EE-559 – Deep learning

1a. Introduction

François Fleuret
https://fleuret.org/dlc/
February 20, 2018

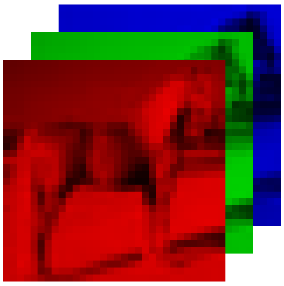
Why learning

Many applications require the automatic extraction of “refined” information from raw signal (*e.g.* image recognition, automatic speech processing, natural language processing, robotic control, geometry reconstruction).

(ImageNet)

Our brain is so good at interpreting visual information that the “semantic gap” is hard to assess intuitively.

This  is a horse 




```
>>> from torchvision import datasets
>>> cifar = datasets.CIFAR10('./data/cifar10/', train=True, download=True)
Files already downloaded and verified
>>> x = torch.from_numpy(cifar.train_data)[43].transpose(2, 0).transpose(1, 2)
>>> x.size()
torch.Size([3, 32, 32])
>>> x.narrow(1, 0, 4).narrow(2, 0, 12)

(0 ,.,.) =
   99   98  100  103  105  107  108  110  114  115  117  118
  100  100  102  105  107  109  110  112  115  117  119  120
  104  104  106  109  111  112  114  116  119  121  123  124
  109  109  111  113  116  117  118  120  123  124  127  128

(1 ,.,.) =
  166  165  167  169  171  172  173  175  176  178  179  181
  166  164  167  169  169  171  172  174  176  177  179  180
  169  167  170  171  171  173  174  176  178  179  182  183
  170  169  172  173  175  176  177  178  179  181  183  184

(2 ,.,.) =
  198  196  199  200  200  202  203  204  205  206  208  209
  195  194  197  197  197  199  200  201  202  203  206  207
  197  195  198  198  198  199  201  202  203  204  206  207
  197  196  199  198  198  199  200  201  203  204  207  208
[torch.ByteTensor of size 3x4x12]
```

Extracting semantic automatically requires models of extreme complexity, which cannot be designed by hand.

Techniques used in practice consist of

1. defining a parametric model, and
2. optimizing its parameters by "making it work" on training data.

This is similar to biological systems for which the model (*e.g.* brain structure) is DNA-encoded, and parameters (*e.g.* synaptic weights) are tuned through experiences.

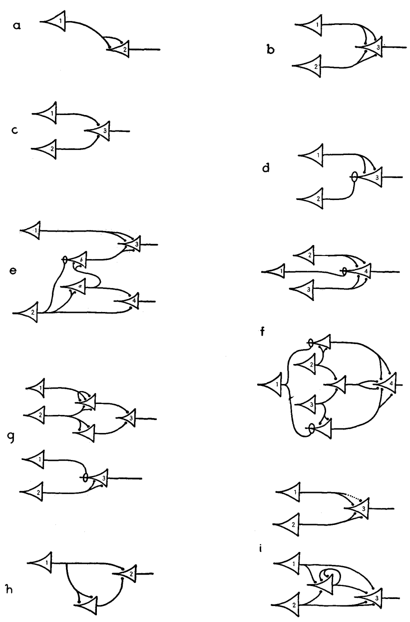
Deep learning encompasses software technologies to scale-up to billions of model parameters and as many training examples.

There are strong connections between standard statistical modeling and machine learning.

Classical ML methods combine a "learnable" model from statistics (e.g "linear regression") with prior knowledge in pre-processing.

"Artificial neural networks" pre-dated these approaches, and do not follow that dichotomy. They consist of "deep" stacks of parametrized processing.

From artificial neural networks to "Deep Learning"



Networks of "Threshold Logic Unit"

(McCulloch and Pitts, 1943)

1949 – Donald Hebb proposes the Hebbian Learning principle.

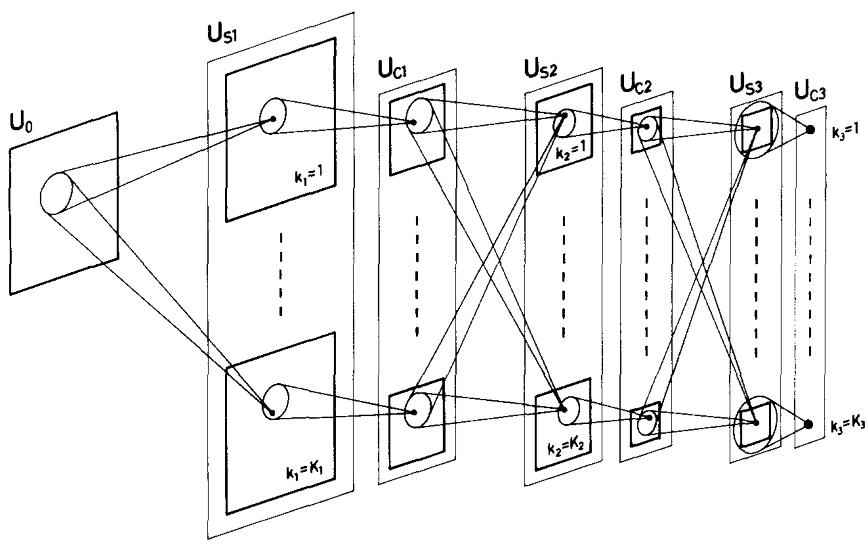
1951 – Marvin Minsky creates the first ANN (Hebbian learning, 40 neurons).

1958 – Frank Rosenblatt creates a perceptron to classify 20×20 images.

1959 – David H. Hubel and Torsten Wiesel's demonstrate orientation selectivity and columnar organization in the cat's visual cortex.

1982 – Paul Werbos proposes back-propagation for ANNs.

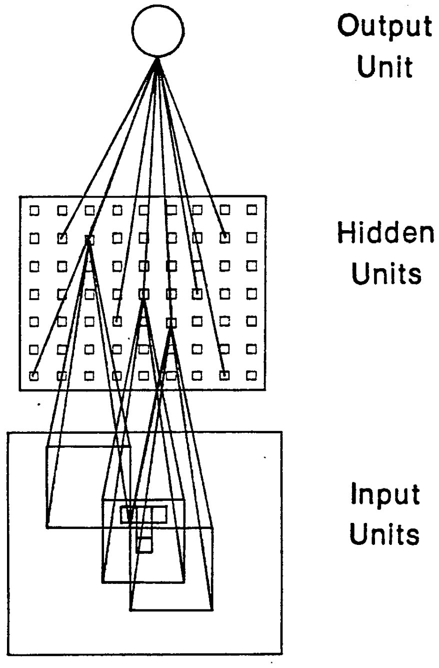
Neocognitron



Follows Hubel and Wiesel's results.

(Fukushima, 1980)

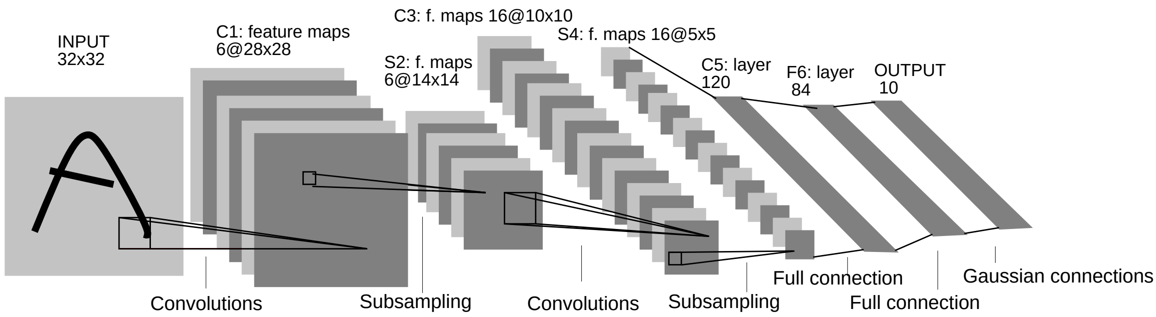
Network for the T-C problem



Trained with back-prop.

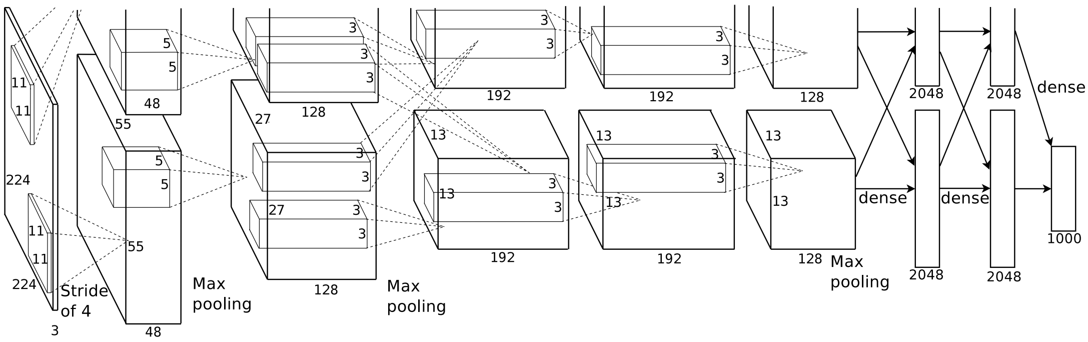
(Rumelhart et al., 1988)

LeNet-5



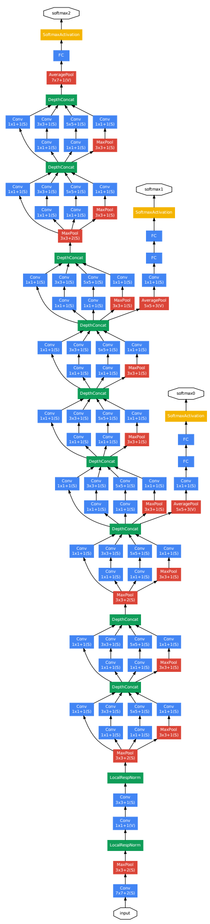
(leCun et al., 1998)

AlexNet



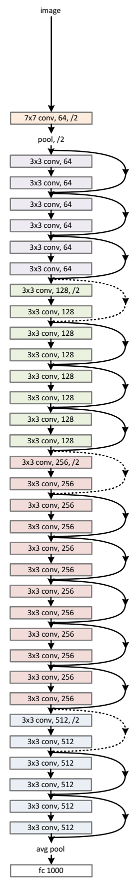
(Krizhevsky et al., 2012)

GoogLeNet



(Szegedy et al., 2015)

Resnet



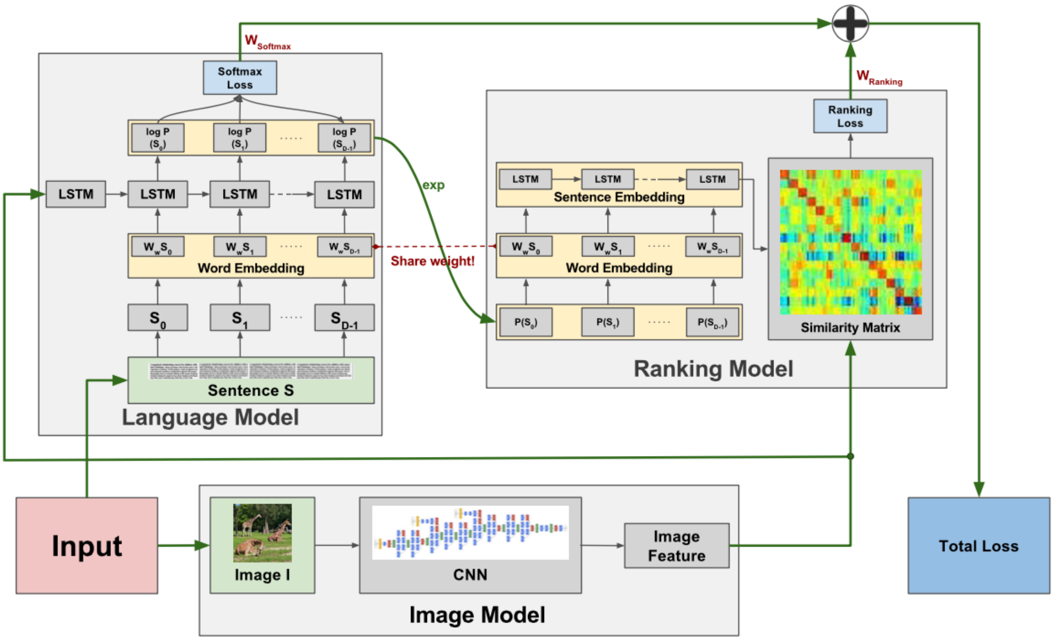
(He et al., 2015)

Deep learning is built on a natural generalization of a neural network: **a graph of tensor operators**, taking advantage of

- the chain rule (aka "back-propagation"),
- stochastic gradient decent,
- convolutions,
- parallel operations on GPUs.

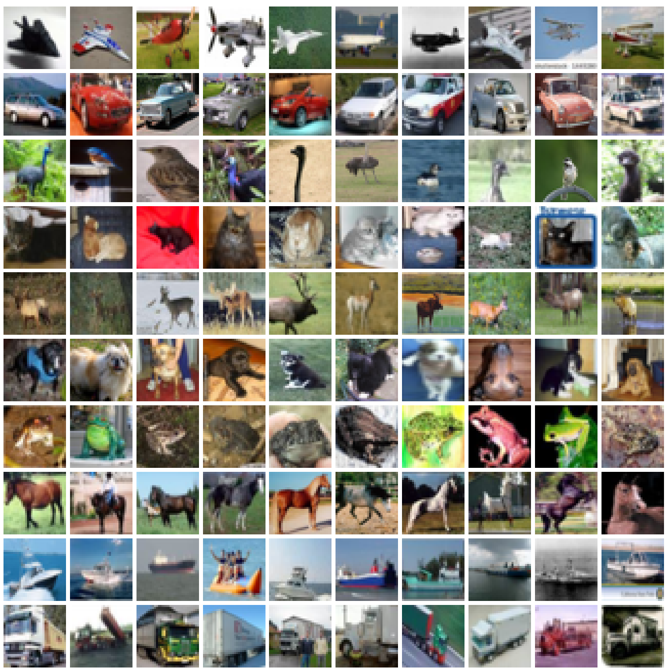
This does not differ much from networks from the 90s

This generalization allows to design complex networks of operators dealing with images, sound, text, sequences, etc. and to train them end-to-end.



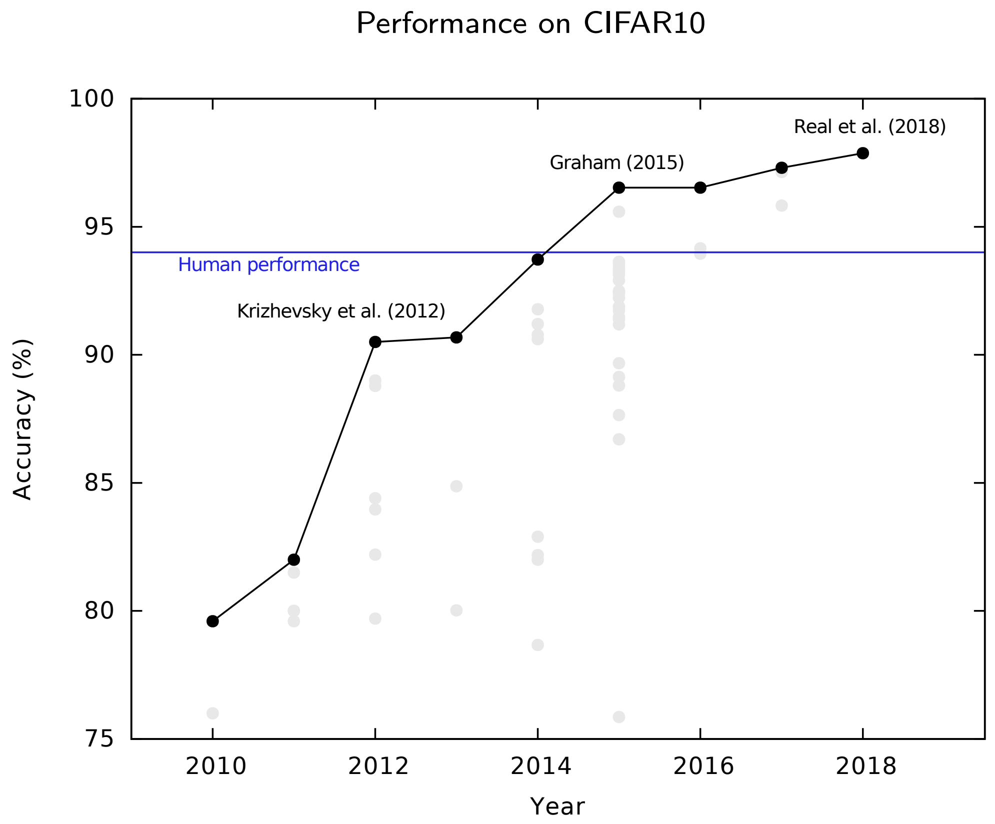
(Yeung et al., 2015)

CIFAR10



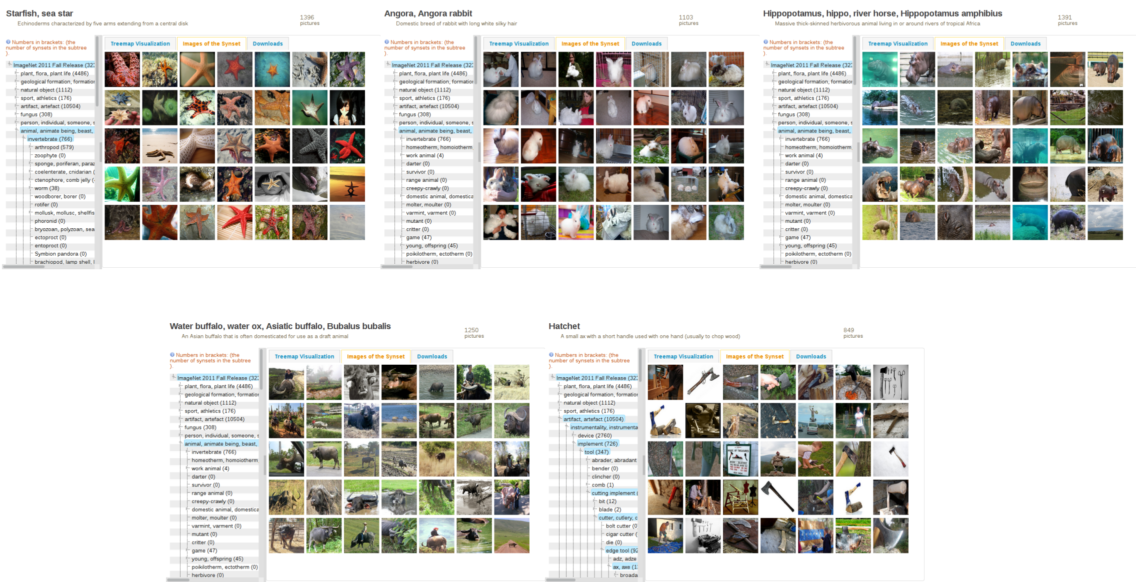
32×32 color images, 50k train samples, 10k test samples.

(Krizhevsky, 2009, chap. 3)



ImageNet Large Scale Visual Recognition Challenge.

1000 categories, > 1M images



(http://image-net.org/challenges/LSVRC/2014/browse-synsets)

method	top-1 err.	top-5 err.
VGG [41] (ILSVRC'14)	-	8.43†
GoogLeNet [44] (ILSVRC'14)	-	7.89
VGG [41] (v5)	24.4	7.1
PReLU-net [13]	21.59	5.71
BN-inception [16]	21.99	5.81
ResNet-34 B	21.84	5.71
ResNet-34 C	21.53	5.60
ResNet-50	20.74	5.25
ResNet-101	19.87	4.60
ResNet-152	**19.38**	**4.49**

Table 4. Error rates (%) of **single-model** results on the ImageNet validation set (except † reported on the test set).

method	top-5 err. (**test**)
VGG [41] (ILSVRC'14)	7.32
GoogLeNet [44] (ILSVRC'14)	6.66
VGG [41] (v5)	6.8
PReLU-net [13]	4.94
BN-inception [16]	4.82
ResNet (ILSVRC'15)	**3.57**

Table 5. Error rates (%) of **ensembles**. The top-5 error is on the test set of ImageNet and reported by the test server.

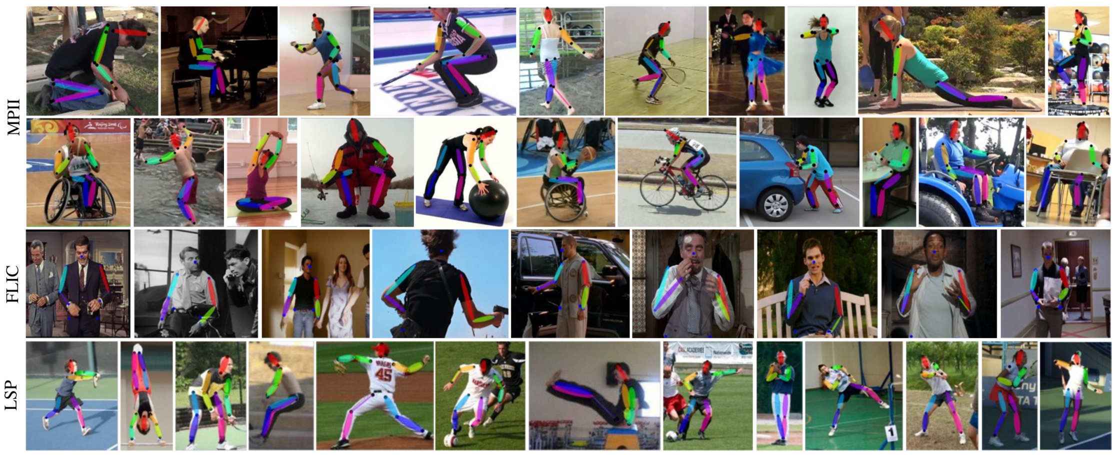
(He et al., 2015)

Current application domains

Object detection and segmentation

(Pinheiro et al., 2016)

Human pose estimation



(Wei et al., 2016)

Image generation

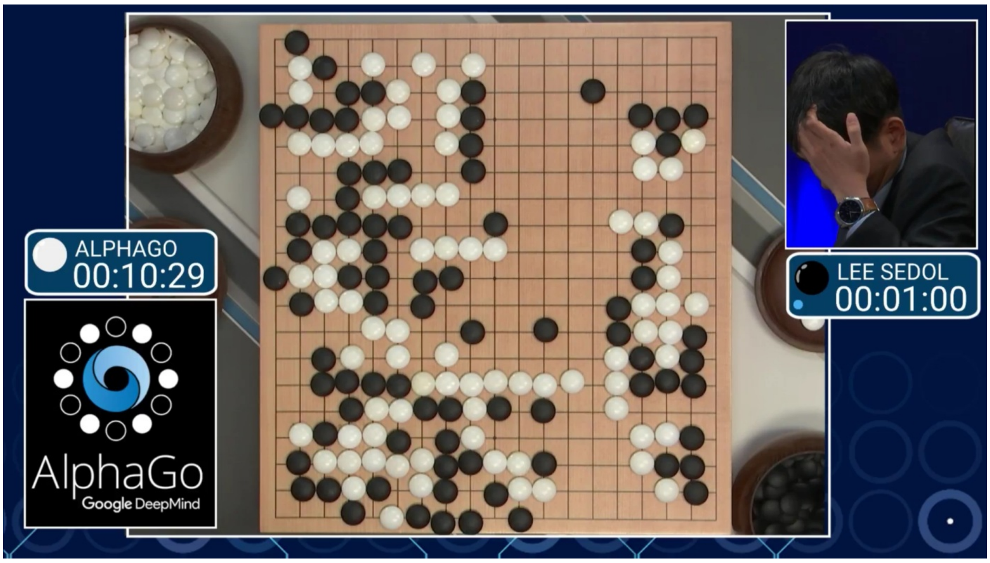
(Radford et al., 2015)

Reinforcement learning

Self-trained, plays 49 games at human level.

(Mnih et al., 2015)

Strategy games



March 2016, 4-1 against a 9-dan professional without handicap.

(Silver et al., 2016)

Translation

"The reason Boeing are doing this is to cram more seats in to make their plane more competitive with our products," said Kevin Keniston, head of passenger comfort at Europe's Airbus.

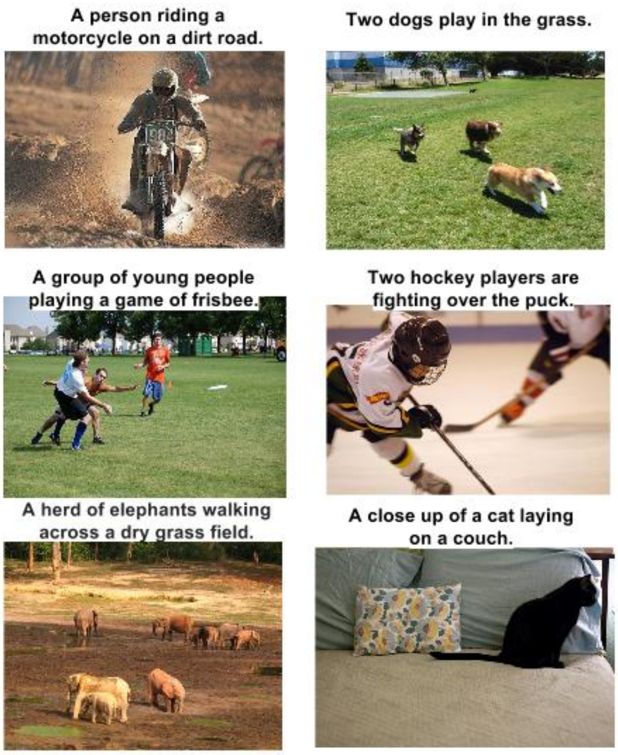
→ "La raison pour laquelle Boeing fait cela est de créer plus de sièges pour rendre son avion plus compétitif avec nos produits", a déclaré Kevin Keniston, chef du confort des passagers chez Airbus.

When asked about this, an official of the American administration replied: "The United States is not conducting electronic surveillance aimed at offices of the World Bank and IMF in Washington."

→ Interrogé à ce sujet, un fonctionnaire de l'administration américaine a répondu: "Les États-Unis n'effectuent pas de surveillance électronique à l'intention des bureaux de la Banque mondiale et du FMI à Washington"

(Wu et al., 2016)

Auto-captioning



(Vinyals et al., 2015)

Question answering

```
I: Jane went to the hallway.
I: Mary walked to the bathroom.
I: Sandra went to the garden.
I: Daniel went back to the garden.
I: Sandra took the milk there.
Q: Where is the milk?
A: garden

I: It started boring, but then it got interesting.
Q: What's the sentiment?
A: positive
```

(Kumar et al., 2015)

Why does it work now?

The success of deep learning is multi-factorial:

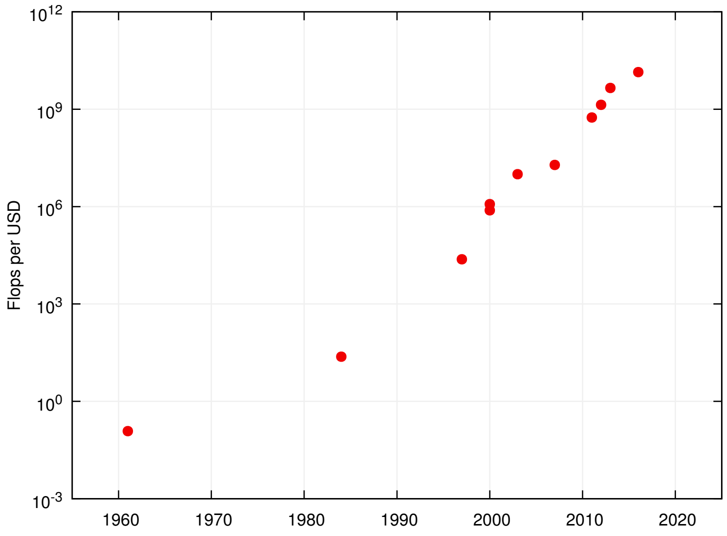
- Five decades of research in machine learning,
- CPUs/GPUs/storage developed for other purposes,
- lots of data from "the internet",
- tools and culture of collaborative and reproducible science,
- resources and efforts from large corporations.

Five decades of research in ML provided

- a taxonomy of ML concepts (classification, generative models, clustering, kernels, linear embeddings, etc.),
- a sound statistical formalization (Bayesian estimation, PAC),
- a clear picture of fundamental issues (bias/variance dilemma, VC dimension, generalization bounds, etc.),
- a good understanding of optimization issues,
- efficient large-scale algorithms.

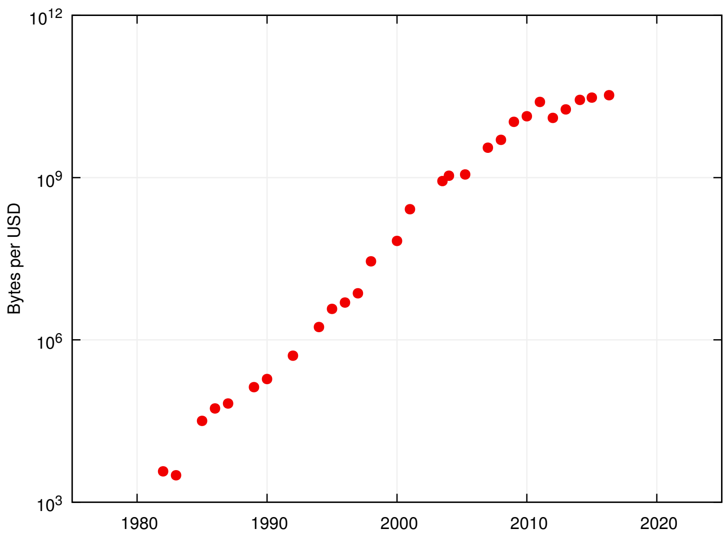
From a practical perspective, deep learning

- lessens the need for a deep mathematical grasp,
- makes the design of large learning architectures a system/software development task,
- allows to leverage modern hardware (clusters of GPUs),
- does not plateau when using more data,
- makes large trained networks a commodity.



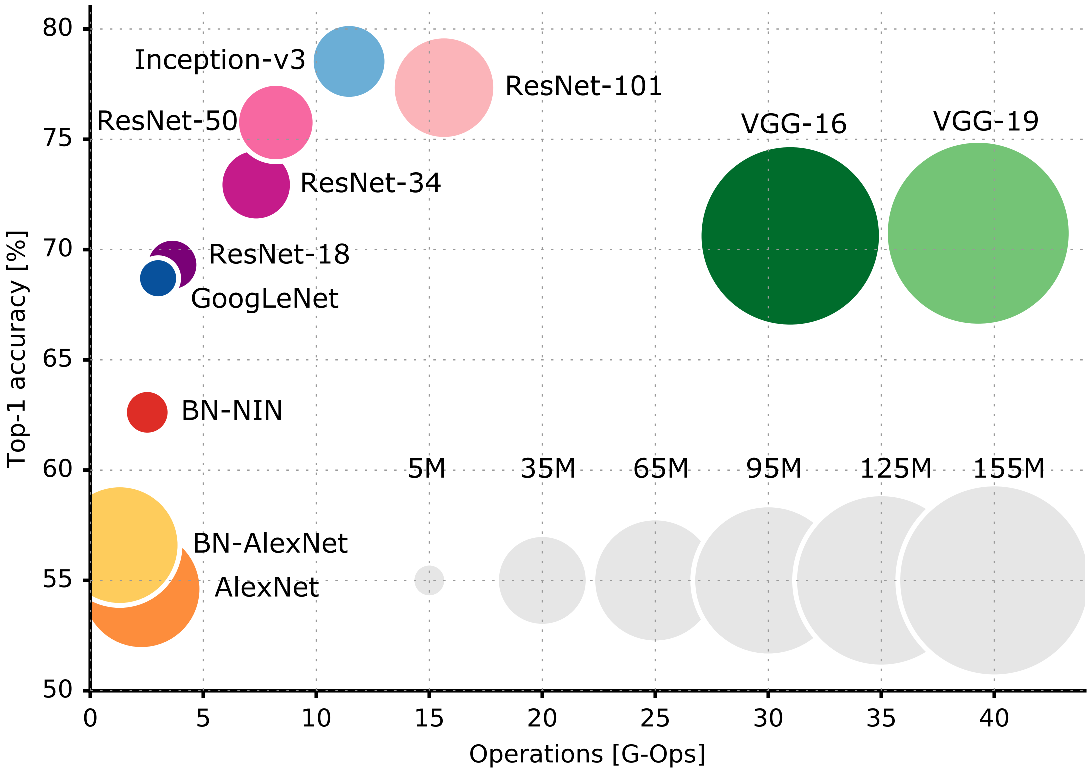
(Wikipedia "FLOPS")

	TFlops (10^{12})	Price	GFlops per $
Intel i7-6700K	0.2	$344	0.6
AMD Radeon R-7 240	0.5	$55	9.1
NVIDIA GTX 750 Ti	1.3	$105	12.3
AMD RX 480	5.2	$239	21.6
NVIDIA GTX 1080	8.9	$699	12.7



(John C. McCallum)

The typical cost of a 4Tb hard disk is $120 (Dec 2016).



(Canziani et al., 2016)

Data-set	Year	Nb. images	Resolution	Nb. classes
MNIST	1998	6.0×10^4	28×28	10
NORB	2004	4.8×10^4	96×96	5
Caltech 101	2003	9.1×10^3	$\simeq 300 \times 200$	101
Caltech 256	2007	3.0×10^4	$\simeq 640 \times 480$	256
LFW	2007	1.3×10^4	250×250	–
CIFAR10	2009	6.0×10^4	32×32	10
PASCAL VOC	2012	2.1×10^4	$\simeq 500 \times 400$	20
MS-COCO	2015	2.0×10^5	$\simeq 640 \times 480$	91
ImageNet	2016	14.2×10^6	$\simeq 500 \times 400$	$21,841$
Cityscape	2016	25×10^3	$2,000 \times 1000$	30

"Quantity has a Quality All Its Own."

(Thomas A. Callaghan Jr.)

Implementing a deep network, PyTorch

Deep-learning development is usually done in a framework:

	Language(s)	License	Main backer
PyTorch	**Python**	BSD	Facebook
Caffe2	C++, Python	Apache	Facebook
TensorFlow	Python, C++	Apache	Google
MXNet	Python, C++, R, Scala	Apache	Amazon
CNTK	Python, C++	MIT	Microsoft
Torch	Lua	BSD	Facebook
Theano	Python	BSD	U. of Montreal
Caffe	C++	BSD 2 clauses	U. of CA, Berkeley

A fast, low-level, compiled backend to access computation devices, combined with a slow, high-level, interpreted language.

We will use the PyTorch framework for our experiments.

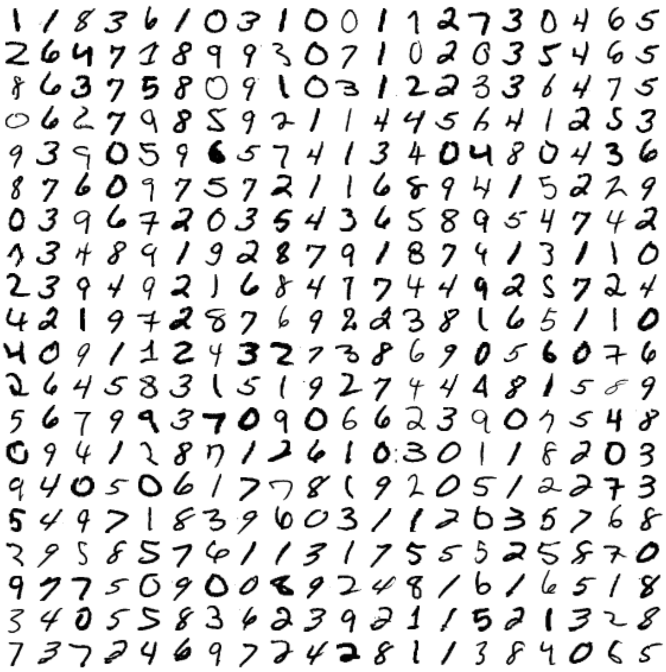
`http://pytorch.org`

"PyTorch is a python package that provides two high-level features:

- *Tensor computation (like numpy) with strong GPU acceleration*
- *Deep Neural Networks built on a tape-based autograd system*

You can reuse your favorite python packages such as numpy, scipy and Cython to extend PyTorch when needed."

MNIST data-set



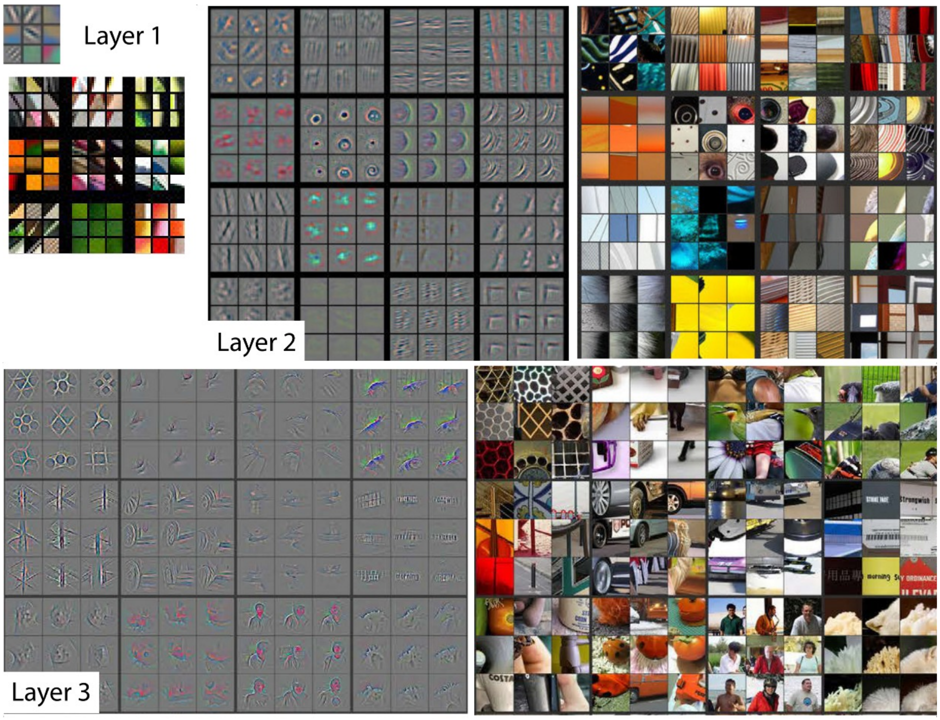
28×28 grayscale images, 60k train samples, 10k test samples.

(leCun et al., 1998)

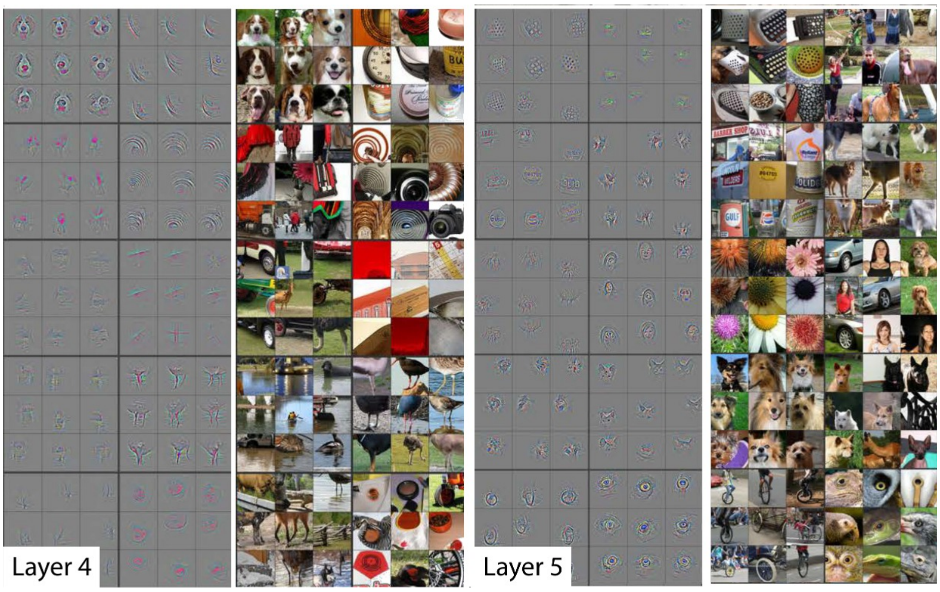

```
class Net(nn.Module):
    def __init__(self):
        super(Net, self).__init__()
        self.conv1 = nn.Conv2d(1, 32, kernel_size=5)
        self.conv2 = nn.Conv2d(32, 64, kernel_size=5)
        self.fc1 = nn.Linear(256, 200)
        self.fc2 = nn.Linear(200, 10)

    def forward(self, x):
        x = F.relu(F.max_pool2d(self.conv1(x), kernel_size=3))
        x = F.relu(F.max_pool2d(self.conv2(x), kernel_size=2))
        x = x.view(-1, 256)
        x = F.relu(self.fc1(x))
        x = self.fc2(x)
        return x

model = Net()

mu, std = train_input.data.mean(), train_input.data.std()
train_input.data.sub_(mu).div_(std)

optimizer = optim.SGD(model.parameters(), lr = 1e-1)
criterion, bs = nn.CrossEntropyLoss(), 100

model.cuda()
criterion.cuda()
train_input, train_target = train_input.cuda(), train_target.cuda()

for e in range(10):
    for b in range(0, nb_train_samples, bs):
        output = model(train_input.narrow(0, b, bs))
        loss = criterion(output, train_target.narrow(0, b, bs))
        model.zero_grad()
        loss.backward()
        optimizer.step()
```

$\simeq$7s on a GTX1080, $\simeq$1% test error

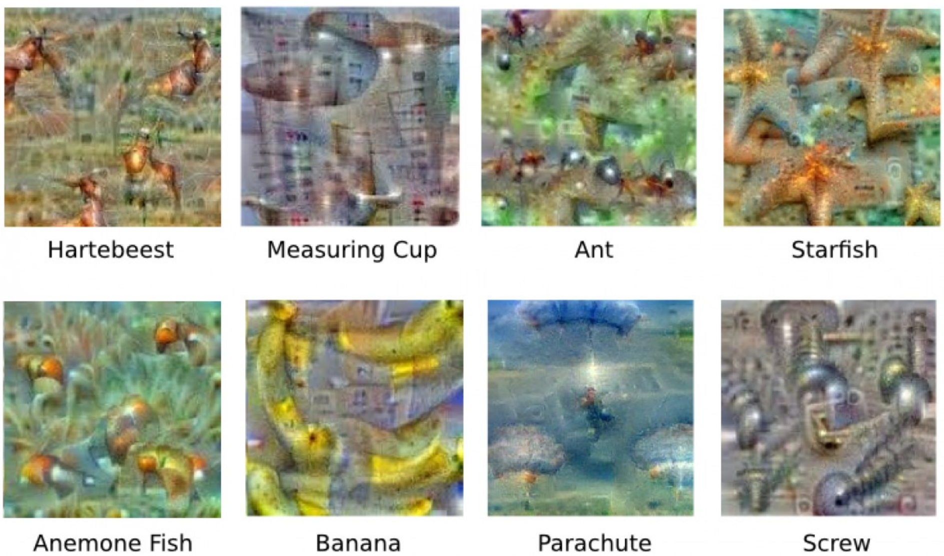
What is really happening?



(Zeiler and Fergus, 2014)



(Zeiler and Fergus, 2014)



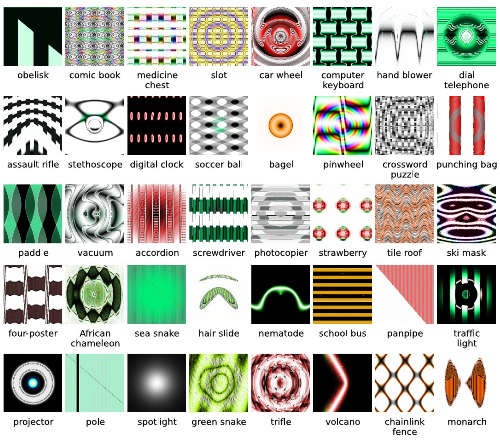
(Google's Deep Dreams)

(Google's Deep Dreams)

(Thorne Brandt)

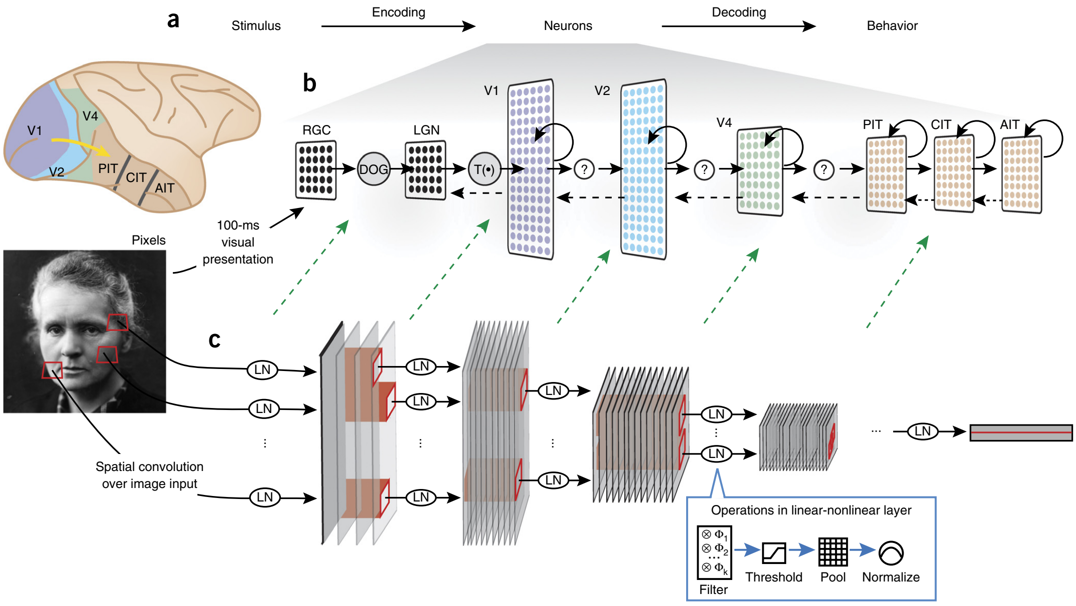
(Duncan Nicoll)

(Szegedy et al., 2014)

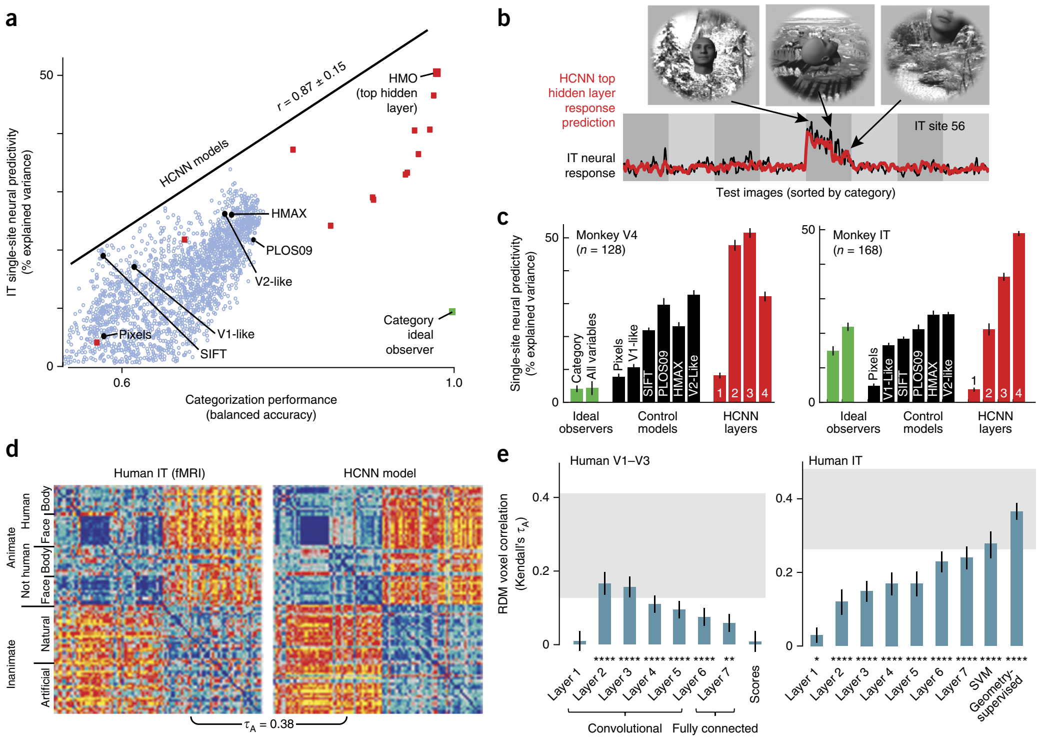


(Nguyen et al., 2015)

Relations with the biology



(Yamins and DiCarlo, 2016)



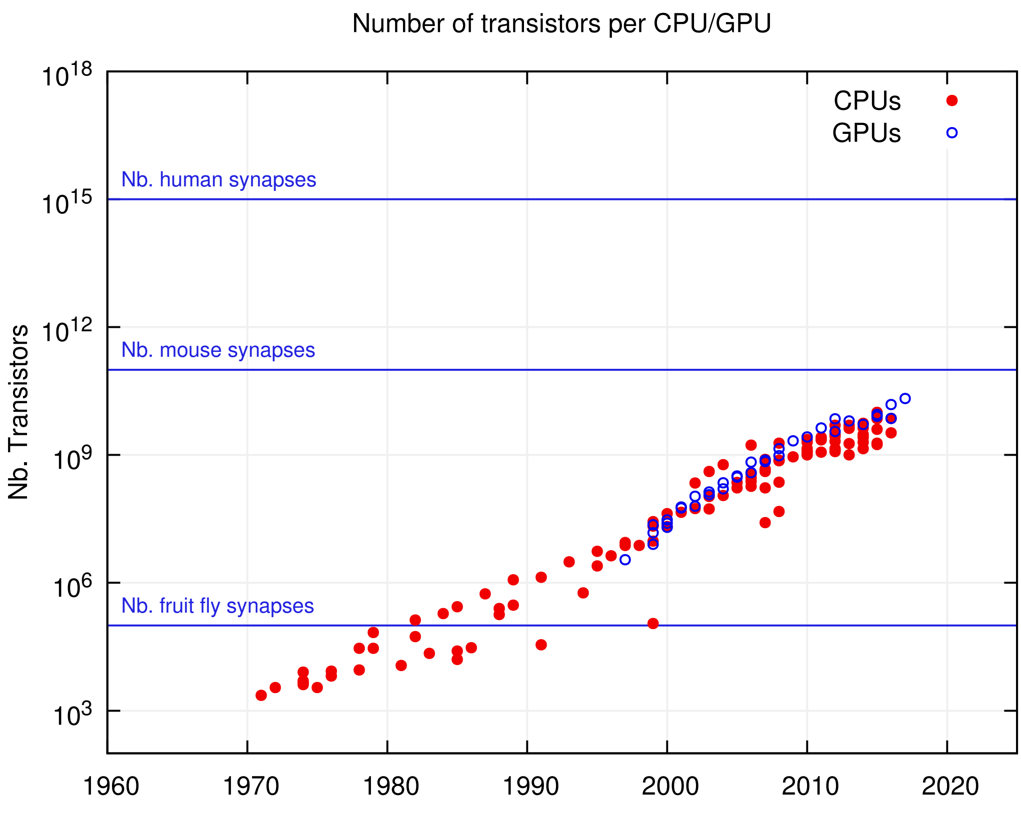
(Yamins and DiCarlo, 2016)

Species	Nb. neurons	Nb. synapses
Roundworm	302	7.5×10^3
Jellyfish	800	
Sea slug	1.8×10^4	
Fruit fly	1.0×10^5	1.0×10^7
Ant	2.5×10^5	
Cockroach	1.0×10^6	
Frog	1.6×10^7	
Mouse	7.1×10^7	1.0×10^{11}
Rat	2.0×10^8	4.5×10^{11}
Octopus	3.0×10^8	
Human	8.6×10^{10}	1.0×10^{15}

(Wikipedia "List of animals by number of neurons")

Device	Nb. transistors
Intel i7 Haswell-E (8 cores)	2.6×10^9
Intel Xeon Broadwell-E5 (22 cores)	7.2×10^9
AMD Epyc (32 cores)	19.2×10^9
Nvidia GeForce GTX 1080	7.2×10^9
AMD Vega 10	12.5×10^9
NVidia GV100	21.1×10^9

(Wikipedia "Transistor count")



(Wikipedia "Transistor count")

Plan, pre-requisites and grading

Lecture content:

1. Introduction.
2. Standard machine-learning concepts and tools.
3. Multi-layer perceptrons, back-prop, stochastic gradient descent.
4. Convolutional networks, arbitrary graphs of operators.
5. Initialization, optimization, and regularization.
6. Going deeper.
7. Deep models for Computer Vision.
8. Analysis of deep models.
9. Auto-encoders, embeddings, and generative models.
10. Generative adversarial networks.
11. Recurrent models, memory networks, NLP.
12. **Invited speaker (Soumith Chintala, Facebook).**
13. **Invited lecture (Andreas Steiner, Google).**
14. **Invited lecture (Andreas Steiner, Google).**

Pre-requisites:

- Linear algebra (vector and Euclidean spaces),
- differential calculus (gradient, Jacobian, Hessian, chain rule),
- Python programming,

... but there is more!

- basics in probabilities and statistics (discrete and continuous distributions, law of large numbers, conditional probabilities, Bayes, PCA),
- basics in optimization (notion of minima, gradient descent),
- basics in algorithmic (computational costs),
- basics in signal processing (Fourier transform, wavelets).

The evaluation will be:

50% – One mini-project, by groups of one to three students. Group report and source code, 5 min oral for each student/project.

50% – Final written exam.

References

A. Canziani, A. Paszke, and E. Culurciello. An analysis of deep neural network models for practical applications. *CoRR*, abs/1605.07678, 2016.

K. Fukushima. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological Cybernetics*, 36(4): 193–202, April 1980.

K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. *CoRR*, abs/1512.03385, 2015.

A. Krizhevsky. Learning multiple layers of features from tiny images. Master's thesis, Department of Computer Science, University of Toronto, 2009.

A. Krizhevsky, I. Sutskever, and G. Hinton. Imagenet classification with deep convolutional neural networks. In *Neural Information Processing Systems (NIPS)*, 2012.

A. Kumar, O. Irsoy, J. Su, J. Bradbury, R. English, B. Pierce, P. Ondruska, I. Gulrajani, and R. Socher. Ask me anything: Dynamic memory networks for natural language processing. *CoRR*, abs/1506.07285, 2015.

Y. leCun, L. Bottou, Y. Bengio, and P. Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

W. S. McCulloch and W. Pitts. A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4):115–133, 1943.

V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, Feb. 2015.

A. M. Nguyen, J. Yosinski, and J. Clune. Deep neural networks are easily fooled: High confidence predictions for unrecognizable images. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

P. O. Pinheiro, T.-Y. Lin, R. Collobert, and P. Dollár. Learning to refine object segments. In *European Conference on Computer Vision (ECCV)*, pages 75–91, 2016.

A. Radford, L. Metz, and S. Chintala. Unsupervised representation learning with deep convolutional generative adversarial networks. *CoRR*, abs/1511.06434, 2015.

D. E. Rumelhart, G. E. Hinton, and R. J. Williams. *Neurocomputing: Foundations of Research*, chapter Learning Representations by Back-propagating Errors, pages 696–699. MIT Press, 1988.

D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. van den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel, and D. Hassabis. Mastering the game of go with deep neural networks and tree search. *Nature*, 529:484–503, 2016.

C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus. Intriguing properties of neural networks. In *International Conference on Learning Representations (ICLR)*, 2014.

C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

O. Vinyals, A. Toshev, S. Bengio, and D. Erhan. Show and tell: A neural image caption generator. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

S. Wei, V. Ramakrishna, T. Kanade, and Y. Sheikh. Convolutional pose machines. *CoRR*, abs/1602.00134, 2016.

Y. Wu, M. Schuster, Z. Chen, Q. V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey, J. Klingner, A. Shah, M. Johnson, X. Liu, L. Kaiser, S. Gouws, Y. Kato, T. Kudo, H. Kazawa, K. Stevens, G. Kurian, N. Patil, W. Wang, C. Young, J. Smith, J. Riesa, A. Rudnick, O. Vinyals, G. Corrado, M. Hughes, and J. Dean. Google's neural machine translation system: Bridging the gap between human and machine translation. *CoRR*, abs/1609.08144, 2016.

D. L. K. Yamins and J. J. DiCarlo. Using goal-driven deep learning models to understand sensory cortex. *Nature neuroscience*, 19:356–65, Feb 2016.

S. Yeung, O. Russakovsky, G. Mori, and L. Fei-Fei. End-to-end learning of action detection from frame glimpses in videos. *CoRR*, abs/1511.06984, 2015.

M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. In *European Conference on Computer Vision (ECCV)*, 2014.