Configurable Cloud-Scale Real-Time Deep Learning

Bita Rouhani

Senior Research Manager
Cloud AI Systems & Technologies (CAST)
Microsoft Azure

AI/DL ubiquitously fuels our technology

Text Analytics
Translator Text
Bing Spell Check
Anomaly Detector

Language

Language Understanding

Decision

Content Moderator
QnA Maker
Personalizer
Content Moderator
Speech transcription
Conversation transcription capability
Custom Speech

Speech

Text-to-Speech
Neural Text-to-Speech
Bing Custom Search
Bing Video Search
Bing Entity Search
Bing News Search
Bing Local Business Search

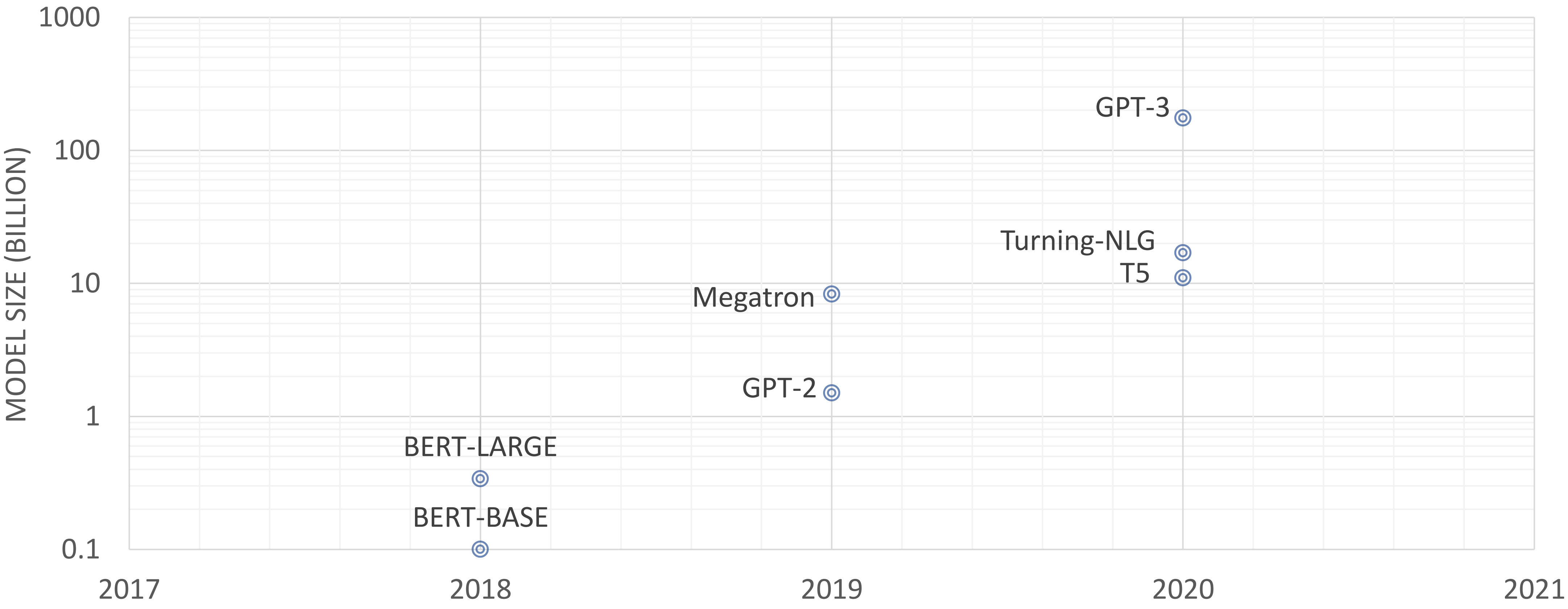
Web search

Bing Web Search
Bing Image Search
Bing Autosuggest
Bing Visual Search
Ink Recognizer
Computer Vision
Custom Vision

Vision

Video Indexer
Form Recognizer

Scale of the model plays a key role in the quality of the AI solution



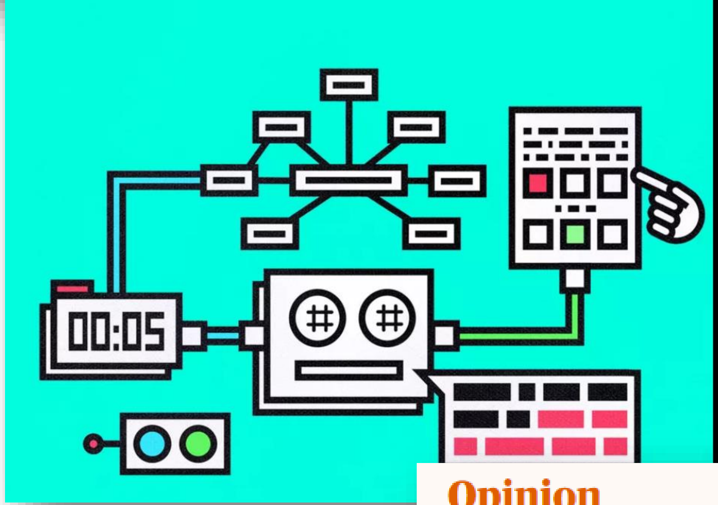
GPT-3: Powerful language model and generator

GPT-3, explained: This new language AI is uncanny, funny — and a big deal

Computers are getting closer to passing the Turing Test

By Kelsey Piper | Aug 13, 2020, 9:50am EDT

THE VERGE



REPORT

OPENAI'S LATEST BREAKTHROUGH IS ASTONISHINGLY POWERFUL, BUT STILL FIGHTING ITS FLAWS

The ultimate autocomplete

By James Vincent | Jul 30, 2020, 10:01am EDT

Opinion

GPT-3: an AI game-changer or an environmental disaster?

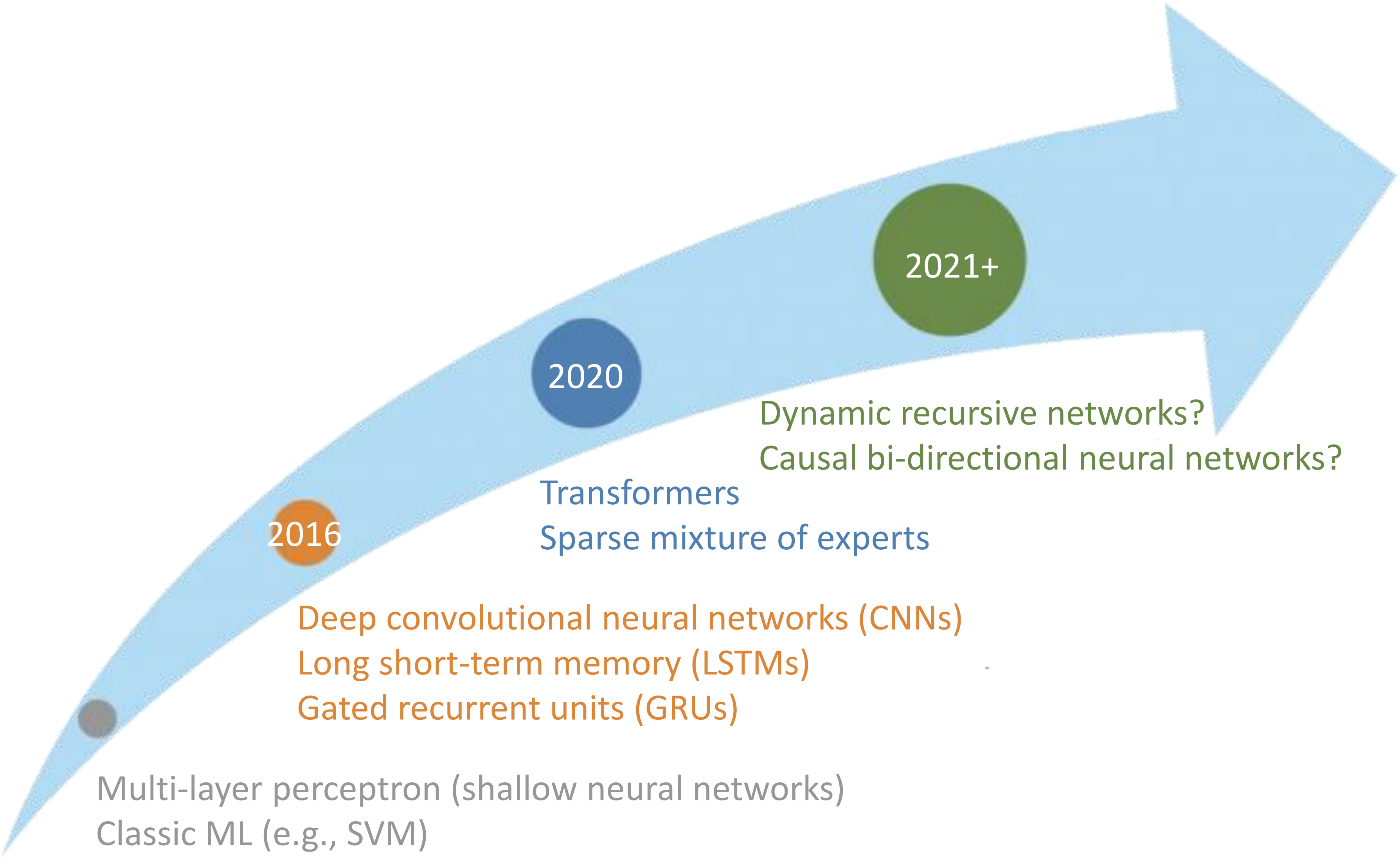
https://www.vox.com/future-perfect/21355768/gpt-3-ai-openai-turing-test-language
https://www.theverge.com/21346343/gpt-3-explainer-openai-examples-errors-agi-potential
https://www.theguardian.com/commentisfree/2020/aug/01/gpt-3-an-ai-game-changer-or-an-environmental-disaster

"The Industrial Revolution has given us the gut feeling that we are not prepared for the major upheavals that intelligent technological change can cause. There is evidence that the world began to collapse once the Luddites started smashing modern automated looms. It is therefore important to use reason and the faculty of wisdom to continue the changes as we have done before time and time again."

GPT-3, Editorial, The Guardian, September 8, 2020

Source: https://www.theguardian.com/commentisfree/2020/sep/08/robot-wrote-this-article-gpt-3

Dominant AI building blocks evolve rapidly



AI infrastructures should be:

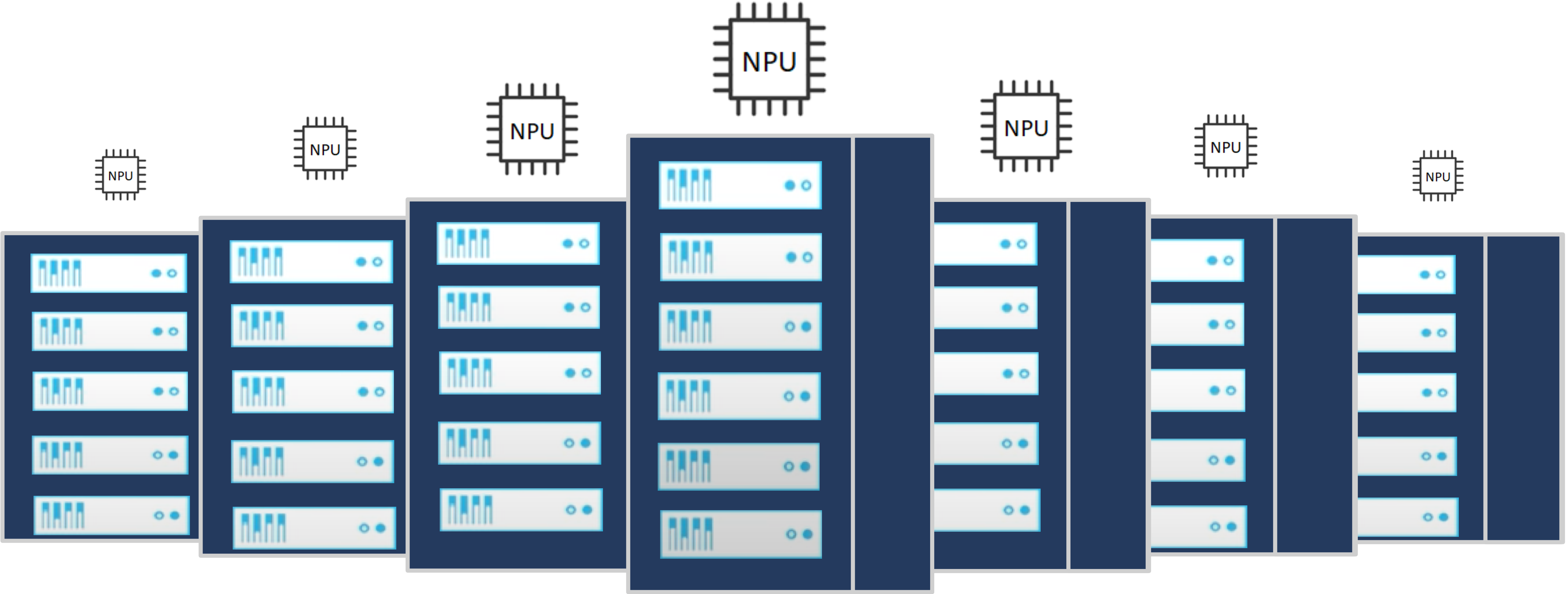
- Scalable
- Future proof
- Sustainable

Project Brainwave

Adaptable

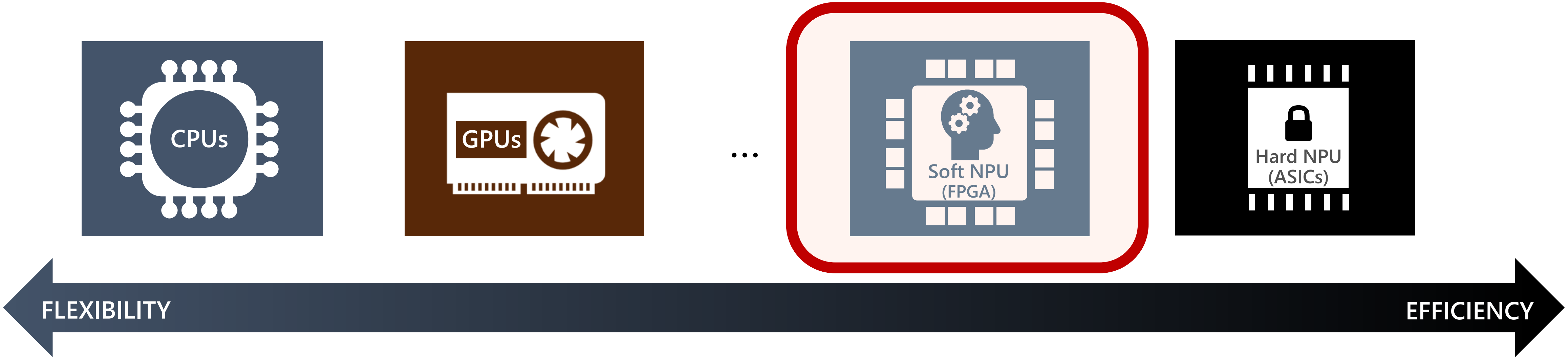
Low Latency

Low Cost



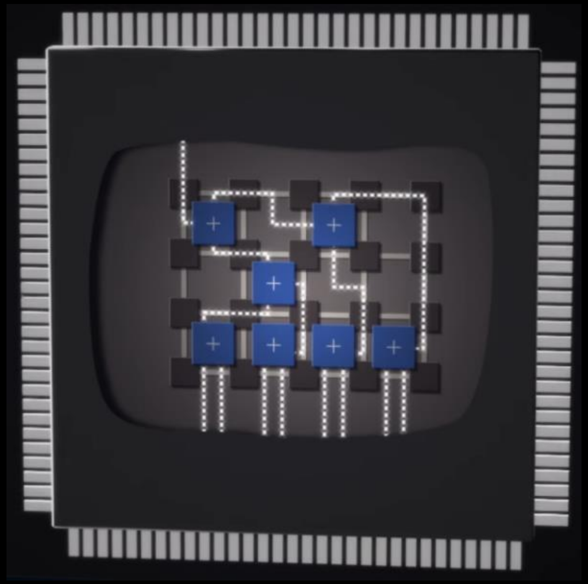
Inside Project Brainwave

Silicon alternatives for AI models



Project Catapult + Brainwave History

Field Programmable Gate Arrays



2011: Project Catapult Launched

2013: Bing pilot runs decision trees 40X faster

2015: Bing ranking throughput increased 2X

2016: Azure Accelerated Networking delivers industry-leading cloud performance

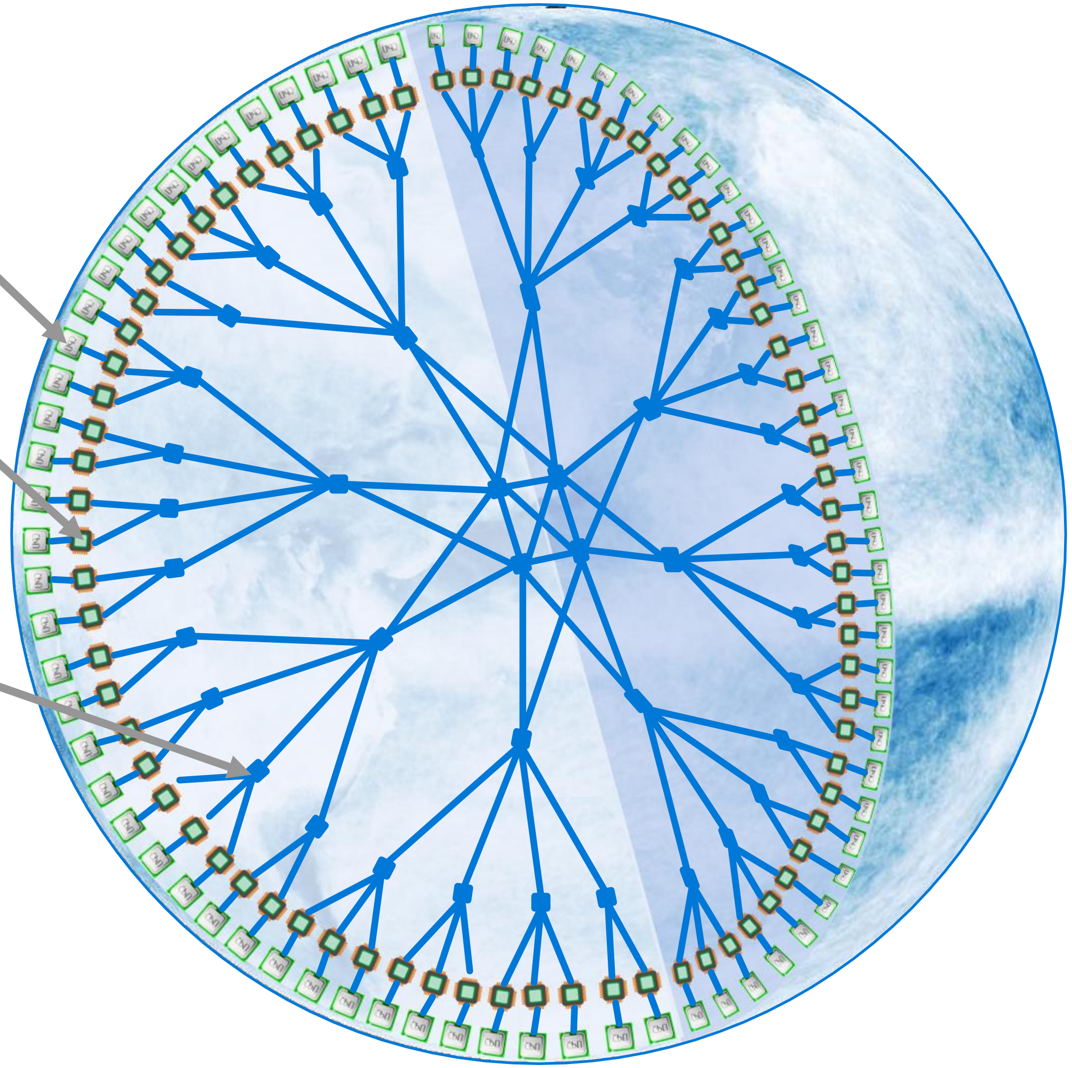
2017: Over 1M servers deployed with FPGAs at hyperscale

2017: Hardware Microservices harness FPGAs for distributed computing

2017: FPGAs enable real-time AI, ultra-low latency inferencing without batching; Bing launches first FPGA-accelerated Deep Neural Network

2018: Project Brainwave launched in Azure Machine Learning

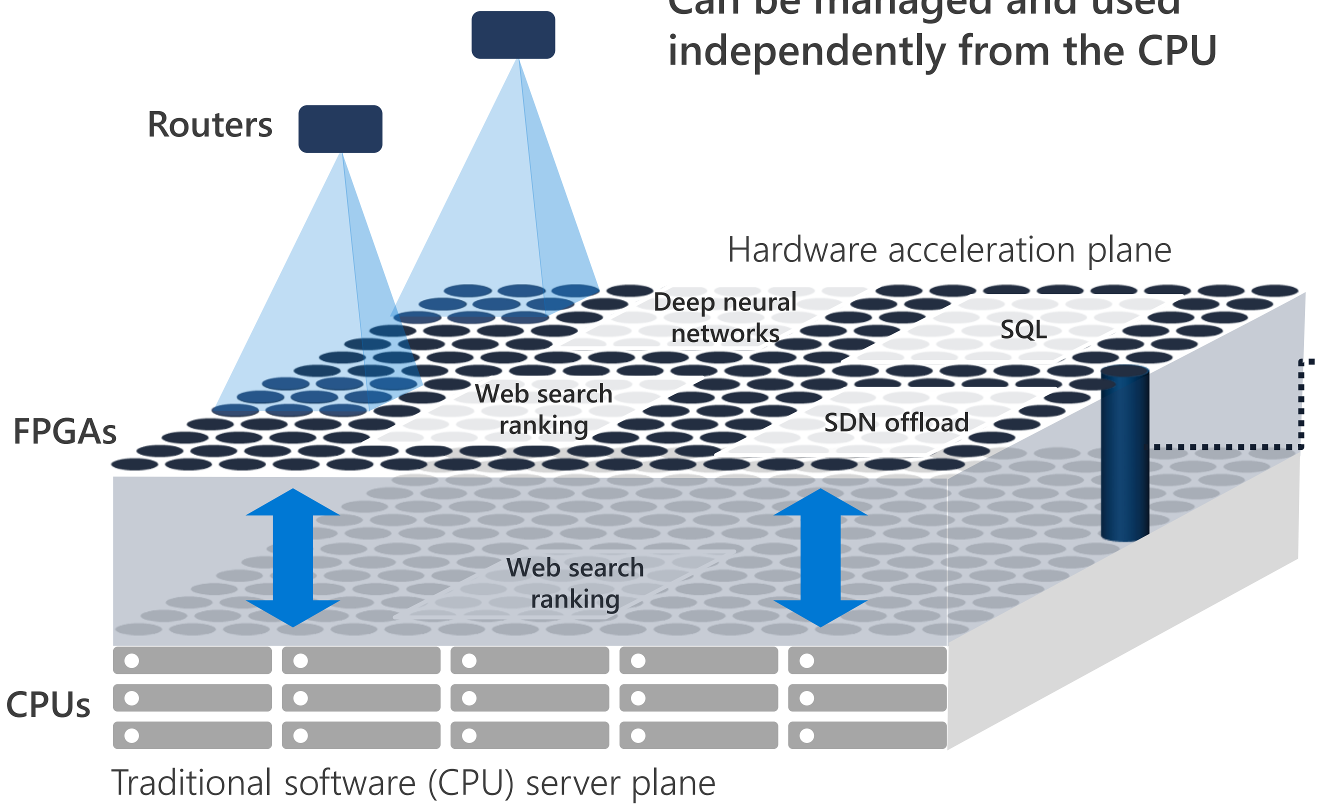
Brainwave runs on a configurable cloud at massive scale

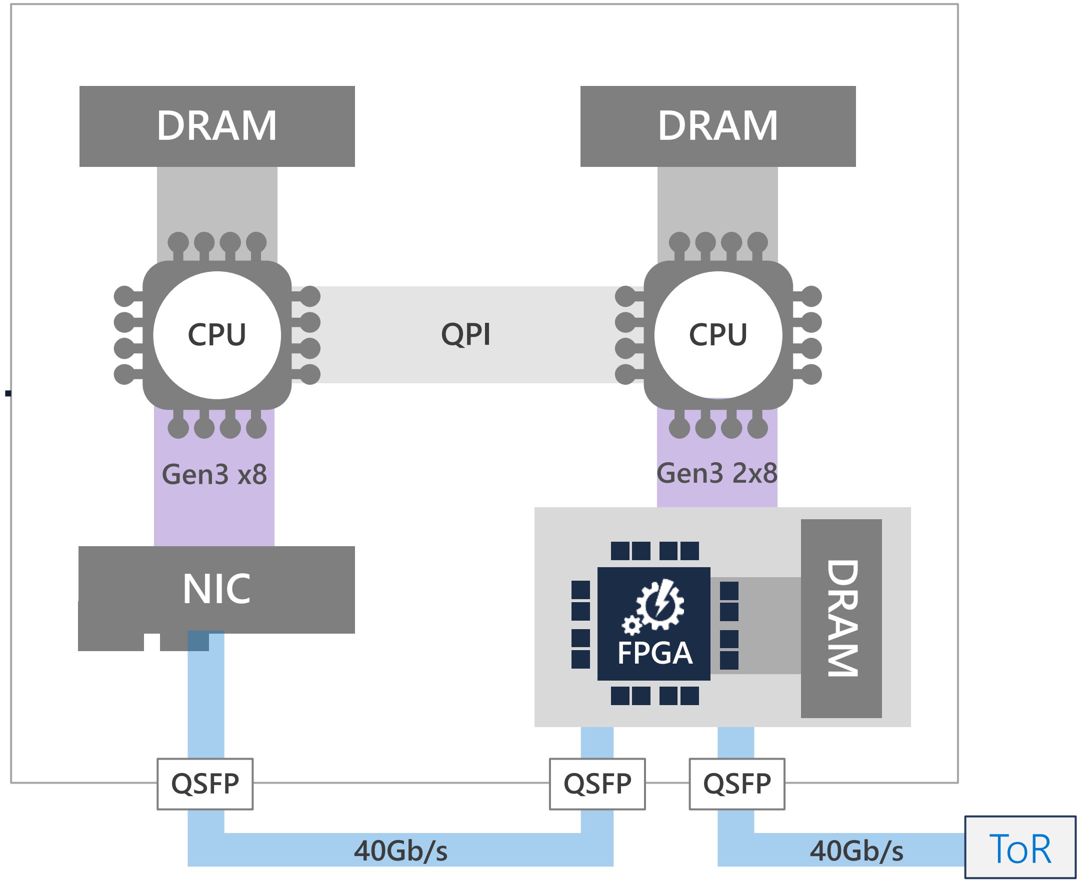


Scalable hardware microservice

Interconnected FPGAs form a separate plane of computation

Can be managed and used independently from the CPU





Why FPGAs for AI/DL?

Balance: Performance and flexibility

Scale: Multiple Exa-Ops of aggregate AI capacity

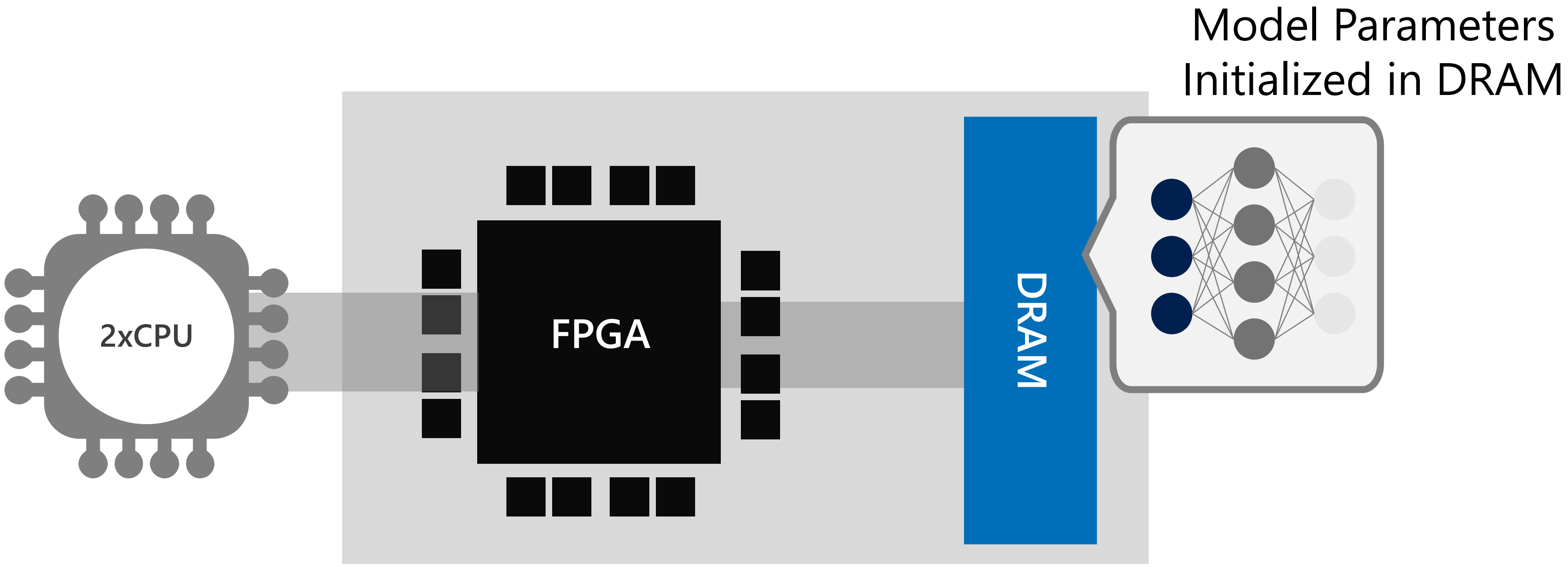
Optimize: Synthesize variants of the DNN engine based on individual model requirements

Adaptability: Ability to add customized datatypes, sparsity, etc.

Future proof: Ability to pivot as fundamental shifts in models happen

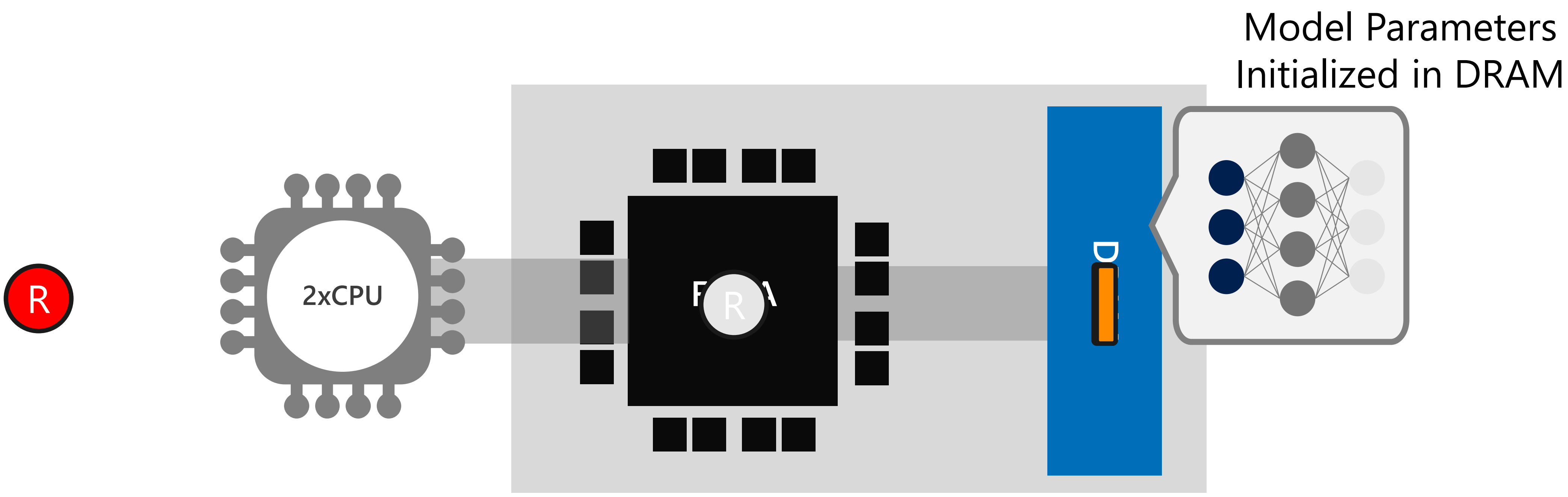
Conventional acceleration approach

Local offload and streaming



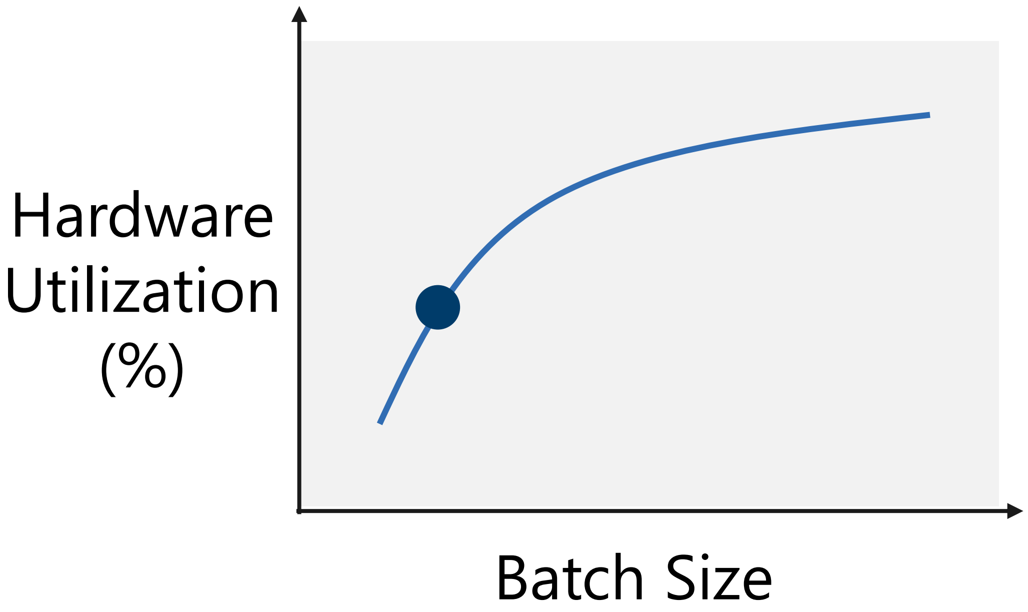
Conventional acceleration approach

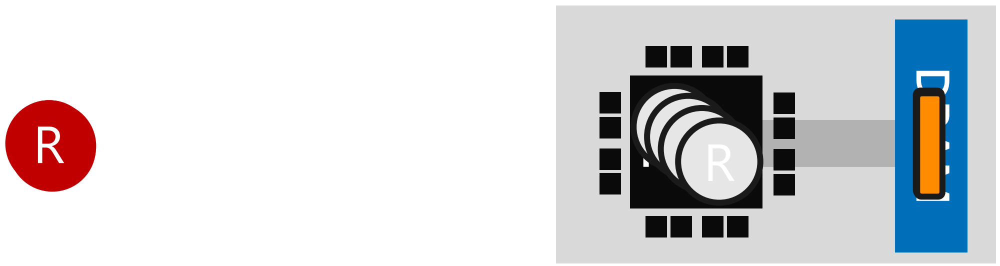
Local offload and streaming



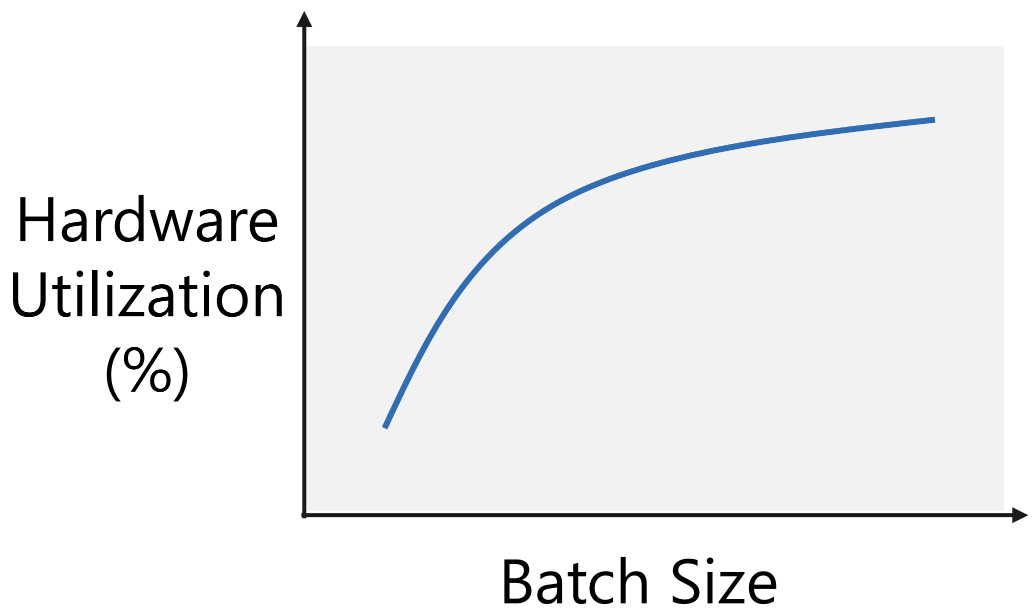
For memory-intensive DNNs with low compute-to-data ratios (e.g., LSTM), HW utilization limited by off-chip DRAM bandwidth

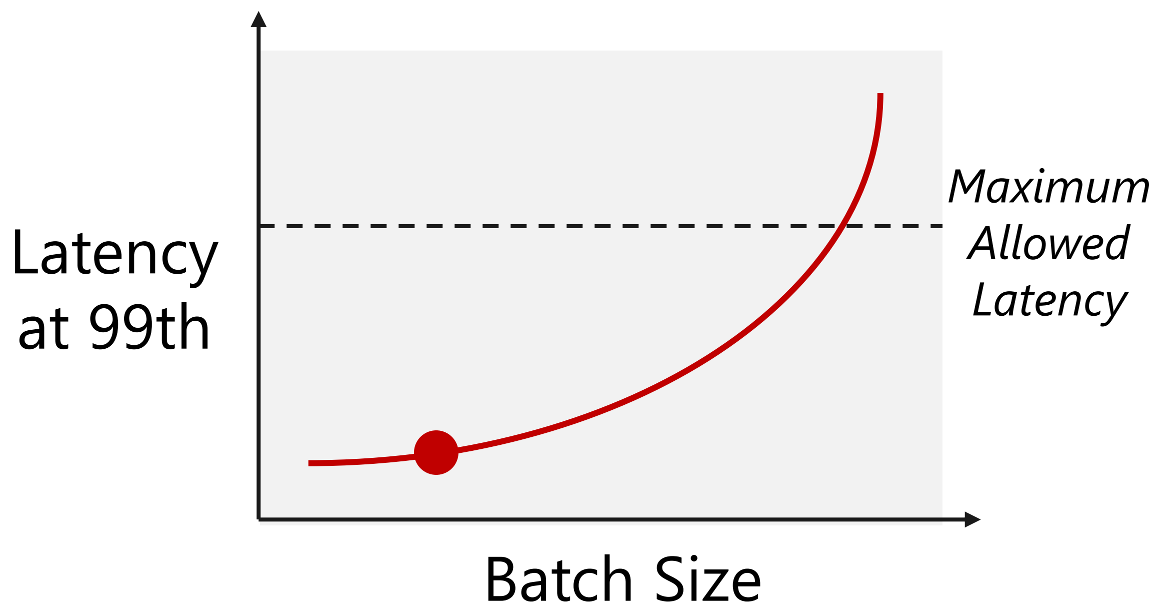
Improving HW utilization with batching





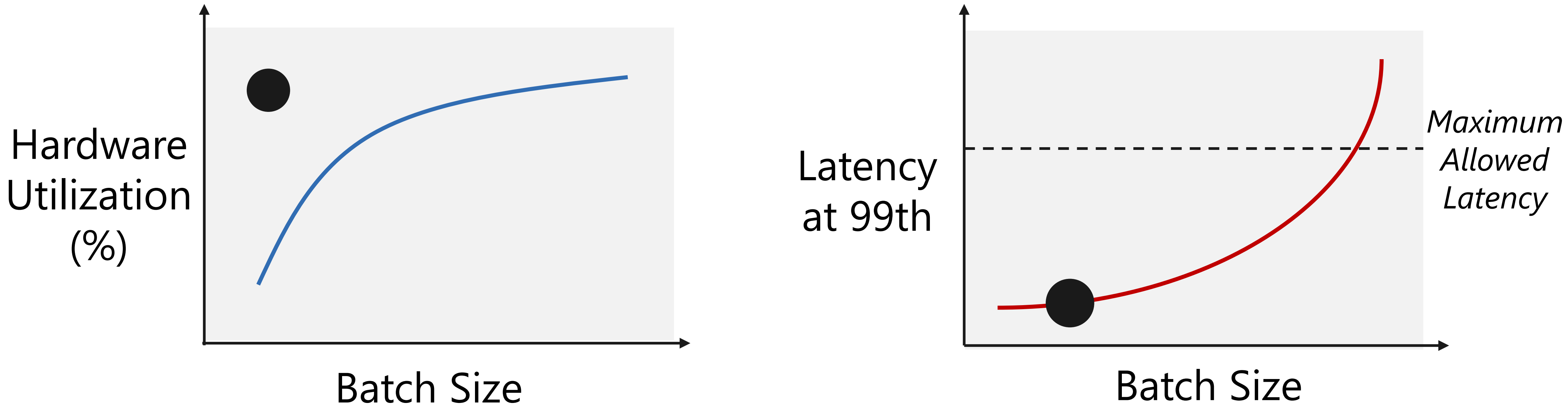
Improving HW utilization with batching





Batching improves HW utilization but also increases latency

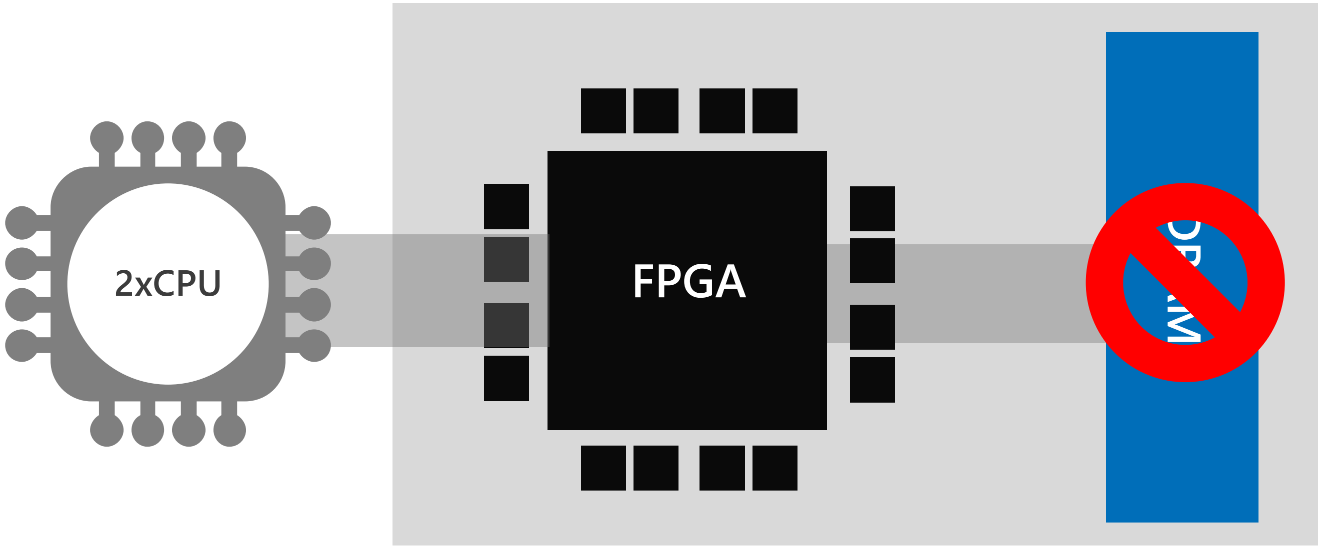
Improving HW utilization with batching



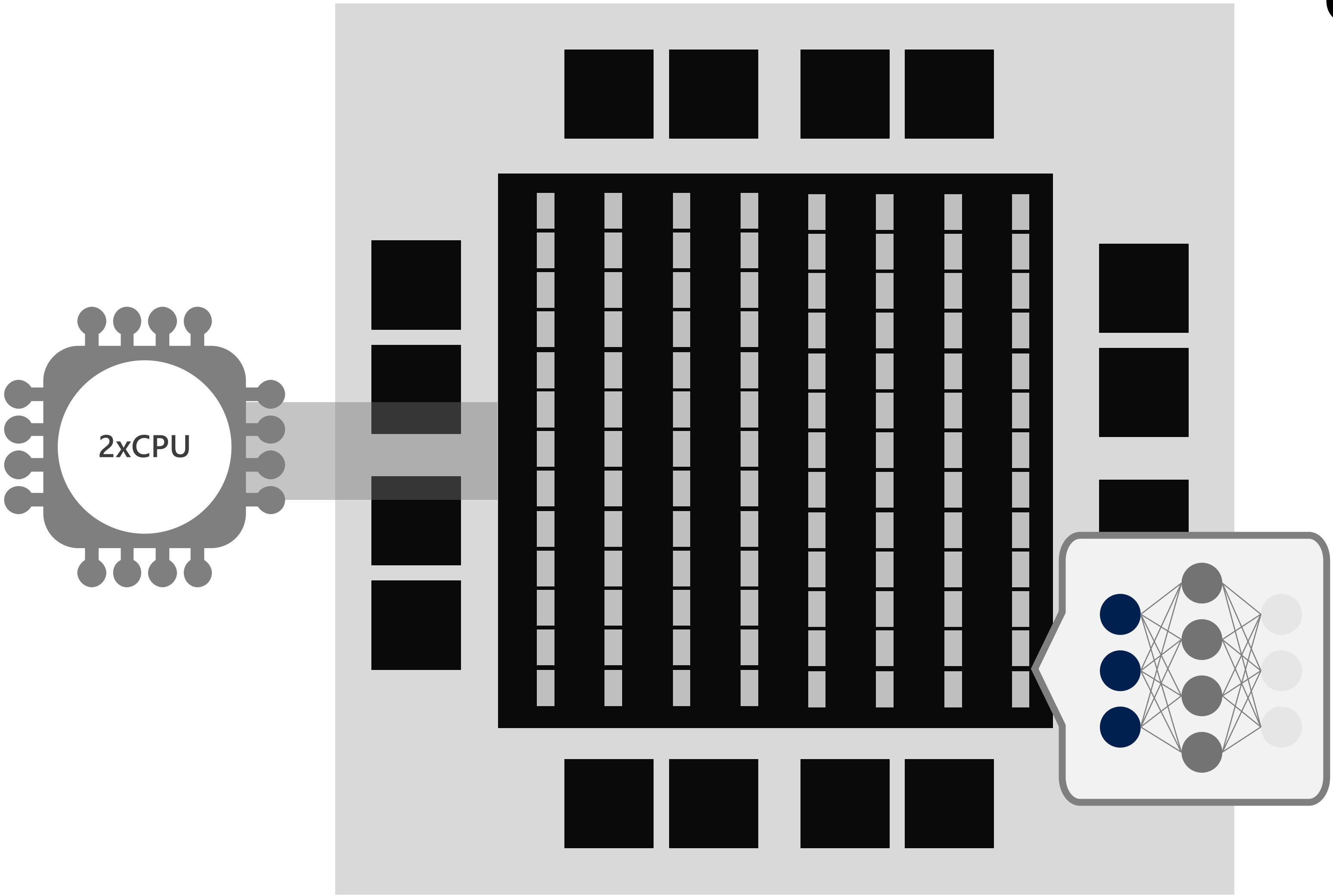
Batching improves HW utilization but increases latency

Ideally want high HW utilization at low batch sizes

Alternative: “persistent” neural nets



Alternative: "persistent" neural nets



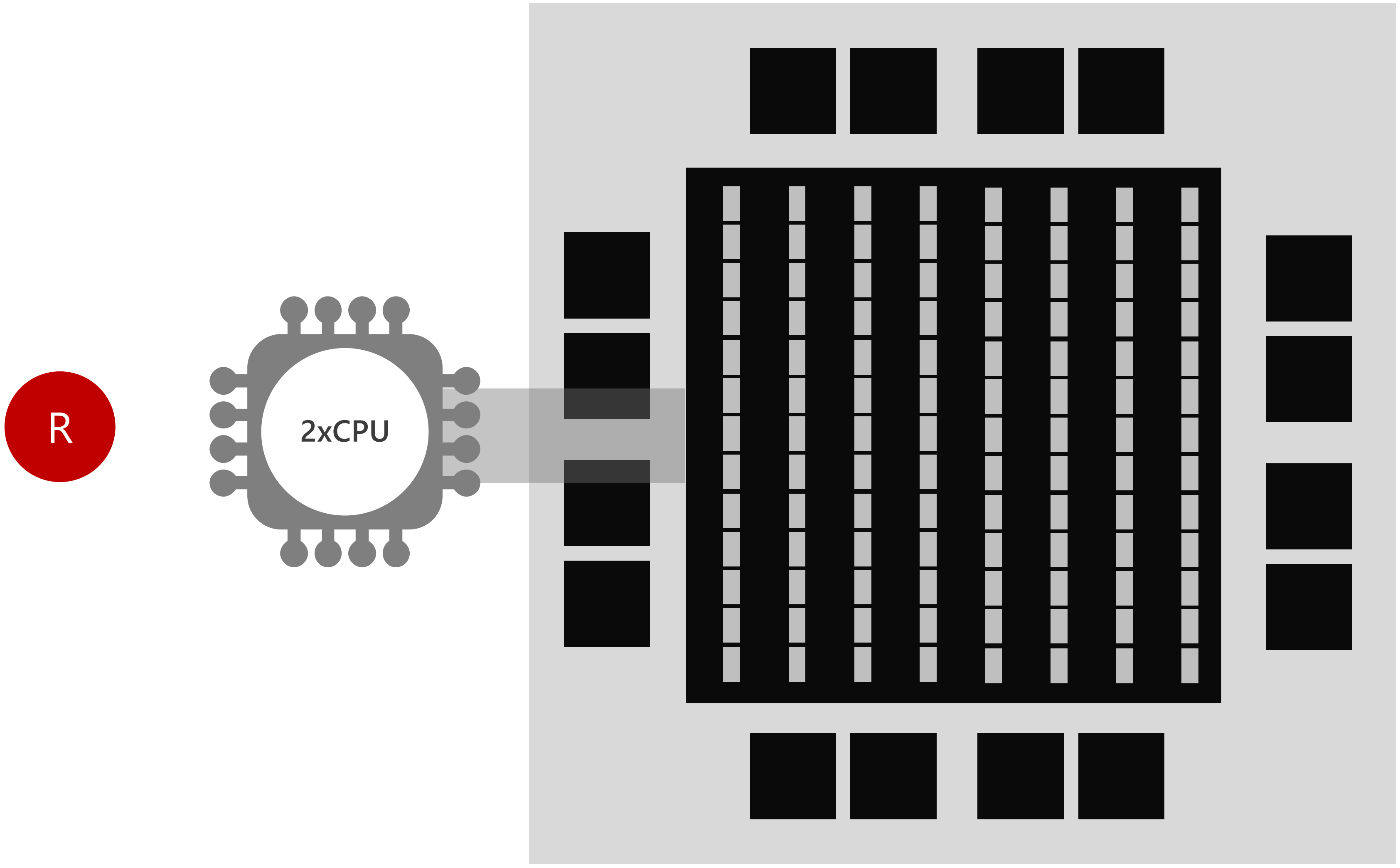
Observations

State-of-art FPGAs have O(10K) distributed Block RAMs O(10MB)
➔ Tens of TB/sec of memory BW

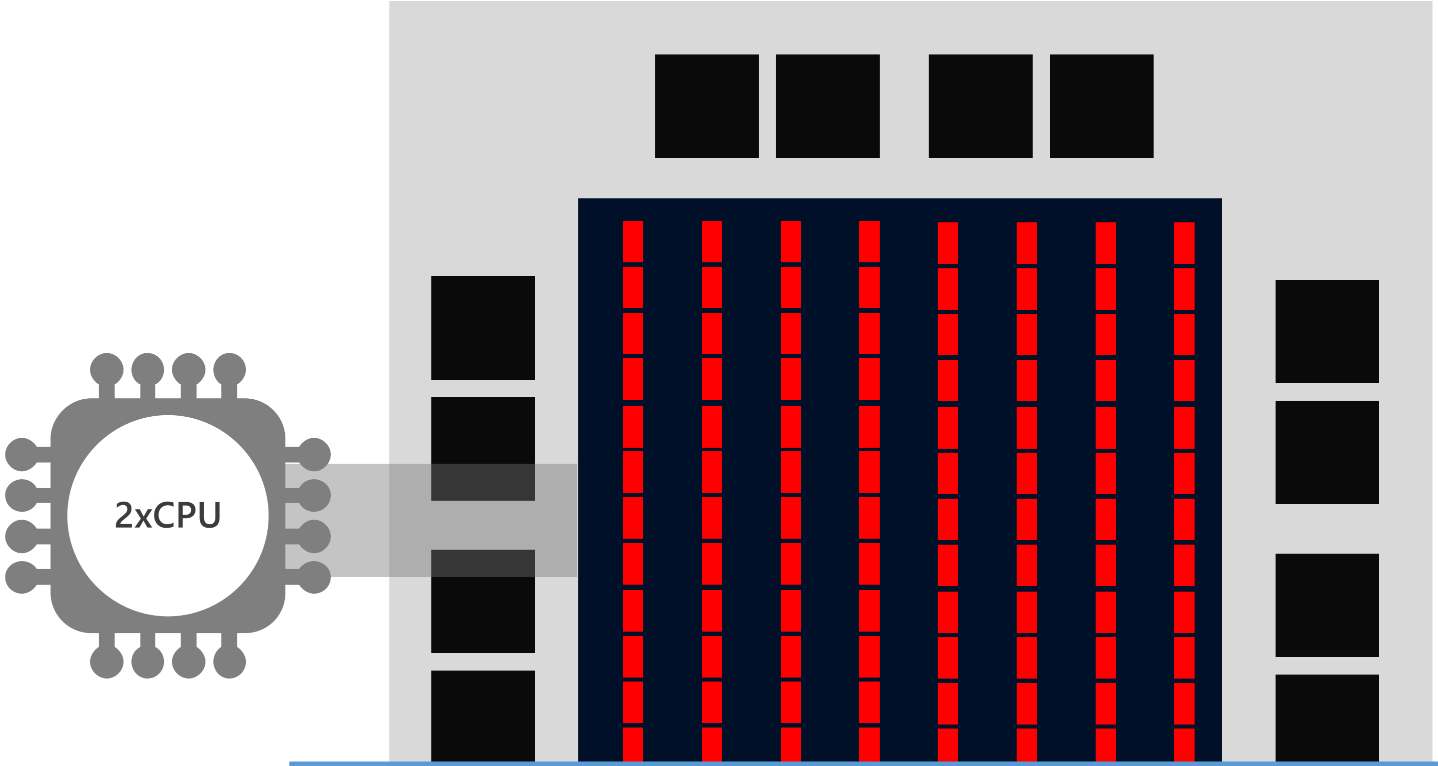
Large-scale cloud services and DNN models run persistently

Solution: persist all model parameters in FPGA on-chip memory during service lifetime

Alternative: “persistent” neural nets

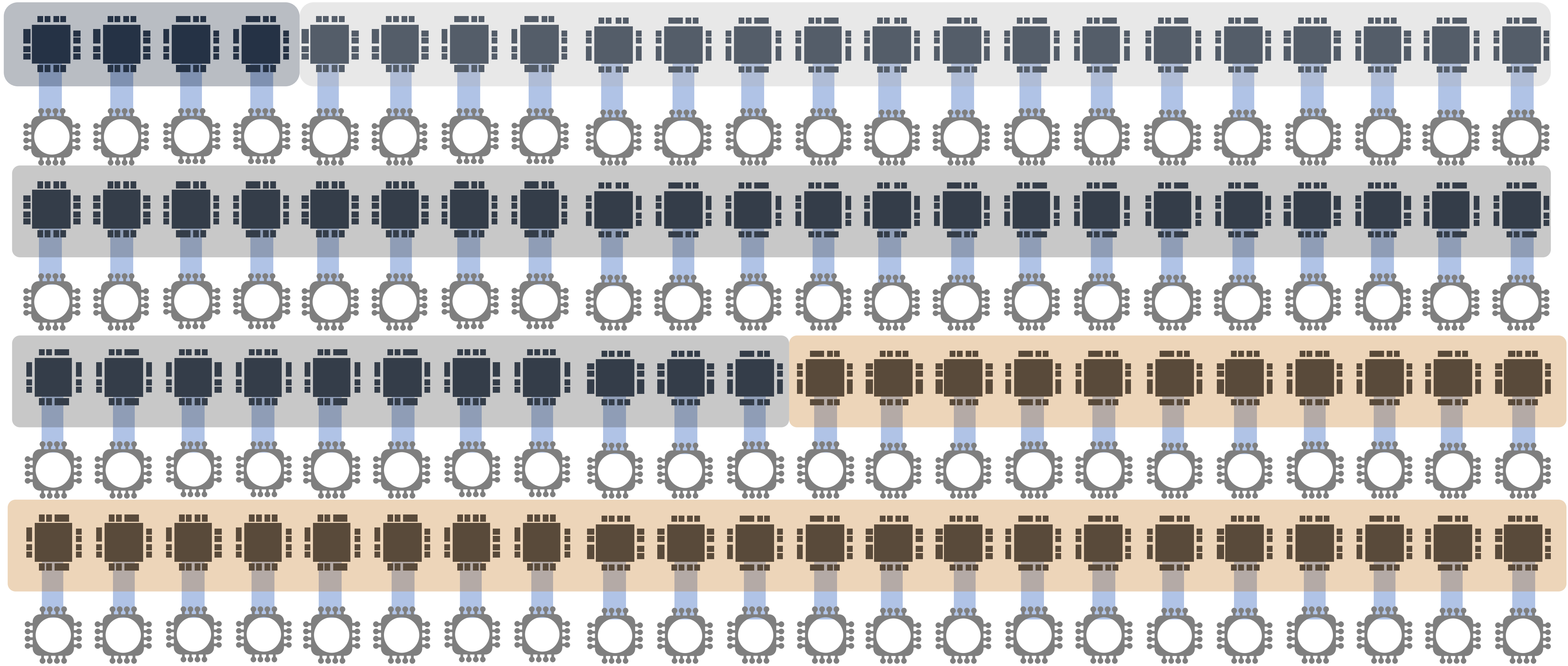


Alternative: “persistent” neural nets



When single request arrives, all chip resources (on-chip memories and compute units) are used to process a single query (no batching required)

Persistency at datacenter scale



Multiple FPGAs at datacenter scale can form a persistent DNN HW microservice, enabling scale-out of models at ultra-low latencies

Why FPGAs for AI/DL?

Balance: Performance and flexibility

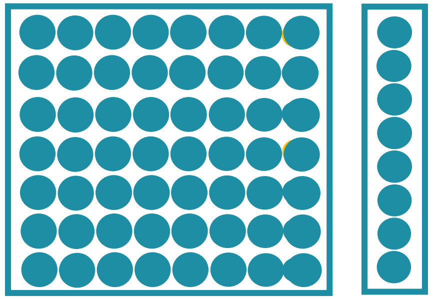
Scale: Multiple Exa-Ops of aggregate AI capacity

Optimize: Synthesize variants of the DNN engine based on individual model requirements

Adaptability: Ability to add customized datatypes, sparsity, etc.

Future proof: Ability to pivot as fundamental shifts in models happen

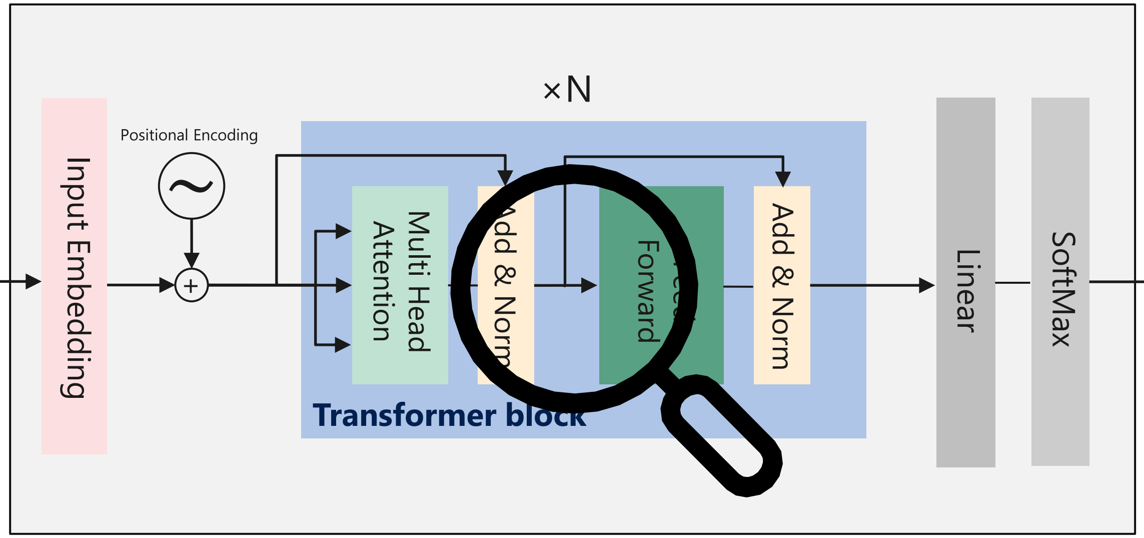
Matrix multiplication is a key part of current DL models



Dense Matrix Multiply
$y = Wx$

Example input

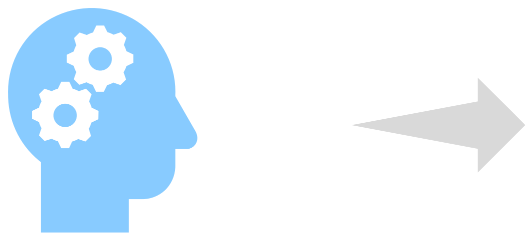
Task description	`Translate English to French:`
Context (e.g., examples)	`sea otter => loutre de mer`
	`peppermint => menthe poivrée`
	`plush girafe => girafe peluche`
Prompt	`cheese =>`

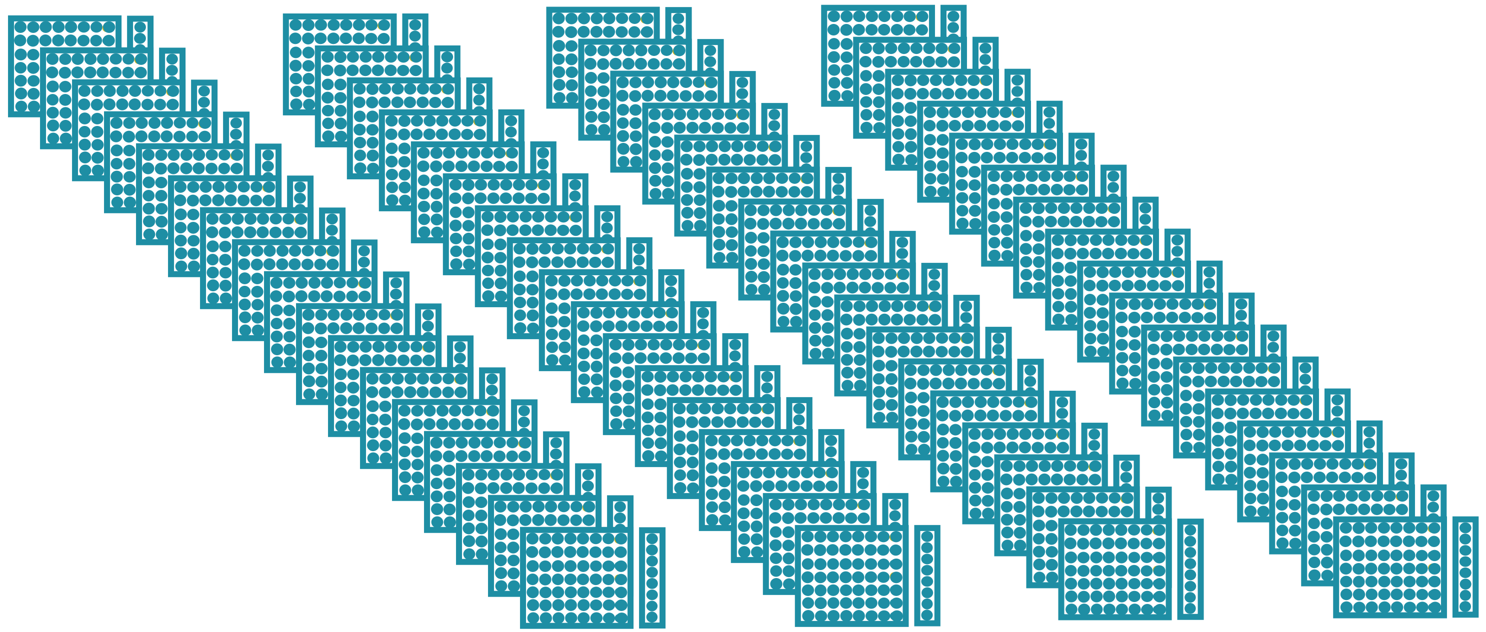


Output probability

Matrix multiplication is a key part of current DL models

Model

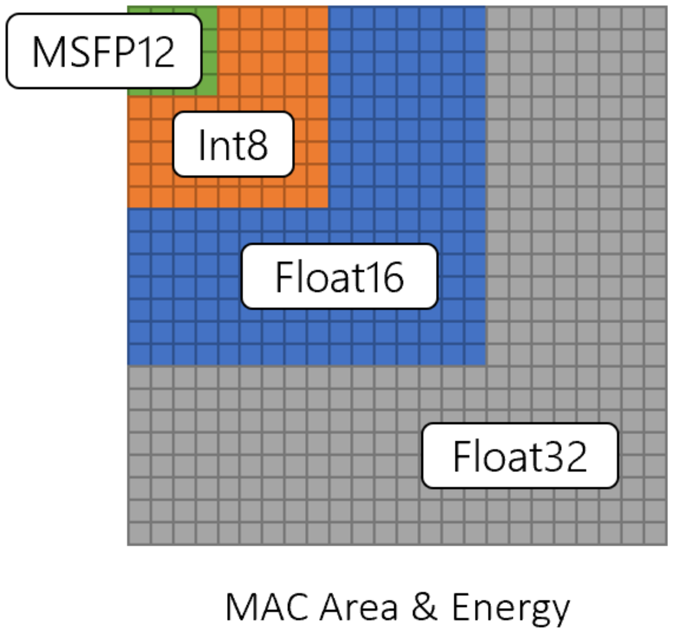


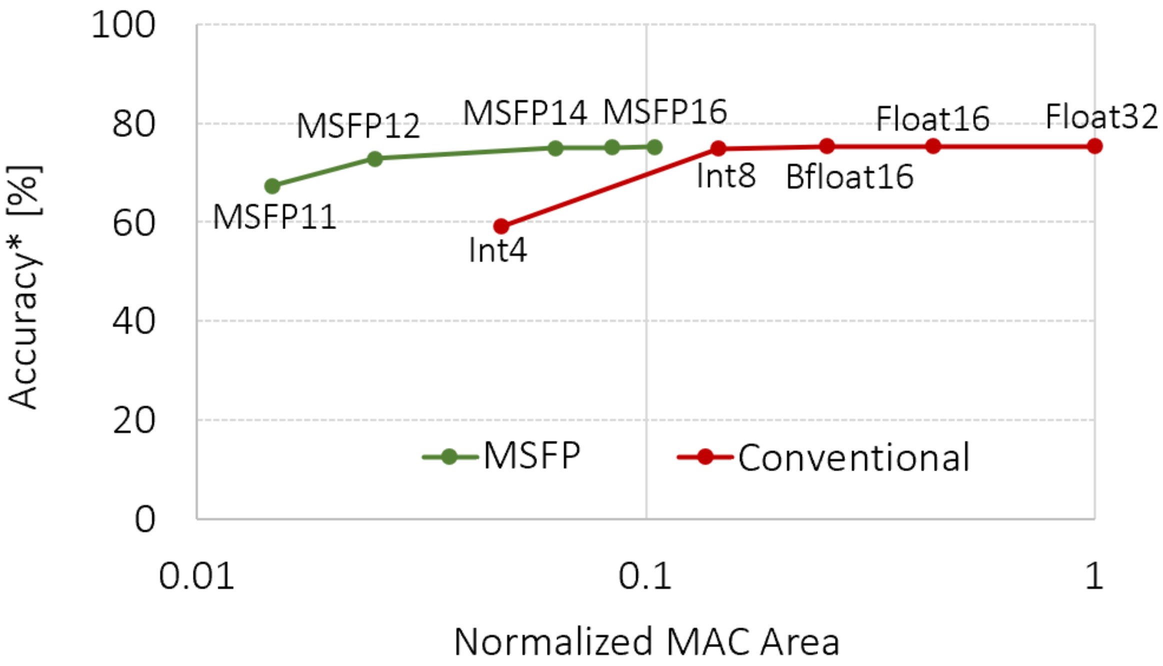


Millions to Billions of operations per sample

Datatype plays a key role in cost of matrix multiplication

Variants of MSFP together form a new Pareto frontier for computational performance compared to a collection of industry standard datatypes such as Bfloat16 and INT8.

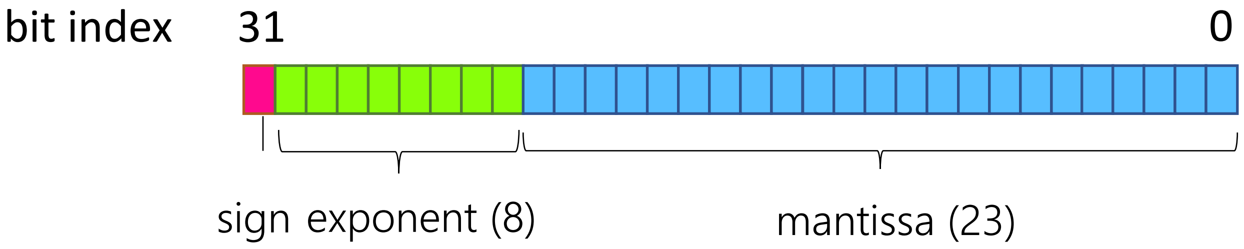




*ImageNet classification using ResNet50

IEEE floating-point datatype

Floating-point encodes values using **sign**, **exponent**, and **mantissa**

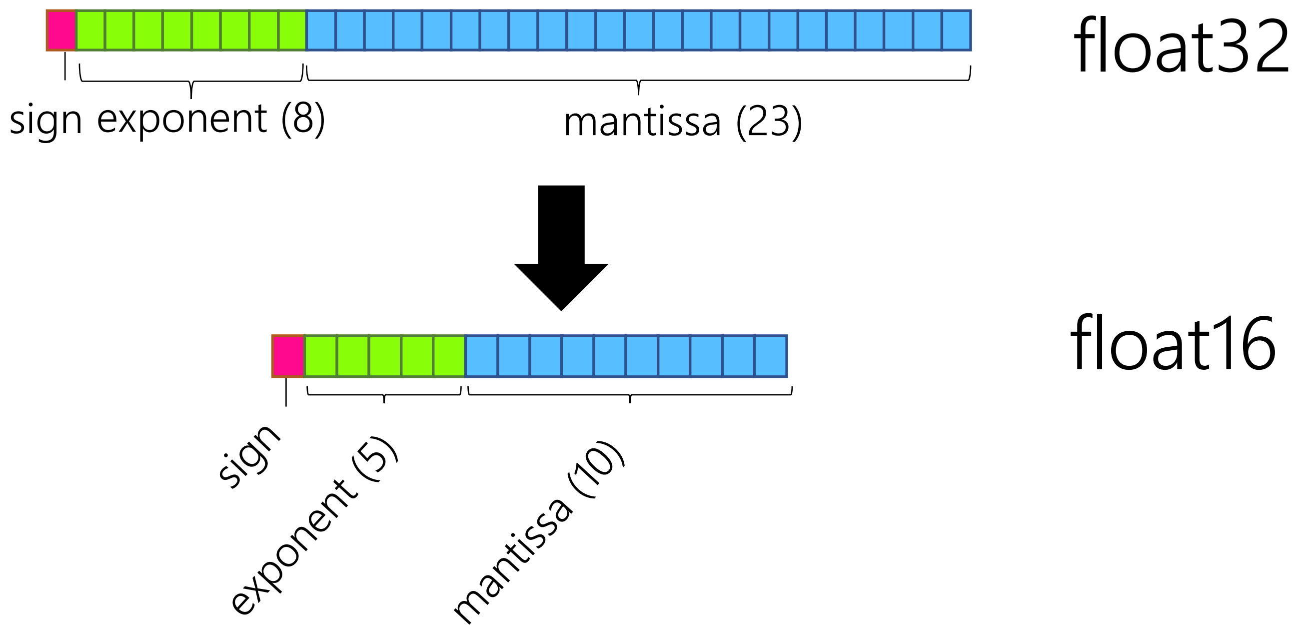


float32

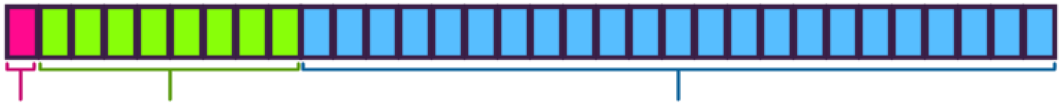
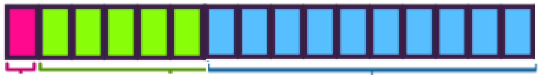
$$value = (-1)^{sign} \times 2^{e-127} \times (1 + \sum_{i=1}^{23} b_{23-i} 2^{-i})$$

IEEE floating-point datatype

Traditional reduced precision data type

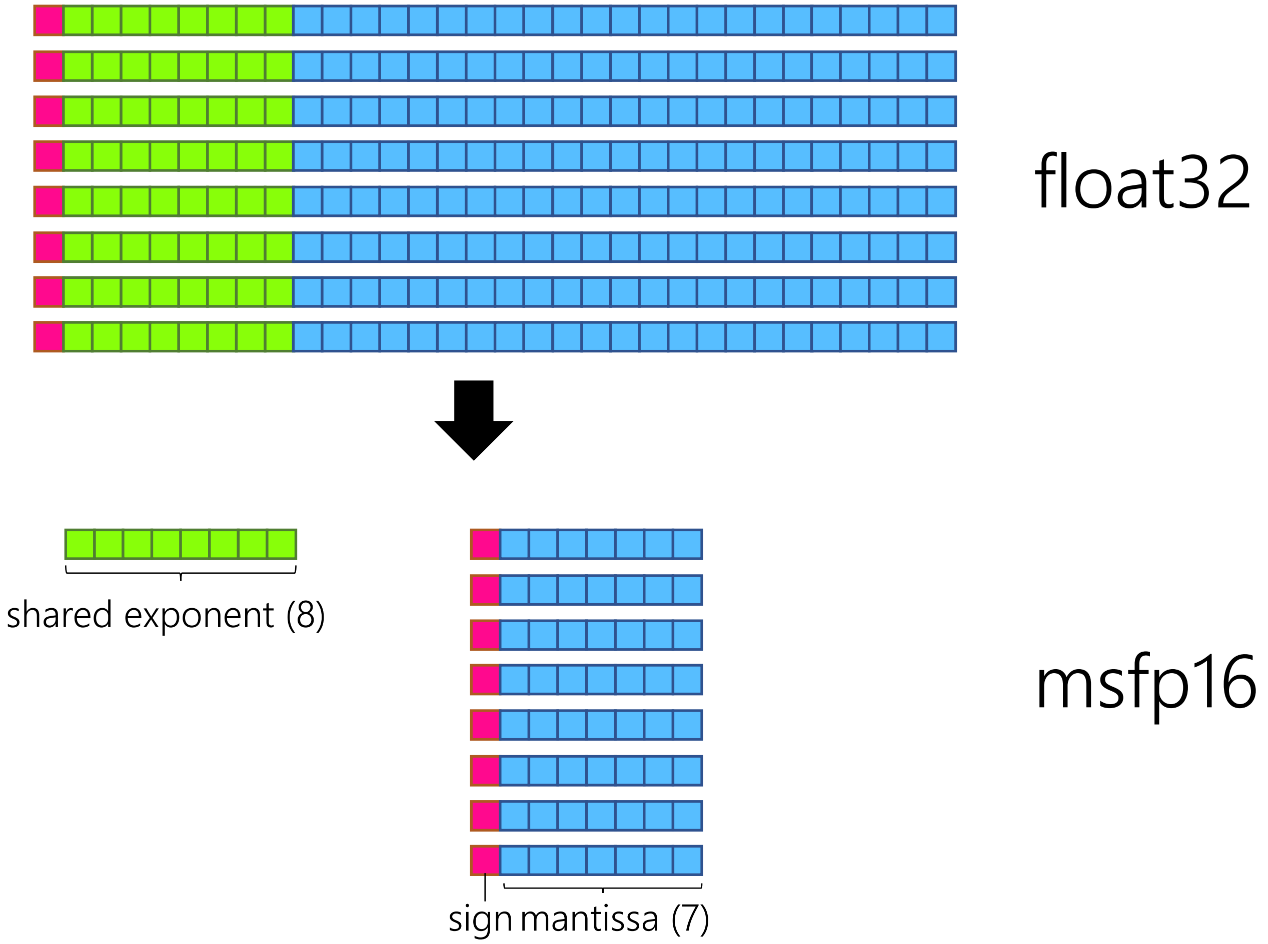


IEEE floating-point datatype

		Memory Density	Multiplier Density
IEEE Float-32	 Sign Exponent [8 bits] Mantissa [23 bits]	**1X**	**1X**
IEEE Float-16	 Mantissas [10 bits] Sign Exponent [5 bits]	**2X**	**2X**

MSFP: custom datatype for DL

Represent a vector of N numbers using **1 shared exponent** and **N low-precision mantissas**



Conversion to MSFP

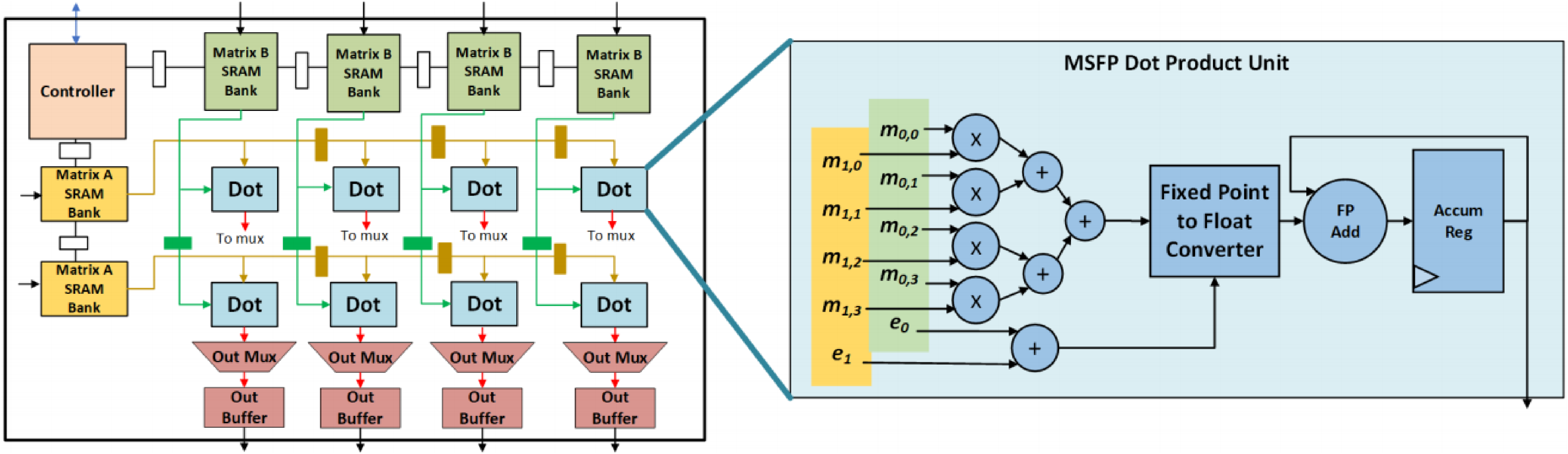
Shared Exp.

Determine Shared Exponent	Shift to shared exponent
$2^6 \times 1.01011$	$2^6 \times 1.01011$
$2^5 \times 1.00110$	$2^6 \times 0.10011$
$2^5 \times 0.10111$	$2^6 \times 0.01011$
$2^4 \times 1.00011$	$2^6 \times 0.01000$
$2^4 \times 1.00010$	$2^6 \times 0.01000$
$2^3 \times 0.11001$	$2^6 \times 0.00011$

Shared exponent = exponent of largest element

* We refer to the span of a shared exponent as the bounding box size
**shared exponent can be selected based on other metrics such as standard deviation of elements

Computing with MSFP datatype

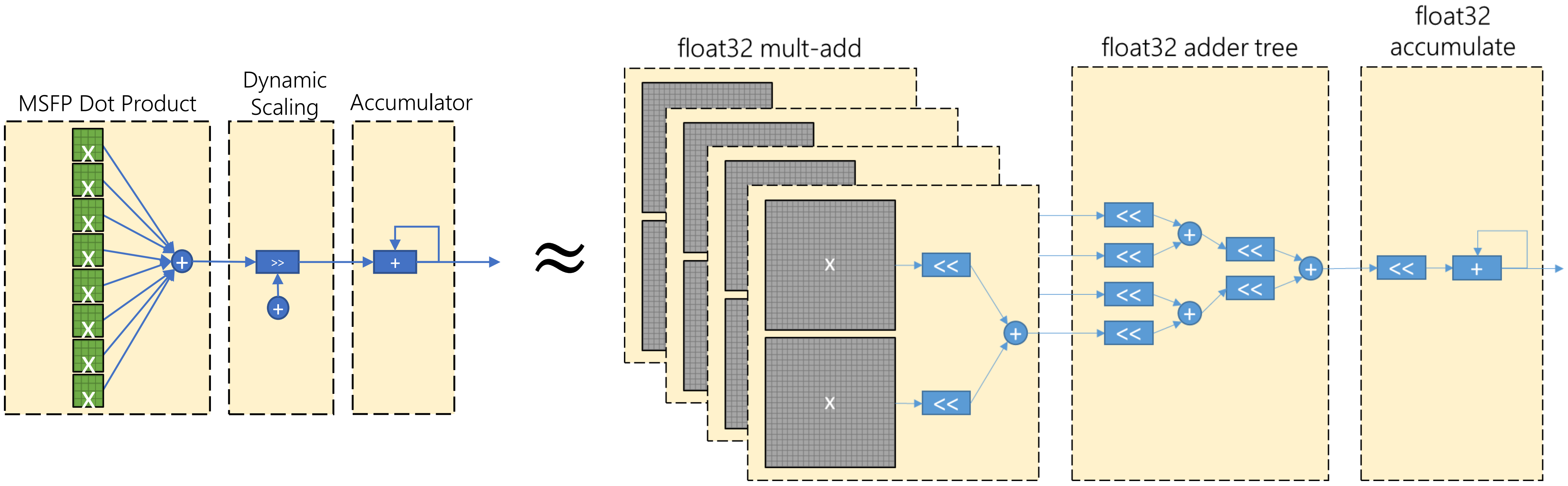


$$
\begin{aligned}
\vec{x_0}.\vec{x_1}^T &= 2^{e_0}\left[(-1)^{s_{0,0}}\ m_{0,0}\ ,\ (-1)^{s_{0,1}}\ m_{0,1},\ \ldots\ ,(-1)^{s_{0,n-1}}\ m_{0,n-1}\right].\\
&\quad 2^{e_1}\left[(-1)^{s_{1,0}}\ m_{1,0}\ ,\ (-1)^{s_{1,1}}\ m'_{1,1},\ \ldots\ ,(-1)^{s_{1,n-1}}\ m_{1,n-1}\right]^T\\
&= 2^{e_0+e_1}\sum_{i=0}^{n-1}\left((-1)^{s_{0,i}\oplus s_{1,i}}\ m_{0,i} * m_{1,i}\right),
\end{aligned}
$$

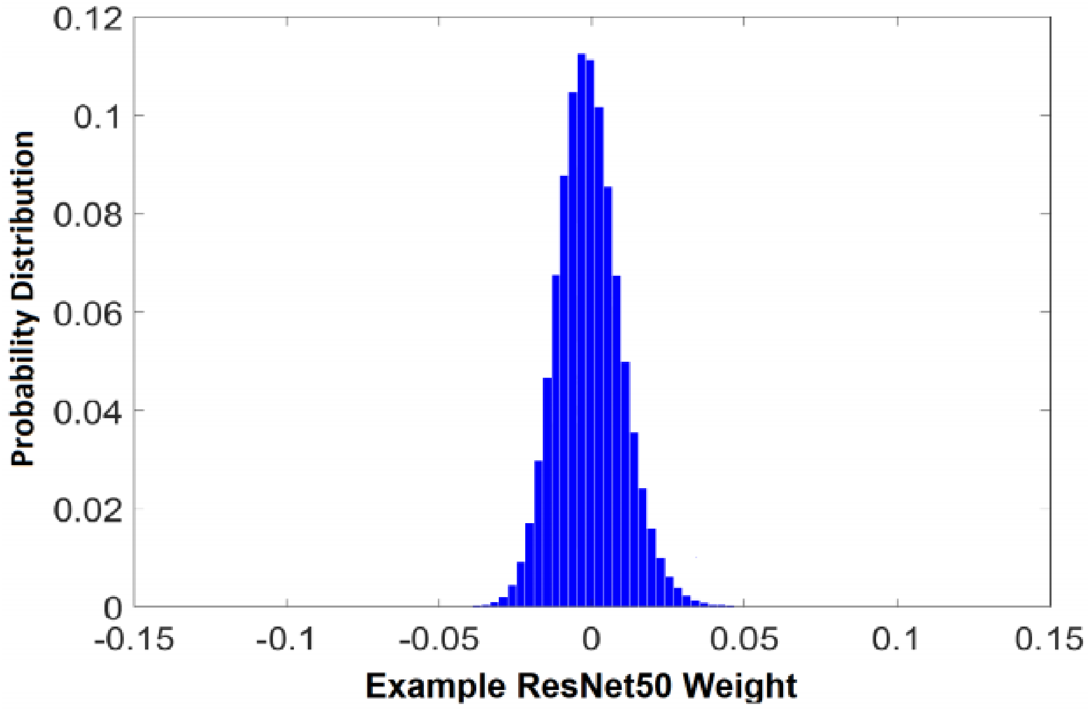
Computing with MSFP datatype

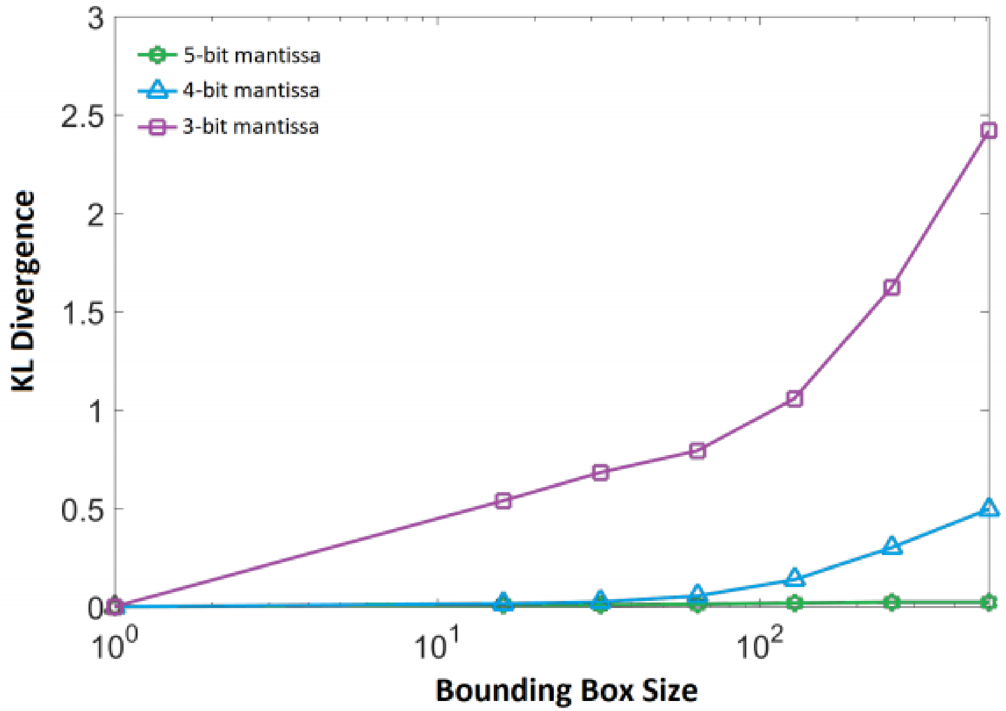
Dynamic scaling hardware is small compared to multiplier area reduction

The input vectors within a single mat-mul operations that reduce to a single accumulated output are assigned a shared exponent in contiguous static-sized bounding box



Trade-off between graduality of shared exponent and mantissa bits





(Bounding box size is the span of a shared exponent)

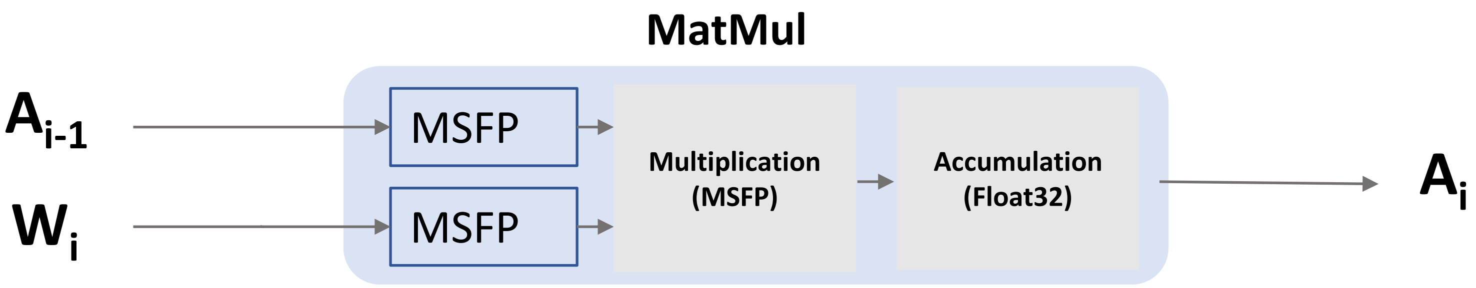
MSFP: Efficient custom data format for DL

		Memory Density	Multiplier Density
IEEE Float-32	Sign, Exponent [8 bits], Mantissa [23 bits]	**1X**	**1X**
IEEE Float-16	Sign Exponent [5 bits], Mantissa [10 bits]	**2X**	**2X**
Bfloat16	Sign, Exponent [8 bits], Mantissa [7 bits]	**2X**	**3X**
MSFP16	Shared Exponent [8 bits], Signs and Mantissa [8 bits]	**4X**	**9X**

MSFP mat-mul configuration

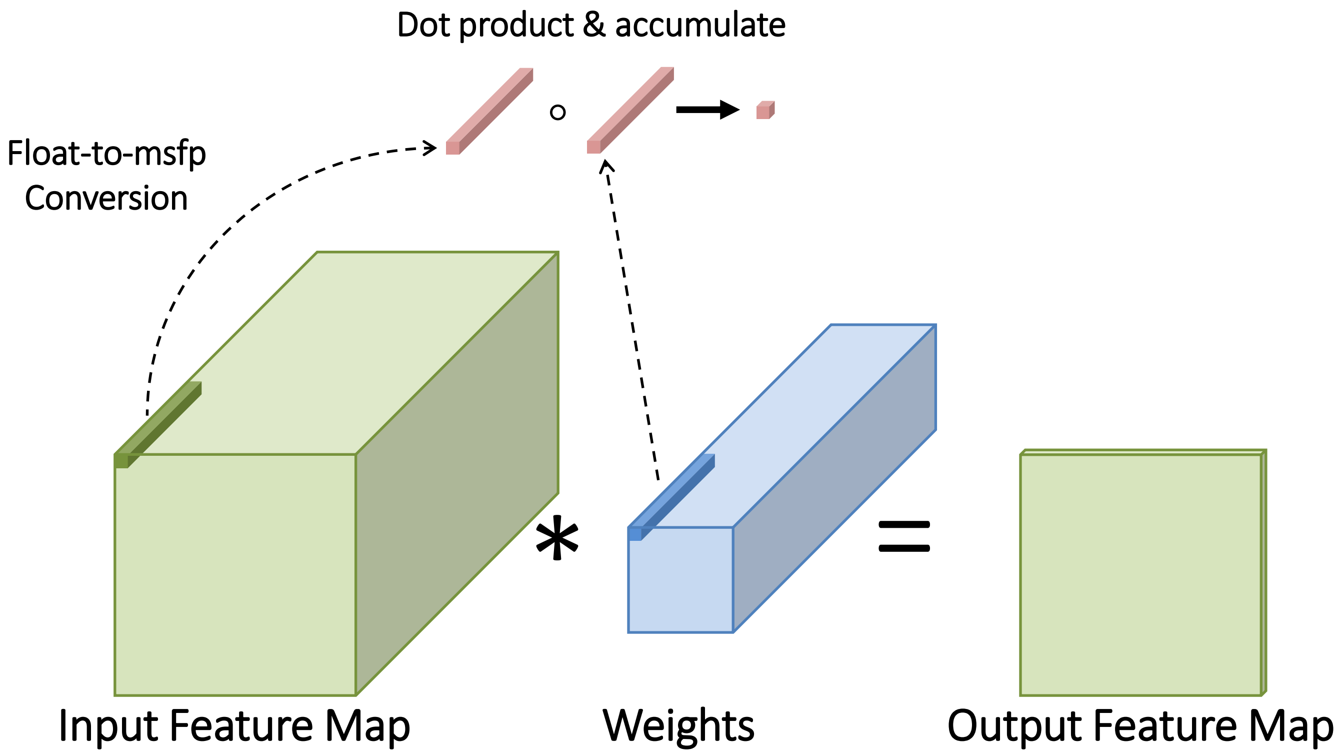
All conversions being handled directly in hardware through special instructions

The dimension of shared exponent is dictated by the inner dimension in the mat-mul



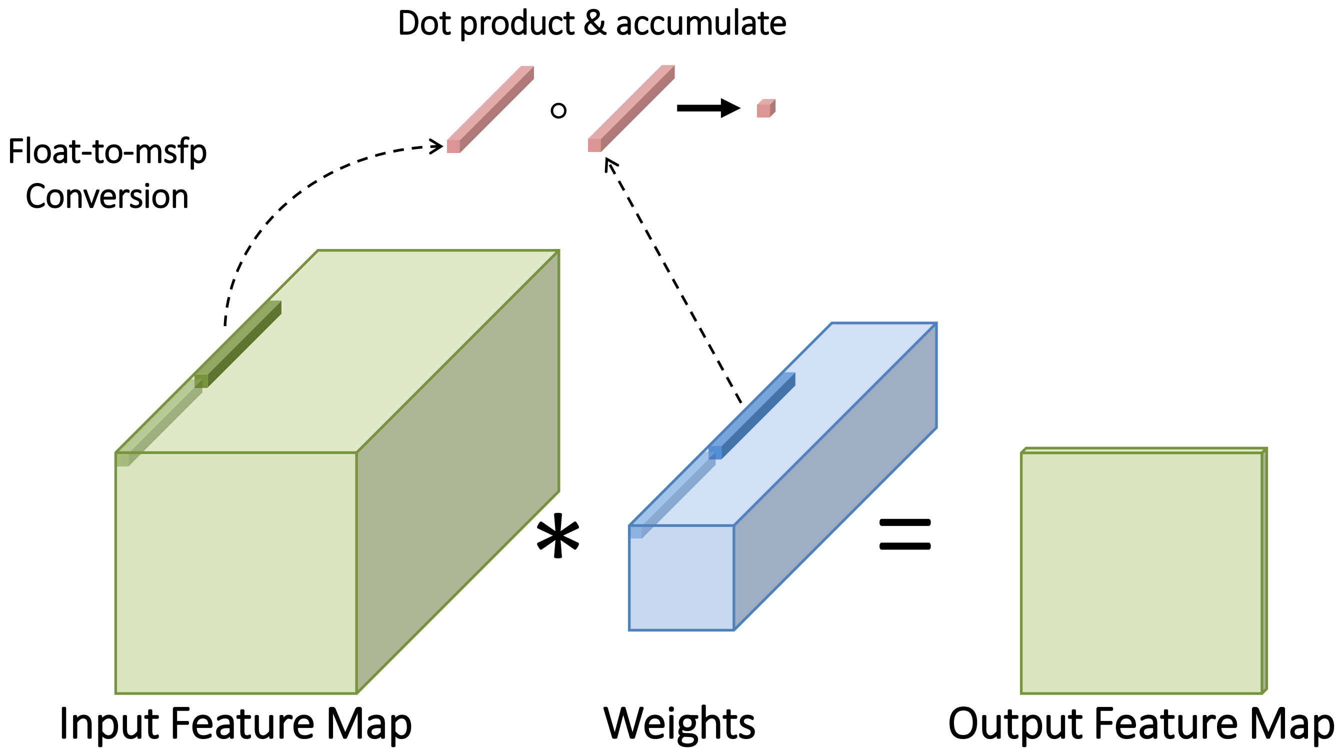
Example of using MSFP in a convolution layer

All conversions being handled directly in hardware through special instructions



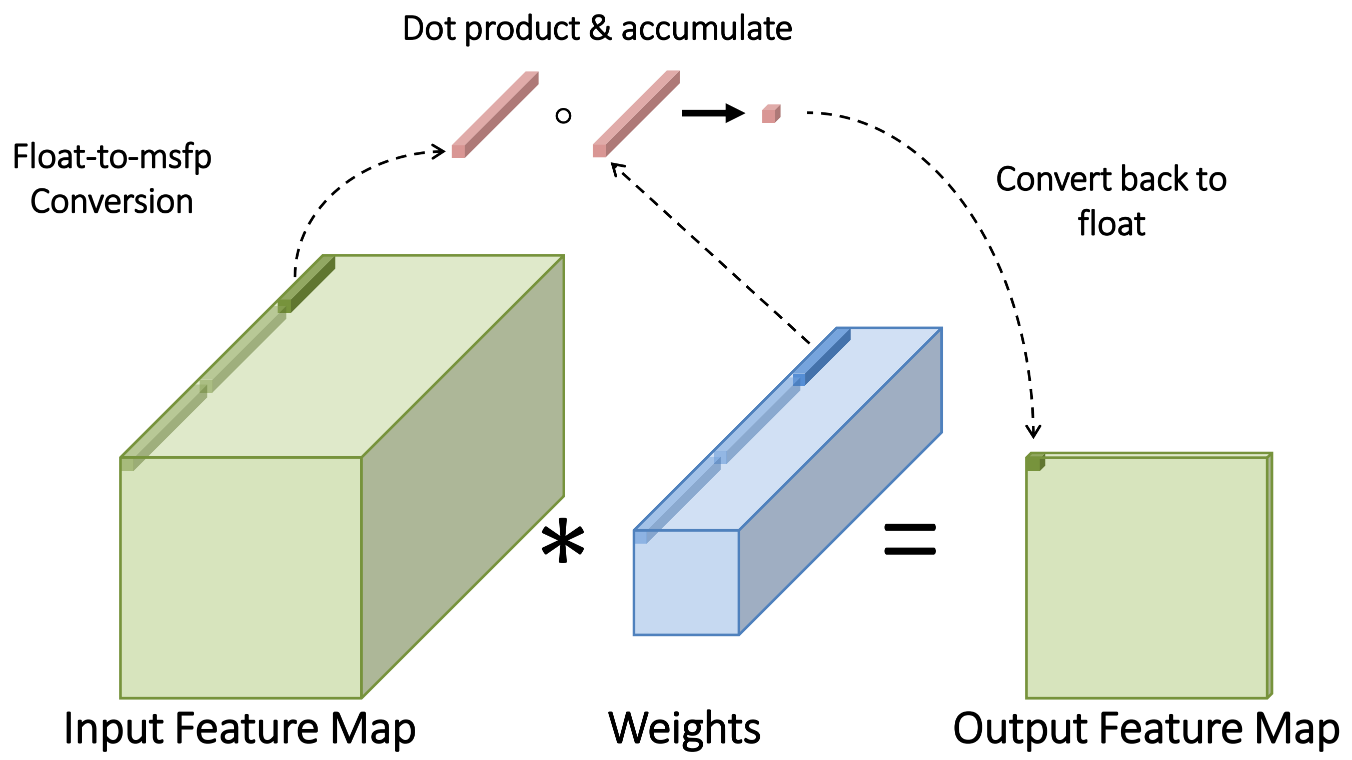
Example of using MSFP in a convolution layer

All conversions being handled directly in hardware through special instructions

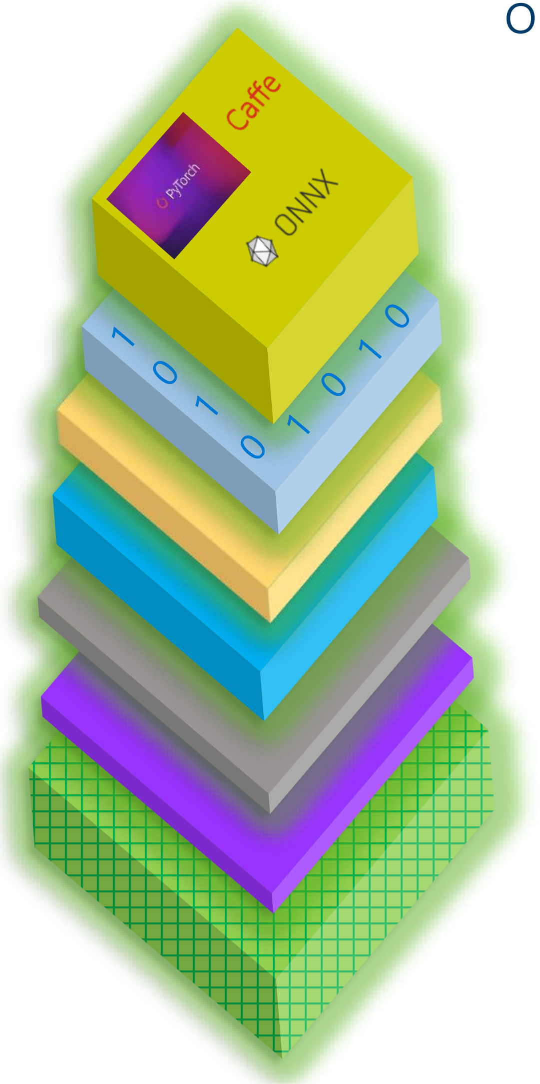


Example of using MSFP in a convolution layer

All conversions being handled directly in hardware through special instructions



End-to-end HW + SW integrated stack at cloud-scale



Opensource DNN Frameworks

↓ MSFP Quantization Libs

Quantized DNN Models

↓ Compiler

Model Firmware

↓ Runtime

Brainwave Microarchitecture

↓ Runtime Exception Handling

Microsoft Shell

↓ HW Exception Handling

FPGA boards & chassis

↓

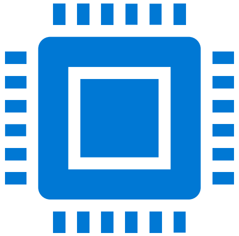
Model Deployment

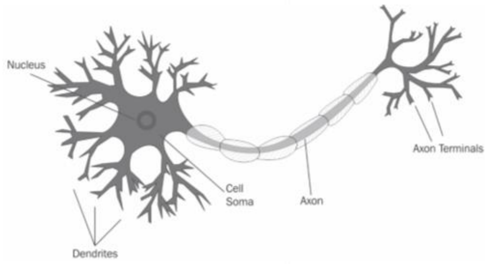
Generalizability of MSFP datatype

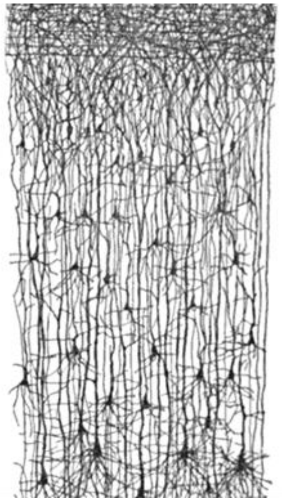
Models	Float32	MSFP16	MSFP15	MSFP14	MSFP13	MSFP12
Resnet-50	1.000 (75.26)	1.000	0.999	**0.994**	0.989	0.967
Resnet-101	1.000 (76.21)	1.000	1.000	0.998	**0.991**	0.964
Resnet-152	1.000 (76.58)	1.000	1.001	0.997	**0.991**	0.968
Inception-v3	1.000 (77.98)	1.000	1.005	1.001	**0.990**	0.943
Inception-v4	1.000 (80.18)	1.000	1.001	1.000	**0.993**	0.963
MobileNet-V1	1.000 (70.90)	0.998	0.997	**0.990**	0.965	0.863
VGG16	1.000 (70.93)	1.000	1.004	1.005	1.003	**1.002**
VGG19	1.000 (71.02)	1.000	1.002	1.001	1.002	**1.000**
EfficientNet-S	1.000 (77.61)	1.000	0.998	**0.992**	0.979	0.949
EfficientNet-M	1.000 (78.98)	1.000	0.998	**0.993**	0.980	0.950
EfficientNet-L	1.000 (80.47)	1.000	0.999	**0.993**	0.974	0.945
RNN-DR	1.000 (76.10)	1.000	1.008	1.003	1.009	**1.000**
RNN-DS	1.000 (73.10)	1.000	1.012	1.005	1.022	**0.992**
BERT-MRPC	1.000 (88.39)	1.000	1.005	1.002	1.008	**1.018**
BERT-SQuAD1.1	1.000 (88.45)	1.000	0.998	0.998	0.997	**0.990**
BERT-SQuADv2	1.000 (77.23)	1.000	0.999	0.999	**0.993**	0.989
Memory density	1.0x	3.8x	4.3x	4.9x	5.8x	7.1x
Arithmetic density	1.0x	8.8x	10.8x	13.9x	18.3x	31.9x

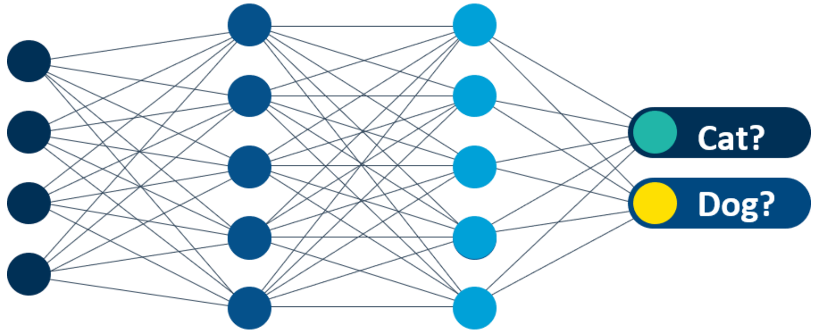
Closing Thoughts …

Biology vs. Deep Learning

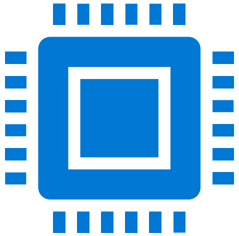






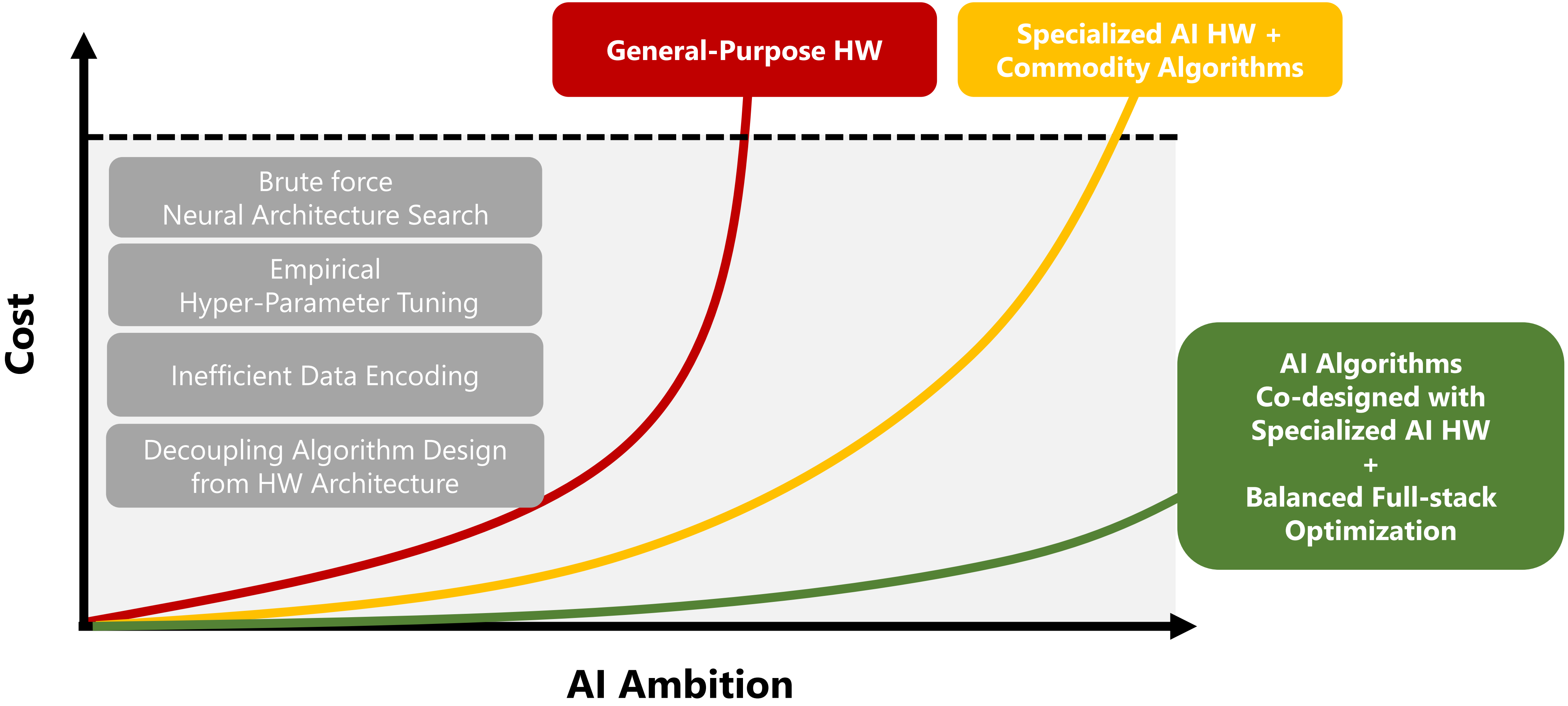


Biology vs. Deep Learning



Biology	Deep Learning
Low power ~25W	Up to tens of MWs at scale
Low precision ~ few bits	High precision (floating point)
Tens of hertz	Gigahertz clock speeds
Complex neuron model	Artificial (linear) neuron model
Unsupervised learning algorithm	Supervised (or semi-supervised) with stochastic gradient descent
Few (unlabeled) samples needed to train	Many labeled samples required to train
Sparsely connected, sparsely activated	Dense weights and dense activations
Sparsely computed in time domain	Densely computed with no timing
1 quadrillion (biological) weights @ ~25W	Less than 1 trillion weights @ ~30MW

Bending the AI ambition-cost curve



We are hiring

Sends Resumes To:

bita.rouhani@microsoft.com