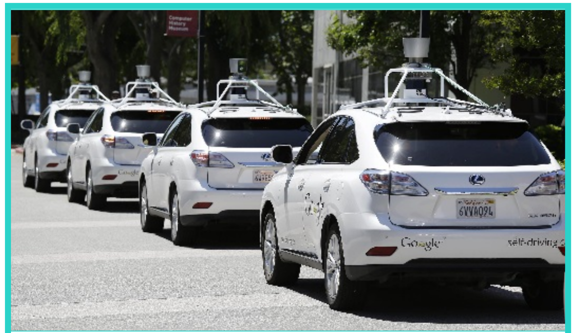
Deep Learning Cookbook: technology recipes to run deep learning workloads

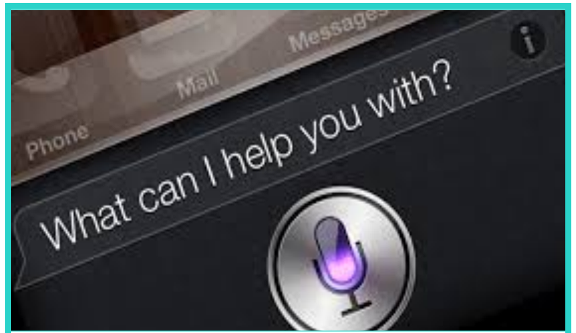
Natalia Vassilieva, Sergey Serebryakov

Deep learning applications



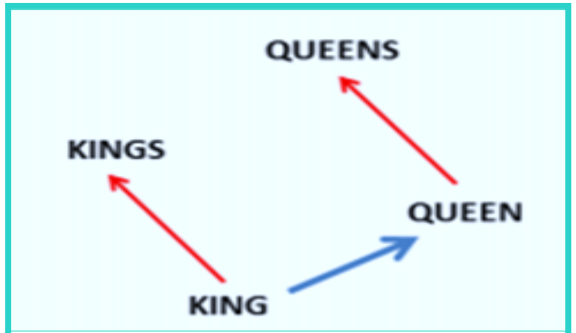
Vision

- Search & information extraction
- Security/Video surveillance
- Self-driving cars
- Medical imaging
- Robotics



Speech

- Interactive voice response (IVR) systems
- Voice interfaces (Mobile, Cars, Gaming, Home)
- Security (speaker identification)
- Health care
- Simultaneous interpretation



Text

- Search and ranking
- Sentiment analysis
- Machine translation
- Question answering

Other

- Recommendation engines
- Advertising
- Fraud detection
- AI challenges
- Drug discovery
- Sensor data analysis
- Diagnostic support

Deep learning ecosystem

Software

Caffe
Keras
theano
torch
KALDI
Caffe2
Microsoft CNTK
mxnet
TensorFlow
Chainer

Hardware

NVIDIA.
intel
Movidius
nervana systems
ARM
Google
XILINX
AMD
QUALCOMM

How to pick the right hardware/software stack?

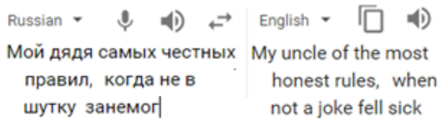
Does one size fit all?

Applications break down

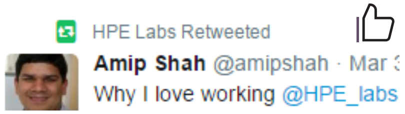
Data	Application
Images	Tissue classification in medical images
Video	Video surveillance
Speech	Speech recognition
Text	Sentiment analysis
Sensor	Predictive maintenance
Other	Fraud detection

Detection
Look for a known object/pattern

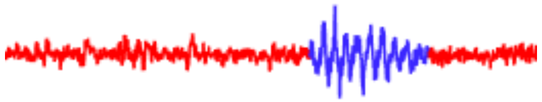
Generation
Generate content



Classification
Assign a label from a predefined set of labels



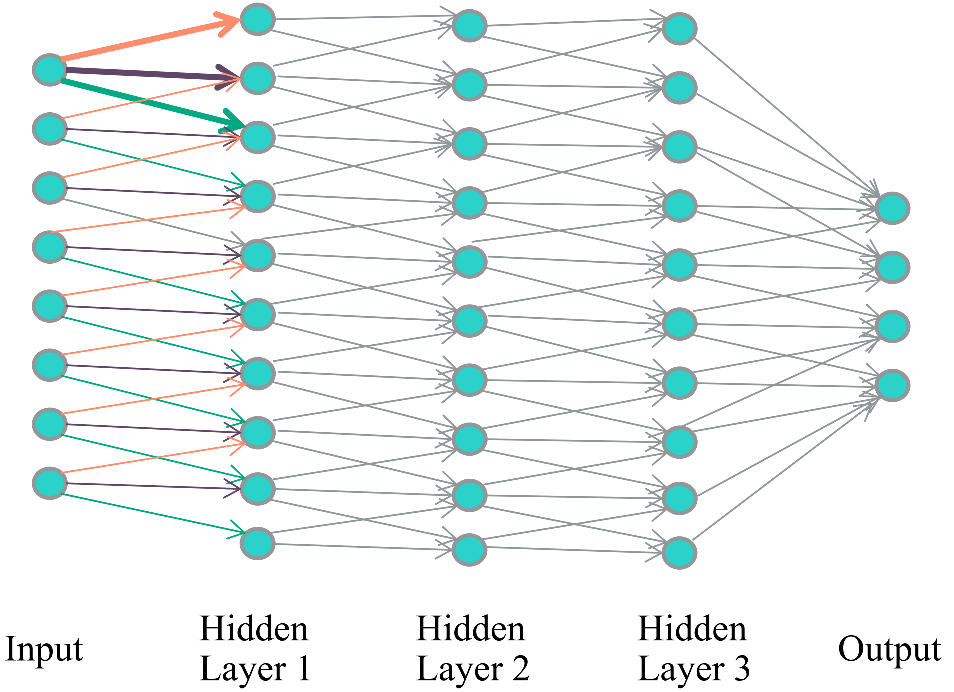
Anomaly detection
Look for abnormal, unknown patterns



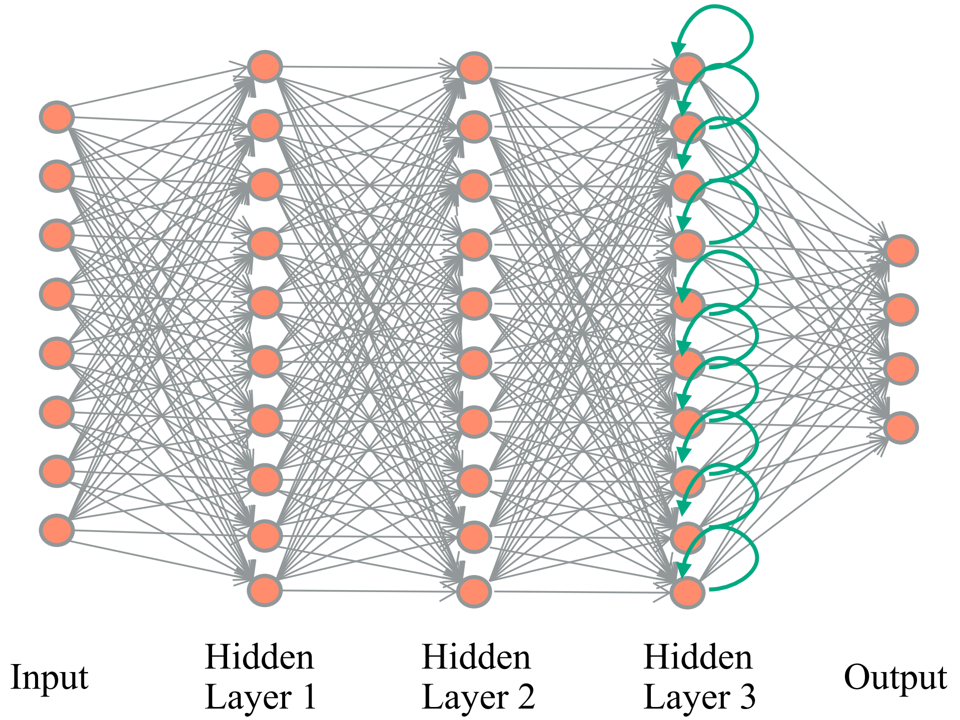
Types of artificial neural networks

Topology to fit data characteristics

Images:
Convolutional (CNN)



Speech, time series, sequences:
Fully Connected (FC), Recurrent (RNN)



One size does NOT fit all

Application

Data type

Data size

Model (topology of artificial neural network):

- How many layers
- How many neurons per layer
- Connections between neurons (types of layers)

Popular models

Name	Type	Model size (# params)	Model size (MB)	GFLOPs (forward pass)
AlexNet	CNN	60,965,224	233 MB	0.7
GoogleNet	CNN	6,998,552	27 MB	1.6
VGG-16	CNN	138,357,544	528 MB	15.5
VGG-19	CNN	143,667,240	548 MB	19.6
ResNet50	CNN	25,610,269	98 MB	3.9
ResNet101	CNN	44,654,608	170 MB	7.6
ResNet152	CNN	60,344,387	230 MB	11.3
Eng Acoustic Model	RNN	34,678,784	132 MB	0.035
TextCNN	CNN	151,690	0.6 MB	0.009

Popular models

Name	Type	Model size (# params)	Model size (MB)	GFLOPs (forward pass)
AlexNet	**CNN**	**60,965,224**	**233 MB**	**0.7**
GoogleNet	CNN	6,998,552	27 MB	1.6
VGG-16	CNN	138,357,544	528 MB	15.5
VGG-19	CNN	143,667,240	548 MB	19.6
ResNet50	CNN	25,610,269	98 MB	3.9
ResNet101	CNN	44,654,608	170 MB	7.6
ResNet152	**CNN**	**60,344,387**	**230 MB**	**11.3**
Eng Acoustic Model	RNN	34,678,784	132 MB	0.035
TextCNN	CNN	151,690	0.6 MB	0.009

Compute requirements

Name	Type	Model size (# params)	Model size (MB)	GFLOPs (forward pass)
ResNet152	**CNN**	**60,344,387**	**230 MB**	**11.3**

Training data: 14M images (ImageNet)

FLOPs per epoch: $3 * 11.3 * 10^9 * 14 * 10^6 \approx 5 * 10^{17}$

1 epoch per hour: ~140 TFLOPS

Today's hardware:

Google TPU2: 180 TFLOPS Tensor ops

NVIDIA Tesla V100: 15 TFLOPS SP (30 TFLOPS FP16, 120 TFLOPS Tensor ops), 12 GB memory

NVIDIA Tesla P100: 10.6 TFLOPS SP, 16 GB memory

NVIDIA Tesla K40: 4.29 TFLOPS SP, 12 GB memory

NVIDIA Tesla K80: 5.6 TFLOPS SP (8.74 TFLOPS SP with GPU boost), 24 GB memory

INTEL Xeon Phi: 2.4 TFLOPS SP

Model parallelism

– Can be achieved with scalable distributed matrix operations

– Requires a certain compute/bandwidth ratio

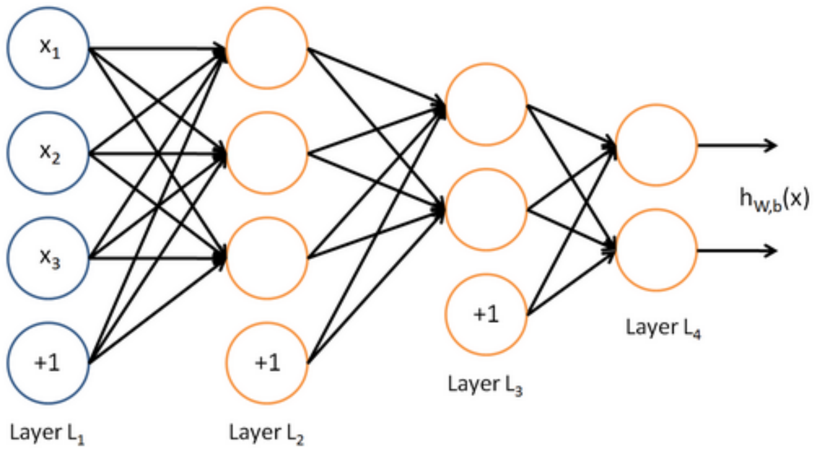
Let's assume:

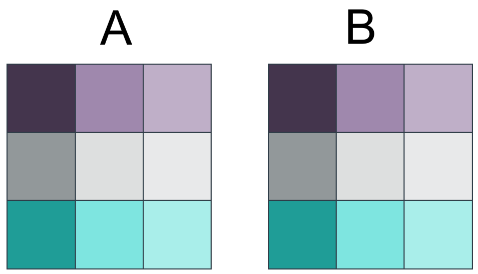
n – input size = batch size = output size
γ – compute power of the device (FLOPS)
β – bandwidth (memory or interconnect)
p^2 – number of compute devices

$$T_{compute} = \frac{2n^3}{p^2\gamma} \qquad T_{data\_read} = \frac{2n^2}{p\beta}$$

for FP32

$$\beta \geq \frac{4p\gamma}{n}$$





"SUMMA: Scalable Universal Matrix Multiplication Algorithm", R.A. van de Geijn, J. Watts

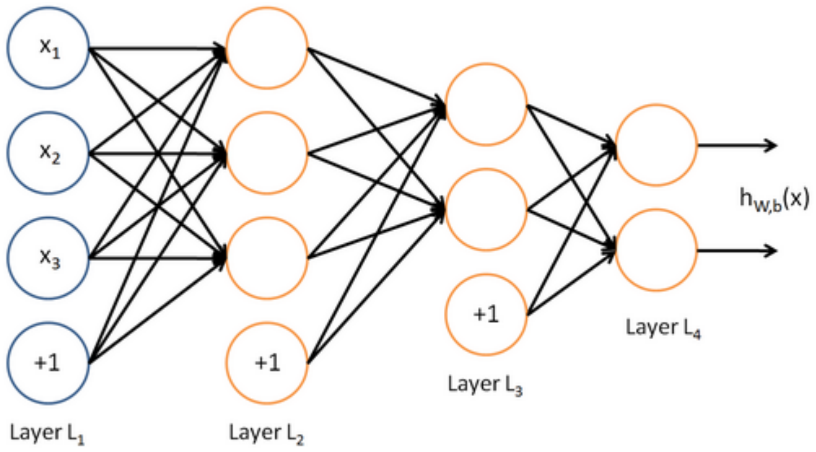
Model parallelism

– Can be achieved with scalable distributed matrix operations

– Requires a certain compute/bandwidth ratio

Let's assume:

n – input size = batch size = output size
γ – compute power of the device (FLOPS)
β – bandwidth (memory or interconnect)
p^2 – number of compute devices



$$T_{compute} = \frac{2n^3}{p^2\gamma} \qquad T_{data\_read} = \frac{2n^2}{p\beta}$$

for FP32

$$\beta \geq \frac{4p\gamma}{n}$$

$$n = 2000, \qquad \gamma = 15\,TFLOPS$$

$$p = 10, \qquad \beta \geq 300\,GB/s$$

$$p = 1, \qquad \beta \geq 30\,GB/s$$

Hewlett Packard Enterprise

Data parallelism

$$T_{compute}(p, c, \gamma) = c/(p\gamma)$$

$$T_{communicate}(p, w, \beta) = 2wlog(p)/\beta$$

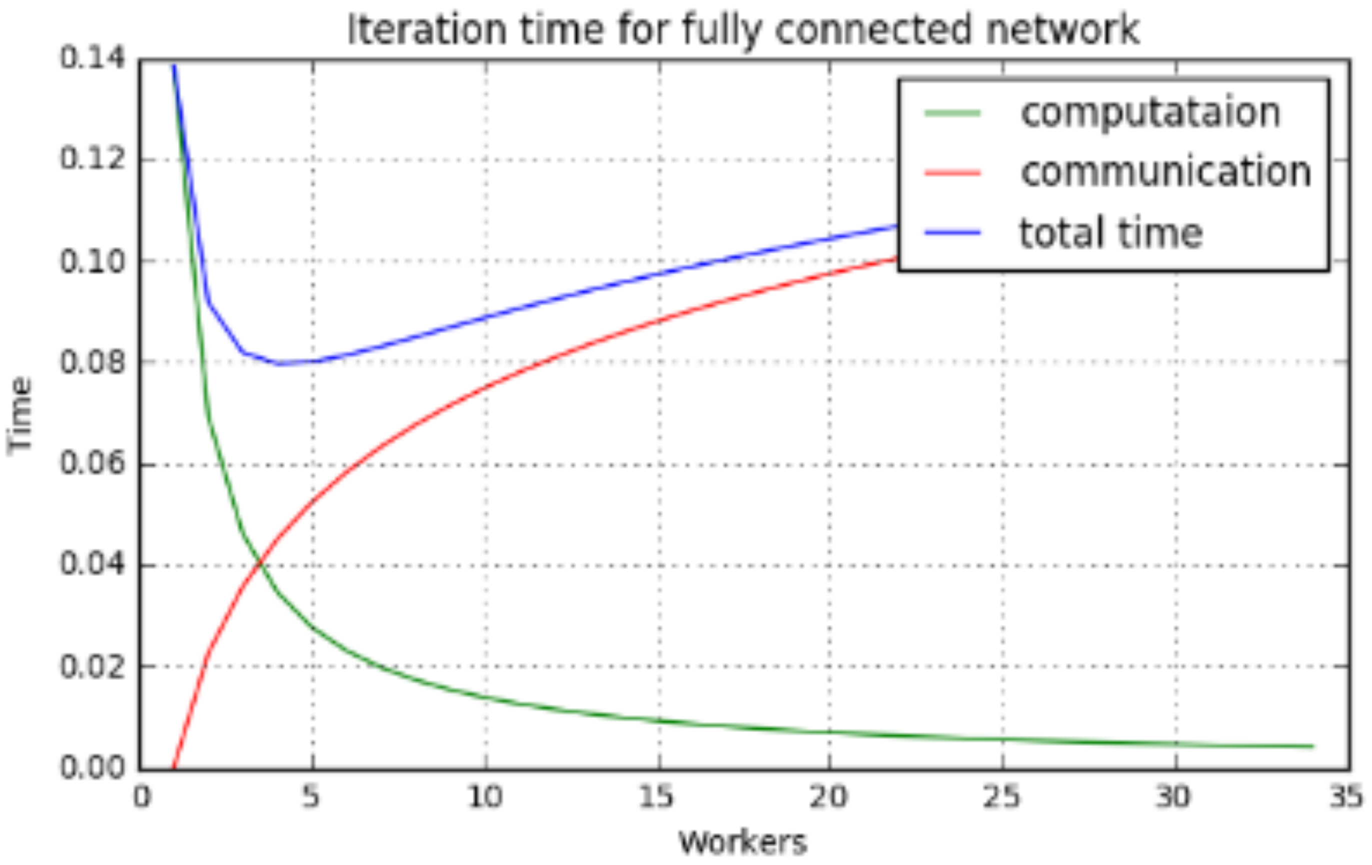
p – number of workers (nodes),
γ – the computational power of the node,
c – the computational complexity of the model,
β – bandwidth,
w – the size of the weights in bits.

Data parallelism

$$T_{compute}(p, c, \gamma) = c/(p\gamma)$$

$$T_{communicate}(p, w, \beta) = 2wlog(p)/\beta$$

p – number of workers (nodes),
γ – the computational power of the node,
c – the computational complexity of the model,
β – bandwidth,
w – the size of the weights in bits.



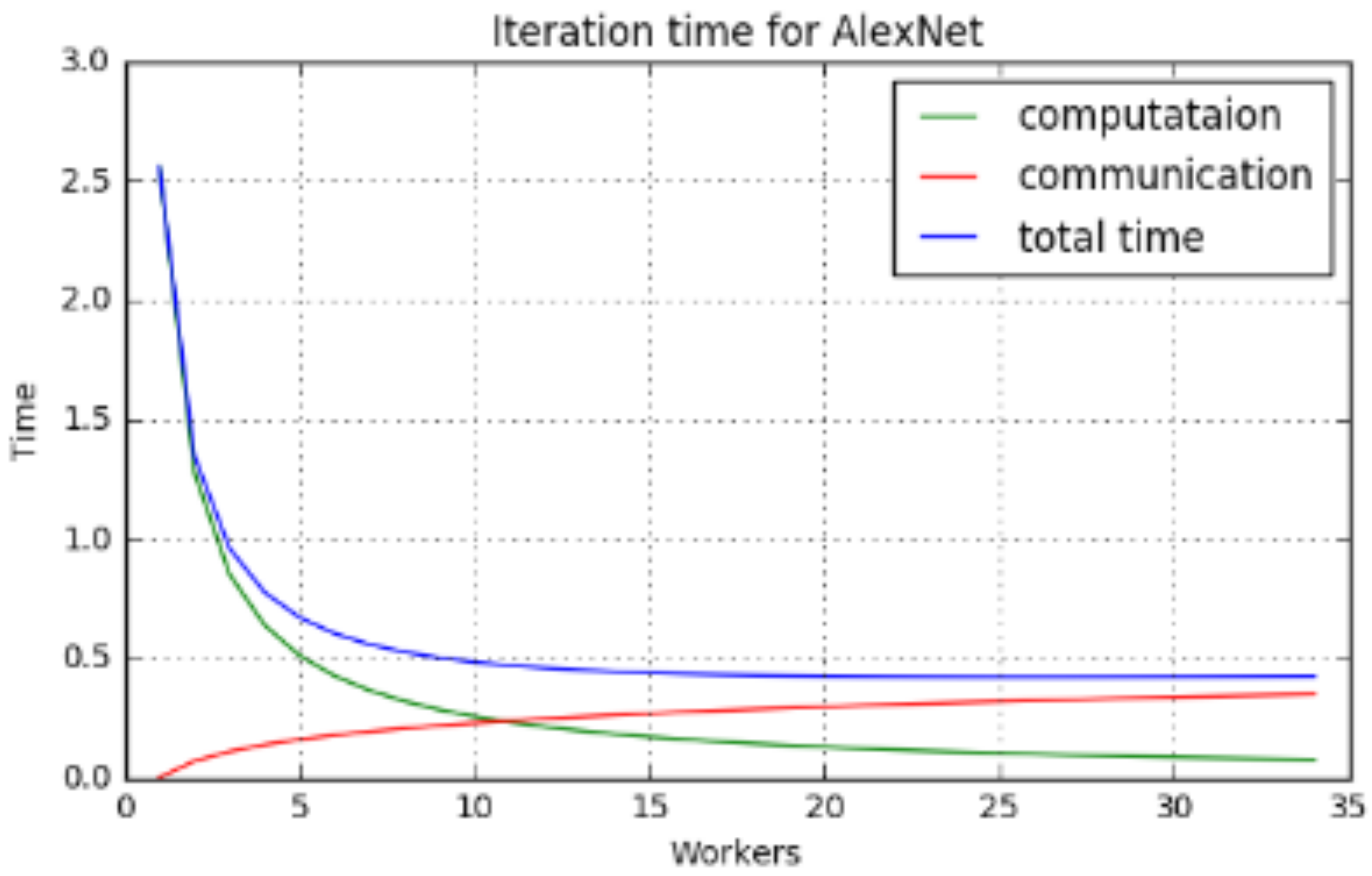
NVIDIA K40 (~4 TFLOPS), PCIe v3 (~16 GB/s)

Data parallelism

$$T_{compute}(p, c, \gamma) = c/(p\gamma)$$

$$T_{communicate}(p, w, \beta) = 2wlog(p)/\beta$$

p – number of workers (nodes),
γ – the computational power of the node,
c – the computational complexity of the mode
β – bandwidth,
w – the size of the weights in bits.



NVIDIA K40 (~4 TFLOPS), Infiniband (~56 Gb/s)

Hewlett Packard Enterprise

Deep Learning Cookbook helps to pick the right HW/SW stack

- **Benchmarking suite**
  - Benchmarking scripts
  - Set of benchmarks (for core operations and reference models)
- **Performance measurements** for a subset of applications, models and HW/SW stacks
  - 11 models
  - 8 frameworks
  - 6 hardware systems
- **Analytical performance and scalability models**
  - Performance prediction for arbitrary models
  - Scalability prediction
- Reference solutions, white papers

Deep Learning Cookbook

Automatic Meeting Notes | Video Surveillance | Hospital Smart Care Unit | **Custom**

- [x] Images
- [ ] Videos
- [ ] Text
- [ ] Speech
- [ ] Sensor Data

- [x] Classification
- [ ] Detection
- [ ] Generation
- [ ] Anomaly Detection

- (•) Training
  - (•) Large
  - ( ) Medium
  - ( ) Small
- ( ) Inference

Recommend

Data and Model

Data size	Epochs	Model
100000000	10	VGG16

Hardware

Server	Processor unit	
Apollo 6500	NVIDIA P100	

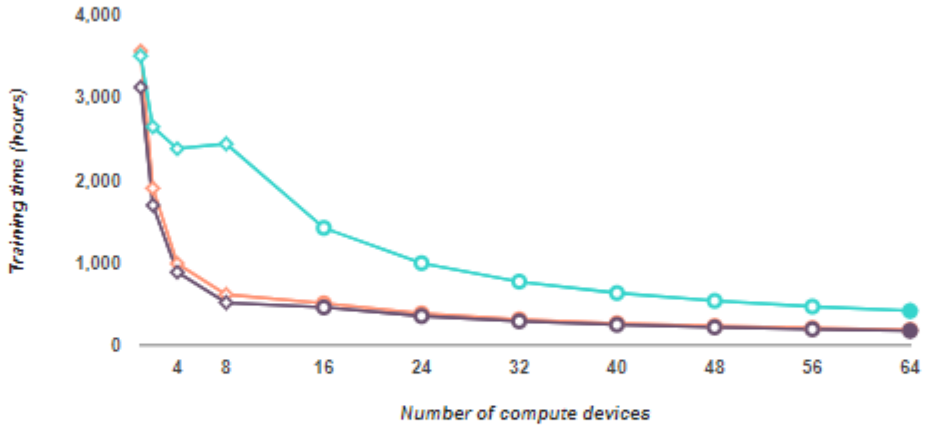
Count	Cluster size	Interconnect
8	8	InfiniBand FDR

Software

Framework	Batch size	Scaling
TensorFlow	16	weak

Add

Training performance

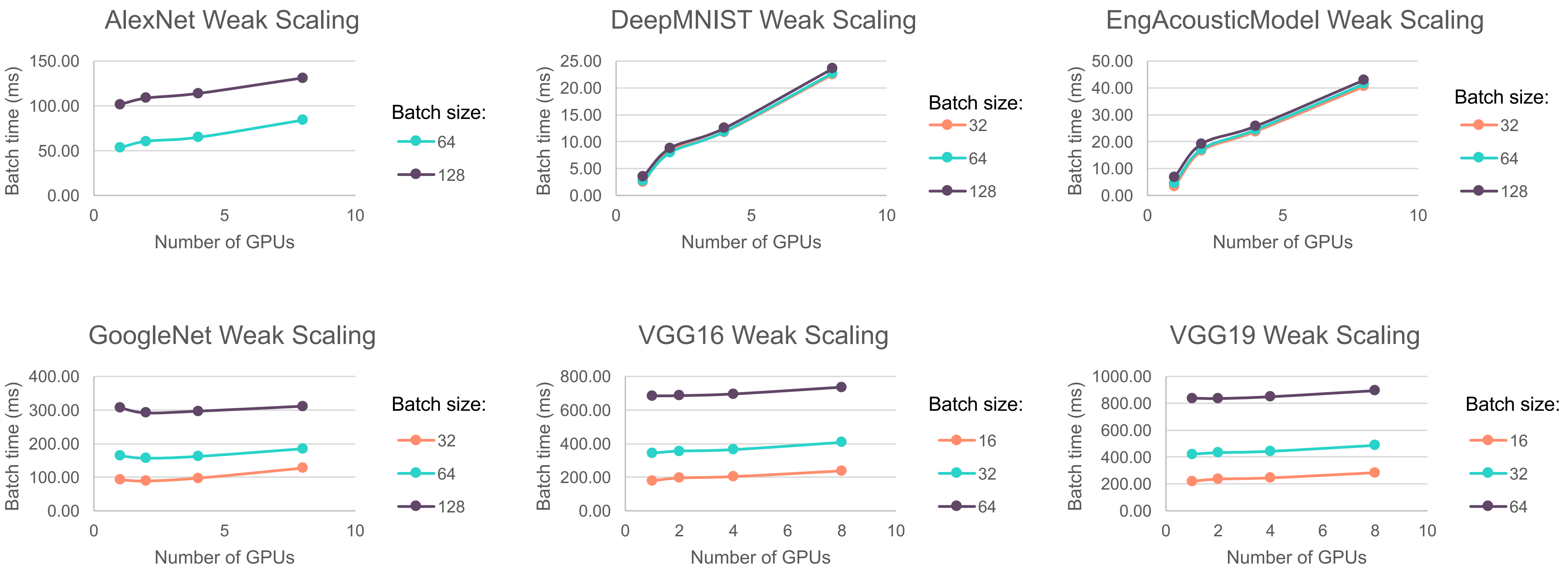


Data	Hardware	Software	Time (hours)
Size 100000000, Epochs 10, Model VGG16	Server Apollo 6500, PU NVIDIA P100, Count 8, Cluster size 8, Interconnect IB	Framework BVLC Caffe, Batch 16(weak)	188.1
Size 100000000, Epochs 10, Model VGG16	Server Apollo 6500, PU NVIDIA P100, Count 8, Cluster size 8, Interconnect IB	Framework Caffe2, Batch 16(weak)	175.8
Size 100000000, Epochs 10, Model VGG16	Server Apollo 6500, PU NVIDIA P100, Count 8, Cluster size 8, Interconnect IB	Framework TensorFlow, Batch 16(weak)	416.1

Remove all

Selected scalability results

HPE Apollo 6500 (8 x NVIDIA P100)



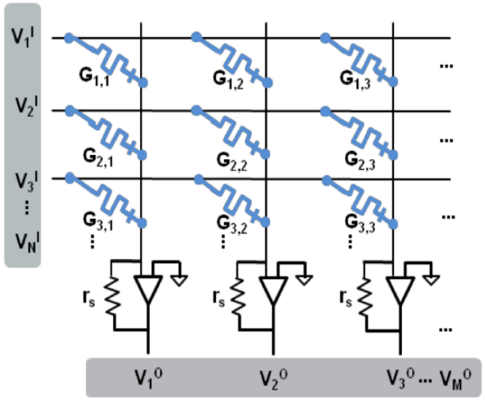
Selected observations and tips

- Larger models are easier to scale (such as ResNet and VGG)
  - A single GPU can hold only small batches (the rest of memory is occupied by a model)
- Fast interconnect is more important for less compute-intensive models (FCC)
- A rule of thumb: 1 or 2 CPU cores per GPU
- PCIe topology of the system is important

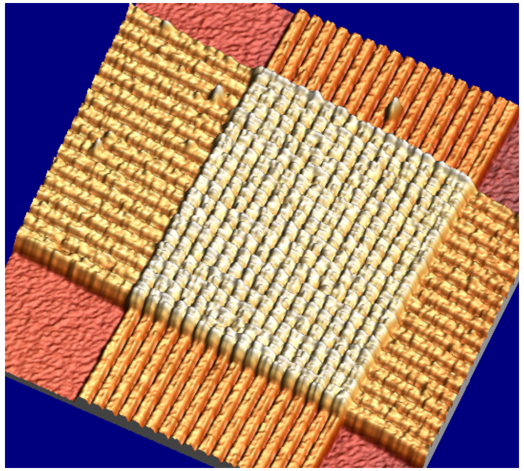
Further into the future: neuromorphic research projects

Neuromorphic Computing – the integration of algorithms, architectures, and technologies, informed by neuroscience, to create new computational approaches.

- **Memristor Dot-Product Engine (DPE) – successfully demonstrated**
  - Memristor crossbar analog vector-matrix multiplication accelerator
- **Hopfield Network (electronic and photonic) – in progress**



$$I_j^O = \sum_j G_{ij} \cdot V_i^I$$



Hewlett Packard Enterprise

Thank you

Natalia Vassilieva
nvassilieva@hpe.com