

Deep Learning at Scale

CSCS Summer School 2021

Henrique Mendonça and Rafael Sarmiento, CSCS

30 July 2021

Deep Learning at Scale

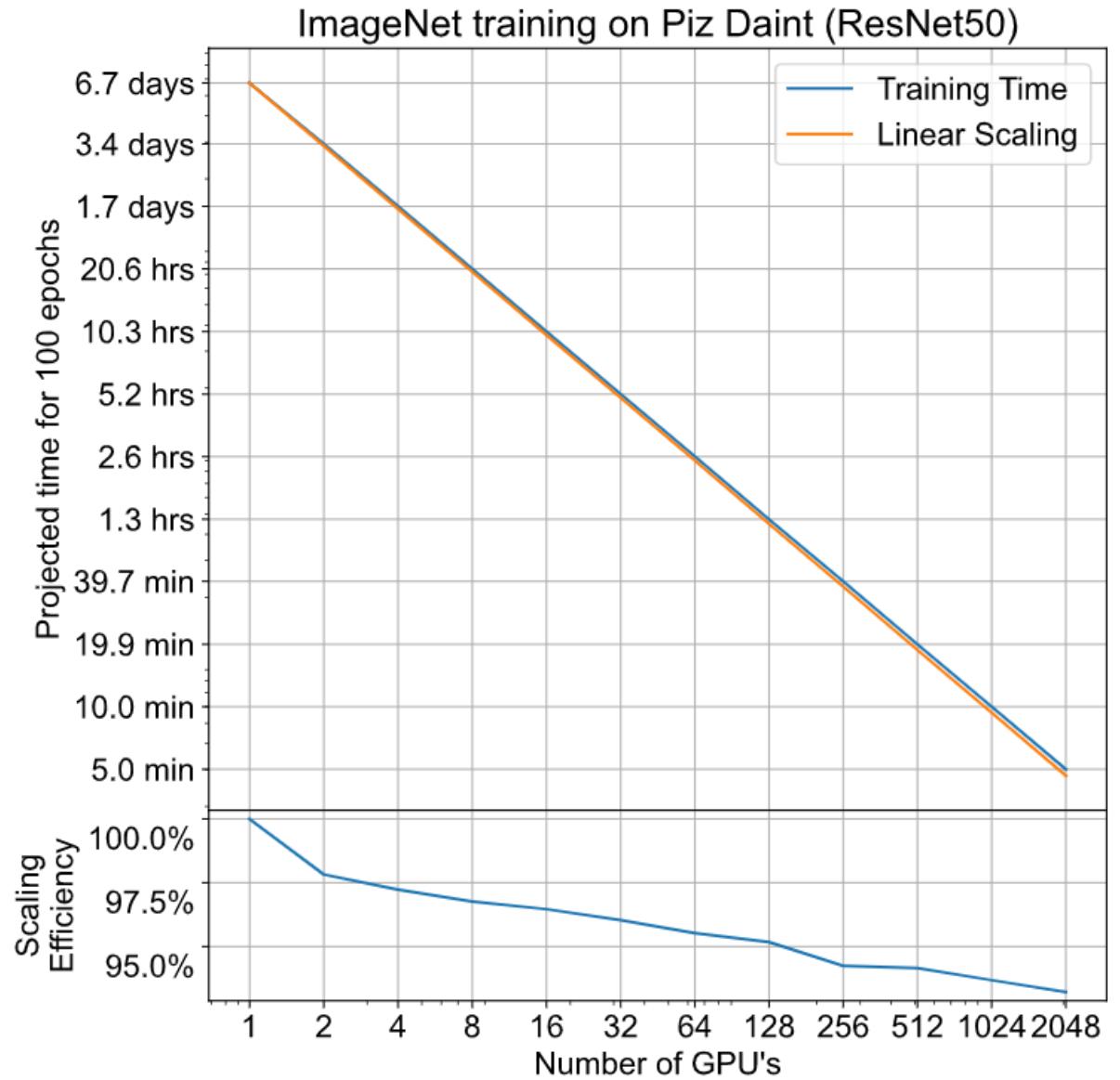
1. Motivation
2. ImageNet use case
 - Dataset and TF records
 - Convolutional Networks
 - Data Augmentation
3. Data parallelism
 - Batch Norm synchronization

Deep Learning at Scale

■ Motivation

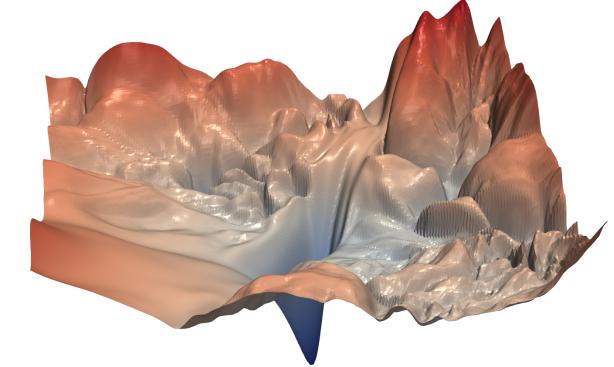
1. Speed:

- Fast experimentation
- Fail fast, try new ideas

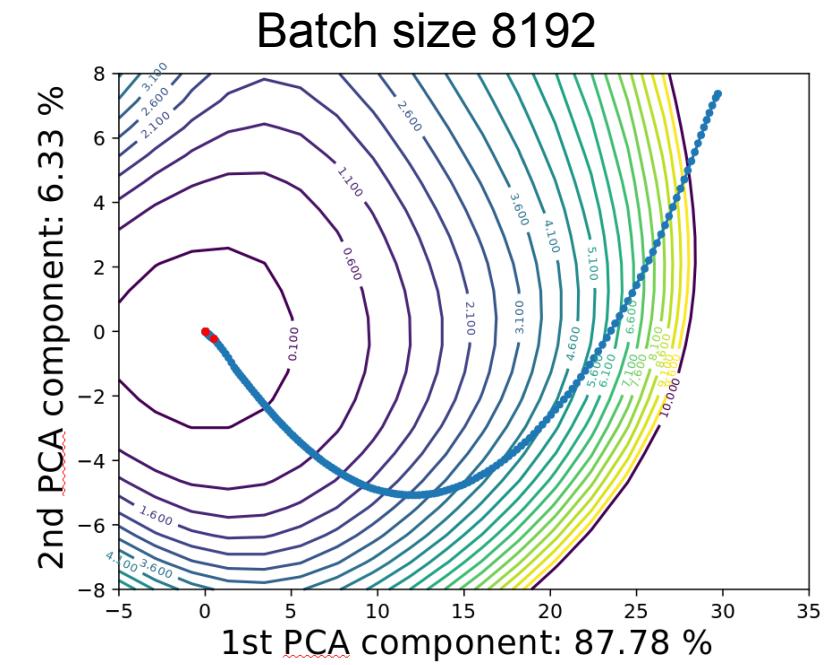
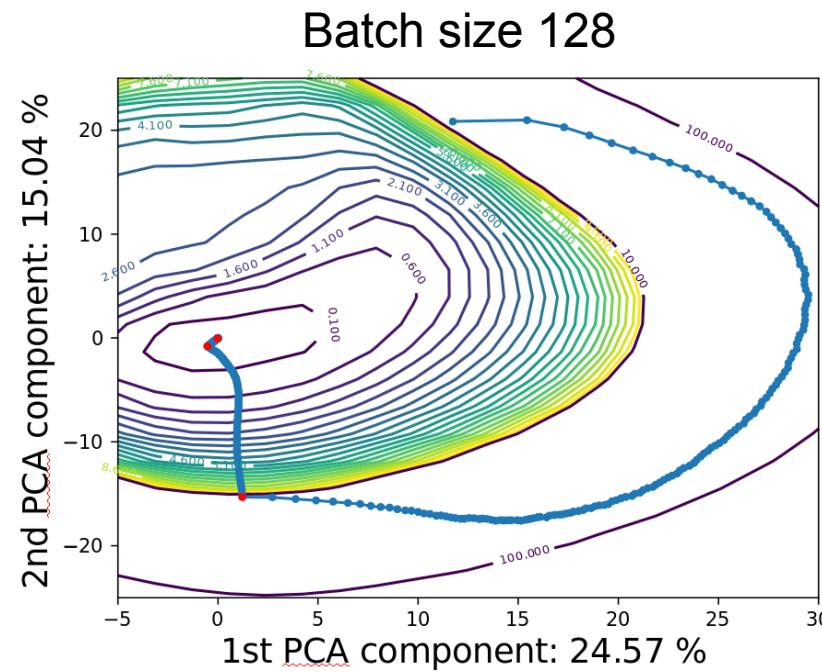


Motivation

- 2. Quality:
 - Allows SGD with larger batch sizes, less noise
 - Search a large hyper-parameter space

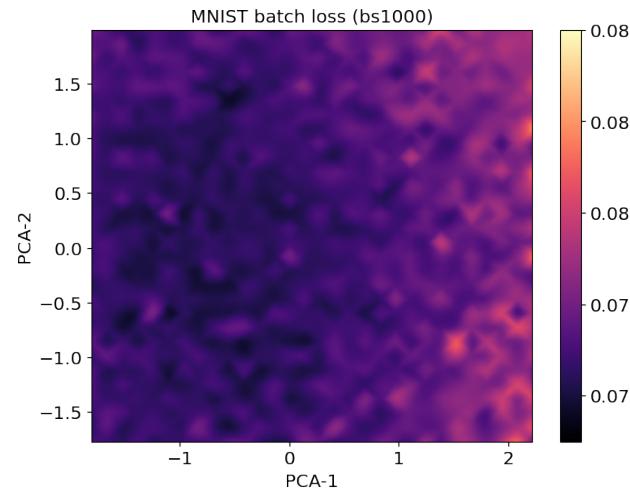
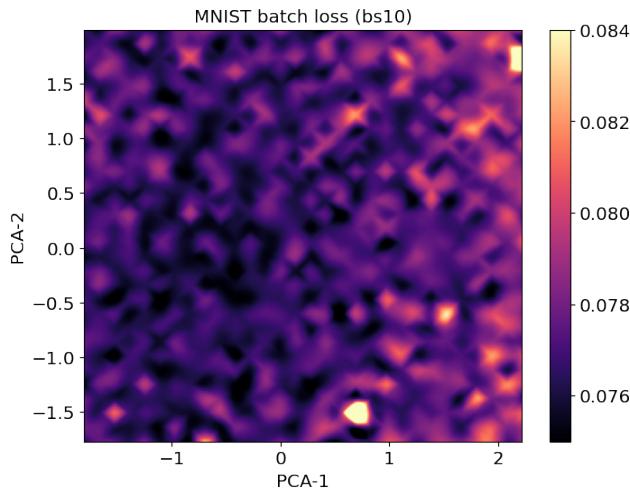


Visualizing the Loss Landscape of Neural Nets
<https://arxiv.org/pdf/1712.09913.pdf>

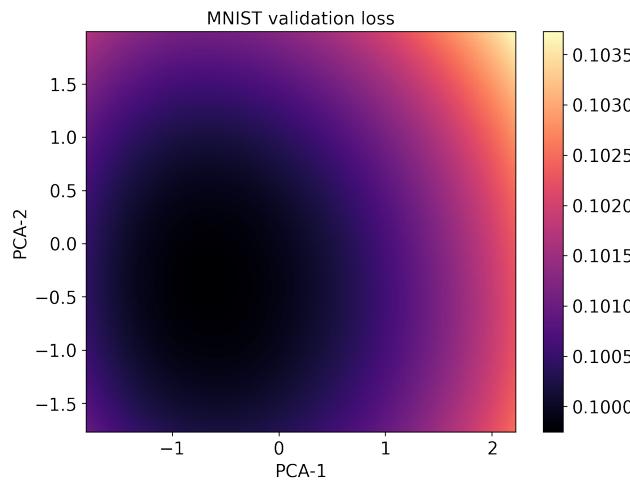
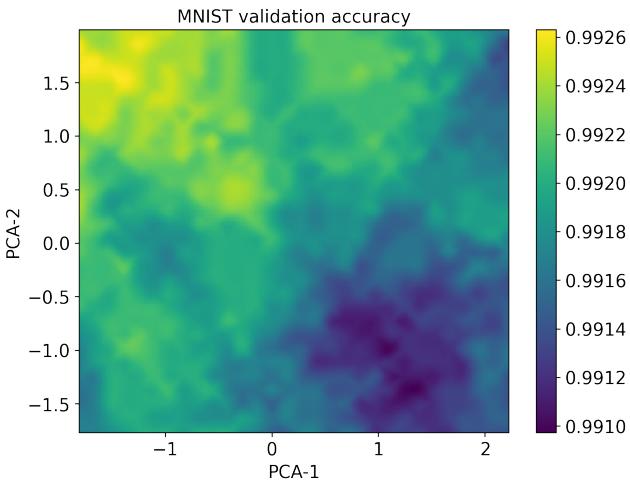


Motivation

- Training:



- Validation:

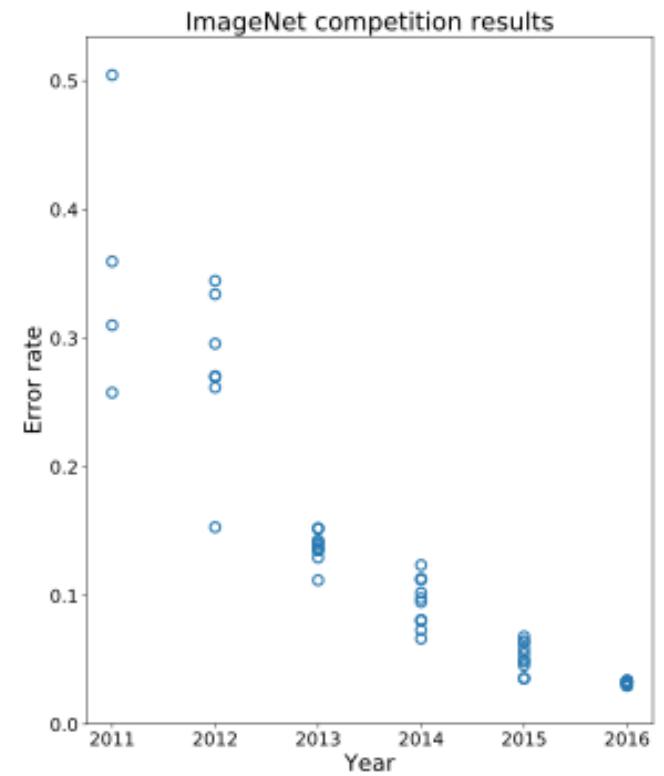


Deep Learning at Scale

- Motivation
 - 1. Speed:
 - Fast experimentation
 - Fail fast, try new ideas
 - 2. Quality:
 - Allows SGD with larger batch sizes, less noise
 - Search a large hyper-parameter space
 - 3. Potential:
 - Train models larger than memory (not covered in this course)
e.g. GPT-3 175 billion parameters won't fit in any available accelerator

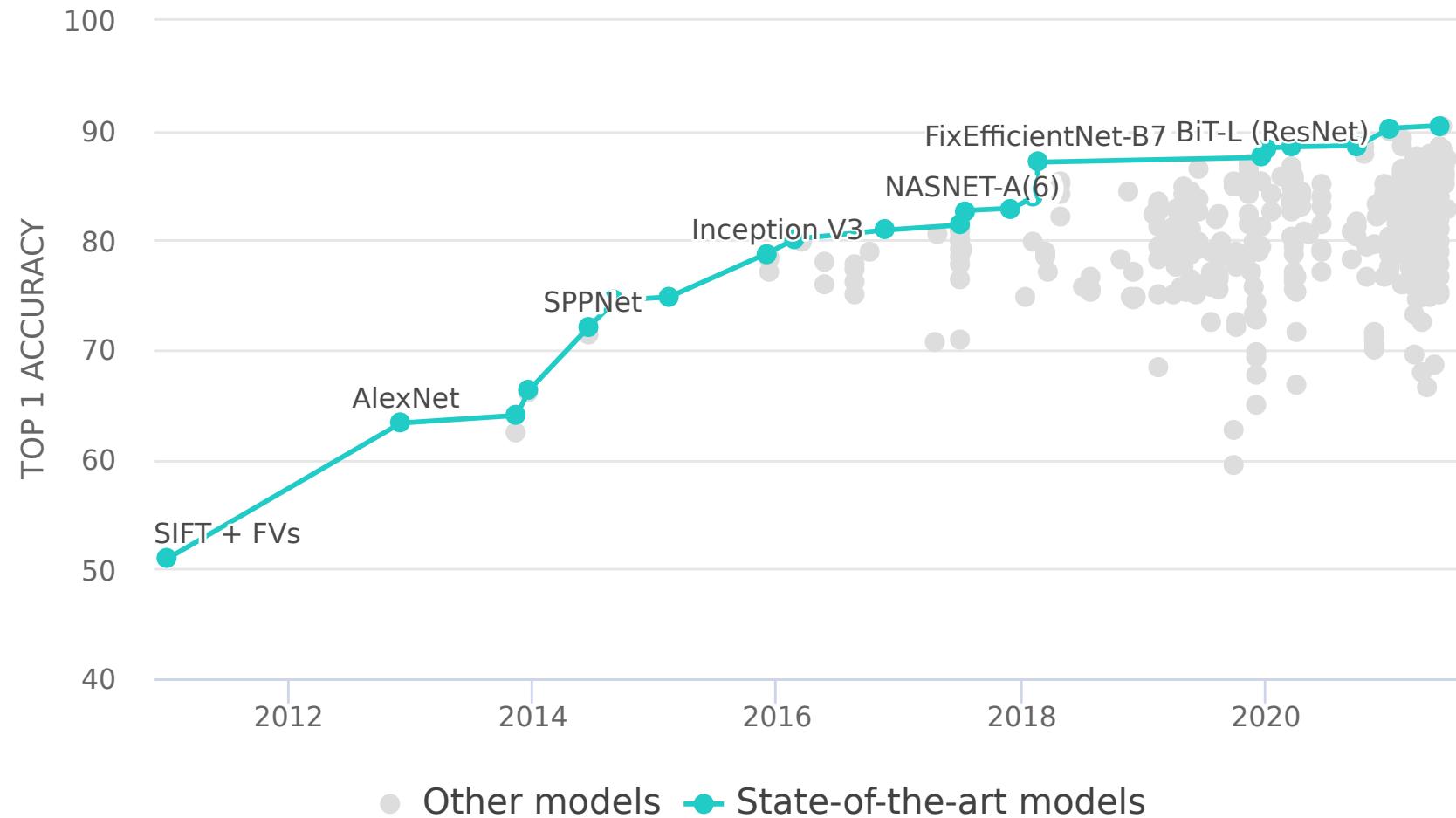
ImageNet Large Scale Visual Recognition Challenge (ILSVRC)

- First large scale image dataset
 - 14,197,122 images
 - Manually annotated to 21,841 WordNet synsets
 - High class imbalance
- 2012 subset with 1000 classes
 - 1,281,166 images
 - 50,000 images for validation
 - AlexNet greatly outperforms traditional ML methods
- Made deep learning popular
 - Large pretrained models allowed *transfer-learning* to other domains

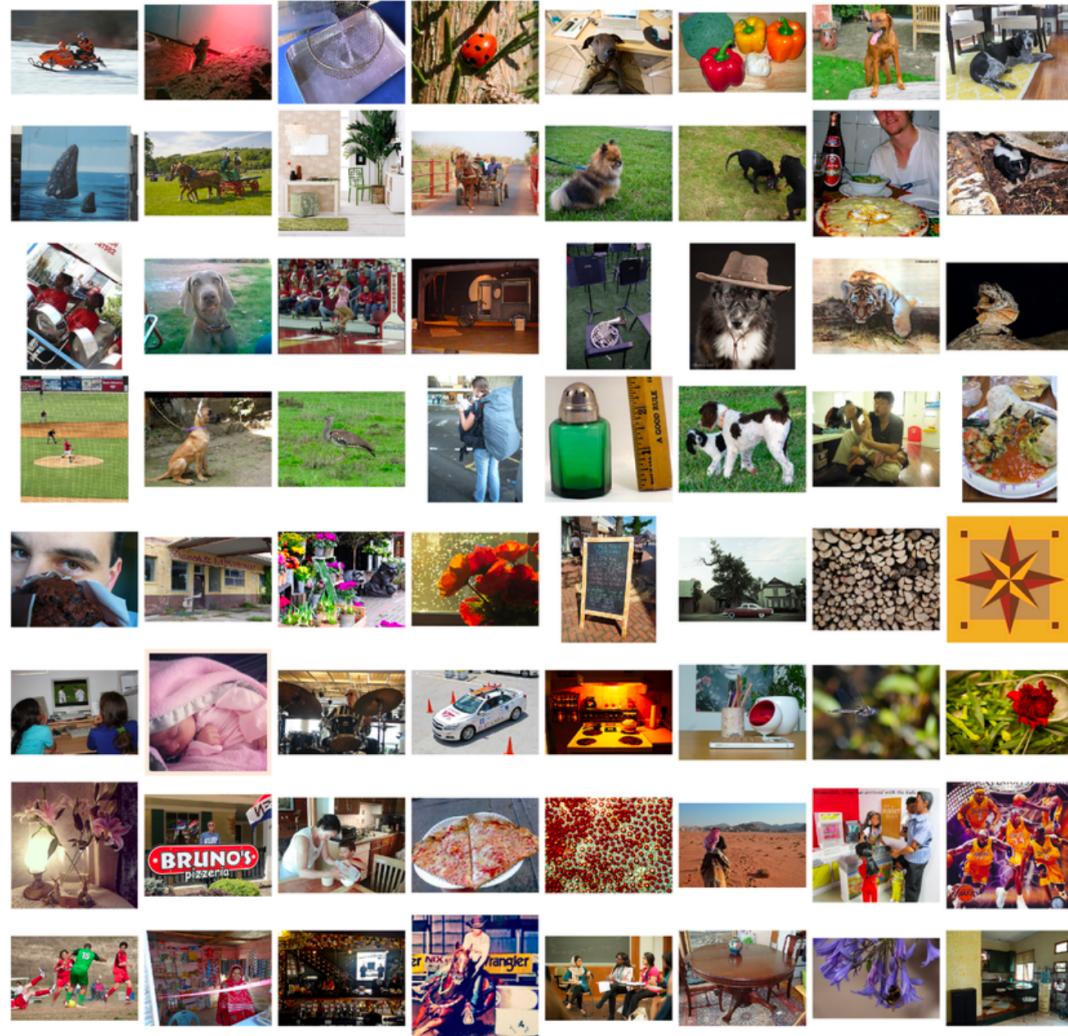


Top5-Accuracy error: <https://en.wikipedia.org/wiki/ImageNet>

ImageNet Large Scale Visual Recognition Challenge (ILSVRC)



ImageNet Large Scale Visual Recognition Challenge (ILSVRC)



{0: 'tench, *Tinca tinca*',
1: 'goldfish, *Carassius auratus*',
2: 'great white shark, white shark, man-eater, man-eating shark, *Carcharodon carcharias*',
3: 'tiger shark, *Galeocerdo cuvieri*',
4: 'hammerhead, hammerhead shark',
5: 'electric ray, crampfish, numbfish, torpedo',
6: 'stingray',
7: 'cock',
8: 'hen',
9: 'ostrich, *Struthio camelus*',
10: 'brambling, *Fringilla montifringilla*',
11: 'goldfinch, *Carduelis carduelis*',
12: 'house finch, linnet, *Carpodacus mexicanus*',
[...]
152-269: 117 DOG BREEDS
[...]
988: 'acorn',
989: 'hip, rose hip, rosehip',
990: 'buckeye, horse chestnut, conker',
991: 'coral fungus',
992: 'agaric',
993: 'gyromitra',
994: 'stinkhorn, carrion fungus',
995: 'earthstar',
996: 'hen-of-the-woods, hen of the woods, *Polyporus frondosus, Grifola frondosa*',
997: 'bolete',
998: 'ear, spike, capitulum',
999: 'toilet tissue'}

Storage Access: TF Records

- Accessing millions of small files concurrently can be challenging
 - Disk and Network drives work better with larger files
- TF Records
 - Can pack several images and metadata serially on each file
 - Can read image by image without loading the whole file to memory

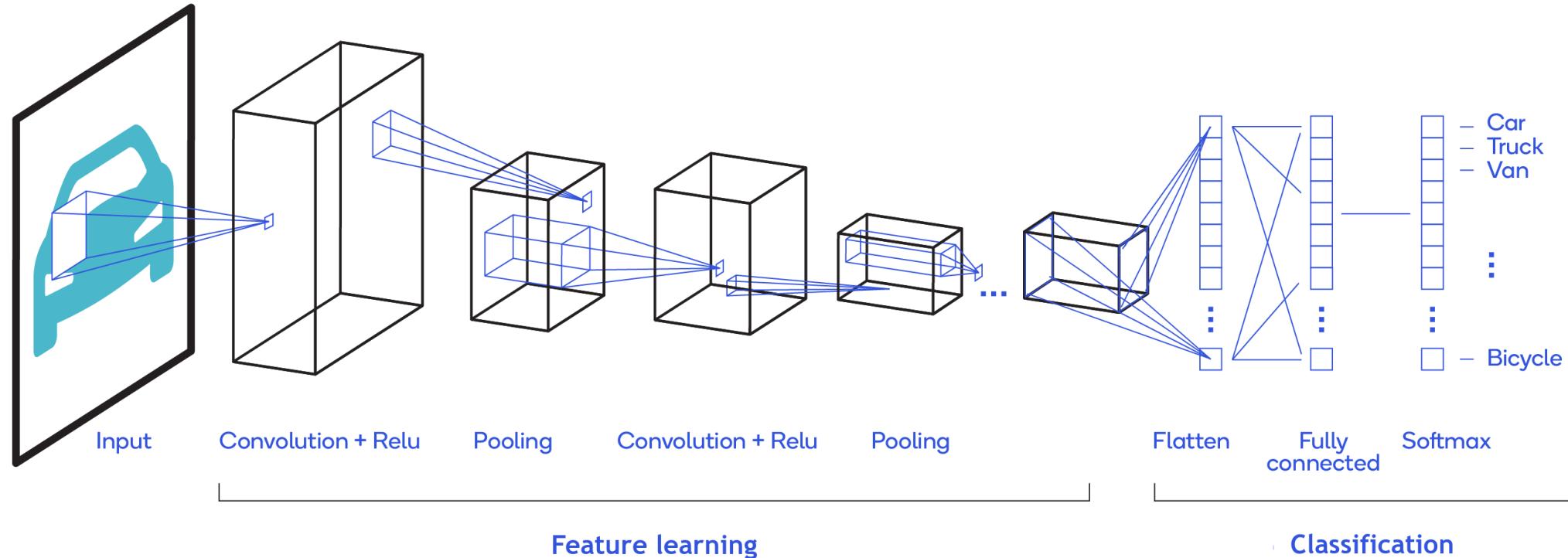
```
features = {  
    'filename': tf.train.Feature(bytes_list = tf.train.BytesList(value = [bytes(filename, 'utf-8')])),  
    'Image shapes': tf.train.Feature(int64_list = tf.train.Int64List(value = [img_shape[0], img_shape[1]])),  
    'image': tf.train.Feature(bytes_list = tf.train.BytesList(value = [image])),  
    'label': tf.train.Feature(bytes_list = tf.train.BytesList(value = [label])),  
    ...  
}  
  
records = [{'image': [...], 'label': 123}, {'image': [...], 'label': 42}, {'image': [...], 'label': 789}, ...]
```

TF Records

- Demo Notebook
 - Decode TF records
 - Encode

Convolutional Neural Networks

- Exploits translational equivariance on natural images
- Large filters can be decomposed into smaller ones



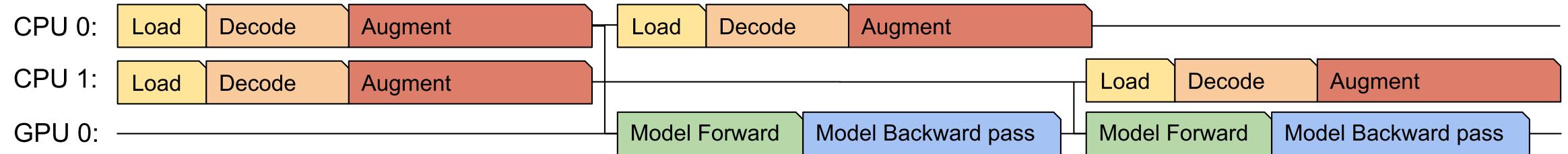
Convolutional Neural Networks

- Practice Notebook
 - 01_imagenet-conv.ipynb

Data Augmentation on CNN's

- Convolutional layers are invariant to translation (spatial shift)
 - But not to rotations
 - Only to certain degree to color and luminosity shift
- Data Augmentation is yet another form of regularization
 - Implicitly part of the modeling
 - Extensive literature, and several open source libraries
 - **Generally done on CPU in the background**
 - Some GPU implementations are also available
 - Augmentation can be used as a platform for self-supervised learning

Data Augmentation

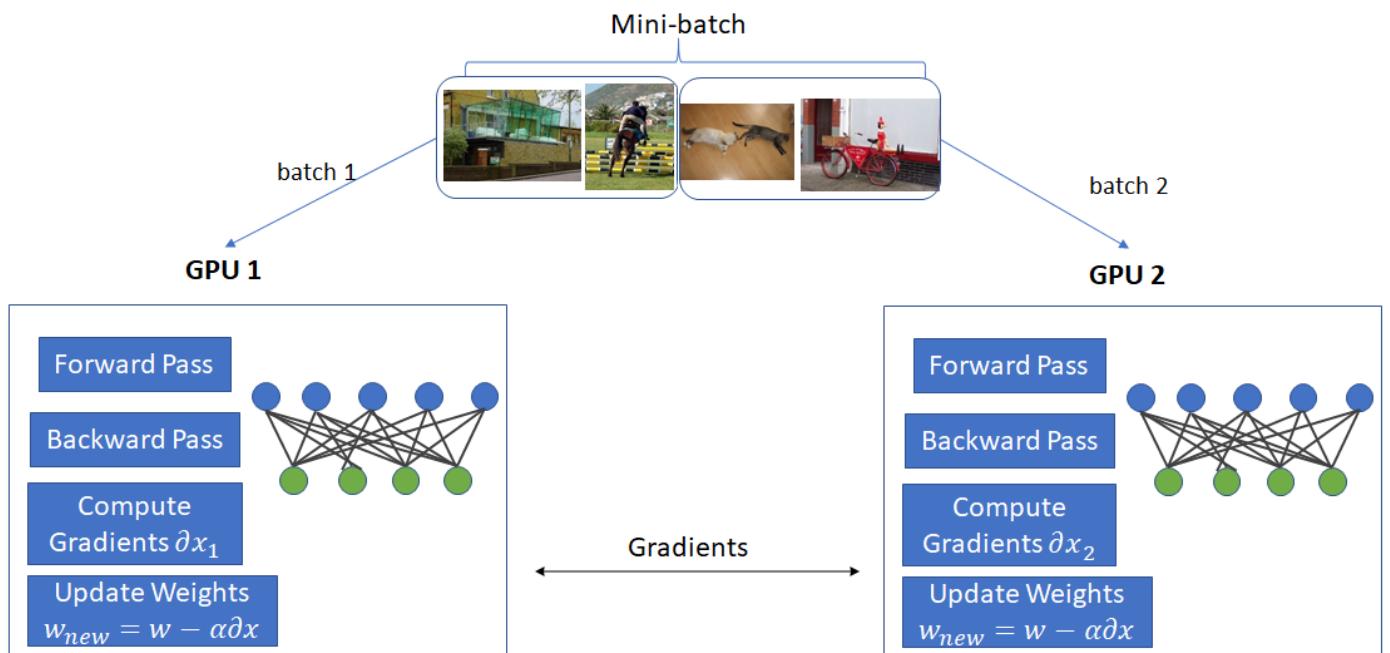


Data Augmentation

- Practice Notebooks
 - 02_imagenet_tfa.ipynb
 - 02b_imagenet_albumentations.ipynb
 - 03_imagenet.ipynb

Deep Learning at Scale

- Data parallelism
 - Sharded data-loader
 - $1/N$ of the data on each N th node
 - Gradient synchronization
 - AllReduce gradients



Deep Learning at Scale

- Practice Notebook
 - 04_tf-data-shard.ipynb
 - 05_dist-imagenet.ipynb

Batch Norm Synchronization

- Exponential Moving Average per channel

$$x_i^* = \frac{x_i - \mu}{\sigma + \epsilon}$$

$$y_i = \gamma x_i^* + \beta$$

$$\mu = (1 - \alpha)\mu + \alpha\mu_B \quad \mu_B = \frac{1}{n} \sum_i^n x_i$$

$$\sigma^2 = (1 - \alpha)\sigma^2 + \alpha\sigma_B^2 \quad \sigma_B^2 = \frac{1}{n} \sum_i^n (x_i - \mu)^2$$

- SyncBatchNormalization Layer
 - Synchronizes all statistics during forward pass
 - AllReduce across BN layers may induce additional latency

Batch Norm Synchronization

- Demo Notebook
 - 06_bn-sync.ipynb

Machine Learning Engineering - Good Practices

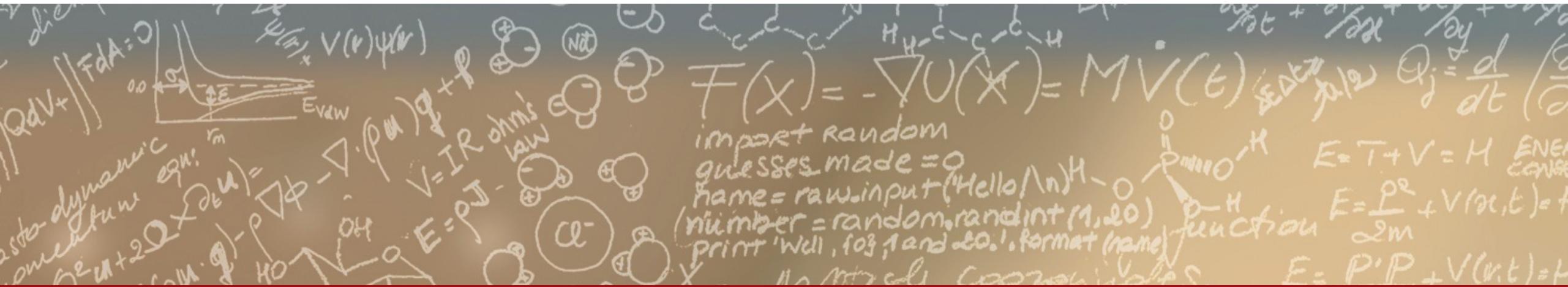
- Use reliable metrics and validation split (data science first)
- Unit-test model, data pipeline and loss functions
 - Integration test: overfit to subset of data
 - Test checkpoints
- Reuse code, data and models
 - Transfer-learning when possible
- Hyper-parameter search
 - including data augmentation, batch size and LR scheduling
 - Only then run architecture search



CSCS

Centro Svizzero di Calcolo Scientifico
Swiss National Supercomputing Centre

ETH zürich



Thank you for your attention.