Convolutional Neural Network

Black Box Model

 $\begin{array}{c}
x_1 \\
x_2 \\
x_3 \\
\vdots \\
x_N \\
\end{array}$ Black Box $\longrightarrow y$

$$y = f(x_1, x_2, ..., x_N)$$

261458 & 261753 Computer Vision

#14

261458 & 261753 Computer Vision

#14

Black Box Model

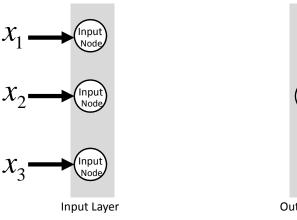
Regression/Prediction

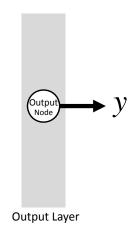
Yesterday	Today	Tomorrow
8.50	8.45	8.40
8.80	8.80	8.80
9.40	9.30	9.20
9.95	9.85	9.60
10.30	10.40	10.50
11.30	10.80	10.70

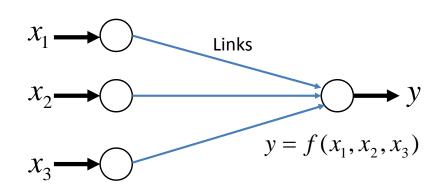
Classification

Feature 1	Feature 2	Class
0.2	0.5	0
-0.1	0.7	0
0.1	0.6	0
0.5	0.8	1
-0.6	-0.1	1
-0.7	-0.9	1

Artificial Neural Network ⁴

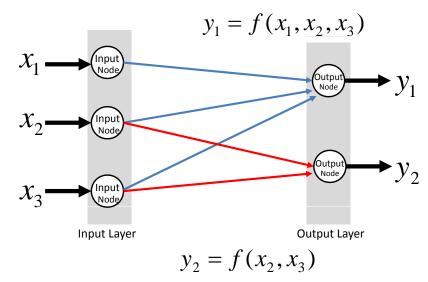






261458 & 261753 Computer Vision

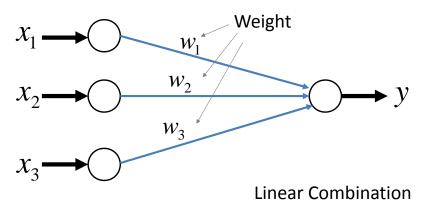
Artificial Neural Network ⁶



261458 & 261753 Computer Vision

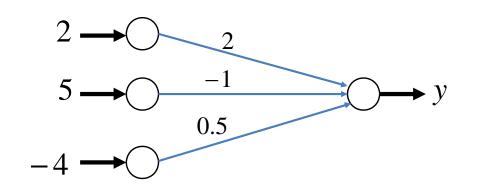
#1/

Artificial Neural Network 7



 $y = w_1 x_1 + w_2 x_2 + w_3 x_3$

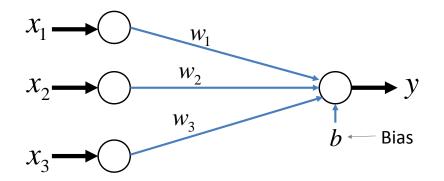
Artificial Neural Network 8



$$y = 2 \times 2 + (-1) \times 5 + 0.5 \times (-4) = -3$$

#14

Artificial Neural Network ⁹



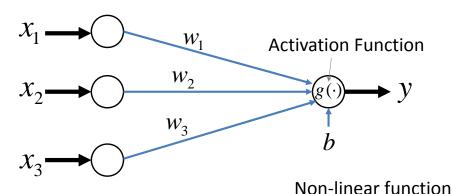
$$y = w_1 x_1 + w_2 x_2 + w_3 x_3 + b$$

261458 & 261753 Computer Vision

261458 & 261753 Computer Vision

#14

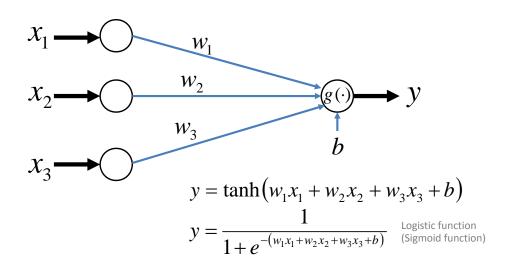
Artificial Neural Network



 $y = g(w_1x_1 + w_2x_2 + w_3x_3 + b)$

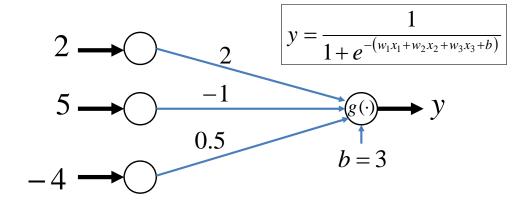
261458 & 261753 Computer Vision

Artificial Neural Network



Artificial Neural Network

Non-linear function $y = \max(x_1, x_2, x_3)$ $y = \min(x_1, x_2, x_3)$



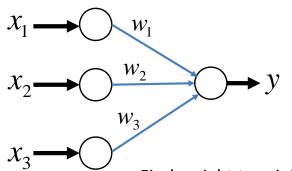
$$y = \frac{1}{1 + e^{-(2 \times 2 + (-1) \times 5 + 0.5 \times (-4) + 3)}} = \frac{1}{1 + e^0} = 0.5$$

261458 & 261753 Computer Vision

#14

15

Training Process



Training Set

\mathcal{X}_1	x_2	x_3	у
0	0	1	2
1	1	1	4
-1	0	2	1

Find weight to minimize prediction error

$$\sum_{\substack{\text{Output}\\ \text{output}}}^{\text{Actual}} (y_i - \hat{y}_i)^2$$

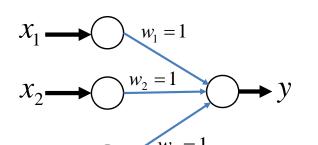
$$TrainingSet$$

261458 & 261753 Computer Vision

#14

16

Training Process



Training Set

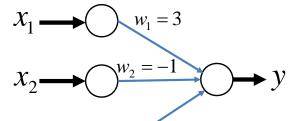
			_	
x_1	x_2	x_3	у	ŷ
0	0	1	2	1
1	1	1	4	3
-1	0	2	1	1

Prediction error

$$\sum_{\text{TrainingSet}} (y_i - \hat{y}_i)^2 = (2 - 1)^2 + (4 - 3)^2 + (1 - 1)^2$$

$$= 2$$

Training Process





Training Set

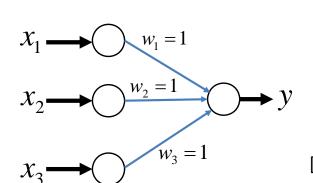
			0 - 1	
x_1	x_2	x_3	у	ŷ
0	0	1	2	
1	1	1	4	
-1	0	2	1	

Prediction error

$$\sum_{\text{Contains Sat}} (y_i - \hat{y}_i)^2 = ?$$

Back Propagation

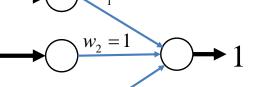
18

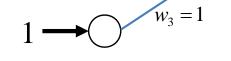


Training Set					
\mathcal{X}_1	x_2	x_3	у	ŷ	
0	0	1	2		
1	1	1	4		
-1	0	2	1		

[1] Initialize weight

0 –	$w_1 = 1$
_	- w -1





[2] Try to predict one sample with current weight

261458 & 261753 Computer Vision

#14

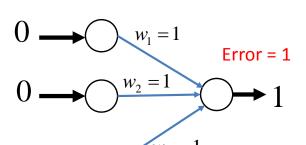
19

261458 & 261753 Computer Vision

#14

20

Back Propagation



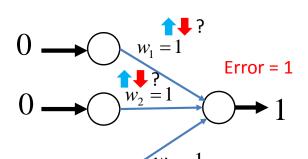
Hallillig Set				
x_1	x_2	x_3	у	ŷ
0	0	1	2	1
1	1	1	4	
-1	0	2	1	

Training Sat

 $W_3 = 1$ [3] Calculate the error

$$E = (y - \hat{y})^2 = (2 - 1)^2 = 1$$

Back Propagation



Training Set					
\mathcal{X}_1	x_2	x_3	У	ŷ	
0	0	1	2	1	
1	1	1	4		
-1	0	2	1		

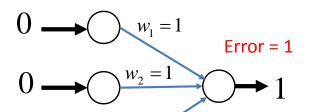
[4] Update the weights based on Error (Derivatives)

Training Set

 $x_1 \quad x_2 \quad x_3 \quad y$

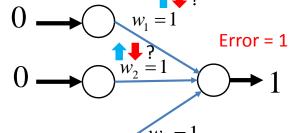
Back Propagation

Tra	inin	g Se	et
v	v	32	-





[4] Update the weights based on Error (Derivatives)



[4] Update the weights based on Error (Derivatives)

$$E = (y - \hat{y})^2 = (y - w_1 x_1 - w_2 x_2 - w_3 x_3)^2$$

$$= (2 - w_3)^2$$

$$\frac{\partial E}{\partial w_3} = -2(2 - w_3) = -2$$

261458 & 261753 Computer Vision

#14

23

261458 & 261753 Computer Vision

24

Back Propagation

Training Set

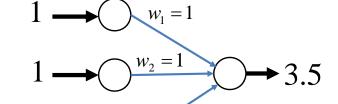
<i>x</i> ₁ –	$w_1 = 1$
<i>x</i> ₂ -	$w_2 = 1$
	1.7

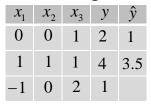
			0 - 1	
x_1	x_2	x_3	у	ŷ
0	0	1	2	1
1	1	1	4	
-1	0	2	1	

[4] Update the weights based on Error (Derivatives)

Back Propagation

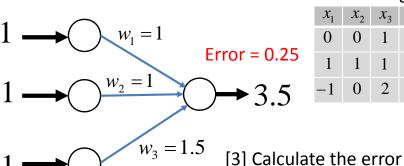
Training Set





 $w_3 = 1.5$ [2] Try to predict one sample with current weight

Back Propagation



Training Set					
x_1	x_2	x_3	У	ŷ	
0	0	1	2	1	
1	1	1	4	3.5	
-1	0	2	1		

Training Set

$$E = (y - \hat{y})^2 = (4 - 3.5)^2 = 0.25$$

Error = 0.25

Training Set						
\mathcal{X}_1	$x_1 \mid x_2 \mid x_3 \mid y \mid \hat{y}$					
0	0	1	2	1		
1	1	1	4	3.5		
-1	0	2	1			

[4] Update the weights based on Error

$$E = (y - \hat{y})^2 = (y - w_1 x_1 - w_2 x_2 - w_3 x_3)^2$$
$$= (4 - w_1 - w_2 - w_3)^2$$

261458 & 261753 Computer Vision

#14

27

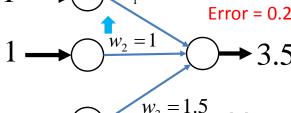
261458 & 261753 Computer Vision

#14

28

Back Propagation

Error = 0.25



[4] Update the weights based on Error

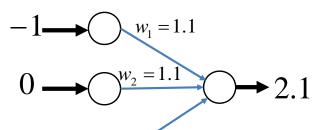
Back Propagation

 $w_1 = 1.1$



Training Set

[4] Update the weights based on Error

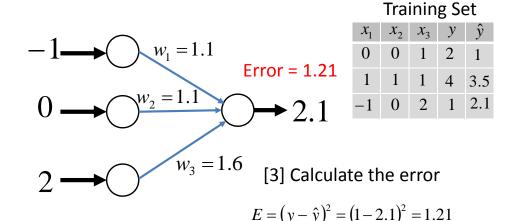


Training Set					
\mathcal{X}_1	x_2	x_3	у	ŷ	
0	0	1	2	1	
1	1	1	4	3.5	
-1	0	2	1	2.1	

 $w_3 = 1.6$

[2] Try to predict one sample with current weight

Back Propagation



261458 & 261753 Computer Vision

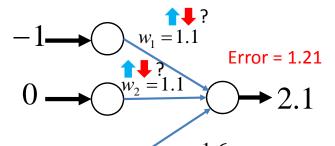
#14

31

#14

261458 & 261753 Computer Vision

Back Propagation

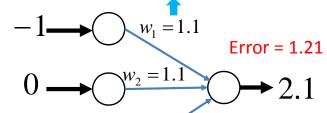


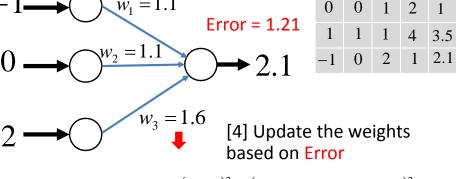
Iraining Set					
\mathcal{X}_1	x_2	x_3	у	ŷ	
0	0	1	2	1	
1	1	1	4	3.5	
-1	0	2	1	2.1	

[4] Update the weights based on Error

$$E = (y - \hat{y})^2 = (y - w_1 x_1 - w_2 x_2 - w_3 x_3)^2$$
$$= (1 + w_1 - 2w_3)^2$$

Back Propagation





$$E = (y - \hat{y})^2 = (y - w_1 x_1 - w_2 x_2 - w_3 x_3)^2$$
$$= (1 + w_1 - 2w_3)^2$$

Training Set

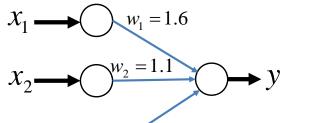
32

261458 & 261753 Computer Vision

2 1.4

4 3.5

Training Set



manning oct						
x_1	x_2	x_3	у	ŷ		
0	0	1	2	1		
1	1	1	4	3.5		
-1	0	2	1	2.1		

 $w_3 = 1.4$

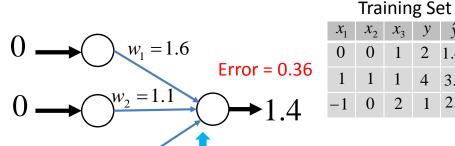
[4] Update the weights based on Error

261458 & 261753 Computer Vision

#14

35

Back Propagation



[2] Try to predict one sample with current weight

[3] Calculate the error

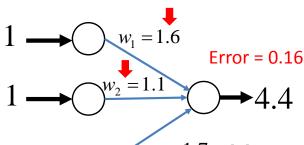
[4] Update the weights based on Error

261458 & 261753 Computer Vision

#14

36

Back Propagation



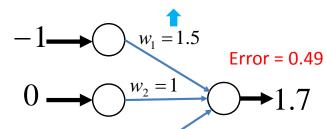
Training Set					
\mathcal{X}_1	x_2	x_3	у	ŷ	
0	0	1	2	1.4	
1	1	1	4	4.4	
-1	0	2	1	2.1	

[2] Try to predict one sample with current weight

[3] Calculate the error

[4] Update the weights based on Error

Back Propagation



Hallillig Set					
x_1	x_2	x_3	у	ŷ	
0	0	1	2	1.4	
1	1	1	4	4.4	
-1	0	2	1	1.7	

Training Sat

[2] Try to predict one sample with current weight

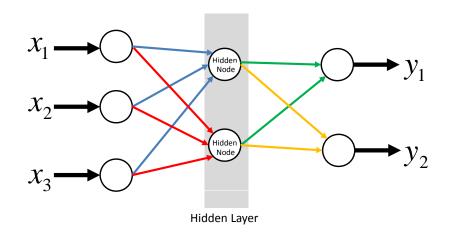
[3] Calculate the error

[4] Update the weights based on Error

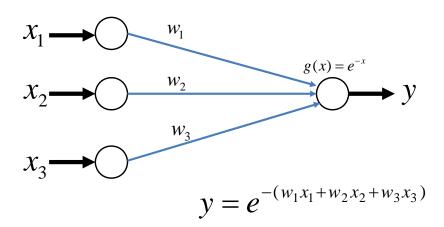
#14

261458 & 261753 Computer Vision

261458 & 261753 Computer Vision



Artificial Neural Network

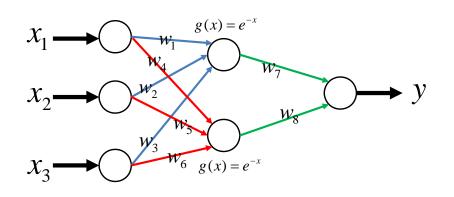


261458 & 261753 Computer Vision

#14

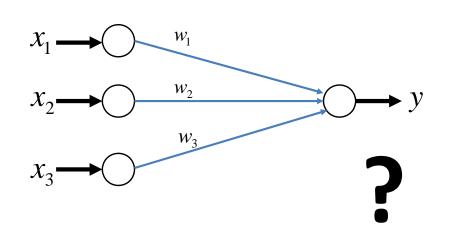
261458 & 261753 Computer Vision

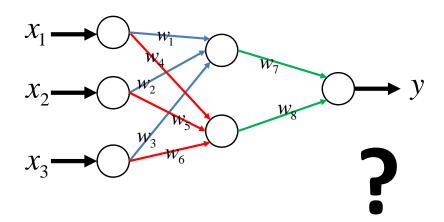
Artificial Neural Network



$$y = w_7 e^{-(w_1 x_1 + w_2 x_2 + w_3 x_3)} + w_8 e^{-(w_4 x_1 + w_5 x_2 + w_6 x_3)}$$

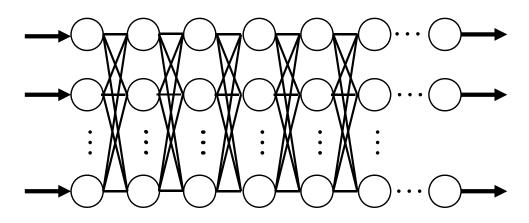
Artificial Neural Network





Deep learning

เชรียแม่ง Deep



261458 & 261753 Computer Vision

#14

43

#14

261458 & 261753 Computer Vision

#14

Image Filter

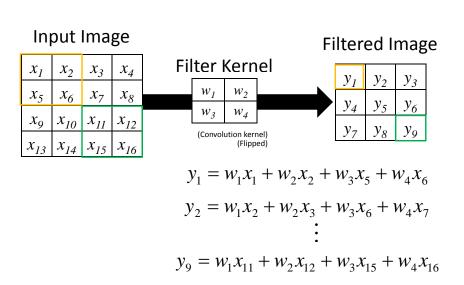
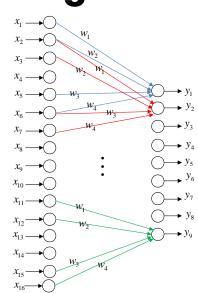


Image Filter

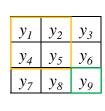


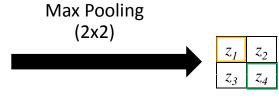
$$y_1 = w_1 x_1 + w_2 x_2 + w_3 x_5 + w_4 x_6$$
$$y_2 = w_1 x_2 + w_2 x_3 + w_3 x_6 + w_4 x_7$$

 $y_9 = w_1 x_{11} + w_2 x_{12} + w_3 x_{15} + w_4 x_{16}$

261458 & 261753 Computer Vision

Pooling (Downsampling)





$$z_1 = \max(y_1, y_2, y_4, y_5)$$

$$z_2 = \max(y_3, y_6)$$

$$z_3 = \max(y_7, y_8)$$

$$z_4 = y_9$$

#14

47

1	1	2	4
5	6	7	8
3	2	1	0
1	2	3	4

max pool with 2x2 filters and stride 2

3

http://cs231n.github.io/convolutional-networks/

261458 & 261753 Computer Vision

261458 & 261753 Computer Vision

#14

Image Filter + Pooling

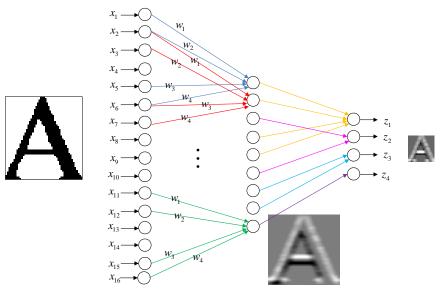


Image Filter + Pooling

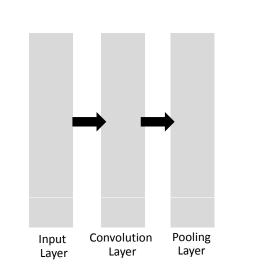
Pooling Filter 1 Pooling Filter 2 Filter N **Pooling**

261458 & 261753 Computer Vision

261458 & 261753 Computer Vision

Image Filter + Pooling

50



Convolution with various filter

224x224x64

pool

112x112x64

pool

downsampling

112

112

112

http://cs231n.github.io/convolutional-networks/

261458 & 261753 Computer Vision

#14

261458 & 261753 Computer Vision

#14

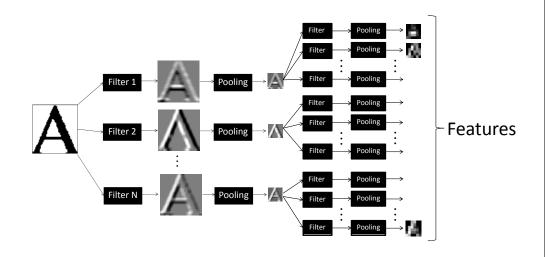
Image Filter + Pooling

51

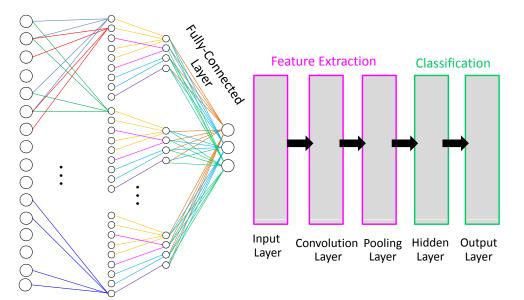
Input Convolution Pooling Convolution Pooling Convolution Layer Layer Layer Layer Layer

Image Filter + Pooling

52



Convolutional Neural Network (CNN)

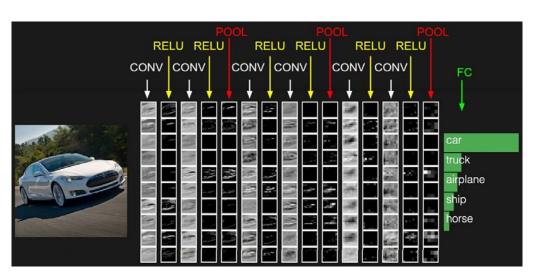


261458 & 261753 Computer Vision

#14

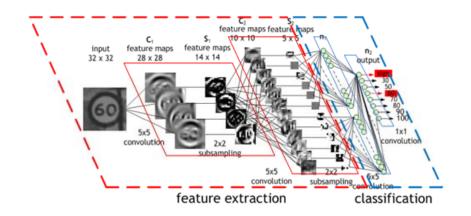
#14

Convolutional Neural Network (CNN)



http://cs231n.github.io/convolutional-networks/

Convolutional Neural Network (CNN)



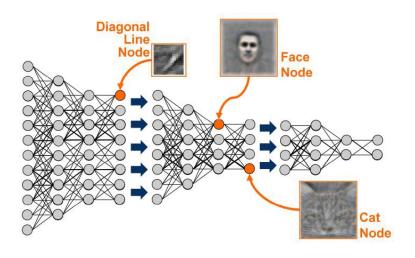
https://devblogs.nvidia.com/wp-content/uploads/2015/11/fig1.png

261458 & 261753 Computer Vision

#14

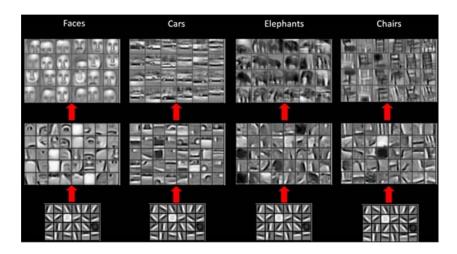
56

Convolutional Neural Network (CNN)



http://www.kdnuggets.com/2015/11/crazy-deep-learning-topological-data-analysis.html

Convolutional Neural Network (CNN)



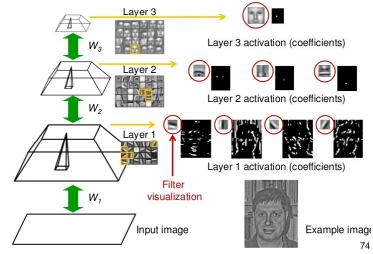
http://stats.stackexchange.com/questions/146413/why-convolutional-neural-networks-belong-to-deep-learning

261458 & 261753 Computer Vision

#14

Convolutional Neural Network (CNN)

Convolutional deep belief networks illustration



http://cs.stackex.change.com/questions/16545/what-is-the-difference-between-a-neural-network-a-deep-learning-system-and-a-deep-learning-system-a-de-geep-learning-system-a-de-geep-learning-system-a-de-geep-learning-system-a-de-geep-learning-system-a-de-geep-learning-system-a-de-geep-learning-system-a-de-geep-learning-system-a-de-geep-learning-system-a-de-geep-learning-system-a-de-geep-learning-system-a-de-geep-learning-system-a-de-geep-learning-system-a-de-geep-learning-system-a-de-geep-learning-system-a-de-geep-learning-system-a-de-geep-learning-system-a-de-geep-learning-system-a-de-gee

261458 & 261753 Computer Vision