DEEPCUBE

CubeEngine

CubeEngine PRODUCT BRIEF

ACCELERATE DEEP LEARNING INFERENCE PERFORMANCE

Run DNN models at 10X performance over any x86 compatible hardware

MAIN FEATURES

- Composable sparse inference engine, designed with multiple dynamically linked low-level libraries and DeepCube's sparse-pattern kernels
- Designed with C++ 14 and Assembly
- Supports mixed precision
- Throughput strong scaling with number of threads (90%)
- Scales across multi-socket
- Latency mode scales with number of threads
- Cross OS support for both Linux and Windows
- SDK available with examples
- C++, Python and REST APIs
- Available on-prem, cloud or at the edge. Packaged as containers, bare-metal or as C++ library
- Executes DeepCube optimized models (DCM) generated by CubeIQ
- Tested with Intel and AMD servers (Xeon, Core i7/8/9, EPYC, Ryzen and more)

Get Started Today

Self service CubeEngine portal to experiment, benchmark and evaluate

www.deepcube.com/portal

Executing DNN inferences at scale

Executing machine inferences is hard – compute budget, cost budget, latency and many other metrics are all critical to a feasible ML deployment. The first-generation of inference engines focused on a model's coverage (supporting all neural networks operators).

The first-generation engines were monolithic in nature and had one execution flow for all models. These 10-years old implementations fall short of the ML-at-scale challenge. Today a wave of innovation is sweeping the ML market – models are thinned out and compressed, without losing accuracy (primarily with sparsification). Hence, models can carry 90% less parameters (i.e. zero-out parameters).

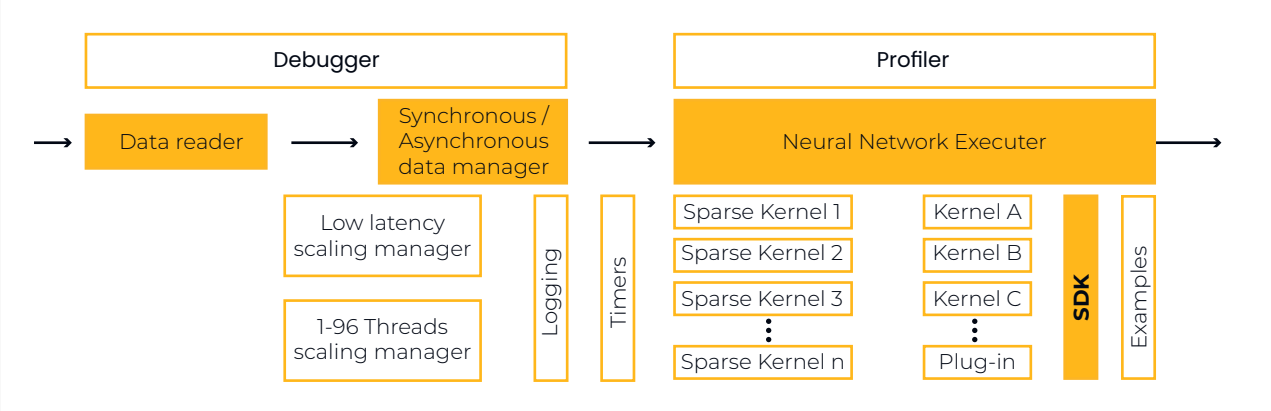
In theory, a new generation of inference engines, which can execute inferences by cleverly skipping zero's and unuseful patterns can enable a 10X speed-up or a massive latency improvement. That's what we have created with CubeEngine.

Why CubeEngine

In the past few years, DeepCube researchers have accomplished several breakthrough algorithmic enhancements, which apply structure pruning during training to any neural model. It implies that weights can be zero-ed in unique patterns: block, tile, diagonal, mix, etc. Secondly, we have been able to distill for each compute architecture the unique compressed-pattern that will execute optimally the model.

Those sparse patterns are associated with the processor micro-architecture, processor instruction sets (and extensions) and the sparse kernels. The end-result models with a specialized prune-pattern are fed to CubeEngine's inference software - which is designed to execute the patterns optimally with the specific processor. Essentially, we morphed the x86's software to a GPU-like machine.

CubeEngine



CubeEngine architecture

Two cutting-edge foundations drive CubeEngine's architecture:

Composability

Every model, every sparse pattern in each processor architecture is unique and hence calls out a different set of sparse kernels. Unlike prior generation inference engines, which are designed as a monolithic execution flow, CubeEngine dynamically rearranges kernels to execute a specific model in a specific processor architecture with a specific sparse pattern

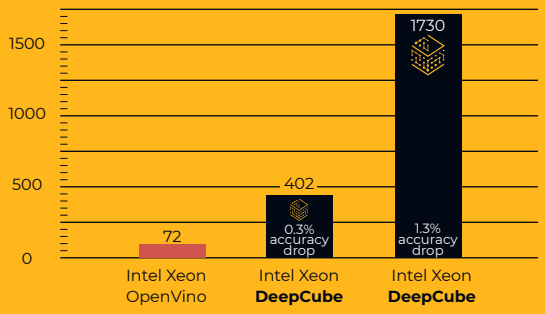
Optimized sparse kernels

Designed by DeepCube's low-level C++/Assembly team (CUDA and OpenCL pioneers) to fit the exact model patterns generated during training. Therefore, model sparsity is directly translated to increased speed and latency reduction

CubeEngine runs models with extreme throughput or reduced latency. Some of our benchmarks are listed below.

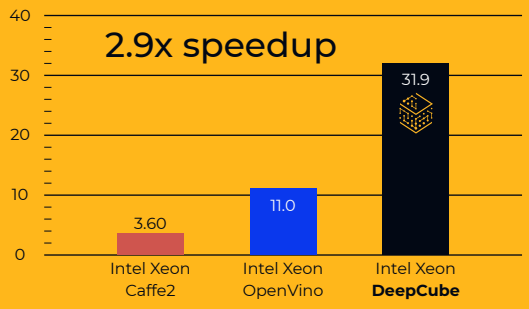
MLP (Ranking)

Ranking Algorithm (inferences/second)



Computer Vision

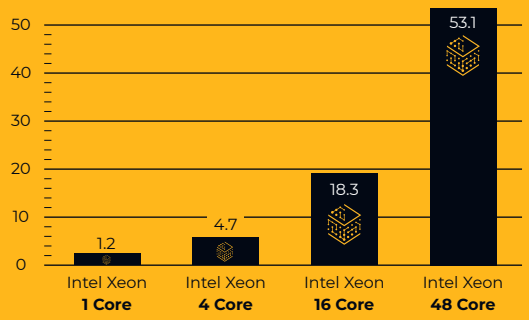
ResNet-50 frames per second (FPS) single thread



CubeEngine can scale up and down with the number of assigned threads/cores, and it provides near-linear performance scaling

Strong Scaling

Ranking Algorithm (inferences/second)



Start Now

A full benchmark suite can be accessed at

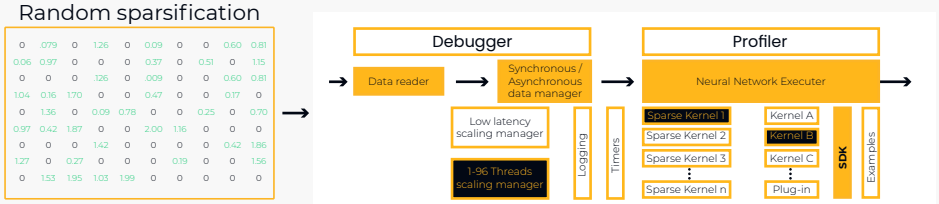
www.deepcube.com/portal

End-to-end ML platform

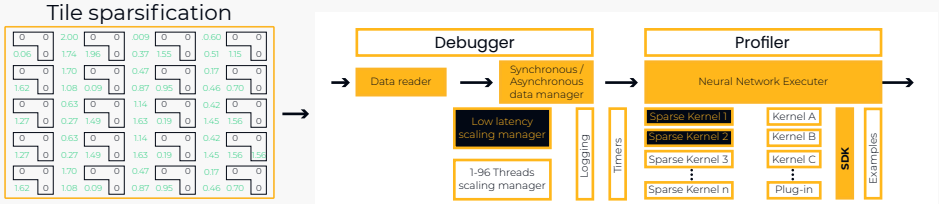
DeepCube built an end-to-end ML flow to accelerate deployments and performance. Models are first compressed with CubeIQ – our training framework, to accomplish 90% compression. Then the model is deployed and executed by CubeEngine at the target x86 hardware.

The end-to-end workflow is what creates the accelerated performance. The chosen sparsity pattern executed in the training process by CubeIQ matches the sparse kernels that execute the model. This match is unique to different models and model families. For CNN, one pattern may apply, for BERT a different pattern, for MLP yet another and the list goes on.

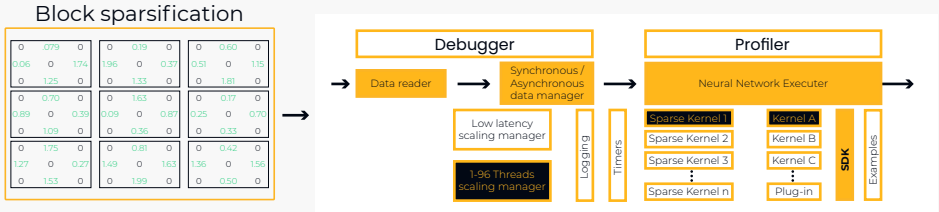
Random sparsification



Tile sparsification



Block sparsification



Ease of use – designed for scale

As a second-generation inference engine, CubeEngine is designed with simplicity and ease of integration in mind. It comes as an SDK with examples and it has a data reader with multiple modes to allow different batch sizes. Additionally, it has easy to manipulate configuration knobs (number of threads, latency mode, throughput mode, etc.). We also built it to be OS agnostics, so both Linux and Windows are available. APIs are flexible: C++ , Python and REST APIs.

Start with CubeEngine today

You can start evaluating CubeEngine with our self-service portal and gain access to a self explanatory Jupyter notebook and model zoo today by signing up on www.deepcube.com.

Start your own project pilot by getting a free trial license and installing CubeEngine on your target x86 machine - in the cloud, on-premise or at the edge. We will be able to provide broad, in-depth assistance for your ML project, by tapping into our ML experts, and tailor a solution to fit your needs with the optional CubeAdvisor service.

Contact us anytime at contact@deepcube.com to make your ML deployments meet a new level of economics and stretch goals.

About DeepCube

DeepCube is an award-winning deep learning pioneer that provides the industry's first software-based deep learning acceleration platform that drastically improves performance on existing hardware. Modeled after the way the human brain develops during childhood, DeepCube's patented technology is the first to be purpose-built for deployment of deep learning models in data centers and on intelligent edge devices. Its proprietary framework can be deployed on top of any existing hardware, resulting in drastic speed improvement and memory reduction.

Led by a team of experienced deep learning researchers and developers, DeepCube has patented numerous innovations, including methods for faster and more accurate training of deep learning models, and drastically improved inference performance.

For more information, please contact us at: www.deepcube.com | in | contact@deepcube.com