ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 5, Issue 2, February 2018

Deep Neural Networks for Big Image Data Classification

[1] R. Swathi, [2] Dr.R. Seshadri
[1] Research Scholor, [2] Professor
[1][2] SV University,Tirupati

Abstract - **With big data development in biomedical and medical industries, accurate examination of medical data benefits early disease discovery, patient care and group administrations. In any case, the examination accuracy is lessened when the nature of therapeutic information is deficient. Medical imaging plays a vital role in diagnostic healthcare and deals with a high volume of data collection and processing. In this paper, we streamline machine learning algorithms for classification and prediction of chronic disease outbreak in disease-frequent groups. Deep Learning has emerged as another era in machine learning and is applied to various image processing applications. The fundamental motivation behind the work exhibited in this paper is to apply the idea of Deep Learning algorithms to be specific, Convolutional neural networks (CNN) in image classification. This paper presents the classification of images using deep learning algorithms through spark. Classified different types of skin cancers using Convolutional neural networks in distributed environment achieved in less time with more accuracy.**

Keywords: Big Data, Convolutional neural networks, Deep learning, Machine learning, Spark.

I. INTRODUCTION

We live in the data age. It is difficult to quantify the aggregate volume of information put away electronically however an IDC gauge put the span of the "computerized universe" at 0.18 zettabytes in 2006, and is determining a ten times development by 2011 to 1.8 zettabytes. Facebook hosts approximately 10 billion photos, taking up one petabyte of storage. [1] Statistics shows that 500 terabytes of data is generating from social media sites like facebook and twitter.etc.

Organizations likewise need to consider new wellsprings of information created by machines, for example, sensors. Other new data sources are human produced, for example, information from online networking and the click stream information created from site associations. Moreover, the accessibility and appropriation of more up to date, more capable cell phones, combined with pervasive access to worldwide systems will drive the production of new sources for information.

This information creates in type of images and videos etc. [2] The volume of information being made openly accessible expands each year, as well. Data is increasing, and big data is produced. Big data alludes to informational indexes whose size is past the capacity of database programming devices to catch, store, oversee and investigate. Because of use of internet and internet of things high volumes of data is generating and are different types of data.

From above analysis we can stat that more web data is creating from 2000 onwards web application data includes more unstructured data and log files. In this manner enormous information incorporates immense volume high speed and extensible assortment of information. Traditional database tools failed for storing, processing, curating and analysing these large volumes, high velocity and variety of data. Apache introduced Hadoop opensource framework for storing, accessing, and analysing such huge data.

Anyone working in Big Data area will realize what MapReduce does and what its deficiencies are. It is not totally reasonable for say that there are weaknesses because MapReduce alongside HDFS was a wonder when it discharged. Be that as it may, at this moment Spark has overwhelmed the world. Presently is a decent time to comprehend the contrasts amongst Spark and MapReduce.
The difference in Spark is that it performs in-memory preparing of information. This in-memory preparing is a speedier procedure as there is no time spent in moving the information to and from disk, while MapReduce requires a considerable measure of time to play out these in and out operations accordingly expanding inactivity.

Big data is not magic. It does not matter how much data you have if you can't make sense of it. Over the previous decade, machine learning methods have been broadly received in various gigantic and complex information serious fields, for example, medicine [3], astronomy, science, etc, for these strategies give conceivable answers for mine the data covered up in the information. By and by, as the ideal opportunity for huge information is coming, the gathering of informational

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 5, Issue 2, February 2018

collections is so extensive and complex that it is hard to manage utilizing customary learning strategies since the set-up procedure of gaining from ordinary datasets was not intended to and won't function admirably with high volumes of information. For example, most customary machine learning algorithms are intended for information that would be totally stacked into memory [4], which does not hold any more with regards to enormous information. In this manner, although gaining from these various information is relied upon to bring noteworthy science and engineering advances along with improvements in quality of our life , it brings tremendous challenges at the same time.

All machine learning algorithms work iteratively. As we have seen before, iterative calculations include I/O bottlenecks in the MapReduce usage. Spark with the help of Mesos – a distributed system kernel, caches the intermediate dataset after each iteration and runs multiple iterations on this cached dataset which reduces the I/O and helps to run the algorithm faster in a fault tolerant manner.
Spark has a built-in scalable machine learning library called MLlib which contains high-quality algorithms that leverages iterations and yields better results than one pass approximations sometimes used on MapReduce. And also, which provides more features like SparkR, SparkSQL and SparkMLlib.

II. LITERATURE REVIEW

The literature in this area includes work that has been done in the fields of big data, Spark, machine learning, deep learning and image processing. Big data are presently quickly growing in all science and engineering disciplines. While the capability of these enormous data is without a doubt huge, completely comprehending them requires better approaches for considering and novel learning procedures to address the different difficulties. In this paper author, exhibited writing study of the most recent advances in looks into on machine learning for big data processing.[5]

With the spreading pervasiveness of Big Data, many advances have as of late been made in this field. Structures, for example, Apache Hadoop and Apache Spark have picked up a considerable measure of footing over the previous decades and have turned out to be hugely prevalent, particularly in ventures. It is winding up progressively apparent that successful enormous information examination is vital to taking care of artificial insight issues. In this way, a multi-calculation library was executed in the Spark framework, called MLlib. While this library bolsters various machine learning algorithms, there is still degree to utilize the Spark setup proficiently for exceptionally time-intensive and computationally costly methodology like deep learning learning. [6]

Apache Spark is a dispersed memory-based registering system which is characteristic appropriate for machine learning. Contrasted with Hadoop, Spark has a superior capacity of computing. In this paper author break down Spark's essential structure, centre innovations, and run a machine learning occasion on it. [7]

MLlib, Spark's open-source appropriated machine learning library. MLlib gives proficient usefulness to an extensive variety of learning settings and incorporates a few basic stastical, optimization, and linear algebra primitives. Delivered with Spark, MLlib bolsters a few dialects and gives an abnormal state API that use Spark's rich environment to rearrange the improvement of end-to-end machine learning pipelines [8].

In this paper author presents feature extraction, include determination and machine learning-based order strategies for dust acknowledgment from images. The quantity of pictures is little contrasted both with the quantity of inferred quantitative highlights and to the quantity of classes. The primary subject is examination of the adequacy of 11 highlight extraction/include choice calculations and of 12 machine learning-based classifiers. It is discovered that a portion of the predefined highlight extraction/choice calculations and a portion of the classifiers displayed steady conduct for this dataset [9].

III. WORKING OF SPARK AND CNN

A.CNN

A neural network system is an arrangement of interconnected manufactured "neurons" that trade messages between each other. The associations have numeric weights that are tuned amid the preparation procedure, so an appropriately prepared system will react effectively when given a picture or example to perceive. The system comprises of multiple layers of highlight recognizing "neurons". Each layer has numerous neurons that react to various mixes of contributions from the past layers. The layers are developed so the primary layer identifies an arrangement of crude examples in the info, the second layer recognizes examples of examples, the third layer distinguishes examples of those examples, et cetera. Regular CNNs utilize 5 to 25 distinct layers of pattern recognition.
A CNN is a unique instance of the neural network. A CNN

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 5, Issue 2, February 2018

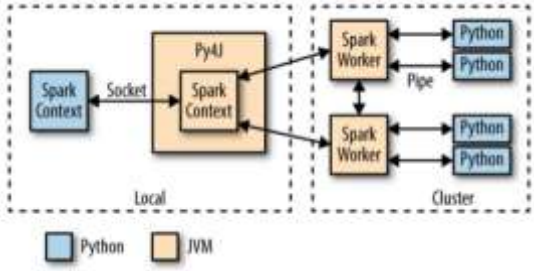
comprises of at least one convolutional layers, frequently with a subsampling layer, which are followed by at least one completely associated layers as in a standard neural network. The outline of a CNN is spurred by the disclosure of a visual system, the visual cortex, in the brain. The visual cortex contains a considerable measure of cells that are responsible of distinguishing light in little, covering sub- regions of the visual field, which are called receptive fields. These cells go about as neighbourhood channels over the information space, and the more mind-boggling cells have bigger responsive fields [10-12]. The convolution layer in a Convolutional Neural Networks plays out the capacity that is performed by the cells in the visual cortexCNNs are utilized as a part of assortment of territories, including image and speech recognition, natural language processing, and video analysis[13-14].

B. Spark

In this work we exhibit MLlib, Spark's conveyed machine learning library, and the biggest such library. The library targets expansive scale taking in settings that advantage from data parallelism or model-parallelism to store and work on data or models. MLlib comprises of quick and versatile usage of standard learning algorithms for basic learning settings including classification, regression, collaborative filtering, clustering, and dimensionality reduction. It likewise gives an assortment of basic insights, linear algebra, and optimization natives. Written in Scala and utilizing local (C++ based) direct variable-based math libraries on every hub, MLlib incorporates Java, Scala, and Python APIs, and is discharged as a component of the Spark project under the Apache 2.0 license.

C. PySpark

At the point when PySpark's Python interpreter begins, it additionally begins a JVM with which it conveys through a socket. PySpark utilizes the Py4J project to deal with this correspondence. The JVM capacities as the genuine Spark driver and loads a JavaSparkContext that communicates with the Spark executors over the clusters.



Python API calls to the SparkContext protest are then converted into Java API calls to the JavaSparkContext. For instance, the usage of PySparks sc.textFile() dispatches a call to the .textFile() strategy for the JavaSparkContext, which at last speaks with the Spark agent JVMs to stack the content information from HDFS.

The Spark executors on the group begin a Python interpreter for each center, with which they convey data through a pipe when they have to execute client code[15].A Python RDD in the local PySpark client compares to a PythonRDD object in the local JVM. The data related with the RDD really lives in the Spark JVMs as Java objects. For instance, running sc.textFile() in the Python interpreter will call the JavaSparkContext's textFile()method, which loads the information as Java String objects in the group. Also, stacking a Parquet/Avro record utilizing newAPIHadoopFile will stack the items as Java Avro objects.

At the point when an API call is made on the Python RDD, any related code (e.g., Python lambda function) is serialized by means of "cloudpickle " and distributed to the executors. The information is then changed over from Java items to a Python-perfect portrayal (e.g., pickle questions) and gushed to agent related Python executors through a pipe. Any essential Python handling is executed in the mediator, and the subsequent information is put away back as a RDD (as pickle questions as a matter of course) in the JVMs.

IV. METHODOLOGY

Neural Networks have seen dynamite improvement amid the most recent couple of years and they are presently the best in class in image recognition and robotized interpretation. Tensor Flow is another system released by Google for numerical calculations and neural networks. In this Paper, we will use Tensor Flow and Spark together to train and apply deep learning models.

1.Hyperparameter Tuning: utilize Spark to locate the best arrangement of hyperparameters for neural network training,, prompting 10X lessening in preparing time and 34% lower error rate.

2. Deploying models at scale: utilize Spark to apply a trained neural network display on a lot of information.

A case of a deep learning machine learning (ML) system is artificial neural networks. They take an intricate input, for example, an image(picture) or a sound, and afterward apply

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 5, Issue 2, February 2018

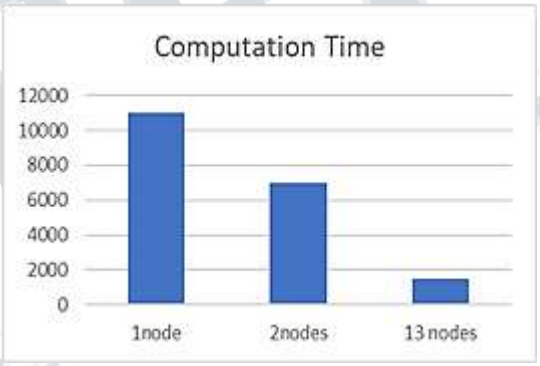
complex numerical changes on these signs. The yield of this change is a vector of numbers that is less demanding to control by other ML algorithms. Artificial neural networks play out this change by impersonating the neurons in the visual cortex of the human brain.

Similarly, as people figure out how to translate what they see, simulated neural systems should be prepared to perceive examples that are 'interesting'. For instance, these can be basic examples, for example, edges, circles, however they can be substantially more complicated. The real procedure of building a neural network, in any case, is more confounded than simply running some function on a dataset. There are ordinarily various vital hyperparameters (setup parameters in layman's terms) to set, which influences how the model is learned. Picking the correct parameters prompts elite, while bad parameters can prompt delayed preparing and awful execution. In practice, machine learning specialists rerun a similar model various circumstances with various hyperparameters to locate the best set. This is an established method called hyperparameter tuning. When assembling a neural network, there are numerous imperative hyperparameters to pick precisely. For example:

• Number of neurons in each layer: Too couple of neurons will lessen the articulation energy of the system, yet an excessive number of will significantly build the running time and return loud gauges.
• Learning rate: If it is too high, the neural network will just concentrate on the last couple of tests seen and slight all the experience collected some time recently. If it is too low, it will take too long to achieve a decent state.
The intriguing thing here is that even though Tensor Flow itself isn't conveyed, the hyperparameter tuning process is "embarrassingly parallel" and can be appropriated utilizing Spark. For this situation, we can utilize Spark to communicate the regular components, for example, information and model portrayal, and afterward plan the individual tedious calculations over a group of machines in a blame tolerant way.

V. RESULT AND CONCLUSION

Downloaded the images from isic archive, and loaded, it is undergone the pre-processing techniques using median filter it, transforms it to a series then detrend and computed a fourier transform on each pixel, then it is converted in to an array [16-17].
Images are of 2D images. By using Thunder package in python and is used for the analysis of images and time series information in Python. It gives information structures and algorithms to loading, processing, and analysing these information, and can be valuable in an assortment of areas, including neuroscience, medical imaging[18-20], video handling, and geospatial and weather analysis. It can be utilized locally, yet in addition bolsters large scale analysis through the distributed computing engine spark.
All data structures and examinations in Thunder are intended to run indistinguishably and with similar API whether local or distributed. Local operations can be supported through numpy and distributed operations can be suppoterd through Spark. After installing Spark and thunder, all the loading functions are passed through SparkContext. Which inturn automatically creates sc variable and data loading function is passed as parameter to this variable [21-22].
And by using cnn algorithm classified these images of different types of skin cancer are classified. Large image dataset is classified by using python in spark in distributed manner.



The accuracy with the default set of hyperparameters is 99.21%. Our best outcome with hyperparameter tuning has a 99.37% accuracy on the test set, which is a 33% lessening of the test blunder. Conveying the calculations scaled straightly with the quantity of hubs added to the bunch: utilizing a 13-hub group, we could prepare 13 models in parallel, which converts into a 7x speedupcompared to training the models each one in turn on one machine. Here is a chart of the computation times (in seconds) with respect of machines on the cluster.

REFERENCES

[1] From Gantz et al., “The Diverse and Exploding Digital Universe, ”March, 2008, (http: //www .emc.com /collateral/analyst-reports/diverse-exploding-digital universe. pdfn)

[2] http://www.guru99.com/what-is-big-data.html#1

[3] D. W. Bates, S. Saria, L. Ohno-Machado, A. Shah, and G. Escobar, "Big data in health care: using analytics to identify and manage high-risk and high-cost patients," Health Affairs, vol. 33, no. 7, pp. 1123–1131, 2014.

[4] Min chen,yixue hao ,"Disease Prediction by Machine Learning over Big Data from Healthcare Communities" IEEE communications,vol 5,pp. 8869 – 8879,2017.

[5] Junfei Qiu,Qihui Wu,Guoru Ding,Yuhua Xu,Shuo Feng, "A survey of machine learning for big data processing",springer ,2016.

[6] Anand Gupta ; Hardeo Kumar Thakur ; Ritvik Shrivastava ; Pulkit Kumar ; Sreyashi Nag "A Big Data Analysis Framework Using Apache Spark and Deep Learning", IEEE international conference,2017.

[7] Jian Fu , Junwei Sun, Kaiyuan Wang, "Spark–a big data processing platform for machine learning" IEEE international conference,2017.

[8] Mllib: Machine learning in apache spark, Journal of Machine Learning Research,

[9] Madalina Cosmina Popescu, Lucian Mircea Sasu, "Feature extraction, feature selection and machine learning for image classification" IEEE international conference,2014.

[10] 10. J. Andreu-Perez, C. C. Poon, R. D. Merrifield, S. T. Wong, and G.-Z. Yang, "Big data for health," Biomedical and Health Informatics, IEEE Journal of, vol. 19, no. 4, pp. 1193–1208, 2015.

[11] 11. S. Sarraf, C. Saverino, H. Ghaderi, and J. Anderson, "Brain network extraction from probabilistic ica using functional magnetic resonance images and advanced template matching techniques," in Electrical and Computer Engineering (CCECE), 2014 IEEE 27th Canadian Conference on, pp. 1–6, IEEE, 2014.

[12] S. M. Smith, "Fast robust automated brain extraction," Human brain mapping, vol. 17, no. 3, pp. 143–155, 2002.

[13] Konstantinos Kamnitsas, Christian Ledig, Virginia FJ Newcombe, Joanna P Simpson, Andrew D Kane, David K Menon, Daniel Rueckert, and Ben Glocker. Efficient multi-scale 3d cnn with fully connected crf for accurate brain lesion segmentation. Medical Image Analysis, 36:61–78, 2017.

[14] C. T. R. Kathirvel. Classifying Diabetic Retinopathy using Deep Learn- ing Architecture. International Journal of Engineering Research Tech- nology, 5(6), 2016.

[15] Fabian Pedregosa, Gael Varoquaux, Alexandre Gramfort, et al. Scikit-learn: Machine learning in Python. Journal of Machine Learning Research, 2011.

[16] D. Oliver, F. Daly, F. C. Martin, and M. E. McMurdo, "Risk factors and risk assessment tools for falls in hospital in-patients: a systematic review," Age and ageing, vol. 33, no. 2, pp. 122–130, 2004.

[17] S. Marcoon, A. M. Chang, B. Lee, R. Salhi, and J. E. Hollander, "Heart score to further risk stratify patients with low timi scores," Critical pathways in cardiology, vol. 12, no. 1, pp. 1–5, 2013

[18] M. Chen, Y. Ma, Y. Li, D. Wu, Y. Zhang, C. Youn, "Wearable 2.0: Enable Human-Cloud Integration in Next Generation Healthcare System," IEEE Communications, Vol. 55, No. 1, pp. 54–61, Jan. 2017.

[19] M. Chen, Y. Ma, J. Song, C. Lai, B. Hu, "Smart Clothing: Connecting Human with Clouds and Big Data for Sustainable Health Monitoring," ACM/Springer Mobile Networks and Applications, Vol. 21, No. 5, pp. 825C845, 2016.

[20] M. Viceconti, P. Hunter, and R. Hose, "Big data, big knowledge: Big data for personalized healthcare," Biomedical and Health Informatics, IEEE Journal of, vol. 19, no. 4, pp. 1209–1215, 2015.

[21] D. Tian, J. Zhou, Y. Wang, Y. Lu, H. Xia, and Z. Yi, "A dynamic and self-adaptive network selection method for multimode communications in heterogeneous vehicular telematics," IEEE Transactions on Intelligent Transportation Systems, vol. 16, no. 6, pp. 3033–3049, 2015.

[22] J. Wan, S. Tang, D. Li, S. Wang, C. Liu, H. Abbas and A. Vasilakos, "A Manufacturing Big Data Solution for Active Preventive Mainte-nance", IEEE Transactions on Industrial Informatics, DOI: 10.1109/TI-I.2017.2670505, 2017.