Distributed DeepLearning at Scale

Soumith Chintala

Facebook AI Research

GTC DC 2016
Washington DC

Overview

- Deep Learning Research at Facebook
- Deep Learning at scale

Deep Learning Research at Facebook AI Research

Image Intelligence: Classification

Image Intelligence

Language Translation from Visual Learning

English	French	English	French
oas	oea	uzbekistan	ouzbekistan
infrared	infrarouge	mushroom	champignons
tomatoes	tomates	filmed	serveur
bookshop	librairie	mauritania	mauritanie
server	apocalyptique	pencils	crayons

Learning Visual Features from Large Weakly Supervised Data

Armand Joulin, Laurens van der Maaten, Allan Jabri, Nicolas Vasilache

Image Intelligence : Detection

Image Intelligence : Detection

Image Intelligence : Detection

Image Intelligence : Detection

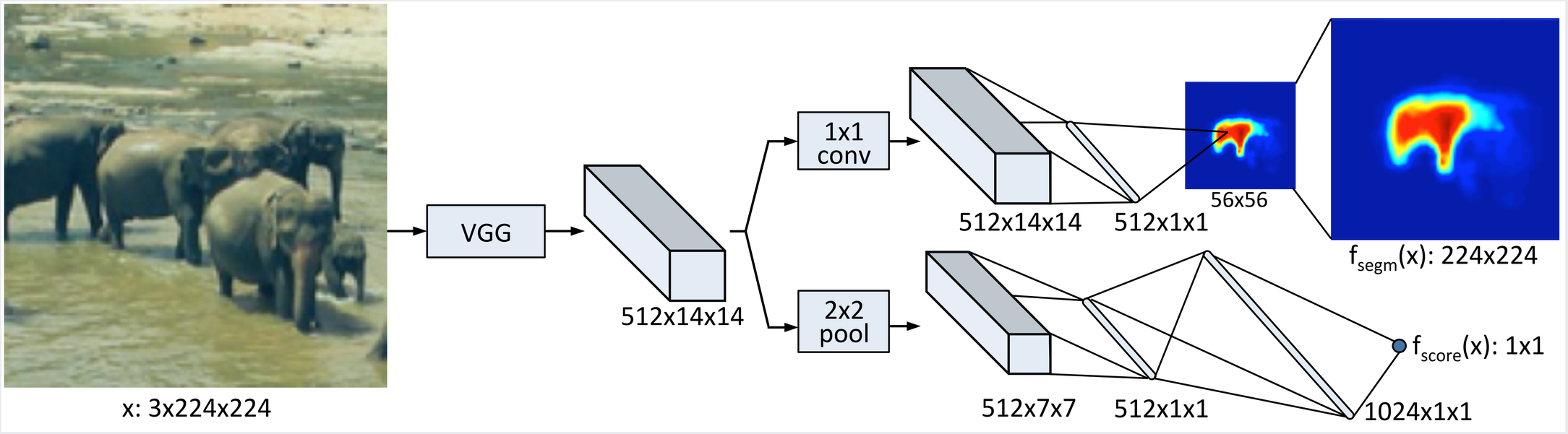
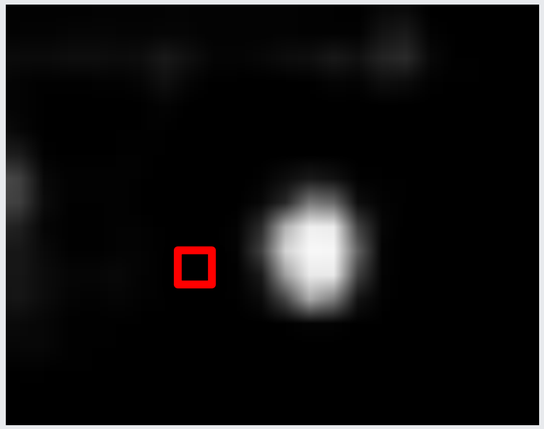


Image Intelligence : Detection

image

scores



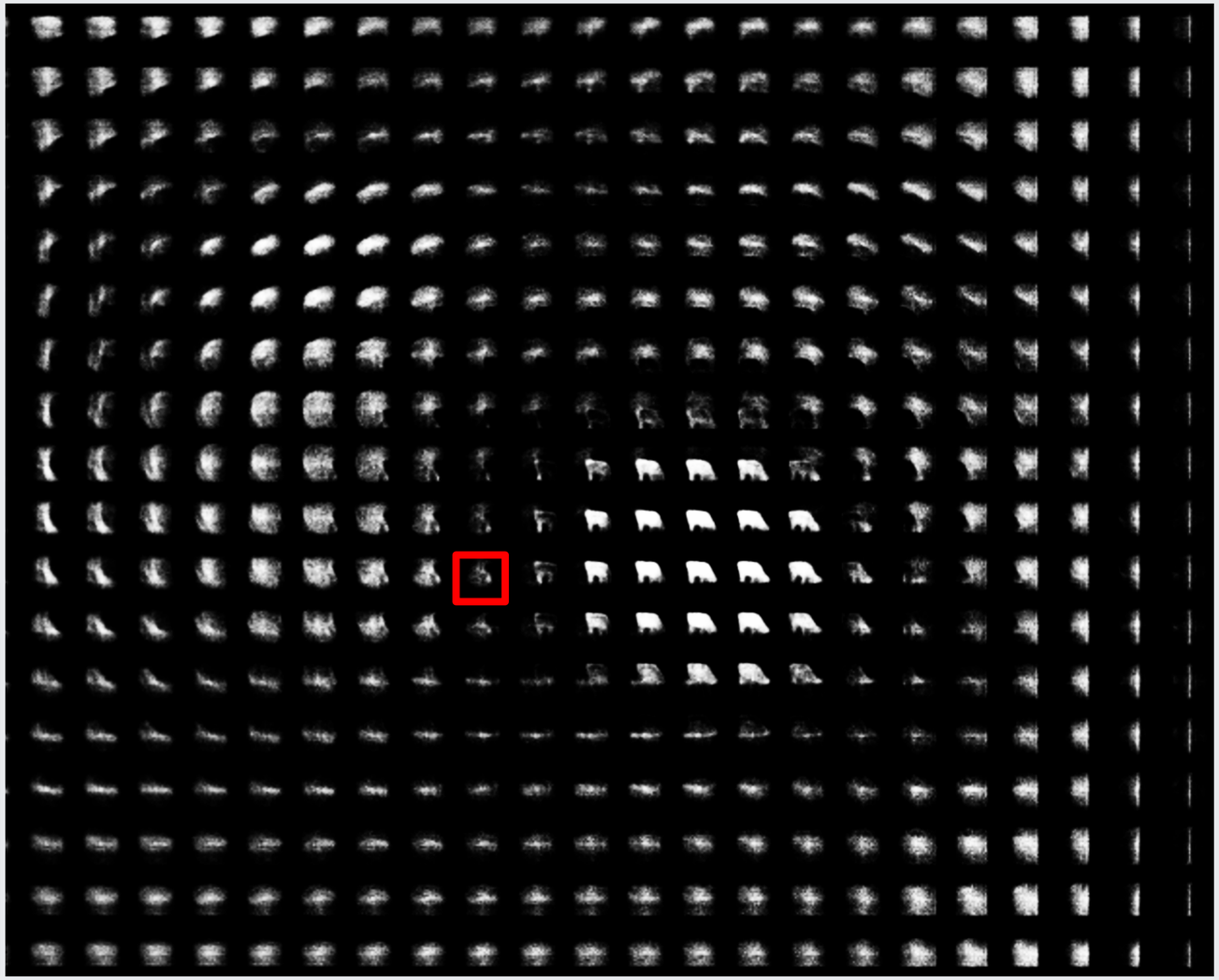
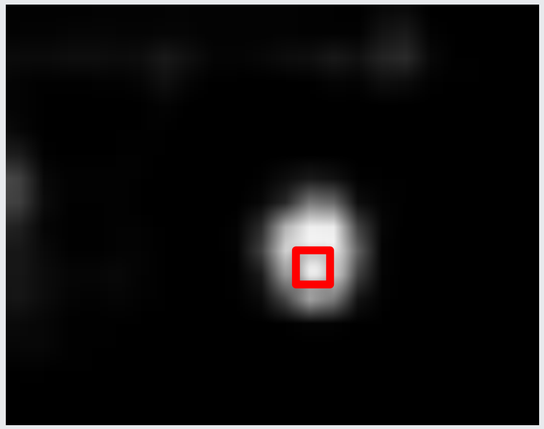


Image Intelligence : Detection

image

scores



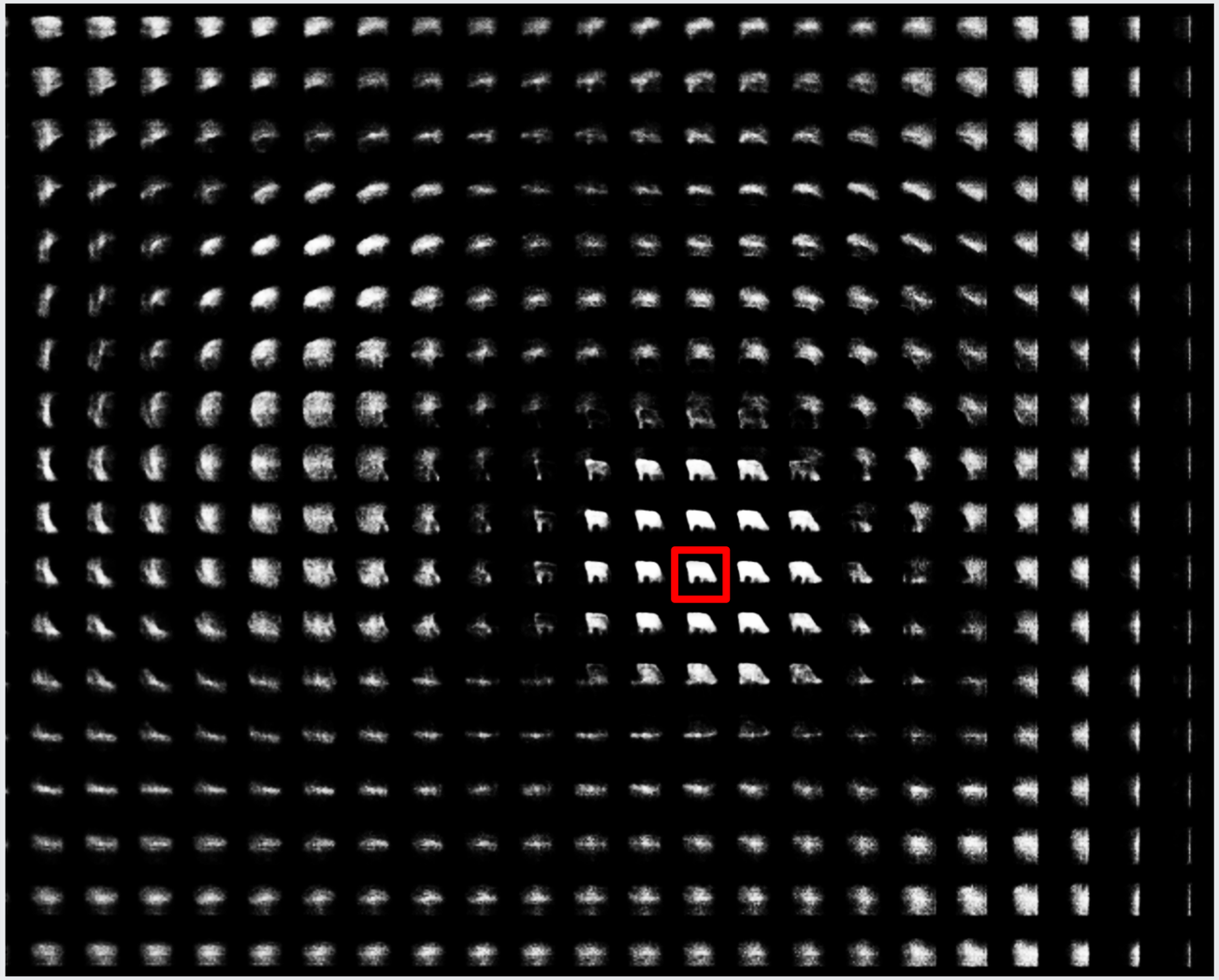


Image Intelligence : Detection

Image Intelligence : Detection

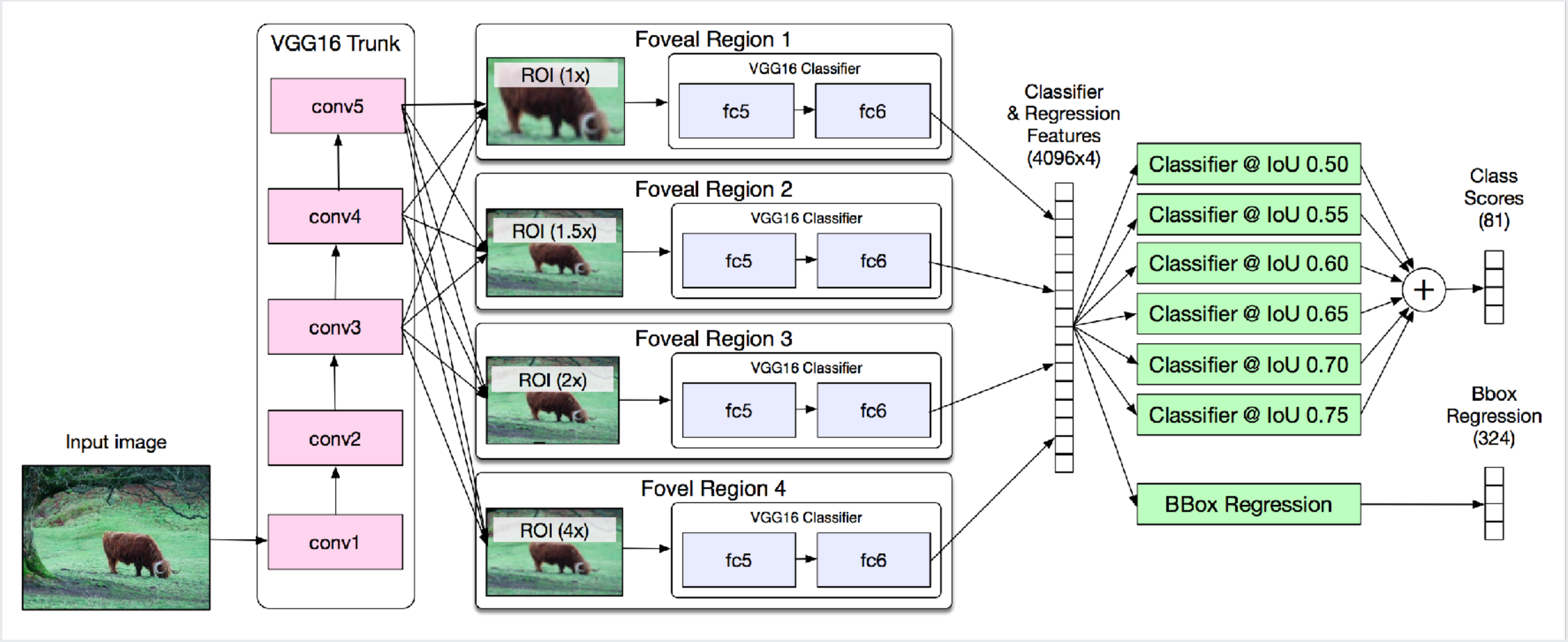


Image Intelligence : Detection

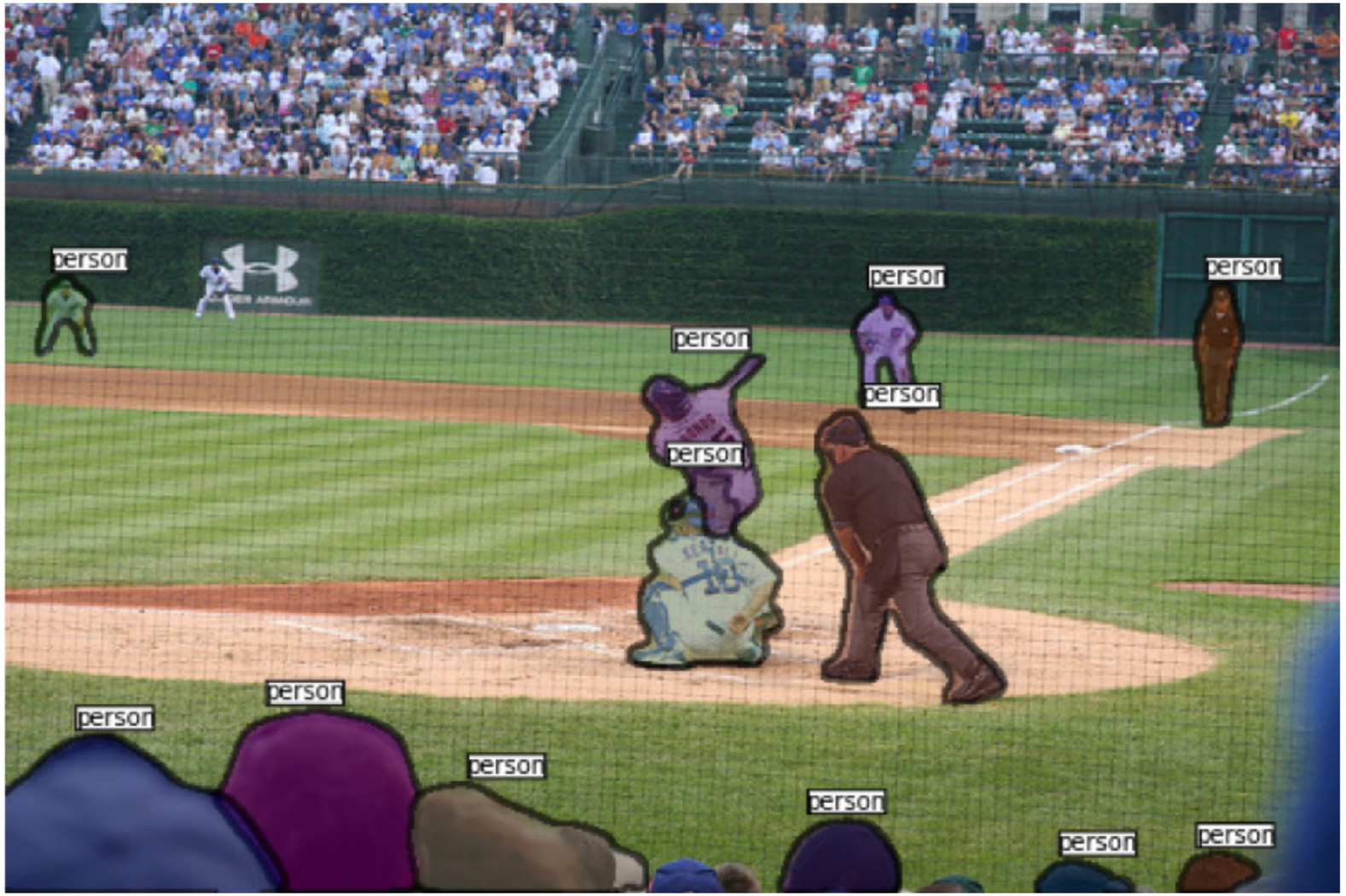


Image Intelligence

https://code.facebook.com/posts/accessibility/

Video Intelligence

Image and Video Generation

Predicting the Future

Natural Language Understanding

chatbots, personal assistants

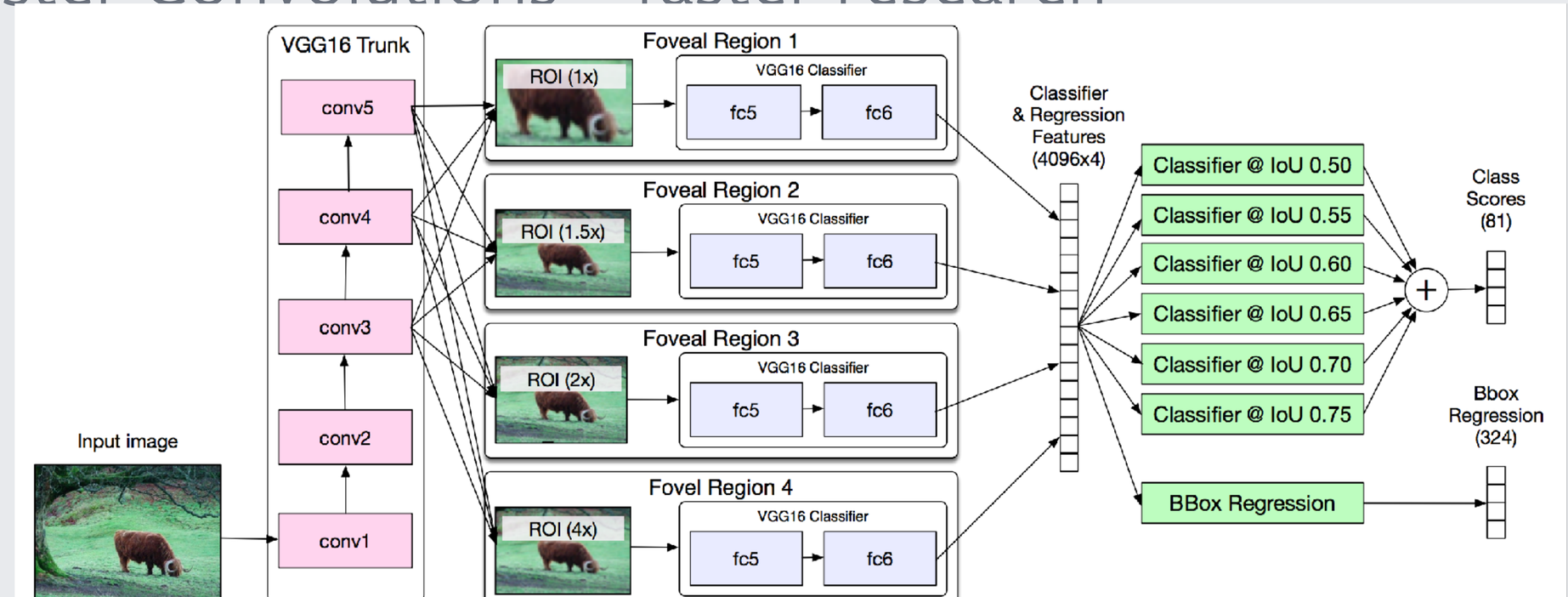
- Memory networks
- Language Translation
- Reading, Writing and answering Questions

Deep Learning at Scale

Deep Learning at Scale

GPU-powered Convolution Neural Networks

- Convolutions, GEMM take all the time
- Massive amounts of exploitable parallelism
  - Faster Convolutions = faster research

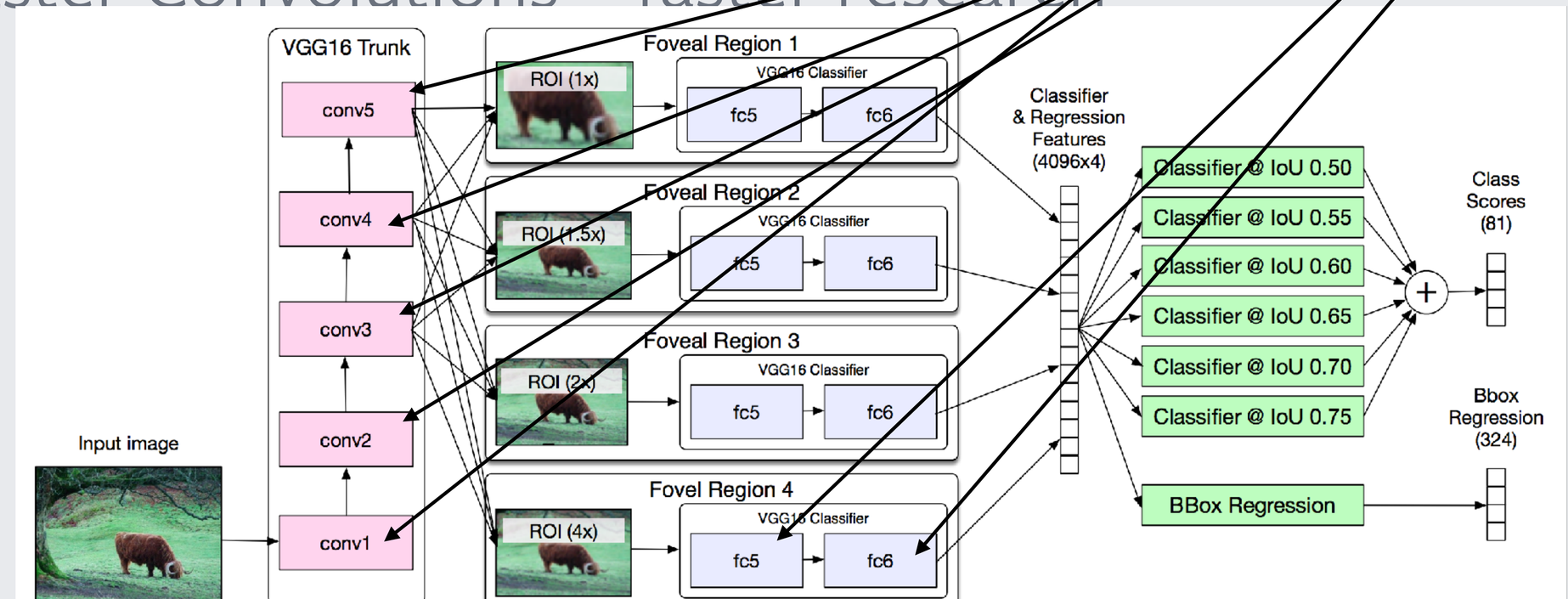


Deep Learning at Scale

GPU-powered Convolution Neural Networks

Massively parallel

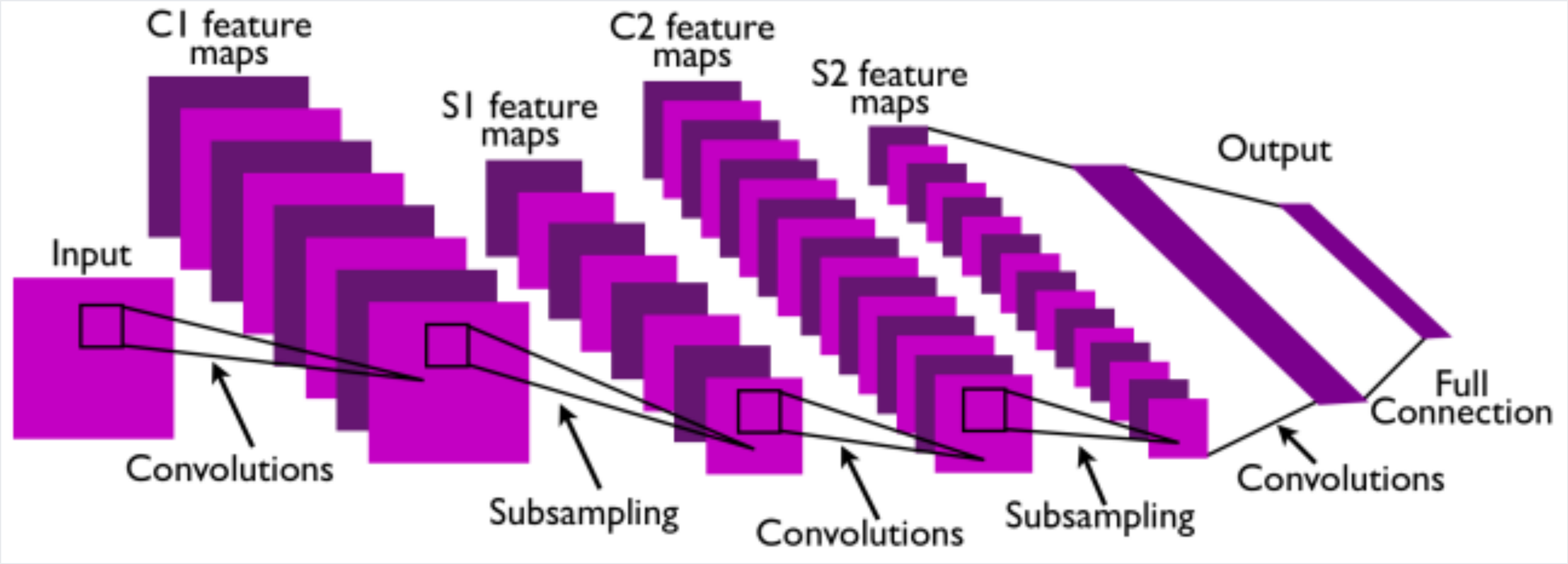
- Convolutions, GEMM take all the time
- Massive amounts of exploitable parallelism
  - Faster Convolutions = faster research



Deep Learning at Scale

GPU-powered Convolution Neural Networks

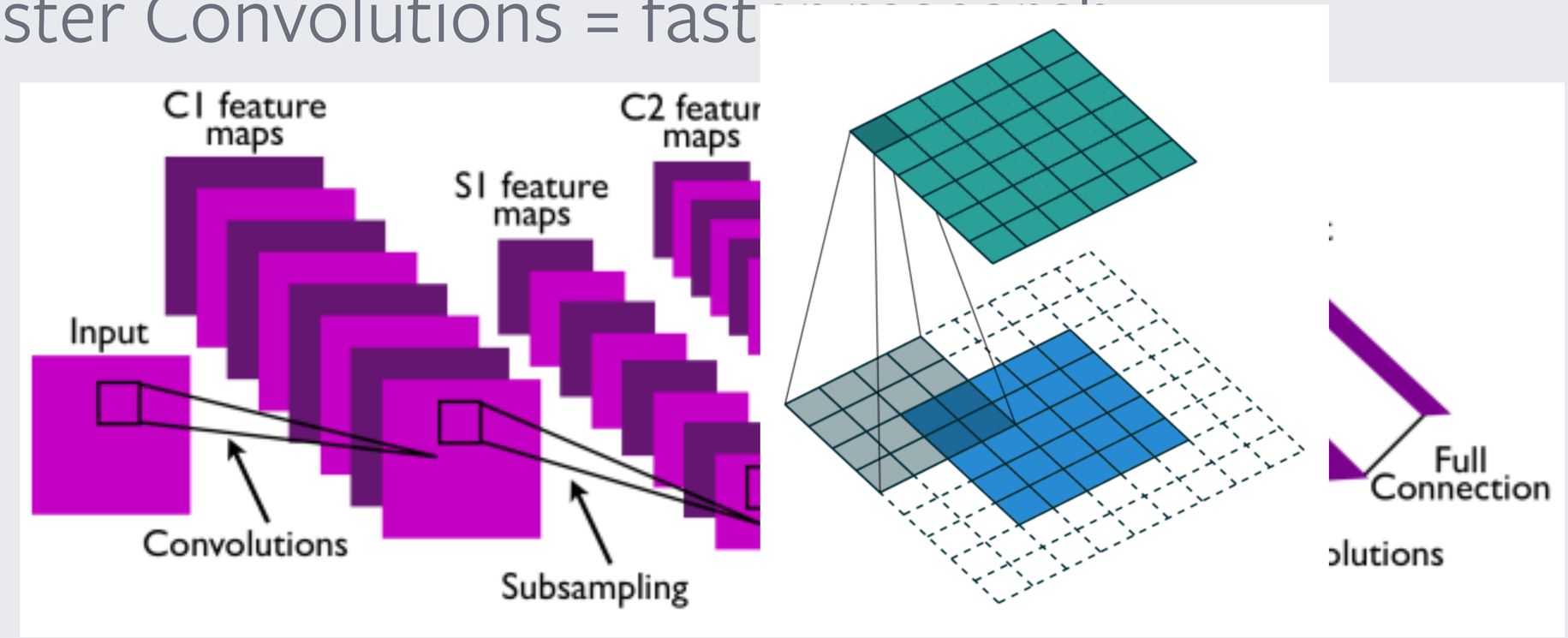
- Convolutions, GEMM take all the time
- Massive amounts of exploitable parallelism
  - Faster Convolutions = faster research



Deep Learning at Scale

GPU-powered Convolution Neural Networks

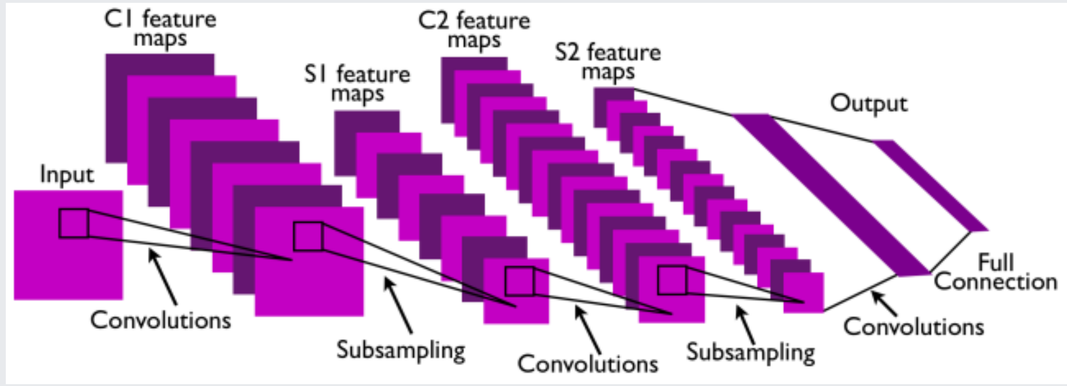
- Convolutions, GEMM take all the time
- Massive amounts of exploitable parallelism
  - Faster Convolutions = fast



Deep Learning at Scale

Parallelism to exploit

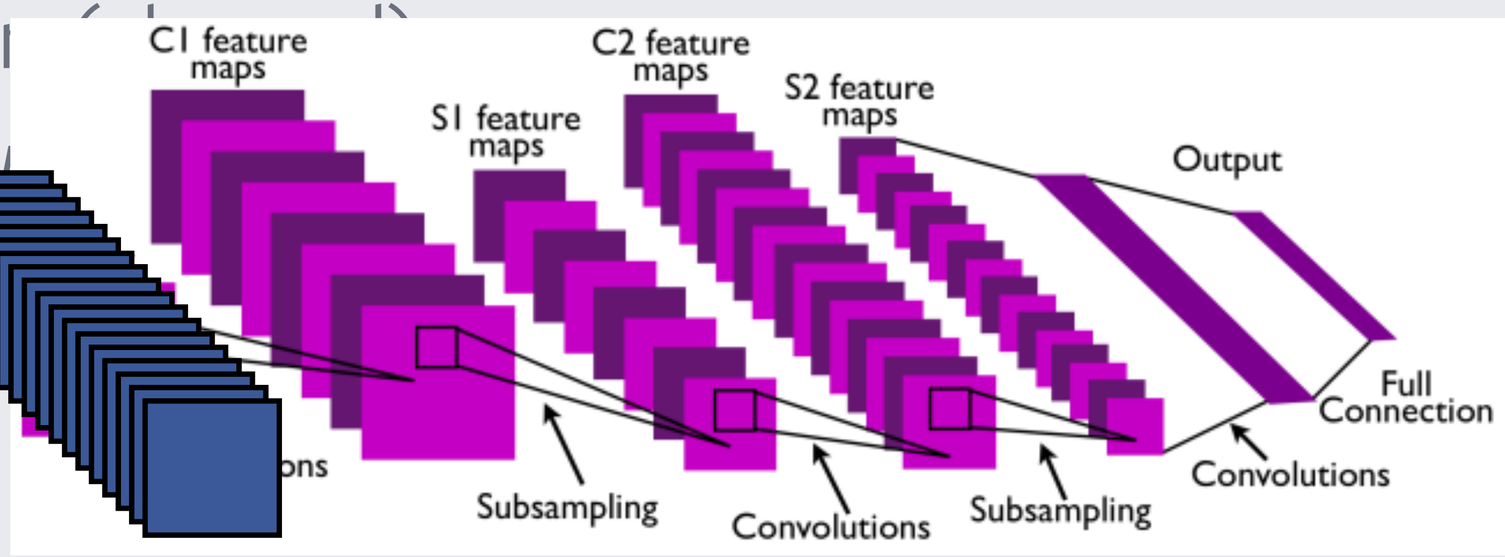
- for epoch in (total_epochs):
  - *for* image in (dataset):
    - *for* channel in (image):
      - *for* row in (channel):
        - *for column* in (row):
          - multiply-accumulate with kernel



Deep Learning at Scale

Parallelism to exploit

- for epoch in (total_epochs):
  - *for* 128_images in (dataset): **Mini-batch parallelism**
    - *for* channel in (image):
      - *for* row i
        - *for* colu
          - mul

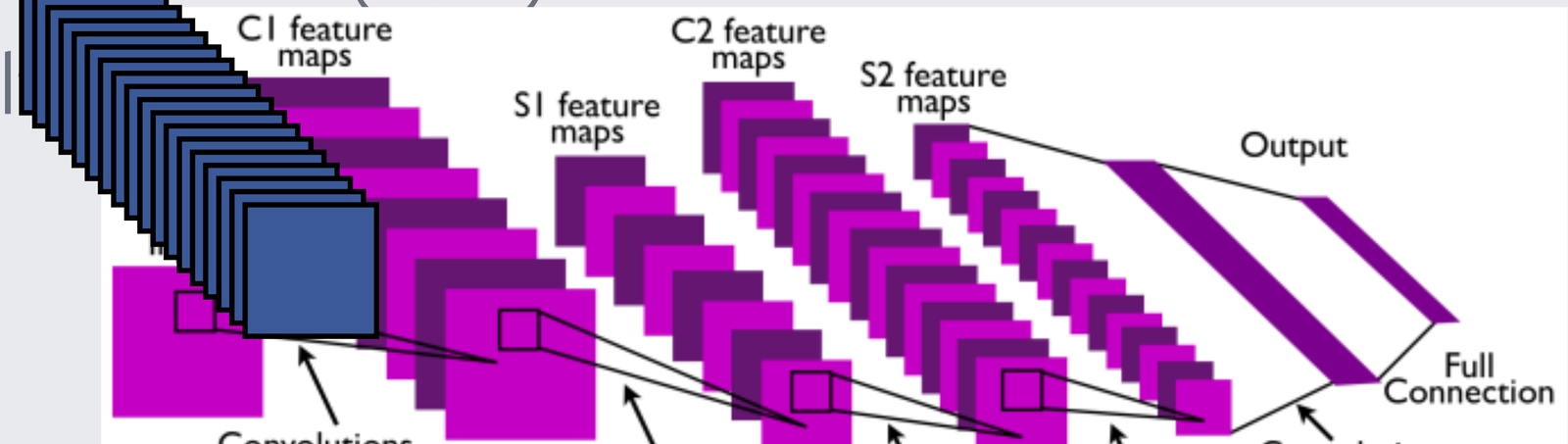


Deep Learning at Scale

Parallelism to exploit

- for epoch in (total_epochs):
  - *for* 128_images in (dataset):
    - *for* channel in (image):
      - *for* row in (channel):
        - *for column* in (row):
          - mul

Parallelize over each GPU SM



Deep Learning at Scale

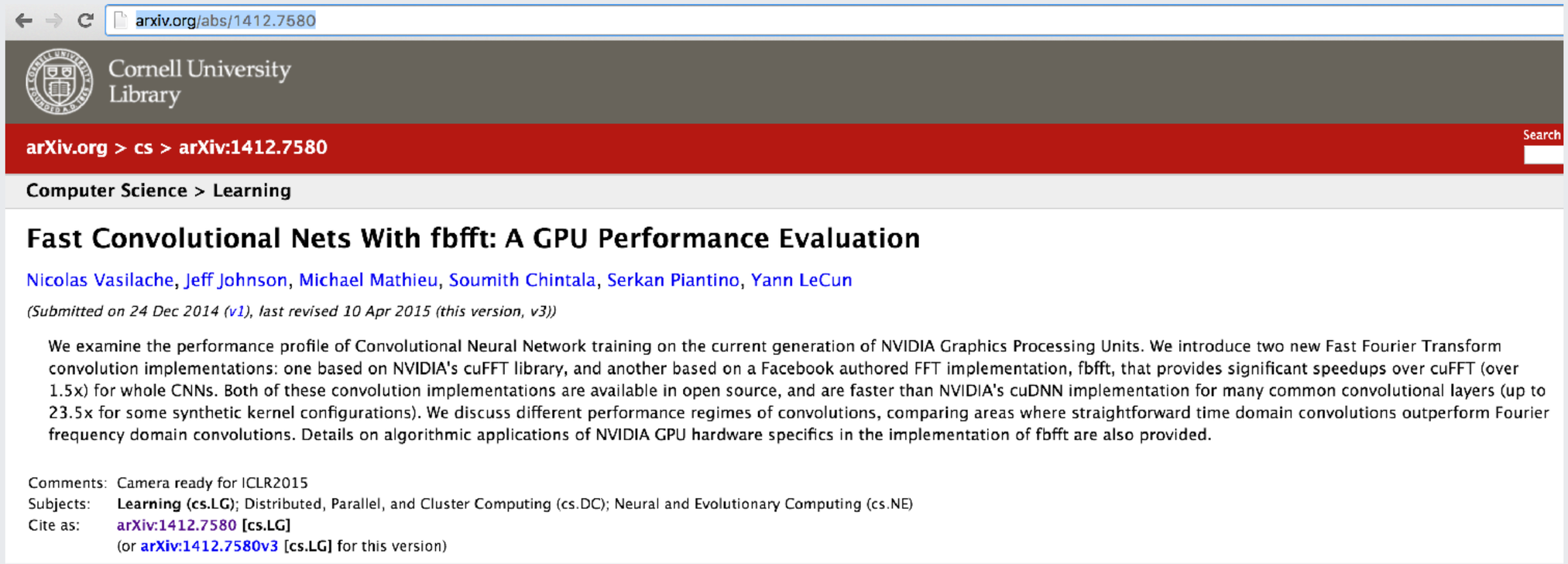
Parallelism to exploit

- for epoch in (total_epochs):
  - *for* 128_images in (dataset):
    - *for* channel in (image):
      - *for* row in (channel):
        - for row_slice in (row):
          - *for column* in (row_slice):
            - for cloumn_slice in (column):
              - multiply-accumulate with kernel

Tiling over image sub-regions (GPU-friendly)

Deep Learning at Scale

GPU-powered Convolution Neural Networks



arxiv.org/abs/1412.7580

Cornell University Library

arXiv.org > cs > arXiv:1412.7580

Search

Computer Science > Learning

Fast Convolutional Nets With fbfft: A GPU Performance Evaluation

Nicolas Vasilache, Jeff Johnson, Michael Mathieu, Soumith Chintala, Serkan Piantino, Yann LeCun

(Submitted on 24 Dec 2014 (v1), last revised 10 Apr 2015 (this version, v3))

We examine the performance profile of Convolutional Neural Network training on the current generation of NVIDIA Graphics Processing Units. We introduce two new Fast Fourier Transform convolution implementations: one based on NVIDIA's cuFFT library, and another based on a Facebook authored FFT implementation, fbfft, that provides significant speedups over cuFFT (over 1.5x) for whole CNNs. Both of these convolution implementations are available in open source, and are faster than NVIDIA's cuDNN implementation for many common convolutional layers (up to 23.5x for some synthetic kernel configurations). We discuss different performance regimes of convolutions, comparing areas where straightforward time domain convolutions outperform Fourier frequency domain convolutions. Details on algorithmic applications of NVIDIA GPU hardware specifics in the implementation of fbfft are also provided.

Comments: Camera ready for ICLR2015
Subjects: **Learning (cs.LG)**; Distributed, Parallel, and Cluster Computing (cs.DC); Neural and Evolutionary Computing (cs.NE)
Cite as: arXiv:1412.7580 **[cs.LG]**
(or **arXiv:1412.7580v3 [cs.LG]** for this version)

Deep Learning at Scale

GPU-powered Convolution Neural Networks

All tiling / mathematical single-GPU optimizations in NVIDIA CuDNN

Deep Learning at Scale

Parallelism to exploit

- for epoch in (total_epochs):
  - *for* 128_images in (dataset):
    - *for* channel in (image):
      - *for* row in (channel):
        - for row_slice in (row):
          - *for column* in (row_slice):
            - for cloumn_slice in (column):
              - multiply-accumulate with kernel

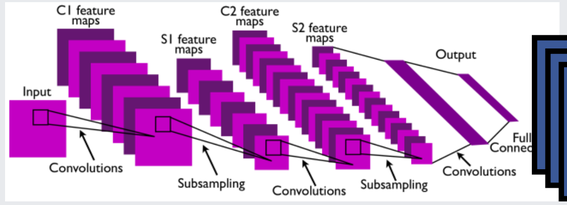
This takes us to multi-machine

Deep Learning at Scale

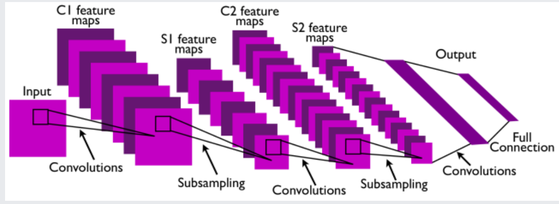
Parallelism to exploit

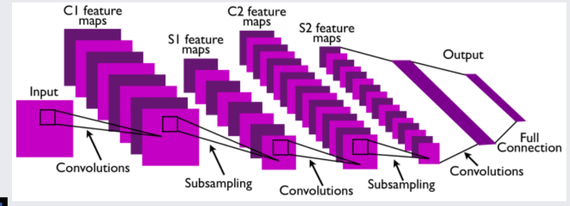
- for epoch in (total_epochs):
  - @over each machine
    - *parallel_for* 128_images in (dataset):
  - synchronize weights

machine 1

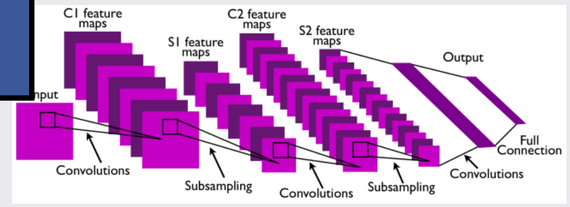


machine 2





machine 3



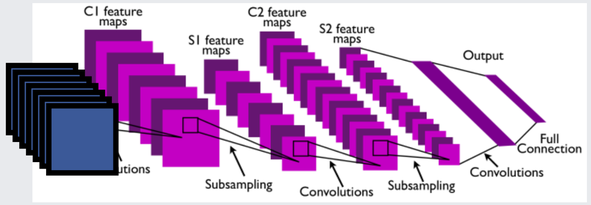
machine 4

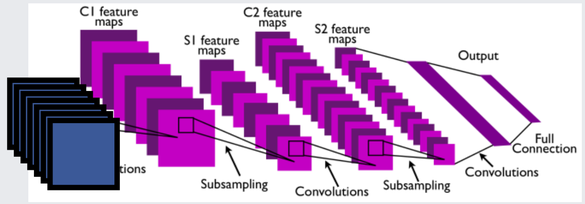
Deep Learning at Scale

Parallelism to exploit

- for epoch in (total_epochs):
  - **@over each machine**
    - ***parallel_for* 128_images in (dataset):**
  - synchronize weights

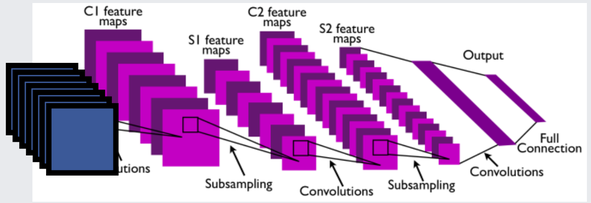
machine 1

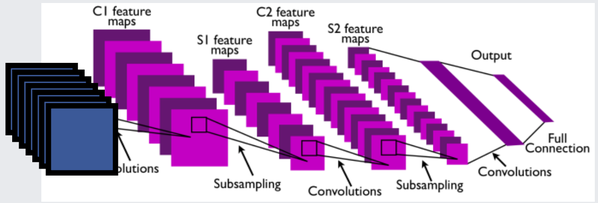




machine 3

machine 2



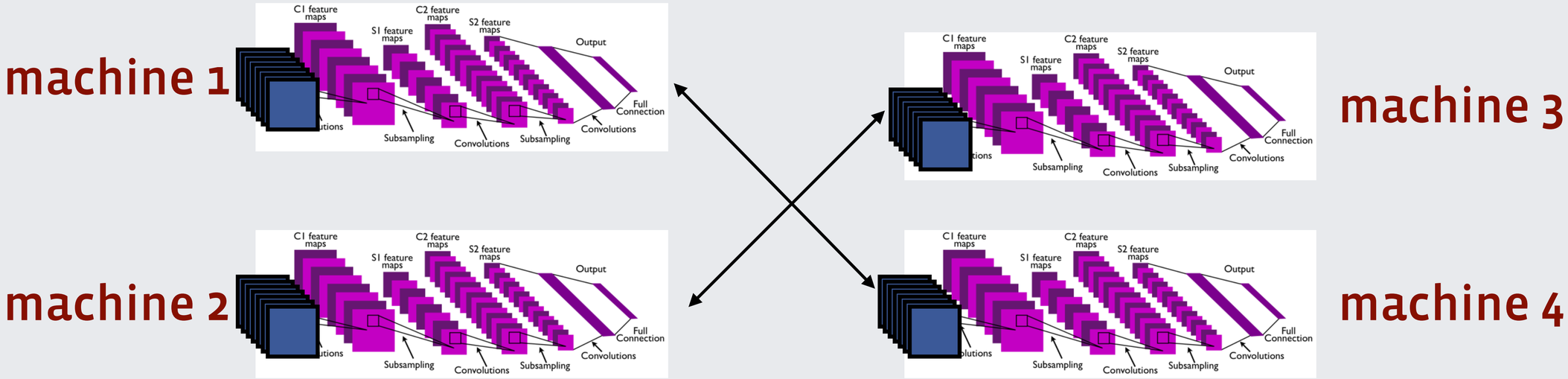


machine 4

Deep Learning at Scale

Parallelism to exploit

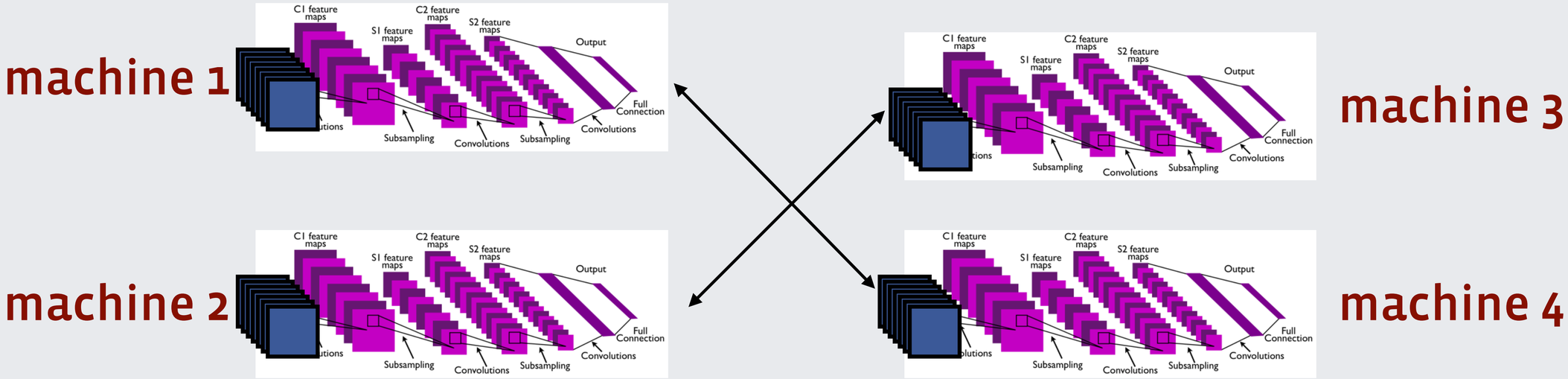
- for epoch in (total_epochs):
  - @over each machine
    - *parallel_for* 128_images in (dataset):
  - **synchronize weights**



Deep Learning at Scale

Parallelism to exploit

- for epoch in (total_epochs):
  - @over each machine
    - *parallel_for* 128_images in (dataset):
  - **synchronize weights** **bottleneck!**



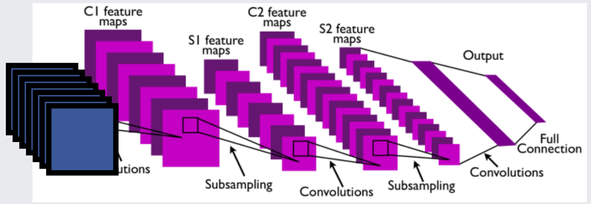
Deep Learning at Scale

Parallelism to exploit

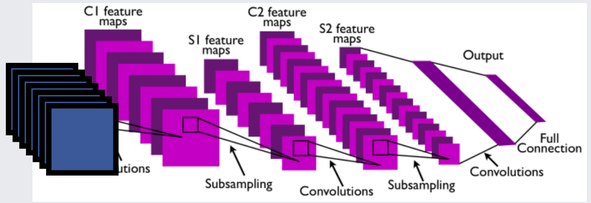
- for epoch in (total_epochs):
  - @over each machine
    - *parallel_for* 128_images in (dataset):
    - **synchronize weights**

what happens if we relax synchronization?

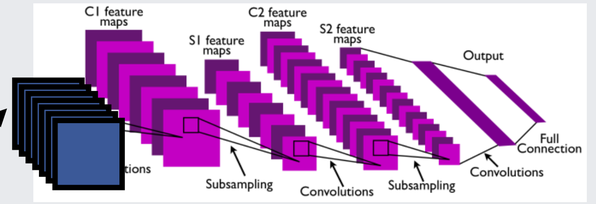
machine 1



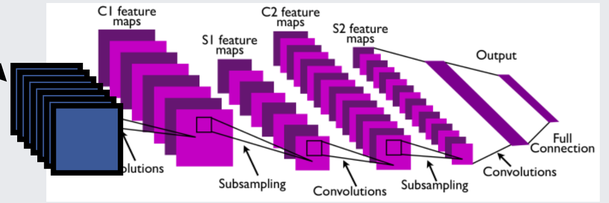
machine 2



machine 3



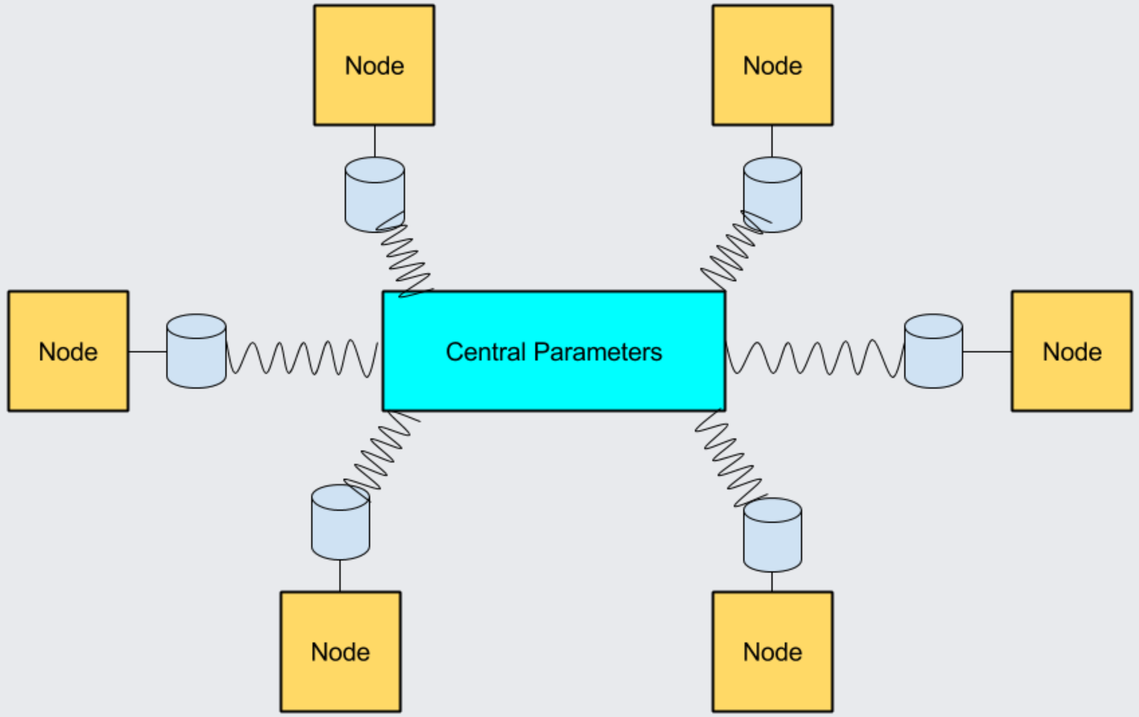
machine 4



Deep Learning at Scale

Multi-Machine Training

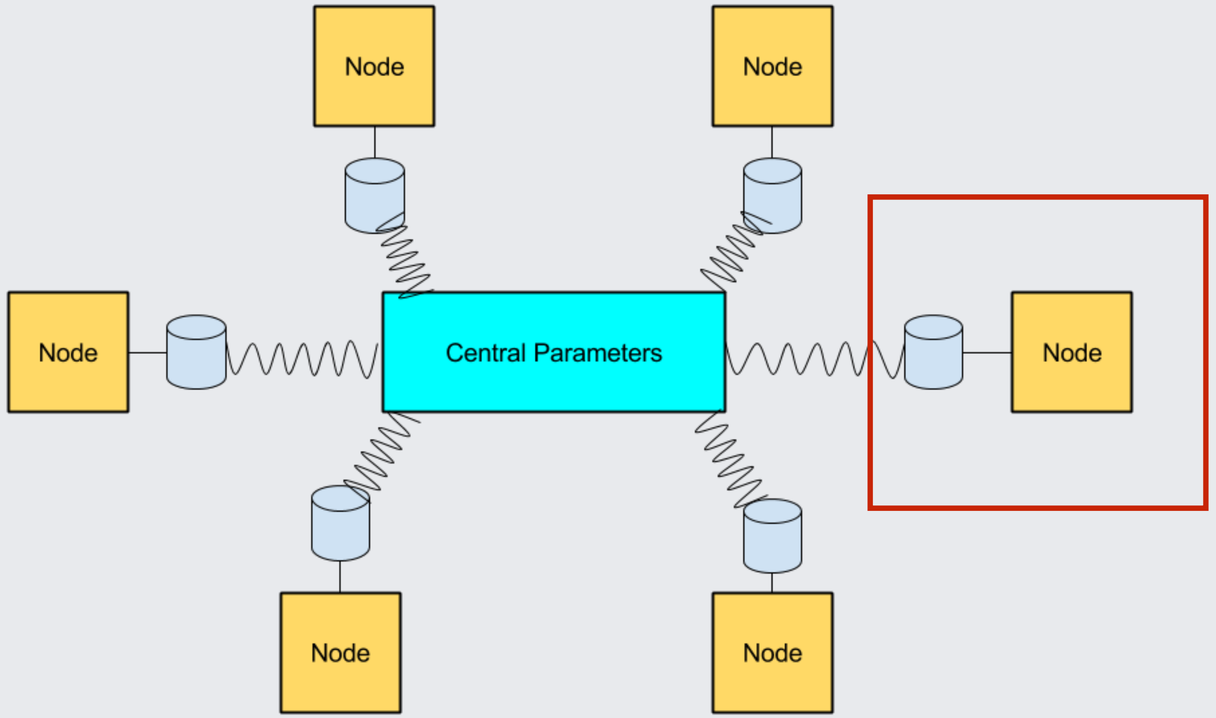
- Elastic Averaging SGD! (Sixin Zhang, Anna Choromanska, Yann LeCun)



Deep Learning at Scale

Multi-Machine Training

- Elastic Averaging SGD!

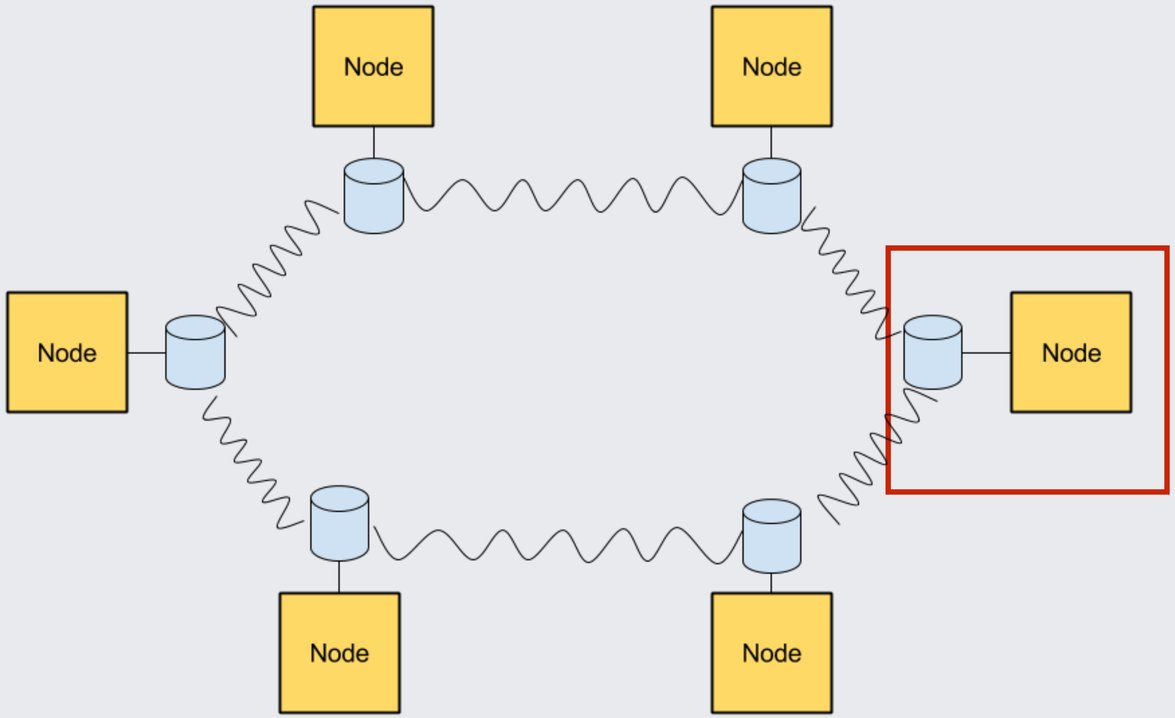


Train synchronously
Occasionally, check with master
Dont go too far from everyone else

Deep Learning at Scale

Multi-Machine Training

- Elastic Averaging SGD!



Train synchronously
Occasionally, check with neighbors
Dont go too far from everyone else

Deep Learning at Scale

Multi-Machine Training

- Elastic Averaging SGD!
  - Empirical speedup of SquareRoot(N)
    - N = number of nodes
  - No communication overhead with pre-fetching
    - 128 GPUs (32 clients * 4 GPUs)

Exploit all the parallelism

https://research.facebook.com/ai

facebook