habana
Goya™

Goya™ Inference Platform & Performance Benchmarks

Rev. 1.6.1
January 2019

Habana Goya™ Inference Platform

Table of Contents

Table of Figures

1. Introduction

Machine Learning (ML), a subfield of Artificial Intelligence (AI), is no longer science-fiction. One prominent field within ML is Deep Learning, in which the models are Deep Neural Networks (DNNs). Many problems in multiple domains, which several years ago were considered difficult for machines to solve (object detection and classification in images/videos, speech recognition and more), are now solved as accurately as by human beings and more using deep learning models. As such, deep learning is a transformational technology.

A typical deep learning algorithm comprises of multiple operators, such as matrix multiplication, convolutions and other tensor operations, which add up to billions of compute-intensive operations. This massive amount of operations can be accelerated by using the inherent parallel processing that advanced GPUs offer. However, GPUs which are primarily designed to render graphics in a super-fast way, are not optimized for deep learning workloads. The existing solutions inefficiency for deep learning workloads has a severe impact on the operational cost of the cloud and data centers. To address this issue, a new class of software programmable AI processors are emerging, designed from the bottom-up for DNN workloads – Pure AI™ Processors.

Habana's Goya is a product line of AI processors dedicated to inference workloads. The HL-1000 processor is the first commercially available, deep learning inference processor, designed specifically to deliver superior performance, power efficiency and cost savings for cloud, data centers and other emerging applications.

For Information about the HL-100/101/102 line-cards, incorporating the HL-1000 processor, please see www.habana.ai

2. Deep Learning Workflows – Training and Inference

A deep learning workflow consists of two conceptual steps
- Training a model (training)
- Using a model on new data (inference)

For a model to address a specific use case, one first needs to train it and then use it (inference). Both training and inference have similar characteristics, but have different hardware resources usage requirements.

During training, a large dataset is processed to train a neural network model to distinguish between different statistical properties of the samples within the dataset. For example, recognizing images of apples from an arbitrary set of input images. After the model is ready for use (which means that the model met the accuracy goals set for successfully recognizing which images contained an apple and which did not), the model is ready for deployment. In a production environment, the model is used to efficiently recognize a new set of images to which it was not exposed during the training phase. This operation is called inference and the goal of this phase is to infer attributes in the new data using the trained model (in our case, whether there is an apple in the image).

Training workloads require high bandwidth memories with large capacity in addition to the memory requirements chip-to-chip communication. These requirements greatly increase solution BOM and its power consumption, and they are not needed for the inference workload. Low latency inferencing is critical in supporting real time applications such as neural machine translation, virtual assistant and many other applications. Providing high throughput in low batch sizes is critical for providing inference to different topologies, some of which may be latency sensitive. To provide comprehensive inference capabilities, an inference solution should provide high throughput, low latency, low power and be cost effective – Enter Goya!

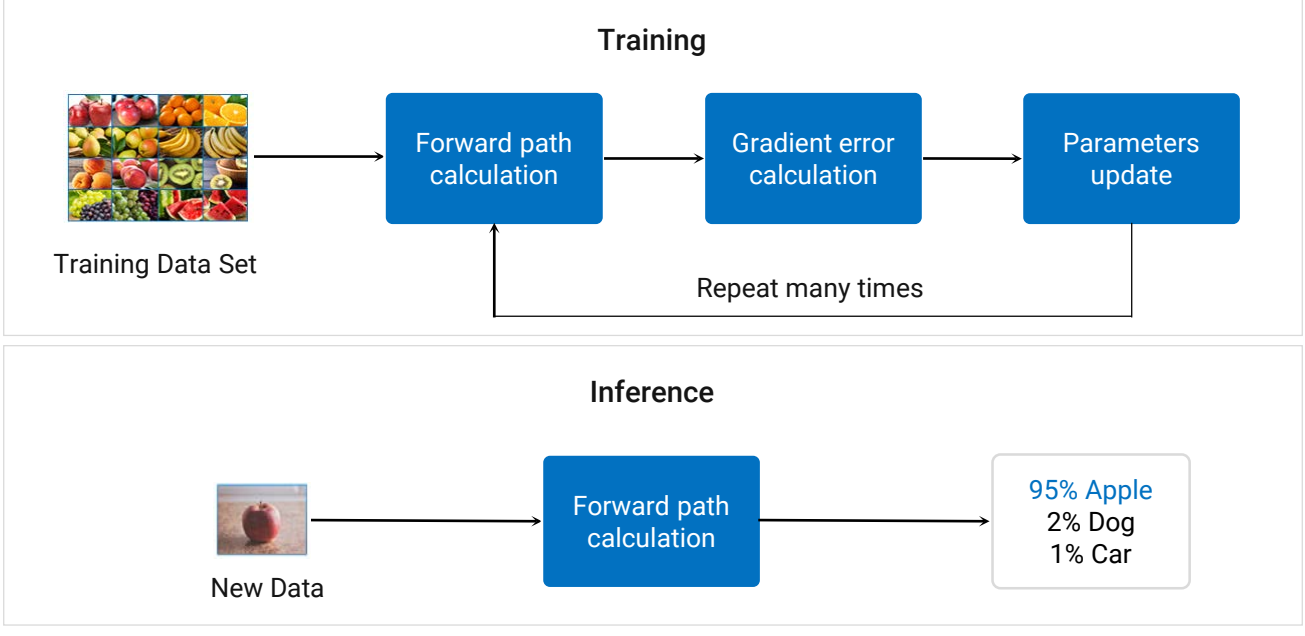

FIGURE 1: Deep Learning Workflows – Training and Inference

3. Goya™ Deep Learning Inference Platform

The Goya platform architecture has been designed from the ground up for deep learning inference workloads. It comprises a fully programmable Tensor Processing Core (TPC™) along with its associated development tools, libraries and compiler, which collectively deliver a comprehensive, flexible platform that greatly simplifies the development and deployment of Deep Learning systems for mass markets and cloud computing. The platform is capable of massive data crunching with low latency and high accuracy, as required by the workloads.

All major deep learning frameworks are supported, including TensorFlow, MXNet, Caffe2, Microsoft Cognitive Toolkit, PyTorch and Open Neural Network Exchange Format (ONNX).

Habana's software stack interfaces seamlessly with deep learning frameworks. The model is converted into internal representation. Following this step, ahead-of-time (AOT) compilation is used to optimize the model and create a working plan for the network model execution on the Goya hardware.

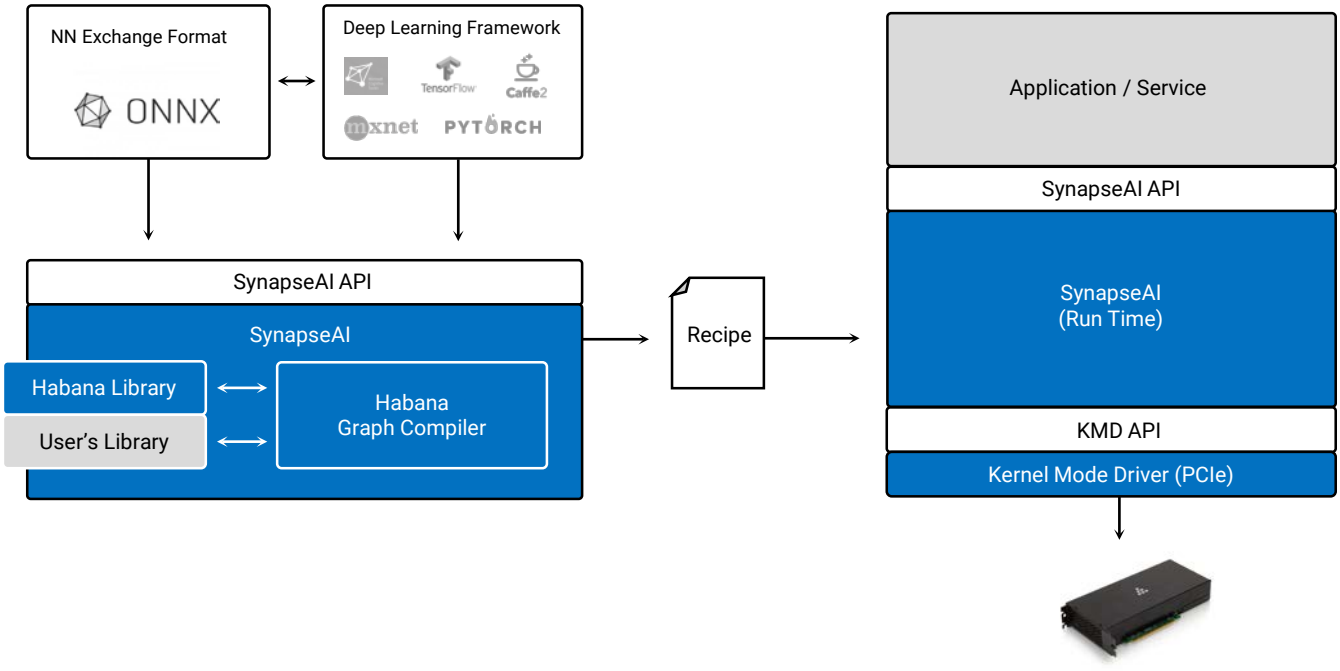

FIGURE 2: Goya Inference Platform – Software Stack

3.1. SynapseAI™ - Optimizer and Runtime

Habana Lab's SynapseAI is a comprehensive inference software toolkit that simplifies the development and deployment of deep learning models (topologies) for mass-market use. The SynapseAI software provides inference network model compilation (Graph Complier) and runtime.

The Goya platform is training-platform-agnostic. The DNN can be trained on any hardware platform (GPU, TPU, CPU or any other platform) to obtain a model (topology). SynapseAI imports the trained model and compiles it for use on the Goya™ platform. The result is an optimized execution code and plan in terms of accuracy, latency, throughput and efficiency.

SynapseAI supports automatic quantization of models trained in floating-point format with near-zero accuracy loss. It receives a model description and representative inputs, and automatically quantizes the model to fixed-point data types, thus greatly reducing execution time and increasing power efficiency.

The user can specify the level of performance gain required and whether some accuracy may be sacrificed to improve performance.

SynapseAI provides two APIs:
- The **C API** for describing a neural network to be executed on the platform.
- A **Python API** that can load an existing native framework (TensorFlow, MXNet, etc) or via ONNX (that can import from any framework).

The SynapseAI Run Time is the user mode driver. It is a layer between the user's code to Goya's PCIe driver that is used when inference is executed.

3.1.1. Example SynapseAI™ Workloads

Given TPC™ programmability, Goya is a very flexible platform. It enables quick adoption of different deep learning models and is not limited to supporting specific workloads or workloads from a specific domain. Goya™ supports models from various domains, including, but not limited to, vision (for example, object detection, classification, segmentation), NLP (for example, NMT, text classification) and speech (recognition, synthesis). Some examples of models fully executed on Goya with SynapseAI are described below.

Vision
- **ResNet (used for object classification in images)**
  The ResNet family of models is the most commonly used for performing vision-related tasks.

NMT
- **LSTM-based Neural Machine Translation**
  Neural Machine Translation (NMT) uses deep neural networks to translate sequences from one language to another. Sequence-to-Sequence (Seq2Seq) models have gained popularity for developing NMT applications due to their high accuracy.

Sentiment Analysis

- **LSTM-based Sentiment Analysis**
  This model converts text to metrics by analyzing the text and scoring it. Sentiment is broadly utilized in CRM systems for tracking notes. It rates each conversation and derives meaningful metrics from a CRM entry. Sentiment analysis enables a section of text to be rapidly scanned and provides a 1–10 rating about whether the text reflects a positive or negative conversation. We demonstrate an LSTM-based topology inspired by http://deeplearning.net/tutorial/lstm.html.

Recommender Systems

- **MLP-based Recommendation System**
  Multi-layer Perceptron (MLP) is the most fundamental construct in deep learning models. In particular, it can be useful for recommender systems and is deployed in many applications, such as media streaming, ad placement and online shopping. Recommender systems proposing specific items, based on past user actions or characteristics of the user and items. We demonstrate an MLP-based recommendation system that follows an autoencoder based collaborative filtering approach published in https://github.com/NVIDIA/DeepRecommender/.

4. Goya™ Processor High-level Architecture

Goya is based on the scalable architecture of the TPC uses a cluster of eight TPC cores. The TPC core was designed to support deep learning workloads. It is a VLIW SIMD vector processor with ISA and hardware that was tailored to serve these workloads efficiently.

The TPC core is C-programmable, providing the user with maximum flexibility to innovate, coupled with many workload-oriented features such as:

- GEMM operation acceleration
- Special functions dedicated hardware
- Tensor addressing
- Latency hiding capabilities

The TPC core natively supports the following mixed-precision data types –

- FP32
- INT32
- INT16
- INT8
- UINT32
- UINT16
- UINT8

To achieve maximum hardware efficiency, the data type is selected by the SynapseAI quantizer by balancing throughput and performance versus accuracy.

For predictability and low latency, Goya is based on software-managed, on-die memory along with programmable centralized DMAs. For robustness, all memories are ECC-protected.

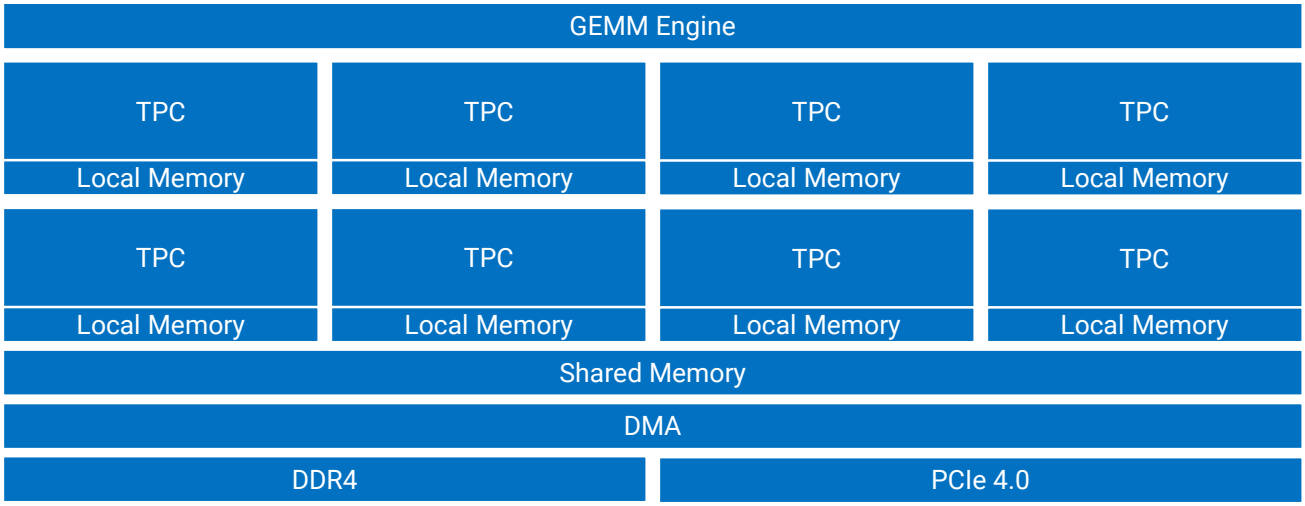

FIGURE 3: Goya High-level Architecture

4.1. Software Development Tools

SynapseAI enables users to unleash the power of deep learning by executing the algorithms efficiently using its high-level software abstraction. However, advanced users can do further optimization and add their own proprietary code using the provided software development tools. The Goya platform comes with state-of-the-art development tools, including visual real time performance profiling and TPC development tools for advanced users (including an LLVM-C compiler) to combine third-party TPC kernels. Thus, users can quickly and easily deploy a variety of network models and algorithms.

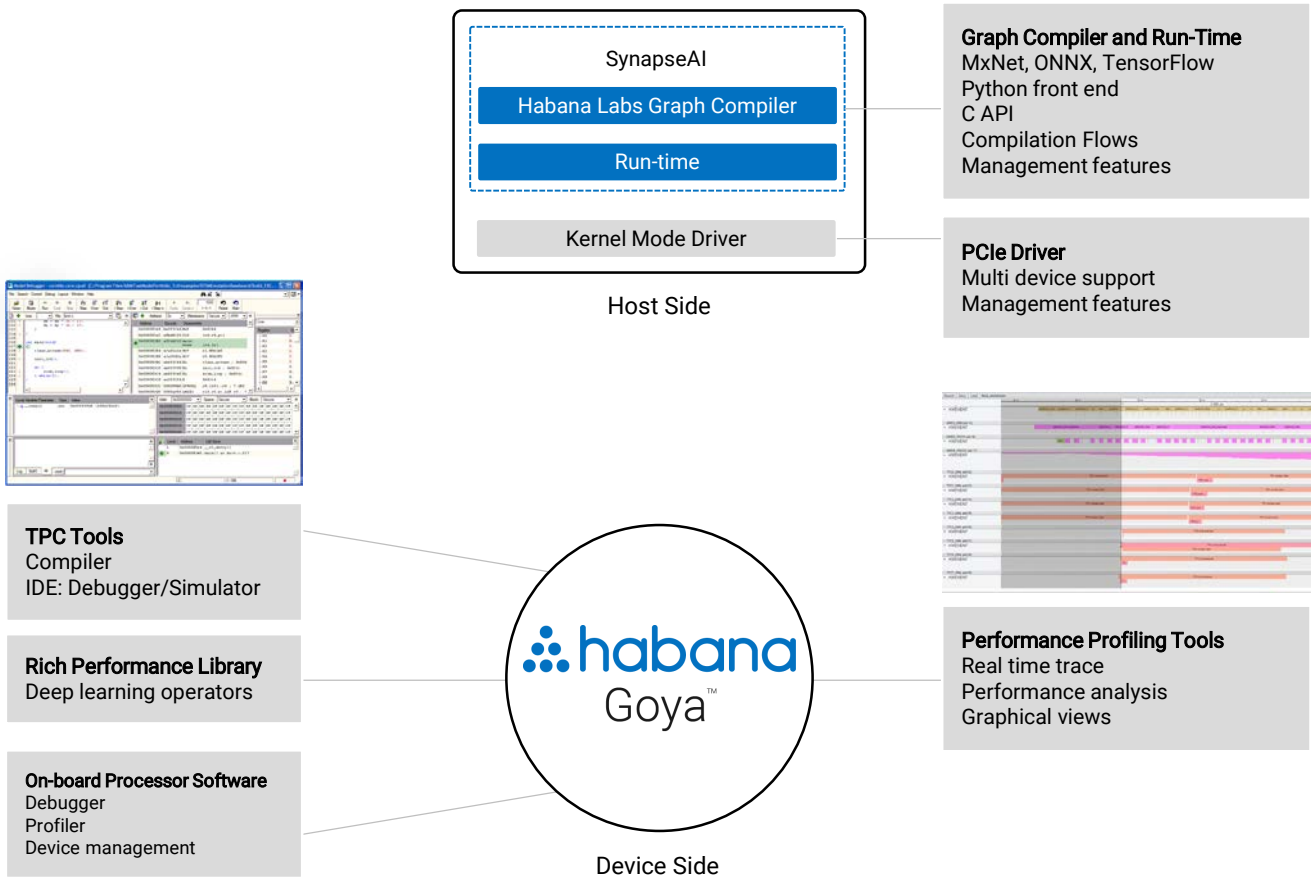

FIGURE 4: Goya Platform Software Developments Tools

5. Goya Inference Performance

The key factors that are used in assessing the performance of the Goya inference platform compared to other solutions are throughput (speed), power efficiency, latency and the ability to support small batch sizes. While consuming ~100 Watt, the Goya HL-100 PCIE card provides a throughput of ~15,000 images/second for a ResNet-50 workload at a latency of ~1 msec, which is well below the industry requirement of 7 msec. Moreover, Goya's performance is sustainable at a small batch size, which simpilifies its application.

Below are performance results for various topologies from Tensorflow, ONNX public repositories and in-house topologies based on public sources.
Software: Ubuntu 16.04, SynapseAI software release: 0.1.6
Hardware: Goya HL-100 PCIE card, Host: Intel Xeon E5

Topology	Source	Throughput [images/sec] per batch size				Latency [msec] per batch size				Goya HL-100 card		Model source
		1	4	8	10	1	4	8	10	Power [W]	Efficiency [Img/s/W]	
resnet18v1	ONNX	12,862	26,953	30,397	31,331	0.12	0.35	0.59	0.66	109	287	https://s3.amazonaws.com/onnx-model-zoo/resnet/resnet18v1/resnet18v1.onnx
resnet34v1	ONNX	7,994	15,657	17,410	18,288	0.18	0.52	0.77	0.9	116	157	https://s3.amazonaws.com/onnx-model-zoo/resnet/resnet34v1/resnet34v1.onnx
resnet50v1	ONNX	6,879	13,225	14,542	15,393	0.29	0.58	0.87	1.01	103	149	https://s3.amazonaws.com/onnx-model-zoo/resnet/resnet50v1/resnet50v1.onnx
resnet50v1	Tensorflow	6,399	12,135	13,538	14,353	0.21	0.58	0.92	1.06	115	125	http://download.tensorflow.org/models/official/20181001_resnet/savedmodels/resnet_v1_fp32_savedmodel_NHWC.tar.gz
resnet101v1	ONNX	4,342	7,013	8,939	8,971	0.43	0.87	1.24	1.5	120	75	https://s3.amazonaws.com/onnx-model-zoo/resnet/resnet101v1/resnet101v1.onnx
resnet152v1	ONNX	1,282	3,430	4,270	4,560	1.04	1.54	2.3	2.77	99	46	https://s3.amazonaws.com/onnx-model-zoo/resnet/resnet152v1/resnet152v1.onnx
resnet152v1	Tensorflow	1,290	3,467	4,893	5,259	0.99	1.39	1.96	2.3	107	49	http://download.tensorflow.org/models/resnet_v1_152_2016_08_28.tar.gz
inception_v1	Tensorflow	10,981	17,805	19,330	19,487	0.16	0.47	0.74	0.87	91	214	http://download.tensorflow.org/models/inception_v1_2016_08_28.tar.gz
ssd300_vgg16	Inhouse	980										SSD for 300x300 images with VGG16 backbone, based on https://github.com/zhreshold/mxnet-ssd/blob/master/symbol/legacy_vgg16_ssd_300.py
Googlenet	ONNX	4,130				0.45				66	63	https://s3.amazonaws.com/onnx-model-zoo/resnet/resnet18v1/resnet18v1.onnx
Googlenet_bn_no_lrn	Inhouse	10,502	16,223	17,484		0.14	0.49	0.77		88	199	Developed inhouse based on Googlenet with batch norm and without LRN (see section 5.1 on motivation)

Table 1: Goya Inference Performance Benchmarks

5.1 Googlenet

The Googlenet topology was developed in 2014, opting to use Local Response Normalization (LRN) over Batch Normalization. LRN is more computationally expensive and we've improved throughput when replacing it with BN – in fact while also improving accuracy, 72% top1 versus 68% originally.
As this reduces the amount of computation while improving accuracy, we expect it to perform better on any inference processor and are offering this improved topology (Googlenet_bn_no_lrn) on demand.
Other than replacing LRN with BatchNorm, the topology has exactly the same structure.

6. Summary

Deep learning revolutionizes computing, impacting enterprises across multiple industrial sectors. Neural networks are becoming exponentially larger and more complex, driving massive computing demand and cost. Modern neural networks are too compute-intensive for traditional CPUs and GPUs, and the future belongs to dedicated high throughput, low latency, low power AI processors.

Inference performance is measured by throughput, power efficiency, latency and accuracy, all of which are critical to delivering both data center efficiency and great user experiences. Goya HL-1000 is a world class, high performance, production ready AI processor for inference workloads, supported by state-of-the-art software development tools, and a full software stack.

An effective deep learning platform must have the following characteristics:

- It must have a processor that is custom-built for deep learning.
- It must be software-programmable.
- Deep learning frameworks must be executed efficiently on the platform, powered by a developer ecosystem that is accessible and adopted around the world.

Habana's Goya Deep Learning Inference Platform meets all these requirements.