Machine Learning Systems Design

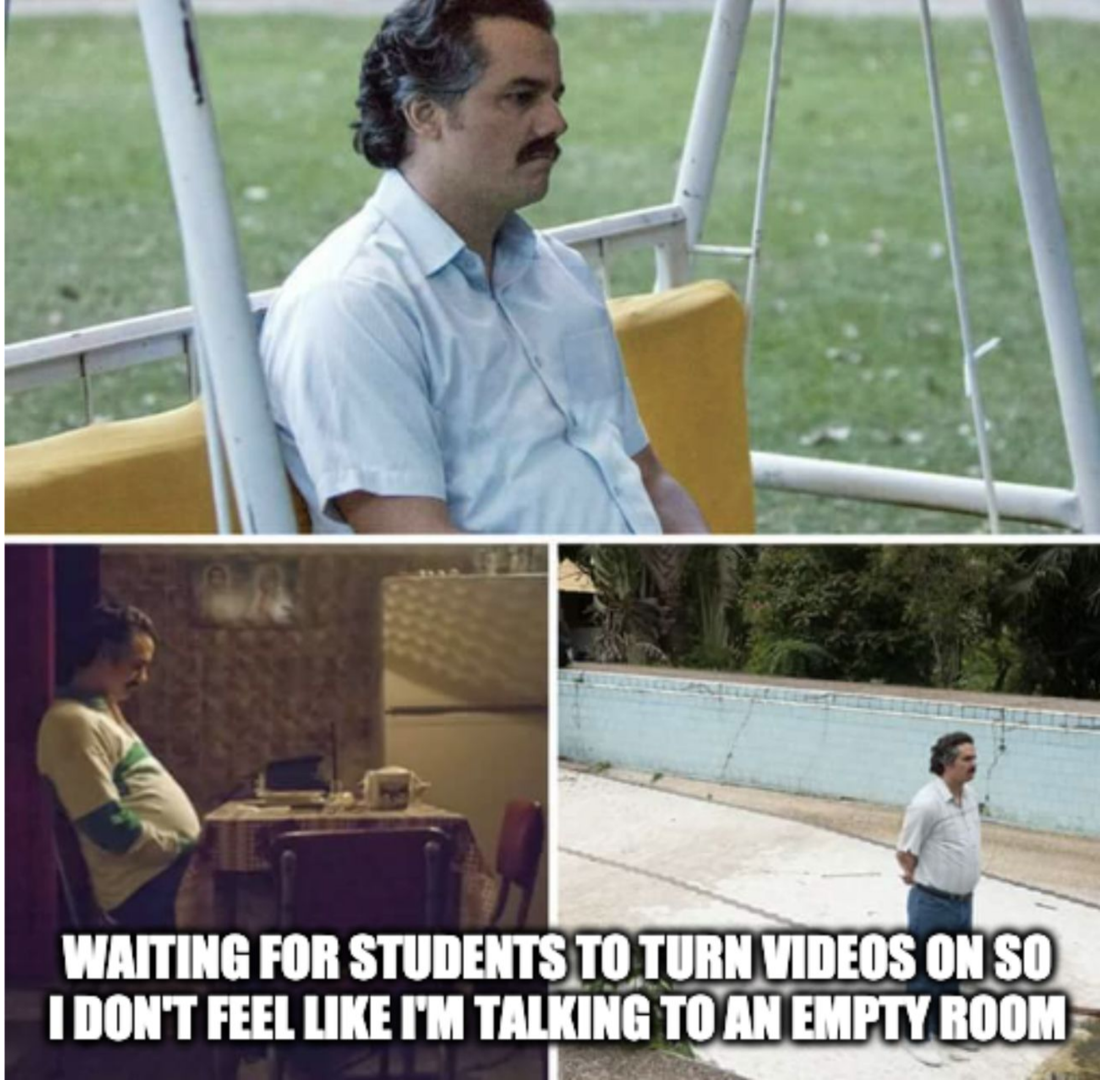
Lecture 6: Model Offline Evaluation

Stanford University

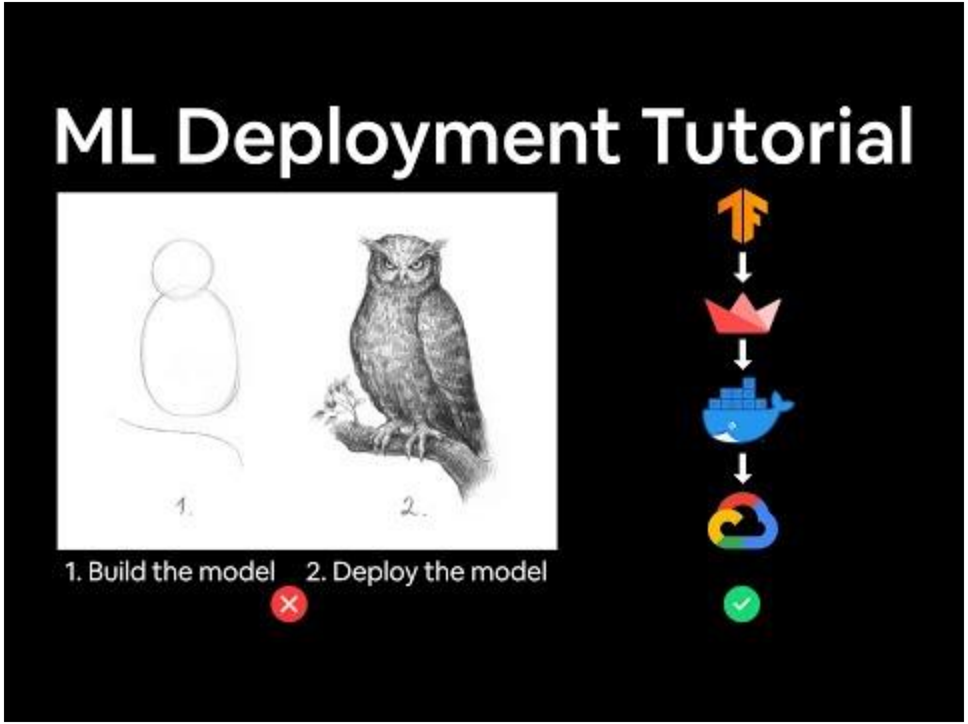
Zoom etiquettes

We appreciate it
if you keep videos on!

- More visual feedback for us
  to adjust materials
- Better learning environment
- Better sense of who you’re with
  in class!



Deploying on Google Cloud Tutorials



Agenda

1. Distributed training
2. Breakout exercise
3. Model offline evaluation

Lecture note is on course website / syllabus

Distributed training

Chloe He

Ways a model can scale

1. In complexity: architecture, number of parameters

Ways a model can scale

1. In complexity: architecture, number of parameters
2. In prediction traffic

Ways a model can scale

1. In complexity: architecture, number of parameters
2. In prediction traffic
3. In number of models

Rise of Incredibly Large DL Models

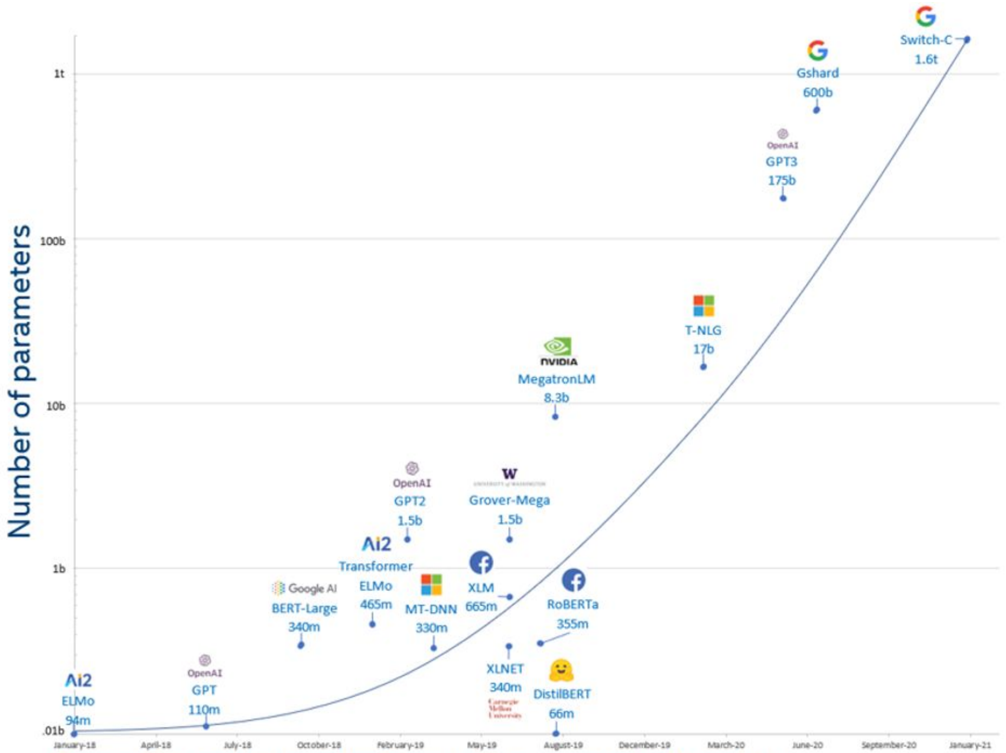
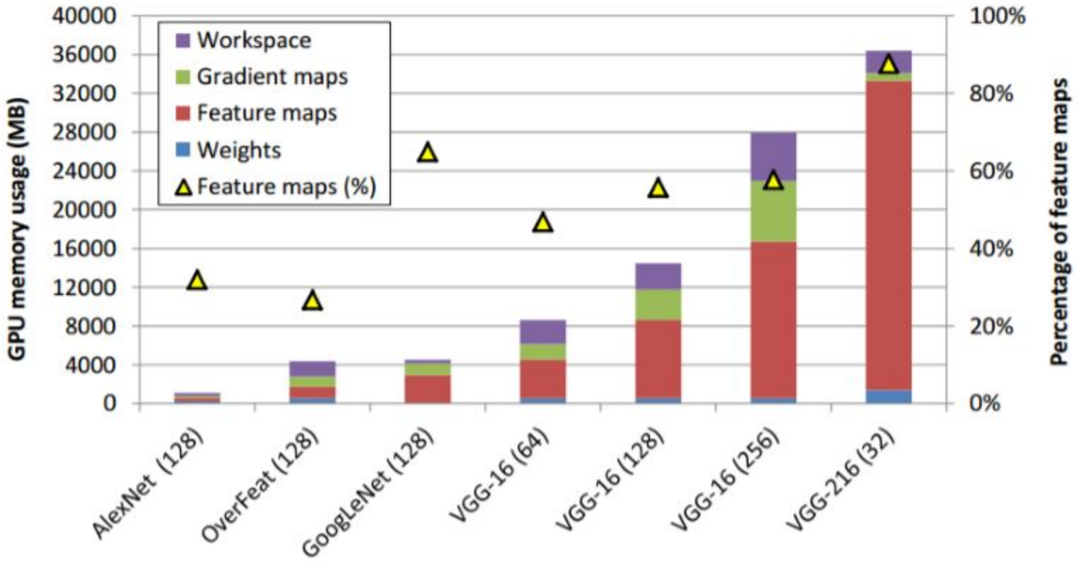


Figure 1: Exponential growth of number of parameters in DL models

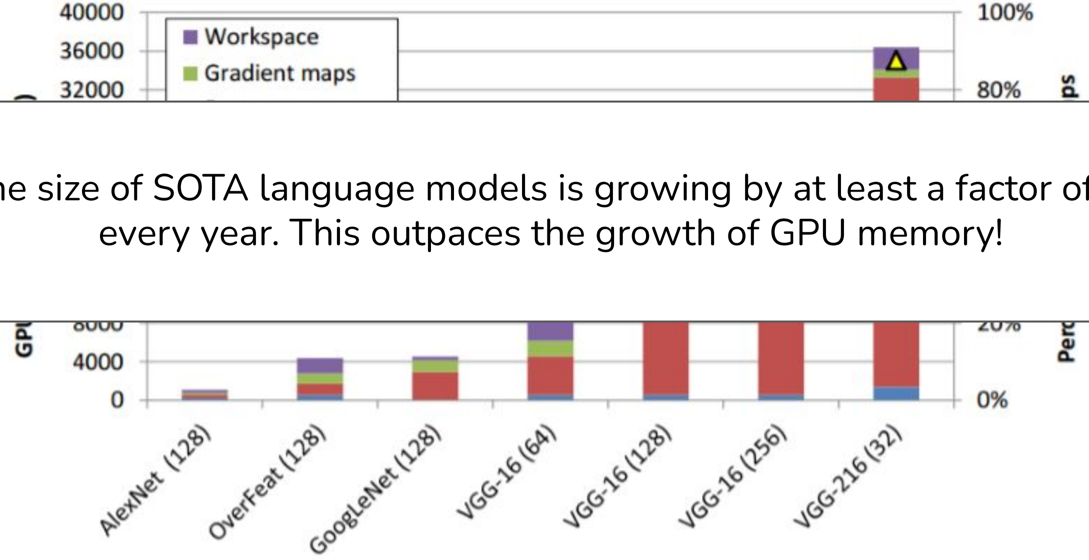
Illustration by [Towards Data Science]()

GPU Usage



Rhu, M., et al, [vDNN: Virtualized Deep Neural Networks for Scalable, Memory-Efficient Neural Network Design]

GPU Usage



The size of SOTA language models is growing by at least a factor of 10 every year. This outpaces the growth of GPU memory!

Rhu, M., et al, [vDNN: Virtualized Deep Neural Networks for Scalable, Memory-Efficient Neural Network Design]

Issues

- A smaller batch size can lead to
  - More iterations necessary to converge
  - Decreased stability

-> What about when the model itself doesn’t fit into GPU memory? Or when even a single data sample doesn’t fit into GPU memory?

1 https://developer.nvidia.com/gtc/2020/video/s21496-vid
2 https://lambdalabs.com/blog/demystifying-gpt-3/

Distributed Training

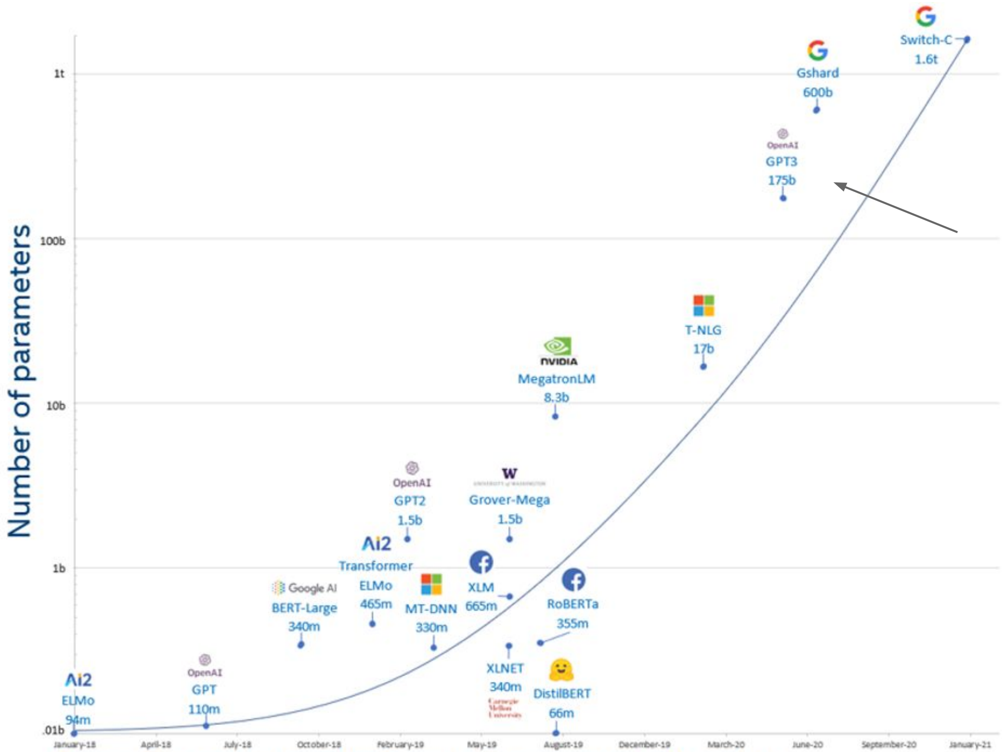


Figure 1: Exponential growth of number of parameters in DL models

700GB memory to store the parameters; 355 GPU-years and $4.6M for a single training run on NVIDIA V100 GPUs[2]

Illustration by Towards Data Science

1 https://developer.nvidia.com/gtc/2020/video/s21496-vid
2 https://lambdalabs.com/blog/demystifying-gpt-3/

Distributed Training

8-way model parallelism, 64-way data parallelism on 512 GPUs[1]

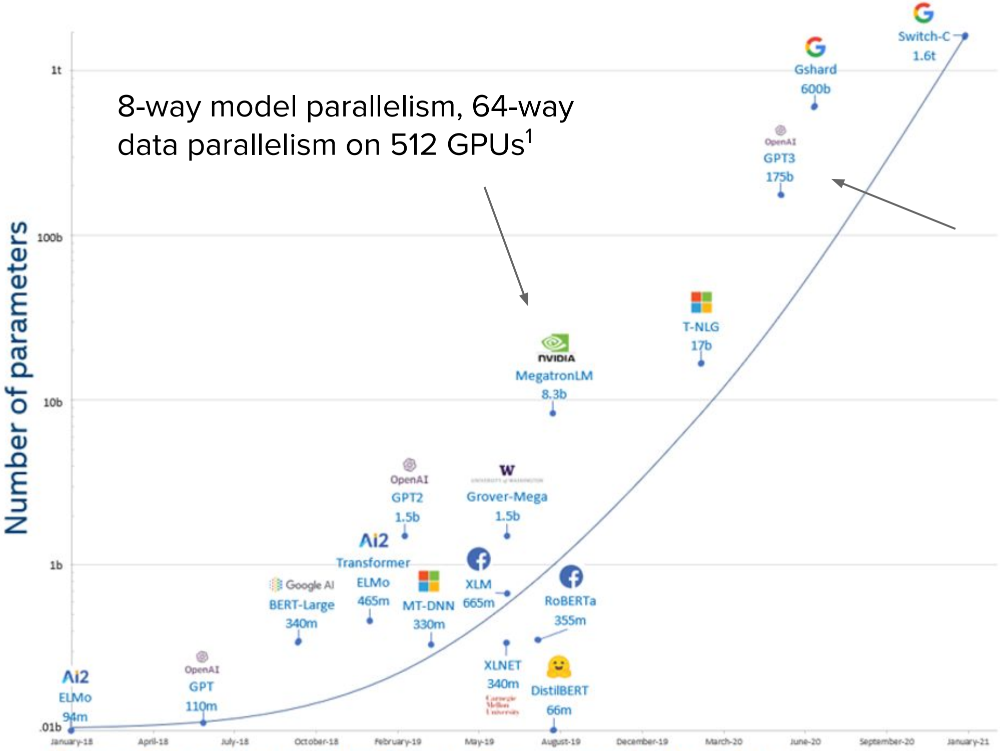


Figure 1: Exponential growth of number of parameters in DL models

700GB memory to store the parameters; 355 GPU-years and $4.6M for a single training run on NVIDIA V100 GPUs[2]

Illustration by Towards Data Science

Distributed Training

Data parallelism

Model parallelism

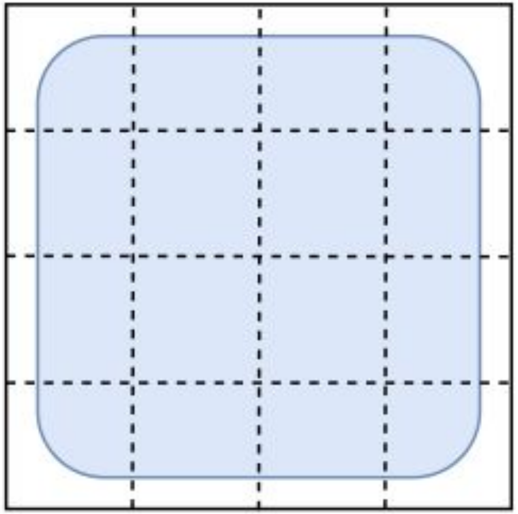
Data Parallelism for Large Batch Training

Split the data across devices

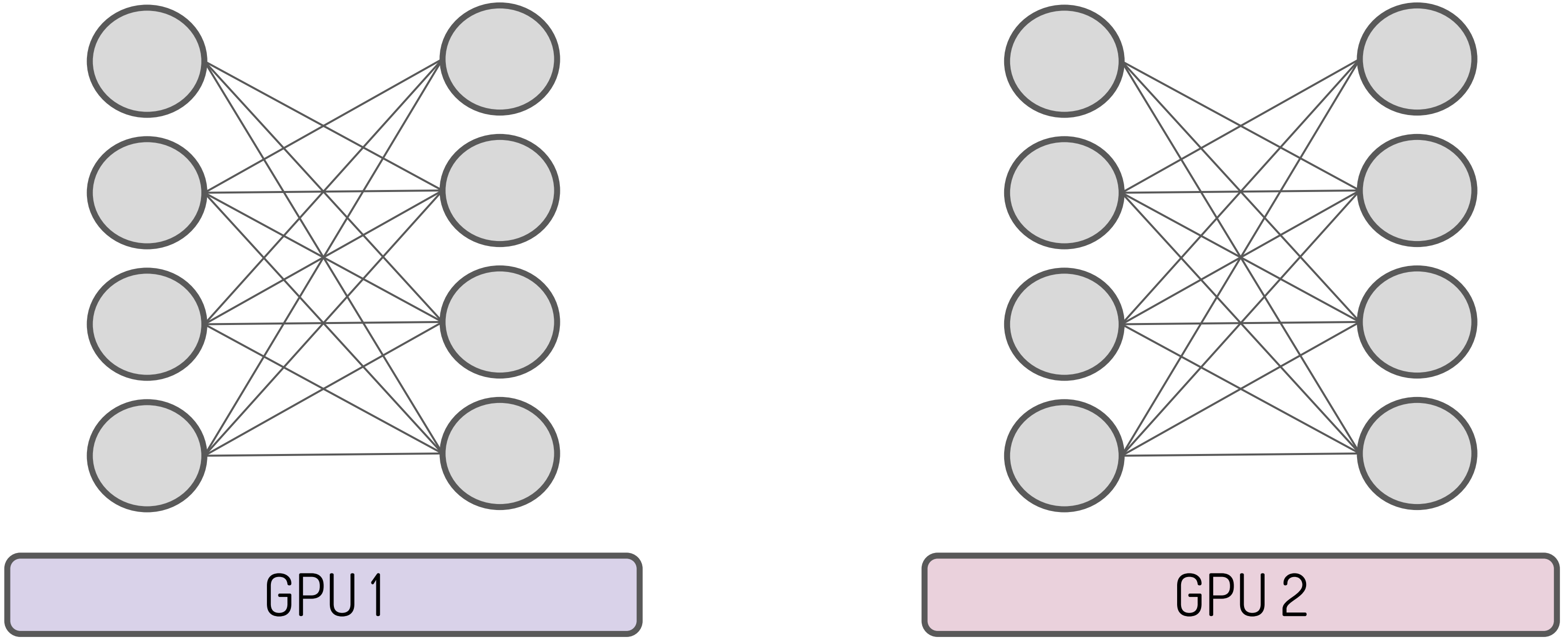
Each device sees a fraction of the batch

Each device replicates the model

Each device replicates the optimizer

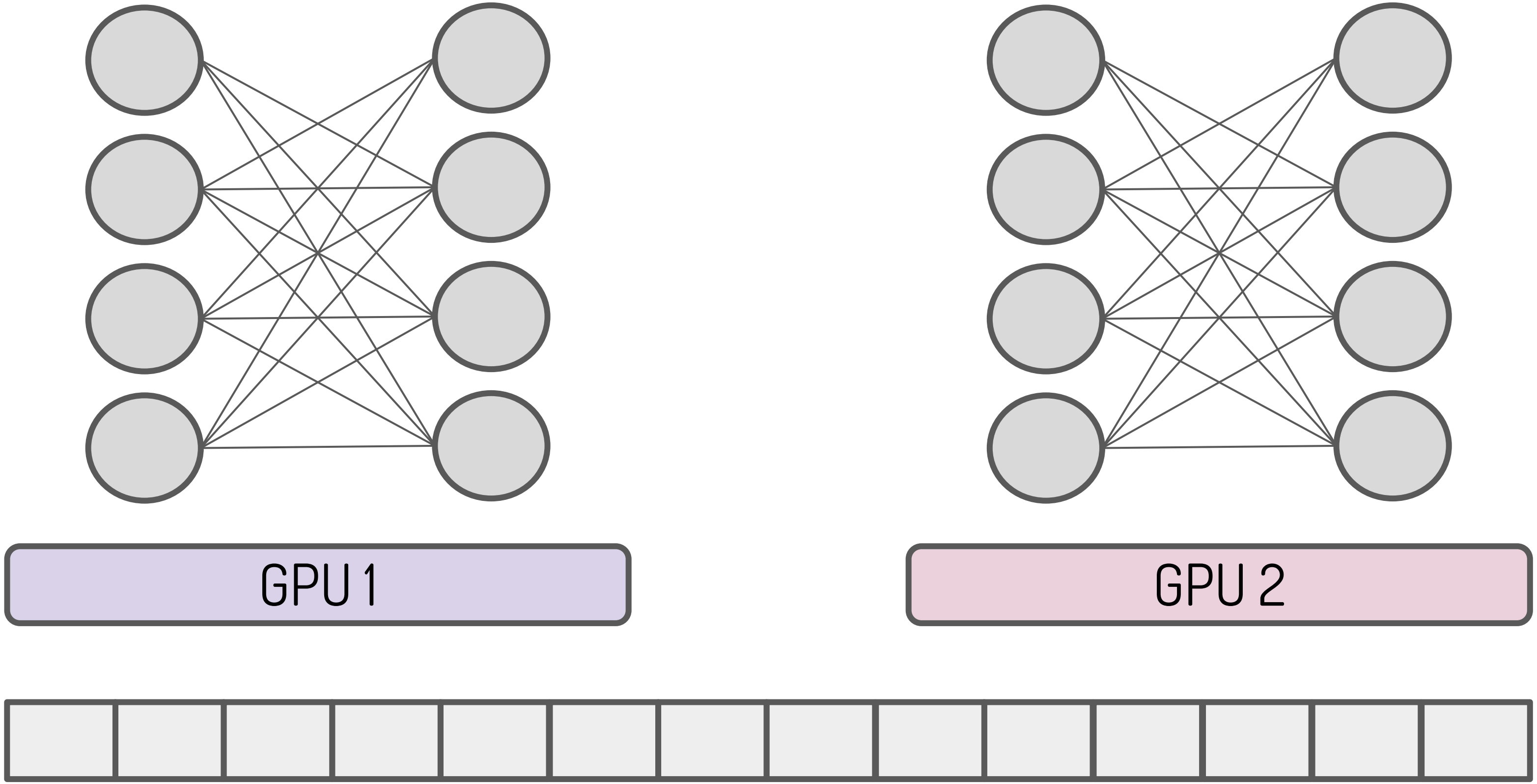


Replicate model across devices

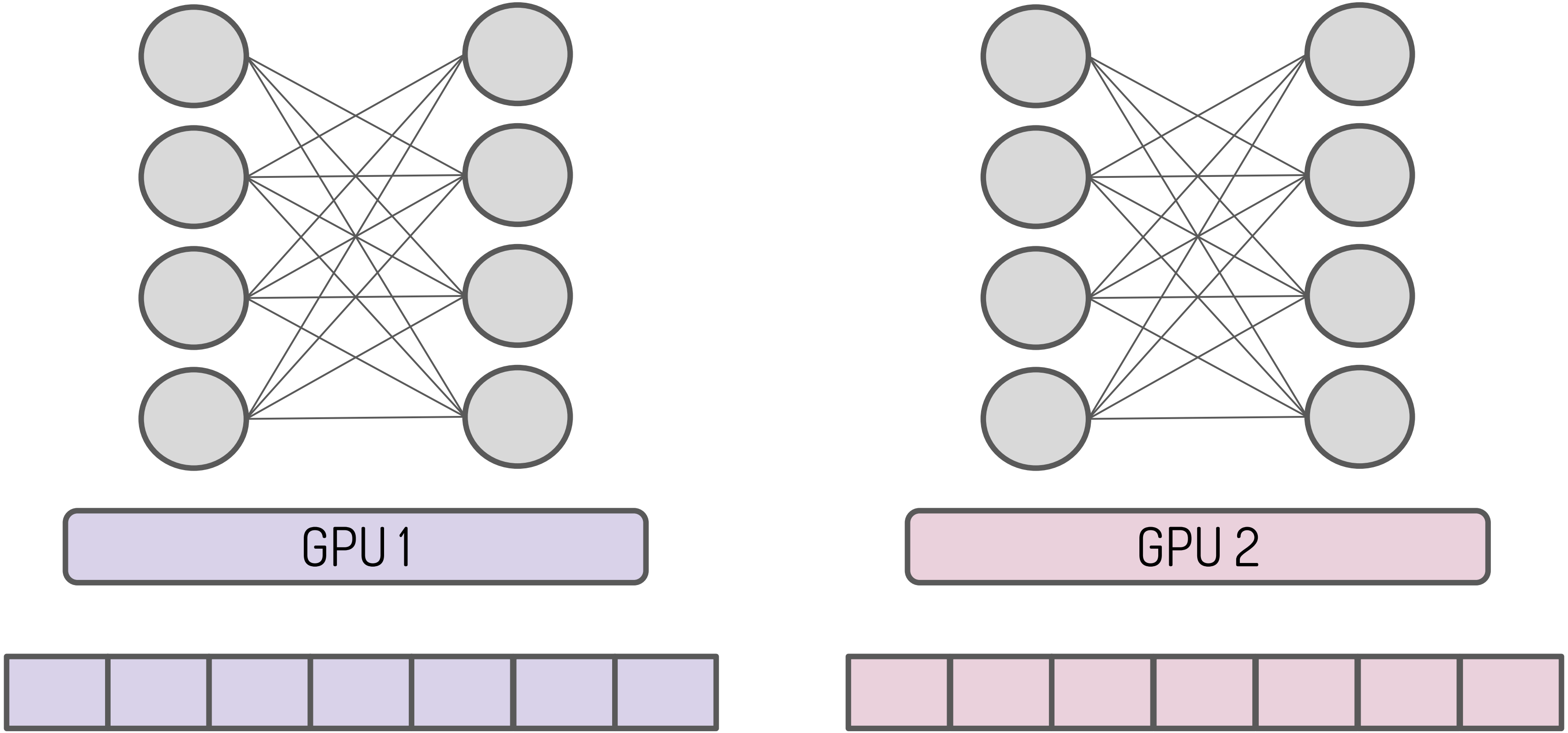


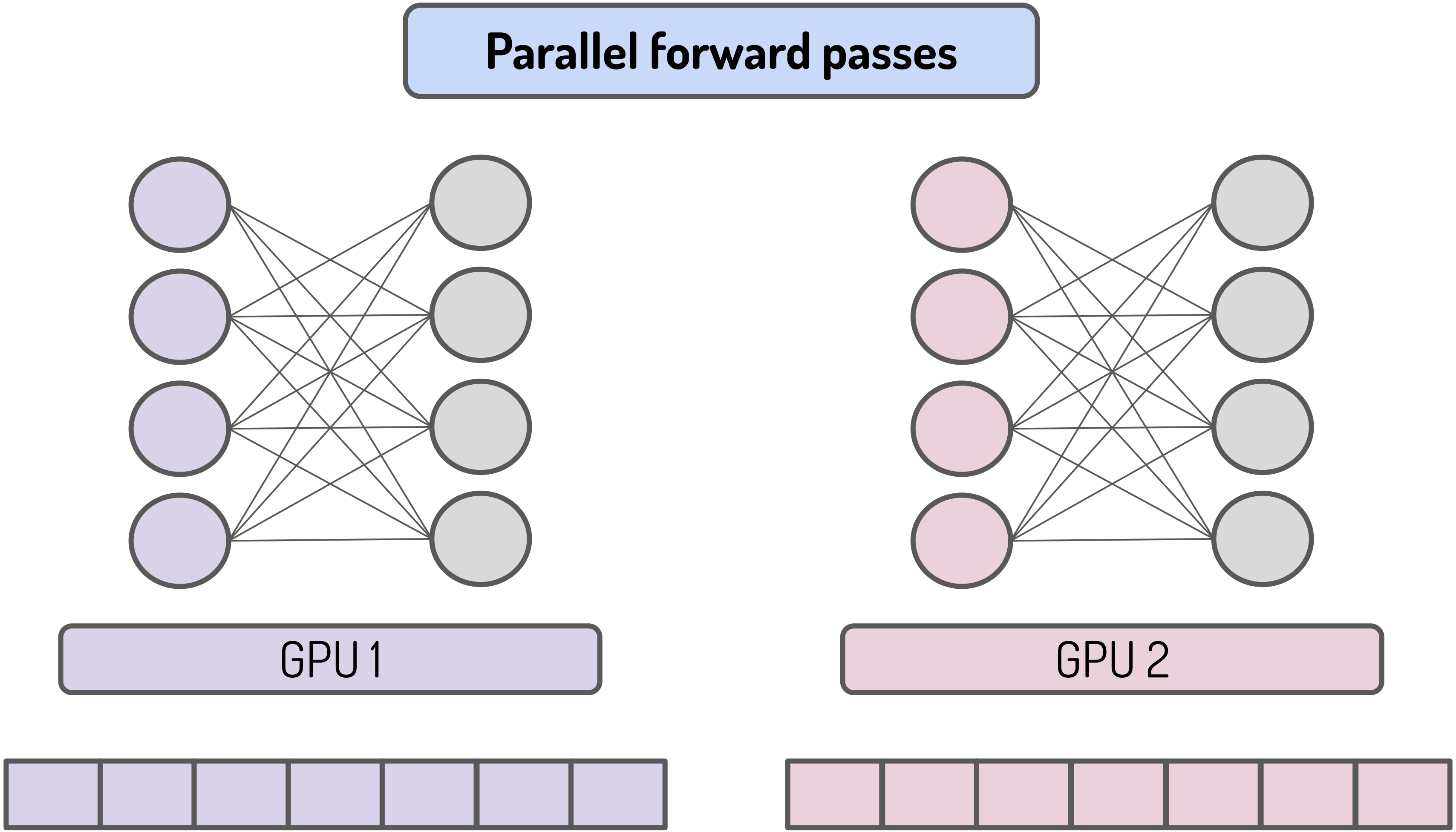
GPUs could be on same or multiple nodes

To push in a batch of data

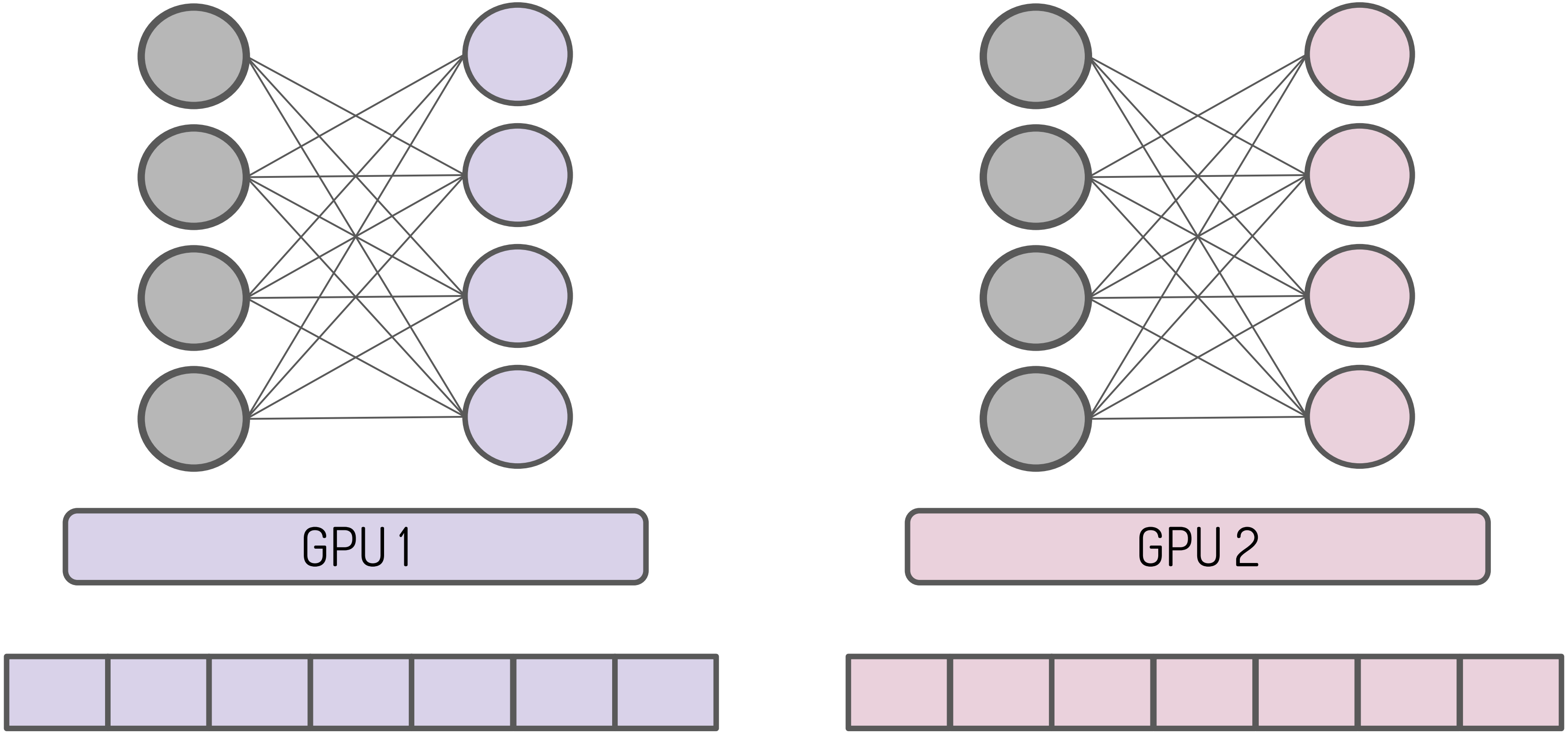


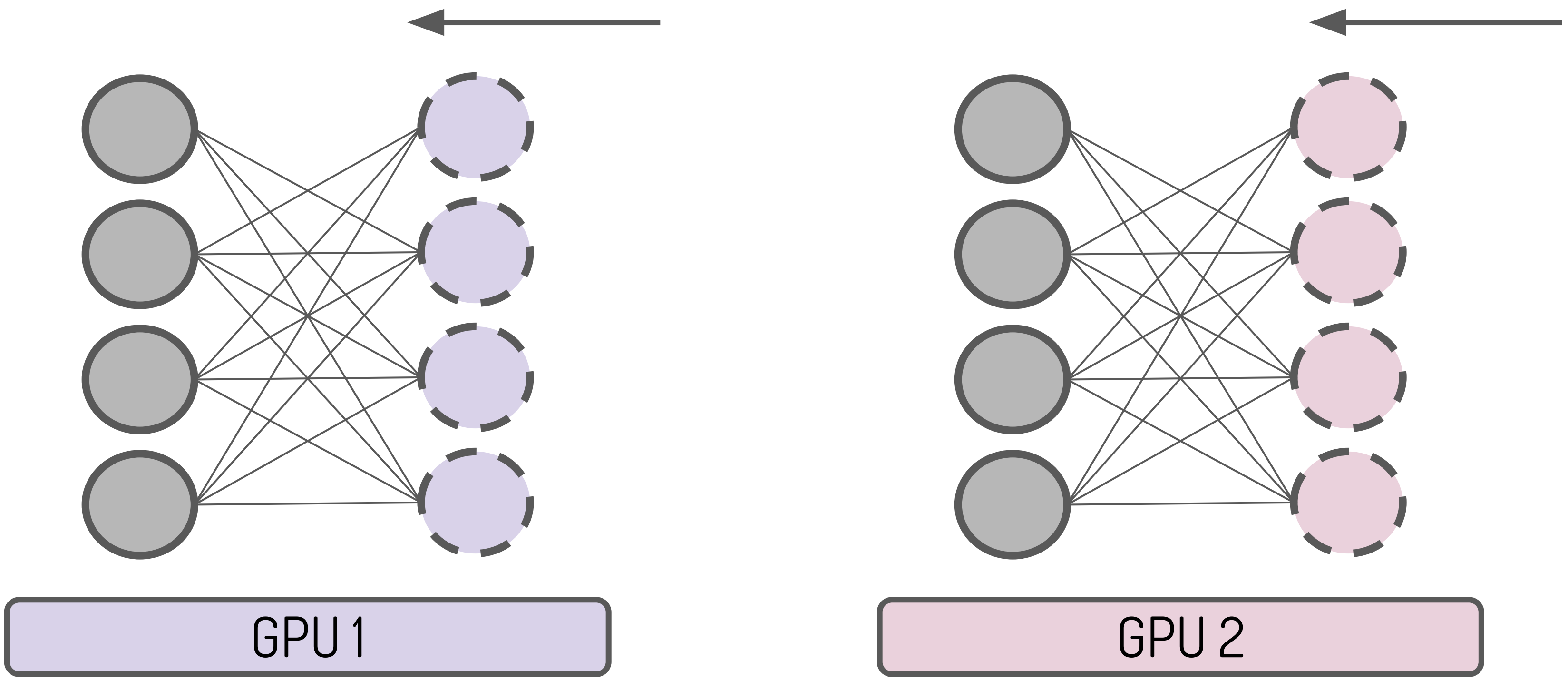
Split batch across devices



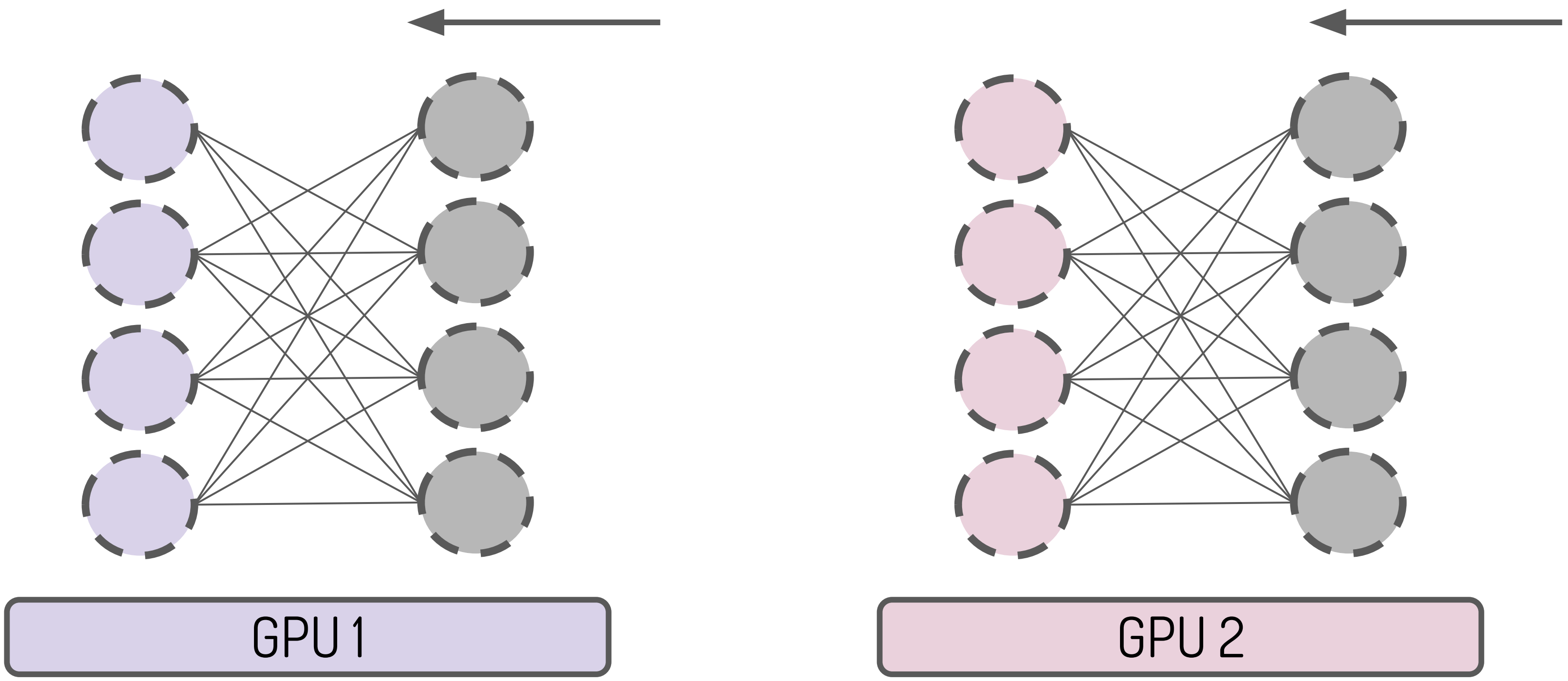

Parallel forward passes
GPU 1
GPU 2

Parallel forward passes

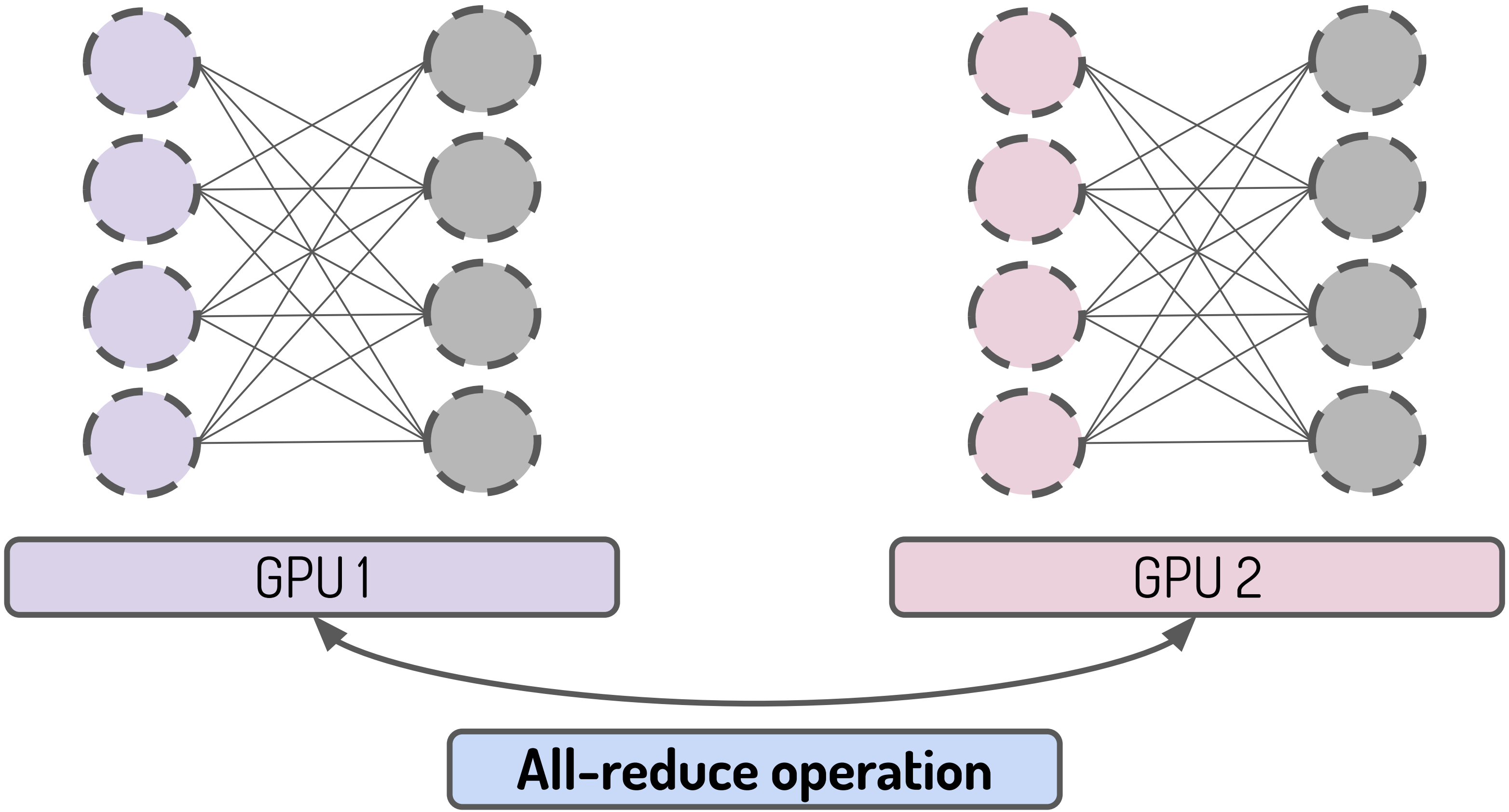


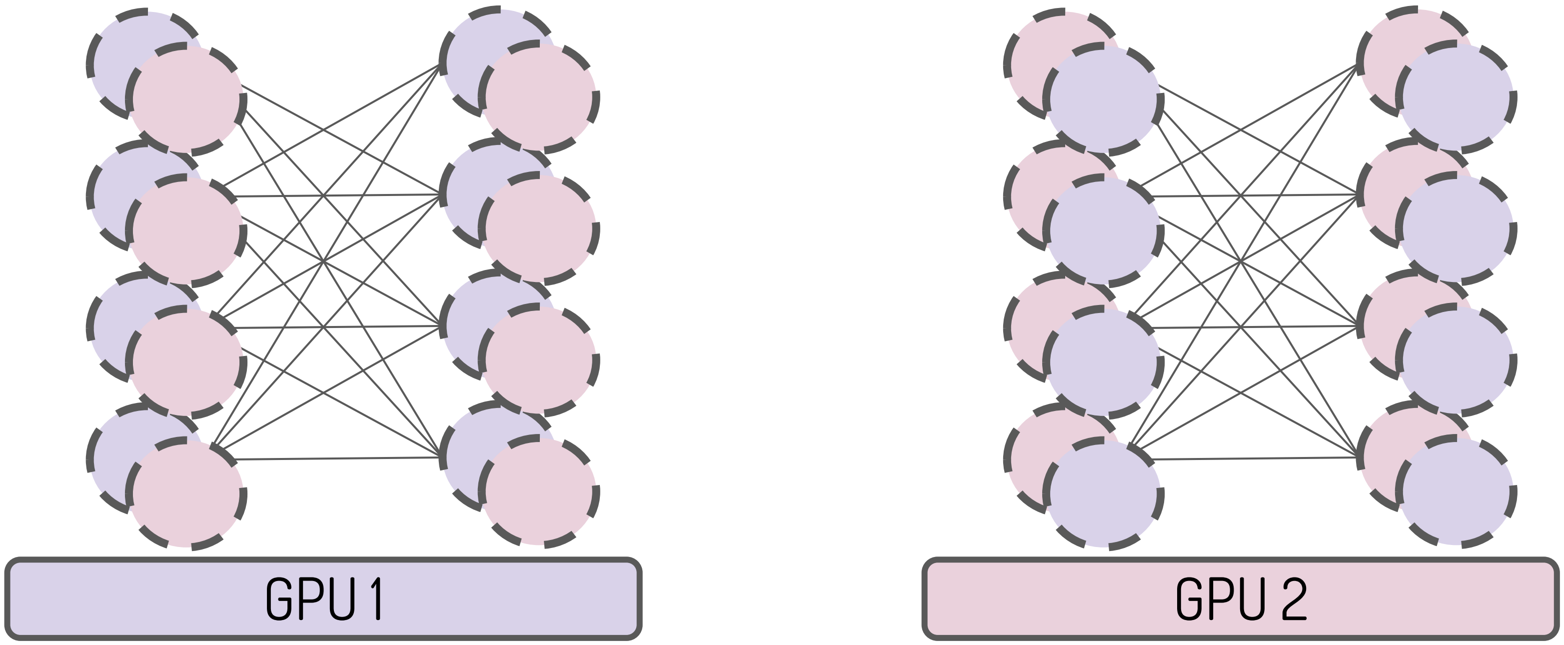


Backpropagate gradients

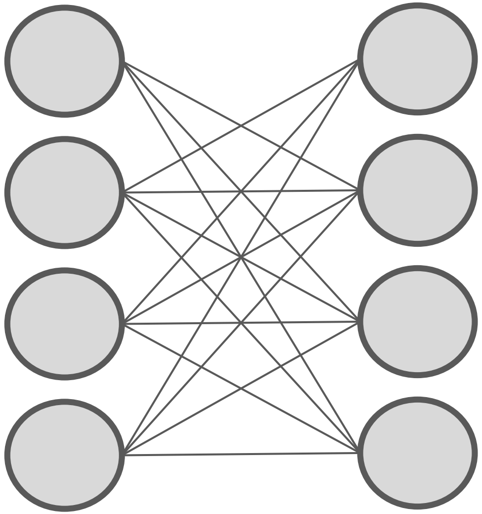


Backpropagate gradients

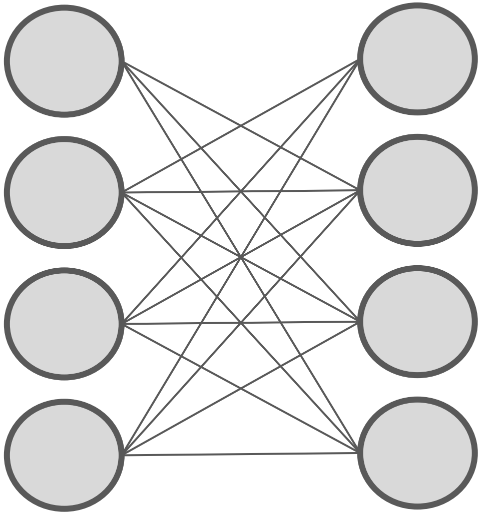

GPU 1
GPU 2
All-reduce operation



All devices do the same gradient updates



GPU 1



GPU 2

All parameters stay synchronized!

Data Parallelism

Split the data across devices

Each device sees a fraction of the batch

Each device replicates the model

Each device replicates the optimizer

GPT-3: 3.2M batch size

1M samples

- 1000 samples/batch/machine
- 1 machine: 1000 batches
- 100 machines: **10 batches**

Scaling SGD Batch Size to 32K for ImageNet Training (You et al., 2017)

Data Parallelism

Split the data across devices

Each device sees a fraction of the batch

Each device replicates the model

Each device replicates the optimizer

GPT-3: 3.2M batch size

Challenge 1: Learning rate

- Too small -> too long to converge
- Too large -> unstable learning

Scaling SGD Batch Size to 32K for ImageNet Training (You et al., 2017)

Data Parallelism: LR Scaling

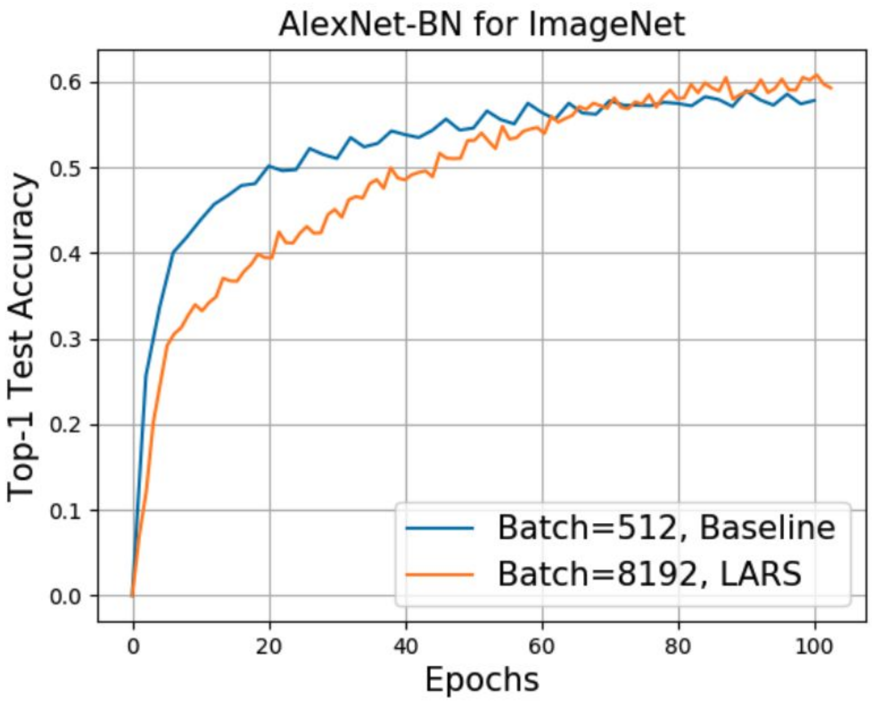
Split the data across devices

Each device sees a fraction of the batch

Each device replicates the model

Each device replicates the optimizer

GPT-3: 3.2M batch size



Scaling SGD Batch Size to 32K for ImageNet Training (You et al., 2017)

Data Parallelism: Gradient Updates

Challenge 2: How to aggregate gradient updates?

- Synchronous: have to wait for stragglers
- Asynch: gradients become stale

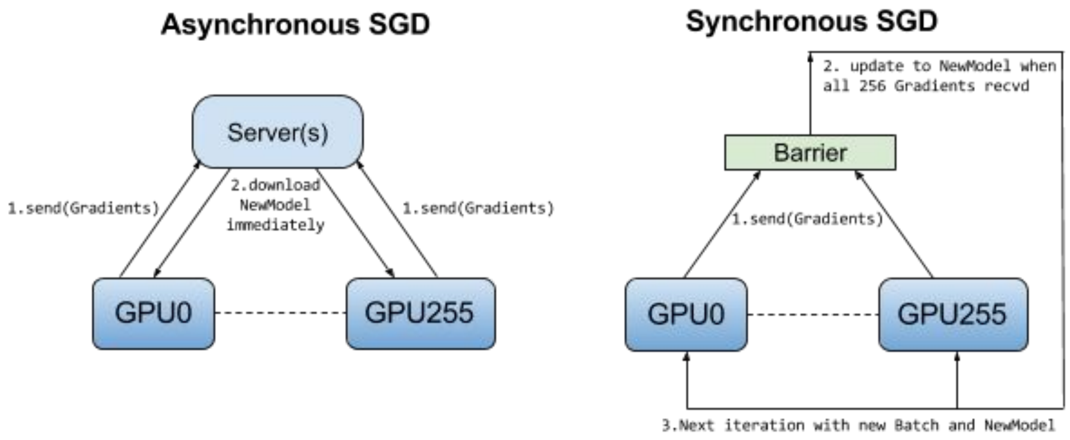
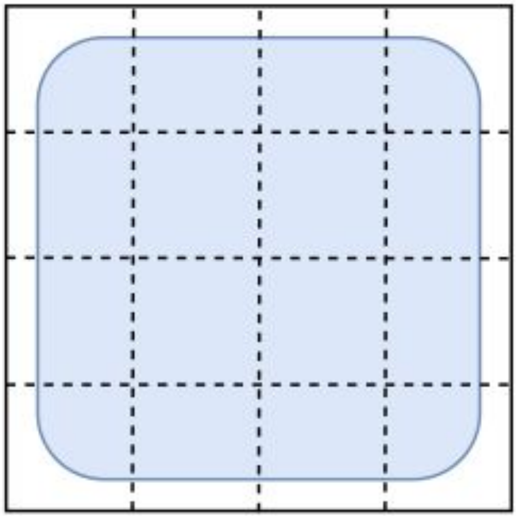


Image from Distributed TensorFlow (Jim Dowling, O'Reilly 2017)

Solution: Model Parallelism for Large Model Training

Split the model across devices

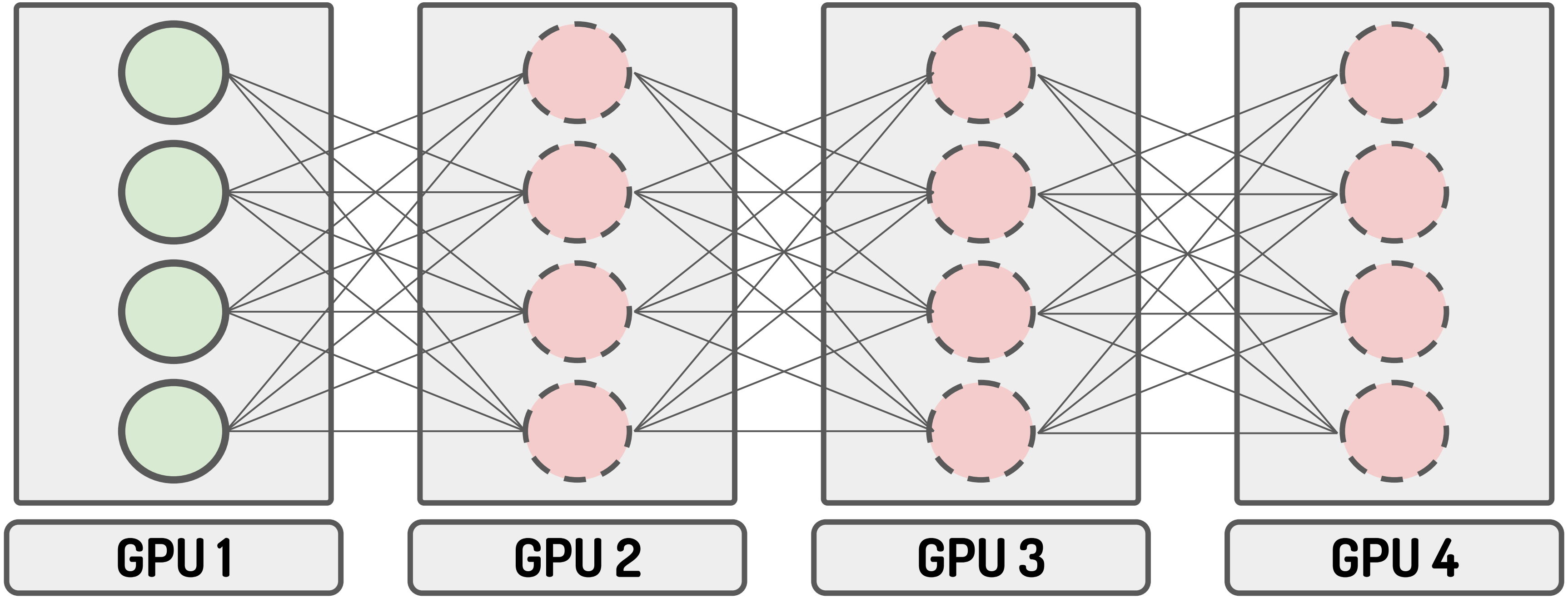
Each device runs a fragment of the model



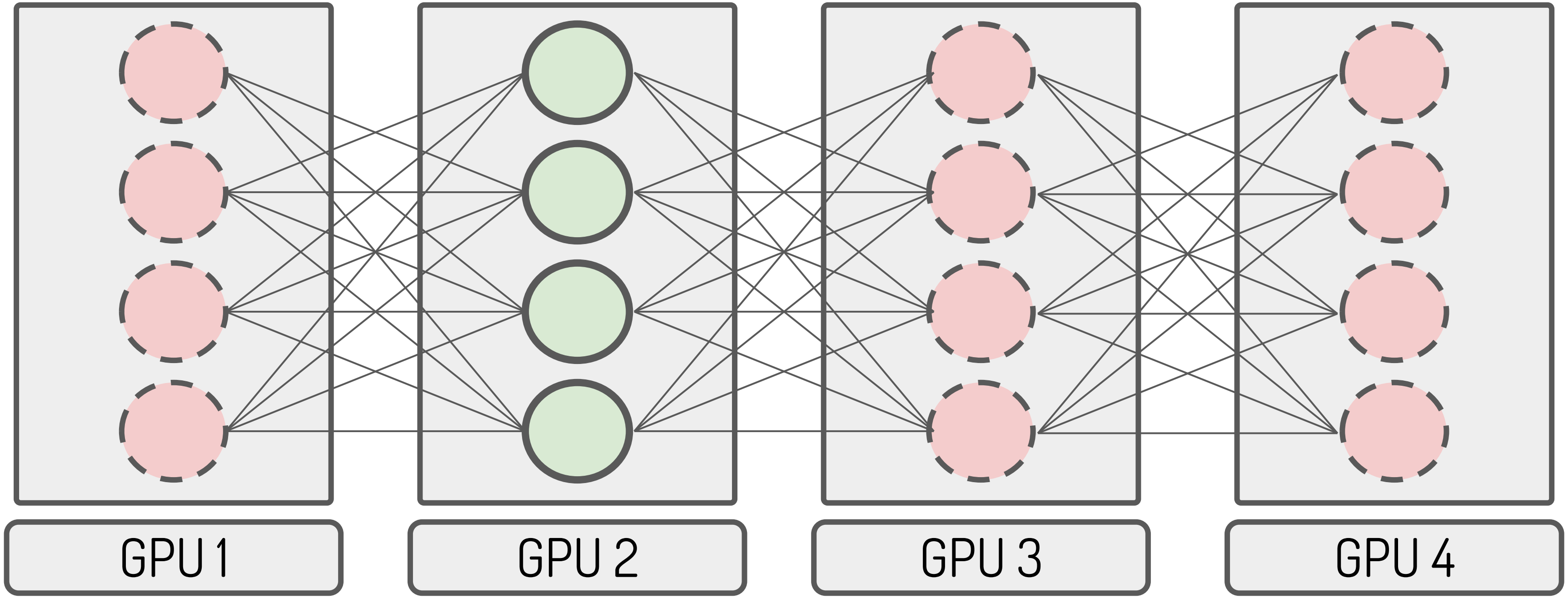
Credit: Fedus et al. (Switch Transformers: Scaling to Trillion Parameter Models with Simple and Efficient Sparsity

Model Parallelism: Naive

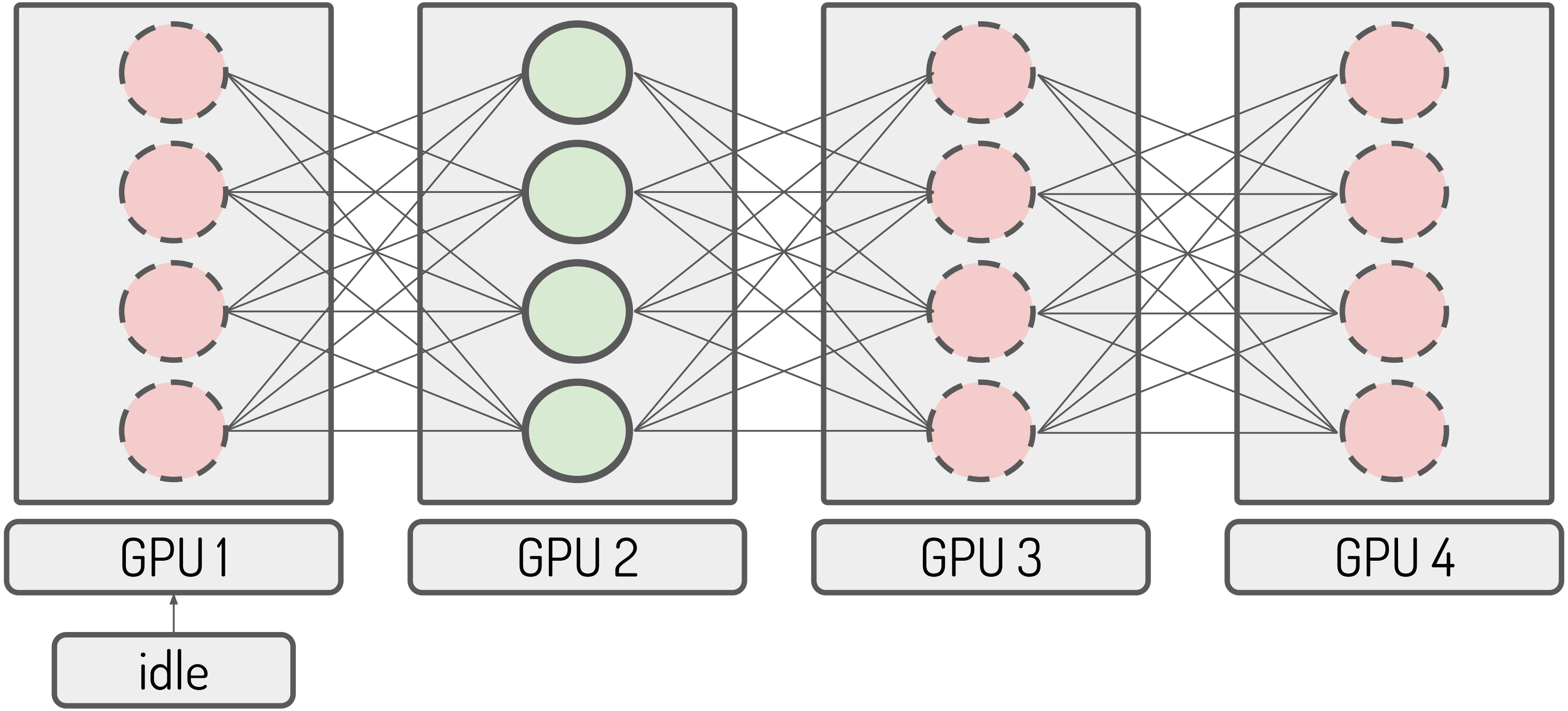
Model Parallelism: Naive



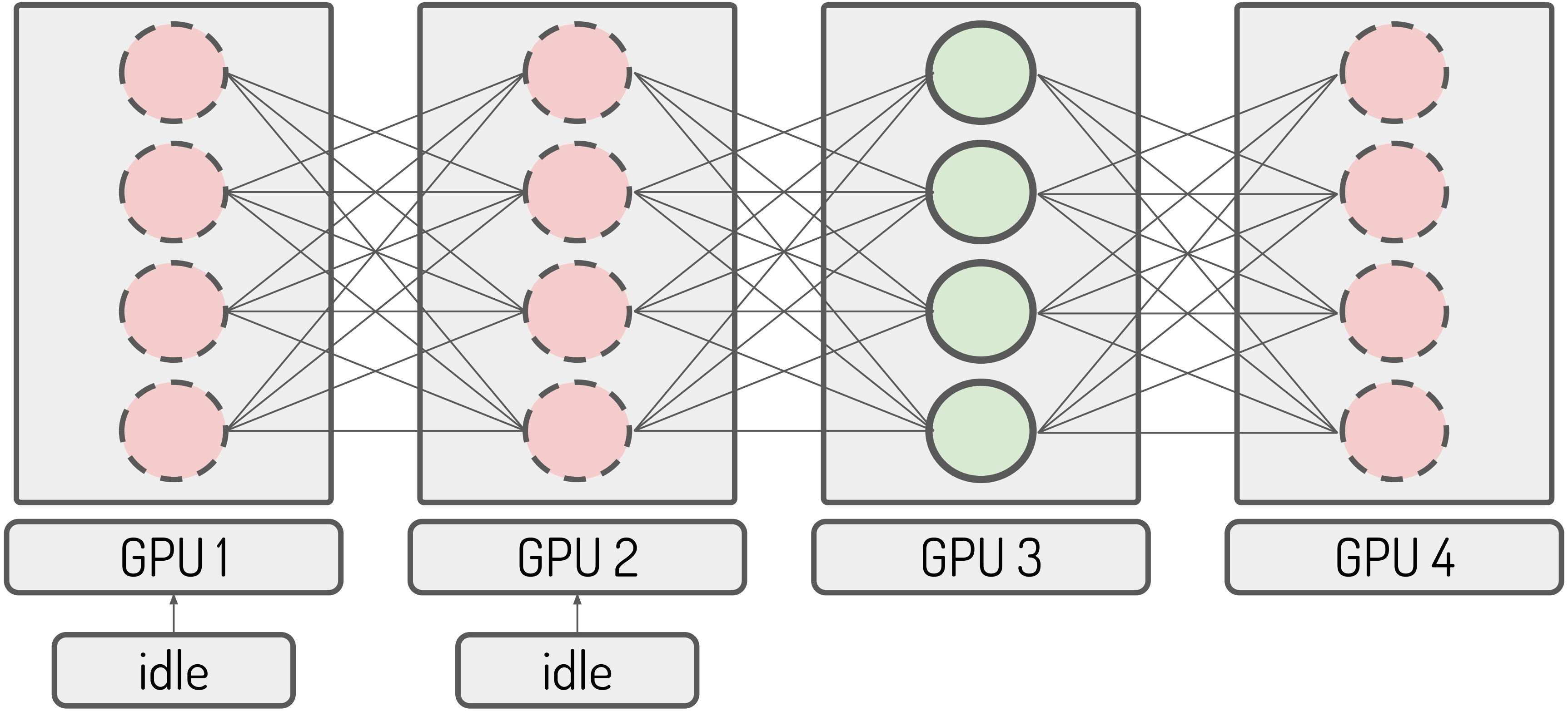
Model Parallelism: Naive



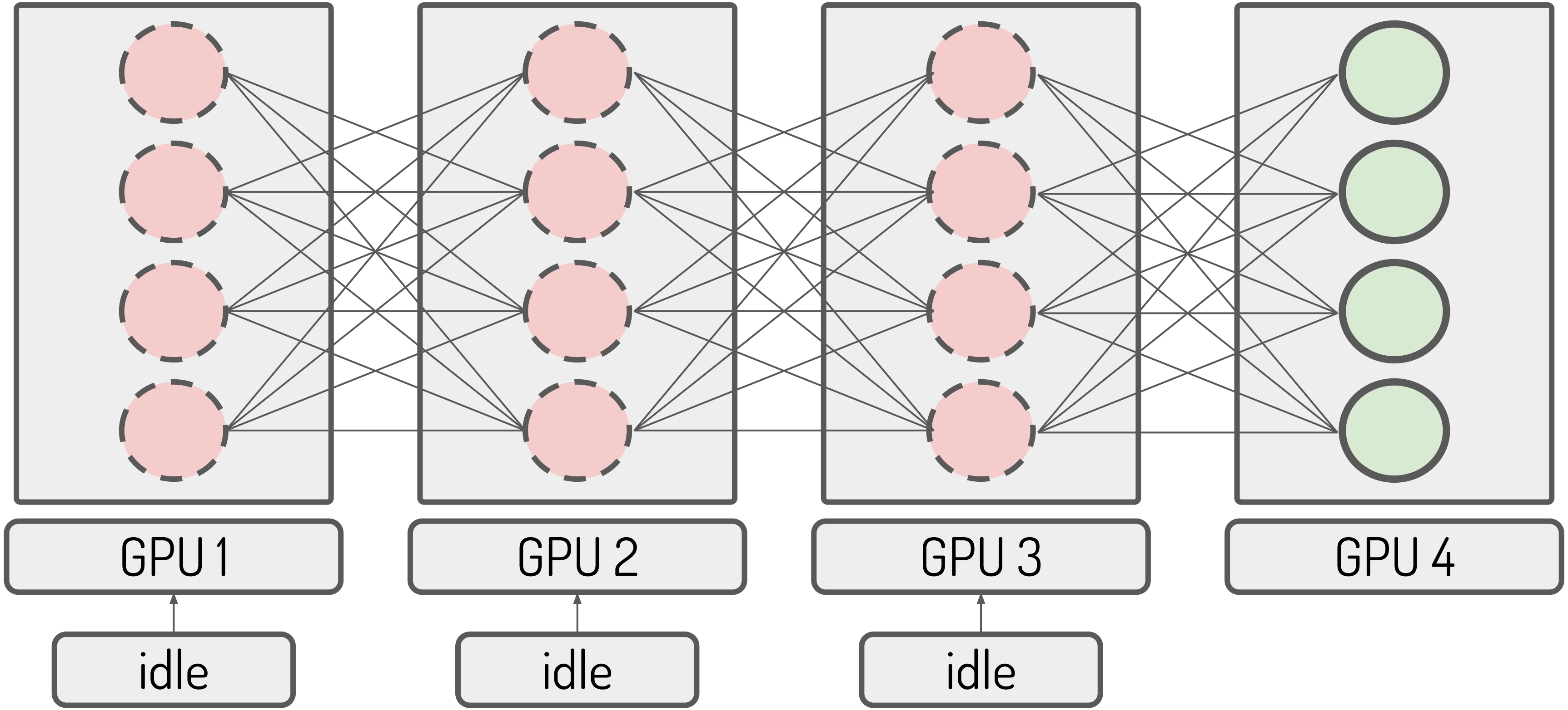
Model Parallelism: Naive



Model Parallelism: Naive

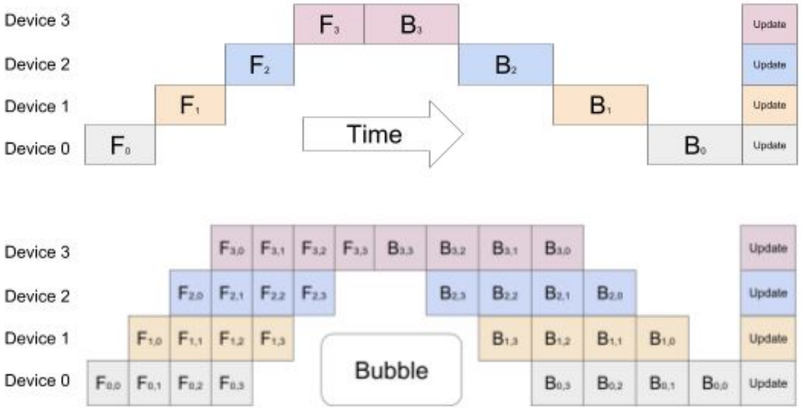


Model Parallelism: Naive



Pipeline Parallelism

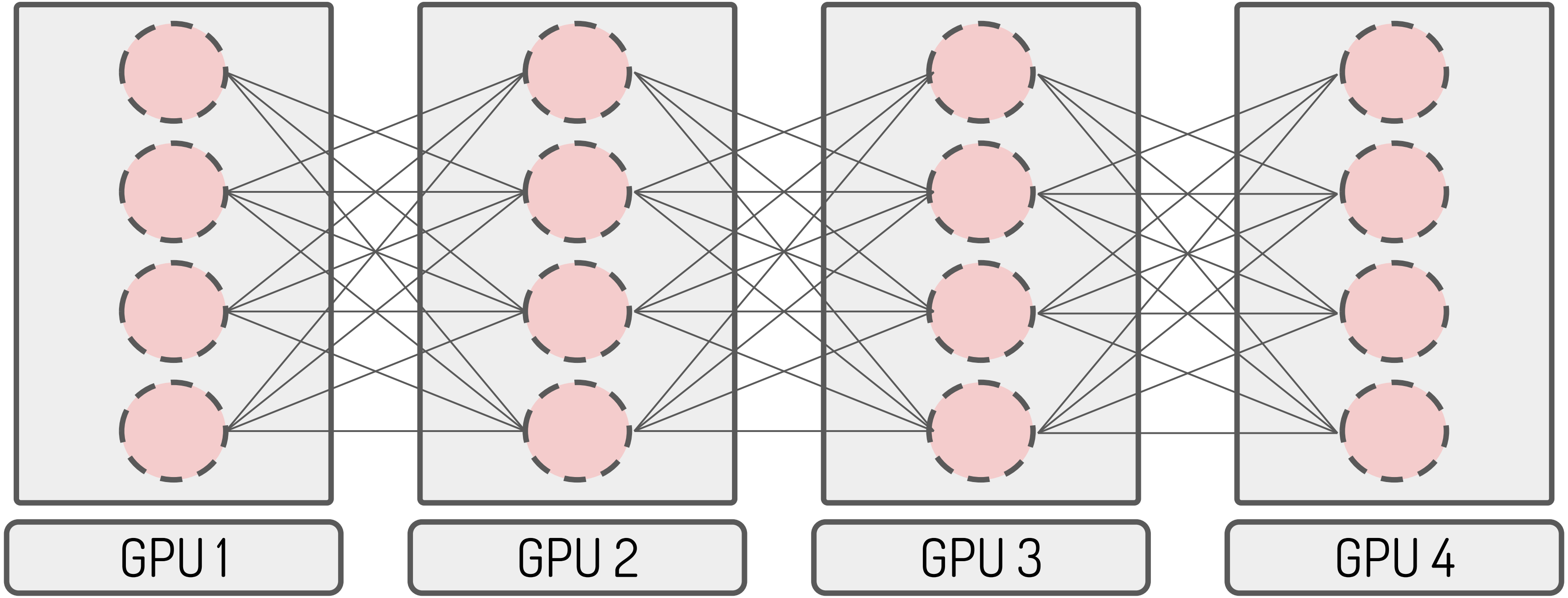
Pipeline Parallelism



Top: The naive model parallelism strategy leads to severe underutilization due to the sequential nature of the network. Only one accelerator is active at a time. **Bottom:** GPipe divides the input mini-batch into smaller micro-batches, enabling different accelerators to work on separate micro-batches at the same time.

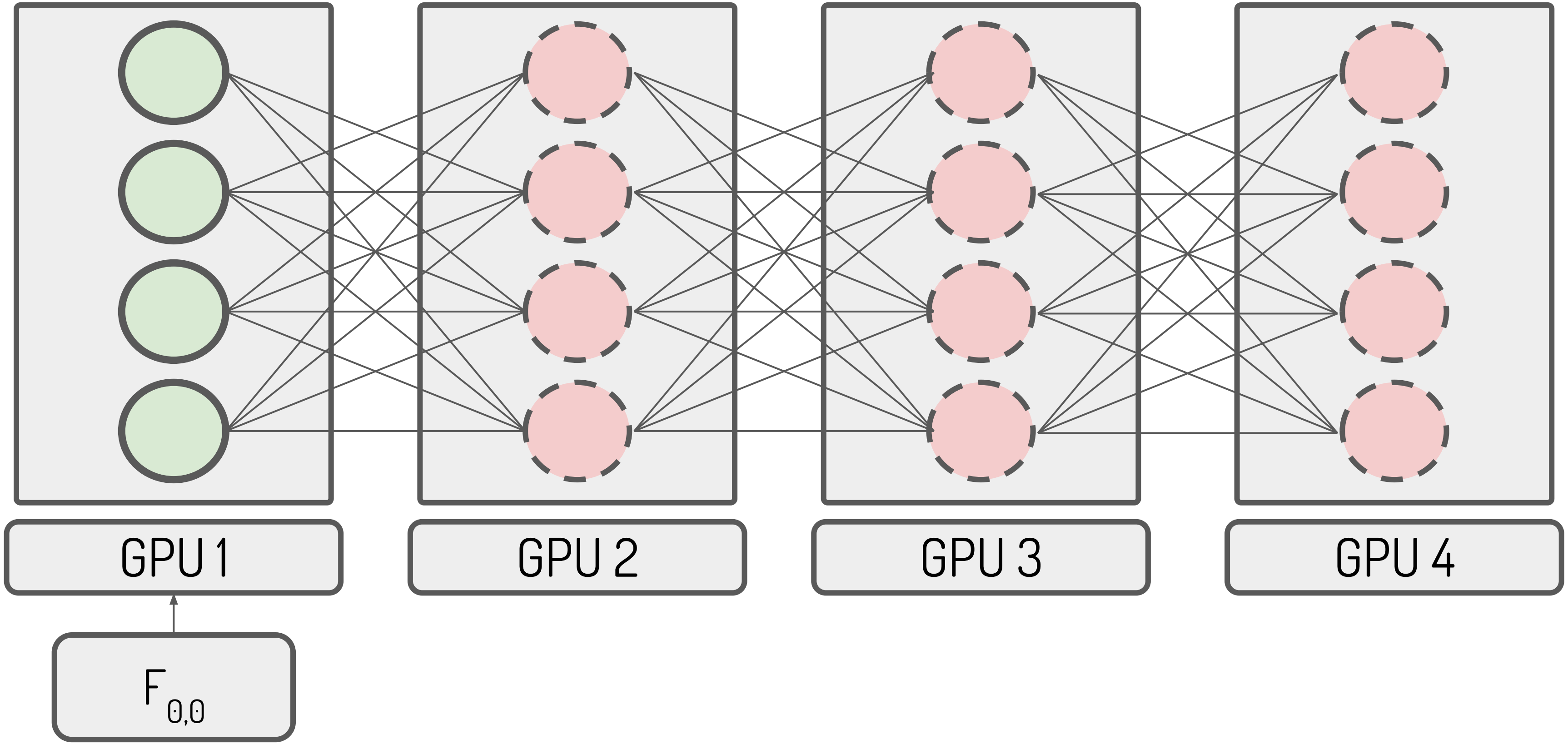
Illustration by [Google AI Blog (GPipe)]

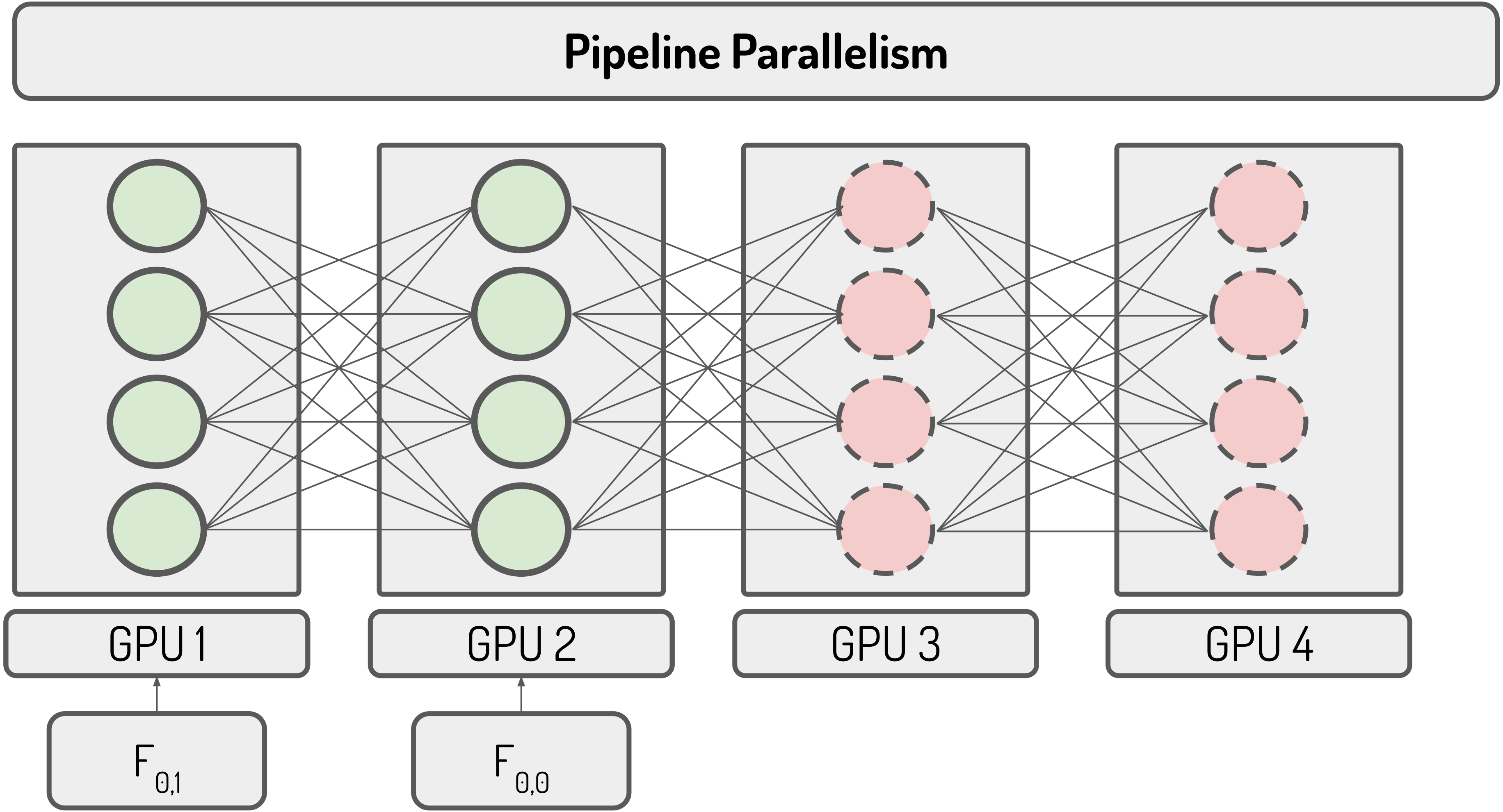
Pipeline Parallelism



Split mini-batch into sequential micro-batches

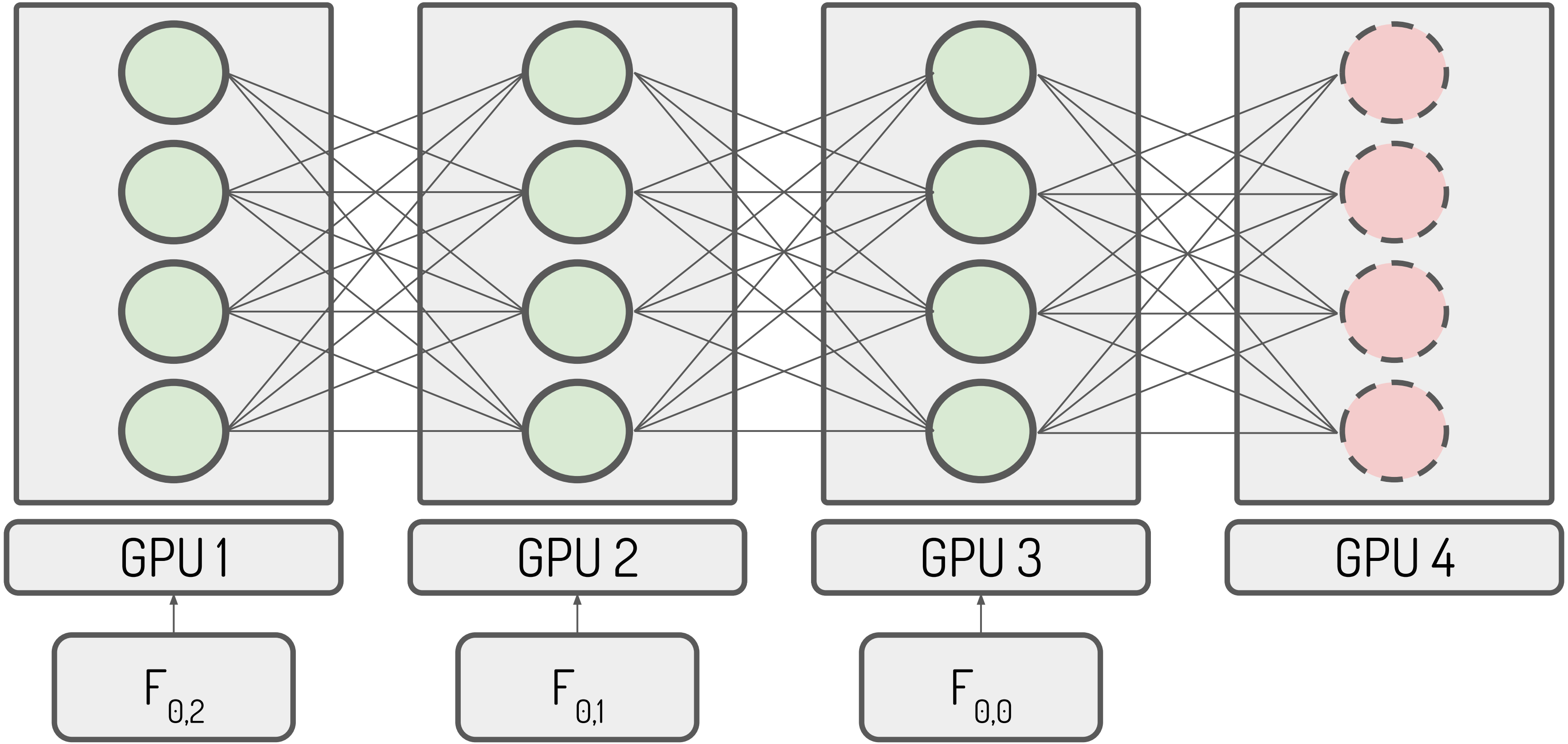
Pipeline Parallelism

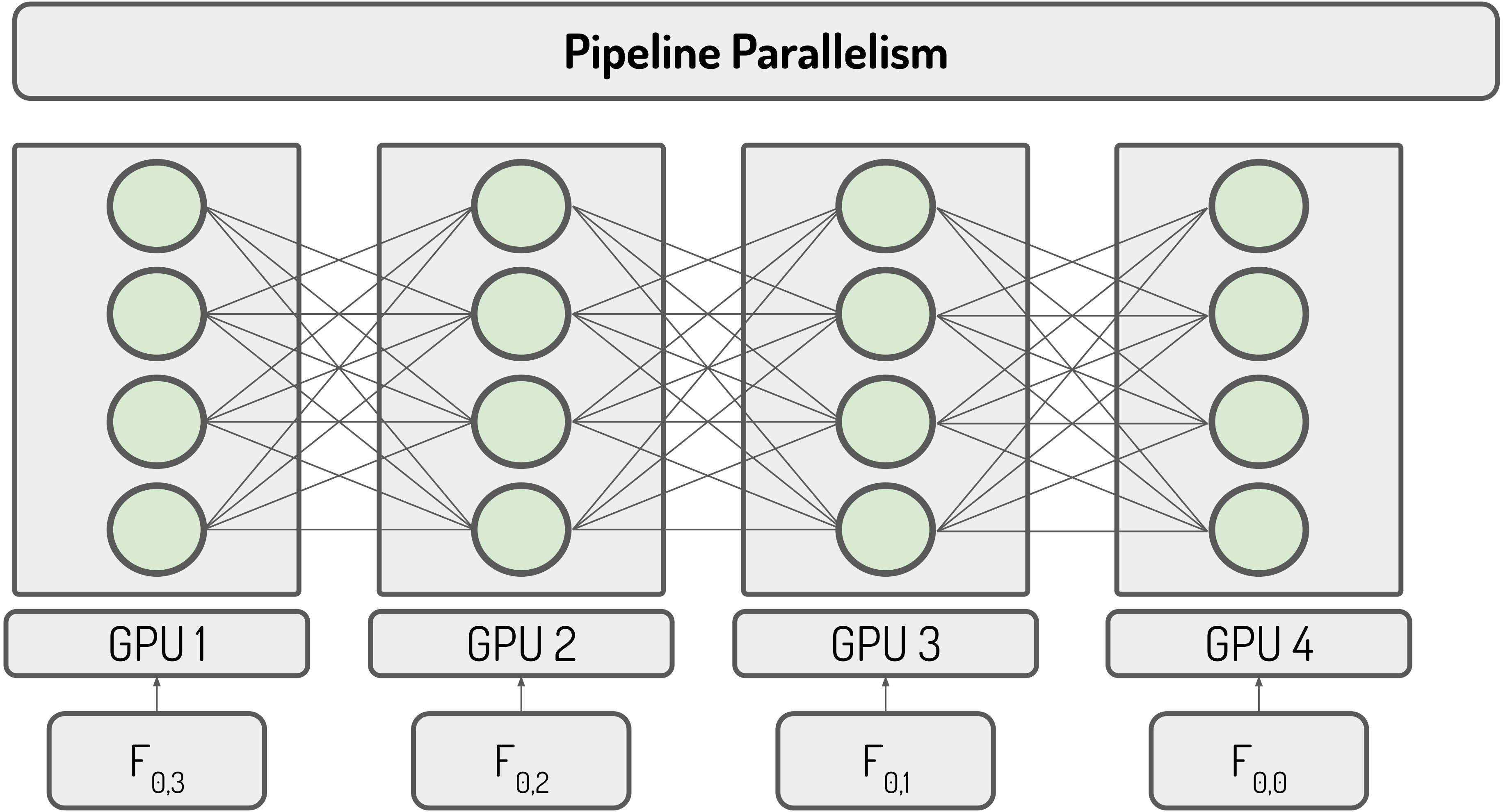




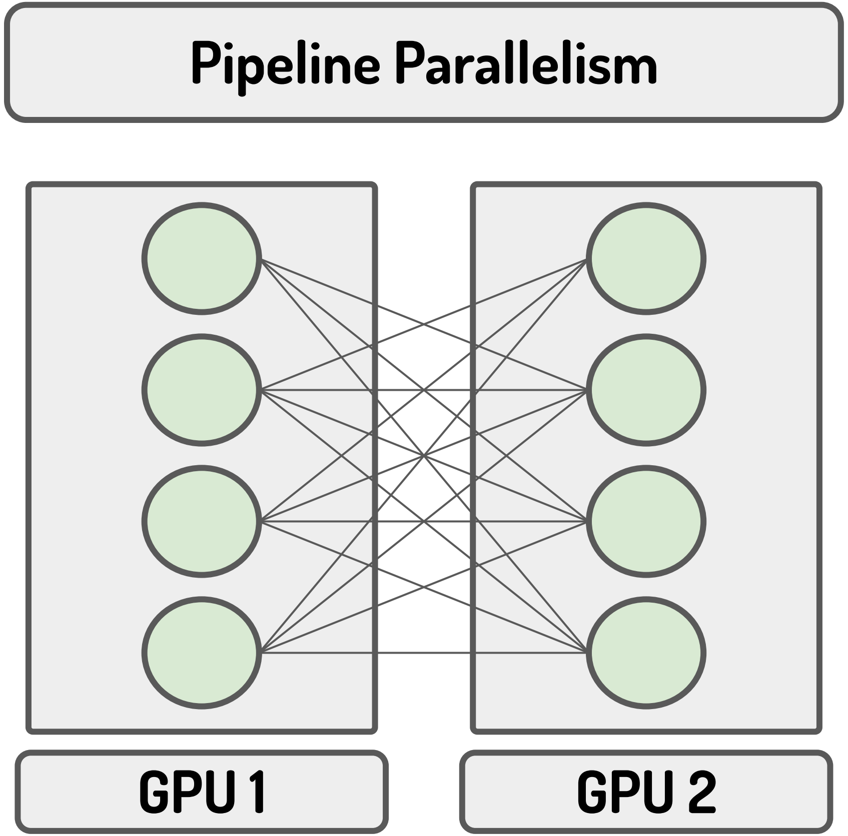
Pipeline Parallelism
GPU 1
GPU 2
GPU 3
GPU 4
$F_{0,1}$
$F_{0,0}$

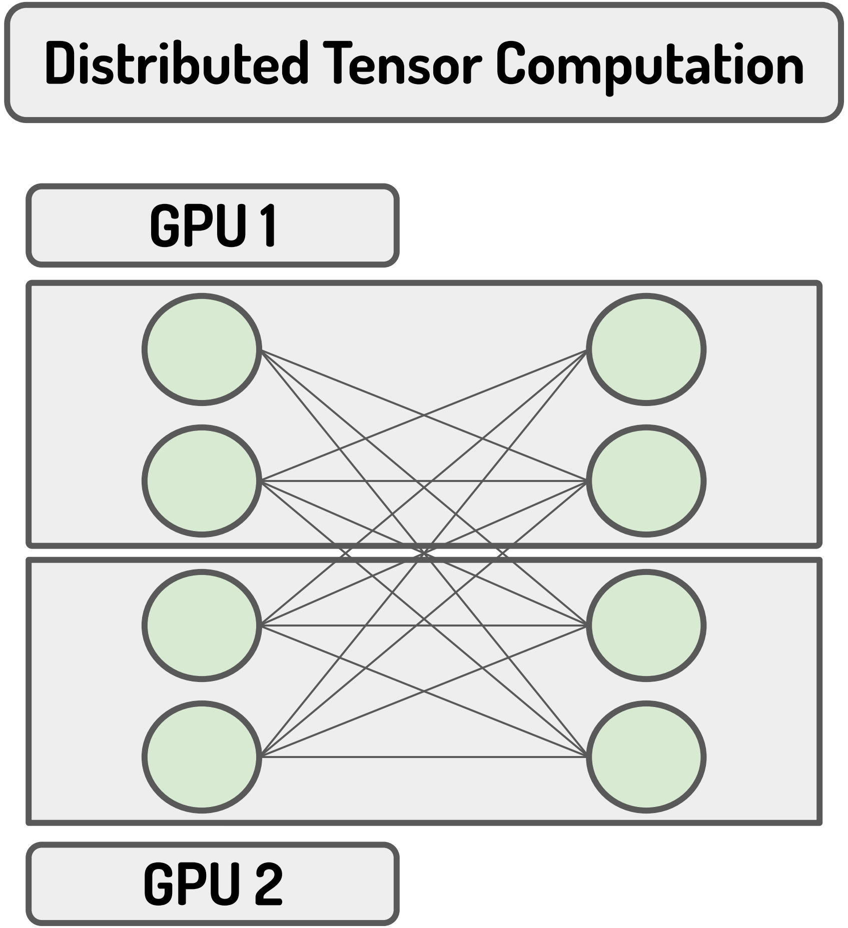
Pipeline Parallelism





Pipeline Parallelism
GPU 1
GPU 2
GPU 3
GPU 4
$F_{0,3}$
$F_{0,2}$
$F_{0,1}$
$F_{0,0}$


Pipeline Parallelism
GPU 1
GPU 2


Distributed Tensor Computation
GPU 1
GPU 2

Combining Ideas!

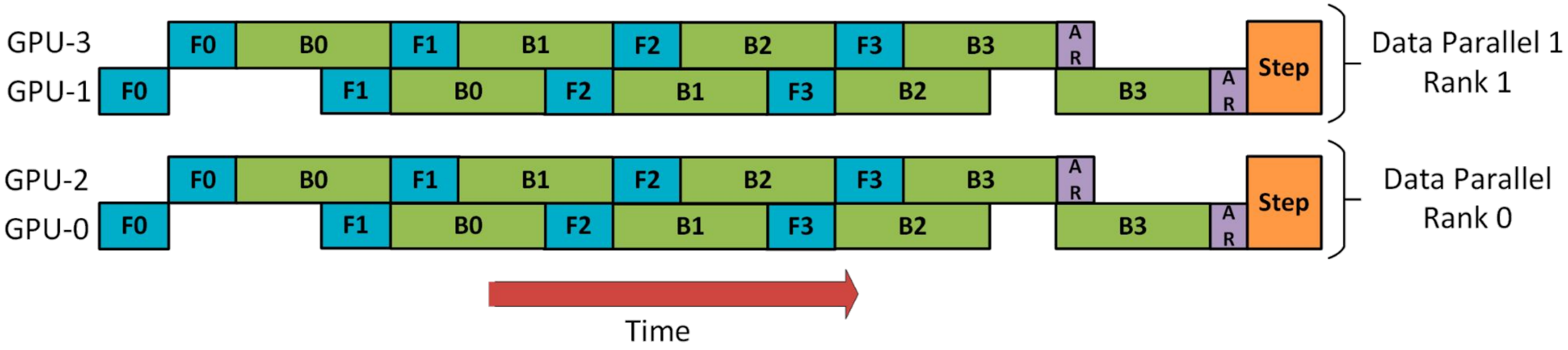
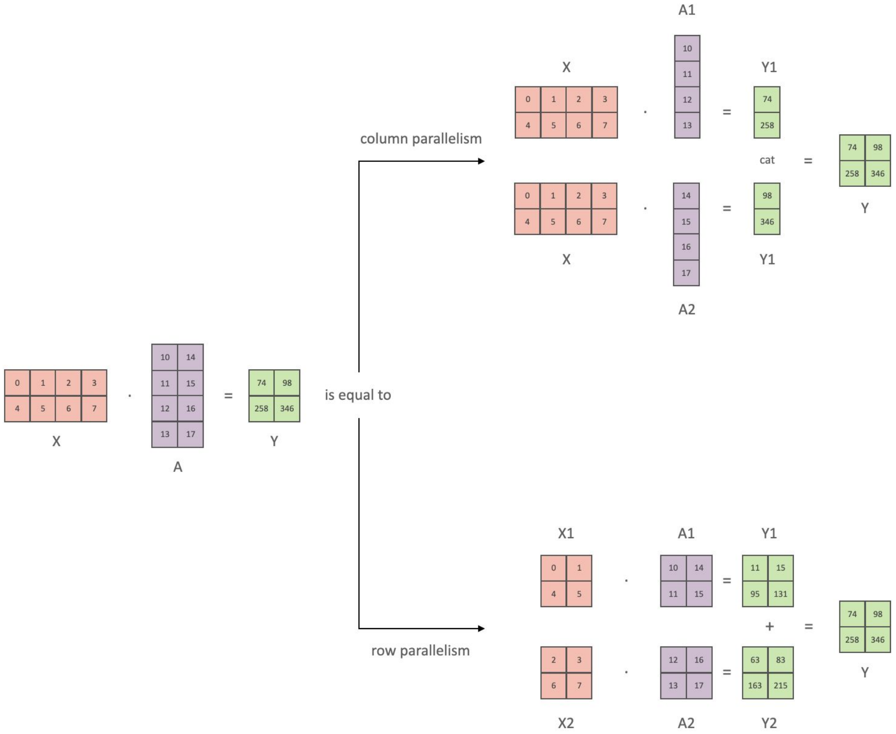
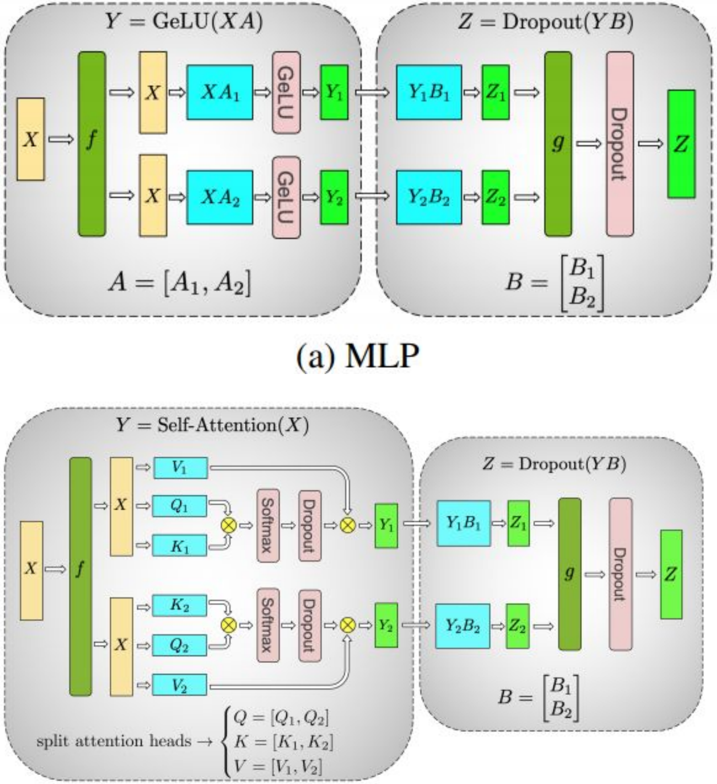


Illustration by [DeepSpeed]

Tensor Parallelism

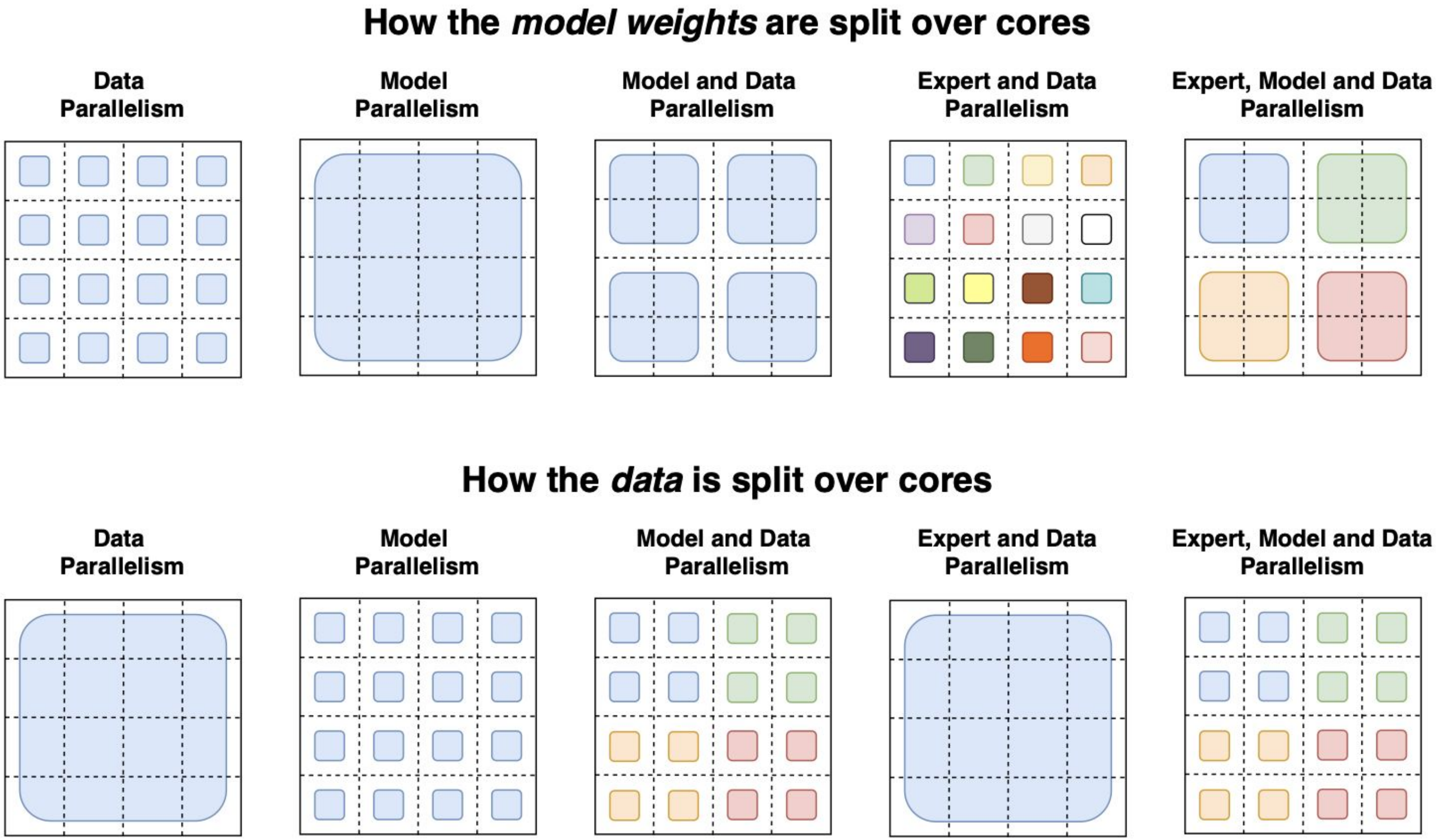




(a) MLP

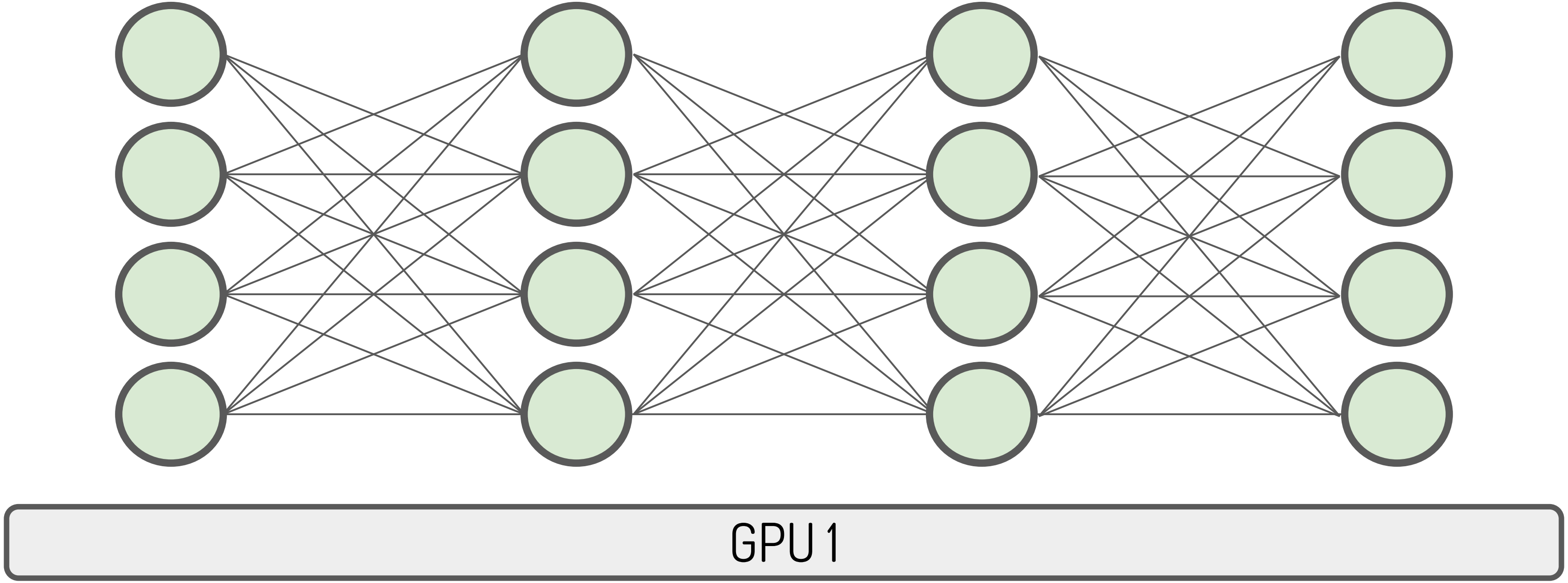
(b) Self-Attention

Illustration by [NVIDIA (Megatron-LM)]



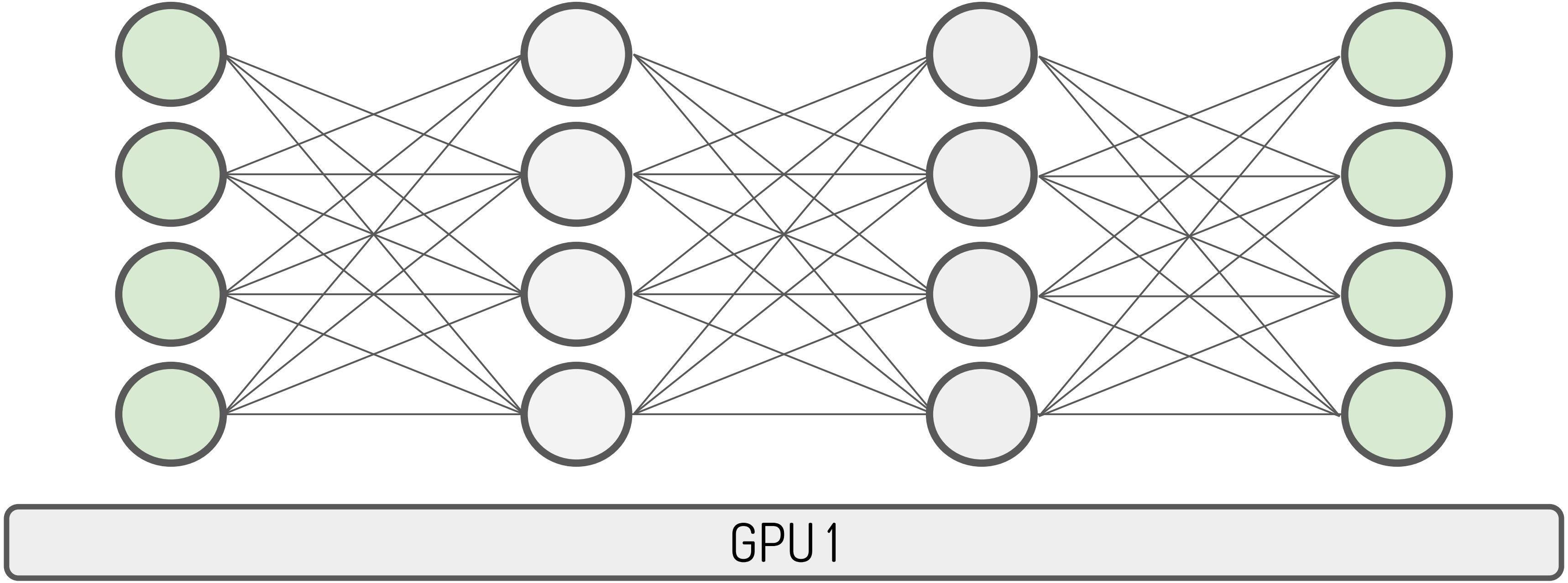
Credit: Fedus et al. (Switch Transformers: Scaling to Trillion Parameter Models with Simple and Efficient Sparsity)

Gradient Checkpointing

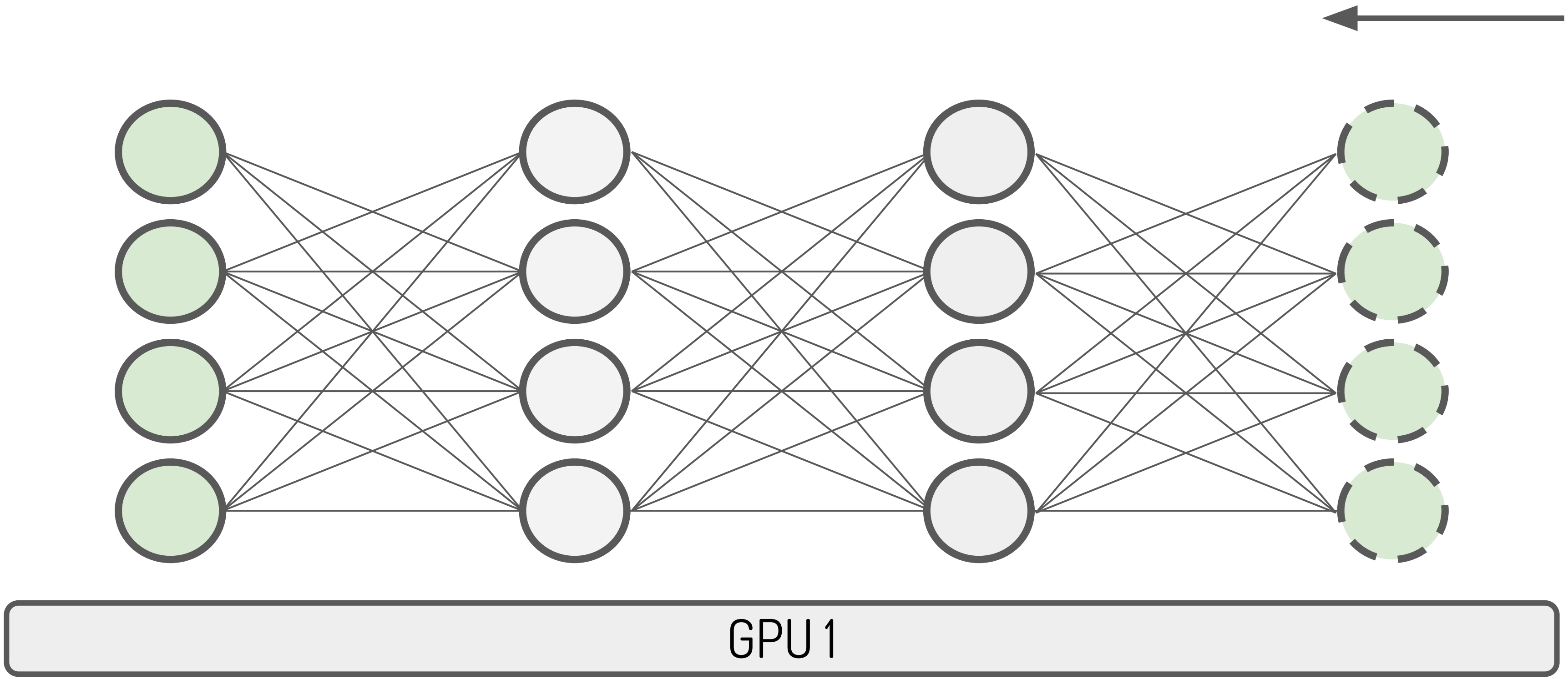


Trade off memory for compute

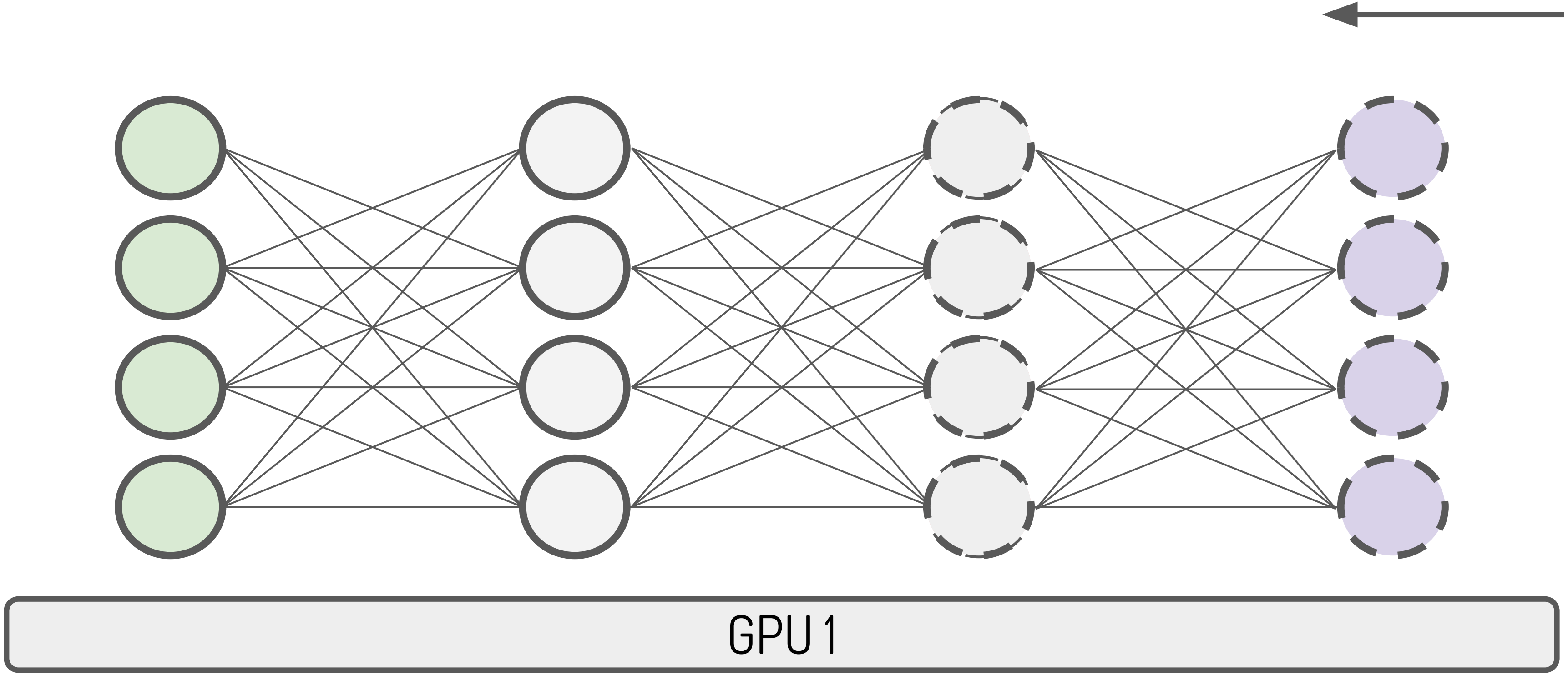
Gradient Checkpointing



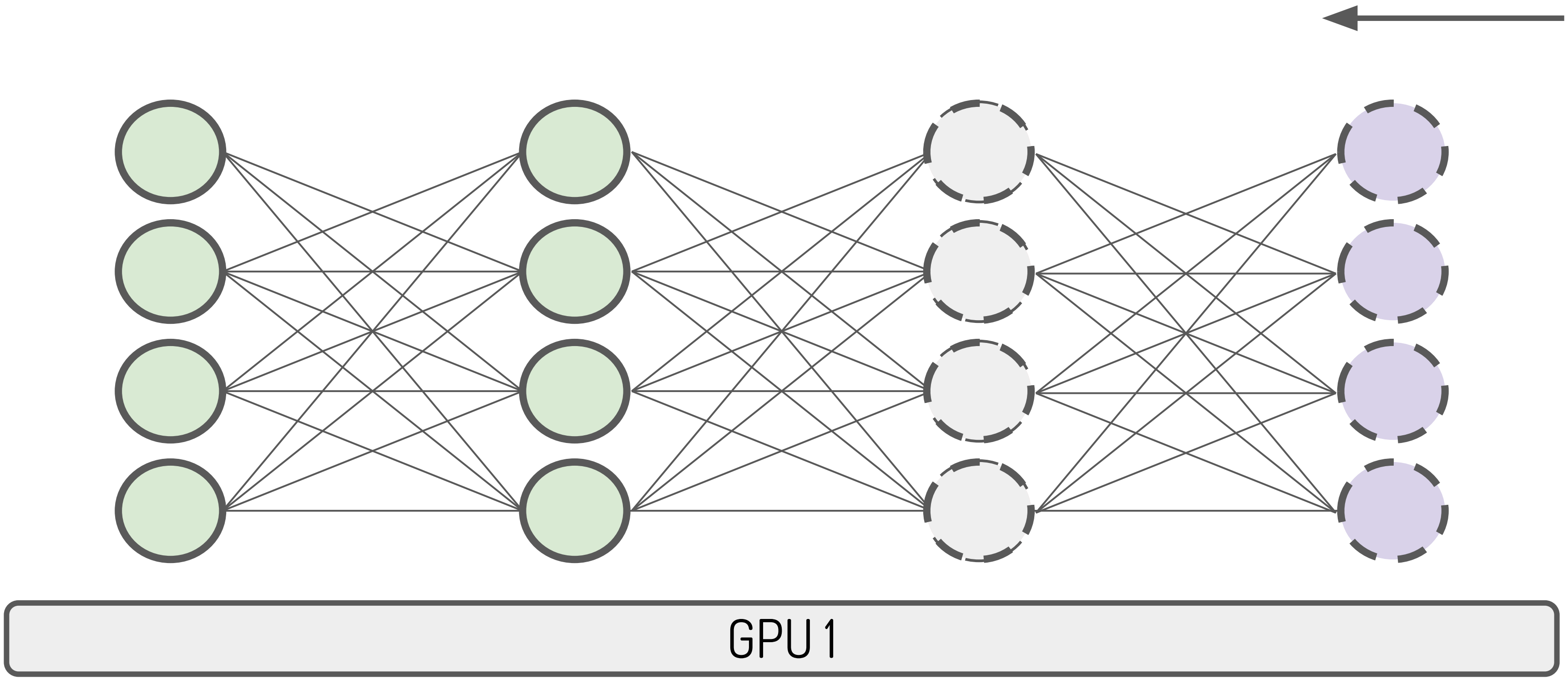
Don't store some activations in forward pass



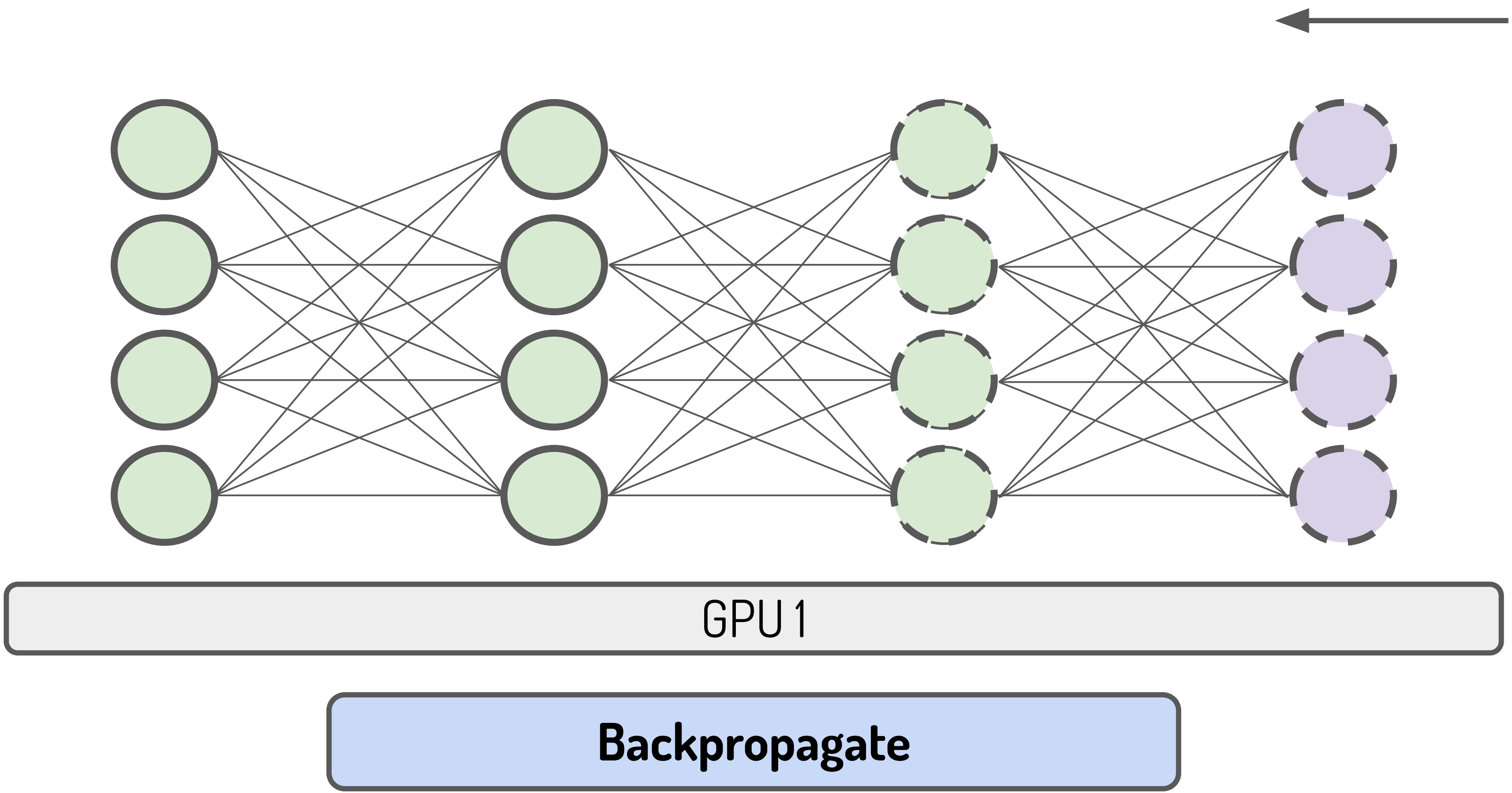
Backpropagate

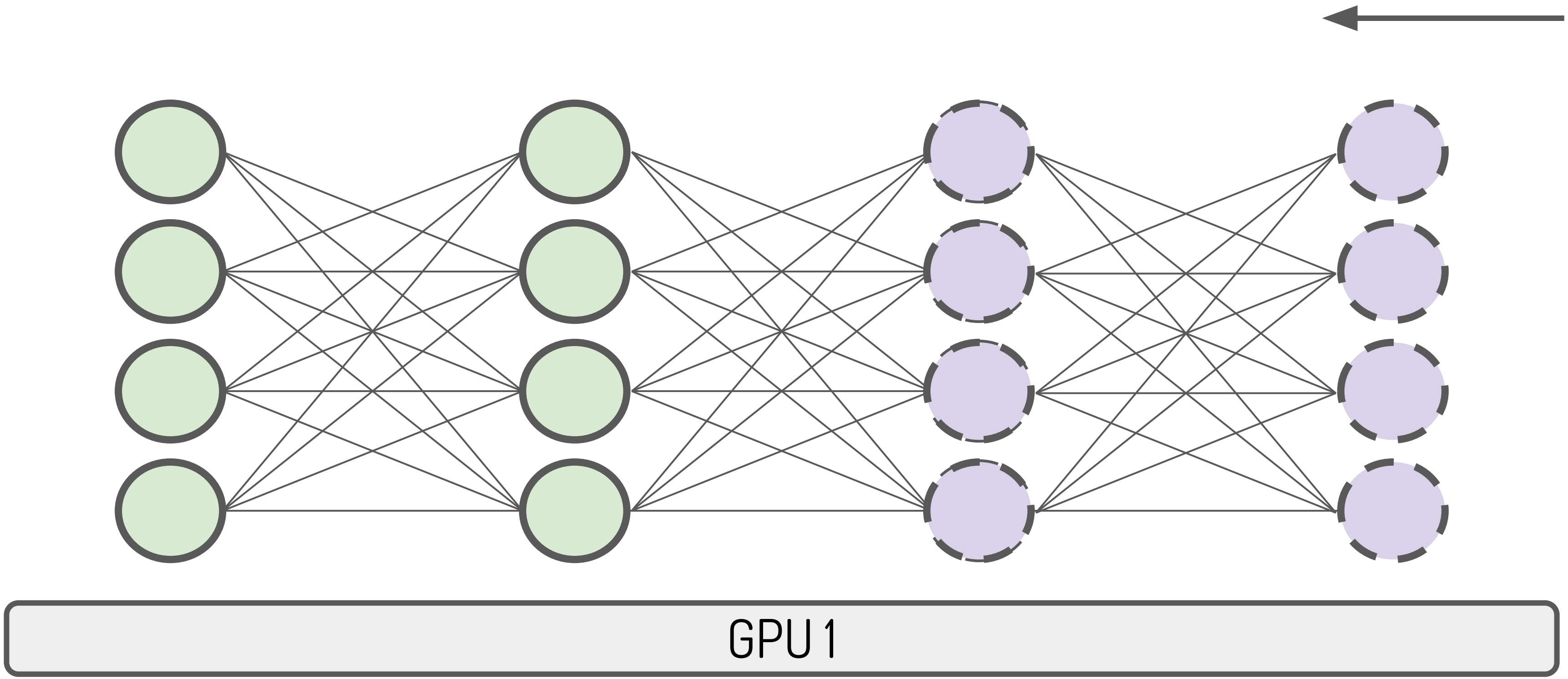


Don’t have activations!

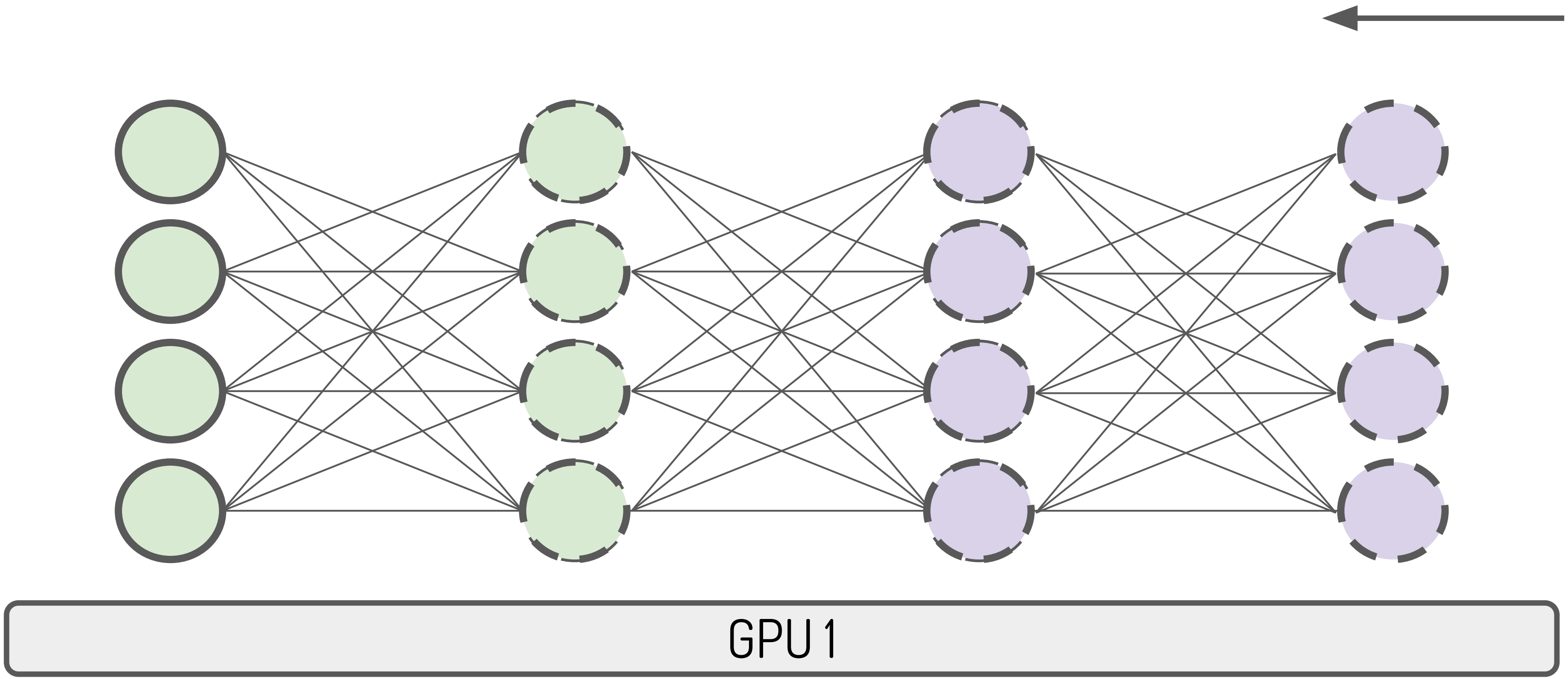


Recompute activations from checkpoint

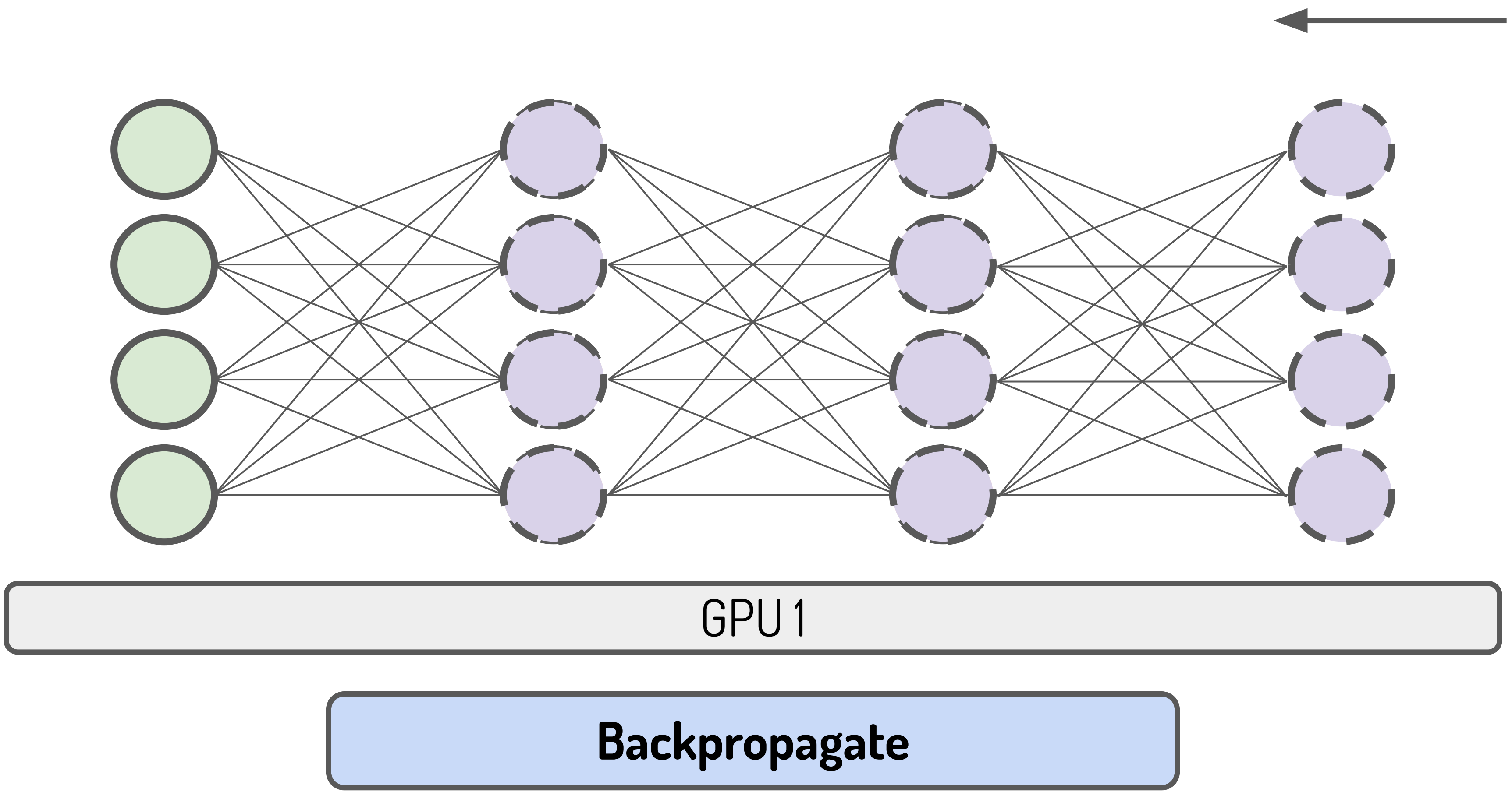

GPU 1
Backpropagate

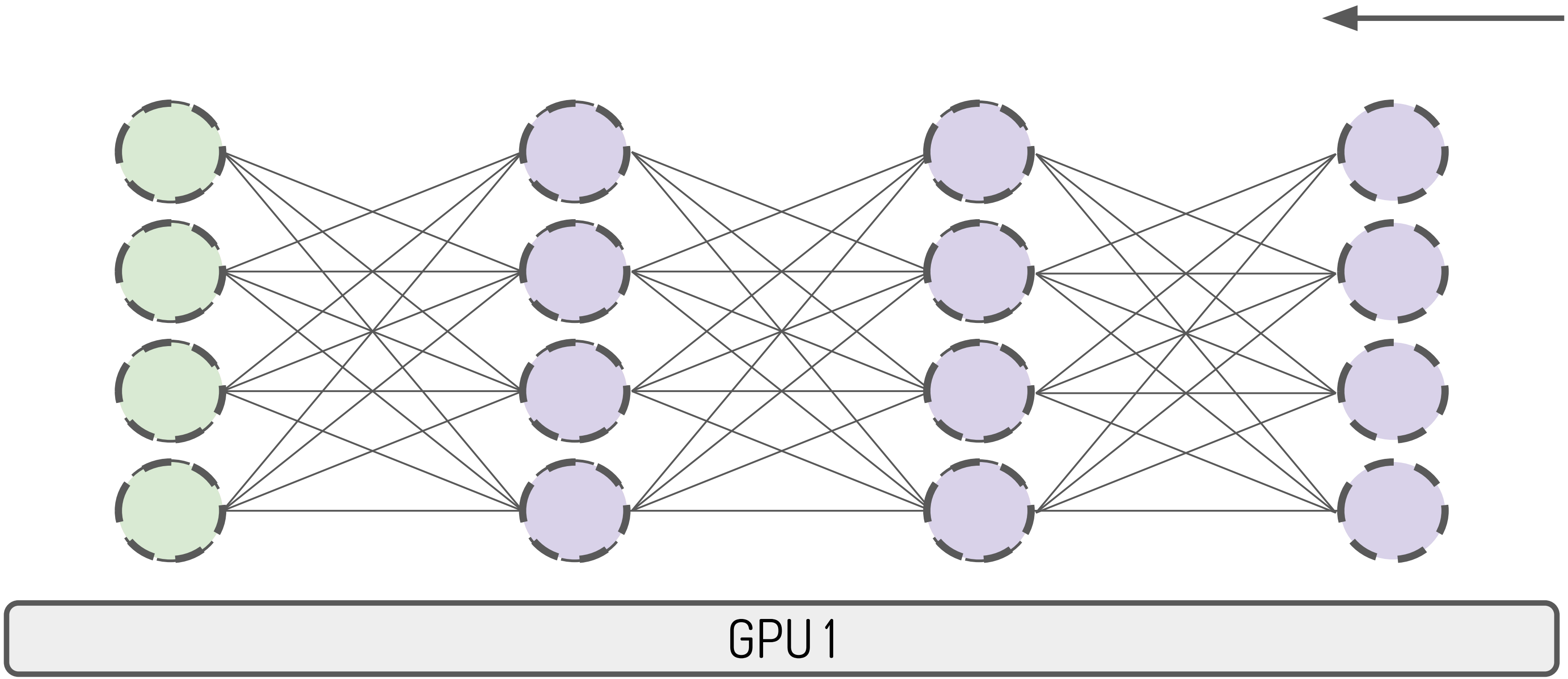


Backpropagate

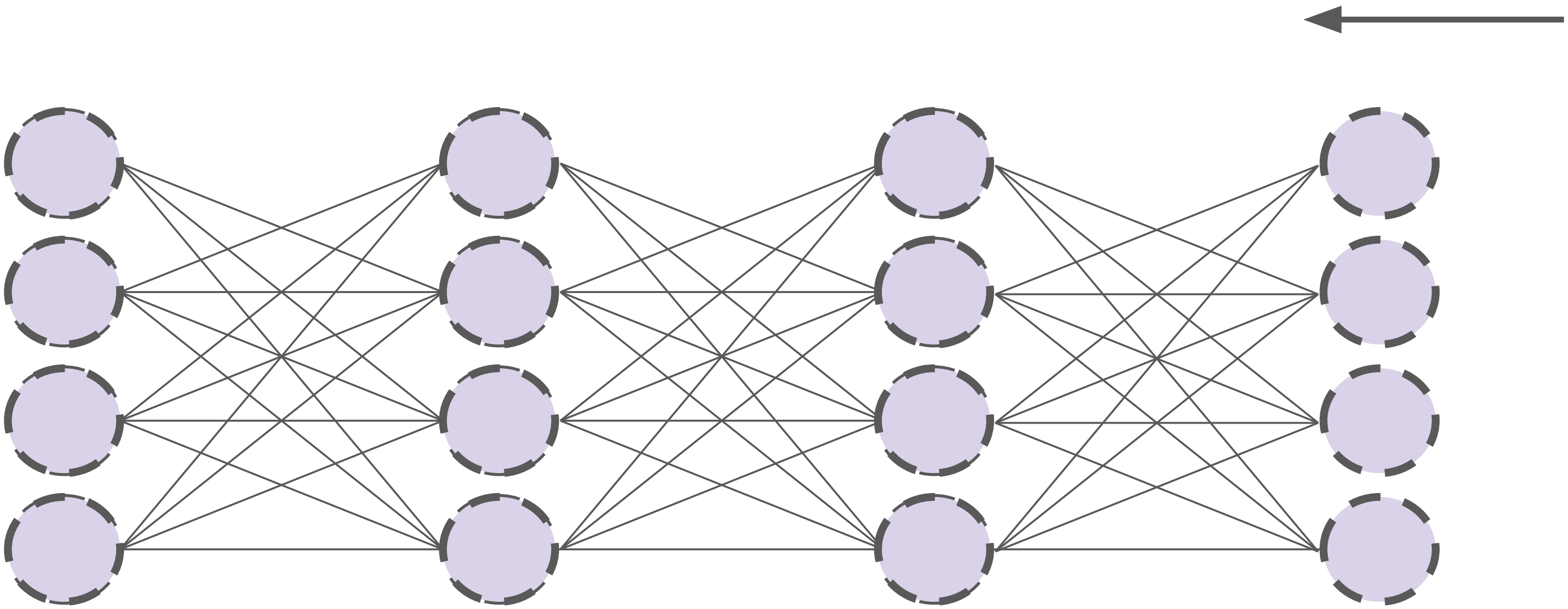


Backpropagate


GPU 1
Backpropagate



Backpropagate



GPU 1

Backpropagate

Breakout exercise

What went wrong with Zillow Offers?

Zillow, facing big losses, quits flipping houses and will lay off a quarter of its staff.

The real estate website had been relying on its algorithm that estimates home values to buy and resell homes. That part of its business lost about $420 million in three months.

Zillow is sitting on thousands of houses worth less than what the company paid for them. Caitlin O'Hara for The New York Times

https://www.nytimes.com/2021/11/02/business/zillow-q3-earnings-home-flipping-ibuying.html (November, 2021)

Blaming game

1. Prophet
2. Kaggle-style data science
3. Leadership
4. ML/DS team

Sean Kross
@seankross

Zillow Prize: Zillow's Home Value Prediction (Zestimate)

Can you improve the algorithm that changed the world of real estate?

$1,200,000
Prize Money

Zillow · 3,770 teams · 4 years ago

What went wrong with Zillow Offers?

1. Use ML to predict home prices
2. Use predicted prices to flip houses
3. ML models over-predict house prices
4. Buy houses at higher prices

Group of 5, 10 minutes

1. What might be the causes of ML models over-predicting house prices?
   a. Hint: what market conditions have changed in the last 2 years?
2. If you were on their team, what would you have done to prevent this problem?

ML offline evaluation

Facebook translates 'good morning' into 'attack them', leading to arrest

Palestinian man questioned by Israeli police after embarrassing mistranslation of caption under photo of him leaning against bulldozer

0 People Like You

Model evaluation

- Offline evaluation: before deployed
- Online evaluation: after deployed

Test in production. Will cover this later!

Model offline evaluation

- Baselines
- Evaluation methods

Baselines

- Numbers by themselves mean little
- Task: binary classification, 90% POSITIVE, 10% NEGATIVE
- F1 score: 0.90

Is it model good or bad?

Model selection: baselines

- **Random baseline**
  - Predict at random:
    - uniform
    - following label distribution

Model selection: baselines

- **Random baseline**
  - Predict at random:
    - uniform
    - following label distribution

- **Example**: misinformation classification
  - n = 1,000,000
  - 99% negative (label = 0)
  - 1% positive (label = 1)

	Accuracy	**F1**
Random [uniform]	0.5	?
Random [label distribution]	0.98	?

Model selection: baselines

- **Random baseline**
  - Predict at random:
    - uniform
    - following label distribution

- **Example**: misinformation classification
  - n = 1,000,000
  - 99% negative (label = 0)
  - 1% positive (label = 1)

	Accuracy	**F1**
Random [uniform]	0.5	0.02
Random [label distribution]	0.98	0.01

Model selection: baselines

- **Random baseline**
  - Predict at random:
    - uniform
    - following label distribution
- **Zero rule baseline**
  - Always predict the most common class

- **Example**: misinformation classification
  - n = 1,000,000
  - 99% negative (label = 0)
  - 1% positive (label = 1)

	Accuracy	**F1**
Random [uniform]	0.5	0.02
Random [label distribution]	0.98	0.01
Most common [preds = [0] * n]	?	?

Model selection: baselines

- **Random baseline**
  - Predict at random:
    - uniform
    - following label distribution
- **Zero rule baseline**
  - Always predict the most common class
- **Simple heuristics**
  - E.g.: classify tweets based on whether they contain links to unreliable sources

- **Example**: misinformation classification
  - n = 1,000,000
  - 99% negative (label = 0)
  - 1% positive (label = 1)

	Accuracy	**F1**
Random [uniform]	0.5	0.02
Random [label distribution]	0.98	0.01
Most common [preds = [0] * n]	?	?
Simple heuristics	?	?

Model selection: baselines

- **Random baseline**
  - Predict at random:
    - uniform
    - following label distribution
- **Zero rule baseline**
  - Always predict the most common class
- **Simple heuristics**
  - E.g.: classify tweets based on whether they contain links to unreliable sources
- **Human baseline**
  - What’s human-level performance?

- **Example**: misinformation classification
  - n = 1,000,000
  - 99% negative (label = 0)
  - 1% positive (label = 1)

	Accuracy	**F1**
Random [uniform]	0.5	0.02
Random [label distribution]	0.98	0.01
Most common [preds = [0] * n]	?	?
Simple heuristics	?	?
Human expert	?	?

Model selection: baselines

- **Random baseline**
  - Predict at random:
    - uniform
    - following label distribution
- **Zero rule baseline**
  - Always predict the most common class
- **Simple heuristics**
  - E.g.: classify tweets based on whether they contain links to unreliable sources
- **Human baseline**
  - What's human-level performance?
- **Existing solutions**

- **Example**: misinformation classification
  - n = 1,000,000
  - 99% negative (label = 0)
  - 1% positive (label = 1)

	Accuracy	**F1**
Random [uniform]	0.5	0.02
Random [label distribution]	0.98	0.01
Most common [preds = [0] * n]	?	?
Simple heuristics	?	?
Human expert	?	?
3rd party API	?	?

Evaluation methods

1. Perturbation Tests
2. Invariance Tests
3. Directional Expectation Tests
4. Model Calibration
5. Confidence Measurement
6. Slice-based Evaluation

Perturbation tests

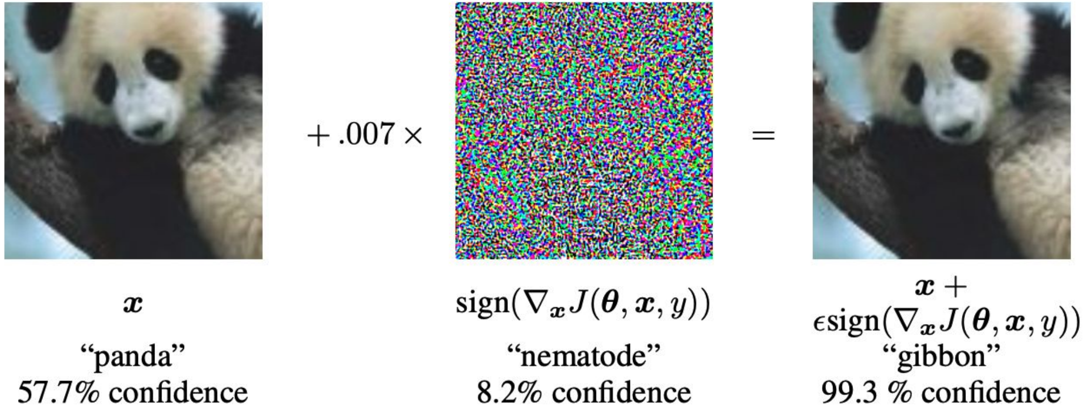
- Problem: users input might contain noise, making it different from test data
  - Examples:
    - Speech recognition: background noise
    - Object detection: different lighting
    - Text inputs: typos, intentional misspelling (e.g. looooooooong)
  - Model does well on test set, but fails in production

Perturbation tests

- Motivation: users input might contain noise, making it different from test data
- Idea: randomly add small noise to test data to see how much outputs change

Perturbation tests

- Motivation: users input might contain noise, making it different from test data
- Idea: randomly add small noise to test data to see how much outputs change
- The more sensitive the model is to noise:
    - The harder it is to maintain
    - The more vulnerable the model is to adversarial attacks



Perturbation tests

- Motivation: users input might contain noise, making it different from test data
- Idea: randomly add small noise to test data to see how much outputs change

If small changes cause model's performance to fluctuate,
you might want to make model more robust:

- Add noise to training data
- Add more training data
- Choose another model

Invariance tests

- Motivation: some input changes shouldn't lead to changes in outputs
  - Changing race/gender info shouldn't change predicted approval outcome
  - Changing name shouldn't affect resume screening results

The Berkeley study found that both face-to-face and online lenders rejected a total of 1.3 million creditworthy black and Latino applicants between 2008 and 2015. Researchers said they believe the applicants "would have been accepted had the applicant not been in these minority groups." That's because when they used the income and credit scores of the rejected applications but deleted the race identifiers, the mortgage application was accepted.

Disparity in home lending costs minorities millions, researchers find (CBS News, 2019)

Invariance tests

- Motivation: some input changes shouldn’t lead to changes in outputs
- Idea: keep certain features the same, but randomly change values of sensitive features

If changing sensitive features can change model’s outputs, there might be biases!

Directional expectation tests

- Motivation: some changes to inputs should cause predictable changes in outputs
    - E.g. when predicting housing prices:
        - Increasing lot size shouldn't decrease the predicted price
        - Decreasing square footage shouldn't increase the predicted price

Directional expectation tests

- Motivation: some changes to inputs should cause predictable changes in outputs
- Idea: keep most features the same, but change certain features to see if outputs change predictably

If increasing lot size consistently reduces the predicted price, you might want to investigate why!

Model calibration

"One of the most important tests of a forecast — I would argue that it is the single most important one — is called calibration."

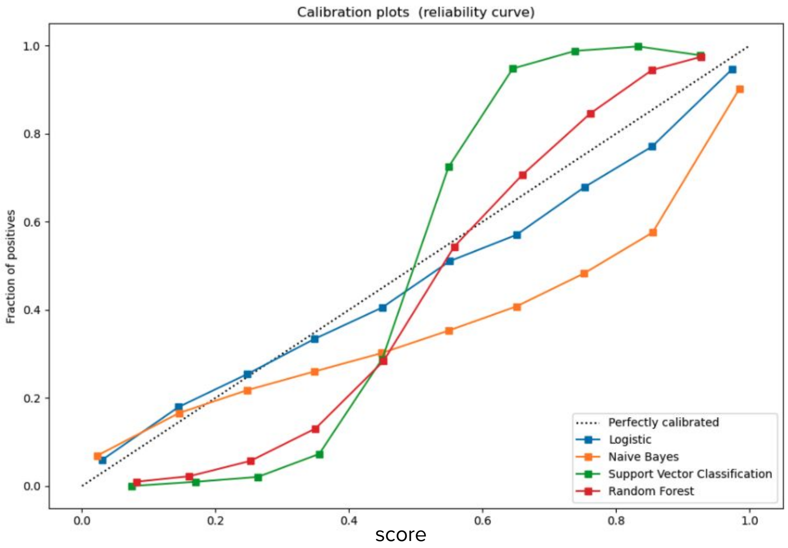
Nate Silver, **The Signal and the Noise**

Model calibration

- If you predict team A wins in A vs. B match with 60% probability:
  - In 100 A vs. B match, A should win 60% of the time!

Model calibration: binary case

Among all samples predicted POSITIVE with propa 80%, .
80% of them should be POSITIVE



Need to ensure the **top class is correct on average**

Image from Probability calibration (sklearn)

Model calibration: recsys

- Recommend movies to a user who watches 70% comedy, 30% action
- What happens if you recommend most likely watched movies?

Movie title	**Watch probability**
Comedy 1	0.8
Comedy 2	0.73
Comedy 3	0.68
Comedy 4	0.67
Action 1	0.29
Action 2	0.2
Science fiction	0.04

Model calibration: recsys

- Recommend movies to a user who watches 70% comedy, 30% action
- What happens if you recommend most likely watched movies?

Movie title	**Watch probability**
Comedy 1	0.8
Comedy 2	0.73
Comedy 3	0.68
Comedy 4	0.67
Action 1	0.29
Action 2	0.2
Science fiction	0.04

Need to calibrate recommendations to include 70% comedy, 30% action

Model calibration: CTR

- 2 ads: A & B
- Model predicts click probability: A (10%), B (8%)
- How to estimate number of clicks you’ll actually get if model isn’t calibrated?

Confidence measurement

- Usefulness threshold for each individual prediction
- Uncertain predictions can cause annoyance & catastrophic consequences

Confidence measurement

- How to measure the confidence level of each prediction?
- What to do with predictions below the confidence threshold?
    - Skip
    - Ask for more information
    - Loop in humans

Slice-based evaluation

Different performance on different slices

- Classes
  - Might perform worse on minority classes
- Subgroups
  - Gender
  - Location
  - Time of using the app
  - etc.

Same performance on different slices with different cost

- User churn prediction
  - Paying users are more critical
- Predicting adverse drug reactions
  - Patients with underlying conditions are more critical

⚠️ Focusing on improving only overall metrics might hurt performance on subgroups ⚠️

Slice-based evaluation: example

- Majority group: 90%
- Minority group: 10%

Zoom poll: Which model would you go with?

	Majority accuracy	Minority accuracy
Model A	98%	80%
Model B	95%	95%

Slice-based evaluation: example

- Majority group: 90%
- Minority group: 10%

Coarse-grained evaluation can hide:
- model biases
- potential for improvement

	Majority accuracy	Minority accuracy	Overall accuracy
Model A	98%	80%	96.2%
Model B	95%	95%	95%

Simpson's paradox

- Models A and B to predict whether a customer will buy your product
- A performs better than B overall
- B performs better than A on both female & male customers

Simpson's paradox

	Treatment 1	Treatment 2
Group A	**93% (81/87)**	87% (234/270)
Group B	**73% (192/263)**	69% (55/80)
Overall	78% (273/350)	**83% (289/350)**

Numbers from a kidney stone treatment study. (Charig et al., 1986)

Simpson's paradox: Berkeley graduate admission '73

	All		Men		Women	
	Applicants	Admitted	Applicants	Admitted	Applicants	Admitted
Total	12,763	41%	8442	**44%**	4321	35%

Bias against women in the process, or is there?

Sex Bias in Graduate Admissions: Data from Berkeley (Bickel et al., 1975)

Simpson's paradox: Berkeley graduate admission '73

Department	All		Men		Women	
	Applicants	Admitted	Applicants	Admitted	Applicants	Admitted
A	933	64%	***825***	62%	108	**82%**
B	585	63%	***560***	63%	25	**68%**
C	918	35%	325	**37%**	***593***	34%
D	792	34%	417	33%	375	**35%**
E	584	25%	191	**28%**	***393***	24%
F	714	6%	373	6%	341	**7%**

⚠️ Aggregation can conceal and contradict actual situation ⚠️

Sex Bias in Graduate Admissions: Data from Berkeley (Bickel et al., 1975)

Slice-based evaluation

- Evaluate your model on different slices
  - E.g. when working with website traffic data, slice data among:
    - gender
    - mobile vs. desktop
    - browser
    - location
- Check for consistency over time
  - E.g. evaluate your model on data slices from each day

Slice-based evaluation

- Improve model's performance both overall and on critical data
- Help avoid biases
- Even when you don't think slices matter, slicing can:
  - give you confidence on your model (to convince your boss)
  - might reveal non-ML problems

How to identify slices?

- Heuristics
    - Might require subject matter expertise
- Error analysis
    - Patterns among misclassified samples
- Slice finder
    - Exhaustive/beam search
    - Clustering
    - Decision tree

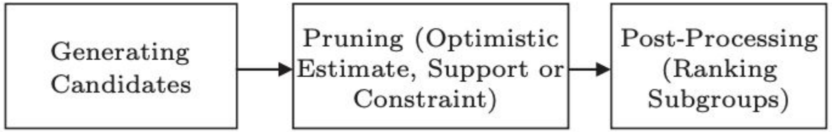


Fig.1. Methodology for subgroup discovery.

Slice finder: Automated data slicing for model validation (Chung et al., 2019)
Subgroup Discovery Algorithms: A Survey and Empirical Evaluation (Sumyea Helal, 2016)

How to identify slices?

Will go into details next lecture!

- Heuristics
  - Might require subject matter expertise
- Error analysis
  - Patterns among misclassified samples
- Slice finder
  - Exhaustive/beam search
  - Clustering
  - Decision tree

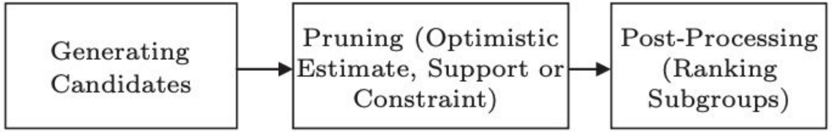


Fig.1. Methodology for subgroup discovery.

[Slice finder: Automated data slicing for model validation](#) (Chung et al., 2019)
[Subgroup Discovery Algorithms: A Survey and Empirical Evaluation](#) (Sumyea Helal, 2016)

Machine Learning Systems Design

Next class:

Evaluation Tutorial with Goku Mohandas + Chloe He