PyTorch 2 Benchmark

Boyuan Feng, Bin Bao
PyTorch

Research Papers: < 10 Models

(limited coverage)

PyTorch Benchmark: ~180 Models

(extensive coverage)

Torch Bench

(75 highly cited research models)

Hugging Face

(46 models, mostly transformers)

TIMM

(61 models, primarily vision models)

$$\frac{\text{eager peak memory}}{\text{compiled peak memory}}$$

peak memory compression ratio

(higher is better)

$$\frac{\text{eager latency}}{\text{compiled latency}}$$

speedup

(higher is better)

$$\frac{\text{num of supported models}}{\text{total num of models}}$$

pass rate

(higher is better)

$$\text{time of end-to-end compile}$$

compilation time

(lower is better)

`--accuracy` or `--performance`: selects between checking correctness and measuring speedup (both are run for dashboard).

`--training` or `--inference`: selects between measuring training or inference (both are run for dashboard).

`--device=cuda` or `--device=cpu`: selects device to measure.

`--amp`, `--bfloat16`, `--float16`, `--float32`: selects precision to use `--amp` is used for training and `--bfloat16` for inference.

`--cold-start-latency`: disables caching to accurately measure compile times.

`--backend=inductor`: selects TorchInductor as the compiler backend to measure. Many more are available, see `--help`.

Backend Examples: eager, aot_eager, inductor, cudagraphs

Command for PyTorch 2 [ASPLOS'24] Artifact Evaluation

```
TORCHINDUCTOR_MAX_AUTOTUNE=1 ./benchmarks/dynamo/huggingface.py \
    --performance --no-skip \
    -dcuda --float16 --inference \
    --inductor --freezing \
    --output=`pwd`/results.csv
```

Common Mistakes

bfloat16 vs float32
tf32 on/off

ignore important flags
(e.g., max autotune, freezing)

skip cudagraph

More Info

https://pytorch.org/assets/pytorch2-2.pdf

TorchInductor Performance DashBoard

Time Range: Last 7 Days | Granular...: hour | Suite: Torchbench | Mode: training | Precision: amp | Branch: main | Base Commit: 5669334175 (2024/04/12) —Diff→ Branch: main | New Commit: 704fac5618 (2024/04/18)

**This report was generated by CI running on PyTorch main branch at commit 704fac5618 on 2024/04/18 comparing with main branch at commit 5669334175. The running logs per shard are: Torchbench (#1, #2, #3, #4) Huggingface (#1, #2, #3) TIMM models (#1, #2, #3, #4, #5).*

Passrate (threshold = 90%) Base value (L) → New value (R)

Inductor config	Torchbench	Huggingface	TIMM models	[Dynamic]	[Blueberries]
cudagraphs	96%, 65/68	98%, 45/46	100%, 61/61	78%, 7/9	67%, 2/3
cudagraphs_d...	91%, 61/67	65%, 30/46	98%, 60/61 → 97%	78%, 7/9	67%, 2/3
default	97%, 66/68	98%, 45/46	100%, 61/61	78%, 7/9	67%, 2/3

Geometric mean speedup (threshold = 0.95x)

Inductor config	Torchbench	Huggingface	TIMM models	[Dynamic]	[Blueberries]
cudagraphs	2.01x	2.03x → 2.09x	1.81x → 1.82x	1.87x → 1.88x	2.81x → 2.84x
cudagraphs_d...	1.98x	2.07x	1.61x → 1.62x	1.87x → 1.89x	2.81x → 2.83x
default	1.30x → 1.27x	1.80x → 1.82x	1.71x → 1.72x	1.34x → 1.32x	1.29x → 1.25x

Mean compilation time (seconds)

Inductor config	Torchbench	Huggingface	TIMM models	[Dynamic]	[Blueberries]
cudagraphs	96s → 97s	94s → 96s	143s	29s	62s → 63s
cudagraphs_d...	105s → 106s	139s → 142s	175s	38s	63s → 64s
default	86s → 87s	88s → 90s	138s	27s → 28s	59s → 60s

Peak memory footprint compression ratio (threshold = 0.9x)

Inductor config	Torchbench	Huggingface	TIMM models	[Dynamic]	[Blueberries]
cudagraphs	0.79x	1.26x	1.11x	0.75x	0.94x
cudagraphs_d...	0.85x	1.08x	1.11x	0.75x	0.94x
default	0.79x	1.26x	1.11x	0.75x	0.93x

Notebook Demo

https://fb.me/pt2-bench-asplos24