Habitat

A Runtime-Based Computational Performance Predictor for Deep Neural Network Training

Geoffrey X. Yu, Yubo Gao, Pavel Golikov, Gennady Pekhimenko

Get started: github.com/geoffxy/habitat
What this talk is about

The problem:
- Many GPUs available for deep neural network (DNN) training
  - Each has a different $\text{cost}$ and $\text{performance}$
  - Which should a user choose for training?

Key observations:
- DNN training computation is $\text{highly repetitive}$
- Predict a GPU’s training performance by $\text{predicting}$ the execution time of a $\text{single iteration}$
What this talk is about

The problem:

• Many GPUs available for deep neural network (DNN) training
  • Each has a different 📊 cost and 🔥 performance
  • Which should a user choose for training?

Our work:

• Use an existing GPU to 👀 predict execution times on a different GPU using ≈ wave scaling
  and 🎯 pre-trained multilayer perceptrons (MLPs)

• Implement ideas in a new tool called ✨ Habitat (open source, supports PyTorch)

• Show 🔍 two case studies where Habitat leads users to the ✔️ correct GPU choice

Habitat is open source: github.com/geoffxy/habitat
Deep neural networks (DNNs) are everywhere

- **Image Classification**
  - ResNet [CVPR’16]
  - VGG [ICLR’15]
  - AlexNet [NeurIPS’12]

- **Machine Translation**
  - Transformer [NeurIPS’17]
  - Seq2Seq NMT [NeurIPS’14]

- **Object Detection**
  - YOLO [CVPR’16]
  - SSD [ECCV’16]
  - Fast R-CNN [ICCV’15]

- **Speech Recognition**
  - Deep Speech 2 [ICML’16]
  - End-to-End w/ RNNs [ICML’14]

But they are often **computationally expensive** to train!

Habitat is open source: github.com/geoffxy/habitat
A Cambrian explosion in hardware for training

GPUs (workstation, cloud)
- A100
- V100
- P100
- T4
- 2080Ti
- 2070
- P4000
- TITAN V
- 3090

TPUs (v2, v3, v4)

Other Emerging Accelerators
- Cerebras WSE
- Habana Gaudi
- AWS Trainium

Which accelerator should you use?

Habitat is open source: github.com/geoffxy/habitat
A Cambrian explosion in hardware for training

Which GPU should you use?

GPUs (workstation, cloud):
- A100
- V100
- P100
- T4
- 2080Ti
- 2070
- P4000
- TITAN V
- 3090

TPUs (v2, v3, v4):

Other Emerging Accelerators:
- Cerebras WSE
- Habana Gaudi
- AWS Trainium
Choosing a GPU: The paradox of choice

<table>
<thead>
<tr>
<th>Cost?</th>
<th>Performance?</th>
</tr>
</thead>
<tbody>
<tr>
<td>🧨</td>
<td>▶️</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Workstation</th>
<th>On-Premise</th>
<th>Cloud</th>
</tr>
</thead>
<tbody>
<tr>
<td>3090</td>
<td>P100</td>
<td>V100</td>
</tr>
<tr>
<td></td>
<td>T4</td>
<td></td>
</tr>
</tbody>
</table>

No one-size-fits-all choice. The “correct” choice depends on the user’s needs!
Why not just…

- Measure directly?
  - ₪ Need to pay to access the GPU(s)
  - 😴 Tedious to repeat for many models

- Use existing benchmarking results?
  - ⚠ Not available for all models / GPUs

- Use simple heuristics?
  - ❌ Do not always work

Simple heuristics can lead to high (> 43%) prediction errors!
Key observations

- Deep learning users may already have an **existing GPU**
- DNN training is a **repetitive process** (short training iterations)
- Use **existing GPU** to make iteration execution time **predictions** for **other GPUs**

Make measurements on the 2080 Ti ... ➔ This work ➔ A100 ➔ V100 ➔ T4 ➔ ... to predict execution time on the A100, V100, etc.
Habitat: A runtime-based performance predictor

1. **Profile** all operations in a training iteration on an **existing GPU**

2. Predict each operation using **wave scaling** or a **multilayer perceptron (MLP)**

3. Add predictions together to get an **iteration execution time prediction**

Habitat is an open source Python library; it supports PyTorch 1.4.0
How does Habitat work?
Background: GPU execution model

- **GPU kernels**: “work” divided into thread blocks (same code, different data)

- Streaming multiprocessors (SMs) run a finite number of blocks concurrently

- Blocks **round-robin scheduled** onto SMs

- GPU kernels execute in “**waves**” of thread blocks
Wave scaling

GPU A — 2 SMs

- GPU Kernel
- 8 Thread Blocks

GPU B — 4 SMs

- SM 1
- SM 2
- SM 3
- SM 4

Kernel execution time on GPU A

200 µs

Kernel execution time on GPU B

Kernel Execution Time
Wave scaling

GPU A — 2 SMs
- SM 1
- SM 2

GPU B — 4 SMs
- SM 1
- SM 2
- SM 3
- SM 4

Kernel execution time on GPU A: 200 µs

Kernel execution time on GPU B: ?
Wave scaling

GPU A — 2 SMs

SM 1

SM 2

GPU B — 4 SMs

SM 1

SM 2

SM 3

SM 4

Wave execution time on GPU A

100 µs

Wave execution time on GPU B

?

Scaling Factors

- ▲ Memory bandwidth
- 🌊 Wave size
- 🕒 Clock frequency

Wave Execution Time

Habitat is open source: github.com/geoffxy/habitat
### Wave scaling

**GPU A — 2 SMs**
- SM 1
- SM 2

**GPU B — 4 SMs**
- SM 1
- SM 2
- SM 3
- SM 4

**Wave execution time on GPU A**
- 100 µs

**Wave execution time on GPU B**
- ?

**Scaling Factors**
- Memory bandwidth
- Wave size
- Clock frequency

Habitat is open source: [github.com/geoffxy/habitat](https://github.com/geoffxy/habitat)
Wave scaling

- **GPU A** — 2 SMs
  - SM 1
  - SM 2

- **GPU B** — 4 SMs
  - SM 1
  - SM 2
  - SM 3
  - SM 4

- **Wave execution time on GPU A**: 100 µs

- **Scaling Factors**
  - ✅ Memory bandwidth
  - 🌊 Wave size
  - ⏰ Clock frequency

Habitat is open source: [github.com/geoffxy/habitat](https://github.com/geoffxy/habitat)
Wave scaling

GPU A — 2 SMs

- SM 1
- SM 2

GPU B — 4 SMs

- SM 1
- SM 2
- SM 3
- SM 4

Scaling Factors
- Memory bandwidth
- Wave size
- Clock frequency

Wave execution time on GPU A: 100 µs

Wave Execution Time

Habitat is open source: github.com/geoffxy/habitat
Wave scaling

GPU A — 2 SMs
SM 1
SM 2

GPU B — 4 SMs
SM 1
SM 2
SM 3
SM 4

GPU Kernel
8 Thread Blocks

Wave execution time on GPU A

100 µs

Wave execution time on GPU B

Scaling Factors
- Memory bandwidth
- Wave size
- Clock frequency

Wave Execution Time
Wave scaling

- **GPU A — 2 SMs**
  - SM 1
  - SM 2

- **GPU B — 4 SMs**
  - SM 1
  - SM 2
  - SM 3
  - SM 4

Scaling Factors
- Memory bandwidth
- Wave size
- Clock frequency

- **Wave execution time on GPU A**
  - 100 µs

Wave Execution Time
Wave scaling

<table>
<thead>
<tr>
<th>GPU A — 2 SMs</th>
<th>GPU B — 4 SMs</th>
</tr>
</thead>
<tbody>
<tr>
<td>SM 1</td>
<td>SM 1</td>
</tr>
<tr>
<td></td>
<td>SM 2</td>
</tr>
<tr>
<td></td>
<td>SM 3</td>
</tr>
<tr>
<td></td>
<td>SM 4</td>
</tr>
</tbody>
</table>

Wave execution time on GPU A: 100 µs

Scaling Factors
- 📈 Memory bandwidth
- 🌊 Wave size
- 🔒 Clock frequency
Wave scaling

GPU A — 2 SMs

GPU B — 4 SMs

Wave execution time on GPU A

100 µs

Wave execution time on GPU B

?

Scaling Factors
- Memory bandwidth
- Wave size
- Clock frequency

Wave Execution Time

Habitat is open source: github.com/geoffxy/habitat
Wave scaling

- **GPU A** — 2 SMs
  - SM 1
  - SM 2

- **GPU B** — 4 SMs
  - SM 1
  - SM 2
  - SM 3
  - SM 4

Scaling Factors
- Memory bandwidth
- Wave size
- Clock frequency

Wave execution time on GPU A: 100 µs

Wave Execution Time
Wave scaling

GPU A — 2 SMs

- SM 1
- SM 2

GPU B — 4 SMs

- SM 1
- SM 2
- SM 3
- SM 4

Wave execution time on GPU A: 100 µs

Scaling Factors
- Memory bandwidth
- Wave size
- Clock frequency

Habitat is open source: github.com/geoffxy/habitat
Wave scaling

GPU A — 2 SMs

SM 1

SM 2

GPU B — 4 SMs

SM 1

SM 2

SM 3

SM 4

Wave execution time on GPU A: 100 µs

Wave execution time on GPU B: ?

Wave Execution Time

Scaling Factors
- Memory bandwidth
- Wave size
- Clock frequency
Wave scaling

GPU A — 2 SMs
- SM 1
- SM 2

GPU B — 4 SMs
- SM 1
- SM 2
- SM 3
- SM 4

Wave execution time on GPU A

100 µs

Scaling Factors
- 📊 Memory bandwidth
- 💧 Wave size
- 🕒 Clock frequency

Wave Execution Time

Habitat is open source: https://github.com/geoffxy/habitat
Wave scaling

GPU A — 2 SMs
- SM 1
- SM 2

GPU B — 4 SMs
- SM 1
- SM 2
- SM 3
- SM 4

Wave execution time on GPU A: 100 µs

Scaling Factors
- 🌊 Memory bandwidth
- 🌟 Wave size
- 🕒 Clock frequency

Wave Execution Time

Habitat is open source: github.com/geoffxy/habitat
Wave scaling

- **GPU A** — 2 SMs
  - SM 1
  - SM 2
  - Wave execution time on GPU A: 100 µs

- **GPU B** — 4 SMs
  - SM 1
  - SM 2
  - SM 3
  - SM 4
  - Wave execution time on GPU B: 100 µs

**Scaling Factors**
- Memory bandwidth
- Wave size
- Clock frequency
Wave scaling

**GPU A — 2 SMs**
- SM 1
- SM 2

**GPU B — 4 SMs**
- SM 1
- SM 2
- SM 3
- SM 4

---

**Scaling Factors**
- 📊 Memory bandwidth
- 🌊 Wave size
- 🕒 Clock frequency

---

**Kernel execution time on GPU A**
- 200 µs

**Kernel execution time on GPU B**
- 100 µs

---

Habitat is open source: [github.com/geoffxy/habitat](https://github.com/geoffxy/habitat)
One last wrinkle: Kernel-varying operations

- 🌊 Wave scaling assumes the same kernel is used across GPUs

- ⚠️ A few DNN operations use architecture-specific kernels ("kernel-varying")
  - Convolutions, linear (dense) layers, LSTMs

-💡 Habitat uses pre-trained multilayer perceptrons (MLP) for these operations
Evaluation

- How **accurate** are Habitat’s predictions?
- Does using Habitat lead to making “**correct**” decisions?

- **Six** GPUs (spanning three generations):
  - P4000
  - P100
  - V100
  - 2070
  - 2080Ti
  - T4

- **Five** models:
  - ResNet-50
  - Inception v3
  - GNMT
  - Transformer
  - DCGAN

- PyTorch 1.4.0
How accurate is Habitat?

- Predict iteration execution time (GPU, model, batch size)
How accurate is Habitat?

- Predict iteration execution time (GPU, model, batch size)

Predicted: 96 ms
Actual: 94 ms
Error: 1.9%
How accurate is Habitat?

Habitat makes **accurate** predictions, with an average error of **11.8%** across all configurations (30 GPU pairs x 5 models x 3 batch sizes).

Habitat is open source: [github.com/geoffxy/habitat](http://github.com/geoffxy/habitat)
Does Habitat lead to the “correct” decision?
Rent a GPU in the cloud?

Scenario: Want to train GNMT, have access to a P4000. Which cloud GPU to use, if any?
Rent a GPU in the cloud?

Scenario: Want to train GNMT, have access to a **P4000**. Which cloud GPU to use, if any?

![GNMT Throughput Normalized to the P4000](chart.png)
Rent a GPU in the cloud?

Scenario: Want to train GNMT, have access to a P4000. Which cloud GPU to use, if any?

Habitat correctly predicts that the V100 is the best choice for performance.
Rent a GPU in the cloud?

Scenario: Want to train GNMT, have access to a P4000. Which cloud GPU to use, if any?

Habitat correctly predicts that the T4 (or P4000) is the best choice for cost.
Always use the “best” GPU?

**Scenario:** Want to train DCGAN, have access to a 2080 Ti. Use the V100?
Always use the “best” GPU?

Scenario: Want to train DCGAN, have access to a 2080 Ti. Use the V100?

Habitat correctly predicts that the V100 only offers a marginal improvement (1.1x) over the 2080 Ti.
More details and results in the paper

- Prediction breakdowns
- MLP sensitivity study
- Predictions onto additional GPUs
- Discussion about extensibility
- Distributed training
- Mixed precision training
- Other types of hardware

Habitat is open source: github.com/geoffxy/habitat
Key takeaways

- 🔁 DNN computation is special (repetitive), enabling new analysis opportunities.
- 🕒 Use runtime-based information to make iteration execution time predictions.
- ✅ Habitat leads to the correct decision in the case studies.
- 🏞️ The hardware landscape is growing; users need help choosing effectively!

Habitat is open source: github.com/geoffxy/habitat
Habitat

A Runtime-Based Computational Performance Predictor for Deep Neural Network Training

Geoffrey X. Yu, Yubo Gao, Pavel Golikov, Gennady Pekhimenko

Get started: github.com/geoffxy/habitat