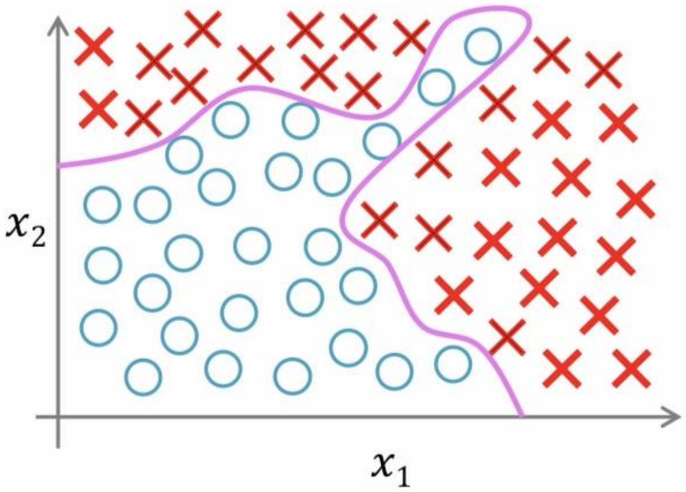
Machine Learning

By Ghazal Lalooha

Artificial Neural Networks

Motivation

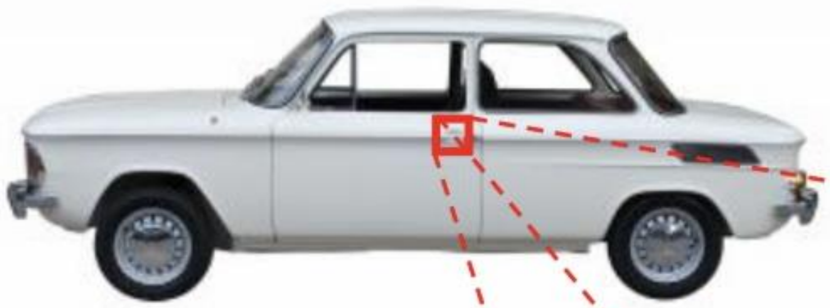


$$g(\theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_1 x_2 + \theta_4 x_1^2 x_2 + \theta_5 x_1^3 x_2 + \theta_6 x_1 x_2^2 + \cdots)$$

x_1 = size
x_2 = # bedrooms
x_3 = # floors
x_4 = age
...
x_{100}

For 100 features:
Number of quadratic
Sentences O(n^2) (5000)
The number of
sentences of the
third degree O(n^3) (170000)

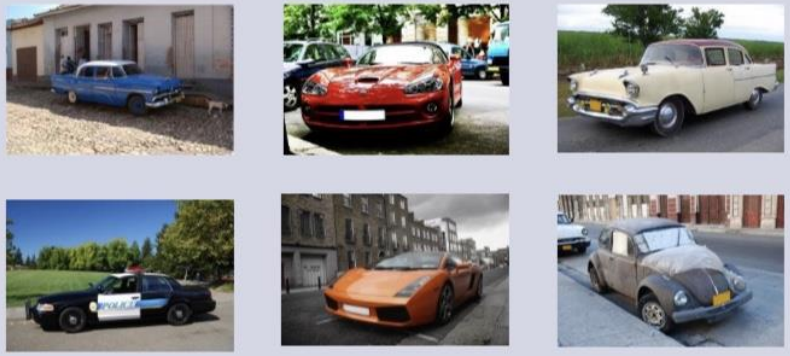
Machine vision: vehicle recognition



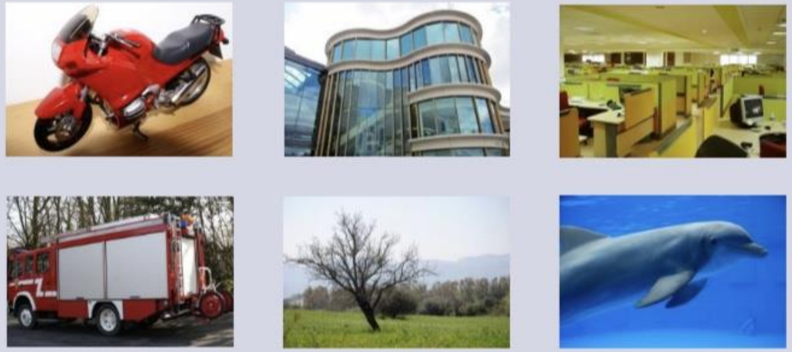
But the camera sees this:

194	210	201	212	199	213	215	195	178	158	182	209
180	189	190	221	209	205	191	167	147	115	129	163
114	126	140	188	176	165	152	140	170	106	78	88
87	103	115	154	143	142	149	153	173	101	57	57
102	112	106	131	122	138	152	147	128	84	58	66
94	95	79	104	105	124	129	113	107	87	69	67
68	71	69	98	89	92	98	95	89	88	76	67
41	56	68	99	63	45	60	82	58	76	75	65
20	43	69	75	56	41	51	73	55	70	63	44
50	50	57	69	75	75	73	74	53	68	59	37
72	59	53	66	84	92	84	74	57	72	63	42
67	61	58	65	75	78	76	73	59	75	69	50

Machine vision: vehicle recognition



Some examples of car

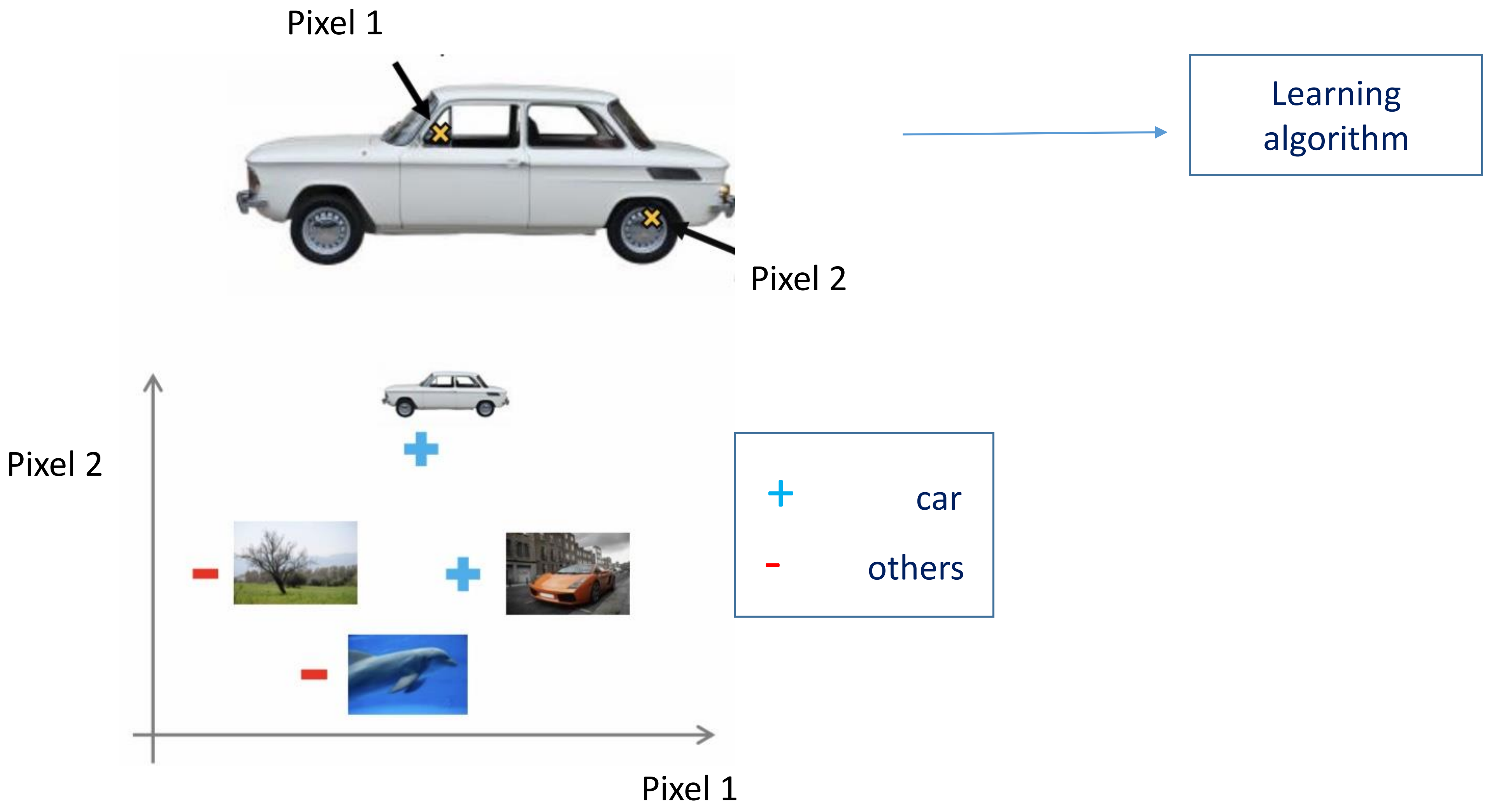


Some examples of other things

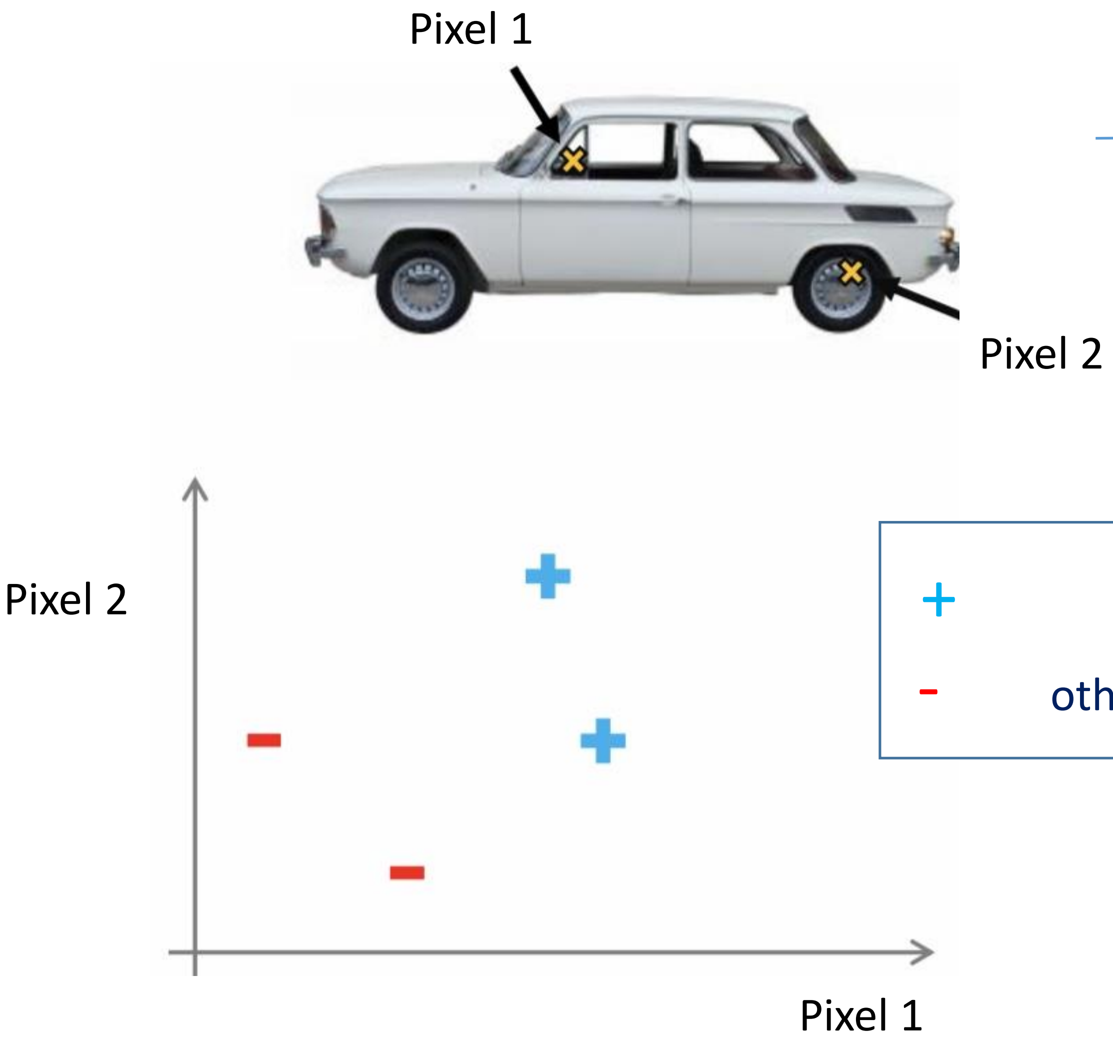
Test:

Is the above image related to a car or not?

Machine vision: vehicle recognition



Machine vision: vehicle recognition



Learning algorithm

+ car
- others

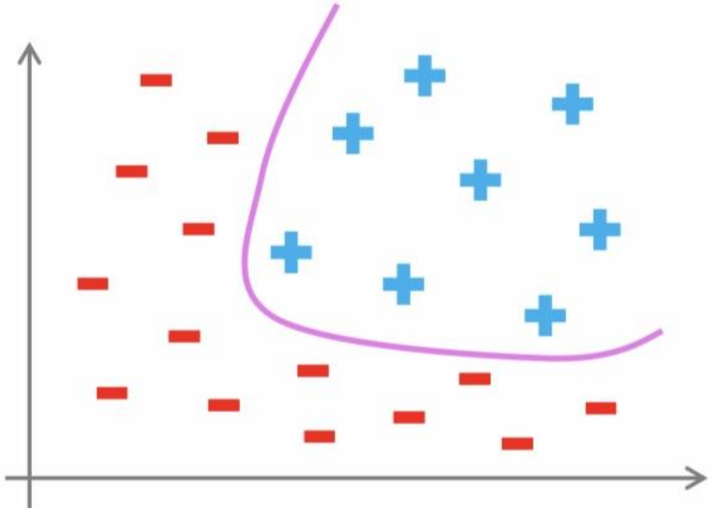
Machine vision: vehicle recognition

Pixel 1

Pixel 2

Learning algorithm

Pixel 2



Pixel 1

Images with dimensions of 50 × 50 pixels → 2500 features
(For color images: 7500 features)

$$x = \begin{bmatrix} \text{Pixel intensity 1} \\ \text{Pixel intensity 2} \\ \vdots \\ \text{Pixel intensity 2500} \end{bmatrix}$$

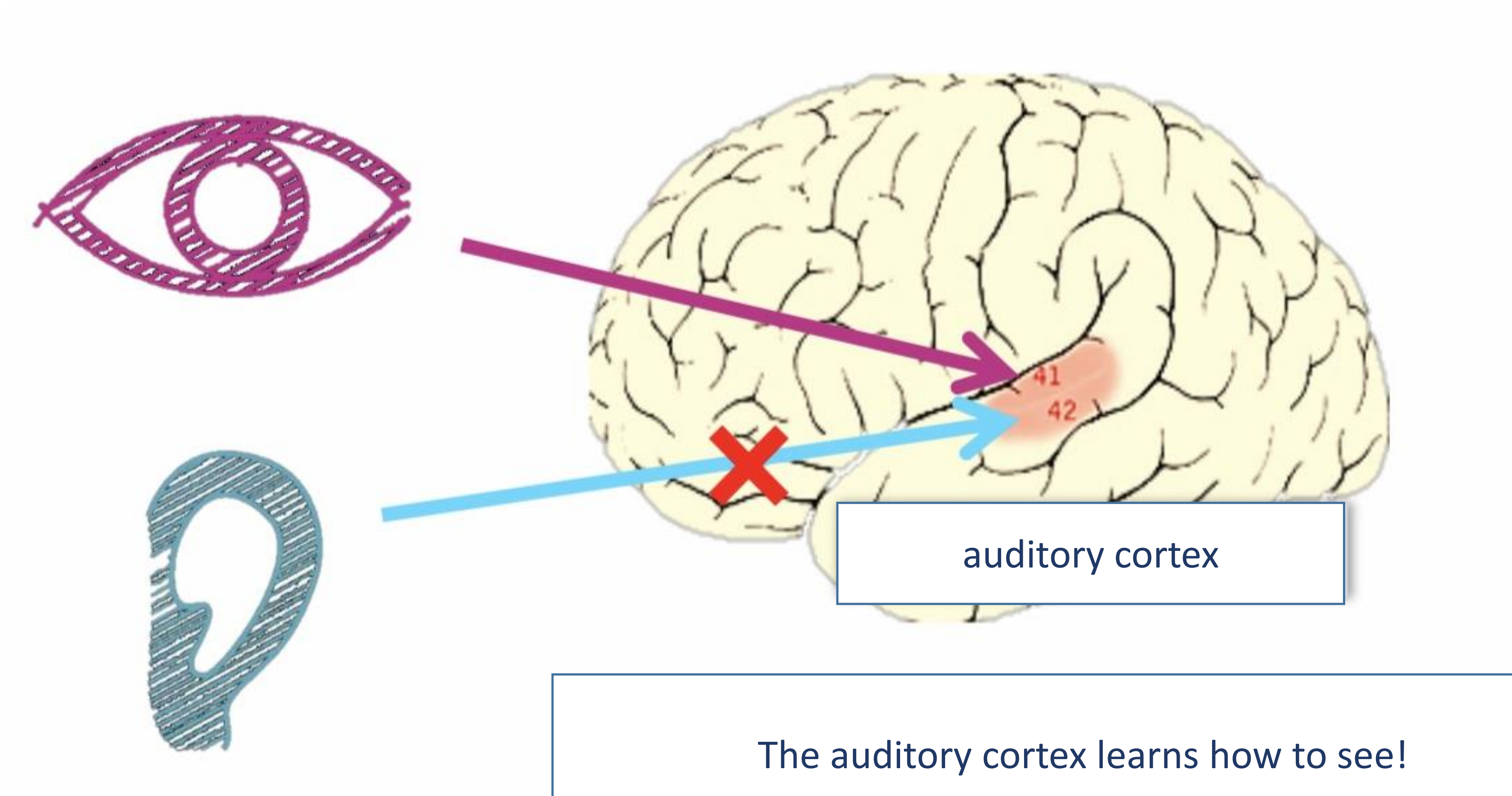
Quadratic features: almost 3 million features

Neurons and Brain

Neural Networks

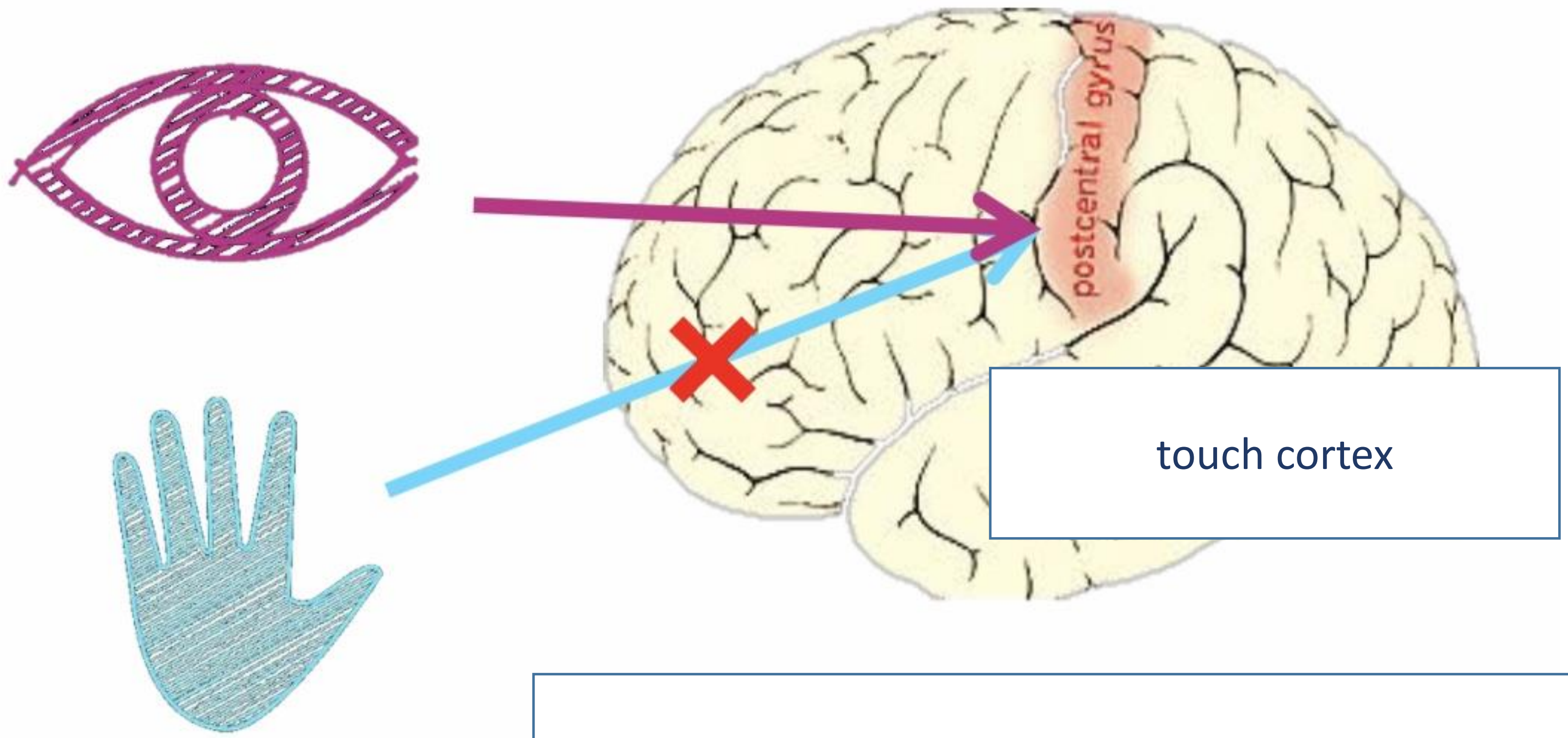
- Dating: Algorithms that try to imitate the brain.
- The use of neural networks was very common in the 80s and early 90s.
- But their popularity almost disappeared in the late 90s.
- Currently: Neural networks are currently the most advanced method for many applications.
  - Due to the increase in the speed of computers, today very large networks can be trained at high speed.

"A Learning Algorithm" Hypothesis



The auditory cortex learns how to see!

"A Learning Algorithm" Hypothesis



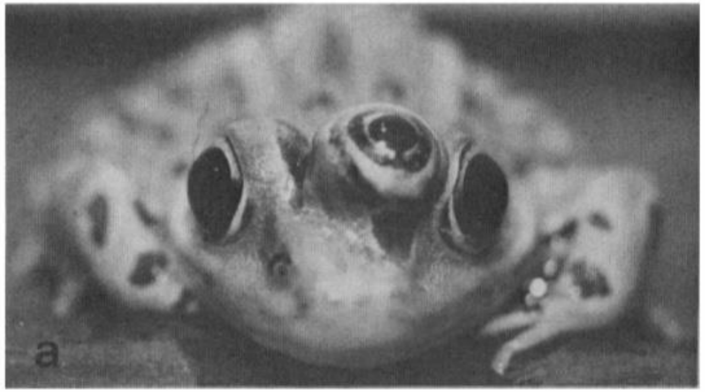
The touch cortex learns how to see!

Representation of sensors in the brain

Seeing with the tongue

Human voice localization (sonar)

Orientation belt



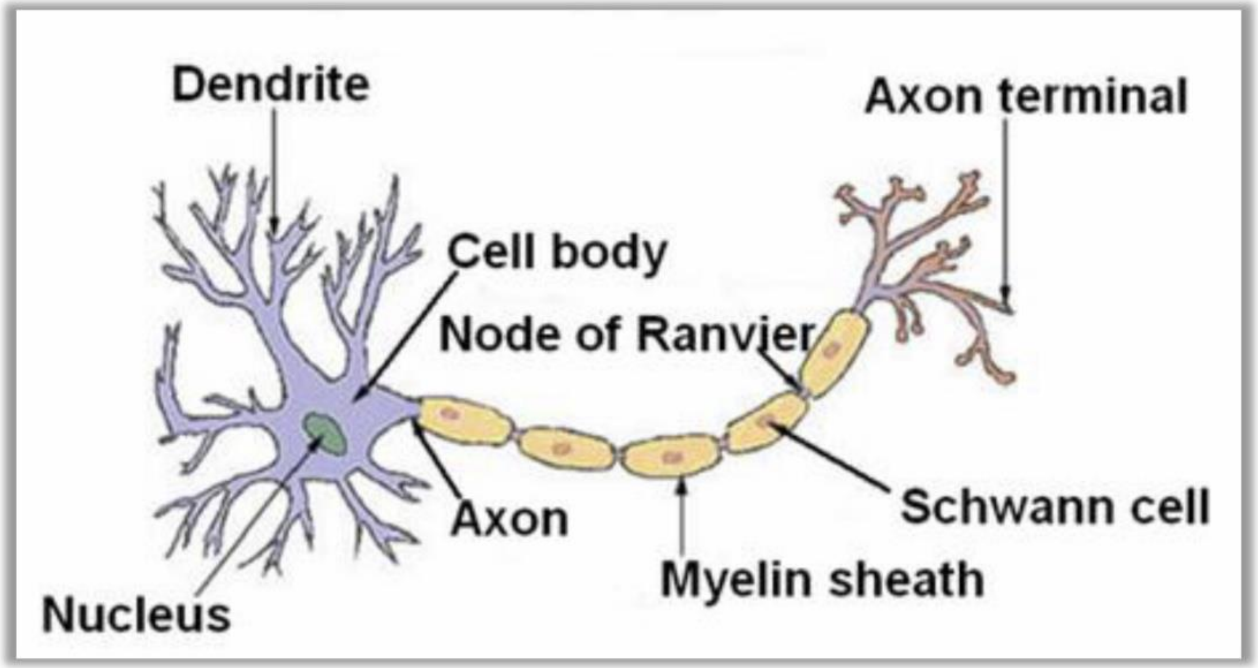
Embedding the third eye in the frog

Almost any type of sensor can be connected to the brain and the brain will learn how to use it!

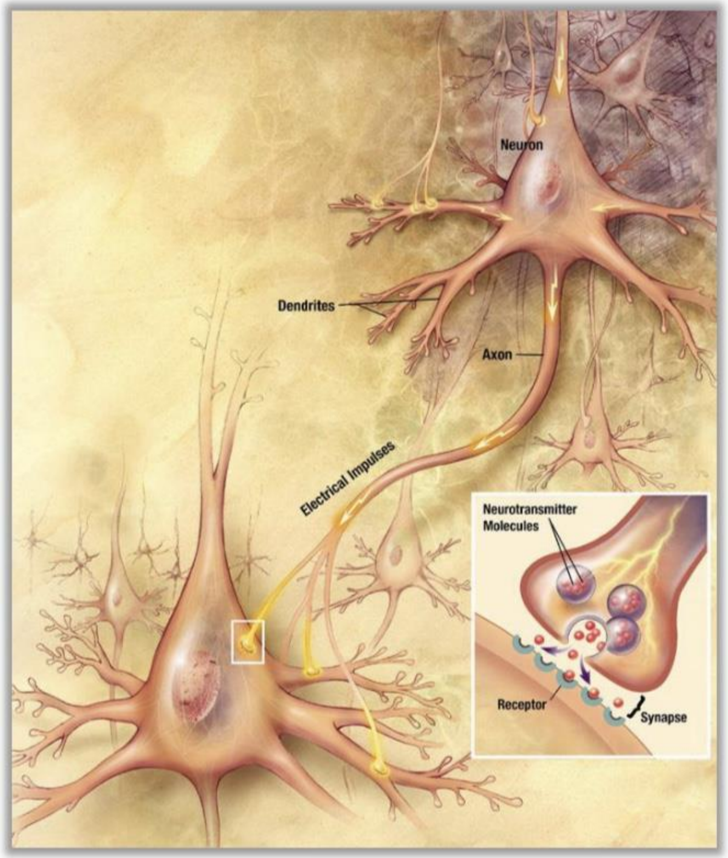
Hypothesis representation in neural networks

A Neuron's Structure

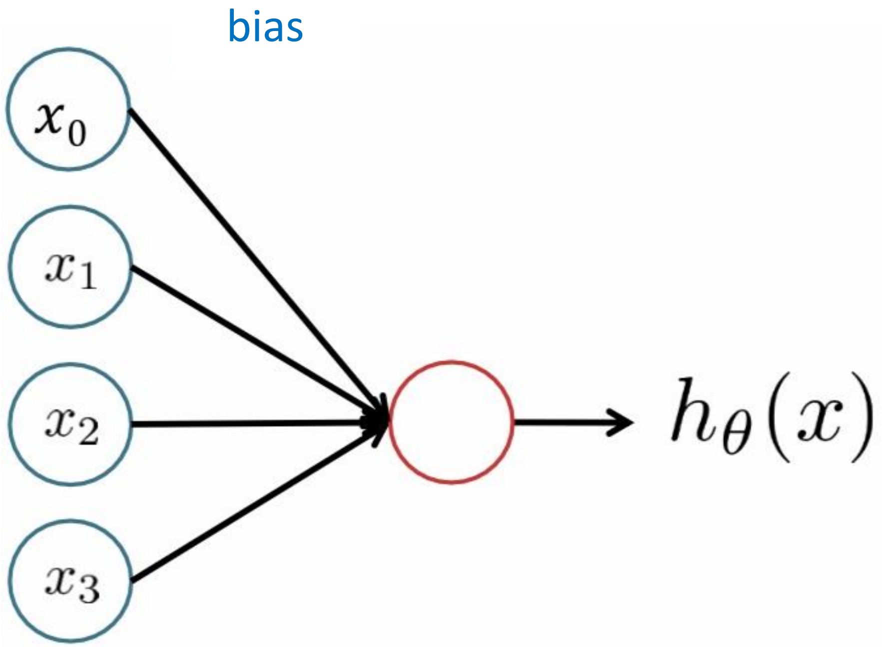
- Neuron: Each neuron is a complete computing system that receives inputs, processes them and then sends the result to the output.



Communication between neurons



Artificial Model of Neuron

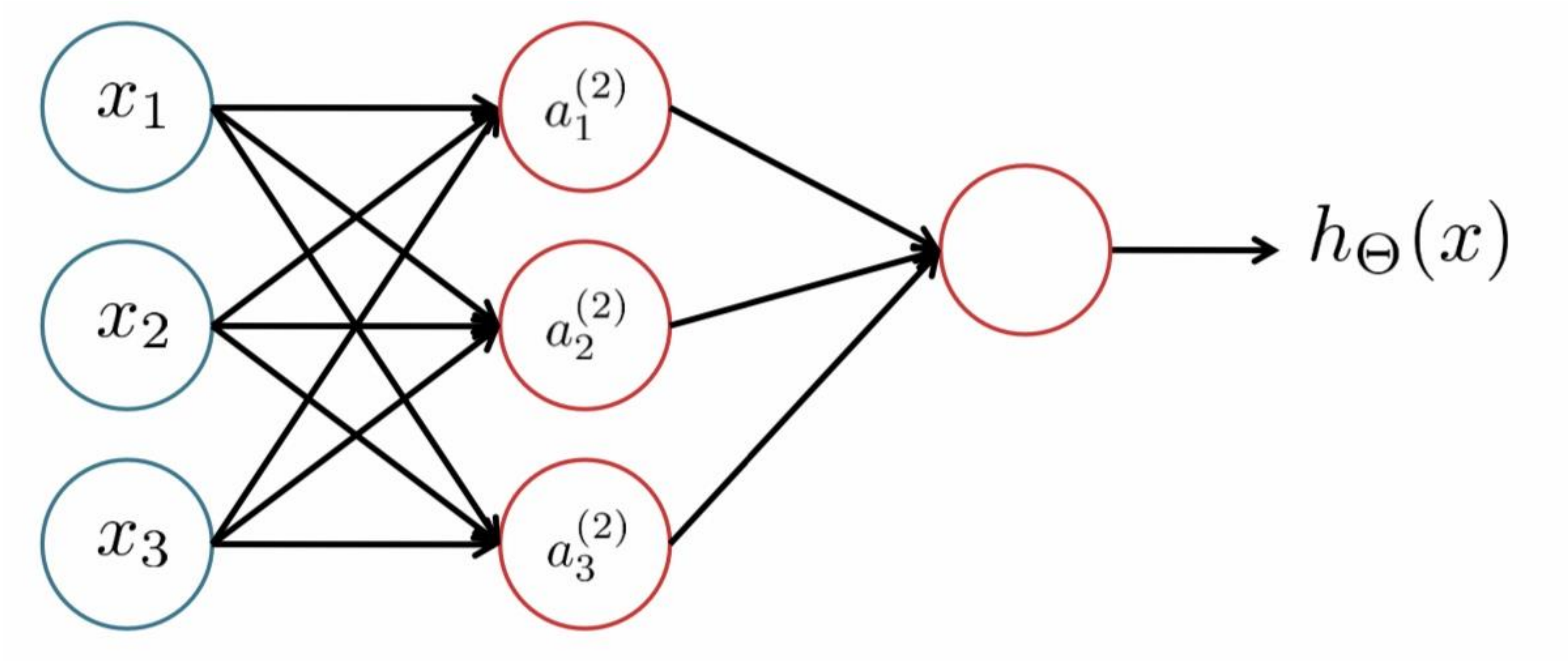


$$x = \begin{bmatrix} x_0 \\ x_1 \\ x_2 \\ x_3 \end{bmatrix} \qquad \theta = \begin{bmatrix} \theta_0 \\ \theta_1 \\ \theta_2 \\ \theta_3 \end{bmatrix}$$

Activity function: logistic sigmoid function:

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

Neural Network

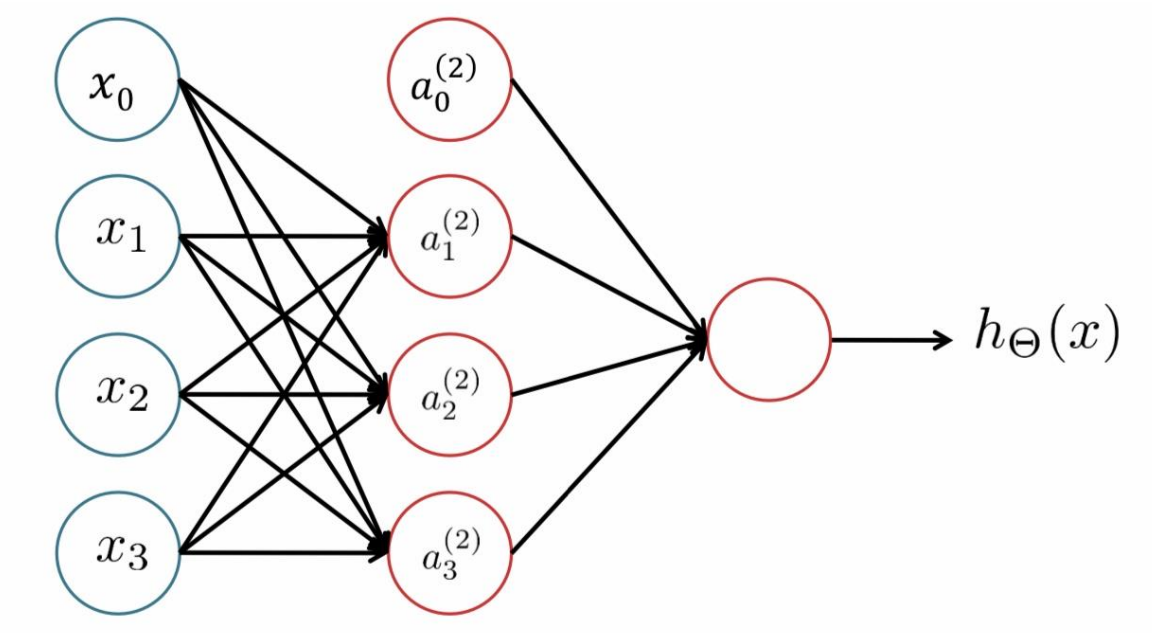


Layer 1 — Input layer

Layer 2 — hidden layer

Layer 3 — output layer

Neural Network



Layer 1 | Layer 2 | Layer 3

Input layer | hidden layer | output layer

Neural Network

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

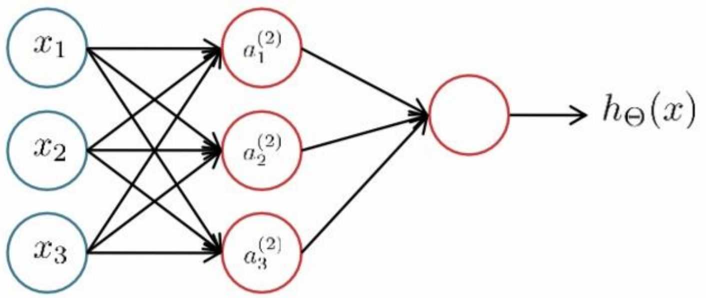
Weights matrix from layer j to layer j+1 = $\Theta^{(j)}$

$$a_1^{(2)} = g\left(\Theta_{10}^{(1)} x_0 + \Theta_{11}^{(1)} x_1 + \Theta_{12}^{(1)} x_2 + \Theta_{13}^{(1)} x_3\right)$$

$$a_2^{(2)} = g\left(\Theta_{20}^{(1)} x_0 + \Theta_{21}^{(1)} x_1 + \Theta_{22}^{(1)} x_2 + \Theta_{23}^{(1)} x_3\right)$$

$$a_3^{(2)} = g\left(\Theta_{30}^{(1)} x_0 + \Theta_{31}^{(1)} x_1 + \Theta_{32}^{(1)} x_2 + \Theta_{33}^{(1)} x_3\right)$$

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



If the network has s_j unit in layer j and s_{j+1} unit in layer j+1 , the dimensions of $\theta^{(j)}$ is equal to:

$$s_{j+1} \times (s_j + 1)$$

Neural Network

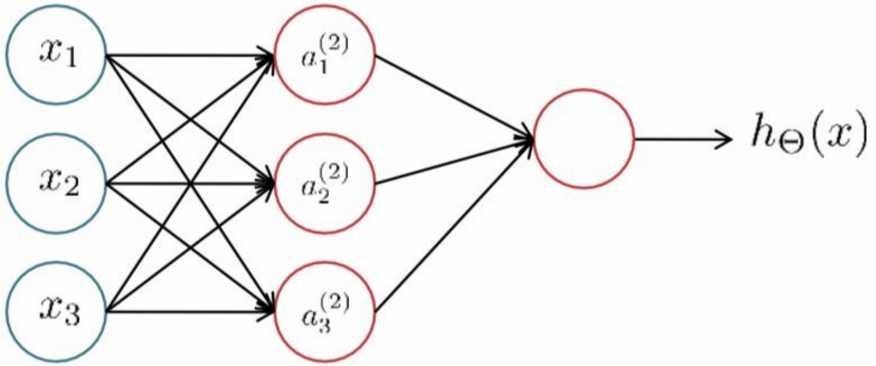
Activation of unit i in layer j = $a_i^{(j)}$

Weights matrix from layer j to layer j+1 = $\theta^{(j)}$

$$a_1^{(2)} = g\left(\Theta_{10}^{(1)} x_0 + \Theta_{11}^{(1)} x_1 + \Theta_{12}^{(1)} x_2 + \Theta_{13}^{(1)} x_3\right)$$

$$a_2^{(2)} = g\left(\Theta_{20}^{(1)} x_0 + \Theta_{21}^{(1)} x_1 + \Theta_{22}^{(1)} x_2 + \Theta_{23}^{(1)} x_3\right)$$

$$a_3^{(2)} = g\left(\Theta_{30}^{(1)} x_0 + \Theta_{31}^{(1)} x_1 + \Theta_{32}^{(1)} x_2 + \Theta_{33}^{(1)} x_3\right)$$

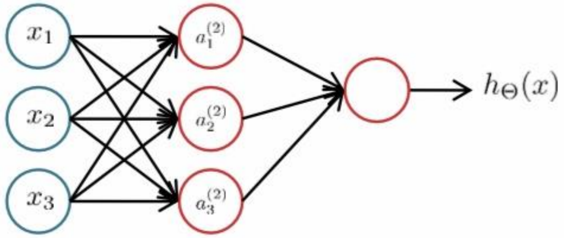


$$h_\theta(x) = a_1^{(3)} = g\left(\Theta_{10}^{(2)} a_0^{(2)} + \Theta_{11}^{(2)} a_1^{(2)} + \Theta_{12}^{(2)} a_2^{(2)} + \Theta_{13}^{(2)} a_3^{(2)}\right)$$

If the network has s_j unit in layer j and s_{j+1} unit in layer j+1 , the dimensions of $\theta^{(j)}$ is equal to:

$$s_{j+1} \times (s_j + 1)$$

Upcoming release: vector implementation



$$a_1^{(2)} = g\left(\Theta_{10}^{(1)}x_0 + \Theta_{11}^{(1)}x_1 + \Theta_{12}^{(1)}x_2 + \Theta_{13}^{(1)}x_3\right)$$

$$a_2^{(2)} = g\left(\Theta_{20}^{(1)}x_0 + \Theta_{21}^{(1)}x_1 + \Theta_{22}^{(1)}x_2 + \Theta_{23}^{(1)}x_3\right)$$

$$a_3^{(2)} = g\left(\Theta_{30}^{(1)}x_0 + \Theta_{31}^{(1)}x_1 + \Theta_{32}^{(1)}x_2 + \Theta_{33}^{(1)}x_3\right)$$

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

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

$$z^{(2)} = \Theta^{(1)}x$$

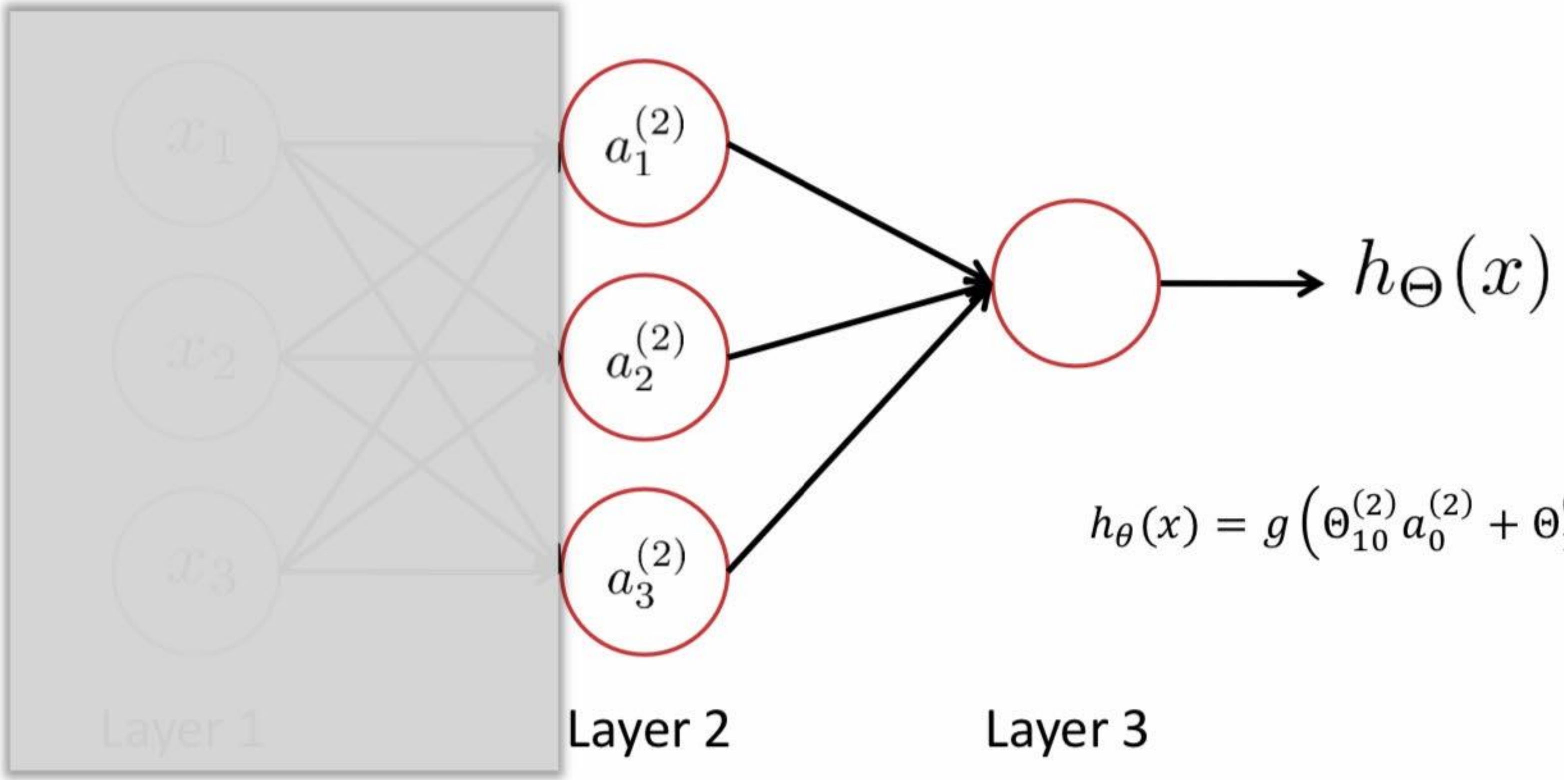
$$a^{(2)} = g(z^{(2)})$$

Adding $1 = a_0^{(2)}$

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

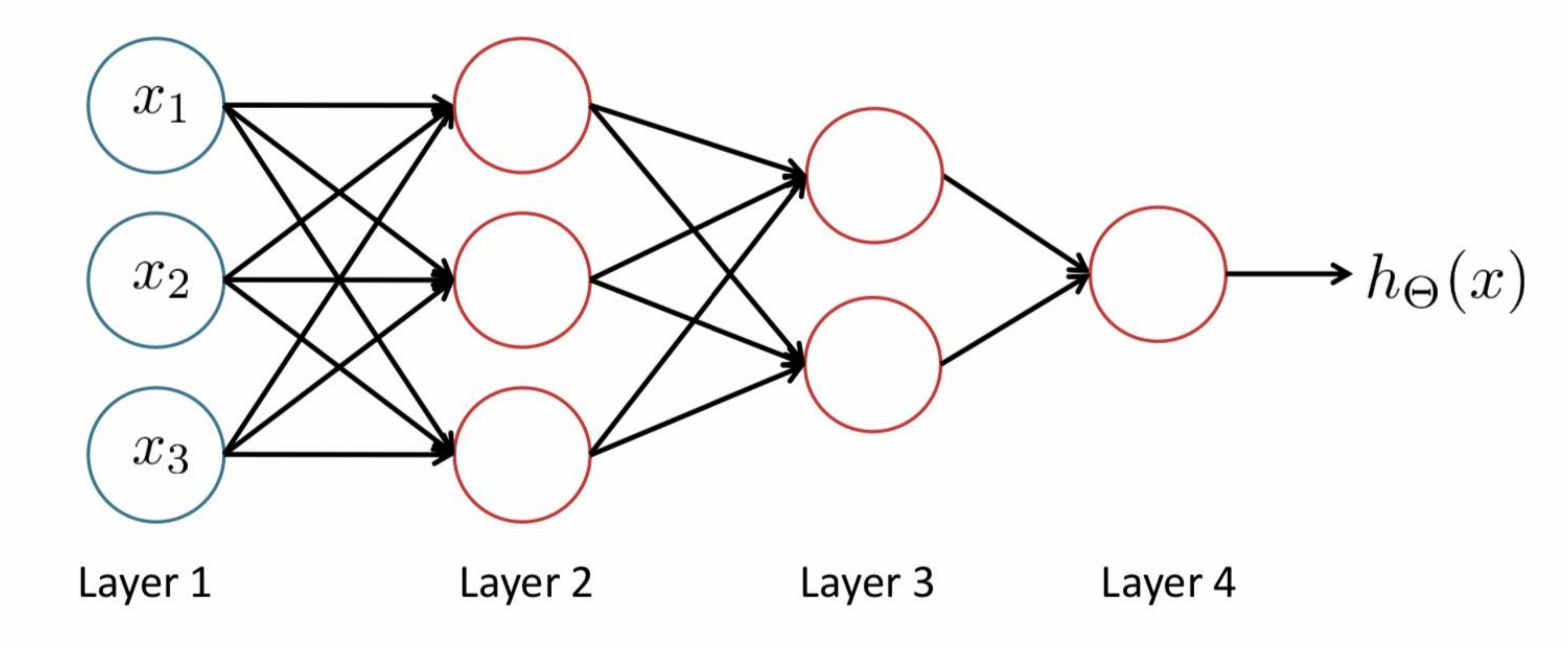
$$a^{(3)} = g(z^{(3)}) = h_\theta(x)$$

Neural Networks learn a new set of features.



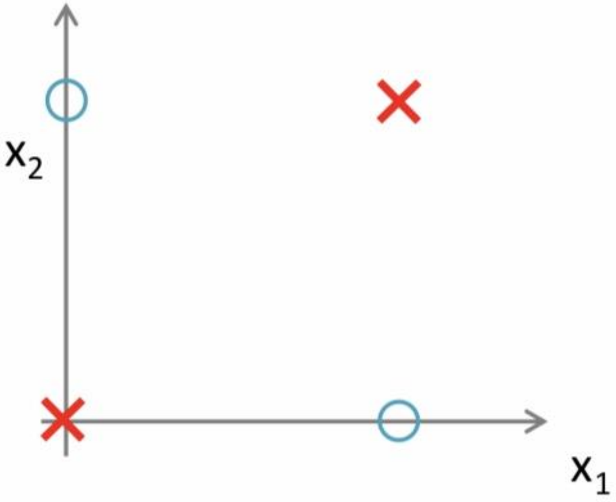
$$h_\theta(x) = g\left(\Theta_{10}^{(2)} a_0^{(2)} + \Theta_{11}^{(2)} a_1^{(2)} + \Theta_{12}^{(2)} a_2^{(2)} + \Theta_{13}^{(2)} a_3^{(2)}\right)$$

Other kinds of network architecture

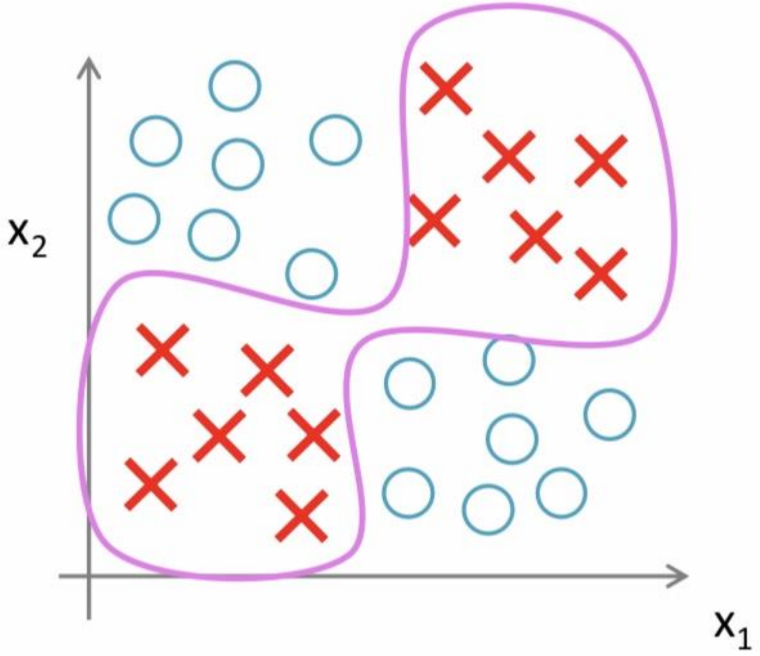


Examples

Non-linear classification: XOR/XNOR function



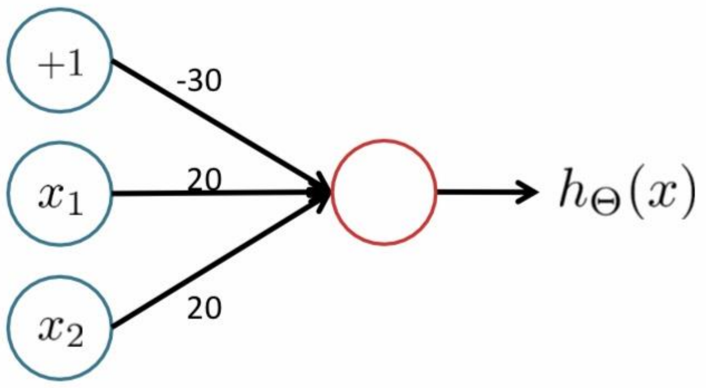
$$y = x_1 \text{ XOR } x_2$$
$$x_1 \text{ XNOR } x_2$$
$$\text{NOT } (x_1 \text{ XOR } x_2)$$



A simple example: AND function

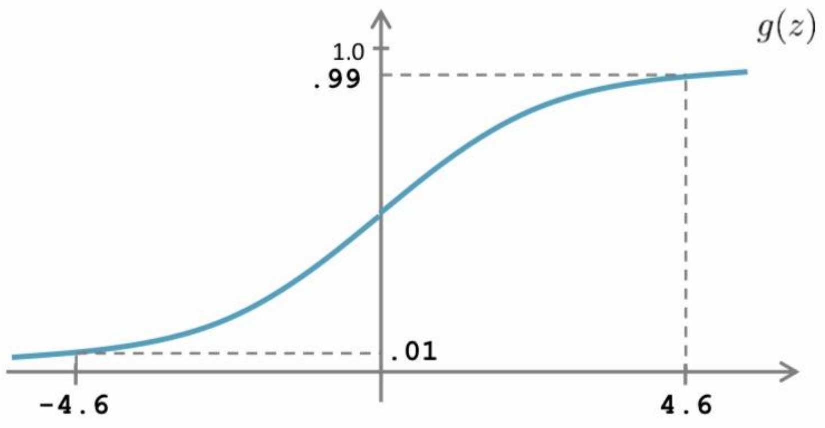
$x_1, x_2 \in \{0, 1\}$

$y = x_1 \text{ AND } x_2$



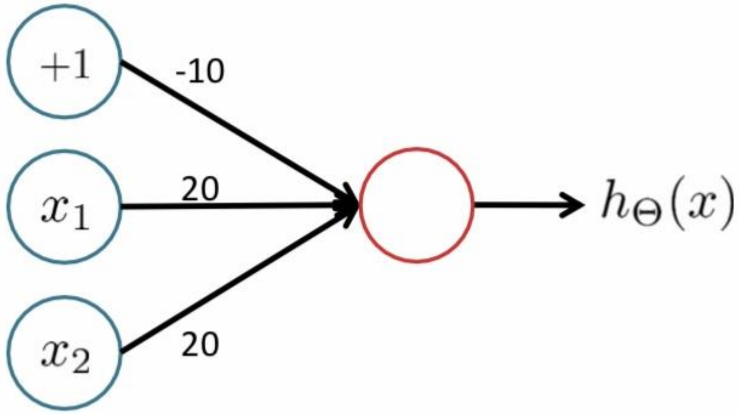
$$h_\Theta(x) = g(-30 + 20x_1 + 20x_2)$$

$$\Theta^{(1)}_{10} \quad \Theta^{(1)}_{11} \quad \Theta^{(1)}_{12}$$



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: OR function

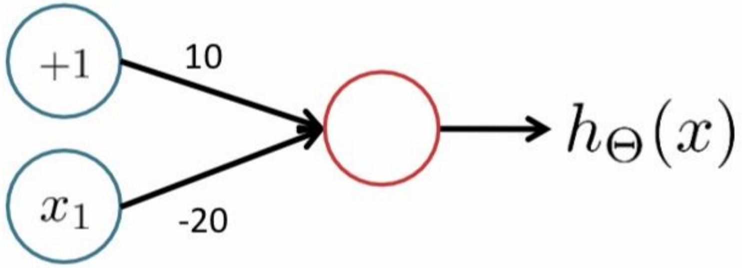


$$h_\Theta(x) = g(-10 + 20x_1 + 20x_2)$$

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	$g(30) \approx 1$

Inverse function

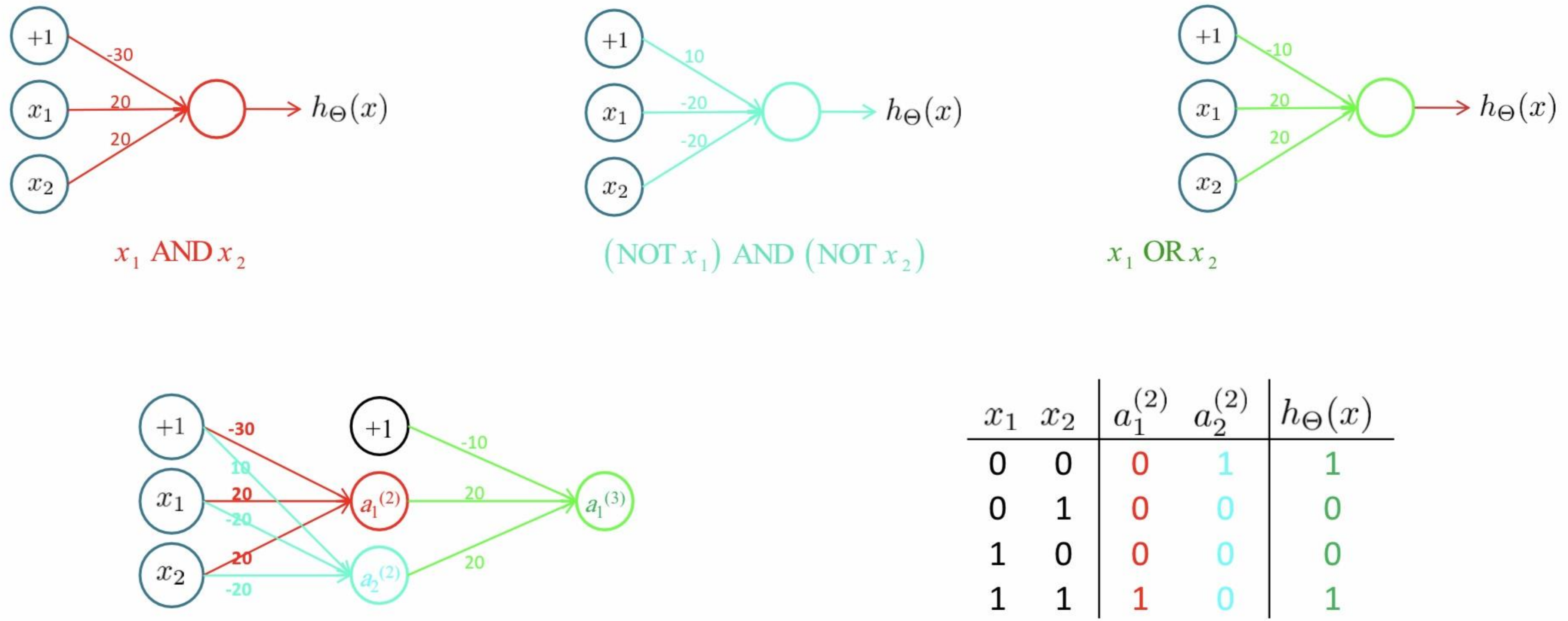
- Inverse



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

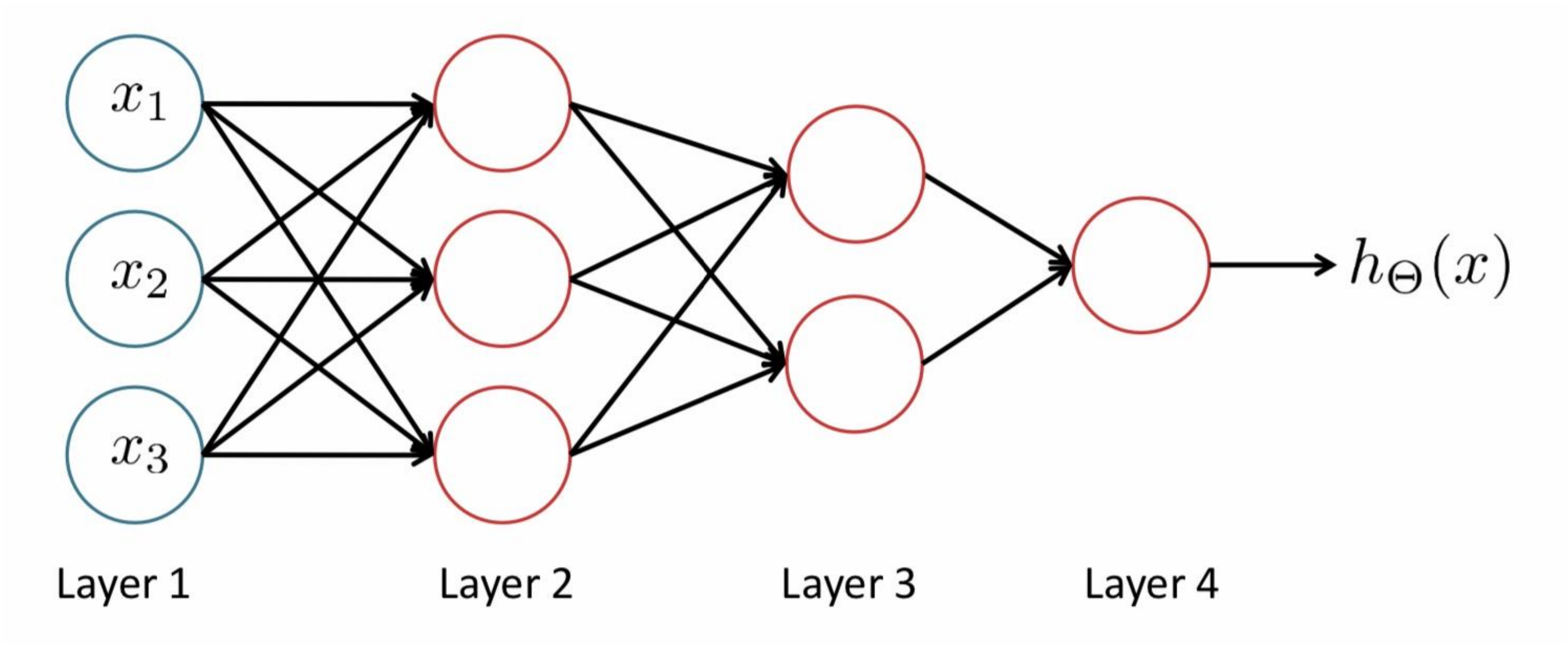
$$h_\theta(x) = g(10 - 20x_1)$$

Example: XNOR function



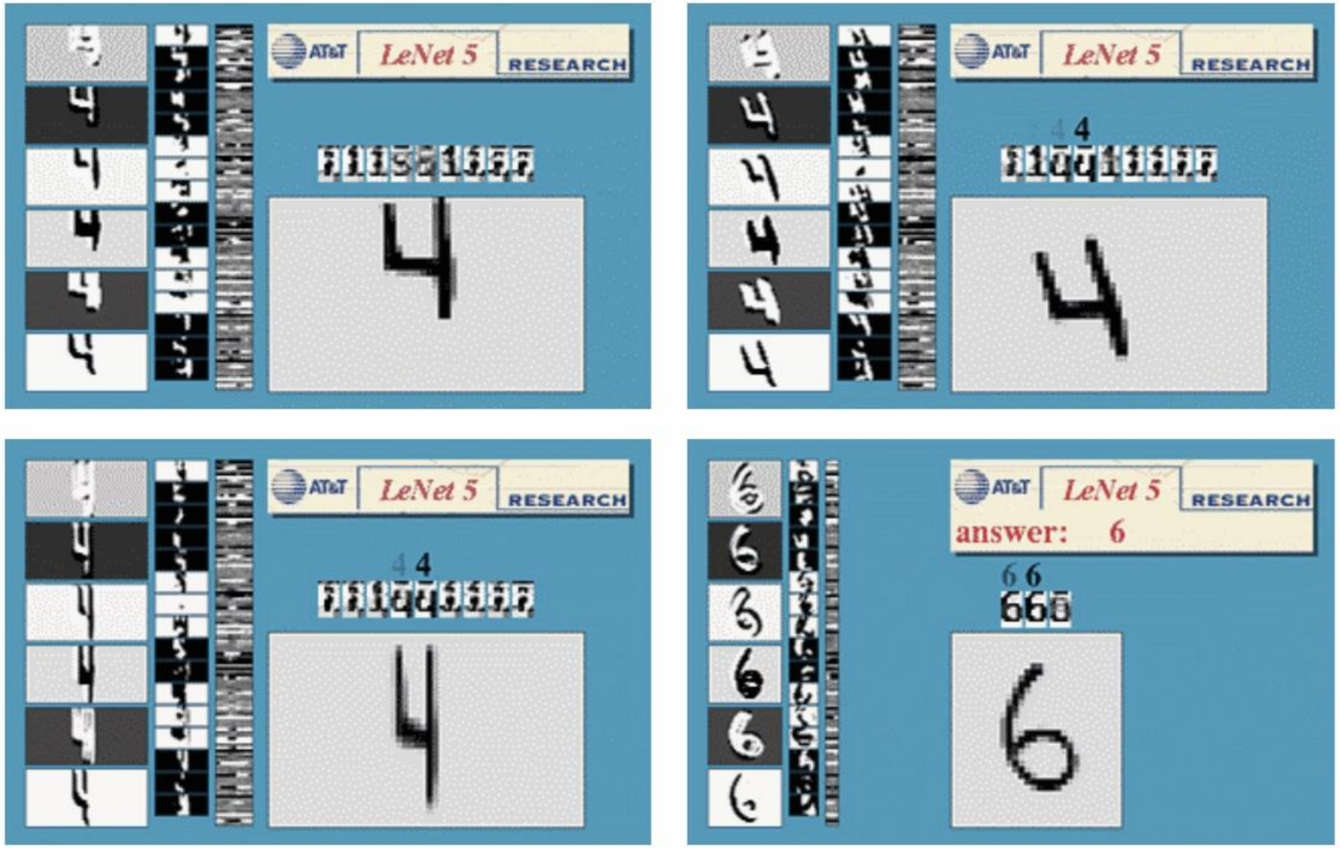
x_1	x_2	$a_1^{(2)}$	$a_2^{(2)}$	$h_\Theta(x)$
0	0	0	1	1
0	1	0	0	0
1	0	0	0	0
1	1	1	0	1

Computing more complex functions

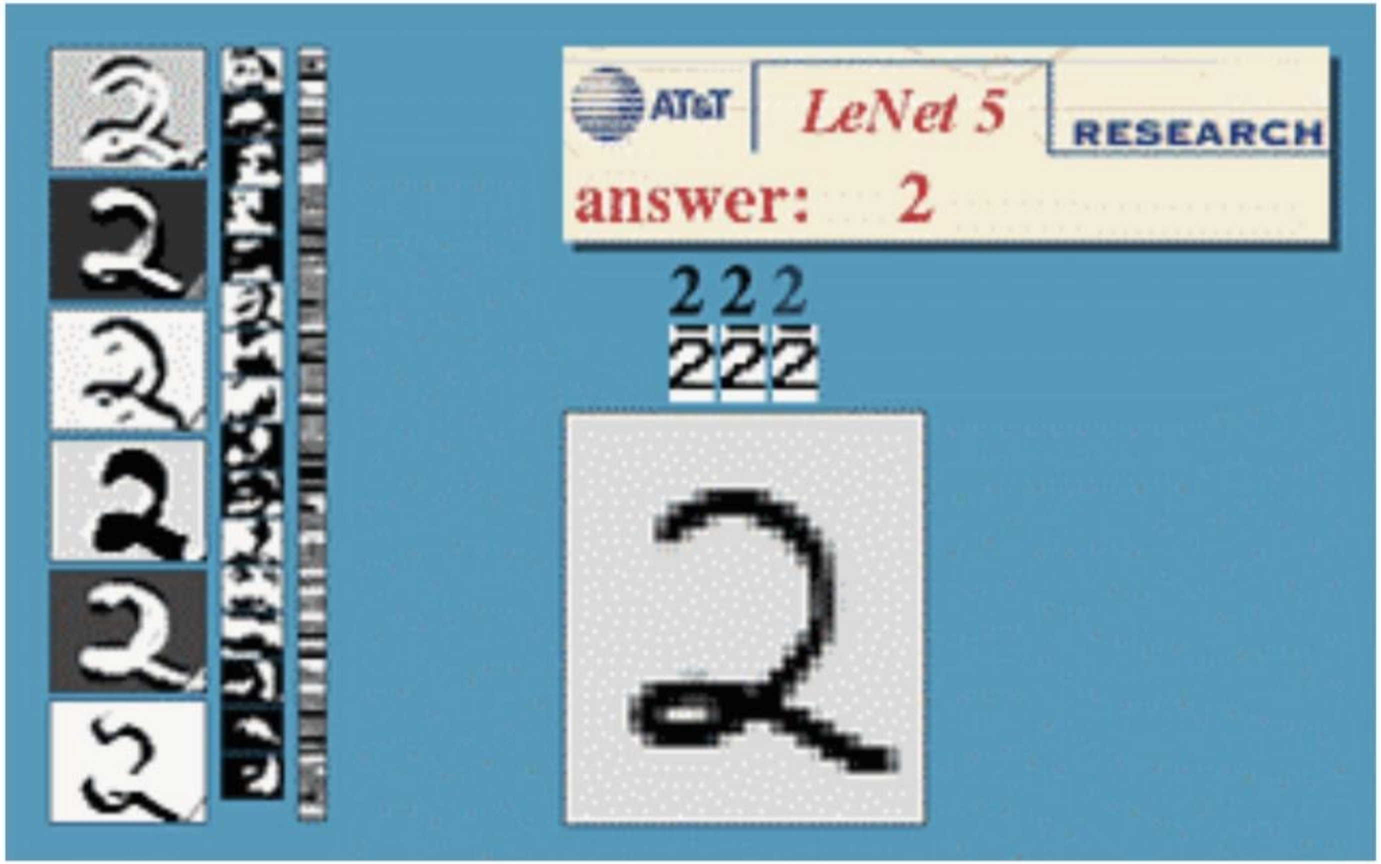


Recognition of handwritten figures [Yann LeCun]

Move, rotate, resize operators

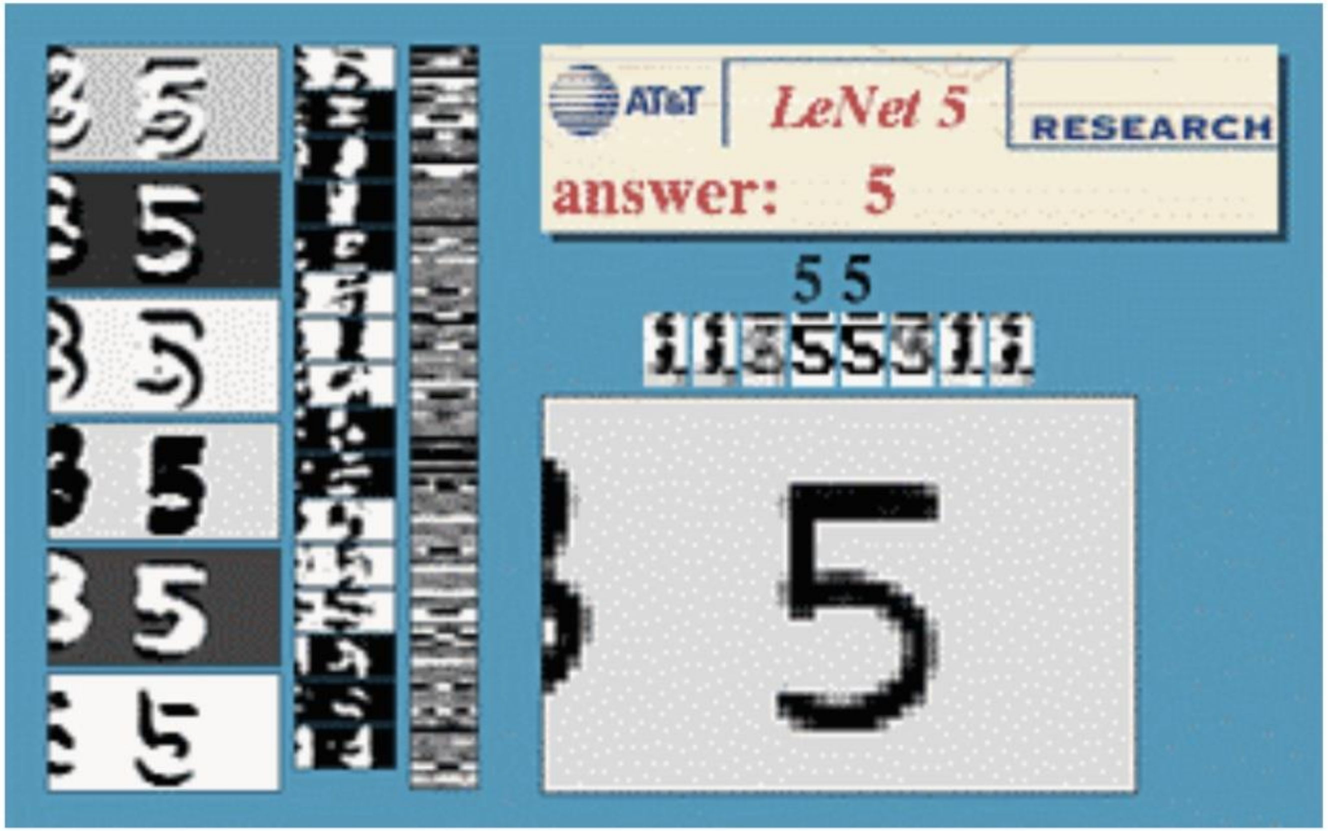


Noise resistance



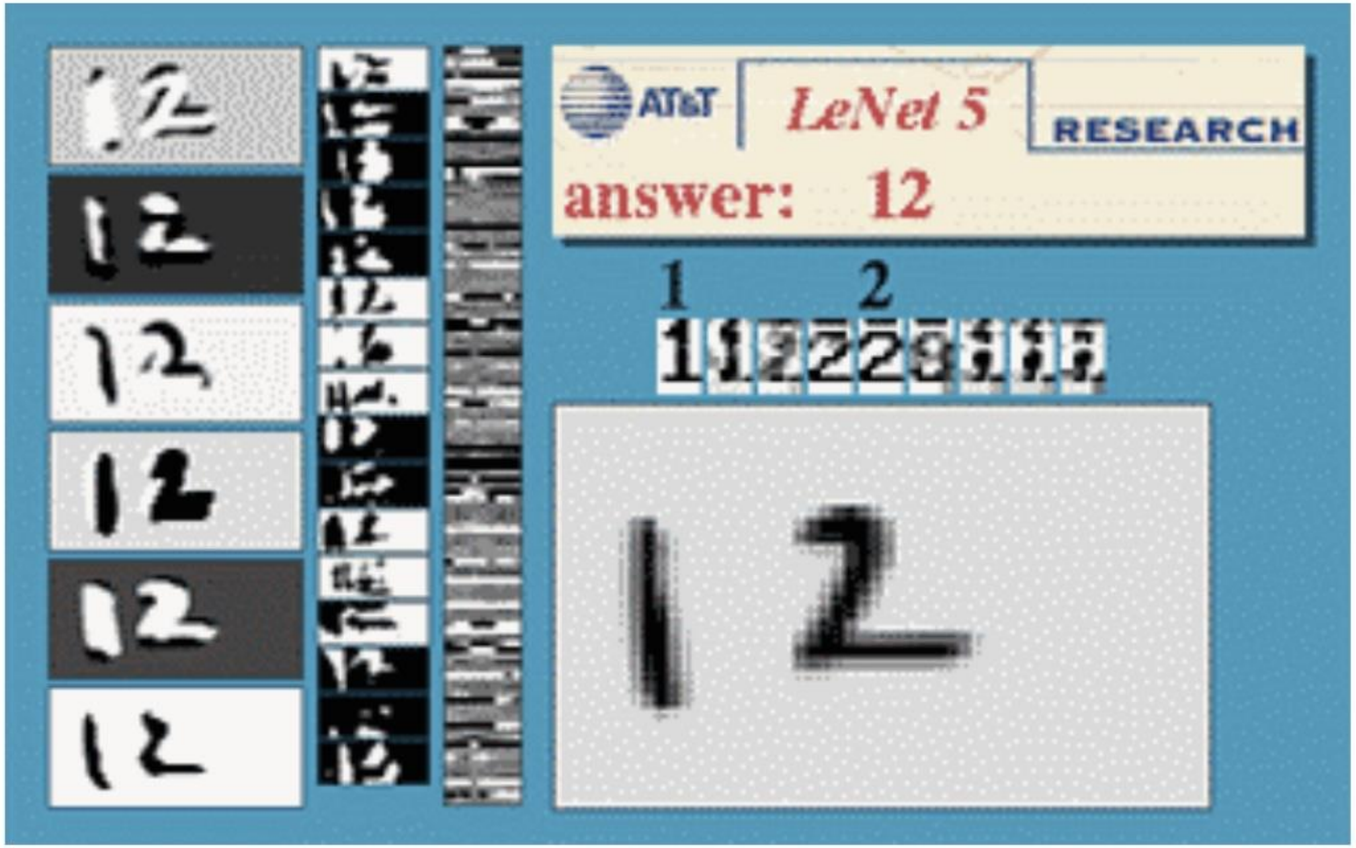
Multiple character recognition

More complex examples

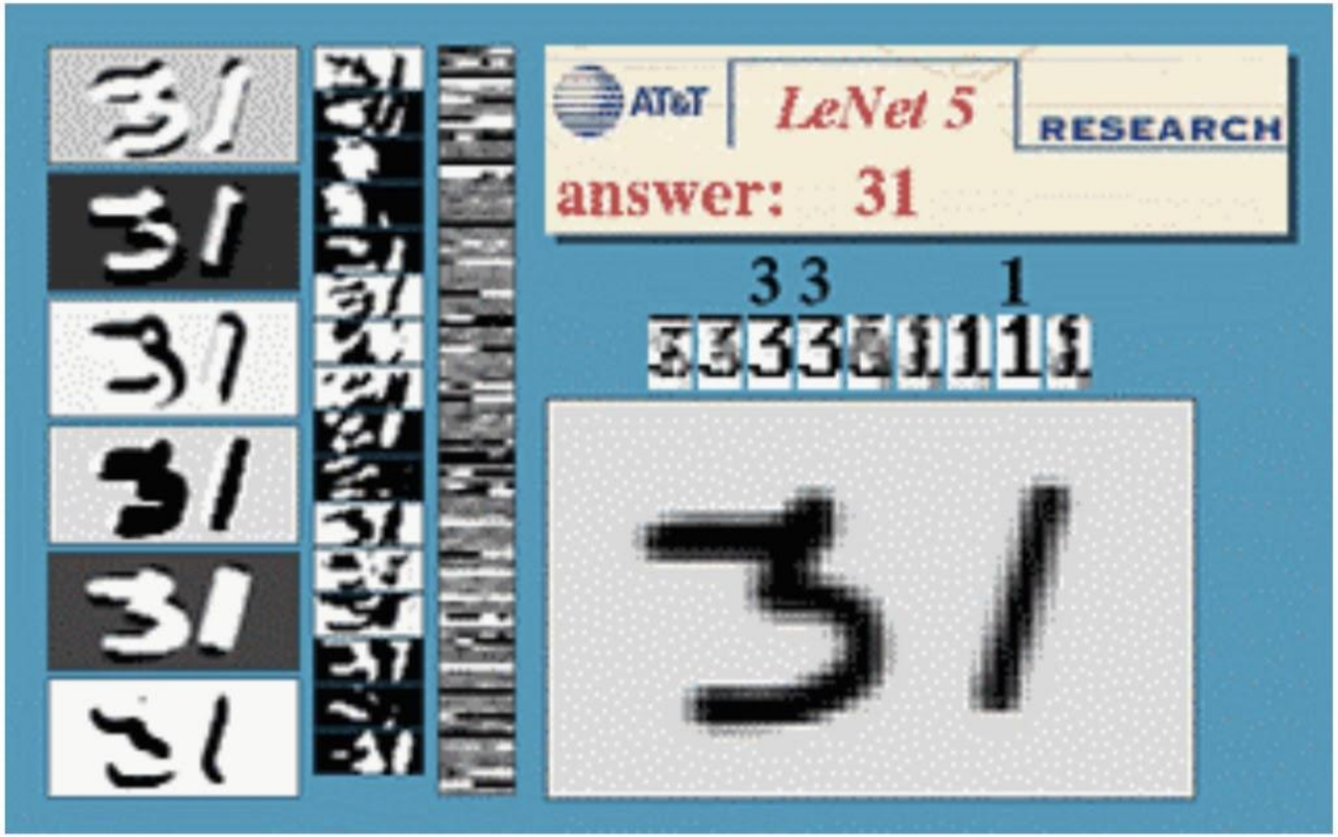


More complex examples

More complex examples



More complex examples



Multiclass classification

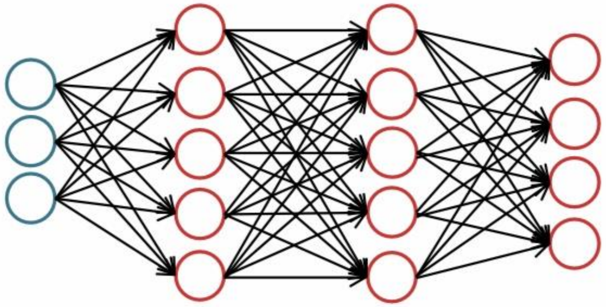
Many output unit: one to many


passerby


car


motorcycle


truck



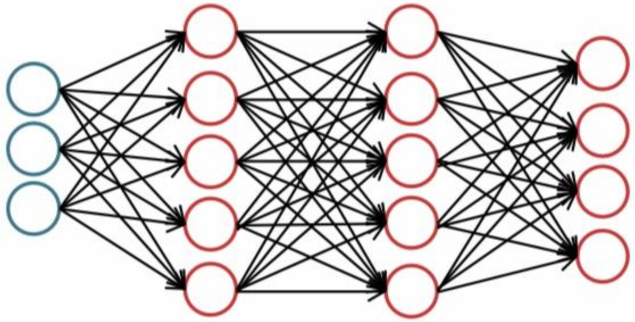
$$h_\Theta(x) \in \mathbb{R}^4$$

$$h_\Theta(x) \approx \begin{bmatrix} 1 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$
passerby

$$h_\Theta(x) \approx \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}$$
car

$$h_\Theta(x) \approx \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \end{bmatrix}$$
motorcycle

Many output unit: one to many



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

$h_\Theta(x) \approx \begin{bmatrix}1\\0\\0\\0\end{bmatrix}$ $h_\Theta(x) \approx \begin{bmatrix}0\\1\\0\\0\end{bmatrix}$ $h_\Theta(x) \approx \begin{bmatrix}0\\0\\1\\0\end{bmatrix}$

passerby car motorcycle

Training set: $(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \ldots, (x^{(m)}, y^{(m)})$

$y^{(i)}$ $\begin{bmatrix}1\\0\\0\\0\end{bmatrix}$ $\begin{bmatrix}0\\1\\0\\0\end{bmatrix}$ $\begin{bmatrix}0\\0\\1\\0\end{bmatrix}$ $\begin{bmatrix}0\\0\\0\\1\end{bmatrix}$

passerby car motorcycle truck