facebook

facebook

Accelerating Deep Learning at Facebook

Keith Adams
Facebook AI Research
XLDB 5/20/2015

Agenda

1. Intro to Deep Learning
2. Challenges
3. Single System Parallelization
4. Distributed Deep Learning
5. Conclusions

Deep learning: a hurried introduction

Review: Why Machine Learning?

- Suppose we wish to learn some function that is *so*:
  - Complicated
  - Ill-posed
  - Full of fiddly little corner-cases
  - …that manual programming is infeasible

Example: Image categorization

- *x:* [[(138,27,17), (135,28,18), …]]
- *y*: “Cat”

Example: Translation

x: A quarter of a century ago, barely half the children of primary school age in sub-Saharan Africa were enrolled in school.

y: Čtvrt stoletím, sotva polovina dětí ve školním věku v subsaharské Africe byli zařazeni do školy.

Example: Program Inference

- (3,2) -> 11
- (2,1) -> 7
- (-3, 4) -> 13

```
def inferred(a, b):

    return math.abs(a) * 3 + b
```

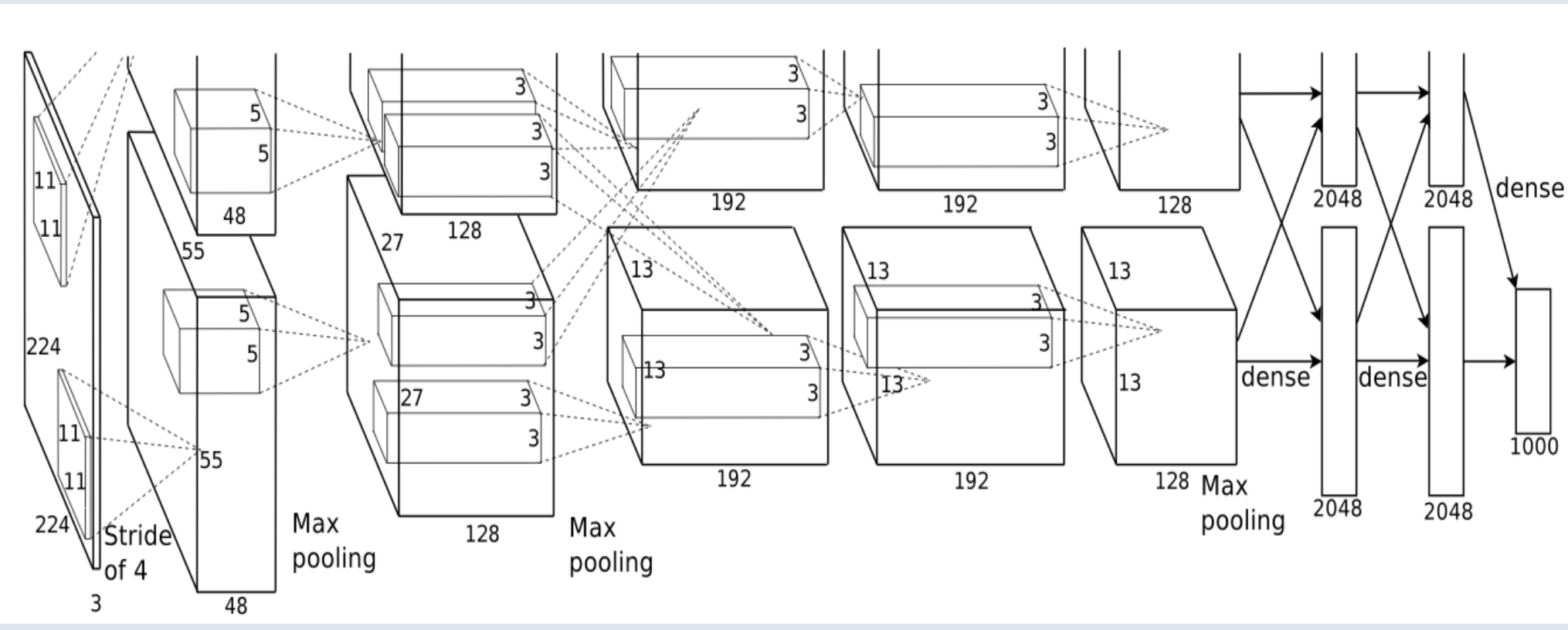
Classic ML vs. Deep Learning

- Raw inputs -> hand-engineered feature extractor -> learned model-> outputs

- Raw inputs -> learned feature extractor -> learned model-> outputs

What's "deep" about deep learning?

- Raw inputs -> learned low-level features -> learned intermediate-level features-> learned high-level features -> …-> outputs



Krizhevsky et al. NIPS'12

Video: Sport Classification Results

Method	Number of Nets	Clip hit@1	Video hit@1	Video hit@5
Deep Video's Single-Frame + Multires [19]	3 nets	42.4	60.0	78.5
Deep Video's Slow Fusion [19]	1 net	41.9	60.9	80.2
C3D (trained from scratch)	1 net	44.9	60.0	84.4
C3D (fine-tuned from I380K pre-trained model)	1 net	**46.1**	**61.1**	**85.2**

- Tran et al., http://arxiv.org/abs/1412.0767

Deep learning: challenges

Top 7 Problems with Deep Learning

1. Long training times
2. Long training times
3. Long training times
4. Long training times
5. Long training times
6. Long training times
7. Long training times

GPGPUs

- FLOP/s
- Memory bandwidth

Stochastic Gradient Descent (*SGD*)

while true:

Compute model's *error* on subset of training data

Use the chain rule* to compute a *gradient* for every parameter in the model

Adjust model parameters a small amount in direction of error gradient

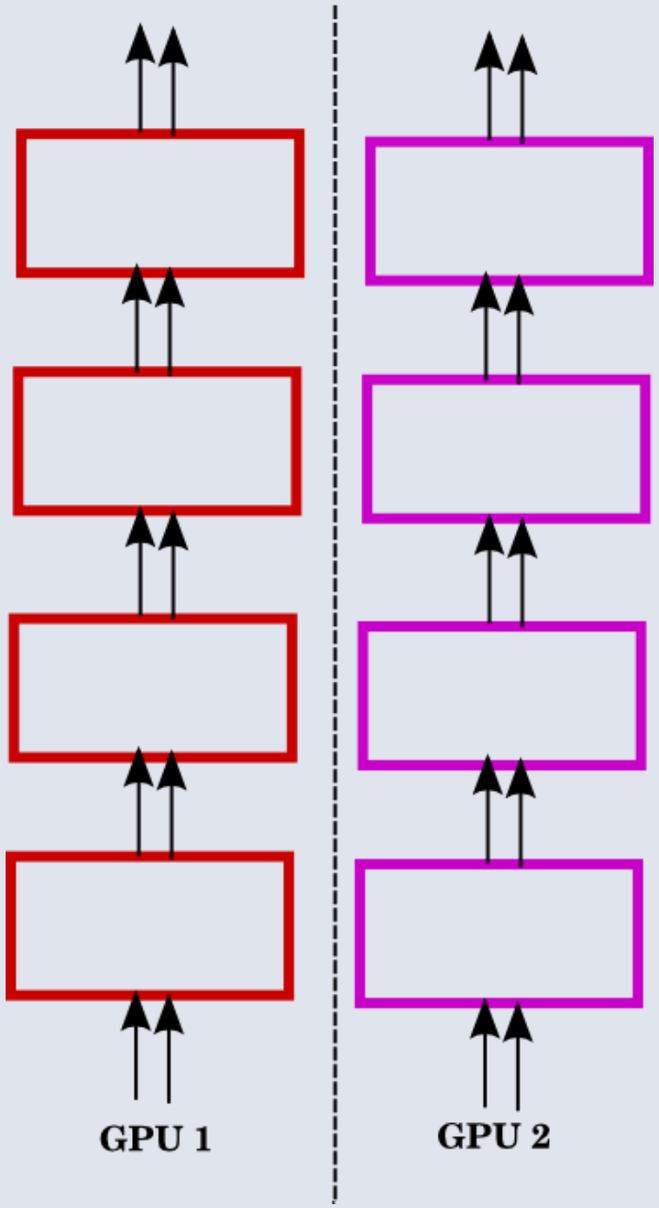
* *Really!*

SGD Observations

- Inherently *serial*
- Gradients are big
- Models are big
- Training datasets are *big*
- SGD is a hard algorithm to parallelize

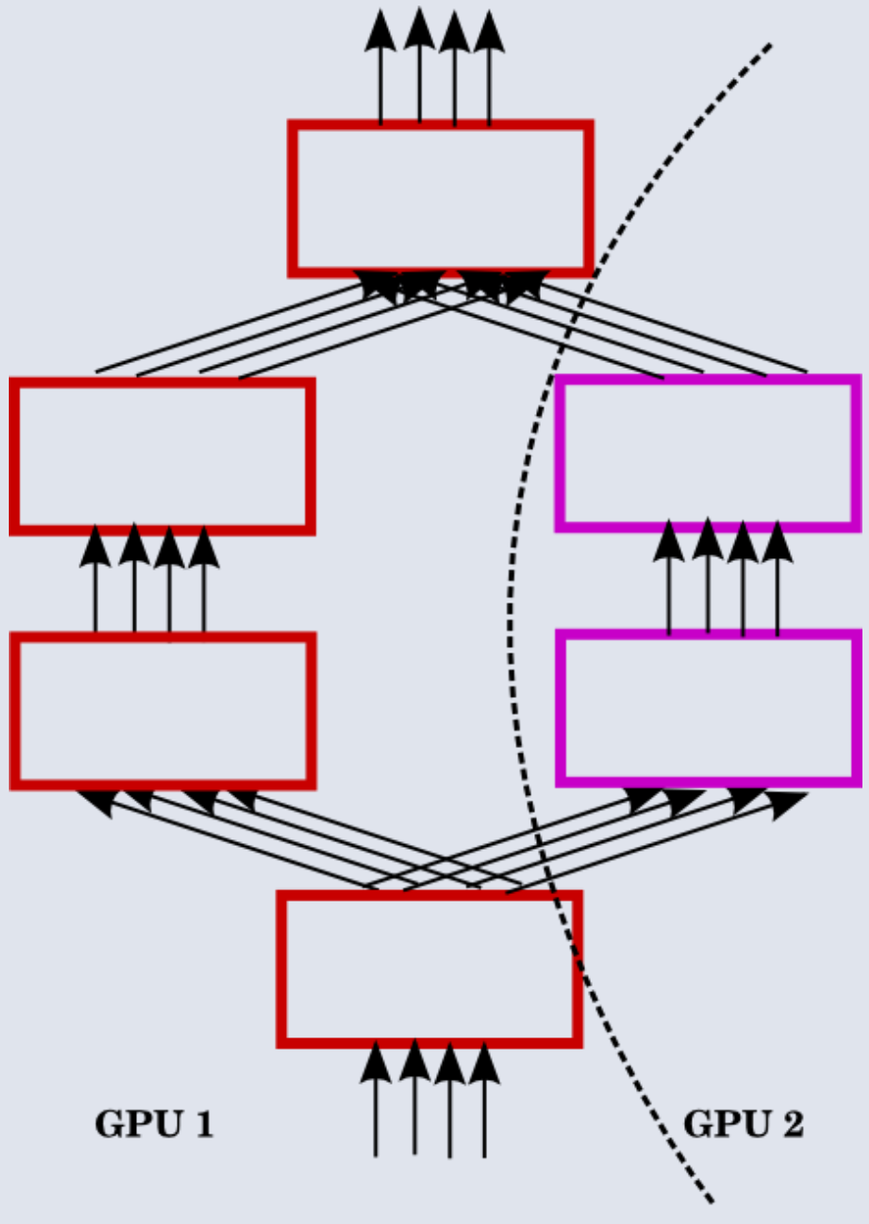
Strategies for Parallelism: Data Parallel

- N Workers
- Each gets 1/N of the training data
- After each minibatch:
  - Gradients averaged
  - Each worker applies average gradient
- Communication proportional to the number of parameters in model

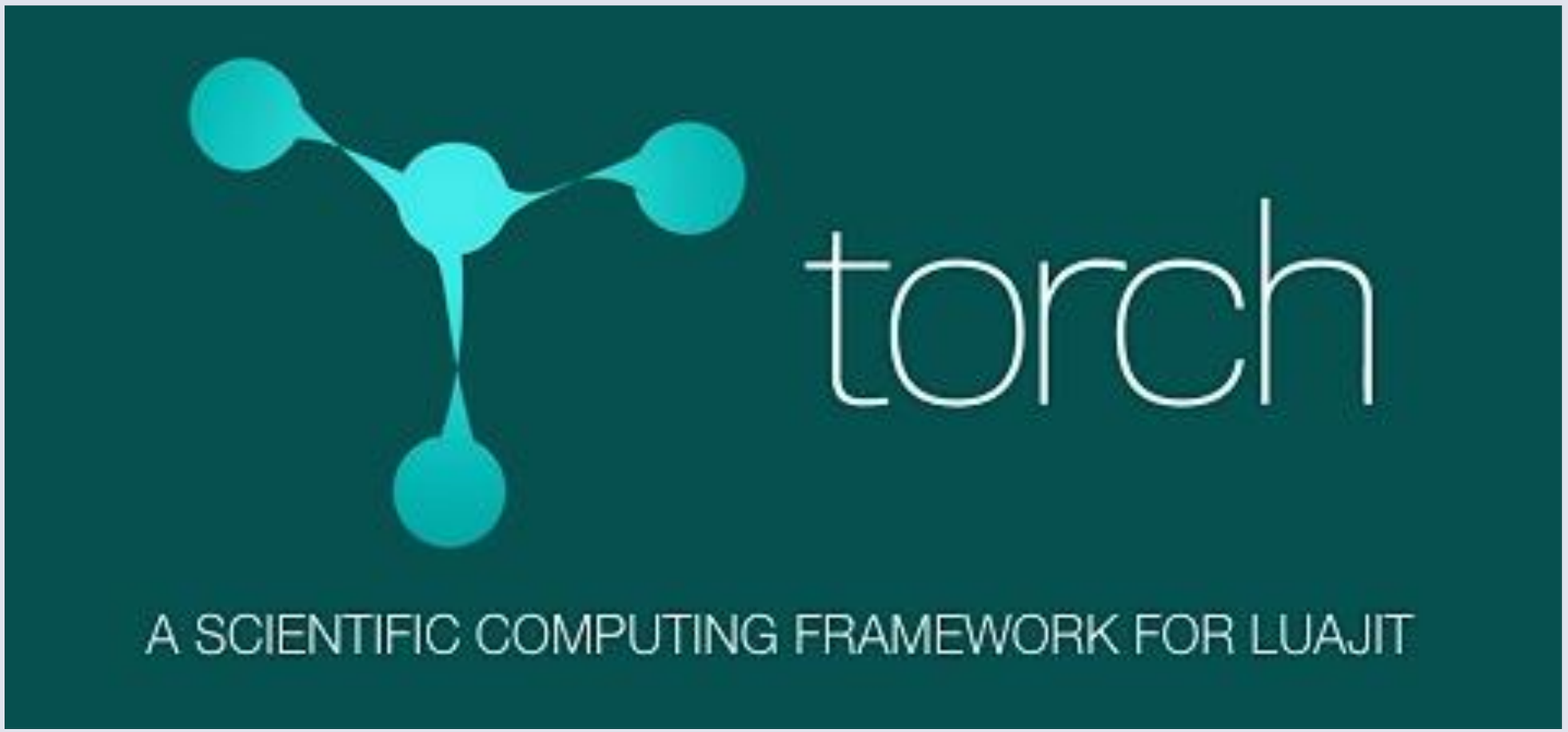


Strategies for Parallelism: Model Parallel

- Exploits parallelism inherent in each layer
- When fusing columns, communication proportional to layer's output size



Productive Deep Learning with Torch7



- **https://github.com/torch/torch7**

Data Parallel in not too much lua

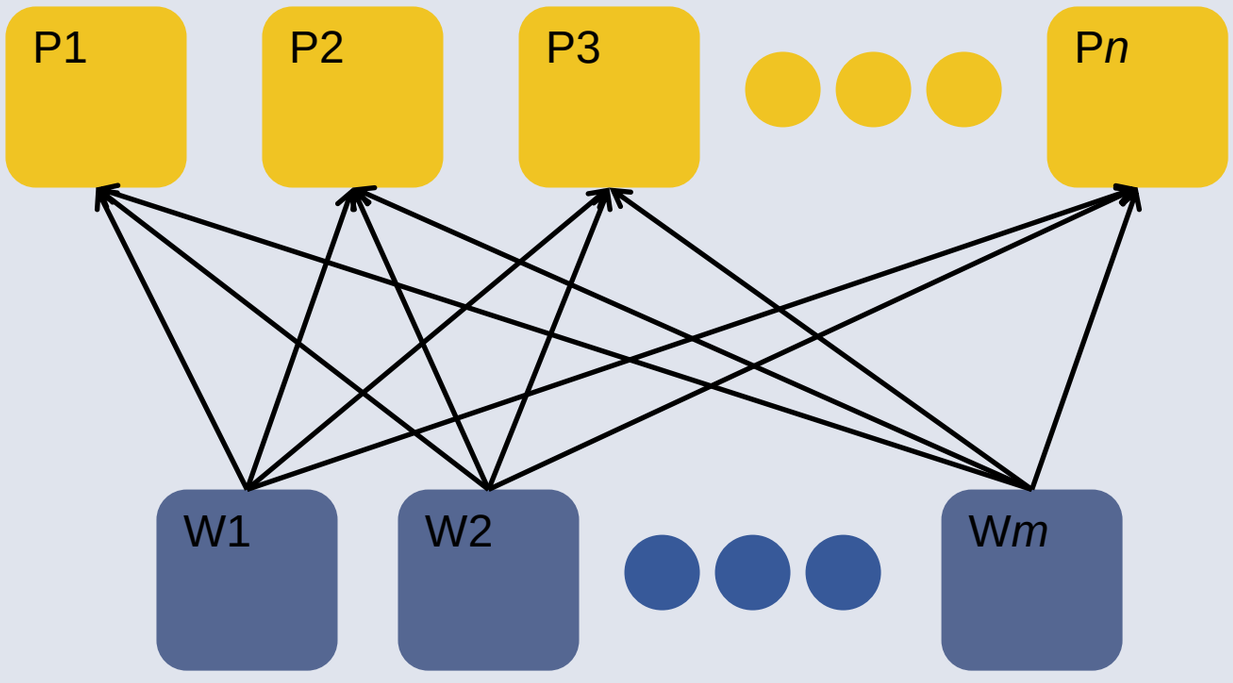
```
local features = nn.Sequential()
features:add(nn.Linear(100, 100))
features:add(nn.Threshold())
features:add(nn.Linear(100, 100))
features:add(nn.Threshold())
```

```
local model = nn.DataParallel(1)
model:add(features)
model:add(features:clone())
```

Distributed Deep Learning

- Still wide open area
- Data parallel, model parallel, asynchrony
- More questions than answers

Parameter Server Architecture



Dean, Jeffrey, et al. "Large scale distributed deep networks." Advances in Neural Information Processing Systems. 2012.

Consensus in Sensor Networks

- Problem: get *n* communicating processes to agree on a point in high-dimensional space
- Distributed training algorithms share same goal

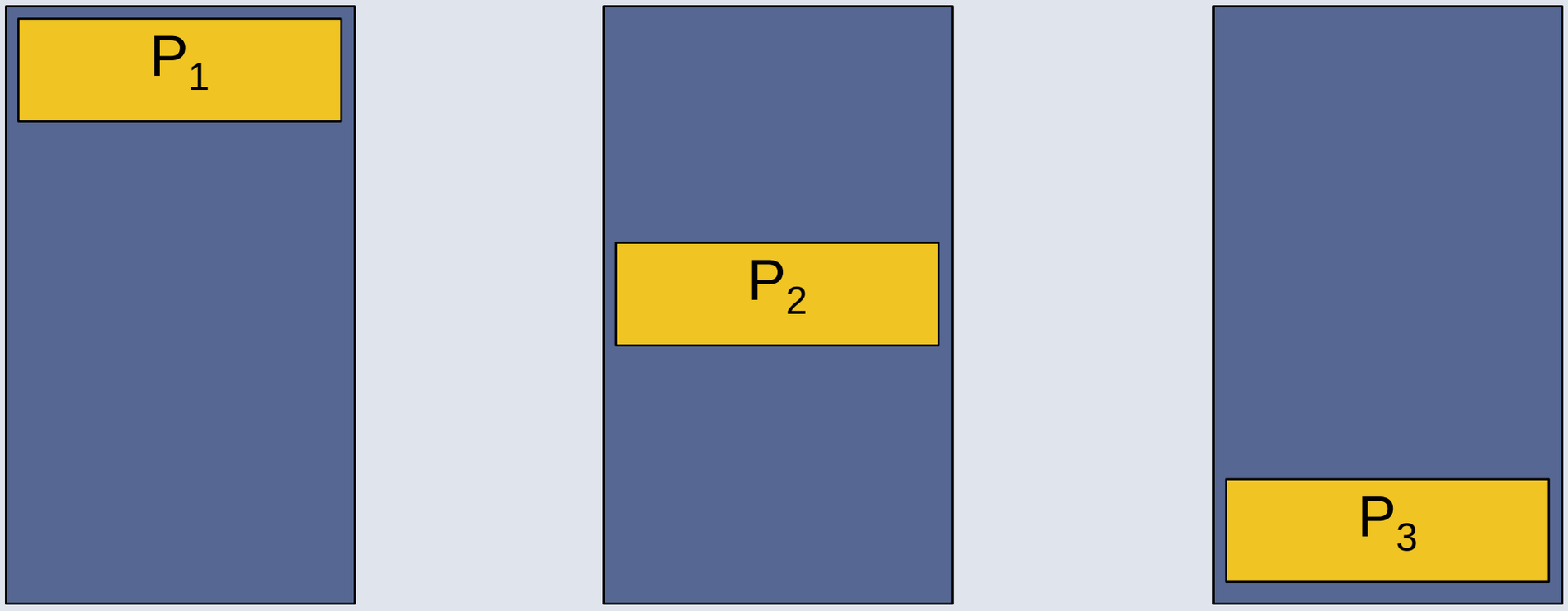
Kar, Soummya, and José MF Moura. "Distributed consensus algorithms in sensor networks with imperfect communication: Link failures and channel noise." Signal Processing, IEEE Transactions on 57.1 (2009): 355-369.

Tensor DSM?

- What if workers are param servers?

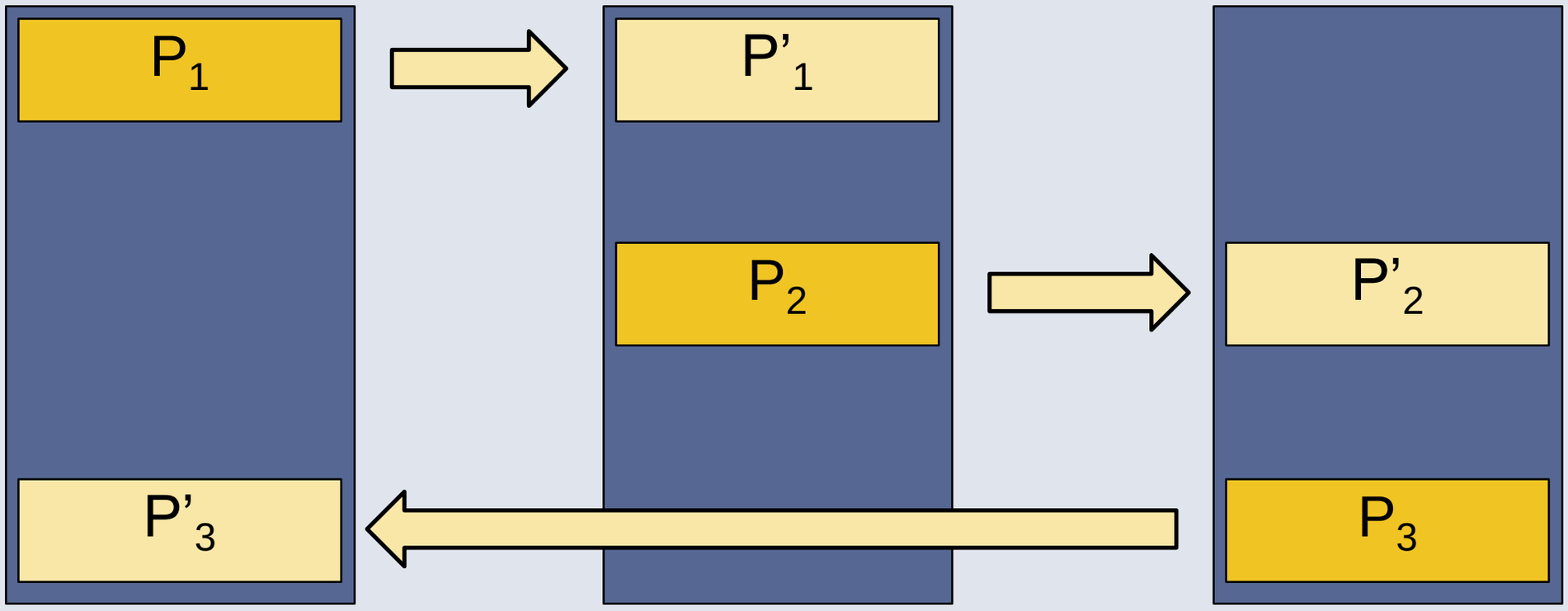
Tensor DSM

Shard Parameters



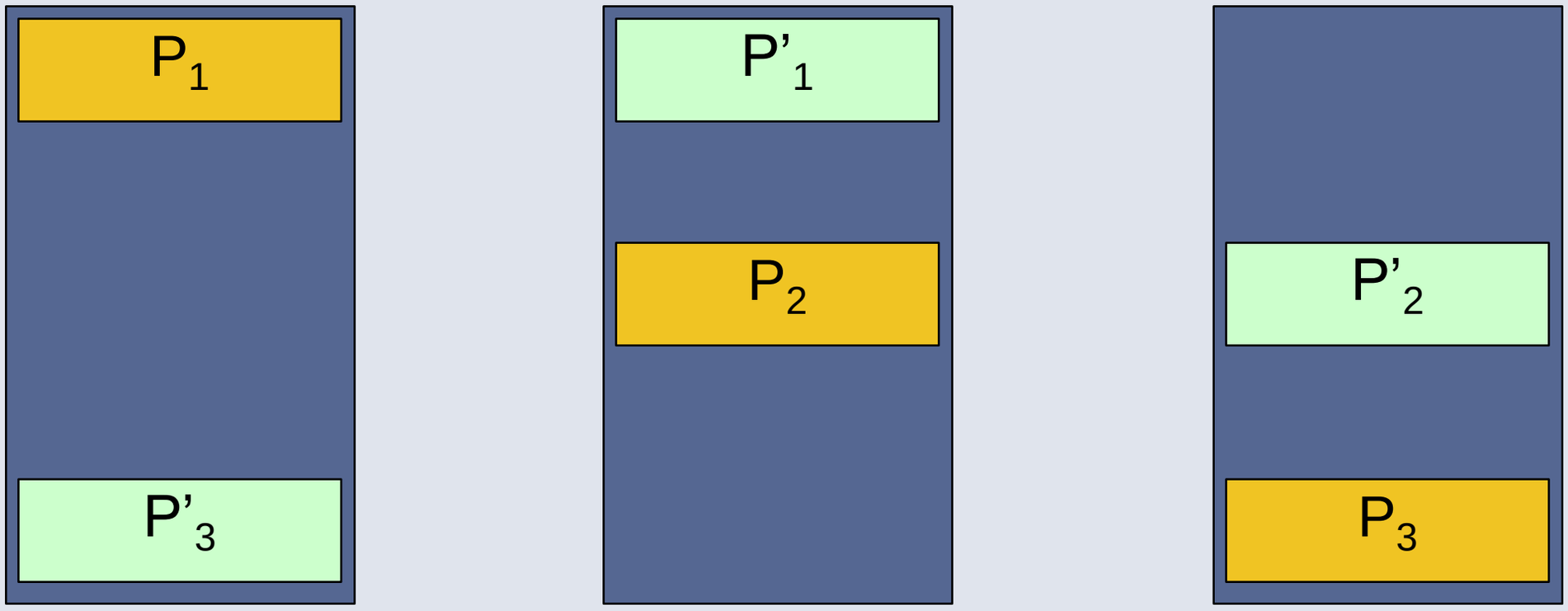
Tensor DSM

Fetch local copies



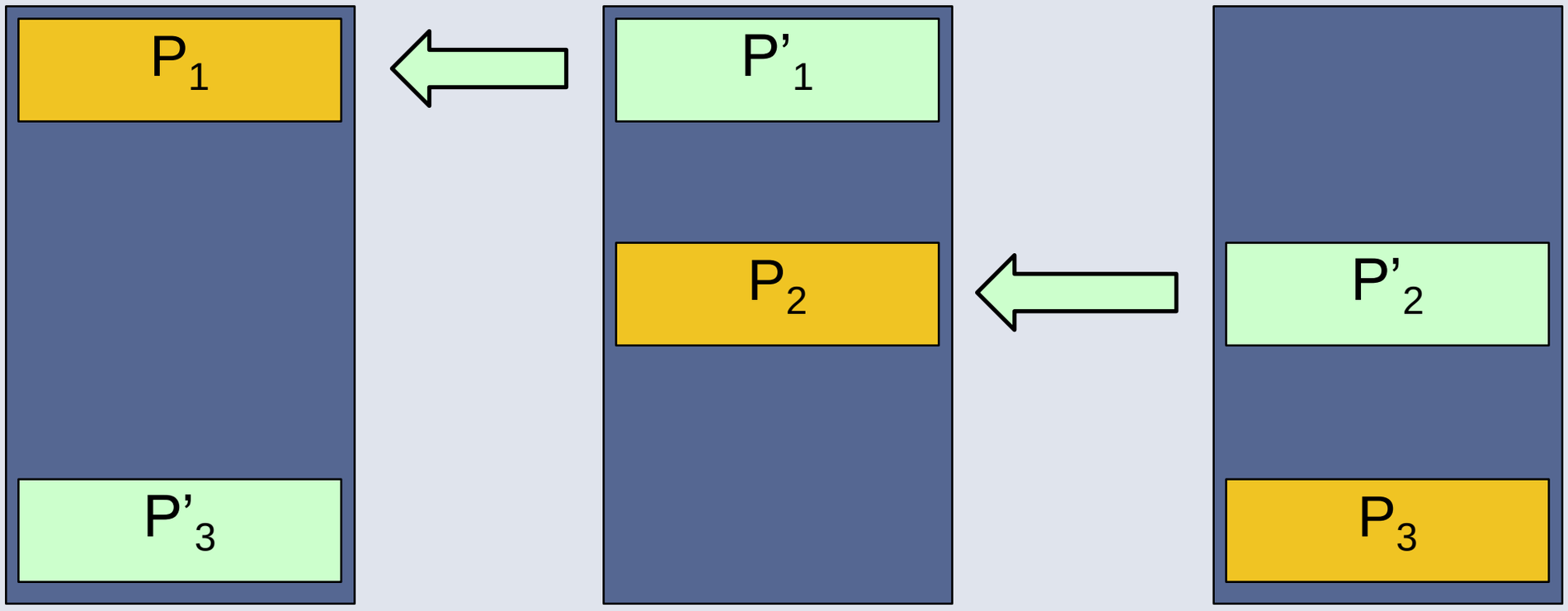
Tensor DSM

Update local copies



Tensor DSM

Update global copies



Tensor DSM

- Not entirely hypothetical…
- But work still in progress
- Stay tuned

Conclusions

- Deep learning is here to stay
- Scaling out training is still an open problem
- Opportunities for innovation
  - Training algorithms
  - Distributed architectures
  - Software

facebook