From Regression to Deep Learning

Practice **LESS** Deep Learning

Learn - **E**xperiment - **S**hare - **S**eek

Barathi Ganesh HB

Centre for Excellence in Computational Engineering and Networking (CEN)
Amrita School of Engineering, Coimbatore
Amrita Vishwa Vidyapeetham, India
email: barathiganesh.hb@gmail.com

Outline

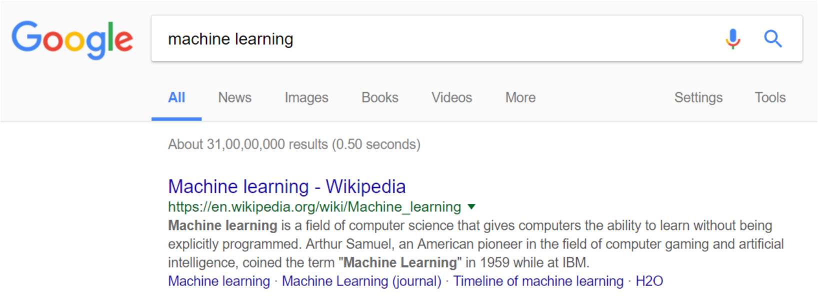
ML Introduction

Regression to Deep Learning

Need of Deep Learning

Matrix Representation

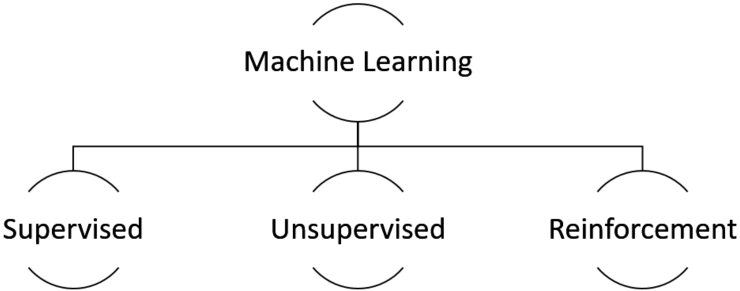
Machine Learning Introduction


Google
machine learning
All
News
Images
Books
Videos
More
Settings
Tools
About 31,00,00,000 results (0.50 seconds)
Machine learning - Wikipedia
https://en.wikipedia.org/wiki/Machine_learning
Machine learning is a field of computer science that gives computers the ability to learn without being explicitly programmed. Arthur Samuel, an American pioneer in the field of computer gaming and artificial intelligence, coined the term "Machine Learning" in 1959 while at IBM.
Machine learning · Machine Learning (journal) · Timeline of machine learning · H2O

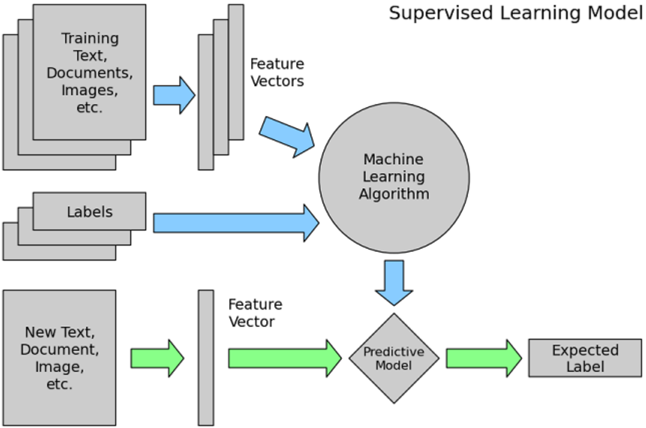
- Reducing human/machine efforts required to perform a task (time optimization).
- Increasing the performance of a task (efficiency optimization).

Steps in Machine Learning

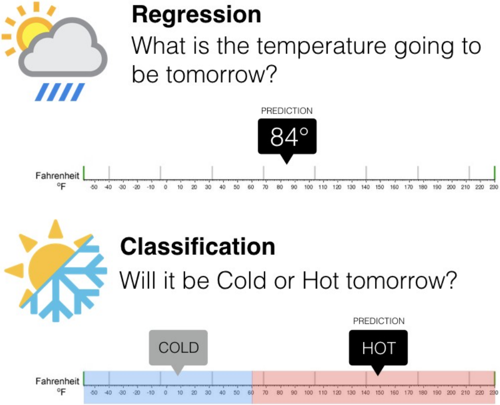
- Collecting data
- Preparing the data
- Training a model
- Evaluating the model
- Improving the performance


Machine Learning
Supervised
Unsupervised
Reinforcement

Supervised Learning



source: www.allprogrammingtutorials.com/tutorials/introduction-to-machine-learning.php

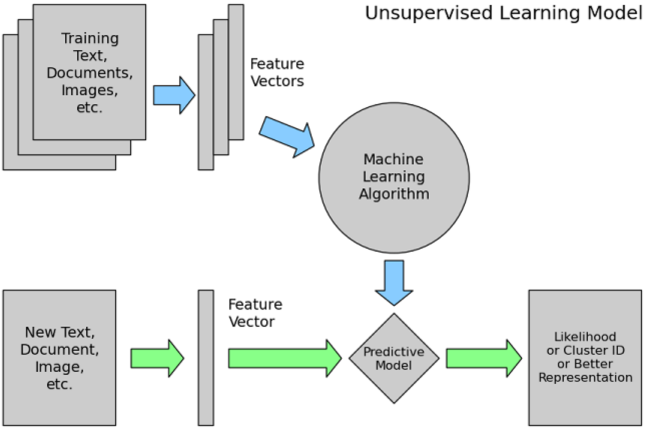


source: `https://medium.com/@ali_88273/regression-vs-classification-87c224350d69`

Common Supervised Learning Algorithms

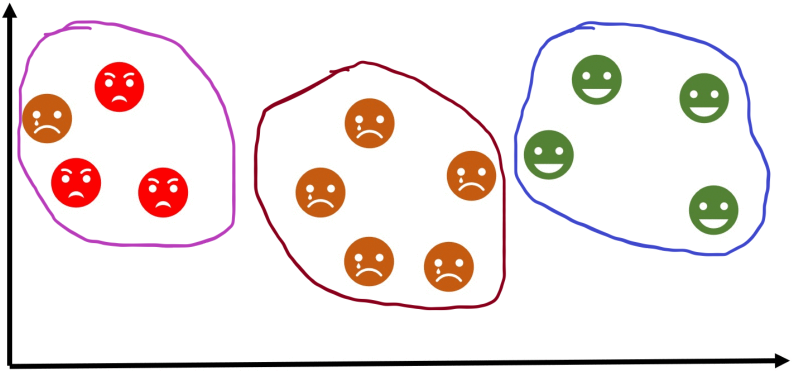
- Linear Regression
- Logistic Regression
- Support Vector Machines
- Support Vector Regression
- Decision Trees
- Random Forest Tree
- Naive Bayes

Unsupervised Learning



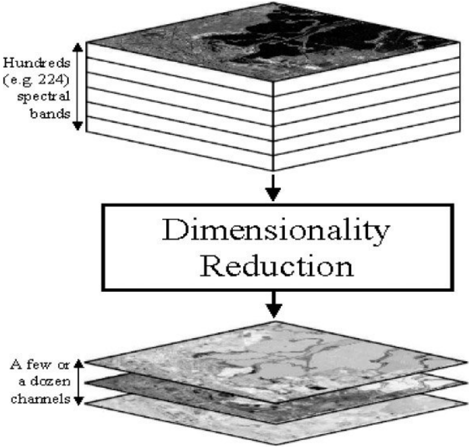
source: `www.allprogrammingtutorials.com/tutorials/introduction-to-machine-learning.php`

Clustering



source: https://towardsdatascience.com/clustering-unsupervised-learning-788b215b074b

Dimensionality Reduction



source: http://spie.org/newsroom/3560-dimensionality-reduction-of-multidimensional-satellite-imagery?SSO=1

Common Unsupervised Learning Algorithms

- K-means
- Affinity Propagation
- Singluar Value Decomposition
- Non-negative matrix factorization

You have already learned the Machine Learning. When?

?

You have already learned the Machine Learning. When?

$$2x = 6 \tag{1}$$

$$(2x - 6) = 0 \tag{2}$$

$$x = ? \tag{3}$$

You have already learned the Machine Learning. When?

$$2x = 6 \tag{4}$$

$$(2x - 6) = 0 \tag{5}$$

$$x = ? \tag{6}$$

$$x = 6/2 = 3 \tag{7}$$

$$2(3) - 6 = 0 \tag{8}$$

You have already learned the Machine Learning. When?

$$2a + b + c = 4 \tag{9}$$

$$a + 3b + 2c = 5 \tag{10}$$

$$a = 6 \tag{11}$$

You have already learned the Machine Learning. When?

$$\begin{bmatrix} 2 & 1 & 1 \\ 1 & 3 & 2 \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} 4 \\ 5 \\ 6 \end{bmatrix} \tag{12}$$

You have already learned the Machine Learning. When?

$$A = \begin{bmatrix} 2 & 1 & 1 \\ 1 & 3 & 2 \\ 1 & 0 & 0 \end{bmatrix}, x = \begin{bmatrix} a \\ b \\ c \end{bmatrix}, b = \begin{bmatrix} 4 \\ 5 \\ 6 \end{bmatrix} \tag{13}$$

You have already learned the Machine Learning. When?

$$\begin{bmatrix} 2 & 1 & 1 \\ 1 & 3 & 2 \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} 4 \\ 5 \\ 6 \end{bmatrix} \tag{14}$$

$$Ax = b \tag{15}$$

$$(Ax - b) = ? \tag{16}$$

You have already learned the Machine Learning. When?

$$\begin{bmatrix} 2 & 1 & 1 \\ 1 & 3 & 2 \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} 4 \\ 5 \\ 6 \end{bmatrix} \tag{17}$$

$$Ax = b \tag{18}$$

$$(Ax - b) = 0 \tag{19}$$

$$x = \begin{bmatrix} a \\ b \\ c \end{bmatrix} = ? \tag{20}$$

What is Regression

Regression?

What is Regression

$$x + y = z$$

Solving Ax=b

$$\begin{bmatrix} 2 & 1 & 1 \\ 1 & 3 & 2 \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} 4 \\ 5 \\ 6 \end{bmatrix} \tag{21}$$

$$X \ W = Y \tag{22}$$

$$(X \ W - Y) = 0 \tag{23}$$

$$W = \begin{bmatrix} a \\ b \\ c \end{bmatrix} = ? \tag{24}$$

$$X^{\dagger} \ X \ W = X^{\dagger} \ Y \tag{25}$$

$$I \ W = X^{\dagger} \ Y \tag{26}$$

$$W = X^{\dagger} \ Y \tag{27}$$

Decimal Value Prediction

ID	digit1	digit2	digit3	value
1	0	0	0	**0**
2	0	0	1	**1**
3	0	1	0	**2**
4	0	1	1	**3**
5	1	0	0	**4**
6	1	0	1	**5**
7	1	1	0	**6**
8	1	1	1	**7**

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} w1 \\ w2 \\ w3 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \end{bmatrix} \tag{28}$$

$$X \ \mathbf{w} = \mathbf{y} \tag{29}$$

$$X^{\dagger} \ X \ \mathbf{w} = X^{\dagger} \ \mathbf{y} \tag{30}$$

$$\mathbf{w} = X^{\dagger} \ \mathbf{y} \tag{31}$$

$$\mathbf{w} = X^{\dagger}\,\mathbf{y} = X^{\dagger}\begin{bmatrix}0\\1\\2\\3\\4\\5\\6\\7\end{bmatrix} = \begin{bmatrix}4\\2\\1\end{bmatrix} = \begin{bmatrix}w1\\w2\\w3\end{bmatrix} \tag{32}$$

$$X \ \mathbf{w} = \mathbf{y} \tag{33}$$

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} 4 \\ 2 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \end{bmatrix} \tag{34}$$

$$X \ \mathbf{w} = \mathbf{y}^{pre} \tag{35}$$

$$training \ \ error = abs(\mathbf{y} - \mathbf{y}^{pre}) \tag{36}$$

$$training\ \ error = sum(abs(\mathbf{y} - \mathbf{y}^{pre})) \tag{37}$$

$$\mathbf{y} - \mathbf{y}^{pre} = \begin{bmatrix}0\\1\\2\\3\\4\\5\\6\\7\end{bmatrix} - \begin{bmatrix}0\\1\\2\\3\\4\\5\\6\\7\end{bmatrix} = sum\begin{bmatrix}0\\0\\0\\0\\0\\0\\0\\0\end{bmatrix} = 0 \tag{38}$$

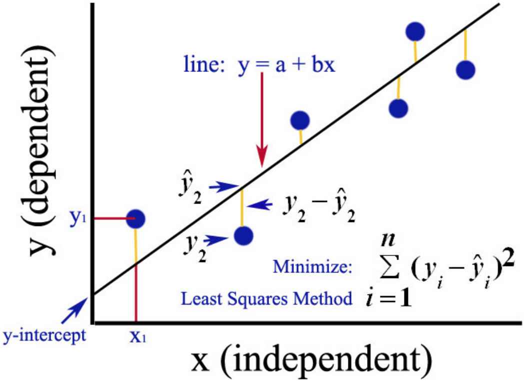
$$\begin{bmatrix} digit1 & digit2 & digit1 \end{bmatrix} \begin{bmatrix} w1 \\ w2 \\ w3 \end{bmatrix} = [value] \tag{39}$$

$$\mathbf{digit1 * w1 + digit2 * w2 + digit3 * w3 = value} \tag{40}$$

$$\begin{bmatrix} digit1 & digit2 & digit1 \end{bmatrix} \begin{bmatrix} 4 \\ 2 \\ 1 \end{bmatrix} = [value] \tag{41}$$

$$\mathbf{digit1 * 4 + digit2 * 2 + digit3 * 1 = value} \tag{42}$$

Linear Regression



source: `solutions4statistics.com`

Decimal Value Prediction

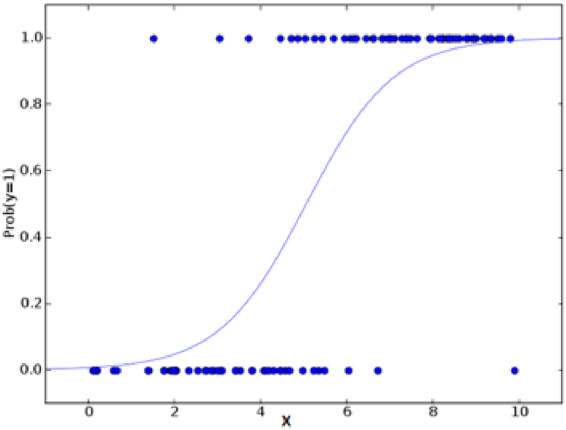
ID	digit1	digit2	digit3	value	decision
1	0	0	0	**0**	**0**
2	0	0	1	**1**	**0**
3	0	1	0	**2**	**0**
4	0	1	1	**3**	**0**
5	1	0	0	**4**	**1**
6	1	0	1	**5**	**1**
7	1	1	0	**6**	**1**
8	1	1	1	**7**	**1**

$$\begin{bmatrix} digit1 & digit2 & digit3 \end{bmatrix} \begin{bmatrix} w1 \\ w2 \\ w3 \end{bmatrix} = [value] \tag{43}$$

$$\mathbf{digit1 * w1 + digit2 * w2 + digit3 * w3 = value} \tag{44}$$

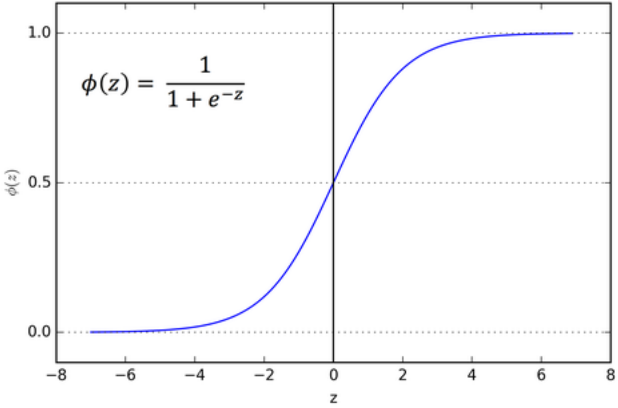
$$Prediction = \begin{cases} 1 & if \;\; 4 \geqslant value \\ 0 & \quad\quad else \end{cases} \tag{45}$$

Logistic Regression



source: solutions4statistics.com

Logistic - Sigmoid Function



https://sebastianraschka.com/images/faq/logisticregr-neuralnet/sigmoid.png

Logistic - Sigmoid

$$\Phi(z) = \frac{1}{1 + exp^{-z}} \tag{46}$$

$$\Phi(-6) = \frac{1}{1 + exp^{-(-6)}} = \frac{1}{1 + 403.42} = 0.0024 \tag{47}$$

$$\Phi(0) = \frac{1}{1 + exp^{0}} = \frac{1}{1 + 1} = 0.5 \tag{48}$$

$$\Phi(6) = \frac{1}{1 + exp^{-(6)}} = \frac{1}{1 + 0.0024} = 0.997 \tag{49}$$

Logistic Regression

$$\Phi(z) = \frac{1}{1 + exp^{-z}} \tag{50}$$

$$\mathbf{y} = \Phi(X\ \mathbf{w}) = \frac{1}{1 + exp^{-(X\ \mathbf{w})}} \tag{51}$$

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} w1 \\ w2 \\ w3 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \tag{52}$$

$$X \ \mathbf{w} = \mathbf{y} \tag{53}$$

$$X^{\dagger} \ X \ \mathbf{w} = X^{\dagger} \ \mathbf{y} \tag{54}$$

$$\mathbf{w} = X^{\dagger} \ \mathbf{y} \tag{55}$$

$$\mathbf{w} = X^{\dagger}\ \mathbf{y} = X^{\dagger} \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 1.24054754 \\ -0.11269202 \\ -0.11269202 \end{bmatrix} = \begin{bmatrix} w1 \\ w2 \\ w3 \end{bmatrix} \tag{56}$$

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} 1.24054754 \\ -0.11269202 \\ -0.11269202 \end{bmatrix} = sigmoid\left(\begin{bmatrix} 0.0 \\ -0.11269202 \\ -0.11269202 \\ -0.22538404 \\ 1.24054754 \\ 1.12785552 \\ 1.12785552 \\ 1.0151635 \end{bmatrix} \right) \tag{57}$$

$$X\ \mathbf{w} = sigmoid(\mathbf{y}^{pre}) \tag{58}$$

$$sigmoid\left(\begin{bmatrix} 0.0 \\ -0.11269202 \\ -0.11269202 \\ -0.22538404 \\ 1.24054754 \\ 1.12785552 \\ 1.12785552 \\ 1.0151635 \end{bmatrix}\right) = \begin{bmatrix} 0.5 \\ 0.47185 \\ 0.47185 \\ 0.44389 \\ 0.77565 \\ 0.75544 \\ 0.75544 \\ 0.73402 \end{bmatrix} \tag{59}$$

$$X\ \mathbf{w} = sigmoid(\mathbf{y}^{pre}) \tag{60}$$

$$training\ \ error = sum(abs(\mathbf{y} - sigmoid(\mathbf{y}^{pre}))) \tag{61}$$

$$training\ \ error = sum(abs(\mathbf{y} - sigmoid(\mathbf{y}^{pre}))) \tag{62}$$

$$abs(\mathbf{y} - sigmoid(\mathbf{y}^{pre}) = \begin{bmatrix}0\\0\\0\\0\\1\\1\\1\\1\end{bmatrix} - \begin{bmatrix}0\\0\\0\\0\\1\\1\\1\\1\end{bmatrix} = \begin{bmatrix}0\\0\\0\\0\\0\\0\\0\\0\end{bmatrix} \tag{63}$$

$$training\ \ error = 0 \tag{64}$$

$$\begin{bmatrix} 0.5 \\ 0.47185 \\ 0.47185 \\ 0.44389 \\ 0.77565 \\ 0.75544 \\ 0.75544 \\ 0.73402 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \tag{65}$$

$$Prediction = \begin{cases} 1 & if \;\; sigmoid(\mathbf{y}^{pre}) \geqslant 0.5 \\ 0 & else \end{cases} \tag{66}$$

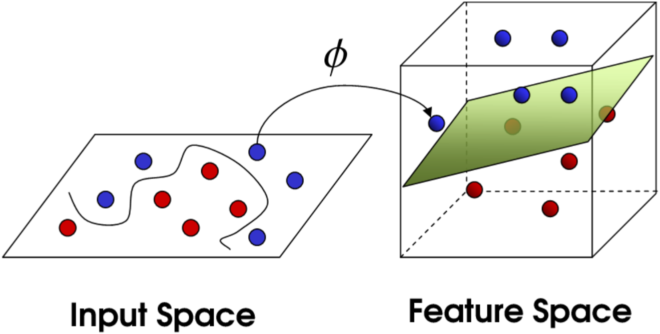
$$\begin{bmatrix} digit1 & digit2 & digit3 \end{bmatrix} \begin{bmatrix} 1.24054754 \\ -0.11269202 \\ -0.11269202 \end{bmatrix} = [value] \tag{67}$$

$$\mathbf{digit1 * w1 + digit2 * w2 + digit3 * w3 = value} \tag{68}$$

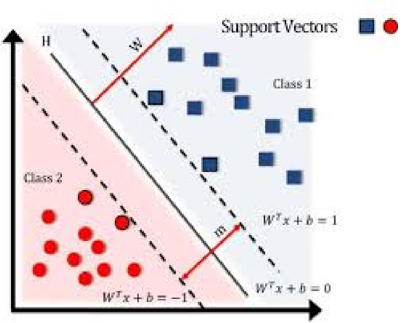
$$sigmoid(\mathbf{y}^{pre}) = \frac{\mathbf{1}}{\mathbf{1 + exp^{-(digit1 * w1 + digit2 * w2 + digit3 * w3)}}} \tag{69}$$

$$Prediction = \begin{cases} 1 & if \;\; sigmoid(\mathbf{y}^{pre}) \geqslant 0.5 \\ 0 & else \end{cases} \tag{70}$$

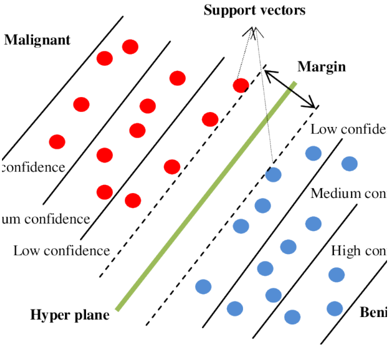
Support Vector Machine



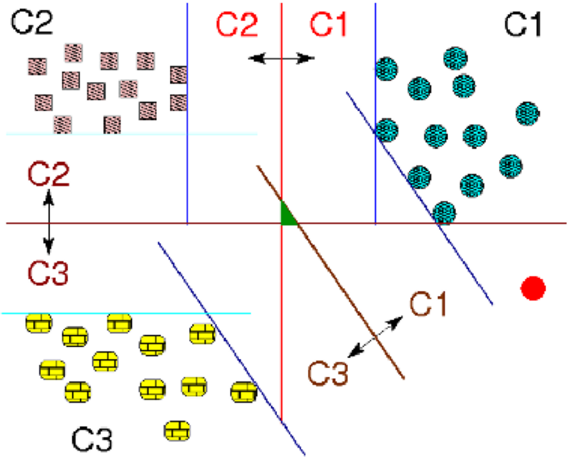
source: `https://www.linkedin.com/pulse/support-vector-machine-srinivas-kulkarni/`



source: `https://www.researchgate.net/figure/Figure-3-SVM-classification-scheme-H-is-the-classification-hyperplane-W-is-the-normal_286268965`

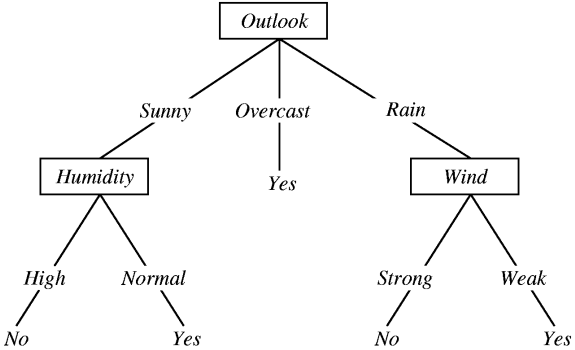


source: https://www.researchgate.net/figure/267510750_Figure-4-An-example-of-support-vector-machine-with-confident-level-of-tumour-label-assign



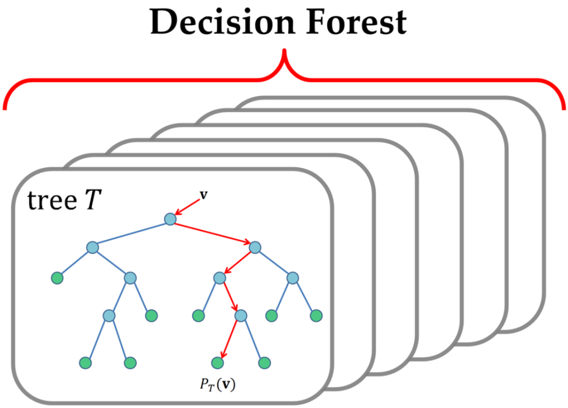
source: https://www.researchgate.net/figure/Figure-2-Illustration-of-the-SVM-principle-and-of-the-one-versus-one-multiclass-classification_220098164_fig2

Decision Trees



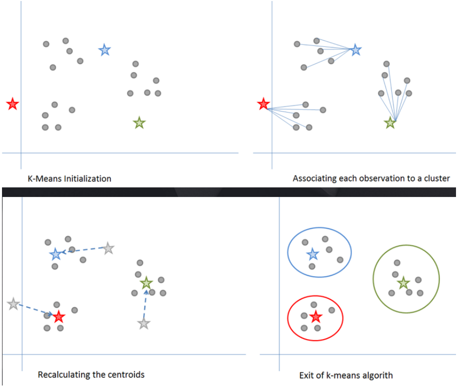
source: https://dzone.com/articles/machine-learning-with-decision-trees

Random Forest Tree



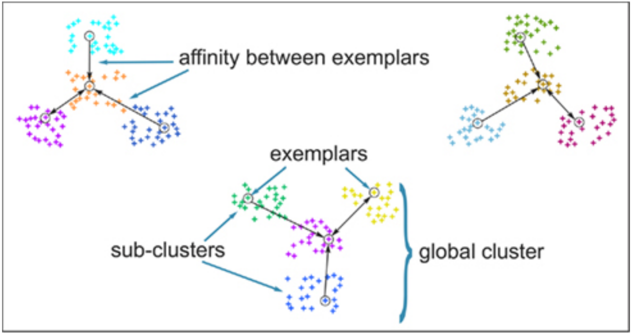
source: https://dimensionless.in/introduction-to-random-forest/

K-Means



source: `https://www.re-work.co/blog/top-10-machine-learning-algorithms-for-beginners`

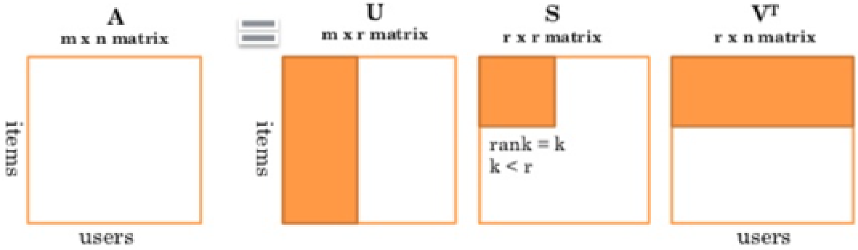
Affinity Propagation



source: http://madhukaudantha.blogspot.in/2015/04/affinitypropagation-clustering-algorithm.html

Singluar Value Decomposition

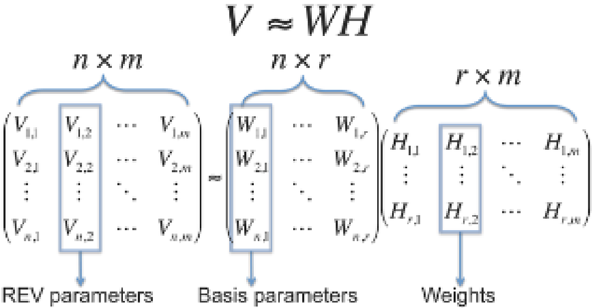
$$A = U \times S \times V^T$$



$$A_k = U_k \times S_k \times V_k^T$$

source: http://xieyan87.com/2015/06/stochastic-gradient-descent-sgd-singular-value-decomposition-svd-algorithms-notes/

Non-negative matrix factorization



source: `http://journals.plos.org/plosone/article?id=10.1371/journal.pone.0028898`

Evaluating the model

- Accuracy

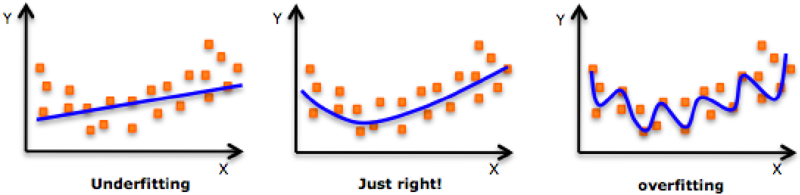
$$Accuracy = \frac{\#\ correctly\ classified\ instances}{total\#\ instances} \tag{71}$$

Evaluating the model

$$\begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 1 \\ 0 \end{bmatrix} == \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} False \\ True \\ True \\ True \\ True \\ True \\ True \\ False \end{bmatrix} \tag{72}$$

Accuracy = 6 / 8 * 100 = 75 %

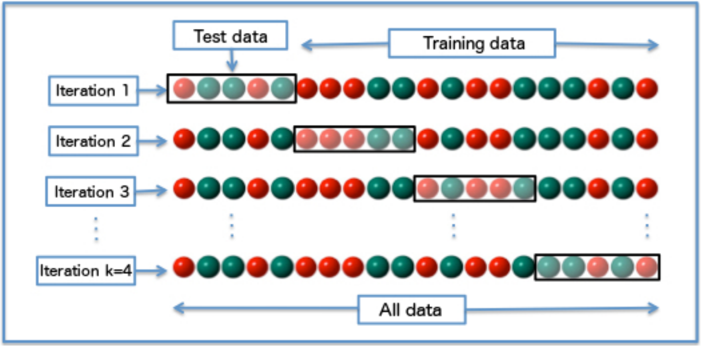
Evaluating the model



Source: `www.quora.com/Whats-the-difference-between-overfitting-and-underfitting`

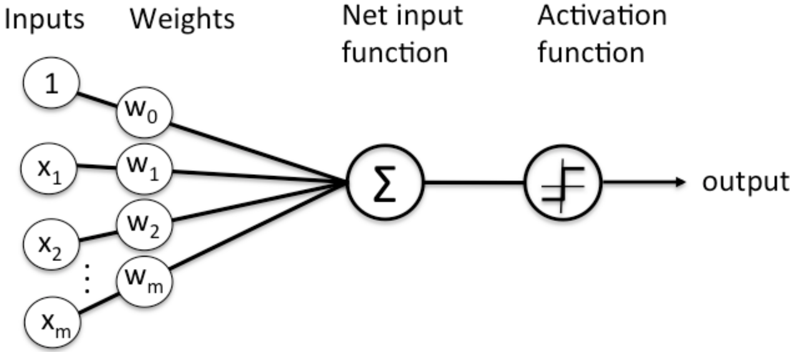
Improving the performance

10 - fold 10-cross validation



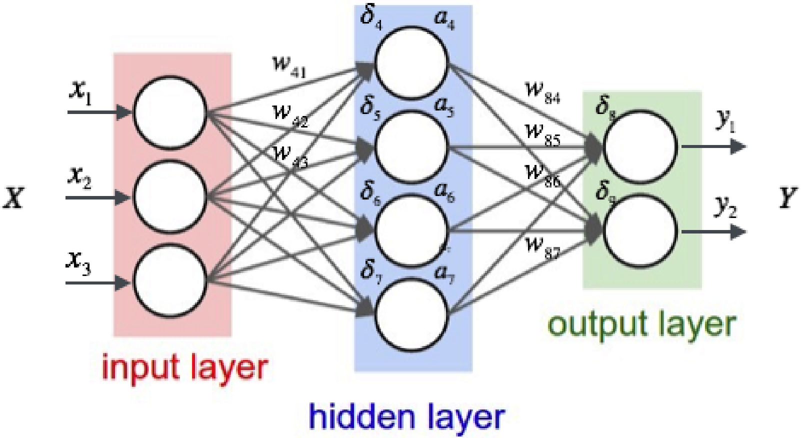
Source: wikipedia

Logistic Regression as a Neuron



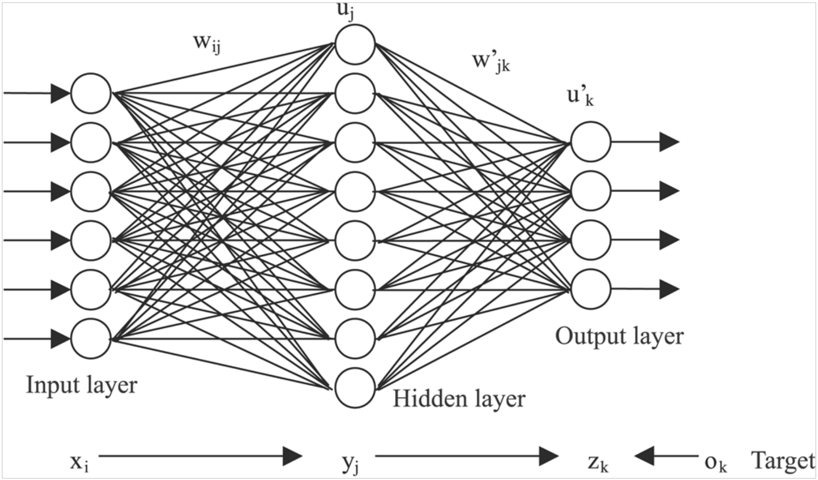
www.techmaru.com/technology/artificial-neural-networks/neural-network-elements

Neuron to Neurons



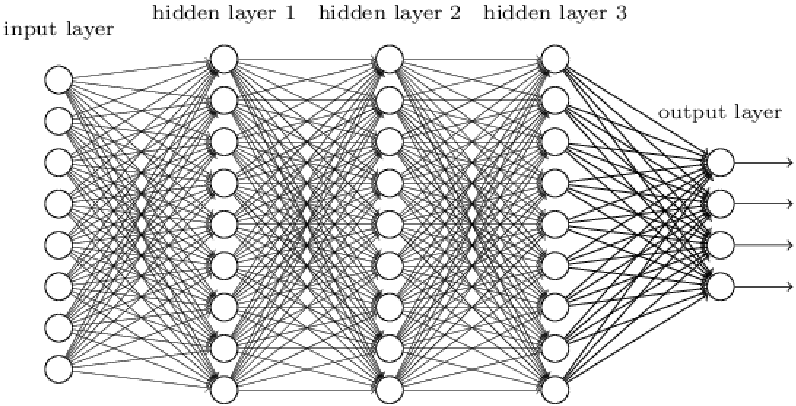
medium.com/@curiousily/tensorflow-for-hackers-part-iv-neural-network-from-scratch-1a4f504dfa8

Single Layer Network



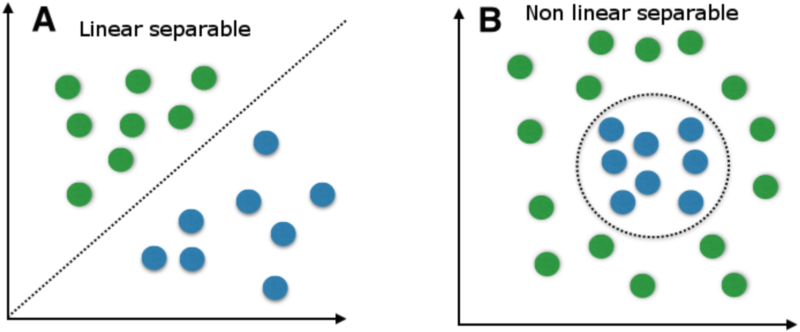
www.extremetech.com/extreme/215170-artificial-neural-networks-are-changing-the-world-what-are-they

Multi Layer Network



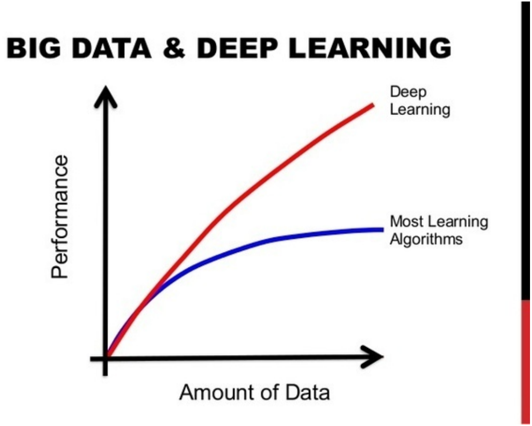
in.mathworks.com/matlabcentral/fileexchange/64247-simple-neural-network

Why Deep Learning?



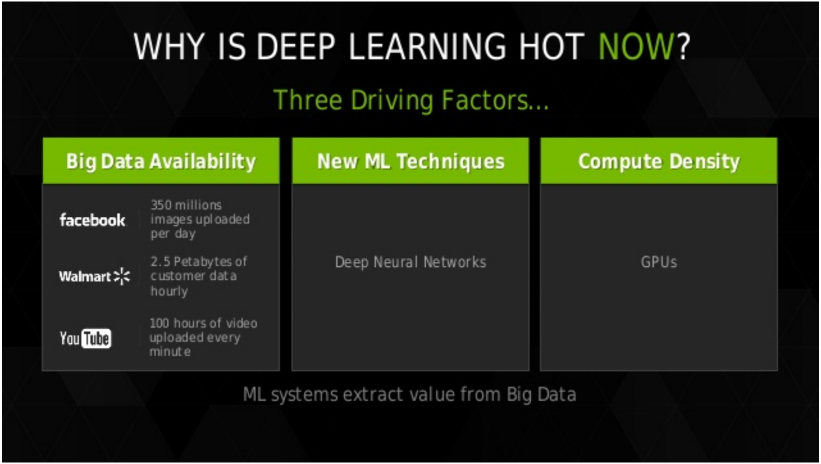
Source: https://leonardoaraujosantos.gitbooks.io/artificial-inteligence/content/linear_classification.html

Why Deep Learning?



Source: `https://qph.ec.quoracdn.net/main-qimg-bf69c291005e68620a1bef39ae8f029e-c`

Why now Deep Learning?

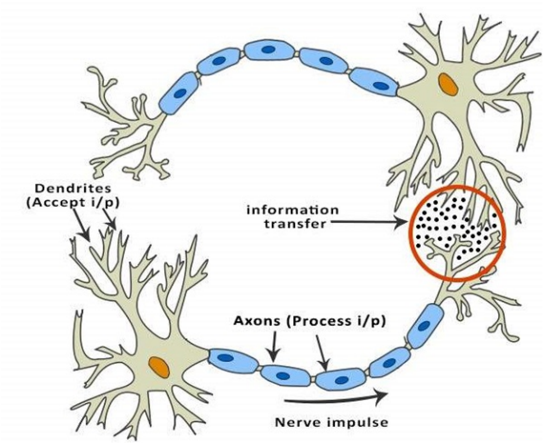


https://www.slideshare.net/DataScienceMD/deep-learning-with-gpus

Common Deep Learning Algorithms

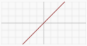
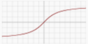
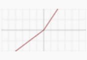
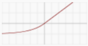
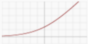
- Convolutional Neural Network
- Recurrent Neural Network
- Long-Short Term Memory Network
- Deep Neural Network
- Auto Encoders

Neuron



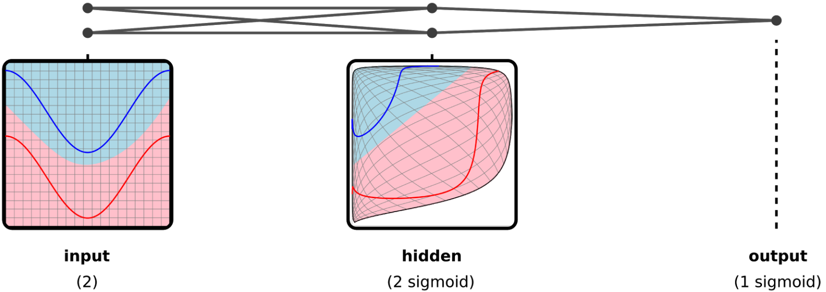
www.learnopencv.com

Activation Functions

Name	Plot	Equation	Derivative
Identity		$f(x) = x$	$f'(x) = 1$
Binary step		$f(x) = \begin{cases} 0 & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$	$f'(x) = \begin{cases} 0 & \text{for } x \neq 0 \\ ? & \text{for } x = 0 \end{cases}$
Logistic (a.k.a Soft step)		$f(x) = \frac{1}{1+e^{-x}}$	$f'(x) = f(x)(1 - f(x))$
TanH		$f(x) = \tanh(x) = \frac{2}{1+e^{-2x}} - 1$	$f'(x) = 1 - f(x)^2$
ArcTan		$f(x) = \tan^{-1}(x)$	$f'(x) = \frac{1}{x^2+1}$
Rectified Linear Unit (ReLU)		$f(x) = \begin{cases} 0 & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$	$f'(x) = \begin{cases} 0 & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$
Parameteric Rectified Linear Unit (PReLU) [2]		$f(x) = \begin{cases} \alpha x & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$	$f'(x) = \begin{cases} \alpha & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$
Exponential Linear Unit (ELU) [3]		$f(x) = \begin{cases} \alpha(e^x - 1) & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$	$f'(x) = \begin{cases} f(x) + \alpha & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$
SoftPlus		$f(x) = \log_e(1 + e^x)$	$f'(x) = \frac{1}{1+e^{-x}}$

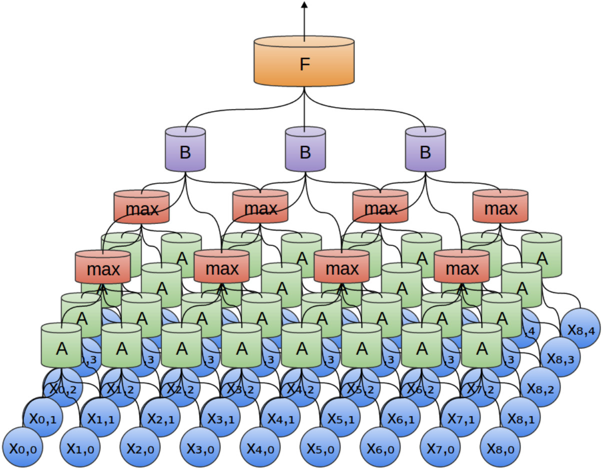
`cdn-images-1.medium.com/`

Logistic Regression as Neuron



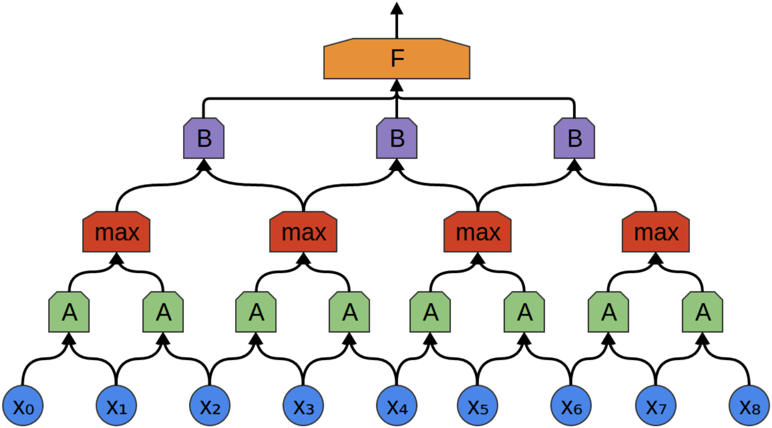
colah.github.io

Convolutional Neural Network



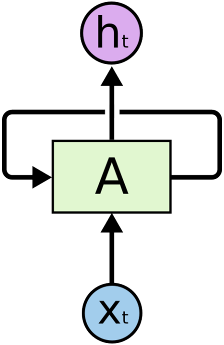
colah.github.io

Convolutional Neural Network



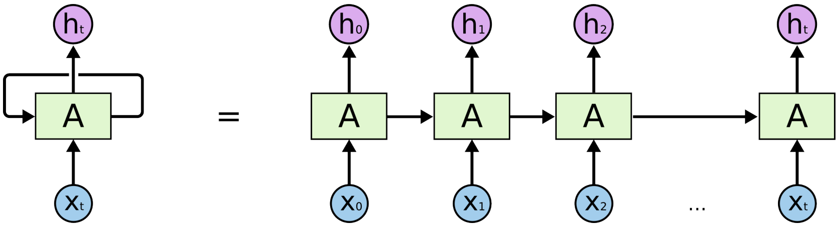
colah.github.io

Recurrent Neural Network



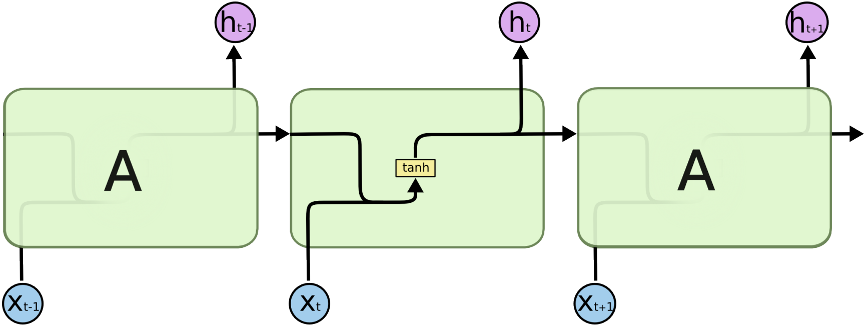
colah.github.io

Recurrent Neural Network



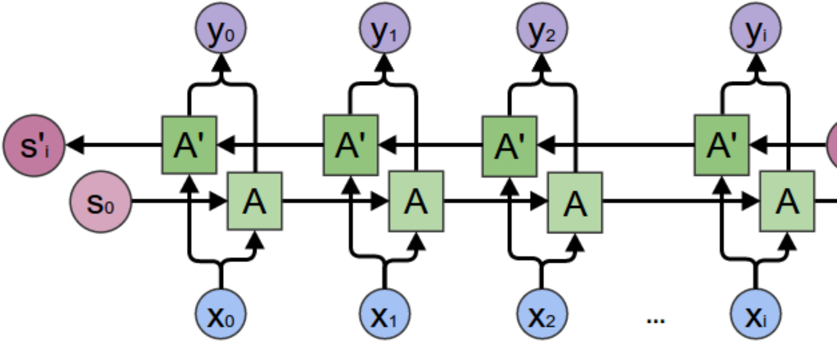
colah.github.io

Recurrent Neural Network



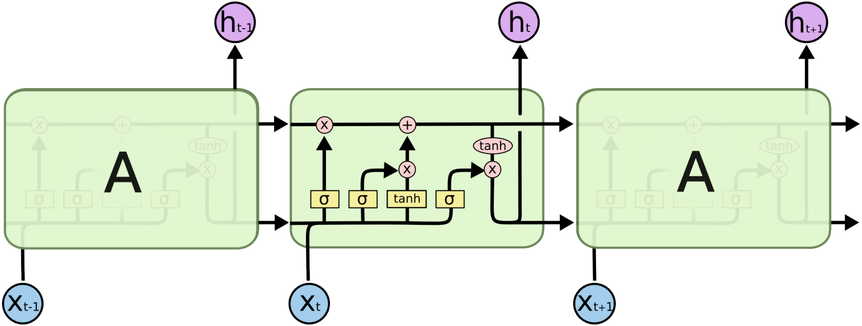
colah.github.io

Bi-directional Recurrent Neural Network



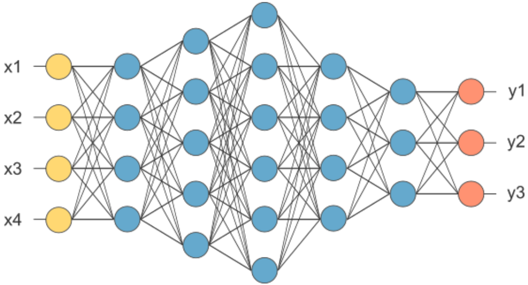
colah.github.io

Long-Short Term Memory Network



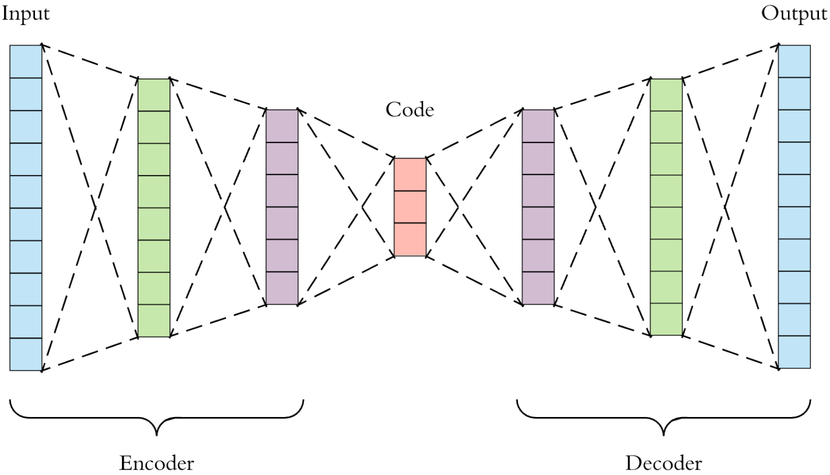
colah.github.io

Deep Neural Network



qph.fs.quoracdn.net

Auto Encoder



cdn-images-1.medium.com

Matrix Representation

Linear Equations to Matrix

$$2a + b + c = 4 \tag{73}$$

$$a + 3b + 2c = 5 \tag{74}$$

$$a = 6 \tag{75}$$

Linear Equations to Matrix

$$\begin{bmatrix} 2 & 1 & 1 \\ 1 & 3 & 2 \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} 4 \\ 5 \\ 6 \end{bmatrix} \tag{76}$$

Linear Equations to Matrix

$$A = \begin{bmatrix} 2 & 1 & 1 \\ 1 & 3 & 2 \\ 1 & 0 & 0 \end{bmatrix}, \mathbf{x} = \begin{bmatrix} a \\ b \\ c \end{bmatrix}, \mathbf{b} = \begin{bmatrix} 4 \\ 5 \\ 6 \end{bmatrix} \tag{77}$$

Term - Document Matrix

S1 : we are in cen

S2 : cen is in amrita

Vocabulary : $\{amrita, are, cen, in, is, we\}$

	amrita	**are**	**cen**	**in**	**is**	**we**
S1	0	1	1	1	0	1
S2	1	0	1	1	1	0

Term Frequency and Inverse Document Frequency

S1 : we are in cen

S2 : cen is in amrita

Vocabulary : $\{amrita, are, cen, in, is, we\}$

	amrita	are	cen	in	is	we
S1	0/1	1/1	1/2	1/2	0/1	1/1
S2	1/1	0/1	1/2	1/2	1/1	0/1

	amrita	are	cen	in	is	we
S1	0	1	0.5	0.5	0	1
S2	1	0	0.5	0.5	1	0

N-grams in BOT

- uni-gram : $\{amrita,\ are,\ cen,\ in,\ is,\ we\}$
- bi-gram : uni-gram $\cup$ $\{we\ are,\ are\ in,\ in\ cen\}$
- tri-gram : uni-gram $\cup$ bi-gram $\cup$ $\{we\ are\ in, are\ in\ cen\}$
- n-gram : uni-gram $\cup$ . . . $\cup$ n-gram

Feature - Term Matrix : Co-occurrence Matrix

	amrita	are	cen	in	is	we
amrita	0	0	0	1	0	0
are	0	0	0	1	0	1
cen	0	0	0	0	1	0
in	1	0	0	0	1	0
is	0	0	1	1	0	0
we	0	1	0	0	0	0

Symmetric - Counts in both side of the terms (bi-directional)
Asymmetric - Counts in one side of the terms (uni-directional)
Window Size - Reflects the context : Number of words to be included for count

Distributional Representation

	amrita	are	cen	in	is	we
S1	0	1	1	1	0	1
S2	1	0	1	1	1	0

$$A = \begin{bmatrix} 0 & 1 & 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 & 1 & 0 \end{bmatrix}$$

$$U\Sigma V^T = svd(A)$$

$$WH^T = nmf(AA^T) \ for \ document \ representation$$

$$WH^T = nmf(A^T A) \ for \ word \ representation$$

Co-occurrence Matrix for Documents Representation (U)

$$AA^T = \begin{bmatrix} 0 & 1 & 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 & 1 & 0 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ 1 & 0 \\ 1 & 1 \\ 1 & 1 \\ 0 & 1 \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} S_1 S_1 & S_1 S_2 \\ S_2 S_1 & S_2 S_2 \end{bmatrix}$$

Co-occurrence Matrix for Words Representation (V)

$$A^T A = \begin{bmatrix} 0 & 1 \\ 1 & 0 \\ 1 & 1 \\ 1 & 1 \\ 0 & 1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} 0 & 1 & 1 & 1 & 0 & 1 \\ 1 & 0 & 1 & 1 & 1 & 0 \end{bmatrix}$$

$$= \begin{bmatrix} W_1 W_1 & W_1 W_2 & W_1 W_3 & W_1 W_4 & W_1 W_5 & W_1 W_6 \\ W_2 W_1 & W_2 W_2 & W_2 W_3 & W_2 W_4 & W_2 W_5 & W_2 W_6 \\ W_3 W_1 & W_3 W_2 & W_3 W_3 & W_3 W_4 & W_3 W_5 & W_3 W_6 \\ W_4 W_1 & W_4 W_2 & W_4 W_3 & W_4 W_4 & W_4 W_5 & W_4 W_6 \\ W_5 W_1 & W_5 W_2 & W_5 W_3 & W_5 W_5 & W_5 W_5 & W_5 W_6 \\ W_6 W_1 & W_6 W_2 & W_6 W_3 & W_6 W_5 & W_6 W_5 & W_6 W_6 \end{bmatrix}$$

Distributed Representation - Word2Vec

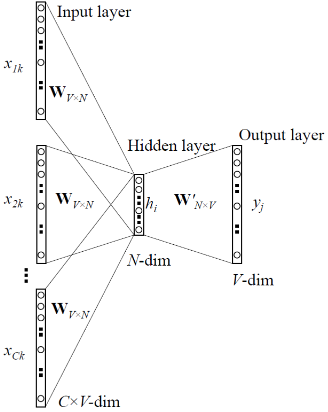


Figure: Continuous Bag Of Words Model

Distributed Representation - Word2Vec

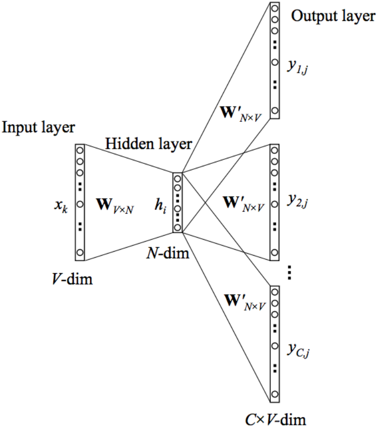


Figure: Skip-Gram Model

Distributed Representation - Word2Vec

$$[we \;\; are \;\; in \;\; cen] \approx \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \times \begin{bmatrix} 1 & 1 \\ 2 & 2 \\ 3 & 3 \\ 4 & 4 \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 2 & 2 \\ 3 & 3 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 1 \\ 2 & 2 \\ 3 & 3 \end{bmatrix} \times \begin{bmatrix} 1 & 2 & 3 & 4 \\ 1 & 2 & 3 & 4 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Distributed Representation - Word2Vec

$$[we\ are\ in\ cen] \approx \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \times W1_{V \times N} \times W2_{N \times V} = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Distributed Representation - Word2Vec

$$we\ are\ in\ cen = \begin{matrix} | \\ we \\ | \end{matrix} + \begin{matrix} | \\ are \\ | \end{matrix} + \begin{matrix} | \\ in \\ | \end{matrix} + \begin{matrix} | \\ cen \\ | \end{matrix}$$

$$we\ are\ in\ cen = matrix\ factorization\left(\begin{bmatrix} | & & | & & | & & | \\ we & + & are & + & in & + & cen \\ | & & | & & | & & | \end{bmatrix}\right)$$

$$matrix\ factorization \in \{svd,\ nmf\}$$

Distributed Representation - Doc2Vec

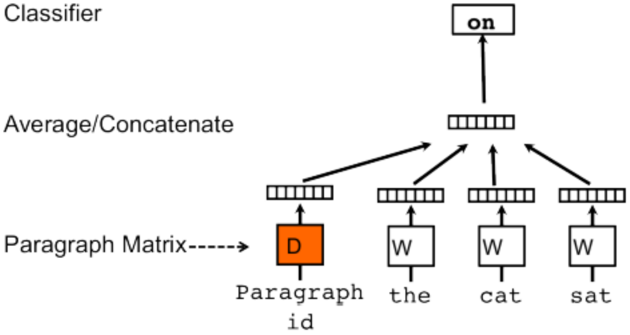


Figure: Distributed Memory Model

Distributed Representation - Doc2Vec

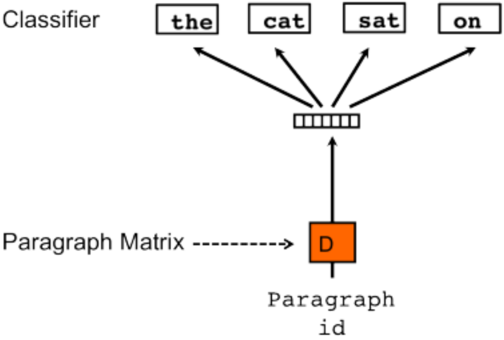
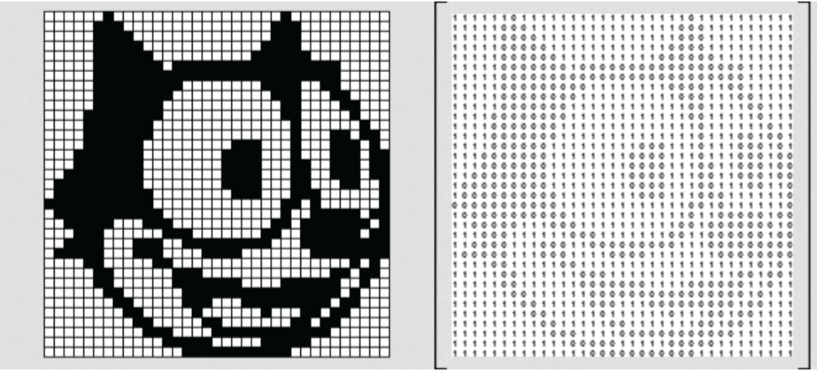


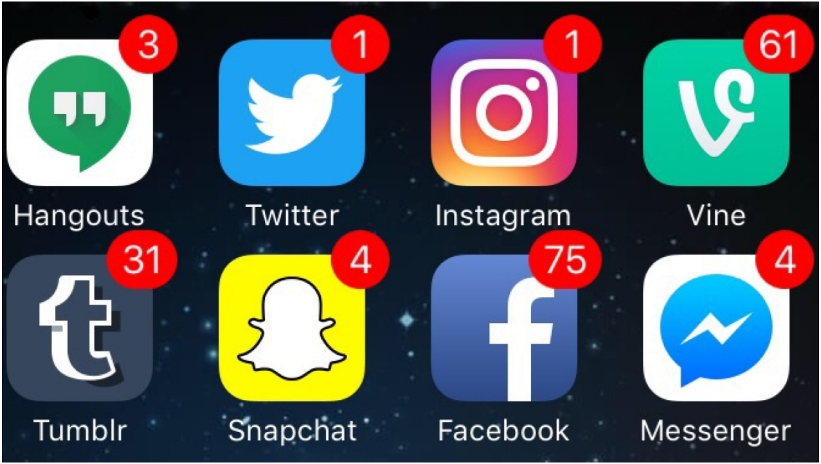
Figure: Distributed Bag Of Words Model

Image to Matrix



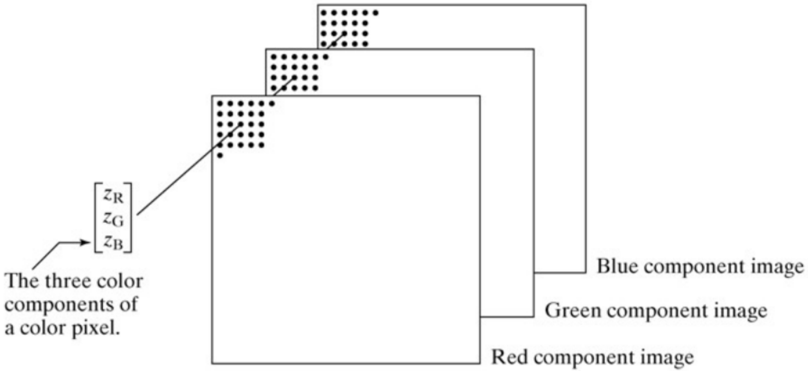
Source: `blog.kleinproject.org/?p=588`

Image to Matrix



Source: `www.cbc.ca/news/trending`

Image to Matrix



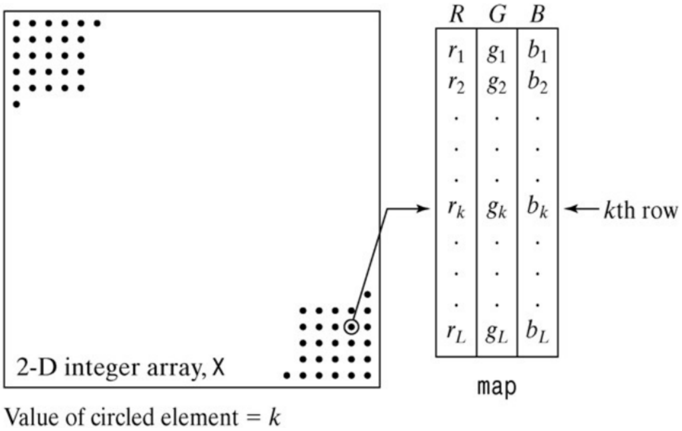
Source: `http://slideplayer.com/slide/8752313/`

Image to Matrix

88	82	84	88	85	83	80	93	102
88	80	78	80	80	78	73	94	100
85	79	80	78	77	74	65	91	99
38	35	40	35	39	74	77	70	65
20	25	23	28	37	69	64	60	57
22	26	22	28	40	65	64	59	34
24	28	24	30	37	60	58	56	66
21	22	23	27	38	60	67	65	67
23	22	22	25	38	59	64	67	66

Source: `www1.adept.com/main/KE/DATA/ACE/AdeptSight_User/Vision_Basics_Mode.html`

Image to Matrix



Source: `slideplayer.com/slide/8752313/`

Thank You.

you can follow me through:

www.linkedin.com/in/barathiganesh-hb

https://barathiganesh-hb.github.io/

https://github.com/BarathiGanesh-HB/