Neural Networks

KAUSHIK ROY

Motivation

➢ Non-linear hypothesis

What we already have:

- High order polynomial logistic regression

Why do we still need Neuron Networks (NNs) ?

Example: Housing sale prediction to a potential buyer

- Feature: # features = 100

  $x_1 = Size$

  $x_2 = \# \ bedrooms$

  $\vdots$

  x_{100}
- Prediction:

  If $y = 1$, predict house will be sold

  If $y = 0$, predict house will not be sold
- Hypothesis: suppose 2^{nd} order poly logistic regression

  $$h_\theta(x) = g(\theta_0 + \theta_1 x_1^2 + \theta_2 x_1 x_2 + \cdots + \theta_{100} x_1 x_{100} + \theta_{101} x_2^2 + \theta_{102} x_2 x_3 + \cdots + \theta_{100} x_2 x_{100} + \cdots)$$

  # features in $h_\theta(x)$ is $O(n^2)$, (≈ 5000 features)

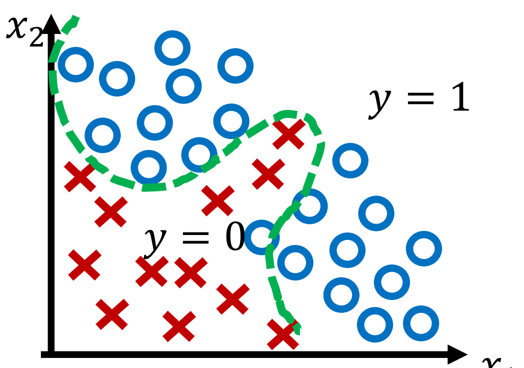
In high order polynomial logistic regression hypothesis:

The # features in k^{th} order polynomial is $O(n^k)$

For many machine learning problems, n and k are large,

which cause the learning model to be:

- Computationally expensive
- Severely overfitting



House sale prediction

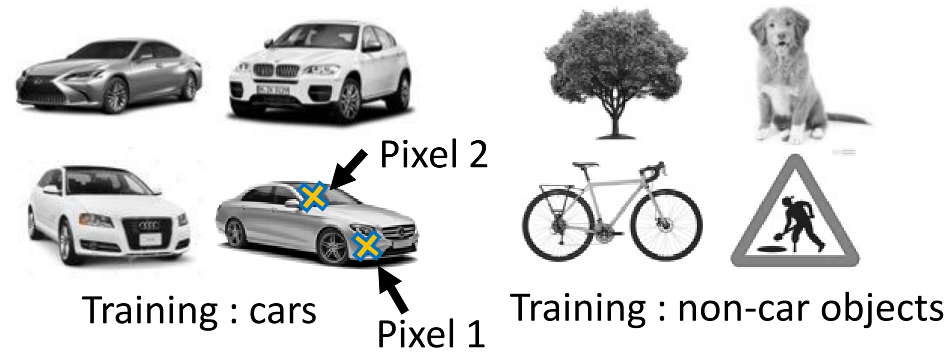
Motivation

➢ Non-linear hypothesis

What we already have:

- High order polynomial logistic regression

Why do we still need Neural Networks (NNs) ?



Computer vision for car detection

Input image size = 50x50 gray scale

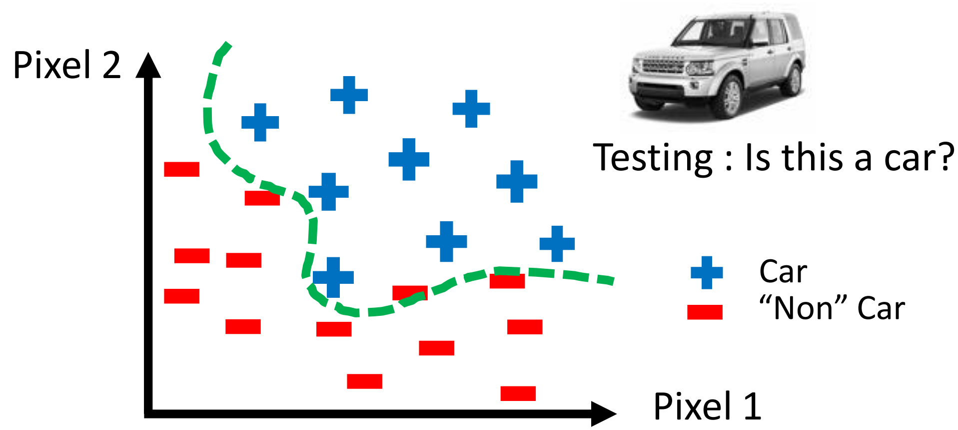
- Feature:

  $x_1 = pixel\ 1\ intensity$

  $x_2 = pixel\ 2\ intensity$

  $\vdots$

  x_{2500}



Computer vision for car detection

- Prediction:

  If $y = 1$, predict: object is a car

  If $y = 0$, predict: no car detected

- Hypothesis:

  Suppose Quadratic polynomial logistic regression

  # Quadratic features $(x_i \times x_j)$: ≈ 3 million

C-BRIC

Motivation

➢ Neurons and the brain

What is Neural Networks (NNs) ?

- Neural Networks are algorithms that try to mimic the brain.

Neural Networks In the past :

- Was very widely used in 80s and early 90s; popularity diminished in late 90s.

Neural Networks nowadays:

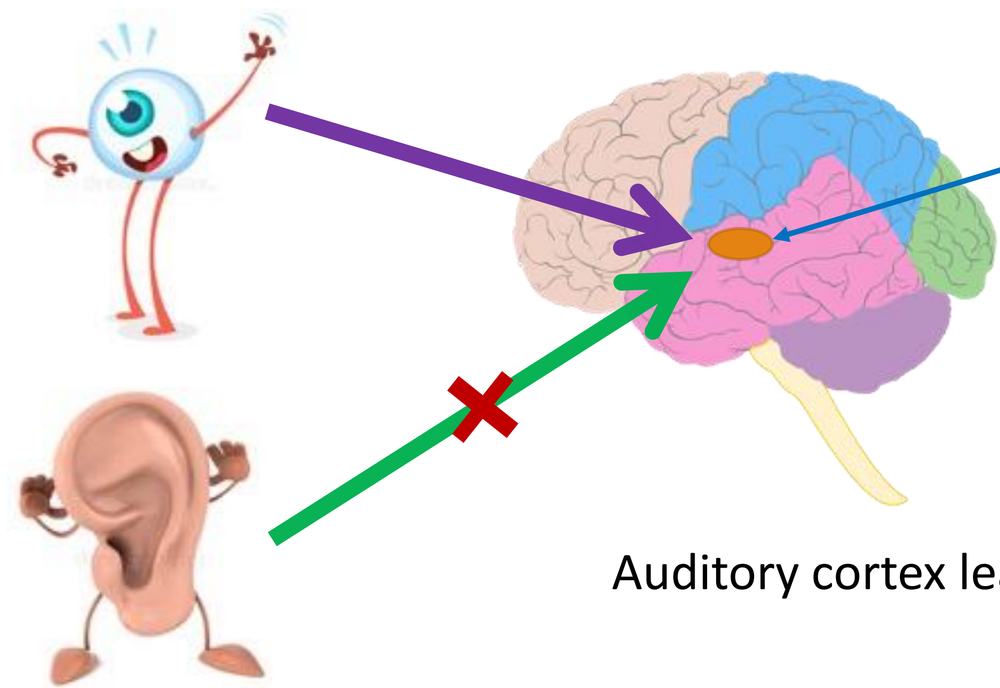
- Recent resurgence; state-of-the-art technique for many applications.

C-BRIC

Motivation

➢ Neurons and the brain

The “one learning algorithm” hypothesis



Auditory cortex learns to see

C-BRIC

Motivation

- Neurons and the brain

Seeing with your tongue

Human echolocation (sonar)

Haptic belt: Direction sense

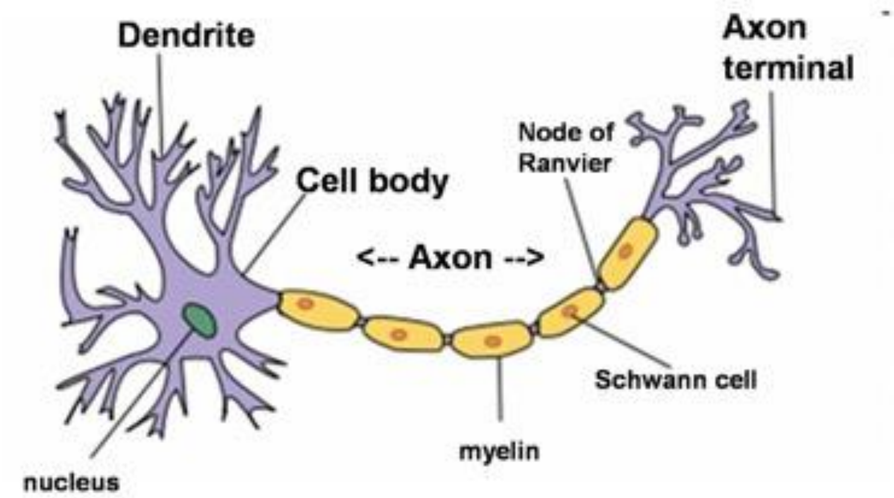
Implanting a 3rd eye

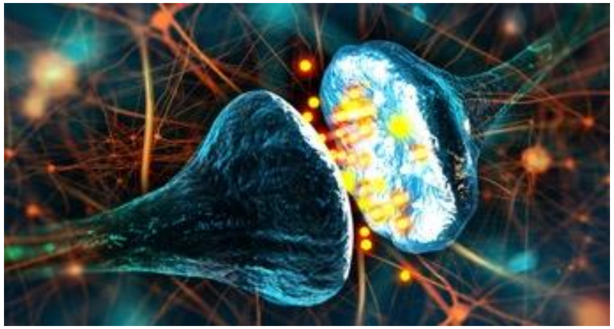
[BrainPort; Welsh & Blashch, 1997; Nagel et al., 2005; Constantine-Paton & Law, 2009]

Neural Networks Representation

➤ Model representation

Neurons in the brain

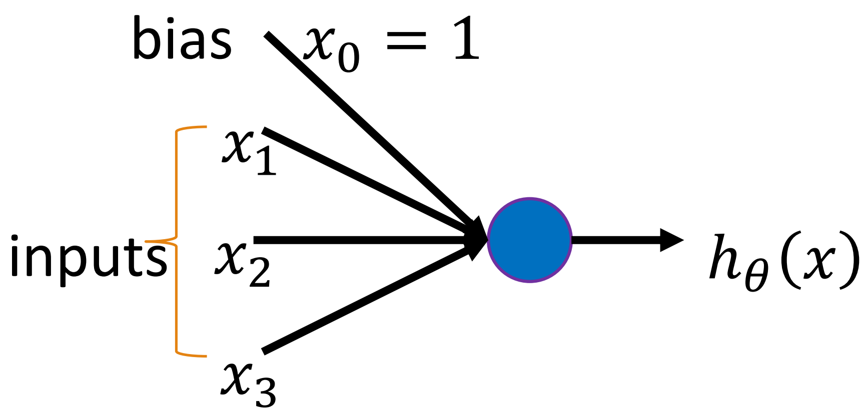




Neural Networks Representation

➢ Model representation

Neurons model: Logistic unit



$$x = \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \end{bmatrix} \quad \theta = \begin{bmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \\ \theta_3 \end{bmatrix}$$

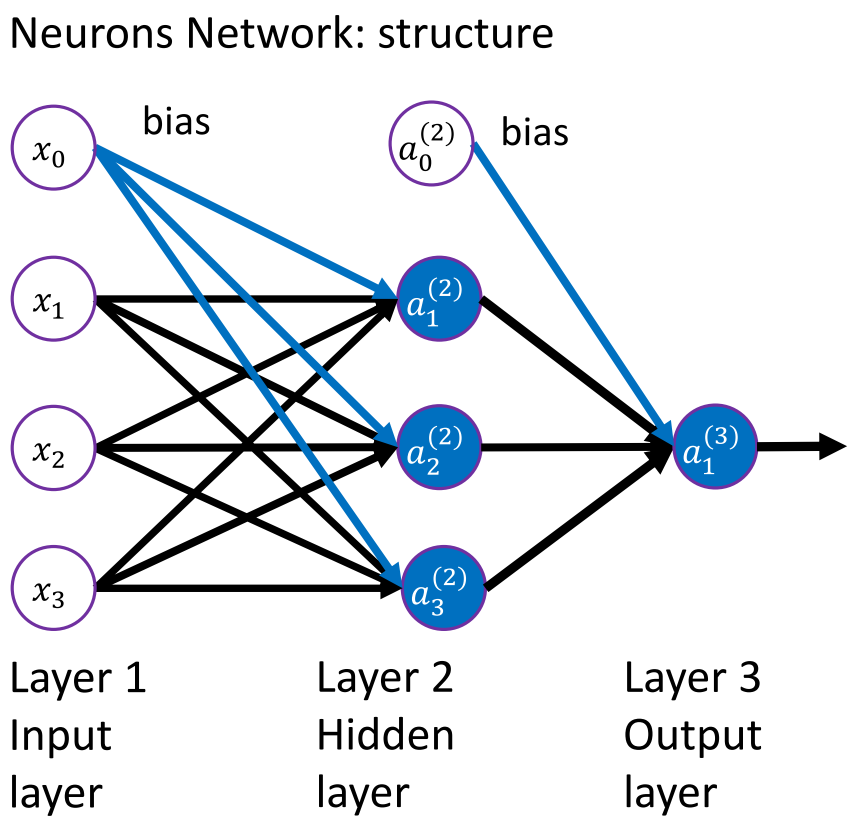
weights (parameters)

Sigmoid (logistic) activation function

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

Neural Networks Representation

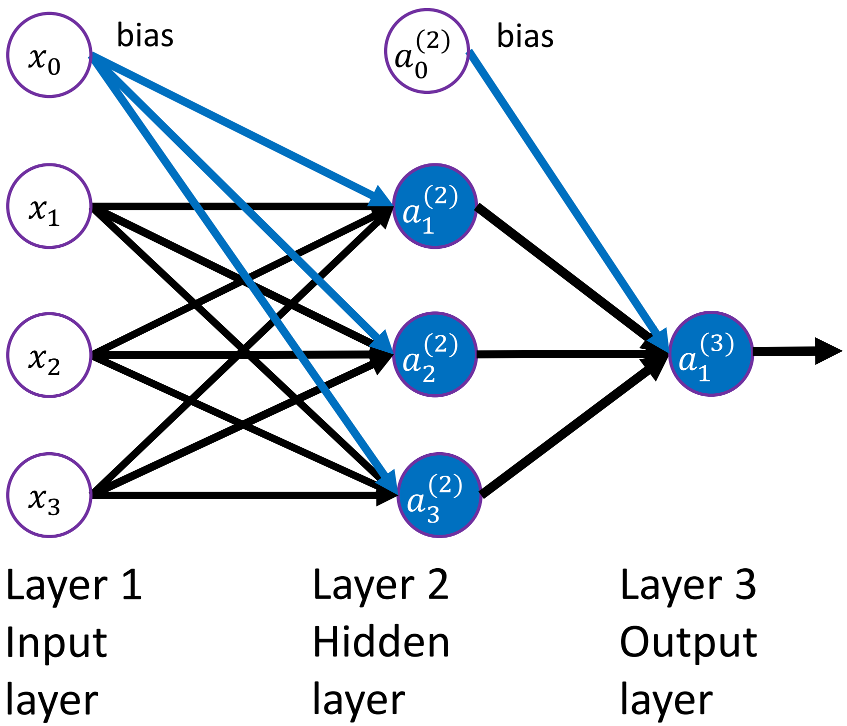
- Model representation



Neural Networks Representation

- Model representation

Neurons Network: computation



$a_i^{(j)}$ = "activation" of unit i in layer j

$\theta^{(j)}$ = matrix of weights controlling function mapping from layer j to layer $j+1$

$$a_1^{(2)} = g\left(\theta_{1,0}^{(1)}x_0 + \theta_{1,1}^{(1)}x_1 + \theta_{1,2}^{(1)}x_2 + \theta_{1,3}^{(1)}x_3\right)$$

$$a_2^{(2)} = g\left(\theta_{2,0}^{(1)}x_0 + \theta_{2,1}^{(1)}x_1 + \theta_{2,2}^{(1)}x_2 + \theta_{2,3}^{(1)}x_3\right)$$

$$a_3^{(2)} = g\left(\theta_{3,0}^{(1)}x_0 + \theta_{3,1}^{(1)}x_1 + \theta_{3,2}^{(1)}x_2 + \theta_{3,3}^{(1)}x_3\right)$$

$$h_\theta(x) = a_1^{(3)} = g\left(\theta_{1,0}^{(2)}a_0 + \theta_{1,1}^{(2)}a_1 + \theta_{1,2}^{(2)}a_2 + \theta_{1,3}^{(2)}a_3\right)$$

If network has s_j units in layer j, s_{j+1} units in layer $j+1$,
Then $\theta^{(j)}$ will be of dimension $s_{j+1} \times (s_j + 1)$

Neural Networks Representation

➢ Model representation

Forward propagation: Vectorized implementation

$$a_1^{(2)} = g\left(\theta_{1,0}^{(1)} x_0 + \theta_{1,1}^{(1)} x_1 + \theta_{1,2}^{(1)} x_2 + \theta_{1,3}^{(1)} x_3\right)$$

$$a_2^{(2)} = g\left(\theta_{2,0}^{(1)} x_0 + \theta_{2,1}^{(1)} x_1 + \theta_{2,2}^{(1)} x_2 + \theta_{2,3}^{(1)} x_3\right)$$

$$a_3^{(2)} = g\left(\theta_{3,0}^{(1)} x_0 + \theta_{3,1}^{(1)} x_1 + \theta_{3,2}^{(1)} x_2 + \theta_{3,3}^{(1)} x_3\right)$$

$$\begin{aligned} h_\theta(x) &= a_1^{(3)} \\ &= g\left(\theta_{1,0}^{(2)} a_0 + \theta_{1,1}^{(2)} a_1 + \theta_{1,2}^{(2)} a_2 + \theta_{1,3}^{(2)} a_3\right) \end{aligned}$$

$$x = \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \end{bmatrix} \qquad z^{(2)} = \begin{bmatrix} z_1^{(2)} \\ z_2^{(2)} \\ z_3^{(2)} \end{bmatrix}$$

$$z^{(2)} = \theta^{(1)} a^{(1)}$$

$$a^{(2)} = g\left(z^{(2)}\right)$$

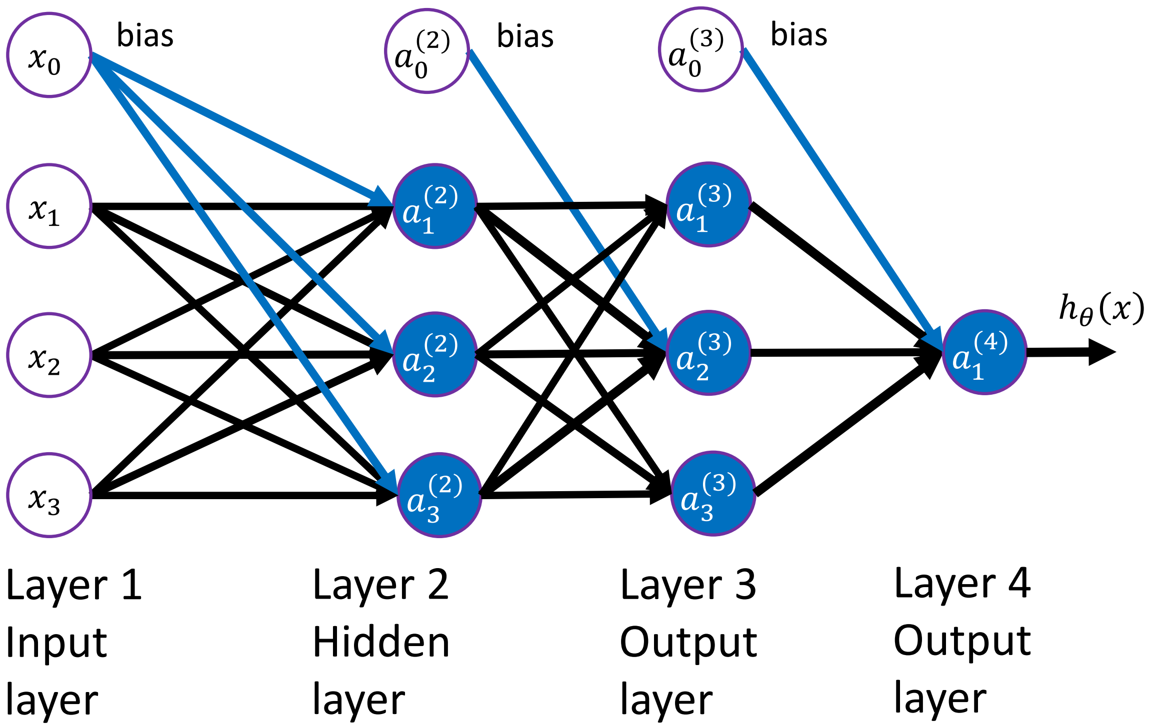
$$z^{(3)} = \theta^{(2)} a^{(2)}$$

$$h_\theta(x) = a^{(3)} = g\left(z^{(3)}\right)$$

Neural Networks Representation

➢ Network Architecture

Other Neural Network architectures:



Neural Network with two hidden layers.

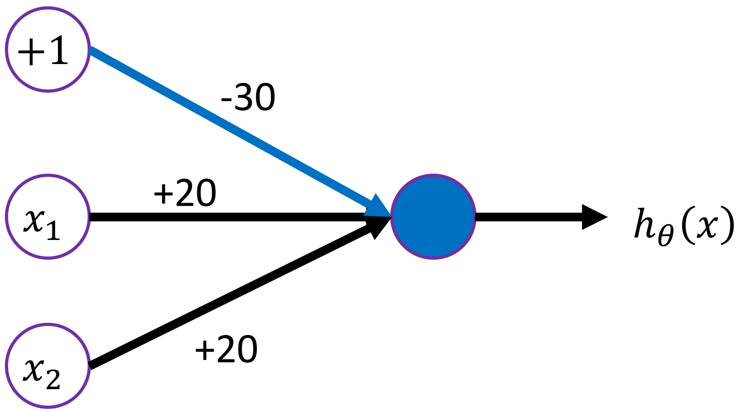
Neural Networks Representation

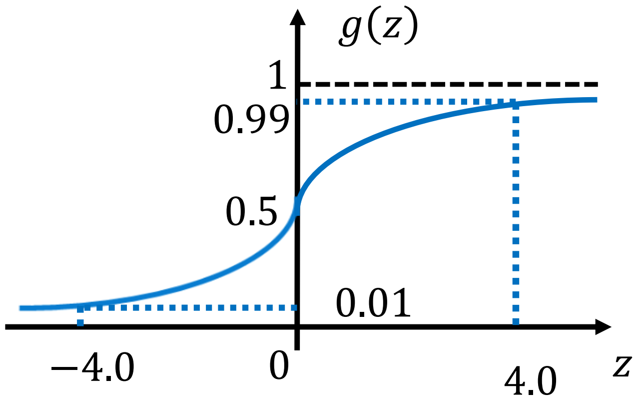
- Examples and intuitions

Simple example: AND

$x_1, x_2 \in \{0, 1\}$

$y = x_1 \text{ AND } x_2$





$$h_\theta(x) = g(-30 + 20x_1 + 20x_2)$$

x_1	x_2	$h_\theta(x)$
0	0	$g(-30) \approx 0$
0	1	$g(-10) \approx 0$
1	0	$g(-10) \approx 0$
1	1	$g(10) \approx 1$

C-BRIC

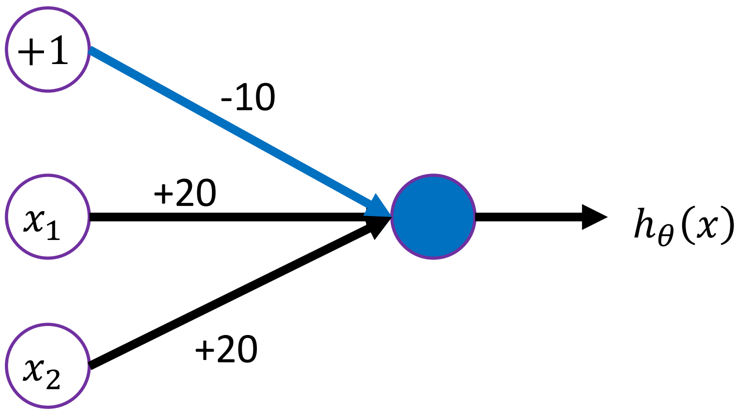
Neural Networks Representation

➢ Examples and intuitions

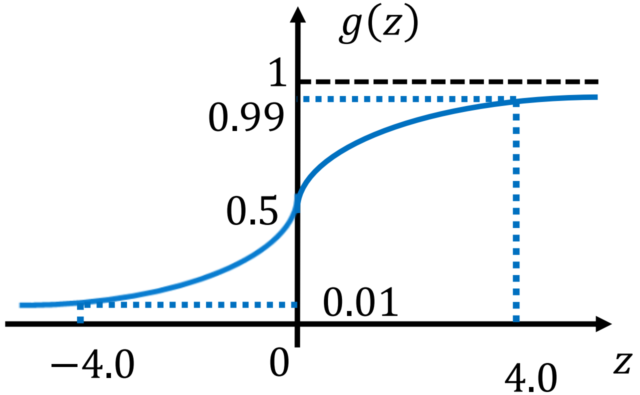
Simple example: OR

$x_1, x_2 \in \{0, 1\}$

$y = x_1 \text{ OR } x_2$



$$h_\theta(x) = g(-10 + 20x_1 + 20x_2)$$



x_1	x_2	$h_\theta(x)$
0	0	$g(-10) \approx 0$
0	1	$g(10) \approx 1$
1	0	$g(10) \approx 1$
1	1	$g(30) \approx 1$

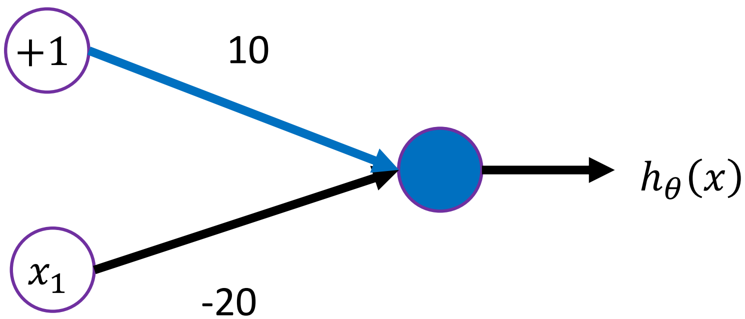
Neural Networks Representation

- Examples and intuitions

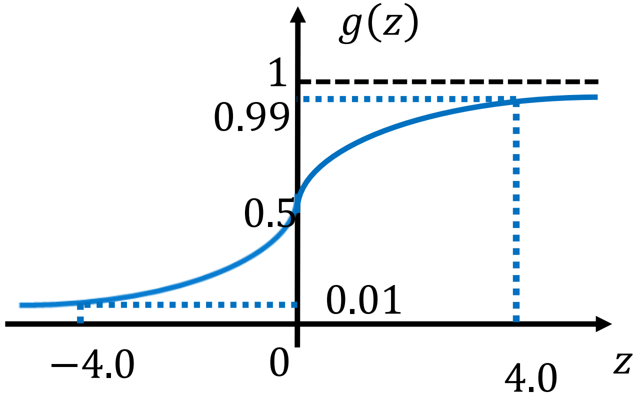
Simple example: NOT

$x_1, x_2 \in \{0, 1\}$

$y = x_1 \text{ NOT } x_2$



$h_\theta(x) = g(10 - 20x_1)$



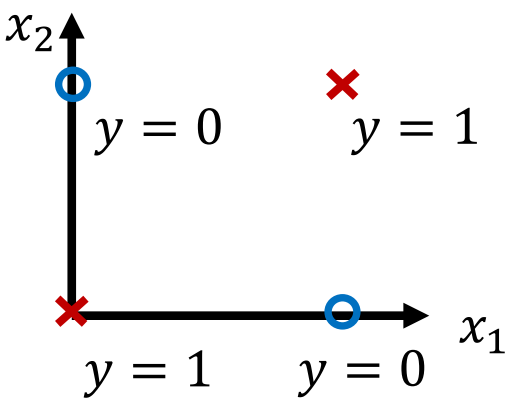
x_1	$h_\theta(x)$
0	$g(10) \approx 1$
1	$g(-10) \approx 1$

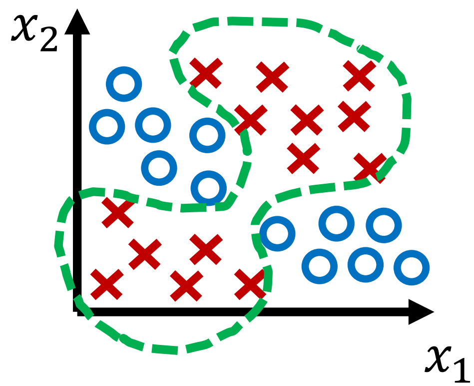
Neural Networks Representation

- Examples and intuitions

Non-linear classification examples: XOR/XNOR

x_1, x_2 are binary (0 or 1).

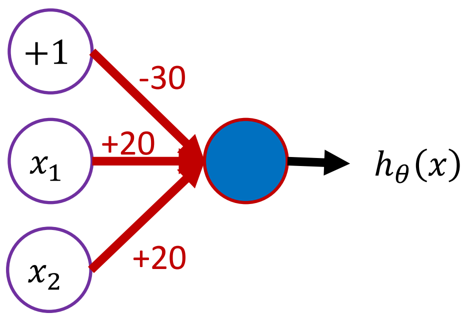




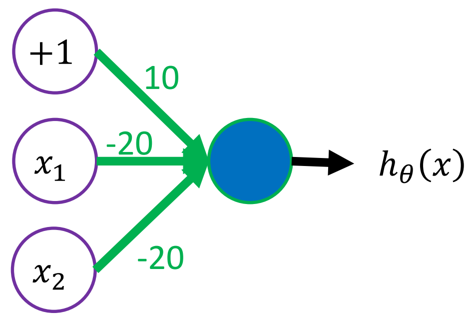
$$y = x_1 \text{ XOR } x_2$$
$$y = x_1 \text{ XNOR } x_2$$
$$y = \text{NOT } (x_1 \text{ XOR } x_2)$$

Neural Networks Representation

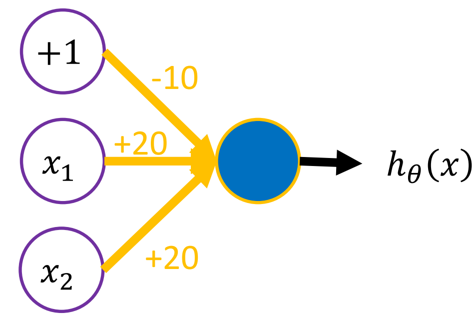
- Examples and intuitions



x_1 AND x_2

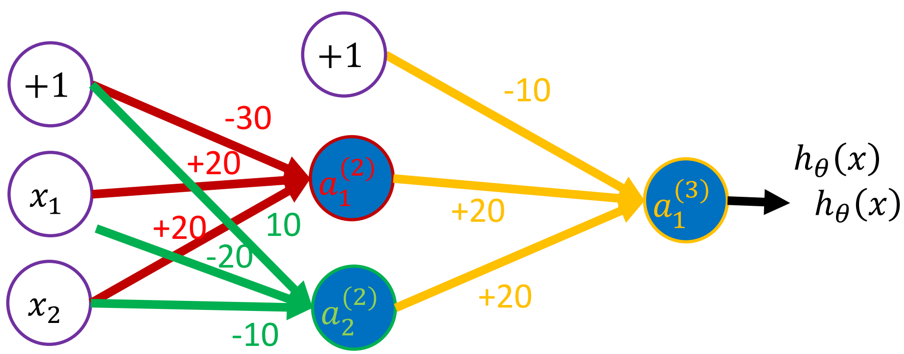


(NOT x_1) OR (NOT x_2)



x_1 OR x_2

Putting them them together: x_1 XOR x_2



x_1	x_2	$a_1^{(2)}$	$a_2^{(2)}$	$h_\theta(x)$
0	0	0	1	1
0	1	0	0	0
1	0	0	0	0
1	1	1	0	1

Multi-class Classification

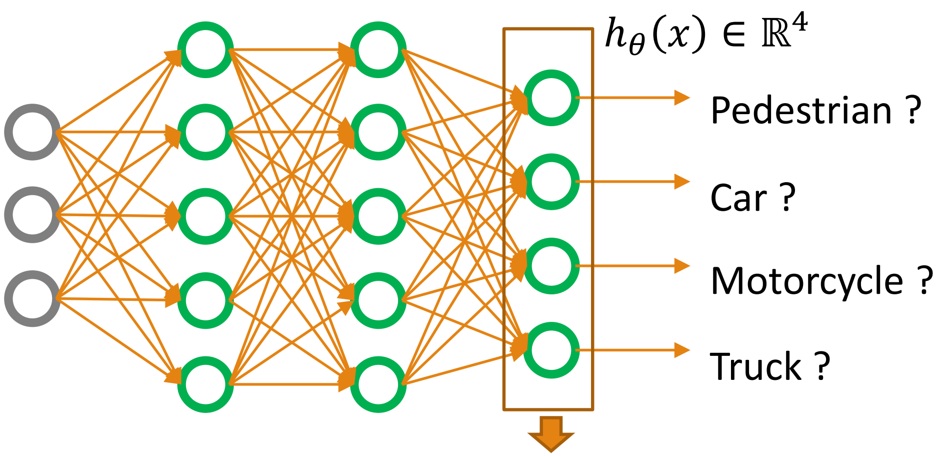
- Multiple output units: One-vs-all

Pedestrian

Car

Motorcycle

Truck



Each output neuron performs a binary classification task.

Want $h_\theta(x) = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$, when pedestrian

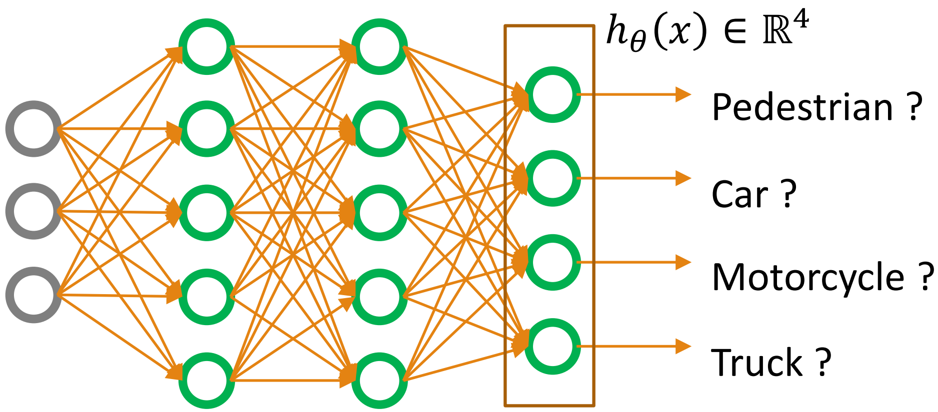
Want $h_\theta(x) = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}$, when car

Want $h_\theta(x) = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}$, when motorcycle

C-BRIC

Multi-class Classification

- Multiple output units: One-vs-all



Want $h_\theta(x) = \begin{bmatrix}1\\0\\0\\0\end{bmatrix}$, when pedestrian

Want $h_\theta(x) = \begin{bmatrix}0\\1\\0\\0\end{bmatrix}$, when car

Want $h_\theta(x) = \begin{bmatrix}0\\0\\1\\0\end{bmatrix}$, when motorcycle

Training set: $(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), (x^{(3)}, y^{(3)}), (x^{(4)}, y^{(4)})$

$y^{(i)}$ is one of $\begin{bmatrix}1\\0\\0\\0\end{bmatrix}, \begin{bmatrix}0\\1\\0\\0\end{bmatrix}, \begin{bmatrix}0\\0\\1\\0\end{bmatrix}, \begin{bmatrix}0\\0\\0\\1\end{bmatrix}$

Pedestrian Car Motorcycle Truck

C-BRIC