BigDL: Bringing Ease of Use of Deep Learning for Apache Spark

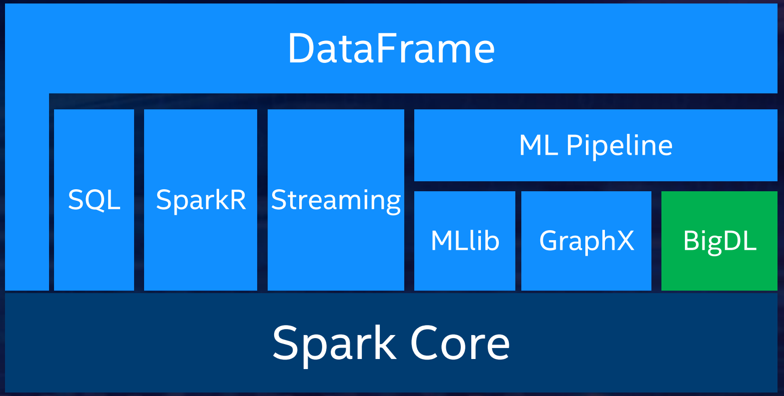
Jason Dai

Radhika Rangarajan

BigDL

Bringing Deep Learning To Big Data Platform

- Distributed deep learning framework for Apache Spark*
- Make deep learning more accessible to big data users and data scientists
  - Write deep learning applications as ***standard Spark programs***
  - Run on existing Spark/Hadoop clusters (***no changes needed***)
- Feature parity with popular deep learning frameworks
  - E.g., Caffe, Torch, Tensorflow, etc.
- High performance
  - Powered by Intel MKL and multi-threaded programming
- Efficient scale-out
  - Leveraging Spark for distributed training & inference



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

http://software.intel.com/bigdl

Why BigDL?

Chasm b/w Deep Learning and Big Data Communities

BigDL Answering The Needs

Make deep learning more accessible to big data and data science communities

- Continue the use of familiar SW tools and HW infrastructure to build deep learning applications
- Analyze “big data” using deep learning on the same Hadoop/Spark cluster where the data are stored
- Add deep learning functionalities to the Big Data (Spark) programs and/or workflow
- Leverage existing Hadoop/Spark clusters to run deep learning applications
  - Dynamically share with other workloads (e.g., ETL, data warehouse, feature engineering, statistic machine learning, graph analytics, etc.)

Overview of BigDL

Distributed Execution of BigDL Programs

Iterative

Mini-batch

Data parallel

Embarrassingly (data) parallel in nature

Training

```
for (i <- 1 to N) {
  batch = next_batch()
  output = model.forward(batch.input)
  loss = criterion.forward(output, batch.target)
  error = criterion.backward(output, batch.target)
  model.backward(input, error)
  optimMethod.optimize(model.weight, model.gradient)
}
```

Synchronous SGD

Inference

```
for (b <- 1 to D) {
  input = next_data(i)
  output = model.forward(input)
}
```

Run as standard Spark Programs

Standard Spark jobs

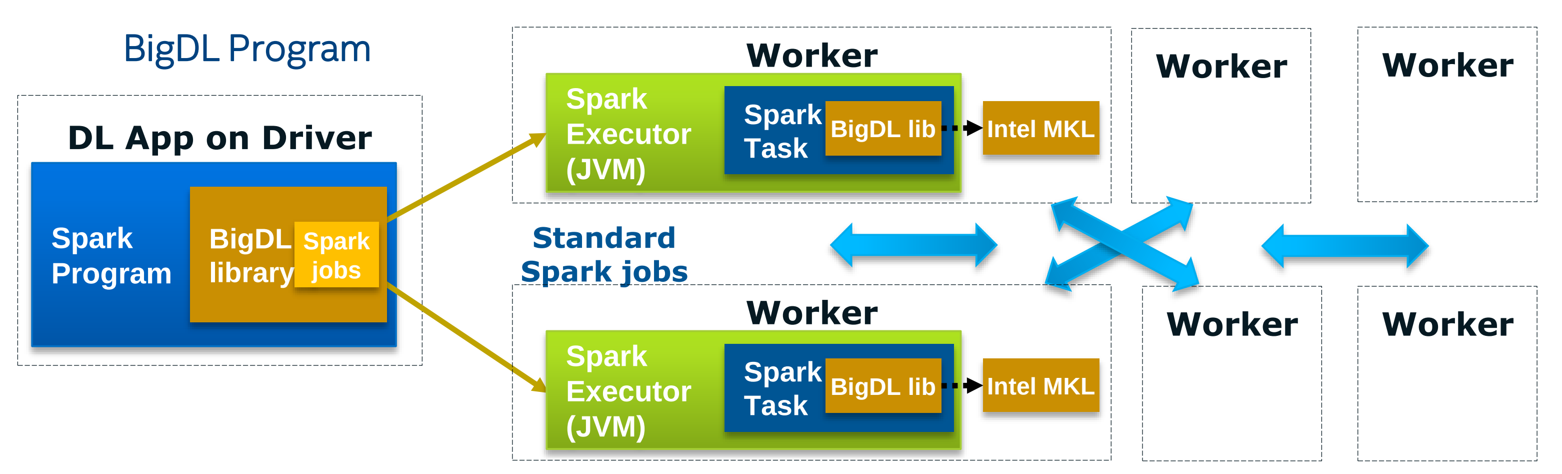
- No changes to the Spark or Hadoop clusters needed

Iterative

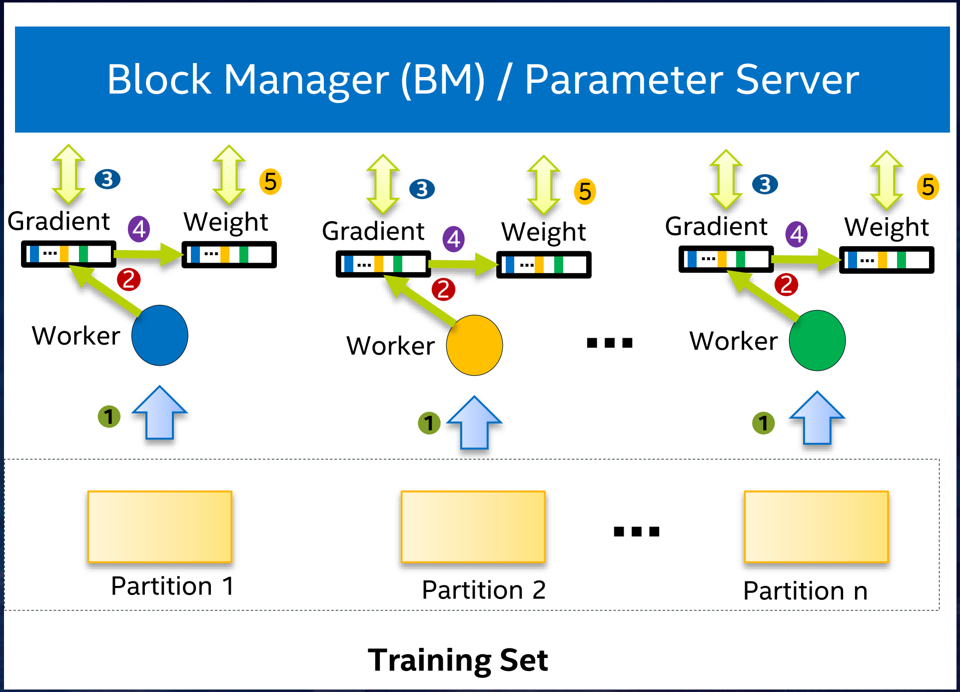
- Each iteration of the training runs as a Spark job

Data parallel

- Each Spark task runs the same model on a subset of the data (batch)



Synchronous Mini-Batch SGD



Peer-2-Peer All-Reduce synchronization implemented on top of Block Manager in Spark

BigDL APIs

Tensor

- Multi-dimensional array of numeric types (e.g., Float, Double, etc.)
- Generic support of numerical computing (using Intel MKL)

Sample

- Tuple of Tensors *(Input, Target)* representing a training / test sample

Module

- (100+) Layers of neural network (such as ReLU, Linear, SpatialConvolution, Sequential, etc.)

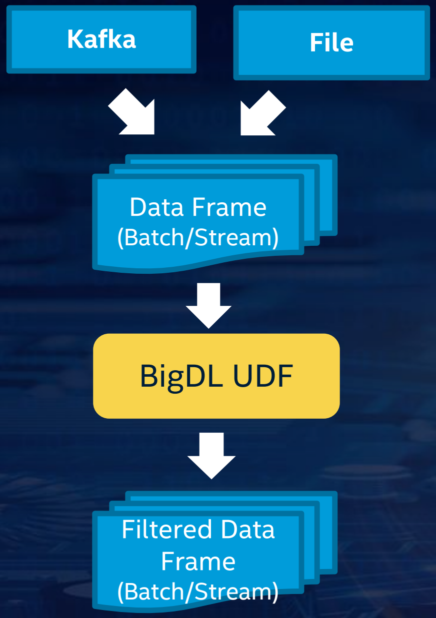
Criterion

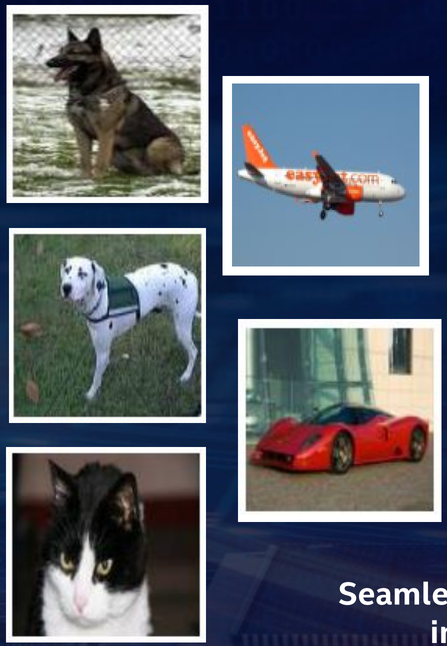
- Given input and target, computing gradient per given loss function

Optimizer

- Local & distributed optimizer (synchronous mini-batch SGD)
- *OptimMethod*: SGD, Adam, AdaGrad, RMSprop, etc.

Integration with Spark SQL, DataFrames and Structure Streaming



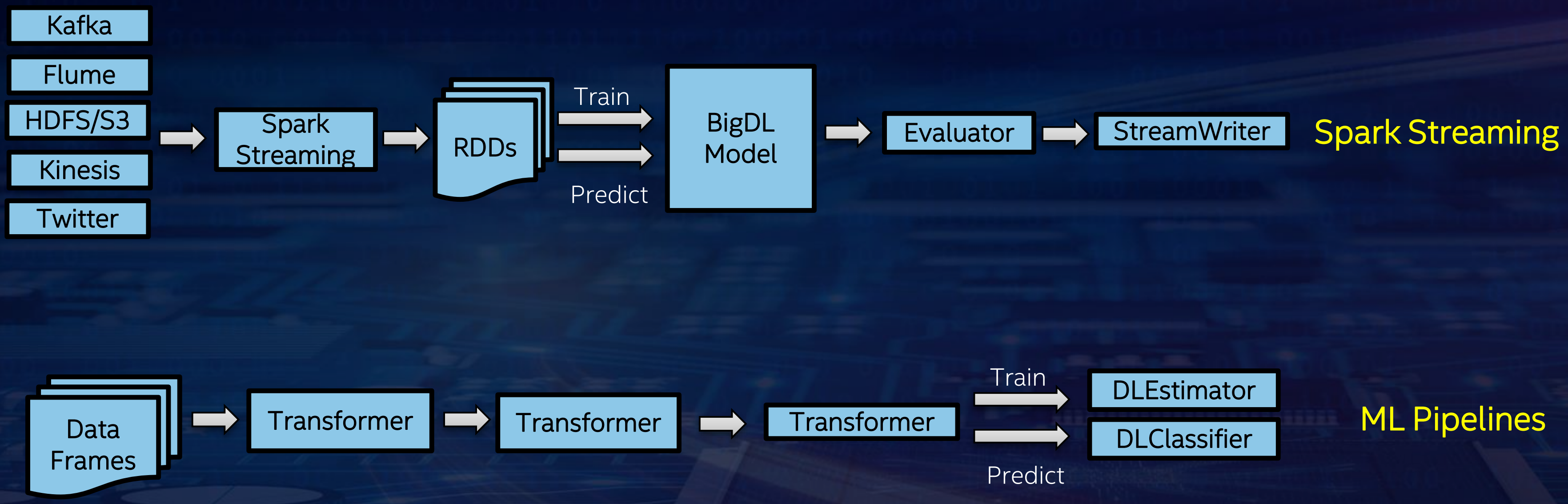


```
df.select($'image')
  .withColumn( "image_type",
     ImgClassifyInBigDL("image"))
  .filter($'image_type' == 'dog')
```

Seamless support of deep learning functionalities in *SQL queries* and *stream processing*

ImageNet dataset (http://www.image-net.org)

Integration with Spark Streaming and ML Pipelines



Latest BigDL Features

BigDL Releases

intel-analytics / BigDL

Unwatch 166 | Unstar 1,770 | Fork 337

Code | Issues 91 | Pull requests 30 | Projects 0 | Wiki | Settings | Insights

Downloads

Edit | New Page

Ian Wong edited this page 20 days ago · 3 revisions

These are built BigDL packages including dependency and python files. You can download these packages instead of building them by yourself. This is useful when you want to do something like run some examples or develop python code.

- BigDL 0.1.0

	Linux x64	Mac
Spark 1.5.1	download	download
Spark 1.6.0	download	download
Spark 2.0.0	download	download
Spark 2.1.0	download	download

- BigDL Nightly Build

Here are the folders for nightly build packages. The packages are built from latest master code. You can download the .zip files with a timestamp suffix in the name.

	Linux x64	Mac
Spark 1.5.1	download	download
Spark 1.6.0	download	download
Spark 2.0.0	download	download
Spark 2.1.0	download	download

Pages 23

- Overview
- BigDL Google Group and Mail List
- Documents
  - Scala Doc
  - Build
  - Getting Started
  - Python Support
  - Tutorials
  - Visualization with TensorBoard
  - Running on EC2
  - Examples
  - Programming Guide
  - Known Issues
  - Powered By
- Downloads

Clone this wiki locally

https://github.com/intel-an

Clone in Desktop

- **Open sourced in Dec 2016**
- **Latest release v0.1.0 (beginning of April'17)**
- **v0.1.1 targeting the coming week**
- **Next major release v0.2.0 soon**

https://github.com/intel-analytics/BigDL/wiki/Downloads

BigDL 0.1: Python Support & Notebook

Python API support

- Built on top of PySpark
- *Python 2.7 support since BigDL 0.1.0*
- *Python 3.5 support since BigDL 0.1.1*

Auto-packing Python dependency for YARN

- No need to pre-install any Python packages in the cluster

Jupyter notebook integration

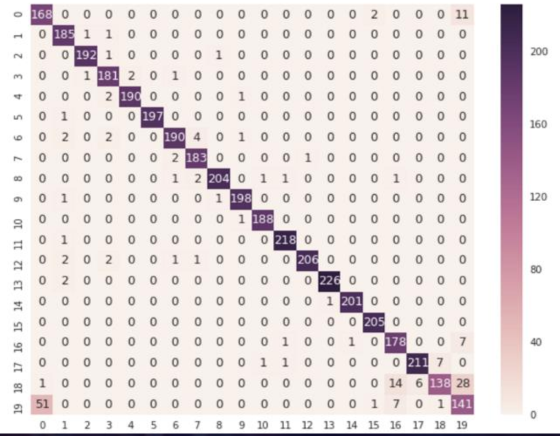
In [11]:

```
predictions = trained_model.predict(val_rdd).collect()

def map_predict_label(l):
    return np.array(l).argmax()
def map_groundtruth_label(l):
    return l[0] - 1

y_pred = np.array([ map_predict_label(s) for s in predictions])

y_true = np.array([map_groundtruth_label(s.label) for s in val_rdd.collect()])
```

In [12]:

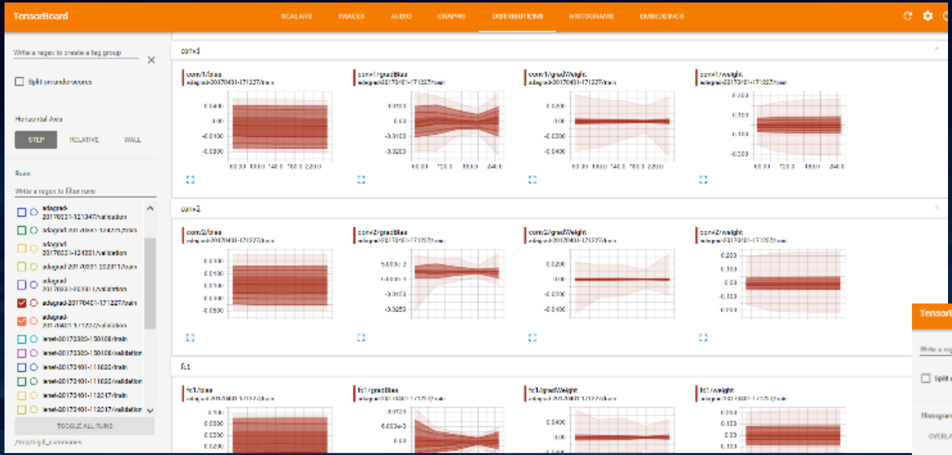
```
acc = accuracy_score(y_true, y_pred)
print("The prediction accuracy is %.2f%%"%(acc*100))

cm = confusion_matrix(y_true, y_pred)
cm.shape
df_cm = pd.DataFrame(cm)
plt.figure(figsize = (10,8))
sn.heatmap(df_cm, annot=True,fmt='d');
```

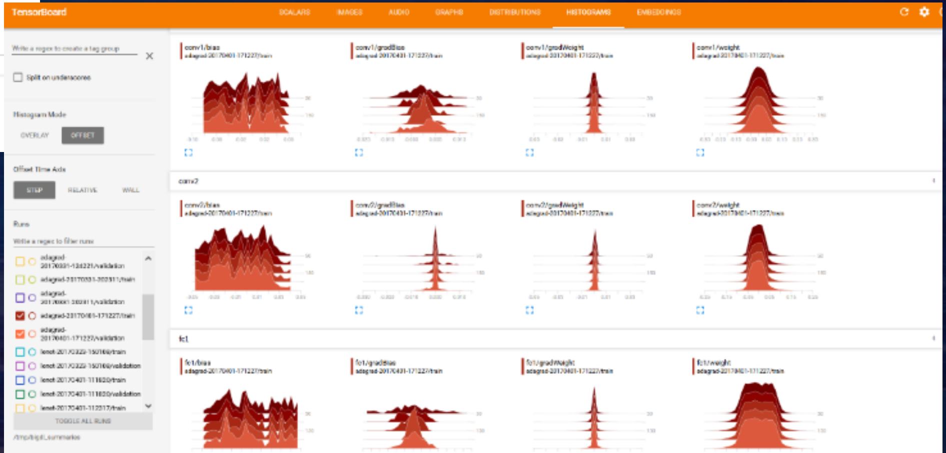
The prediction accuracy is 95.41%



https://github.com/intel-analytics/BigDL/blob/branch-0.1/pyspark/dl/example/tutorial/simple_text_classification/text_classfication.ipynb

BigDL 0.1: Integration With TensorBoard for Visualization

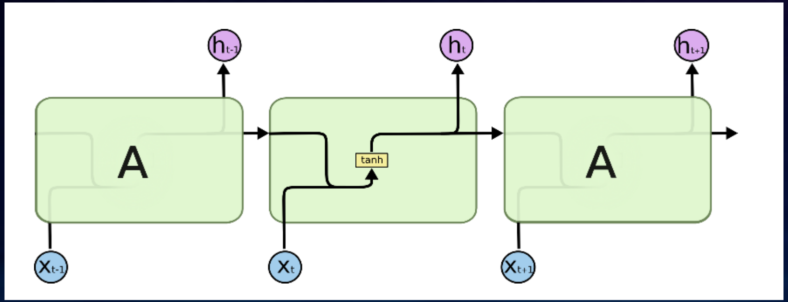




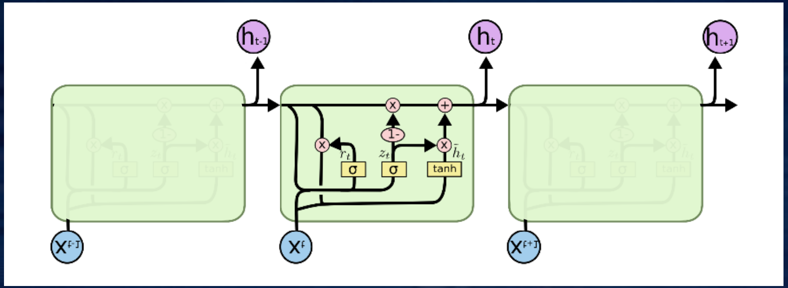
TensorBoard integration for visualizing BigDL program behaviors (available since BigDL 0.1.0)

https://github.com/intel-analytics/BigDL/blob/branch-0.1/pyspark/dl/example/tutorial/simple_text_classification/text_classfication.ipynb

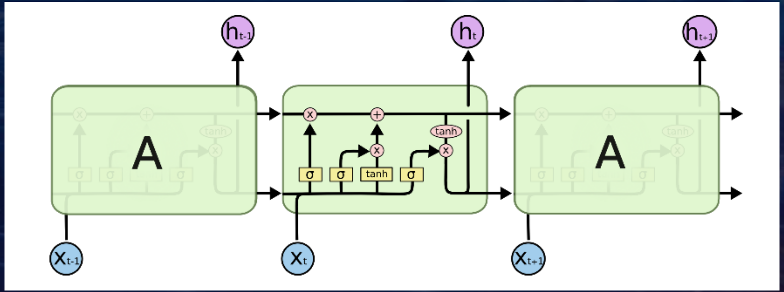
BigDL 0.1: Recurrent Neural Network Support



Simple RNN



LSTM (*since BigDL 0.1.0*)



GRU (*since BigDL 0.1.0*)

BigDL 0.2: Functional APIs

Functional API support in upcoming releases (*BigDL 0.2*)

- Similar to that in Keras and PyTroch
  - Each layer / module is callable
- Much easier to construct complex models
  - E.g., multi-input multi-output models, directed acyclic graphs, etc.

```
fc1 = Linear(4, 2)()
fc2 = Linear(4, 2)()
cadd = CAddTable()([fc1, fc2])
output1 = ReLU()(cadd)
output2 = Threshold(10.0)(cadd)

optimizer = Optimizer(
        model = Model([fc1, fc2], [output1, output2]),
        training_rdd=train_rdd,
        criterion=ClassNLLCriterion(),
        end_trigger=MaxEpoch(max_epoch),
        batch_size=batch_size,
        optim_method=Adagrad(learningrate=0.01,
                    learningrate_decay=0.0002))
train_model = optimizer.optimize()
```

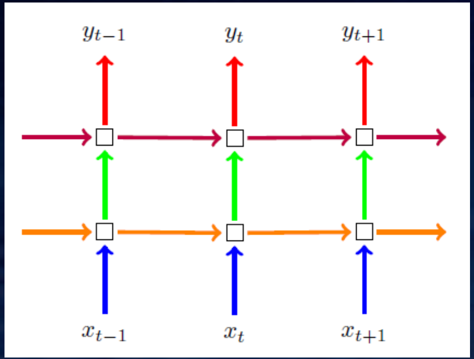
BigDL 0.2: Models Interoperability Support
(e.g., between TensorFlow, Caffe, Torch, BigDL models)

Load existing TensorFlow (in addition to Caffe and Torch) models into BigDL
- Allow model deployment in distributed analytics pipelines using Spark
- Allow for transfer learning, model tuning, model sharing (b/w data scientists and data engineers), etc.

Generate TensorFlow, Caffe and Torch models
- Allow BigDL models to be loaded into existing DL frameworks

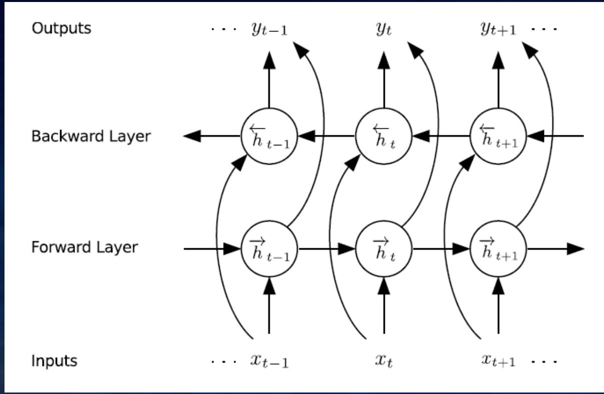
Run BigDL (model training & inference) as standalone program in local JVM
- Allow flexible deployment and serving of BigDL models in Java applications

BigDL 0.2: Advanced DL Functionalities



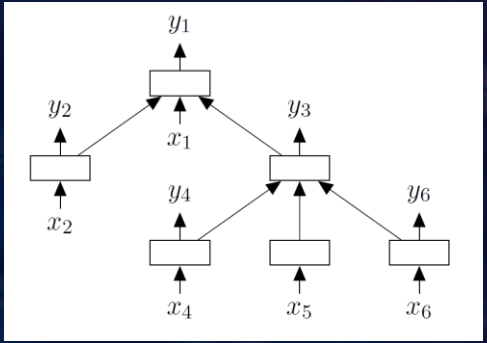
Recurrent Dropout

"A Theoretically Grounded Application of Dropout in Recurrent Neural Networks", Gal et al., NIPS 2016



Bi-directional RNN

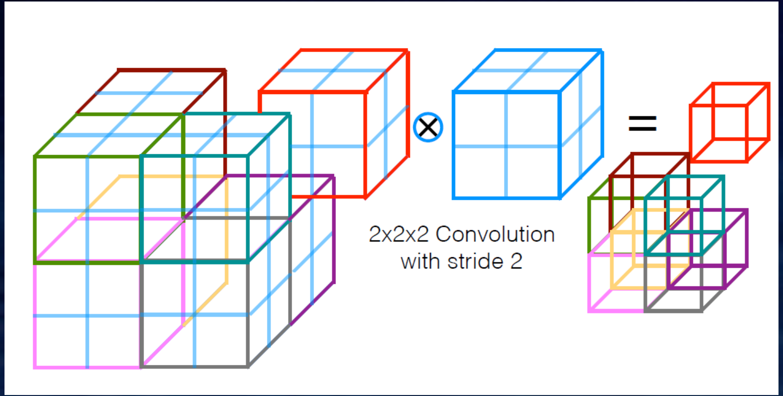
"Hybrid Speech Recognition with Deep Bidirectional LSTM", Graves et al., ASRU 2013



Tree-LSTM

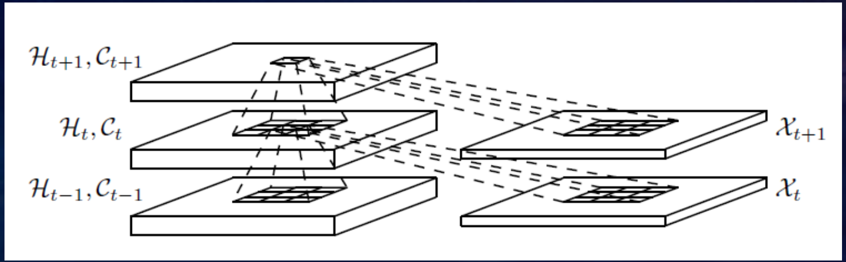
"Improved Semantic Representations From Tree-Structured Long Short-Term Memory Networks", Tai et al., ACL 2015

BigDL 0.2: Advanced DL Functionalities



3D Convolution

"V-Net: Fully Convolutional Neural Networks for Volumetric Medical Image Segmentation", Milletari et al., 3DV 2016



Convolutional LSTM

"Convolutional LSTM Network: a machine learning approach for precipitation nowcasting", Shi et al., NIPS 2015

BigDL Use Cases

Cloud & Big Data Platforms

Running BigDL, Deep Learning for Apache Spark, on AWS* (Amazon* Web Service)
https://aws.amazon.com/blogs/ai/running-bigdl-deep-learning-for-apache-spark-on-aws/

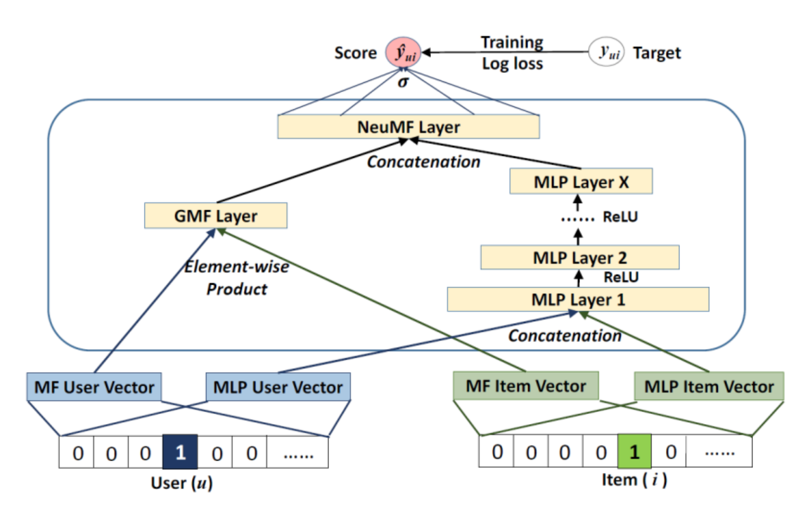
Use BigDL on Microsoft* Azure* HDInsight*
https://azure.microsoft.com/en-us/blog/use-bigdl-on-hdinsight-spark-for-distributed-deep-learning/

BigDL on Alibaba* Cloud E-MapReduce*
https://yq.aliyun.com/articles/73347

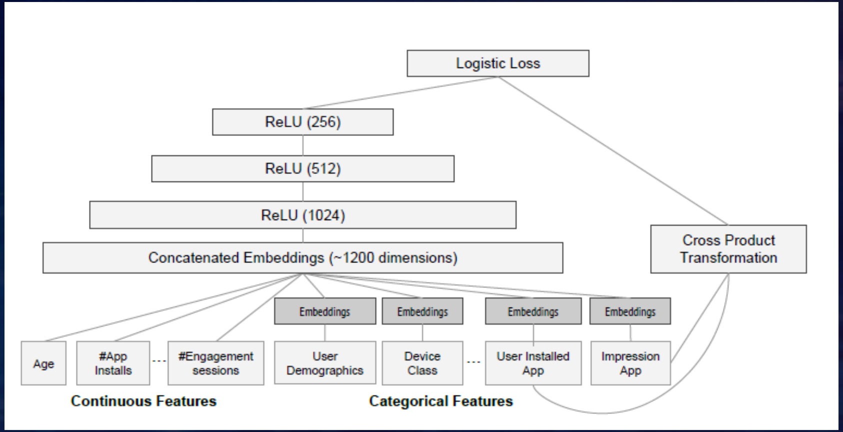
BigDL on CDH* and Cloudera* Data Science Workbench*
http://blog.cloudera.com/blog/2017/04/bigdl-on-cdh-and-cloudera-data-science-workbench/

Intel's BigDL on Databricks*
https://databricks.com/blog/2017/02/09/intels-bigdl-databricks.html

Recommendation

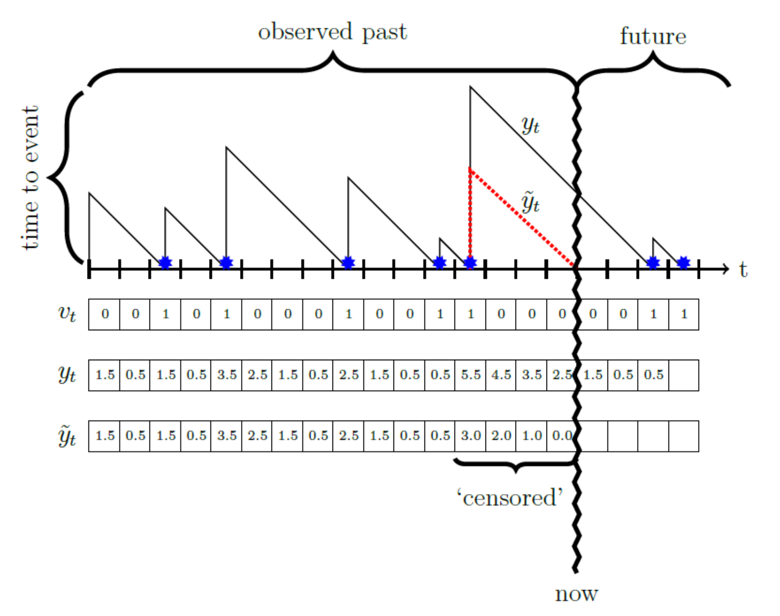


Neural Collaborative Filtering
He et al, WWW 2017



Wide & Deep Learning for Recommender Systems
Cheng et al, DLRS 2016

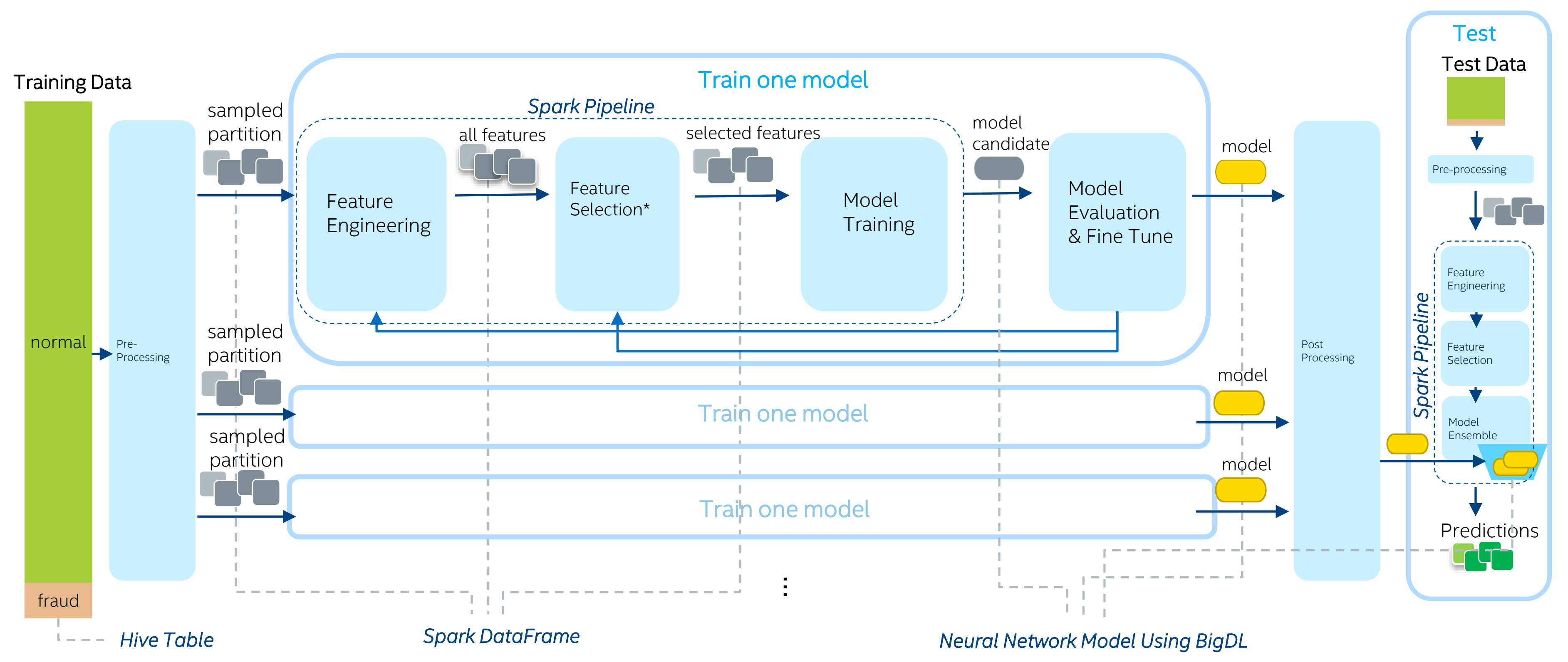
Churn Analysis



WTTE-RNN (Weibull Time-to-event Recurrent Neural Network)

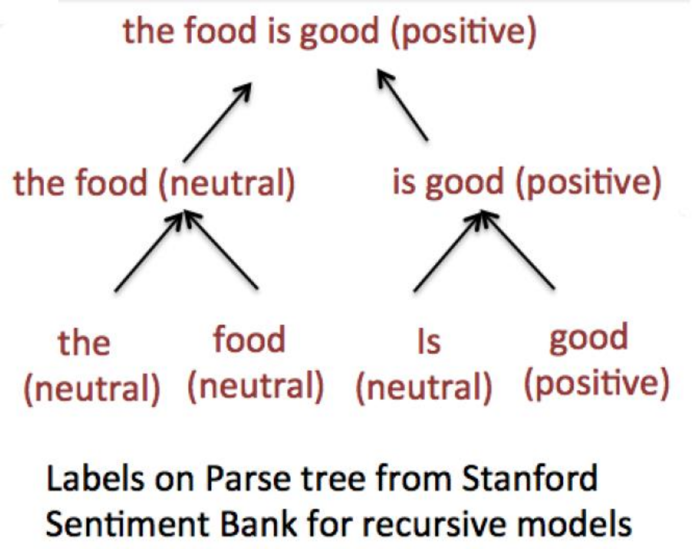
"WTTE-RNN : Weibull Time To Event Recurrent Neural Network", Egil Martinsson, Master's thesis in Engineering Mathematics & Computational Science, Chalmers University of Technology and University of Gothenburg

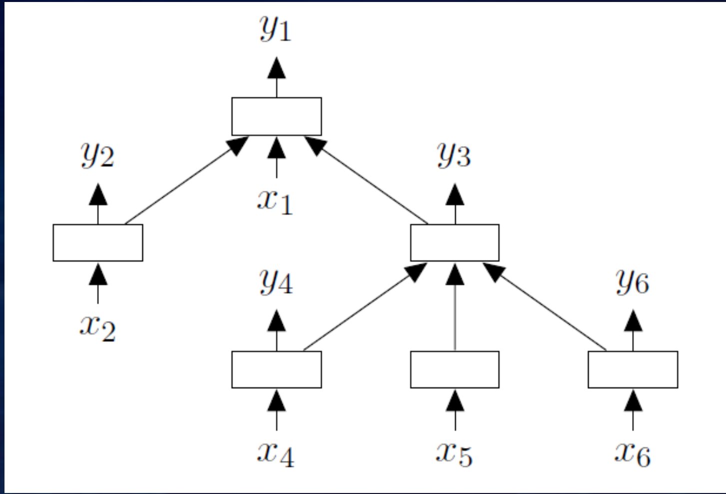
Fraud Detection in UnionPay



https://mp.weixin.qq.com/s?__biz=MzI3NDAwNDUwNg==&mid=2648307335&idx=1&sn=8eb9f63eaf2e40e24a90601b9cc03d1f

Sentiment Analysis for Natural Language

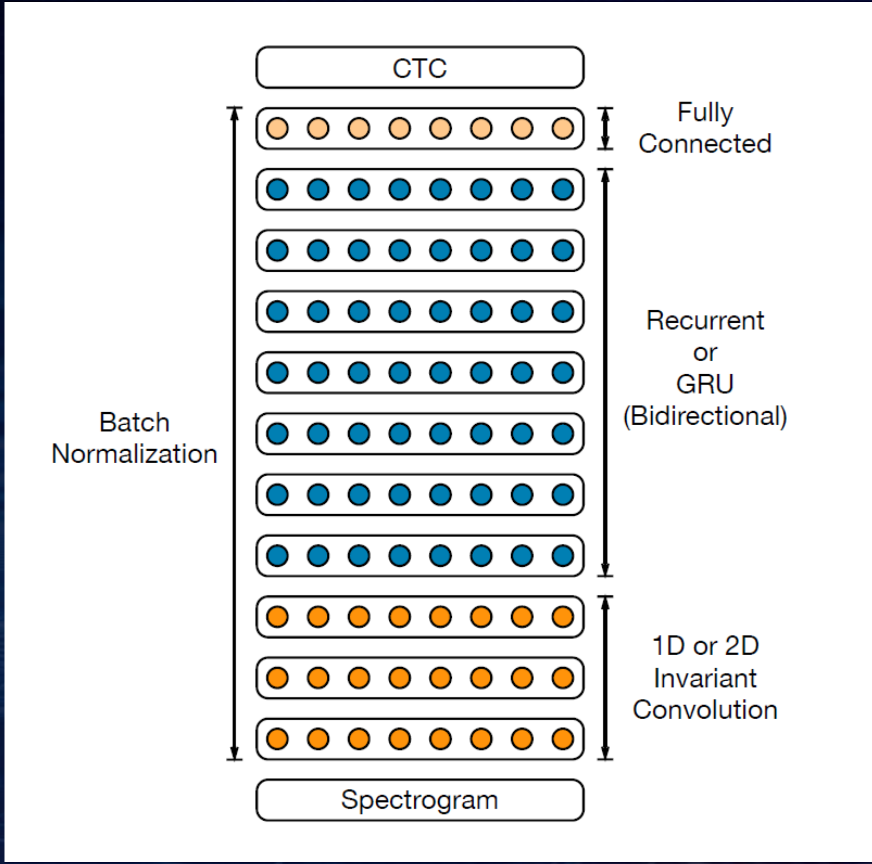




"When Are Tree Structures Necessary for Deep Learning of Representations?", Li et al., EMNLP 2015

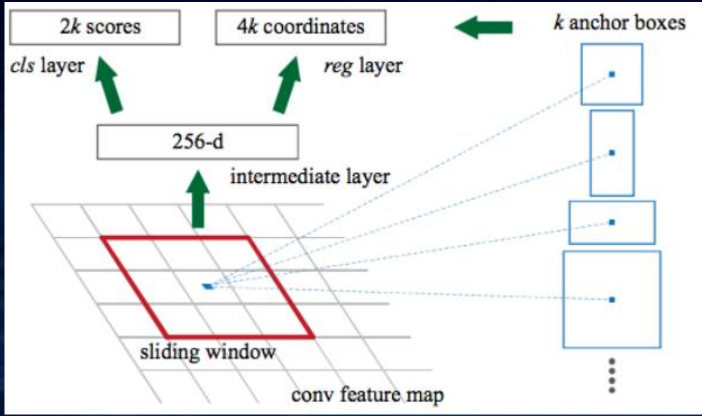
"Improved Semantic Representations From Tree-Structured Long Short-Term Memory Networks", Tai et al., ACL 2015

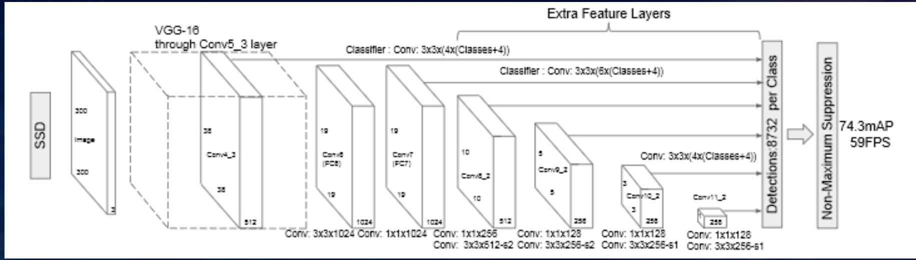
Speech Recognition



"Deep speech 2: end-to-end speech recognition in English and mandarin", Amodei et al., ICML'16

Image Recognition and Object Detection

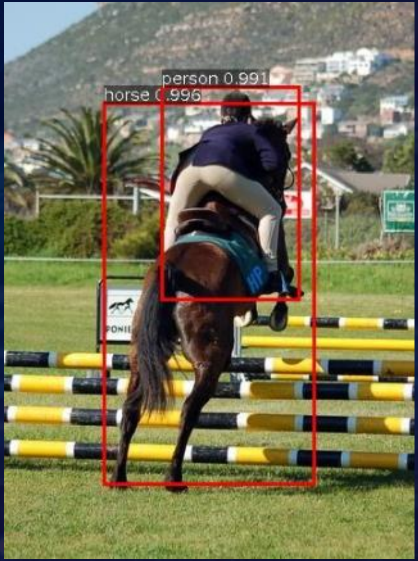




Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks, Ren et al., NIPS 2015

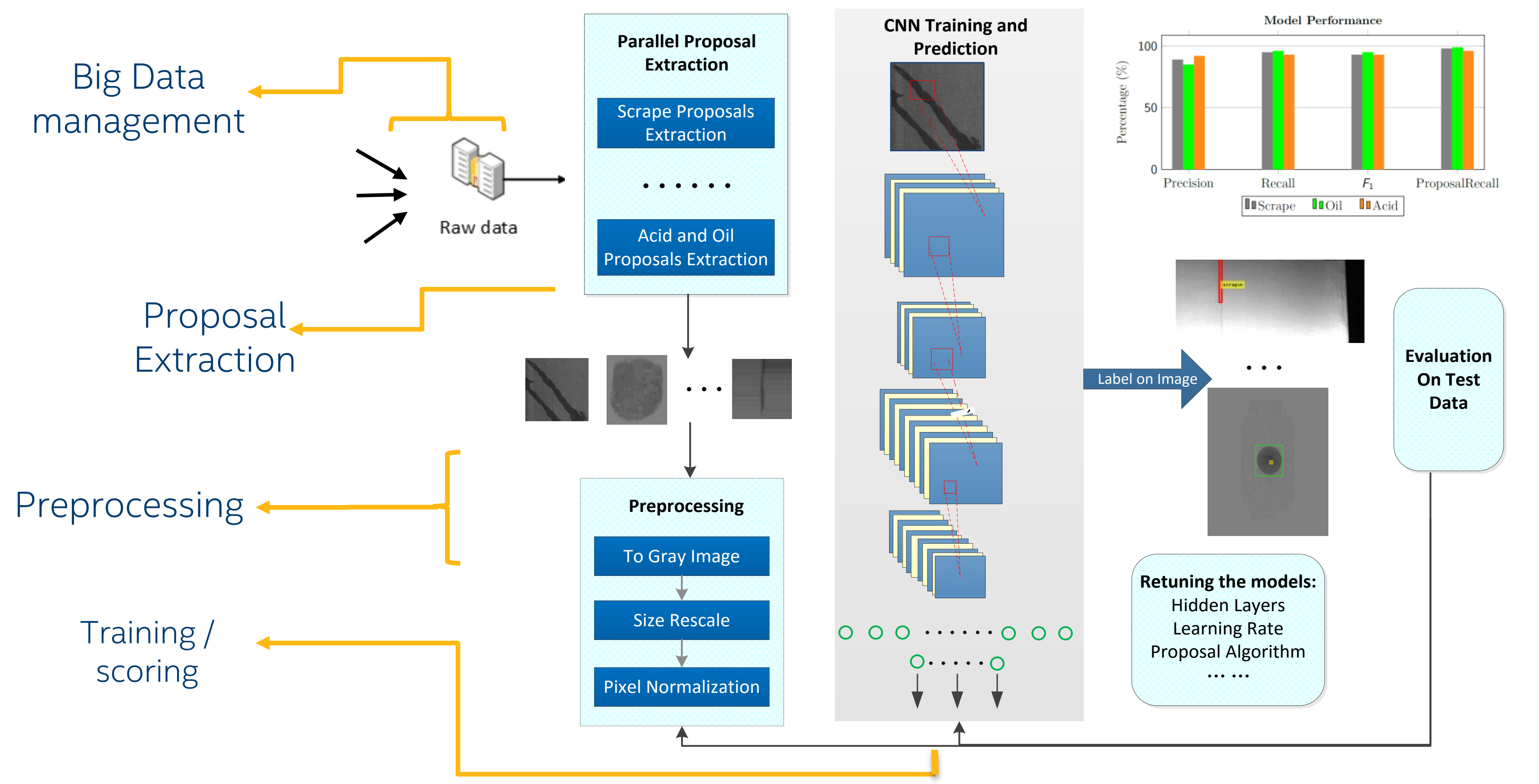
SSD: Single Shot MultiBox Detector, Liu et al., ECCV 2016

Image Recognition and Object Detection

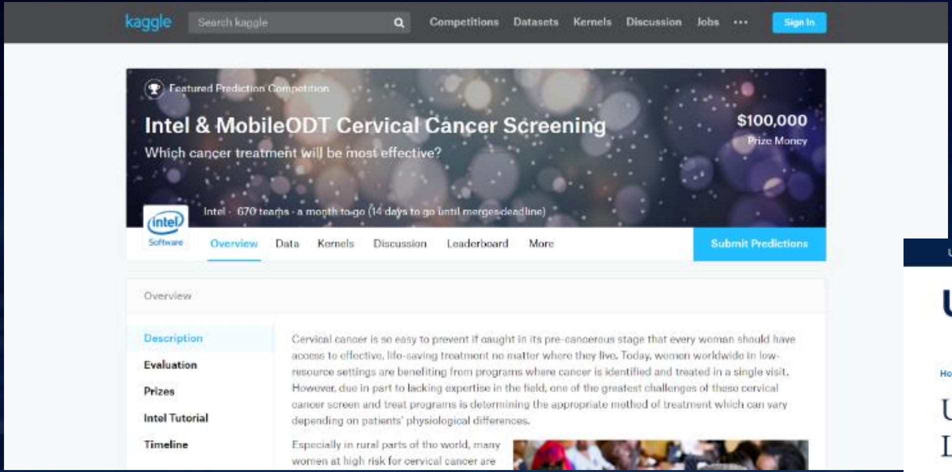


Pascal VOC data sets (http://host.robots.ox.ac.uk/pascal/VOC/)

Defect Detection in Manufacturing



3D Medical Imaging



https://www.kaggle.com/c/intel-mobileodt-cervical-cancer-screening

https://www.ucsf.edu/news/2017/01/405536/ucsf-intel-join-forces-develop-deep-learning-analytics-health-care

Partner With Us

- Use BigDL & Share your Experience
- Use Intel Optimized Libraries & Frameworks
- Leverage Intel Developer Zone Resources

https://github.com/intel-analytics/BigDL   http://software.intel.com/ai

AI: UNLEASHING THE NEXT WAVE

intel Artificial Intelligence

intel.com/ai
#intelAI

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.

intel®