Distributed Deep Learning Framework for Big Data

Prasoon Majumdar

Contents

- Research Area
- Problem Statement
- Objective
- BigDL Framework
- Execution Model
- Experiments
- Applications
- Related Work
- Summary & Conclusion

Research Area

- Applying deep learning to production big data pipelines
- Integrating deep learning with big data analytics workflows

Problem Statement

- Deep learning solutions separate from big data platforms
- Causes inefficiencies in data transfer, development, deployment
- Impedance mismatch between deep learning and big data systems

Objective

- Develop unified framework for distributed deep learning and big data analytics
- Enable efficient end-to-end data pipelines for production data

Methodology - Distributed Deep Learning on Spark

- Open source library on top of Apache Spark
- Unified API for deep learning model development
- Large-scale distributed training and inference on Spark clusters

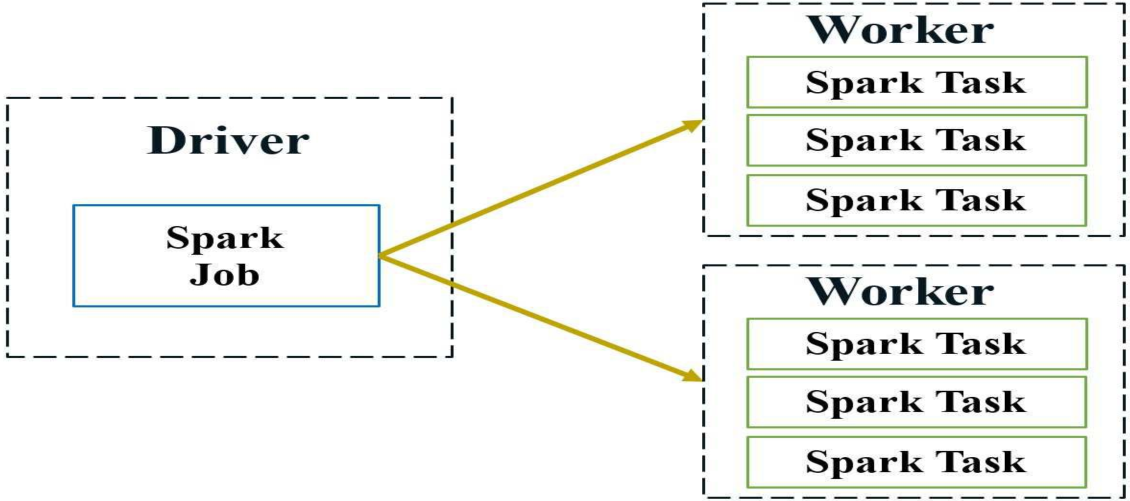
End to End classification pipeline on spark-bigDL

```
#distributed data processing
spark = SparkContext(appName="text_classifier", ...)
input_rdd = spark.textFile("hdfs://...")
train_rdd = input_rdd.map(lambda x: read_text_and_label(x))
                     .map(lambda data: decode_to_ndarrays(data))
                     .map(lambda arrays: to_sample(arrays))

#distributed training
model = Sequential().add(Recurrent().add(LSTM(...)))
                    .add(Linear(...)).add(LogSoftMax())
optimizer = Optimizer(model=model, training_rdd=train_rdd,
                      criterion=ClassNLLCriterion(),
                      optim_method=Adagrad(), ...)
trained_model = optimizer.optimize()

#distributed inference
test_rdd = ...
prediction_rdd = trained_model.predict(test_rdd)
```

Execution Model - Spark



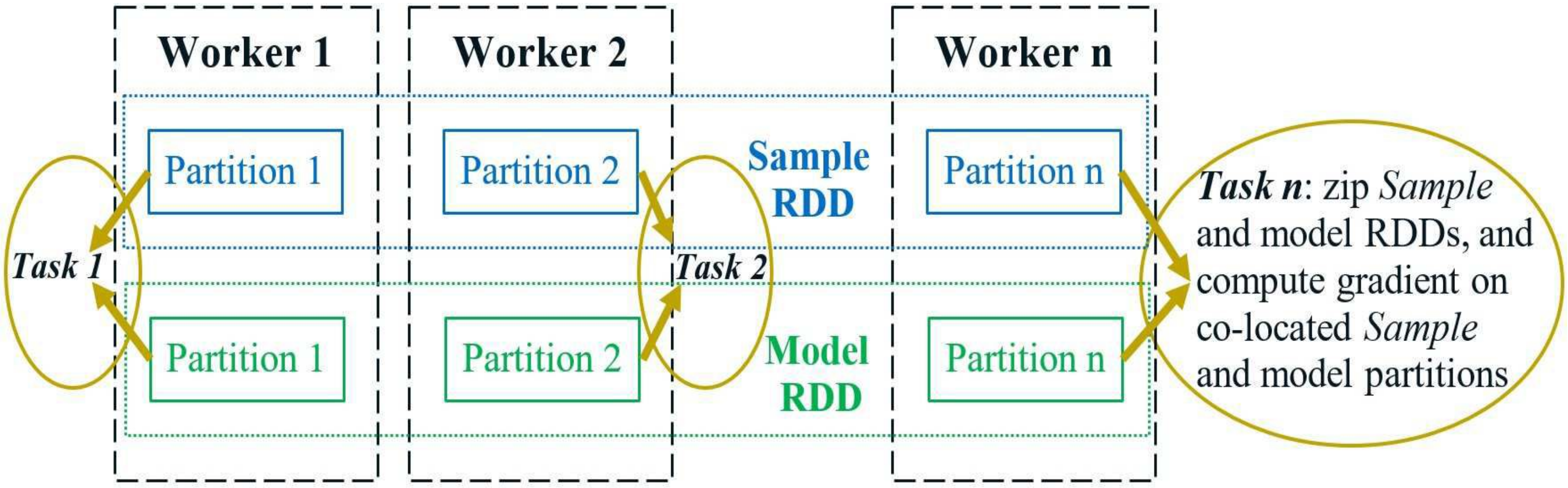
BigDL algo in the works (pseudocode)

Algorithm 1 Data-parallel training in BigDL
1: for i = 1 to M do
2: //"model forward-backward" job
3: for each task in the Spark job do
4: read the latest weights;
5: get a random batch of data from local Sample partition;
6: compute local gradients (forward-backward on local model replica);
7: end for
8: //"parameter synchronization" job
9: aggregate (sum) all the gradients;
10: update the weights per specified optimization method;
11: end for

Data Parallel Training in Big DL

- Iterative process of compute gradients and update parameters
- Co-partitioned and co-located RDDs for models and data
- Parallel gradient computation using Spark tasks

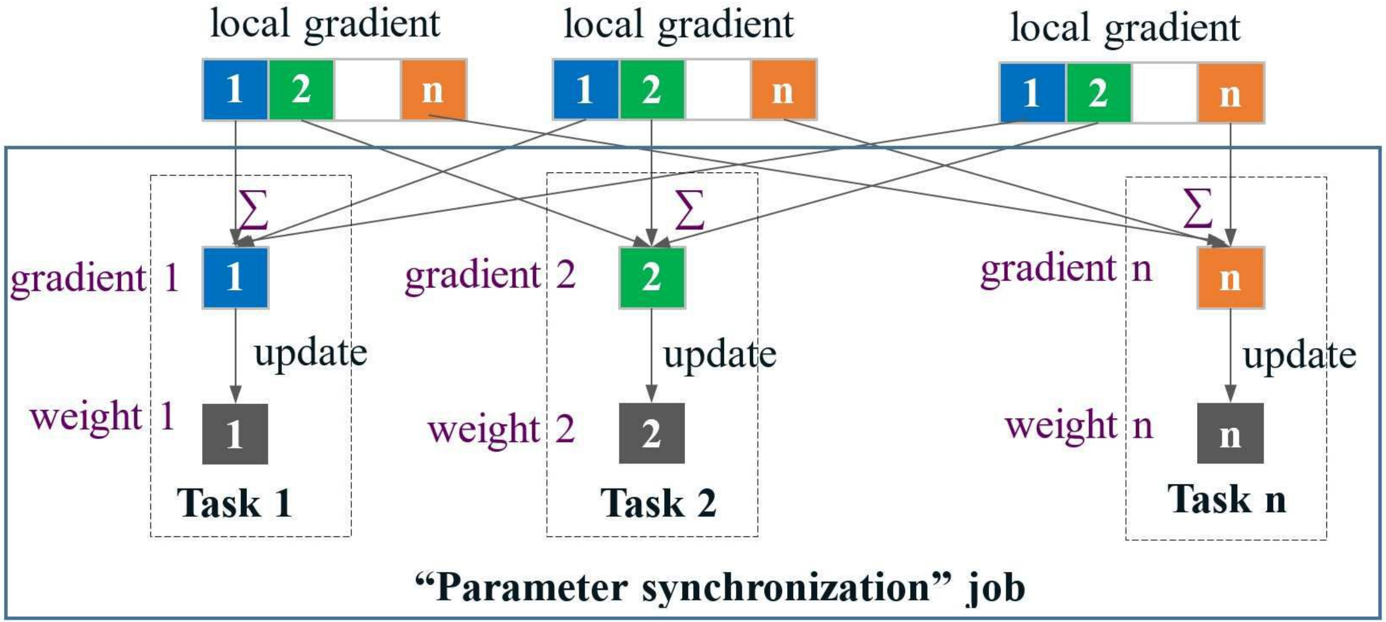
Forward Backward Model for Gradient Computation



Parameter Sync

- Implements AllReduce using Spark primitives (shuffle, broadcast)
- Mimic parameter server architecture in a different way

Parameter Sync Job



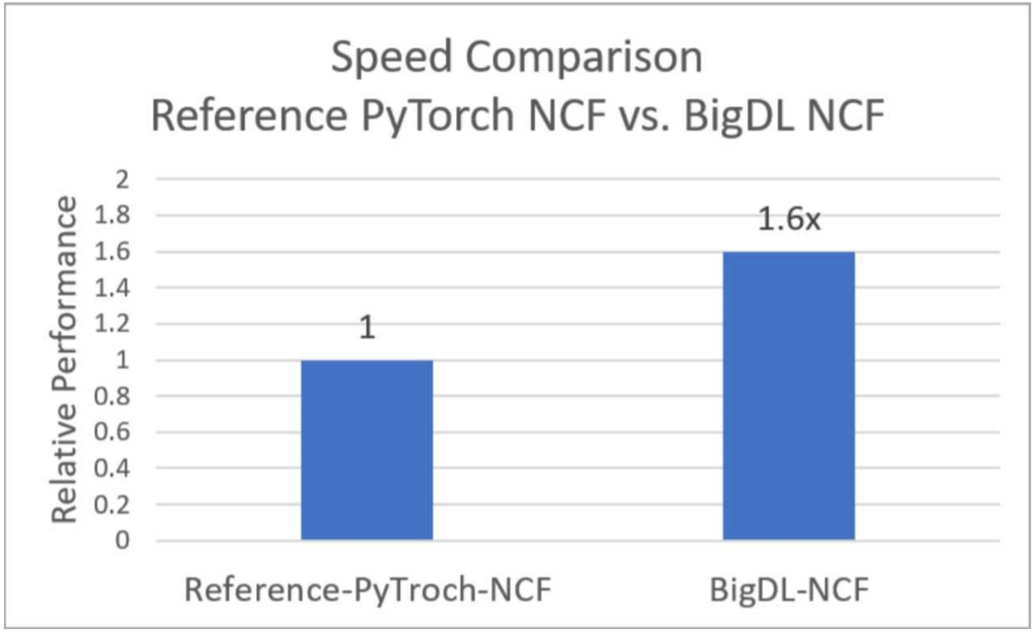
Discussions

- Alternative design for distributed training
- Handles failures, resource changes efficiently
- Logically centralized control with stateless tasks

Experiment Setup

- Neural Collaborative Filtering (NCF)
- Convolutional Neural Networks (CNNs) - Inception-v1 on ImageNet

NCF Performance



NCF Training Performance

- 1.6x faster than PyTorch implementation on Nvidia P100 GPU
- Evaluated on single Intel Xeon server

Inception Scalability

Scalability of Inception-v1 Training

- Efficient parameter sync overheads (<7% on 32 nodes)
- Near linear scaling up to 96 nodes, reasonable up to 256 nodes

Efficiency of Task Scheduling

- Launching large number of short tasks can be a bottleneck
- Group scheduling in Drizzle reduces overheads significantly

Drizzle , a
low latency execution engine for Spark, can help schedule multiple iterations (or a group) of computations at once, so as to greatly reduce scheduling overheads even if there are a large number of tasks in each iteration

BigDL with Drizzle lib

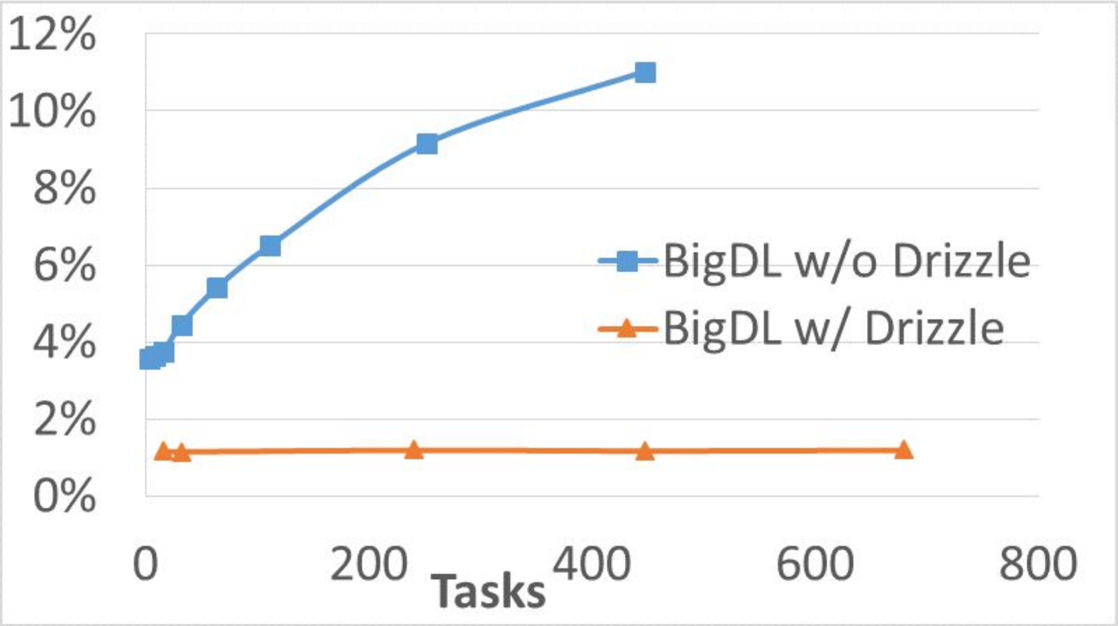
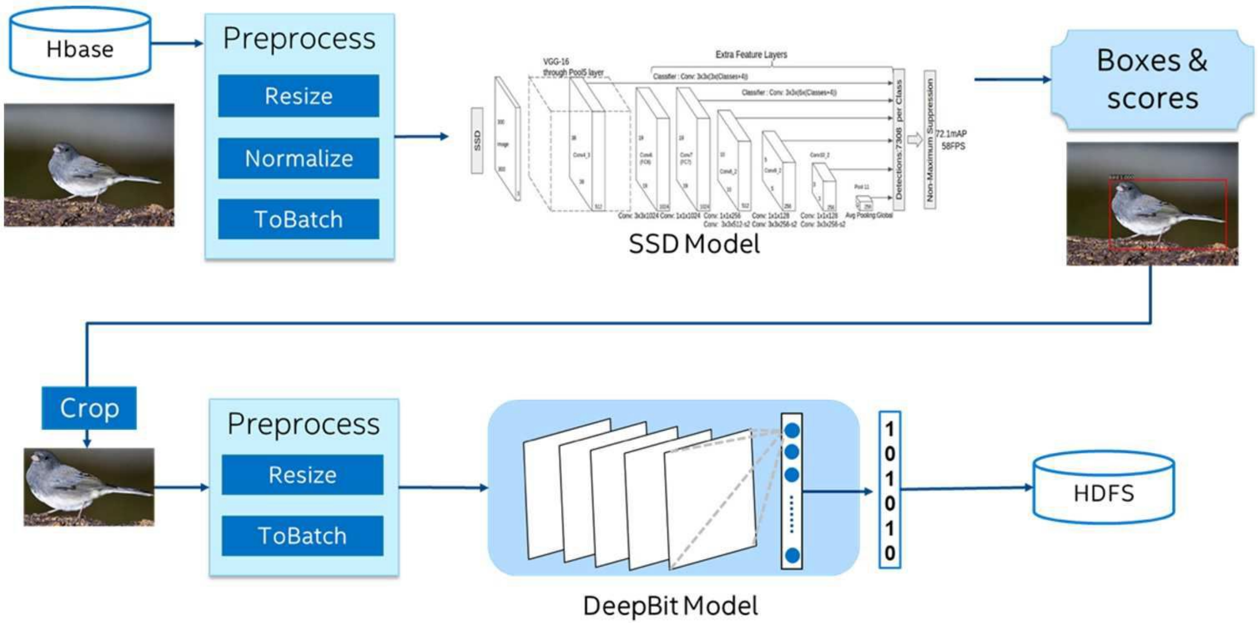


Image Feature Extraction Steps

- End-to-end pipeline for object detection and feature extraction
- Using SSD and DeepBit models on Spark and BigDL

SSD and DeepBit Models



Pipeline Steps

- Data loading and preprocessing on Spark
- Distributed object detection with SSD
- Distributed feature extraction with DeepBit
- Store features in HDFS

End to End Flow at JD.com using BigDL

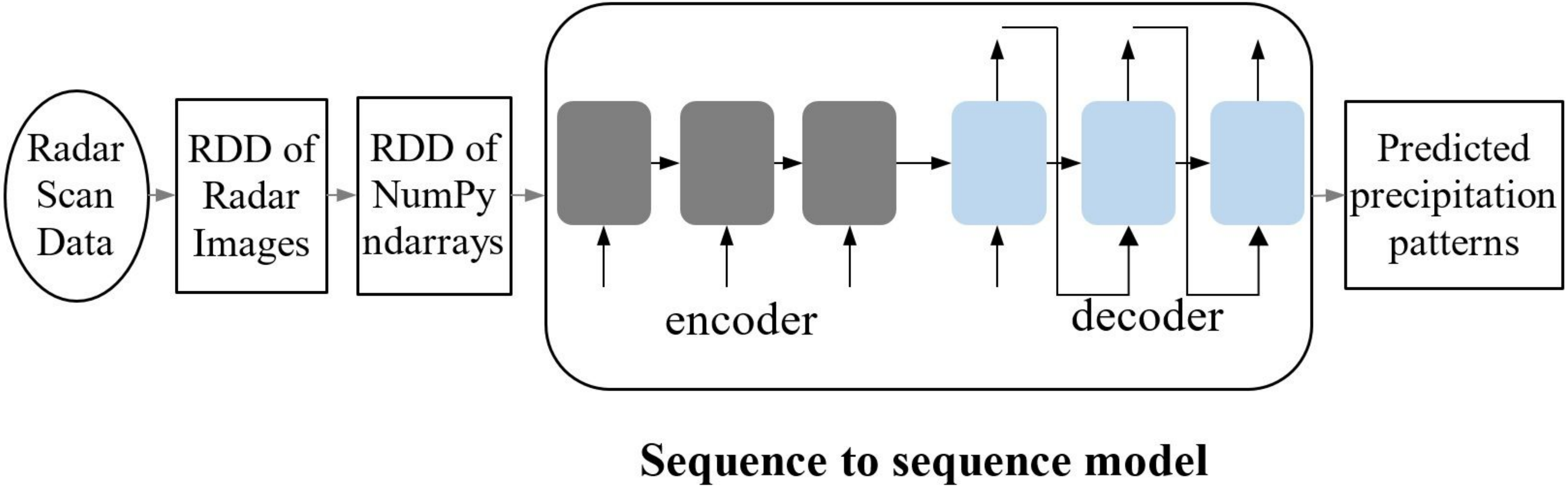


Image Feature Extraction built at JD.com

1. Data Retrieval and Pre-processing

Hundreds of millions of images are retrieved from a distributed database into Spark and pre-processed in a distributed manner.

2. Object Detection

Utilizing BigDL, a pre-trained SSD model is loaded for distributed object detection on Spark, producing coordinates and scores for detected objects.

Extraction cont..

3. Target Image Generation
   Target images are generated by selecting the object with the highest score and cropping the original picture based on its coordinates.

4. Feature Extraction
   Another pre-trained model from Caffe, DeepBit, is employed via BigDL for distributed feature extraction of the target images, storing results in HDFS.

More Applications and use-cases

Precipitation Nowcasting at Cray

- Sequence-to-sequence model for precipitation prediction
- End-to-end data preparation, training, inference on BigDL

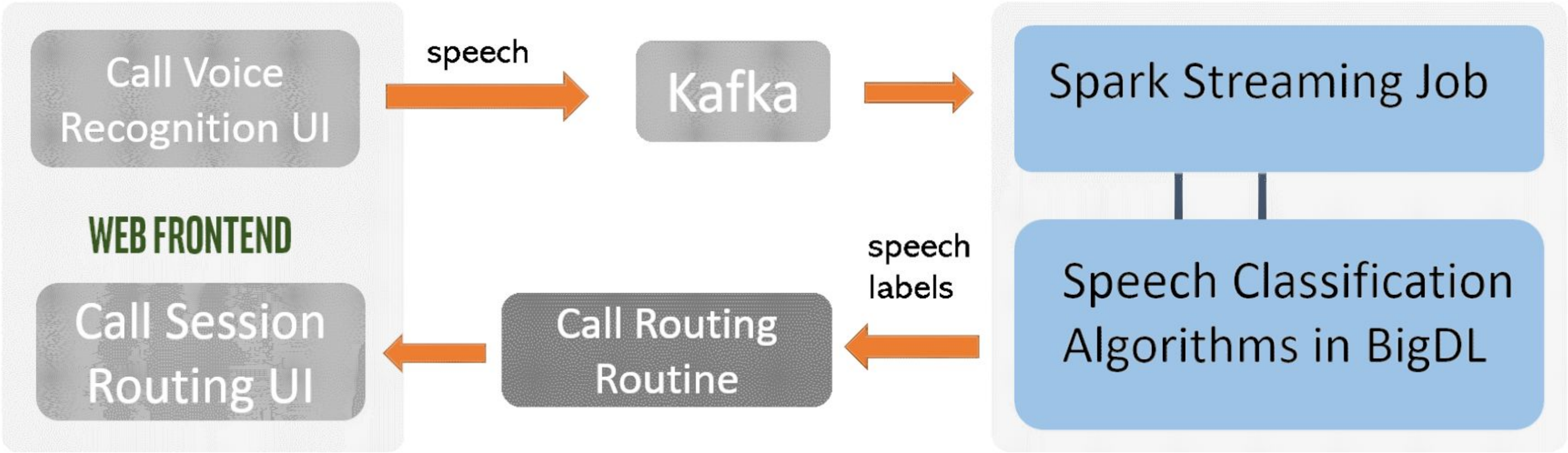
Precipitation Nowcasting Workflow (Cray application)

- Ingest radar scan data into Spark
- Train seq2seq model on historical data
- Predict future precipitation patterns

Real-time Streaming Workflow (Giga spaces application)

- Speech recognition -> Kafka
- Spark Streaming job with BigDL model
- Classify call and route to specialist

Real time streaming speech classification on BigDL



How its done

- When a customer calls the call center, his or her speech is first processed on the fly by a speech recognition unit and result is stored in Kafka.
- • A Spark Streaming job then reads speech recognition results from Kafka and classifies each call using the BigDL model in real-time.
- • The classification result is in turn used by a routing system to redirect the call to the proper support specialist.

Related Wok: Existing DL frameworks

- TensorFlow, MXNet, Petuum, ChainerMN
- Fine-grained data access, in-place mutation
- Different execution model than BigDL

Summary

- BigDL provides alternative distributed training design
- Seamless integration of DL and data analytics
- Adopted by users across industries

Future To-Dos

- Improve training performance and scalability
- Extend to more DL models and applications
- Enhance support for real-time streaming

References

[1] Jia, Yangqing and Shelhamer, Evan and Donahue, Jeff and Karayev, Sergey and Long, Jonathan and Girshick, Ross and Guadarrama, Sergio and Darrell, Trevor. Caffe: Convolutional architecture for fast feature embedding. in Proceedings of the 22nd ACM international conference on Multimedia. MM'14.
[2] Collobert, Ronan and Kavukcuoglu, Koray and Farabet, Cl.ment. Torch7: A matlab-like environment for machine learning. in BigLearn, NIPS workshop. (2011).
[3] Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., Kudlur, M., Levenberg, J., Monga, R., Moore, S., Murray, D. G., Steiner, B., Tucker, P., Vasudevan, V., Warden, P., Wicke, M., Yu, Y., and Zheng, X. Tensorflow: A system for large-scale machine learning. in Proceedings of the 12th USENIX Conference on Operating Systems Design and Implementation. OSDI'16.

References

[4] Chen, Tianqi and Li, Mu and Li, Yutian and Lin, Min and Wang, Naiyan and Wang, Minjie and Xiao, Tianjun and Xu, Bing and Zhang, Chiyuan and Zhang, Zheng. Mxnet: A flexible and efficient machine learning library for heterogeneous distributed systems. In Proceedings ofWorkshop on Machine Learning Systems (LearningSys) in The Twenty-ninth Annual Conference on Neural Information Processing Systems (NIPS). (2015).

[5] Tokui, Seiya and Oono, Kenta and Hido, Shohei and Clayton, Justin Chainer: a next-generation open source framework for deep learning in In Proceedings of workshop on machine learning systems (LearningSys) in the twenty-ninth annual conference on neural information processing systems (NIPS). (2015).

[6] Paszke, Adam and Gross, Sam and Chintala, Soumith and Chanan, Gregory and Yang, Edward and DeVito, Zachary and Lin, Zeming and Desmaison, Alban and Antiga, Luca and Lerer, Adam Automatic differentiation in pytorch. NIPS 2017 Autodiff Workshop. (2017).

[7] Apache spark Apache software foundation. (2014) (https://spark.apache.org).

[8] Apache hadoop Apache software foundation. (2006) (https://hadoop.apache.org).

[9] Chollet,F.et al. Keras. (https://keras.io).

[10] Zaharia, Matei and Chowdhury, Mosharaf and Das, Tathagata and Dave, Ankur and Ma, Justin and McCauley, Murphy and Franklin, Michael J and Shenker, Scott and Stoica, Ion. Resilient distributed datasets: A fault-tolerant abstraction for in-memory cluster computing. in Proceedings of the 9th USENIX conference on Networked Systems Design and Implementation. NSDI'12.

[11] Armbrust, Michael and Xin, Reynold S and Lian, Cheng and Huai, Yin and Liu, Davies and Bradley, Joseph K and Meng, Xiangrui and Kaftan, Tomer and Franklin, Michael J and Ghodsi, Ali and others. Spark sql: Relational data processing in spark. in 2015 ACM SIGMOD international conference on management of data. SIGMOD'15.

[12] Russakovsky,Olga andDeng, Jia and Su,Hao and Krause, Jonathan and Satheesh, Sanjeev and Ma, Sean and Huang, Zhiheng and Karpathy, Andrej and Khosla, Aditya and Bernstein, Michael and others. Imagenet large scale visual recognition challenge. International journal of computer vision(IJCV). (2015).

[13] Rajpurkar,P and Zhang,J and Lopyrev,K and Liang,P. Squad: 100,000+ questions for machine comprehension of text. In Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP. (2016).

[14] Jawaheer,Gand Szomszor,Mand Kostkova,P.Comparison of implicit and explicit feedback froman online music recommendation service. in proceedings of the 1st international workshop on information heterogeneity and fusion in recommender systems. (2010) HetRec'10.

[15] Baylor, D., Breck, E., Cheng, H.-T., Fiedel, N., Foo, C. Y., Haque, Z., Haykal, S., Ispir, M., Jain, V., Koc, L., Koo, C. Y., Lew, L., Mewald, C., Modi, A. N., Polyzotis, N., Ramesh, S., Roy, S., Whang, S. E., Wicke, M., Wilkiewicz, J., Zhang, X., and Zinkevich, M. Tfx: A tensorflow-based production-scale machine learning platform in Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. KDD'17.

[16] CaffeOnSpark. Yahoo. (2016) (https://github.com/yahoo/CaffeOnSpark).

[17] TensorflowOnSpark. Yahoo. (2017) (https://github.com/yahoo/TensorFlowOnSpark).

[18] Sagemaker. Amazon. (2017) (https://aws.amazon.com/sagemaker/).

[19] Lin, Jimmy and Ryaboy, Dmitriy Scaling big data mining infrastructure: the twitter experience. ACM SIGKDD Explorations Newsletter 14(2). (December 2012).

[20] Reynold Xin. "project hydrogen: Unifying state-of-the-art ai and big data in apache spark". spark + ai summit 2018.

[21] Gang scheduling. (https://en.wikipedia.org/wiki/Gang_scheduling/).

[22] Zaharia, Matei and Borthakur, Dhruba and Sen Sarma, Joydeep and Elmeleegy, Khaled and Shenker, Scott and Stoica, Ion. Delay scheduling: a simple technique for achieving locality and fairness in cluster scheduling. in Proceedings of the 5th European conference on Computer systems,. EuroSys'10.

[23] Dean,J., Corrado,G., Monga,R., Chen,K., Devin,M., Mao,M., Ranzato, Marc'aurelio, Senior,A., Tucker,P., Yang,K., Le,Q.V., Ng,A.Y. Large scale distributed deep networks. in Proceedings of the 25th International Conference on Neural Information Processing Systems. NIPS'12.

[24] Li,M., Andersen,D.G., Park,J.W., Smola,A.J., Ahmed,A., Josifovski,V., Long,J., Shekita,E.J., and Su,B.-Y. Scaling distributed machine learning with the parameter server. in Proceedings of the 11th USENIX Conference on Operating Systems Design and Implementation. OSDI'14.

[25] Chilimbi,T., Suzue,Y., Apacible,J., and Kalyanaraman,K. Project adam: Building an efficient and scalable deep learning training system. in Proceedings of the 11th USENIX Conference on Operating Systems Design and Implementation. OSDI'14.

[26] Xing,E.P., Ho,Q., Dai,W., Kim,J.-K.,Wei,J., Lee,S., Zheng,X., Xie,P., Kumar,A., and Yu,Y. Petuum: A new platform for distributed machine learning on big data. Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. KDD'15.

[27] Zhang,H., Zheng,Z., Xu,S., Dai,W., Ho,Q., Liang,X., Hu,Z., Wei,J., Xie,P., and Xing,E.P. Poseidon: An efficient communication architecture for distributed deep learning on gpu clusters. in 2017 USENIX Annual Technical Conference (USENIX ATC 17). (2017).

[28] Jeffrey Dean, Sanjay Ghemawat Mapreduce: simplified data processing on large clusters. Proceedings of the 6th conference on Symposium on Operating Systems

References

Design & Implementation,{OSDI}. (2004).
[29] Michael Isard,Mihai Budiu, Yuan Yu,AndrewBirrell, andDennis Fetterly. Dryad: distributed data-parallel programs from sequential building blocks in Proceedings of the 2nd ACM SIGOPS/EuroSys European Conference on Computer Systems 2007. EuroSys'07.
[30] Chen,J., Monga,R., Bengio,S., and Jozefowicz,R. Revisiting distributed synchronous sgd. In International Conference on Learning Representations Workshop Track. (2016).
[31] Gibiansky,Andrew. "bringing hpc techniques to deep learning". (http://andrew.gibiansky.com/blog/machine-learning/baidu-allreduce/).
[32] Akiba,T., Fukuda,K., and Suzuki,S. Chainermn: Scalable distributed deep learning framework. Proceedings of Workshop on ML Systems in The Thirty-first Annual Conference on Neural Information Processing Systems (NIPS). (2017).
[33] Eric Liang, Richard Liaw, Philipp Moritz, Robert Nishihara, Roy Fox, Ken Goldberg, Joseph E. Gonzalez, Michael I. Jordan, Ion Stoica. Rllib: Abstractions for distributed reinforcement learning. International Conference on Machine Learning (ICML). (2018).
[34] He, Xiangnan and Liao, Lizi and Zhang, Hanwang and Nie, Liqiang and Hu, Xia and Chua, Tat-Seng Neural collaborative filtering. in Proceedings of the 26th international conference on world wide web. InternationalWorldWideWeb Conferences Steering Committee. (2017).
[35] Mlperf. (https://mlperf.org/).
[36] Szegedy,C., Liu,W., Jia,Y., Sermanet,P., Reed,S., Anguelov,D., Erhan,D., Vanhoucke, V., and Rabinovich,A. Going deeper with convolutions in Computer Vision and Pattern Recognition (CVPR). (2015).
[37] Deng,J., Socher,R., Fei-Fei,L., Dong,W., Li,K., and Li,L.-J. Imagenet: A large-scale hierarchical image database. in 2009 IEEE conference on computer vision and pattern recognition(CVPR). (2009).
[38] Szegedy,C., Vanhoucke,V., Ioffe,S., Shlens,J., and Wojna,Z. Rethinking the inception architecture for computer vision in 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2016).
[39] He, Kaiming and Zhang, Xiangyu and Ren, Shaoqing and Sun, JianDeep residual learning for image recognition. in Proceedings of the IEEE conference on computer vision and pattern recognition. (2016).
[40] Reference ncf implementation using pytorch in mlperf. (https://github.com/mlperf/training/blob/ master/recommendation/pytorch/README.md).
[41] Harper, F Maxwell and Konstan, Joseph A. "the movielens datasets: History and context". ACM Trans. Interact. Intell. Syst. 5(4):19. (2015).
[42] Ncf implementation in bigdl. (https://github.com/mlperf/training_results_v0.5/tree /master/v0.5.0/intel/intel_ncf_submission).
[43] Mlperf 0.5 training results. (https://mlperf.org/training-results-0-5).
[44] Jason (Jinquan) Dai, and Ding Ding. Very large-scale distributed deep learning with bigdl. o'reilly ai conference, san francisco. (2017).
[45] Alex Heye, et al. "scalable deep learning with bigdl on the urika-xc software suite". (https://www.cray.com/blog/scalable-deep-learning-bigdl-urika-xcsoftware-suite/).
[46] Shivaram Venkataraman, et al. "accelerating deep learning training with bigdl and drizzle on apache spark". (https://rise.cs.berkeley.edu/blog/acceleratingdeep-learning-training-with-bigdl-and-drizzle-on-apache-spark/).
[47] Venkataraman,S., Panda,A., Ousterhout,K., Armbrust,M., Ghodsi,A., Franklin,M.J., Recht,B., and Stoica,I. Drizzle: Fast and adaptable stream processing at scale in Proceedings of the 26th Symposium on Operating Systems Principles. SOSP'17.
[48] Jason (Jinquan) Dai, et al. Building large-scale image feature extraction with bigdl at jd.com. (https://software.intel.com/en-us/articles/building-large-scaleimage-feature-extraction-with-bigdl-at-jdcom).
[49] Liu,W., Anguelov,D., Erhan,D., Szegedy,C., Reed,S.E., Fu,C.-Y., and Berg,A.C. Ssd: Single shot multibox detector in ECCV. (2016).
[50] Lin, K., Lu, J., Chen, C.-S., and Zhou, J. Learning compact binary descriptorswith unsupervised deep neural networks. in 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). (2016).
[51] Sutskever,I., Vinyals,O., and Le,Q.V. Sequence to sequence learning with neural networks. in Proceedings of the 27th International Conference on Neural Information Processing Systems. Vol. 2. NIPS'14.
[52] Shi,X., Chen,Z., Wang,H., Yeung,D.-Y., Wong,W.-k., and Woo,W.-c. Convolutional lstm network: A machine learning approach for precipitation nowcasting. in Proceedings of the 28th International Conference on Neural Information Processing Systems. Vol. 1. NIPS'15.
[53] Rajiv Shah. Gigaspaces integrates insightedge platformwith intel's bigdl for scalable deep learning innovation. (https://www.gigaspaces.com/blog/gigaspacesto-demo-with-intel-at-strata-data-conference-and-microsoft-ignite/).
[54] Apache Kafka. (https://kafka.apache.org/).
[55] Matei Zaharia, Tathagata Das, Haoyuan Li, Timothy Hunter, Scott Shenker, and Ion Stoica. Discretized streams: fault-tolerant streaming computation at scale in The Twenty-Fourth ACM Symposium on Operating Systems Principles. (2013) SOSP'13.