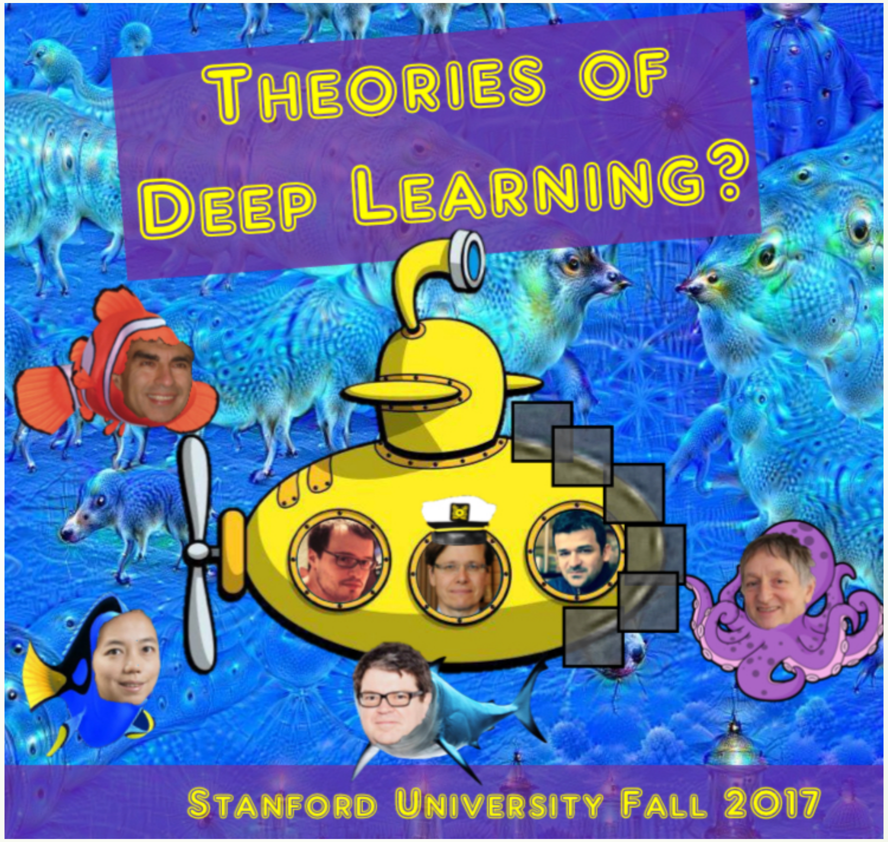
On Mathematical Theories of Deep Learning

Yuan YAO
HKUST

Acknowledgement



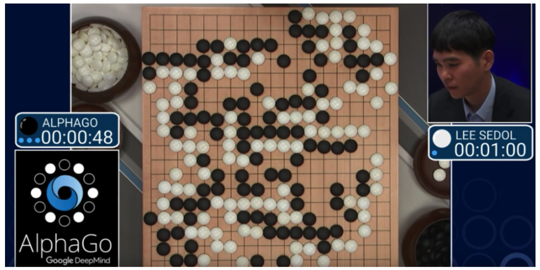
A following-up course at HKUST: https://deeplearning-math.github.io/

Outline

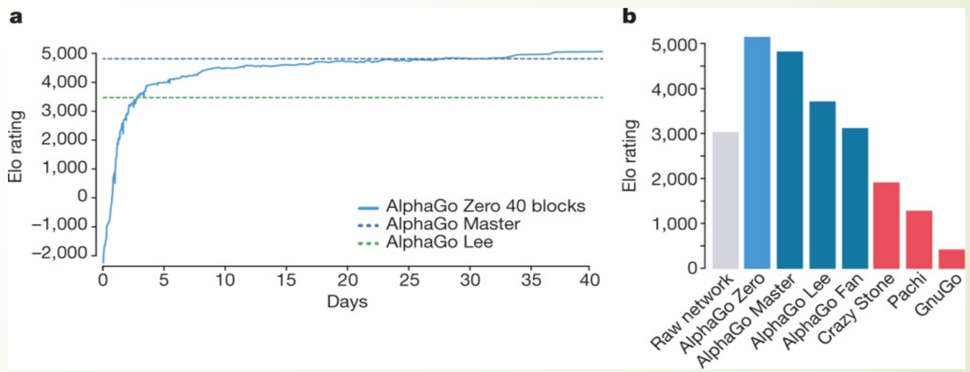
- Why mathematical theories of Deep Learning?
  - The tsunami of deep learning in recent years…
- What Theories Do We Have or Need?
  - Harmonic Analysis: what are optimal representation of functions?
  - Approximation Theory: when deep networks are better than shallow ones?
  - Optimization: what are the landscapes of risk and how to efficiently find a good optimum?
  - Statistics: how deep net models can generalize well?

Reaching Human Performance Level

Deep Blue in 1997



AlphaGo “LEE” 2016



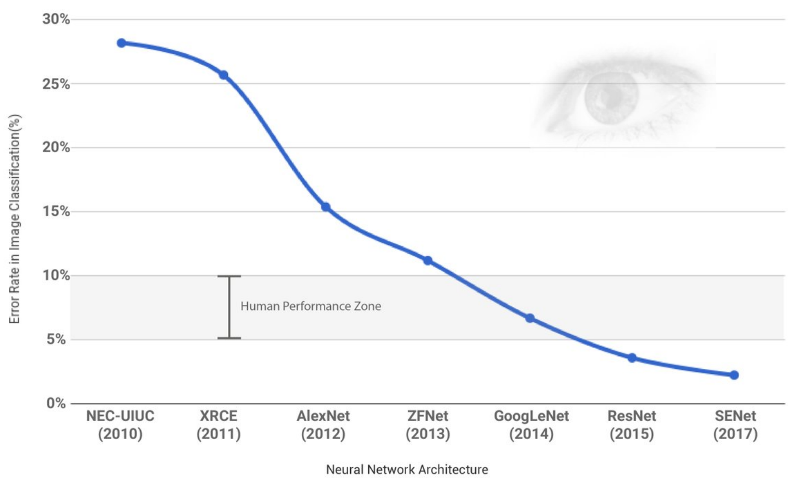
AlphaGo ”ZERO” D Silver *et al. Nature* **550,** 354–359 (2017) doi:10.1038/nature24270

ImageNet Dataset

- 14,197,122 labeled images
- 21,841 classes
- Labeling required more than a year of human effort via Amazon Mechanical Turk

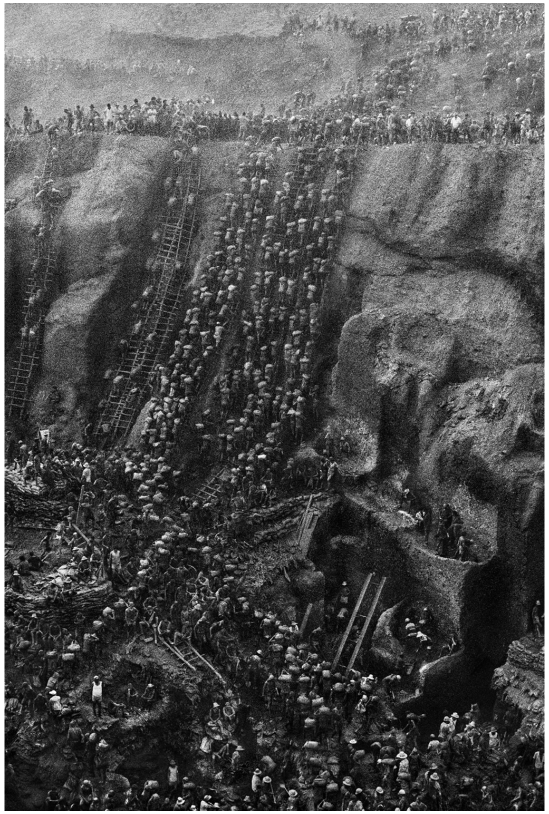
ImageNet Top 5 classification error

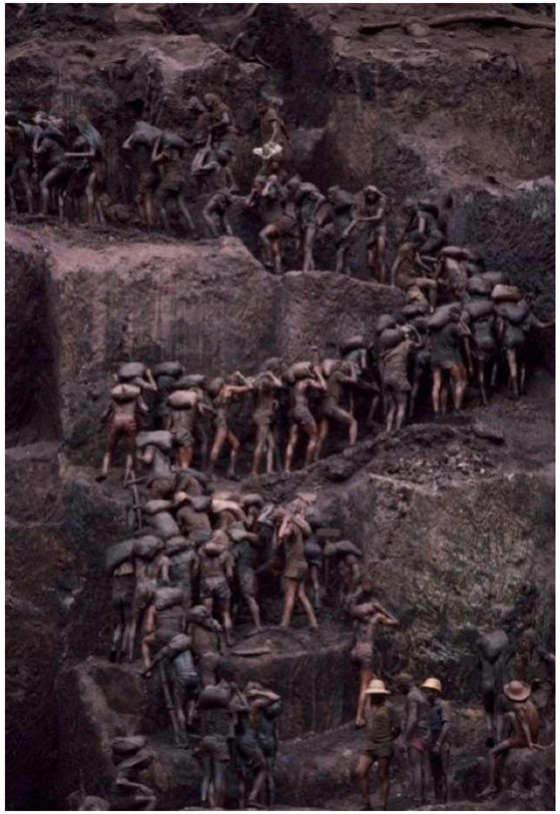
- ImageNet (subset):
  - 1.2 million training images
  - 100,000 test images
  - 1000 classes
- ImageNet large-scale visual recognition Challenge



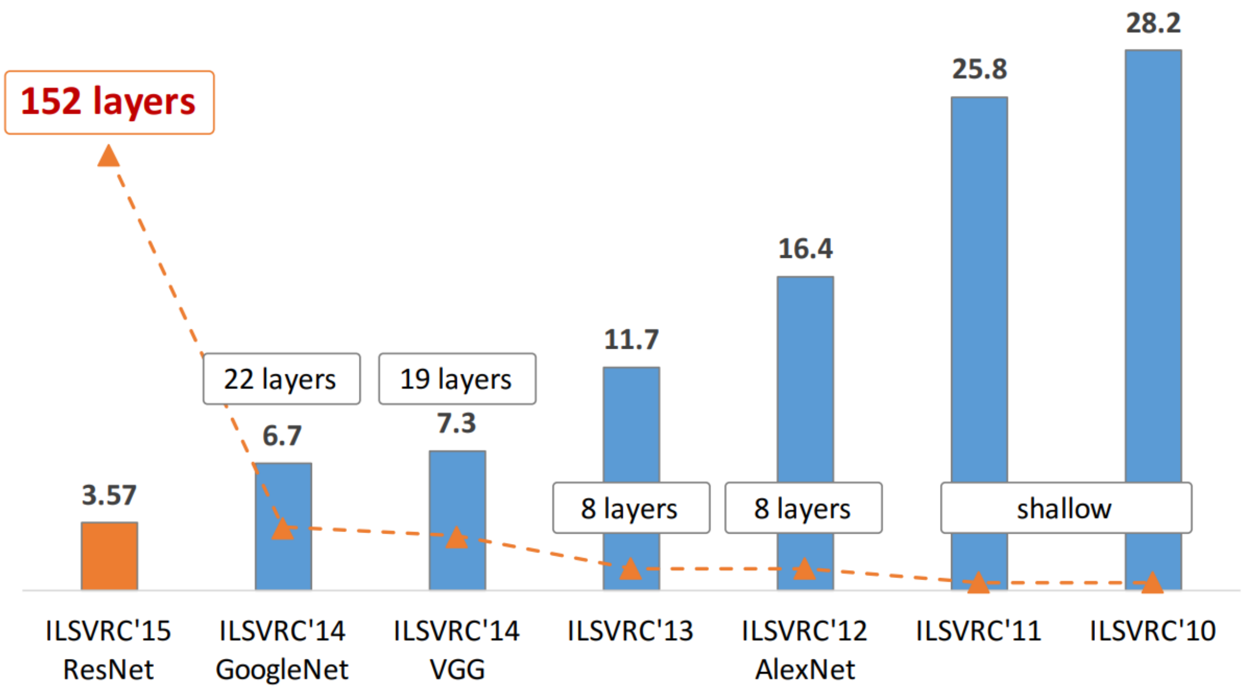
source: https://www.linkedin.com/pulse/must-read-path-breaking-papers-image-classification-muktabh-mayank

Crowdcomputing:
researchers raising the competition record



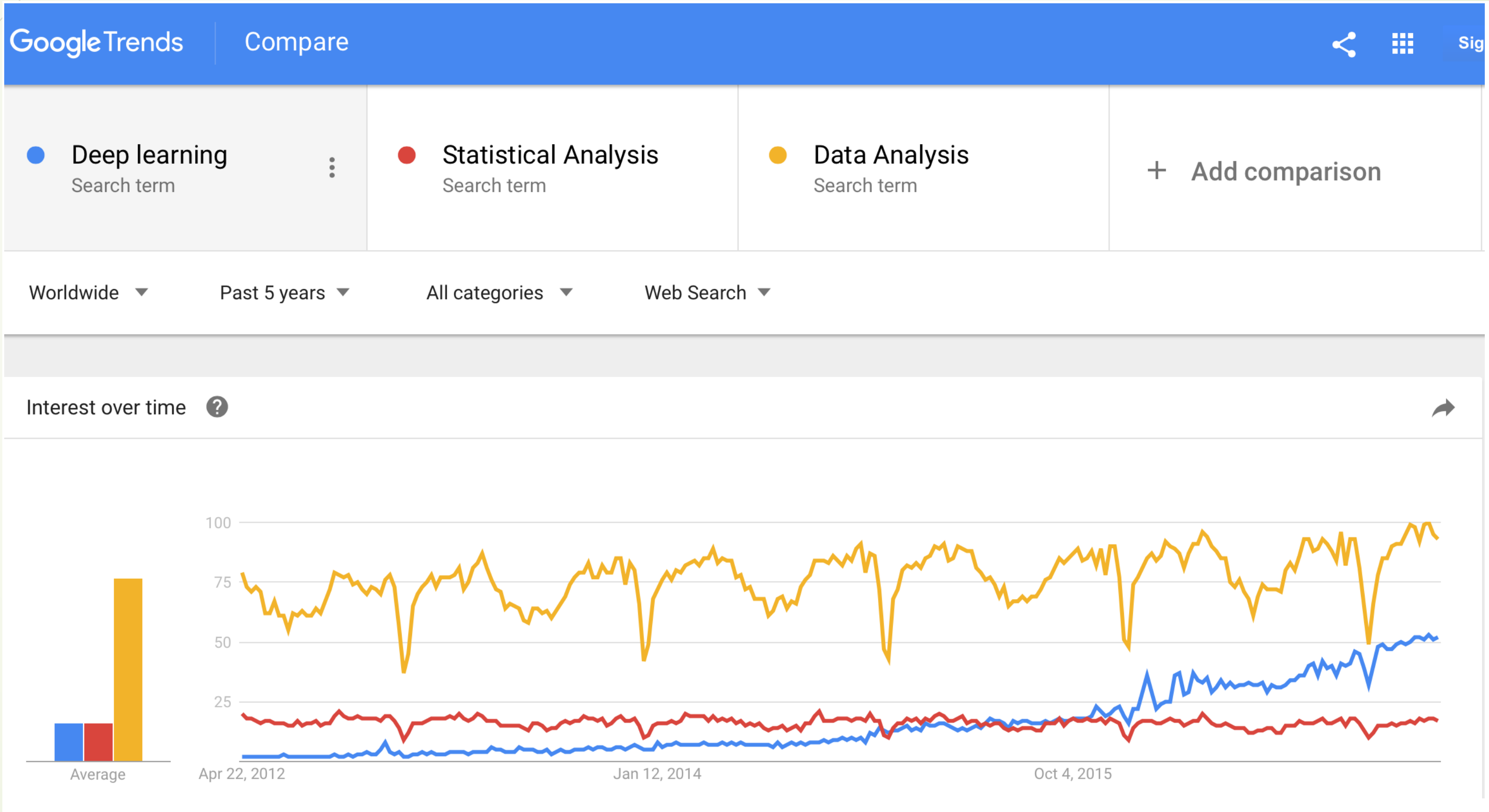


Depth as function of year



[He et al., 2016]

Growth of Deep Learning

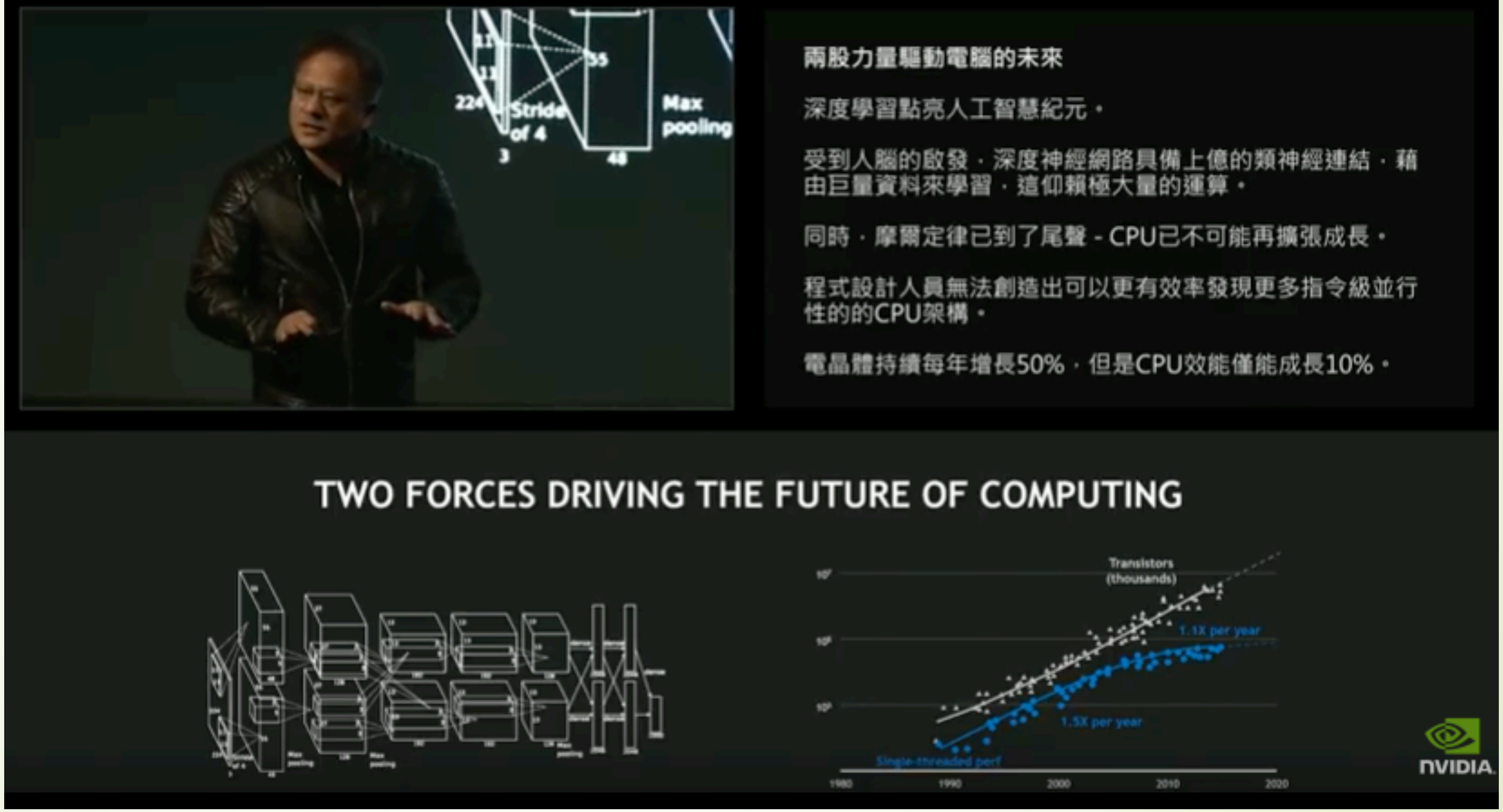


New Moore's Laws

CS231n attendance

NIPS registrations

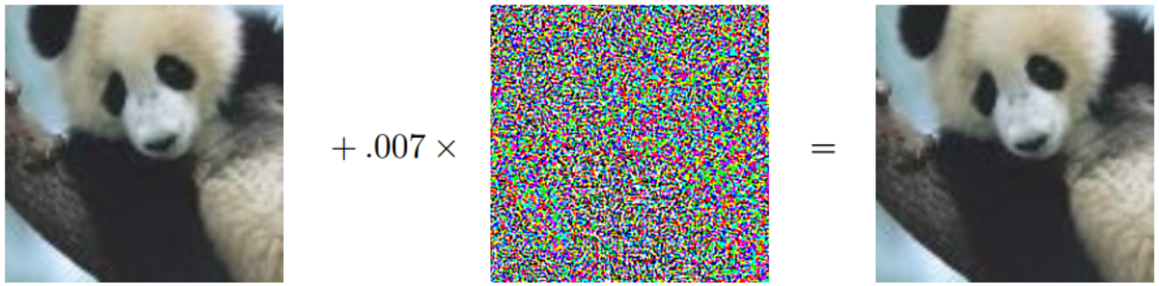
"We're at the beginning of a new day…
This is the beginning of the AI revolution."
— Jensen Huang, GTC Taiwan 2017



Some Cold Water: Tesla Autopilot Misclassifies Truck as Billboard

Problem: Why? How can you trust a blackbox?

Deep Learning may be fragile in generalization against noise!



$\boldsymbol{x}$ — "panda" 57.7% confidence

$\text{sign}(\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y))$ — "nematode" 8.2% confidence

$\boldsymbol{x} + \epsilon\text{sign}(\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y))$ — "gibbon" 99.3 % confidence

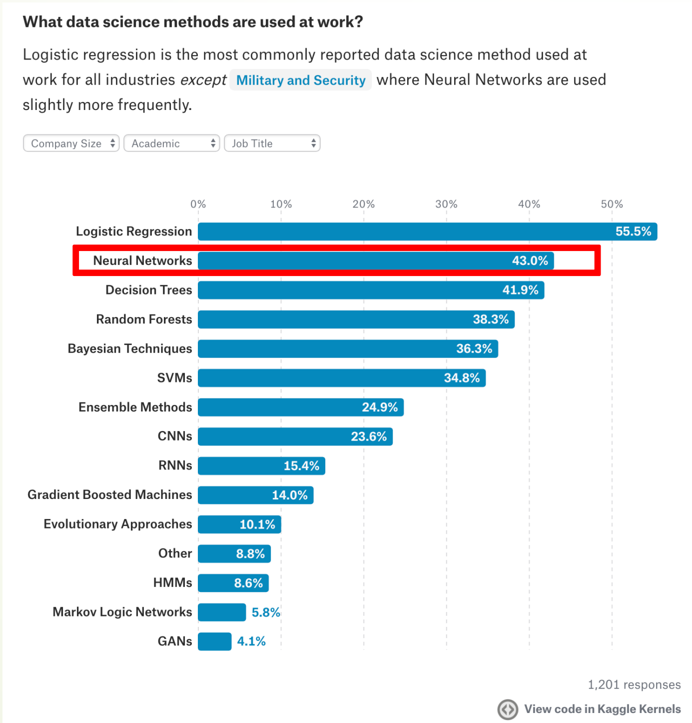
[Goodfellow et al., 2014]

- Small but malicious perturbations can result in severe misclassification
- Malicious examples generalize across different architectures
- What is source of instability?
- Can we robustify network?

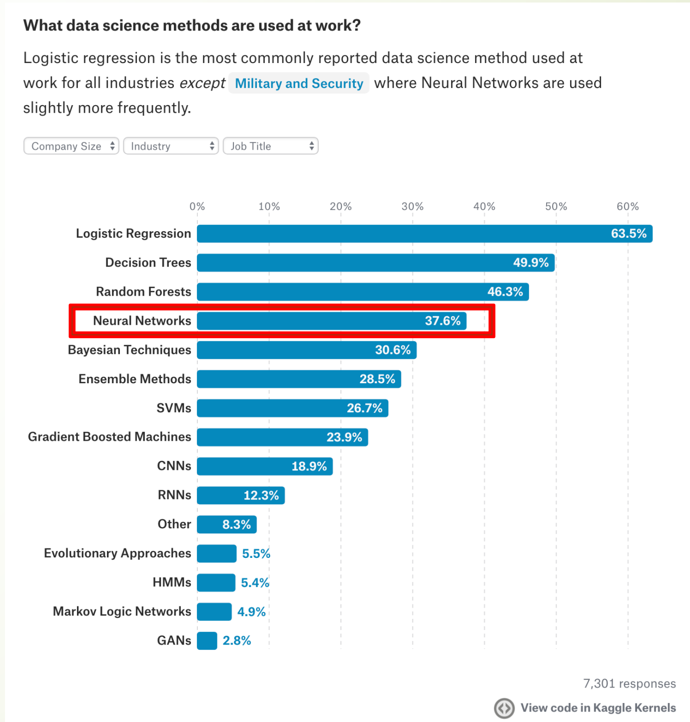
Kaggle survey: Top Data Science Methods

https://www.kaggle.com/surveys/2017

Academic



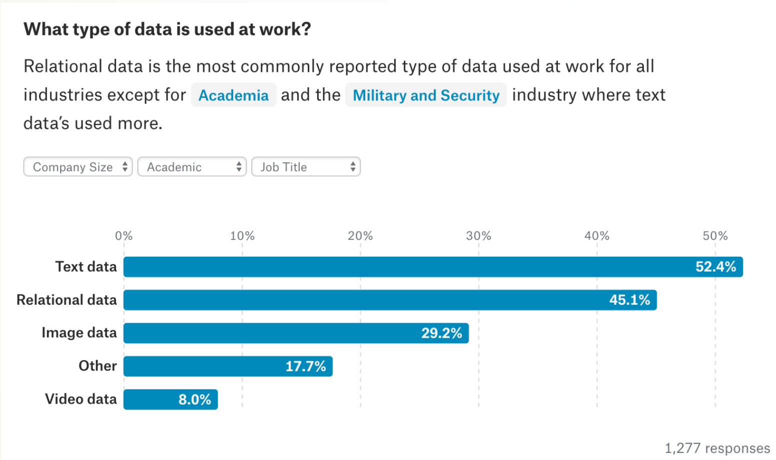
Industry



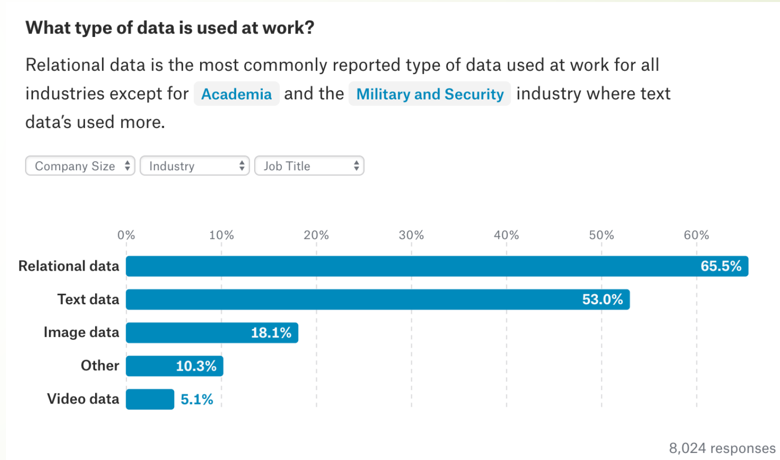
What type of data is used at work?

https://www.kaggle.com/surveys/2017

Academic



Industry



What's wrong with deep learning?

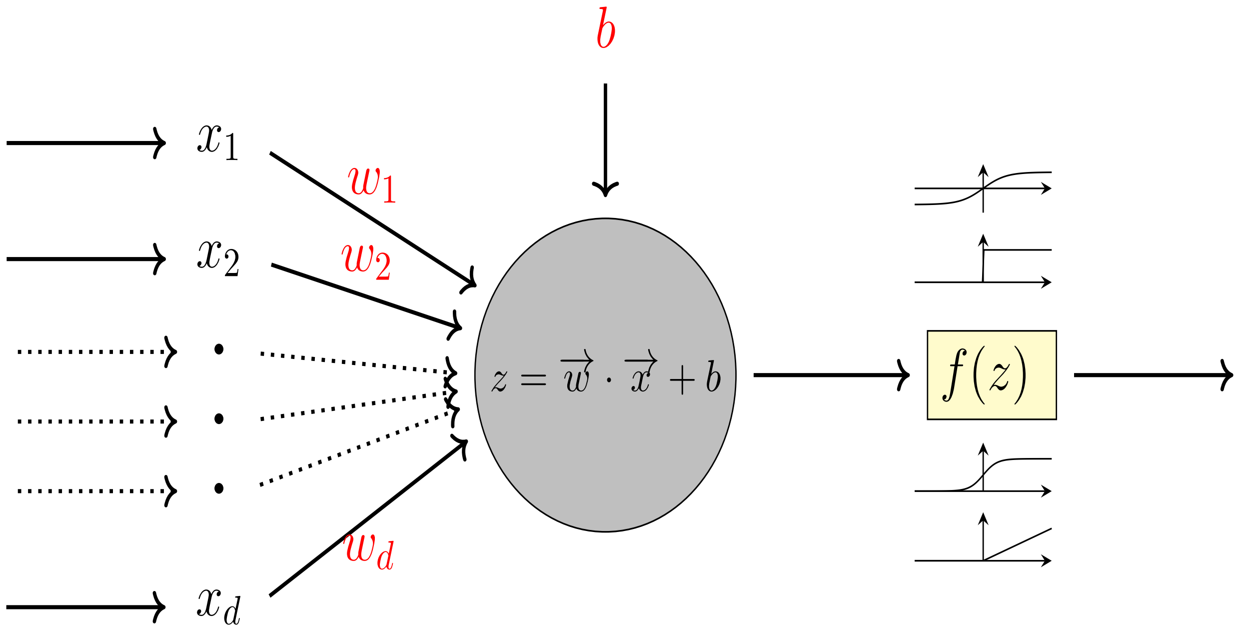
Ali Rahimi NIPS'17: Machine (deep) Learning has become **alchemy**.
https://www.youtube.com/watch?v=ORHFOnaEzPc

Yann LeCun CVPR'15, invited talk: **What's wrong with deep learning?**
One important piece: **missing some theory**!

http://techtalks.tv/talks/whats-wrong-with-deep-learning/61639/

Perceptron: single-layer

- Invented by Frank Rosenblatt (1957)



Locality or Sparsity of Computation

Minsky and Papert, 1969

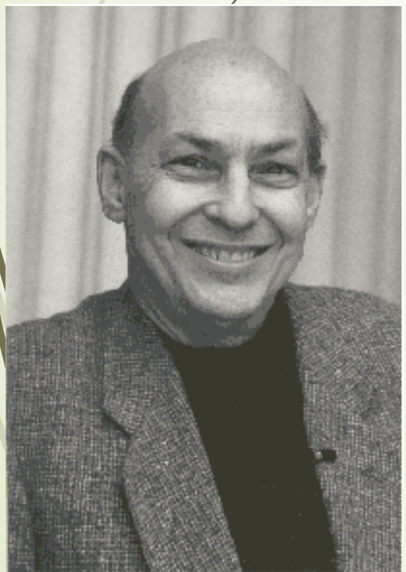
Perceptron can't do **XOR** classification

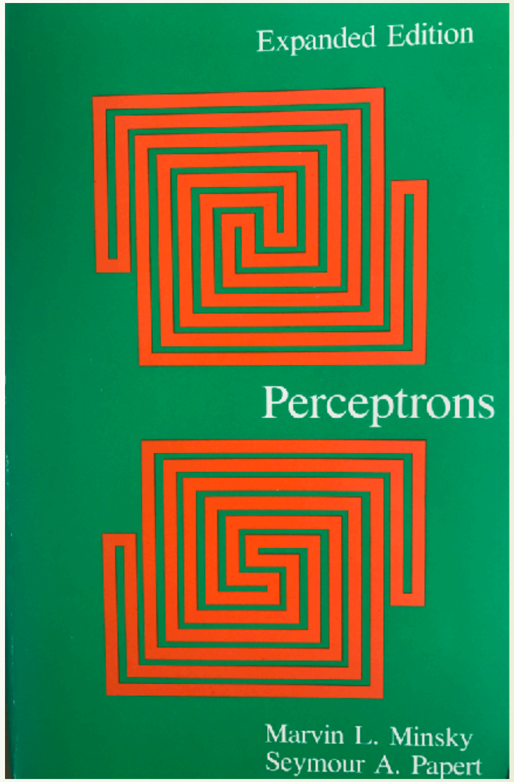
Perceptron needs infinite global information to compute **connectivity**

Locality or **Sparsity** is important:

Locality in time?

Locality in space?



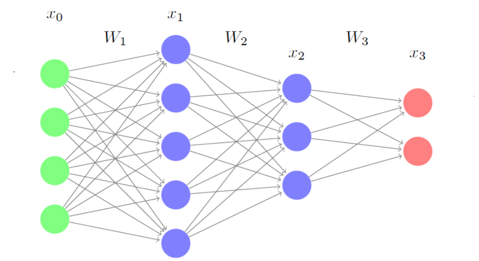


Multilayer Perceptrons (MLP) and Back-Propagation (BP) Algorithms

Rumelhart, Hinton, Williams (1986)
Learning representations by back-propagating errors, Nature, 323(9): 533-536

BP algorithms as **stochastic gradient descent** algorithms (**Robbins–Monro 1950; Kiefer-Wolfowitz 1951**) with Chain rules of Gradient maps

MLP classifies XOR, but the global hurdle on topology (connectivity) computation still exists



Learning representations by back-propagating errors

David E. Rumelhart*, Geoffrey E. Hinton† & Ronald J. Williams*

* Institute for Cognitive Science, C-015, University of California, San Diego, La Jolla, California 92093, USA
† Department of Computer Science, Carnegie-Mellon University, Pittsburgh, Philadelphia 15213, USA

We describe a new learning procedure, back-propagation, for networks of neurone-like units. The procedure repeatedly adjusts the weights of the connections in the network so as to minimize a measure of the difference between the actual output vector of the net and the desired output vector. As a result of the weight adjustments, internal 'hidden' units which are not part of the input or output come to represent important features of the task domain, and the regularities in the task are captured by the interactions of these units. The ability to create useful new features distinguishes back-propagation from earlier, simpler methods such as the perceptron-convergence procedure[1].

There have been many attempts to design self-organizing neural networks. The aim is to find a powerful synaptic modification rule that will allow an arbitrarily connected neural network to develop an internal structure that is appropriate for a particular task domain. The task is specified by giving the desired state vector of the output units for each state vector of the input units. If the input units are directly connected to the output units it is relatively easy to find learning rules that iteratively adjust the relative strengths of the connections so as to progressively reduce the difference between the actual and desired output vectors[2]. Learning becomes more interesting but more difficult when we introduce hidden units whose actual or desired states are not specified by the task. (In perceptrons, there are 'feature analysers' between the input and output that are not true hidden units because their input connections are fixed by hand, so their states are completely determined by the input vector: they do not learn representations.) The learning procedure must decide under what circumstances the hidden units should be active in order to help achieve the desired input–output behaviour. This amounts to deciding what these units should represent. We demonstrate that a general purpose and relatively simple procedure is powerful enough to construct appropriate internal representations.

The simplest form of the learning procedure is for layered networks which have a layer of input units at the bottom; any number of intermediate layers; and a layer of output units at the top. Connections within a layer or from higher to lower layers are forbidden, but connections can skip intermediate layers. An input vector is presented to the network by setting the states of the input units. Then the states of the units in each layer are determined by applying equations (1) and (2) to the connections coming from lower layers. All units within a layer have their states set in parallel, but different layers have their states set sequentially, starting at the bottom and working upwards until the states of the output units are determined.

The total input, x_j, to unit j is a linear function of the outputs, y_i, of the units that are connected to j and of the weights, w_{ji}, on these connections

$$x_j = \sum_i y_i w_{ji} \qquad (1)$$

Units can be given biases by introducing an extra input to each unit which always has a value of 1. The weight on this extra input is called the bias and is equivalent to a threshold of the opposite sign. It can be treated just like the other weights.

A unit has a real-valued output, y_j, which is a non-linear function of its total input

$$y_j = \frac{1}{1+e^{-x_j}} \qquad (2)$$

† To whom correspondence should be addressed

Convolutional Neural Networks: shift invariances and locality

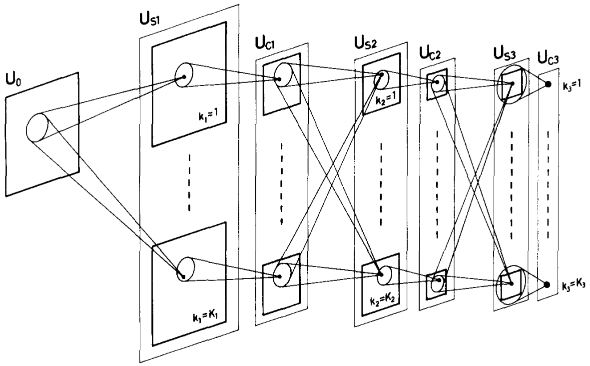
- Can be traced to *Neocognitron* of Kunihiko Fukushima (1979)
- Yann LeCun combined convolutional neural networks with back propagation (1989)
- Imposes **shift invariance** and **locality** on the weights
- Forward pass remains similar
- Backpropagation slightly changes – need to sum over the gradients from all spatial positions

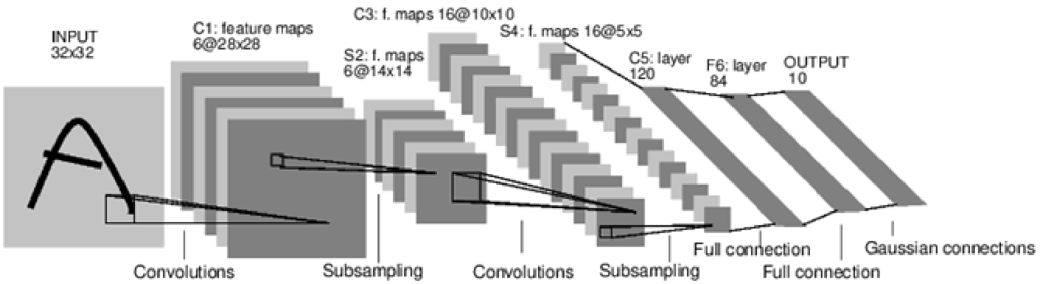
Biol. Cybernetics 36, 193–202 (1980)

Neocognitron: A Self-organizing Neural Network Model for a Mechanism of Pattern Recognition Unaffected by Shift in Position

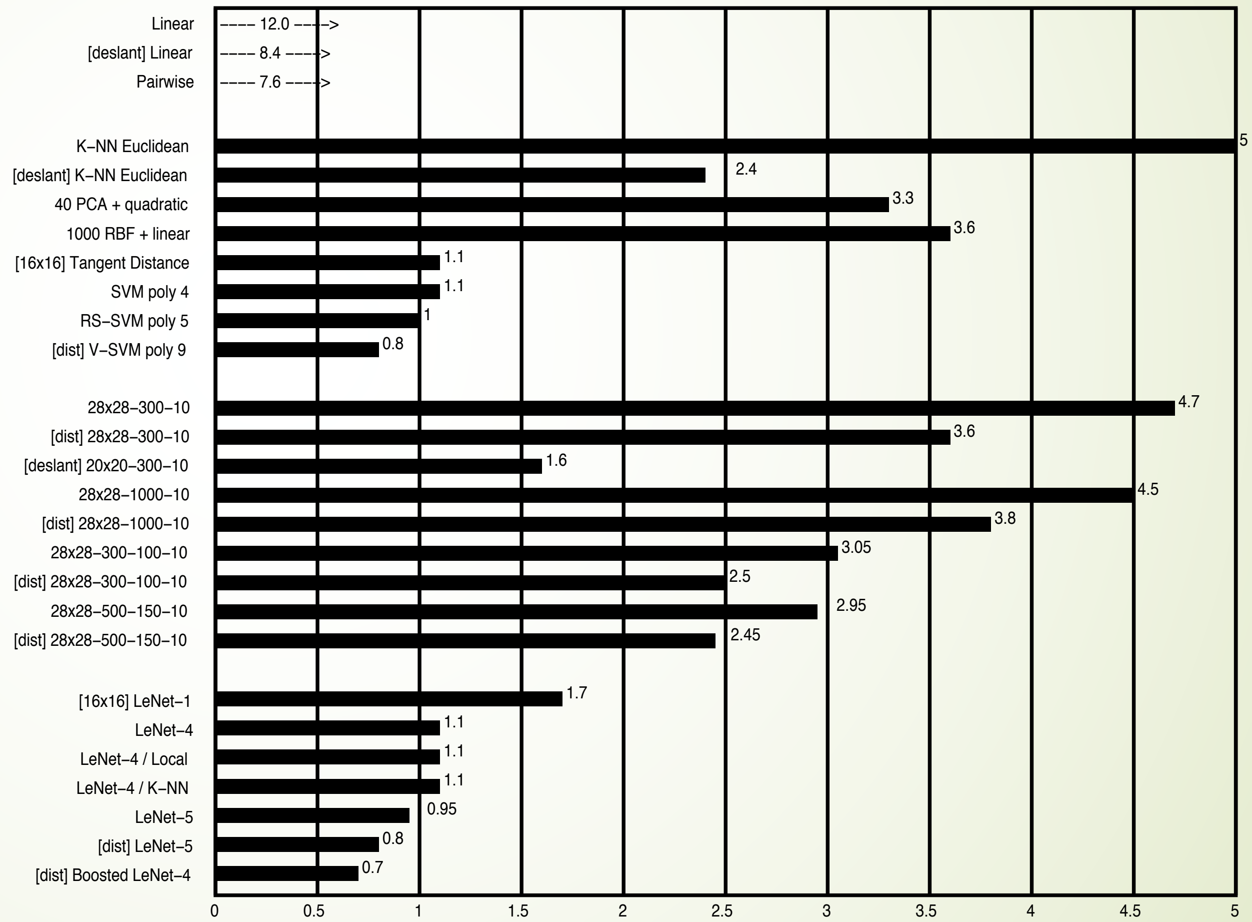
Kunihiko Fukushima

NHK Broadcasting Science Research Laboratories, Kinuta, Setagaya, Tokyo, Japan





MNIST Dataset Test Error
LeCun et al. 1998

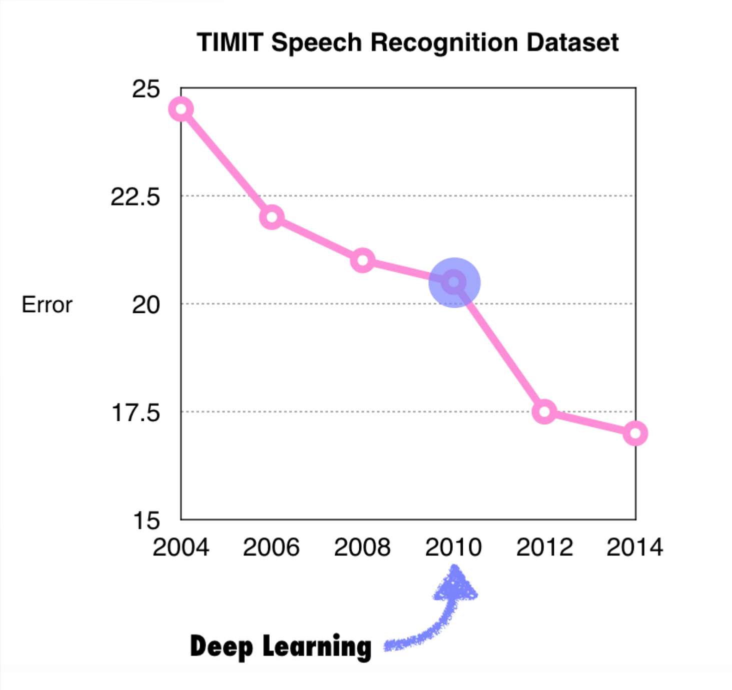


Simple SVM performs as well as Multilayer Convolutional Neural Networks which need careful tuning (LeNets)

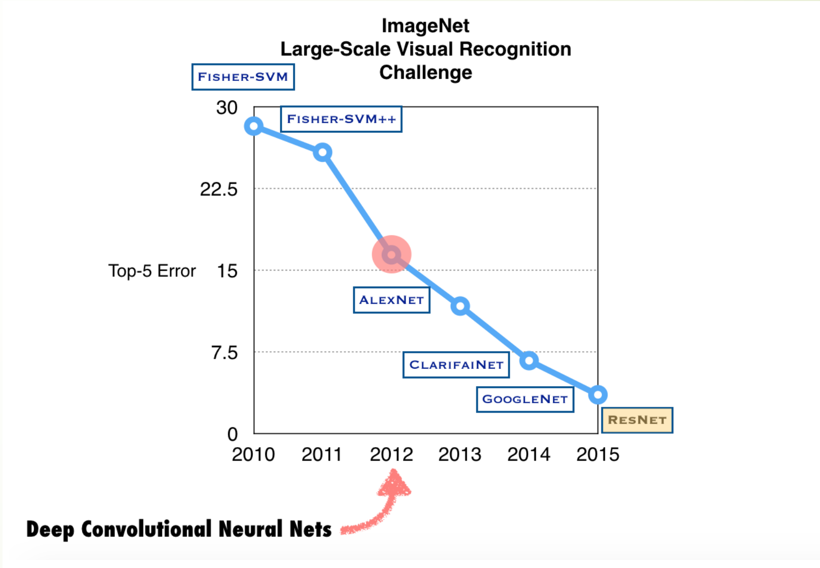
Dark era for NN: 1998-2012

Around the year of 2012…

Speech Recognition: TIMIT

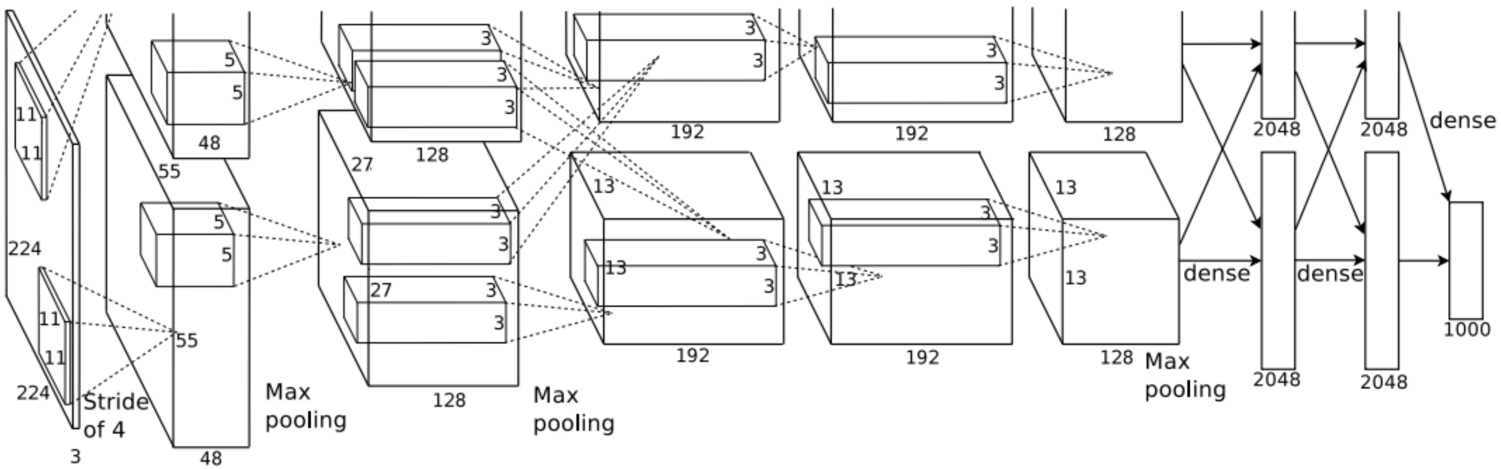


Computer Vision: ImageNet



AlexNet (2012)

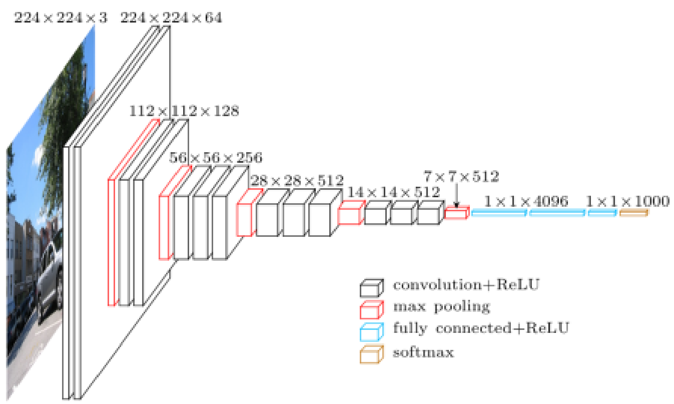
- 8 layers: first 5 convolutional, rest fully connected
- ReLU nonlinearity
- Local response normalization
- Max-pooling
- Dropout



Source: [Krizhevsky et al., 2012]

VGG (2014) [Simonyan-Zisserman'14]

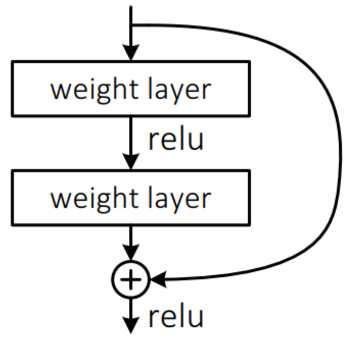
- Deeper than AlexNet: 11-19 layers versus 8
- No local response normalization
- Number of filters multiplied by two every few layers
- Spatial extent of filters 3×3 in all layers
- Instead of 7×7 filters, use three layers of 3×3 filters
  - Gain intermediate nonlinearity
  - Impose a regularization on the 7×7 filters

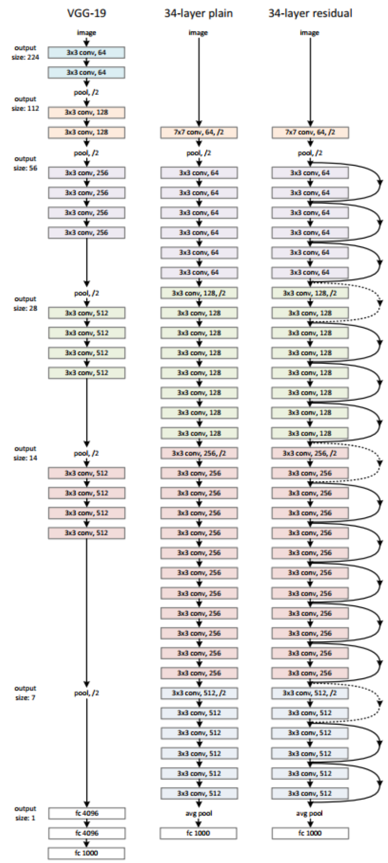


Source: `https://blog.heuritech.com/2016/02/29/`

ResNet (2015) [HGRS-15]

- Solves problem by adding skip connections
- Very deep: 152 layers
- No dropout
- Stride
- Batch normalization





Source: Deep Residual Learning for Image Recognition

Visualizing Deep Neural Networks

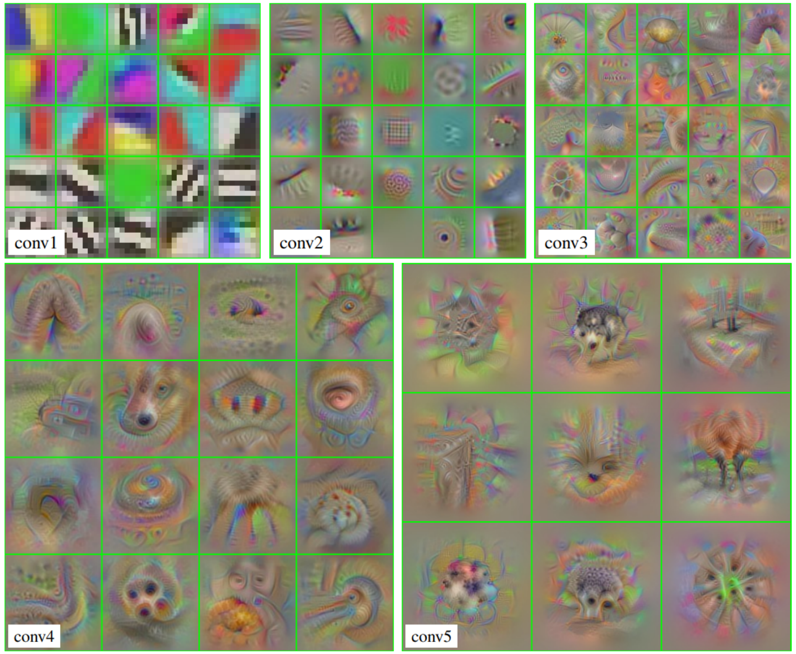
- Filters in first layer of CNN are easy to visualize, while deeper ones are harder
- *Activation maximization* seeks input image maximizing output of the i-th neuron in the network
- Objective

$$x^* = \arg\min_x \mathcal{R}(x) - \langle \Phi(x), e_i \rangle$$

- e_i is indicator vector
- $\mathcal{R}(x)$ is simple natural image prior

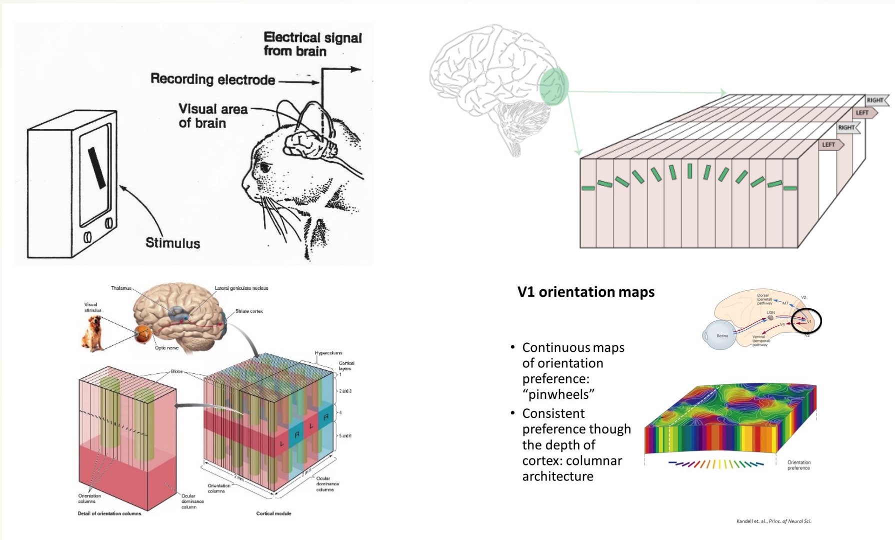
Visualizing VGG

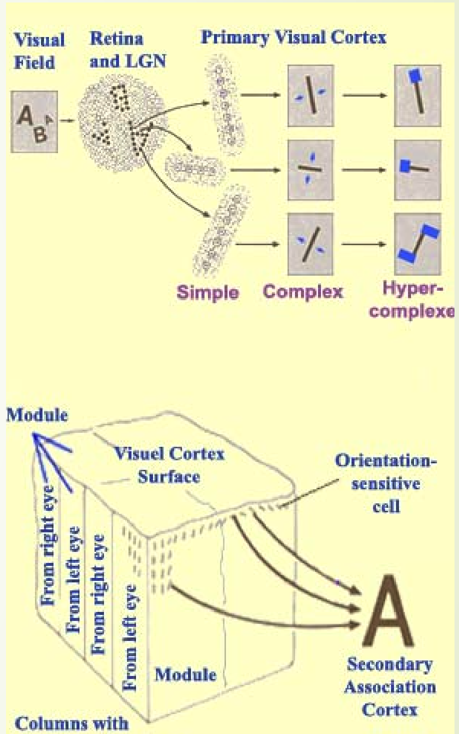
- Gabor-like images in first layer
- More sophisticated structures in the rest



[Mahendran and Vedaldi, 2016]

Visual Neuroscience: Hubel/Wiesel, …





Olshausen and Field 1996

Experimental Neuroscience uncovered the

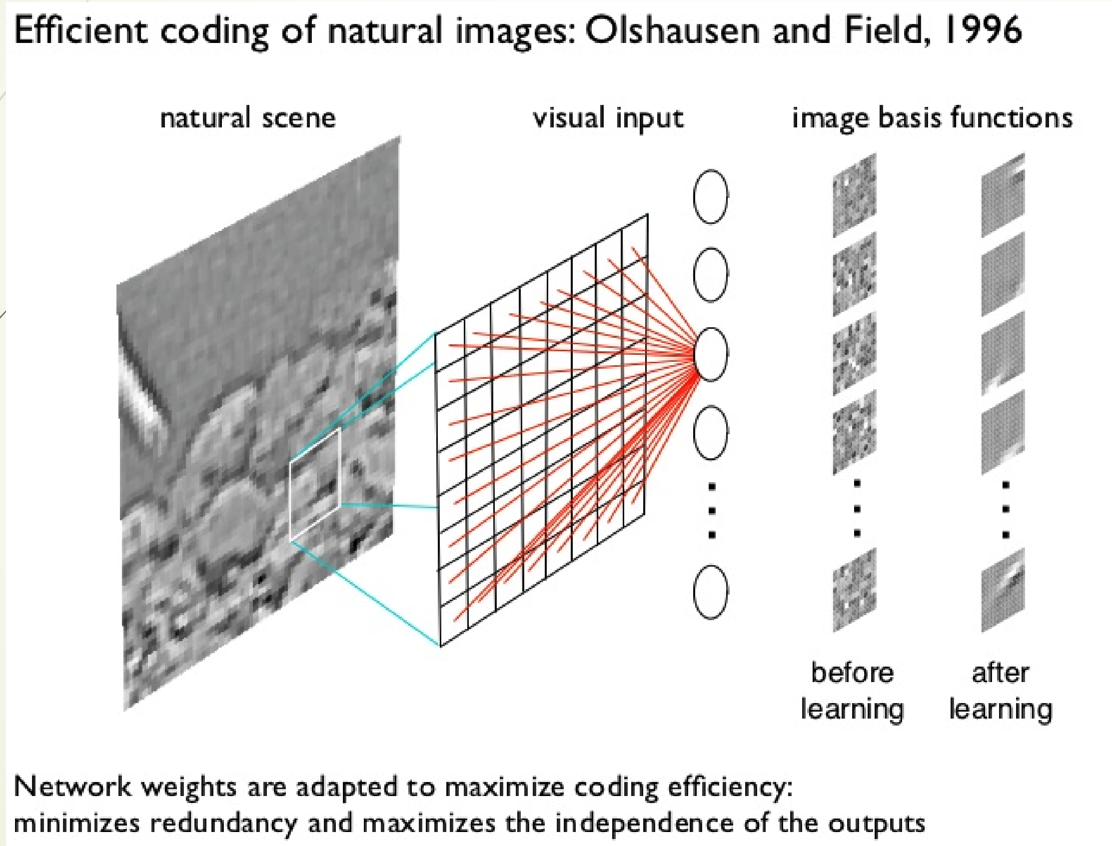
- ... neural architecture of Retina/LGN/V1/V2/V3/ etc
- ... existence of neurons with weights and activation functions (simple cells)
- ... pooling neurons (complex cells)

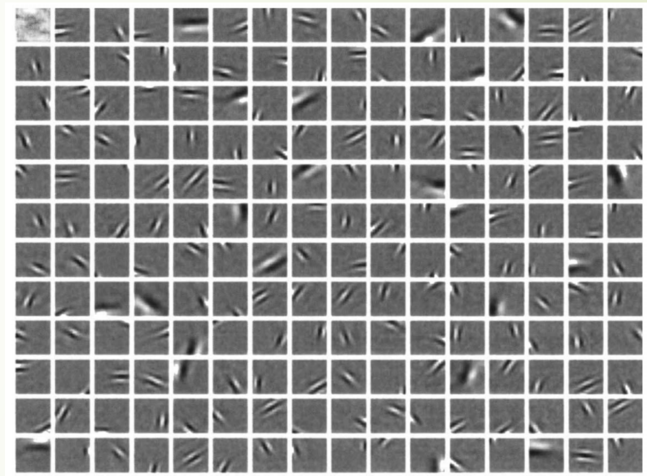
All these features are somehow present in today's sucessful Deep Learning systems

Neuroscience	Deep Network
Simple cells	First layer
Complex celle	Pooling Layer
Grandmother cells	Last layer

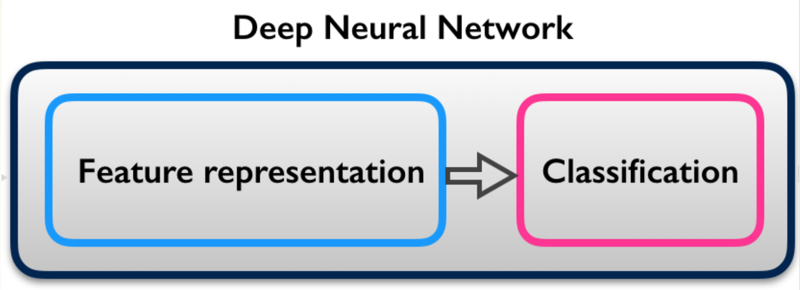
Theorists Olshausen and Field (Nature, 1996) demonstrated that receptive fields learned from image patches

First layers learned …





Transfer Learning?



- Filters learned in first layers of a network are transferable from one task to another
- When solving another problem, no need to retrain the lower layers, just fine tune upper ones
- Is this simply due to the large amount of images in ImageNet?
- Does solving many classification problems simultaneously result in features that are more easily transferable?
- Does this imply filters can be learned in unsupervised manner?
- Can we characterize filters mathematically?

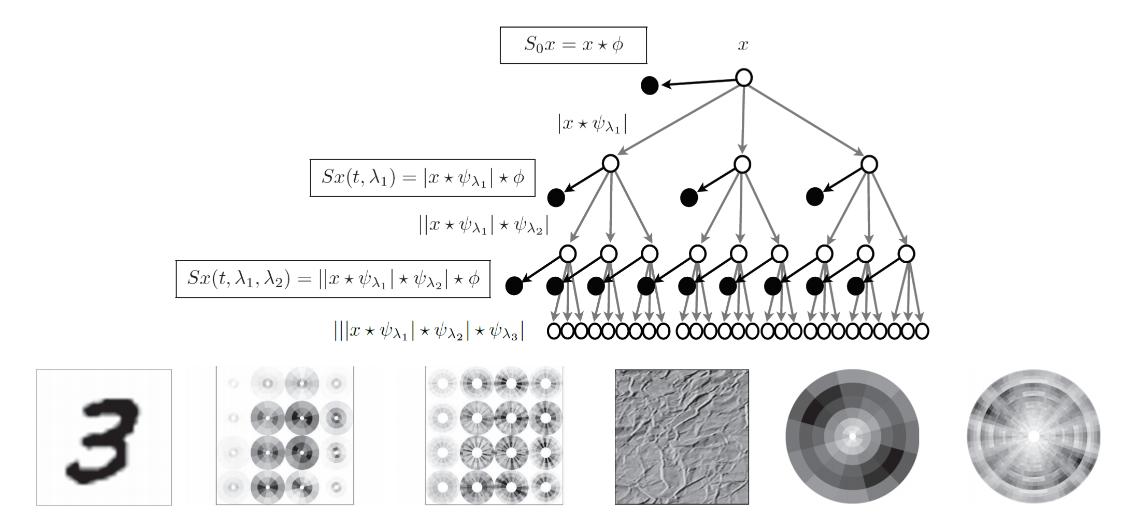
Some Open Theoretical Problems

- *Harmonic Analysis:* What are the optimal (transferrable) representations of functions as input signals (sounds, images, …)?
- *Approximation Theory:* When and why are deep networks better than shallow networks?
- *Optimization:* What is the landscape of the empirical risk and how to minimize it efficiently?
- *Statistics:* How can deep learning generalize well without overfitting the noise?

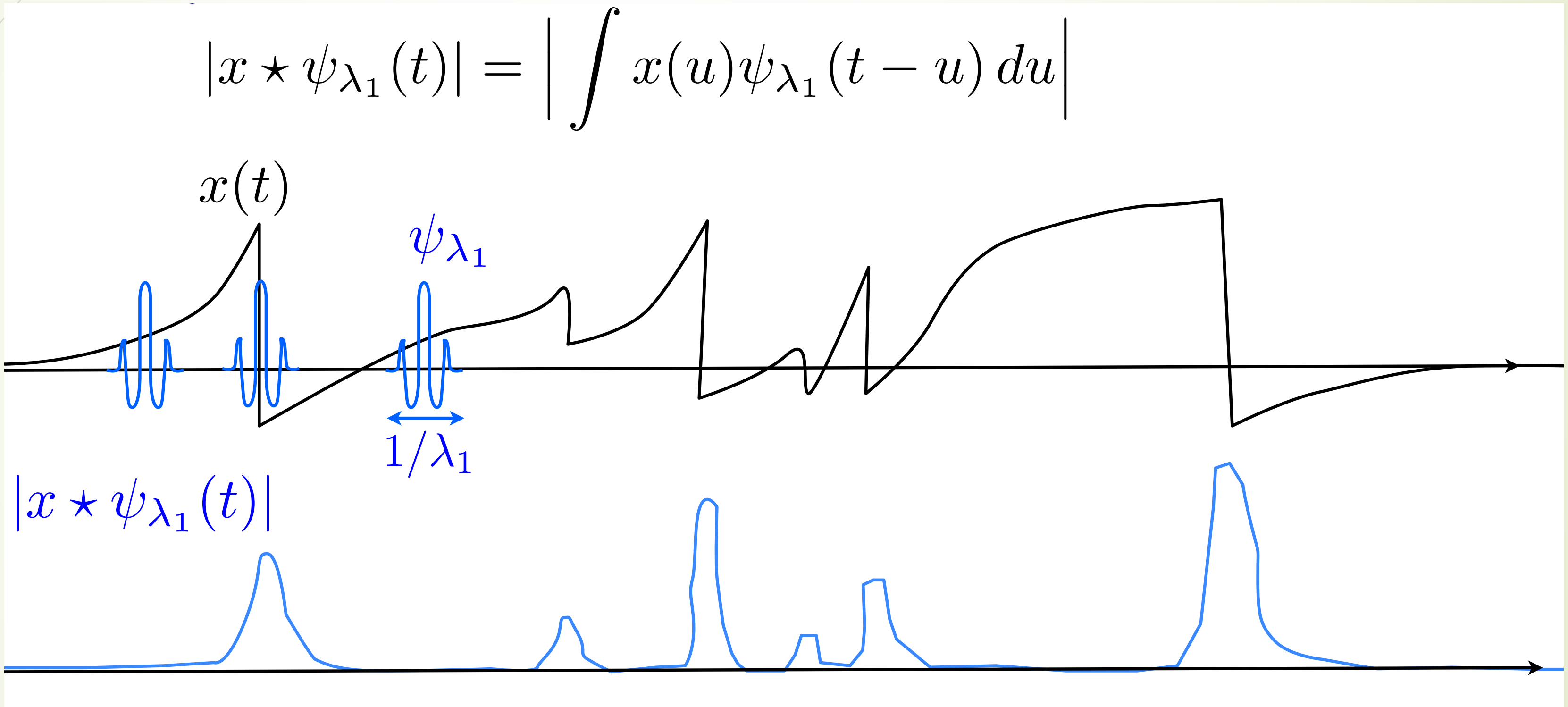
Harmonic Analysis

- Harmonic analysis: optimal representation of input signals
- Wavelets are optimal sparse representations for certain class of images
- **Stephane Mallat:** Deep Scattering Transform – translational, small deformational, rotational and scaling *invariances*; the deeper is the network, the larger are the invariances
- **Mathew Hirn** @IAS-HKUST talked about scattering net for energy functions on 3-D densities (images)

Scattering Transform: Mallat'12

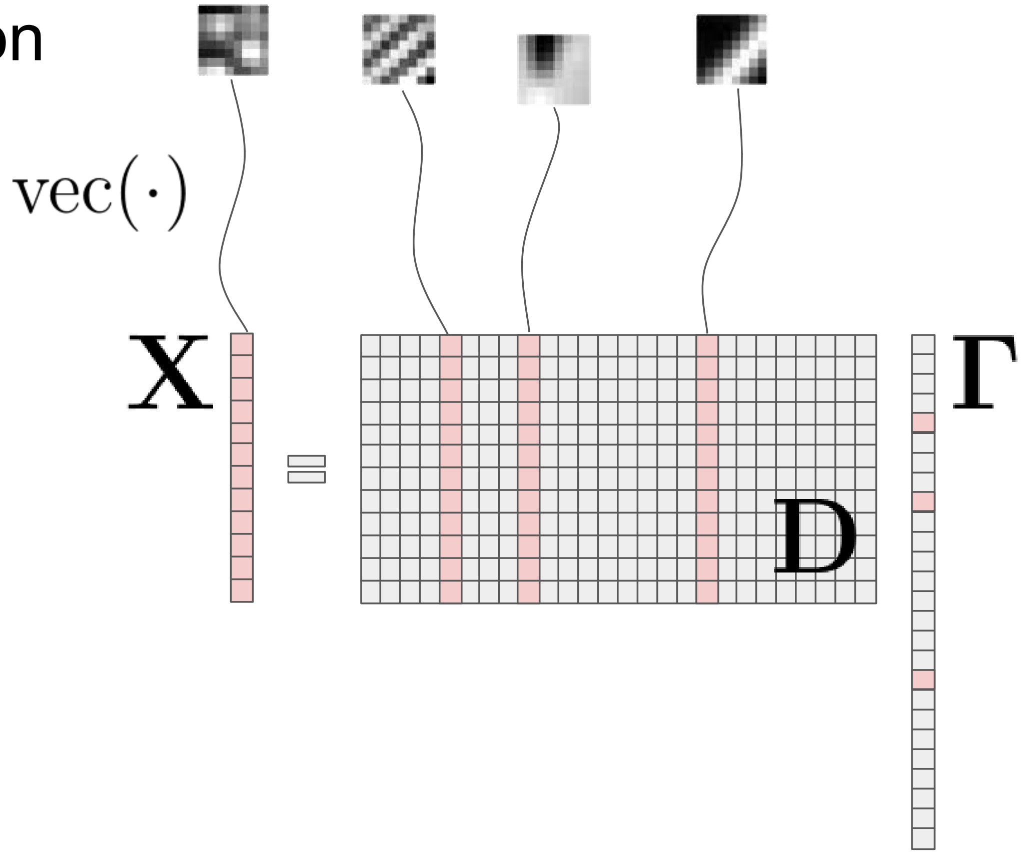


Sparse Representations: Wavelet convolutions



Compressed Sensing

Matrix Notation



Compressed Sensing

Given a signal, we would like to find its sparse representation

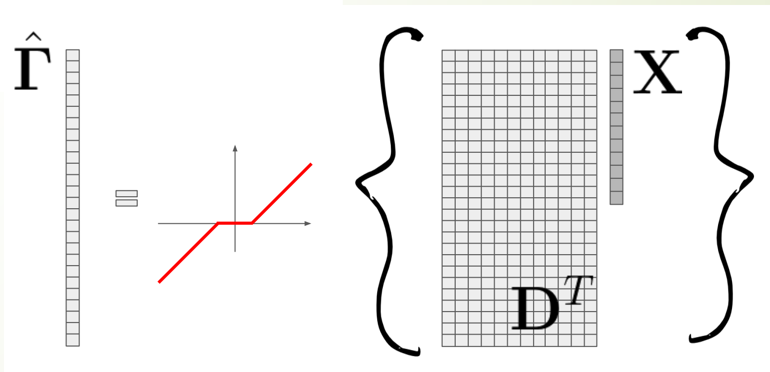
$$\min_{\mathbf{\Gamma}} \ \|\mathbf{\Gamma}\|_0 \quad \text{s.t.} \quad \mathbf{X} = \mathbf{D}\mathbf{\Gamma}$$

Convexify

$$\min_{\mathbf{\Gamma}} \ \|\mathbf{\Gamma}\|_1 \quad \text{s.t.} \quad \mathbf{X} = \mathbf{D}\mathbf{\Gamma}$$

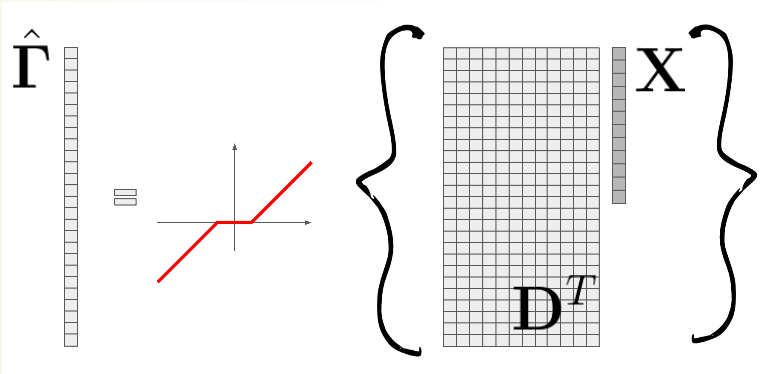
Crude approximation

$$\mathcal{S}_\beta\{\mathbf{D}^T\mathbf{X}\}$$

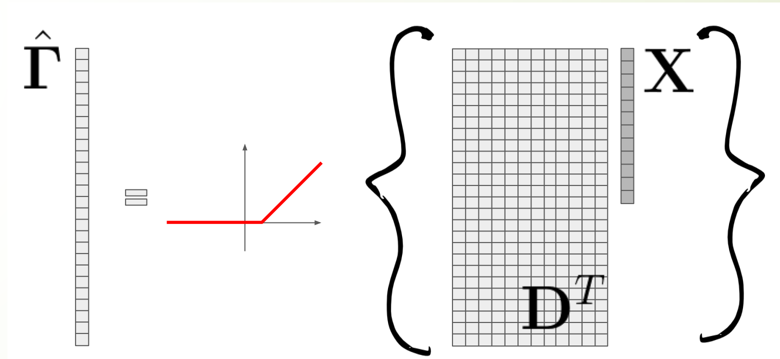


From Soft Thresholding to ReLU

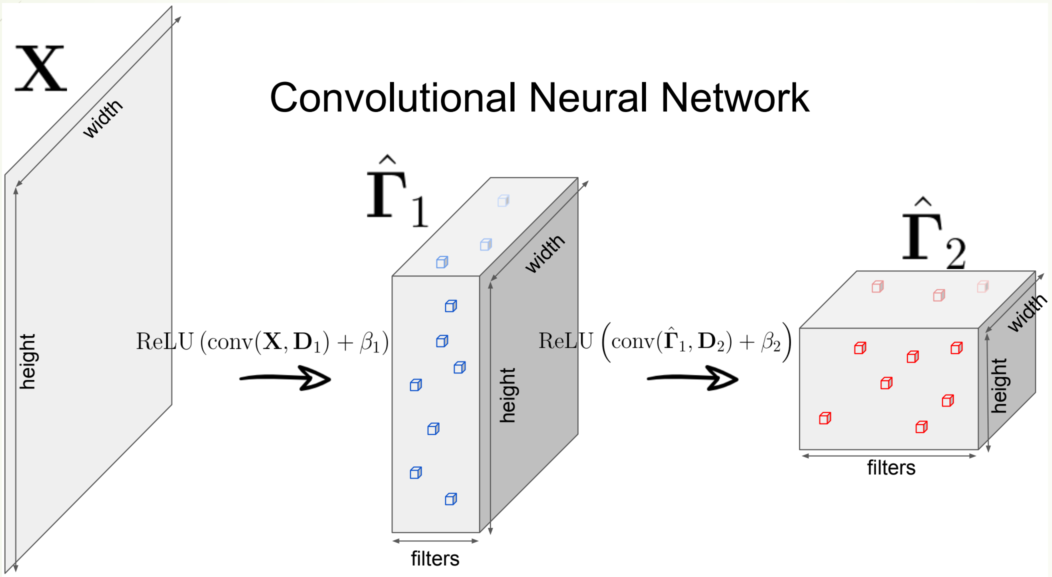
Soft Thresholding



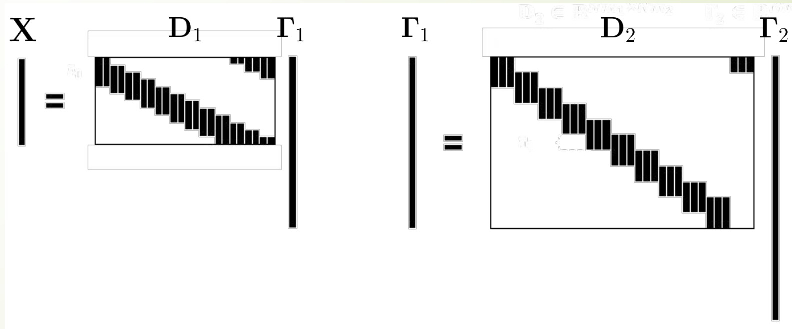
ReLU: Soft Nonnegative Thresholding



Convolutional Neural Network



Can we simultaneously learn dictionaries $\mathbf{D}$s and Γ s?



Incoherence…
Papyan, Sulam, and Elad 2016

Approximation Theory

- Class prediction rule can be viewed as function $f(x)$ of high-dimensional argument
- *Curse of Dimensionality*
  - Traditional theoretical obstacle to high-dimensional approximation
  - "*Functions of high dimensional x can wiggle in too many dimensions to be learned from finite datasets*"

Approximation Theory

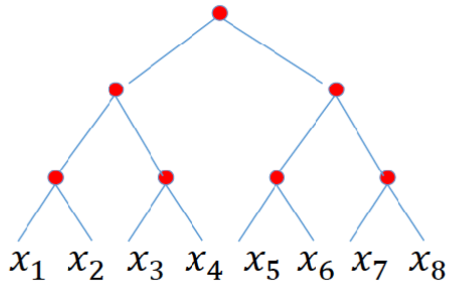
- Ridge Functions $\rho(u'x)$ mathematically same as deep learning first layer outputs.
- Sums of Ridge Functions mathematically same as input to second layer.
- Approximation by Sums of Ridge Functions $f \approx \sum_i \rho_i(u_i'x)$ studied for decades
- Theorists (1990's-Today): certain functions $f(x)$ approximated by ridge sums with no curse of dimensionalty

(Sparse) Compositional Functions

- Compositional functions $f(x) = h(g_1(x_{i_{1,1}}, ..., x_{i_{1,k}}), g_2(x_{i_{2,1}}, \ldots, x_{i_{2,k}}), \ldots, g_\ell(x_{i_{\ell,1}}, \ldots, x_{i_{\ell,k}}))$ are functions of small number of functions; $\ell, k \ll d$.
- VGG Nets are deep compositions
- Approximation by Compositional Functions studied for decades
- Theorists (1990's-Today): certain functions $f(x)$ avoid curse of dimensionalty using multilayer compositions
- T. Poggio (MIT) and Hrushikesh Mhaskar (Caltech) have several papers analyzing deepnets as deep compositions.

Mhaskar-Poggio-Liao'16

$$f(x_1, x_2, \ldots, x_8) = g_3(g_{21}(g_{11}(x_1, x_2), g_{12}(x_3, x_4))g_{22}(g_{11}(x_5, x_6), g_{12}(x_7, x_8)))$$



Theorem (informal statement)

Suppose that a function of d variables is hierarchically, locally, compositional . Both shallow and deep network can approximate f equally well. The number of parameters of the shallow network depends exponentially on d as $O(\varepsilon^{-d})$ with the dimension whereas for the deep network dance is $O(d\varepsilon^{-2})$

IAS-HKUST workshop talks

- *9 Jan 2018, Tuesday:*
  - **Ding-Xuan ZHOU** *Approximation Analysis of Distributed Learning and Deep CNNs*
- *10 Jan 2018, Wednesday:*
  - **Philipp Grohs** *Approximation Results for Deep Neural Networks*
- *11 Jan 2018, Thursday:*
  - **Gitta Kutyniok** *Optimal Approximation with Sparsely Connected Deep Neural Networks*
  - **Philipp Petersen** *Optimal Approximation of Classifier Functions by Deep ReLU Networks*