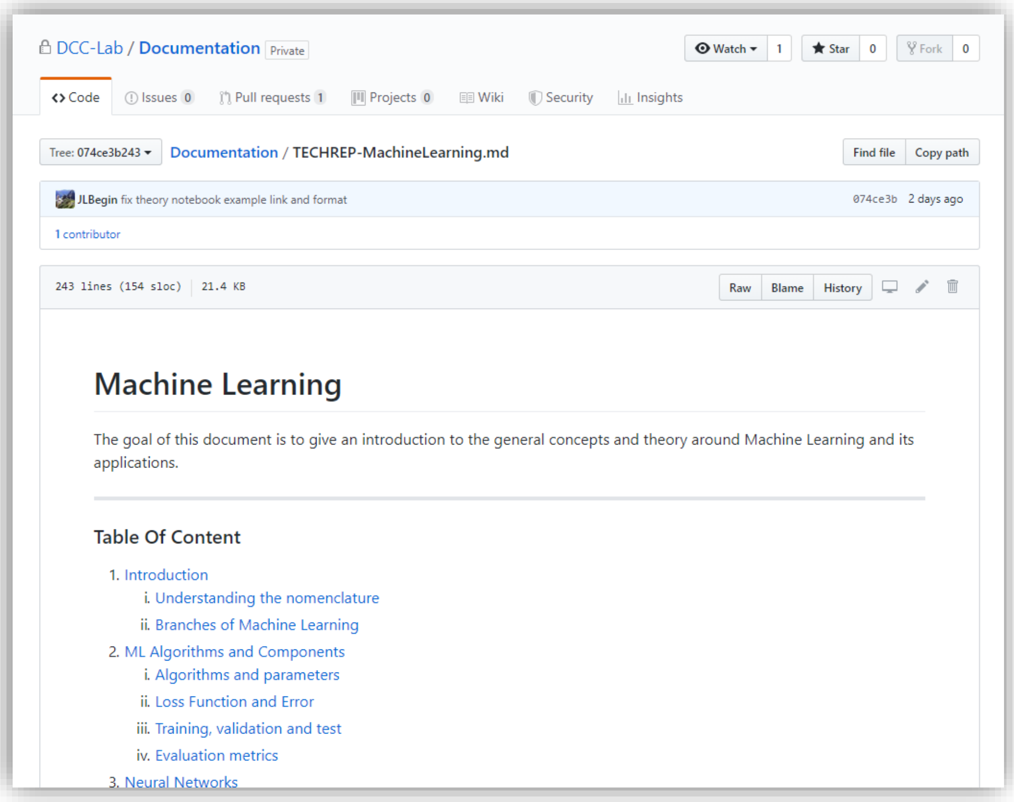
Machine Learning

DCCLab

GitHub



DCC-Lab/Documentation/TECHREP-MachineLearning.md - .html

Machine Learning

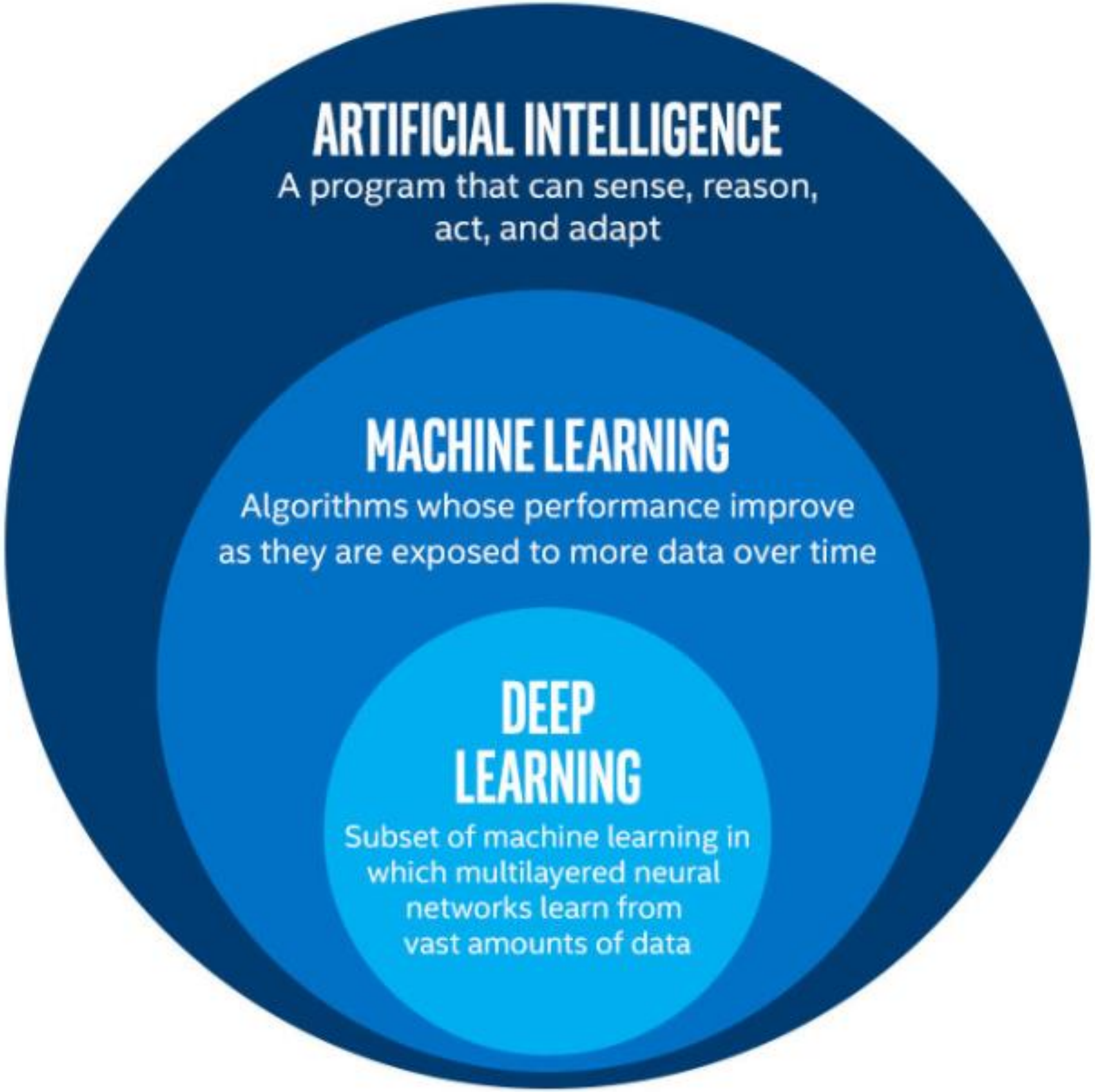
1. Introduction
2. ML Algorithms and components
3. Neural Networks
4. Convolutional Neural Networks

(+ Coding procedure & example)

Machine Learning

1. **Introduction**
2. ML Algorithms and components
3. Neural Networks
4. Convolutional Neural Networks

(+ Coding procedure & example)

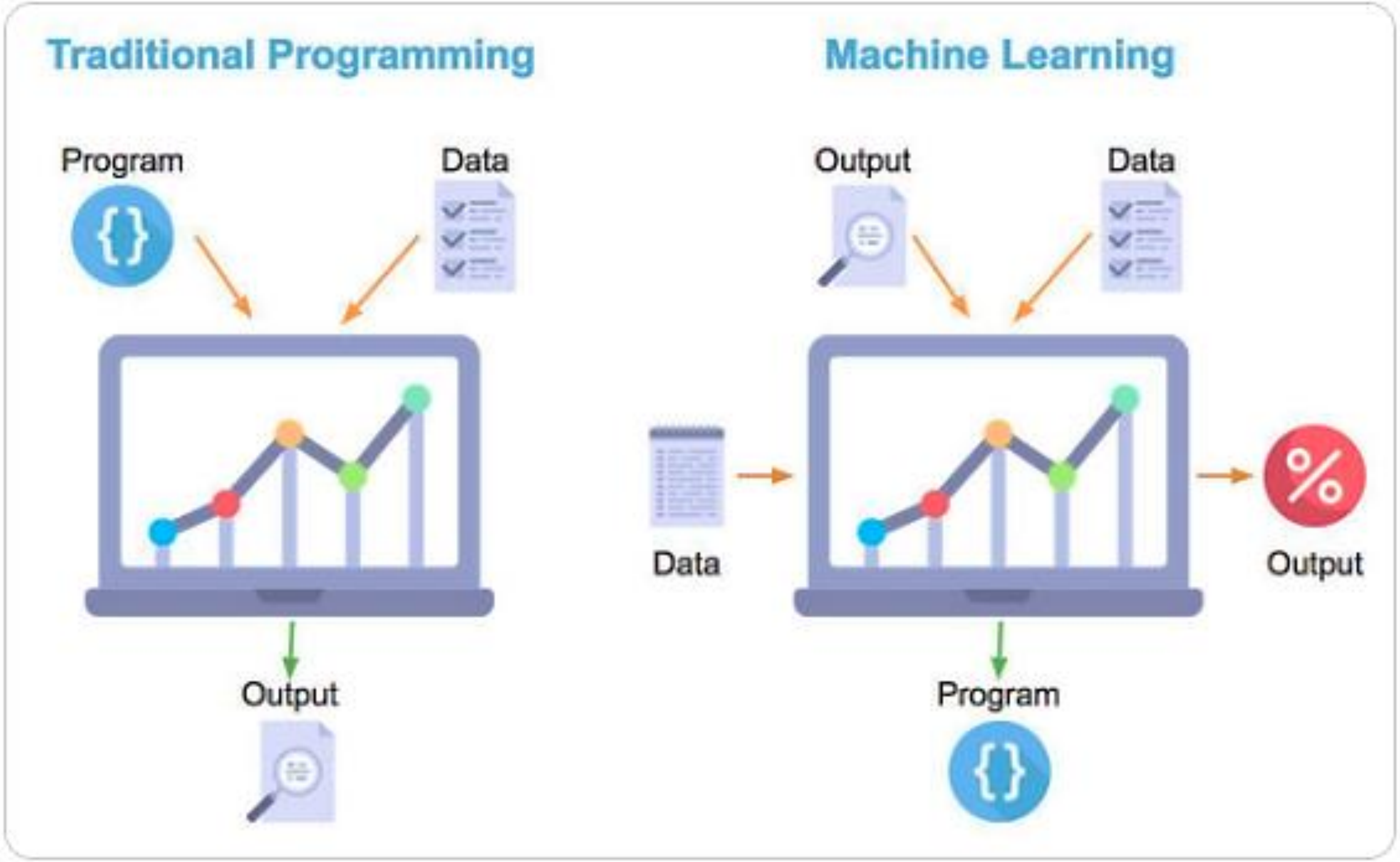

ARTIFICIAL INTELLIGENCE
A program that can sense, reason, act, and adapt
MACHINE LEARNING
Algorithms whose performance improve as they are exposed to more data over time
DEEP LEARNING
Subset of machine learning in which multilayered neural networks learn from vast amounts of data

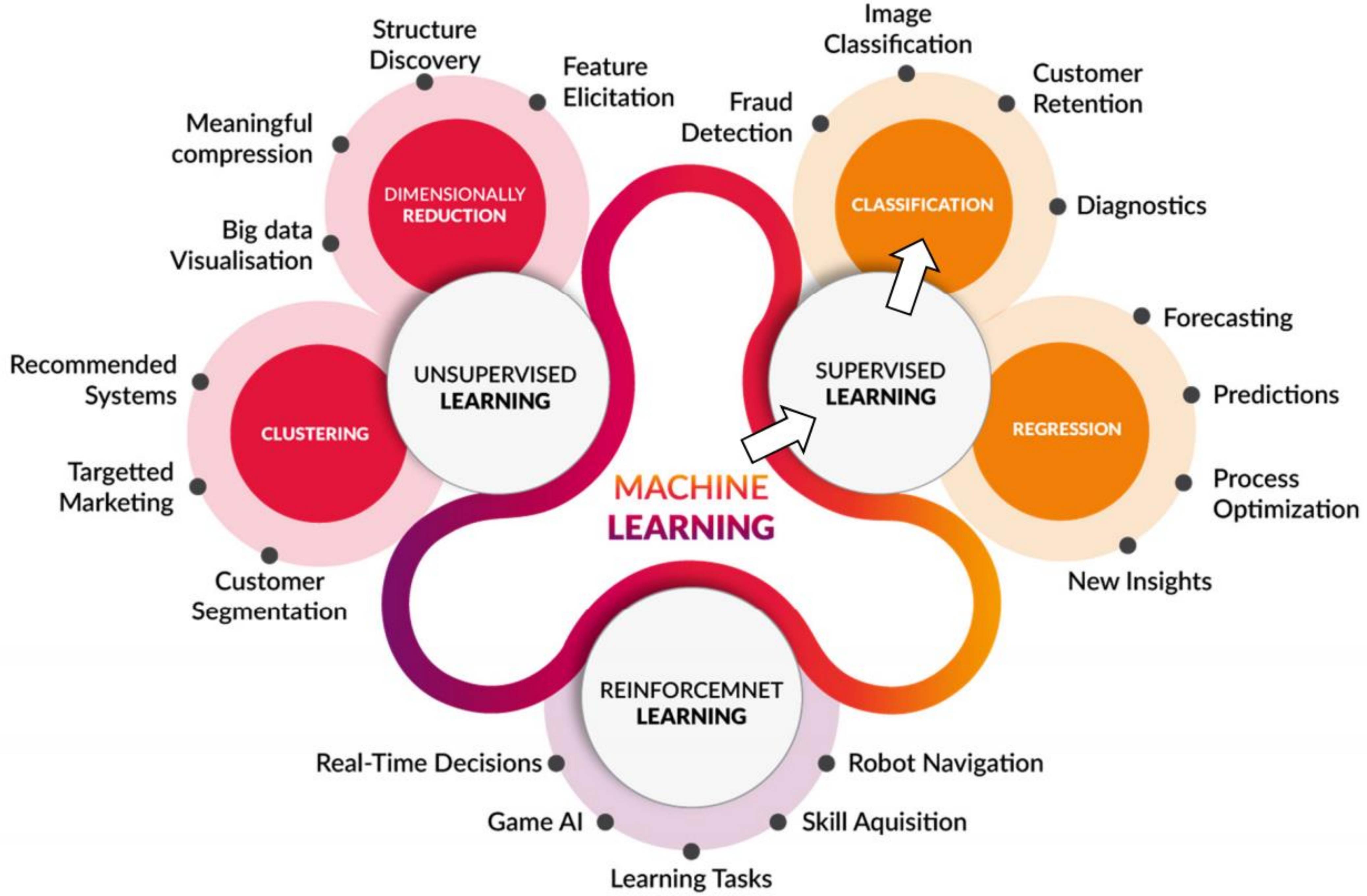
What is Machine Learning ? (Apprentissage automatique)

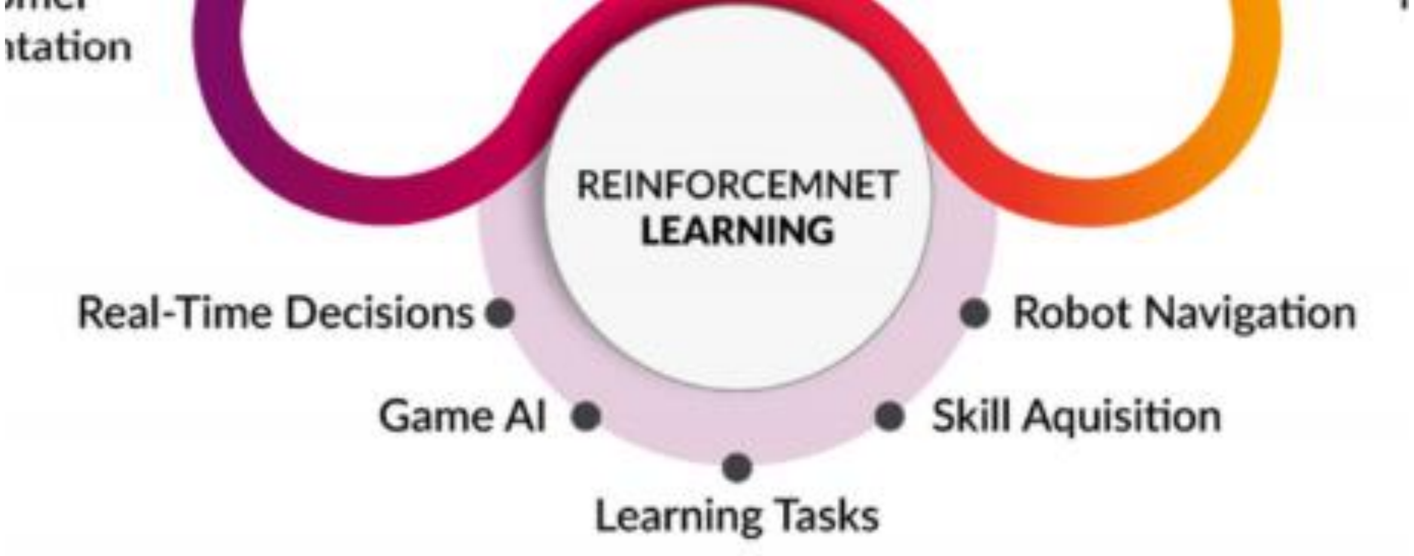
Machine Learning is a field of study that gives computers the ability to learn without being explicitly programmed.

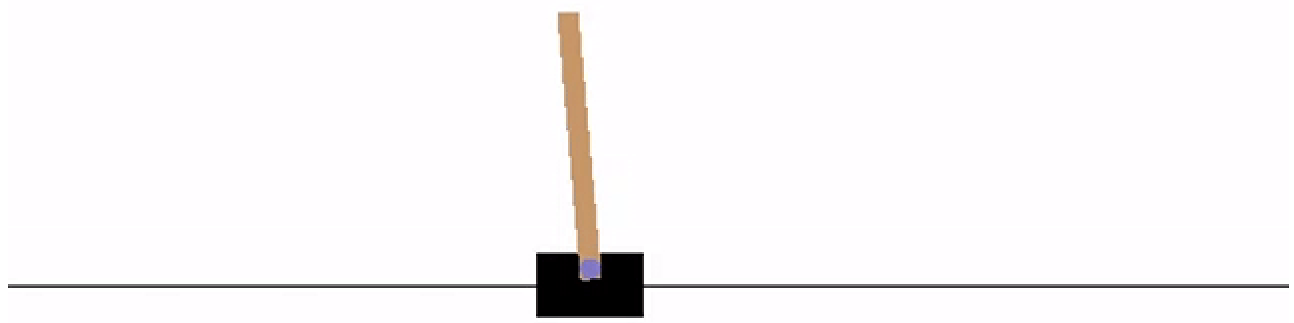
- **Arthur Samuel (1959)**

Traditional programming VS Machine Learning

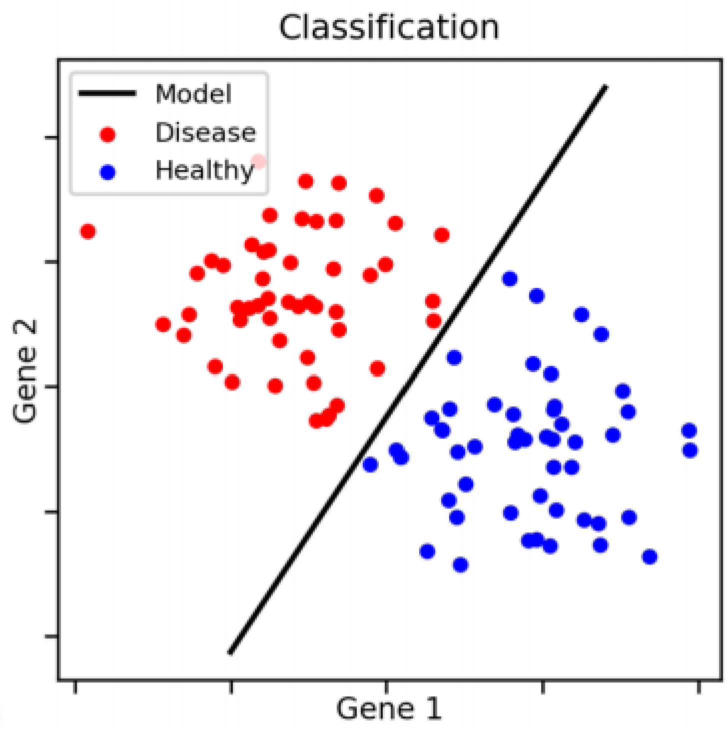


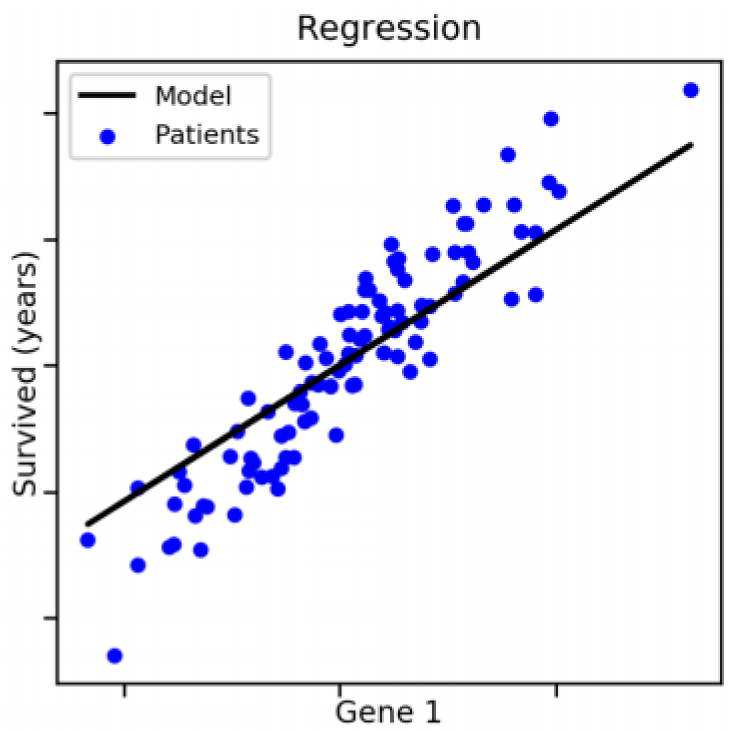

MACHINE
LEARNING
UNSUPERVISED
LEARNING
SUPERVISED
LEARNING
REINFORCEMNET
LEARNING
DIMENSIONALLY
REDUCTION
CLUSTERING
CLASSIFICATION
REGRESSION
Structure
Discovery
Feature
Elicitation
Meaningful
compression
Big data
Visualisation
Recommended
Systems
Targetted
Marketing
Customer
Segmentation
Image
Classification
Customer
Retention
Fraud
Detection
Diagnostics
Forecasting
Predictions
Process
Optimization
New Insights
Real-Time Decisions
Robot Navigation
Game AI
Skill Aquisition
Learning Tasks



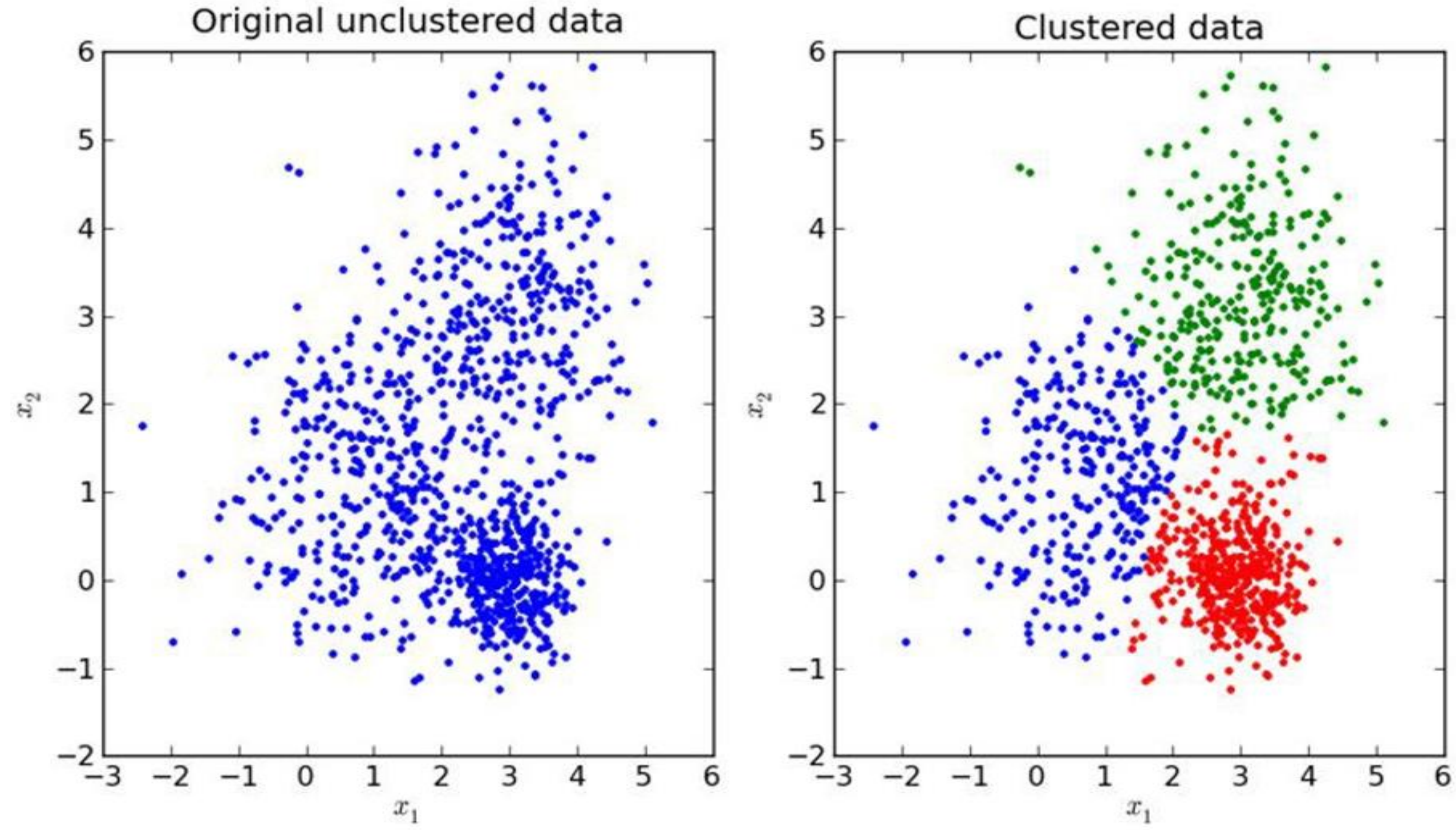


Optimizing a decision making policy with experiences and rewards.

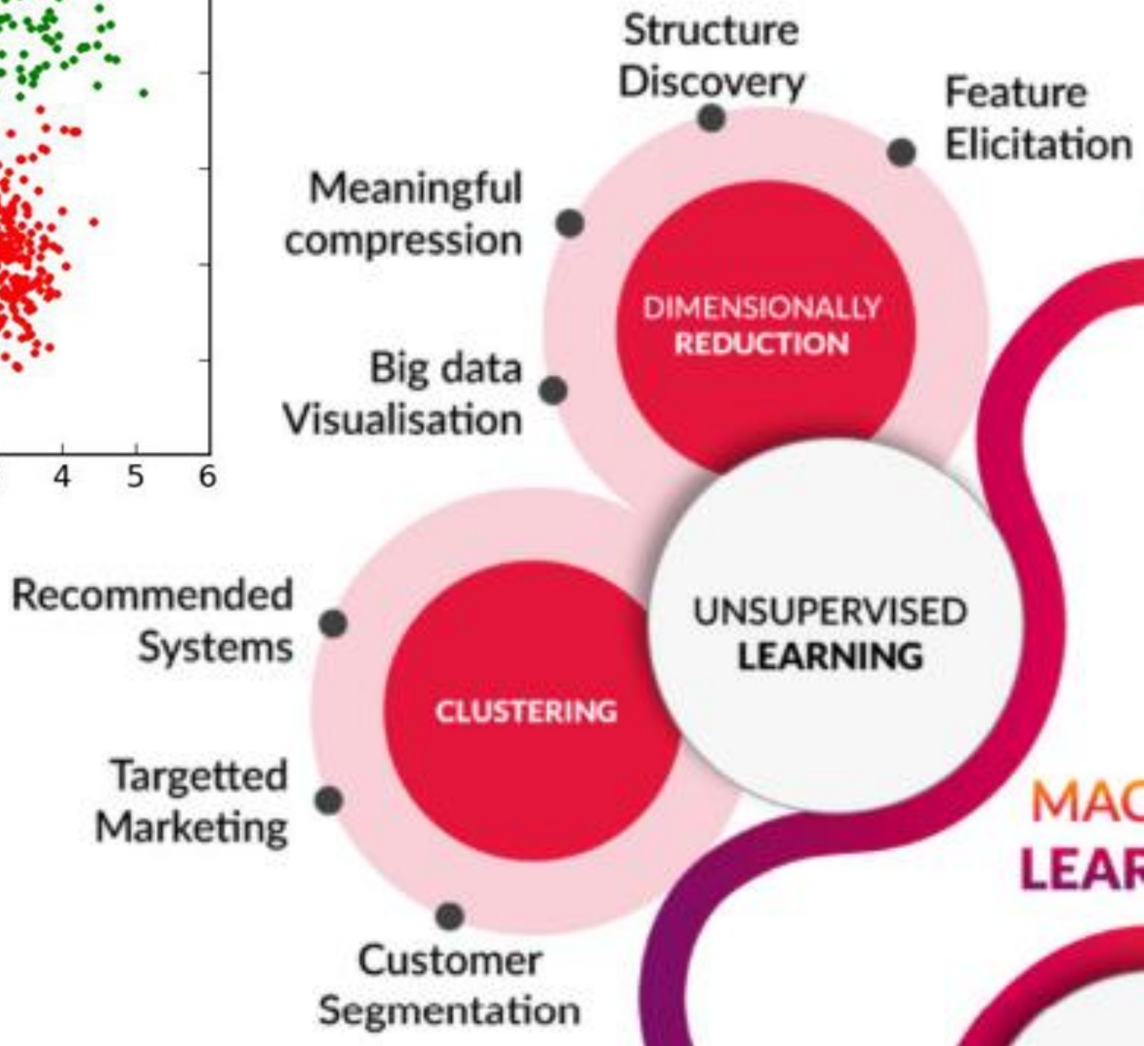


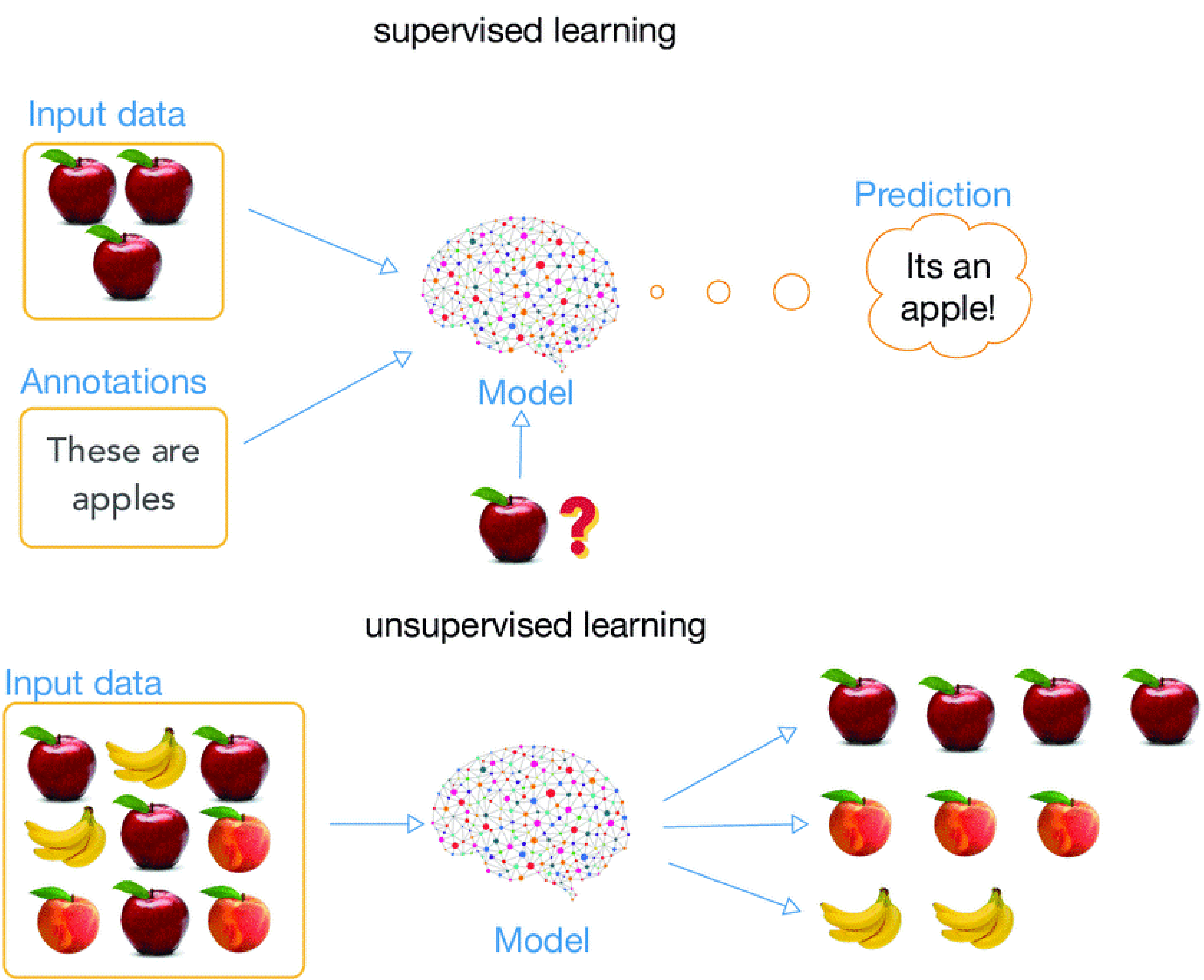


Learn a function that maps an input to an output based on a learning process over training examples.



Find unknown patterns in an input dataset without pre-existing labels to regroup input into clusters or reduce dimensions




supervised learning
Input data
Annotations
These are apples
Model
Prediction
Its an apple!
unsupervised learning
Input data
Model

Machine Learning

1. Introduction
2. **ML Algorithms and components**
3. Neural Networks
4. Convolutional Neural Networks

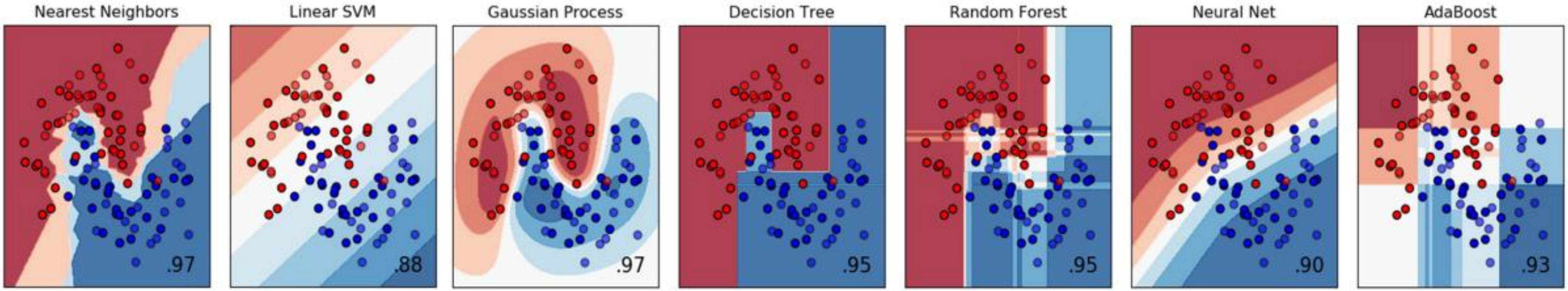
2. ML Algorithms and components

1. Algorithms and parameters
2. Loss Function and Error
3. Training, validation and test
4. Evaluation metrics

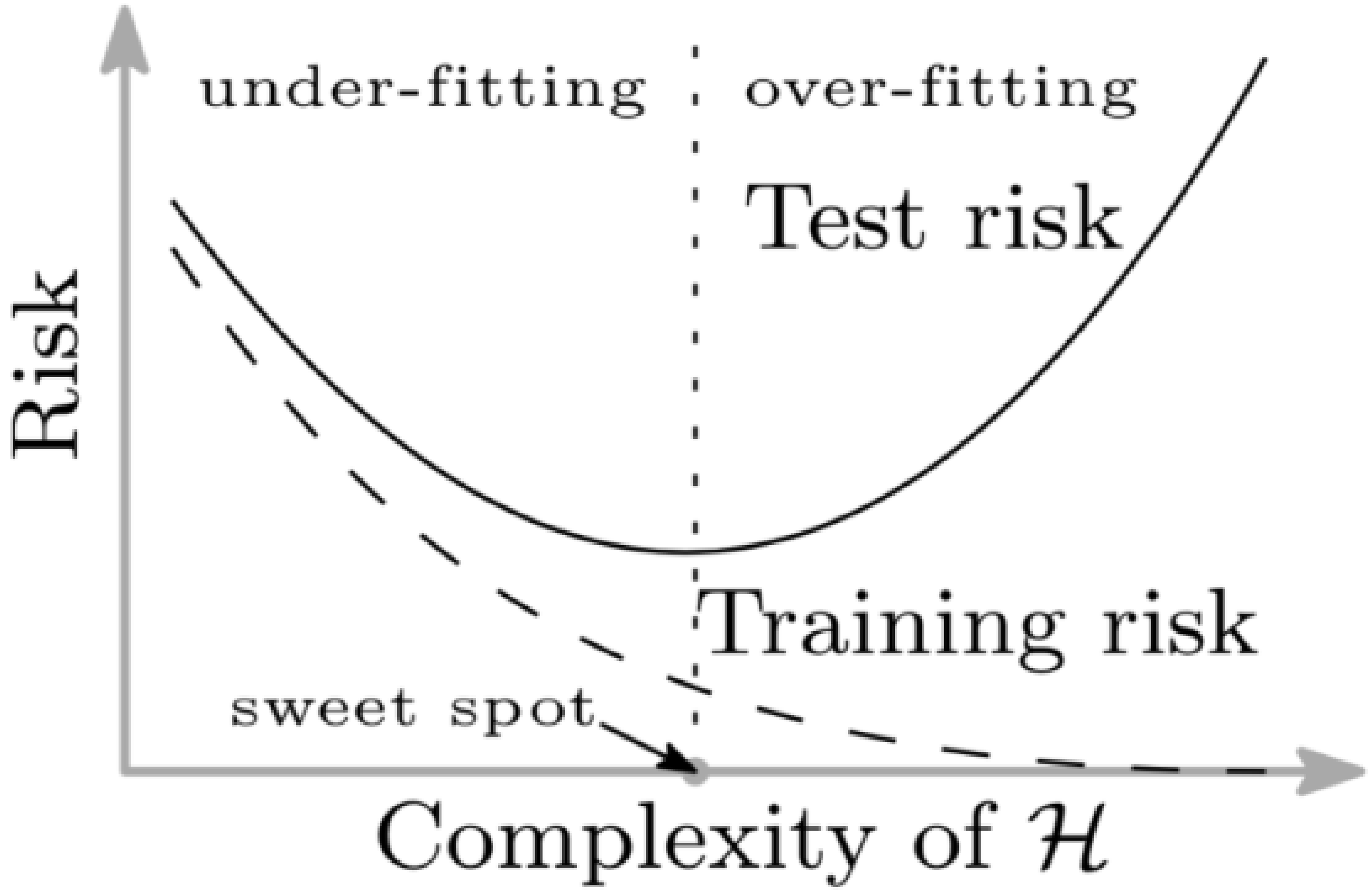
Algorithms and parameters

- Parameters vs hyperparameters
- Not all algorithms are parametric (*k-nearest neighbors*)

- More parameters ≈ more **capacity** to learn complex task



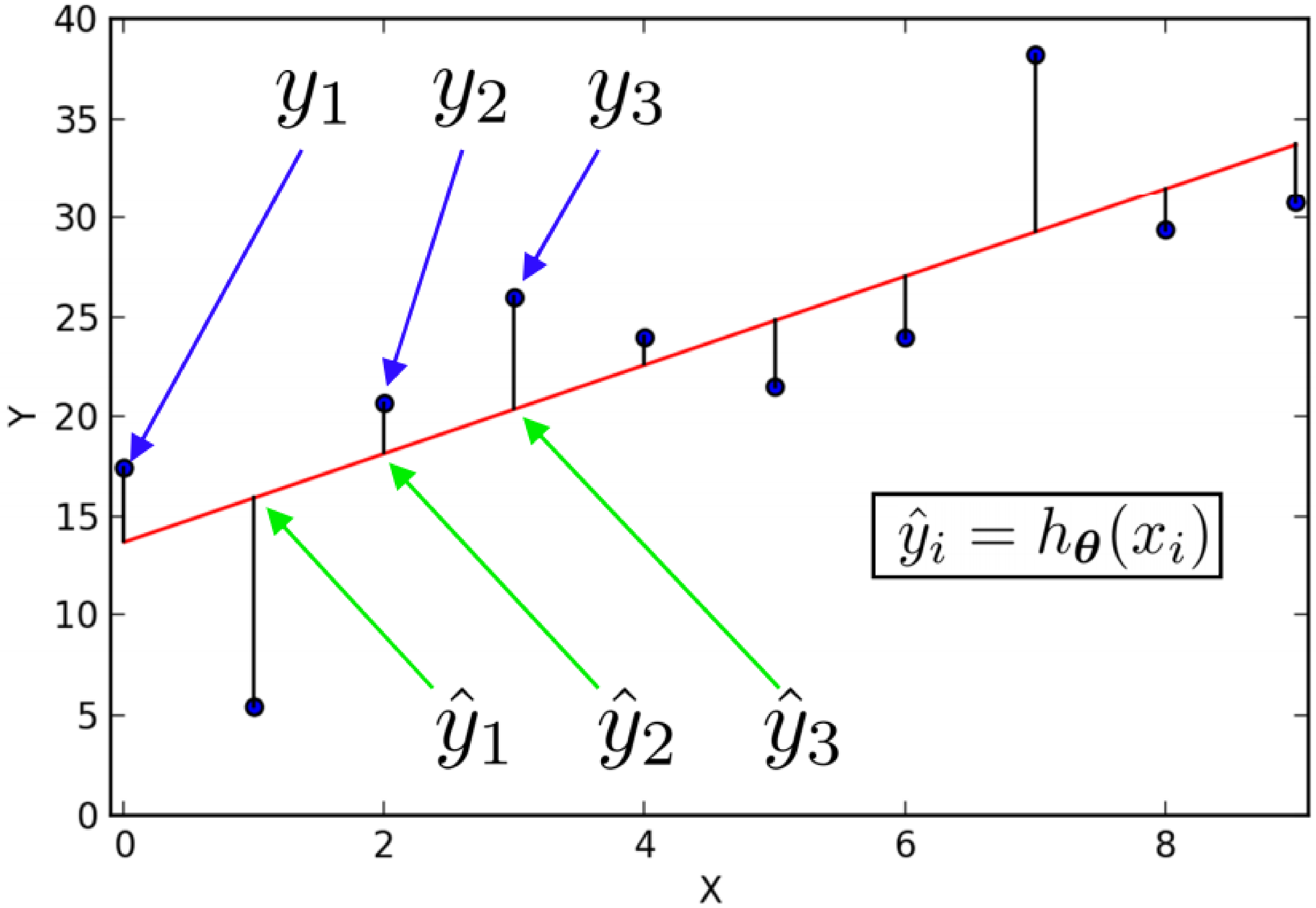
Capacity



(=> Regularization !)

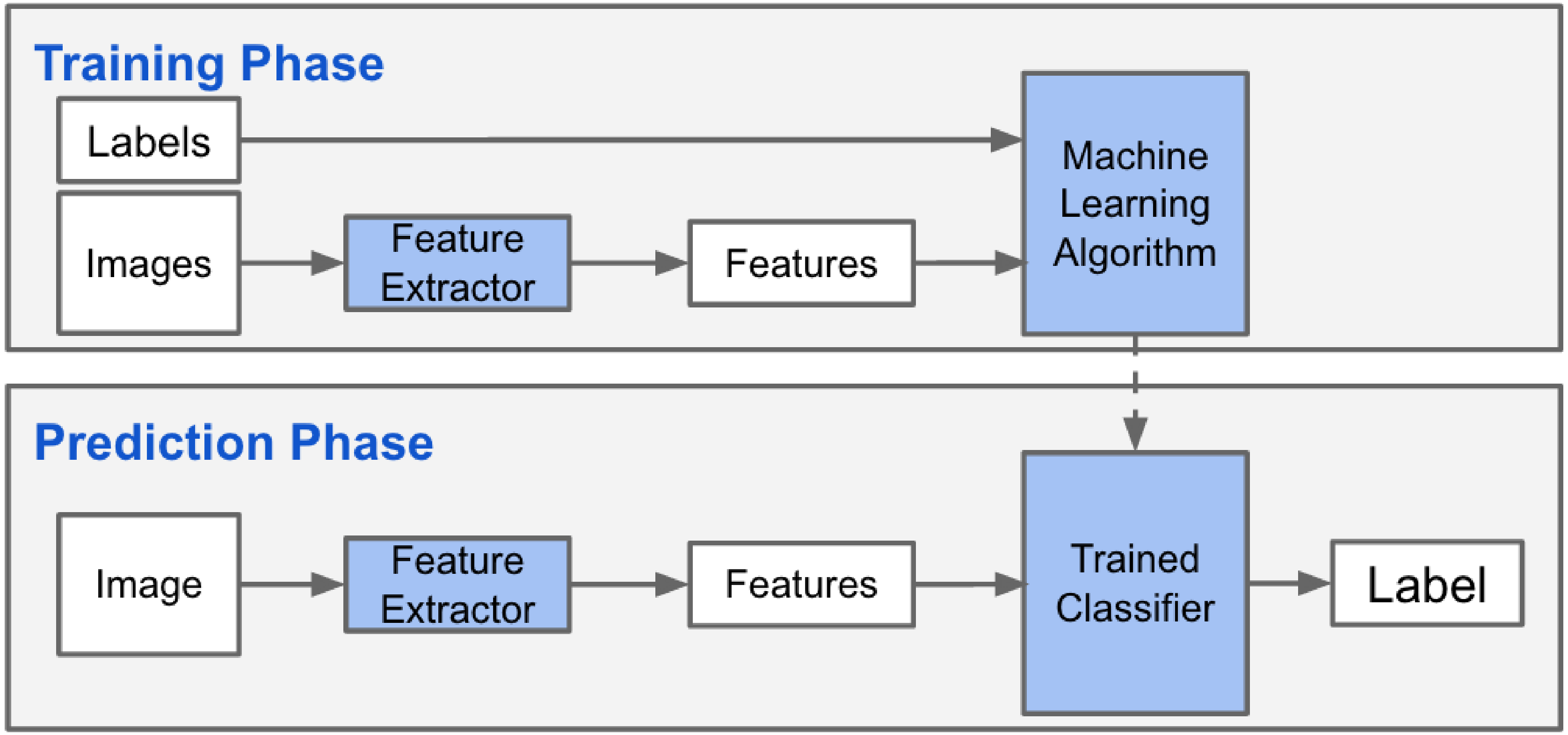
Error and Loss Function

$$R_{emp}(h_{\boldsymbol{\theta}}) = \frac{1}{N}\Big(\mathcal{L}(y_1, \hat{y}_1) + \mathcal{L}(y_2, \hat{y}_2) + \cdots + \mathcal{L}(y_N, \hat{y}_N)\Big)$$



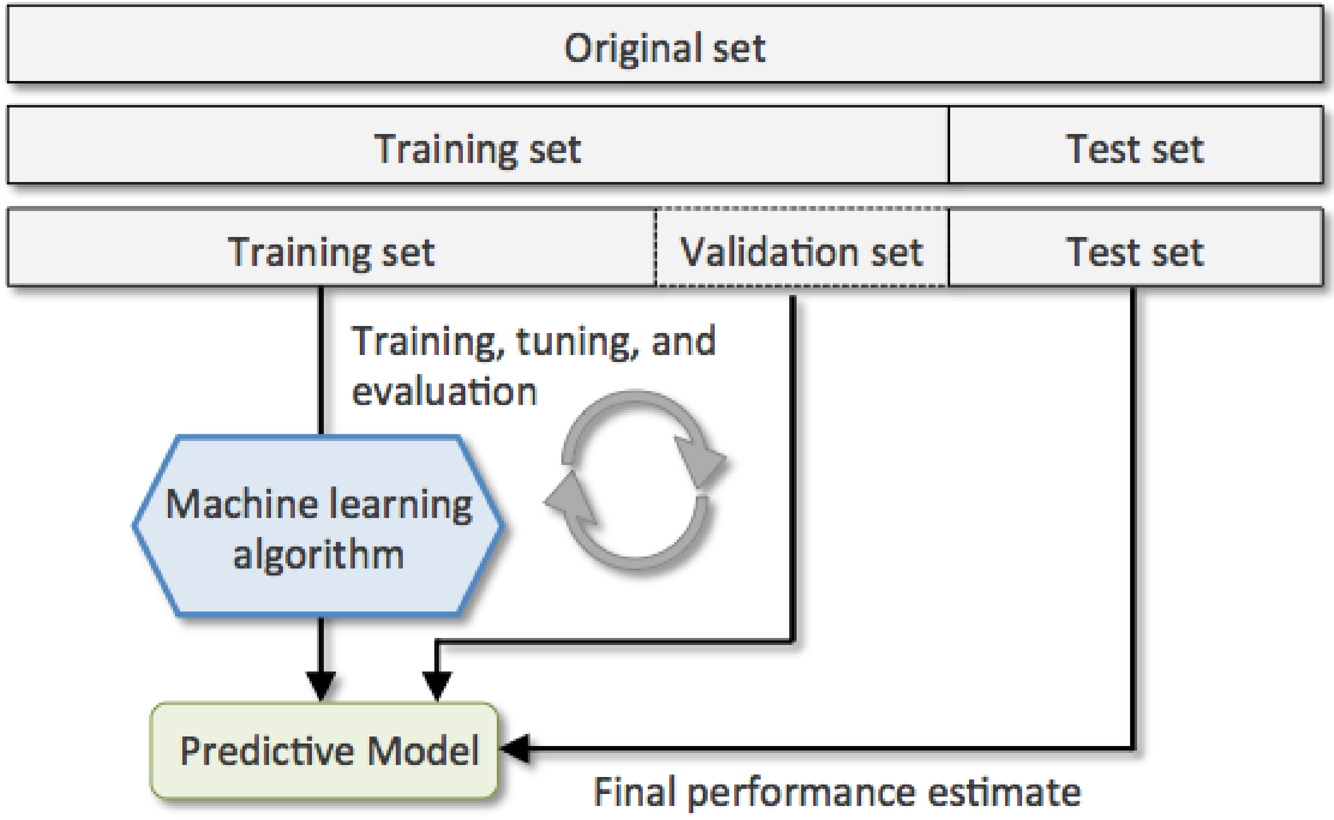
Training

- Batch size
- Epochs

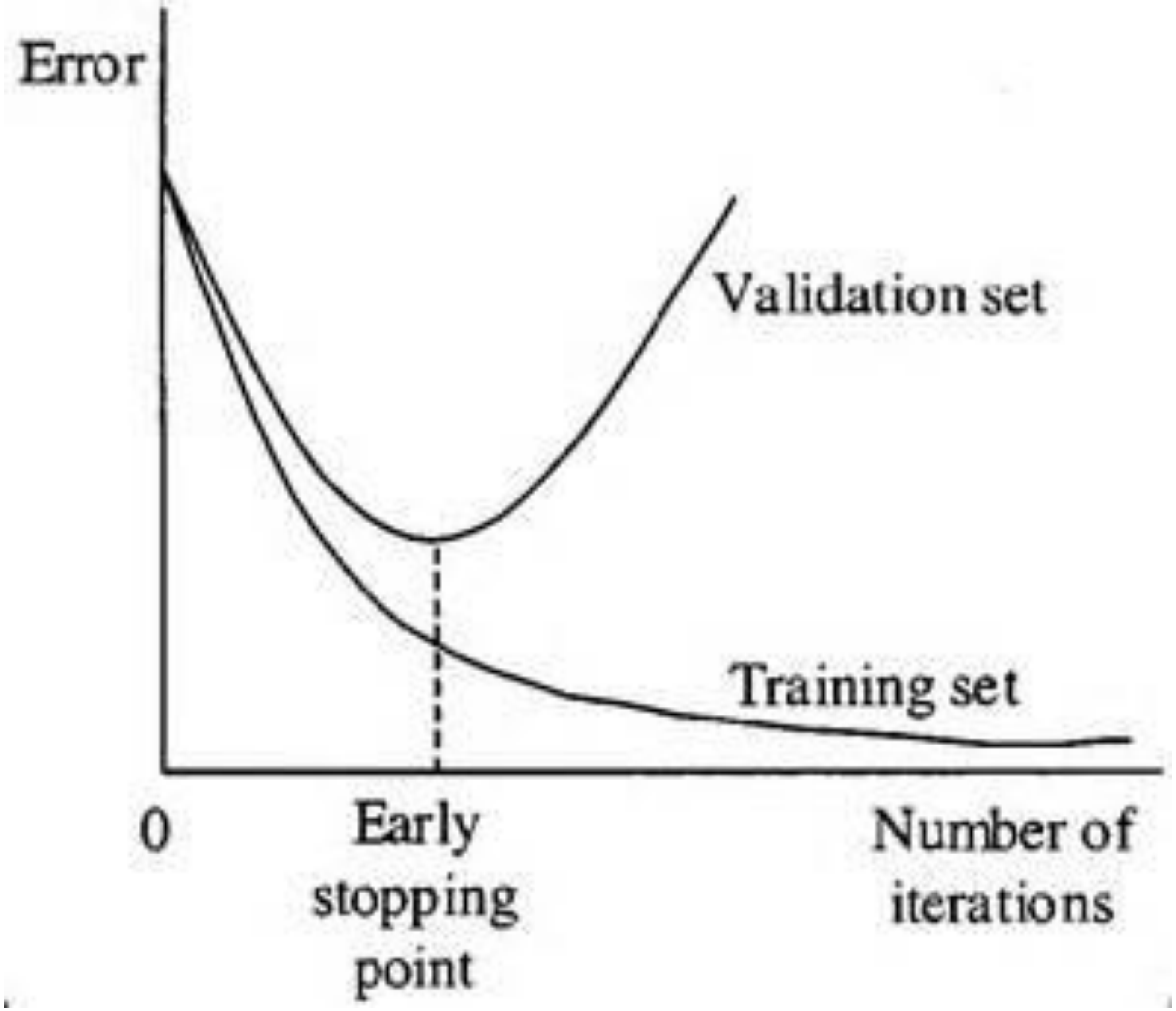


Machine Learning Phases

Train, validation and test



* Training and validation errors give
a **biased** approximation of the risk of the model.


Error
Validation set
Training set
0
Early
stopping
point
Number of
iterations

Evaluation metrics

Regression

- R^2
- MSE/MAE

Classification

- Classification Accuracy
- Area under ROC curve
- Precision
- F1-Score

		Actual	
		Positive	Negative
Predicted	Positive	True Positive	False Positive
	Negative	False Negative	True Negative

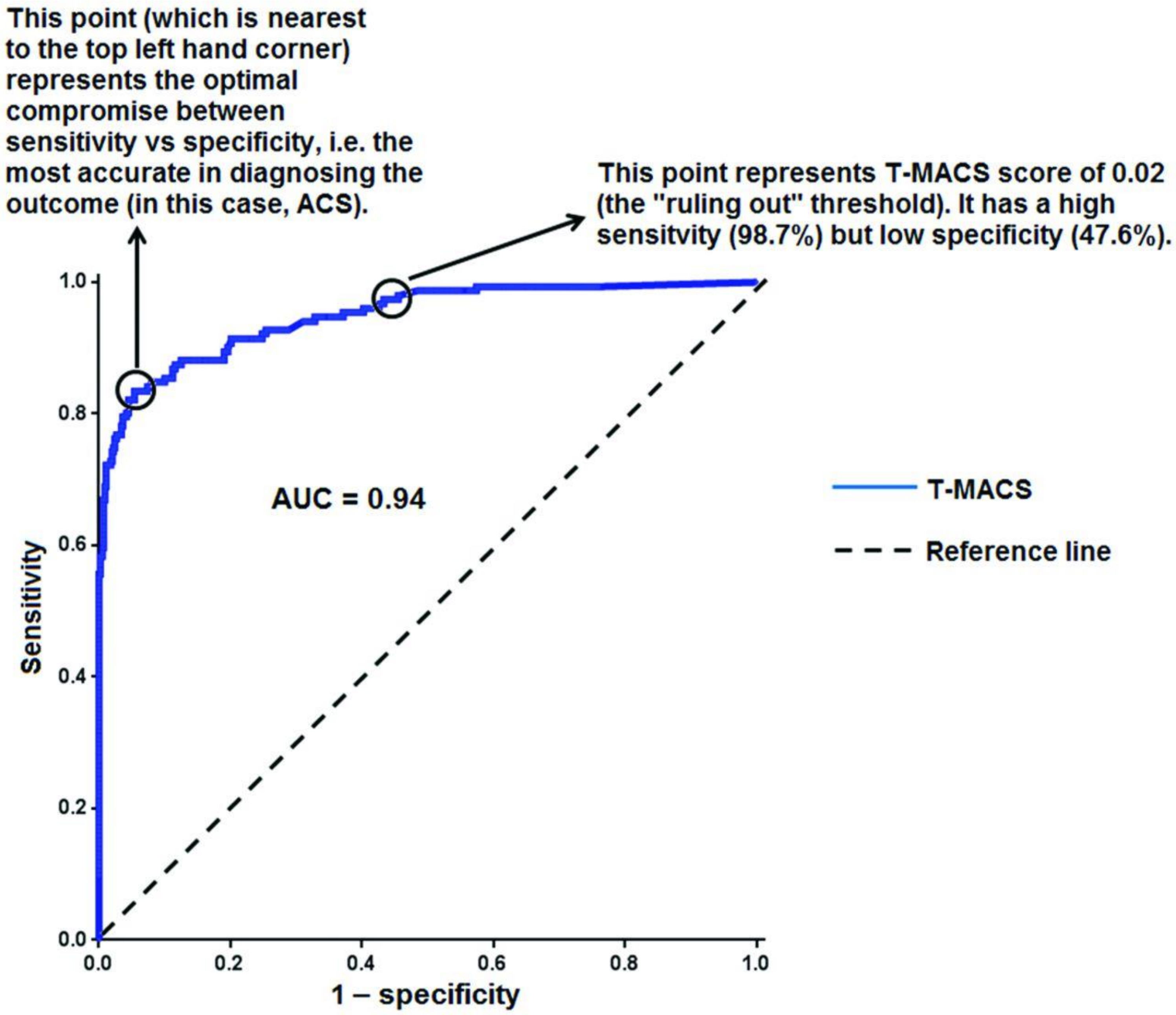
Classification accuracy

$$\textbf{Accuracy} = \frac{\text{Number of correct predictions}}{\text{Total number of predictions}}$$

Only suitable when there is an **equal amount of observations for each class** and that all predictions are **equally important**.

AUC
(Area under ROC curve)

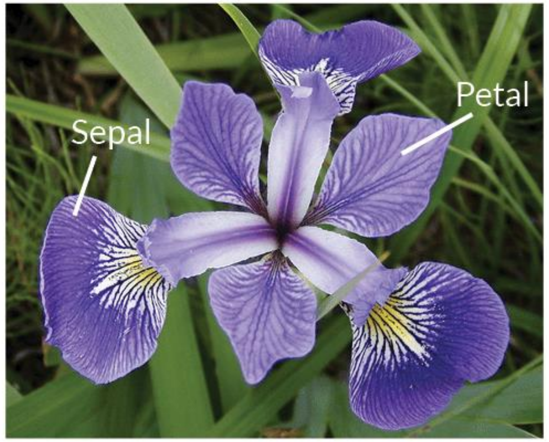
For **binary classification**



Plot of the true positive rate against the false positive rate at various threshold settings.

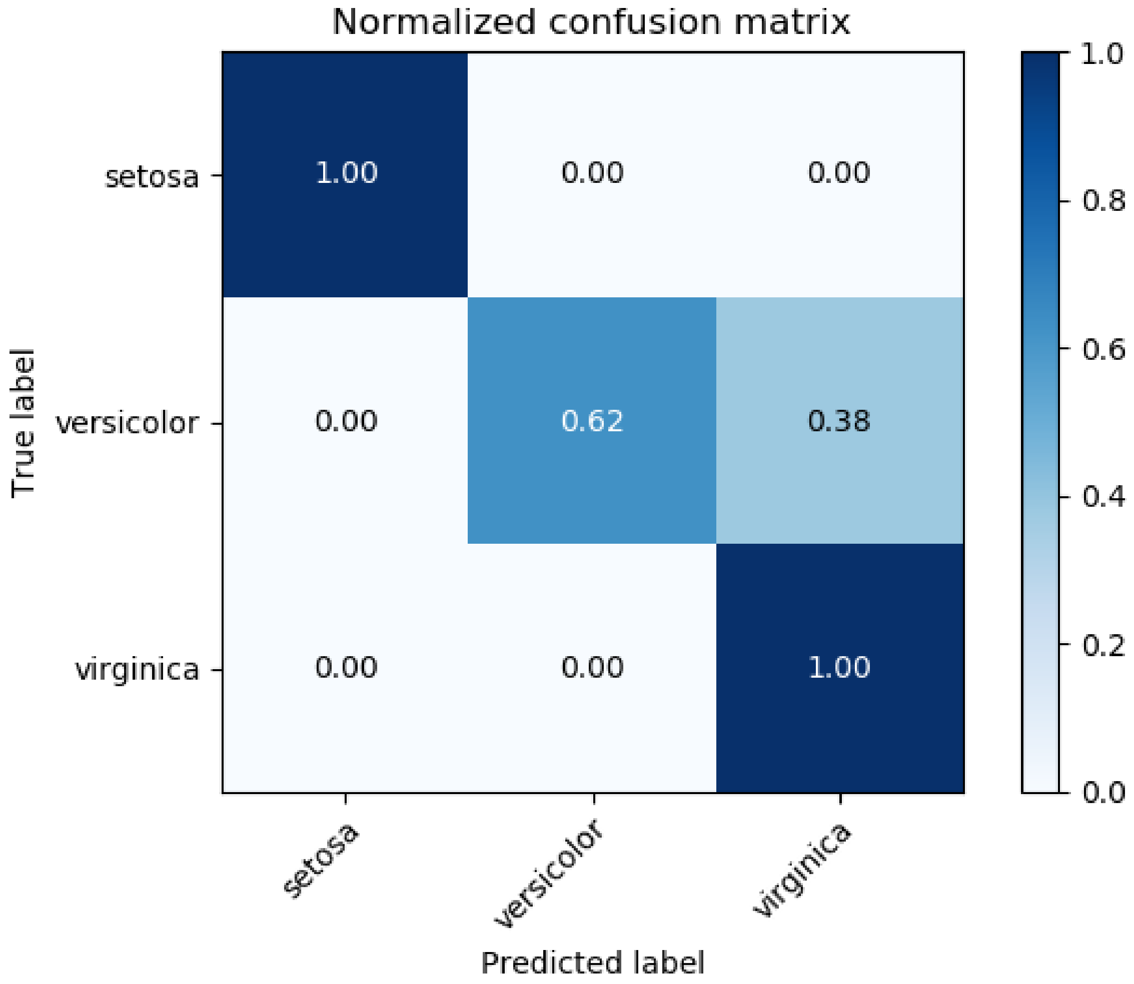
Confusion Matrix

Iris Setosa



Iris Versicolor

Iris Virginica



Precision and sensitivity (recall)

precision: TP/cancer diagnoses

		Diagnosis	
		No cancer	Cancer
True state	No cancer	*TN*	*FP*
	Cancer	*FN*	*TP*

recall: TP/cancer true states

F1 Score

$$F_1 = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

F1 score is the harmonic mean of precision and recall. Values range from 0 (bad) to 1 (good).

Chris Albon

Evaluation metrics

Regression

- R^2
- MSE/MAE

Classification

- Classification Accuracy
- Area under ROC curve
- Precision
- F1-Score

		Actual	
		Positive	Negative
Predicted	Positive	True Positive	False Positive
	Negative	False Negative	True Negative

Maxence Larose

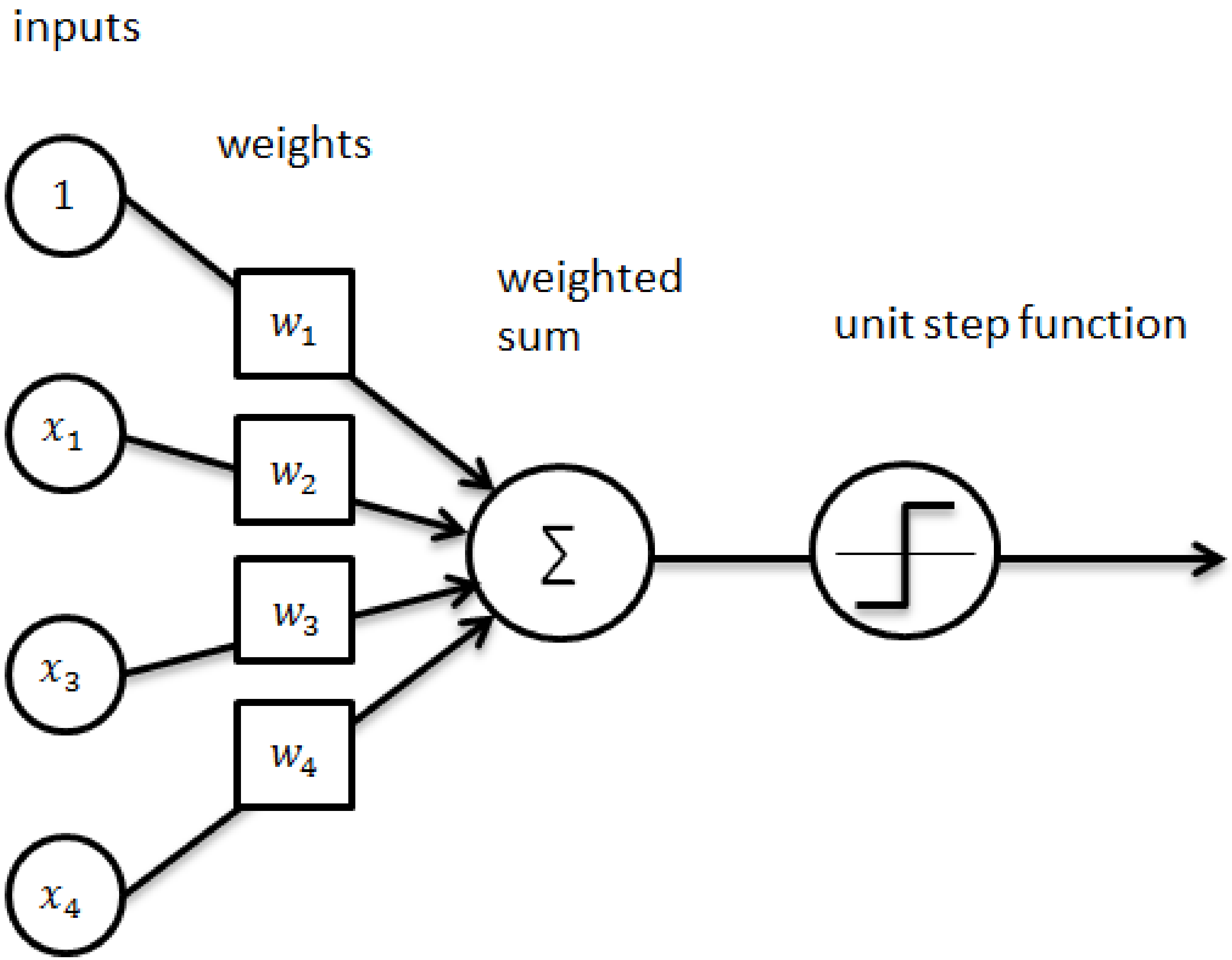
1. Introduction
2. ML Algorithms and components
3. **Neural Networks**
4. Convolutional Neural Networks

Neural Networks

1. Perceptron (single-layer)
2. Neural Network (multi-layer)
3. Backpropagation

Perceptron

Linear binary classifier

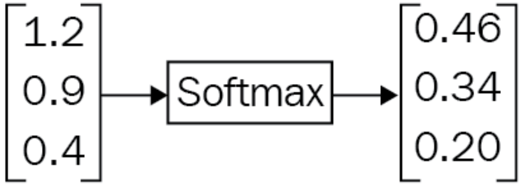


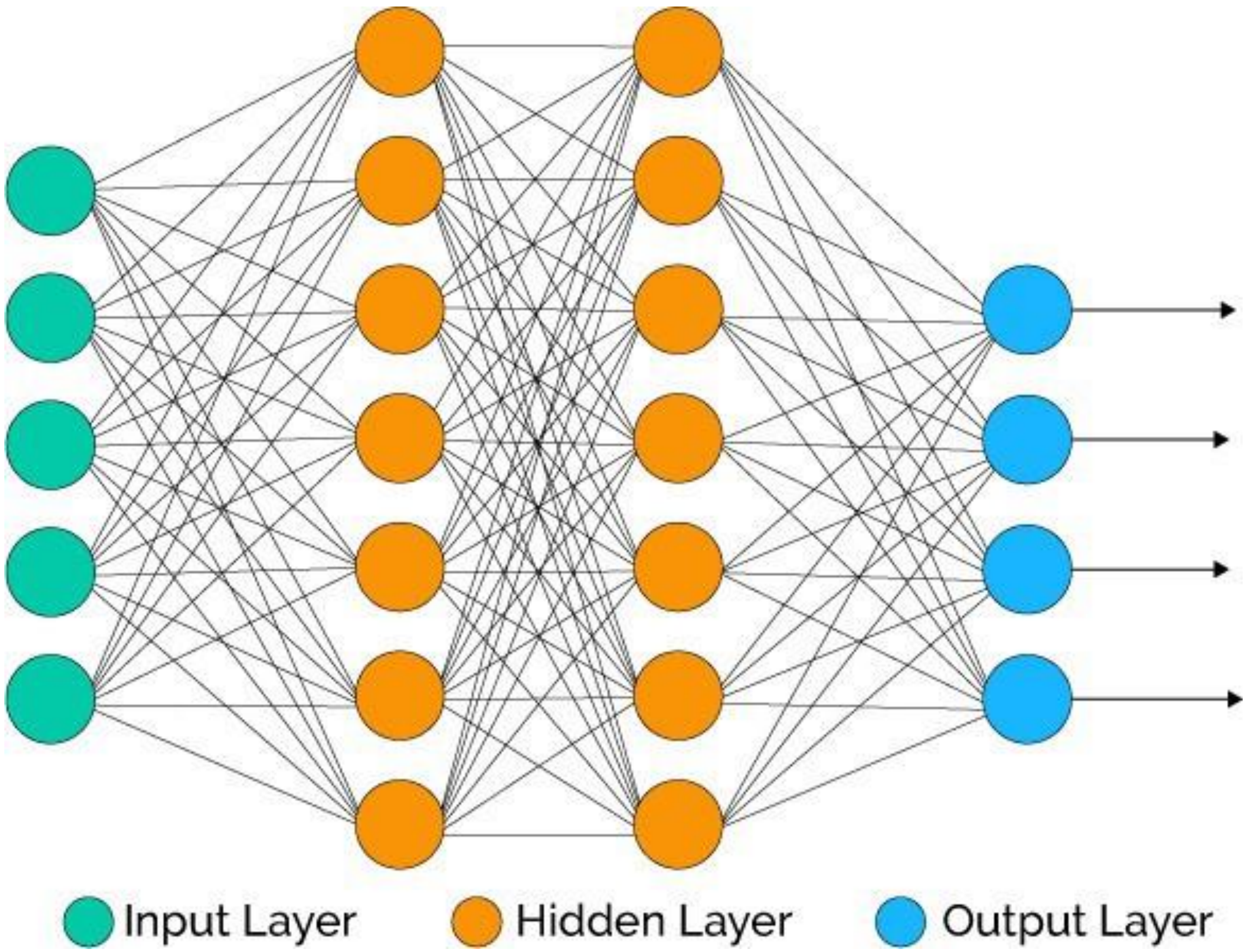
Neural Network

- ➢**Nodes**
- ➢**Edges**
- ➢**Weights** (and bias)
- ➢**Activation** (non-linear)

  (Sigmoid, Tanh, ReLU)

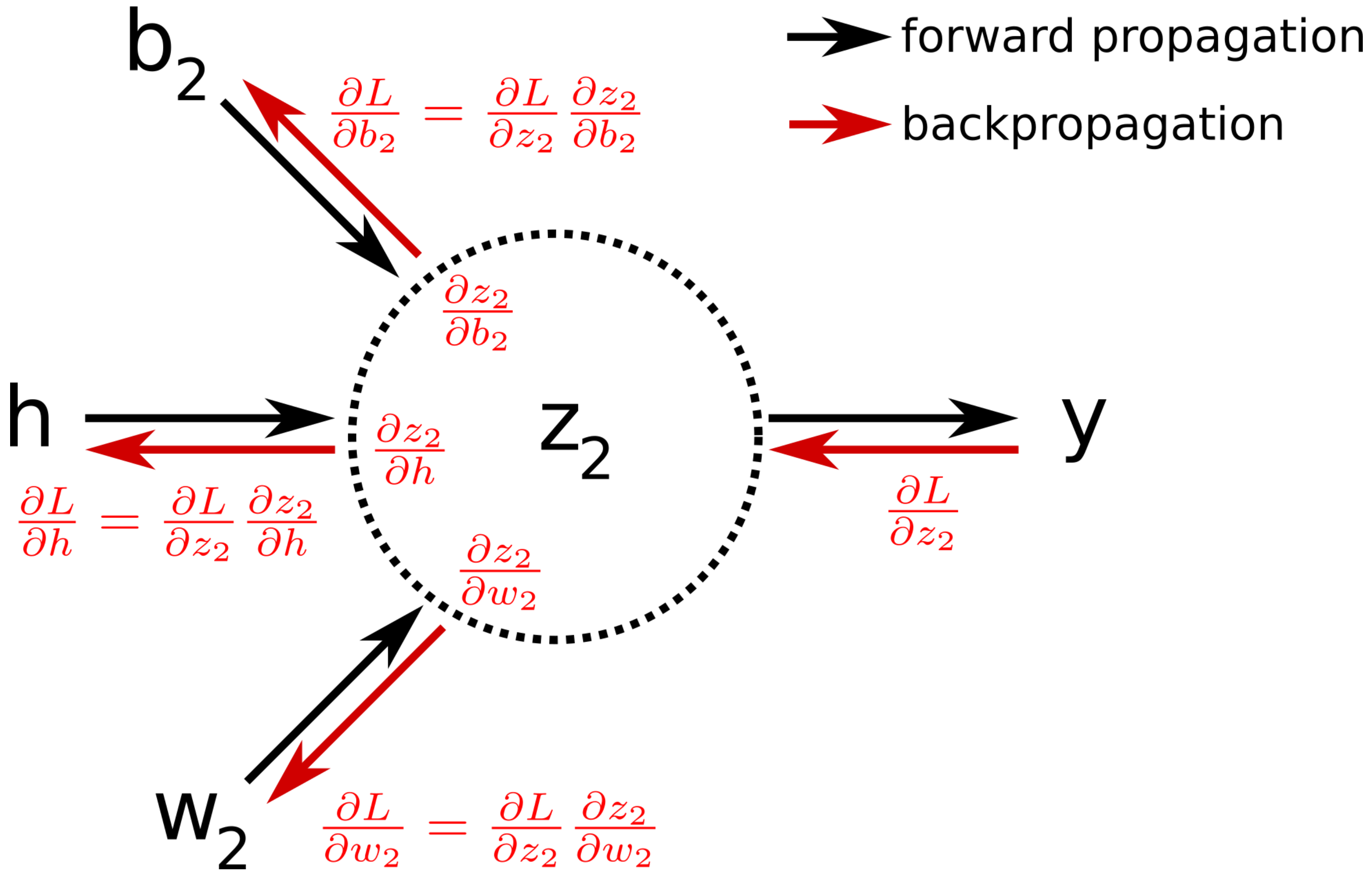
- ➢**Softmax**





Backpropagation

- Compute the error and its derivative at each node !
- Update weight with gradient descent



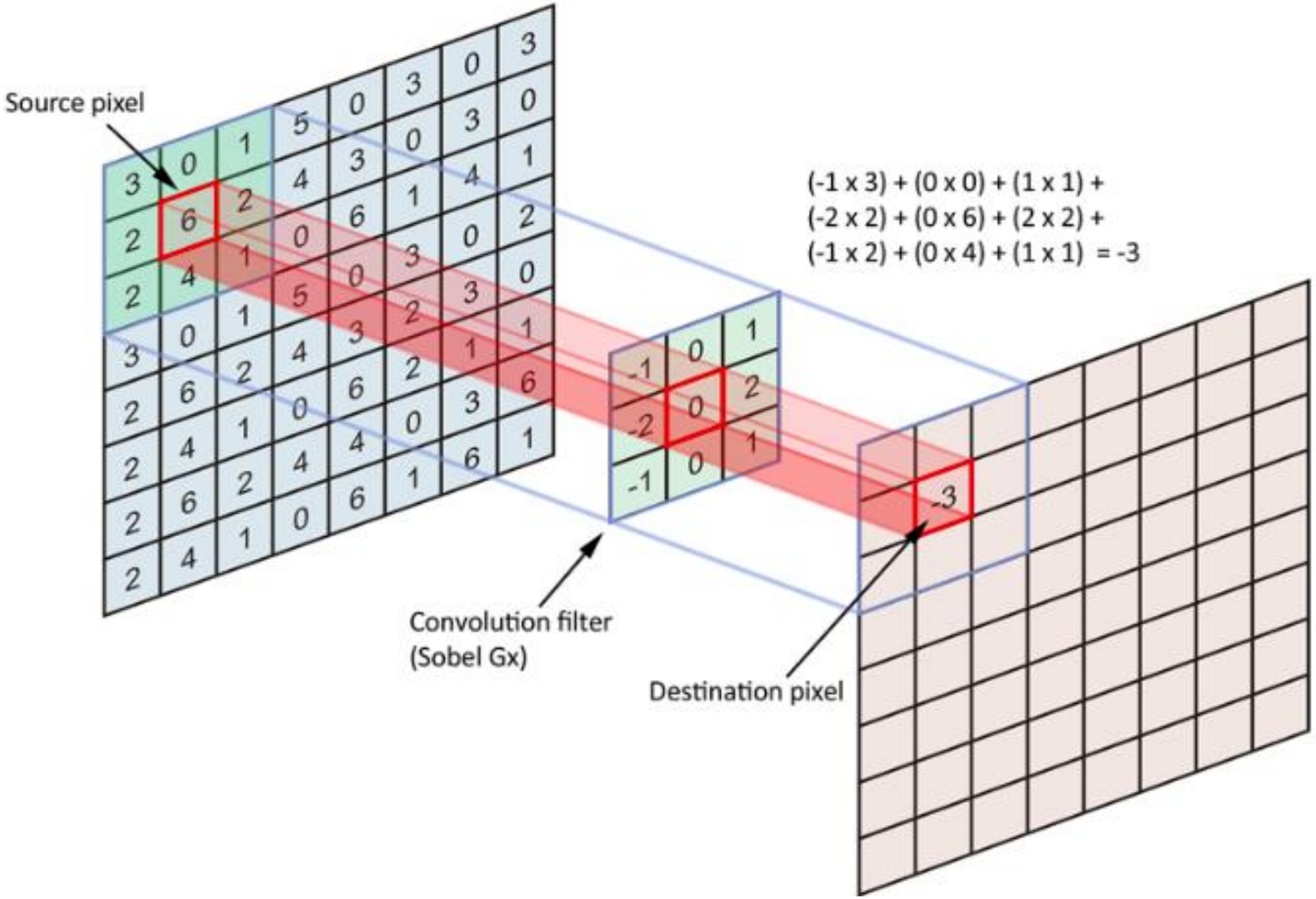
Machine Learning

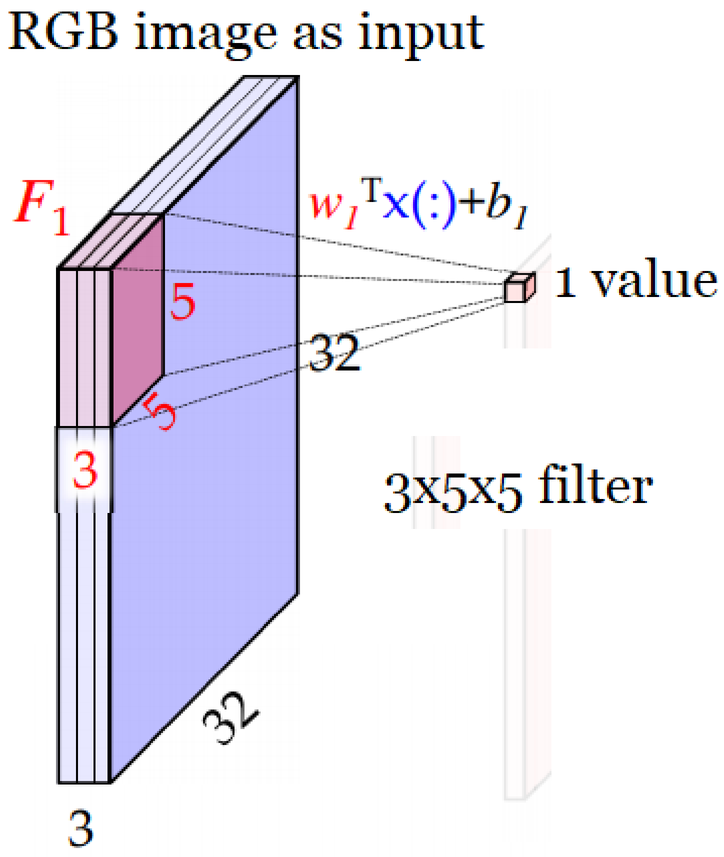
1. Introduction
2. ML Algorithms and components
3. Neural Networks
4. **Convolutional Neural Networks**

Convolutional Neural Networks

1. Convolutional filters
2. Convolutional layers
3. Pooling layers
4. Dense layers
5. Deep learning
6. Transfer learning

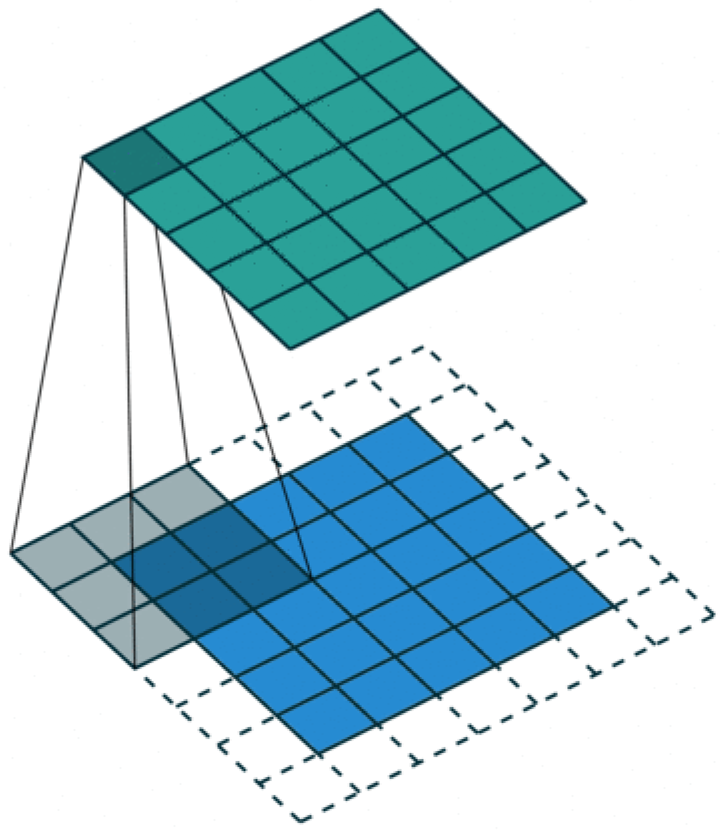
Convolutional Filters



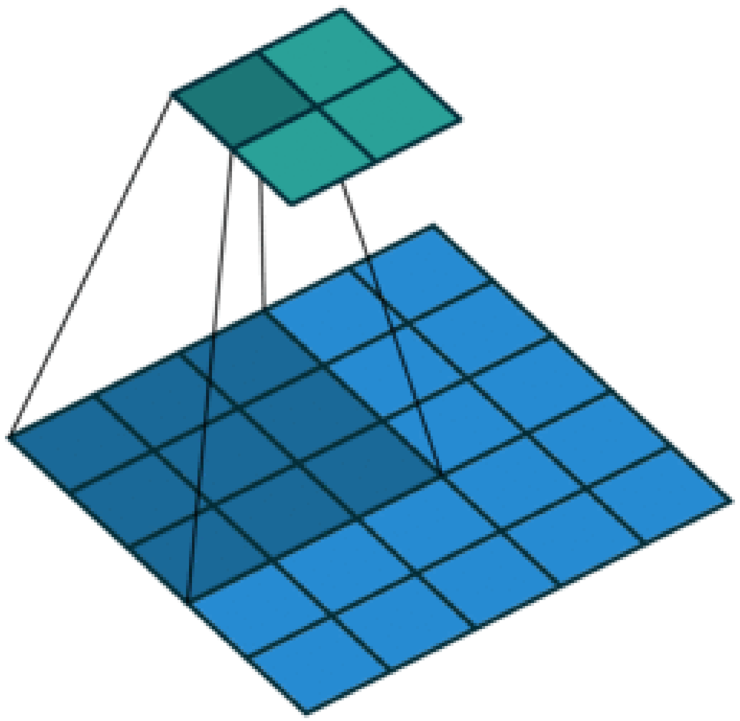


Convolutional Filters

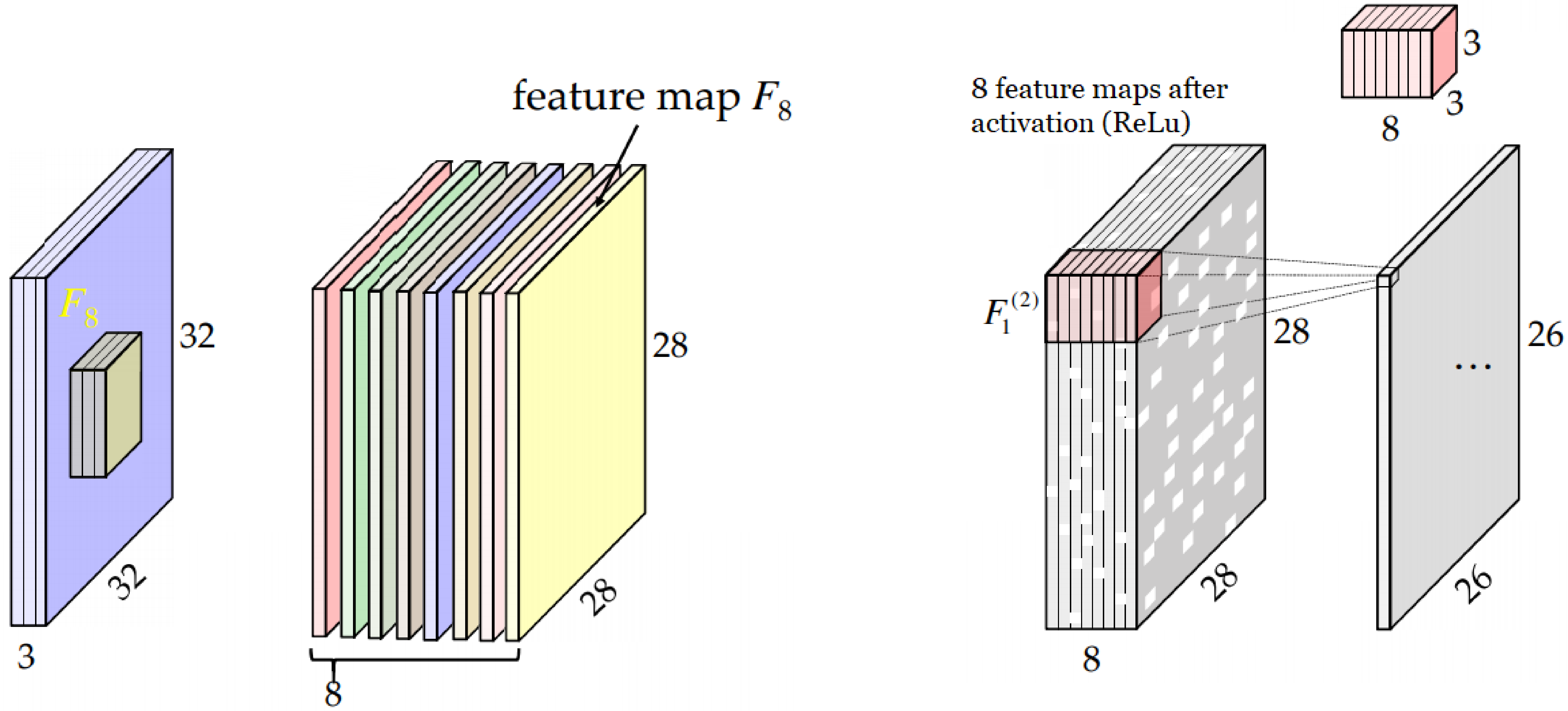
Padding = 1



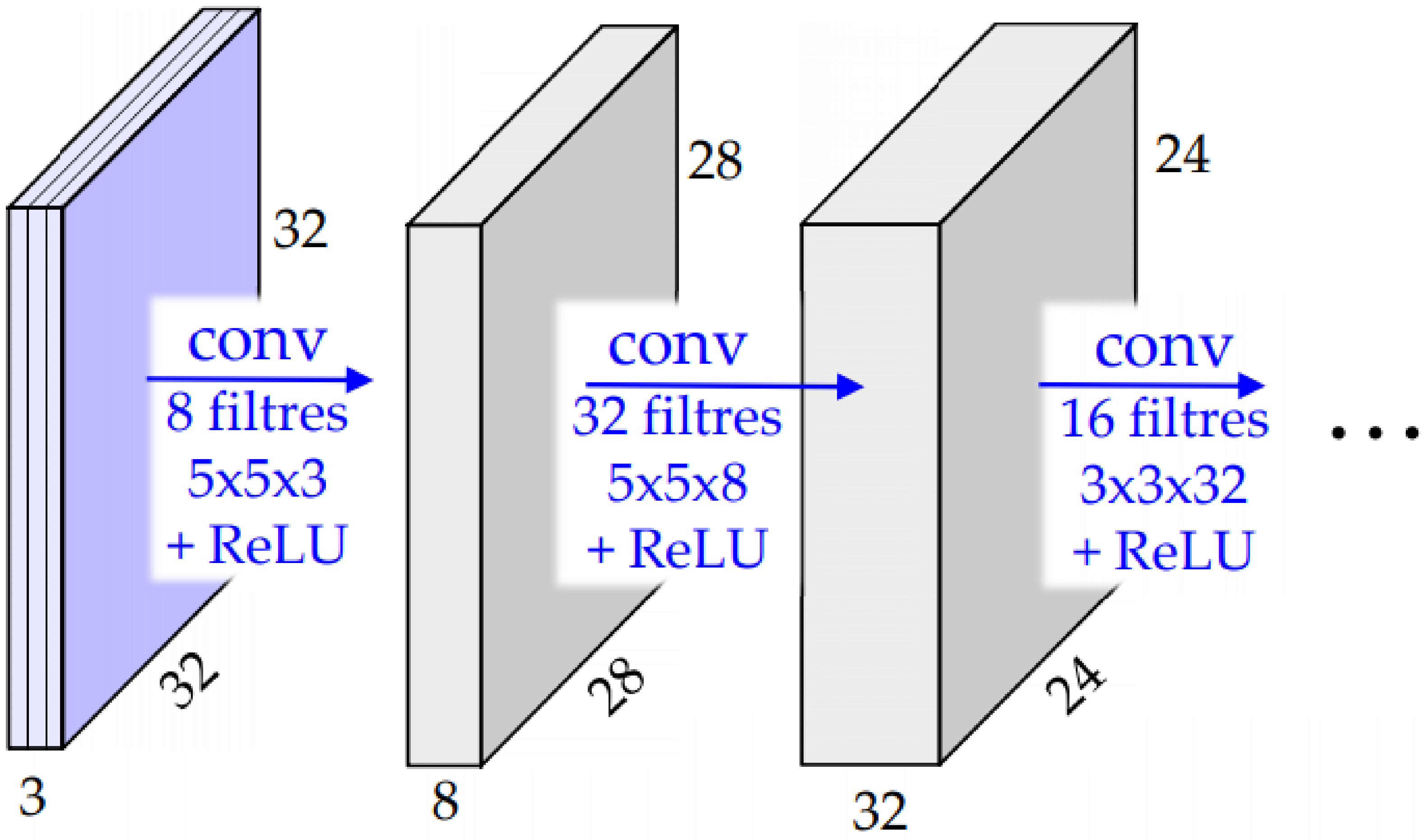
Stride = 2



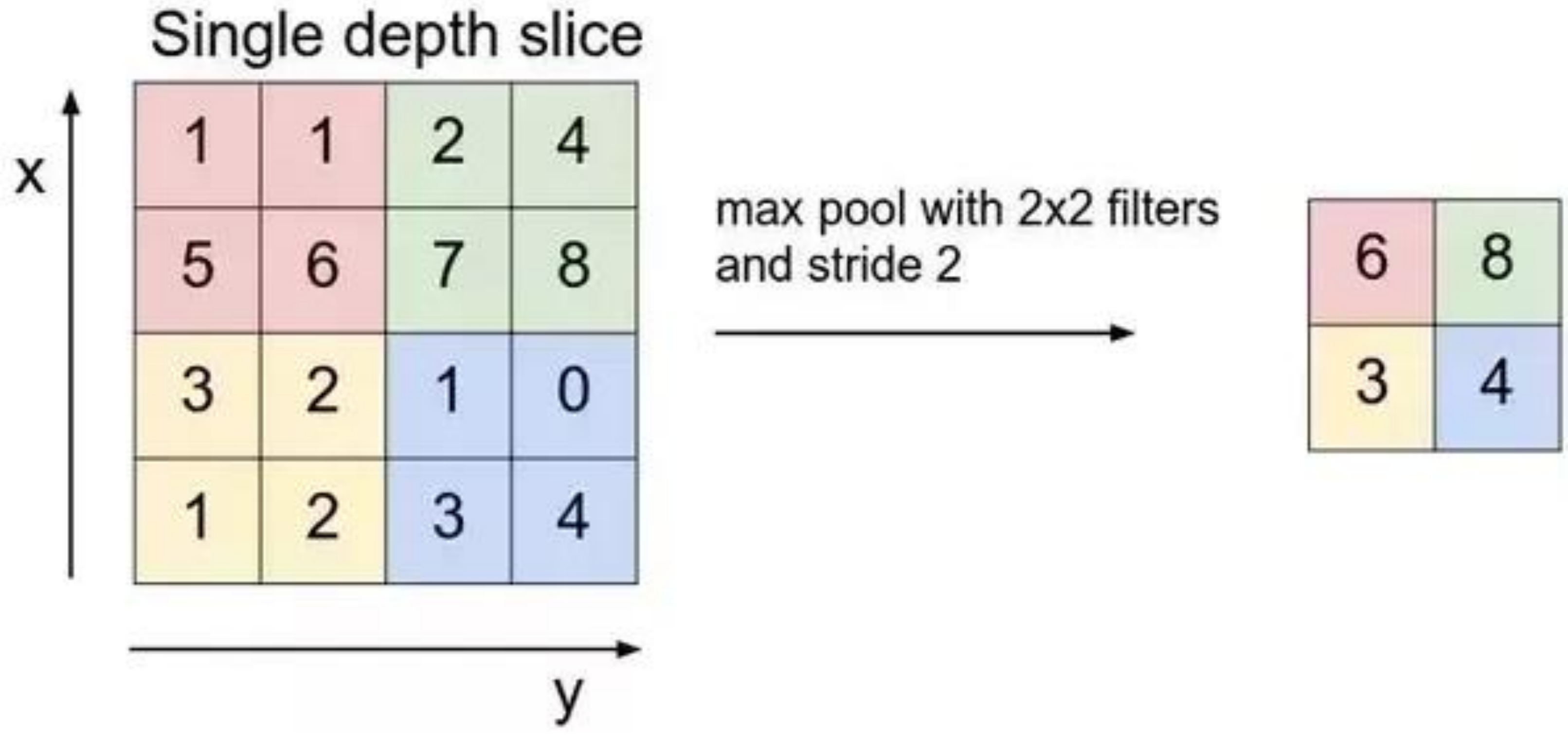
Convolutional Layers



Convolutional Layers



Pooling Layers (scales down the output size, like *stride*)

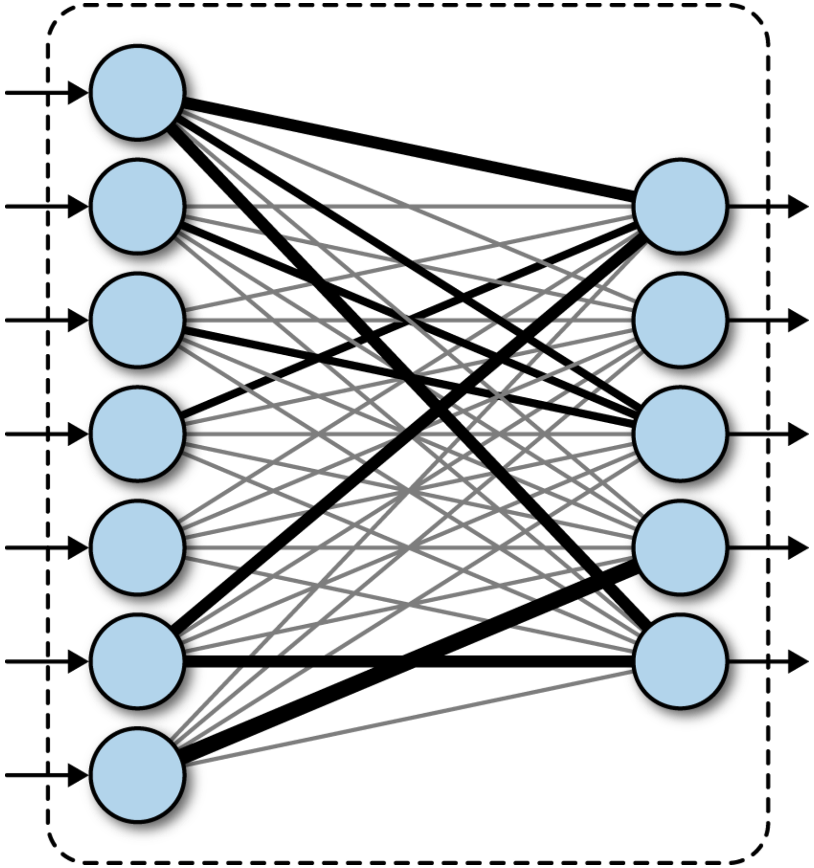


Dense layers   or *fully-connected* layers

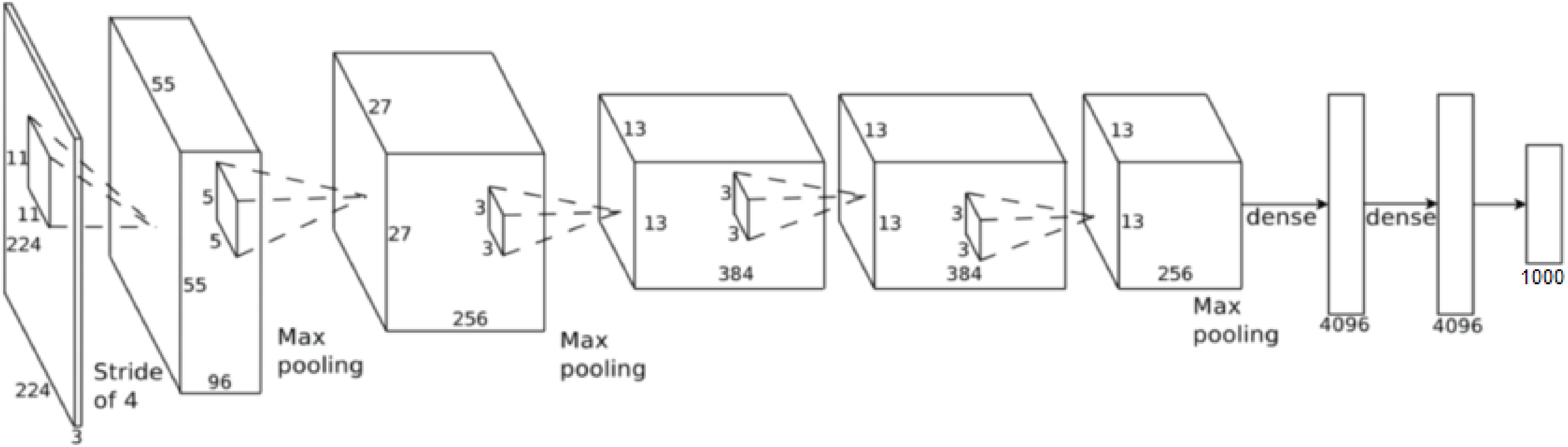
Simple Neural Network layer

- ➢Used for final classification from a flattened array

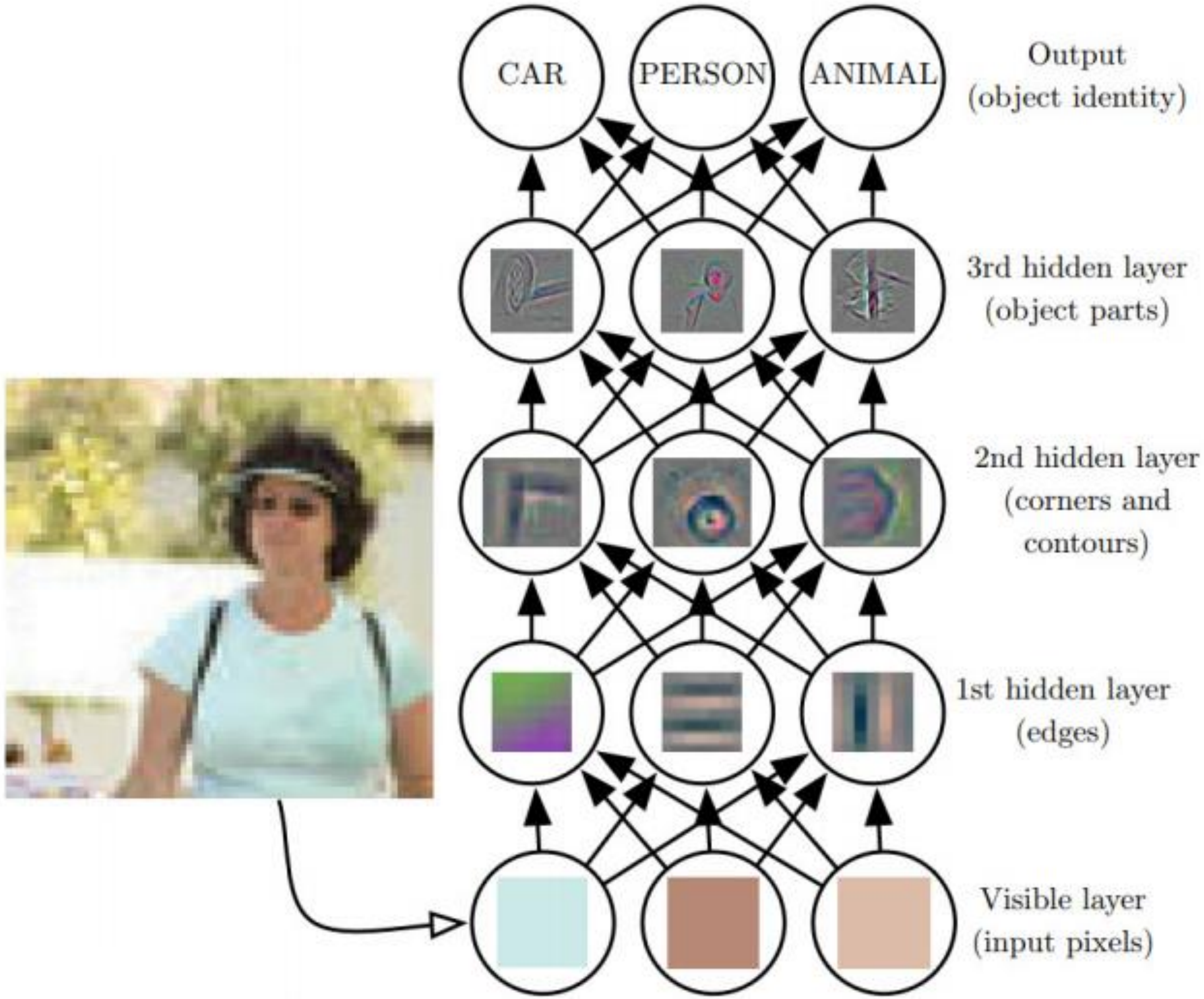
a.k.a. the classifier

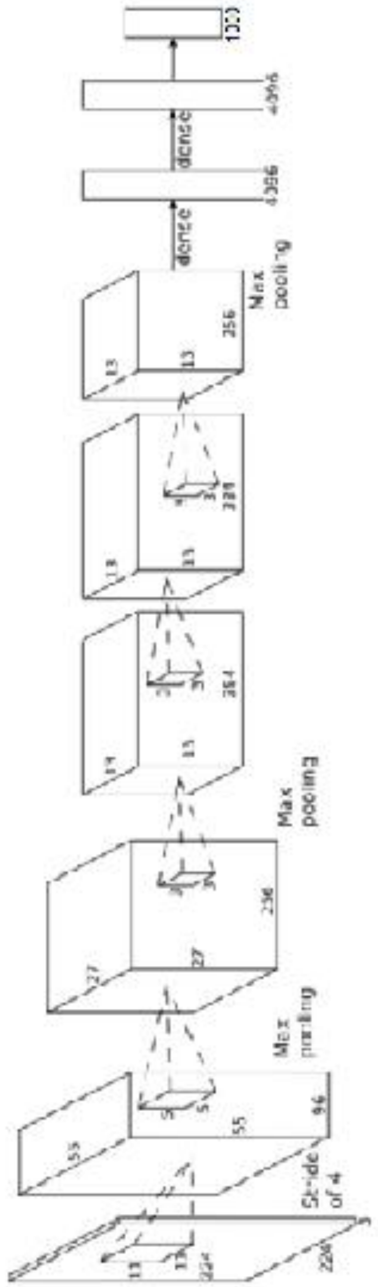


Deep Learning

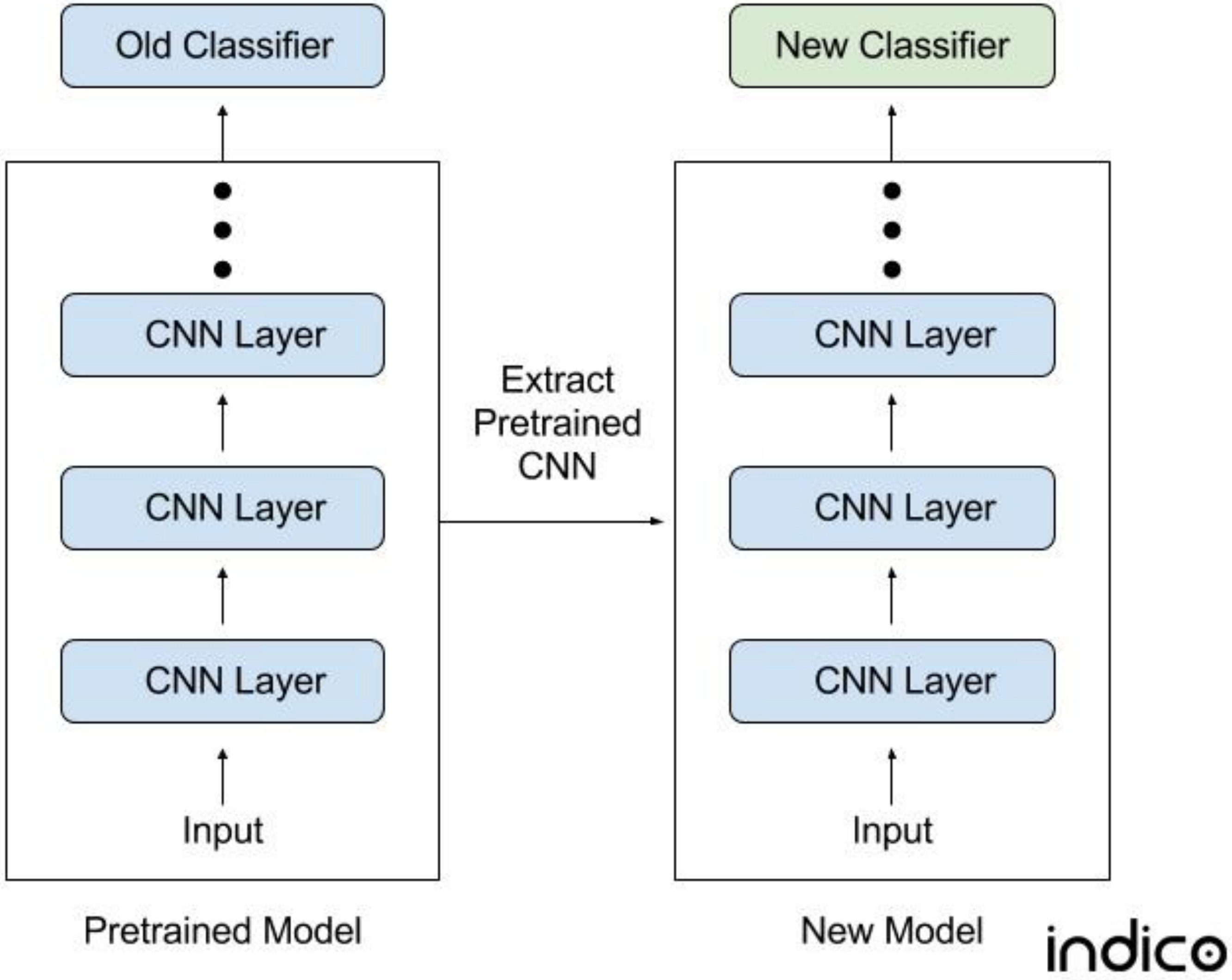


*AlexNet'*s architecture (half) for image classification into 1000 different classes.


CAR
PERSON
ANIMAL
Output
(object identity)
3rd hidden layer
(object parts)
2nd hidden layer
(corners and
contours)
1st hidden layer
(edges)
Visible layer
(input pixels)



Transfer Learning



The architecture depends on the problem

Classification	➢ Convolutional NN, ResNets …
Semantic Classification	➢ U-Net
Denoising	➢ Autoencoder / U-Net
Speech recognition (time-dependent)	➢ Recurrent Neural Network (RNN)
…	➢ …

Procedure

- Load data and labels

  > Might have to resize images, apply filters...
  > If its too big to load, think about writing a DataGenerator class.

- Split dataset into training, validation and test

  > Usually around 60% training, 20% validation, 20% test.
  > Make sure the data is different for each set

- Scale

  > Simple normalization (remove minimum, divide by max value)
  > Or StandardScale (unit-variance and zero-mean)

- Build the model

  > Look at other models online, recycle, transfer learning if possible.
  > Compile with an appropriate loss function and optimizer for your problem.

- Train

  > Batch size usually around 32, 64, 128.
  > Use callbacks for early stopping, checkpoints. Then nb. of epochs can be anything high enough...

- Evaluate on test set

Coding Example

Supervised classification