data-driven pipelines high performance computing machine and deep learning relationship and graph analytics streaming data analytics iot technology consulting high performance analytics architecture learned and intelligent systems technology vision, strategy, and practice blockchain and smart contracts data ontology and design data-driven pipelines high performance computing machine and deep learning relationship and graph analytics streaming data analytics iot technology consulting high performance analytics architecture infrastructure intelligent systems event-driven real-time analytics blockchain and smart contracts data ontology and design data-driven pipelines high performance computing machine messaging topologies relationship and graph analytics streaming data analytics iot technology consulting high performance analytics architecture infrastructure intelligent systems event-driven real-time event-driven systems blockchain and smart contracts data ontology and design data-driven pipelines high performance computing machine and deep learning relationship and graph analytics streaming data analytics iot technology consulting high performance analytics architecture infrastructure intelligent systems real-time event-driven systems blockchain and smart contracts data ontology and design data-driven pipelines high performance computing machine and deep learning relationship and graph analytics streaming data analytics iot technology consulting high performance analytics architecture infrastructure intelligent systems real-time event-driven systems blockchain and
Machine and Deep Learning: Practical Deployments and Best Practices for the next Two Years

Arno Kolster
Principal & Co-Founder

September 7, 2017
ARTIFICIAL INTELLIGENCE
Early artificial intelligence stirs excitement.

MACHINE LEARNING
Machine learning begins to flourish.

DEEP LEARNING
Deep learning breakthroughs drive AI boom.


Deep Learning?

Some definitions are in order...

Machine Learning:

- Supervised - guided (shallow)
- Unsupervised - unguided (shallow)
- Deep Learning – graph traversal

Generally speaking, the line lies between mimicking human learning patterns and allowing the machine to develop its own relationships and inferences.
Machine Learning Projects

What’s it like to actually try and do this right now?
“Files Of Interest”

Requirements:

Given an organized structure of files...

- determine groups of similar files, without opening the file itself

- classify a statistically relevant sample of each group to confirm validity of grouping

- at regular intervals, scan for patterns of interest against all files, using class-specific patterns

- send event notifications to command and control systems when patterns of interest are detected, including enough insight for those systems to handle the event
“Files Of Interest”

Technology Components:

- Messaging middleware
- Metadata Schema (relational)
- Unsupervised Learning (eventual Deep Learning)
- Supervised Learning (classification)
- Pattern/Anomaly Detection
- Event Generation
“Files Of Interest”

Technology Components:

- Messaging middleware
- Metadata Schema (relational)
- Unsupervised Learning (eventual Deep Learning)
- Supervised Learning (classification)
- Pattern/Anomaly Detection
- Event Generation
General Project Characteristics

The State Of Today

• ML/DL components are generally part of a larger workflow.

• Algorithm research making great strides both in commercial and open source offerings. Commercial edge in algorithms, but open source wins in overall solutions.

• Interoperability and workflow management lag (except perhaps for Spark).

• Messaging/Eventing required, many open source offerings require messaging and streaming systems.

• Lines between Supervised, Unsupervised, Deep Learning are still fluid and changing.

• Overfitting is a real problem, but at several levels (reuse is very limited).
  - Traditional ML notion within an algorithm/dataset
  - Within a workflow
  - Between ‘similar’ workflows
General Project Characteristics

The State Of Today (Continued)

• Development style (agile, use-case focused) means that general solutions are a long way off.

• **Algorithm and data formats are very closely linked now.** Adjustments to one require adjustments to the other. This lockstep is predictable, but also limiting in scope, reuse, general projects.

• “Experimental”. Monitoring, alerting, and operational support is very limited and tends to follow a devops approach. Operational aspects are almost universally open source.

• Performance ties to system technology choices by use-case, **tuning for the general case is not supported.**
Machine Learning Systems

Specialization is forced upon us, for now.
“The Ideal System”

• Workflow management is fluid, predictable, monitored, scales with infrastructure for ML and other components.

• Event management and message flow well-integrated into all components throughout.

• Algorithm choice and dataset formatting independent and interoperable.

• Algorithm performance detects and adjusts to system capabilities.

• Acceleration is dynamic, reusable, and independently scalable.

• Analyst tools are easily understood, run on all platforms, and support SME language.

• Visualization capabilities tuned to the consumer (mobile, tablet, social media are first class citizens).

• Availability, interoperability, metrics, and KPIs well-exposed and generally understood. Integration with existing NOC tools complete.
Building for 12-18 Months
Era of “Use-Specific Systems”

• Try and collate use-cases and design for as many similarities as possible.

• Rely on event ontologies to handle multiple cases where algorithms are the same. These really are out there, but you cannot expect anyone (commercial/open source) to point them out, much less automate them.

• Externalize rules into software; teach analysts and SMEs how to explain your rules, but use true developers to build them.

• Leverage high-powered workstations (Mac, PC) for algorithm development and acceleration testing. Once algorithms are chosen, design small clusters or servers to handle acceleration for those functions.

• Scale-out, not up. 8 GPU/node is probably the limit for now – especially without prevalence for alternate cooling.

This is common right now: the news is full of stories about systems for learning to understand ordering in online stores, learning to drive autonomous vehicles, recognizing persons of interest on cameras, tracking human movement in games/film production/animation, etc. There are no stories of “general purpose deep learning”. And there won’t be for a while.
Building for 12-18 Months

Era of “Use-Specific Systems”

• **Workflow management** must be planned from the outset. Don’t bolt components together. This is hard work, but is also where most projects fail. Going from the mind of the researcher to business insight automatically is very difficult. There is no easy fix here and don’t believe anyone trying to convince you otherwise (yet).

• “**Codes are antiquated and inefficient**”. To start and stay current, take a streaming/service approach to your application. Handling failure will be a challenge for non-traditional HPC shops (no checkpoints).

• **Each “Learning System” is distinct**. So cross-region/datacenter availability is different here. Enterprise teams used to replication for availability will face challenges.

• Reuse of the system is limited because of the specificity in the choices above. **Expect to build several of these**, so don’t go crazy building giant clusters. Scale incrementally. This is not the “data lake” or “Hadoop” approach. There is not even one size – and they definitely don’t fit all.

*This is common right now: the news is full of stories about systems for learning to understand ordering in online stores, learning to drive autonomous vehicles, recognizing persons of interest on cameras, tracking human movement in games/film production/animation, etc. There are no stories of “general purpose deep learning”. And there won’t be for a while.*
Era of “Software Interoperability” with Specialized Acceleration

- Gain some economies of scale for capex/opex by centralizing workflow, management, monitoring & alerting, messaging, and data storage (by class of persistence). These will not be “world class” yet as requirements are still in constant flux and the charge still led by open source.

- Development will focus on improving tools for analysts (both in research/modeling and in visualization/reporting)

- Algorithms will continue to improve, but still specific to dataset and accelerator.

The next shift will be around the “glue” holding things together. We will see common ontologies for events, messaging, metadata, rules, classification, etc. emerging. This will allow rapid prototyping for new use cases and provide some savings by centralizing workflow, messaging, monitoring, etc. but acceleration will remain algorithm and dataset specific in most cases. Most serious shops will have abandoned the “codes/batch” approach.
Building for 18-36 Months

Era of “Software Interoperability” with Specialized Acceleration

- **Scale-Out continues**, although we will see a few early prototypes for scale-up systems (3DXPoint, many-core, better GPU intercommunication). Water-cooling emerges for accelerated systems.

- **Network compute begins** accelerating some primitives and patterns. This increases overall model throughput and starts to allow for “general algorithms” by differentiating dataset from algorithm. This is hard programming though and will progress slowly. SMEs are likely to fight against this effort because the logic is divorced from the data. Education required to overcome this.

- **Some standards for the most popular algorithms** allow some clusters to be combined/collapsed, but this is still largely use-case specific.

The next shift will be around the “glue” holding things together. We will see common ontologies for events, messaging, metadata, rules, classification, etc. emerging. This will allow rapid prototyping for new use cases and provide some savings by centralizing workflow, messaging, monitoring, etc. but acceleration will remain algorithm and dataset specific in most cases. Most serious shops will have abandoned the “codes/batch” approach.
Building for 36+ Months

Era of “Generally Specialized” Systems

• **Capex/Opex synergies continue**, with both open source and commercial offerings to leverage shared hardware and software platforms.

• **Visualization tools, especially for mobile/wearables key focus** for application developers. Techniques for rendering/leveraging mobile SoCs for stunning interaction and detail beyond the capabilities of standard browsers.

• **Monitoring/alerting systems finally integrating** into enterprise-level systems. Still buggy and limited but much of that is due to legacy interfaces on the enterprise side. Commercial systems in catch-up mode.

By now, several vendors have well-integrated software suites which cover workflow, messaging, algorithms, data persistence, and support acceleration which is well integrated. Visualization and analyst tools are not well-integrated yet, because the number of successful deployments across industries is limited. These integration platforms will provide specialized systems tailored to “general learning”. The efficacy of those systems will be unproven for several more years however. Market leaders now (Baidu, Google, Facebook, etc.) will continue releasing open source versions of the bundled suites which outperform, but at the cost of internal specialization as before.
Building for 36+ Months

Era of “Generally Specialized” Systems

• **Scale-up systems become a focus** again as interconnects, memory technologies, and core-counts coupled with software ubiquity make running large-scale workloads easier. Paradigms do not change however – streaming and messaging still underlie the vertical solutions, just the platform containers shift and better interconnects/shared memory out-scale distributed deployments for certain use-cases.

• Network compute built into several open source and commercial offerings, and **simple messaging patterns are routinely accelerated**.

By now, several vendors have well-integrated software suites which cover workflow, messaging, algorithms, data persistence, and support acceleration which is well integrated. Visualization and analyst tools are not well-integrated yet, because the number of successful deployments across industries is limited. These integration platforms will provide specialized systems tailored to “general learning”. The efficacy of those systems will be unproven for several more years however. Market leaders now (Baidu, Google, Facebook, etc.) will continue releasing open source versions of the bundled suites which outperform, but at the cost of internal specialization as before.
Thank you.

arno.kolster@providentiaworldwide.com

@providentia_ww

www.providentiaworldwide.com

Special Thanks To Hyperion Research