Home Page

Title Page

Contents

◀◀ ▶▶

◀ ▶

Page 17 of 30

Go Back

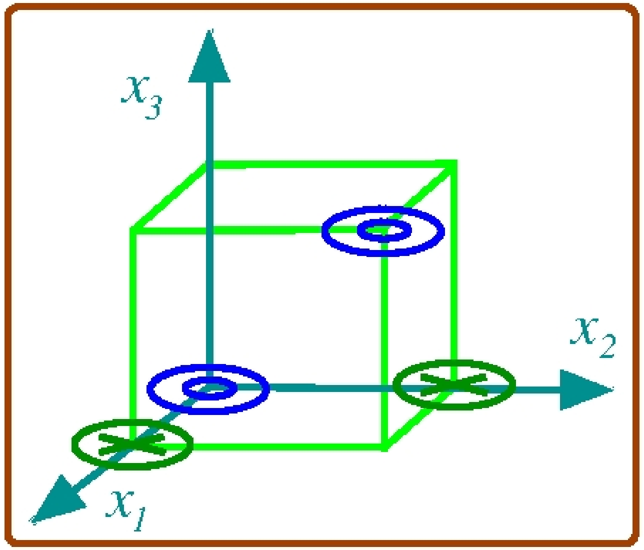
Full Screen

Close

Quit

XOR-1: Feature X'formation/Kernel

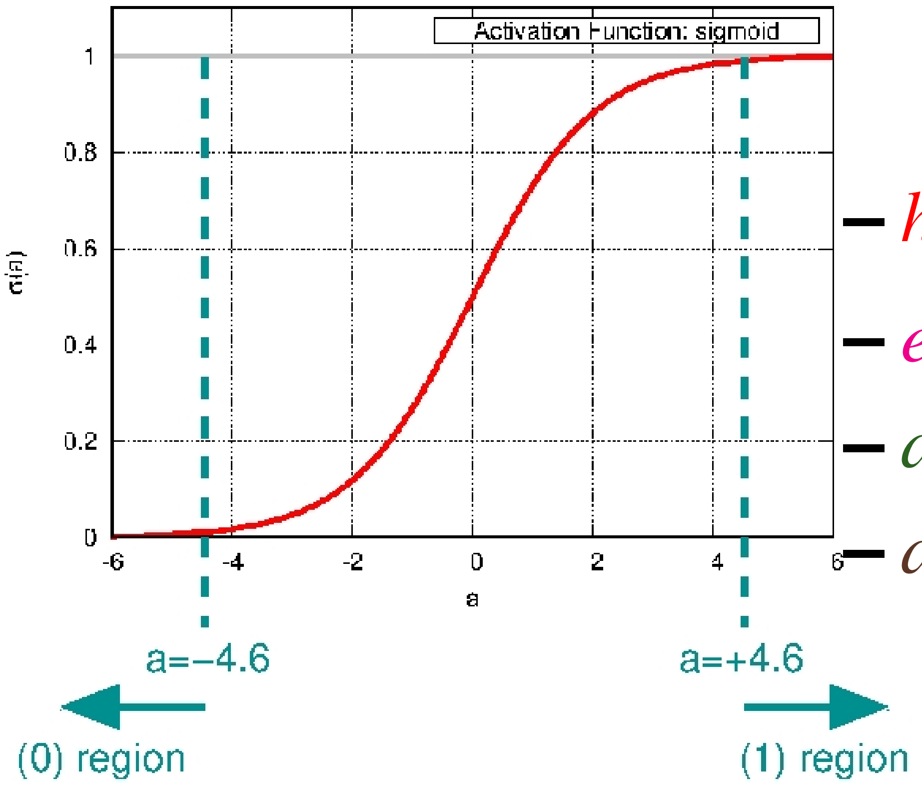
x_2	x_1	$x_3 \triangleq x_2 \cdot x_1$	$x_2 \oplus x_1$
0	0	0	0
0	1	0	1
1	0	0	1
1	1	1	0



- Hidden layer: Multi-Layer Perceptron

XOR-2: from Basic Logic Gates

- Handcrafted Example: sigmoid ($\sigma(a)$) activation

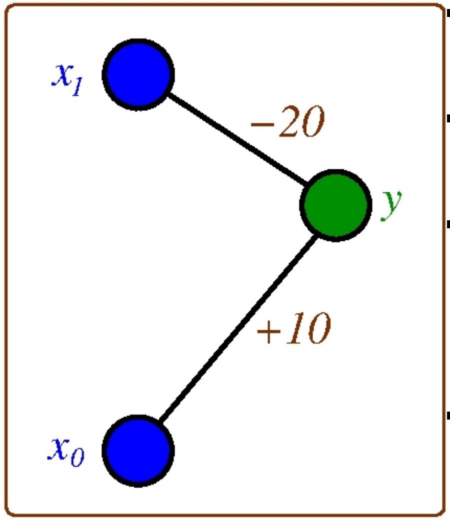


  - $h(a) = \frac{1}{1+e^{-a}}$
  - $e^{4.6} \approx 100$
  - $a = -4.6 : h(a) \approx 0.01$
  - $a = +4.6 : h(a) \approx 0.99$

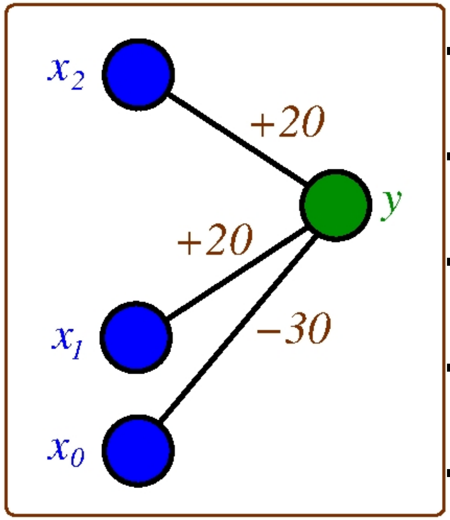
- Boolean Algebra: any function (AND, OR, NOT)

- $x_2 \oplus x_1 \stackrel{\triangle}{=} x_2 \cdot \overline{x_1} + \overline{x_2} \cdot x_1$: Create from AND, OR, NOT

XOR-2 (contd)

- 

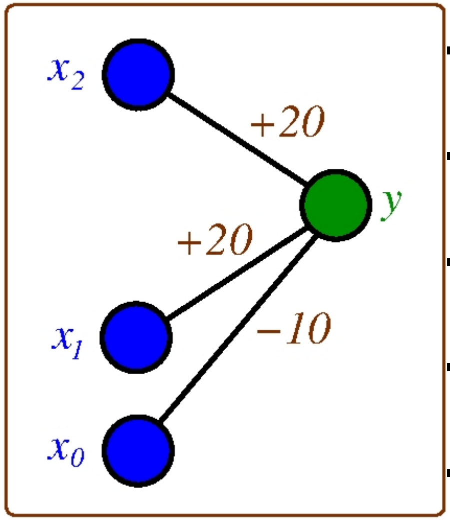
  - NOT: $a = wx + b = -20x_1 + 10$
  - $x_1 = 0,\ a = +10,\ h(+10) \approx 1$
  - $x_1 = 1,\ a = -10,\ h(-10) \approx 0$
x_1	$\overline{x_1}$
0	1
1	0

- 

  - AND: $a = \mathbf{w}^T\mathbf{x} + b = 20x_2 + 20x_1 - 30$
  - $x_2 = 0,\ x_1 = 0,\ a = -30,\ h(-30) \approx 0$
  - $x_2 = 0,\ x_1 = 1,\ a = -10,\ h(-10) \approx 0$
  - $x_2 = 1,\ x_1 = 0,\ a = -10,\ h(-10) \approx 0$
  - $x_2 = 1,\ x_1 = 1,\ a = +10,\ h(+10) \approx 1$

x_2	x_1	$x_2 \cdot x_1$	x_2	x_1	$x_2 \cdot x_1$	x_2	x_1	$x_2 \cdot x_1$	x_2	x_1	$x_2 \cdot x_1$
0	0	0	0	1	0	1	0	0	1	1	1

XOR-2 (contd)

- 

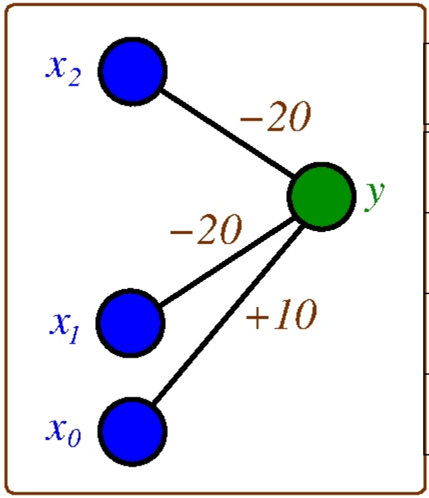
  - OR: $a = \mathbf{w}^T\mathbf{x} + b = 20x_2 + 20x_1 - 10$
  - $x_2 = 0,\ x_1 = 0,\ a = -10,\ h(-10) \approx 0$
  - $x_2 = 0,\ x_1 = 1,\ a = +10,\ h(+10) \approx 1$
  - $x_2 = 1,\ x_1 = 0,\ a = +10,\ h(+10) \approx 1$
  - $x_2 = 1,\ x_1 = 1,\ a = +30,\ h(+30) \approx 1$

x_2	x_1	$x_2 \cdot x_1$	x_2	x_1	$x_2 \cdot x_1$	x_2	x_1	$x_2 \cdot x_1$	x_2	x_1	$x_2 \cdot x_1$
0	0	0	0	1	1	1	0	1	1	1	1

- $x_2 \oplus x_1 \stackrel{\triangle}{=} x_2 \cdot \overline{x_1} + \overline{x_2} \cdot x_1$: Create from AND, OR, NOT
- Block-wise, easy: 2 NOT, 2 AND, 1 OR
- *jugAD*: Optimise with domain-spec info
- Idea: $\overline{x_2} \cdot x_1$ difficult with symmetric 0/1, weights
- $x_2 \odot x_1 = x_2 \cdot x_1 + \overline{x_2} \cdot \overline{x_1}$: sym, AND compl: OR neg
- Half of this ($\overline{x_2} \cdot \overline{x_1}$): easy [1][2][3][4][5][6]

Home Page
Title Page
Contents
◀◀ ▶▶
◀ ▶
Page 21 of 30
Go Back
Full Screen
Close
Quit

XOR-2 (contd)



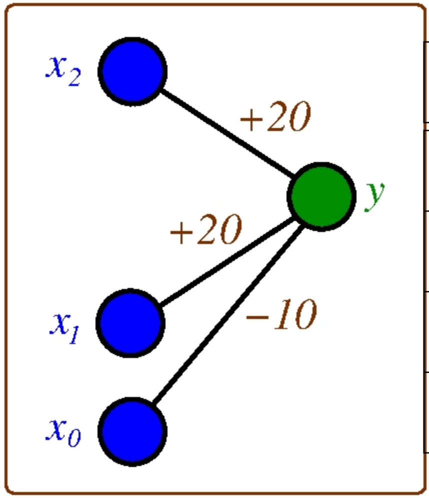
x_2	x_1	$\overline{x_2} \cdot \overline{x_1}$
0	0	1
0	1	0
1	0	0
1	1	0

- $a = -20x_2 - 20x_1 + 10$
- $0, 0 : a = +10, h(+10) = 1$
- $0, 1 : a = -10, h(-10) = 0$
- $1, 0 : a = -10, h(-10) = 0$
- $1, 1 : a = -30, h(-30) = 0$

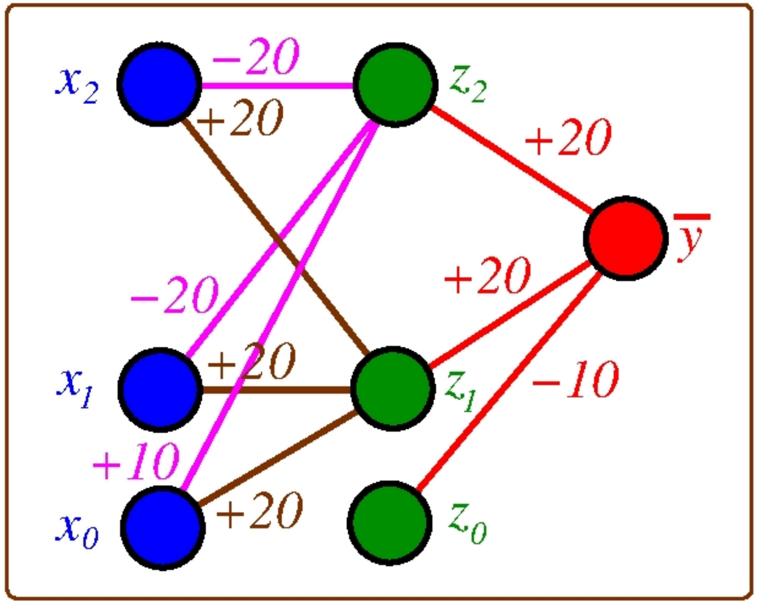
Sign flip gives reqd fn!

Above $\uparrow$: $\overline{x_2 + x_1}$

$\overline{x_2 + x_1} = \overline{x_2} \cdot \overline{x_1}$



x_2	x_1	$x_2 + x_1$
0	0	0
0	1	1
1	0	1
1	1	1

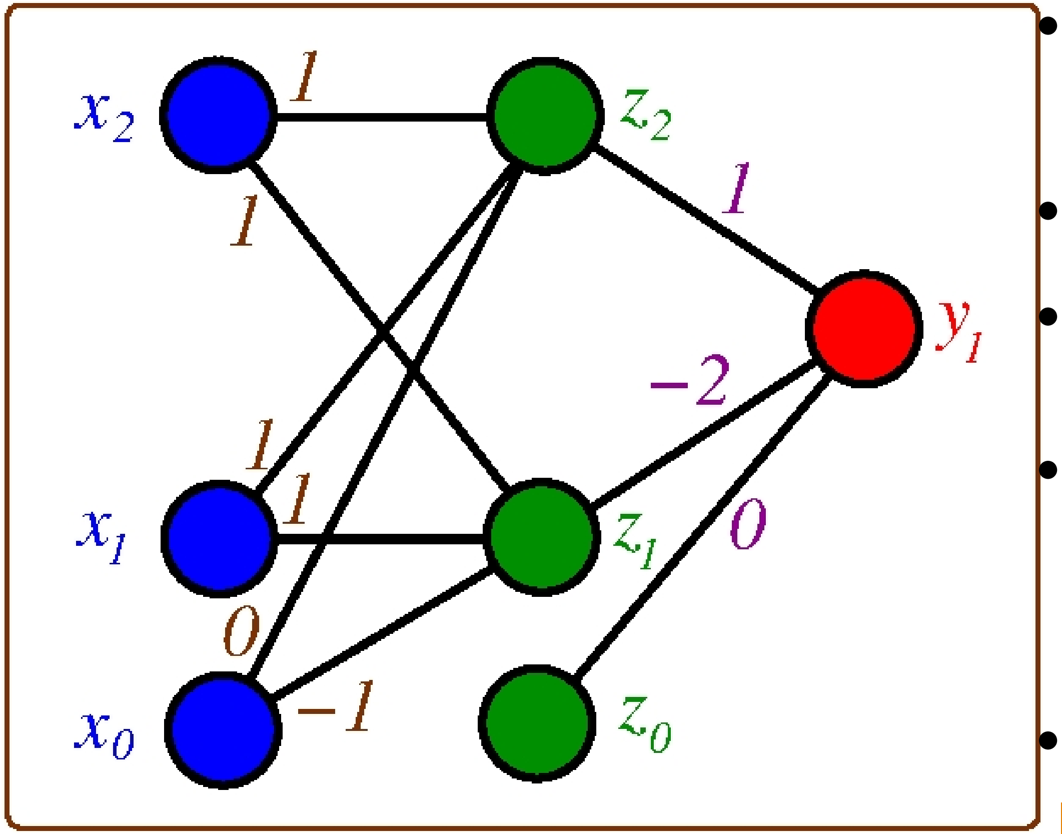


- **[Magenta]** $z_2 = \overline{x_2} \cdot \overline{x_1}$, **[Choco]** $z_1 = x_2 \cdot x_1$, **[Red]** $\overline{y} = z_2 + z_1$

XOR-3: Hidden Layer: minterms

- 2-bit inputs x_2, x_1: 2^2 minterms in the hidden layer
- A multi-input OR gate takes in suitable minterms
- Non-math (on-meth?) intuitive manifestation of a fundamental notion: "A feedforward NN with one hidden layer can represent any Boolean function"
- Non-math (on-meth?) intuitive perspective of a general result: multi-layer feed-forward NNs with non-linear activation fns approx any fn arbit well

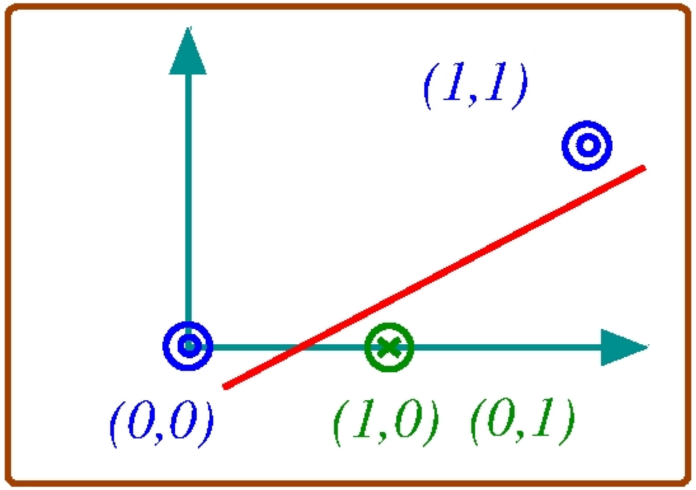
XOR-4: Handcrafted 2-level, ReLU



- All handcrafted: $\mathbf{w}, b, \mathbf{x}$ [00:10]
- $z_j = h(a_j)$, $h(\cdot)$: $ReLU$
- $y_k = \sigma(a_k)$, $\sigma(\cdot)$: $unit$
- $\begin{array}{cccc} \left[\begin{array}{cccc} 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & 1 \end{array}\right. & & & \left.\right] \\ \mathbf{x}_{(1)} & \mathbf{x}_{(2)} & \mathbf{x}_{(3)} & \mathbf{x}_{(4)} \end{array}$: All $\begin{bmatrix} x_2 \\ x_1 \end{bmatrix}$
- $\mathbf{w}_j^{(1)} \begin{bmatrix} 1 \\ 1 \end{bmatrix}$, $\mathbf{b}^{(1)} \begin{bmatrix} 0 \\ -1 \end{bmatrix}$
- $\mathbf{w}_j^{(2)} = \begin{bmatrix} w_2^{(2)} \\ w_1^{(2)} \end{bmatrix} = \begin{bmatrix} 1 \\ -2 \end{bmatrix}$, $b^{(2)} = 0$
- $\mathbf{z} = \begin{bmatrix} z_2 \\ z_1 \end{bmatrix} = \begin{bmatrix} h(a_2^{(1)}) \\ h(a_1^{(1)}) \end{bmatrix}$; $\mathbf{a}^{(1)} = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} \mathbf{x} + \begin{bmatrix} 0 \\ -1 \end{bmatrix}$

XOR-4 (contd)

- $\mathbf{a}^{(1)}_{(1)} = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ -1 \end{bmatrix} = \begin{bmatrix} 0 \\ -1 \end{bmatrix}$; $\mathbf{z}_{(1)} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$
- $\mathbf{a}^{(1)}_{(2)} = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} 0 \\ 1 \end{bmatrix} + \begin{bmatrix} 0 \\ -1 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$; $\mathbf{z}_{(2)} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$
- $\mathbf{a}^{(1)}_{(3)} = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} + \begin{bmatrix} 0 \\ -1 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$; $\mathbf{z}_{(3)} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$
- $\mathbf{a}^{(1)}_{(4)} = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} + \begin{bmatrix} 0 \\ -1 \end{bmatrix} = \begin{bmatrix} 2 \\ 1 \end{bmatrix}$; $\mathbf{z}_{(4)} = \begin{bmatrix} 2 \\ 1 \end{bmatrix}$



- $a^{(2)}_{(1)} = [1 \; -2][0 \; 0]^T = 0$
- $a^{(2)}_{(2)} = [1 \; -2][1 \; 0]^T = 1$
- $a^{(2)}_{(3)} = [1 \; -2][1 \; 0]^T = 1$
- $a^{(2)}_{(4)} = [1 \; -2][2 \; 1]^T = 0$

XOR-5: A Failed Attempt

[Skip Connection]▶

NN: Function Approximator

- A non-linear fn $\mathbf{x}_{D\times 1}$, x_i, $i \in \{1, D\} \rightarrow \mathbf{y}_{K\times 1}$, y_k, $k \in \{1, K\}$, controlled by $\mathbf{w}$: adjustable parameters
- Architecture: computational mechanism
- Best: closed-form e.g., $y(\theta) = sin\theta$
- Boolean fn: Truth table, exhaustively enumerate all 2^N inputs and the corresponding outputs
-

$\mathbf{x}_i$	$f(\mathbf{x}_i)$
$\mathbf{x}_1$	$f(\mathbf{x}_1)$
$\mathbf{x}_2$	$f(\mathbf{x}_2)$
$\cdots$	$\cdots$
$\mathbf{x}_N$	$f(\mathbf{x}_N)$

Non-Boolean case: not possible to enumerate all inputs (+outputs!). Given an $\mathbf{x}_i$ which lies 'between' two training points $\mathbf{x}_i$ and $\mathbf{x}_{i+1}$, we need to interpolate/approximate the output value

- NN: function approximator: 'reasonable' output for an 'unseen' input (not in the training set)