ELTE | FACULTY OF INFORMATICS

# Supervised Learning

**Introduction to Machine Learning – Lecture 4**
Computer Science BSc Course, ELTE Faculty of Informatics

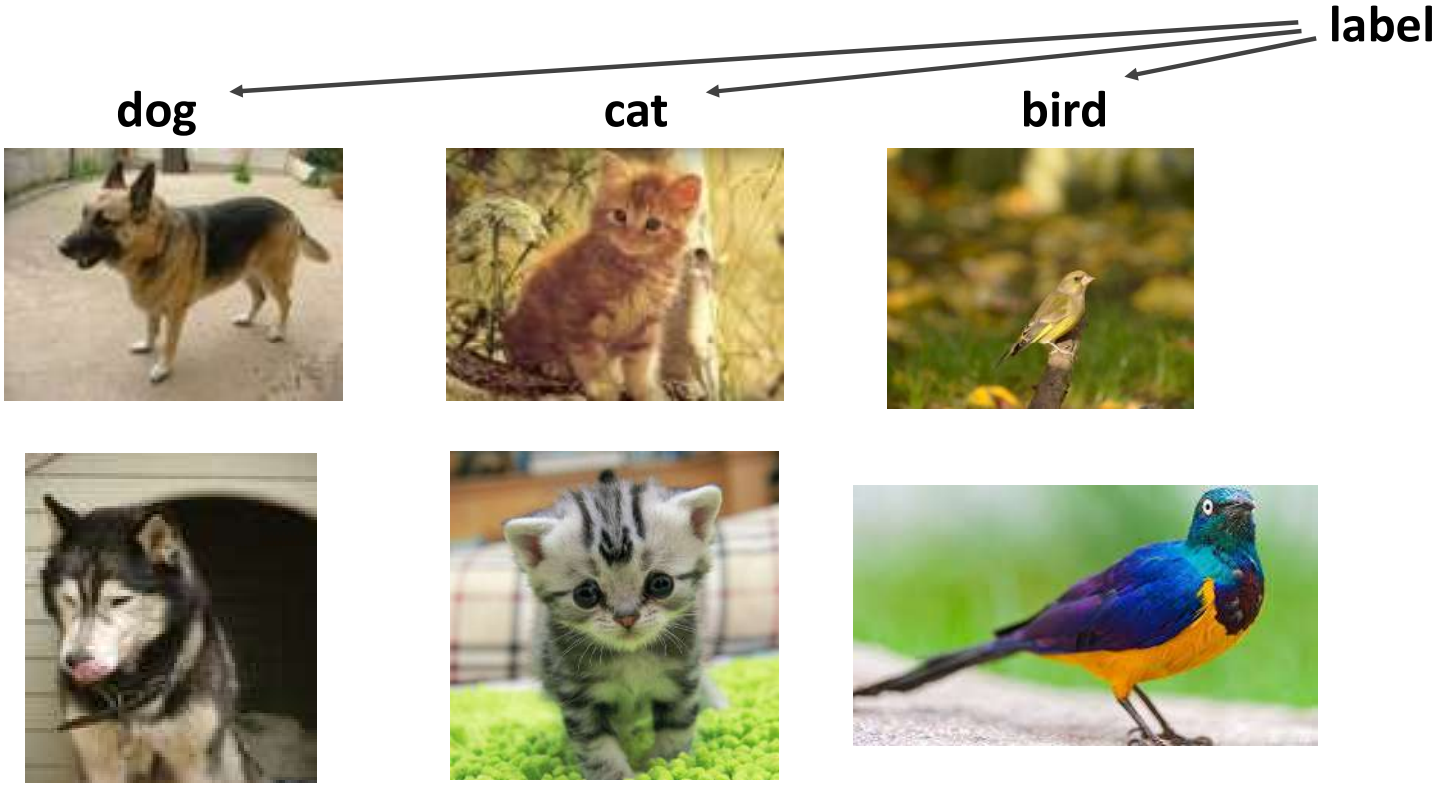
János Botzheim

Associate Professor
Department of Artificial Intelligence
botzheim@inf.elte.hu

ELTE | IK
DEPARTMENT OF ARTIFICIAL INTELLIGENCE

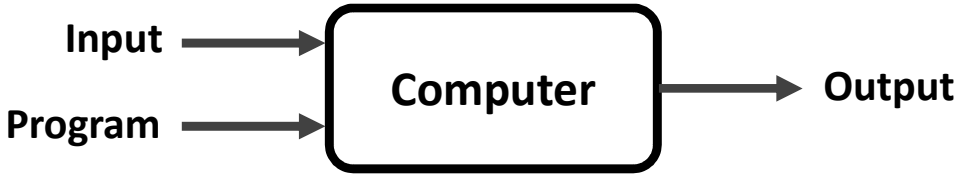
# Unsupervised learning

# Supervised learning

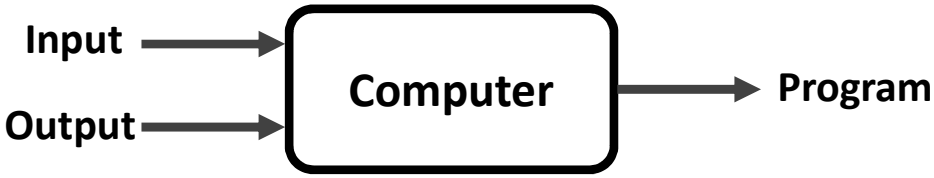


# Learning - Programming

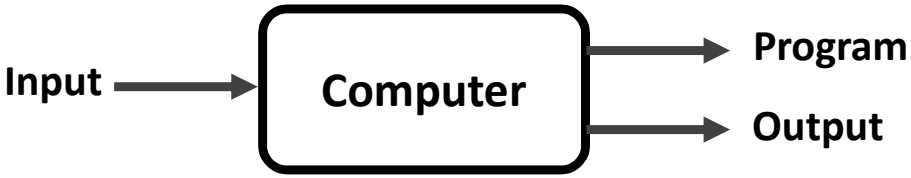
- Traditional computer programming:




- Supervised learning:




- Unsupervised learning:



# Types of learning

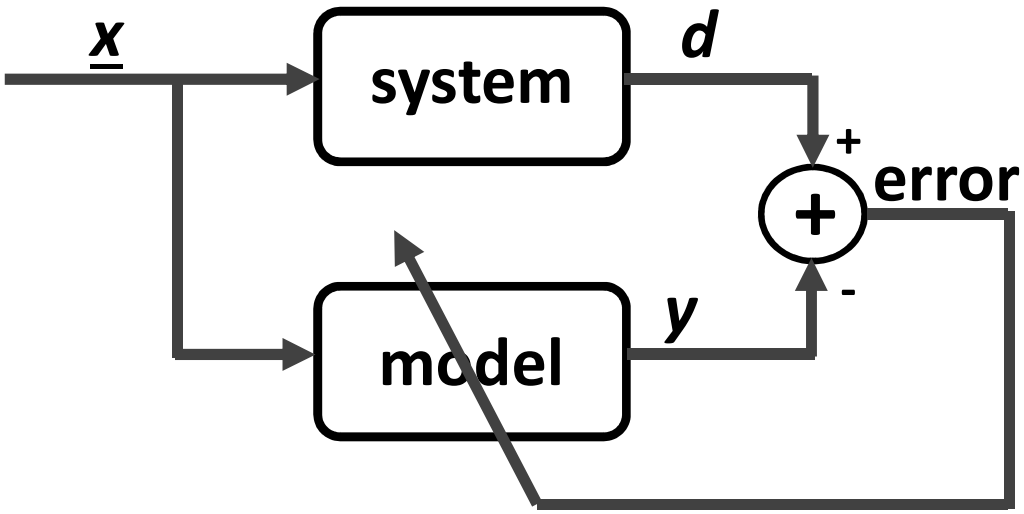
- **Learning**
  - Agent can improve its performance based on observations
- **Machine Learning**
  - Subset of artificial intelligence, computer system learns by patterns in observation data

- Learning tasks
  - **Classification:** output is one of a finite set of values
  - **Regression:** output is a numeric prediction

- Learning Feedback
  - **Supervised Learning** – learn a function that maps inputs to outputs by observing input-output pairs
  - **Unsupervised Learning** – learn patterns in data without explicit feedback (e.g. clustering)
  - **Reinforcement Learning** – learn beneficial actions based on rewards and punishments

# Supervised learning

- given: input-output training patterns $(x_1^{(p)}, x_2^{(p)}, \ldots, x_n^{(p)}, d^{(p)})$
- $p$: number of patterns, $n$-dimensional input, scalar output



# Supervised learning

- **Representation:**
  - rule based system
  - fuzzy system
  - decision tree
  - neural network
  - support vector machine
  - etc.
- **Evaluation:**
  - error
  - accuracy
  - cost
  - entropy
  - etc.
- **Optimization:**
  - gradient based algorithm
  - evolutionary algorithm
  - etc.

# Supervised Learning

- More formally

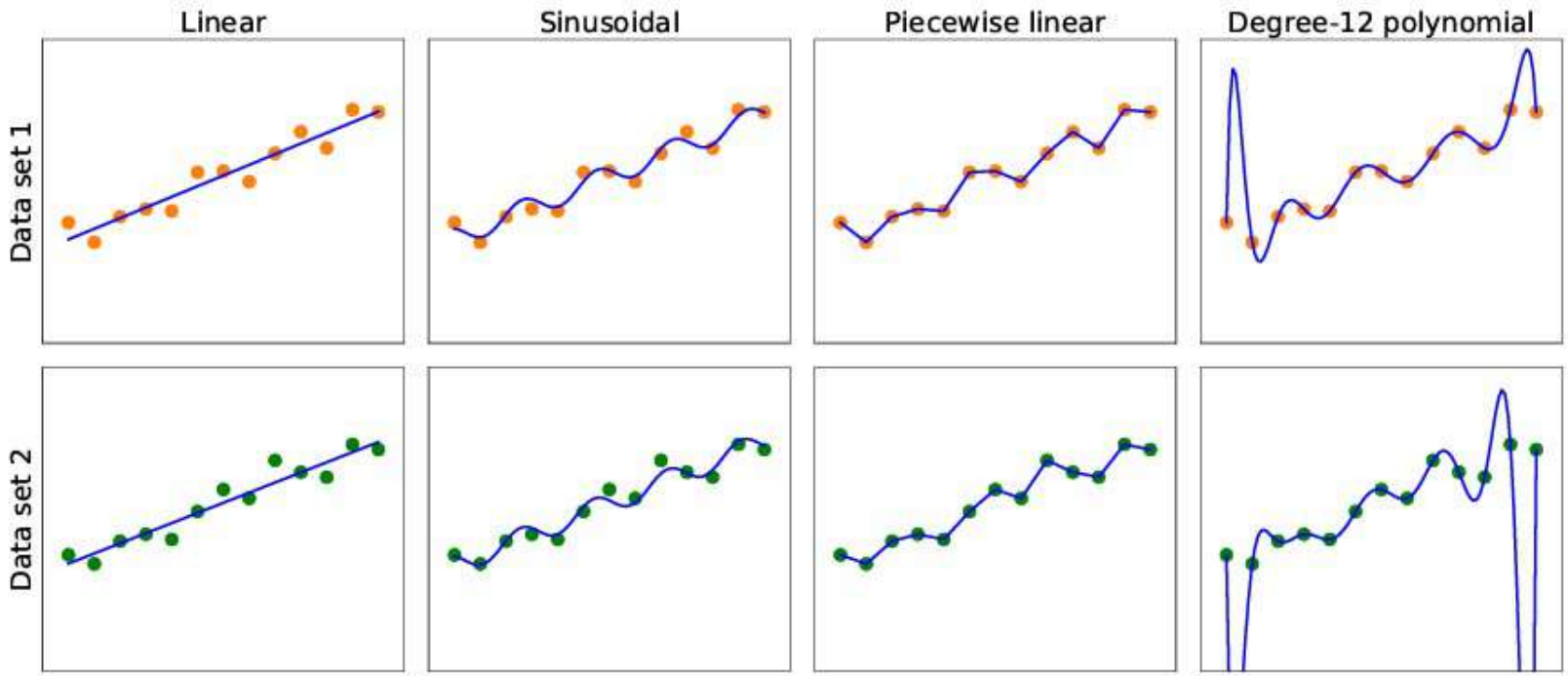
Given a **training set** of $N$ example input–output pairs

$$(x_1, y_1), (x_2, y_2), \ldots (x_N, y_N),$$

where each pair was generated by an unknown function $y = f(x)$, discover a function $h$ that approximates the true function $f$.

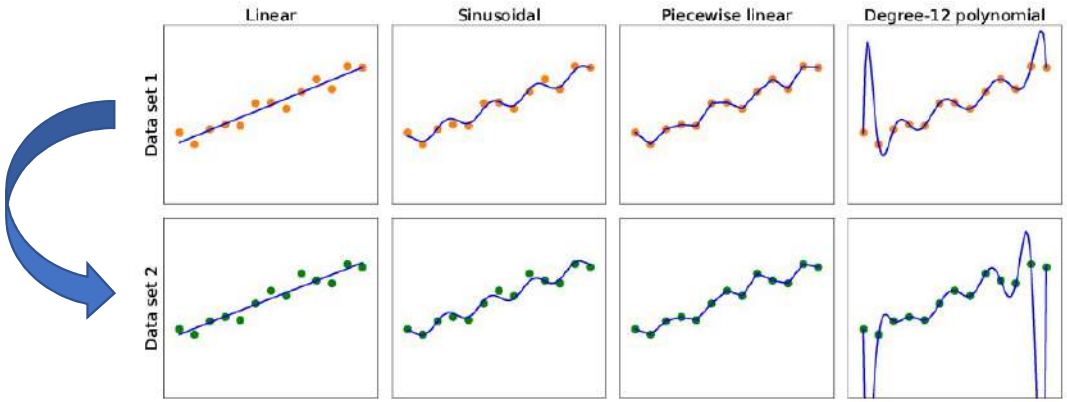
- The function ***h*** is called a **hypothesis** or a **model** of the data

# Supervised Learning



# Supervised Learning - Generalization

- Performance measures
  - Performance on the training set
  - Performance on test set
- **Generalization**
  - ***h*** generalizes well if it accurately predicts the outputs of the **test set**



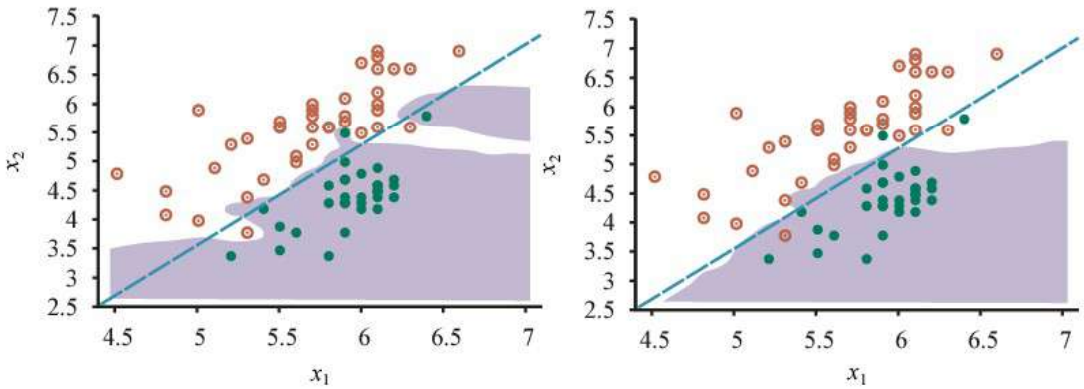
# Supervised Learning - Bias & Variance

- One way to analyze **hypothesis spaces** is by the **bias** they impose (regardless of the training data set) and the **variance** they produce (from one training set to another).
- **Bias**
  - The tendency of a predictive hypothesis to deviate from the expected value when averaged over different training sets
- **Variance**
  - The amount of change in the hypothesis due to fluctuation in the training data

- **Bias–variance tradeoff**
  - more complex, low-bias hypotheses that fit the training data well
  - simpler, low-variance hypotheses that may generalize better

# Supervised Learning - Underfitting & Overfitting

- **Underfitting**
  - The algorithm fails to find a pattern in the data
  - Model is too simple
- **Overfitting**
  - Fits too much to the **particular data set** it is trained on
  - Performs poorly on unseen data (train-test accuracy)
  - Model is too complex
  - Amount of data is not enough
  - Data is not diverse enough



# Decision Trees

- Maps a vector of **attribute values** to a single output **"decision"**
- Sequence of tests
  - Start from root
  - Explore branches
  - Until a leaf is reached
- **Node = test of input attribute value**
- Input and output values can be **discrete** or **continuous**

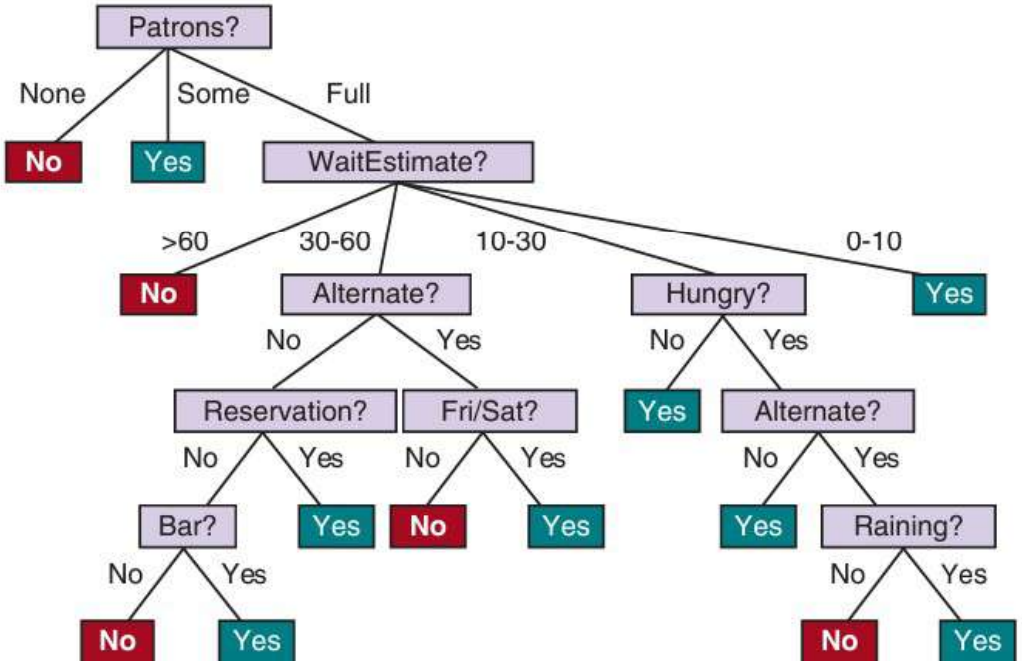
- Simple example: **Boolean decision tree**
- A Boolean decision tree is equivalent to a logical statement of the form:

$$Output \Leftrightarrow (Path_1 \vee Path_2 \vee \cdots),$$

- Where $(A_m = v_x \wedge A_n = v_y \wedge \cdots)$ are attribute-value tests corresponding to a path from the root to a true leaf

# Decision Trees

- Example: Waiting in restaurant



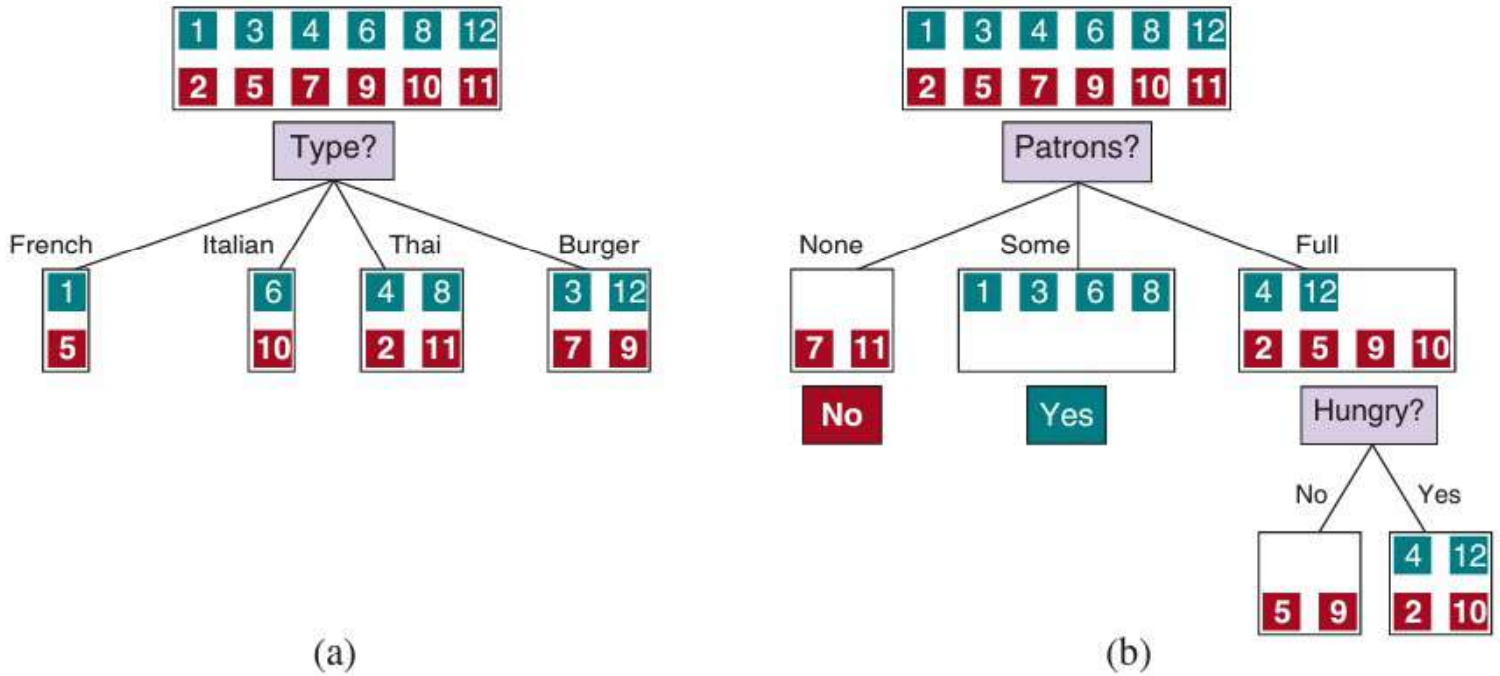
# Decision Trees

- Training dataset

| Example | Input Attributes | | | | | | | | | | Output |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | *Alt* | *Bar* | *Fri* | *Hun* | *Pat* | *Price* | *Rain* | *Res* | *Type* | *Est* | *WillWait* |
| $\mathbf{x}_1$ | *Yes* | *No* | *No* | *Yes* | *Some* | *$$$* | *No* | *Yes* | *French* | *0–10* | $y_1 = Yes$ |
| $\mathbf{x}_2$ | *Yes* | *No* | *No* | *Yes* | *Full* | *$* | *No* | *No* | *Thai* | *30–60* | $y_2 = No$ |
| $\mathbf{x}_3$ | *No* | *Yes* | *No* | *No* | *Some* | *$* | *No* | *No* | *Burger* | *0–10* | $y_3 = Yes$ |
| $\mathbf{x}_4$ | *Yes* | *No* | *Yes* | *Yes* | *Full* | *$* | *Yes* | *No* | *Thai* | *10–30* | $y_4 = Yes$ |
| $\mathbf{x}_5$ | *Yes* | *No* | *Yes* | *No* | *Full* | *$$$* | *No* | *Yes* | *French* | *>60* | $y_5 = No$ |
| $\mathbf{x}_6$ | *No* | *Yes* | *No* | *Yes* | *Some* | *$$* | *Yes* | *Yes* | *Italian* | *0–10* | $y_6 = Yes$ |
| $\mathbf{x}_7$ | *No* | *Yes* | *No* | *No* | *None* | *$* | *Yes* | *No* | *Burger* | *0–10* | $y_7 = No$ |
| $\mathbf{x}_8$ | *No* | *No* | *No* | *Yes* | *Some* | *$$* | *Yes* | *Yes* | *Thai* | *0–10* | $y_8 = Yes$ |
| $\mathbf{x}_9$ | *No* | *Yes* | *Yes* | *No* | *Full* | *$* | *Yes* | *No* | *Burger* | *>60* | $y_9 = No$ |
| $\mathbf{x}_{10}$ | *Yes* | *Yes* | *Yes* | *Yes* | *Full* | *$$$* | *No* | *Yes* | *Italian* | *10–30* | $y_{10} = No$ |
| $\mathbf{x}_{11}$ | *No* | *No* | *No* | *No* | *None* | *$* | *No* | *No* | *Thai* | *0–10* | $y_{11} = No$ |
| $\mathbf{x}_{12}$ | *Yes* | *Yes* | *Yes* | *Yes* | *Full* | *$* | *No* | *No* | *Burger* | *30–60* | $y_{12} = Yes$ |

# Decision Trees

- Find most important features and solve sub-problems sequentially



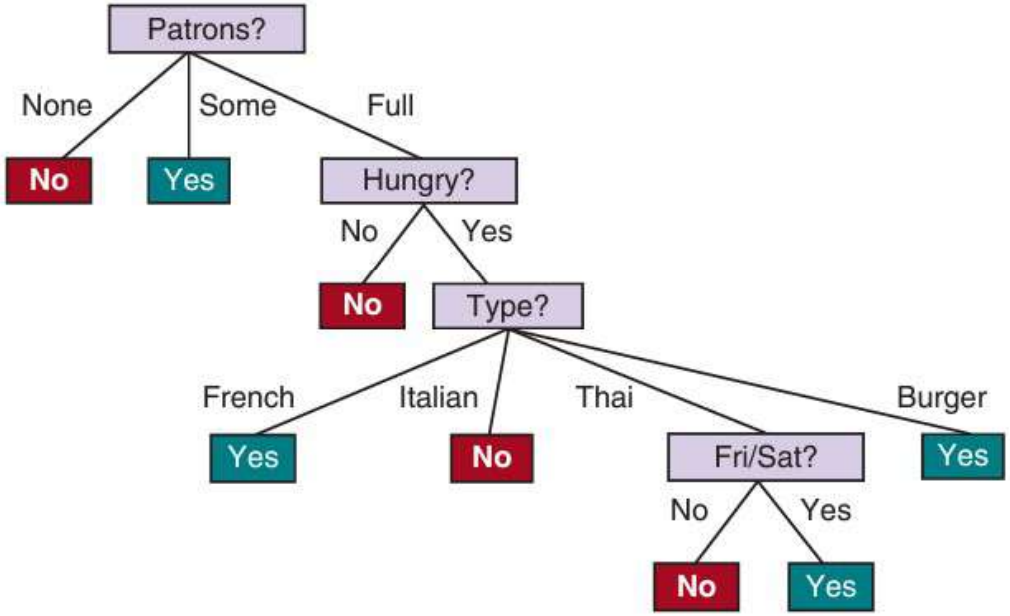
# Decision Trees

```
function LEARN-DECISION-TREE(examples, attributes, parent_examples) returns a tree

  if examples is empty then return PLURALITY-VALUE(parent_examples)
  else if all examples have the same classification then return the classification
  else if attributes is empty then return PLURALITY-VALUE(examples)
  else
     A ← argmax_{a ∈ attributes} IMPORTANCE(a, examples)
     tree ← a new decision tree with root test A
     for each value v of A do
        exs ← {e : e ∈ examples and e.A = v}
        subtree ← LEARN-DECISION-TREE(exs, attributes − A, examples)
        add a branch to tree with label (A = v) and subtree subtree
     return tree
```

# Decision Trees

- Final decision tree has short paths
- Simpler than the original tree
- Fitted to the training set
- **BUT** The learning algorithm looks at the examples, not at the correct function

- Selecting most important attributes based on **information gain** defined by **expected reduction in entropy** of the output variable



# Decision Trees

- **Generalization**
  - If we **increase** the **number of attributes**, overfitting is **more likely**
  - If we **increase** the **number of training samples**, overfitting is **less likely**

- **Pruning**: improving generalization for decision trees
  - Eliminate nodes that are not clearly relevant
    - Look at a test node that has only leaf nodes as descendants
    - If the test appears to be irrelevant - eliminate and replace it with a leaf node
    - Use significance test to measure low information gain threshold

# Decision Trees

- **Pros**
  - Ease of understanding
  - Scalability to large data sets
  - Versatility in handling discrete and continuous inputs
  - Performing classification and regression

- **Cons**
  - Suboptimal accuracy (largely due to the greedy search)
  - If trees are very deep, then getting a prediction for a new example can be expensive
  - Decision trees are unstable – adding just one new example can change the entire tree

# Linear Regression and Classification

- **Univariate Linear Regression** = "fitting a straight line"

  *fit model on* ***(x, y) training examples*** *so that* $\mathbf{y = w_1 x + w_0}$
  *where* $\mathbf{w_0}$ *and* $\mathbf{w_1}$ *are* ***learned weights***

- The learned linear function

$$h_{\mathbf{w}}(x) = w_1 x + w_0$$

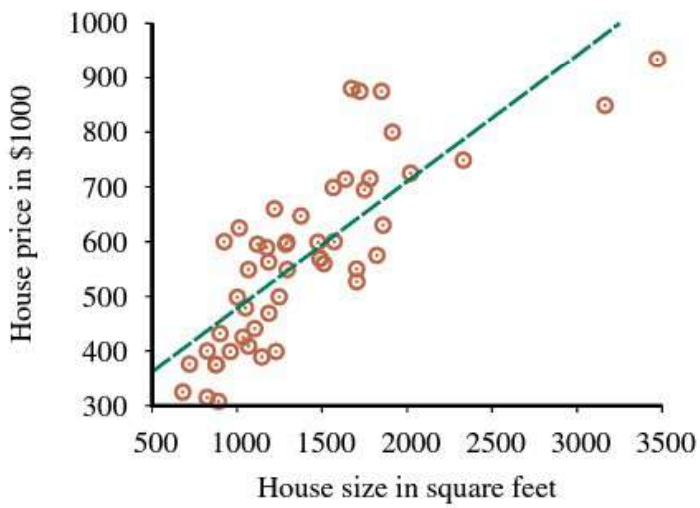
- The task of finding this linear function based on training data is called **linear regression**
  - Find values of $w_0$, $w_1$ that minimizes the empirical loss (e.g. L2 loss)

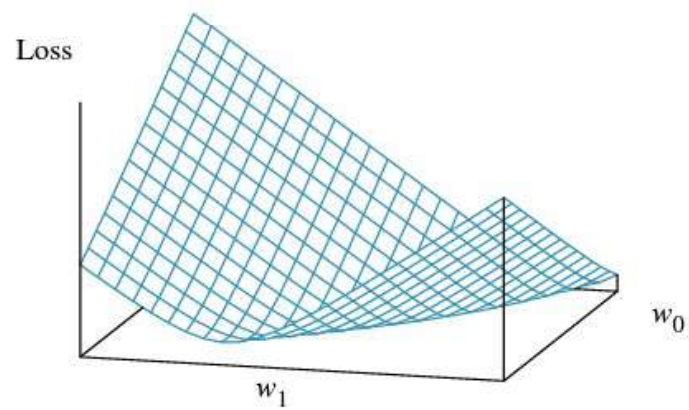
$$Loss(h_{\mathbf{w}}) = \sum_{j=1}^{N} L_2(y_j, h_{\mathbf{w}}(x_j)) = \sum_{j=1}^{N} (y_j - h_{\mathbf{w}}(x_j))^2 = \sum_{j=1}^{N} (y_j - (w_1 x_j + w_0))^2$$

# Linear Regression and Classification

- We can find the function that minimizes the loss using partial derivatives

$$\frac{\partial}{\partial w_0}\sum_{j=1}^{N}(y_j-(w_1x_j+w_0))^2=0 \text{ and } \frac{\partial}{\partial w_1}\sum_{j=1}^{N}(y_j-(w_1x_j+w_0))^2=0$$



# Linear Regression and Classification

- For more complex cases we need to introduce **gradient descent**
  - Algorithm for finding a point in the **weight space** that **minimizes the loss**
  - Similar to the hill climbing but here we are minimizing the loss, not maximizing the gain

- **Gradient Descent**
  - Select random starting point in weight space
  - Compute an estimate of the gradient
  - Move a small amount in the steepest downhill direction
  - Repeat until we converge on a point with minimum loss

  - Parameters
    - Learning rate ($\alpha$) – determines the step size for the descent

$\mathbf{w} \leftarrow$ any point in the parameter space
**while not** converged **do**
  **for each** $w_i$ **in** $\mathbf{w}$ **do**

$$w_i \leftarrow w_i - \alpha \frac{\partial}{\partial w_i} Loss(\mathbf{w})$$

# Linear Regression and Classification

- For the univariate case the loss was quadratic - the partial derivative is linear
- We apply the chain rule

$$\frac{\partial}{\partial w_i} Loss(\mathbf{w}) = \frac{\partial}{\partial w_i}(y - h_{\mathbf{w}}(x))^2 = 2(y - h_{\mathbf{w}}(x)) \times \frac{\partial}{\partial w_i}(y - h_{\mathbf{w}}(x))$$

$$= 2(y - h_{\mathbf{w}}(x)) \times \frac{\partial}{\partial w_i}(y - (w_1 x + w_0)).$$

$$\frac{\partial}{\partial w_0} Loss(\mathbf{w}) = -2(y - h_{\mathbf{w}}(x)); \qquad \frac{\partial}{\partial w_1} Loss(\mathbf{w}) = -2(y - h_{\mathbf{w}}(x)) \times x$$

- Create the update function with the selected learning rate

$$w_0 \leftarrow w_0 + \alpha(y - h_{\mathbf{w}}(x)); \quad w_1 \leftarrow w_1 + \alpha(y - h_{\mathbf{w}}(x)) \times x$$ Single example

$$w_0 \leftarrow w_0 + \alpha \sum_j (y_j - h_{\mathbf{w}}(x_j)); \quad w_1 \leftarrow w_1 + \alpha \sum_j (y_j - h_{\mathbf{w}}(x_j)) \times x_j$$ Batch

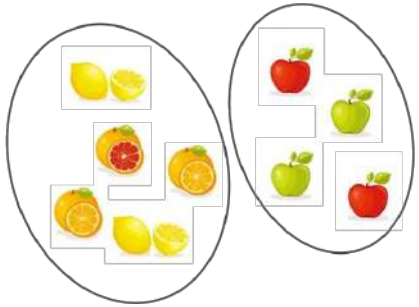
# Unsupervised Learning

# Unsupervised learning

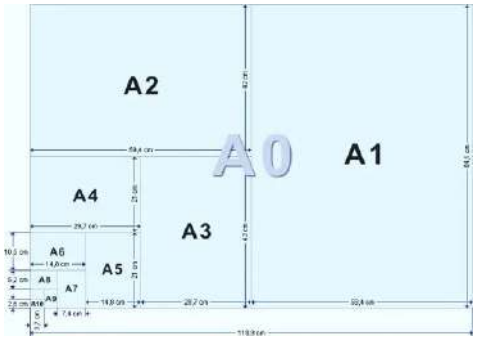
- Labeled data (data whose classification is known) are sometimes not available
- Sometimes not even the classes and their characteristic features are known
- Clusters are data groups in which the points have small distances/high similarity, where the different clusters have a large distance/low similarity

# Unsupervised learning

- Raw data processing
- No desired response vectors
- Finding similarities
- Distinguish groups
- Cluster analysis
- Dimensionality reduction



Cluster analysis

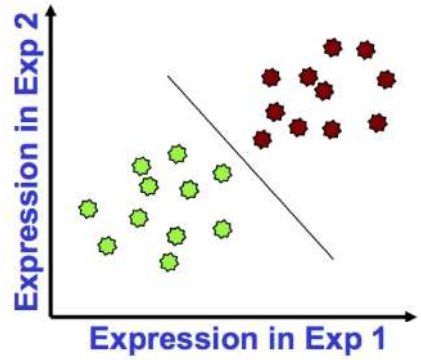


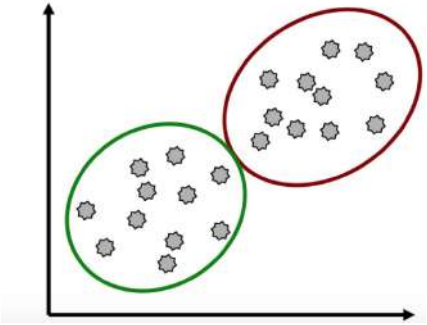

Dimensionality reduction

# Clustering

# Clustering vs. classification

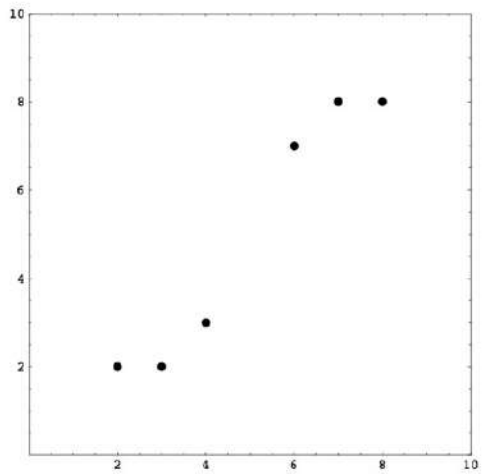
- Objects are characterized by one or more features
- <u>Classification</u>
  - Points have labels
  - Want a “rule” that will accurately assign labels to new points
  - Supervised learning
- <u>Clustering</u>
  - No labels
  - Group points into clusters based on how “near” they are to one another
  - Identify structure in the data
  - Unsupervised learning

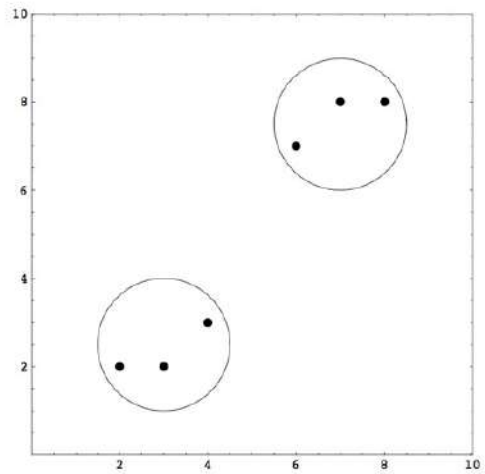


# What is a cluster?

- A group of data in which the elements are very similar/close to each other, while the data in different clusters are significantly different/distant from each other.



# Reasons for the usage

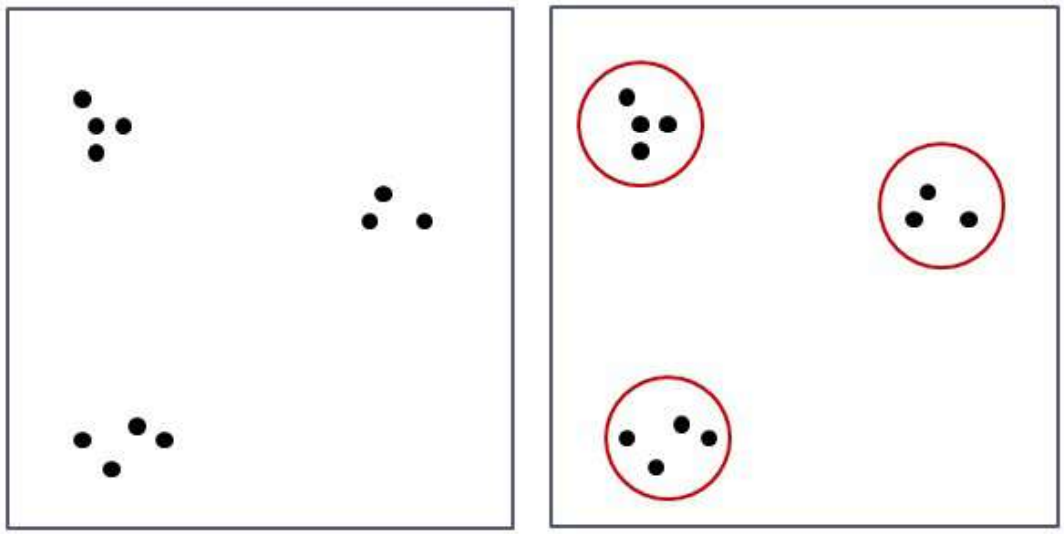
- Labeled (classified) data are not always available
- Sometimes even classes or characteristic properties are not known
- Reducing the number of data

# Few application

- Signal and image decomposition
- Data analysis
- Market analysis (identification of target customer groups)
- Content management (grouping similar documents)
- Fuzzy rule extraction
- ...

# A simple example

- We want to build the rule base from a dataset
- The data set must first be clustered
- Simple example:



# Basic requirements

- We have *n* data samples in *p*-dimensional space

$$(i = 1, \dots, n):$$

$$x_i \in X_1 \times \dots \times X_p$$

- So, all the data are:

$$X = \{x_1, \dots, x_n\}$$

The goal is to define a subset system $C_1, C_2, \dots, C_k \subseteq X$ where:

$$C_i \neq \emptyset \quad \forall\, i$$

$$C_1 \cup \dots \cup C_k = X$$

$$C_i \cap C_j = \emptyset \quad \forall\, i \neq j$$

# Agglomerative clustering

- We start with the most rigorous clustering, i.e. *n* single-element clusters
- Then we iteratively merge the two closest clusters
- We continue until we have only one cluster with all the data elements

- We do not create a partition, but a sequence of partitions
- We do not need to specify the number of clusters in advance
- Most of these methods are slow and only work on small data sets

# SAHN Clustering

Sequential Agglomerative Hierarchical Non-overlapping Clustering

1. Given: $\{x_1, \dots, x_n\} \subseteq R^p$ and $d$ distance metrics
2. Initially: $K_0 = n,\ \Gamma_0 = \{C_{0,1}, \dots, C_{0,k_0}\} = \{\{x_1\}, \dots, \{x_n\}\}$
3. For $i = 0, \dots, n-1$ we look for the $(a, b)$ pair, for which $d(C_{i,a}, C_{i,b})$ is minimal.

$$\Gamma_{i+1} = (\Gamma_i \setminus \{C_{i,a}, C_{i,b}\}) \cup \{C_{i,a} \cup C_{i,b}\}$$

4. Output: $\Gamma_0, \dots, \Gamma_{n-1}$

# SAHN distance metrics

- Minimum

$$d(C_a, C_b) = \min_{x \in C_a,\, y \in C_b} \|x - y\|$$

- Maximum

$$d(C_a, C_b) = \max_{x \in C_a,\, y \in C_b} \|x - y\|$$

- Average

$$d(C_a, C_b) = \frac{1}{|C_a| \cdot |C_b|} \sum_{x \in C_a, y \in C_b} \|x - y\|$$

# Prototype based clustering

- Not all individual data are kept in clusters
- Rather, in each $C_i$ cluster, we designate a point $v_i$ in the data space, usually called the cluster center
- Each data point belongs to that cluster whose center point is the closest to the data
  - So: $x_k \in C_i$ if

# Partition matrix

- Which cluster each data point belongs to
- $U = \begin{pmatrix} u_{11} & \cdots & u_{1n} \\ \vdots & \ddots & \vdots \\ u_{K1} & \cdots & u_{Kn} \end{pmatrix}$ $\qquad u_{ik} = \begin{cases} 1 & if\ \ x_k \in C_i \\ 0 & otherwise \end{cases}$
- All data are elements of a cluster:
- There is no empty cluster:

# k-means clustering

- Iterative process for clustering
  - Minimize the following function

- Calculating the centroids

(1)

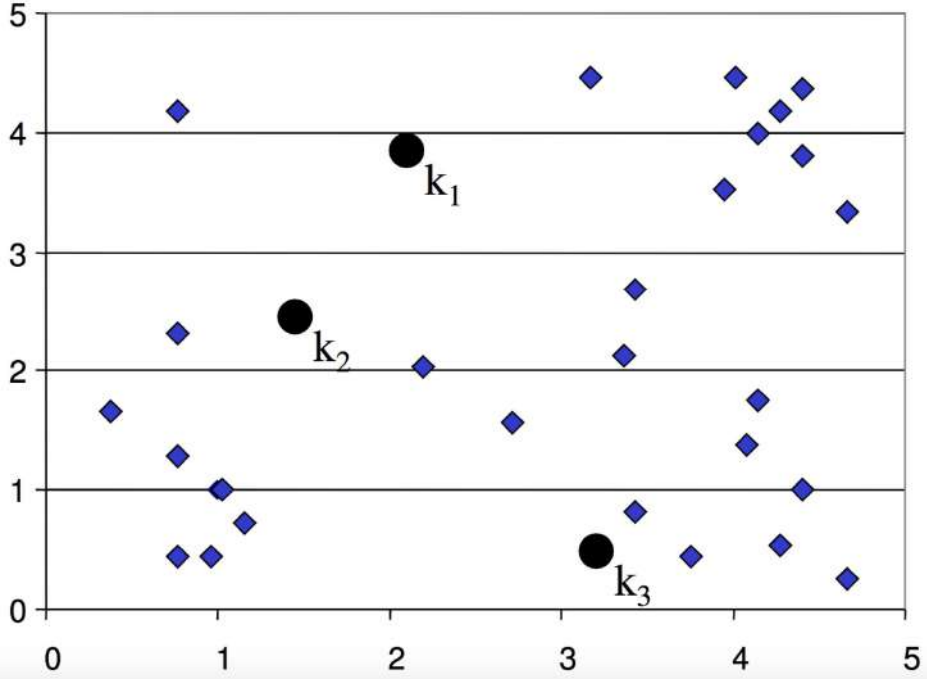
# k-means clustering

1. Input: a data set $\{x_1, \ldots, x_n\} \subseteq R^p$, $\|.\|$ is a norm defined on $R^p$ , $K$ is the predefined number of clusters, $t$_max is the maximum number of iterations, $\|.\|_v$ is a distance measure, $\varepsilon$ is the tolerance
2. Initialization: $V^{(0)} \subseteq R^p$
3. For $t = 0, \ldots, t_{\max}$
   1. Determine the partition matrix $U^{(t)}\left(V^{(t-1)}\right)$
   2. Determine the cluster centroids $V^{(t)}\left(U^{(t)}\right)$ (from equation (1))
   3. If $\left\|V^{(t)} - V^{(t-1)}\right\|_v \leq \varepsilon$ ,end
4. Result:

   The partition matrix *U* and the cluster centers *V*

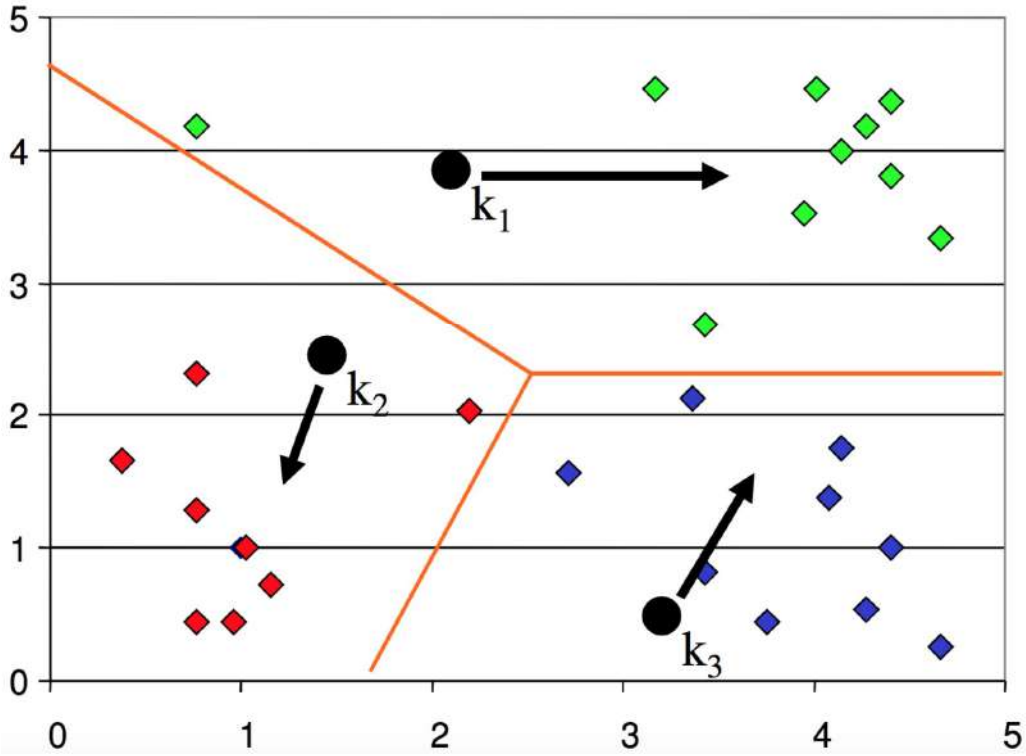
# k-means clustering

- Initialization:
  - decide K
  - initialize K cluster centers randomly



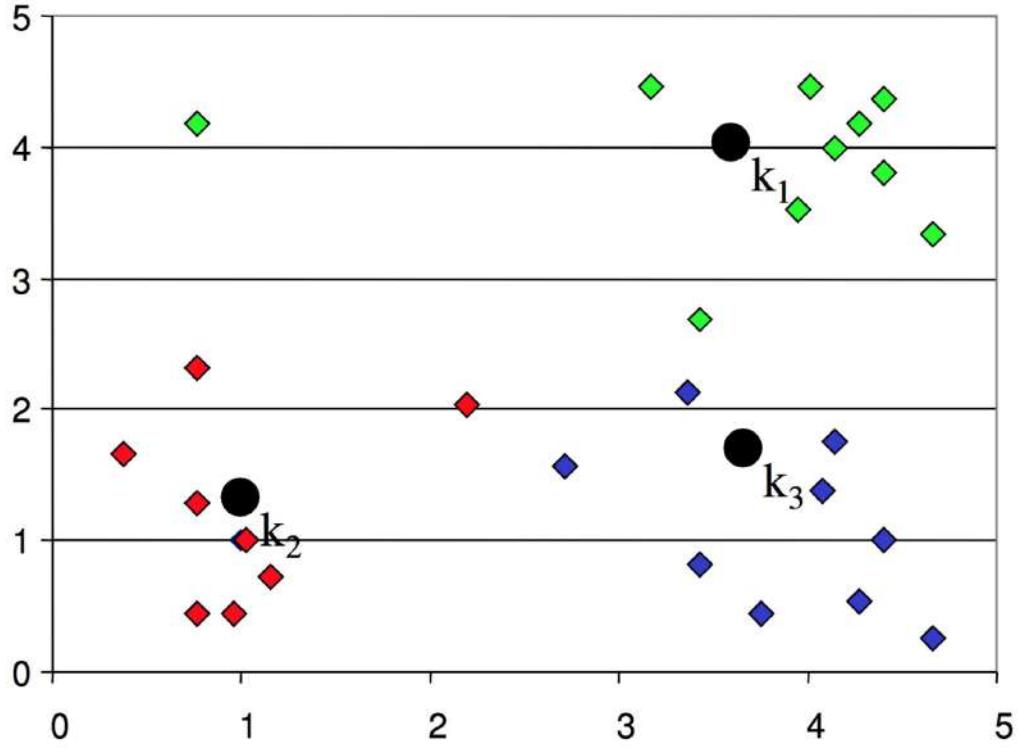
# k-means clustering

- Iteration:
  - assign all points to the nearest cluster center
  - move the centers to the mean of their members



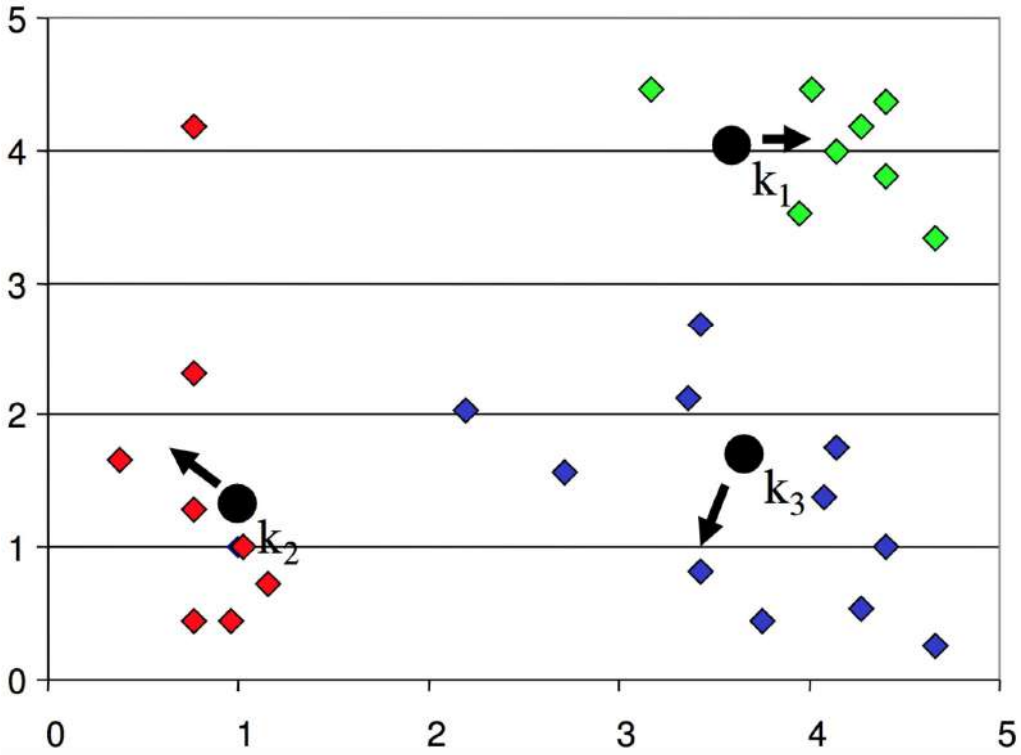
# k-means clustering

- Iteration:
  - after moving centers, re-assign the points to the nearest center



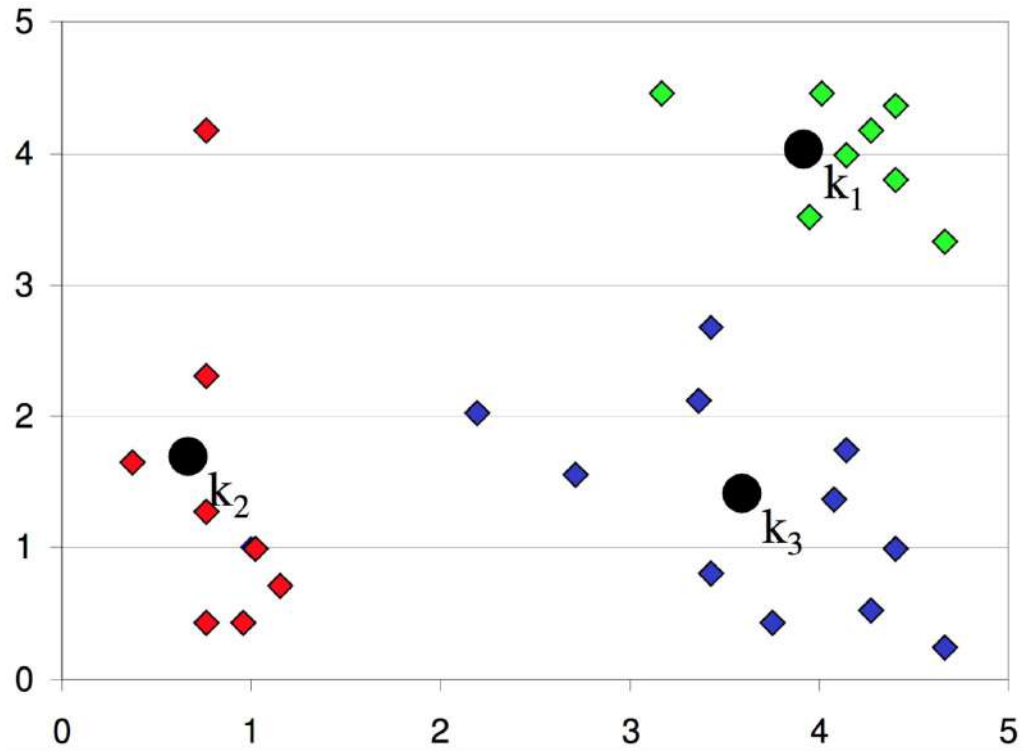
# k-means clustering

- Iteration:
  - move the centers to the mean of their members



# k-means clustering

- Finished:
  - re-assign and move centers, until no points changed membership



# k-means clustering

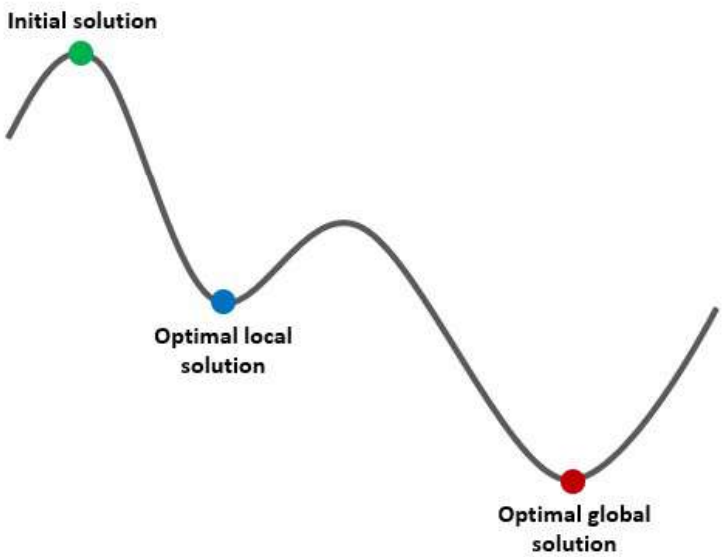
- 1. Decide the value of *K*, the number of clusters
- 2. Initialize the *K* cluster centers randomly
- 3. Decide the class membership of the *N* points by assigning them to the nearest cluster center
- 4. Re-estimate the *K* cluster centers, by assuming the memberships found above are correct
- 5. Repeat 3 and 4 until none of the *N* points changed membership in the last iteration

# k-means clustering

- **<u>Strength</u>:**
  - Simple, easy to implement
  - Intuitive objective function: optimizes intra-cluster similarity
  - Relatively efficient: $O(t \cdot k \cdot n)$, where $n$: number of points (data), $k$: number of clusters, $t$: number of iterations, usually: $k,t \ll n$
- **<u>Weakness</u>:**
  - Applicable only when *mean* is defined
  - Often terminates at a local optimum. Initialization is important
  - Need to specify $K$, the number of clusters, in advance
  - Unable to handle noisy data and outliers
  - Not suitable to discover clusters with non-convex shapes

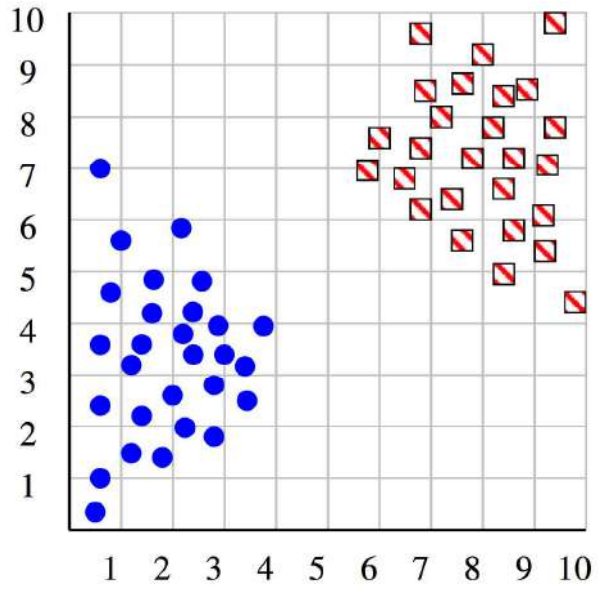
# k-means clustering

Initialization of centroids can influence the final clustering result



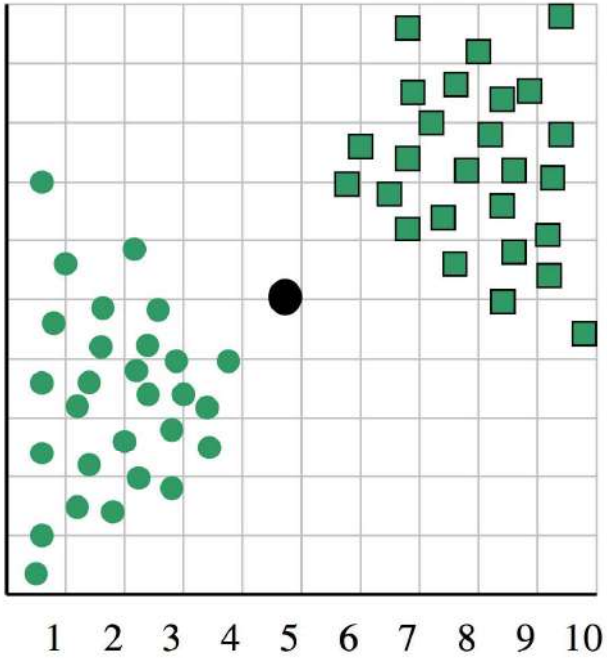
# How to define the right number of clusters?

In general, this is a unsolved problem. However there are many approximate methods. In the next few slides we will see an example.



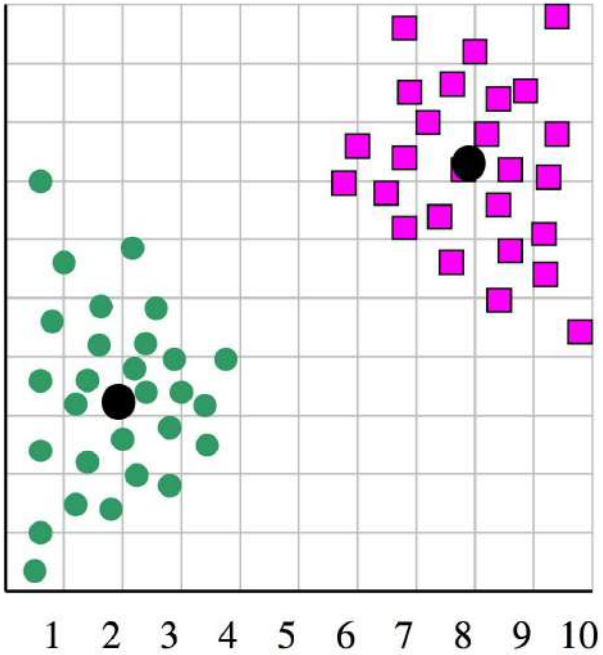
# How to define the right number of clusters?

When k = 1, the objective function is 873.0



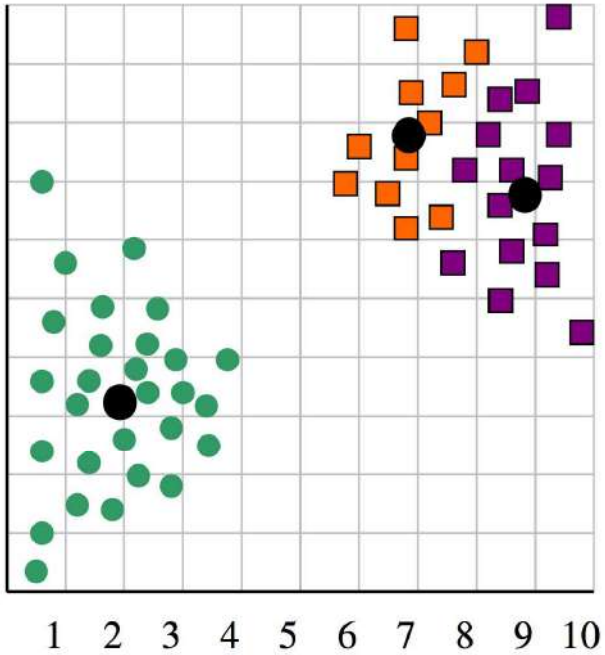
# How to define the right number of clusters?

When k = 2, the objective function is 173.1



# How to define the right number of clusters?

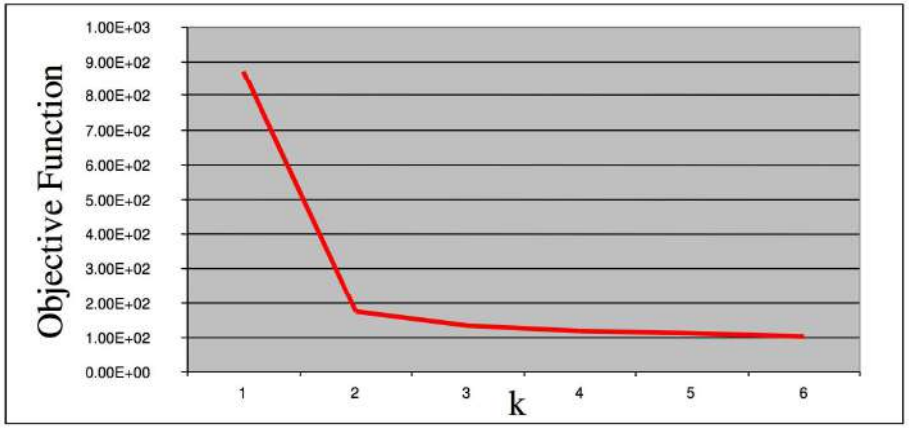
When k = 3, the objective function is 133.6



# How to define the right number of clusters?

We can plot the objective function values for k equals 1 to 6…

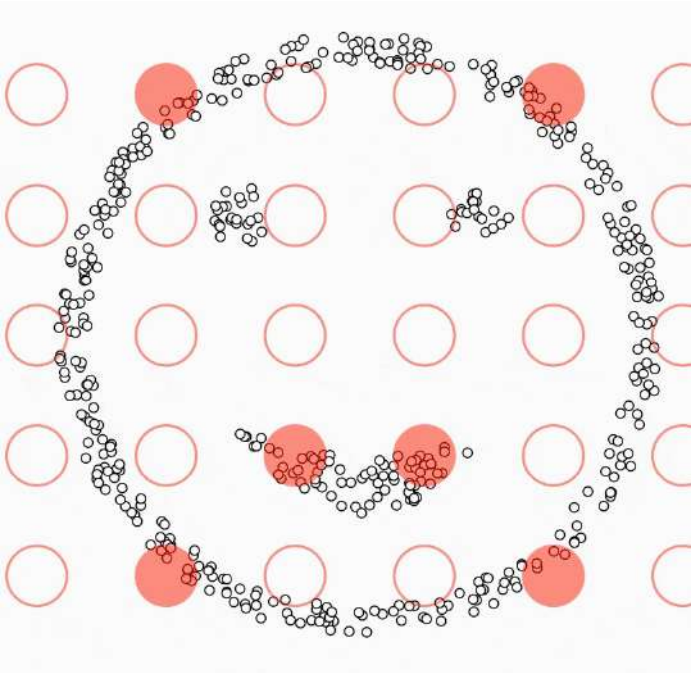
The abrupt change at k = 2, is highly suggestive of two clusters in the data. This technique for determining the number of clusters is known as "knee finding" or "elbow finding".



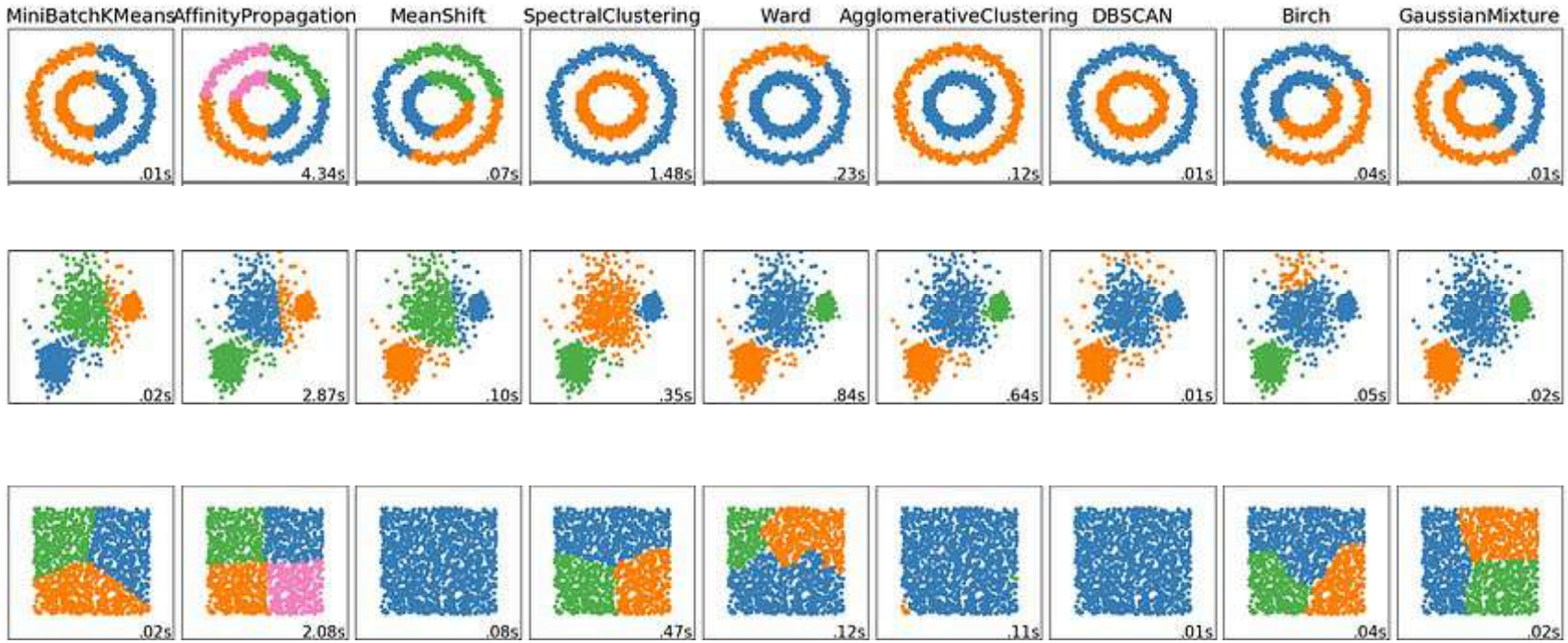

Note that the results are not always as clear cut as in this toy example

# Density-Based Spatial Clustering of Applications with Noise (DBSCAN)

1. Set the parameters:
   - Epsilon (ε): The maximum distance between two data points for them to be considered neighbors.
   - MinPts: The minimum number of data points required to form a dense region.
2. Start with an arbitrary unvisited data point.
   - If the point has not been visited, mark it as visited.
   - Retrieve all its neighboring points within distance ε.
3. Check the density of the current point:
   - If the number of neighboring points is greater than or equal to MinPts, the point is considered a core point.
   - If the number of neighboring points is less than MinPts, the point is considered a border point.
4. Expand the cluster:
   - If the point is a core point, create a new cluster and add the point to the cluster.
   - Retrieve all the reachable neighboring points (directly or transitively) from the core point within distance ε.
   - Add these points to the cluster and mark them as visited.
5. Repeat steps 2 to 4 for all unvisited data points in the dataset.
   - If a border point is encountered, it is not added to any cluster.
6. The algorithm continues until all data points have been visited.
7. Classification of noise: Any unvisited data point that does not belong to any cluster is considered noise.
8. The final result is a set of clusters, where each cluster consists of the core points and their reachable neighboring points.



# Comparison of clustering algorithms



ELTE | FACULTY OF INFORMATICS

# Neural networks

**Introduction to Machine Learning – Lecture 10**
Computer Science BSc Course, ELTE Faculty of Informatics

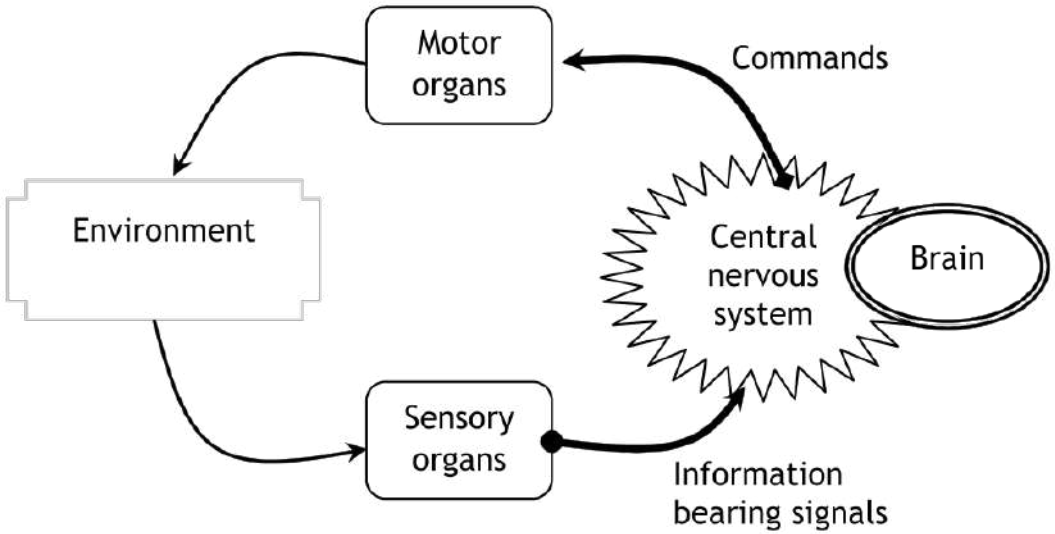
János Botzheim

Associate Professor
Department of Artificial Intelligence
botzheim@inf.elte.hu

ELTE | IK
DEPARTMENT OF ARTIFICIAL INTELLIGENCE

# Human nervous system

- Human nervous system is a very complex system capable of thinking, remembering, problem solving, making decision and etc.

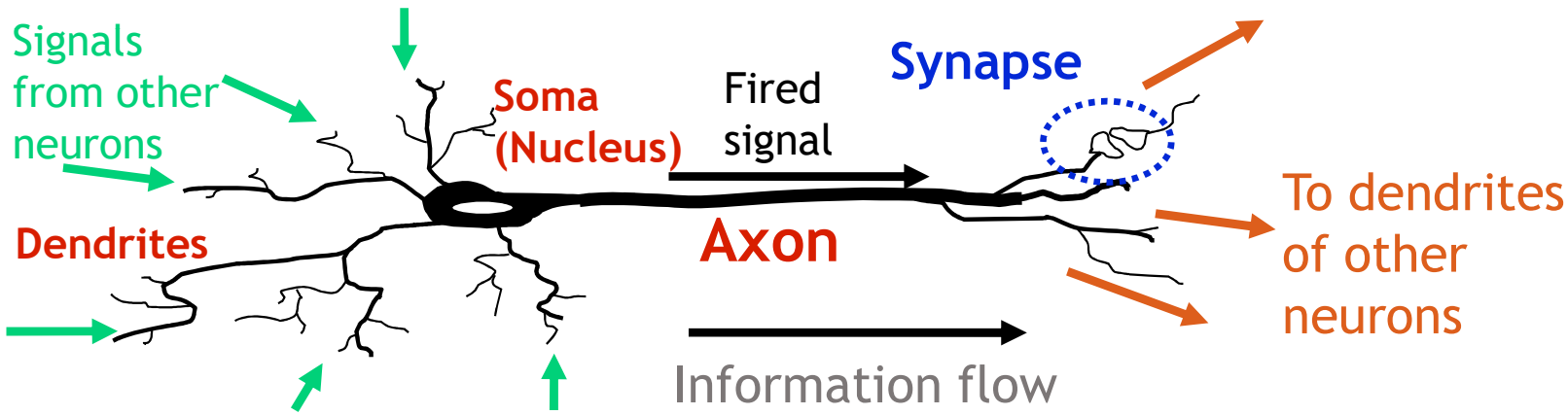


# Few data

- The fundamental cellular unit of nervous system is called neuron.
- A baby is born with about $10^{11}$ neurons in the brain.
- Neuron is a simple processing unit connected to approximately 1000 neurons. Therefore, there are approximately $10^{14}$ connections or transmission paths in the nervous system.
- The estimated memory capacity of the human brain varies between 1 and 1000 terabytes (for comparison, the US Library of Congress currently stores around 10 terabytes of data).

# Biological neurons

- The function of a neuron is receiving and combining signals from other neurons through input paths called dendrites.
- If the combined input signal is strong enough, the neuron fires and produces an output signal and sends the output along the axon to other neurons.

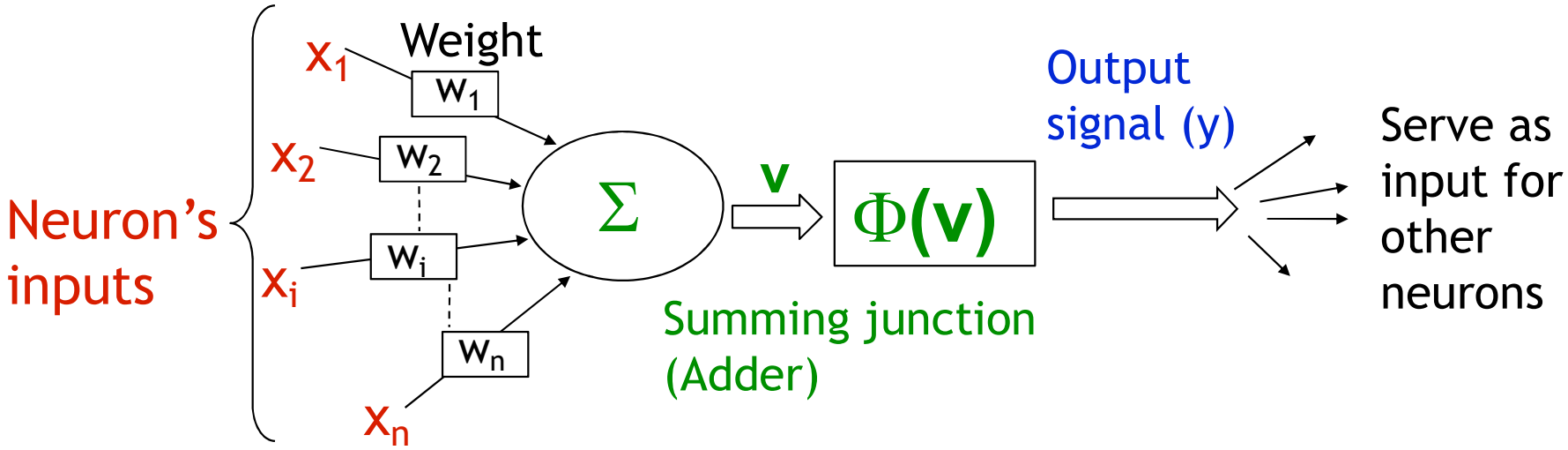


# Computer vs. human nervous system

| | Computer | Human |
|---|---|---|
| Speed | Above 4 GHz | 40-50Hz<br>(Neuron response time - 20-30 ms) |
| Operation mode | Serial mode | Parallel mode (Low speed of brain is compensated by high processing speed inside the brain) |
| Recognizing a person after seeing | A complicated task, depends on the background, viewing point, etc. | Half a second |

# Artificial neuron

- An information processing unit that is fundamental of an artificial neural network.
- A mimicking model for biological neuron.



# Artificial neuron

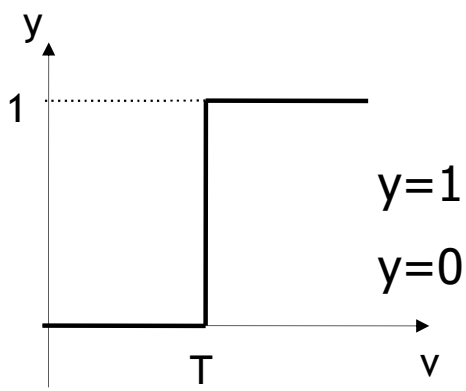
- A neuron can only perform some simple information processing; therefore it is not able to solve complex problems.
- However, neurons can be interconnected to deliver collective complex behavior and thus they can be used for solving complex problems.

# Biological vs. artificial neuron

| Biological neuron | Artificial neuron |
|---|---|
| Soma | Neuron |
| Dendrite | Input |
| Axon | Output |
| Synapse | Weight |

# Activation function (Φ)

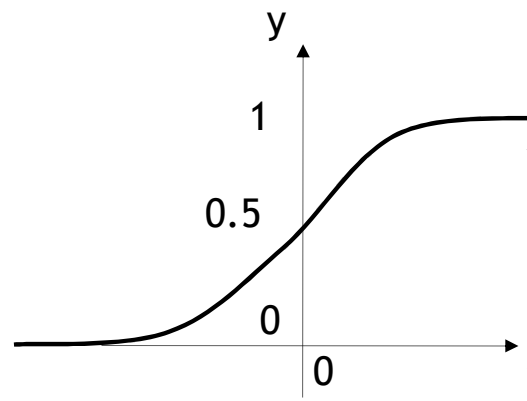
- Limits the output between two asymptotes so that it can keep neuron output within reasonable dynamic range




$y=1 \quad v \geq T$

$y=0 \quad v < T$

Threshold or step Function




$y = 1/(1+e^{-\alpha v})$

$\alpha$: adjusts the abruptness of the function

Sigmoid function

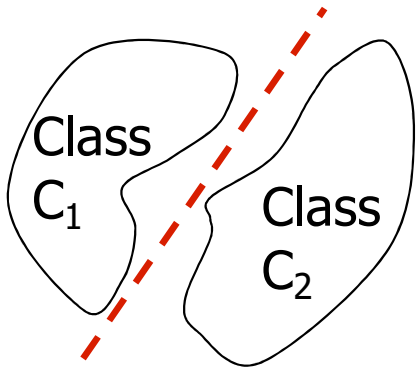
# Single layer perceptron

- Single layer perceptron was proposed by Rosenblatt in 1958 as the 1st model for learning with a teacher (i.e., supervised learning).
- His perceptron was based on McCulloch-Pitts model of neuron.

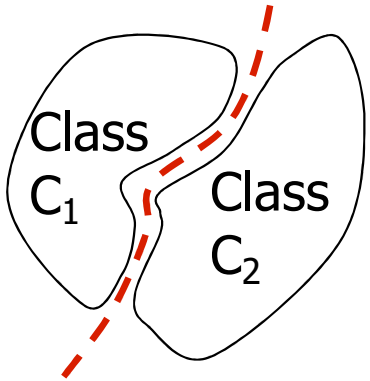
# Single layer perceptron

- Used for solving linearly separable patterns.
- This caused serious disappointment in the neural network community and put neural network researchers in a dark period (1970s).
- However, this statement was not correct in general, because multilayer perceptrons are capable of separating linearly inseparable variables (including XOR problem).

**linearly** separable patterns

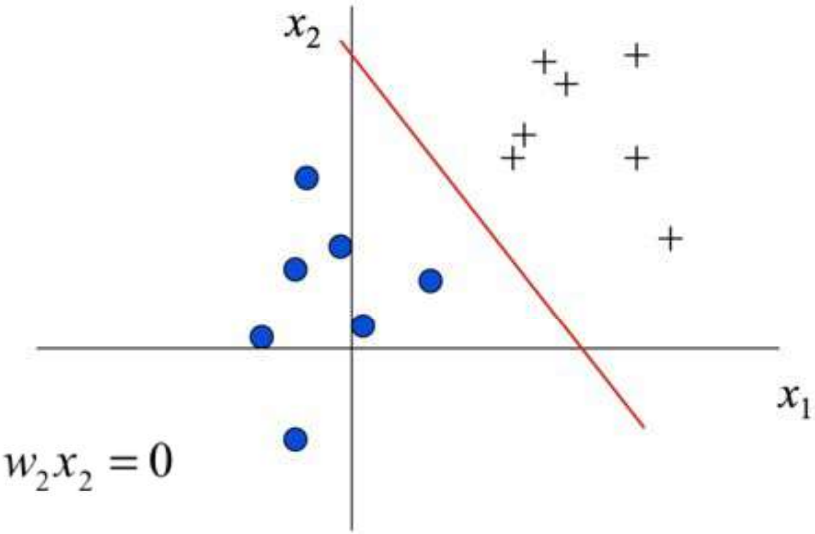



**nonlinearly** separable patterns



# Linear separability

- Consider the following example with two inputs ($x_1$,$x_2$)
- Can the trained network define the "separation line"?

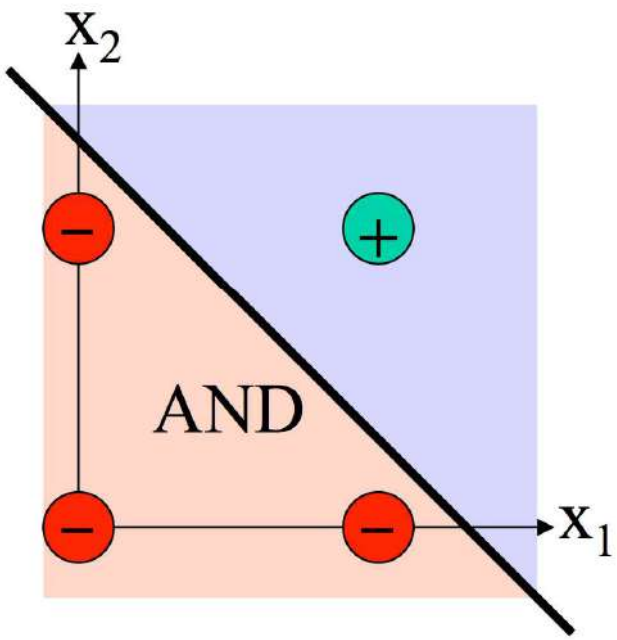


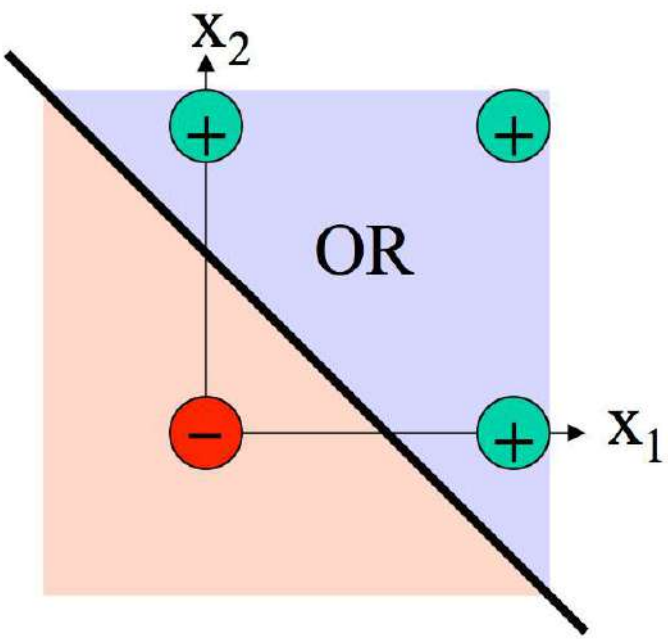

$$-w_0 + w_1 x_1 + w_2 x_2 = 0$$

- What is its equation?

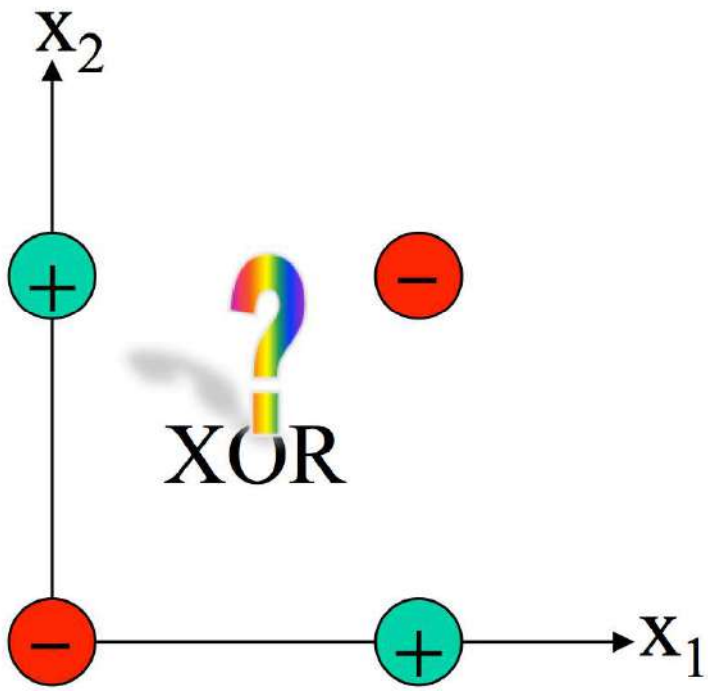
$$x_2 = -\frac{w_1}{w_2} x_1 + \frac{w_0}{w_2}$$

# Linear separability

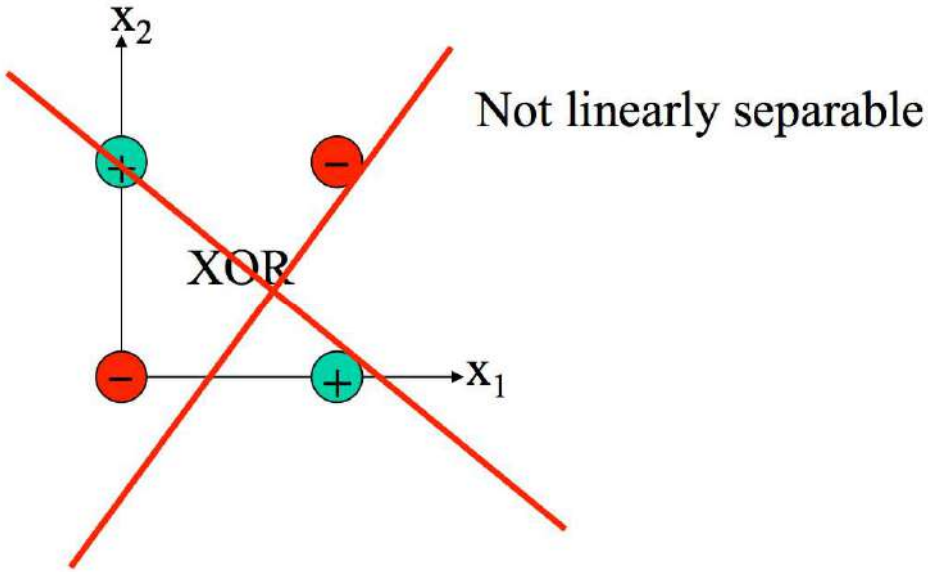


# Linear separability



# Linear separability




Minsky & Papert (1969)

Bad News: Perceptrons can only represent linearly separable functions.

# Implementing logic gates

- We can use McCulloch-Pitts neurons to implement the basic logic gates
- All we need to do is find the appropriate connection weights and neuron threshold to produce the right outputs for each set of inputs
- We shall see explicitly how one can construct simple networks that perform NOT, AND, and OR.
- It is then a well known result from logic that we can construct any logical function from these three operations
- The resulting networks, however, will usually have a much more complex architecture that a simple Perceptron
- We generally want to avoid decomposing complex problems into simple logic gates, by finding the weights and thresholds that work directly in a Perceptron architecture

# Implementation of logical NOT, AND, and OR

- In each case we have inputs $in_i$ and output *out*, and need to determine the weights and thresholds

NOT

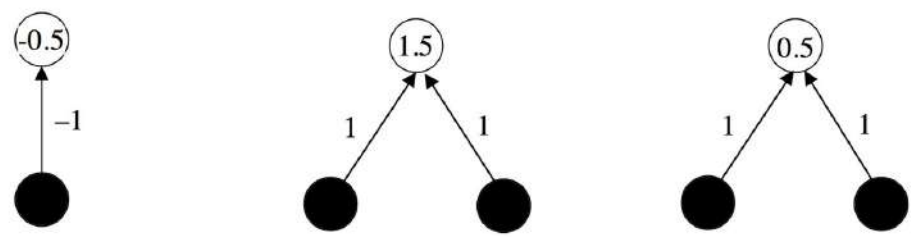
| *in* | *out* |
|---|---|
| 0 | 1 |
| 1 | 0 |

AND

| $in_1$ | $in_2$ | *out* |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

OR

| $in_1$ | $in_2$ | *out* |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

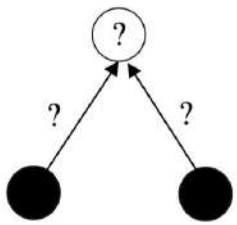


# The need to find weights analytically

- Constructing simple networks by hand is one thing. But what about harder problems?
- For example:

XOR

| $in_1$ | $in_2$ | $out$ |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |




- How long do we keep looking for a solution?
- We need to be able to calculate appropriate parameters rather than looking for solutions by trial and error.
- Each training pattern produces a linear inequality for the output in terms of the inputs and the network parameters. These can be used to compute the weights and thresholds.

# Finding weights analytically for the AND network

- We have two weights, $w_1$ and $w_2$, and the threshold $\theta$, and for each training pattern we need to satisfy:

$$out = \mathrm{sgn}(w_1 in_1 + w_2 in_2 - \theta)$$

- So the training data lead to four inequalities:

| $in_1$ | $in_2$ | $out$ |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

$\Rightarrow$

$$\begin{aligned} w_1 0 + w_2 0 - \theta &< 0 \\ w_1 0 + w_2 1 - \theta &< 0 \\ w_1 1 + w_2 0 - \theta &< 0 \\ w_1 1 + w_2 1 - \theta &\geq 0 \end{aligned}$$

$\Rightarrow$

$$\begin{aligned} \theta &> 0 \\ w_2 &< \theta \\ w_1 &< \theta \\ w_1 + w_2 &\geq \theta \end{aligned}$$

- It is easy to see that there are an infinite number of solutions. Similarly, there are an infinite number of solutions for the NOT and OR networks.

# Limitations of simple perceptrons

- We can follow the same procedure for the XOR network:

| $in_1$ | $in_2$ | $out$ |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

$\Rightarrow$

$$\begin{aligned} w_1 0 + w_2 0 - \theta &< 0 \\ w_1 0 + w_2 1 - \theta &\geq 0 \\ w_1 1 + w_2 0 - \theta &\geq 0 \\ w_1 1 + w_2 1 - \theta &< 0 \end{aligned}$$

$\Rightarrow$

$$\begin{aligned} \theta &> 0 \\ w_2 &\geq \theta \\ w_1 &\geq \theta \\ w_1 + w_2 &< \theta \end{aligned}$$

- Clearly the second and third inequalities are incompatible with the fourth, so there is in fact no solution. We need more complex networks, e.g. that combine together many simple networks, or use different activation/thresholding/transfer functions.
- It then becomes much more difficult to determine all the weights and thresholds by hand.

# Limitations of simple perceptrons

- We can follow the same procedure for the XOR network:

| $in_1$ | $in_2$ | $out$ |
|---|---|---|
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

$\Rightarrow$

$$\begin{aligned} w_1 0 + w_2 0 - \theta &< 0 \\ w_1 0 + w_2 1 - \theta &\geq 0 \\ w_1 1 + w_2 0 - \theta &\geq 0 \\ w_1 1 + w_2 1 - \theta &< 0 \end{aligned}$$

$\Rightarrow$

$$\begin{aligned} \theta &> 0 \\ w_2 &\geq \theta \\ w_1 &\geq \theta \\ w_1 + w_2 &< \theta \end{aligned}$$

- Clearly the second and third inequalities are incompatible with the fourth, so there is in fact no solution. We need more complex networks, e.g. that combine together many simple networks, or use different activation/thresholding/transfer functions.
- It then becomes much more difficult to determine all the weights and thresholds by hand.

# Training of a perceptron

- The algorithm for training a perceptron
  1. Initialization
  2. Activation
  3. Weight learning
  4. Iteration

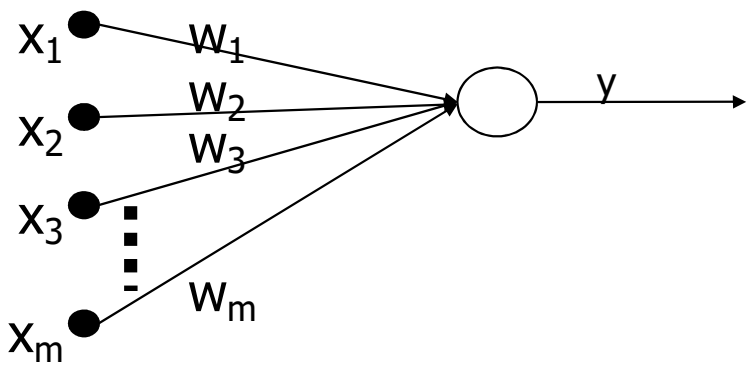
# Perceptron training

- 1) Initialization

  Set initial weights $w_1, w_2, \ldots, w_m$

  Set the threshold $\theta$ to a random number in the range [-0.5, 0.5]

  Set the learning rate $\eta$ to a positive value less than 1



# Perceptron training

- 2) Activation

Calculate the actual output at the iteration p=1

output:

$$y = \Phi\left(\sum_{i=1}^{m} x_i \cdot w_i\right)$$

Activation function:

Threshold/step function (or sign function)

output:

$$y(p) = step\left[\left(\sum_{i=1}^{m} x_i \cdot w_i\right) - \theta\right]$$

# Perceptron training

- 3) Weight training

  Update the weights of the perceptron

new weight $w_i(p+1) = w_i(p) + \Delta w_i(p)$

weight correction $\Delta w_i(p) = \eta . x_i(p) . e(p)$

error $e(p) = d(p) - y(p)$

# Perceptron training

$$y(p) = step\left[\left(\sum_{i=1}^{m} x_i \cdot w_i\right) - \theta\right]$$

# An example

- Train a perceptron to perform logical operation AND

| Epoch | Iteration | Inputs | | Initial weights | | Desired output | Actual output | Error | Final weights | |
|---|---|---|---|---|---|---|---|---|---|---|
| | p | $x_1(p)$ | $x_2(p)$ | $w_1(p)$ | $w_2(p)$ | d(p) | y(p) | e(p) | $w_1(p+1)$ | $w_2(p+1)$ |
| 1 | 1 | 0 | 0 | 0.3 | -0.1 | 0 | 0 | 0 | 0.3 | -0.1 |
| | 2 | 0 | 1 | 0.3 | -0.1 | 0 | 0 | 0 | 0.3 | -0.1 |
| | 3 | 1 | 0 | 0.3 | -0.1 | 0 | 1 | -1 | | |
| | 4 | 1 | 1 | | | | | | | |

# An example

- 1) Initialization

1 epoch has 4 possible input patterns: 00, 01, 10, 11

Initial weights are given $w_1 = 0.3$, $w_2 = -0.1$; $\theta=0.2$, $\eta=0.1$

| Epoch | Iteration | Inputs | | Initial weights | | Desired output | Actual output | Error | Final weights | |
|---|---|---|---|---|---|---|---|---|---|---|
| | p | $x_1(p)$ | $x_2(p)$ | $w_1(p)$ | $w_2(p)$ | d(p) | y(p) | e(p) | $w_1(p+1)$ | $w_2(p+1)$ |
| 1 | 1 | 0 | 0 | 0.3 | -0.1 | | | | | |
| | 2 | 0 | 1 | | | | | | | |
| | 3 | 1 | 0 | | | | | | | |
| | 4 | 1 | 1 | | | | | | | |

# An example

- 2) Activation

$$y(p) = step\left[\left(\sum_{i=1}^{m} x_i \cdot w_i\right) - \theta\right]$$

Actual output

y = step[ (0(0.3)+0(-0.1)) – 0.2] = step[-0.2] = 0

| Epoch | Iteration | Inputs | | Initial weights | | Desired output | Actual output | Error | Final weights | |
|---|---|---|---|---|---|---|---|---|---|---|
| | p | $x_1(p)$ | $x_2(p)$ | $w_1(p)$ | $w_2(p)$ | d(p) | y(p) | e(p) | $w_1(p+1)$ | $w_2(p+1)$ |
| 1 | 1 | 0 | 0 | 0.3 | -0.1 | | **0** | | | |
| | 2 | 0 | 1 | | | | | | | |
| | 3 | 1 | 0 | | | | | | | |
| | 4 | 1 | 1 | | | | | | | |

# Example

- 3) Weight training

AND operation for inputs 00 is 0

Error $e(p) = d(p) - y(p) = 0 - 0 = 0$

| Epoch | Iteration | Inputs | | Initial weights | | Desired output | Actual output | Error | Final weights | |
|---|---|---|---|---|---|---|---|---|---|---|
| | p | $x_1(p)$ | $x_2(p)$ | $w_1(p)$ | $w_2(p)$ | d(p) | y(p) | e(p) | $w_1(p+1)$ | $w_2(p+1)$ |
| 1 | 1 | 0 | 0 | 0.3 | -0.1 | 0 | 0 | 0 | | |
| | 2 | 0 | 1 | | | | | | | |
| | 3 | 1 | 0 | | | | | | | |
| | 4 | 1 | 1 | | | | | | | |

# Example

- 3) Weight training

For $w_1$: $\Delta w_1(p) = \eta . x_1(p).e(p) = 0.1(0)(0) = 0$

$w_1(p+1) = w_1(p) + \Delta w_1(p) = 0.3 + 0 = 0.3$

For $w_2$ in similar way, $w_2(p+1) = -0.1$

| Epoch | Iteration | Inputs | | Initial weights | | Desired output | Actual output | Error | Final weights | |
|---|---|---|---|---|---|---|---|---|---|---|
| | p | $x_1(p)$ | $x_2(p)$ | $w_1(p)$ | $w_2(p)$ | d(p) | y(p) | e(p) | $w_1(p+1)$ | $w_2(p+1)$ |
| 1 | 1 | 0 | 0 | 0.3 | -0.1 | 0 | 0 | 0 | **0.3** | **-0.1** |
| | 2 | 0 | 1 | | | | | | | |
| | 3 | 1 | 0 | | | | | | | |
| | 4 | 1 | 1 | | | | | | | |

# Example

- 4) Iteration

p+1 = 2

Repeat steps 2, 3, 4……until convergence

| Epoch | Iteration | Inputs | | Initial weights | | Desired output | Actual output | Error | Final weights | |
|---|---|---|---|---|---|---|---|---|---|---|
| | p | $x_1(p)$ | $x_2(p)$ | $w_1(p)$ | $w_2(p)$ | d(p) | y(p) | e(p) | $w_1(p+1)$ | $w_2(p+1)$ |
| 1 | 1 | 0 | 0 | 0.3 | -0.1 | 0 | 0 | 0 | 0.3 | -0.1 |
| | 2 | 0 | 1 | 0.3 | -0.1 | | | | | |
| | 3 | 1 | 0 | | | | | | | |
| | 4 | 1 | 1 | | | | | | | |

# Example

- Answer for Epoch 1

| Epoch | Iteration | Inputs | | Initial weights | | Desired output | Actual output | Error | Final weights | |
|---|---|---|---|---|---|---|---|---|---|---|
| | p | $x_1(p)$ | $x_2(p)$ | $w_1(p)$ | $w_2(p)$ | d(p) | y(p) | e(p) | $w_1(p+1)$ | $w_2(p+1)$ |
| 1 | 1 | 0 | 0 | 0.3 | -0.1 | 0 | 0 | 0 | 0.3 | -0.1 |
| | 2 | 0 | 1 | 0.3 | -0.1 | 0 | 0 | 0 | 0.3 | -0.1 |
| | 3 | 1 | 0 | 0.3 | -0.1 | 0 | 1 | -1 | 0.2 | -0.1 |
| | 4 | 1 | 1 | 0.2 | -0.1 | 1 | 0 | 1 | 0.3 | 0.0 |

# Example

- Answer for Epoch 2

| Epoch | Iteration | Inputs | | Initial weights | | Desired output | Actual output | Error | Final weights | |
|---|---|---|---|---|---|---|---|---|---|---|
| | p | $x_1(p)$ | $x_2(p)$ | $w_1(p)$ | $w_2(p)$ | d(p) | y(p) | e(p) | $w_1(p+1)$ | $w_2(p+1)$ |
| 2 | 5 | 0 | 0 | 0.3 | 0.0 | 0 | 0 | 0 | 0.3 | 0.0 |
| | 6 | 0 | 1 | 0.3 | 0.0 | 0 | 0 | 0 | 0.3 | 0.0 |
| | 7 | 1 | 0 | 0.3 | 0.0 | 0 | 1 | -1 | 0.2 | 0.0 |
| | 8 | 1 | 1 | 0.2 | 0.0 | 1 | 1 | 0 | 0.2 | 0.0 |

# Example

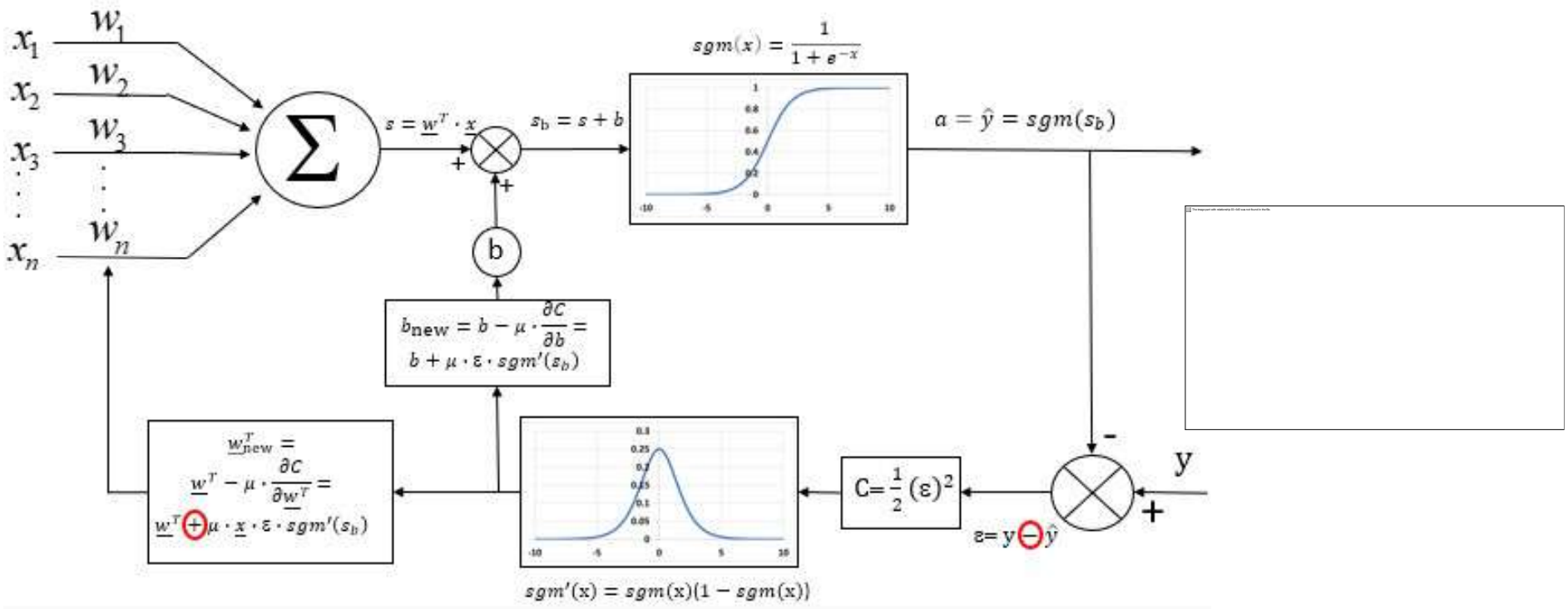
- Answer for Epoch 5

| Epoch | Iteration | Inputs | | Initial weights | | Desired output | Actual output | Error | Final weights | |
|---|---|---|---|---|---|---|---|---|---|---|
| | p | $x_1(p)$ | $x_2(p)$ | $w_1(p)$ | $w_2(p)$ | d(p) | y(p) | e(p) | $w_1(p+1)$ | $w_2(p+1)$ |
| 5 | 17 | 0 | 0 | 0.1 | 0.1 | 0 | 0 | 0 | 0.1 | 0.1 |
| | 18 | 0 | 1 | 0.1 | 0.1 | 0 | 0 | 0 | 0.1 | 0.1 |
| | 19 | 1 | 0 | 0.1 | 0.1 | 0 | 0 | 0 | 0.1 | 0.1 |
| | 20 | 1 | 1 | 0.1 | 0.1 | 1 | 1 | 0 | 0.1 | 0.1 |

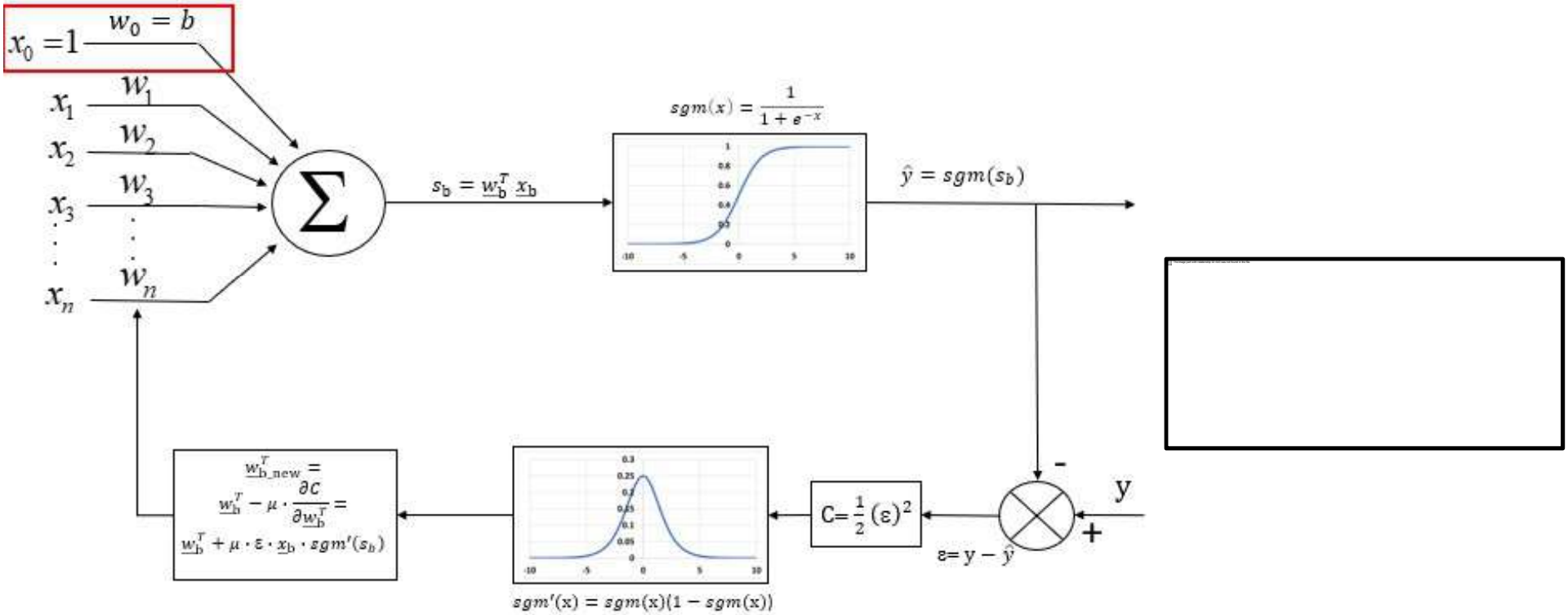
# Artificial neural networks

- Artificial neural network is a data processing system consisting of a large number of simple, highly interconnected processing elements (artificial neurons) in an architecture inspired by the structure of the cerebral cortex of the brain to yield a network with rather complex behavior.
- In ANN, neurons (processing elements) are organized into a sequence of layers with full or partial connection between the layers.
- For simplicity, usually the neurons in the same layer have the same activation function.

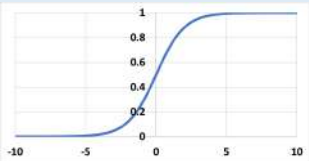
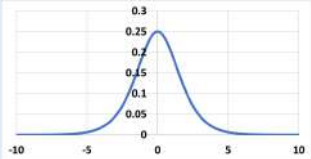
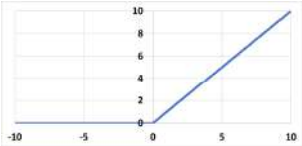
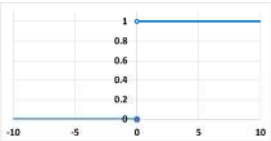
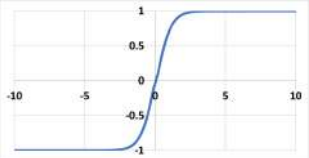
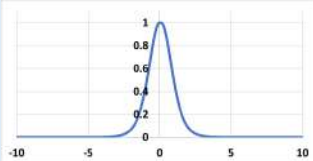
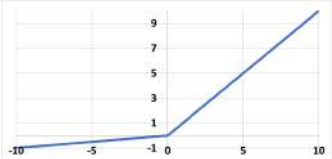
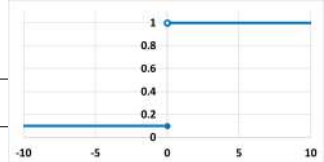
# Neuron


$x_1$ $w_1$
$x_2$ $w_2$
$x_3$ $w_3$
$x_n$ $w_n$
$\Sigma$
$s = \underline{w}^T \cdot \underline{x}$
$s_b = s + b$
$sgm(x) = \frac{1}{1+e^{-x}}$
$a = \hat{y} = sgm(s_b)$
b
$b_{new} = b - \mu \cdot \frac{\partial C}{\partial b} = b + \mu \cdot \varepsilon \cdot sgm'(s_b)$
$\underline{w}^T_{new} = \underline{w}^T - \mu \cdot \frac{\partial C}{\partial \underline{w}^T} = \underline{w}^T + \mu \cdot \underline{x} \cdot \varepsilon \cdot sgm'(s_b)$
$C = \frac{1}{2}(\varepsilon)^2$
$\varepsilon = y - \hat{y}$
y
$sgm'(x) = sgm(x)(1 - sgm(x))$

# Neuron


$x_0 = 1$
$w_0 = b$
$x_1$ $w_1$
$x_2$ $w_2$
$x_3$ $w_3$
$x_n$ $w_n$
$\sum$
$s_b = \underline{w}_b^T \underline{x}_b$
$sgm(x) = \frac{1}{1+e^{-x}}$
$\hat{y} = sgm(s_b)$
y
-
+
$\varepsilon = y - \hat{y}$
$C = \frac{1}{2}(\varepsilon)^2$
$sgm'(x) = sgm(x)(1 - sgm(x))$
$\underline{w}_{b_new}^T = \underline{w}_b^T - \mu \cdot \frac{\partial C}{\partial \underline{w}_b^T} = \underline{w}_b^T + \mu \cdot \varepsilon \cdot \underline{x}_b \cdot sgm'(s_b)$

# Activation function

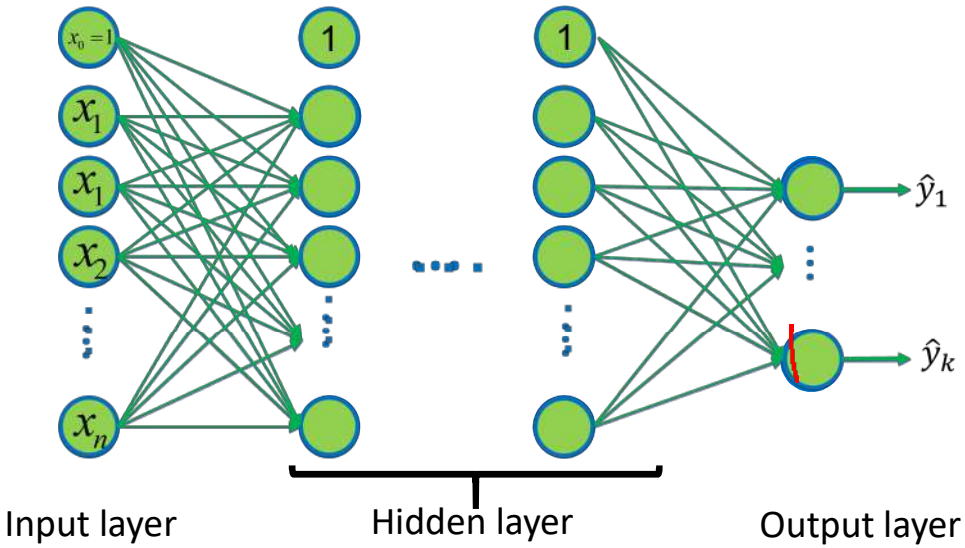
| Equation | Graph | Derivative | Graph |
|---|---|---|---|
| Sigmoid |  | |  |
| Rectifier Linear Unit (ReLU) |  | |  |
| Tanh |  | |  |
| Leaky ReLU |  | |  |

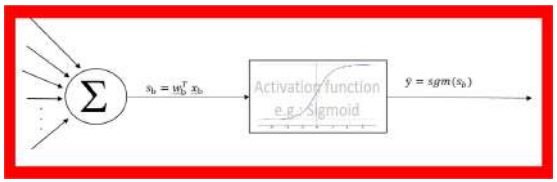
# Artificial neural networks

- In terms of density, 2 important groups can be distinguished in a first approximation
  - **Fully Connected Artificial Neural Networks**
    All neurons in a layer are connected to all neurons in the next layer.
    - **Advantage:**
      - simple architecture
      - maximizes the potential of the network size
    - **Disadvantage: (for large networks)**
      - requires a lot of memory
      - requires a lot of computation
  - **Partially coupled artificial neural networks**
    All neurons in a layer are connected to only a few neurons in the next layer.

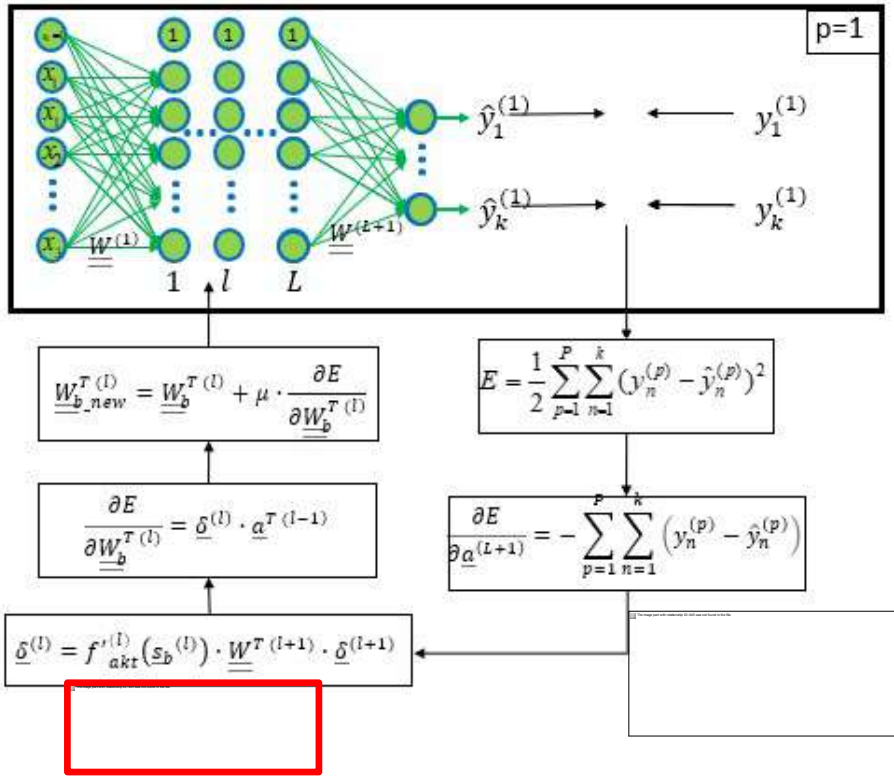
# Fully connected ANN

- The input to neural networks is determined by the available data and the preparation of the data.
- Its output is determined by the target task.
- The depth of its hidden layer is determined by its decision complexity.
- Its width is determined by the amount of data needed for decisions.
- In general, the appropriate dimensions of the network cannot be determined in advance.



# Training of fully connected ANN



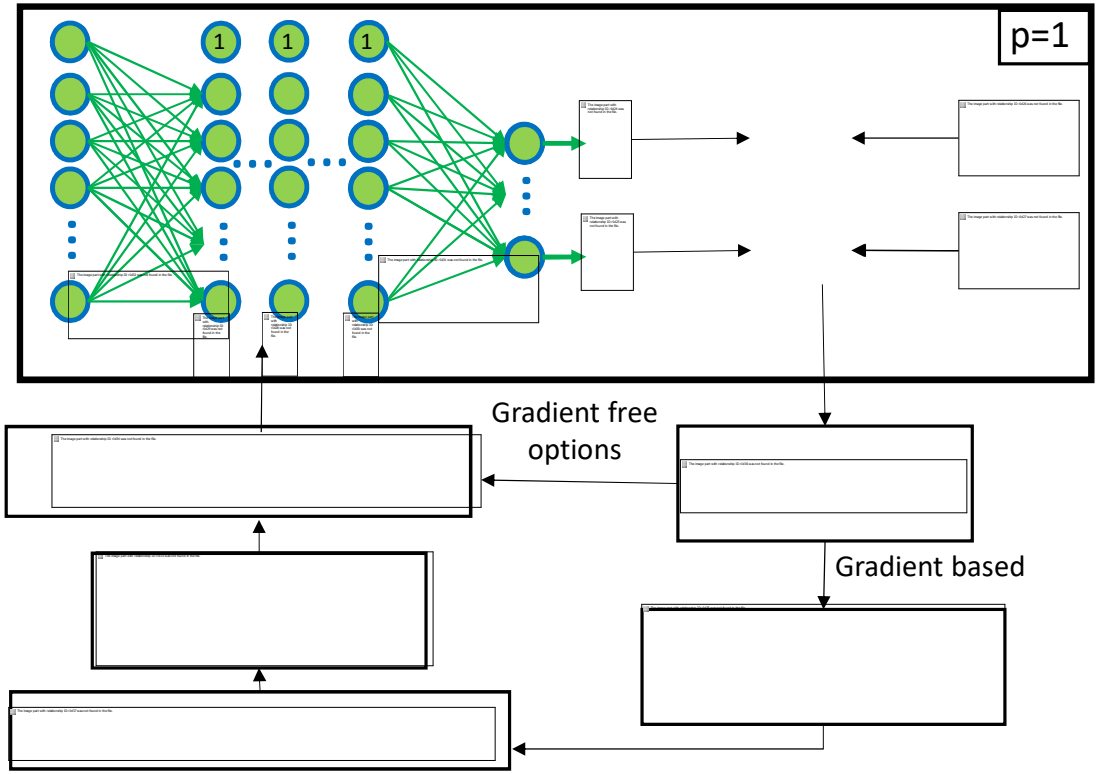
# Training of fully connected ANN

# Fully connected ANN

- The followings can be chosen when teaching a neural network:
  - Other error functions (E)
  - Other optimization methods

The optimization methods can be:

- Gradient-based
  - Other methods can be based on gradient descent, Gauss-Newton, Quasi-Newton, Levenberg-Marquardt, ADAM, ...
- Gradient-free
  - Random search, Evolutionary algorithms (later lecture)...
- Miscellaneous
  - Memetic algorithms (later lecture)...



# Artificial Neural Networks

## Partially coupled artificial neural networks

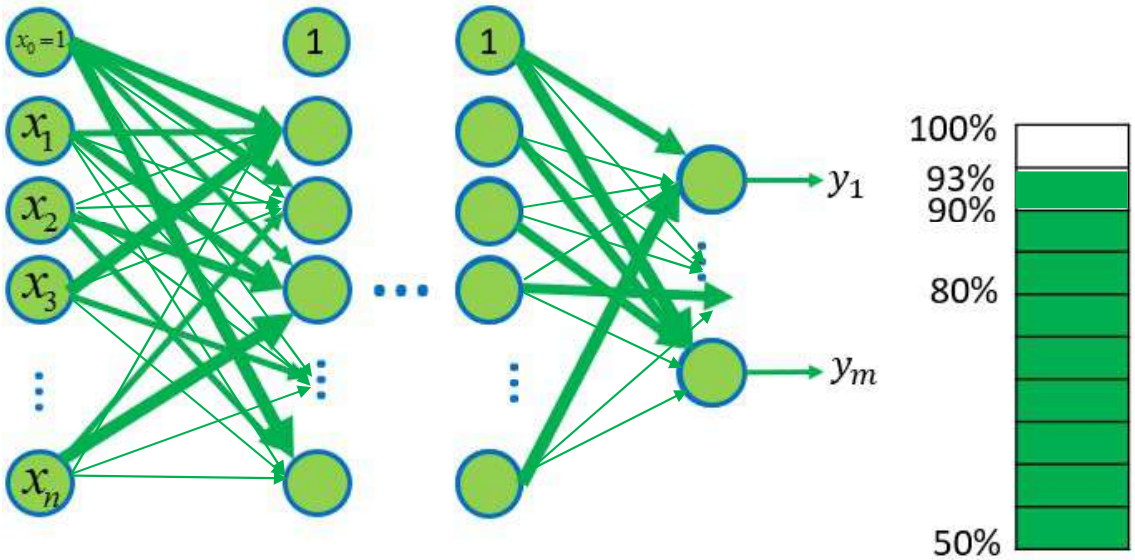
All neurons in a layer are connected to only a few neurons in the next layer.

- Advantage:
  - lower memory requirements
  - less computational effort
- Disadvantage:
  - it is usually not known in advance which weights should be omitted

- Can be generated e.g.:
  - From fully coupled artificial neural networks by dropping less significant weights
  - By engineering considerations (e.g.: stronger association of nearby words/pixels)
  - Neural network growth (Neuroevolution)

# Omission of less significant weights

- After training, only a minimal reduction in the accuracy of the neural network is achieved by dropping the weights of minor importance, in exchange for a significant reduction in the number of parameters.
- In some cases (e.g., extreme learning machine, more typical for wide structures), a partially coupled ANN can achieve competitive learning speed by adding and subtracting weights from the initial state even using CPU.
- In between dropping weights, it may be worthwhile to retrain the model, in which case other paths may be strengthened to take the role of the dropped links, in which case the final accuracy is less degraded.



# Neuroevolution

The essence of the method is to optimize the structure of the network in addition to the parameters.

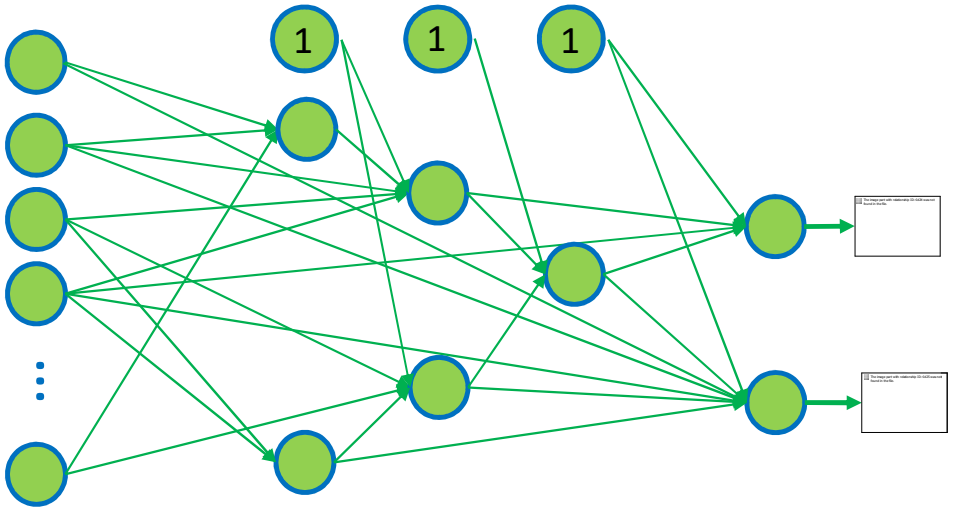
This is true automatic machine learning for neural networks.

Advantage:

- Optimal structure generation
- Automatic operation

Disadvantage:

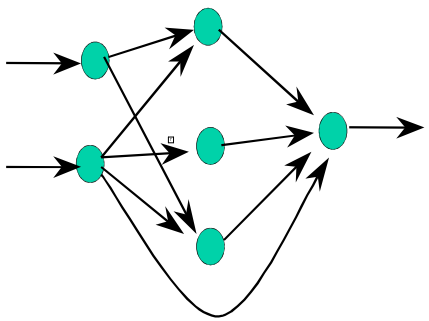
- Resource intensive



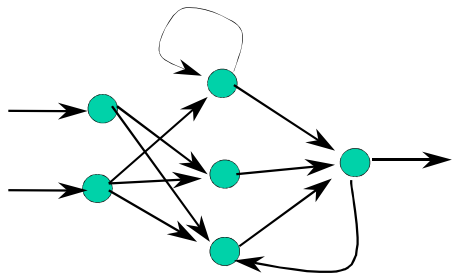

Simplest case is when individual neurons and the connections between them are modified by an evolutionary algorithm.

Under development (explorable): optimization through multi-level mapping that can use its own substructures. (further theory: genetic programming) Here too, it is worth working with structured input data.

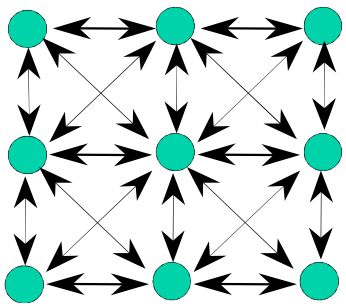
# Grouping by signal flow



Feedforward



Recurrent



Cellular

**Feedforward neural networks**
Use of:
- fast processing of time-independent data,
- non real-time accurate processing of time-dependent data

**Recurrent neural networks**
Use of:
- fast processing of time-dependent data

A **cellular neural network** is a network where there is a connection only between neighboring units.
Usage:
- mainly image processing

# Grouping of certain layers

Different layer types have been developed for each (sub)task

- 1 layer usually contains the same type of neurons

**Evaluation**

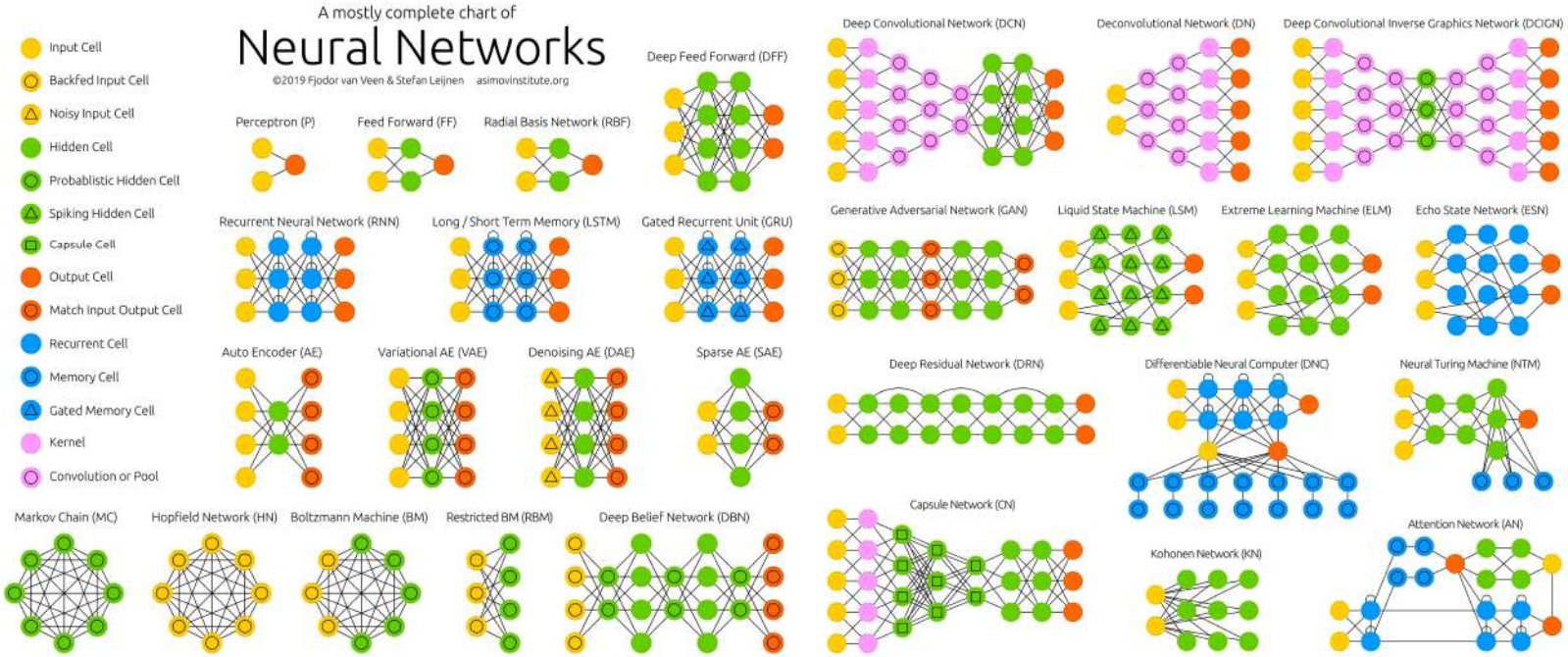
- Input
- Output
- Fully coupled layer
- Partially coupled layer
- Activation layer
- Link layers (to link more than 2 layers)
- Convolutional
- Pooling layer
- Reconnected layer
- Memory cell layers
- Highlighting type (ROI, Attention)

**Training specific**

- Normalization layers
  - Batch normalization layer (on sample batch)
  - Batch normalization layer (on individual samples)
- Regularization layers
  - Noise generation layers (random noise addition)
  - Dropout layer: Punctuated (random neurons) or Contiguous (random but contiguous neurons)

# An overview of NNs

Different neural networks have been developed for different tasks




Source

ELTE | FACULTY OF INFORMATICS

# Swarm intelligence

**Introduction to Machine Learning – Lecture 9**
Computer Science BSc Course, ELTE Faculty of Informatics

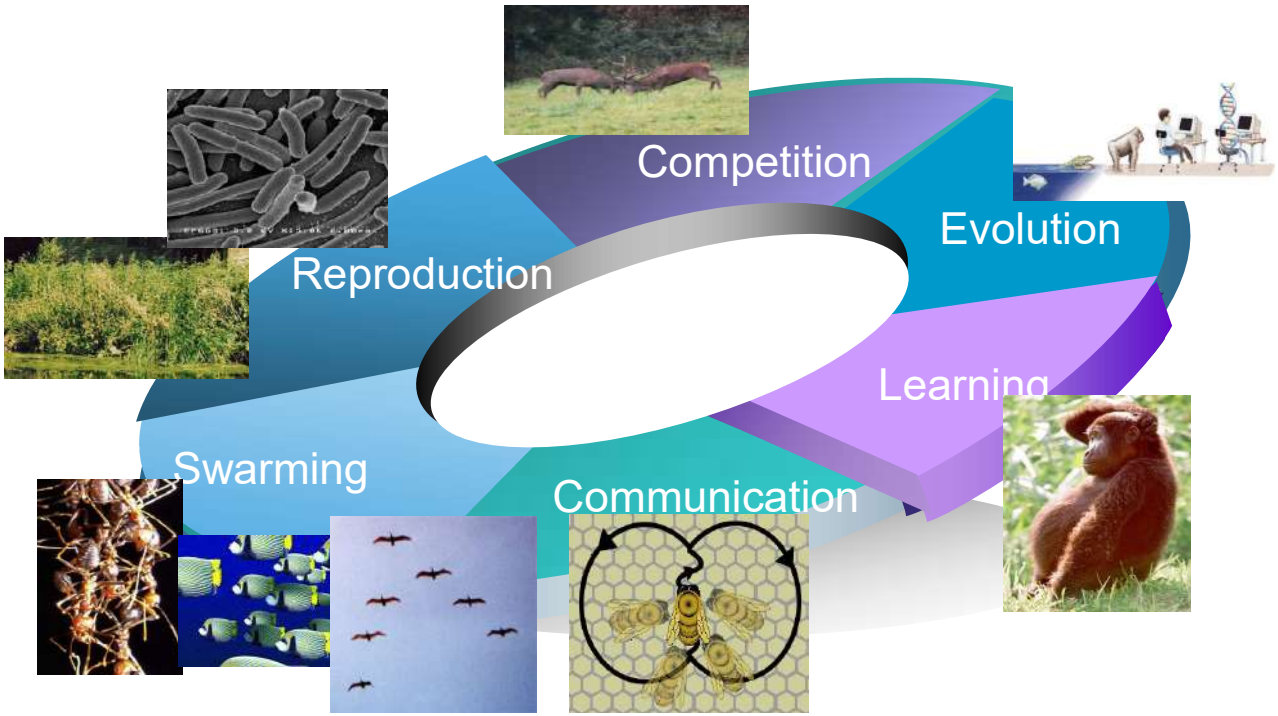
János Botzheim

Associate Professor
Department of Artificial Intelligence
botzheim@inf.elte.hu

ELTE | IK
DEPARTMENT OF ARTIFICIAL INTELLIGENCE

# Introduction, motivations

# Elements of intelligence in biological systems



# Swarm intelligence

- A collective system capable of performing complex tasks in a dynamic and changing environment without any external control or central coordination
- Capable of achieving a collective performance that cannot normally be achieved by the organism alone
- On this basis, we can build a natural model that is suitable for distributed problem solving

# Swarm intelligence

# Swarm intelligence

# Swarm intelligence

# Swarm intelligence

# Particle Swarm Optimization

# Particle Swarm Optimization (PSO)

- PSO applies the concept of social interaction to problem solving
- It uses a number of so-called particles that move in swarms in the search space in search of the best solution
- It treats each particle as a spatial point, which adjusts its "flight" based on its own flight experience as well as the flight experience of other particles
- James Kennedy and Russell Eberhart, 1995

# Particle flight model

- **pbest**: the best solution achieved by the particle so far
- **gbest**: best solution achieved by the whole swarm
- The principle of PSO is that individual particles accelerate towards the **pbest** and **gbest** sites, each time with a randomly weighted acceleration

ELTE | FACULTY OF INFORMATICS

ELTE | FACULTY OF INFORMATICS

ELTE | FACULTY OF INFORMATICS

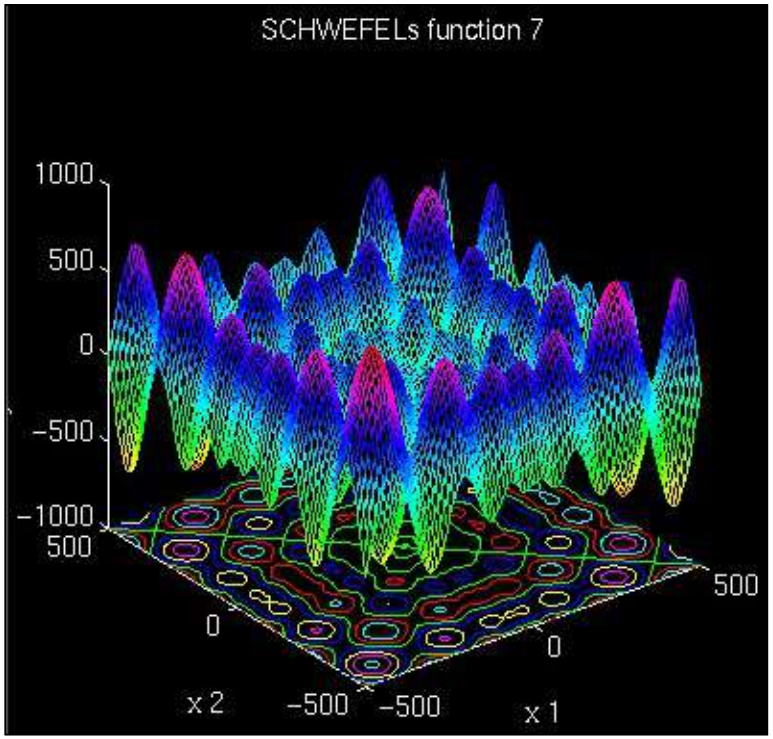
# PSO algorithm

```
For each particle
   Initialize particle
END

Do
  For each particle
      Calculate fitness value
      If the fitness value is better than the best fitness value (pbest) in history
         set current value as the new pbest
  End

  Choose the particle with the best fitness value of all the particles as the gbest
  For each particle
      Calculate particle velocity according equation (*)
      Update particle position according equation (**)
  End
While maximum iterations or minimum error criteria is not attained
```

(*)

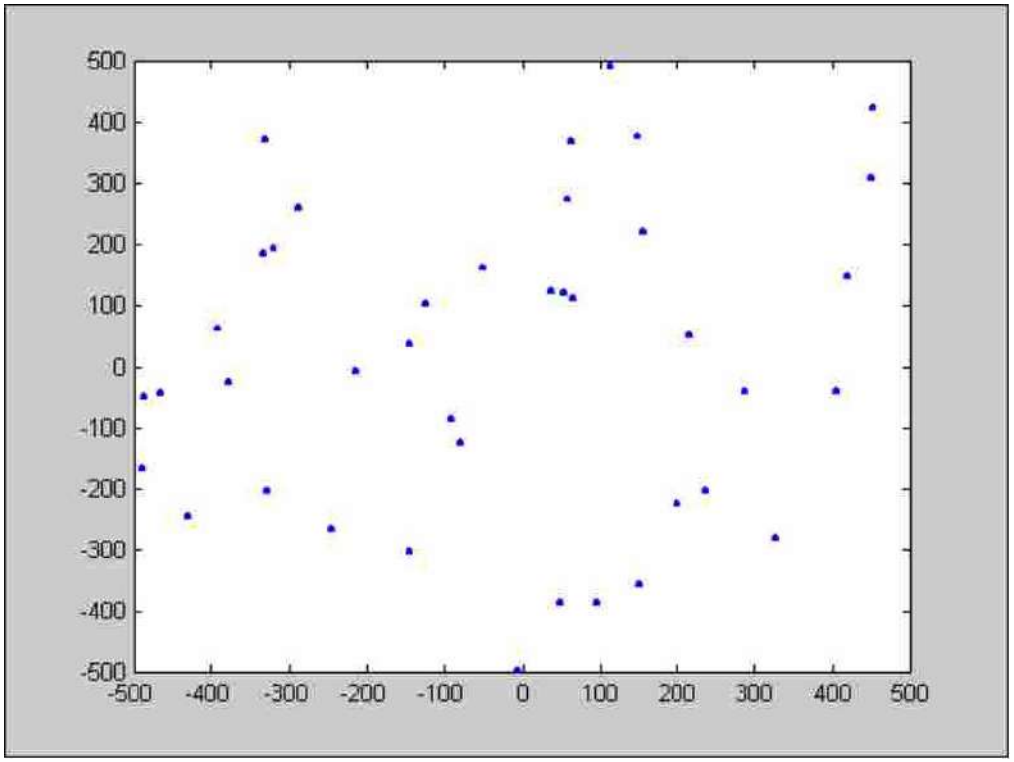
(**)

# PSO example

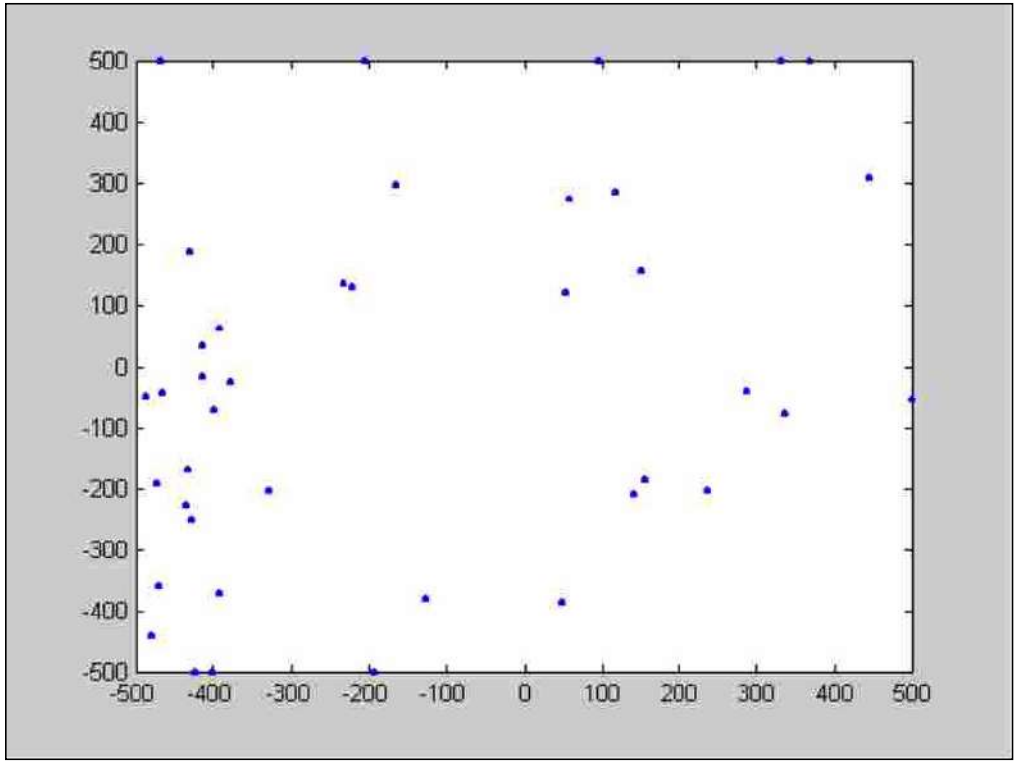



Global minimum:

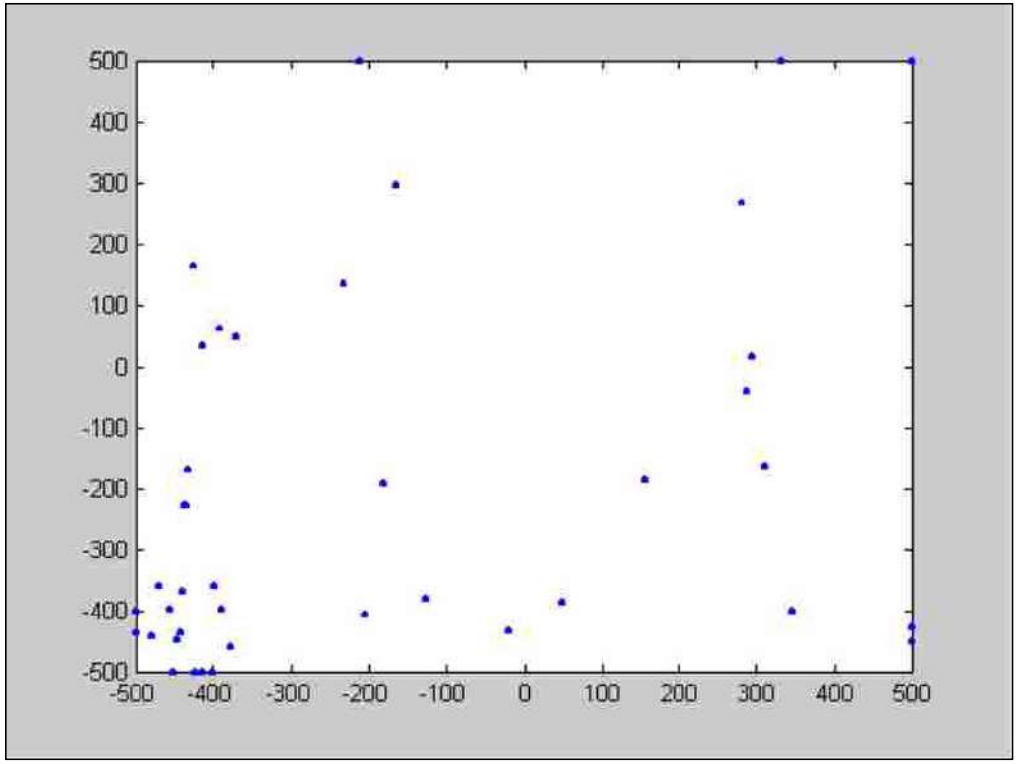
# PSO example - Initialization



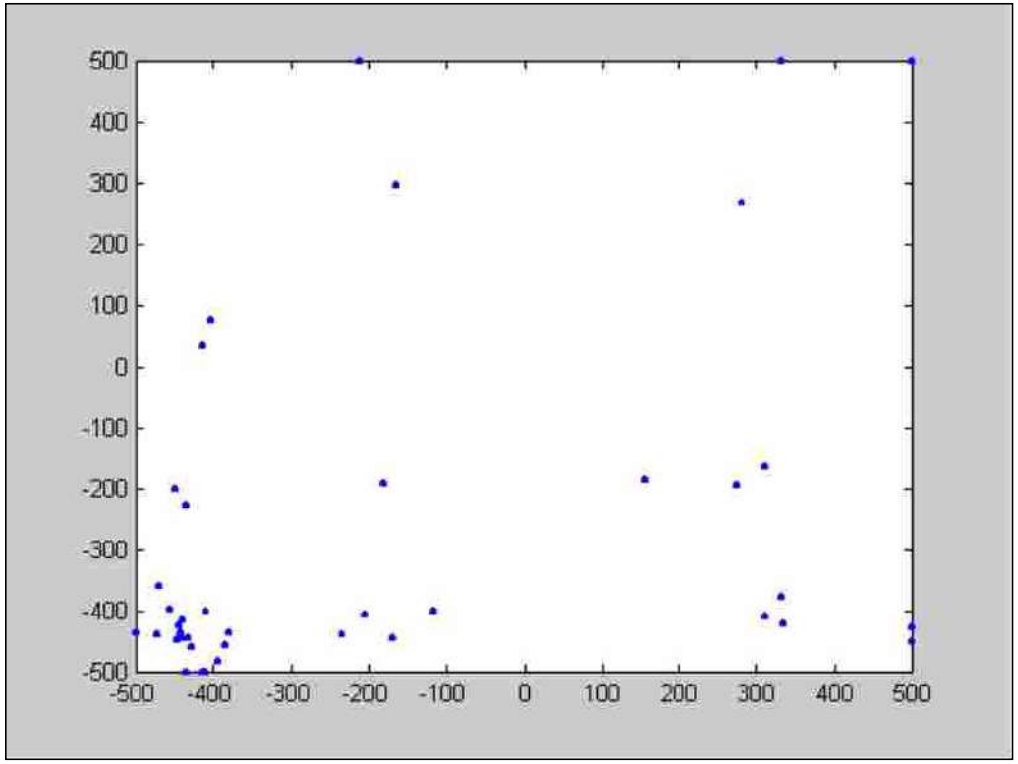
# PSO example – after 5 generations



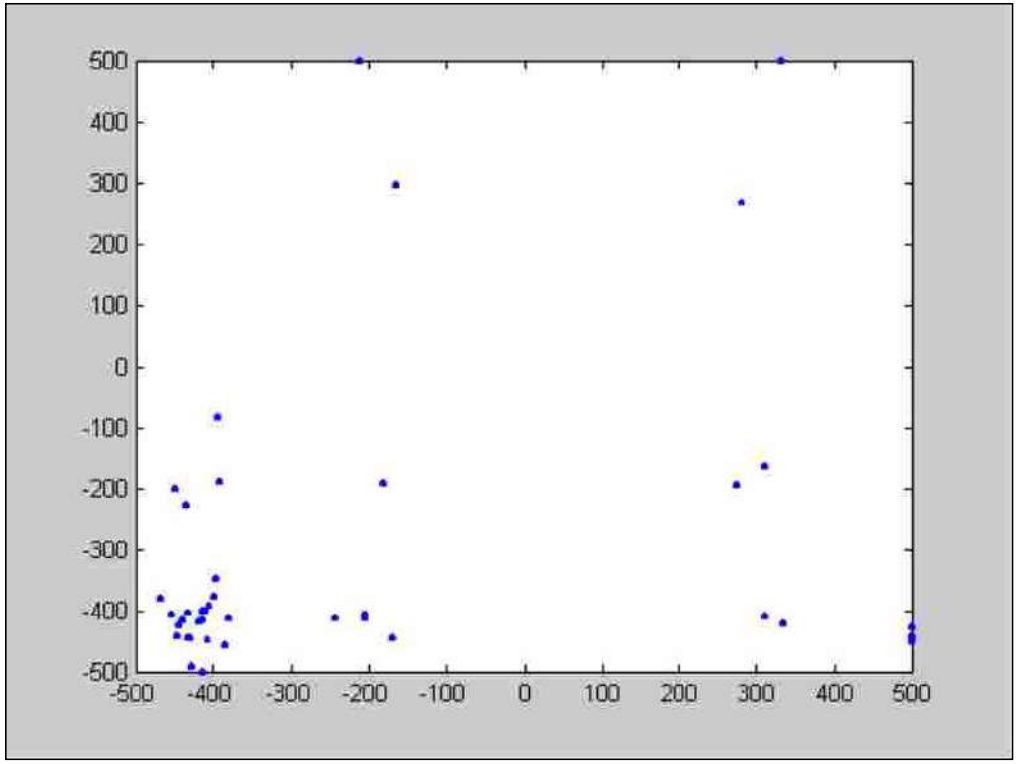
# PSO example – after 10 generations



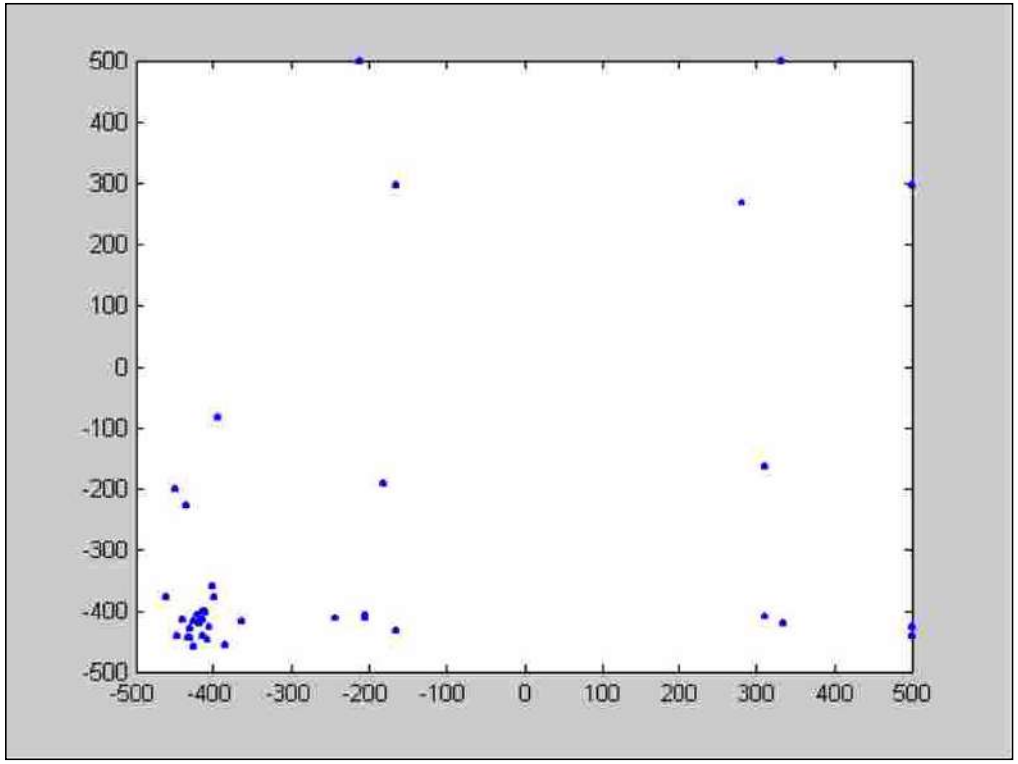
# PSO example – after 15 generations



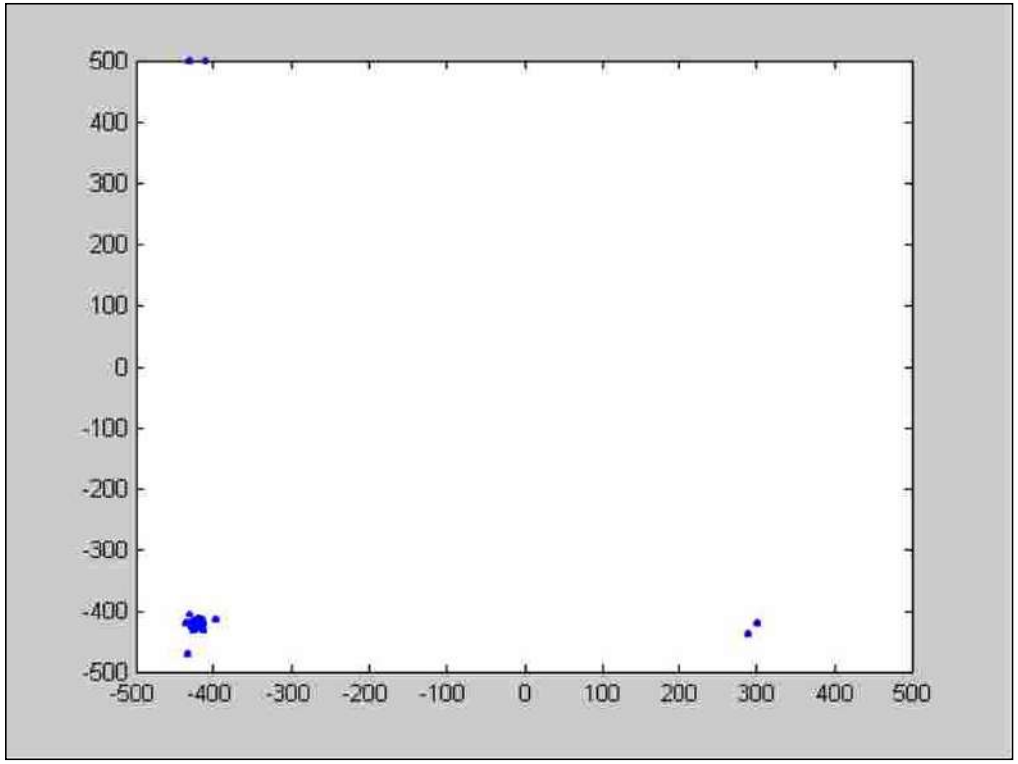
# PSO example – after 20 generations



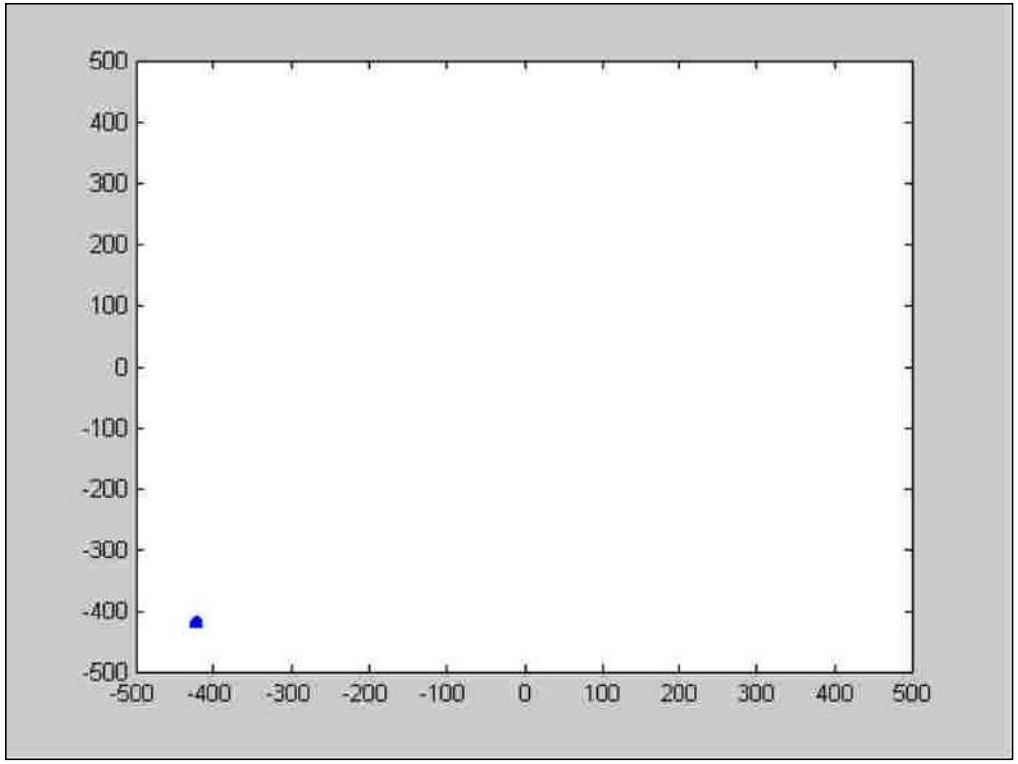
# PSO example – after 25 generations



# PSO example – after 100 generations

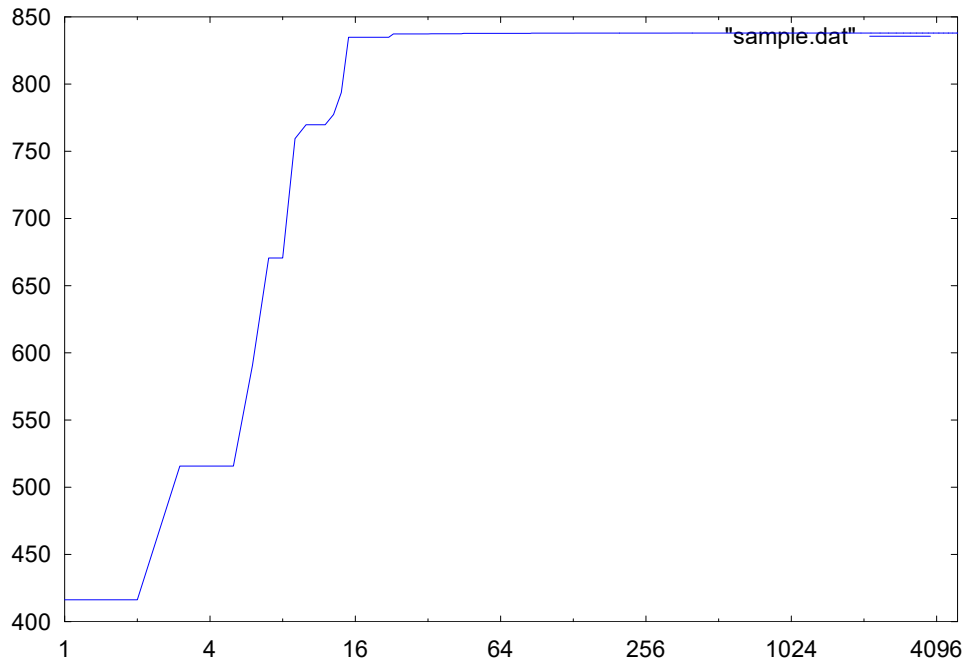


# PSO example – after 500 generations



# PSO example - result

| iteration | gBest |
|---|---|
| 0 | 416.245599 |
| 5 | 515.748796 |
| 10 | 759.404006 |
| 15 | 793.732019 |
| 20 | 834.813763 |
| 100 | 837.911535 |
| 5000 | 837.965771 |
| optimum | 837.9658 |



# Simplified Swarm Optimization (SSO)

# SSO

$$x_{ij}^{t+1} = \begin{cases} x_{ij}^{t}, & \text{if } \rho \in [0, C_w = c_w) \\ p_{ij}, & \text{if } \rho \in [C_w, C_p = C_w + c_p) \\ g_j, & \text{if } \rho \in [C_p, C_g = C_p + c_g) \\ x, & \text{if } \rho \in [C_g, 1) \end{cases}$$

# Firefly Algorithm

# About Fireflies – General

- One of the family of insects
- Live in tropical environment
- Have wings
- Produce «cold light» chemically
- Yellow, green, pale-red lights
- Their larvae called glowworm
- ~2000 species
- Flightless females

# About Fireflies - Video

# About Fireflies - Behavior

- Their purpose of flashing:
  - Attract mating partners (communication)
  - Attract potential prey
  - Protective warning mechanism

- They have unique flashing pattern
- In some species, females can mimic «mating pattern» to hunt other species

- They have limited light intensity

# The Algorithm

- Like Particle Swarm Optimization
- Inspired by the behavior of fireflies
- Developer of the algorithm is Dr. Xin-She Yang
- Three main assumptions:
  1. All fireflies are unisex
  2. Attractiveness $\propto$ Brigtness & Attractiveness $\propto$ 1/Distance
  3. Brightness is determined by objective function

# Formulas - Attractiveness

# Formulas - Distance

- In our case, *d* is going to be the Euclidean distance

$$d_{i,j} = \text{Distance}(\mathbf{x}^i, \mathbf{x}^j) = \sqrt{\sum_{k=1}^{n} (x_k^i - x_k^j)^2}$$

# Formulas - Movement

$$
\begin{aligned}
\mathbf{x}^i &= \mathbf{x}^i + \text{Attractiveness}(I_0, \gamma, \text{Distance}(\mathbf{x}^i, \mathbf{x}^j)) \cdot (\mathbf{x}^j - \mathbf{x}^i) + \alpha \cdot (Random() - \frac{1}{2}) \\
&= \mathbf{x}^i + \text{Attractiveness}(I_0, \gamma, d_{i,j}) \cdot (\mathbf{x}^j - \mathbf{x}^i) + \alpha \cdot (Random() - \frac{1}{2}) \\
&= \mathbf{x}^i + I_0 e^{-\gamma d_{i,j}^2} \cdot (\mathbf{x}^j - \mathbf{x}^i) + \alpha \cdot (Random() - \frac{1}{2}) \\
&= \mathbf{x}^i + \beta \cdot (\mathbf{x}^j - \mathbf{x}^i) + \alpha \cdot (Random() - \frac{1}{2}) \\
&= (1 - \beta) \cdot \mathbf{x}^i + \beta \cdot \mathbf{x}^j + \alpha \cdot (Random() - \frac{1}{2}),
\end{aligned}
$$

# Special Cases

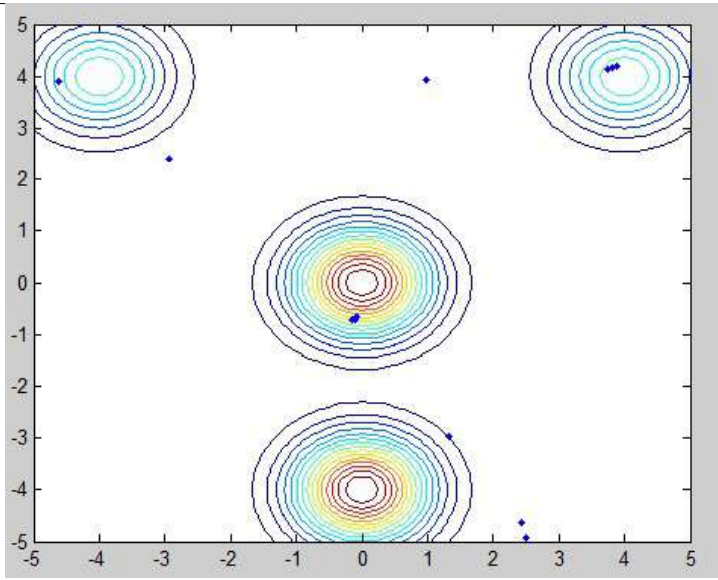
# Pseudo Code

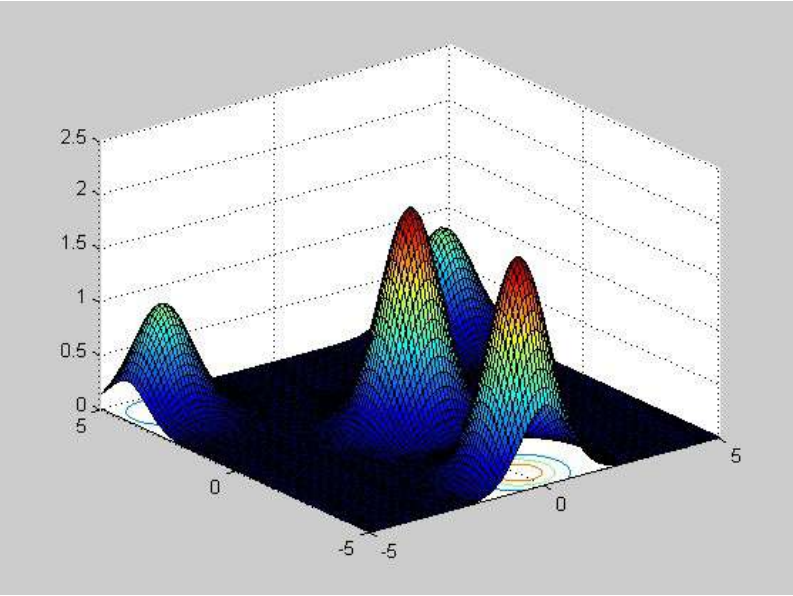
```
Objective function f(x),        x = (x_1, ..., x_d)^T
Generate initial population of fireflies x_i (i = 1, 2, ..., n)
Light intensity I_i at x_i is determined by f(x_i)
Define light absorption coefficient γ
while (t < MaxGeneration)
for i = 1 : n all n fireflies
   for j = 1 : i all n fireflies
         if (I_j > I_i), Move firefly i towards j in d-dimension; end if
         Attractiveness varies with distance r via exp[−γr]
         Evaluate new solutions and update light intensity
   end for j
end for i
Rank the fireflies and find the current best
end while
Postprocess results and visualization
```

# Applications

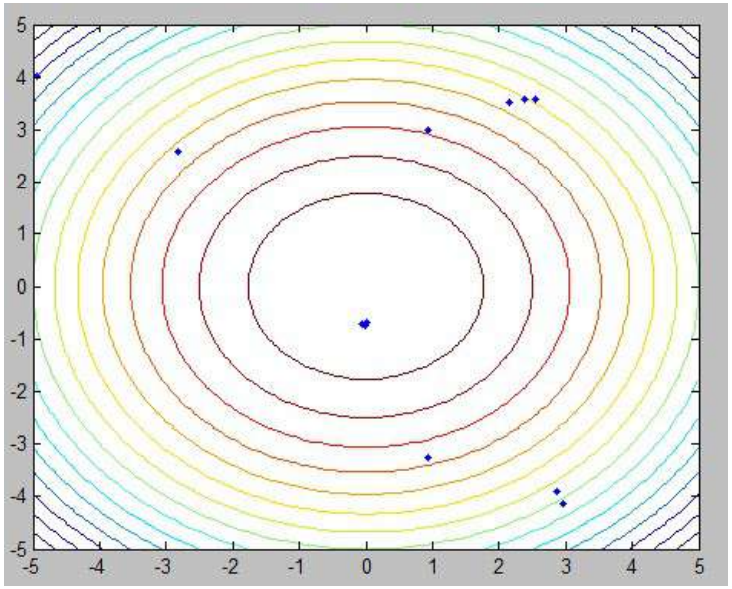
- Digital Image Compression and Image Processing
- Feature Selection and Fault Detection
- Antenna Design
- Structural Design
- Scheduling
- Semantic Web Composition
- Chemical Phase Equilibrium
- Clustering
- Dynamic Problems
- Rigid Image Registration Problems

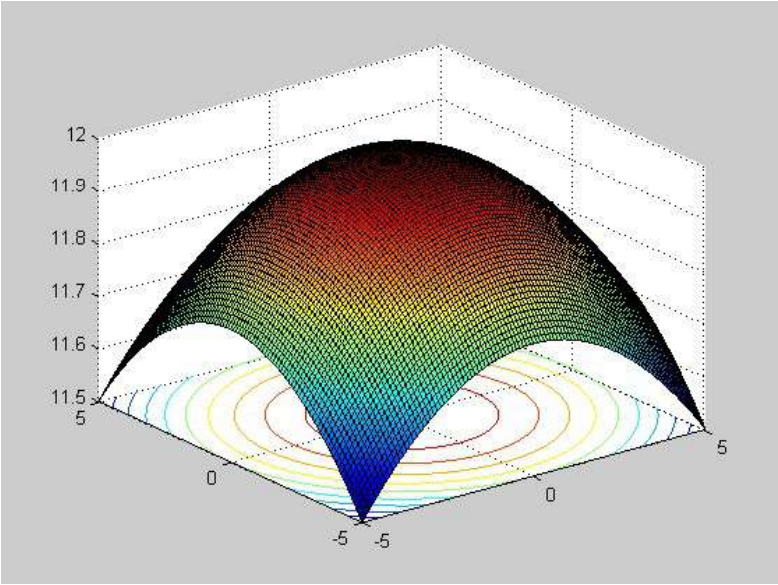
# Fireflies in Use



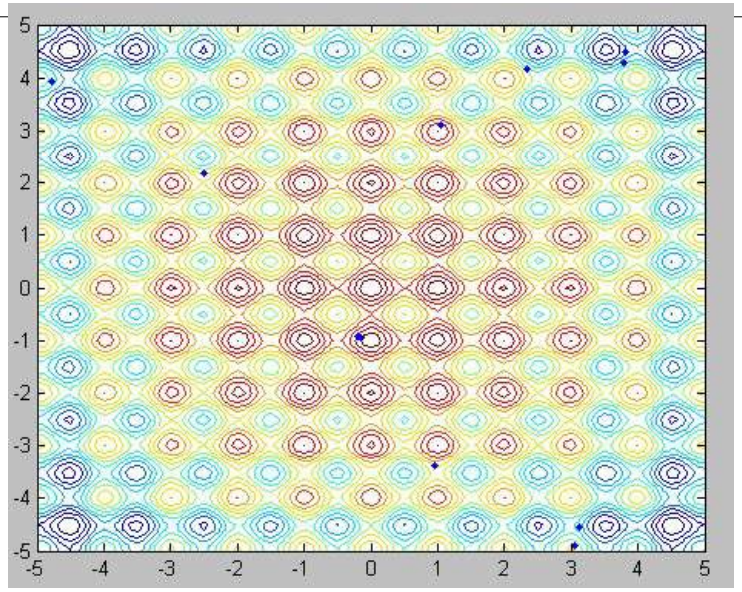



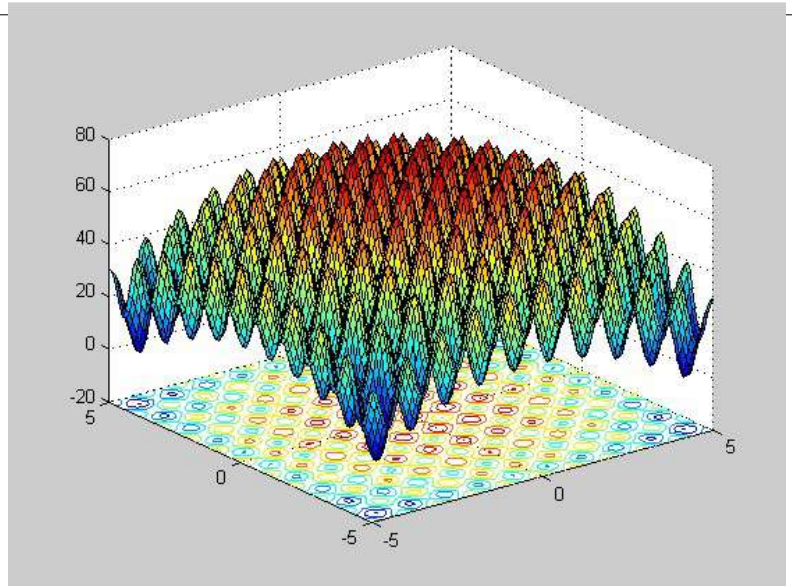
# Fireflies in Use



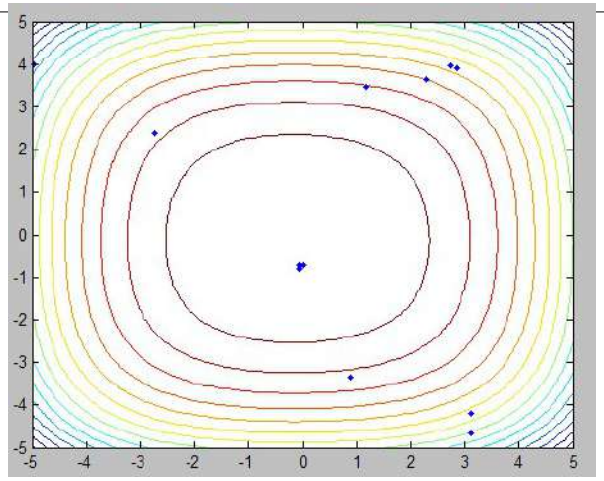



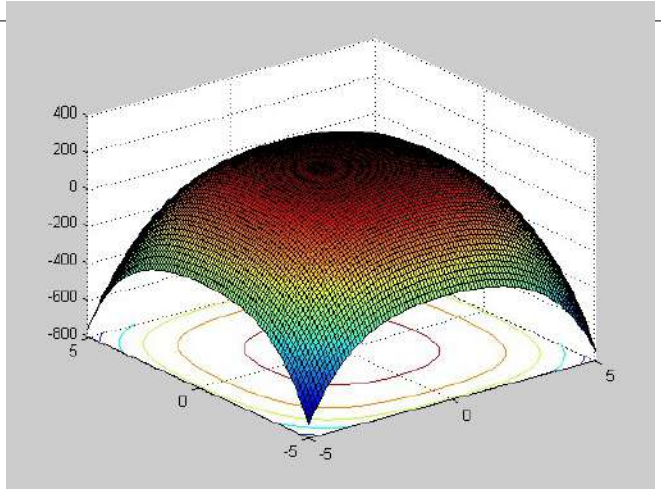
# Fireflies in Use



# Fireflies in Use



# Comparison with PSO

| Function | N | PSO | Firefly Algorithm |
|---|---|---|---|
| Four-peak | 15 | 1,5356 | 1,4840 |
| | 20 | 2,0135 | 1,9326 |
| | 25 | 2,4959 | 2,3652 |
| Parabolic | 15 | 1,5482 | 1,5039 |
| | 20 | 2,0884 | 1,9296 |
| | 25 | 2,6466 | 2,3534 |
| Rastrigin | 15 | 9,6761 | 9,5298 |
| | 20 | 12,6412 | 12,5404 |
| | 25 | 15,6878 | 15,5457 |
| Styblinski | 15 | 1,6444 | 1,5478 |
| | 20 | 2,1504 | 2,0725 |
| | 25 | 2,6144 | 2,5323 |

# Performance Comparison

| | Genetic Algorithm | Particle Swarm | Firefly Algorithm |
|---|---|---|---|
| Michalewicz | 95% | 98% | 99% |
| Rosenbrock | 90% | 98% | 99% |
| De Jong | 100% | 100% | 100% |
| Schwefel | 95% | 97% | 100% |
| Ackley | 90% | 92% | 100% |
| Rastrigin | 77% | 90% | 100% |
| Easom | 92% | 90% | 100% |
| Griewank | 90% | 92% | 100% |
| Shubert (18 min) | 89% | 92% | 100% |
| Yang | 83% | 90% | 100% |

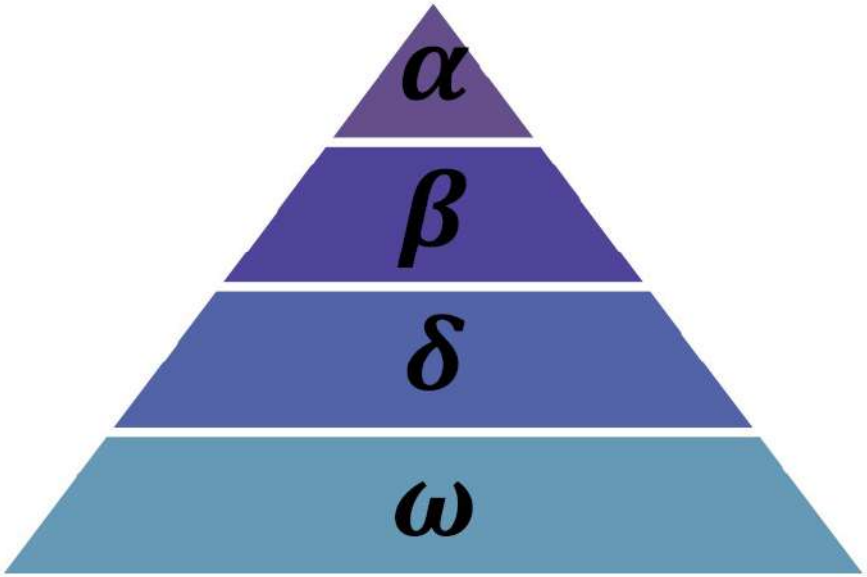
# Grey Wolf Optimizer

# Grey Wolf Optimizer

- Grey wolf optimizer (GWO) is a population based meta-heuristics algorithm simulating the leadership hierarchy and hunting mechanism of grey wolves in nature
- Grey wolves are considered as apex predators, which means they are at the top of the food chain.
- Grey wolves prefer to live in a groups (packs), each group contains 5-12 members on average.
- All the members in the group have a very strict social dominant hierarchy as shown in the following figure.

# Hierarchy of grey wolf

- The social hierarchy consists of four levels as follow.
- Dominance decreases from top down
- The first level is called Alpha ($\alpha$). The alpha wolves are the leaders of the pack and they are a male and a female.



# Hierarchy of grey wolf

- Alpha is responsible for making decisions about hunting, time to walk, sleeping place and so on.
- The pack members have to dictate the alpha decisions and they acknowledge the alpha by holding their tails down.
- The alpha wolf is considered the dominant wolf in the pack and all his/her orders should be followed by the pack members.

# Hierarchy of grey wolf

- The second level is called Beta ($\beta$).
- The betas are subordinate wolves, which help the alpha in decision making.
- The beta wolf can be either male or female and it considers the best candidate to be the alpha when the alpha passes away or becomes very old.
- The beta reinforces the alpha's commands throughout the pack and gives the feedback to alpha.

# Hierarchy of grey wolf

- The third level is called Delta ($\delta$)
- The delta wolves are not alpha or beta wolves and they are called subordinates.
- Delta wolves have to submit to the alpha and beta but they dominate the omega (the lowest level in wolves social hierarchy).
- There are different categories of delta as follows:
  - Scouts: The scout wolves are responsible for watching the boundaries of the territory and warning the pack in case of any danger.
  - Sentinels: The sentinel wolves are responsible for protecting the pack.

# Hierarchy of grey wolf

- Elders: The elder wolves are the experienced wolves who used to be alpha or beta.
- Hunters: The hunter wolves are responsible for helping the alpha and beta wolves in hunting and providing food for the pack.
- Caretakers: The caretakers are responsible for caring for the ill, weak and wounded wolves in the pack.

- The fourth (lowest) level is called Omega ($\omega$)
- The omega wolves are considered the scapegoat in the pack, they have to submit to all the other dominant wolves.
- They may seem not important individuals in the pack and they are the last allowed wolves to eat.
- The whole pack is fighting in the case of losing the omega.

# Hierarchy of grey wolf

- In the grey wolf optimizer (GWO), we consider the fittest solution as the alpha, and the second and the third fittest solutions are named beta and delta, respectively.
- The rest of the solutions are considered omega
- In GWO algorithm, the hunting is guided by $\alpha$, $\beta$, and $\delta$
- The $\omega$ solutions follow these three wolves.

# Grey wolf hunting phases

- The main phases of grey wolf hunting are as follows:
  - Tracking, chasing, and approaching the prey. (A)
  - Pursuing, encircling, and harassing the prey until it stops moving. (B,C,D)
  - Attack towards the prey. (E)

# Grey wolf encircling prey

- During the hunting, the grey wolves encircle prey.
- The mathematical model of the encircling behavior is presented in the following equations:

$$\vec{D} = |\vec{C} \cdot \vec{X}_p(t) - \vec{X}(t)|$$

$$\vec{X}(t+1) = \vec{X}_p(t) - \vec{A} \cdot \vec{D}$$

- where $t$ is the current iteration, $A$ and $C$ are coefficient vectors, $X_p$ is the position vector of the prey, and $X$ indicates the position vector of a grey wolf.
- The vectors $A$ and $C$ are calculated as follows:

$$\vec{A} = 2\vec{a} \cdot \vec{r}_1 - \vec{a}$$

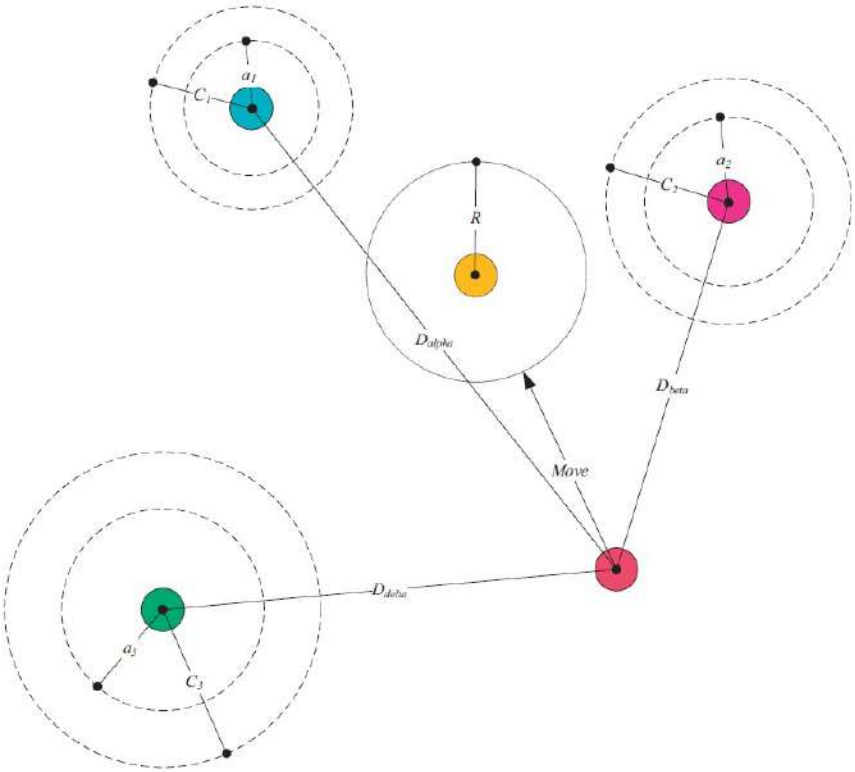
$$\vec{C} = 2 \cdot \vec{r}_2$$

- where components of $a$ are linearly decreased from 2 to 0 over the course of iterations and $r_1$, $r_2$ are random vectors in [0, 1]

# Grey wolf hunting

- The hunting operation is usually guided by the alpha.
- The beta  and delta might participate in hunting occasionally.
- In the mathematical model of hunting behavior of grey wolves, we assumed the alpha, beta, and delta have better knowledge about the potential location of prey.
- The first three best solutions are saved and the other agent are obliged to update their positions according to the position of the best search agents as shown in the following equations.

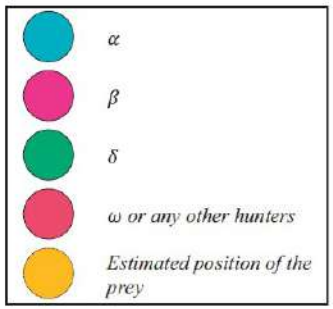
# Grey wolf hunting (position update)




$$\vec{D}_\alpha = |\vec{C}_1 \cdot \vec{X}_\alpha - \vec{X}|, \vec{D}_\beta = |\vec{C}_2 \cdot \vec{X}_\beta - \vec{X}|, \vec{D}_\delta = |\vec{C}_3 \cdot \vec{X}_\delta - \vec{X}|$$

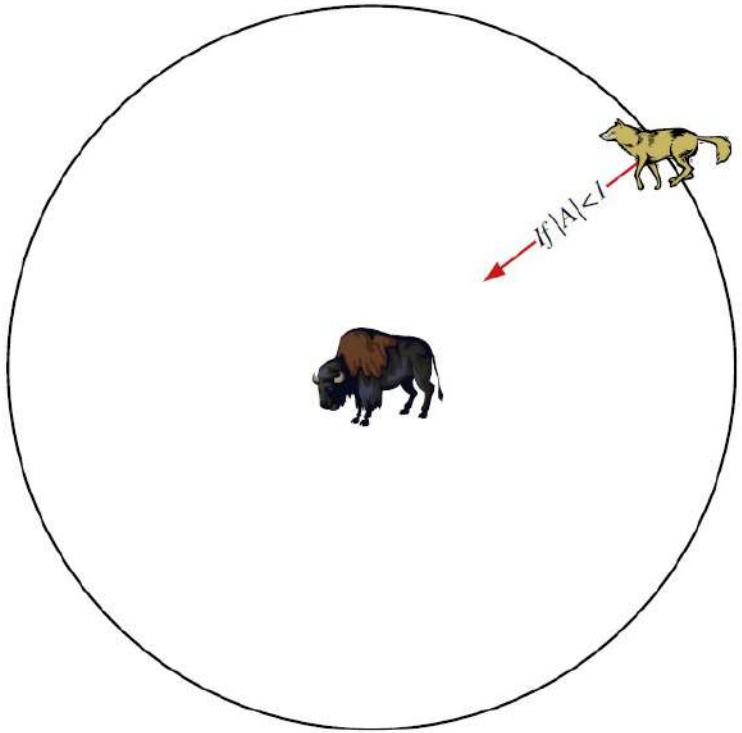
$$\vec{X}_1 = \vec{X}_\alpha - \vec{A}_1 \cdot (\vec{D}_\alpha), \vec{X}_2 = \vec{X}_\beta - \vec{A}_2 \cdot (\vec{D}_\beta), \vec{X}_3 = \vec{X}_\delta - \vec{A}_3 \cdot (\vec{D}_\delta)$$

$$\vec{X}(t+1) = \frac{\vec{X}_1 + \vec{X}_2 + \vec{X}_3}{3}$$



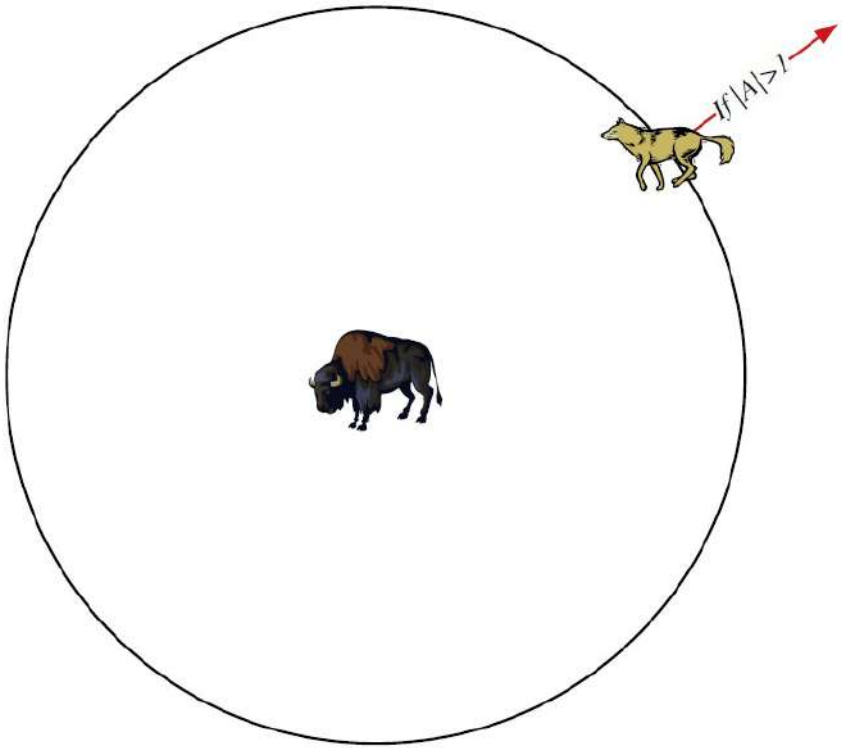
# Attacking prey (exploitation)

- The grey wolf finishes the hunt by attacking the prey when it stops moving.
- The vector $A$ is a random value in interval $[-2a, 2a]$, where $a$ is decreased from 2 to 0 over the course of iterations.
- When $|A| < 1$, the wolves attack towards the prey, which represents an exploitation process.



# Search for prey (exploration)

- The exploration process in GWO is applied according to the position, and that diverges from each other to search for prey and converges to attack prey.
- The exploration process is modeled mathematically by utilizing *A* with random values greater than 1 or less than -1 to oblige the search agent to diverge from the prey.
- When|A| > 1, the wolves are forced to diverge from the prey to find a fitter prey.



# GWO algorithm

```
Initialize the grey wolf population X_i (i = 1, 2, ..., n)
Initialize a, A, and C
Calculate the fitness of each search agent
X_α=the best search agent
X_β=the second best search agent
X_δ=the third best search agent
while (t < Max number of iterations)
    for each search agent
            Update the position of the current search agent
    end for
    Update a, A, and C
    Calculate the fitness of all search agents
    Update X_α, X_β, and X_δ
    t=t+1
end while
return X_α
```

ELTE | IK
INFORMATIKAI KAR

# Multi-Agent Simulation & Learning

**Introduction to Machine Learning**
Computer Science BSc Course, ELTE Faculty of Informatics

László Gulyás

Associate Professor
Department of Artificial Intelligence
lgulyas@inf.elte.hu

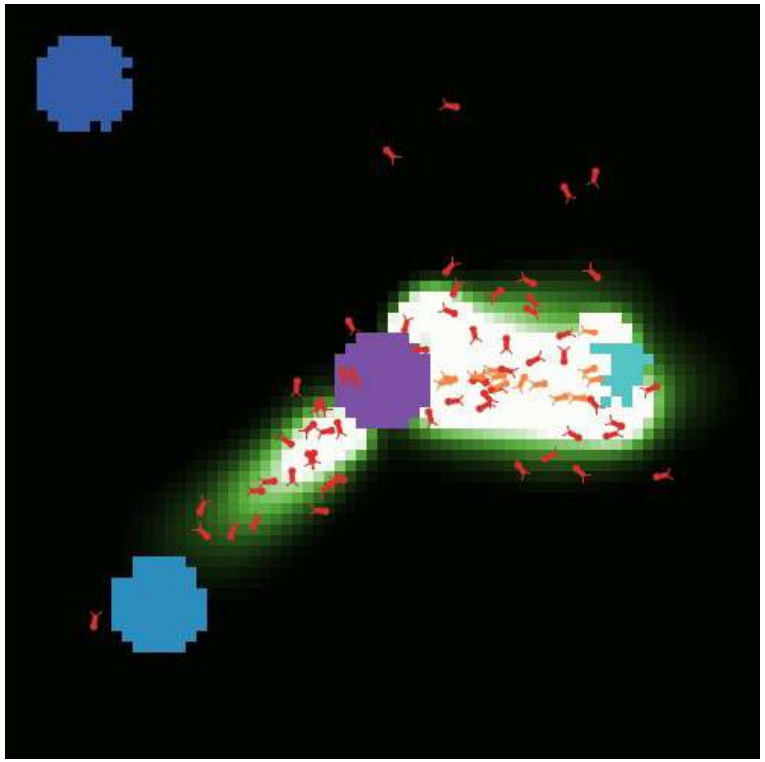
ELTE | IK
DEPARTMENT OF ARTIFICIAL INTELLIGENCE

# Motivation

Multi-Agent Systems

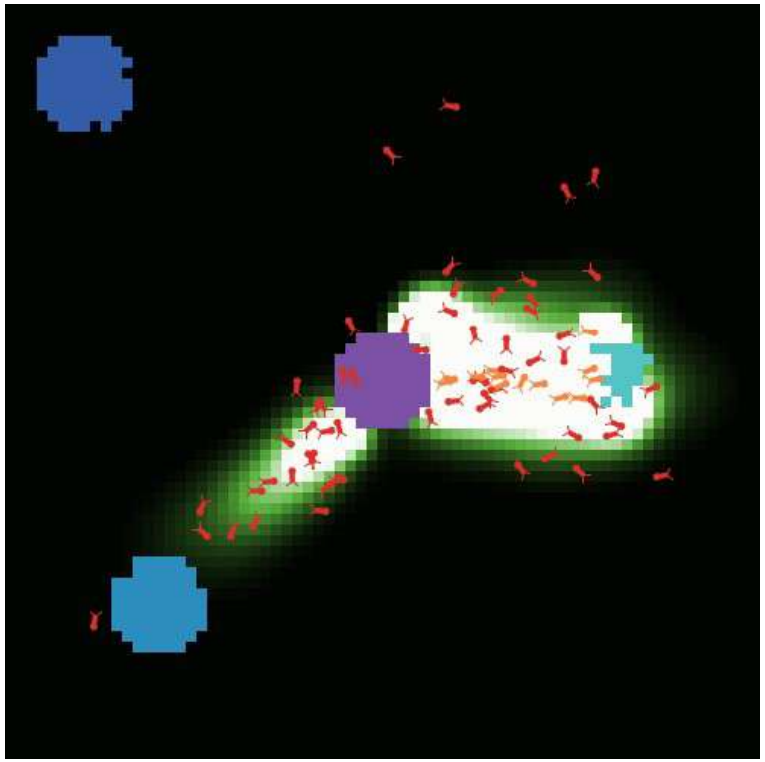
# Who is Intelligent?

- Foraging Ants



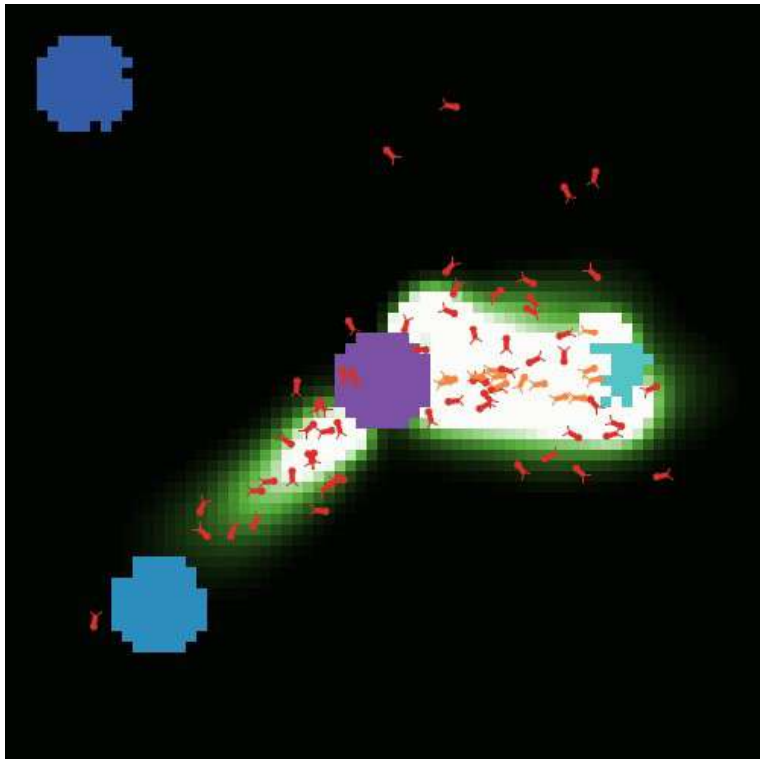
# Who is Intelligent?

- Foraging Ants
  - Intelligent?
  - Solves the problem
  - Robust and cheap…



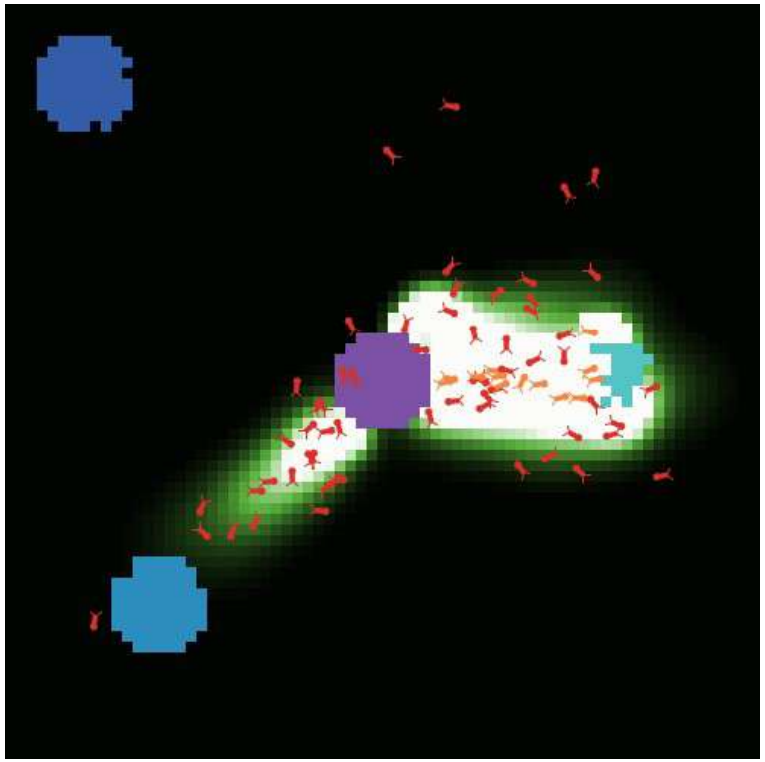
# Who is Intelligent?

- Foraging Ants
  - Intelligent?
  - Solves the problem
  - Robust and cheap…
- In the eye of the beholder



# Who is Intelligent?

- Foraging Ants
  - Intelligent?
  - Solves the problem
  - Robust and cheap…
- In the eye of the ~~beholder~~ observer
  - And there are different levels of observation (consideration)

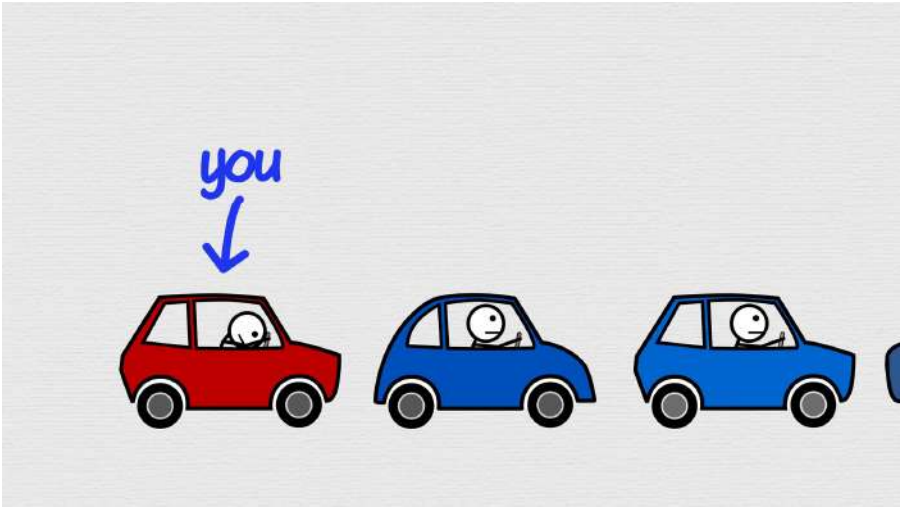


# Micro and Macro Behavior

- Building Systems
  - From top-down
  - From bottom-up
- When parts also have autonomy
  - Make decisions (have behavior)
  - Are potentially rational/intelligent
  - Come from different (perhaps unknown) provenance
  - Have potentially different goals
- When intelligence is (also) measured at system level
  - Collective Intelligence

# Unintended consequences
## Driving on the road

© CGP Grey https://www.cgpgrey.com/



# The micro-macro dichotomy

How to work with it?

# Basic Notion

# Emergence

- Something observable at a ‚higher level'
  - That does not exist at ‚lower levels'
  - That is not expressable at ‚lower levels'
- This can be a property or a behavior
- Hard concept to define formally
  - Subject to discussions and interpretations
- **Yet, at the core of ‚Micro vs Macro'**

# Two Related Topics

**Analysis:**
**Agent-Based Simulation**

Experimenting with
- Possible *micro* rules that
- G*enerate*
- (observed) *macro* behavior

→ A tool of explanation

**Construction:**
**Multi-Agent (Reinf.) Learning**

- Learning to Cooperate
- Learning to Swarm

# Agent-Based Simulation

An Example, A Tool

# New Segregation: Races accept divide

[US News Excerpt]

Exclusive Report

## New segregation: Races accept divide

But experts say financial, social costs of living apart are just as detrimental

By Ron French / *The Detroit News*

More than four decades after Americans fought and died to end segregation, many in Metro Detroit are comfortable living apart.

# Why?

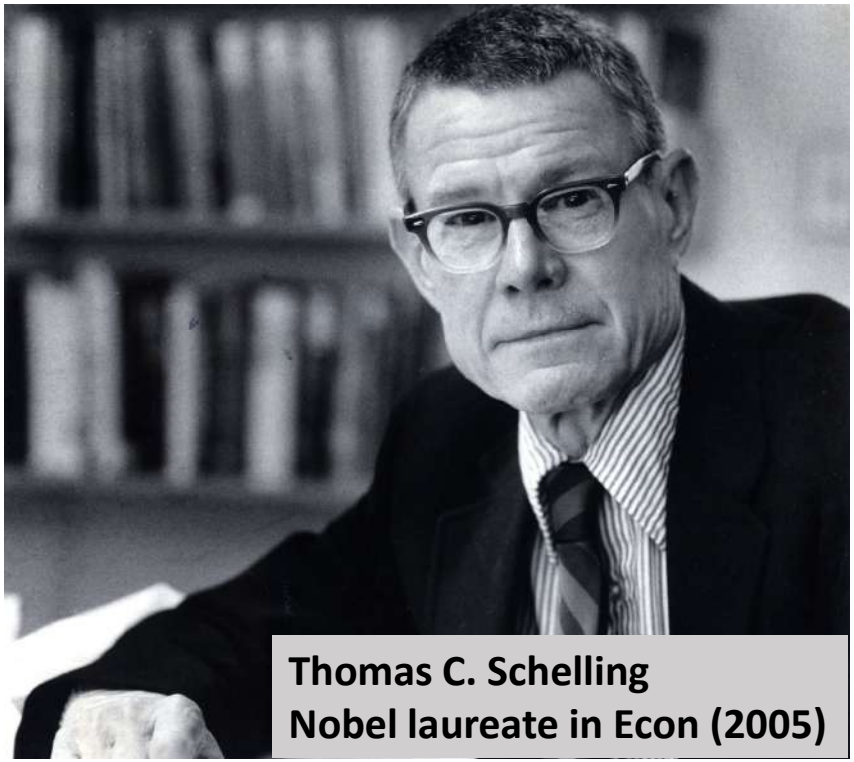
- Endless number of opinions
  - Economical reasons
  - The level of racism in the society
  - Etc.

- Untested opinions
  - To test: *simplifications* → *models*

# A Model of Housing Segregation by *Thomas C. Schelling* (1978)

- Residential choices
  - Stylized 2D environment
  - “Oranges” and “Blues”
  - The level of personal tolerance (given as a %)

- **Where does a general 60% tolerance level lead us?**
  - **And 70%?**

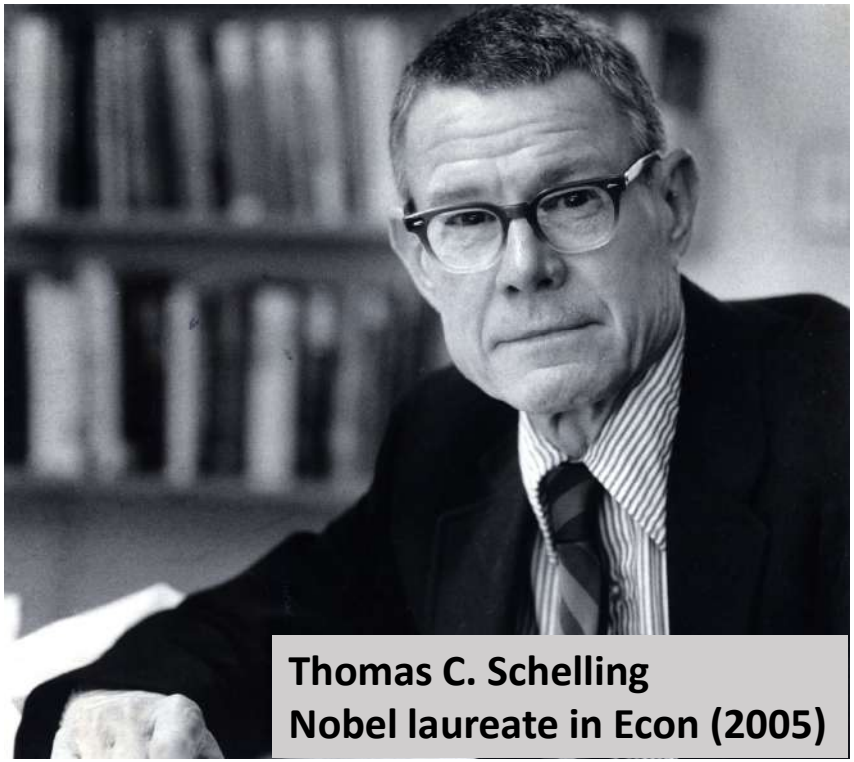


**Thomas C. Schelling**
**Nobel laureate in Econ (2005)**

# A Model of Housing Segregation by *Thomas C. Schelling* (1978)

```
Tolerance (int) in [0,100]: (%)

Neighbors (int) in [0,100]:
  #neighbors_of_other_color //
  #all_neighbors

Happy (boolean):

  Neighbors <= Tolerance


Movement Rule:
     If (not Happy):
       Move_to_an_empty_location


Variants:
```

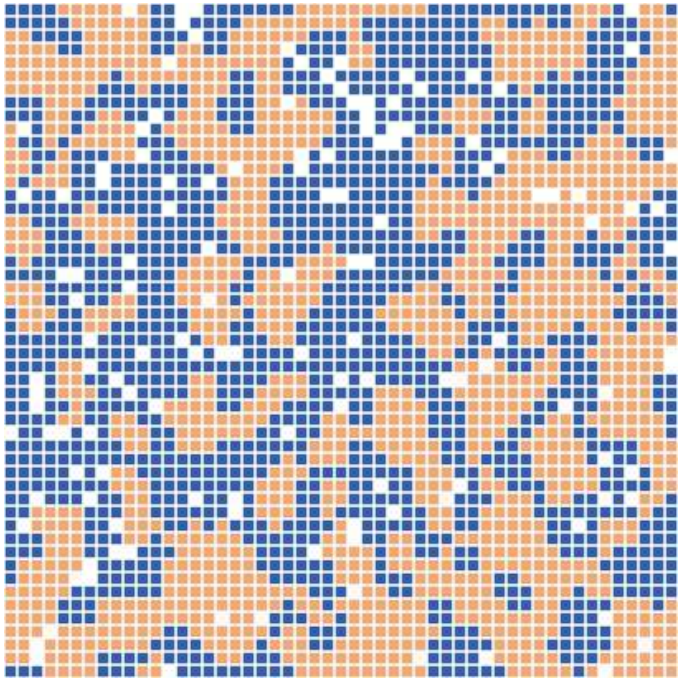
- Move_to_random_empty_location_if_happy_there
- Move_to_closest_empty_location
- Etc.
- Various neighborhood definitions possible



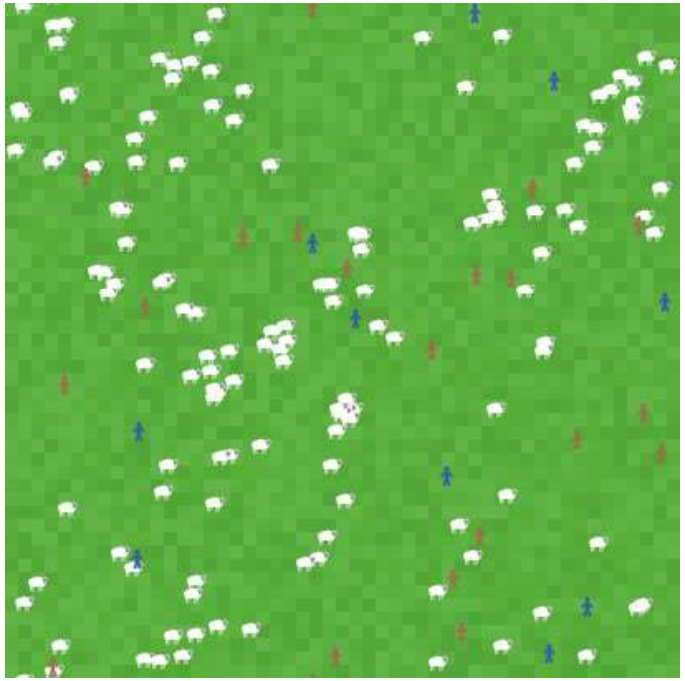
**Thomas C. Schelling**
**Nobel laureate in Econ (2005)**

# 70 percent tolerance, random initialisation

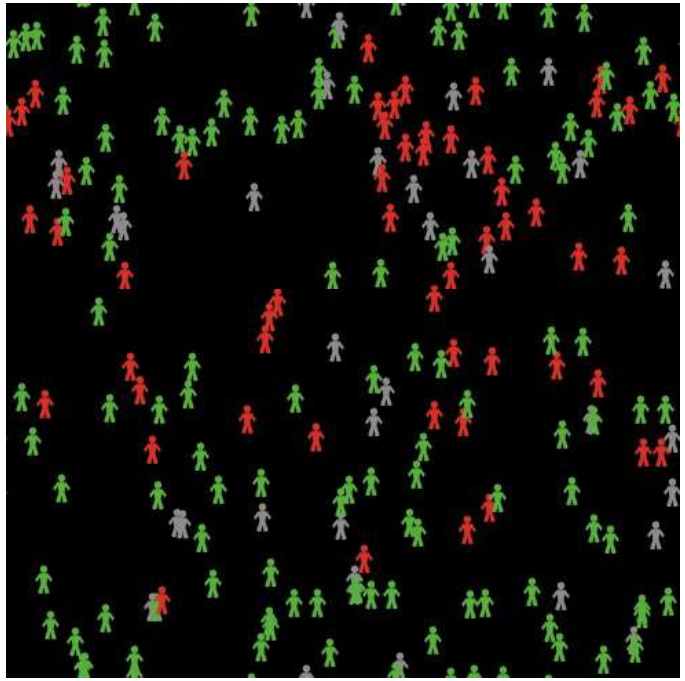
# 70 percent tolerance, random initialisation



# NetLogo
https://ccl.northwestern.edu/netlogo/



Models library/Biology/Shepherds



Models library/Biology/Virus

# Multi-Agent (Reinf.) Learning

Some Examples

# Goal – „Learning to Swarm”

- Training teams of mobile robots
  - To solve a problem collectively
  - → Multi-Agent Reinforcement Learning (MARL)

- ‚Swarm robotics in spirit’:
  - No control hierarchy
  - No explicit, direct communication
  - No complex negotiation protocols
  - No explicit dependence on team size
    - Cf. Graceful degradation

# Background of Examples

**Technical Setup**

- StableBaseline3
  - Multi layer perceptron (MLP)
  - LSTM
- Training
  - Proximal Policy Optimisation
  - Q-learning
  - Actor-Critic Model
- Negative reward for collision

**Challenges Explored**

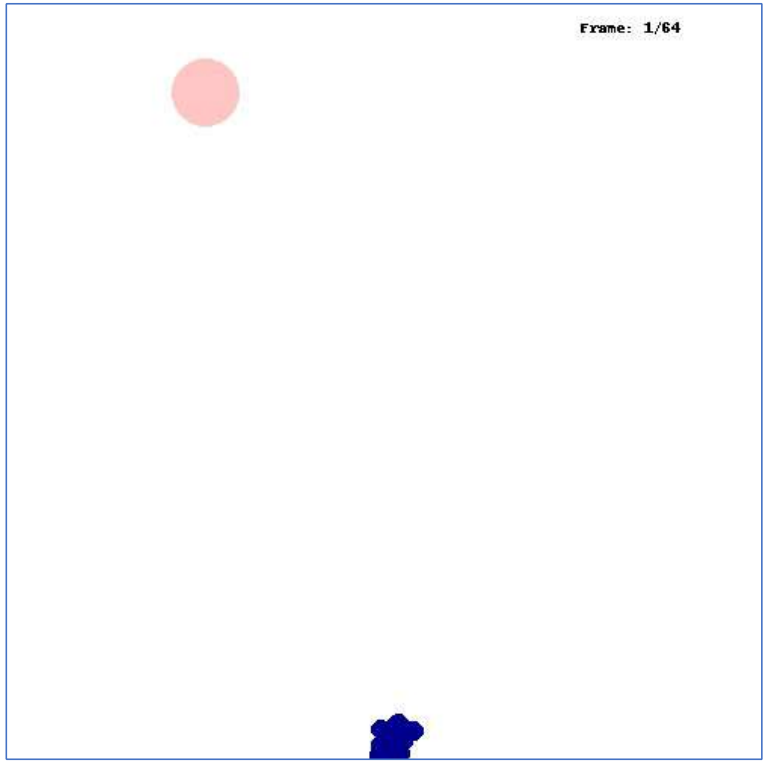
- Traditional models
  - Fixed input size (observation space )
- Varying #agents
  - Challenge →
  - Masking, based on distance
  - Pooling based on distance (k-Nearest Neighbors)

# Simple Tag Game
# with Multi-Agent Actor Critic policy learning

- Normal approach
  - Single agent policy learning for each agent
- The Multi-Agent Actor Critic
  - Decentralized actors (policies)
  - Centralized critic
  - → Info sharing
- Better for multi-agent environments

# Move the team to the target area

- Conditions
  - Agents have limited vision
    - N closest agent and the goal state if in vision range
  - No direct communication
  - Grid environment
  - Terminate: >80% at target
- Approach
  - MLP model and PPO policy learning
- Behavior:
  - Agents spreading out as much as they can see each other
  - After finding the goal, they oscillate to propagate the information

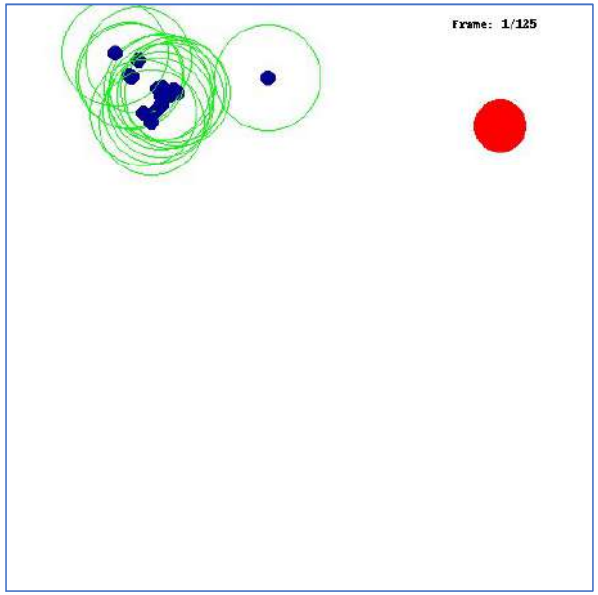


# Move the team to the target area

## 2 distinct strategies

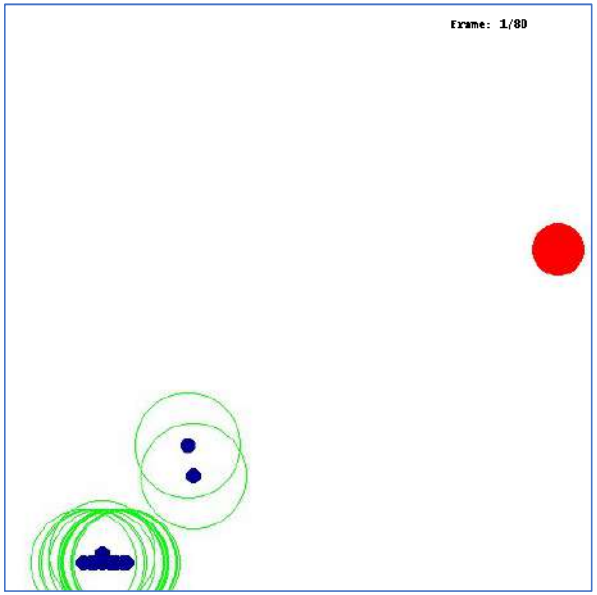
**Fast Exploration**

Spread wide, communicate later



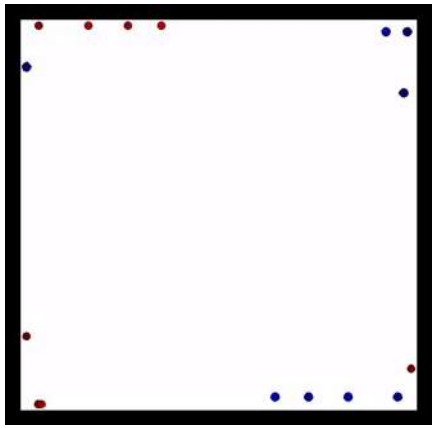

**Fast communication**

Stick and explore together

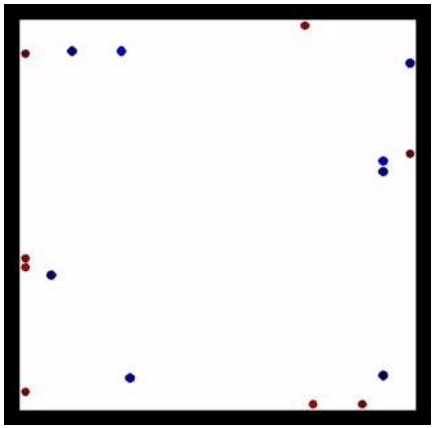


# Move to the Target
## Multi-Agent Multi-Target Case

- Incremental and curriculum learning
  - Agents learn step by step from individual “walk” to move collectively

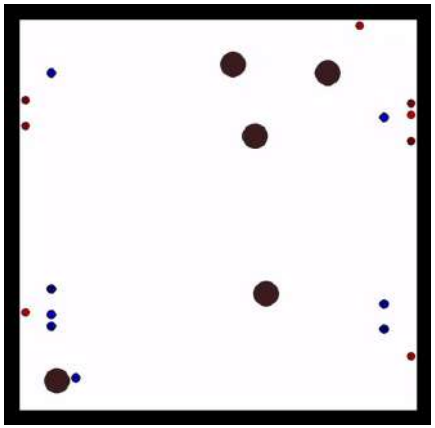


Punishment for staying on the border



Punishment for being far away from target

Punishment for colliding with agents



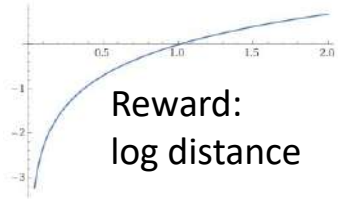
Punishment for colliding with obstacles

**Red:** target, **Blue:** agent, **Black:** obstacle

# Move to the Target
## Multi-Agent Multi-Target Case
## w/o incremental learning

**Red:** target, **Blue:** agent, **Black:** obstacle

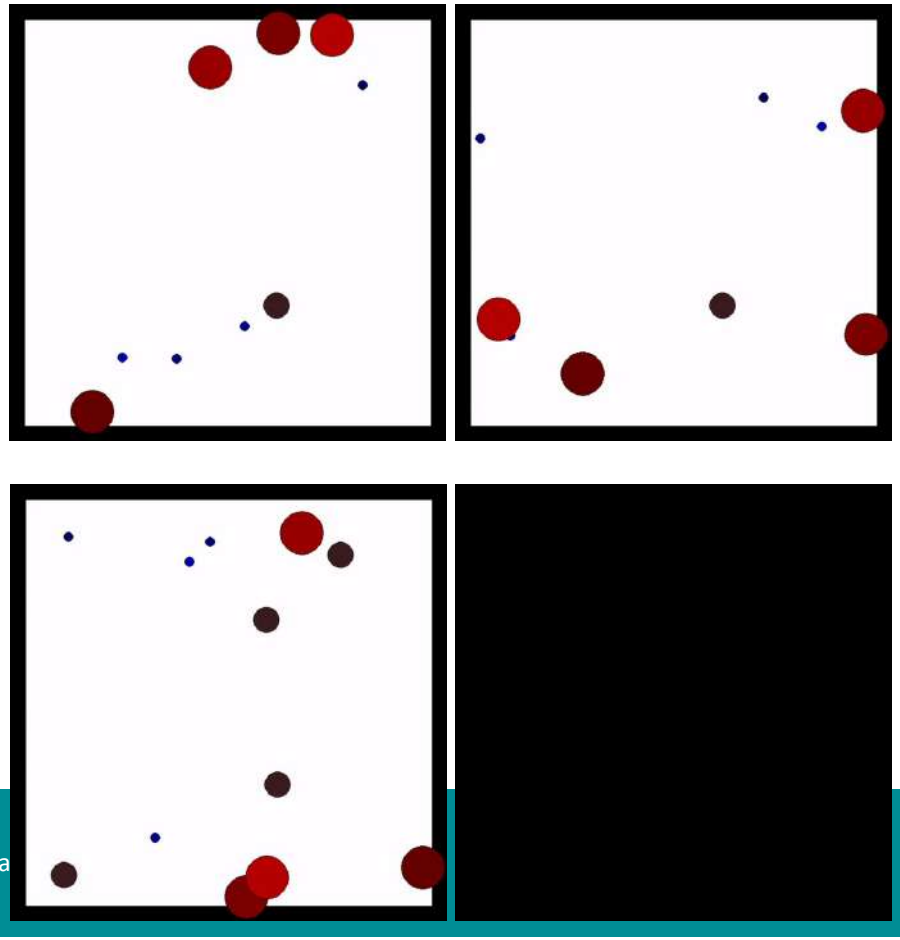
- Non-linear (log) reward




Reward:
log distance

  - Maintains local focus
  - Helps avoiding collision

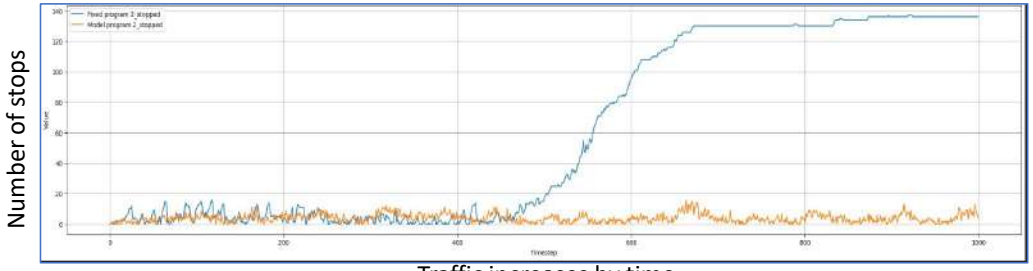
- Simplified observation space
  - K nearest objects
  - Same trained model for arbitrary #agents, #targets



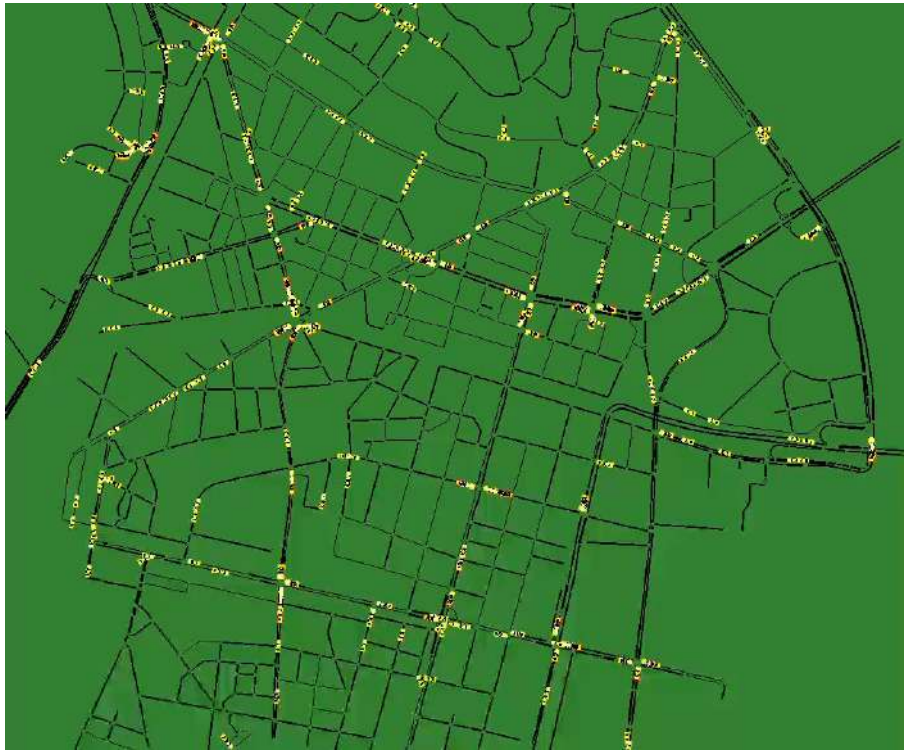
# Traffic Signal Control
## with local communication

### Classic approach
- Independent intersections, or
- Centralized control

### Our approach
- Independent intersections (agents)
  - Communicating with nearest neighbors
- Agent classes
  - Based on #neighbors




Traffic increases by time



ELTE | IK
INFORMATIKAI KAR

# Thank you!


## László Gulyás

Associate Professor
Department of Artificial Intelligence
lgulyas@inf.elte.hu


ELTE | IK
DEPARTMENT OF ARTIFICIAL INTELLIGENCE

ELTE | FACULTY OF INFORMATICS

# REINFORCMENT LEARNING

Introduction to Machine Learning – Lecture 6.

Balázs Nagy, PhD

ELTE | IK
DEPARTMENT OF ARTIFICIAL INTELLIGENCE

BOSCH

# References

- MIT course

  © Alexander Amini and Ava Amini

  MIT 6.S191: Introduction to Deep Learning

  http://introtodeeplearning.com/

- Georgia Tech – Machine Learning course

  https://www.youtube.com/watch?v=Jk2V9yA82YU

- What is general intelligence? How can it be defined?

*A very general mental capability that, among other things, involves* ***the ability to*** *reason, plan,* ***solve problems****, think abstractly, comprehend complex ideas, learn quickly* ***and learn from experience****. It is not merely book learning, a narrow academic skill, or test-taking smarts. Rather, it reflects a broader and deeper capability for comprehending our surroundings—"catching on," "****making sense****" of things, or "****figuring out****" ****what to do****.*

*/Wall Street Journal, 1994/*

- Intelligence in the context of machines
  – Artificial Intelligence (AI)

    *The theory and development of computer systems able to perform tasks normally requiring human intelligence, such as visual perception, speech recognition, decision-making, and translation between languages.*

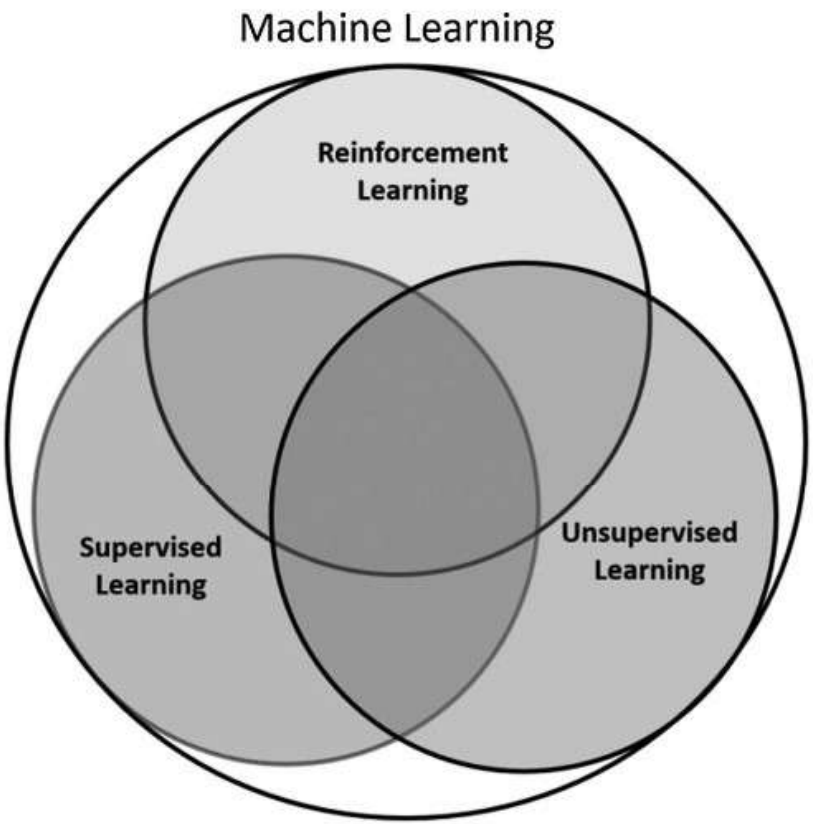
- Goal:

    Study the theory and design of algorithms that help machines (**agents**) gain the capability to perform tasks by **interacting with the environment** and **continually learning** from its successes, failures, and **rewards**

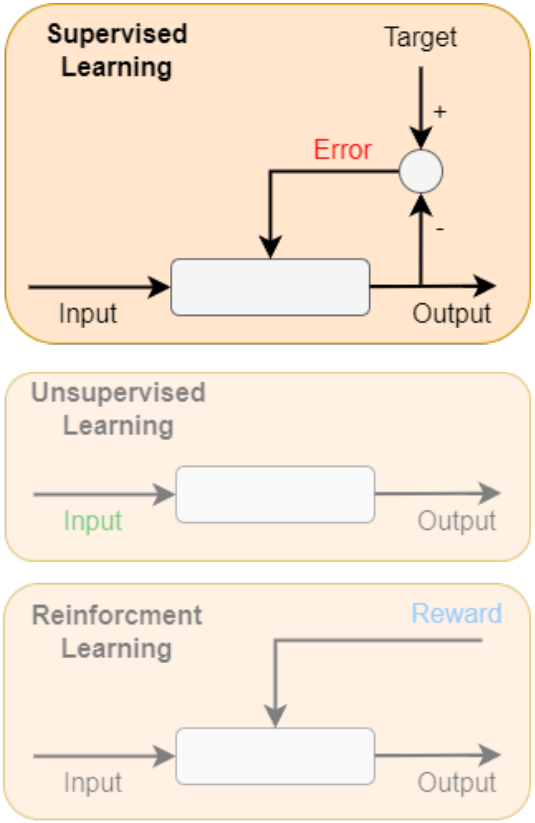
**Artificial Intelligence**
*Any technique that enables computers to mimic human intelligence.*

**Machine Learning**
*A subset of AI that includes abstruse statistical techniques that enable machines to improve at tasks with experience.*

**Deep Learning**
*The subset of machine learning composed of algorithms that permit software to train itself to perform tasks by exposing multilayered neural networks to vast amount of data.*

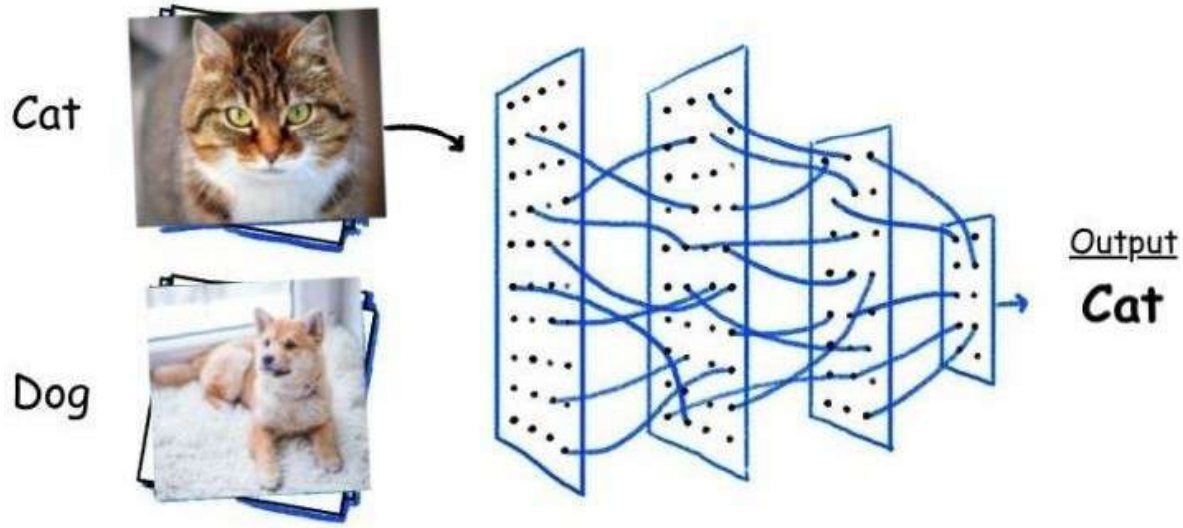


# Learning methods

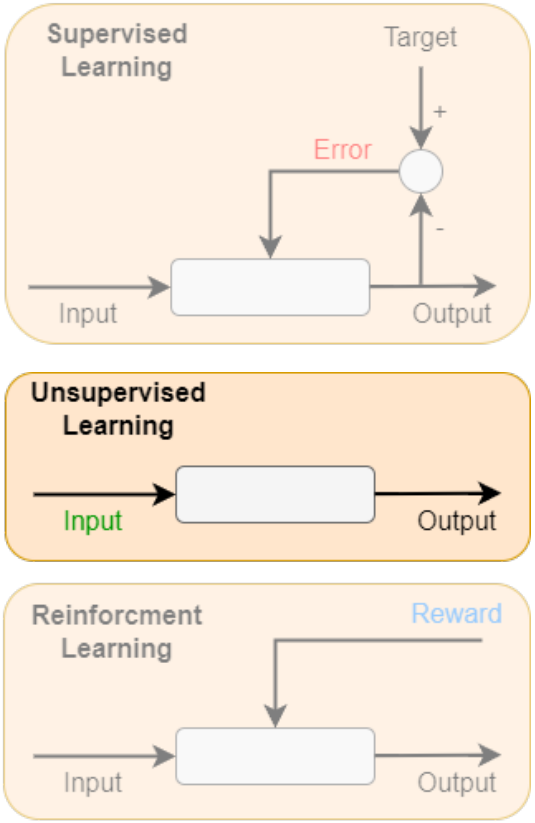



## Supervised Learning

- system is presented with the labeled data
- the objective is to generalize the knowledge so that new unlabeled data can be labeled

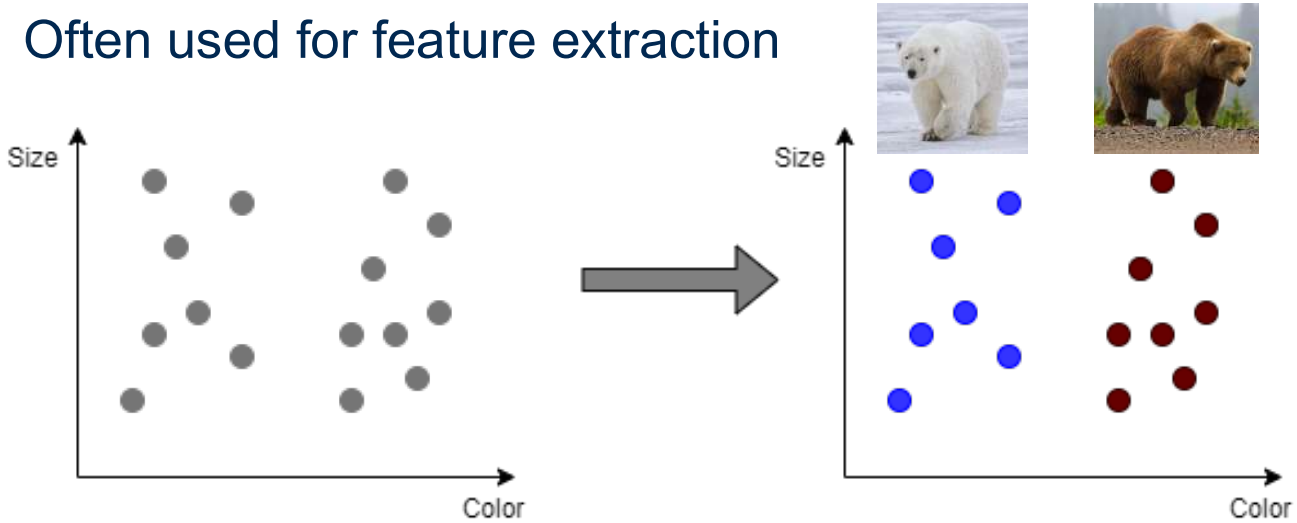


# Learning methods

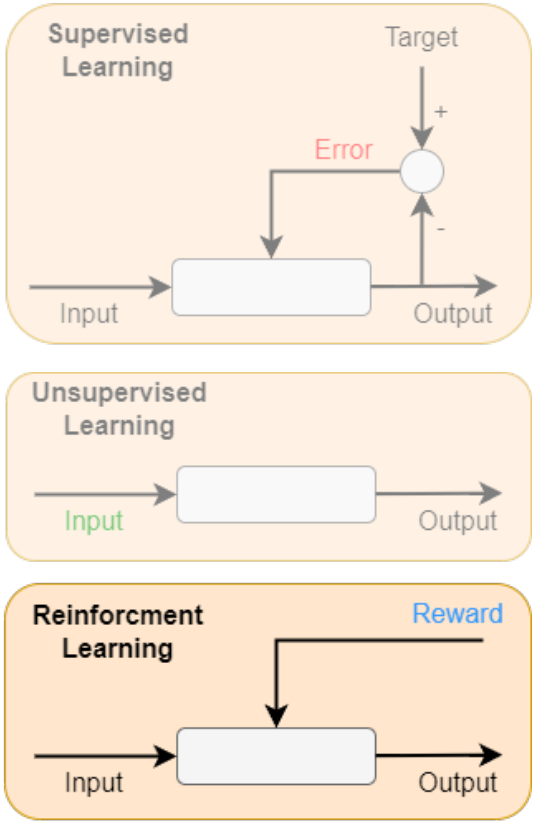



## Unsupervised Learning

- No labels, only has the inputs
- The system uses this data to learn the hidden structure of the data so that it can cluster/categorize the data into some broad categories.
- Often used for feature extraction



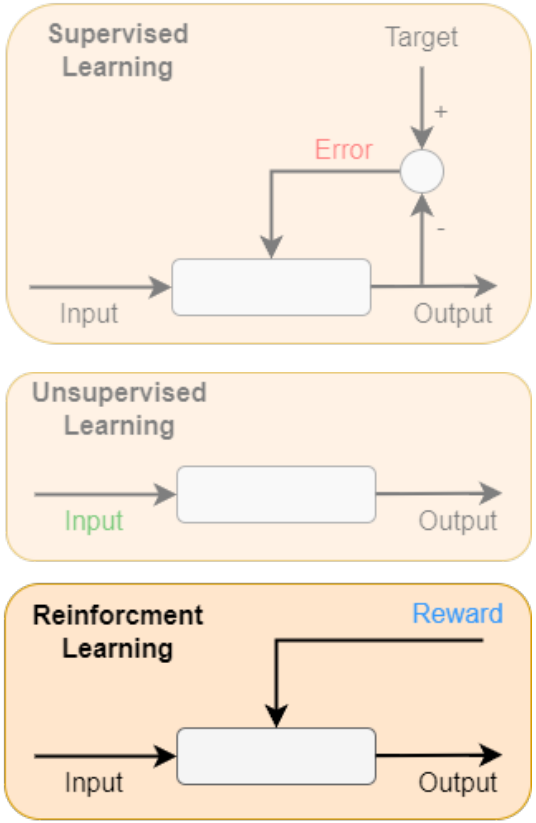
# Learning methods




## Reinforcment Learning

- The agent does not have prior knowledge of the system
- It gathers feedback and uses that feedback to plan/learn actions to maximize a specific objective.
- As it does not have enough information about the environment initially, it must explore to gather insights.
- Once it gathers “enough” knowledge, it needs to **exploit** that **knowledge** to start **adjusting** its **behavior** to **maximize the objective** it is chasing.

# Learning methods



# Reinforcement Learning (RL) – Key concept

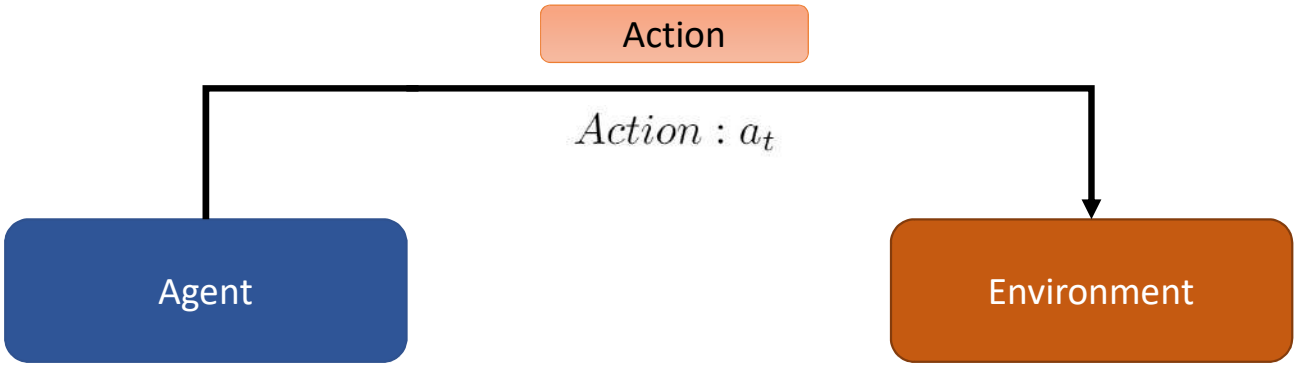
# Reinforcement Learning (RL) – Key concept

**Agent:** takes actions

# Reinforcement Learning (RL) – Key concept

**Environment:** the world in wich the agent exists and operates

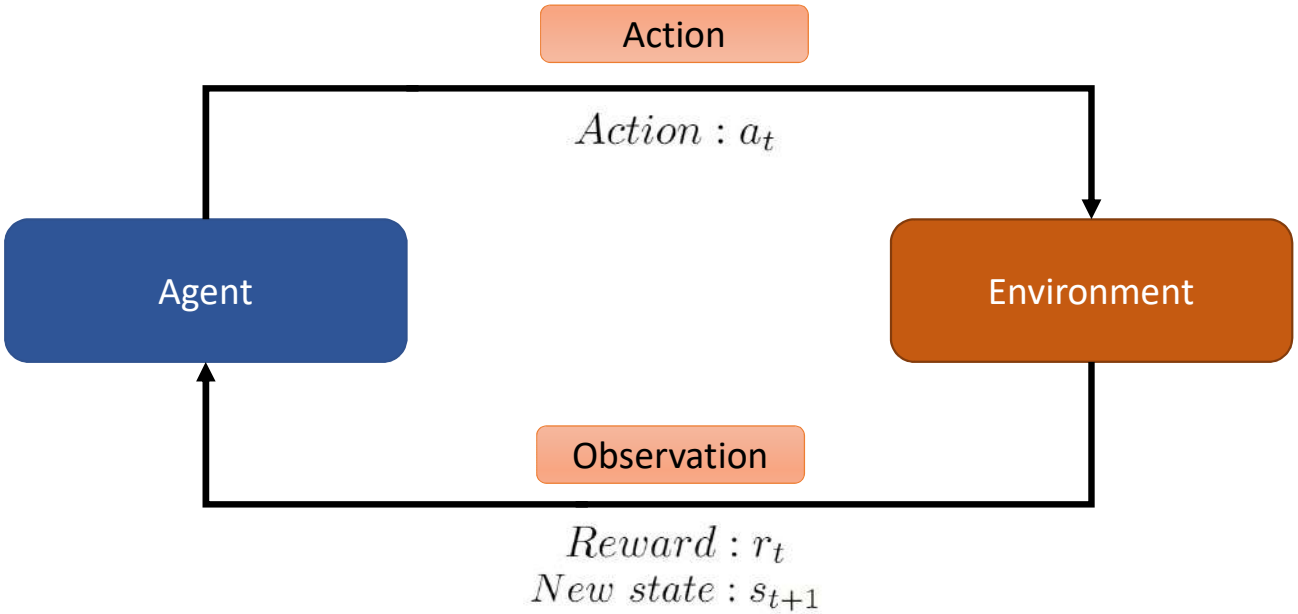
# Reinforcement Learning (RL) – Key concept




**Action:** a movement the agent can make in the environment

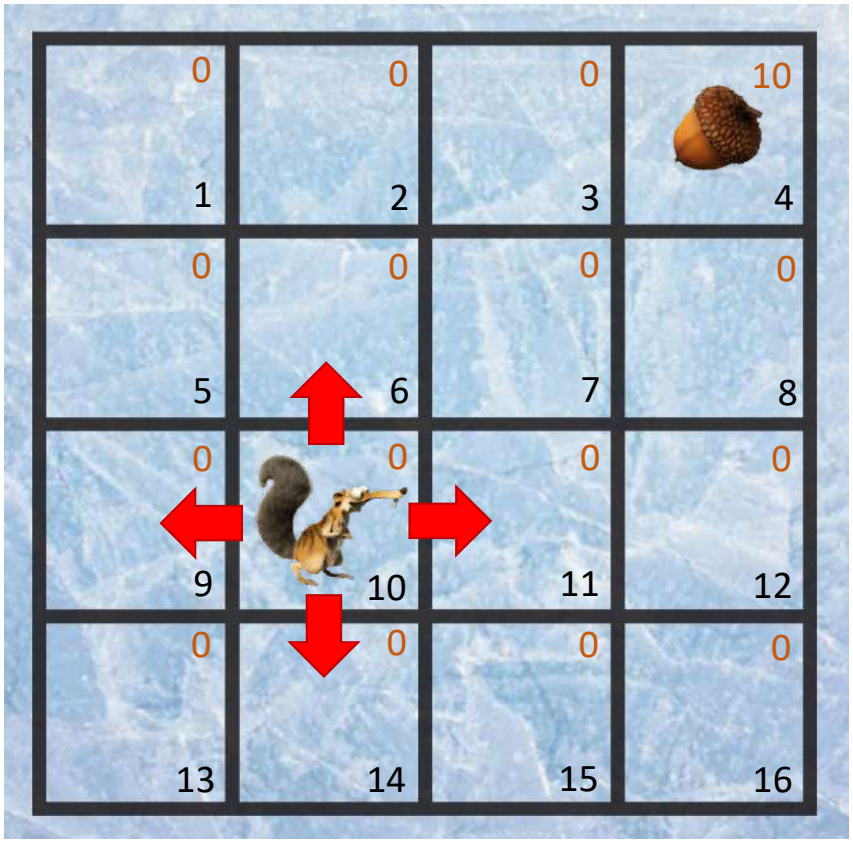
**Action space *A***: the set of possible actions an agent can make in the environment

# Reinforcement Learning (RL) – Key concept




**Reward:** feedback that measures the success or failure of the agent's action

# Grid world example




**Environment**

**Agent**

**State**

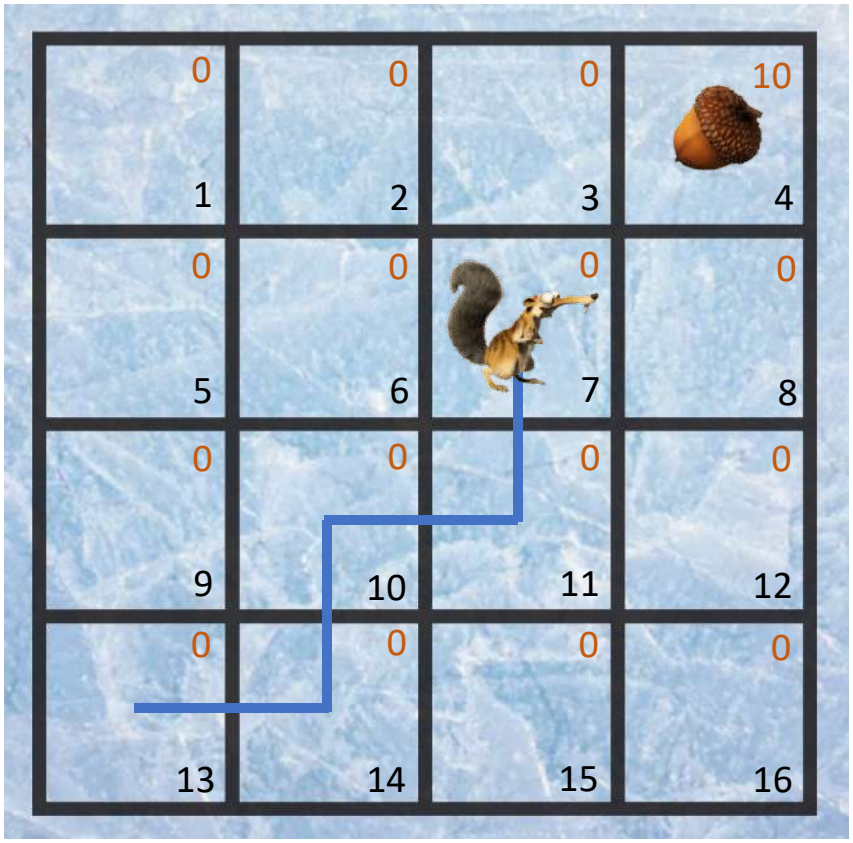
**Action**

**Reward**

**Model:** rules of the game, the physics of the world

**Policy:** A policy is a strategy that an agent uses in pursuit of goals.

# Grid world example




**Environment**

**Agent**

**State**

**Action**

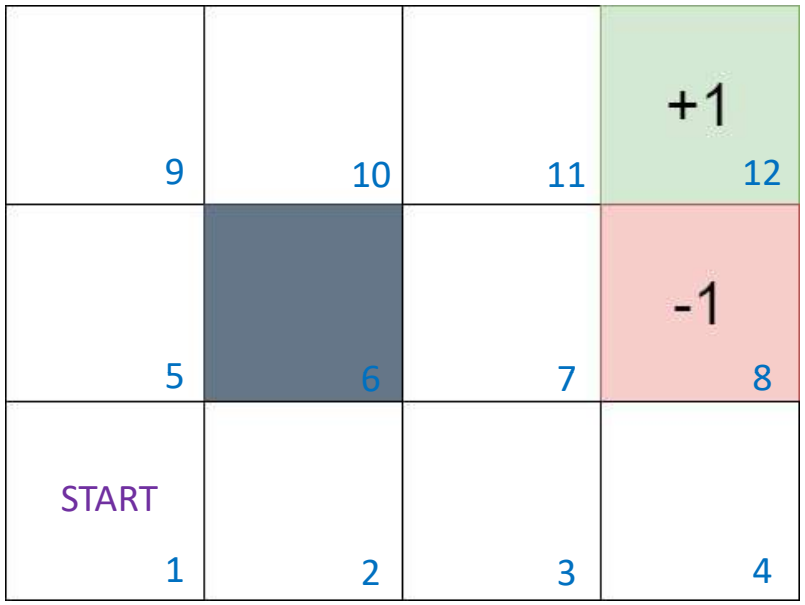
**Reward**

} Trajectory

$$\boxed{S_{13} \rightarrow A_r \rightarrow R(+0)} \rightarrow \boxed{S_{14} \rightarrow A_u \rightarrow R(+0)} \rightarrow$$
$$\rightarrow \boxed{S_{10} \rightarrow A_r \rightarrow R(+0)} \rightarrow \ldots$$

# Markov Decision Process

# Markov Decision Process




[up, down, left, right]

**States:** $S$ =[1,2,3,4,5,6,7,8,9,10,11,12]

**Model:** $T(s, a, s') \sim Pr(s'|s, a)$

**Actions:** $A(s), A$ =[up, down, left, right]

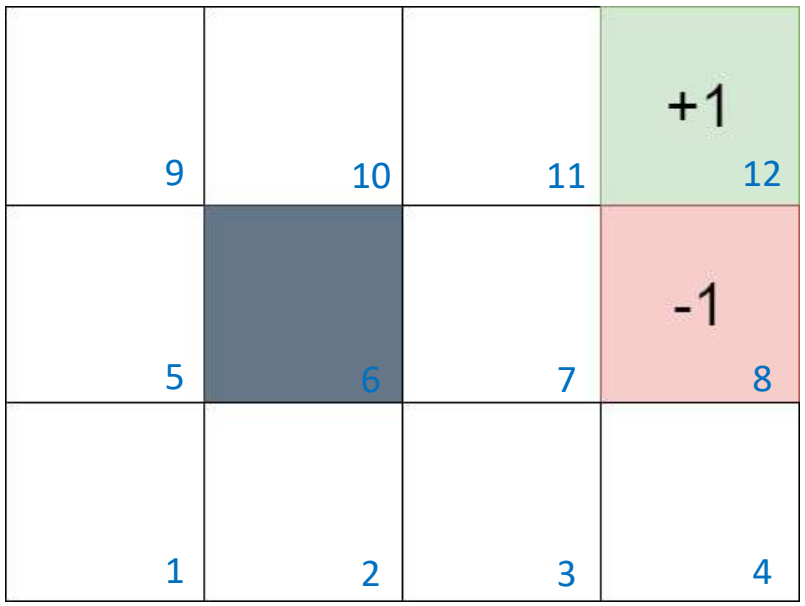
**Reward:** $R(s), R(s, a), R(s, a, s')$

---

**Policy:** $\pi(s) \rightarrow a$

$\pi^*$

# Markov Decision Process




[up, down, left, right]

**Model:** $T(s, a, s') \sim Pr(s'|s, a)$

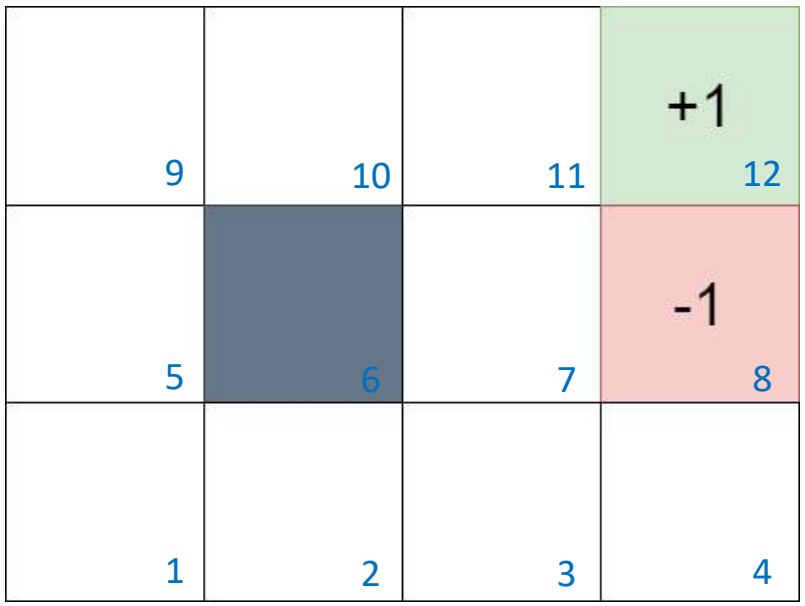
**Deterministic:**

$$\left.\begin{aligned} Pr(s_5|s_1, a_{up}) &= 1 \\ Pr(s_2|s_1, a_{up}) &= 0 \end{aligned}\right\} \text{Sum} = 1$$

**Non-deterministic or Stochastic:**

$$\left.\begin{aligned} Pr(s_7|s_3, a_{up}) &= 0.8 \\ Pr(s_4|s_3, a_{up}) &= 0.1 \\ Pr(s_2|s_3, a_{up}) &= 0.1 \\ Pr(s_1|s_3, a_{up}) &= 0 \end{aligned}\right\} \text{Sum} = 1$$

# Markov Decision Process




[up, down, left, right]

**Markovian property:**

- Only present matters
- Stationary (rules do not change)

**States:** $S$

**Model:** $T(s, a, s') \sim Pr(s'|s, a)$

**Actions:** $A(s), A$

**Reward:** $R(s), R(s, a), R(s, a, s')$

MDP

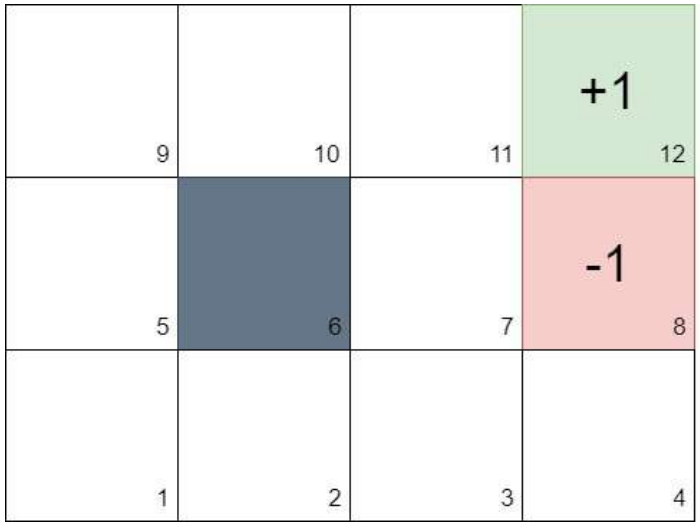
*Problem definition*

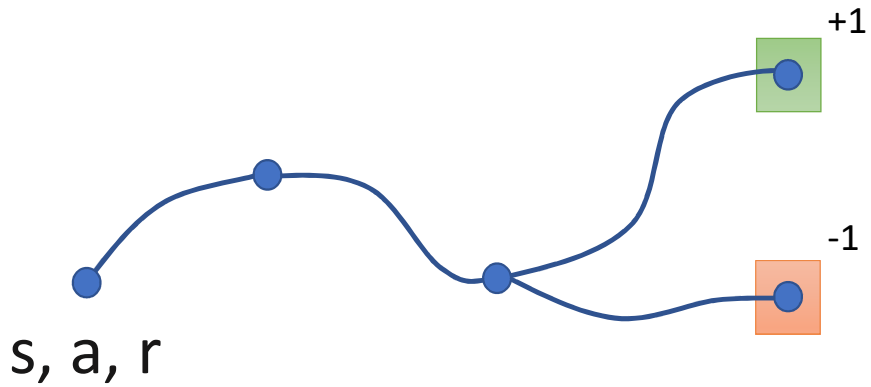
**Policy:** $\pi(s) \rightarrow a$

*Solution*

$\pi^*$ **Optimal policy:** maximises the long term expected reward

# Markov Decision Process

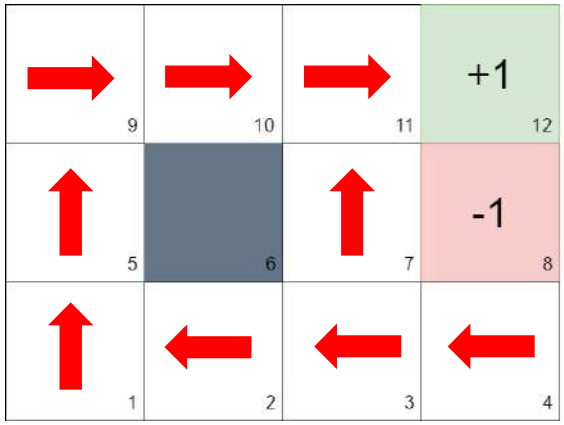



- Delayed reward
- Minor changes matters

**Temporal Credit Assigment**

# Markov Decision Process




**Rules:**

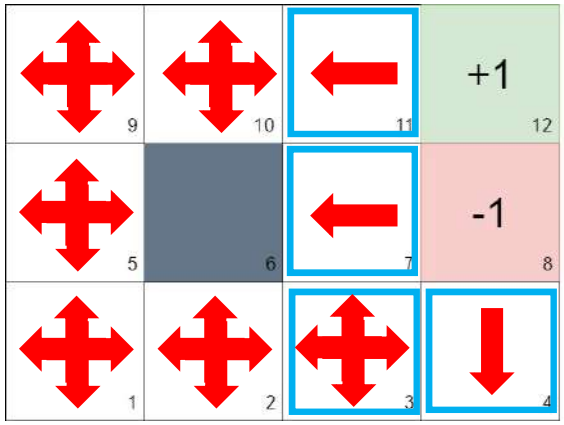
- **Stochastic**
- **Rewards given**

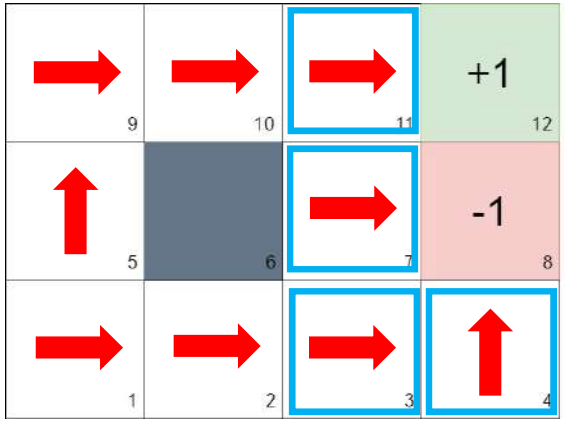
$$R(s) = -0.04$$

$$R(s) = +2$$

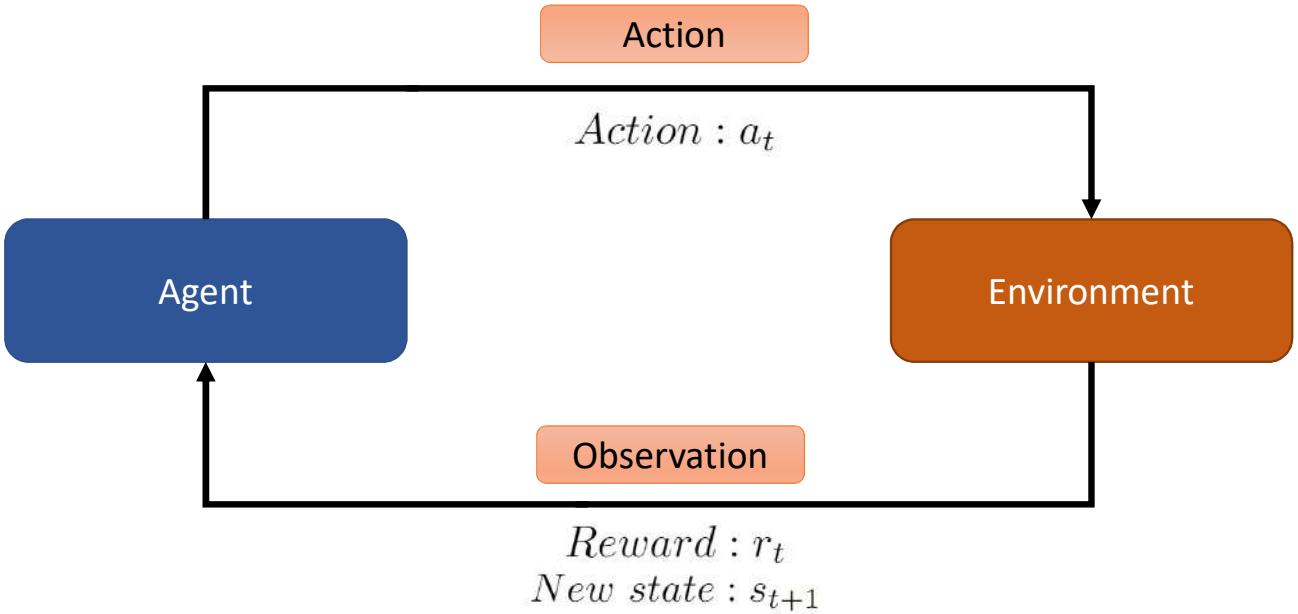
**Action Space:**

$$R(s) = -2$$



# Reinforcement Learning (RL) – Key concept




Total Reward

$$R_t = \sum_{i=t}^{\infty} r_i$$

$$R_t = \sum_{i=t}^{\infty} \gamma^i r_i$$

Discounted Total Reward

## Defining the Q-function

$$R_t = r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \gamma^3 r_{t+3} + \cdots$$

Total reward, $\mathbf{R_t}$, is the discounted sum of all rewards obtained from time **t**

$$Q(s_t, a_t) = \mathbb{E}[R_t | s_t, a_t]$$

The Q-function captures the expected total future reward an agent in state, s, can receive by executing a certain action, a

## How to take actions given a Q-function?

$$Q(s_t, a_t) = \mathbb{E}[R_t | s_t, a_t]$$

Ultimately, the agent needs a **policy *π*(*s*)**, to infer the best action to take at its state, ***s***

**Strategy**: the policy should choose an action that maximizes future reward

$$\pi^*(s) = \arg\max_a Q(s, a)$$

# Deep Reinforcment Learning Algorithms

## Value Learning

Find $Q(s,a)$

$a = \underset{a}{\operatorname{argmax}}\, Q(s,a)$

## Policy Learning

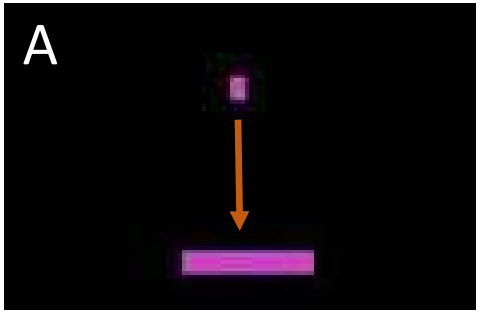
Find $\pi(s)$

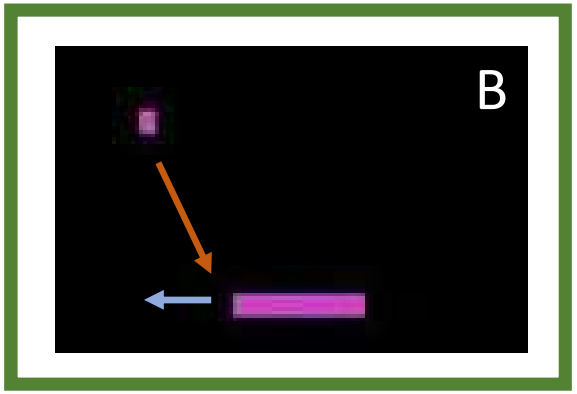
Sample $a \sim \pi(s)$

# Q function intuition

Atari game - Breakout

It can be very difficult for humans to accurately estimate Q-values

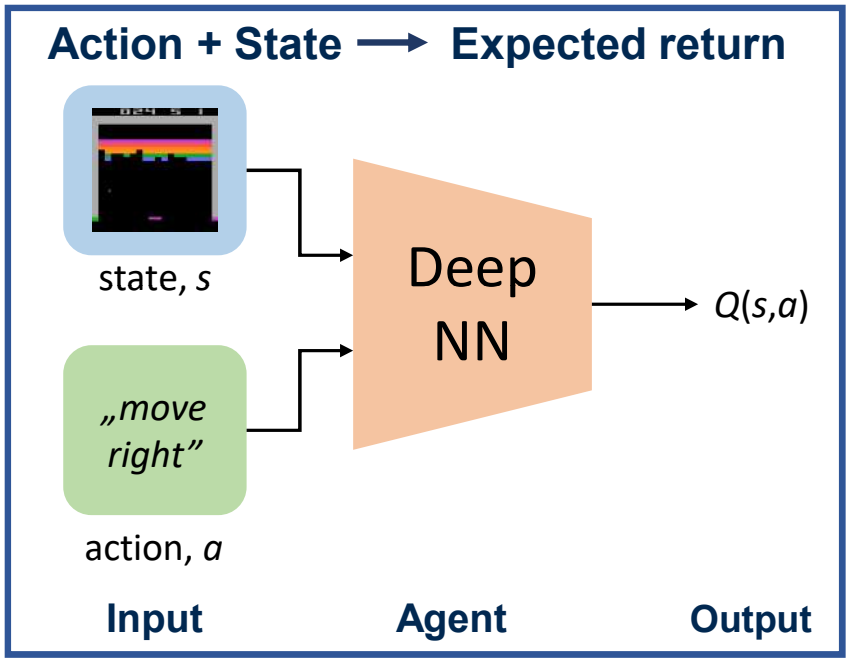


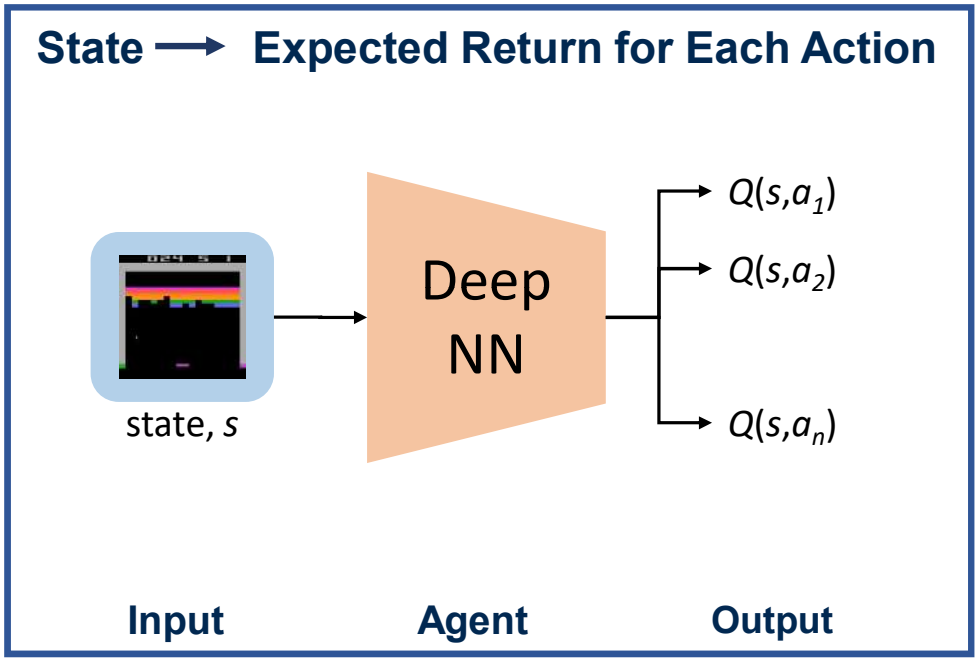




Which (s, a) pair has a higher Q-value?

# Deep Q Networks (DQN): Training

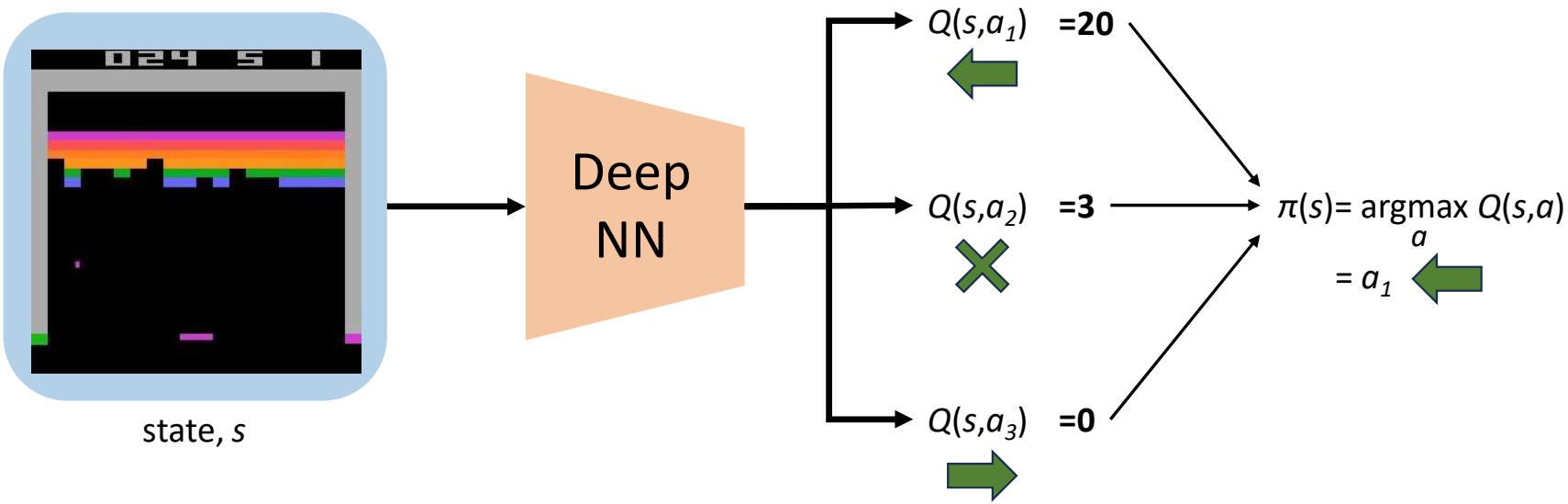
How can we use deep neural networks to model Q-functions?



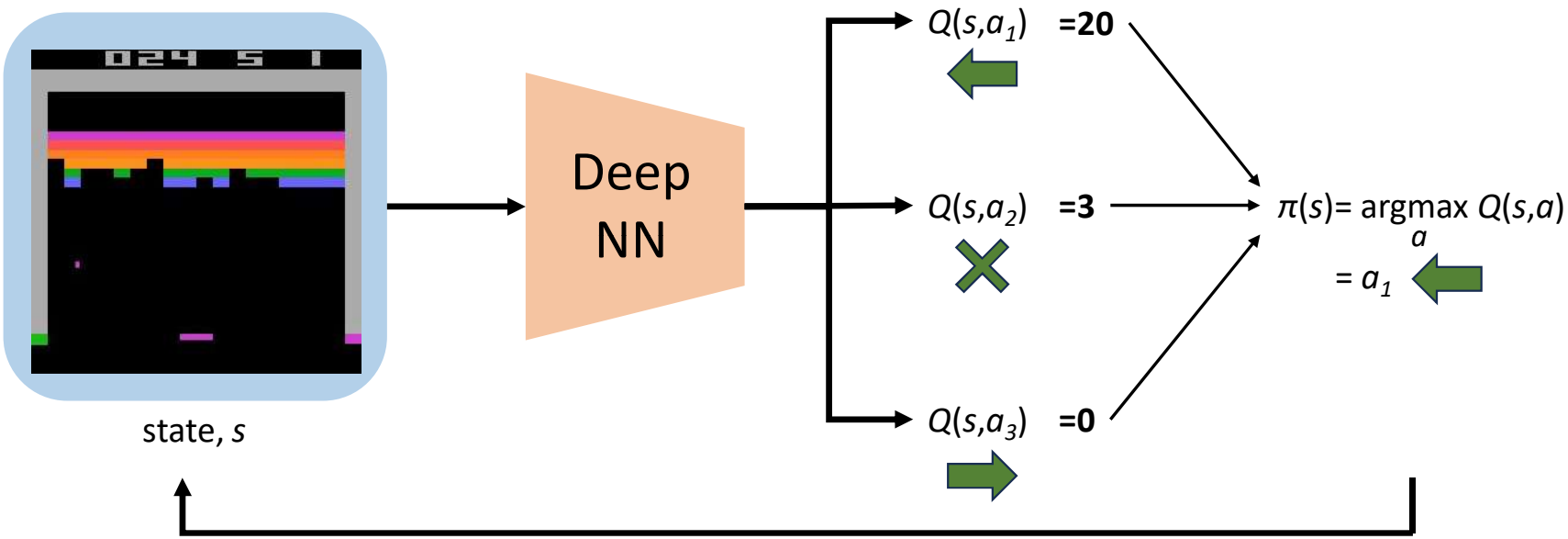
# Deep Q Networks Summary

Use NN to learn Q-function and then use to infer the optimal policy, $\pi(s)$



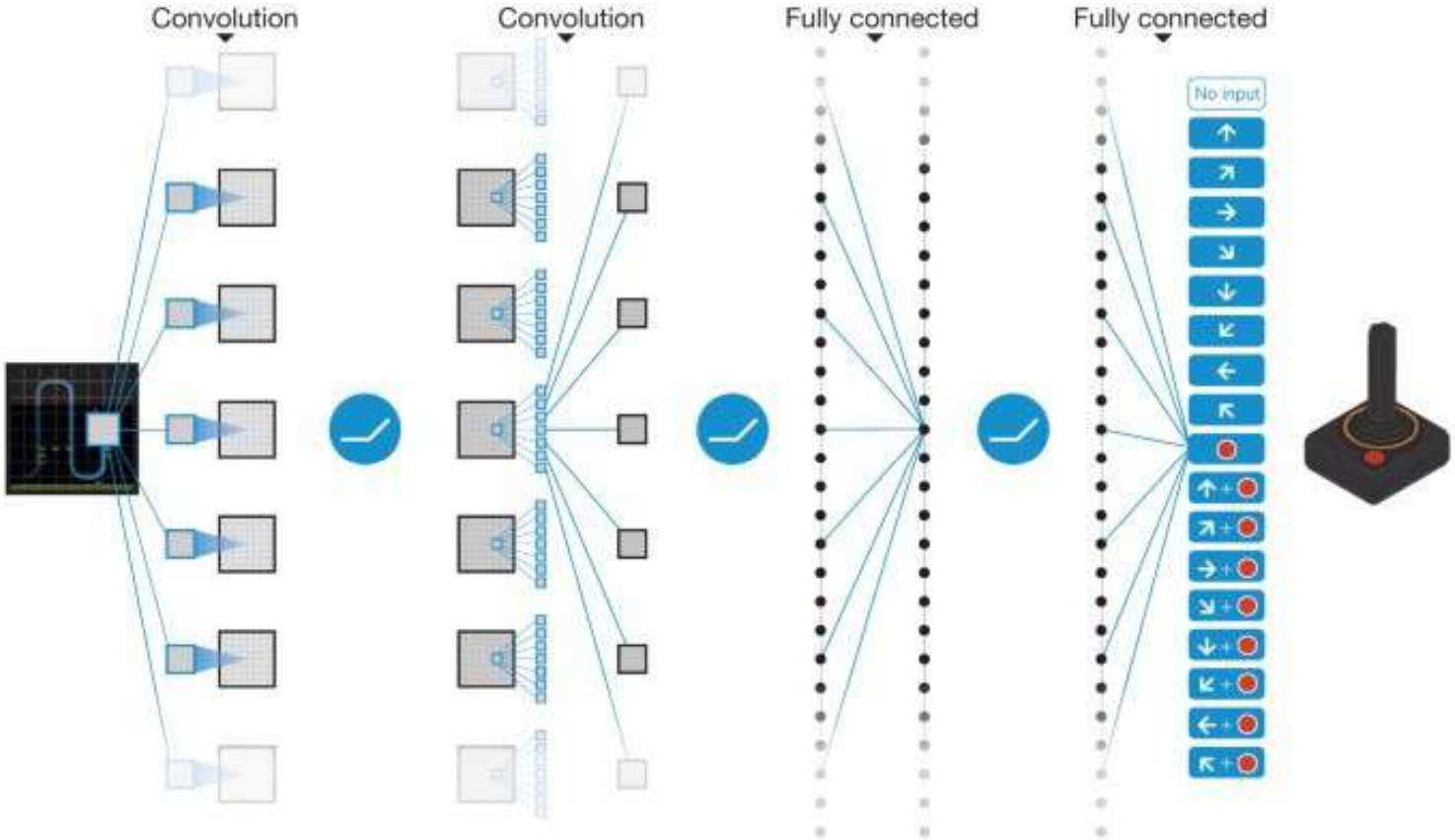
# Deep Q Networks Summary

Use NN to learn Q-function and then use to infer the optimal policy, $\pi(s)$



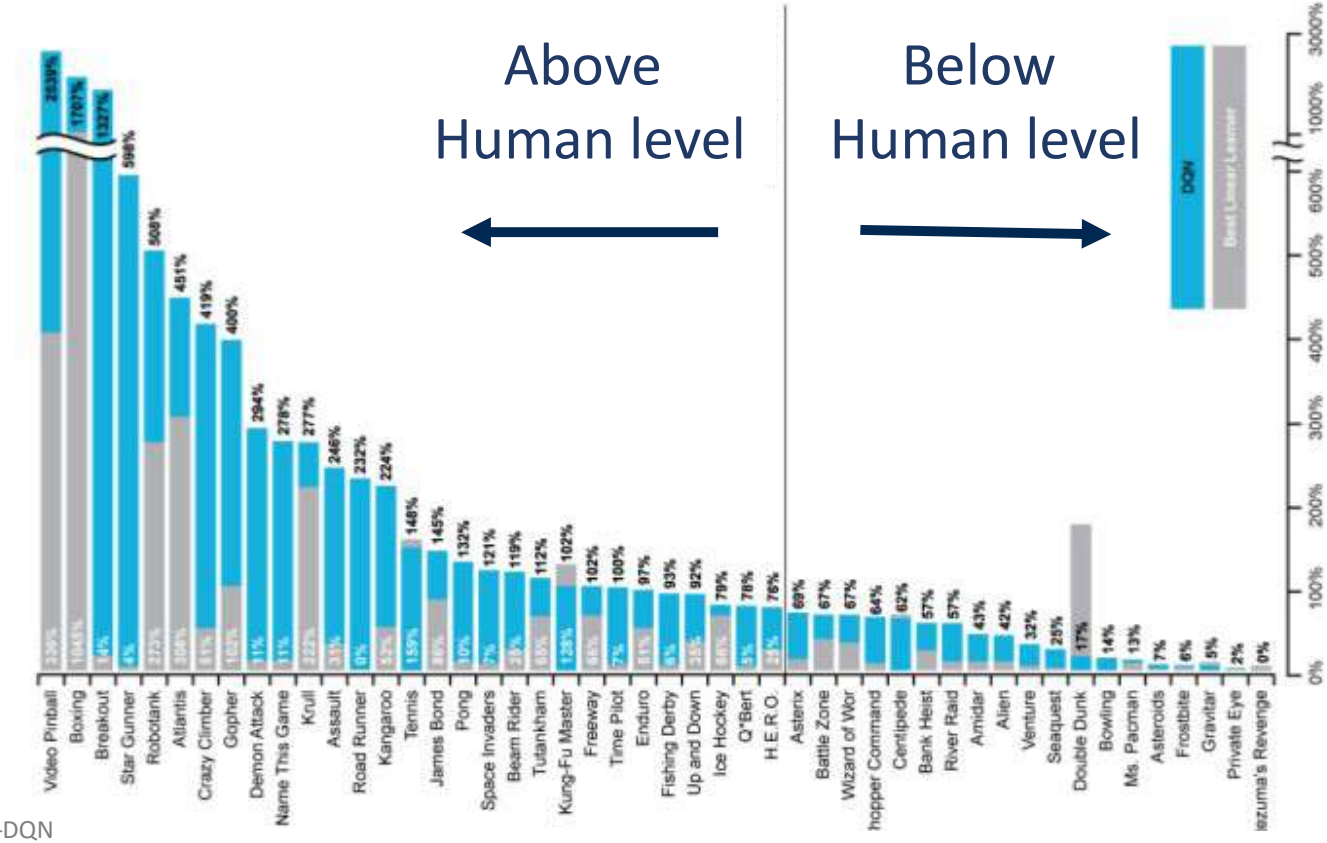

Send action back to the environment and receive next state

# DQN Atari playing Network




Mnih, V., Kavukcuoglu, K., Silver, D. *et al.* Human-level control through deep reinforcement learning. *Nature* **518**, 529–533 (2015). https://doi.org/10.1038/nature14236

# DQN Atari Results




https://github.com/Neo-47/Atari-DQN

## Downsides of Q-learning

Complexity:

- Can model scenarios where the action space is discrete and small
- Cannot handle continuous action spaces

Flexibility:

- Policy is deterministically computed from the Q function by maximizing the reward → cannot learn stochastic policies

**To address these, consider a new class of RL training algorithms:**

**Policy gradient methods**

# Deep Reinforcement Learning Algorithms

## Value Learning

Find $Q(s,a)$

$a = \underset{a}{\operatorname{argmax}}\, Q(s,a)$

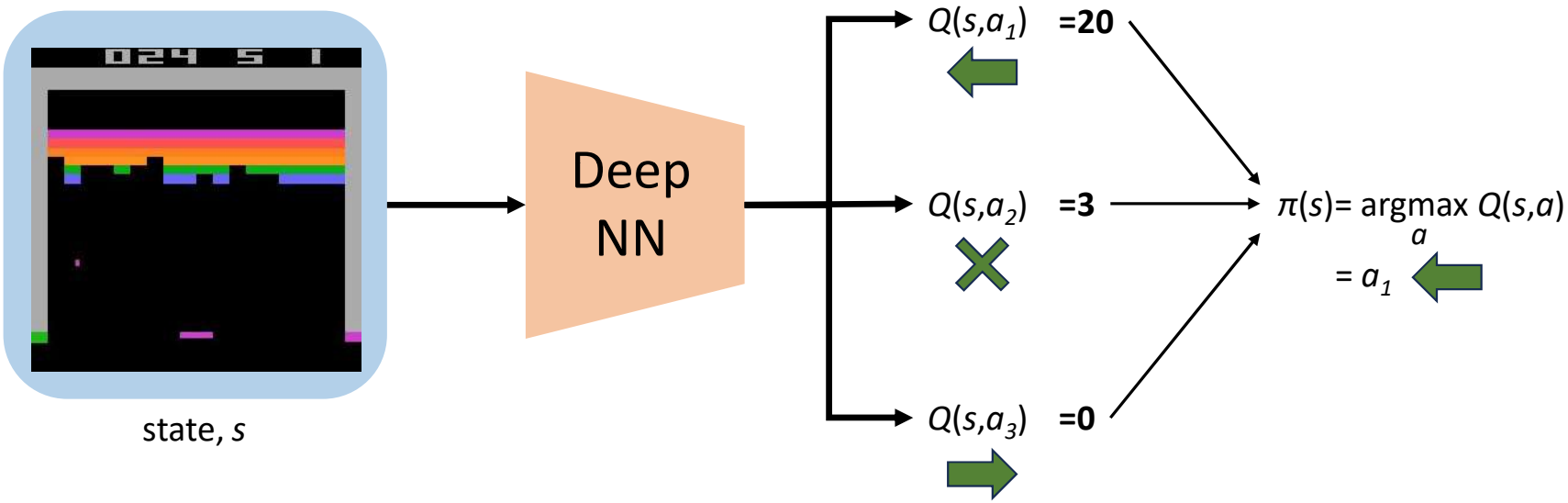
## Policy Learning

Find $\pi(s)$

Sample $a \sim \pi(s)$

# Deep Q Networks Summary

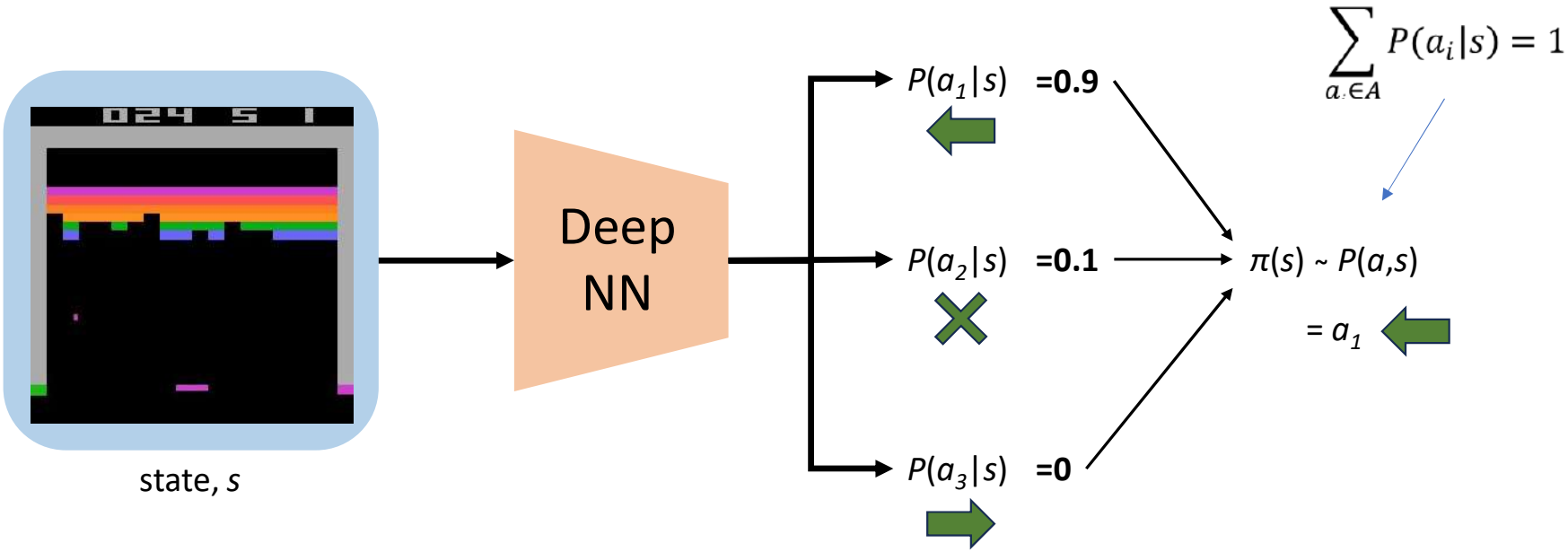
**DQN**: Approximate Q function and use to infer the optimal policy, $\pi$(s)



# Policy Gradient (PG): Key Idea

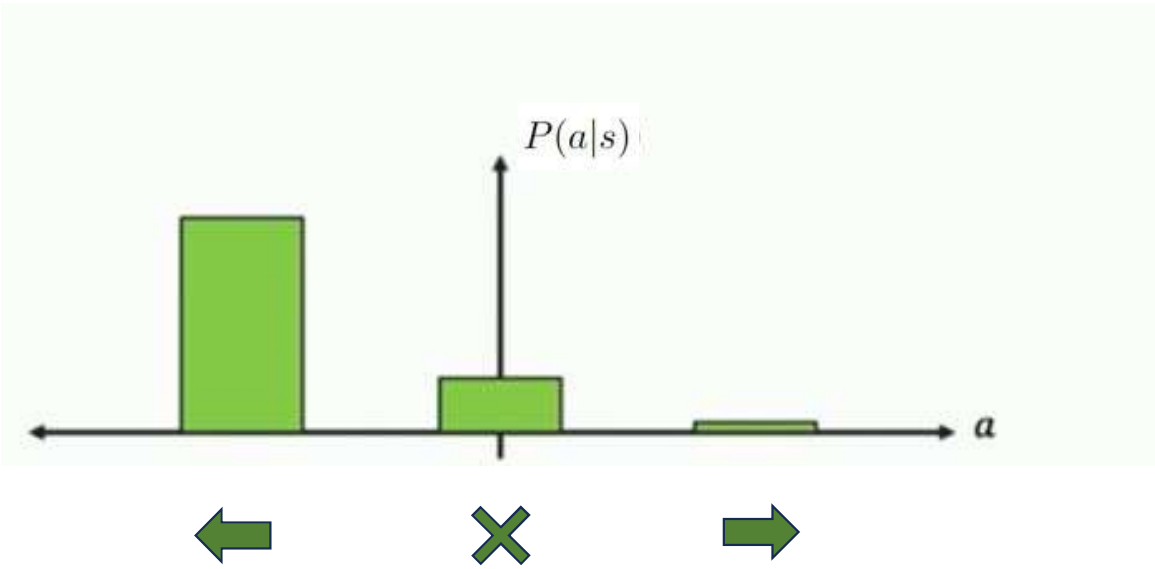
**DQN**: Approximate Q function and use to infer the optimal policy, $\pi$(s)

**Policy Gradient**: Directly optimize the policy $\pi$(s)



# Discrete vs Continous Action Spaces

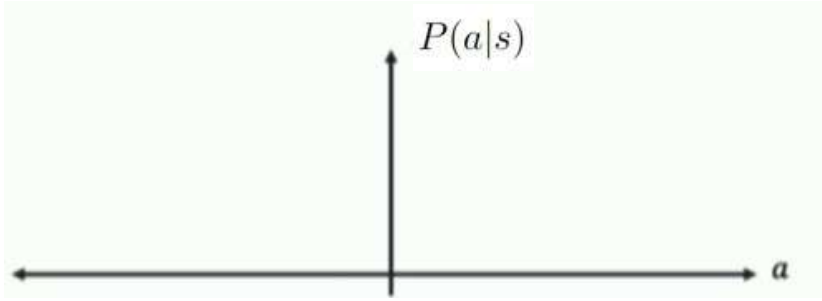
**Discrete action space**: which direction should I move?



# Discrete vs Continous Action Spaces

**Discrete action space**: which direction should I move?

**Continous action space**: how fast should I move? 7 m/s



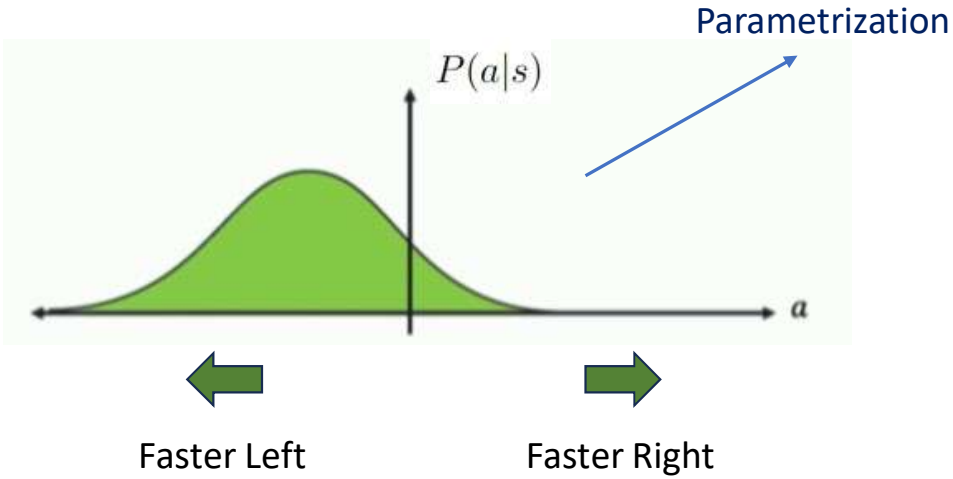

Faster Left

Faster Right

# Discrete vs Continous Action Spaces

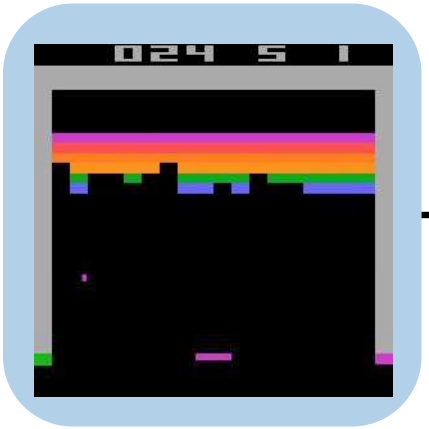
**Discrete action space**: which direction should I move?

**Continous action space**: how fast should I move? 7 m/s



# Policy Gradient (PG): Key Idea

**Policy Gradient:** Enables modeling of continous action space



state, *s*

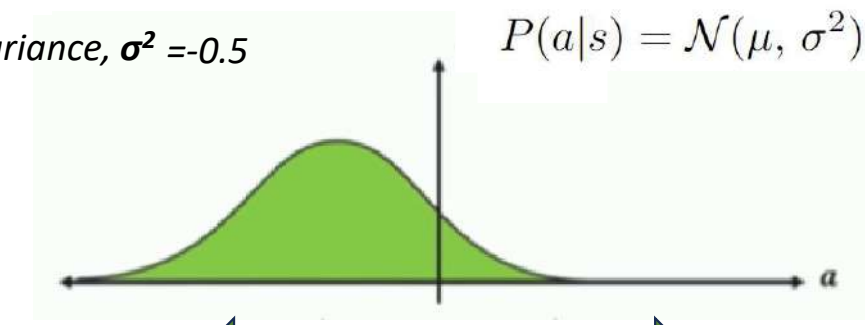
Deep NN

*Mean,* $\boldsymbol{\mu}$ *=-1*

*Variance,* $\boldsymbol{\sigma^2}$ *=-0.5*

$$P(a|s) = \mathcal{N}(\mu, \sigma^2)$$
$$\pi^*(s) \sim P(a|s)$$
$$= -0.8[m/s]$$

$$\int_{a=-\infty}^{\infty} P(a|s) = 1$$




Faster Left

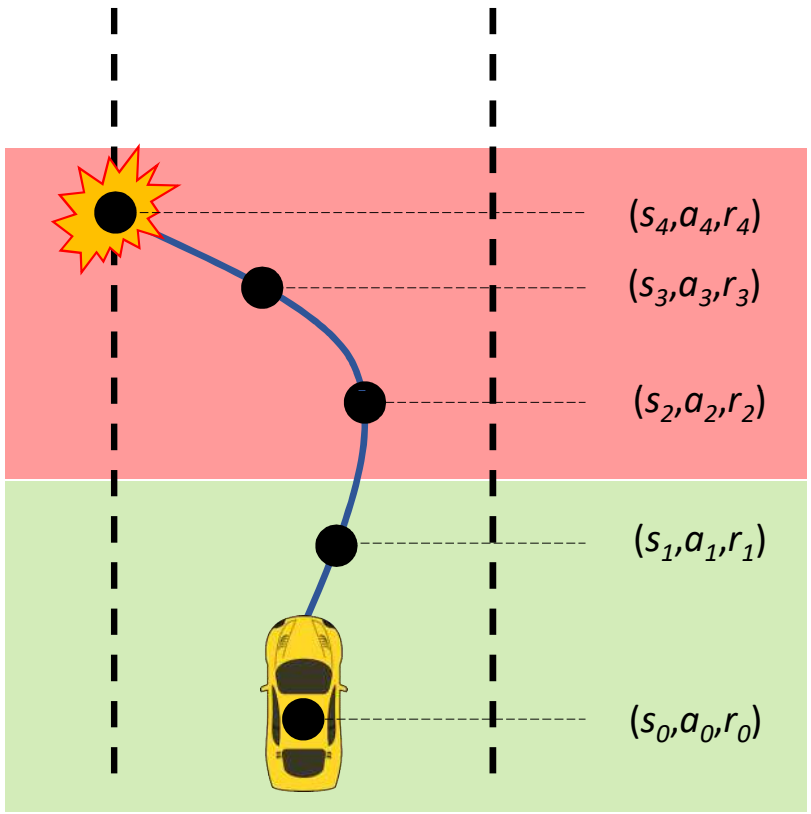
Faster Right

# Trainig Policy Gradients

**Training Algorithm**

1. Initialize the agent
2. Run a policy until termination
3. Record all states, actions, rewards
4. Decrease probability of actions that resulted in low reward
5. Increase probability of actions that resulted in high reward

RUN AGAIN



# Trainig Policy Gradients

**Training Algorithm**

1. Initialize the agent
2. Run a policy until termination
3. Record all states, actions, rewards
4. Decrease probability of actions that resulted in low reward
5. Increase probability of actions that resulted in high reward

Log-likelihood of action

$$\textbf{loss} = -\log P(a_t \mid s_t) R_t$$

reward

**Gradient descent update**

$$w' = w - \nabla \textbf{loss}$$

$$w' = w + \nabla \log P(a_t \mid s_t) R_t$$

Policy gradinet

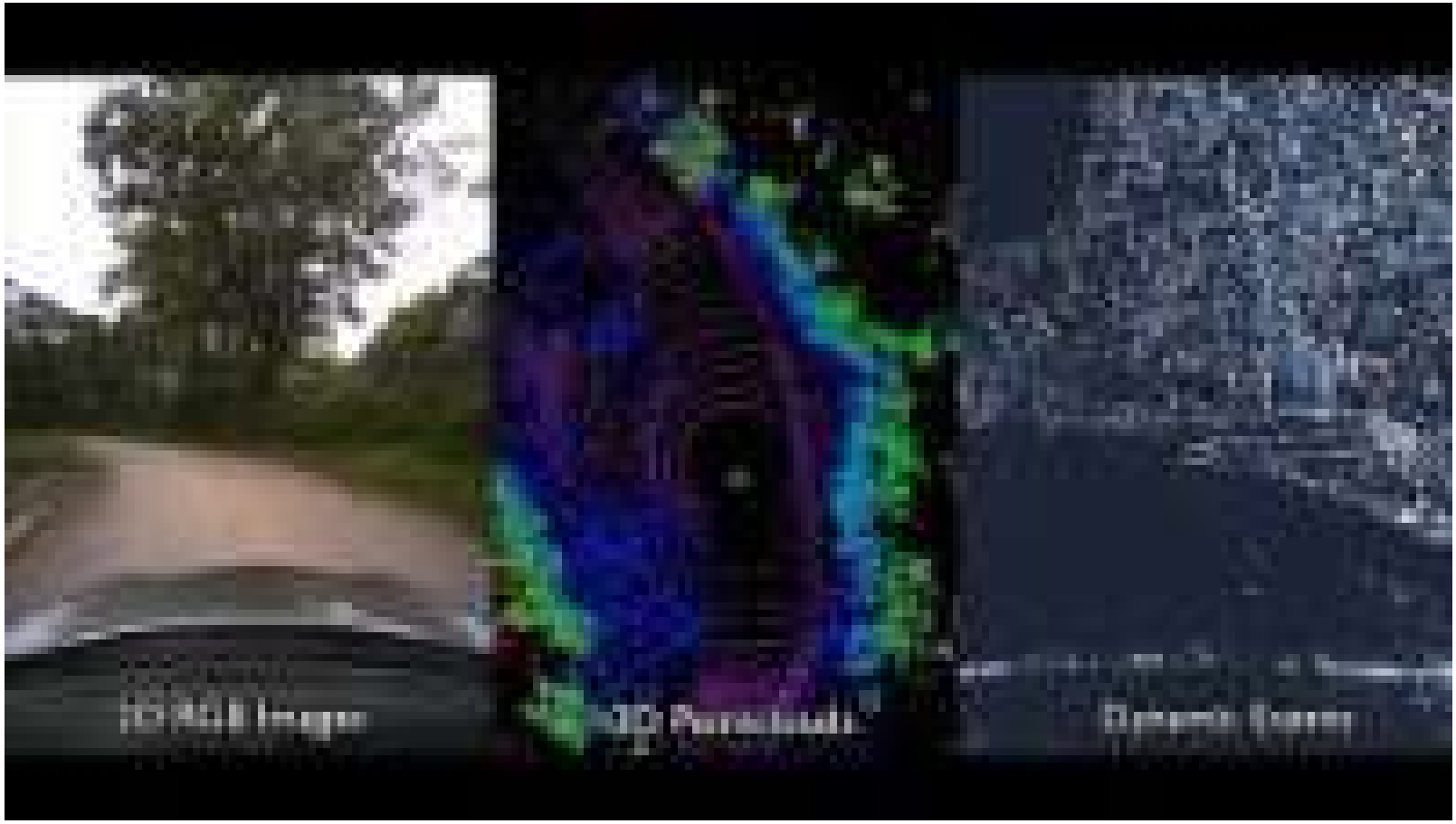
# Reinforcment learning in real life

**SIMULATION**

**Training Algorithm**

1. Initialize the agent
2. Run a policy until termination
3. Record all states, actions, rewards
4. Decrease probability of actions that resulted in low reward
5. Increase probability of actions that resulted in high reward

# VISTA – An open source simulator for self-driving cars



ELTE | FACULTY OF INFORMATICS

# Thank you for your attention!

ELTE | FACULTY OF INFORMATICS

# ETHOROBOTICS

Dr. Balázs Nagy

ELTE | IK
DEPARTMENT OF ARTIFICIAL INTELLIGENCE

BOSCH

# Motivation

# Motivation

- Building a Transformer is the ultimate goal
- Anki's Vector
  - Behaviour as a tool in communication
  - Robots can be treated as a new spices
  - Social robots becoming widespread

# Vector

- Background
  - 2018 - Developed by Anki
  - 2019 - Anki is no longer in business
  - 2019 - Digital Dream Labs purchased
- AI related skills
  - Navigation
    - Path planning
    - Object avoidance
  - Voice recognition (cloud based) - NLP
  - Realtime CNN architecture
    - Person detection
    - Novelty detection
    - Object classification
  - Emotion engine
    - Cat like ’personality’

# Emotions of Vector

# Emotions of Vector

# What is the Goals?

- Integrate robots into the everyday life
- Overcome the communication gap
- Interact with humans
- Help with simple task
  - Weather forecast
  - News
  - Entertainment
- Give the technology a body to connect with people more easily

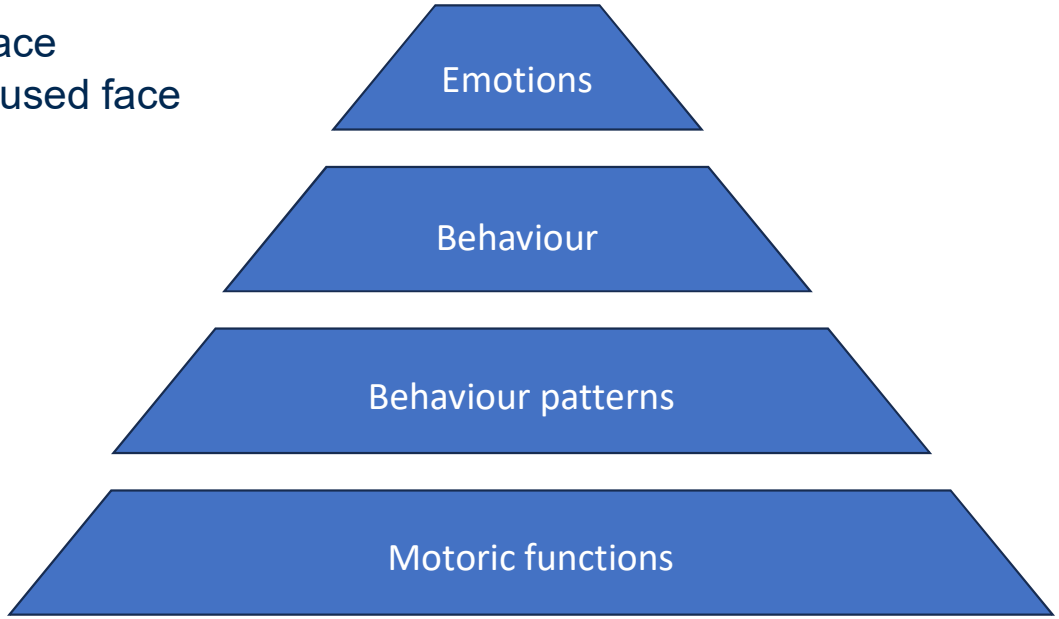
# How?

- Behaviour model
- Social robotics
  - A social robot is an artificial intelligence (AI) system that is designed to interact with humans and other robots.

- Ethology + Robotics = Ethorobotics
  - Ethology = is the scientific study of animal behaviour
  - Robotics = design, construction, and use of machines (robots) to perform tasks

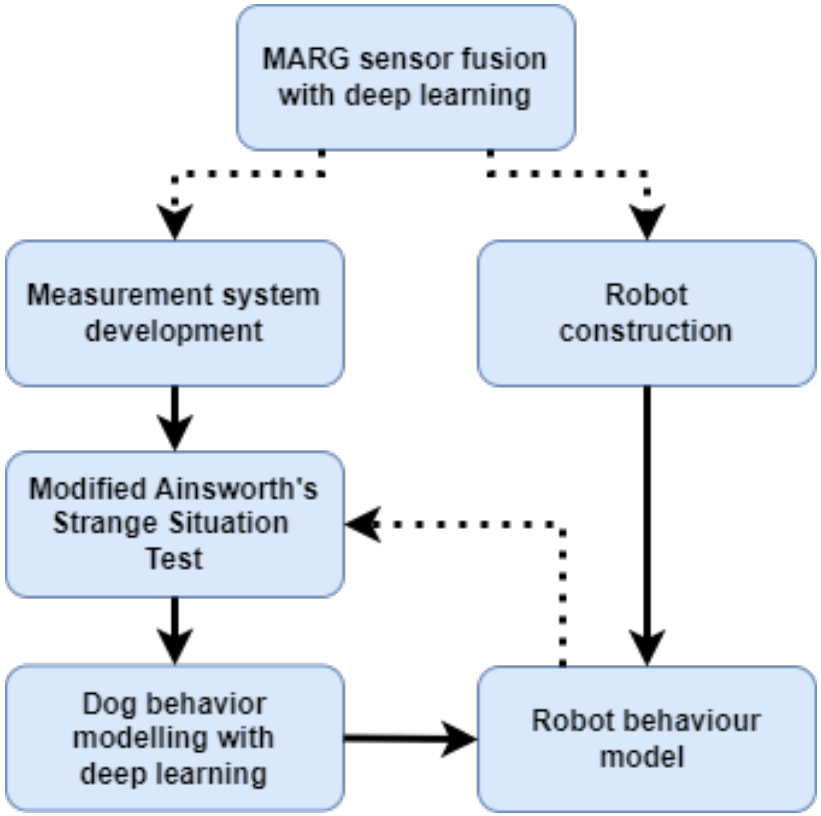
# How?

- Emotion
  - Tune the behaviour
    - Do it fast = Angry + Display Angry face
    - Do it slow = Focused + Display Focused face
- Behaviour
  - Sequence of behaviour patterns
    - Lift up actuator
    - Put down actuator
    - Display face
    - Play sound
- Behaviour pattern
  - Lift up and put down actuator
- Motoric functions
  - Lift up actuator



# Overview

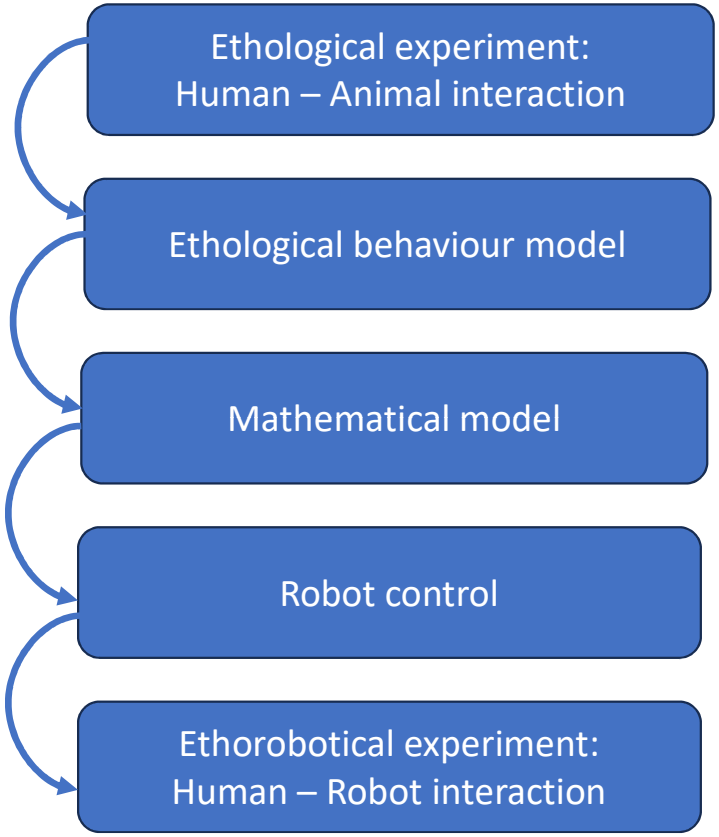
- Ethologically inspired robot behaviour
- Leading questions:
  - How can we measure animals quantitatively?
  - How can we use deep learning to learn animal behaviour pattern?
  - How can we implement animal like behaviour on an autonomous robot?



# Base behaviour model

- Behaviour models based on social animals: dogs
- During domestication, dogs acquired social skills that helped them to integrate into the human environment
  - Communication
  - Cooperation
  - Attachment
- Help dogs
  - Guide dogs for the blind
  - Search and rescue dogy
  - Guarding dogs

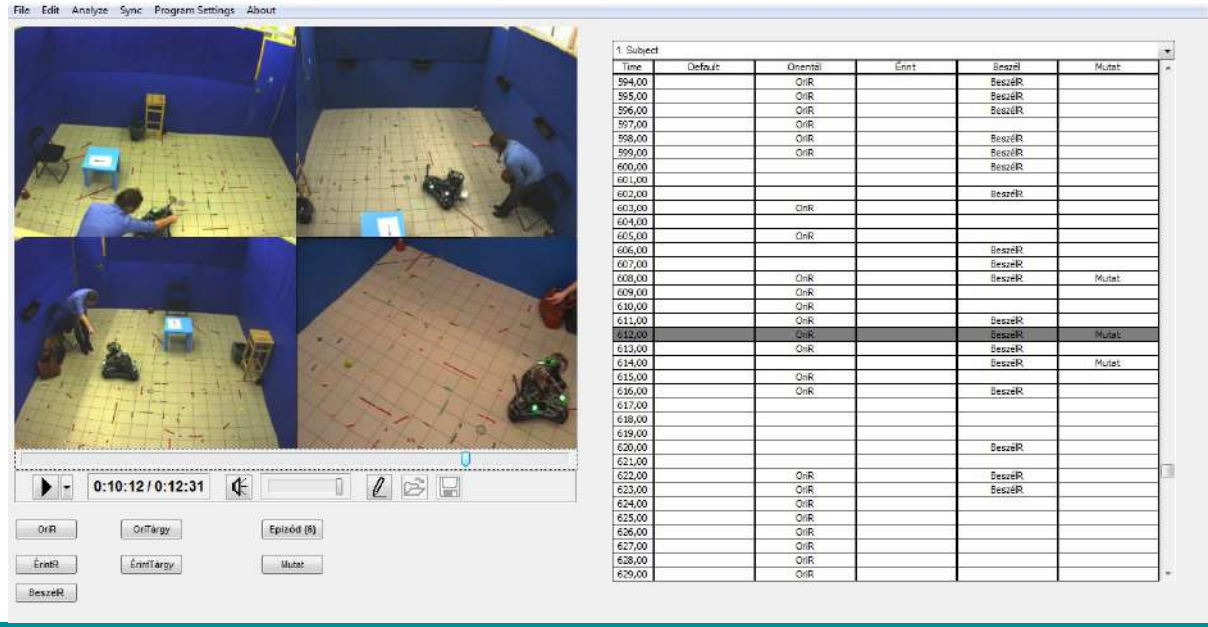
# Research and Development flow



# Ethology research process

- Experiment/observation
- Recording measurements (usually video recordings)
- Behaviour coding
- Data
- Statistics
- Behavioural model

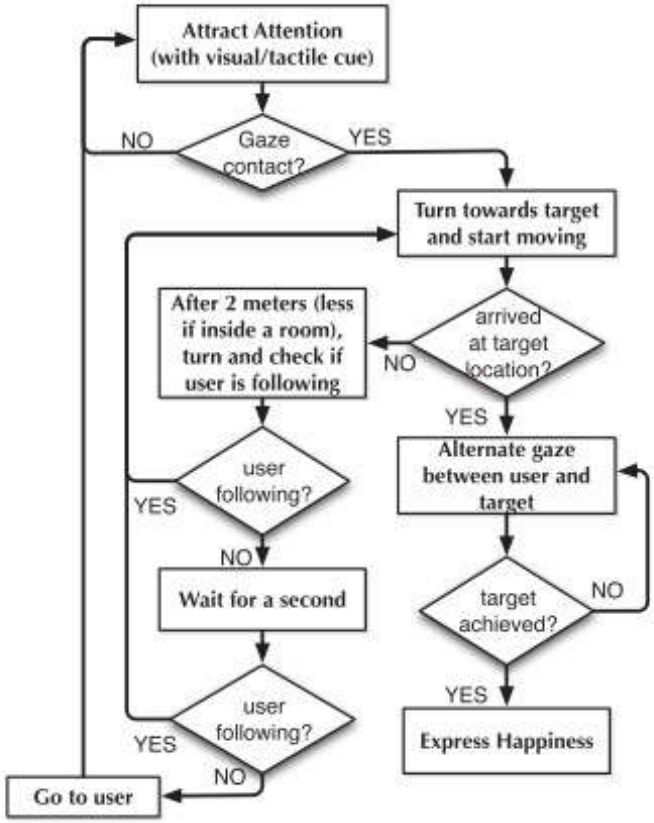
Ethograms



# Observed behaviour

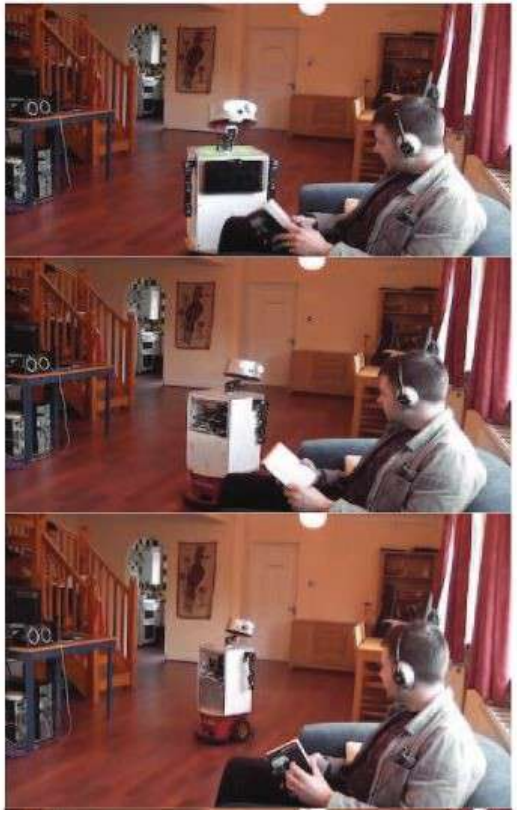
- How long does the dog:
  - Play
  - Wait
    - beside the owner
    - beside the door
  - Explore
- How many times does the dog:
  - Initiate contact
    - With owner
    - With stranger

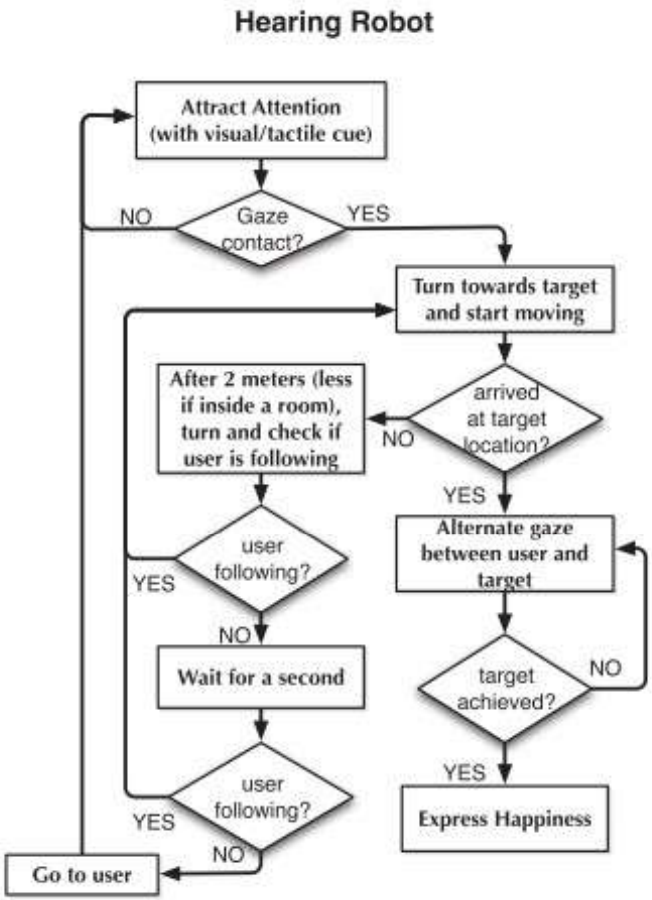
# Leading to sound source



# Leading to sound source

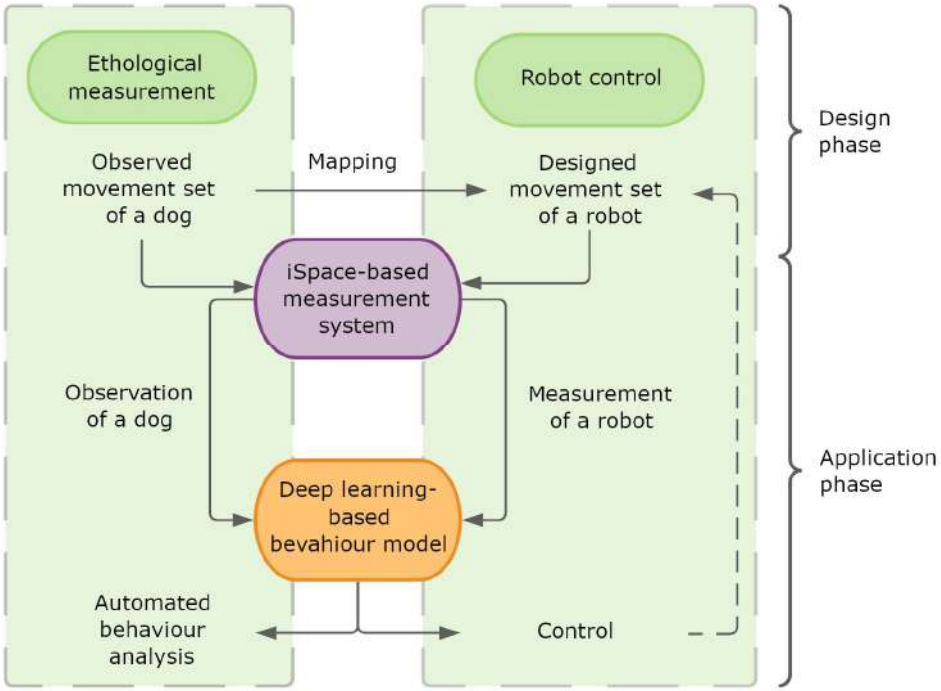
Sunflower robot





# Behaviour Transfer System

- Define an ethological measurement
  - Ainsworth's strange situation test
- Develop a measurement system
  - MoCap (iSpace)
  - Collect data (quality and quantity)
- Design and build a robot
  - Mecanumbot
- Use deep learning to process the data and learn behaviour patterns
- Implement the learned behaviour patterns on the robot

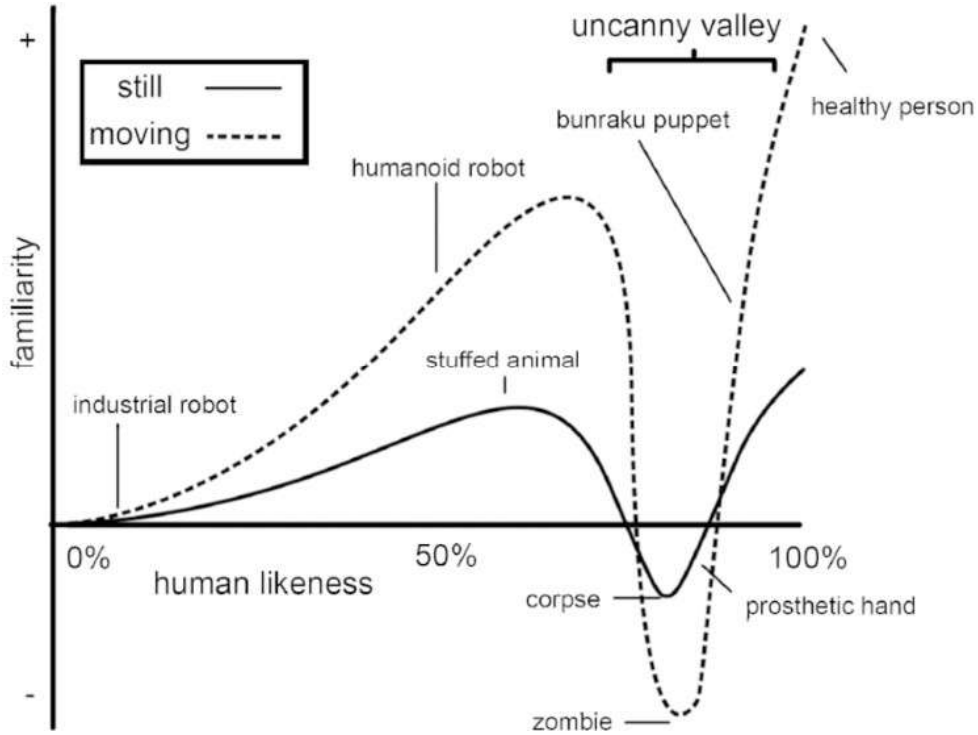


# Robot design approaches

- **Top-down:** breaking down of a system to gain insight into its compositional subsystems in a reverse-engineering fashion. In a top–down approach an overview of the system is formulated, specifying, but not detailing, any first-level subsystems.
- **Bottom-up:** piecing together of systems to give rise to more complex systems, thus making the original systems subsystems of the emergent system.
- Be aware of the „Uncanny Valley” effect
  - Form from function

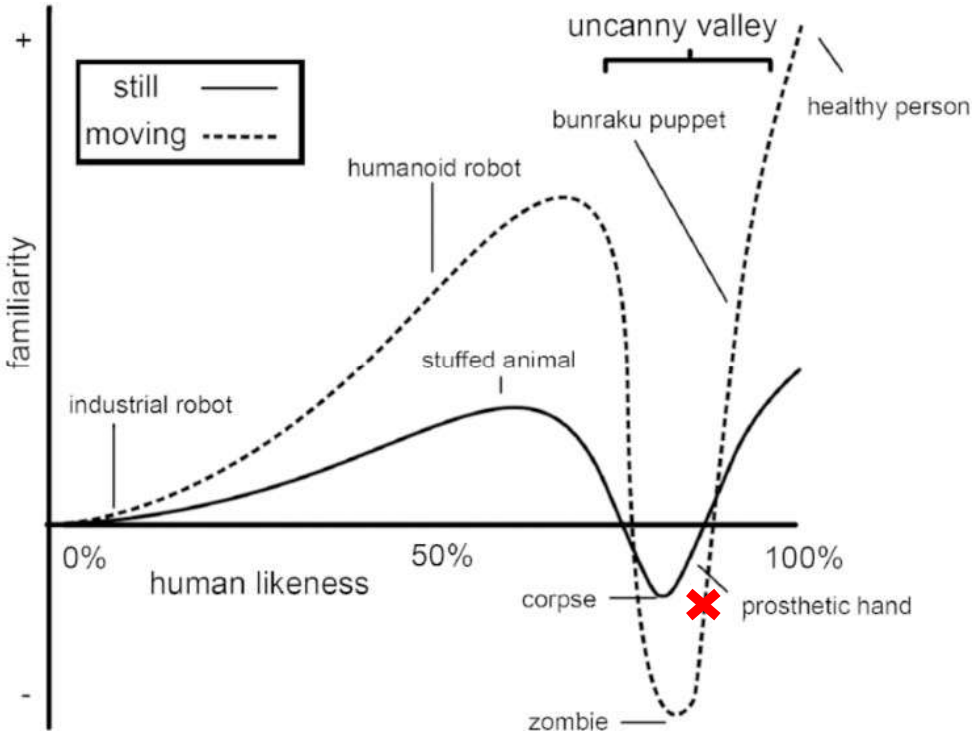
# Uncanny Valley Theory (Masahiro Mori, 1970)

As the appearance of a robot is made more human, some observers' emotional response to the robot becomes increasingly positive and empathetic, until it reaches a point beyond which the response quickly becomes strong revulsion. However, as the robot's appearance continues to become less distinguishable from a human being, the emotional response becomes positive once again and approaches human-to-human empathy levels.



# Uncanny Valley

Hiroshi Ishiguro with Geminoid HI-4
2013, Osaka University



# Feature matching

| | | |
|---|---|---|
| Get his cube, Blackjack | **Play** | Fetch a ball |
| Using worm wheels | **Move** | Using legs |
| Eyes on LCD screen, movements | **Emotion expression** | Complex mimic, tail movement, body language |

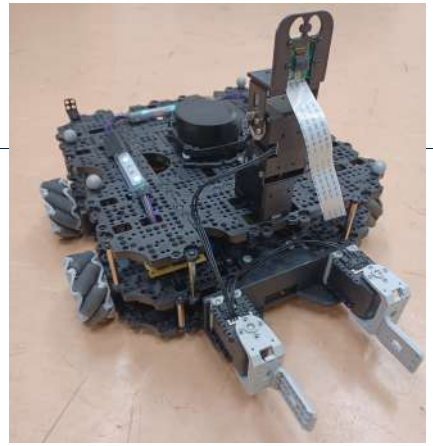
Behaviour elements
social skills

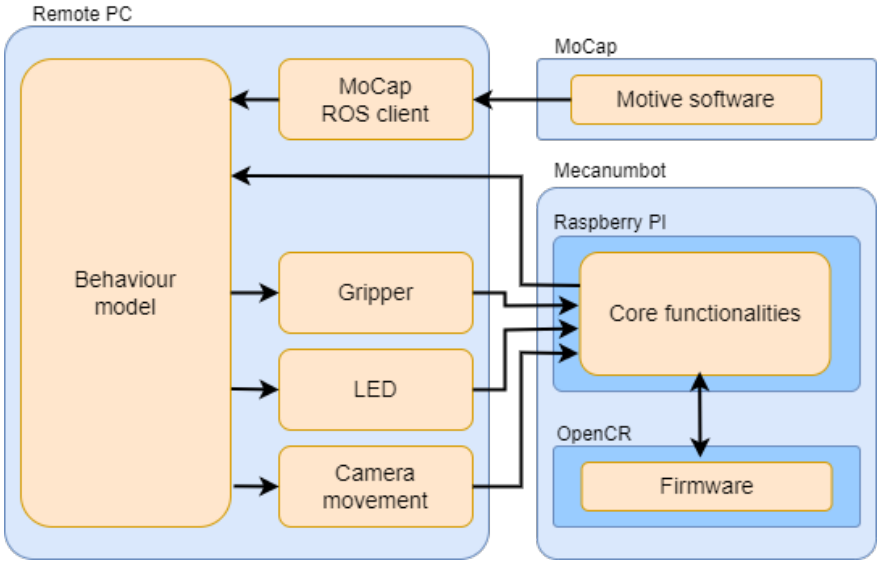
# Ethon

- First version of the ethorobot project
  - Ethon
  - Biscee
- Moduls
  - SLAM
  - Camera
  - MARG sensor
  - Microphone

# Mecanumbot

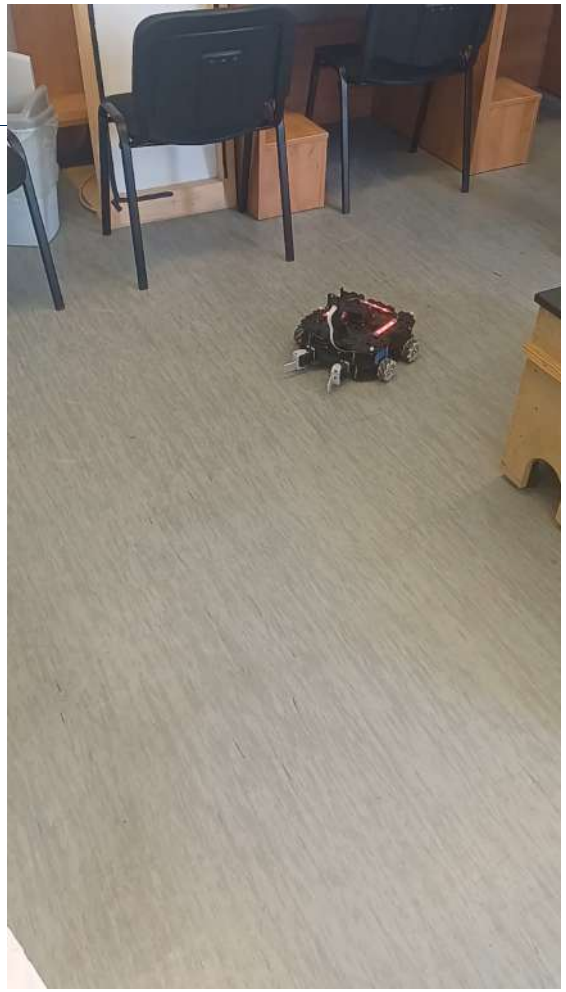
- External intelligence - Remote PC
  - Deep learning-based behaviour model
  - High level robot control
  - Data collection from observer
- Robot – Mecanumbot
  - Motor control – OpenCR
  - Core functionalities – Raspberry PI
  - Dog like features
- External observer – MoCap
  - Marker based position tracking
  - Environment monitoring





# Mecanumbot – Play

- Search the toy
  - Based on colour discrimination
  - Red light
- Find human
  - Using Yolo neural network to identify humans
  - Blue light
- Bring the toy to the human
  - Green light



# Ainsworth test

- Ainswort's strange situation test (Human – Human)
  - The strange situation is a standardized procedure devised by Mary Ainsworth in the 1970s to observe attachment security in children within the context of caregiver relationships.
- Modified Ainsworth test (Human – Dog)
  - The ethologists of ELTE redefined the procedure to observe attachment between a dog and its owner.
- Projected Ainsworth test (Human – Robot)
  - Extend the procedure to examine behaviour between a robot and a human

# Ainsworth's test with a dog

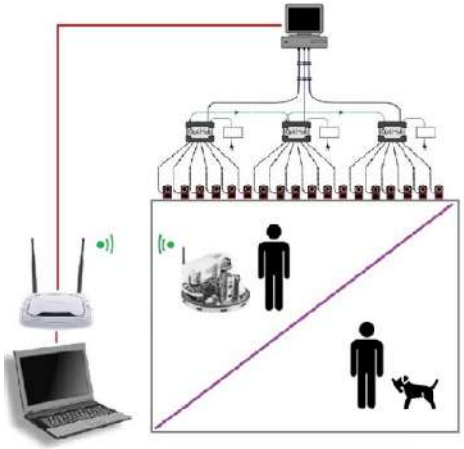
- #1: Acclimatisation
- #2: Introduction to STR
- #3: OWN leaves, first separation
- #4: First reunion with OWN
- #5: Dog alone, second separation
- #6: Separation continuation with STR
- #7: 2nd reunion with OWN
- *Instructions:*
  - *First half of every scenario is passive, the second is active*
  - *Use the dominant hand with the marker set*

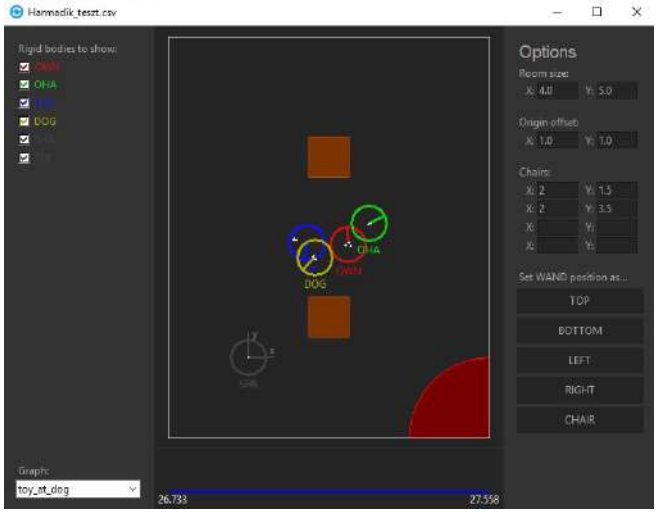
| Episode | Subject | Duration |
|---|---|---|
| 1 | DOG, OWN, TOY | 2 min |
| 2 | DOG, OWN, STR, TOY | 2 min |
| 3 | DOG, STR, TOY | 2 min |
| 4 | DOG, OWN, TOY | 2 min |
| 5 | DOG, TOY | 2 min |
| 6 | DOG, STR, TOY | 2 min |
| 7 | DOG, OWN, TOY | 2 min |

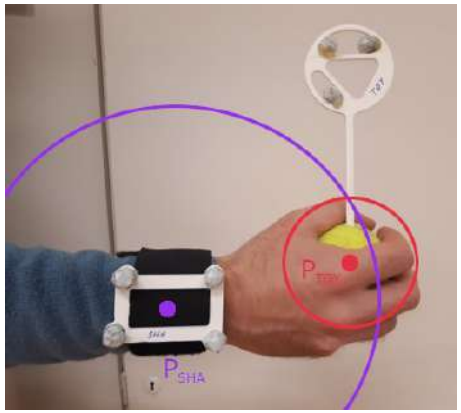
DOG – dog
OWN – owner of the dog
STR – Stranger to the dog
TOY - toy

# MoCap

- Measurement system
  - Contains 18 infra cameras
  - Capable of tracking the position and orientation of marker sets made from infra reflective markers
- Intelligent space
  - Automated measurement
  - Sound controlling the participants

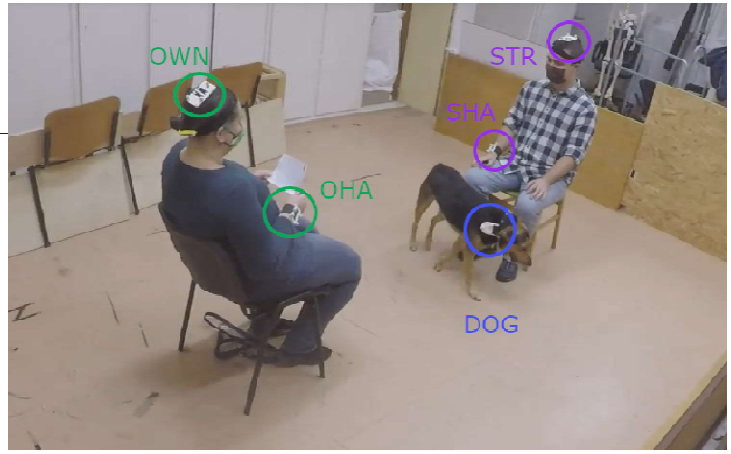


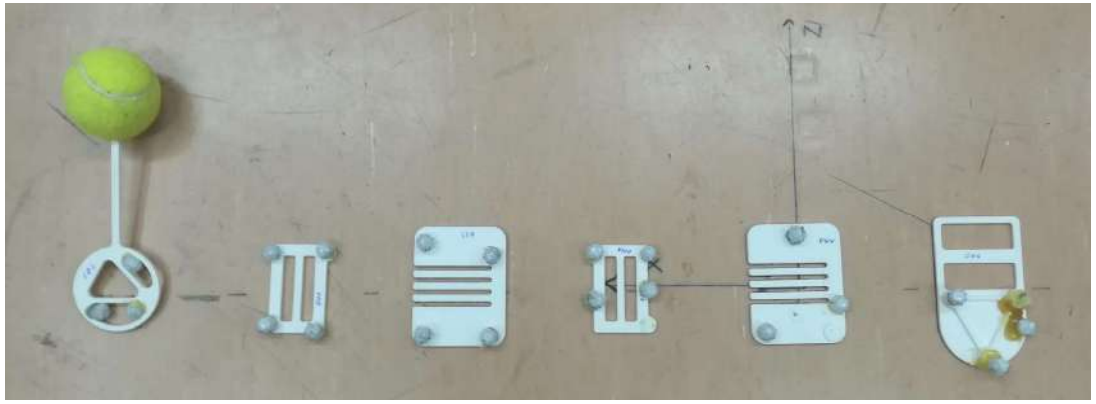


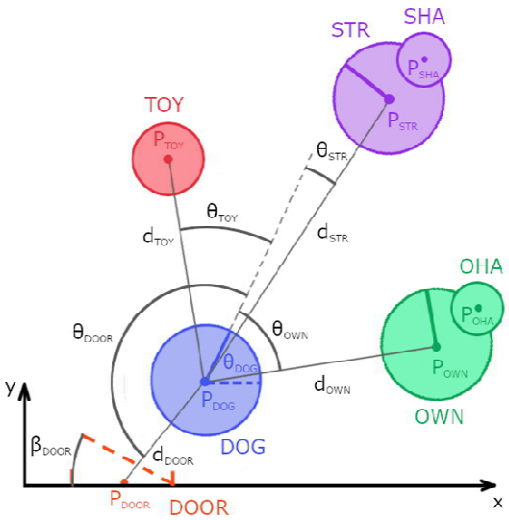
# Modell and Tracking

- 3D printed marker sets
- Infra reflective markers
- At least 3 markers for a set
  - Position tracking
  - Orientation tracking



TOY SHA STR OHA OWN DOG



# Results

- Examined behaviours of dog:
  - Tail wagging
  - Contact seeking
  - Attention
- Neural networks
  - 8-10 hidden layers
  - 10-100 neurons in each layer

| Pattern | Train | Valid | Test |
|---|---|---|---|
| Contact | 99% | 92% | 88% |
| Tail wag | 94% | 88% | 82% |
| Attention | 96% | 74% | 88% |

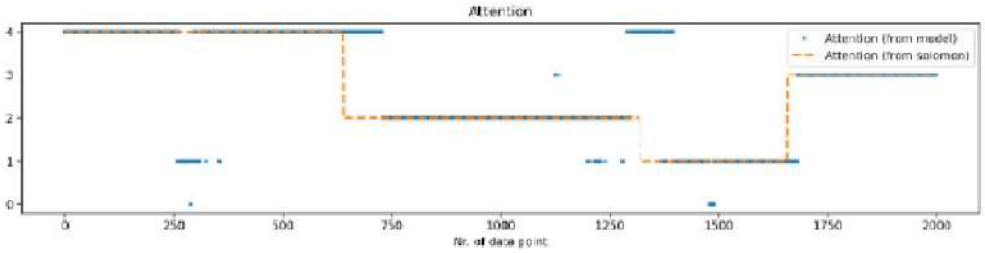



FIGURE 6. Result of attention prediction. (Dog looking at 0: non specified location, 1: owner, 2: stranger, 3: door, 4: toy)

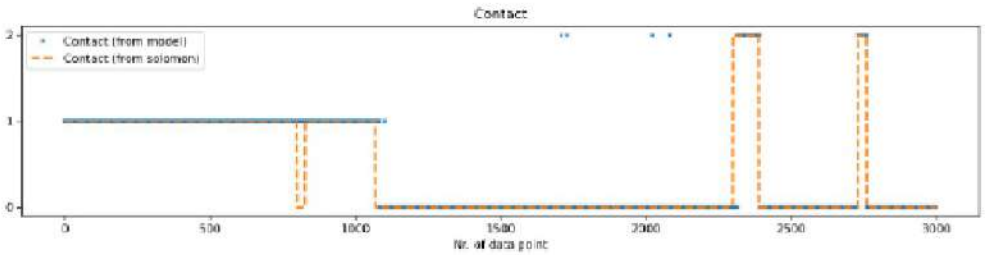



FIGURE 7. Result of contact prediction. (0: No contact, 1: Contact with owner, 2: Contact with stranger)

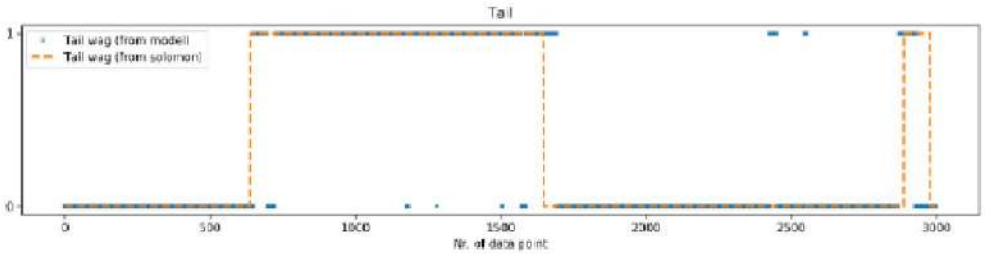



FIGURE 8. Result of tail wag prediction. (0: No tail wag, 1: Tail wag)

ELTE | FACULTY OF INFORMATICS

# Thank you for your attention!

ELTE | FACULTY OF INFORMATICS

# ETHOROBOTICS

Dr. Balázs Nagy

ELTE | IK
DEPARTMENT OF ARTIFICIAL INTELLIGENCE

BOSCH

# Motivation

# Motivation

- Building a Transformer is the ultimate goal
- Anki's Vector
  - Behaviour as a tool in communication
  - Robots can be treated as a new spices
  - Social robots becoming widespread

# Vector

- Background
  - 2018 - Developed by Anki
  - 2019 - Anki is no longer in business
  - 2019 - Digital Dream Labs purchased
- AI related skills
  - Navigation
    - Path planning
    - Object avoidance
  - Voice recognition (cloud based) - NLP
  - Realtime CNN architecture
    - Person detection
    - Novelty detection
    - Object classification
  - Emotion engine
    - Cat like ’personality’

# Emotions of Vector

# Emotions of Vector

# What is the Goals?

- Integrate robots into the everyday life
- Overcome the communication gap
- Interact with humans
- Help with simple task
  - Weather forecast
  - News
  - Entertainment
- Give the technology a body to connect with people more easily

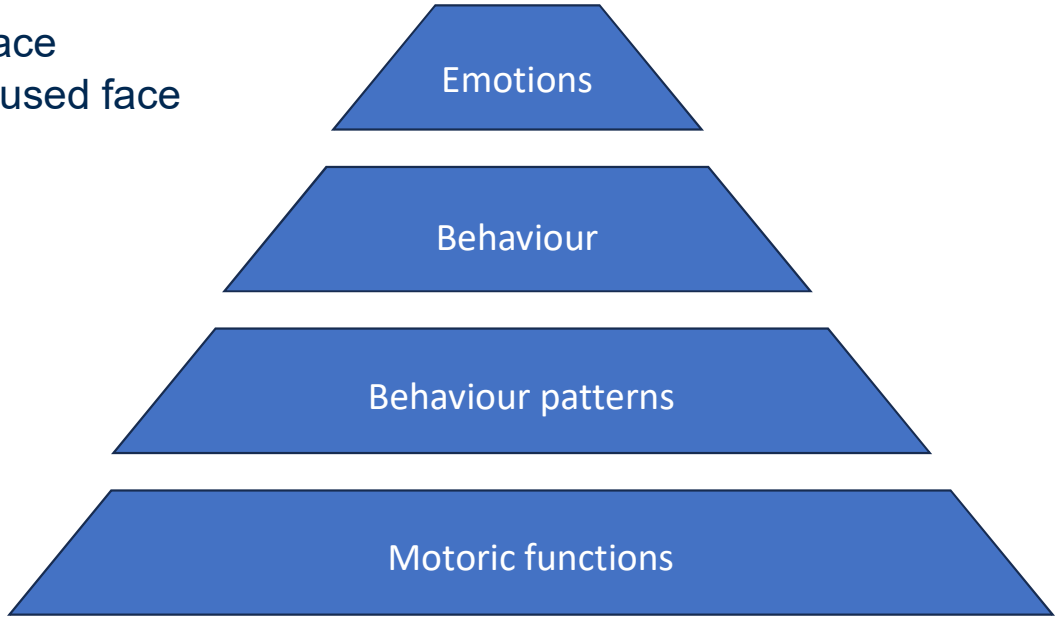
# How?

- Behaviour model
- Social robotics
  - A social robot is an artificial intelligence (AI) system that is designed to interact with humans and other robots.

- Ethology + Robotics = Ethorobotics
  - Ethology = is the scientific study of animal behaviour
  - Robotics = design, construction, and use of machines (robots) to perform tasks

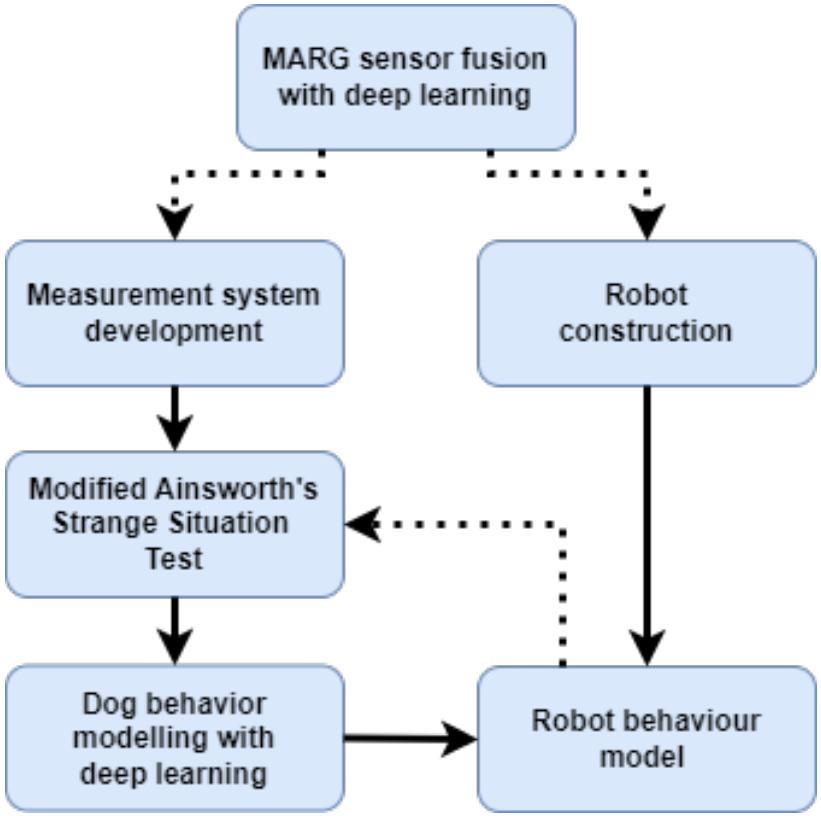
# How?

- Emotion
  - Tune the behaviour
    - Do it fast = Angry + Display Angry face
    - Do it slow = Focused + Display Focused face
- Behaviour
  - Sequence of behaviour patterns
    - Lift up actuator
    - Put down actuator
    - Display face
    - Play sound
- Behaviour pattern
  - Lift up and put down actuator
- Motoric functions
  - Lift up actuator



# Overview

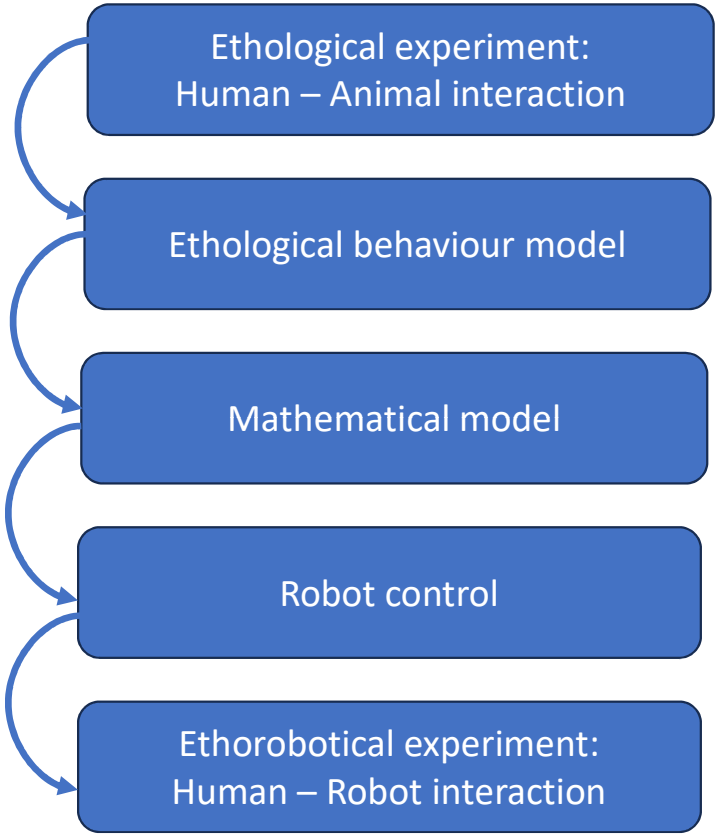
- Ethologically inspired robot behaviour
- Leading questions:
  - How can we measure animals quantitatively?
  - How can we use deep learning to learn animal behaviour pattern?
  - How can we implement animal like behaviour on an autonomous robot?



# Base behaviour model

- Behaviour models based on social animals: dogs
- During domestication, dogs acquired social skills that helped them to integrate into the human environment
  - Communication
  - Cooperation
  - Attachment
- Help dogs
  - Guide dogs for the blind
  - Search and rescue dogy
  - Guarding dogs

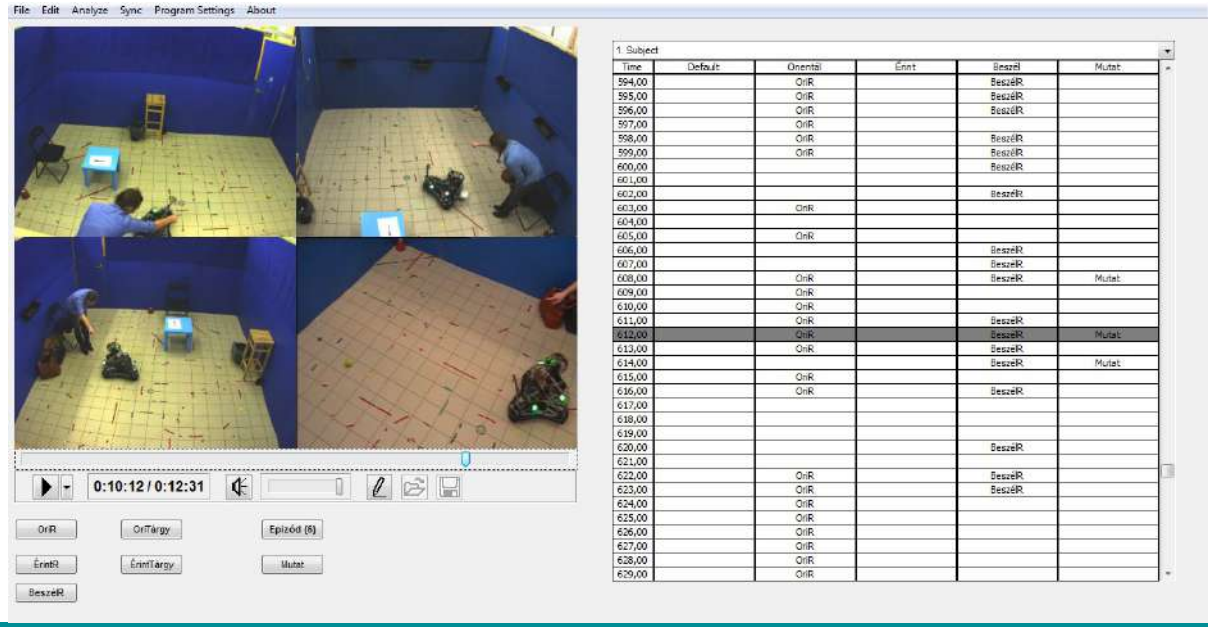
# Research and Development flow



# Ethology research process

- Experiment/observation
- Recording measurements (usually video recordings)
- Behaviour coding
- Data
- Statistics
- Behavioural model

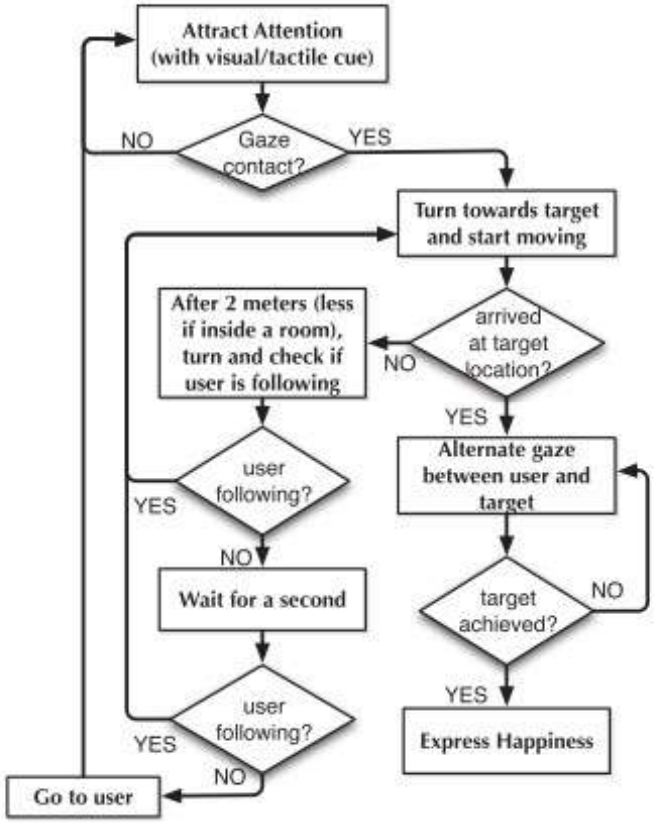
Ethograms



# Observed behaviour

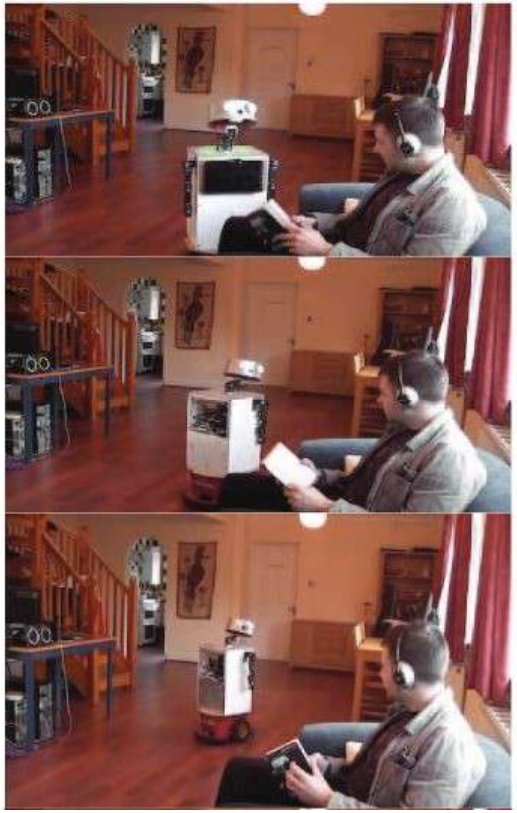
- How long does the dog:
  - Play
  - Wait
    - beside the owner
    - beside the door
  - Explore
- How many times does the dog:
  - Initiate contact
    - With owner
    - With stranger

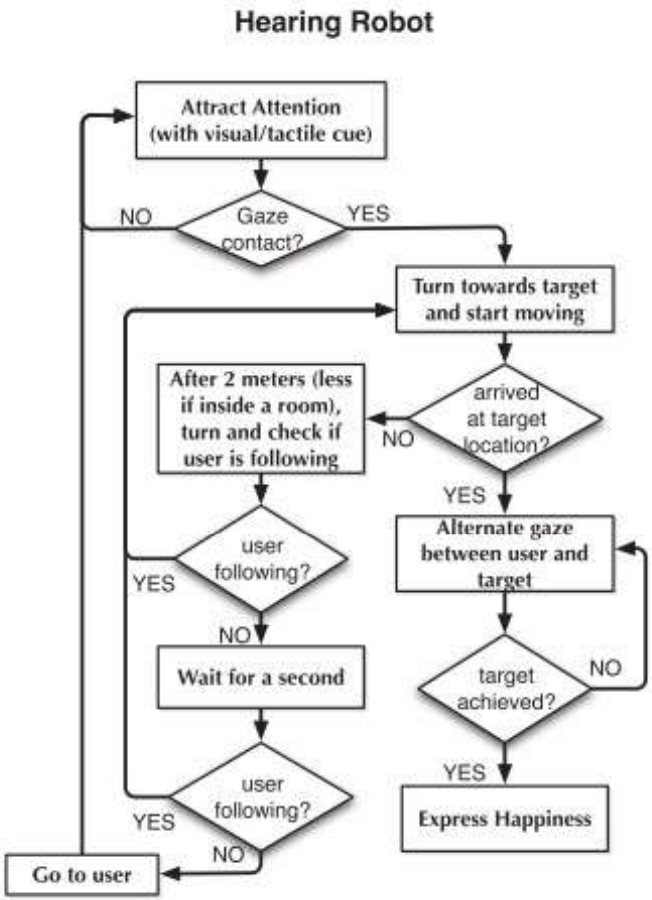
# Leading to sound source



# Leading to sound source

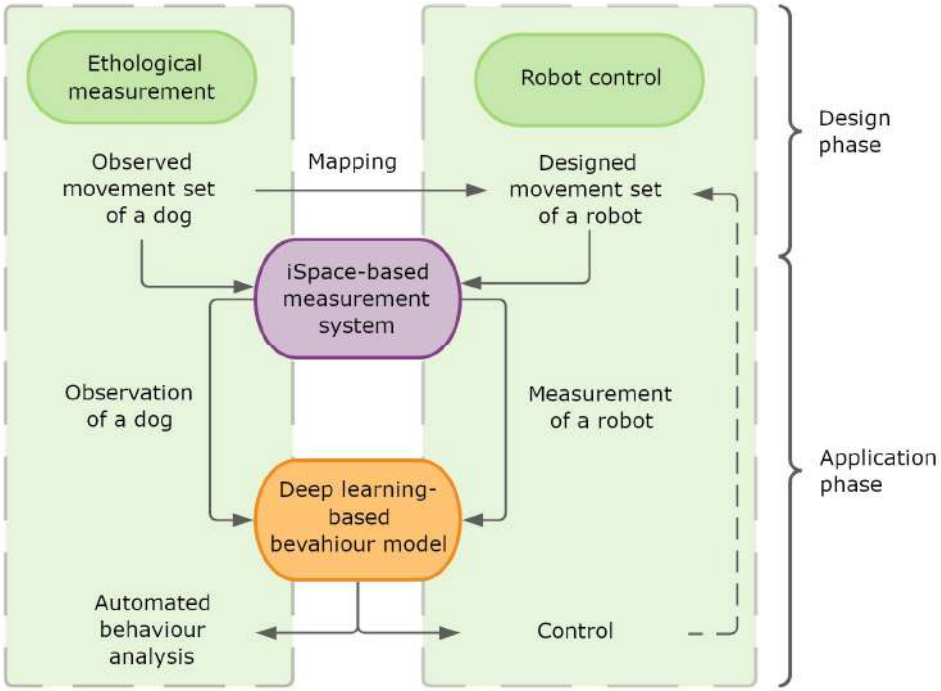
Sunflower robot





# Behaviour Transfer System

- Define an ethological measurement
  - Ainsworth's strange situation test
- Develop a measurement system
  - MoCap (iSpace)
  - Collect data (quality and quantity)
- Design and build a robot
  - Mecanumbot
- Use deep learning to process the data and learn behaviour patterns
- Implement the learned behaviour patterns on the robot

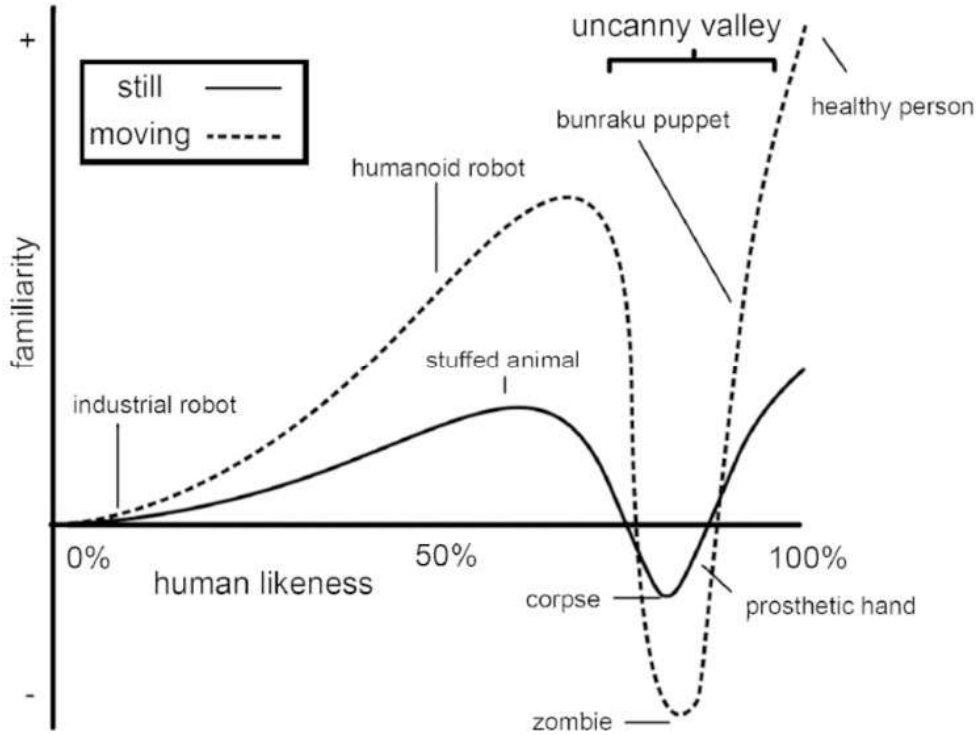


# Robot design approaches

- **Top-down:** breaking down of a system to gain insight into its compositional subsystems in a reverse-engineering fashion. In a top–down approach an overview of the system is formulated, specifying, but not detailing, any first-level subsystems.
- **Bottom-up:** piecing together of systems to give rise to more complex systems, thus making the original systems subsystems of the emergent system.
- Be aware of the „Uncanny Valley” effect
  - Form from function

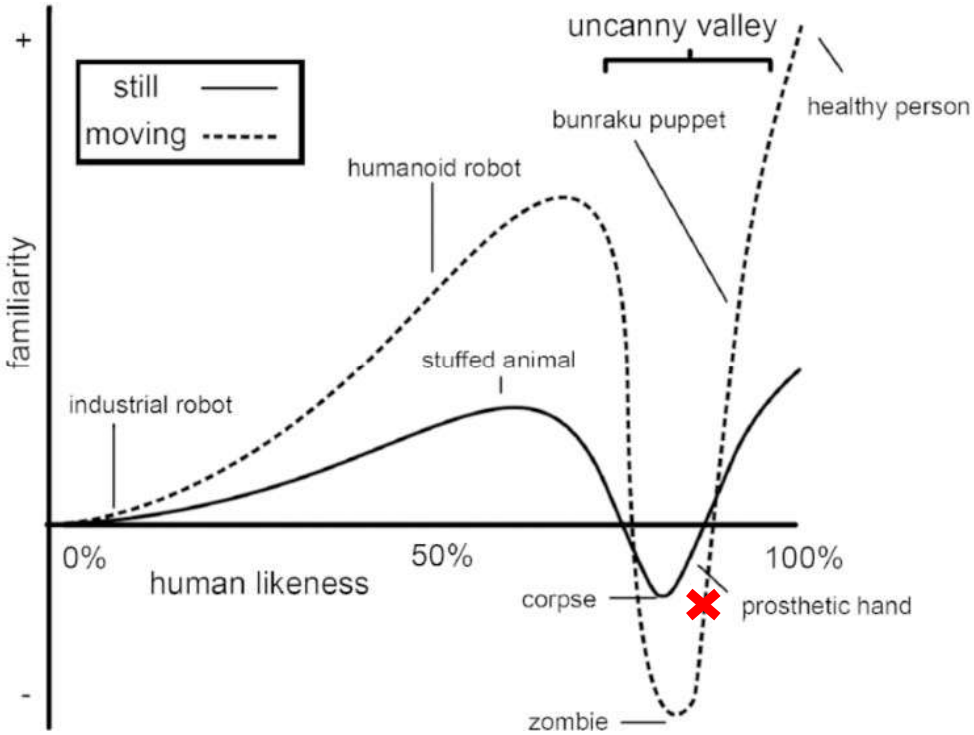
# Uncanny Valley Theory (Masahiro Mori, 1970)

As the appearance of a robot is made more human, some observers' emotional response to the robot becomes increasingly positive and empathetic, until it reaches a point beyond which the response quickly becomes strong revulsion. However, as the robot's appearance continues to become less distinguishable from a human being, the emotional response becomes positive once again and approaches human-to-human empathy levels.



# Uncanny Valley

Hiroshi Ishiguro with Geminoid HI-4
2013, Osaka University



# Feature matching

| | | |
|---|---|---|
| Get his cube, Blackjack | **Play** | Fetch a ball |
| Using worm wheels | **Move** | Using legs |
| Eyes on LCD screen, movements | **Emotion expression** | Complex mimic, tail movement, body language |

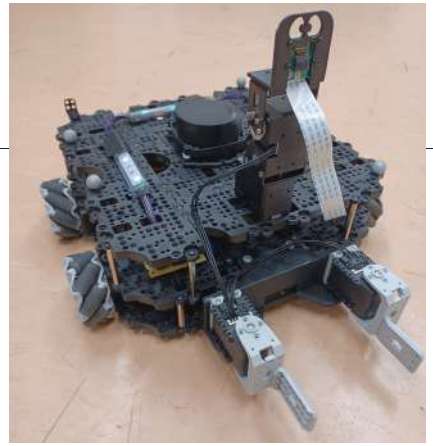
Behaviour elements
social skills

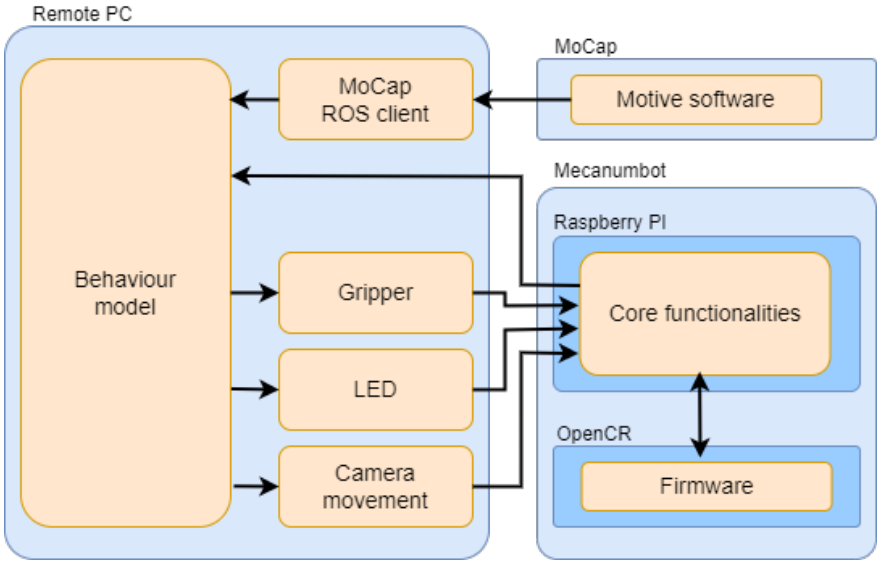
# Ethon

- First version of the ethorobot project
  - Ethon
  - Biscee
- Moduls
  - SLAM
  - Camera
  - MARG sensor
  - Microphone

# Mecanumbot

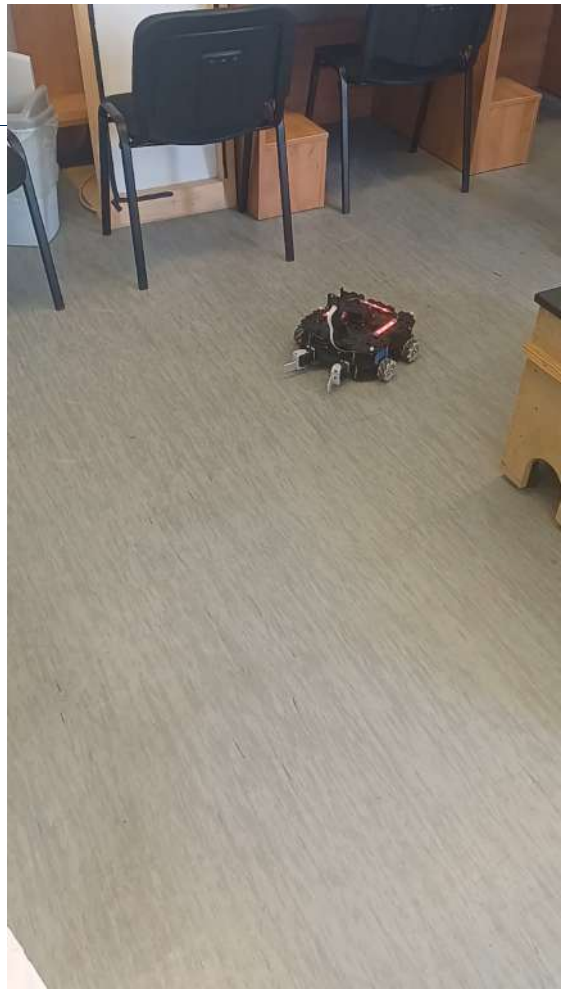
- External intelligence - Remote PC
  - Deep learning-based behaviour model
  - High level robot control
  - Data collection from observer
- Robot – Mecanumbot
  - Motor control – OpenCR
  - Core functionalities – Raspberry PI
  - Dog like features
- External observer – MoCap
  - Marker based position tracking
  - Environment monitoring





# Mecanumbot – Play

- Search the toy
  - Based on colour discrimination
  - Red light
- Find human
  - Using Yolo neural network to identify humans
  - Blue light
- Bring the toy to the human
  - Green light



# Ainsworth test

- Ainswort's strange situation test (Human – Human)
  - The strange situation is a standardized procedure devised by Mary Ainsworth in the 1970s to observe attachment security in children within the context of caregiver relationships.
- Modified Ainsworth test (Human – Dog)
  - The ethologists of ELTE redefined the procedure to observe attachment between a dog and its owner.
- Projected Ainsworth test (Human – Robot)
  - Extend the procedure to examine behaviour between a robot and a human

# Ainsworth's test with a dog

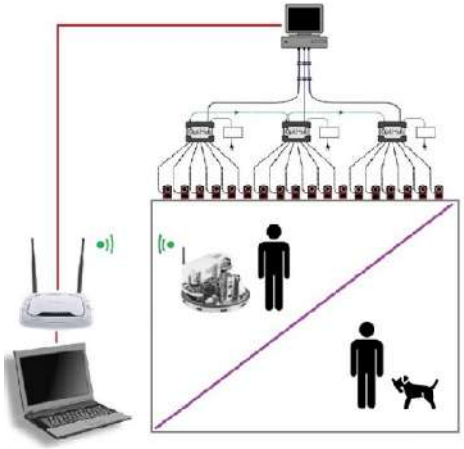
- #1: Acclimatisation
- #2: Introduction to STR
- #3: OWN leaves, first separation
- #4: First reunion with OWN
- #5: Dog alone, second separation
- #6: Separation continuation with STR
- #7: 2nd reunion with OWN
- *Instructions:*
  - *First half of every scenario is passive, the second is active*
  - *Use the dominant hand with the marker set*

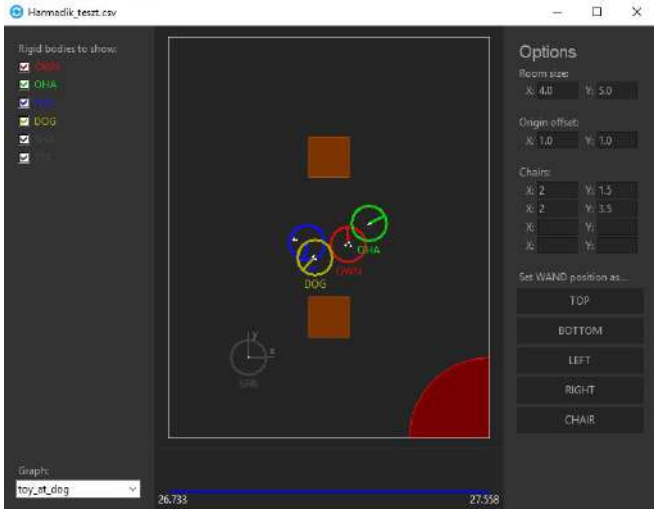
| Episode | Subject | Duration |
|---|---|---|
| 1 | DOG, OWN, TOY | 2 min |
| 2 | DOG, OWN, STR, TOY | 2 min |
| 3 | DOG, STR, TOY | 2 min |
| 4 | DOG, OWN, TOY | 2 min |
| 5 | DOG, TOY | 2 min |
| 6 | DOG, STR, TOY | 2 min |
| 7 | DOG, OWN, TOY | 2 min |

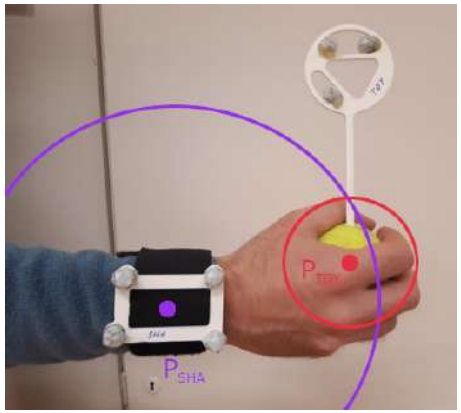
DOG – dog
OWN – owner of the dog
STR – Stranger to the dog
TOY - toy

# MoCap

- Measurement system
  - Contains 18 infra cameras
  - Capable of tracking the position and orientation of marker sets made from infra reflective markers
- Intelligent space
  - Automated measurement
  - Sound controlling the participants

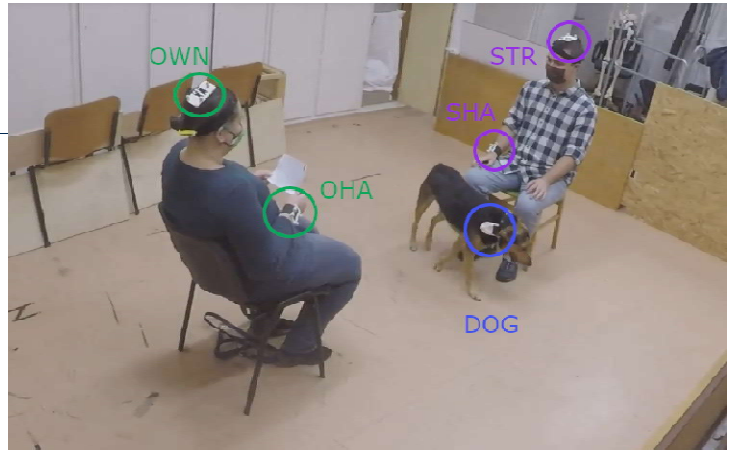


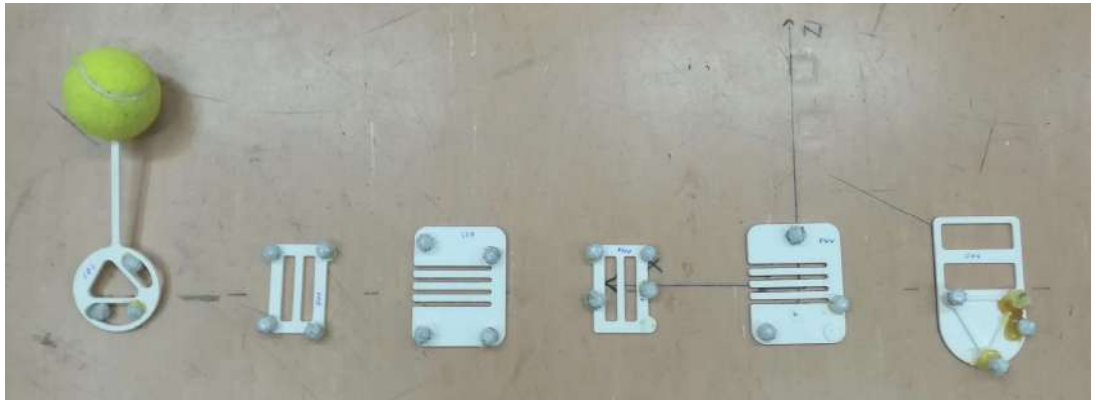


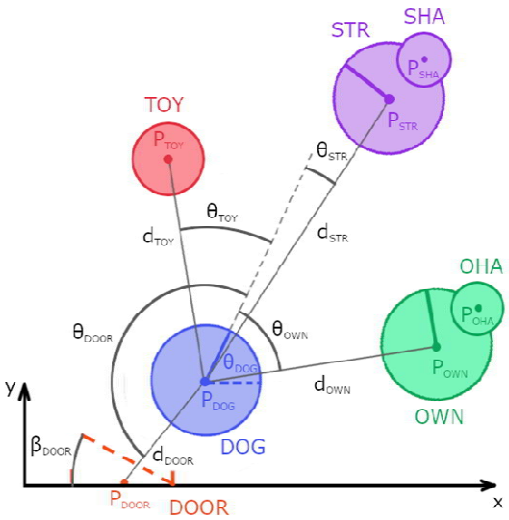
# Modell and Tracking

- 3D printed marker sets
- Infra reflective markers
- At least 3 markers for a set
  - Position tracking
  - Orientation tracking



TOY SHA STR OHA OWN DOG



# Results

- Examined behaviours of dog:
  - Tail wagging
  - Contact seeking
  - Attention
- Neural networks
  - 8-10 hidden layers
  - 10-100 neurons in each layer

| Pattern | Train | Valid | Test |
|---|---|---|---|
| Contact | 99% | 92% | 88% |
| Tail wag | 94% | 88% | 82% |
| Attention | 96% | 74% | 88% |

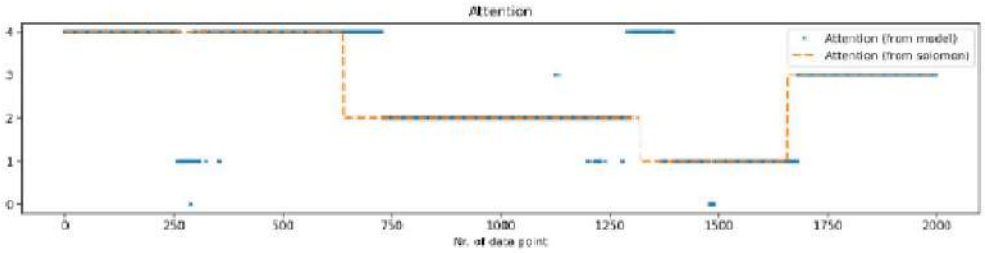



FIGURE 6. Result of attention prediction. (Dog looking at 0: non specified location, 1: owner, 2: stranger, 3: door, 4: toy)

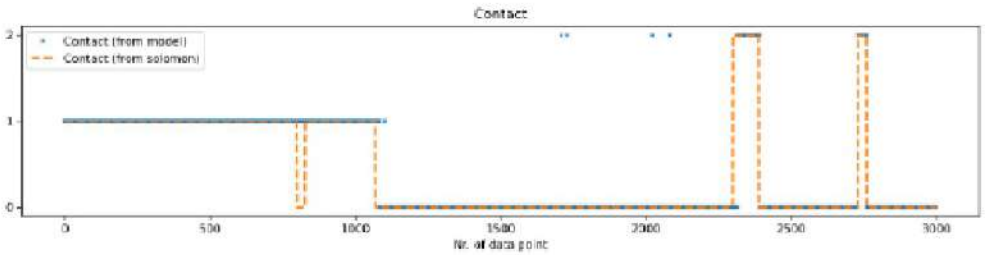



FIGURE 7. Result of contact prediction. (0: No contact, 1: Contact with owner, 2: Contact with stranger)

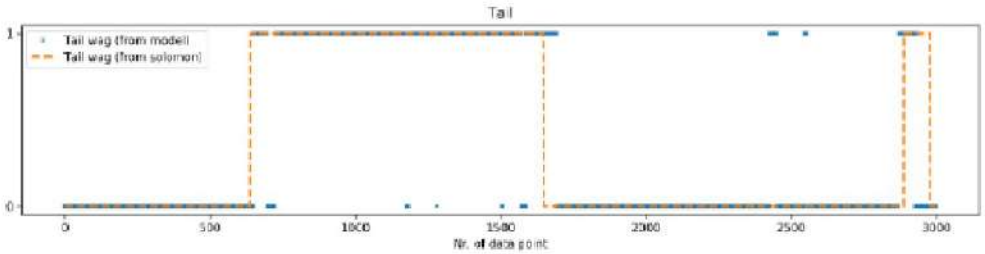



FIGURE 8. Result of tail wag prediction. (0: No tail wag, 1: Tail wag)

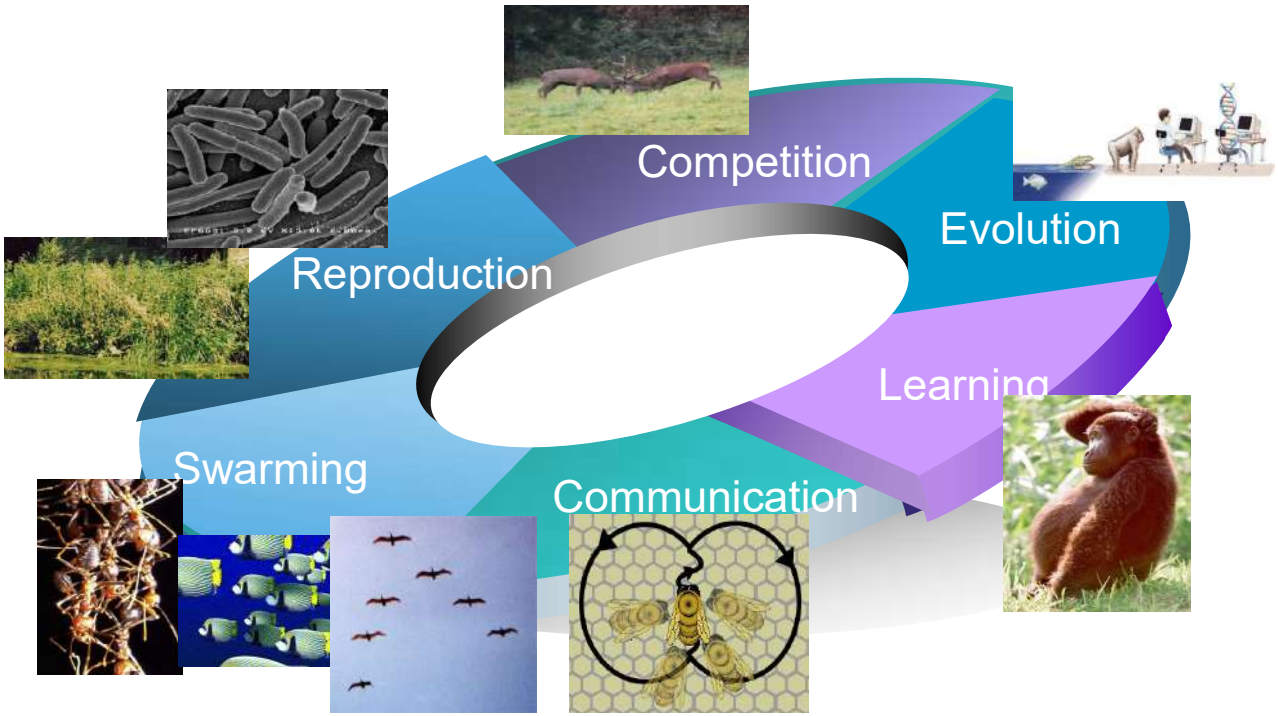
ELTE | FACULTY OF INFORMATICS

# Thank you for your attention!

# Evolutionary algorithms

# Introduction, motivations

# Elements of intelligence in biological systems



# Learning, evolution, adaptation

- Learning:
  - Changes will be made after the evaluations
  - Learning is the purpose of change
- Evolution:
  - Evaluations happen after the changes
  - Evolution is the result of change
- Adaptation:
  - Changes occur after evaluations or as a result of adaptation to environmental conditions

# General optimization algorithms

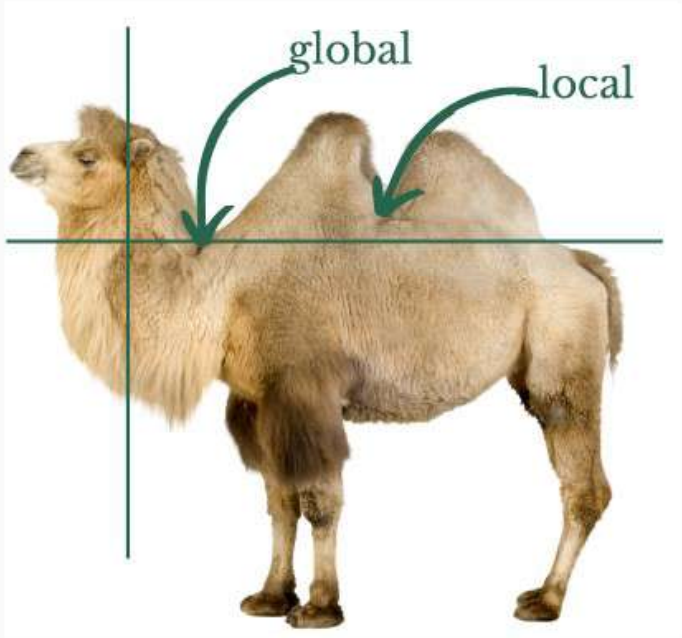
- Deterministic
  - Calculus based
  - Hill climbing
  - …
- Stochastic
  - Random search
  - Simulated annealing
  - …
- Evolutionary algorithms: Stochastic search methods, computationally simulate the natural evolutionary process using the idea of survival of the fittest

# Continuous optimization problem

**General Continuous Optimization Problem**

$$\min_{x \in \mathbb{R}^n} f(x)$$

- **Local Minimum**: $x^*$ in $D$ where $f(x^*) \leq f(x)$ in a small neighborhood.
- **Global Minimum**: $x^*$ in $D$ where $f(x^*) \leq f(x)$ for all $x$ in the domain $D$.



# Evolutionary algorithms

- Their basic principle is the search on the population of solutions guided by laws known from biology
- The individuals in the population are the solutions of the given problem
- The population is evolving, we obtain better and better individuals

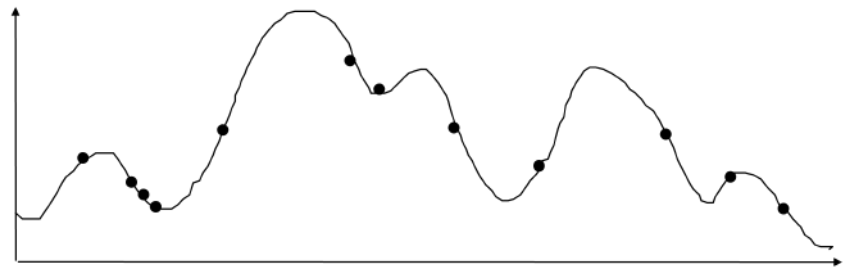
# Evolutionary computation - history

- The idea of using simulated evolution to solve engineering and design problems have been around since the 1950's
- However, it wasn't until the early 1960's that we began to see three influential forms of EA emerge:
  - Evolutionary programming (Lawrence Fogel, 1962)
  - Genetic algorithms (Holland, 1975)
  - Evolution strategies (Rechenberg, 1965 & Schwefel, 1968)
- The designers of each of the EA techniques saw that their particular problems could be solved via simulated evolution
  - Fogel was concerned with solving prediction problems
  - Rechenberg & Schwefel were concerned with solving parameter optimization problems
  - Holland was concerned with developing robust adaptive systems

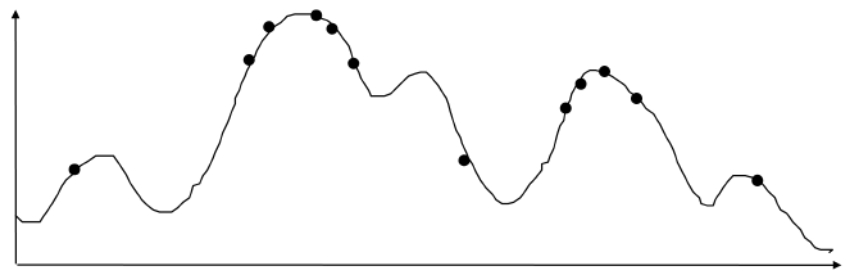
# Terminology

- Gene: functional entity that encodes a specific feature of the individual (e.g. hair color)
- Allele: value of gene (e.g. blonde)
- Genotype: the specific combination of alleles carried by an individual
- Phenotype: the physical makeup of an organism
- Locus: position of the gene within the chromosome
- Individual: chromosome, represents a candidate solution for the problem
- Population: collection of individuals currently alive

# Evolution of the population



Distribution of Individuals in Generation 0

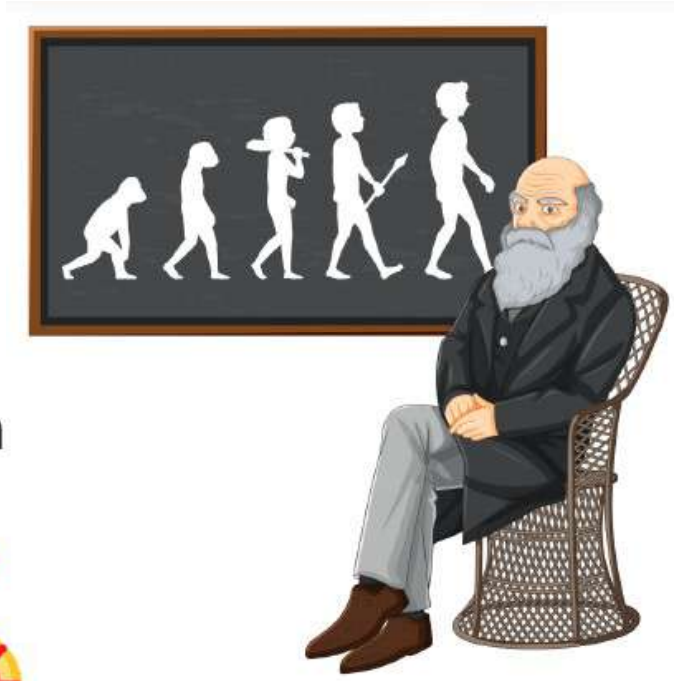


Distribution of Individuals in Generation N

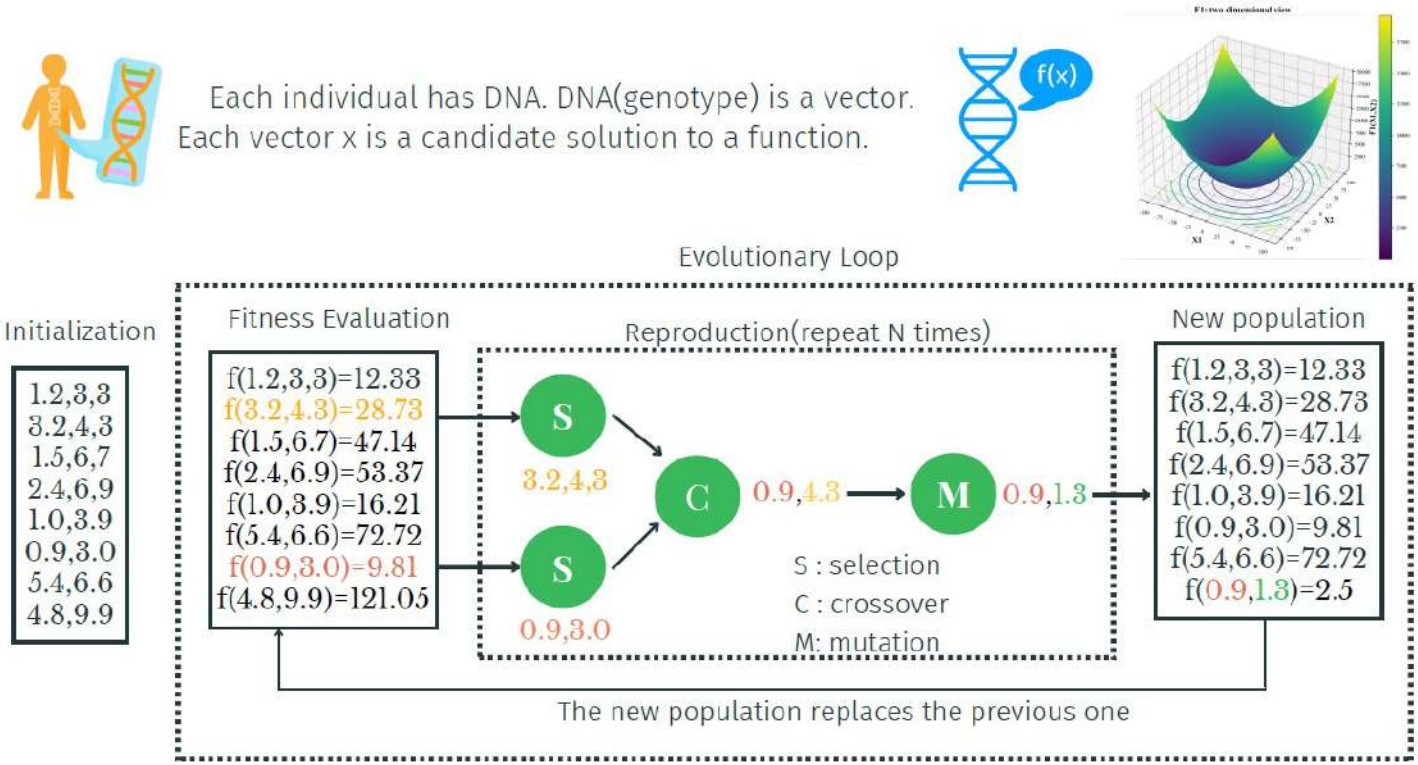
# Genetic algorithms

# Genetic algorithm

- Population-based optimization method
- Inspired by Darwinian theory of evolution
- Stochastic in nature
- Utilizes 3 bio-inspired operators



# Genetic algorithms

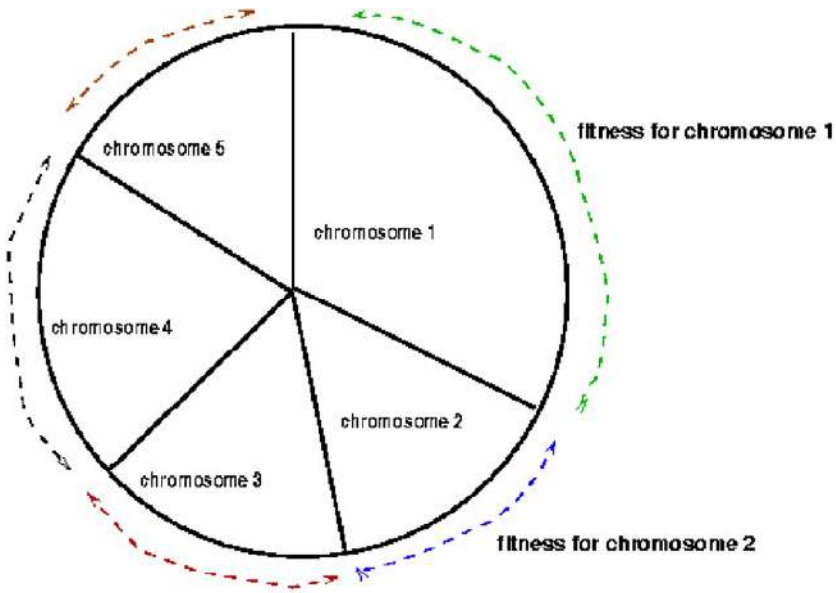


# The individual

- The individual is a solution candidate to the problem
- A possible solution to the problem is encoded into the individual in some form
  - e.g. binary, or real
- Fitness value: the individuals are evaluated according to some criterion how good solution they can provide to the given problem
- Better individuals have a higher fitness value, thus they have a higher chance to survive

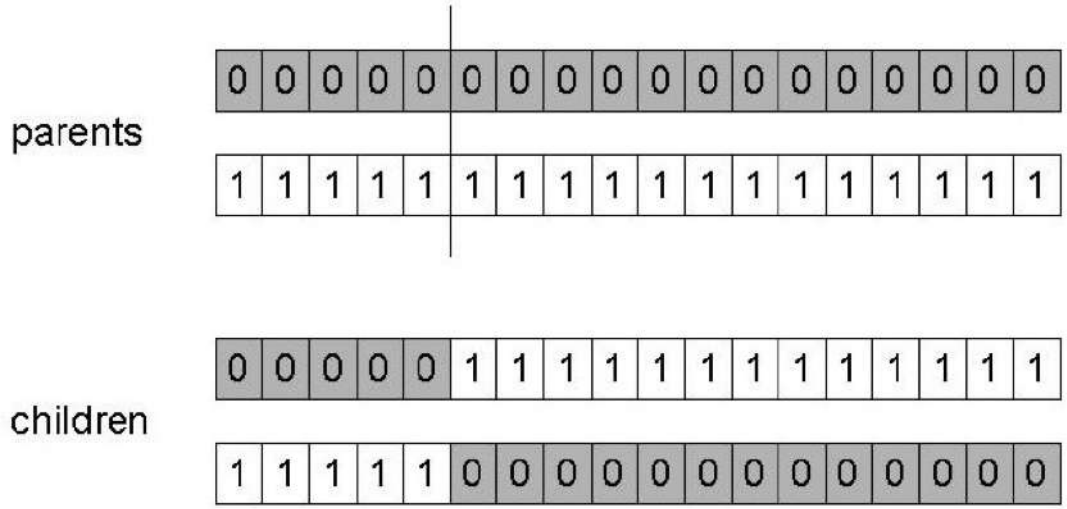
# Selection methods

- There are many ways to select chromosomes to survive to the next generation
- *Roulette wheel selection*: the better the chromosome, the more chance for selection it possesses; imagine a roulette wheel where every chromosome is represented in proportion to its fitness function
- Then a ‘roulette’ ball is thrown and selects chromosomes – chromosomes with larger fitness will be selected with a higher probability



# Crossover

- Choose a random point on the two parents
- Split parents at this crossover point
- Create children by exchanging tails



# Mutation

- Alter each gene independently with a probability $p_m$ (mutation rate)

| | | | | | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| parent | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| child | 0 | 1 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 0 | 1 | 1 | 0 | 0 | 1 |

# An example (Goldberg)

- Simple problem: max $x^2$ over {0,1,…,31}
- Genetic algorithm approach:
  - Representation: binary code, e.g. 01101 $\longleftrightarrow$ 13
  - Population size: 4
  - Crossover, mutation
  - Roulette wheel selection
  - Random initialization

# An example: selection

| String no. | Initial population | $x$ Value | Fitness $f(x) = x^2$ | $Prob_i$ | Expected count | Actual count |
|---|---|---|---|---|---|---|
| 1 | 0 1 1 0 1 | 13 | 169 | 0.14 | 0.58 | 1 |
| 2 | 1 1 0 0 0 | 24 | 576 | 0.49 | 1.97 | 2 |
| 3 | 0 1 0 0 0 | 8 | 64 | 0.06 | 0.22 | 0 |
| 4 | 1 0 0 1 1 | 19 | 361 | 0.31 | 1.23 | 1 |
| Sum | | | 1170 | 1.00 | 4.00 | 4 |
| Average | | | 293 | 0.25 | 1.00 | 1 |
| Max | | | 576 | 0.49 | 1.97 | 2 |

# An example: crossover

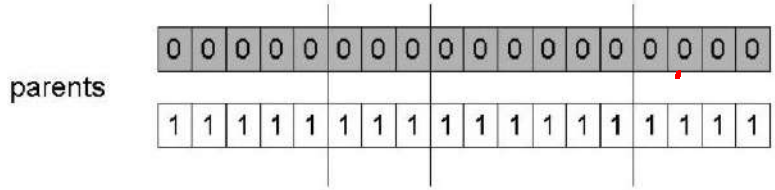
| String no. | Mating pool | Crossover point | Offspring after xover | $x$ Value | Fitness $f(x) = x^2$ |
|---|---|---|---|---|---|
| 1 | 0 1 1 0 \| 1 | 4 | 0 1 1 0 0 | 12 | 144 |
| 2 | 1 1 0 0 \| 0 | 4 | 1 1 0 0 1 | 25 | 625 |
| 2 | 1 1 \| 0 0 0 | 2 | 1 1 0 1 1 | 27 | 729 |
| 4 | 1 0 \| 0 1 1 | 2 | 1 0 0 0 0 | 16 | 256 |
| Sum | | | | | 1754 |
| Average | | | | | 439 |
| Max | | | | | 729 |

# An example: mutation

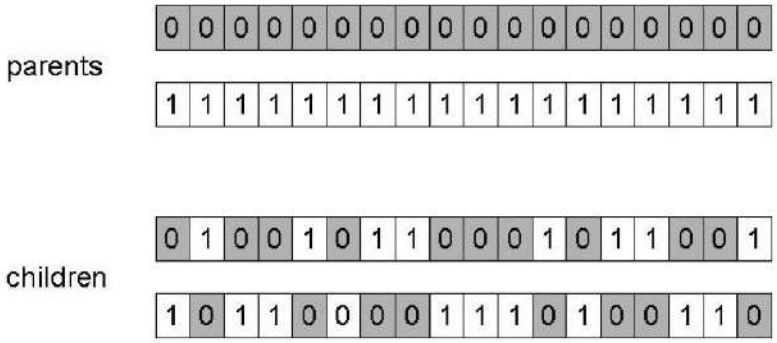
| String no. | Offspring after xover | Offspring after mutation | $x$ Value | Fitness $f(x) = x^2$ |
|---|---|---|---|---|
| 1 | 0 1 1 0 0 | 1 1 1 0 0 | 28 | 784 |
| 2 | 1 1 0 0 1 | 1 1 0 0 1 | 25 | 625 |
| 2 | 1 1 0 1 1 | 1 1 0 1 1 | 27 | 729 |
| 4 | 1 0 0 0 0 | 1 0 1 0 0 | 20 | 400 |
| Sum | | | | 2538 |
| Average | | | | 634.5 |
| Max | | | | 784 |

# Alternative crossover operators

n-point:

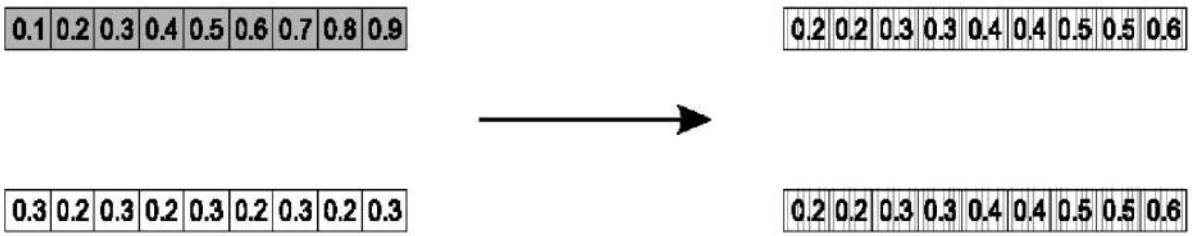



uniform:



# Real-coded GA

- Crossover for example:
  - parents: $\langle x_1,\ldots,x_n\rangle$ and $\langle y_1,\ldots,y_n\rangle$
  - offspring$_1$: $\alpha\underline{x}+(1-\alpha)\underline{y}$
  - swapped for the other offspring
  - e.g.: ($\alpha$ = 0.5)

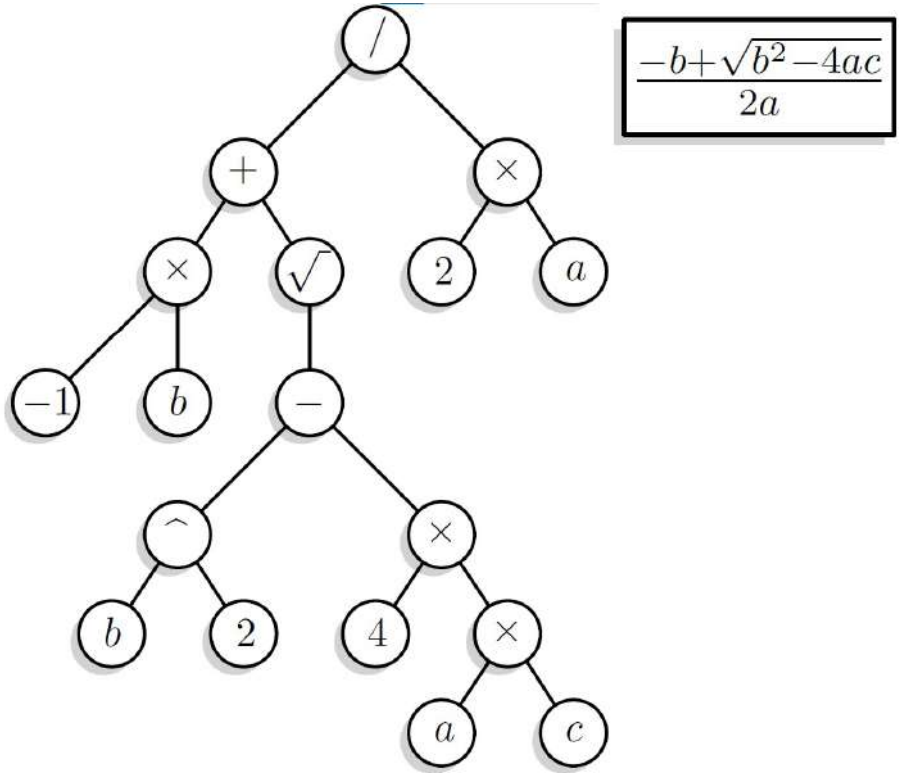



- Mutation: $\underline{x}=\langle x_1,\ldots,x_n\rangle$ $\underline{x}'=\langle x'_1,\ldots,x'_n\rangle$ $x_i,x'_i \in[LB_i,UB_i]$

# Genetic programming

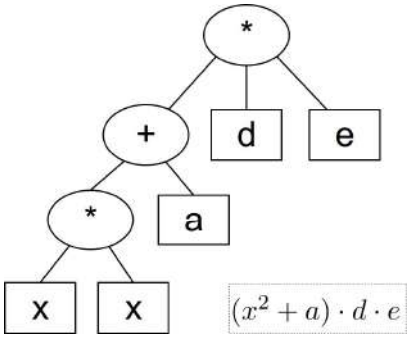
# Genetic programming

- John Koza (~1990)
- Genetic programming applies the approach of genetic algorithm to the space of possible computer programs
- A wide variety of seemingly different problems from many different fields can be reformulated as a search for a computer program to solve the problem
- Individuals are described by an expression tree

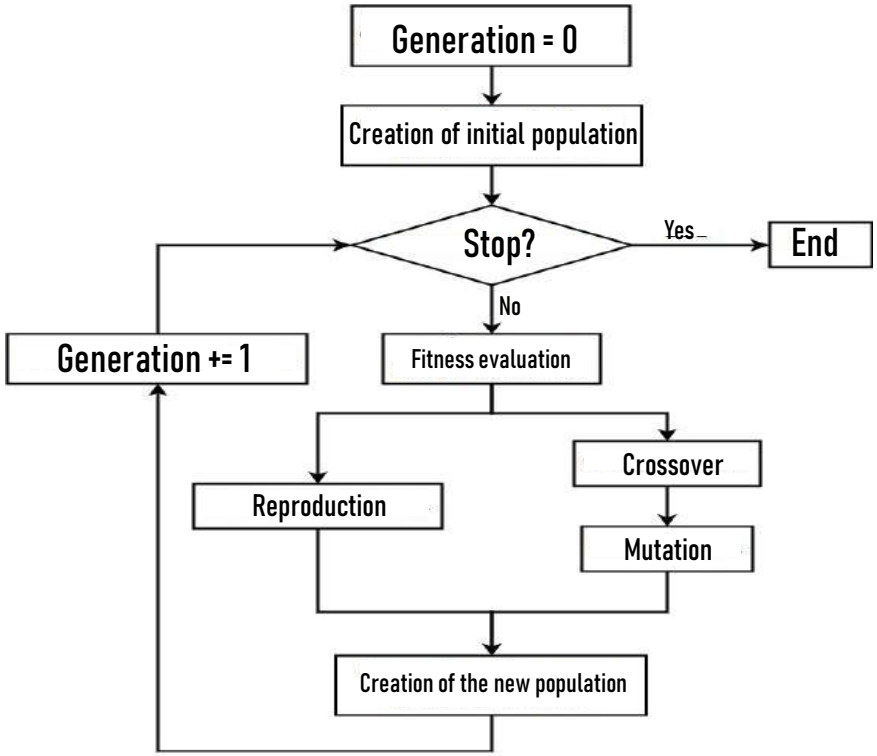


# Genetic programming

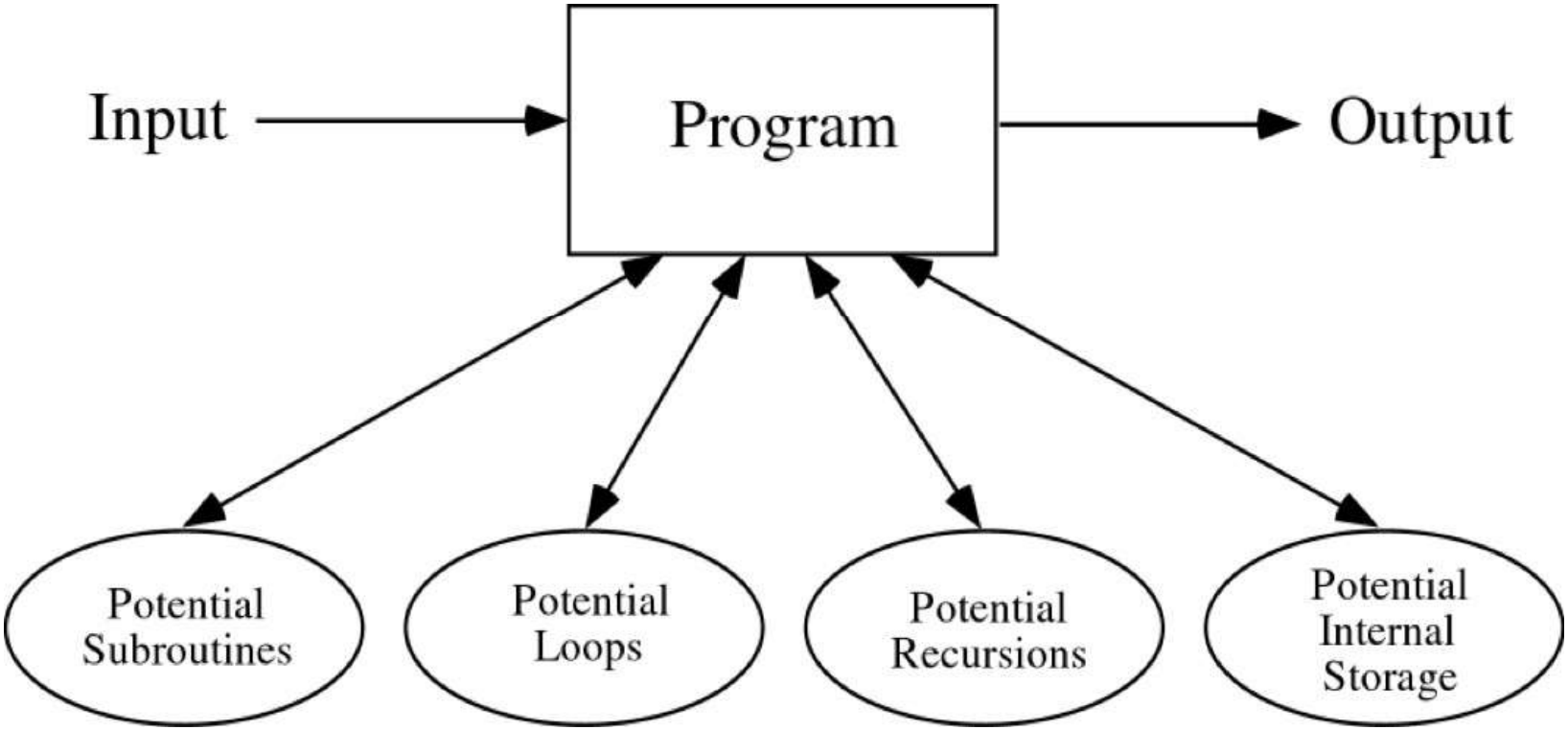
- The individual is a tree-based structure




- Evolutionary operators
  - Reproduction
  - Crossover
  - Mutation

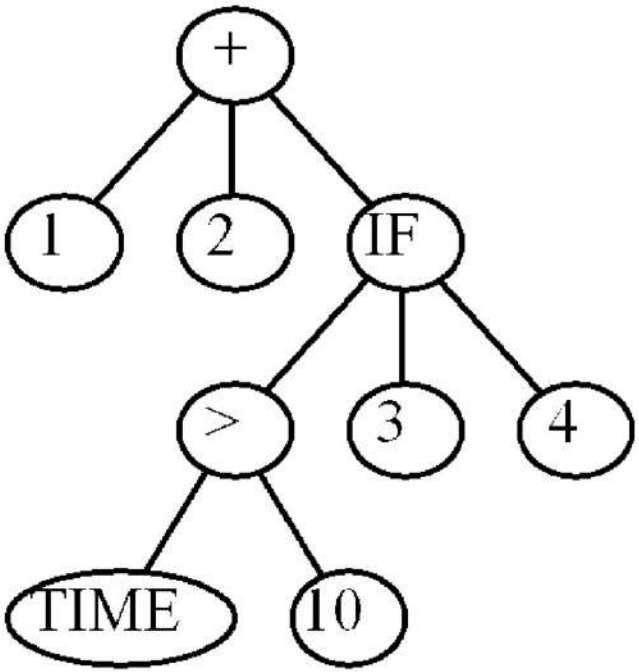


# A computer program



# A computer program in C and its tree representation

```
int foo (int time)
{
    int temp1, temp2;
    if (time > 10)
        temp1 = 3;
    else
        temp1 = 4;
    temp2 = temp1 + 1 + 2;
    return (temp2);
}
```




(+ 1 2 (IF (> TIME 10) 3 4))

# Creating random programs

- Available functions:
  - e.g. F = {+, -, *, %, IF}
- Available terminals:
  - e.g. T = {X, Y, constants}
- The random programs are:
  - syntactically valid
  - executable
- The trees may have different sizes and shapes

# Genetic operators in GP

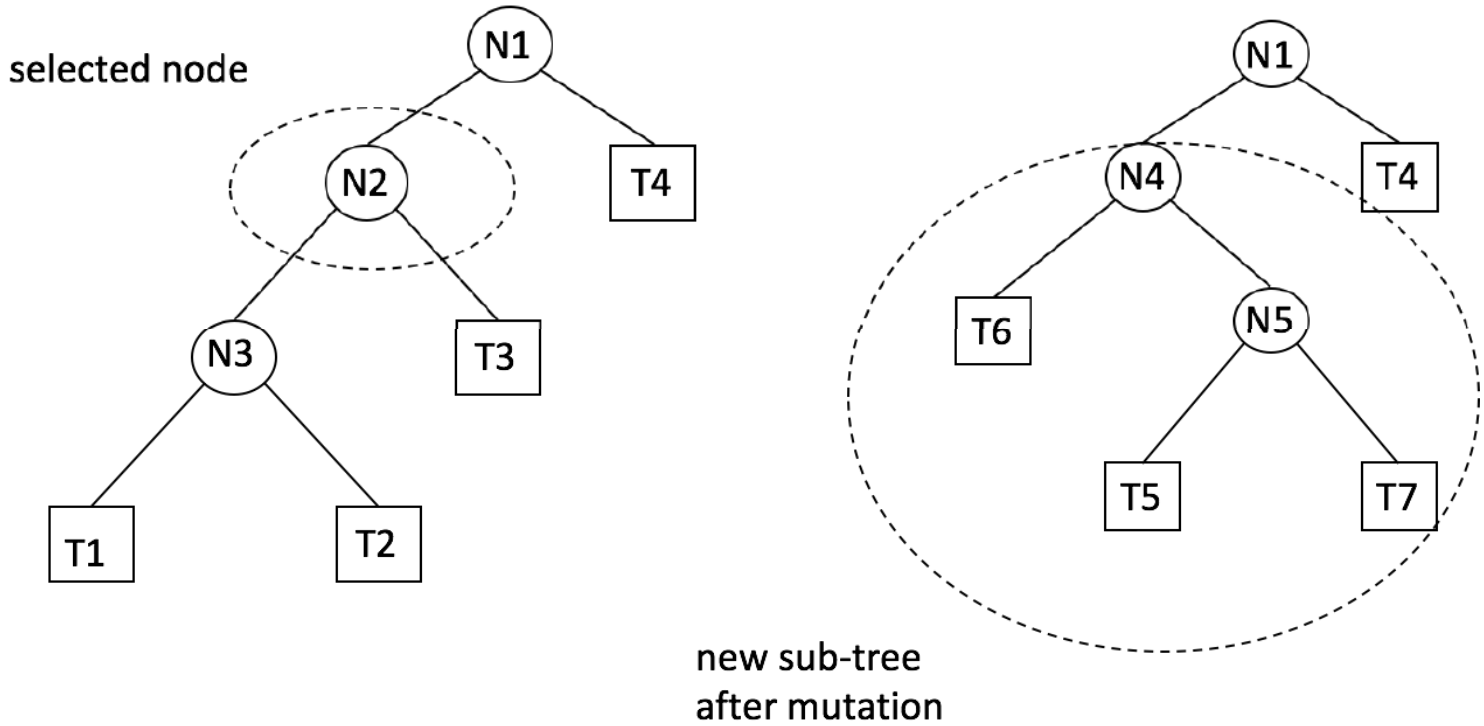
- Reproduction
- Mutation
- Crossover

- Reproduction:
  - select parent based on fitness
  - copy it (unchanged) into the next generation of the population

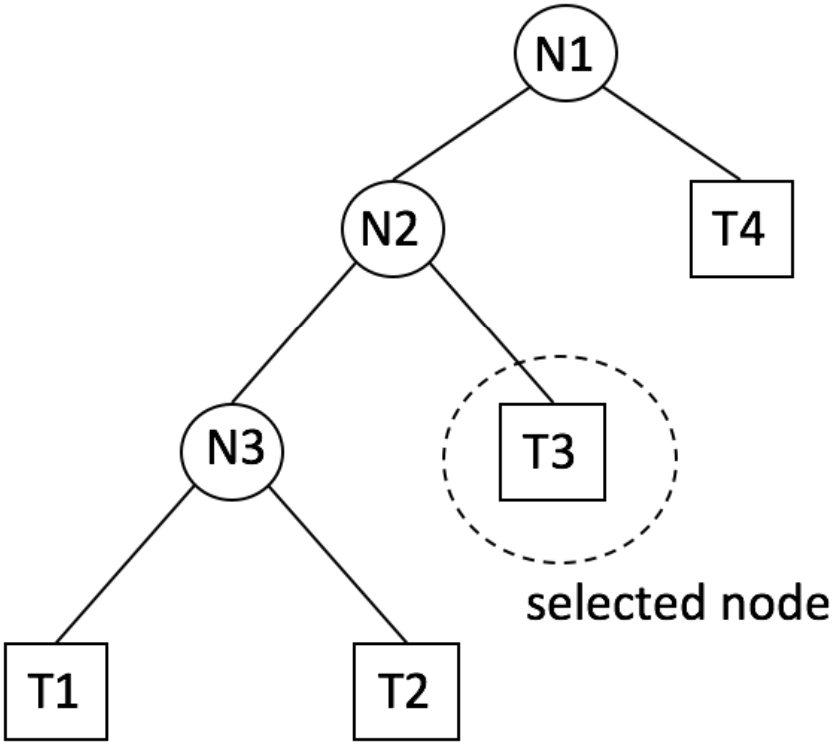
# Mutation

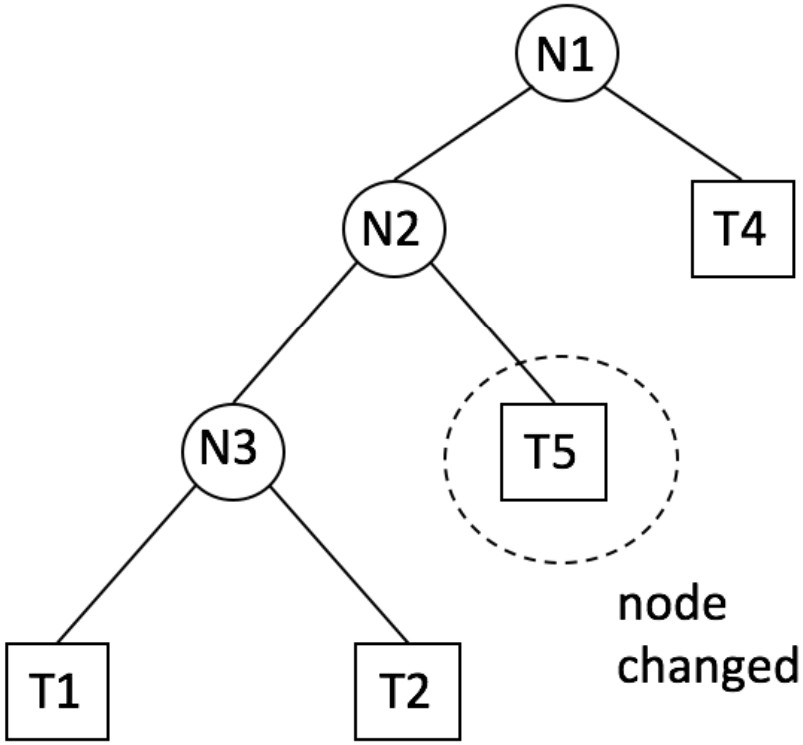
- Select 1 parent (based on fitness)
- Pick a point in the tree
- Delete subtree at the picked point
- Grow new subtree at the mutation point in same way as generated trees for initial random population
- The result is a syntactically valid executable program
- Put the offspring into the next generation of the population

# Subtree mutation



# Node mutation

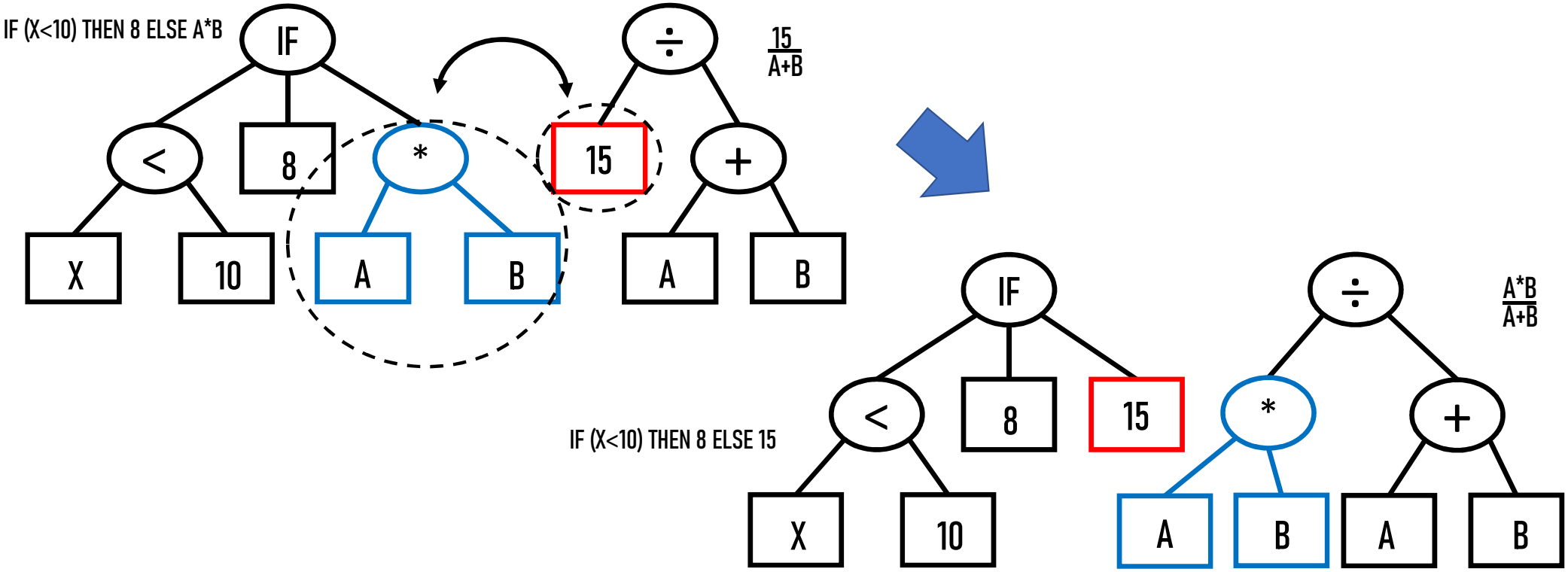


# Crossover

- Select 2 parents (based on fitness)
- Randomly pick a node in the tree for first parent
- Independently randomly pick a node for second parent
- Exchange the subtrees at the two picked points
- The result is a syntactically valid executable program
- Put the offspring into the next generation of the population

# Crossover


IF (X<10) THEN 8 ELSE A*B
IF
<
8
*
X
10
A
B
÷
15
+
A
B
15
A+B
IF (X<10) THEN 8 ELSE 15
IF
<
8
15
X
10
÷
*
+
A
B
A
B
A*B
A+B

# Preparatory steps

- Determining the set of terminals
- Determining the set of functions
- Determining the fitness measure
- Determining the parameters for the run
- Determining the criterion for terminating a run

ELTE | FACULTY OF INFORMATICS

# AI Today – Trends

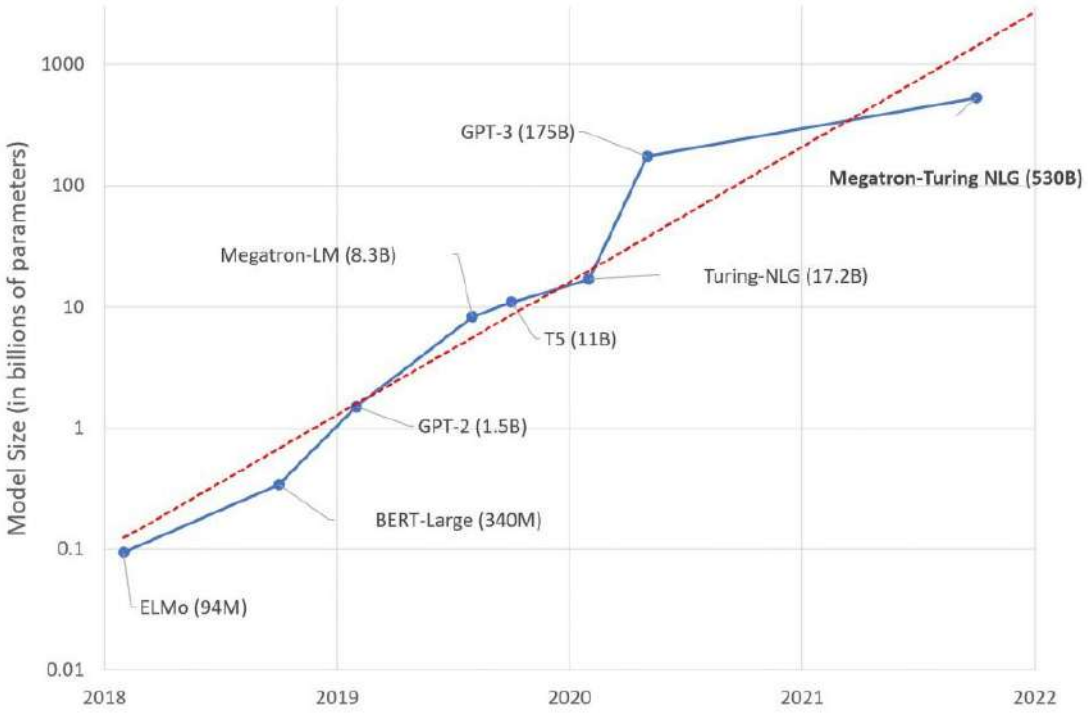
**Introduction to Machine Learning**
Computer Science BSc Course, ELTE Faculty of Informatics

László Gulyás

Associate Professor
Department of Artificial Intelligence
lguiyas@inf.elte.hu

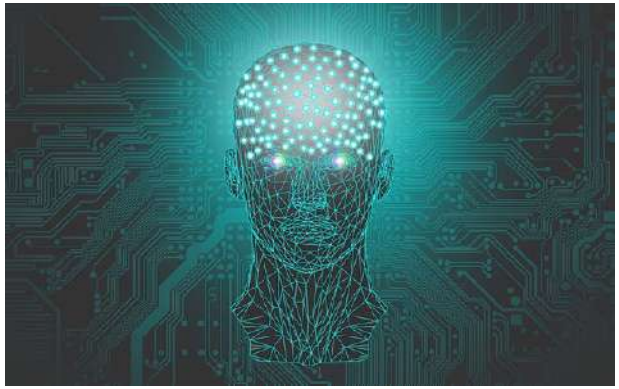
ELTE | IK
DEPARTMENT OF ARTIFICIAL INTELLIGENCE

# Trends – Not Discussed Today



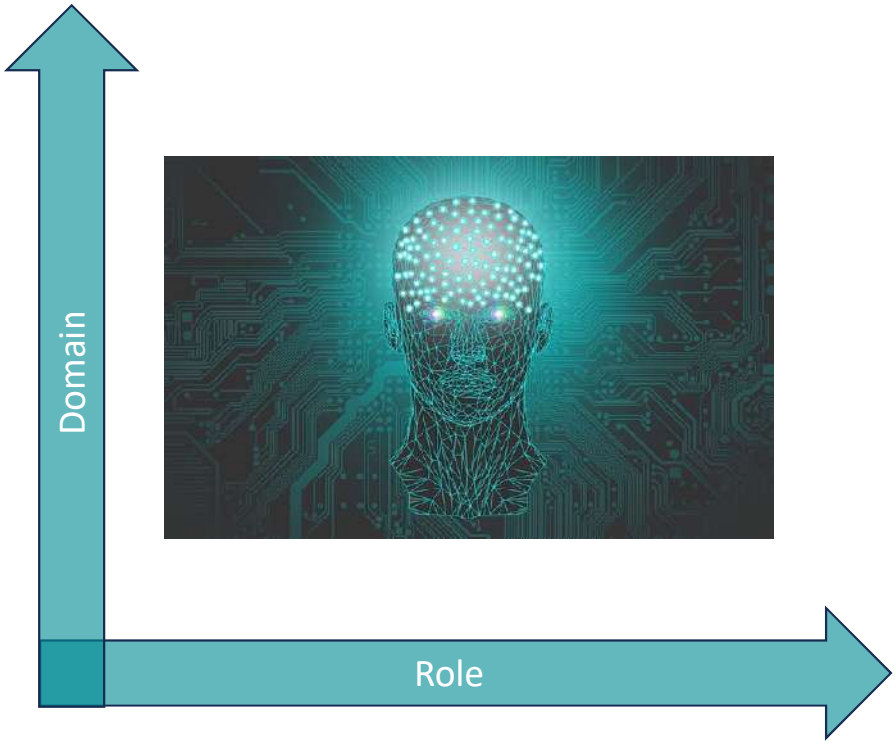

https://www.microsoft.com/en-us/research/blog/using-deepspeed-and-megatron-to-train-megatron-turing-nlg-530b-the-worlds-largest-and-most-powerful-generative-language-model/

# Trends Discussed – Specialisation



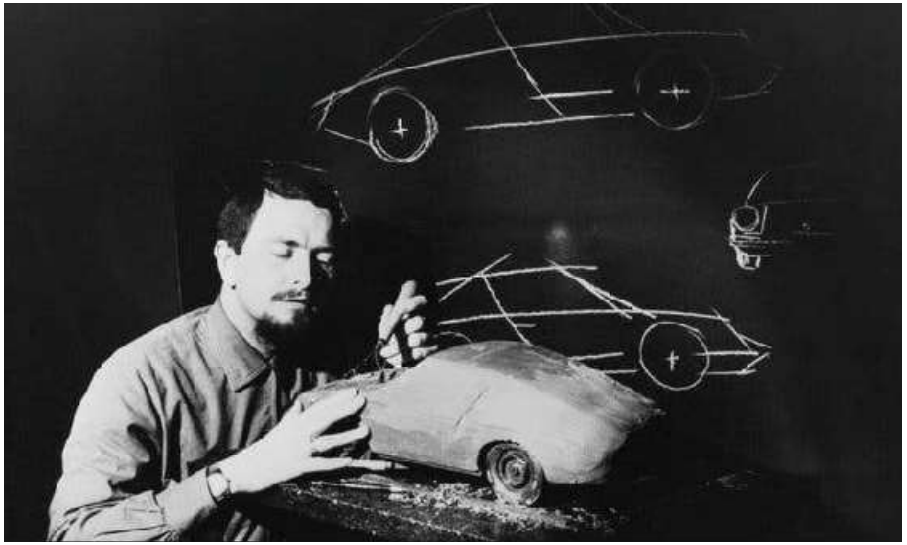
Role

# Trends Discussed – Specialisation



# Analogy

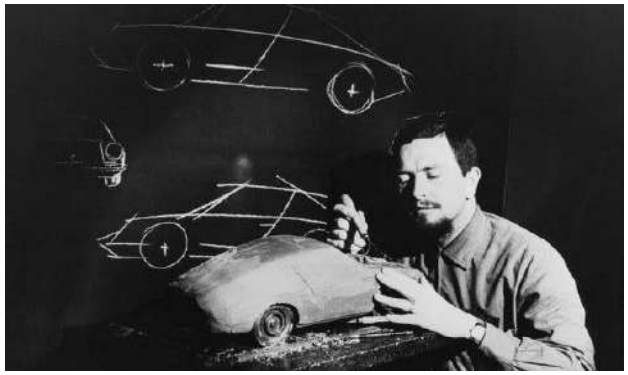
# Analogy

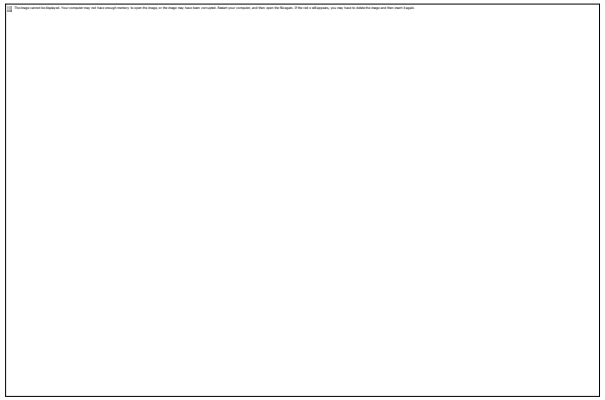


Ferdinand Porsche
Car Designer Genius

Lewis Hamilton
7-times F1 Champion

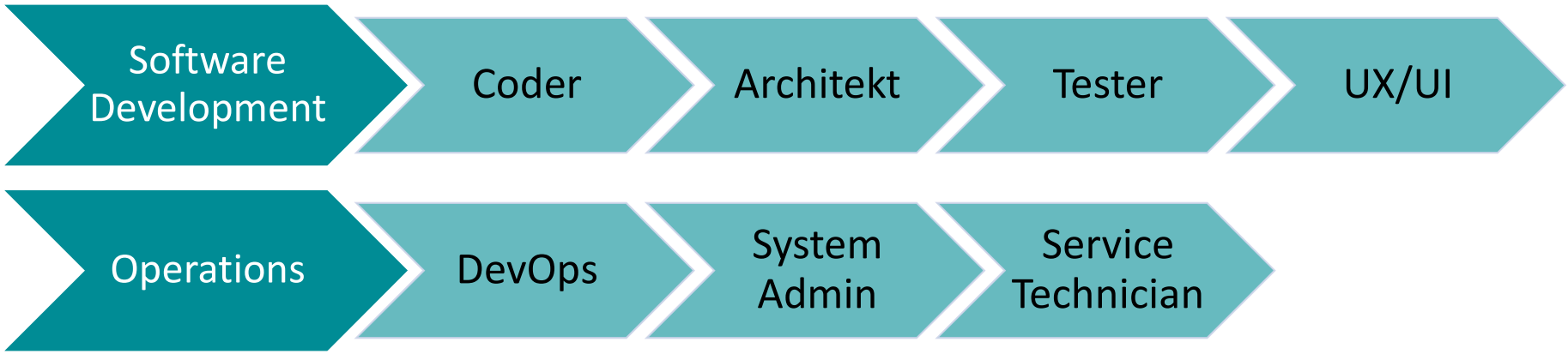
# Role Specialisation



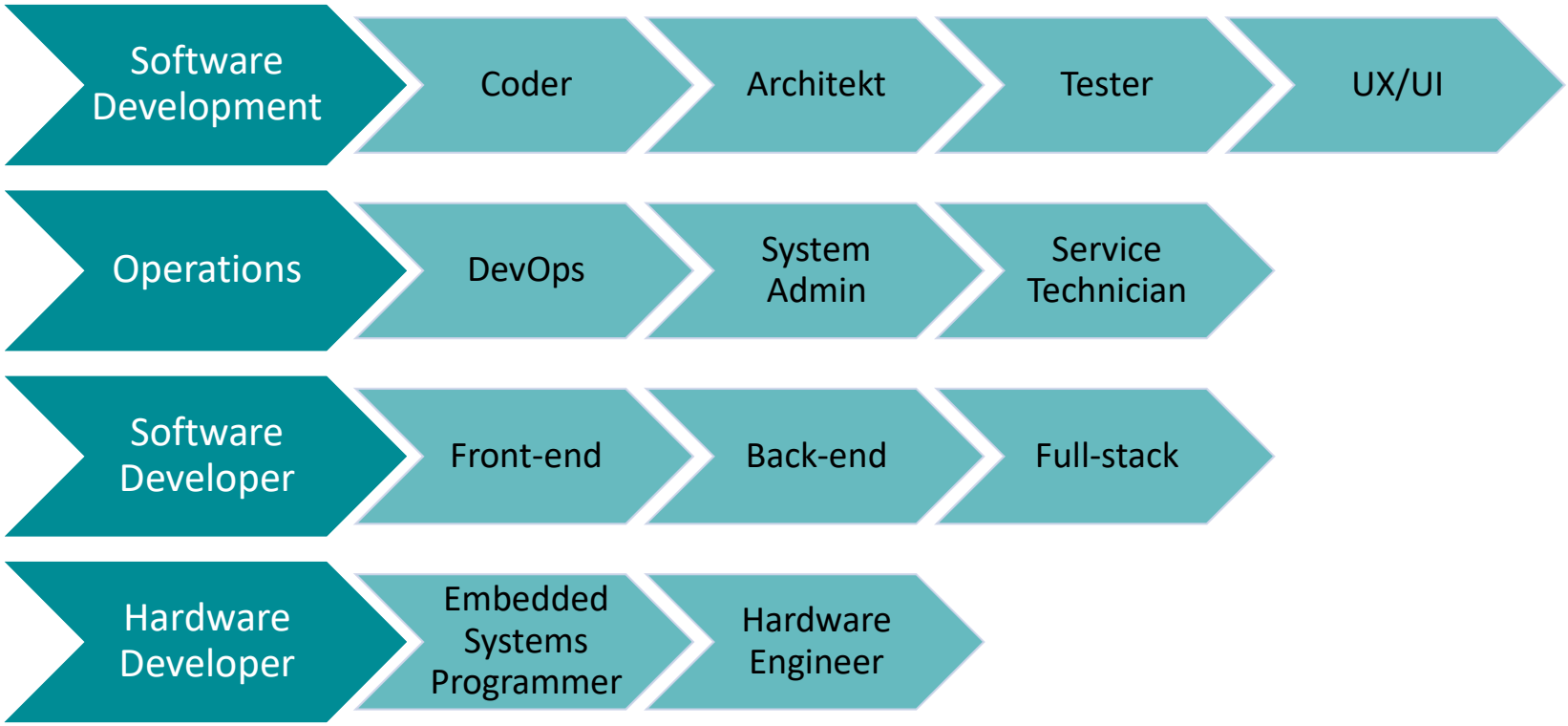


# „Computer Expert”

# „Computer Expert”

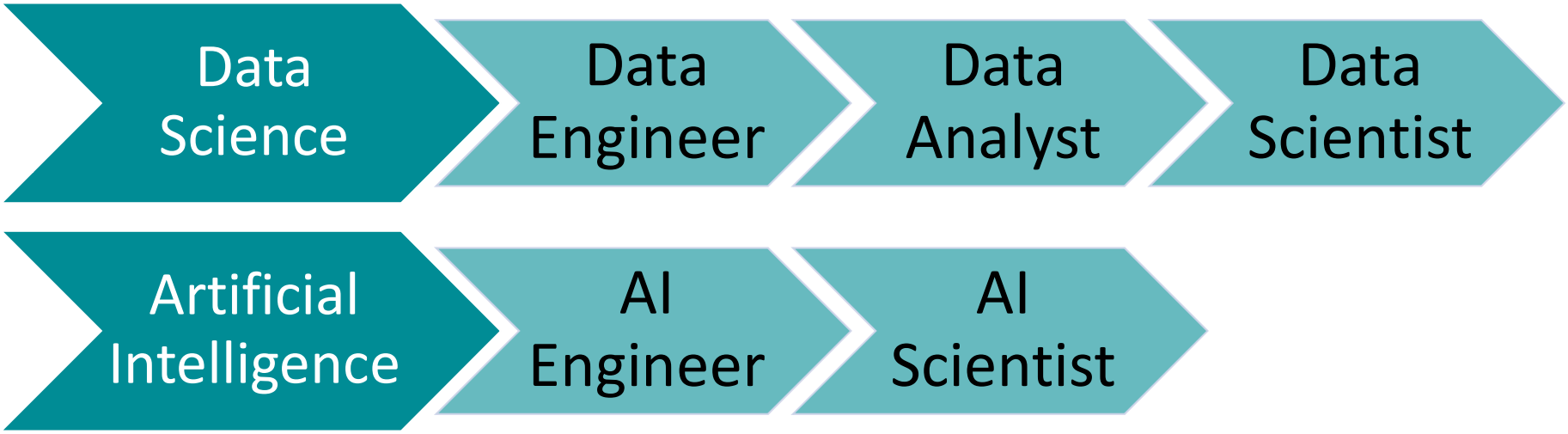


# „Computer Expert”

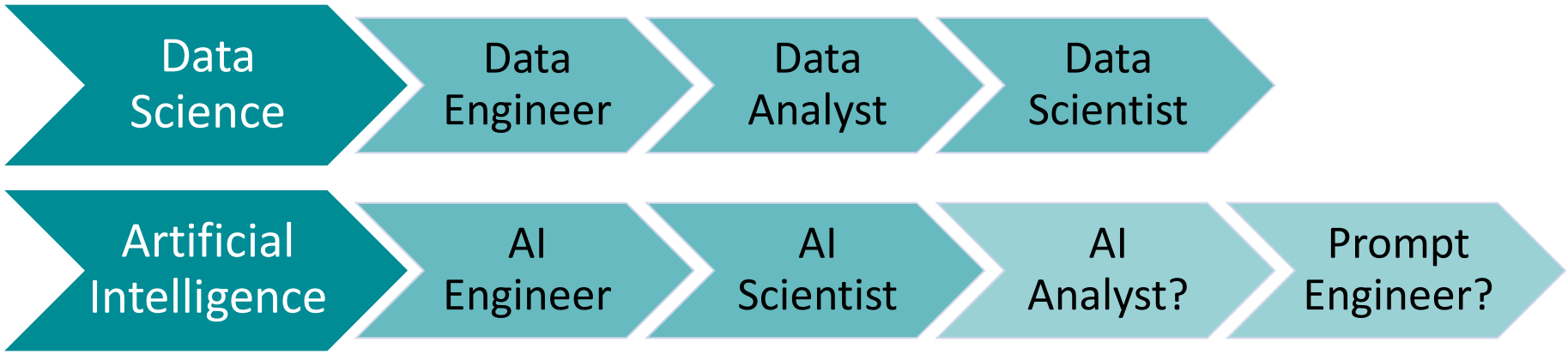


# Role Specialisation

# Role Specialisation



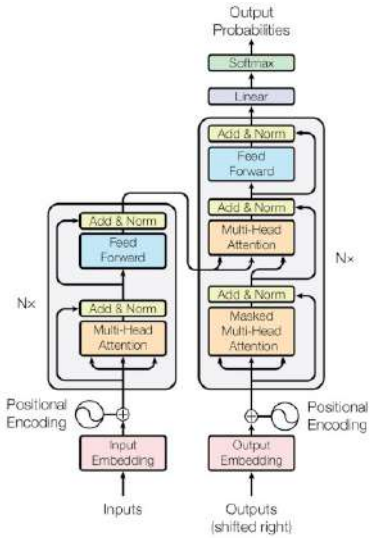
# Role Specialisation

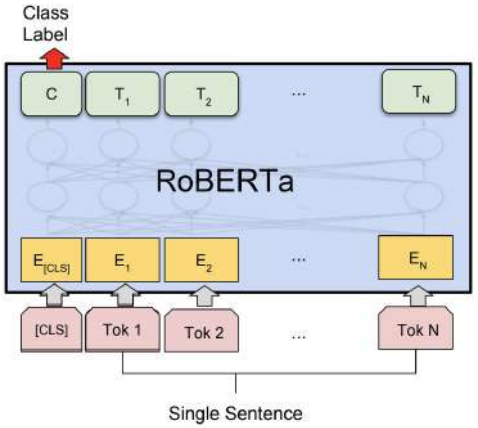


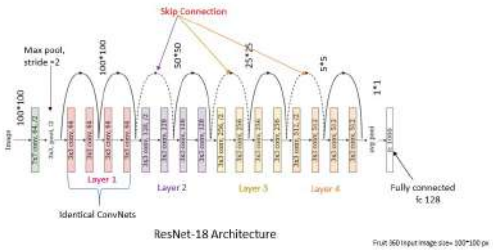
# Background

- **De facto standard model architectures**
  - C.f. Classic Algorithms
  - C.f. Design Patterns
- **Transfer Learning**
  - Downloadable pre-trained models (ResNet-18, YOLOv8, etc.)
- **Starting from scratch is rare**
  - Rapidly growing ‚library'
  - Best practices to be applied

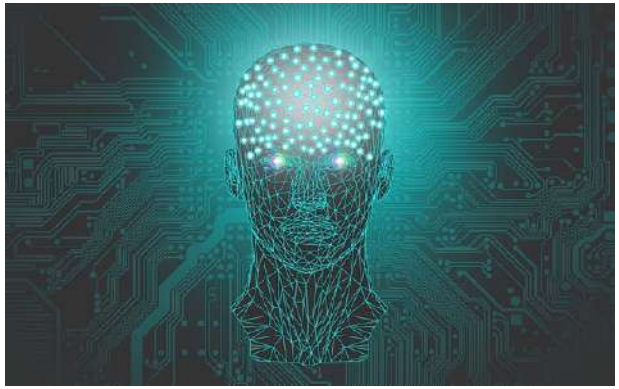
# Background


Output Probabilities
Softmax
Linear
Add & Norm
Feed Forward
Add & Norm
Multi-Head Attention
Add & Norm
Masked Multi-Head Attention
Nx
Add & Norm
Feed Forward
Add & Norm
Multi-Head Attention
Nx
Positional Encoding
Positional Encoding
Input Embedding
Output Embedding
Inputs
Outputs (shifted right)


Class Label
C
T1
T2
...
TN
RoBERTa
E[CLS]
E1
E2
...
EN
[CLS]
Tok 1
Tok 2
...
Tok N
Single Sentence

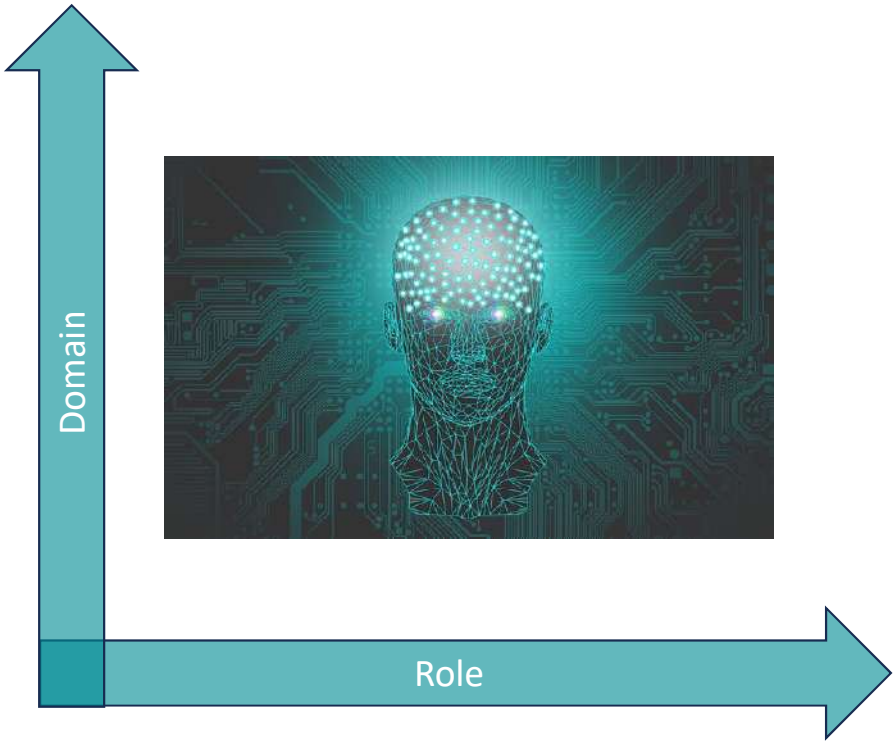

Skip Connection
Max pool, stride = 2
100*100
50*50
25*25
5*5
1*1
Layer 1
Layer 2
Layer 3
Layer 4
Fully connected fc 128
Identical ConvNets
ResNet-18 Architecture

# Specialisation

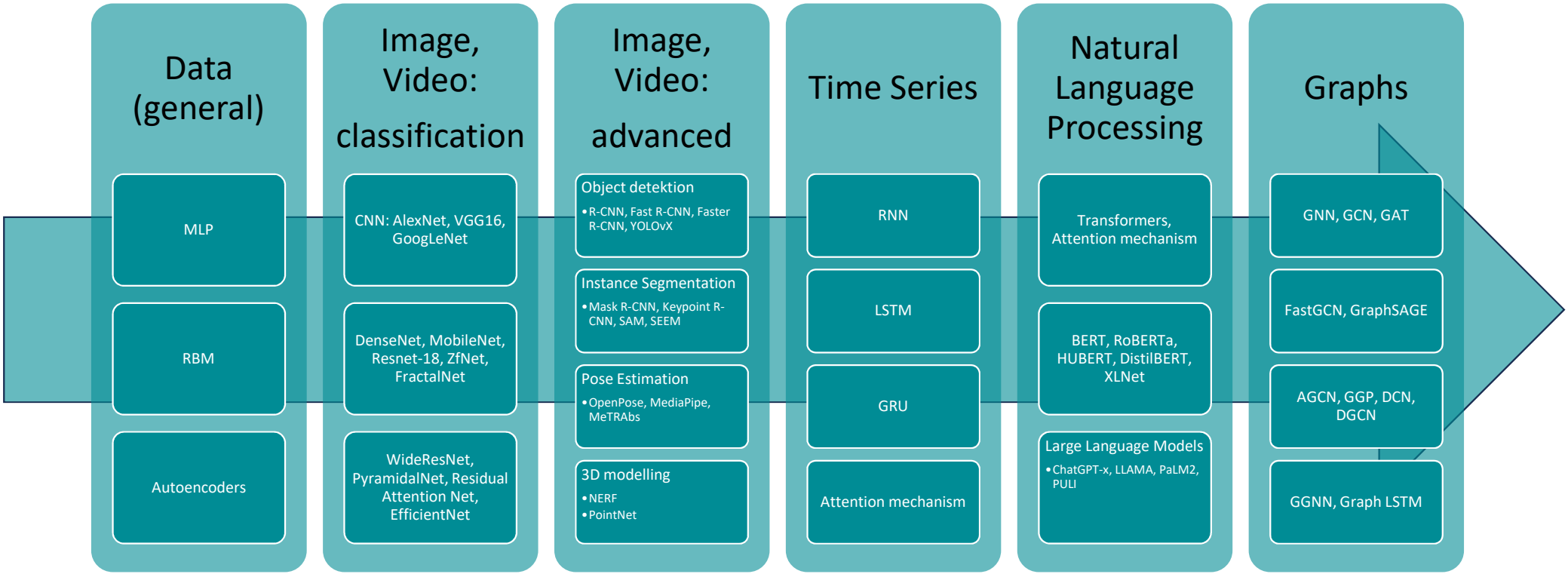


Role

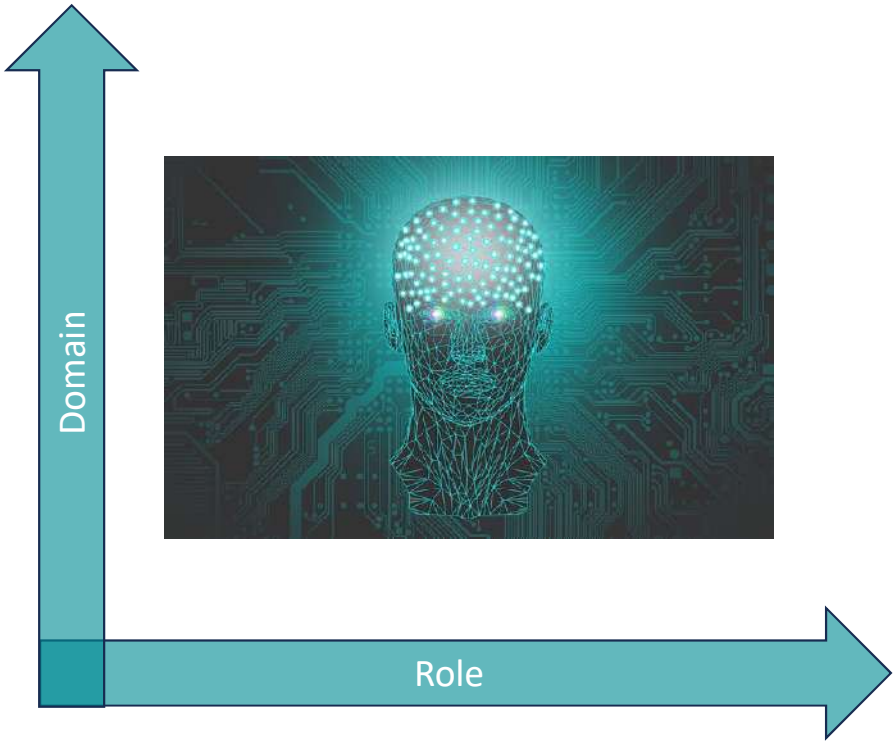
# Specialisation



# Domain


Data (general)
MLP
RBM
Autoencoders
Image, Video: classification
CNN: AlexNet, VGG16, GoogLeNet
DenseNet, MobileNet, Resnet-18, ZfNet, FractalNet
WideResNet, PyramidalNet, Residual Attention Net, EfficientNet
Image, Video: advanced
Object detektion
• R-CNN, Fast R-CNN, Faster R-CNN, YOLOvX
Instance Segmentation
• Mask R-CNN, Keypoint R-CNN, SAM, SEEM
Pose Estimation
• OpenPose, MediaPipe, MeTRAbs
3D modelling
• NERF
• PointNet
Time Series
RNN
LSTM
GRU
Attention mechanism
Natural Language Processing
Transformers, Attention mechanism
BERT, RoBERTa, HUBERT, DistilBERT, XLNet
Large Language Models
• ChatGPT-x, LLAMA, PaLM2, PULI
Graphs
GNN, GCN, GAT
FastGCN, GraphSAGE
AGCN, GGP, DCN, DGCN
GGNN, Graph LSTM

# Specialisation



# Implications

## Specialisation

Diversified expectations at workplace

Specialised education

- AI Engineer
- AI Designer

## New Requirements

Testing

Proof of Correctness

Explainable Models

ELTE | FACULTY OF INFORMATICS

# Thank you!


## László Gulyás

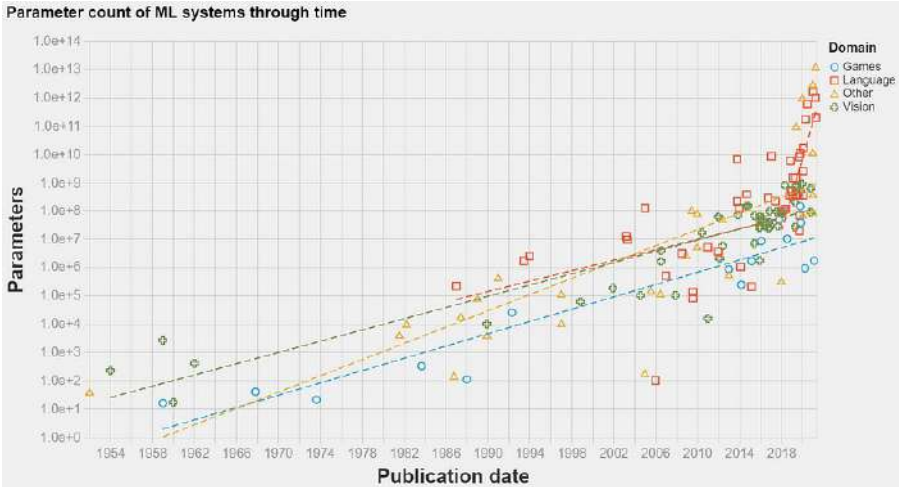
Associate Professor
Department of Artificial Intelligence
lgulyas@inf.elte.hu

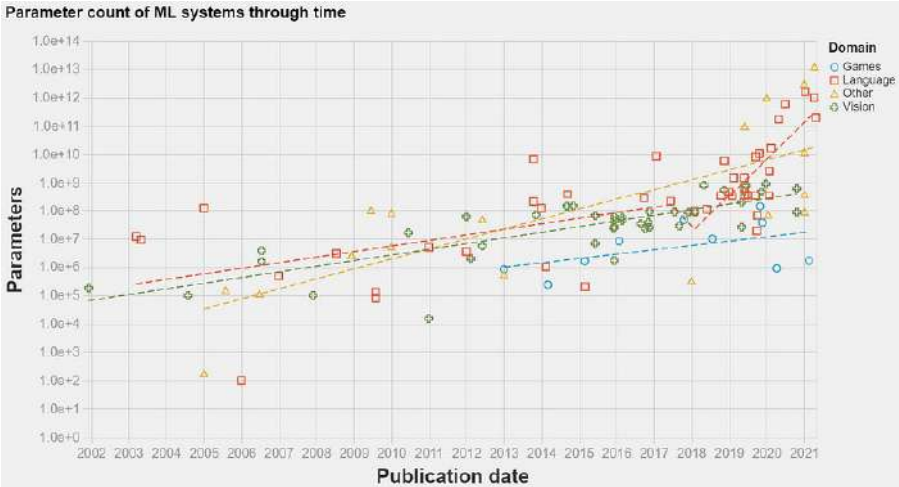

ELTE | IK
DEPARTMENT OF ARTIFICIAL INTELLIGENCE

ELTE | FACULTY OF INFORMATICS

# BACKUP

ELTE | IK
DEPARTMENT OF ARTIFICIAL INTELLIGENCE


Parameter count of ML systems through time
Parameters
Publication date
Domain
Games
Language
Other
Vision


Parameter count of ML systems through time
Parameters
Publication date
Domain
Games
Language
Other
Vision

ELTE | IK
INFORMATIKAI KAR

# AI Today – Optimisation

**Introduction to Machine Learning**
Computer Science BSc Course, ELTE Faculty of Informatics

László Gulyás

Associate Professor
Department of Artificial Intelligence
lguiyas@inf.elte.hu

ELTE | IK
DEPARTMENT OF ARTIFICIAL INTELLIGENCE

# What is AI?

AI = Optimisation

# What is AI?

$$f^* = max_{x \in X} f(x)$$

# What is AI?

$$f^* = argmax_{x \in X} f(x)$$

# What is AI?

$$f^* = argmin_{x \in X} f(x)$$

# What is AI?

$$f^* = argmin_{w \in W} f(w, x)$$

w – Learned parameters
x -- Input instances of the task (images to be recognized, etc.)

# Intelligence = Optimisation??

# Intelligent Behavior: Problem Solving

- Shortest Path from A to B
  - *Fewest* steps, *shortest* path, *shortest* time…
  - C.f., navigation

- Solve the 15 stone puzzle
  - *Fewest* number of moves

- Generally, find the moves that yields the *highest* score

# Intelligent Behavior: Planning

- In what order shall moves be made?
  - To achieve the *most*...
  - To gain the *highest* combined benefit…

- Example: scheduling factory production
  - Incoming series of orders (various products, batch sizes)
  - Different tooling requirements
  - Varying unit production time

  →
  - *Minimize* time lost on switching between products

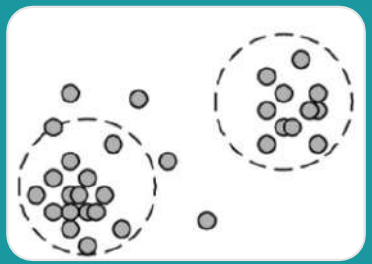
# Intelligent Behavior: Recognising Images, Voices, Text

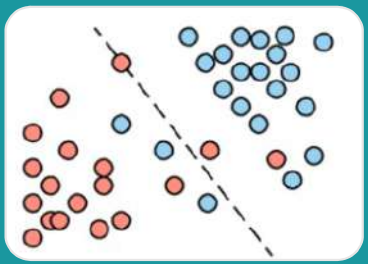
- Error (*loss*):
  - E.g., in what percent of the cases did it err?
- *Goal to **minimize***
  - *I.e., increase accuracy*

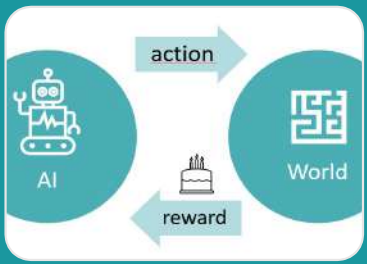
# Intelligent Behavior: *Learning*

- The Key
  - *Improvement*
- Example:
  - Aim at target
  - Assess Results
  - Change → Get better
- Again
  - Minimize error
  - Maximize accuracy

# Machine Learning
## See upcoming lectures








**Unsupervised learning**

- Raw data

**Supervised learning**

- Training data (with labels)

**Reinforcement learning**

- No data, but observable outcomes

# Remark

- Learning is usually a slow process
  - Large amounts of data is required
  - Many training examples need to be seen
  - Many games should be played (cf. Practice, Sports training)

- Humans also learn for *decades*
  - Basic skills – first few years
  - Elementary knowledge – about 12 years
  - Professional knowledge – 3+2 years (BSc + MSc)

  →To get a *junior* position (read: unexperienced)

  - It never really stops

# Intelligent Behavior: Generating Something New

© Night Café Studio

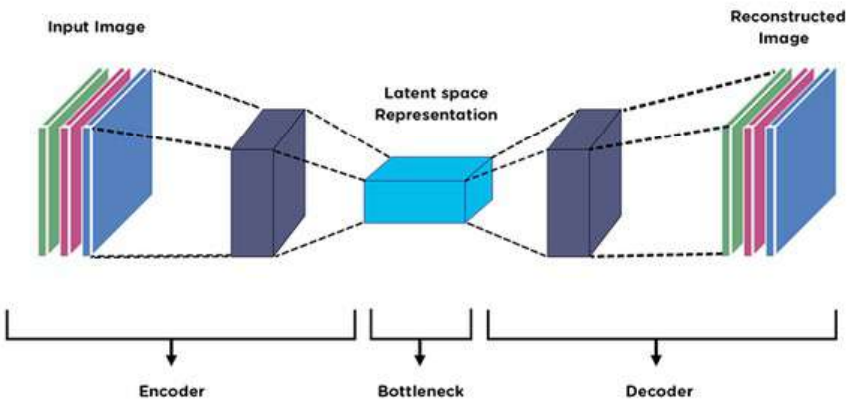
Synthetic images produced by StyleGAN, a GAN created by Nvidia researchers.

# Autoencoders
## An abstract approach

- Input: $\mathbb{R}^D$
- Output: $\mathbb{R}^D$

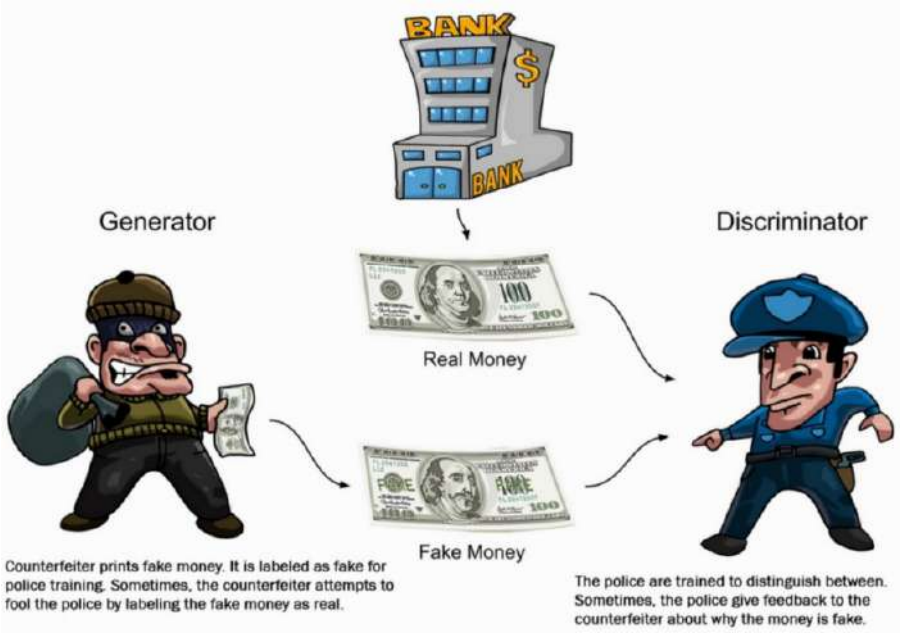
- Embedding: $\boldsymbol{f}: \mathbb{R}^{\boldsymbol{D}} \to \mathbb{R}^{\boldsymbol{d}}$
  - $d \ll D$
- Decoding: $\boldsymbol{g}: \mathbb{R}^{\boldsymbol{d}} \to \mathbb{R}^{\boldsymbol{D}}$
  - $\forall x \in \mathbb{R}^D: g(f(x)) \approx x$

- Generative output: $\boldsymbol{g}(\boldsymbol{y})$
  - $y \in \mathbb{R}^d$, random



# Intelligent Behavior: Generating Something New

- **Generative Adversarial Network (GAN)**
  - First widely successful generative solution
- **Bandit vs Cops game**
  - ‚Generative duel'
  - They both learn (optimise)
  - After a while, the output of the bandit is **new creation**




©Prajwol Lamichhane

# What is hard in optimisation?

OK, AI is optimisation, but…

# AI = Optimisation
# What is hard in this?

$$f^* = argmin_{w \in W} f(w, x)$$

# AI = Optimisation
# What is hard in this? Dimensionality

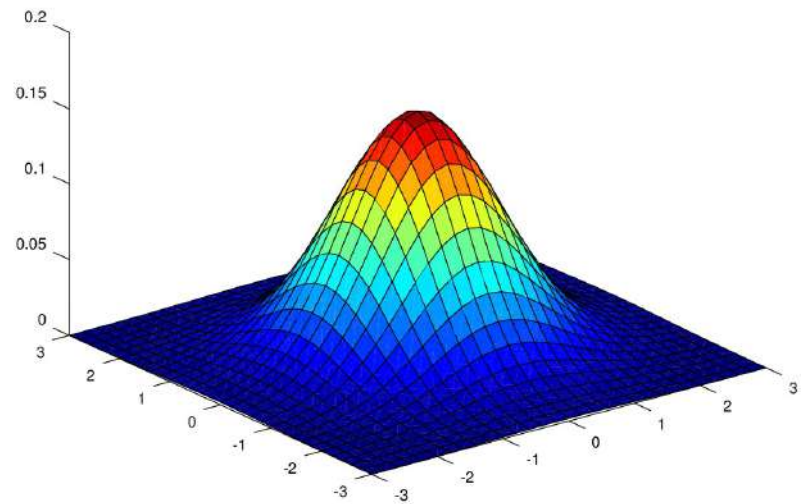
$$f^* = argmin_{w \in W} f(w, x)$$

$$W \subseteq \mathbb{R}^N, N > 10^6$$

# AI = Optimisation
# What is hard in this?

$$f^* = argmin_{w \in W} f(w, x)$$

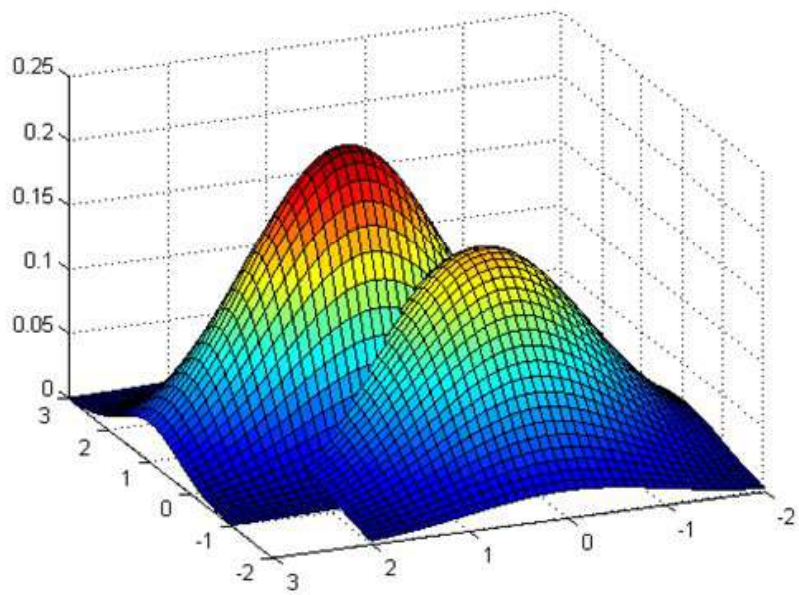
$$\mathrm{W} \subseteq \mathbb{R}^N, \boldsymbol{N} > \mathbf{10^6}$$



# AI = Optimisation
# What is hard in this? Non-monotonicity

$$f^* = argmin_{w \in W} f(w, x)$$

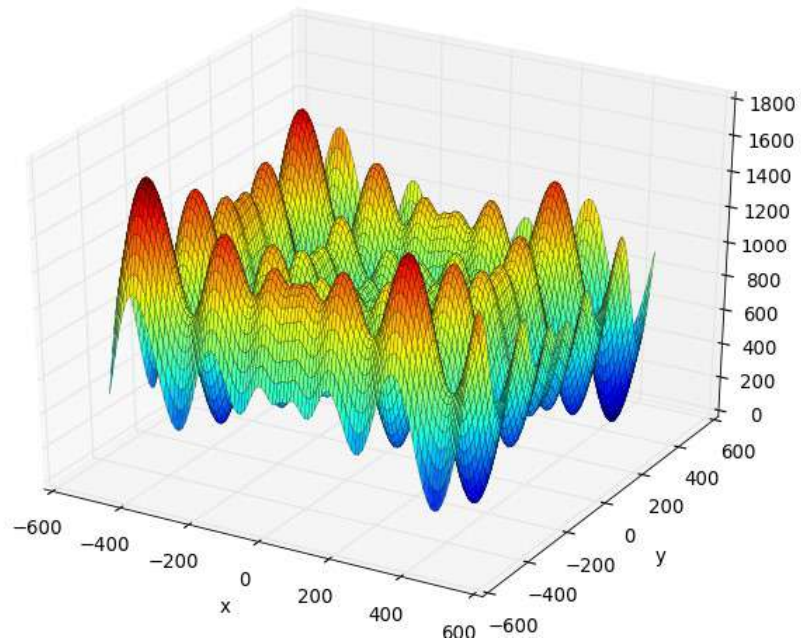
$$\mathrm{W} \subseteq \mathbb{R}^N, \boldsymbol{N} > \mathbf{10^6}$$



# AI = Optimisation
# What is hard in this? Ruggedness

$$f^* = argmin_{w \in W} f(w, x)$$

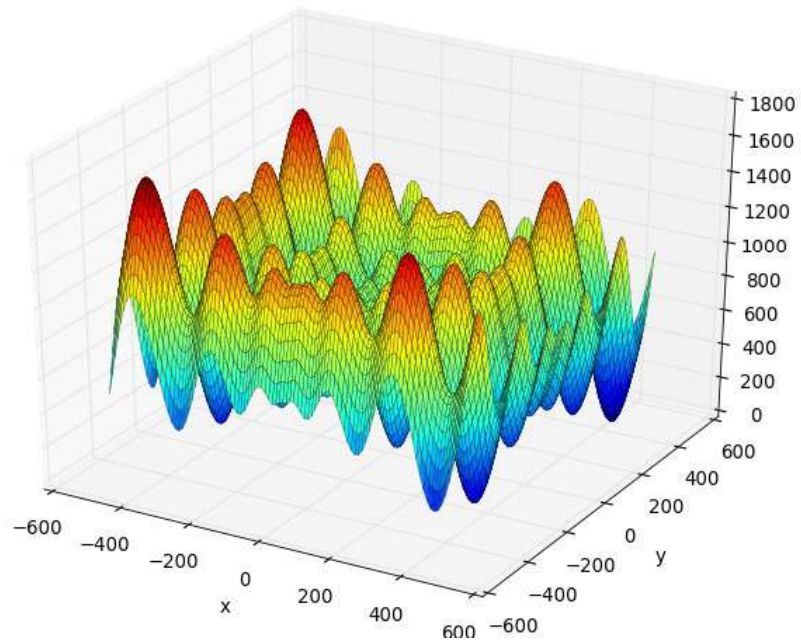
$$W \subseteq \mathbb{R}^N, \boldsymbol{N > 10^6}$$



# This is a hard problem – No known general solution

$$f^* = argmin_{w \in W} f(w, x)$$

$$\mathrm{W} \subseteq \mathbb{R}^N, \boldsymbol{N} > \mathbf{10^6}$$



# Inspirations

Borrowing from where it works...

# Biologically Inspired Methods –

- Brain
- Evolution
- Social Insects

- **Artificial Neural Networks**
- Evolutionary Algorithms
- Swarm Intelligence Methods

ELTE | IK
INFORMATIKAI KAR

# Thank you!


## László Gulyás

Associate Professor
Department of Artificial Intelligence
lgulyas@inf.elte.hu


ELTE | IK
DEPARTMENT OF ARTIFICIAL INTELLIGENCE

ELTE | FACULTY OF INFORMATICS

# ETHICS AND CHAT GPT

Introduction to Machine Learning – Lecture 3.

Balázs Nagy, PhD

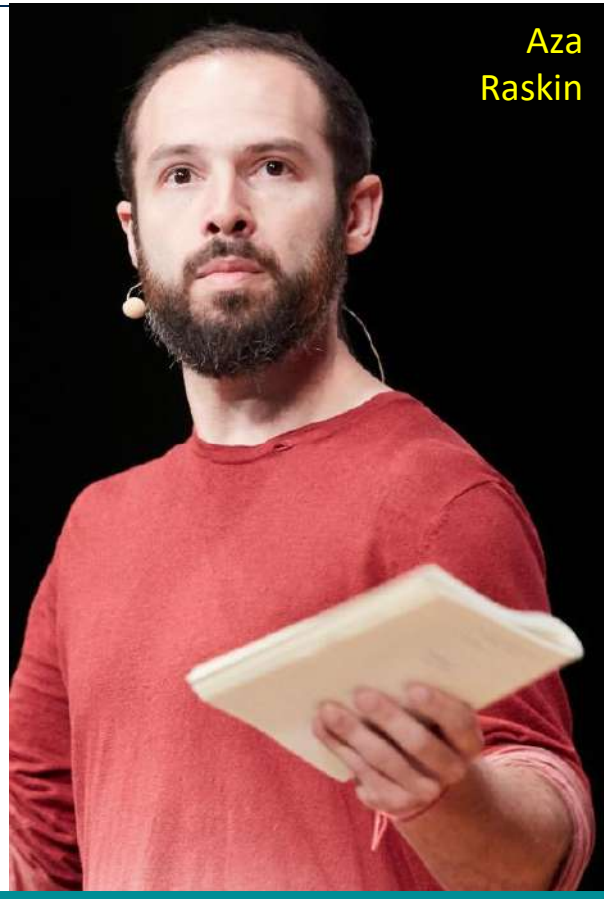
ELTE | IK
DEPARTMENT OF ARTIFICIAL INTELLIGENCE

BOSCH

# Mindset

Center for Humane Technology

Our mission is to shift technology towards a more humane future that supports our well-being, democratic functioning, and shared information environment.



# The AI dilemma

References:

- The AI dilemma speech

https://www.youtube.com/watch?v=cB0_-qKbal4&t=1163s

https://www.youtube.com/watch?v=xoVJKj8lcNQ&t=203s

- Video

https://www.youtube.com/watch?v=uG8vItscFKc

# Théâtre D'opéra Spatial - Allen Jason

# How do we treat AI?

## AI won an art contest, and artists are furious

By Rachel Metz, CNN Business
Published 10:54 AM EDT, Sat September 3, 2022

## An AI-generated artwork's state fair victory fuels arguments over 'what art is'

## He used AI to win a fine-arts competition. Was it cheating?

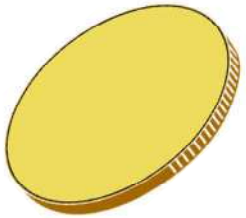
One judge said the striking piece evoked Renaissance art. But some critics compared it to 'entering a marathon and driving a Lamborghini to the finish line.'

By Drew Harwell
September 2, 2022 at 11:08 a.m. EDT

- How do we treat AI?
- AI taking over our jobs?
- Tool or Competition?

# Two sides

Helping Tool



OR

Fierce Rival

*„We can achieve great things with it."*

*„AI take our jobs."*

# Closure?

## An old master? No, it's an image AI just knocked up ... and it can't be copyrighted

**US ruling on works created through artificial intelligence gives boost to creative workers fighting for livelihoods**

The decision, delivered by the US copyright office review board, found that Théâtre d'Opéra Spatial, an AI-generated image that won first place at the 2022 Colorado state fair annual art competition, was not eligible because copyright protection "excludes works produced by non-humans".

Artist Jason Allen claimed his use of the online AI-platform Midjourney allowed him to claim authorship of the image because he "entered a series of prompts, adjusted the scene, selected portions to focus on, and dictated the tone of the image". But the board ruled that "if all of a work's 'traditional elements of authorship' were produced by a machine, the work lacks human authorship, and the Office will not register it".

https://www.theguardian.com/technology/2023/sep/24/an-old-master-no-its-an-image-ai-just-knocked-up-and-it-cant-be-copyrighted?fbclid=IwAR0XaFtPaDvqG-IsjVal_XuxjR5qXiRN_hIHreusy2LTL_kqtizwGlg24-Q

# Attempt #1

Prompt:
A big red fire breathing dragon flying over a ruined castle on a cliff overlooking the sea, with a stormy sky and crashing waves, eerie, moody, atmospheric

Software: SD online
Iteration: 2

# Road to success

# Attempt # many

Prompt:
Acrylic painting of a mountain landscape, with a stormy sky and a cabin nestled in the forest, high contrast, high resolution

Software: SD offline
Iteration: 40
Batch: 4

# Optional Homework #1 (10 points)

- Generative Art Competition
- Generate a picture with a generative AI
- Topic:

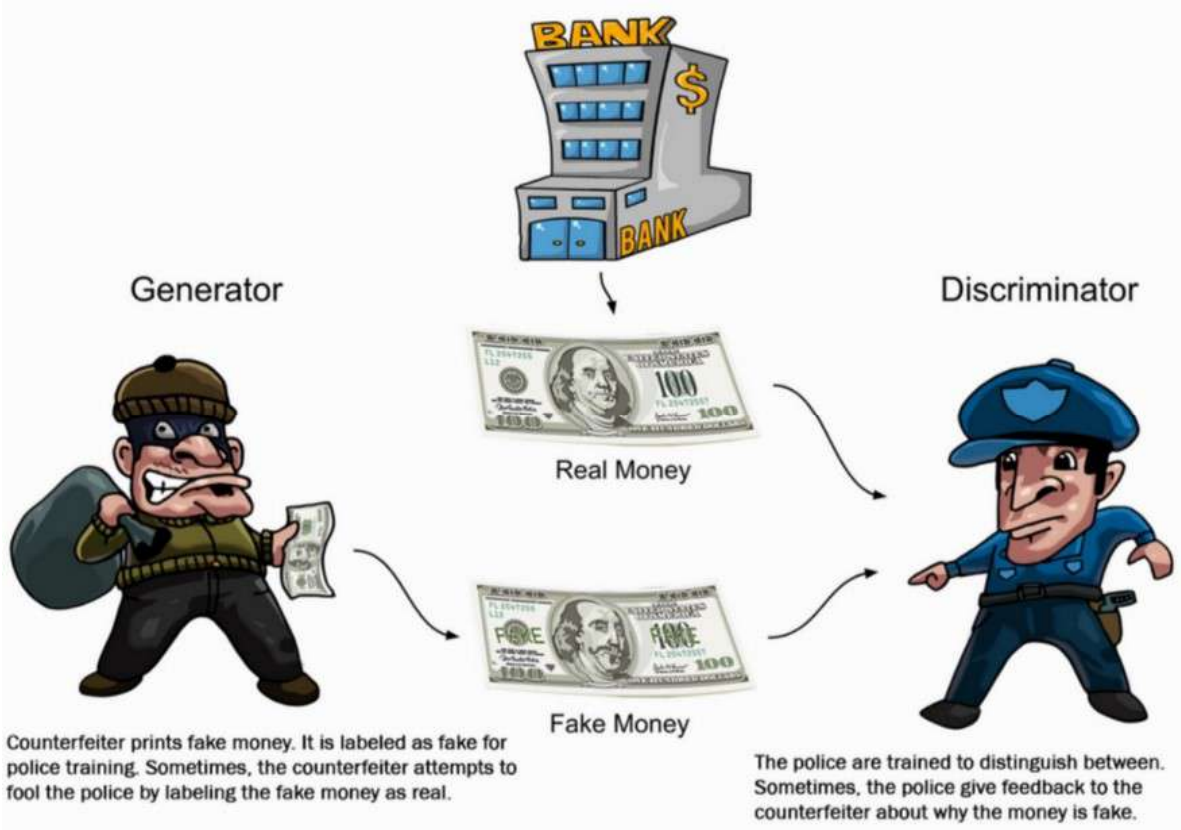
  „Majestic creature”
- Document the used software or application and the steps that lead to the final image
- Include the used prompt

- Option 1:
  Dall-E
  https://openai.com/dall-e-2
- Option 2:
  Stable Diffusion XL online
  https://clipdrop.co/stable-diffusion
- Option 3:
  Stable Diffusion XL offline
  https://github.com/AbdBarho/stable-diffusion-webui-docker
- Option 4:
  Midjourney
- Option 5:
  Any other Generative model

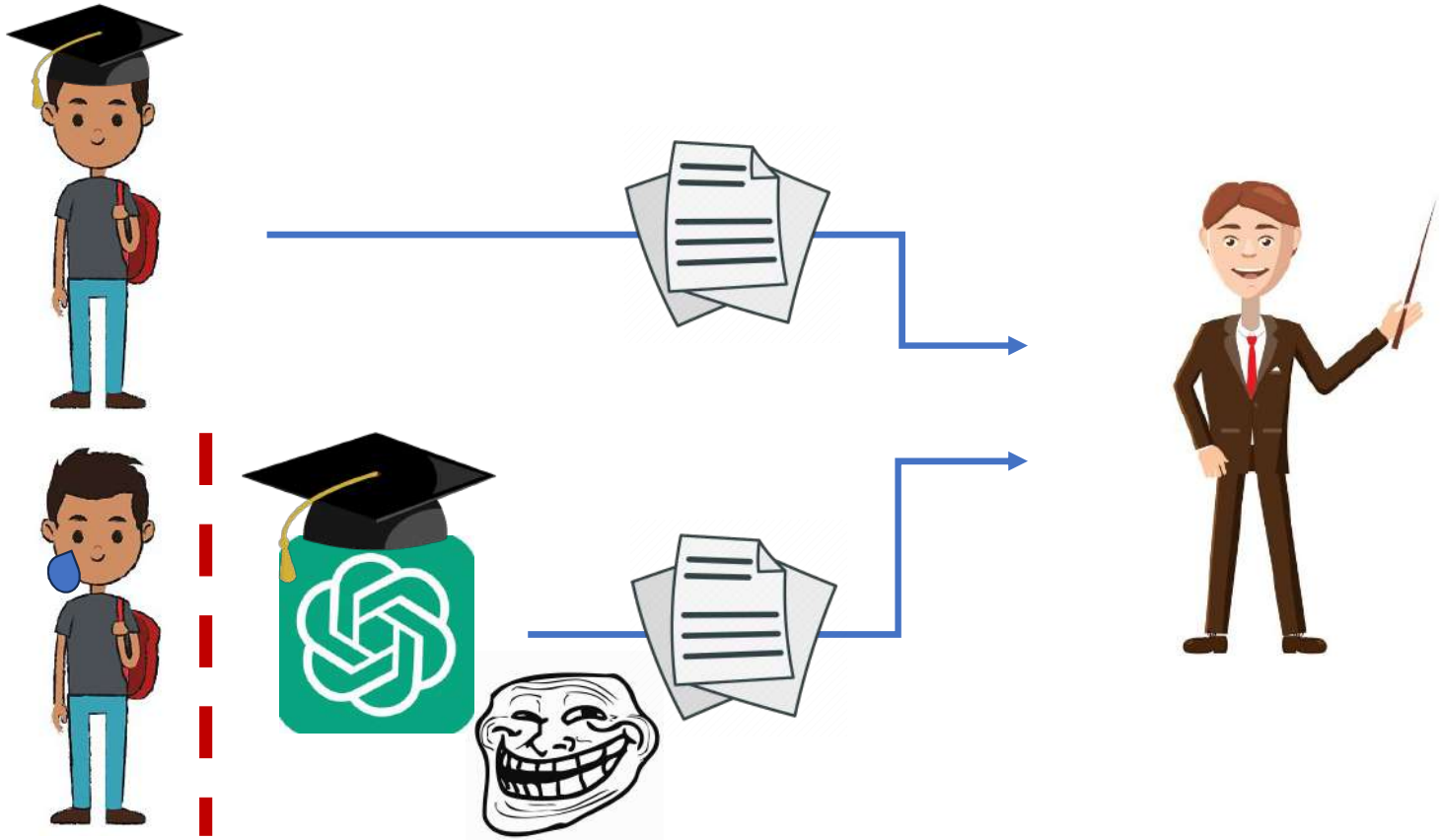
# Elephant in the room

# Chat GPT use case

# Generative mechanism



# Possible scenario



# Different AI, same problem

George R.R. Martin Among 17 Top Authors Suing OpenAI, Alleging ChatGPT Steals Their Works: 'We Are Here to Fight'

By Todd Spangler

ChatGPT vs Human Writers: Which is Better for Your Content Needs?

# 3 rules of technology

- 1: When you invent a new technology, you uncover a new class of responsibilities
- 2: If the tech confers power, it starts a race
- 3: If you do not coordinate, the race ends in tragedy

# Social media

- ”Give everyone a voice”
- ”Connect with your friends”
- ”Join like-minded communities”

Is social media free?

# No free lunch

There is always a price.

If there is no price to pay, **you are the price**.

# Data driven word

# First contact with AI – Social media

- Information Overload
- Addiction
- Doomscrolling
- Influencer Culture
- Sexualization
- Shortened attention spans
- Polarization
- Bots, DeepFakes
- Fake news

Engagement maximalization

# Engagement maximalization

GDP

Elections

Politics

National security

Reaching Customers

Media and Journalism

# Second contact with AI – Generative AI

2017

## AI fields before

- Computer Vision
- Speech recognition
- Robotics
- Music Generation
- Speech Synthesis
- Image generation

## AI fields after

- Language processing

Generative Large Language Multi-modal Modell

GLLMM

Transformer model

# Godfather of AI

'Godfather of AI' Geoffrey Hinton quits Google and warns over dangers of misinformation

"I console myself with the normal excuse: If I hadn't done it, somebody else would have."

Dr. Geoffrey Hinton

# Chat GPT mistake

One car travelling to Portland at 80 miles per hour, and travels a distance of 14 miles. Another car starts at the same point and travelling tha same route at 40 miles per hour. How far does the second car travel?

To find out how far the second car travels, you can use the formula:

Distance = Speed x Time

First, let's find out how long it takes for the first car to travel 14 miles at 80 miles per hour:

Time for first car = Distance / Speed = 14 miles / 80 mph = 0.175 hours

Now, you can use this time to calculate how far the second car travels at 40 miles per hour:

Distance for second car = Speed x Time for second car = 40 mph x 0.175 hours = 7 miles

So, the second car travels a distance of 7 miles.

# Optional Homework #2 (10 points)

- Chat with Chat GPT
- Find a mistake and document it.

Link:
https://chat.openai.com/auth/login

ELTE | FACULTY OF INFORMATICS

# Thank you for your attention!

ELTE | FACULTY OF INFORMATICS

# Bioinspired Collective Intelligence

**Introduction to Machine Learning**
Computer Science BSc Course, ELTE Faculty of Informatics

László Gulyás

Associate Professor
Department of Artificial Intelligence
lgulyas@inf.elte.hu

ELTE | IK
DEPARTMENT OF ARTIFICIAL INTELLIGENCE

# Biologically Inspired Methods –

- Brain
- Evolution
- Social Insects

- Artificial Neural Networks
- Evolutionary Algorithms
- **Swarm Intelligence Methods**

# Swarm Intelligence
(Biologically inspired self-organisation)

- A collective system capable of performing complex tasks in a dynamic and changing environment without any external control or central coordination
  - Capable of achieving a collective performance that cannot normally be achieved by the organism alone
  - On this basis, we can build a natural model that is suitable for distributed problem solving

# Properties of Swarm Intelligence

- Distributed, massively parallel (many agents)
  - Individual agents are simple and disposable (‚cheap')
  - Typically partially stochastic (i.e., non-deterministic)
- Intelligence ~ Optimising / Performing a task
  - Understood in a stochastic sense
    - Approximations
    - Different runs may give slightly different results
  - Typically ***continuous optimisation*** / performance of a series of tasks (no stopping)
- Robustness
  - Adapts to changing environments (and performs well)
    - Cf. Continuous optimisation
  - Graceful degradation
    - Withstands removal of agents (potentially many of them)

# „Collective Mind”

# Beehive Finding
## Louis Rosenberg (Unanimous AI)

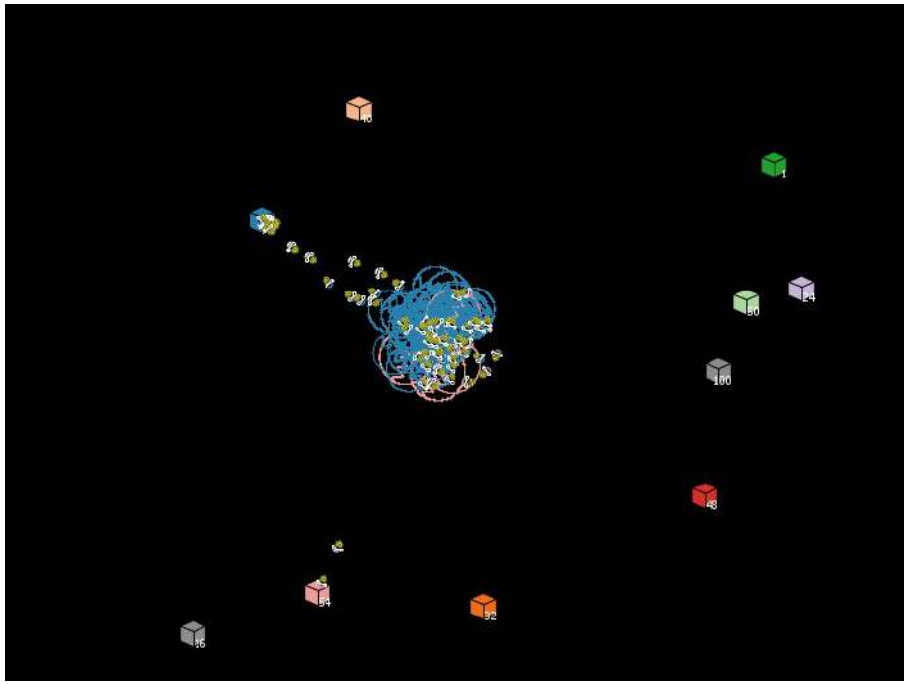
# Beehive Finding (Decision Making)

## Model (Guo and Wilensky, 2014)

- Hive site attributes
  - Location, Color, Quality
- Waggle dance
  - Length proportional to site quality

  →Higher chance idle bees see it
  → follow to inspect the site
  - New visitors also advertise site (dance)
  - Upon revisits, interest in site decays
- When ‚quorum' is observed at a site
  - Bees go home to ‚pipe'
  - When piping heard, bees start to pipe, too

  → signal to stop voting & move

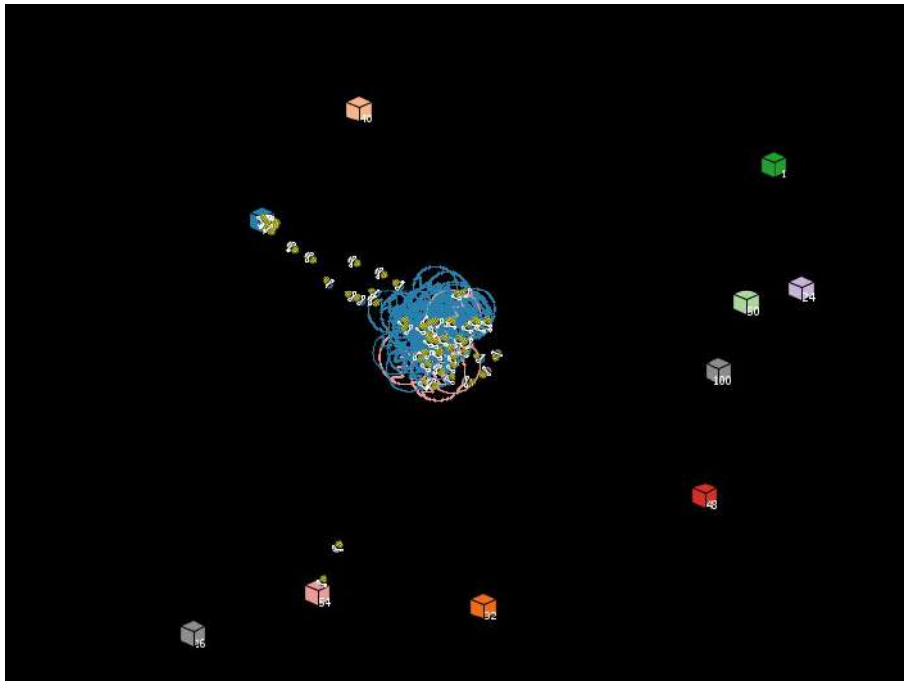


# Beehive Finding (Decision Making)

## Model (Guo and Wilensky, 2014)

Typical progression of bee states

- Initial scouts:
  - Discover →
  - Inspect-hive →
  - Go-home →
  - Dance →
  - Re-visit →
  - Pipe
- Non-initial scouts:
  - Watch-dance →
  - Re-visit →
  - Inspect-hive →
  - Go-home →
  - Dance →
  - Re-visit →
  - Pipe

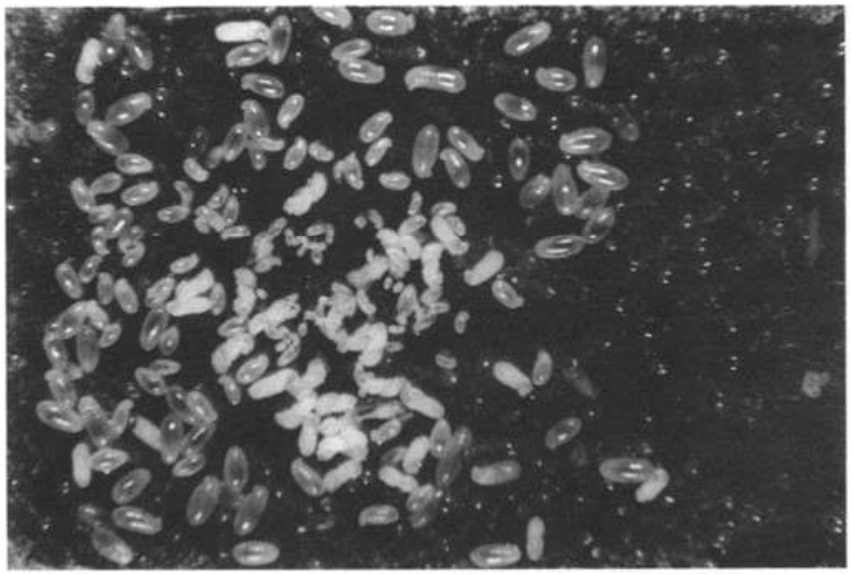


# Spatial Clustering

Ant Sort

# Brood Sorting / Cemetery organisation

## Clustering by Ants



© Franks & Sendova-Franks (1992)

© Jost et al. (2007)

# Brood Sorting / Cemetery organisation

## System Behavior

The ant colony keeps

- larvae, eggs, cocoons & food
- sorted by kind

E.g., an egg hatches, the larva is moved to the area for larvae

In several species of ants,

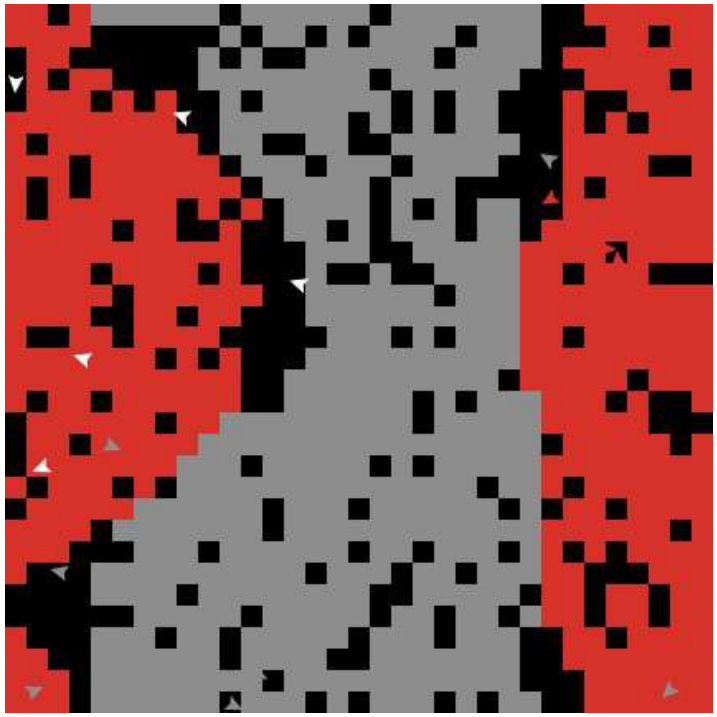
- workers form piles of corpses (cemeteries)

to clean the nests.

# Brood Sorting / Cemetery organisation

## Generic Desctiption, variants

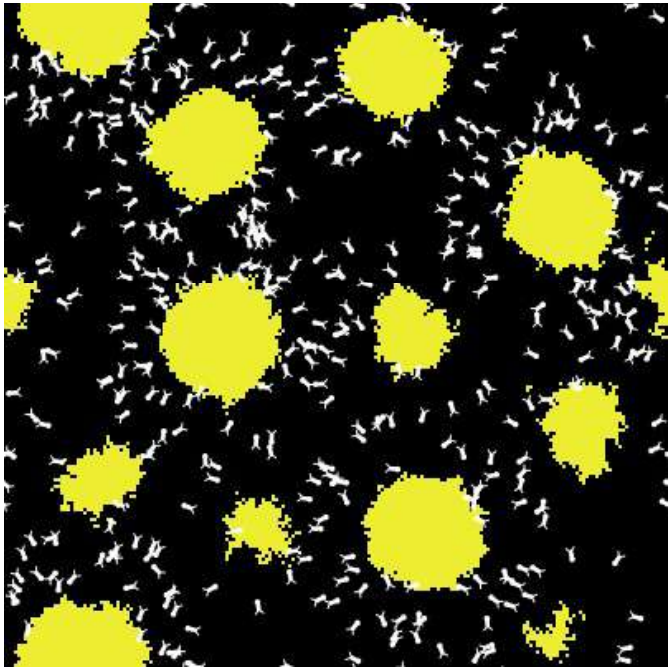
- 2D Grid
- One object at a place/cell or piles?
- #Types/Colors (1 or more)
- Ants can step on objects
- Multiple ants can occupy the same cell



# Brood Sorting / Cemetery organisation

## Simple Algorithm

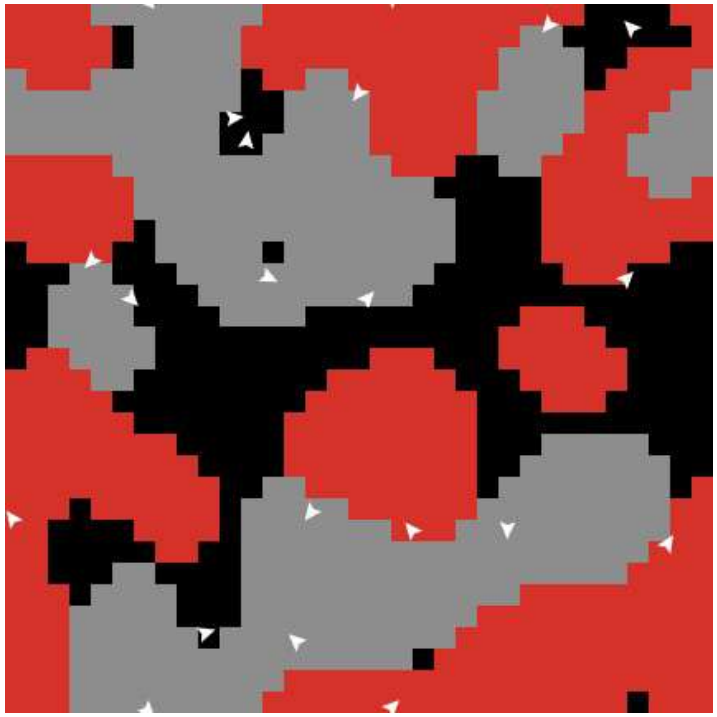
- Ants wander randomly on 2D grid
- They observe their immediate environment
- Pick up an object
  - From ‚low-density' locations (when not carrying)
- Put it down
  - At ‚high density' locations (if carrying)



# Brood Sorting / Cemetery organisation

## Model (Simple)

- One object at a place
- #Types: may vary
- Environment:
  - 8 Immediate cells (cf. Schelling)
- Decision making
  - #same color items (@location or carried)
  - Below / Above threshold (3, 4, 5)
- Tuning / Difficulty
  - Optimal threshold depends on
    - Density
    - #types
  - Non-optimal threshold
    - Early convergence (no clustering)
    - Ever-carrying ants



# Brood Sorting / Cemetery organisation
## Applications

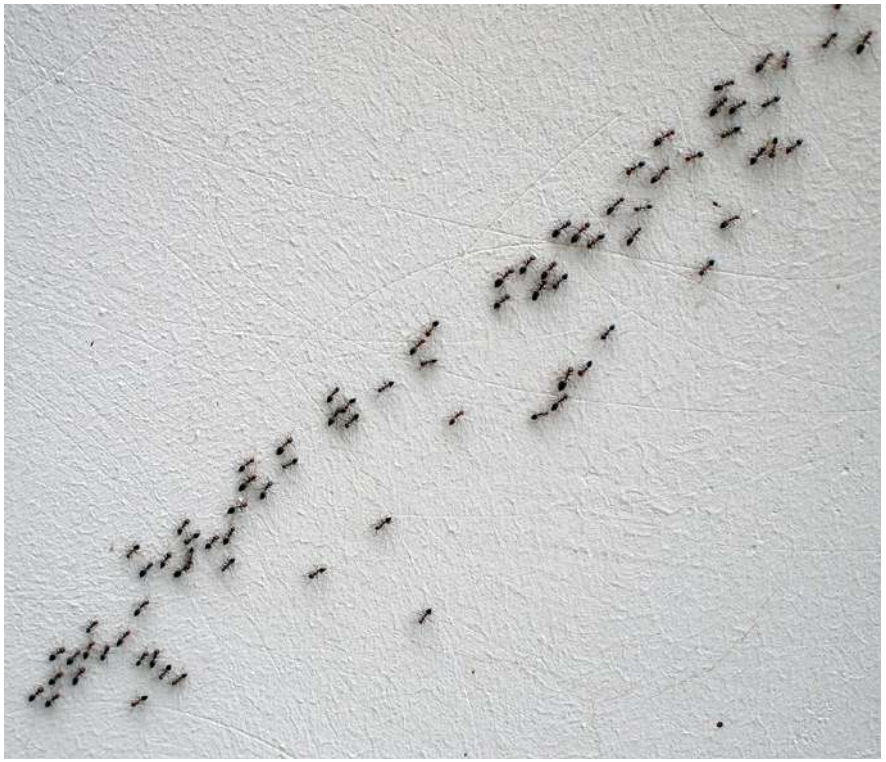
- Distributed Data Mining
  - Data & Text retrieval
  - Graph partitioning
  - Mostly in networked environments
  - *Mostly before the Deep Learning boom*
- <u>Swarm robotics</u>
  - Most recent applications
- Edge computing
  - Mobile agents
- Using mostly variations of the classic model
  - Lumer and Faieta (1994)
  - Gutowitz (1993)
  - Etc.

# Shortest Path

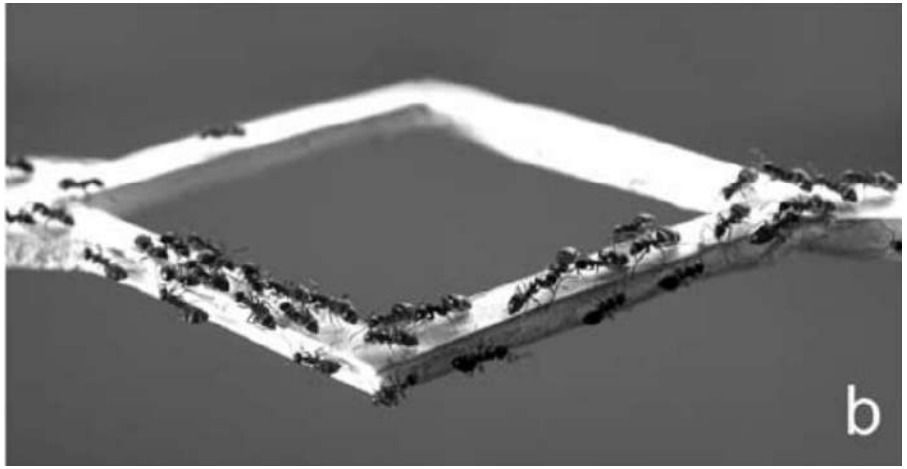
Ant Foraging

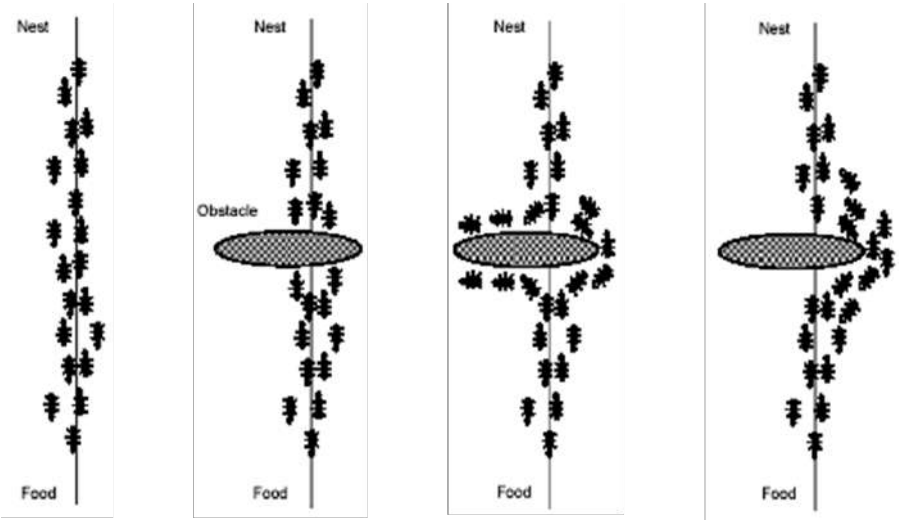
# Ant Foraging
## Observations

- Ants tend to ‚march' in line
  - Carrying food
- They tend to pick the closest food source first
- Direct line is shortest
  - From food source to nest



# Ant Foraging
## Observations



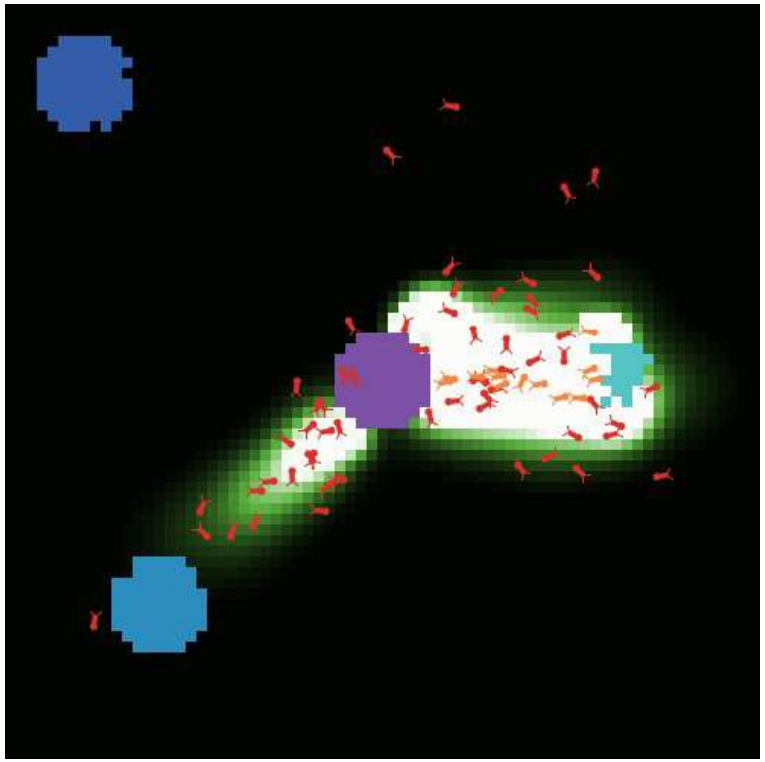



# Ant Foraging
## Observation (Workings)

- It's all about **pheromones**
  - Fragrant, volatile substance
  - Triggers social response
  - In members of the same species
- Word coined by Peter Karlson & Martin Lüsche (1959)
  - From Ancient Greek φέρω (phérō) 'to bear' and hormone
- Ants lay 2 different pheromones
  - One when searching for food (‚A')
  - One when carrying food (‚B')
- Ants follow gradient (stochastically) of the ‚other' pheromone
  - ‚B' when searching for food
  - ‚A' when homing
- Pheromone evaporates
  - Non-reinforced trails fade
  - Time/effort to reinforce trail is *proportional to length*

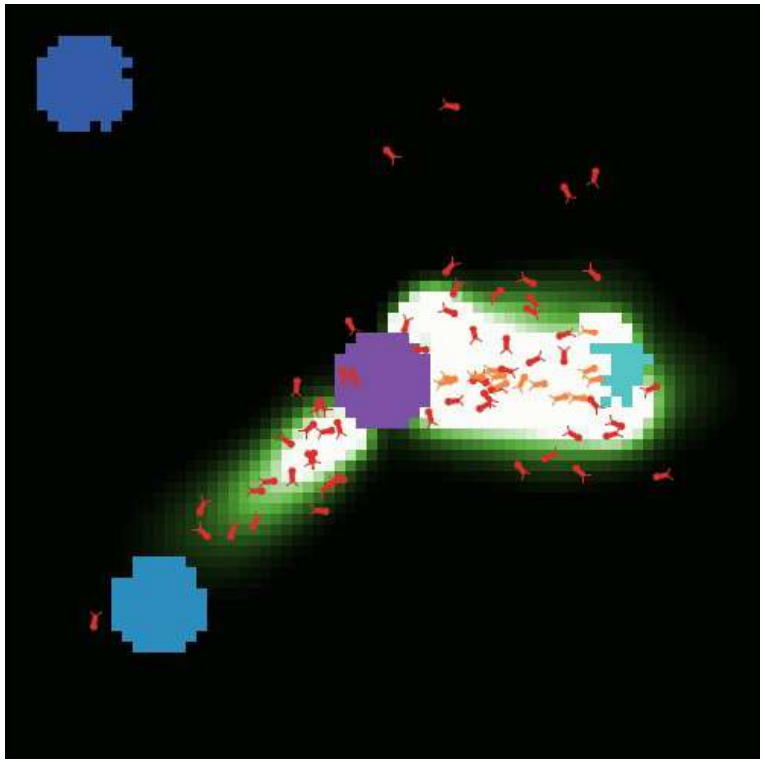
# Ant Foraging
## Model

- Ants wander around on 2D grid
  - Starting from nest
  - Avoid obstacles (if any)
  - Follow pheromone gradient (*probabilistically*)
- Pick up a piece of food
  - From their current location,
    if not carrying
- Put down food
  - If at nest and carrying
- Ants deposite a unit of pheromone
  - At current location
    One when searching for food (‚A')
  - One when carrying food (‚B')
- Pheromone
  - Diffuses and evaporates
  - By constant rate, uniformly across space



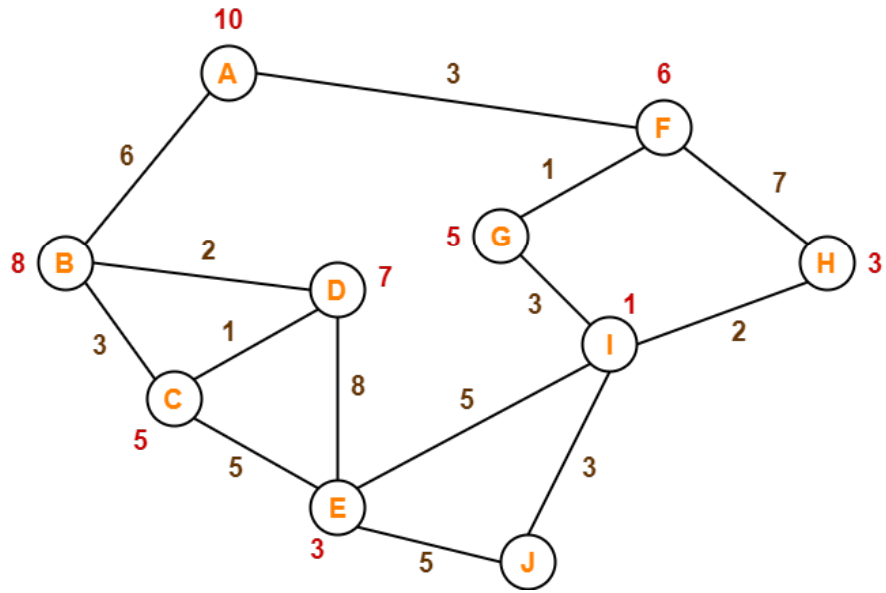
# Ant Foraging
## Model (Alternative)

- Pheromone ‚A' (nest scent)
  - Comes from the nest
  - Makes homing easier
  - But less general (i.e., hard to find nest from a larger distance)



# Ant Foraging
## Applications

- Many
- Finding shortest path is a very generic problem
  - Routing problems (telco, vehicles, etc.)
  - Finding approximate solutions (graph search) for large problems
  - Scheduling problems
  - Quadratic Assignment problems
  - Travelling Salesman Problem



# Ant Foraging

## Applications (Generalization: Marco Dorigo, 1992)

<u>Ant Colony Optimization (ACO)</u>

- Graph (network) instead of 2D grid
  - Pheromone accumulates on edges
- Starting node selected at random
  - Next node is selected <u>*probabilistically*</u> following edge pheromone gradient
- When a solution is found
  - Pheromone amounts on path are adjusted: proportionally to quality
- Stopping condition
  - When most ants select the same solution

**Marco Dorigo**
**Inventor of ACO**

# Summary:
# Swarm Intelligence

- Surveyed collective, massively multi-agent algorithms for specific problems
  - Biologically inspired
  - Cheap and simple individual behaviours
  - Massive number of agents
  - Continuous operations, graceful degradation

- Simple behavior may be enough for emerging intelligent behavior

ELTE | FACULTY OF INFORMATICS

# Thank you!


## László Gulyás

Associate Professor
Department of Artificial Intelligence
lgulyas@inf.elte.hu


ELTE | IK
DEPARTMENT OF ARTIFICIAL INTELLIGENCE

ELTE | FACULTY OF INFORMATICS

# BACKUP

ELTE | IK
DEPARTMENT OF ARTIFICIAL INTELLIGENCE

# Flocking / Schooling

# Flocking / Swarming / Schooling

© Hakai Institute

# Flocking / Swarming / Schooling

## System Behavior

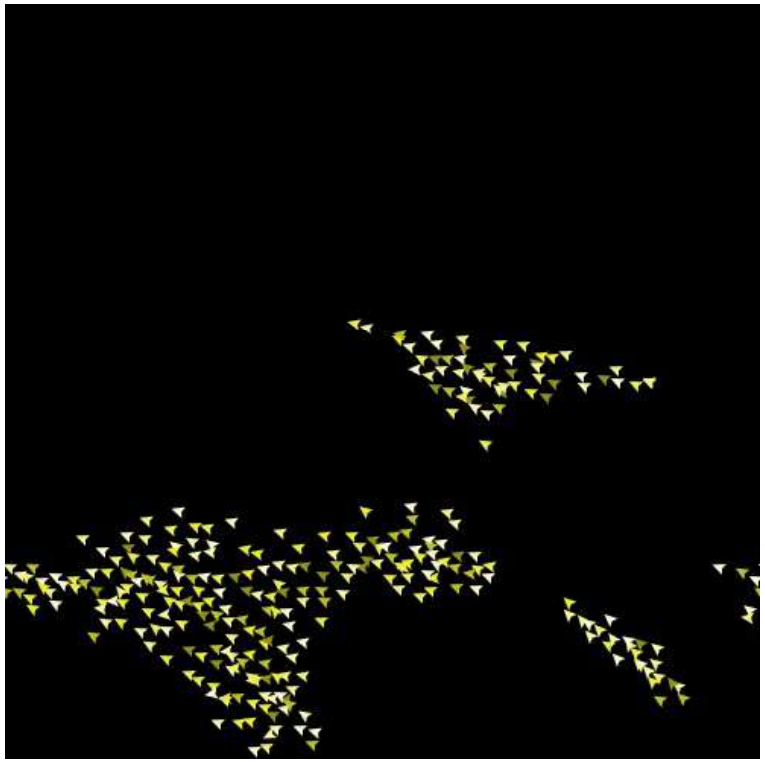
Flocks of birds, shools of fish

- stay together,
- coordinate turns, and
- avoid collisions with obstacles and each other

# Flocking / Swarming / Schooling
## Model

Three simple rules:
[Reynolds 1987, Heppner 1990]

1. Maintain a specified minimum separation from the nearest object or other birds
2. Match velocity (magnitude and direction) to nearby birds
3. Stay close to the center of the flock



# Flocking / Swarming / Schooling
## Applications

In human societies similar problems
- in air-traffic control,
- convoys of ships,
- **movement of robots, drones!**

Conventional solutions depend on
- sophisticated communication and
- central coordination, plus
- **cannot handle the density of coordinated entities**

# 1. Lecture quiz

**Due** Sep 21 at 4:17pm **Points** 2 **Questions** 2
**Available** after Sep 21 at 4pm **Time Limit** 5 Minutes

# Instructions

You have 5 minutes to complete this quiz.

It also acts as a catalog, so if you don't take it, it will mean that you skipped class. 0 point is also a valid proof of attendance.

## Attempt History

| | Attempt | Time | Score |
|---|---|---|---|
| **LATEST** | Attempt 1 | 2 minutes | 1.2 out of 2 |

Score for this quiz: **1.2** out of 2
Submitted Sep 21 at 4:14pm
This attempt took 2 minutes.

### Question 1

**1 / 1 pts**

AI's performance (today) is subhuman because

- it can't deal with high-dimensional inputs
- Correct! it can be fooled by low-amplitude noise
- it can't combine old-fashioned AI and deep learning

### Question 2

**0.2 / 1 pts**

Machine learning includes the following algorithmic components (among

others)

Correct! ☑ deep learning

orrect Answer ☐ unsupervised learning

ou Answered ☑ artificial general intelligence

Correct! ☑ reinforcement learning

orrect Answer ☐ rule-based systems

Correct! ☑ supervised learning

ou Answered ☑ artificial intelligence

Quiz Score: **1.2** out of 2

# 2. Lecture quiz

**Due** Sep 28 at 5:30pm **Points** 2 **Questions** 2 **Time Limit** 5 Minutes

## Instructions

You have 5 minutes to complete this quiz.

It also acts as a catalog, so if you don't take it, it will mean that you skipped class. 0 point is also a valid proof of attendance.

## Attempt History

| | Attempt | Time | Score |
|---|---|---|---|
| LATEST | Attempt 1 | 4 minutes | 2 out of 2 |

Score for this quiz: **2** out of 2
Submitted Sep 28 at 5:30pm
This attempt took 4 minutes.

### Question 1

**1 / 1 pts**

- If X is a discrete stochastic variable that can take values {x1, x2, ...xn} with probabilities p(x1), ...p(xn), respectively, then what is the expected value of X?

p(xn)*xn

Correct!

Sum(p(xi)*xi) for every i

Prod(p(xi)+xi) for every i

Sum(p(xi)/xi) for every i

### Question 2

**1 / 1 pts**

- Let the weights of a neuron be $w \in R^n$ (w is element of R^n) the input, $x \in R^n$ (x is element of R^n) and the threshold $b \in R$ (b is element of R). What is the output of the neuron (y)?

I don't know.

y = wT*x +b

y = w*x -b

Correct!

y = $w^T$*x -b

Quiz Score: **2** out of 2

# 3. Lecture quiz

**Due** Oct 5 at 4:15pm **Points** 2 **Questions** 2 **Available** after Oct 5 at 4pm
**Time Limit** 5 Minutes

## Instructions

You have 5 minutes to complete this quiz.

It also acts as a catalog, so if you don't take it, it will mean that you skipped class. 0 point is also a valid proof of attendance.

## Attempt History

| | Attempt | Time | Score |
|---|---|---|---|
| **LATEST** | Attempt 1 | 3 minutes | 0.5 out of 2 |

Score for this quiz: **0.5** out of 2
Submitted Oct 18 at 2:32pm
This attempt took 3 minutes.

### Question 1

**0.5 / 1 pts**

What makes possible to transfer the scientific discoveries of 20000 years in 20 years to a child? (multiple answers)

Correct! ☑ Component learning

Correct! ☑ Linear proofs of combinatorial problems

☐ The evolution of the human brain in the last 100 years.

ou Answered ☑ Improvement in teaching methods.

## Question 2

0 / 1 pts

What is the difference between creativity and intelligence?

orrect Answer

Creativity is the formation of new concepts; intelligence is the cognitive manipulation of the concepts.

They are practically the same.

Intelligence is about to create new concepts; creativity is the manipulation of the concept.

ou Answered

Creativity is only used in human sciences, while intelligence is used in natural sciences.

Quiz Score: **0.5** out of 2

# 4. Lecture quiz

**Due** Oct 13 at 8pm **Points** 2 **Questions** 2 **Time Limit** 5 Minutes

## Instructions

You have 5 minutes to complete this quiz.

It also acts as a catalog, so if you don't take it, it will mean that you skipped class. 0 point is also a valid proof of attendance.

## Attempt History

| | Attempt | Time | Score |
|---|---|---|---|
| LATEST | Attempt 1 | 1 minute | 2 out of 2 |

Score for this quiz: **2** out of 2
Submitted Oct 13 at 7:39pm
This attempt took 1 minute.

### Question 1

1 / 1 pts

The size of the representation of a sparse autoencoder "THIS IS YOUR ANSWER" as/than the dimension of the input.

- the same
- Correct! Larger
- smaller
- is either larger, smaller or the same

### Question 2

1 / 1 pts

What does the Chinese Room example state?

- Lookup tables can make sense of the representation
- Feedforward processing can make sense of the representation
- Correct! Feedforward processing can not make sense of the representation
- The homunculus cannot make sense of the representation

Quiz Score: **2** out of 2

# 6. Lecture quiz

**Due** Oct 26 at 11:59pm **Points** 2 **Questions** 2
**Available** Oct 26 at 7pm - Oct 26 at 11:59pm about 5 hours **Time Limit** 5 Minutes

## Instructions

You have 5 minutes to complete this quiz.

It also acts as a catalog, so if you don't take it, it will mean that you skipped class. 0 point is also a valid proof of attendance.

This quiz was locked Oct 26 at 11:59pm.

## Attempt History

| | Attempt | Time | Score |
|---|---|---|---|
| LATEST | Attempt 1 | less than 1 minute | 2 out of 2 |

Score for this quiz: **2** out of 2
Submitted Oct 26 at 9:31pm
This attempt took less than 1 minute.

### Question 1

**1 / 1 pts**

The Markov assumption means that:

- the optimal strategy can be stochastic
- The transition probability matrix is a function of the decision series
- Reward is a function of the transition probability matrix
- past does not count in the decision making process

Correct!

## Question 2

1 / 1 pts

What is the 'reward' in Reinforcement Learning (Goal-Oriented System)?

Correct!

- a real number
- a real vector
- a nonnegative real number
- a nonnegative real vector

Quiz Score: **2** out of 2

# 7. Lecture quiz

**Due** Nov 2 at 4:11pm **Points** 2 **Questions** 2
**Available** until Nov 2 at 4:53pm **Time Limit** 5 Minutes

## Instructions

You have 5 minutes to complete this quiz.

It also acts as a catalog, so if you don't take it, it will mean that you skipped class. 0 point is also a valid proof of attendance.

This quiz was locked Nov 2 at 4:53pm.

## Attempt History

| | Attempt | Time | Score |
|---|---|---|---|
| **LATEST** | Attempt 1 | 3 minutes | 2 out of 2 |

Score for this quiz: **2** out of 2
Submitted Nov 2 at 4:28pm
This attempt took 3 minutes.

### Question 1

**1 / 1 pts**

Which nonlinearity can produce outputs larger than 1?

- Sigmoid
- Tanh
- I don't know.
- ReLU

Correct!

## Question 2

1 / 1 pts

What does a pooling layer do?

- It performs maximum operation
- Correct! It can perform either maximum or averaging operation among other things
- It performs averaging
- It does maximum and averaging operations

Quiz Score: **2** out of 2

# 8. Lecture quiz

**Due** Nov 9 at 5:30pm **Points** 2 **Questions** 2
**Available** until Nov 9 at 5:30pm **Time Limit** 5 Minutes

## Instructions

You have 5 minutes to complete this quiz.

It also acts as a catalog, so if you don't take it, it will mean that you skipped class. 0 point is also a valid proof of attendance.

This quiz was locked Nov 9 at 5:30pm.

## Attempt History

| | Attempt | Time | Score |
|---|---|---|---|
| **LATEST** | Attempt 1 | 3 minutes | 0.67 out of 2 |

Score for this quiz: **0.67** out of 2
Submitted Nov 9 at 5:30pm
This attempt took 3 minutes.

### Question 1

**0.67 / 1 pts**

Which of the following is/are classical dimensionality reduction method(s)?

Correct!
- [x] PCA
- [ ] CCP

Correct!
- [x] MDS
- [ ] RT-PCR

orrect Answer
- [ ] LLE

## Question 2

0 / 1 pts

Why do we need low-dimensional embedding and/or distance preservation and/or noise filtering? (multiple answers)

orrect Answer

- [ ] to be able to learn to control faster we need low dimensional embedding, distance preservation, and noise filtering

ou Answered

- [x] for improving the smoothness of temporal changes we need distance preservation

ou Answered

- [x] to ease object recognition in deep learning

Correct!

- [x] for enabling reinforcement learning , because it scales exponentially with the number of variables

Correct!

- [x] to overcome combinatorial explosion we need low-dimensional embedding

Quiz Score: **0.67** out of 2

# Midterm

**Due** Oct 19 at 4:40pm **Points** 18 **Questions** 18
**Available** Oct 19 at 4:07pm - Oct 19 at 4:40pm 33 minutes **Time Limit** 25 Minutes

# Instructions

During the quiz completion, we will automatically monitor if they leave the Canvas site.
**Anyone who is found to change window/page during the quiz will have their quiz invalidated.**
A short time will be allowed for the midterm, 25 minutes. This should be more than enough time to complete the series of questions, but if we see that this fails en masse, the scores will be corrected.
**You will see one question at a time, and once answered, the questions will be locked.**
The results and solutions will be made public after the midterm is solved.
**Since the midterm will be very short, you cannot leave the room until the end. If you have to leave the room your exam is over.**

This quiz was locked Oct 19 at 4:40pm.

## Attempt History

| | Attempt | Time | Score |
|---|---|---|---|
| LATEST | Attempt 1 | 18 minutes | 9 out of 18 |

Score for this quiz: **9** out of 18
Submitted Oct 19 at 4:32pm
This attempt took 18 minutes.

### Question 1 — 1 / 1 pts

What is the curse of dimensionality?

Correct!

- (selected) search space increases exponentially with the number of dimensions
- the dimension of the search space decreases with the number of variables
- 

in a bounded world, the volume of a sphere of radius 1 increases with the number of dimensions

the larger the dimension of the problem, the larger the number of dependent variables

## Question 2

1 / 1 pts

General Intelligence is the capability to "THIS IS YOUR ANSWER"

- approximate a function
- maximize reward
- solve a fixed set of problems
- Correct! solve arbitrary new problems

## Question 3

1 / 1 pts

What is the autoencoder-based resolution of the homunculus fallacy?

- The homunculus is immaterial
- The decoded input makes sense of the input by representing it
- The input makes sense of the representation by decoding it
- Correct! The representation makes sense of the input by reconstructing it

## Question 4

0 / 1 pts

What is the starting point of the homunculus fallacy?

- making sense is internal to the brain
- ou Answered: representation should make sense
- orrect Answer: somebody should make sense of the representation
- making sense is possible

## Question 5

0 / 1 pts

Which of the following is NOT part of a Reinforcement Learning (Goal-Oriented System) setting?

- ou Answered: environment
- orrect Answer: target variable
- model
- strategy

## Question 6

0 / 1 pts

What is the explanation of the Necker cube illusion?

- there are delays in action processing

orrect Answer

- only one interpretation is available for consciousness at a time

ou Answered

- there are delays in sensory processing
- there are delays in sensory and in action processing

## Question 7

1 / 1 pts

The size of the representation of a sparse autoencoder "THIS IS YOUR ANSWER" as/than the dimension of the input.

- the same
- smaller

Correct!

- typically larger
- is either larger, smaller or the same

## Question 8

0 / 1 pts

Sign of transposition is '. We have a squared matrix Q. All matrix elements are positive

ou Answered

- Matrix Q'Q is positive semidefinite

orrect Answer

- Matrix Q'Q is positive definite or positive semidefinite
- Matrix Q'Q is positive definite

## Question 9

0 / 1 pts

In the case of Hebbian learning the source of learning is "THIS IS YOUR ANSWER".

ou Answered

- Input and output of the neuron multiplied
- The activity value of the output of the neuron
- The activity value of the input to the neuron

orrect Answer

- Input and output of the weight multiplied

## Question 10

1 / 1 pts

Why can deep networks be fooled so easily?

- gradient descent can be stuck in local minima

Correct!

- they work with many dimensional inputs
- they work with many parameters to be optimized
- they work with many layers to be optimized

## Question 11

1 / 1 pts

What is the dualist resolution of the homunculus fallacy?

- Outputs are the results of feedforward processing

Measurement theory of quantum physics resolves the fallacy

Correct!

The homunculus is immaterial

The homunculus estimates the output

## Question 12

0 / 1 pts

The average of independent, identically distributed stochastic variables converges to the "THIS IS YOUR ANSWER" when the number of variables approaches infinity.

variance

ou Answered

standard deviation

orrect Answer

expected value

Gaussian distribution

## Question 13

0 / 1 pts

It follows from the No Free Lunch Thereom that

orrect Answer

Random search is just as good as any other method for the full set of optimization problems

ou Answered

Random search is the best method amongst all search problems.

orrect Answer

No algorithmic solution therefore offers a "short cut".

orrect Answer

Intelligence is not efficient on all kinds of optimization problems

## Question 14

0 / 1 pts

What is "false belief"?

orrect Answer

if our observations are thought to be known by the partner

ou Answered

if the belief is not supported by facts

if the belief about the "other mind" is false

if arguments to explain a phenomenon are false

## Question 15

1 / 1 pts

What is crowdsourcing?

Correct!

many people work for a common goal, often innovation, problem solving, or efficiency

a company gives the task to another that gives to another and so on, forming a chain

many tasks are solved sequentially (after each other) in several optimizations

many tasks are solved simultaneously in one optimization

## Question 16

1 / 1 pts

Vector A = [ 2, 3, 4] , B = [ 3, 4, 5] What is A*B (dot product)?

Correct!

38

orrect Answers

38 (with margin: 0)

## Question 17

1 / 1 pts

What is the average of the following numbers? [2,5,4,5]

Correct!

4

orrect Answers

4 (with margin: 0)

## Question 18

0 / 1 pts

Given the following random variable x:

P(x=10) = 0.3

P(x=20) = 0.7

What is the expected value of x?

ou Answered

16

orrect Answers

17 (with margin: 0)

Quiz Score: **9** out of 18

Can a complex movement program be inherited?

- (●) Only for primates
- ( ) No
- ( ) Only for humans
- ( ) Yes, in general

**Question 2** 1 pts

Which one is the third industrial revolution?

- (●) Production using computers and internet
- ( ) Production using deep learning
- ( ) Production using artificial intelligence
- ( ) Production using conveyor belt

Which one is the third industrial revolution?

//canvas.elte.hu/courses/11425/quizzes/25940/take 110%

**Question 3** 1 pts

What is "false belief"?

- ( ) if arguments to explain a phenomenon are false
- ( ) if the belief about the "other mind" is false
- (●) if our observations are thought to be known by the partner
- ( ) if the belief is not supported by facts

What is false belief?

**Question 4** 1 pts

Shallow multilayer feedforward networks

- ( ) can not approximate concave sets
- ( ) can not approximate non-convex sets
- (●) are linear function approximators
- ( ) are universal approximators

Shallow multilayer feedforward networks

In the case of Hebbian learning the source of learning is "THIS IS YOUR ANSWER".

- ( ) the activity value of the output of the neuron
- ( ) the activity value of the input to the neuron
- ( ) input and output of the weight multiplied
- ( ) input and output of the neuron multiplied

In the case of Hebbian learning the source of learning is

**Input and output of the weight multiplied**

| | |
|---|---|
| The size of the representation of a sparse autoencoder "THIS IS YOUR ANSWER" as/than the dimension of the input.<br><br>○ is either larger, smaller or the same<br>○ the same<br>○ smaller<br>○ larger | The size of the representation of a sparse auto encoder<br><br>larger |
| **Question 8** 1 pts<br><br>What does the complexity of an optimization task depend on?<br><br>○ the number of dependent variables of the task<br>○ the wall-clock time to solve<br>○ the number of independent variables of the task<br>○ the number of available processors | What does the complexity of an optimization task depend on?<br><br>the synthesis of energy systems is a complex optimization task depending on multiple time series |
| | |
| What is the curse of dimensionality?<br><br>○ the dimension of the search space decreases with the number of variables<br>○ search space increases exponentially with the number of dimensions<br>○ the larger the dimension of the problem, the larger the number of dependent variables<br>○ in a bounded world, the volume of a sphere of radius 1 increases with the number of dimensions | lecture 3 - 10th slide<br><br>What is the curse of dimenstionality? |
| **Question 10** 1 pts<br><br>What is missing from typical Deep Networks that could promote explainability?<br><br>○ synchronous operation<br>○ recognition by components<br>○ holistic recognition<br>○ few layer recognition | What is missing from typical Deep Networks that could promote explainability?<br><br>Answer: **Recognition by components**. |
| | |

vas.elte.hu/courses/11425/quizzes/25940/take 110%

What does the Chinese Room example state?

- Lookup tables can make sense of the representation
- Feedforward processing can make sense of the representation
- The homunculus can not make sense of the representation
- Feedforward processing can not make sense of the representation

What does the Chinese Room example state?
(this picture has the wrong answer)
Correct answer:
**Feedforward processing can not make sense of the representation**

**Question 12** 1 pts

What is the Markov assumption?

- The transition-probability matrix depends on state-action-state triplets
- The strategy is a probability distribution on the state-action space
- Past actions need to be included into the strategy
- The past does not influence the current optimal decision

**What is the Markov assumption?**

Complexity classes are "THIS IS YOUR ANSWER"

- easy to solve, hard to solve, easy to prove, hard to prove
- hard to prove, easy to prove
- hard to verify, easy to verify
- hard to solve, easy to solve, hard to verify, easy to verify

https://duongquangduc.wordpress.com/2017/02/13/complexity-classes/

Complexity classes are

**Question 14** 1 pts

Which feature can corrupt the Markov property?

- the strategy is a function of state and action
- an action depends on the past
- an action depends on the state
- a state depends on the past

Which feature can corrupt the Markov property?

A state depends on the past

In the Necker cube illusion "THIS IS YOUR ANSWER"

- one can not switch the illusions, but can stop one interpretation
- one can switch the illusions, and can stop one interpretation
- ◉ one can not switch the illusions, and can not stop one interpretation
- one can switch the illusions, but can not stop one interpretation

In the Necker cube illusion

„we cant control which one we see, and cant save one of the interpretation, cant stop changing either."

**Question 16** 1 pt

How do you train a deep network for superresolution?

- Input downscaled image, output blurred image
- Input downscaled image, output real image
- ◉ Input real image, output downscaled image
- Input upsampled image, output downsampled image

How do you train a deep network for superresolution?

Super-resolution is the process of recovering a high-resolution (HR) image from a low-resolution (LR) image. Super-resolution models are trained with LR images as input and HR images as target.

How can you improve the Markov property?

- ◉ by learning the transition-probability matrix
- by the optimization of the decisions
- by including the past into decision making
- by learning the reward function

how can you improve the Markov property? Answer: by including the past into decision making.

**Question 18** 1 pts

What is crowdsourcing?

- many tasks are solved simultaneously in one optimization
- many tasks are solved sequentially (after each other) in several optimizations
- a company gives the task to another that gives to another and so on, forming a chain
- many people work for a common goal, often innovation, problem solving, or efficiency

What is crowdsourcing?

What is the dualist resolution of the homunculus fallacy?

- The homunculus estimates the output
- The homunculus is immaterial
- Outputs are the results of feedforward processing
- Measurement theory of quantum physics resolves the fallacy

A dualist might argue that the homunculus inside the brain is an immaterial one

What is the dualist resolution of the homunculus fallacy?

What is the explanation of the Necker cube illusion?

- there are delays in action processing
- there are delays in sensory processing
- there are delays in sensory and in action processing
- only one interpretation is available for consciousness at a time

What is the explanation of the Necker cube illusion?

Answer: **There are delays in sensory and in action processing**

What is the 'reward' in Reinforcement Learning (Goal-Oriented System)?

- a nonnegative real vector
- a real number
- a nonnegative real number
- a real vector

What is the reward in Reinforcement learning?

Answer: a real number

Why can deep networks be fooled so easily?

- they work with many layers to be optimized
- they work with many parameters to be optimized
- gradient descent can be stuck in local minima
- they work with many dimensional inputs

Why can deep networks be fooled so easily?

Answer:**They work with many dimensional inputs**

What can be the reason for the imitation of facial expression in neonates?

- It generates association between sensory and emotional signals
- It helps to understand the mood of the others
- It helps others to understand the mood
- It helps to show facial expressions

What can be the reason for the imitation of facial expression in neonates?

Answer: **It generates association between sensory and emotional signals.**

There are more than one component types. One of them is Lego-like: you can put the components together and take them apart. Another is modifiable. The representation of space and the related metric belong to "THIS IS YOUR ANSWER"

- both
- neither of them (a third one)
- the modifiable components
- Lego-like components

There are more than one component types. One of them is Lego like:

Answer:both

A Deep Neural Network is a differentiable function of "THIS IS YOUR ANSWER".

Group of answer choices

two variables (parameter, input)

three variables (hyper-parameter, parameter, input)

a single variable (input)

four variables (meta-parameter, hyper-parameter, parameter, input)

General Intelligence is the capability to "THIS IS YOUR ANSWER"

Group of answer choices

maximize reward

solve arbitrary new problems

approximate a function

solve a fixed set of problems

General Intelligence is the capability to:

Answer: solve arbitrary new problems

What is ConceptNet?

Group of answer choices

set of answers for the question "what is it used for?"

collection of concepts

collection of captions of images

collection of relations between words

What is ConceptNet:

Answer: Collection of relations between words

Which of the following is NOT part of a Reinforcement Learning (Goal-Oriented System) setting?

Group of answer choices

environment

strategy

target variable

model

Which of the following is NOT part of a Reinforcement learning(Goal-Oriented System) setting?
Answer: Model

A Deep Neural Network is a differentiable function of **two variables(parameter,input).**

What can be the reason for the imitation of tongue protrusion? It should be an answer related to breastfeeding.

What is the relation between a fully convolutional network (i.e., there are no dense layers) and the weight values of the templates?
**Weight values of the templates correspond to some of the weights of the CNN.**

Can a complex movement program be inherited?
**Yes, in general.**

Which one is the third industrial revolution?
**Production using computers and Internet.**

Paper Folding
**Fold a paper of 0.1 mm thickness 50 times. How thick will that be?**

Much Larger. (Other possible answers were: 100m, Earth-Moon distance, 100km)

Communication:
**What is the learning task that is worth to communicate/talk about?**
Hard to learn and easy to verify

Neural Response:
**What can be the reason for the similar neural responses when the motion is executed by the self or by others?**
It reinforces the making sense process: the muscle related neurons ‘confirm’ that they could execute the same action

Tongue Protrusion:
**What can be the reason for tongue protrusion?**
It is part if the general imitation system

Homunculus Fallacy:
**What is the autoencoder-based resolution of the homunculus fallacy?**
The representation makes sense of the input by reconstructing it.

**What is the starting point of the homunculus fallacy?**
Somebody should make sense of the representation.

# machine learning cheatsheet

This attempt took 32 minutes.

**Question 1** 1 / 1 pts

What is gradient descent?

- Computation of the loss
- Computation of the gradient
- Computation of the weight update
- Computation of the output

---

what is gradient descent

nal exam 2021. 12. 29. IK

·idze Laila (TPX2HJ) submitted Dec 29, 2021 at 9:03am

## Question 2

1 / 1 pts

What is the curse of dimensionality?

- the larger the dimension of the problem, the larger the number of dependent variables
- search space increases exponentially with the number of dimensions
- the dimension of the search space decreases with the number of variables
- in a bounded world, the volume of a sphere of radius 1 increases with the number of dimensions

what is the curse of dimensionality

# Final exam 2021. 12. 29. IK

Beridze Laila (TPX2HJ) submitted Dec 29, 2021 at 9:03am

## Question 3

1 / 1 pts

What is crowdsourcing?

- a company gives the task to another that gives to another and so on, forming a chain
- many tasks are solved sequentially (after each other) in several optimizations
- many tasks are solved simultaneously in one optimization
- many people work for a common goal, often innovation, problem solving, or efficiency

crowdsourcing

Partial

## Question 4

0.33 / 1 pts

Which of the following is /are part of a Reinforcement Learning (Goal Oriented System) setting?

- [ ] strategy
- [ ] target variable
- [x] environment
- [ ] model

reinforcement learning

Incorrect

## Question 5

0 / 1 pts

What is feedback control?

- error-to-control mapping
- output-to-input mapping
- control-to-error mapping
- state-to-error mapping

feedback control

## Question 6

1 / 1 pts

What is the relation between a fully convolutional network (i.e., there are no dense layers) and the weight values of the templates?

- There is a one-to-one relation between them
- I don't know.
- Nothing
- Weight values of the templates correspond to some of the weights of the CNN

convolutional network

Incorrect

## Question 7

0 / 1 pts

How does the Fourier transform of the normal A sound (440 Hz) look like?

- nonzero everywhere
- nonzero around 440 Hz, zero elsewhere
- zero around 440 Hz, nonzero elsewhere
- zero everywhere

fourier transform a sound 440 hz

Incorrect

## Question 8

0 / 1 pts

What is feedforward control?

- control-to-state mapping
- state-to-control mapping
- control-to-error mapping
- output-to-input mapping

feedforward control

## Question 9

1 / 1 pts

How can you improve the Markov property?

- by learning the reward function
- by the optimization of the decisions
- by including the past into decision making
- by learning the transition-probability matrix

markov property

Incorrect

## Question 10

0 / 1 pts

What does the complexity of an optimization task depend on?

- the wall-clock time to solve
- the number of independent variables of the task
- the number of dependent variables of the task
- the number of processors

complexity of optimization

Incorrect

**Question 11** 0 / 1 pts

What does a pooling layer do?

- It does maximum and averaging operations
- It can perform either maximum or averaging operation among other things
- It performs maximum operation
- It performs averaging

Add a

Sav

pooling layer

Incorrect

**Question 12** 0 / 1 pts

What is the ReLU function?

- 0 on negative axis, 1 on positive axis
- 0 on negative axis, -1 on positive axis
- 0 on negative axis, identity on positive axis
- -1 on negative axis, 1 on positive axis

relu function

## Question 14

1 / 1 pts

What is the result of average_pooling(A), where pooling window = 2x2, stride=2, and

A =

| | | | |
|---|---|---|---|
| 2 | 3 | 4 | 1 |
| 2 | 3 | 5 | 2 |
| 2 | 3 | 8 | 1 |
| 2 | 3 | 6 | 3 |

- [4.5, 2.5, 3, 2]
- [2.5, 3, 2.5, 4.5]
- [2, 3, 3, 4.5]
- [2.5, 3.1, 4.6, 2]

average pooling window stride

## Question 15

1 / 1 pts

Vector A = [ 3, 4, 5] , B = [1, 2, 3] What is A*B (dot product)?

26

dot product

Incorrect

## Question 16

0 / 1 pts

We are working with an artificial neuron. Let w^T * x=6 ;b = 4; What is the output of the neuron, if we use ReLu as the nonlinearity function?

4

Incorrect

## Question 17

0 / 1 pts

What is the convolution of [0, 1, 3, 4, 2] with [2, 0, 1] (without padding)?

- [2, 6, 8]
- [3, 6, 8]
- [3, 5, 8]
- [3, 6, 9]

convolution padding

## Question 18

3 / 3 pts

What is an autoencoder and what can it be used for? The first 10 words are rated.

Your Answer:

it compresses the input in a much smaller dimension and then decompresses it and tries to get it as close to the input as possible. Is used for image compression, image denoising etc.

autoencoder

**Question 19** 1.5 / 3 pts

Why is it difficult to legislate against artificial intelligence? List three problems.

Your Answer:

Algorithm transparency

Cybersecurity vulnerabilities

unfairness

legislate against artificial intelligence

**Question 20** 3 / 3 pts

What tasks can artificial intelligence perform? What are the professions it can replace? List at least three!

Your Answer:

AI can be used for automating business process, gaining insight through data analysis,engaging with customers and employees.

it can replace

Customer service executives. Customer service executives don't require a high level of social or emotional intelligence to perform. Bookkeeping and data entry. Receptionists.

artificial intelligence perform