TECHNICAL OVERVIEW

NVIDIA DEEP LEARNING PLATFORM

Giant Leaps in Performance and Efficiency for AI Services, From the Data Center to the Network's Edge

Introduction

Artificial intelligence (AI), the dream of computer scientists for over half a century, is no longer science fiction—it is already transforming every industry. AI is the use of computers to simulate human intelligence. AI amplifies our cognitive abilities—letting us solve problems where the complexity is too great, the information is incomplete, or the details are too subtle and require expert training.

While the machine learning field has been active for decades, deep learning (DL) has boomed over the last five years. In 2012, Alex Krizhevsky of the University of Toronto won the ImageNet image recognition competition using a deep neural network trained on NVIDIA GPUs—beating all the human expert algorithms that had been honed for decades. That same year, recognizing that larger networks can learn more, Stanford's Andrew Ng and NVIDIA Research teamed up to develop a method for training networks using large-scale GPU computing systems. These seminal papers sparked the "big bang" of modern AI, setting off a string of "superhuman" achievements. In 2015, Google and Microsoft both beat the best human score in the ImageNet challenge. In 2016, DeepMind's AlphaGo recorded its historic win over Go champion Lee Sedol and Microsoft achieved human parity in speech recognition.

GPUs have proven to be incredibly effective at solving some of the most complex problems in deep learning, and while the NVIDIA deep learning platform is the standard industry solution for training, its inferencing capability is not as widely understood. Some of the world's leading enterprises from the data center to the edge have built their inferencing solution on NVIDIA GPUs. Some examples include:

- **Twitter Periscope** runs inferencing on GPUs to understand video content in real-time, enabling more sophisticated video searches and user recommendations.
- **Pinterest** uses cloud-based GPUs to minimize user wait time (or latency) for its Related Pins service, delivering engaging recommendations based on users' interests.
- **JD.com** runs inference-based intelligent video analysis in real time on every frame of video of 1,000 HD video channels, and increased its per-server throughput by 20x.
- **iFLYTEK** switched to Tesla GPUs for its Mandarin speech recognition service in China, and is now able to handle 10x the number of concurrent requests, and reduced its operational TCO by 20%.
- **Cisco's Spark Board and Spark Room Kit**, powered by Jetson GPU, are re-inventing the meeting room, enabling wireless 4K video sharing, and using deep learning for voice and facial recognition, as well as enhancing resource planning.

NVIDIA deep learning platform spans from the data center to the network's edge. In this paper, we will describe how the platform delivers giant leaps in performance and efficiency, resulting in dramatic cost savings in the data center and power savings at the edge.

The Deep Learning Workflow

The two major operations from which deep learning produces insight are training and inference. While similar, there are significant differences. Training feeds examples of objects to be detected/recognized like animals, traffic signs, etc., allowing it to make predictions, as to what these objects are. The training process reinforces correct predictions and corrects the wrong ones. Once trained, a production neural network can achieve upwards of 90-98% correct results. "Inference" is the deployment of a trained network to evaluate new objects, and make predictions with similar predictive accuracy.

Figure 1: High-level deep learning workflow showing training, then followed by inference.

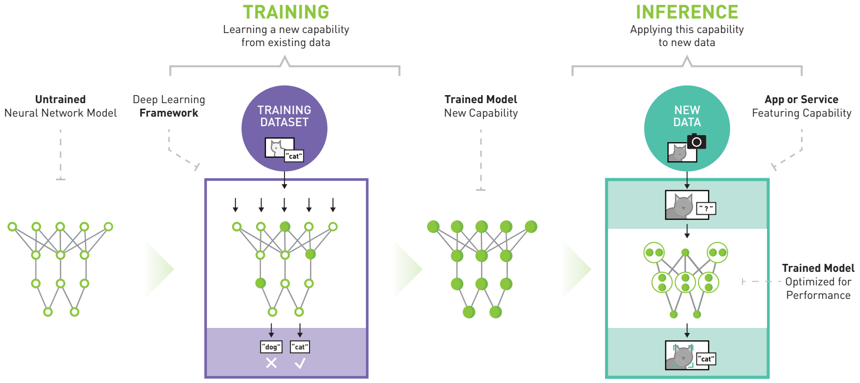

TRAINING
Learning a new capability from existing data
INFERENCE
Applying this capability to new data
Untrained Neural Network Model
Deep Learning Framework
TRAINING DATASET
"cat"
"dog"
"cat"
✕
✓
Trained Model New Capability
NEW DATA
"?"
"cat"
App or Service Featuring Capability
Trained Model Optimized for Performance

Figure 1

Both training and inference start with the forward propagation calculation, but training goes further. After forward propagation when training, the results from the forward propagation are compared against the (known) correct answer to compute an error value. A backward propagation phase propagates the error back through the network's layers and updates their weights using gradient descent to improve the network's performance on the task it is trying to learn. It is common to batch hundreds of training inputs (for example, images in an image classification network or spectrograms for speech recognition) and operate on them simultaneously during deep neural network (DNN) training to amortize loading weights from GPU memory across many inputs, increasing computational efficiency.

Inference can also batch hundreds of samples to achieve optimal throughput on jobs run overnight in data centers to process substantial amounts of stored data. These jobs tend to emphasize throughput over latency. However, for real-time usages, high batch sizes also carry a latency penalty, and for these usages, lower batch sizes (as low as a single sample) are used, trading off throughput to get lowest latency. A hybrid approach, sometimes referred to as “auto-batching”, sets a time threshold of say, 10 milliseconds, and batches as many samples as possible within those ten milliseconds before sending them on for inference. This approach achieves better throughput while maintaining a set latency amount.

The NVIDIA Deep Learning Platform

The NVIDIA platform is designed to make deep learning accessible to every developer and data scientist anywhere in the world. All major DL frameworks, including CAFFE, Caffe2, TensorFlow, Microsoft Cognitive Toolkit, PyTorch, and MXNet, are accelerated on the NVIDIA platform. It includes productivity tools like NVIDIA DIGITS™, which enables developers to quickly design the best network for their data without writing any code. Developers have access to state-of-the-art tools in the NVIDIA Deep Learning SDK for applications in the data center, in autonomous vehicles, and in devices at the edge. Inference can be deployed from data center to the edge and the engine in the NVIDIA deep learning platform that optimizes a neural network for optimal performance across these deployments is TensorRT.

The Tesla V100, Based on NVIDIA Volta Architecture

The NVIDIA® Tesla® V100 accelerator incorporates the powerful new NVIDIA Volta GPU architecture. Volta not only builds upon the advances of its predecessor, the NVIDIA Pascal™ GPU architecture, but significantly improves both performance and scalability, adding many new features that improve programmability. These advances are supercharging HPC, data center, supercomputer, and deep learning systems and applications.

VOLTA KEY FEATURES

Key compute features of Tesla V100 include:

- **New Streaming Multiprocessor (SM) Architecture** Optimized for Deep Learning: Volta features a major new redesign of the SM processor architecture that is at the center of the GPU. New Tensor Cores designed specifically for deep learning deliver up to 12x higher peak TFLOPS for training and 6x higher peak TFLOPS for inference.
- **Next-Generation NVIDIA NVLink™:** The next-generation of NVIDIA’s NVLink high-speed interconnect delivers higher bandwidth, more links, and improved scalability for multi-GPU server configurations. Volta GV100 supports up to six NVLink links and total bandwidth of 300 GB/sec.

- **HBM2 Memory: Faster, Higher Efficiency:** Volta's highly tuned 16 GB HBM2 memory subsystem delivers 900 GB/sec of peak memory bandwidth. The combination of both a new-generation HBM2 memory from Samsung, and a next-generation memory controller in Volta deliver up to 95% memory bandwidth utilization running many workloads.
- **Volta Multi-Process Service:** Volta Multi-Process Service (MPS) is a new feature of the Volta GV100 architecture providing hardware acceleration of critical components of the CUDA MPS server, enabling improved performance, isolation, and better quality of service (QoS) for multiple compute applications sharing the GPU.
- **Enhanced Unified Memory and Address Translation Services:** V100 Unified Memory technology includes new access counters to allow more accurate migration of memory pages to the processor that accesses them most frequently, improving efficiency for memory ranges shared between processors.
- **Maximum Performance and Maximum Efficiency Modes:** In Maximum Performance mode, the Tesla V100 accelerator will operate up to its TDP (Thermal Design Power) level of 300 W to deliver the highest data throughput. Maximum Efficiency Mode allows data center managers to tune power usage of their Tesla V100 accelerators to operate with optimal performance per watt.
- **Cooperative Groups and New Cooperative Launch APIs:** Cooperative Groups is a new programming model introduced in CUDA 9 for organizing groups of communicating threads. Cooperative Groups allows developers to express the granularity at which threads are communicating, helping them to express richer, more efficient parallel decompositions.
- **Volta Optimized Software:** New versions of deep learning frameworks such as Caffe2, MXNet, Microsoft Cognitive Toolkit, PyTorch, TensorFlow, and others harness the performance of Volta to deliver dramatically faster training times and higher multi-node training performance.

To learn more, download the Volta Architecture Whitepaper (link: https://www.nvidia.com/object/volta-architecture-whitepaper.html)

TensorRT - The Programmable Inference Accelerator

NVIDIA TensorRT™ is a high-performance deep learning inference optimizer and runtime that delivers low latency, high-throughput inference for deep learning applications. TensorRT can be used to rapidly optimize, validate, and deploy trained neural networks for inference to hyperscale data centers, embedded, or automotive product platforms.

Once the neural network is trained, TensorRT enables the network to be compressed, optimized and deployed as a runtime without the overhead of a framework. TensorRT can be accessed three ways: A C++ API for describing a neural network to run, a high-level Python interface that can load existing Caffe or TensorFlow models, or a Representational State Transfer (REST) API interface for easy use in a devops environment. TensorRT combines layer merges and model compaction, and also performs normalization and conversion to optimized matrix math depending on the specified precision (FP32, FP16 or INT8) for improved latency, throughput, and efficiency.

Inference computations can use lower-precision tensor operations with minimal accuracy loss. Tesla V100 and P4 accelerators implement 16-bit floating-point (FP16) and 8-bit integer (INT8) instructions, respectively, for dot product operations. The result is improved model size capacity, memory utilization, latency, and throughput as well as power efficiency.

In measured benchmarks, Tesla P4 delivers up to a 3x throughput improvement using INT8, better latency, and higher power efficiency.

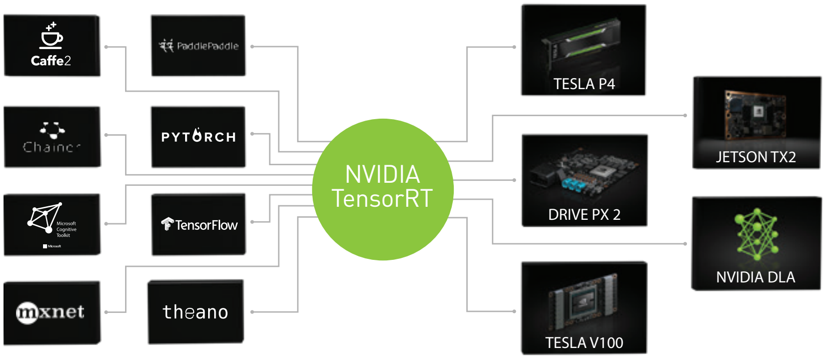


Figure 2

Figure 2: TensorRT can ingest trained neural networks from diverse deep learning frameworks, and optimize them for deployed inference on any NVIDIA deep learning platform.

NVIDIA TensorRT is a high-performance deep learning inference optimizer and runtime for production deployment of deep learning applications. It can be used to rapidly optimize, validate and deploy trained neural networks for inference to hyperscale data centers, embedded, or automotive GPU platforms. With it, developers can unleash the full potential of NVIDIA Volta architecture's Tensor Cores to deliver three times more performance than the previous generation architecture.

TensorRT 3's key features include:

- **TensorFlow Support:** TensorFlow models can be directly ingested, optimized and deployed with up to 18x faster performance compared to TensorFlow framework inference on Tesla V100.

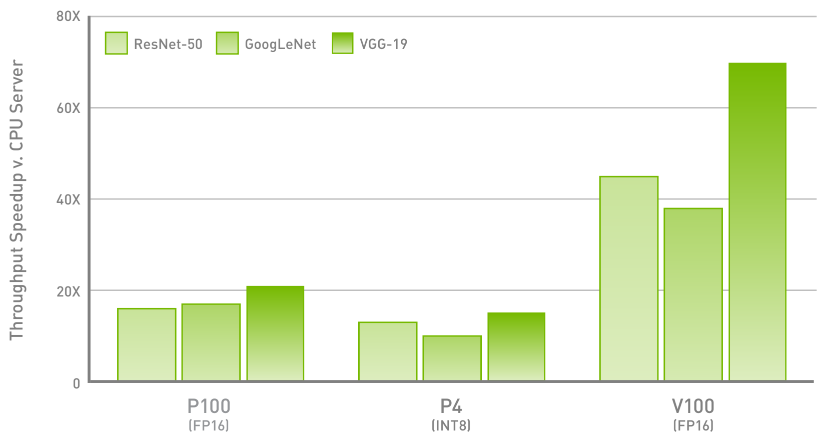
- **Python API Support:** Ease of use improvement, allowing developers to call TensorRT using the Python scripting language.
- **Weight and Activation Precision Optimization:** Significantly improves inference performance of models trained in FP32 full precision by quantizing them to FP16 and INT8, while minimizing accuracy loss.
- **Layer and Tensor Fusion (Graph Optimization):** Improves GPU utilization and optimizes memory storage and bandwidth by combining successive nodes into a single node, for single kernel execution.
- **Kernel Auto-Tuning (Auto-tuning):** Optimizes execution time by choosing the best data layer and best parallel algorithms and kernels for the target Jetson, Tesla or DrivePX GPU platforms.
- **Dynamic Tensor Memory (Memory optimization):** Reduces memory footprint and improves memory re-use by allocating memory for each tensor only for the duration its usage
- **Multi Stream Execution:** Scales to multiple input streams, by processing them parallel using the same model and weights Figure 2: TensorRT can ingest trained neural networks from diverse deep learning frameworks, and optimize them for deployed inference on any NVIDIA deep learning platform.

While it is possible to do inference operations within a deep learning framework, TensorRT easily optimizes networks to deliver far more performance. TensorRT also takes full advantage of the Volta architecture, and this combination delivers up to 70x more throughput vs. a CPU-only server.

This chart shows maximum throughput numbers for several image-based convolutional neural networks (CNNs), using a larger batch size of 128, which delivers the best throughput.

Chart 1: TensorRT supports both FP16 and INT8 precisions with near-zero accuracy loss. Here, Tesla V100 and TensorRT combine to deliver 70x more inference throughput versus a CPU-only server.

70X Higher Throughput vs. CPU on CNNs



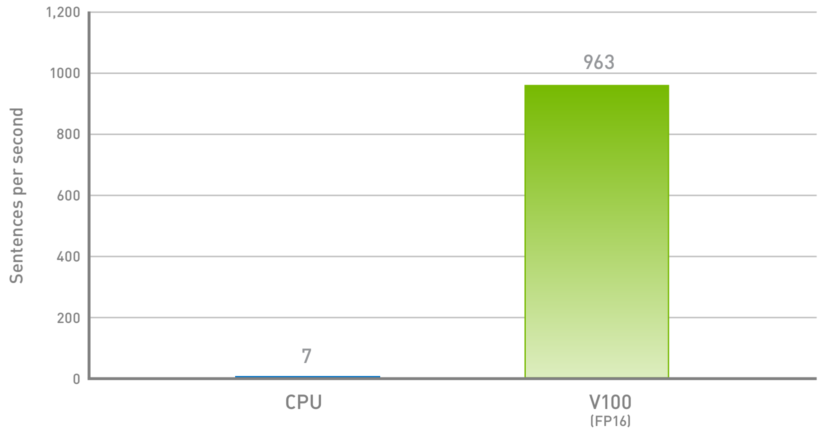
Workloads: ResNet-50, GoogleNet, VGG-19 | Data-set: ImageNet | CPU Servers: Xeon E5-2690 v4 @ 2.6GHz | GPU: add 1X NVIDIA® Tesla® V100 or Tesla P4

Chart 1

As deep learning expands the use-cases it can examine, new types of neural networks are emerging on an almost-monthly cadence, as evidenced by the number of academic papers published on sites like Cornell University's arXiv (link: https://arxiv.org/). An emerging class of neural networks for speech recognition, natural language processing and translation is recurrent neural networks (RNNs).

Chart 2: OpenNMT is a full-featured, open-source neural machine translation system that uses the Torch math toolkit. The results shown here are on a workload called WMT-DE that is doing translation from English to German. Our inference tests here use Intel's Deep Learning SDK beta 2 for CPU tests, and TensorRT 3 for the Tesla V100. Tesla V100 delivers over 130 times more throughput.

130X Higher Throughput vs. CPU on RNNs



Workload: OpenNMT, WMT-DE English to German translation | CPU Servers: Xeon E5-2690 v4 @ 2.6GHz | GPU: add 1X NVIDIA® Tesla® V100

Chart 2

GPU Inference: Business Implications

Tesla V100 and P4 deliver massive performance boosts and power efficiency, but how does that benefit acquisition and operations budgets? Simply put: big performance translates into big savings.

The chart below shows that a single HGX server with eight Tesla V100s delivers the same throughput performance of 120 dual-socket high-end Xeon Scalable Processor CPU-based servers that take up three server racks. That translates into a 13X reduction in Total Cost of Ownership (TCO).

13x Better Data Center TCO

Delivering 48,000 images/second at
1/12 the Cost | 1/20 the Power | 3 Racks in a Box

Image 1

Image 1: A single HGX server with 8 Tesla V100s (left) delivers image recognition throughput of about 48,000 images/second on ResNet-50, the same performance as three racks containing 120 dual-socket CPU Xeon Scalable Processor servers (right). To estimate Xeon Scalable Processor performance, we used measured performance of a Xeon E5-2690v4, and applied a performance scale factor of 1.5x.

Inference Performance: Getting the Complete Picture

Measured performance in computing tends to fixate on speed of execution. But in deep learning inference performance, speed is one of four critical factors that come into play. Here, it is about speed (throughput), latency, efficiency, and accuracy. Two of these factors are key contributors to end-users' quality of experience (accuracy and latency), while the other pair (throughput and efficiency) are critical to data center efficiency.

Anatomy of Inference Performance

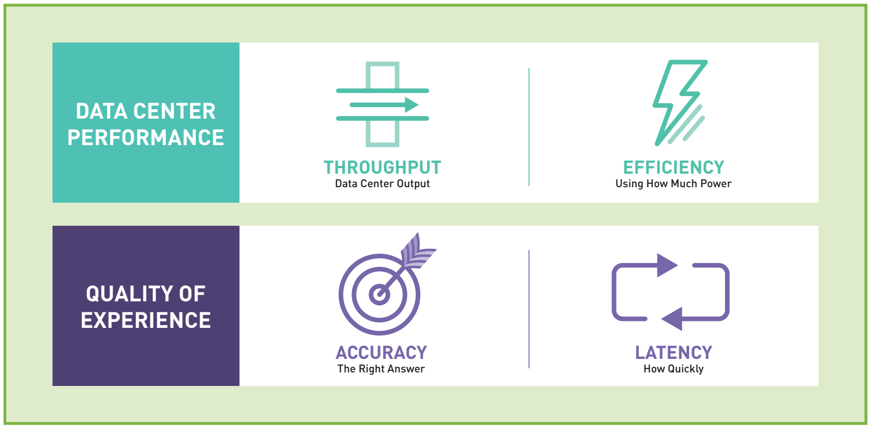


Figure 3

Figure 3: Optimal inference performance must deliver on four fronts to satisfy both data center performance and user experience requirements.

Data Center Efficiency

Throughput: The volume of output within a given period. Often measured in inferences/second or samples/second, per-server throughput is critical to cost-effective scaling in data centers, and to Tesla accelerators are industry's best end-to-end platform for training and inferencing in the data center.

Efficiency: Amount of throughput delivered per unit-power, often expressed as performance/watt. Efficiency is another key factor to cost-effective data center scaling, since servers, server racks and entire data centers must operate within fixed power budgets.

Quality of Experience

Latency: Time to execute an inference, usually measured in milliseconds. Low latency is critical to delivering rapidly growing, real-time inference-based services. As an example, for speech-based services to feel natural and conversational, answers need to come back to the end-user as quickly as possible. Even a lag of a one second starts to feel unnatural. **Google has stated**[1] that 7ms is an optimal latency target for usages such as search, and so throughput delivered within those 7ms becomes another important measure. For other real-time usages, a recent **Google presentation**[2] discussed 200ms as a viable latency target for speech-to-text or translation.

Accuracy: A trained neural network's ability to deliver the correct answer. For image-based usages, the critical metric is expressed as a Top-5 or Top-1 percentage. These "Top" metrics represent the inference's estimation as to what the analyzed sample most likely is. A higher percentage Top-5 or Top-1 indicates higher confidence of the inference's answer. Speech and translation services look at other metrics, such as a Bilingual Evaluation Understudy (BLEU) score. Generally, neural network training requires higher precision, whereas inference can often be carried out using reduced precision. Using lower precision improves throughput, efficiency and even latency, but maintaining high accuracy is essential for best user experiences. TensorRT offers FP32 and FP16 floating-point precisions, as well as INT8 integer precision with near-zero loss in accuracy.

Both Tesla V100 and Tesla P4 bring massive increases in inference in the four critical inference performance metrics versus CPU-only data center servers, and a single Tesla GPU-equipped server node can replace up to 70 CPU-only nodes.

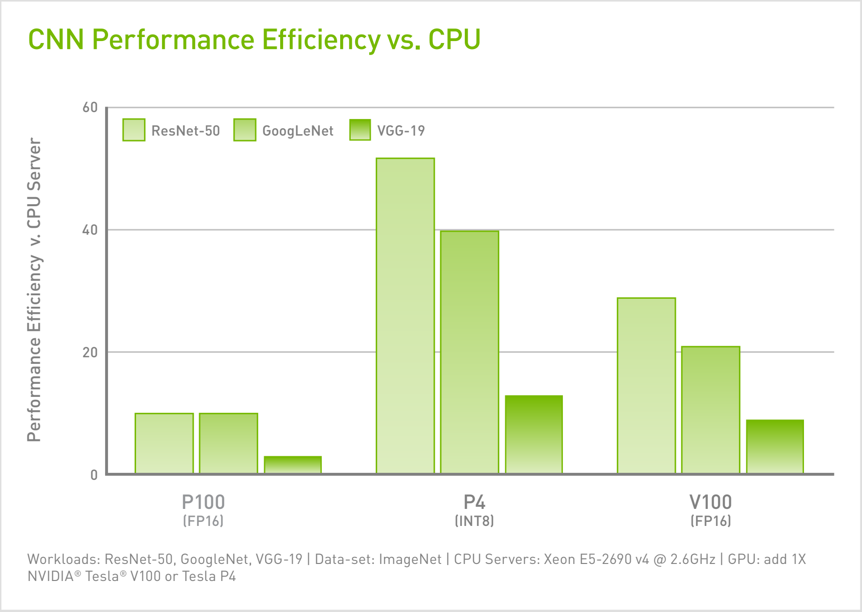
[1] ref: https://arxiv.org/ftp/arxiv/papers/1704/1704.04760.pdf

[2] https://atscaleconference.com/videos/google-translate-breaking-language-barriers-in-emerging-markets/

Performance Efficiency

We have covered maximum throughput already, and while very high throughput on deep learning workloads is a key consideration, so too is how efficiently a platform can deliver that throughput.

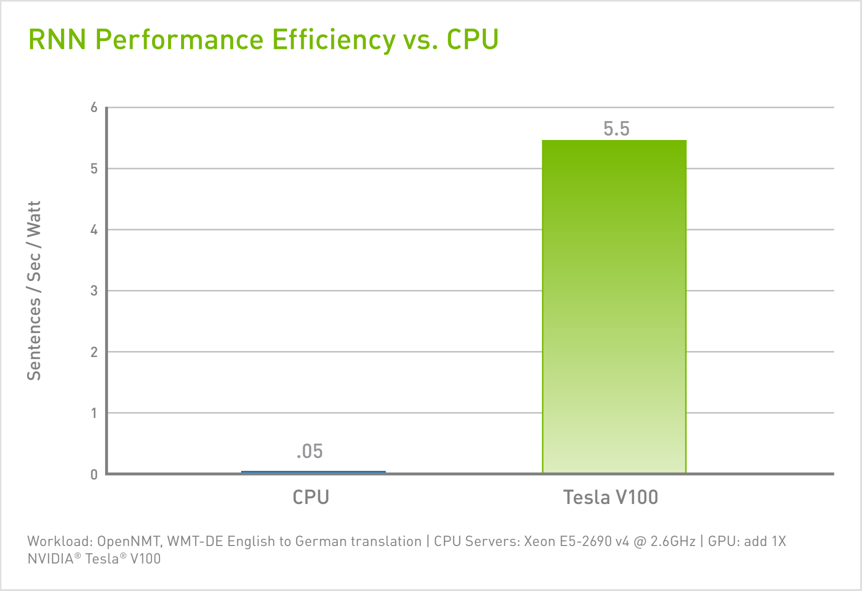
Chart 3: Both Tesla V100 and Tesla P4 offer massive increases in performance efficiency as measured by performance/ watt. For scale-out inference and data center edge solutions, Tesla P4 offer excellent efficiency with a TDP of 75W. Tesla V100 delivers much higher throughput thanks to the Volta architecture, and higher TDP.


CNN Performance Efficiency vs. CPU

Workloads: ResNet-50, GoogleNet, VGG-19 | Data-set: ImageNet | CPU Servers: Xeon E5-2690 v4 @ 2.6GHz | GPU: add 1X NVIDIA® Tesla® V100 or Tesla P4

Chart 3

Chart 4: A Tesla V100 GPU-equipped server can deliver over 100x better efficiency than a CPU-only server on speech-focused inference on an RNN.


RNN Performance Efficiency vs. CPU

Workload: OpenNMT, WMT-DE English to German translation | CPU Servers: Xeon E5-2690 v4 @ 2.6GHz | GPU: add 1X NVIDIA® Tesla® V100

Chart 4

Accuracy

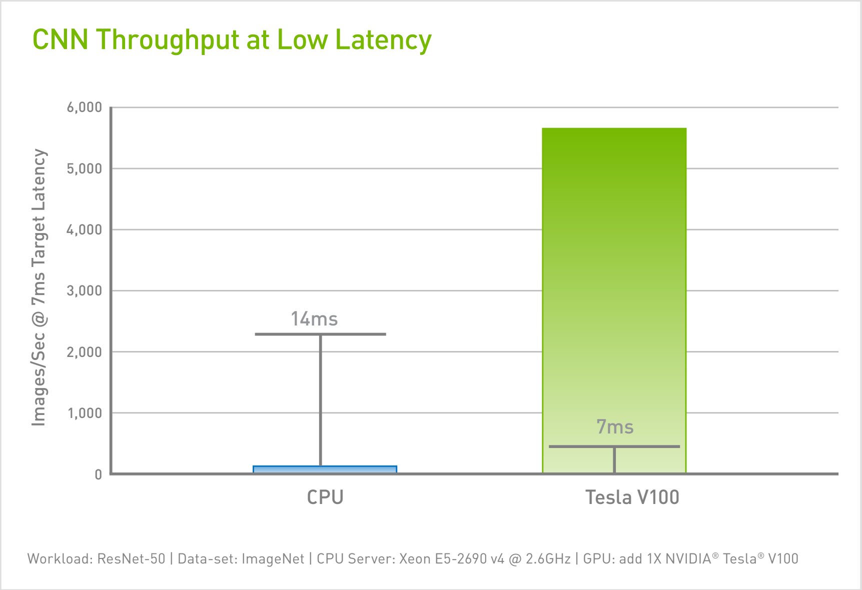
What becomes evident as we examine the various aspects of inference performance is that they are all inextricably linked. A platform that delivers on one, but falters on others will ultimately not get the job done. Tesla V100 and Tesla P4 combined with TensorRT 3 are ideal inference platforms, as they both can deliver massive performance improvements with near-zero loss in accuracy.

No Accuracy Loss at FP16, INT8 Using TensorRT

GoogLeNet (FP16)	FP32	FP16	Difference
	72.23%	72.25%	+0.02%
GoogLeNet (INT8)	FP32	INT8	Difference
	73.11%	72.54%	-0.57%

Table 1

Latency


CNN Throughput at Low Latency

Workload: ResNet-50 | Data-set: ImageNet | CPU Server: Xeon E5-2690 v4 @ 2.6GHz | GPU: add 1X NVIDIA® Tesla® V100

Chart 5

Chart 5: With the emergence of real-time AI-based services, latency becomes an increasingly important facet of inference performance. So not only is high throughput critical, but delivering high throughput within a specified latency budget to optimize end-user experience. **Google has stated**[3] that 7ms is an optimal latency target, and applying that latency target here, the above chart shows that Tesla V100 delivers 40x more performance than a CPU-only server within the 7-millisecond latency budget. Meanwhile, the CPU server is unable to deliver its throughput of 140 images/sec within the specified latency budget.

[3] ref: https://arxiv.org/ftp/arxiv/papers/1704/1704.04760.pdf

Chart 6: For speech applications, **Google has described**[4] in a recent presentation a target latency of about 200ms for speech applications. Here again, Tesla V100 is not only outperforming a CPU-only server by a factor of more than 150x, but is staying well within the specified 200ms latency budget. Meanwhile, the CPU can only deliver about 4 sentences per second, and misses the required latency boundary.

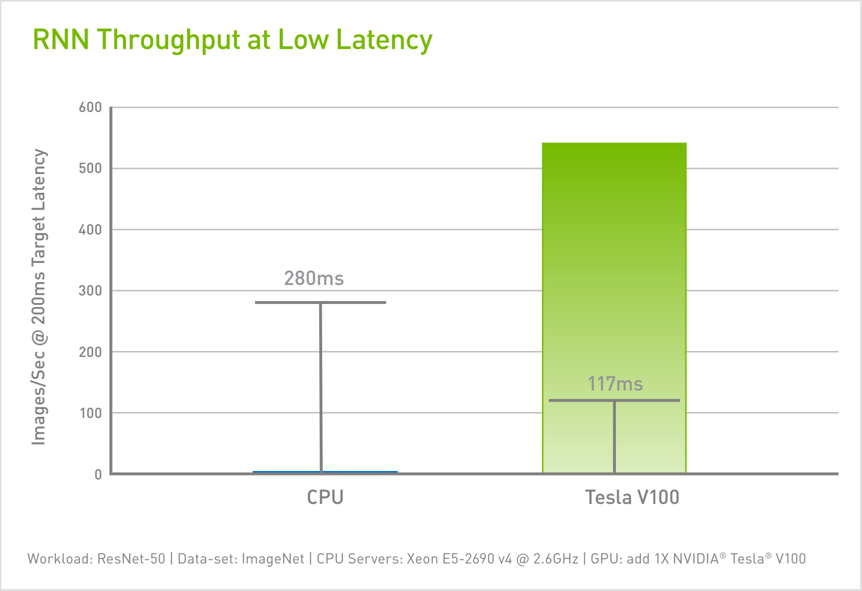


Chart 6

Jetson: Inference at the Edge

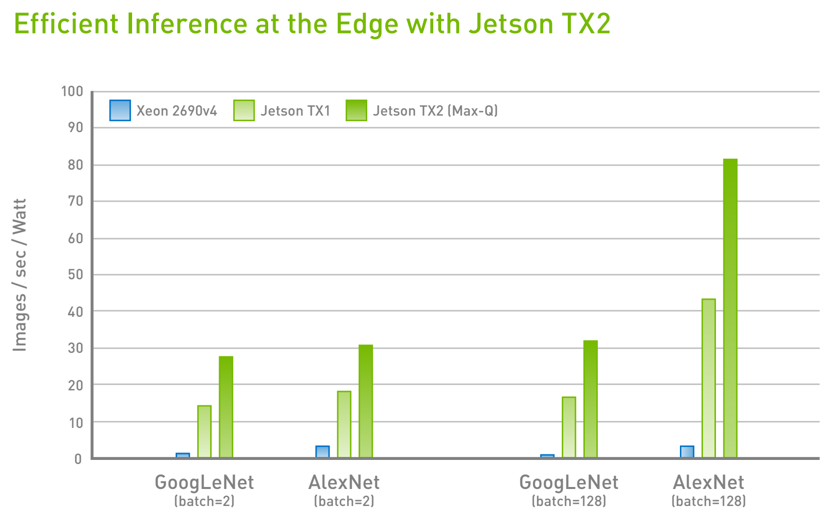
NVIDIA Jetson™ TX2 is a credit card-sized open platform that delivers AI computing at the edge—opening the door to powerfully intelligent factory robots, commercial drones, and smart cameras for AI cities. Based on NVIDIA's Pascal architecture, Jetson TX2 offers twice the performance of its predecessor, or it can run at more than twice the power efficiency while drawing less than 7.5 watts of power. This allows Jetson TX2 to run larger, deeper neural networks on edge devices. The result: smarter devices with higher accuracy and faster response times for tasks like image classification, navigation, and speech recognition. Deep learning developers can use the very same development tools for Jetson that they use on the Tesla platform such as CUDA, cuDNN, and TensorRT.

Jetson TX2 was designed for peak processing efficiency at 7.5W of power. This level of performance, referred to as Max-Q, represents the maximum performance and maximum power efficiency range on the power/performance curve. Every component on the module including the power supply is optimized to provide the highest efficiency at this point. The Max-Q frequency for the GPU is 854MHz, and for the ARM A57 CPUs, it is 1.2GHz. While Dynamic Voltage and Frequency Scaling (DVFS) permits Jetson TX2's Tegra "Parker" SoC to adjust clock speeds at run time according to user load and power consumption, the Max-Q configuration sets a cap on the clocks to ensure that the application is operating in the most efficient range only.

Jetson enables real-time inferencing when connectivity to a AI data center is either not possible (e.g. remote sensing) or the end-to-end latency is too high for real time use (e.g. autonomous drone). Although most platforms with a limited power budget will benefit most from

[4] ref: https://atscaleconference.com/videos/google-translate-breaking-language-barriers-in-emerging-markets/

Max-Q behavior, others may prefer maximum clocks to attain peak throughput, albeit with higher power consumption and reduced efficiency. DVFS can be configured to run at a range of other clock speeds, including underclocking and overclocking. Max-P, the other preset platform configuration, enables maximum system performance in less than 15W. The Max-P frequency is 1.12GHz for the GPU and 2GHz for the CPU when either the ARM A57 cluster is enabled, or the Denver 2 cluster is enabled, and 1.4GHz when both clusters are enabled.



Workloads: GoogLeNet and AlexNet | Data-set: ImageNet | CPU Server: Xeon E5-2690 v4 @ 2.6GHz | NVIDIA® Jetson™ TX1 reference platform, 256 Maxwell CUDA Cores, CPU: Quad ARM A57 | Jetson TX2 reference platform, 256 Pascal CUDA Cores, CPUs: HMP Dual Denver + Quad ARM A57

Chart 7

Chart 7: Jetson TX2 performs GoogLeNet inference up to 33.2 images/sec/Watt, nearly double the efficiency of Jetson TX1. Additionally, Jetson TX2 delivers up to 27x better performance/watt versus a Xeon* CPU-based server.

For many network-edge applications, low latency is a must-have. Executing inference on-device is a far more optimal approach than trying to send this work over a wireless network and in and out of a CPU-based server in a remote data center. In addition to its on-device locality, Jetson TX2 also delivers outstanding low-latency on small batch workloads, usually under ten milliseconds. For comparison, a CPU-based server has a latency of around 23 milliseconds, and, adding roundtrip network and data center travel time, that figure can be well over 100 milliseconds.

The Rise of Accelerated Computing

Google* has announced its Cloud Tensor Processing Unit (TPU), and its applicability to deep learning training and inference. And while Google and NVIDIA chose different development paths, there are several themes common to both our approaches. Specifically, AI requires accelerated computing. Accelerators provide the significant data processing necessary to keep up with the growing demands of deep learning in an era when Moore's law is slowing. Tensor processing is at the core of delivering performance for deep learning training and inference. Tensor processing is a major new workload that enterprises must consider when building modern data centers. Accelerating tensor processing can dramatically reduce the cost of building modern data centers.

According to Google, the Cloud TPU (also referred to as "TPU2") will be available later this year, and that a single Cloud TPU can deliver 45 teraflops of computing horsepower. NVIDIA's Tesla V100 can deliver 125 teraflops of deep learning performance for both training and inference. An 8-GPU configuration such as DGX-1 can now deliver a petaflop of deep learning computing power.

NVIDIA's approach democratizes AI computing for every company, every industry, every computing platform and accelerates every development framework – from the cloud, to the enterprise, to cars, and to the edge. Google and NVIDIA are the clear leaders – we collaborate closely while taking different approaches to enable the world with AI.

Note on FPGA

As the deep learning field continues to grow rapidly, other types of hardware have been proposed as potential solutions for inference, such as Field Programmable Gate Arrays (FPGA). FPGAs are used for specific functions in network switches, 4G base stations, motor control in automotive, and test equipment in semiconductors among other use cases. It is a sea of general-purpose programmable logic gates designed for various usages, so long as the problem fits on the chip. But because these are programmable gates rather than a hard-wired ASIC, FPGAs are inherently less efficient.

Low-latency Inference Throughput vs. FPGA Projections

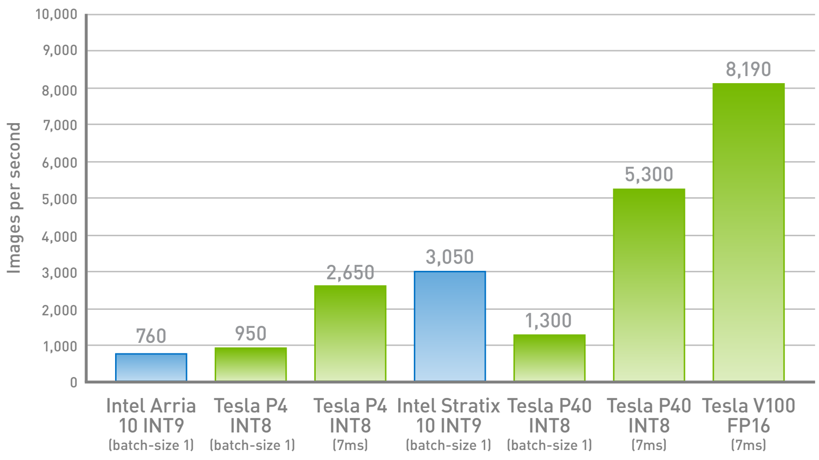


Chart 8: Comparing throughput on the GoogLeNet network of measured Tesla GPU data versus Intel's stated projections of its Arria 10 and Stratix 10 FPGAs.[5] Using a batch size of 1 produces a theoretical number to get to lowest latency, which is critical for those inference-based services that depend on fast response times. However, an improved approach sets a latency limit, and then gets maximum throughput within that limit, giving developers and end-users the best of both worlds: higher throughput and low latency. Google has stated that 7ms is a good target for real-time inference-based workloads, and Tesla GPUs are able to deliver significantly more throughput performance and performance/watt efficiency using this improved approach.

Workload: GoogLeNet | Data-set: ImageNet | GPU Servers: Xeon E5-2690 v4 @ 2.6GHz and 1X NVIDIA® Tesla® P40, Tesla P4 or Tesla V100 data measured on pre-production hardware | Source for FPGA Data: "Accelerating Deep Learning with the OpenCL™ Platform and Intel® Stratix® 10 FPGAs", Intel

Chart 8

[5] Accelerating Deep Learning with the OpenCL™ Platform and Intel: www.altera.com/content/dam/altera-www/global/en_US/pdfs/literature/wp/wp-01269-accelerating-deep-learning-with-opencl-and-intel-stratix-10-fpgas.pdf

Programmability and Time to Solution Considerations

The speed of deep learning innovation drives the need for a programmable platform that enables developers to quickly try new network architectures, and iterate as new findings come to light.

Image 2: The diversity and complexity of neural network types continues to expand rapidly, and a platform that enables developers to quickly experiment and iterate is critical to driving deep learning innovation forward.

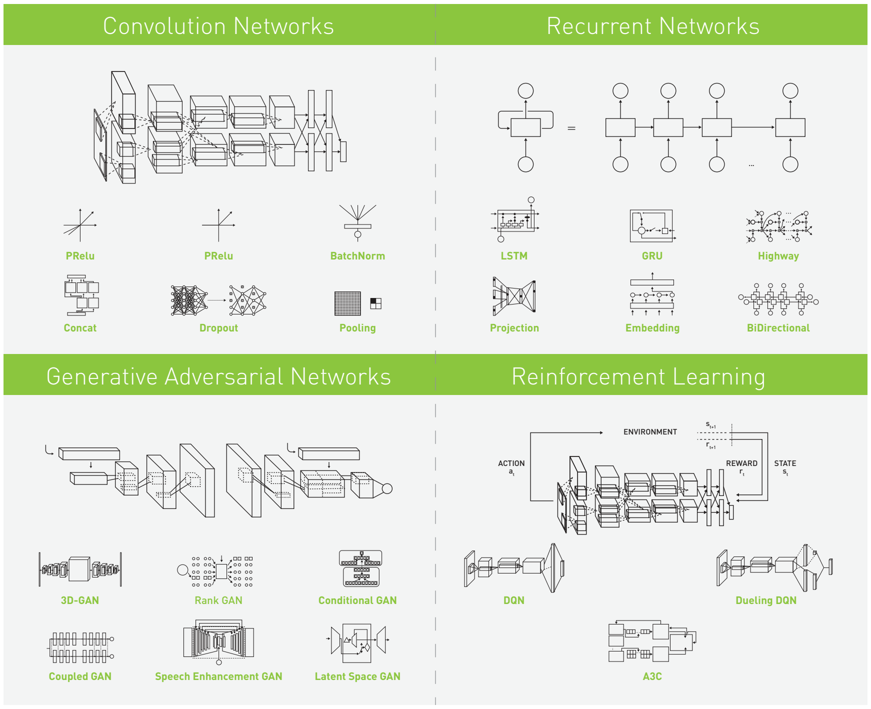


Image 2

Another challenge posed by FPGAs is that in addition to software development, FGPAs must be reconfigured at the hardware level to run each iteration of new neural network architectures. This complex hardware development slows time to solution by weeks and sometimes months, and hence, innovation. Whereas GPUs continue to be the most programmable platform of choice for quickly prototyping, testing and iterating cutting-edge network designs, thanks to robust framework acceleration support, dedicated deep learning logic like Tesla V100's Tensor Cores, and TensorRT to optimize trained networks for deployed inference.

Conclusion

Deep learning is revolutionizing computing, impacting enterprises across multiple industrial sectors. The NVIDIA deep learning platform is the industry standard for training, and leading enterprises are already deploying GPUs for their inferencing workloads, leveraging its powerful benefits. Neural networks are rapidly becoming exponentially larger and more complex, driving massive computing demand and cost. In cases where AI services need to be responsive, modern networks are too compute-intensive for traditional CPUs.

Inference performance has four aspects—throughput, efficiency, latency and accuracy—that are critical to delivering both data center efficiency and great user experiences. This paper demonstrates how Tesla GPUs can deliver up to 13X TCO savings in the data center for "offline inferencing" use cases. In fact, the savings in energy cost alone more than pays for the Tesla-powered server. And at network's edge, the new Jetson TX2 brings server-class inference performance in less than 10W of power and enables device-local inference to significantly cut inference latency times.

An effective deep learning platform must have three distinct qualities: It must have a processor custom-built for deep learning. It must be software-programmable. And industry frameworks must be optimized for it, powered by a developer ecosystem that is accessible and adopted around the world. The NVIDIA deep learning platform is designed around these three qualities and is the only end-to-end deep learning platform. From training to inferencing. From data center to the network's edge.

To learn more about NVIDIA's Tesla products visit:
www.nvidia.com/tesla

To learn more about JetsonTX2, visit:
www.nvidia.com/object/embedded-systems.html

To learn more about TensorRT and other NVIDIA development tools visit: **developer.nvidia.com/tensorrt**

To see the extensive list of applications that already take advantage of GPU acceleration today visit: **www.nvidia.com/gpu-applications**

*All trademarks and registered trademarks are the property of their respective owners.

Performance Data Tables

CNNs		TESLA V100 (FP16/FP32 MIXED PRECISION)			
NETWORK	BATCH SIZE	PERF (IMGS/ SEC)	TOTAL BOARD POWER	PERFORMANCE/ WATT	LATENCY (MS)
GoogLeNet	1	876	98.6	8.88	1.14
	2	1,235	65.4	18.88	1.62
	4	2,194	80.4	27.29	1.82
	8	3,776	112.2	33.65	2.12
	64	8,630	209.2	41.25	7.42
	128	9,404	225.6	41.68	13.61
ResNet-50	1	504	94.2	5.35	1.99
	2	797	66.8	11.93	2.51
	4	1,450	83.7	17.32	2.76
	8	2,493	113.6	21.95	3.21
	64	5,572	196.4	28.37	11.49
	128	6,024	210.1	28.67	21.25
VGG-19	1	464	144	3	2
	2	718	138.7	5.18	2.79
	4	1,032	173.4	5.95	3.88
	8	1,334	203.4	6.56	6
	64	1,979	241	8.21	32.34
	128	2,030	238.4	8.52	63.04

CNNs		TESLA P4 (INT8 PRECISION)			
NETWORK	BATCH SIZE	PERF (IMGS/ SEC)	TOTAL BOARD POWER	PERFORMANCE/ WATT	LATENCY (MS)
GoogLeNet	1	837	42.4	19.74	1.19
	2	1,106	45.6	24.25	1.81
	4	1,489	49.1	30.33	2.69
	8	1,930	56.66	34.06	4.15
	64	2,531	64.25	39.39	25.29
	128	2,566	64.2	39.97	49.89
ResNet-50	1	600	32.9	18.24	1.67
	2	765	32.8	23.32	2.61
	4	1,019	33	30.88	3.93
	8	1,319	33.1	39.85	6.07
	64	1,715	33.2	51.66	37.32
	128	1,721	32.9	52.31	74.36
VGG-19	1	204	32.6	6	4.9
	2	273	32.9	8.30	7.33
	4	338	32.8	10.30	11.82
	8	380	32.66	11.64	21.04
	64	414	32.7	12.66	153.23
	128	438	32.8	13.35	292

RNN		TESLA V100 (FP16/FP32 MIXED PRECISION)	
NETWORK	BATCH SIZE	PERF (SENTENCES/ SEC)	LATENCY (MS)
OpenNMT	1	23	42
	2	46	43
	4	82	49
	8	156	51
	64	541	118
	128	725	176

			JETSON TX2 (MAXQ MODE)				
NETWORK	BATCH SIZE	PERF (IMGS/ SEC)	AP+DRAM POWER UPSTREAM* (WATTS)	AP+DRAM PERFORMANCE / WATT	GPU POWER DOWNSTREAM* (WATTS)	GPU PERFORMANCE / WATT	LATENCY (MS)
AlexNet	1	119	6.6	18.0	2.3	52.4	8.4
	2	188	6.6	28.4	2.6	73.4	10.6
	4	264	6.7	39.3	2.9	92.6	15.2
	8	276	6.1	45.1	2.8	99.6	29.0
	64	400	6.4	62.6	3.2	125.7	160.0
	128	425	6.4	66.4	3.2	132.6	301.3
GoogLeNet	1	141	5.7	24.7	2.6	54.3	7.1
	2	156	5.9	26.2	2.7	57.6	12.8
	4	170	6.2	27.7	2.8	59.8	23.5
	8	180	6.4	28.2	3.0	60.6	44.5
	64	189	6.6	28.8	3.1	61.6	337.8
	128	191	6.6	28.9	3.1	61.6	671.8
ResNet-50	1	64.3	5.4	11.9	2.3	28.3	15.6
	2	76.5	5.3	14.4	2.3	33.7	26.2
	4	81.0	5.4	15.1	2.3	34.8	49.4
	8	83.4	5.4	15.4	2.4	35.4	95.9
	64	89.4	5.5	16.2	2.4	37.6	715.5
	128	89.9	5.5	16.2	2.4	37.7	1,424.3
VGG-19	1	18.8	7.2	2.6	2.9	6.4	53.1
	2	21.5	7.2	3.0	3.1	6.9	93.1
	4	22.6	7.3	3.1	3.1	7.2	176.8
	8	22.8	7.2	3.2	3.1	7.3	351.3
	64	22.9	7.2	3.2	3.2	7.1	2,792.4
	128	22.6	7.1	3.2	3.2	7.2	5,660.6

*Up = upstream power (above voltage regulators), and Down = downstream power (below the voltage regulators)

			JETSON TX2 (MAXP MODE)				
NETWORK	BATCH SIZE	PERF (IMGS/ SEC)	AP+DRAM POWER UPSTREAM* (WATTS)	AP+DRAM PERFORMANCE / WATT	GPU POWER DOWNSTREAM* (WATTS)	GPU PERFORMANCE / WATT	LATENCY (MS)
AlexNet	1	146	8.9	16.3	3.62	40.3	6.85
	2	231	9.2	25.2	4.00	57.7	8.66
	4	330	9.5	34.8	4.53	72.9	12.12
	8	349	8.8	39.8	4.42	79.0	22.90
	64	515	9.5	54.1	5.21	98.8	124.36
	128	546	9.6	56.9	5.28	103.5	234.32

GoogLeNet	1	179	8.2	21.8	4.14	43.2	5.6
	2	199	8.6	23.2	4.36	45.6	10.1
	4	218	9.0	24.2	4.61	47.2	18.4
	8	231	9.3	24.8	4.83	47.8	34.7
	64	243	9.7	25.1	5.03	48.3	263.6
	128	244	9.6	25.3	5.02	48.6	524.2
ResNet-50	1	82	7.4	11.1	3.49	23.5	12.2
	2	98	7.5	13.0	3.63	26.9	20.5
	4	104	7.6	13.6	3.71	27.9	38.6
	8	107	8.0	13.4	3.95	27.1	74.8
	64	115	7.9	14.6	3.81	30.1	558.9
	128	115	7.9	14.6	3.82	30.1	1,113.2
VGG-19	1	23.7	10	2.3	5	5.0	42.2
	2	26.8	10	2.6	4.93	5.4	74.7
	4	28.2	10	2.7	4.97	5.7	142.0
	8	28.3	10	2.8	4.96	5.7	282.7
	64	28.7	10	2.8	5.16	5.6	2,226.7
	128	28.4	10	2.8	5.09	5.6	4,514.0

*Up = upstream power (above voltage regulators), and Down = downstream power (below the voltage regulators)

JETSON TX1

NETWORK	BATCH SIZE	PERF (IMGS/SEC)	AP+DRAM POWER UPSTREAM* (WATTS)	AP+DRAM PERFORMANCE / WATT	GPU POWER DOWNSTREAM* (WATTS)	GPU PERFORMANCE / WATT	LATENCY (MS)
AlexNet	1	95	9.2	10.3	5.1	18.6	10.5
	2	158	10.3	15.2	6.4	24.5	12.7
	4	244	11.3	21.7	7.6	32.0	16.4
	8	253	11.3	22.3	7.8	32.5	31.6
	64	418	12.5	33.5	9.4	44.5	153.2
	128	449	12.5		9.6	46.9	284.9
GoogLeNet	1	119	10.7	11.1	7.2	16.4	8.4
	2	133	11.2	12.0	7.7	17.4	15.0
	4	173	11.6	14.9	8.0	21.6	23.2
	8	185	12.3	15.1	9.0	20.6	43.2
	64	196	12.7	15.5	9.4	20.7	327.0
	128	196	12.7	15.5	9.5	20.7	651.7
ResNet-50	1	60.8	9.5	6.4	6.3	9.7	16.4
	2	67.8	9.8	6.9	6.5	10.5	29.5
	4	80.5	9.7	8.3	6.6	12.1	49.7
	8	84.2	10.2	8.3	7.0	12.0	95.0
	64	91.2	10.0	9.1	6.9	13.2	701.7
	128	91.5	10.4	8.8	7.3	12.6	1,399.3
VGG-19	1	13.3	11.3	1.2	7.6	1.7	75.0
	2	16.4	12.0	1.4	8.6	1.9	122.2
	4	19.2	12.2	1.6	8.9	2.2	207.8
	8	19.5	12.0	1.6	8.6	2.3	410.6
	64	20.3	12.2	1.7	9.1	2.2	3,149.6
	128	20.5	12.5	1.6	9.3	2.2	3,187.3

*Up = upstream power (above voltage regulators), and Down = downstream power (below the voltage regulators)

Test Methodology

For our performance analysis, we focus on four neural network architectures. AlexNet (2012 ImageNet winner), and the more recent GoogLeNet (2014 ImageNet winner), a much deeper and more complicated neural network compared to AlexNet, are two classical networks. VGG-19 and ResNet-50 are more recent ImageNet competition winners.

To cover a range of possible inference scenarios, we will consider two cases. The first case allows batching many input images together, to model use cases like inference in the cloud where thousands of users submit images every second. Here, large batches are acceptable, as waiting for a batch to assemble does not add significant latency. The second case covers applications that are extremely latency-focused; in this case, some batching is usually still feasible, but for our testing, we consider the low-batch case of batch size of two.

We compare five different devices: The NVIDIA Tegra X1 and X2 client-side processors, and the NVIDIA Tesla P4, V100 and the Intel* Xeon* data center processor. To run the neural networks on the GPU, we use TensorRT 2 EA, which will be released in an upcoming JetPack updated slated for release in 2Q'17. For the Intel* Xeon* E5-2690 v4, we run Intel* Deep Learning SDK v2016.1.0.861 Deployment Tool.

For all the GPU results, we run the "giexec" binary included in all builds of TensorRT. It takes prototxt network descriptor and caffe model files and populates the images with random image and weight data using a Gaussian distribution. For the CPU results, we run the "ModelOptimizer" binary with prototxt network descriptor and caffe model files to generate the .xml model file necessary to execute the "classification_sample" binary linked with MKL-DNN. We run the Intel* Deep Learning SDK Inference Engine using images from imagenet12 rescaled and reformatted to RGB .bmp files. Both TensorRT and Intel* Deep Learning SDK Inference Engine use image sizes of 227x227 for AlexNet and 224x224 for GoogLeNet, VGG-19, and ResNet-50. The Intel* Deep Learning SDK Inference Engine threw the "bad_alloc" exception when running with a batch size of one for all networks we tested. Instead we use Intel Caffe for batch size of one linked with MKL 2017.1.132 where we start with the default_vgg_19 protocol buffer files, and use Caffe's standard performance benchmarking mode "caffe time" with the same images as Intel* Deep Learning SDK.

We compare FP32 and FP16 results on V100, and FP32 and INT8 results on P4. All Tegra X1 and X2 results are using FP16. Intel* Deep Learning SDK only supports FP32 since Xeon* E5-2690 v4 does not have native support for reduced precision floating-point numbers.

To compare power between different systems, it is important to measure power at a consistent point in the power distribution network. Power is distributed at a high voltage (pre-regulation), and then voltage regulators convert the high voltage to the correct level for the system-on-chip and DRAM (post-regulation). For our analysis, we are comparing pre-regulation power of the entire application processor (AP) and DRAM combined.

On the Xeon* E5-2690 v4, Intel* Deep Learning SDK is running on only one socket. CPU socket and DRAM power are as reported by the pcm-power utility, which we believe are measured on the input side of the associated regulators. To measure pre-regulation (upstream) power for Tegra X1 and X2, we use production Jetson™ TX1 and TX2 modules both powered by a 9V supply. TX1 has major supply rails instrumented at the input side of the regulators, and TX2 has onboard INA power monitors. On the Tesla P4 and V100, we report the total board power consumed by a production cards using the NVSMI utility. We do not include the system CPU's power in our Tesla measurements as the entire computation is happening on the GPU; the CPU only submits the work to the GPU.