CH 1 INTRODUCTION

DREAM ▶ GOAL ! **CREATE MACHINES THAT THINK**

From ancient Greece
to Lovelace
to AI.

TASKS

	TASKS	
✗	abstract, informal chess ...	✓
✓	recognize object, speech	✗ → ✓ with DEEP LEARNING

TRUE CHALLENGE

▼

Solving tasks that are easy for people to perform but hard for people to describe

▼

Problems that we solve INTUITIVELY
like... recognizing spoken words or faces in images

HOW WOULD A COMPUTER LEARN?

↳ learn from experience
↳ understand the world in terms of hierarcy of concepts

▼

ability to learn complicated concepts!

Concept is defined through its relation to simpler concepts

buid via many layers
DEEP LEARNING

HISTORICAL TRENDS

1 CYBERNETICS 1940s - 1960s

2 CONNECTIONISM NEURAL NETWORKS 1980s - 1990s

3 DEEP LEARNING 2006 -

Significant changes over time

↑ DATASET SIZES

↑ MODEL SIZES

↑ ACCURACY | COMPLEXITY REAL-WORLD IMPACT

HOW TO GET THE INFORMAL KNOWLEDGE INTO A COMPUTER ?

1 KNOWLEDGE BASE

✗ ↳ hard-code knowledge in formal languages
Cyc (1989)

2 MACHINE LEARNING

✓ ↳ problems involving knowledge of the real world
↳ make decisions that seem subjective
LOGISTIC REGRESSION ; NAIVE BAYES
↳ depends heavily on the representation of data FEATURES

▷ not applicable to all features
▷ difficult to know which features to extract

3 REPRESENTATION LEARNING

✓ ↳ discover set of features ; learns representation
↳ ability for AI system to adopt to new tasks
↳ minimal human intervention

AUTOENCODER
ENCODER input ▶ different representation
DECODER new representation ▶ original format

FACTORS OF VARIATION
↳ separate sources of influence
↳ often not directly observed
image of a car — color , angle

! many factors influence every single piece of data observed

HOW TO EXTRACT HIGH-LEVEL ABSTRACT FEATURES ?

4 DEEP LEARNING

✓ ↳ discover representations for all the different tasks that we want to solve ▶ REPRESENTATION expressed in terms of other , simpler representations

MLP (MULTILAYER PERCEPTRON)

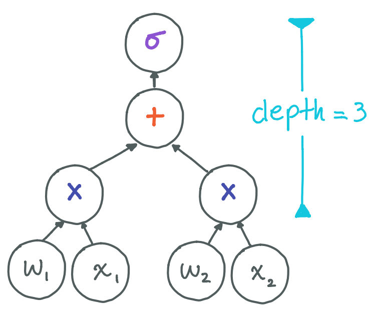
↳ DEPTH ▶ multistep ability ▶ SEQUENTIAL
layer = state of comp. memory after executing another set of instructions in parallel.

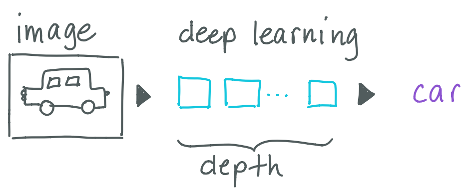
HOW TO MEASURE

COMPUTATIONAL GRAPH
1. number of sequential instructions exectuded

OR

PROBABILISTIC MODELS
2. how concepts are related



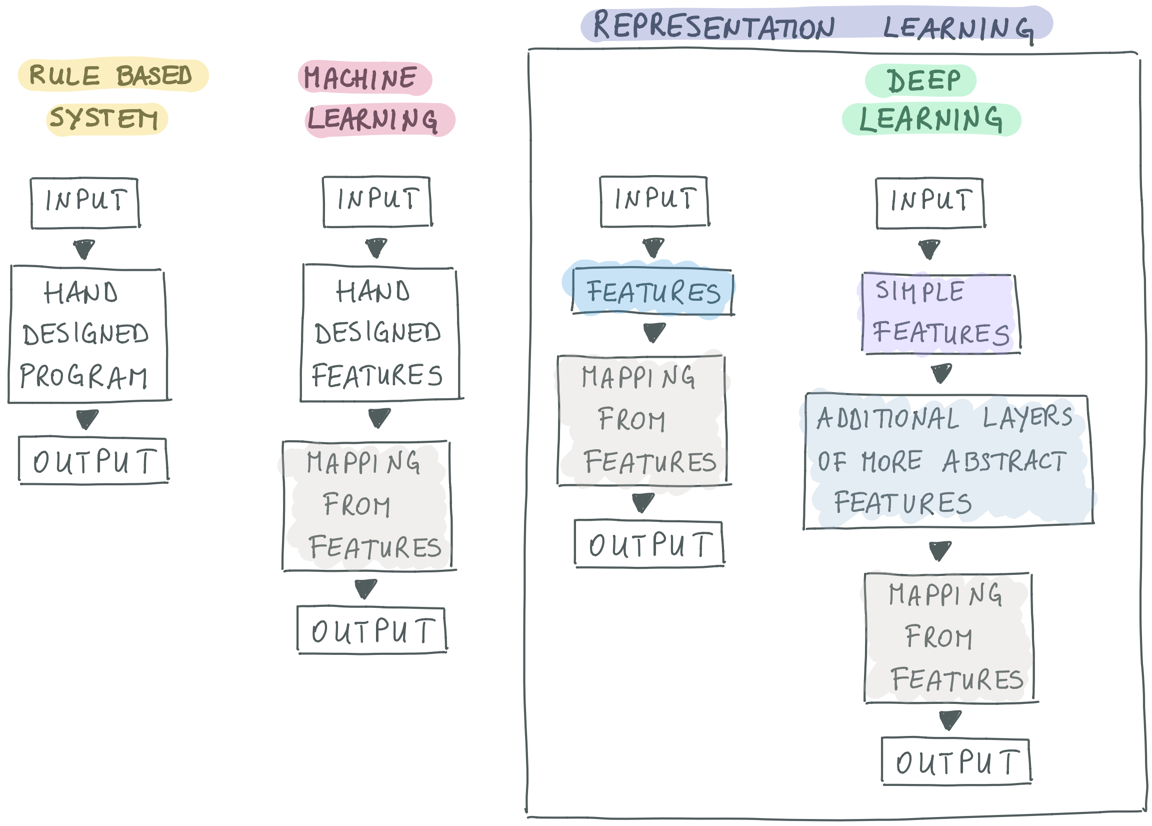


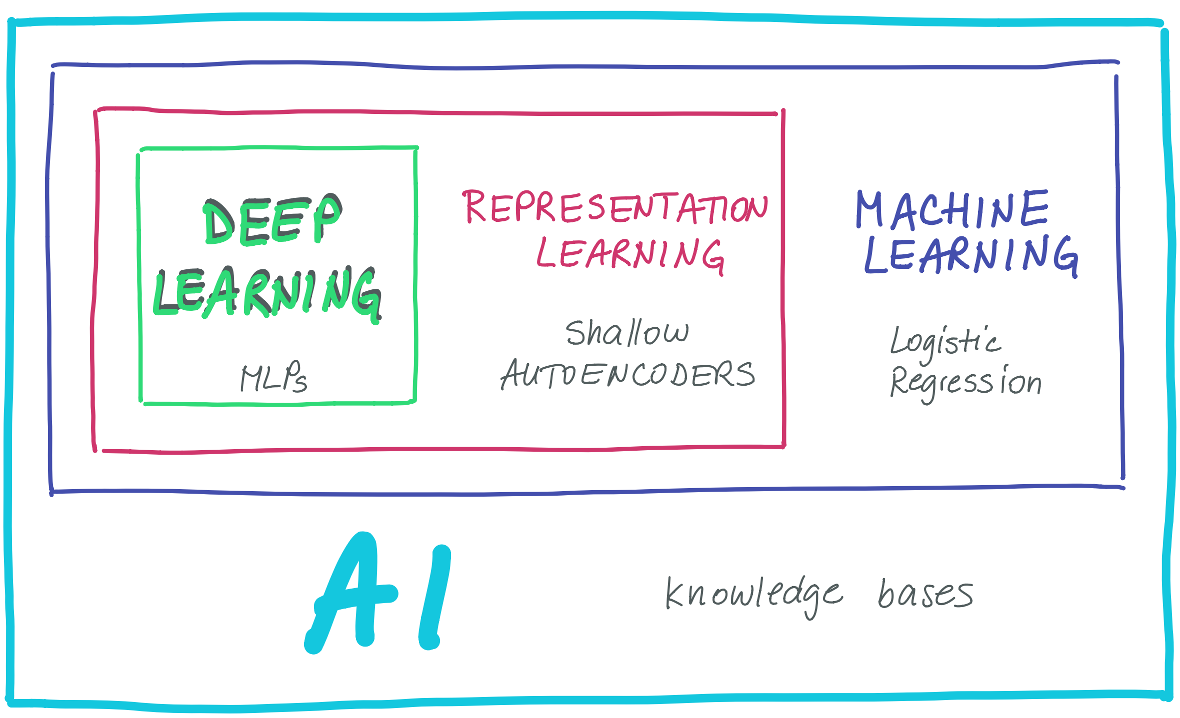
different sequential representations

later instructions can refer back to results from earlier instructions

also store state information

HOW DIFFERENT PARTS OF AN AI SYSTEM RELATE ...





HISTORICAL TRENDS

1 CYBERNETICS 1940s – 1960s

└ biological learning name ⊳ ARTIFICIAL NEURAL NETWORK

McCulloch and Pitts (1943)
- early model of brain function
- linear model
- weights set manually

▸ NN not designed to be realistic models of biological function

Rosenblatt (1958, 1962)
- perceptron (training of a single neuron)
- 1st model that could learn the weights

Widrow and Hoff (1960)
ADALINE (adaptive linear element)
└ training algorithm used ▸ a special case of **SGD** (stochastic gradient descent)

SIMPLE LINEAR MODELS

$x_1, \dots, x_n$ | $f(\boldsymbol{x}, \boldsymbol{w}) = x_1 w_1 + \dots + x_n w_n$

y | learn weights $\boldsymbol{w}$

LINEAR MODELS

models based on $f(\boldsymbol{x}, w)$
used by perceptron and ADALINE

▸ cannot learn the XOR function
Minsky and Papert (1969)

NEUROSCIENCE
- was important source of inspiration (no longer)
- not enough information about the brain

neocognitron – powerful model architecture for processing images
Fukushima (1980)

basis ↓

CONVOLUTIONAL NETWORK
LeCun et al. (1998b)

2

CONNECTIONISM (PARALLEL DISTRIBUTED PROCESSES)

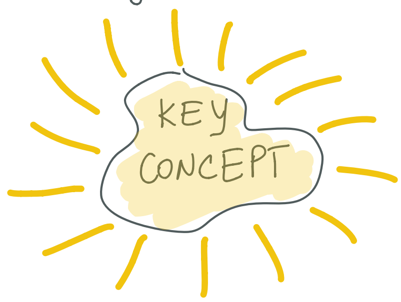
1980s - 1990s

- in context of cognitive science
- central idea ➡ large number of simple computational units can achieve intelligent behavior when networked together.

!!! **DISTRIBUTED REPRESENTATION**

Hinton et al. (1986)

- each input should be represented by many features
- each feature should be involved in the representation of many possible inputs



BACK-PROPAGATION algorithm ▶ dominant for training deep NN

Rumelhart et al. (1986a)

LeCun (1987)

1990s ▷ MODELLING SEQUENCES

Hochreiter (1991) ; Bengio et al. (1994)

modeling long sequences ; identified some of the math. difficulties

Hochreiter (1997) ▷ **LSTM** (long short term memory)

- used for many sequence based tasks
- NLP tasks at Google

3

DEEP LEARNING 2006 -

DEEP BELIEF NETWORK

Hinton et al. (2006)

- can be efficiently trained using greedy layer-wise pretraining

Bengio et al (2007) ;

Ranzato et al. (2007)

- can also be used to train other kinds of deep networks
- systematically help improve generalization

▶ train deeper networks

▶ focus on theoretical concept of depth

▶ focus on unsupervised learning techniques ➡ today: more interest in supervised learning algorithms ; large labeled datasets

DATASET SIZES

TRAINING DATA ↑ SKILL REQUIRED ↓

- datasets have expanded significantly over time
  - early 1990s — hundreds | thousand manuall compiled
  - 1950s - 1980s — small synthetic datasets
  - 1980s - 1990s — MNIST
  - 2000s — CIFAR-10
  - 2010s — $>10^4$ ImageNet, Sports-1M; 10^9 WMT
- trend driven by digitalization
- "Big Data"
- 2016 rule of thumb ; supervised DL

No. of labeled examples	Performance
5000	OK
10 million	human or better

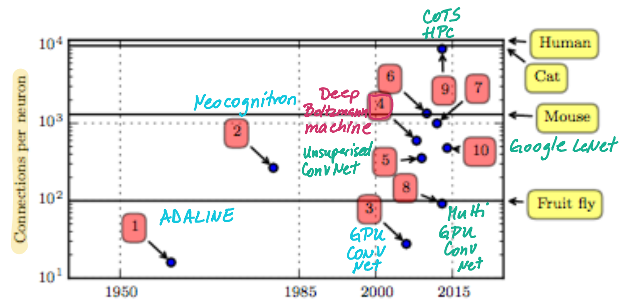
MODEL SIZES

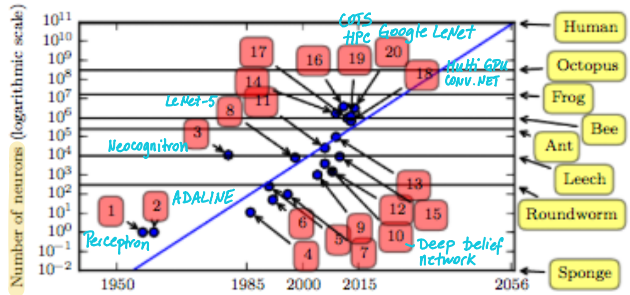
☑ computational resources to run large models

↳ faster CPUs ; advent of GPUs ; faster network connectivity ; better software infrastructure for distributed computing

- with hidden units , ANNs have doubled in size every 2.4 yrs.

☑ larger networks → higher accuracy





ACCURACY | COMPLEXITY REAL-WORLD IMPACT

* dramatic impact on image recognition; speech recognition
* significant drops in error rates

scale and accuracy ↑
complexity ↑

* sequence -to- sequence modeling
  MACHINE TRANSLATION !

* neural Turing machines
  learn to read from memory cells and write arbitrary content to memory cells

REINFORCEMENT LEARNING

* autonomous agent learns to perform a task by trial and error
  !!! no guidance from human operator !!!

Deep Mind → learn to play Atari games
→ human-level performance

! SOFTWARE INFRASTRUCTURE !

TensorFlow Caffe
Torch MXNet