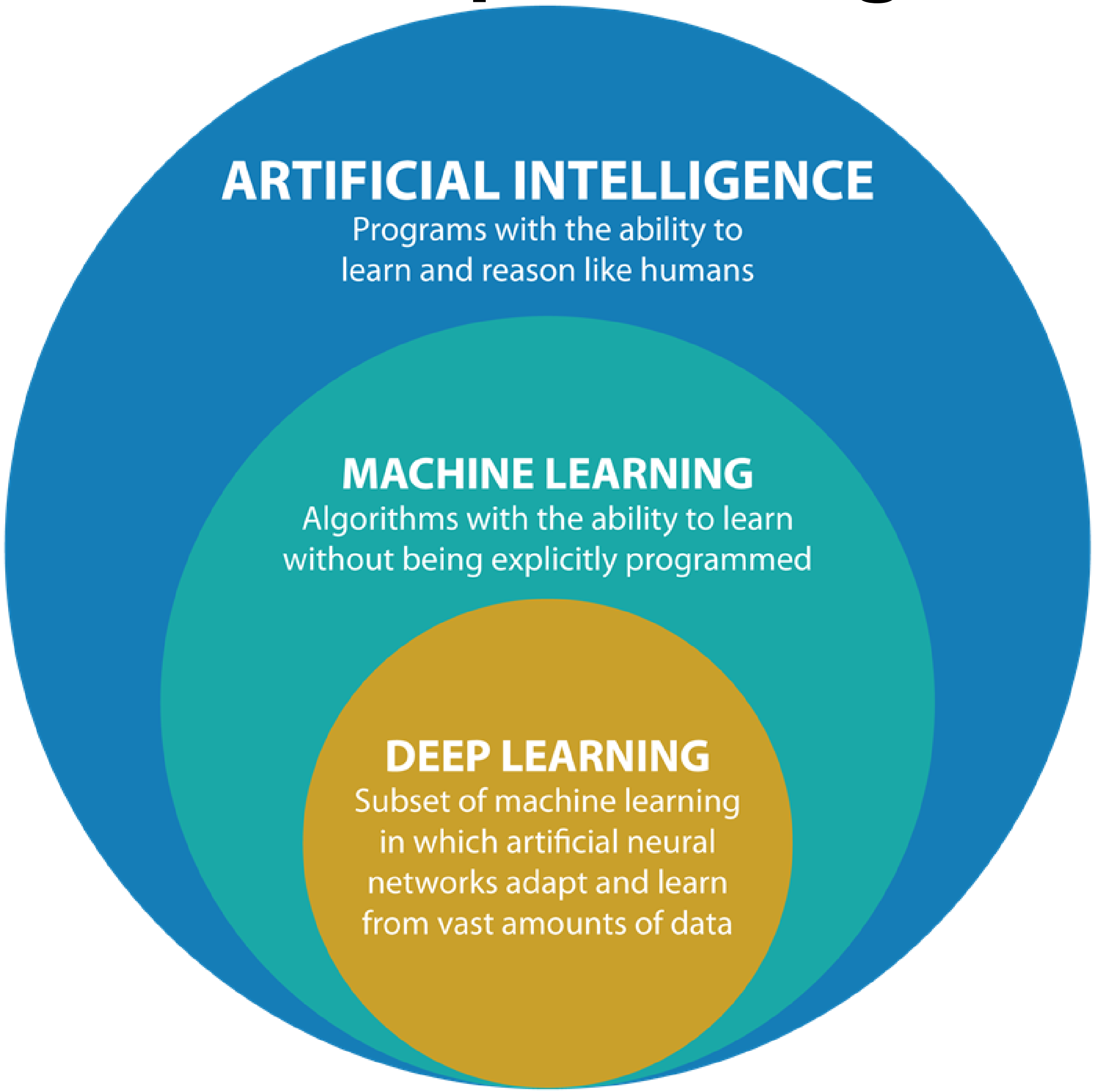
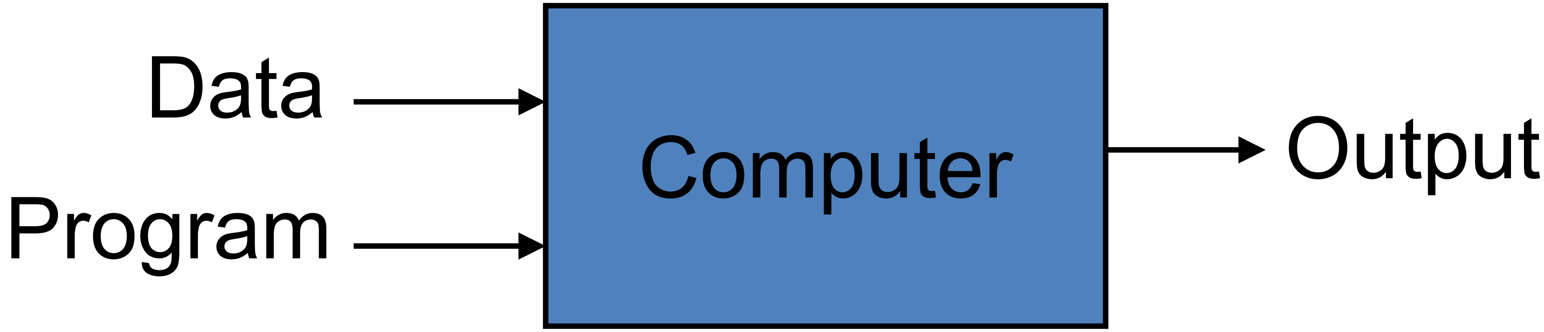
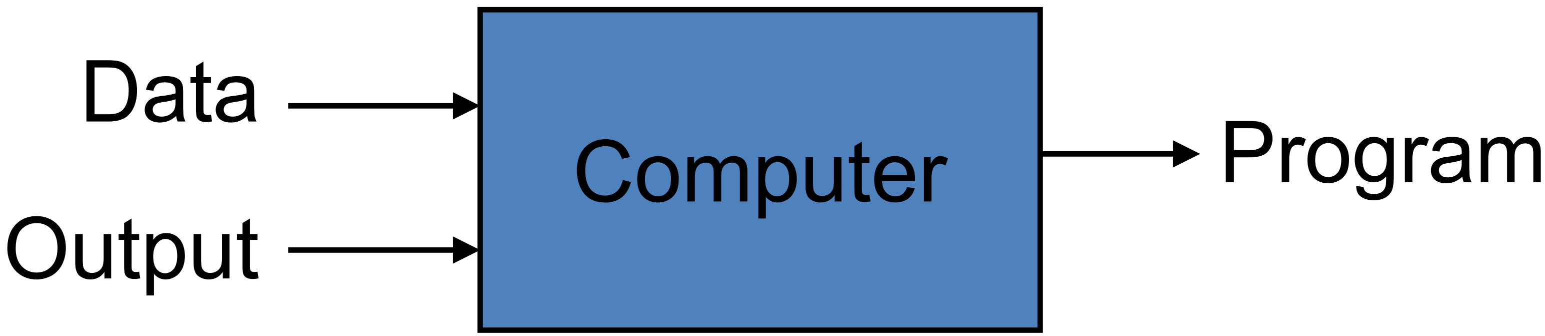
Classification, Machine learning et Deep learning



Traditional Programming



Machine Learning



Machine Learning Problems

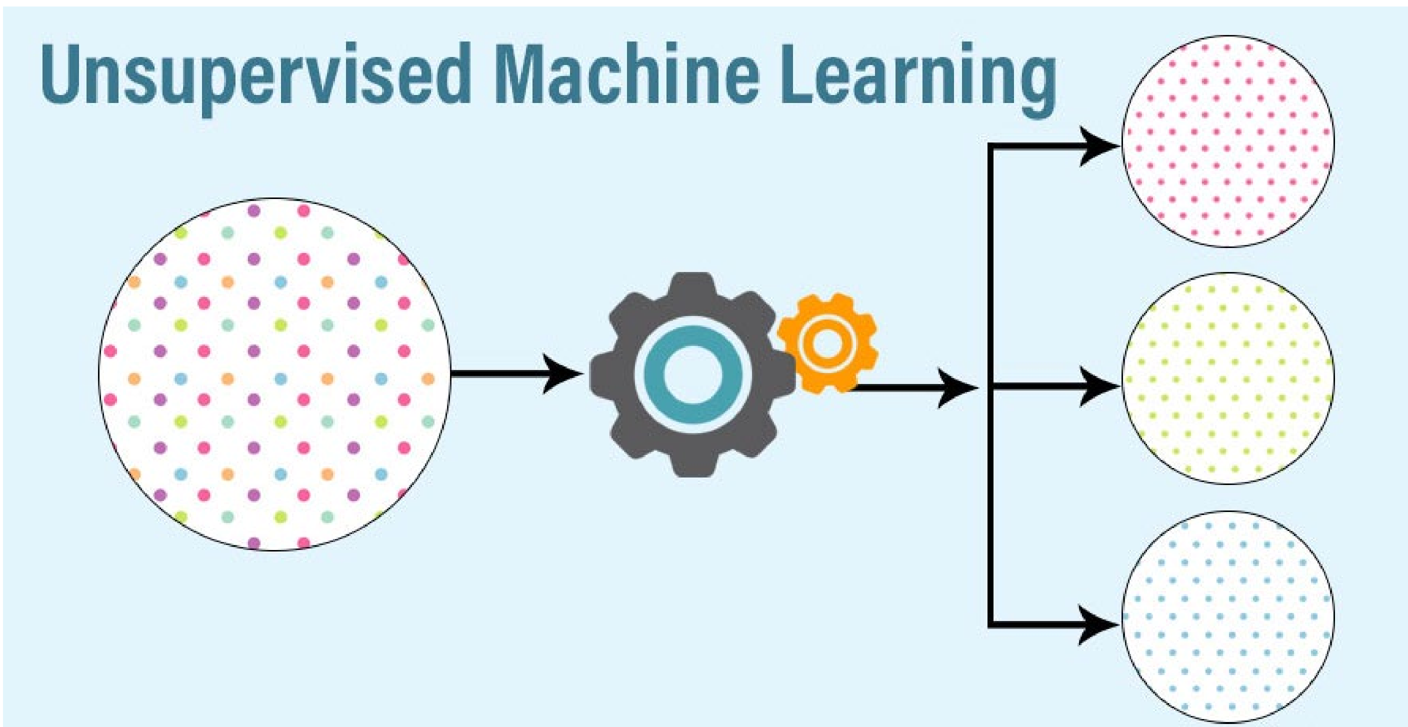
Supervised Learning	*Unsupervised Learning*
classification or categorization	clustering

Machine Learning Problems

Supervised Learning	***Unsupervised Learning***
classification or categorization	clustering

Clustering Strategies

- K-means
  - Iteratively re-assign points to the nearest cluster center.



Machine Learning Problems

Supervised Learning	***Unsupervised Learning***
classification or categorization	clustering

Sample Applications

- Face recognition
- Character recognition
- Speech recognition
- Medical diagnosis
- Industrial applications
- Web search
- Space exploration
- Robotics
- Information extraction
- Social networks

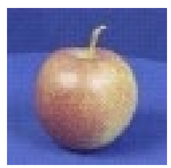
Face Recognition

Training examples of a person

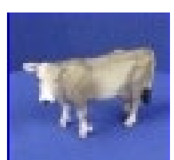
Test images

The machine learning framework

- Apply a prediction function to a feature representation of the image to get the desired output:

f() = “apple”

f() = “tomato”

f() = “cow”

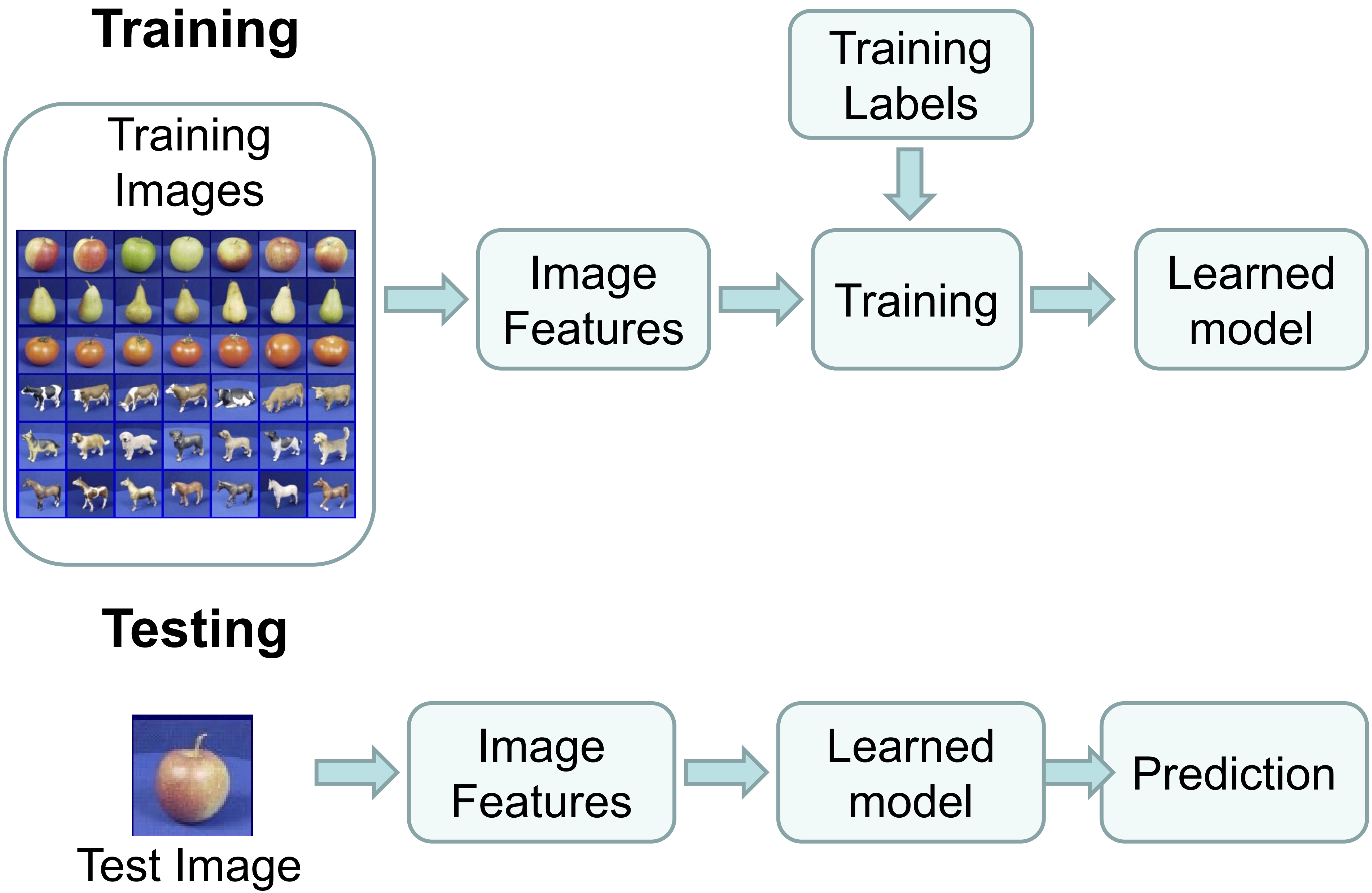
The machine learning framework

$$\mathbf{y} = f(\mathbf{x})$$

output — prediction function — Image feature

- **Training:** given a *training set* of labeled examples $\{(\mathbf{x}_1,y_1), \ldots, (\mathbf{x}_N,y_N)\}$, estimate the prediction function f by minimizing the prediction error on the training set
- **Testing:** apply f to a never before seen *test example* $\mathbf{x}$ and output the predicted value $y = f(\mathbf{x})$

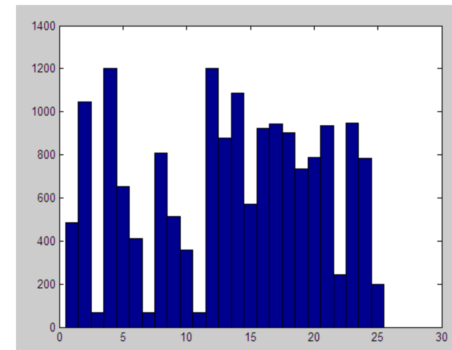
Steps

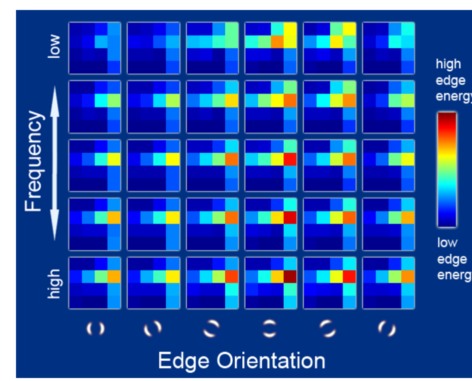


Training
Training Images
Training Labels
Image Features
Training
Learned model
Testing
Test Image
Image Features
Learned model
Prediction

Features

- Raw pixels
- Histograms
- Other descriptors

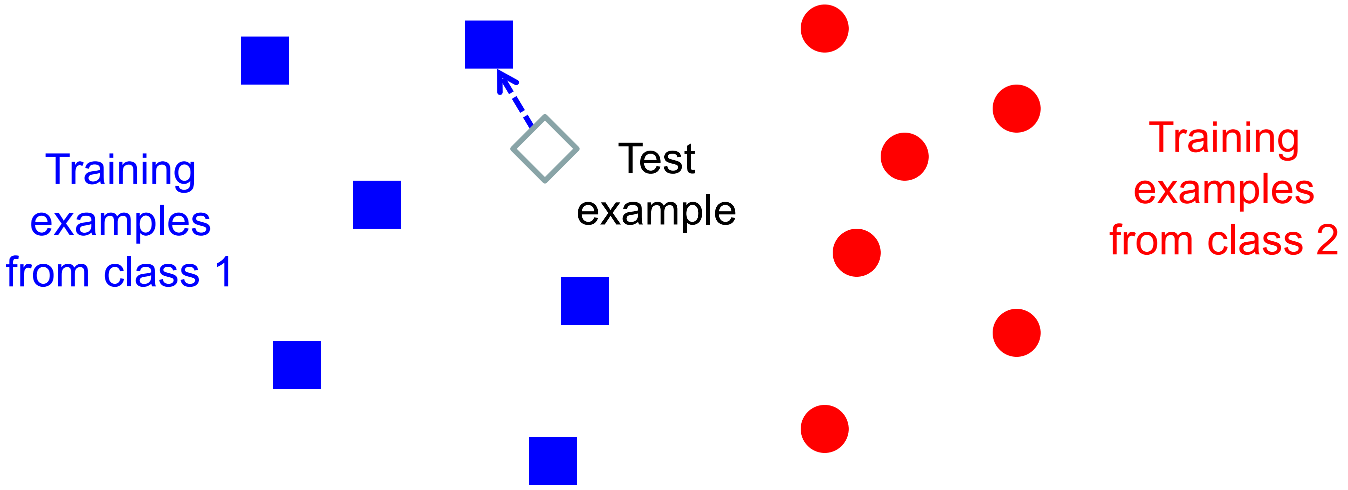




Many classifiers to choose from

- K-nearest neighbor
- Neural networks
- SVM
- Deep Neural networks
- Etc.

Classifiers: Nearest neighbor

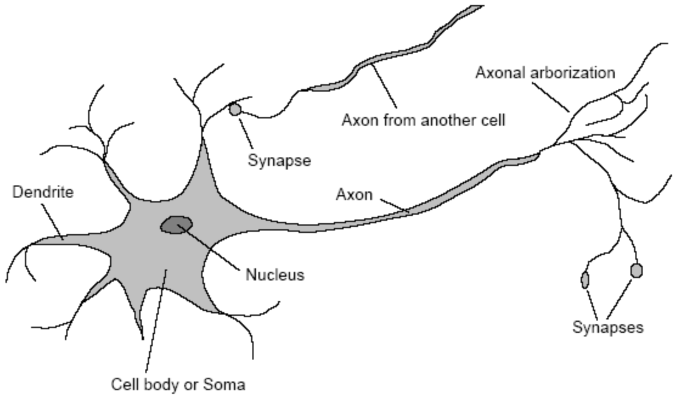


f(**x**) = label of the training example nearest to **x**

- All we need is a distance function for our inputs
- No training required!

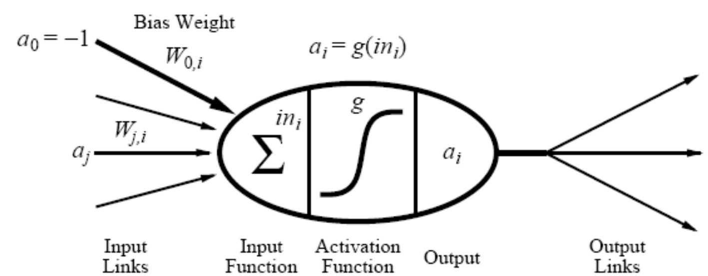
(Artificial) Neural Networks

- Motivation: human brain
  - massively parallel (*10^{11} neurons, ~20 types*)
  - small computational units with simple low-bandwidth communication (*10^{14} synapses, 1-10ms cycle time*)

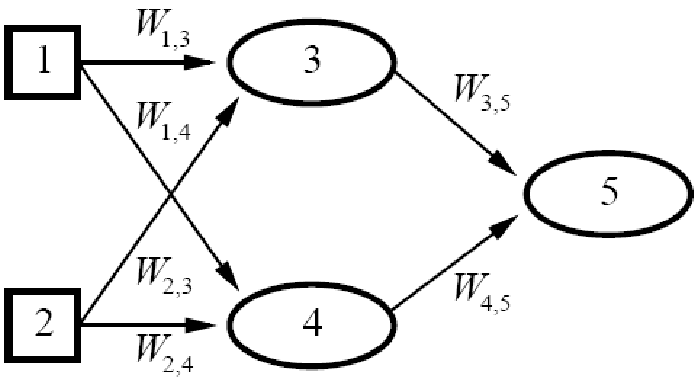


- Realization: neural network
  - *units* (≈ neurons) connected by *directed weighted links*
  - *activation function* from inputs to output

$$a_i \leftarrow g(in_i) = g\left(\Sigma_j W_{j,i} a_j\right)$$



Neural Networks (*continued*)



$$
\begin{aligned}
a_5 &= g(W_{3,5} \cdot a_3 + W_{4,5} \cdot a_4) \\
&= g(W_{3,5} \cdot g(W_{1,3} \cdot a_1 + W_{2,3} \cdot a_2) + W_{4,5} \cdot g(W_{1,4} \cdot a_1 + W_{2,4} \cdot a_2))
\end{aligned}
$$

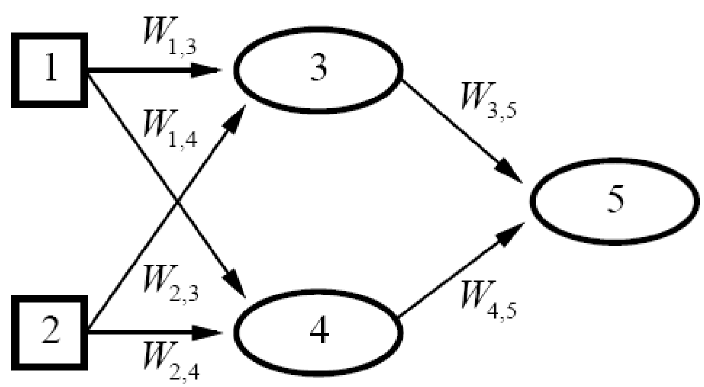
- *neural network = parameterized family of nonlinear functions*
- *types*
  - *feed-forward* (acyclic): single-layer perceptrons, multi-layer networks
  - *recurrent* (cyclic): Hopfield networks, Boltzmann machines

Neural Network Learning

Key Idea: Adjusting the weights changes the function represented by the neural network (*learning = optimization in weight space*).

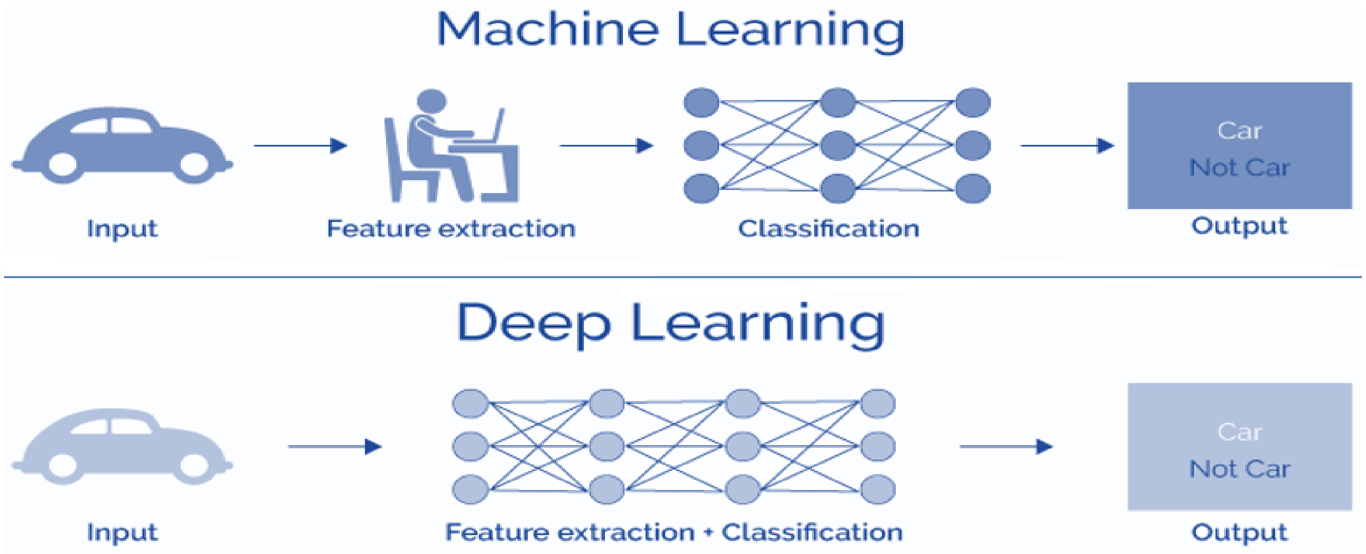
Iteratively *adjust weights* to reduce *error* (difference between network output and target output).

- Weight Update
  - *backpropagation*

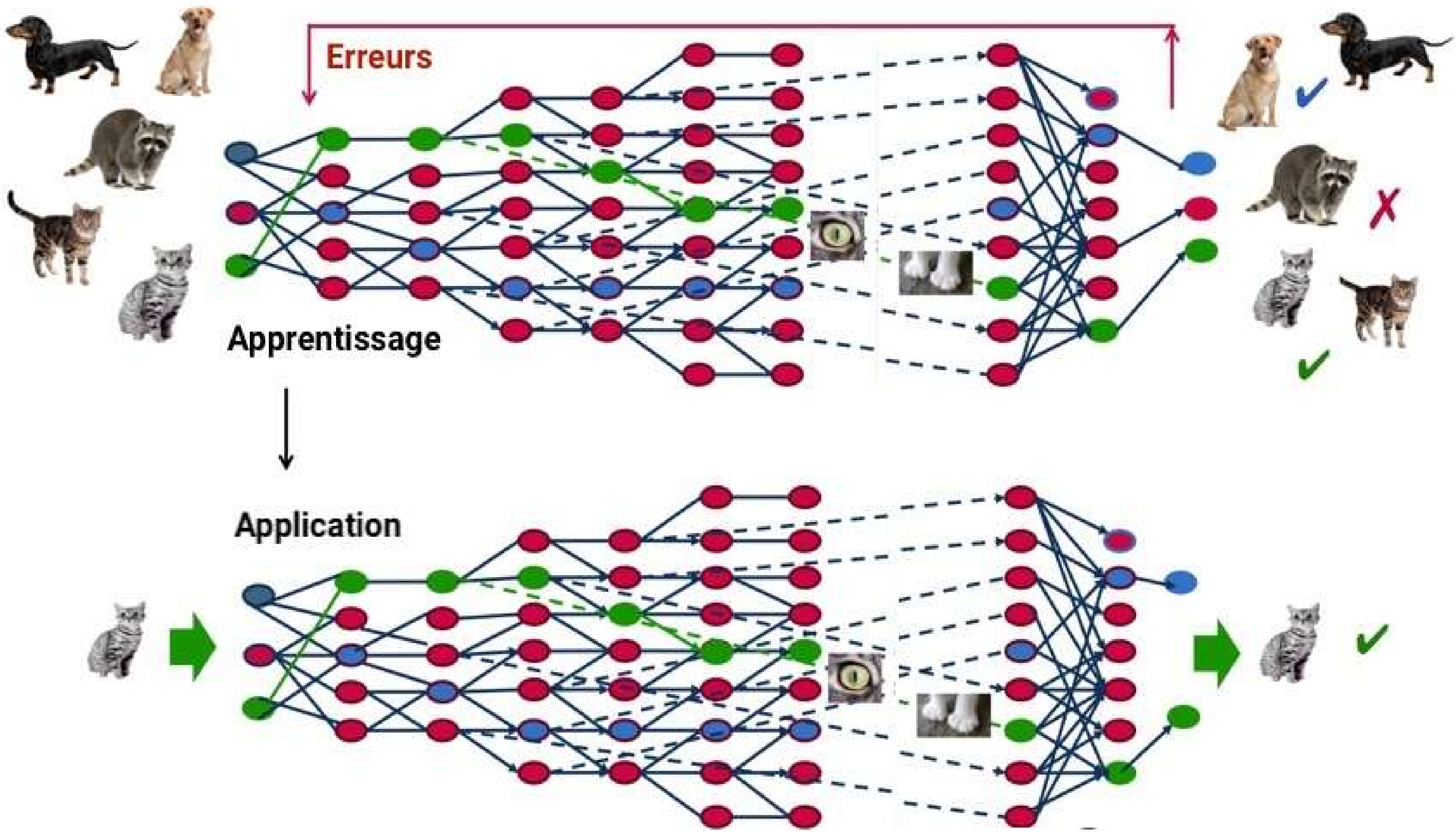


Deep Learning

- Deep learning (**DL**) is a **subtype** of machine learning (ML). DL can process a wider range of data resources, requires less data preprocessing by humans (e.g. feature labelling), and can sometimes produce more accurate results than traditional ML approaches (although it requires a larger amount of data to do so).
- However, it is computationally more expensive in time to execute, hardware costs and data quantities.



Deep Learning



Deep Learning