CS60010: Deep Learning

Spring 2023

Sudeshna Sarkar

Module 1 Part A
Introduction

Sudeshna Sarkar

4 Jan 2023

Instructors

- Teacher: Sudeshna Sarkar
  - Email: sudeshna@cse.iitkgp.ac.in

- TAs:
  1. Alapan Kuila
  2. Vasudha Joshi
  3. Somnath Jena
  4. Debajyoti Dasgupta
  5. Sayan Mahapatra
  6. Upasana Mandal

Class Timetable

Room: NR222

Theory in Slot: G3

WED (11:00-11:55)

THURS (12:00-12:55)

FRI (08:00-08:55)

Hands-on Session with our TAs online on MS Teams

Thursday 7 pm

MS Teams for the class: DL23 CS60010

- Slides and other resources will be uploaded in MS Teams

Course Information

Prerequisites:

- Machine Learning
- Knowledge of calculus and linear algebra
- Probability and Statistics
- Python Proficiency: A few links to get started.

  https://docs.python.org/3/tutorial/

  http://cs231n.github.io/python-numpy-tutorial/

Books and References:

1. Deep Learning (Adaptive Computation and Machine Learning series) by Ian Goodfellow, Yoshua Bengio, Aaron Courville

   https://www.deeplearningbook.org/

2. Dive into Deep Learning

   https://d2l.ai/

3. Deep Learning for Coders with fastai and PyTorch by Jeremy Howard, Sylvain Gugger

   https://github.com/fastai/fastbook

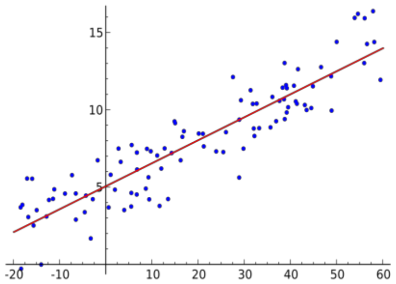
Evaluation

1. Tests: 60% (25 + 35)
2. Quiz + Assignments: 30%

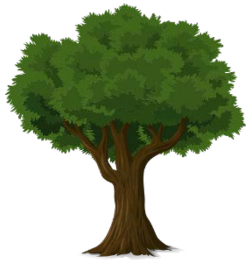
3. Research Paper reading
   Slide and Presentation: 10%

4. Quiz 1 on 13th Jan 2023

Different types of learning problems



predict y from x

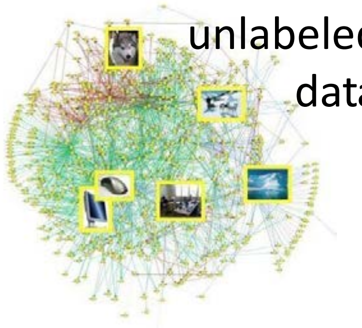


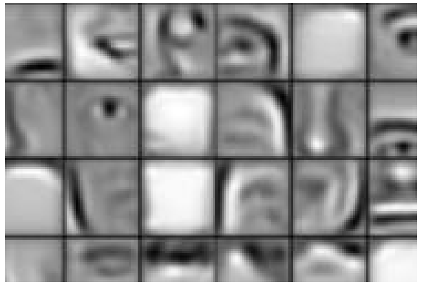
$f_\theta(x) = y$

[object label]

supervised learning

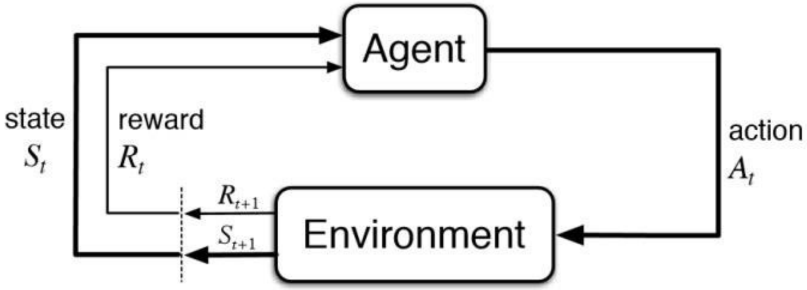
unlabeled data

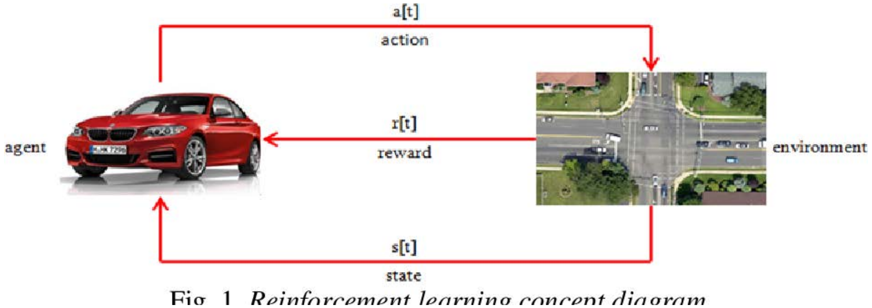




representation

unsupervised learning



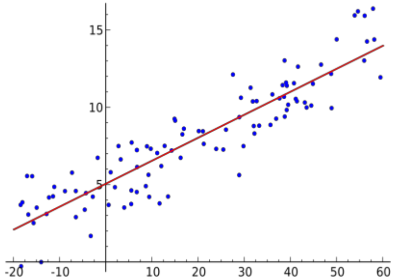


reinforcement learning

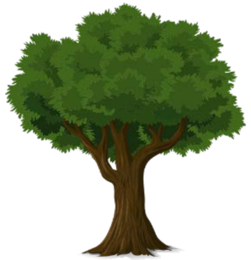
Supervised learning

Given: $\mathcal{D} = \{(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)\}$

learn $f_\theta(x) \approx y$

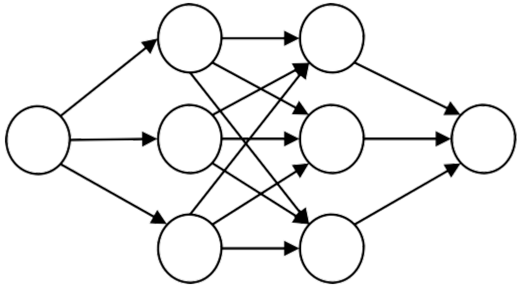


predict y from x



$f_\theta(x) = y$

[object label]



Questions to answer:

how do we represent $f_\theta(x)$?

$$f_\theta(x) = \theta_1 x_1 + \theta_2 x_2 + \theta_3$$

$$f_\theta(x) = \theta_1 x + \theta_2 x^2 + \theta_3 x^3$$

how do we measure difference between $f_\theta(x_i)$ and y_i?

$$||f_\theta(x_i) - y_i||^2$$

probability?

$$\delta(f_\theta(x_i) \neq y_i)$$

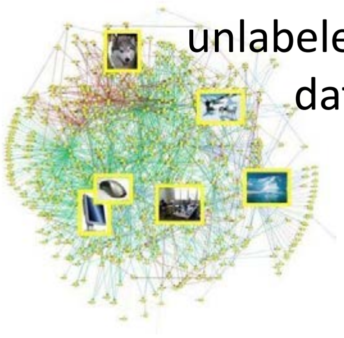
how do we find the best setting of θ?

gradient descent

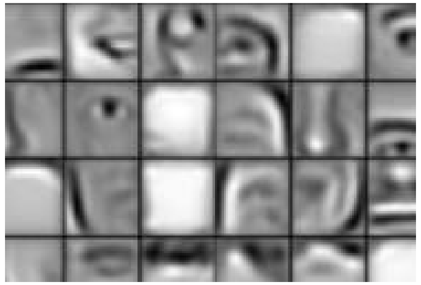
random search

least squares

Unsupervised learning



unlabeled
data



representation

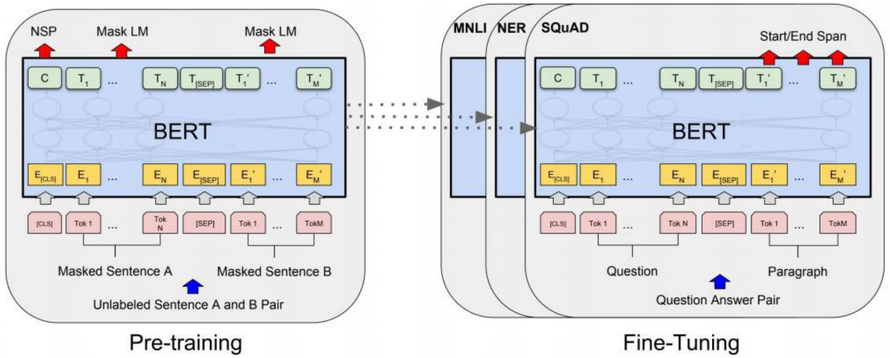
Generative Modelling

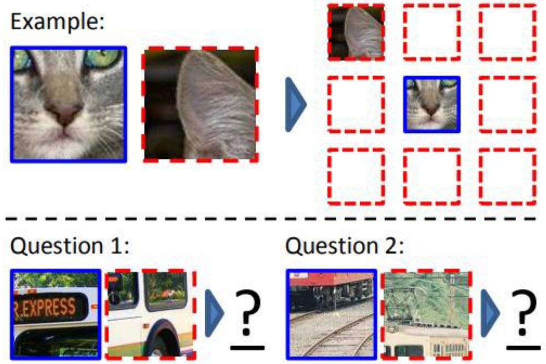
GANs

VAEs

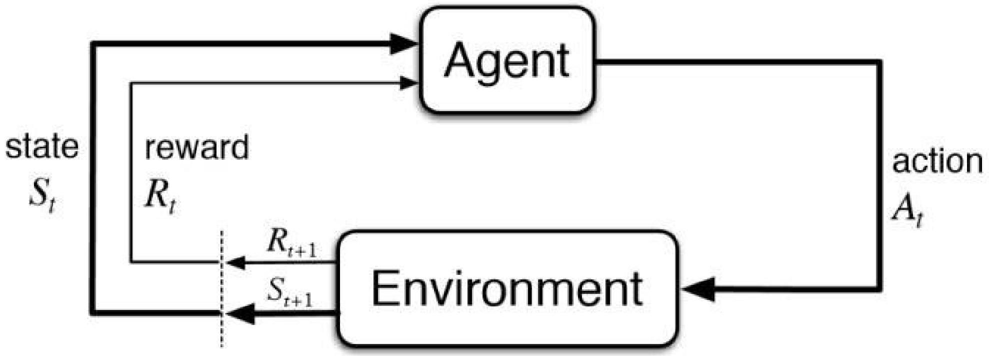
pixel RNN, etc.

self-supervised
representation learning:





Reinforcement learning



choose $f_\theta(s_t) = a_t$

to maximize $\sum_{t=1}^{H} r(s_t, a_t)$

actually subsumes (generalizes) supervised learning!

supervised learning: get $f_\theta(x_i)$ to match y_i

reinforcement learning: get $f_\theta(s_t)$ to maximize reward (could be anything)

Actions: muscle contractions
Observations: sight, smell
Rewards: food

Actions: motor current or torque
Observations: camera images
Rewards: task success measure (e.g., running speed)

Actions: what to purchase
Observations: inventory levels
Rewards: profit

Phases of Neural Network Research

1940s-1960s: Cybernetics: Brain like electronic systems, morphed into modern control theory and signal processing.

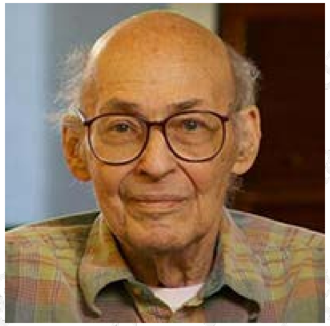
1960s-1980s: Digital computers, automata theory, computational complexity theory: simple shallow circuits are very limited…

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.

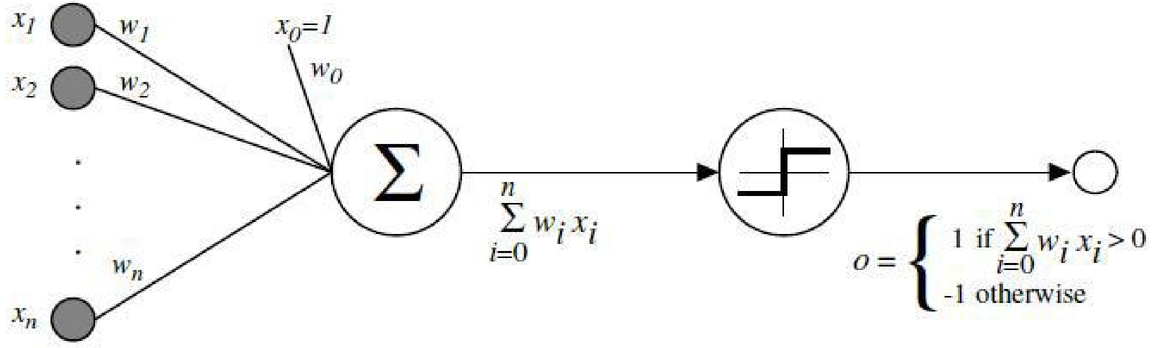
First appearance (roughly)

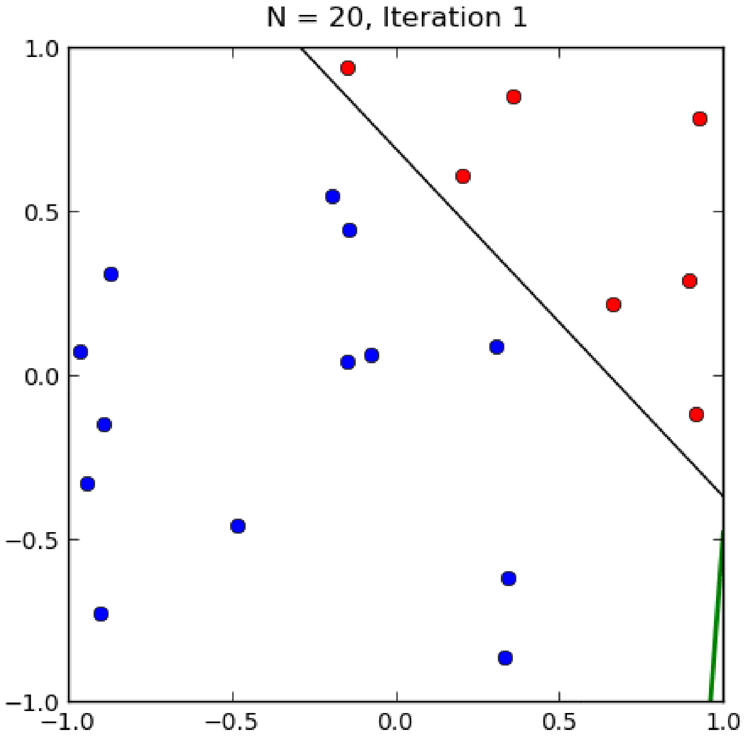


- 1958 — Perceptrons, Rosenblatt
- 1960 — Adaline, Widrow and Hoff
- 1969 — Perceptrons, Minsky and Papert

Perceptrons

- Rosenblatt proposed perceptrons for binary classifications and a learning algorithm for perceptrons

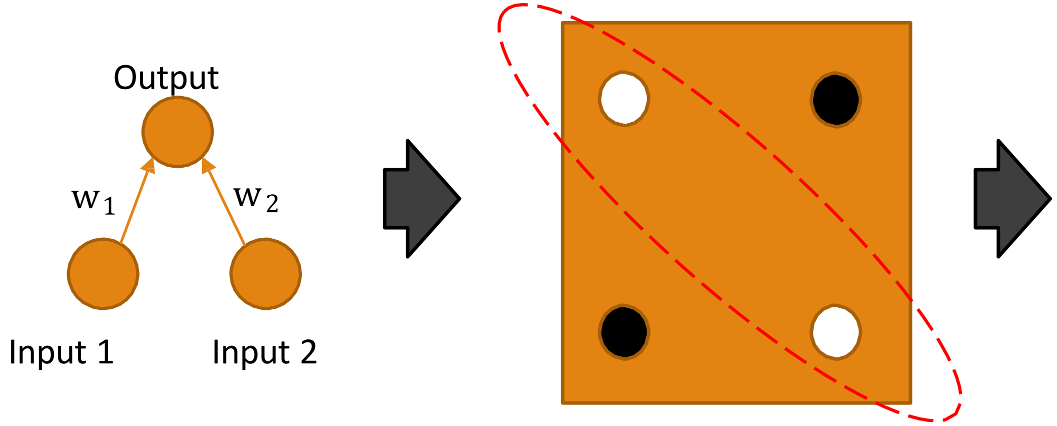




Limitations of Perceptrons

- The original perceptron has trouble with simple non-linear tasks

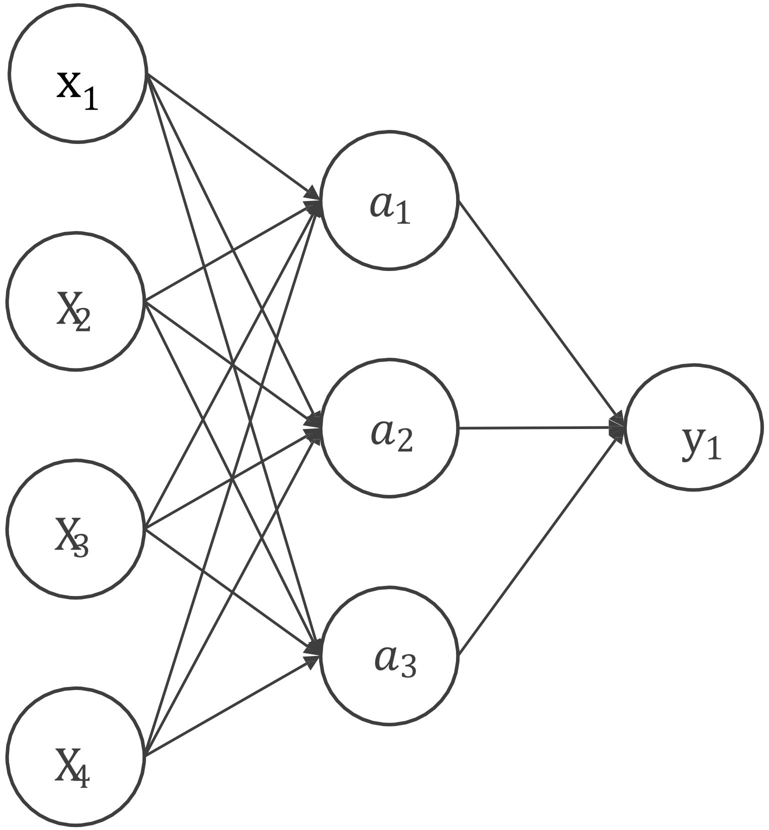
Minsky and Papert, "Perceptrons", 1969



No line can separate the white from the black

Multi-layer perceptrons

- Multi-layer perceptrons (MLP) can solve XOR
  - One layer's output is input to the next layer
  - Add nonlinearities between layers, e.g., sigmoids
- Problem: how to train a multi-layer perceptron?
- Learning depends on "ground truth" for updating weights
- For the intermediate neurons there is no "ground truth"



AI Winter despite successes

- 1958 — Perceptrons, Rosenblatt
- 1960 — Adaline, Widrow and Hoff
- 1969 — Perceptrons, Minsky and Papert
- 1970 — Backpropagation, Linnainmaa
- 1974 — Backpropagation, Werbos
- 1986 — Backpropagation, Rumelhart, Hinton and Williams
- 1997 — LSTM, Hochreiter and Schmidhuber
- 1998 — OCR, LeCun, Bottou, Bengio and Haffner
- 2006 — Deep Learning, Hinton, Osindero, Teh
- 2009 — Imagenet, Deng et al.
- 2013 — Alexnet, LeCun, Bottou, Bengio and Haffner
- 2015 — Resnet (154 layers), MSRA
- today — GO, Deepmind

The first *AI Winter*

- What everybody thought

“If a perceptron cannot even solve XOR, why bother?”

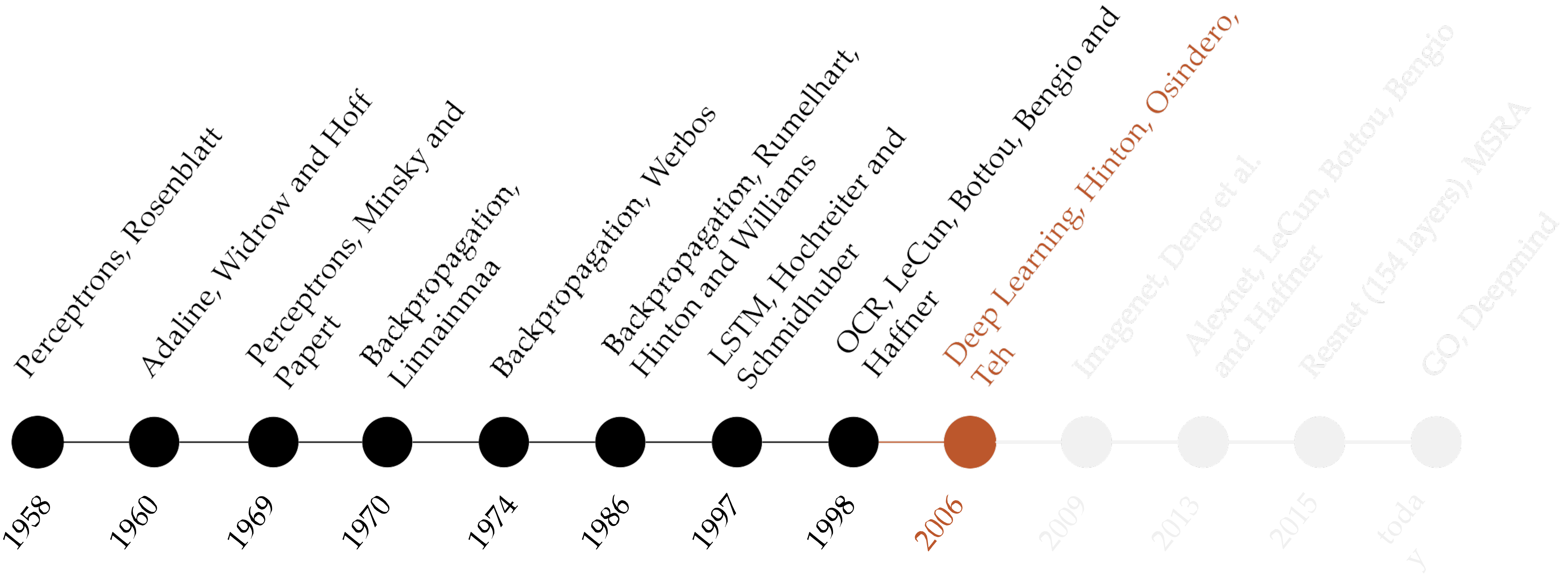
- Results not as promised (too much hype!)
  → no further funding
  → AI Winter

- Still, significant discoveries were made in this period
  - Backpropagation → Learning algorithm for MLPs
  - Recurrent networks → Neural Networks for infinite sequences

The 2nd “AI Winter”

- Concurrently with Backprop and Recurrent Nets, new and promising Machine Learning models were proposed
- Kernel Machines & Graphical Models
  - Similar accuracies with better math and proofs and fewer heuristics
  - Neural networks could not improve beyond a few layers

The thaw of the “AI Winter”



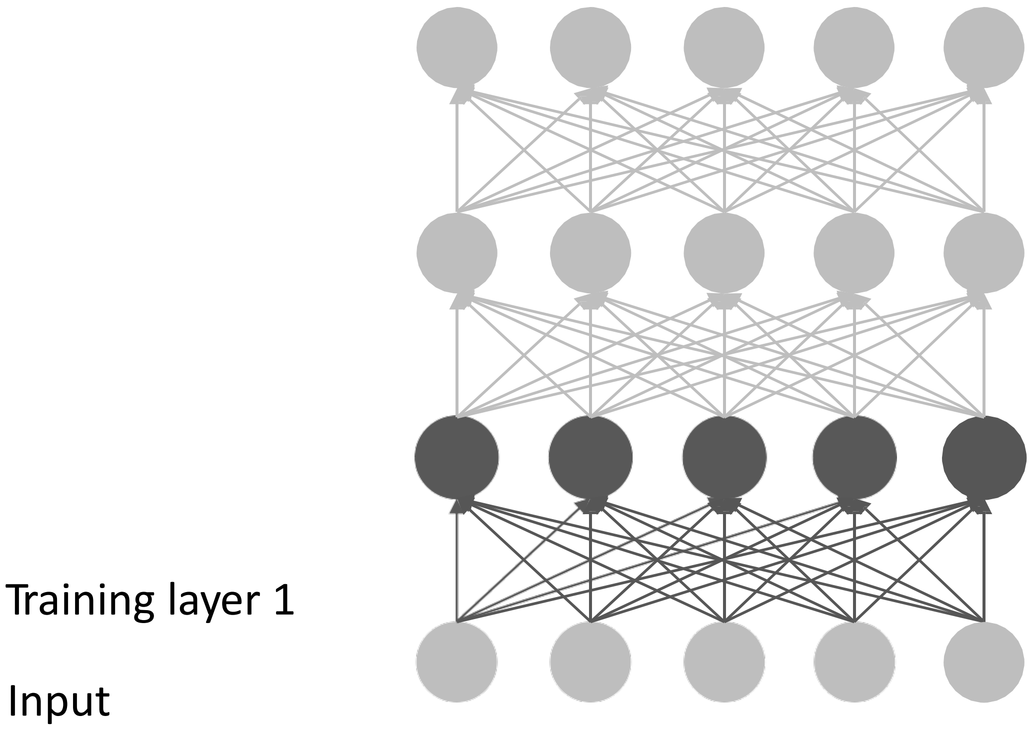
Neural Networks: A decade ago

- Lack of processing power
- Lack of data
- Overfitting
- Vanishing gradients
- Experimentally, training multi-layer perceptrons was not that useful

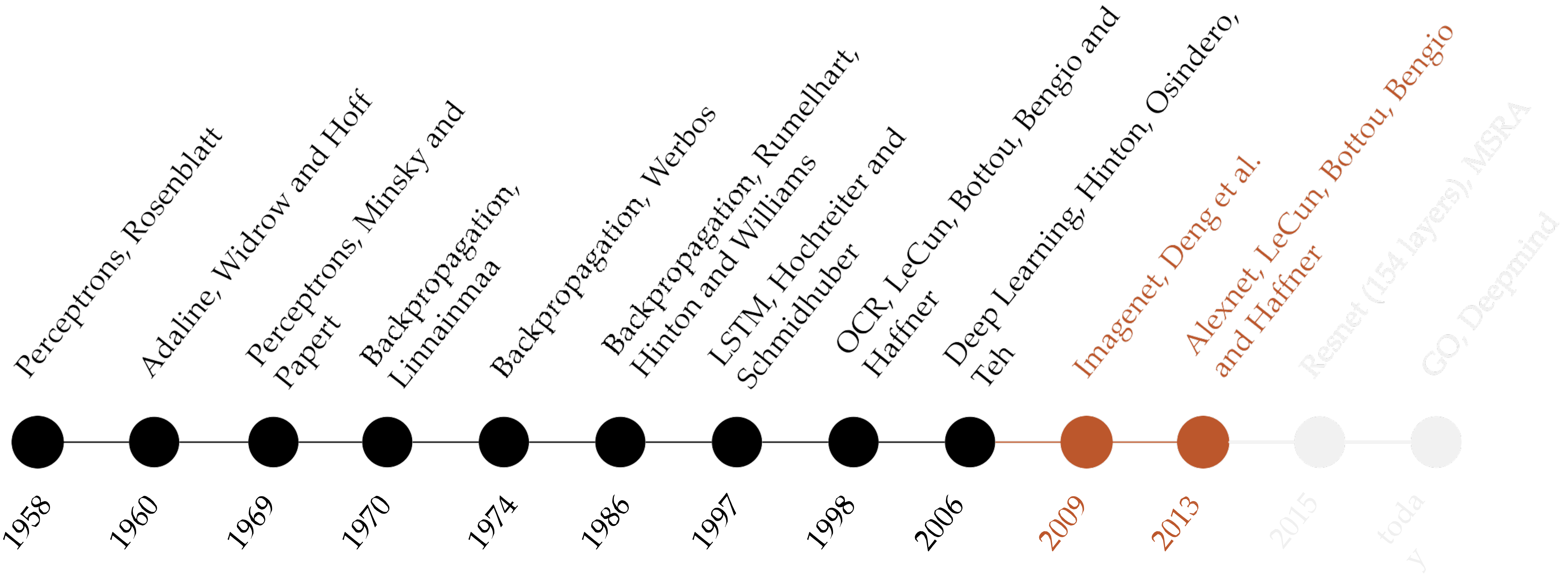
"Are 1-2 hidden layers the best neural networks can do?"

Deep Learning arrives

- Easier to train one layer at a time → Layer-by-layer training
- Training multi-layered neural networks became easier
- After, keep training with contrastive divergence



Deep Learning Renaissance



Deep Learning is Data Hungry

- In 2009 the Imagenet dataset was published [Deng et al., 2009]
  - Collected images for all 100K terms in Wordnet (16M images in total)
  - Terms organized hierarchically: “Vehicle”→“Ambulance”
- Imagenet Large Scale Visual Recognition Challenge (ILSVRC)
  - 1 million images, 1,000 classes, top-5 and top-1 error measured

The Deep Learning Revolution

- Speech recognition: 2010
- Image recognition: 2013
- Natural language processing: 2015

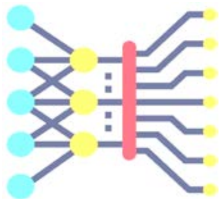
The success of NN

1. More data
2. More computational power
3. Improved techniques (though they're not brand-new)

But, Driven primarily by intuition and empirical success

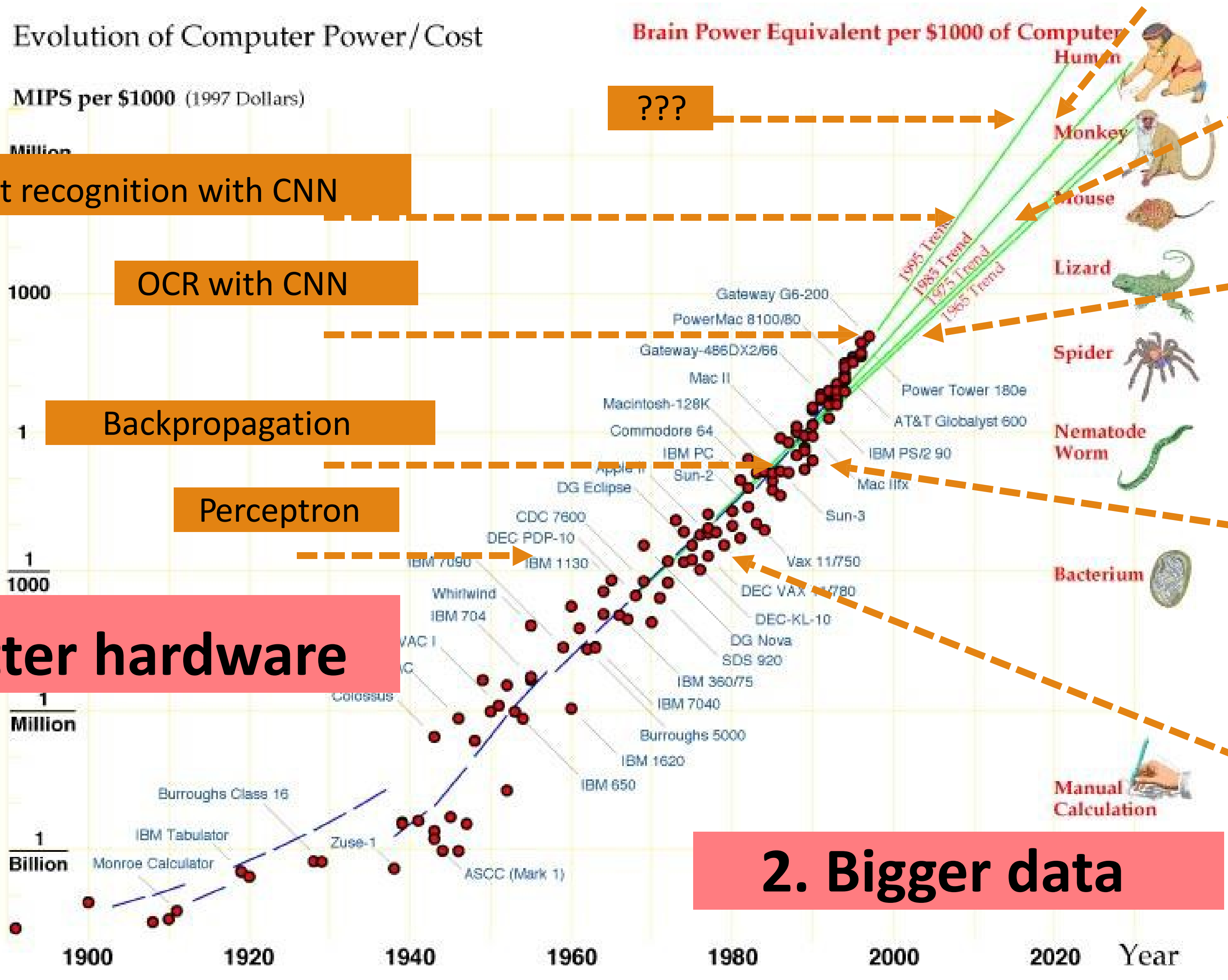
- Good research and progress based on solid intuition
- Practice leads the way
- Theory lags dramatically
  - no guarantees
  - little understanding of limitations
  - limited interpretability
- More interestingly, classic theory suggests currently successful DL practices, wouldn't be likely to succeed.

Datasets of everything (captions, question-answering, …), reinforcement learning, ???



Imagenet: 1,000 classes from real images, 1,000,000 images

Results:
- Persian cat: 0.35211
- Egyptian cat: 0.23635
- hamster: 0.20282
- tiger cat: 0.05896
- lynx: 0.05759



???

Object recognition with CNN

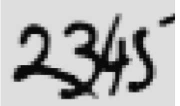
OCR with CNN

Backpropagation

Perceptron

1. Better hardware

Bank cheques

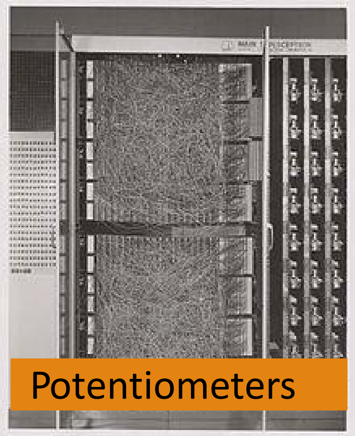


Parity, negation problems

D1	D2	D3	Even-Parity
0	0	0	True
0	0	1	False
0	1	0	False
0	1	1	True
1	0	0	False
1	0	1	True
1	1	0	True
1	1	1	False

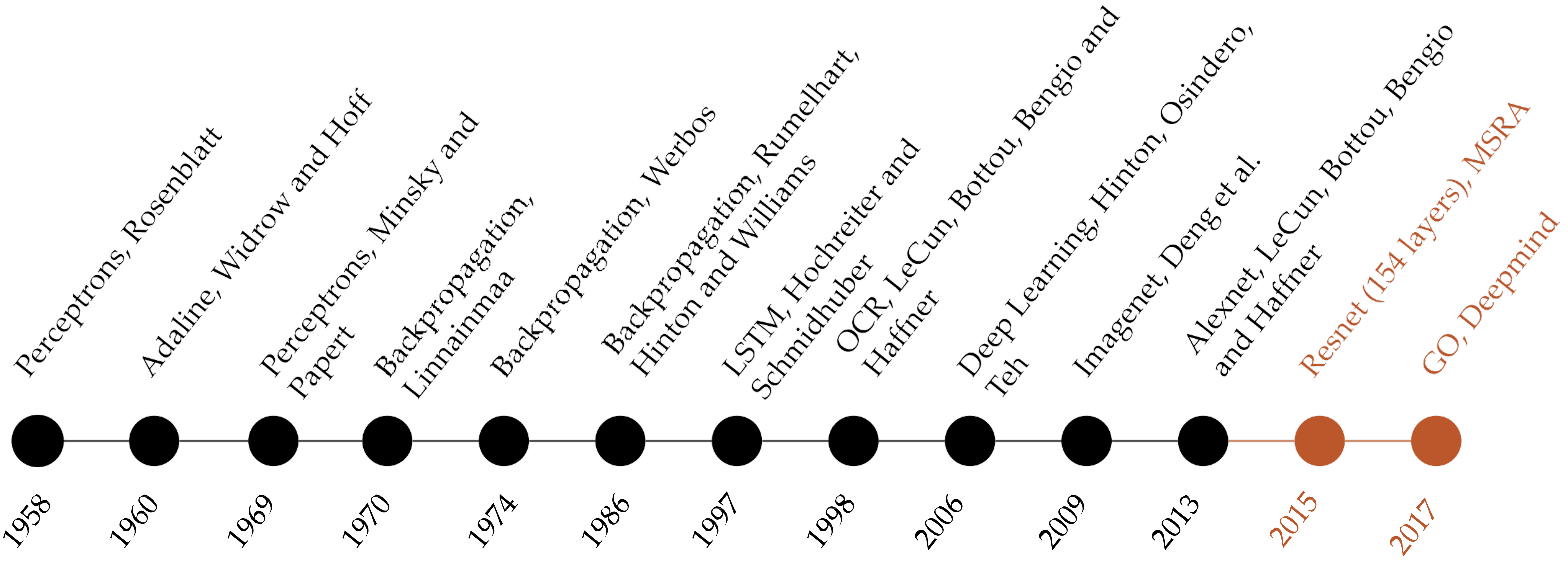
2. Bigger data

Mark I Perceptron



Potentiometers

Deep Learning Golden Era



What is so impressive?

- Vision is ultra challenging!
  - For 256x256 resolution → $2^{524,288}$ of possible i m
  - Large semantic & visual object variations

Intra-class overlap

- Robotics is typically considered in controlled environments

Inter-class variation

- Game AI involves extreme number of possible games states ($10^{10^{48}}$ possible GO games)
- NLP is extremely high dimensional and vague (just for English: 150K words)
- Deep learning seems to casually solve many (supervised) problems thought to be extremely hard

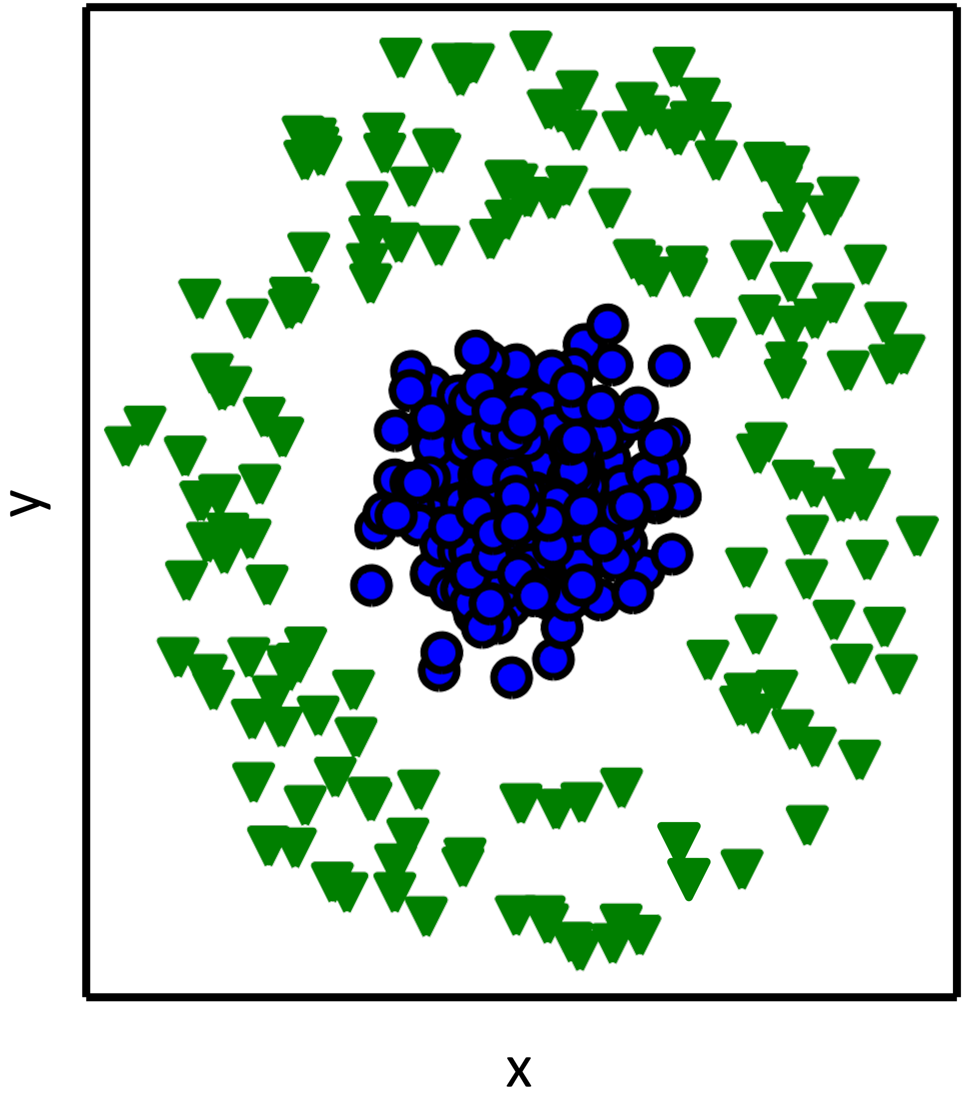
Representation Learning

- 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.

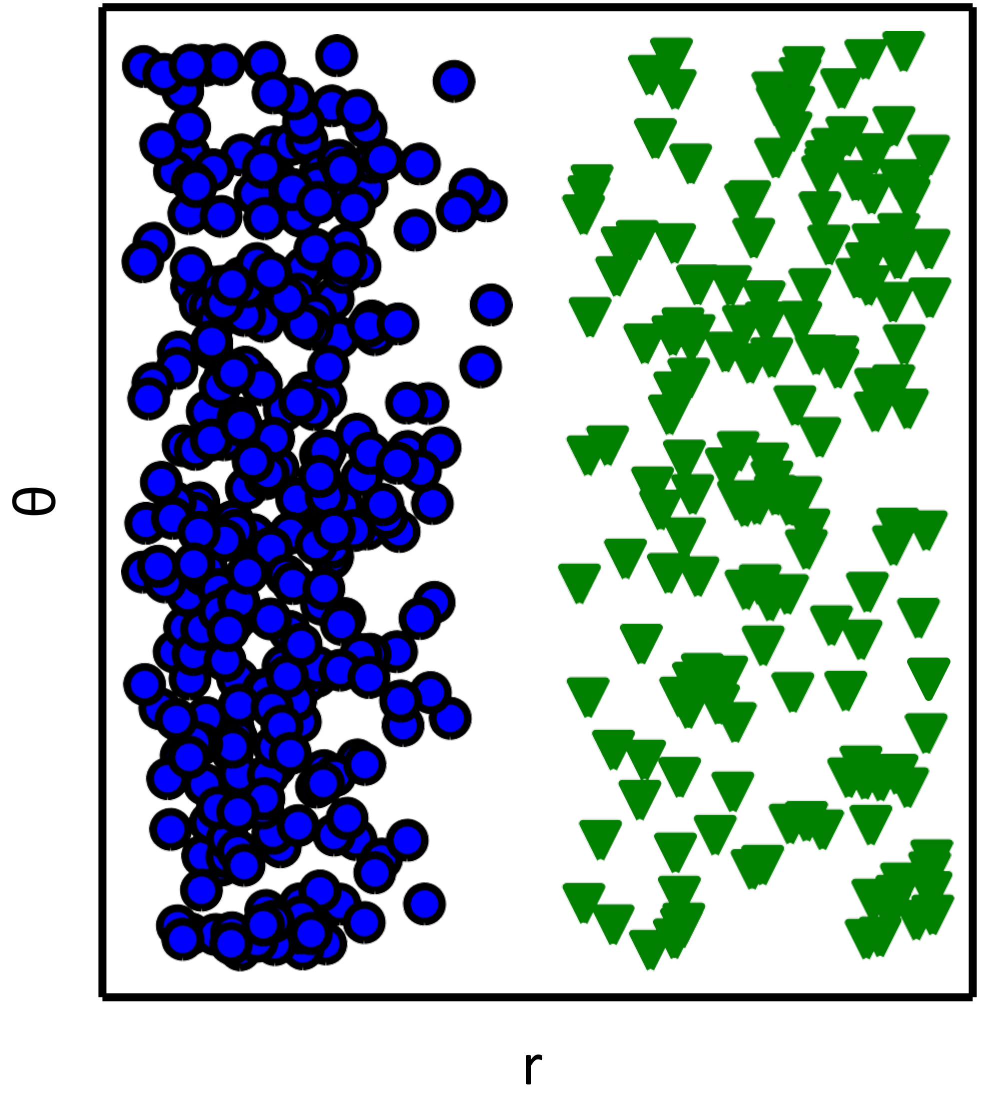
Representation learning

- Handling complex inputs requires representations
- The power of deep learning lies in its ability to learn such representations automatically from data

Cartesian coordinates



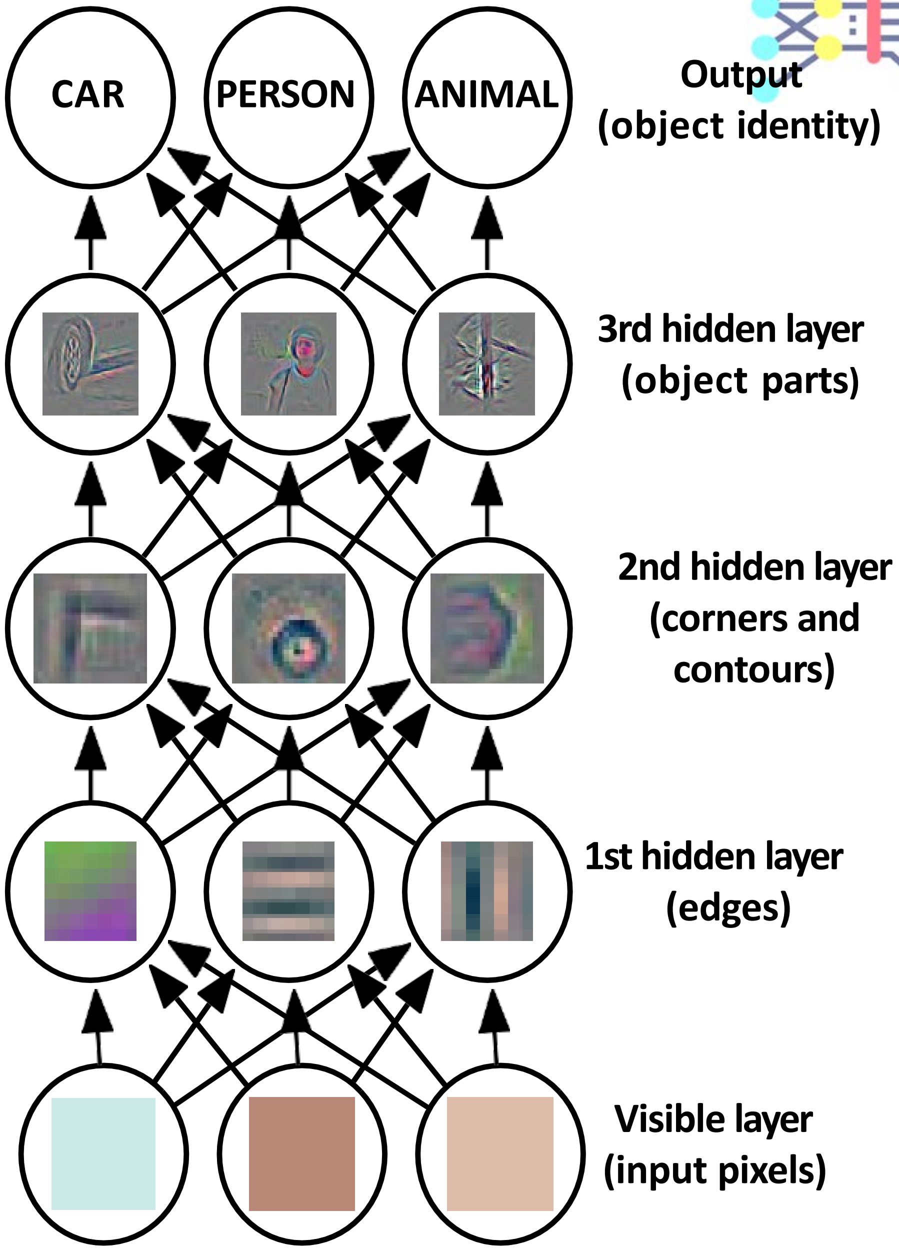
Polar coordinates

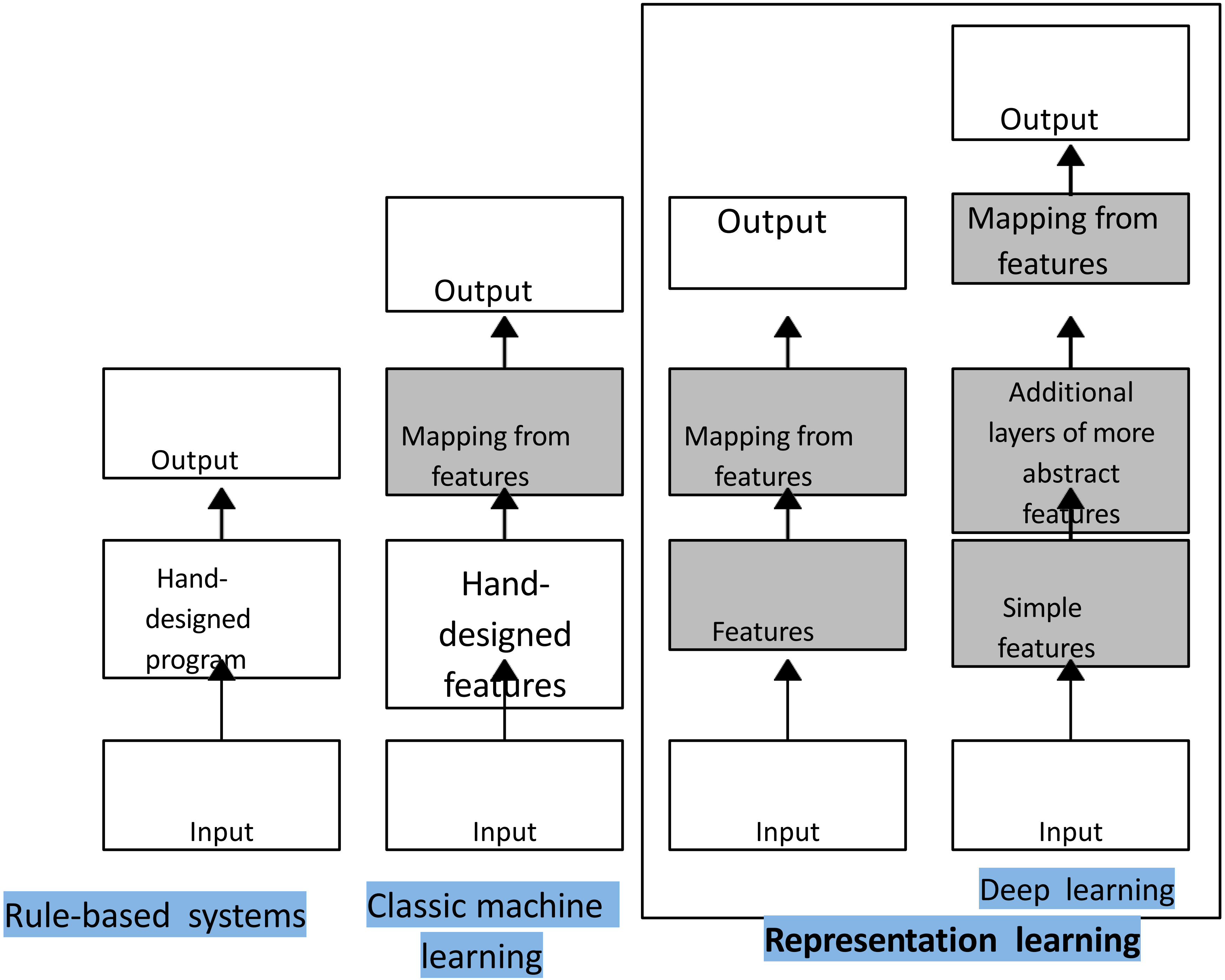


Depth

Deep learning ⇔ Learning Hierarchical Representations

- A pipeline of successive, differentiable modules (transformations)
  - Each module's output is the input for the next module
- Each subsequent module produce higher abstraction features




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

Inspiration for Deep Learning: The Brain!