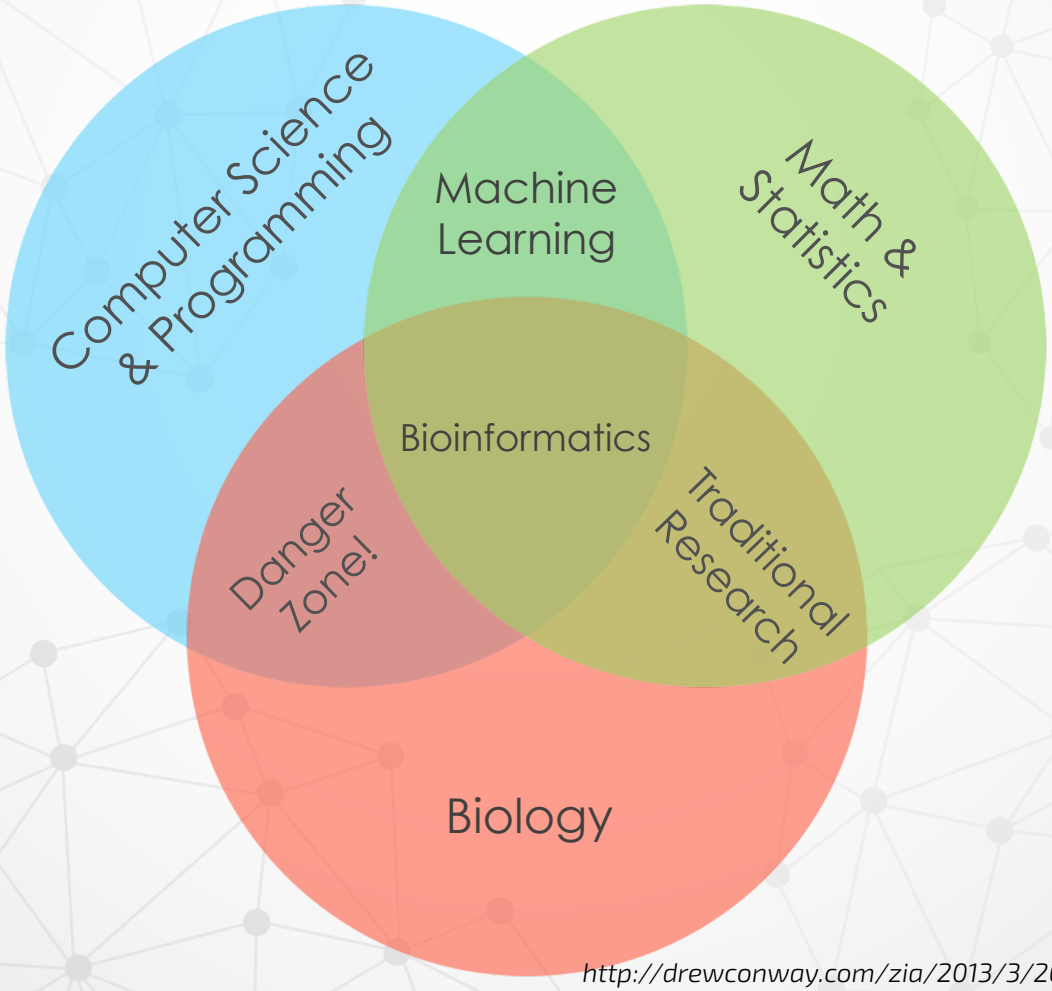
Machine Learning Algorithms

by Eka Antonius Kurniawan

Outline

- Linear Regression with One Variable
- Linear Regression with Multiple Variable
- Collaborative Filtering
- Neural Network

Adjusted Data Science Venn Diagram

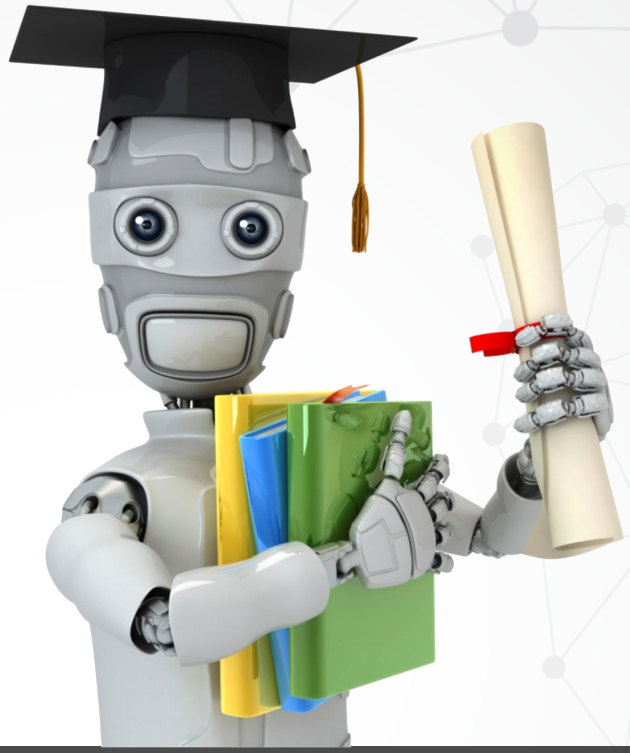


http://drewconway.com/zia/2013/3/26/the-data-science-venn-diagram

What is Machine Learning?

Machine learning is a subfield of **computer science** that evolved from the study of pattern recognition and computational learning theory in **artificial intelligence**. Machine learning explores the construction and study of **algorithms** that can **learn** from and **make predictions** on **data**. Such algorithms operate by **building a model** from example inputs in order to make **data-driven predictions or decisions**, rather than following strictly static program instructions.

https://en.wikipedia.org/wiki/Machine_learning



Machine Learning at Stanford offered by Coursera

Machine learning is the science of getting computers to **act without being explicitly programmed.** In the past decade, machine learning has given us self-driving cars, practical speech recognition, **effective web search**, and a vastly improved **understanding of the human genome**.

https://www.coursera.org/learn/machine-learning

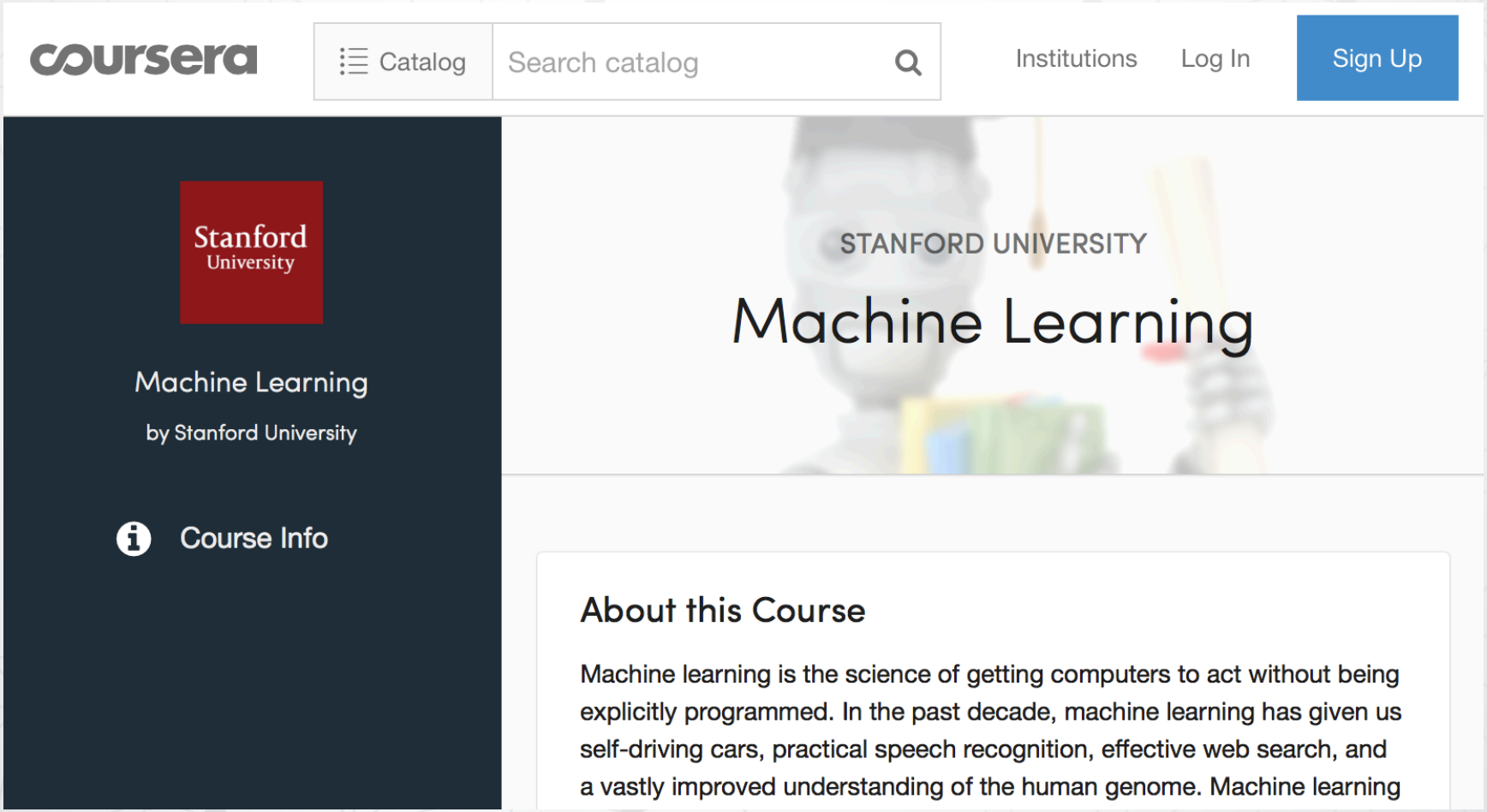
Machine Learning course materials from Stanford University

Lecture Notes

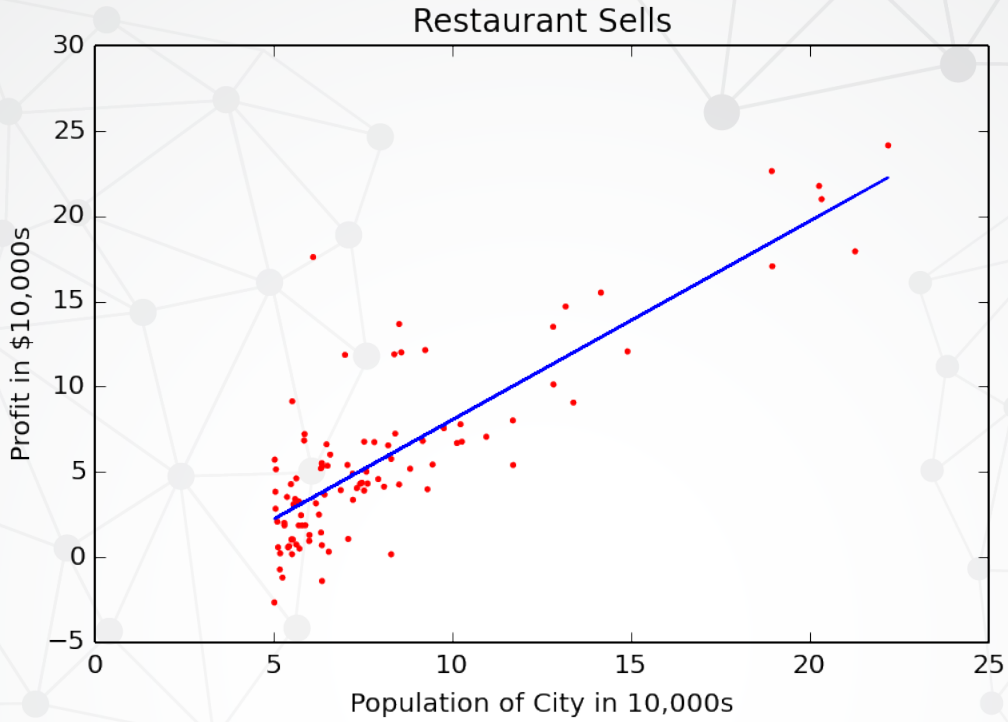
- Lecture notes 1 (ps) (pdf) Supervised Learning, Discriminative Algorithms
- Lecture notes 2 (ps) (pdf) Generative Algorithms
- Lecture notes 3 (ps) (pdf) Support Vector Machines
- Lecture notes 4 (ps) (pdf) Learning Theory
- Lecture notes 5 (ps) (pdf) Regularization and Model Selection
- Lecture notes 6 (ps) (pdf) Online Learning and the Perceptron Algorithm. (optional reading)
- Lecture notes 7a (ps) (pdf) Unsupervised Learning, k-means clustering.
- Lecture notes 7b (ps) (pdf) Mixture of Gaussians
- Lecture notes 8 (ps) (pdf) The EM Algorithm
- Lecture notes 9 (ps) (pdf) Factor Analysis
- Lecture notes 10 (ps) (pdf) Principal Components Analysis
- Lecture notes 11 (ps) (pdf) Independent Components Analysis
- Lecture notes 12 (ps) (pdf) Reinforcement Learning and Control

http://cs229.stanford.edu/materials.html

Machine Learning by Stanford University



https://www.coursera.org/learn/machine-learning

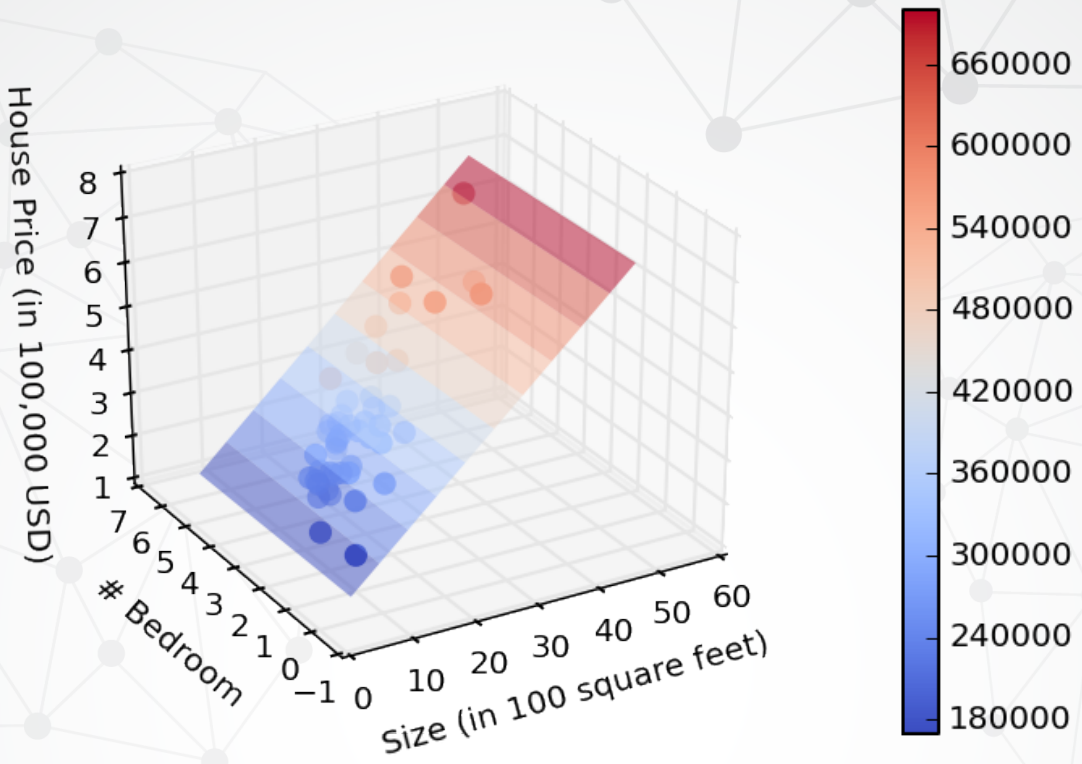


Linear Regression with One Variable

To predict sells revenue

Demo

http://nbviewer.ipython.org/github/ekaakurniawan/iPyMaceLern/blob/master/W1/Linear%20Regression%20With%20One%20Variable.ipynb



Linear Regression with Multiple Variables

To predict housing price

Demo

http://nbviewer.ipython.org/github/ekaakurniawan/iPyMaceLern/blob/master/W2/Linear%20Regression%20With%20Multiple%20Variables.ipynb

Original ratings provided:
Rated 4 for Toy Story (1995)
Rated 3 for Twelve Monkeys (1995)
Rated 5 for Usual Suspects, The (1995)
Rated 4 for Outbreak (1995)
Rated 5 for Shawshank Redemption, The (1994)
Rated 3 for While You Were Sleeping (1995)
Rated 5 for Forrest Gump (1994)
Rated 2 for Silence of the Lambs, The (1991)
Rated 4 for Alien (1979)
Rated 5 for Die Hard 2 (1990)
Rated 5 for Sphere (1998)

Top recommendations for you:
Predicting rating 9.0 for movie Titanic (1997)
Predicting rating 8.9 for movie Star Wars (1977)
Predicting rating 8.8 for movie Shawshank Redemption, The (1994)
Predicting rating 8.5 for movie As Good As It Gets (1997)
Predicting rating 8.5 for movie Good Will Hunting (1997)
Predicting rating 8.5 for movie Usual Suspects, The (1995)
Predicting rating 8.5 for movie Schindler's List (1993)
Predicting rating 8.4 for movie Raiders of the Lost Ark (1981)
Predicting rating 8.4 for movie Empire Strikes Back, The (1980)
Predicting rating 8.4 for movie Braveheart (1995)

Collaborative Filtering

To recommend movie

Andrew Y. Ng, 2015. ***Machine Learning****. Week 9: Recommender Systems. Coursera. https://www.coursera.org/learn/machine-learning*

Demo

http://nbviewer.ipython.org/github/ekaakurniawan/iPyMaceLern/blob/master/W9/Collaborative%20Filtering.ipynb

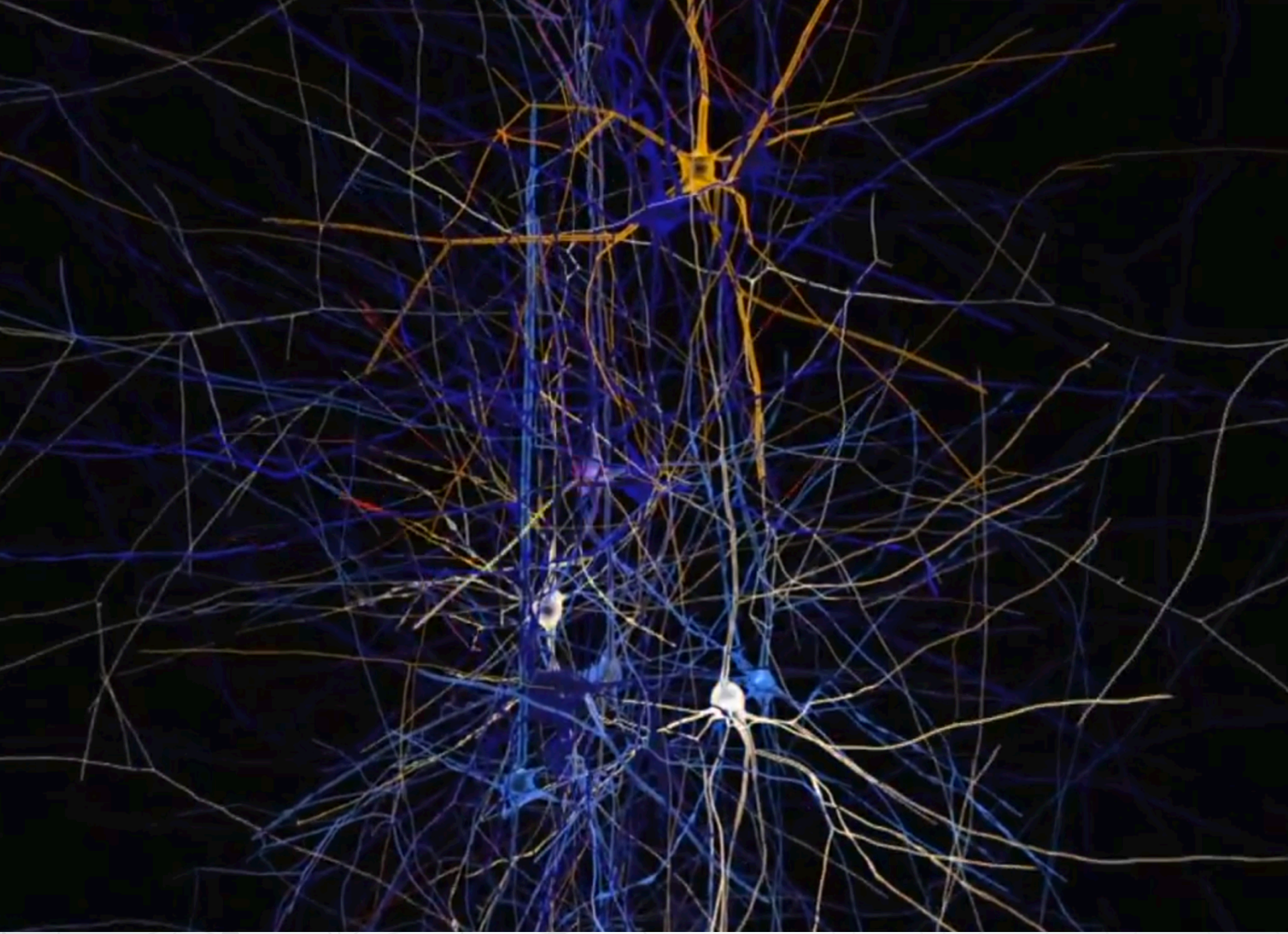
Neural Network

Inspired by Brain

Some of the representations are inspired by advances in **neuroscience** and are loosely based on interpretation of **information processing and communication patterns** in a **nervous system**, such as neural coding which attempts to define a relationship between the stimulus and the neuronal responses and the relationship among the electrical activity of the neurons in the brain.

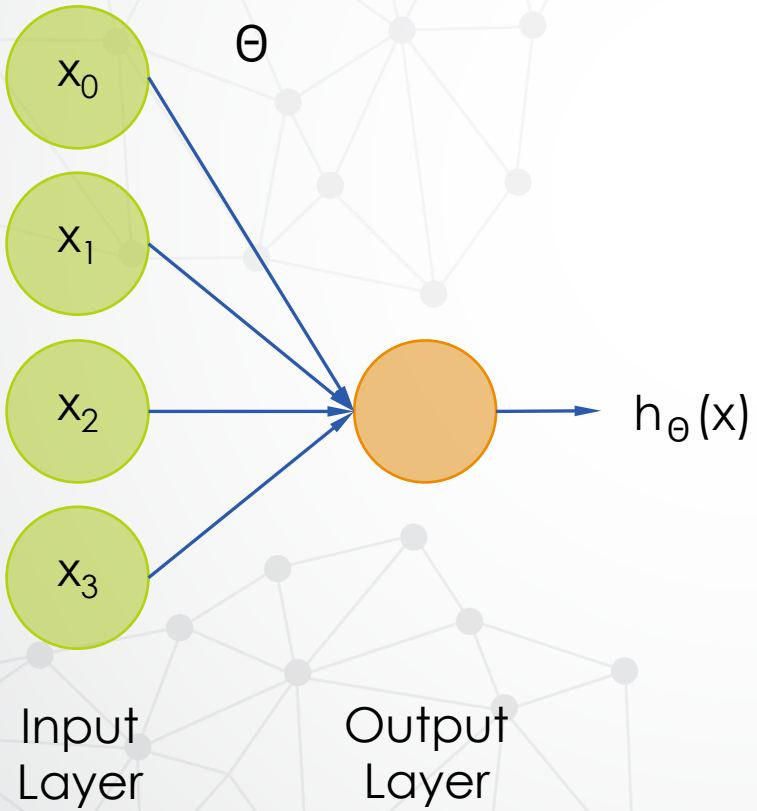
https://en.wikipedia.org/wiki/Deep_learning


THE BRAIN IS THE MOST IMPORTANT ORGAN
ACCORDING TO THE BRAIN
quickmeme.com



https://www.youtube.com/watch?v=plEn-sU2CVo

Neuron Model: Logistic Unit

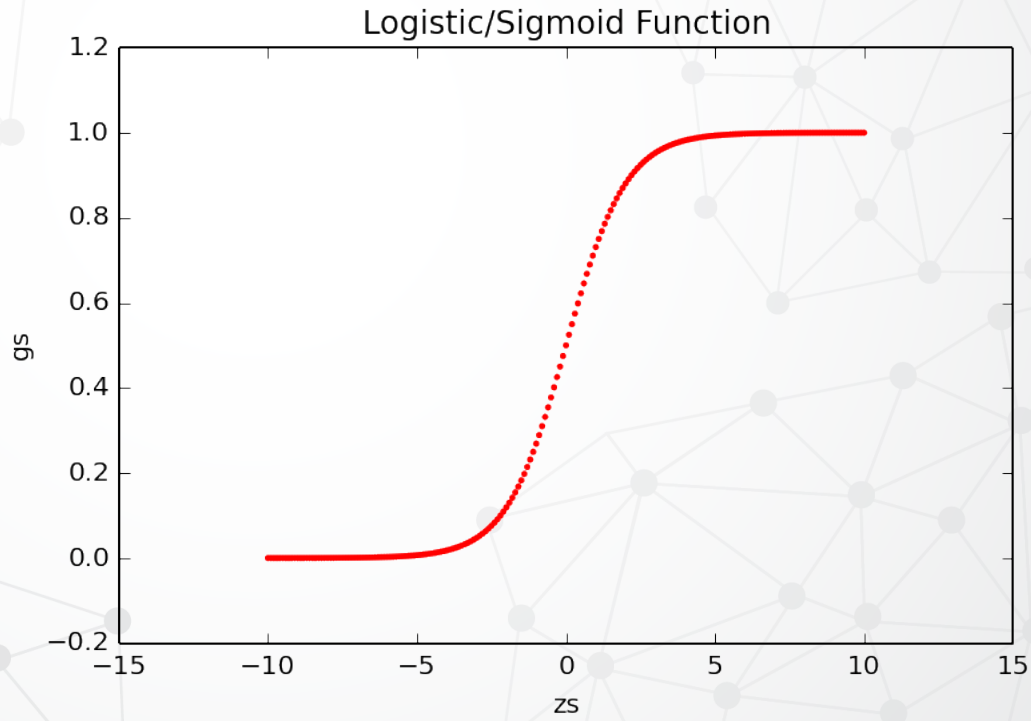


$$h_\Theta(x) = \frac{1}{1 + e^{-\Theta^T x}}$$

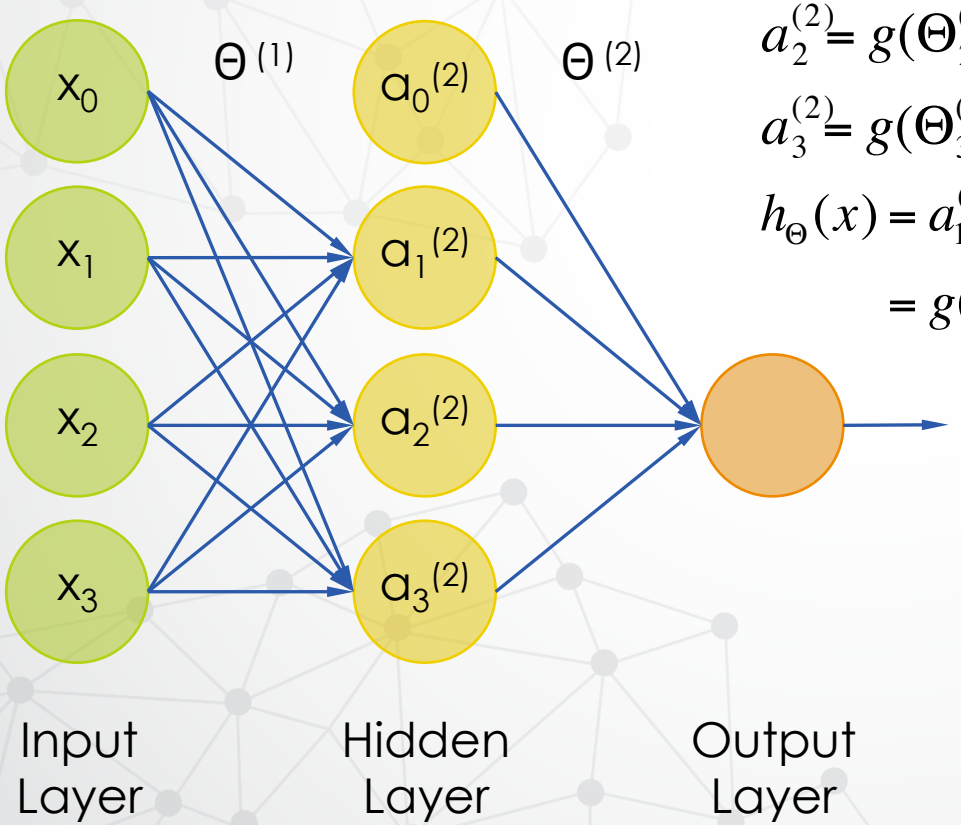
$$x = \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \end{bmatrix} \quad \Theta = \begin{bmatrix} \Theta_0 \\ \Theta_1 \\ \Theta_2 \\ \Theta_3 \end{bmatrix}$$

Sigmoid/Logistic Activation Function

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



Neural Network Model (feedforward)



$$a_1^{(2)} = g(\Theta_{10}^{(1)} x_0 + \Theta_{11}^{(1)} x_1 + \Theta_{12}^{(1)} x_2 + \Theta_{13}^{(1)} x_3)$$

$$a_2^{(2)} = g(\Theta_{20}^{(1)} x_0 + \Theta_{21}^{(1)} x_1 + \Theta_{22}^{(1)} x_2 + \Theta_{23}^{(1)} x_3)$$

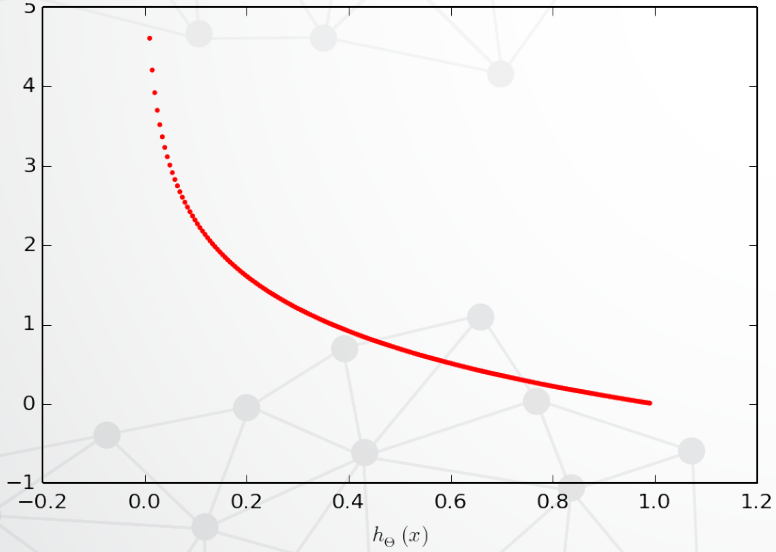
$$a_3^{(2)} = g(\Theta_{30}^{(1)} x_0 + \Theta_{31}^{(1)} x_1 + \Theta_{32}^{(1)} x_2 + \Theta_{33}^{(1)} x_3)$$

$$h_\Theta(x) = a_1^{(3)}$$

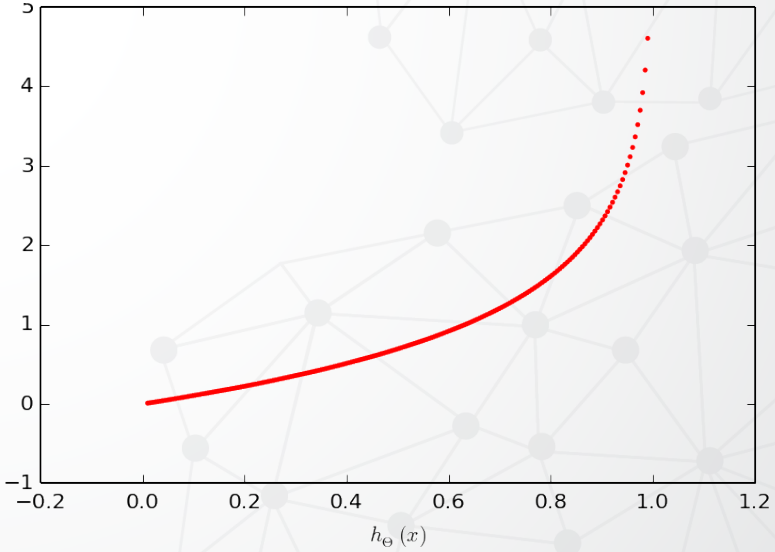
$$= g(\Theta_{10}^{(2)} a_0^{(2)} + \Theta_{11}^{(2)} a_1^{(2)} + \Theta_{12}^{(2)} a_2^{(2)} + \Theta_{13}^{(2)} a_3^{(2)})$$

Cost Function Intuitive

$$Cost(h_\Theta(x), y) = \begin{cases} -\log(h_\Theta(x)) & \text{if y = 1} \\ -\log(1 - h_\Theta(x)) & \text{if y = 0} \end{cases}$$



if y = 1



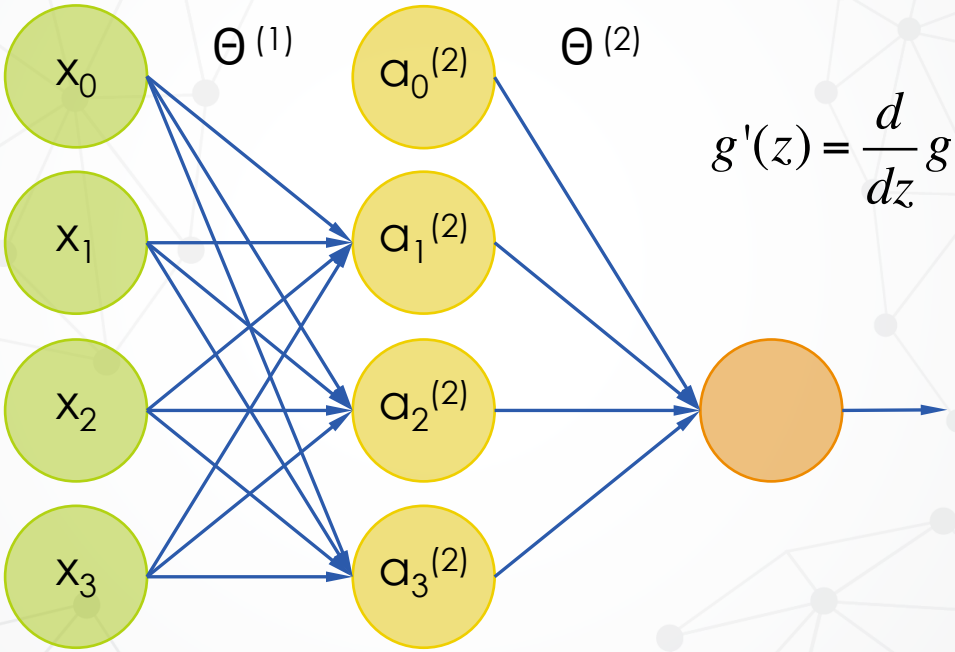
if y = 0

Cost Function

$$J(\Theta) = \frac{1}{m}\sum_{i=1}^{m}\sum_{k=1}^{K}\left[-y_k^{(i)}\log((h_\Theta(x^{(i)}))_k) - (1-y_k^{(i)})\log(1-(h_\Theta(x^{(i)}))_k)\right]$$

$$+\frac{\lambda}{2m}\left[\sum_{j=1}^{L^{hidden}}\sum_{k=1}^{L^{input}}(\Theta_{j,k}^{(1)})^2 + \sum_{j=1}^{L^{output}}\sum_{k=1}^{L^{hidden}}(\Theta_{j,k}^{(2)})^2 + \right]$$

Gradient Function (backpropagation algorithm)



$$g'(z) = \frac{d}{dz} g(z) = g(z)(1 - g(z))$$

$$\delta_k^{(3)} = a_k^{(3)} - y_k$$

$$\delta^{(2)} = (\Theta^{(2)})^T \delta^{(3)} \times g'(z^{(2)})$$

Neural Network

To recognize handwriting

Andrew Y. Ng, 2015. ***Machine Learning****. Week 5: Neural Networks: Learning. Coursera. https://www.coursera.org/learn/machine-learning*

Demo

http://nbviewer.ipython.org/github/ekaakurniawan/iPyMaceLern/blob/master/W4_5/Neural%20Networks.ipynb

Q&A