Deep Learning Optimisé - Jean Zay

Bonnes pratiques et état de l'art

INSTITUT DU
DÉVELOPPEMENT ET DES
RESSOURCES EN
INFORMATIQUE
SCIENTIFIQUE

Où trouver les bonnes pratiques et l'état de l'art ?

Fast.ai ◂

MLPerf ◂

TIMM ◂

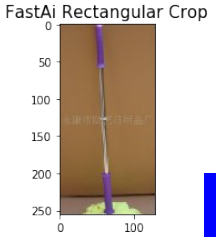
Hugging Face◂

Astuces et ingénierie Fast.ai

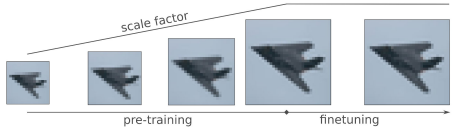
fast.ai

"An AI speed test shows clever coders can still beat tech giants like Google and Intel." DAWNBench competition 2018

👏 OneCycle lr scheduler + lr finder → Popularise les travaux de Leslie N. Smith



Grâce au Global Average Pooling



Test Rectangular Validation Technique

Progressive image resizing

Dynamic batch size

ML Perf - Référence pour le Supercomputing en IA

Référence pour l'industrie

- Hardware
- Framework de Deep Learning
- Techniques SOTA

ML Perf - Benchmarks

MLPerf

Training

Speech Recognition
RNN-T

NLP
BERT

Recommender
DLRM

Reinforcement Learning
MiniGo

Biomedical Image Segmentation
UNet-3D

Object Detection (Light weight)
SSD

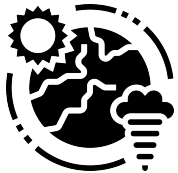
Object Detection (Heavy weight)
Mask R-CNN

Image Classification
ResNet-50 v1.5

ML Perf - Benchmarks

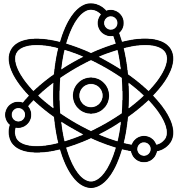
MLPerf

Training HPC



Climate segmentation
DeepCAM

Cosmology parameter prediction
CosmoFlow



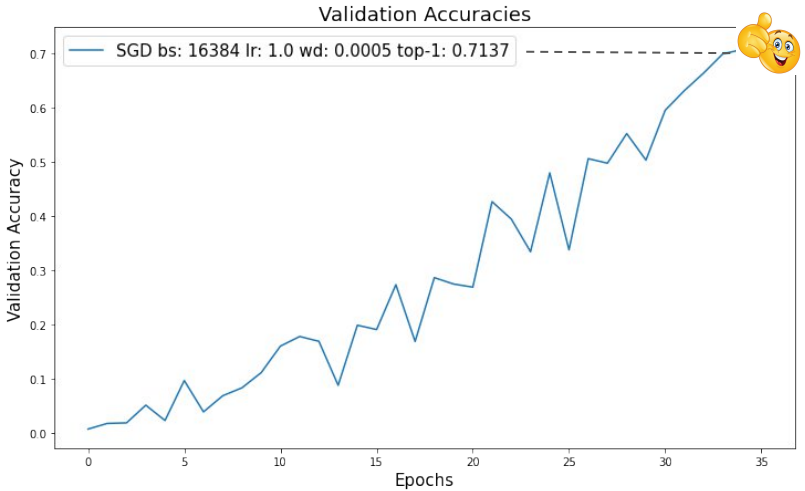
Quantum molecular modeling
DimNet++

Inference :

- Datacenter
- Edge
- Mobile
- Tiny

ML Perf - Benchmarks

Training rule



A metric threshold to reach

Time Results

MLPerf

ML Perf – sources pour connaître l'état de l'art

Menu

June 29, 2022 - Training

v2.0 Results

Other Rounds

Training v2.0

Closed | Open

	Submitter	System	Processor	#	Accelerator	#	Software	Benchmark results (minutes)								Details	Code
								Image classification	Image segmentation (medical)	Object detection, light-weight	Object detection, heavy-weight	Speech recognition	NLP	Recom-mendation	Reinforce-ment Learning		
								ImageNet	KiTS19	OpenImages	COCO	LibriSpeech	Wikipedia	1TB Clickthrough	Go		
								ResNet	3D U-Net	RetinaNet	Mask R-CNN	RNN-T	BERT [1]	DLRM	Minigo		
	Azure	ND96amsr_A100_v4_n1	AMD EPYC 7V12 64-Core Processor	2	NVIDIA A100-SXM-80GB	8	merlin_hugectr NVIDIA Release 22.04							1.849		details	code
	Azure	ND96amsr_A100_v4_n1	AMD EPYC 7V12 64-Core Processor	2	NVIDIA A100-SXM-80GB	8	MxNet NVIDIA Release 22.04	29.374	24.116							details	code
	Azure	ND96amsr_A100_v4_n1	AMD EPYC 7V12 64-Core Processor	2	NVIDIA A100-SXM-80GB	8	PyTorch NVIDIA Release 22.04			87.093	44.165	33.546	19.213			details	code
	Azure	ND96amsr_A100_v4_n2	AMD EPYC 7V12 64-Core Processor	4	NVIDIA A100-SXM-80GB	16	PyTorch NVIDIA Release 22.04			46.958						details	code
	Azure	ND96amsr_A100_v4_n4	AMD EPYC 7V12 64-Core Processor	8	NVIDIA A100-SXM-80GB	32	PyTorch NVIDIA Release 22.04				13.675					details	code
	Azure	ND96amsr_A100_v4_n8	AMD EPYC 7V12 64-Core Processor	16	NVIDIA A100-SXM-80GB	64	MxNet NVIDIA Release 22.04	4.587								details	code
	Azure	ND96amsr_A100_v4_n8	AMD EPYC 7V12 64-Core Processor	16	NVIDIA A100-SXM-80GB	64	PyTorch NVIDIA Release 22.04			15.647			3.079			details	code
	Azure	ND96amsr_A100_v4_n9	AMD EPYC 7V12 64-Core Processor	18	NVIDIA A100-SXM-80GB	72	MxNet NVIDIA Release 22.04		3.437							details	code
	Azure	ND96amsr_A100_v4_n16	AMD EPYC 7V12 64-Core Processor	32	NVIDIA A100-SXM-80GB	128	PyTorch NVIDIA Release 22.04					17.380				details	code
	Google	tpu-v4-2048	AMD EPYC 7B12	512	TPU-v4	1024	Tensorflow v2.8.0				3.025					details	code
	Google	tpu-v4-4096	AMD EPYC 7B12	1024	TPU-v4	2048	Tensorflow v2.8.0			2.343	2.253					details	code
	Google	tpu-v4-6912	AMD EPYC 7B12	1728	TPU-v4	3456	Tensorflow v2.8.0	0.230					0.227			details	code
	Google	tpu-v4-8192	AMD EPYC 7B12	2048	TPU-v4	4096	Tensorflow v2.8.0	0.191					0.179			details	code
	HazyResearch	Azure ND96amsr_a100_v4_ngc22.04_pytorch	AMD EPYC 7V12	2	NVIDIA A100-SXM-80GB	8	PyTorch NVIDIA Release 22.04						17.402			details	code
	ASUSTeK	ESC8000A-E11-8xA100-PCIE-80GB-NVBridge	AMD EPYC 7763	2	NVIDIA A100-PCIe-80GB	8	NVIDIA Release 22.04 Tensorflow, MxNet, pytorch, hugectr	31.171	25.855	151.209	46.822	32.534	25.076		269.37	details	code
	ASUSTeK	ESCN4A-E11	AMD EPYC 7773X	1	NVIDIA A100-SXM-80GB	4	NVIDIA Release 22.04 Tensorflow, MxNet, pytorch, hugectr	54.956	49.446	219.613	79.835	59.989	37.512	3.143		details	code
	Baidu	Bow-Pod16	AMD EPYC 7742	2	Graphcore Bow IPU	16	PaddlePaddle (branch: develop, commitID: b2d8f6d)						20.810			details	code
	Baidu	Bow-Pod64	AMD EPYC 7742	2	Graphcore Bow IPU	64	PaddlePaddle (branch: develop, commitID: b2d8f6d)						6.740			details	code
	Baidu	1_node_8_A100_PyTorch	Intel(R) Xeon(R) Platinum 8350C	2	NVIDIA A100-SXM-80GB	8	PyTorch NVIDIA Release 22.04						19.392			details	code
	Baidu	XMan4.0_1_node_8_A100_PaddlePaddle	Intel(R) Xeon(R) Platinum 8350C	2	NVIDIA A100-SXM-80GB	8	PaddlePaddle (branch: develop, commitID: 108aeb2)						16.599			details	code
	CASIA	XIANGXUE-3B_A100-PCIe-40GBx16_7773X_mxnet	AMD EPYC 7773X	2	NVIDIA A100-PCIe-40GB	16	MxNet NVIDIA Release 21.09	21.444	15.871							details	code
	CASIA	XIANGXUE-3B_A100-PCIe-40GBx16_7773X_pytorch	AMD EPYC 7773X	2	NVIDIA A100-PCIe-40GB	16	Pytorch NVIDIA Release 21.09				34.751					details	code
	CASIA	XIANGXUE-3B_A100-PCIe-80GBx16_7763_mxnet	AMD EPYC 7763	2	NVIDIA A100-PCIe-80GB	16	MxNet NVIDIA Release 21.09	18.706								details	code
	CASIA	XIANGXUE-3B_A100-PCIe-80GBx16_7763_pytorch	AMD EPYC 7763	2	NVIDIA A100-PCIe-80GB	16	Pytorch NVIDIA Release 21.09				29.615					details	code
	CASIA	XIANGXUE-3B_A100-PCIe-80GBx16_7773X_pytorch	AMD EPYC 7773X	2	NVIDIA A100-PCIe-80GB	16	Pytorch NVIDIA Release 21.09			54.989						details	code
	CASIA	XIANGXUE-3B_A100-PCIe-80GBx16_7773X_tensorflow	AMD EPYC 7773X	2	NVIDIA A100-PCIe-80GB	16	Tensorflow NVIDIA Release 21.09								174.41	details	code

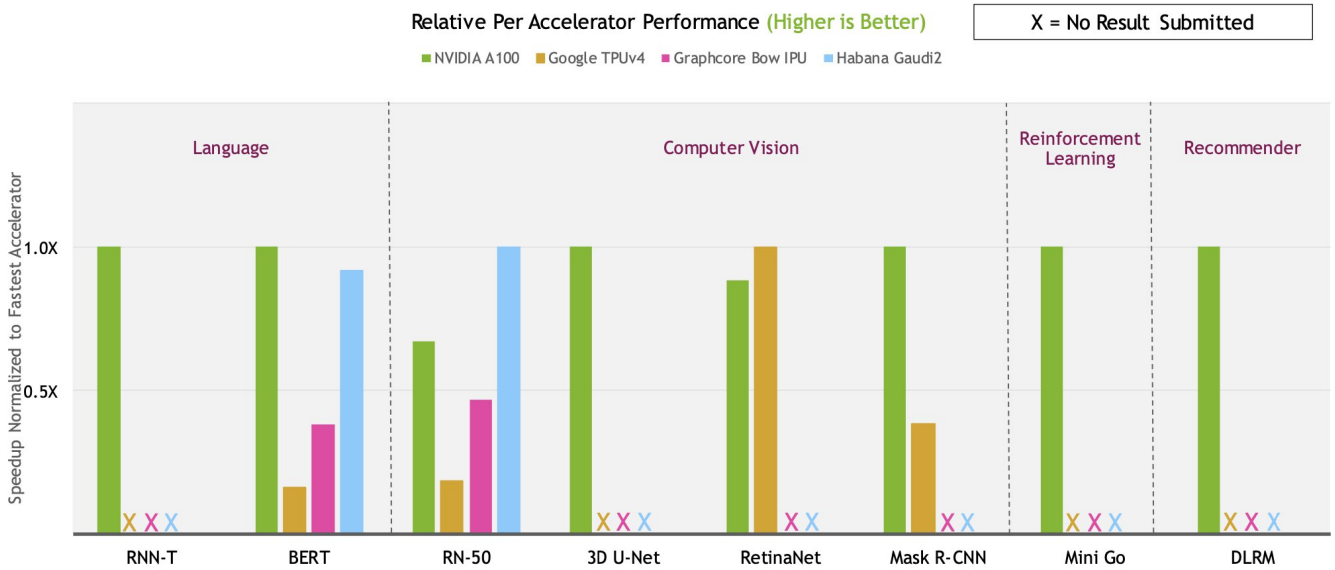
View Fullscreen

ML Perf - Result v2.0

Juin 29, 2022 – Training

NVIDIA A100 CONTINUES LEADERSHIP – FASTEST ON 6 OF 8 TESTS

Fourth Submission on A100 - Only Platform to Submit Across All Benchmarks



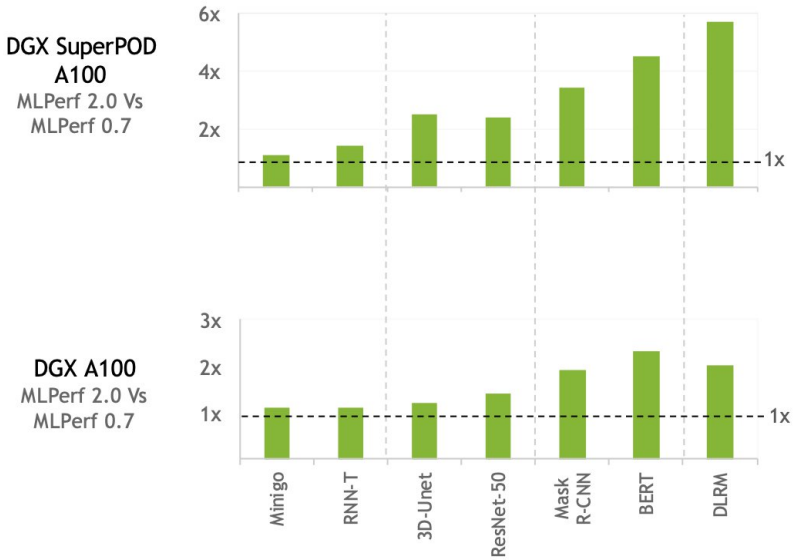
To calculate per-chip performance, this chart normalizes every submission to the closest scale of the fastest competitor. The fastest competitors are shown with 1x. To determine the fastest competitor, we selected the scale common to most submitters. | Format: Chip count, Submitter, MLPerf-ID | ResNet-50: 8x NVIDIA 2.0-2069, 3456x Google 2.0-2011, 16x Graphcore 2.0-2047, 8x Intel-HabanaLabs 2.0-2073 | BERT: 8x Inspur 2.0-2070, 3456x Google 2.0-2011, 16x Graphcore 2.0-2045, 8x Intel-HabanaLabs 2.0-2073 | DLRM: 8x Inspur 2.0-2068 | Mask R-CNN: 384x NVIDIA 2.0-2099, 1024x Google 2.0-2009 | RetinaNet: 1280x NVIDIA 2.0-2103, 2048x Google 2.0-2010 | RNN-T 8x Inspur 2.0-2066 | 3D-UNet: 8x H3C 2.0-2060, | MiniGo: 8x H3C 2.0-2059 | MLPerf name and logo are trademarks. See www.mlperf.org for more information.

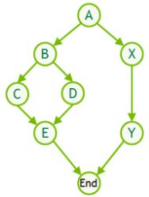
ML Perf - Évolution

MLPerf

NVIDIA AI DELIVERS 6X HIGHER PERFORMANCE IN 2 YEARS

Fueled By Continuous Software Innovation On Same Ampere Architecture GPUs





CUDA Graphs
Minimize Launch Overheads

cuDNN

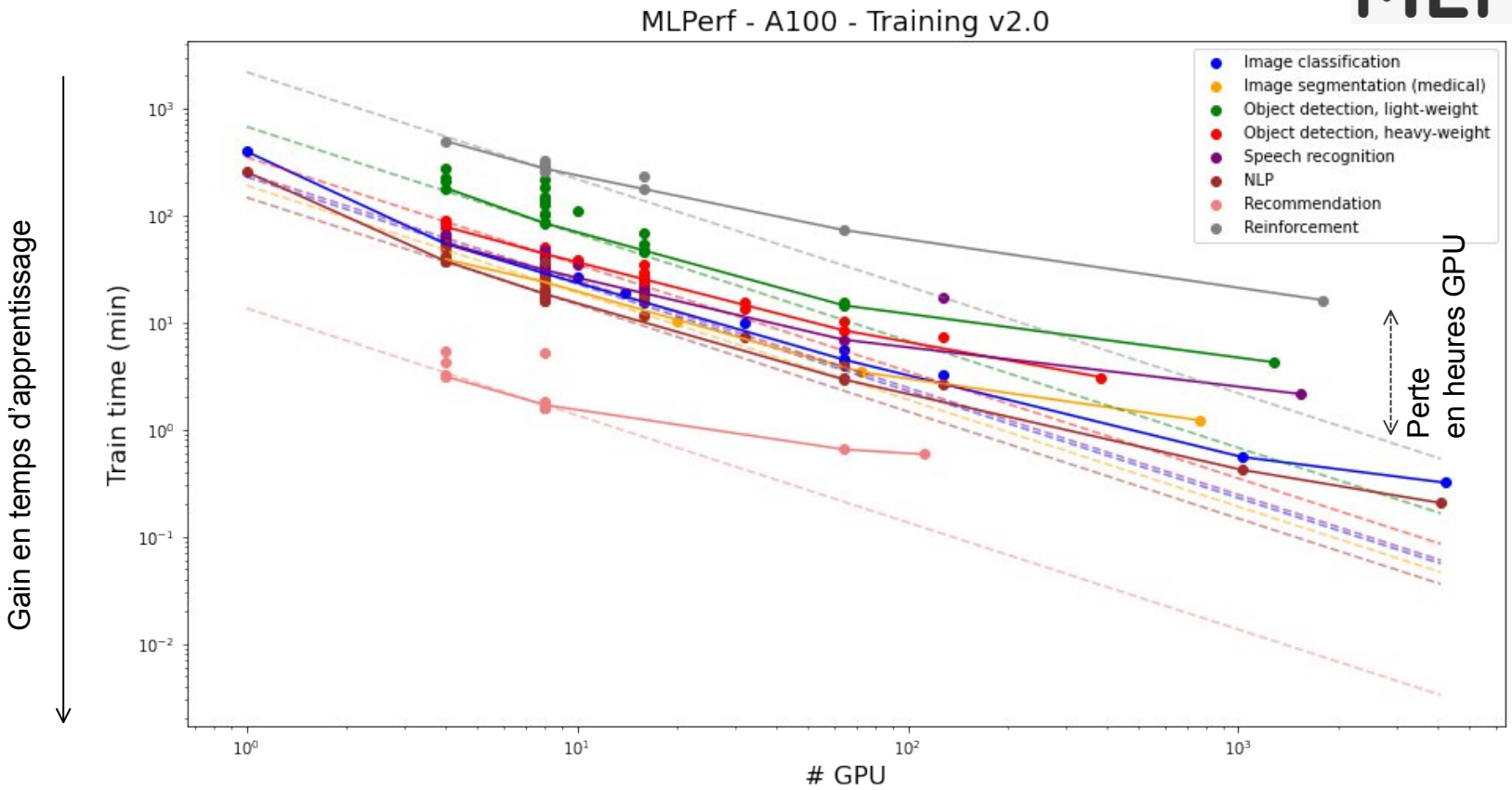
Optimized Libraries
Optimized Kernels in cuBLAS/cuDNN/...

DALI
GPU Accelerated Pre-Processing

DL Network Stack
MagnumIO - IB SHARP, NCCL

Key Technology Advancements

ML Perf - Scaling



Torch IMage Models

TIMM is a library containing SOTA **computer vision models**, layers, utilities, optimizers, schedulers, data-loaders, augmentations, and training/evaluation scripts.

doc : https://timm.fast.ai/

papers with code :
https://paperswithcode.com/lib/timm

Documentation

https://huggingface.co/docs/timm/index

Hugging Face

The AI community building the future.

Build, train and deploy state of the art models powered by the reference open source in machine learning.

Star 104,839

- **Transformers**
  State-of-the-art ML for Pytorch, TensorFlow, and JAX.
- **Diffusers**
  State-of-the-art diffusion models for image and audio generation in PyTorch.
- **Hub**
  Host Git-based models, datasets and Spaces on the Hugging Face Hub.
- **Datasets**
  Access and share datasets for computer vision, audio, and NLP tasks.
- **Gradio**
  Build machine learning demos and other web apps, in just a few lines of Python.
- **Hub Python Library**
  Client library for the HF Hub: manage repositories from your Python runtime.
- **Huggingface.js**
  A collection of JS libraries to interact with Hugging Face, with TS types included.
- **Transformers.js**
  Community library to run pretrained models from Transformers in your browser.
- **Inference API**
  Use more than 50k models through our public inference API, with scalability built-in.
- **Inference Endpoints**
  Easily deploy your model to production on dedicated, fully managed infrastructure.
- **PEFT**
  Parameter efficient finetuning methods for large models
- **Accelerate**
  Easily train and use PyTorch models with multi-GPU, TPU, mixed-precision.
- **Optimum**
  Fast training and inference of HF Transformers with easy to use hardware optimization tools.
- **Optimum Neuron**
  Train and Deploy Transformers & Diffusers with AWS Trainium and AWS Inferentia.
- **Tokenizers**
  Fast tokenizers, optimized for both research and production.
- **Evaluate**
  Evaluate and report model performance easier and more standardized.
- **Tasks**
  All things about ML tasks: demos, use cases, models, datasets, and more!
- **Datasets-server**
  API to access the contents, metadata and basic statistics of all Hugging Face Hub datasets.
- **Simulate**
  Create and share simulation environments for intelligent agents and synthetic data generation.
- **Amazon SageMaker**
  Train and Deploy Transformer models with Amazon SageMaker and Hugging Face DLCs.
- **timm**
  State-of-the-art computer vision models, layers, optimizers, training/evaluation, and utilities.
- **Safetensors**
  Simple, safe way to store and distribute neural networks weights safely and quickly.
- **AutoTrain**
  AutoTrain API and UI

Frameworks & Optimisation du compileur

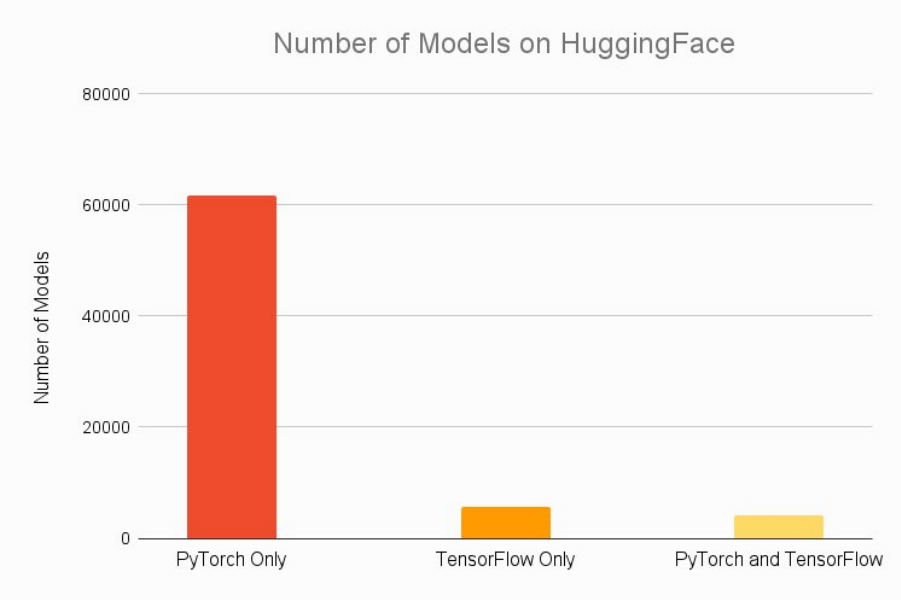
Pytorch vs Tensorflow ◂

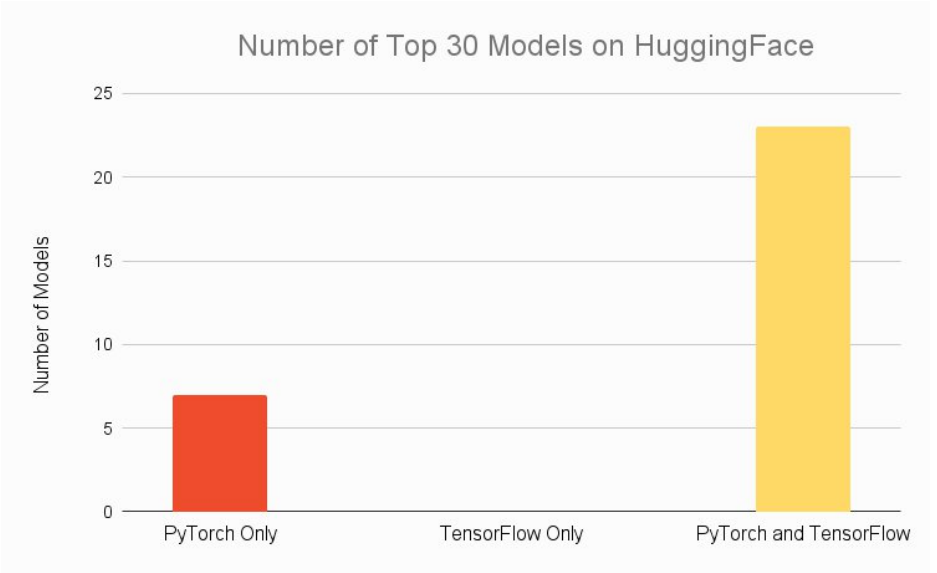
Compilation ◂

JAX ◂

Pytorch 2.0 ◂

Pytorch vs Tensorflow

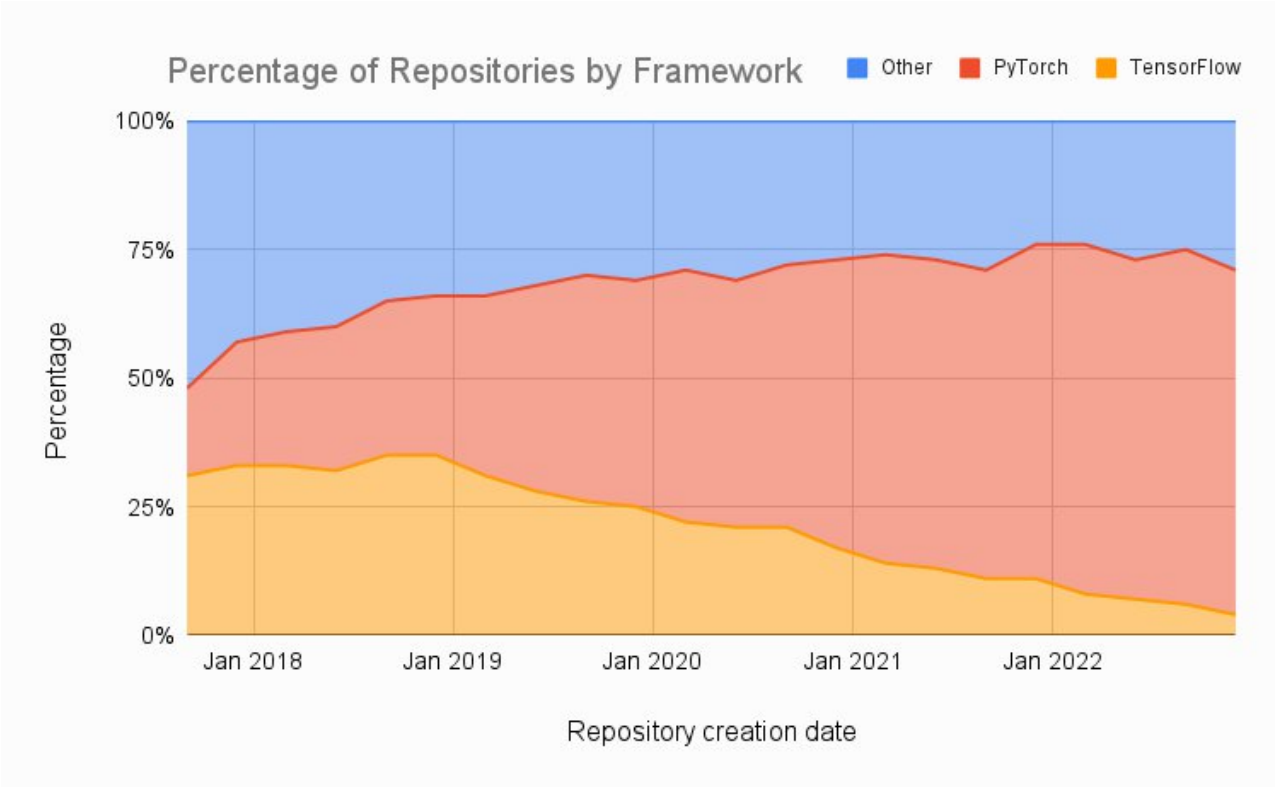




https://www.assemblyai.com/blog/pytorch-vs-tensorflow-in-2023/

Pytorch vs Tensorflow

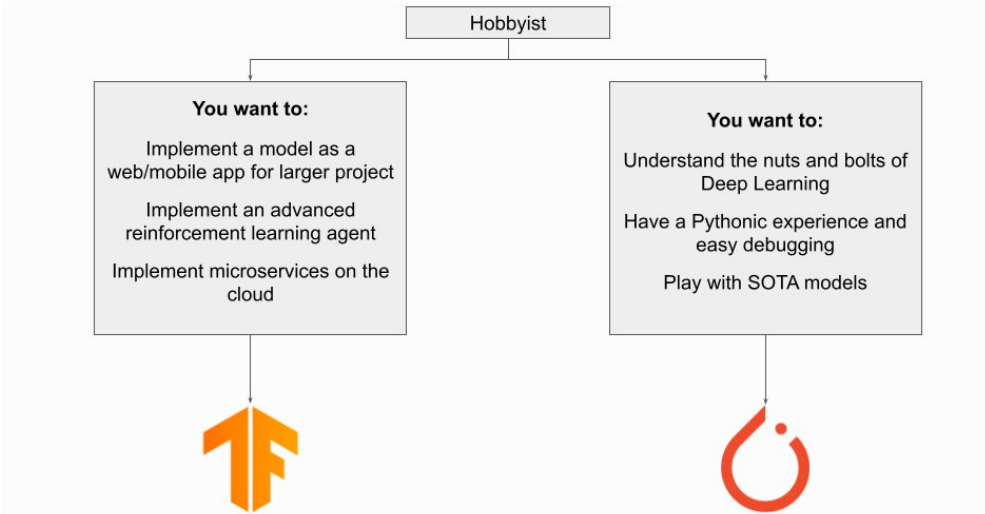
Paper With Code :



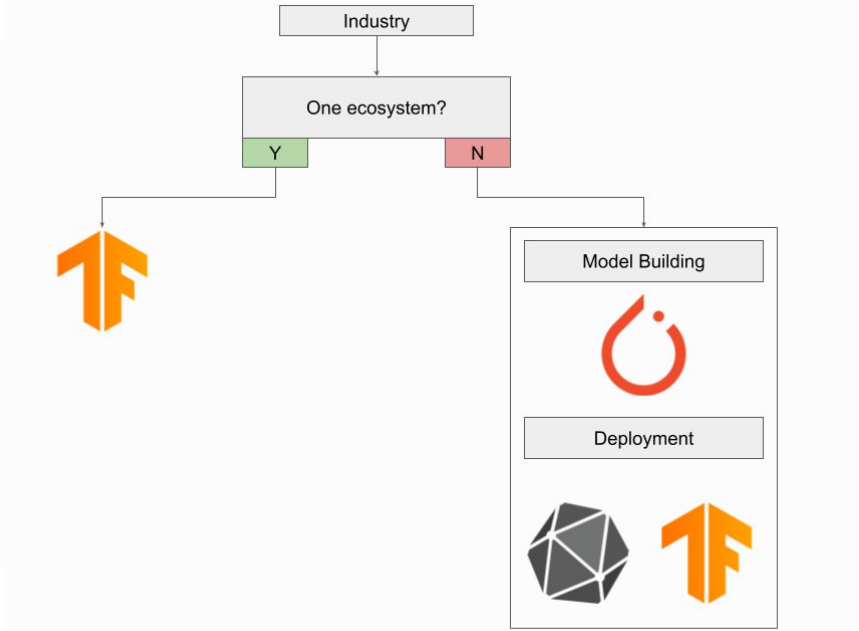
https://www.assemblyai.com/blog/pytorch-vs-tensorflow-in-2023/

Pytorch vs Tensorflow

What if I'm in Industry?



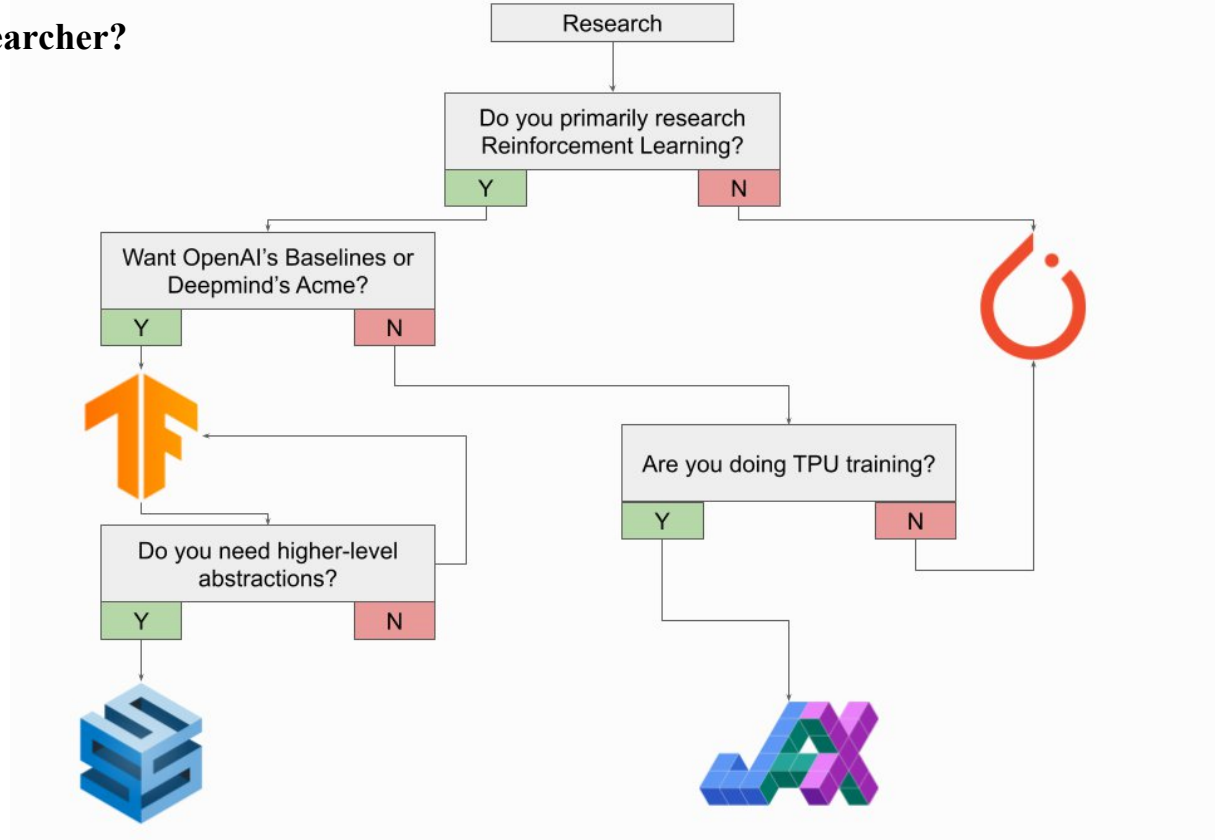
What if I'm in Industry?



https://www.assemblyai.com/blog/pytorch-vs-tensorflow-in-2023/

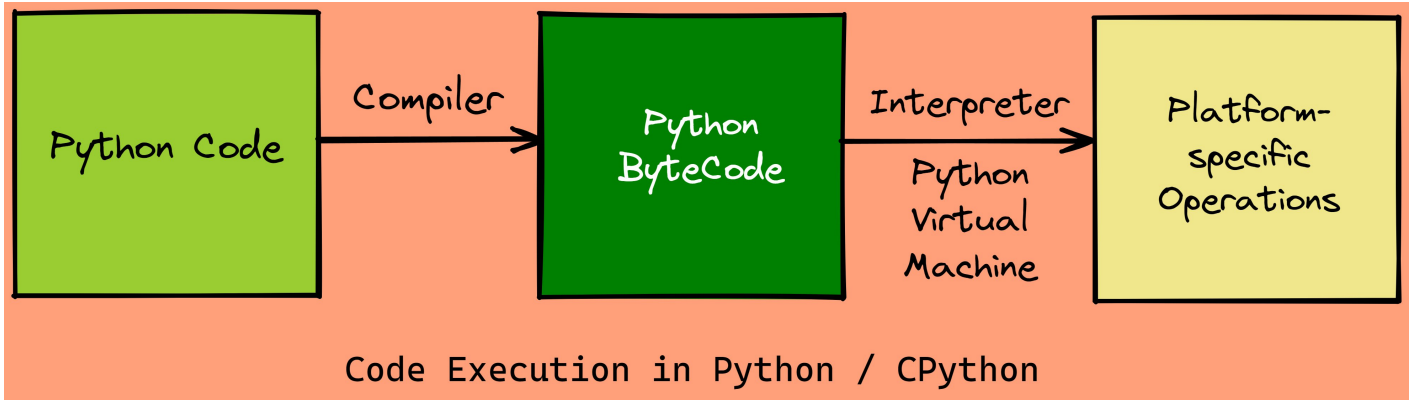
Pytorch vs Tensorflow vs JAX

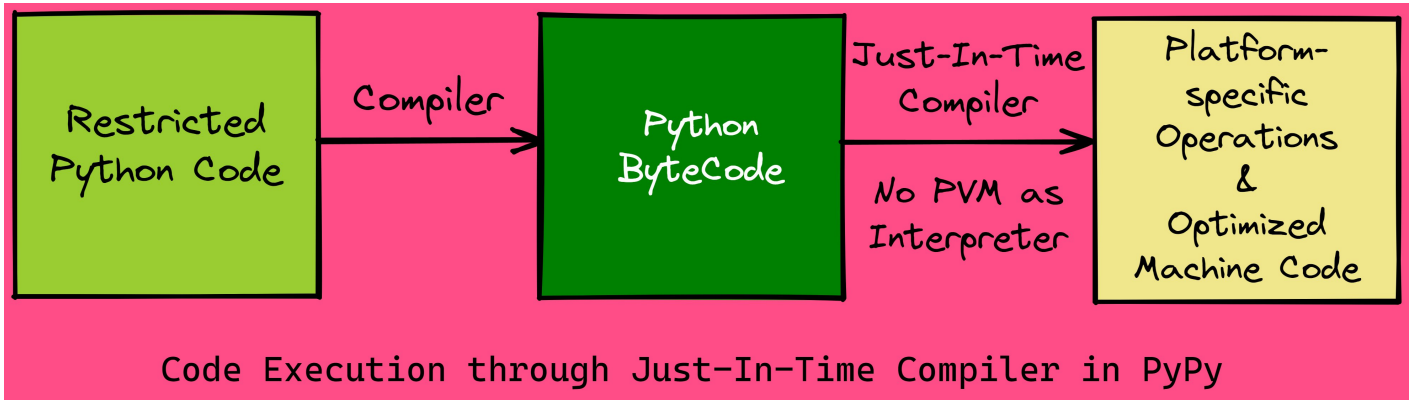
What if I'm a Researcher?



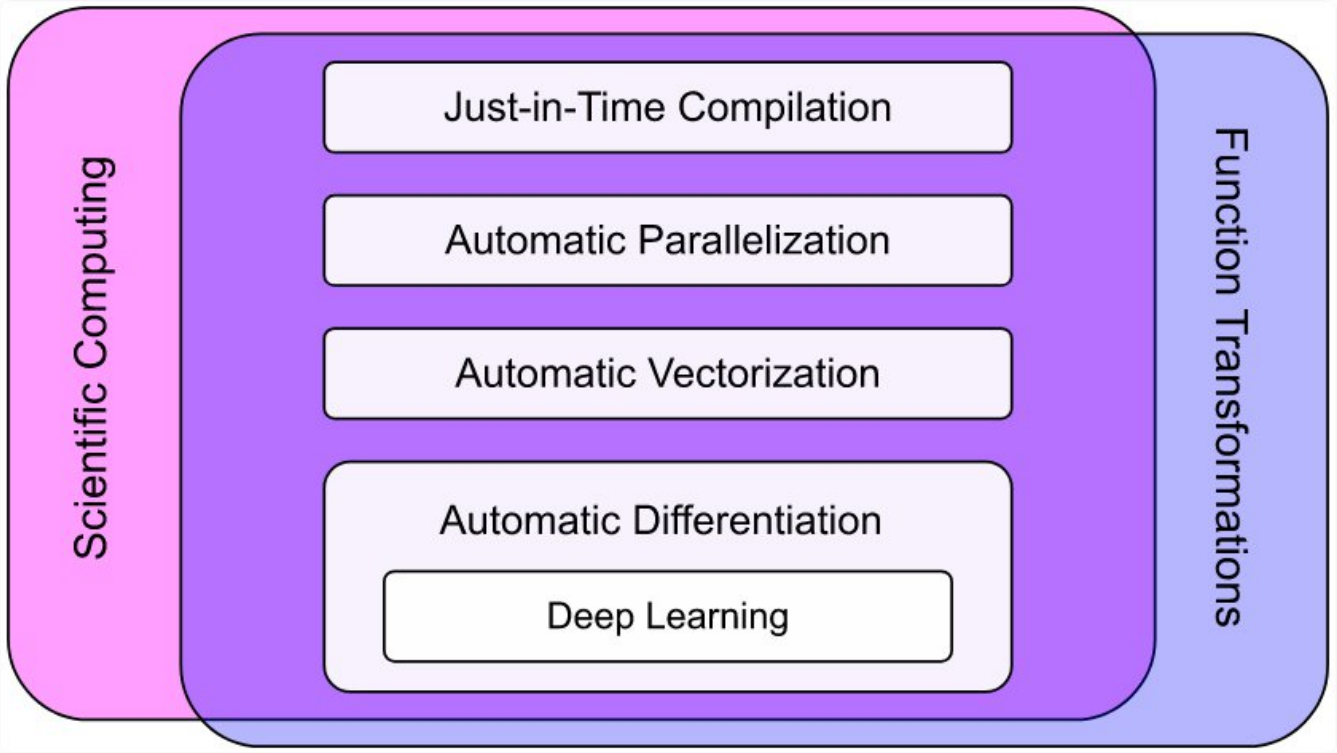
https://www.assemblyai.com/blog/pytorch-vs-tensorflow-in-2023/

JIT : Just In Time Compilation





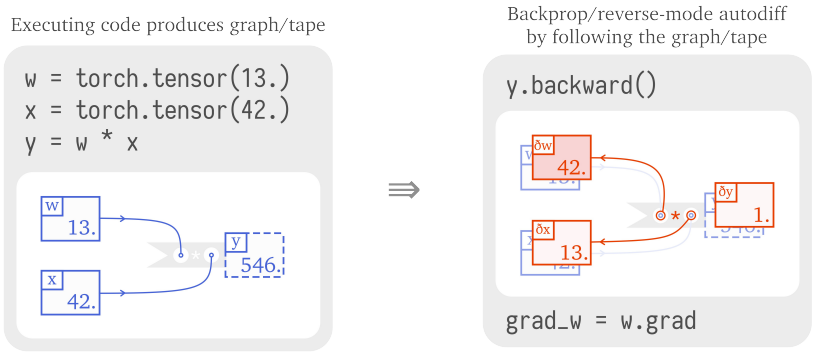
Jax



JAX lies at the intersection of Scientific Computing and Function Transformations, yielding a wide range of capability beyond the ability to train Deep Learning models

Pytorch vs Jax

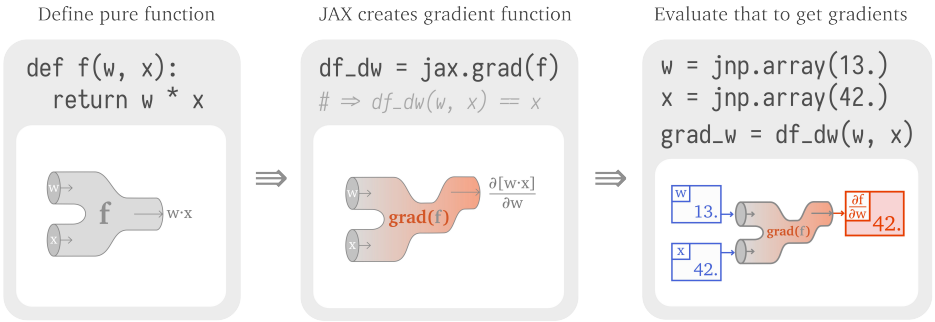
PyTorch



Pros :
Pythonic, dynamic, popular framework,
NVIDIA GPU oriented, Jean Zay adapted

Cons :
Slower compare to some other frameworks (Jax, Mxnet, ...)

JAX

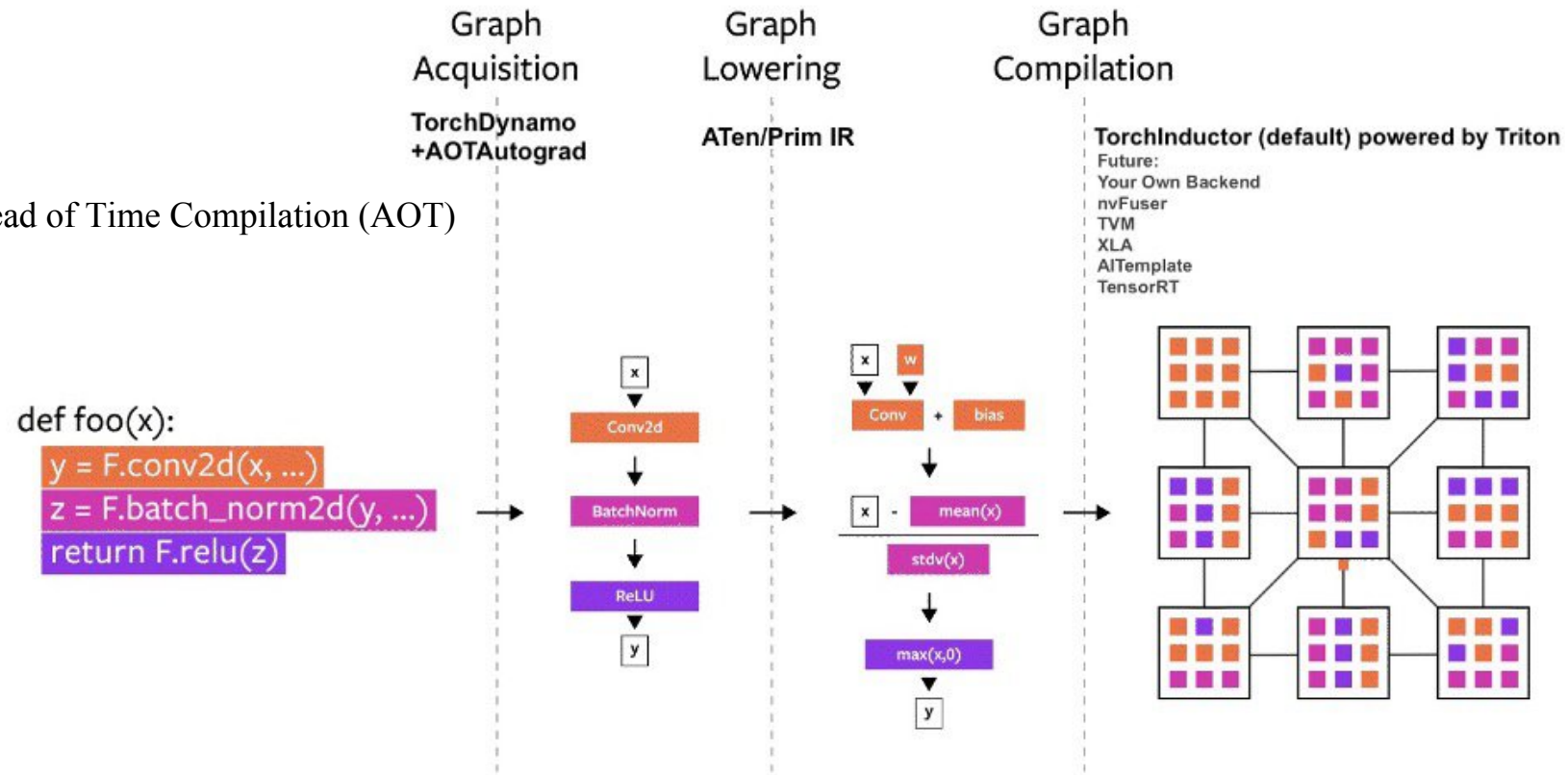


Pros :
Fast Computation
numpy-like, functional programming,
Hessian computation (2nd order)
efficiency

Cons :
Google/TPU oriented, Jean Zay unadapted

Pytorch 2.0

Ahead of Time Compilation (AOT)



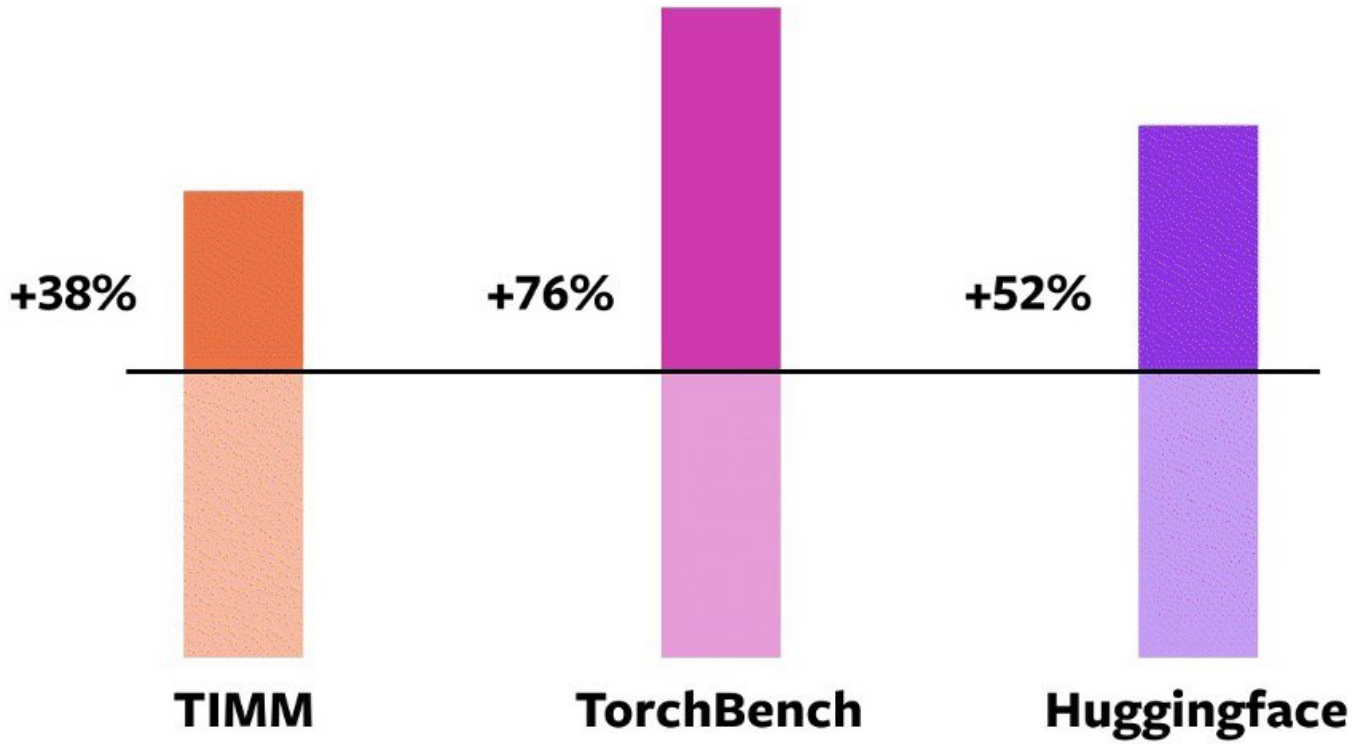
Pytorch 2.0

```
# API NOT FINAL
# default: optimizes for large models, low compile-time
#          and no extra memory usage
torch.compile(model)

# reduce-overhead: optimizes to reduce the framework overhead
#                and uses some extra memory. Helps speed up small models
torch.compile(model, mode="reduce-overhead")

# max-autotune: optimizes to produce the fastest model,
#               but takes a very long time to compile
torch.compile(model, mode="max-autotune")
```

Pytorch 2.0



Annexes

MLPerf – Resnet50

NVIDIA A100
MXNET, Horovod, DALI

train samples : 1281167

- For training: Normalization, Scale to 256x256, Random resized crop to 224x224, Random horizontal flip
- For validation: Normalization, Scale to 256x256, Center crop to 224x224

	Num epochs	Warmup epochs	Lr scheduler	Global Batch size	Optimizer	LR	WD	mom	Label smoothing	Throughput samples/s
REF (TF)	41	5	polynomial 2	**2048**	lars	**8.5**	2e-4	0.9	0.1	-
4 GPU	35	2	polynomial 2	**1632**	lars	**7.4**	5e-5	0.9	0.1	13900
8 GPU	37	2	polynomial 2	**3264**	lars	**10.5**	5e-5	0.9	0.1	27180
64 GPU	37	2	polynomial 2	**3264**	lars	**10.5**	5e-5	0.9	0.1	167700
1024 GPU	**60**	**17**	polynomial 2	**35840**	lars	**21.9**	**2.5e-5**	**0.94**	0.1	2300000
4216 GPU	**90**	**31**	polynomial 2	**67456**	lars	**24.699**	**1e-4**	**0.951807**	0.1	40000000

Training : Unet 3D

NVIDIA A100
MXNET, Horovod, DALI

training input shape : 128 x 128 x 128
validation input shape : 128 x 128 x 128

train samples : 168

	Num epochs	Warmup epochs	Lr scheduler	Global Batch size	Optimizer	LR	WD	mom	oversampling	Throughput samples/s
8 GPU	~2200	1000	constant	**56**	**nag**	**2.**	0.	0.9	0.4	288
72 GPU	~2200	1000	constant	**72**	**nag**	**2.**	0.	0.9	0.4	1810
768 GPU	~2200	1500	constant	**84**	**nag**	**1.5**	0.	0.9	0.4	6100

Training : SSD

NVIDIA A100
Pytorch, DALI

Data Augmentation : random horizontal flip, Normalize, resize image to 800x800

train samples : 4572

	Num epochs	Warmup epochs	Lr scheduler	Global Batch size	Optimizer	LR	WD
8 GPU	**4**	0	constant	**256**	**adam**	**8.5 e-5**	0.
64 GPU	**4**	1	constant	**256**	**adam**	**1 e-4**	0.
1280 GPU	**7**	1	constant	**1280**	**adam**	**1.2 e-4**	0.

Training : Mask RCNN

NVIDIA A100
Pytorch

min image size : 800
max image size : 1300

train samples : 40000

	Num epochs	Warmup steps	Lr scheduler	Global Batch size	Optimizer	LR	WD	mom	Throughput samples/s
8 GPU	**14**	625	Steps (12000, 16000)	**96**	**sgd**	**0.12**	1e-4	0.9	640
64 GPU	**14**	625	Steps (9000, 12000)	**128**	**sgd**	**0.16**	1e-4	0.9	3300
384 GPU	**17**	1800	Steps (5625, 7500)	**256**	**sgd**	**0.24**	1e-4	0.9	11400

Training : BERT

NVIDIA A100
Pytorch

Max predictions per seq : 76

train samples : ~3000000

	Num epochs	Warmup epochs	Lr scheduler	Global Batch size	Optimizer	LR	Beta 1	Beta 2	WD
8 GPU	~2800000	0	Linear decay	448	**lamb**	**4.25 e-4**	0.9	0.999	0.
64 GPU	~3000000	100	Linear decay	**3072**	**lamb**	**1.5 e-3**	**0.83**	**0.925**	0.
1024 GPU	~3000000	256	Linear decay	**4096**	**lamb**	**2.55 e-3**	**0.71**	**0.88**	0.
4096 GPU	~5500000	290	Linear decay	**16384**	**lamb**	**3.3 e-3**	**0.75**	**0.9**	0.

Training : RNN-T

NVIDIA A100
Pytorch, DALI

train samples : 278528

	Num epochs	Warmup epochs	Lr scheduler	Global Batch size	Optimizer	LR	Beta 1	Beta 2	WD	Throughput samples/s
8 GPU	~ 50	5	**Poly 0.92**	**1536**	**lamb**	**7.2 e-3**	0.9	0.999	1e-3	7560
64 GPU	~ 55	6	**Poly 0.939**	**2048**	**lamb**	**7 e-3**	0.9	0.999	1e-3	39500
1536 GPU	~106	16	**Poly 0.97**	**6144**	**lamb**	**0.014**	0.82	0.98	1e-3	250000

+ gradient clip norm : 1

Training : DLRM

NVIDIA A100
Merlin hugectr

train samples : 4000000000

	Num epochs	Warmup steps	Lr scheduler	Global Batch size	Optimizer	LR	Throughput samples/s
8 GPU	2	2750	Poly 2	**55296**	sgd	**24.**	37000000
64 GPU	2	2500	Poly 2	**71680**	sgd	**26.**	98380000
112 GPU	2	2500	Poly 2	**71680**	sgd	**26.**	119000000

Training : Minigo

NVIDIA A100
TensorFlow

	Num epochs	Warmup steps	Lr scheduler	Global Batch size	Optimizer	LR	WD	Throughput games/s
8 GPU	71	128	Steps (10000, 20000)	4096	sgd	0.16	1 e-4	**0.0024**
64 GPU	81	128	Steps (10000, 20000)	4096	sgd	0.16	1 e-4	**0.0097**
1792 GPU	96	128	Steps (10000, 20000)	4096	sgd	0.16	1 e-4	**0.07**

Training HPC : cosmoflow

NVIDIA A100
Mxnet, Horovod, DALI

train samples : 524288

	Num epochs	Warmup epochs	Lr scheduler	Global Batch size	Optimizer	LR	WD	Throughput samples/s
128 GPU	**19**	4	epochs (16, 32)	**128**	sgd	0.004	0.	**8600**
1024 GPU	**35**	0	epochs (32, 64)	**1024**	sgd	0.012	0.	**64600**

+ dropout : 0.5

Training HPC : deepcam

NVIDIA A100
Pytorch, DALI

train samples : 121266

	Num epochs	Warmup steps	Lr scheduler	Global Batch size	Optimizer	LR	Beta 1	Beta 2	WD
512 GPU	16	200	Steps(1100, 4096)	1024	lamb	0.004	0.9	0.999	0.01
2048 GPU	25	400	Steps(800)	2048	lamb	0.0055	0.9	0.999	0.01

+ gradient clip norm : 1