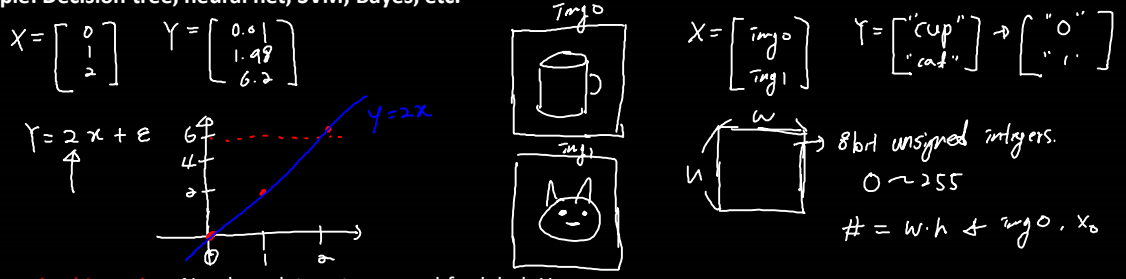
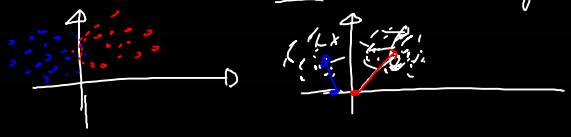
Neural Network

Week1: Basics

Supervised Learning: Need a x – y data set -> most of Machine Learning (ML)
Example: Decision tree, neural net, SVM, Bayes, etc.



Unsupervised Learning: Need a x data set, no need for label. However, you need a measure to tell whether the output is proper.
Example: genetic algorithm, KNN → k – nearest neighbor



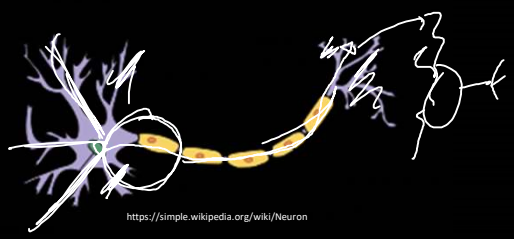
wish i knew

Neural Network

Week1: Basics

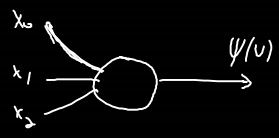
Perceptron

Neuron



https://simple.wikipedia.org/wiki/Neuron

McCulloch-Pits Perceptron

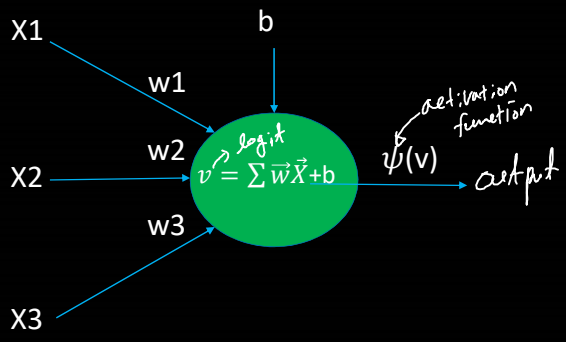


wish i knew

Neural Network

Week1: Basics

Perceptron: a basic unit for neural net



Math notation:

$$v = x_1 w_1 + x_2 w_2 + x_3 w_3 + b$$
$$= \sum_{i=0}^{n} x_i w_i + b \qquad (x_0 = 1)$$
$$= x_1 w_1 + x_2 w_2 + x_3 w_3 + 1 \cdot w_0$$
$$= [1 \; x_1 \; x_2 \; x_3] \, [w_0 \; w_1 \; w_2 \; w_3]^T$$
$$= \vec{X} \cdot \vec{w}^T$$

$$\psi(v) = \begin{cases} 1 & v \geq 0 \\ 0 & o.w \end{cases}$$

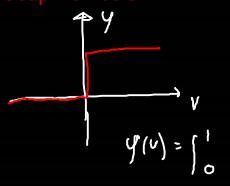
wish i knew

Neural Network

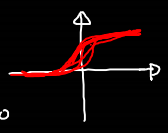
Week1: Basics

Activation Functions

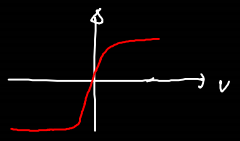
Step Function



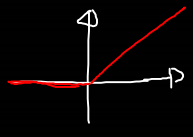
Sigmoid Function



Hyperbolic Tangent Function



Relu Function



wish i knew

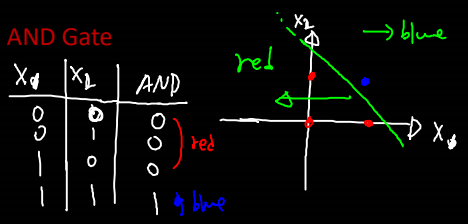
Neural Network

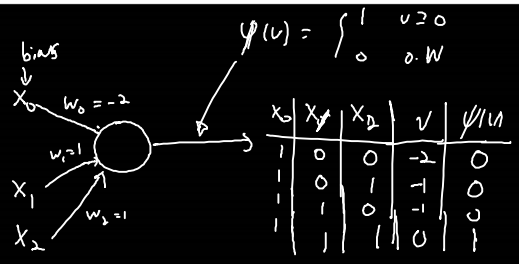
Week1: Basics

Perceptron

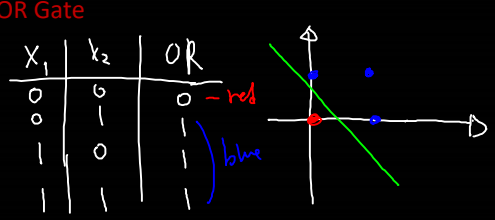
Linear Separability

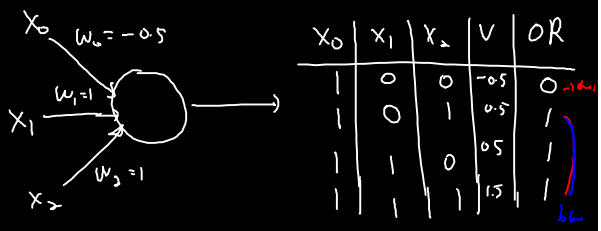
AND Gate





OR Gate





wish i knew

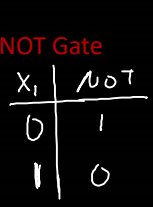
Neural Network

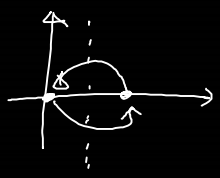
Week1: Basics

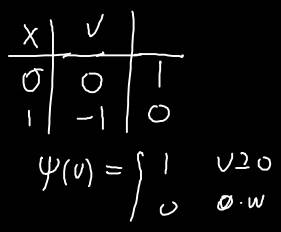
Perceptron

Linear Separability

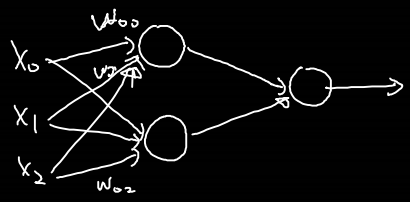
NOT Gate







XOR Gate



wish i knew

Neural Network

Week1: Basics

Perceptron

Code: HW

Perceptron Learning Rule: $w(n+1) = w(n) + \Delta w$ ← next weight values, ← current W. val., ← change of W. val.

$= w(n) + \eta(y - \hat{y})x(n)$

η = learning rate, eta

$y - \hat{y}$ = desired − predicted, error

$x(n)$ = current input

$\eta = 0.5$

$w = [0\ 0\ 0]^T$

$\psi(v) = \begin{cases} 1 & v \geq 0 \\ 0 & 0 w \end{cases}$

A = (0,1), blue = 1

B = (1,0), red = -1

	X	W	V = wx	y	$\hat{y}$	Error y − $\hat{y}$	dw	Next w
A=(0,1) b=1	$\begin{bmatrix}1\\0\\1\end{bmatrix}$	$\begin{bmatrix}0\\0\\0\end{bmatrix}$	0	1	1	0	$\Delta w = \eta \cdot E \cdot input = (0.5)(0)\begin{pmatrix}1\\0\\1\end{pmatrix} = \begin{bmatrix}0\\0\\0\end{bmatrix}$	$\begin{bmatrix}-1\\-1\\0\end{bmatrix}$
B=(1,0) b=1	$\begin{bmatrix}1\\1\\0\end{bmatrix}$	$\begin{bmatrix}0\\0\\0\end{bmatrix}$	0	-1	1	-2	$\Delta w = 0.5(-2)\begin{pmatrix}1\\1\\0\end{pmatrix} = \begin{pmatrix}-1\\-1\\0\end{pmatrix}$	

$\Sigma(\Delta w) = \begin{bmatrix}-1\\-1\\0\end{bmatrix} \rightarrow w(n)$

wish i knew

Neural Network

Week1: Basics

Perceptron

Code: HW

Perceptron Learning Rule: $w(n+1) = w(n) + \Delta w$

$= w(n) + \eta(y - \hat{y})x(n)$

	X	W	V = wx	y	$\hat{y}$	Error y − $\hat{y}$	dw	Next w
A=(0,1) b=1	$\begin{bmatrix}1\\0\\1\end{bmatrix}$	$\begin{bmatrix}-1\\-1\\0\end{bmatrix}$	V = -1 + 0 + 0 = -1	1	-1	2	$\Delta w = 0.5(2)\begin{pmatrix}1\\0\\1\end{pmatrix} = \begin{pmatrix}1\\0\\1\end{pmatrix}$	$w + \Delta w$ $\begin{pmatrix}-1\\-1\\0\end{pmatrix} + \begin{pmatrix}1\\0\\1\end{pmatrix} = \begin{pmatrix}0\\-1\\1\end{pmatrix}$
B=(1,0) b=1	$\begin{bmatrix}1\\1\\0\end{bmatrix}$	$\begin{bmatrix}-1\\-1\\0\end{bmatrix}$	V = -1 - 1 = -2	-1	-1	0	$\Delta w = 0$	

$\Sigma \Delta w = \begin{pmatrix}1\\0\\1\end{pmatrix}$

wish i knew

Neural Network

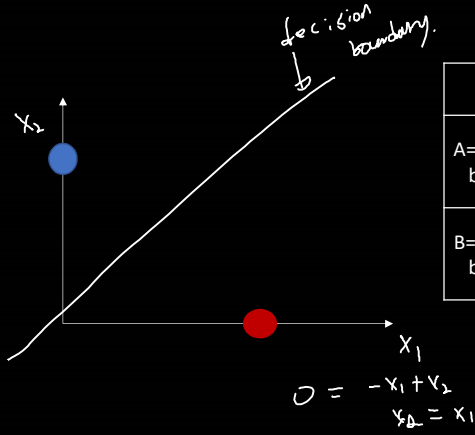
Week1: Basics

Code: HW

Perceptron

Perceptron Learning Rule: $w(n+1) = w(n) + \Delta w$

$= w(n) + \eta(y - \hat{y})x(n)$



	X	W	V = wx	y	$\hat{y}$	Error $y - \hat{y}$	dw	Next w
A=(0,1) b=1	$\begin{pmatrix}1\\0\\1\end{pmatrix}$	$\begin{pmatrix}0\\-1\\1\end{pmatrix}$	1	1	1	0	0	X
B=(1,0) b=1	$\begin{pmatrix}1\\1\\0\end{pmatrix}$	$\begin{pmatrix}0\\-1\\1\end{pmatrix}$	-1	-1	-1	0	0	

$V = X \cdot W = (1 \quad x_1 \quad x_2)\begin{pmatrix}0\\-1\\1\end{pmatrix} = -x_1 + x_2$ $\quad W = \begin{pmatrix}0\\-1\\1\end{pmatrix}$ $\quad bias = 0$

$\psi = \begin{cases} v \geq 0 \\ \psi < 0 \end{cases}$

wish i knew

Neural Network

Week1: Basics

Numerical Solution

First, let's find a root of second order polynomial. $y = ax^2 + bx + c = 0$

Analytical solution: $x = \dfrac{-b + \sqrt{b^2 - 4ac}}{2a}$

Numerical solution: Newton – Rapson method $y = x^2 - 5x + 6 = 0$

$$x_2 = x_1 - \frac{f(x_1)}{f'(x_1)}$$

$$x_3 = x_2 - \frac{f(x_2)}{f'(x_2)}$$

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$

wish i knew

Neural Network

Week1: Basics

Numerical Solution

Code: HW

Numerical solution: Newton − Rapson method $y = x^2 - 5x + 6 = 0$

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$

wish i knew