CS 181AI
Lecture 14

# Overview: ML Resource Usage

Arthi Padmanabhan

Mar 6, 2023

# Logistics

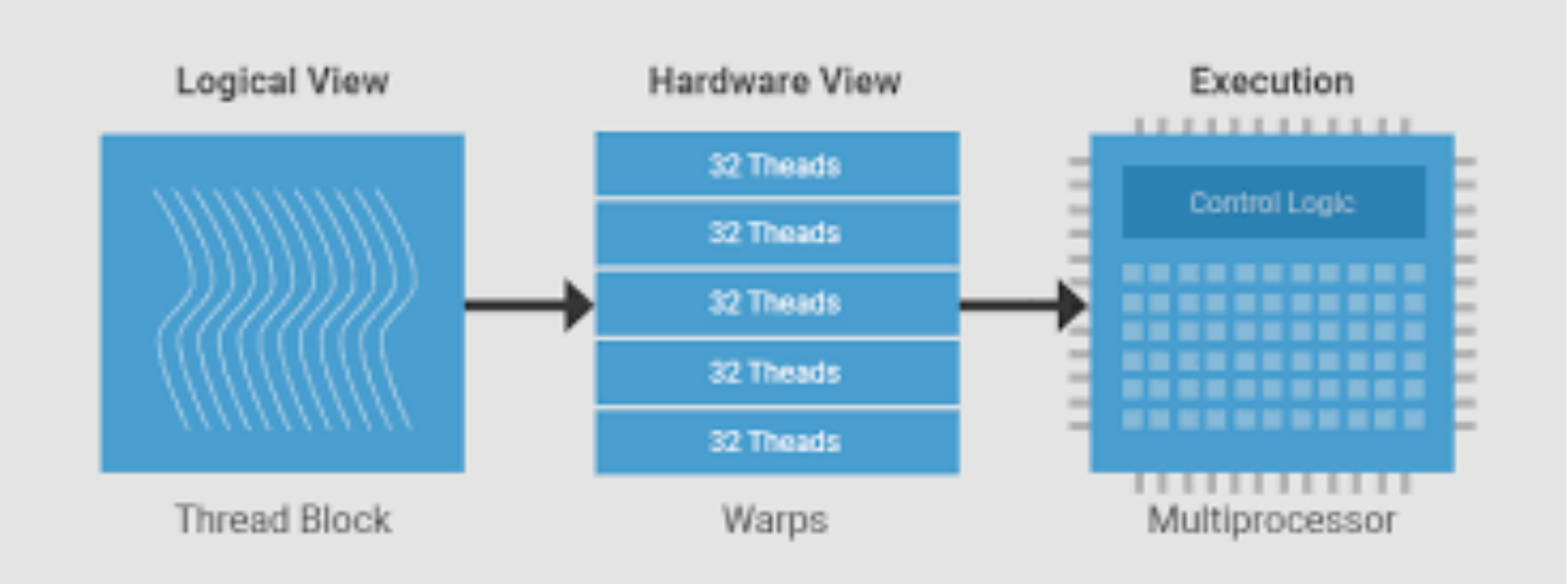
- Assignment 4 out today, due 3/24
- Project survey (~10 min) out Wednesday after class, due Thursday evening
- Week after spring break: one day will be working session for project proposal (due 3/27)
  - Problem you're trying to solve
  - Motivation
  - Plan: how are you dividing up work, what is your timeline for each part, what are your goals and reach goals, what do you hope to learn more about?

# Feedback Review

- Things going well:
  - Content itself
  - Assignments
  - Paper choice
  - Hands-on aspect of course
- Areas for Improvement:
  - Pace of course could be quicker
  - Spend too long on paper reading
  - Demos can be a little fast
  - Could post papers earlier so Groups A and B have more time to prepare

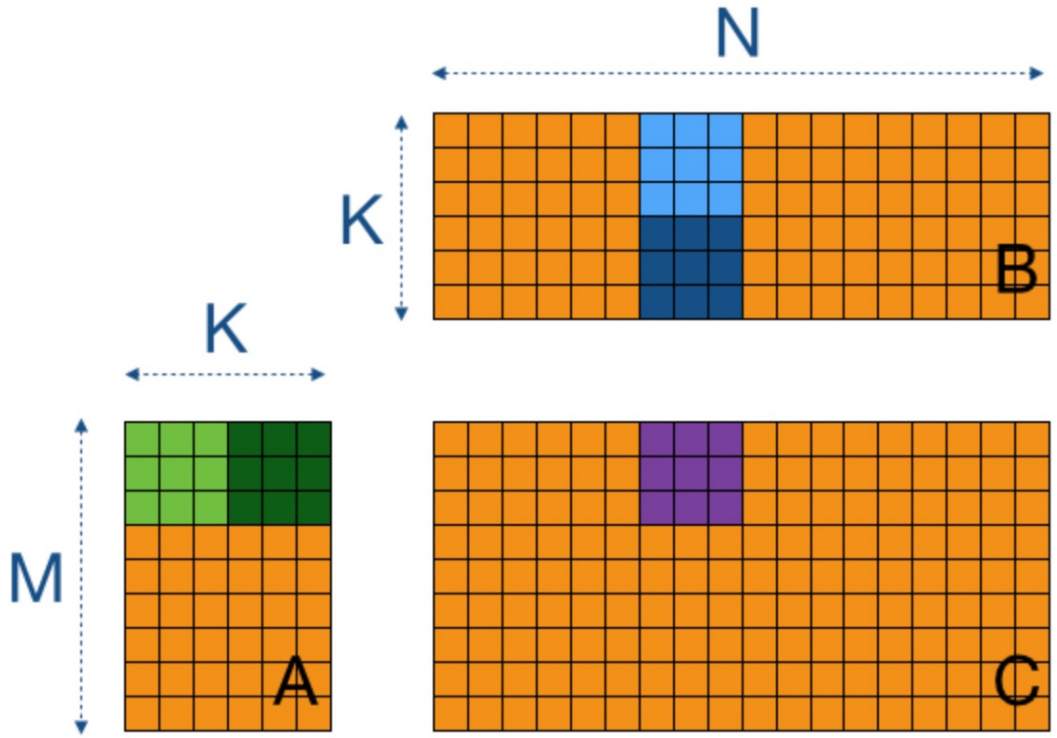
# Last Time

- GPU usage in details: threads, warps, blocks, grids, kernel



# Last Time

- GPU usage in details: threads, warps, blocks, grids, kernel
- Matrix block multiplication: matrix can be divided into blocks so that when we fetch an element from memory, it's used more than once



# Important Concepts from Last Time

- True or False: no matter how big your computation is, the GPU uses the same number of threads per block and blocks per grid

# Today

- What do we mean by “resources” used by running ML and why are people trying to lower resource usage?

# Resources

- Computation
- Memory
- Bandwidth
- Energy

# Computation: why is it important?

- Latency: often, we need to respond to queries very quickly

# Computation: why is it important?

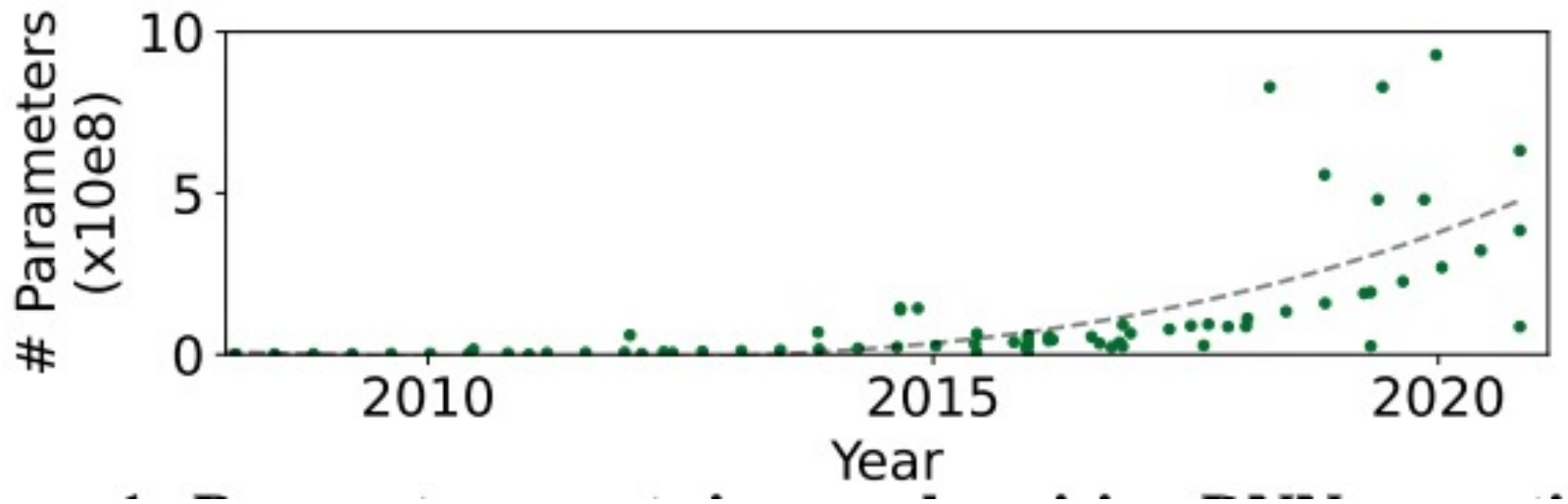
- Latency: often, we need to respond to queries very quickly
- Money -> GPUs are expensive!

| Amazon Web Services GPU Instances | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| | | GPU Performance, Teraflops | | | 3 Year | $ Per Teraflops | | |
| P3 | Tesla V100 | DP | SP | HP | RI Cost | DP | SP | HP |
| p3.2xlarge | 1 | 7.5 | 15.0 | 120.0 | $35,029 | $4,671 | $2,335 | $292 |
| p3.8xlarge | 4 | 30.0 | 60.0 | 480.0 | $140,089 | $4,670 | $2,335 | $292 |
| p3.16xlarge | 8 | 60.0 | 120.0 | 960.0 | $280,205 | $4,670 | $2,335 | $292 |
| | | | | | | | | |
| | | GPU Performance, Teraflops | | | 3 Year | $ Per Teraflops | | |
| P2 | Tesla GK210B | DP | SP | HP | RI Cost | DP | SP | HP |
| p2.xlarge | 1 | 1.5 | 4.4 | - | $12,071 | $8,296 | $2,765 | - |
| p2.8xlarge | 8 | 11.6 | 34.9 | - | $96,566 | $8,296 | $2,765 | - |
| p2.16xlarge | 16 | 23.3 | 69.8 | - | $193,133 | $8,296 | $2,765 | - |
| | | GPU Perf, Teraflops | | | Purchase | $ Per Teraflops | | |
| | Tesla GV100 | DP | SP | HP | Price | DP | SP | HP |
| Nvidia DGX-1V | 8 | 60.0 | 120.0 | 960.0 | $149,000 | $2,483 | $1,242 | $155 |
| | Tesla GP100 | DP | SP | HP | Price | DP | SP | HP |
| Nvidia DGX-1P | 8 | 42.5 | 84.8 | 169.6 | $129,000 | $3,035 | $1,521 | $761 |

# Training vs Inference

- Whether the time is an issue depends heavily on the workload
  - Often, we'd like to run several models on one GPU -> depends on the models' times to run (can we batch multiple images?)
  - Let's time different models, classifiers and object detectors (first on CPU, then make necessary changes to run on GPU). Then, try
    - Different ResNets, VGG16, VGG19, AlexNet
    - SSD, Faster RCNN (`torchvision.models.detection.fasterrcnn_resnet50_fpn`)
    - If you're done early, try fetching two images and running with a batch size of 2 (remove unsqueeze(0) and instead create a 4D tensor with 2 as the first dimension - see torch documentation)
  - What trends do you see across different models and sizes?

# Model Size is Increasing



# Training vs Inference

- Training takes much longer – constantly retraining models with new info is often what makes large company's workloads so expensive
- Training time depends on:

# Training vs Inference

- Training takes much longer – constantly retraining models with new info is often what makes large company's workloads so expensive
- Training time depends on:
  - model architecture
  - size of dataset
  - image resolution
  - grayscale vs not
  - accuracy you're trying to achieve (which affects number of epochs)
  - batch size
  - available resources (memory!)

# Example: ChatGPT

- ChatGPT (175 billion parameters)
  - How long would it take to train on a single good GPU?

# Example: ChatGPT

- ChatGPT (175 billion parameters)
  - How long would it take to train on a single good GPU?
    - 355 years
    - OpenAI was launched in 2020
  - Instead they used 1024 GPUs and it took months

# Training vs Inference

- Training takes much longer – constantly retraining models with new info is often what makes large company's workloads so expensive
- Training time depends on:
  - model architecture
  - size of dataset
  - image resolution
  - grayscale vs not
  - accuracy you're trying to achieve (which affects number of epochs)
  - **batch size**
  - available resources (memory!)

# Batch Size

- Why can’t I just keep increasing batch size (train on more and more images at once)?
- GPU Memory!

# GPU Memory

- This is often a major bottleneck, both for training faster and for being able to run inference on several models on one GPU
- Even if they're not running at the same time, this is hard. Why? They take memory just to sit in GPU, so unless you want to move them back and forth between CPU and GPU (which takes a long time), they are using memory

# Memory and Time to Load in GPU and Run

- Memory in GB, Time in ms

| Model | Load Memory (Time) | Run Memory (Time) | | |
|---|---|---|---|---|
| | | BS=1 | BS=2 | BS=4 |
| YOLOv3 | 0.24 (49.5) | 0.52 (17.0) | 0.73 (24.0) | 1.22 (39.9) |
| ResNet152 | 0.24 (73.3) | 0.65 (24.8) | 0.98 (26.3) | 1.71 (26.7) |
| ResNet50 | 0.12 (27.1) | 0.35 (8.4) | 0.50 (8.5) | 0.84 (8.5) |
| VGG16 | 0.54 (72.2) | 0.74 (2.1) | 0.89 (2.4) | 1.18 (2.4) |
| Tiny YOLOv3 | 0.04 (6.7) | 0.15 (3.0) | 0.18 (5.2) | 0.24 (5.2) |
| Faster RCNN | 0.73 (117.3) | 3.70 (115.4) | 6.96 (210.1) | 12.47 (379.4) |
| Inceptionv3 | 0.12 (11.8) | 0.19 (9.1) | 0.23 (9.1) | 0.34 (9.1) |
| SSD-VGG | 0.11 (16.1) | 0.23 (16.5) | 0.33 (25.7) | 0.51 (44.6) |

# Let’s look at memory numbers

- First, there is some memory needed to start CUDA
- To assess memory, we can probe nvidia-smi
- To assess memory over time, we can write a script that saves to a file
- Colab -> script is there for you; remember to restart runtime for a new model

# Next Time

- Go over ideas for final project