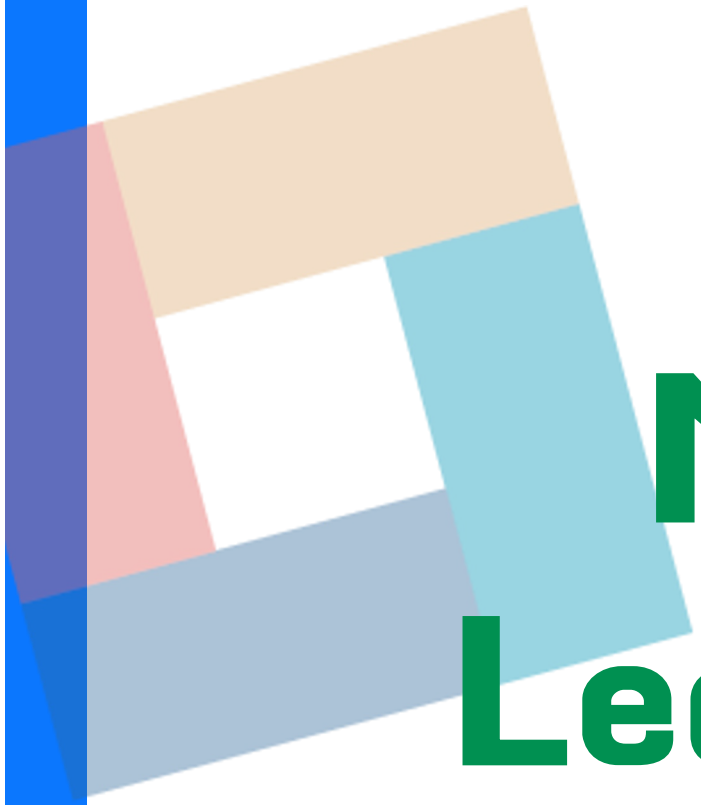


Machine Learning 06

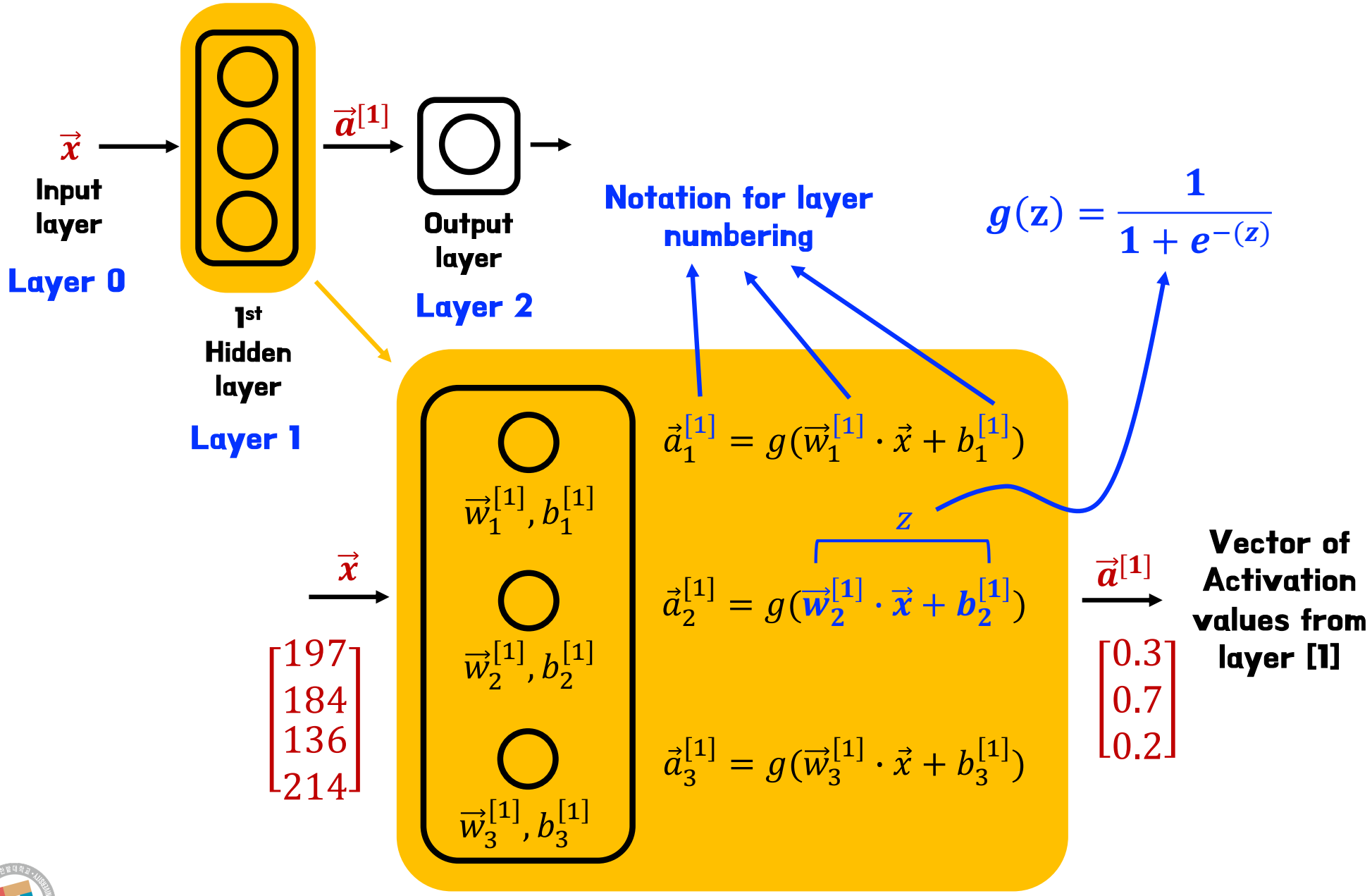
Kihyun Shin
DMSE, HBNU

TheMAD

Neural network layer

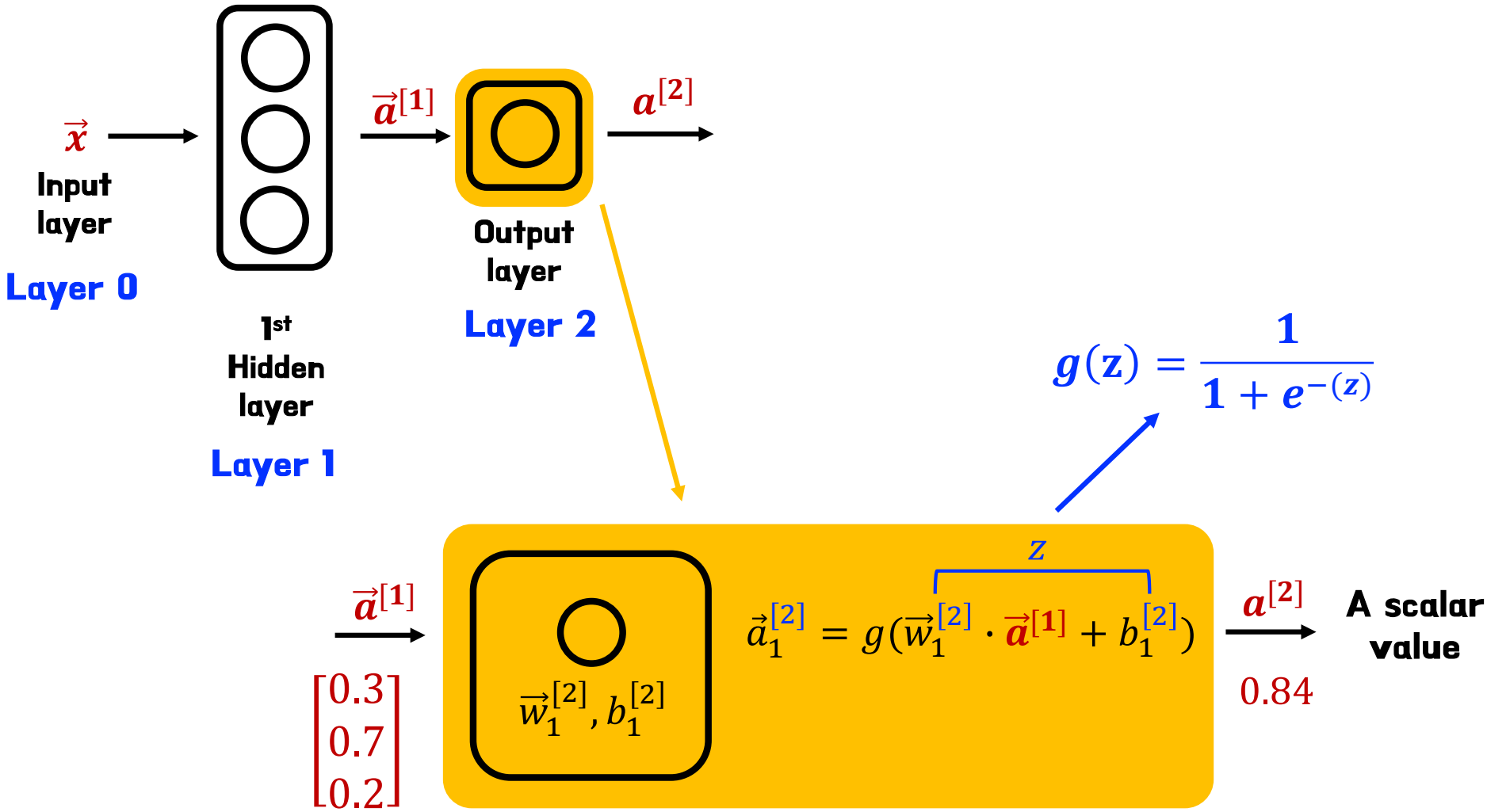
TheMAD

Neural network layer

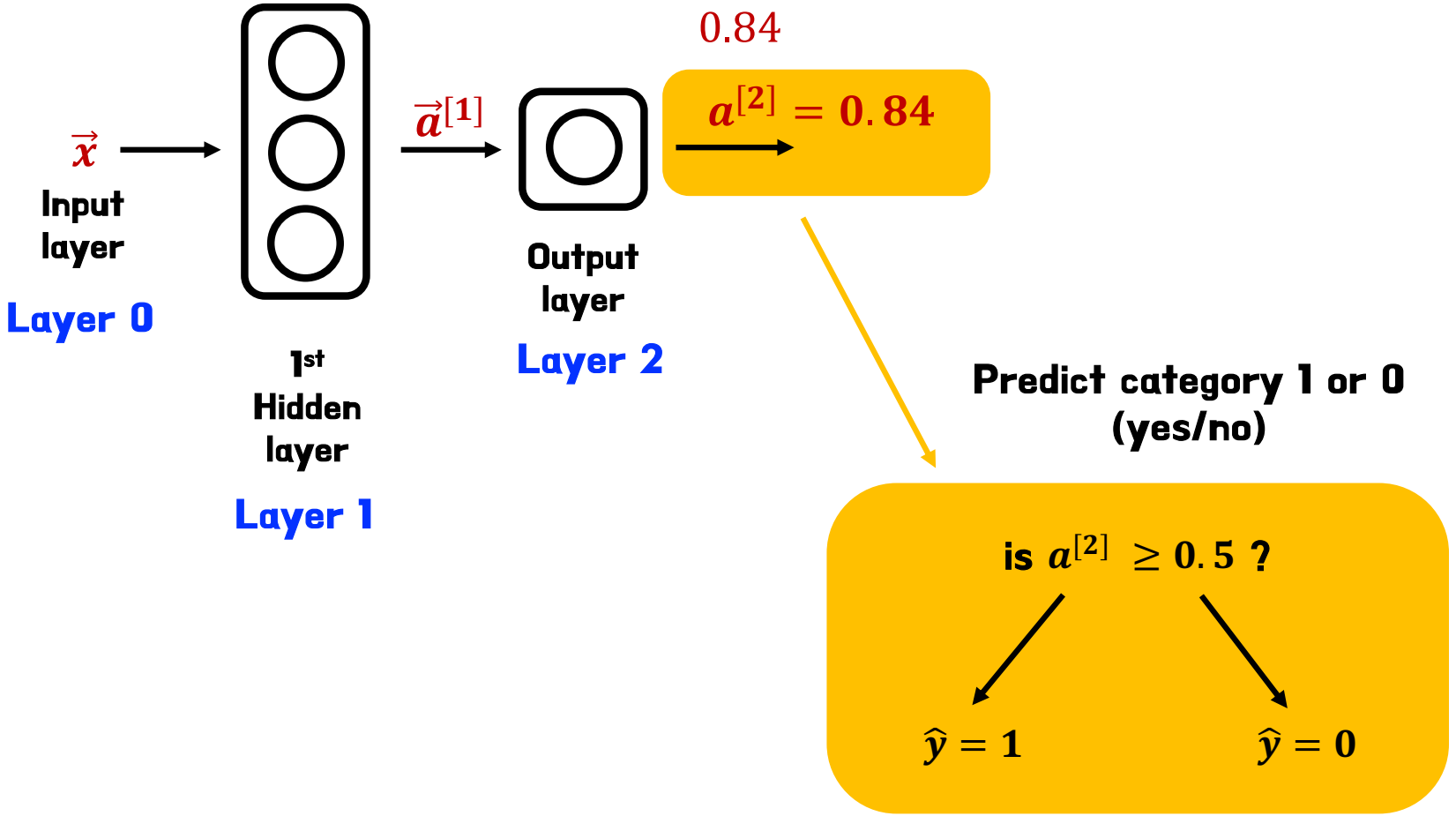


TheMAC

Neural network layer

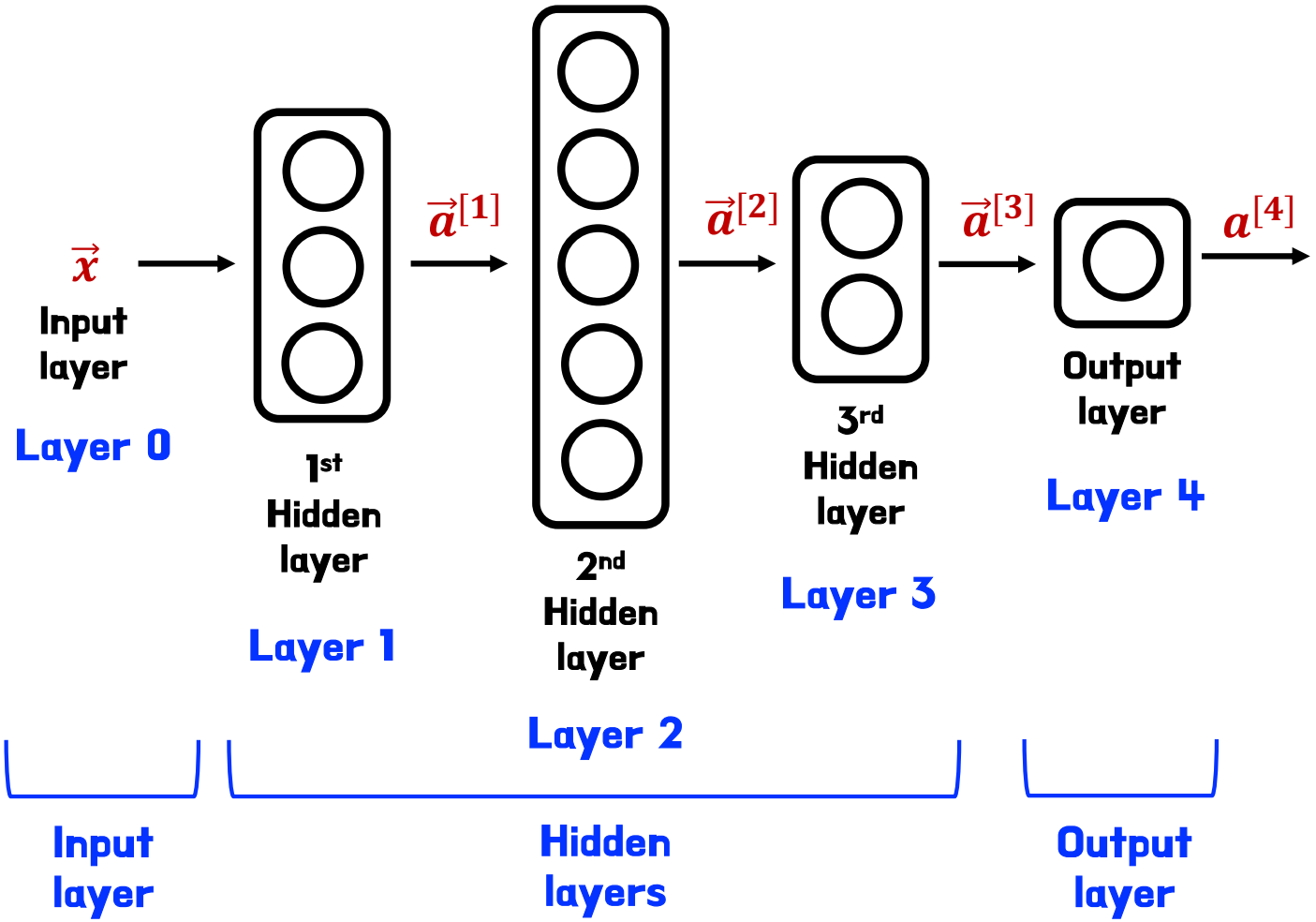


Neural network layer



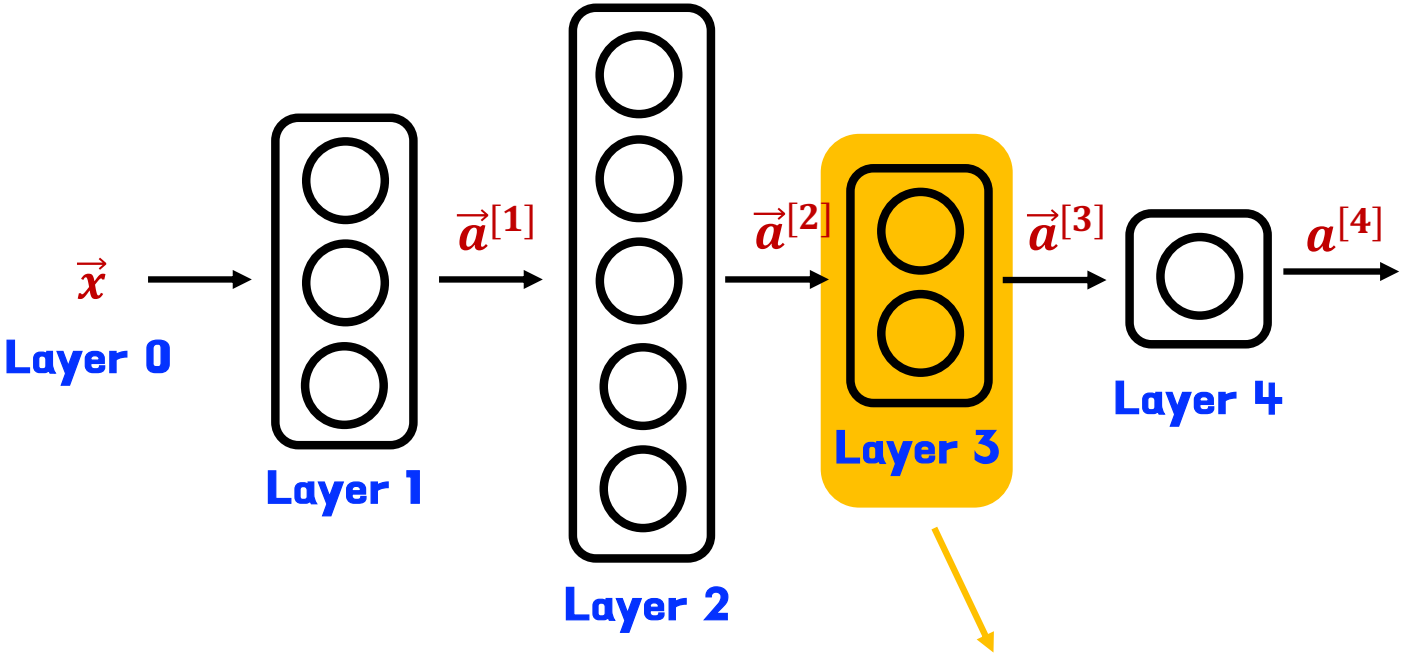
TheMAC

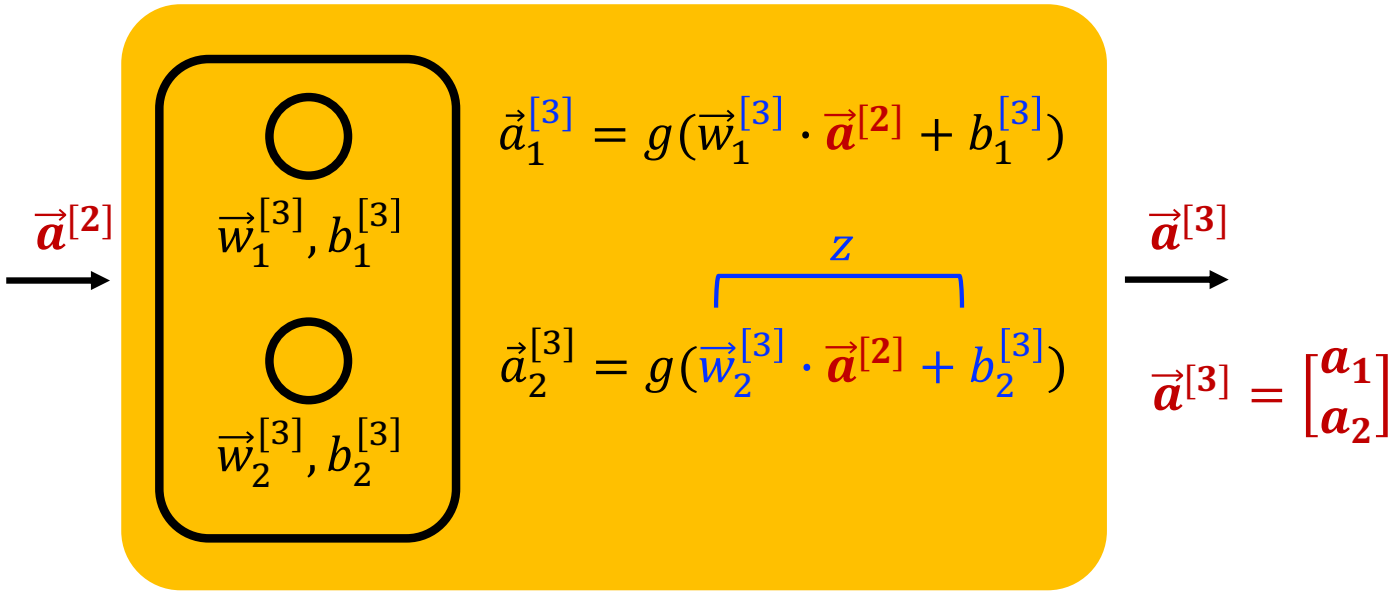
More complex neural network



TheMAD

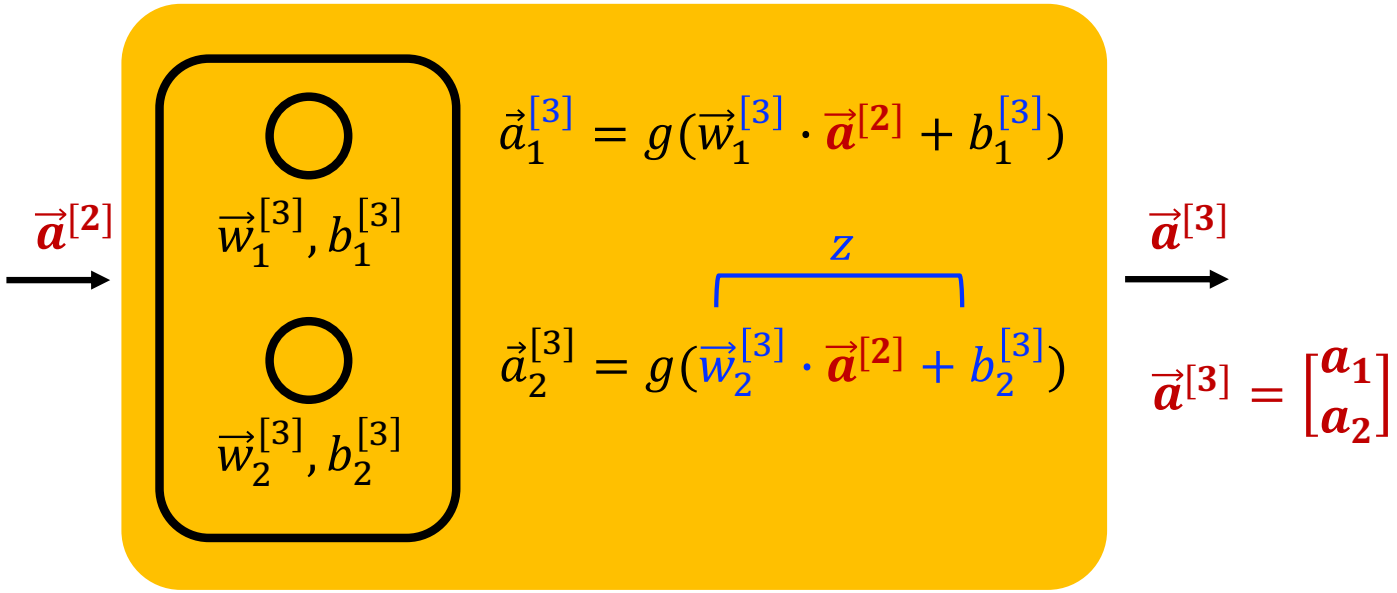
More complex neural network





TheMAC

More complex neural network



Activation value of layer l, unit (neuron) j

Output of layer $l-1$ (previous layer)

$$\vec{a}_j^{[l]} = g(\vec{w}_j^{[l]} \cdot \vec{a}^{[l-1]} + b_j^{[l]})$$

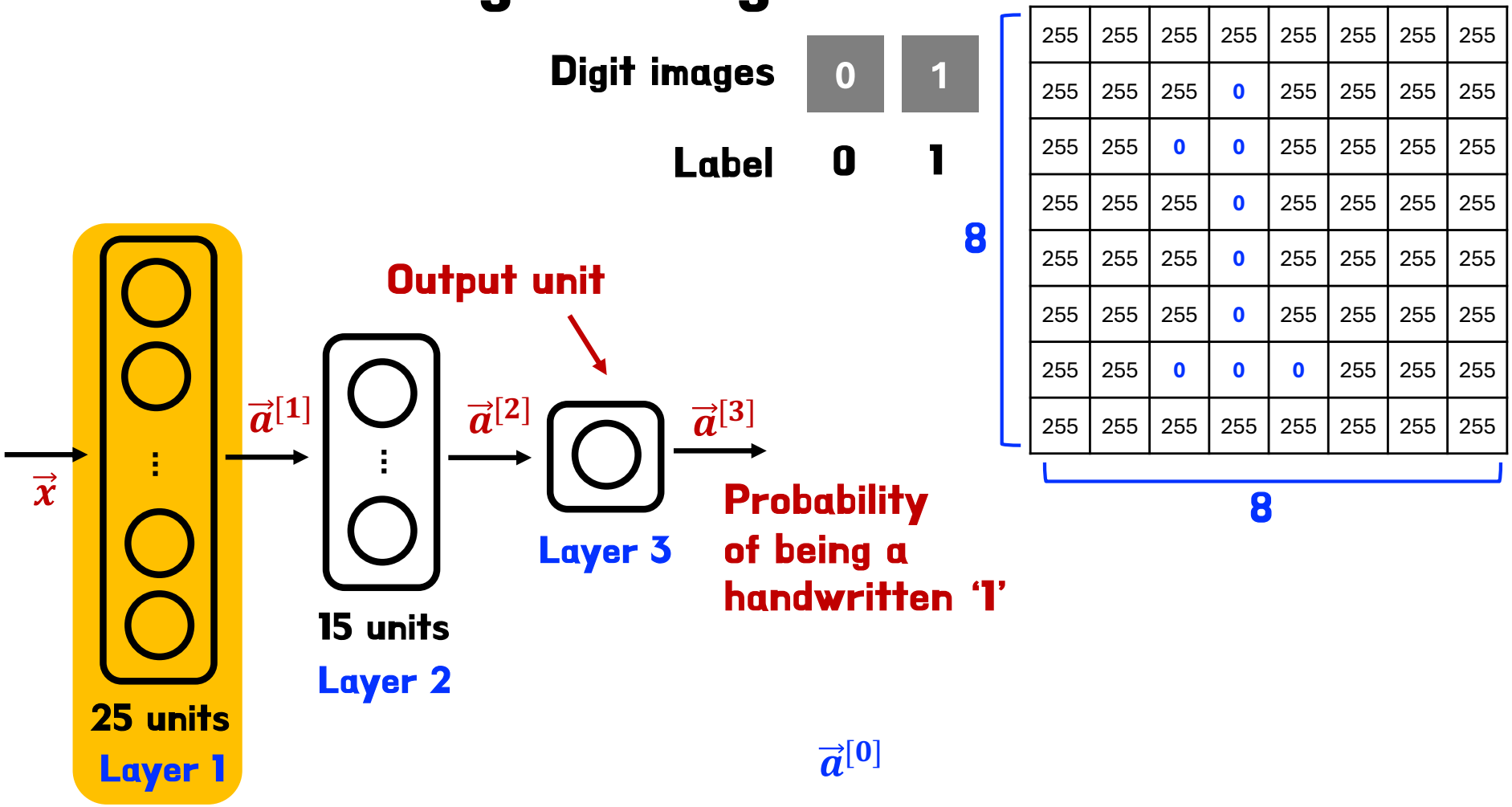
Sigmoid "activation function"

Parameters w & b of layer l, unit j

TheMAD

Inference: making predictions (forward propagation)

Handwritten digit recognition

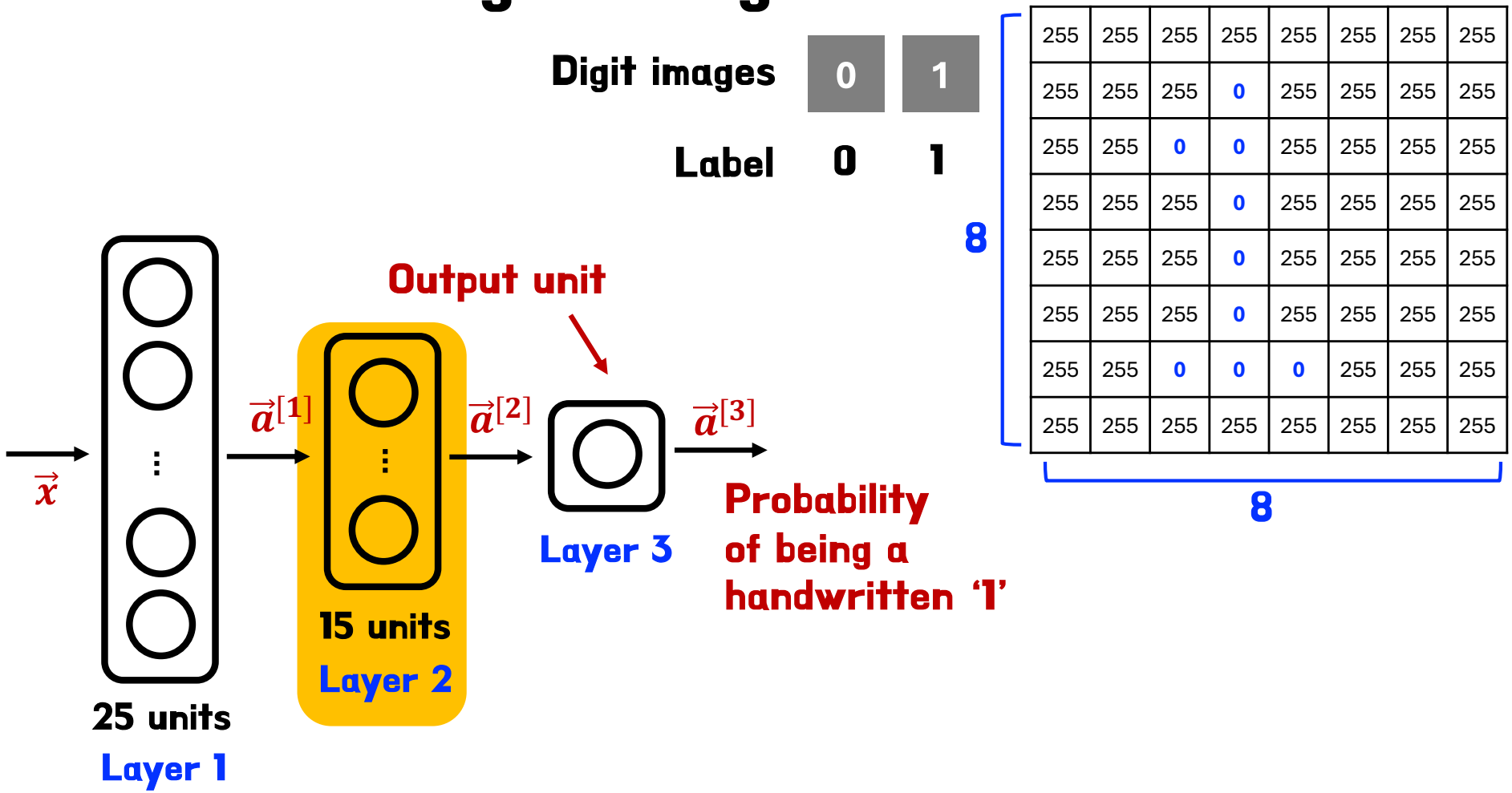


Hidden units (neurons)

$$\vec{a}^{[0]}$$

$$\vec{a}^{[1]} = \begin{bmatrix} g(\vec{w}_1^{[1]} \cdot \vec{x} + b_1^{[1]}) \\ \cdots \\ g(\vec{w}_{25}^{[1]} \cdot \vec{x} + b_{25}^{[1]}) \end{bmatrix}$$

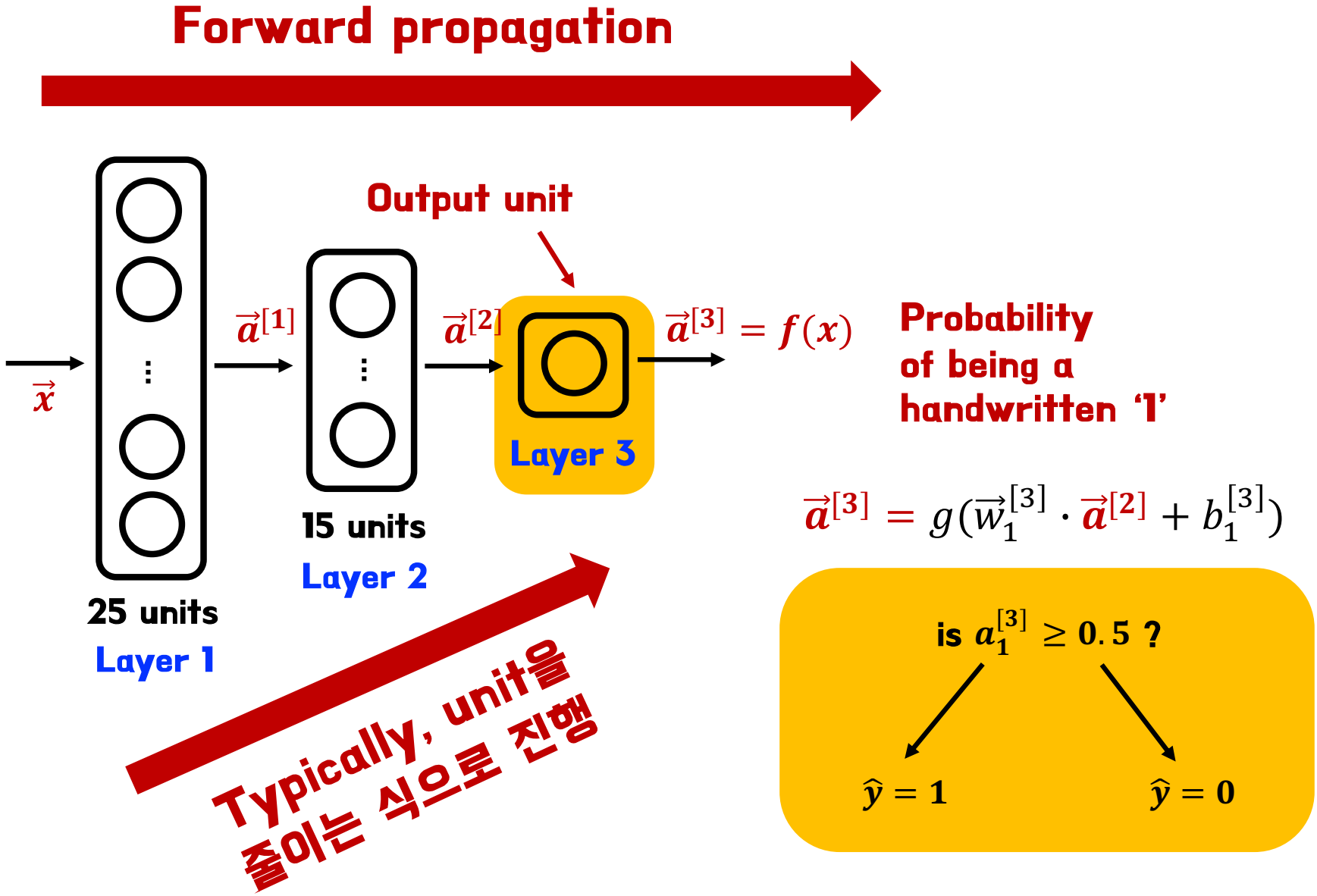
Handwritten digit recognition



Hidden units (neurons)

$$\vec{a}^{[2]} = \begin{bmatrix} g(\vec{w}_1^{[2]} \cdot \vec{a}^{[1]} + b_1^{[2]}) \\ \dots \\ g(\vec{w}_{15}^{[2]} \cdot \vec{a}^{[1]} + b_{15}^{[2]}) \end{bmatrix}$$

Handwritten digit recognition



Speculation on artificial general intelligence (AGI)

AI

ANI

(artificial narrow intelligence)

e.g.) smart speaker, self-driving car, web search, AI in farming and factories

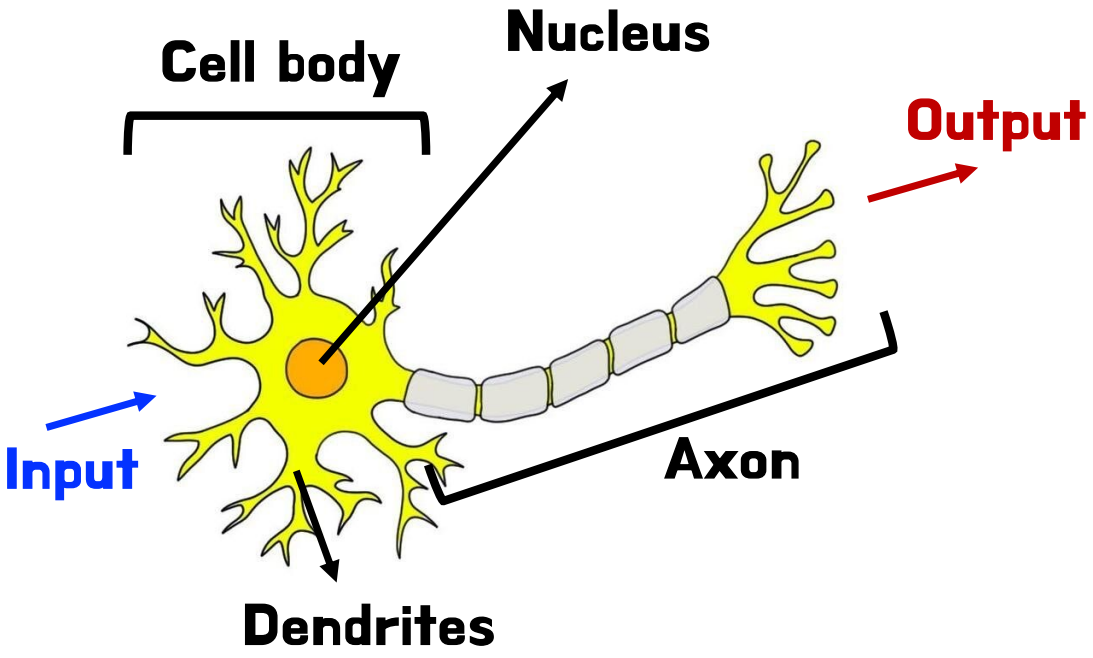
AGI

(artificial narrow intelligence)

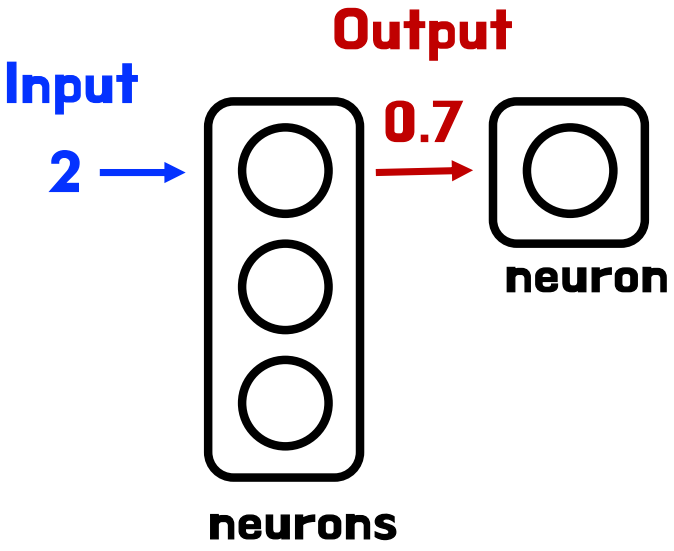
Do anything a human can do

Neural network and the brain

Biological neuron



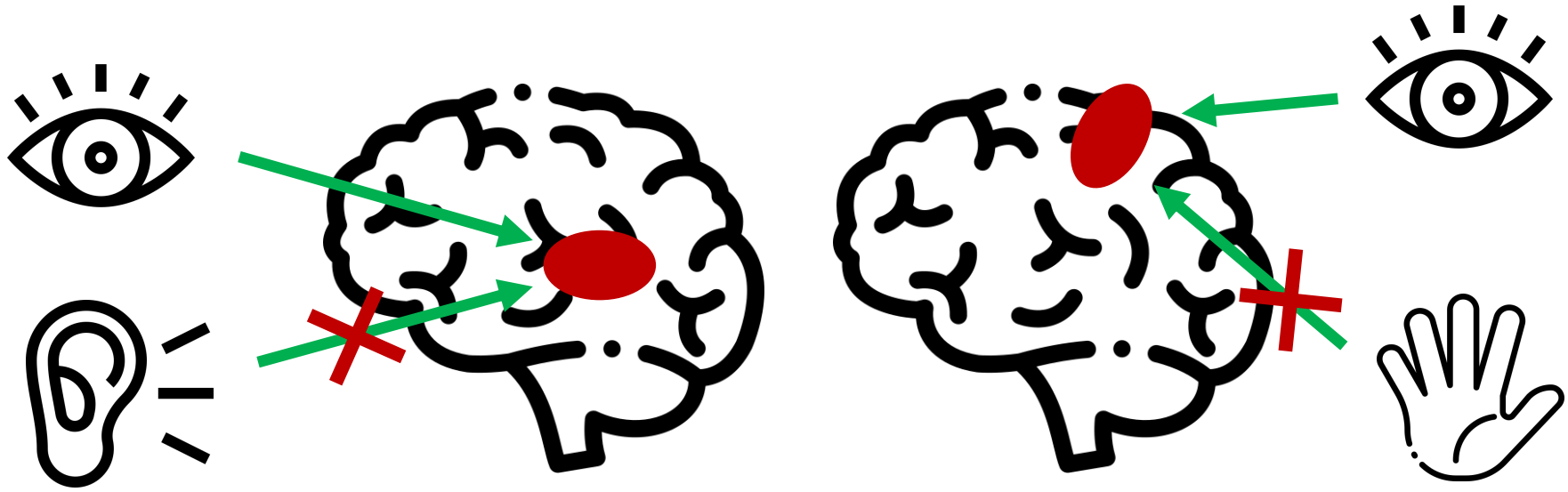
Simplified mathematical model of a neuron



Can we mimic the human brain ?

1. Machine learning is too simple
2. We don't know how the brain workrs

The “one learning algorithm” hypothesis

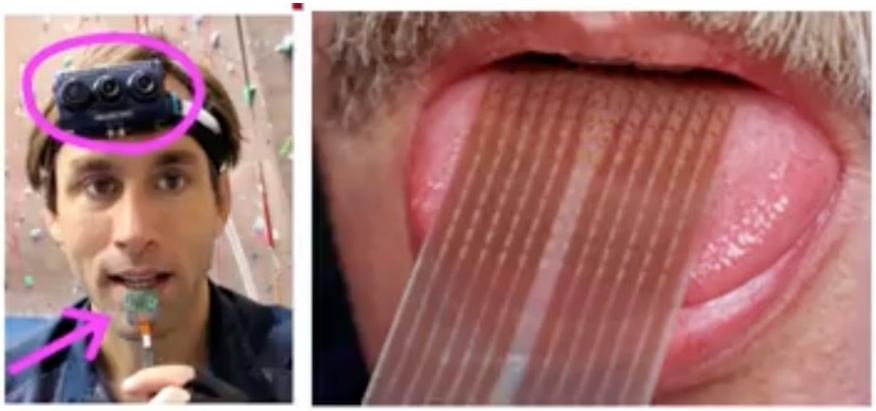


Auditory cortex Somatosensory cortex

One brain cortex can learn other sense as well

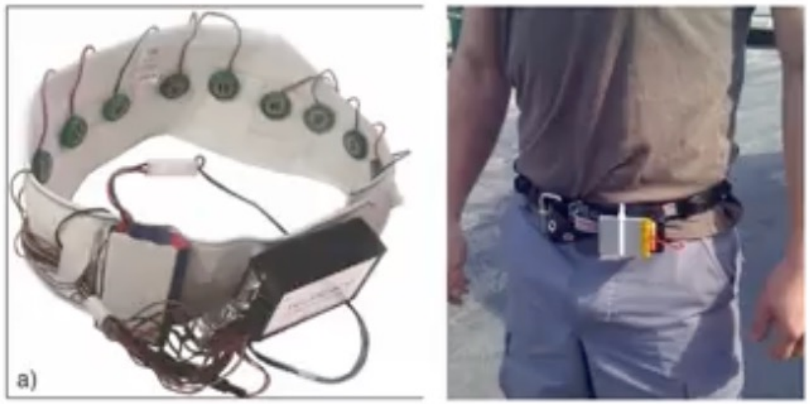
Sensor representations in the brain

We don't know if it is possible in machine learning



Seeing with your tongue

Human echolocation (sonar)



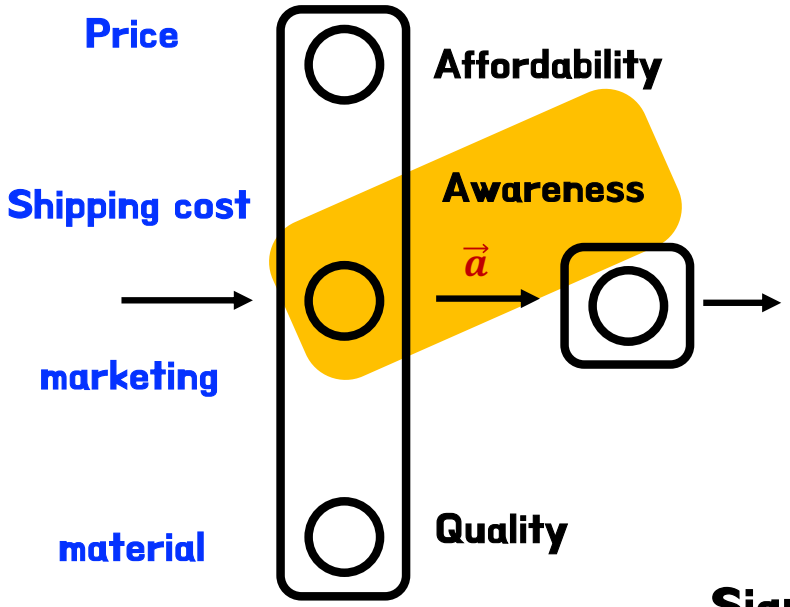
Haptic belt: Direction sense

Implanting a 3rd eye

TheMAD

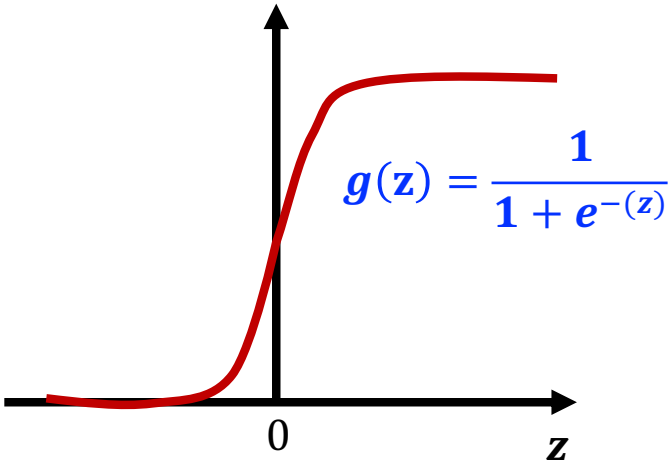
Activation function

Demand prediction example



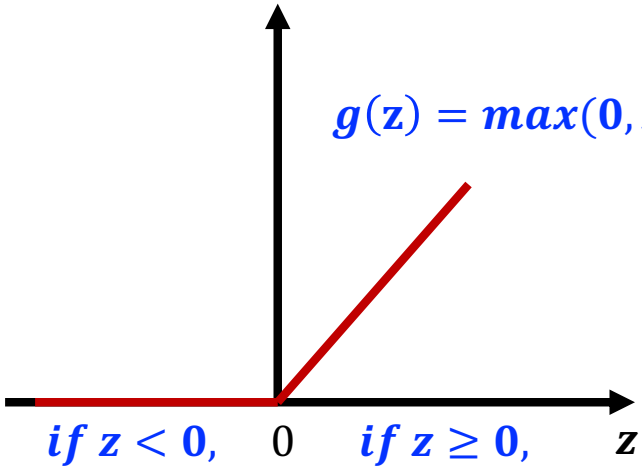
$$a_2^{[1]} = g(\vec{w}_2^{[1]} \cdot \vec{x} + b_2^{[1]})$$

Sigmoid



$0 < g(z) < 1$

ReLU
(Rectified Linear Unit)



$if\ z < 0,\ g(z) = 0$

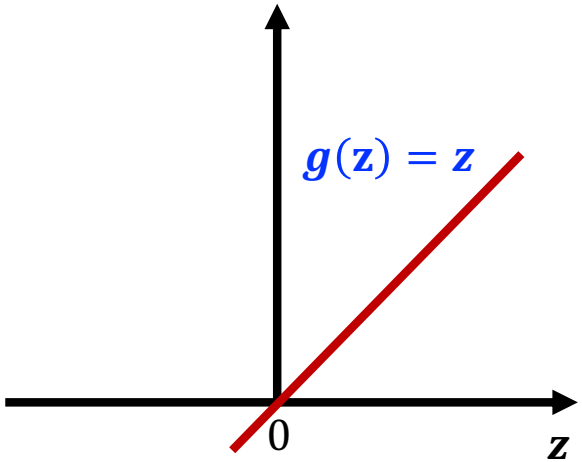
$if\ z \geq 0,\ g(z) = z$

Examples of activation functions

$$a_2^{[1]} = g(\vec{w}_2^{[1]} \cdot \vec{x} + b_2^{[1]})$$

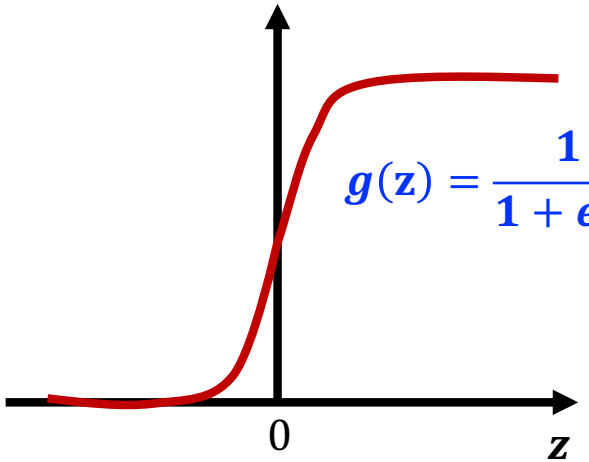
"No activation function"

Linear activation function



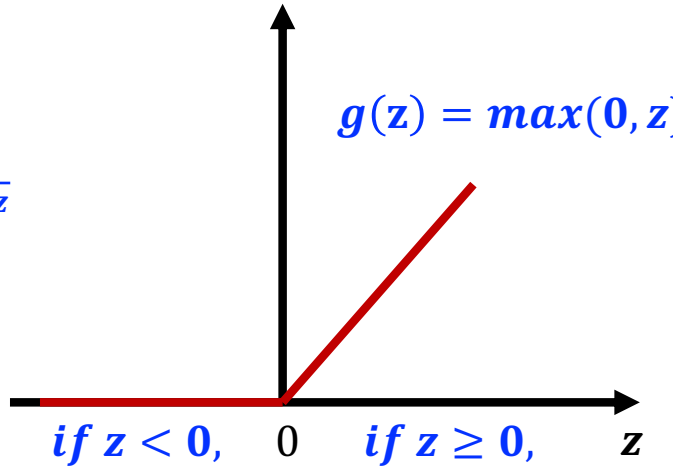
$a = g(z) = \vec{w} \cdot \vec{x} + b$

Sigmoid



$0 < g(z) < 1$

ReLU (Rectified Linear Unit)

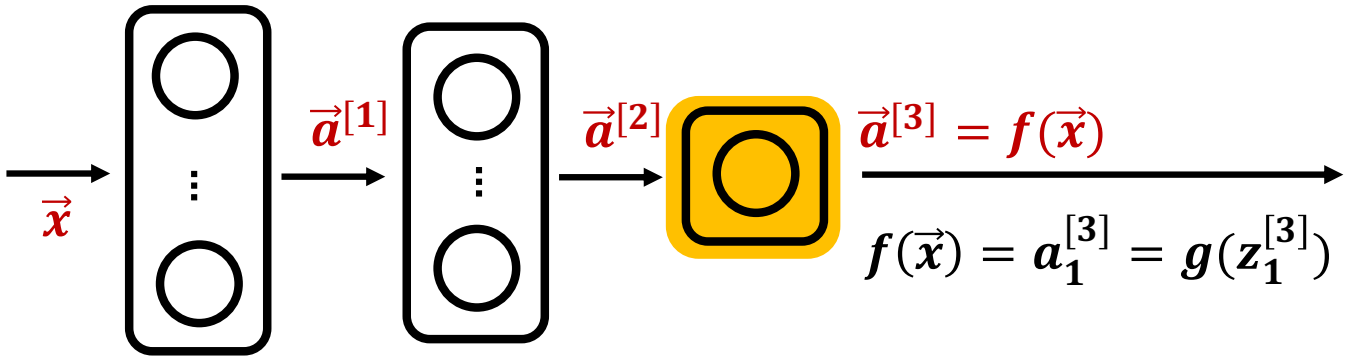


$if\ z < 0,\ g(z) = 0$

$if\ z \geq 0,\ g(z) = z$

TheMAC

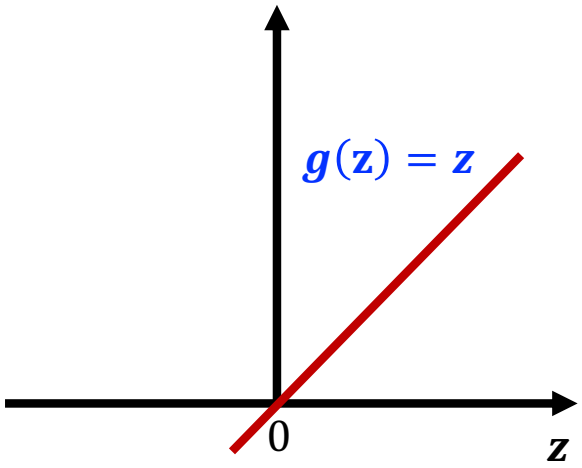
Output layer



Choosing $g(z)$ for output layer ?

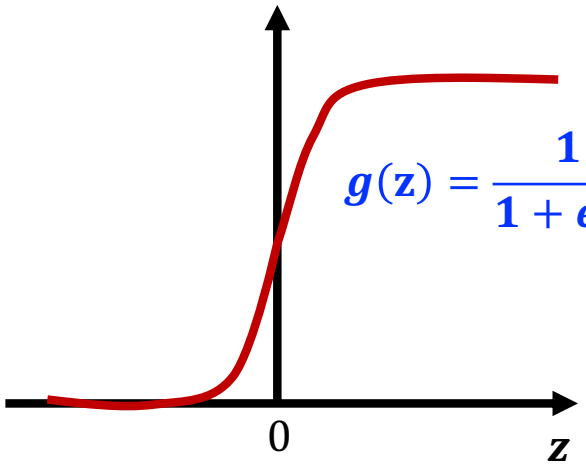
Regression

Linear activation function
y = +/-



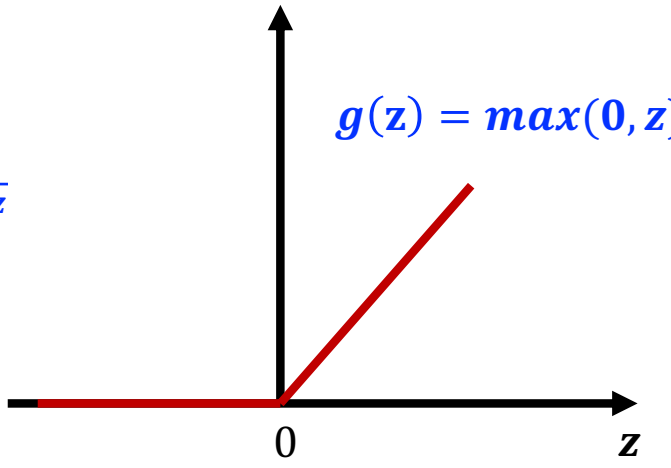
Binary classification

Sigmoid
y = 0/1

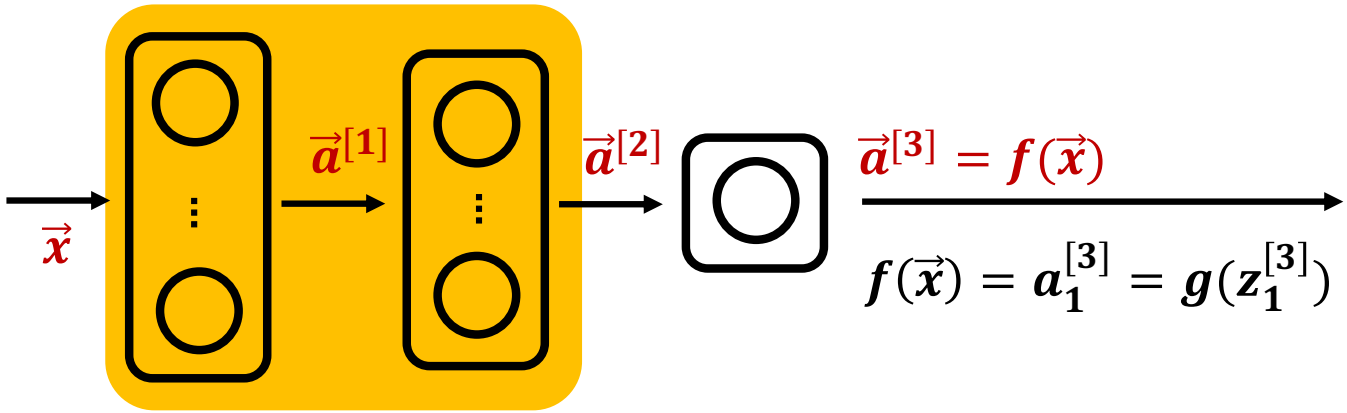


Regression

ReLU
y = 0 or +

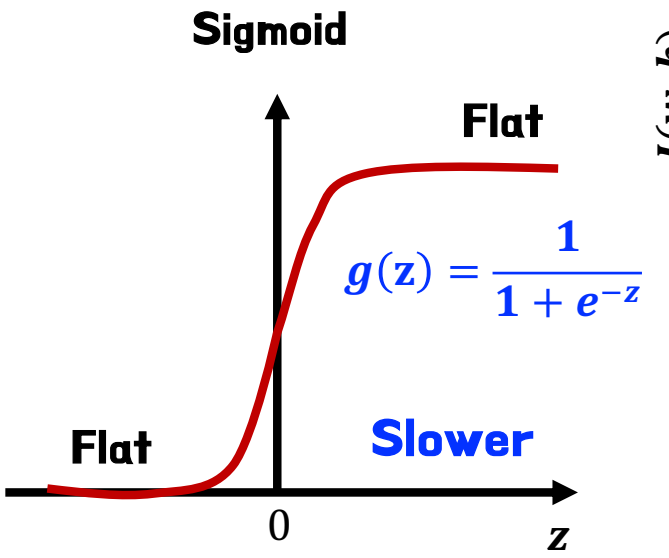


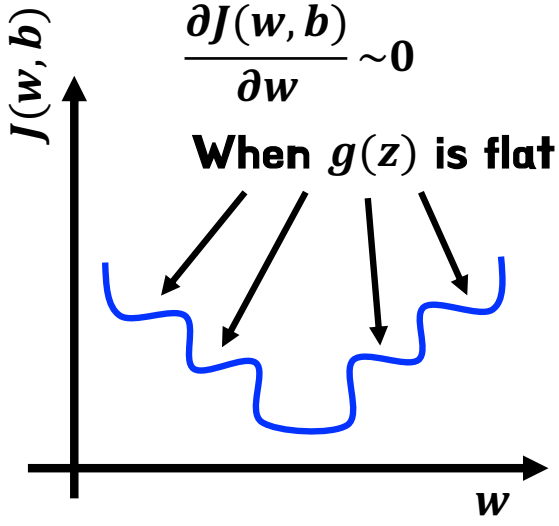
Hidden layer



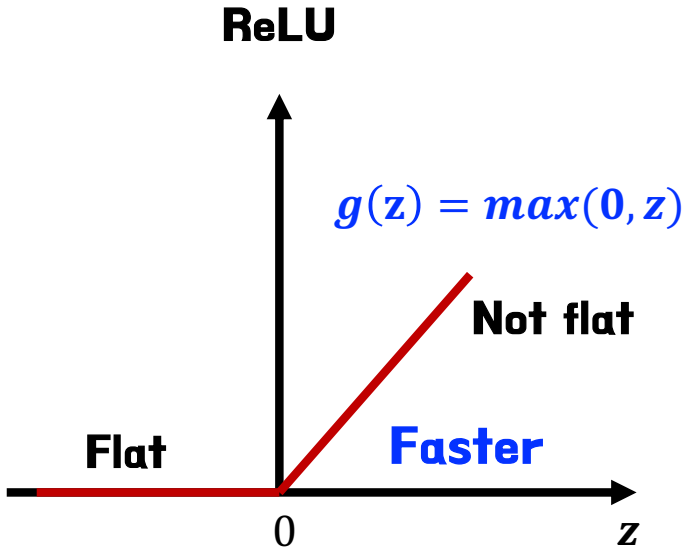
Choosing $g(z)$ for hidden layer ?

Classical

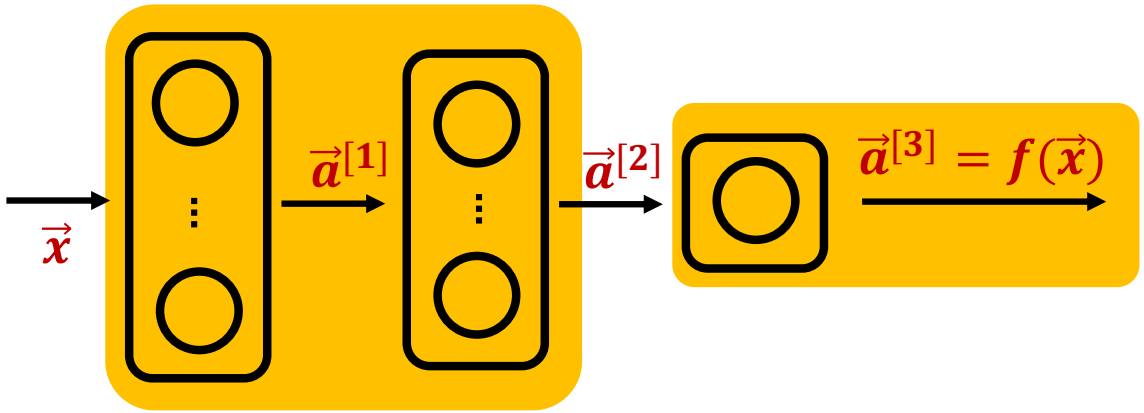




Most common



Choosing activation summary



ReLU for hidden layers

Binary classification
Activation = 'sigmoid'

Regression (y: -/+)
Activation = 'linear'

Regression (y: 0/+)
Activation = 'relu'

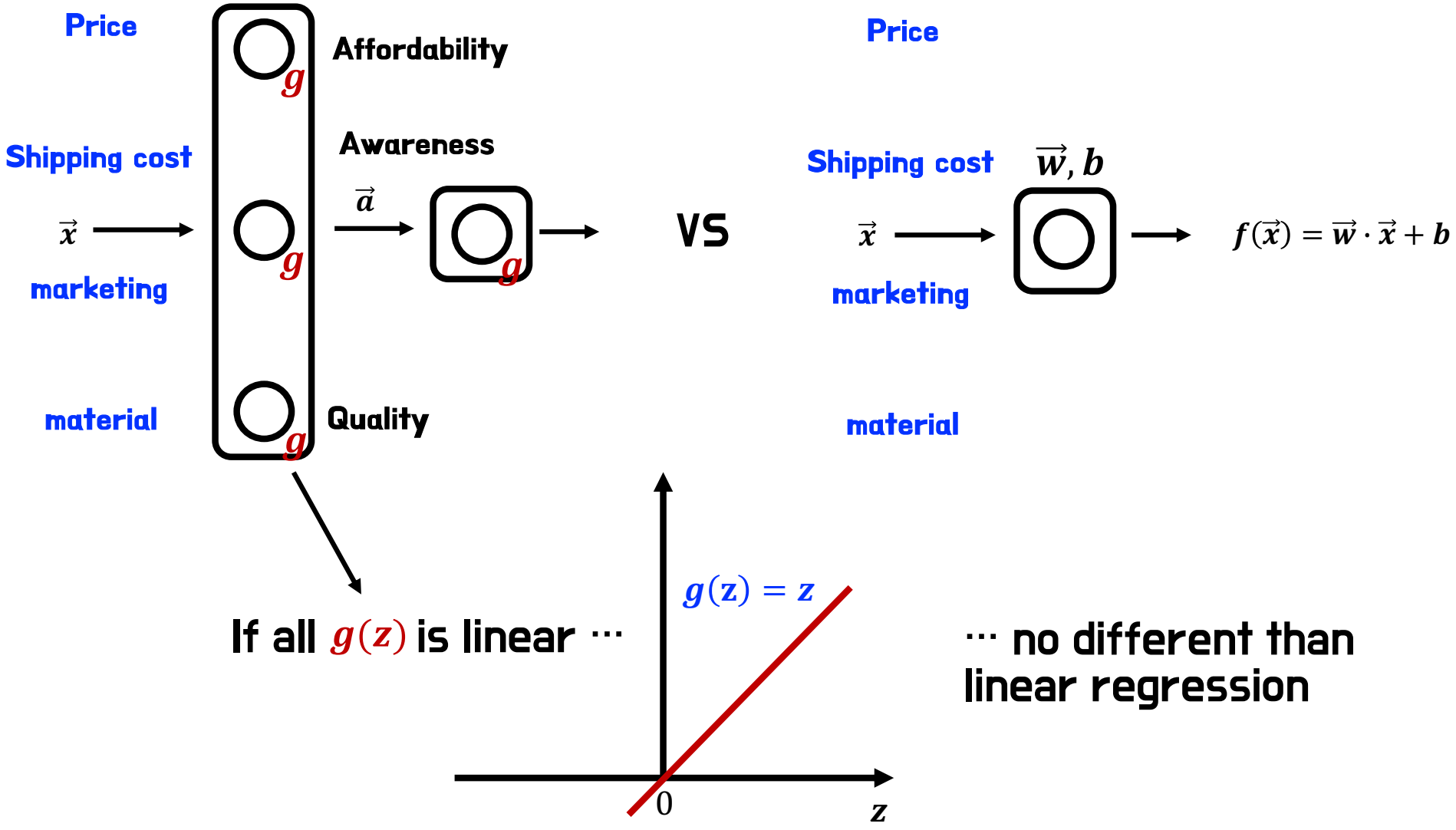
Typical coding

```
from tf.keras.layers import Dense
model = Sequential([
   Dense(units=25, activation='relu'),
   Dense(units=15, activation='relu'),
   Dense(units=1, activation='sigmoid'),
])
```

layer1
layer2
layer3 = output layer

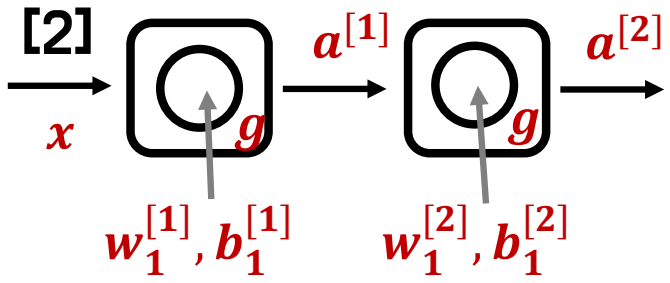
There are another activation functions: 'Reacky ReLU', 'Tanh', 'Softmax'

Why do we need activation functions ?



We can't solve complicated cases

Linear example



$$g(z) = z$$

$$a^{[1]} = w_1^{[1]} x + b_1^{[1]}$$

$$a^{[2]} = w_1^{[2]} a^{[1]} + b_1^{[2]}$$

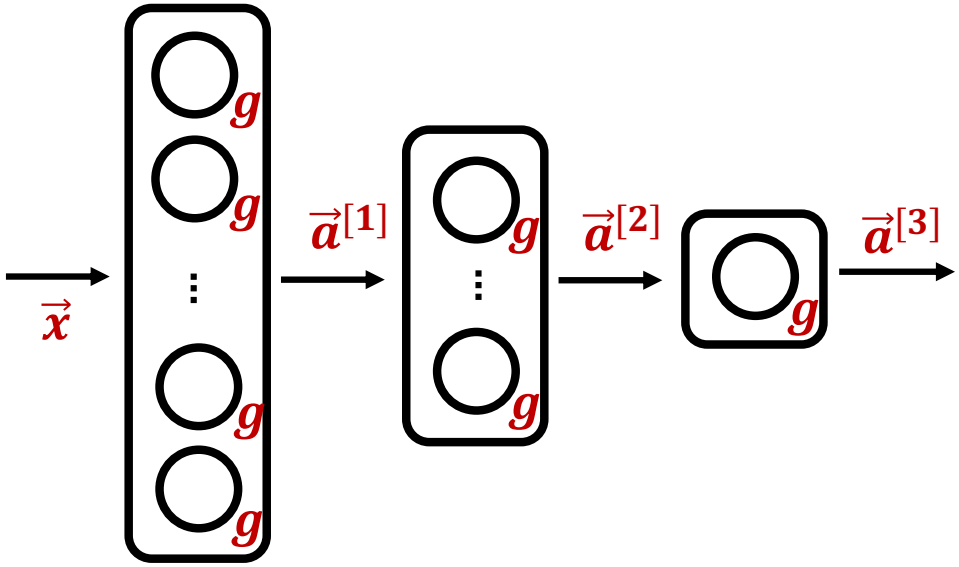
$$= w_1^{[2]}(w_1^{[1]} x + b_1^{[1]}) + b_1^{[2]}$$

$$= \underbrace{w_1^{[2]} w_1^{[1]}}_{w} x + \underbrace{(w_1^{[2]} b_1^{[1]} + b_1^{[2]})}_{b}$$

$$a^{[2]} = wx + b$$

Again, same linear regression

Example



$$g(z) = z$$

$$\vec{a}^{[3]} = \vec{w}_1^{[3]} \cdot \vec{a}^{[2]} + b_1^{[3]}$$

All linear (including output)
→ Equivalent to linear regression

$$g(z) = \frac{1}{1 + e^{-z}}$$

$$\vec{a}^{[3]} = \frac{1}{1 + e^{-(\vec{w}_1^{[3]} \cdot \vec{a}^{[2]} + b_1^{[3]})}}$$

Output : sigmoid (hidden : all linear)
→ Equivalent to logistic regression

Don't use linear activation in hidden layers (use ReLU)