Machine Learning and Data Mining:

Neural Network

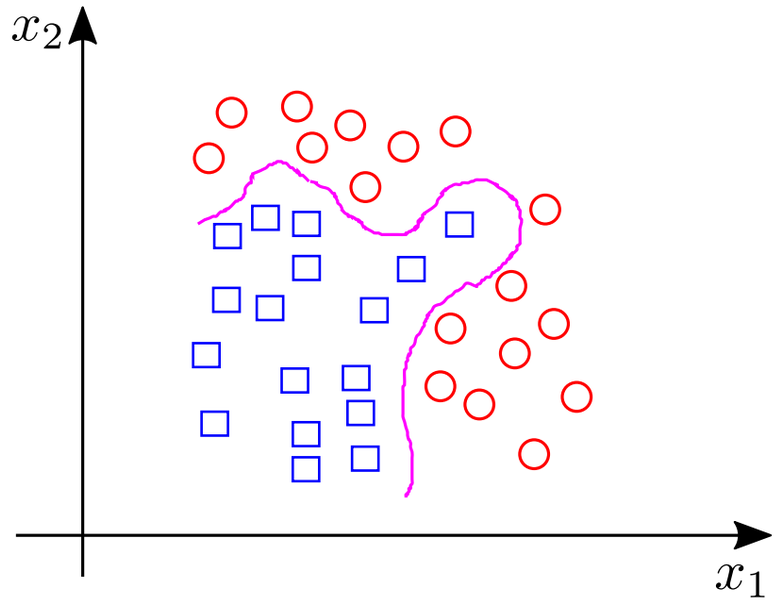
Xiuxia Du, Ph.D.
Department of Bioinformatics and Genomics
University of North Carolina at Charlotte

Some of the materials in this lecture are taken from the Machine Learning Course by Andrew Ng on Coursera.com.

Introduction

- Origins: algorithms that try to mimic the brain.
- Was very widely used in the 80's and early 90's.
- Popularity diminished in late 90's.
- Recent resurgence: state-of-the-art technique for many application.

Nonlinear classification



- If logistic regression is used, higher-order terms need to be considered for a complex nonlinear classification:

$$g(z) = g(\theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_1 x_2 + \theta_4 x_1^2 x_2 + \theta_5 x_1^3 x_2 + \theta_6 x_1 x_2^2 + \cdots)$$

- However, the number of higher-order terms increases exponentially with the number of features.

  Quadratic terms, like $x_1^2, x_1 x_2, x_1 x_3, \cdots$, are $\sim \mathcal{O}(n^2)$

  Cubic terms, like $x_1 x_2 x_3, x_1^2 x_2, x_{20} x_{30} x_{40}, \cdots$, are $\sim \mathcal{O}(n^3)$.

- To train a logistic regression model with so many terms is very challenging.

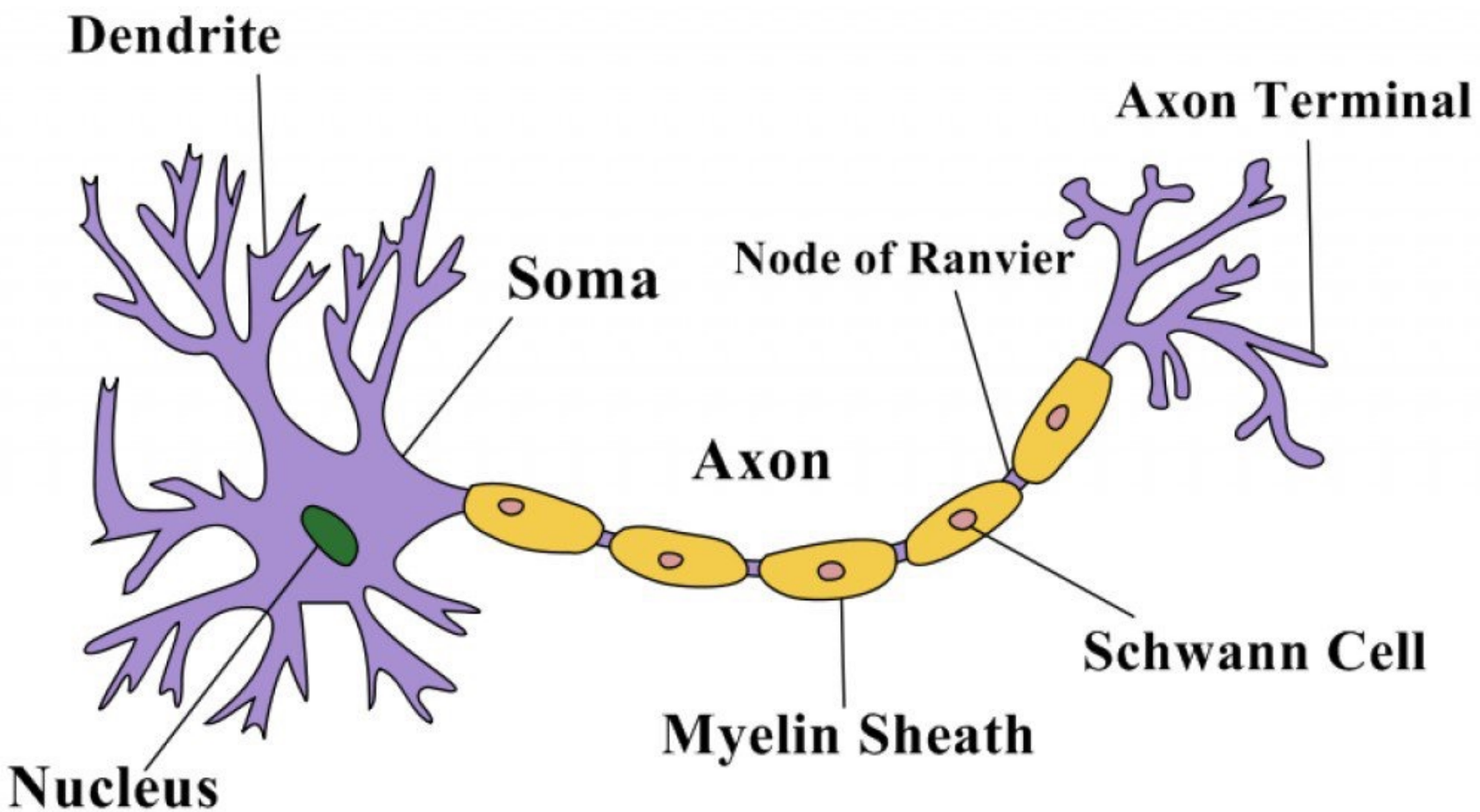
Nonlinear classification

- Example: image recognition

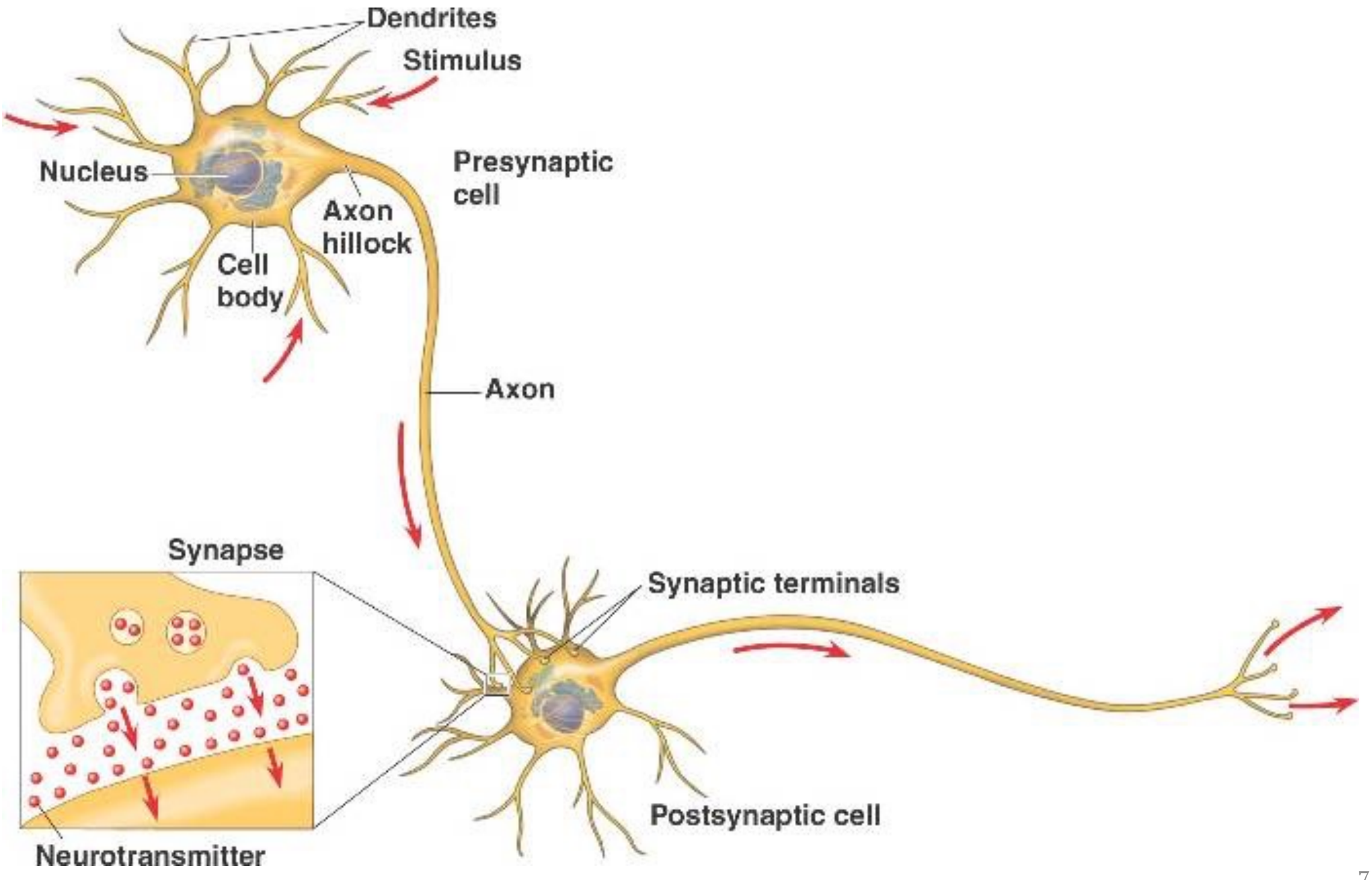
  If the number of pixels in an image is 50×50, then the total number of pixels is 2500. That means that the number of features n is 2500 if gray scale image is used. If pixel value are represented in RGB format, then the number of features n is 7500.

  If quadratic or cubic terms are also considered, using a logistic regression classifier becomes very difficult.

Neuron

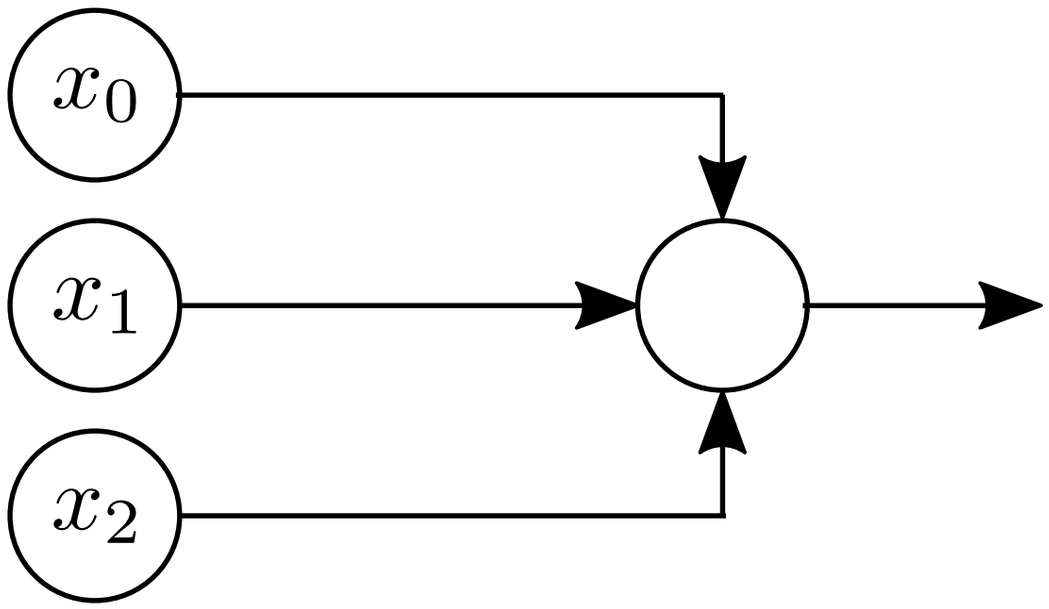

Dendrite
Soma
Axon Terminal
Node of Ranvier
Axon
Schwann Cell
Myelin Sheath
Nucleus

Neural networks



Neural Network Representation

Neuron model



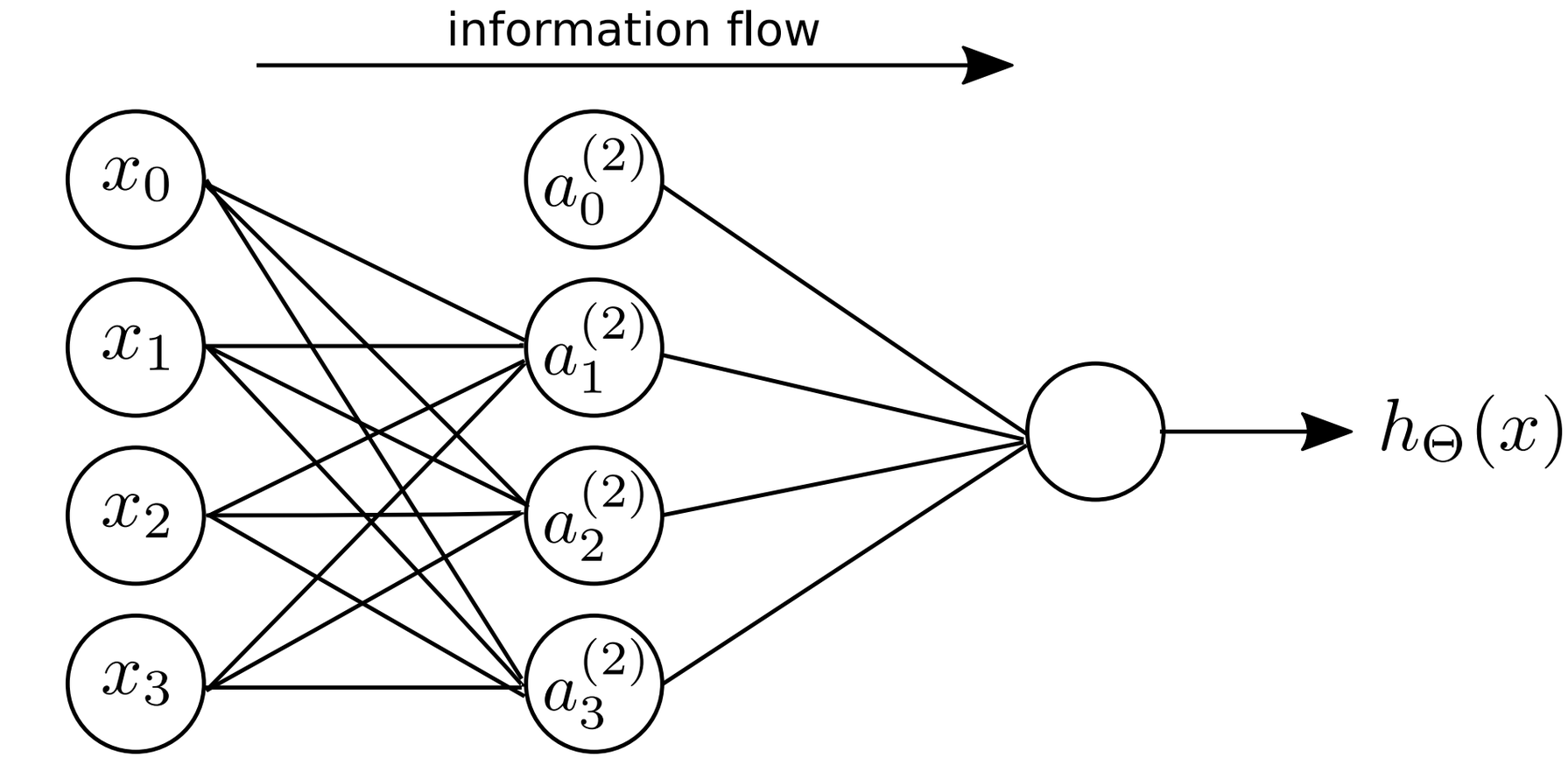
$$h_\theta(x) = \frac{1}{1 + e^{-(\theta_0 x_0 + \theta_1 x_1 + \theta_2 x_2)}}$$

with $x_0 = 1$ representing the bias term and h_θ is the sigmoid/logistic activation function. $\theta_0, \theta_1, \theta_2$ are called weights or parameters.

In general,

$$x = \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}, \quad \theta = \begin{bmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \\ \vdots \\ \theta_n \end{bmatrix}, \quad h_\theta(x) = \frac{1}{1 + e^{-\theta^T x}}$$

Neural network model



layer 1
input layer

layer 2
hidden layer

layer 3
output layer

Each neuron is called a unit.

Neural network representation

- Denote a matrix of weights, Θ, with the Ith row, $\Theta^{(j)}_{I,:}$, representing the function mapping to unit I in layer $j+1$ from units in layer j.
- Denote $a_i^{(j)}$ to represent the "activation" of unit i in layer j.
- Then we have

$$\begin{aligned} a_1^{(2)} &= g\left(\Theta_{10}^{(1)}x_0 + \Theta_{11}^{(1)}x_1 + \Theta_{12}^{(1)}x_2 + \Theta_{13}^{(1)}x_3\right) \\ a_2^{(2)} &= g\left(\Theta_{20}^{(1)}x_0 + \Theta_{21}^{(1)}x_1 + \Theta_{22}^{(1)}x_2 + \Theta_{23}^{(1)}x_3\right) \\ a_3^{(2)} &= g\left(\Theta_{30}^{(1)}x_0 + \Theta_{31}^{(1)}x_1 + \Theta_{32}^{(1)}x_2 + \Theta_{33}^{(1)}x_3\right) \end{aligned}$$

with $\Theta^{(1)} \in \mathbb{R}^{3\times 4}$ and

$$h_\Theta(x) = a_1^{(3)} = g\left(\Theta_{10}^{(2)}a_0^{(2)} + \Theta_{11}^{(2)}a_1^{(2)} + \Theta_{12}^{(2)}a_2^{(2)} + \Theta_{13}^{(2)}a_3^{(2)}\right)$$

with $\Theta^{(2)} \in \mathbb{R}^{1\times 4}$.

Neural network representation

- In general, if a network has s_j units in layer j and $s_{(j+1)}$ units in layer $j+1$, then $\Theta^{(j)}$ will be of dimension $s_{(j+1)} \times (s_j + 1)$.

Forward propagation

- Denote $z_1^{(2)}, z_2^{(2)}, z_3^{(2)}$ as the input to the first, second, and third unit in the hidden layer:

$$\begin{aligned} z_1^{(2)} &= \Theta_{10}^{(1)} x_0 + \Theta_{11}^{(1)} x_1 + \Theta_{12}^{(1)} x_2 + \Theta_{13}^{(1)} x_3 \\ z_2^{(2)} &= \Theta_{20}^{(1)} x_0 + \Theta_{21}^{(1)} x_1 + \Theta_{22}^{(1)} x_2 + \Theta_{23}^{(1)} x_3 \\ z_3^{(2)} &= \Theta_{30}^{(1)} x_0 + \Theta_{31}^{(1)} x_1 + \Theta_{32}^{(1)} x_2 + \Theta_{33}^{(1)} x_3 \end{aligned}$$

  and denote

$$z^{(2)} = \begin{bmatrix} z_1^{(2)} \\ z_2^{(2)} \\ z_3^{(2)} \end{bmatrix}$$

- Then we have

$$\begin{aligned} z^{(2)} &= \Theta^{(1)} x \\ a^{(2)} &= g\left(z^{(2)}\right) \end{aligned}$$

  where $g(\cdot)$ applies to each element of $z^{(2)}$ separately.

Forward propagation

- Define

$$a^{(1)} := x = \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \end{bmatrix}$$

then

$$\begin{aligned} z^{(2)} &= \Theta^{(1)} a^{(1)} \\ a^{(2)} &= g\left(z^{(2)}\right) \end{aligned}$$

with $a^{(1)} \in \mathbb{R}^{4\times 1}$, $\Theta^{(1)} \in \mathbb{R}^{3\times 4}$, $z^{(2)} \in \mathbb{R}^{3\times 1}$ and $a^{(2)} \in \mathbb{R}^{3\times 1}$.

- Add $a_0^{(2)} = 1$, then we have

$$\begin{aligned} z^{(3)} &= \Theta^{(2)} a^{(2)} \\ h_\Theta(x) &= a^{(3)} = g\left(z^{(3)}\right) \end{aligned}$$

Forward propagation

- In general,

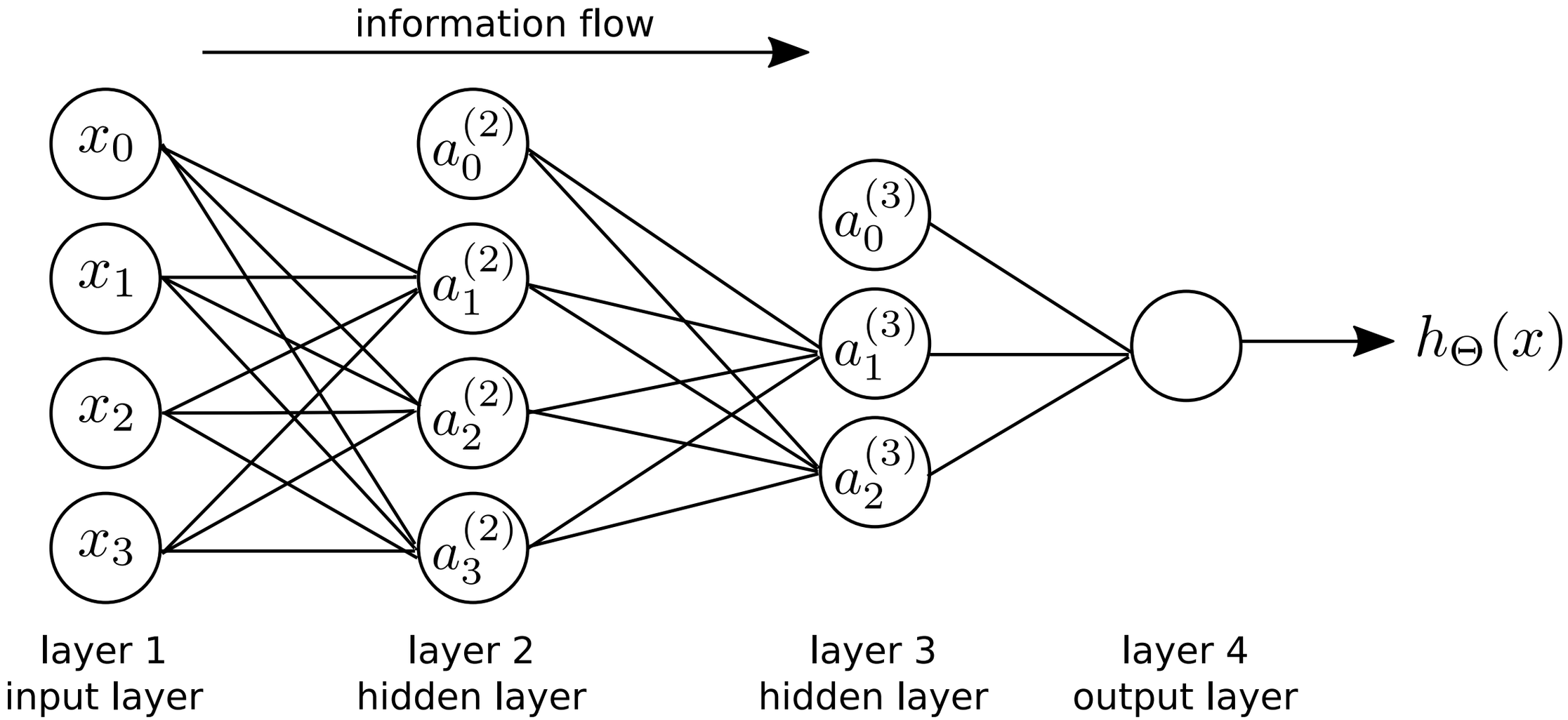
$$\begin{aligned} z^{(j)} &= \Theta^{(j-1)} a^{(j-1)} \\ a^{(j)} &= g\left(z^{(j)}\right) \end{aligned}$$

- If the network has a total of n layers, then

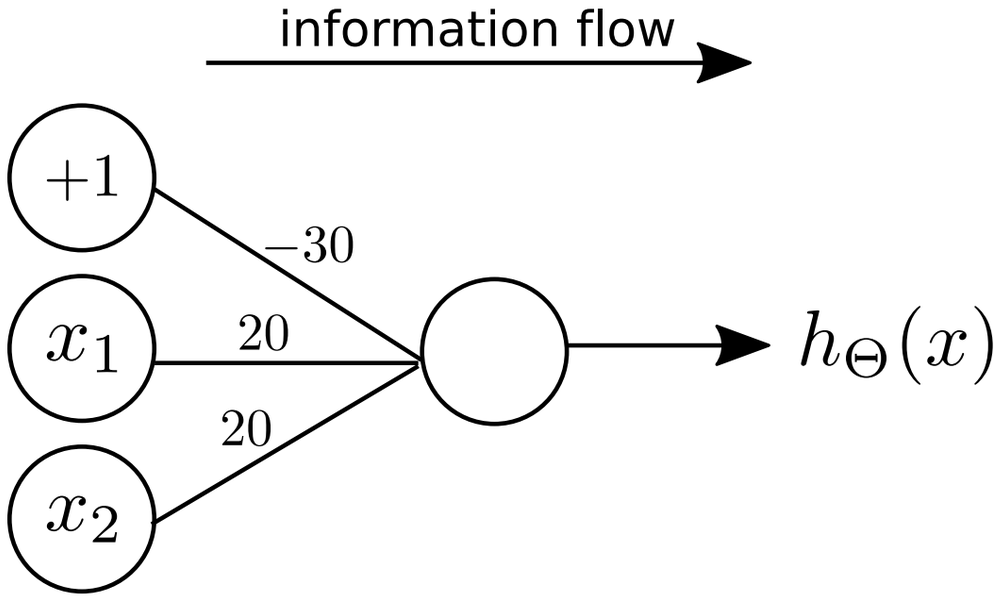
$$h_\Theta(x) = a^{(n)} = g\left(z^{(n)}\right)$$

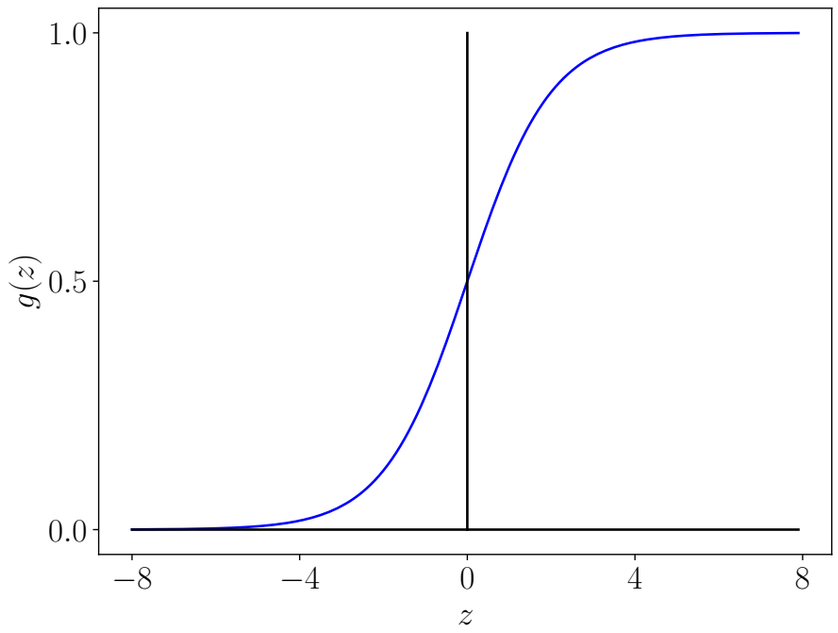
- Remark: The forward propagation in a neural network consists of a series of logistic regression.

Other ANN architecture



Example: logical AND function

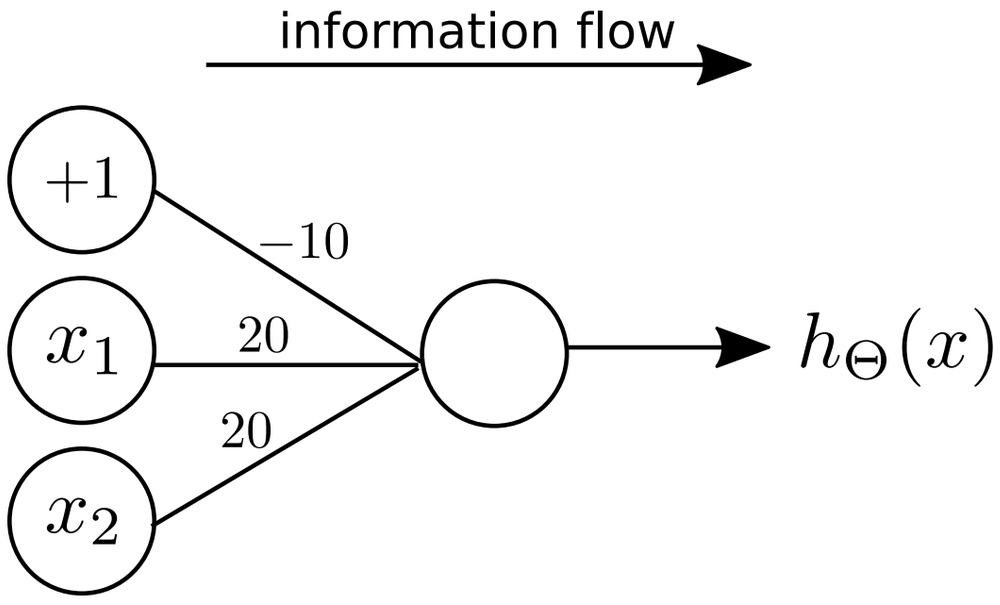




- $x_1, x_2 \in \{0, 1\}$
- $y = x_1 \text{ AND } x_2$
- $h_\Theta(x) = -30 + 20x_1 + 20x_2$

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	$g(+10) \approx 1$

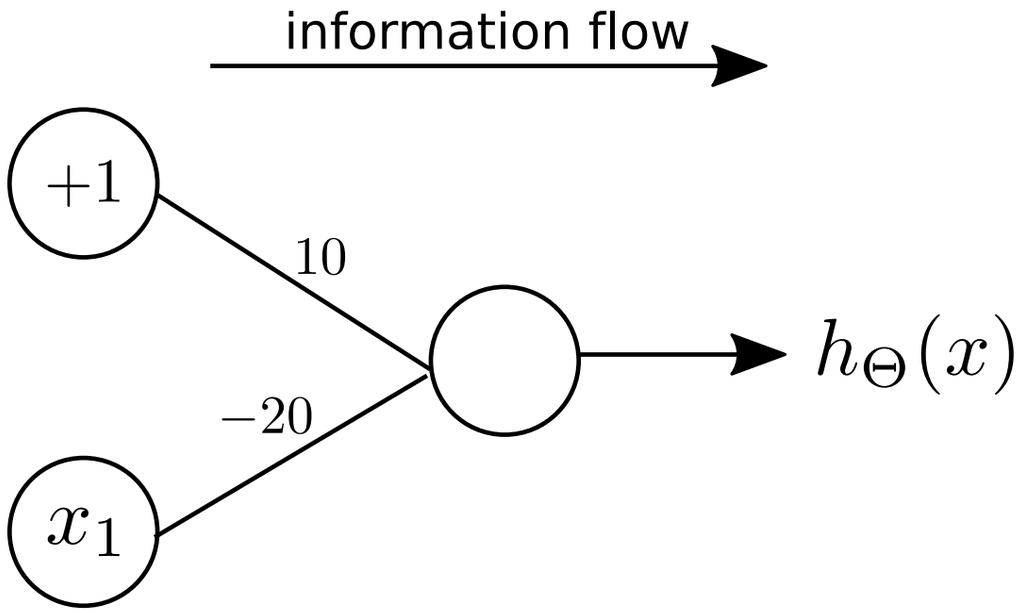
Example: logical OR function



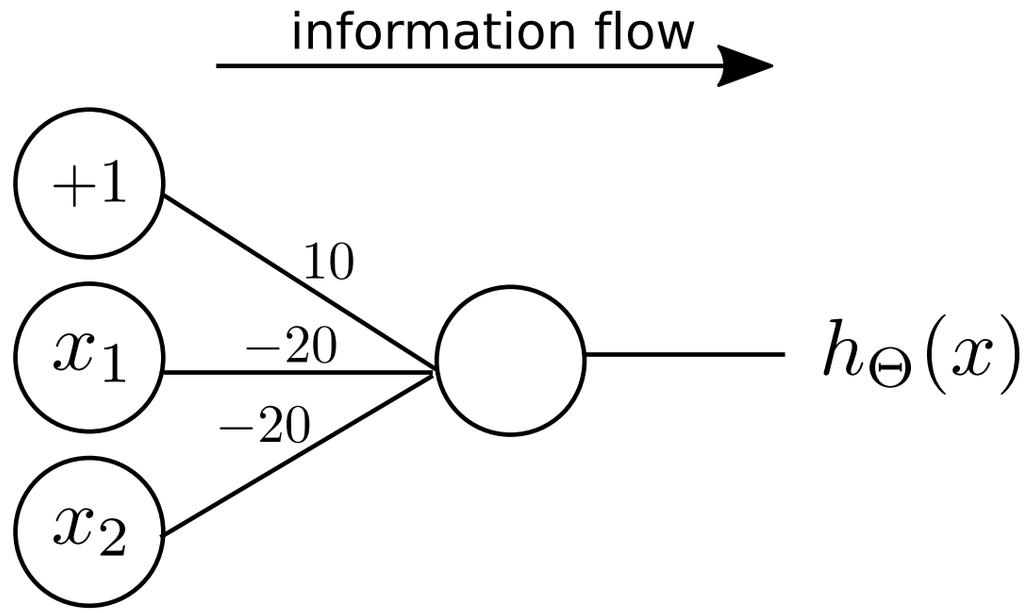
- $x_1, x_2 \in \{0, 1\}$
- $y = x_1 \text{ OR } x_2$
- $h_\Theta(x) = -10 + 20x_1 + 20x_2$

Example: Other logical functions

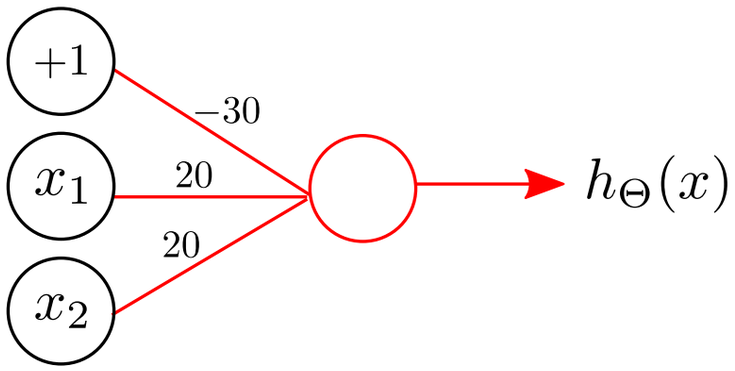
NOT x_1



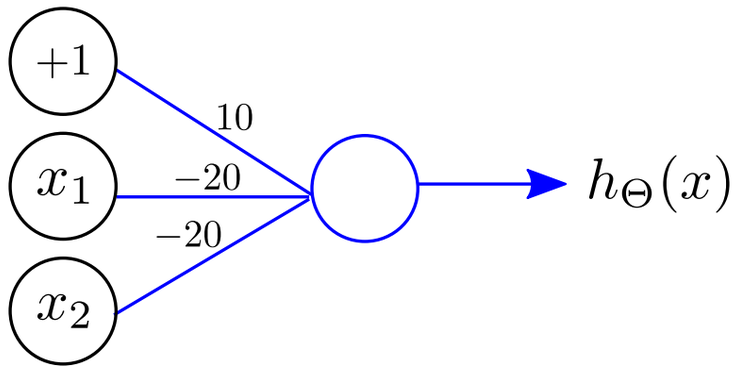
$(\texttt{NOT}\ x_1)\ \texttt{AND}\ (\texttt{NOT}\ x_2)$



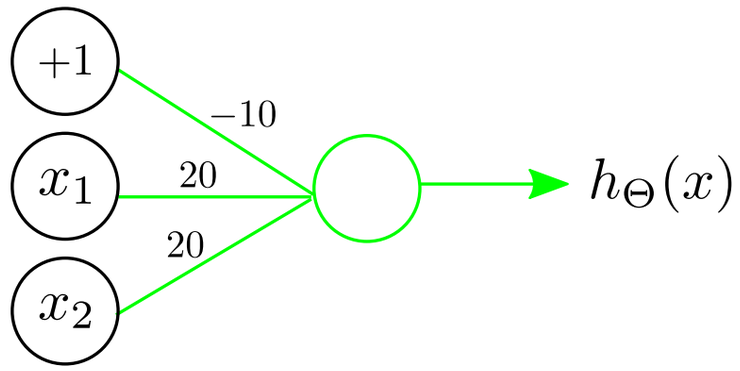
Example: logical XNOR



x_1 AND x_2

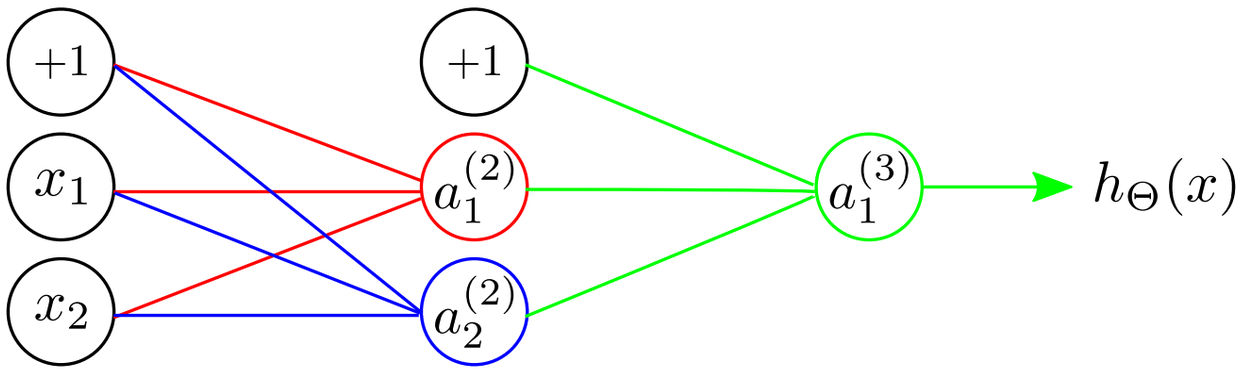


(NOT x_1) AND (NOT x_2)



x_1 OR x_2

Putting together:



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

with

$$\Theta^{(1)} = \begin{bmatrix} -30 & 20 & 20 \\ 10 & -20 & -20 \end{bmatrix}, \quad \Theta^{(2)} = \begin{bmatrix} -10 & 20 & 20 \end{bmatrix}$$

and

$$a^{(2)} = g\left(\Theta^{(1)} \cdot x\right), \quad a^{(3)} = g\left(\Theta^{(2)} \cdot a^{(2)}\right), \quad h_\Theta(x) = a^{(3)}$$

Neural Network Learning

Neural network for classification

- Notations

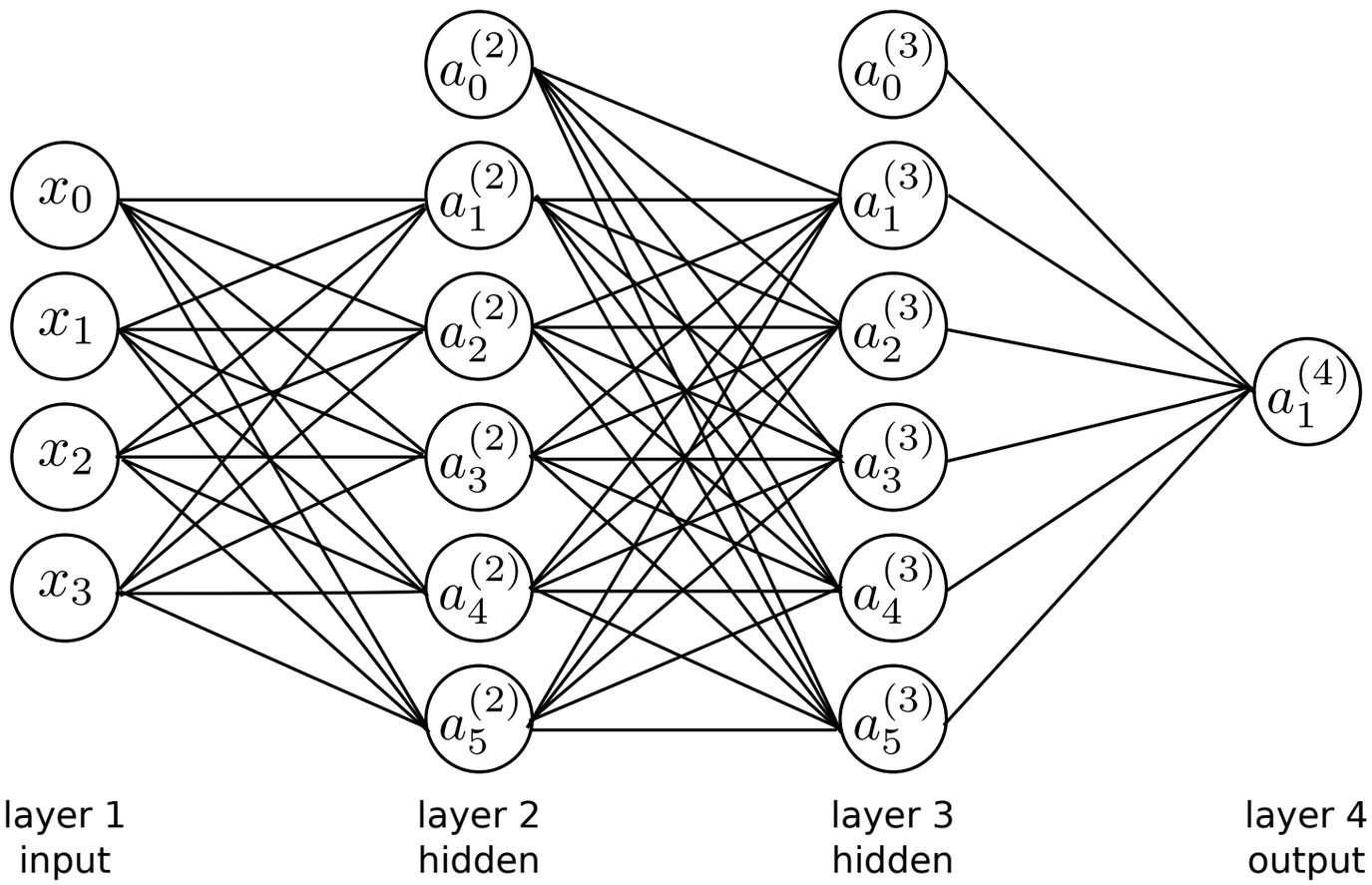
  Samples: $\{\left(x^{(1)}, y^{(1)}\right), \left(x^{(2)}, y^{(2)}\right), \cdots, \left(x^{(m)}, y^{(m)}\right)\}$

  L: total number of layers in the network

  s_l: number of units (not counting bias unit) in layer l

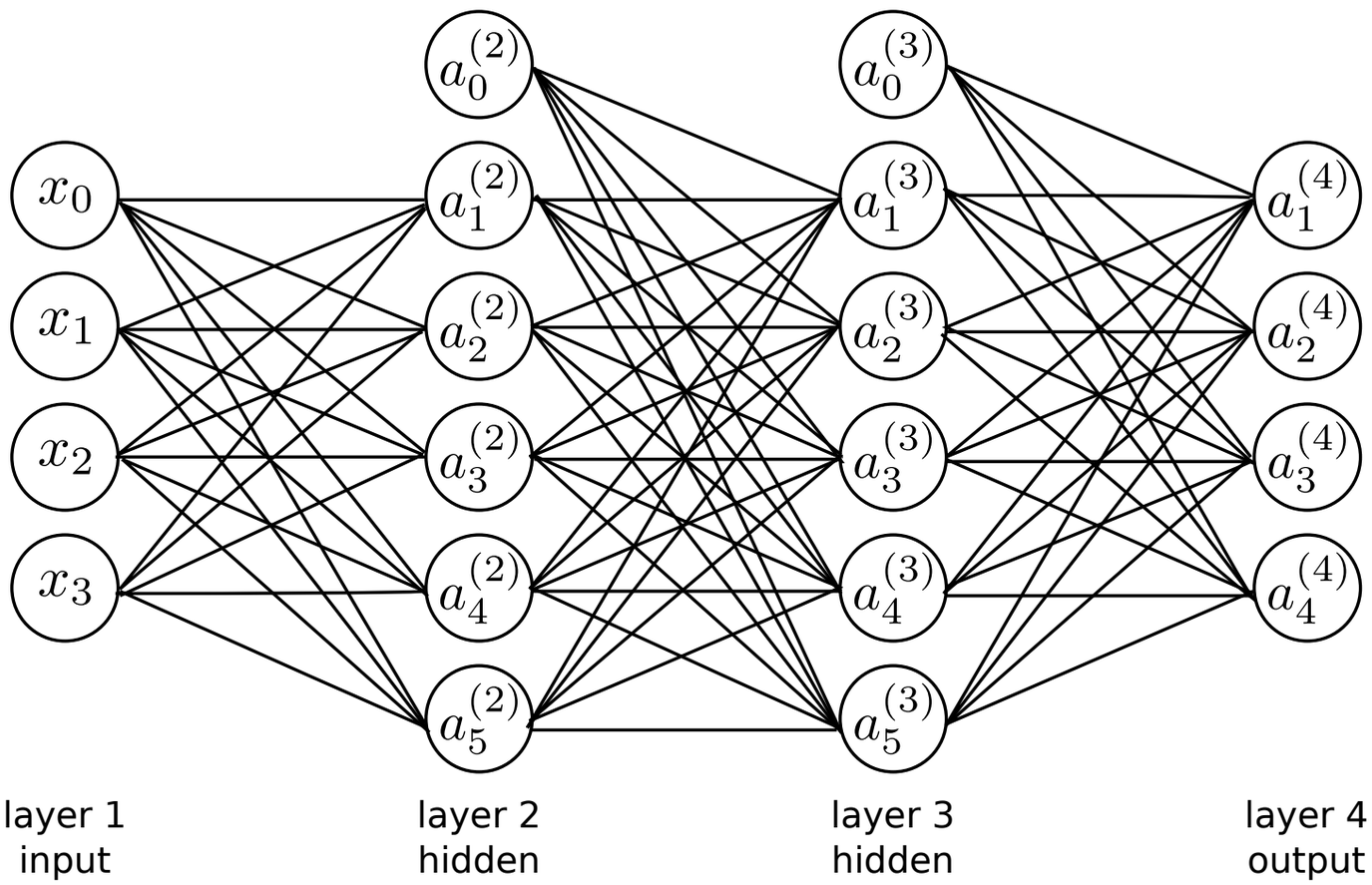
  K: number of classes

Binary classification



$$y \in \{0, 1\}$$
$$K = 1,\ h_\Theta(x) \in \mathbb{R}^K,\ \text{i.e. } h_\Theta(x) \in \mathbb{R}$$
$$L = 4,\ s_1 = 3, s_2 = 5, s_3 = 5, s_4 = 1$$

Multiclass classification



Multiclass classification

- This is for $K \geq 3$ classification.
- For the example on the previous slide, $K = 4$, $y \in \mathbb{R}^K$ and we use

$$y = \left\{ \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \right\}$$

to represent 4 classes, e.g., pedestrian, car, motorcycle, and truck, respectively.

$L = 4,\ s_1 = 3, s_2 = 5, s_3 = 5, s_4 = 4$

$h_\Theta(x) \in \mathbb{R}^4$

Cost function for classification

- Logistic regression

$$J(\theta) = -\frac{1}{m}\left[\sum_{i=1}^{m} y^{(i)} \log h_\theta(x^{(i)}) + (1 - y^{(i)}) \log\left(1 - h_\theta(x^{(i)})\right)\right]$$

- Neural network: Let

$$h_\Theta(x) \in \mathbb{R}^K \quad \text{and} \quad (h_\Theta(x))_k = k^{\text{th}} \text{ output}$$

then the cost function can be written as

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

Cost function for regression

- Use the mean squared error
- Linear regression

$$J(\theta) = \frac{1}{2m} \sum_{i=1}^{m} (\hat{y}_i - y_i)^2 = \frac{1}{2m} \sum_{i=1}^{m} \left(\theta^T x_i - y_i\right)^2$$

- Neural network

$$J(\theta) = \frac{1}{2m} \sum_{i=1}^{m} \sum_{k=1}^{K} \left[\left(h_\theta(x^{(i)}) \right)_k - y_k^{(i)} \right]^2$$

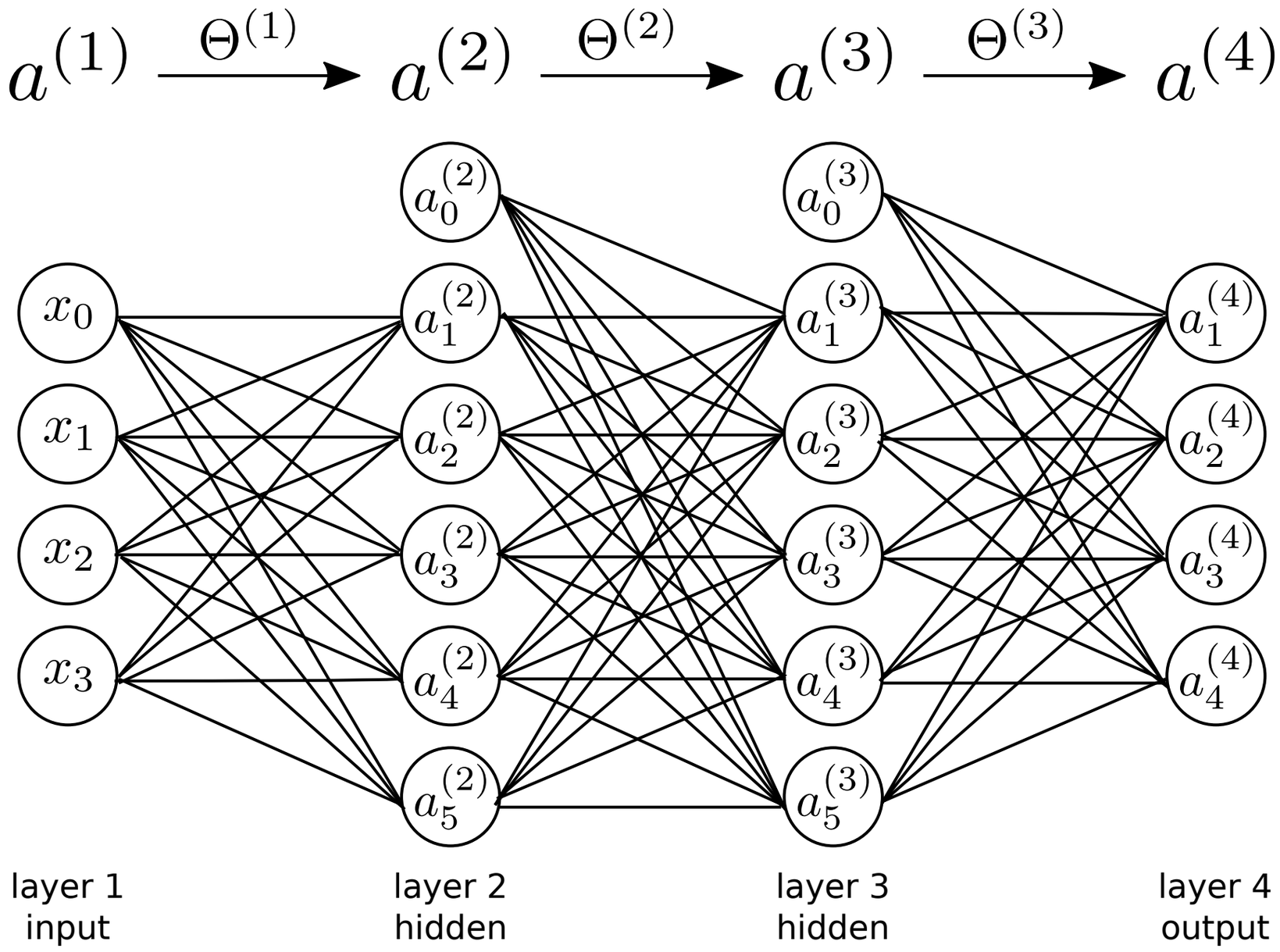
where K is the total number of output units.

Minimize the cost function

- Next we need to find Θ that would minimize $J(\Theta)$.
- The minimization requires code to compute
  - $J(\Theta)$
  - $\dfrac{\partial}{\partial \Theta_{ij}^{(l)}} J(\Theta)$

- Computation of $J(\Theta)$ is straightforward.
- The optimal Θ can be found by gradient descent.

Gradient computation: forward propagation



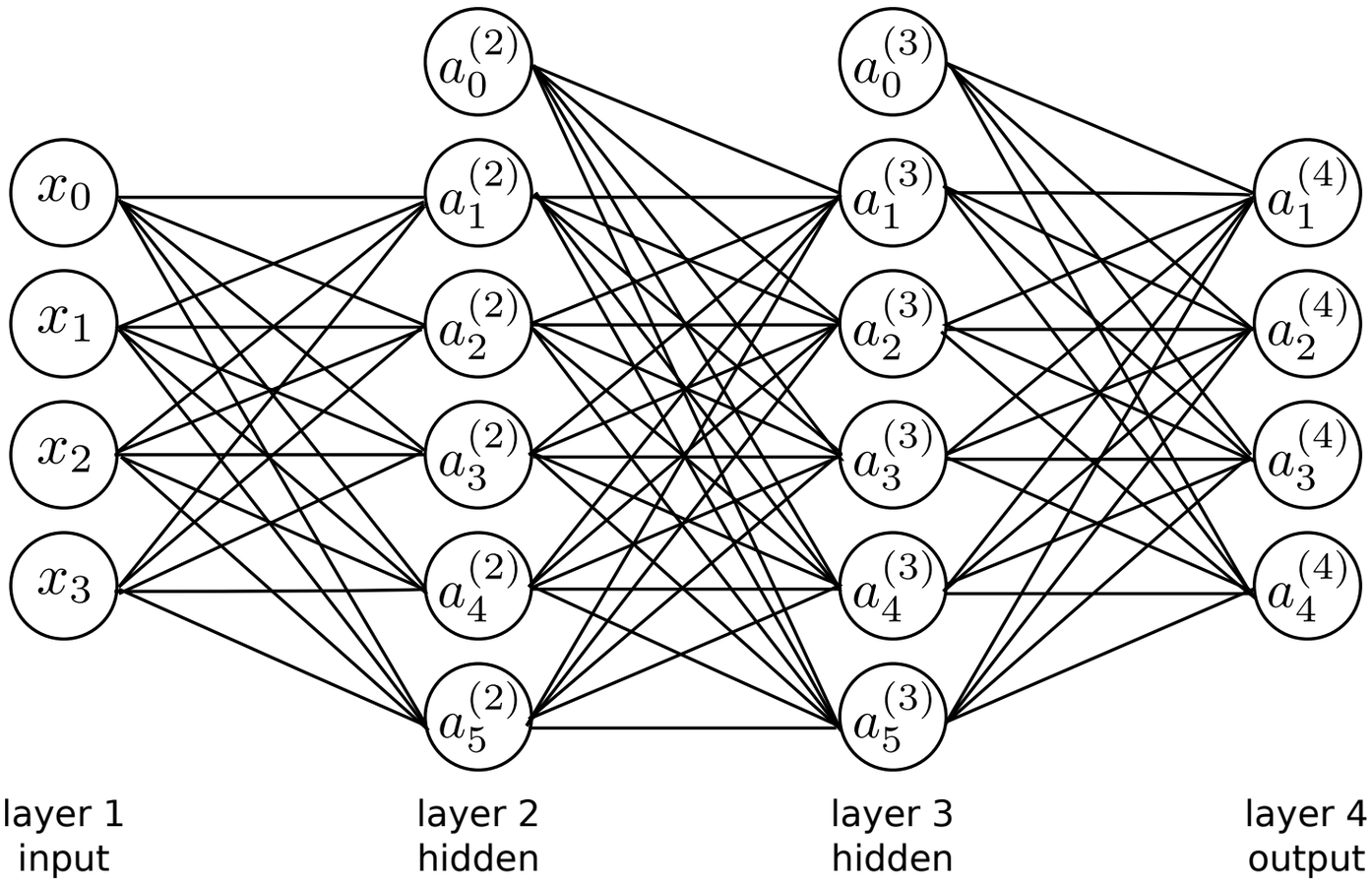
Gradient computation: forward propagation

- Given one training sample (x, y), forward propagation is done as follows for the neural network architecture shown on the previous slide:

$$
\begin{aligned}
a^{(1)} &= x \\
z^{(2)} &= \Theta^{(1)} a^{(1)} \\
a^{(2)} &= g\left(z^{(2)}\right) \\
&\quad \text{then add } a_0^{(2)} \text{ to } a^{(2)} \\
z^{(3)} &= \Theta^{(2)} a^{(2)} \\
a^{(3)} &= g\left(z^{(3)}\right) \\
&\quad \text{then add } a_0^{(3)} \text{to } a^{(3)} \\
z^{(4)} &= \Theta^{(3)} a^{(3)} \\
a^{(4)} &= g\left(z^{(4)}\right) = h_\Theta(x)
\end{aligned}
$$

Gradient computation: back propagation

$$\delta^{(2)} \xleftarrow[a^{(2)}]{\Theta^{(2)}} \delta^{(3)} \xleftarrow[a^{(3)}]{\Theta^{(3)}} \delta^{(4)}$$

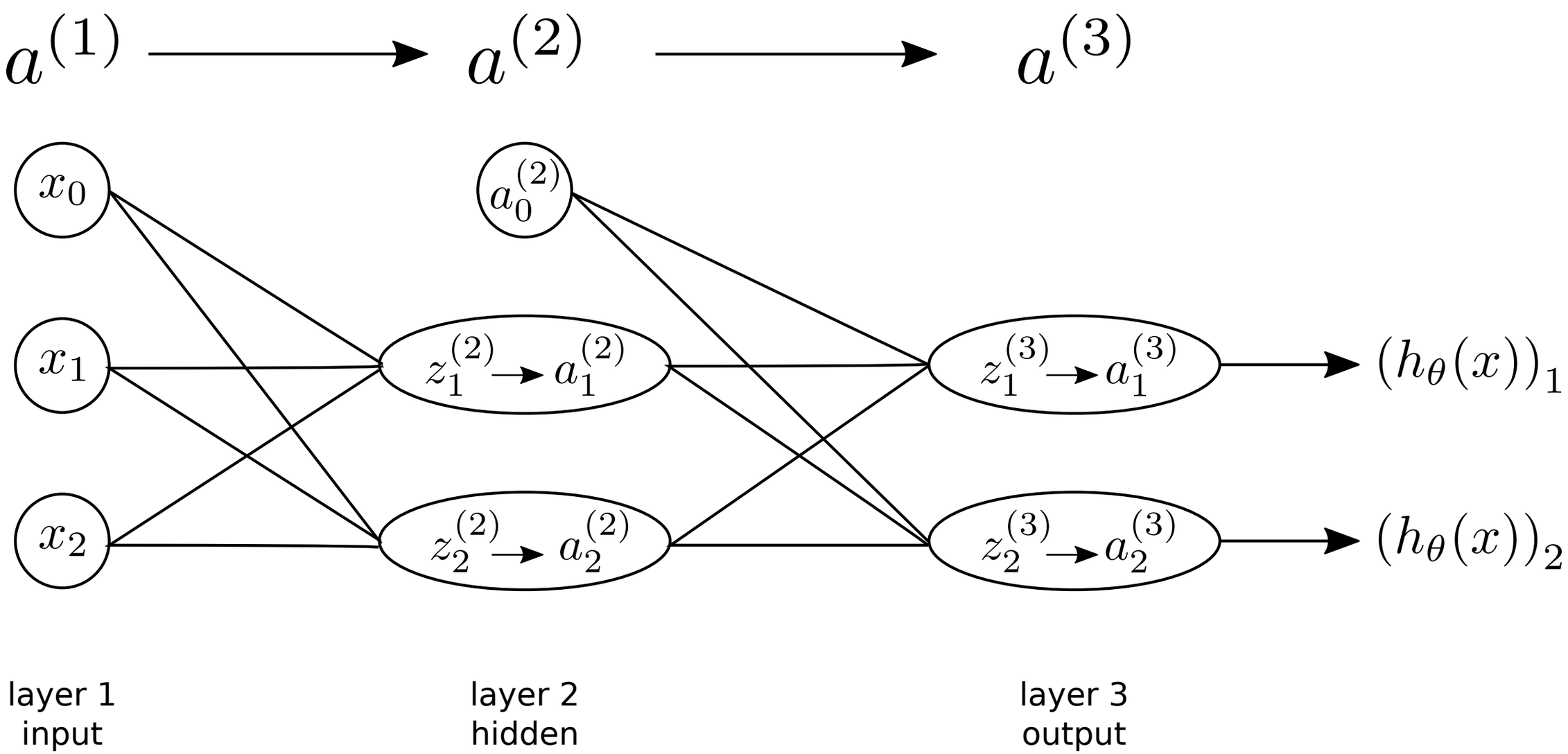


- Use back propagation to calculate

$$\frac{\partial}{\partial \Theta_{ij}^{(l)}} J(\Theta)$$

for all i, j and $l = 1, 2, \cdots, L-1$.

Example: forward propagation



Example: forward propagation

- From $a^{(1)}$ to $a^{(2)}$:

$$z^{(2)} = \begin{bmatrix} z_1^{(2)} \\ \\ z_2^{(2)} \end{bmatrix} = \begin{bmatrix} \Theta_{10}^{(1)} & \Theta_{11}^{(1)} & \Theta_{12}^{(1)} \\ \\ \Theta_{20}^{(1)} & \Theta_{21}^{(1)} & \Theta_{22}^{(1)} \end{bmatrix} \begin{bmatrix} a_0^{(1)} \\ \\ a_1^{(1)} \\ \\ a_2^{(1)} \end{bmatrix}, \quad a^{(2)} = \begin{bmatrix} a_1^{(2)} \\ \\ a_2^{(2)} \end{bmatrix} = g\left(z^{(2)}\right)$$

- From $a^{(2)}$ to $a^{(3)}$:

$$z^{(3)} = \begin{bmatrix} z_1^{(3)} \\ \\ z_2^{(3)} \end{bmatrix} = \begin{bmatrix} \Theta_{10}^{(2)} & \Theta_{11}^{(2)} & \Theta_{12}^{(2)} \\ \\ \Theta_{20}^{(2)} & \Theta_{21}^{(2)} & \Theta_{22}^{(2)} \end{bmatrix} \begin{bmatrix} a_0^{(2)} \\ \\ a_1^{(2)} \\ \\ a_2^{(2)} \end{bmatrix}, \quad a^{(3)} = \begin{bmatrix} a_1^{(3)} \\ \\ a_2^{(3)} \end{bmatrix} = g\left(z^{(3)}\right)$$

- Total cost:

$$J(\Theta) = \frac{1}{2}\left[\left(y_1 - a_1^{(3)}\right)^2 + \left(y_2 - a_2^{(3)}\right)^2\right]$$

Example: back propagation

- The goal is to calculate $\frac{\partial J\Theta}{\partial \Theta_{ij}^{(l)}}$ for $l = 1, 2$. We will first compute these partial derivatives for $l = 2$ and then for $l = 1$.

- $l = 2$:

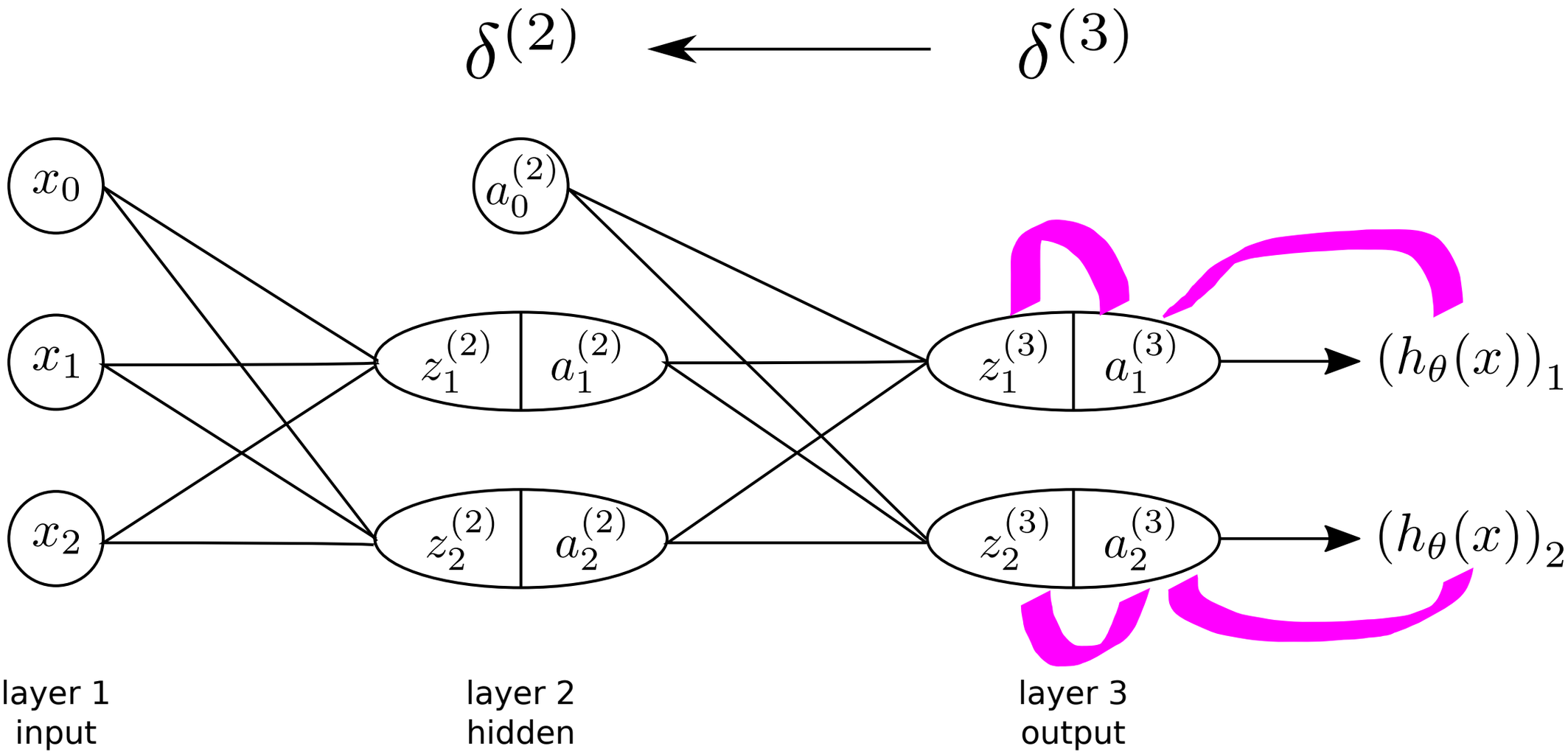
$$\left[\frac{\partial J(\Theta)}{\partial \Theta_{10}^{(2)}}, \frac{\partial J(\Theta)}{\partial \Theta_{11}^{(2)}}, \frac{\partial J(\Theta)}{\partial \Theta_{12}^{(2)}}\right] = \frac{\partial J(\Theta)}{\partial z_1^{(3)}} \cdot \left[a_0^{(2)}, a_1^{(2)}, a_2^{(2)}\right]$$

and

$$\left[\frac{\partial J(\Theta)}{\partial \Theta_{20}^{(2)}}, \frac{\partial J(\Theta)}{\partial \Theta_{21}^{(2)}}, \frac{\partial J(\Theta)}{\partial \Theta_{22}^{(2)}}\right] = \frac{\partial J(\Theta)}{\partial z_2^{(3)}} \cdot \left[a_0^{(2)}, a_1^{(2)}, a_2^{(2)}\right]$$

where $\frac{\partial J(\Theta)}{\partial z_1^{(3)}} \in \mathbb{R}$ and $\frac{\partial J(\Theta)}{\partial z_2^{(3)}} \in \mathbb{R}$.

Example: back propagation



Example: back propagation

- By the chain rule in calculus, we have

$$\frac{\partial J(\Theta)}{\partial z_1^{(3)}} = \frac{\partial J(\Theta)}{\partial a_1^{(3)}} \cdot \frac{\partial a_1^{(3)}}{\partial z_1^{(3)}} = \left(a_1^{(3)} - y_1\right) \cdot \left[a_1^{(3)} \cdot \left(1 - a_1^{(3)}\right)\right]$$

  with

$$\frac{\partial J(\Theta)}{\partial a_1^{(3)}} = a_1^{(3)} - y_1$$

  for the mean squared error cost function and

$$\frac{\partial a_1^{(3)}}{\partial z_1^{(3)}} = a_1^{(3)} \cdot \left(1 - a_1^{(3)}\right)$$

  for the logistic activation function.

- Similarly,

$$\frac{\partial J(\Theta)}{\partial z_2^{(3)}} = \frac{\partial J(\Theta)}{\partial a_2^{(3)}} \cdot \frac{\partial a_2^{(3)}}{\partial z_2^{(3)}} = \left(a_2^{(3)} - y_2\right) \cdot \left[a_2^{(3)} \cdot \left(1 - a_2^{(3)}\right)\right]$$

Example: back propagation

- Define

$$\delta^{(3)} := \begin{bmatrix} \dfrac{\partial J(\Theta)}{\partial z_1^{(3)}} \\ \\ \dfrac{\partial J(\Theta)}{\partial z_2^{(3)}} \end{bmatrix} = \begin{bmatrix} \left(a_1^{(3)} - y_1\right) \cdot \left[a_1^{(3)} \cdot \left(1 - a_1^{(3)}\right)\right] \\ \\ \left(a_2^{(3)} - y_2\right) \cdot \left[a_2^{(3)} \cdot \left(1 - a_2^{(3)}\right)\right] \end{bmatrix}$$

- Then

$$\begin{bmatrix} \dfrac{\partial J(\Theta)}{\partial \Theta_{10}^{(2)}}, \dfrac{\partial J(\Theta)}{\partial \Theta_{11}^{(2)}}, \dfrac{\partial J(\Theta)}{\partial \Theta_{12}^{(2)}} \\ \\ \dfrac{\partial J(\Theta)}{\partial \Theta_{20}^{(2)}}, \dfrac{\partial J(\Theta)}{\partial \Theta_{21}^{(2)}}, \dfrac{\partial J(\Theta)}{\partial \Theta_{22}^{(2)}} \end{bmatrix} = \delta^{(3)} \cdot \left[a_0^{(2)}, a_1^{(2)}, a_2^{(2)}\right]$$

Example: back propagation

- Calculate $\dfrac{\partial J\Theta}{\partial \Theta_{ij}^{(l)}}$ for $l = 1$:

$$\left[\frac{\partial J(\Theta)}{\partial \Theta_{10}^{(1)}}, \frac{\partial J(\Theta)}{\partial \Theta_{11}^{(1)}}, \frac{\partial J(\Theta)}{\partial \Theta_{12}^{(1)}}\right] = \frac{\partial J(\Theta)}{\partial z_1^{(2)}} \cdot \left[a_0^{(1)}, a_1^{(1)}, a_2^{(1)}\right]$$

and

$$\left[\frac{\partial J(\Theta)}{\partial \Theta_{20}^{(1)}}, \frac{\partial J(\Theta)}{\partial \Theta_{21}^{(1)}}, \frac{\partial J(\Theta)}{\partial \Theta_{22}^{(1)}}\right] = \frac{\partial J(\Theta)}{\partial z_2^{(2)}} \cdot \left[a_0^{(1)}, a_1^{(1)}, a_2^{(1)}\right]$$

where $\dfrac{\partial J(\Theta)}{\partial z_1^{(2)}} \in \mathbb{R}$ and $\dfrac{\partial J(\Theta)}{\partial z_2^{(2)}} \in \mathbb{R}$.

Example: back propagation

- The total cost is the sum of the costs for unit 1 and unit 2 in the output layer:

$$J(\Theta) = \frac{1}{2m} \sum_{i=1}^{m} \sum_{k=1}^{K} \left[\left(h_\Theta(x^{(i)}) \right)_k - y_k^{(i)} \right]^2$$

- For a single sample $\{(x, y)\}$, the total cost can be written as

$$J(\Theta) = \frac{1}{2} \left[\left(a_1^{(3)} - y_1 \right)^2 + \left(a_2^{(3)} - y_2 \right)^2 \right]$$

- Define $J_1(\Theta)$ and $J_2(\Theta)$ as the cost for unit 1 and unit 2 in the output layer, respectively, i.e.

$$J_1(\Theta) = \frac{1}{2} \left(a_1^{(3)} - y_1 \right)^2, \quad J_2(\Theta) = \frac{1}{2} \left(a_2^{(3)} - y_2 \right)^2$$

- Then

$$J(\Theta) = J_1(\Theta) + J_2(\Theta)$$

Example: back propagation

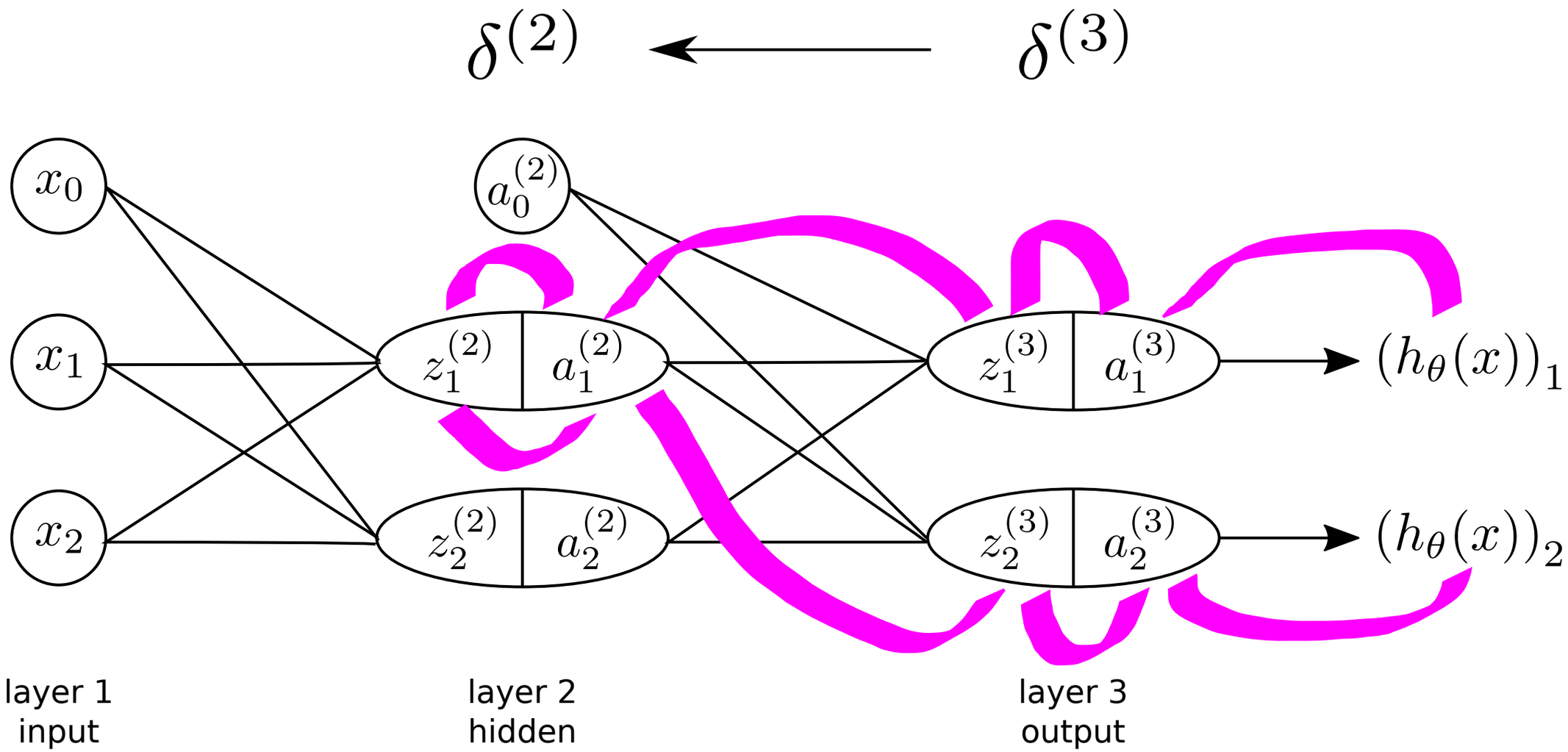
- Now we compute $\dfrac{\partial J(\Theta)}{\partial z_1^{(2)}}$.

$$\frac{\partial J(\Theta)}{\partial z_1^{(2)}} = \frac{\partial J_1(\Theta)}{\partial z_1^{(2)}} + \frac{\partial J_2(\Theta)}{\partial z_1^{(2)}}$$

with

$$\begin{aligned}
\frac{\partial J_1(\Theta)}{\partial z_1^{(2)}} &= \frac{\partial J_1(\Theta)}{\partial a_1^{(3)}} \cdot \frac{\partial a_1^{(3)}}{\partial z_1^{(3)}} \cdot \frac{\partial z_1^{(3)}}{\partial a_1^{(2)}} \cdot \frac{\partial a_1^{(2)}}{\partial z_1^{(2)}} \\
&= \left(a_1^{(3)} - y_1\right) \cdot \left[a_1^{(3)}\left(1 - a_1^{(3)}\right)\right] \cdot \Theta_{11}^{(2)} \cdot \left[a_1^{(2)}\left(1 - a_1^{(2)}\right)\right] \\
\frac{\partial J_2(\Theta)}{\partial z_1^{(2)}} &= \frac{\partial J_2(\Theta)}{\partial a_2^{(3)}} \cdot \frac{\partial a_2^{(3)}}{\partial z_2^{(3)}} \cdot \frac{\partial z_2^{(3)}}{\partial a_1^{(2)}} \cdot \frac{\partial a_1^{(2)}}{\partial z_1^{(2)}} \\
&= \left(a_2^{(3)} - y_2\right) \cdot \left[a_2^{(3)}\left(1 - a_2^{(3)}\right)\right] \cdot \Theta_{21}^{(2)} \cdot \left[a_1^{(2)}\left(1 - a_1^{(2)}\right)\right]
\end{aligned}$$

Example: back propagation



Example: back propagation

- Similarly for $\dfrac{\partial J(\Theta)}{\partial z_2^{(2)}}$, we have

$$\frac{\partial J(\Theta)}{\partial z_2^{(2)}} = \frac{\partial J_1(\Theta)}{\partial z_2^{(2)}} + \frac{\partial J_2(\Theta)}{\partial z_2^{(2)}}$$

with

$$\begin{aligned}
\frac{\partial J_1(\Theta)}{\partial z_2^{(2)}} &= \frac{\partial J_1(\Theta)}{\partial a_1^{(3)}} \cdot \frac{\partial a_1^{(3)}}{\partial z_1^{(3)}} \cdot \frac{\partial z_1^{(3)}}{\partial a_2^{(2)}} \cdot \frac{\partial a_2^{(2)}}{\partial z_2^{(2)}} \\
&= \left(a_1^{(3)} - y_1\right) \cdot \left[a_1^{(3)}\left(1 - a_1^{(3)}\right)\right] \cdot \Theta_{12}^{(2)} \cdot \left[a_2^{(2)}\left(1 - a_2^{(2)}\right)\right]
\end{aligned}$$

$$\begin{aligned}
\frac{\partial J_2(\Theta)}{\partial z_2^{(2)}} &= \frac{\partial J_2(\Theta)}{\partial a_2^{(3)}} \cdot \frac{\partial a_2^{(3)}}{\partial z_2^{(3)}} \cdot \frac{\partial z_2^{(3)}}{\partial a_2^{(2)}} \cdot \frac{\partial a_2^{(2)}}{\partial z_2^{(2)}} \\
&= \left(a_2^{(3)} - y_2\right) \cdot \left[a_2^{(3)}\left(1 - a_2^{(3)}\right)\right] \cdot \Theta_{22}^{(2)} \cdot \left[a_2^{(2)}\left(1 - a_2^{(2)}\right)\right]
\end{aligned}$$

Example: back propagation

- Define

$$\delta^{(2)} := \begin{bmatrix} \dfrac{\partial J(\Theta)}{\partial z_1^{(2)}} \\ \\ \dfrac{\partial J(\Theta)}{\partial z_2^{(2)}} \end{bmatrix}$$

- Then

$$\begin{bmatrix} \dfrac{\partial J(\Theta)}{\partial \Theta_{10}^{(1)}}, \dfrac{\partial J(\Theta)}{\partial \Theta_{11}^{(1)}}, \dfrac{\partial J(\Theta)}{\partial \Theta_{12}^{(1)}} \\ \\ \dfrac{\partial J(\Theta)}{\partial \Theta_{20}^{(1)}}, \dfrac{\partial J(\Theta)}{\partial \Theta_{21}^{(1)}}, \dfrac{\partial J(\Theta)}{\partial \Theta_{22}^{(1)}} \end{bmatrix} = \delta^{(2)} \cdot \left[a_0^{(1)}, a_1^{(1)}, a_2^{(1)} \right]$$

- Recall that $a_0^{(1)} = a_0^{(2)} = 1$.
- This completes one iteration of the forward and back propagation.
- Continue this process until $\Theta_{ij}^{(l)}$ converge for $l = 1, 2$.

Example: back propagation

- How do we streamline the process of back propagation from $\delta^{(j+1)}$ to $\delta^{(j)}$?

Example: back propagation

- Note that

$$\left[\frac{\partial J(\Theta)}{\partial a_1^{(2)}}, \frac{\partial J(\Theta)}{\partial a_2^{(2)}}\right] = \left[\left(a_1^{(3)} - y_1\right) \cdot \left[a_1^{(3)}\left(1 - a_1^{(3)}\right)\right], \left(a_2^{(3)} - y_2\right) \cdot \left[a_2^{(3)}\left(1 - a_2^{(3)}\right)\right]\right] \begin{bmatrix} \Theta_{11}^{(2)} & \Theta_{12}^{(2)} \\ \Theta_{21}^{(2)} & \Theta_{22}^{(2)} \end{bmatrix}$$

$$= \left[\delta_1^{(3)}, \delta_2^{(3)}\right] \begin{bmatrix} \Theta_{11}^{(2)} & \Theta_{12}^{(2)} \\ \Theta_{21}^{(2)} & \Theta_{22}^{(2)} \end{bmatrix} = \left(\delta^{(3)}\right)^T \begin{bmatrix} \Theta_{11}^{(2)} & \Theta_{12}^{(2)} \\ \Theta_{21}^{(2)} & \Theta_{22}^{(2)} \end{bmatrix}$$

and

$$\left(\delta^{(2)}\right)^T = \left[\frac{\partial J(\Theta)}{\partial z_1^{(2)}}, \frac{\partial J(\Theta)}{\partial z_2^{(2)}}\right] = \left[\frac{\partial J(\Theta)}{\partial a_1^{(2)}}, \frac{\partial J(\Theta)}{\partial a_2^{(2)}}\right] \begin{bmatrix} \frac{\partial a_1^{(2)}}{\partial z_1^{(2)}} & 0 \\ 0 & \frac{\partial a_2^{(2)}}{\partial z_2^{(2)}} \end{bmatrix}$$

$$= \left(\delta^{(3)}\right)^T \begin{bmatrix} \Theta_{11}^{(2)} & \Theta_{12}^{(2)} \\ \Theta_{21}^{(2)} & \Theta_{22}^{(2)} \end{bmatrix} \begin{bmatrix} \frac{\partial a_1^{(2)}}{\partial z_1^{(2)}} & 0 \\ 0 & \frac{\partial a_2^{(2)}}{\partial z_2^{(2)}} \end{bmatrix}$$

Back propagation

- In general,

$$\left(\delta^{(j)}\right)^T = \left(\delta^{(j+1)}\right)^T \Theta^{(j)} \begin{bmatrix} \frac{\partial a_1^{(j)}}{\partial z_1^{(j)}} & 0 & \cdots & 0 \\ 0 & \frac{\partial a_2^{(j)}}{\partial z_2^{(j)}} & \cdots & 0 \\ \vdots & \vdots & \ldots & \vdots \\ 0 & 0 & \cdots & \frac{\partial a_{s_j}^{(j)}}{\partial z_{s_j}^{(j)}} \end{bmatrix}$$

where $\left(\delta^{(j)}\right)^T$ is of size $1 \times s_j$, $\left(\delta^{(j+1)}\right)^T$ is of size $1 \times s_{(j+1)}$, $\Theta^{(j)}$ is of size $s_{(j+1)} \times s_j$, and the diagonal matrix is of size $s_j \times s_j$.

Back propagation

- Back propagation starts with computing $\delta^{(L)}$.
- If neural network is used for regression, the total cost is

$$J(\Theta) = \frac{1}{2m} \sum_{i=1}^{m} \sum_{k=1}^{K} \left[\left(h_\theta(x^{(i)}) \right)_k - y_k^{(i)} \right]^2$$

  where m is the total number of samples and K is the total number of output units.
- For a specific sample, the associated cost is

$$J(\Theta) = \frac{1}{2} \sum_{k=1}^{K} \left[(h_\theta(x))_k - y_k \right]^2 = \frac{1}{2} \sum_{k=1}^{K} \left[a_k^{(L)} - y_k \right]^2$$

Back propagation

- $\delta^{(L)}$ can be computed as

$$\delta^{(L)} = \begin{bmatrix} \frac{\partial J(\Theta)}{\partial z_1^{(L)}} \\ \frac{\partial J(\Theta)}{\partial z_2^{(L)}} \\ \vdots \\ \frac{\partial J(\Theta)}{\partial z_{s_L}^{(L)}} \end{bmatrix} = \begin{bmatrix} \frac{\partial J(\Theta)}{\partial a_1^{(L)}} \cdot \frac{\partial a_1^{(L)})}{\partial z_1^{(L)}} \\ \frac{\partial J(\Theta)}{\partial a_2^{(L)}} \cdot \frac{\partial a_2^{(L)})}{\partial z_2^{(L)}} \\ \vdots \\ \frac{\partial J(\Theta)}{\partial a_{s_L}^{(L)}} \cdot \frac{\partial a_{s_L}^{(L)})}{\partial z_{s_L}^{(L)}} \end{bmatrix} = \begin{bmatrix} \frac{\partial J(\Theta)}{\partial a_1^{(L)}} \\ \frac{\partial J(\Theta)}{\partial a_2^{(L)}} \\ \vdots \\ \frac{\partial J(\Theta)}{\partial a_{s_L}^{(L)}} \end{bmatrix} \cdot \begin{bmatrix} \frac{\partial a_1^{(L)}}{\partial z_1^{(L)}} & 0 & \cdots & 0 \\ 0 & \frac{\partial a_2^{(L)}}{\partial z_2^{(L)}} & \cdots & 0 \\ \vdots & \vdots & \cdots & \vdots \\ 0 & 0 & \cdots & \frac{\partial a_{s_L}^{(L)}}{\partial z_{s_L}^{(L)}} \end{bmatrix}$$

$$= \begin{bmatrix} \left(a_1^{(L)} - y_1\right) \\ \left(a_2^{(L)} - y_2\right) \\ \vdots \\ \left(a_{s_L}^{(L)} - y_{s_L}\right) \end{bmatrix} \cdot \begin{bmatrix} a_1^{(L)}\left(1 - a_1^{(L)}\right) & 0 & \cdots & 0 \\ 0 & a_2^{(L)}\left(1 - a_2^{(L)}\right) & \cdots & 0 \\ \vdots & & & \\ 0 & 0 & \cdots & a_{s_L}^{(L)}\left(1 - a_{s_L}^{(L)}\right) \end{bmatrix}$$

Back propagation

- If a neural network is used for classification, the total cost is

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

  where m is the total number of samples and K is the total number of output units.

- For a specific sample, the associated cost is

$$\begin{aligned} J(\Theta) &= -\sum_{k=1}^{K}\left[y_k\log\left(h_\Theta(x)\right)_k + (1-y_k)\log\left(1-\left(h_\Theta(x)\right)_k\right)\right] \\ &= -\sum_{k=1}^{K}\left[y_k\log a_k^{(L)} + (1-y_k)\log\left(1-a_k^{(L)}\right)\right] \end{aligned}$$

Back propagation

- $\delta^{(L)}$ can be computed as

$$\delta^{(L)} = \begin{bmatrix} \frac{\partial J(\Theta)}{\partial z_1^{(L)}} \\ \frac{\partial J(\Theta)}{\partial z_2^{(L)}} \\ \vdots \\ \frac{\partial J(\Theta)}{\partial z_{s_L}^{(L)}} \end{bmatrix} = \begin{bmatrix} \frac{\partial J(\Theta)}{\partial a_1^{(L)}} \cdot \frac{\partial a_1^{(L)})}{\partial z_1^{(L)}} \\ \frac{\partial J(\Theta)}{\partial a_2^{(L)}} \cdot \frac{\partial a_2^{(L)})}{\partial z_2^{(L)}} \\ \vdots \\ \frac{\partial J(\Theta)}{\partial a_{s_L}^{(L)}} \cdot \frac{\partial a_{s_L}^{(L)})}{\partial z_{s_L}^{(L)}} \end{bmatrix} = \begin{bmatrix} \frac{\partial J(\Theta)}{\partial a_1^{(L)}} \\ \frac{\partial J(\Theta)}{\partial a_2^{(L)}} \\ \vdots \\ \frac{\partial J(\Theta)}{\partial a_{s_L}^{(L)}} \end{bmatrix} \cdot \begin{bmatrix} \frac{\partial a_1^{(L)}}{\partial z_1^{(L)}} & 0 & \cdots & 0 \\ 0 & \frac{\partial a_2^{(L)}}{\partial z_2^{(L)}} & \cdots & 0 \\ \vdots & \vdots & \cdots & \vdots \\ 0 & 0 & \cdots & \frac{\partial a_{s_L}^{(L)}}{\partial z_{s_L}^{(L)}} \end{bmatrix}$$

$$= \begin{bmatrix} \frac{\left(a_1^{(L)} - y_1\right)}{a_1^{(L)}\left(1 - a_1^{(L)}\right)} \\ \frac{\left(a_2^{(L)} - y_2\right)}{a_2^{(L)}\left(1 - a_2^{(L)}\right)} \\ \vdots \\ \frac{\left(a_{s_L}^{(L)} - y_{s_L}\right)}{a_{s_L}^{(L)}\left(1 - a_{s_L}^{(L)}\right)} \end{bmatrix} \cdot \begin{bmatrix} a_1^{(L)}\left(1 - a_1^{(L)}\right) & 0 & \cdots & 0 \\ 0 & a_2^{(L)}\left(1 - a_2^{(L)}\right) & \cdots & 0 \\ \vdots & & & \\ 0 & 0 & \cdots & a_{s_L}^{(L)}\left(1 - a_{s_L}^{(L)}\right) \end{bmatrix} = \begin{bmatrix} a_1^{(L)} - y_1 \\ a_2^{(L)} - y_2 \\ \vdots \\ a_{s_L}^{(L)} - y_{s_L} \end{bmatrix}$$

Gradient computation for one iteration

- Given training samples $\left\{\left(x^{(1)}, y^{(1)}\right), \left(x^{(2)}, y^{(2)}\right), \cdots, \left(x^{(m)}, y^{(m)}\right)\right\}$
- Denote $D_{i,j}^{(l)} := \dfrac{\partial J(\Theta)}{\partial \Theta_{i,j}^{(l)}}$. Gradient for one iteration is computed as follows:

Set $\Delta_{i,j}^{(l)} = 0$ for all l, i, j.

for $i = 1$ to m {

Set $a^{(1)} = x^{(i)}$ and $a^{(L)} = y^{(i)}$.

Perform forward propagation to compute $a^{(l)}$ for $l = 2, 3, \cdots, L$.

Compute $\delta^{(L)}$ according to the total cost function.

Compute $\delta^{(L-1)}, \delta^{(L-2)}, \cdots, \delta^{(2)}$ using back propagation.

Compute $\Delta_{i,j}^{(l)} := \Delta_{i,j}^{(l)} + a_j^{(l)} \delta_i^{(l+1)}$ OR $\Delta^{(l)} := \Delta^{(l)} + \delta^{(l+1)} \left(a^{(l)}\right)^T$

}

$D^{(l)} := \dfrac{1}{m} \Delta^{(l)}$ and $\dfrac{\partial J(\Theta)}{\partial \Theta_{ij}^{(l)}} = D_{ij}^{(l)}$

Overall gradient descent algorithm

- Given learning rate α and training samples $\left\{\left(x^{(1)}, y^{(1)}\right), \left(x^{(2)}, y^{(2)}\right), \cdots, \left(x^{(m)}, y^{(m)}\right)\right\}$
- Denote $D_{i,j}^{(l)} := \dfrac{\partial J(\Theta)}{\partial \Theta_{i,j}^{(l)}}$. Gradient descent proceeds as follows:

for index_iter $= 1$ to total_number_of_iterations {

Set $\Delta_{i,j}^{(l)} = 0$ for all l, i, j.

for $i = 1$ to m {

Set $a^{(1)} = x^{(i)}$ and $a^{(L)} = y^{(i)}$.

Perform forward propagation to compute $a^{(l)}$ for $l = 2, 3, \cdots, L$.

Compute $\delta^{(L)}$ according to the total cost function.

Compute $\delta^{(L-1)}, \delta^{(L-2)}, \cdots, \delta^{(2)}$ using back propagation.

Compute $\Delta_{i,j}^{(l)} := \Delta_{i,j}^{(l)} + a_j^{(l)} \delta_i^{(l+1)}$ OR $\Delta^{(l)} := \Delta^{(l)} + \delta^{(l+1)} \left(a^{(l)}\right)^T$

}

$D^{(l)} := \dfrac{1}{m} \Delta^{(l)}$ and $\dfrac{\partial J(\Theta)}{\partial \Theta_{ij}^{(l)}} = D_{ij}^{(l)}$

$\Theta^{(l)} := \Theta^{(l)} - \alpha D^{(l)}$

}

Thank you!