Machine Learning and Deep Learning for IIOT

Chanchal Chatterjee, Dell EMC
Reston, March 22 2016
Goals of the Meeting

➢ Provide insights on methods and systems for machine learning and deep learning.
➢ Provide machine/deep learning use cases for IIOT.
➢ Provide architectures and frameworks for machine/deep learning for IIOT.
Machine Learning & Deep Learning – Confusing, Eh!

From Machine Learning Mastery (http://machinelearningmastery.com/)
Machine Learning and Deep Learning Dependencies

• Types of Data

• Types of Learning

• Types of Algorithms
Types of Data

• Structured Data
  • Time Series
  • Events
  • Graph

• Unstructured Data
  • Video/Images
  • Voice
  • Text
Types of Learning

• **Un-Supervised**
  • Do not require training data
  • Assume normal instances far more frequent than anomalies

• **Semi-Supervised**
  • Training data has labeled instances for only the normal class
  • Assume normal instances far more frequent than anomalies

• **Supervised**
Types of Algorithms

**ML: Machine Learning**
- Anomaly Detection
- Trends, Predictions & Forecasting
- Association & Grouping

**DL: Deep Learning**
- Ladder Network
- Convolutional Neural Network
- Recurrent Neural Network
- Deep Belief Networks
Some Details
Machine Learning

- Anomaly Detection
  - Point Anomaly
  - Contextual Anomaly
  - Collective Anomaly
  - Graph Anomaly

- Trends, Predictions & Forecasting

- Associations & Grouping
Deep Learning

- Ladder Network
- Convolutional NN (CNN)
- Recurrent NN (RNN)
  - Recurrent Recursive NN (R²NN)
  - Long Short Term Memory (LSTM)
- Deep Belief Networks (DBM)
  - Restricted Boltzmann Machine (RBM)
Deep Learning Networks

Source: http://www.asimovinstitute.org/neural-network-zoo/
Small Fraction of Real-World ML Systems Have ML Code

From Hidden Technical Debt in Machine Learning Systems
Scully et al, NIPS 2016
<table>
<thead>
<tr>
<th>Use Cases</th>
<th>Drivers</th>
<th>Challenges</th>
</tr>
</thead>
<tbody>
<tr>
<td>• Predictive maintenance</td>
<td>• Increase yield/asset utilization</td>
<td>• Low latencies</td>
</tr>
<tr>
<td>• Process optimization</td>
<td>• New revenue streams</td>
<td>• Data security</td>
</tr>
<tr>
<td>• Warehouse/supply chain</td>
<td>• Operational efficiencies</td>
<td>• Interoperability between diverse sets of equipment</td>
</tr>
<tr>
<td>optimization</td>
<td>• Increased worker satisfaction/safety</td>
<td>(typically with their own proprietary control system and data interchange</td>
</tr>
<tr>
<td>• Remote asset maintenance and</td>
<td>• Eco-sustainability</td>
<td>standard)</td>
</tr>
<tr>
<td>control</td>
<td></td>
<td>• Rapid interpretation of large volumes of data</td>
</tr>
<tr>
<td>• Product lifecycle</td>
<td></td>
<td>• Reliable indoor/outdoor coverage in harsh environments</td>
</tr>
<tr>
<td>monitoring</td>
<td></td>
<td>• Connectivity across different access technologies</td>
</tr>
<tr>
<td>• Integrated plant management</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Product-as-a-service</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Source: SDx Central IoT Infrastructure Report 2017
IIoT Architectures
Three Tier Deployment Model

Taken from Industrial Internet Reference Architecture v1.8
Edge2Core Crosscutting Functions

From IIC Edge2Code TG
IIOT Analytics Implementation Challenges

• How to Combine Streaming and Batch Processing Frameworks
• How to Introduce Human Domain Knowledge
  • NLP, Reinforced Learn, ...
• How to Distribute Processing and Data at the Tiers
IIOT Analytics Implementation Challenges

- How to train DL with Unlabeled data using Algorithms & Domain knowledge

- How to scale DL into multiple nodes

- How to tune DL Nws
  - architecture, parameters
IIOT End to End Frameworks

Collection & Aggregation Framework

Streaming Framework

Feature Engineering | Anomaly Detection | DL Model Execution

Machine Learning Framework (CPU?)

Processed Data

CPU Attached Storage

Deep Learning Framework (GPU?)

GPU Attached Storage

Models

Human Domain Knowledge

Proximity Network

Access Network

Service Network

IO Intensive

Compute Intensive
Open Source Frameworks for ML and DL
Deep Learning Frameworks

- Apache SINGA
- Brainstorm
- Caffe
- Chainer
- CNTK (Microsoft)
- DL4J
- DMLC
- Fbcunn (Facebook)
- Lasagne
- Minerva
- Mocha.jl (Julia)
- MXnet
- Neon (Nervana)
- Purine
- Tensorflow (Google)
- Theano
- Torch
- Warp-CTC (Baidu)
- Brain (Javascript)
- Cudamat
- Deep Learning Framework (Intel)
- Deepnet
- Hebel
- Infer.NET
- Keras
- Leaf
- MLPNeuralNet
- Neural Network Toolbox (MatLab)
- Neurltalk
- Neurolab
- OpenDeep
- PyBrain
- Swift-AI
- VELES (Samsung)

Each differ on – Licensing, Language implemented, OpenMP Support, OpenCL support, CUDA support, Various networks implemented, Pretrained model support and parallel implementations
# Deep Learning Frameworks

<table>
<thead>
<tr>
<th>Software</th>
<th>Software license</th>
<th>Platform</th>
<th><strong>OpenMP</strong> support</th>
<th><strong>OpenCL</strong> support</th>
<th><strong>CUDA</strong> support</th>
<th>Recurrent nets</th>
<th>Convolutional nets</th>
<th>RBM/DBNs</th>
</tr>
</thead>
<tbody>
<tr>
<td>TensorFlow</td>
<td>Apache 2.0</td>
<td>Linux, Mac OS X, Windows</td>
<td>No</td>
<td>On roadmap</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Caffe</td>
<td>BSD 2-Clause License</td>
<td>Linux, Mac OS X, unoffl Android, Windows</td>
<td>No</td>
<td>3rd party implementation</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Keras</td>
<td>MIT license</td>
<td>Linux, Mac OS X, Windows</td>
<td>Only if Theano backend</td>
<td>Under dev for Theano backend</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Deeplearning4j</td>
<td>Apache 2.0</td>
<td>Linux, Mac OS X, Windows, Android (Cross-platform)</td>
<td>Yes</td>
<td>On roadmap</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>MXNet</td>
<td>Apache 2.0</td>
<td>Linux, Mac OS X, Windows, AWS, Android, iOS, JavaScript</td>
<td>Yes</td>
<td>On roadmap</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Theano</td>
<td>BSD license</td>
<td>Cross-platform</td>
<td>Yes</td>
<td>Under development</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Facebook Torch</td>
<td>BSD License</td>
<td>Linux, Mac OS X, Windows, Android, iOS</td>
<td>Yes</td>
<td>3rd party implementations</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>
TensorFlow for Deep Learning

- Open source library for Machine Learning and Deep Learning by Google.
- Supports CUDA, CNN, RNN and DBN. Distributed TensorFlow offers flexibility to scale up to hundreds of GPUs, train models with a huge number of parameters.
- Has a well documented Python API, less documented C++ and Java APIs.
- Uses XLA, JIT, AOT, and other compilation techniques to minimize execution time and maximize computing resources.
- TensorFlow – Visualize TensorFlow graphs, monitor training performance, and explore how models represent data.
- Layers, Estimators, and Canned Estimators for defining models.
- Keras DL framework can be used in Tensorflow. DeepMind also uses TensorFlow.
- TensorFlow models can be deployed in iOS and Android apps, and Raspberry Pi.
- TensorFlow Serving, a flexible, high-performance ML serving system designed for production environments.
- TensorFlow has a toolkit of ML algorithms.
Apache Spark for Streaming and Machine Learning

- Open source library for SQL, Streaming, ML and Graph in a distributed cluster.
- Provides APIs for Scala, Java, Python and R.
- DAG execution engine supports acyclic data flow and in-memory computing.
- Runs on Hadoop, Mesos, standalone, or in the cloud. It can access diverse data sources including HDFS, Cassandra, HBase, Hive, and S3.
- Supports standalone (native Spark cluster), Hadoop YARN, or Apache Mesos.
- Spark Streaming has support built-in to consume from Kafka, Flume, Twitter, ZeroMQ, Kinesis, and TCP/IP sockets.
- Spark MLib simplifies large scale machine learning pipelines, including:
  - Summary statistics, correlations, stratified sampling, hypothesis testing, random data generation[16]
  - Classification and regression: support vector machines, logistic regression, linear regression, decision trees, naive Bayes
  - Collaborative filtering techniques including alternating least squares (ALS)
  - Cluster analysis methods including k-means, and Latent Dirichlet Allocation (LDA)
  - Dimensionality reduction techniques: singular value decomposition (SVD), and principal component analysis (PCA)
  - Feature extraction and transformation functions
  - Optimization algorithms such as stochastic gradient descent, limited-memory BFGS (L-BFGS)
- GraphX is a distributed graph processing framework.
THANK YOU