecks ; and (b) linear bottlenecks.

IV. DISCUSSION AND RESULTS

In this paper, transfer learning models have been used : a) XLM ; and b) MobileNetV2. Both trained models are used in Google Colab. MobileNetV2 is a trained model which is capable of classifying images. XLM-Roberta trained models are used in categorizing text based on certain categories. XLM-RoBERTa is a scaled cross-lingual sentence encoder. It is trained on 2.5T of data across 100 languages that is filtered from Common Crawl. XLM-R achieves state-of-the-art results on multiple cross-lingual benchmarks. Necessary library to use xlm-roberta model author has used transformers [sentencepiece]. The pseudocode for the xlm-roberta in Google Colab is as following:

```
classifier = pipeline("zero-shot-classification",model="joeddav/xlm-roberta-large-xnli")
candidate_labels = ["English", "public health", "politics"]
classifier("who will vote for you", candidate_labels)
```

The result shows that the sentence “who will vote for you is 64% belongs to category politics, 27% belongs to category English and 8% belongs to public health.

The result is given as follows :

```
{'labels': ['politics', 'English', 'public health'],
'scores': [0.6409937739372253, 0.27375608682632446, 0.08525016158819199],
'sequence': 'who will vote for you'}
```

MobileNetV2

MobileNet is the first version of the transfer learning compare to MobileNetV2. MobileNet uses 16 MB of disk space. MobileNetV2 uses both point wise and depth wise convolution neural network. However, MobileNetV2 introduces shortcut connections between the bottlenecks and linear bottlenecks between the layers. One of the advantages of using MobileNetV2 is that it uses only 14 MB of disk space. The author used MobileNetv2 to classify and predict the image as shown in Fig. 1.

The pseudocode for the MobileNetV2 in Google Colab to classify the image is given as follows :

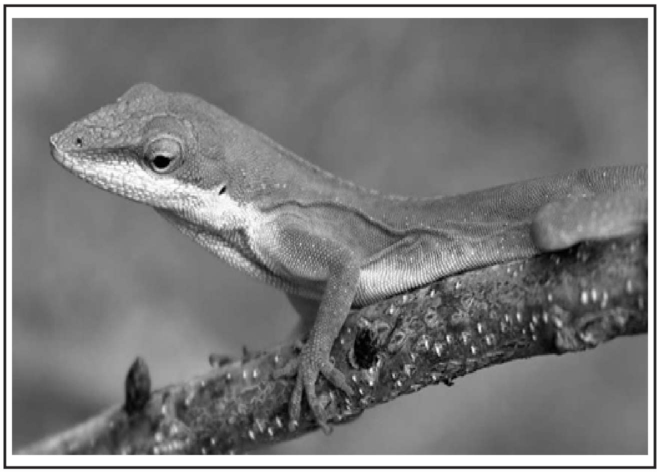

Fig. 1. Input Image to Classify Using MobileNetV2

```
img =
tf.keras.preprocessing.image.load_img('/content/drive/
My Drive/Example/lizard.jpg', target_size=[224, 224]')
x = tf.keras.preprocessing.image.img_to_array(img)
model = tf.keras.applications.MobileNetV2()
predictions = model(x)
```

The output for the above pseudocode shows that the input image is "American Chameleon"

Output (Fig. 1).

```
['American chameleon' 'green lizard' 'agama' 'whiptail'
'tree frog']
```

V. CONCLUSION

In this paper, the researcher presented Natural Language Processing using Transfer Learning in Google Colaboratory xlm-roberta and MobileNetV2 for image classification using a trained model. MobileNetV2 uses only 14 MB of disk space. This paper has attempted to demonstrate the use of Google Colab for any Deep Learning application. The overall conclusions of this study are summarized as follows :

- MobileNetV2 gives the accuracy of 90%
- Xlm-Roberta accuracy may vary from 60% to 80% depending on the context.

For future research direction, researchers can employ the GPU to process the deep learning application using Google Colab.

AUTHORS' CONTRIBUTION

Dr. Praveen Gujjar J. and Professor Naveen Kumar V. conceived the idea to study Transfer Learning. The implantation of the Transfer Learning on Google Colab was carried out jointly be Dr. Praveen Gujjar and Prof. Naveen Kumar V. Dr. Praveen Gujjar J. wrote the manuscript in consultation with Prof. Naveen Kumar.

CONFLICT OF INTEREST

The authors certify that they have no affiliations with or involvement in any organization or entity with any financial interest, or non-financial interest in the subject matter, or materials discussed in this manuscript.

FUNDING ACKNOWLEDGEMENT

The authors received no financial support for the research, authorship and/or for the publication of this article.

REFERENCES

[1] Y. Bengio, G. Hinton, and Y. LeCun, *"Deep learning," Nature,* vol. 521*,* no. 7553, p. 436, 2015.

[2] A. R. Brodtkorb, C. Dyken, T. R. Hagen, J. M. Hjelmervik, and O. O. Storaasli, "State-of-the-art in heterogeneous computing," *Scientific Programming,* vol. 18*,* no. 1, pp. 1–33, 2010. https://doi.org/10.3233/SPR-2009-0296

[3] NVIDIA Corporation, *"Tesla V100 performance guide: Deep learning and HPC applications,"* NVIDIA Corporation Whitepaper, 2016.

[4] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. H. Katz, A. Konwinski, G. Lee, D. A. Patterson, A. Rabkin, I. Stoica, and M. Zaharia, *"Above the clouds: A Berkeley view of cloud computing,"* Technical Report No. UCB/EECS-2009-28, EECS Department, University of California, Berkeley, Technical Report 2009. [Online]. Available: https://www2.eecs.berkeley.edu/Pubs/TechRpts/2009/EECS-2009-28.html

[5] NVIDIA Corporation, *"Introduction to NVIDIA GPU cloud,"* NVIDIA Corporation Application Note, 2018.

[6] Google, *"Colaboratory : Frequently asked questions,"* 2018. [Online]. Available: https://research.google.com/colaboratory/faq.html. Accessed on June 21, 2018.

[7] G. Juve, E. Deelman, K. Vahi, G. Mehta, B. Berriman, B. P. Berman, and P. Maechling, "Scientific workflow applications on Amazon Ec2," in *2009 5th IEEE Int. Conf. on E-Science Workshops.* IEEE, pp. 59–66, 2009.

[8] K. R. Jackson, L. Ramakrishnan, K. Muriki, S. Canon, S. Cholia, J. Shalf, H. J. Wasserman, and N. J. Wright, "Performance analysis of high-performance computing applications on the Amazon Web Services cloud," in *2010 IEEE Second Int.Conf. on Cloud Computing*

Tech.and Sci. IEEE, 2010, pp. 159–168. Doi: https://doi.org/10.1109/CloudCom.2010.69

[9] R. R. Expósito, G. L. Taboada, S. Ramos, J. Touriño, and R. Doallo, "Performance analysis of HPC applications in the cloud," *Future Generation Comp. Syst.,* vol. 29*,* no. 1, pp. 218–229, 2013. Doi: https://doi.org/10.1016/j.future.2012.06.009

[10] N. Sharma, V. Jain, V., and A. Mishra, "An analysis of Convolutional Neural Networks for image classification," *Procedia Comp. Sci.,* vol. *132,* pp. 377–384, 2018. Doi: https://doi.org/10.1016/j.procs.2018.05.198

[11] M. Shaha, and M. Pawar, "Transfer learning for image classification," In *2018 Second Int. Conf. on Electronics, Communication, and Aerospace Tech.* (ICECA), IEEE, 2018, pp. 656–660. Doi: https://doi.org/10.1109/ICECA.2018.8474802

[12] P. Gujjar J. and P. Kumar H. R., "Sentiment analysis for running text in Email conversation," *Int. J. of Comp. Sci. Eng.,* vol. 9*,* no. 4, pp. 259–263, 2020. [Online]. Available: http://www.ijcse.net/docs/IJCSE20-09-04-007.pdf

[13] T. Manjunatha and P. Gujjar J., "Performance analysis of Indian information technology companies using DuPont Model," *IUP J. of Manage. Res.,* vol. 17, no. 4, pp. 7-14, 2018. [Online].Available: https://www.proquest.com/openview/dd55d1644ea7a53a495edee2d4395fc8/1?pq-origsite=gscholar&cbl=54462

About the Authors

Dr. Praveen Gujjar J. has completed B.E. (Information science), M.B.A. in Technology Management, M. Tech in Computer Science and Engineering, and Ph.D. in Business Administration. Dr. Praveen won second prize in Doctoral competition organized by AIMS at Symbiosis Institute of Management Studies, Pune on January 3, 2019.

Prof. Naveen Kumar V. worked in the corporate sector for eight and a half years. He worked as a Process Associate in HCL & Market Analyst in Reuters for two years and three months and Senior Analyst and Team Manager in Deutsche Bank for four years and two years respectively.