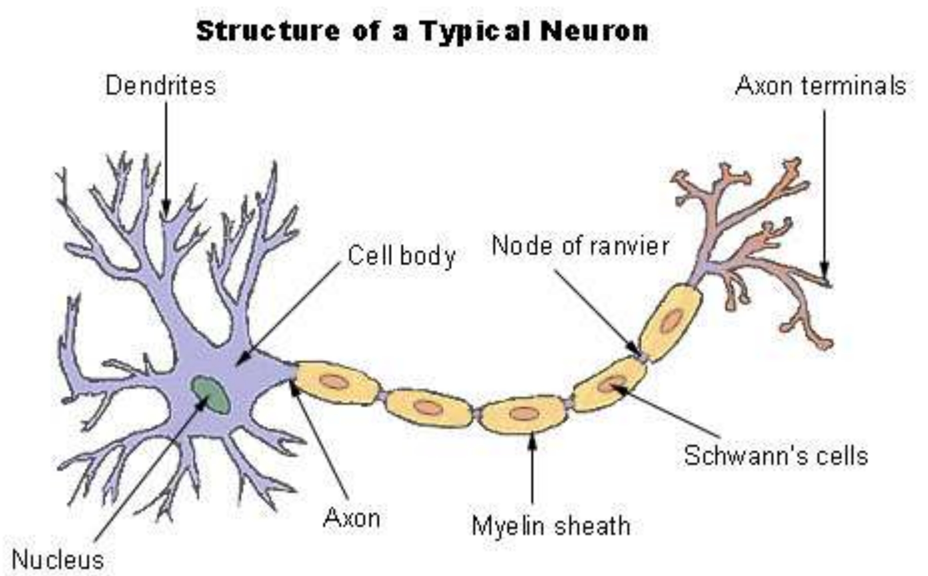
Neural Networks

The Perceptron

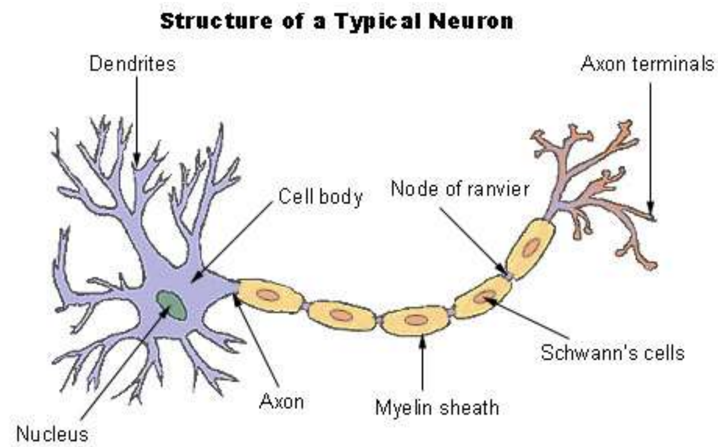
START
01

An Early Machine Learning Model

A Typical Neuron

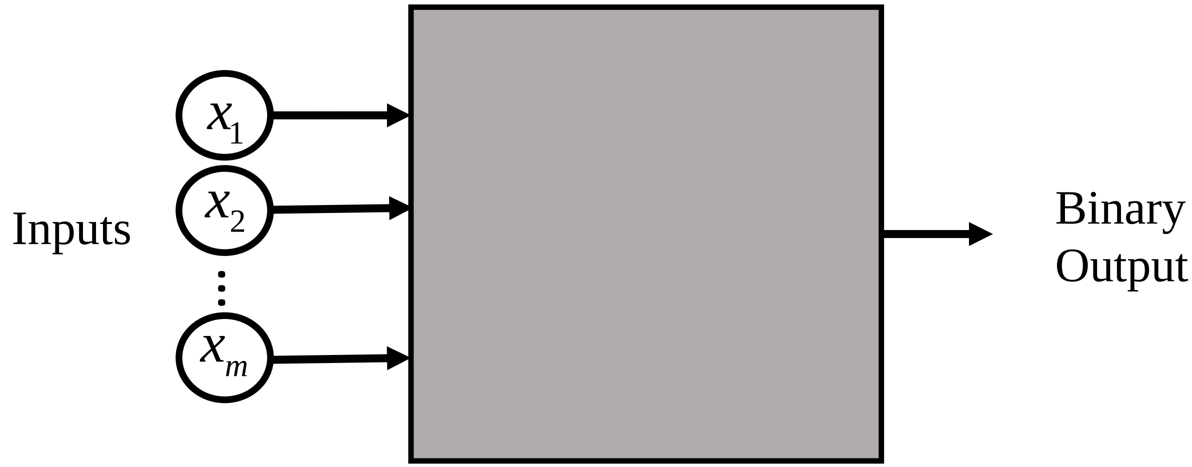


MCP Neuron



- Described by McCullocch and Pitts in 1943

Perceptron

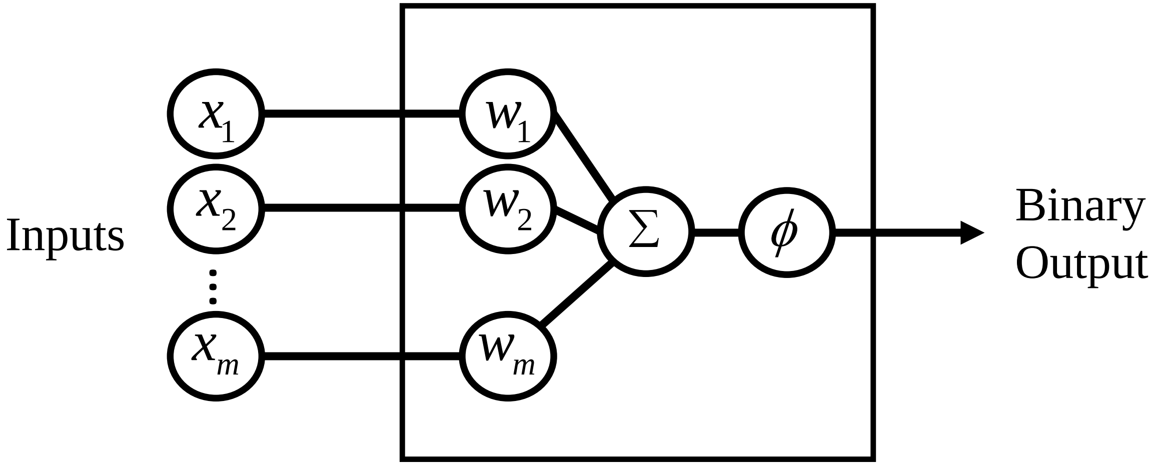


- A **Binary Classifier**
- Inputs are the **Features** of the object to be classified
- Output is the whether the object is in the class or not {1,0}
- A **Single Layer Neural Network**

STOP

START
03

Inside the Perceptron Black Box

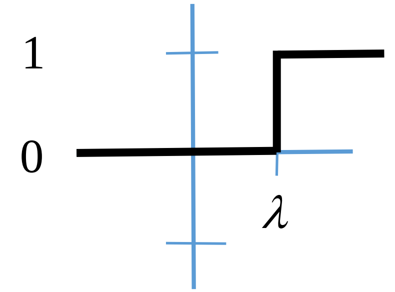


- x_i inputs
- w_i weights
- Σ sum of $w_i x_i$
- ϕ activation function

Perceptron Activation Function, $\phi(\tilde{z})$

$$\tilde{z} = \boldsymbol{w} \cdot \boldsymbol{x} = \boldsymbol{w}^T \boldsymbol{x} = \sum_{i=1}^{m} w_i x_i$$

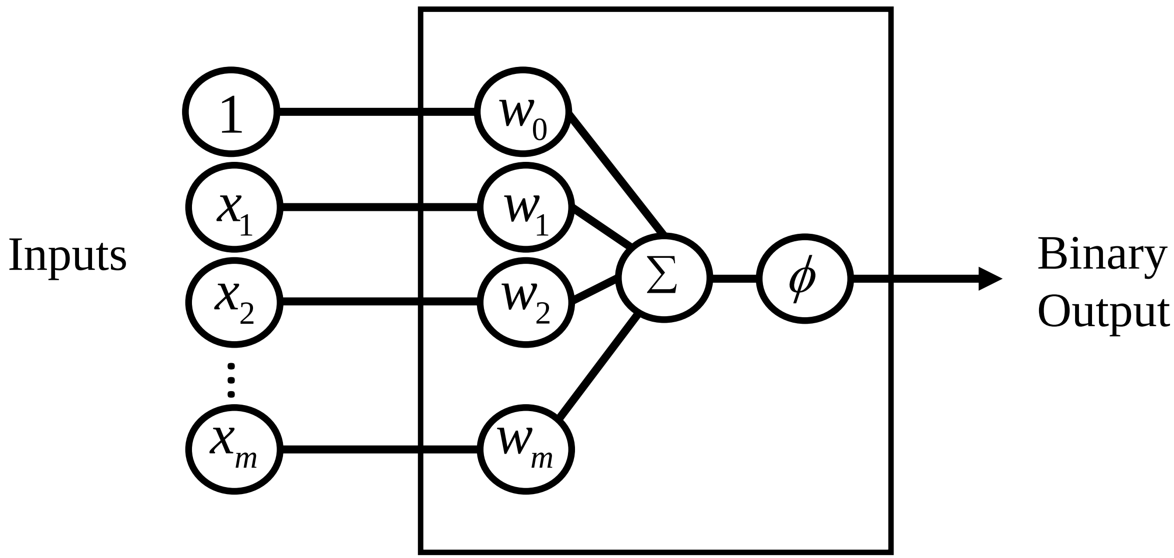
$$\phi(\tilde{z}) = \begin{cases} 1 \text{ if } \tilde{z} \geq \lambda \\ 0 \text{ otherwise} \end{cases}$$



$\phi(\tilde{z})$

STOP

Slightly Improved Perceptron Black Box

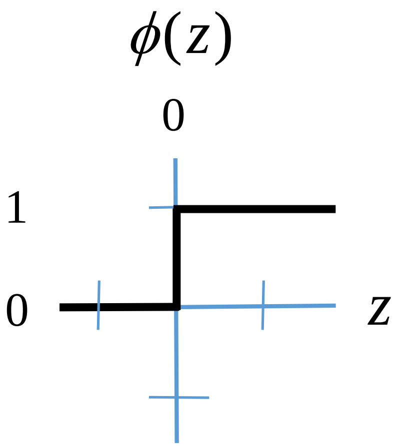


- x_i input
- w_i weight
- Σ sum of $w_i x_i$
- ϕ activation function

$$x_0 = 1$$
$$w_0 = -\lambda$$

Revised Perceptron Activation Function, $\phi(z)$

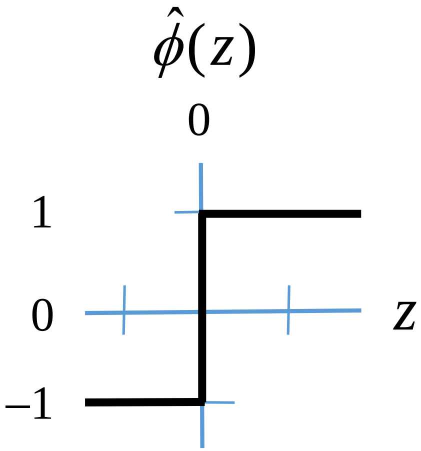
$$\phi(z) = \begin{cases} 1 \text{ if } z \geq 0 \\ 0 \text{ otherwise} \end{cases}$$



$\phi(z)$ is the Heaviside step function

Alternate Perceptron Activation Function,

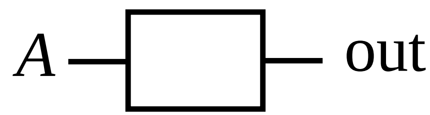
$$\hat{\phi}(z) = \begin{cases} 1 \text{ if } z \geq 0 \\ -1 \text{ otherwise} \end{cases}$$



STOP

START
06

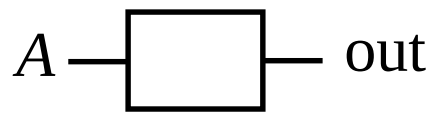
Application of a Perceptron: is a number >= 0 or not



A	out
|a|	1
-|b|	0

b≠0

Application of a Perceptron to sign of an input



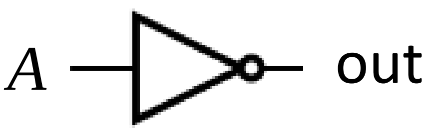
A	out	x_0	x_1	$\Phi(z)$	z	$z = w_0x_0+w_1x_1$	w_0	w_1
|a|	1	1	|a|	1	≥ 0	$w_0+w_1\|a\|$	0	1
-|b|	0	1	-|b|	0	< 0	$w_0-w_1\|b\|$		

STOP

START
07

Application of a Perceptron to Logic Gates

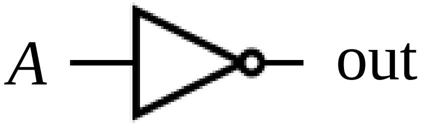
NOT



NAND

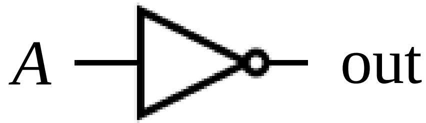
XOR

Application of a Perceptron to Logical NOT



A	out
0	1
1	0

Application of a Perceptron to Logical NOT

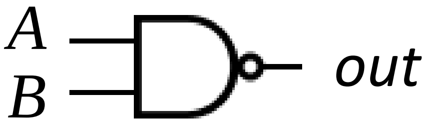


<table>
<tr><th>A</th><th>out</th><th>x_0</th><th>x_1</th><th>$\Phi(z)$</th><th>z</th><th>$z = w_0x_0+w_1x_1$</th><th>w_0</th><th>w_1</th></tr>
<tr><td>0</td><td>1</td><td>1</td><td>0</td><td>1</td><td>≥ 0</td><td>w_0</td><td rowspan="2">1</td><td rowspan="2">-2</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>1</td><td>0</td><td>< 0</td><td>w_0+w_1</td></tr>
</table>

STOP

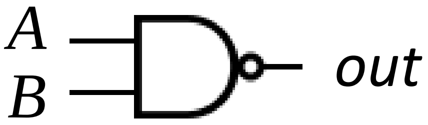
START
08

Application of a Perceptron to Logical NAND

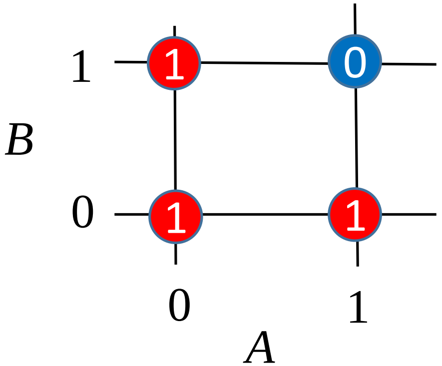


A	B	Out
0	0	1
0	1	1
1	0	1
1	1	0

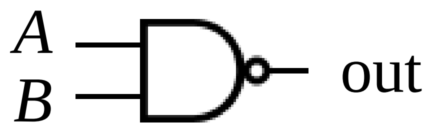
Application of a Perceptron to Logical NAND



A	B	Out
0	0	1
0	1	1
1	0	1
1	1	0



Application of a Perceptron to Logical NAND



A	B	Out	x_0	x_1	x_2	$\Phi(z)$	z	$z = w^T x$	w_0	w_1	w_2
0	0	1	1	0	0	1	≥ 0	w_0	3	-2	-2
0	1	1	1	0	1	1	≥ 0	w_0+w_2			
1	0	1	1	1	0	1	≥ 0	w_0+w_1			
1	1	0	1	1	1	0	< 0	$w_0+w_1+w_2$			

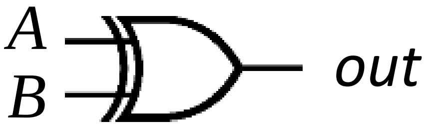
STOP

START 09

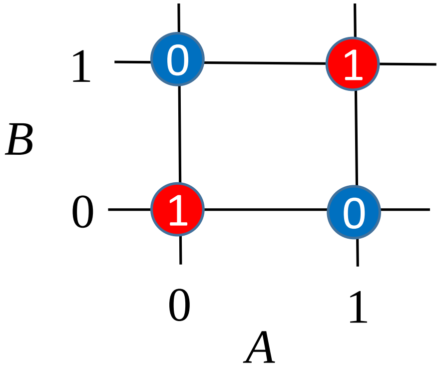
Application of a Perceptron to Logical XOR

A	B	Out
0	0	0
0	1	1
1	0	1
1	1	0

Application of a Perceptron to Logical XOR



A	*B*	Out
0	0	0
0	1	1
1	0	1
1	1	0

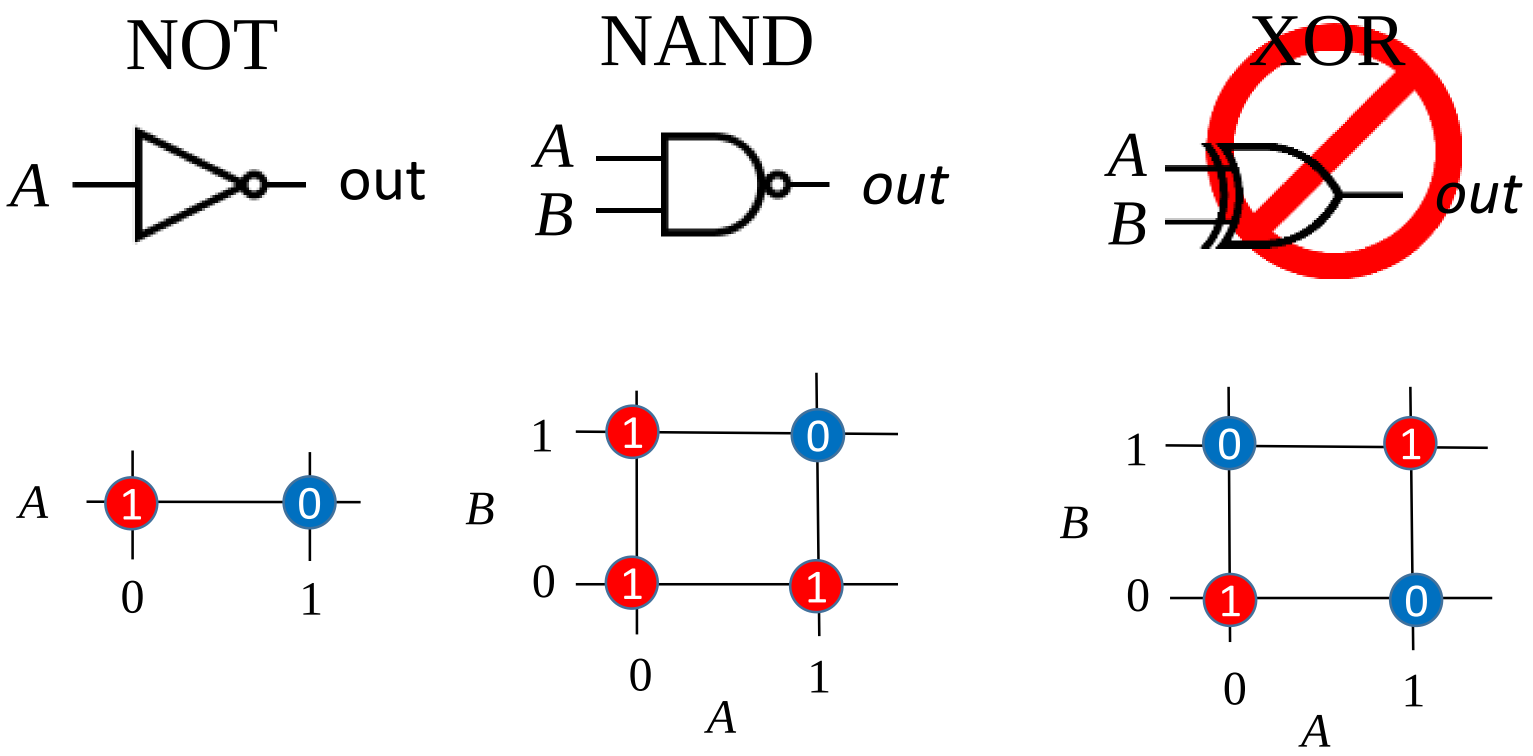


Application of a Perceptron to Logical XOR

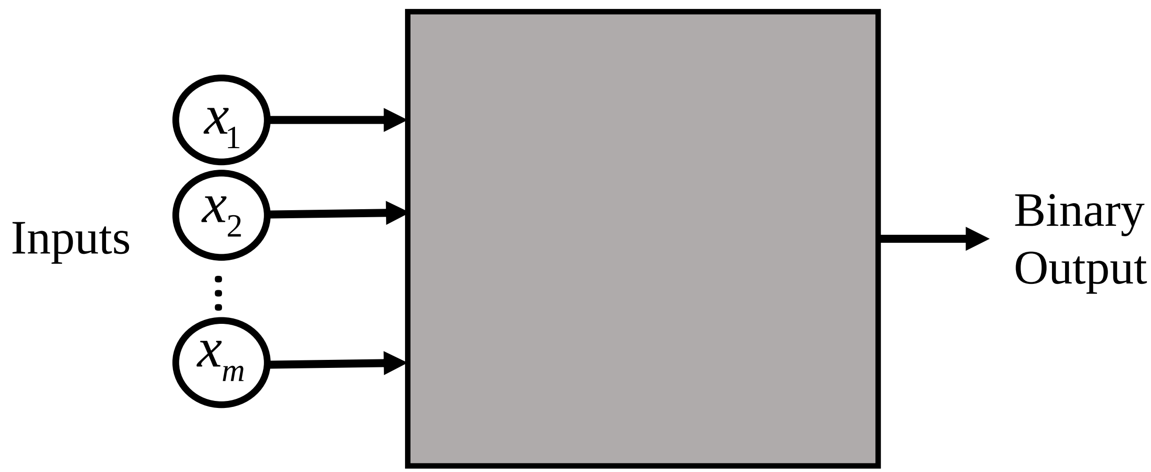
A	B	Out	x_0	x_1	x_2	$\Phi(z)$	z	$z = w^T x$	w_0	w_1	w_2
0	0	0	1	0	0	0	< 0	w_0			
0	1	1	1	0	1	1	≥ 0	w_0+w_2		{}	
1	0	1	1	1	0	1	≥ 0	w_0+w_1			
1	1	0	1	1	1	0	< 0	$w_0+w_1+w_2$			

STOP

Limits to Application of a Single Perceptron to Logic Gates



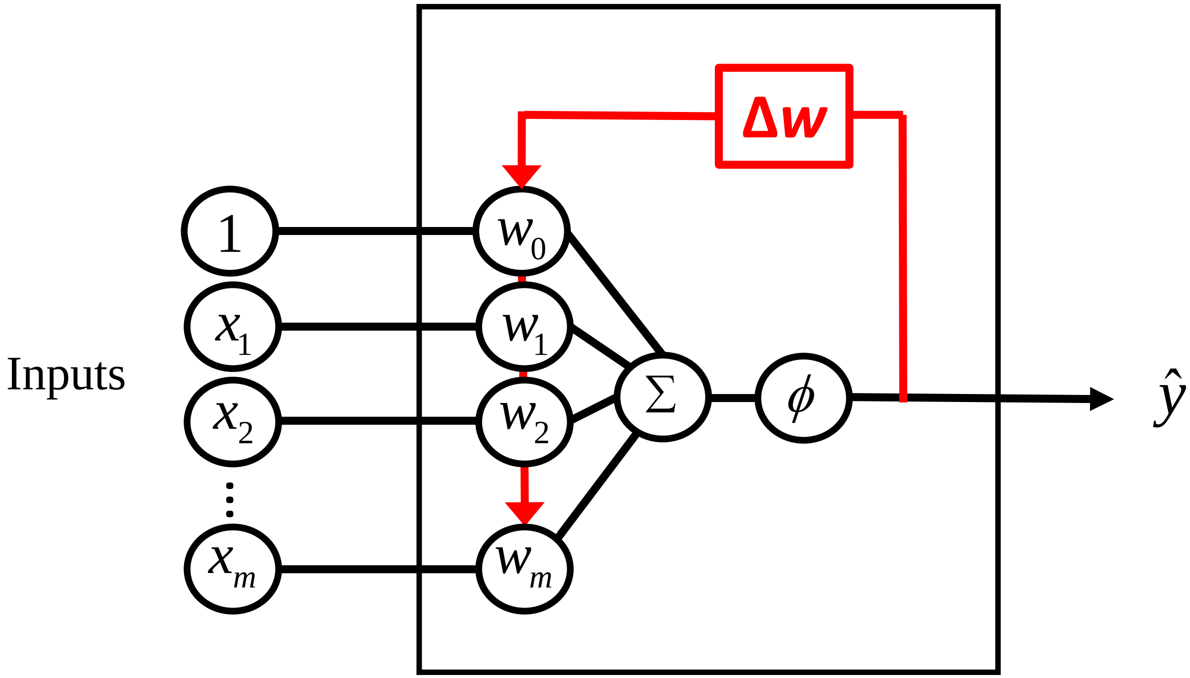
Perceptron



- How the input signals are accumulated
- How the output is determined
- **An algorithm for training**

START
12

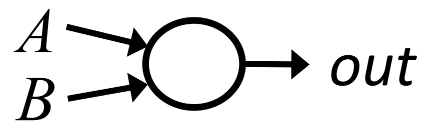
Training a Perceptron



- $\boldsymbol{x}$ input
- $\boldsymbol{w}$ weight
- $\sum$ $z = \boldsymbol{w}^T\boldsymbol{x}$
- ϕ activation function
- $\hat{y}$ computed output value

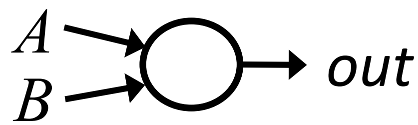
$\boldsymbol{\Delta w}$ **= adjustments to** w

Application to logical XOR



A	B	out	x_0	x_1	x_2	$\Phi(z)$	z	$z = \boldsymbol{w}^T\boldsymbol{x}$	w_0	w_1	w_2
0	0	0									
0	1	1									
1	0	1									
1	1	0									

Application to logical XOR



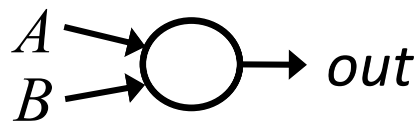
A	B	out	x_0	x_1	x_2	$\Phi(z)$	z	$z = w^T x$	w_0	w_1	w_2
0	0	0	1	0	0						
0	1	1	1	0	1						
1	0	1	1	1	0						
1	1	0	1	1	1						

$$x_0 = 1$$

$$\mathrm{x}_1 = A$$

$$x_2 = B$$

Application to Logical XOR

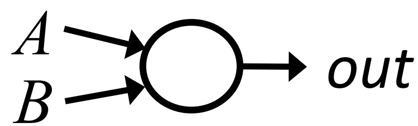


A	B	out	x_0	x_1	x_2	$\Phi(z)$	z	$z = w^T x$	w_0	w_1	w_2
0	0	0	1	0	0	0	< 0				
0	1	1	1	0	1	1	≥ 0				
1	0	1	1	1	0	1	≥ 0				
1	1	0	1	1	1	0	< 0				

$$\phi(z) = out$$

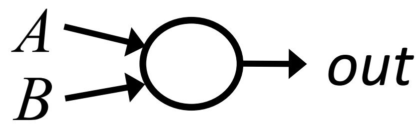
$$z = \phi^{-1}(\phi(z))$$

Application of a Perceptron to logical XOR



A	B	out	x_0	x_1	x_2	$\Phi(z)$	z	$z = w^T x$	w_0	w_1	w_2
0	0	0	1	0	0	0	< 0	w_0			
0	1	1	1	0	1	1	≥ 0	w_0+w_2			
1	0	1	1	1	0	1	≥ 0	w_0+w_1			
1	1	0	1	1	1	0	< 0	$w_0+w_1+w_2$			

Application to logical XOR



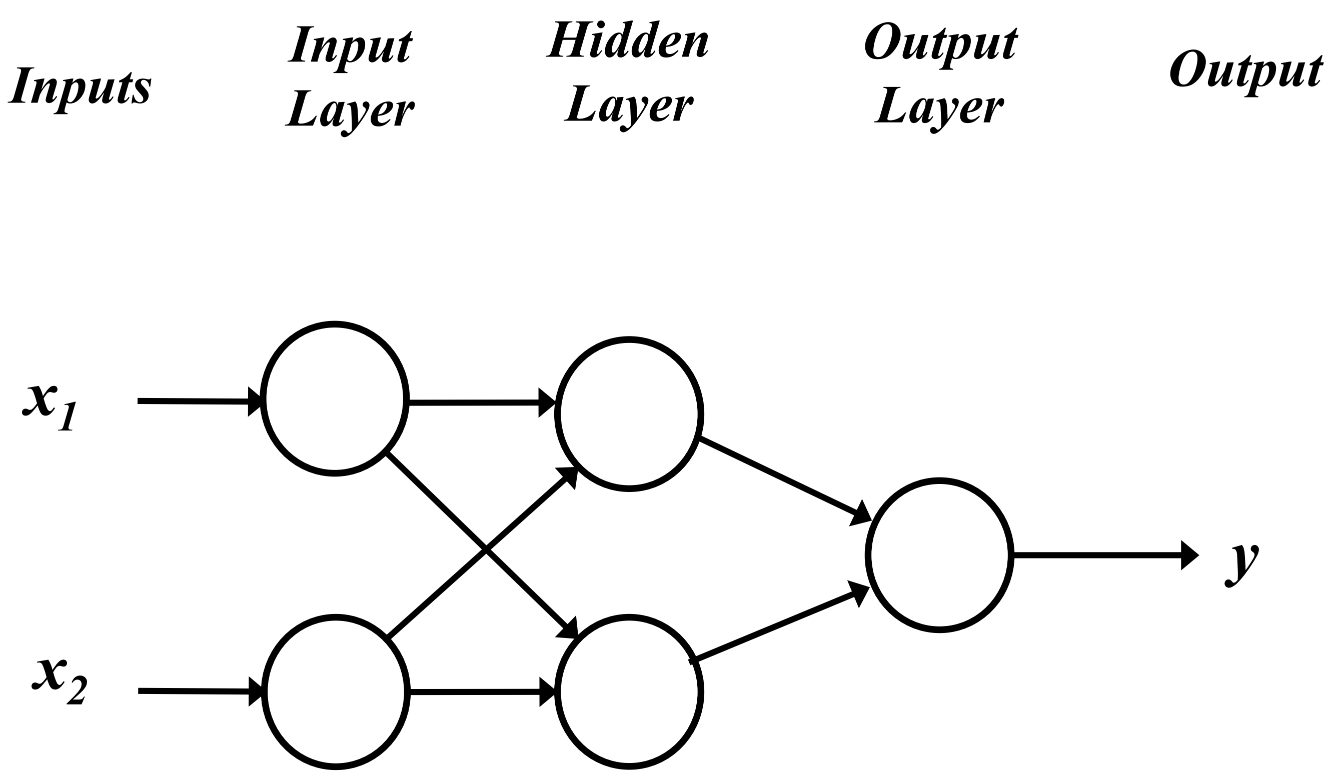
<table>
<tr><th>A</th><th>B</th><th>out</th><th>x_0</th><th>x_1</th><th>x_2</th><th>$\Phi(z)$</th><th>z</th><th>$z = w^T x$</th><th>w_0</th><th>w_1</th><th>w_2</th></tr>
<tr><td>0</td><td>0</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>< 0</td><td>w_0</td><td rowspan="4" colspan="3">{}</td></tr>
<tr><td>0</td><td>1</td><td>1</td><td>1</td><td>0</td><td>1</td><td>1</td><td>≥ 0</td><td>w_0+w_2</td></tr>
<tr><td>1</td><td>0</td><td>1</td><td>1</td><td>1</td><td>0</td><td>1</td><td>≥ 0</td><td>w_0+w_1</td></tr>
<tr><td>1</td><td>1</td><td>0</td><td>1</td><td>1</td><td>1</td><td>0</td><td>< 0</td><td>$w_0+w_1+w_2$</td></tr>
</table>

A single perceptron doesn’t work

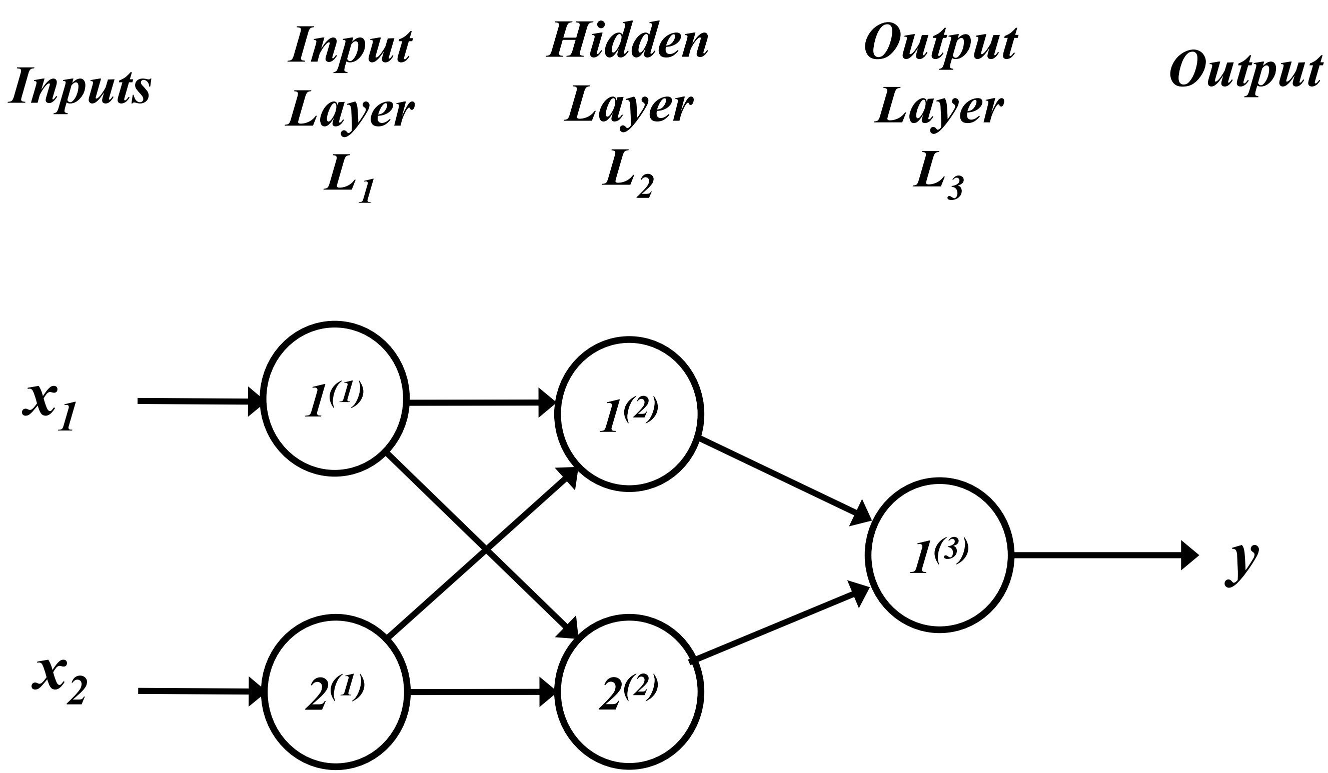
STOP

START
AUDIO 01

Candidate neural network for logical XOR



Candidate neural network for logical XOR



STOP

START RISE

Notation

$a_i^{(l)}$ ith activation unit in the lth layer

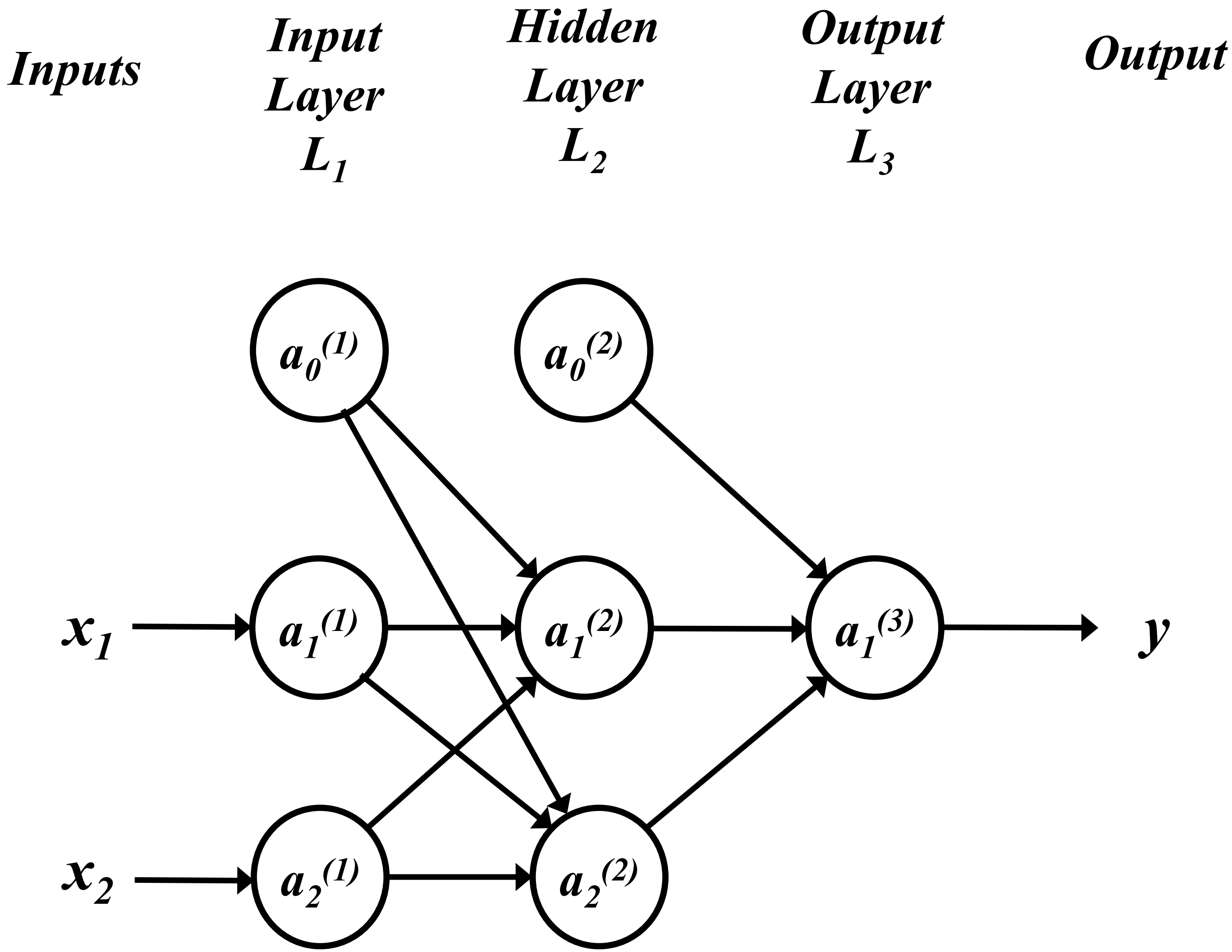
$w_{ji}^{(l)}$ weight that connects $a_i^{(l)}$ to $a_j^{(l+1)}$

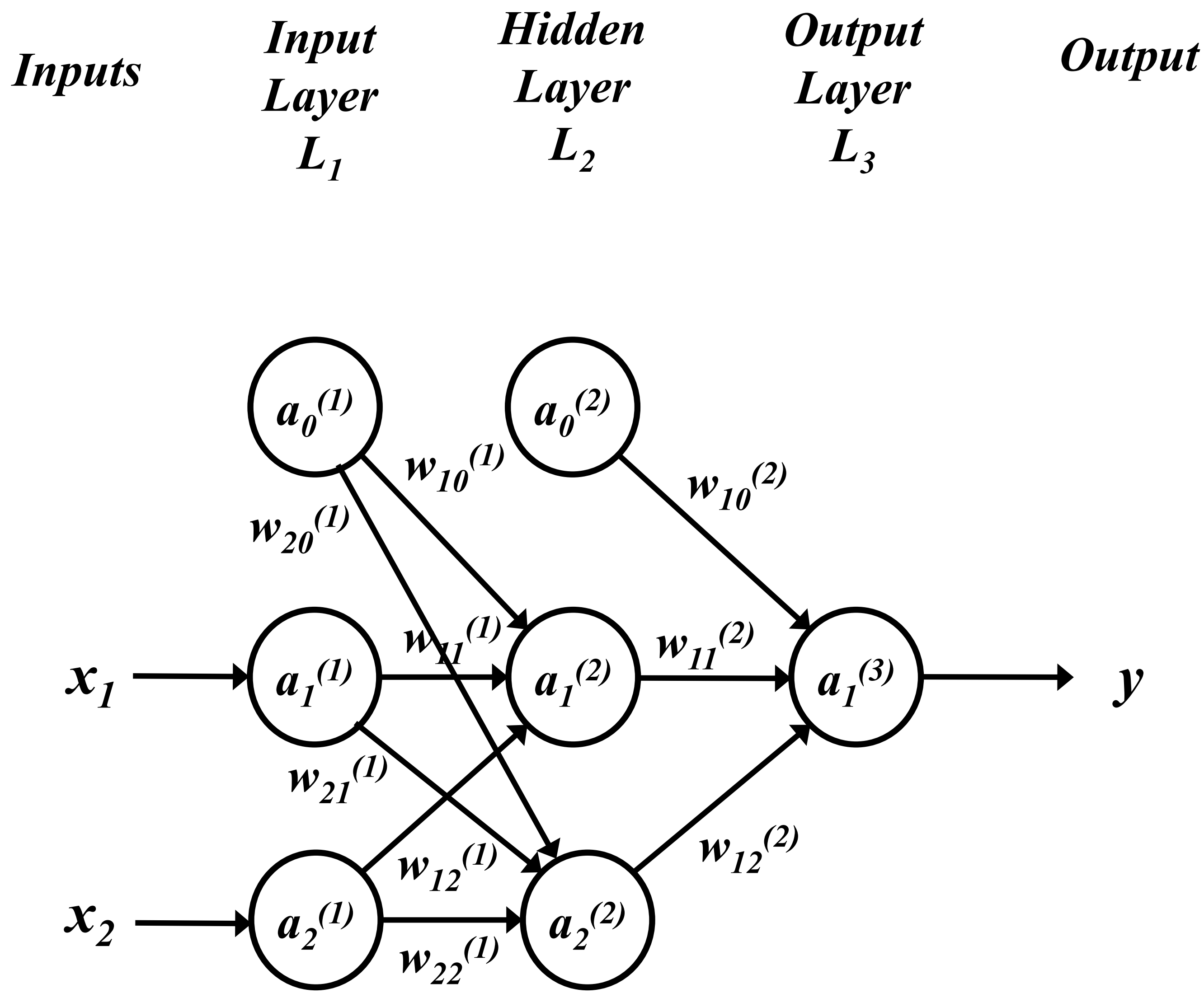
$$z_i^{(l+1)} = a_0^{(l)} w_{i0}^{(l)} + a_1^{(l)} w_{i1}^{(l)} + \ldots + a_m^{(l)} w_{im}^{(l)}$$

$$= \sum_{j=1}^{m} a_j^{(l)} w_{ij}^{(l)}$$

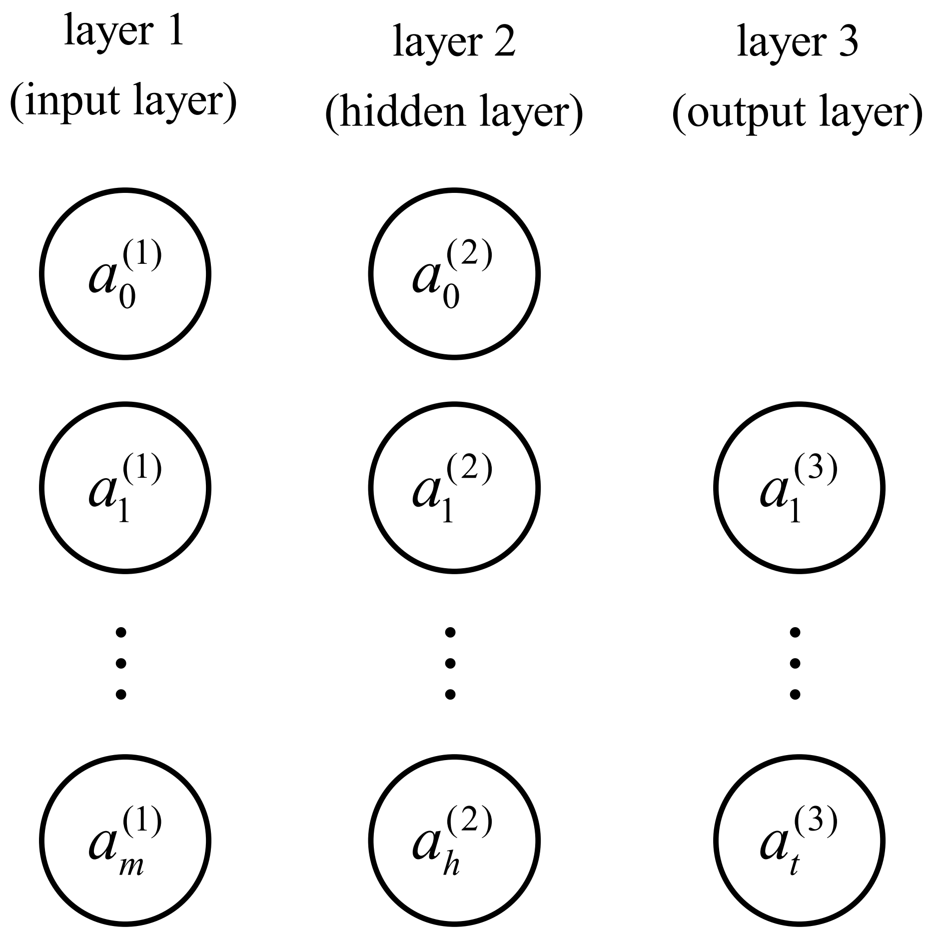
$$a_i^{(l+1)} = \phi\left(z_i^{(l+1)}\right)$$

where ϕ(z) is a non-linear activation function.

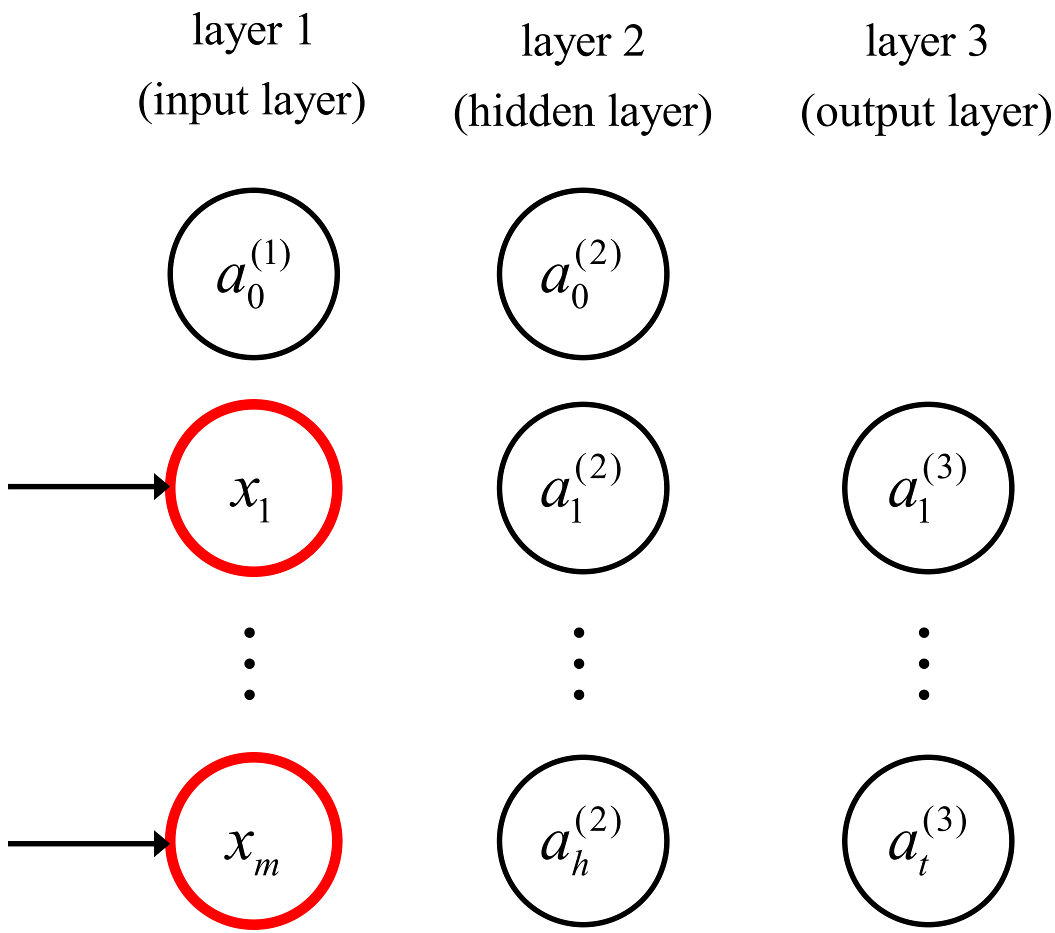

Inputs
Input Layer L_1
Hidden Layer L_2
Output Layer L_3
Output
$a_0^{(1)}$
$a_0^{(2)}$
x_1
$a_1^{(1)}$
$a_1^{(2)}$
$a_1^{(3)}$
y
x_2
$a_2^{(1)}$
$a_2^{(2)}$


Inputs
Input Layer L_1
Hidden Layer L_2
Output Layer L_3
Output
$a_0^{(1)}$
$a_0^{(2)}$
$w_{10}^{(1)}$
$w_{10}^{(2)}$
$w_{20}^{(1)}$
x_1
$a_1^{(1)}$
$w_{11}^{(1)}$
$a_1^{(2)}$
$w_{11}^{(2)}$
$a_1^{(3)}$
y
$w_{21}^{(1)}$
$w_{12}^{(1)}$
$w_{12}^{(2)}$
x_2
$a_2^{(1)}$
$a_2^{(2)}$
$w_{22}^{(1)}$

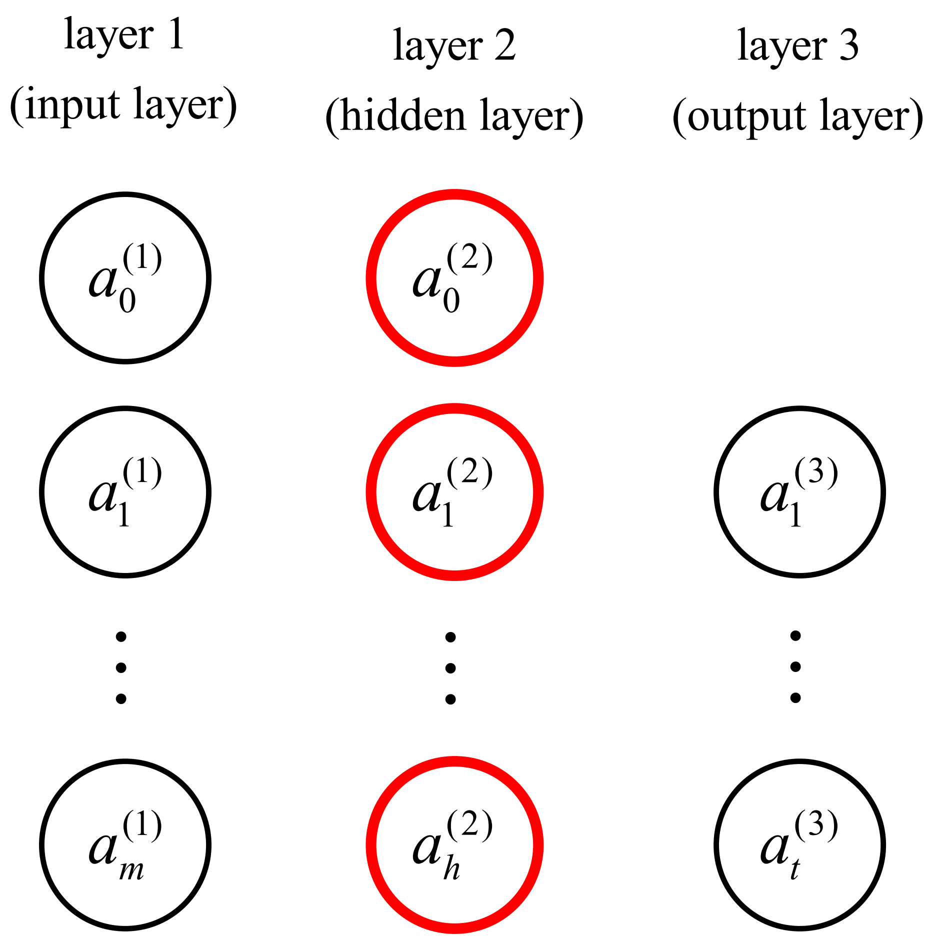
Activation units, *a*



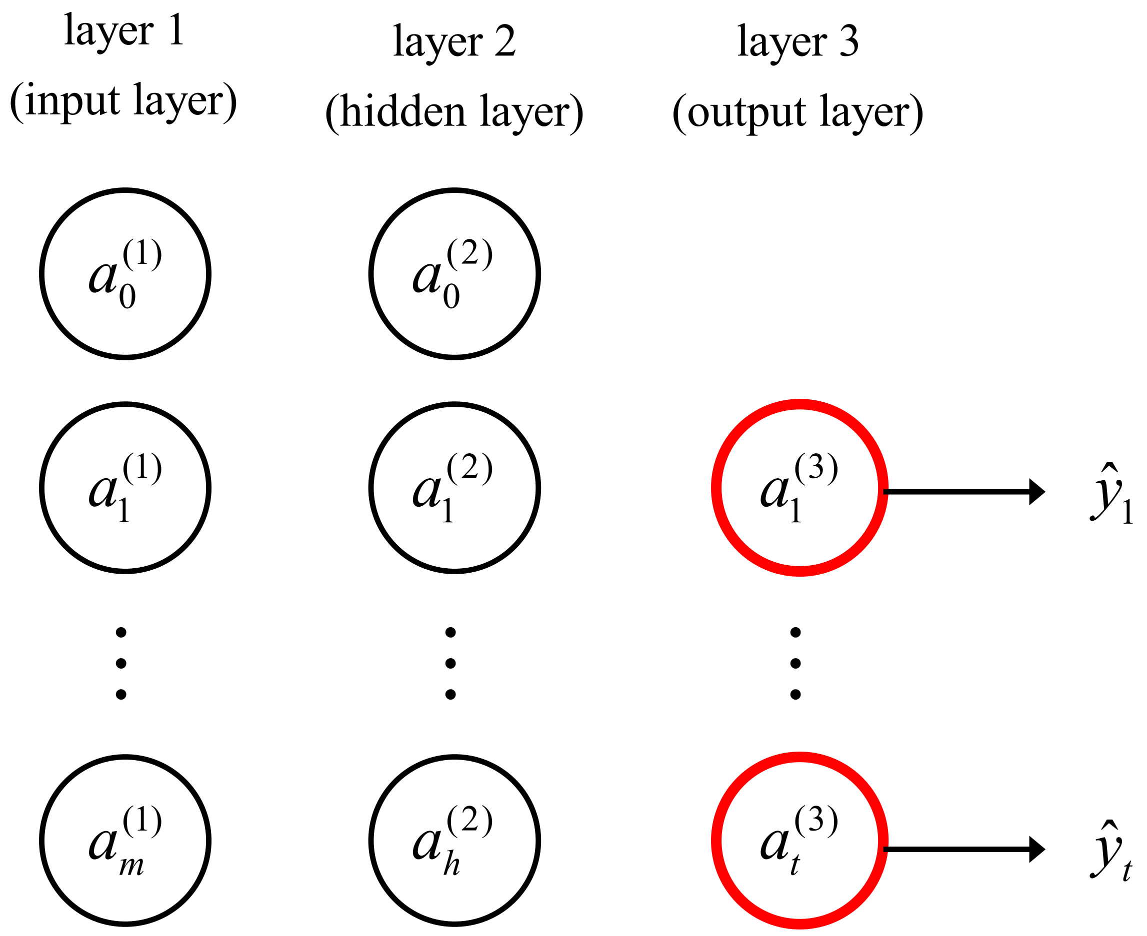
Input units, $a^{(1)}$



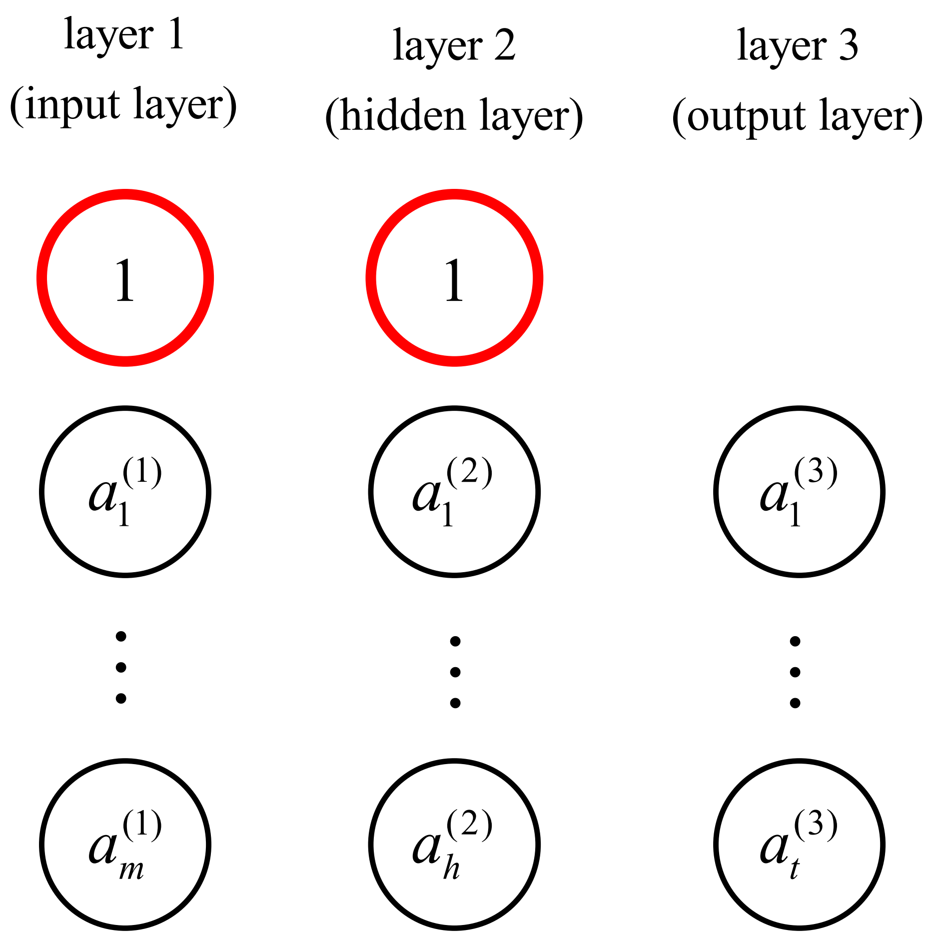
Hidden units, $a^{(2)}$



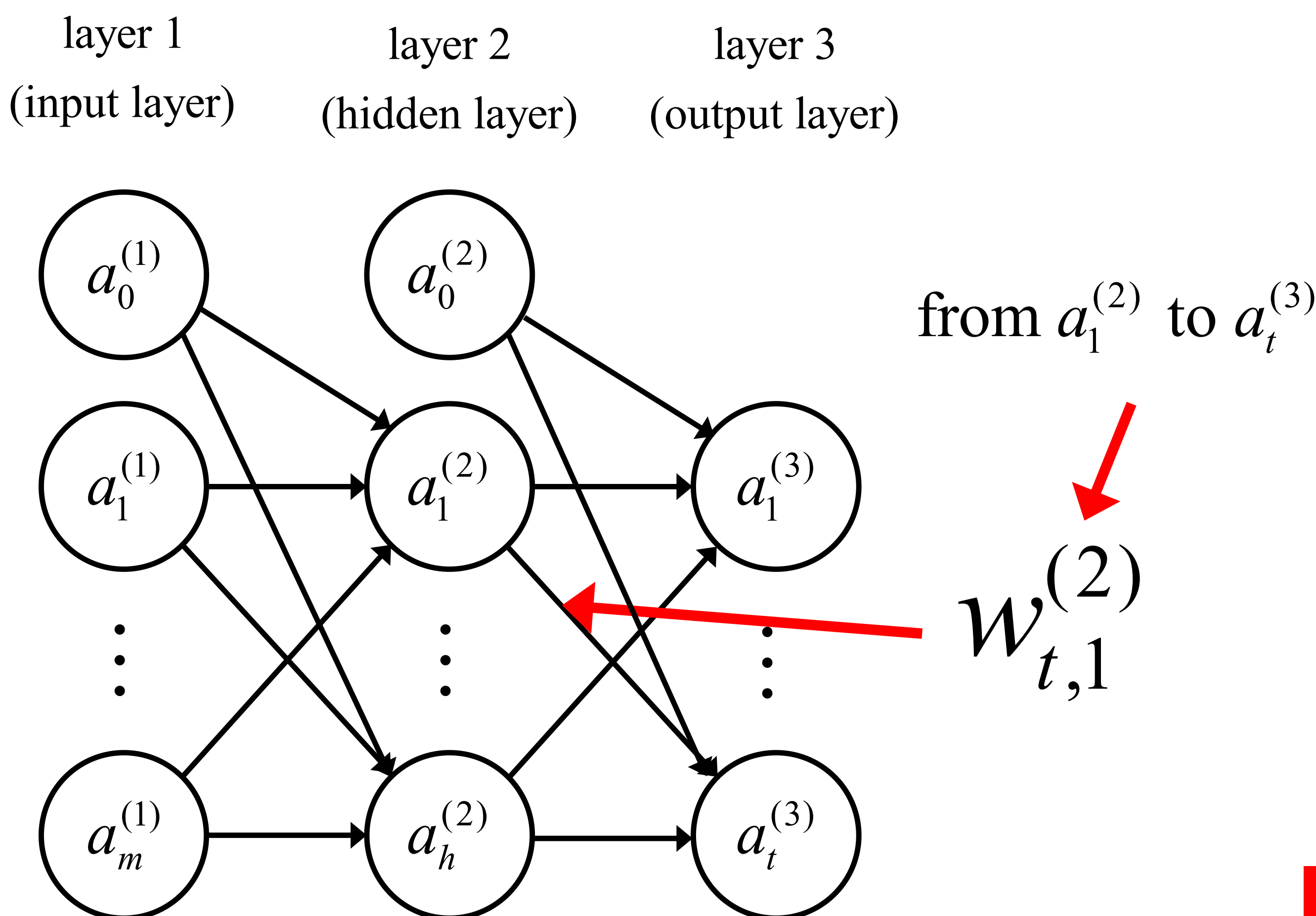
Output units, $a^{(3)}$



Bias activation units, a_0



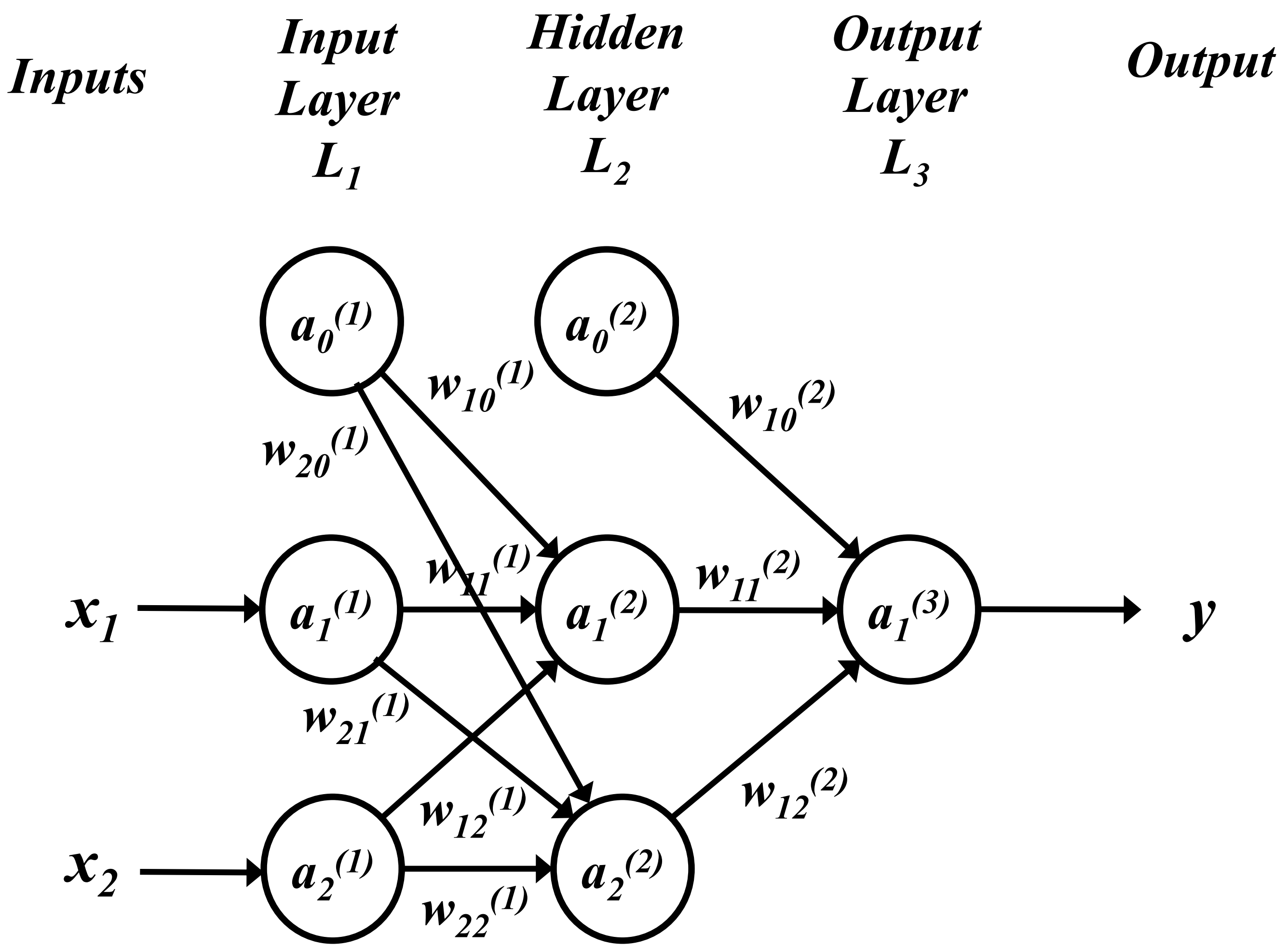
Weights, *w*



STOP

START
AUDIO 02

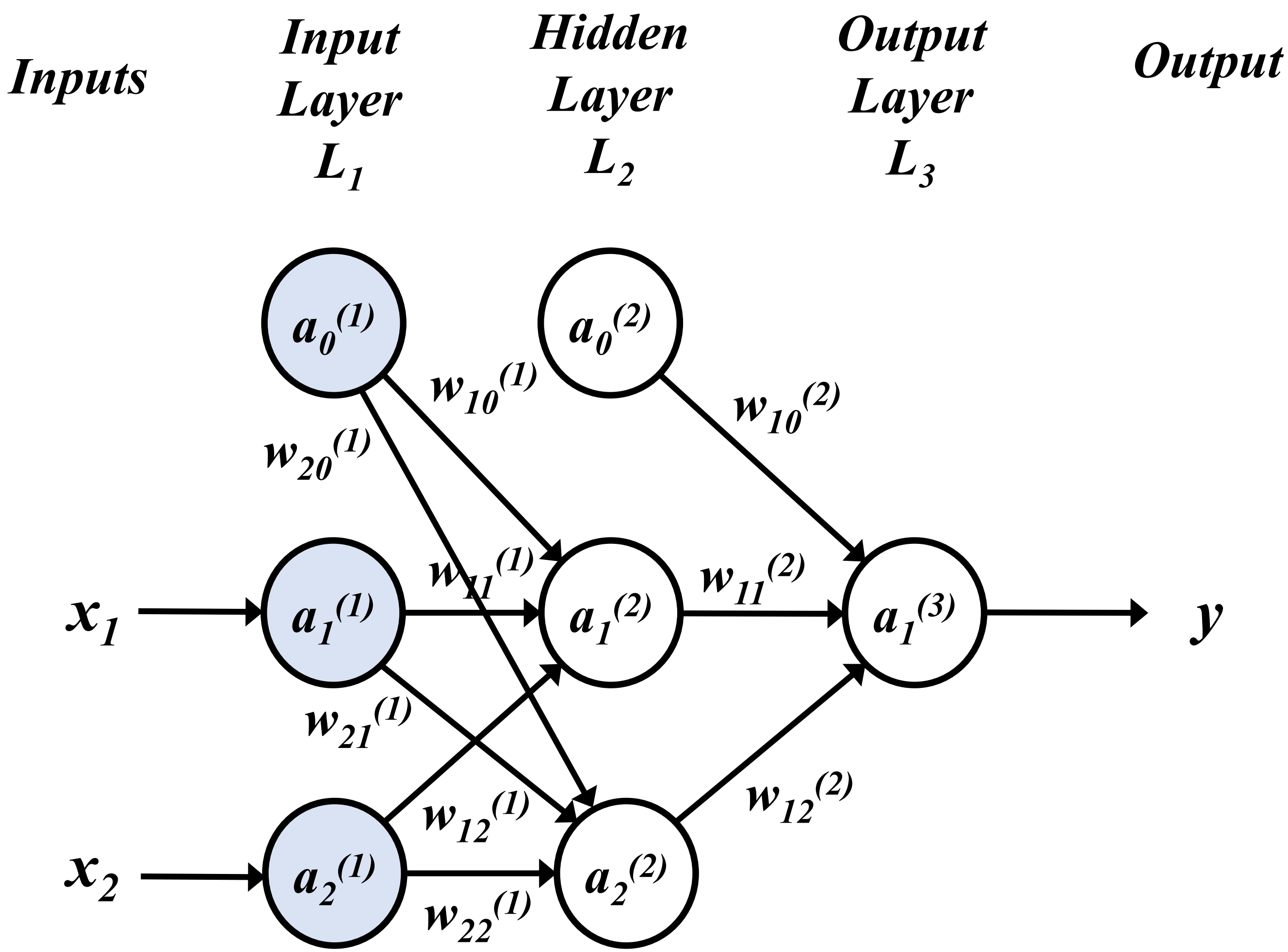
Candidate neural network for logical XOR



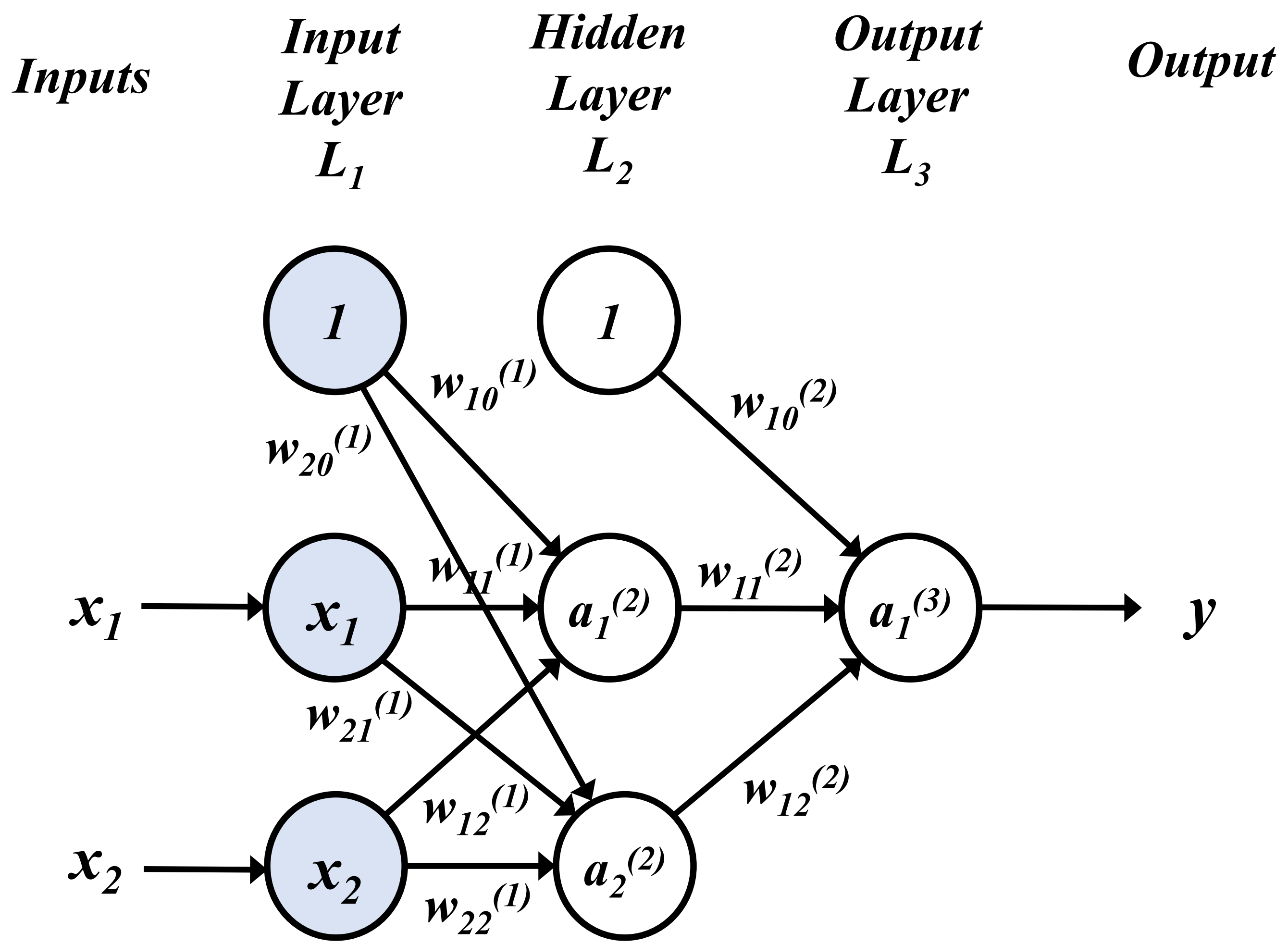
Truth Table

x_1	x_2	y
0	0	0
0	1	1
1	0	1
1	1	0

Input Layer


Inputs
Input Layer L_1
Hidden Layer L_2
Output Layer L_3
Output
$a_0^{(1)}$
$a_1^{(1)}$
$a_2^{(1)}$
$a_0^{(2)}$
$a_1^{(2)}$
$a_2^{(2)}$
$a_1^{(3)}$
x_1
x_2
y
$w_{10}^{(1)}$
$w_{20}^{(1)}$
$w_{11}^{(1)}$
$w_{21}^{(1)}$
$w_{12}^{(1)}$
$w_{22}^{(1)}$
$w_{10}^{(2)}$
$w_{11}^{(2)}$
$w_{12}^{(2)}$

Input Layer



Inputs
Input Layer L_1
Hidden Layer L_2
Output Layer L_3
Output
1
1
$w_{10}^{(1)}$
$w_{20}^{(1)}$
$w_{10}^{(2)}$
x_1
x_1
$w_{11}^{(1)}$
$a_1^{(2)}$
$w_{11}^{(2)}$
$a_1^{(3)}$
y
$w_{21}^{(1)}$
$w_{12}^{(1)}$
$w_{12}^{(2)}$
x_2
x_2
$a_2^{(2)}$
$w_{22}^{(1)}$

Input layer to hidden layer

x_1	x_2	y
0	0	0
0	1	1
1	0	1
1	1	0

$1^{(1)}$

$a_0^{(1)}$	$a_1^{(1)}$	$a_2^{(1)}$	$w_{10}^{(1)}$	$w_{11}^{(1)}$	$w_{12}^{(1)}$	$z_1^{(2)}$	ϕ	$a_1^{(2)}$
1	0	0						
1	0	1						
1	1	0						
1	1	1						

$2^{(1)}$

$a_0^{(1)}$	$a_1^{(1)}$	$a_2^{(1)}$	$w_{20}^{(1)}$	$w_{21}^{(1)}$	$w_{22}^{(1)}$	$z_2^{(2)}$	ϕ	$a_2^{(2)}$
1	0	0						
1	0	1						
1	1	0						
1	1	1						

Input layer to hidden layer

x_1	x_2	y
0	0	0
0	1	1
1	0	1
1	1	0

$1^{(1)}$

$a_0^{(1)}$	$a_1^{(1)}$	$a_2^{(1)}$	$w_{10}^{(1)}$	$w_{11}^{(1)}$	$w_{12}^{(1)}$	$z_1^{(2)}$	ϕ	$a_1^{(2)}$
1	0	0	-1	2	2			
1	0	1						
1	1	0						
1	1	1						

$2^{(1)}$

$a_0^{(1)}$	$a_1^{(1)}$	$a_2^{(1)}$	$w_{20}^{(1)}$	$w_{21}^{(1)}$	$w_{22}^{(1)}$	$z_2^{(2)}$	ϕ	$a_2^{(2)}$
1	0	0	-3	2	2			
1	0	1						
1	1	0						
1	1	1						

Input layer to hidden layer

x_1	x_2	y
0	0	0
0	1	1
1	0	1
1	1	0

$1^{(1)}$

$a_0^{(1)}$	$a_1^{(1)}$	$a_2^{(1)}$	$w_{10}^{(1)}$	$w_{11}^{(1)}$	$w_{12}^{(1)}$	$z_1^{(2)}$	ϕ	$a_1^{(2)}$
1	0	0				-1		
1	0	1	-1	2	2	1		
1	1	0				1		
1	1	1				3		

$2^{(1)}$

$a_0^{(1)}$	$a_1^{(1)}$	$a_2^{(1)}$	$w_{20}^{(1)}$	$w_{21}^{(1)}$	$w_{22}^{(1)}$	$z_2^{(2)}$	ϕ	$a_2^{(2)}$
1	0	0				-3		
1	0	1	-3	2	2	-1		
1	1	0				-1		
1	1	1				1		

Input layer to hidden layer

x_1	x_2	y
0	0	0
0	1	1
1	0	1
1	1	0

$1^{(1)}$

$a_0^{(1)}$	$a_1^{(1)}$	$a_2^{(1)}$	$w_{10}^{(1)}$	$w_{11}^{(1)}$	$w_{12}^{(1)}$	$z_1^{(2)}$	ϕ	$a_1^{(2)}$
1	0	0				-1	0	
1	0	1	-1	2	2	1	1	
1	1	0				1	1	
1	1	1				3	1	

$2^{(1)}$

$a_0^{(1)}$	$a_1^{(1)}$	$a_2^{(1)}$	$w_{20}^{(1)}$	$w_{21}^{(1)}$	$w_{22}^{(1)}$	$z_2^{(2)}$	ϕ	$a_2^{(2)}$
1	0	0				-3	0	
1	0	1	-3	2	2	-1	0	
1	1	0				-1	0	
1	1	1				1	1	

Input layer to hidden layer

x_1	x_2	y
0	0	0
0	1	1
1	0	1
1	1	0

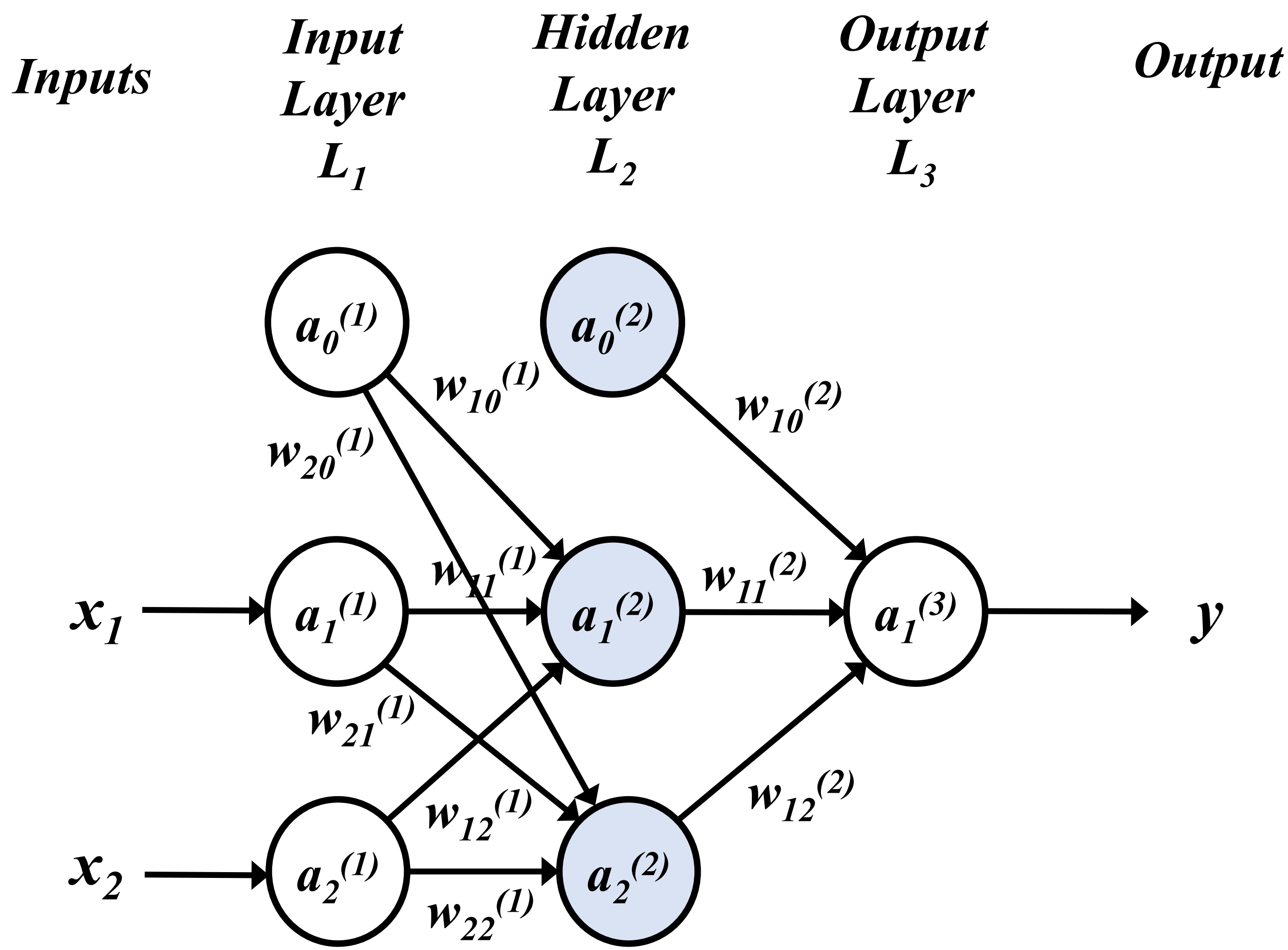
$1^{(1)}$

$a_0^{(1)}$	$a_1^{(1)}$	$a_2^{(1)}$	$w_{10}^{(1)}$	$w_{11}^{(1)}$	$w_{12}^{(1)}$	$z_1^{(2)}$	ϕ	$a_1^{(2)}$
1	0	0	-1	2	2	-1	0	0
1	0	1				1	1	1
1	1	0				1	1	1
1	1	1				3	1	1

$2^{(1)}$

$a_0^{(1)}$	$a_1^{(1)}$	$a_2^{(1)}$	$w_{20}^{(1)}$	$w_{21}^{(1)}$	$w_{22}^{(1)}$	$z_2^{(2)}$	ϕ	$a_2^{(2)}$
1	0	0	-3	2	2	-3	0	0
1	0	1				-1	0	0
1	1	0				-1	0	0
1	1	1				1	1	1

Hidden Layer



Hidden layer to output layer

$a_1^{(2)}$	$a_2^{(2)}$
0	0
1	0
1	0
1	1

Hidden layer to output layer

$1^{(3)}$

$a_1^{(2)}$	$a_2^{(2)}$
0	0
1	0
1	0
1	1

$a_0^{(2)}$	$a_1^{(2)}$	$a_2^{(2)}$	$w_{10}^{(2)}$	$w_{11}^{(2)}$	$w_{12}^{(2)}$	$z_1^{(3)}$	ϕ	$a_1^{(3)}$
1	0	0						
1	1	0						
1	1	0						
1	1	1						

Hidden layer to output layer

$1^{(3)}$

$a_1^{(2)}$	$a_2^{(2)}$
0	0
1	0
1	0
1	1

$a_0^{(2)}$	$a_1^{(2)}$	$a_2^{(2)}$	$w_{10}^{(2)}$	$w_{11}^{(2)}$	$w_{12}^{(2)}$	$z_1^{(3)}$	ϕ	$a_1^{(3)}$
1	0	0	-1	2	-2			
1	1	0						
1	1	0						
1	1	1						

Hidden layer to output layer

$1^{(3)}$

$a_1^{(2)}$	$a_2^{(2)}$
0	0
1	0
1	0
1	1

$a_0^{(2)}$	$a_1^{(2)}$	$a_2^{(2)}$	$w_{10}^{(2)}$	$w_{11}^{(2)}$	$w_{12}^{(2)}$	$z_1^{(3)}$	ϕ	$a_1^{(3)}$
1	0	0	-1	2	-2	-1		
1	1	0				1		
1	1	0				1		
1	1	1				-1		

Hidden layer to output layer

$\mathbf{1^{(3)}}$

$a_1^{(2)}$	$a_2^{(2)}$
0	0
1	0
1	0
1	1

$a_0^{(2)}$	$a_1^{(2)}$	$a_2^{(2)}$	$w_{10}^{(2)}$	$w_{11}^{(2)}$	$w_{12}^{(2)}$	$z_1^{(3)}$	ϕ	$a_1^{(3)}$
1	0	0				-1	0	
1	1	0	-1	2	-2	1	1	
1	1	0				1	1	
1	1	1				-1	0	

Hidden layer to output layer

$1^{(3)}$

$a_1^{(2)}$	$a_2^{(2)}$
0	0
1	0
1	0
1	1

$a_0^{(2)}$	$a_1^{(2)}$	$a_2^{(2)}$	$w_{10}^{(2)}$	$w_{11}^{(2)}$	$w_{12}^{(2)}$	$z_1^{(3)}$	ϕ	$a_1^{(3)}$
1	0	0	-1	2	-2	-1	0	0
1	1	0	-1	2	-2	1	1	1
1	1	0	-1	2	-2	1	1	1
1	1	1	-1	2	-2	-1	0	0

Output layer to output

$a_1^{(3)}$
0
1
1
0

Output layer to output

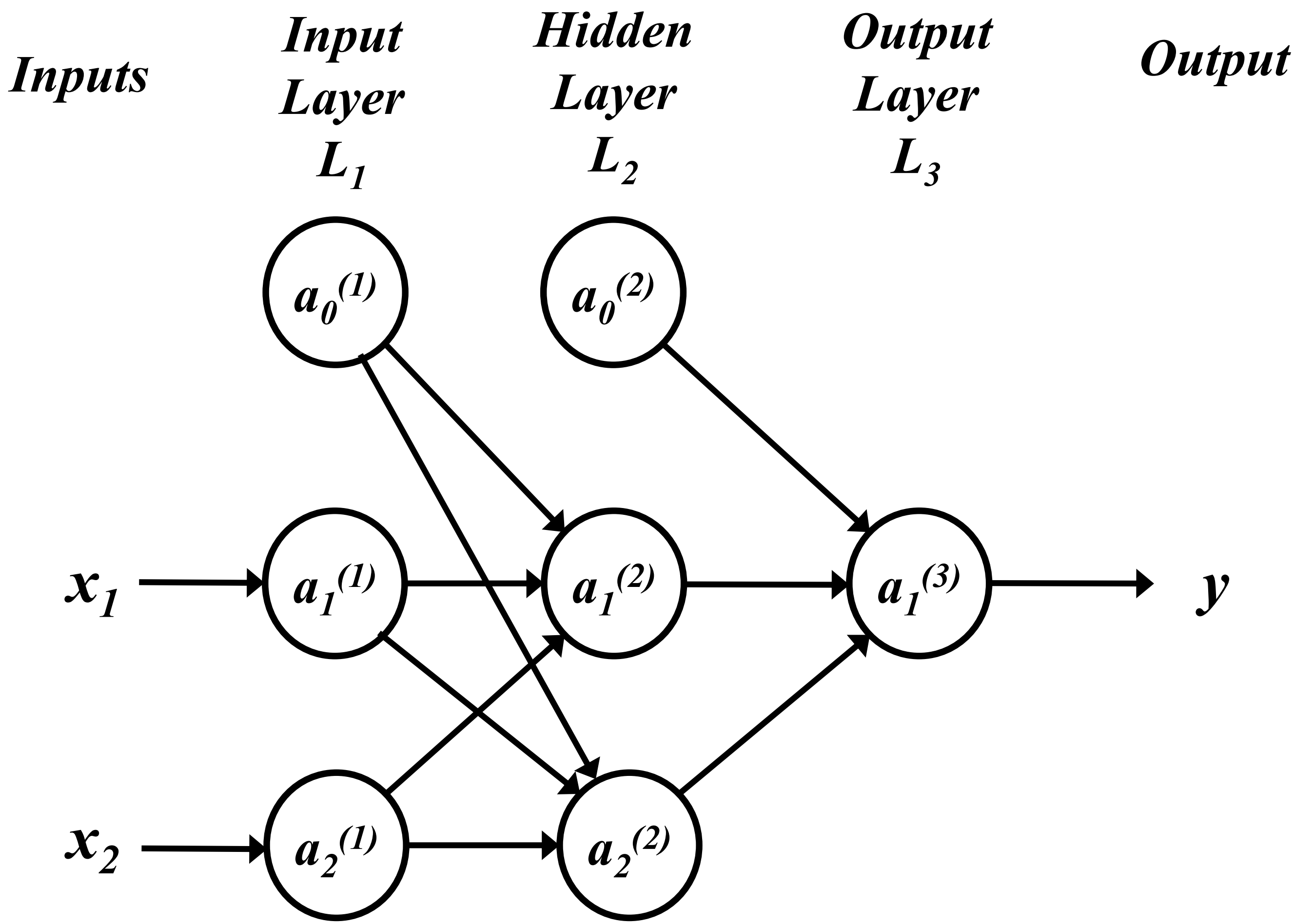
$a_1^{(3)}$	$\hat{y}$
0	0
1	1
1	1
0	0

$$\hat{y} = y$$

x_1	x_2	y
0	0	0
0	1	1
1	0	1
1	1	0

$\hat{y}$
0
1
1
0

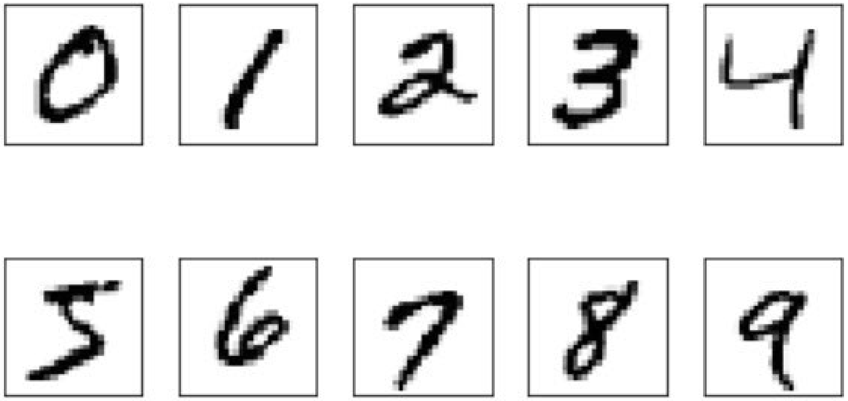
This network can work for logical XOR



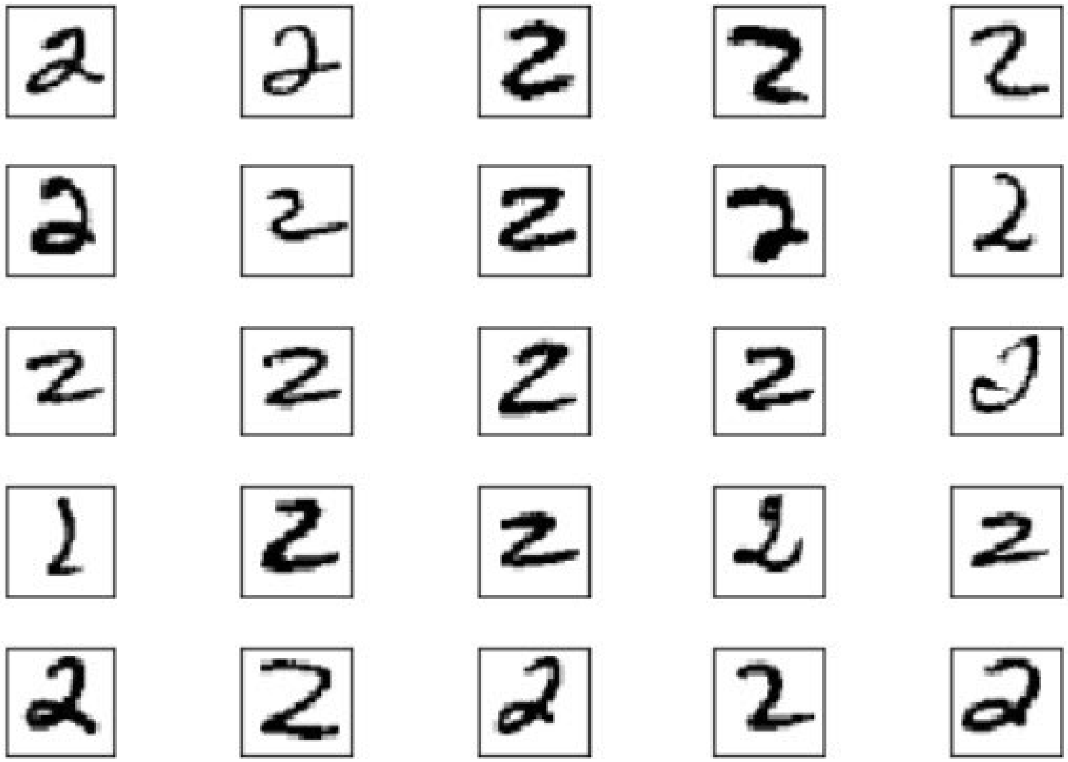
STOP

START
AUDIO 03

Another example: recognizing handwritten digits



This is a hard problem



MNIST Handwritten Digit Database

- MNIST – Modified National Institute of Standards and Technology database

MNIST Handwritten Digit Database

- MNIST – <u>M</u>odified <u>N</u>ational <u>I</u>nstitute of <u>S</u>tandards and <u>T</u>echnology database
- 250 individuals

MNIST Handwritten Digit Database

- MNIST – Modified National Institute of Standards and Technology database
- 250 individuals
- 60,000 training images
- 10,000 test images

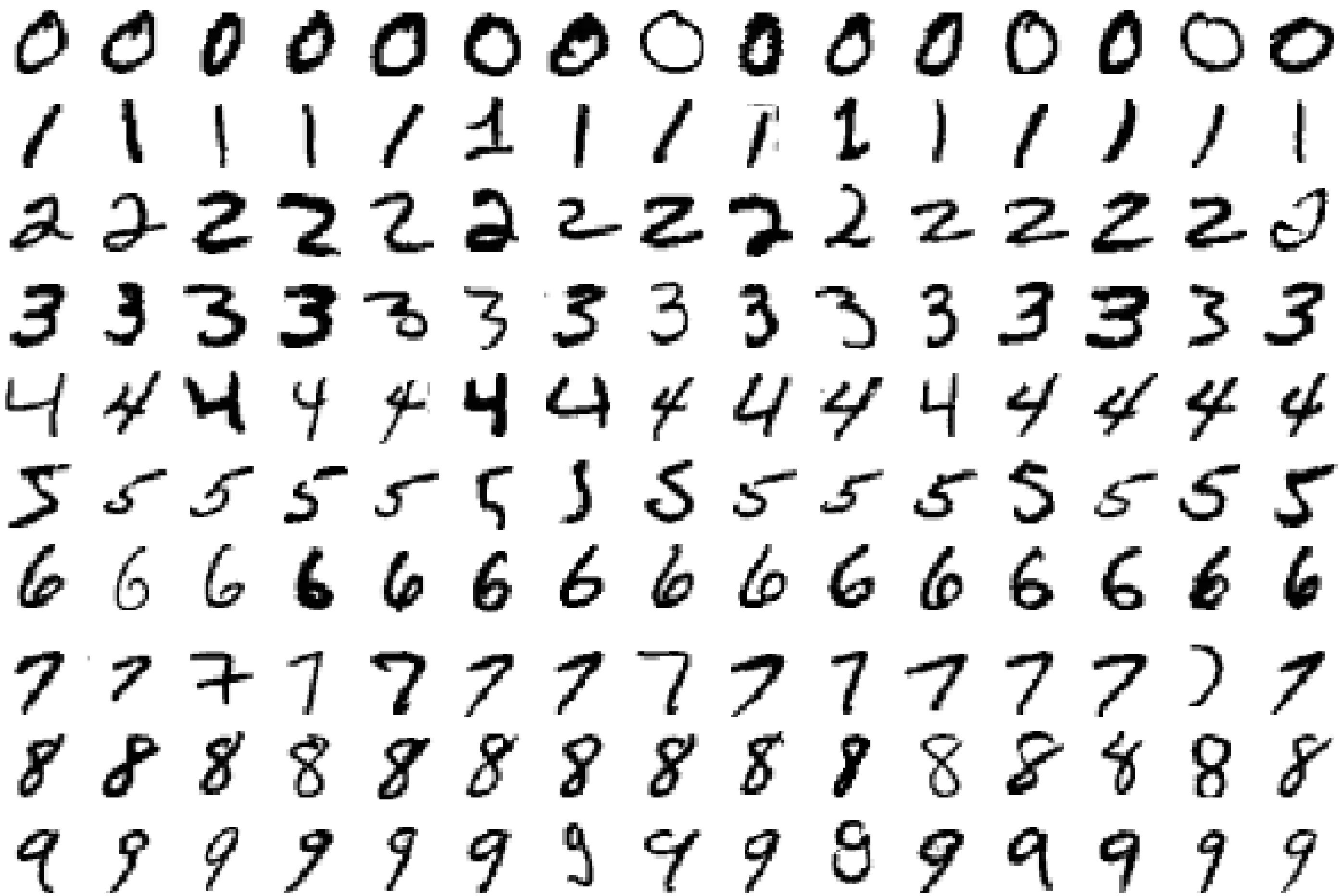
MNIST Handwritten Digit Database

- MNIST – Modified National Institute of Standards and Technology database
- 250 individuals
- 60,000 training images
- 10,000 test images
- Handwritten digits (0,1,2,3,4,5,6,7,8,9)

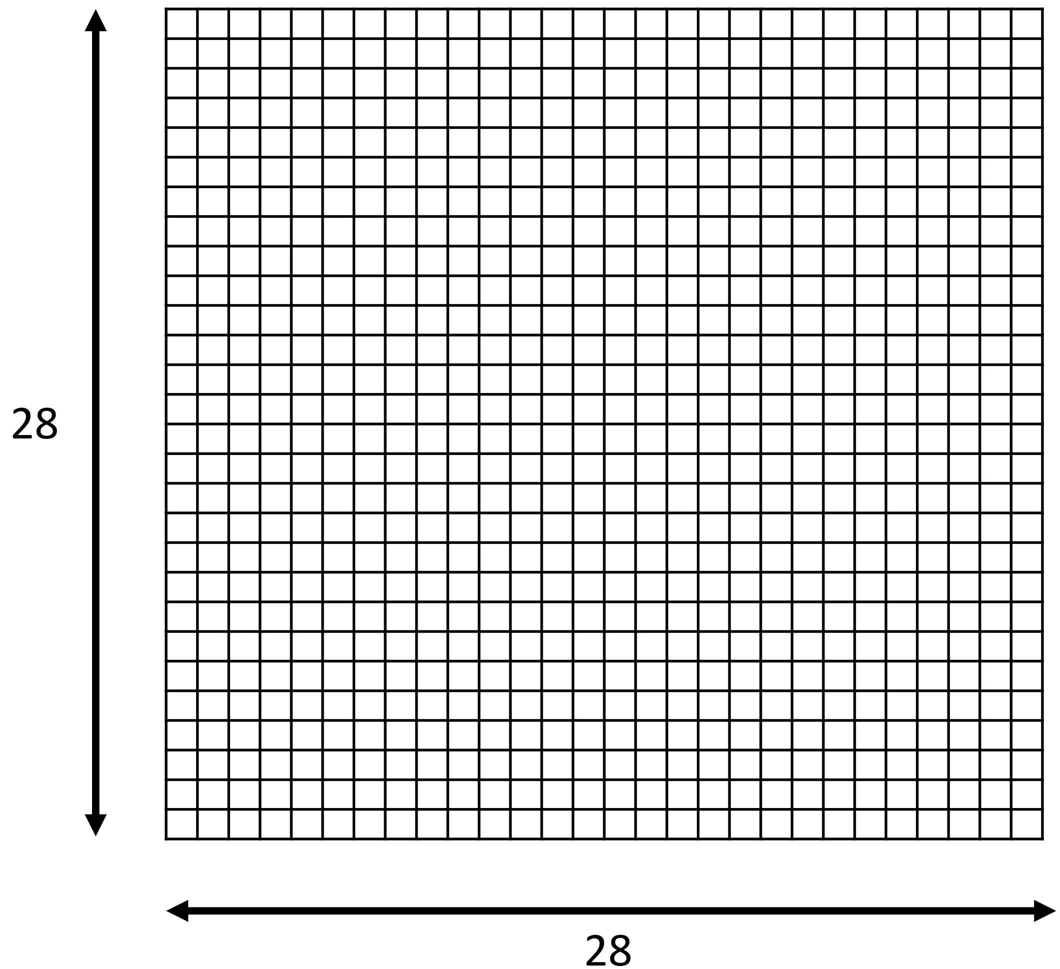
MNIST Handwritten Digit Database

- MNIST – Modified National Institute of Standards and Technology database
- 250 individuals
- 60,000 training images
- 10,000 test images
- Handwritten digits (0,1,2,3,4,5,6,7,8,9)
- Each image is 28x28 pixels (784 pixels)
- Each pixel has grayscale values [0,1,…,255]

Sample data



Each image is a set of pixels

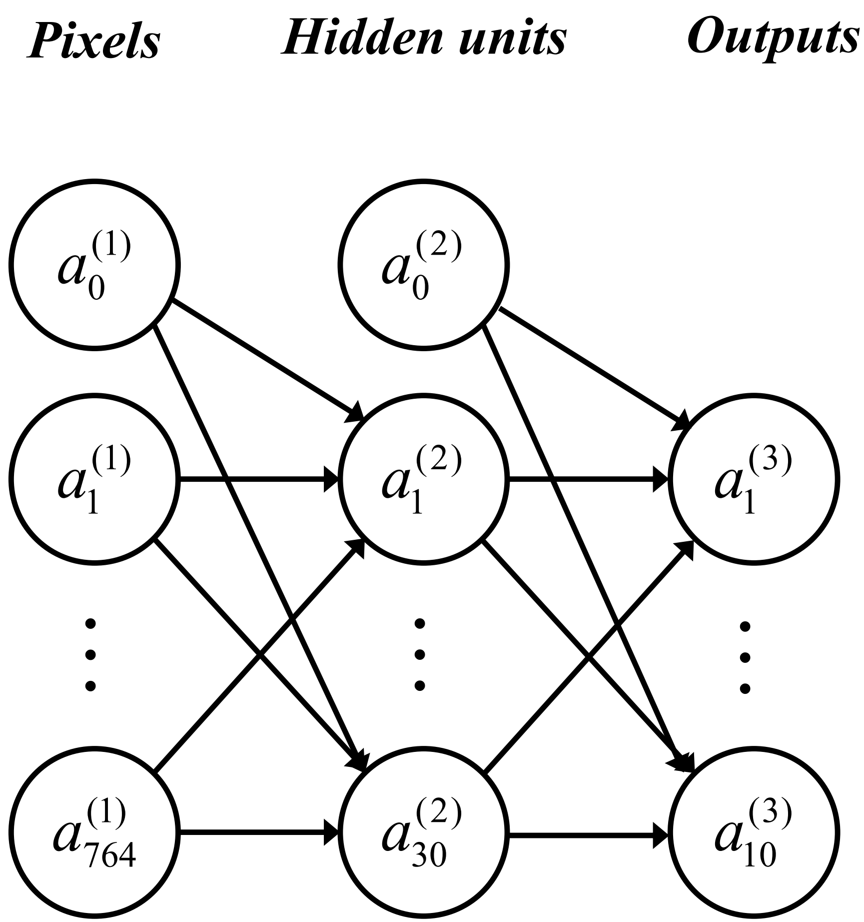


$28 \times 28 = 784$

each pixel has
a grayscale
value from
0 to 255

256^{784}
$\sim 10^{1,888}$
possible
images

Candidate neural network



Output coding options

Label encoding

$$a_1^{(3)} \longrightarrow \hat{y} = \{1, 2..., t\}$$

One-hot encoding

$$a_1^{(3)} \longrightarrow \hat{y}_1 = \{0,1\}$$

$$\vdots$$

$$a_t^{(3)} \longrightarrow \hat{y}_t = \{0,1\}$$

$$\hat{y}_t = \{0,1\}$$

$$\sum_{i=1}^{t} \hat{y}_i = 1$$

Label encoding

In **label encoding** we use cardinal numbers as the labels of the categorical feature of interest.

sepal length	sepal width	petal length	petal width	class
5.1	3.5	4.0	0.2	0
4.9	3.0	1.4	0.2	0
7.0	3.2	4.7	1.4	1
6.4	3.2	4.5	1.5	1
6.3	3.3	6.0	2.5	2
5.8	2.7	5.1	1.9	2

class	species
0	Setosa
1	Versicolour
2	Virginica

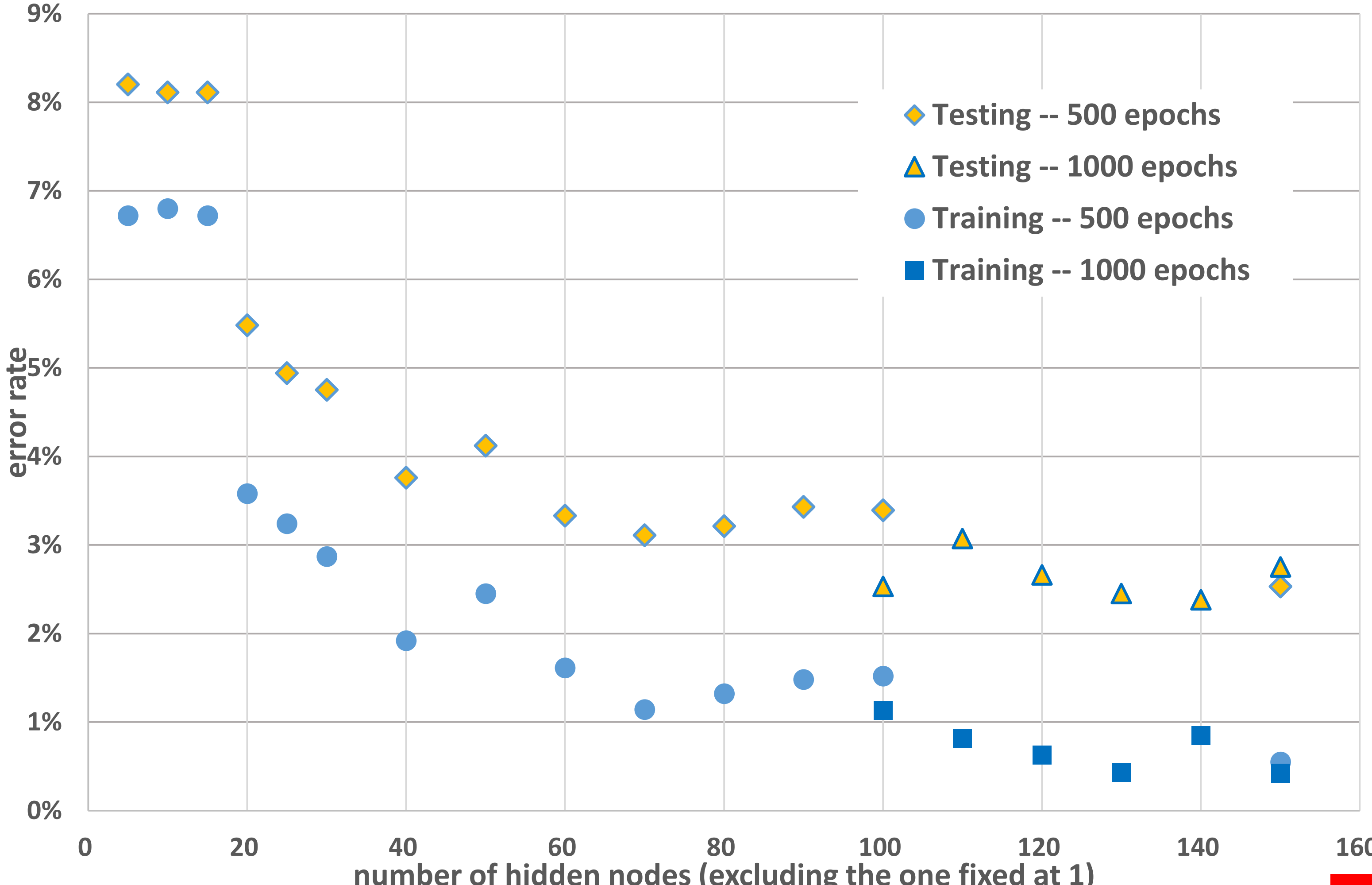
from Wikipedia

One-hot encoding

In **one-hot** encoding we use a separate feature for each unique label of the categorical feature of interest.

sepal length	sepal width	petal length	petal width	Setosa	Versicolour	Virginica
5.1	3.5	4.0	0.2	1	0	0
4.9	3.0	1.4	0.2	1	0	0
7.0	3.2	4.7	1.4	0	1	0
6.4	3.2	4.5	1.5	0	1	0
6.3	3.3	6.0	2.5	0	0	1
5.8	2.7	5.1	1.9	0	0	1

Results as the size of the hidden layer is changed



STOP

Typical non-linear activation functions

Logistic / Sigmoid $\phi(\text{z}) = \dfrac{1}{1+e^{-z}}$

Hyperbolic tangent $\phi(z) = \dfrac{e^{z} - e^{-z}}{e^{z} + e^{-z}}$

Rectified linear (ReLU) $\phi(z) = \begin{cases} z & \text{if } z \geq 0 \\ 0 & \text{if } z < 0 \end{cases}$

Leaky rectified linear $\phi(z) = \left\{\begin{matrix} z & \text{if } z \geq 0 \\ \alpha z & \text{if } z < 0 \end{matrix}\right\},$

(Leaky ReLU) where $\alpha > 0$ and $\alpha \approx 0$

Softplus $\phi(z) = \ln\left(1 + e^{z}\right)$

Typical non-linear activation functions