Dell® Precision Cognitive Technologies PoV

Technical White Paper

Presenting new solutions to historically complex problems

Ran Taig, PhD – Senior Data Scientist – Dell IT Data Science Solutions

Shiri Gaber – Data Scientist – Dell IT Data Science Solutions

Marc Hammons – Software Architect – Client CTO Office

Tyler Cox – Software Architect – Client CTO Office

Contents

Introduction

The Dell Precision 7920 Tower Performance Class Fixed Workstation is engineered to provide a cost effective platform for development and deployment of cognitive technologies. With a versatile lineup of CPU, GPU, memory, storage, and networking configurations it offers flexible and scalable options suitable to multiple data science activities making it a data science platform of choice.

This paper examines two use cases realized on the Dell Precision 7920 Tower by the Dell IT Data Science solutions team. These use cases exercised various components of the workstation configuration, utilizing powerful CPU and GPU configurations to achieve iterative improvements while developing data models for predictive analytics and image classification. Without the power of the 7920 Tower these iterations might have taken days or weeks to complete.

Background

Facing the challenging use cases of the new data era, a data scientist knows the importance of a powerful machine to support daily tasks with data analytics, processing, modelling, and visualizations.

When building an environment for any data science team, the emergence of deep learning methods and state-of-the-art tools puts powerful computing capabilities as the essential foundation of any data science platform. The new Dell Precision Performance Class Fixed Workstation (7920 Tower) answers the computing power challenges that arise as a result of this new and innovative set of technologies and the ever increasing volume of data available for analysis.

The Dell IT Data Science team extracts valuable insights from the data collected and managed by Dell Technologies. Whether analyzing customer and sales data or evaluating events captured from Dell consumer and enterprise machines around the world, it is always a challenge to choose the best path to turn that data into true business value.

Our team was fortunate enough to obtain a 7920 Tower as a platform for the point of view (PoV) presented herein. The goal of this PoV was to demonstrate the power of the 7920 Tower for machine learning and data science use cases with a practical application. We chose to explore two high interest use cases based upon open data sets available today to the data science community.

The first use case is an image multi-classification task, based upon a data set released as part of the recent "Intel & MobileODT Cervical Cancer Screening" [1] competition. This computationally demanding task was chosen to showcase the raw GPU power of the workstation, as well as its scalable CPU and memory capacity for model training.

The second use case comes from one of the most valuable use cases for the IT industry: predicting hard drive failures. This use case is part of the emerging category of predictive maintenance which is becoming an important safeguard in the modern data center. The huge amount of log data collected is being utilized to identify bottlenecks in data flow, find root cause analysis of hardware and software malfunction, and to predict component failures well ahead of time, reducing or eliminating down time for customers.

This use case is based upon the open dataset updated yearly by Backblaze, Inc.[2] The Backblaze® data set contains daily S.M.A.R.T attribute collections from as many as 50,000 HDDs installed in Backblaze's data centers. This use case illustrates how the huge amount of RAM, disk memory and CPU power allowed us to build our entire data science workflow on top of the 7920 Tower platform.

Next, we explore the use cases in more detail.

[1] https://www.kaggle.com/c/intel-mobileodt-cervical-cancer-screening. We thank MobileODT®, Intel®, and Kaggle™ for releasing the data and for the organization of the competition.
[2] https://www.backblaze.com/b2/hard-drive-test-data.html. We thank Backblaze for releasing this data.

Cervical Classification

Cervical cancer is the fourth most common cancer in women, and the seventh overall, with an approximate 528,000 new cases in 2012. A large majority, around 85%, occur in the less developed parts of the world, where it accounts for almost 12% of all female cancers. In high-risk regions, an estimated age-standardized rate (ASR) over 30 per 100,000 occur in Eastern Africa (42.7), Melanesia (33.3), Southern Africa (31.5), and Middle Africa (30.6). Rates are lowest in Australia/New Zealand (5.5) and Western Asia (4.4). There were an approximate 266,000 deaths from cervical cancer worldwide in 2012, accounting for 7.5% of all female cancer deaths.

Early detection of cervical cancer is highly important. In fact, if it is diagnosed in its pre-cancerous stage and is given a suitable and quick treatment, the patient recovery is close to certain.

Choosing the suitable course of treatment for cervical cancer is dependent in the physiological shape of the cervix. Determining the type of cervical shape is done using imaging and analysis by a trained healthcare provider. In rural populations, the healthcare provider can often determine that treatment is required but may not have the skill to correctly determine the shape of the cervix and thus the appropriate course of treatment.

A technology that can assist the healthcare professional in providing an accurate cervical classification for this specific purpose can be a real game changer, allowing healthcare providers to provide the correct treatment to women around the world, including in those rural areas and countries in which cervical cancer is most common.

We thought this would be a great opportunity to illustrate the power of machine learning to assist with a real world healthcare need and as such developed a classification model using the Dell Precision 7920 Tower.

Using Deep Learning on Medical Imaging

The evolution of hardware to support the creation of deep neural networks, coupled with tremendous research in the deep learning field has provided an opportunity to analyze vast amounts of data using deep learning. Naturally, one of the fields benefiting from significant advancements in these new technologies is healthcare.

In many cases machine learning practitioners are saving lives when they create easy to use applications for physicians and other healthcare professionals powered by deep learning models. One area of interest in this vertical is the processing of medical imaging. Deep learning models can be used to assist with analysis of different pathologies based upon medical imaging, resulting in a more efficient and accurate treatment during the early stages of disease, giving patients a higher chance of recovery.

This creates significant opportunities for innovation in healthcare. In addition to the innovative tools available to medical staff, these technologies are now being developed to run on mobile devices, making medical diagnosis inexpensive and readily available in remote areas, particularly in developing countries where millions of people suffer from the lack of modern medicine infrastructure and a shortage of healthcare professionals.

We trained a deep learning model in a multi-class image classification task over the medical high-resolution cervical images dataset to demonstrate the Dell Precision 7920 Tower deep network training

capabilities. The data was released as part of a recent data science competition, established as a partnership between MobileODT and Intel. MobileODT has developed the Enhanced Visual Assessment (EVA) System, a digital toolkit for health care workers of every level to provide expert services to patients, anchored at the point-of-care by an FDA-approved, intelligent, mobile-phone based medical device.

Multi-Class Image Classification

For this use case we set out to deliver a model trained on ~1,600 high resolution cervical images from different patients, each labeled by one of the three cervical types. The resulting model would then be capable of classifying new cervical images by their physiological shape into one of the three different types.

While there are many classification algorithms with some very good implementations for use by data scientists, a deep learning approach was an obvious choice for this task.

Neural networks have become the state-of-the-art in image classification task over the last few years. Multiple, well-known neural network architectures have been released by industry practitioners and academic researchers along with open datasets used to maintain to improve the models.

However, deep learning model training and image processing in particular are computationally exhaustive tasks. State-of-the-art neural network model architectures tend to be deep and complex. Training these networks, e.g. the process of calculating the coefficient matrix that underlies the output model based on repeated iterations over the training data fed into the network, is the most-demanding task in the learning process. Training requires large scale matrix multiplication and the computation of derivatives of complex, high-order equations.

As such, the deployed hardware and software infrastructure can become critical performance bottlenecks when training a deep networks.

For each new use case, a data scientist performs a complicated, iterative R&D process, working interactively with the data and the trained model over and over. In this process, tools and infrastructure can become a real bottleneck. Data pre-processing and parameter optimization are just an example of two tasks in this pipeline. Training a complex network on sub-optimal infrastructure can take days to complete. The time and effort involved in fine tuning such a model can add up quickly when tools and infrastructure are insufficient and often leads to frustration and delay.

The 7920 Tower is an extremely attractive platform for multiple deep learning tasks. For our PoV, pre-processing the images was enabled by the abundance of memory and power of the two Intel® Xeon® CPUs. Additionally, the NVIDIA® Quadro® GP100 card was essential for iteratively optimizing the neural network and delivering the target model.

Challenges

The use case required dealing with two non-trivial computational challenges:

- Data preprocessing – The test and training sets are composed of ~1,600 high-resolution images (3-6MB per image for 7.2GB total). Resizing the images to find the optimal neural network is a resource demanding task that is efficiently and easily handled by the scalability of the 7920 Tower.

- Grid Search for neural network model optimization – To determine the best neural network architecture for the task and then fine tune that network for optimal performance is a complex process which requires repetitive training of the model with different settings. The 7920 Tower allowed a substantial reduction in training time from what is typically days and hours to hours and minutes, making the process smooth, efficient, and time effective.

Neural Network Model

As stated, we experimented with different known architectures for image classification and finalized on a common variation of a convolutional deep neural network (CNN): ResNet, a residual CNN, based on the following resources:

Theory:

1. He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. "Deep Residual Learning for Image Recognition." In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770-778. 2016.
2. Srivastava, Rupesh K., Klaus Greff, and Jürgen Schmidhuber. "Training Very Deep Networks." In *Advances in Neural Information Processing Systems*, pp. 2377-2385. 2015.

For basis of Implementation we used the Python notebook in the article “Complete process using ResNet as a starting point” released by Rodney Thomas at Kaggle. [3] For further background on residual CNNs refer to the background section on cognitive technologies below.

[3] https://www.kaggle.com/vasilkor/complete-process-using-resnet-as-a-starting-point/notebook/notebook

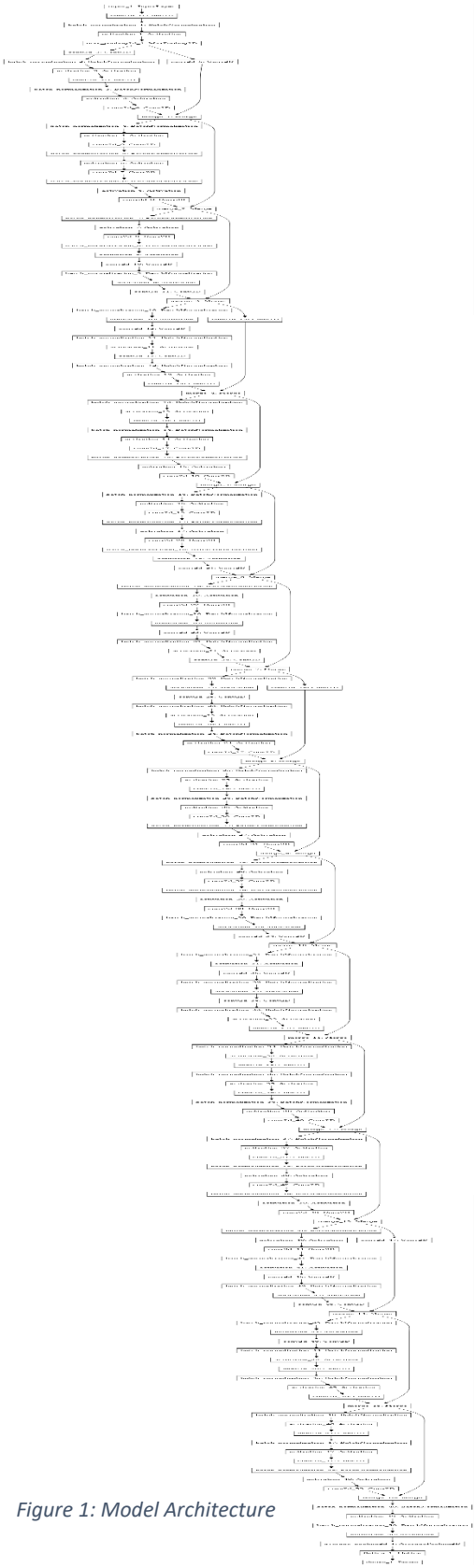
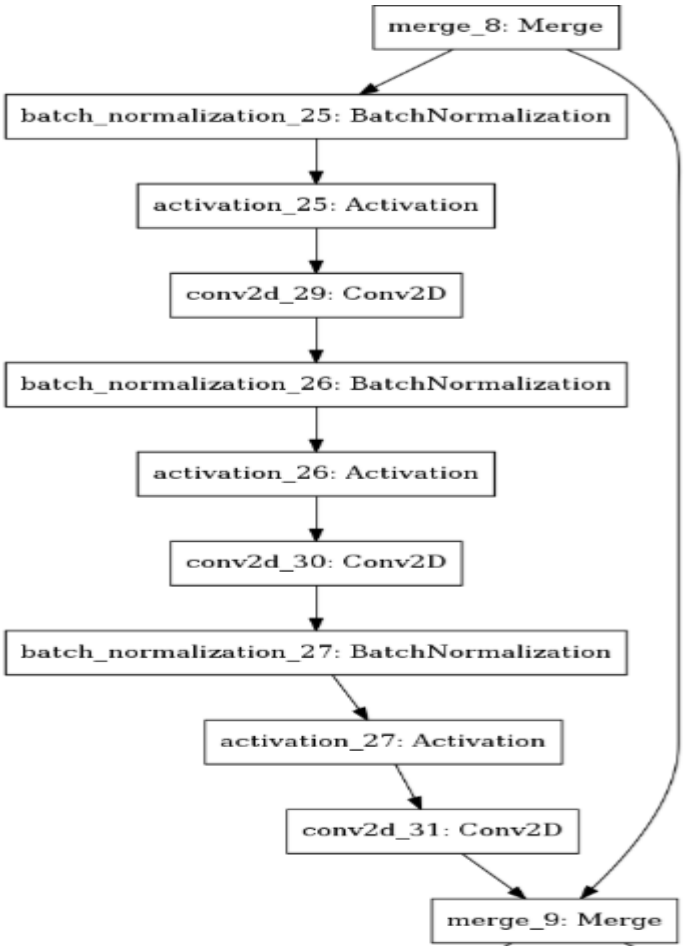


Figure 1: Model Architecture

The best residual-CNN we were able to train on the machine is a 50-convolutional mini network r-CNN, the final architecture is presented in the figure on the left.

The network is composed of 50 repetitions of following mini-CNNs, each equipped with a “highway” edge allowing the identity function flow straight from the input to the output per the rational of residual CNNs:



The network was optimized with respect to the loss function of categorical cross entropy, commonly used for classification optimization. The optimizer algorithm used is ‘Adam’ with learning rate of 0.001 and 0 decay rate.

While the input (RGB) image’s common original sizes were: 3264x2448 or 4128x3096 we needed only moderate down sampling to 2080x2064 before feeding it to the model.

The fact that the NVIDIA Quadro GP100 GPU internal memory allowed us to build a network which supports such a quality image has tremendous effect on the accuracy of predictions. We find it to be the most important factor compared to using bigger mini-batches or deeper networks.

Results

We note that our results were achieved with a simple R&D process as this effort was developed for demonstration only. With more development time it could certainly be further optimized. Great results are easier to achieve when using a deep network with quality input on a powerful platform.

As part of this R&D process the data pre-processing, number of mini-CNNs, epoch, and batch size went through several iterations, resulting in varying training times, accuracy, and loss results. For comparison some of that information has been captured below.

Table 1: Neural Network Training Variants

#mini-CNNs	#Epochs	Batch size	Resizing images to	Training time	Accuracy	Loss
50	**50**	**1**	**2080*2064**	**4:36 hrs**	**0.83**	**0.83**
151	15	2	1024X1024	3:20 hrs	0.732	1.623
101	20	7	512x512	9:32 hrs	0.51	0.92
50	15	1	1024X1024	0:57 hrs	0.83	0.94
50	50	3	256*256	1:22 hrs	0.21	1.81
101	50	2	1024*1024	6:54 hrs	0.73	1.62

The final resulting network minimized the loss to 0.83 while maintaining classification accuracy of 83%.

Table 2: Final Neural Network Statistics

Network Parameters	23,586,691
Training and Validation Samples	1,111
Test Samples	512
Mini-Batch Size	1
Epoch Count	50
GPU Training Time	4 hours, 36 minutes
Neural Network Layers	170

Hard Drive Failure Prediction

Predictive maintenance techniques are designed to help determine the condition of in-service equipment in order to predict when maintenance should be performed. This approach promises cost savings over routine or time-based preventive maintenance, as tasks are performed only when warranted.

The promise of predictive maintenance allows convenient scheduling of corrective maintenance, and to prevent unexpected equipment failures and data loss. The key is "the right information in the right time". Determining which equipment needs maintenance, maintenance work can be better planned (spare parts, people, etc.) and what would have been "unplanned stops" are transformed to shorter and fewer "planned stops", thus increasing uptime and availability. Other potential advantages include

increased equipment lifetime, increased plant safety, fewer accidents with negative impact on environment, and optimized spare parts handling.

Predictive maintenance differs from its cousin preventive maintenance because it relies on the actual condition of equipment, rather than average or expected life statistics, to predict when maintenance will be required.

Modeling Hard Drive Failures

High reliability of storage systems is extremely important for large-scale IT systems, and especially for high-performance computing systems where data must not only be highly accessible, but the data itself is irreplaceable if lost. Modern installations are in fact clusters of thousands of storage installations capable of maintaining unprecedented demands. However, HDDs used within these networks can and will fail over time. HDDs are considered less reliable than SSDs due to the presence of moving mechanical parts (Hu, Eleftheriou, Haas, Iliadis, & Pletka, 2009), however HDDs are still more commonly used in the industry due to the higher costs of SSDs.

Predictive failure analysis refers to the monitoring and "learning" process of corrected error trends in order to predict future failures and issue a failure alert before it occurs. In recent years, this field has been heavily studied both in academia and industry, given the extensive use of hard drives and high return on investment for an accurate implementable predictive model. An accurate model capable of generating early warnings can enhance the backup policy, allow for the improvement of maintenance policies, and improve the general customer satisfaction of a service provider.

The most commonly used statistical measures for representing the reliability of a drive are the annualized failure rate (AFR), which is the average percentage of disk drives in a population that fail within one year and mean time to failure (MTTF), which is the number of power on hours per year divided by the AFR. These statistics have been the subject of many papers; as their name suggest, these concern the average characteristics of population of drives. As in (Schroeder & Gibson, 2007), they usually suggest that the datasheet values denoted by the manufacturer are far off from what is actually observed in the field. The designated statistics pertain to the drive's functionality under controlled lab environments (accelerated life and stress tests) that do not always apply to real production environments conditions and workloads. In addition, many of the studies assume independency of different drive failures, a fact not trivial when it comes to clusters of storage devices containing thousands of drives as is the case in many storage configurations.

As the aforementioned failure metrics are static for the entire population, often studies will also list those factors highly correlated with a drive failure; these are variables that will, on average, present higher values among drives facing impending failure. For example (Pinheiro, Weber, & Barroso, 2007), which collected data regarding the environment, utilization, error, configuration and repair events, concluded that the drive age may be highly correlated with a drive failure, but temperature and activity levels are much less correlated with drive failures than previously reported.

In 1995, the drive industry adopted SMART, self-monitoring, analysis and reporting technology (Hughes, Murray, Kreutz-Delgado, & Elkan, 2002). SMART is a monitoring system installed in computer hard disk drives (HDDs) and solid-state drives (SSDs) that collects and reports on various indicators of drive reliability. This technology uses attributes collected during normal operation (and during off-line tests) such as reallocation sectors count, reported uncorrectable errors, power-on hours, read-write errors

and more to set a failure prediction flag. Backblaze, an online backup company, released its drive statistics in a series of blogs (One Billion Drive Hours and Counting: Q1 2016 Hard Drive Stats) which found that some drive stats are highly correlated with drive failures, i.e. once these are detected the probability of the drive to fail increases significantly.

In most host systems today, an alarm is set whenever a drive exceeds vendor defined threshold for single attributes. These thresholds tend to be highly conservative due to the expenses related to sending a drive back to the manufacturer for warranty replacements. This price effectively increases due to the low proportion of failed drive, a ~1% annual failure rate (AFR) for most drive families; A false alarm rate (FAR) of as little as 0.2% of total drives per year often implies a comparable number of falsely and justly classified failed drives, a price most vendors avoid undertaking (Hughes, Murray, Kreutz-Delgado, & Elkan, 2002). As a results, industry-used thresholds implemented provide 3-10% detection with as low as 0.1% false alarm rate (FAR).

Predictive Statistical Analysis

We used a predictive random forest model developed on the 7920 Tower. The process included using the Backblaze 2016 dataset (6GB, 23,326,401 rows, ~50K unique HDDs) and processing it through the full data science cycle on a single workstation; no need for clusters, big data platforms, etc. With its huge storage capacity, memory, and high core count CPUs the 7920 Tower powers through all that you need for an efficient process on a local Jupyter notebook. This includes a non-trivial extraction of time series features based on the linear regression that ran on the ~50K different drives in the data.

Data features and Distributions

The Backblaze dataset contains SMART measurements collected daily over a period of 1 year from approximately 50,000 drives in their data center. Since each drive manufacturer collects different SMART attributes the developed model was only applied on Seagate® drives which count for almost 50% of the total number of drives, but it could easily be modified to apply on other types of drives as well.

The data contains ~100 individual raw and normalized SMART attributes with basic drive information such as drive model, serial number, etc. Backblaze references the five following SMART attributes [4] to determine if a drive is about to fail: SMART 5 (Reallocated Sectors Count), SMART 187 (Reported Uncorrectable errors), SMART 188 (Command Timeout), SMART 197 (Current Pending Sector Count) and SMART 198 (Uncorrectable Sector Count). Once the raw value of one of these SMART attributes is different than zero, Backblaze issues an alert to examine the drive.

Out of the entire population, 90% of (operational) drives are sampled throughout the year (~300 samples per drive). 2% of the drives fail at some point during the year of the study. In the training period, a sample is considered as positive if the corresponding drive has failed within the next two weeks following the date in which the sample was taken.

[4] https://www.backblaze.com/blog/what-smart-stats-indicate-hard-drive-failures

Challenges

Typically, within a data science lifecycle these stages are the most time-consuming and complex parts.

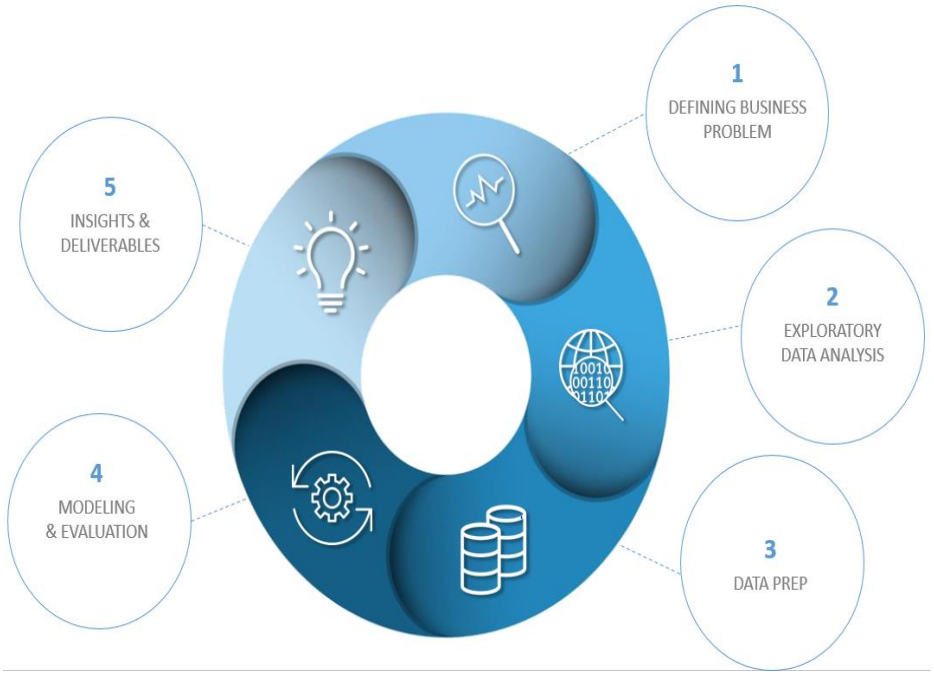


Figure 2: The Data Science Life Cycle

For this process to be efficient, the ability to have quick interaction with the data and easily visualize different features of it is essential. The process typically includes data consolidation, importing data, exploratory data analysis, identifying data issues (corrupted data, missing values, etc.), and modify the data to address them. When the raw data goes behind a very limited size (~2GB – very much system dependent) local, interactive work with becomes quite painful. The solution, in most of these cases is to move the work to a big data platform, allowing distributed execution of code between different clusters.

While the frameworks for big data develop constantly, offering well-built distributed implementation of common functions, it still requires non-trivial effort in many cases:

- Although some frameworks offer native language support today, e.g. SQL on Greenplum® or Python on PySpark, many of the standard packages like pandas for the preprocessing tasks lack sufficient distributed implementations for non-trivial aggregations and data processing functions. The memory requirements for objects like pandas' data frames are enormous (up-to 6-7x of the raw data size) and consume standard memory quite quickly.
- Price and availability: A big data platform requires complex infrastructure for deploying the clusters in-network or a data science team has to utilize cloud-based clusters for their work. In both cases these solutions are expensive for constant use and require knowledge and IT professionals for deployment and maintenance.

The memory capacity of the 7920 Tower and the parallelization power of its Intel Xeon CPUs allowed us to execute the data science cycle locally, manipulating the 6GB raw data, parsing dates, filling of missing

values, feature engineering and modelling itself without any need of external resources and in a matter of minutes. While the Python Imaging Library is not optimized for parallelization, we used the multi-processing package and process pooling to run computations of different parts of the data frame in parallel.

The ability to use our full standard toolbox and the interactive environment of a local Jupyter notebook saved time, increased creativity and flexibility in the development and overall offered much convenience.

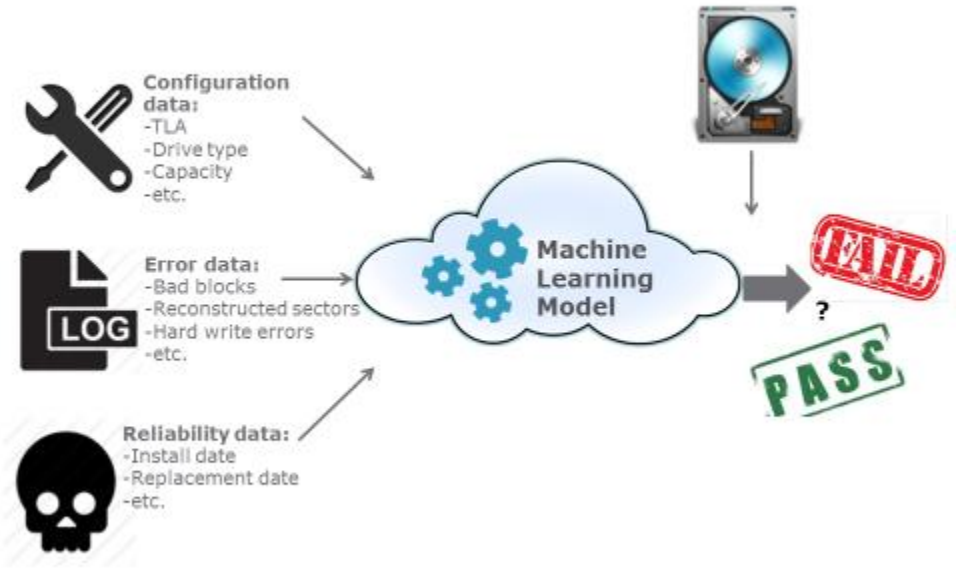


Figure 3: Overview of the modelling approach

Modelling Approach

As the main goal of this PoV was to demonstrate the machine performance over CPU intensive tasks we did not perform much research and optimization of the model as most of the intensive tasks were around treating and preparing the data as described in the previous section.

We ran a standard feature engineering process, extracted basic aggregated SMART values to capture the dynamic nature of SMART attributes which was suggested to correlate into drive failures. We also ran model selection and a grid search over the random forest (selected model) parameters. In addition we used stratified sampling for the train/test split as there were only 2% of failed drives in the data e.g. classes were significantly imbalanced.

As before, all of the process was done easily and efficiently using the 7920 Tower. The final data set consisted of 53,000 rows (each row representing a unique drive). Training the models took only a few seconds.

Results

Data science always works with the business and for the business. As such, the model goals should be well agreed upon with business objectives in mind from the beginning of the data science process. Consulting with a Dell storage subject matter expert (SME) we worked under the following assumption: 10 days in advance is a sufficient period in which to deliver an alert regarding an eminent drive failure. This gives an operator enough time to schedule maintenance or leverage routine visits for replacement of the suspected drive.

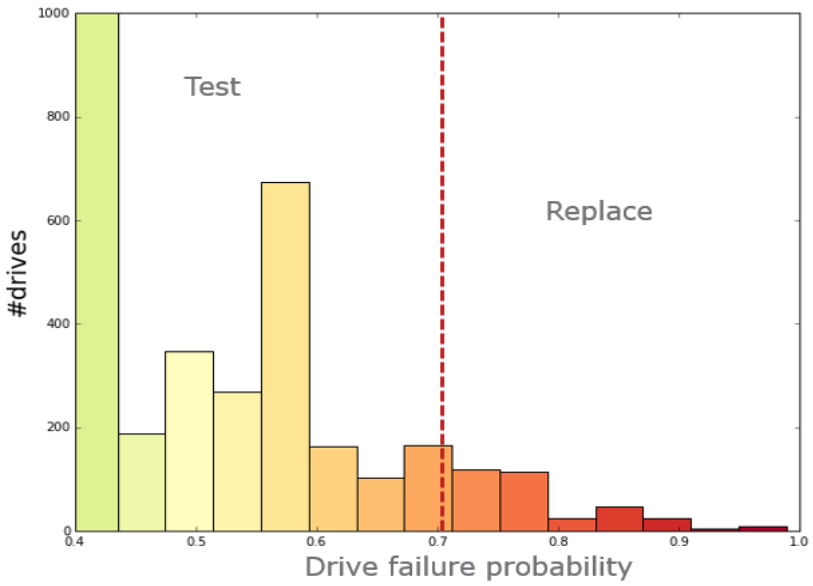


Figure 4: Example of model output - allows an SME to set the confidence threshold

Model evaluation and fine-tuning always requires a balance. Which case is worse, incorrectly predicting a healthy drive as a candidate for failure or improperly assessing a drive with an eminent failure as healthy? For this use case it was balancing the price of shipping a new drive, scheduling a special visit, and replacing a drive that would have stayed healthy for a long time.

Thus, we need to reduce and optimize the false positive rate, i.e. reduce the number of healthy drives improperly predicted to fail. To achieve this, we allow the user to define the confidence threshold used to classify a drive as failed while running predictions on its data.

We were able to achieve, with the limited scope of optimization we performed and a reasonable, 0.7 confidence threshold, the following results:

- Precision: 0.96 – 96% of the drives predicted to stay healthy stayed healthy.
- Recall: 0.42% - The portion of drive predicted to fail from the total numbers of drives failed.
- False Positive Rate (FPR): 0.002%

Although not optimal, for an almost out-of-the-box model and given the optimization goals these results reveal the potential of predicting HDD failures and predictive maintenance in general.

Cognitive Technologies

The following section presents a brief overview of some cognitive technologies for the purposes of discussion and categorization. As this is a wide and rapidly growing field, this section does not come close to a comprehensive list, rather just offers a few suggestions of algorithms and applications that might be run on a 7920 Tower workstation. There are many, many more.

Some of these algorithms might be executed on CPU and/or GPU. Other algorithms are better suited to utilization of a specific computing resource, e.g. using the GPU to train a neural network for the purposes of image classification. Often these choices are dependent upon multiple factors; the task at hand, desired software frameworks, ability to parallelize the workload, computing, storage and other resources available, experience and background of the data scientist. As a platform for cognitive technologies the 7920 Tower is versatile enough to support multiple desired variants and approaches to solving data science problems.

Rule Based Analytics

Rule based analytics, sometimes referred to as complex event processing (CEP) uses pre-defined rules or cases as a set of instructions that guide a decision making process. Generally an engineer and/or knowledgeable craftsperson with domain expertise derives rules that represent cause and effect relationships with defined outcomes. Rule based systems are quite common, well understood, and widely deployed, often in conjunction with more complex forms of cognitive technologies. Rule based analytics typically execute using the CPU of the workstation and may use input from more advanced forms of cognitive technologies in their decision making process.

Machine Learning (ML)

Machine learning generally includes well-known algorithms such as linear regression, logistic regression, support vector machines (SVM), naïve Bayes, random forest, KNN, etc. These algorithms might be run on CPU or GPU depending on the case and are typically used for prediction and/or classification.

ML for predictive analytics use statistical algorithms to find a complex function that will appropriately predict a result using an existing data set as a basis for deriving the function. Once derived, feeding new data into the function yields a desired accuracy level of prediction, or confidence. An example of this would be finding a function that predicts hard drive failures. In the Backblaze use case presented above, the hard drive failure prediction model was created using a random forest-based predictive algorithm. We chose to develop this model on the CPUs of the 7920 Tower.

Deep Learning (DL)

Deep learning is the process of determining the significant data features given a large data set using a multi-level, complex neural network. Deep learning involves large amounts of data crunching with lots of matrix multiplications and is best suited to GPUs or other purpose built silicon. Due to the large number of features present in the data and the training set size, the training phase of deep learning requires a large amount of computation power in order to yield a timely result.

Neural networks used in deep learning come in many variants; CNNs, RNN, R-CNNs, Deep Residual Learning, etc. with new algorithms becoming available as research in this area forges ahead. We decided upon a residual convolutional neural network (ResNet) for the cervical classification use case and used the NVIDIA Quadro GP100 to develop our model.

Residual CNNs

Deep residual learning is a relatively new and innovative method for training neural networks that are substantially deep where under fitting and optimization can be difficult. Residual networks (ResNets) are composed from repetitive mini-convolutional neural network (mini-CNN) modules. The principle of residuals allows one to add a new module only if there is an extracted value out of adding that module.

The way to make this happen is to also provide the next mini-CNN output its own input without any transformation (identity). Intuitively, this triggers the next module to learn something different from what the input has already learned. The other advantage in establishing these connections is to deal with the vanishing gradient problem in very deep networks.

Reinforcement Learning (RL)

Reinforcement learning is an area of ML in which the algorithm seeks to maximize its rewards based upon its behavior. Q-learning, Double-Q learning, Sarsa, Temporal differences are examples of reinforcement learning algorithms. RL is used when information about the environment may be limited and actions must be learned by interacting with the available environment, i.e. trial and error is required.

Data Science Software Tools

There has been an explosion of software frameworks around cognitive technologies in general, with updates and extensions occurring at a rapid pace. Some of the more popular ones include TensorFlow™, Caffe, MXNet, Torch, Theano, Microsoft® Cognitive Toolkit (formerly CNTK) … the list goes on and on, and that doesn't include the massive list of supporting libraries and utilities. Below we outline some of the frameworks, libraries, tools and utilities that were used in creating this particular PoV.

NVIDIA Docker

To allow quick onboarding of our toolkit on the 7920 Tower, while allowing full utilization of the machine's power; GPU, CPU, memory, and storage, we chose to build our environment on top of a dedicated Docker® container running on the Nvidia-Docker engine.[5] This practice is highly recommended when an independent software environment on a remote machine is desirable. Nvidia-Docker is easily installed and abstracts configuration of the GPUs of the machine. It then functions as an out-of-the-box mediator between the development environments on top of Docker to the hardware layer, allowing the machine learning practitioner more time to focus on what's important, the R&D! Additionally, using Docker as a base allows for a seamless transition between development and production installations.

Jupyter Notebooks

For development we used Jupyter notebooks with R and Python 2.7 kernels in the backend.

Python, Python, and Python

The data processing and cleansing were performed with the standard data science Python packages for this purpose, e.g.: pandas, NumPy, SciPy, and Matplotlib.

[5] https://github.com/NVIDIA/nvidia-docker

TensorFlow and Keras

The cervical classification use case required deep learning development packages. For this case we chose to work with Keras with a TensorFlow 1.2 backend. Motivated by the rich documentation and ease of use, the Dell IT Data Science teams currently favor this tool set for deep learning development. Additionally, OpenCV was used to precondition and resize the images to efficiently fit into memory.

Pandas, Dask, and Scikit Learn®

For the Predictive Hard Drive Failure CPU use case we had to use workload parallelization in order to support working locally with the amount of data (~6GB CSV per year of data) and to allow us to fully utilize the power offered by the 7920 Tower's pool of cores. We also utilized the relatively new Dask package for that same matter. During the statistical modeling stage (feature engineering, grid search, model selection, training and predicting) we used Scikit Learn 0.18.

System Resources

Since the 1997 launch of the Dell Precision line, Dell has committed to providing our customers with cutting-edge optimized solutions to their most challenging problems. Below we will take a look at the 7920 Tower configuration and some of its features as can be applied to problems in the cognitive technologies domain.

Platform	Dell Precision 7920 Tower Fixed Workstation
Processor(s)	2 x Intel Xeon SkyLake-SP up to 28 cores
GPU	2 x NVIDIA Quadro GP100 (3584 FP32 cores)
Memory	512GB DDR4 @ 2667Mhz
Storage	4 x 1TB SSD (RAID 0)
OS	Ubuntu 14.04 LTS

CPU

As has been discussed in previous publications, [6], [7] selecting the correct CPU for your use case can be a complex task. For Data Science workloads, the right CPU depends on your targeted use cases. For classic machine learning and moderate neural networks, using software frameworks that are integrated with the Intel® Math Kernel Library (MKL) make the CPU an excellent option upon which to run machine learning workloads.

At the heart of the 7920 Tower, the Intel Xeon scalable performance architecture, SkyLake-SP, offers up to 28 cores, 56 threads, 6 channels of DDR4 at 2666 MHz and 48 Lanes of PCIe Gen 3.

SkyLake-SP is a descendant of the client core, but has a larger 1 MB L2 cache with a non-inclusive L3 cache that has more granular 1.375 MB slices. It also includes AVX-512 support with 512 bit-wide vector units. Additionally there is a new on-die mesh fabric that is used to connect the cores and the L3-cache slices together.

[6] http://i.dell.com/sites/content/business/smb/sb360/en/Documents/wp-workstation-chossing-he-right-cpu.pdf
[7] http://i.dell.com/sites/doccontent/shared-content/data-sheets/en/Documents/Whitepaper_Selecting_the_Right_Workstation.pdf

The memory and I/O controllers have also been updated to provide greater performance and connectivity. SkyLake-SP also has the new Ultra Path coherent interconnect (UPI) interface that is faster and more efficient than the QPI interconnect.

Core Count vs Frequency

Often people look at core count and frequency and wonder which is more important.

The number of cores required for a workload depends upon two things: the parallelism of the workload and the desired number of concurrent operations. If the software that supports the application isn't written to take advantage of multiple cores, then there is little benefit to having them in place. In short, today most software still has to be written correctly in order to take advantage of multiple cores. Software frameworks are making big strides in this area to achieve automated parallelism when possible, but there is still much room for improvement.

We found that TensorFlow was able to distribute a workload across CPU and GPU cores quite well, achieving a high occupancy rate for our compute resources when training a model for our cervical classification use case.

- CPU-based training
  - CPU occupancy (active cores) – 71.67% avg, 18.07% min, 91.33% max
  - System memory occupancy – 70.31%
  - GPU occupancy – 1%
  - GPU memory occupancy – 0%
- GPU-based training (single NVIDIA® Quadro GP100)
  - CPU occupancy (active cores) – 2.34%
  - System memory occupancy – 12.32%
  - GPU occupancy – 94% avg, 86% min, 99% max
  - GPU memory occupancy – 100%

Conversely, we found that for Backblaze use case there were some issues with multithreading of the underlying Python libraries and we weren't able to use all of the cores effectively during all of the stages of model production - date separation and sorting, data loading, model generation.

- CPU-based training
  - CPU occupancy (active cores) – 9.29%
  - CPU memory– 6.51%
  - GPU occupancy – 1%
  - GPU memory occupancy – 0%

In cases where there are some sequential, blocking, non-scalable operations a high frequency processor might be a good option. Processor frequency is still a significant factor in performance and is paramount when the workload cannot be improved by the parallelism of a multi-core solution.

In addition, with Intel Turbo Boost Technology, the processor can clock above the published frequency as long as thermal and power are maintained below established thresholds, offering an even greater boost in performance.

GPU

The 7920 Tower is equipped to handle up to 3 double wide cards at 300W, leaving room for up to three NVIDIA Quadro GP100s as needed. Incorporated into our 7920 Tower were two NVIDIA Quadro GP100 cards for our deep learning needs, which we could use either independently or together depending upon how our application was written. The GP100 is powered by the new NVIDIA® Pascal® GPU architecture is capable of meeting the needs of the most demanding data science workflows.

Pascal GPU[8] is capable of delivering over 5 TeraFLOPS of FP32 performance for high precision workloads and over 20 TeraFLOPS for FP16 workloads, the most common need for today's advanced deep learning applications.

Compared to the current generation of GPU architectures on the market, the GP100 offers a 12x increase[9] in neural network training, reducing training time from weeks to hours and a 7X increase in deep learning inference throughput.

Workload

For our purposes we used the GP100 to train a moderately sized residual convolutional neural network. This is a good example of the type of deep learning application to which a GP100 is extremely well suited. Early in the process, the GP100 could train the network in around 3 hours with a moderately good confidence level, ~70% which allowed us to iterate quickly through the process of improvement. As the model was refined and improved, the time to train increased but so too did the confidence level in the classification, ultimately yielding a 6.2 hour training time with .83 loss while maintaining classification accuracy of 83% before we had to halt the efforts.

Multiple GPUs and NVIDIA® NVLink™

Our environment is set up with the NVIDIA NVLink [10] interconnect between our two NVIDIA Quadro GP100 cards. NVLink is a high-bandwidth connection enabling ultra-fast communication between connected GPUs (and GPUs and CPUs as available). Since NVLink allows data sharing between these cards at 5 to 12 times faster than the PCIe Gen3 interface, application performance can double instantly for given workloads.

We found that our software application had to be written correctly in order to take advantage of this configuration. As developed, our image processing model only makes use of a single GP100 with acceptable performance and iteration time for our data science team. We are certainly interested in exploring the multi-GPU use case further.

There's a definite trade off to be made between the time that is spent to develop software that trains a model over multiple GPUs vs the time to develop software to train a model over a single GPU. The latter might be an easier path with acceptable results when dealing with moderate data sets and complexity. We were quite pleased with our results using the single GP100 given the depth of and complexity of our network. With recent advances in the ability of software frameworks to automatically parallelize

[8] https://images.nvidia.com/content/pdf/tesla/whitepaper/pascal-architecture-whitepaper.pdf
[9] http://www.nvidia.com/object/gpu-architecture.html
[10] http://www.nvidia.com/object/nvlink.html

workloads, the burden on the software developer to manually distribute the workload may be alleviated.

Memory

The 7920 Tower offers support for 6 channel DDR4 ECC at 2667 MHz with up to 24 RDIMM (32GB) for 768GB of memory or using LRDIMM up to a whopping 3 TB with select CPU models. Additionally the 7920 Tower offers support for NVDIMMs, the next generation in memory technology, allowing for power restoration of the machine's previous memory state from a non-volatile store.

Storage

The 7920 Tower offers an abundance of storage options, from high speed to high reliability and many options in between. Currently available are:

- Up to 10 bays for 3.5"/2.5" SATA / SAS
- Up to 4 x M.2/U.2 PCIe NVMe SSD with hot plug support
- Ultra-speed drive M.2 PCIe NVMe SSD

Storage Control

By default the 7920 Tower ships with integrated SATA3 6 Gb/s supporting SW RAID 0, 1, 5, 10. Optionally available are:

- Integrated Intel® VROC 32 Gb/s for PCIe NVME SSDs RAID 0,1,10 (RAID 10 with Ultra-speed)
- SATA3 6 Gb/s – SAS2 12 Gb/s – PCIe 32 Gb/s RAID 0, 1, 5, 10
- SATA3 6 Gb/s – SAS2 12 Gb/s – PCIe 32 Gb/s (4 GB cache & supercap) RAID 0, 1, 5, 10

We had available 4 x 1 TB SATA SSDs and used SW RAID 0 to stripe and make three of them appear as a single logical volume in Ubuntu® 14.04. Our data wasn't mission critical as it's readily downloadable from external sources, but we want to try out some more advanced options. We have a few additional storage solutions available to us and will be looking to determine the performance of them in the future. These include:

- 4 x 1.2 TB 10k SAS drives with HW accelerated RAID 10.
- 4 x 1 TB M.2 PCIe NVMe SSDs via an Ultra-Speed PCIe card with SW RAID 0 for aggregation.

U.2

U.2 is a new standard that is rapidly approaching implementation. It uses the same mechanical connection as the current enterprise SAS/SATA specification for backplanes, but allows new U.2 SSDs to directly utilize 4 PCIe lanes in a 2.5 inch drive. This allows for the communication bandwidth of an M.2 drive while enabling better cooling and higher capacity and maintaining compatibility with legacy enterprise standards. This also reduces the system complexity and maintenance concerns, allowing direct upgrades from HDDs to U.2 SSDs as appropriate.

Conclusion

With this PoV we've merely scratched the surface of data science applications for the Dell Precision 7920 Tower Performance Class Fixed Workstation. From standard machine learning algorithms to advanced deep and reinforcement learning pipelines the 7920 Tower is versatile enough to scale with the problem set at hand.

Need to perform simple regression models today? Choose from the wide variety of Intel Xeon CPUs, memory footprint, and storage options that are available. Need to use deep learning to train a complex neural network? Add multiple NVIDIA Quadro GP100s to the mix for blazing fast model training and inference. There is no end to the possibilities of machine learning and deep learning applications that can be realized on the 7920 Tower.

Dell Precision Workstations are a cost effective platform for development and deployment of cognitive technologies and solutions. When considering the costs of cloud infrastructure to support data science activities; network transmission time, large scale storage costs, hourly processing costs, operational knowledge and training, these items can add up quickly over multiple projects when having to iterate to find optimal solutions. The 7920 Tower allows unlimited use of its resources with the ability to scale up to sizeable workloads over time as well as choose the tools that a team needs for the tasks at hand.

With powerful Intel Xeon CPUs, up to 3 TB of memory, 10 drive bays for an abundance of storage or ultra-fast PCIe NVME speeds, room for 3 NVIDIA Quadro GP100 cards for deep learning applications, and up to 2 x 10 Gb/s network connectivity, the 7920 is a data science powerhouse, available in both tower and rack (2U) form factors.

We found that the 7920 Tower offers a wide variety of options to fulfill our data science needs, allowing for ML and DL applications that are emerging at all points along the continuum of IT infrastructure. Dell Technologies is well positioned to offer platform solutions along this entire continuum and the 7920 Tower is at the forefront of a Data Scientist's daily work routine.

From avoiding costly downtime and potential loss of mission critical data by predicting hard drive failures to constructing a neural network to assist a healthcare professional in finding an appropriate course of treatment for a patient, Dell Technologies can provide your data science team with the platform and tools to make a greater impact on the world.