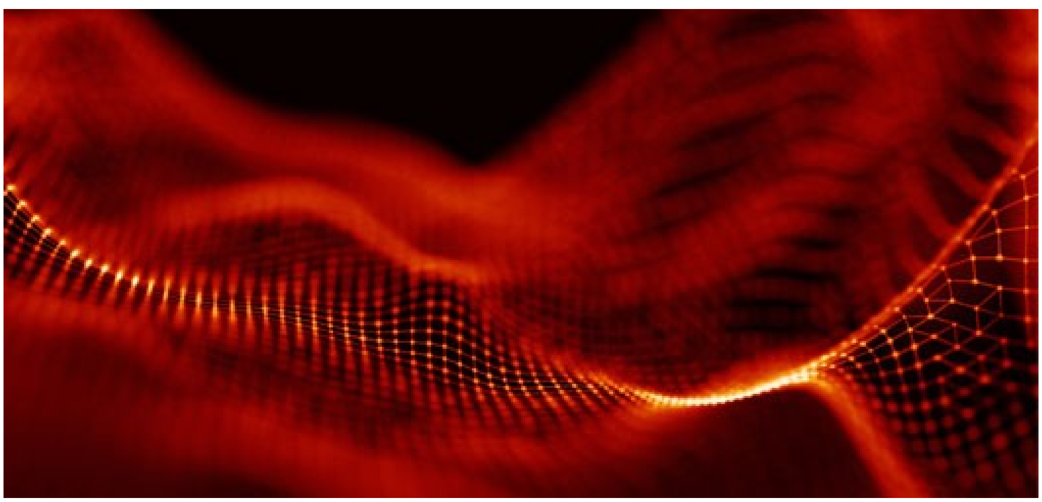
IBM Systems MEDIA

IBM Machine Learning for z/OS Is Supported By Apache Spark

It's possible to build machine learning systems that learn from each user interaction, or from new data collected by an IoT device.



By Jean-Francois Puget and Mythili Venkatakrishnan

07/05/2017

Businesses must constantly adapt to changing conditions: competitors introduce new offerings, consumer habits evolve, the economic and political environment change, etc. While this isn't new, the velocity at which business conditions change is accelerating. This pace of change places a new burden on technology solutions developed for a business.

Over the years, application developers moved from V-shaped projects (with multiyear turnaround) to agile development methodologies (with turnaround in months or quicker). This has enabled businesses to adapt their application and services much more rapidly. Examples include:

- A saleS forecasting system for a retailer: The forecast must take into account today's market trends, not just those from last month. And, for real-time personalization, it must account for what happened as recently as one hour ago.
- A product recommendation system for a stockbroker: They must leverage current interests, trends and

movements, not just last month’s
- A personalized healthcare system: Offerings must be tailored to an individual’s unique circumstance. Healthcare devices, connected using the Internet of Things (IoT), can collect data on human-machine interaction and behavior.

These scenarios, and others like them, create a unique opportunity for machine learning, which was designed to address the fluid nature of these problems.

Machine learning moves application development from programming to training: Instead of writing new code, the application developer trains the same application with new data. This is a fundamental shift in application development because new, updated applications can be obtained automatically on a weekly, if not daily, basis. This shift is at the core of the cognitive era in IT.

Machine learning enables the automated production of actionable insights where the data is (i.e., where business value is greatest). It’s possible to build machine learning systems that learn from each user interaction, or from new data collected by an IoT device. These systems then produce output that takes into account the latest available data. This wouldn’t be possible with traditional IT development, even if agile methodologies were used.

Gather Information

While most clients understand machine learning, too few are turning this into action. They are either slowed down by concerns over their data assets or they attempt it one time and then curtail efforts, claiming the results weren’t worth continuing with machine learning. These are common concerns and considerations, but they should be recognized as items that are easily surmounted, with the right approach.

A common trap is to believe that data is all that’s needed for a successful machine learning project. Machine learning projects that start with a large amount of data, but lack a clear business goal or outcome, are likely to fail. Projects that start with little or no data, yet have a clear and measurable business goal, are more likely to succeed. The business goal should dictate the collection of relevant data and also guide the development of machine learning models. This approach provides a mechanism for assessing the effectiveness of machine learning models. The second trap in a machine-learning project is to view it as a one-time event. Machine learning, by definition, is a continuous process and projects must be operated with that consideration.

Machine learning projects are often run as follows:

1. They start with data and a new business goal
2. Data is prepared, because it wasn’t collected with the new business goal in mind
3. Once prepared, machine learning algorithms are run on the data to produce a model
4. The model is evaluated on new, unforeseen data to see whether it captured something sensible. If it does, it’s deployed in a production environment where it’s used to make predictions on new data.

This typical approach is valuable, but it’s limited by the fact the models learn only once. While you may have developed a great model, changing business conditions may make it irrelevant. For instance, assume machine learning is used to detect an anomaly in credit card transactions. The model is created using years of past transactions, and anomalies are fraudulent transactions. With a good data science team and the right algorithms, it’s possible to obtain a fairly accurate model. This model can then be deployed in a payment system where it flags anomalies when it detects them. Transactions with anomalies are then rejected. This is effective in the short term, but clever criminals will soon recognize that their scam is detected. They will adapt, and find new ways to use stolen credit card information. The model won’t detect these new ways because they weren’t present in the data that was used to produce it. As a result, the model effectiveness will drop.

To avoid this performance degradation, monitor the effectiveness of model predictions by comparing them with actuals. For instance, after some delay, a bank will know which transactions were fraudulent. Then it’s possible to compare the fraudulent transactions with the anomalies detected by the machine learning model. From this comparison, one can compute the accuracy of the predictions. One can then monitor this accuracy over time and watch for drops. When a drop happens, it’s time to refresh the machine learning model with more up-to-date data, a feedback loop (see Figure 1).

With a feedback loop, the system learns continuously by monitoring the effectiveness of predictions and retraining when needed. Monitoring and using the resulting feedback are at the core of machine learning. This is no different from how humans perform a new task. We learn from our mistakes, adjust and act.

Business Use Strategy

The first pillar of the IBM Machine Learning strategy is to support users through this end-to-end workflow by providing tools to assist users in every step. A first set of tools, IBM Data Science Experience, deals with data preparation, model creation and model selection. A second set of tools deals with models once they are created. They include a model repository with governance support, a deployment interface where one can create a deployment in one click or one API call, and a console to monitor deployed models accuracy. Machine learning is the union of these.

The second pillar of the IBM Machine Learning strategy is to leverage open-source algorithms. The open-source community is providing state-of-the-art machine learning technology for free. IBM has started with Spark ML, a leading-edge set of algorithms that scale to petabyte data sets. IBM is in the process of adding support for popular packages for R, Python and Java users. We are also in the process of including support for major deep learning packages.

The third pillar of the IBM Machine Learning strategy is to deliver it on every platform that matters. This started with IBM Machine Learning for z/OS* (ibm.co/2pGrjVj (https://www.ibm.com/ms-en/marketplace/machine-learning-for-zos)). For other platforms, IBM Watson* Machine Learning is a public cloud version in beta test, and a private-cloud offering is in the works. One key feature of this strategy is to allow cross platform workflows; for instance, have models created on the public cloud platform, and then deploy these models on a mainframe for scoring. This hybrid cloud strategy will be key to support the user journey to hybrid cloud.

Machine learning techniques can provide tremendous value and non-obvious insight to business decisions. As machine learning adoption grows, implementation characteristics are becoming key factors contributing to a successful outcome. For example, the currency of data that feeds the algorithms can determine whether the insight is usable in a real-time context. Using most current data for applications of machine learning such as predicting client loyalty and churn, determining likelihood of default or fraud, or identifying the most appropriate candidates and time frame for offers, will increase the value and usability of the insight.

Efficient execution of machine learning capabilities is another contributing factor clients use in determining the value they derive. Efficiencies can be realized from various aspects such as cost, skill availability and implementation complexity. The increasing implications and regulations around security and privacy are already considerations for clients implementing analysis of data, but these will become even more relevant as traditional data obfuscation methods as part of extraction, transformation and loading mechanisms won’t suffice to meet the demands of real-time results and the growing pervasiveness of analytics across the enterprise.

With these factors in mind, the approach IBM has taken for meeting these implementation requirements for clients is a federated data science foundation serving as the runtime for the IBM Machine Learning for z/OS capability. The runtime is supported by the IBM z/OS Platform for Apache Spark, which was made generally available in March 2016 (see Figure 2).

This implementation of Apache Spark preserves a job's interaction with the runtime consistently with it on any other platform. However, the integration between Apache Spark on z/OS and data environments is differentiated and highly optimized. In Figure 3, key to the implementation is the ability to federate data from a variety of sources while preserving "data-in-place."

Data Advantage

This federated approach, leveraging current data, is atypical of most distributed implementations of Apache Spark that are built on top of Hadoop Distributed File Systems populated with extracted, transformed and loaded data which can be multihour or multiday latent. The differentiation for data access and integration is achieved through the optimized data integration layer, a component unique to the z/OS Platform for Apache Spark implementation.

With this approach, analytics using z/OS Platform for Apache Spark as the runtime can issue SparkSQL to most current client, transaction and merchant data that exists on a multitude of environments, blend in access to off-platform data, and produce insight that is consumable for business decisions in or near real-time. Efficiency of execution is achieved through a variety of mechanisms.

First, from the point of view of impact to cost, the entire environment is eligible for IBM z Systems* Integrated Information Processor (zIIP), averaging mid-90 percent utilization on zIIP, resulting in very little impact to other software pricing. From a skills perspective, there's efficiency in the ability to leverage industry-standard skillsets that are familiar with open technologies such as Apache Spark. From a complexity standpoint, z/OS Platform for Apache Spark installation and configuration is aligned with typical z/OS products, while the application usability is aligned with industry Spark skills. In addition, because z/OS excels as a multiworkload environment, the impact to other workloads can be minimized through specifying "dirty read" for data to avoid contention, and through Workload Manager enhancements within z/OS, to cap zIIP resources allotted to certain workloads in order to minimize cross-workload impact.

From a security perspective, there are the obvious advantages to leaving data in place and preserving the security associated with the originating source of data. However, in addition to that, clients have the flexibility to determine and configure what "view" of data they wish to surface for machine learning.

In one example, an organization may configure a view of client information that removes personally identifiable information such as a Social Security number. The data scientist consuming this data through IBM Machine Learning for z/OS can still perform the full scope analysis, but the information isn't proliferated. The ability to achieve this kind of obfuscation through configuration is a unique capability in the implementation of the z/OS Platform for Apache Spark.

Support Expands

As the functionality for IBM Machine Learning for z/OS grows, the z Systems platform will expand the scope of foundational platform support that's optimized for enterprise environments. In February 2017, IBM announced a joint partnership with Continuum Analytics and Rocket Software to make Python and Anaconda capabilities available as part of the foundation which IBM Machine Learning for z/OS and other solutions can leverage to deliver optimal value for clients.

Bringing the capabilities of IBM Machine Learning to the mainframe environment enables enterprise clients to obtain critical business insights from most current data while leveraging the security and efficiency benefits of data-in-place analytics as well as the expanded base of a broad set of data science skills.

Barbara Neumann contributed to the technical and editorial review of this article.

About the author

Jean-Francois Puget is an IBM Distinguished Engineer and IBM Machine Learning technical lead for public and private cloud divisions.

Mythili Venkatakrishnan is an IBM Senior Technical Staff Member and is the z Systems Architecture and Technology Lead for analytics.