IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

IMAGE CLASSIFICATION WITH BIG DATA AND DISTRIBUTED DEEP LEARNING

Krishna Shasank Turaga[1], Dr. Ravi Shankar [2]

[1] Research Scholar, Department of Computer Science and Engineering, GITAM University, Hyderabad, Telangana, India

[2] Professor, Department of Computer Science and Engineering, GITAM University, Hyderabad, Telangana, India

ABSTRACT

Image Classification is one of the most comprehensively used algorithms in the area of Artificial Intelligence. In recent times, Convolutional Neural Networks (CNN) has been the strongest advocates for Deep Learning. Many challenges may be overcome, and inspiring results are often obtained by combining Big Data technologies with deep learning. The analysts can classify images with high efficiency with the blend of the Apache Spark and Deep Learning technologies and build models that can be run on K8/cloud/on-premise clusters. Most AI projects start with a Python notebook running on one laptop however, one usually must bear a mountain of pains to scale it to handle larger data set during a distributed fashion. We have built a unified system that helps us in running big data analytics and deep learning workloads on the same cluster. In practice, we discover that keeping one big data cluster for the whole pipeline is more efficient and cost-effective. the target of our paper is to develop a distributed data multiprocessing pipeline, a critical component for large-scale Big Data AI applications by building a Convolutional Neural Network and training it in an exceedingly structured way employing a standard single node python program which might easily be scaled out and deployed on a Kubernetes Cluster or Spark based big data cluster to be run in a very distributed fashion. We write Python code and Test if our architecture is capable of classifying images and categorize them with high precision & accuracy and build models that can be run on large computing clusters. We used the Fashion-MNIST dataset in this experiment. The Fashion-MNIST is a Zalando's article images-based version of the popular MNIST handwritten digit database, with clothes rather than numbers. Image classification techniques cab be used by e-commerce businesses to straighten out many problems such as clothes search, recognition of clothing and fashion recommendation. With its performance obtained by our model, we can claim that our architecture is capable classifying images and categorize them with high precision & accuracy and experimental results show that our model achieved accuracy over 84%.

Key words: Big Data, Deep Learning, Apache Spark, convolutional neural network (CNN), Image Classification dataset.

1. INTRODUCTION

Many challenges can be overcome and encouraging results can be obtained by combining Big Data technologies with deep learning. The combination of the two technologies Apache Spark and Deep Learning enables data analysts to classify images with high efficiency and create models that can run on clusters [1]. The objective of our paper is to develop a distributed data-parallel processing pipeline, a critical component for large-scale AI applications using BigDL (A distributed deep learning library for Apache Spark) and Analytics Zoo which is a open source Big Data AI platform, and includes the features for seamlessly scale out TensorFlow and PyTorch for end-to-end AI to distributed Big Data. Majority of the Deep Learning projects start with a Python notebook running on a single laptop but scaling it up is characterized by hardships that one usually needs to go through to handle larger data set in a distributed fashion. The Analytics Zoo with orca libraries seamlessly scales out your single node TensorFlow or PyTorch notebook across large clusters so as to process distributed Big Data [2] [3]. We build a unified system that allows us to run big data analytics and deep learning workloads on the same cluster. In practice, we find that keeping a single big data cluster for the entire pipeline is more efficient and cost-effective. This architecture will be able to classify Fashion-MNIST images obtained from Zalando's. This model will be an intelligent platform dedicated to assisting us in making better decisions about Image Classification in order to use its advanced techniques as a tool in a distributed fashion.

2. RELATED WORK

2.1. Big Data

Big Data is a term that refers to the Big data is a catch-all term for the unconventional strategies and technologies required to collect, organize, and process insights from massive datasets [4]. There are numerous definitions of Big Data, and it can be difficult to agree on a single definition; each theme focuses on a different aspect of this concept. One of the World's Top SAS Company in business analytics software defines Big Data as [5]: "Big data is a term that describes the large volume of data including semi-structured, unstructured and structured that overwhelms a business on a daily basis. But it’s not the huge data volume that is important but what matters to the organizations is how it is used. Big data needs to be analyzed for an accurate and deep understanding that can lead to better decisions based on evidence and calculated hunches". Multiple platforms for the treatment of Big Data have been built for Multiple purposes. The Apache Spark framework is the most well-known platform for Big Data analytics [6]. Spark has the ability to process data with a variety of structures. It is extremely fast, supports a variety of programming languages, combines machine learning functionality, and integrates with a variety of platforms.

2.2. Big Data Analytics

Big data analytics is analyzing huge volumes of data which includes semi-structured, unstructured and structured data usually from multiple sources and having a contrast of size with the use of advanced analytic techniques. Big data Analysis allows analysts, organizations, business, and researchers to obtain accurate and quick decisions. Advanced analytics techniques such as forecasting, pattern matching, machine learning, deep learning, graph analysis, etc., in the context of the big data, to pick up new insights, new approaches, and new ideas [7]. The aim is to uncover patterns and connections that can be hidden, and that enhance valuable insights about the users who created it. For example, the health care sector produces valuable insights from data to make informed decisions, improve diagnosis & treatments and help develop affordable quality of care and to present best results [8].

2.3. Apache Spark

Apache Spark is an open-source data-processing framework for executing data engineering, deep learning and machine learning and expressed by speed, ease of use, and advanced analytics. Spark runs in-memory on single-node machines and/or multi node clusters and has lightning-fast performance [9]. Spark can be run standalone or on the Cloud Platform (AWS/GCP/Azure etc), or on a top of Hadoop YARN or Mesos, where it can read data directly from HDFS, Mongo, Cassandra, Hive, Hbase. Along with in-memory processing, machine learning, deep learning and graph processing it can also handle streaming [10]. Spark uses the data structure called Resilient Distributed Datasets (RDDs) to store data in-memory. Data can be read from the disk and can be written in the RDDs to create a job with the iterative operation by users and they can also execute several queries on the same subset of data uninterrupted by keeping it in-memory with interactive mode [11]. Spark offers incredible speed advantages with batch processing trading off high memory usage. Spark Streaming is a scalable fault-tolerant stream processing solution for workloads that value throughput over latency [12].

2.4. Deep Learning

Deep learning or deep structured learning represents a class of advanced machine learning techniques. It relies on algorithms using mathematical operations based essentially on Artificial Neural Networks [13]. Deep Learning uses many hidden layers of extracting and transforming features. Each layer takes as input the output of the previous one, they accept data to be processed and they intend to deliver the result of the calculation [14]. The machine becomes capable of learning without explicitly needing to be programmed with deep learning techniques. Deep Learning was applied to multiple problems for example in automatic speech recognition, image recognition, natural language processing, drug discovery and toxicology, customer relationship management, recommendation systems, and bioinformatics [15].

Conventional approaches to build such a pipeline would normally set up two separate clusters, one dedicated to big data processing, and the other dedicated to deep learning training (e.g., a GPU cluster). Unfortunately, this not only introduces a lot of overhead for data transfer, but also requires extra efforts for managing separate workflows and systems in production. While popular deep learning frameworks [16,17,18] and Horovod [19] from Uber provide support for data parallel distributed training (using either parameter server architecture [20] or MPI [21] based AllReduce), it can be very difficult to correctly set them up in production. To Illustrate, all relevant Python packages should be pre-installed on master node and also on every node and the master node has SSH permission to all the other nodes, which in all probability infeasible for the production environment and inconvenient for cluster management. To address these challenges, we propose and implement a unified system using Orca library which seamlessly scales out your single node Python notebook across large clusters so as to process distributed Big Data which runs the end-to-end data processing and deep learning training pipeline on the same big data cluster.

2.5. Convolutional Neural Network for Image Classification

Convolutional Neural Network (CNN) named ConvNet is a type of artificial neural network in which the connection between neurons is inspired by the visual cortex of animals. [22]. It is a Deep Learning algorithm that is able to classify input images, Fashion-MNIST images for example, as types of clothing items [20]. ConvNet can itself extract the features, which makes the classification faster, and more accurate. There are four main layers in the CNN: The Convolution layer, Non-Linearity (ReLU) layer, Pooling or Subsampling layer, and the Classification (Fully Connected Layer) [23]. The first one is used to extract features from the input image. The second layer is an element-wise operation, and it is integrated to replace all negative pixel values in the feature map by zero. The third one, the Pooling layer, is applied to decrease the dimensionality of each feature map but to conserve the most significant information. Finally, the last one means that every neuron in the precedent layer is attached to every neuron on the next layer [24].

3. METHODOLOGY

3.1 Fashion MNIST dataset

Fashion-MNIST is a dataset of Zalando's article images with a training and test sets of 60,000 and 10,000 examples respectively which was developed as a drop-in replacement for the MNIST handwritten digits dataset. Each example in this dataset is a 28x28 grayscale image and is associated with a label from 10 classes. We intend Fashion-MNIST to serve as a direct drop-in replacement for the original MNIST dataset for benchmarking machine learning algorithms. It shares the same image size and structure of training and testing splits [25]. torchvision already has the Fashion MNIST dataset. The Fashion MNIST dataset provides a challenge with more complicated features to train on. The Fashion MNIST is only 28x28 px in size, so we don't need a very complicated network. We can just build a simple CNN like this:

We got two convolution layers each one with 5x5 kernels. After every convolution layer we got a max-pooling layer with a stride of 2. Using which we can extract the required features from the images. Then we flatten the tensors and put them into a dense layer, pass through a Multi-Layer Perceptron (MLP) to carry out the task of classification of our 10 categories [26].

3.2. Neural Network architecture

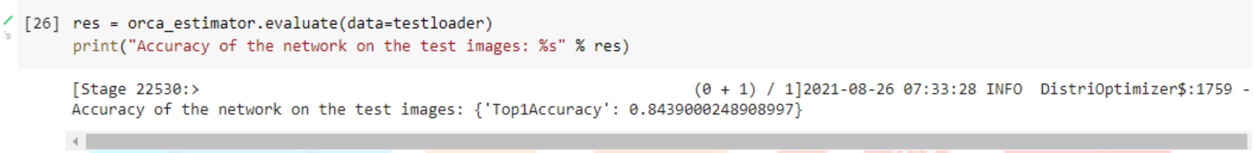
Because the Fashion MNIST is only 28x28 pixels in size, we don't need a complicated network. A convolutional neural network is used to train images (CNN). CNNs are a type of deep layer neural network that is used to learn filters that can be used to extract features when convolved with an image. We begin with two convolution layers, each with five 5x5 kernels, followed by a fully connected layer and a final activation for the final output layer. Following each convolution layer is a max-pooling layer with a stride of 2. As a result, we can extract the necessary features from the images. The tensors are then flattened and placed in a dense layer before being processed by a Multi-Layer Perceptron (MLP) to complete the task of classifying our ten categories. We define the layers by utilizing the nn package's provided modules. We define a Sequential as a sequence of a layer, normalization, activation and pooling. For Example, a CNN layer is defined as nn.Conv2d(in_channels, out_channels, kernel_size, padding, stride). We finish the network with a Fully Connected layer that outputs to 10 classes after the two convolution layers and activations. The forward function on the Neural Network is called for a set of inputs, and it passes that input through the various layers we have defined. To get the output, we pass x through the first layer, its output through the second layer, and that through the final fully connected layer. In the code, the view function reshapes the output to match the dimensions specified for the final layer.

3.3 Apparatus

For our experience we used the Google Colab virtual machine. In this for Environment Preparation we have installed Java 8, we have set up setting up conda environment on Colab and then installed the latest pre-release version of Analytics Zoo.In this platform, Installing Analytics Zoo from pip will automatically install pyspark, bigdl, and their dependencies. This is a collaborative environment that permits users to implement all their analytical processes in a single space and to manage machine learning models throughout their life cycle. We can also create a cluster to execute our model as a set of commands. The following are the versions of the dependencies we have Installed: analytics-zoo-0.12.0b20210816, bigdl-0.13.0, conda-pack-0.3.1, numpy-1.21.2, py4j-0.10.7, pyspark-2.4.6, torch==1.7.1, torchvision==0.8.2 and tensorboard-2.6.0.

4. RESULTS

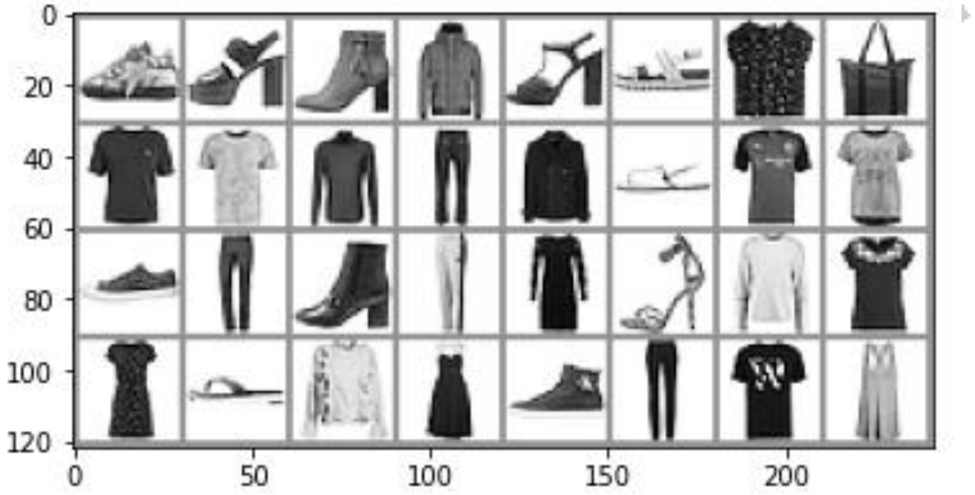
In this study, we implemented a simple convolutional neural network-based model in pytorch trained and tested on Fashion-MNIST images enabling us to classify fashion images and categories. We defined our model, loss and optimizer in the same way as in any standard single node PyTorch program. We also defined the dataloader using standardtorch.utils.data.DataLoader and put it into the dataset. Then we have created an Estimator and set its backend to BigDL. Next fit and evaluate using the Estimator. . The __init__ and forward functions will be implementd. Here we define the layers using the provided modules from the nn package. We define a Sequential as a sequence of a layer, normalization, activation and pooling. For Example, a CNN layer is defined as nn.Conv2d(in_channels, out_channels, kernel_size, padding, stride). We end the network with a Fully Connected layer that outputs to 10 classes after the two convolution layers and activations. We read in the images and labels from the batch, use network class to do the forward propagation and get the predictions. With predictions, we can calculate the loss of this batch using nn.CrossEntropyLoss() function. We reset the gradients after loss calculation is done, so that PyTorch won't accumulate the gradients.We do one back propagation using the loss.backward()method to calculate all the gradients of the weights/biases. Then, to update the weights/biases we can use the optimizer which is defined above. Now we can now calculate the loss and number of correct predictions as the network is updated for the current batch. We have decided to visualize the loss from the training over the number of epochs. We can see the convolutions on the fashion images are able to properly classify its label with high accuracy.



```
[26] res = orca_estimator.evaluate(data=testloader)
     print("Accuracy of the network on the test images: %s" % res)

     [Stage 22530:>                                          (0 + 1) / 1]2021-08-26 07:33:28 INFO  DistriOptimizer$:1759 -
     Accuracy of the network on the test images: {'Top1Accuracy': 0.8439000248908997}
```

Figure 1: The accuracy of this model has reached 84%



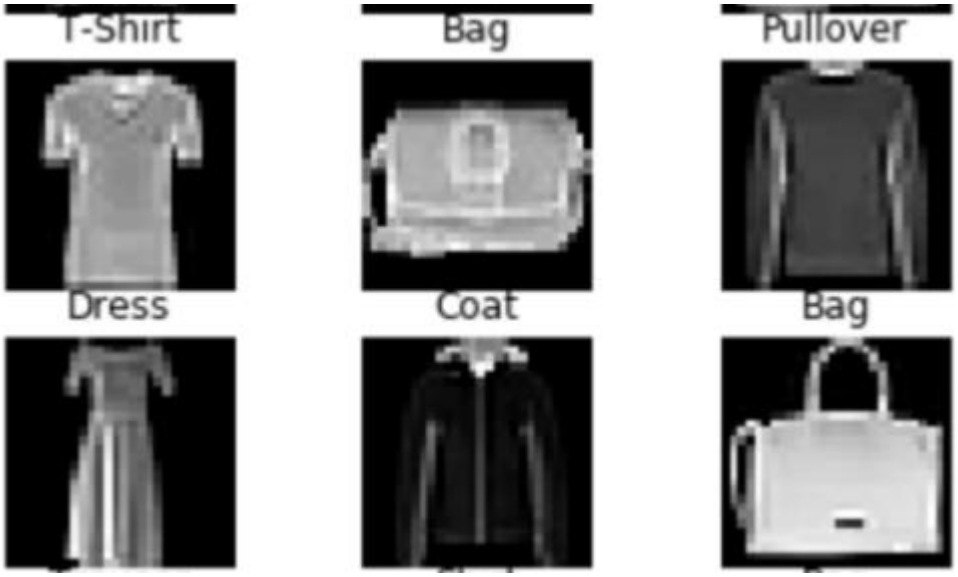
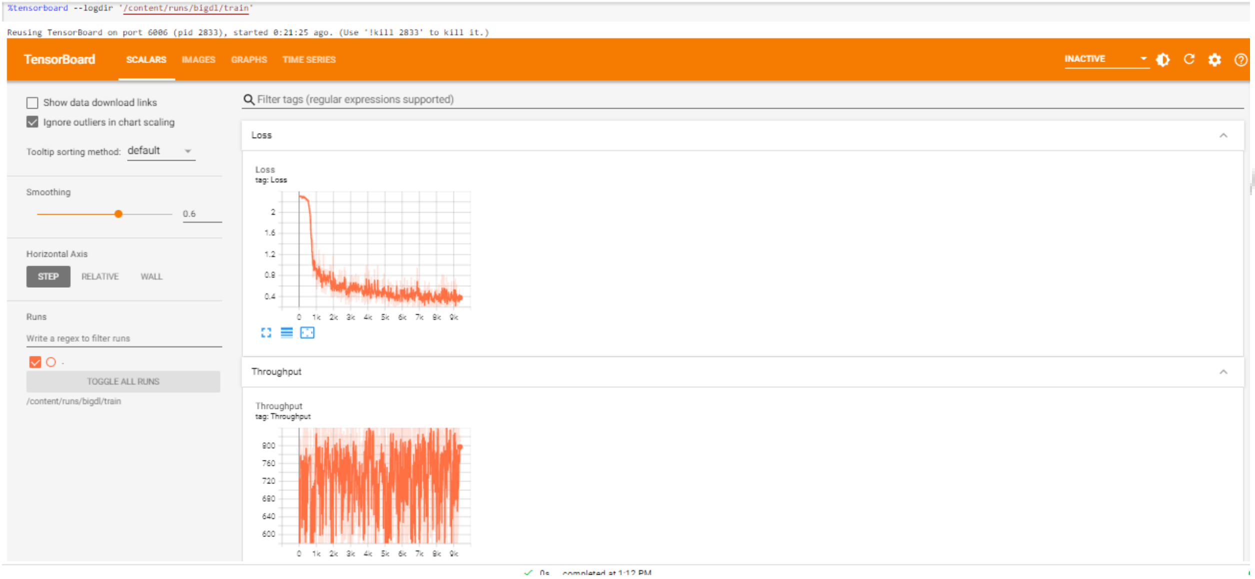


Figure 2: The output of the runs in the notebook along with the Classified Fashion MNIST Images



```
%tensorboard --logdir '/content/runs/bigdl/train'

Reusing TensorBoard on port 6006 (pid 2833), started 0:21:25 ago. (Use '!kill 2833' to kill it.)
```

□ Show data download links

☑ Ignore outliers in chart scaling

Tooltip sorting method: default

Smoothing: 0.6

Horizontal Axis: STEP RELATIVE WALL

Runs

Write a regex to filter runs

☑ ○ .

TOGGLE ALL RUNS

/content/runs/bigdl/train

✓ 0s completed at 1:12 PM

Figure 3: This dashboard shows how the loss and accuracy change with every iteration.

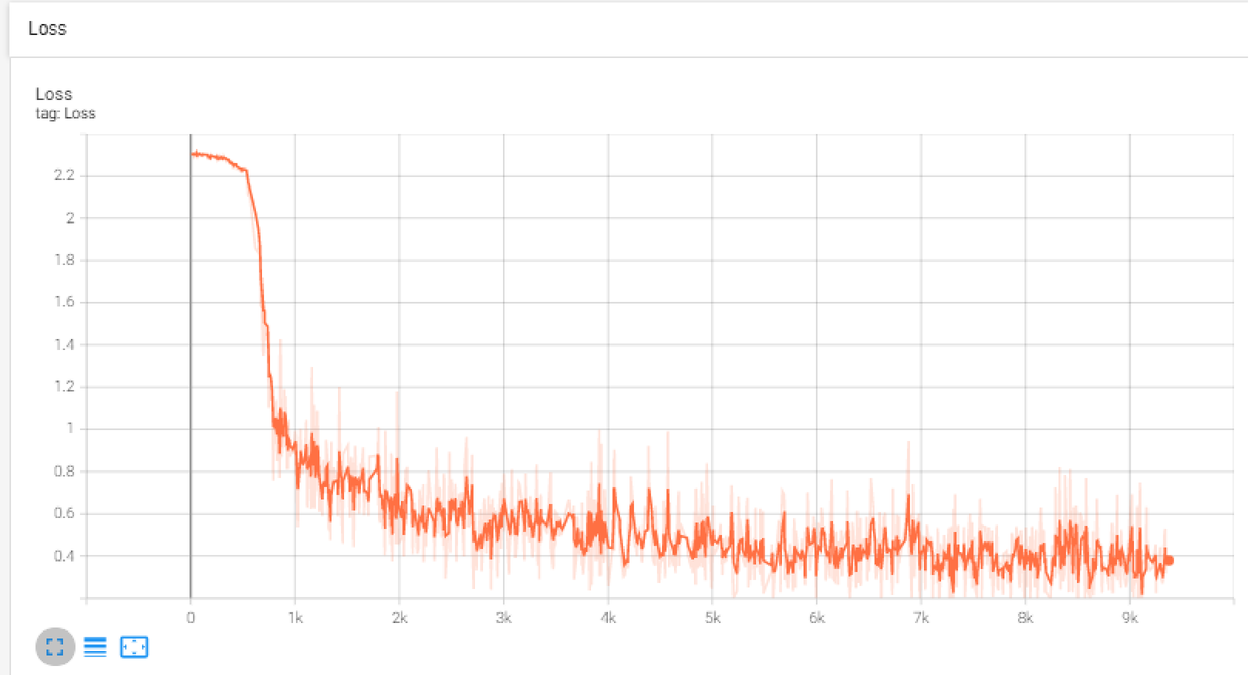


Figure 4: This dashboard shows how the loss change with every iteration.

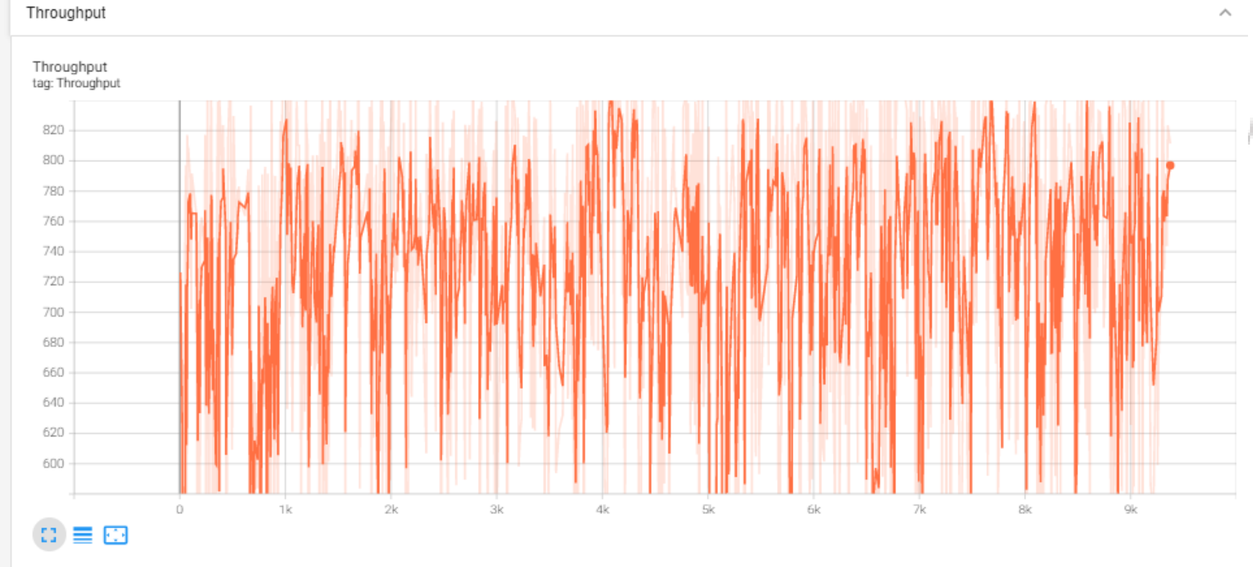


Figure 5: This dashboard shows the throughput changes with every iteration.

5. CONCLUSION

Since the main focus of this study is to showcase how to use PyTorch to build a Convolutional Neural Network and training it in a structured way, we have described how to scale out standard single node PyTorch program using Orca library, Analytics zoo withSpark &BigDL and visualize the results with Tensorboard which can easily be deployed on a Kubernetes Cluster or Spark based big data cluster to be run in a distributed fashion. As we can see, PyTorch as a machine learning framework is flexible, powerful and expressive. With its performance obtained by our model, we can claim that our architecture is capable of classifying fashion images and categories with high accuracy. With the combination of deep learning and Spark technologies a model running on large computing clusters can be built.

REFERENCES

[1] KHUMOYUN A., CUI Y., HANKU L., Spark based distributed deep learning framework for big data applications. In: 2016 International Conference on Information Science and Communications Technologies (ICISCT). IEEE, 2016. pp. 1-5.

[2] J. J. Dai et al., "Bigdl: A distributed deep learning framework for big data," in Proceedings o f the ACM Symposium on Cloud Computing, 2019, pp. 50-60.

[3] J. Dai, Analytics zoo. https://github.com/intelanalytics/analytics-zoo, 2018.

[4] CHEN M., MAO S., LIU, Y., Big data: A survey. Mobile networks and applications, 2014, vol. 19, no 2, pp. 171-209.

[5] SAS Company, Big Data What it is and why it matters, https://www.sas.com/en_us/insights/bigdata/what-is-big-data.html, accessed 16 April 2020.

[6] SHORO A. G., SOOMRO T. R., Big data analysis: Apache spark perspective. Global Journal of Computer Science and Technology, 2015.

[7] SATYANARAYANA L. V., A Survey on challenges and advantages in big data. IJCST, 2015, vol. 6, no 2, pp. 115-119.

[8] RAGHUPATHI W., RAGHUPATHI V., Big data analytics in healthcare: promise and potential. Health information science and systems, 2014, vol. 2, no 1, pp. 3.

[9] WANG K., KHAN M. M. H., Performance prediction for apache spark platform. In: 2015 IEEE 17th International Conference on High Performance Computing and Communications, 2015 IEEE 7th International Symposium on Cyberspace Safety and Security, and 2015 IEEE 12th International Conference on Embedded Software and Systems. IEEE, 2015. pp. 166-173.

[10] FRAMPTON M., Mastering apache spark. Packt Publishing Ltd, 2015.

[11] ZAHARIA M., CHOWDHURY M., DAS T. et al., Resilient distributed datasets: A fault-tolerant abstraction for in-memory cluster computing. In: Presented as part of the 9th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 12). 2012. pp. 15-28.

[12] MARCU O. C., COSTAN A., ANTONIU G., PÉREZ-HERNÁNDEZ, M. S., Spark versus flink: Understanding performance in big data analytics frameworks. In: 2016 IEEE International Conference on Cluster Computing (CLUSTER). IEEE, 2016. pp. 433-442.

[13] BENGIO Y., COURVILLE A., VINCENT P., Representation learning: A review and new perspectives. IEEE transactions on pattern analysis and machine intelligence, 2013, vol. 35, no 8, pp. 1798-1828.

[14] LEE J. G., JUN S., CHO Y. W., LEE H., KIM G. B., SEO J. B., KIM, N., Deep learning in medical imaging: general overview. Korean journal of radiology, 2017, vol. 18, no 4, pp. 570-584.

[15] Hordri, N. F., Yuhaniz, S. S., & Shamsuddin, S. M., Deep learning and its applications: a review. In: Conference on Postgraduate Annual Research on Informatics Seminar, 2016.

[16] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, et al., Tensorflow: A system for largescale machine learning, in 12th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 16), 2016, pp. 265–283.

[17] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, Automatic differentiation n pytorch, NIPS 2017 Autodiff Workshop, (2017).

[18] T. Chen, M. Li, Y. Li, M. Lin, N. Wang, M. Wang, T. Xiao, B. Xu, C. Zhang, and Z. Zhang, Mxnet: A flexible and efficient machine learning library for heterogeneous distributed systems, arXiv preprint arXiv:1512.01274, (2015).

[19] A. Sergeev and M. Del Balso, Horovod: fast and easy distributed deep learning in tensorflow, arXiv preprint arXiv:1802.05799, (2018).

[20] M. Li, D. G. Andersen, J. W. Park, A. J. Smola, A. Ahmed, V. Josifovski, J. Long, E. J. Shekita, and B.-Y. Su, Scaling distributed machine learning with the parameter server, in 11th {USENIX} Symposium on Operating Systems Design and Implementation ({OSDI} 14), 2014, pp. 583–598.

[21] R. L. Graham, G. M. Shipman, B. W. Barrett, R. H. Castain, G. Bosilca, and A. Lumsdaine, Open mpi: A high-performance, heterogeneous mpi, in 2006 IEEE International Conference on Cluster Computing, IEEE, 2006, pp. 1–9.

[22] KIM, Y., Convolutional neural networks for sentence classification. arXiv preprint arXiv:1408.5882. 2014.

[23] HIDAKA, A., KURITA, T. Consecutive dimensionality reduction by canonical correlation analysis for visualization of convolutional neural networks. In: Proceedings of the ISCIE International Symposium on Stochastic Systems Theory and its Applications. The ISCIE Symposium on Stochastic Systems Theory and Its Applications, 2017, pp. 160-167.

[24] YAMASHITA R., NISHIO M., DO, R. K. G., TOGASHI, K., Convolutional neural networks: an overview and application in radiology. Insights into imaging, 2018, vol. 9, no 4, pp. 611-629.

[25] Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms. Han Xiao, Kashif Rasul, Roland Vollgraf. arXiv:1708.07747

[26] Michael Li, Let's Build a Fashion-MNIST CNN, PyTorch Style https://towardsdatascience.com/build-a-fashion-mnist-cnn-pytorch-style-efb297e22582, 2019.

[27] Abadi, Mart&#x27;in, Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., … others. (2016). Tensorflow: A system for large-scale machine learning.