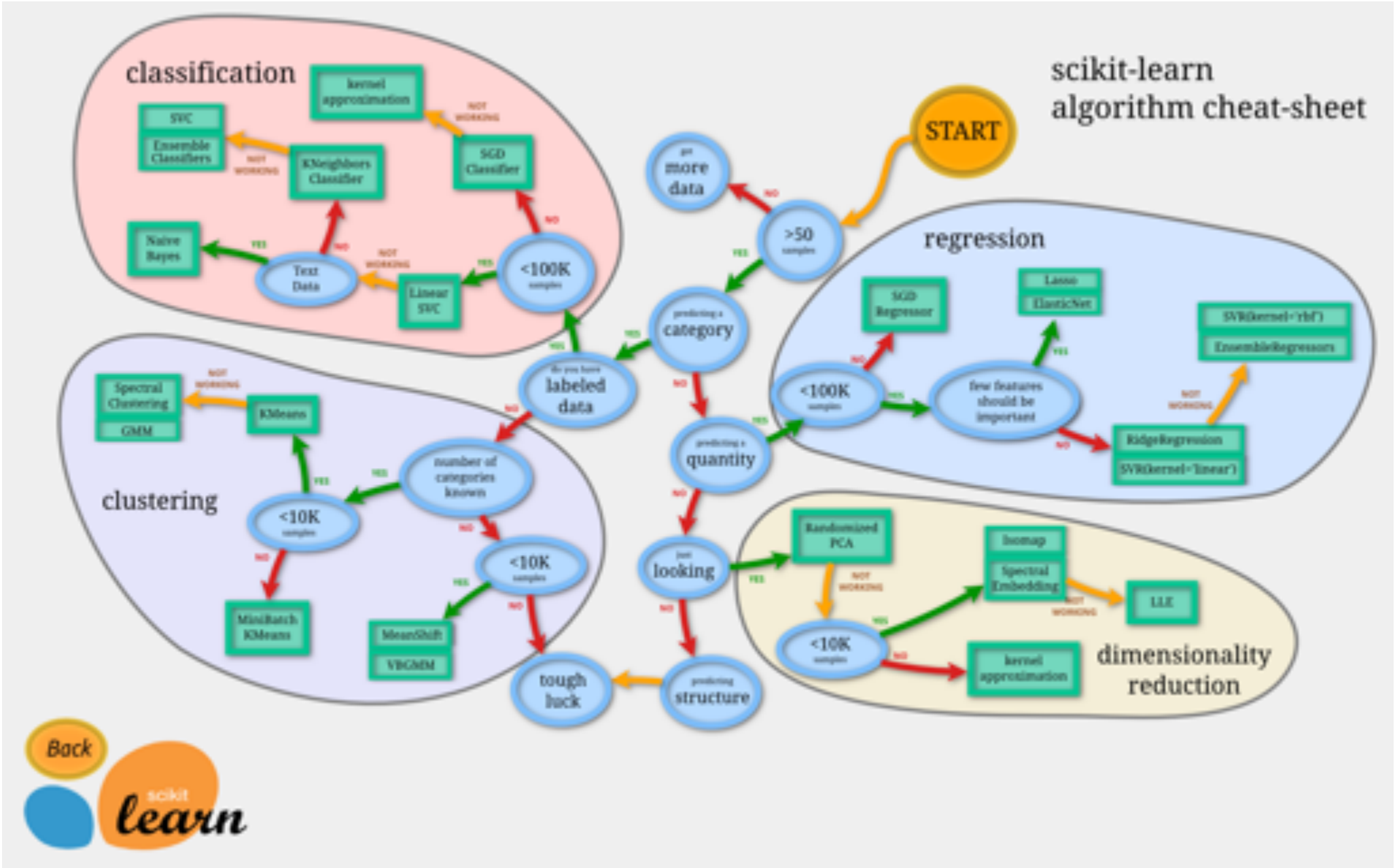
Classification

Qingkai Kong

University of California Berkeley Seismological Laboratory

http://seismo.berkeley.edu/qingkaikong/



Interactive version: http://scikit-learn.org/stable/tutorial/machine_learning_map/index.html

Pipeline of training a machine learning model.

Data examples

01100
10110
11110

Tunable Model

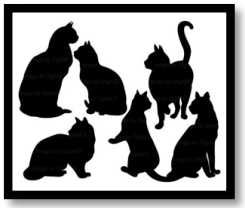
Optimization algorithm

Trained Model

Representation of data

Raw data

Documents



Images

Numbers

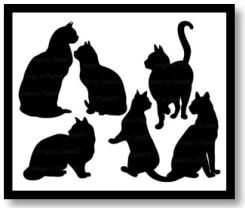
Sounds

●●●

Representation of data

Raw data

Documents



Images

Numbers

Sounds

●●●

Feature matrix (X)

n_features ⟶

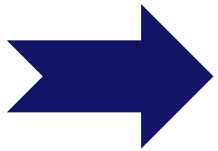
n_samples ↓

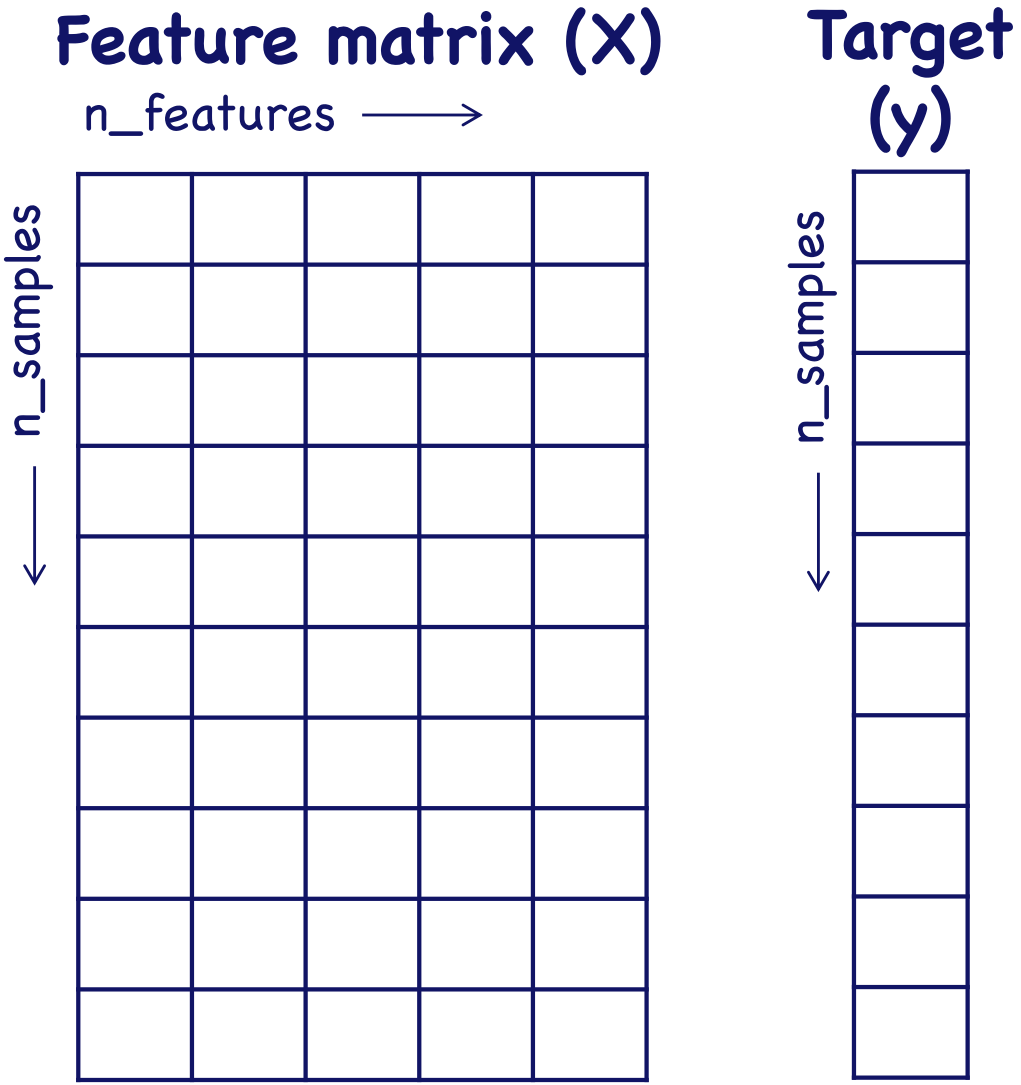
Target (y)

n_samples ↓

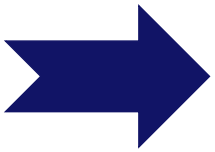
Example

Example





Example



Feature matrix (X)

n_features ⟶

n_samples ↓

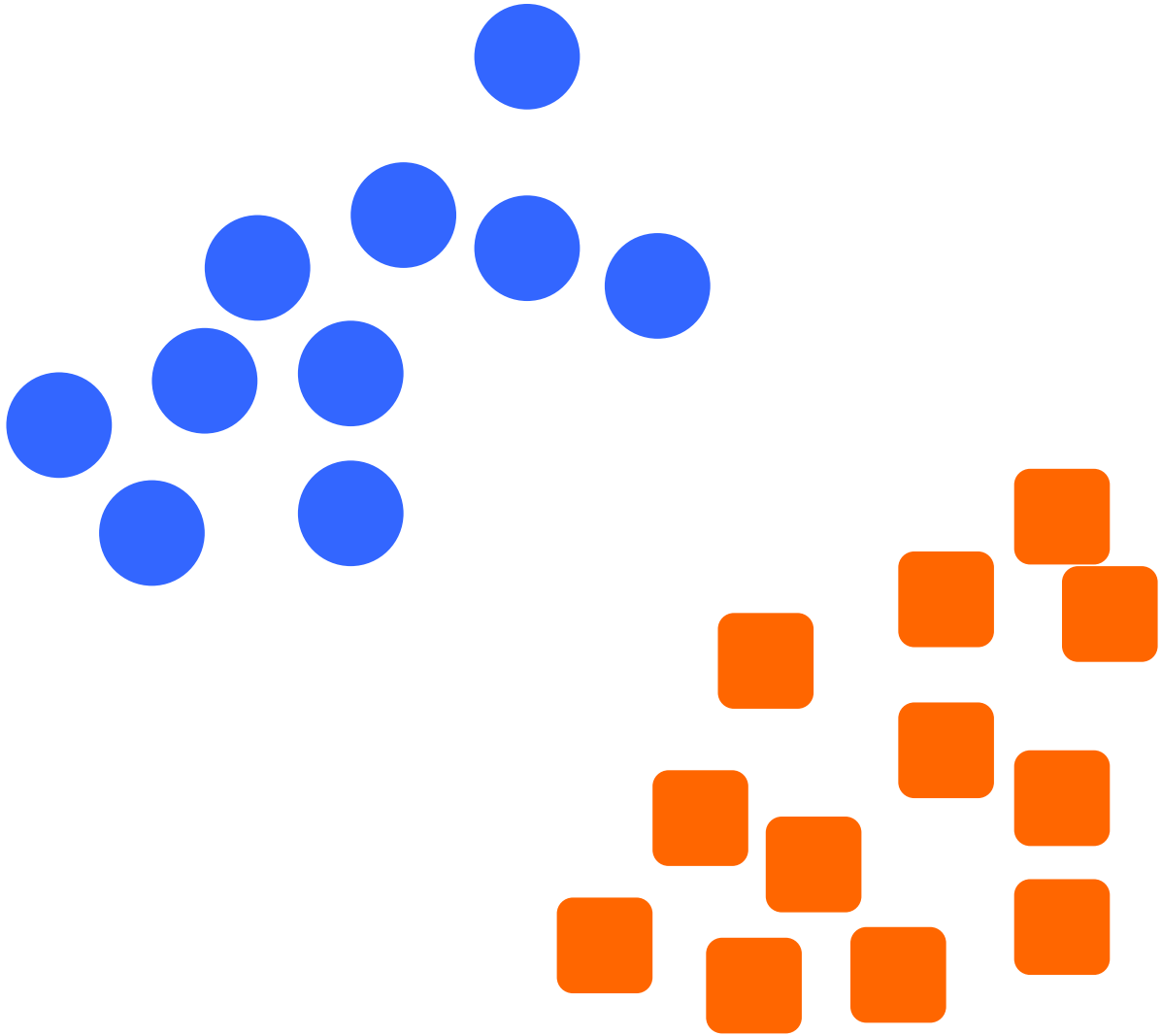
Color	Texture	Shape	Smell	Has core?

Target (y)

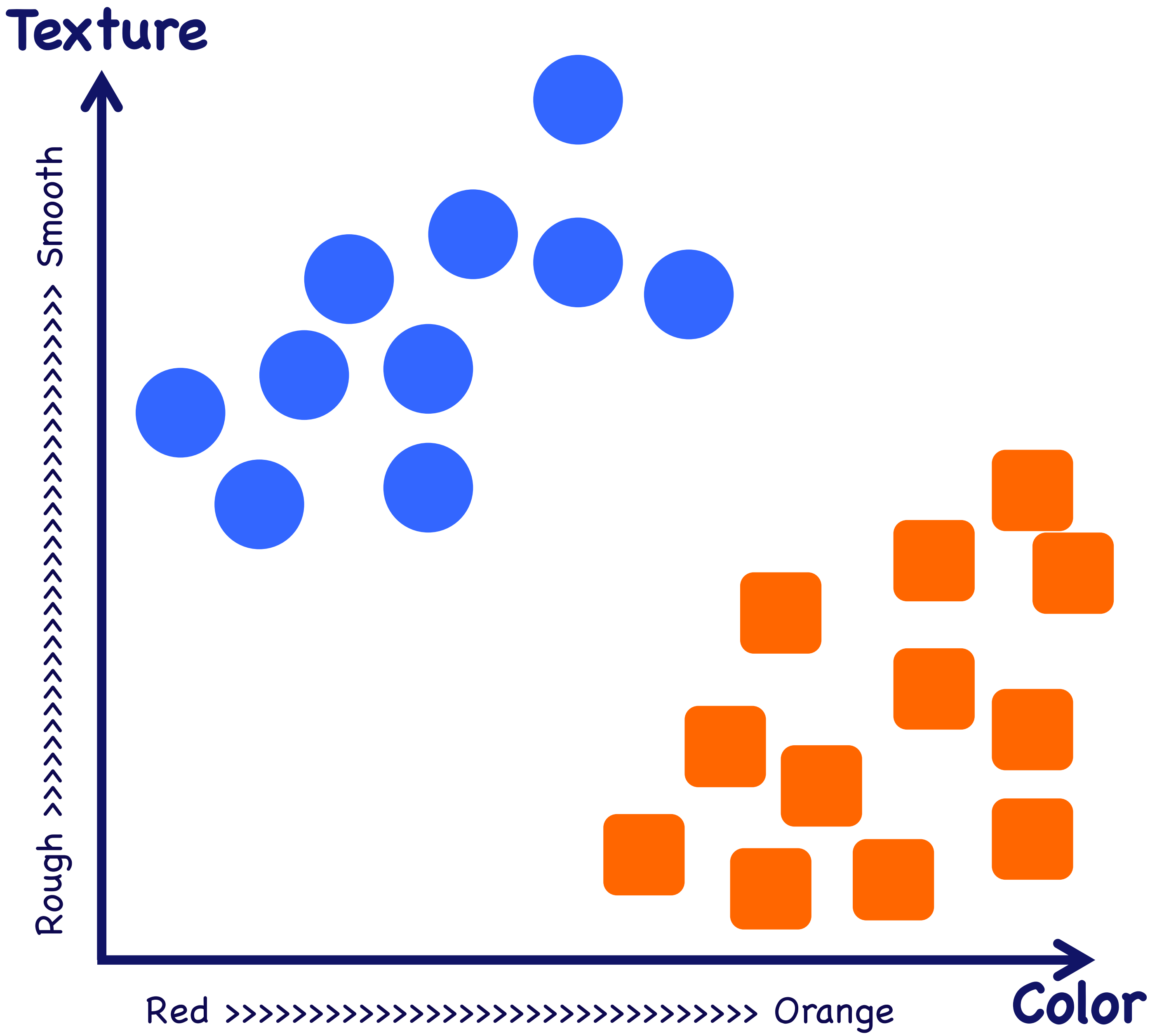
n_samples ↓

0
1
1
0
1
1
1
0
0
1

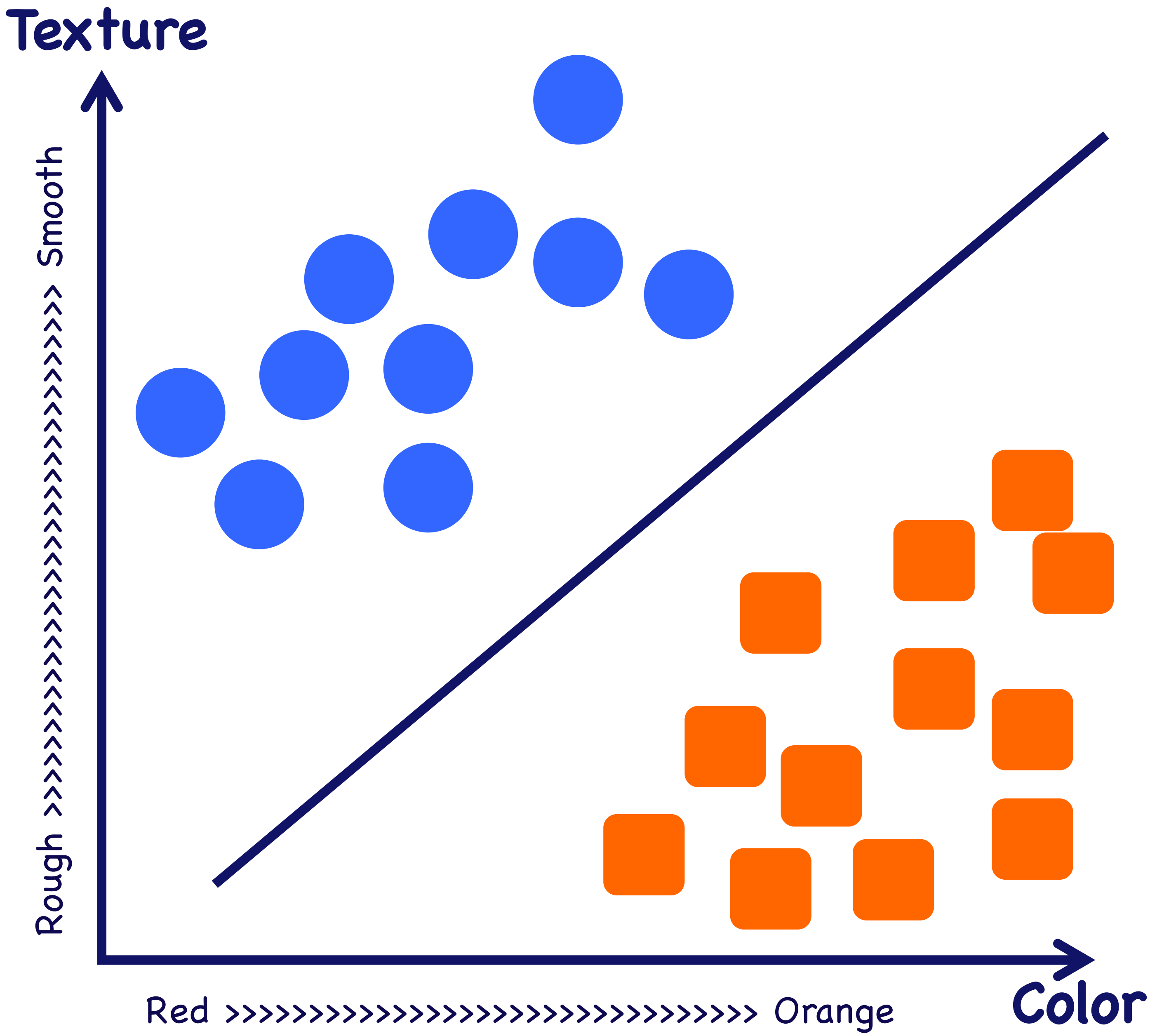
Classification



Classification

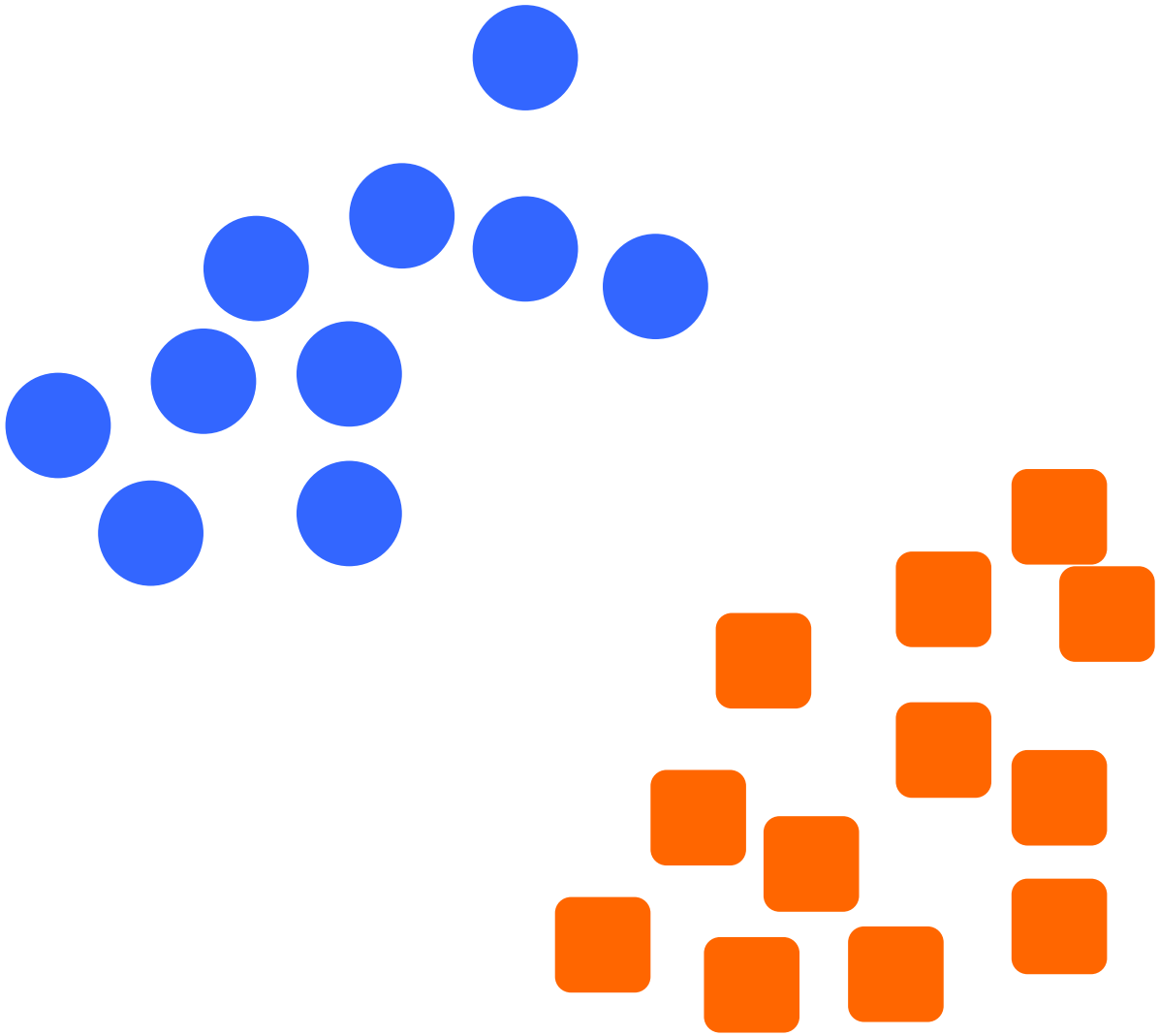


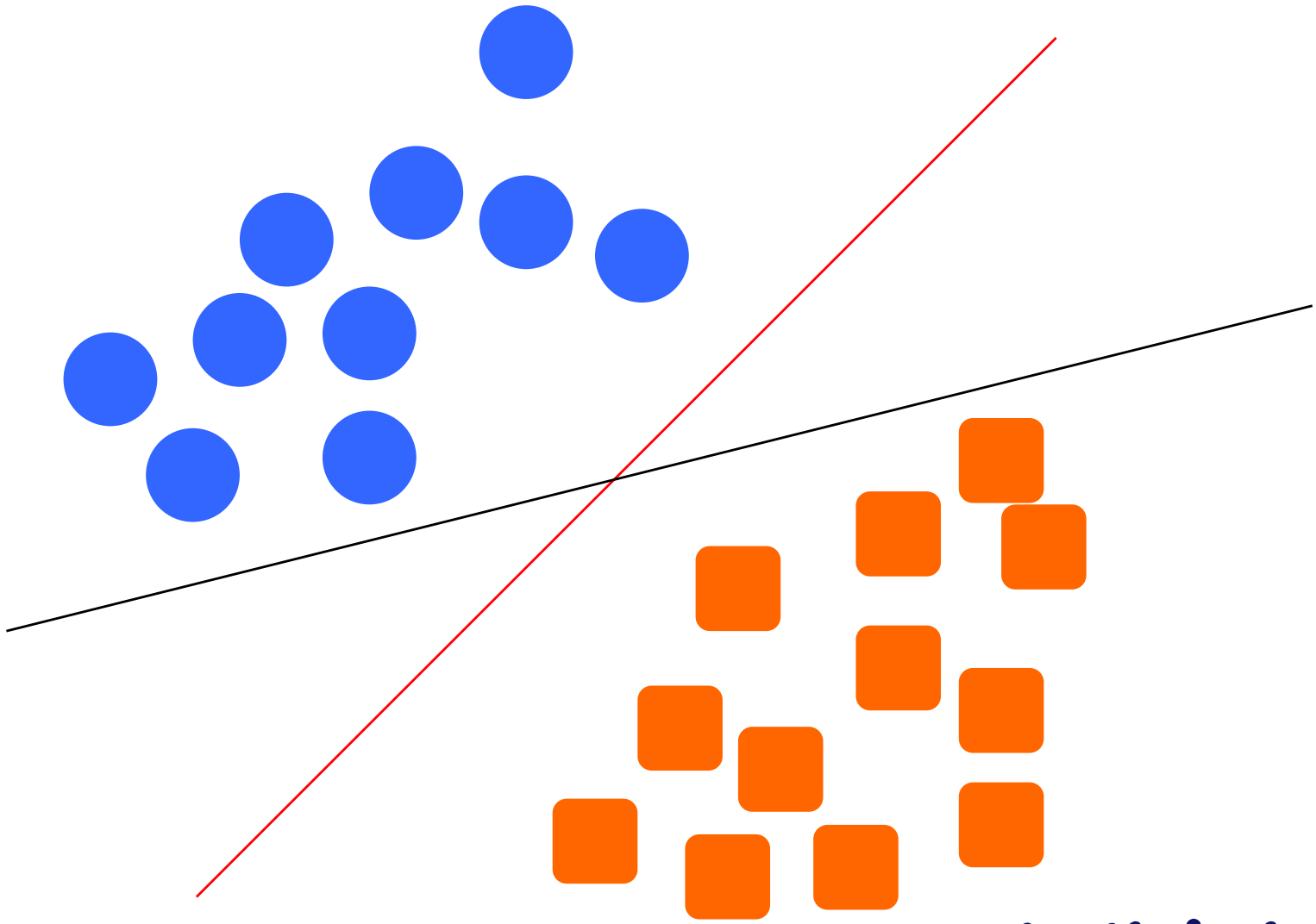
Classification



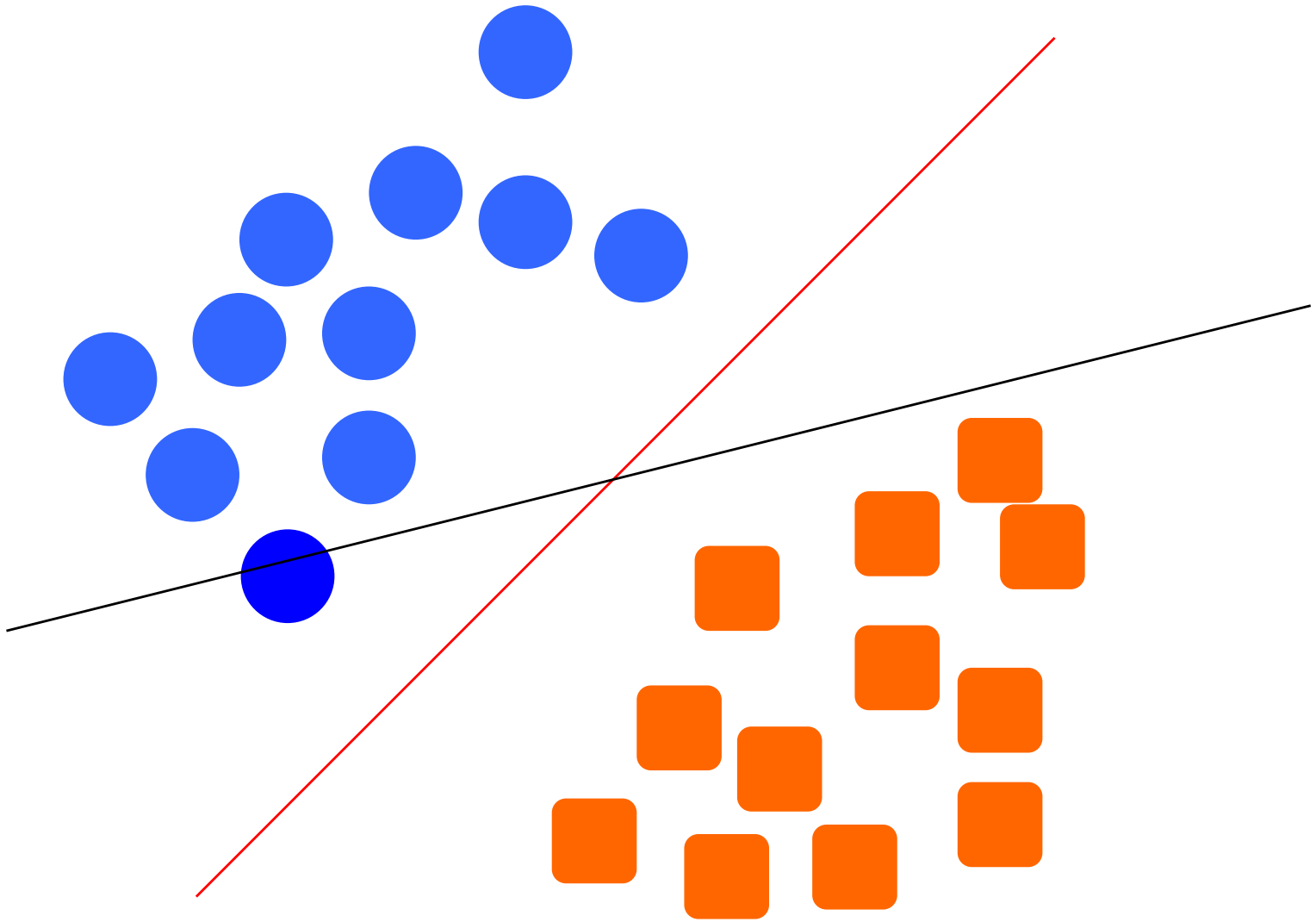
Classification

Support Vector Machine
Artificial Neural Network

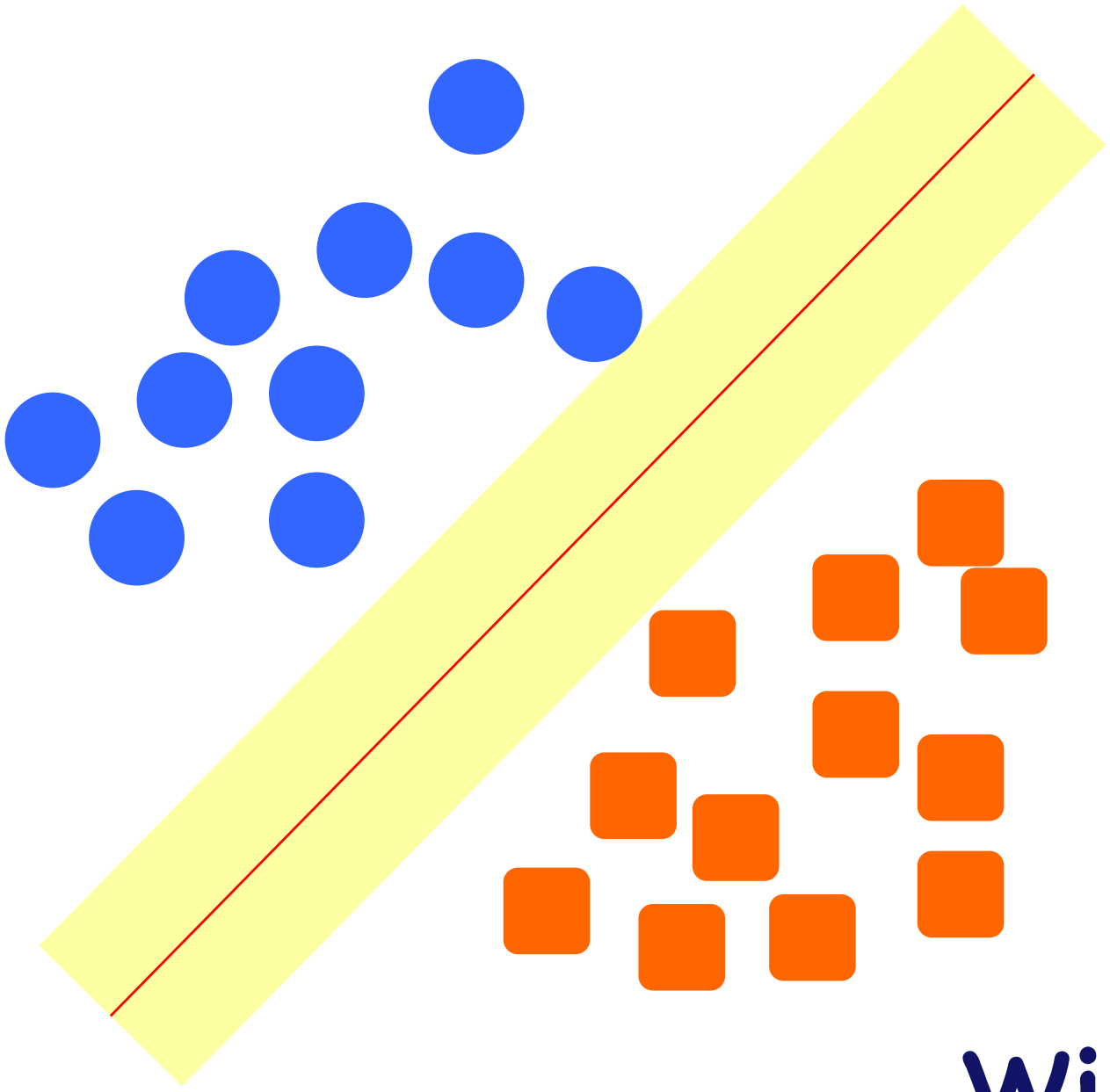




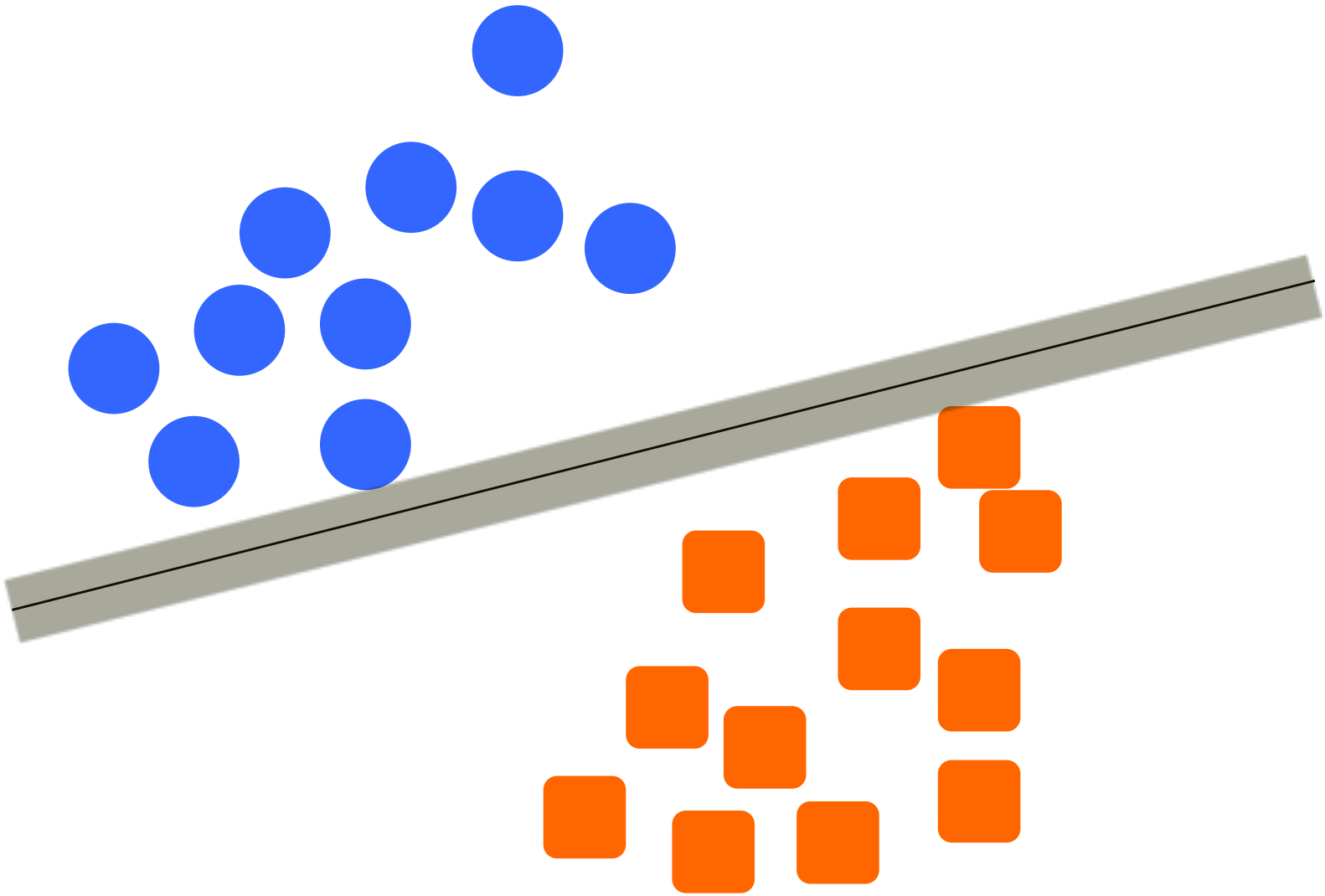
Which one is better?



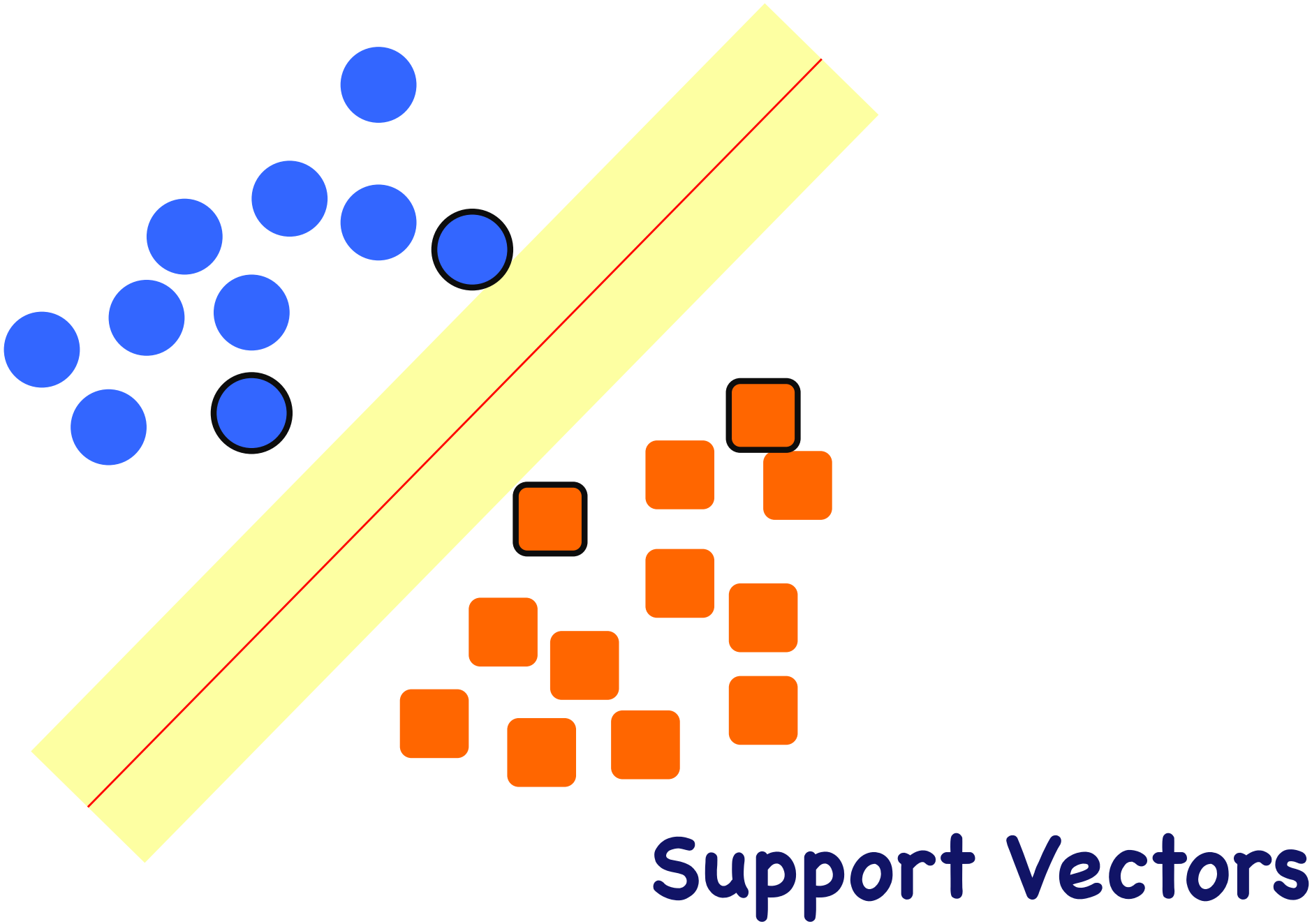
A new data point



Wide margin



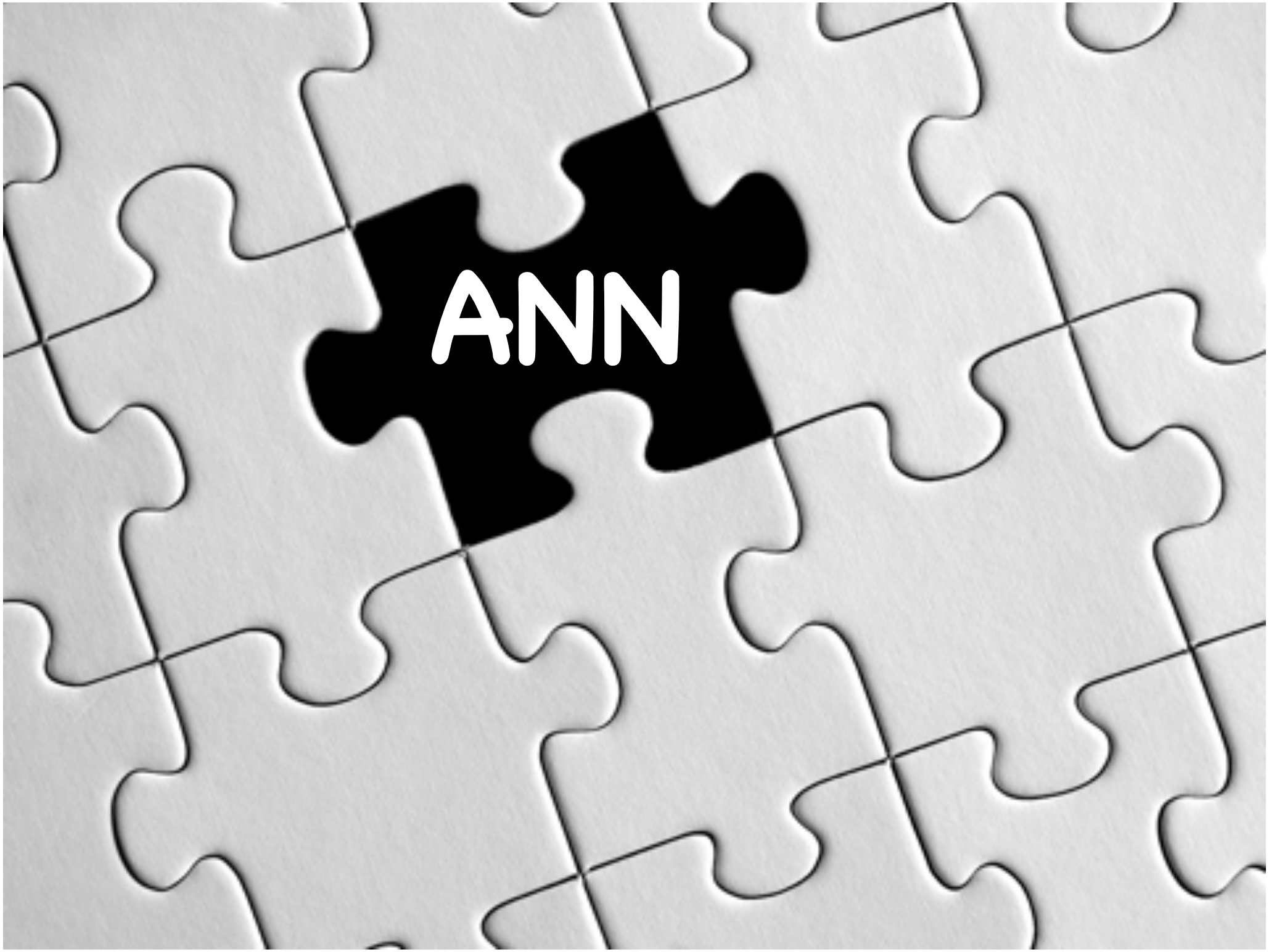
Narrow margin

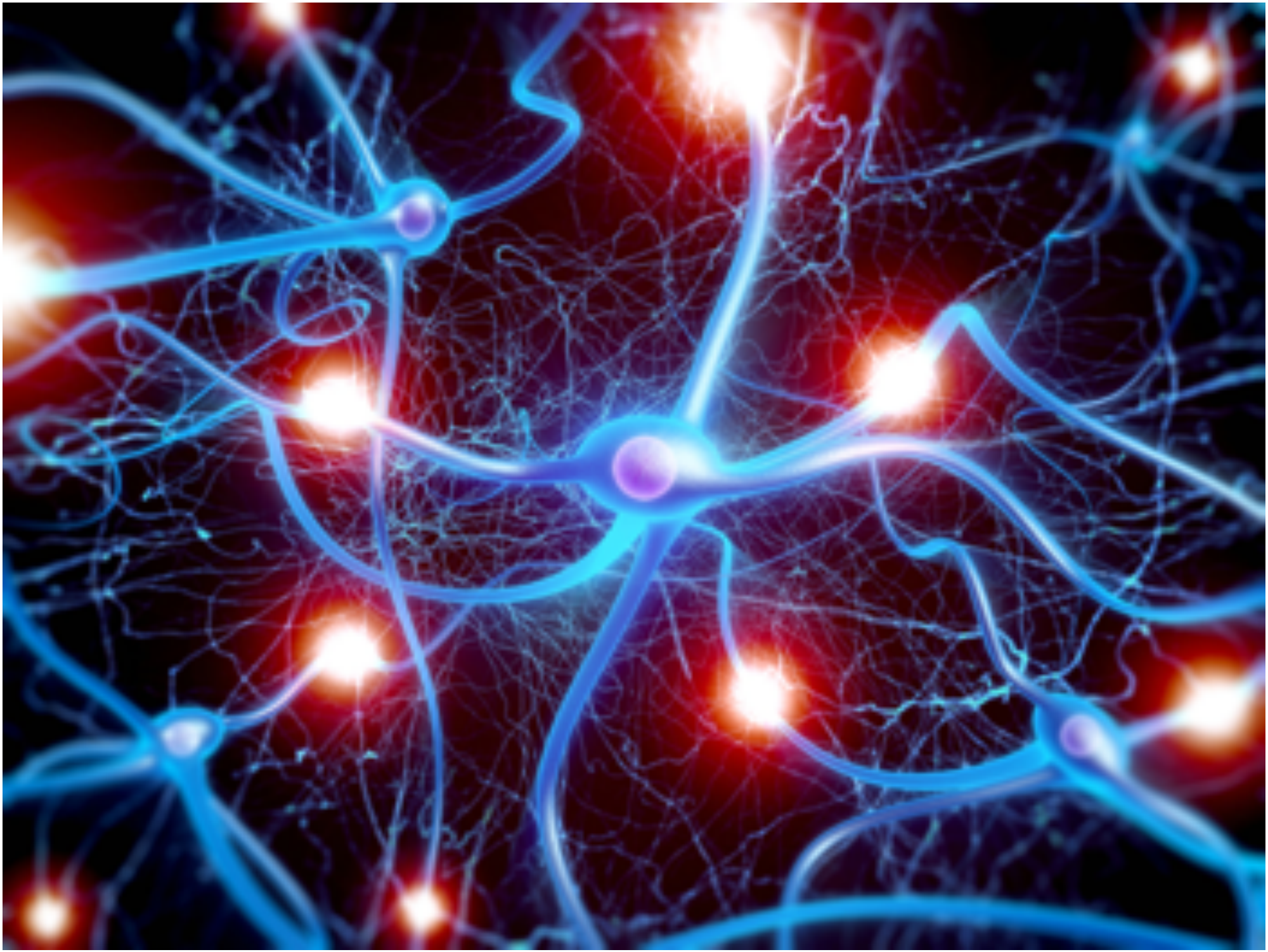

Support Vectors

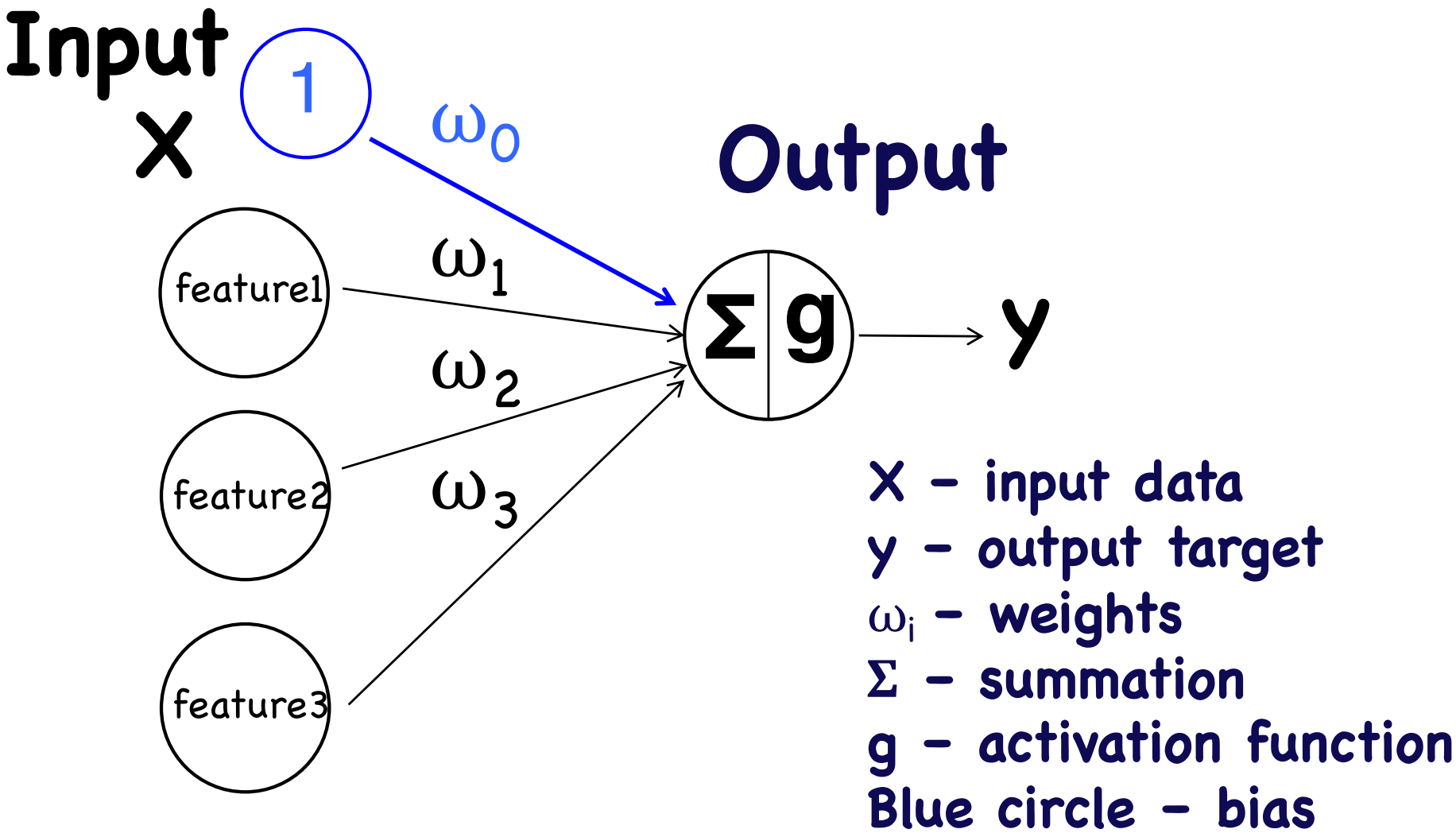
It is maximize the margin!

Support Vectors

More details: https://www.youtube.com/watch?v=eHsErlPJWUU

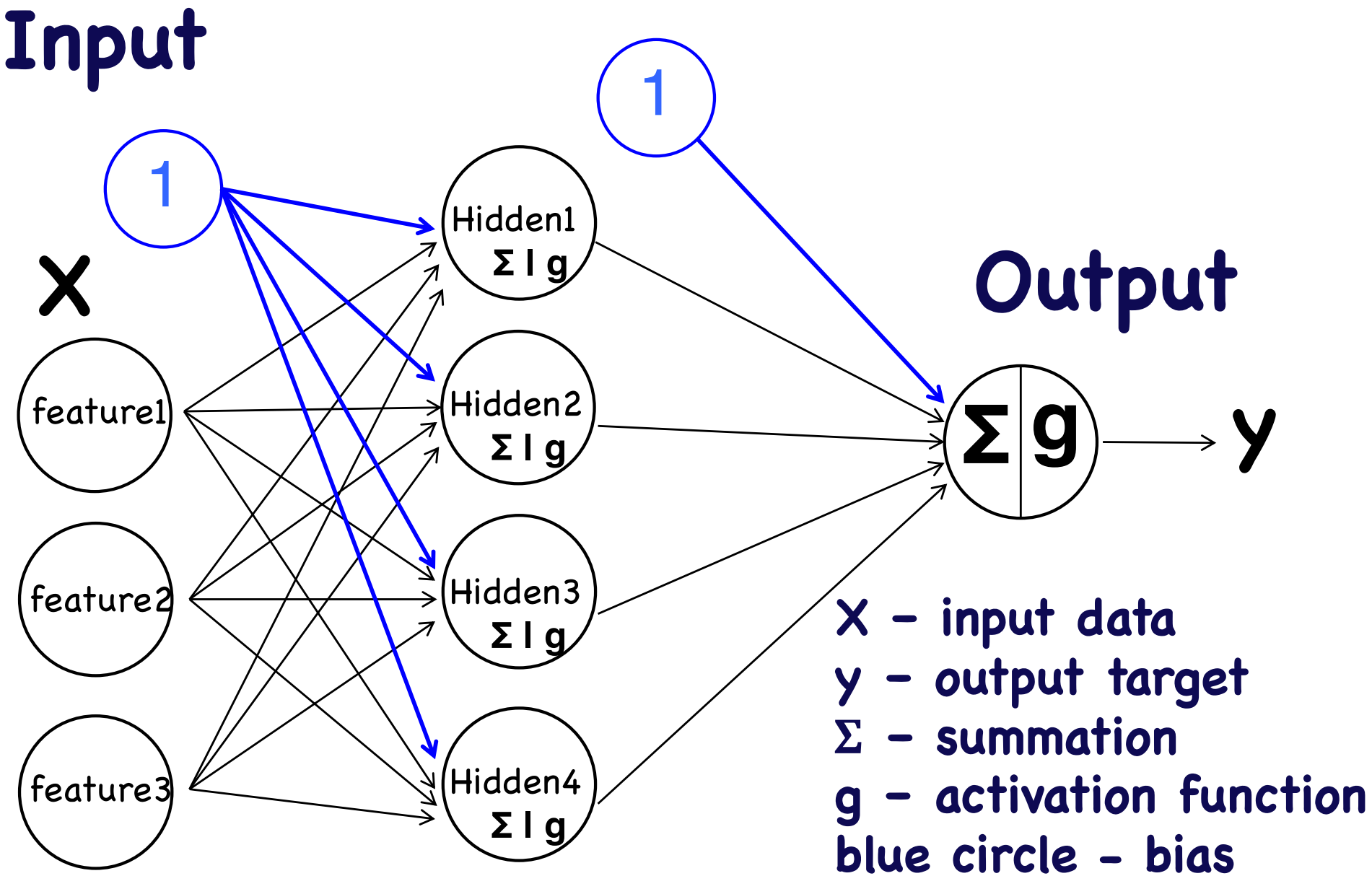

ANN

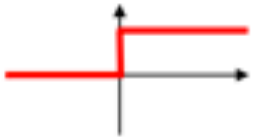
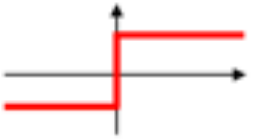
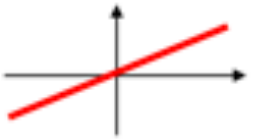
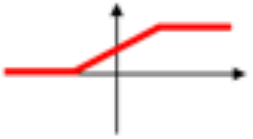
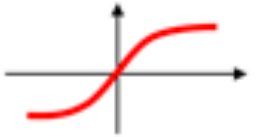
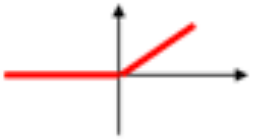




$$Z = \Sigma = \omega_0 x_0 + \omega_1 x_1 + \omega_2 x_2 + \omega_3 x_3 + \ldots + \omega_n x_n$$

$$y = g(\omega_0 x_0 + \omega_1 x_1 + \omega_2 x_2 + \omega_3 x_3 + \ldots + \omega_n x_n)$$


Input
1
1
X
feature1
feature2
feature3
Hidden1
Σ I g
Hidden2
Σ I g
Hidden3
Σ I g
Hidden4
Σ I g
Output
Σ g
y
X - input data
y - output target
Σ - summation
g - activation function
blue circle - bias

Activation function	Equation	Example	1D Graph
Unit step (Heaviside)	$\phi(z) = \begin{cases} 0, & z < 0, \\ 0.5, & z = 0, \\ 1, & z > 0, \end{cases}$	Perceptron variant	
Sign (Signum)	$\phi(z) = \begin{cases} -1, & z < 0, \\ 0, & z = 0, \\ 1, & z > 0, \end{cases}$	Perceptron variant	
Linear	$\phi(z) = z$	Adaline, linear regression	
Piece-wise linear	$\phi(z) = \begin{cases} 1, & z \geq \frac{1}{2}, \\ z + \frac{1}{2}, & -\frac{1}{2} < z < \frac{1}{2}, \\ 0, & z \leq -\frac{1}{2}, \end{cases}$	Support vector machine	
Logistic (sigmoid)	$\phi(z) = \frac{1}{1 + e^{-z}}$	Logistic regression, Multi-layer NN	
Hyperbolic tangent	$\phi(z) = \frac{e^{z} - e^{-z}}{e^{z} + e^{-z}}$	Multi-layer Neural Networks	
Rectifier, ReLU (Rectified Linear Unit)	$\phi(z) = max(0, z)$	Multi-layer Neural Networks	
Rectifier, softplus	$\phi(z) = \ln(1 + e^{z})$	Multi-layer Neural Networks	


YOU'RE IN MY SPOT
GRAPHICS GARAGE

Intuitive
Artificial Neural Network

Input

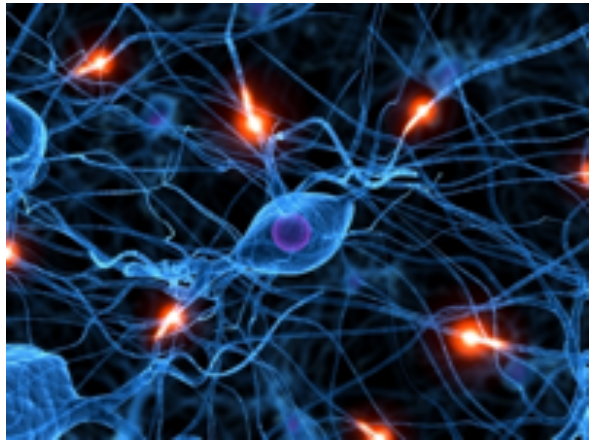
.
.
.

w_1

w_2

w_n

$F(eye \times w_1 + nose \times w_2 + \ldots + mouth \times w_n)$



Output

Intuitive
Artificial Neural Network

Input

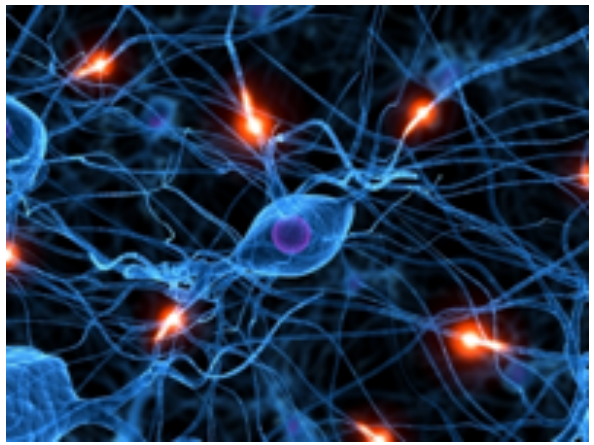
.
.
.

w_1

w_2

w_n

$F(eye \times w_1 + nose \times w_2 + \ldots + mouth \times w_n)$



Output

error
feedback

Intuitive
Artificial Neural Network

Input

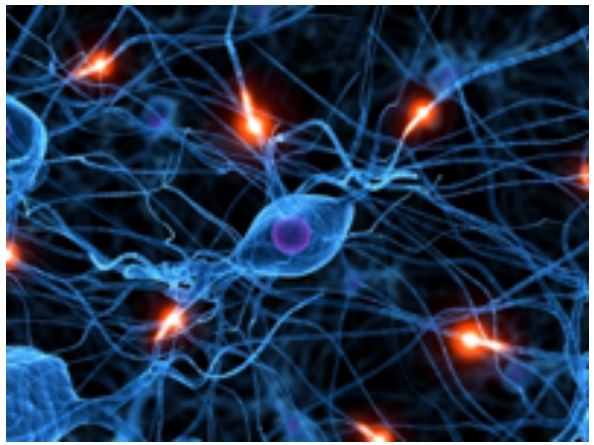
.
.
.

w_1

w_2

w_n

$F(eye \times w_1 + nose \times w_2 + \ldots + mouth \times w_n)$



Output

error
feedback

Intuitive
Artificial Neural Network

Input

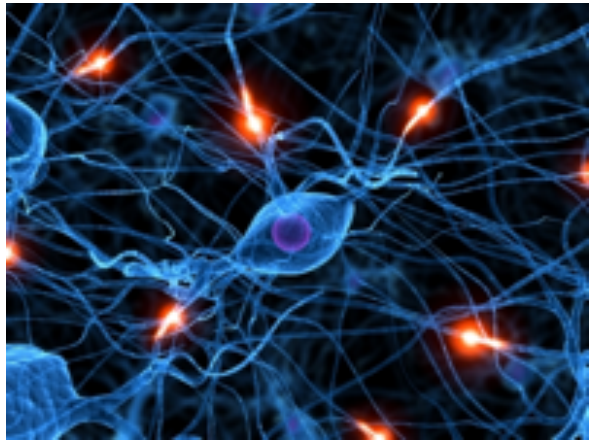
.
.
.

w_1

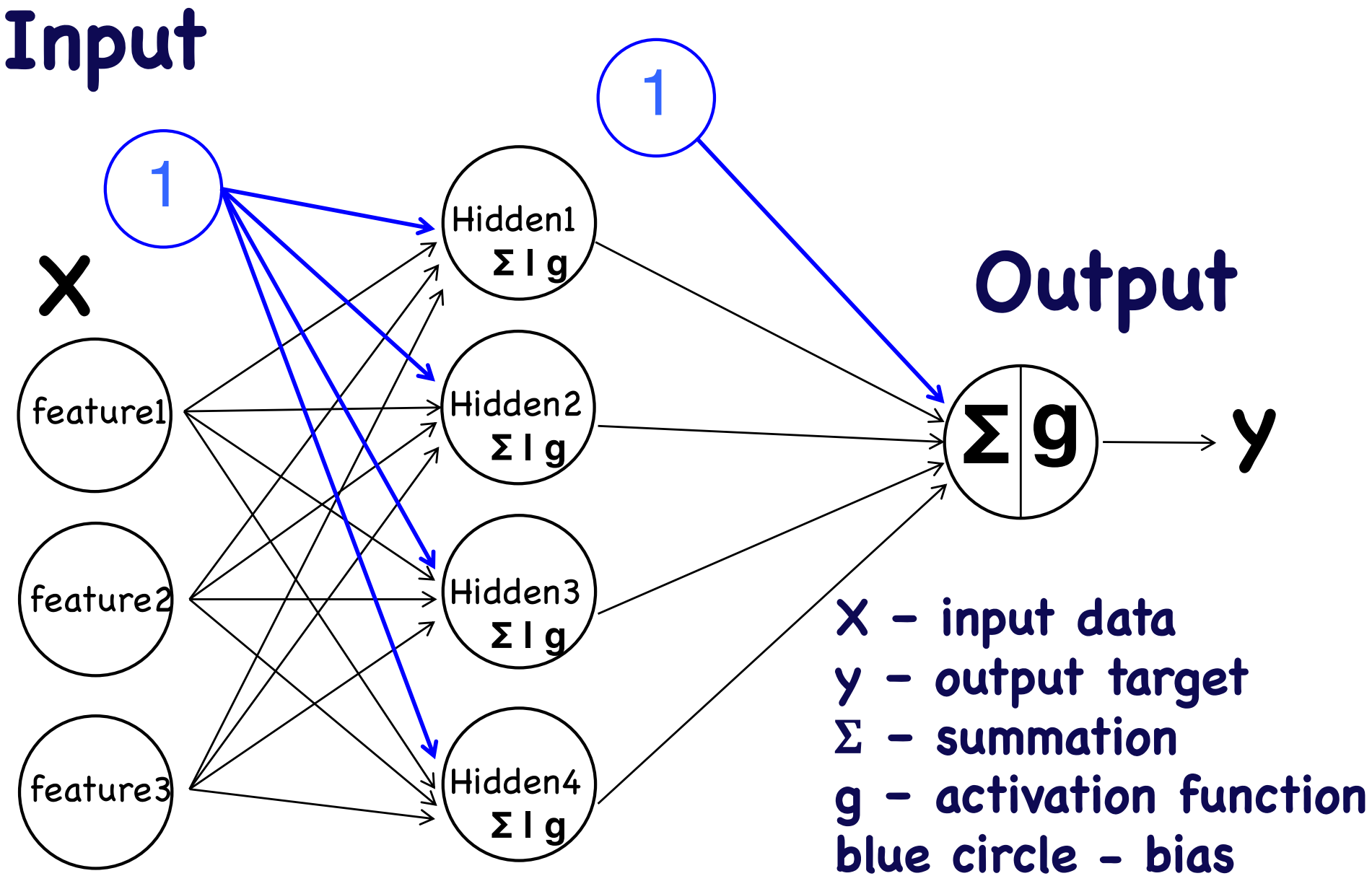
w_2

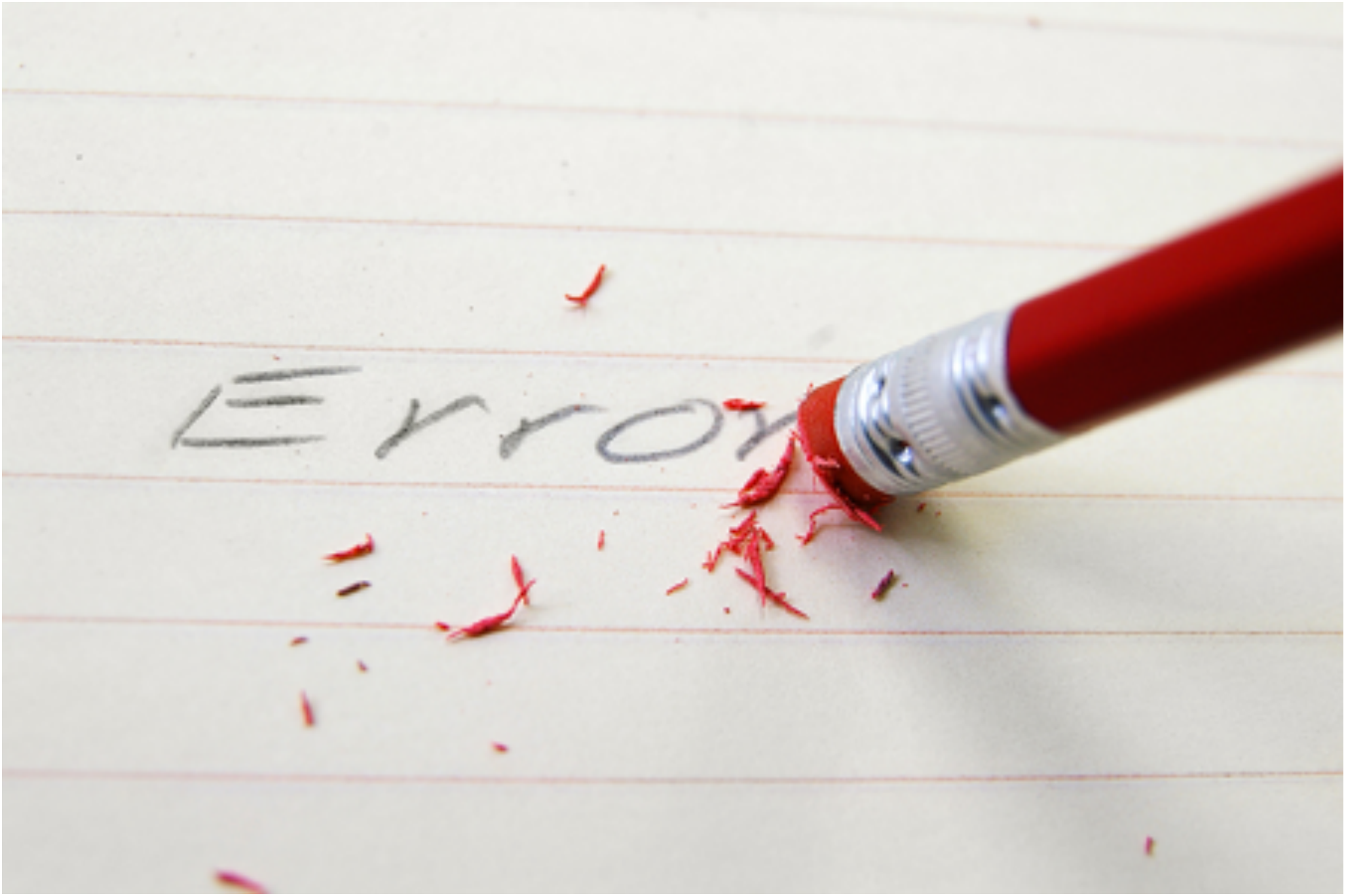
w_n

$F(eye \times w_1 + nose \times w_2 + \ldots + mouth \times w_n)$

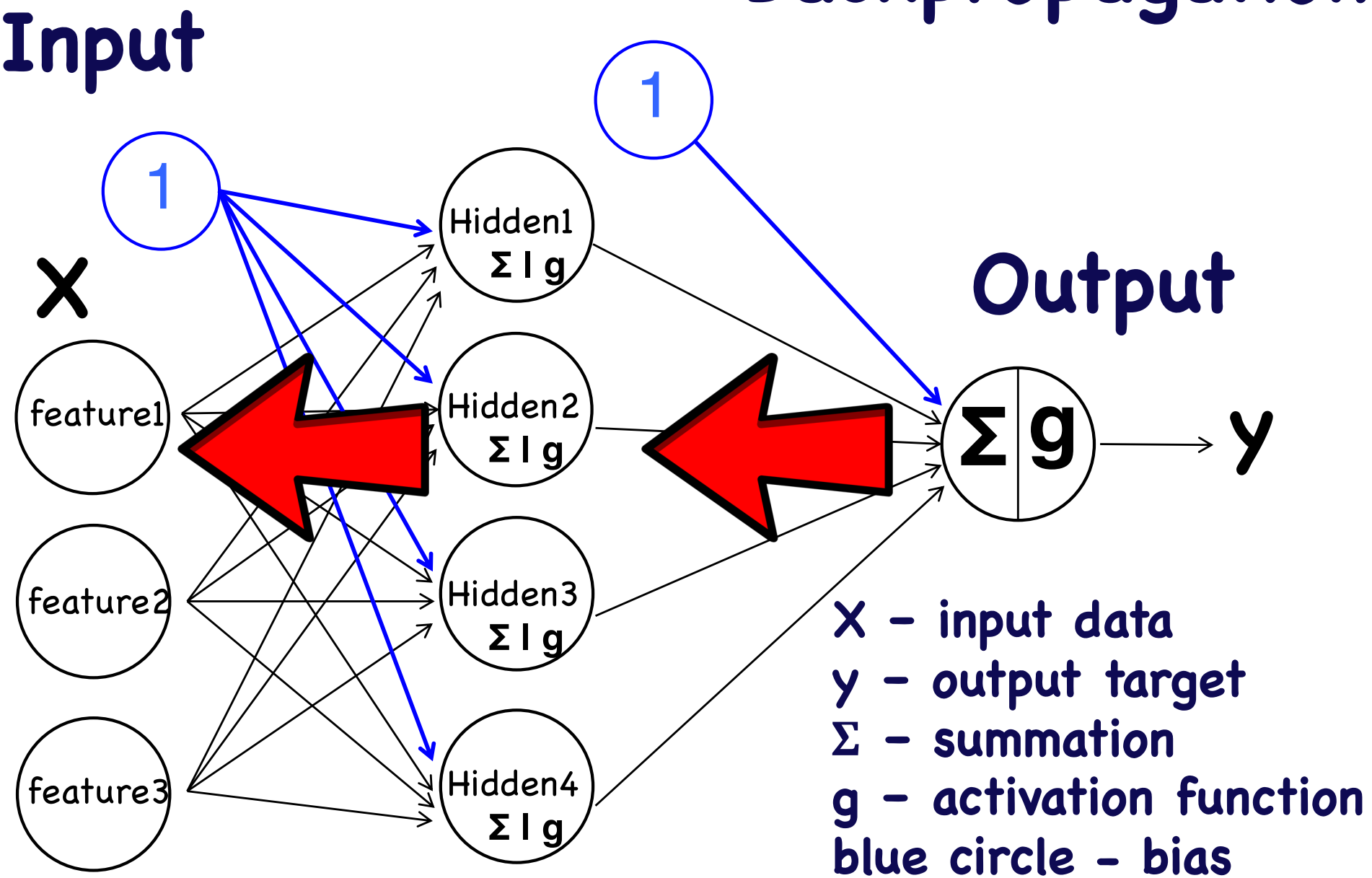


Output


Input
1
1
X
feature1
feature2
feature3
Hidden1
Σ I g
Hidden2
Σ I g
Hidden3
Σ I g
Hidden4
Σ I g
Output
Σ g
y
X - input data
y - output target
Σ - summation
g - activation function
blue circle - bias

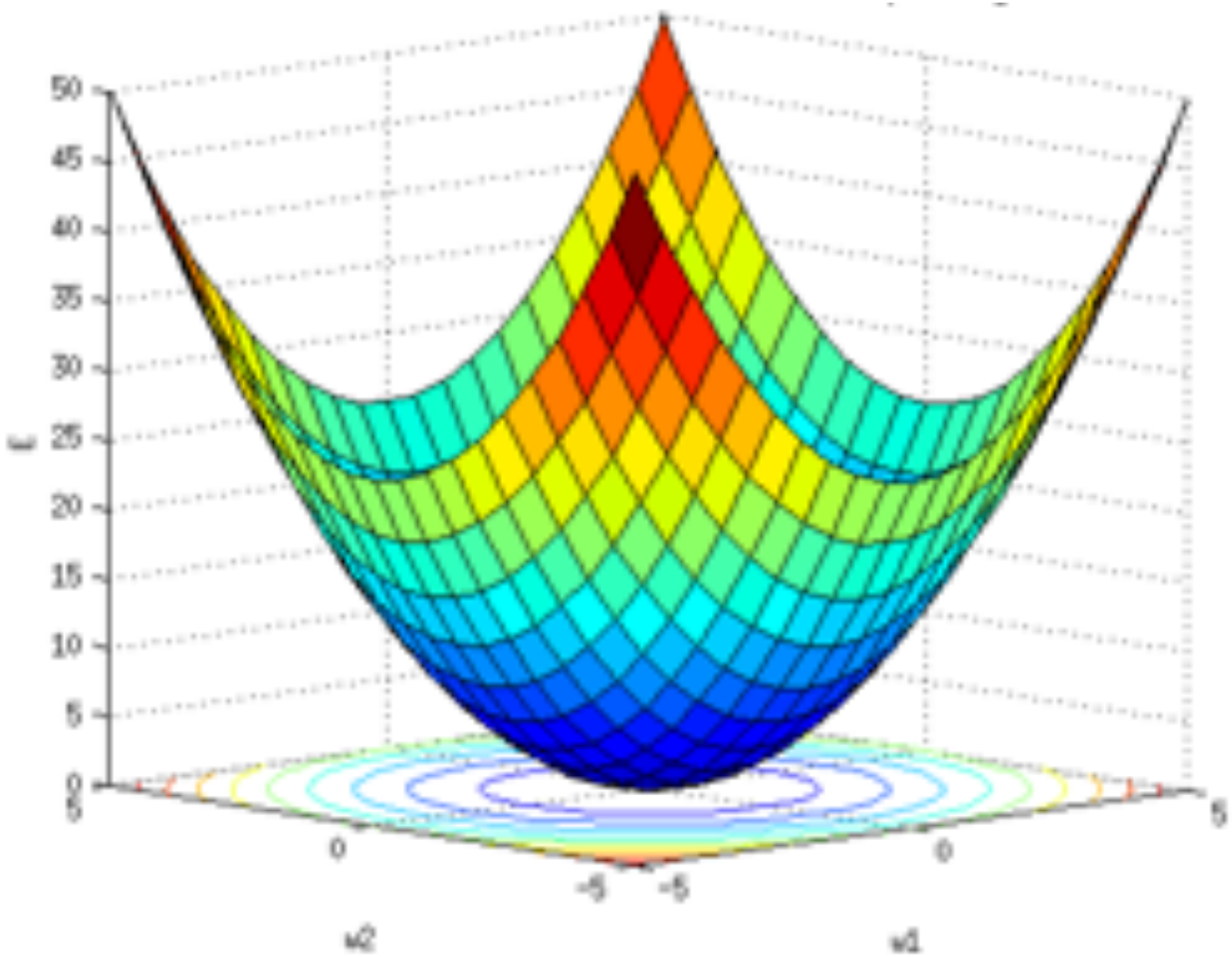

ERROR

Backpropagation



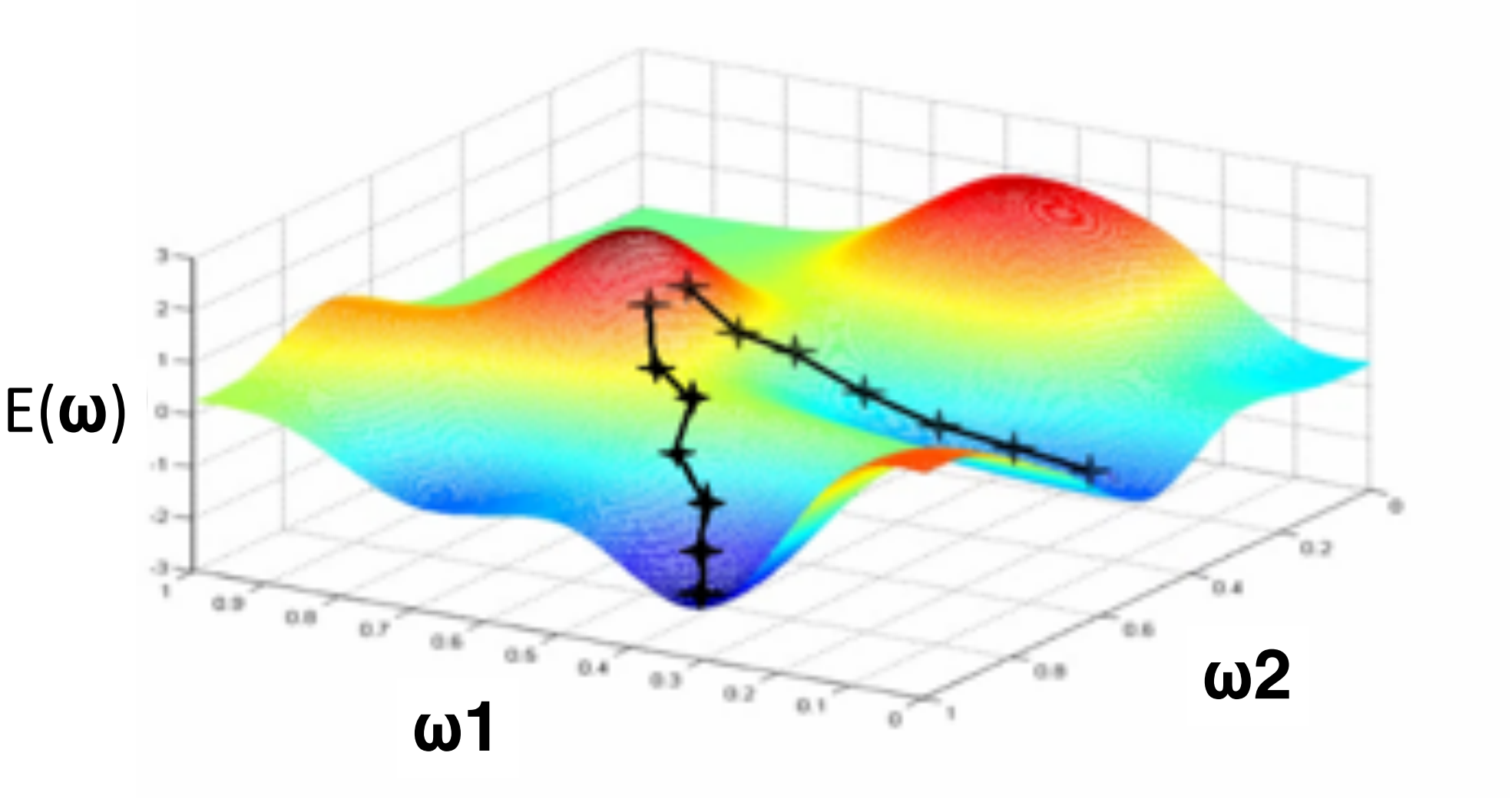
More details: https://www.youtube.com/watch?v=x_Eamf8MHwU

Ideal Cost Function



Real-world Cost Function

Gradient descend



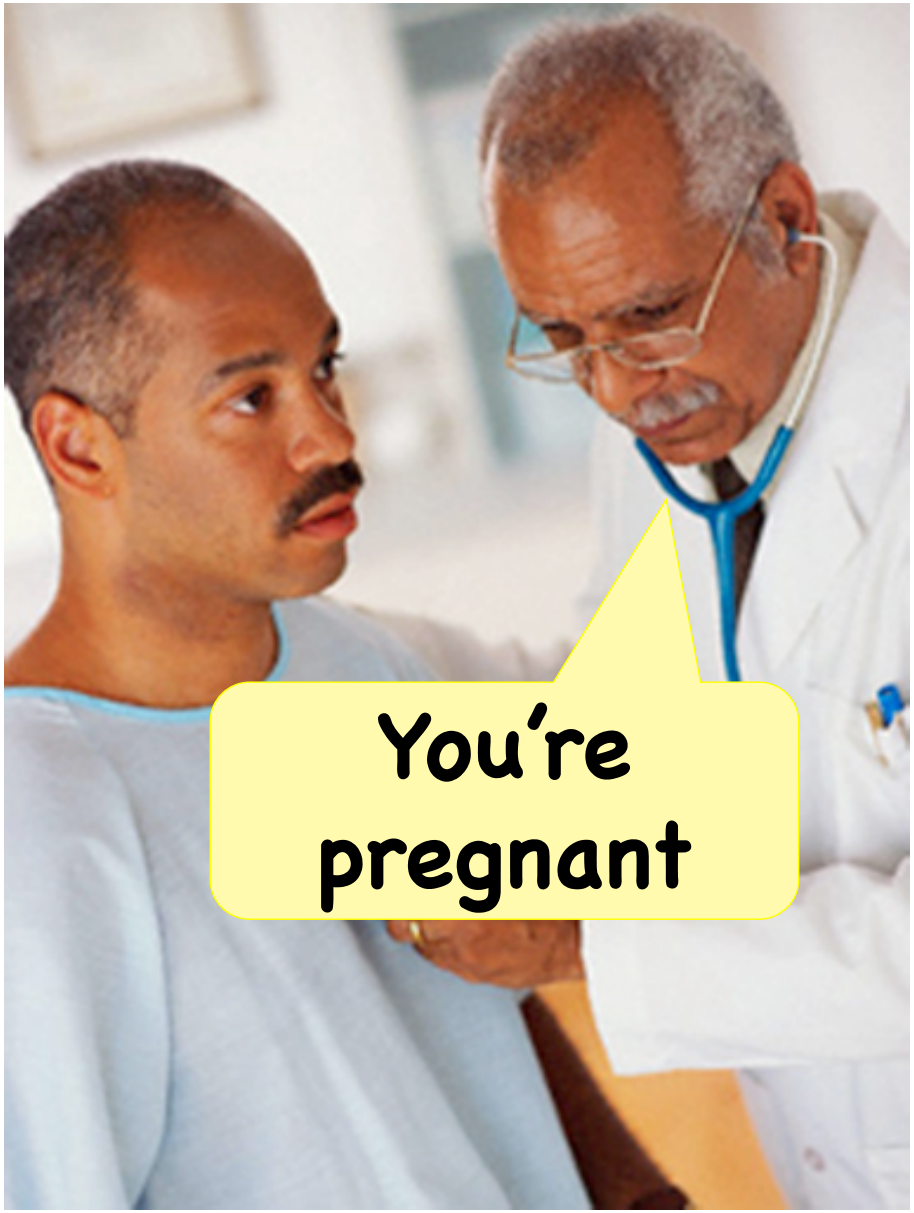
More details: https://www.youtube.com/watch?v=TveJaJs5Fqk

Iterate many times

Evaluate performance

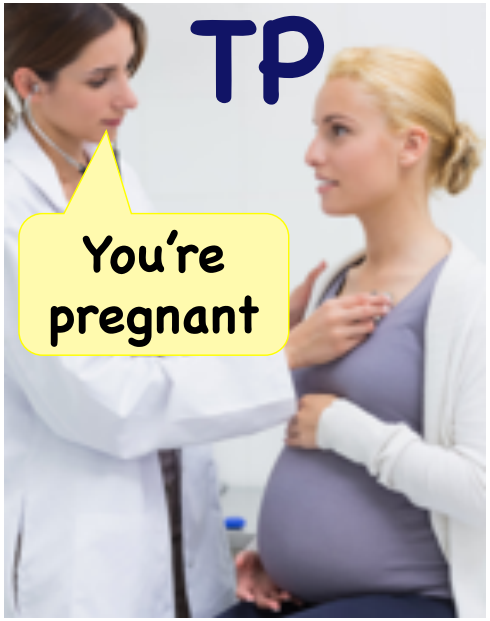
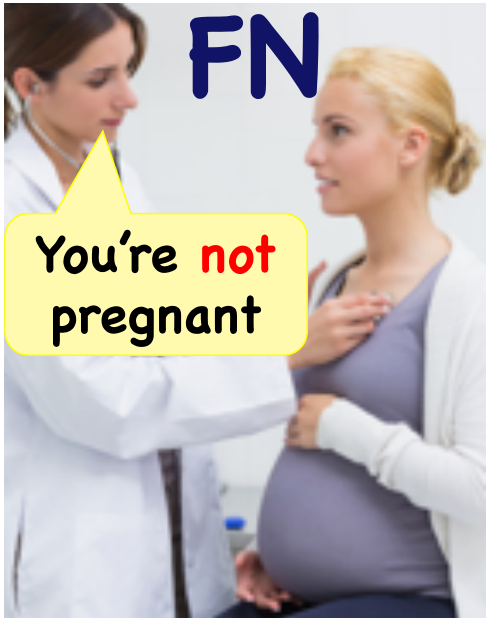
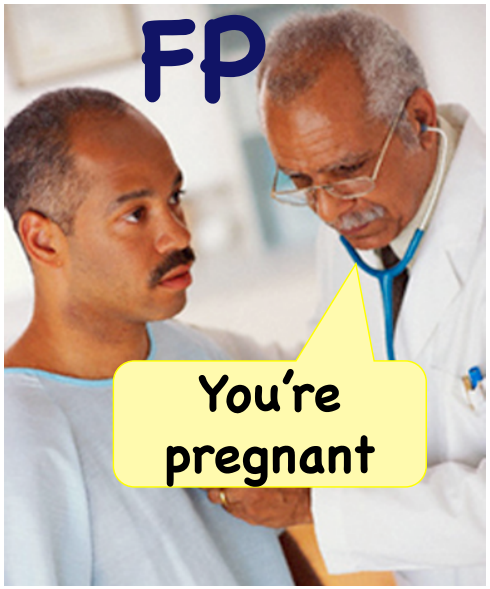
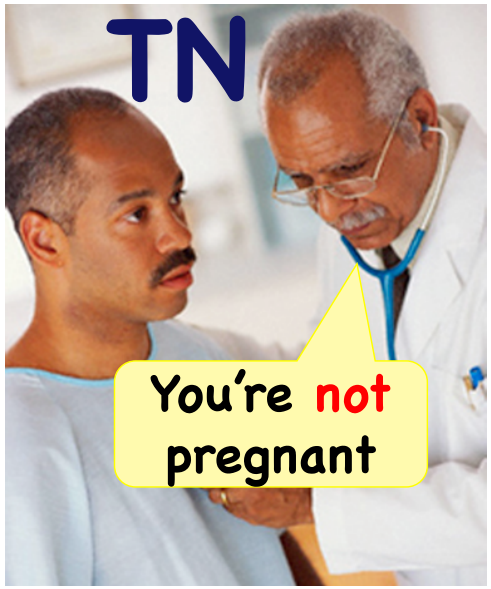
- Confusion matrix
- Precision & Recall
- ROC curve (Receiver operating characteristic)

False Positive



False Negative

Confusion matrix

True class \ Predicted class	P	N
P	TP 	FN 
N	FP 	TN 

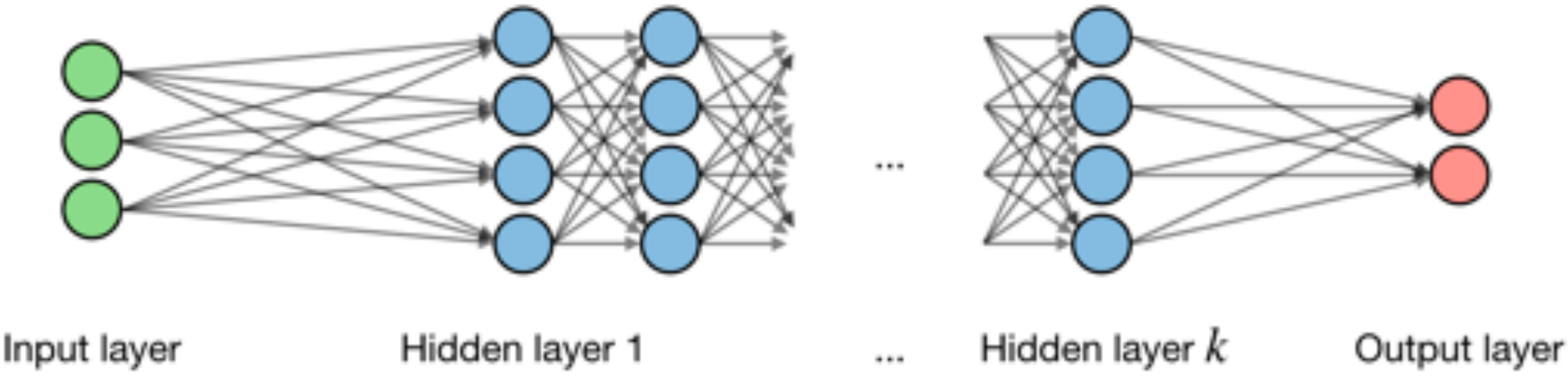
Some useful resources

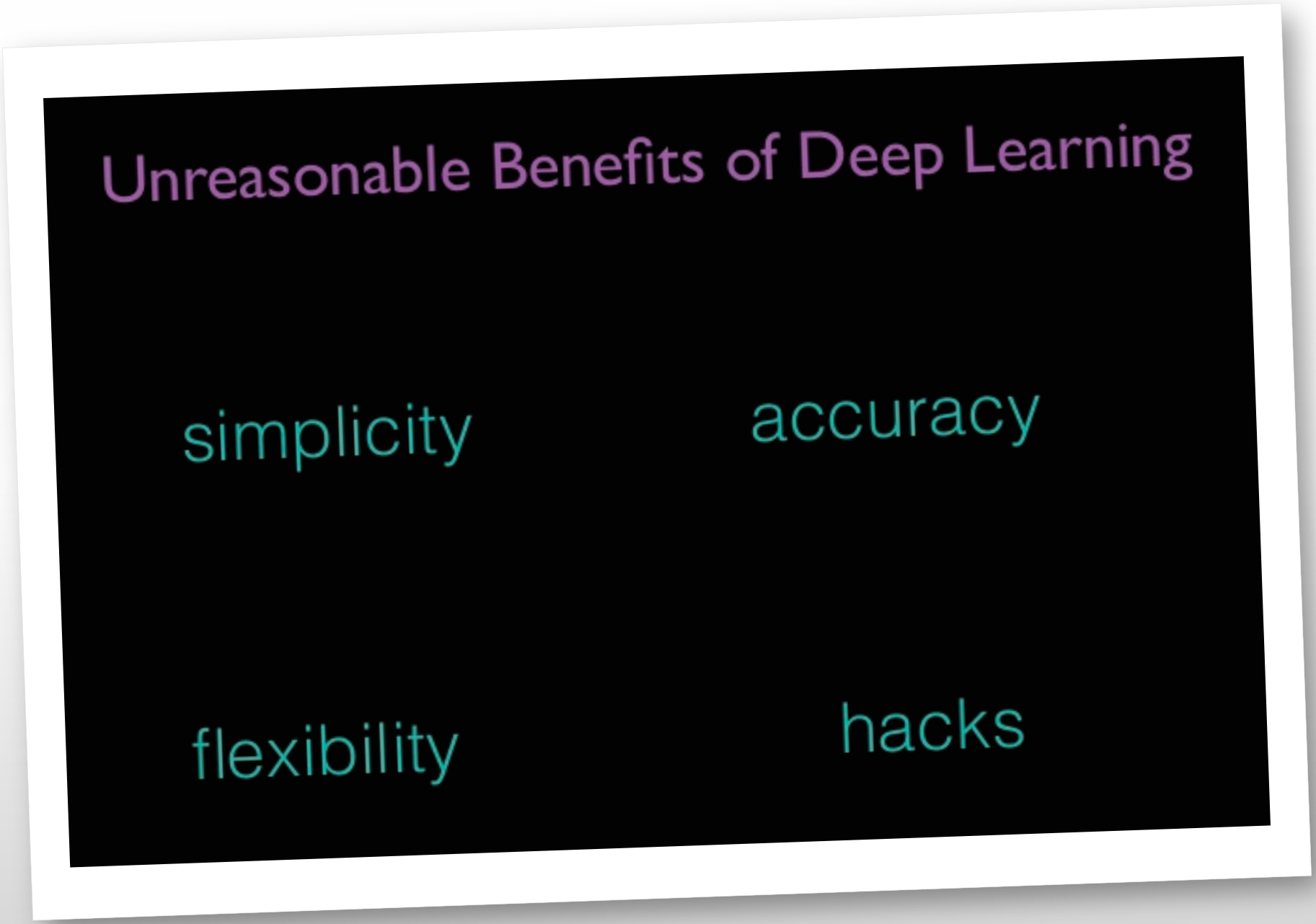
- https://playground.tensorflow.org
- https://seat.massey.ac.nz/personal/s.r.marsland/MLBook.html
- http://neuralnetworksanddeeplearning.com/
- Python machine learning - Sebastian Raschka
- Deep learning with Python - Francois Chollet
- Machine Learning - An Algorithmic Perspective - Stephen Marsland
- Datasets put together by Men-Andrin and Zach Ross
  - http://scedc.caltech.edu/research-tools/deeplearning.html

Many pictures in this talk are from internet, I thank all the authors here!

Go to the notebook

Deep neural network





https://www.slideshare.net/indicods/the-unreasonable-benefits-of-deep-learning

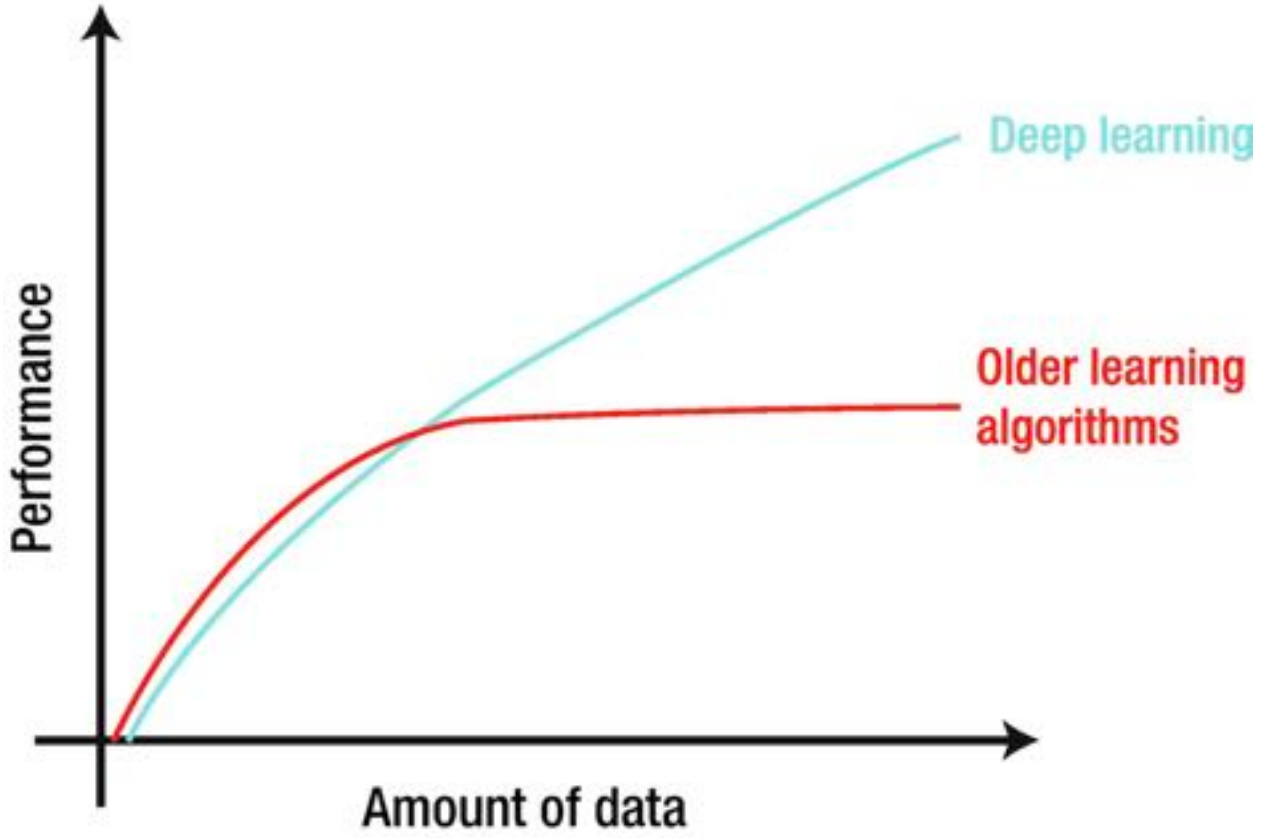
Benefits of deep learning
Simplicity

Traditional Machine Learning Flow

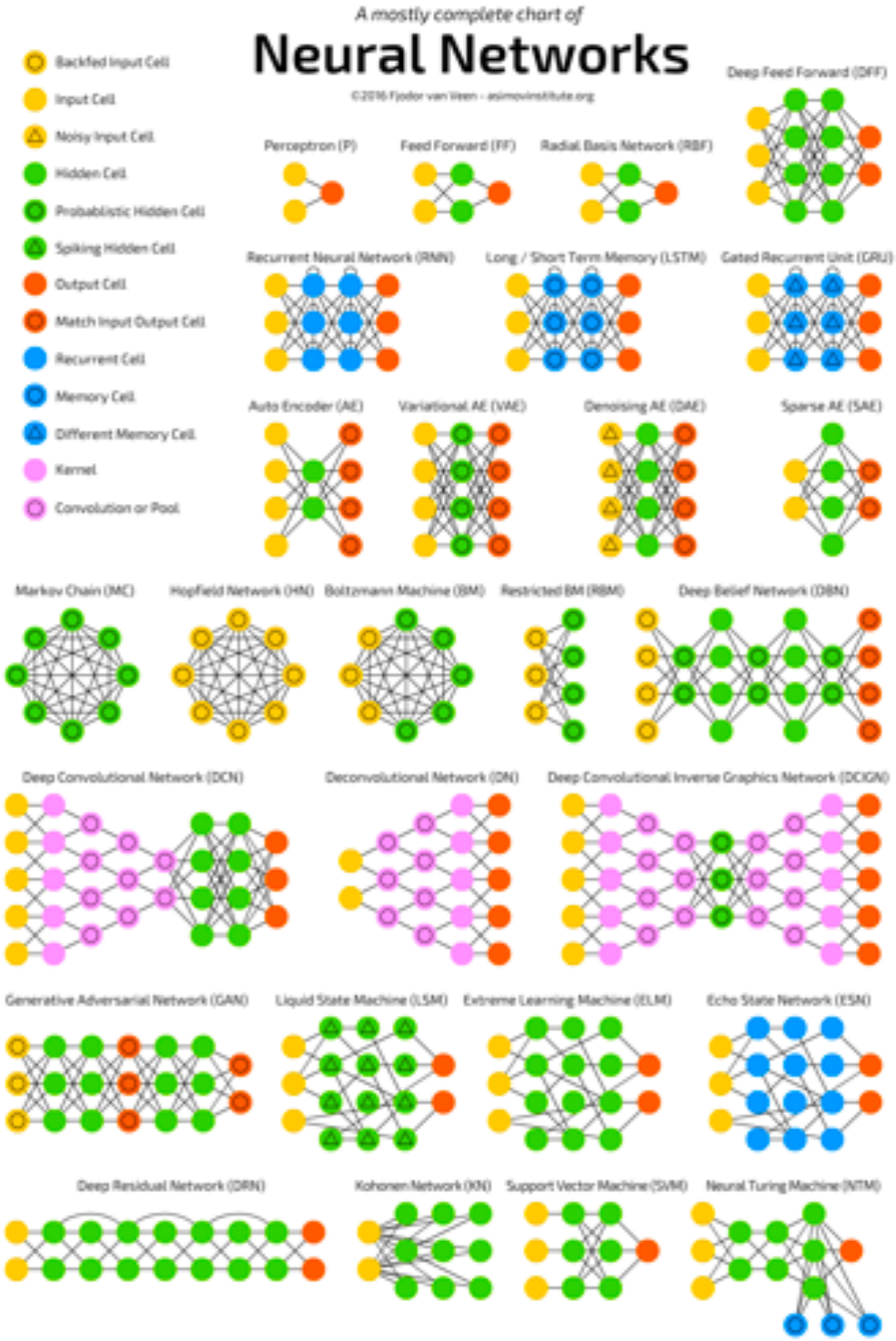
Deep Learning Flow

Benefits of deep learning

Accuracy



https://link.springer.com/chapter/10.1007/978-1-4842-3685-7_1

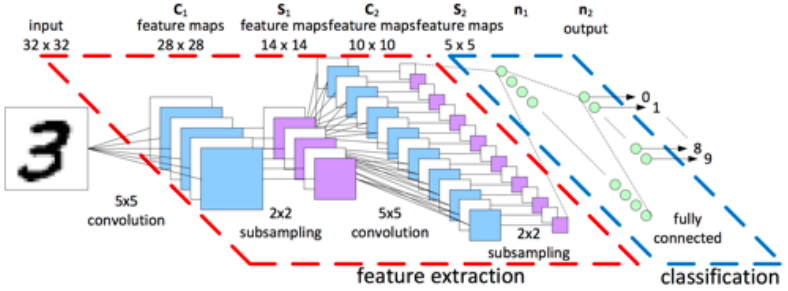


https://towardsdatascience.com/the-mostly-complete-chart-of-neural-networks-explained-3fb6f2367464

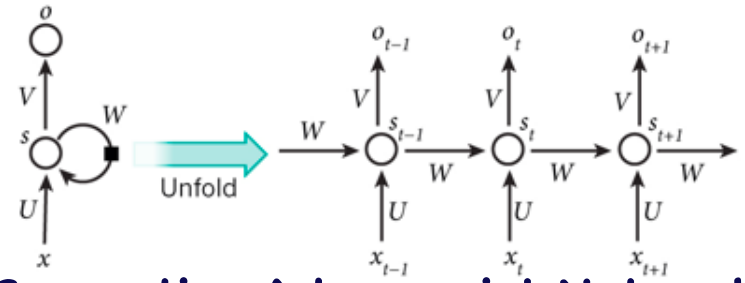
Benefits

Flexible

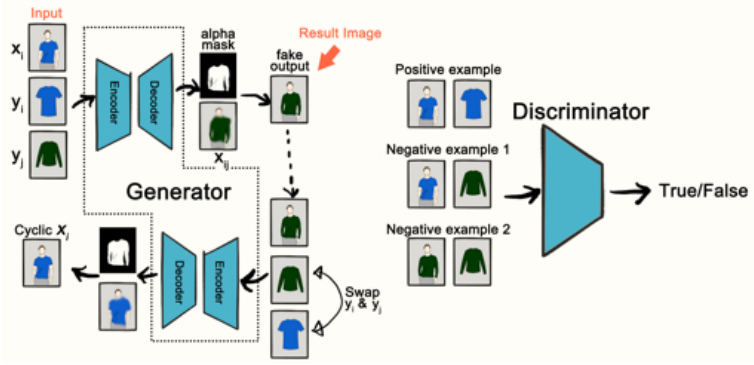
Convolutional Neural Network



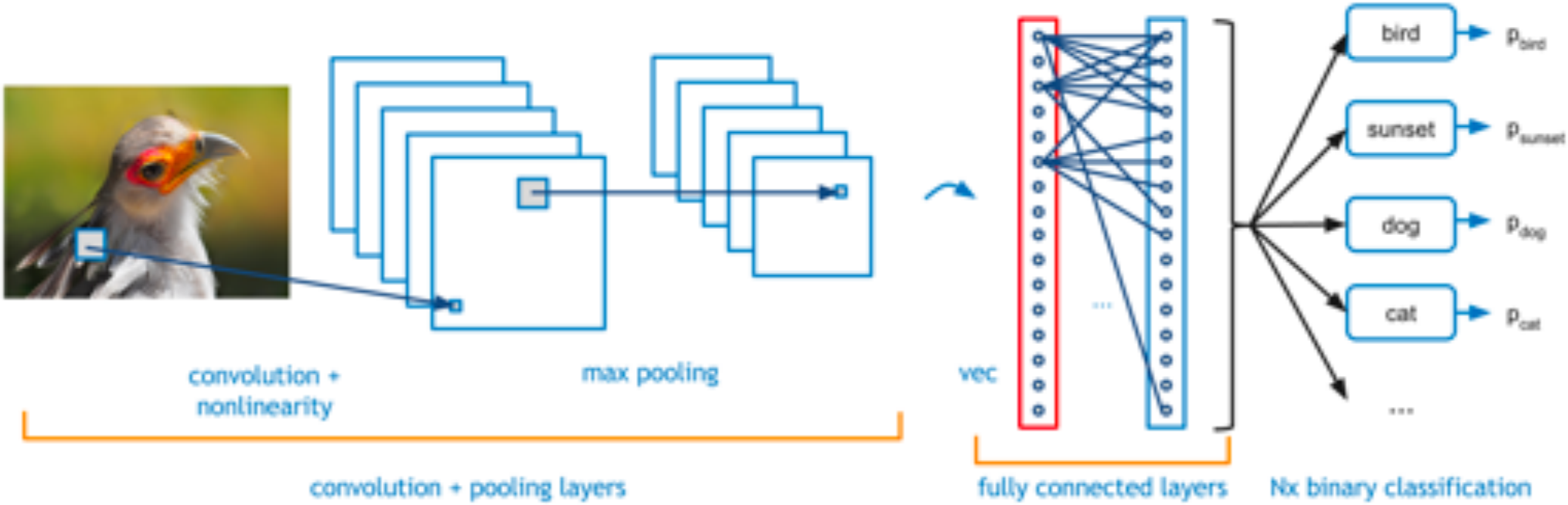
Recurrent Neural Network



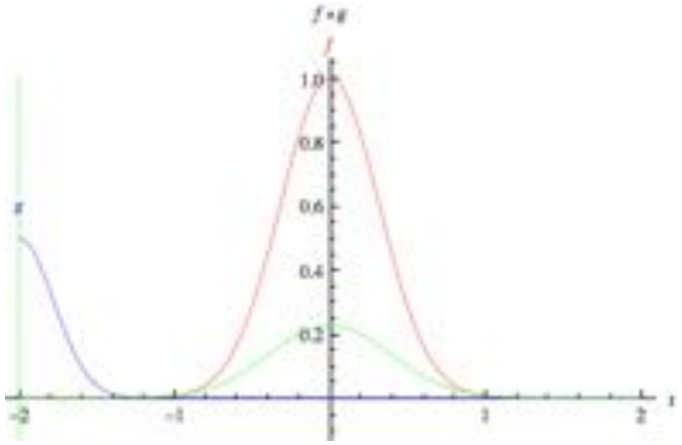
Generative Adversarial Network



Convolutional neural network

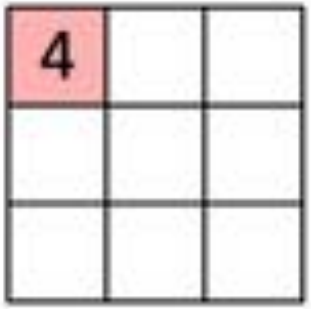


Convolution operation



http://mathworld.wolfram.com/

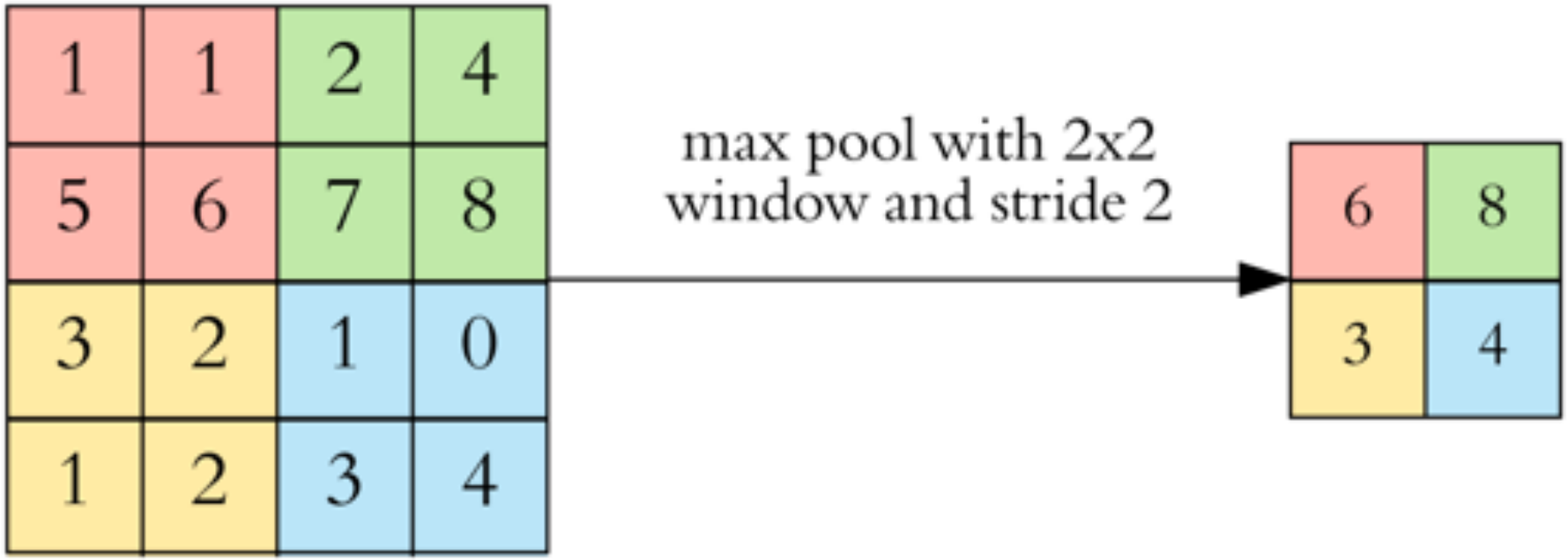
Image



Convolved Feature

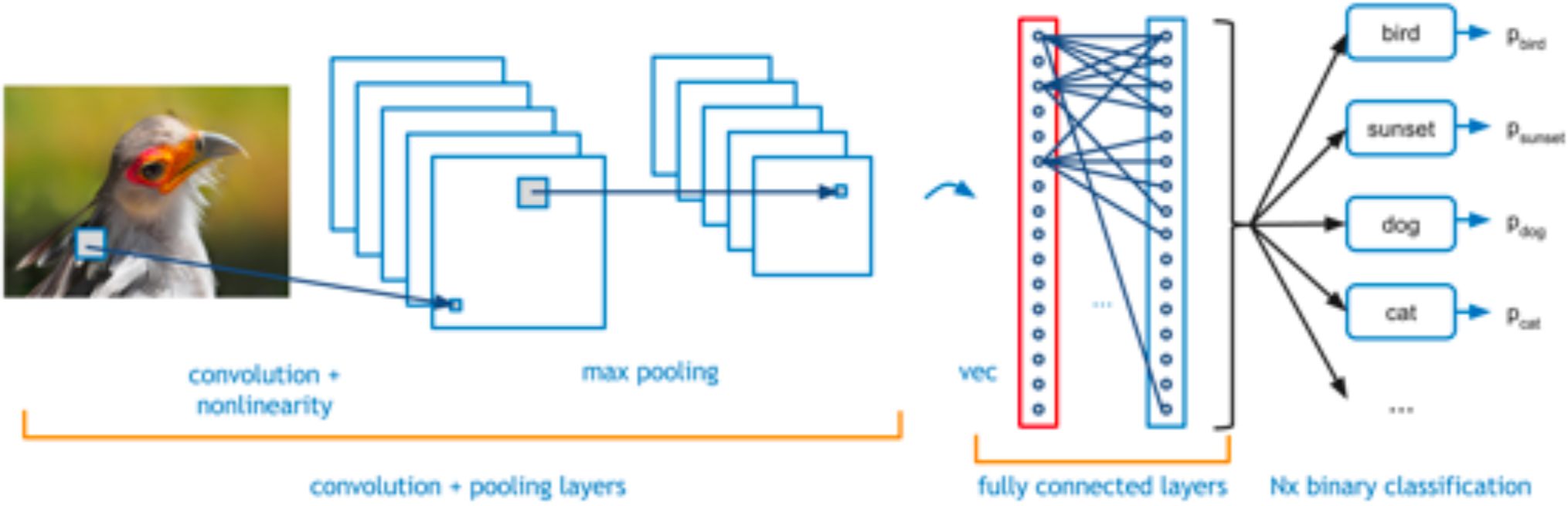
https://ujjwalkarn.me/2016/08/11/intuitive-explanation-convnets/

Max pooling



https://towardsdatascience.com/applied-deep-learning-part-4-convolutional-neural-networks-584bc134c1e2

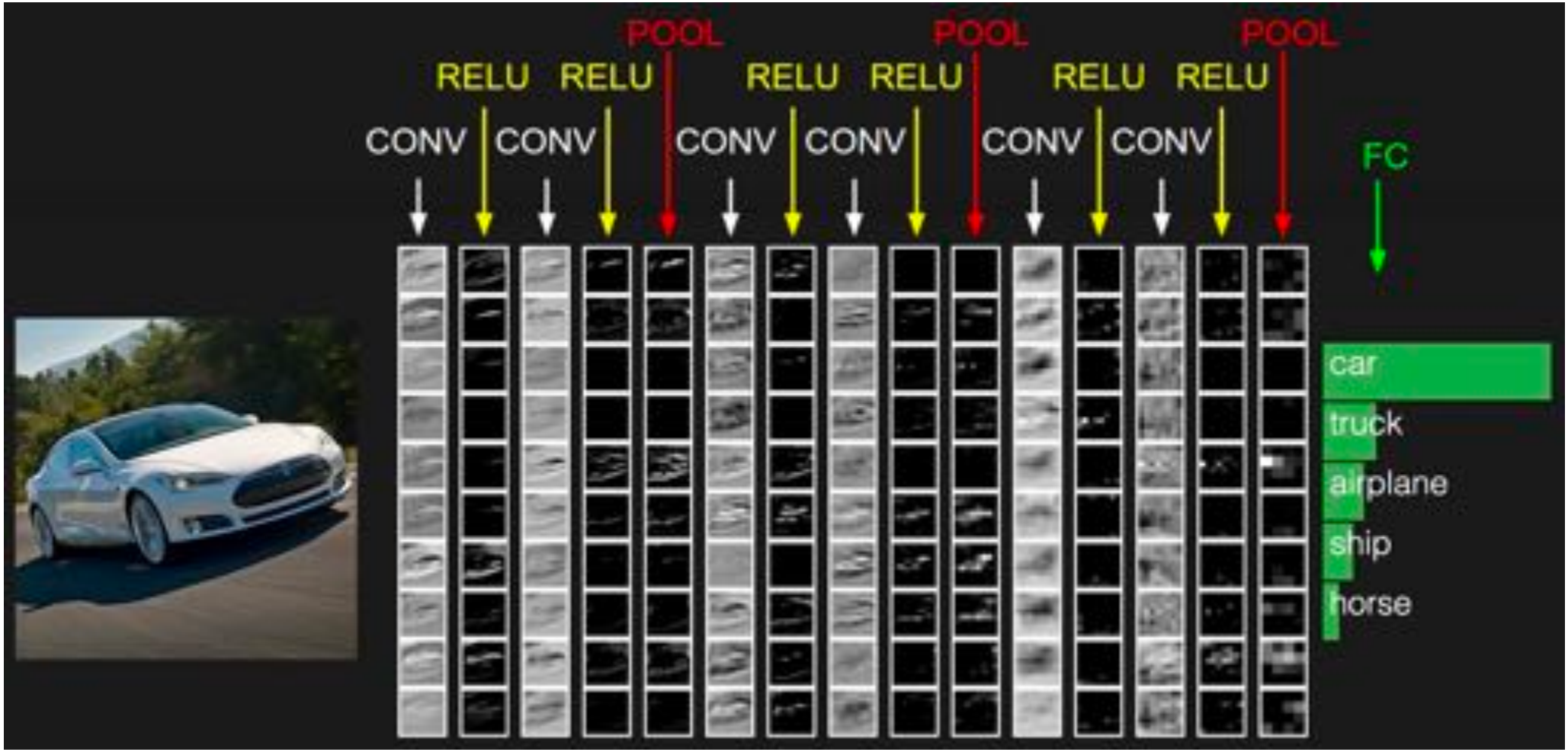
Convolutional neural network



Feature extraction

Classification

CNN example


POOL
POOL
POOL
RELU
RELU
RELU
RELU
RELU
RELU
CONV
CONV
CONV
CONV
CONV
CONV
FC
car
truck
airplane
ship
horse

Some useful resources

- http://cs231n.github.io/convolutional-networks/
- https://ujjwalkarn.me/2016/08/11/intuitive-explanation-convnets/
- http://neuralnetworksanddeeplearning.com/
- Deep learning with Python - Francois Chollet
- Deep learning – Ian Goodfellow
- Datasets put together by Men-Andrin and Zach Ross
  - http://scedc.caltech.edu/research-tools/deeplearning.html

Many pictures in this talk are from internet, I thank all the authors here!

Go to the notebook

Thanks

kongqk@Berkeley.edu

University of California Berkeley Seismological Laboratory

http://seismo.berkeley.edu/qingkaikong/