(An Almost) No Math Introduction to Deep Learning

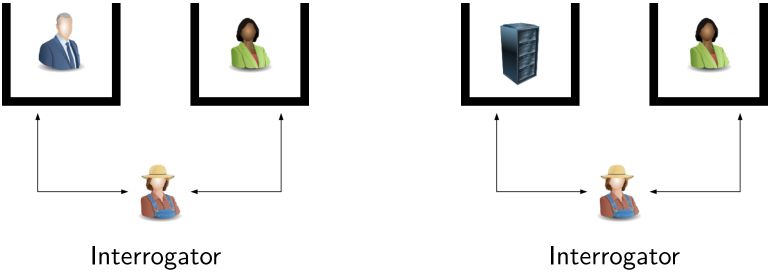
Abhijat Vatsyayan [1]

November 17, 2020

Summary

- Part O
  - Artificial intelligence alphabet soup
- Part I
  - Problem definition (rely on supervised learning)
  - Compute graph and gradients
  - A little about deep learning libraries.
- Part II
  - Need for different architectures
  - Convolution networks
  - Transfer learning, autoencoders
  - Recurrent networks
  - Miscellaneous topics
    - Adversarial attacks
    - GAN
    - Attention
- Not covered

Imitation game



Artificial intelligence

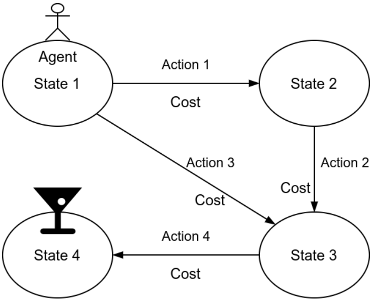
- 1956 - Dartmouth workshop
- Logic machines (50s)
- Knowledge based expert systems (80s)
- Language translation (60s) , 2000s, 2014 and later.
- Machine learning
  - Neural networks including deep learning (started in 1943)
  - Support vector machines
  - Baysian learning
- Graphs
- Genetic algorithms and genetic programs.

Expert systems

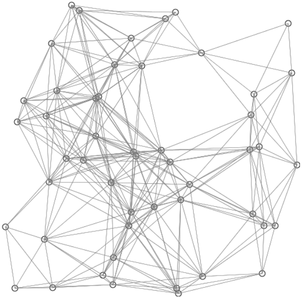
- Database of formally described "facts" or "knowledge".
- A reasoning engine for answering questions or solving problems.

Not to be confused with a true natural language processing and question-answering system.

Search


Agent
State 1
Action 1
Cost
State 2
Action 3
Cost
Action 2
Cost
State 4
Action 4
Cost
State 3

?

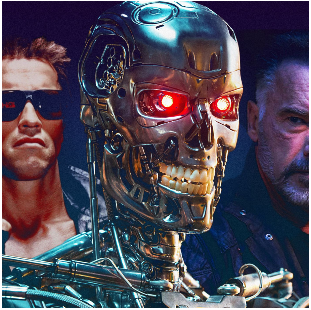


?

Neural Networks

- 1943: Warren McCulloch and Walter Pits connected neurons, computation, logic and learning.
- 1950: Minsky and Dean Edmonds build first neural network computer. 3000 vacuum tubes, surplus auto-pilot parts from B-24 bomber. 40 Neurons.
- 1969: Minsky and Papert publish perceptron - simple linear networks could not learn basic functions.
- 1980s: Jeff Hinton, David Rumelhart, Jeff Hinton and Ronald Williams applied back propagation (again) for training multi-layer neural networks. Rumelhart's work also created the foundations for Recurrent Neural Networks.
- 1990s: LSTM networks by Hochreiter and Schmidhuber 1997. CNN for handwritten digit recognition - Yann LeCun.
- 2000s: LSTMs show promise in speech recognition
- 2012: Deep learning

Different views



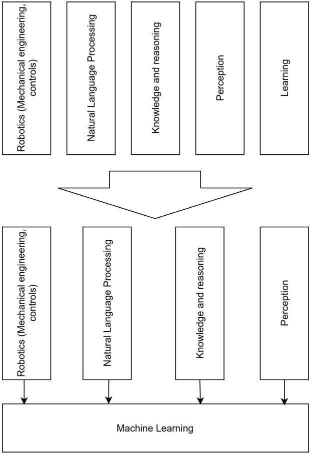
Agent

Tool

General

Narrow

Late 2000s - machine learning dominates



- Image classification, localization and segmentation
- Neural machine translation, question answering, summary.
- Game playing, helicopter flying (stunts)
- Planning, self driving cars
- Text, audio and video processing, generation
- . . .

Machine learning models

Models

- Build a model of the world
- Infer/predict using the model.

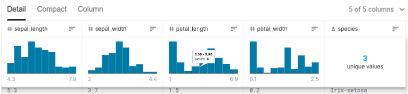
Machine learning

- Supervised learning
- Unsupervised learning
- Reinforcement learning

Supervised learning

Detail Compact Column

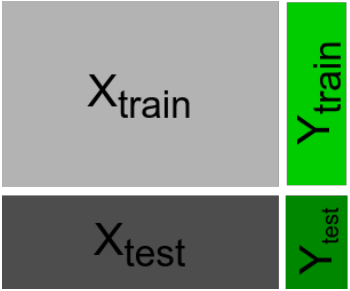
# sepal_length	# sepal_width	# petal_length	# petal_width	species
5.7	2.9	4.2	1.3	Iris-versicolor
6.2	2.9	4.3	1.3	Iris-versicolor
5.1	2.5	3	1.1	Iris-versicolor
5.7	2.8	4.1	1.3	Iris-versicolor
6.3	3.3	6	2.5	Iris-virginica
5.8	2.7	5.1	1.9	Iris-virginica
7.1	3	5.9	2.1	Iris-virginica
6.3	2.9	5.6	1.8	Iris-virginica
6.5	3	5.8	2.2	Iris-virginica



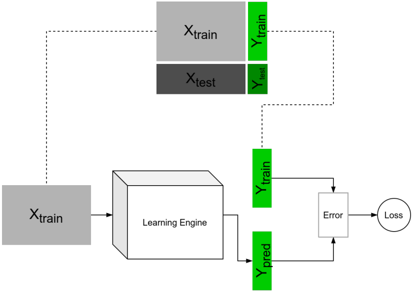
# sepal_length	# sepal_width	# petal_length	# petal_width	species
5.3	3.7	1.5	0.2	Iris-setosa
5	3.3	1.4	0.2	Iris-setosa
7	3.2	4.7	1.4	Iris-versicolor
6.4	3.2	4.5	1.5	Iris-versicolor
6.9	3.1	4.9	1.5	Iris-versicolor

Source: *https://www.kaggle.com/arshid/iris-flower-dataset?select=IRIS.csv*

Supervised learning process flow

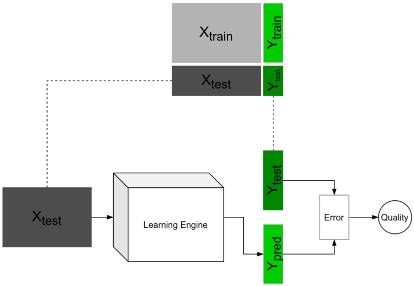


Data

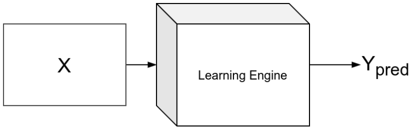


Training

Supervised learning process flow



Testing



Predicting

Machine learning

Unsupervised learning

Reinforcement learning

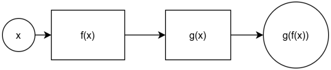
Composition of functions

Consider the following functions

$$f(x) = ax + b$$

$$g(x) = \frac{1}{e^{-x} + 1}$$

$$g(f(x)) = \frac{1}{e^{-(ax+b)} + 1}$$



Composition blocks

Derivatives

$$
\begin{aligned}
f(x) =& ax + b \\
\frac{df(x)}{dx} =& a \\
\sigma(z) =& \frac{1}{e^{-z} + 1} \\
\frac{d\sigma(z)}{dz} =& \sigma(z)(1 - \sigma(z)) \\
\frac{d\sigma(f(x))}{dx} =& ?
\end{aligned}
$$

Derivatives

$$
\begin{aligned}
z =& ax + b \\
\frac{dz}{dx} =& a \\
\sigma(z) =& \frac{1}{e^{-z} + 1} \\
\frac{d\sigma}{dz} =& \sigma(z)(1 - \sigma(z)) \\
\frac{d\sigma}{dx} =& \frac{d\sigma}{dz}\frac{dz}{dx}
\end{aligned}
$$

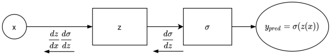
$$z = ax + b \quad (1)$$

$$\frac{dz}{dx} = a \quad (2)$$

$$\sigma(z) = \frac{1}{e^{-z} + 1} \quad (3)$$

$$\frac{d\sigma}{dz} = \sigma(z)(1 - \sigma(z)) \quad (4)$$

$$\frac{d\sigma}{dx} = \frac{d\sigma}{dz}\frac{dz}{dx} = \frac{dz}{dx}\frac{d\sigma}{dz} \quad (5)$$



Back propagation of gradients

References I

V. Dumoulin and F. Visin, "A guide to convolution arithmetic for deep learning," 2018.

Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, pp. 2278–2324, Nov 1998.

I. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," in *International Conference on Learning Representations*, 2015.

A. Kurakin, I. Goodfellow, and S. Bengio, "Adversarial examples in the physical world," *ICLR Workshop*, 2017.

I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. The MIT Press, 2016.

References II

I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Advances in Neural Information Processing Systems 27* (Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, eds.), pp. 3104–3112, Curran Associates, Inc., 2014.

S. J. Russell and P. Norvig, *Artificial Intelligence: a modern approach*. Pearson, 3 ed., 2009.