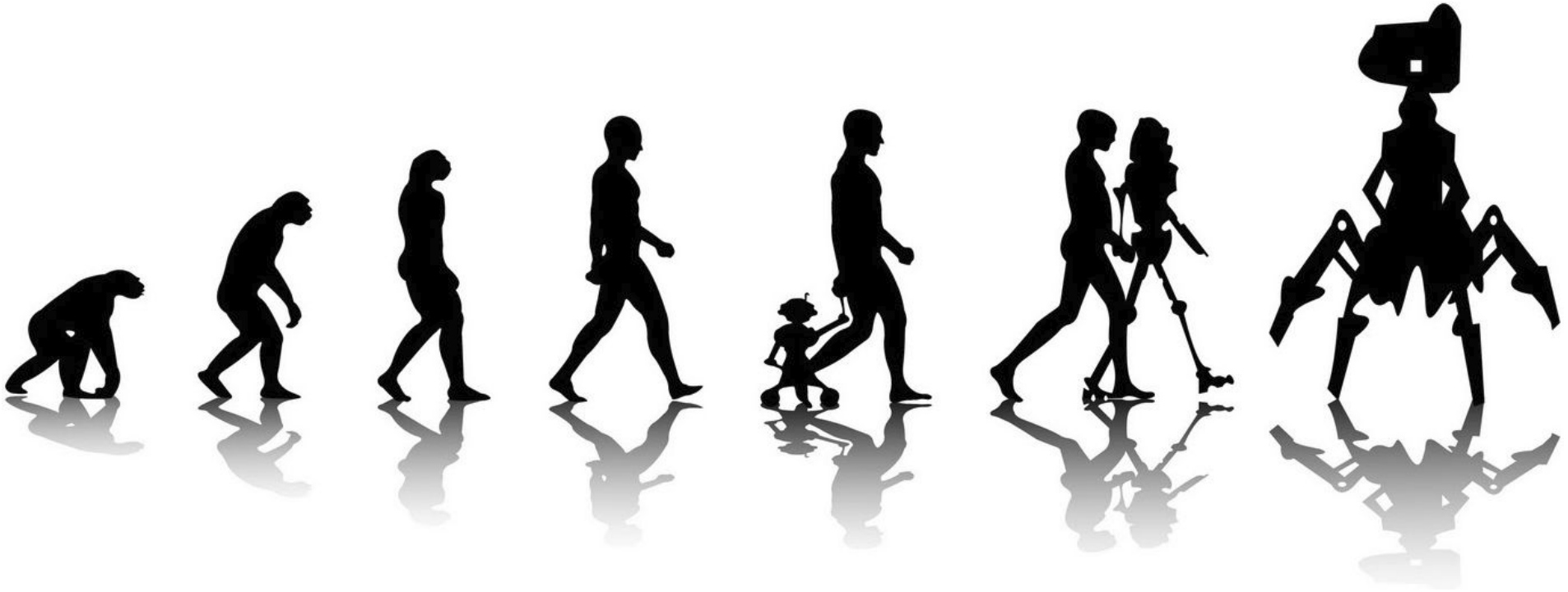


Modern Robotics: Evolutionary Robotics

COSC 4560 / COSC 5560

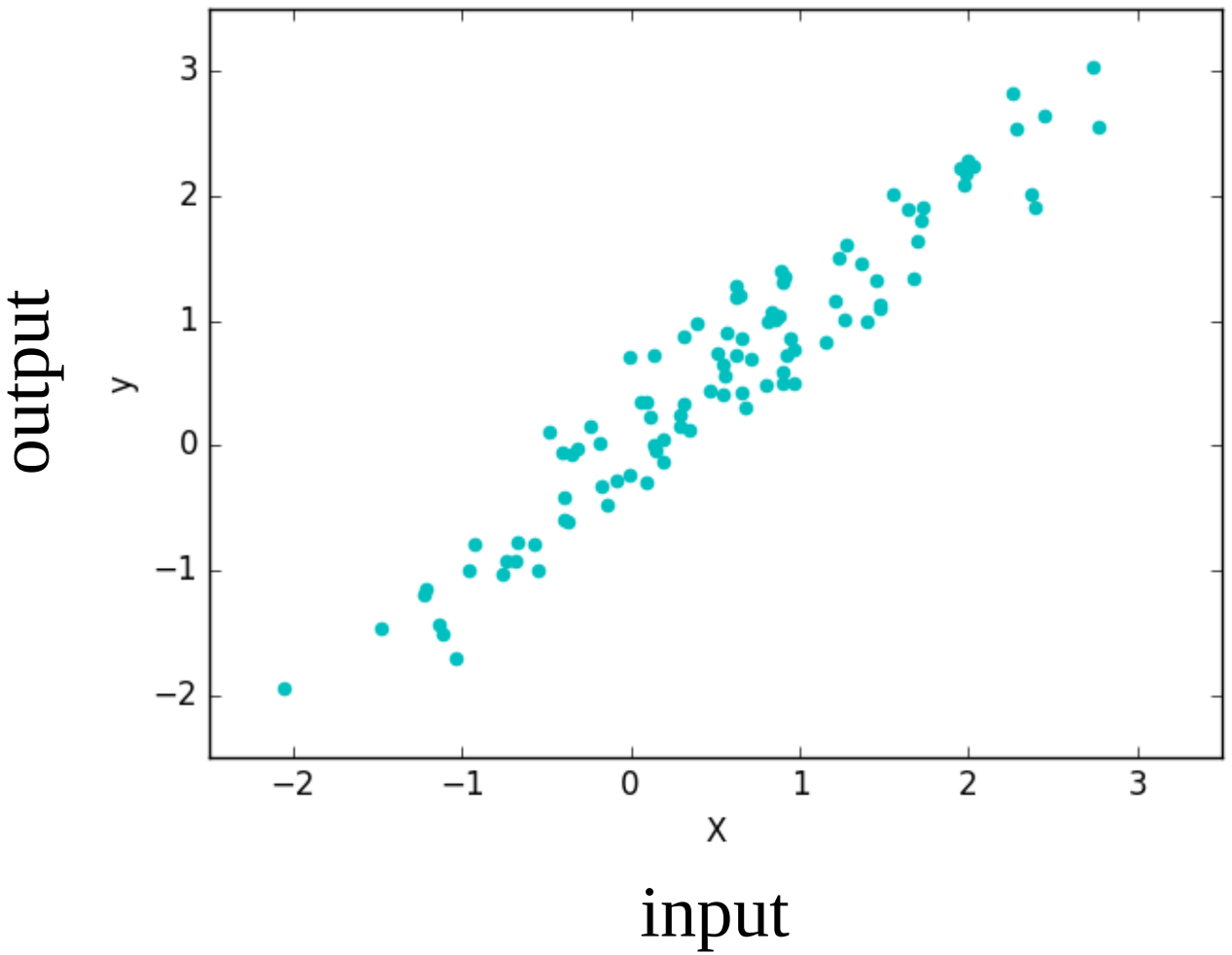
Professor Cheney

4/16/18

Other Ways to Train a Neural Network

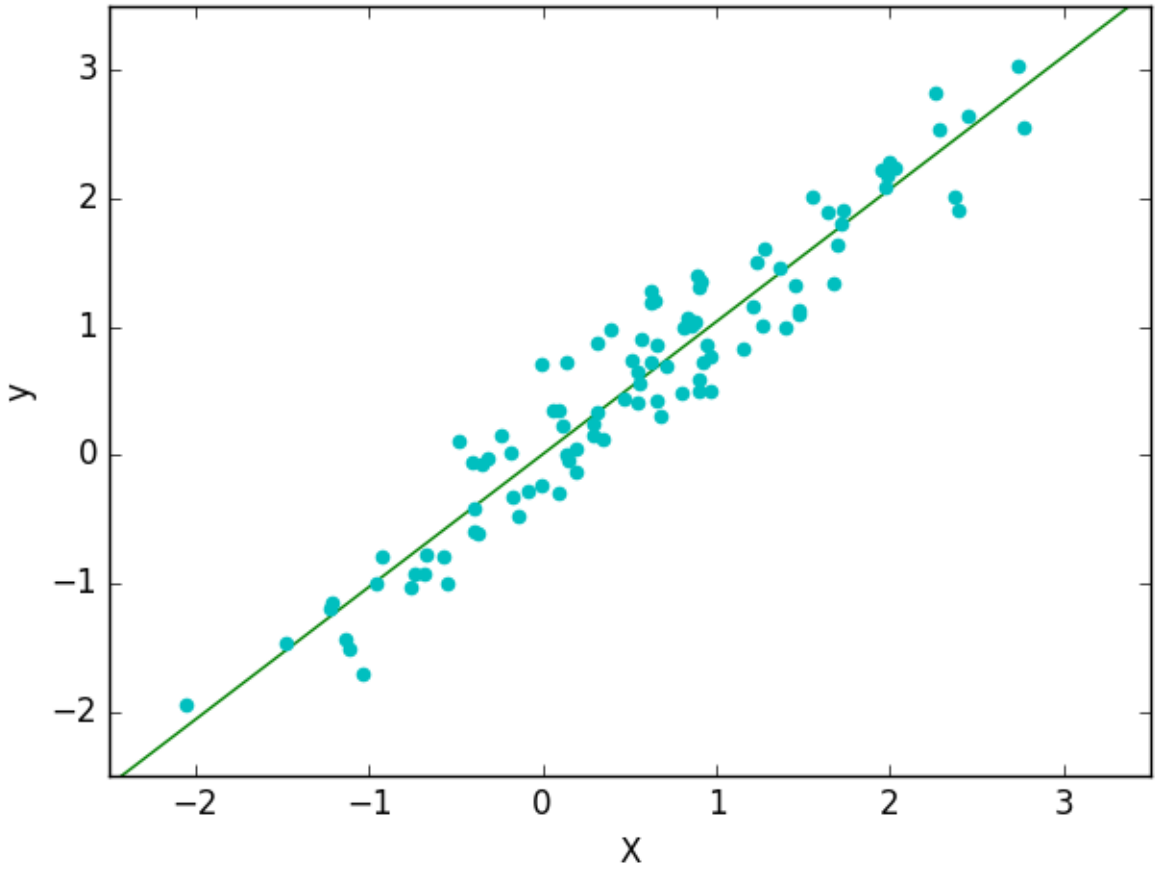
Feature Detection
(for Decision Making)

Logistic Regression

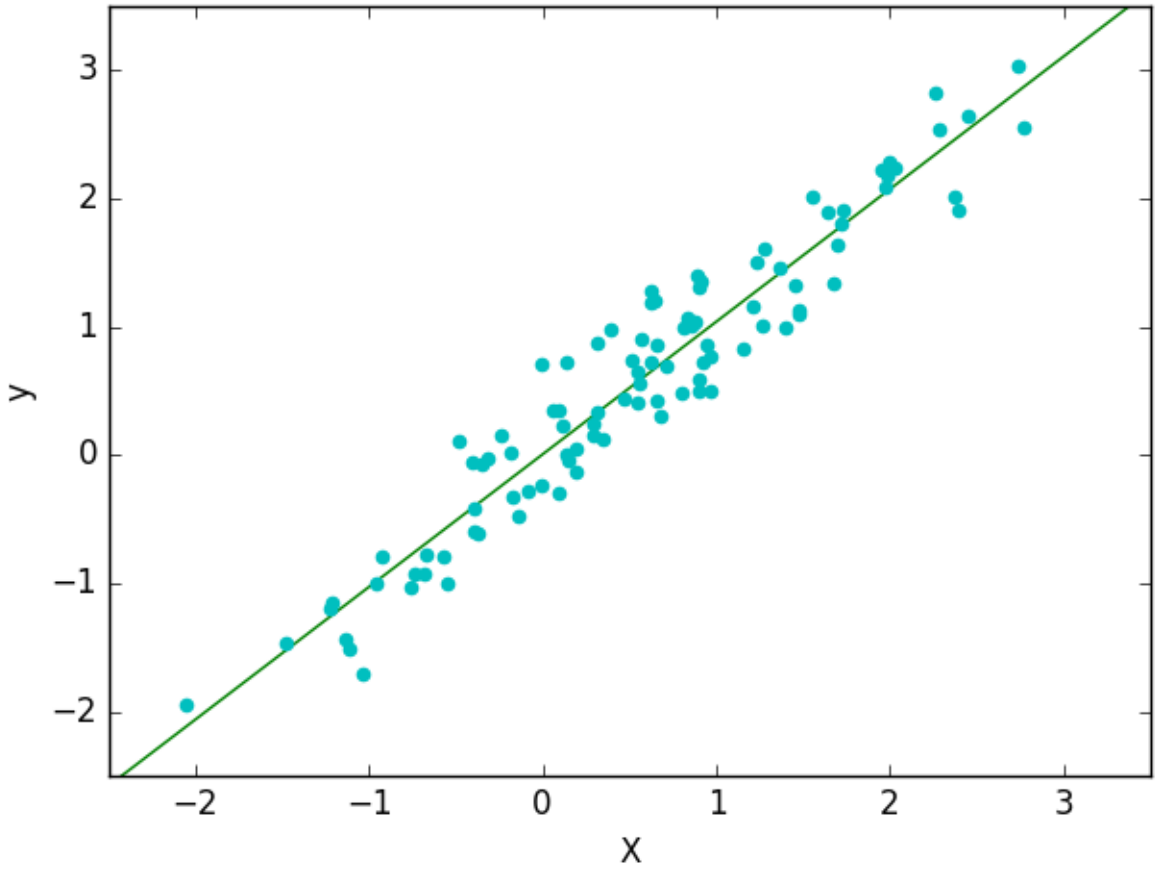

output
y
3
2
1
0
−1
−2
−2
−1
0
1
2
3
X
input

linear regression

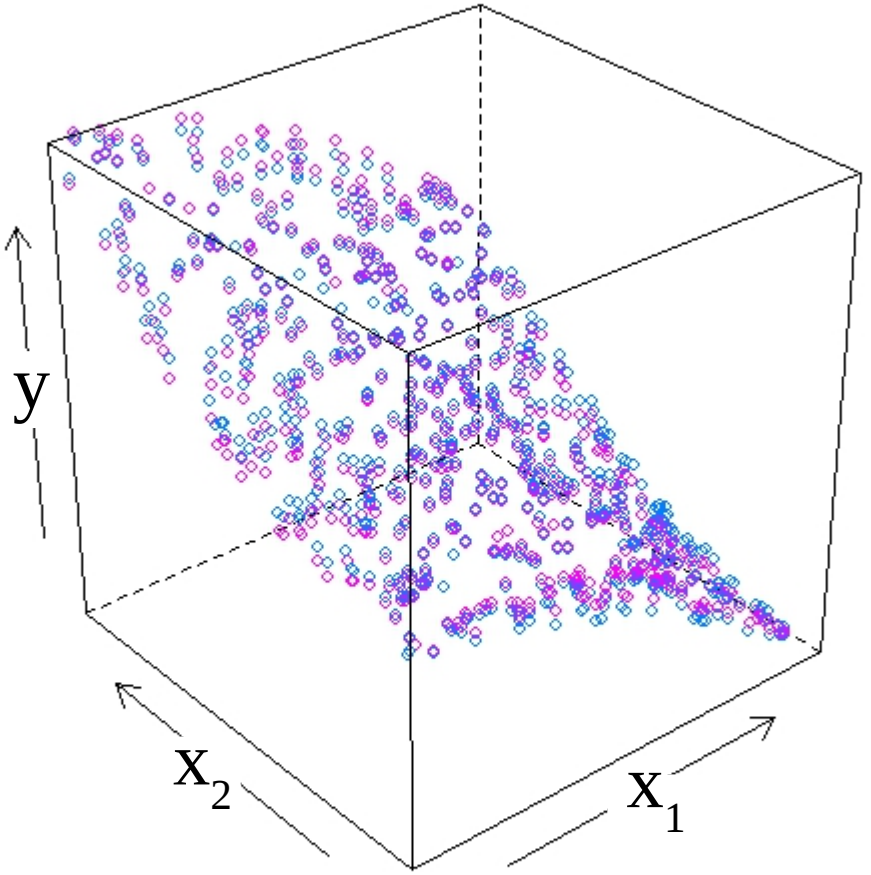
$y = a * x + b$



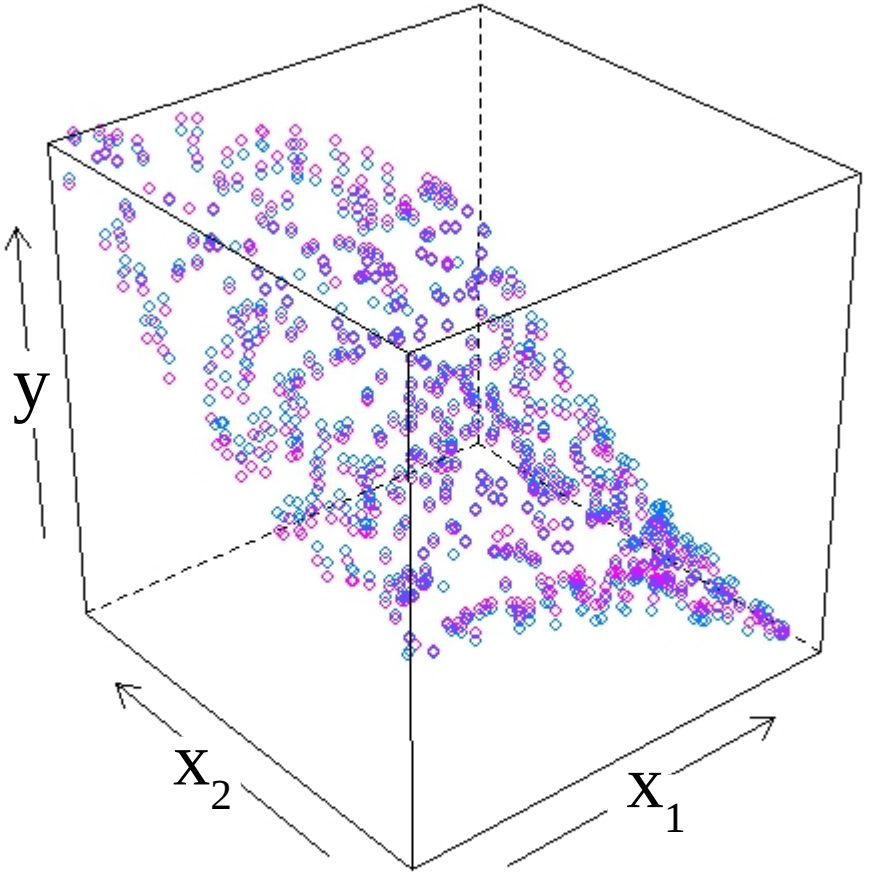
$$y = w * x + b$$

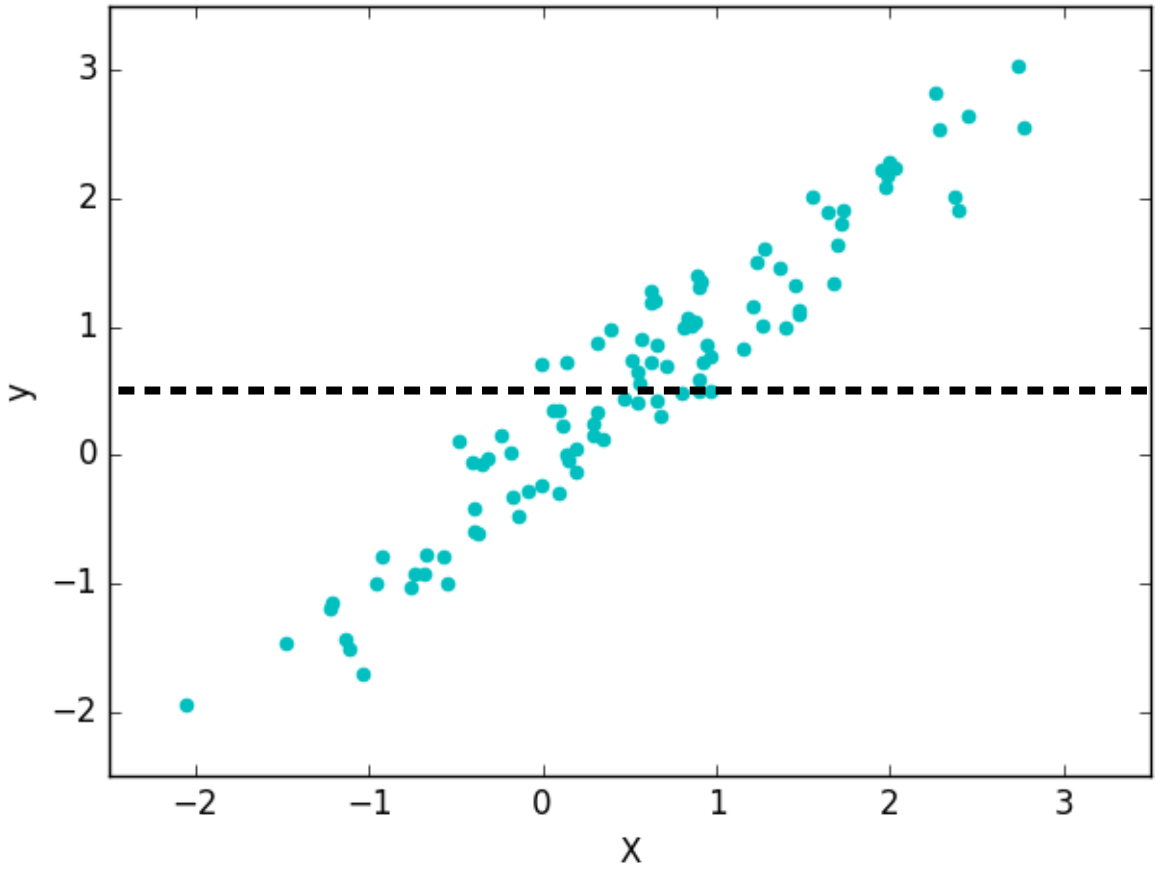


$$y = w_1 * x_1 + w_2 * x_2 + b$$

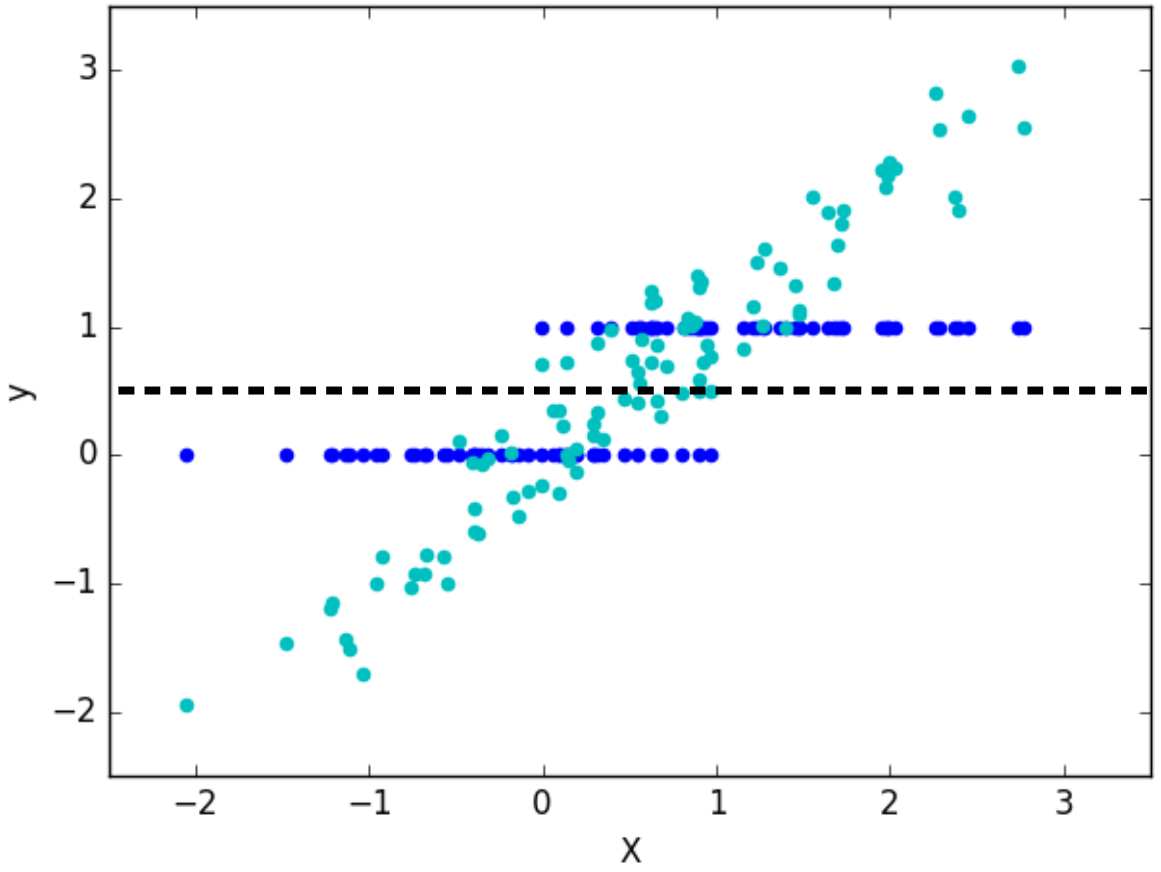


$$y = \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} \bullet \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + b$$

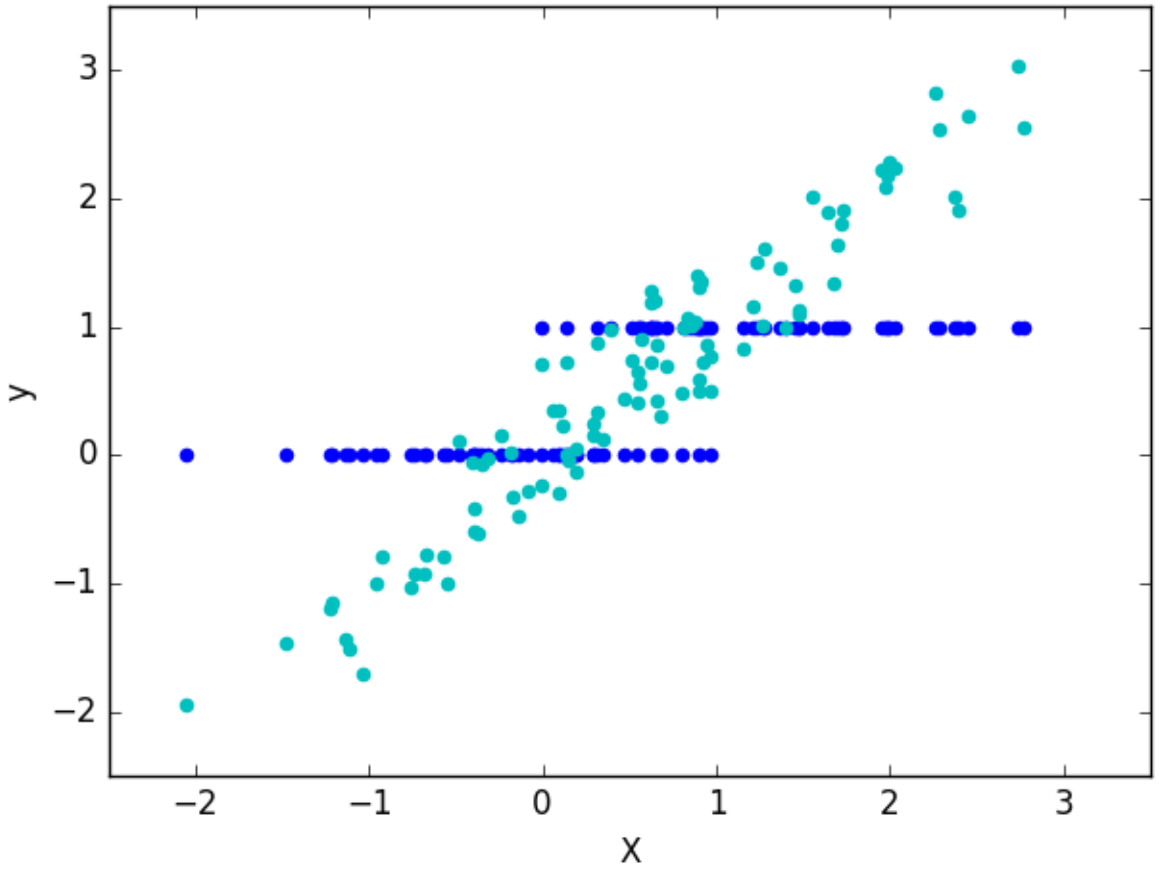



3
2
1
0
−1
−2
y
−2
−1
0
1
2
3
X

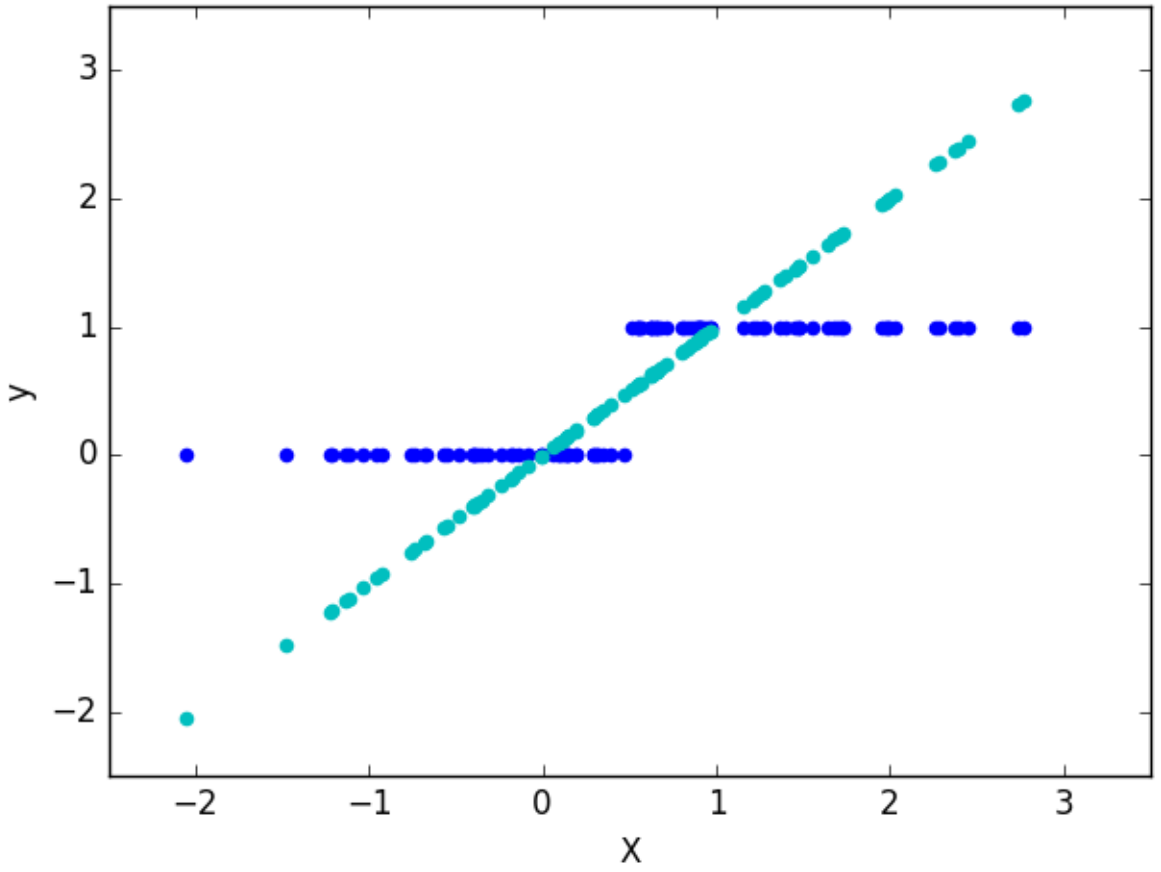
regression → classification



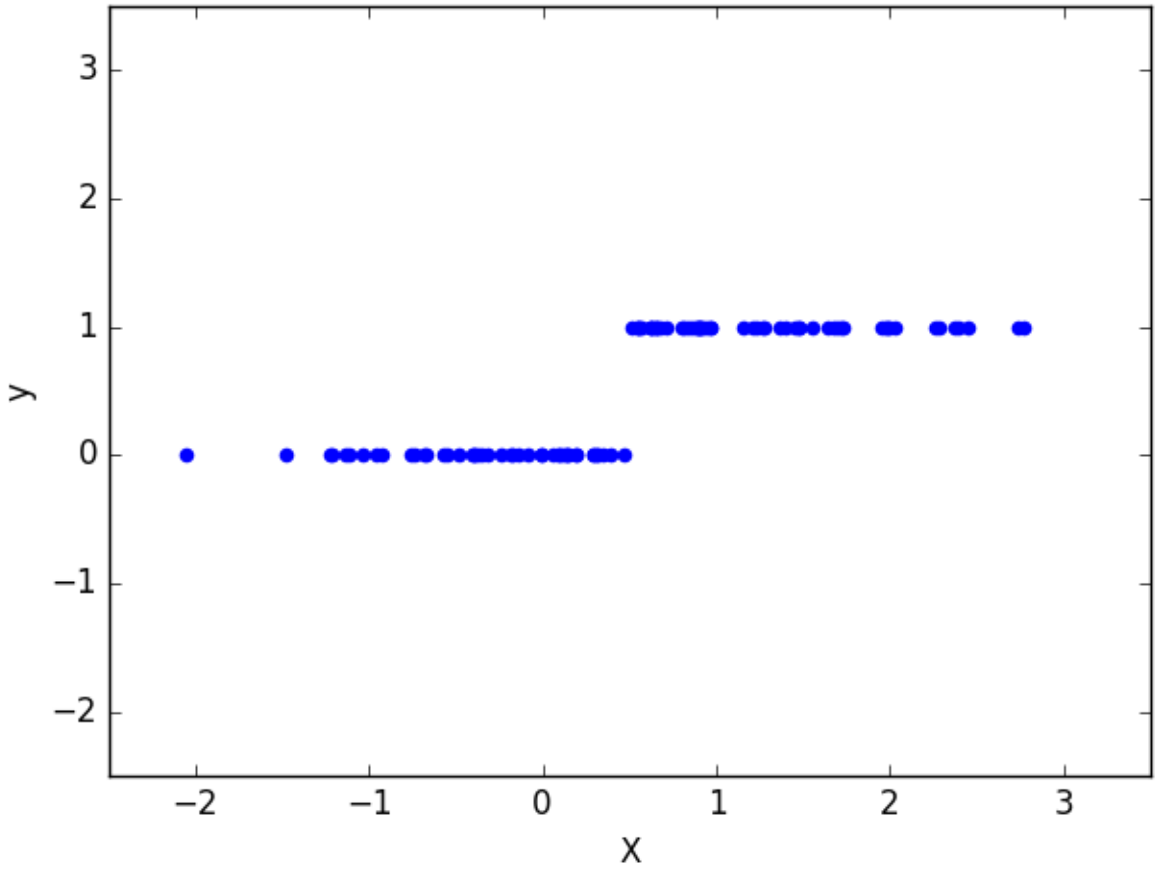
regression → classification



(without noise)



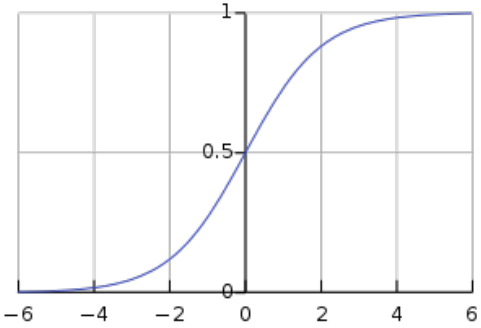
(without noise)

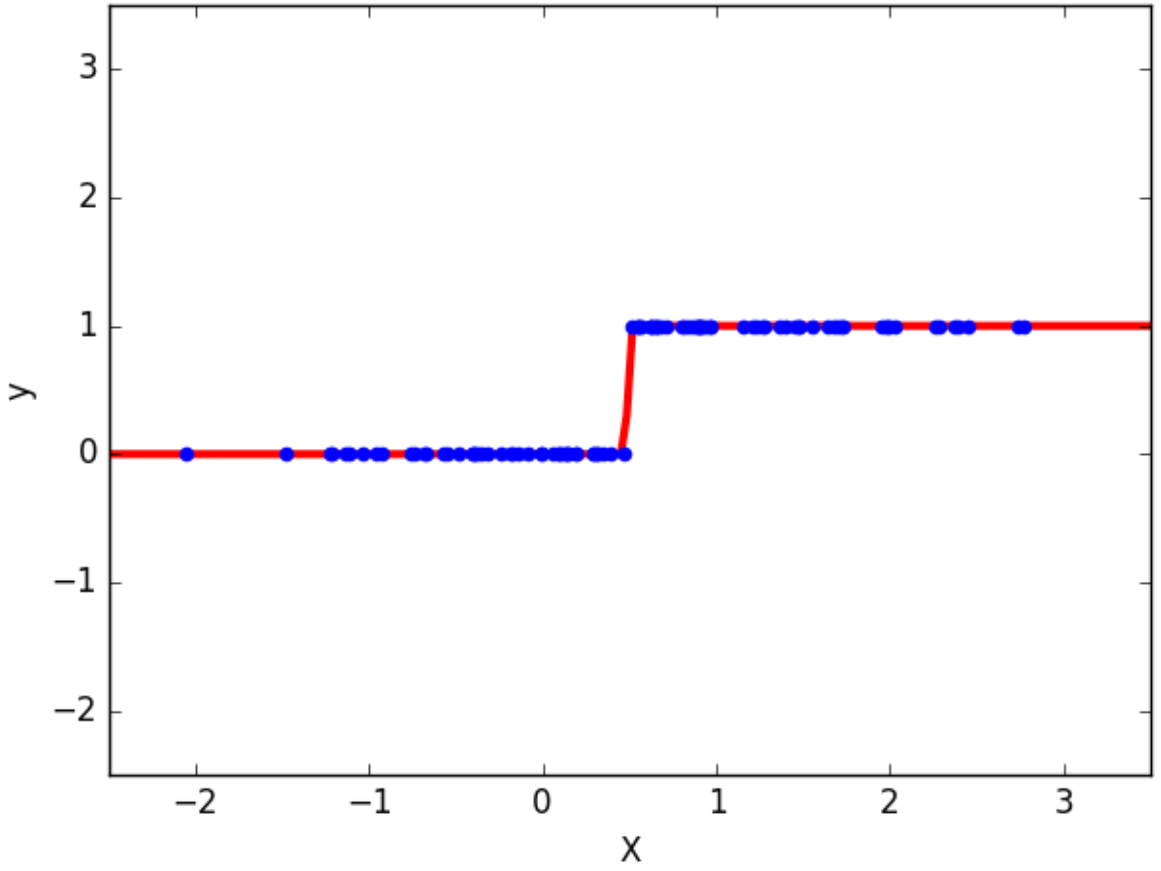


logistic (sigmoid) function

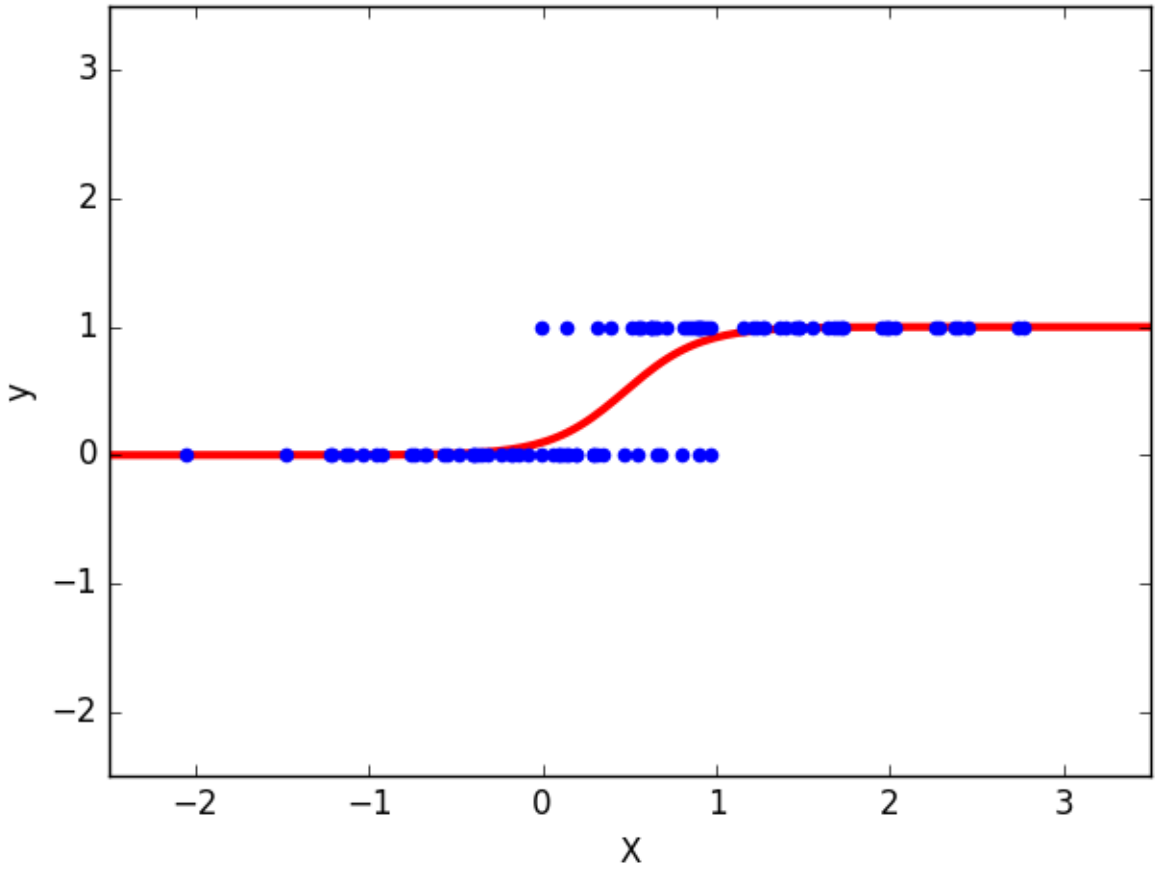
logistic regression

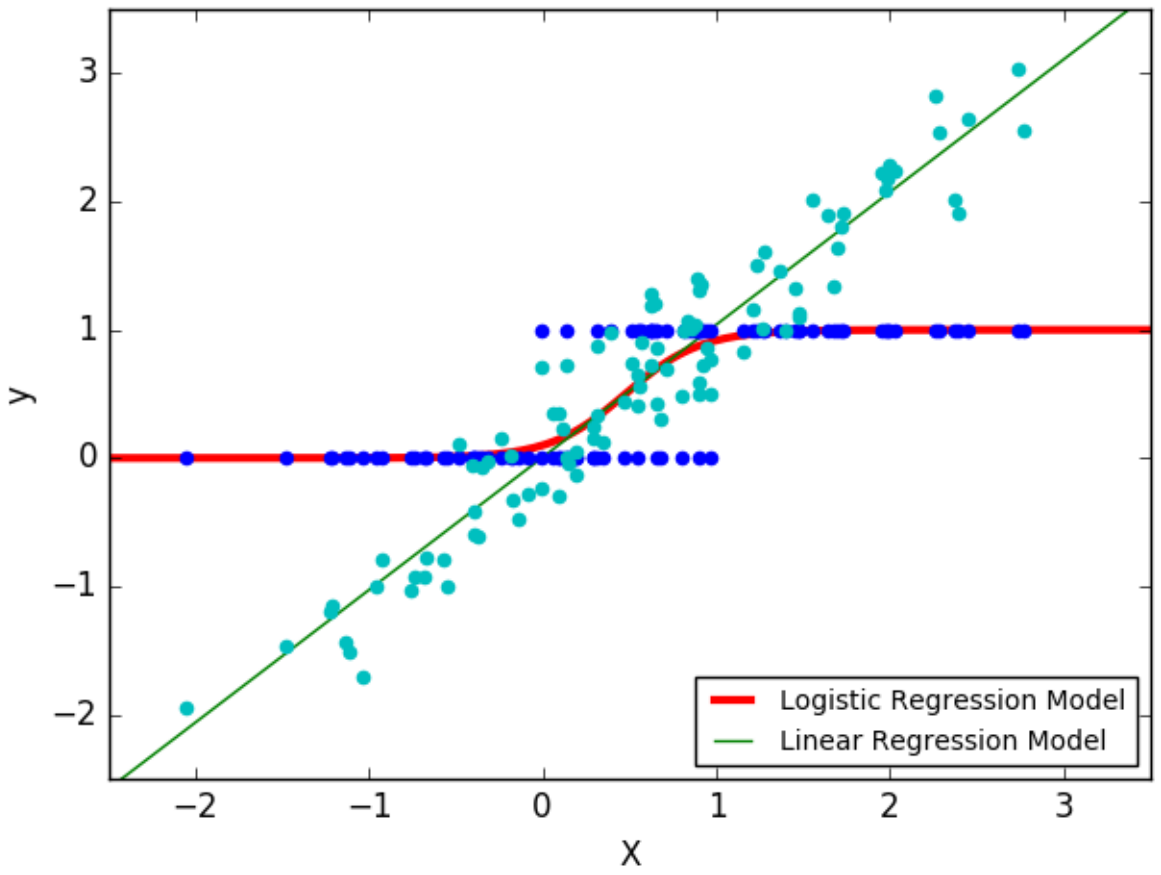
$$y = \sigma (w * x + b)$$



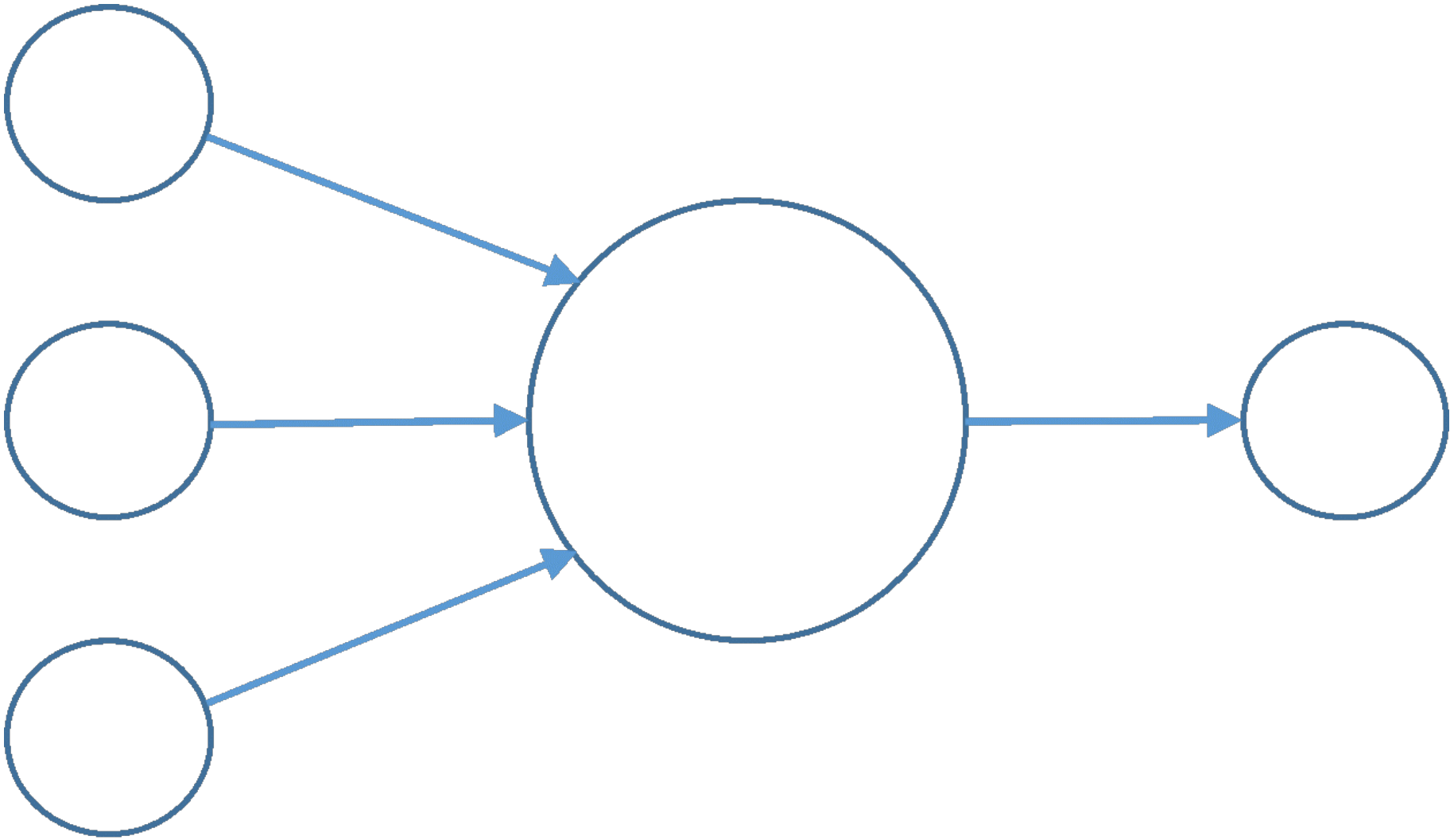


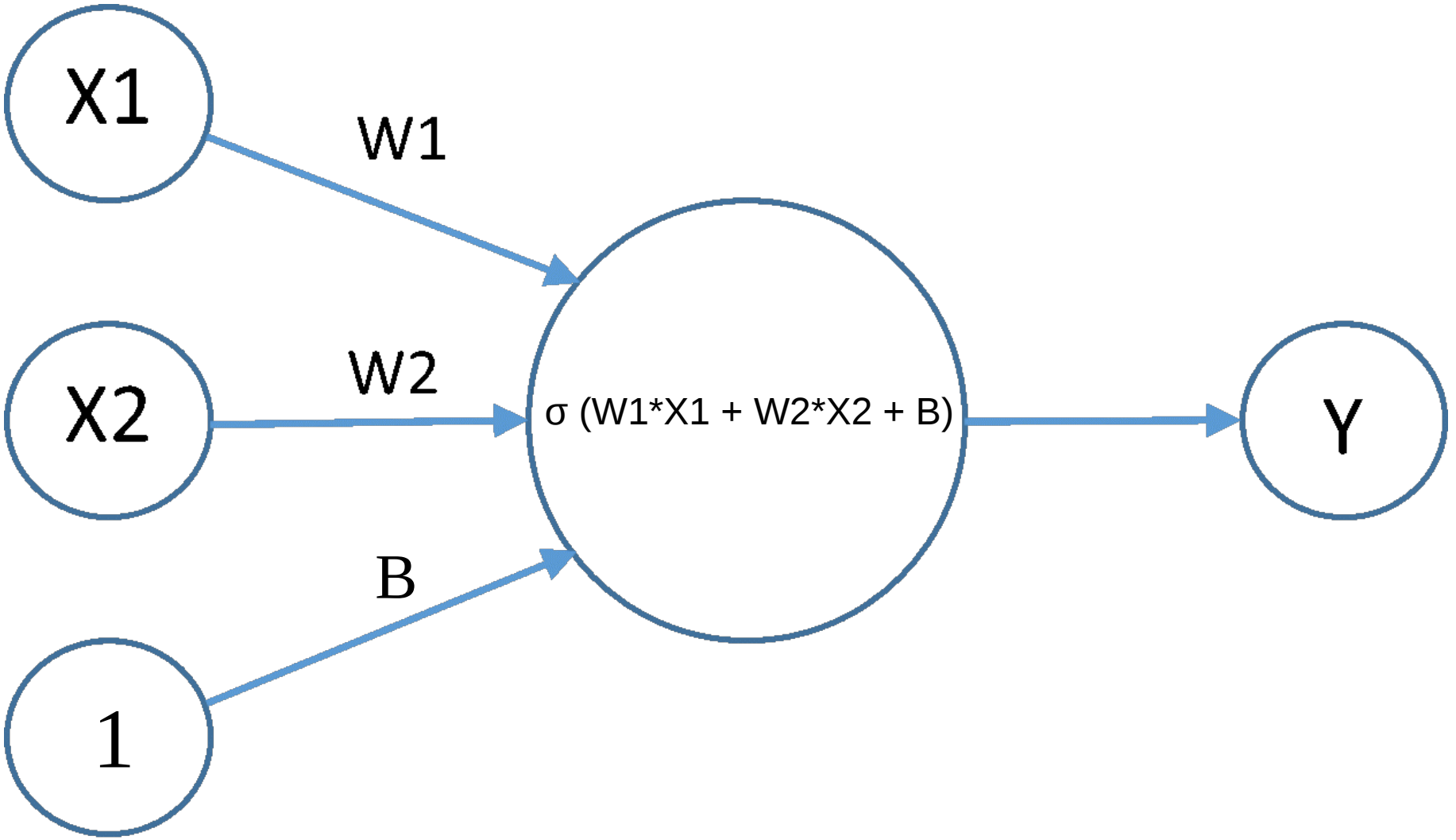
(with the noise)

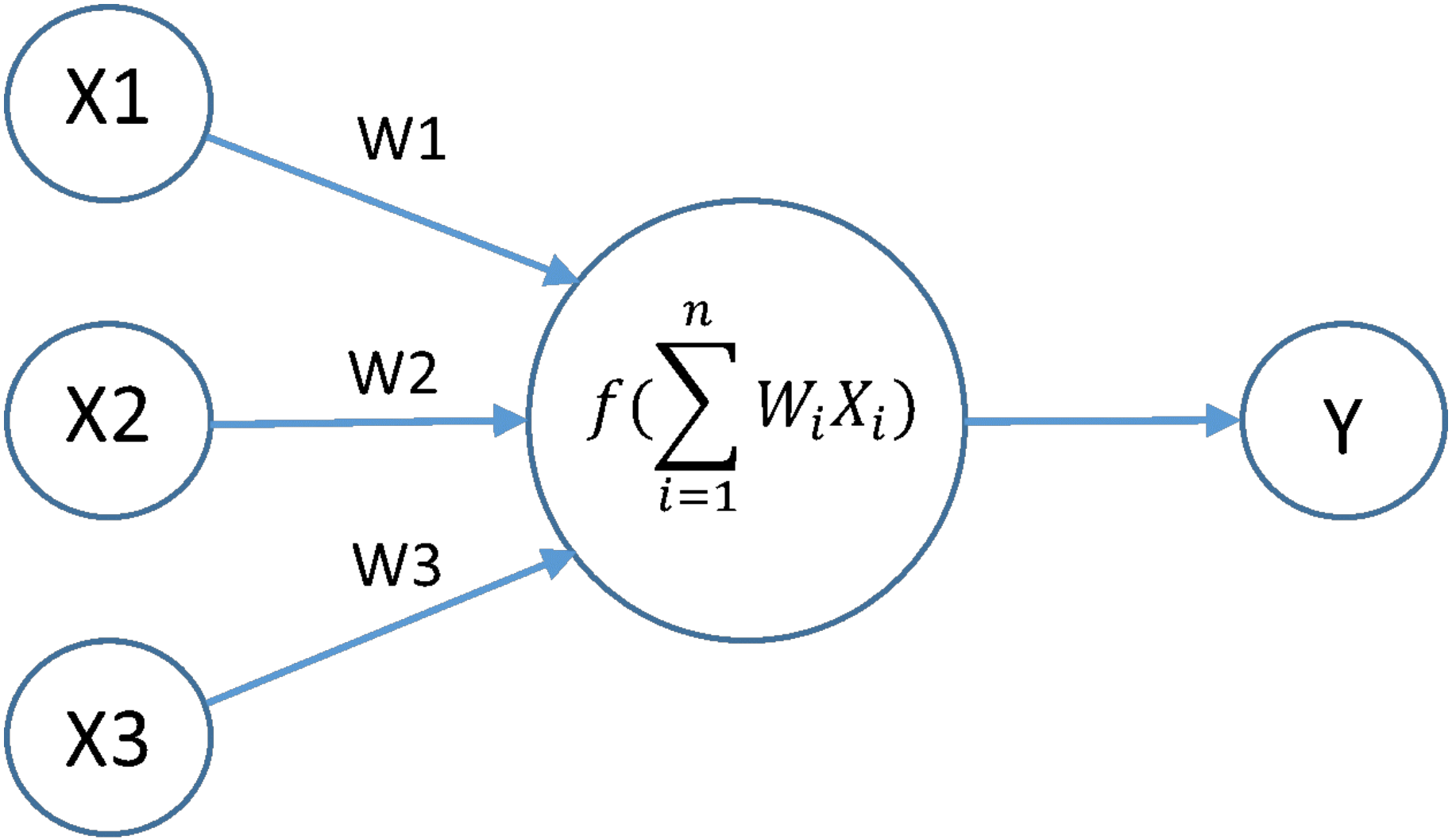



3
2
1
0
−1
−2
y
−2
−1
0
1
2
3
X
Logistic Regression Model
Linear Regression Model

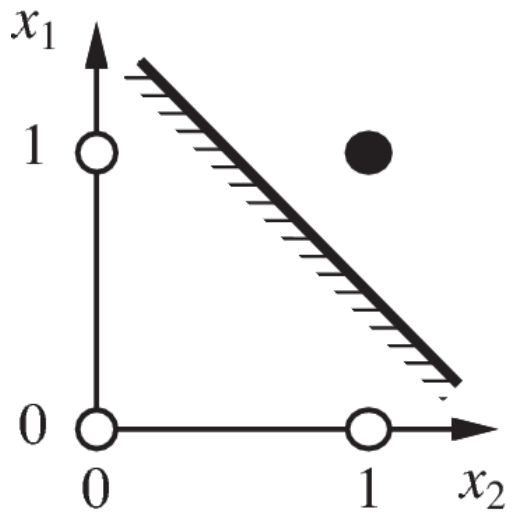
Perceptron



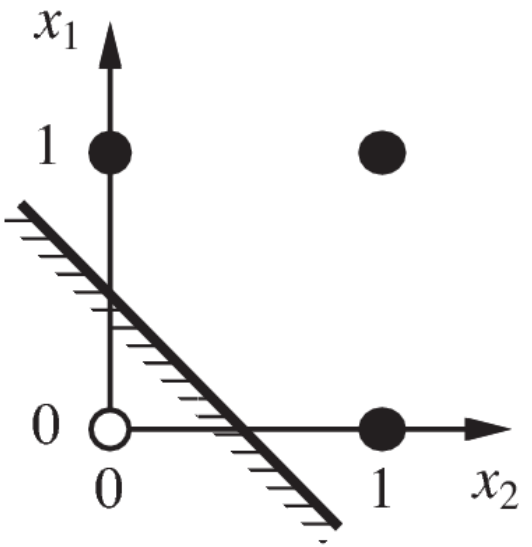

X1
W1
X2
W2
σ (W1*X1 + W2*X2 + B)
Y
B
1


X1
X2
X3
W1
W2
W3
$f(\sum_{i=1}^{n} W_i X_i)$
Y

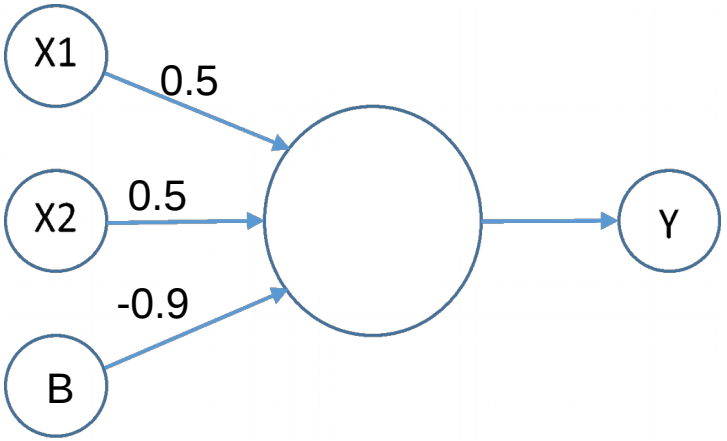
perceptron as logical opperator

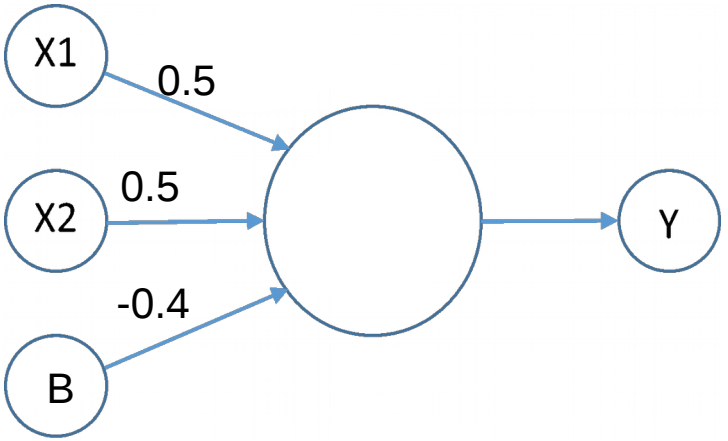


(a) x_1 **and** x_2

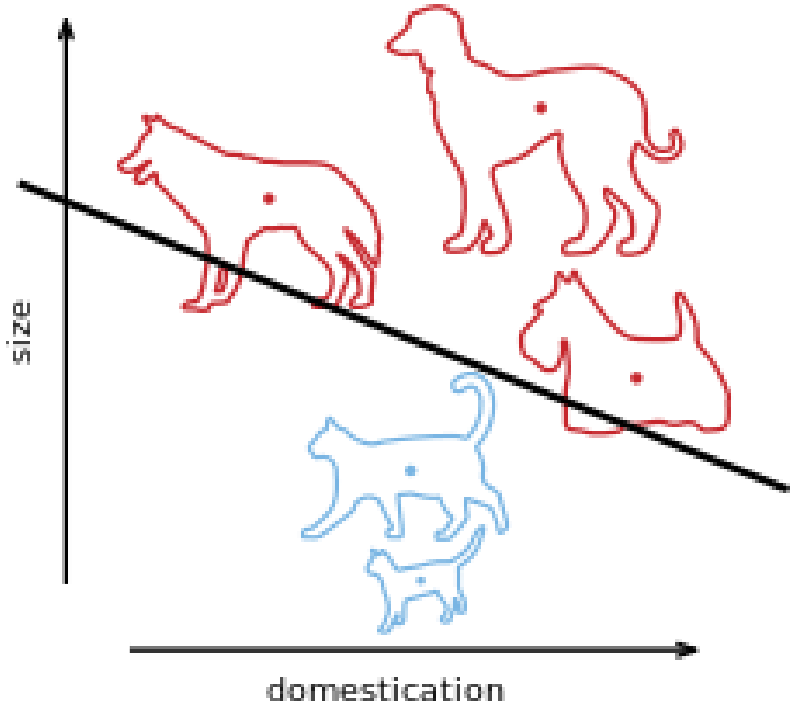


(b) x_1 **or** x_2





axes are features of your data



which features should we use???

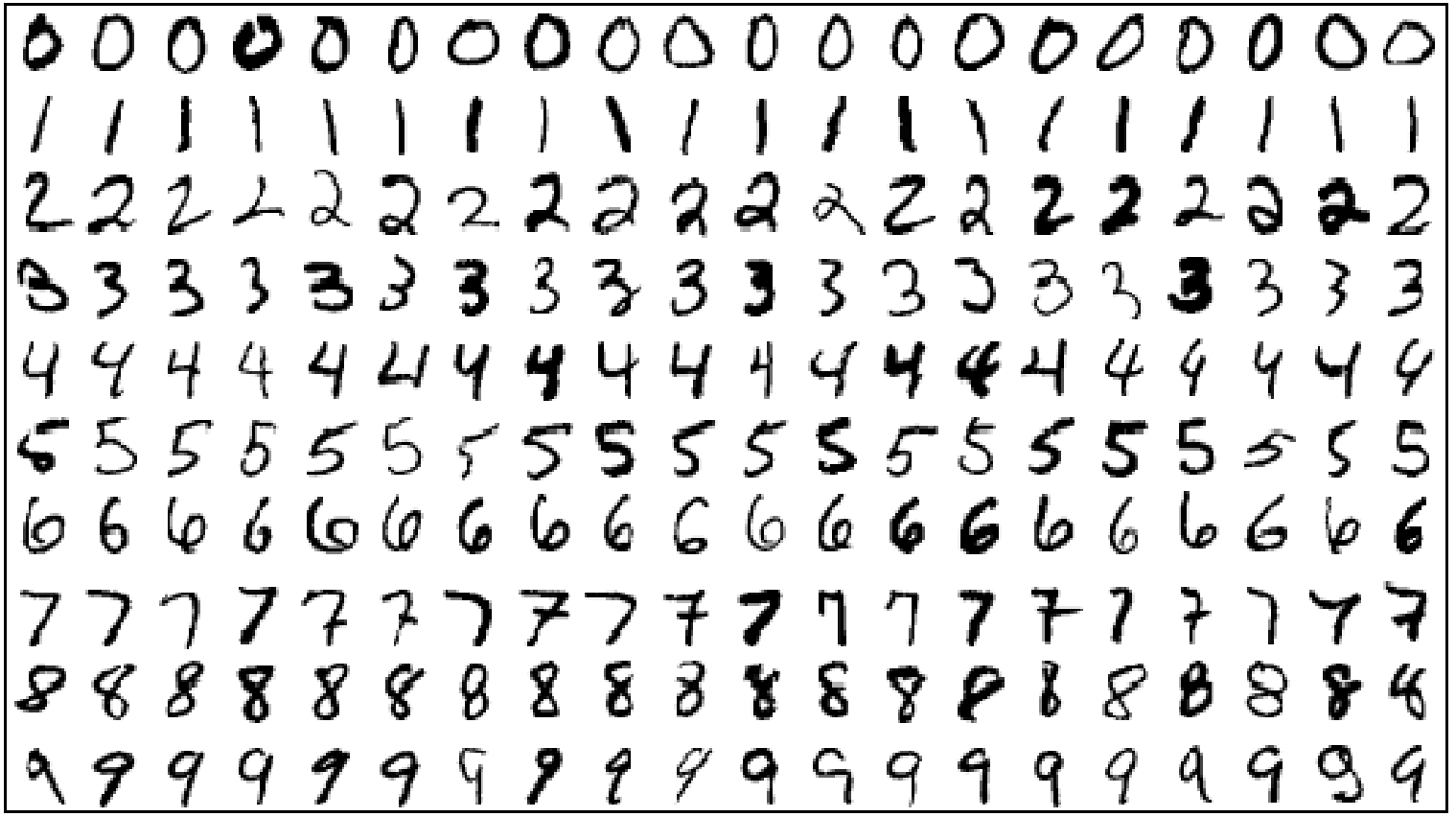
Multi-Layer Perceptrons

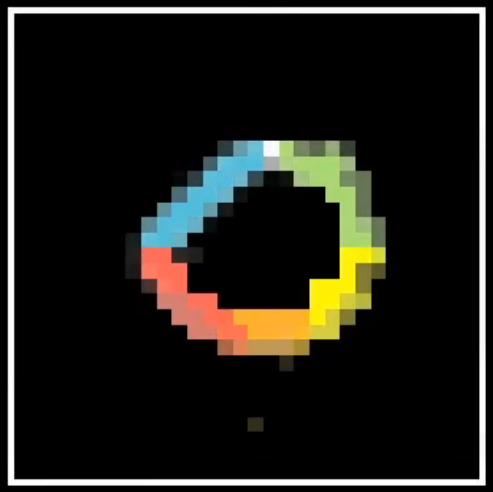
Let's gradually build up abstract features
by applying a classifier to the
outputs of a lower-level classifier!

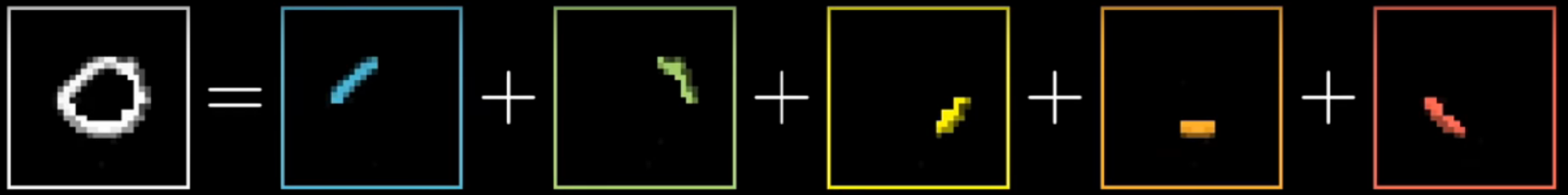
MNIST

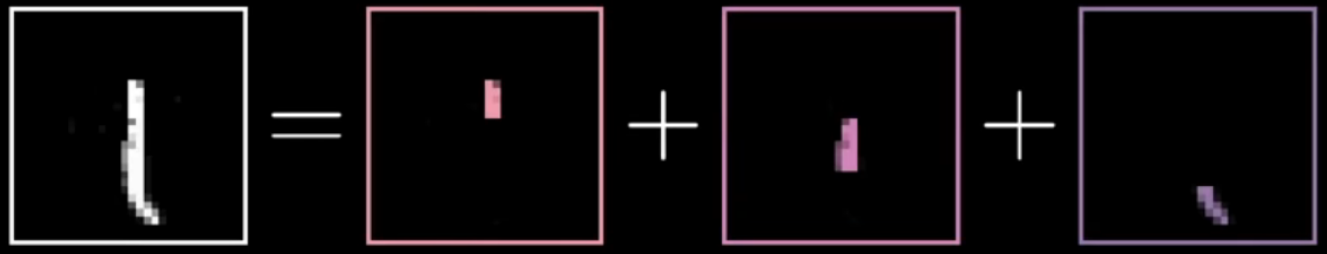
(Modified National Institute of Standards and Technology)

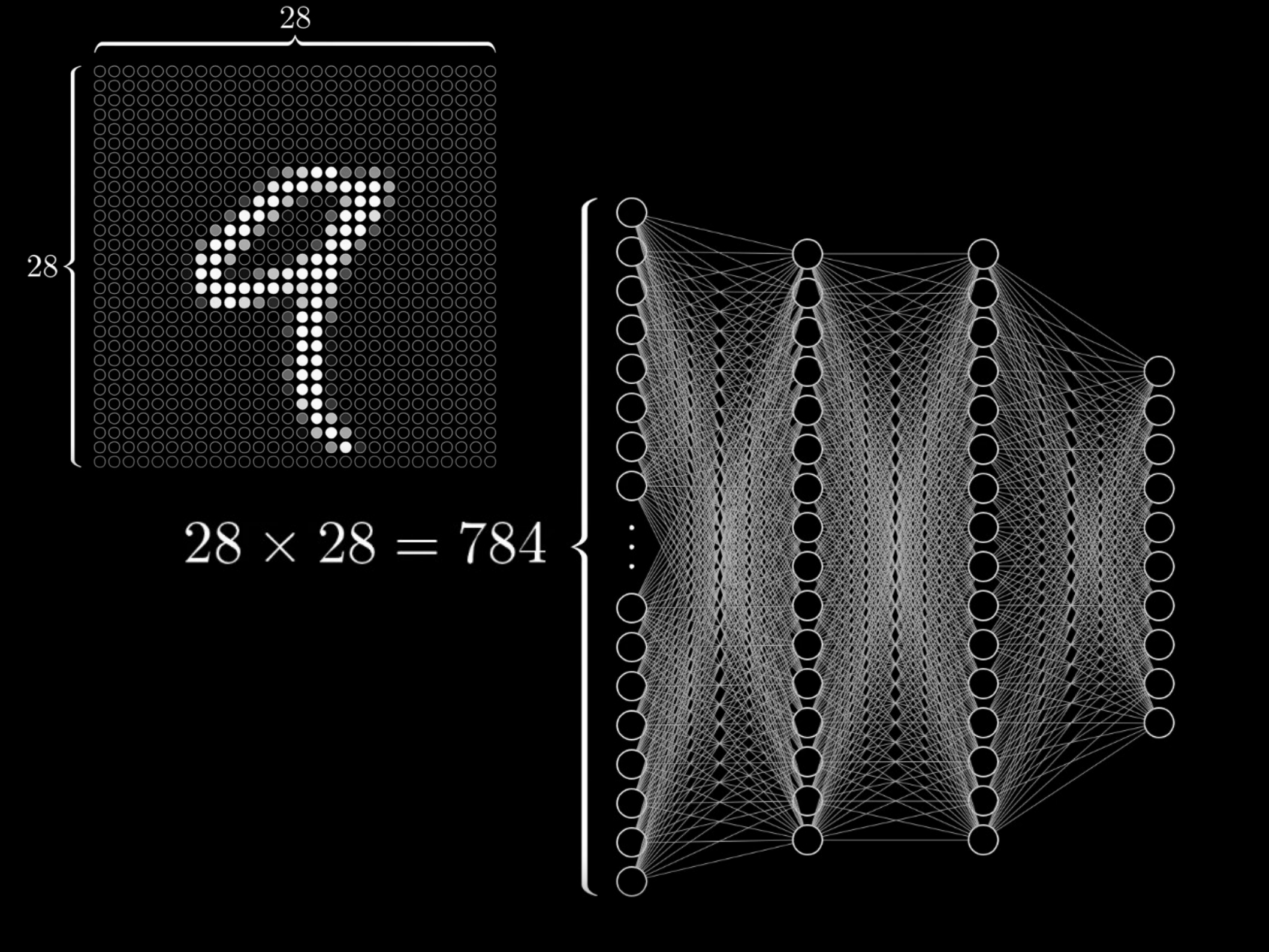
60,000 training images, 10,000 testing images

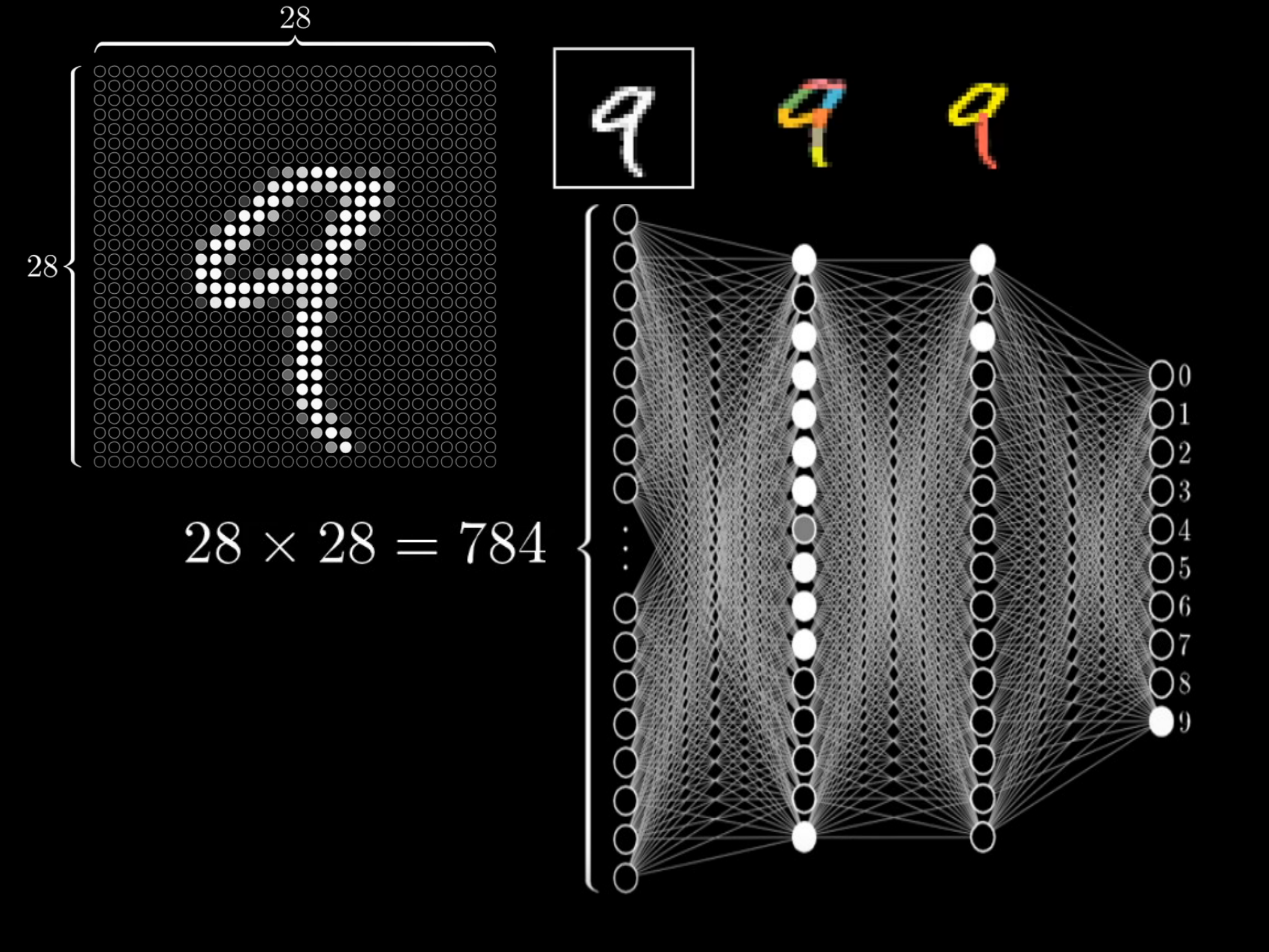


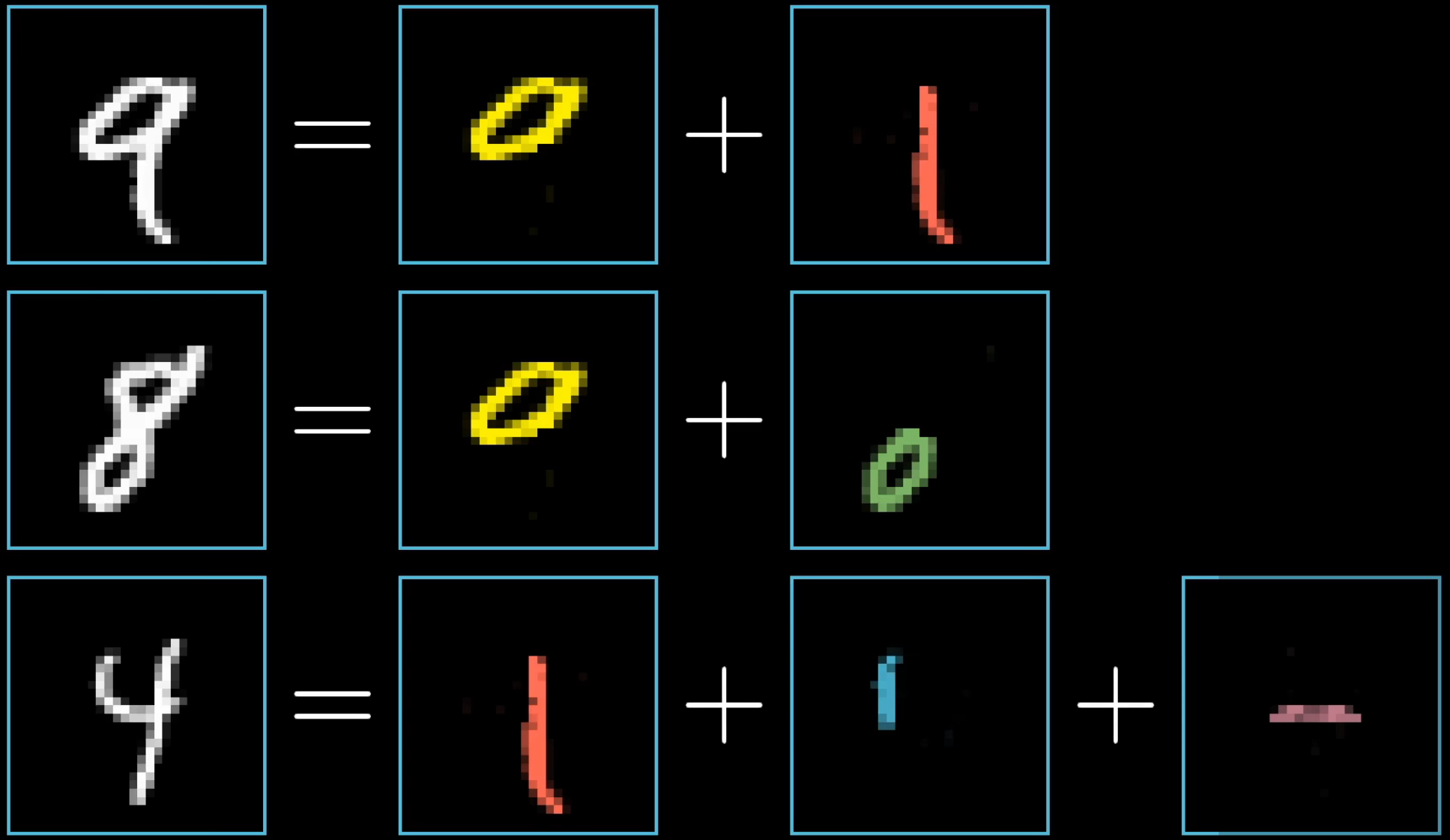


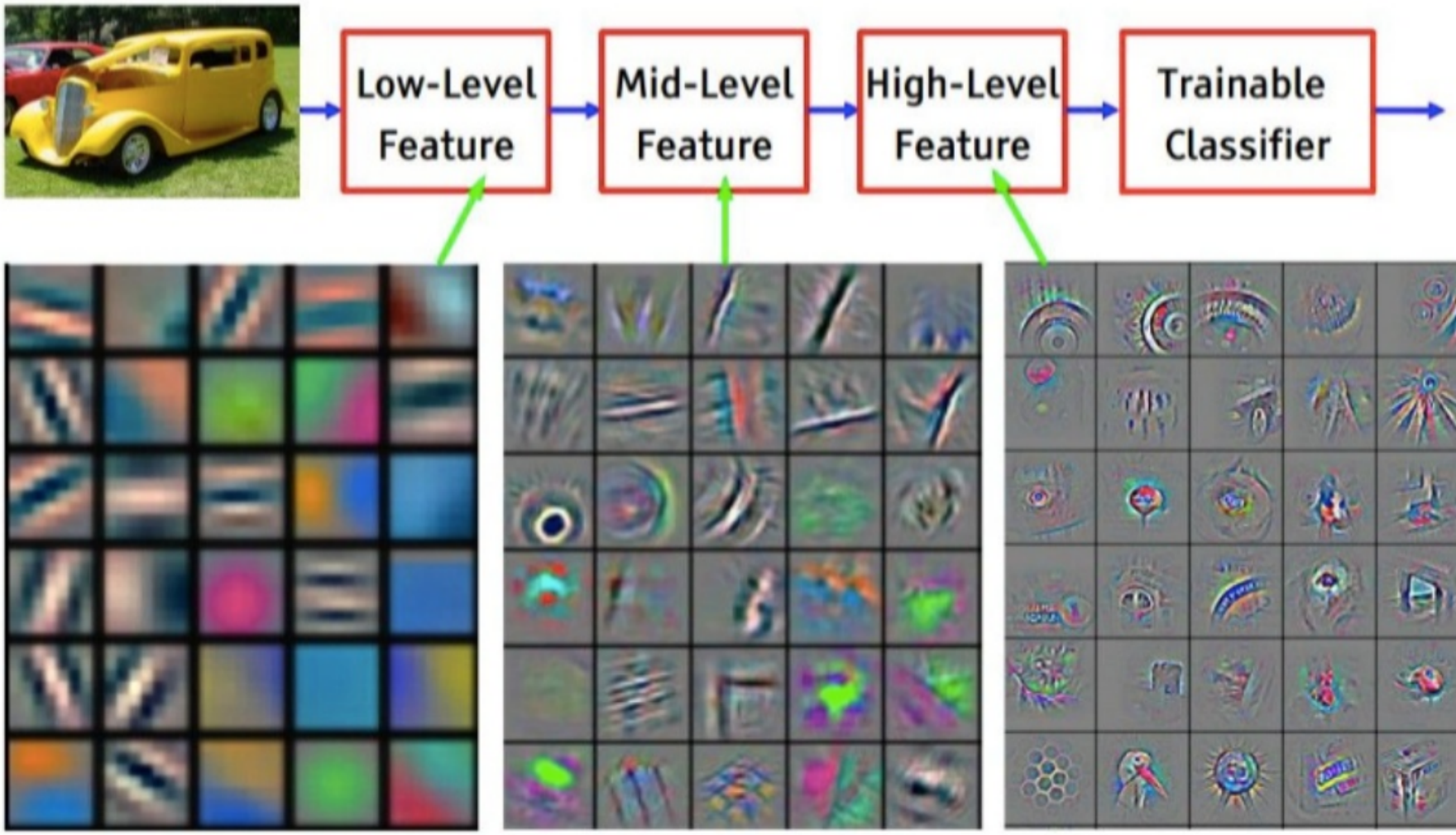




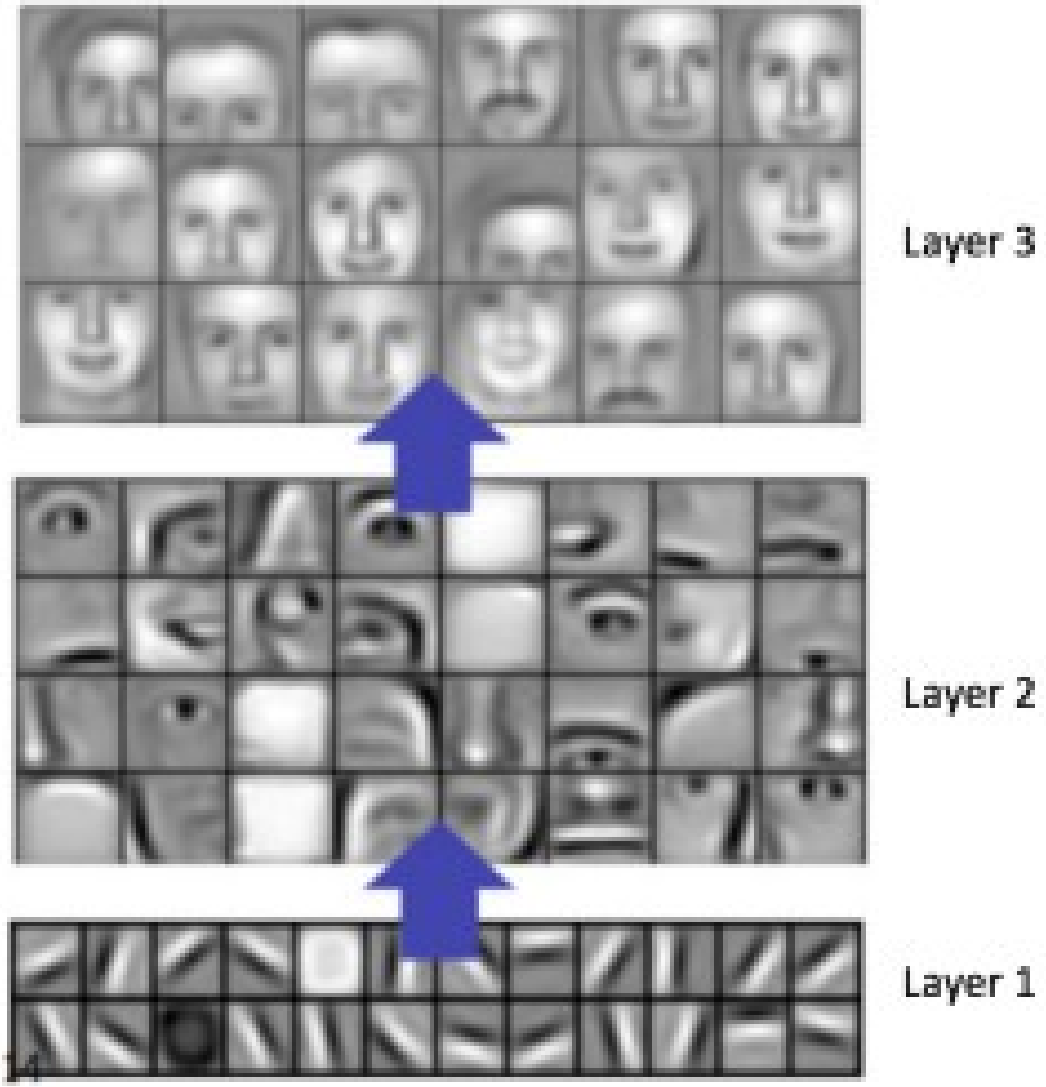

28
28
$28 \times 28 = 784$


28
28
$28 \times 28 = 784$
0
1
2
3
4
5
6
7
8
9





Feature visualization of convolutional net trained on ImageNet from [Zeiler & Fergus 2013]

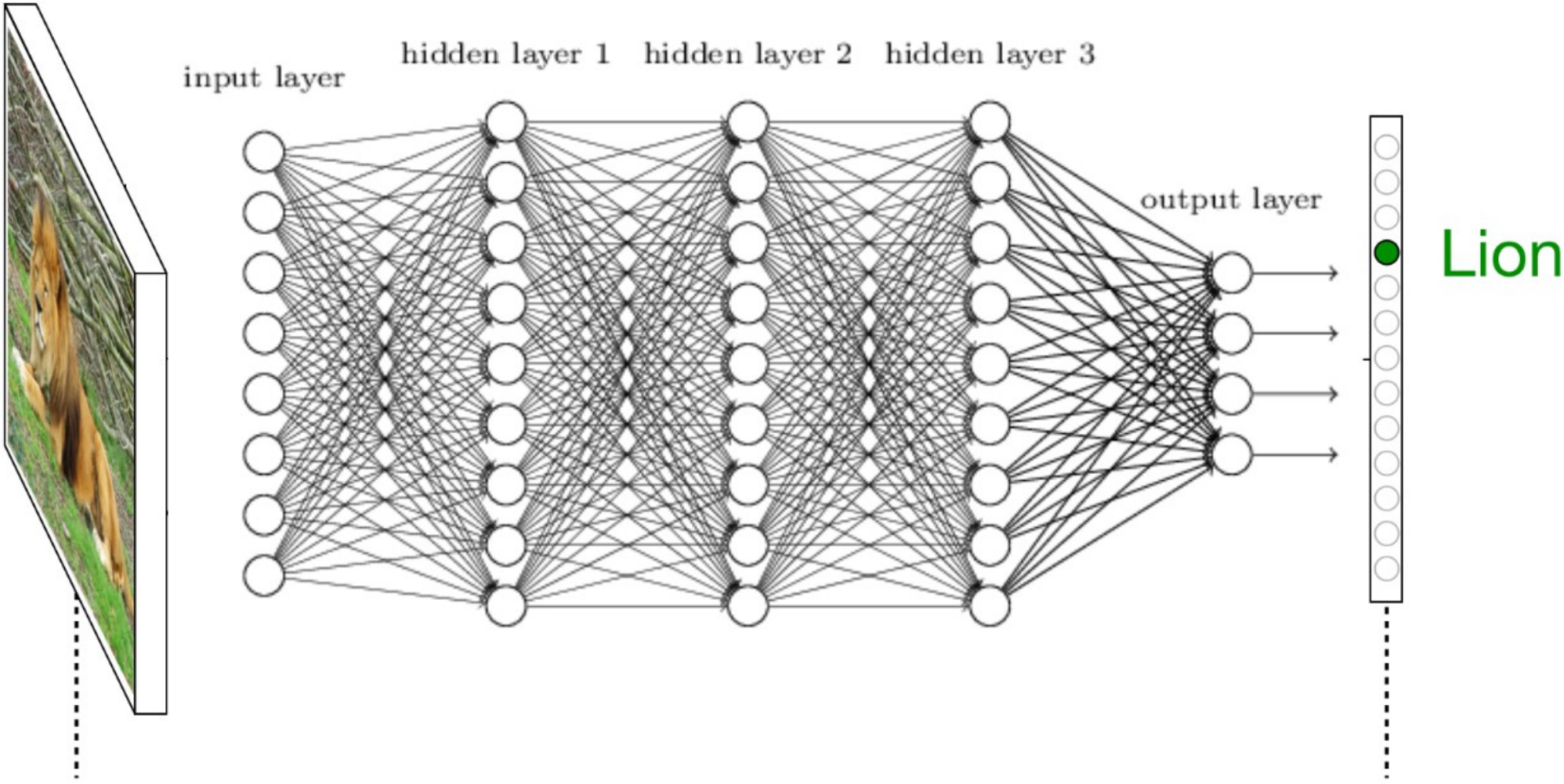

Layer 3
Layer 2
Layer 1

Supervised Learning

Given (x, y) pairs from the function y = *f*(x), find *f*

→ 9

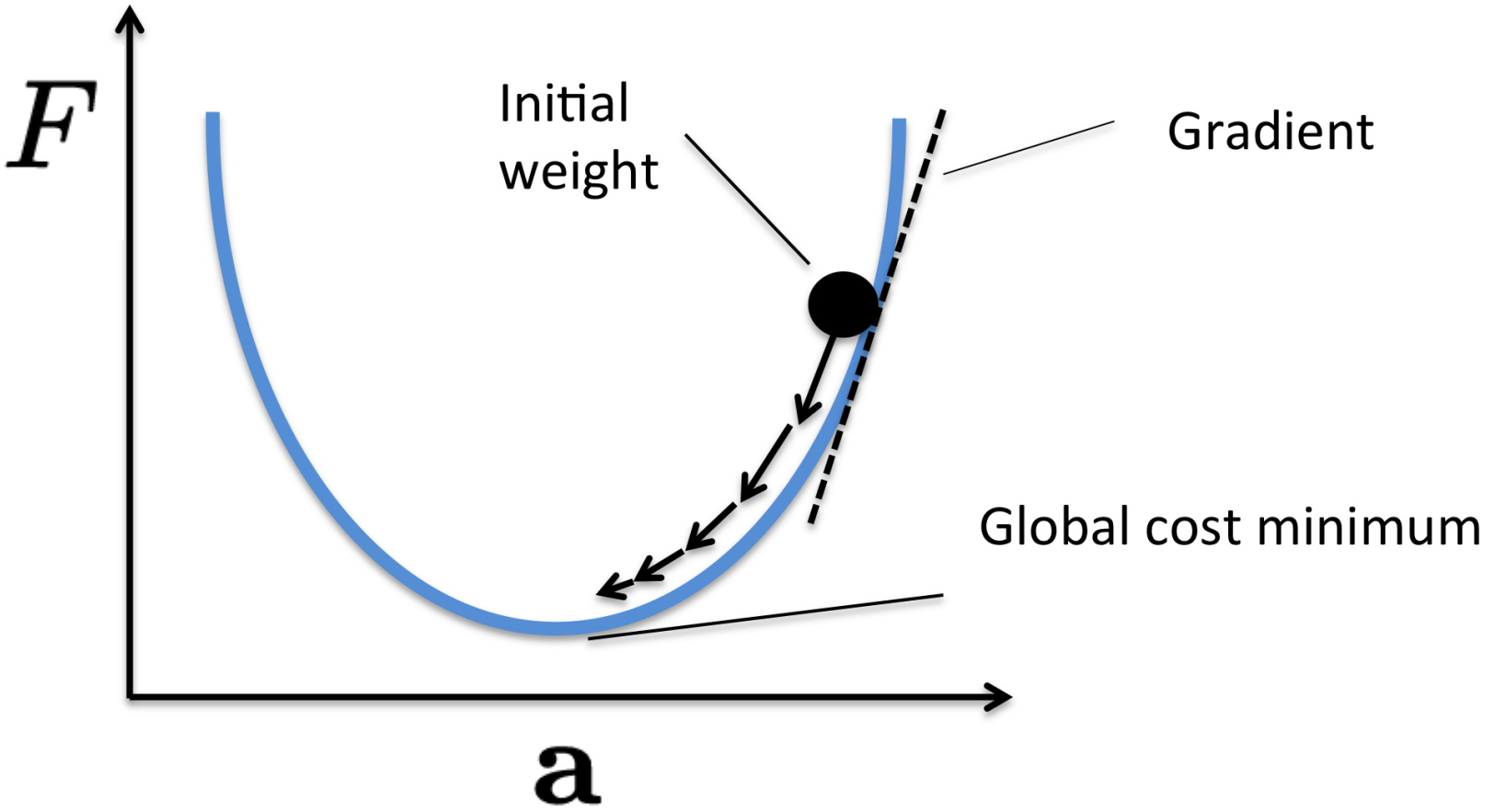
→ lion


input layer
hidden layer 1
hidden layer 2
hidden layer 3
output layer
Lion
Image
labels

$$Cost\ (W) = RSS(W) = \sum_{i=1}^{N} \{y_i - \hat{y}_i\}^2$$

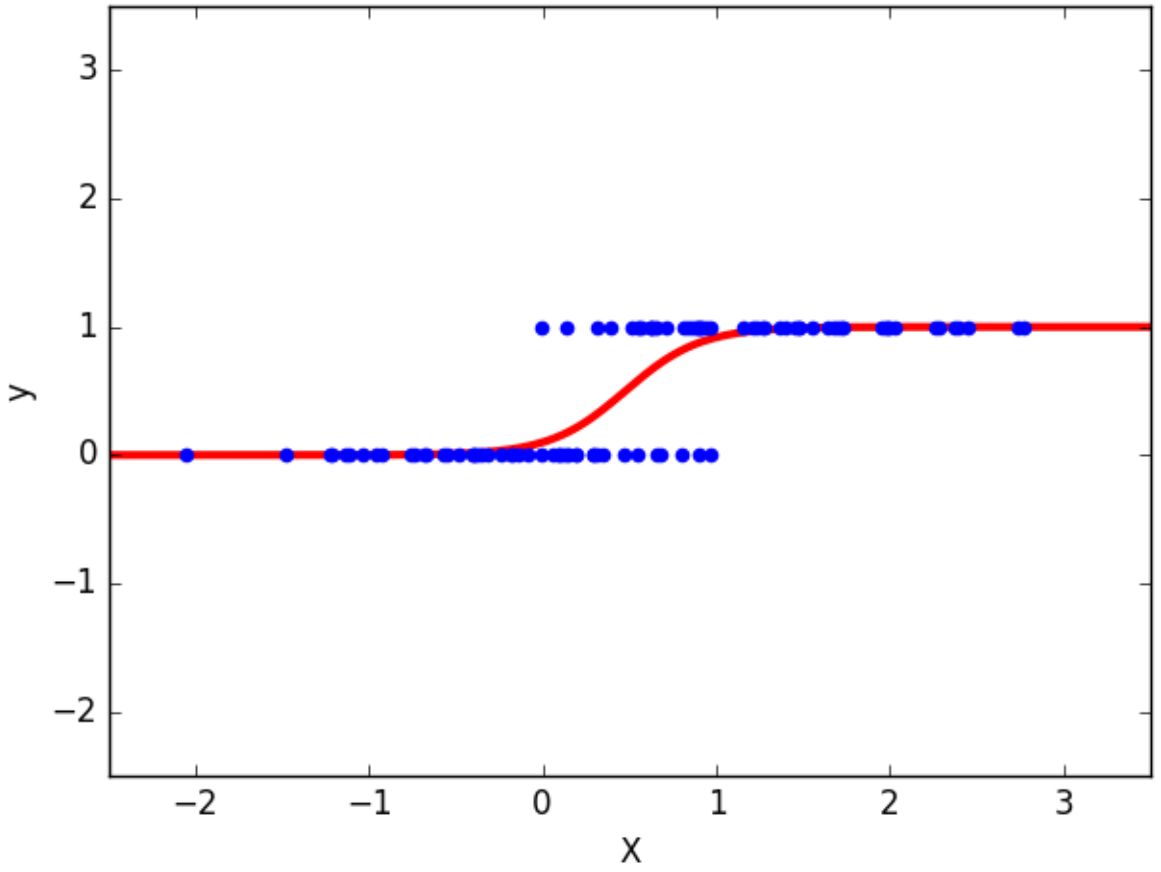
Gradient Descent

$$\mathbf{a}_{n+1} = \mathbf{a}_n - \gamma \nabla F(\mathbf{a}_n)$$

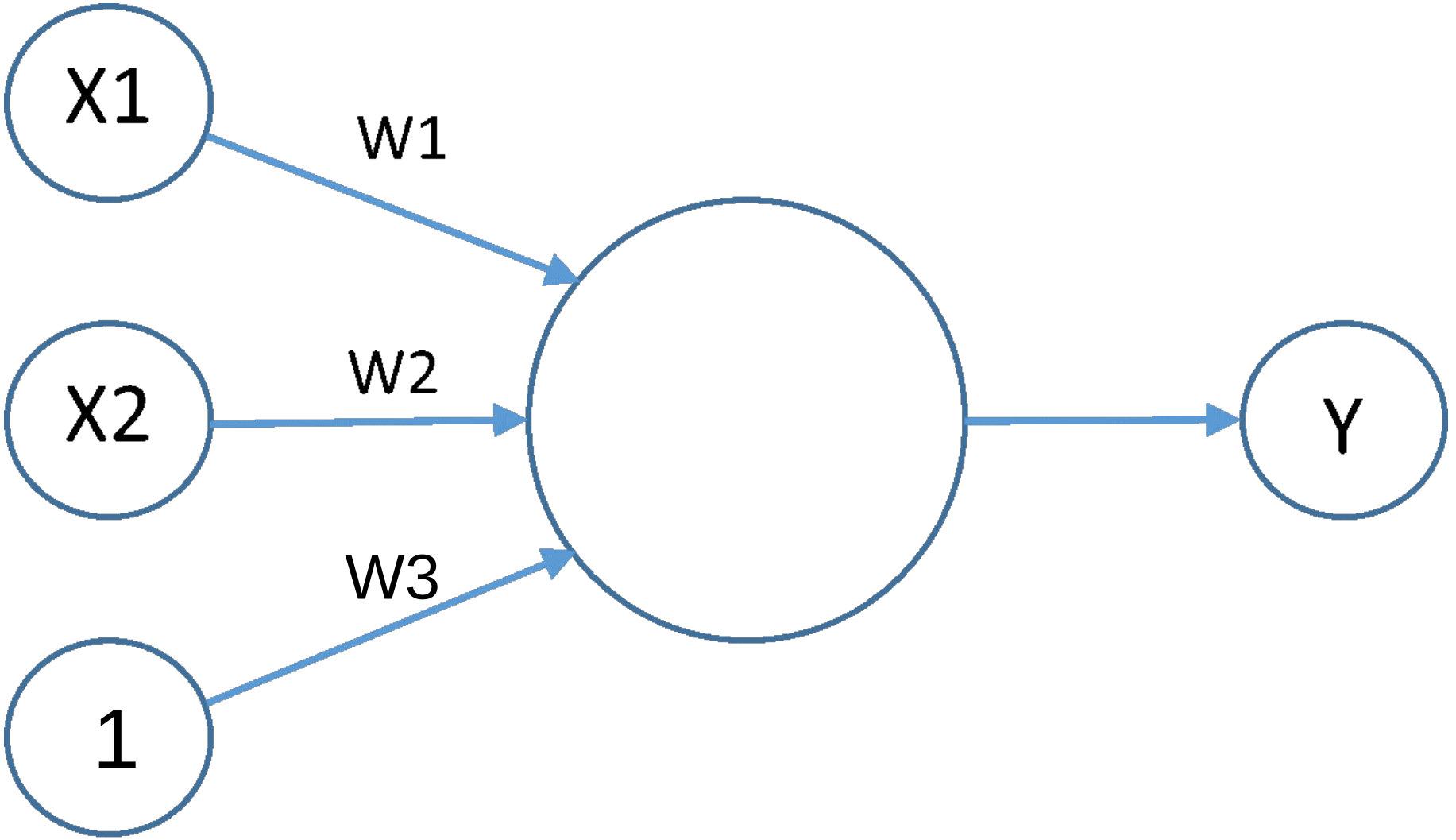


Backpropogation

$$y = \sigma(w * x + b)$$



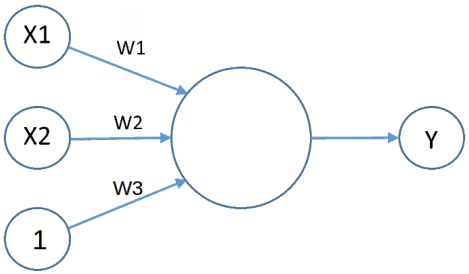
$$y = \sigma(w_1 * x_1 + w_2 * x_2 + w_3 * 1)$$



$$y = \sigma(w_1 * x_1 + w_2 * x_2 + w_3 * 1)$$

$$y = \sigma(z)$$

$$z = w_1 * x_1 + w_2 * x_2 + w_3 * 1$$



$$\frac{\partial y}{\partial w_1} = \frac{\partial y}{\partial z} * \frac{\partial z}{\partial w_1}$$

$$\frac{\partial y}{\partial z} = \sigma(z) * (1 - \sigma(z))$$

$$\frac{\partial z}{\partial w_1} = x_1$$

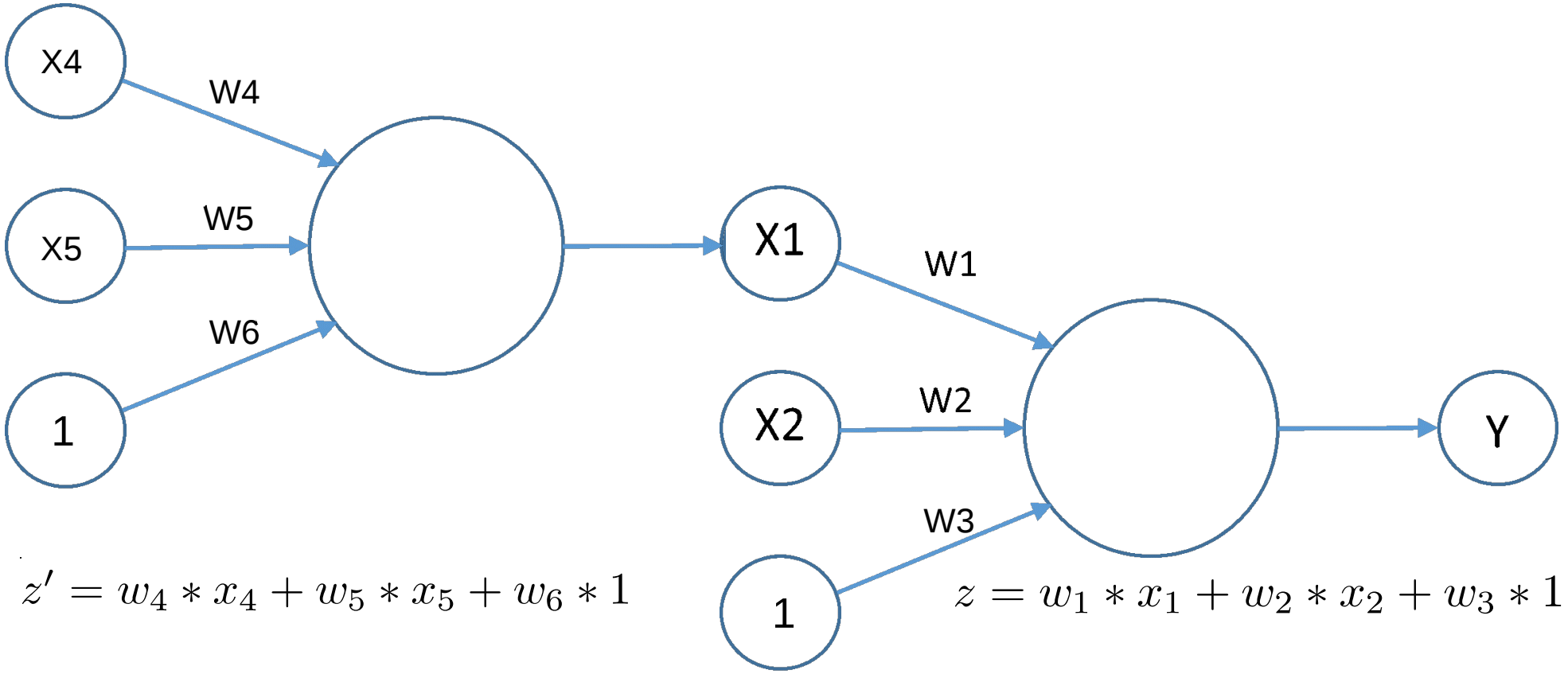
$$\frac{\partial y}{\partial w_1} = \frac{\partial y}{\partial z} * \frac{\partial z}{\partial w_1} = \sigma(z) * (1 - \sigma(z)) * x_1$$

$$\frac{\partial y}{\partial w_2} = \sigma(z) * (1 - \sigma(z)) * x_2$$

$$\frac{\partial y}{\partial w_4} = \frac{\partial y}{\partial x_1} * \frac{\partial x_1}{\partial w_4}$$

$$\frac{\partial y}{\partial x_1} = \sigma(z) * (1 - \sigma(z)) * w_1$$

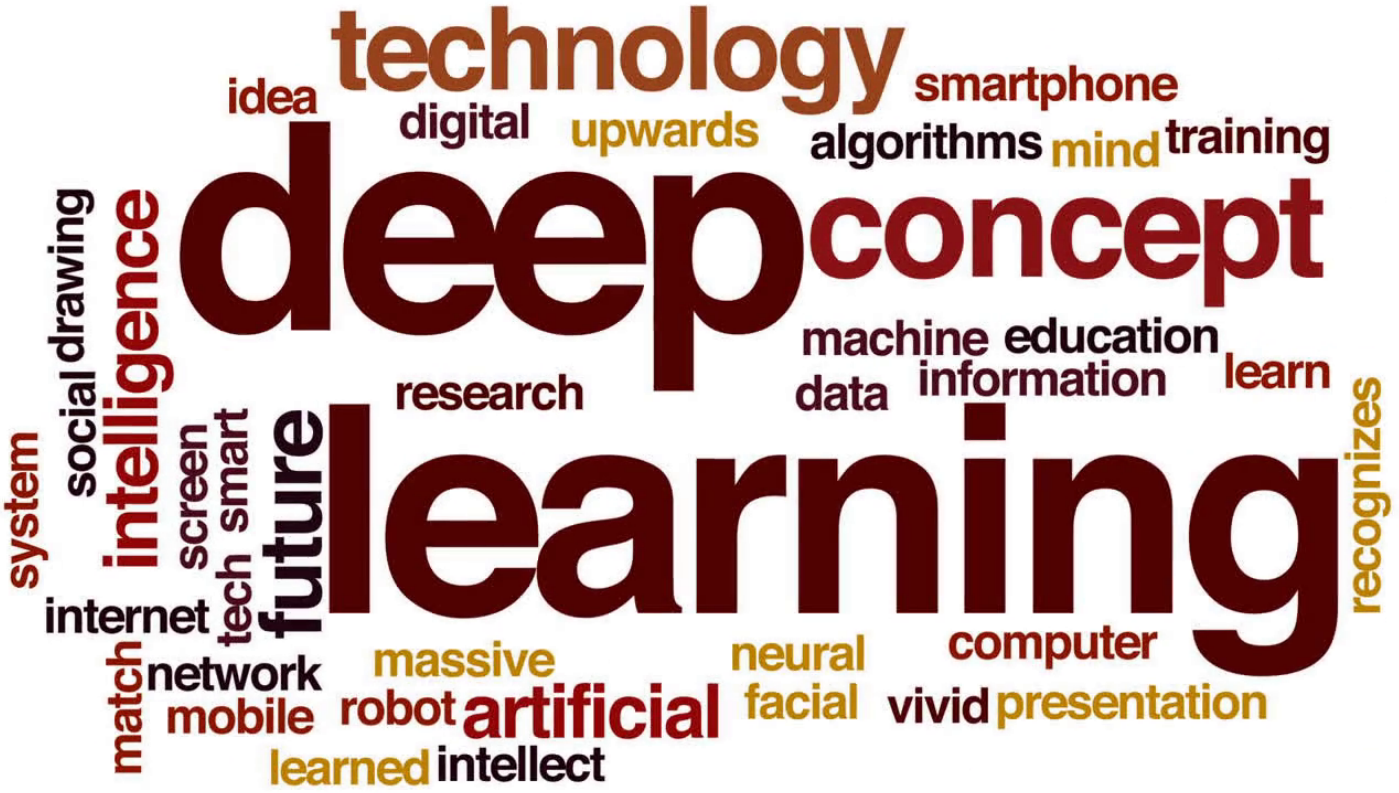
$$\frac{\partial x_1}{\partial w_4} = \sigma(z') * (1 - \sigma(z')) * x_4$$



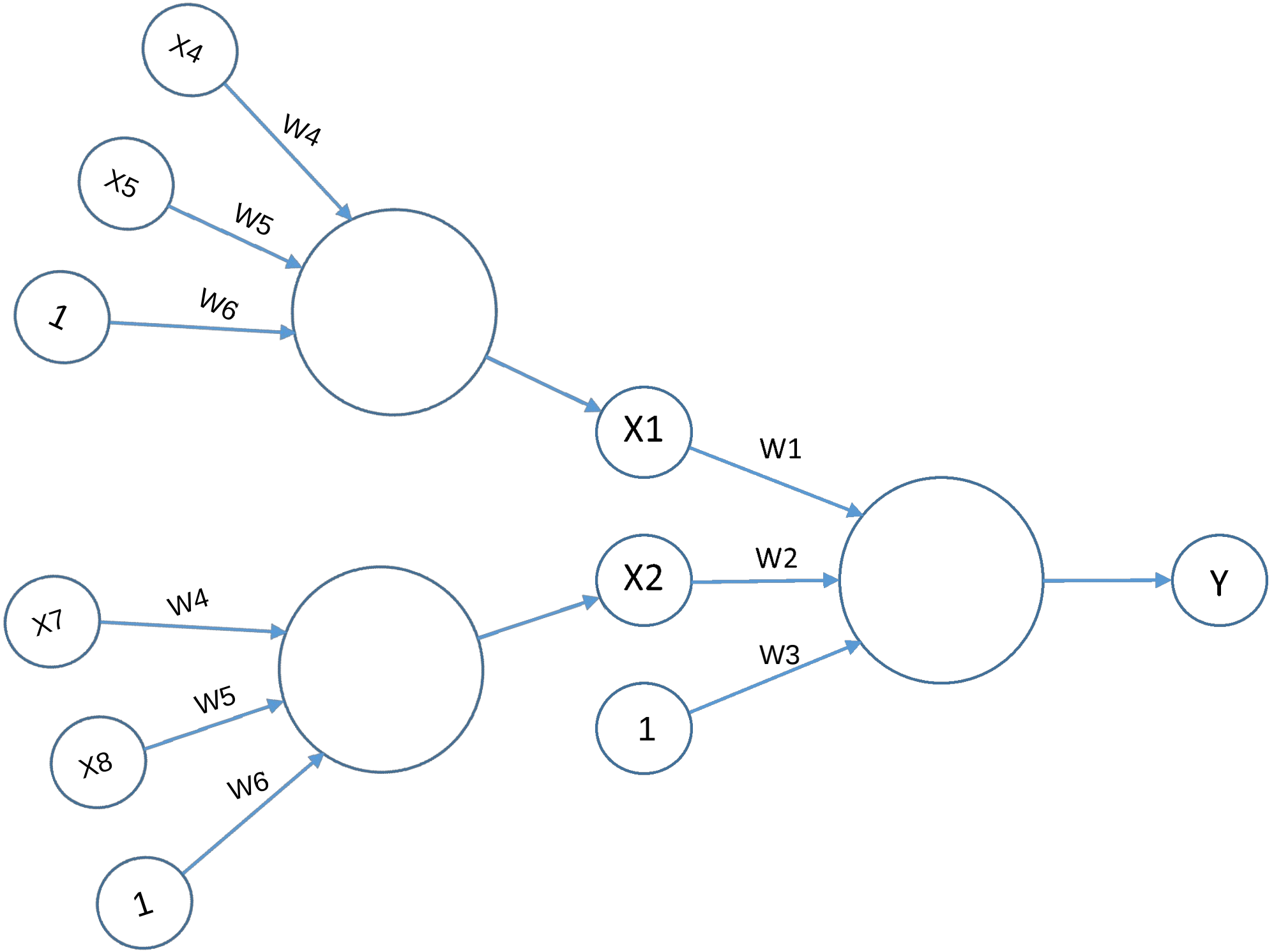
$$z' = w_4 * x_4 + w_5 * x_5 + w_6 * 1$$

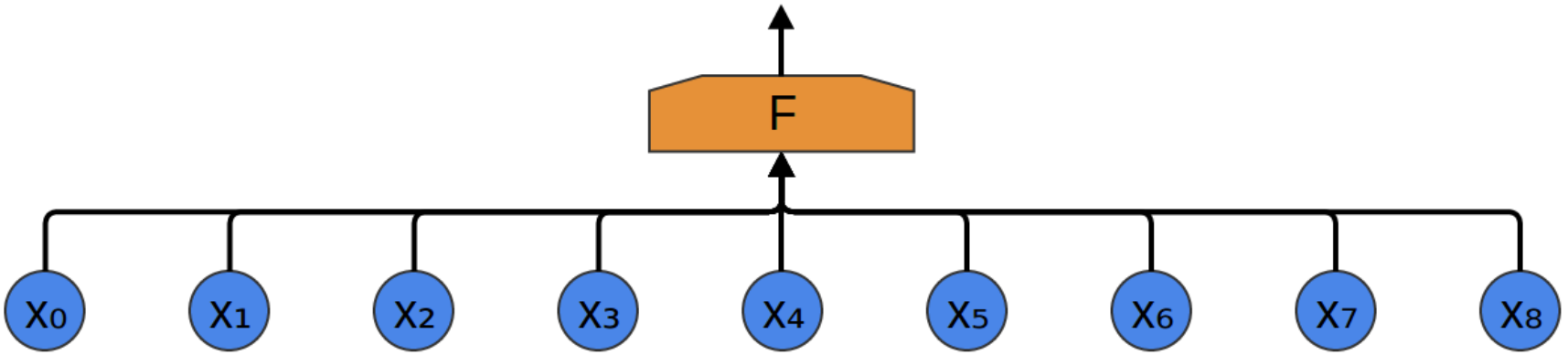
$$z = w_1 * x_1 + w_2 * x_2 + w_3 * 1$$

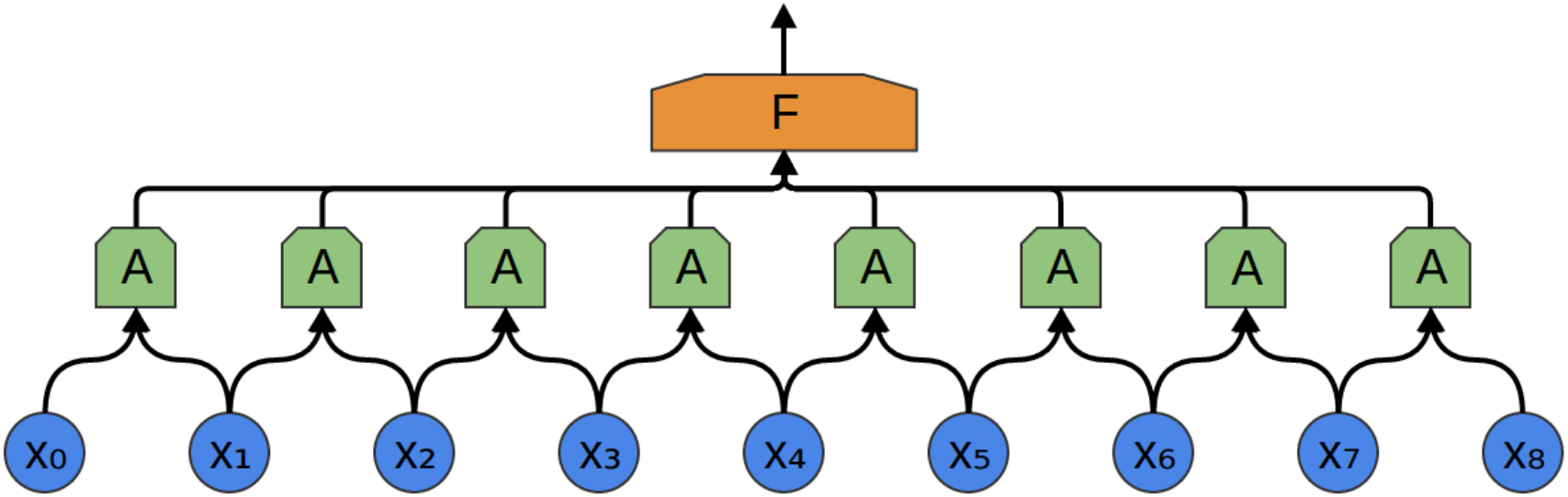
Tricks that make this work/scale!

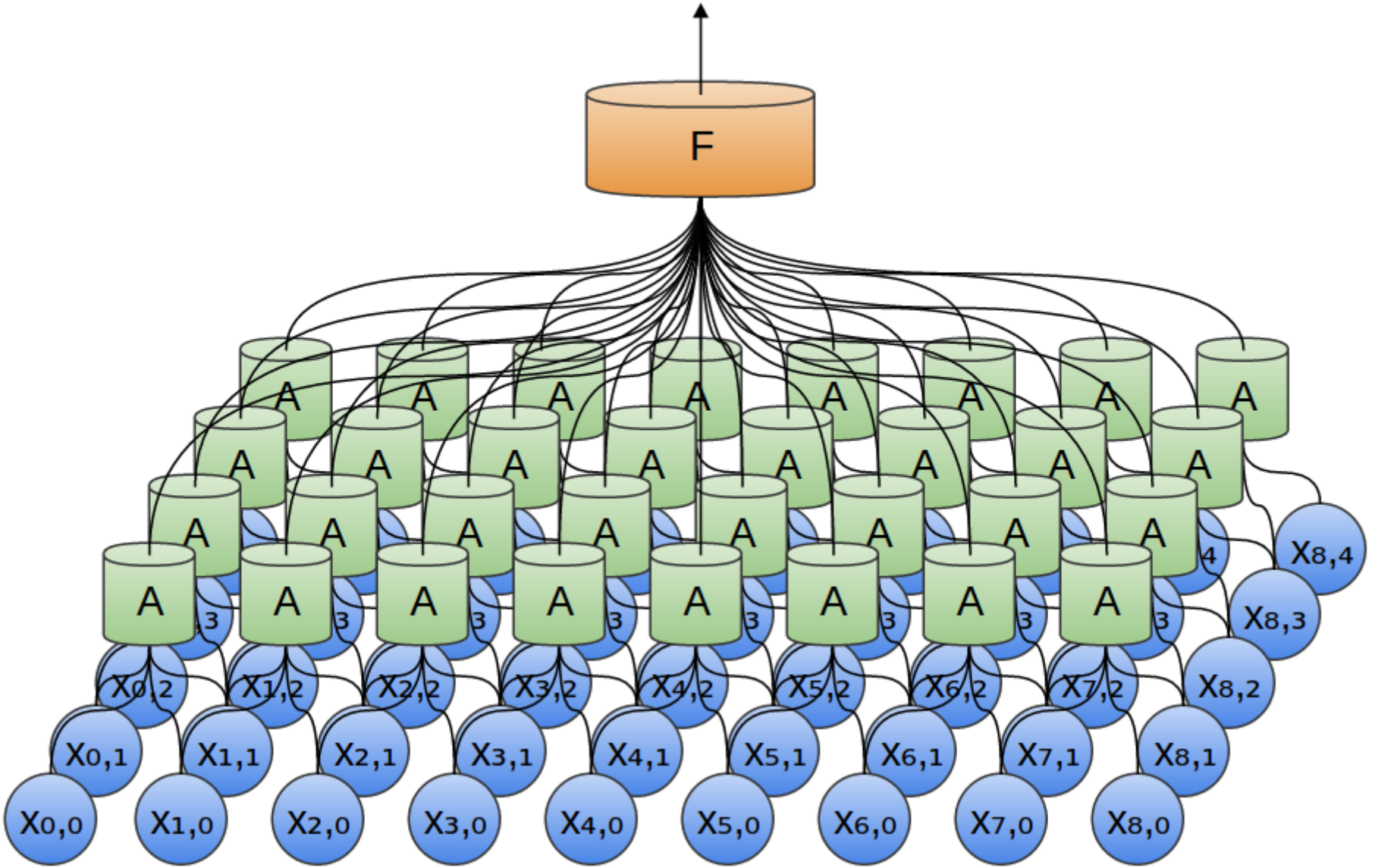

technology
idea
digital
upwards
smartphone
algorithms
mind
training
deep
concept
drawing
social
intelligence
machine
education
research
data
information
learn
system
screen
smart
future
learning
recognizes
internet
tech
match
network
massive
neural
computer
mobile
robot
artificial
facial
vivid
presentation
learned
intellect

Weight Sharing (via convolution)

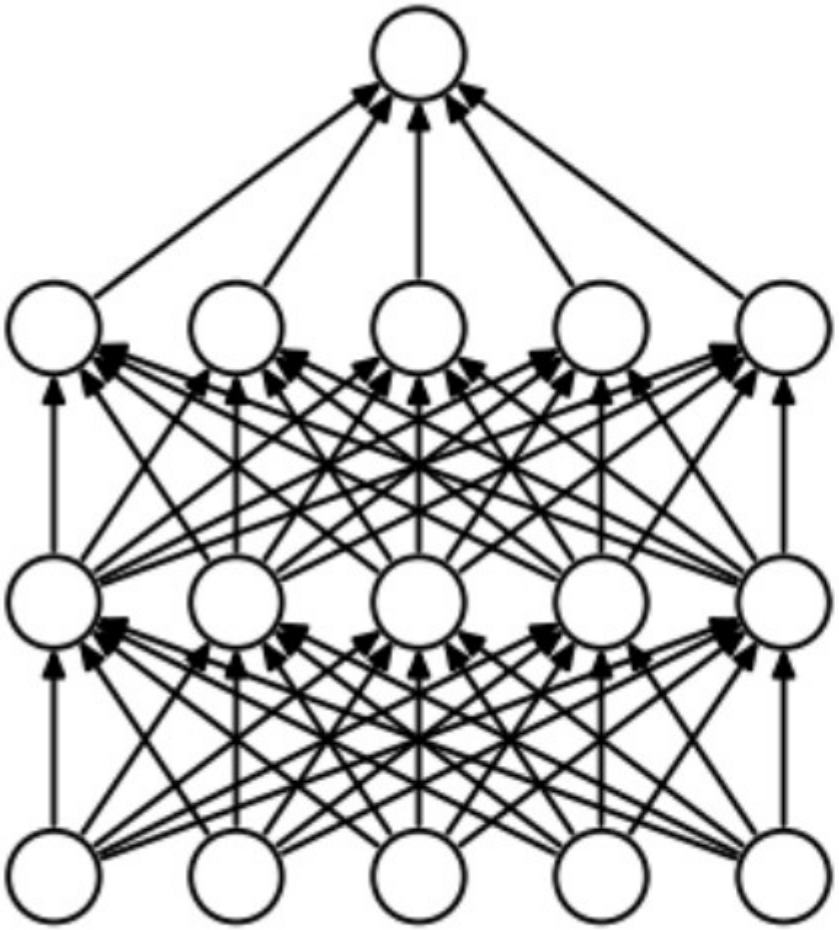

X4
W4
X5
W5
1
W6
X1
W1
X7
W4
X2
W2
Y
W5
X8
W3
1
W6
1


F
X0
X1
X2
X3
X4
X5
X6
X7
X8

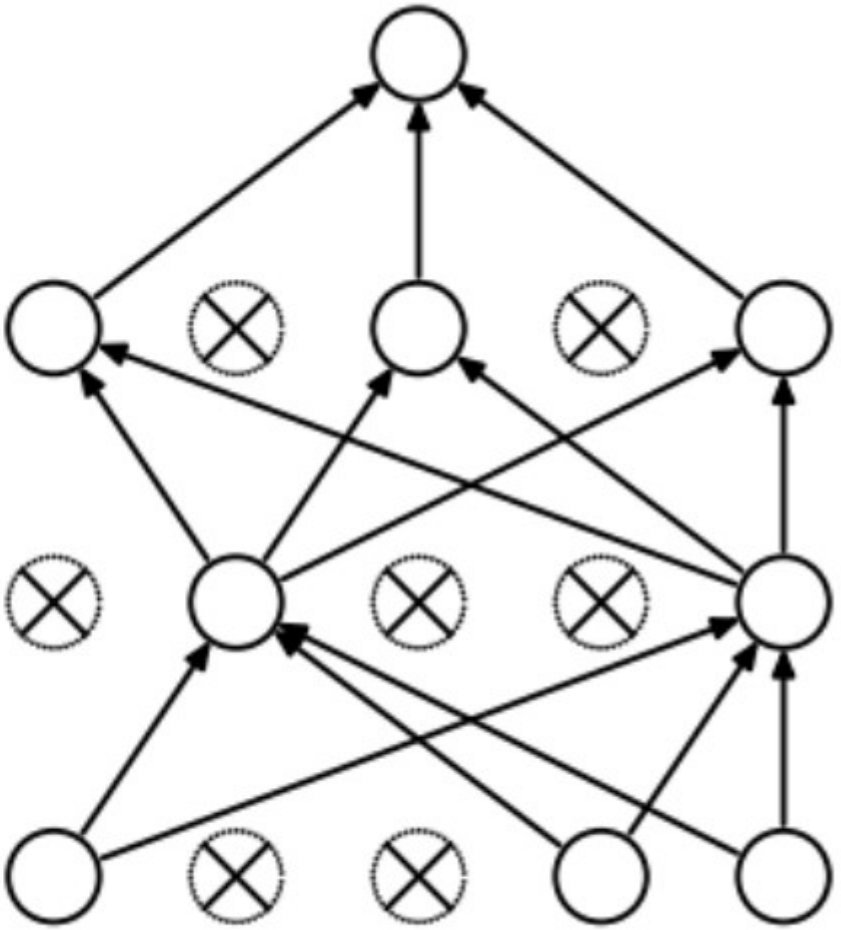

F
A
A
A
A
A
A
A
A
X0
X1
X2
X3
X4
X5
X6
X7
X8


F
A
X0,0
X1,0
X2,0
X3,0
X4,0
X5,0
X6,0
X7,0
X8,0
X0,1
X1,1
X2,1
X3,1
X4,1
X5,1
X6,1
X7,1
X8,1
X0,2
X1,2
X2,2
X3,2
X4,2
X5,2
X6,2
X7,2
X8,2
X8,3
X8,4

Dropout



(a) Standard Neural Net



(b) After applying dropout.