From Artificial Neural Networks to Deep Learning
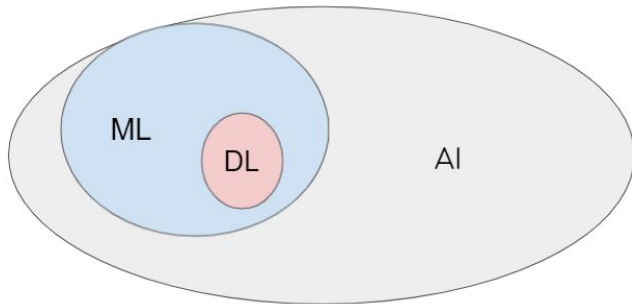
Dr. Konda Reddy Mopuri

kmopuri@iittp.ac.in

Dept. of CSE, IIT Tirupati

Aug-Dec 2021

What is DL?

What is DL?

- Subset of ML that is essentially neural networks with more layers
- Crude attempt to imitate the humam brain in learning

What is DL?

Classical ML vs. DL

- Classical ML: Handcrafted features + learnable model
- Need strong domain expertise

Classical ML vs. DL

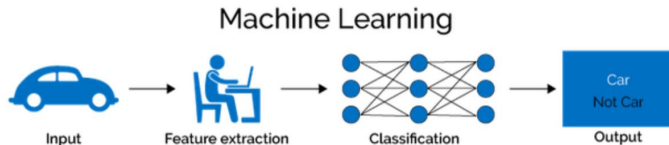- Classical ML: Handcrafted features + learnable model
- Need strong domain expertise

Figure credits: Jay Shaw & Quora

Classical ML vs. DL

- Deep Learning: Deep stack of parameterized processing
- End-to-End learning

Classical ML vs. DL

- Deep Learning: Deep stack of parameterized processing
- End-to-End learning

Input   Feature extraction + Classification   Output
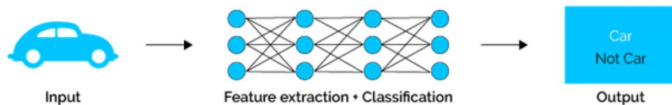

Figure credits: Jay Shaw & Quora

Classical ML vs. DL

- ANNs predate some of the classical ML techniques
- We are now dealing with a new generation ANNs

Neuron

- About 100 billion neurons in human brain

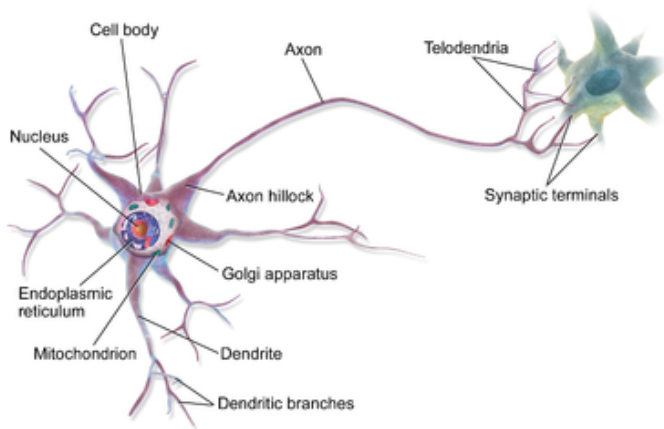Figure credits: Wikipedia

History of Neural Networks

1. McCulloch Pitts neuron (1943) - Threshold Logic Unit

History of Neural Networks

1. McCulloch Pitts neuron (1943) - Threshold Logic Unit
2. Donald Hebb (1949) - Hebbian Learning Principle

History of Neural Networks

1. McCulloch Pitts neuron (1943) - Threshold Logic Unit
2. Donald Hebb (1949) - Hebbian Learning Principle
3. Marvin Minsky (1951) - created the first ANN (Hebbian Learning, 40 neurons)

History of Neural Networks

1. McCulloch Pitts neuron (1943) - Threshold Logic Unit
2. Donald Hebb (1949) - Hebbian Learning Principle
3. Marvin Minsky (1951) - created the first ANN (Hebbian Learning, 40 neurons)
4. Frank Rosenblatt (1958) - created perceptron to classify 20X20 images

History of Neural Networks

1. McCulloch Pitts neuron (1943) - Threshold Logic Unit
2. Donald Hebb (1949) - Hebbian Learning Principle
3. Marvin Minsky (1951) - created the first ANN (Hebbian Learning, 40 neurons)
4. Frank Rosenblatt (1958) - created perceptron to classify 20X20 images
5. David H Hubel and Torsten Wiesel (1959) demonstrated orientation selectivity and columnar organization in cat's visual cortex

Backpropagation

- Paul Werbos (1982) proposed back-propagation for ANNs

History (contd.)

1. Neocognitron -Network for TC problem

History (contd.)

1. Neocognitron -Network for TC problem
2. LeNet family (Lecun et al. 1989)

History (contd.)

1. Neocognitron -Network for TC problem
2. LeNet family (Lecun et al. 1989)
3. AlexNet (Alex krizhevsky et al. 2012)

History (contd.)

1. Neocognitron -Network for TC problem
2. LeNet family (Lecun et al. 1989)
3. AlexNet (Alex krizhevsky et al. 2012)
4. GoogLeNet and ResNet, etc.

History (contd.)

1. Neocognitron -Network for TC problem
2. LeNet family (Lecun et al. 1989)
3. AlexNet (Alex krizhevsky et al. 2012)
4. GoogLeNet and ResNet, etc.
5. Transformers

Deep Learning

1. Natural generalization to ANNs - Doesn't differ much from the 90s NNs

Deep Learning

1. Natural generalization to ANNs - Doesn't differ much from the 90s NNs
2. Computational graph of tensor operations that take advantage of
   - Chain rule (back-propagation)
   - SGD
   - GPUs
   - Huge datasets
   - Convolutions, etc.

Deep Learning

- This generalization enables us to build complex networks that work with Images, text, speech and sequences and train end-to-end

ILSVRC Error
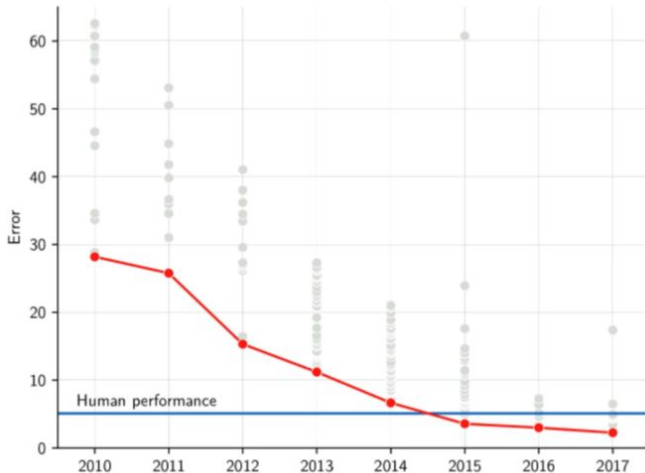
Gershgorn 2017

What makes it work now?

What makes it work now?

1. Huge research and progress in ML

What makes it work now?

1. Huge research and progress in ML
2. Hardware developments - CPUs/GPUs/Storage technologies

What makes it work now?

1. Huge research and progress in ML
2. Hardware developments - CPUs/GPUs/Storage technologies
3. Piles of data over the Internet

What makes it work now?

1. Huge research and progress in ML
2. Hardware developments - CPUs/GPUs/Storage technologies
3. Piles of data over the Internet
4. Collaborative development (open source tools and forums for sharing/discussions, etc)

What makes it work now?

1. Huge research and progress in ML
2. Hardware developments - CPUs/GPUs/Storage technologies
3. Piles of data over the Internet
4. Collaborative development (open source tools and forums for sharing/discussions, etc)
5. Collective efforts from large institutions/corporations

What makes it work now?

1. Huge research and progress in ML
2. Hardware developments - CPUs/GPUs/Storage technologies
3. Piles of data over the Internet
4. Collaborative development (open source tools and forums for sharing/discussions, etc)
5. Collective efforts from large institutions/corporations
6. ...

What makes it work now?

- We have been doing a lot of ML already
  - Taxonomy of ML concepts: Classification, regression, generative models, clustering, etc.
  - Rich statistical formalizations: Bayesian estimation, PAC, etc.
  - Understood fundamentals: Bias-Variance, VC dimension, etc.
  - Good understanding of optimization
  - Efficient large-scale algorithms

Deep Learning - practical perspective

1. Doesn't require a deep mathematical grasp

Deep Learning - practical perspective

1. Doesn't require a deep mathematical grasp
2. Makes the design of large models a system/software development task

Deep Learning - practical perspective

1. Doesn't require a deep mathematical grasp
2. Makes the design of large models a system/software development task
3. Leverages modern hardware

Deep Learning - practical perspective

1. Doesn't require a deep mathematical grasp
2. Makes the design of large models a system/software development task
3. Leverages modern hardware
4. Doesn't seem to plateau with more data

Deep Learning - practical perspective

1. Doesn't require a deep mathematical grasp
2. Makes the design of large models a system/software development task
3. Leverages modern hardware
4. Doesn't seem to plateau with more data
5. Makes the trained models a commodity

Implementation

	Language(s)	License	Main backer
PyTorch	**Python**, C++	BSD	Facebook
TensorFlow	Python, C++	Apache	Google
JAX	Python	Apache	Google
MXNet	Python, C++, R, Scala	Apache	Amazon
CNTK	Python, C++	MIT	Microsoft
Torch	Lua	BSD	Facebook
Theano	Python	BSD	U. of Montreal
Caffe	C++	BSD 2 clauses	U. of CA, Berkeley

We use PyTroch

http://pytorch.org