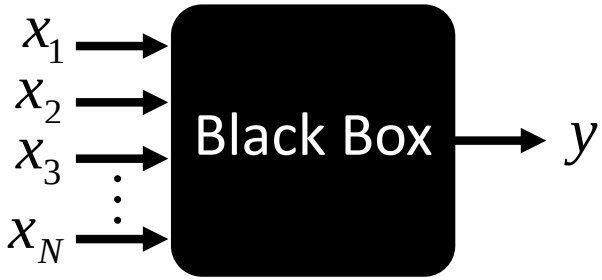
# Convolutional Neural Network



# Black Box Model






$$y = f(x_1, x_2, \ldots, x_N)$$



# 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 |

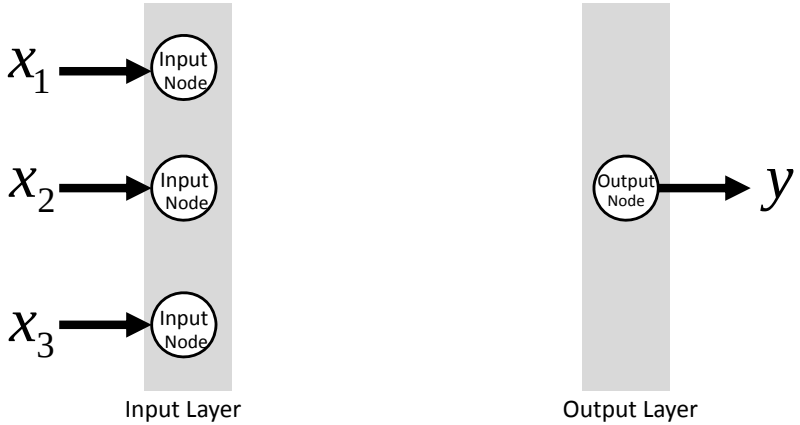
Classification

| 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 |



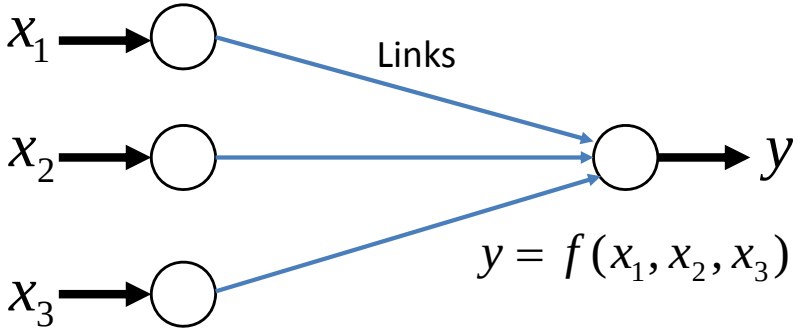
# Artificial Neural Network





# Artificial Neural Network

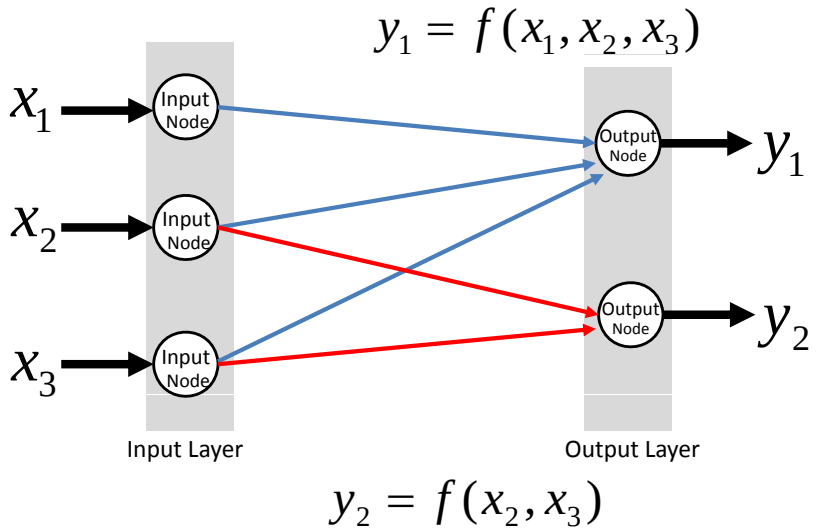







# Artificial Neural Network

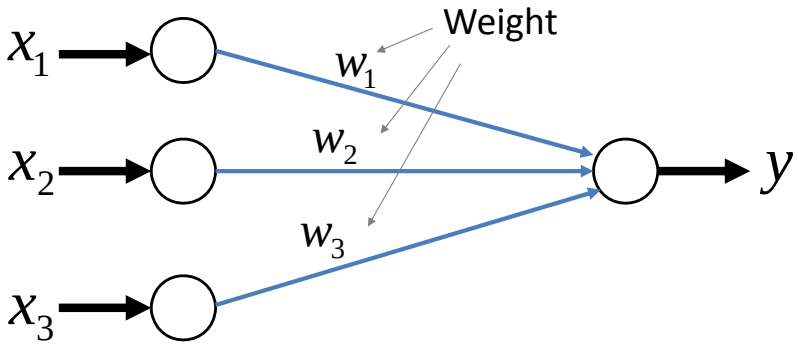







# Artificial Neural Network





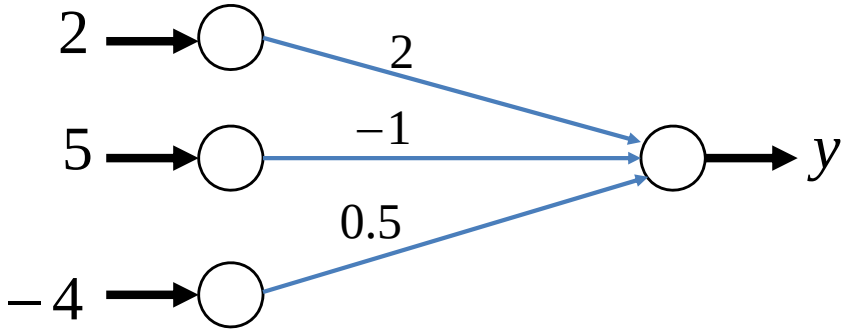

Linear Combination

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



# Artificial Neural Network

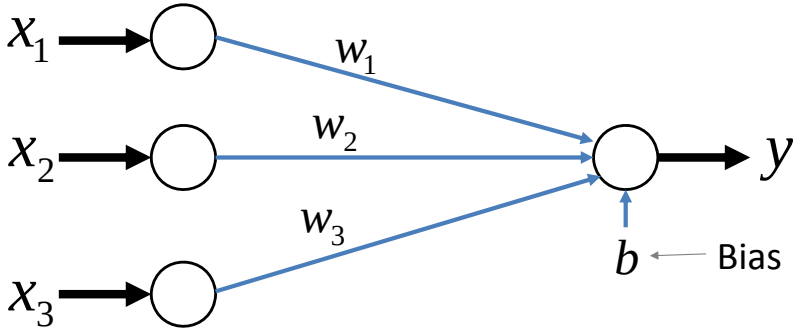





$$y = 2 \times 2 + (-1) \times 5 + 0.5 \times (-4) = -3$$

# Artificial Neural Network



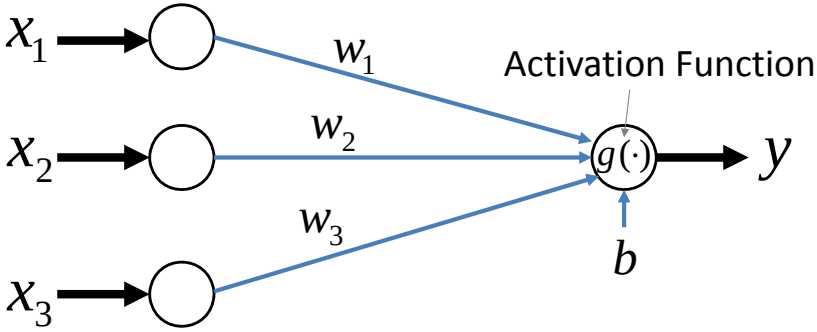



$$y = w_1x_1 + w_2x_2 + w_3x_3 + b$$



# Artificial Neural Network





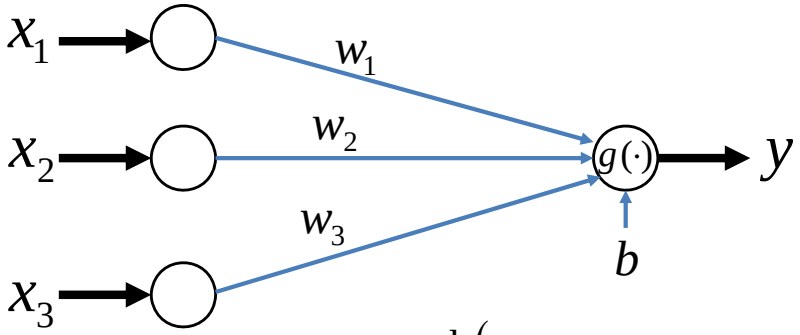

Non-linear function

$$y = g\left(w_1x_1 + w_2x_2 + w_3x_3 + b\right)$$



# Artificial Neural Network






$$y = \tanh\left(w_1x_1 + w_2x_2 + w_3x_3 + b\right)$$

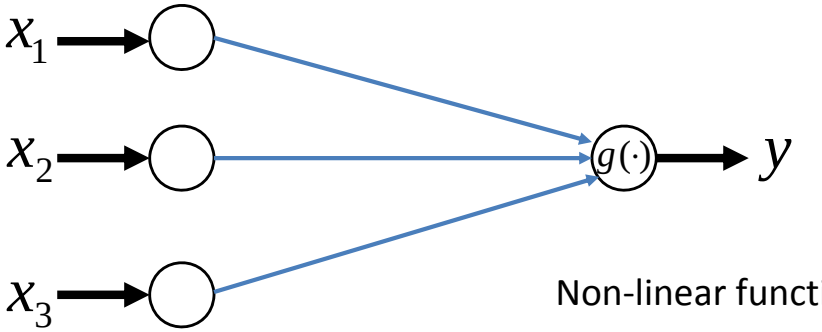
$$y = \frac{1}{1 + e^{-\left(w_1x_1 + w_2x_2 + w_3x_3 + b\right)}}$$

Logistic function (Sigmoid function)



# Artificial Neural Network





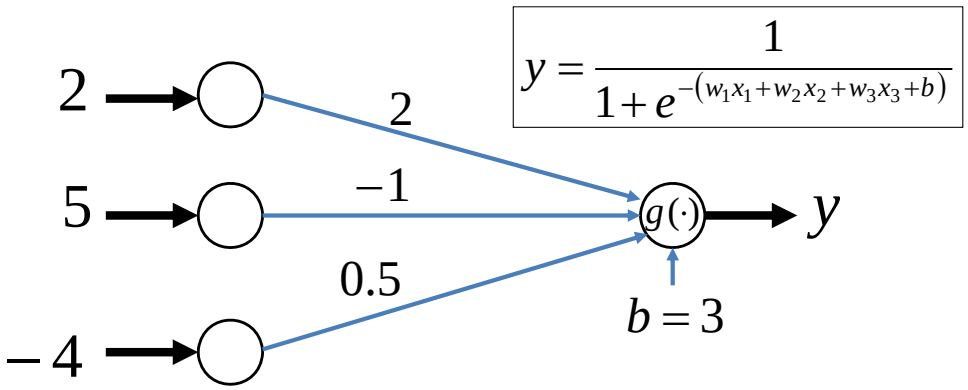

Non-linear function

$$y = \max(x_1, x_2, x_3)$$

$$y = \min(x_1, x_2, x_3)$$

# Artificial Neural Network





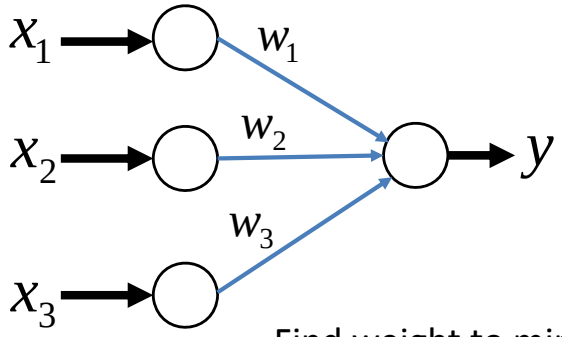

$$y = \frac{1}{1+e^{-(w_1x_1+w_2x_2+w_3x_3+b)}}$$

$$y = \frac{1}{1+e^{-(2\times2+(-1)\times5+0.5\times(-4)+3)}} = \frac{1}{1+e^{0}} = 0.5$$



# Training Process






Training Set

| $x_1$ | $x_2$ | $x_3$ | $y$ |
|---|---|---|---|
| 0 | 0 | 1 | 2 |
| 1 | 1 | 1 | 4 |
| −1 | 0 | 2 | 1 |

Find weight to minimize prediction error

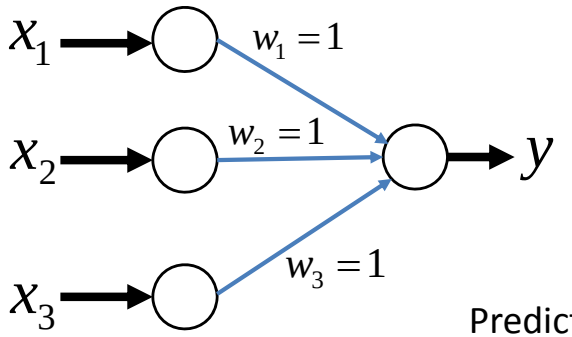
$$\sum_{TrainingSet} (y_i - \hat{y}_i)^2$$

Actual output: $y_i$; Predicted output: $\hat{y}_i$



# Training Process






Training Set

| $x_1$ | $x_2$ | $x_3$ | $y$ | $\hat{y}$ |
|---|---|---|---|---|
| 0 | 0 | 1 | 2 | 1 |
| 1 | 1 | 1 | 4 | 3 |
| −1 | 0 | 2 | 1 | 1 |

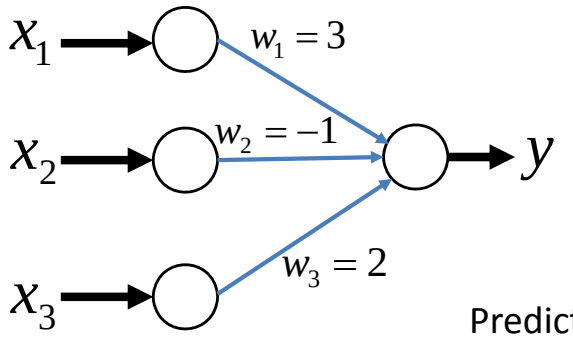
Prediction error

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



# Training Process






Training Set

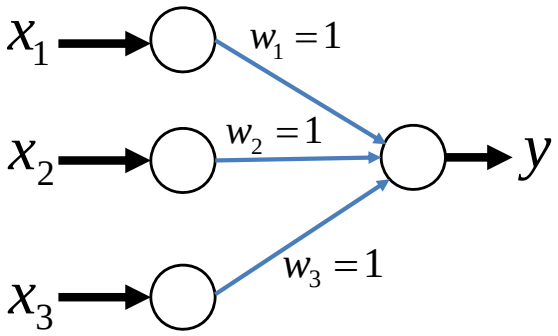
| $x_1$ | $x_2$ | $x_3$ | $y$ | $\hat{y}$ |
|---|---|---|---|---|
| 0 | 0 | 1 | 2 | |
| 1 | 1 | 1 | 4 | |
| −1 | 0 | 2 | 1 | |

Prediction error

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

# Back Propagation






Training Set

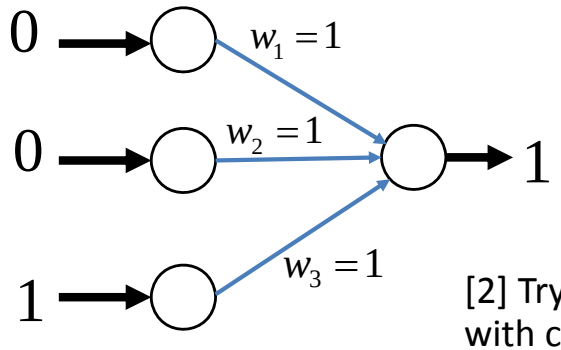
| $x_1$ | $x_2$ | $x_3$ | $y$ | $\hat{y}$ |
|---|---|---|---|---|
| 0 | 0 | 1 | 2 | |
| 1 | 1 | 1 | 4 | |
| −1 | 0 | 2 | 1 | |

[1] Initialize weight



# Back Propagation






Training Set

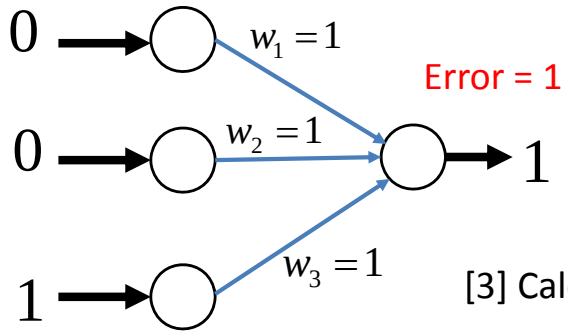
| $x_1$ | $x_2$ | $x_3$ | $y$ | $\hat{y}$ |
|---|---|---|---|---|
| 0 | 0 | 1 | 2 | 1 |
| 1 | 1 | 1 | 4 | |
| −1 | 0 | 2 | 1 | |

[2] Try to predict one sample with current weight



# Back Propagation






Training Set

| $x_1$ | $x_2$ | $x_3$ | $y$ | $\hat{y}$ |
|---|---|---|---|---|
| 0 | 0 | 1 | 2 | 1 |
| 1 | 1 | 1 | 4 | |
| −1 | 0 | 2 | 1 | |

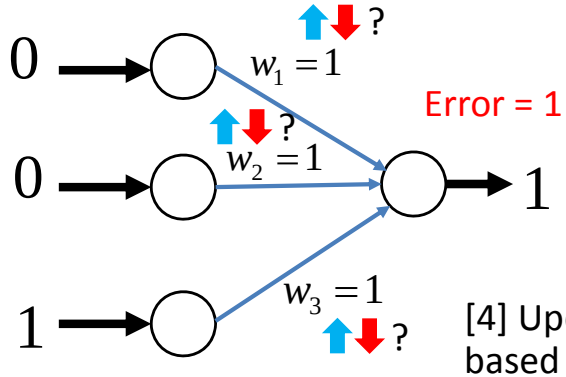
[3] Calculate the error

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



# Back Propagation





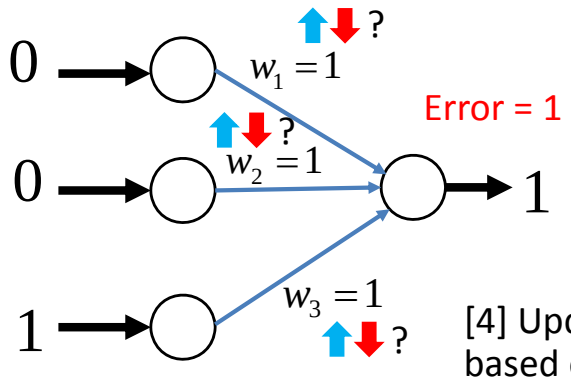

Training Set

| $x_1$ | $x_2$ | $x_3$ | $y$ | $\hat{y}$ |
|---|---|---|---|---|
| 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

| $x_1$ | $x_2$ | $x_3$ | $y$ | $\hat{y}$ |
|---|---|---|---|---|
| 0 | 0 | 1 | 2 | 1 |
| 1 | 1 | 1 | 4 | |
| −1 | 0 | 2 | 1 | |

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

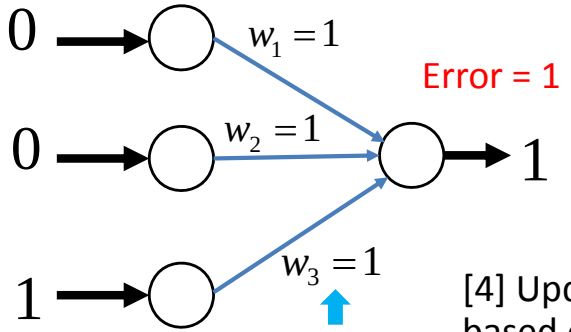
$$E = (y - \hat{y})^2 = (y - w_1x_1 - w_2x_2 - w_3x_3)^2 = (2 - w_3)^2$$

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



# Back Propagation






Training Set

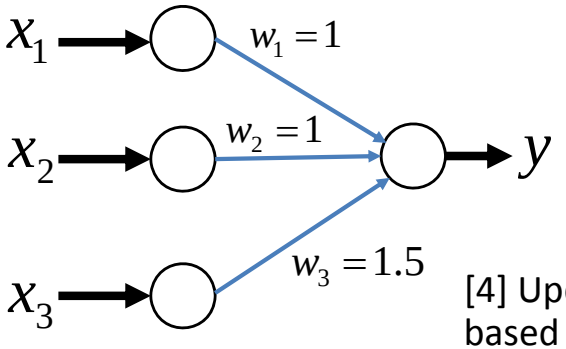
| $x_1$ | $x_2$ | $x_3$ | $y$ | $\hat{y}$ |
|---|---|---|---|---|
| 0 | 0 | 1 | 2 | 1 |
| 1 | 1 | 1 | 4 | |
| −1 | 0 | 2 | 1 | |

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



# Back Propagation






Training Set

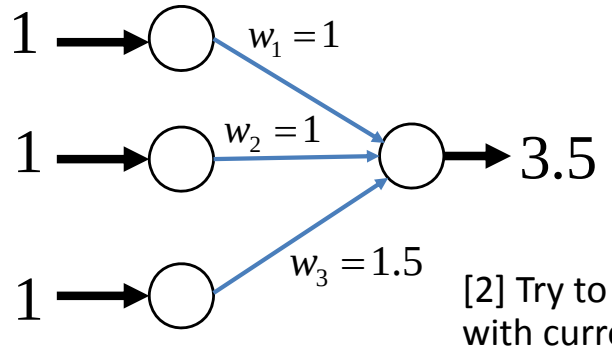
| $x_1$ | $x_2$ | $x_3$ | $y$ | $\hat{y}$ |
|---|---|---|---|---|
| 0 | 0 | 1 | 2 | 1 |
| 1 | 1 | 1 | 4 | |
| −1 | 0 | 2 | 1 | |

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



# Back Propagation






Training Set

| $x_1$ | $x_2$ | $x_3$ | $y$ | $\hat{y}$ |
|---|---|---|---|---|
| 0 | 0 | 1 | 2 | 1 |
| 1 | 1 | 1 | 4 | 3.5 |
| −1 | 0 | 2 | 1 | |

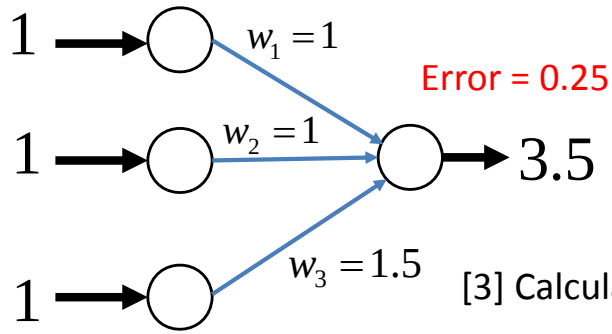
[2] Try to predict one sample with current weight

# Back Propagation



Training Set

| $x_1$ | $x_2$ | $x_3$ | $y$ | $\hat{y}$ |
|---|---|---|---|---|
| 0 | 0 | 1 | 2 | 1 |
| 1 | 1 | 1 | 4 | 3.5 |
| −1 | 0 | 2 | 1 | |




Error = 0.25

[3] Calculate the error

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

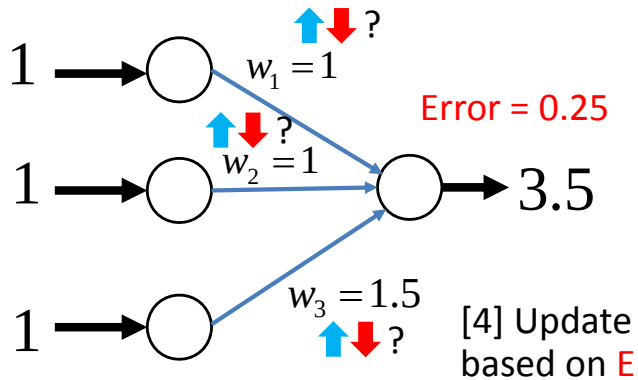


# Back Propagation



Training Set

| $x_1$ | $x_2$ | $x_3$ | $y$ | $\hat{y}$ |
|---|---|---|---|---|
| 0 | 0 | 1 | 2 | 1 |
| 1 | 1 | 1 | 4 | 3.5 |
| −1 | 0 | 2 | 1 | |




Error = 0.25

[4] Update the weights based on Error

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

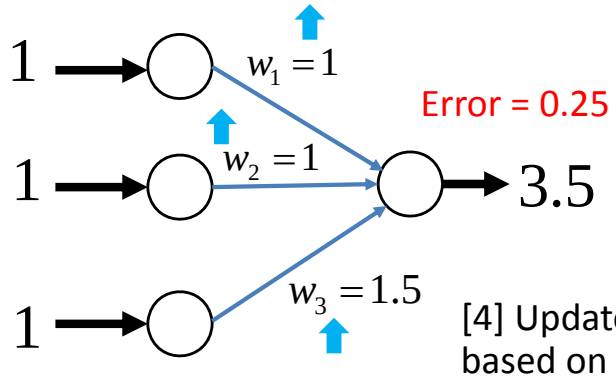


# Back Propagation



Training Set

| $x_1$ | $x_2$ | $x_3$ | $y$ | $\hat{y}$ |
|---|---|---|---|---|
| 0 | 0 | 1 | 2 | 1 |
| 1 | 1 | 1 | 4 | 3.5 |
| −1 | 0 | 2 | 1 | |




Error = 0.25

[4] Update the weights based on Error

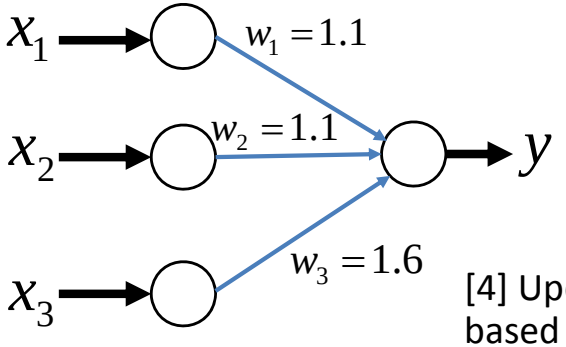


# Back Propagation



Training Set

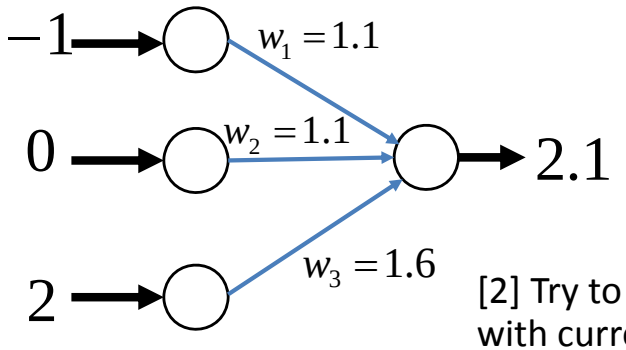
| $x_1$ | $x_2$ | $x_3$ | $y$ | $\hat{y}$ |
|---|---|---|---|---|
| 0 | 0 | 1 | 2 | 1 |
| 1 | 1 | 1 | 4 | 3.5 |
| −1 | 0 | 2 | 1 | |




[4] Update the weights based on Error

# Back Propagation






Training Set

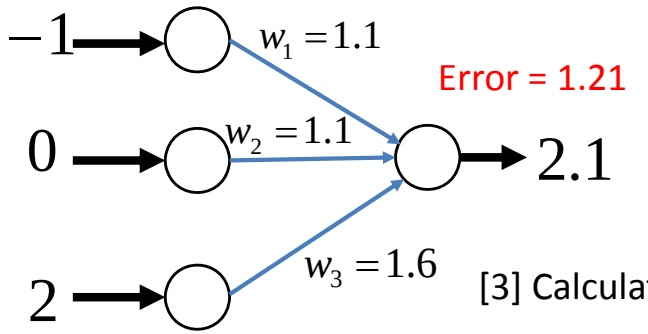
| $x_1$ | $x_2$ | $x_3$ | $y$ | $\hat{y}$ |
|---|---|---|---|---|
| 0 | 0 | 1 | 2 | 1 |
| 1 | 1 | 1 | 4 | 3.5 |
| −1 | 0 | 2 | 1 | 2.1 |

[2] Try to predict one sample with current weight



# Back Propagation






Error = 1.21

Training Set

| $x_1$ | $x_2$ | $x_3$ | $y$ | $\hat{y}$ |
|---|---|---|---|---|
| 0 | 0 | 1 | 2 | 1 |
| 1 | 1 | 1 | 4 | 3.5 |
| −1 | 0 | 2 | 1 | 2.1 |

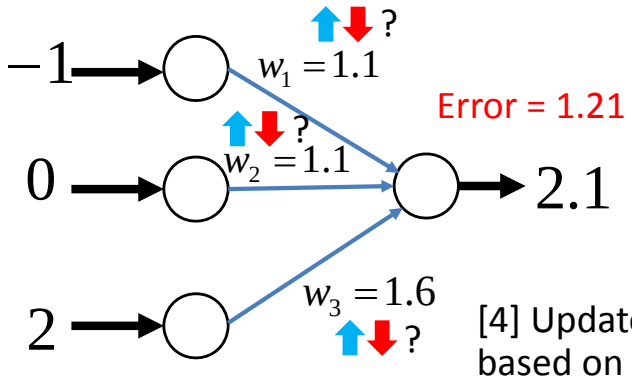
[3] Calculate the error

$$E = (y - \hat{y})^2 = (1 - 2.1)^2 = 1.21$$



# Back Propagation






Error = 1.21

Training Set

| $x_1$ | $x_2$ | $x_3$ | $y$ | $\hat{y}$ |
|---|---|---|---|---|
| 0 | 0 | 1 | 2 | 1 |
| 1 | 1 | 1 | 4 | 3.5 |
| −1 | 0 | 2 | 1 | 2.1 |

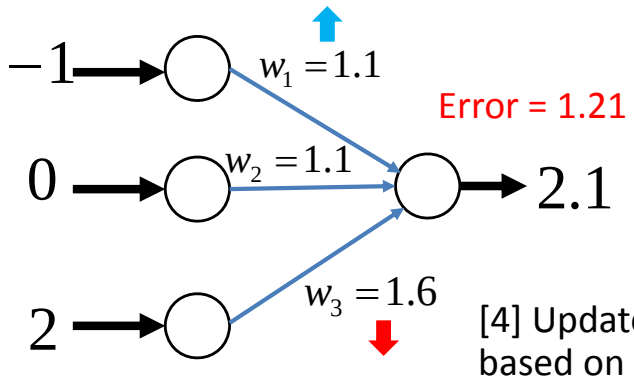
[4] Update the weights based on Error

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



# Back Propagation






Error = 1.21

Training Set

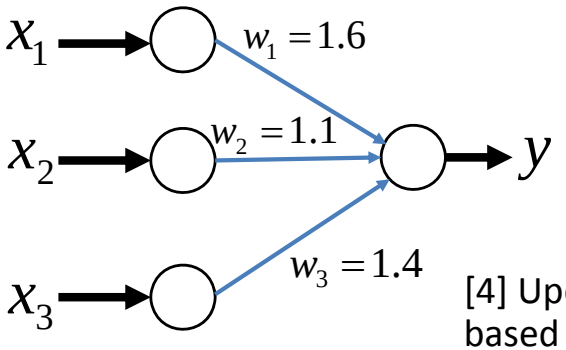
| $x_1$ | $x_2$ | $x_3$ | $y$ | $\hat{y}$ |
|---|---|---|---|---|
| 0 | 0 | 1 | 2 | 1 |
| 1 | 1 | 1 | 4 | 3.5 |
| −1 | 0 | 2 | 1 | 2.1 |

[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






Training Set

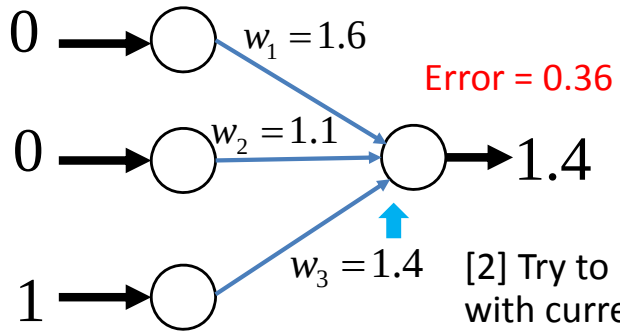
| $x_1$ | $x_2$ | $x_3$ | $y$ | $\hat{y}$ |
|---|---|---|---|---|
| 0 | 0 | 1 | 2 | 1 |
| 1 | 1 | 1 | 4 | 3.5 |
| −1 | 0 | 2 | 1 | 2.1 |

[4] Update the weights based on Error



# Back Propagation






Training Set

| $x_1$ | $x_2$ | $x_3$ | $y$ | $\hat{y}$ |
|---|---|---|---|---|
| 0 | 0 | 1 | 2 | 1.4 |
| 1 | 1 | 1 | 4 | 3.5 |
| −1 | 0 | 2 | 1 | 2.1 |

[2] Try to predict one sample with current weight

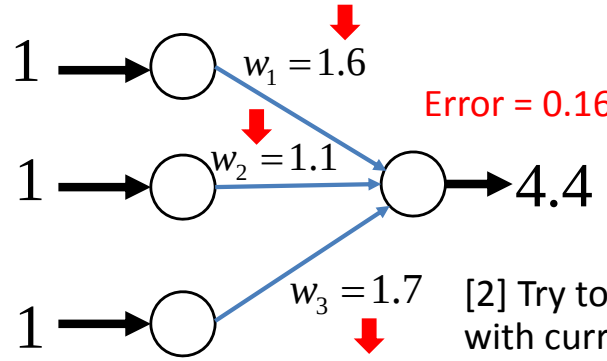
[3] Calculate the error

[4] Update the weights based on Error



# Back Propagation






Training Set

| $x_1$ | $x_2$ | $x_3$ | $y$ | $\hat{y}$ |
|---|---|---|---|---|
| 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

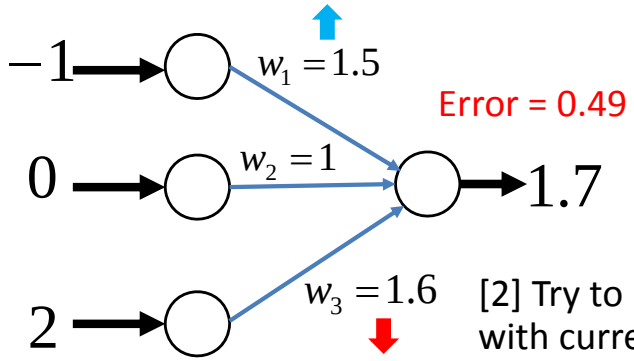
[3] Calculate the error

[4] Update the weights based on Error



# Back Propagation






Training Set

| $x_1$ | $x_2$ | $x_3$ | $y$ | $\hat{y}$ |
|---|---|---|---|---|
| 0 | 0 | 1 | 2 | 1.4 |
| 1 | 1 | 1 | 4 | 4.4 |
| −1 | 0 | 2 | 1 | 1.7 |

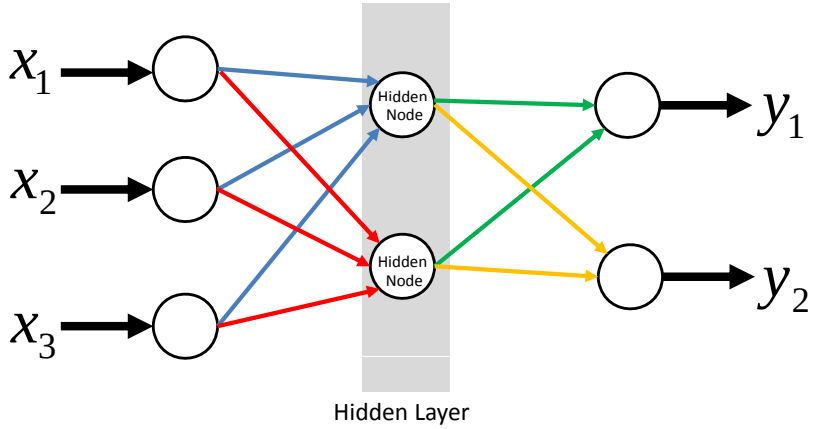
[2] Try to predict one sample with current weight

[3] Calculate the error

[4] Update the weights based on Error

# Artificial Neural Network

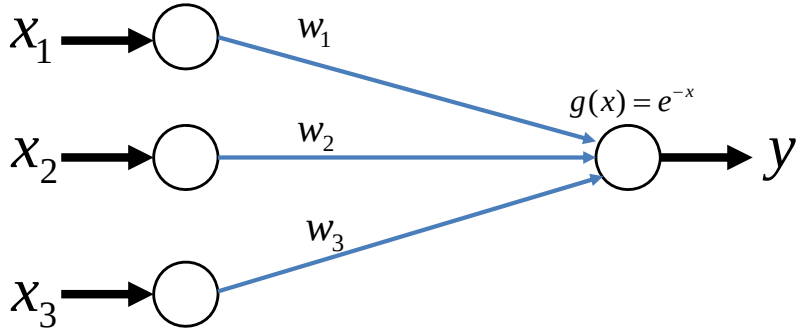







# Artificial Neural Network



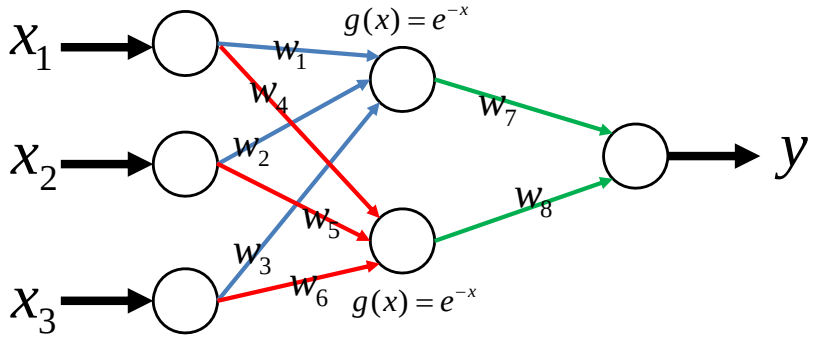



$$y = e^{-(w_1 x_1 + w_2 x_2 + w_3 x_3)}$$



# Artificial Neural Network



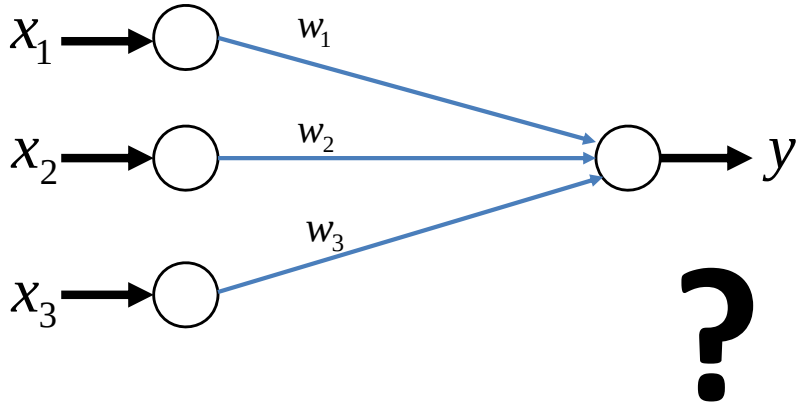



$$y = w_7 e^{-(w_1 x_1 + w_2 x_2 + w_3 x_3)} + w_8 e^{-(w_4 x_1 + w_5 x_2 + w_6 x_3)}$$



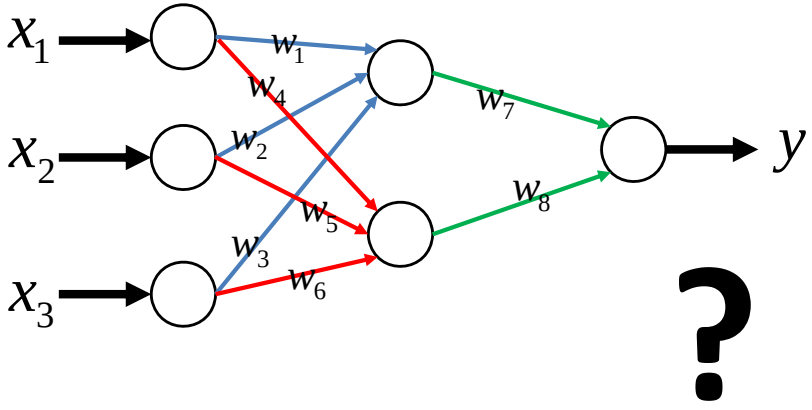
# Artificial Neural Network





# Artificial Neural Network



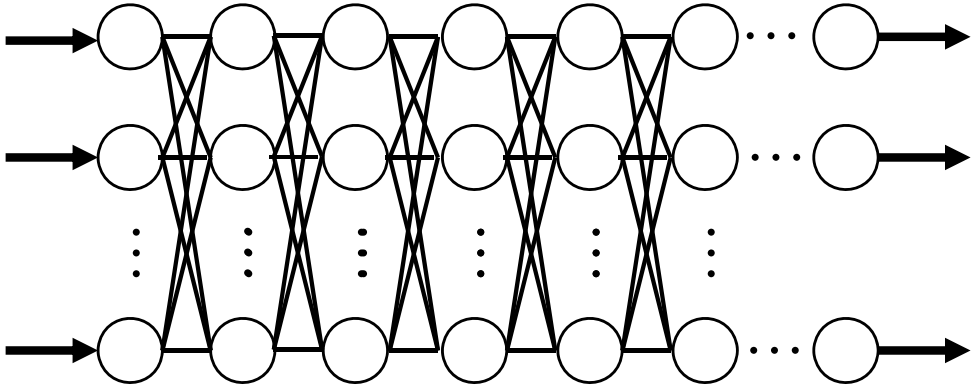





# Deep learning



เชรี้ยแม่ง Deep ....





# Image Filter



Input Image

| | | | |
|---|---|---|---|
| $x_1$ | $x_2$ | $x_3$ | $x_4$ |
| $x_5$ | $x_6$ | $x_7$ | $x_8$ |
| $x_9$ | $x_{10}$ | $x_{11}$ | $x_{12}$ |
| $x_{13}$ | $x_{14}$ | $x_{15}$ | $x_{16}$ |

Filter Kernel

| | |
|---|---|
| $w_1$ | $w_2$ |
| $w_3$ | $w_4$ |

(Convolution kernel)
(Flipped)

Filtered Image

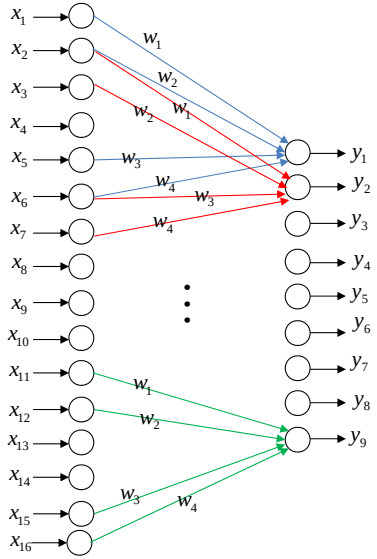
| | | |
|---|---|---|
| $y_1$ | $y_2$ | $y_3$ |
| $y_4$ | $y_5$ | $y_6$ |
| $y_7$ | $y_8$ | $y_9$ |

$$y_1 = w_1x_1 + w_2x_2 + w_3x_5 + w_4x_6$$
$$y_2 = w_1x_2 + w_2x_3 + w_3x_6 + w_4x_7$$
$$\vdots$$
$$y_9 = w_1x_{11} + w_2x_{12} + w_3x_{15} + w_4x_{16}$$



# Image Filter

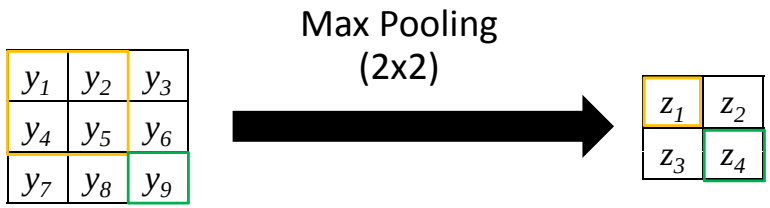





$$y_1 = w_1x_1 + w_2x_2 + w_3x_5 + w_4x_6$$
$$y_2 = w_1x_2 + w_2x_3 + w_3x_6 + w_4x_7$$
$$\vdots$$
$$y_9 = w_1x_{11} + w_2x_{12} + w_3x_{15} + w_4x_{16}$$

# Pooling (Downsampling)






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

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

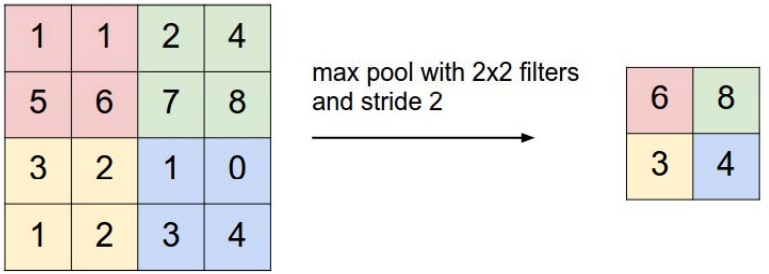
$$z_3 = \max(y_7, y_8)$$

$$z_4 = y_9$$



# Pooling (Downsampling)



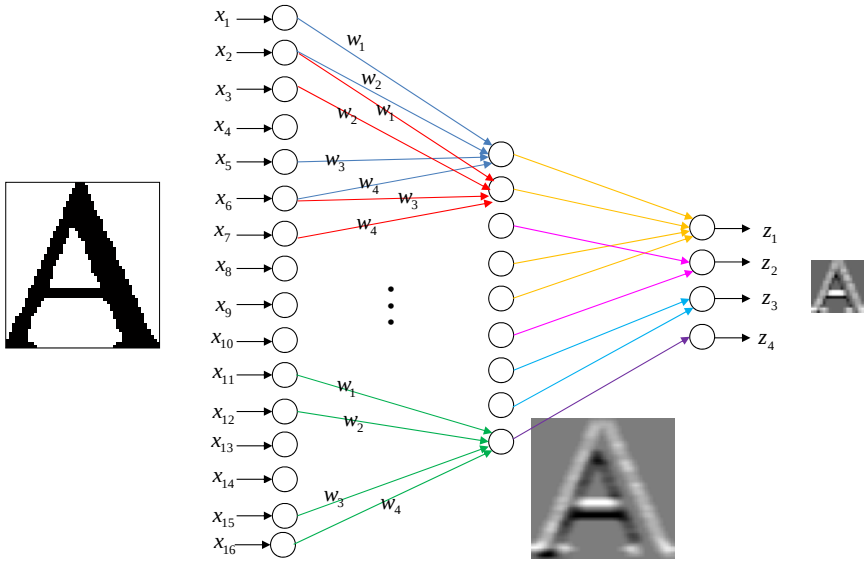



http://cs231n.github.io/convolutional-networks/



# Image Filter + Pooling

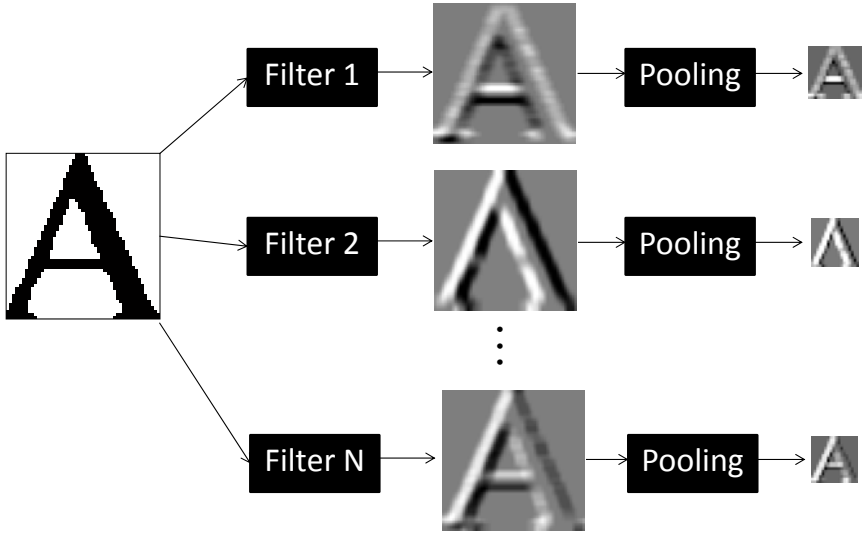






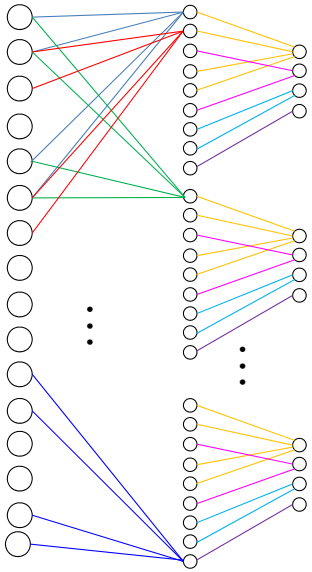

# Image Filter + Pooling

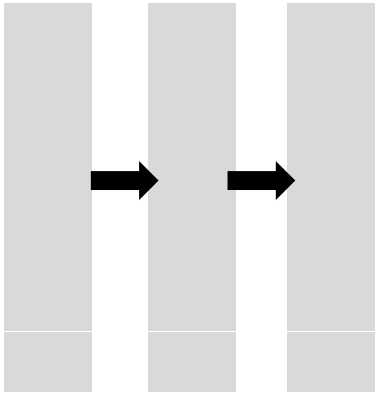




# Image Filter + Pooling



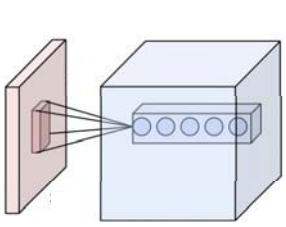




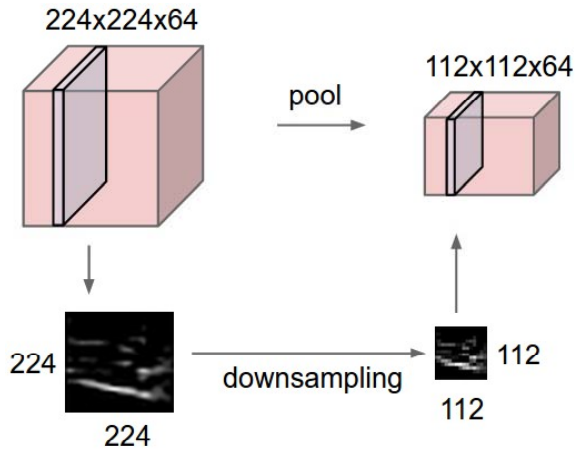



# Image Filter + Pooling





Convolution with various filter

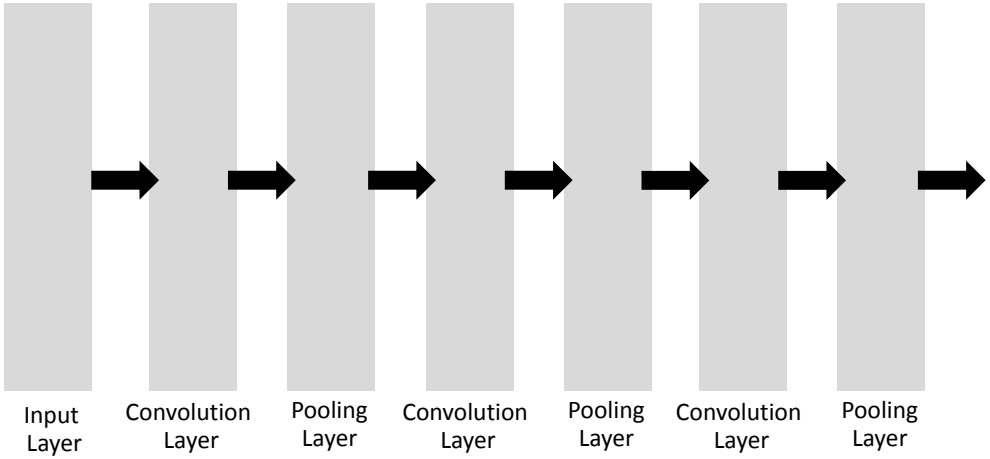



http://cs231n.github.io/convolutional-networks/



# Image Filter + Pooling

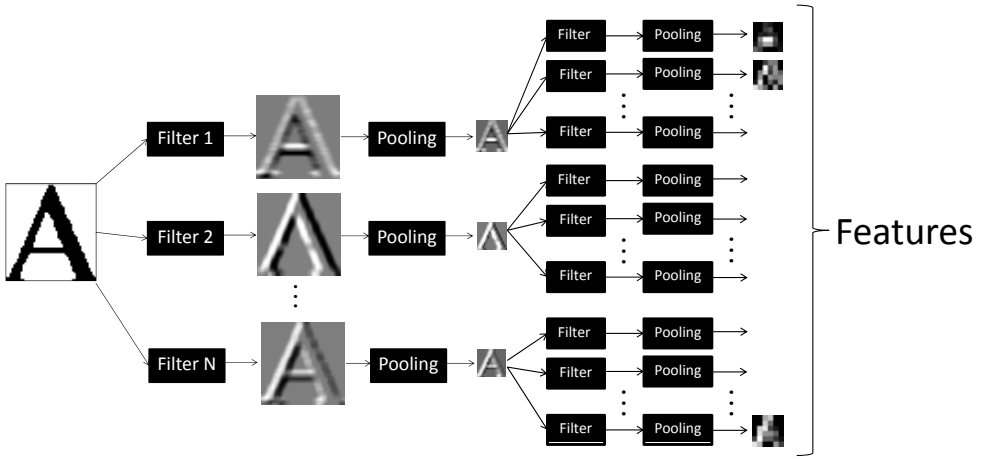






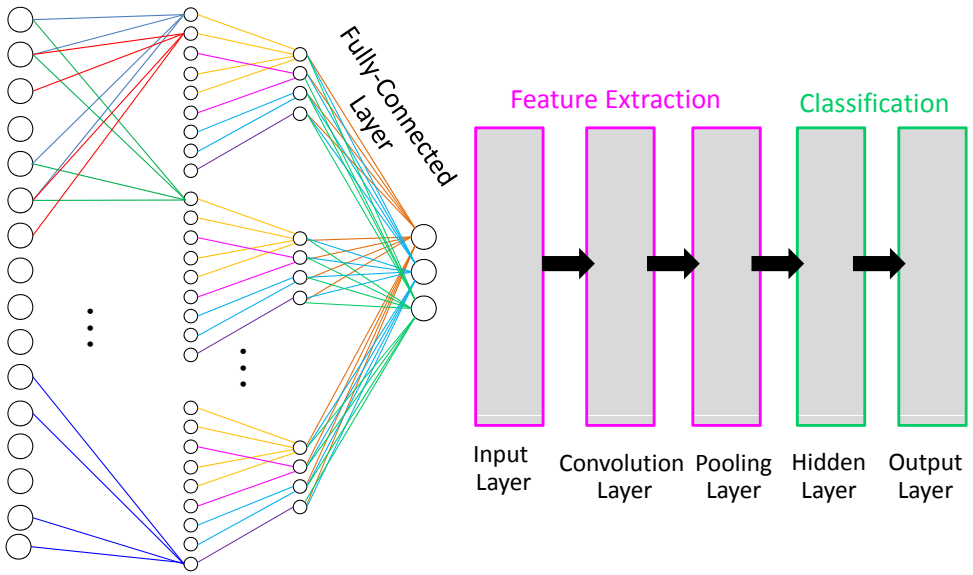

# Image Filter + Pooling





# Convolutional Neural Network (CNN)

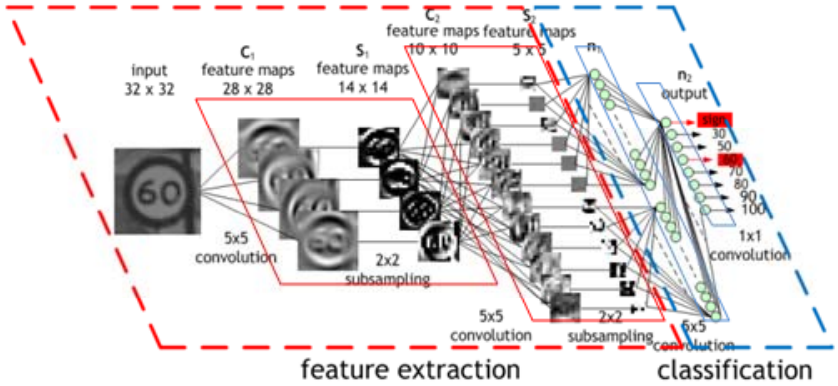







# Convolutional Neural Network (CNN)



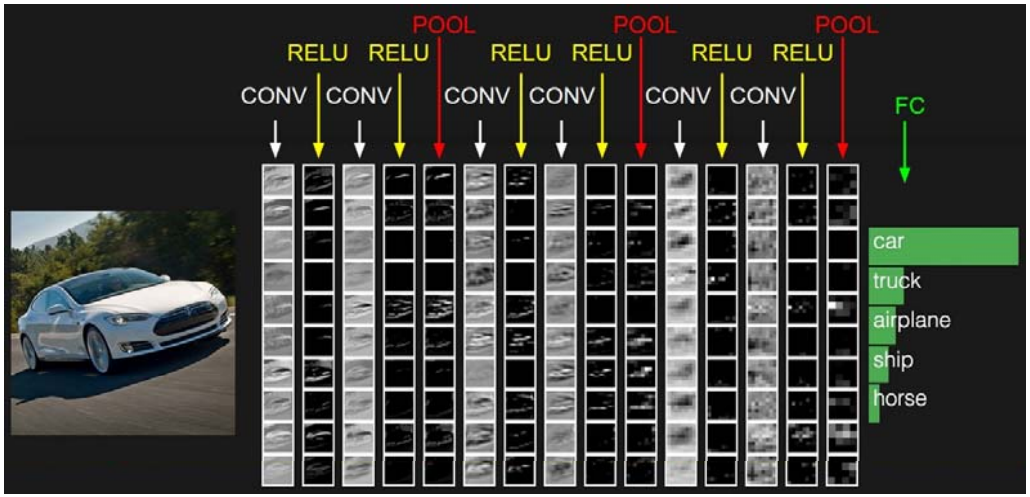



https://devblogs.nvidia.com/wp-content/uploads/2015/11/fig1.png



# Convolutional Neural Network (CNN)



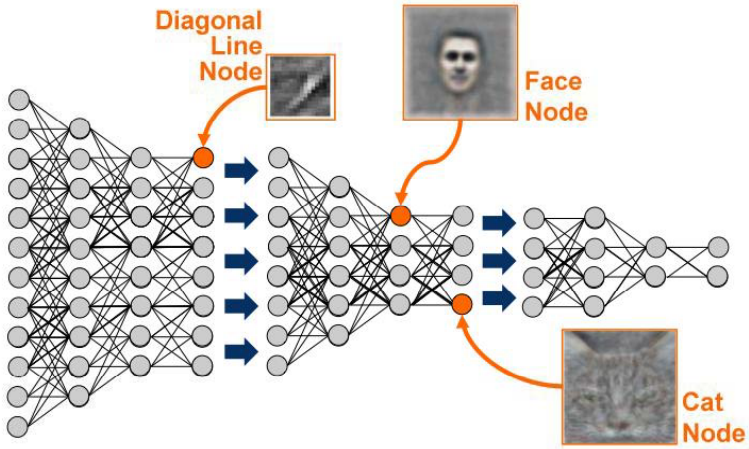



http://cs231n.github.io/convolutional-networks/



# Convolutional Neural Network (CNN)

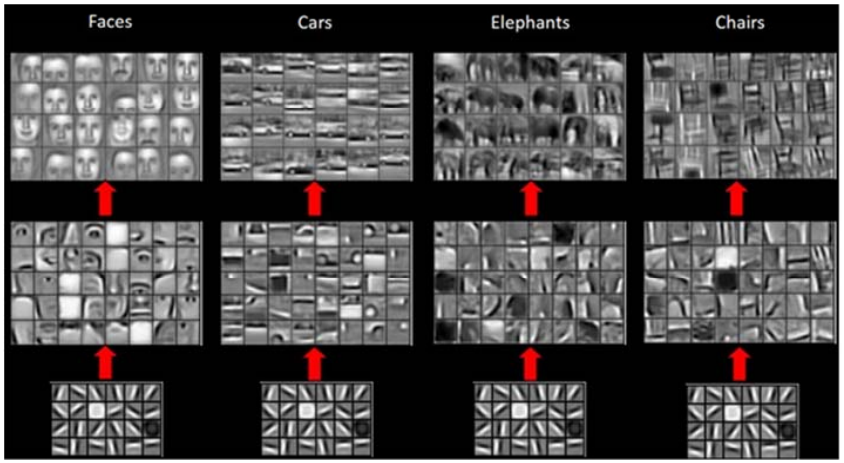





http://www.kdnuggets.com/2015/11/crazy-deep-learning-topological-data-analysis.html

# Convolutional Neural Network (CNN)





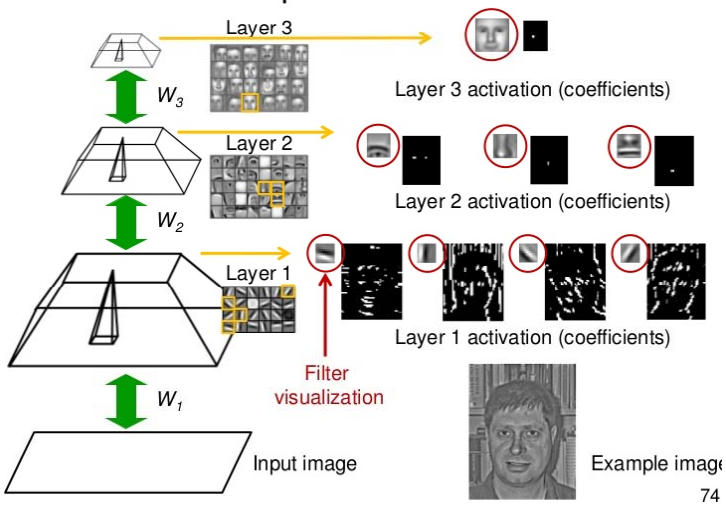

http://stats.stackexchange.com/questions/146413/why-convolutional-neural-networks-belong-to-deep-learning



# Convolutional Neural Network (CNN)



Convolutional deep belief networks illustration




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