Use Analytics Zoo to build an intelligent recommendation system on Office Depot

Kai Huang

Mar 26th, 2020

OUTLINE

- Background and use case overview
- Introduction to Analytic Zoo
- Recommenders on Analytics Zoo
- Performance and deployment by Office Depot
- Conclusion

Why Recommendation Systems?

- Help customers choose from a variety of products.
- Maintain user satisfaction and royalty.
- Turn ordinary users into potential customers.
- Increase revenue per user visit.
- ……

Big Data Journey for Recommendation

Stage I :

Office Depot tried to build intelligent models for product recommendation using Python/SAS/R.

Challenges:

They can not process this amount of data on a single machine:

- Over 100,000,000 distinct sessions monthly.
- More than 300,000 active products selling online.
- Training data often exceed 10G.

Big Data Journey for Recommendation

Stage II :

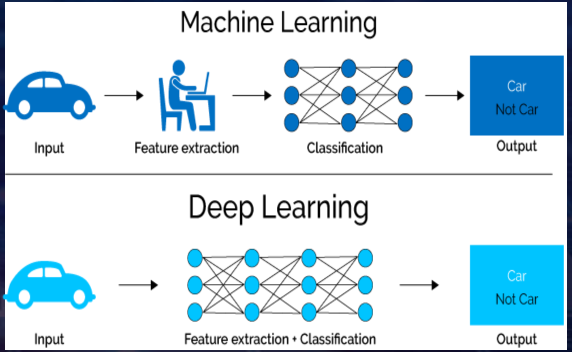
Office Depot incorporated Spark and AWS cloud into their workflow.

Challenge:

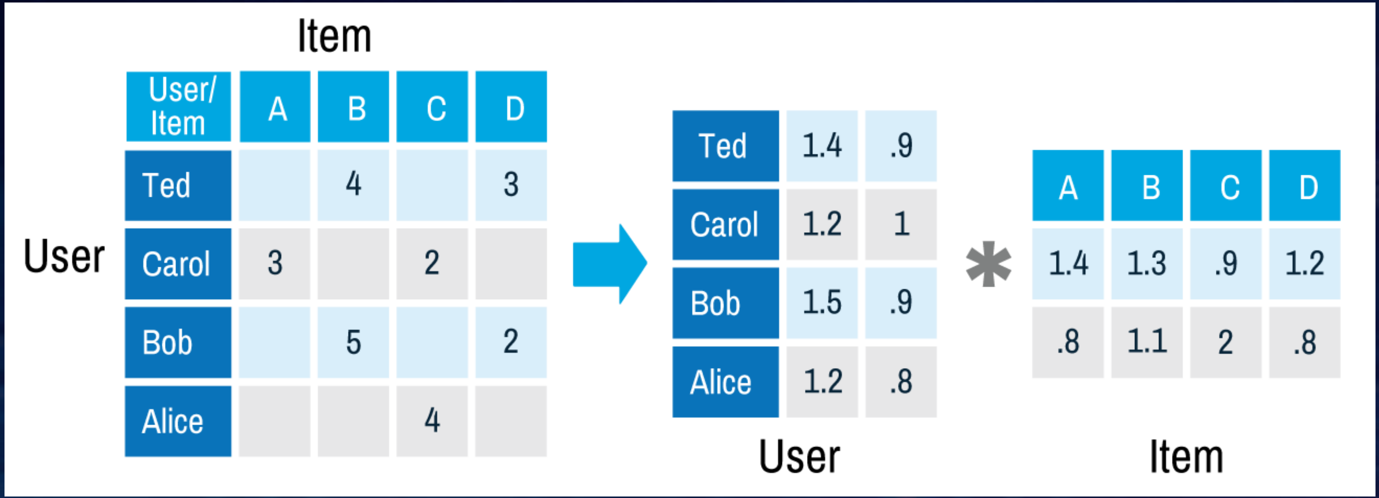
Deep learning libraries such as TensorFlow/Keras/PyTorch cannot run directly on Spark clusters.

Why deep learning?

- Better performance on larger data.
- Less manual feature engineering needed.
- Easier to involve complex functions and combine different architectures.

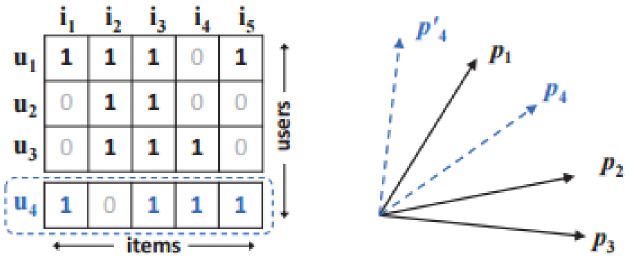


Collaborative Filtering (ALS)



- The Collaborative filtering approach works by decomposing the user-item interaction matrix into the product of two lower dimensionality rectangular matrices.
- Spark ALS (Alternating Least Squares) implementation runs matrix factorization in a parallel fashion and therefore has a pretty good scalability and performance.

Collaborative Filtering (ALS)



(a) user–item matrix (b) user latent space

Figure 1: An example illustrates MF's limitation. From data matrix (a), u_4 is most similar to u_1, followed by u_3, and lastly u_2. However in the latent space (b), placing p_4 closest to p_1 makes p_4 closer to p_2 than p_3, incurring a large ranking loss.

Limitations of matrix factorization:

- Simple choice of the interaction function will hinder the performance.
- Data sparse problement.
- Not able to do incremental training.
- Cold start problem.
- Not able to capture the latest purchase intent.

...

AI ON APACHE Spark™

BigDL

Distributed, High-Performance
Deep Learning Framework
for Apache Spark
https://github.com/intel-analytics/BigDL

ANALYTICS ZOO

A unified analytics and AI platform
for distributed Tensorflow, Keras, PyTorch and Ray
on Apache Spark
https://github.com/intel-analytics/analytics-zoo

Accelerating Data Analytics + AI Solutions At Scale

Analytics Zoo

Unified Big Data Analytics and AI Platform

Models & Algorithms
- Recommendation
- Time Series
- Computer Vision
- NLP

ML Workflow
- AutoML for Time Series
- Automatic Cluster Serving

Integrated Analytics & AI Pipelines
- Distributed TensorFlow & PyTorch on Spark
- RayOnSpark
- Spark Dataframes & ML Pipelines for DL
- Model Serving

Library & Framework
- Distributions (Cloudera/Databricks/....)
- Distributed Analytics (Spark/Flink/Ray/...)
- DL Frameworks (TF/PyTorch/...)
- Python Libraries (Numpy/Pandas/...)

https://github.com/intel-analytics/analytics-zoo

Unified Big Data Analytics and AI Platform

Seamless Scaling from Laptop to Production

- Easily prototype the **integrated data analytics & AI solution**
- **"Zero" code change** from laptop to distributed cluster
- **Directly access production data** (Hadoop/Hive/HBase) without data copy
- Seamlessly deployed on **production big data clusters**

Real-World Applications

NLP Based Customer Service Chatbot for Microsoft Azure*

https://software.intel.com/en-us/articles/use-analytics-zoo-to-inject-ai-into-customer-service-platforms-on-microsoft-azure-part-1
https://www.infoq.com/articles/analytics-zoo-qa-module/

Industrial Product Defect Detection in Midea*

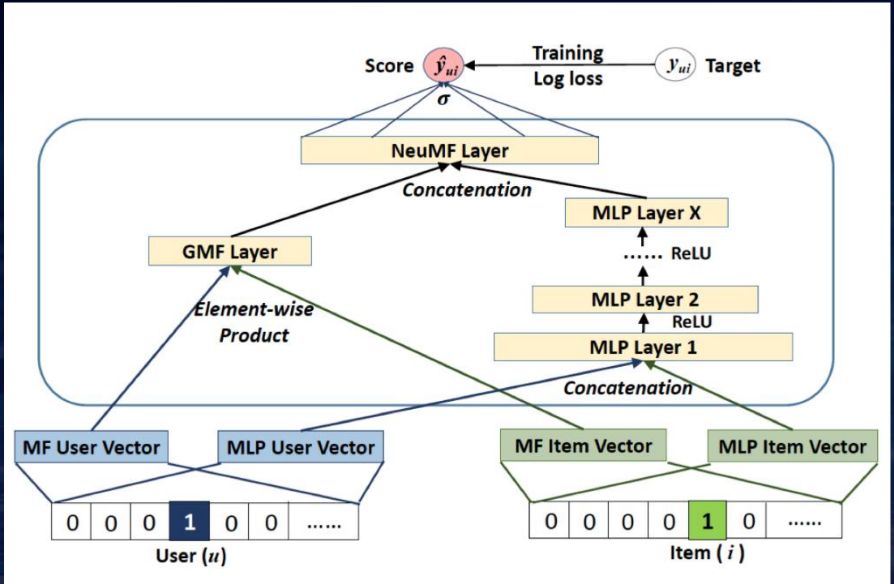
https://software.intel.com/en-us/articles/industrial-inspection-platform-in-midea-and-kuka-using-distributed-tensorflow-on-analytics

Unsupervised Time Series Anomaly Detection for Baosight*

https://software.intel.com/en-us/articles/lstm-based-time-series-anomaly-detection-using-analytics-zoo-for-apache-spark-and-bigdl

Any many more…

Neural Collaborative Filtering (NCF)

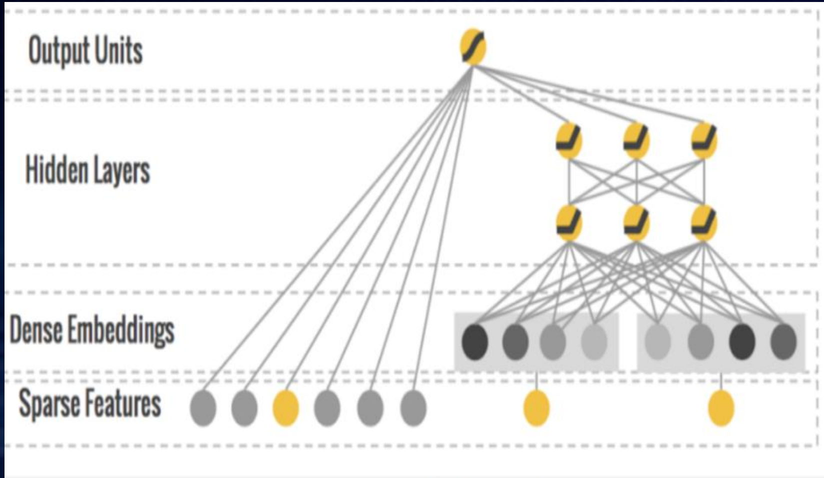


- NCF stimulates matrix factorization using DNN approach and is severed as a guideline for deep learning methods for recommendation services.
- It combines GMF with MLP to model user-item interactions.

```
from zoo.models.recommendation import NeuralCF

ncf = NeuralCF(user_count, item_count, class_num, user_embed=20,
               item_embed=20, hidden_layers=[40, 20, 10],
               include_mf=True, mf_embed=20)
ncf.compile(optimizer= "adam",
            loss= "sparse_categorical_crossentropy",
            metrics=['accuracy'])
ncf.fit(train_rdd,
        nb_epoch,
        batch_size,
        validation_data=val_rdd)
```

https://github.com/intel-analytics/analytics-zoo/tree/master/apps/recommendation-ncf

Wide & Deep Learning



Wide & Deep Models

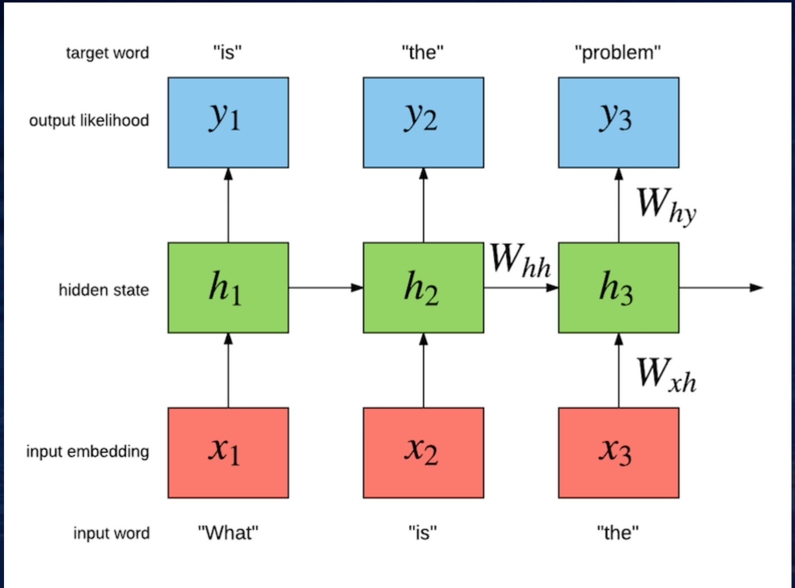
- Wide and Deep learning model can take rich data as input.
- The wide part can effectively memorize sparse feature interactions using cross-product feature transformations.
- The deep part can generalize to previously unseen feature interactions through low dimensional user and item embeddings similar to NCF.

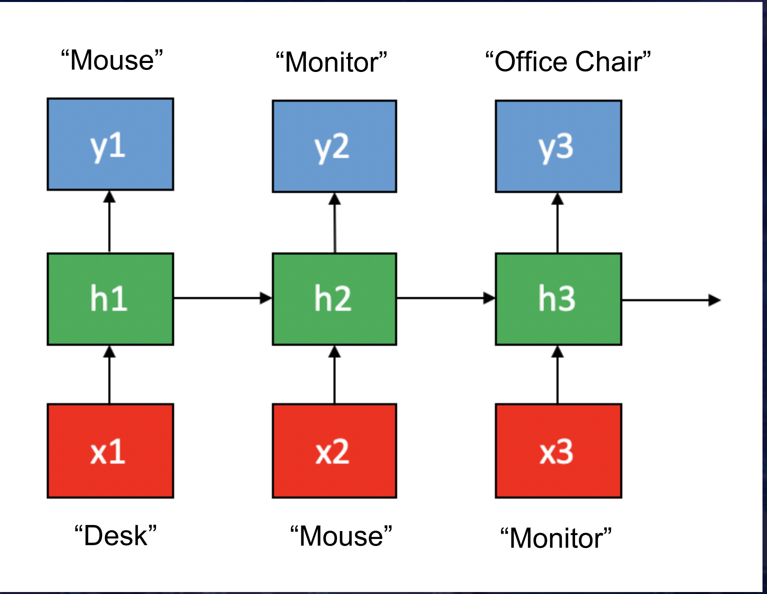
```python
from zoo.models.recommendation import WideAndDeep

# column_info can be shared by feature generation and model, where you can specify
# columns and their dimensions for each part of the model.
wnd = WideAndDeep(user_count, class_num, column_info,
                  model_type="wide_n_deep", hidden_layers=[40, 20, 10])
```

https://github.com/intel-analytics/analytics-zoo/tree/master/apps/recommendation-wide-n-deep

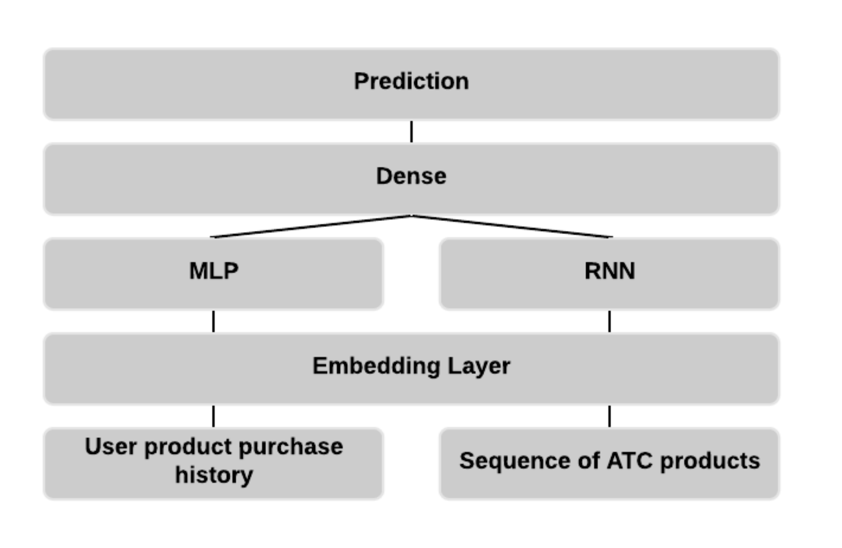
Session Recommender

- Each user session in an e-commerce system could be modeled as a sequence of web pages.
- A deep RNN could track how users browse the website using multiple hidden layers.





Session Recommender



The Good:

- Can catch the latest purchase intent from current session behavior and adjust its product recommendation in real time.
- Can work with both anonymous / identified customers.
- No pre-filtering mechanism required, simpler serving architect.

The Bad:

- Sequence window size is hard to set.
- Online inference requires lots of resources.

```
from zoo.models.recommendation import SessionRecommender

model = SessionRecommender(item_count, item_embed=100, rnn_hidden_layers=[40, 20],
                           session_length=5, include_history=True,
                           mlp_hidden_layers=[40, 20], history_length=10)
```

Performance Comparison

Offline measurement:

Method	Top 5 Accuracy
Session Recommender	52.3%
Wide & Deep	45.2%
NCF	46.7%
ALS	16.2%

Online measurement:

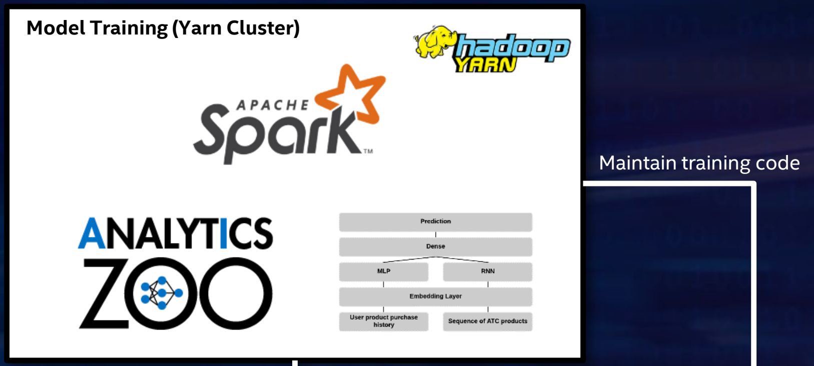
Online A/B testing shows the test group using Session Recommender lifted sales by 1% and average order value by 1.6% compared to control group.

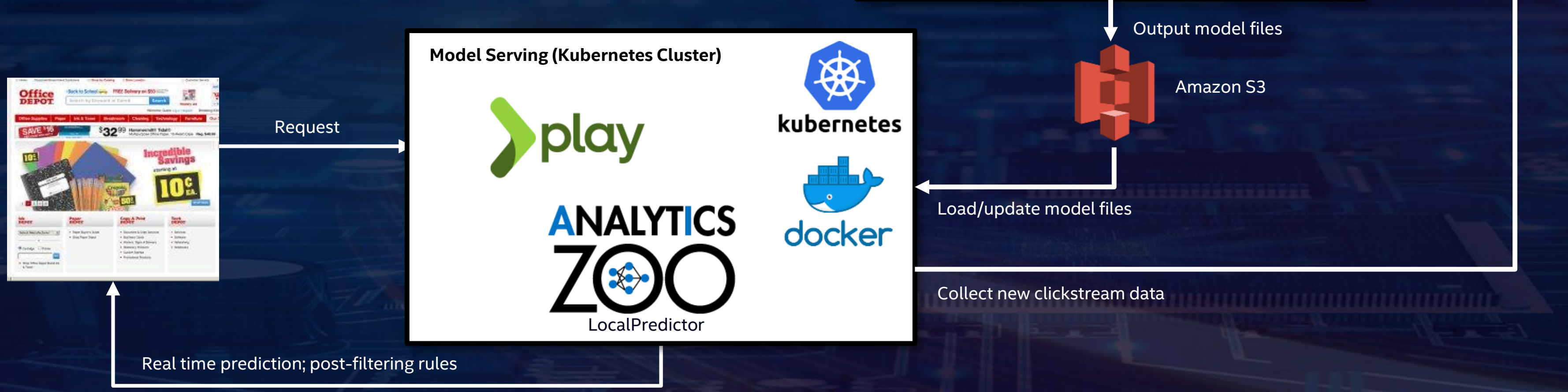
*Tested by Office Depot

Note: test data provided by Office Depot

Recommendation System In Production

- Automated model deployment pipeline.
- No down time when update model in production.
- Ability to scale up / down according to the current workload using Kubernetes.





Conclusion and Takeaways

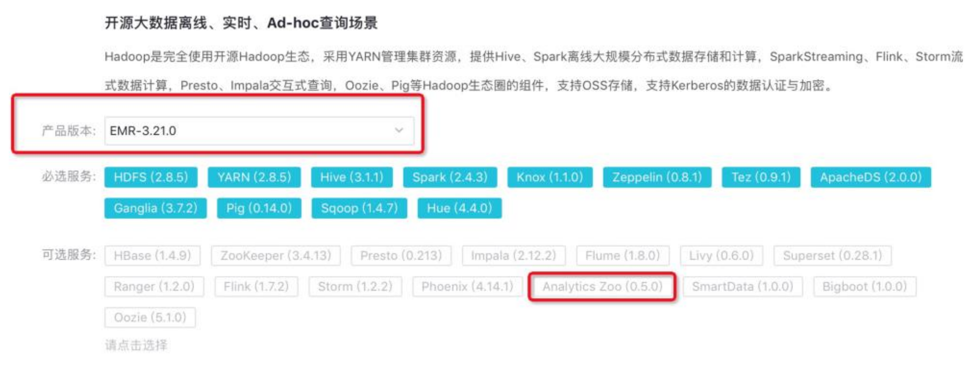
- Analytics Zoo integrates well into existing big data pipelines.
- Analytics Zoo provides model serving API for high performance real-time inference.
- Deep learning based recommendation provides more flexibility to combine different model architectures for different use cases.
- Lots of NLP algorithms (for example, transformers) can be utilized for recommendation.
- Check out the joint blog for more information:

https://software.intel.com/en-us/articles/real-time-product-recommendations-for-office-depot-using-apache-spark-and-analytics-zoo-on

Analytics Zoo on Ali E-MR

ANALYTICS ZOO + Alibaba Cloud aliyun.com

Analytics Zoo is already out-of-box on Ali EMR：



* Version upgrade for Analytics Zoo is on-going.

For more information and support, contact Wesley：

Email: wesley.du@intel.com
DingTalk:

More Information on Analytics Zoo

- Project websites
  - https://analytics-zoo.github.io/master/
  - https://github.com/intel-analytics/analytics-zoo
  - https://github.com/intel-analytics/bigdl
- Tutorials
  - CVPR 2018: https://jason-dai.github.io/cvpr2018/
  - AAAI 2019: https://jason-dai.github.io/aaai2019/
- "BigDL: A Distributed Deep Learning Framework for Big Data"
  - *In proceedings of ACM Symposium on Cloud Computing 2019 (SOCC'19)*
  - https://dl.acm.org/doi/10.1145/3357223.3362707
- Use cases
  - *Microsoft Azure, CERN, MasterCard, Baosight, Tencent, Midea, etc.*
  - https://analytics-zoo.github.io/master/#powered-by/

ANALYTICS ZOO

Unified Analytics + AI Platform

Distributed TensorFlow, Keras and BigDL on Apache Spark

https://github.com/intel-analytics/analytics-zoo intel®

LEGAL DISCLAIMERS

- Intel technologies' features and benefits depend on system configuration and may require enabled hardware, software or service activation. Learn more at intel.com, or from the OEM or retailer.
- No computer system can be absolutely secure.
- Tests document performance of components on a particular test, in specific systems. Differences in hardware, software, or configuration will affect actual performance. Consult other sources of information to evaluate performance as you consider your purchase. For more complete information about performance and benchmark results, visit **http://www.intel.com/performance**.