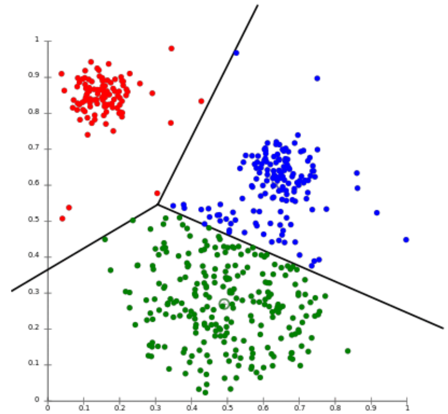
LESSON 19

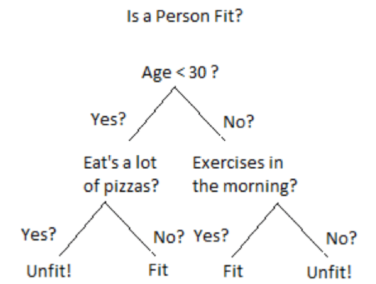
Clustering (k means).
Decision Trees.
Main intelligent systems approaches:
Neural Networks, Fuzzy systems, Evolutionary,
Hybrid systems

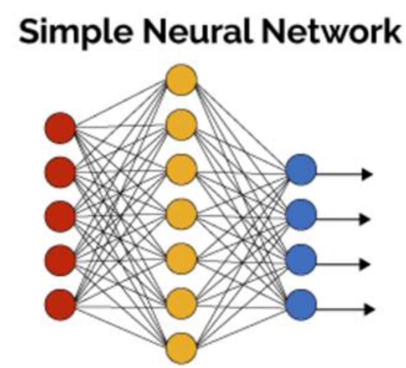


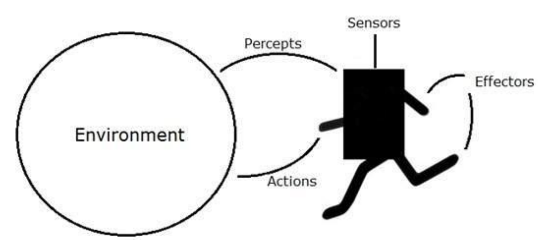

Outline

- Clustering (k means)
- Decision Trees
- Main types of computational intelligence systems
  - Artificial neural networks
  - Fuzzy systems
  - Evolutionary systems
  - Agents
  - Hybrid systems
  - Other systems



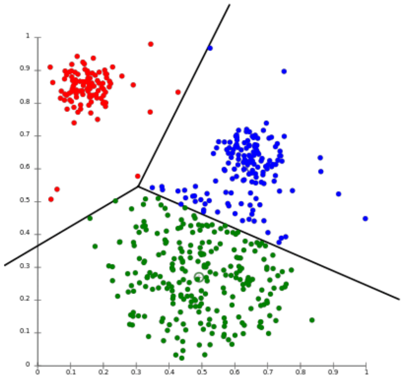




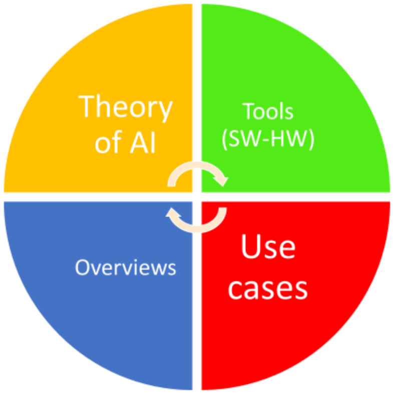


THEORY

Clustering: k-Means

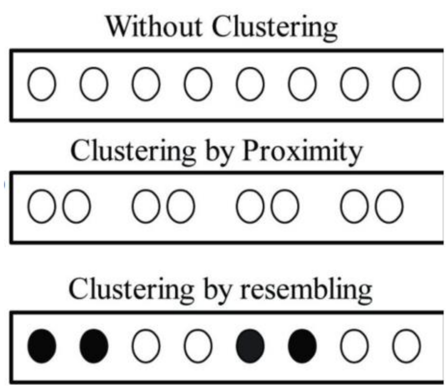


An unsupervised method

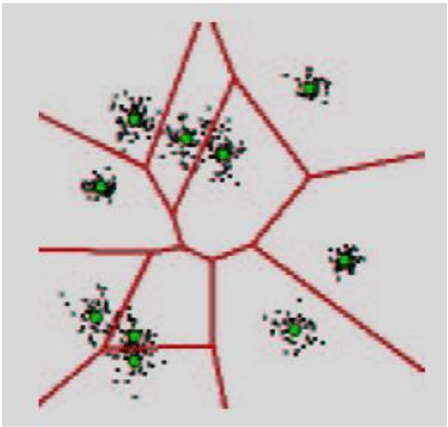


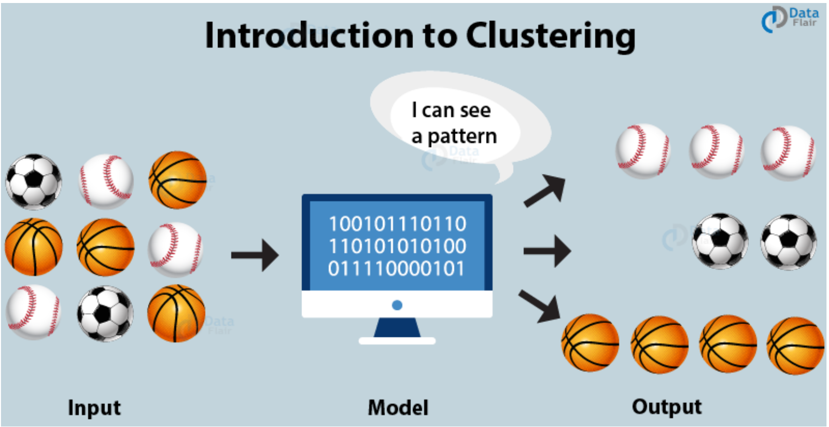
Clustering

Cluster analysis or clustering
is the task of grouping a set of objects in such a way that objects in the same group (called a cluster) are more similar (position, features, shapes, …) to each other than to those in other groups (clusters)

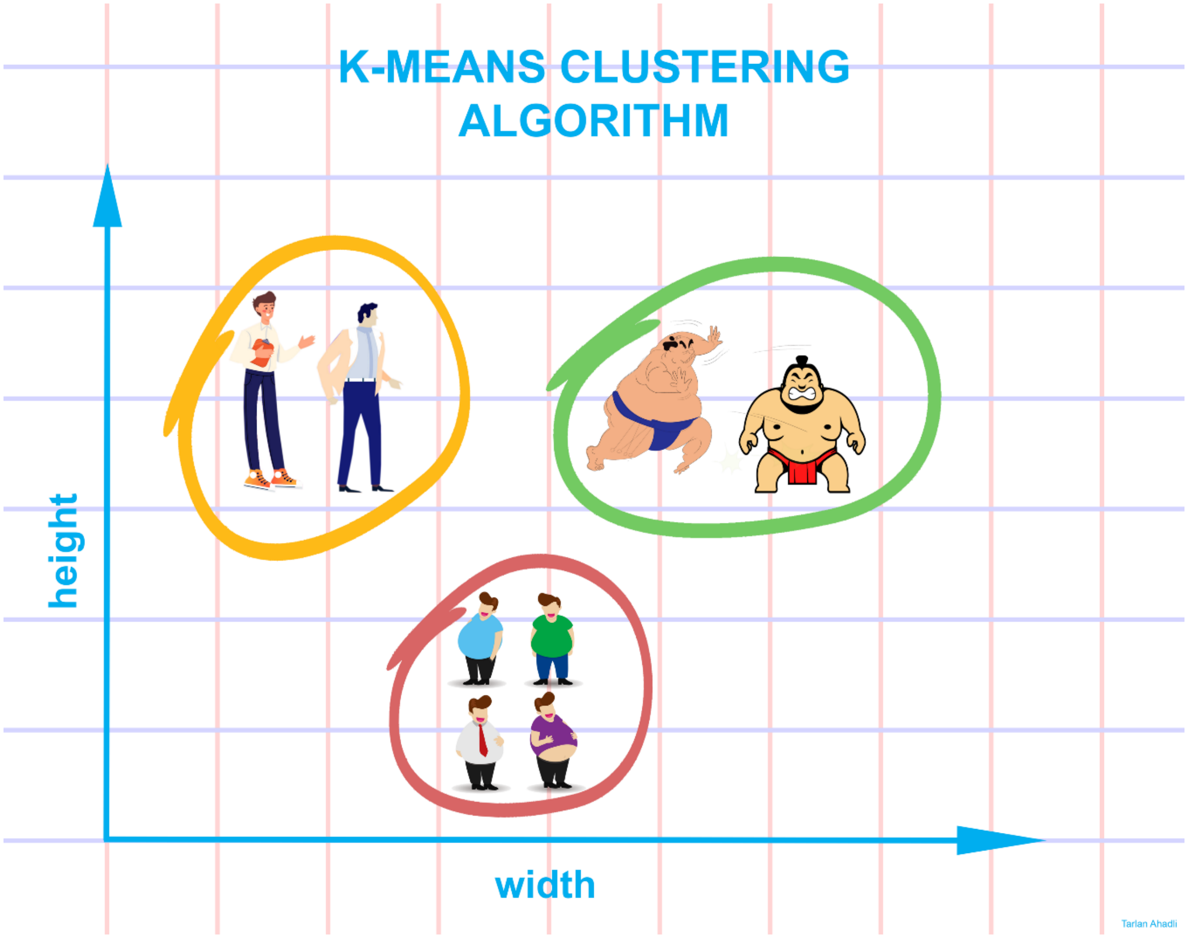


Human Gestalt





Clustering



Clustering

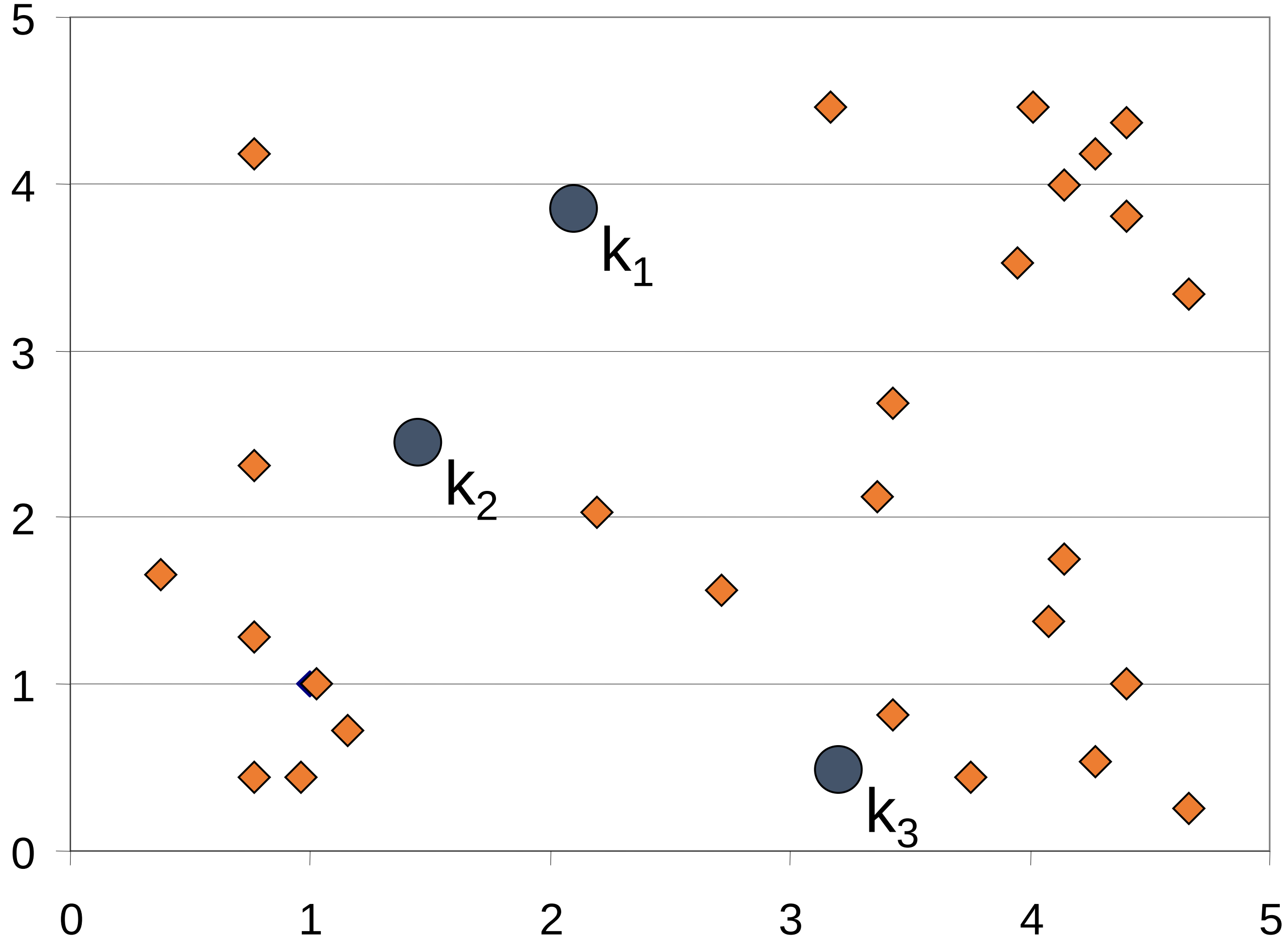
Making groups of similar features

Why do we cluster?

- **Summarizing data**
  – Look at large amounts of data
  – Patch-based compression or denoising
  – Represent a large continuous vector with the cluster number
- **Counting**
  – Histograms of texture, color, SIFT vectors
- **Segmentation**
  – Separate the image into different regions
- **Prediction**
  – Images in the same cluster may have the same labels

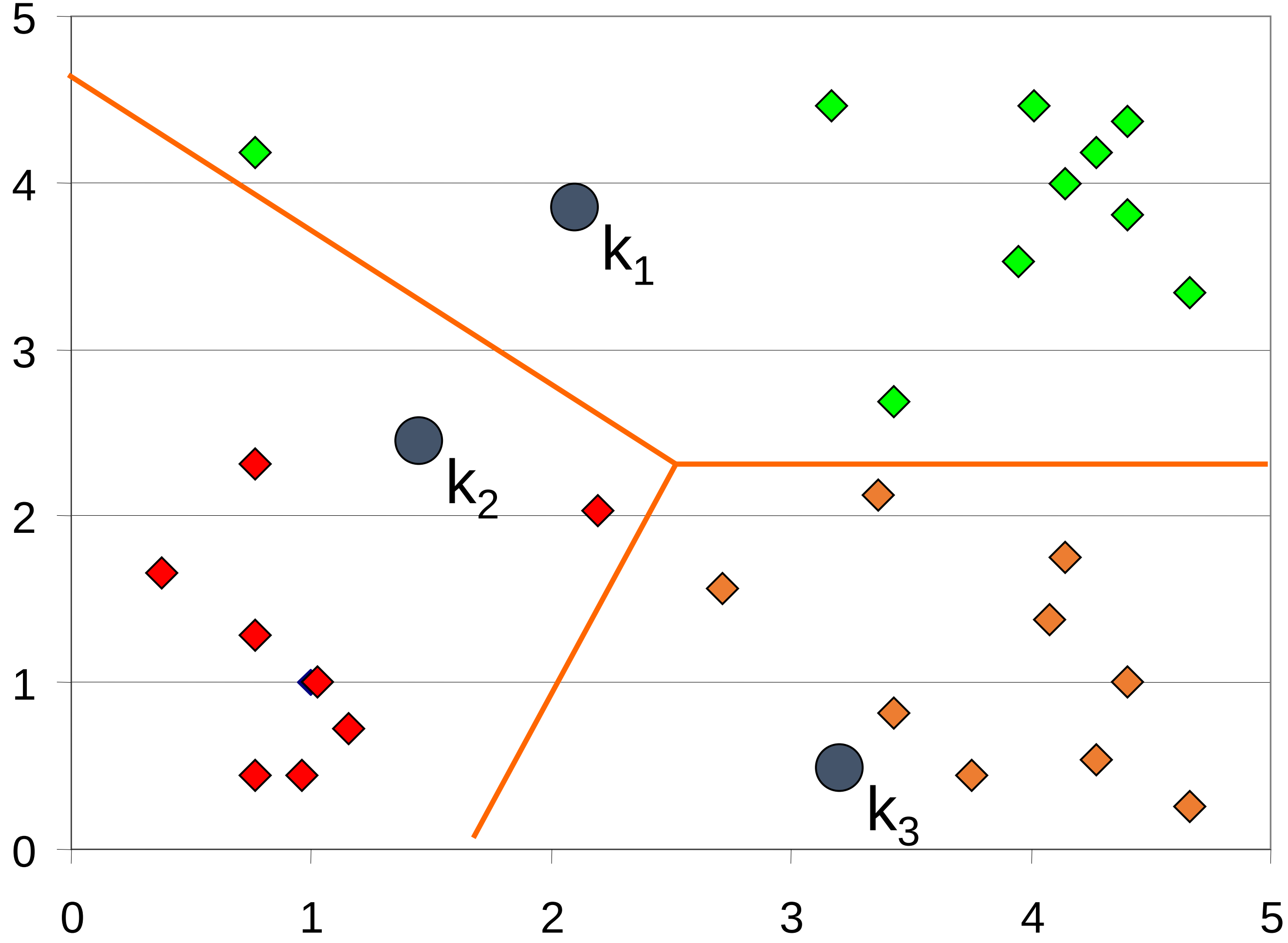
k-means Clustering: Step 1
Random initialization

Algorithm: k-means, Distance Metric: Euclidean Distance



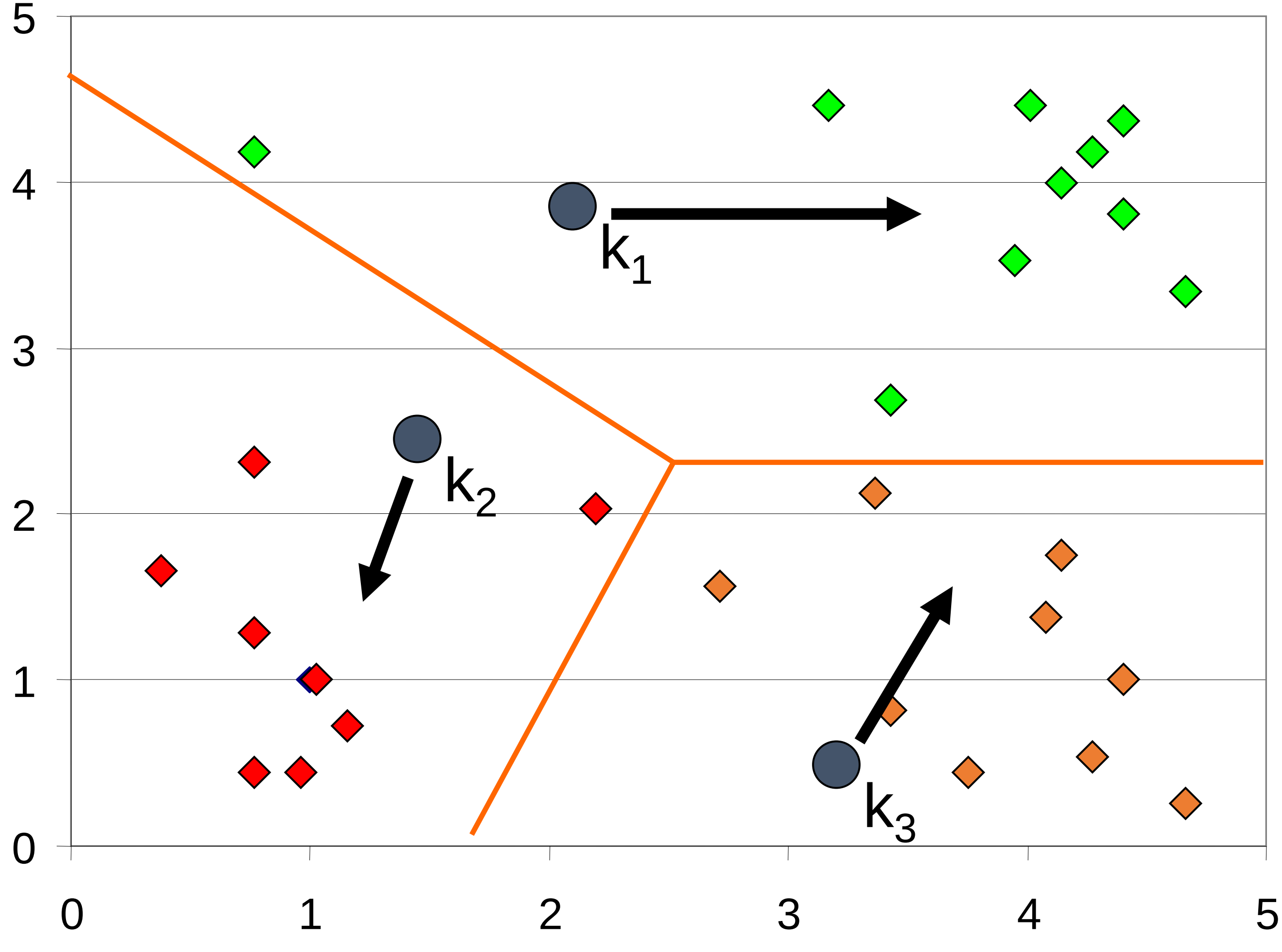
k-means Clustering: Step 2

Algorithm: k-means, Distance Metric: Euclidean Distance



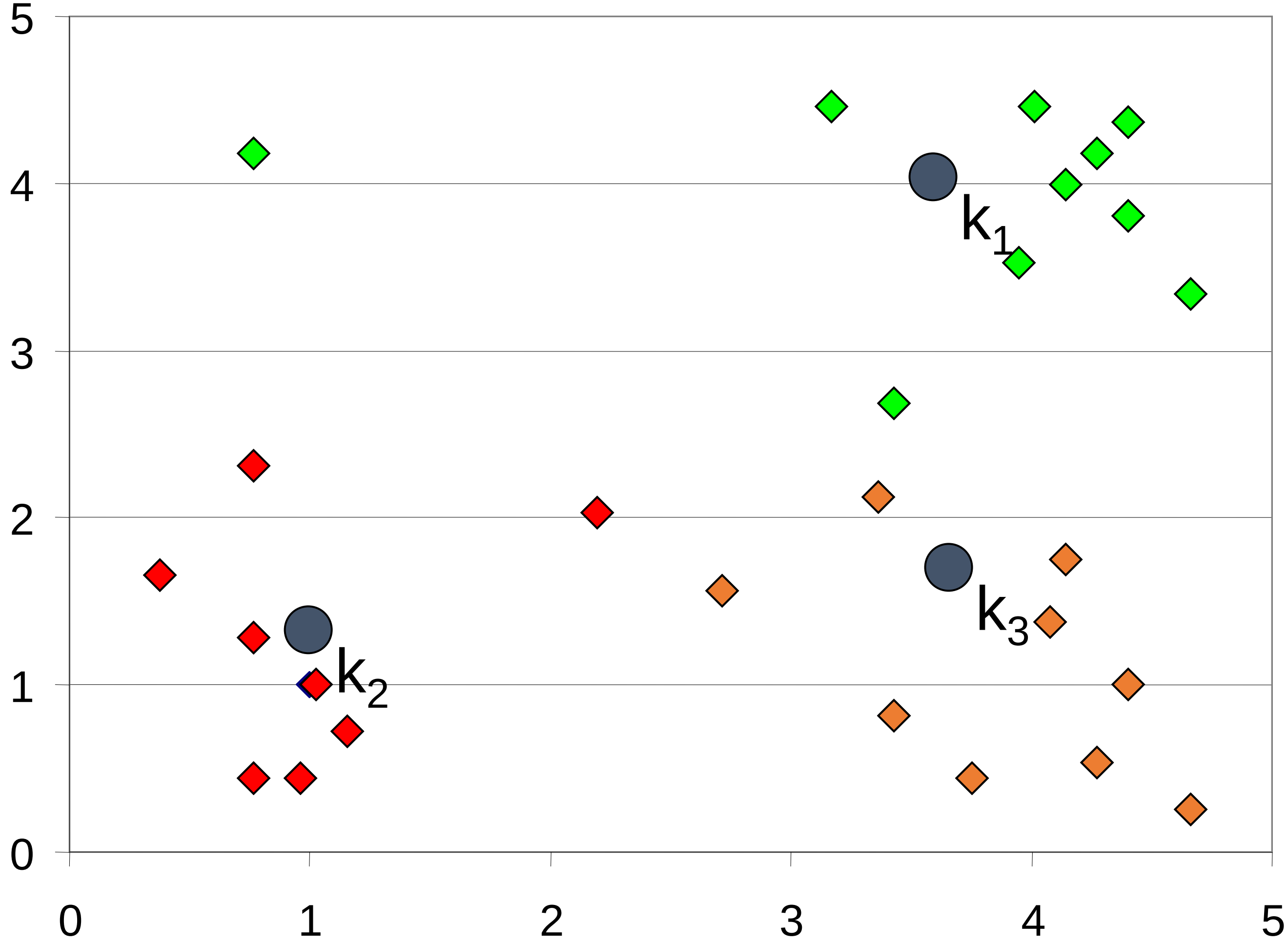
k-means Clustering: Step 2

Algorithm: k-means, Distance Metric: Euclidean Distance



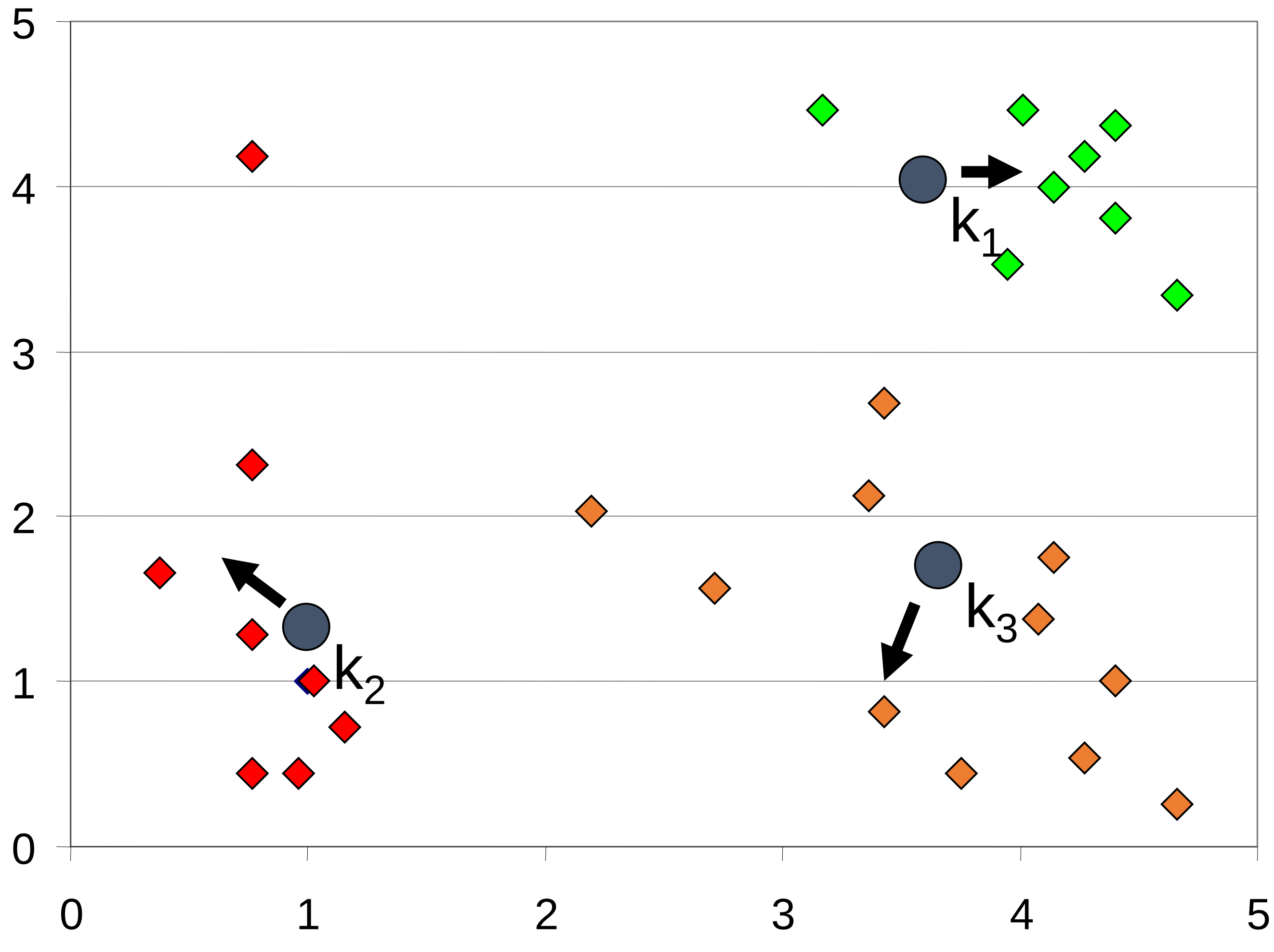
k-means Clustering: Step 3

Algorithm: k-means, Distance Metric: Euclidean Distance



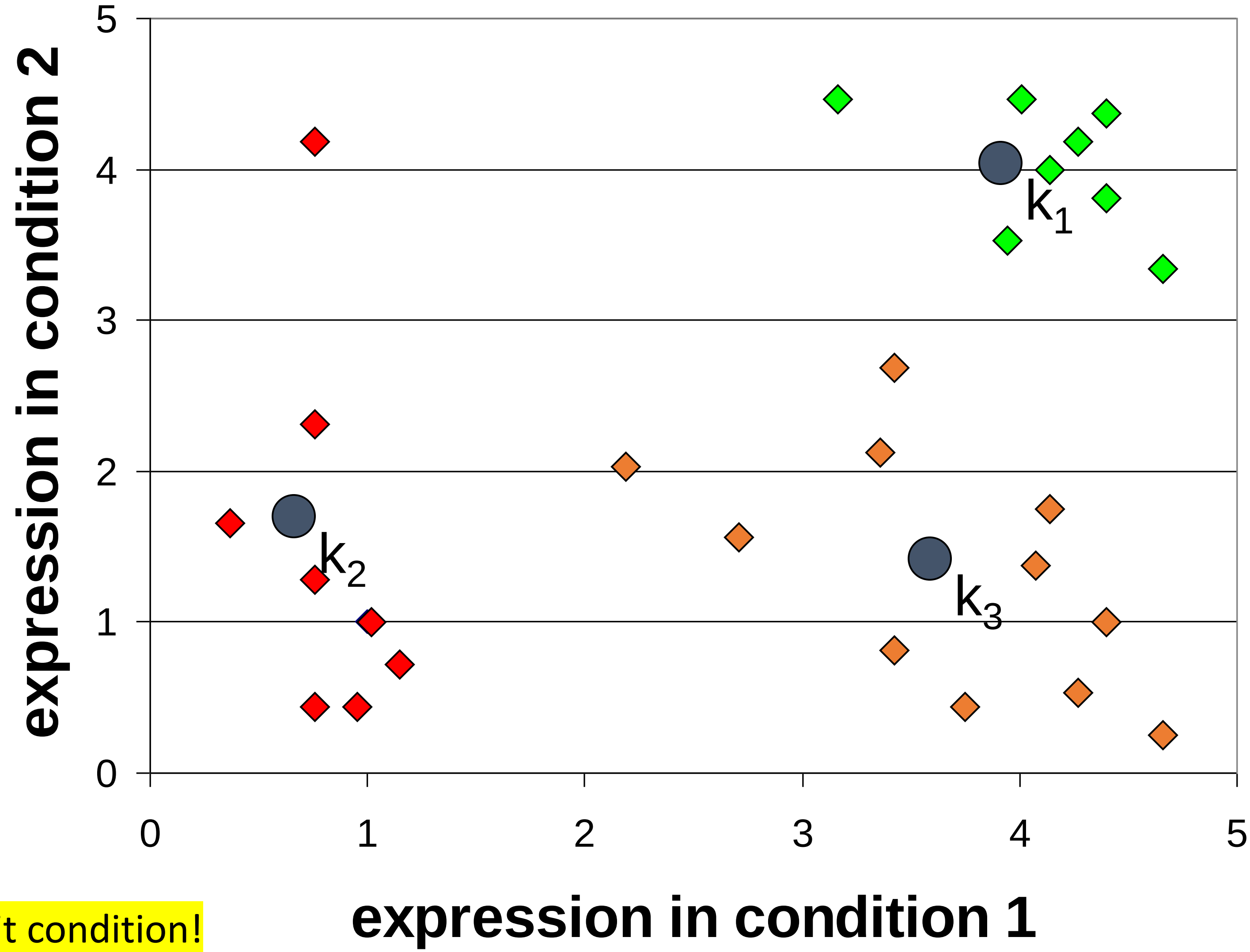
k-means Clustering: Step 4

Algorithm: k-means, Distance Metric: Euclidean Distance



k-means Clustering: Step 5

Algorithm: k-means, Distance Metric: Euclidean Distance



That's the exit condition!
Centers are not moving any more

Clustering using k-means

- Given (x*1*, x2, …, xn) observations,
  k-means clustering aims to partition the n observations into k (≤ n) sets S = {S_1, S_2, …, S_k} so so as to minimize the within-cluster sum of squares.

- In other words, its objective is to find:

$$\underset{\mathbf{S}}{\arg\min} \sum_{i=1}^{k} \sum_{\mathbf{x} \in S_i} \|\mathbf{x} - \boldsymbol{\mu}_i\|^2$$

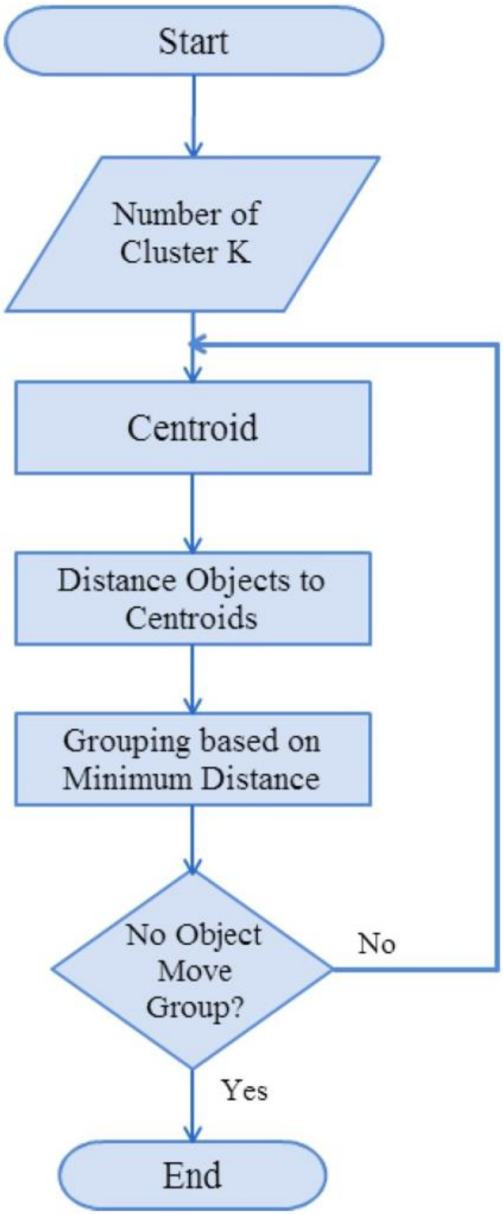
cluster center

Easy to compute ***μ*** given **S** and vice versa.

k (≤ n) sets S = {S_1, S_2, …, S_k}

$$\underset{\mathbf{S}}{\arg\min} \sum_{i=1}^{k} \sum_{\mathbf{x}\in S_i} \left\|\mathbf{x}-\boldsymbol{\mu}_i\right\|^2$$

cluster center



S

Partition Algorithm

1. Decide on a value for *k*.

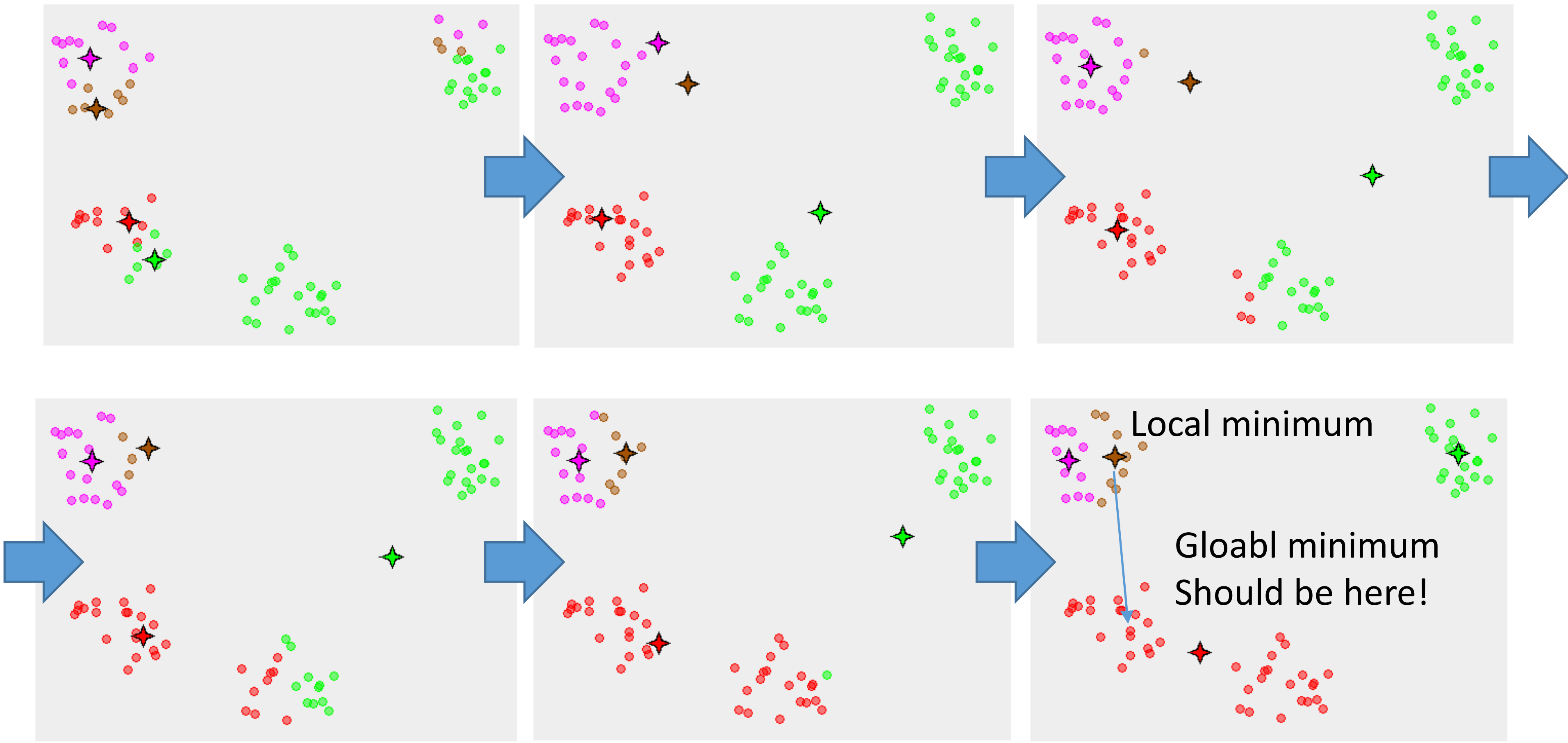
2. Initialize the *k* cluster centers (randomly, if necessary).

3. Decide the class memberships of the *N* objects by assigning them to the nearest cluster center.

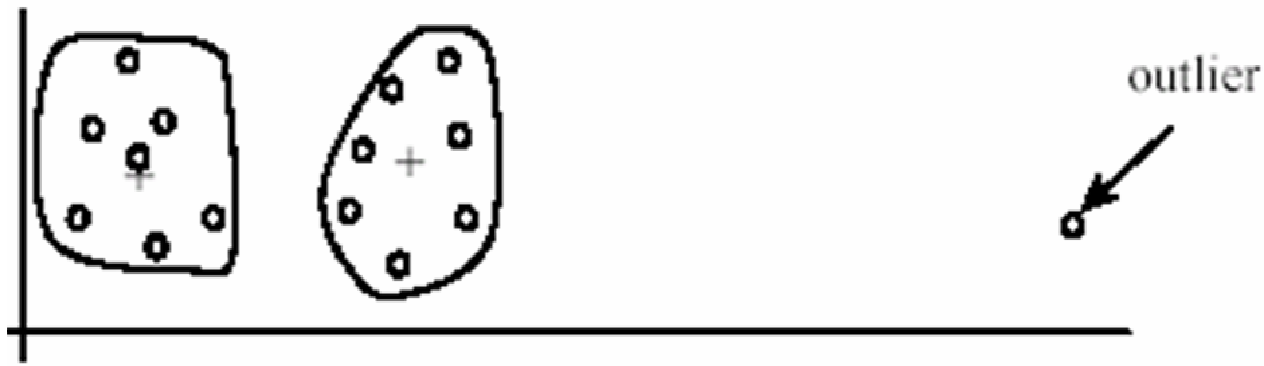
4. Re-estimate the *k* cluster centers, by assuming the memberships found above are correct.

5. If none of the *N* objects changed membership in the last iteration, exit. Otherwise goto 3.

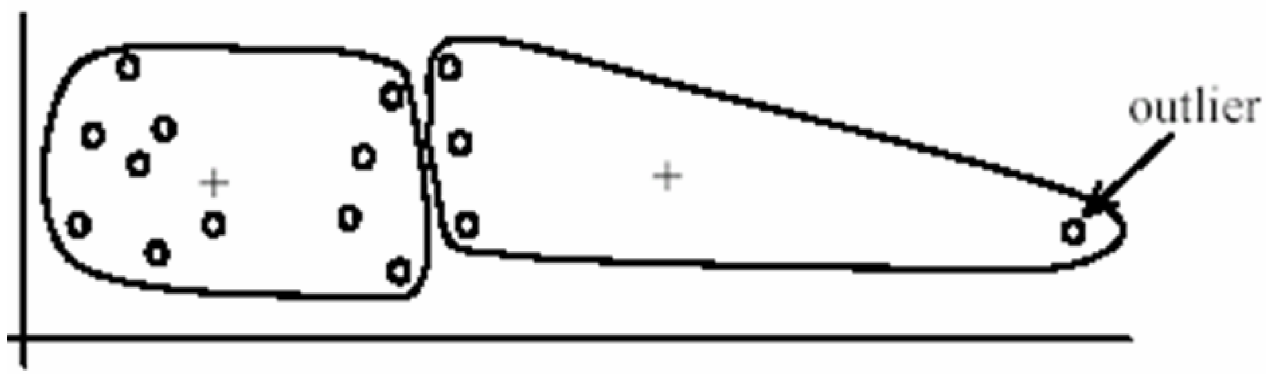
k-means converges to a local minimum



k-Means outliers problems



(B): Ideal clusters



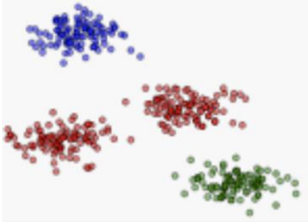
Strengths of k-Means

- *Relatively efficient*: $O(tkn)$, where
    - n is # objects,
    - k is # clusters,
    - t is # iterations.
    - Normally, k, t << n.
- Guarantees convergence
    - (even if terminates at a local minimum)
- Easily adapts to
    - New examples.
    - Generalizes to clusters of different shapes and sizes, such as elliptical clusters.

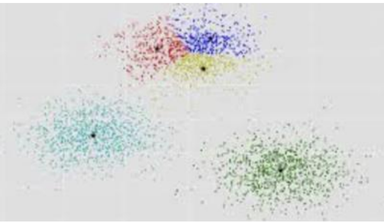
Weakness of k-Means

- Applicable only when mean is defined,
  - what about categorical data
    ("apple", 90mm), ("lemon", 45mm)
    ("orange" , 55mm), ... ?
- Need to specify *k*,
  the number of clusters,
  in advance

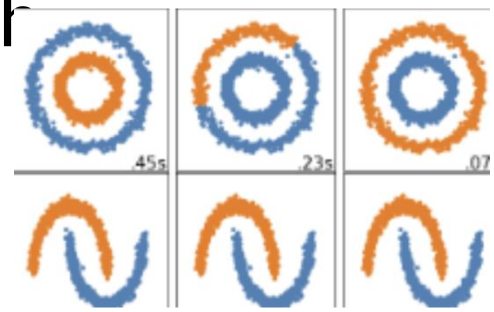
k=3



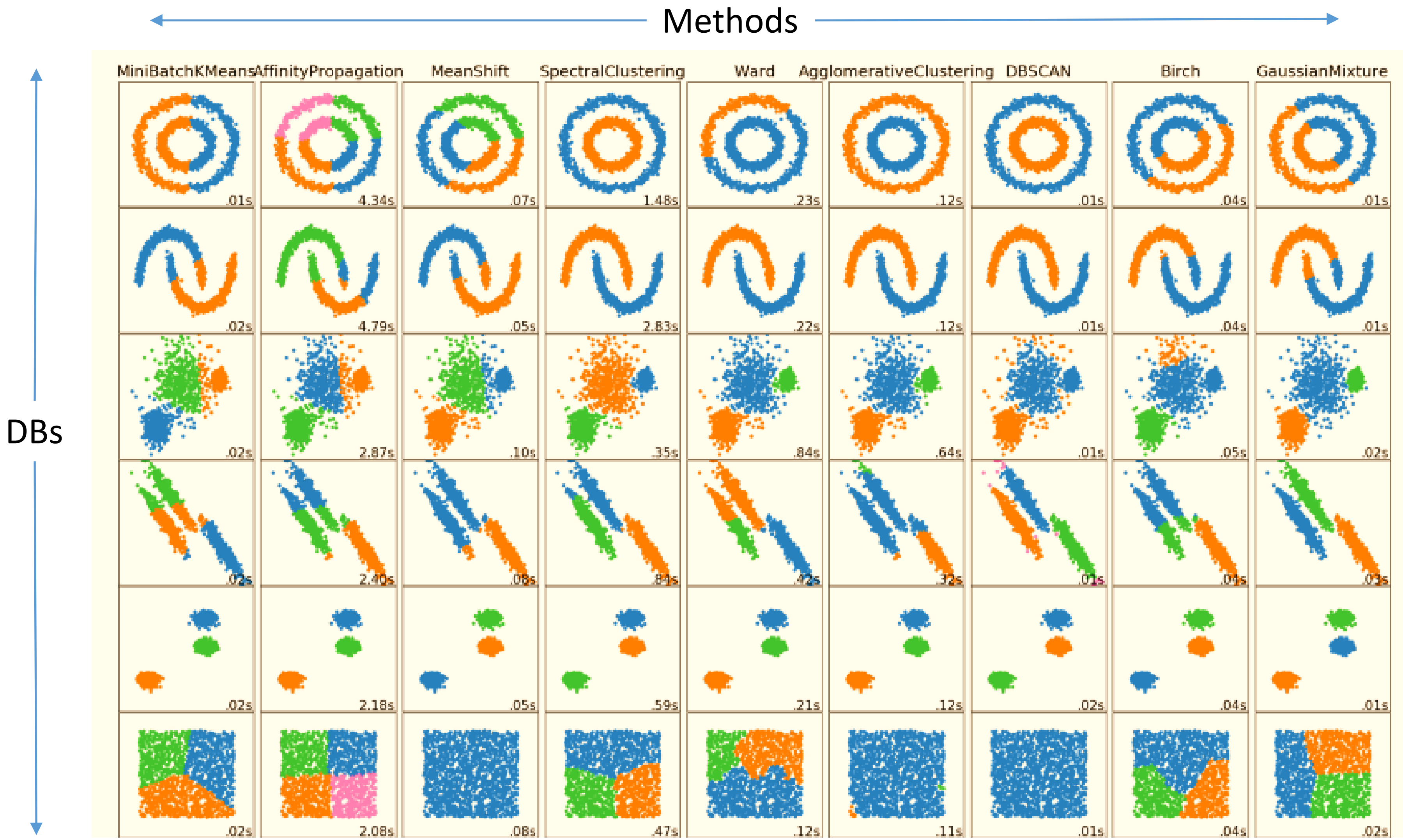
k=5



- Unable to handle noisy data and outliers
- Not suitable to discover clusters with
  non-convex shapes



Different behavior of the clustering methods



Colab k-means coding

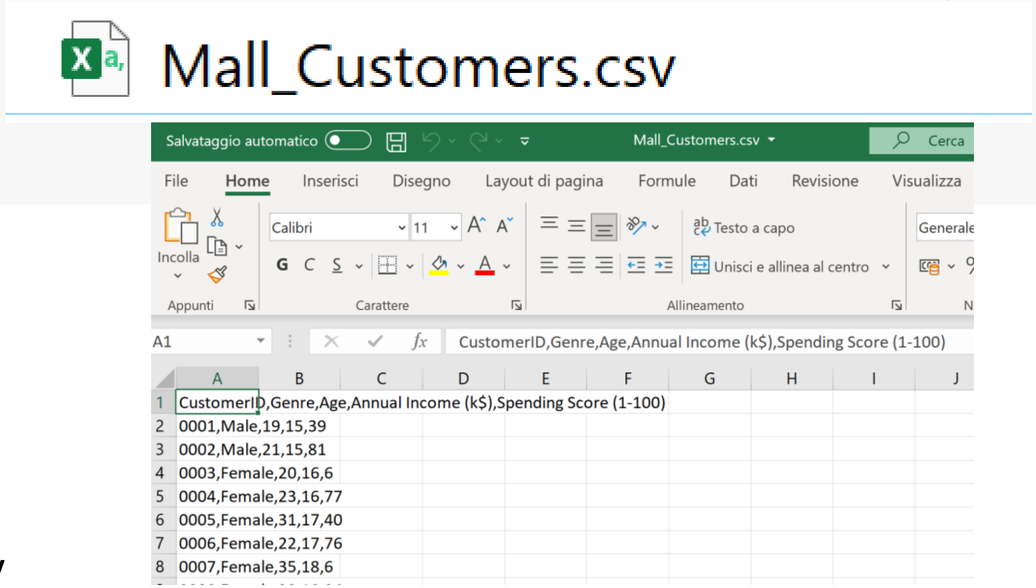
```
[1] from google.colab import files
    uploaded = files.upload()
```

Scegli file Mall_Customers.csv

- ***Mall_Customers.csv***(*application/vnd.ms-excel) - 4286 bytes, last modified: 25/9/2019 - 100% done*

```
Saving Mall_Customers.csv to Mall_Customers.csv
```

```
#import libraries
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
```



pandas - Python Data Analysis Library

Import data and feature selection

```
#import data set
dataset = pd.read_csv('Mall_Customers.csv')   # CustomerID,Genre,Age,Annual Income (k$),Spending Score (1-100)
print(dataset.shape)
#cut the dabase: only  Age, Spending Score (1-100)
X = dataset.iloc[:,[2,4]].values
X[0:9]
```

```
(200, 5)
array([[19, 39],
       [21, 81],
       [20,  6],
       [23, 77],
       [31, 40],
       [22, 76],
       [35,  6],
       [23, 94],
       [64,  3]])
```

	A	B	C	D	E	F
1	CustomerID,Genre,Age,Annual Income (k$),Spending Score (1-100)					
2	0001,Male,19,15,39					
3	0002,Male,21,15,81					
4	0003,Female,20,16,6					
5	0004,Female,23,16,77					
6	0005,Female,31,17,40					
7	0006,Female,22,17,76					
8	0007,Female,35,18,6					

Data exploration 1 .head

```
#DATA EXPLORATION
dataset.head(10)
```

	CustomerID	Genre	Age	Annual Income (k$)	Spending Score (1-100)
0	1	Male	19	15	39
1	2	Male	21	15	81
2	3	Female	20	16	6
3	4	Female	23	16	77
4	5	Female	31	17	40
5	6	Female	22	17	76
6	7	Female	35	18	6
7	8	Female	23	18	94
8	9	Male	64	19	3
9	10	Female	30	19	72

Data exploration 2 .shape

```
dataset.shape
```

(200, 5)

check

	A	B	C	D	E	F
1	CustomerID,Genre,Age,Annual Income (k$),Spending Score (1-100)					
2	0001,Male,19,15,39					
3	0002,Male,21,15,81					
4	0003,Female,20,16,6					
5	0004,Female,23,16,77					
6	0005,Female,31,17,40					
7	0006,Female,22,17,76					
8	0007,Female,35,18,6					

```
dataset.describe().T
```

	count	mean	std	min	25%	50%	75%	max
CustomerID	200.0	100.50	57.879185	1.0	50.75	100.5	150.25	200.0
Age	200.0	38.85	13.969007	18.0	28.75	36.0	49.00	70.0
Annual Income (k$)	200.0	60.56	26.264721	15.0	41.50	61.5	78.00	137.0
Spending Score (1-100)	200.0	50.20	25.823522	1.0	34.75	50.0	73.00	99.0

Data exploration 3 .info

```
dataset.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 200 entries, 0 to 199
Data columns (total 5 columns):
 #   Column                  Non-Null Count  Dtype 
---  ------                  --------------  ----- 
 0   CustomerID              200 non-null    int64 
 1   Genre                   200 non-null    object
 2   Age                     200 non-null    int64 
 3   Annual Income (k$)      200 non-null    int64 
 4   Spending Score (1-100)  200 non-null    int64 
dtypes: int64(4), object(1)
memory usage: 7.9+ KB
```

Data exploration 4 .isnull

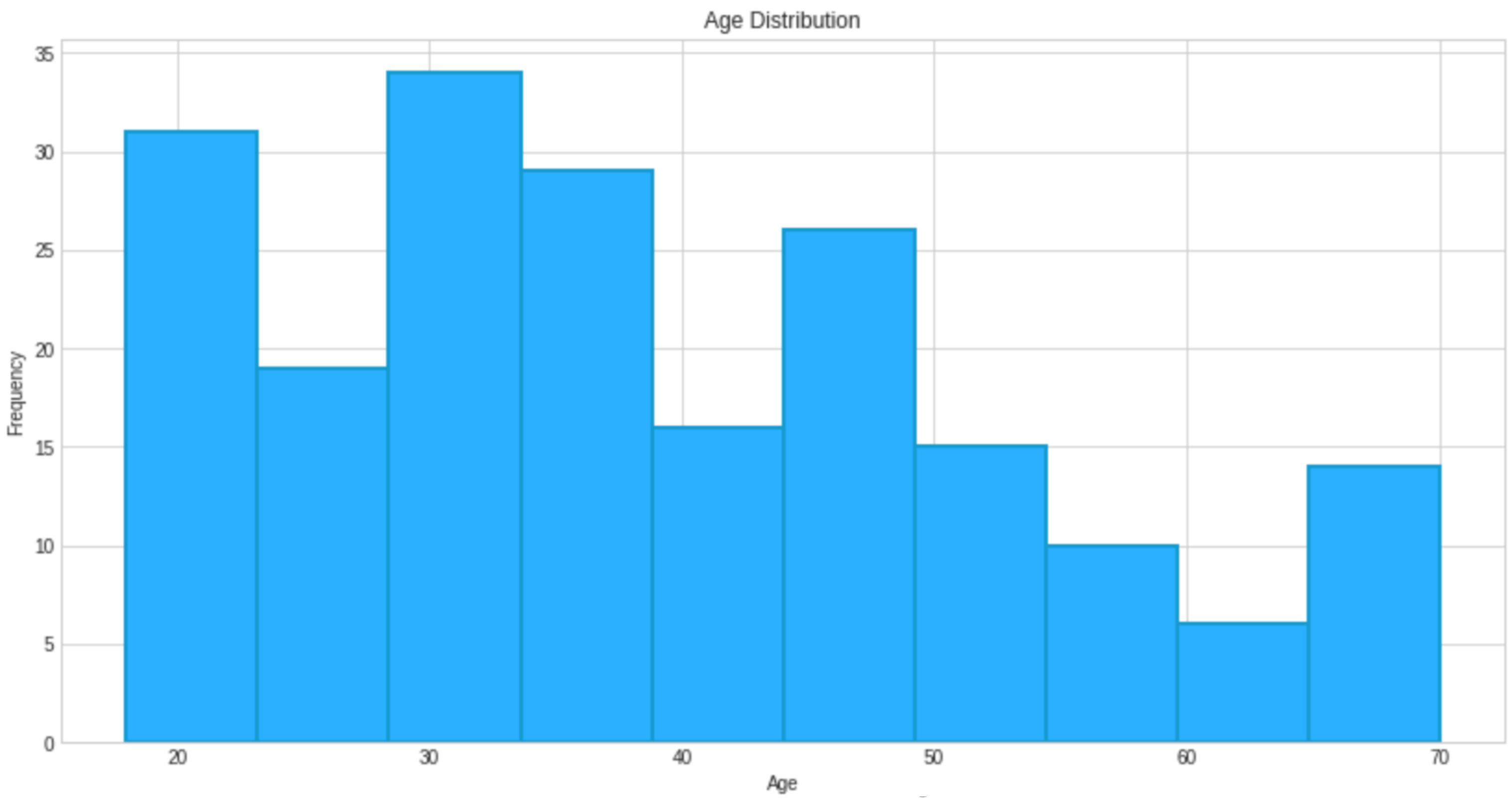
```
dataset.isnull().sum()
```

```
CustomerID                0
Genre                     0
Age                       0
Annual Income (k$)        0
Spending Score (1-100)    0
dtype: int64
```

```
import matplotlib.pyplot as plt
```

Data exploration 5 plt.hist

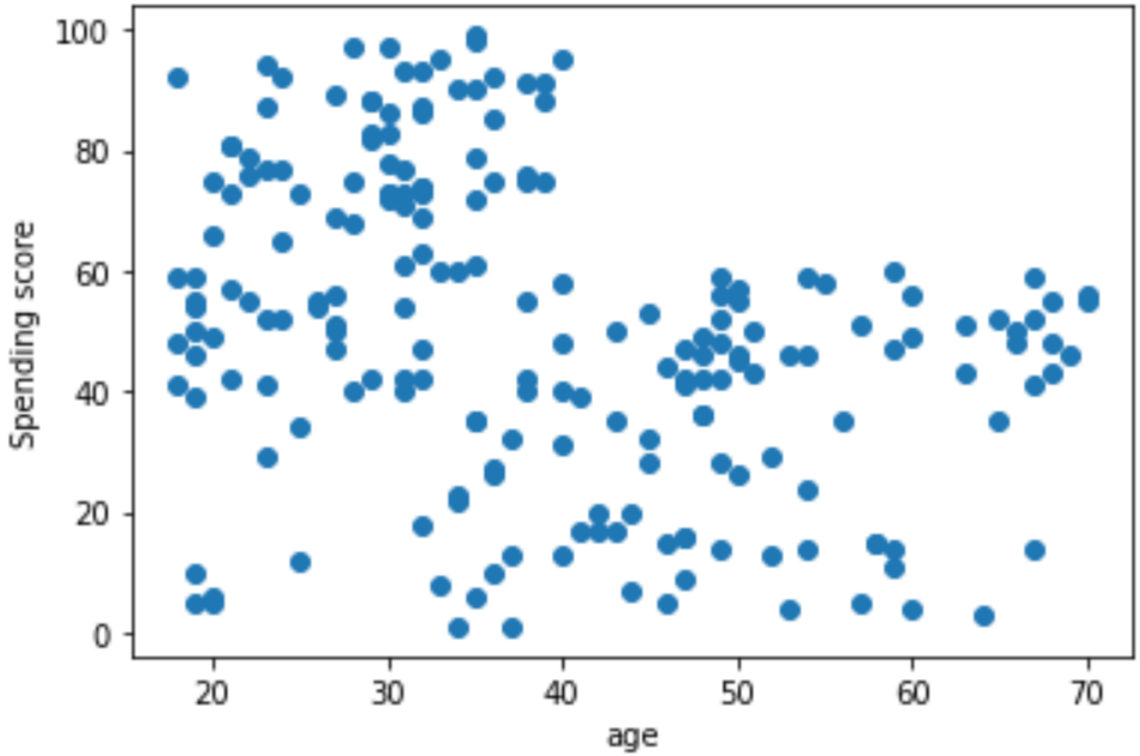
```
# age distribution
plt.style.use('seaborn-whitegrid')
plt.figure(figsize=(14,7))
plt.hist(dataset['Age'], bins=10, facecolor = '#2ab0ff', edgecolor='#169acf', linewidth=2)
plt.title('Age Distribution')
plt.xlabel('Age')
plt.ylabel('Frequency')
plt.show()
```



Plotting of the data

```
plt.xlabel('age')
plt.ylabel('Spending score')
plt.scatter(X[:, 0], X[:, 1])
```

<matplotlib.collections.PathCollection at 0x7f5bb4909710>



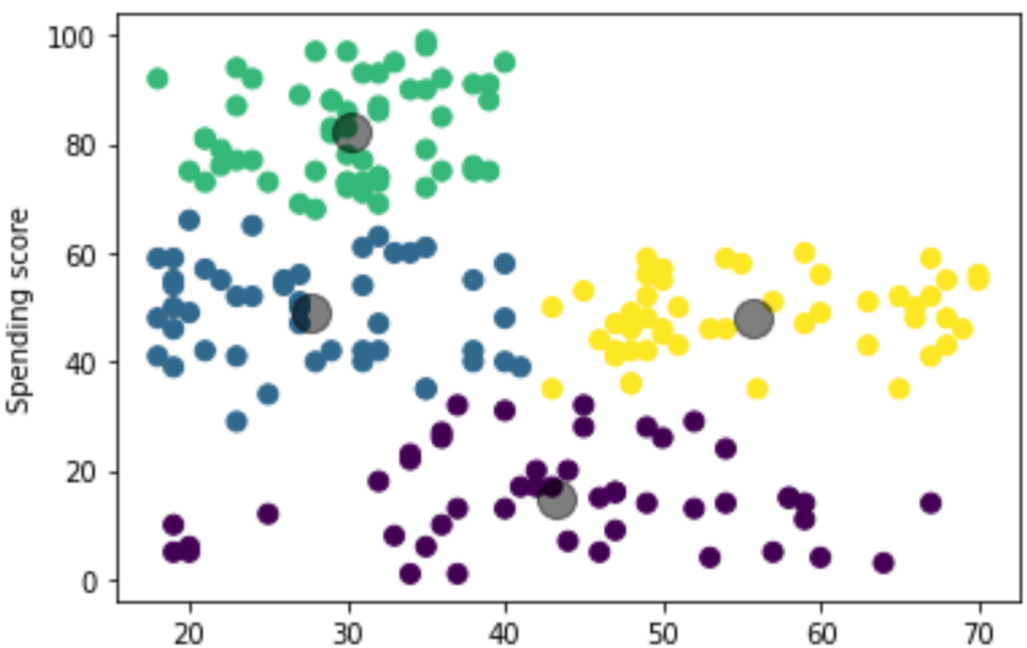
k-means

```
from sklearn.cluster import KMeans
kmeans = KMeans(n_clusters=4)
kmeans.fit(X)
y_kmeans = kmeans.predict(X)
centers = kmeans.cluster_centers_
```

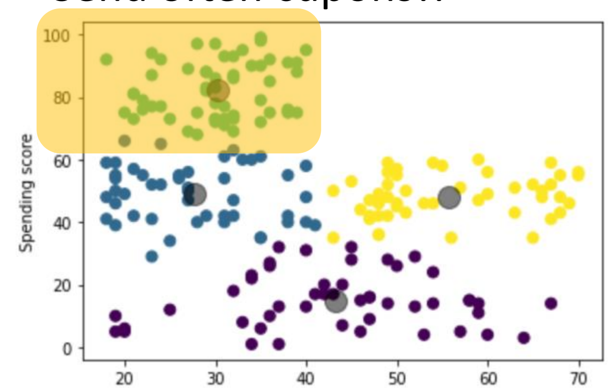
Marker size

```
plt.scatter(X[:, 0], X[:, 1], c=y_kmeans, s=50, cmap='viridis')
plt.scatter(centers[:, 0], centers[:, 1], c='black', s=200, alpha=0.5);
plt.xlabel('age')
plt.ylabel('Spending score')
```

```
Text(0, 0.5, 'Spending score')
```



Send often cupons!!

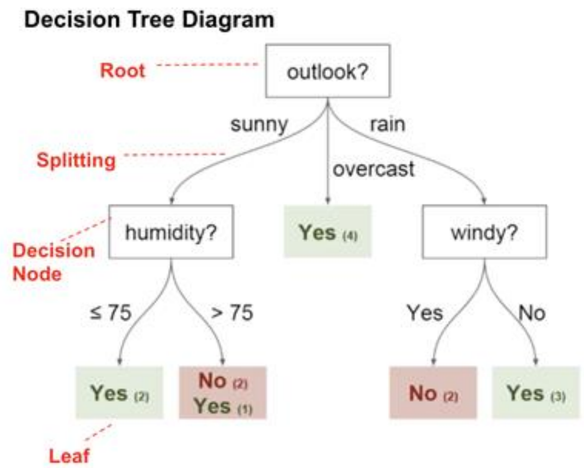


In >3D the clustering can help even more…

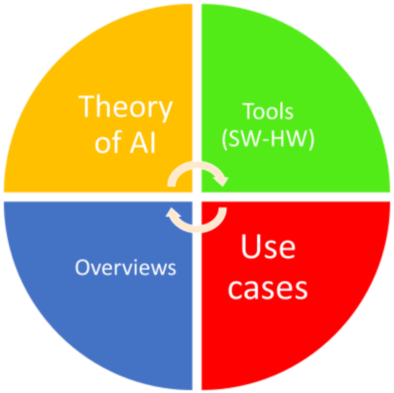



THEORY

Classical models: Decision tree



A very effective classical block



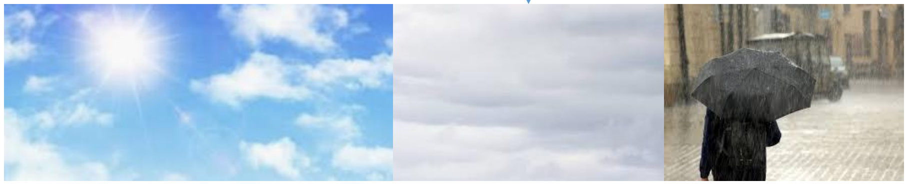
Decision tree: X,Y

Let’s go play tennis?

Dataset

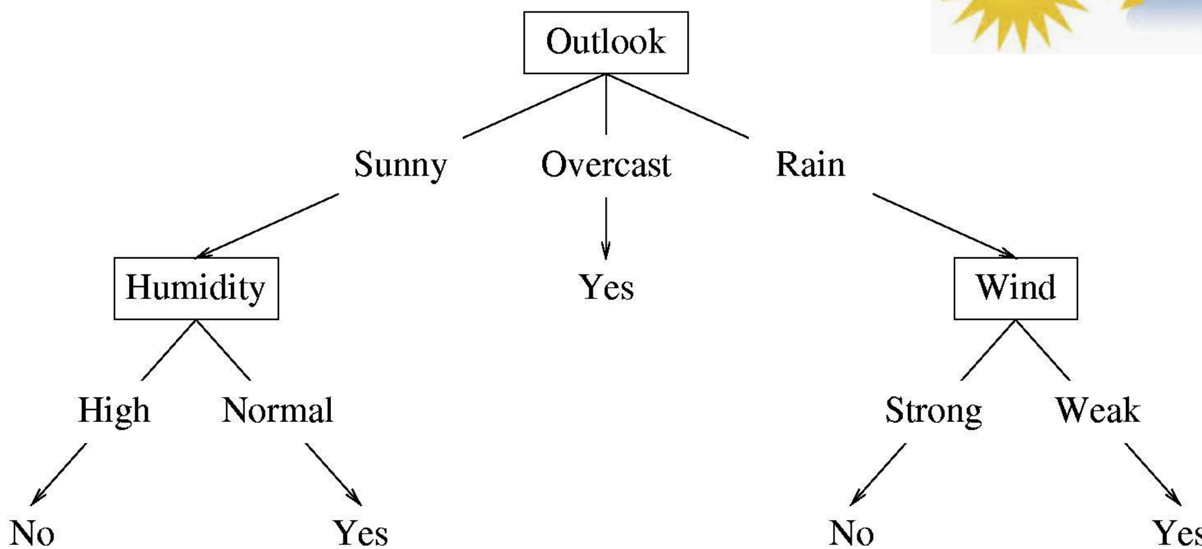
Temperature	Outlook	Humidity	Windy	Played?
Mild	Sunny	80	No	Yes
Hot	Sunny	75	Yes	No
Hot	Overcast	77	No	Yes
Cool	Rain	70	No	Yes
Cool	Overcast	72	Yes	Yes
Mild	Sunny	77	No	No
Cool	Sunny	70	No	Yes
Mild	Rain	69	No	Yes
Mild	Sunny	65	Yes	Yes
Mild	Overcast	77	Yes	Yes
Hot	Overcast	74	No	Yes
Mild	Rain	77	Yes	No
Cool	Rain	73	Yes	No
Mild	Rain	78	No	Yes

X Y



Decision Tree Hypothesis Space

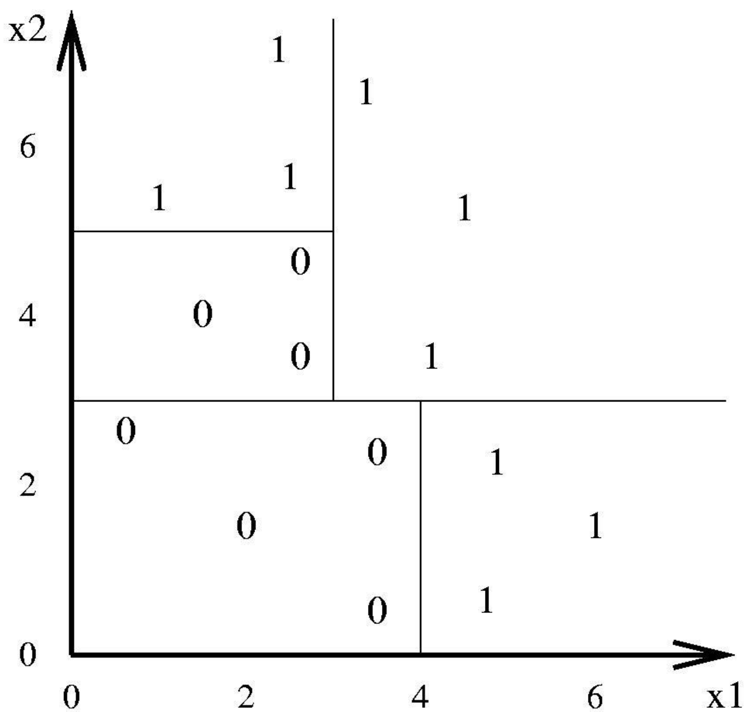
- **Internal nodes** test the value of particular features x_j and branch according to the results of the test.
- **Leaf nodes** specify the class $h(\mathbf{x})$.

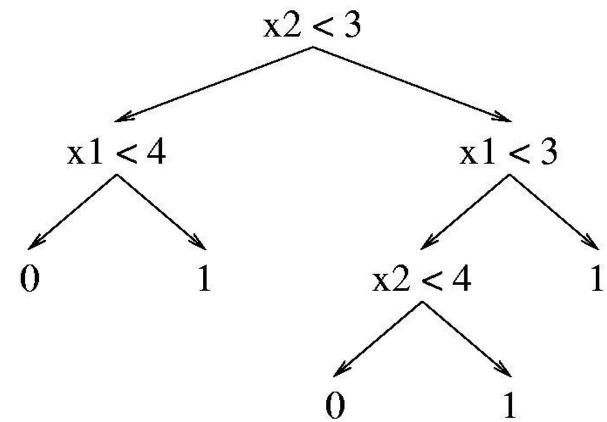


Suppose the features are **Outlook** (x_1), **Temperature** (x_2), **Humidity** (x_3), and **Wind** (x_4). Then the feature vector $\mathbf{x} = (Sunny, Hot, High, Strong)$ will be classified as **No**. The **Temperature** feature is irrelevant.

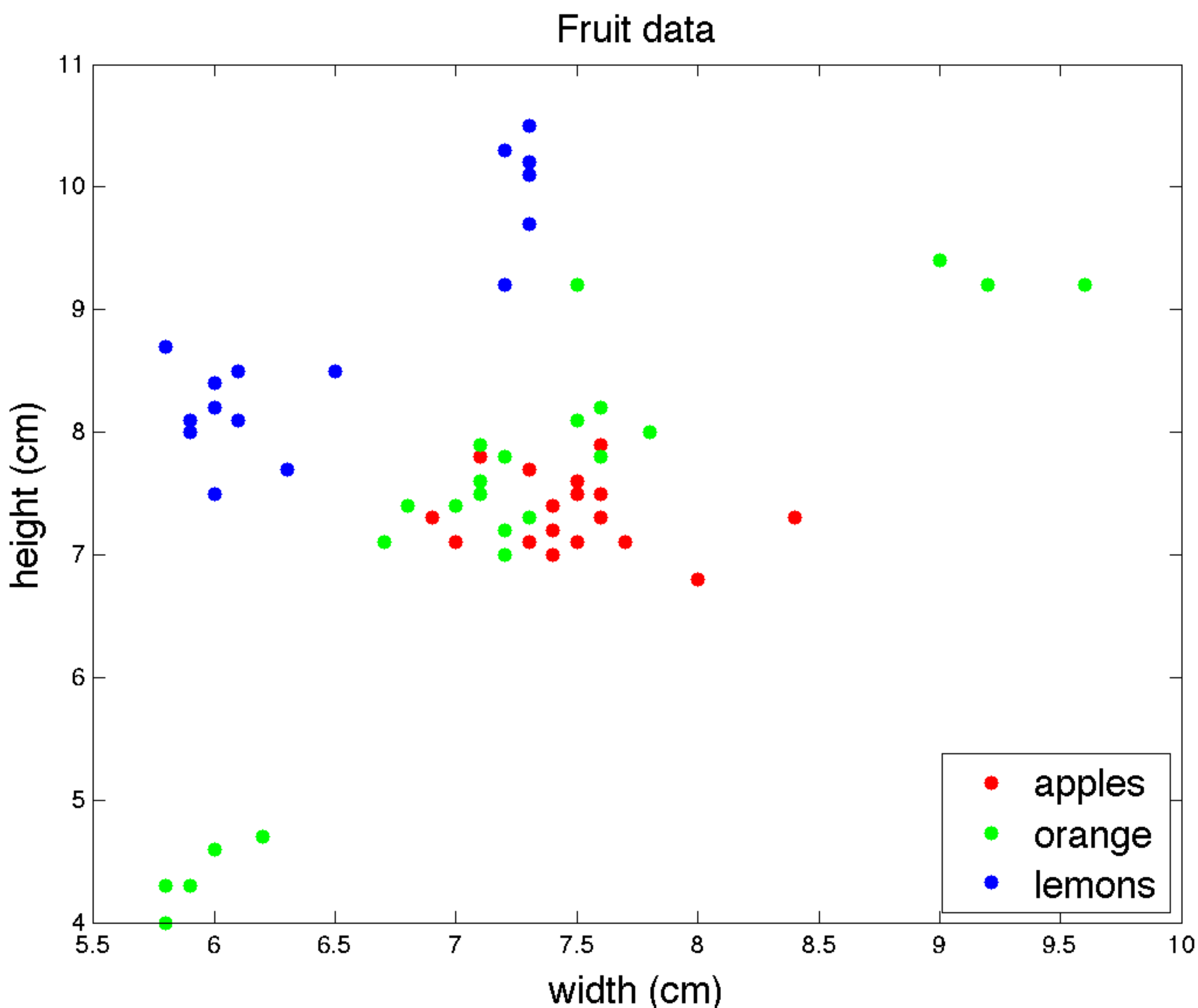
Decision Tree Decision Boundaries

Decision trees divide the feature space into axis-parallel rectangles, and label each rectangle with one of the K classes.

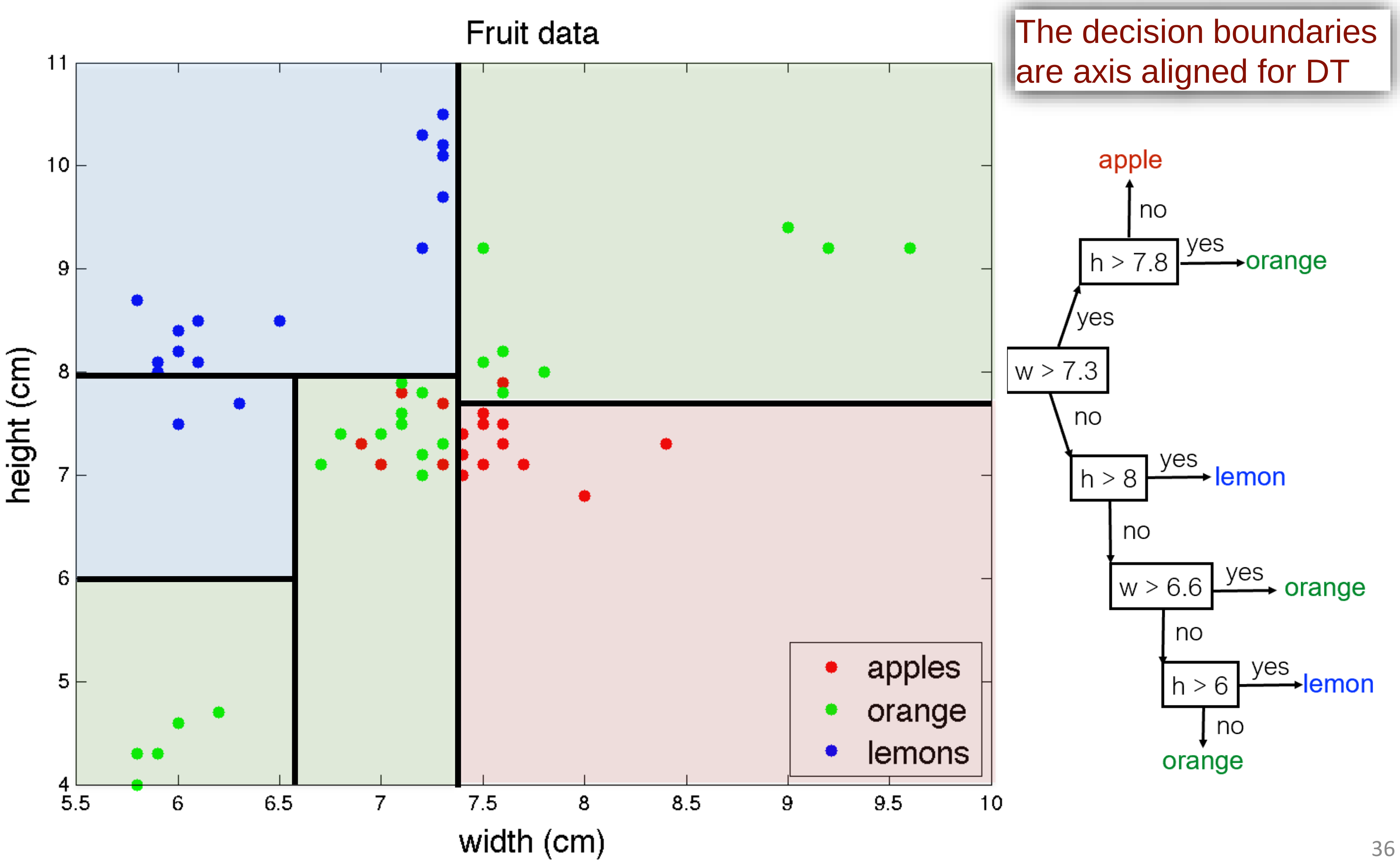




Review the fruit data case for Decision Trees

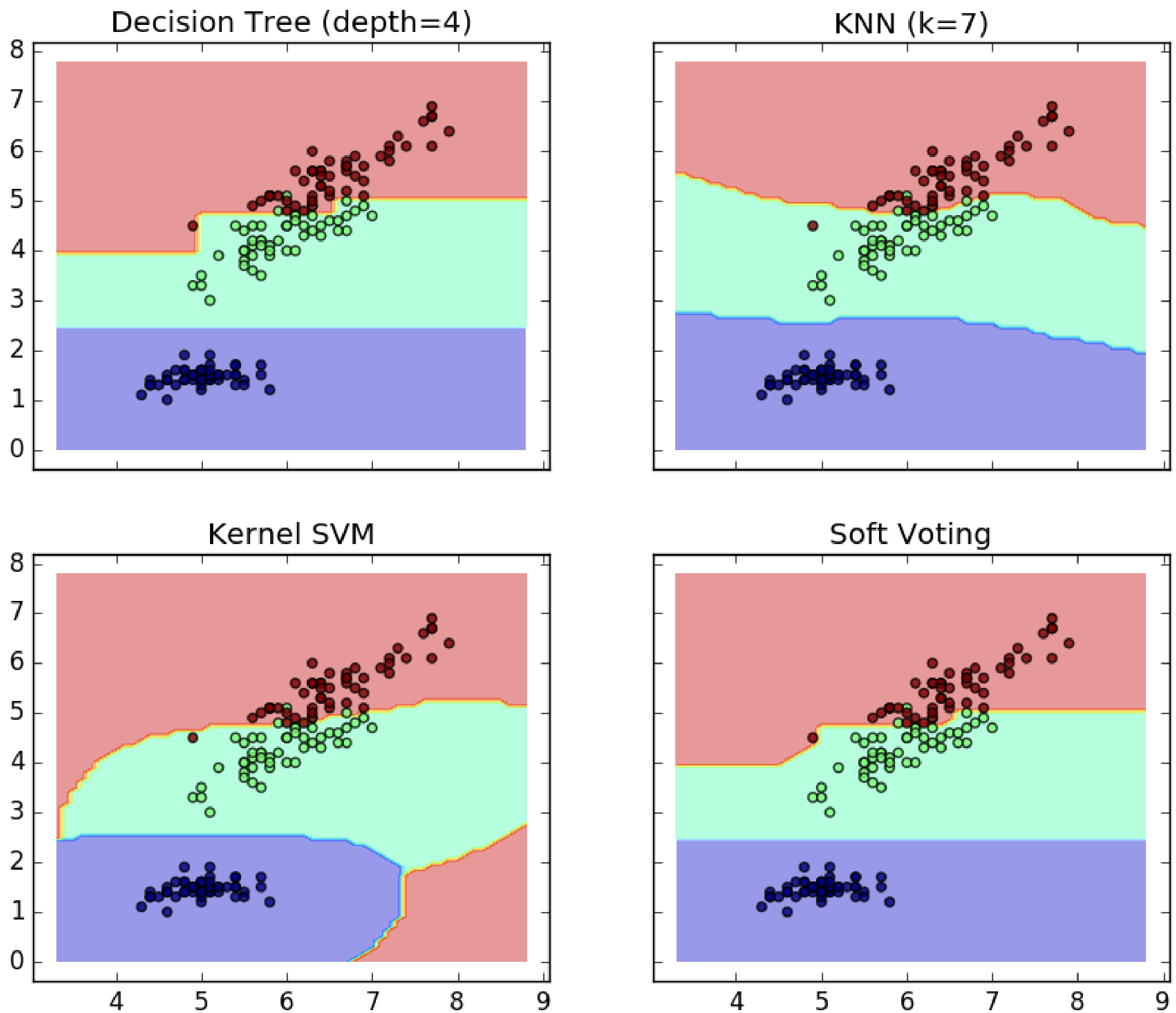


Decision boundaries: **Decision Tree**



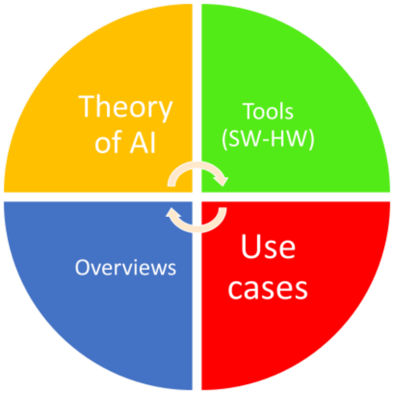
The decision boundaries are axis aligned for DT

Comparing the decision boundaries of different models

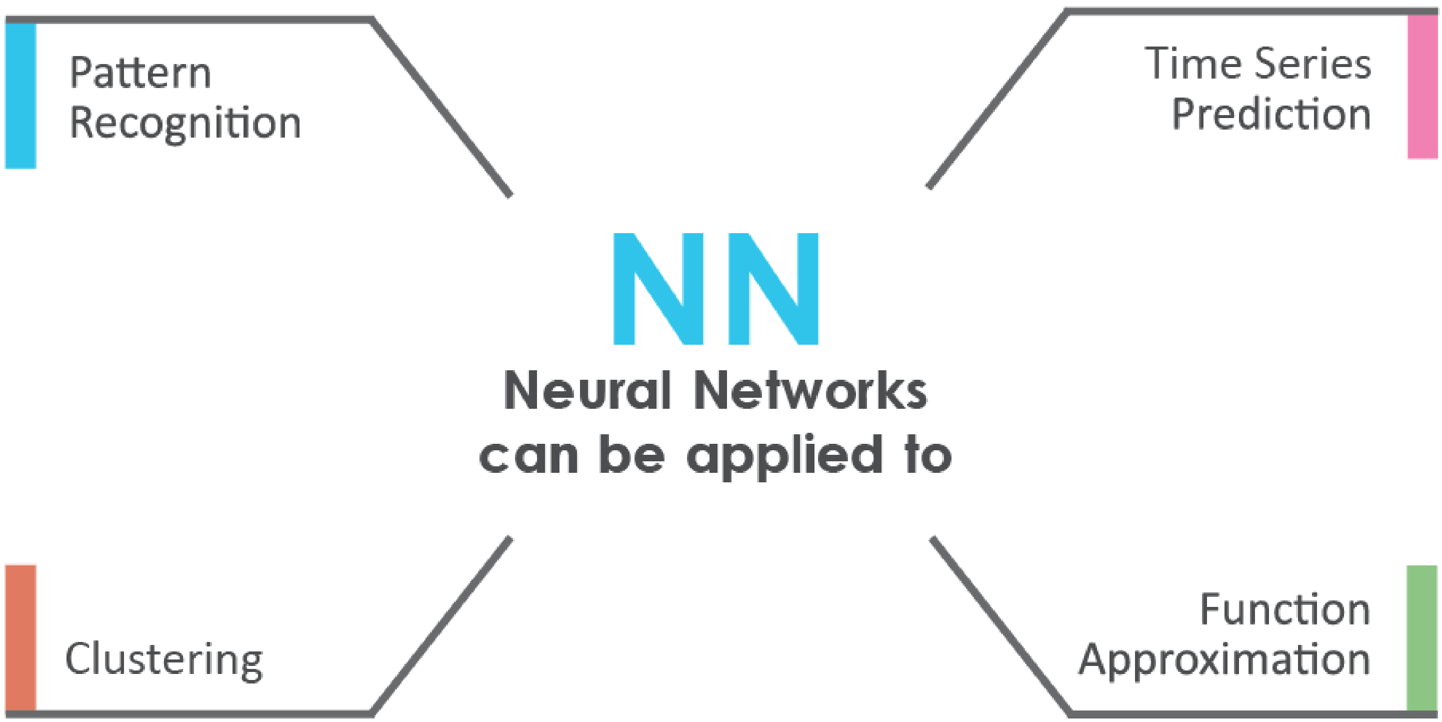





THEORY
Artificial neural networks



Typical Applications



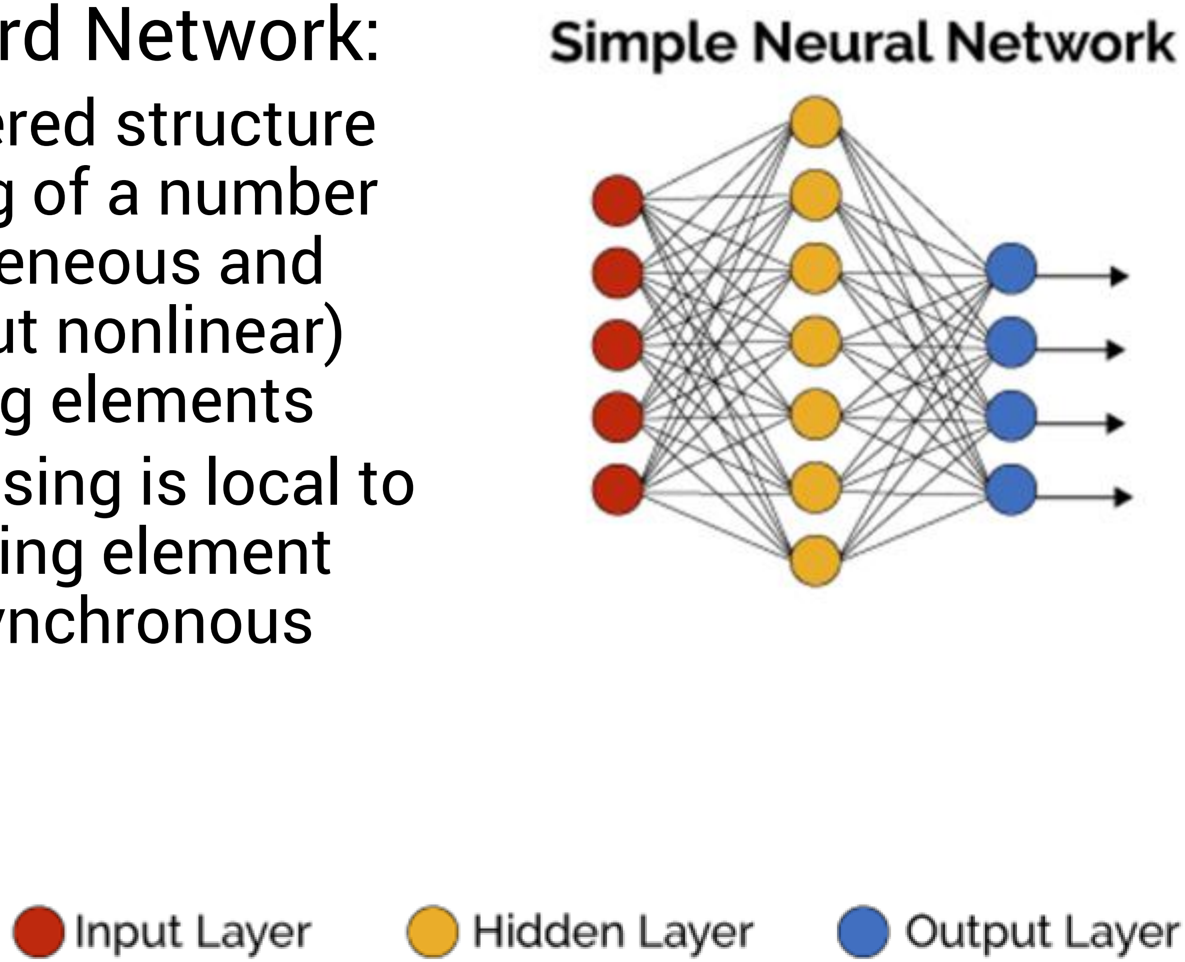
Neural Networks

- Original inspiration was the human brain; emphasis now on usefulness as a computational tool

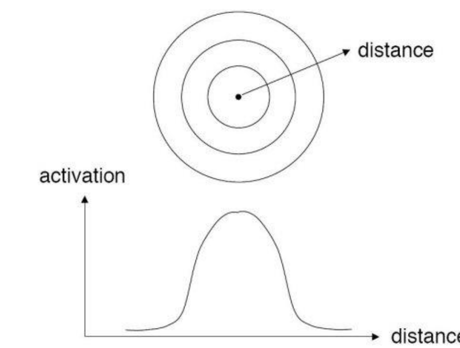
- Many useful NN paradigms, but scope of today's discussion limited to the feed-forward network, the most popular paradigm

Neural Networks (2)

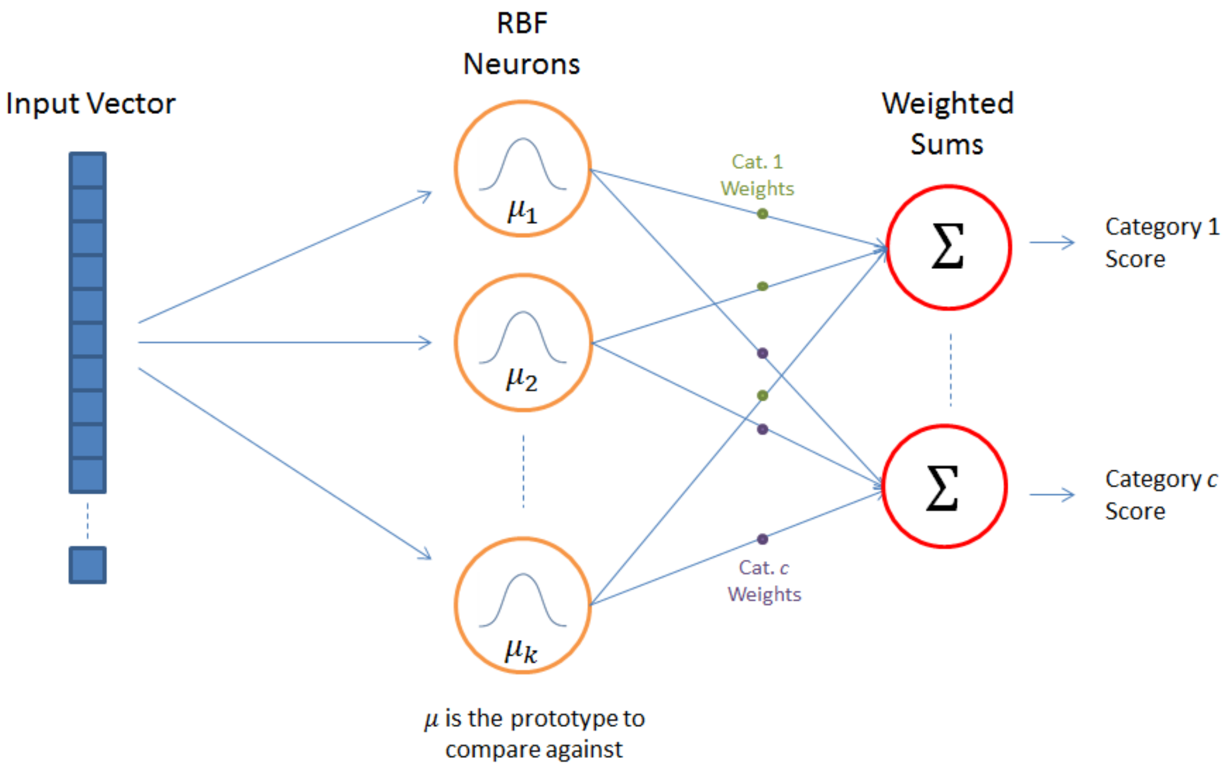
- Feed-forward Network:
  - It is a layered structure consisting of a number of homogeneous and simple (but nonlinear) processing elements
  - All processing is local to a processing element and is asynchronous

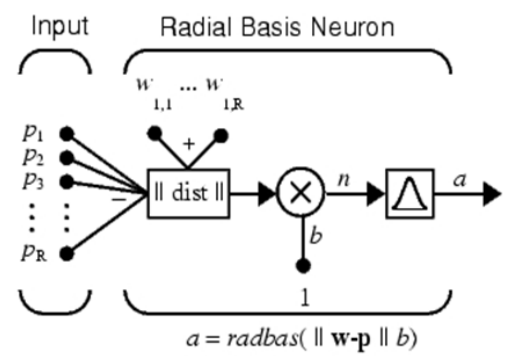


Neural Networks: Radial Basis Function



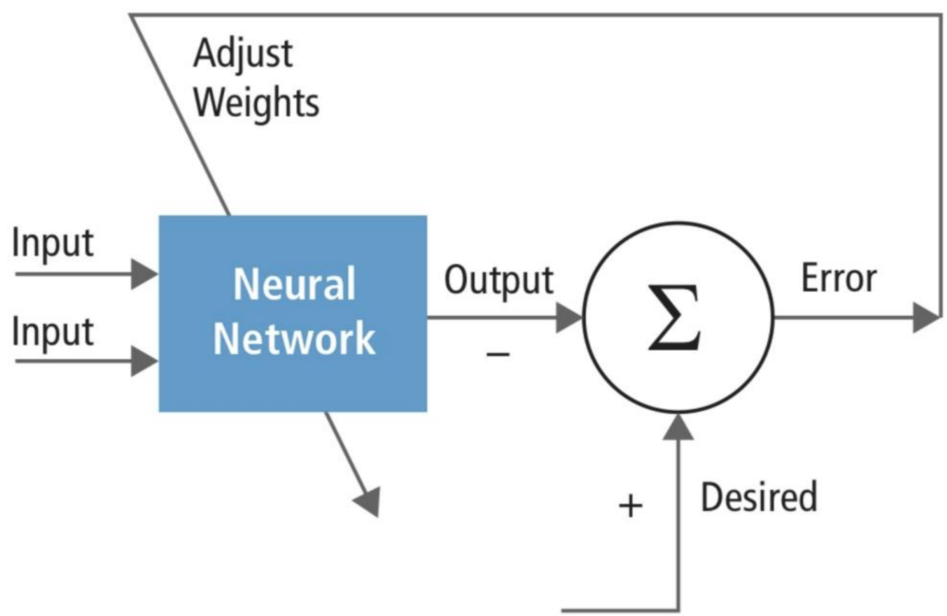
$$g_i\left(x_j\right) = \exp\left(\frac{-\left\|x_j - \mu_i\right\|^2}{2\sigma_i^2}\right)$$



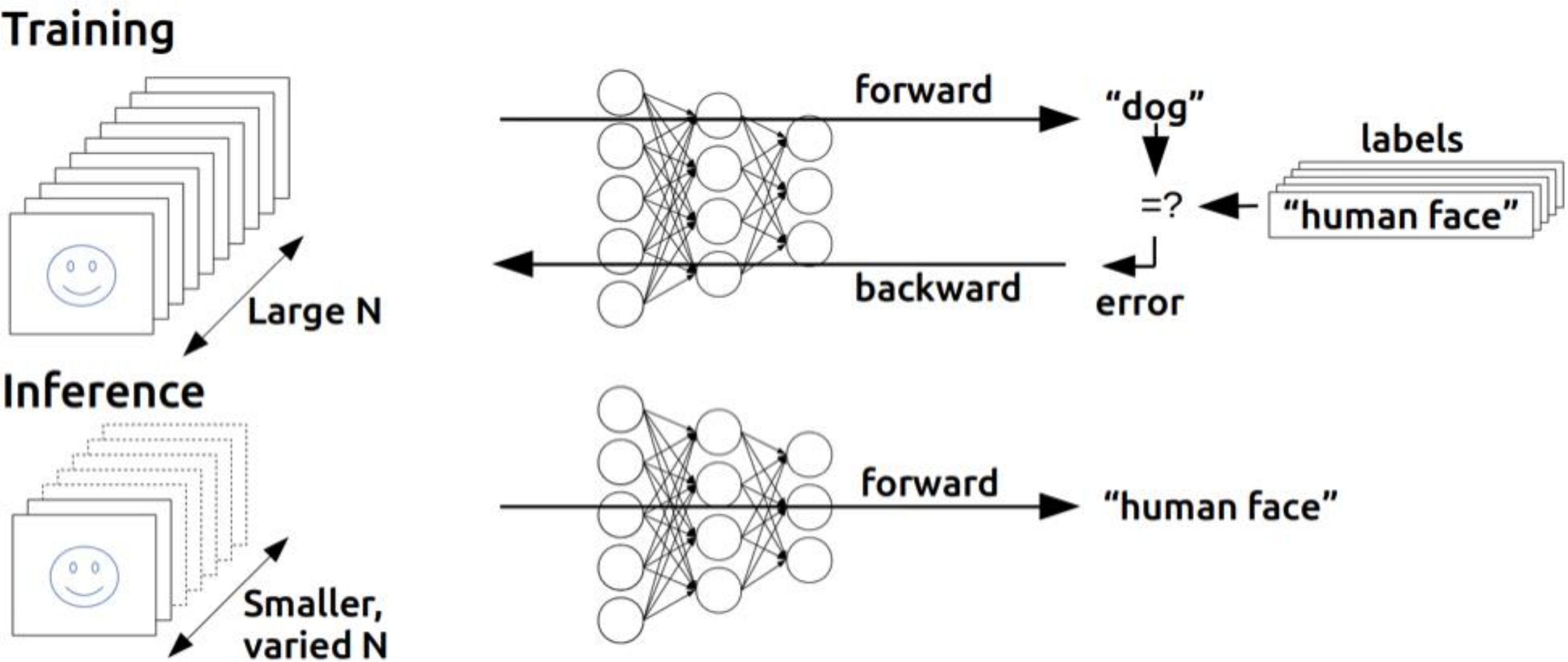


Neural Networks training

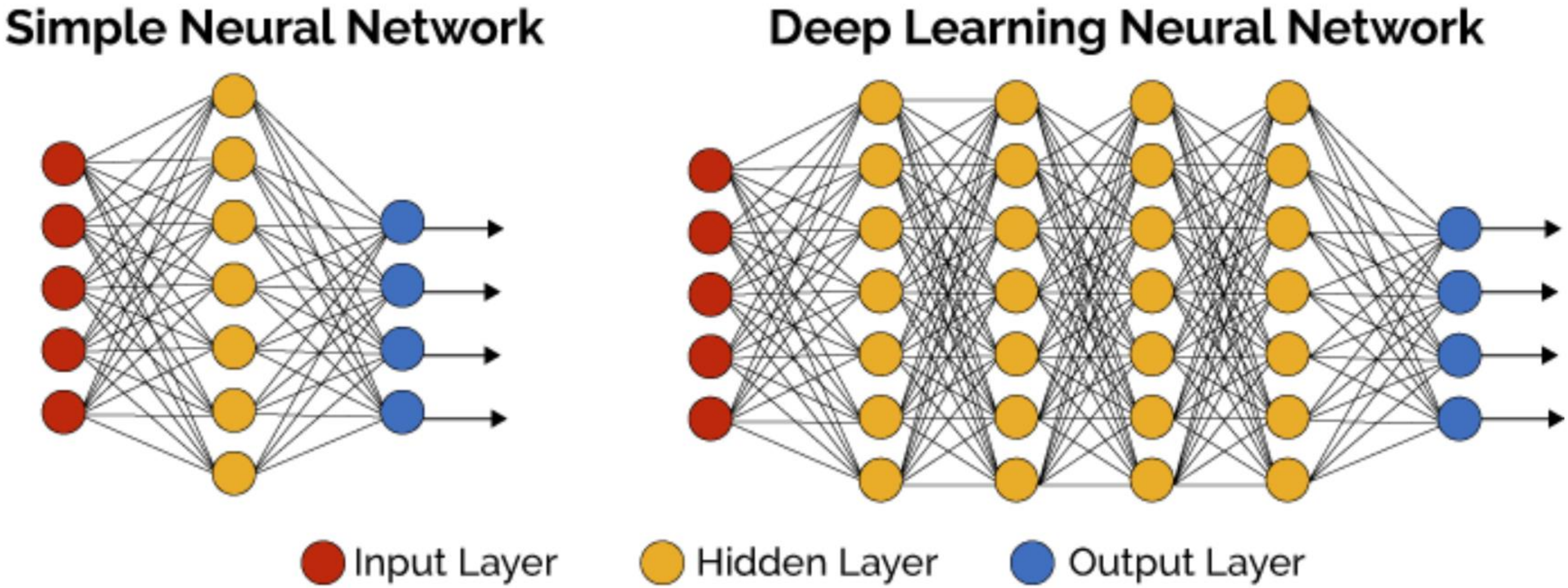
- During training the FN is forced to adjust its parameters so that its response to input data becomes closer to the desired response



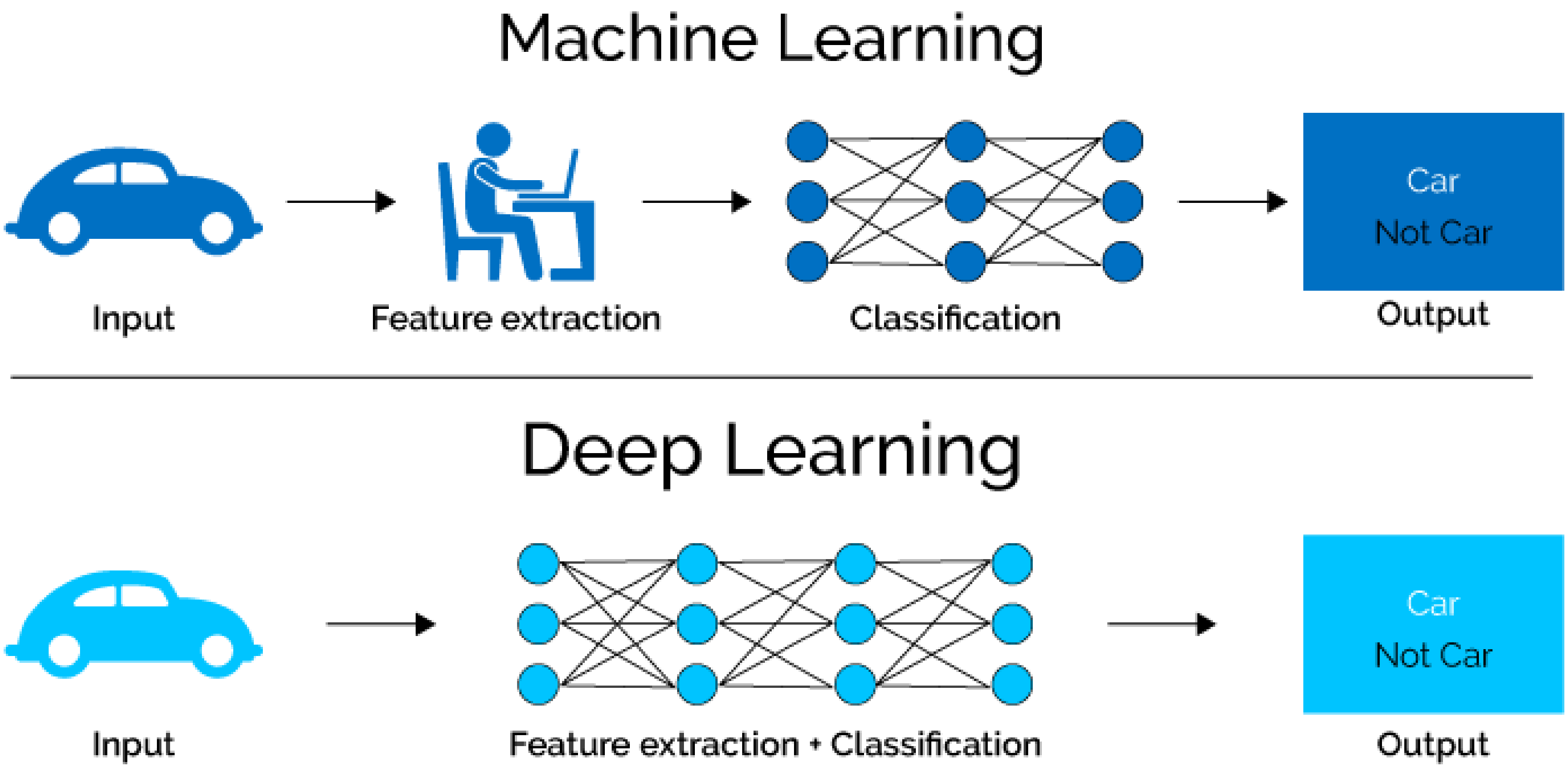
Neural Networks training (2)



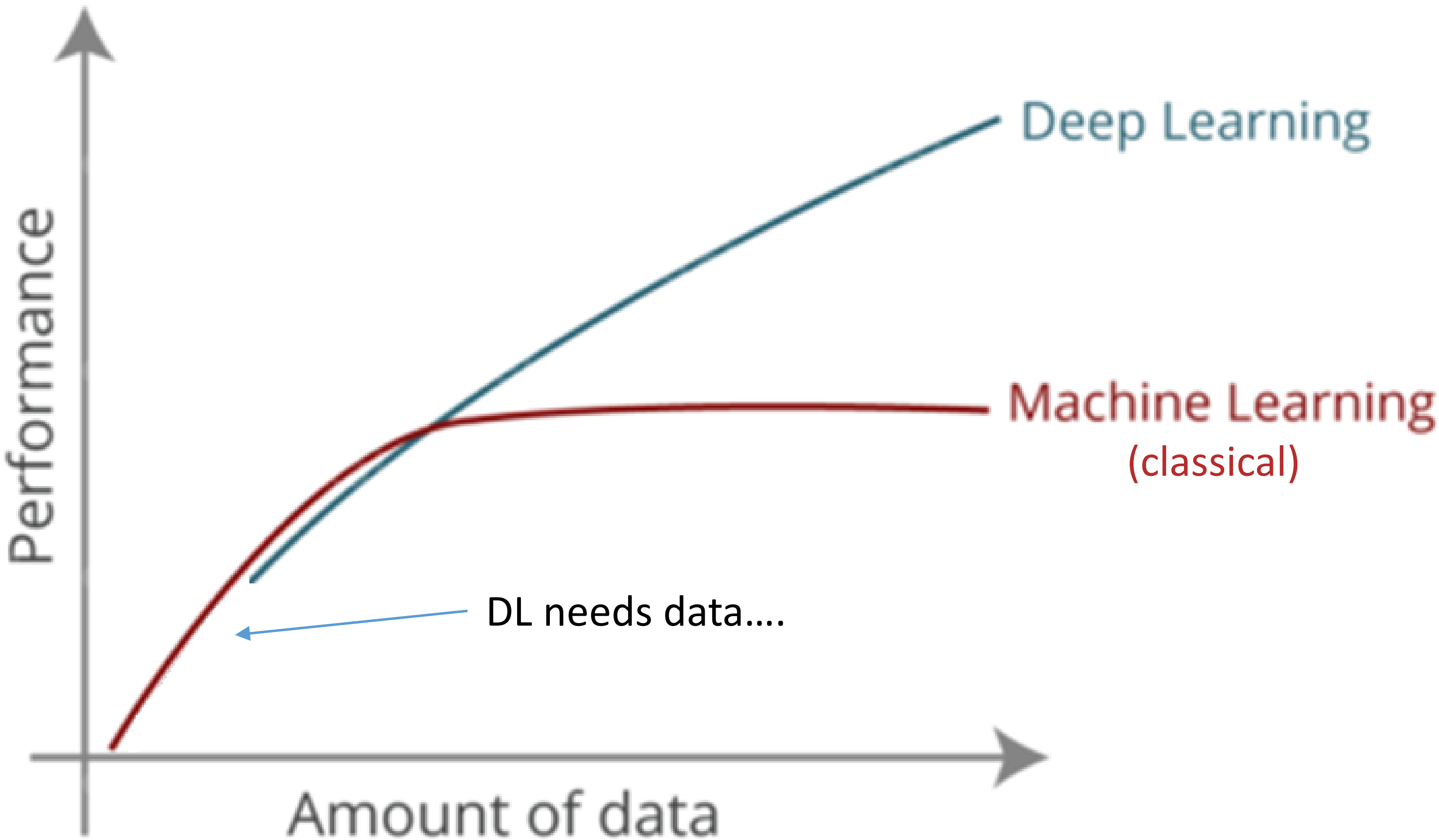
Deep Learning networks



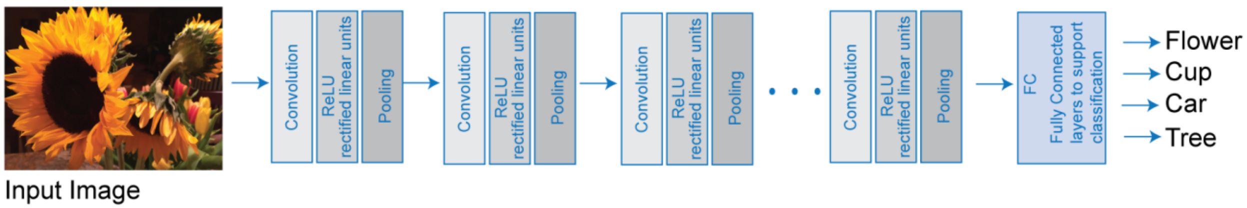
ML vs Deep Learning

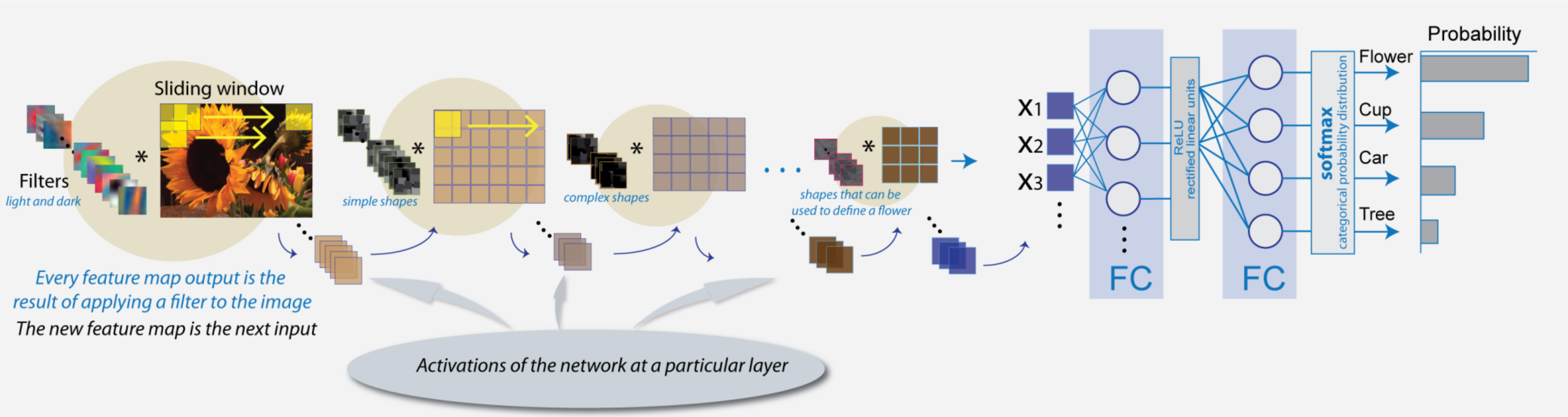


Classical ML vs DeepLearning

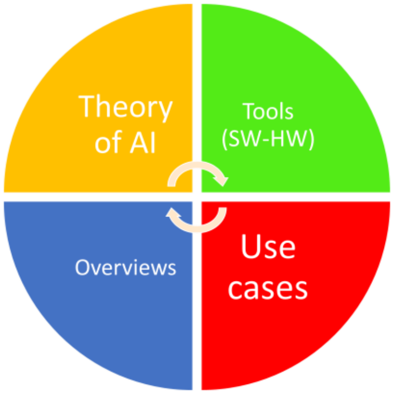


Convolutional Neural Nets.





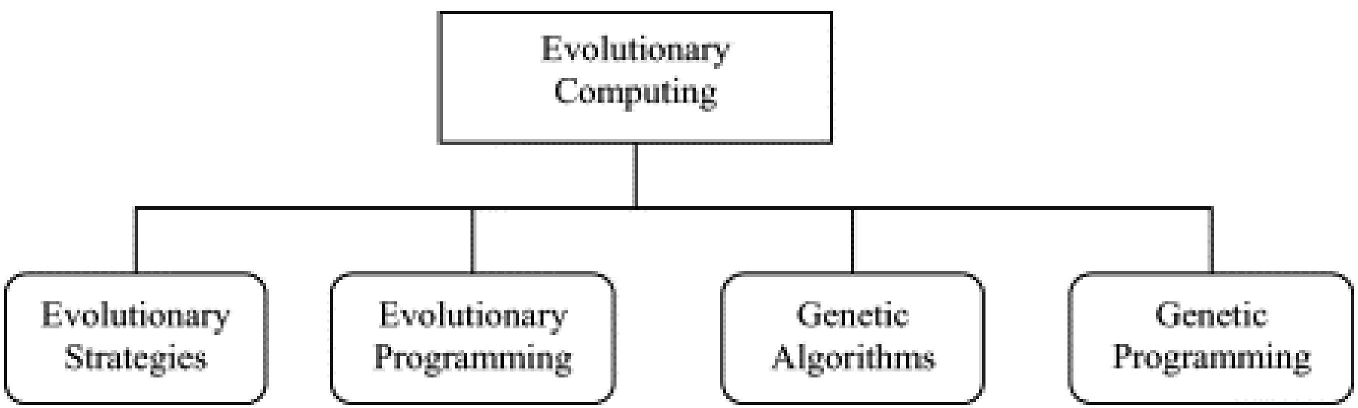
THEORY
Evolutionary computing



Genetic Algorithms (1)

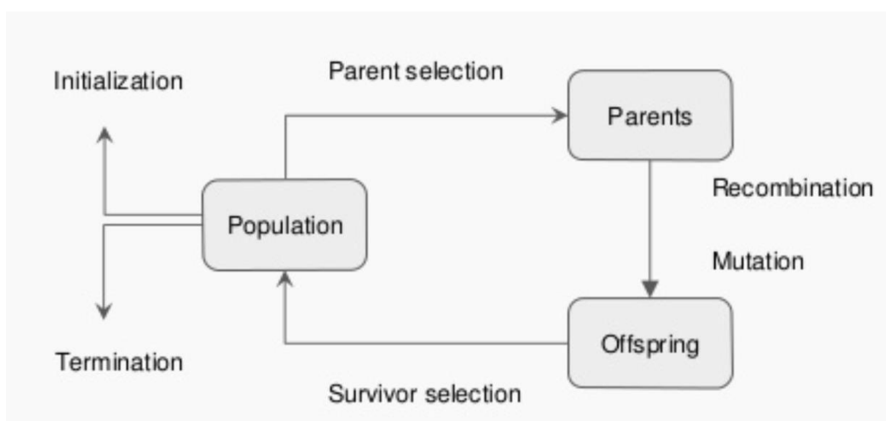
- Based on Darwin's evolutionary principle of 'survival of the fittest'

- GAs require the ability to recognize a good solution, but not how to get to that solution

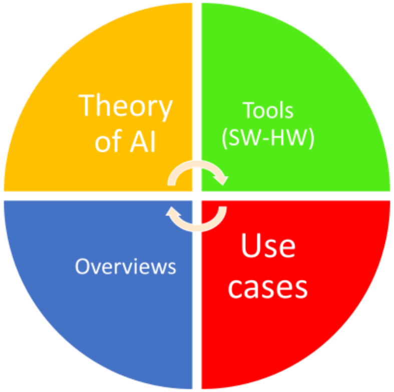


Genetic Algorithms (2)

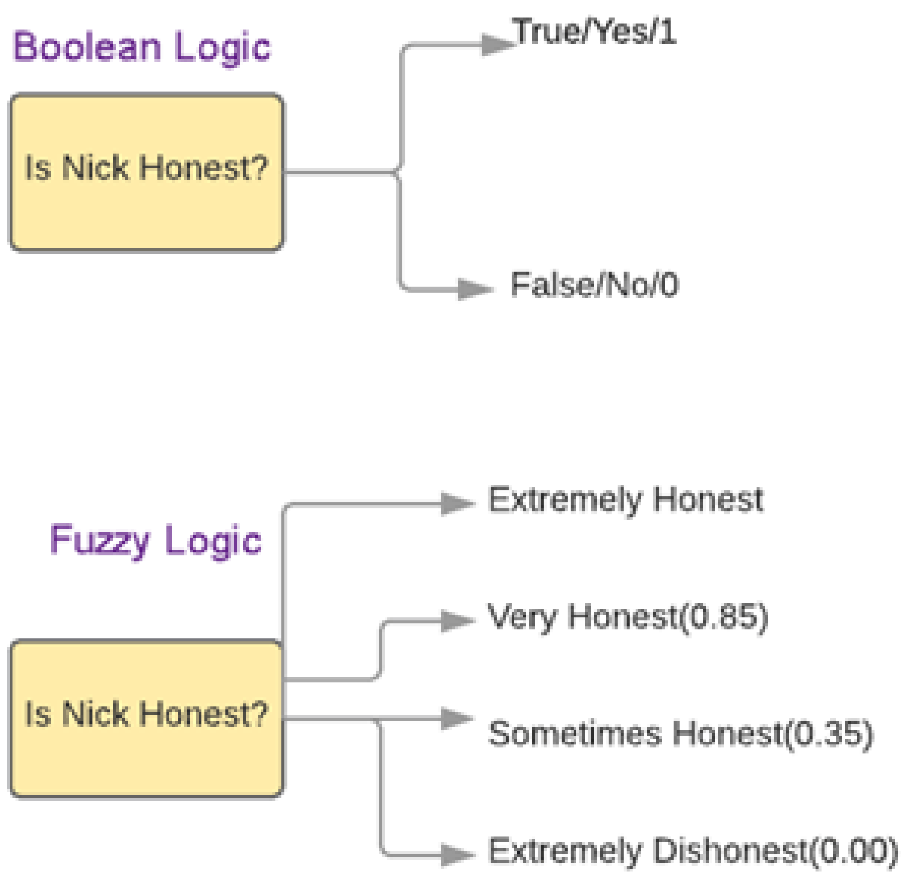
- The procedure:
  - An initial set of random solutions is ranked in terms of ability to solve the problem at hand
  - The best solutions are then crossbred and mutated to form a new set
  - The ranking and formation of new solutions is continued until a good enough solution is found or …



THEORY
Fuzzy systems



Fuzzy vs Boolen Logic



Fuzzy Logic

- Based on the principles of the approximate reasoning faculty that humans use when faced with linguistic ambiguity

- The inputs and outputs of a fuzzy system are precise, only the reasoning is approximate

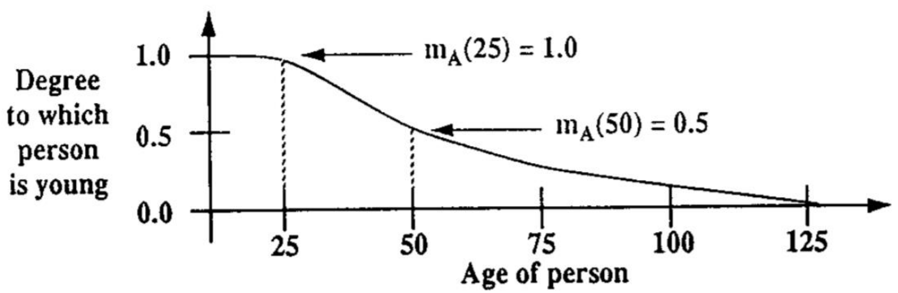
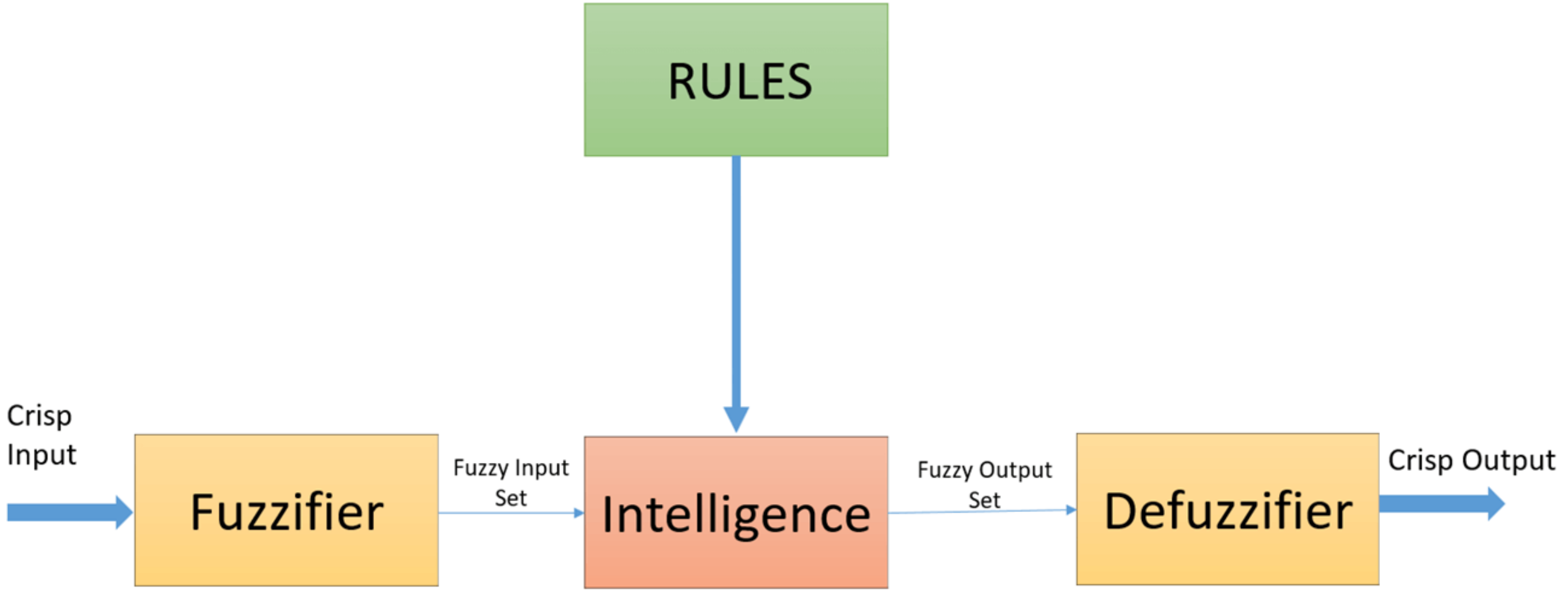


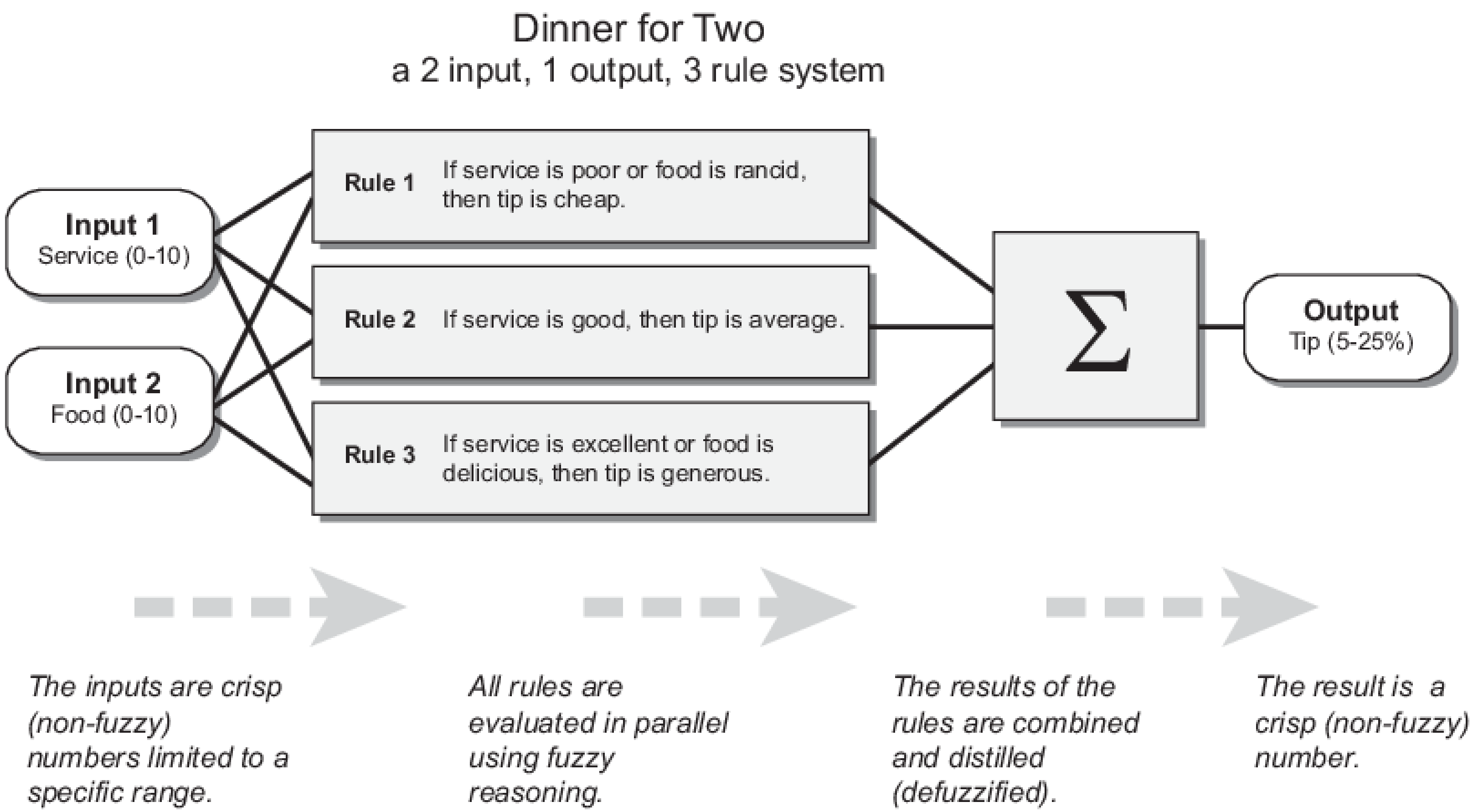
Fig. 1. This membership function describes the relationship between a person's age and the degree to which a person is considered to be young. This membership function determines that a 25-year-old person belongs to A twice as much as a 50-year-old person.

Fuzzy Logic (cont.)

- Parts of the knowledgebase of a fuzzy system:
  - Fuzzy rules
  - Fuzzy sets

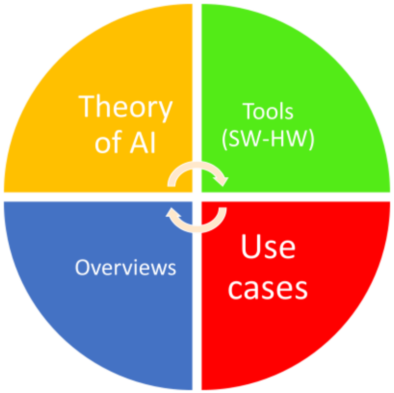


Fuzzy logic example



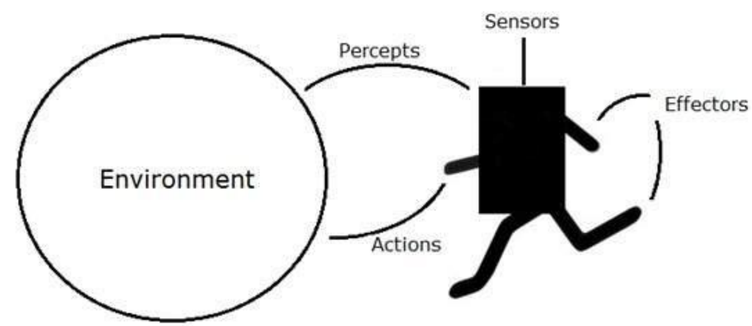



THEORY
Agents



Software Agents

- Core paradigm in AI

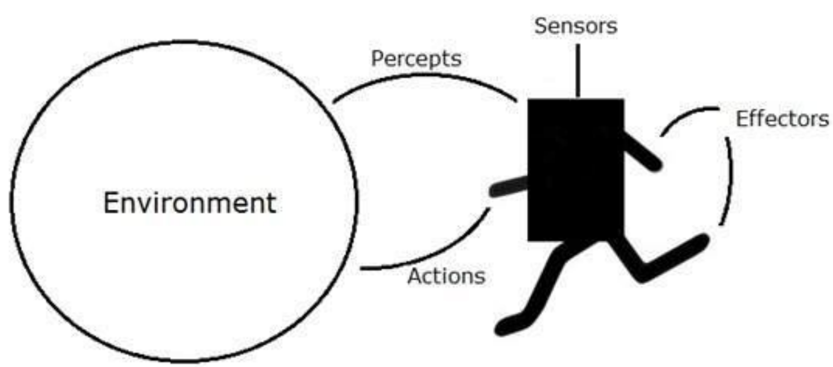


Definition:

A software agent is a long-term operating program whose function can be described as autonomous execution of tasks or tracing of goals via interaction with his environment.

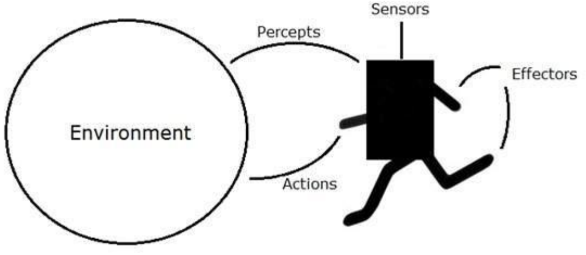
A software agent has…

- A **memory** and the capability to act in his world based on it.
- **Sensors** to perceive information from his environment.
- **Actuators** to influence the external world.
- The capability to **probe** actions



Applications of software agents

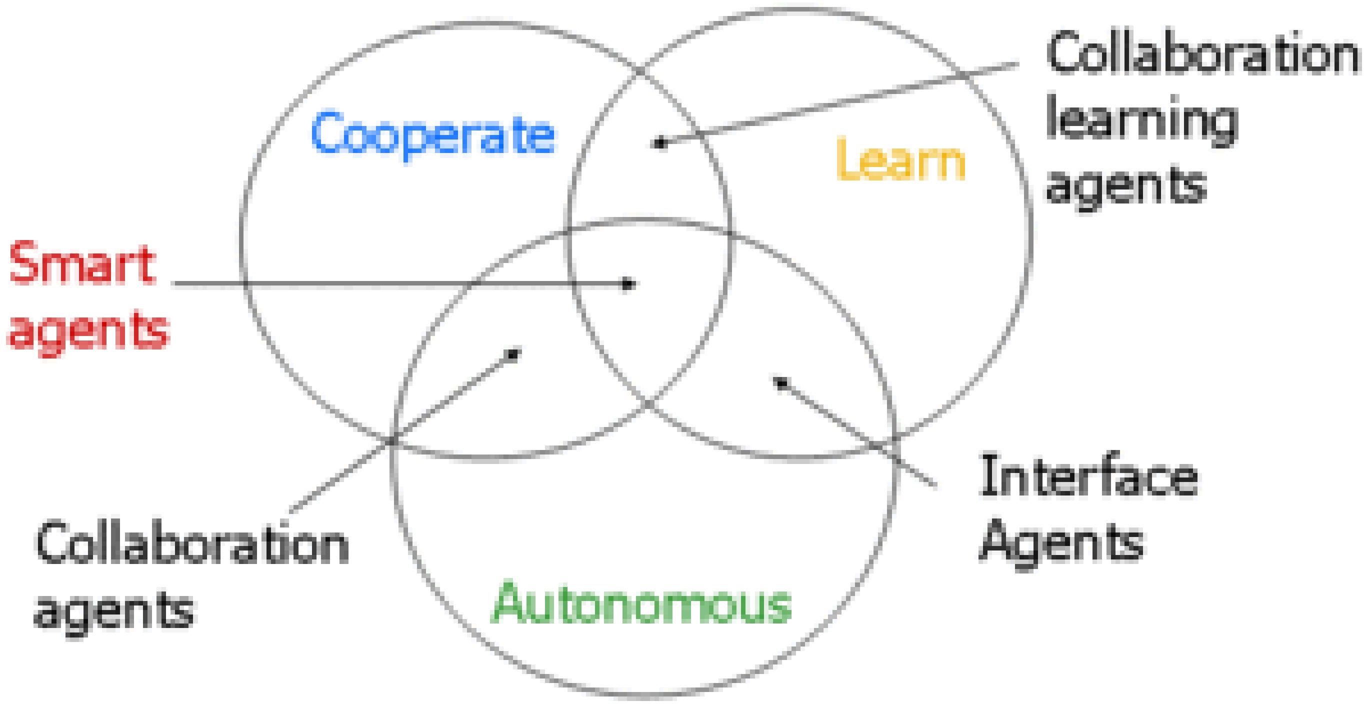
- Data collection and filtering,
- Event notification,
- Planning
- Optimization
- In various application areas
    - commerce
    - production
    - military
    - Education
    - …



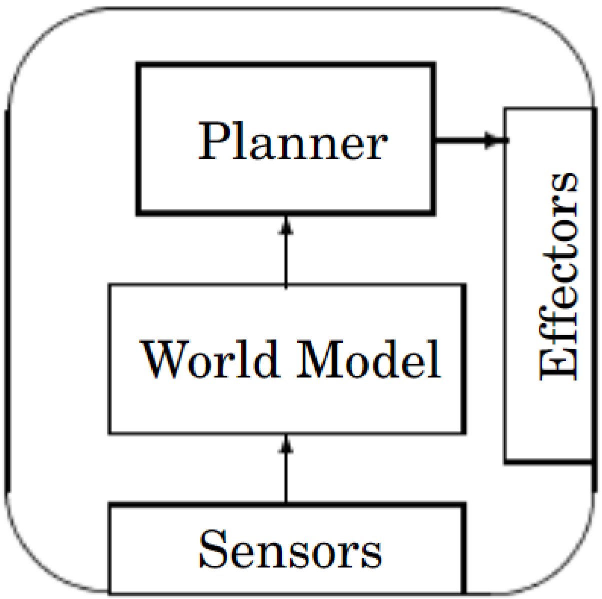
Agents: world model and applications

- Softbots
- Virus
- Database search
- Maintenance activities
- Toxic environments
- Disaster scenarios
- Exploitation
- Servicing, management,
- Assembly tasks
- Warfare

Agent: primary attributes



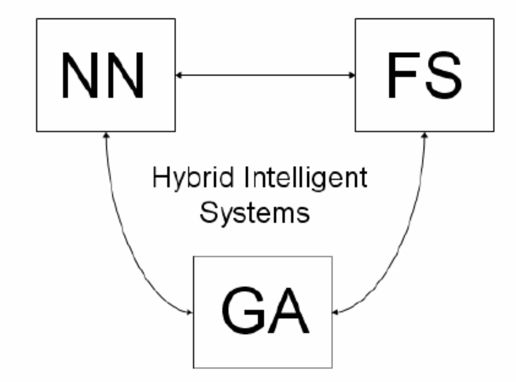
Agents: structure

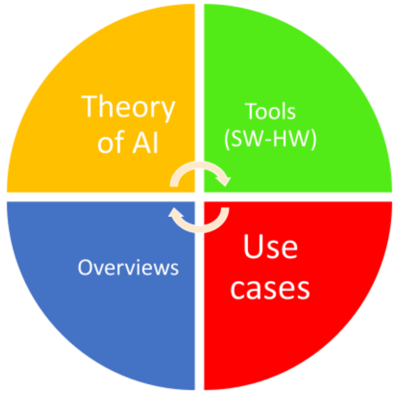


The **world model** is
an internal description
of the agent's external environment

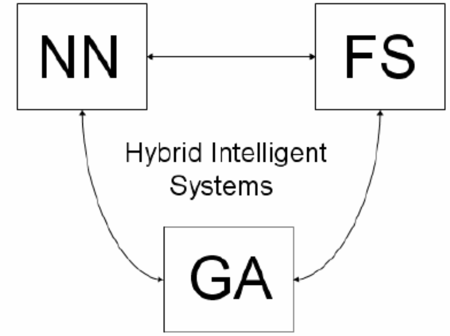
The **planner** uses
the world description
to make a plan of how to
accomplish the agent's goal

THEORY
Hybrid Intelligent Systems





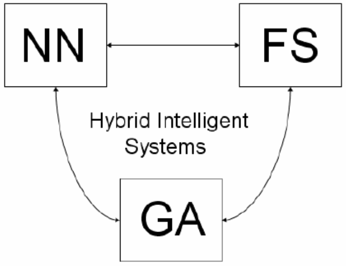
“A software system which employs,
in parallel, a combination
of methods and techniques
from artificial intelligence”



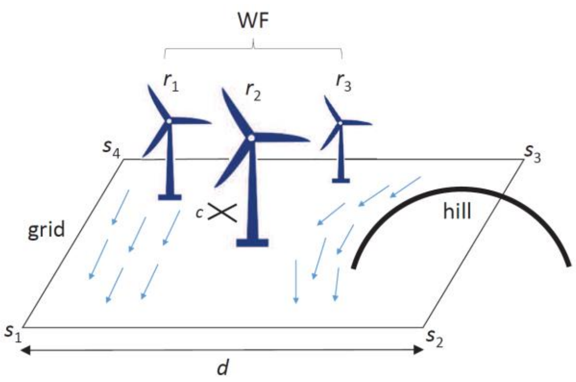
Why hybrid?

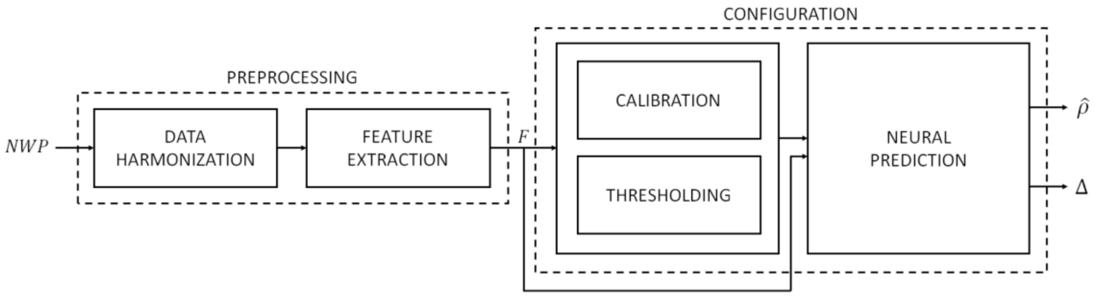
- Neural computing, machine learning, fuzzy logic, evolutionary algorithms, swarm intelligence, agent-based methods, among others (“soft computing”), have been established and shown their strength and drawbacks.
- Find the best combination of techniques to **mitigate the drawbacks** in **your** application
- Fusions:
  - soft computing/AI and hard computing
  - Different soft computing/AI
  - Embedding (e.g. neurofuzzy)

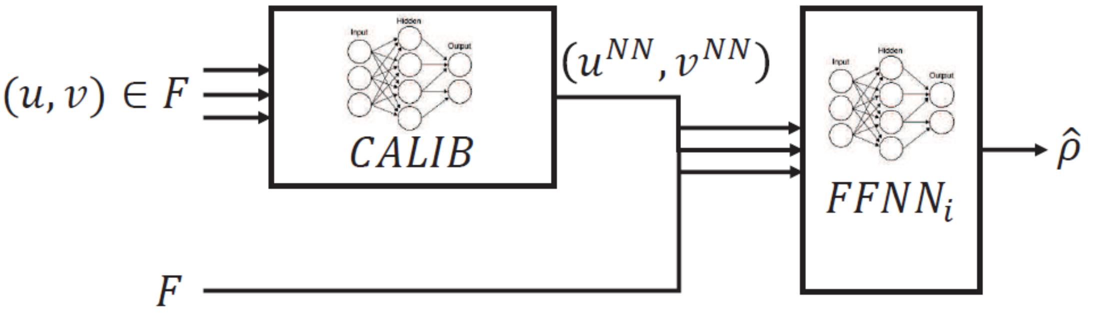
Examples of simple Hybrid Intelligent System



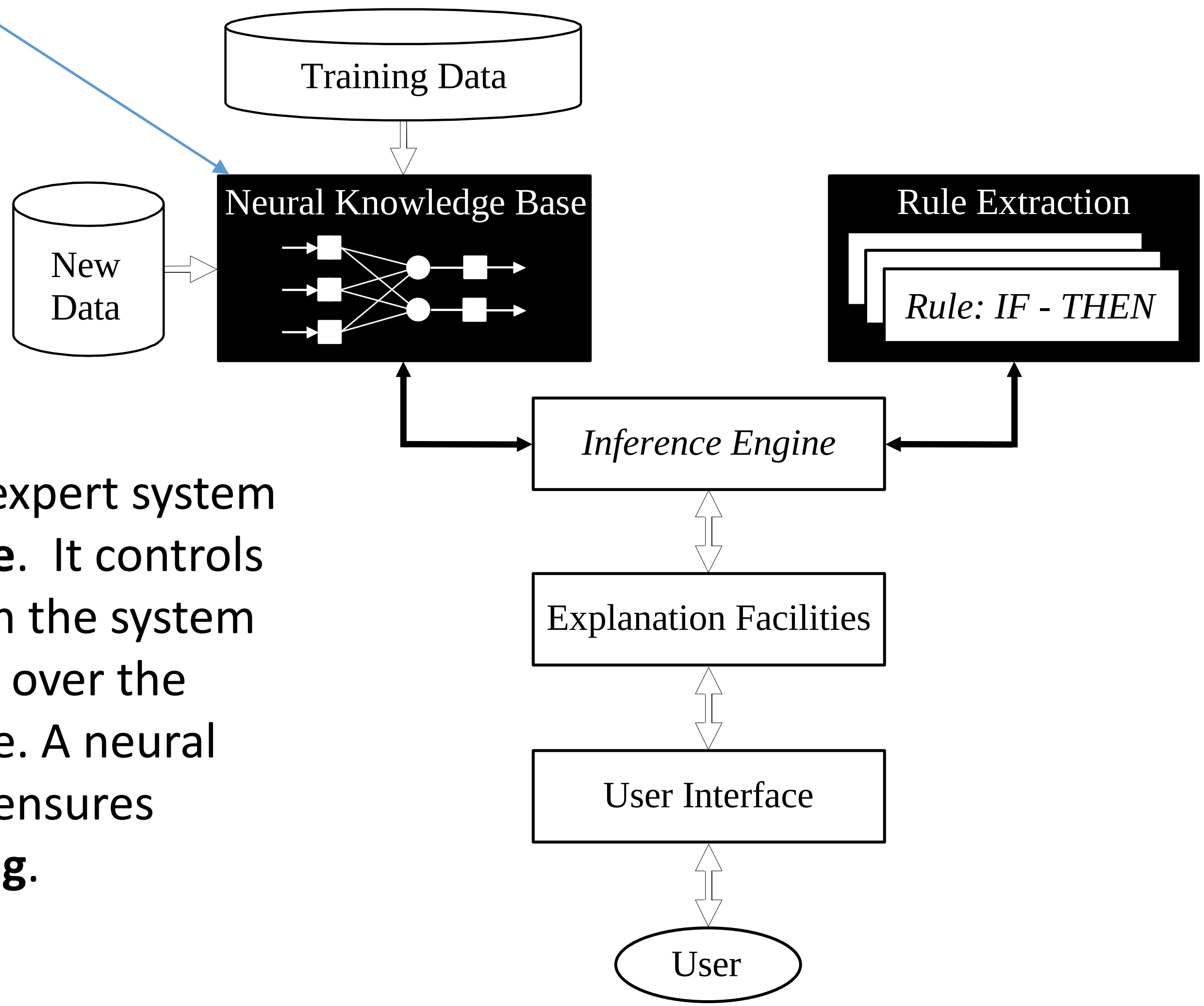
- They are just “in parallel” (non embedded)





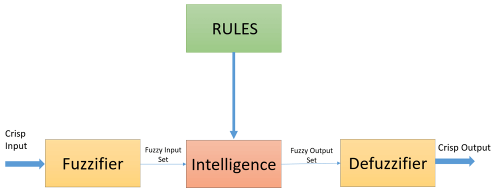


Neural Expert Systems

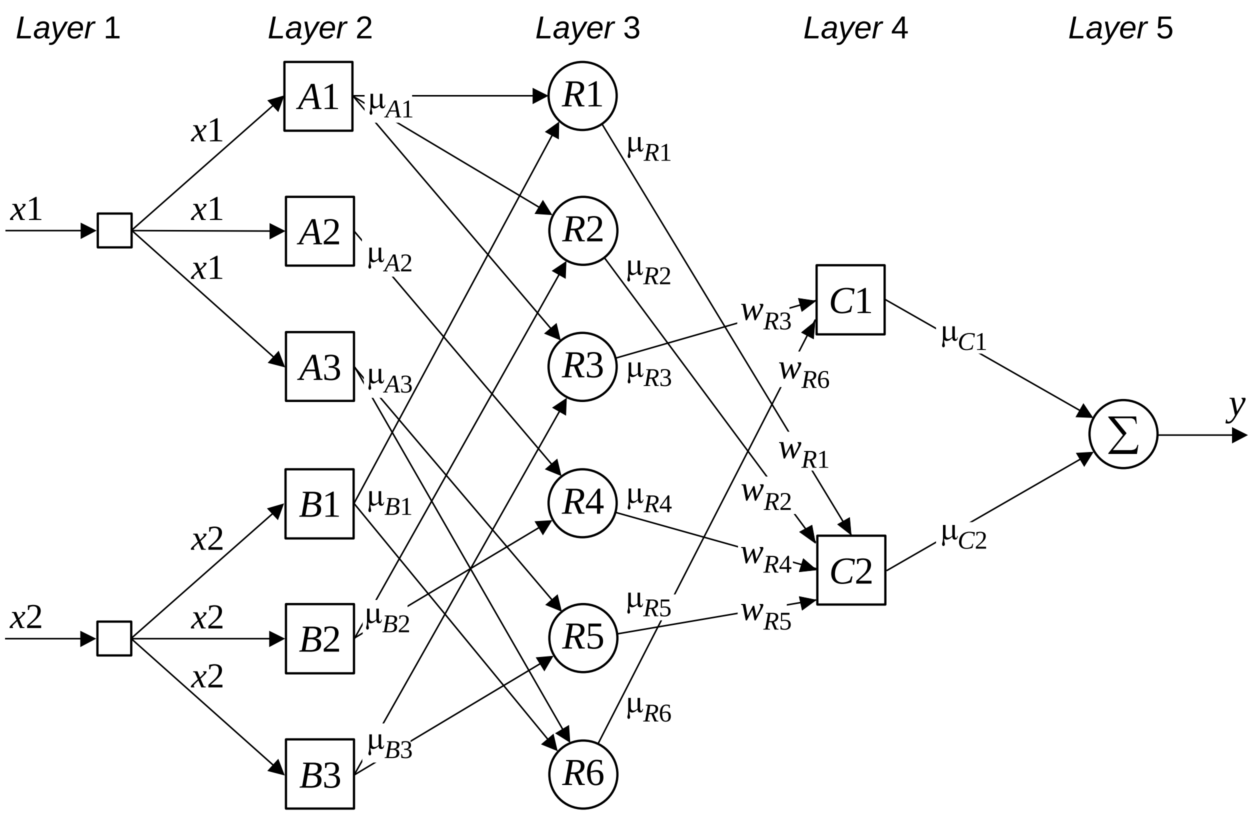


The heart of a neural expert system is the **inference engine**. It controls the information flow in the system and initiates inference over the neural knowledge base. A neural inference engine also ensures **approximate reasoning**.

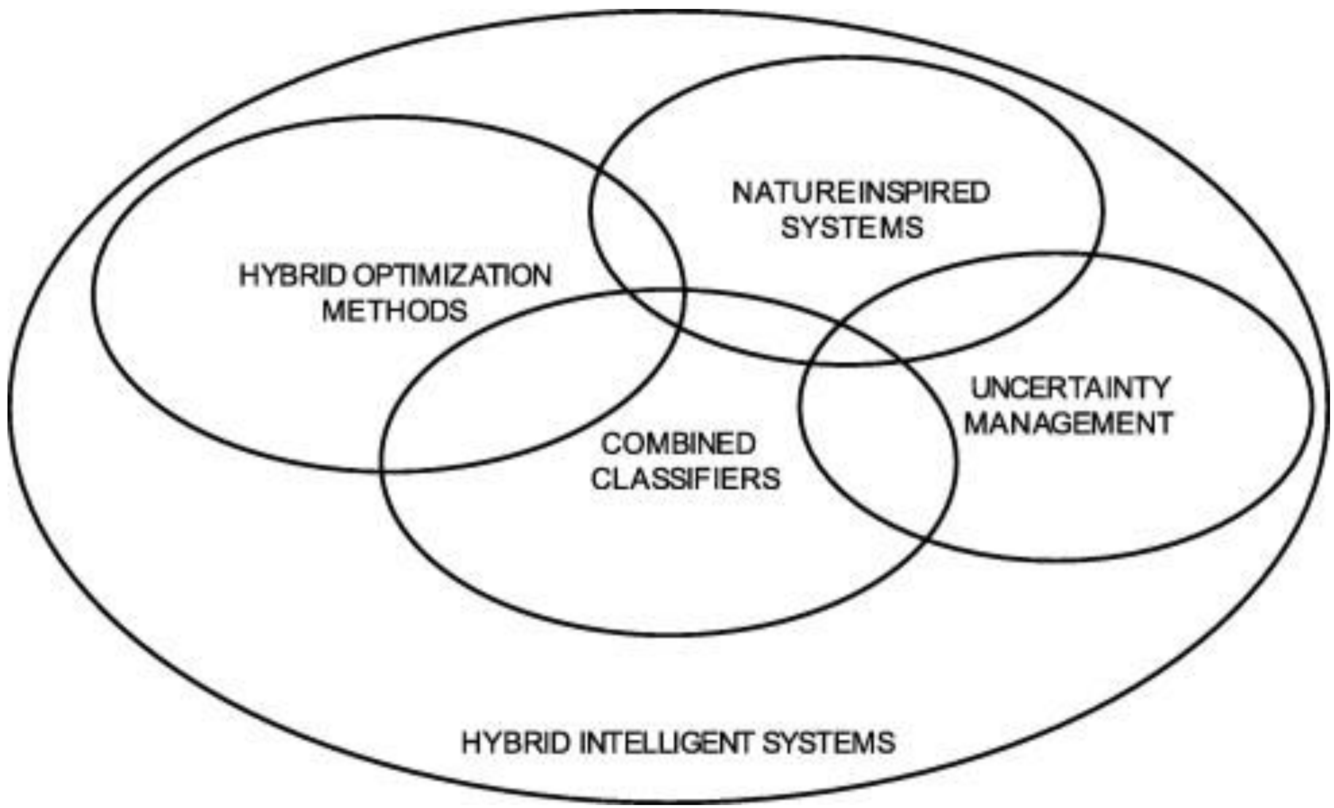
Neuro-Fuzzy Systems
(embedding)



The structure of a neuro-fuzzy system is similar to a multi-layer neural network. In general, a neuro-fuzzy system has input and output layers, and three hidden layers that represent membership functions and fuzzy rules.



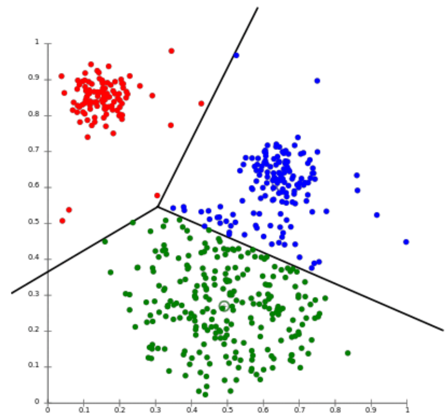
Hybrid Intelligent Systems



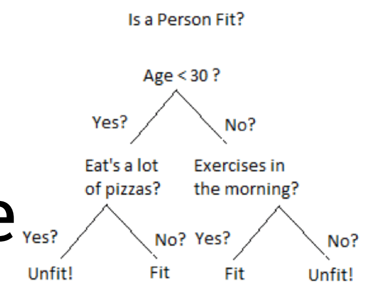
Other systems

- There are many other approaches and types in artificial intelligence and machine learning, but they are outside the scope of this course.

Main points



- Clustering (k means)
- Decision Trees



- Main types of computational intelligence systems
  - Artificial neural networks
  - Fuzzy systems
  - Evolutionary systems
  - Agents
  - Hybrid systems
  - Other systems

Simple Neural Network