

Deep Learning and Neural Networks

Amine BENDAHMANE
PhD candidate in AI & Robotics

Dev Day 2.0, ESI SBA
Dec 15th, 2018

This talk is...

An attempt to simplify Deep Learning concepts to
mathematics haters'

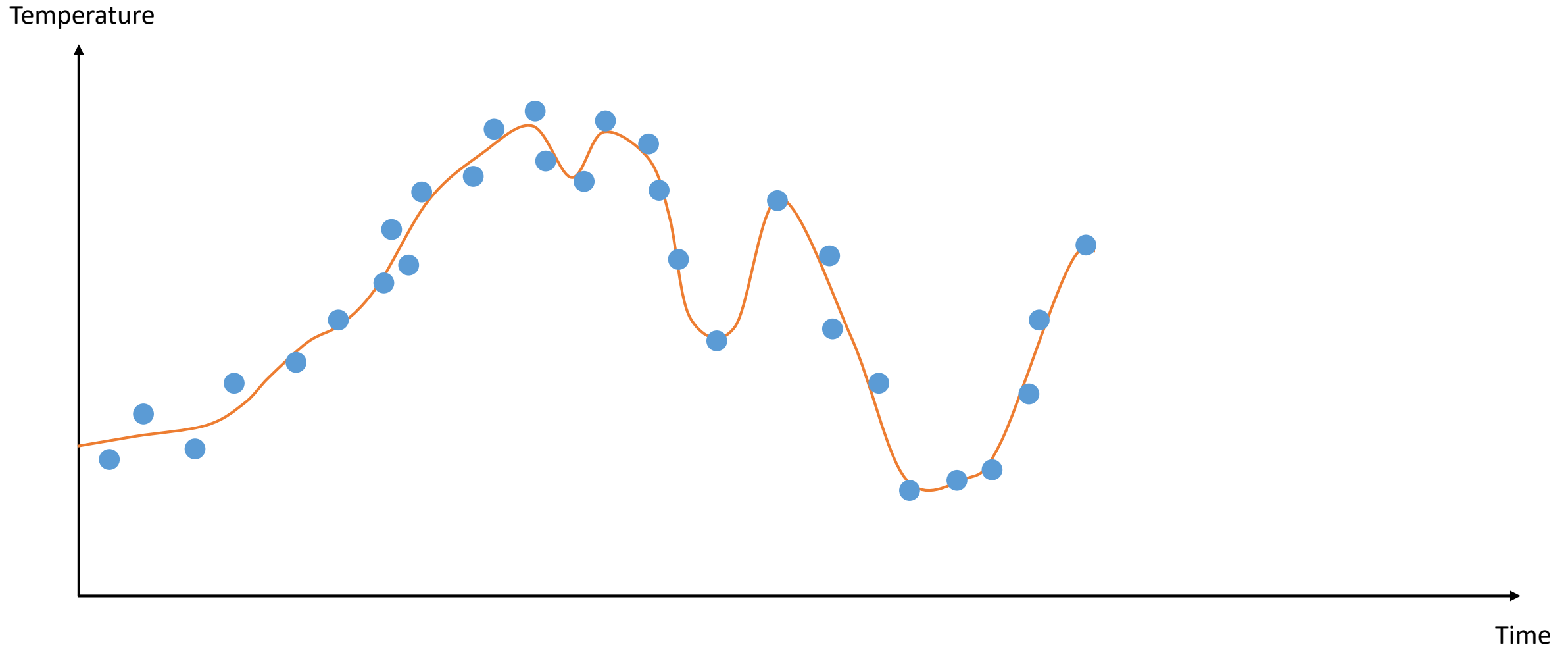
Guess the next 3 points!



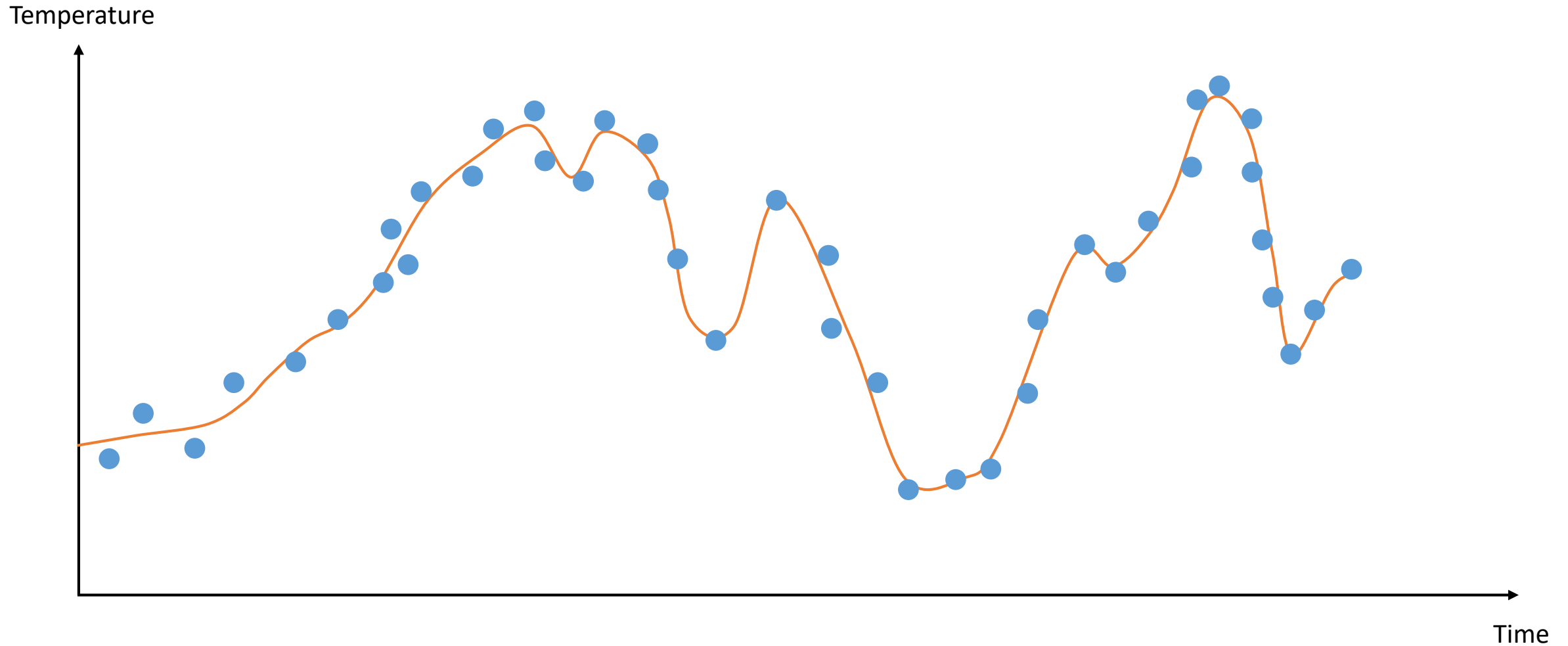
Guess the next 3 points!



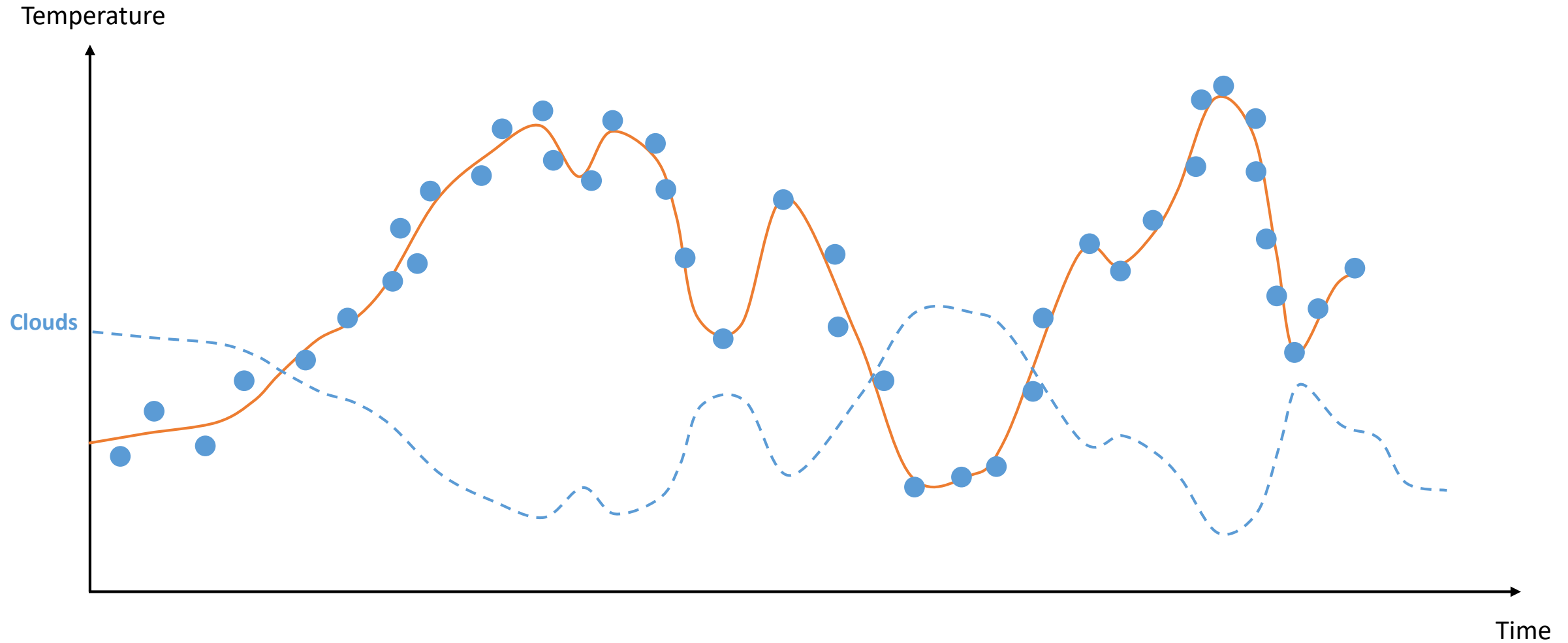
Guess the next 3 points!



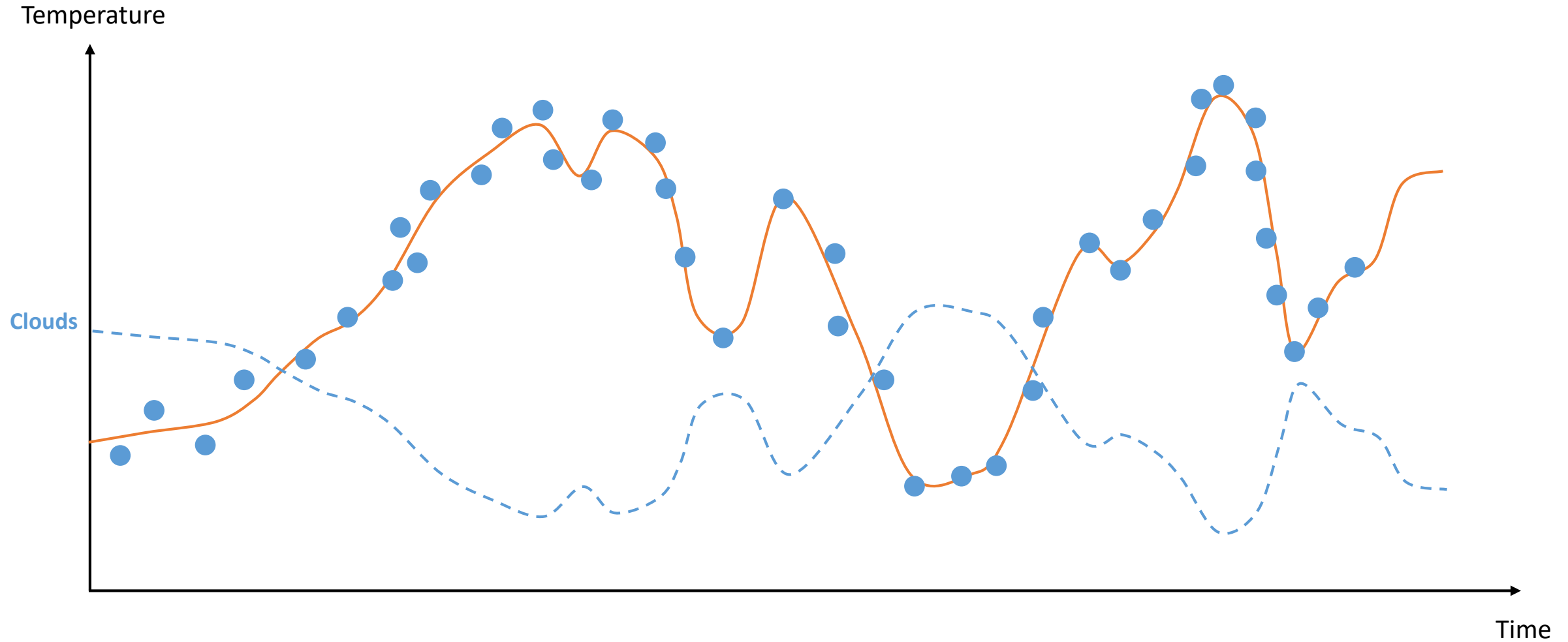
Guess the next 3 points!



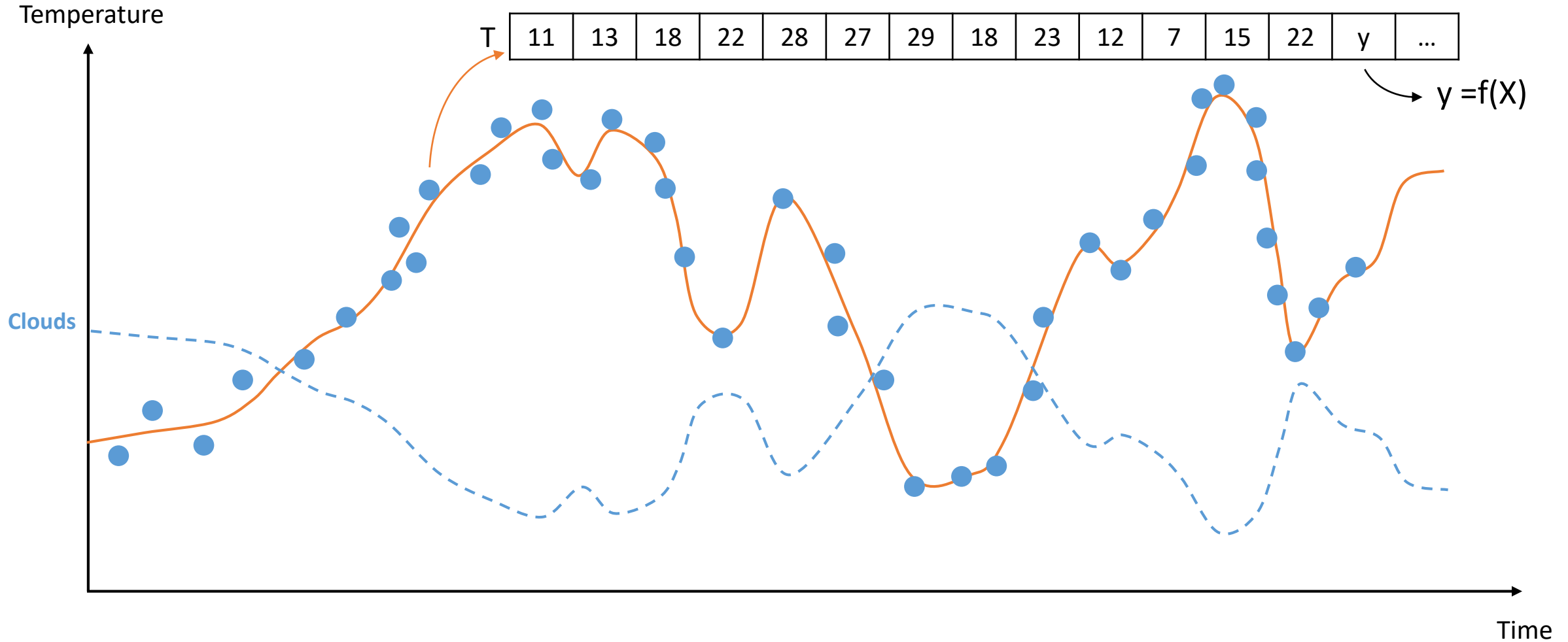
Guess the next 3 points!



Guess the next 3 points!



Guess the next 3 points!



Guess the next 3 points!

T	11	13	18	22	28	27	29	18	23	12	7	15	22	y	...
---	----	----	----	----	----	----	----	----	----	----	---	----	----	---	-----

↘ $y = f(X)$

```
int function f(X) {  
    int y;  
    // computations  
    return y;  
}
```

—————→ Prediction function

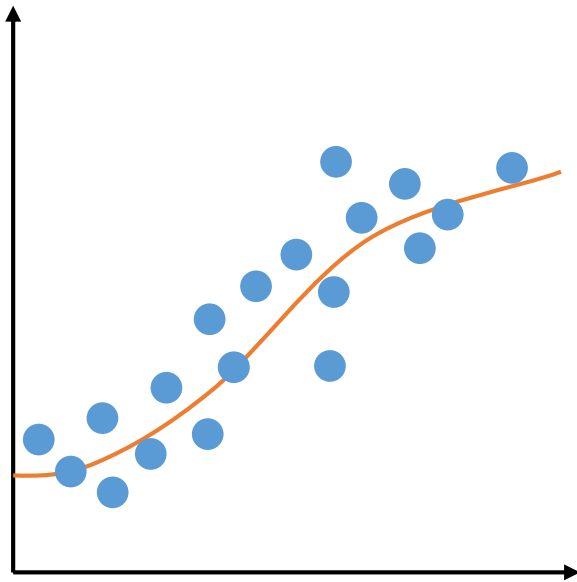
Machine learning

Is all about learning to predict

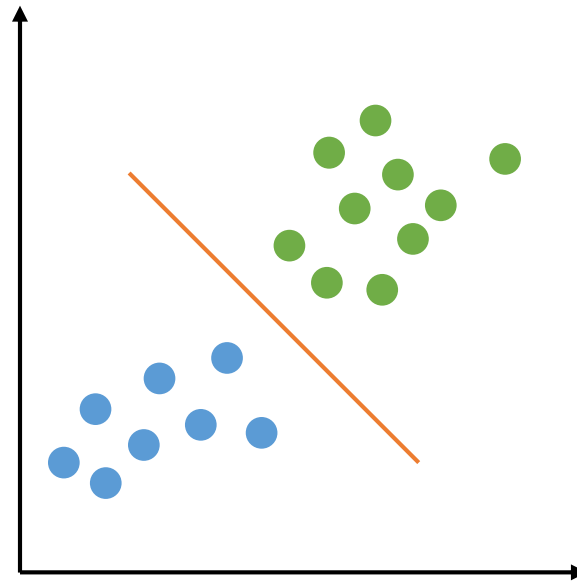
But what kind of predictions?

Machine learning

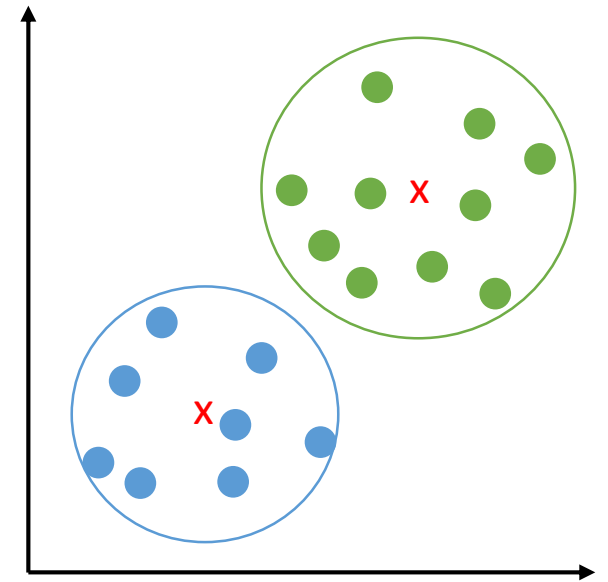
Regression



Classification

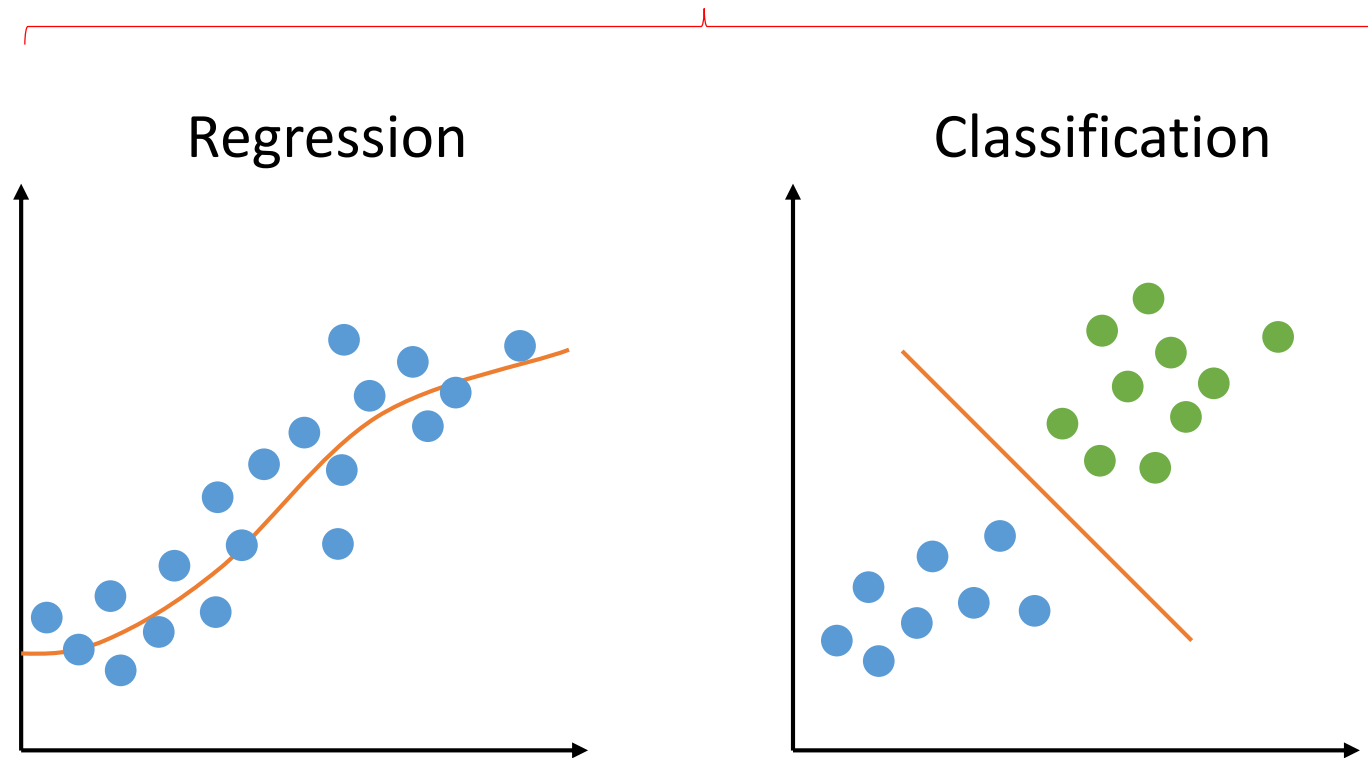


Clustering

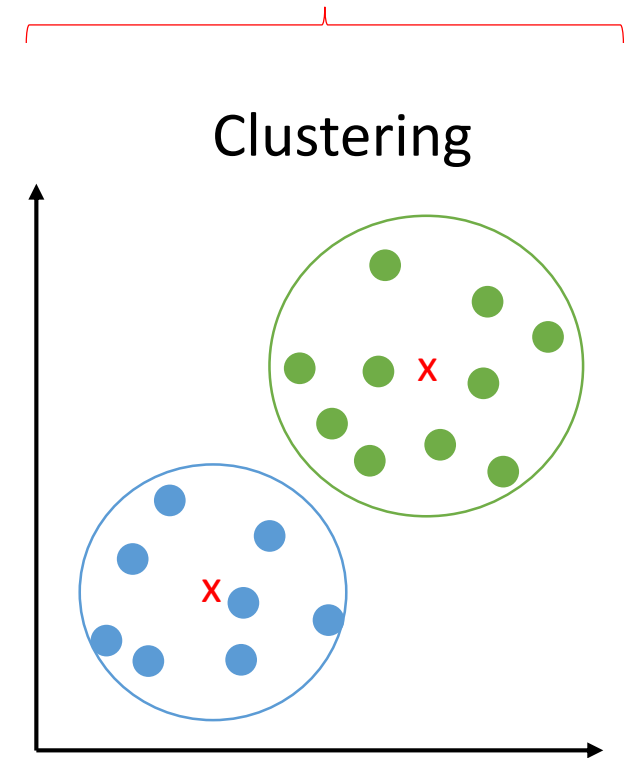


Machine learning

Supervised learning



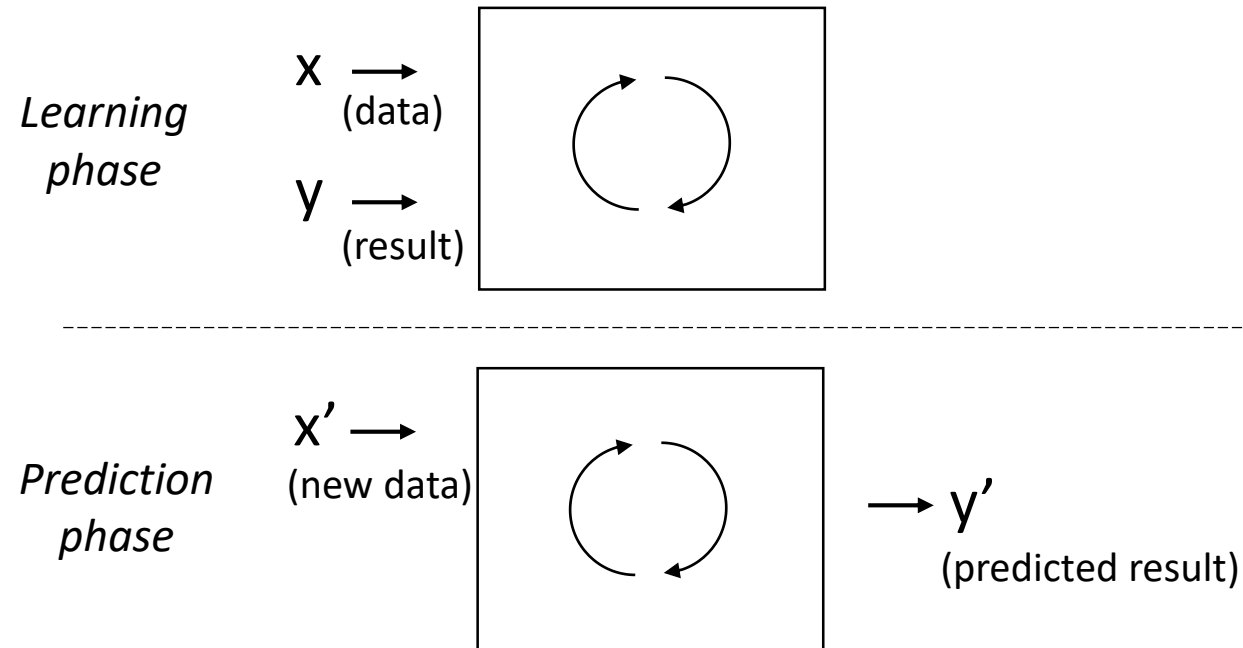
Unsupervised learning



Machine learning

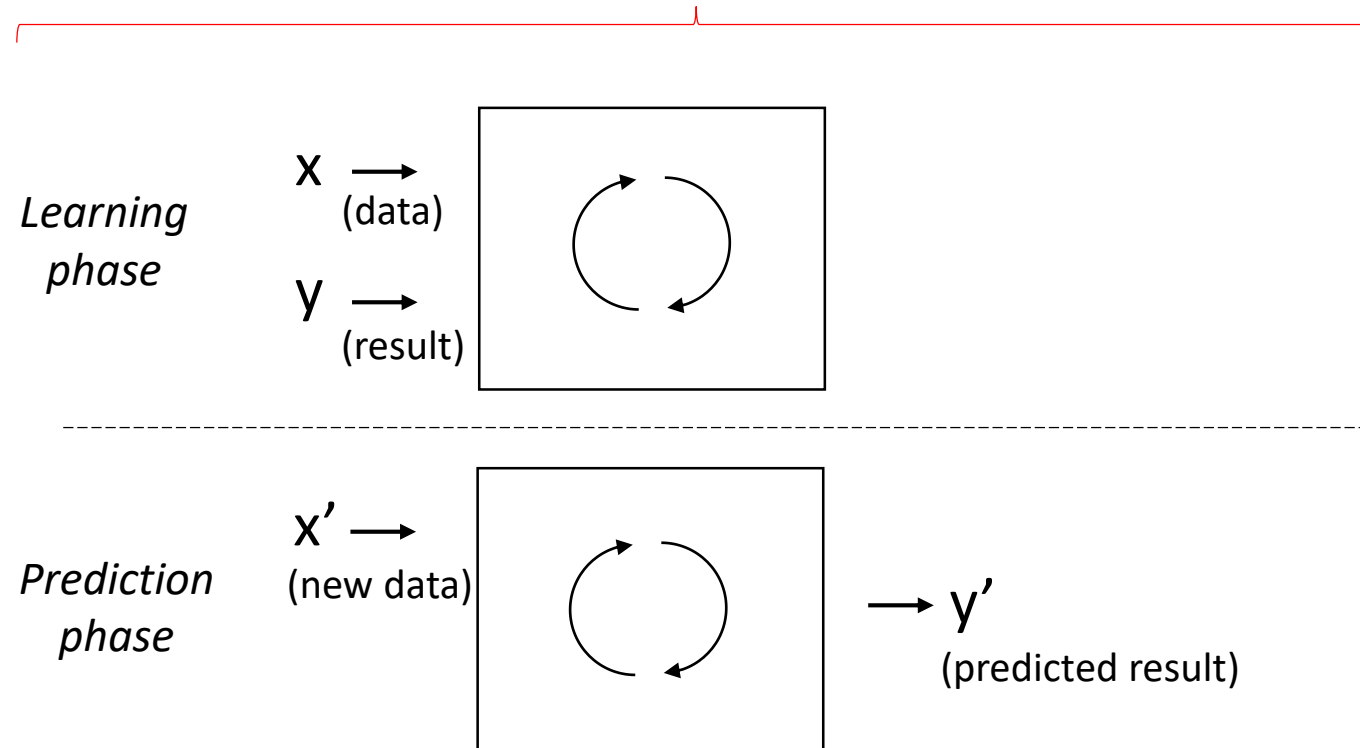
Supervised learning

Unsupervised learning

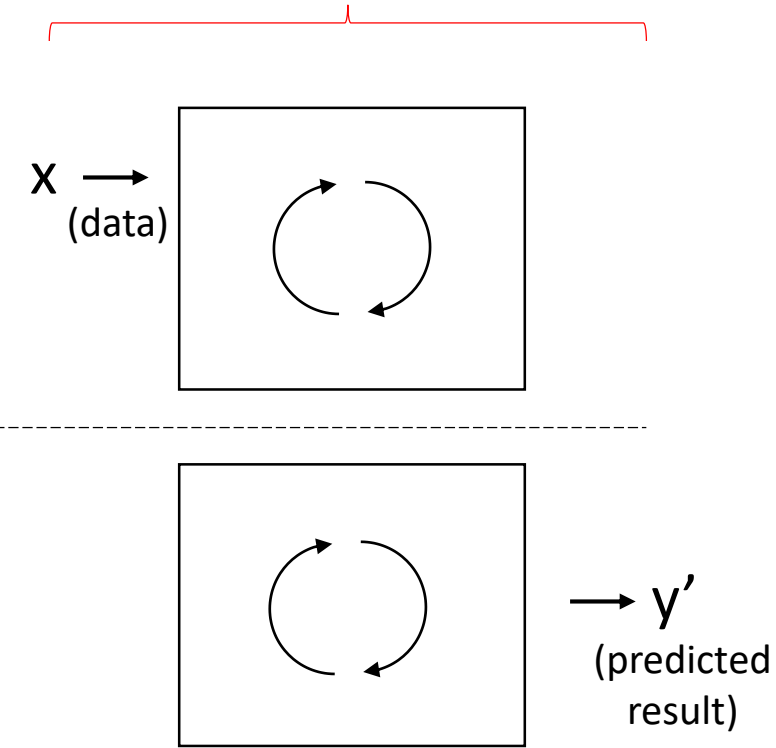


Machine learning

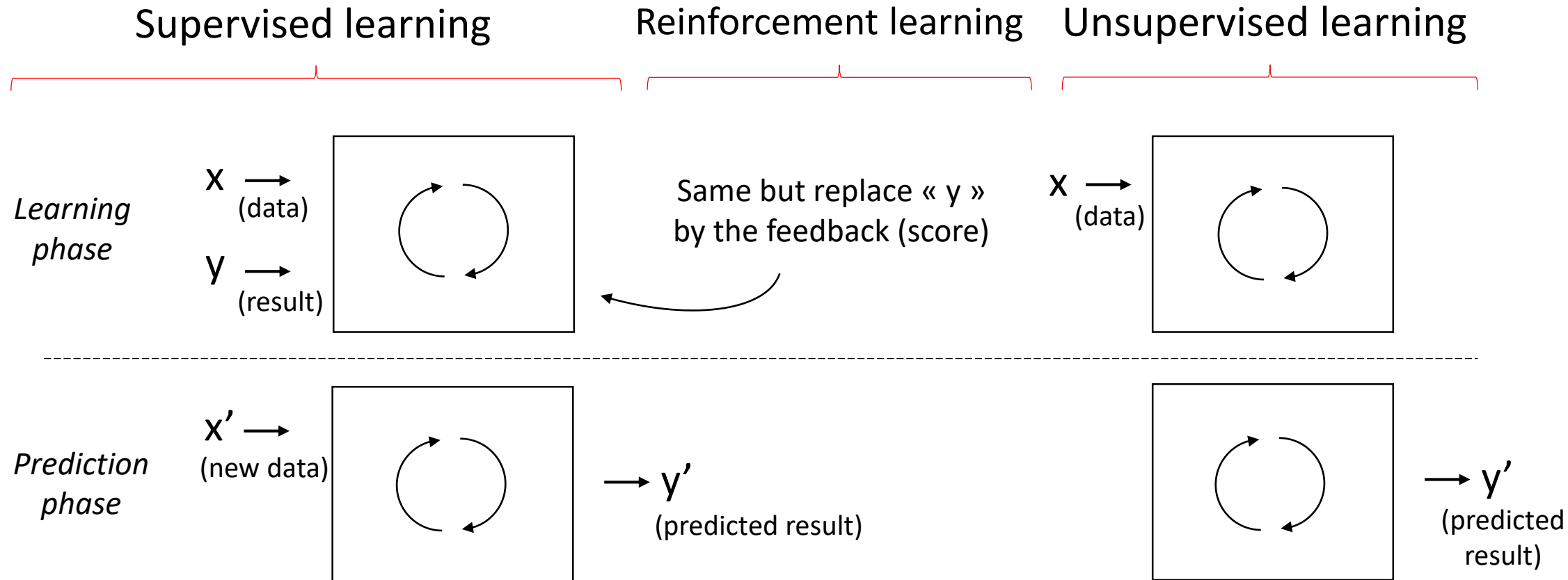
Supervised learning



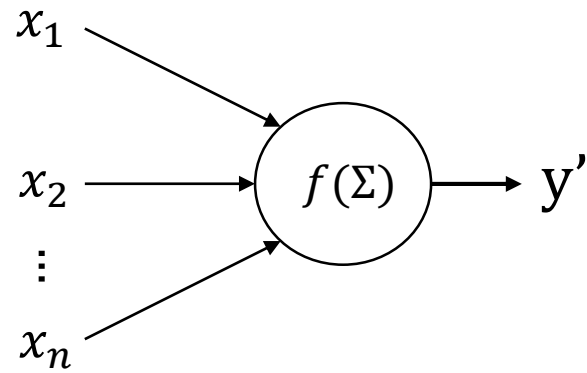
Unsupervised learning



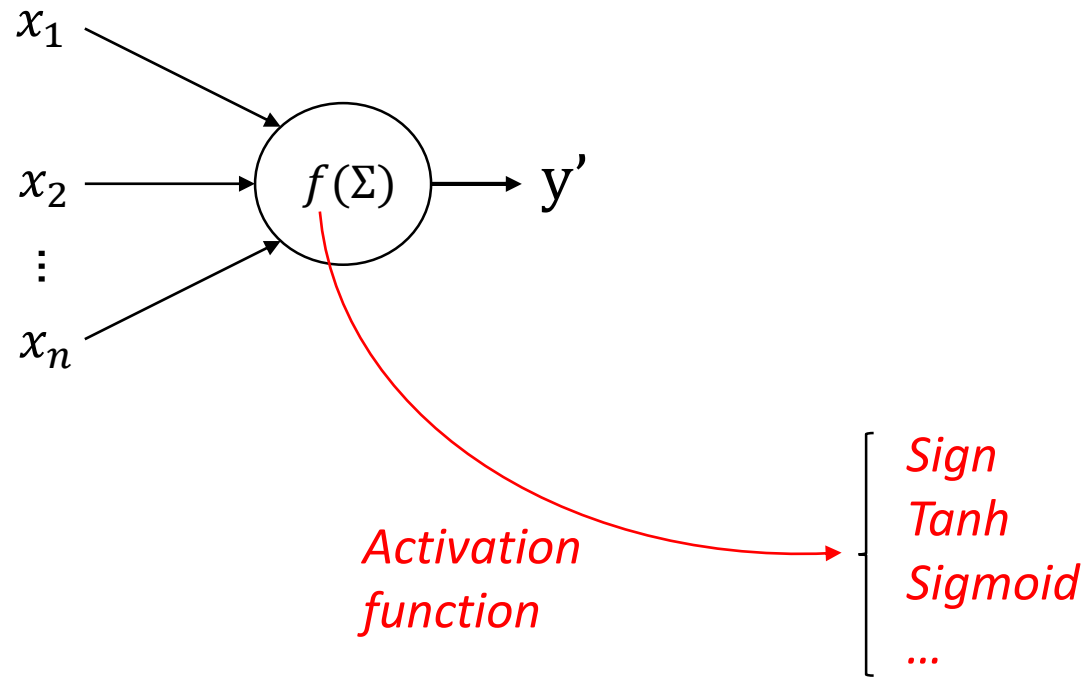
Machine learning



How to predict?

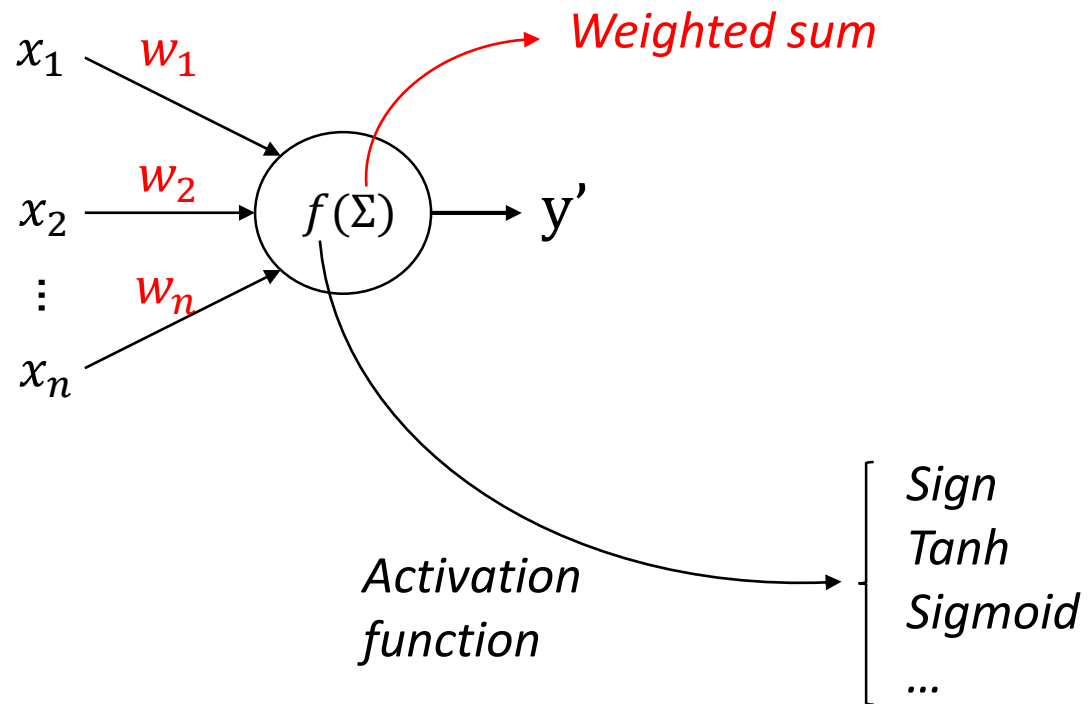


How to predict?



$$y' = f(\Sigma)$$

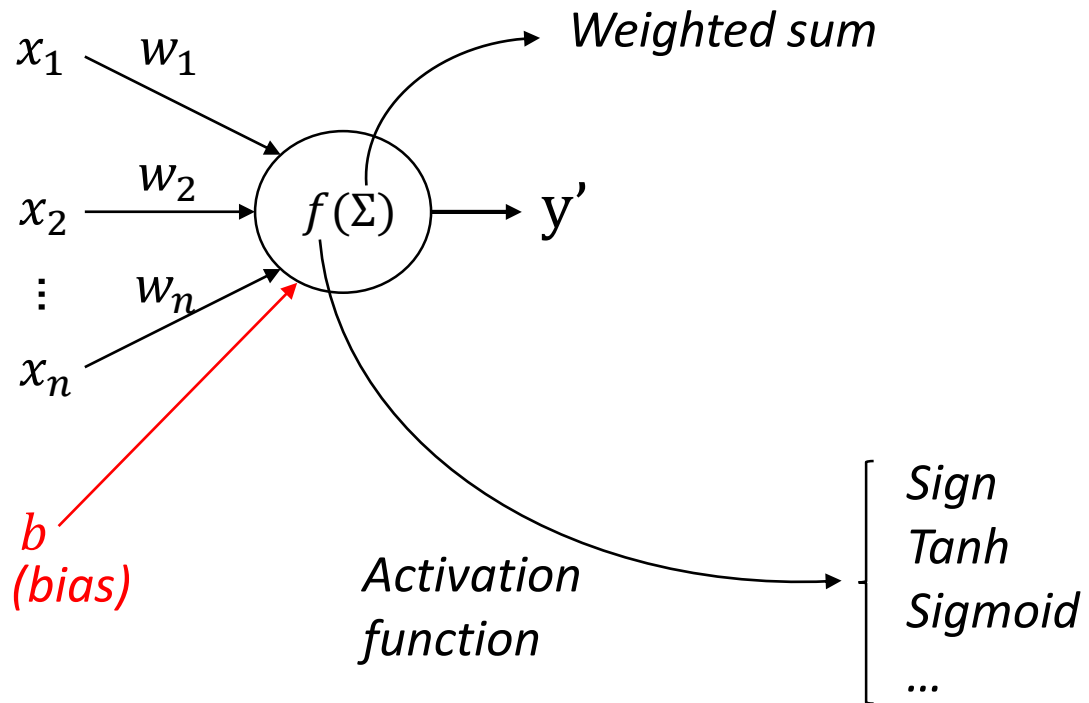
How to predict?



$$\Sigma = W^T * X$$

$$y' = f(\Sigma)$$

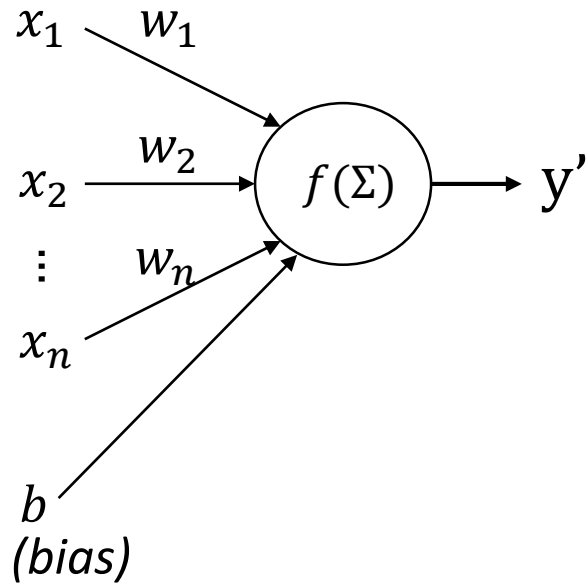
How to predict?



$$\Sigma = W^T * X + b$$

$$y' = f(\Sigma)$$

How to predict?



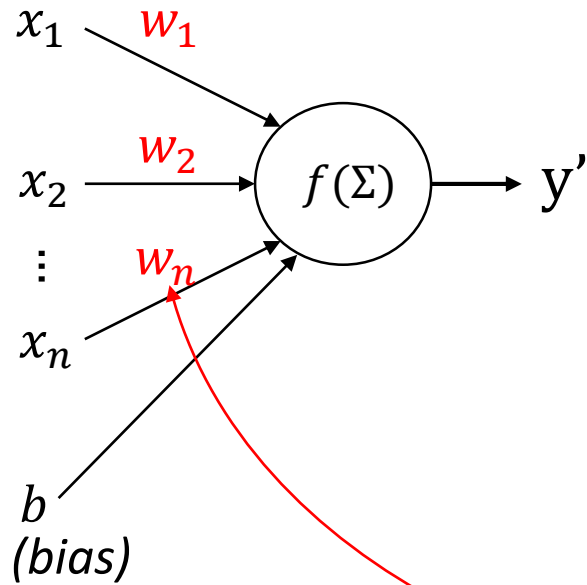
$$\Sigma = W^T * X + b$$

$$y' = f(\Sigma)$$

Perceptron (1957, F. Rosenblatt)

$$\text{Activation fct: } f(\Sigma) = \begin{cases} 1 & \text{if } \Sigma \geq 0 \\ -1 & \text{otherwise} \end{cases}$$

How to predict?



$$\Sigma = W^T * X + b$$

$$y' = f(\Sigma)$$

Perceptron (1957, F. Rosenblatt)

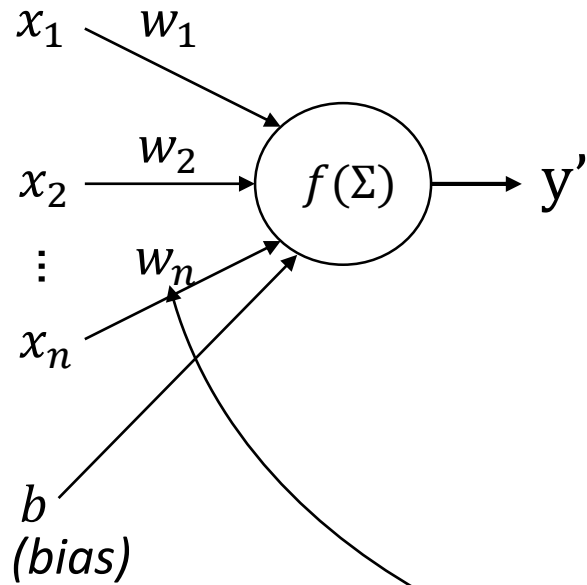
Activation fct:
$$f(\Sigma) = \begin{cases} 1 & \text{if } \Sigma \geq 0 \\ -1 & \text{otherwise} \end{cases}$$

Updating rule:
$$W^{t+1} = W^t + \eta \cdot E * X$$

$$E = y' - y$$

Learning phase

How to predict?



$$\Sigma = W^T * X + b$$

$$y' = f(\Sigma)$$

Perceptron (1957, F. Rosenblatt)

Activation fct:
$$f(\Sigma) = \begin{cases} 1 & \text{if } \Sigma \geq 0 \\ -1 & \text{otherwise} \end{cases}$$

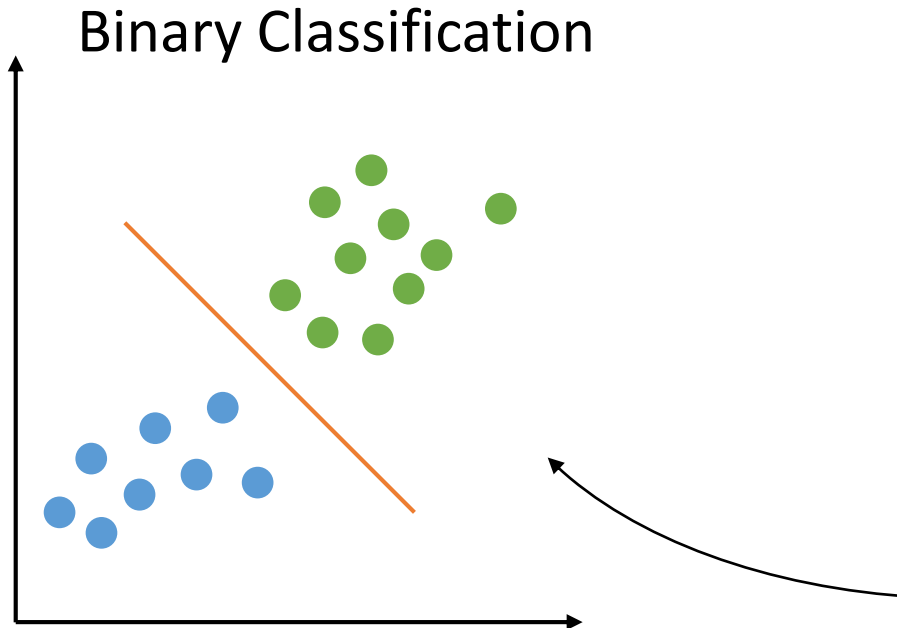
Updating rule:
$$W^{t+1} = W^t + \eta \cdot E * X$$

$$E = y' - y$$

Learning phase

Repeat until $E = 0$ (or max iter)

How to predict?



$$\Sigma = W^T * X + b$$

$$y' = f(\Sigma)$$

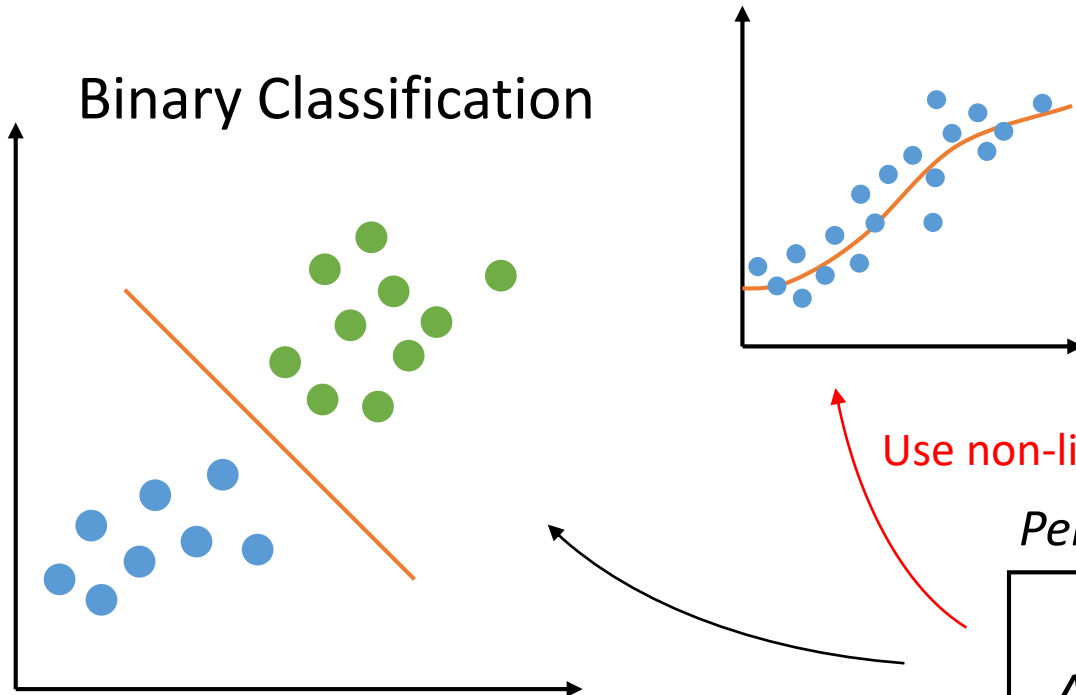
Perceptron (1957, F. Rosenblatt)

Activation fct: $f(\Sigma) = \begin{cases} 1 & \text{if } \Sigma \geq 0 \\ -1 & \text{otherwise} \end{cases}$

Updating rule: $W^{t+1} = W^t + \eta \cdot E * X$

$$E = y' - y$$

How to predict?



$$\Sigma = W^T * X + b$$

$$y' = f(\Sigma)$$

Perceptron (1957, F. Rosenblatt)

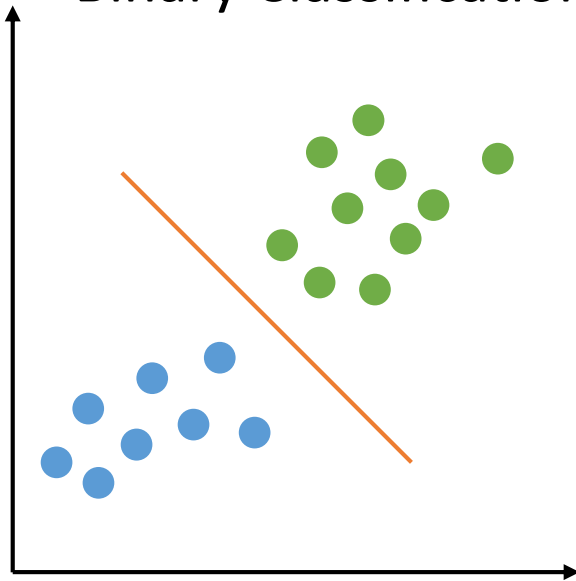
Activation fct: $f(\Sigma) = \begin{cases} 1 & \text{if } \Sigma \geq 0 \\ -1 & \text{otherwise} \end{cases}$

Updating rule: $W^{t+1} = W^t + \eta \cdot E * X$

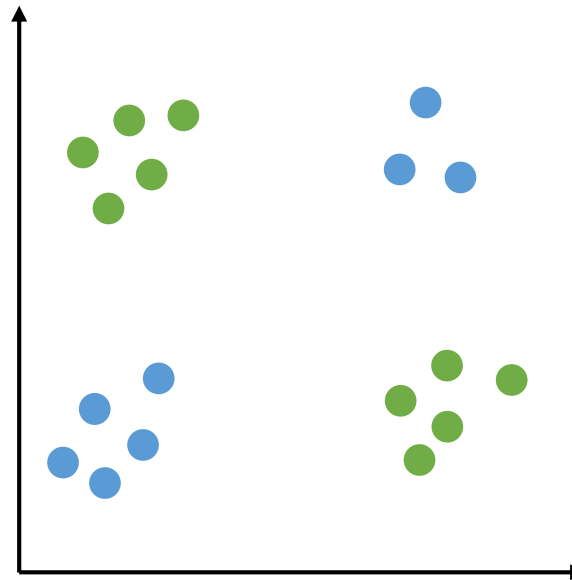
$$E = y' - y$$

What if?

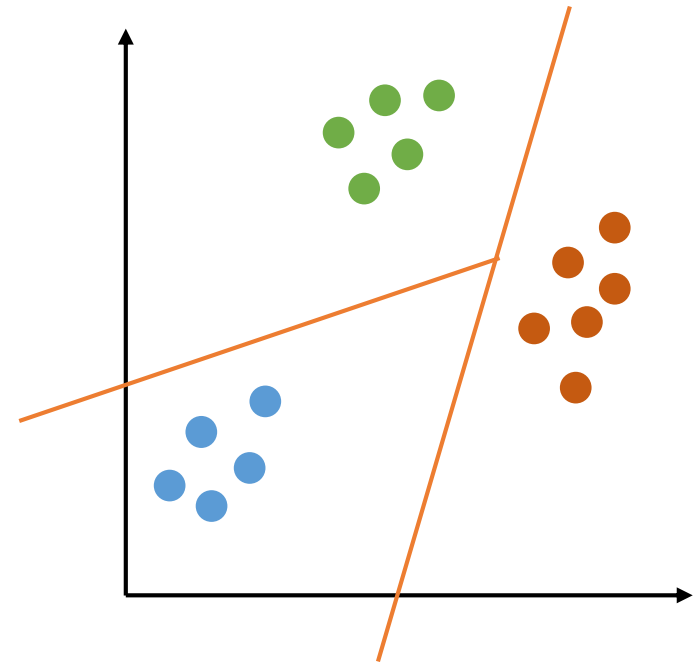
Binary Classification



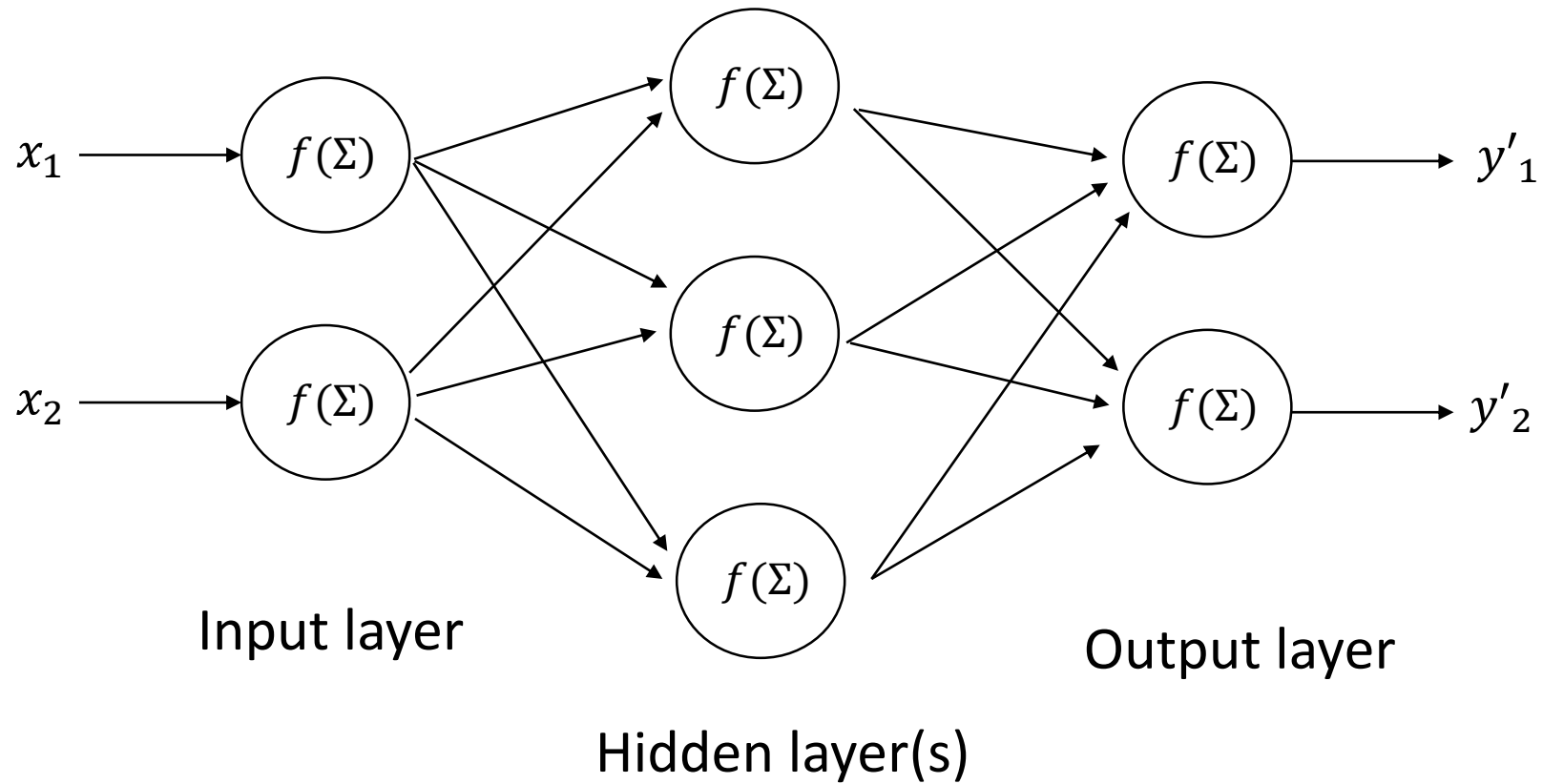
Non linear classification



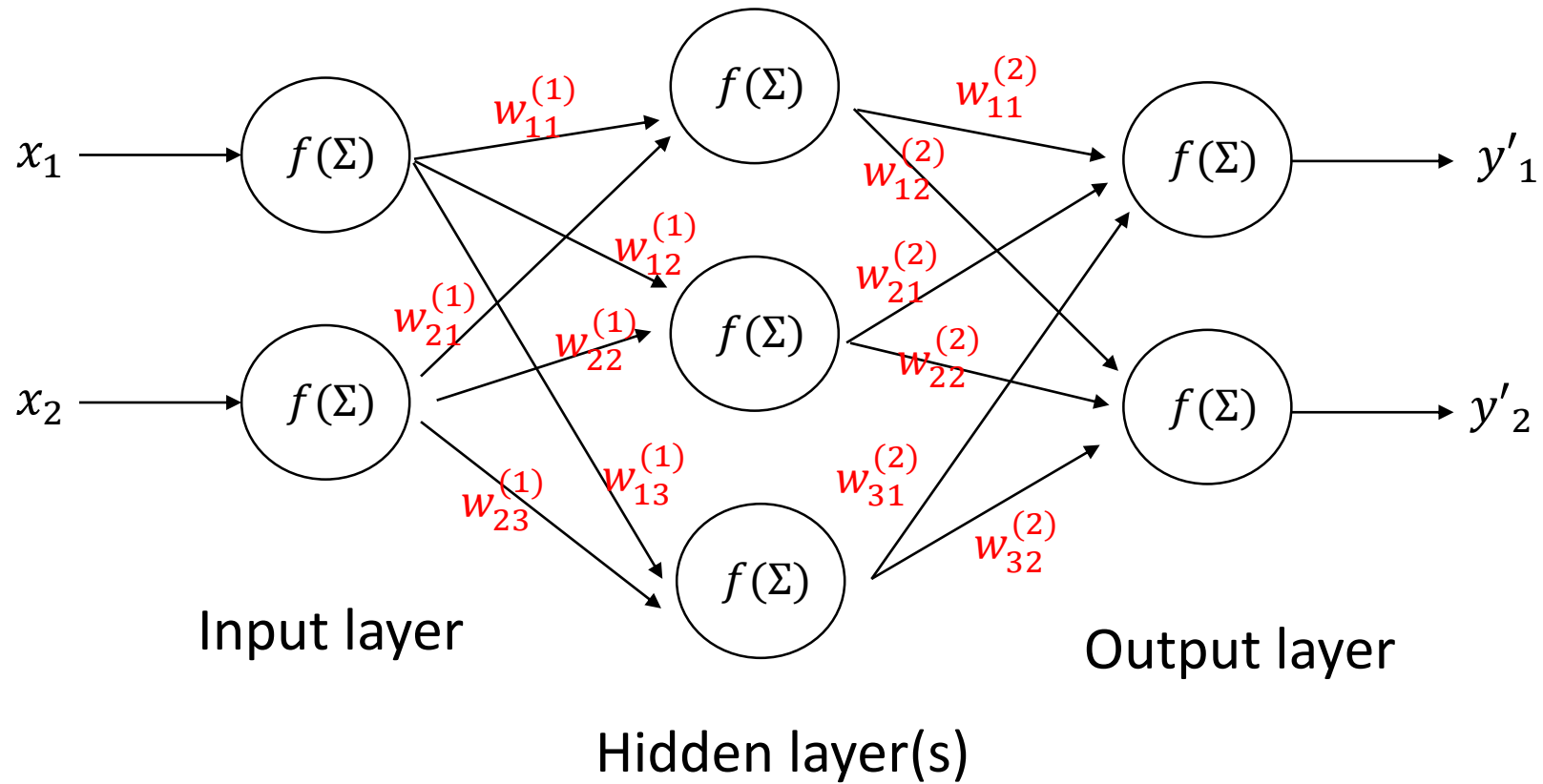
Multi-class classification



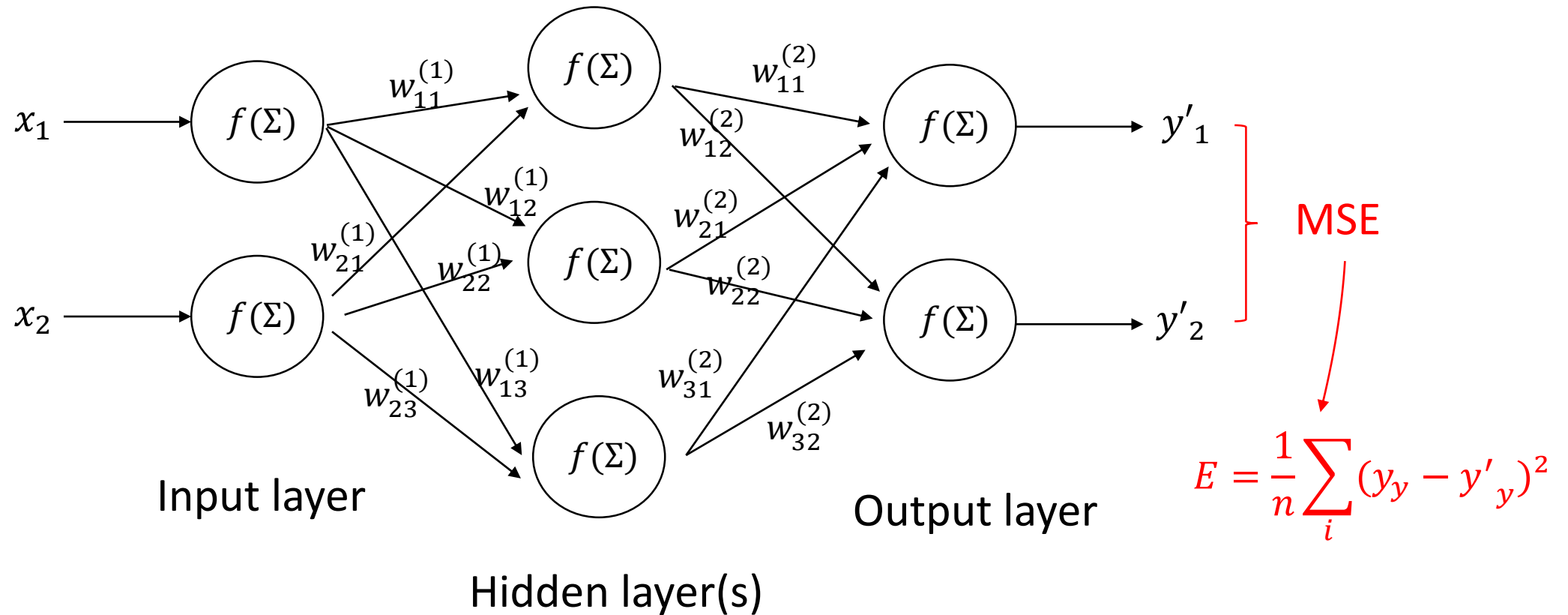
Multi-layer perceptron (MLP)



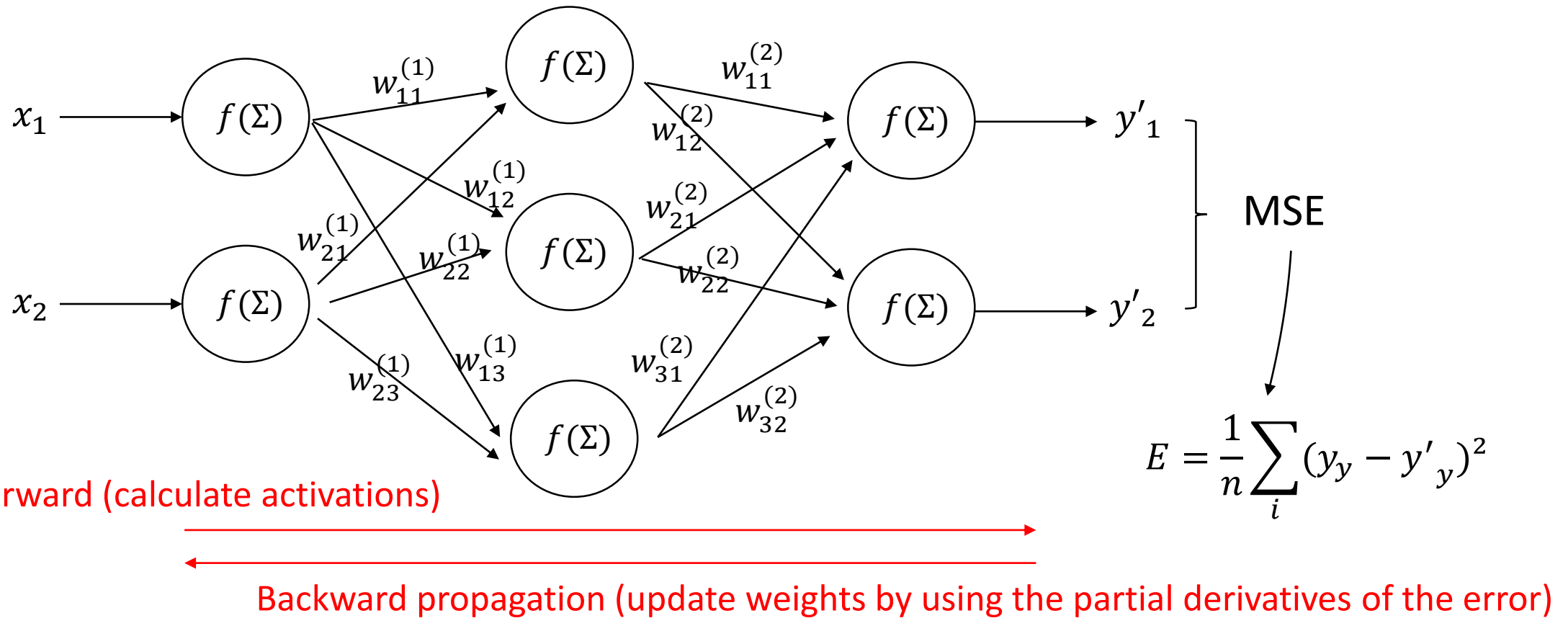
Multi-layer perceptron (MLP)



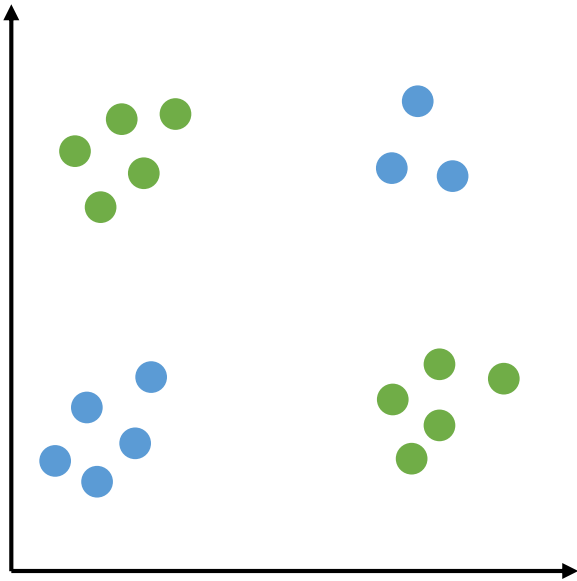
Multi-layer perceptron (MLP)



Multi-layer perceptron (MLP)

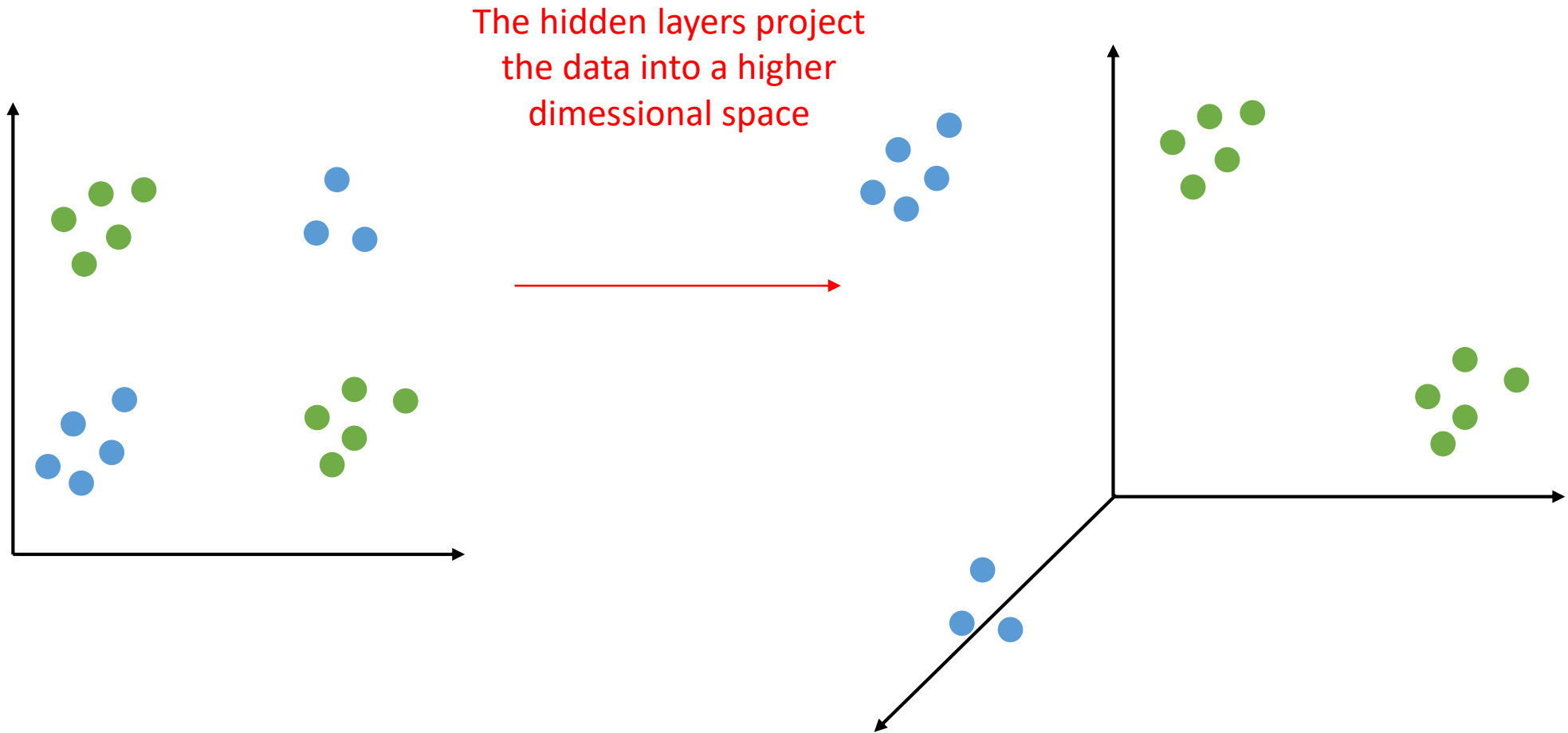


Multi-layer perceptron (MLP)

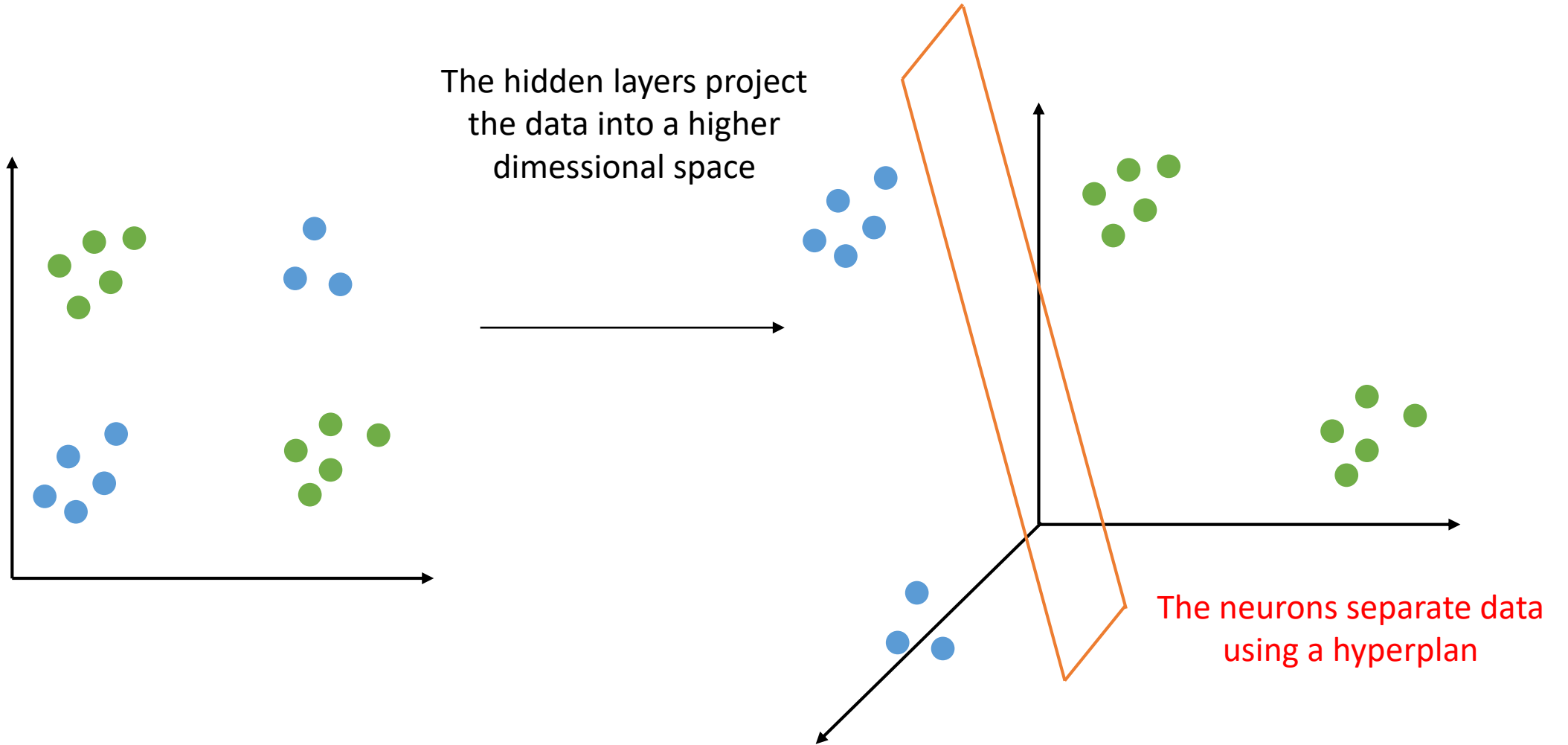


What's really
happening inside
the MLP?

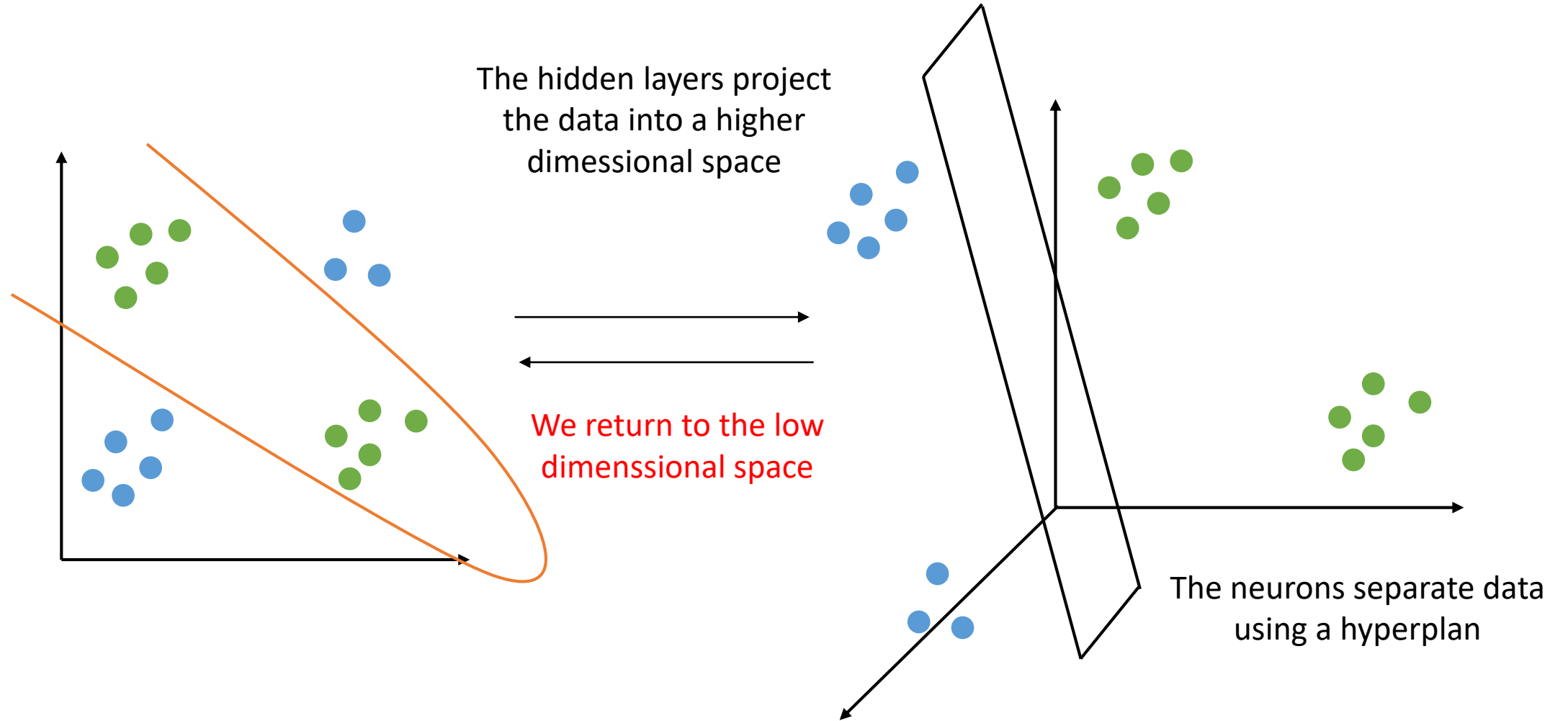
Multi-layer perceptron (MLP)



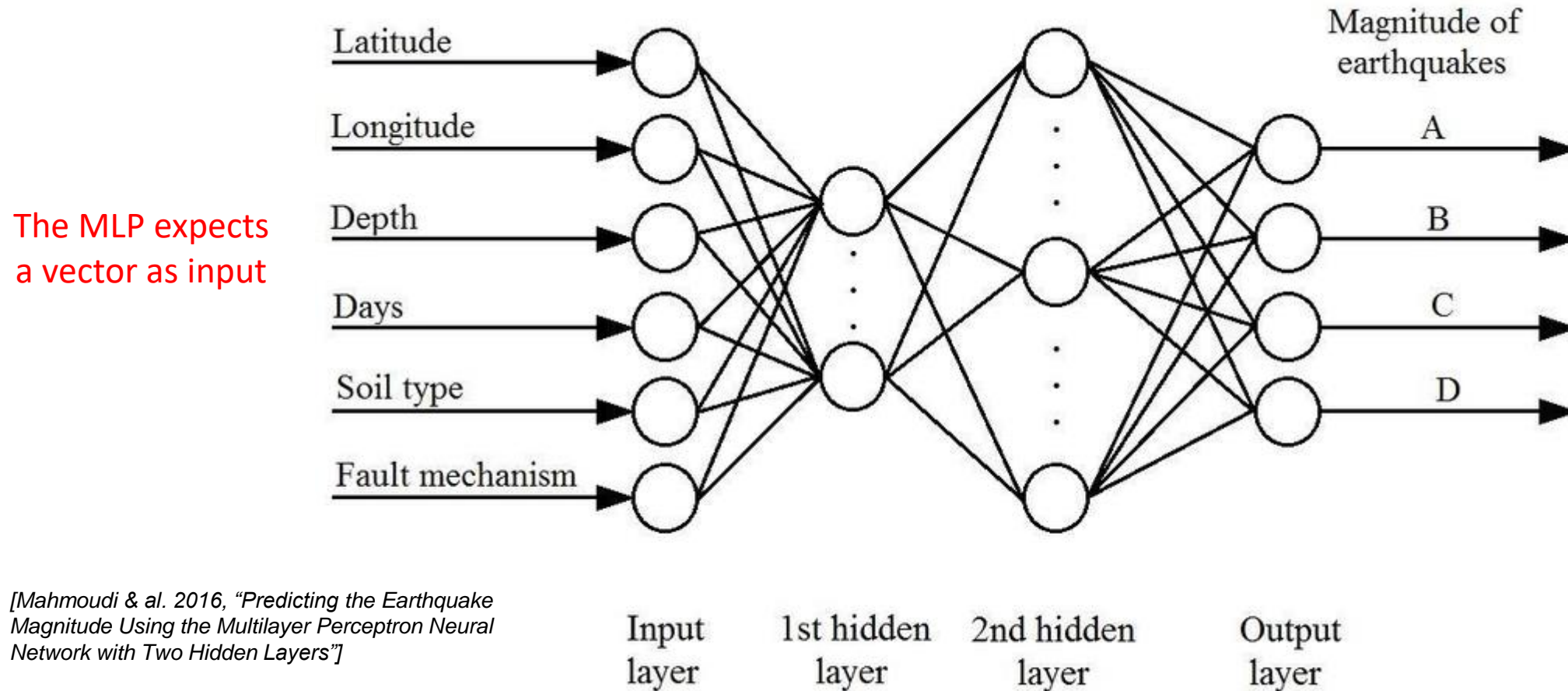
Multi-layer perceptron (MLP)



Multi-layer perceptron (MLP)

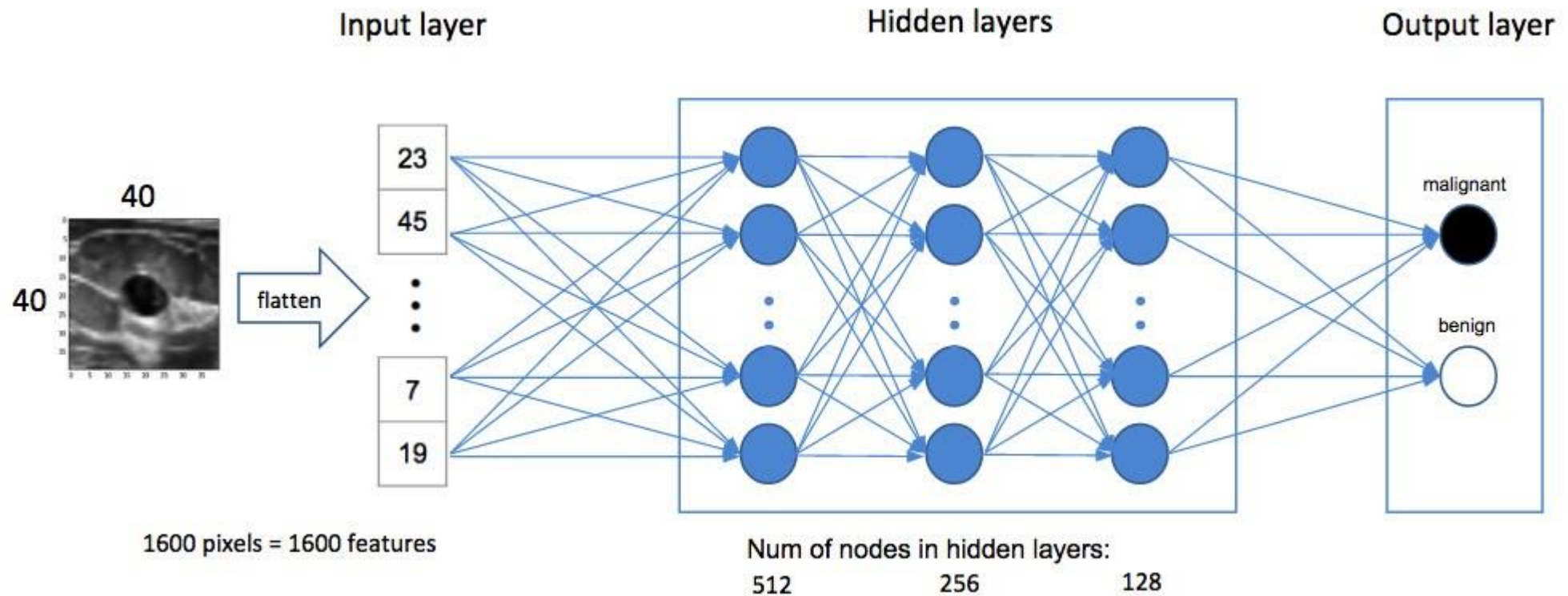


A preprocessing problem



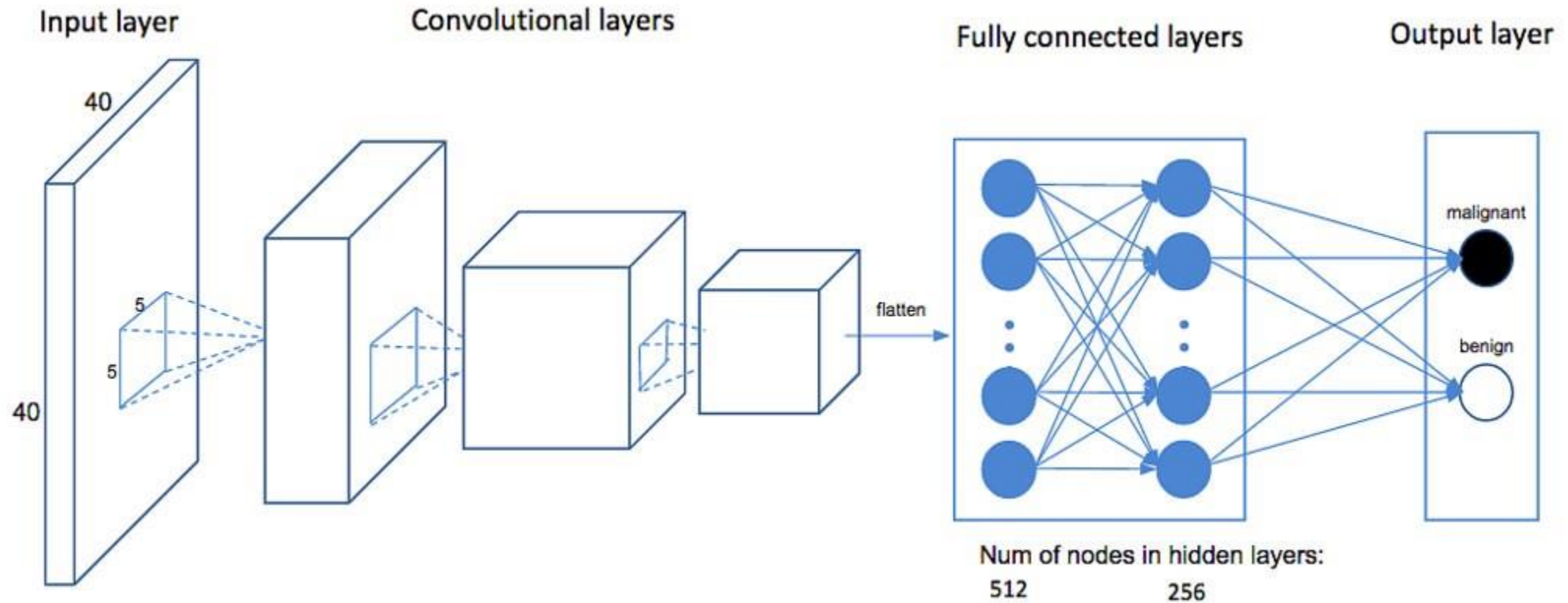
[Mahmoudi & al. 2016, "Predicting the Earthquake Magnitude Using the Multilayer Perceptron Neural Network with Two Hidden Layers"]

A preprocessing problem



Source: <https://blog.insightdatascience.com/automating-breast-cancer-detection-with-deep-learning-d8b49da17950>

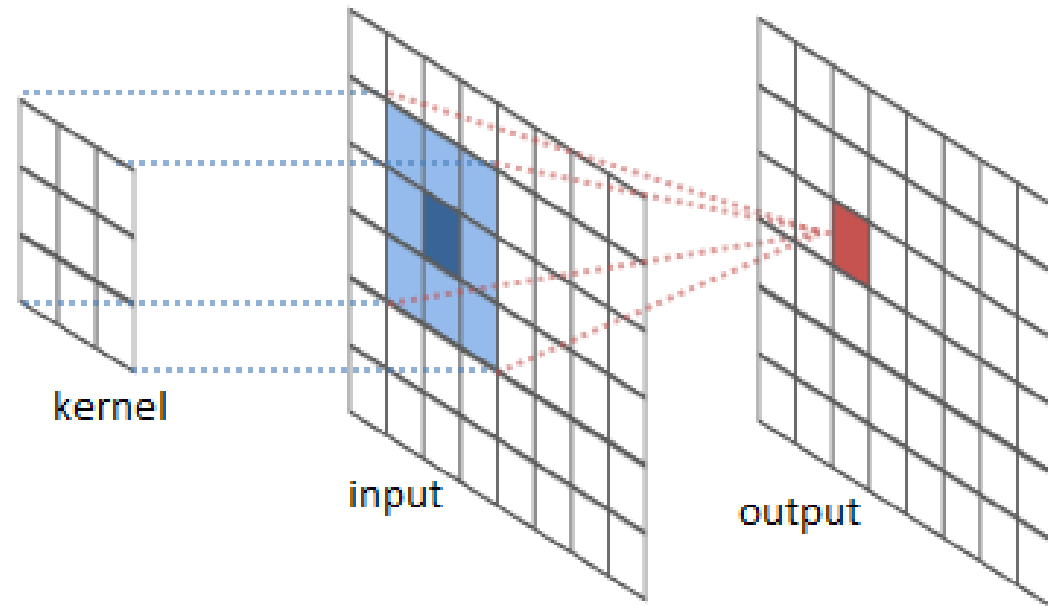
Convolutional Neural Networks (CNN)



Source: <https://blog.insightdatascience.com/automating-breast-cancer-detection-with-deep-learning-d8b49da17950>

Convolutional Neural Networks (CNN)

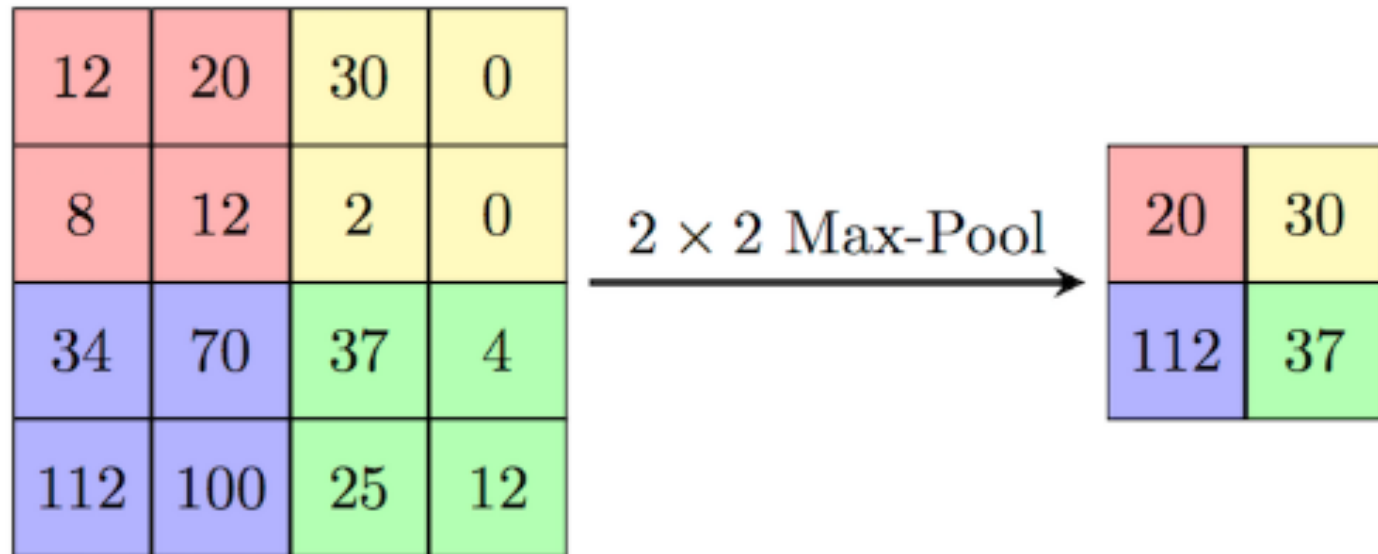
Convolution with a
3x3 filter (kernel)



Source: <http://intellabs.github.io/RiverTrail/tutorial/>

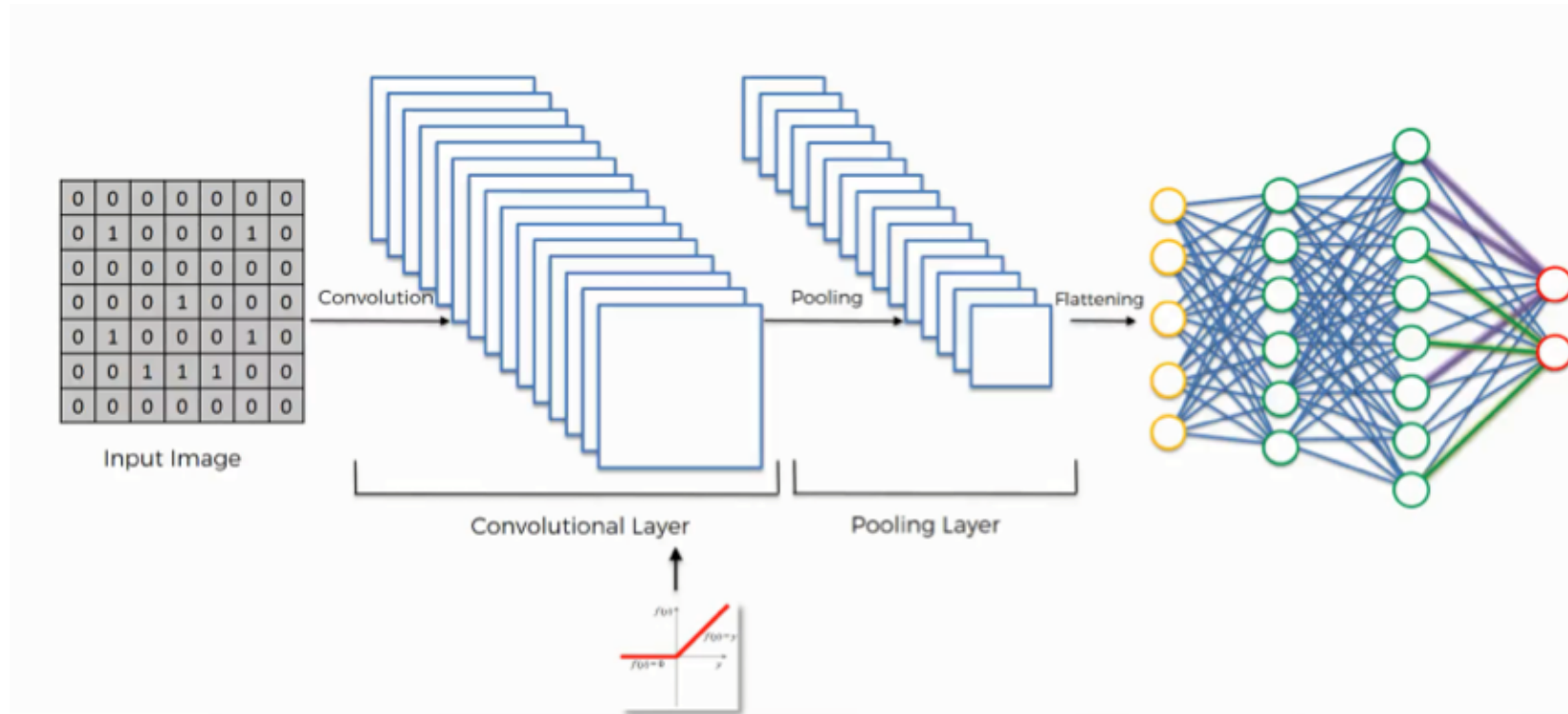
Convolutional Neural Networks (CNN)

Maxpooling with a
2x2 filter



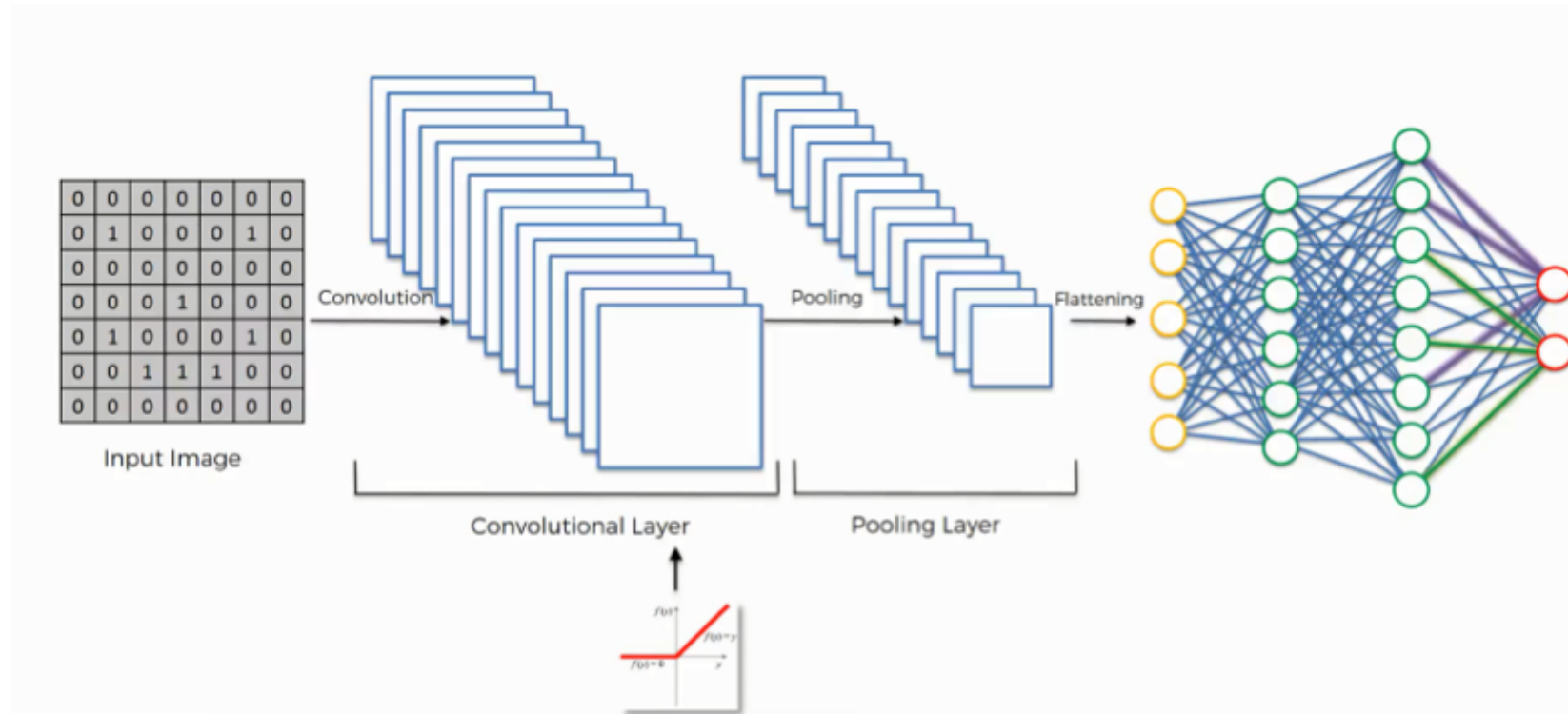
Source: <https://computersciencewiki.org/index.php/File:MaxpoolSample2.png>

Convolutional Neural Networks (CNN)



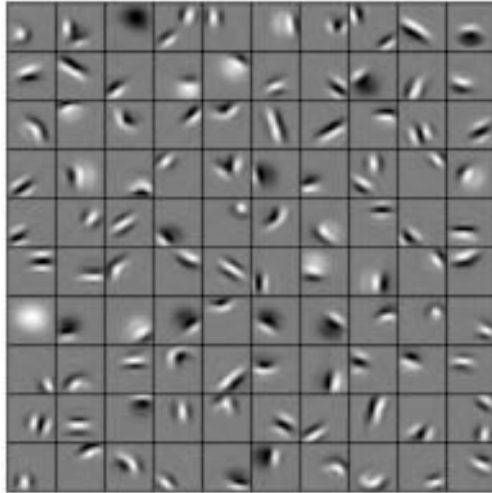
Source: <https://www.superdatascience.com/convolutional-neural-networks-cnn-summary>

Convolutional Neural Networks (CNN)



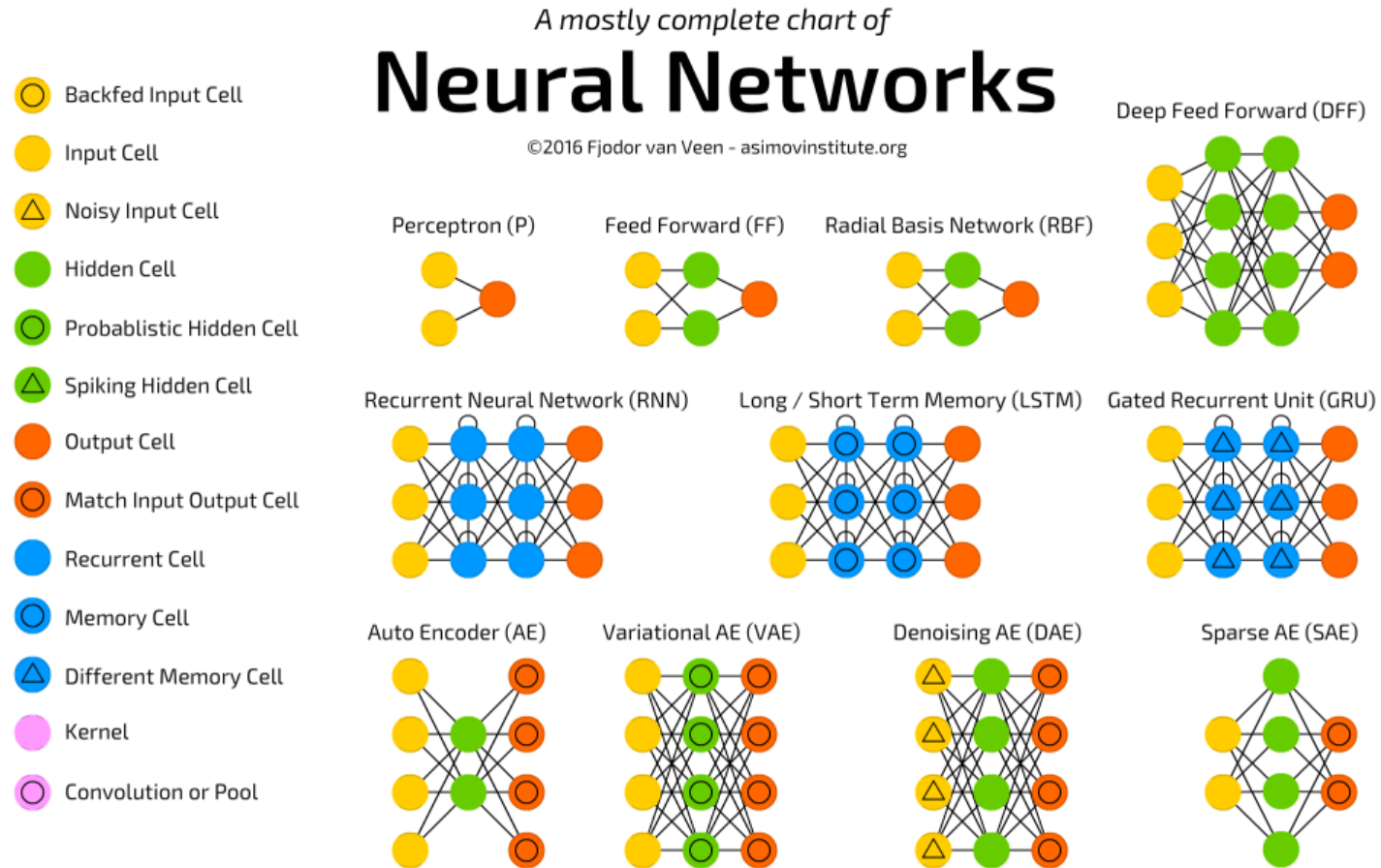
Let the learning algorithm learn the best kernels!

An example of features learned by each layer

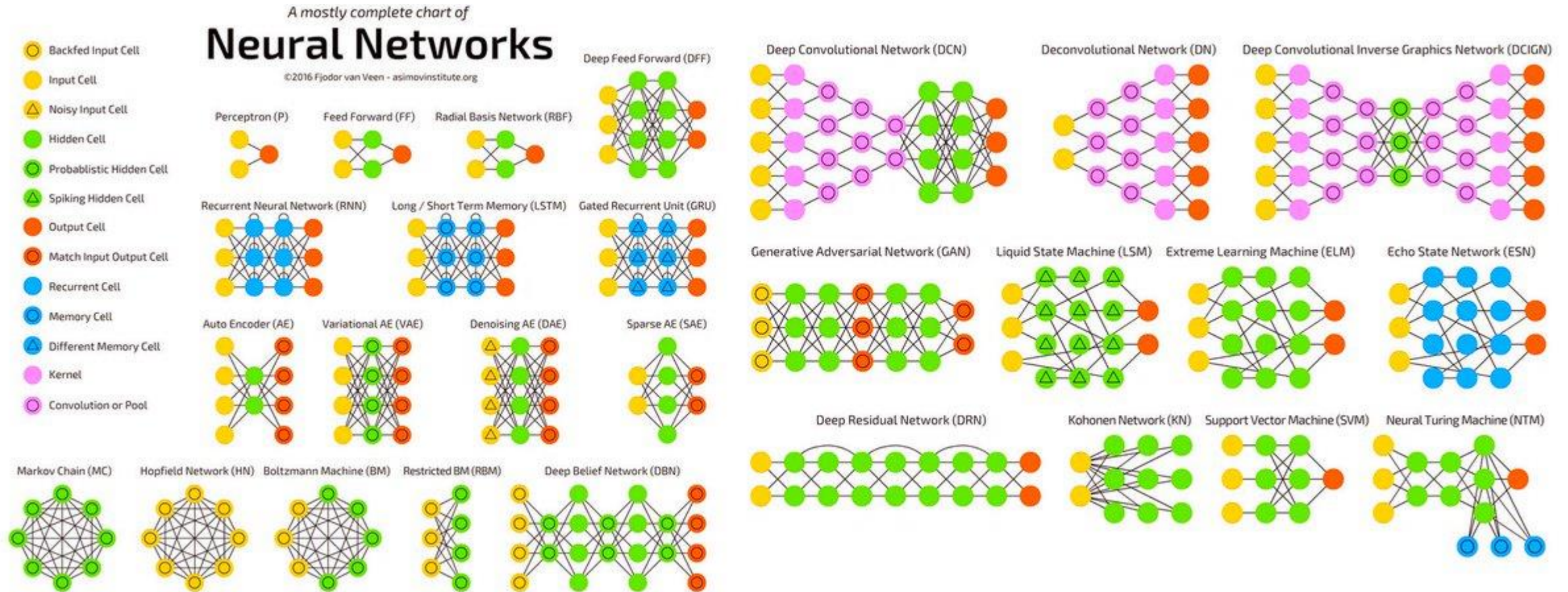


Source: [Honglak Lee, *et al*, "Convolutional Deep Belief Networks for Scalable Unsupervised Learning of Hierarchical Representations"]

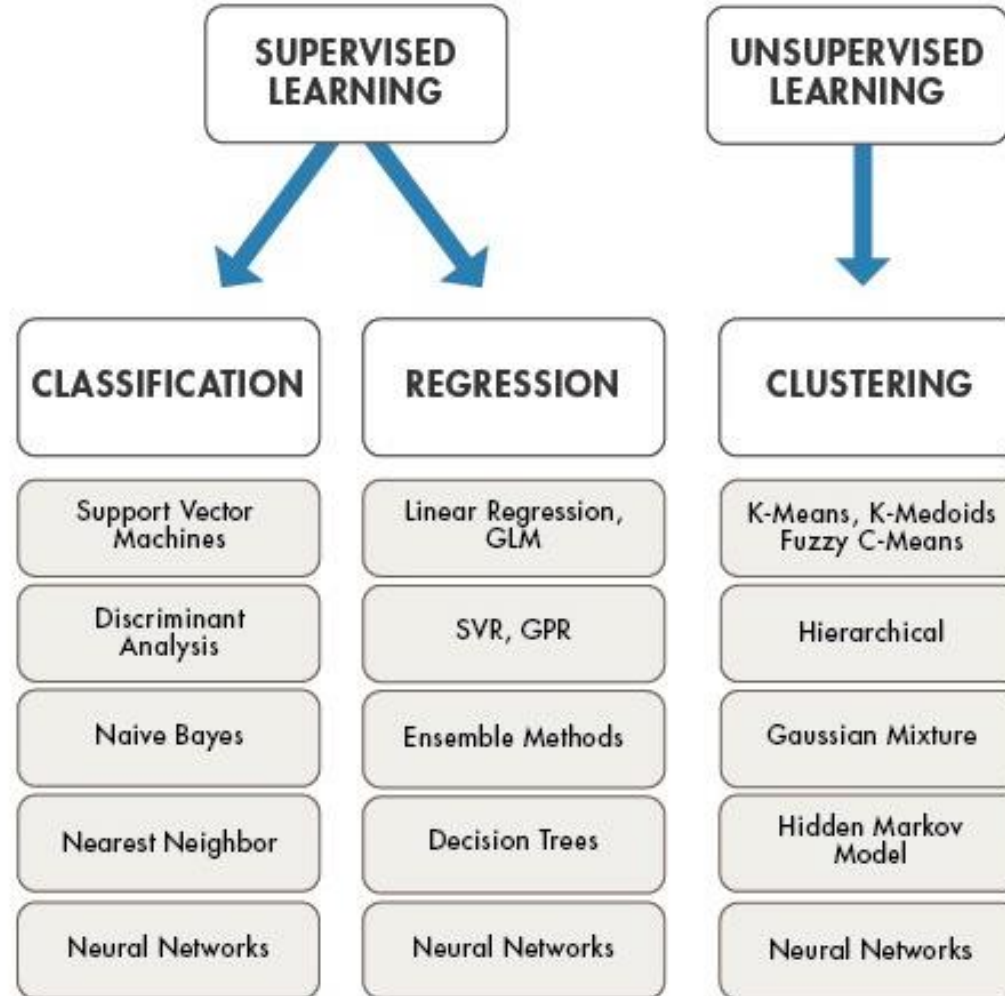
CNN, what else?



CNN, what else?



CNN, what else?



Thank you!



<https://github.com/amineHorseman/my-talks>



bendahmane.amine@gmail.com



AmineHorseman