upstride

ENHANCING NEURAL NETWORKS COMPUTATIONAL EFFICIENCY

AUTHORS

Wilder Lopes, PhD, CTO, UpStride

Mohammad Reza Loghmani, PhD, Deep Learning Scientist, UpStride

A FRESH START

The last decade has raised evidence that, given the standard model of computation used to build artificial neural networks (NNs), we will not be able to provide the right infrastructure for scaling artificial intelligence (AI) and deploying it everywhere. The increasing demand for computational power to train NNs, the need for larger and larger datasets, and the increase in the sheer size (trainable parameters) of NNs have raised the cost of building and scaling AI. This scenario has pushed scientists and engineers to rethink the way computers process information and pave the way for a complete integration of AI into our lives.

Motivated by the current context and standing on top of solid research foundations, we drift away from the established design rules of computation and leverage alternative algebraic methods to increase the representational power of Convolutional Neural Networks (CNNs).

Our research and experiences with industrial clients focus on the development of a technology that enhances computer-vision algorithms, to acquire more information from each sample and benefit deployment of NNs on the edge. Our models can achieve the same performance as standard NNs, but with a fraction of the memory. In addition, when hardware resources are not highly constrained, our models achieve higher performance than their standard counterparts.

Ultimately, UpStride is building a software suite, based on its proprietary technology, to provide these benefits to AI-powered companies. Our software seamlessly integrates more sophisticated mathematics in the background of standard high-level Deep Learning (DL) frameworks, such as Tensorflow. In other words, we are providing new tools for a fresh start in NN computation.

THE RISKS OF TAKING COMPUTING FOR GRANTED

For the past five decades, we have been focusing on scaling the basic model of computation that unlocked the start of the computing era in the 1960s. From mainframes to microchips, information inside computers is organized in tables (vectors and matrices) and operated via addition and multiplication following a linear-algebra formulation.

This mathematical paradigm has served science and engineering very well for decades. Its proven track record of success allowed the industry to continuously scale it by improving the efficiency of integrated circuits and growing the size and capacity of servers. The phrase "just add one more GPU and we will be fine" perfectly captures the feeling of confidence that most people have on the current model of computation based on linear algebra. However, a closer look at the evolution of DL in the last decade shows that the problem of computing is far from being solved. First, DL requires huge amounts of data to unlock its true potential. Second, the computation necessary to train the most state-of-the-art NNs grows exponentially, roughly doubling every 3.4 months (Amodei et al. 2018[1]). This data points to a scenario where AI building capabilities will be concentrated in the hands of big tech companies, the only ones to have access to big data and a massive computational power.

This realization has rekindled the interest of part of the scientific community in alternative computational methods. Expressions like "hardware heterogeneity" and "software/hardware co-design" (Intel 2020[2]), originally part of high performance computing and integrated-circuits jargon, have started to appear more frequently in AI literature. Newly released AI accelerators, such as Google's Tensor Processing Unit (TPU), improve NNs processing time thanks to a new design of the communication bus between memory and computing in the chip. Instead, **UpStride approaches the problem from a different and complementary perspective by questioning linear algebra as the mathematical language of NNs**.

REDESIGNING THE FUNDAMENTALS OF NEURAL NETWORKS COMPUTATION

We introduce a new information-processing pipeline for NNs. Our pipeline combines a novel unit of information with algebraic methods that generalize mathematical operations such as addition and multiplication.

Everything starts with the way data is represented when flowing in the computational graph of a NN. UpStride created Hycor, a data structure carefully designed to take advantage of geometric features present in the data. Hycor's features include efficient parametrization of rigid-body transformations, such as 2D/3D rotation, and enhanced multiplication and addition operations. One can benefit from those characteristics by using **Hycor as the fundamental package of information** when performing NN computations.

Figure 1: The Hycor data structure

[1] Amodei, Dario, Danny Hernandez, Girish Sastry, Jack Clark, Greg Brockman, and Ilya Sutskever. 2018. "AI and Compute." OpenAI. May 16, 2018. https://openai.com/blog/ai-and-compute/

[2] Intel. 2020. "Harnessing the Power of a Heterogeneous Computing Future." August 5, 2020. https://blogs.intel.com/technology/2020/08/harnessing-the-power-of-a-heterogeneous-computing-future/

From a code point-of-view, Hycor is implemented as a new structure in C++ 11, and heavily leverages compiler optimizations to run on top of diverse architectures such as x86, Nvidia's GPUs, and ARM. In addition, the flexibility of our **proprietary technology** makes it possible to adapt it to custom architectures.

To be easily integrated within existing pipelines, Hycor is packaged in the UpStride Engine, a piece of software that uses Hycor to optimize the implementation of all layers and functions necessary to build CNNs. For instance, two-dimensional convolutions and activation functions such as ReLU and sigmoid are rewritten on top of Hycor inside the Engine, offering a familiar way for users to leverage our core technology. This way, the main role of the Engine is to **raise the level of abstraction** from data structure (Hycor) to functions that can be called by other frameworks and libraries.

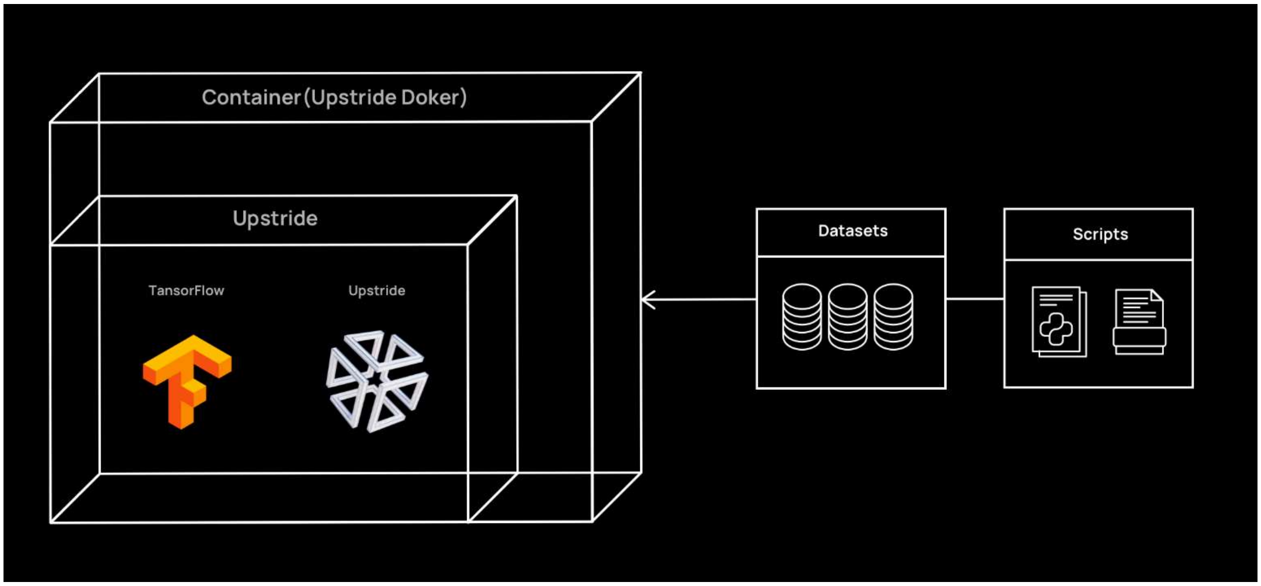

Figure 2: The UpStride Engine

The UpStride Engine is fully compatible with computational libraries used by AI builders such as **Nvidia CUDA/CuDNN** and **Intel's OneAPI**. This allows for promptly connecting UpStride with the backend of **DL frameworks** like Tensorflow, enabling users to start building Hycor-powered AI without having to learn to code with a new library.

DATA AND POWER EFFICIENCY AT THE CORE OF OUR TECHNOLOGY

DATA EFFICIENCY

Every dataset, independently of its size, stores valuable information about the observed phenomenon. However, given the way AI software and hardware operate today, a lot of information contained in the input dataset is not easily recovered, requiring numerous passes of data to the NNs. Additionally, performance improvement becomes more and more dependent on data-augmentation techniques. This increases the burden on small and medium sized enterprises that, differently from big tech companies, have great challenges exploiting the content from their inherently small datasets.

To efficiently learn from data and optimize the use of datasets, one needs to focus on extracting more information from each sample, essentially reducing the time to achieve the accuracy required for deployment in production. The UpStride Engine (via the Hycor-based functions implemented there) enables exactly that, by taking advantage of the correlation -- or more precisely, the **mutual information** -- between the color pixels sitting in different channels of the same image sample.

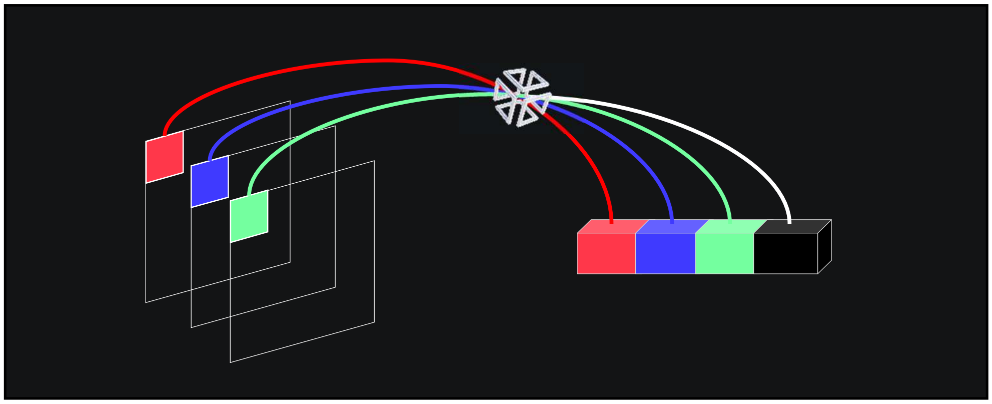

Figure 3: Allocation of pixels into a Hycor

In a computer vision application with color images, the mutual information between the red, blue, and green pixels can be naturally exploited by packing the pixels together in the same Hycor, as depicted in Figure 3.

More importantly, the scalar-valued weights of traditional NNs are substituted by Hycor units to better exploit our enhanced mathematical operations when generating feature maps.

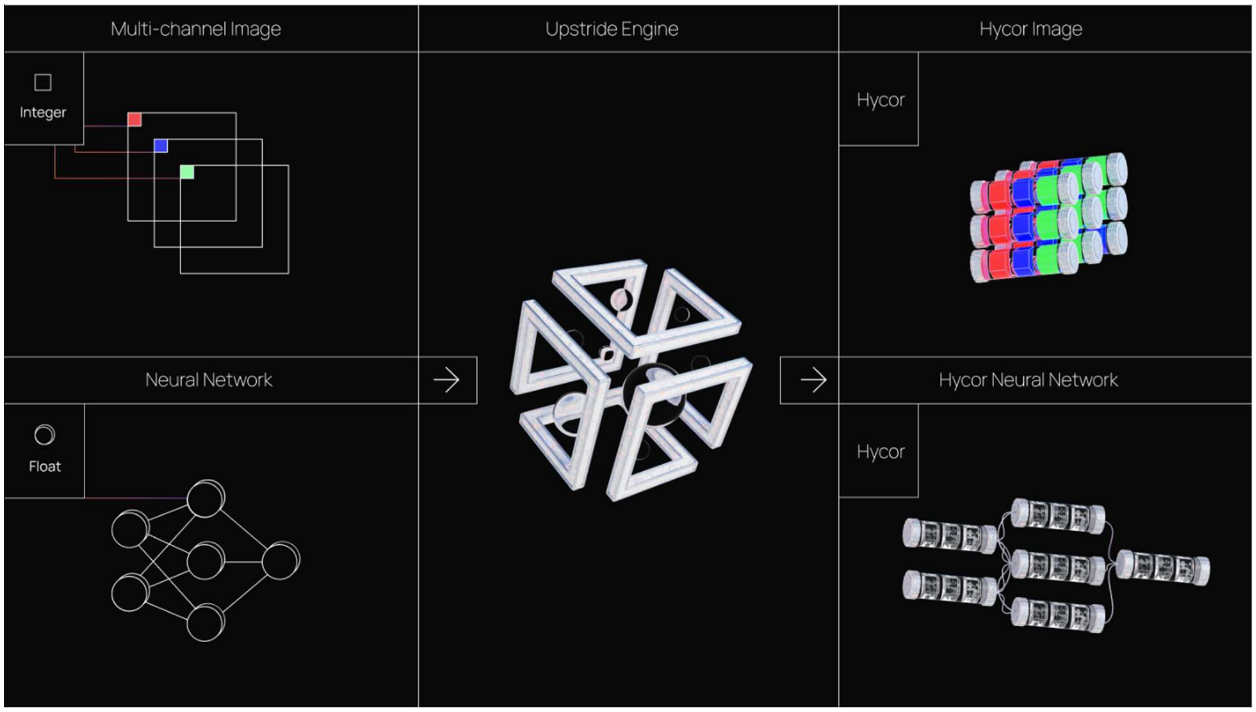

Figure 4: Converting from Tensorflow to UpStride

This process, which occurs inside the **UpStride Engine**, allows us to enhance the information flow through the NN computational graph, ultimately allowing us to acquire more information each time the NN processes the picture.

Let us compare the way UpStride and standard NNs process information. Given two vectors, representing features and weights respectively, a **standard NN** processes them using the inner product, a fundamental operation in linear algebra.

The inner product of two vectors results in a scalar number whose value is a function of the magnitude of the vectors and the angle between them. Such a scalar is therefore a way to represent the relationship between the two considered vectors.

Instead, UpStride's Hycor maps vectors onto manifolds (not only simple scalars) via its enhanced multiplication. From those multi-dimensional surfaces (manifolds) one can recover extra information such as space curvature, which can provide extra insights about the relationship between the vectors.

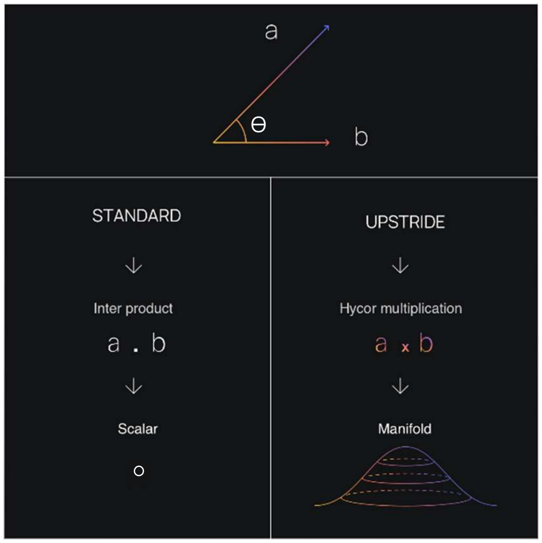

Figure 5:
Overview of the mathematics underlying UpStride's technology

In other words, NNs built with UpStride's Hycor benefit from that **redesigned multiplication**, increasing the efficiency of the data-processing pipeline and allowing for extracting more information from data

MODEL COMPACTNESS

Deployment on the edge is challenging as it requires the NN to comply with speed, storage, memory, and power specifications. The current standard practice is to compact the NN using compression techniques, such as quantization and pruning. These techniques consist of first training the NN on a powerful hardware and then reducing the size of the NN by either removing the least relevant weights or reducing their bit representation.

However, when training a NN, results heavily depend on the input data, the network topology and the utilized hardware. All of these factors can result in accuracy waste when converting the master model (non-compressed and targeted to datacenters) to be deployed in the edge.

UpStride's technology **prevents accuracy loss in the edge-deployment pipeline**. Due to the characteristics of Hycor, CNNs trained with UpStride are able to naturally achieve the required accuracy while keeping the number of parameters lower than their standard counterparts. Essentially, Hycor's higher representation efficiency allows the same learning capacity as standard NNs with a smaller model. If needed, Hycor can also be combined with standard compression techniques to further reduce the memory footprint.

On the following page, we present a simple example of UpStride's intrinsic model-compression capabilities. Figure 6 depicts the accuracy (log-evidence) versus number of parameters for two versions of the same NN architecture, one built with UpStride and the other built with standard Tensorflow. The plot shows that when the number of floating-point parameters grows beyond the flipping point (in this case, approximately 4.5M), the UpStride-based implementation outperforms its Tensorflow counterpart in terms of accuracy. Moving further beyond the region of 10M parameters, we note that:

- The highest log-evidence achieved by Tensorflow (-88) can be attained by UpStride with 90% less parameters. This significant reduction in the total number of parameters allows users to train NNs that are lighter and easier to deploy, especially in edge-computing environments;

- If the total number of parameters is not a constraint, one can use the UpStride-based version in order to obtain an absolutely higher accuracy, surpassing Tensorflow's log-evidence by 4.5

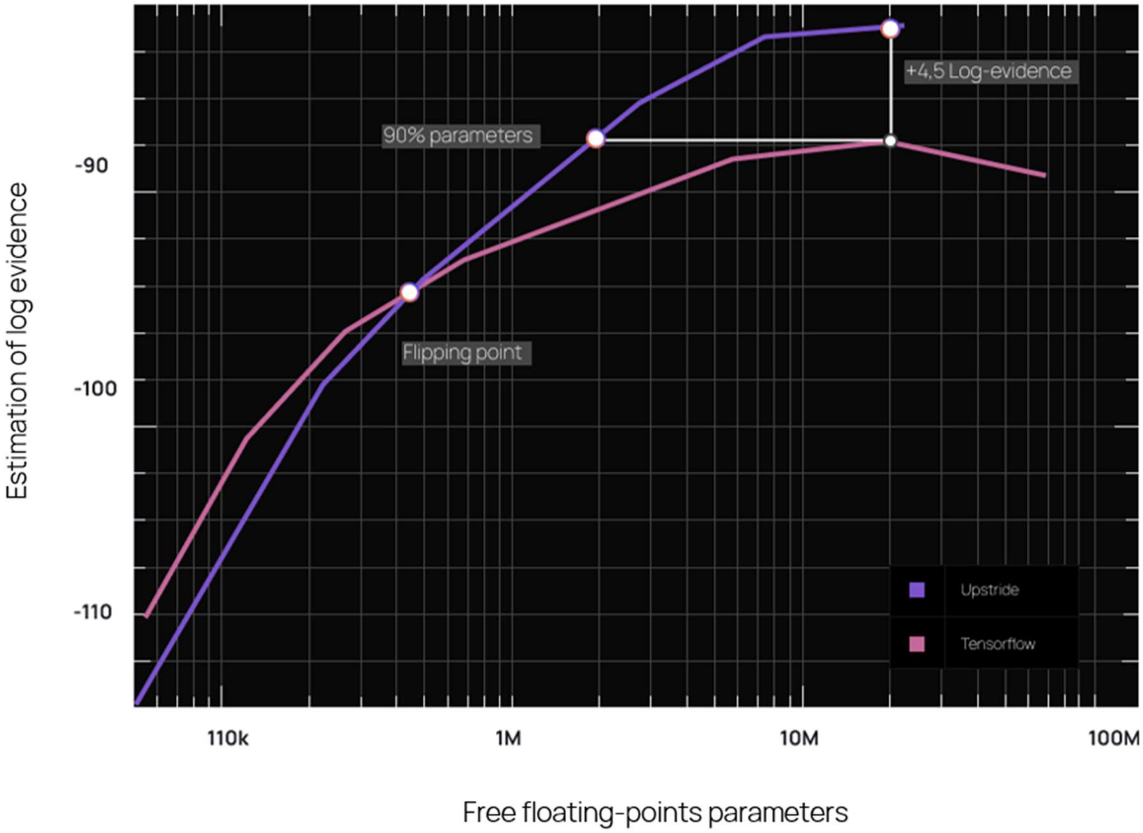

Figure 6: Comparing UpStride against Tensorflow. 3-Layer MNIST autoencoder.

Beyond this research example, industrial clients have been able to benefit from performance gains while using UpStride in **real-life** classification use cases with various NN architectures and datasets.

These successes motivate us to keep enlarging the scope of our R&D to cover more computer-vision use cases across different industries. We will continue to enforce the transfer of knowledge from R&D to efficient production, incorporating the latest research results into our product to empower AI builders all over the world.

Learn more about UpStride and get in contact by visiting www.upstride.io