Artificial Intelligence: Perspectives from Statistics and Computation

Donghui Yan

dyan@umassd.edu

Outline

- Introduction
- Machine learning
- Deep neural network as an example

What is artificial intelligence (AI)?

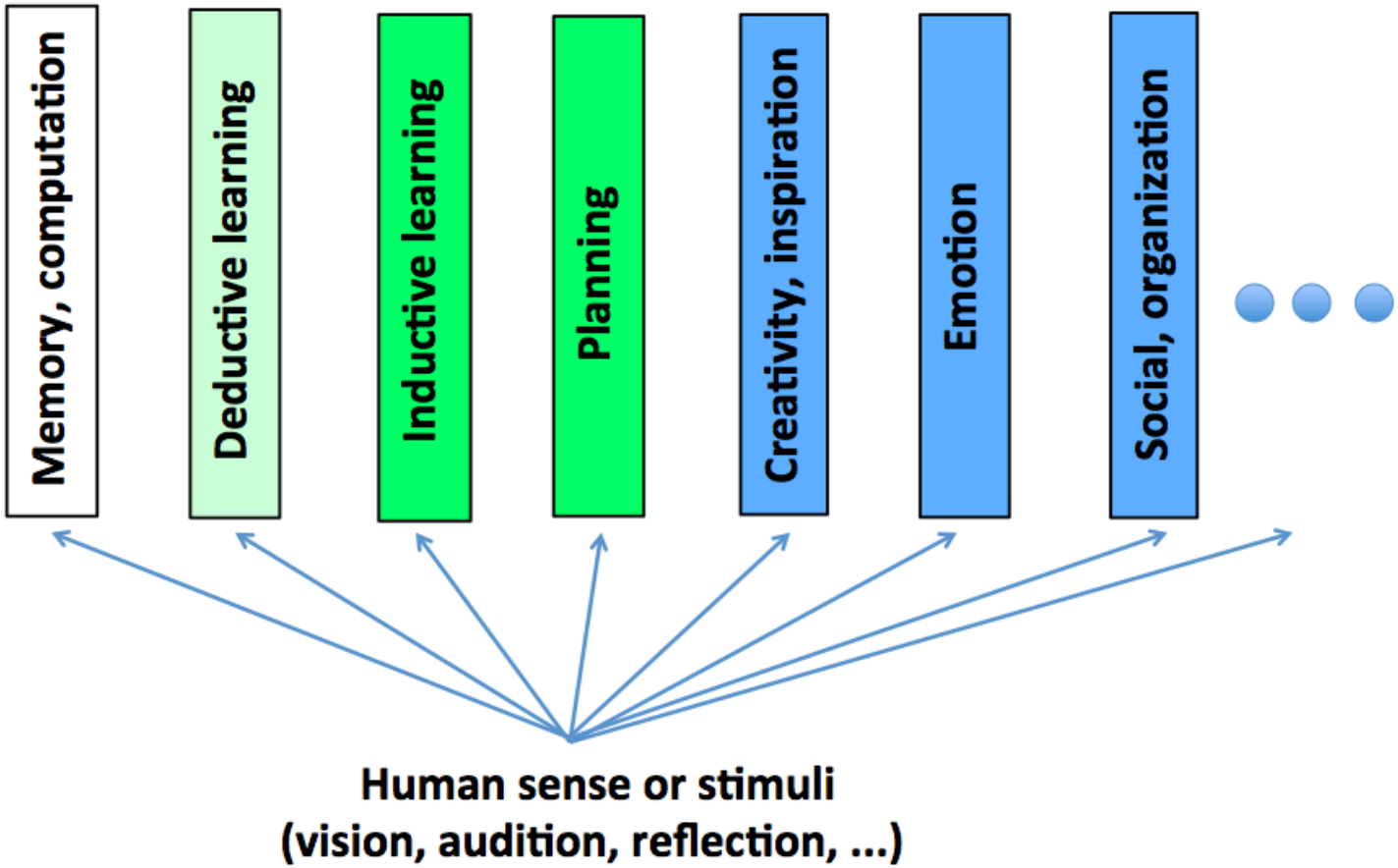
"Everyone talks about AI, everyone thinks everyone else is doing it, but nobody really knows what it is."

- A vague but widely accepted definition

  `Machines or computer programs which can exhibit "intelligent" behavior as we observe from human`

  - In spirit of Turing's test
    - A. M. Turing, "Computing machinery and intelligence", *Mind*, 59(236):433-460, 1950
  - Not necessarily achieving it in the same way as human.

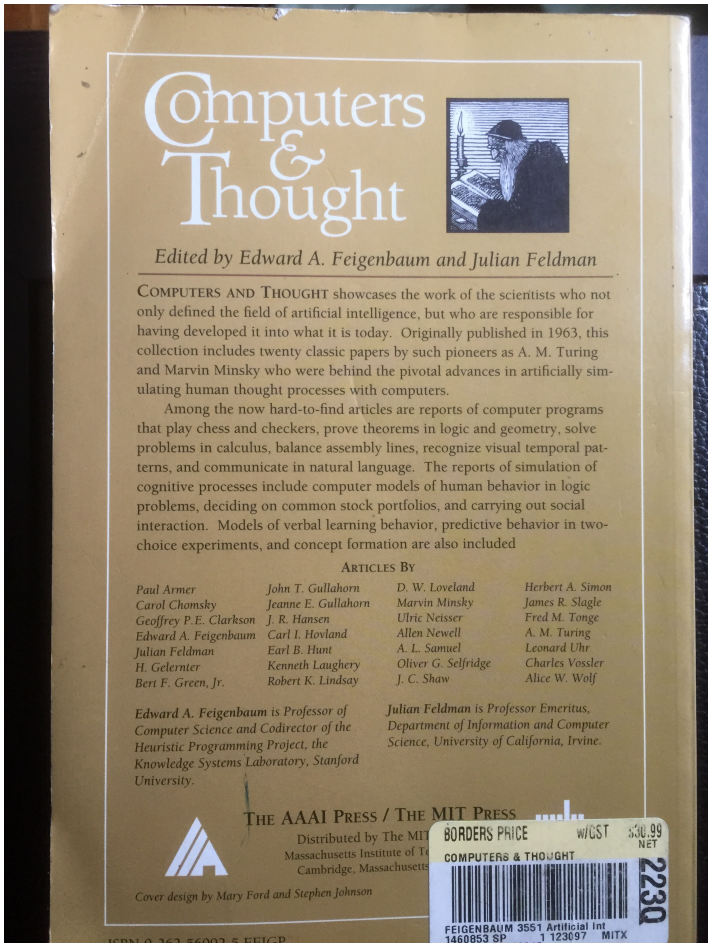
Human intelligence



Towards intelligent behavior

- Characteristics of computers
  - Good memory
    - Potentially much larger than human
  - Fast computation (numerically or symbolically)
    - Fast and more complex computation than human
- Would these translate to intelligence?
  - When problem well-defined and solution space within computational power
    - E.g., exhaustive search in some games
  - When large amount of data available
    - Where there are data, there is statistics
- AI is statistics and computation in nature
  - ♠ Question is how far this goes.

Progress of AI



- Book: *Computers and thought*
  - Collection of 20 classic papers by giants in AI history
    - Survey of AI and attempts
  - Authors include: 1 Nobel + 4 Turing Award laureates + Turing
  - Ironically Simon initially objected publication of the book
- Book originally published in 1963
  - But still quite relevant
    - Interesting to see what has happened since then
  - *My* 1^{st} *encounter of AI* (1999).

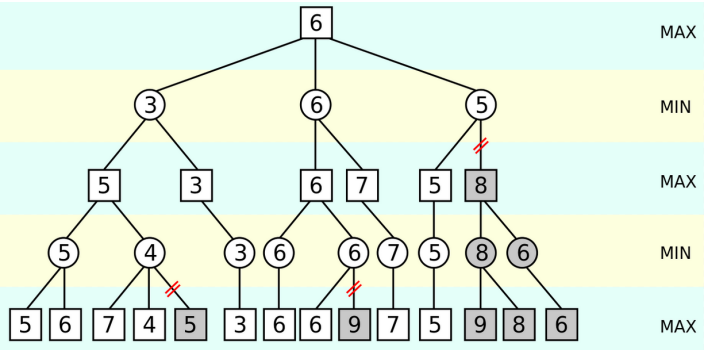
Machines are strong on deductive learning

- Machines that play games
- Machines that prove mathematical theorems.

Machines that play games

- Early progress
  - Chess-playing game by *Newell, Shaw, and Simon* (1958)
  - Game of checkers by *A. L. Samuel* (1959)
- A major milestone: IBM Deepblue
  - Defeated *Kasparov* by $3.5 : 2.5$ (May 1997)
- Google's AlphaGo
  - Beat *Ke Jie*, then world No. 1 ranked Go player (May 2017)
  - AlphaGoZero beat AlphaGo 100-0 (Oct 2017)
    - Self-taught and start from zero.

Ideas of game machines



- Value all possible positions by minimax (*Shannon* 1949)
  - Basis of all game programs
- What if search space is too big?
  - α-β pruning (*McCarthy* 1955)

- Further improvement
  - Various heuristics (learn from human)
    - Statistics comes into play
  - AlphaGo/AlphaGoZero
    - Each move assigned a score based on past games
    - Two programs play against each other and for many games
    - *Surpass thousand years human expertise by several months machine time !*

Automated theorem proving

- Early progress
  - Logic theory machine by *Newell, Shaw and Simon* (1956)
  - Geometry theorem proving machine by *H. Gelernter* (1956)
- Wu's method (late 70's)
  - Solving multivariate polynomial equations
  - Can be used to prove elementary geometry theorems
- Proof verification and first-order theorem proving
  - Each proof step verified by primitive recursive function
  - Many applications
    - To verify proof of 4-color theorem by Appel and Haken (1976)
    - Integrated circuit design verification
    - Correctness of large software system.

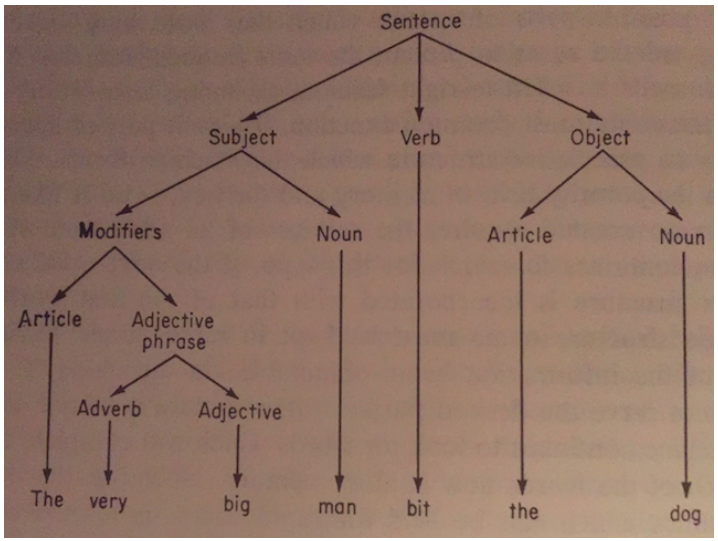
Automated theorem proving (continued)

- Idea of the logic theory machine (*Newell, Shaw and Simon*)
  - Deduction system for propositional logic
  - Start from a small set of axioms, and use 3 deduction rules
    - Modus ponens ($P \Rightarrow Q$, if P is true then Q is true)
    - Variable substitution
    - Replacement of formulas by their definitions
  - Co-NP complete problem
- Wu's method (late 70's)
  - Ritt-Wu characteristic set
  - Repeated division of one polynomial by another
    - Statement true if vanishing remainder and otherwise false
  - Exponential computational complexity though.

Question answering or conversation machines

- Early attempts
  - BASEBALL: automatic Q answer (*Green et al* 1961)
  - Machines that understand natural language (*R. Lindsay* 1962)
  - ELIZA (*J. Weizenbaum* 1966)
    - Rule-based
- More sophisticated systems
  - PARRY (*K. Colby* 1972)
    - The first chatbot to pass the *Turing test*
    - Rule-based + mental model
  - Various phone assistant systems (finite-state based)
- Recent advances
  - Siri (*Apple* 2012)
  - Alexa (*Amazon* 2014)
  - Xiaoice (*Microsoft* 2014)

Ideas of early attempts



- Parse the sentence into a tree
  - By language rules
- Store details related to many contexts
  - Requires huge working memory and computation
- Understand sentence by the parsed tree and contexts.

How does Siri work?

- Active oncology: relational network of concepts
- Learn condition-action rules
  - *Statistical* classifiers to learn contexts or intent.

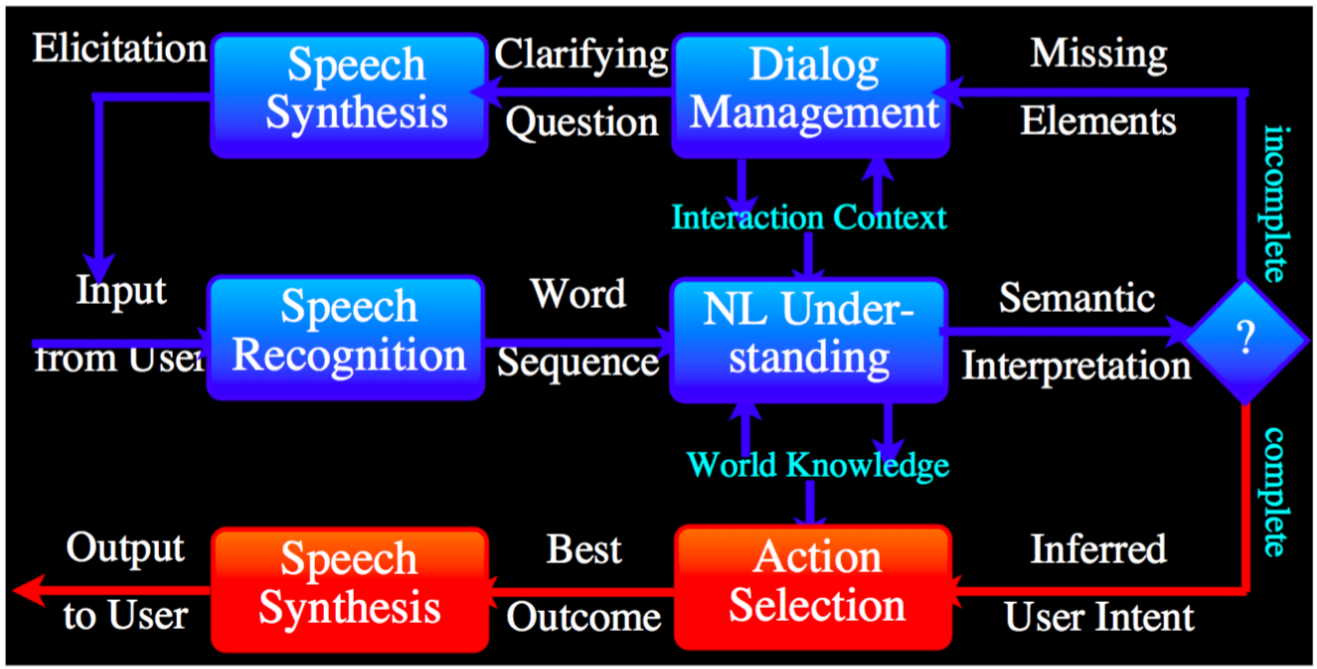
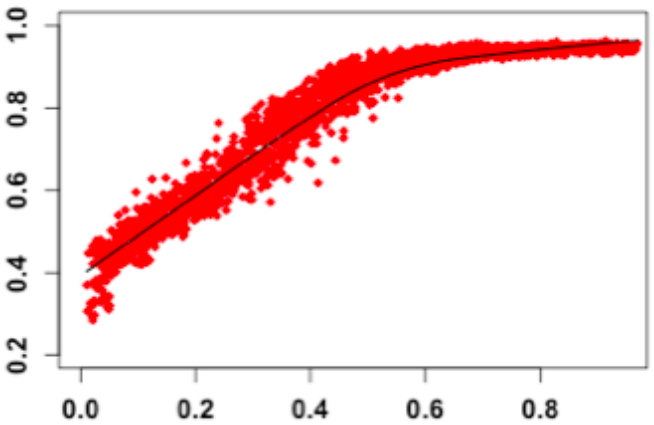


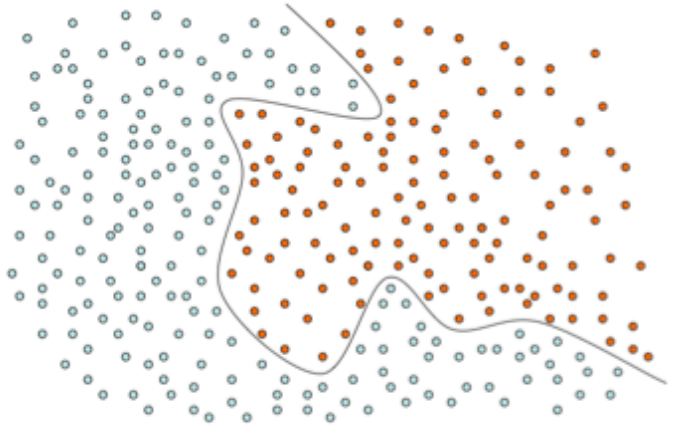
Figure from Jerome Bellegarda

Inductive learning

- Machines are weak on inductive learning
  - While strong on deductive learning
- Arguably more "advanced" form of intelligence
  - Hypothesis formulation by machines
  - Models for machine to make valid and useful prediction
  - General pattern recognition
- Major progress in the subarea of machine learning (ML)
  - Many problems related to pattern recognition or beyond can be solved in the framework of ML
  - Reinforcement learning also picks up.

Machine learning





- Automatics learning of regularities or patterns from data
- Given data $(X_1, Y_1), ..., (X_n, Y_n)$ from some unknown distribution $\mathbb{P}$, solve $\arg\min_{f:X\mapsto Y\in\mathcal{F}} \mathbb{E}_{\mathbb{P}} l(f(X), Y)$
  - Function class $\mathcal{F}$: linear models, or SVM, or tree methods, or deep neural network etc
  - Generalization instead of memory
  - Statistics in nature.

A little history

- Early days
  - The AI community
    - 1956 Dartmouth conference marks the start of AI
    - Perceptron (*Rosenblatt*, 1957)
    - Dying of the research on Neural network late 1960's
    - Various induction machine, expert system, fuzzy system etc
    - PAC learning (*Valiant* 1984)
  - The statistical community
    - Statistical learning theory (*Vapnik and Chervonenkis*, 1964-1974)
    - Fisher's LDA, logistic regression, k-means, mixture analysis etc
    - Early nonparametric statistics (e.g., kNN)
- The revitalization of neural network in late 1980's
- SVM, boosting, Random Forests from mid 1990's
- Emergence of deep neural network from around 2010.

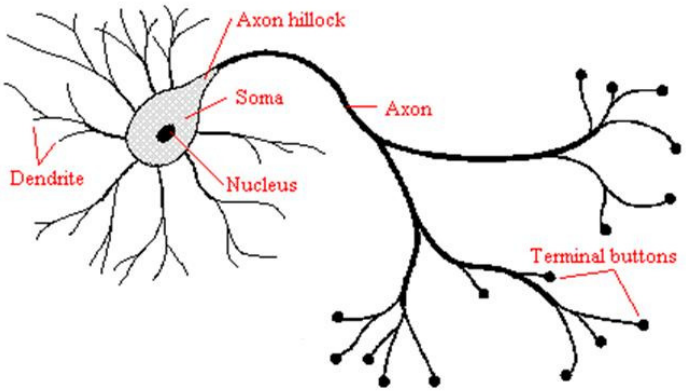
Ingredients of machine learning

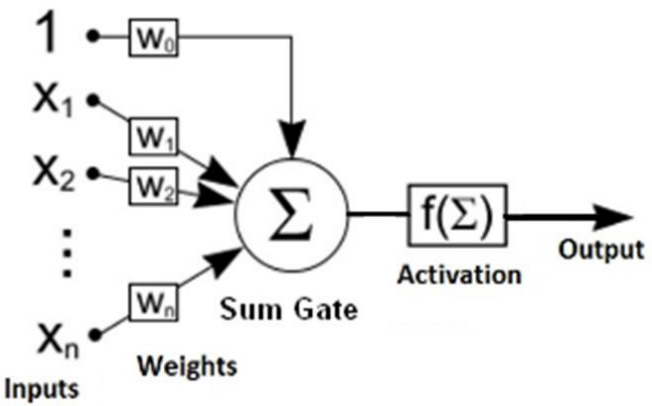
- Empirical (structural) risk minimization as general framework
  - *Vapnik and Chervonekins* theory (1964-1974)
- Statistical efficiency
  - Will algorithm achieve *theoretical optimal* on infinite data?
    - Many popular learning algorithms are consistent, e.g., SVM, Boosting, neural network etc
  - Rate of convergence
    - Related to how much data is needed
    - e.g., how fast error rate decreases with more data
- Computational efficiency
  - How fast does the algorithm converge?
  - Crucial for big data.

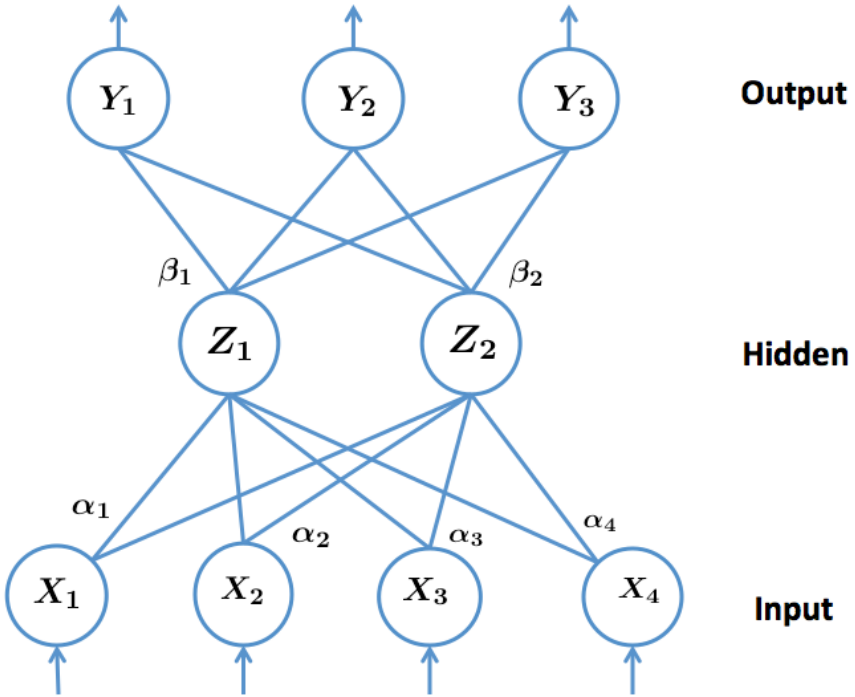
Central role of statistics and computation

- Support vector machines (SVM, *Cortes and Vapnik* 1995)
  - Optimization formulation
  - Representer theorem
  - Numerous papers on improving training speed
- Boosting (*Schapire and Freund* 1996)
  - Hundreds of papers on theoretical understanding
    - Margin view, functional gradient descent etc
  - Choice of basis function class
    - Gradient boost, XGBoost etc
- Random Forests (*Breiman* 1999)
  - Theoretical understanding remains mysterious
    - Consistency by *Breiman, Biau, Scornet, Lugosi* on simplified or randomized version
    - *Wagner* showed consistency on regression (2016).

Network architecture







The neural network architecture

- Input $\to$ Hidden layer

$$X \mapsto \sigma(\alpha_0 + \alpha^T X) \triangleq Z$$

  - Classic choice of activation is $\sigma(x) = 1/(1 + e^{-x})$

- Hidden $\to$ Output

$$\begin{aligned} Z &\mapsto \beta_0 + \beta^T Z \triangleq T, \\ T &\mapsto g(T_1, ..., T_K) \triangleq f(X) \triangleq Y \end{aligned}$$

- # Total parameters

  (# Input +1) $\times$ # Hidden + (# Hidden + 1) $\times$ # Output.

Fitting neural network

- Loss function
  - L_2-loss for regression

$$L(\theta) = \sum_{i=1}^{N} \sum_{k=1}^{K} (y_{ik} - f_k(x_i))^2 = \sum_{i=1}^{N} R_i$$

  - Cross-entropy for classification

$$L(\theta) = -\sum_{i=1}^{N} \sum_{k=1}^{K} y_{ik} \log f_k(x_i)$$

- Minimize $L(\theta)$ by back-propagation.

Back-propogation

- The algorithm to compute gradient (Werbos 1975)
  - *Model fitting* of NN by (stochastic) gradient descent
- Idea: chain rule in calculus.
- Let $L(\theta) = g(h(\theta))$, then

$$\frac{dL}{d\theta} = g'(h(\theta)) \cdot h'(\theta).$$

Back-propagation (continued)

- Taking derivatives to get

$$\frac{\partial R_i}{\partial \beta_{km}} = -2(y_{ik} - f_k(x_i))g_k'(\beta_k^T z_i)z_{mi}$$

$$\frac{\partial R_i}{\partial \alpha_{ml}} = -\sum_{k=1}^{K} 2(y_{ik} - f_k(x_i))g_k'(\beta_k^T z_i)\beta_{km}\sigma'(\alpha_m^T x_i)x_{il}$$

where $i = 1, ..., N$,
l, m, k are indices for the input, hidden and output layers.

Back-propagation (continued)

- Update rule at the $(t+1)^{th}$ iteration (learning rate ϵ_t)

$$\beta_{km}^{(t+1)} = \beta_{km}^{(t)} - \epsilon_t \sum_{i=1}^{N} \frac{\partial R_i}{\partial \beta_{km}^{(t)}},$$

$$\alpha_{ml}^{(t+1)} = \alpha_{ml}^{(t)} - \epsilon_t \sum_{i=1}^{N} \frac{\partial R_i}{\partial \alpha_{ml}^{(t)}}$$

- Each iteration in model fitting requires
  - $O(IMKN)$ computation
    - $O(MKN)$ for β_{km}'s
    - $O(IMKN)$ for α_{ml}'s
  - Very expensive for N large and deep layers.

Statistics aspects of neural network

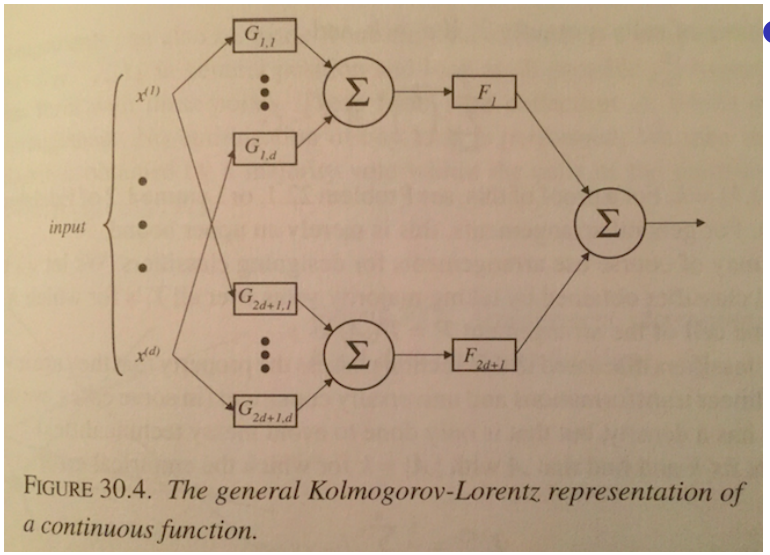
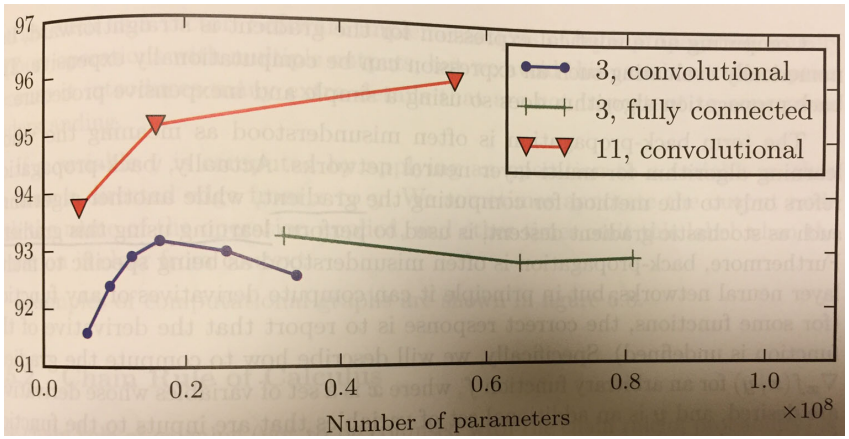


FIGURE 30.4. *The general Kolmogorov-Lorentz representation of a continuous function.*

- The *Kolmogorov-Lorentz* network (1957, 1976)
  - All continuous function can be represented by K-L network
  - Richness in expressive power of neural network

- Consistency of neural network (*Farago and Lugosi* 1993)
- However, network not *deep* enough
  - Non-satisfactory empirical results in many cases
  - Computation and data bottleneck.

Depth is important (Courtesy: Goodfellow et al 2015)



	Removing layers	# parameters
18.1%	None	60m
19.2%	7	44m
23.8%	6,7	10m
21.1%	3,4	59m
51.6%	3,4,6,7	—

Table: *ImageNet by convNet (Krizhevsky, Sutskever, Hinton 2012).*

Deep features may be informative

- 1st layer of a deep NN gives edge-like feature
  - 2nd and higher layer texture or shape etc information.

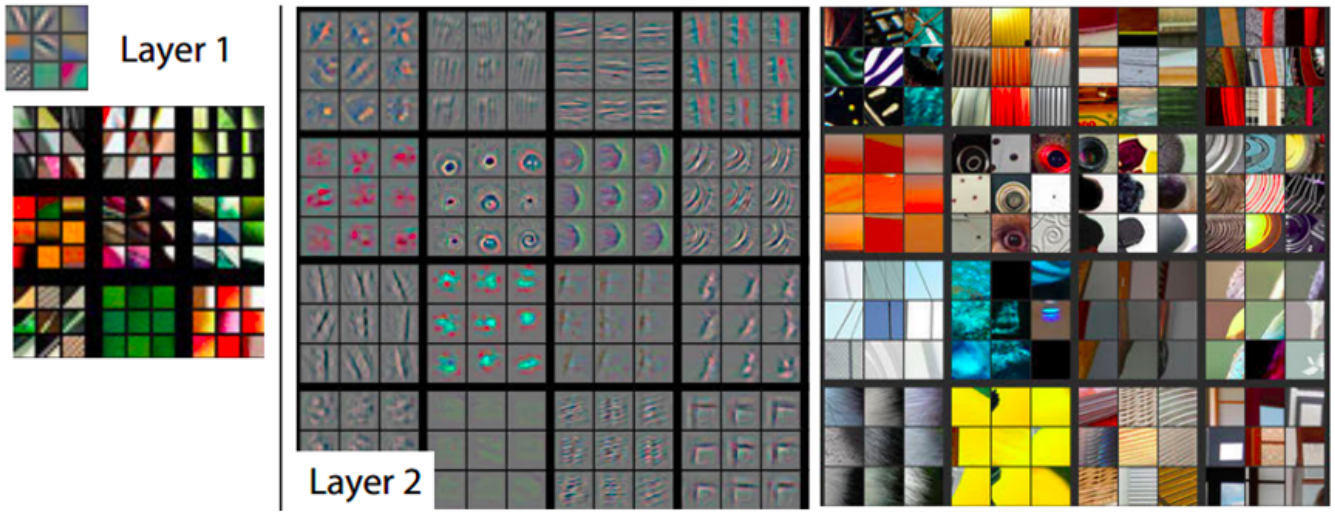


Figure: *Matthew D. Zeiler and Rob Fergus (2014).*

Deep features may be informative (continued)

- 3rd layer of a deep NN gives object parts

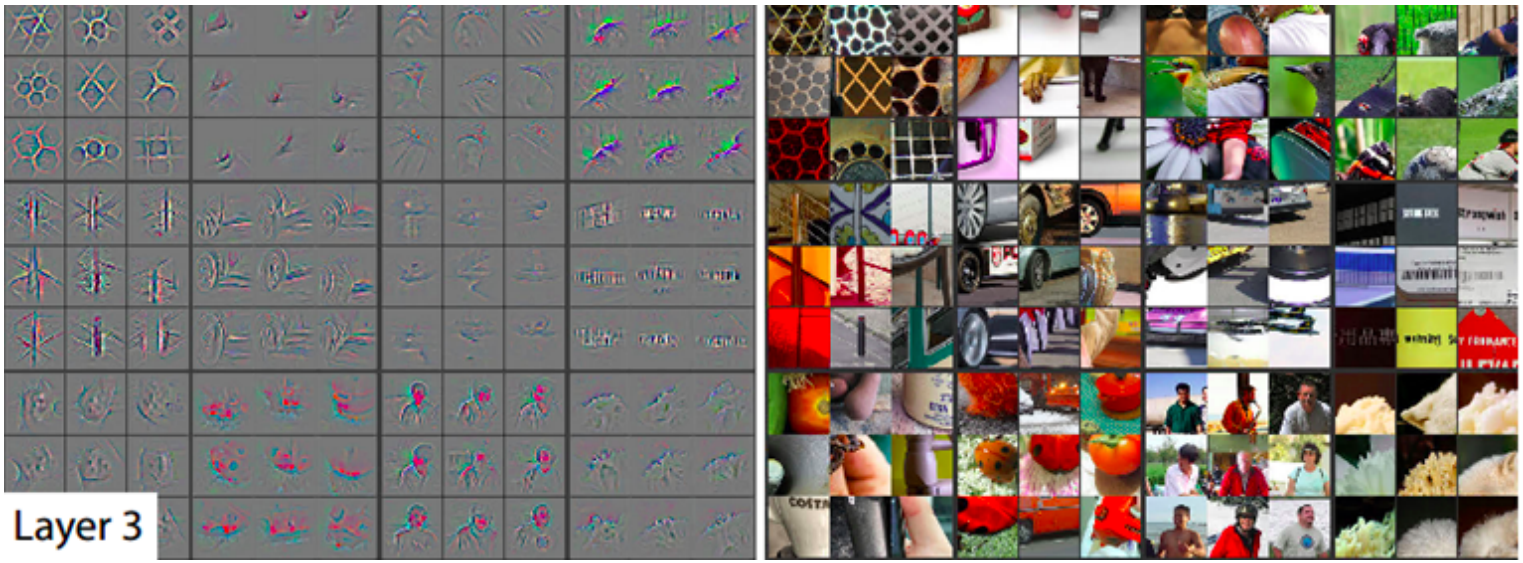


Figure: *Matthew D. Zeiler and Rob Fergus (2014).*

Deep features may be informative (continued)

- 4th and above layers gives simple objects

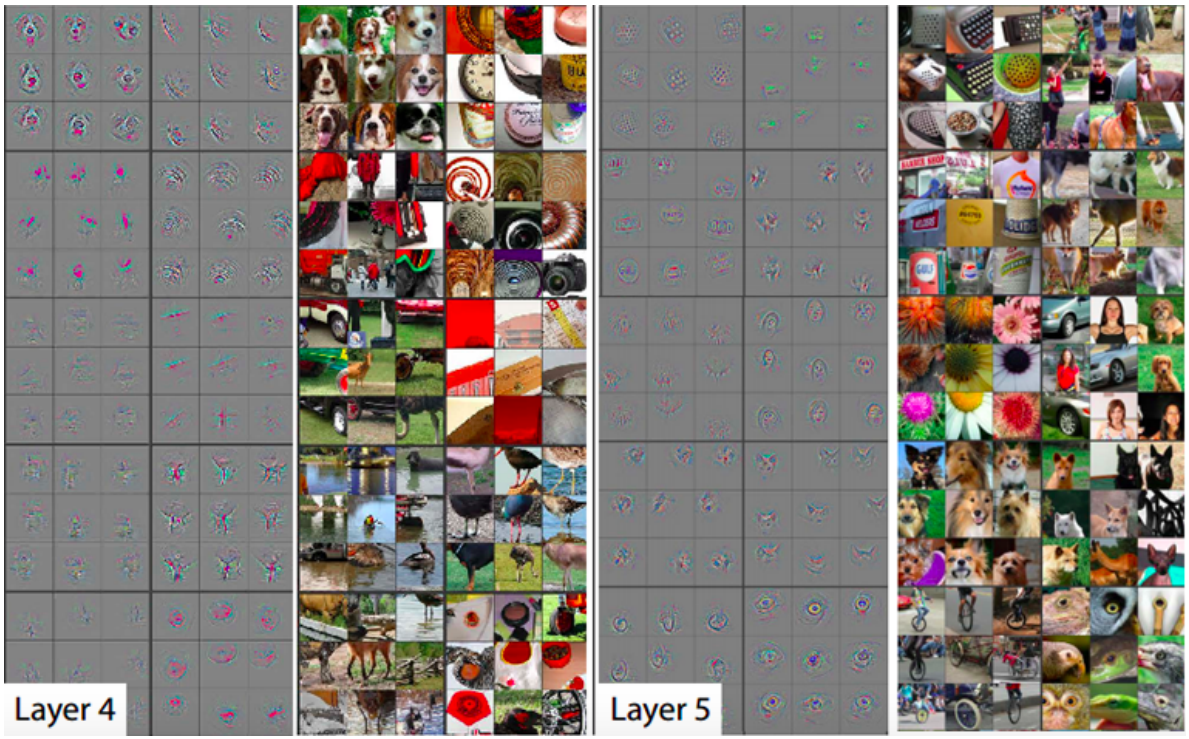


Figure: *Matthew D. Zeiler and Rob Fergus (2014).*

Understanding deep network

- Exponential advantage of a deep network
  - *Montufar et al* show (2004) show # linear regions separated by deep NN with d inputs, depth l, and n units per hidden layer is

$$O\left(\binom{n}{d}^{d(l-1)} n^d\right).$$

- *Johannes Schmidt-Hieber* (2018) show inter-layer sparsity and ReLU activation give statistical guarantee.

Challenges in training deep neural network

- Typically large number of parameters
  - E.g., winners of ImageNet typically have over millions
  - Accordingly, need huge data to avoid overfitting
  - Requires matching computation power
- Computation of back propagation is expensive
  - Gradient calculation at each layer and every node
- Advances in hardware and algorithms
  - Multicore or clustered computers, GPUs
  - Stochastic gradient descent instead of on full data
  - Replace sigmoidal activation with ReLU.

Nonlinear activation

- Options
  - $Tanh(x)$
  - Sigmoid function $1/(1+exp(-x))$
  - Rectified linear unit (ReLU)
    - Easy to optimize with gradient-based methods
    - Good generalization ability as linear models
    - "single most important factor in progress" (Jarrett 2009).

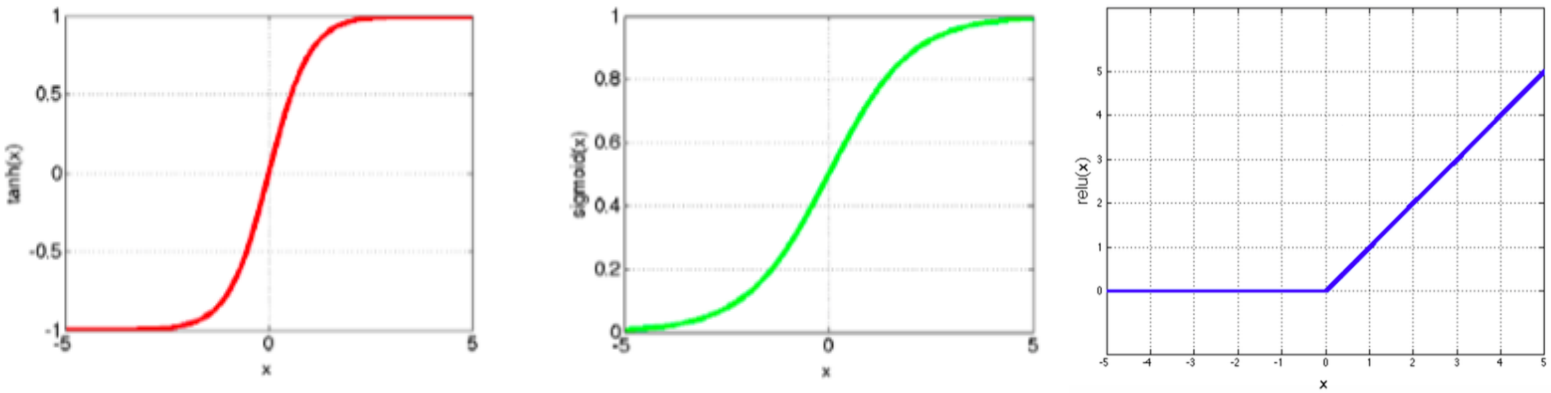


Figure: *Image courtesy LeCun 2014.*

The end

Thank you!

http://www.math.umassd.edu/~dyan