CS60010: Deep Learning

Sudeshna Sarkar

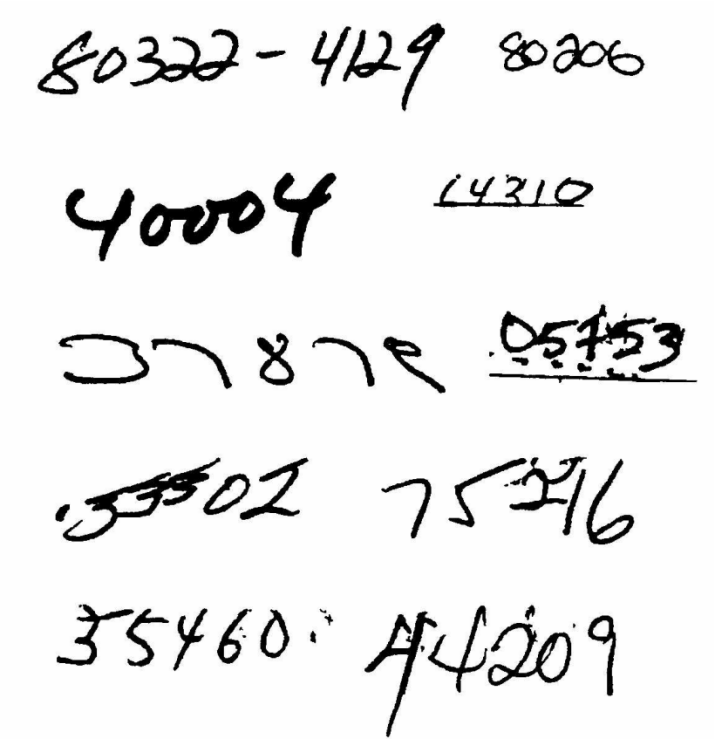
Spring 2018

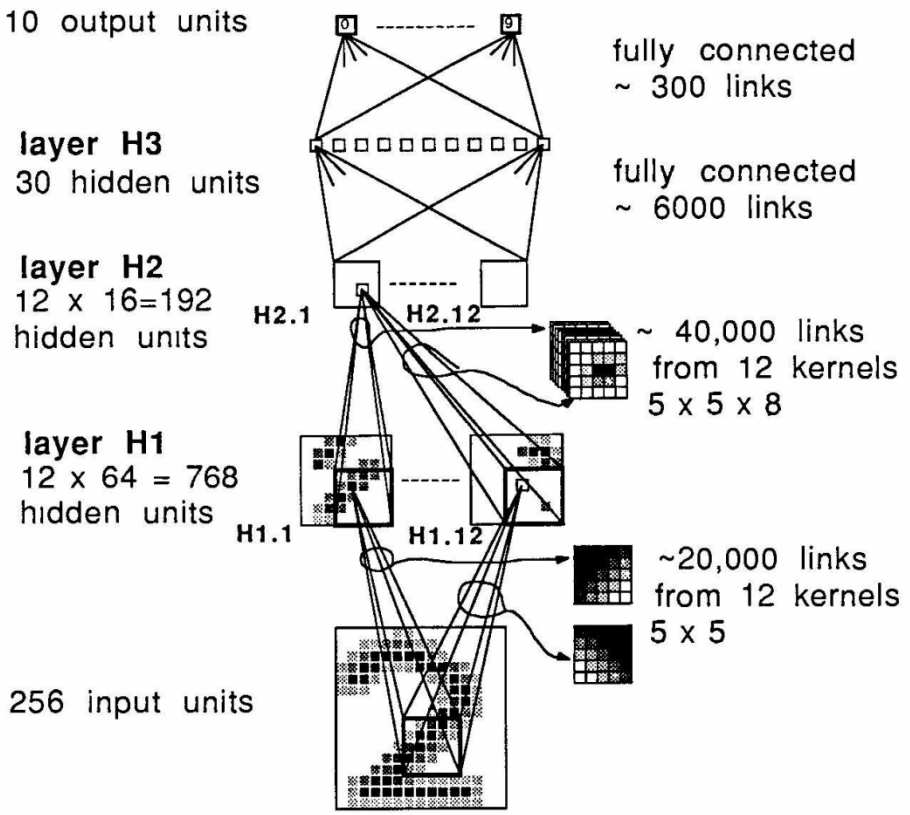
8 Jan 2018

INTRODUCTION

Milestones: Digit Recognition

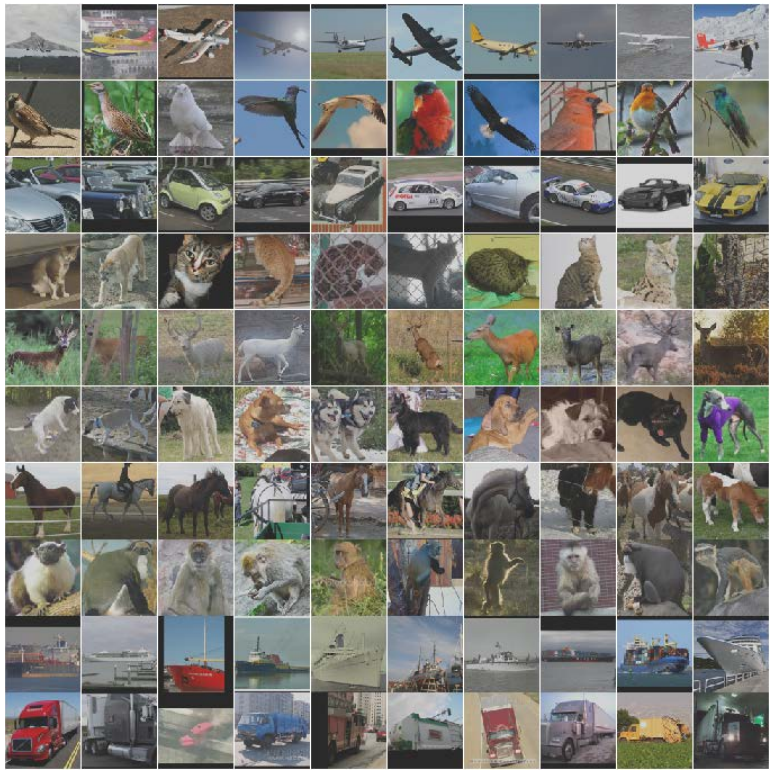
LeNet 1989: recognize zip codes, Yann Lecun, Bernhard Boser and others, ran live in US postal service

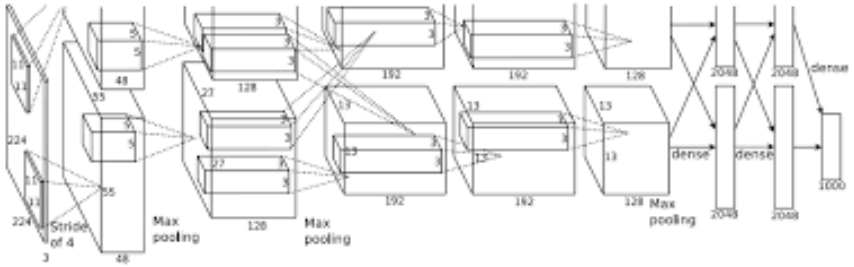


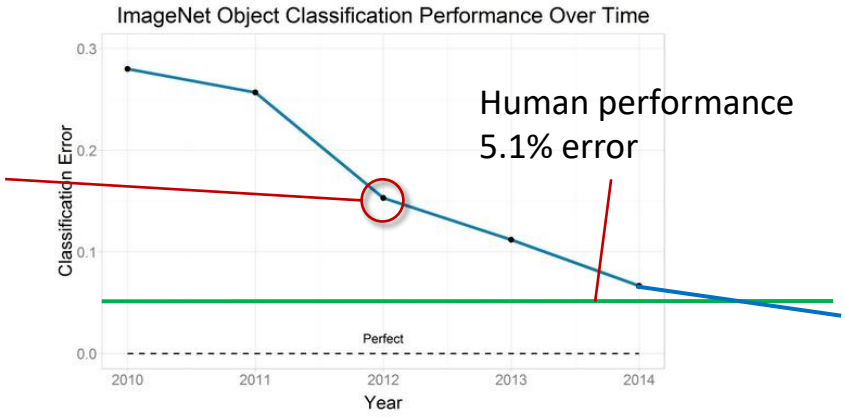


Milestones: Image Classification

Convolutional NNs: AlexNet (2012): trained on 200 GB of ImageNet Data

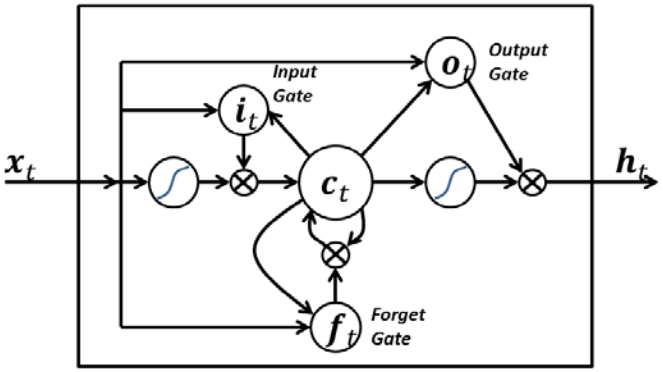


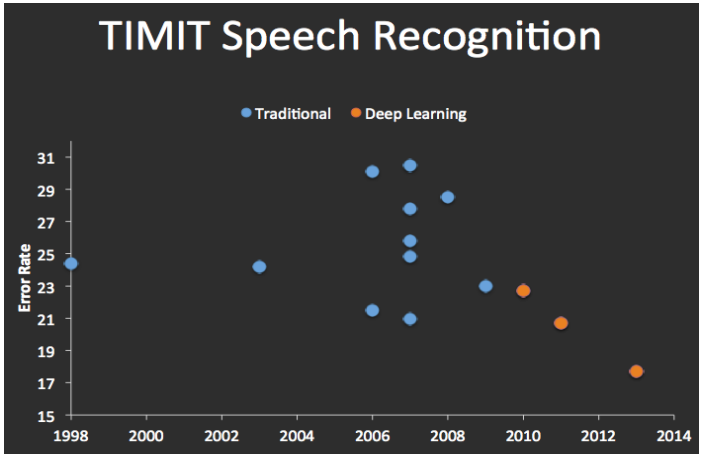


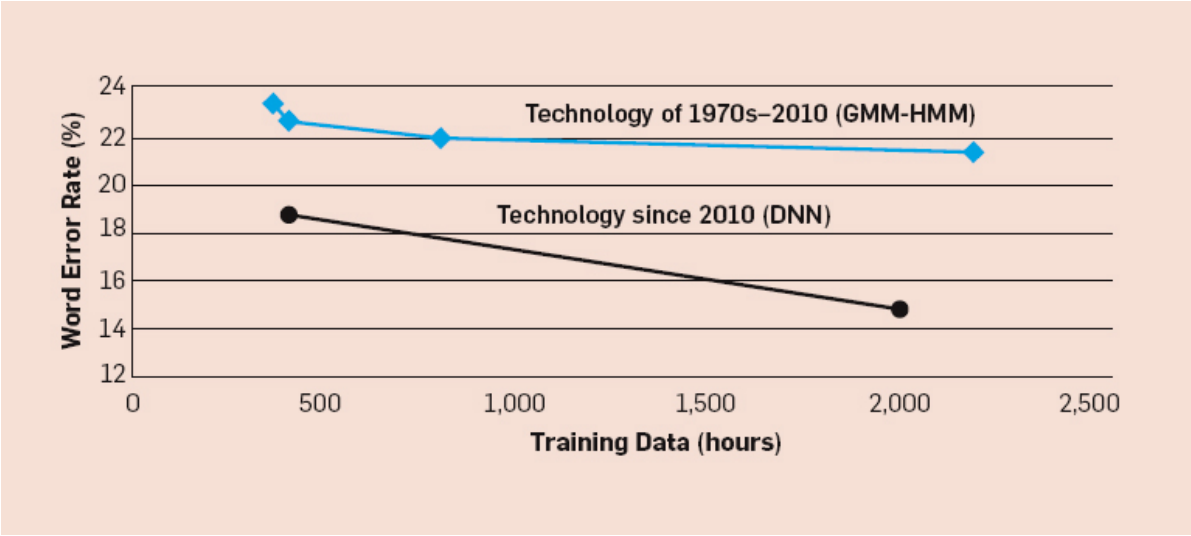


Milestones: Speech Recognition

Recurrent Nets: LSTMs (1997):

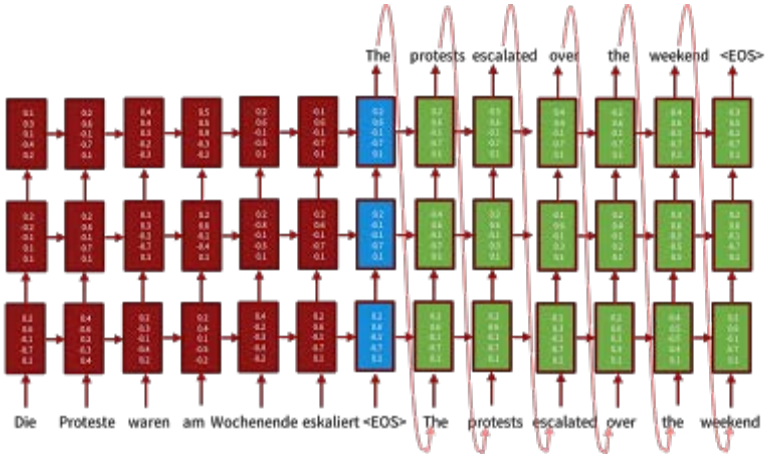


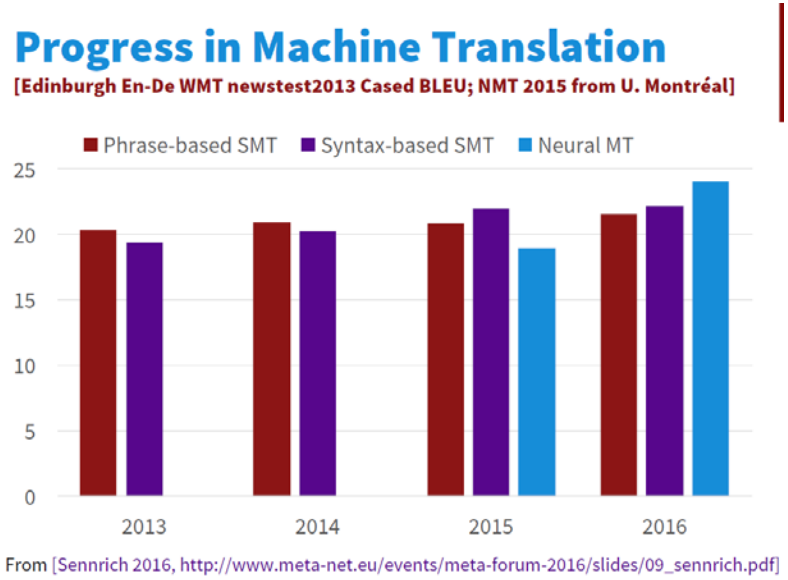




Milestones: Language Translation

Sequence-to-sequence models with LSTMs and attention:



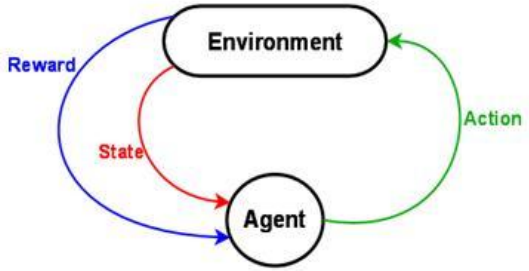


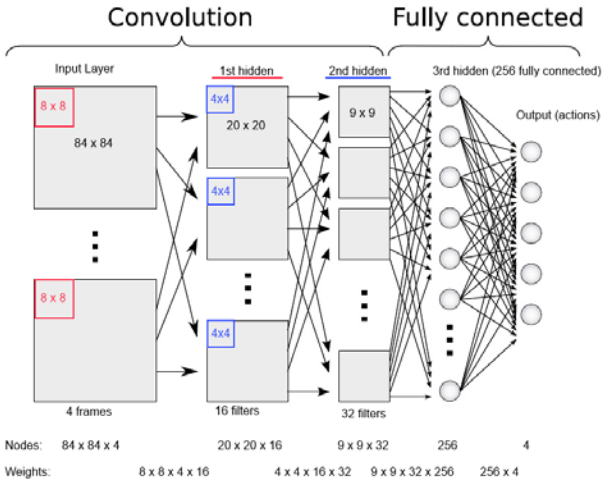
From [Sennrich 2016, http://www.meta-net.eu/events/meta-forum-2016/slides/09_sennrich.pdf]

Source Luong, Cho, Manning ACL Tutorial 2016.

Milestones: Deep Reinforcement Learning

In 2013, Deep Mind's arcade player bests human expert on six Atari Games. Acquired by Google in 2014,.





In 2016, Deep Mind's

alphaGo defeats former world champion Lee Sedol

Learning about Deep Neural Networks

Yann Lecun: DNNs require: “an interplay between intuitive insights, theoretical modeling, practical implementations, empirical studies, and scientific analyses”

i.e. there isn’t a framework or core set of principles to explain everything (c.f. graphical models for machine learning).

This Course

Goals:

- Introduce deep learning.
- Review principles and techniques for understanding deep networks.
- Develop skill at designing networks for applications

This Course

- Times: Mon 12-1, Tue 10-12, Thu 8-9

- **Assignments (pre-midterm): 20%**
- **Post-midterm assignments / Project: 20%**
- **Midterm: 30%**
- **Endterm: 30%**

- TAs: Ayan Das, Alapan Kuila, Aishik Chakraborty, Ravi Bansal, Jeenu Grover

- Moodle: DL Deep Learning
- Course Home Page: cse.iitkgp.ac.in - TBD

Prerequisites

- Knowledge of calculus and linear algebra
- Probability and Statistics
- Machine Learning

- Programming in Python.

Logistics

- 3 hours of lecture
- 1 hour of programming / tutorial

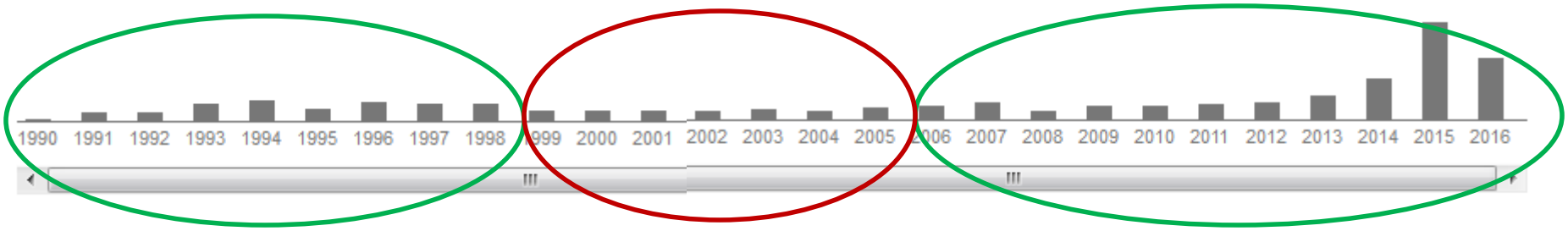
- Attendance is compulsory

Phases of Neural Network Research

- 1940s-1960s: Cybernetics: Brain like electronic systems, morphed into modern control theory and signal processing.
- 1960s-1980s: **D**igital computers, automata theory, computational complexity theory: simple shallow circuits are very limited…
- 1980s-1990s: Connectionism: complex, non-linear networks, back-propagation.
- 1990s-2010s: Computational learning theory, graphical models: Learning is computationally hard, simple shallow circuits are very limited…
- 2006→: Deep learning: End-to-end training, large datasets, explosion in applications.

Citations of the “LeNet” paper

- Recall the LeNet was a modern visual classification network that recognized digits for zip codes. Its citations look like this:



Second phase

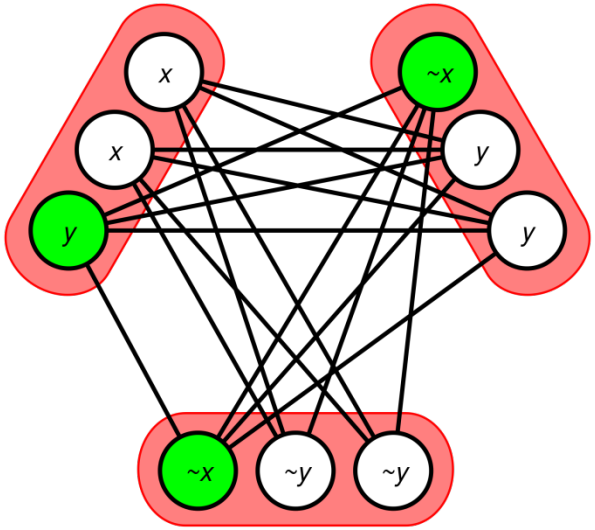
Deep Learning “Winter”

Third phase

- The 2000s were a golden age for machine learning, and marked the ascent of graphical models. But not so for neural networks.

Why the success of DNNs is surprising

- From both complexity and learning theory perspectives, simple networks are very limited.
  - Can't compute parity with a small network.
  - NP-Hard to learn "simple" functions like 3SAT formulae, and i.e. training a DNN is NP-hard.



Why the success of DNNs is surprising

- The most successful DNN training algorithm is a version of gradient descent which will only find local optima. In other words, it's a greedy algorithm. Backprop:

$$\text{loss} = f(g(h(y)))$$

$$d\,\text{loss}/dy = f'(g) \times g'(h) \times h'(y)$$

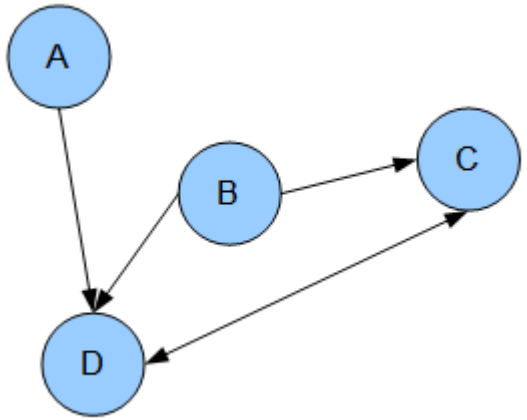
- Greedy algorithms are even more limited in what they can represent and how well they learn.

- If a problem has a greedy solution, its regarded as an "easy" problem.

Why the success of DNNs is surprising

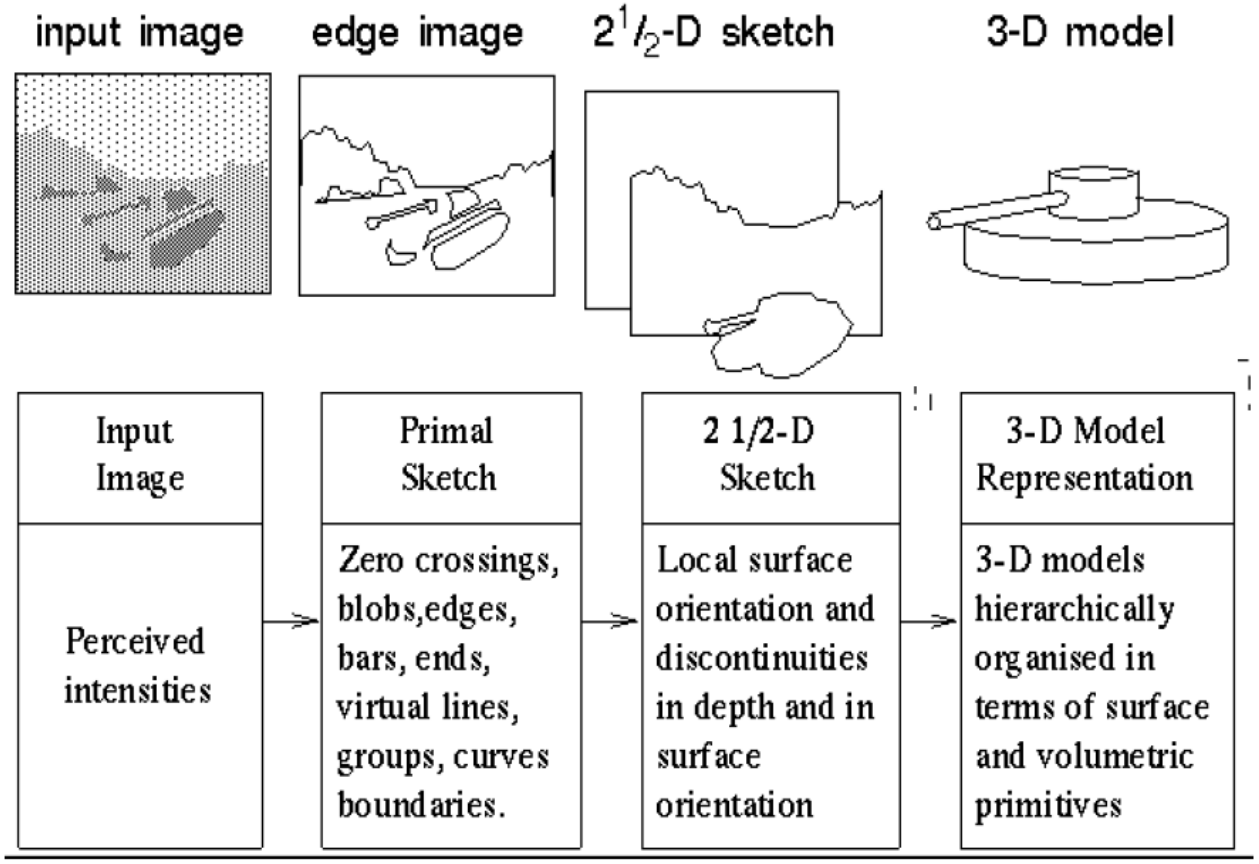
- In graphical models, values in a network represent random variables, and have a clear meaning. The network structure encodes dependency information, i.e. you can represent rich models.

- In a DNN, node activations encode nothing in particular, and the network structure only encodes (trivially) how they derive from each other.



Why the success of DNNs is surprising ~~obvious~~

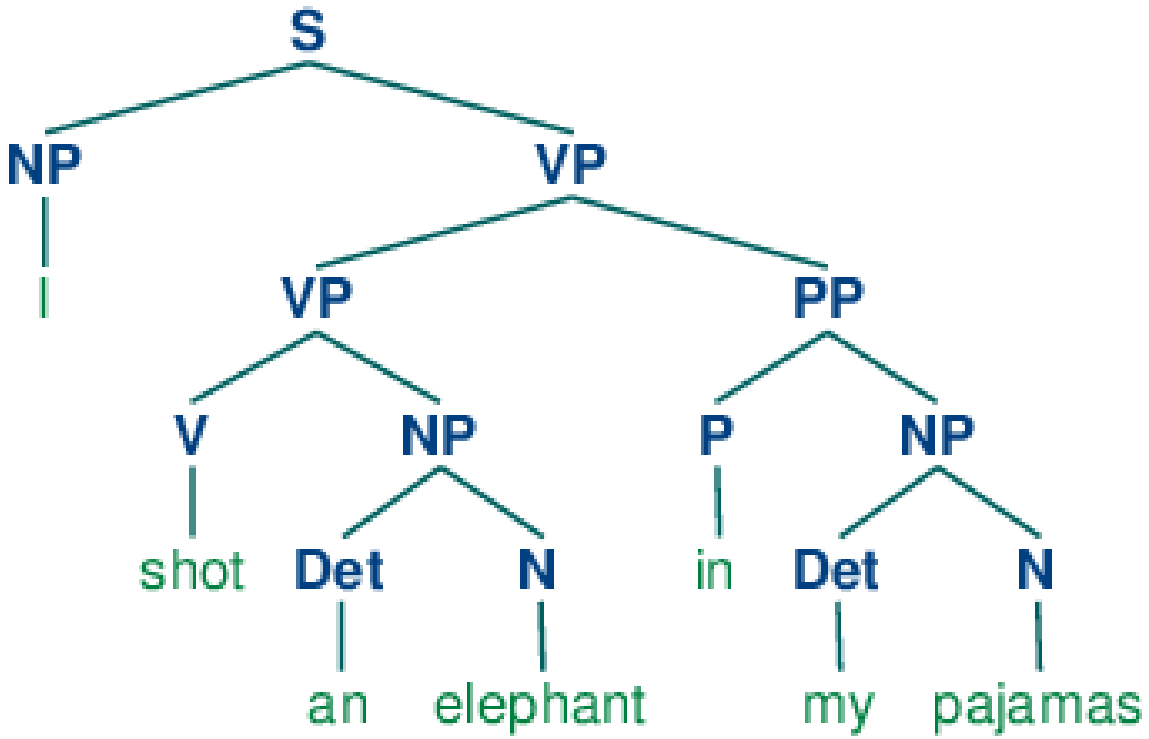
- Hierarchical representations are ubiquitous in AI. Computer vision:



Stages of Visual Representation, David Marr, 1970s

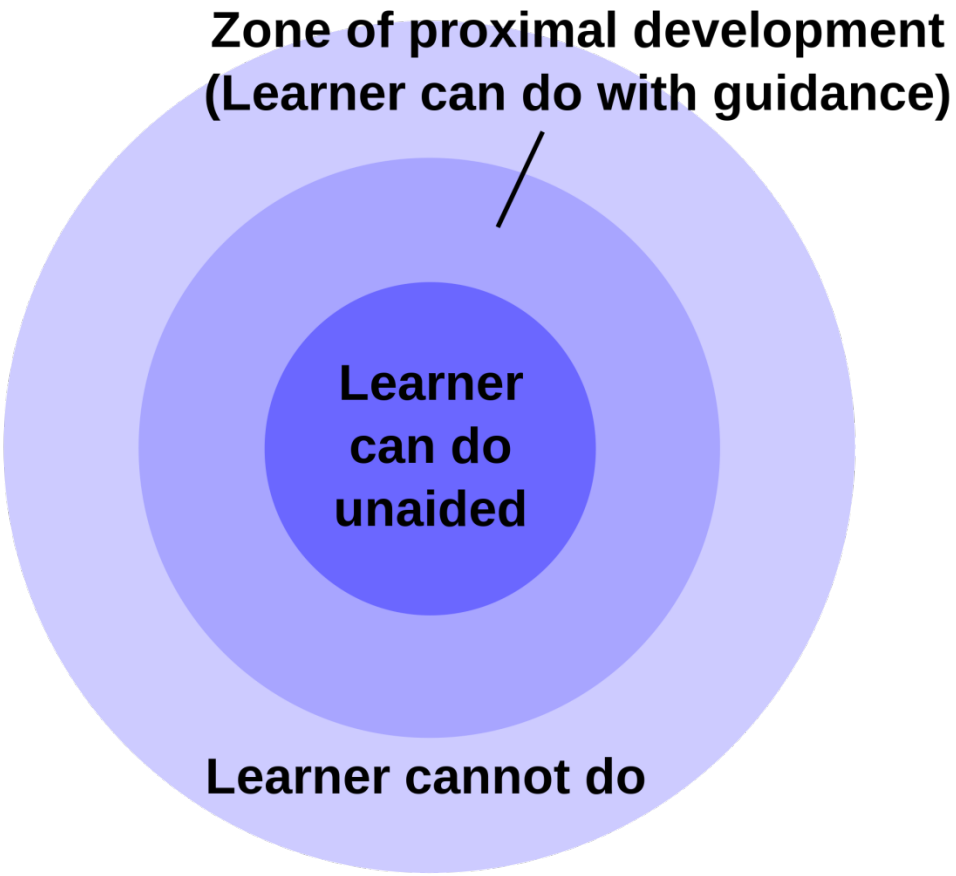
Why the success of DNNs is surprising ~~obvious~~

- Natural language:



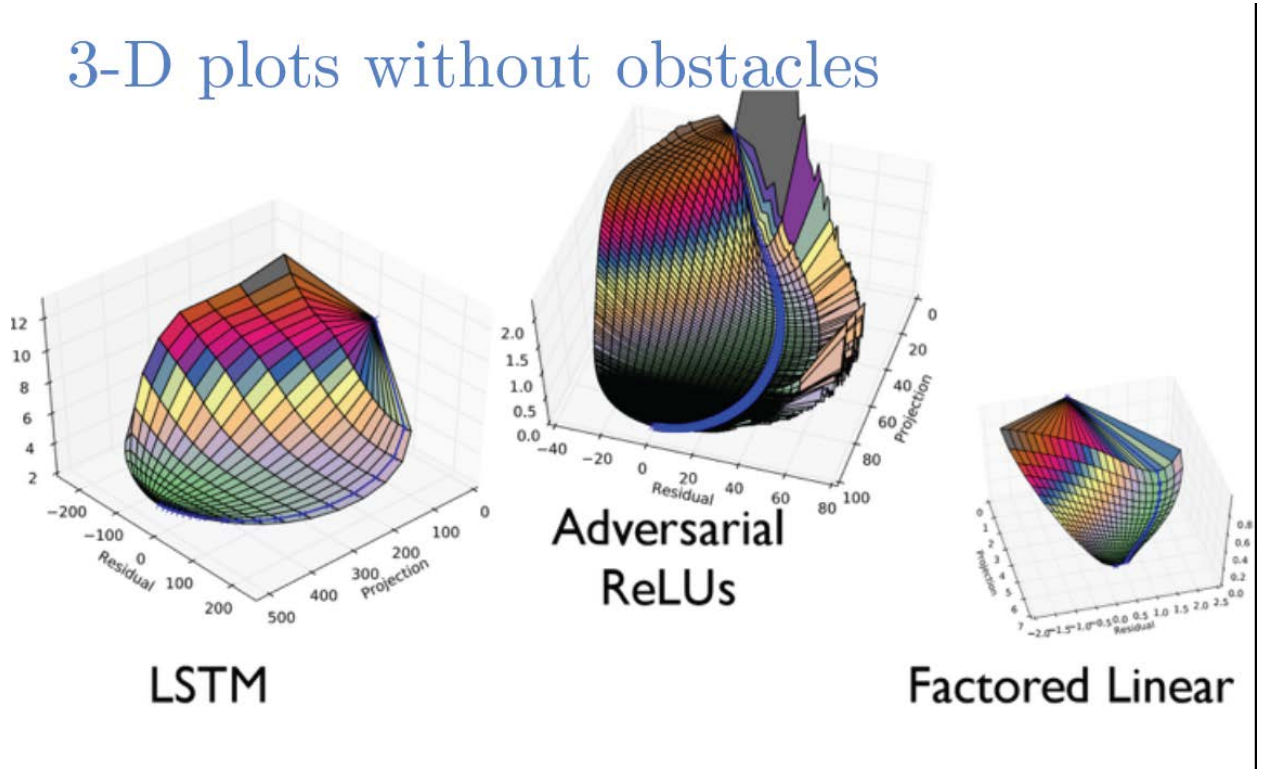
Why the success of DNNs is ~~surprising~~ obvious

- Human Learning: is deeply layered.

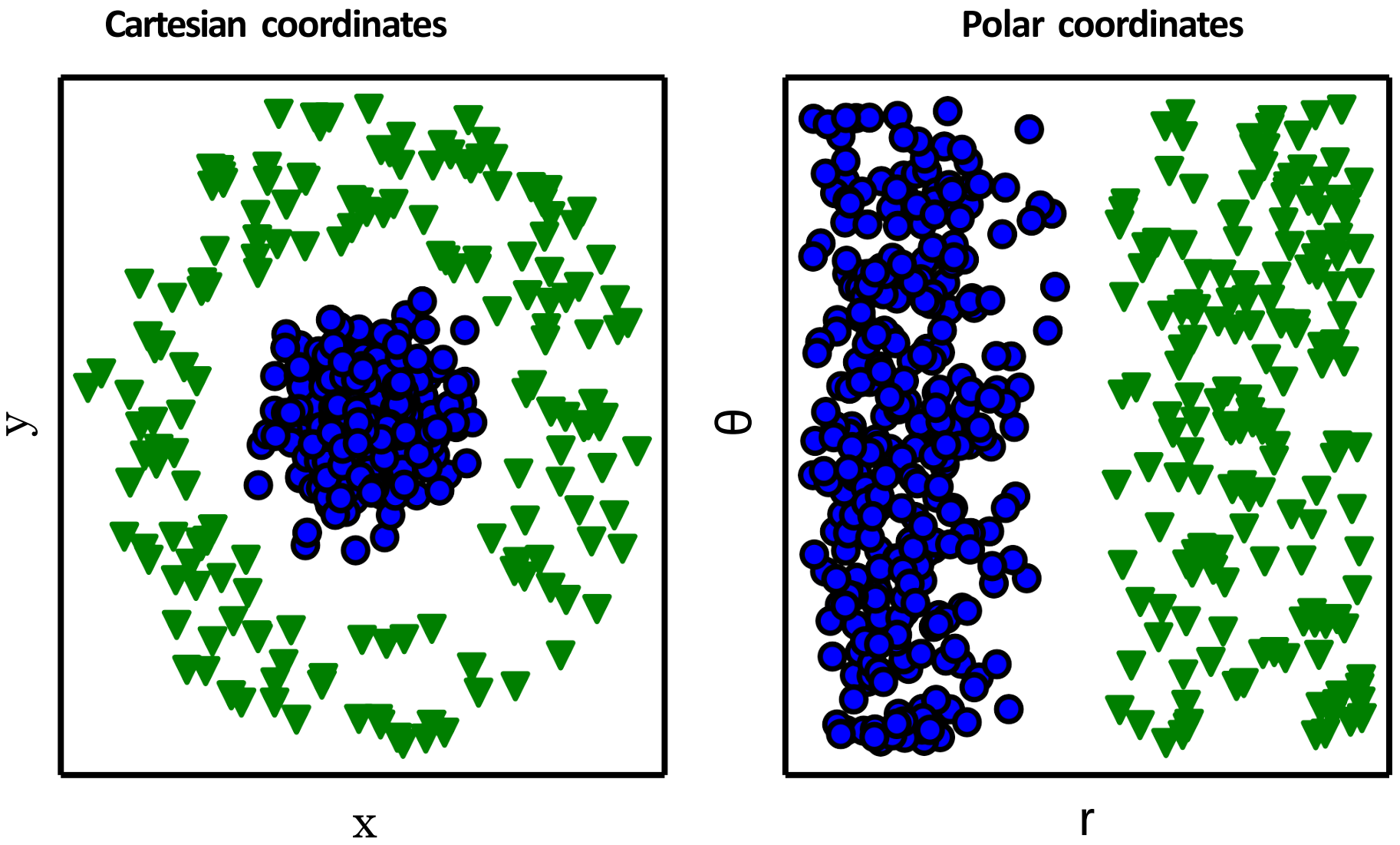


Why the success of DNNs is surprising ~~obvious~~

- What about greedy optimization?
- Less obvious, but it looks like many learning problems (e.g. image classification) are actually “easy” i.e. have reliable steepest descent paths to a good model.



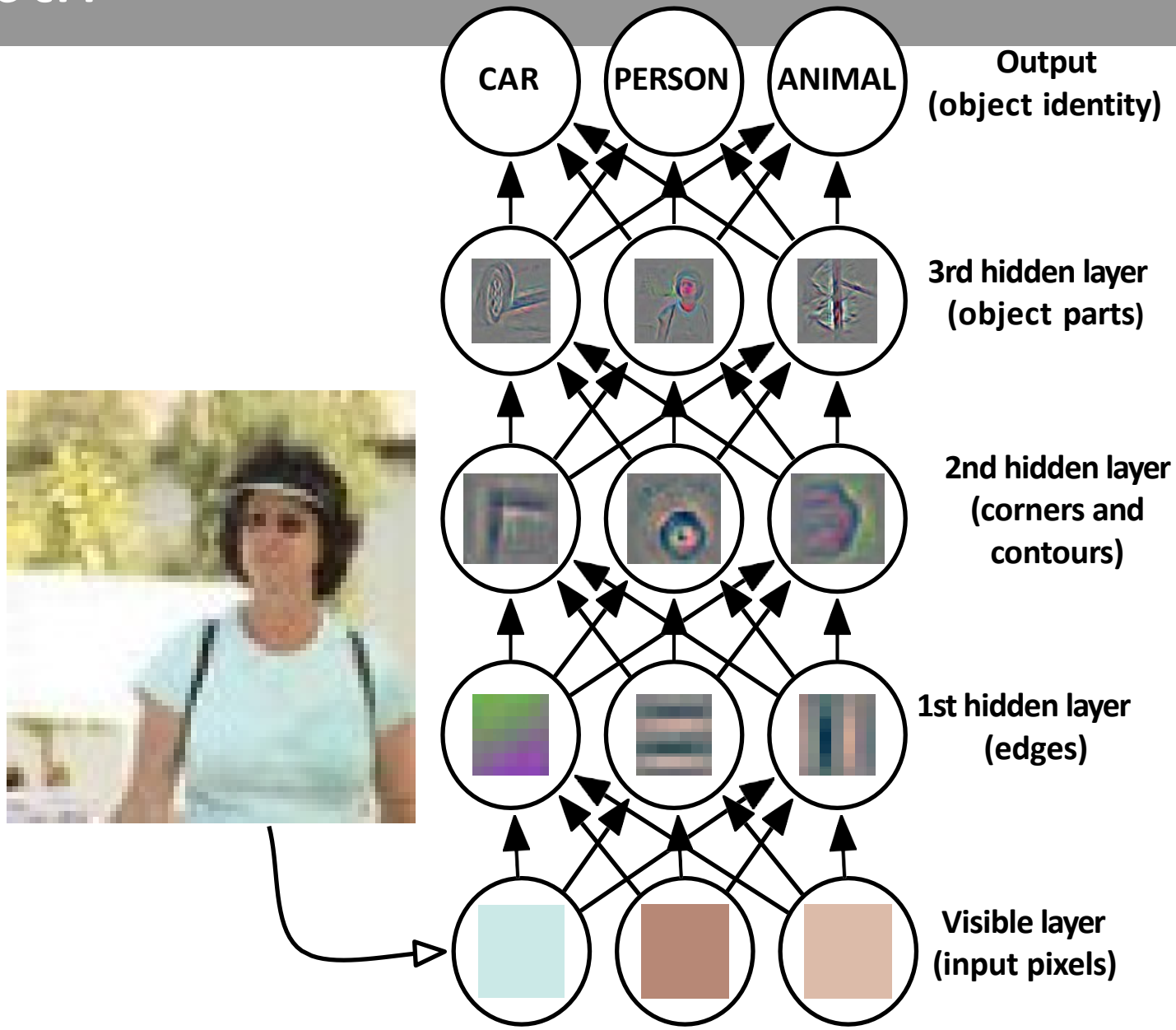
Representations Matter

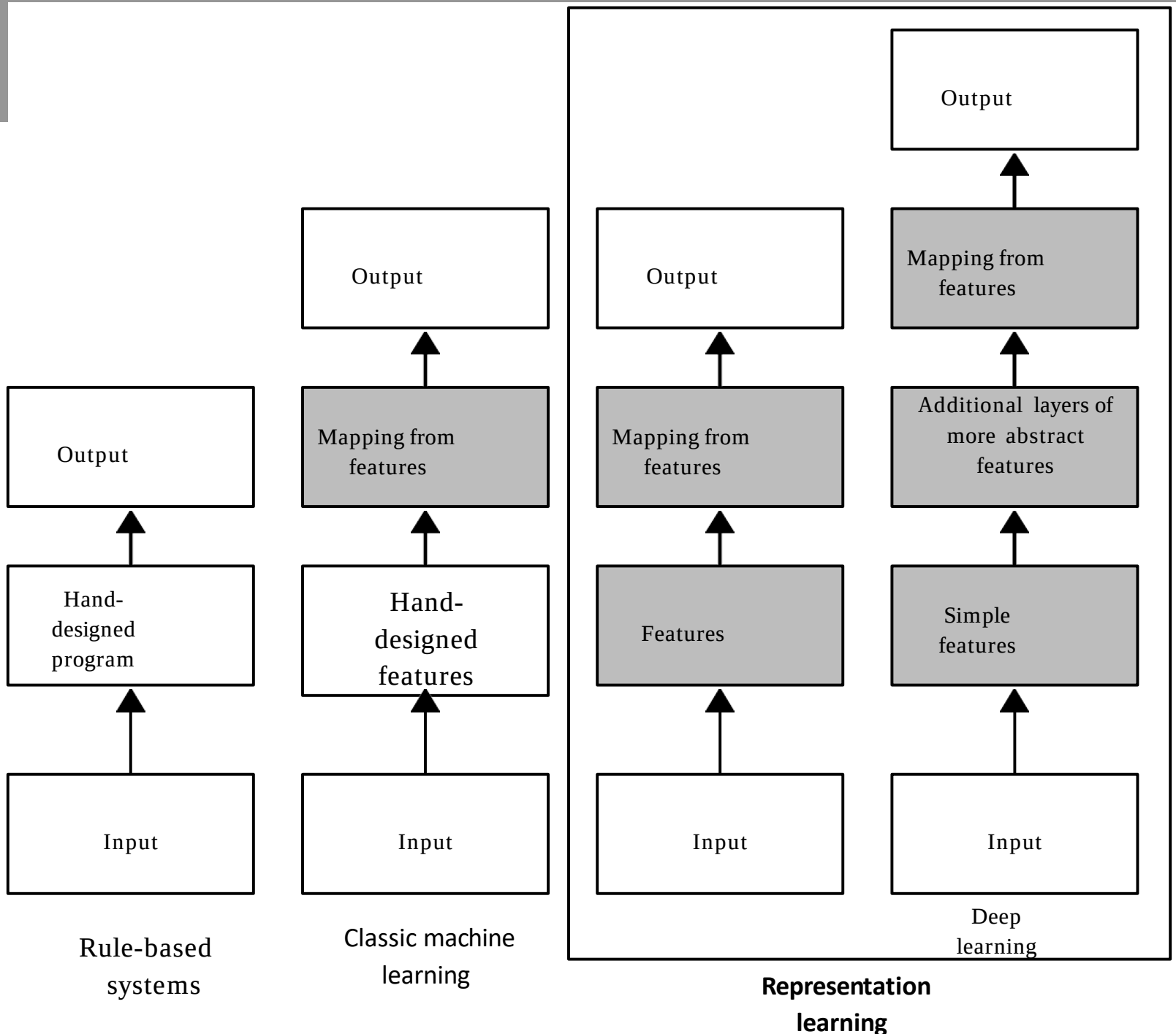


Representation Learning

- Use machine learning to discover not only the mapping from representation to output but also the representation itself.
- **Representation Learning**
- Learned representations often result in much better performance than can be obtained with hand-designed representations.
- They also enable AI systems to rapidly adapt to new tasks, with minimal human intervention.

Depth


CAR
PERSON
ANIMAL
Output
(object identity)
3rd hidden layer
(object parts)
2nd hidden layer
(corners and
contours)
1st hidden layer
(edges)
Visible layer
(input pixels)


Output
Mapping from features
Additional layers of more abstract features
Simple features
Input
Output
Mapping from features
Features
Input
Output
Hand-designed features
Input
Output
Hand-designed program
Input
Rule-based systems
Classic machine learning
Deep learning
Representation learning

ML BASICS

Definition

- Mitchell (1997) “A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E.”

Linear Regression

- In the case of linear regression, the output is a linear function of the input. Let $\hat{y}$ be the value that our model predicts y should take on. We define the output to be

$$\hat{y} = w^T x$$

$$MSE_{test} = \frac{1}{m} \left\| \hat{y}^{(test)} - y^{(test)} \right\|_2^2$$

$$\nabla_{\boldsymbol{w}} \mathrm{MSE}_{\mathrm{train}} = 0$$

$$\Rightarrow \nabla_{\boldsymbol{w}} \frac{1}{m} ||\hat{\boldsymbol{y}}^{(\mathrm{train})} - \boldsymbol{y}^{(\mathrm{train})}||_2^2 = 0$$

$$\Rightarrow \frac{1}{m} \nabla_{\boldsymbol{w}} ||\boldsymbol{X}^{(\mathrm{train})} \boldsymbol{w} - \boldsymbol{y}^{(\mathrm{train})}||_2^2 = 0$$

Normal Equations

$$\Rightarrow \nabla_{\boldsymbol{w}} \left(\boldsymbol{X}^{(\text{train})} \boldsymbol{w} - \boldsymbol{y}^{(\text{train})} \right)^{\top} \left(\boldsymbol{X}^{(\text{train})} \boldsymbol{w} - \boldsymbol{y}^{(\text{train})} \right) = 0$$

$$\Rightarrow \nabla_{\boldsymbol{w}} \left(\boldsymbol{w}^{\top} \boldsymbol{X}^{(\text{train})\top} \boldsymbol{X}^{(\text{train})} \boldsymbol{w} - 2\boldsymbol{w}^{\top} \boldsymbol{X}^{(\text{train})\top} \boldsymbol{y}^{(\text{train})} + \boldsymbol{y}^{(\text{train})\top} \boldsymbol{y}^{(\text{train})} \right) = 0$$

$$\Rightarrow 2\boldsymbol{X}^{(\text{train})\top} \boldsymbol{X}^{(\text{train})} \boldsymbol{w} - 2\boldsymbol{X}^{(\text{train})\top} \boldsymbol{y}^{(\text{train})} = 0$$

$$\Rightarrow \boldsymbol{w} = \left(\boldsymbol{X}^{(\text{train})\top} \boldsymbol{X}^{(\text{train})} \right)^{-1} \boldsymbol{X}^{(\text{train})\top} \boldsymbol{y}^{(\text{train})}$$

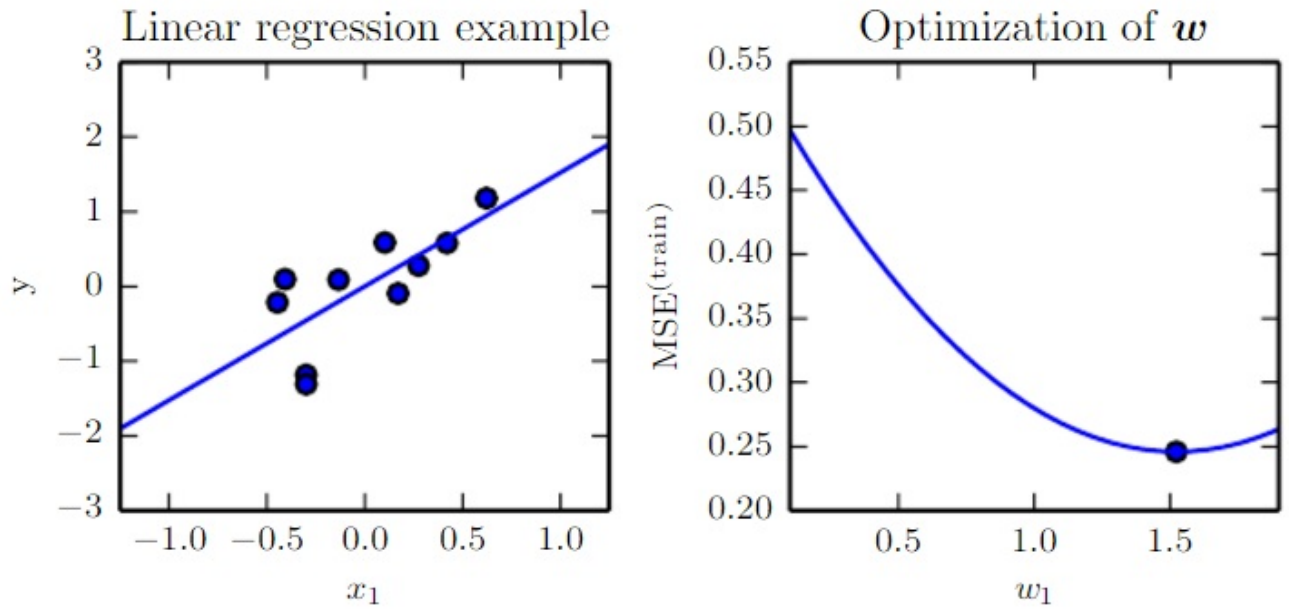


Figure 5.1: A linear regression problem, with a training set consisting of ten data points, each containing one feature. Because there is only one feature, the weight vector $\boldsymbol{w}$ contains only a single parameter to learn, w_1. *(Left)*Observe that linear regression learns to set w_1 such that the line $y = w_1 x$ comes as close as possible to passing through all the training points. *(Right)*The plotted point indicates the value of w_1 found by the normal equations, which we can see minimizes the mean squared error on the training set.