EFOP-3.4.3-16-2016-00009

A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

ARTIFICIAL INTELLIGENCE

Authors:

Tibor Dulai, Ágnes Werner-Stark

SZÉCHENYI 2020

Európai Unió
Európai Szociális Alap

MAGYARORSZÁG KORMÁNYA

BEFEKTETÉS A JÖVŐBE

EFOP-3.4.3-16-2016-00009

A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

ARTIFICIAL INTELLIGENCE

1. Machine Learning - Introduction

Authors:
Tibor Dulai, Ágnes Werner-Stark

SZÉCHENYI 2020

MAGYARORSZÁG KORMÁNYA

Európai Unió
Európai Szociális Alap

BEFEKTETÉS A JÖVŐBE

Introduction

EFOP-3.4.3-16-2016-00009
A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

Elon Musk
@elonmusk
Follow

Worth reading Superintelligence by Bostrom. We need to be super careful with AI. Potentially more dangerous than nukes.

Reply Retweet Favorite More

REUTERS
Business Markets World Politics TV

SUSTAINABLE BUSINESS APRIL 17, 2019 / 1:33 AM / 7 DAYS AGO

Microsoft turned down facial-recognition sales on human rights concerns

Elon Musk
@elonmusk
Follow

China, Russia, soon all countries w strong computer science. Competition for AI superiority at national level most likely cause of WW3 imo.

5:33 AM - 4 Sep 2017

The New York Times

'The Business of War': Google Employees Protest Work for the Pentagon

Introduction

EFOP-3.4.3-16-2016-00009

A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen


Like 15.8M

Wednesday, Apr 24th 2019

Mail Online

Home | News | U.S. | Sport | TV&Showbiz | Australia | Femail | Health | Science | Money | Video

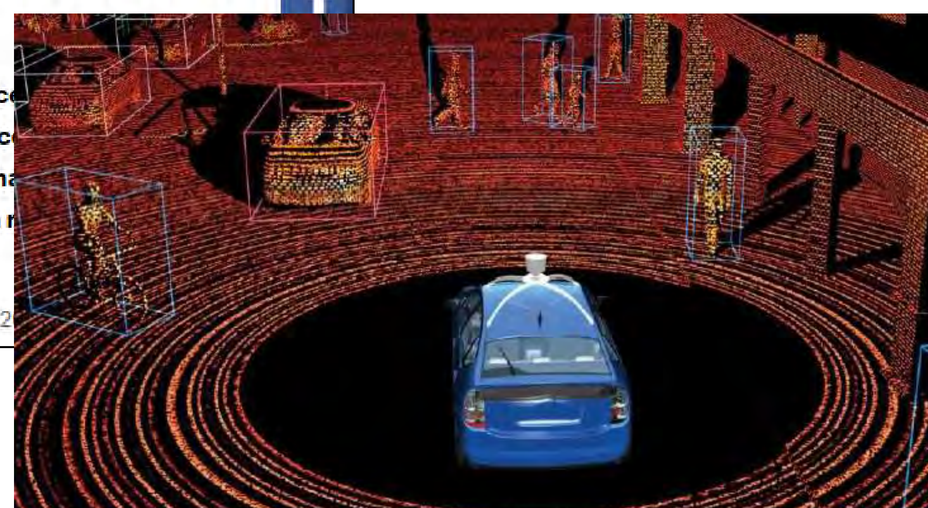
Latest Headlines | Science | Discounts

Alibaba's AI outperforms humans in one of the toughest reading comprehension tests ever created in a remarkable world first

- AI was created by retail firm Alibaba's Institute of Data Science
- It took part in Stanford Question Answering Dataset reading c
- It scored 82.44 in the exact answer category beating the huma
- The company said it's the first time a machine has out-done a r

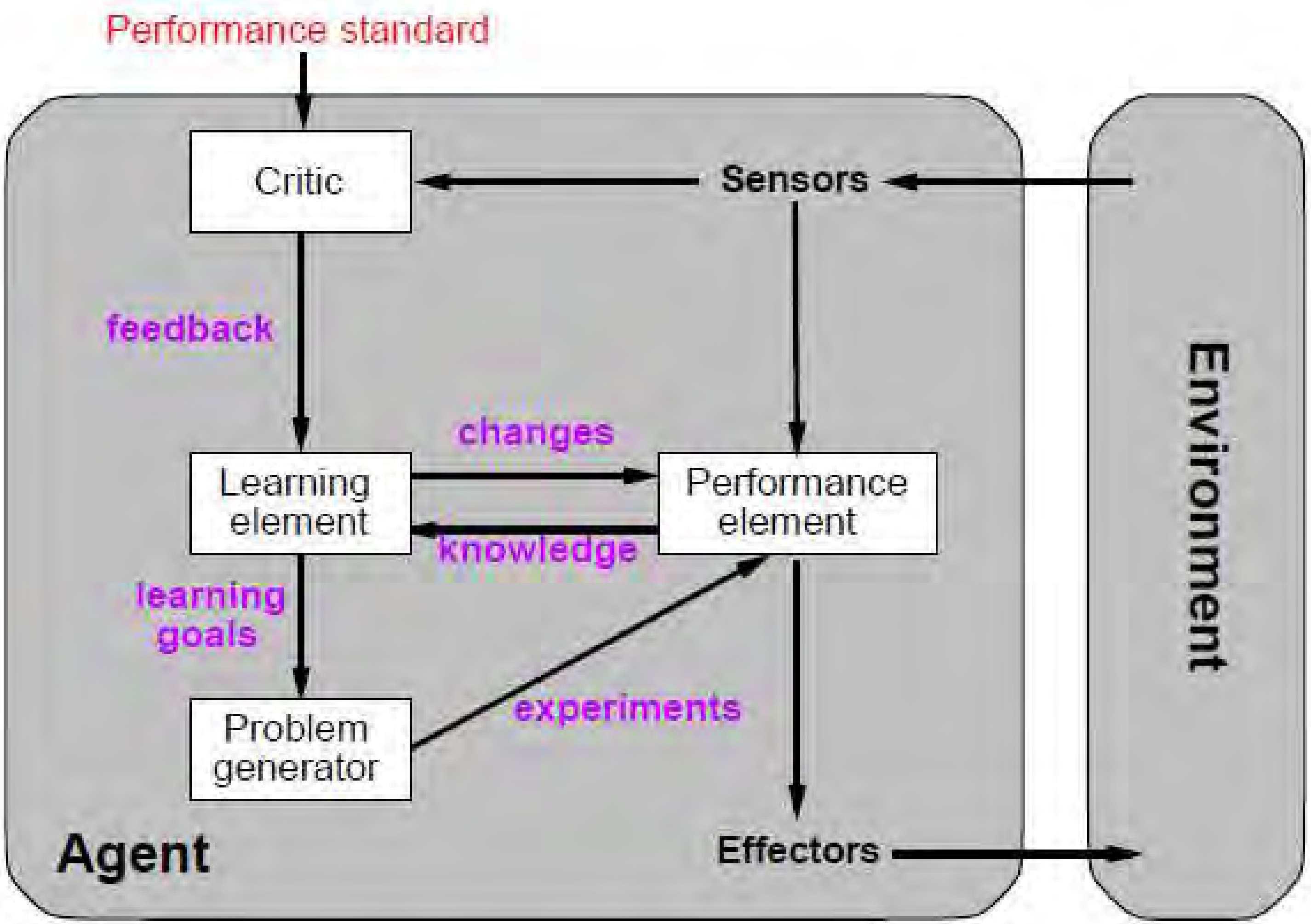
By TIM COLLINS FOR MAILONLINE

PUBLISHED: 15:47 BST, 15 January 2018 | **UPDATED:** 21:47 BST, 15 January 2



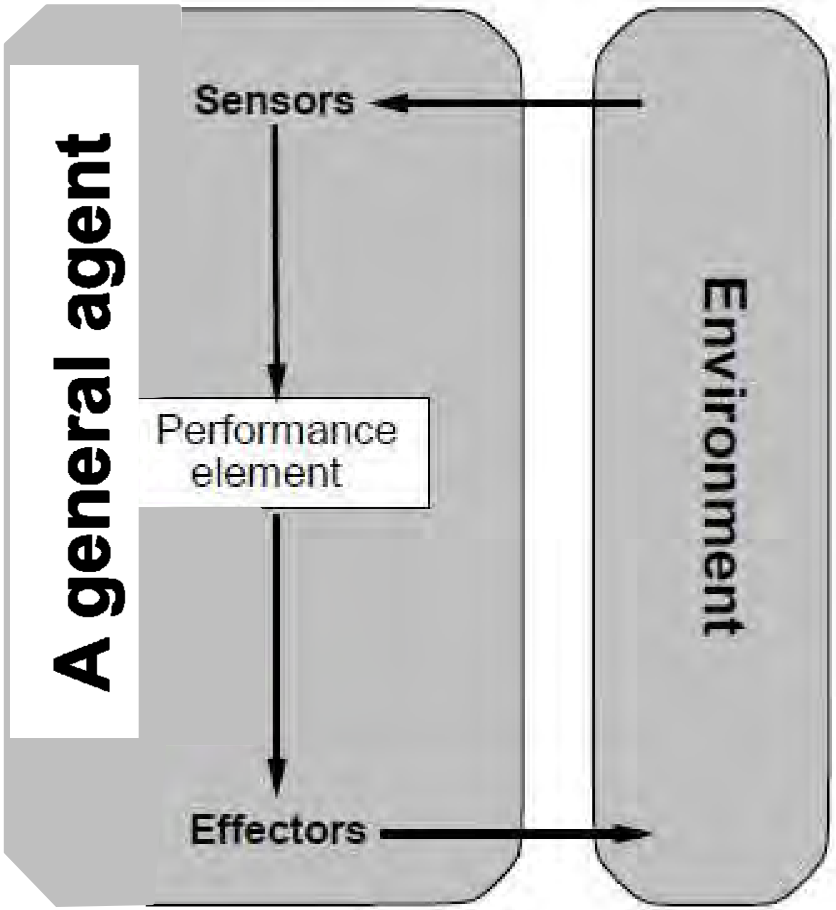
Learning Agent

The structure of a Learning Agent



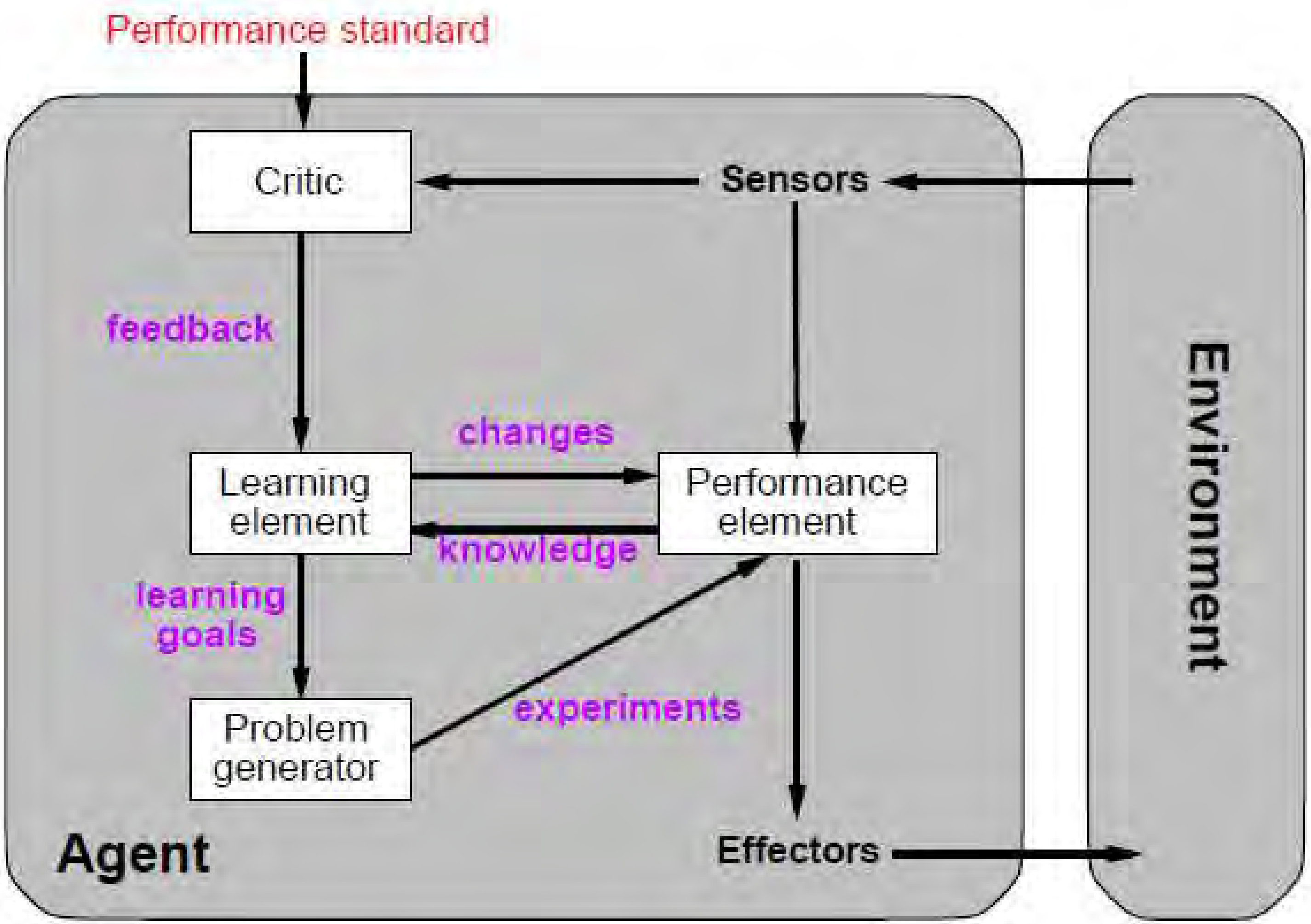
The structure of a Learning Agent

EFOP-3.4.3-16-2016-00009
A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen



The structure of a Learning Agent

EFOP-3.4.3-16-2016-00009
A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

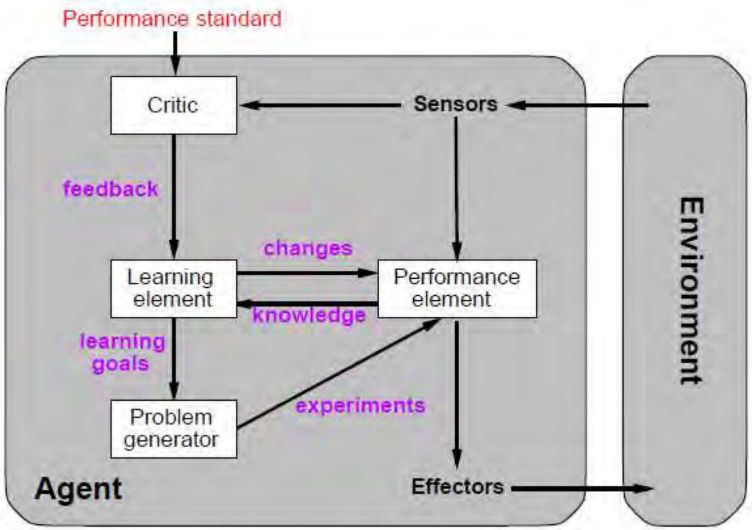


The structure of a Learning Agent

EFOP-3.4.3-16-2016-00009

A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

Main responsibilities



- **Performance element**: selects the appropriate action related to the environment: **senses** the environment **and decides on the action** to perform

- **Learning element**: its goal is to **improve the preformance and efficiency**. It has some knowledge about the performance element and it has feedback about its operation's success. Based on them it **determines how to modify the performance element for better operation**.

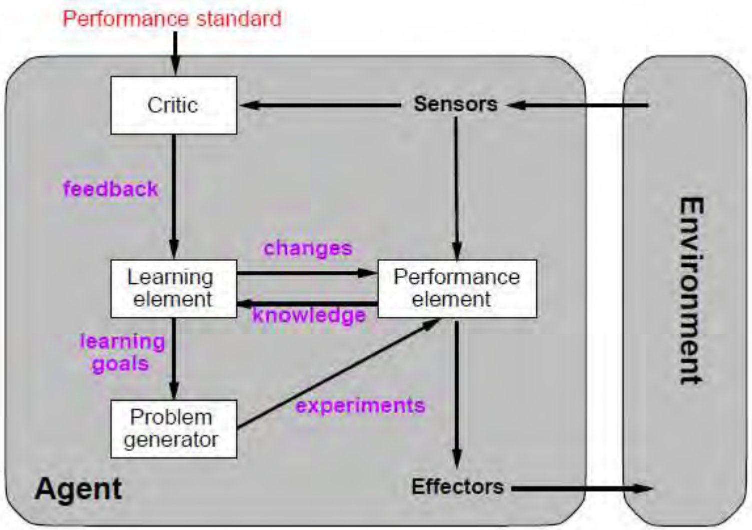
The learning element highly depends on the structure of the performance element.

The structure of a Learning Agent

Main responsibilities (contd.)

- **Critic**: **qualifies the performance** based on an outer – fixed – performance standard. **The standard has to be out of the agent** /otherwise the agent could adjust the standard to its performance/.

- **Problem generator**: suggests such actions that lead to **new experiences** (otherwise the performance element would always choose the actual best action again and again). Discovering the unknown state space, suboptimal actions in short term could lead to much better procedures in long term.



The structure of a Learning Agent

EFOP-3.4.3-16-2016-00009

A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

Example:

Actions:

- turning
- accelerating
- braking
- tooting

…

Learning element:

Intends to **learn more precise rules** about the effect of braking and accelerating.
Wants to know the behaviour of the vehicle in case of rainy/wet circumstances better.
Discovers what the other drivers get angry about.
…

The structure of a Learning Agent

EFOP-3.4.3-16-2016-00009
A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

Example:

Critic:

Informs the learning element about the observed environment.

The structure of a Learning Agent

EFOP-3.4.3-16-2016-00009

A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

Example:

Problem generator:

Experimentalizes new routes.

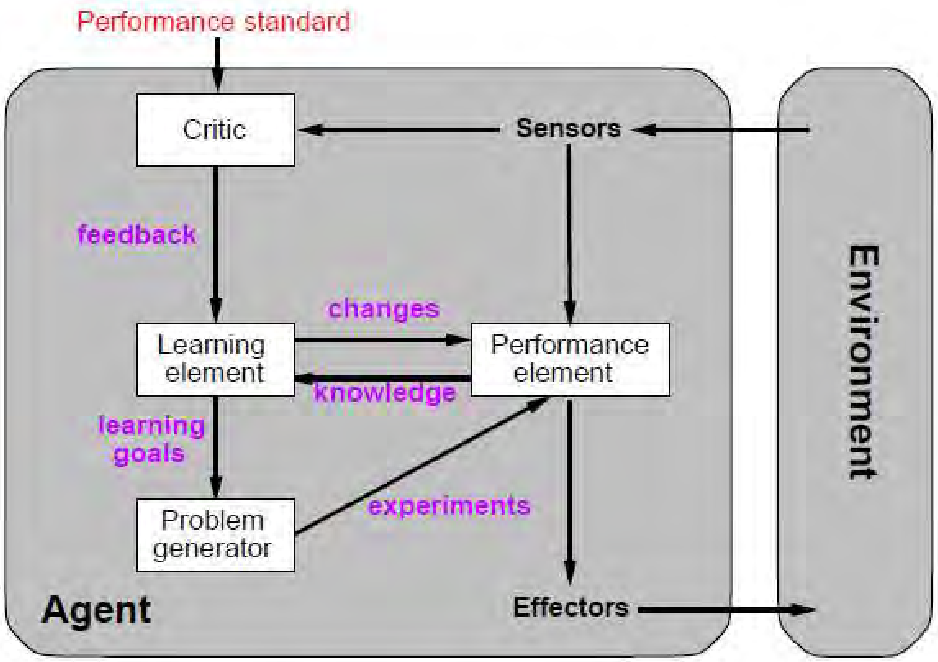
Design of a Learning Element

EFOP-3.4.3-16-2016-00009

A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

The design of a Learning Element is influenced by:

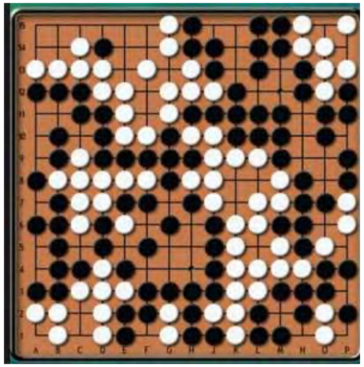
- which component of the performance element is intended to improve,
- the knowledge representation,
- the type of the feedback,
- the a priori information.



Components of the Performance element

EFOP-3.4.3-16-2016-00009
A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

Any component of the Performance Element is a function.



For example:

- Determination of the possible actions based on the state.
- Mapping the current state to action.
- Utility of the states.
- The possible results of the possible actions.
- Action value information (how much an action is desired in a certain state)

…

Components of the Performance element

EFOP-3.4.3-16-2016-00009

A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

How to learn them?

E.g.,

- when a vehicle exceeds the desired brakeforce on a slippery road, it can sense the resulted state for the chosen action

- the amount of the gratuity that a taxi driver gets from the passenger helps to learn the utility function

Knowledge representation

Any component of the Performance element can be represented in several ways, e.g., by:

Deterministic representation

- weighted linear functions

$$w_1 f_1 + w_2 f_2 + \cdots + w_n f_n$$

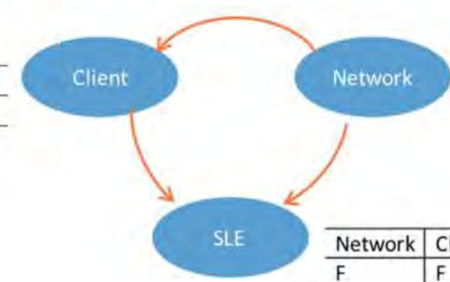
- propositional calculus

$$\exists u\, A(u) \rightarrow \exists n\, S(n, available)$$

Non-deterministic representation

- Bayesian networks

	Client	
Network	F	P
F	0.6	0.4
P	0.05	0.95



Network	
F	P
0.02	0.98

		SLE	
Network	Client	F	P
F	F	1.0	0.0
F	P	0.95	0.05
P	F	0.9	0.1
P	P	0.0	1.0

Feedback

3 types of learning based on the feedback:

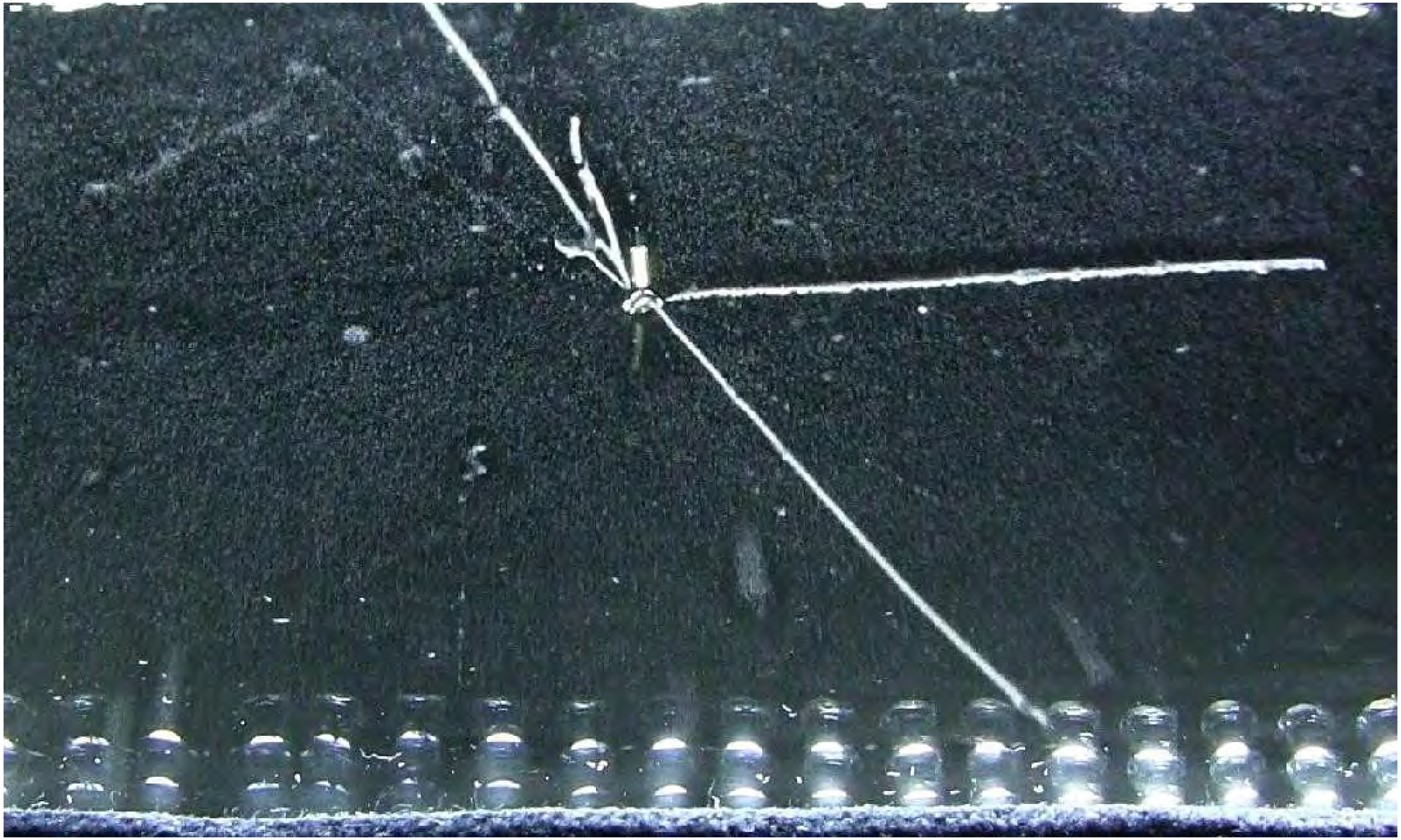
Supervised learning: **both the input and the output** of the component can be sensed.
E.g.: input /action/: braking → output /result/: stop within 15 m

Reinforcement learning: the agent **has only a kind of qualification of the action**, but the proper action is not told.
E.g.: expensive bill /reinforcement – reward or punishment/ after an accident (without the suggestion of earlier braking)

Unsupervised learning: there is no information about the proper result. Agents of unsupervised learning may realize the correspondences (so, it can be able to predict the result of an action) but they have **no information about the utility of the results** (has no information about what to do). Yet, it is enough knowledge for example for classification tasks.

A priori knowledge

What can you see in this picture?



A priori knowledge

What does come into your mind about the following text?

COMB

A priori knowledge

EFOP-3.4.3-16-2016-00009
A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

What does come into your mind about the following text?

COMB

A priori knowledge

What does come into your mind about the following text?

COMB

A priori knowledge

What does come into your mind about the following text?

COMB

Inductive Learning

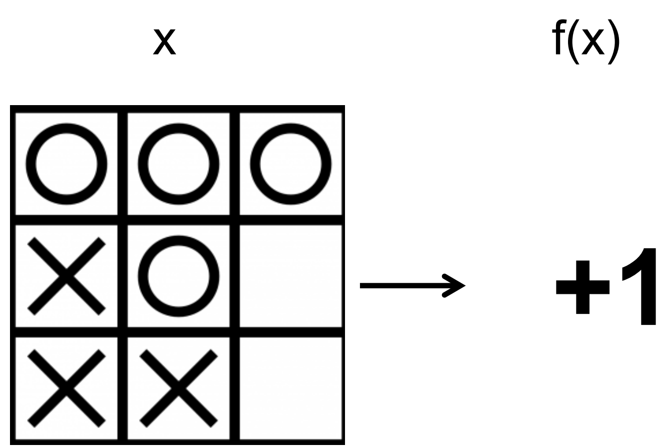
Inductive Learning

EFOP-3.4.3-16-2016-00009
A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

There is a **target function**: *f*

The goal is to learn this *f* function, but **we know the function value only in some points** of the domain of the function.

The pair of a known domain point *x* and the belonging function value *f(x)* is called **example**, e.g.



Inductive Learning

Based on the set of examples (**training set**) we intend to create function: *h,* such that *h* agrees with *f* on the training set:

$$h \approx f$$

This *h* function is called **hypothesis**.

If *h* agrees with *f* on the training set, *h* is **consistent**.

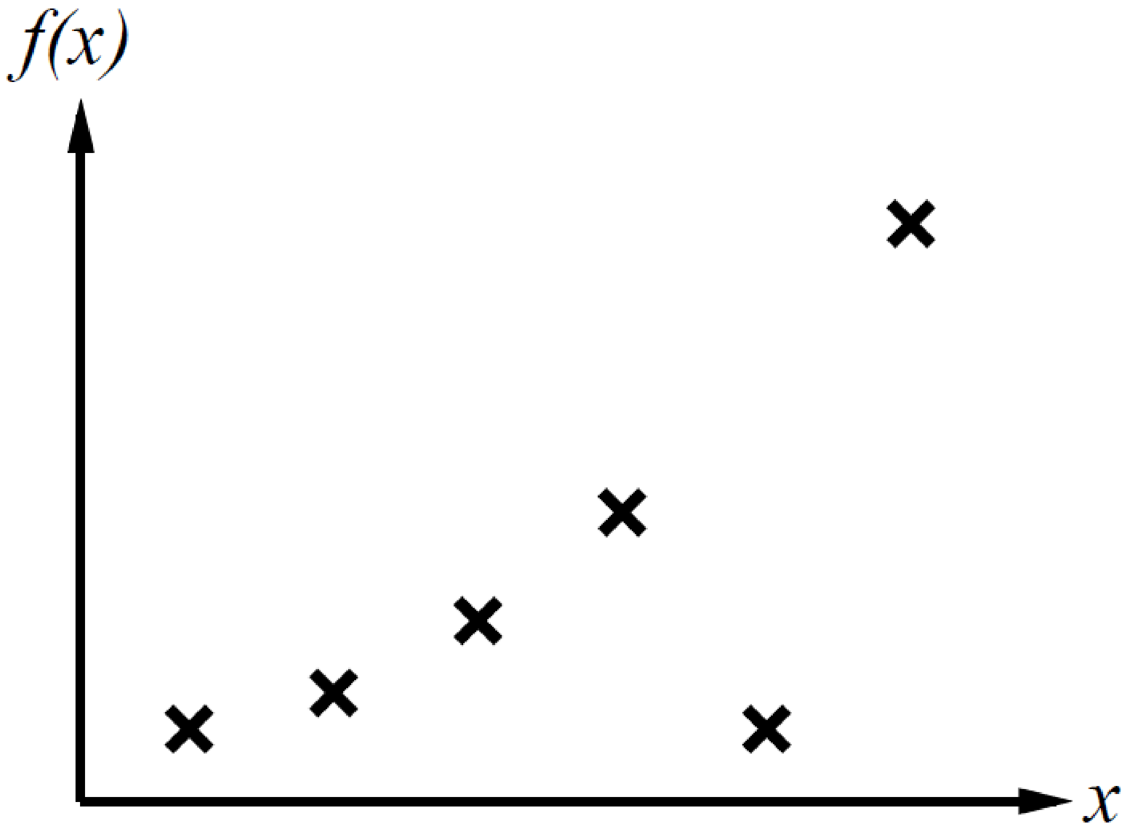
Of course, the number of possible hypothesises is ∞ if we have a finite training set.

Which one to select?

Ockham's razor: the most possible hypothesis is the simplest consistent hypothesis.
Its reason: the number of simple consistant hypothesises are much lower than the cardinality of the set of complex consistant hypothesises. That's why there is a very small chance that any simple but bad hypothesis is consistent. So, if we have two consistent hypotheses, it is more probable that the simpler is the desired one.

Inductive Learning

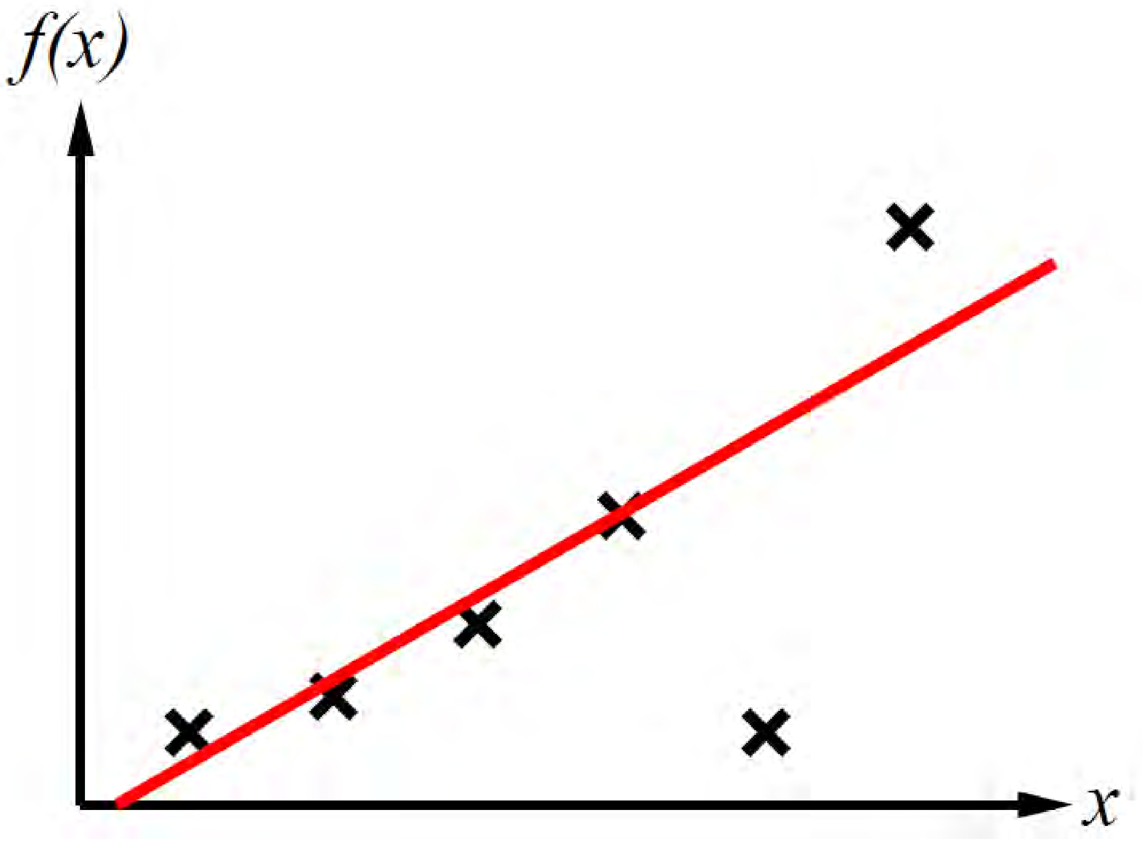
Example:



Inductive Learning

EFOP-3.4.3-16-2016-00009
A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

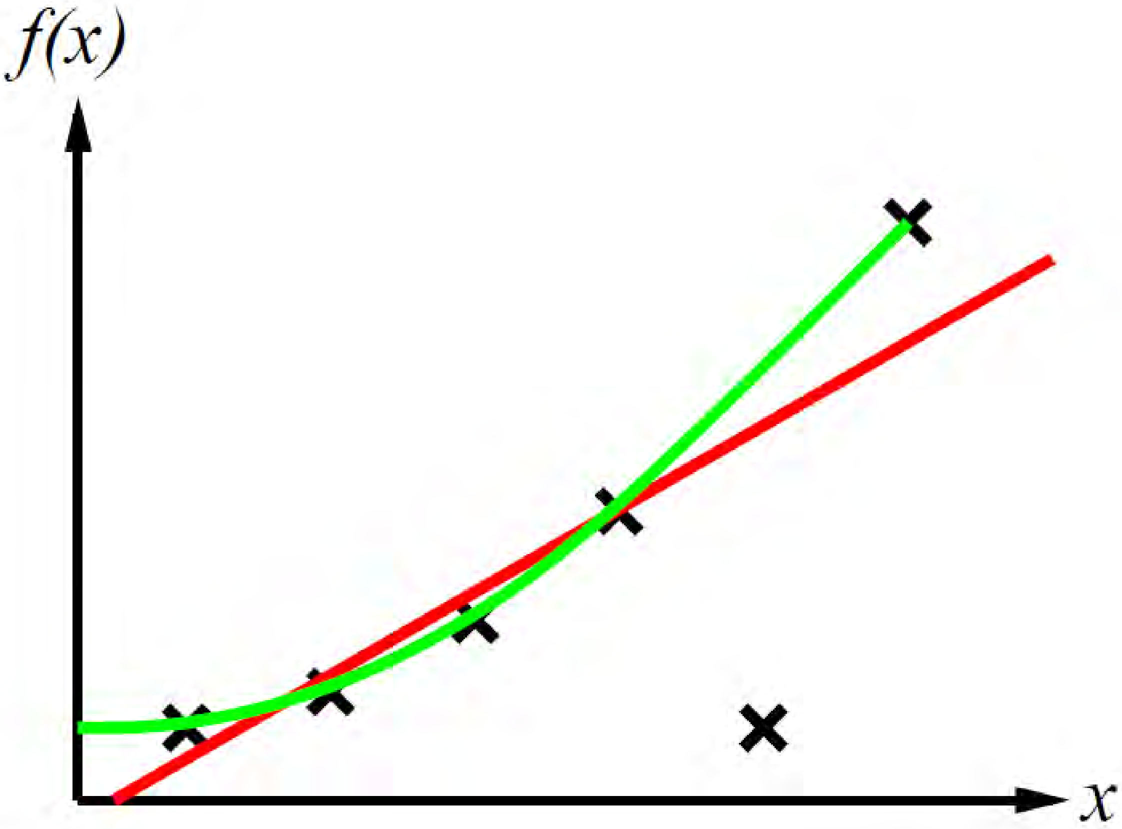
Example:



Inductive Learning

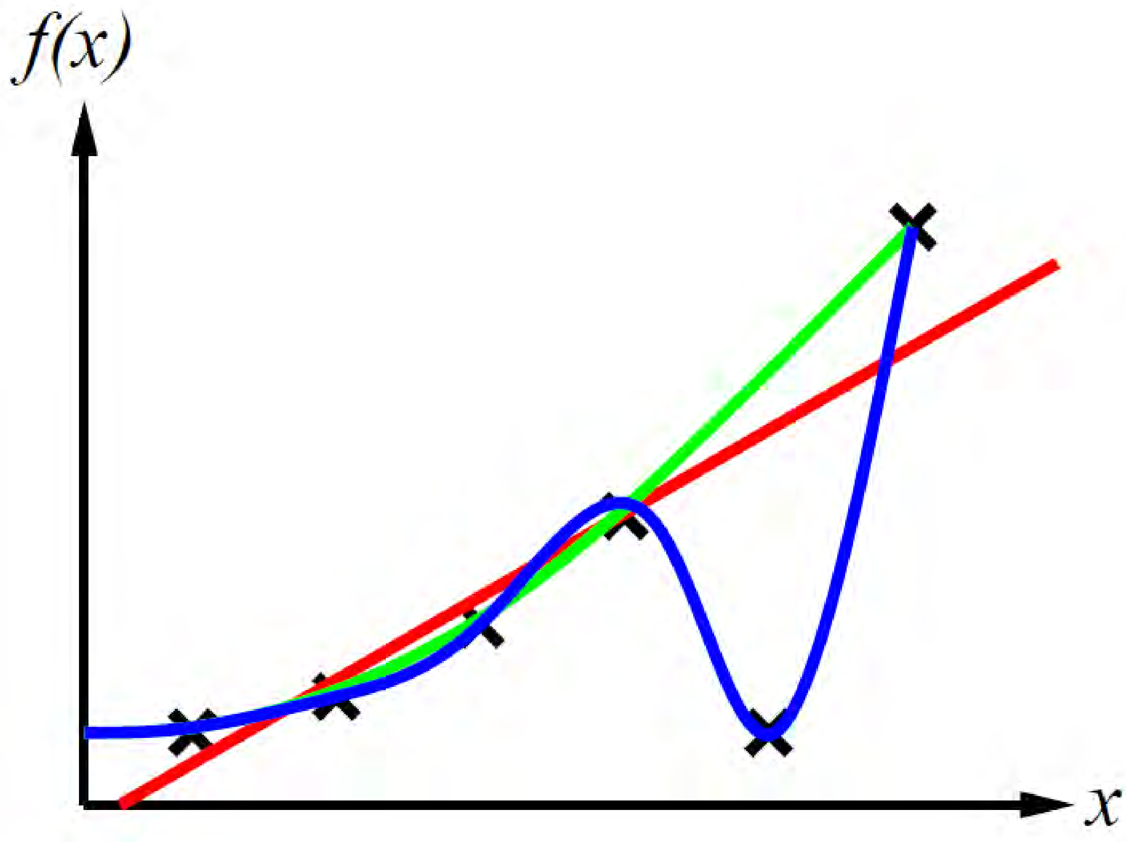
EFOP-3.4.3-16-2016-00009
A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

Example:



Inductive Learning

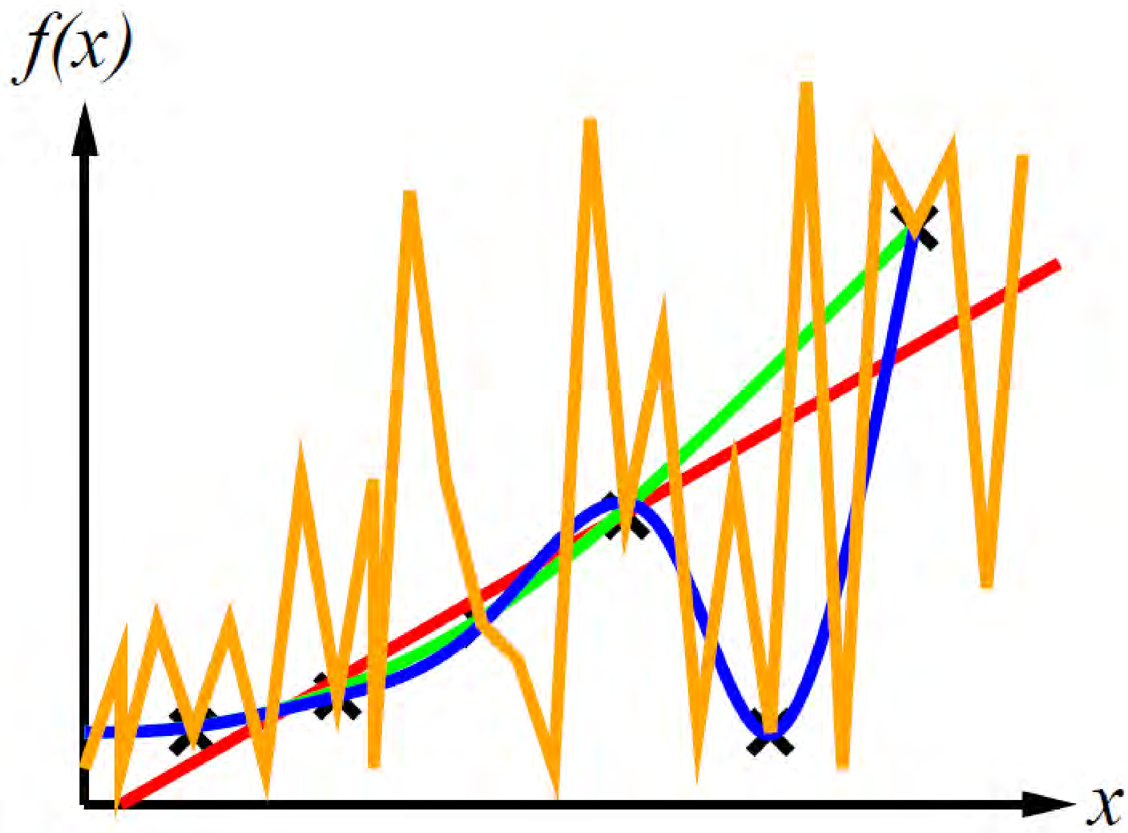
Example:



Inductive Learning

EFOP-3.4.3-16-2016-00009
A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

Example:

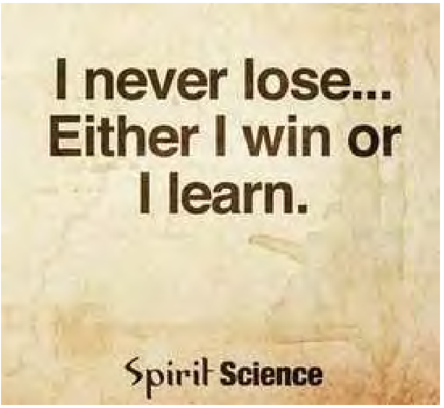


Inductive Learning

EFOP-3.4.3-16-2016-00009
A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

It was a highly **simplified model** of real learning, because it:

- **ignores prior knowledge**
- assumes that the **environment is deterministic**
- assumes that the **environment is observable**
- assumes that the **examples are given**
- assumes that the agent **intends to learn** the target function *f*



EFOP-3.4.3-16-2016-00009

A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

THANK YOU FOR THE ATTENTION!

Reference:

Stuart J. Russel – Peter Norvig:
Artificial Intelligence: A Modern Approach,
Prentice Hall, 2010, ISBN 0136042597

http://aima.cs.berkeley.edu/

EFOP-3.4.3-16-2016-00009

A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

ARTIFICIAL INTELLIGENCE

2-3-4-5. Decision Tree Learning

SZÉCHENYI 2020

MAGYARORSZÁG KORMÁNYA

Európai Unió
Európai Szociális Alap

BEFEKTETÉS A JÖVŐBE

Authors:

Tibor Dulai, Ágnes Werner-Stark

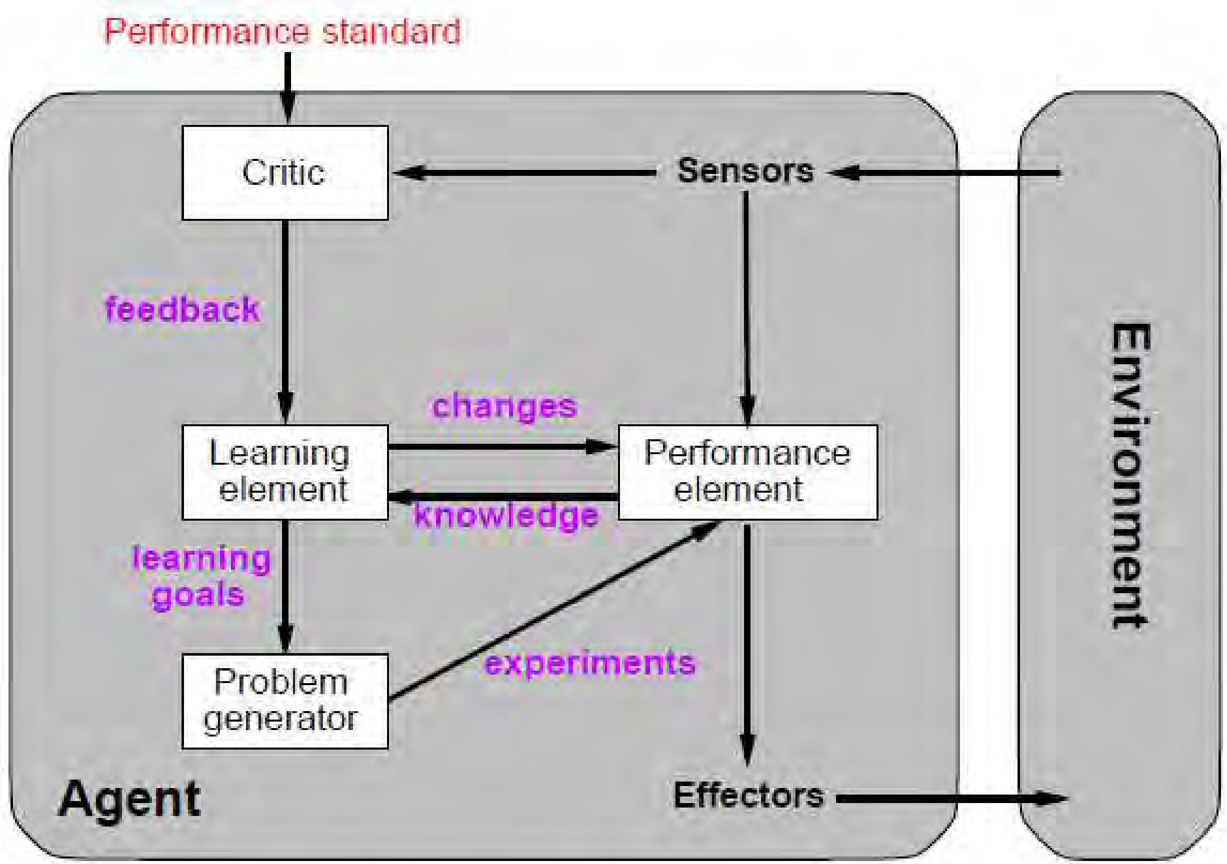
BASICS

Decision tree - Basics

EFOP-3.4.3-16-2016-00009

A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

A decision tree can be applied as a **representation of a performance element**.



It is an **attribute-based representation**. The attributes are the properties that are able to describe the examples.

Decision tree - Basics

For example: attributes that are used for deciding whether to wait for a table in a restaurant:

- ***Alt***: is there any alternative close place for eating? (true/false)
- ***Bar***: does the restaurant has any bar where one can wait for a table? (true/false)
- ***Fri***: is it weekend? (true/false)
- ***Hun***: am I hungry? (true/false)
- ***Pat***: how many people are in the restaurant? (none / some / full)
- ***Price***: how much expensive the restaurant is? ($ / $$ / $$$)
- ***Rain***: is it raining outside? (true/false)
- ***Res***: was a table reserved? (true/false)
- ***Type***: the type of the food in the restaurant (burger / french / italian / thai)
- ***Est***: estimated waiting time for a table (in minutes) (0-10 / 10-30 / 30-60 / >60)

Decision tree - Basics

EFOP-3.4.3-16-2016-00009
A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

Example: a pair of input vales (values of attributes) and the resulted output

Example	Attributes										Target
	Alt	*Bar*	*Fri*	*Hun*	*Pat*	*Price*	*Rain*	*Res*	*Type*	*Est*	*WillWait*
X_1	*T*	*F*	*F*	*T*	*Some*	*$$$*	*F*	*T*	*French*	*0–10*	*T*

The target is also called **goal predicate**. Its value is the **classification of the example**.

If the value of the goal predicate is a logical value, then
- if the classification of an example is „true”, then the example is a **positive example**.
- if the classification of an example is „false”, then the example is a **negative example**.

Decision tree - Basics

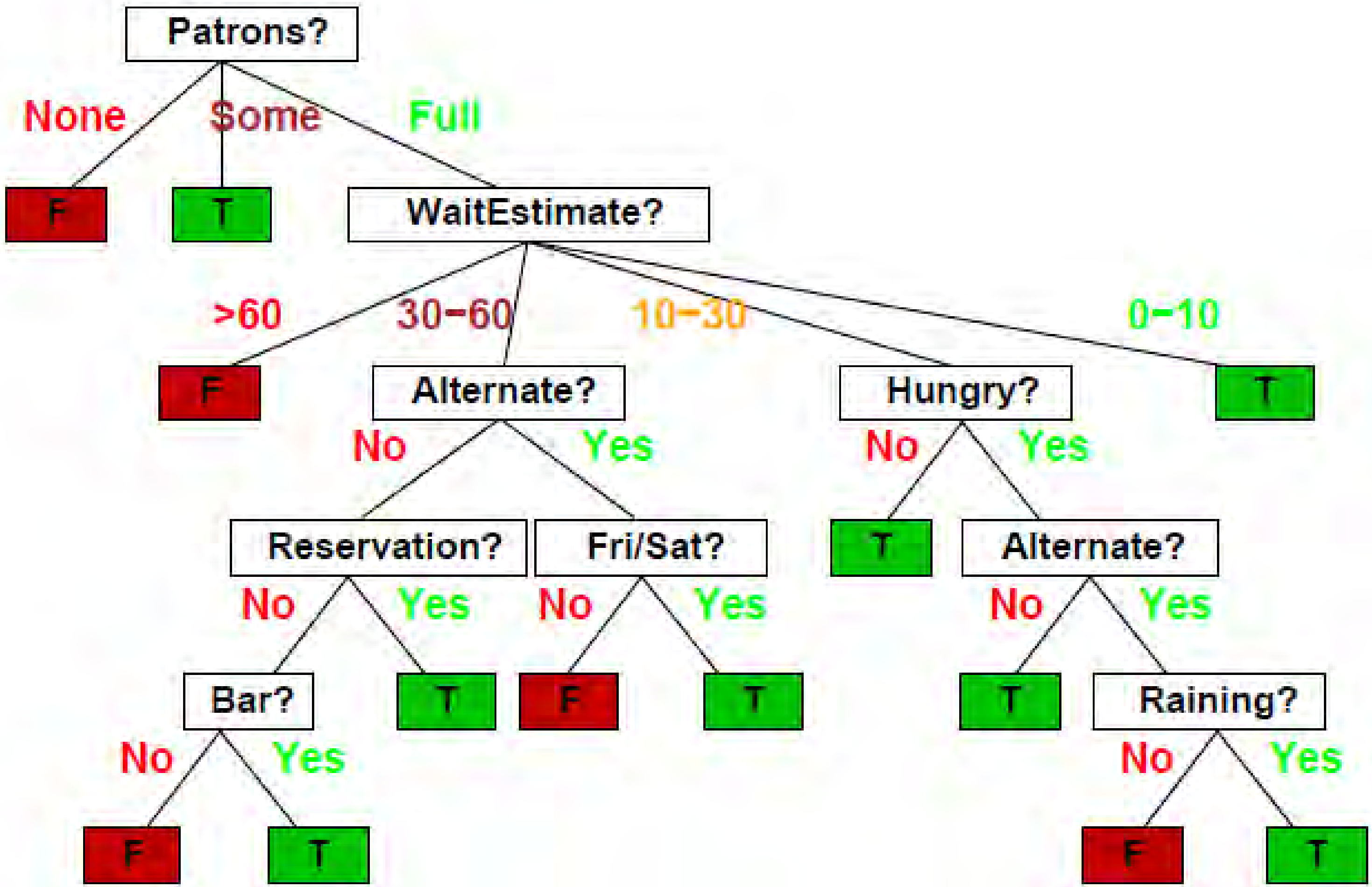
EFOP-3.4.3-16-2016-00009
A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

An example for a decision tree

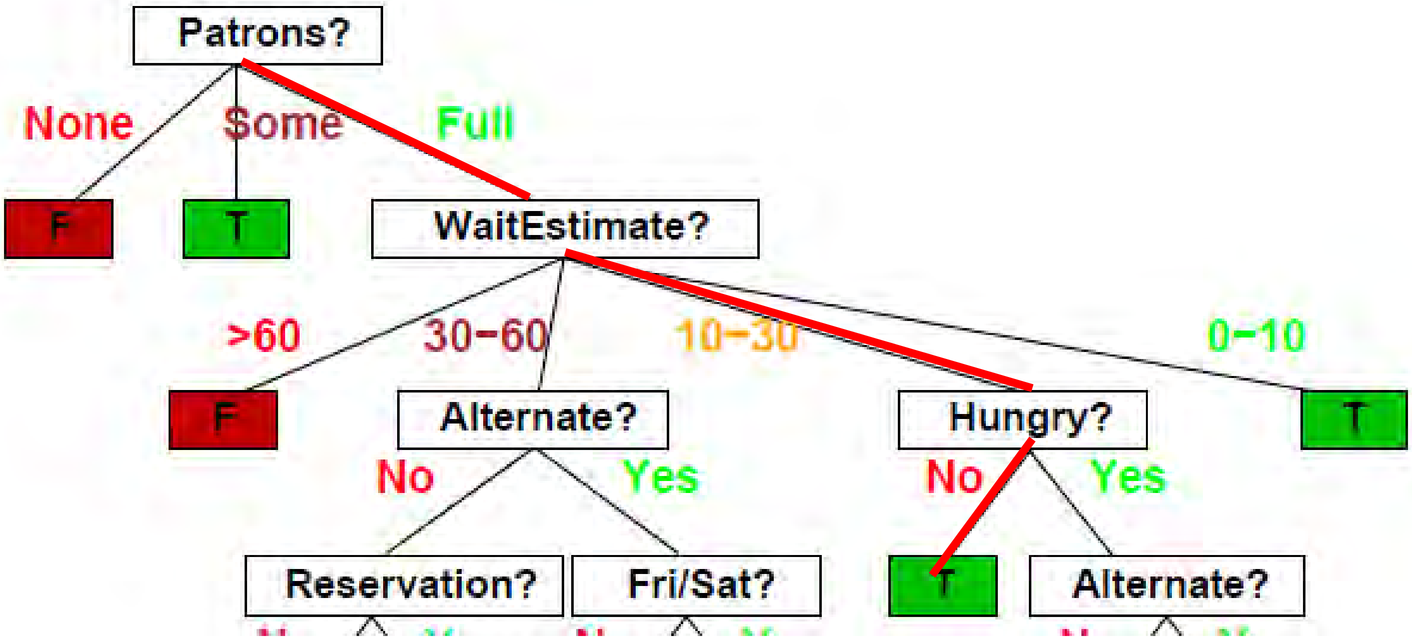
Each **inner node** represents a **test** of an attribute.

The **arcs** starting from an inner node are labelled by a **possible value of the test**.

Each **leaf** is a **decision**.



Decision tree - Basics



...

Logical expression of a path of the tree (of the path that is highlighted by red color):

$$\forall r\ Patrons(r, Full) \wedge WaitEstimate(r, 10-30) \wedge Hungry(r, No) \Rightarrow WillWait(r).$$

Logical expression of the whole decision tree:

Disjunction (OR logical relation) of **all** the logical expressions that belong to a **path that is terminated by T** (true) /they represent the positive examples/.

Decision tree - Basics

EFOP-3.4.3-16-2016-00009

A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

The set of the examples that are applied to train the system is called **training set**.

E.g.:

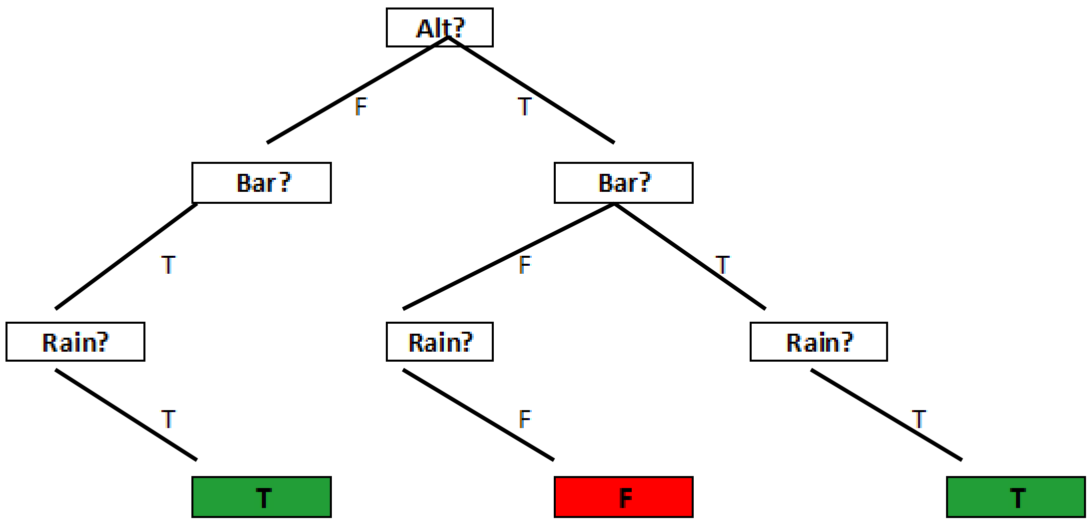
Example	Attributes										Target
	Alt	*Bar*	*Fri*	*Hun*	*Pat*	*Price*	*Rain*	*Res*	*Type*	*Est*	*WillWait*
X_1	T	F	F	T	Some	$$$	F	T	French	0–10	T
X_2	T	F	F	T	Full	$	F	F	Thai	30–60	F
X_3	F	T	F	F	Some	$	F	F	Burger	0–10	T
X_4	T	F	T	T	Full	$	F	F	Thai	10–30	T
X_5	T	F	T	F	Full	$$$	F	T	French	>60	F
X_6	F	T	F	T	Some	$$	T	T	Italian	0–10	T
X_7	F	T	F	F	None	$	T	F	Burger	0–10	F
X_8	F	F	F	T	Some	$$	T	T	Thai	0–10	T
X_9	F	T	T	F	Full	$	T	F	Burger	>60	F
X_{10}	T	T	T	T	Full	$$$	F	T	Italian	10–30	F
X_{11}	F	F	F	F	None	$	F	F	Thai	0–10	F
X_{12}	T	T	T	T	Full	$	F	F	Burger	30–60	T

Decision tree - Basics

A **trivial way** to build up a decision tree from the training set is to represent **each example on a separate path**.
It means that the tree is built up simple: it is the disjunction of all the logical representations that are expressed by the examples of the training set.

E.g.:

Alt	Bar	Rain	WillWait
F	T	T	T
T	F	F	F
T	T	T	T



This solution **represents the training set well** but with **poor generalization capability**. Later we will see a much better algorithm to build up a consistent hypothesis in the format of a decision tree with much better generalization capability.

EFOP-3.4.3-16-2016-00009
A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

EXPRESSIVENESS OF DECISION TREES

Decision tree Expressiveness

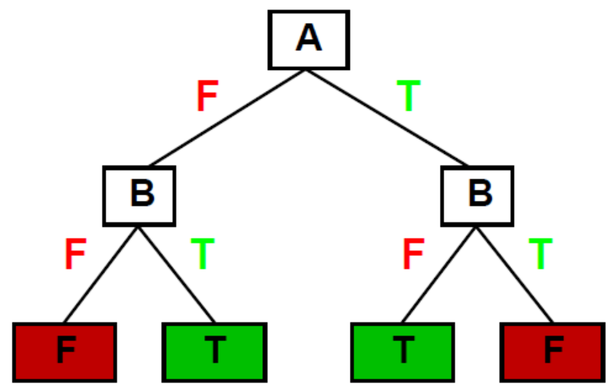
For every logical (Boolean) function a truth table can be written.
The truth table can be handled like a training set of a decision tree.

⬇

Every logical function can be expressed by a decision tree.

E.g.:

A	B	A xor B
F	F	F
F	T	T
T	F	T
T	T	F



However, decision trees are able to express **not only logical functions** (e.g., Patrons attribute can have None/Some/Full values in the previous example).

Decision tree Expressiveness

Are decision trees able to represent any set of logical statements?

The answer is: NO

E.g., consider the following test:

$$\exists\, r_2 \;\; Near(r_2, r) \wedge \text{Price}(r,p) \wedge \text{Price}(r_2, p_2) \wedge \text{Cheaper}(p_2, p)$$

For this test (is there any cheaper close restaurant?) decision tree can not be applied.

If the statements are related to more than one object, decision tree is not applicable.

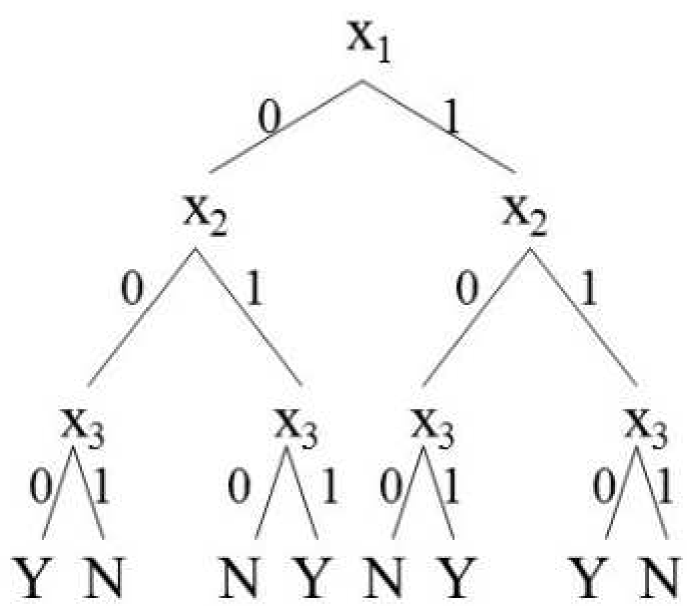
Decision trees are able to express any function of the input attributes if these attributes characterize only one object.

Decision tree Expressiveness

EFOP-3.4.3-16-2016-00009
A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

Is a decision tree an **efficient** function representation?

Decision tree for **parity function** (with three variables):



The **size of the decision tree is** an **exponential** function of the number of inputs.

For this problem decision tree representation is **not efficient**.

/It is also the problem in the case of majority function./

Decision tree Expressiveness

EFOP-3.4.3-16-2016-00009
A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

What is the **reason for existing problems that can not be represented efficiently**?

A truth table with *n* inputs has $\mathbf{2^n}$ **rows**.

A	B	C	Output
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	0
1	0	0	0
1	0	1	0
1	1	0	0
1	1	1	1

There can be *2* different outputs for one row.

Since there are 2^n rows, we can draw up

$$\underbrace{2 * 2 * \ldots * 2}_{2^n \text{ pieces}} = 2^{2^n}$$

different truth tables for *n* logical inputs.

The output column – as a 2^n length binary number – can express 2^n bit of information. Depending on the association rule that is applied in the truth table, sometimes it can be compressed but not in every case.

E.g., for 6 input the number of all the possible truth tables is 18446744073709551616.

Decision tree Expressiveness

EFOP-3.4.3-16-2016-00009

A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

As it was shown before, the inputs of a truth table can be handled as attributes of a decision tree. So, the **truth tables** of the previous example **can be considered as decision trees** (or logical functions).

Moreover, if the attributes of the decision tree are **not Boolean-valued**, then the values „2" in the equation $2 * 2 * \dots * 2 = 2^{2^n}$ can be higher values than two, that results in much more possible decision trees.

E.g., attribute „*Patrons*" of the Restaurant example can have values „*None*" or „*Some*" or „*Full*", even a logical statement **can miss this attribute**.

⬇

To find a consistent hypothesis in such a **huge search space**
a **clever algorithm is needed**.

EFOP-3.4.3-16-2016-00009
A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

HOW TO CREATE A DECISION TREE FROM EXAMPLES?

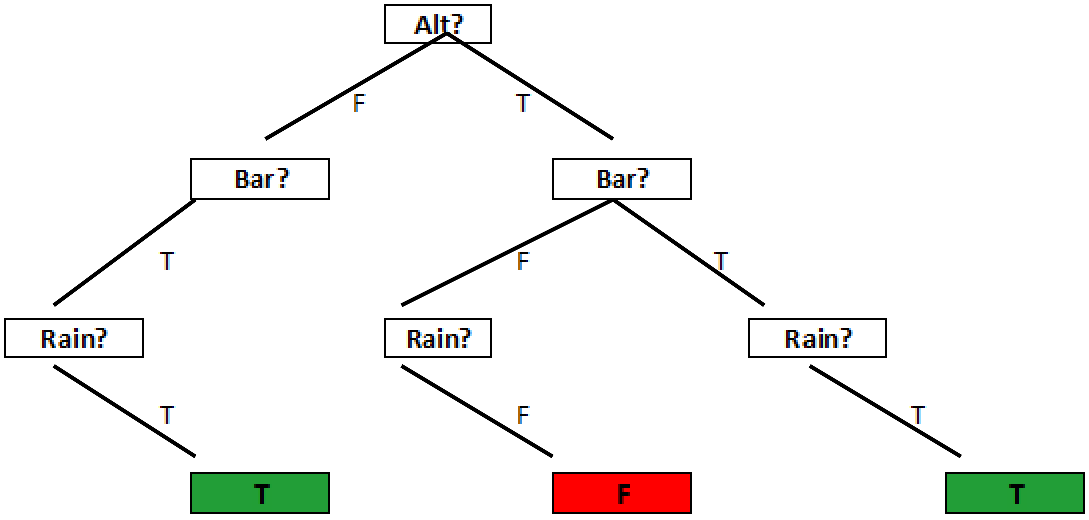
Decision tree - Creation

As we have seen previously:

A **trivial way** to build up a decision tree from the training set is to represent **each example on a separate path**.
It means that the tree is built up simple as the disjunction of all the logical representations that are expressed by the examples of the training set.

E.g.:

Alt	Bar	Rain	WillWait
F	T	T	T
T	F	F	F
T	T	T	T



This solution **represents the training set well** but with **poor generalization capability**.

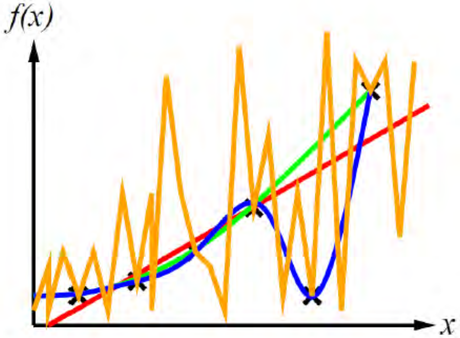
Decision tree - Creation

The main problem of the previously presented tree creation (the **poor generalization capability**) **causes**, that:

- **for** an example that is part of **the training set** it results in a **right classification**
- however, **it knows not much about examples that were not teached** to it.

This type of decision tree creation **memorizes the examples** of the training set (its experiences) and **does not mine characteristic samples** from them.
The lack of these samples **prevents to describe huge number of examples in a compressed form**.

These features validate the truth behind **Ockham's razor**. (select the simplest consistent hypothesis!)



Decision tree - Creation

EFOP-3.4.3-16-2016-00009
A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

To find the smallest consistent decision tree is very hard. However, it can be approached by a **simple heuristic**.

This heuristic works **recursive**ly: searches for the **most informative attribute** in each iteration (tries to find the **attribute that separates the best the examples** of different classifications).

⬇

This approach leads to **small number of** necessary **tests** to classify each example.

⬇

This way the created **tree will be small** (including short paths).

Decision tree - Creation

EFOP-3.4.3-16-2016-00009

A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

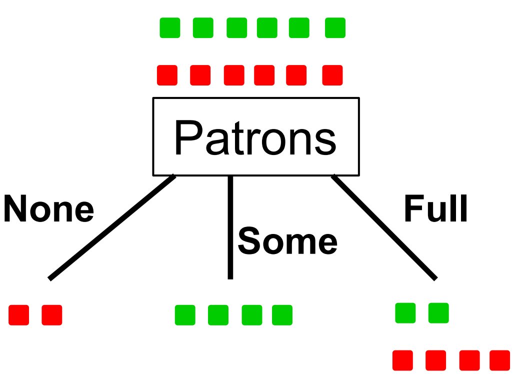
How **informative** an **attribute** is?

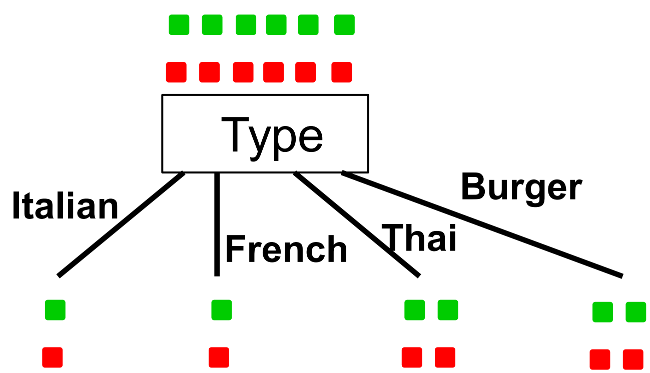
Before testing any attribute, the **initial distribution of classification** is:

- **True**: 6 examples
- **False**: 6 examples

Example	Attributes										Target
	Alt	*Bar*	*Fri*	*Hun*	*Pat*	*Price*	*Rain*	*Res*	*Type*	*Est*	*WillWait*
X_1	T	F	F	T	Some	$$$	F	T	French	0–10	T
X_2	T	F	F	T	Full	$	F	F	Thai	30–60	F
X_3	F	T	F	F	Some	$	F	F	Burger	0–10	T
X_4	T	F	T	T	Full	$	F	F	Thai	10–30	T
X_5	T	F	T	F	Full	$$$	F	T	French	>60	F
X_6	F	T	F	T	Some	$$	T	T	Italian	0–10	T
X_7	F	T	F	F	None	$	T	F	Burger	0–10	F
X_8	F	F	F	T	Some	$$	T	T	Thai	0–10	T
X_9	F	T	T	F	Full	$	T	F	Burger	>60	F
X_{10}	T	T	T	T	Full	$$$	F	T	Italian	10–30	F
X_{11}	F	F	F	F	None	$	F	F	Thai	0–10	F
X_{12}	T	T	T	T	Full	$	F	F	Burger	30–60	T

2 examples for attribute tests:





The attribute „Patrons" leads us much more to **clear final decisions**

Attribute „Patrons" is much more informative than attribute „Type".

Decision tree - Creation

How **to quantify the informativeness of** an **attribute** is?

The basics of the measurement can be: **how much the test of an attribute is able to decrease the uncertainity**.

Uncertainity can be expressed by **entropy**.

⬇

Information theory is able to help.

When we calculate the **entropy** of the system **before testing an attribute and after testing the attribute**, their **difference** expresses the informativeness of the attribute.

This property of the attribute is called: **information gain.**

Other measures also exist, see, e.g., **Gain Ration** and **Gini.**

EFOP-3.4.3-16-2016-00009
A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

The calculation of information gain

In general (1/2):

Before testing any attribute, there are $n = n_p + n_n$ examples, where n_p is the number of positive examples and n_n is the number of negative examples.

So, the **initial entropy** (the amount of information needed for a clear decision) is:

$$H_{init} = \frac{n_p}{n} * log_2 \frac{n}{n_p} + \frac{n_n}{n} * log_2 \frac{n}{n_n} \text{ [bit]}$$

Of course, the applied $p * log_2 \left(\frac{1}{p}\right)$ formula is equivalent with the format of

$-p * log_2(p)$.

EFOP-3.4.3-16-2016-00009
A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

The calculation of information gain

In general (2/2):

After testing an attribute, there are m possible outcomes. Each v_k possible outcome will be represented by $n^k = n^k_p + n^k_n$ examples, where n^k_p is the number of positive examples and n^k_n is the number of negative examples in the k^{th} branch.

Then, the **entropy after testing this attribute** (the amount of information needed for a clear decision after testing the attribute) is:

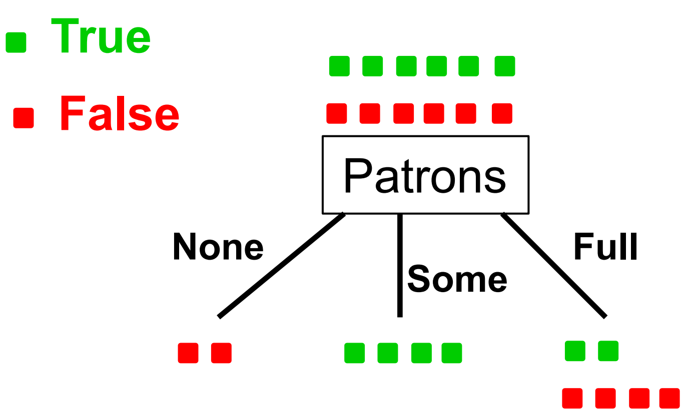
$$H_{afterTesting} = \sum_{k=1}^{m} \frac{n^k}{n} \left(\frac{n_p^k}{n^k} * log_2 \frac{n^k}{n_p^k} + \frac{n_n^k}{n^k} * log_2 \frac{n^k}{n_n^k} \right) \text{ [bit]}$$

Then, **the information gain of the tested attribute** is:

$$information\ gain(tested\ attribute) = H_{init} - H_{afterTesting}$$

Decision tree - Creation

An **example** for **information gain (1/7)**:



The **entropy before testing** any attribute is:

/since the number of positive examples =
= the number of negative examples = 6/

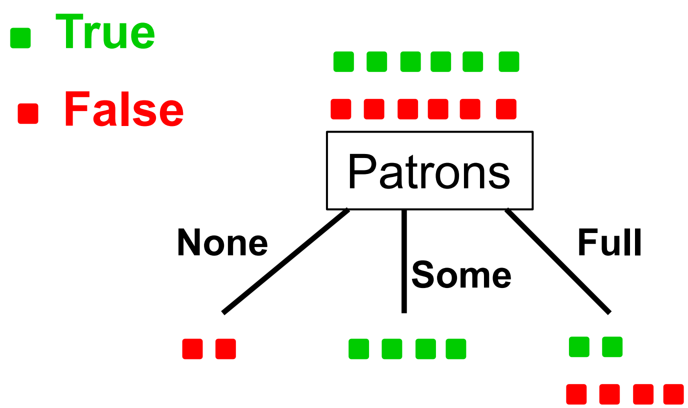
$$\frac{6}{12} * log_2\frac{12}{6} + \frac{6}{12} * log_2\frac{12}{6} =$$

$$\frac{1}{2} * log_2 2 + \frac{1}{2} * log_2 2 = \frac{1}{2} * 1 + \frac{1}{2} * 1 = 1 \text{ [bit]}$$

It means, that before testing any attribute (e.g., attribute „Patrons”), **we need 1 bit information for making a 100% sure decision** (a clear classification of an example).

EFOP-3.4.3-16-2016-00009
A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

An **example** for **information gain (2/7)**:



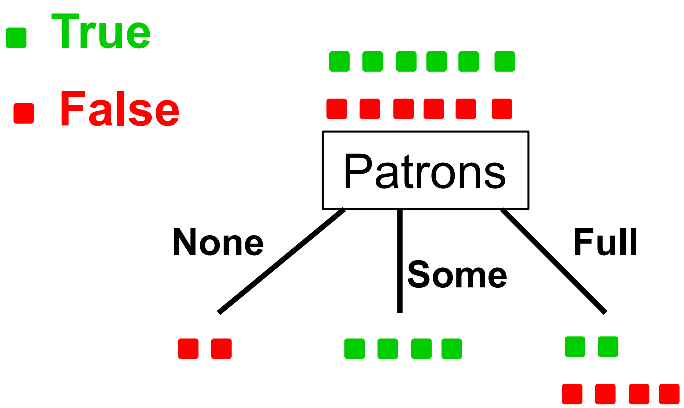
The **entropy after testing attribute „Patrons"** is:
/the method/

The testing of „Patrons" attribute can result in **three different outcomes** (None, Some or Full).

In each branch we may need different amount of decision for making a clear decision. First, **we calculate the entropy for all the three branches, after that we combine them** (weighted by the probability of the branches).

Decision tree - Creation

An **example** for **information gain (3/7)**:



The **entropy after testing attribute „Patrons”** is:

Step 1.

We calculate the entropy for the **branch** where attribute „Patrons” has value „**None**”:

$$\frac{0}{2} * (information\ given\ by\ positive\ examples) + \frac{2}{2} * log_2 \frac{2}{2} =$$

$$0 + 1 * log_2 1 = 0 + 0 = 0 \text{ [bit]}$$

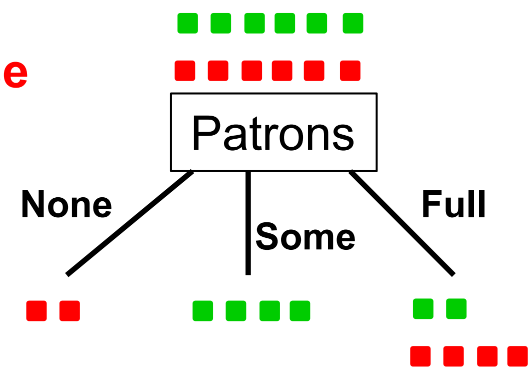
Decision tree - Creation

EFOP-3.4.3-16-2016-00009

A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

An **example** for **information gain (4/7)**:

- **True**
- **False**



The **entropy after testing attribute „Patrons”** is:

Step 2.

We calculate the entropy for the **branch** where attribute „Patrons” has value „**Some**”:

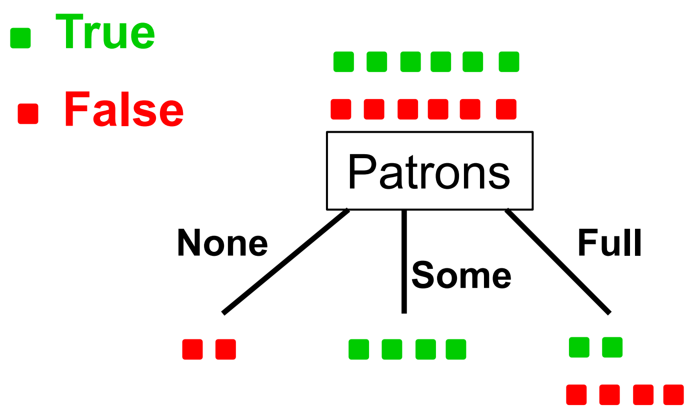
$$\frac{0}{4} * (information\ given\ by\ negative\ examples) + \frac{4}{4} * log_2 \frac{4}{4} =$$

$$0 + 1 * log_2 1 = 0 + 0 = 0 \text{ [bit]}$$

Decision tree - Creation

EFOP-3.4.3-16-2016-00009
A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

An **example** for **information gain (5/7)**:



The **entropy after testing attribute „Patrons"** is:

Step 3.

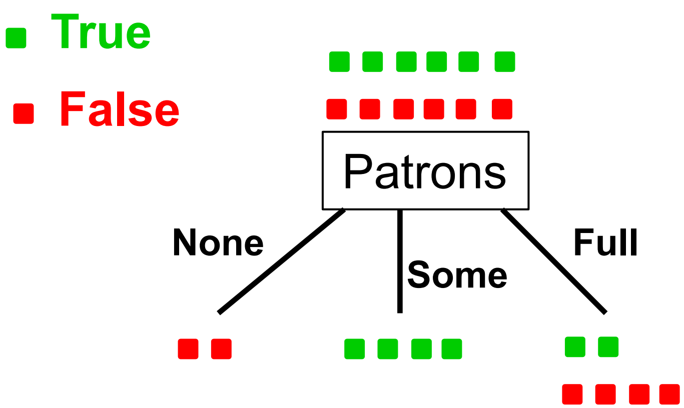
We calculate the entropy for the **branch** where attribute „Patrons" has value „**Full**":

$$\frac{2}{6} * log_2 \frac{6}{2} + \frac{4}{6} * log_2 \frac{6}{4} = \frac{1}{3} * log_2 3 + \frac{2}{3} * log_2 \frac{3}{2} = \frac{1}{3} * \frac{log_{10} 3}{log_{10} 2} + \frac{2}{3} * \frac{log_{10} \frac{3}{2}}{log_{10} 2} \approx$$

$$\approx \frac{1}{3} * 1.58496 + \frac{2}{3} * 0.58496 = 0.918293 \text{ [bit]}$$

Decision tree - Creation

An **example** for **information gain (6/7)**:



The **entropy after testing attribute „Patrons"** is:

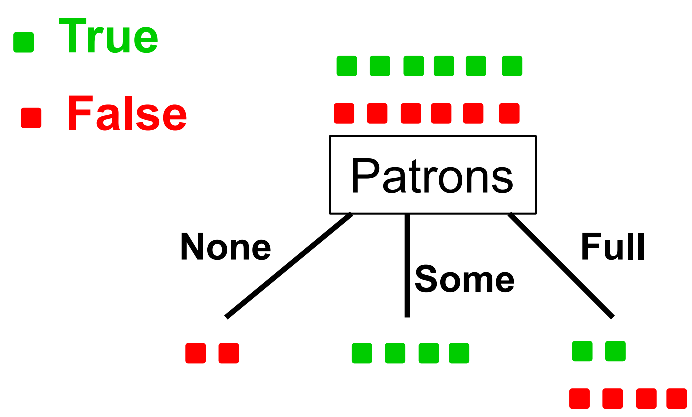
Step 4.

We **combine the probability-weighted entropies** of the branches:

$$\frac{2}{12} * 0\ bit + \frac{4}{12} * 0\ bit + \frac{6}{12} * 0.918293\ bit = 0.4591465 \text{ [bit]}$$

So, after testing attribute „Patrons", **we need more 0.4591465 bit information for making a 100% sure decision** (a clear classification of an example).

An **example** for **information gain (7/7)**:



The **entropy before testing attribute „Patrons”** was: **1** [bit]

The **entropy after testing attribute „Patrons”** is: **0.4591465** [bit]

So, the **information gain of attribute „Patrons”** is:

1 [bit] – 0.4591465 [bit] = **0.5408535** [bit]

Decision tree - Creation

Now, /after determinig the Choose-Attribute() function/ we are ready to **build a decision tree – with good generalization capability – from the training set**.

The Decision Tree Learning algorithm*

```
function DTL(examples, attributes, default) returns a decision tree
    if examples is empty then return default
    else if all examples have the same classification then return the classification
    else if attributes is empty then return MODE(examples)
    else
        best ← CHOOSE-ATTRIBUTE(attributes, examples)
        tree ← a new decision tree with root test best
        for each value v_i of best do
            examples_i ← {elements of examples with best = v_i}
            subtree ← DTL(examples_i, attributes − best, MODE(examples))
            add a branch to tree with label v_i and subtree subtree
        return tree
```

* Stuart J. Russel – Peter Norvig: Artificial Intelligence: A Modern Approach, Prentice Hall, 2010

Decision tree - Creation

EFOP-3.4.3-16-2016-00009
A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

Applying the presented Decision Tree Learning Algorithm, **let us create the first two levels of the decision tree** of the „Restaurant example”!

The root of the decision tree will be the most informative attribute.

Recall, that the **initial entropy was**

$$\frac{6}{12} * log_2\frac{12}{6} + \frac{6}{12} * log_2\frac{12}{6} = \frac{1}{2} * log_2 2 + \frac{1}{2} * log_2 2 = \frac{1}{2} * 1 + \frac{1}{2} * 1 = 1 \text{ [bit]}$$

Starting with that, **the information gain of attribute „Patrons”** was:

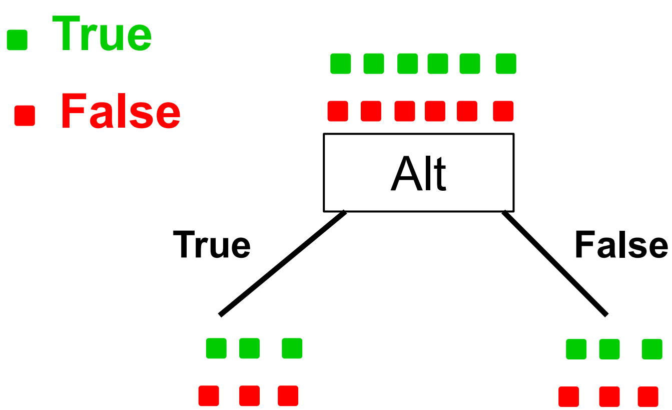
0.5408535 [bit]

Let’s **continue** with testing the other attributes!

Example	Attributes										Target
	Alt	*Bar*	*Fri*	*Hun*	*Pat*	*Price*	*Rain*	*Res*	*Type*	*Est*	*WillWait*
X_1	T	F	F	T	Some	$$$	F	T	French	0–10	T
X_2	T	F	F	T	Full	$	F	F	Thai	30–60	F
X_3	F	T	F	F	Some	$	F	F	Burger	0–10	T
X_4	T	F	T	T	Full	$	F	F	Thai	10–30	T
X_5	T	F	T	F	Full	$$$	F	T	French	>60	F
X_6	F	T	F	T	Some	$$	T	T	Italian	0–10	T
X_7	F	T	F	F	None	$	T	F	Burger	0–10	F
X_8	F	F	F	T	Some	$$	T	T	Thai	0–10	T
X_9	F	T	T	F	Full	$	T	F	Burger	>60	F
X_{10}	T	T	T	T	Full	$$$	F	T	Italian	10–30	F
X_{11}	F	F	F	F	None	$	F	F	Thai	0–10	F
X_{12}	T	T	T	T	Full	$	F	F	Burger	30–60	T

Decision tree - Creation

Information gain of attribute „Alt”:



Example	Attributes										Target
	Alt	*Bar*	*Fri*	*Hun*	*Pat*	*Price*	*Rain*	*Res*	*Type*	*Est*	*WillWait*
X_1	T	F	F	T	Some	$$$	F	T	French	0–10	T
X_2	T	F	F	T	Full	$	F	F	Thai	30–60	F
X_3	F	T	F	F	Some	$	F	F	Burger	0–10	T
X_4	T	F	T	T	Full	$	F	F	Thai	10–30	T
X_5	T	F	T	F	Full	$$$	F	T	French	>60	F
X_6	F	T	F	T	Some	$$	T	T	Italian	0–10	T
X_7	F	T	F	F	None	$	T	F	Burger	0–10	F
X_8	F	F	F	T	Some	$$	T	T	Thai	0–10	T
X_9	F	T	T	F	Full	$	T	F	Burger	>60	F
X_{10}	T	T	T	T	Full	$$$	F	T	Italian	10–30	F
X_{11}	F	F	F	F	None	$	F	F	Thai	0–10	F
X_{12}	T	T	T	T	Full	$	F	F	Burger	30–60	T

The **entropy after testing the attribute** is:

$$\frac{6}{12}*\left(\frac{3}{6}*log_2\frac{6}{3}+\frac{3}{6}*log_2\frac{6}{3}\right)+\frac{6}{12}*\left(\frac{3}{6}*log_2\frac{6}{3}+\frac{3}{6}*log_2\frac{6}{3}\right)=$$

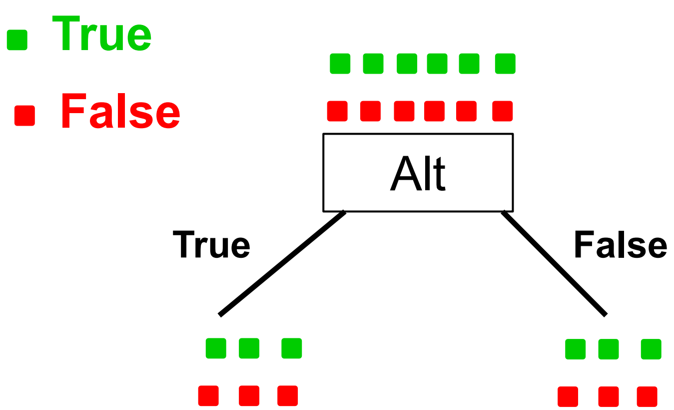
$$=\frac{1}{2}*\left(\frac{1}{2}*log_2 2+\frac{1}{2}*log_2 2\right)+\frac{1}{2}*\left(\frac{1}{2}*log_2 2+\frac{1}{2}*log_2 2\right)=$$

Decision tree - Creation

EFOP-3.4.3-16-2016-00009

A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

Information gain of attribute „Alt”:



Example	Attributes										Target
	Alt	*Bar*	*Fri*	*Hun*	*Pat*	*Price*	*Rain*	*Res*	*Type*	*Est*	*WillWait*
X_1	*T*	*F*	*F*	*T*	*Some*	*$$$*	*F*	*T*	*French*	*0–10*	*T*
X_2	*T*	*F*	*F*	*T*	*Full*	*$*	*F*	*F*	*Thai*	*30–60*	*F*
X_3	*F*	*T*	*F*	*F*	*Some*	*$*	*F*	*F*	*Burger*	*0–10*	*T*
X_4	*T*	*F*	*T*	*T*	*Full*	*$*	*F*	*F*	*Thai*	*10–30*	*T*
X_5	*T*	*F*	*T*	*F*	*Full*	*$$$*	*F*	*T*	*French*	*>60*	*F*
X_6	*F*	*T*	*F*	*T*	*Some*	*$$*	*T*	*T*	*Italian*	*0–10*	*T*
X_7	*F*	*T*	*F*	*F*	*None*	*$*	*T*	*F*	*Burger*	*0–10*	*F*
X_8	*F*	*F*	*F*	*T*	*Some*	*$$*	*T*	*T*	*Thai*	*0–10*	*T*
X_9	*F*	*T*	*T*	*F*	*Full*	*$*	*T*	*F*	*Burger*	*>60*	*F*
X_{10}	*T*	*T*	*T*	*T*	*Full*	*$$$*	*F*	*T*	*Italian*	*10–30*	*F*
X_{11}	*F*	*F*	*F*	*F*	*None*	*$*	*F*	*F*	*Thai*	*0–10*	*F*
X_{12}	*T*	*T*	*T*	*T*	*Full*	*$*	*F*	*F*	*Burger*	*30–60*	*T*

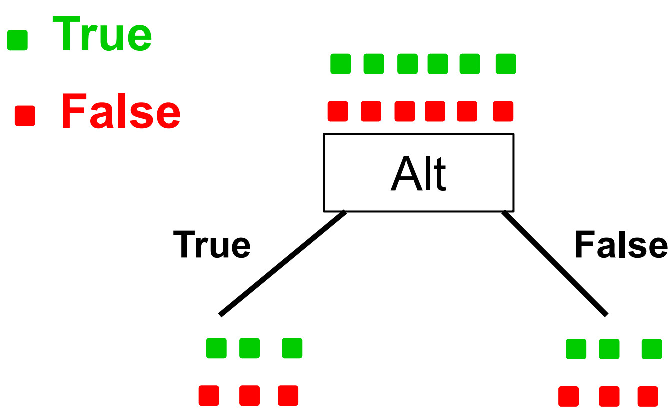
The **entropy after testing the attribute** is:

$$= \frac{1}{2} * \left(\frac{1}{2} * log_2 2 + \frac{1}{2} * log_2 2\right) + \frac{1}{2} * \left(\frac{1}{2} * log_2 2 + \frac{1}{2} * log_2 2\right) =$$

$$\frac{1}{2} * \left(\frac{1}{2} * 1 + \frac{1}{2} * 1\right) + \frac{1}{2} * \left(\frac{1}{2} * 1 + \frac{1}{2} * 1\right) = \frac{1}{2} + \frac{1}{2} = 1 \text{ [bit]}$$

Decision tree - Creation

Information gain of attribute „Alt”:



Example	Attributes										Target
	Alt	*Bar*	*Fri*	*Hun*	*Pat*	*Price*	*Rain*	*Res*	*Type*	*Est*	*WillWait*
X_1	T	F	F	T	Some	$$$	F	T	French	0–10	T
X_2	T	F	F	T	Full	$	F	F	Thai	30–60	F
X_3	F	T	F	F	Some	$	F	F	Burger	0–10	T
X_4	T	F	T	T	Full	$	F	F	Thai	10–30	T
X_5	T	F	T	F	Full	$$$	F	T	French	>60	F
X_6	F	T	F	T	Some	$$	T	T	Italian	0–10	T
X_7	F	T	F	F	None	$	T	F	Burger	0–10	F
X_8	F	F	F	T	Some	$$	T	T	Thai	0–10	T
X_9	F	T	T	F	Full	$	T	F	Burger	>60	F
X_{10}	T	T	T	T	Full	$$$	F	T	Italian	10–30	F
X_{11}	F	F	F	F	None	$	F	F	Thai	0–10	F
X_{12}	T	T	T	T	Full	$	F	F	Burger	30–60	T

The **entropy after testing the attribute** is:

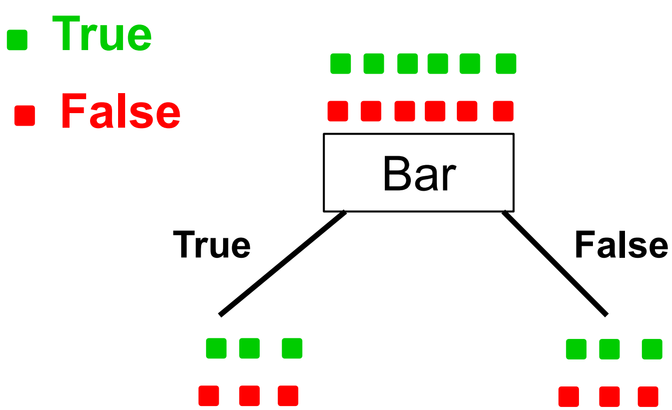
$$\frac{1}{2}*\left(\frac{1}{2}*1+\frac{1}{2}*1\right)+\frac{1}{2}*\left(\frac{1}{2}*1+\frac{1}{2}*1\right)=\frac{1}{2}+\frac{1}{2}=1 \text{ [bit]}$$

So, the **information gain of attribute „Alt” is**: 1 [bit] – 1 [bit] = **0 [bit]**

Decision tree - Creation

EFOP-3.4.3-16-2016-00009
A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

Information gain of attribute „Bar”:



Example	Attributes										Target
	Alt	*Bar*	*Fri*	*Hun*	*Pat*	*Price*	*Rain*	*Res*	*Type*	*Est*	*WillWait*
X_1	*T*	*F*	*F*	*T*	*Some*	*$$$*	*F*	*T*	*French*	*0–10*	*T*
X_2	*T*	*F*	*F*	*T*	*Full*	*$*	*F*	*F*	*Thai*	*30–60*	*F*
X_3	*F*	*T*	*F*	*F*	*Some*	*$*	*F*	*F*	*Burger*	*0–10*	*T*
X_4	*T*	*F*	*T*	*T*	*Full*	*$*	*F*	*F*	*Thai*	*10–30*	*T*
X_5	*T*	*F*	*T*	*F*	*Full*	*$$$*	*F*	*T*	*French*	*>60*	*F*
X_6	*F*	*T*	*F*	*T*	*Some*	*$$*	*T*	*T*	*Italian*	*0–10*	*T*
X_7	*F*	*T*	*F*	*F*	*None*	*$*	*T*	*F*	*Burger*	*0–10*	*F*
X_8	*F*	*F*	*F*	*T*	*Some*	*$$*	*T*	*T*	*Thai*	*0–10*	*T*
X_9	*F*	*T*	*T*	*F*	*Full*	*$*	*T*	*F*	*Burger*	*>60*	*F*
X_{10}	*T*	*T*	*T*	*T*	*Full*	*$$$*	*F*	*T*	*Italian*	*10–30*	*F*
X_{11}	*F*	*F*	*F*	*F*	*None*	*$*	*F*	*F*	*Thai*	*0–10*	*F*
X_{12}	*T*	*T*	*T*	*T*	*Full*	*$*	*F*	*F*	*Burger*	*30–60*	*T*

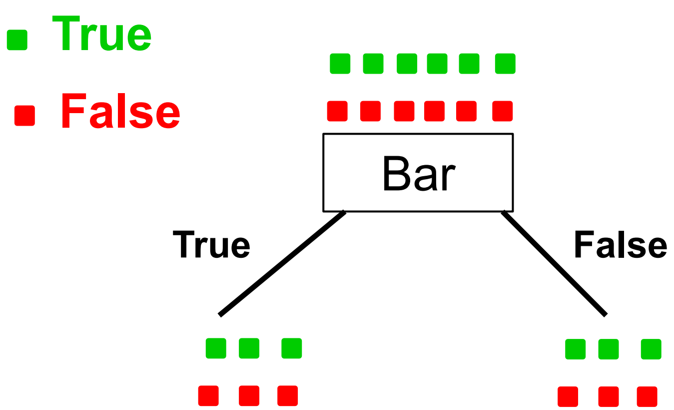
The **entropy after testing the attribute** is:

$$\frac{6}{12} * \left(\frac{3}{6} * log_2 \frac{6}{3} + \frac{3}{6} * log_2 \frac{6}{3}\right) + \frac{6}{12} * \left(\frac{3}{6} * log_2 \frac{6}{3} + \frac{3}{6} * log_2 \frac{6}{3}\right) =$$

$$= \frac{1}{2} * \left(\frac{1}{2} * log_2 2 + \frac{1}{2} * log_2 2\right) + \frac{1}{2} * \left(\frac{1}{2} * log_2 2 + \frac{1}{2} * log_2 2\right) =$$

Decision tree - Creation

Information gain of attribute „Bar”:



Example	Attributes										Target
	Alt	*Bar*	*Fri*	*Hun*	*Pat*	*Price*	*Rain*	*Res*	*Type*	*Est*	*WillWait*
X_1	T	F	F	T	Some	$$$	F	T	French	0–10	T
X_2	T	F	F	T	Full	$	F	F	Thai	30–60	F
X_3	F	T	F	F	Some	$	F	F	Burger	0–10	T
X_4	T	F	T	T	Full	$	F	F	Thai	10–30	T
X_5	T	F	T	F	Full	$$$	F	T	French	>60	F
X_6	F	T	F	T	Some	$$	T	T	Italian	0–10	T
X_7	F	T	F	F	None	$	T	F	Burger	0–10	F
X_8	F	F	F	T	Some	$$	T	T	Thai	0–10	T
X_9	F	T	T	F	Full	$	T	F	Burger	>60	F
X_{10}	T	T	T	T	Full	$$$	F	T	Italian	10–30	F
X_{11}	F	F	F	F	None	$	F	F	Thai	0–10	F
X_{12}	T	T	T	T	Full	$	F	F	Burger	30–60	T

The **entropy after testing the attribute** is:

$$= \frac{1}{2} * \left(\frac{1}{2} * log_2 2 + \frac{1}{2} * log_2 2\right) + \frac{1}{2} * \left(\frac{1}{2} * log_2 2 + \frac{1}{2} * log_2 2\right) =$$

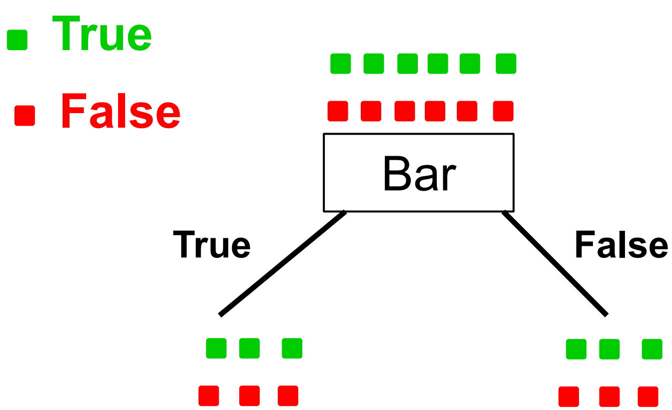
$$\frac{1}{2} * \left(\frac{1}{2} * 1 + \frac{1}{2} * 1\right) + \frac{1}{2} * \left(\frac{1}{2} * 1 + \frac{1}{2} * 1\right) = \frac{1}{2} + \frac{1}{2} = 1 \text{ [bit]}$$

Decision tree - Creation

EFOP-3.4.3-16-2016-00009

A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

Information gain of attribute „Bar”:



Example	Alt	Bar	Fri	Hun	Pat	Price	Rain	Res	Type	Est	WillWait
	Attributes										Target
X_1	T	F	F	T	Some	$$$	F	T	French	0–10	T
X_2	T	F	F	T	Full	$	F	F	Thai	30–60	F
X_3	F	T	F	F	Some	$	F	F	Burger	0–10	T
X_4	T	F	T	T	Full	$	F	F	Thai	10–30	T
X_5	T	F	T	F	Full	$$$	F	T	French	>60	F
X_6	F	T	F	T	Some	$$	T	T	Italian	0–10	T
X_7	F	T	F	F	None	$	T	F	Burger	0–10	F
X_8	F	F	F	T	Some	$$	T	T	Thai	0–10	T
X_9	F	T	T	F	Full	$	T	F	Burger	>60	F
X_{10}	T	T	T	T	Full	$$$	F	T	Italian	10–30	F
X_{11}	F	F	F	F	None	$	F	F	Thai	0–10	F
X_{12}	T	T	T	T	Full	$	F	F	Burger	30–60	T

The **entropy after testing the attribute** is:

$$\frac{1}{2} * \left(\frac{1}{2} * 1 + \frac{1}{2} * 1\right) + \frac{1}{2} * \left(\frac{1}{2} * 1 + \frac{1}{2} * 1\right) = \frac{1}{2} + \frac{1}{2} = 1 \text{ [bit]}$$

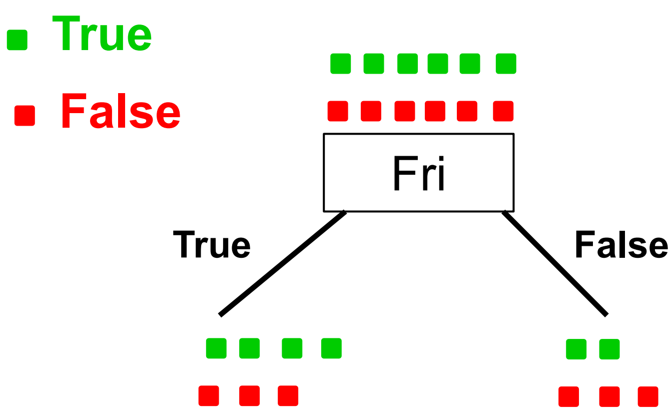
So, the **information gain of attribute „Bar” is**: 1 [bit] – 1 [bit] = **0 [bit]**

Decision tree - Creation

EFOP-3.4.3-16-2016-00009

A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

Information gain of attribute „Fri”:



Example	Attributes										Target
	Alt	*Bar*	*Fri*	*Hun*	*Pat*	*Price*	*Rain*	*Res*	*Type*	*Est*	*WillWait*
X_1	*T*	*F*	*F*	*T*	*Some*	*\$\$\$*	*F*	*T*	*French*	*0–10*	*T*
X_2	*T*	*F*	*F*	*T*	*Full*	*\$*	*F*	*F*	*Thai*	*30–60*	*F*
X_3	*F*	*T*	*F*	*F*	*Some*	*\$*	*F*	*F*	*Burger*	*0–10*	*T*
X_4	*T*	*F*	*T*	*T*	*Full*	*\$*	*F*	*F*	*Thai*	*10–30*	*T*
X_5	*T*	*F*	*T*	*F*	*Full*	*\$\$\$*	*F*	*T*	*French*	*>60*	*F*
X_6	*F*	*T*	*F*	*T*	*Some*	*\$\$*	*T*	*T*	*Italian*	*0–10*	*T*
X_7	*F*	*T*	*F*	*F*	*None*	*\$*	*T*	*F*	*Burger*	*0–10*	*F*
X_8	*F*	*F*	*F*	*T*	*Some*	*\$\$*	*T*	*T*	*Thai*	*0–10*	*T*
X_9	*F*	*T*	*T*	*F*	*Full*	*\$*	*T*	*F*	*Burger*	*>60*	*F*
X_{10}	*T*	*T*	*T*	*T*	*Full*	*\$\$\$*	*F*	*T*	*Italian*	*10–30*	*F*
X_{11}	*F*	*F*	*F*	*F*	*None*	*\$*	*F*	*F*	*Thai*	*0–10*	*F*
X_{12}	*T*	*T*	*T*	*T*	*Full*	*\$*	*F*	*F*	*Burger*	*30–60*	*T*

The **entropy after testing the attribute** is:

$$\frac{7}{12} * \left(\frac{4}{7} * log_2 \frac{7}{4} + \frac{3}{7} * log_2 \frac{7}{3}\right) + \frac{5}{12} * \left(\frac{2}{5} * log_2 \frac{5}{2} + \frac{3}{5} * log_2 \frac{5}{3}\right) =$$

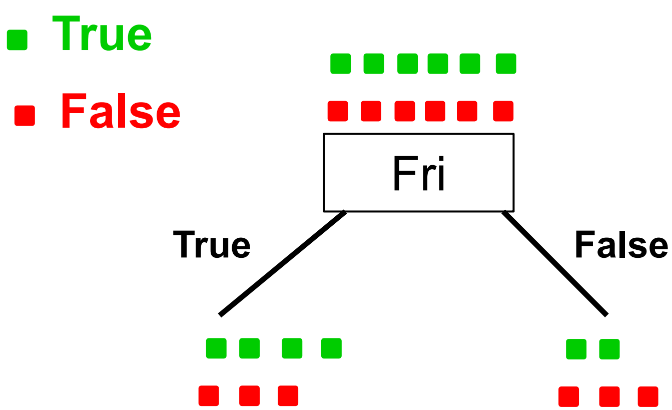
$$= \frac{7}{12} * \left(\frac{4}{7} * \frac{log_{10} \frac{7}{4}}{log_{10} 2} + \frac{3}{7} * \frac{log_{10} \frac{7}{3}}{log_{10} 2}\right) + \frac{5}{12} * \left(\frac{2}{5} * \frac{log_{10} \frac{5}{2}}{log_{10} 2} + \frac{3}{5} * \frac{log_{10} \frac{5}{3}}{log_{10} 2}\right) =$$

Decision tree - Creation

EFOP-3.4.3-16-2016-00009

A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

Information gain of attribute „Fri”:



Example	Attributes										Target
	Alt	*Bar*	*Fri*	*Hun*	*Pat*	*Price*	*Rain*	*Res*	*Type*	*Est*	*WillWait*
X_1	T	F	F	T	Some	$$$	F	T	French	0–10	T
X_2	T	F	F	T	Full	$	F	F	Thai	30–60	F
X_3	F	T	F	F	Some	$	F	F	Burger	0–10	T
X_4	T	F	T	T	Full	$	F	F	Thai	10–30	T
X_5	T	F	T	F	Full	$$$	F	T	French	>60	F
X_6	F	T	F	T	Some	$$	T	T	Italian	0–10	T
X_7	F	T	F	F	None	$	T	F	Burger	0–10	F
X_8	F	F	F	T	Some	$$	T	T	Thai	0–10	T
X_9	F	T	T	F	Full	$	T	F	Burger	>60	F
X_{10}	T	T	T	T	Full	$$$	F	T	Italian	10–30	F
X_{11}	F	F	F	F	None	$	F	F	Thai	0–10	F
X_{12}	T	T	T	T	Full	$	F	F	Burger	30–60	T

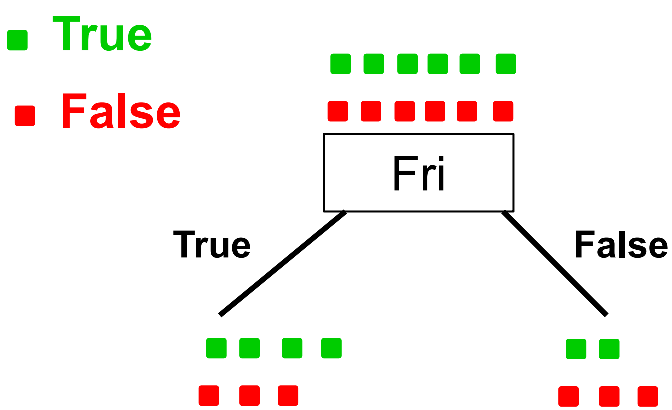
The **entropy after testing the attribute** is:

$$= \frac{7}{12} * \left(\frac{4}{7} * \frac{log_{10}\frac{7}{4}}{log_{10}2} + \frac{3}{7} * \frac{log_{10}\frac{7}{3}}{log_{10}2} \right) + \frac{5}{12} * \left(\frac{2}{5} * \frac{log_{10}\frac{5}{2}}{log_{10}2} + \frac{3}{5} * \frac{log_{10}\frac{5}{3}}{log_{10}2} \right) =$$

$$= \frac{7}{12} * \left(\frac{4}{7} * 0.80735 + \frac{3}{7} * 1.22239 \right) + \frac{5}{12} * \left(\frac{2}{5} * 1.321928 + \frac{3}{5} * 0.736966 \right) =$$

Decision tree - Creation

Information gain of attribute „Fri”:



Example	Attributes										Target
	Alt	*Bar*	*Fri*	*Hun*	*Pat*	*Price*	*Rain*	*Res*	*Type*	*Est*	*WillWait*
X_1	T	F	F	T	Some	$$$	F	T	French	0–10	T
X_2	T	F	F	T	Full	$	F	F	Thai	30–60	F
X_3	F	T	F	F	Some	$	F	F	Burger	0–10	T
X_4	T	F	T	T	Full	$	F	F	Thai	10–30	T
X_5	T	F	T	F	Full	$$$	F	T	French	>60	F
X_6	F	T	F	T	Some	$$	T	T	Italian	0–10	T
X_7	F	T	F	F	None	$	T	F	Burger	0–10	F
X_8	F	F	F	T	Some	$$	T	T	Thai	0–10	T
X_9	F	T	T	F	Full	$	T	F	Burger	>60	F
X_{10}	T	T	T	T	Full	$$$	F	T	Italian	10–30	F
X_{11}	F	F	F	F	None	$	F	F	Thai	0–10	F
X_{12}	T	T	T	T	Full	$	F	F	Burger	30–60	T

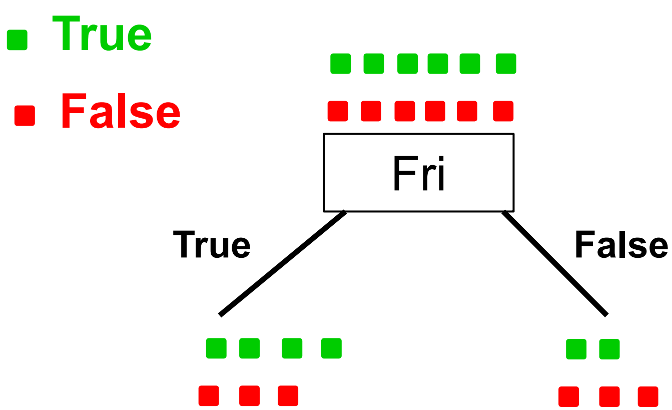
The **entropy after testing the attribute** is:

$$= \frac{7}{12} * \left(\frac{4}{7} * 0.80735 + \frac{3}{7} * 1.22239\right) + \frac{5}{12} * \left(\frac{2}{5} * 1.321928 + \frac{3}{5} * 0.736966\right) =$$

$$= 0.574714 + 0.4045628 = 0.9792768 \text{ [bit]}$$

Decision tree - Creation

Information gain of attribute „Fri”:



Example	Attributes										Target
	Alt	*Bar*	*Fri*	*Hun*	*Pat*	*Price*	*Rain*	*Res*	*Type*	*Est*	*WillWait*
X_1	T	F	F	T	Some	$$$	F	T	French	0–10	T
X_2	T	F	F	T	Full	$	F	F	Thai	30–60	F
X_3	F	T	F	F	Some	$	F	F	Burger	0–10	T
X_4	T	F	T	T	Full	$	F	F	Thai	10–30	T
X_5	T	F	T	F	Full	$$$	F	T	French	>60	F
X_6	F	T	F	T	Some	$$	T	T	Italian	0–10	T
X_7	F	T	F	F	None	$	T	F	Burger	0–10	F
X_8	F	F	F	T	Some	$$	T	T	Thai	0–10	T
X_9	F	T	T	F	Full	$	T	F	Burger	>60	F
X_{10}	T	T	T	T	Full	$$$	F	T	Italian	10–30	F
X_{11}	F	F	F	F	None	$	F	F	Thai	0–10	F
X_{12}	T	T	T	T	Full	$	F	F	Burger	30–60	T

The **entropy after testing the attribute** is:

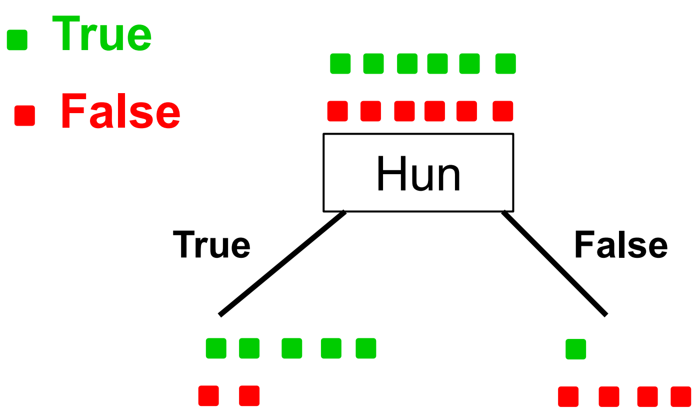
$$= 0.574714 + 0.4045628 = 0.9792768 \text{ [bit]}$$

So, the **information gain of attribute „Fri” is**: 1 [bit] -0.9792768 [bit] =

= **0.0207232 [bit]**

Decision tree - Creation

Information gain of attribute „Hun”:



Example	Attributes										Target
	Alt	*Bar*	*Fri*	*Hun*	*Pat*	*Price*	*Rain*	*Res*	*Type*	*Est*	*WillWait*
X_1	T	F	F	T	Some	$$$	F	T	French	0–10	T
X_2	T	F	F	T	Full	$	F	F	Thai	30–60	F
X_3	F	T	F	F	Some	$	F	F	Burger	0–10	T
X_4	T	F	T	T	Full	$	F	F	Thai	10–30	T
X_5	T	F	T	F	Full	$$$	F	T	French	>60	F
X_6	F	T	F	T	Some	$$	T	T	Italian	0–10	T
X_7	F	T	F	F	None	$	T	F	Burger	0–10	F
X_8	F	F	F	T	Some	$$	T	T	Thai	0–10	T
X_9	F	T	T	F	Full	$	T	F	Burger	>60	F
X_{10}	T	T	T	T	Full	$$$	F	T	Italian	10–30	F
X_{11}	F	F	F	F	None	$	F	F	Thai	0–10	F
X_{12}	T	T	T	T	Full	$	F	F	Burger	30–60	T

The **entropy after testing the attribute** is:

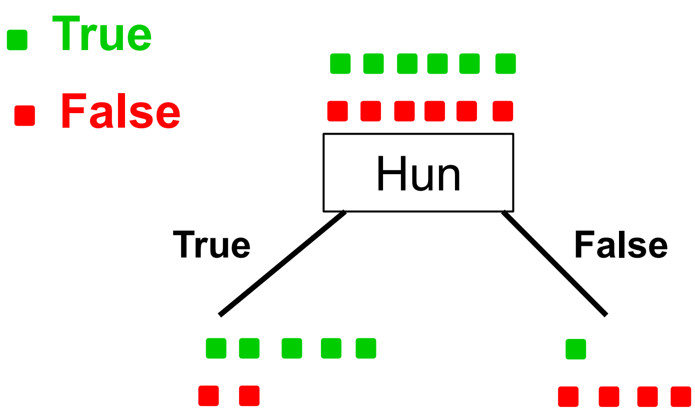
$$\frac{7}{12} * \left(\frac{5}{7} * log_2 \frac{7}{5} + \frac{2}{7} * log_2 \frac{7}{2}\right) + \frac{5}{12} * \left(\frac{1}{5} * log_2 5 + \frac{4}{5} * log_2 \frac{5}{4}\right) =$$

$$= \frac{7}{12} * \left(\frac{5}{7} * \frac{log_{10} \frac{7}{5}}{log_{10} 2} + \frac{2}{7} * \frac{log_{10} \frac{7}{2}}{log_{10} 2}\right) + \frac{5}{12} * \left(\frac{1}{5} * \frac{log_{10} 5}{log_{10} 2} + \frac{4}{5} * \frac{log_{10} \frac{5}{4}}{log_{10} 2}\right) =$$

Decision tree - Creation

EFOP-3.4.3-16-2016-00009
A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

Information gain of attribute „Hun”:



Example	Attributes										Target
	Alt	*Bar*	*Fri*	*Hun*	*Pat*	*Price*	*Rain*	*Res*	*Type*	*Est*	*WillWait*
X_1	T	F	F	T	Some	$$$	F	T	French	0–10	T
X_2	T	F	F	T	Full	$	F	F	Thai	30–60	F
X_3	F	T	F	F	Some	$	F	F	Burger	0–10	T
X_4	T	F	T	T	Full	$	F	F	Thai	10–30	T
X_5	T	F	T	F	Full	$$$	F	T	French	>60	F
X_6	F	T	F	T	Some	$$	T	T	Italian	0–10	T
X_7	F	T	F	F	None	$	T	F	Burger	0–10	F
X_8	F	F	F	T	Some	$$	T	T	Thai	0–10	T
X_9	F	T	T	F	Full	$	T	F	Burger	>60	F
X_{10}	T	T	T	T	Full	$$$	F	T	Italian	10–30	F
X_{11}	F	F	F	F	None	$	F	F	Thai	0–10	F
X_{12}	T	T	T	T	Full	$	F	F	Burger	30–60	T

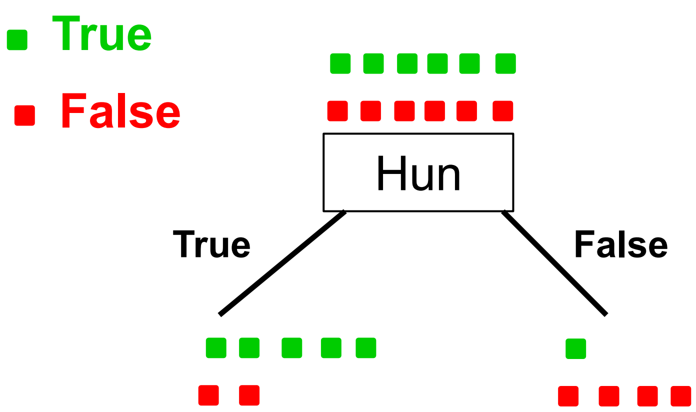
The **entropy after testing the attribute** is:

$$= \frac{7}{12} * \left(\frac{5}{7} * \frac{log_{10} \frac{7}{5}}{log_{10} 2} + \frac{2}{7} * \frac{log_{10} \frac{7}{2}}{log_{10} 2} \right) + \frac{5}{12} * \left(\frac{1}{5} * \frac{log_{10} 5}{log_{10} 2} + \frac{4}{5} * \frac{log_{10} \frac{5}{4}}{log_{10} 2} \right) =$$

$$= \frac{7}{12} * \left(\frac{5}{7} * 0.4854268 + \frac{2}{7} * 1.8073549 \right) + \frac{5}{12} * \left(\frac{1}{5} * 2.321928 + \frac{4}{5} * 0.321928 \right) =$$

Decision tree - Creation

Information gain of attribute „Hun”:



Example	Attributes										Target
	Alt	*Bar*	*Fri*	*Hun*	*Pat*	*Price*	*Rain*	*Res*	*Type*	*Est*	*WillWait*
X_1	T	F	F	T	Some	$$$	F	T	French	0–10	T
X_2	T	F	F	T	Full	$	F	F	Thai	30–60	F
X_3	F	T	F	F	Some	$	F	F	Burger	0–10	T
X_4	T	F	T	T	Full	$	F	F	Thai	10–30	T
X_5	T	F	T	F	Full	$$$	F	T	French	>60	F
X_6	F	T	F	T	Some	$$	T	T	Italian	0–10	T
X_7	F	T	F	F	None	$	T	F	Burger	0–10	F
X_8	F	F	F	T	Some	$$	T	T	Thai	0–10	T
X_9	F	T	T	F	Full	$	T	F	Burger	>60	F
X_{10}	T	T	T	T	Full	$$$	F	T	Italian	10–30	F
X_{11}	F	F	F	F	None	$	F	F	Thai	0–10	F
X_{12}	T	T	T	T	Full	$	F	F	Burger	30–60	T

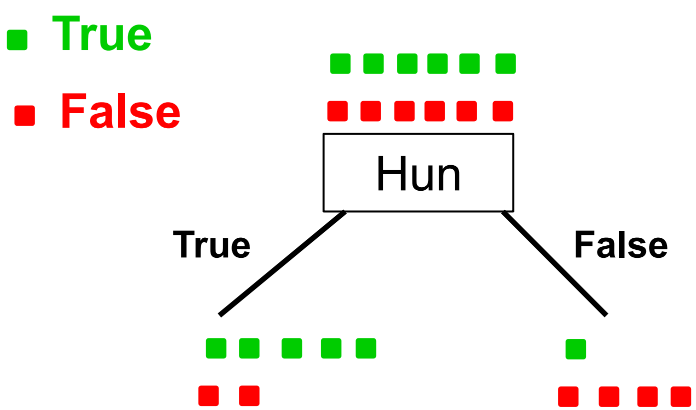
The **entropy after testing the attribute** is:

$$= \frac{7}{12} * \left(\frac{5}{7} * 0.4854268 + \frac{2}{7} * 1.8073549\right) + \frac{5}{12} * \left(\frac{1}{5} * 2.321928 + \frac{4}{5} * 0.321928\right) =$$

$$= 0.50348698 + 0.300803 = 0.80429031 \text{ [bit]}$$

Decision tree - Creation

Information gain of attribute „Hun”:



Example	Attributes										Target
	Alt	*Bar*	*Fri*	*Hun*	*Pat*	*Price*	*Rain*	*Res*	*Type*	*Est*	*WillWait*
X_1	T	F	F	T	Some	$$$	F	T	French	0–10	T
X_2	T	F	F	T	Full	$	F	F	Thai	30–60	F
X_3	F	T	F	F	Some	$	F	F	Burger	0–10	T
X_4	T	F	T	T	Full	$	F	F	Thai	10–30	T
X_5	T	F	T	F	Full	$$$	F	T	French	>60	F
X_6	F	T	F	T	Some	$$	T	T	Italian	0–10	T
X_7	F	T	F	F	None	$	T	F	Burger	0–10	F
X_8	F	F	F	T	Some	$$	T	T	Thai	0–10	T
X_9	F	T	T	F	Full	$	T	F	Burger	>60	F
X_{10}	T	T	T	T	Full	$$$	F	T	Italian	10–30	F
X_{11}	F	F	F	F	None	$	F	F	Thai	0–10	F
X_{12}	T	T	T	T	Full	$	F	F	Burger	30–60	T

The **entropy after testing the attribute** is:

$$= 0.50348698 + 0.300803 = 0.80429031 \text{ [bit]}$$

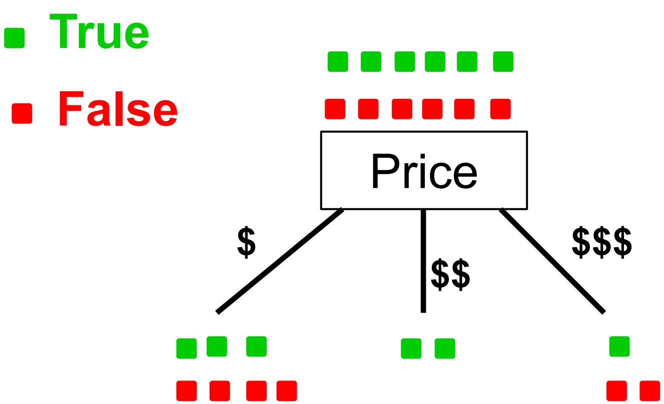
So, the **information gain of attribute „Hun” is**: 1 [bit] -0.80429031 [bit] =

= **0.19570969 [bit]**

Decision tree - Creation

EFOP-3.4.3-16-2016-00009
A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

Information gain of attribute „Price”:



Example	Attributes										Target
	Alt	*Bar*	*Fri*	*Hun*	*Pat*	*Price*	*Rain*	*Res*	*Type*	*Est*	*WillWait*
X_1	T	F	F	T	Some	$$$	F	T	French	0–10	T
X_2	T	F	F	T	Full	$	F	F	Thai	30–60	F
X_3	F	T	F	F	Some	$	F	F	Burger	0–10	T
X_4	T	F	T	T	Full	$	F	F	Thai	10–30	T
X_5	T	F	T	F	Full	$$$	F	T	French	>60	F
X_6	F	T	F	T	Some	$$	T	T	Italian	0–10	T
X_7	F	T	F	F	None	$	T	F	Burger	0–10	F
X_8	F	F	F	T	Some	$$	T	T	Thai	0–10	T
X_9	F	T	T	F	Full	$	T	F	Burger	>60	F
X_{10}	T	T	T	T	Full	$$$	F	T	Italian	10–30	F
X_{11}	F	F	F	F	None	$	F	F	Thai	0–10	F
X_{12}	T	T	T	T	Full	$	F	F	Burger	30–60	T

The **entropy after testing the attribute** is:

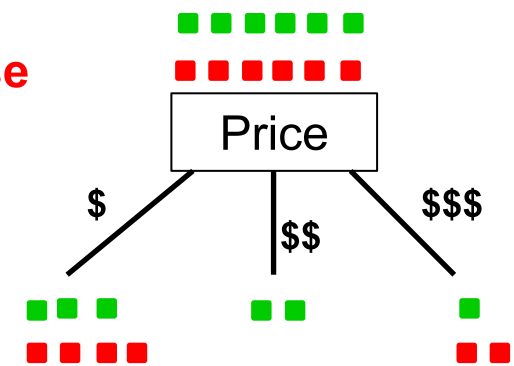
$$\frac{7}{12}*\left(\frac{3}{7}*log_2\frac{7}{3}+\frac{4}{7}*log_2\frac{7}{4}\right)+\frac{2}{12}*\left(\frac{2}{2}*log_2\frac{2}{2}+0\right)+\frac{3}{12}*\left(\frac{1}{3}*log_2 3+\frac{2}{3}*log_2\frac{3}{2}\right)=$$

$$=\frac{7}{12}*\left(\frac{3}{7}*\frac{log_{10}\frac{7}{3}}{log_{10}2}+\frac{4}{7}*\frac{log_{10}\frac{7}{4}}{log_{10}2}\right)+\frac{2}{12}*(0+0)+\frac{3}{12}*\left(\frac{1}{3}*\frac{log_{10}3}{log_{10}2}+\frac{2}{3}*\frac{log_{10}\frac{3}{2}}{log_{10}2}\right)=$$

Decision tree - Creation

Information gain of attribute „Price”:

- True
- False



Example	Attributes										Target
	Alt	*Bar*	*Fri*	*Hun*	*Pat*	*Price*	*Rain*	*Res*	*Type*	*Est*	*WillWait*
X_1	T	F	F	T	Some	$$$	F	T	French	0–10	T
X_2	T	F	F	T	Full	$	F	F	Thai	30–60	F
X_3	F	T	F	F	Some	$	F	F	Burger	0–10	T
X_4	T	F	T	T	Full	$	F	F	Thai	10–30	T
X_5	T	F	T	F	Full	$$$	F	T	French	>60	F
X_6	F	T	F	T	Some	$$	T	T	Italian	0–10	T
X_7	F	T	F	F	None	$	T	F	Burger	0–10	F
X_8	F	F	F	T	Some	$$	T	T	Thai	0–10	T
X_9	F	T	T	F	Full	$	T	F	Burger	>60	F
X_{10}	T	T	T	T	Full	$$$	F	T	Italian	10–30	F
X_{11}	F	F	F	F	None	$	F	F	Thai	0–10	F
X_{12}	T	T	T	T	Full	$	F	F	Burger	30–60	T

The **entropy after testing the attribute** is:

$$= \frac{7}{12} * \left(\frac{3}{7} * \frac{log_{10}\frac{7}{3}}{log_{10}2} + \frac{4}{7} * \frac{log_{10}\frac{7}{4}}{log_{10}2} \right) + \frac{2}{12} * (0 + 0) + \frac{3}{12} * \left(\frac{1}{3} * \frac{log_{10}3}{log_{10}2} + \frac{2}{3} * \frac{log_{10}\frac{3}{2}}{log_{10}2} \right) =$$

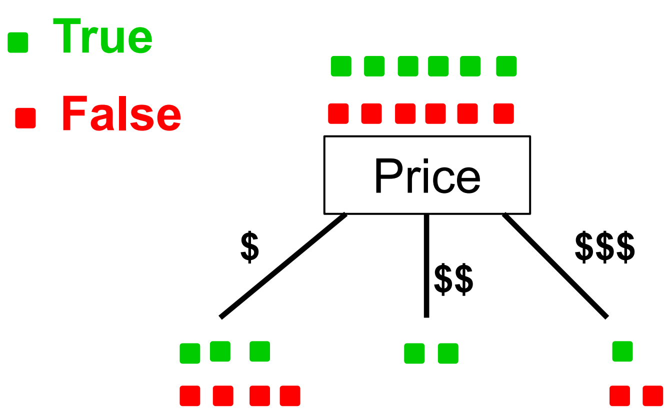
$$= \frac{7}{12} * \left(\frac{3}{7} * 1.22239 + \frac{4}{7} * 0.80735 \right) + \frac{1}{4} * (0.52832 + 0.38997) =$$

Decision tree - Creation

EFOP-3.4.3-16-2016-00009

A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

Information gain of attribute „Price”:



Example	Attributes										Target
	Alt	*Bar*	*Fri*	*Hun*	*Pat*	*Price*	*Rain*	*Res*	*Type*	*Est*	*WillWait*
X_1	T	F	F	T	Some	$$$	F	T	French	0–10	T
X_2	T	F	F	T	Full	$	F	F	Thai	30–60	F
X_3	F	T	F	F	Some	$	F	F	Burger	0–10	T
X_4	T	F	T	T	Full	$	F	F	Thai	10–30	T
X_5	T	F	T	F	Full	$$$	F	T	French	>60	F
X_6	F	T	F	T	Some	$$	T	T	Italian	0–10	T
X_7	F	T	F	F	None	$	T	F	Burger	0–10	F
X_8	F	F	F	T	Some	$$	T	T	Thai	0–10	T
X_9	F	T	T	F	Full	$	T	F	Burger	>60	F
X_{10}	T	T	T	T	Full	$$$	F	T	Italian	10–30	F
X_{11}	F	F	F	F	None	$	F	F	Thai	0–10	F
X_{12}	T	T	T	T	Full	$	F	F	Burger	30–60	T

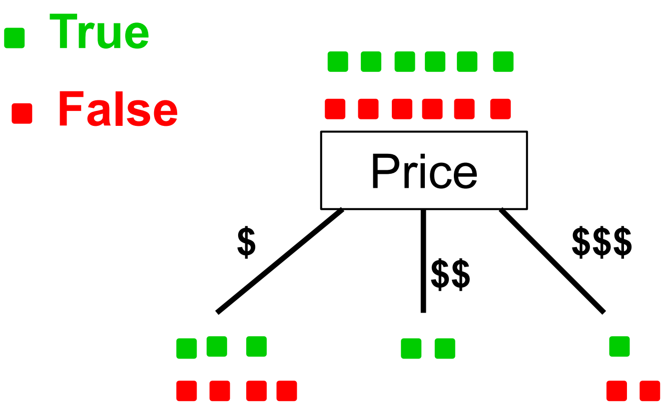
The **entropy after testing the attribute** is:

$$= \frac{7}{12} * \left(\frac{3}{7} * 1.22239 + \frac{4}{7} * 0.80735\right) + \frac{1}{4} * (0.52832 + 0.38997) =$$

$$= 0.2295725 + 0.574714 = 0.804286 \text{ [bit]}$$

Decision tree - Creation

Information gain of attribute „Price”:



Example	Attributes										Target
	Alt	*Bar*	*Fri*	*Hun*	*Pat*	*Price*	*Rain*	*Res*	*Type*	*Est*	*WillWait*
X_1	T	F	F	T	Some	$$$	F	T	French	0–10	T
X_2	T	F	F	T	Full	$	F	F	Thai	30–60	F
X_3	F	T	F	F	Some	$	F	F	Burger	0–10	T
X_4	T	F	T	T	Full	$	F	F	Thai	10–30	T
X_5	T	F	T	F	Full	$$$	F	T	French	>60	F
X_6	F	T	F	T	Some	$$	T	T	Italian	0–10	T
X_7	F	T	F	F	None	$	T	F	Burger	0–10	F
X_8	F	F	F	T	Some	$$	T	T	Thai	0–10	T
X_9	F	T	T	F	Full	$	T	F	Burger	>60	F
X_{10}	T	T	T	T	Full	$$$	F	T	Italian	10–30	F
X_{11}	F	F	F	F	None	$	F	F	Thai	0–10	F
X_{12}	T	T	T	T	Full	$	F	F	Burger	30–60	T

The **entropy after testing the attribute** is:

$$= 0.2295725 + 0.574714 = 0.804286 \text{ [bit]}$$

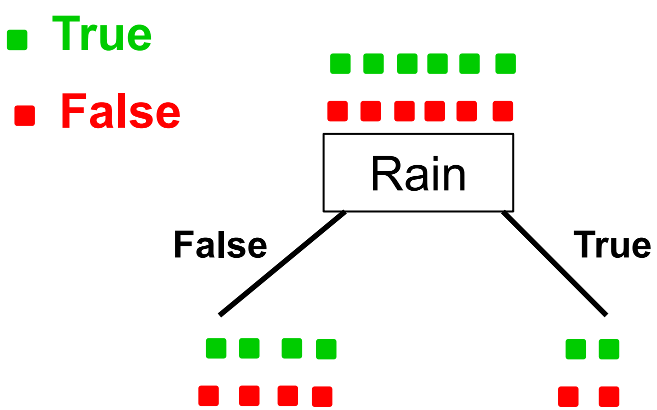
So, the **information gain of attribute „Price” is**: 1 [bit] -0.804286 [bit] =

= **0.195714 [bit]**

Decision tree - Creation

EFOP-3.4.3-16-2016-00009
A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

Information gain of attribute „Rain”:



Example	Attributes										Target
	Alt	*Bar*	*Fri*	*Hun*	*Pat*	*Price*	*Rain*	*Res*	*Type*	*Est*	*WillWait*
X_1	T	F	F	T	Some	$$$	F	T	French	0–10	T
X_2	T	F	F	T	Full	$	F	F	Thai	30–60	F
X_3	F	T	F	F	Some	$	F	F	Burger	0–10	T
X_4	T	F	T	T	Full	$	F	F	Thai	10–30	T
X_5	T	F	T	F	Full	$$$	F	T	French	>60	F
X_6	F	T	F	T	Some	$$	T	T	Italian	0–10	T
X_7	F	T	F	F	None	$	T	F	Burger	0–10	F
X_8	F	F	F	T	Some	$$	T	T	Thai	0–10	T
X_9	F	T	T	F	Full	$	T	F	Burger	>60	F
X_{10}	T	T	T	T	Full	$$$	F	T	Italian	10–30	F
X_{11}	F	F	F	F	None	$	F	F	Thai	0–10	F
X_{12}	T	T	T	T	Full	$	F	F	Burger	30–60	T

The **entropy after testing the attribute** is:

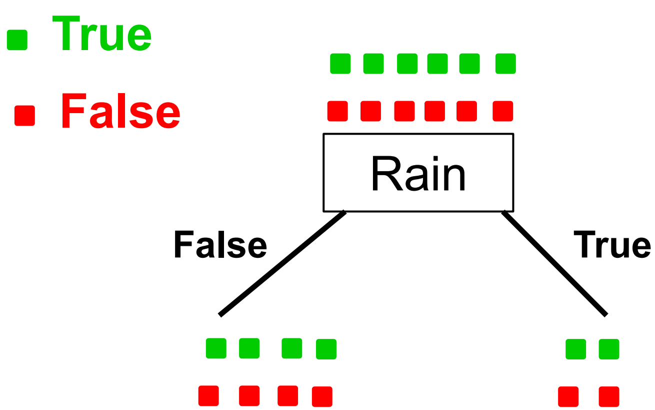
$$\frac{8}{12} * \left(\frac{4}{8} * log_2\frac{8}{4} + \frac{4}{8} * log_2\frac{8}{4}\right) + \frac{4}{12} * \left(\frac{2}{4} * log_2\frac{4}{2} + \frac{2}{4} * log_2\frac{4}{2}\right) =$$

$$= \frac{2}{3} * \left(\frac{1}{2} * log_2 2 + \frac{1}{2} * log_2 2\right) + \frac{1}{3} * \left(\frac{1}{2} * log_2 2 + \frac{1}{2} * log_2 2\right) =$$

Decision tree - Creation

EFOP-3.4.3-16-2016-00009
A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

Information gain of attribute „Rain”:



Example	Attributes										Target
	Alt	*Bar*	*Fri*	*Hun*	*Pat*	*Price*	*Rain*	*Res*	*Type*	*Est*	*WillWait*
X_1	T	F	F	T	Some	$$$	F	T	French	0–10	T
X_2	T	F	F	T	Full	$	F	F	Thai	30–60	F
X_3	F	T	F	F	Some	$	F	F	Burger	0–10	T
X_4	T	F	T	T	Full	$	F	F	Thai	10–30	T
X_5	T	F	T	F	Full	$$$	F	T	French	>60	F
X_6	F	T	F	T	Some	$$	T	T	Italian	0–10	T
X_7	F	T	F	F	None	$	T	F	Burger	0–10	F
X_8	F	F	F	T	Some	$$	T	T	Thai	0–10	T
X_9	F	T	T	F	Full	$	T	F	Burger	>60	F
X_{10}	T	T	T	T	Full	$$$	F	T	Italian	10–30	F
X_{11}	F	F	F	F	None	$	F	F	Thai	0–10	F
X_{12}	T	T	T	T	Full	$	F	F	Burger	30–60	T

The **entropy after testing the attribute** is:

$$= \frac{2}{3} * \left(\frac{1}{2} * log_2 2 + \frac{1}{2} * log_2 2\right) + \frac{1}{3} * \left(\frac{1}{2} * log_2 2 + \frac{1}{2} * log_2 2\right) =$$

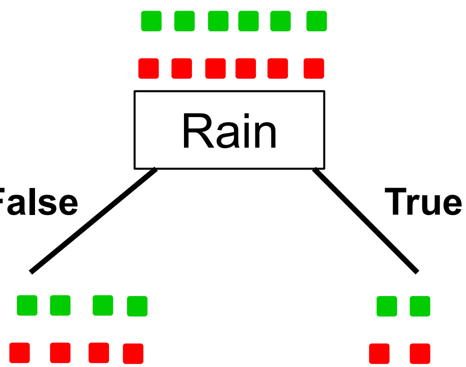
$$= \frac{2}{3} * \left(\frac{1}{2} * 1 + \frac{1}{2} * 1\right) + \frac{1}{3} * \left(\frac{1}{2} * 1 + \frac{1}{2} * 1\right) = \frac{2}{3} + \frac{1}{3} = 1 \text{ [bit]}$$

EFOP-3.4.3-16-2016-00009
A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

Decision tree - Creation

Information gain of attribute „Rain”:

- True
- False



Example	Attributes										Target
	Alt	*Bar*	*Fri*	*Hun*	*Pat*	*Price*	*Rain*	*Res*	*Type*	*Est*	*WillWait*
X_1	T	F	F	T	Some	$$$	F	T	French	0–10	T
X_2	T	F	F	T	Full	$	F	F	Thai	30–60	F
X_3	F	T	F	F	Some	$	F	F	Burger	0–10	T
X_4	T	F	T	T	Full	$	F	F	Thai	10–30	T
X_5	T	F	T	F	Full	$$$	F	T	French	>60	F
X_6	F	T	F	T	Some	$$	T	T	Italian	0–10	T
X_7	F	T	F	F	None	$	T	F	Burger	0–10	F
X_8	F	F	F	T	Some	$$	T	T	Thai	0–10	T
X_9	F	T	T	F	Full	$	T	F	Burger	>60	F
X_{10}	T	T	T	T	Full	$$$	F	T	Italian	10–30	F
X_{11}	F	F	F	F	None	$	F	F	Thai	0–10	F
X_{12}	T	T	T	T	Full	$	F	F	Burger	30–60	T

The **entropy after testing the attribute** is:

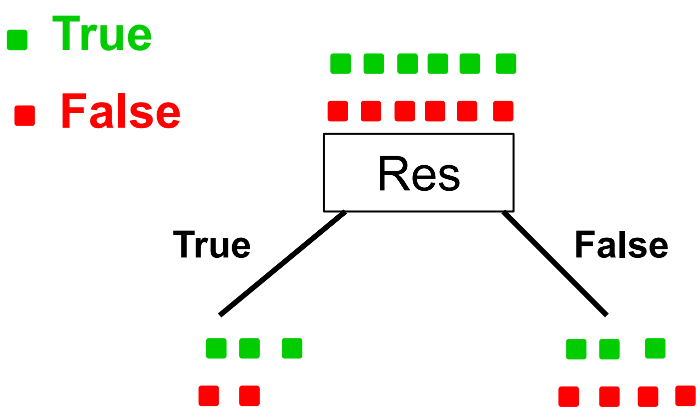
$$= \frac{2}{3} * \left(\frac{1}{2} * 1 + \frac{1}{2} * 1\right) + \frac{1}{3} * \left(\frac{1}{2} * 1 + \frac{1}{2} * 1\right) = \frac{2}{3} + \frac{1}{3} = 1 \text{ [bit]}$$

So, the **information gain of attribute „Rain” is**: 1 [bit] -1 [bit] = **0 [bit]**

Decision tree - Creation

EFOP-3.4.3-16-2016-00009
A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

Information gain of attribute „Res”:



Example	Alt	Bar	Fri	Hun	Pat	Price	Rain	Res	Type	Est	WillWait
	Attributes										*Target*
X_1	T	F	F	T	Some	$$$	F	T	French	0–10	T
X_2	T	F	F	T	Full	$	F	F	Thai	30–60	F
X_3	F	T	F	F	Some	$	F	F	Burger	0–10	T
X_4	T	F	T	T	Full	$	F	F	Thai	10–30	T
X_5	T	F	T	F	Full	$$$	F	T	French	>60	F
X_6	F	T	F	T	Some	$$	T	T	Italian	0–10	T
X_7	F	T	F	F	None	$	T	F	Burger	0–10	F
X_8	F	F	F	T	Some	$$	T	T	Thai	0–10	T
X_9	F	T	T	F	Full	$	T	F	Burger	>60	F
X_{10}	T	T	T	T	Full	$$$	F	T	Italian	10–30	F
X_{11}	F	F	F	F	None	$	F	F	Thai	0–10	F
X_{12}	T	T	T	T	Full	$	F	F	Burger	30–60	T

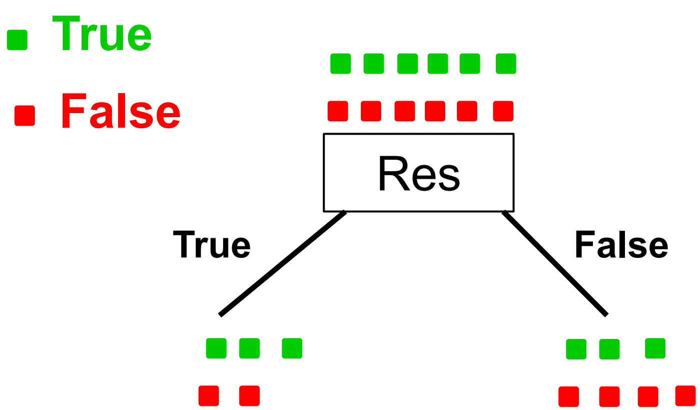
The **entropy after testing the attribute** is:

$$\frac{5}{12} * \left(\frac{3}{5} * log_2\frac{5}{3} + \frac{2}{5} * log_2\frac{5}{2}\right) + \frac{7}{12} * \left(\frac{3}{7} * log_2\frac{7}{3} + \frac{4}{7} * log_2\frac{7}{4}\right) =$$

$$= \frac{5}{12} * \left(\frac{3}{5} * \frac{log_{10}\frac{5}{3}}{log_{10}2} + \frac{2}{5} * \frac{log_{10}\frac{5}{2}}{log_{10}2}\right) + \frac{7}{12} * \left(\frac{3}{7} * \frac{log_{10}\frac{7}{3}}{log_{10}2} + \frac{4}{7} * \frac{log_{10}\frac{7}{4}}{log_{10}2}\right) =$$

Decision tree - Creation

Information gain of attribute „Res”:



Example	Attributes										Target
	Alt	*Bar*	*Fri*	*Hun*	*Pat*	*Price*	*Rain*	*Res*	*Type*	*Est*	*WillWait*
X_1	T	F	F	T	Some	$$$	F	T	French	0–10	T
X_2	T	F	F	T	Full	$	F	F	Thai	30–60	F
X_3	F	T	F	F	Some	$	F	F	Burger	0–10	T
X_4	T	F	T	T	Full	$	F	F	Thai	10–30	T
X_5	T	F	T	F	Full	$$$	F	T	French	>60	F
X_6	F	T	F	T	Some	$$	T	T	Italian	0–10	T
X_7	F	T	F	F	None	$	T	F	Burger	0–10	F
X_8	F	F	F	T	Some	$$	T	T	Thai	0–10	T
X_9	F	T	T	F	Full	$	T	F	Burger	>60	F
X_{10}	T	T	T	T	Full	$$$	F	T	Italian	10–30	F
X_{11}	F	F	F	F	None	$	F	F	Thai	0–10	F
X_{12}	T	T	T	T	Full	$	F	F	Burger	30–60	T

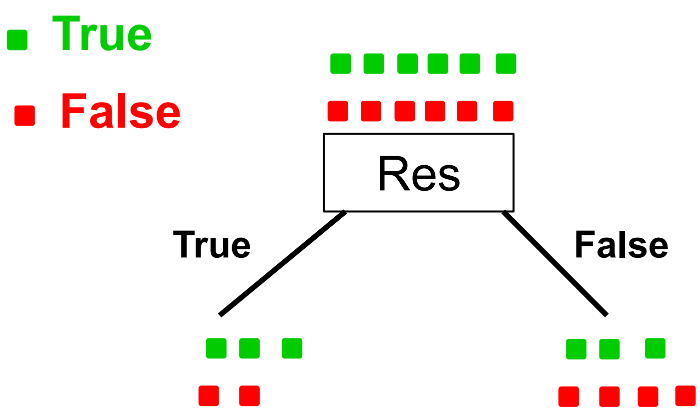
The **entropy after testing the attribute** is:

$$= \frac{5}{12} * \left(\frac{3}{5} * \frac{log_{10}\frac{5}{3}}{log_{10}2} + \frac{2}{5} * \frac{log_{10}\frac{5}{2}}{log_{10}2} \right) + \frac{7}{12} * \left(\frac{3}{7} * \frac{log_{10}\frac{7}{3}}{log_{10}2} + \frac{4}{7} * \frac{log_{10}\frac{7}{4}}{log_{10}2} \right) =$$

$$= \frac{5}{12} * \left(\frac{3}{5} * 0.736966 + \frac{2}{5} * 1.321928 \right) + \frac{7}{12} * \left(\frac{3}{7} * 1.22239 + \frac{4}{7} * 0.80735 \right) =$$

Decision tree - Creation

Information gain of attribute „Res”:



Example	Attributes										Target
	Alt	*Bar*	*Fri*	*Hun*	*Pat*	*Price*	*Rain*	*Res*	*Type*	*Est*	*WillWait*
X_1	T	F	F	T	Some	$$$	F	T	French	0–10	T
X_2	T	F	F	T	Full	$	F	F	Thai	30–60	F
X_3	F	T	F	F	Some	$	F	F	Burger	0–10	T
X_4	T	F	T	T	Full	$	F	F	Thai	10–30	T
X_5	T	F	T	F	Full	$$$	F	T	French	>60	F
X_6	F	T	F	T	Some	$$	T	T	Italian	0–10	T
X_7	F	T	F	F	None	$	T	F	Burger	0–10	F
X_8	F	F	F	T	Some	$$	T	T	Thai	0–10	T
X_9	F	T	T	F	Full	$	T	F	Burger	>60	F
X_{10}	T	T	T	T	Full	$$$	F	T	Italian	10–30	F
X_{11}	F	F	F	F	None	$	F	F	Thai	0–10	F
X_{12}	T	T	T	T	Full	$	F	F	Burger	30–60	T

The **entropy after testing the attribute** is:

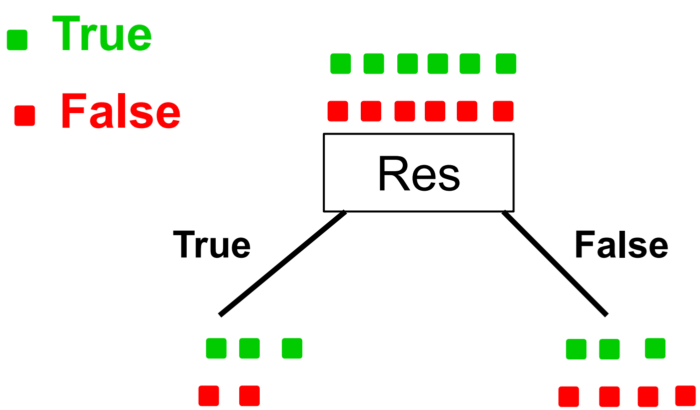
$$= \frac{5}{12} * \left(\frac{3}{5} * 0.736966 + \frac{2}{5} * 1.321928\right) + \frac{7}{12} * \left(\frac{3}{7} * 1.22239 + \frac{4}{7} * 0.80735\right) =$$

$$= 0.4045628 + 0.574714 = 0.9792768 \text{ [bit]}$$

Decision tree - Creation

EFOP-3.4.3-16-2016-00009
A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

Information gain of attribute „Res”:



Example	Attributes										Target
	Alt	*Bar*	*Fri*	*Hun*	*Pat*	*Price*	*Rain*	*Res*	*Type*	*Est*	*WillWait*
X_1	T	F	F	T	Some	$$$	F	T	French	0–10	T
X_2	T	F	F	T	Full	$	F	F	Thai	30–60	F
X_3	F	T	F	F	Some	$	F	F	Burger	0–10	T
X_4	T	F	T	T	Full	$	F	F	Thai	10–30	T
X_5	T	F	T	F	Full	$$$	F	T	French	>60	F
X_6	F	T	F	T	Some	$$	T	T	Italian	0–10	T
X_7	F	T	F	F	None	$	T	F	Burger	0–10	F
X_8	F	F	F	T	Some	$$	T	T	Thai	0–10	T
X_9	F	T	T	F	Full	$	T	F	Burger	>60	F
X_{10}	T	T	T	T	Full	$$$	F	T	Italian	10–30	F
X_{11}	F	F	F	F	None	$	F	F	Thai	0–10	F
X_{12}	T	T	T	T	Full	$	F	F	Burger	30–60	T

The **entropy after testing the attribute** is:

$$= 0.4045628 + 0.574714 = 0.9792768 \text{ [bit]}$$

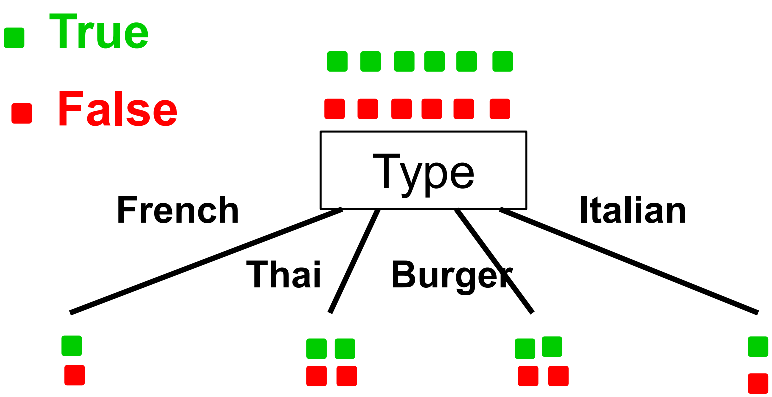
So, the **information gain of attribute „Res” is**: 1 [bit] -0.9792768 [bit] =

= **0.0207232 [bit]**

Decision tree - Creation

EFOP-3.4.3-16-2016-00009
A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

Information gain of attribute „Type”:



Example	Attributes										Target
	Alt	*Bar*	*Fri*	*Hun*	*Pat*	*Price*	*Rain*	*Res*	*Type*	*Est*	*WillWait*
X_1	T	F	F	T	Some	$$$	F	T	French	0–10	T
X_2	T	F	F	T	Full	$	F	F	Thai	30–60	F
X_3	F	T	F	F	Some	$	F	F	Burger	0–10	T
X_4	T	F	T	T	Full	$	F	F	Thai	10–30	T
X_5	T	F	T	F	Full	$$$	F	T	French	>60	F
X_6	F	T	F	T	Some	$$	T	T	Italian	0–10	T
X_7	F	T	F	F	None	$	T	F	Burger	0–10	F
X_8	F	F	F	T	Some	$$	T	T	Thai	0–10	T
X_9	F	T	T	F	Full	$	T	F	Burger	>60	F
X_{10}	T	T	T	T	Full	$$$	F	T	Italian	10–30	F
X_{11}	F	F	F	F	None	$	F	F	Thai	0–10	F
X_{12}	T	T	T	T	Full	$	F	F	Burger	30–60	T

The **entropy after testing the attribute** is:

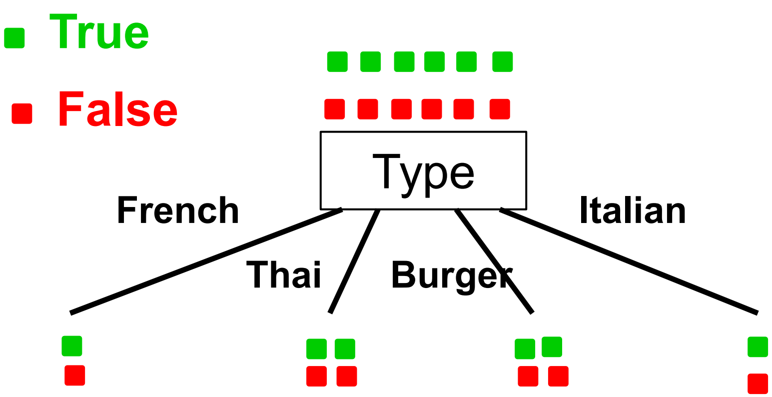
$$\frac{2}{12}*\left(\frac{1}{2}*log_2 2+\frac{1}{2}*log_2 2\right)+\frac{4}{12}*\left(\frac{2}{4}*log_2\frac{4}{2}+\frac{2}{4}*log_2\frac{4}{2}\right)+\frac{4}{12}*\left(\frac{2}{4}*log_2\frac{4}{2}+\frac{2}{4}*log_2\frac{4}{2}\right)+\frac{2}{12}*\left(\frac{1}{2}*log_2 2+\frac{1}{2}*log_2 2\right)=$$

$$=\frac{2}{12}*1+\frac{4}{12}*1+\frac{4}{12}*1+\frac{2}{12}*1=\frac{12}{12}=1 \ \text{[bit]}$$

Decision tree - Creation

EFOP-3.4.3-16-2016-00009
A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

Information gain of attribute „Type”:



Example	Attributes										Target
	Alt	*Bar*	*Fri*	*Hun*	*Pat*	*Price*	*Rain*	*Res*	*Type*	*Est*	*WillWait*
X_1	T	F	F	T	Some	$$$	F	T	French	0–10	T
X_2	T	F	F	T	Full	$	F	F	Thai	30–60	F
X_3	F	T	F	F	Some	$	F	F	Burger	0–10	T
X_4	T	F	T	T	Full	$	F	F	Thai	10–30	T
X_5	T	F	T	F	Full	$$$	F	T	French	>60	F
X_6	F	T	F	T	Some	$$	T	T	Italian	0–10	T
X_7	F	T	F	F	None	$	T	F	Burger	0–10	F
X_8	F	F	F	T	Some	$$	T	T	Thai	0–10	T
X_9	F	T	T	F	Full	$	T	F	Burger	>60	F
X_{10}	T	T	T	T	Full	$$$	F	T	Italian	10–30	F
X_{11}	F	F	F	F	None	$	F	F	Thai	0–10	F
X_{12}	T	T	T	T	Full	$	F	F	Burger	30–60	T

The **entropy after testing the attribute** is:

$$= \frac{2}{12} * 1 + \frac{4}{12} * 1 + \frac{4}{12} * 1 + \frac{2}{12} * 1 = \frac{12}{12} = 1 \text{ [bit]}$$

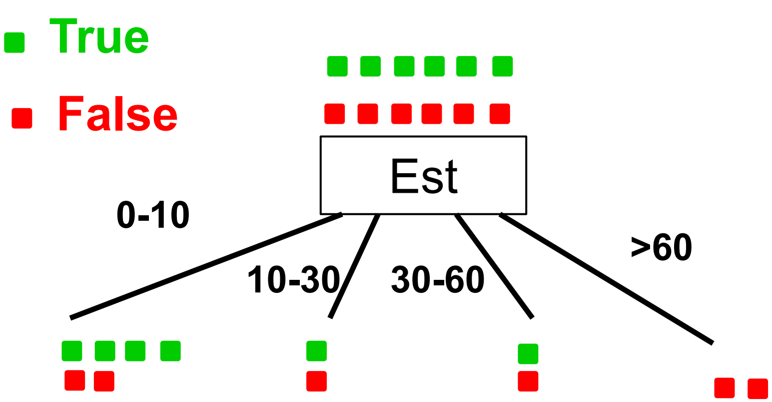
So, the **information gain of attribute „Type” is**: 1 [bit] -1 [bit] = **0 [bit]**

EFOP-3.4.3-16-2016-00009

A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

Decision tree - Creation

Information gain of attribute „Est”:



Example	Attributes										Target
	Alt	*Bar*	*Fri*	*Hun*	*Pat*	*Price*	*Rain*	*Res*	*Type*	*Est*	*WillWait*
X_1	T	F	F	T	Some	$$$	F	T	French	0–10	T
X_2	T	F	F	T	Full	$	F	F	Thai	30–60	F
X_3	F	T	F	F	Some	$	F	F	Burger	0–10	T
X_4	T	F	T	T	Full	$	F	F	Thai	10–30	T
X_5	T	F	T	F	Full	$$$	F	T	French	>60	F
X_6	F	T	F	T	Some	$$	T	T	Italian	0–10	T
X_7	F	T	F	F	None	$	T	F	Burger	0–10	F
X_8	F	F	F	T	Some	$$	T	T	Thai	0–10	T
X_9	F	T	T	F	Full	$	T	F	Burger	>60	F
X_{10}	T	T	T	T	Full	$$$	F	T	Italian	10–30	F
X_{11}	F	F	F	F	None	$	F	F	Thai	0–10	F
X_{12}	T	T	T	T	Full	$	F	F	Burger	30–60	T

The **entropy after testing the attribute** is:

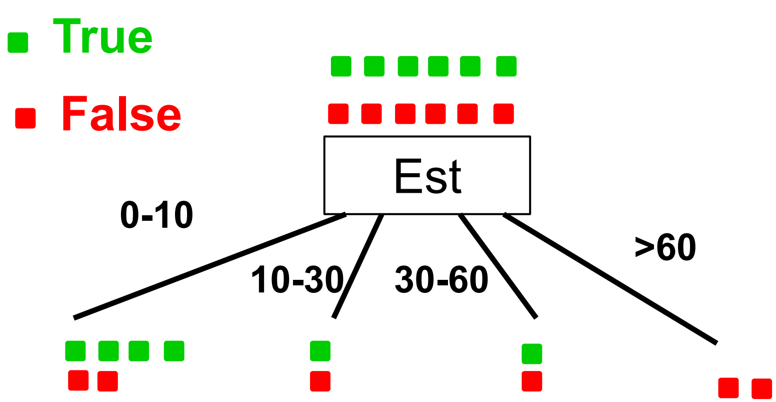
$$\frac{6}{12} * \left(\frac{4}{6} * log_2\frac{6}{4} + \frac{2}{6} * log_2\frac{6}{2}\right) + \frac{2}{12} * \left(\frac{1}{2} * log_2 2 + \frac{1}{2} * log_2 2\right) + \frac{2}{12} * \left(\frac{1}{2} * log_2 2 + \frac{1}{2} * log_2 2\right) + \frac{2}{12} * \left(0 + \frac{2}{2} * log_2\frac{2}{2}\right) =$$

$$= \frac{1}{2} * \left(\frac{2}{3} * log_2\frac{3}{2} + \frac{1}{3} * log_2 3\right) + \frac{2}{12} * 1 + \frac{2}{12} * 1 + \frac{2}{12} * 0 =$$

Decision tree - Creation

EFOP-3.4.3-16-2016-00009
A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

Information gain of attribute „Est”:



Example	Attributes										Target
	Alt	*Bar*	*Fri*	*Hun*	*Pat*	*Price*	*Rain*	*Res*	*Type*	*Est*	*WillWait*
X_1	T	F	F	T	Some	$$$	F	T	French	0–10	T
X_2	T	F	F	T	Full	$	F	F	Thai	30–60	F
X_3	F	T	F	F	Some	$	F	F	Burger	0–10	T
X_4	T	F	T	T	Full	$	F	F	Thai	10–30	T
X_5	T	F	T	F	Full	$$$	F	T	French	>60	F
X_6	F	T	F	T	Some	$$	T	T	Italian	0–10	T
X_7	F	T	F	F	None	$	T	F	Burger	0–10	F
X_8	F	F	F	T	Some	$$	T	T	Thai	0–10	T
X_9	F	T	T	F	Full	$	T	F	Burger	>60	F
X_{10}	T	T	T	T	Full	$$$	F	T	Italian	10–30	F
X_{11}	F	F	F	F	None	$	F	F	Thai	0–10	F
X_{12}	T	T	T	T	Full	$	F	F	Burger	30–60	T

The **entropy after testing the attribute** is:

$$= \frac{1}{2} * \left(\frac{2}{3} * log_2 \frac{3}{2} + \frac{1}{3} * log_2 3\right) + \frac{2}{12} * 1 + \frac{2}{12} * 1 + \frac{2}{12} * 0 =$$

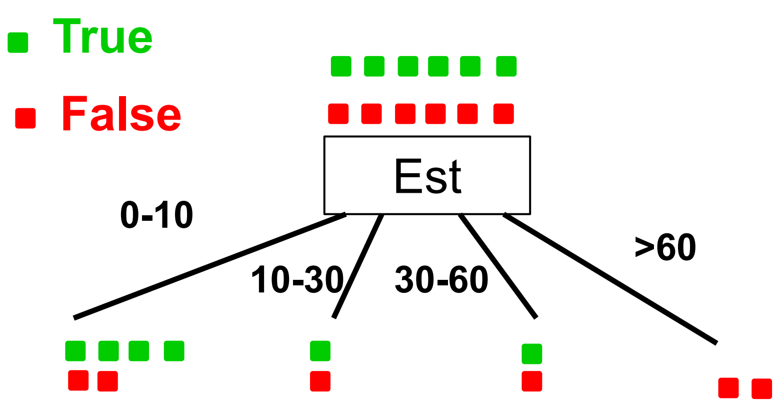
$$= \frac{1}{2} * \left(\frac{2}{3} * \frac{log_{10} \frac{3}{2}}{log_{10} 2} + \frac{1}{3} * \frac{log_{10} 3}{log_{10} 2}\right) + \frac{1}{3} = \frac{1}{2} * (0.38997 + 0.52832) + \frac{1}{3} =$$

Decision tree - Creation

EFOP-3.4.3-16-2016-00009

A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

Information gain of attribute „Est”:



Example	Attributes										Target
	Alt	*Bar*	*Fri*	*Hun*	*Pat*	*Price*	*Rain*	*Res*	*Type*	*Est*	*WillWait*
X_1	T	F	F	T	Some	$$$	F	T	French	0–10	T
X_2	T	F	F	T	Full	$	F	F	Thai	30–60	F
X_3	F	T	F	F	Some	$	F	F	Burger	0–10	T
X_4	T	F	T	T	Full	$	F	F	Thai	10–30	T
X_5	T	F	T	F	Full	$$$	F	T	French	>60	F
X_6	F	T	F	T	Some	$$	T	T	Italian	0–10	T
X_7	F	T	F	F	None	$	T	F	Burger	0–10	F
X_8	F	F	F	T	Some	$$	T	T	Thai	0–10	T
X_9	F	T	T	F	Full	$	T	F	Burger	>60	F
X_{10}	T	T	T	T	Full	$$$	F	T	Italian	10–30	F
X_{11}	F	F	F	F	None	$	F	F	Thai	0–10	F
X_{12}	T	T	T	T	Full	$	F	F	Burger	30–60	T

The **entropy after testing the attribute** is:

$$= \frac{1}{2} * \left(\frac{2}{3} * \frac{log_{10}\frac{3}{2}}{log_{10}2} + \frac{1}{3} * \frac{log_{10}3}{log_{10}2} \right) + \frac{1}{3} = \frac{1}{2} * (0.38997 + 0.52832) + \frac{1}{3} =$$

$$= 0.459145 + \frac{1}{3} = 0.7924783 \text{ [bit]}$$

Decision tree - Creation

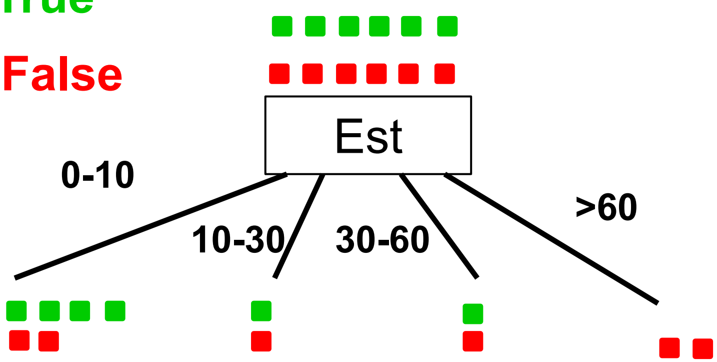
EFOP-3.4.3-16-2016-00009

A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

Information gain of attribute „Est”:

True

False



Example	Attributes										Target
	Alt	*Bar*	*Fri*	*Hun*	*Pat*	*Price*	*Rain*	*Res*	*Type*	*Est*	*WillWait*
X_1	T	F	F	T	Some	$$$	F	T	French	0–10	T
X_2	T	F	F	T	Full	$	F	F	Thai	30–60	F
X_3	F	T	F	F	Some	$	F	F	Burger	0–10	T
X_4	T	F	T	T	Full	$	F	F	Thai	10–30	T
X_5	T	F	T	F	Full	$$$	F	T	French	>60	F
X_6	F	T	F	T	Some	$$	T	T	Italian	0–10	T
X_7	F	T	F	F	None	$	T	F	Burger	0–10	F
X_8	F	F	F	T	Some	$$	T	T	Thai	0–10	T
X_9	F	T	T	F	Full	$	T	F	Burger	>60	F
X_{10}	T	T	T	T	Full	$$$	F	T	Italian	10–30	F
X_{11}	F	F	F	F	None	$	F	F	Thai	0–10	F
X_{12}	T	T	T	T	Full	$	F	F	Burger	30–60	T

The **entropy after testing the attribute** is:

$$= 0.459145 + \frac{1}{3} = 0.7924783 \text{ [bit]}$$

So, the **information gain of attribute „Res” is**: 1 [bit] -0.7924783 [bit] =

= **0.2075217 [bit]**

Decision tree - Creation

The **information gain** values we got **for the attributes** are:

Attribute	Information gain of the attribute
Alt	0 [bit]
Bar	0 [bit]
Fri	0.0207232 [bit]
Hun	0.19570969 [bit]
Patrons	0.540855 [bit]
Price	0.195714 [bit]
Rain	0 [bit]
Res	0.0207232 [bit]
Type	0 [bit]
Est	0.2075217 [bit]

Decision tree - Creation

The **information gain** values we got **for the attributes** are:

Attribute	Information gain of the attribute
Alt	0 [bit]
Bar	0 [bit]
Fri	0.0207232 [bit]
Hun	0.19570969 [bit]
Patrons	**0.540855 [bit]**
Price	0.195714 [bit]
Rain	0 [bit]
Res	0.0207232 [bit]
Type	0 [bit]
Est	0.2075217 [bit]

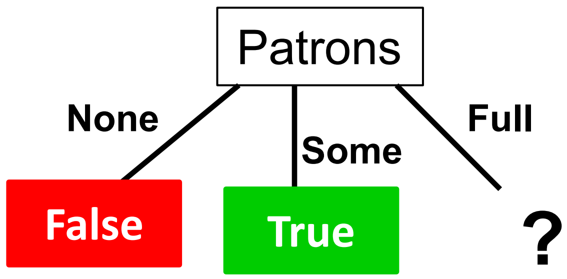
The attribute with the highest information gain is „**Patrons**", so it **will be the root of the decision tree**.

Decision tree - Creation

EFOP-3.4.3-16-2016-00009

A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

After determining the root of the decision tree, **the first level** of that is:



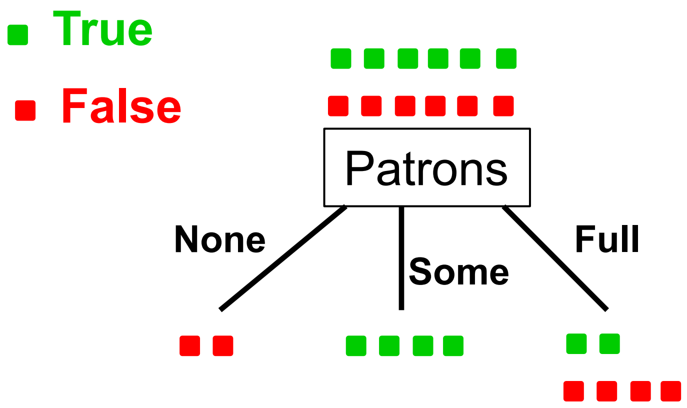
By testing the most informative attribute, we have clear answer in case of its two possible values. However, **for the third branch we need more investigations**.

Let us create the whole second level of the decision tree!

Decision tree - Creation

EFOP-3.4.3-16-2016-00009
A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

Recall, that after testing the attribute „Patrons", **in the third (Full) branch only 6 examples remained, as follows**:



So, for choosing the best attribute for the second level of the decision tree, **the initial entropy has changed**:

$$\frac{2}{6} * log_2 \frac{6}{2} + \frac{4}{6} * log_2 \frac{6}{4} = \frac{2}{6} * \frac{log_{10} 3}{log_{10} 2} + \frac{4}{6} * \frac{log_{10} \frac{3}{2}}{log_{10} 2} \approx$$

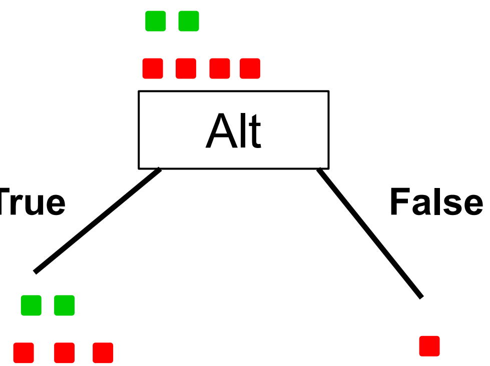
$$\approx \frac{1}{3} * 1.5849625 + \frac{2}{3} * 0.5849625 \approx 0.9182958 \text{ [bit]}$$

After that, we calculate the **information gain for all the remaining attributes**, regarding the remaining examples in this *(Patrons = Full)* branch.

Decision tree - Creation

Information gain of attribute „Alt”:

- True
- False



Example	Attributes										Target
	Alt	*Bar*	*Fri*	*Hun*	*Pat*	*Price*	*Rain*	*Res*	*Type*	*Est*	*WillWait*
X_1	T	F	F	T	Some	$$$	F	T	French	0–10	T
X_2	T	F	F	T	Full	$	F	F	Thai	30–60	F
X_3	F	T	F	F	Some	$	F	F	Burger	0–10	T
X_4	T	F	T	T	Full	$	F	F	Thai	10–30	T
X_5	T	F	T	F	Full	$$$	F	T	French	>60	F
X_6	F	T	F	T	Some	$$	T	T	Italian	0–10	T
X_7	F	T	F	F	None	$	T	F	Burger	0–10	F
X_8	F	F	F	T	Some	$$	T	T	Thai	0–10	T
X_9	F	T	T	F	Full	$	T	F	Burger	>60	F
X_{10}	T	T	T	T	Full	$$$	F	T	Italian	10–30	F
X_{11}	F	F	F	F	None	$	F	F	Thai	0–10	F
X_{12}	T	T	T	T	Full	$	F	F	Burger	30–60	T

The **entropy after testing the attribute** is:

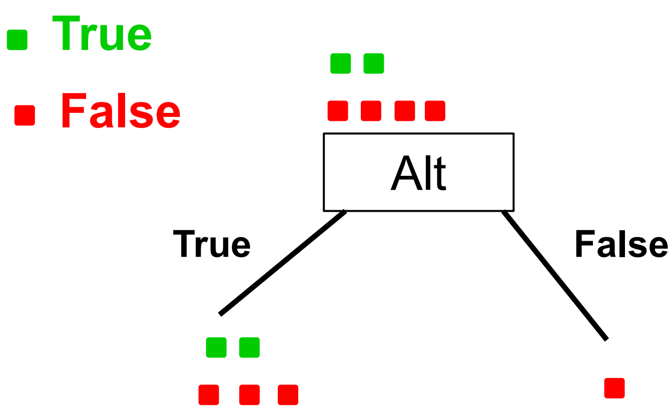
$$\frac{5}{6}*\left(\frac{2}{5}*log_2\frac{5}{2}+\frac{3}{5}*log_2\frac{5}{3}\right)+\frac{1}{6}*(0+1*log_2 1)=$$

$$=\frac{5}{6}*\left(\frac{2}{5}*\frac{log_{10}\frac{5}{2}}{log_{10}2}+\frac{3}{5}*\frac{log_{10}\frac{5}{3}}{log_{10}2}\right)+0\approx\frac{5}{6}*\left(\frac{2}{5}*1.321928+\frac{3}{5}*0.736966\right)=0.8091257 \text{ [bit]}$$

Decision tree - Creation

EFOP-3.4.3-16-2016-00009
A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

Information gain of attribute „Alt”:



Example	Attributes										Target
	Alt	*Bar*	*Fri*	*Hun*	*Pat*	*Price*	*Rain*	*Res*	*Type*	*Est*	*WillWait*
X_1	T	F	F	T	Some	$$$	F	T	French	0–10	T
X_2	T	F	F	T	Full	$	F	F	Thai	30–60	F
X_3	F	T	F	F	Some	$	F	F	Burger	0–10	T
X_4	T	F	T	T	Full	$	F	F	Thai	10–30	T
X_5	T	F	T	F	Full	$$$	F	T	French	>60	F
X_6	F	T	F	T	Some	$$	T	T	Italian	0–10	T
X_7	F	T	F	F	None	$	T	F	Burger	0–10	F
X_8	F	F	F	T	Some	$$	T	T	Thai	0–10	T
X_9	F	T	T	F	Full	$	T	F	Burger	>60	F
X_{10}	T	T	T	T	Full	$$$	F	T	Italian	10–30	F
X_{11}	F	F	F	F	None	$	F	F	Thai	0–10	F
X_{12}	T	T	T	T	Full	$	F	F	Burger	30–60	T

The **entropy after testing the attribute** is:

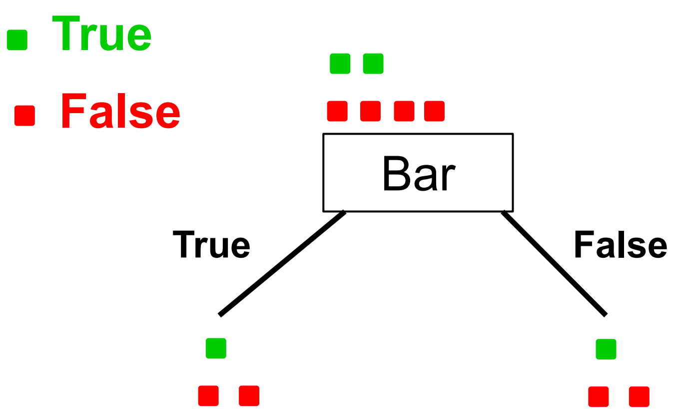
$$= \frac{5}{6} * \left(\frac{2}{5} * \frac{log_{10}\frac{5}{2}}{log_{10}2} + \frac{3}{5} * \frac{log_{10}\frac{5}{3}}{log_{10}2} \right) + 0 \approx \frac{5}{6} * \left(\frac{2}{5} * 1.321928 + \frac{3}{5} * 0.736966 \right) = 0.8091257 \text{ [bit]}$$

So, the **information gain of attribute „Alt” (in the branch *Patrons=Full*) is**:

0.9182958 [bit] - 0.8091257 [bit] = **0.1091701 [bit]**

Decision tree - Creation

Information gain of attribute „Bar”:



Example	Attributes										Target
	Alt	*Bar*	*Fri*	*Hun*	*Pat*	*Price*	*Rain*	*Res*	*Type*	*Est*	*WillWait*
X_1	T	F	F	T	Some	\$\$\$	F	T	French	0–10	T
X_2	T	F	F	T	Full	\$	F	F	Thai	30–60	F
X_3	F	T	F	F	Some	\$	F	F	Burger	0–10	T
X_4	T	F	T	T	Full	\$	F	F	Thai	10–30	T
X_5	T	F	T	F	Full	\$\$\$	F	T	French	>60	F
X_6	F	T	F	T	Some	\$\$	T	T	Italian	0–10	T
X_7	F	T	F	F	None	\$	T	F	Burger	0–10	F
X_8	F	F	F	T	Some	\$\$	T	T	Thai	0–10	T
X_9	F	T	T	F	Full	\$	T	F	Burger	>60	F
X_{10}	T	T	T	T	Full	\$\$\$	F	T	Italian	10–30	F
X_{11}	F	F	F	F	None	\$	F	F	Thai	0–10	F
X_{12}	T	T	T	T	Full	\$	F	F	Burger	30–60	T

The **entropy after testing the attribute** is:

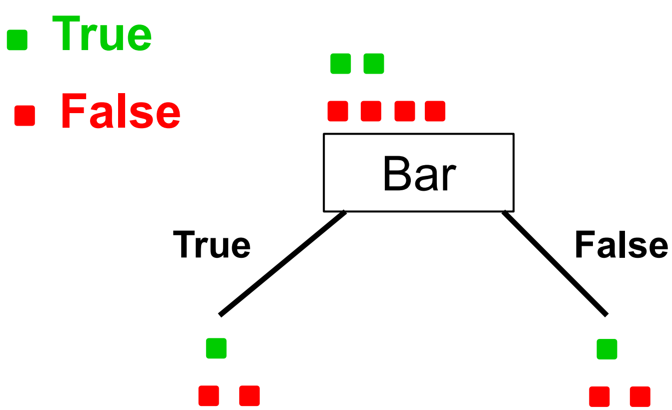
$$\frac{3}{6} * \left(\frac{1}{3} * log_2 3 + \frac{2}{3} * log_2 \frac{3}{2}\right) + \frac{3}{6} * \left(\frac{1}{3} * log_2 3 + \frac{2}{3} * log_2 \frac{3}{2}\right) =$$

$$= 2 * \frac{1}{2} * \left(\frac{1}{3} * \frac{log_{10} 3}{log_{10} 2} + \frac{2}{3} * \frac{log_{10} \frac{3}{2}}{log_{10} 2}\right) \approx 0.52832 + 0.38997 = 0.91829 \text{ [bit]}$$

Decision tree - Creation

EFOP-3.4.3-16-2016-00009
A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

Information gain of attribute „Bar”:



Example	Attributes										Target
	Alt	*Bar*	*Fri*	*Hun*	*Pat*	*Price*	*Rain*	*Res*	*Type*	*Est*	*WillWait*
X_1	T	F	F	T	Some	$$$	F	T	French	0–10	T
X_2	T	F	F	T	Full	$	F	F	Thai	30–60	F
X_3	F	T	F	F	Some	$	F	F	Burger	0–10	T
X_4	T	F	T	T	Full	$	F	F	Thai	10–30	T
X_5	T	F	T	F	Full	$$$	F	T	French	>60	F
X_6	F	T	F	T	Some	$$	T	T	Italian	0–10	T
X_7	F	T	F	F	None	$	T	F	Burger	0–10	F
X_8	F	F	F	T	Some	$$	T	T	Thai	0–10	T
X_9	F	T	T	F	Full	$	T	F	Burger	>60	F
X_{10}	T	T	T	T	Full	$$$	F	T	Italian	10–30	F
X_{11}	F	F	F	F	None	$	F	F	Thai	0–10	F
X_{12}	T	T	T	T	Full	$	F	F	Burger	30–60	T

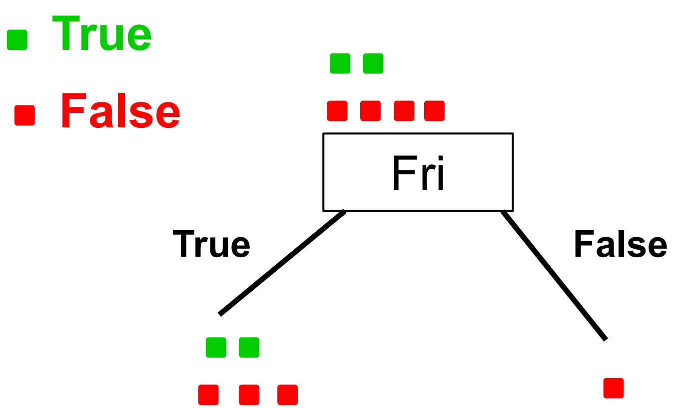
The **entropy after testing the attribute** is:

$$= 2 * \frac{1}{2} * \left(\frac{1}{3} * \frac{log_{10}3}{log_{10}2} + \frac{2}{3} * \frac{log_{10}\frac{3}{2}}{log_{10}2} \right) \approx 0.52832 + 0.38997 = 0.91829 \text{ [bit]}$$

So, the **information gain of attribute „Bar” (in the branch *Patrons=Full*) is:**

0.91829 [bit] - 0.91829 [bit] = **0 [bit]**

Information gain of attribute „Fri”:



Example	Attributes										Target
	Alt	*Bar*	*Fri*	*Hun*	*Pat*	*Price*	*Rain*	*Res*	*Type*	*Est*	*WillWait*
X_1	T	F	F	T	Some	$$$	F	T	French	0–10	T
X_2	T	F	F	T	Full	$	F	F	Thai	30–60	F
X_3	F	T	F	F	Some	$	F	F	Burger	0–10	T
X_4	T	F	T	T	Full	$	F	F	Thai	10–30	T
X_5	T	F	T	F	Full	$$$	F	T	French	>60	F
X_6	F	T	F	T	Some	$$	T	T	Italian	0–10	T
X_7	F	T	F	F	None	$	T	F	Burger	0–10	F
X_8	F	F	F	T	Some	$$	T	T	Thai	0–10	T
X_9	F	T	T	F	Full	$	T	F	Burger	>60	F
X_{10}	T	T	T	T	Full	$$$	F	T	Italian	10–30	F
X_{11}	F	F	F	F	None	$	F	F	Thai	0–10	F
X_{12}	T	T	T	T	Full	$	F	F	Burger	30–60	T

The **entropy after testing the attribute** is:

$$\frac{5}{6} * \left(\frac{2}{5} * log_2 \frac{5}{2} + \frac{3}{5} * log_2 \frac{5}{3}\right) + \frac{1}{6} * (0 + 1 * log_2 1) =$$

$$= \frac{5}{6} * \left(\frac{2}{5} * \frac{log_{10} \frac{5}{2}}{log_{10} 2} + \frac{3}{5} * \frac{log_{10} \frac{5}{3}}{log_{10} 2}\right) + 0 \approx \frac{5}{6} * \left(\frac{2}{5} * 1.321928 + \frac{3}{5} * 0.736966\right) = 0.8091257 \text{ [bit]}$$

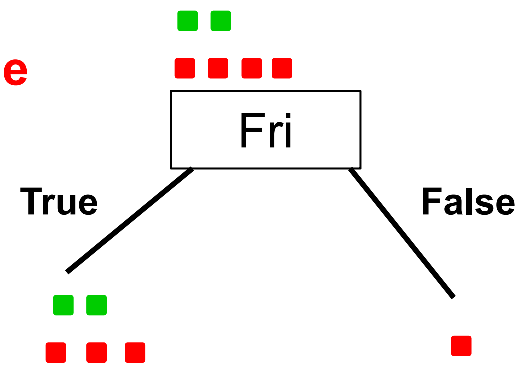
Decision tree - Creation

EFOP-3.4.3-16-2016-00009

A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

Information gain of attribute „Fri":

- True
- False



Example	Attributes										Target
	Alt	*Bar*	*Fri*	*Hun*	*Pat*	*Price*	*Rain*	*Res*	*Type*	*Est*	*WillWait*
X_1	T	F	F	T	Some	$$$	F	T	French	0–10	T
X_2	T	F	F	T	Full	$	F	F	Thai	30–60	F
X_3	F	T	F	F	Some	$	F	F	Burger	0–10	T
X_4	T	F	T	T	Full	$	F	F	Thai	10–30	T
X_5	T	F	T	F	Full	$$$	F	T	French	>60	F
X_6	F	T	F	T	Some	$$	T	T	Italian	0–10	T
X_7	F	T	F	F	None	$	T	F	Burger	0–10	F
X_8	F	F	F	T	Some	$$	T	T	Thai	0–10	T
X_9	F	T	T	F	Full	$	T	F	Burger	>60	F
X_{10}	T	T	T	T	Full	$$$	F	T	Italian	10–30	F
X_{11}	F	F	F	F	None	$	F	F	Thai	0–10	F
X_{12}	T	T	T	T	Full	$	F	F	Burger	30–60	T

The **entropy after testing the attribute** is:

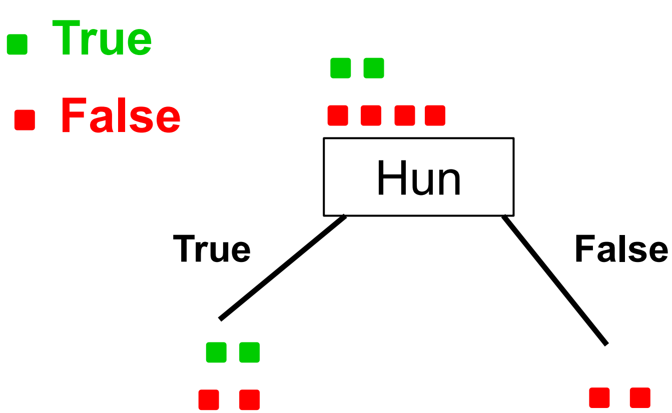
$$= \frac{5}{6} * \left(\frac{2}{5} * \frac{log_{10}\frac{5}{2}}{log_{10}2} + \frac{3}{5} * \frac{log_{10}\frac{5}{3}}{log_{10}2} \right) + 0 \approx \frac{5}{6} * \left(\frac{2}{5} * 1.321928 + \frac{3}{5} * 0.736966 \right) = 0.8091257 \text{ [bit]}$$

So, the **information gain of attribute „Fri" (in the branch *Patrons=Full*) is:**

0.9182958 [bit] - 0.8091257 [bit] = **0.1091701 [bit]**

Decision tree - Creation

Information gain of attribute „Hun”:



Example	Attributes										Target
	Alt	*Bar*	*Fri*	*Hun*	*Pat*	*Price*	*Rain*	*Res*	*Type*	*Est*	*WillWait*
X_1	T	F	F	T	Some	$$$	F	T	French	0–10	T
X_2	T	F	F	T	Full	$	F	F	Thai	30–60	F
X_3	F	T	F	F	Some	$	F	F	Burger	0–10	T
X_4	T	F	T	T	Full	$	F	F	Thai	10–30	T
X_5	T	F	T	F	Full	$$$	F	T	French	>60	F
X_6	F	T	F	T	Some	$$	T	T	Italian	0–10	T
X_7	F	T	F	F	None	$	T	F	Burger	0–10	F
X_8	F	F	F	T	Some	$$	T	T	Thai	0–10	T
X_9	F	T	T	F	Full	$	T	F	Burger	>60	F
X_{10}	T	T	T	T	Full	$$$	F	T	Italian	10–30	F
X_{11}	F	F	F	F	None	$	F	F	Thai	0–10	F
X_{12}	T	T	T	T	Full	$	F	F	Burger	30–60	T

The **entropy after testing the attribute** is:

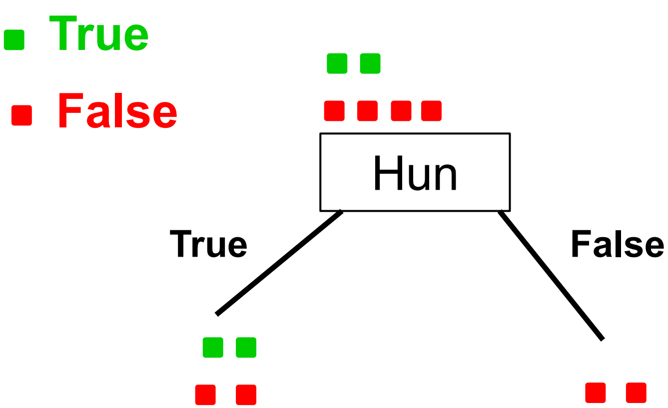
$$\frac{4}{6} * \left(\frac{2}{4} * log_2 \frac{4}{2} + \frac{2}{4} * log_2 \frac{4}{2}\right) + \frac{2}{6} * (0 + 1 * log_2 1) =$$

$$= \frac{2}{3} * \left(2 * \frac{1}{2} * log_2 2\right) + 0 = \frac{2}{3} = 0.6666666 \text{ [bit]}$$

Decision tree - Creation

EFOP-3.4.3-16-2016-00009
A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

Information gain of attribute „Hun”:



Example	Attributes										Target
	Alt	*Bar*	*Fri*	*Hun*	*Pat*	*Price*	*Rain*	*Res*	*Type*	*Est*	*WillWait*
X_1	T	F	F	T	Some	$$$	F	T	French	0–10	T
X_2	T	F	F	T	Full	$	F	F	Thai	30–60	F
X_3	F	T	F	F	Some	$	F	F	Burger	0–10	T
X_4	T	F	T	T	Full	$	F	F	Thai	10–30	T
X_5	T	F	T	F	Full	$$$	F	T	French	>60	F
X_6	F	T	F	T	Some	$$	T	T	Italian	0–10	T
X_7	F	T	F	F	None	$	T	F	Burger	0–10	F
X_8	F	F	F	T	Some	$$	T	T	Thai	0–10	T
X_9	F	T	T	F	Full	$	T	F	Burger	>60	F
X_{10}	T	T	T	T	Full	$$$	F	T	Italian	10–30	F
X_{11}	F	F	F	F	None	$	F	F	Thai	0–10	F
X_{12}	T	T	T	T	Full	$	F	F	Burger	30–60	T

The **entropy after testing the attribute** is:

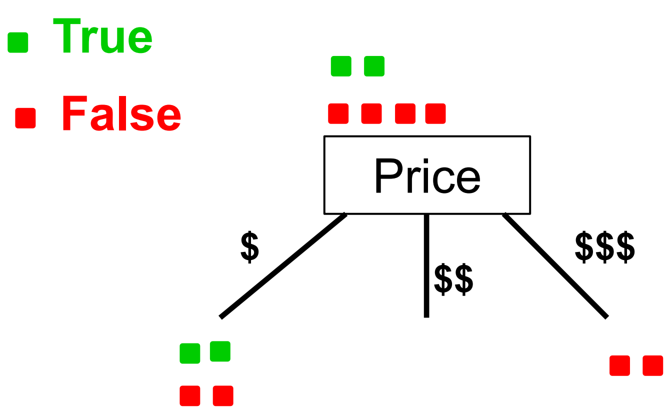
$$= \frac{2}{3} * \left(2 * \frac{1}{2} * log_2 2\right) + 0 = \frac{2}{3} = 0.6666666 \text{ [bit]}$$

So, the **information gain of attribute „Hun” (in the branch *Patrons=Full*) is**:

0.9182958 [bit] - 0.6666666 [bit] = **0.2516292 [bit]**

Decision tree - Creation

Information gain of attribute „Price”:



Example	Attributes										Target
	Alt	*Bar*	*Fri*	*Hun*	*Pat*	*Price*	*Rain*	*Res*	*Type*	*Est*	*WillWait*
X_1	*T*	*F*	*F*	*T*	*Some*	*$$$*	*F*	*T*	*French*	*0–10*	*T*
X_2	*T*	*F*	*F*	*T*	*Full*	*$*	*F*	*F*	*Thai*	*30–60*	*F*
X_3	*F*	*T*	*F*	*F*	*Some*	*$*	*F*	*F*	*Burger*	*0–10*	*T*
X_4	*T*	*F*	*T*	*T*	*Full*	*$*	*F*	*F*	*Thai*	*10–30*	*T*
X_5	*T*	*F*	*T*	*F*	*Full*	*$$$*	*F*	*T*	*French*	*>60*	*F*
X_6	*F*	*T*	*F*	*T*	*Some*	*$$*	*T*	*T*	*Italian*	*0–10*	*T*
X_7	*F*	*T*	*F*	*F*	*None*	*$*	*T*	*F*	*Burger*	*0–10*	*F*
X_8	*F*	*F*	*F*	*T*	*Some*	*$$*	*T*	*T*	*Thai*	*0–10*	*T*
X_9	*F*	*T*	*T*	*F*	*Full*	*$*	*T*	*F*	*Burger*	*>60*	*F*
X_{10}	*T*	*T*	*T*	*T*	*Full*	*$$$*	*F*	*T*	*Italian*	*10–30*	*F*
X_{11}	*F*	*F*	*F*	*F*	*None*	*$*	*F*	*F*	*Thai*	*0–10*	*F*
X_{12}	*T*	*T*	*T*	*T*	*Full*	*$*	*F*	*F*	*Burger*	*30–60*	*T*

The **entropy after testing the attribute** is:

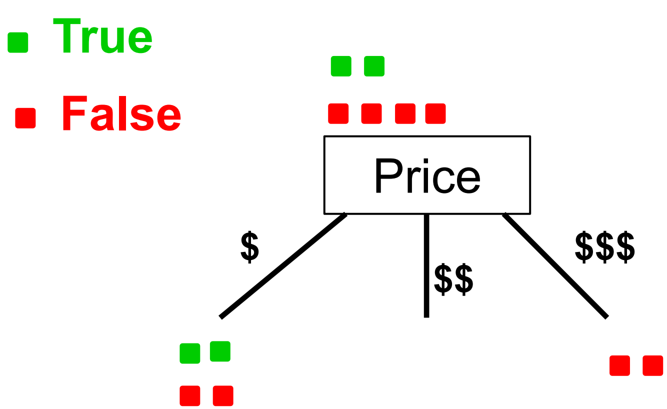
$$\frac{4}{6} * \left(\frac{2}{4} * log_2 \frac{4}{2} + \frac{2}{4} * log_2 \frac{4}{2}\right) + 0 + \frac{2}{6} * (0 + 1 * log_2 1) =$$

$$= \frac{2}{3} * \left(2 * \frac{1}{2} * log_2 2\right) + 0 + 0 = \frac{2}{3} = 0.6666666 \text{ [bit]}$$

Decision tree - Creation

EFOP-3.4.3-16-2016-00009
A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

Information gain of attribute „Price”:



Example	Attributes										Target
	Alt	*Bar*	*Fri*	*Hun*	*Pat*	*Price*	*Rain*	*Res*	*Type*	*Est*	*WillWait*
X_1	T	F	F	T	Some	$$$	F	T	French	0–10	T
X_2	T	F	F	T	Full	$	F	F	Thai	30–60	F
X_3	F	T	F	F	Some	$	F	F	Burger	0–10	T
X_4	T	F	T	T	Full	$	F	F	Thai	10–30	T
X_5	T	F	T	F	Full	$$$	F	T	French	>60	F
X_6	F	T	F	T	Some	$$	T	T	Italian	0–10	T
X_7	F	T	F	F	None	$	T	F	Burger	0–10	F
X_8	F	F	F	T	Some	$$	T	T	Thai	0–10	T
X_9	F	T	T	F	Full	$	T	F	Burger	>60	F
X_{10}	T	T	T	T	Full	$$$	F	T	Italian	10–30	F
X_{11}	F	F	F	F	None	$	F	F	Thai	0–10	F
X_{12}	T	T	T	T	Full	$	F	F	Burger	30–60	T

The **entropy after testing the attribute** is:

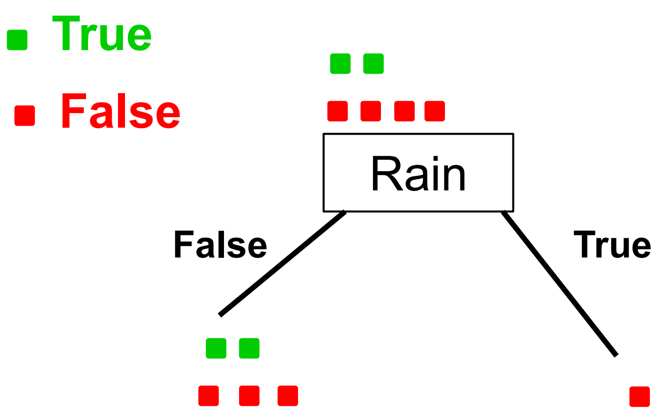
$$= \frac{2}{3} * \left(2 * \frac{1}{2} * log_2 2\right) + 0 + 0 = \frac{2}{3} = 0.6666666 \text{ [bit]}$$

So, the **information gain of attribute „Price” (in the branch *Patrons=Full*) is:**

0.9182958 [bit] - 0.6666666 [bit] = **0.2516292 [bit]**

Decision tree - Creation

Information gain of attribute „Rain”:



Example	Attributes										Target
	Alt	*Bar*	*Fri*	*Hun*	*Pat*	*Price*	*Rain*	*Res*	*Type*	*Est*	*WillWait*
X_1	T	F	F	T	Some	$$$	F	T	French	0–10	T
X_2	T	F	F	T	Full	$	F	F	Thai	30–60	F
X_3	F	T	F	F	Some	$	F	F	Burger	0–10	T
X_4	T	F	T	T	Full	$	F	F	Thai	10–30	T
X_5	T	F	T	F	Full	$$$	F	T	French	>60	F
X_6	F	T	F	T	Some	$$	T	T	Italian	0–10	T
X_7	F	T	F	F	None	$	T	F	Burger	0–10	F
X_8	F	F	F	T	Some	$$	T	T	Thai	0–10	T
X_9	F	T	T	F	Full	$	T	F	Burger	>60	F
X_{10}	T	T	T	T	Full	$$$	F	T	Italian	10–30	F
X_{11}	F	F	F	F	None	$	F	F	Thai	0–10	F
X_{12}	T	T	T	T	Full	$	F	F	Burger	30–60	T

The **entropy after testing the attribute** is:

$$\frac{5}{6}*\left(\frac{2}{5}*log_2\frac{5}{2}+\frac{3}{5}*log_2\frac{5}{3}\right)+\frac{1}{6}*(0+1*log_2 1)=$$

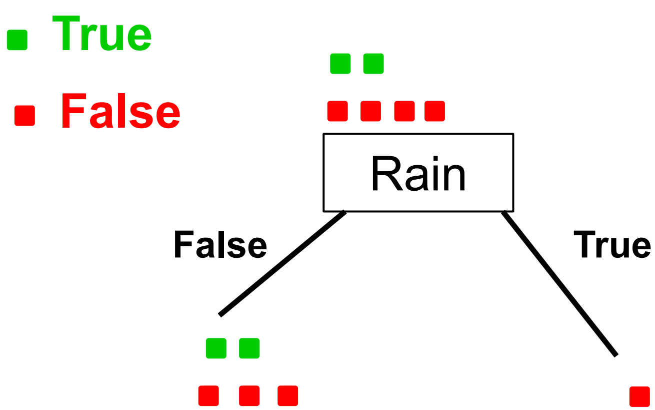
$$=\frac{5}{6}*\left(\frac{2}{5}*\frac{log_{10}\frac{5}{2}}{log_{10}2}+\frac{3}{5}*\frac{log_{10}\frac{5}{3}}{log_{10}2}\right)+0\approx\frac{5}{6}*\left(\frac{2}{5}*1.321928+\frac{3}{5}*0.736966\right)=0.8091257\ \text{[bit]}$$

Decision tree - Creation

EFOP-3.4.3-16-2016-00009

A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

Information gain of attribute „Rain”:



Example	Attributes										Target
	Alt	*Bar*	*Fri*	*Hun*	*Pat*	*Price*	*Rain*	*Res*	*Type*	*Est*	*WillWait*
X_1	T	F	F	T	Some	$$$	F	T	French	0–10	T
X_2	T	F	F	T	Full	$	F	F	Thai	30–60	F
X_3	F	T	F	F	Some	$	F	F	Burger	0–10	T
X_4	T	F	T	T	Full	$	F	F	Thai	10–30	T
X_5	T	F	T	F	Full	$$$	F	T	French	>60	F
X_6	F	T	F	T	Some	$$	T	T	Italian	0–10	T
X_7	F	T	F	F	None	$	T	F	Burger	0–10	F
X_8	F	F	F	T	Some	$$	T	T	Thai	0–10	T
X_9	F	T	T	F	Full	$	T	F	Burger	>60	F
X_{10}	T	T	T	T	Full	$$$	F	T	Italian	10–30	F
X_{11}	F	F	F	F	None	$	F	F	Thai	0–10	F
X_{12}	T	T	T	T	Full	$	F	F	Burger	30–60	T

The **entropy after testing the attribute** is:

$$= \frac{5}{6} * \left(\frac{2}{5} * \frac{log_{10}\frac{5}{2}}{log_{10}2} + \frac{3}{5} * \frac{log_{10}\frac{5}{3}}{log_{10}2} \right) + 0 \approx \frac{5}{6} * \left(\frac{2}{5} * 1.321928 + \frac{3}{5} * 0.736966 \right) = 0.8091257 \text{ [bit]}$$

So, the **information gain of attribute „Rain” (in the branch *Patrons=Full*) is:**

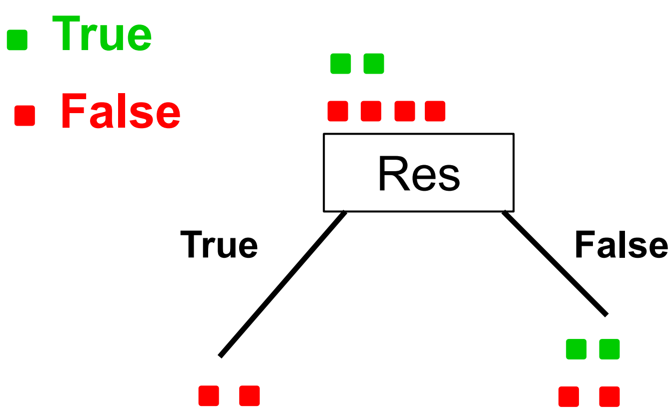
0.9182958 [bit] - 0.8091257 [bit] = **0.1091701 [bit]**

Decision tree - Creation

EFOP-3.4.3-16-2016-00009

A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

Information gain of attribute „Res”:



Example	Attributes										Target
	Alt	*Bar*	*Fri*	*Hun*	*Pat*	*Price*	*Rain*	*Res*	*Type*	*Est*	*WillWait*
X_1	T	F	F	T	Some	$$$	F	T	French	0–10	T
X_2	T	F	F	T	Full	$	F	F	Thai	30–60	F
X_3	F	T	F	F	Some	$	F	F	Burger	0–10	T
X_4	T	F	T	T	Full	$	F	F	Thai	10–30	T
X_5	T	F	T	F	Full	$$$	F	T	French	>60	F
X_6	F	T	F	T	Some	$$	T	T	Italian	0–10	T
X_7	F	T	F	F	None	$	T	F	Burger	0–10	F
X_8	F	F	F	T	Some	$$	T	T	Thai	0–10	T
X_9	F	T	T	F	Full	$	T	F	Burger	>60	F
X_{10}	T	T	T	T	Full	$$$	F	T	Italian	10–30	F
X_{11}	F	F	F	F	None	$	F	F	Thai	0–10	F
X_{12}	T	T	T	T	Full	$	F	F	Burger	30–60	T

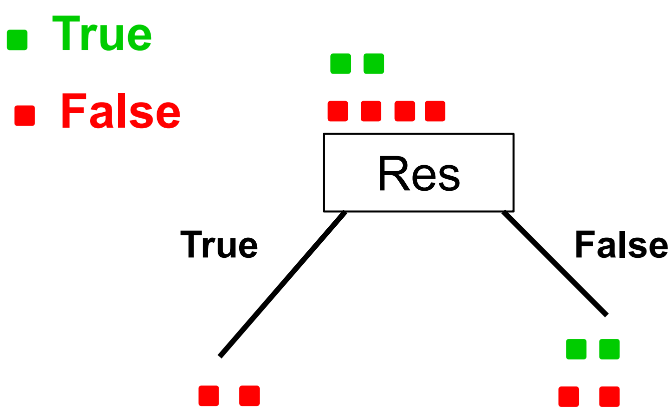
The **entropy after testing the attribute** is:

$$\frac{2}{6}*(0+1*log_2 1)+\frac{4}{6}*\left(\frac{2}{4}*log_2\frac{4}{2}+\frac{2}{4}*log_2\frac{4}{2}\right)=$$

$$=0+\frac{2}{3}*\left(2*\frac{1}{2}*log_2 2\right)=\frac{2}{3}=0.6666666\ \text{[bit]}$$

Decision tree - Creation

Information gain of attribute „Res”:



Example	Attributes										Target
	Alt	*Bar*	*Fri*	*Hun*	*Pat*	*Price*	*Rain*	*Res*	*Type*	*Est*	*WillWait*
X_1	T	F	F	T	Some	$$$	F	T	French	0–10	T
X_2	T	F	F	T	Full	$	F	F	Thai	30–60	F
X_3	F	T	F	F	Some	$	F	F	Burger	0–10	T
X_4	T	F	T	T	Full	$	F	F	Thai	10–30	T
X_5	T	F	T	F	Full	$$$	F	T	French	>60	F
X_6	F	T	F	T	Some	$$	T	T	Italian	0–10	T
X_7	F	T	F	F	None	$	T	F	Burger	0–10	F
X_8	F	F	F	T	Some	$$	T	T	Thai	0–10	T
X_9	F	T	T	F	Full	$	T	F	Burger	>60	F
X_{10}	T	T	T	T	Full	$$$	F	T	Italian	10–30	F
X_{11}	F	F	F	F	None	$	F	F	Thai	0–10	F
X_{12}	T	T	T	T	Full	$	F	F	Burger	30–60	T

The **entropy after testing the attribute** is:

$$= 0 + \frac{2}{3} * \left(2 * \frac{1}{2} * log_2 2\right) = \frac{2}{3} = 0.6666666 \text{ [bit]}$$

So, the **information gain of attribute „Res” (in the branch *Patrons=Full*) is**:

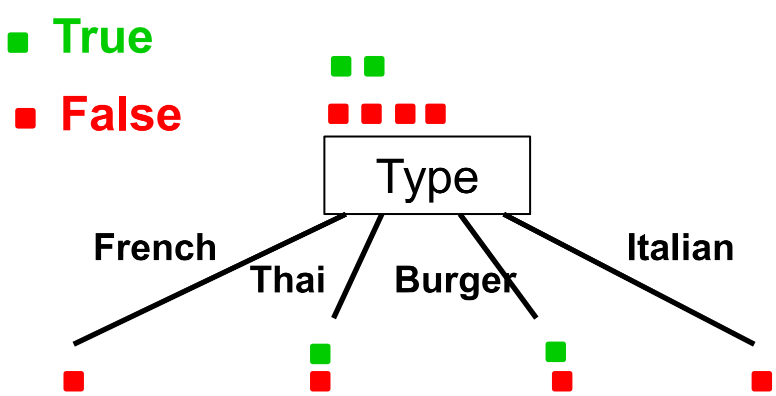
0.9182958 [bit] - 0.6666666 [bit] = **0.2516292 [bit]**

Decision tree - Creation

EFOP-3.4.3-16-2016-00009

A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

Information gain of attribute „Type”:



Example	Attributes										Target
	Alt	*Bar*	*Fri*	*Hun*	*Pat*	*Price*	*Rain*	*Res*	*Type*	*Est*	*WillWait*
X_1	T	F	F	T	Some	$$$	F	T	French	0–10	T
X_2	T	F	F	T	Full	$	F	F	Thai	30–60	F
X_3	F	T	F	F	Some	$	F	F	Burger	0–10	T
X_4	T	F	T	T	Full	$	F	F	Thai	10–30	T
X_5	T	F	T	F	Full	$$$	F	T	French	>60	F
X_6	F	T	F	T	Some	$$	T	T	Italian	0–10	T
X_7	F	T	F	F	None	$	T	F	Burger	0–10	F
X_8	F	F	F	T	Some	$$	T	T	Thai	0–10	T
X_9	F	T	T	F	Full	$	T	F	Burger	>60	F
X_{10}	T	T	T	T	Full	$$$	F	T	Italian	10–30	F
X_{11}	F	F	F	F	None	$	F	F	Thai	0–10	F
X_{12}	T	T	T	T	Full	$	F	F	Burger	30–60	T

The **entropy after testing the attribute** is:

$$\frac{1}{6} * (0 + 1 * log_2 1) + \frac{2}{6} * \left(\frac{1}{2} * log_2 2 + \frac{1}{2} * log_2 2\right) + \frac{2}{6} * \left(\frac{1}{2} * log_2 2 + \frac{1}{2} * log_2 2\right) + \frac{1}{6} * (0 + 1 * log_2 1) =$$

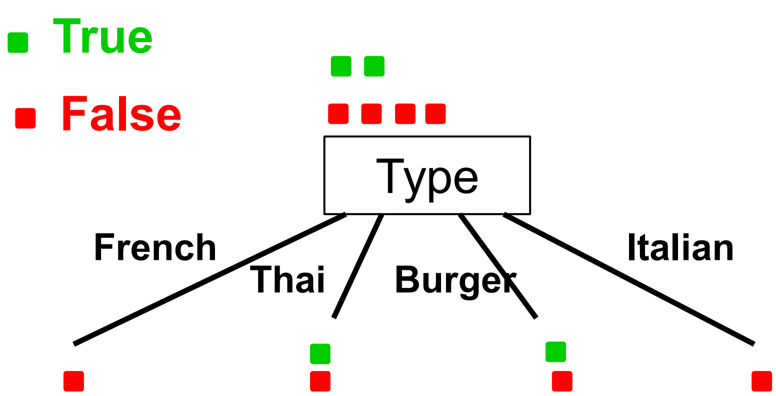
$$= 0 + 2 * \frac{1}{3} * \left(2 * \frac{1}{2} * log_2 2\right) + 0 = \frac{2}{3} = 0.6666666 \text{ [bit]}$$

Decision tree - Creation

EFOP-3.4.3-16-2016-00009

A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

Information gain of attribute „Type”:



Example	Attributes										Target
	Alt	*Bar*	*Fri*	*Hun*	*Pat*	*Price*	*Rain*	*Res*	*Type*	*Est*	*WillWait*
X_1	*T*	*F*	*F*	*T*	*Some*	*$$$*	*F*	*T*	*French*	*0–10*	*T*
X_2	*T*	*F*	*F*	*T*	*Full*	*$*	*F*	*F*	*Thai*	*30–60*	*F*
X_3	*F*	*T*	*F*	*F*	*Some*	*$*	*F*	*F*	*Burger*	*0–10*	*T*
X_4	*T*	*F*	*T*	*T*	*Full*	*$*	*F*	*F*	*Thai*	*10–30*	*T*
X_5	*T*	*F*	*T*	*F*	*Full*	*$$$*	*F*	*T*	*French*	*>60*	*F*
X_6	*F*	*T*	*F*	*T*	*Some*	*$$*	*T*	*T*	*Italian*	*0–10*	*T*
X_7	*F*	*T*	*F*	*F*	*None*	*$*	*T*	*F*	*Burger*	*0–10*	*F*
X_8	*F*	*F*	*F*	*T*	*Some*	*$$*	*T*	*T*	*Thai*	*0–10*	*T*
X_9	*F*	*T*	*T*	*F*	*Full*	*$*	*T*	*F*	*Burger*	*>60*	*F*
X_{10}	*T*	*T*	*T*	*T*	*Full*	*$$$*	*F*	*T*	*Italian*	*10–30*	*F*
X_{11}	*F*	*F*	*F*	*F*	*None*	*$*	*F*	*F*	*Thai*	*0–10*	*F*
X_{12}	*T*	*T*	*T*	*T*	*Full*	*$*	*F*	*F*	*Burger*	*30–60*	*T*

The **entropy after testing the attribute** is:

$$= 0 + 2 * \frac{1}{3} * \left(2 * \frac{1}{2} * log_2 2\right) + 0 = \frac{2}{3} = 0.6666666 \text{ [bit]}$$

So, the **information gain of attribute „Type” (in the branch *Patrons=Full*) is:**

0.9182958 [bit] - 0.6666666 [bit] = **0.2516292 [bit]**

Decision tree - Creation

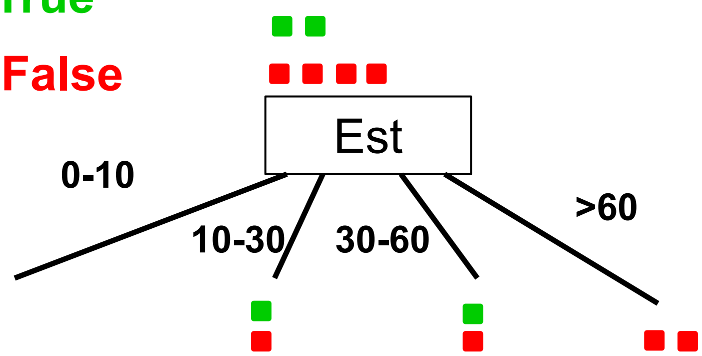
EFOP-3.4.3-16-2016-00009

A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

Information gain of attribute „Est”:

True

False



Example	Attributes										Target
	Alt	*Bar*	*Fri*	*Hun*	*Pat*	*Price*	*Rain*	*Res*	*Type*	*Est*	*WillWait*
X_1	*T*	*F*	*F*	*T*	*Some*	*$$$*	*F*	*T*	*French*	*0–10*	*T*
X_2	*T*	*F*	*F*	*T*	*Full*	*$*	*F*	*F*	*Thai*	*30–60*	*F*
X_3	*F*	*T*	*F*	*F*	*Some*	*$*	*F*	*F*	*Burger*	*0–10*	*T*
X_4	*T*	*F*	*T*	*T*	*Full*	*$*	*F*	*F*	*Thai*	*10–30*	*T*
X_5	*T*	*F*	*T*	*F*	*Full*	*$$$*	*F*	*T*	*French*	*>60*	*F*
X_6	*F*	*T*	*F*	*T*	*Some*	*$$*	*T*	*T*	*Italian*	*0–10*	*T*
X_7	*F*	*T*	*F*	*F*	*None*	*$*	*T*	*F*	*Burger*	*0–10*	*F*
X_8	*F*	*F*	*F*	*T*	*Some*	*$$*	*T*	*T*	*Thai*	*0–10*	*T*
X_9	*F*	*T*	*T*	*F*	*Full*	*$*	*T*	*F*	*Burger*	*>60*	*F*
X_{10}	*T*	*T*	*T*	*T*	*Full*	*$$$*	*F*	*T*	*Italian*	*10–30*	*F*
X_{11}	*F*	*F*	*F*	*F*	*None*	*$*	*F*	*F*	*Thai*	*0–10*	*F*
X_{12}	*T*	*T*	*T*	*T*	*Full*	*$*	*F*	*F*	*Burger*	*30–60*	*T*

The **entropy after testing the attribute** is:

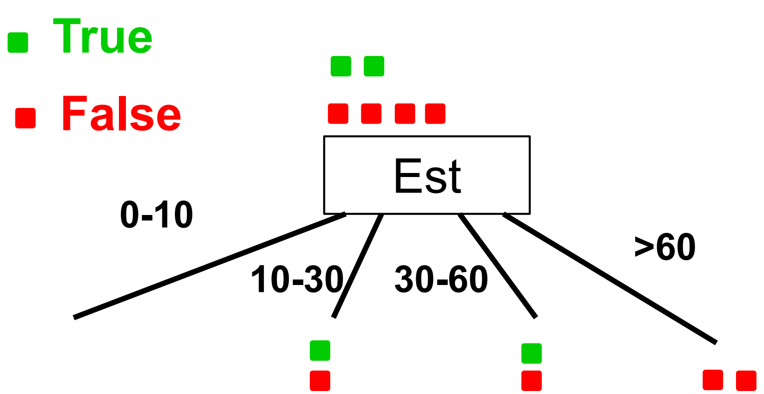
$$0 + \frac{2}{6} * \left(\frac{1}{2} * log_2 2 + \frac{1}{2} * log_2 2\right) + \frac{2}{6} * \left(\frac{1}{2} * log_2 2 + \frac{1}{2} * log_2 2\right) + \frac{2}{6} * (0 + 1 * log_2 1) =$$

$$= 0 + 2 * \frac{1}{3} * \left(2 * \frac{1}{2} * log_2 2\right) + 0 = \frac{2}{3} = 0.6666666 \ \text{[bit]}$$

Decision tree - Creation

EFOP-3.4.3-16-2016-00009
A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

Information gain of attribute „Est”:



Example	Attributes										Target
	Alt	*Bar*	*Fri*	*Hun*	*Pat*	*Price*	*Rain*	*Res*	*Type*	*Est*	*WillWait*
X_1	T	F	F	T	Some	$$$	F	T	French	0–10	T
X_2	T	F	F	T	Full	$	F	F	Thai	30–60	F
X_3	F	T	F	F	Some	$	F	F	Burger	0–10	T
X_4	T	F	T	T	Full	$	F	F	Thai	10–30	T
X_5	T	F	T	F	Full	$$$	F	T	French	>60	F
X_6	F	T	F	T	Some	$$	T	T	Italian	0–10	T
X_7	F	T	F	F	None	$	T	F	Burger	0–10	F
X_8	F	F	F	T	Some	$$	T	T	Thai	0–10	T
X_9	F	T	T	F	Full	$	T	F	Burger	>60	F
X_{10}	T	T	T	T	Full	$$$	F	T	Italian	10–30	F
X_{11}	F	F	F	F	None	$	F	F	Thai	0–10	F
X_{12}	T	T	T	T	Full	$	F	F	Burger	30–60	T

The **entropy after testing the attribute** is:

$$= 0 + 2 * \frac{1}{3} * \left(2 * \frac{1}{2} * log_2 2\right) + 0 = \frac{2}{3} = 0.6666666 \text{ [bit]}$$

So, the **information gain of attribute „Est” (in the branch *Patrons=Full*) is:**

0.9182958 [bit] - 0.6666666 [bit] = **0.2516292 [bit]**

Decision tree - Creation

The **information gain** values we got **for the attributes (in the branch *Patrons=Full*)** are:

Attribute	Information gain of the attribute
Alt	0.1091701 [bit]
Bar	0 [bit]
Fri	0.1091701 [bit]
Hun	0.2516292 [bit]
Price	0.2516292 [bit]
Rain	0.1091701 [bit]
Res	0.2516292 [bit]
Type	0.2516292 [bit]
Est	0.2516292 [bit]

Decision tree - Creation

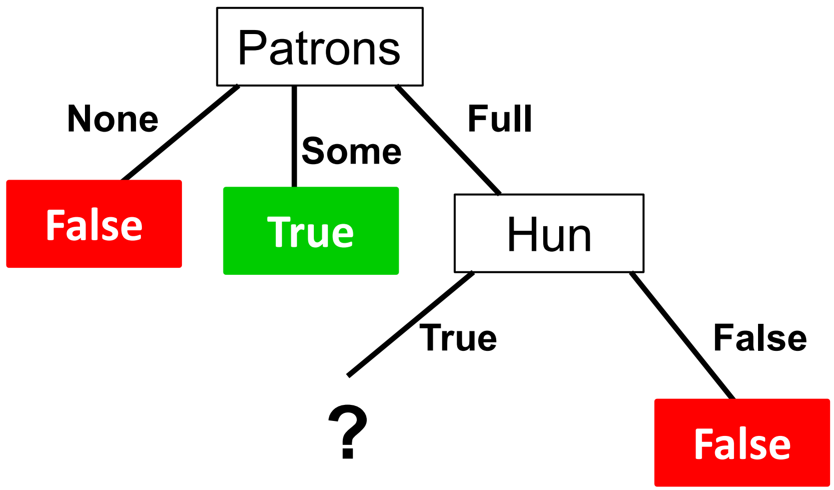
The **information gain** values we got **for the attributes (in the branch *Patrons=Full*)** are:

Attribute	Information gain of the attribute
Alt	0.1091701 [bit]
Bar	0 [bit]
Fri	0.1091701 [bit]
Hun	**0.2516292 [bit]**
Price	**0.2516292 [bit]**
Rain	0.1091701 [bit]
Res	**0.2516292 [bit]**
Type	**0.2516292 [bit]**
Est	**0.2516292 [bit]**

There are 5 candidates for the attribute with the highest information gain in this iteration: „**Hun**”, „**Price**”, „**Res**”, „**Type**” and „**Est**”. So, any of them can **be the next node of the decision tree** (in the branch *Patrons=Full*).

Decision tree - Creation

After determining the second level of the decision tree (selecting attribute „Hun" from the 5 candidates with equal information gain values), we have:

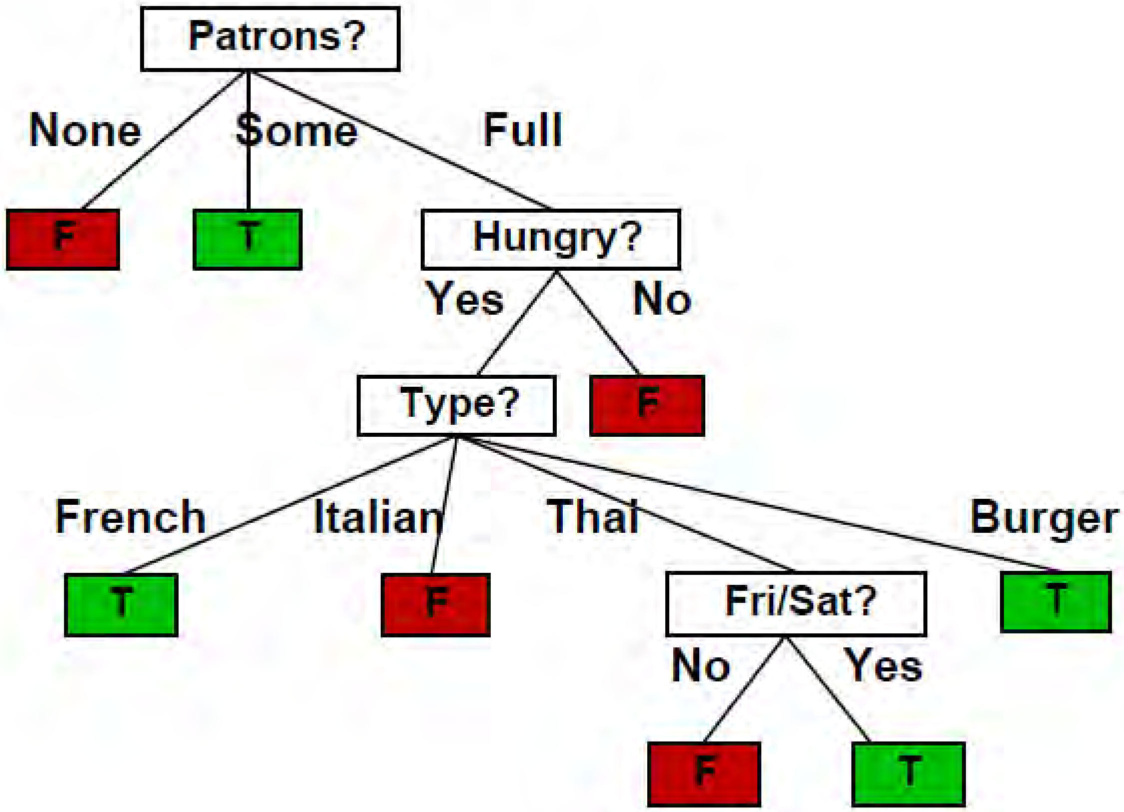


The creation of the decision tree **continues** this way.

Decision tree - Creation

EFOP-3.4.3-16-2016-00009
A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

Finally, **a possible decision tree** – which was built based on the 12 given examples – can look like this:

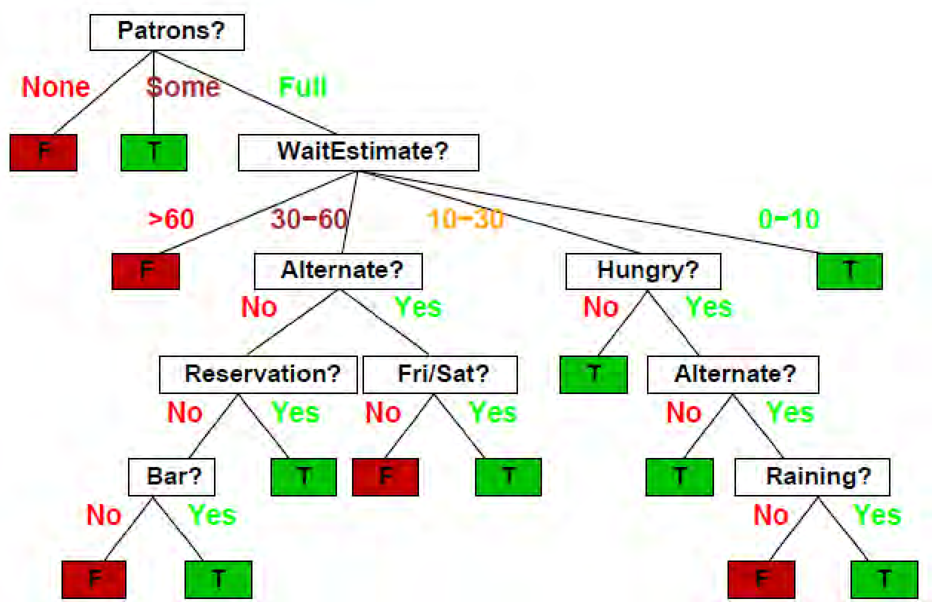


Decision tree - Creation

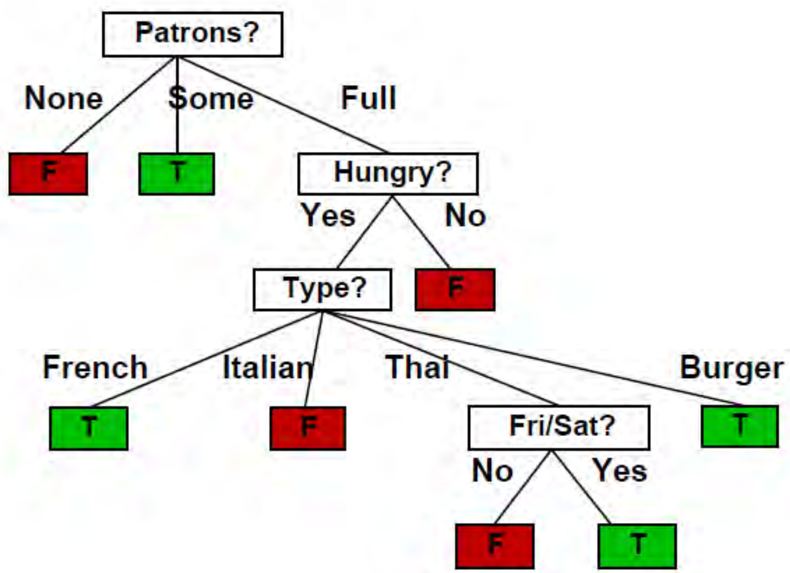
EFOP-3.4.3-16-2016-00009
A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

Let us compare the original tree (from which the 12 examples were created) and the generated decision tree!

The original decision tree



The generated decision tree



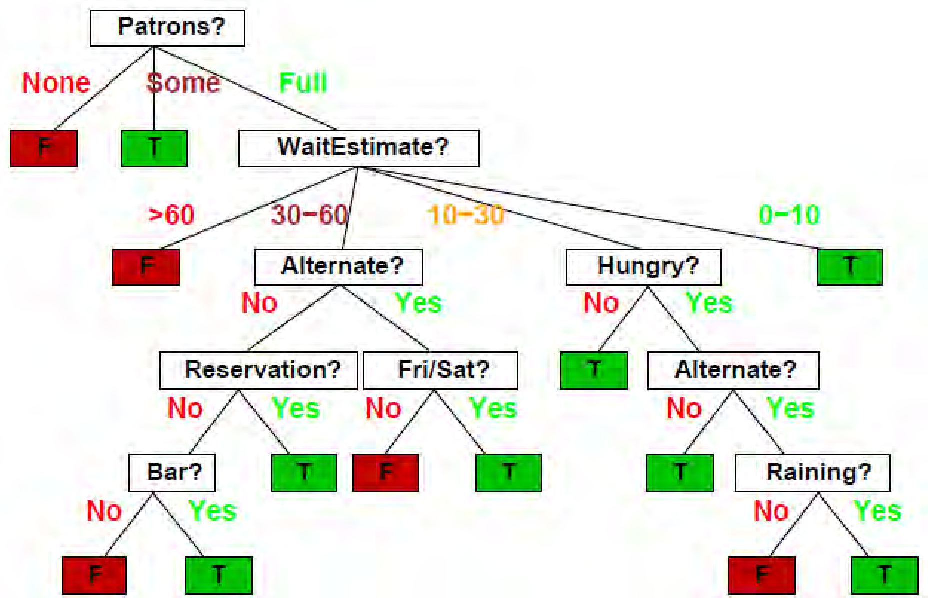
Since the Decision Tree Learning algorithm had **only the 12 examples as input** (and not the original tree), **the two trees differ** from each other.

Decision tree - Creation

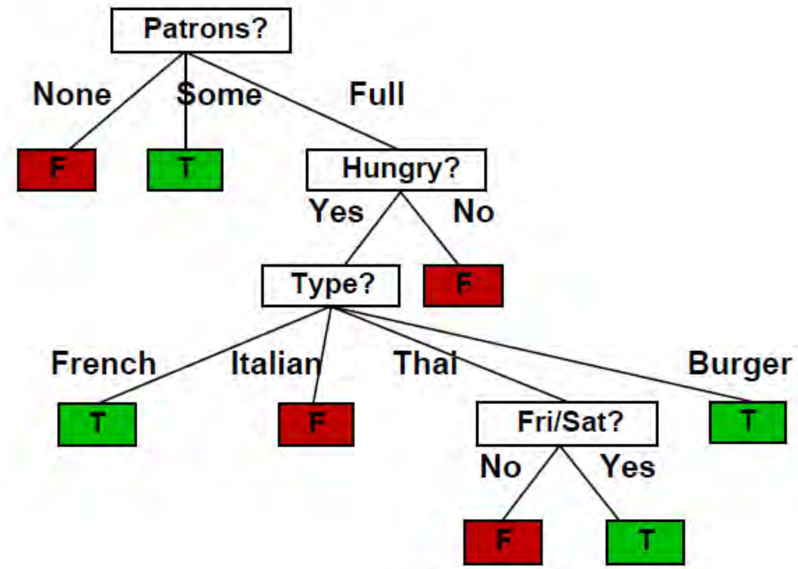
EFOP-3.4.3-16-2016-00009
A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

Let us compare the original tree (from which the 12 examples were created) and the generated decision tree!

The original decision tree



The generated decision tree



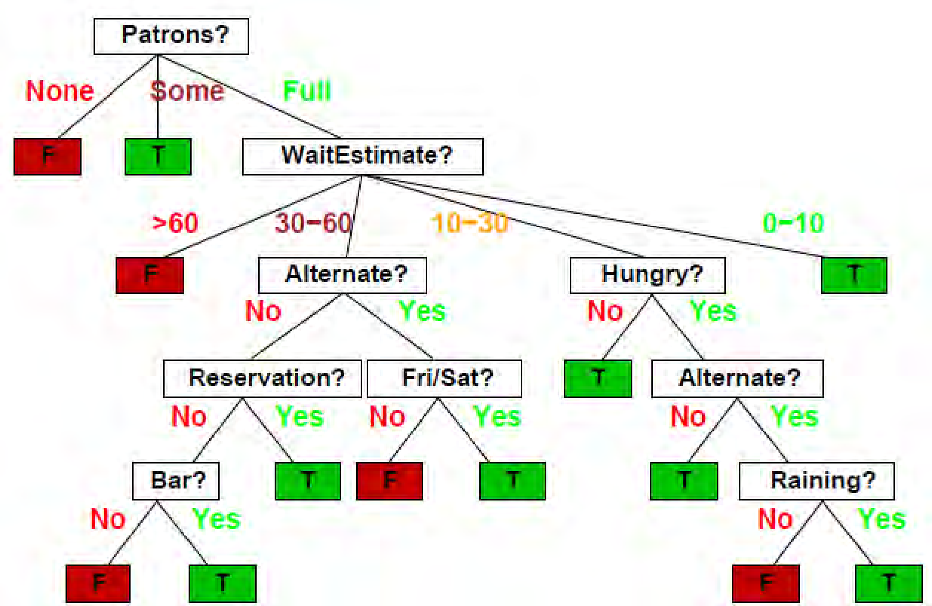
The generated tree does not need to test attribute „Raining" or attribute „Reservation" (the examples can be classified without testing them).

Decision tree - Creation

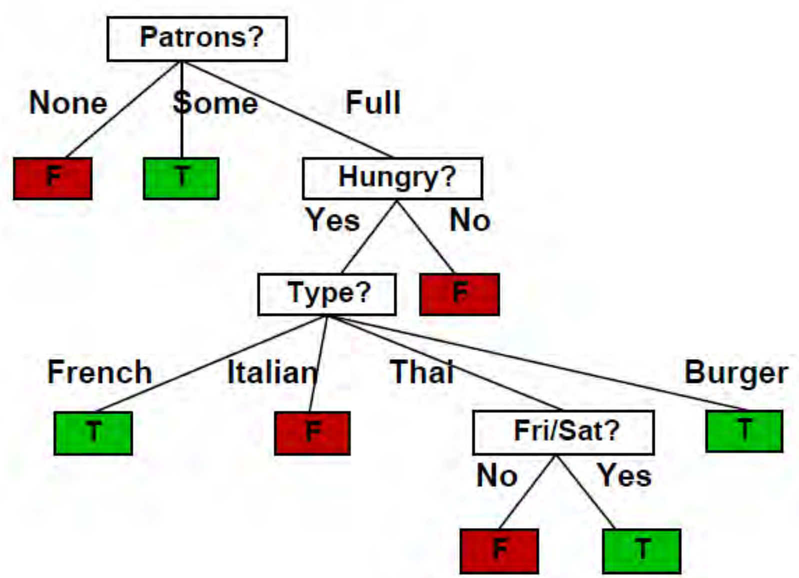
EFOP-3.4.3-16-2016-00009
A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

Let us compare the original tree (from which the 12 examples were created) and the generated decision tree!

The original decision tree



The generated decision tree



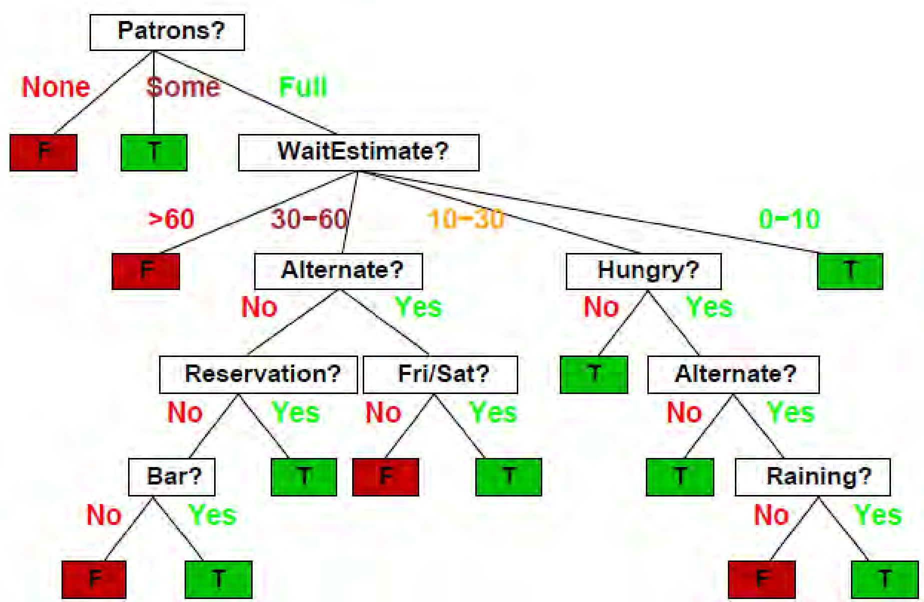
The generated tree discovered new rule: wait for thai foods on the weekends.

Decision tree - Creation

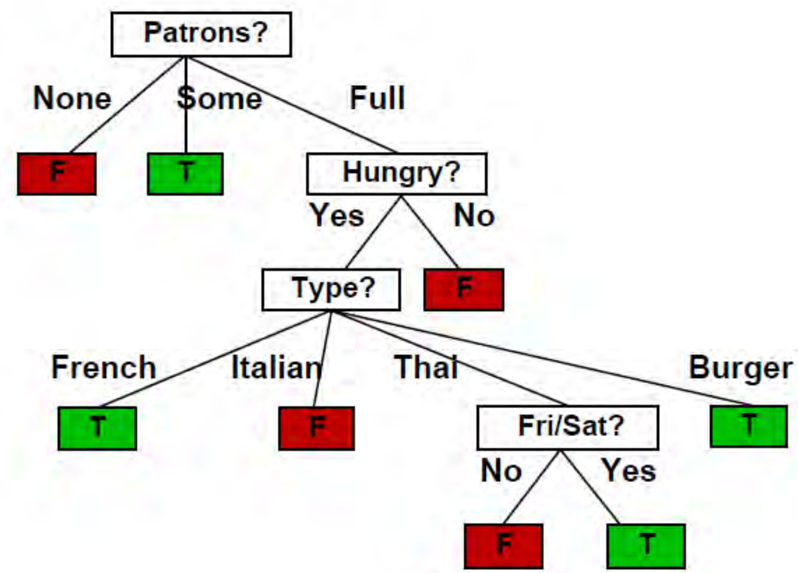
EFOP-3.4.3-16-2016-00009
A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

Let us compare the original tree (from which the 12 examples were created) and the generated decision tree!

The original decision tree



The generated decision tree

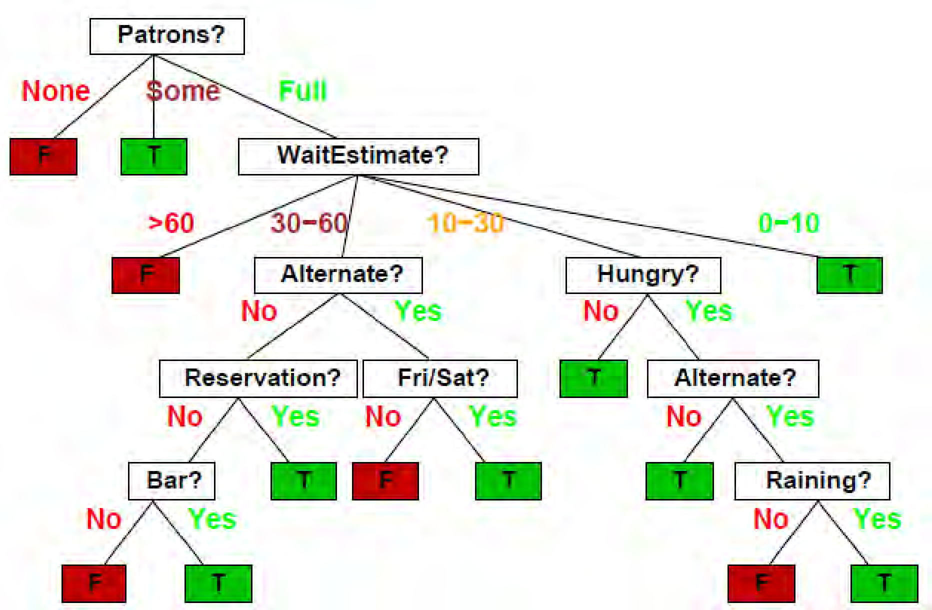


The generated tree **fails**, e.g, if the user is not hungry; moreover it **does not handle the case** when the restaurant is full and the estimated waiting time is between 0 and 10 minutes.

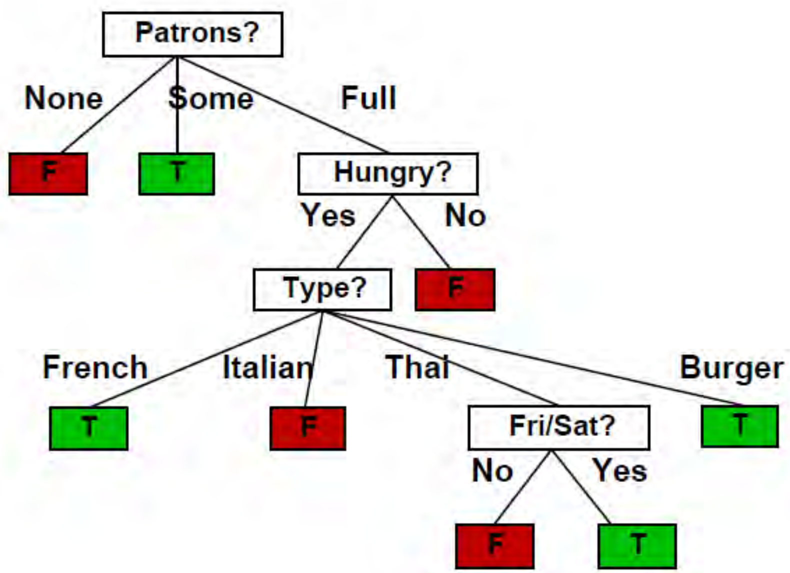
Decision tree - Creation

Let us compare the original tree (from which the 12 examples were created) and the generated decision tree!

The original decision tree



The generated decision tree



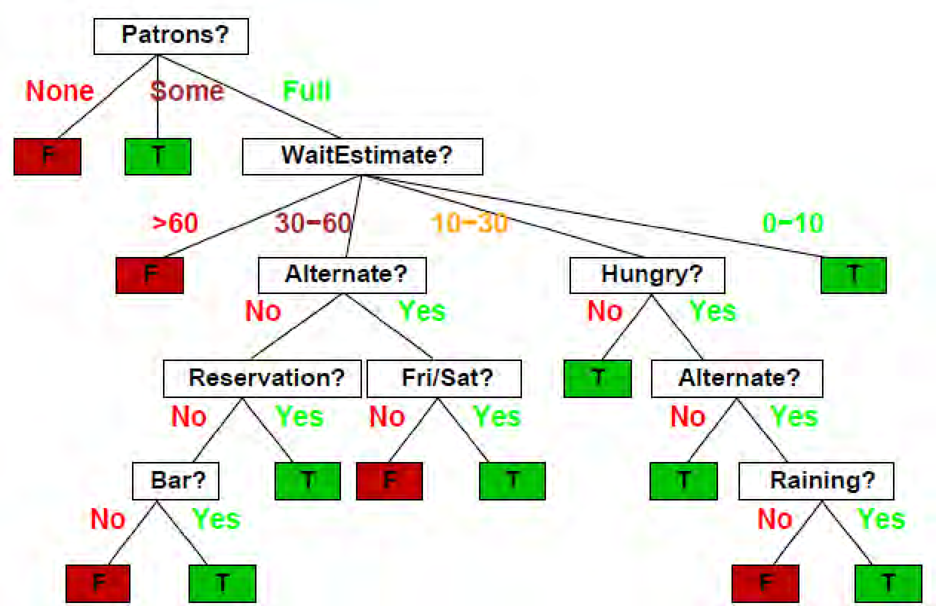
The generated tree is consistent and much simpler than the original one (and – of course – than the tree that could be built from the truth table).

Decision tree - Creation

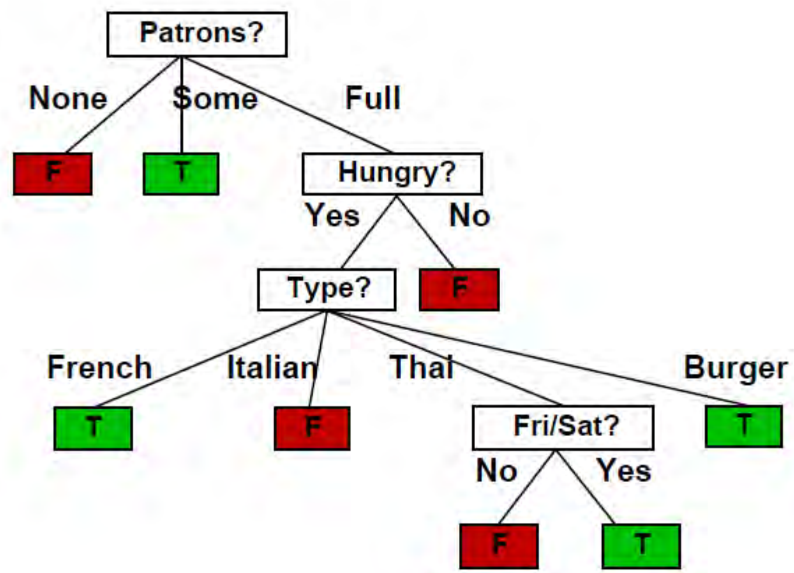
EFOP-3.4.3-16-2016-00009
A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

Let us compare the original tree (from which the 12 examples were created) and the generated decision tree!

The original decision tree



The generated decision tree



Of course, the two trees would be **much more similar** if the training set contains **more examples**.

THE PERFORMANCE OF THE LEARNING PROCESS

How good the created decision tree is?

The performance of the learning process

EFOP-3.4.3-16-2016-00009
A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

The learning algorithm performs well if it offers **good hypotheses for the new inputs** that it never met before.

It **requires examples that were not applied in the training process** and can be used only for testing.

This set of the examples is called **test set**.

So, the original set of the examples has to be divided into two groups:

training set

test set

EFOP-3.4.3-16-2016-00009

A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

How to divide the initial example set into the two groups?

The examples in the two groups have to be **independent** **(similarly to the case of an exam, where students should not know the exact questions before the exam, since we want to get a useful and real feedback from the exam about the students' knowledge)**

The division may happen **randomly** **(taking into account the independence of the examples in the two groups).**

However, **after improving the algorithm based on the results of the tests**, some features of the examples of the test set will be built in the training set. It means that the **test set will be infected**: it is not independent from the training set anymore.

⬇

After each such improvement of the algorithm a **new example set should be used for testing** **(either by creation of new examples – that are independent from the other groups –, or by dividing the original test set into more than one test sets).**

The performance of the learning process

The testing process

After creating the hypothesis from the examples of the training set, it should be tested as follows:

Loop begins for all elements of the test set

- use the input of the example for **getting the result by the decision tree**
- **compare** the output got by the decision tree **with the output of the example**

Loop ends

By performing this loop, we get **how percentage of the test examples are classified right**.

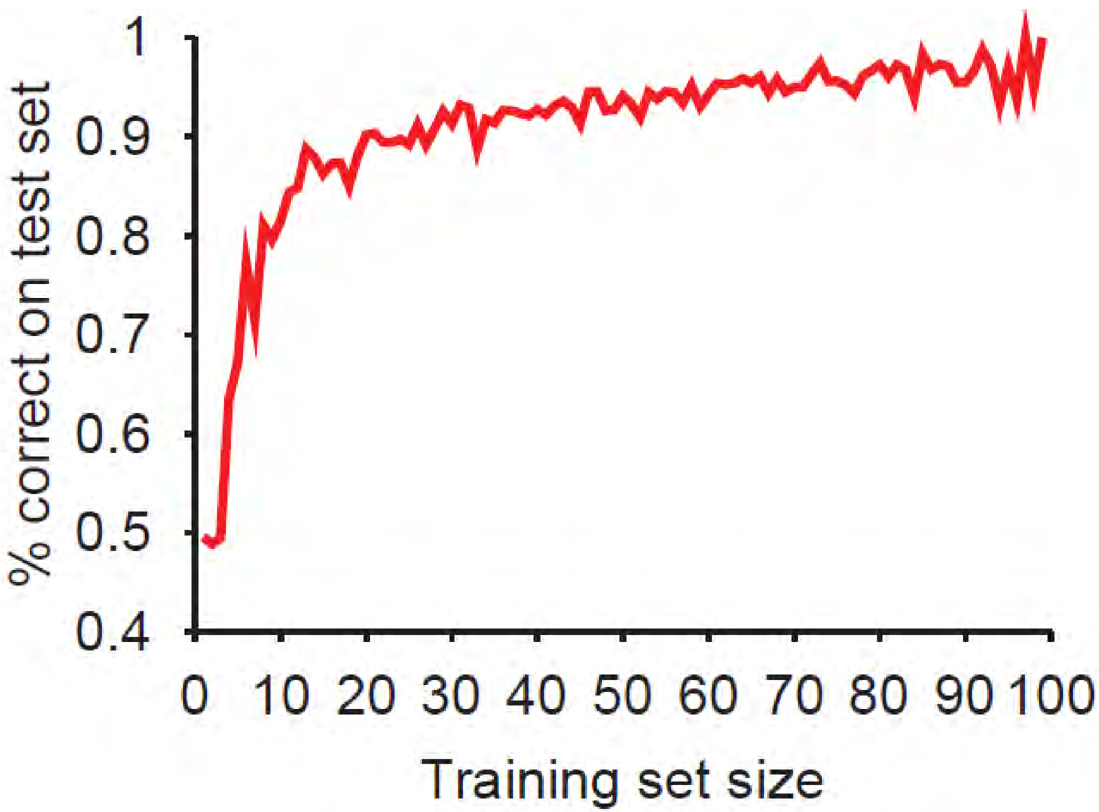
After selecting randomly **different sized training sets** from the initial set of the examples and **repeating the loop above** to test the resulted decision trees, the performance can be presented on a figure.

The performance of the learning process

EFOP-3.4.3-16-2016-00009
A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

The learning curve

The resulted figure shows a curve: the percentage of the right classified examples as the function of the size of the training set.



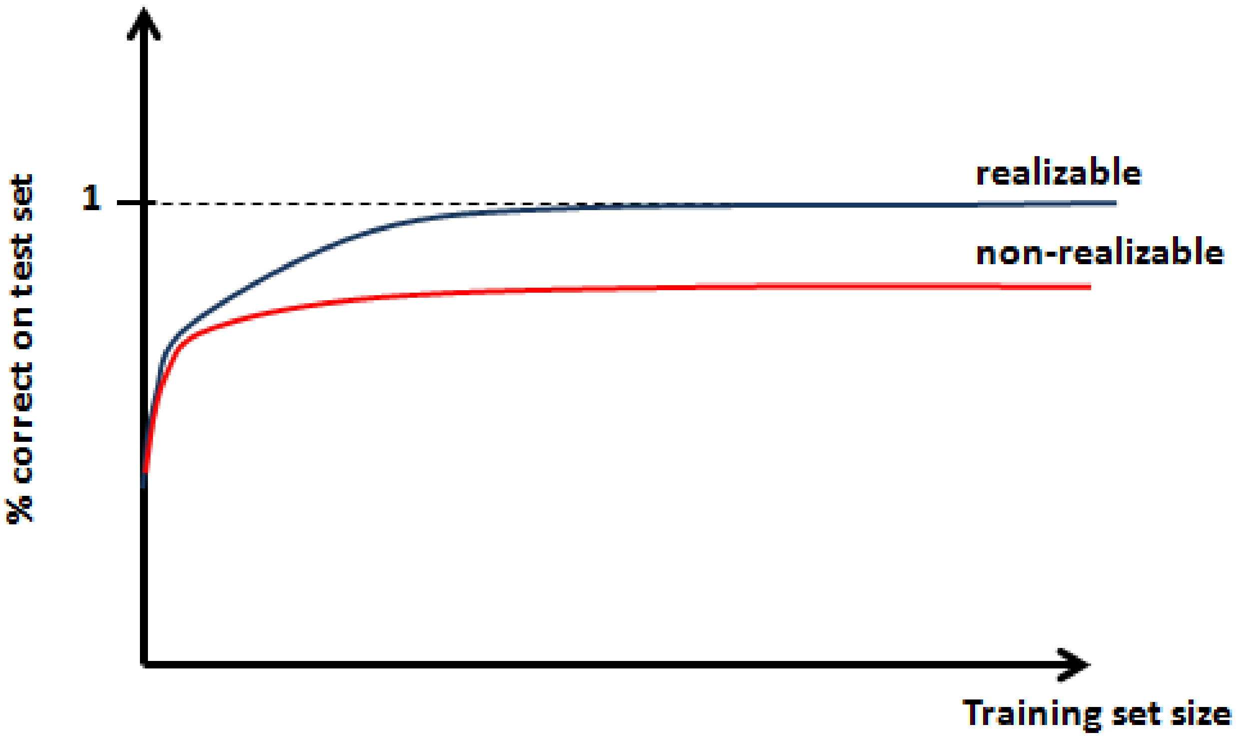
This curve expresses the **general prediction capability** of the generated decision tree.

„**Happy graph**”, since the prediction capability improves as the training set size grows.

The performance of the learning process

The **shape** of the learning curve **lets us to conclude** some facts about the success of the training (and its reason).

Case 1:



The reasons of getting „non-realizable" curve can be:

- **Missing attribute(s)**

- **Restricted hypothesis class**

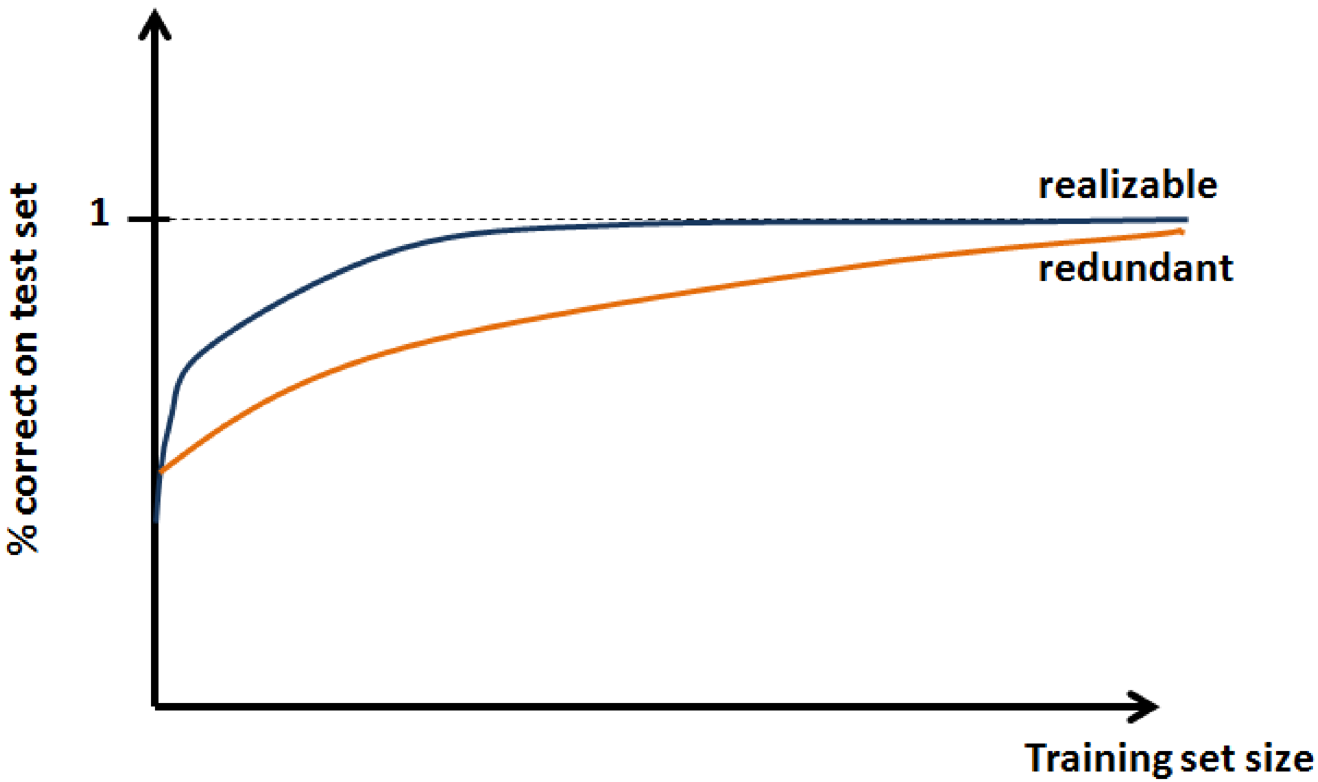
The performance of the learning process

EFOP-3.4.3-16-2016-00009

A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

The **shape** of the learning curve **lets us to conclude** some facts about the success of the training (and its reason).

Case 2:



The reason of getting „redundant” curve can be:

- Irrelevant attributes, redundancy

The performance of the learning process

EFOP-3.4.3-16-2016-00009
A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

Possible problems related to the learning process

1. **Noise**

Its reason:
there are multiple training examples with the **same input** values but with **different classifications**

The problem it causes is:
the created decision tree **can not be consistent**

Possible solutions:
- classification based on **majority** **(it is applied, e.g., in case of the creation of a decision tree that represents a logical function)**

- determination of **probabilities based on the relative frequency** of the classifications **(for agents that work on the basis of decision theory)**

The performance of the learning process

EFOP-3.4.3-16-2016-00009
A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

Possible problems related to the learning process

2. Overfitting

Its reason:
the algorithm uses **irrelevant attributes** for the classification **(it can happen especially, if the space of hypotheses is big – if the freedom is high, it is easy to find meaningless regularity)**

The problem it causes is:
the algorithm finds and learns **rules that do not exist** in reality

Examples:
(1) **flipping a coin**, and input attributes are: **the day of the year of the flip, the hour of the flip**, etc. **(while there do not exist two examples with the same input, man can think, e.g., that the result of a coin flip on 5th Jan @11pm is always „head")**

(2) **choosing a restaurant**, and **the name of the restaurant** is one of the input attributes. **(in this case this attribute separates the best the examples; so it is the most informative attribute)**

The performance of the learning process

EFOP-3.4.3-16-2016-00009
A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

Possible problems related to the learning process

2. Overfitting (cont.)

Possible solutions:

- the usage of **gain ratio instead of information gain**

(gain ratio modifies the value of information gain by calculating its ratio with the cardinality of the relevant example set)

⬇

this way, an attribute that **separates the example set into one element subsets** will not surely be the most informative attribute

?

Possible problems related to the learning process

3. Missing data

Its reason:

some attribute value is missing (e.g., unsuccessful measurement happened)

The problems it causes are:

- **how to classify** an object which misses attribute value(s)?

- **how to calculate information gain** if an example of the training set misses attribute value(s)?

The performance of the learning process

EFOP-3.4.3-16-2016-00009

A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

Possible problems related to the learning process

3. Missing data (cont.)

Possible solutions:

- for applying an object with missing attribute value(s) while **building the decision tree**:

the **attribute values that are analyzed after the testing of the missing attribute are weighted based on the frequency** of their values in all the examples

- **for classifying** an object with missing attribute value(s):

suppose that the object has **every possible attribute value** for that attribute, but **weighting them by the frequency** of the prior examples that reach this node. Then **follow all the outgoing branches**, multiplying the weights along the path.

The performance of the learning process

EFOP-3.4.3-16-2016-00009

A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

Possible problems related to the learning process

4. Continuous-valued attributes

Its reason:

several attributes **(of physical of economic processes, for example)** may have values from a **huge cardinality – or infinite – value set** (e.g., weight, height)

The problem it causes is:

these attributes **do not fit to the decision tree learning process**

Possible solution:

Let's **discretize** the attribute!

In this process discrete values are assigned to the **appropriate intervals** (who have not sufficiently equal length)

/e.g., cheap, middle cost, expensive/

The determination of the intervals may happen in **ad hoc** way or after some **pre-processing of the raw data**.

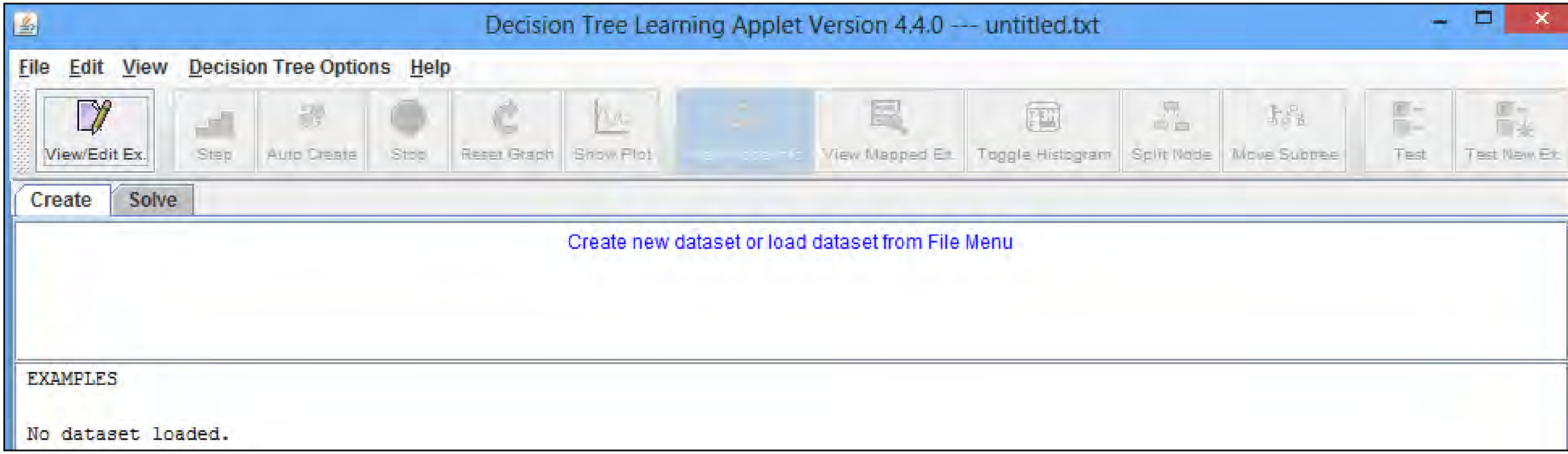
DEMONSTRATION

A decision tree application

*this Chapter is based on the decision tree handler software that is freely available at the URL:

http://www.aispace.org/

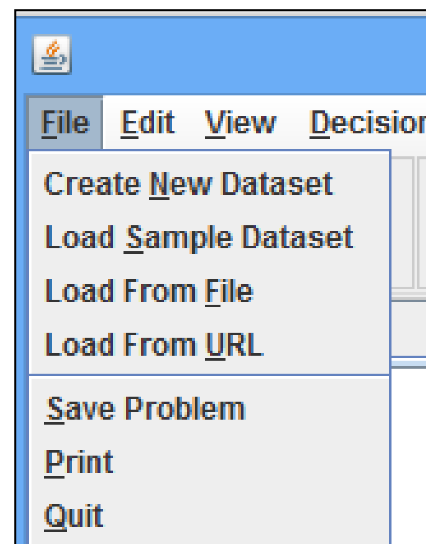
Demonstration



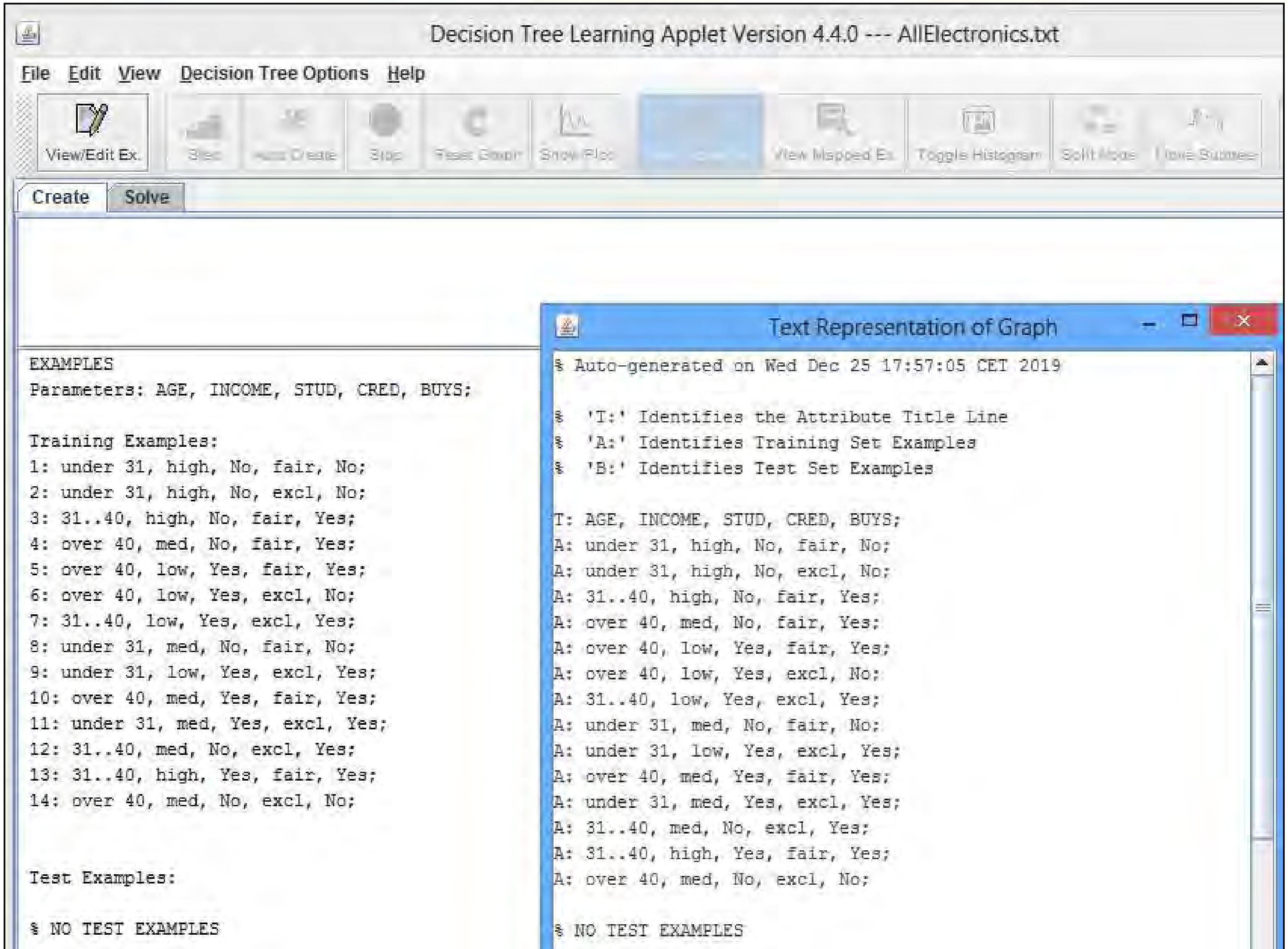
The **two main parts** of the software are:

- „**Create**” tab: Helps managing the example set. It is possible both to create a new dataset or load it from a selected source.

- „**Solve**” tab: Builds a decision tree from the training set and offers testing capabilities.



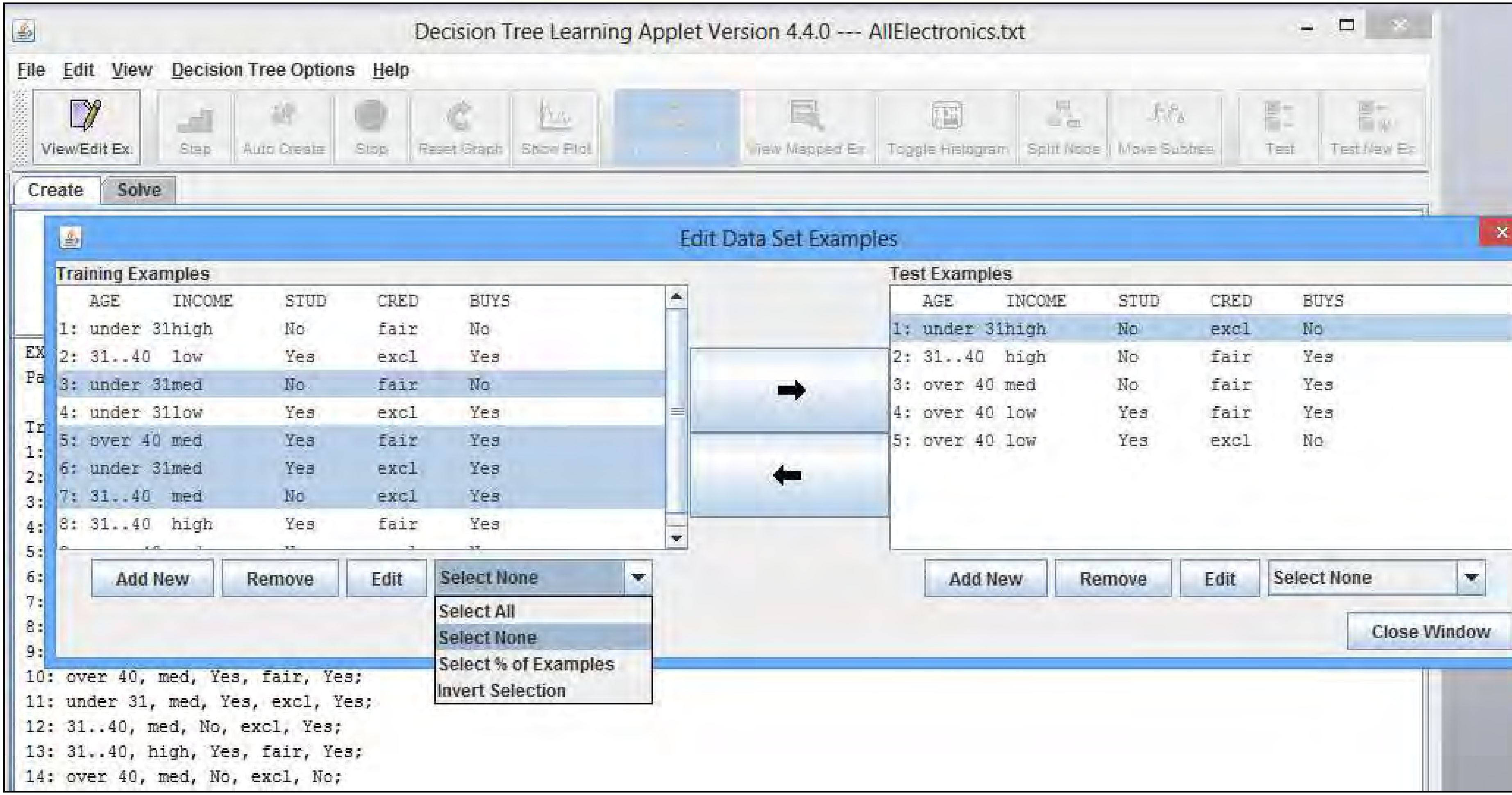
Demonstration



An **example** dataset and its **text representation**.

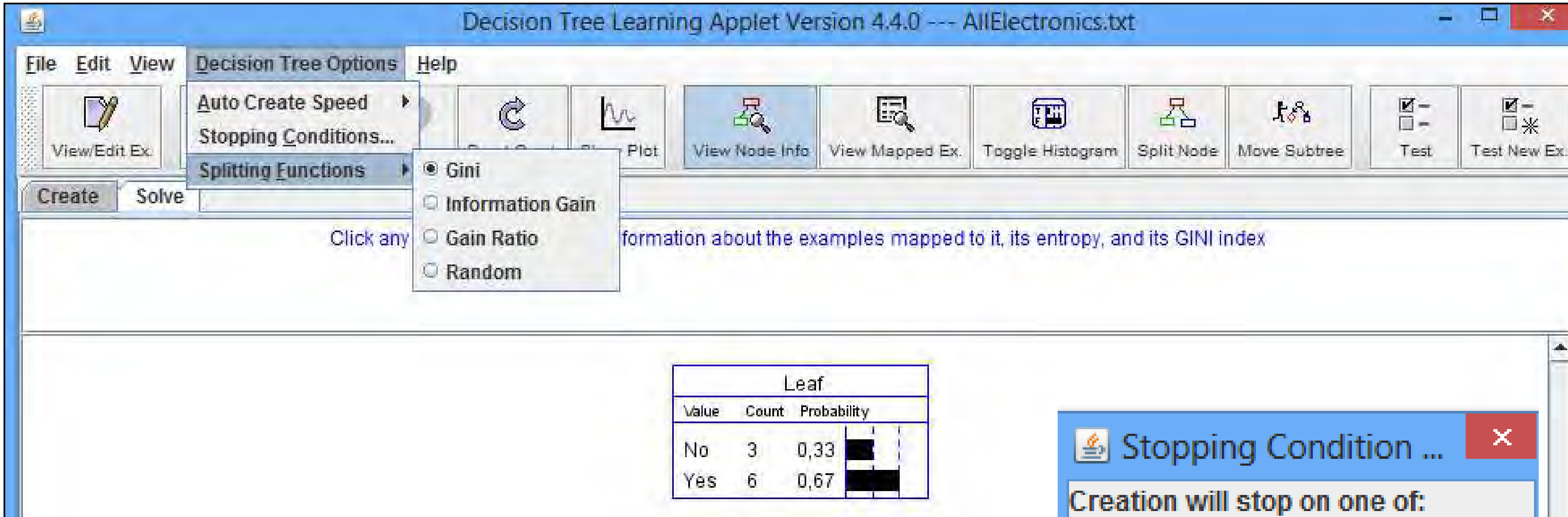
The examples can be imported from or exported to files.

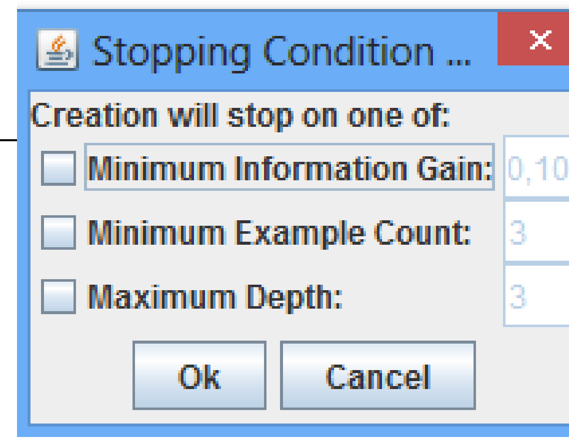
Demonstration



The examples can be separated into the **training set** and the **test set** in a user friendly way.

Demonstration

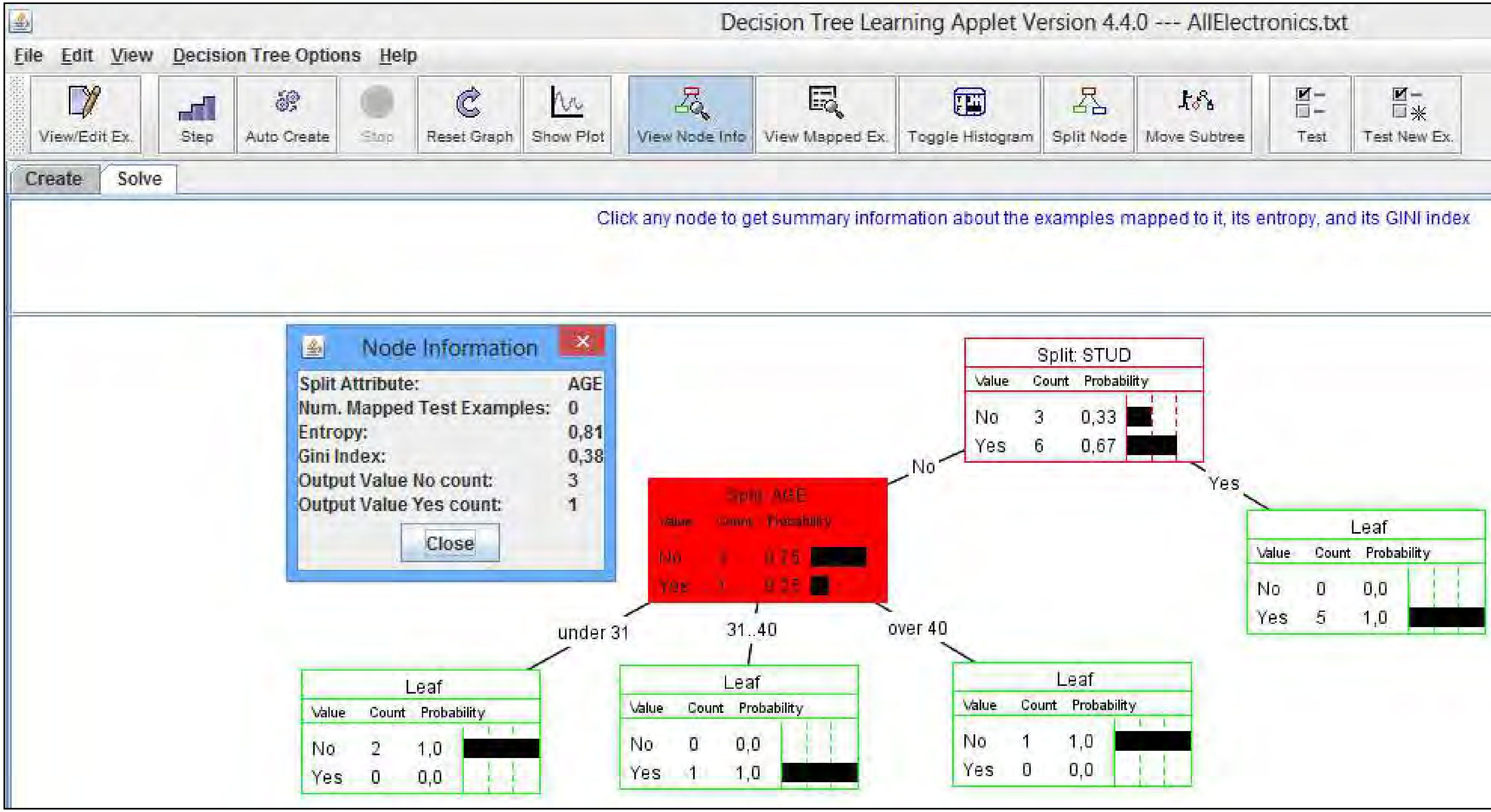




After determining the training set and the test set, the **decision tree generation** can be started on the „Solve” tab.

The decision tree creation process can be controlled through parameters, like the **attribute selection (splitting function)** and the **stopping condition**.

Demonstration



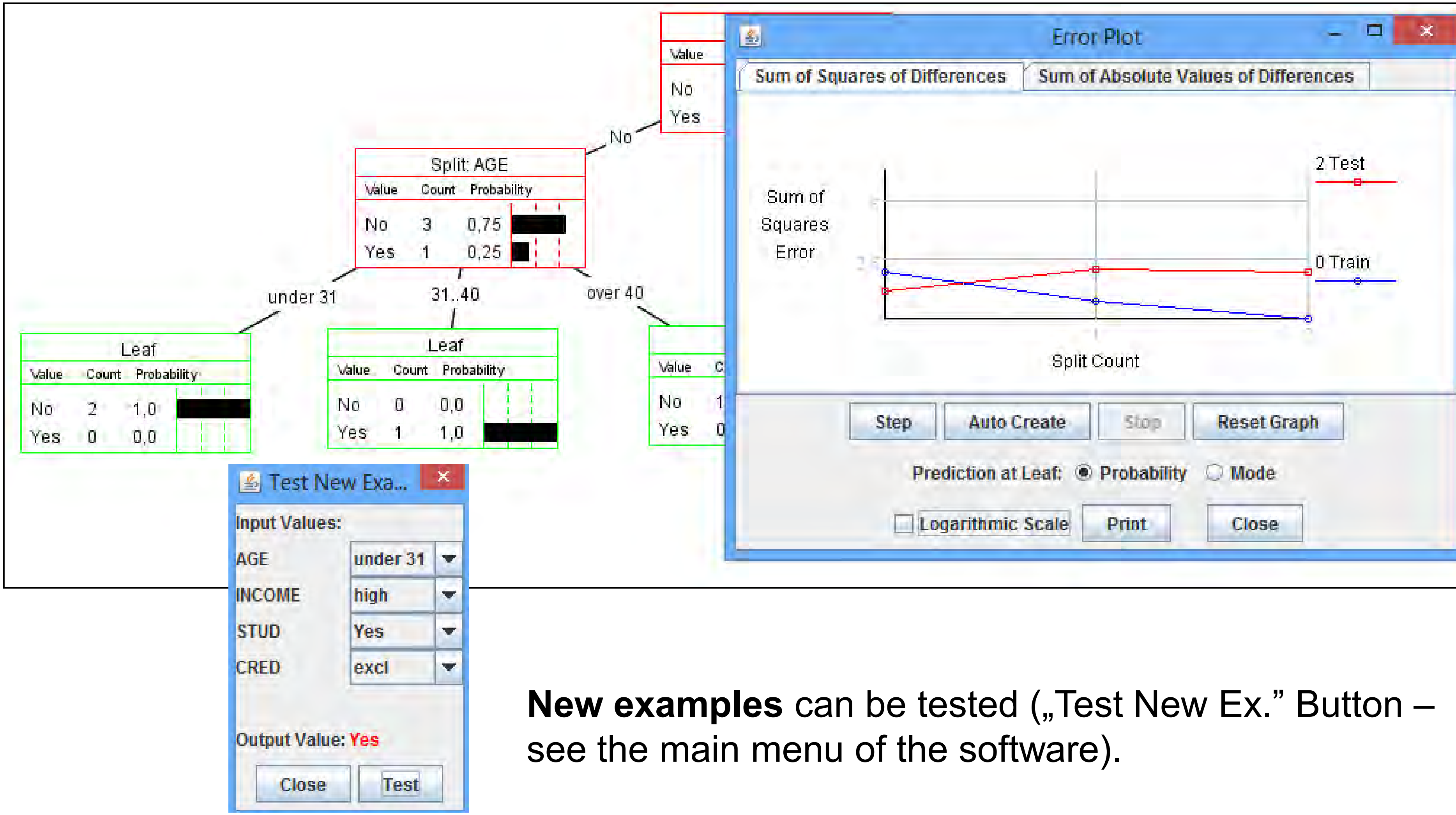
After building the tree (its speed – and steps - also can be controlled) **the created tree can be analyzed** from different viewpoints.

Demonstration

EFOP-3.4.3-16-2016-00009

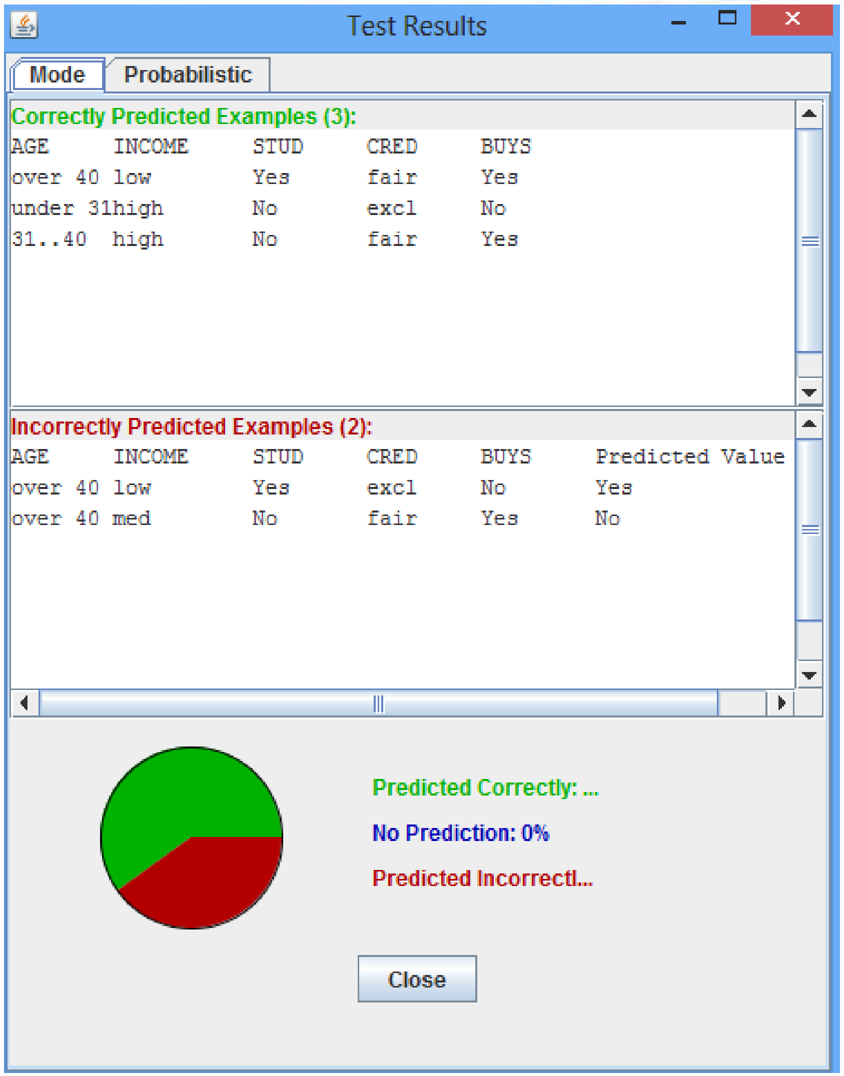
A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

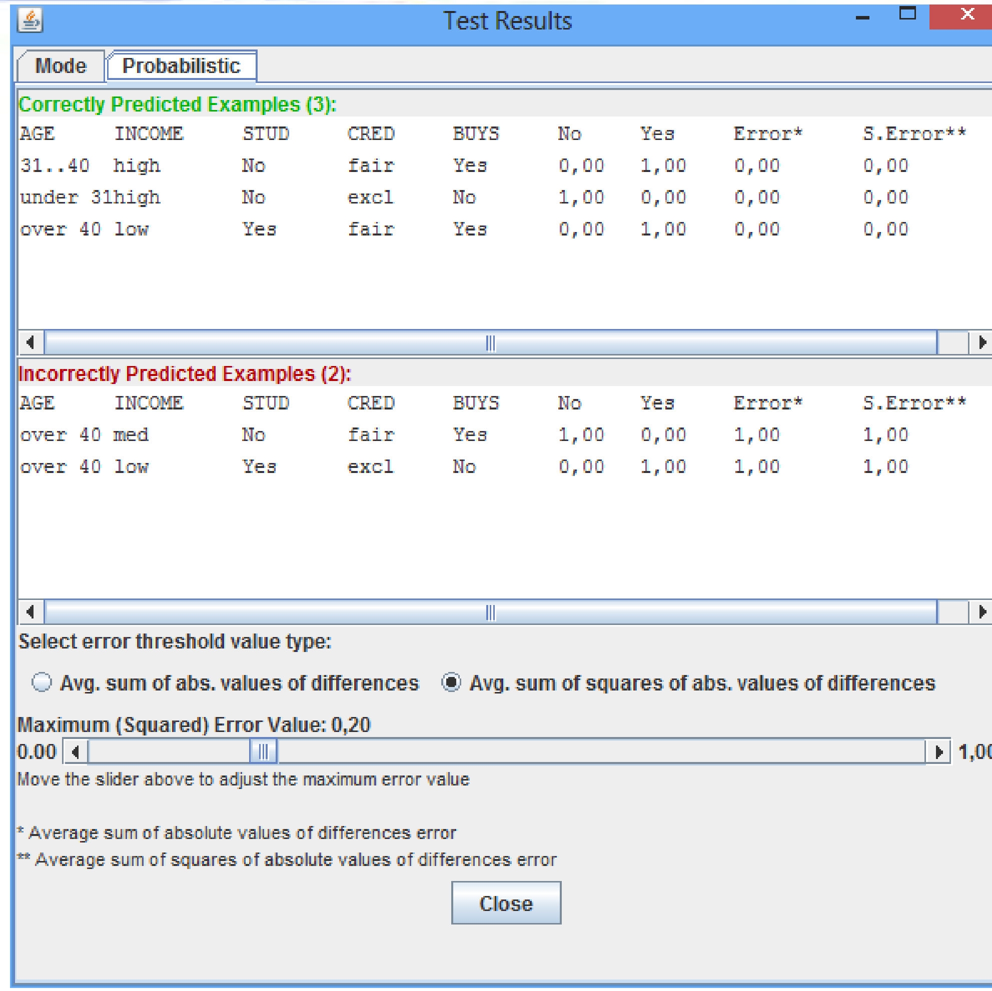
Learning curve related information can be presented graphically (Show Plot button).



New examples can be tested („Test New Ex." Button – see the main menu of the software).

Demonstration





The **performance** of the created decision tree can be analyzed through the test set („Test button” - see the main menu of the software).

EFOP-3.4.3-16-2016-00009

A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

THANK YOU FOR THE ATTENTION!

Reference:

Stuart J. Russel – Peter Norvig:
Artificial Intelligence: A Modern Approach,
Prentice Hall, 2010, ISBN 0136042597

http://aima.cs.berkeley.edu/

SZÉCHENYI 2020

MAGYARORSZÁG KORMÁNYA

Európai Unió
Európai Szociális Alap

BEFEKTETÉS A JÖVŐBE

EFOP-3.4.3-16-2016-00009

A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

ARTIFICIAL INTELLIGENCE

6-7-8. Learning by Neural Networks

Authors:

Tibor Dulai, Ágnes Werner-Stark

SZÉCHENYI 2020

MAGYARORSZÁG KORMÁNYA

Európai Unió
Európai Szociális Alap

BEFEKTETÉS A JÖVŐBE

BASICS

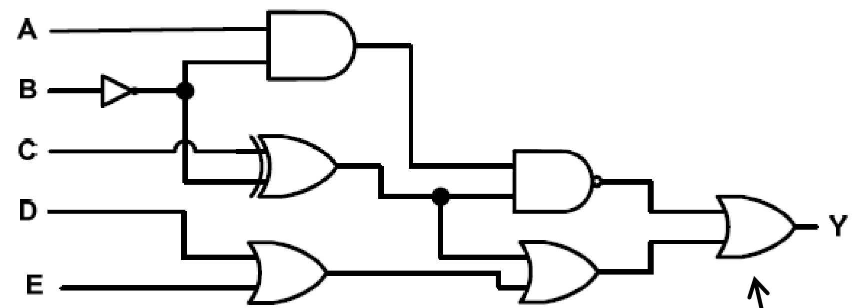
Neural networks - Basics

EFOP-3.4.3-16-2016-00009
A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

Two viewpoins:

(1) **IT viewpoint**:

Function representation by a **network of elements** that are capable to carry out **basic arithmetic calculations (like, e.g., Boole functions)** + its teaching through examples.



…

(2) **Biological viewpoint**:

Mathematical model of the operation of **the brain.**

(a network of neurons)

Neural networks - Basics

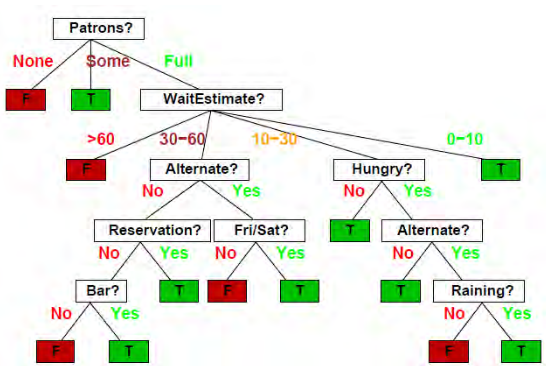
EFOP-3.4.3-16-2016-00009
A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

Neural networks are much better applicable than decision trees, if:

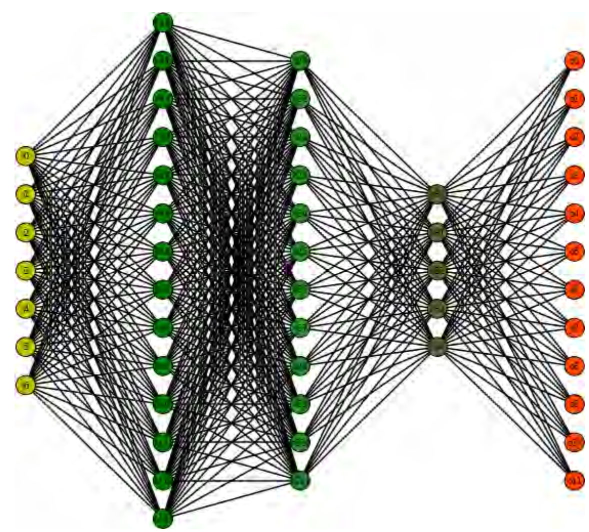
- there is **noise** in the input,

- the **number of input variables is high,**

- the function to represent has **continuous valued codomain**.



<



BRAIN vs. COMPUTER

Brain vs. computer

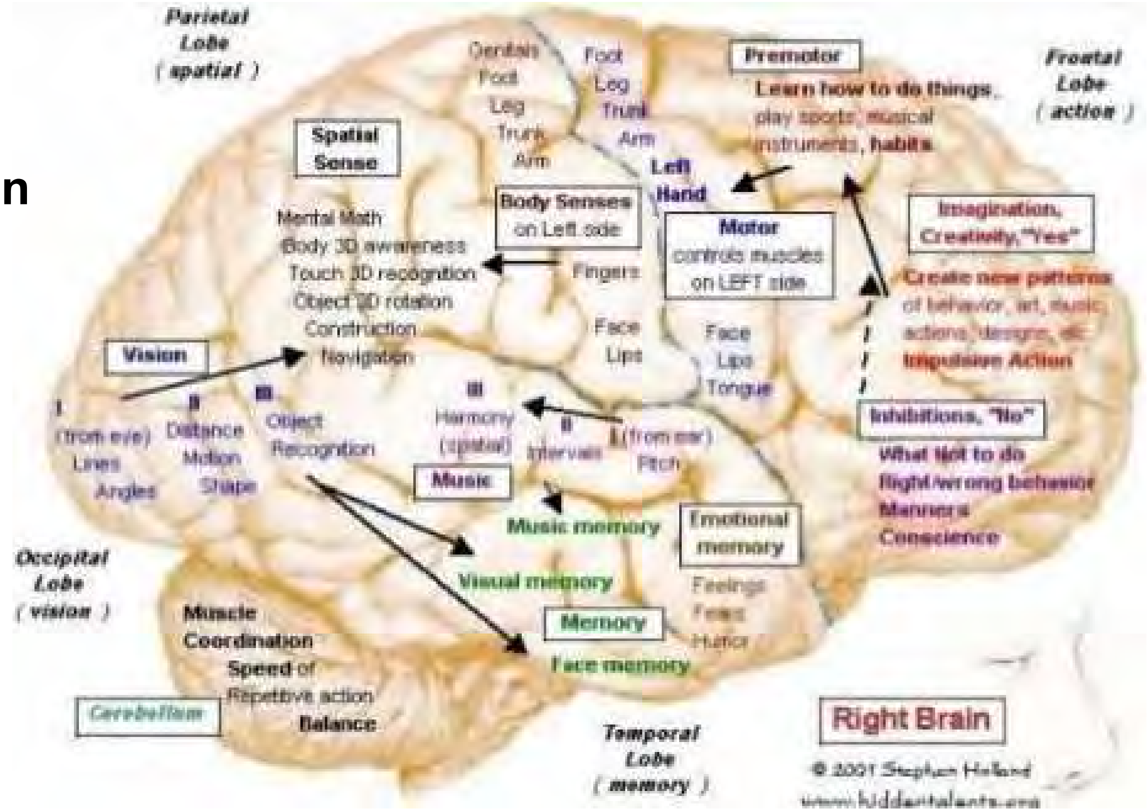
EFOP-3.4.3-16-2016-00009

A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

The **brain is related** „somehow” **to thinking,** however, conscience is still a **mystery**.

The **function map of the human brain** was started to be discovered at the end of XIX. century.

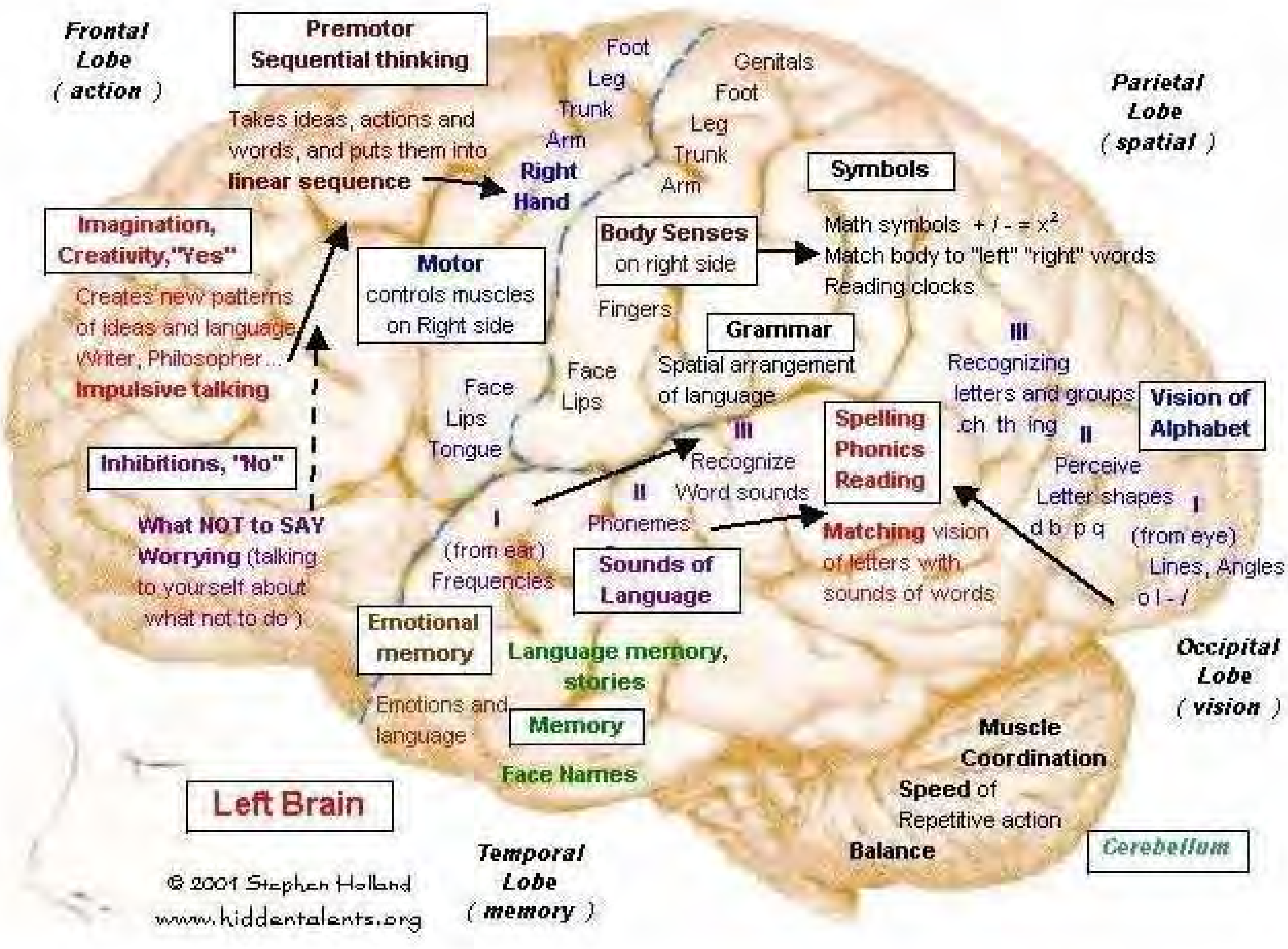
The right (artistic) brain



Sources: **https://universaltcm.com, www.hiddentalents.org**

Brain vs. computer

EFOP-3.4.3-16-2016-00009
A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen



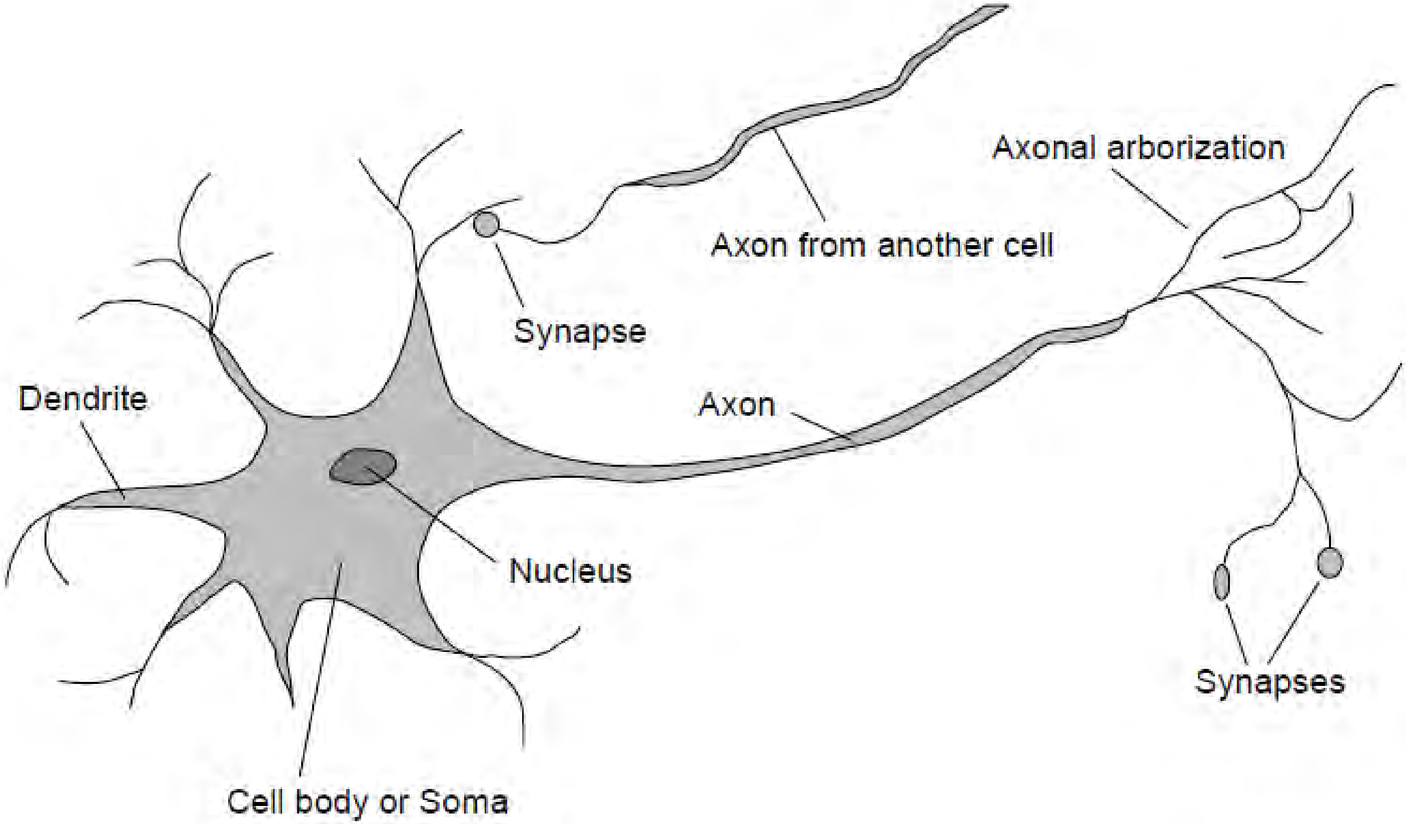
Sources: https://universaltcm.com, www.hiddentalents.org

Brain vs. computer

EFOP-3.4.3-16-2016-00009

A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

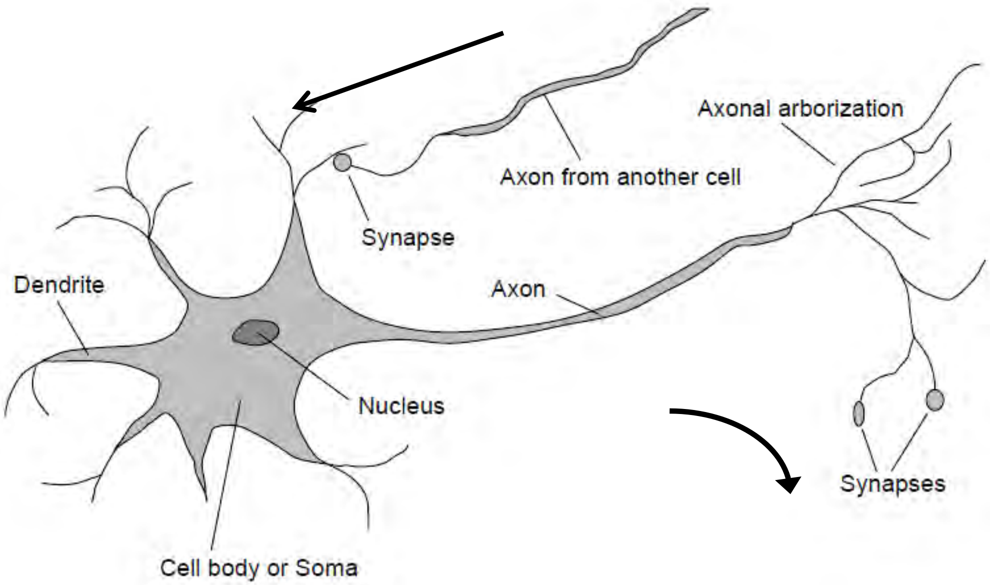
The **basic functional unit of** any nerve tissue – just like of the **brain – is the neuron**. The parts of a neuron are:



In a human brain there are about **10^{11} neurons** of more than 20 types and about **10^{14} synapses**.

Brain vs. computer

EFOP-3.4.3-16-2016-00009
A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen



Communication is made by **electrochemical signals**.

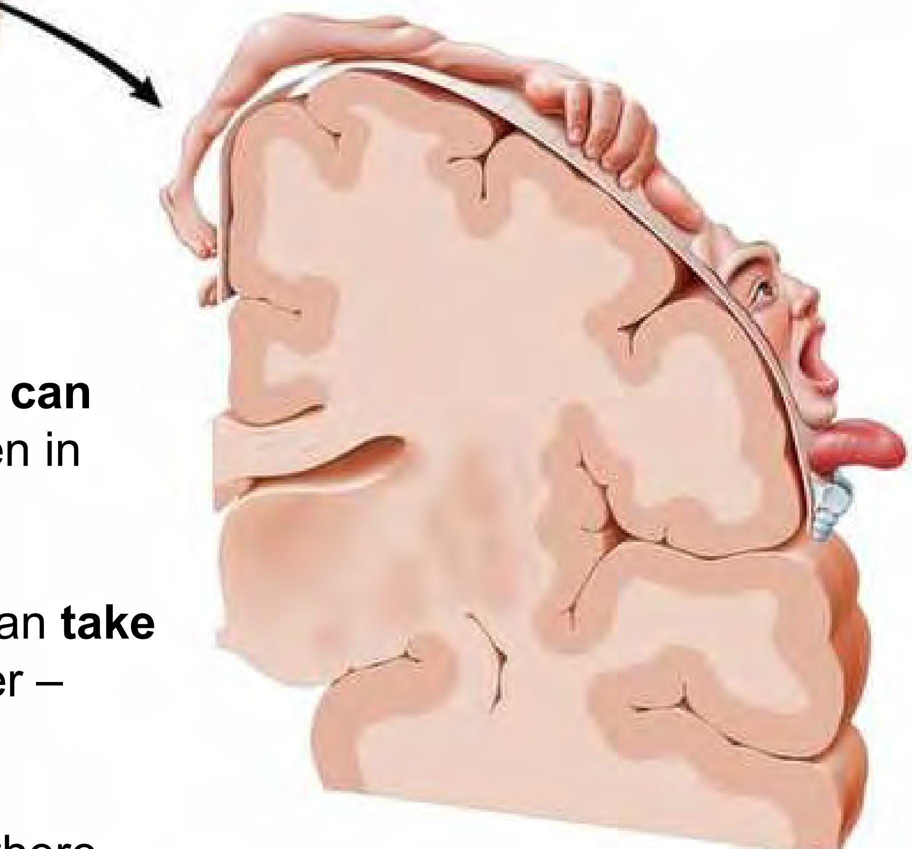
When the **electrical potential** of a neuron reaches a **threshold**, an **action potential** runs through its axon.
This signal may **increase (in case of excitatory synapsis)** or **decrease (in case of inhibitory synapsis) the electrical potential** of the connected neurons.

Plasticity: some **stimulations cause long-term changes** in the strength of the connections.

Learning can be based on it !

Brain vs. computer

EFOP-3.4.3-16-2016-00009
A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen



Information processing happens in the „**cortex cerebri**" part of the brain.

The **functional mapping can be changed** radically even in some weeks.

Some parts of the brain can **take over the function** of other – injured – parts.

In case of some animals there are **multiple mapping**.

It is also a **mystery** how **memories** are stored in the brain.

Source: www.kenhub.com

Brain vs. computer

EFOP-3.4.3-16-2016-00009

A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

Human brain		Digital computer
Number of neurons (slow change)	>	**Number of logic gates** in a CPU (fast growing)
Number of neurons and synapses (slow change)	>	**Number of bits in memory** (fast growing)
Some **milliseconds** for firing	<	about **10 nanoseconds** / instruction
Parallel processing	>	Mostly **serial** processing
High level of **fault tolerance**	>	Very **low** level of **fault tolerance**
Good reaction to new input	>	**Bad reaction to new input**
Graceful degradation	>	**Sudden degradation**

Neural networks aim to **combine** the **high speed of switching** of digital computers with the **parallel processing** of the brain for becoming more powerful than simple making parallel the originally serial algorithms.

EFOP-3.4.3-16-2016-00009
A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

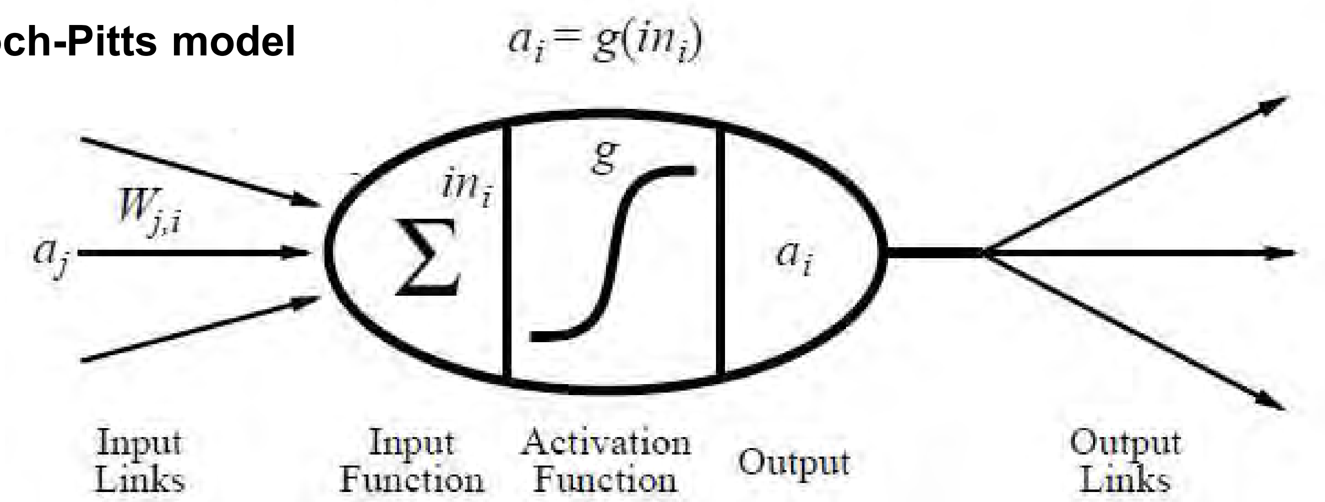
PROCESSING UNITS OF NEURAL NETWORKS

Processing units of neural networks

EFOP-3.4.3-16-2016-00009
A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

A processing unit of a neural network **imitates a neuron** of the nerve tissue: its inputs increase/decrease the unit's activation level (electrical potential) and the level of the output changes when it reaches a threshold value.

McCulloch-Pitts model



Each unit carries out a **simple computation**. **Connecting them** to each other **complex calculations** can be executed.

The **Input Function** value is the **weighted sum** of the inputs of the unit.

Processing units of neural networks

EFOP-3.4.3-16-2016-00009

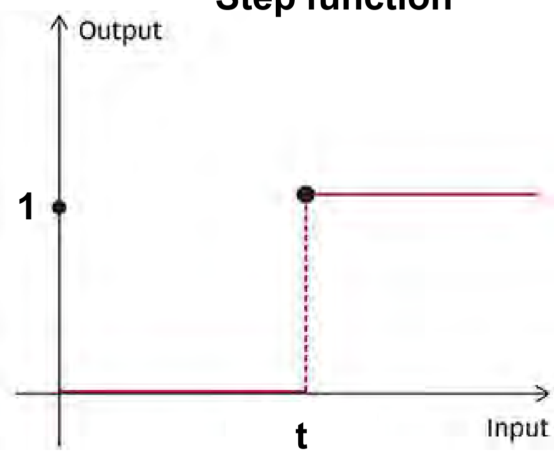
A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

The „**Input Function** part” of the unit is its **linear** step, while the **Activation Function** is **nonlinear**.

The **Activation Function** produces the output of the unit. It operates like a **switch**.

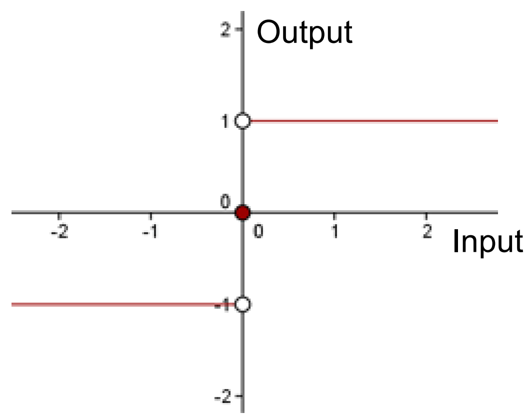
Popular **activation functions** are:

Step function



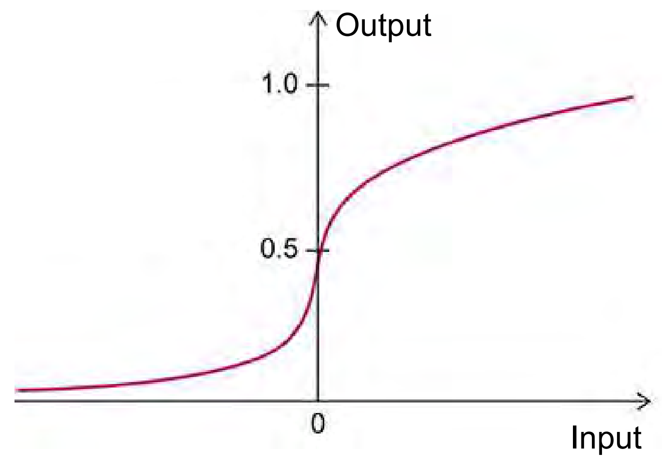
$$step_t(x) = \begin{cases} 1, & if\ x \geq t \\ 0, & if\ x < t \end{cases}$$

Sign function



$$sign_t(x) = \begin{cases} 1, & if\ x > 0 \\ 0, & if\ x = 0 \\ -1, & if\ x < 0 \end{cases}$$

Sigmoid function



$$sigmoid(x) = \frac{1}{1 + e^{-x}}$$

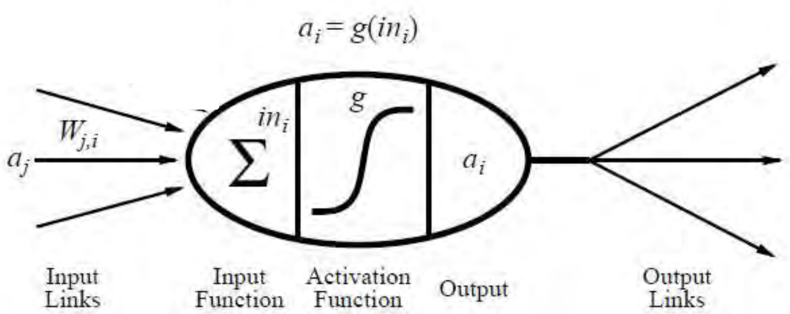
Processing units of neural networks

EFOP-3.4.3-16-2016-00009

A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

Usually, **all the units** of a neural network applies the **same activation function**.

It is also possible to **define a (nonzero) threshold** for the **sign** and the **sigmoid** function.



It means, that in the neural network (that is a connected set of some units) there are **two different variable types** to adjust:

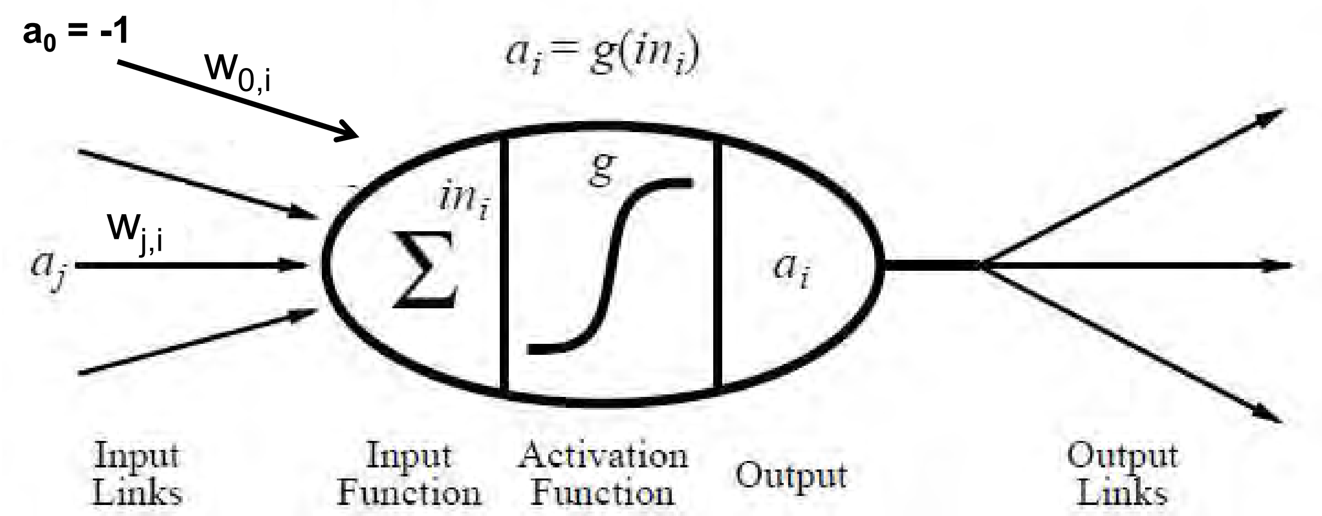
- the **weights**, and
- the **thresholds**.

To make things simpler, threshold (t) can be substituted by an extra input (a_0) /called **bias**/, where the input value is -1 and its weight is the value of the threshold:

$$a_i = step_t\left(\sum_{j=1}^{n} w_{j,i} * a_j\right) = step_0\left(\sum_{j=0}^{n} w_{j,i} * a_j\right), where\ a_0 = -1\ and\ w_{0,i} = t.$$

Processing units of neural networks

This way, **a general unit of a neural network** looks like as follows **(the threshold of function g() is 0)**:



The **output** of a unit can be **calculated** this way:

$$a_i = g(in_i) = g\left(\sum_{j=0}^{n} w_{j,i} * a_j\right) = \underline{\mathbf{W}}_i * \underline{\mathbf{a}}_i$$

Let us plan **one-unit neural networks**, that **realize the logical AND, OR and NOT** functions, supposing that every input and the output have logical (0 or 1) values!

1. **AND function:**

$input_1$	$input_2$	output
0	0	0
0	1	0
1	0	0
1	1	1

Processing units of neural networks

EFOP-3.4.3-16-2016-00009
A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

Let us plan **one-unit neural networks**, that **realize the logical AND, OR and NOT** functions, supposing that every input and the output have logical (0 or 1) values!

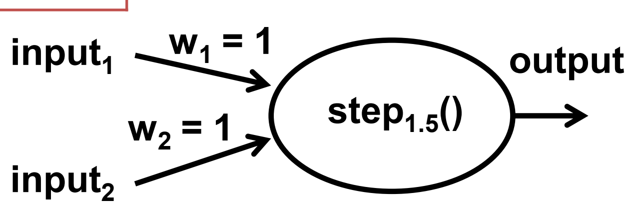
1. **AND function:**

$input_1$	$input_2$	output
0	0	0
0	1	0
1	0	0
1	1	1

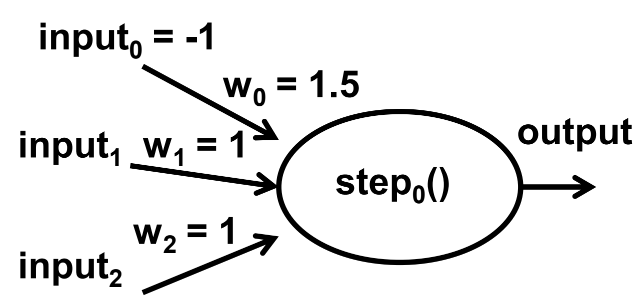
The output is on high level only if both inputs are high.

Simple the sum of the inputs can be created **(its result will be 0 or 1 or 2)**, then the threshold for „swithching" the output to high level can be anywhere between 1 and 2 **(let it be, e.g., 1.5)**.

So, the neural network will be, e.g.:



or



2. **OR function:**

$input_1$	$input_2$	output
0	0	0
0	1	1
1	0	1
1	1	1

Processing units of neural networks

EFOP-3.4.3-16-2016-00009
A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

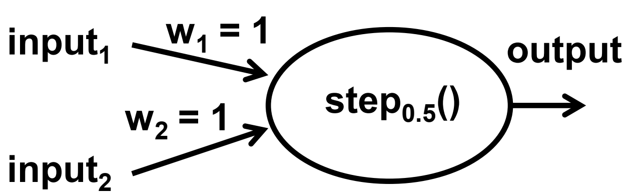
2. **OR function:**

$input_1$	$input_2$	output
0	0	0
0	1	1
1	0	1
1	1	1

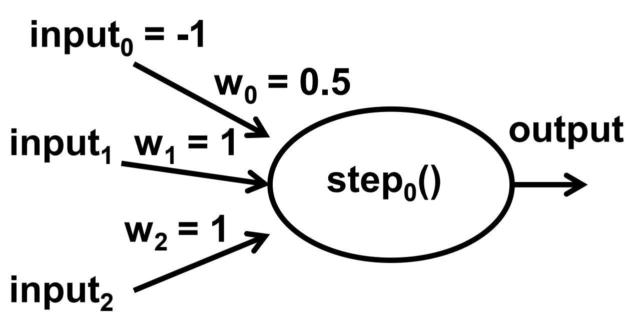
The output is on high level if any of the two inputs is high.

Simple the sum of the inputs can be created **(its result will be 0 or 1 or 2)**, then the threshold for „swithching” the output to high level can be anywhere between 0 and 1 **(let it be, e.g., 0.5)**.

So, the neural network will be, e.g.:



or



EFOP-3.4.3-16-2016-00009
A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

2. **NOT function:**

input	output
0	1
1	0

Processing units of neural networks

EFOP-3.4.3-16-2016-00009
A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

2. **NOT function:**

input	output
0	1
1	0

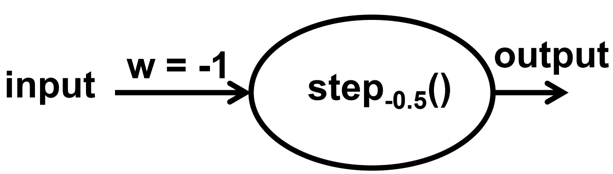
The output is on high level if and only if the input is low.

The activation function has to do a kind of „invertation”, so its original curve has to be mirrored to the line that is parallel with y axis and includes the central point of the original curve of the function.

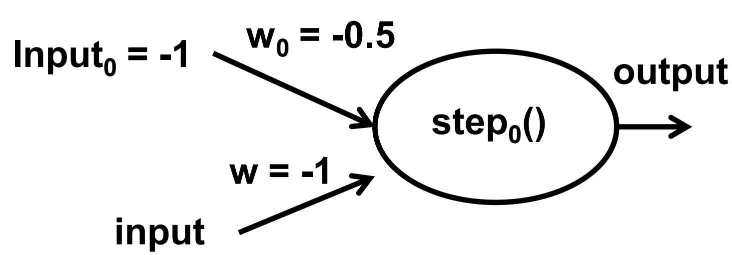
The **weight** of the input **has to be negative**.

If the weight is, e.g., -1, then the possible input function values are 0 or -1. So, the threshold has to be anywhere between -1 and 0 **(let it be, e.g., -0.5)**.

So, the neural network will be, e.g.:



or



Processing units of neural networks

EFOP-3.4.3-16-2016-00009

A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

We have seen that **logical AND, OR and NOT functions** of any logical input variables can be **realized by a single unit neural network**.

⬇

Any Boolean function of the logical input variables **can be realized** by appropriate connections of these units **(just like in case of logical circuits that are built up by logical gates).**

Let us connect some processing units of neural networks!

EFOP-3.4.3-16-2016-00009

A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

NEURAL NETWORKS

Their structure

Structure of neural networks

EFOP-3.4.3-16-2016-00009

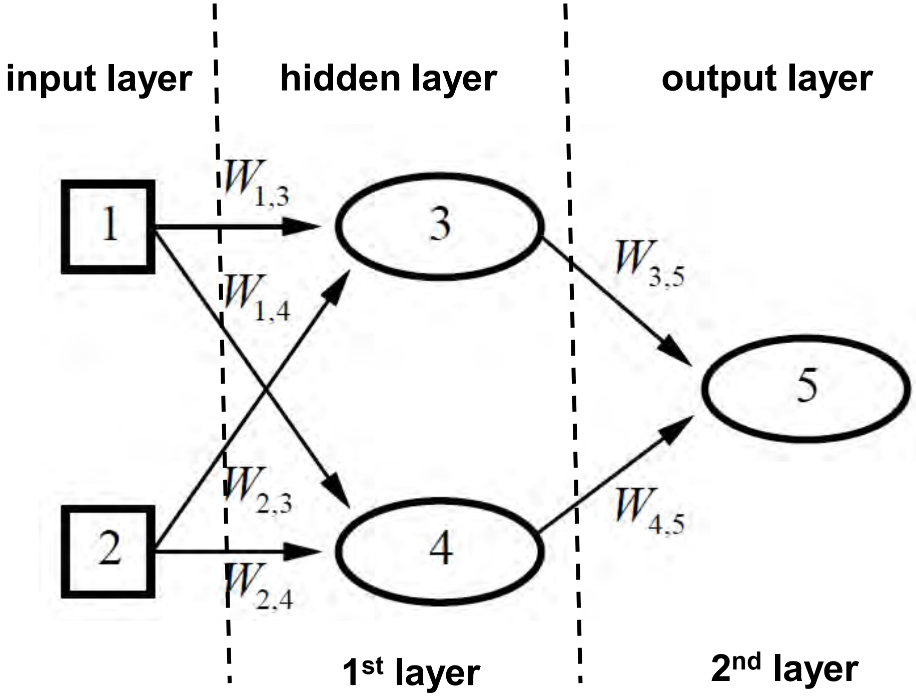
A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

The **units** are connected by **links** and each link has its own **weight**. The **weights store the information** and they are changed during the training process.

Each unit carries out its **local calculation** based on its inputs to obtain its own **activation level**. Then, every unit's output value is distributed to the following units through the links **(just like in case of axons)**. There is **no global control** over the units.

The units usually create **layers**.
The **input layer** is not a real layer: it does not carries out any calculation, only copies the input values to the first layer's neurons.

Layers between the input layer and the **output layer** are „**hidden layers**".

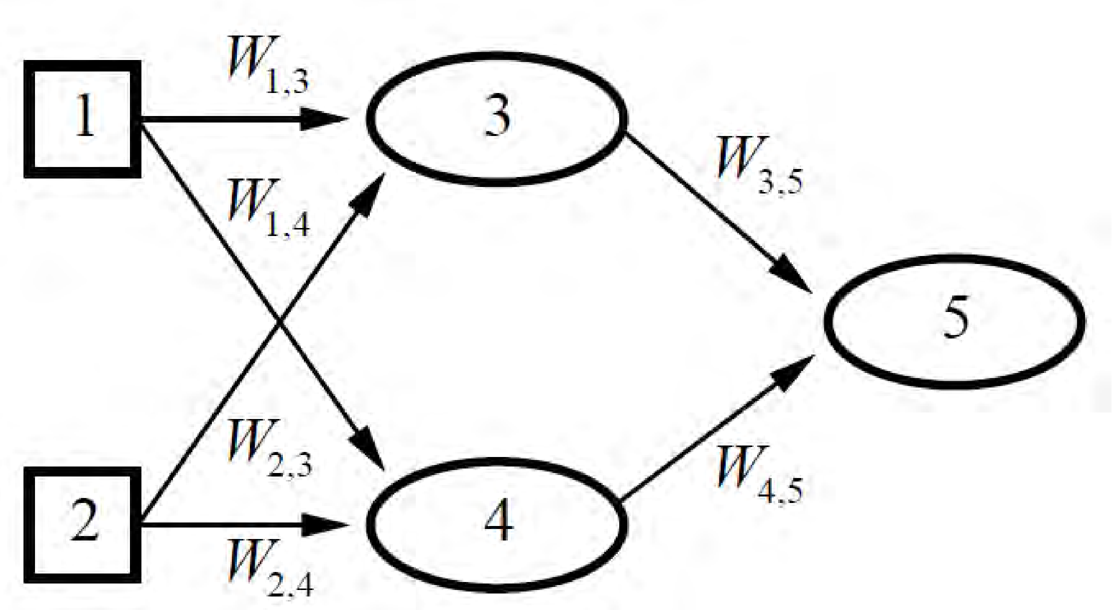


Structure of neural networks

EFOP-3.4.3-16-2016-00009
A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

The **structure and the weights** of a neural network determines the **function of its inputs** that is represented by the network.

For example:



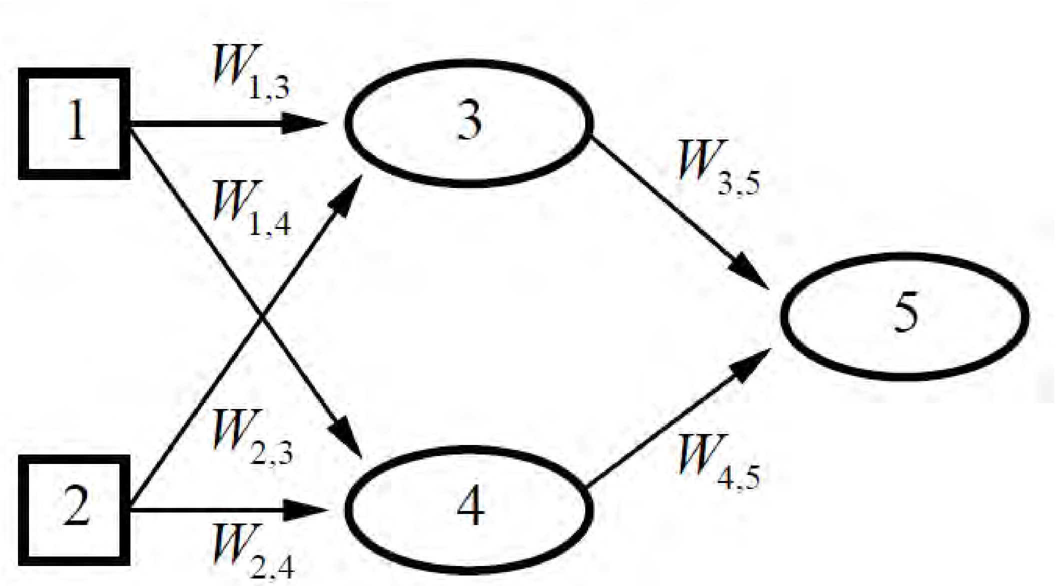
a_5 = **?**

Structure of neural networks

EFOP-3.4.3-16-2016-00009
A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

The **structure and the weights** of a neural network determines the **function of its inputs** that is represented by the network.

For example:

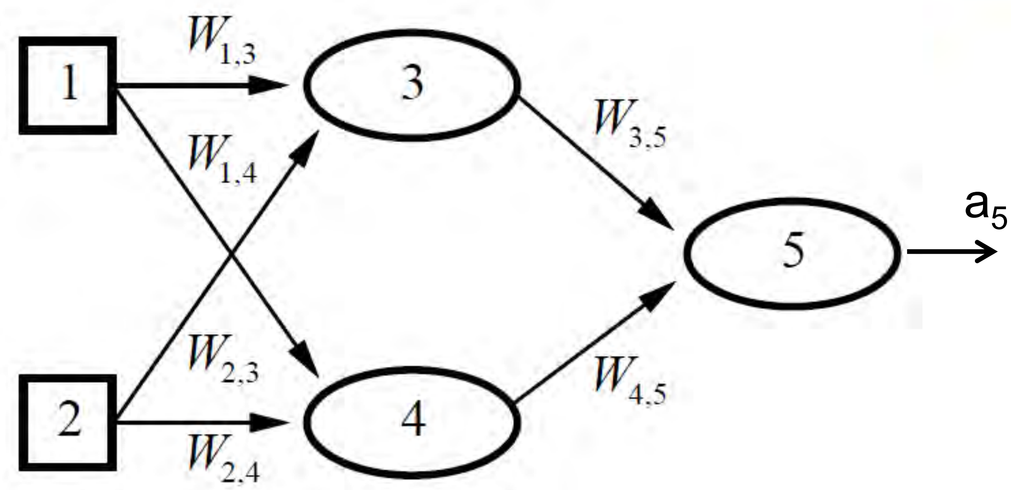


$$a_5 = g(in_5) = g\left(W_{3,5} * a_3 + W_{4,5} * a_4\right) = g(W_{3,5} * g(in_3) + W_{4,5} * g(in_4)) =$$

$$= g\left(W_{3,5} * g\left(W_{1,3} * a_1 + W_{2,3} * a_2\right) + W_{4,5} * g\left(W_{1,4} * a_1 + W_{2,4} * a_2\right)\right)$$

Structure of neural networks

EFOP-3.4.3-16-2016-00009
A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen



$$a_5 = g(in_5) = g\left(W_{3,5} * a_3 + W_{4,5} * a_4\right) = g(W_{3,5} * g(in_3) + W_{4,5} * g(in_4)) =$$

$$= g\left(W_{3,5} * g\left(W_{1,3} * a_1 + W_{2,3} * a_2\right) + W_{4,5} * g\left(W_{1,4} * a_1 + W_{2,4} * a_2\right)\right)$$

This way we can see that the **training** of the network – the **tuning of the weights** of the network in the way that the resulted function fits the best to the training set – is a **nonlinear regression** **(since g() is a nonlinear function).**

Structure of neural networks

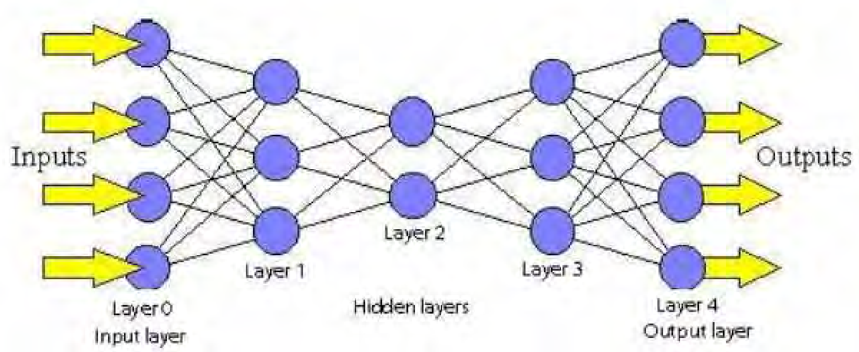
EFOP-3.4.3-16-2016-00009

A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

Different **structures** have different **computational properties**.

The **two most important structures** are:

1. **Feed-forward** neural networks:
   - They have **no loop**, so they are **directed acyclic graphs**, actually.
   - **Arcs** exist from units **only to the** units of the **next neighbour layer**.
   - Loopness causes that these networks **do not have „memory"**.
   - The **output is determined by the current input**.
   - They are a **well-known** class of neural networks.



Structure of neural networks

Different **structures** have different **computational properties**.

The **two most important structures** are:

2. **Recurrent** neural networks:
   - They have **arbitrary topology** (so, they **may have loops**).
   - These networks **have „memory”** → more similar to the brain than feed-forward networks.
   - The **output is determined** not only **by the current input** but **by the current state** of the network, too.
   - These networks may **oscillate**, they may be **instable**.

Two quite well-known classes of recurrent neural networks are:

The **Hopfield networks** and the **Boltzmann machines**.

Structure of neural networks

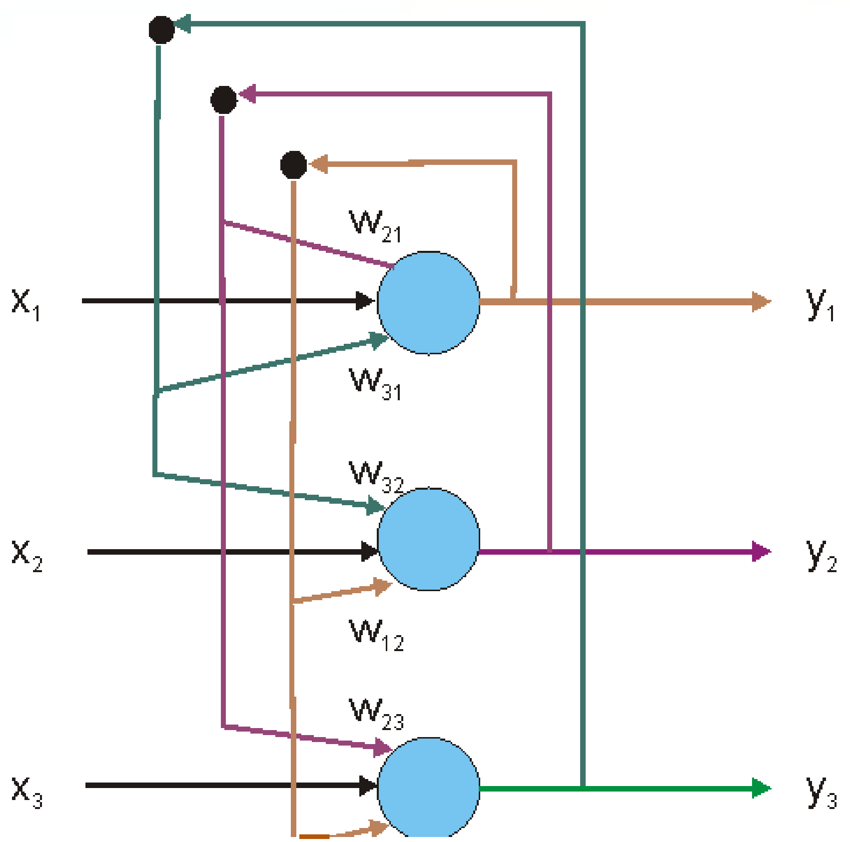
EFOP-3.4.3-16-2016-00009

A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

2.1 **Hopfield networks**

- they have only **one „layer”** (each neuron is both input and output, too)
- $w_{i,j} = w_{j,i}$ (**symmetric weights**)
- the activation function is the **sign() function**
- each **activation value is 1 or -1**
- they behave like **associative memory**

(for a new example they represents the function of the training example that is the most similar to the current input)

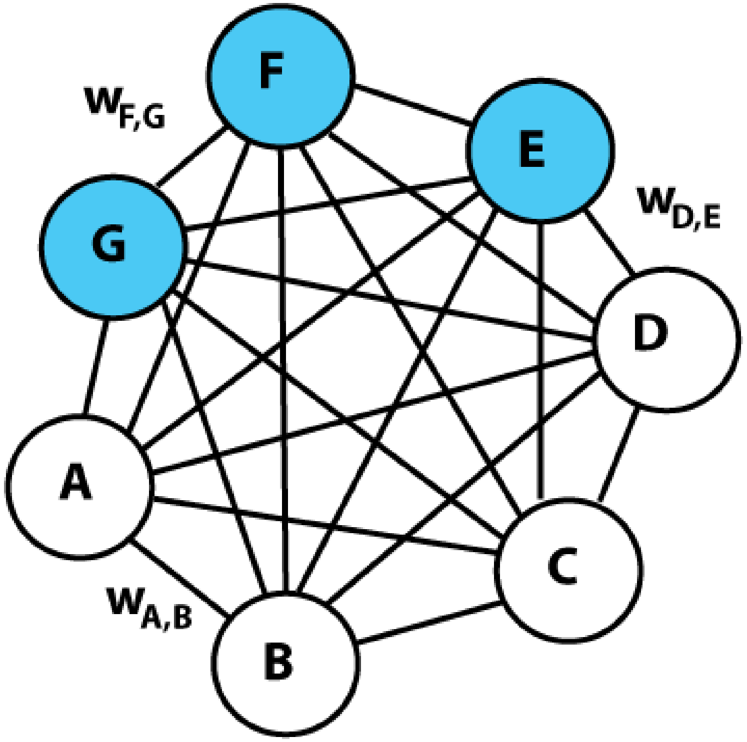


Hopfield networks are capable to store ***0,138*n*** **examples in *n* units**.

2.1 **Boltzmann machines**

- there are units that are **neither input nor output neurons**
- $w_{i,j} = w_{j,i}$ (**symmetric weights**)
- the **activation function is stochastic**: **the probability of being the output value equal to 1 is determined by a function of the weighted input values**

An example for Boltzmann machine:



Blue neurons: hidden units
White neurons: visible units

Since we have much more knowledge in the field of **feed-forward networks** than recurrent networks, **this course covers them**.

Source of the figure: wikipedia.org

Structure of neural networks

EFOP-3.4.3-16-2016-00009
A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

A very **important** feature of a neural network is its **structure**:

How many units does it has and how they are connected.

If the number of units is:

- **too low**, then the network may **not be able to learn the desired function**.
- **too big**, then the number of free parameters is high and **overfitting may happen**, so the neural network will **not be able to generalize**.

There is **no exact method to find optimal structure** for a problem.
Usually „**trial and error**” methods are applied, and the **experiments** to find the appropriate network structure may **take months**.

Structure of neural networks

The looking for the **appropriate network structure** can be handled as a **search problem**.

In the search space of the network structures **genetic algorithm (GA)** or **hill climbing method** can be applied, for example.
However, **GA requires huge computation capacity**: it handles a population of instances and all the instances have to be trained and evaluated in each generation.

The search methods for the desired network structure that are based on one instance have **two main approaches**:

Top-down

Started from a wholly connected big network the less important **links and neurons are eliminated**.

e.g., the **optimal brain damage** method

Bottom-up

Started from the only neuron that gives good result on the biggest part of the training set, **newer units are added** with respect to the other examples, too.

e.g., the **tiling algorithm**

PERCEPTRONS

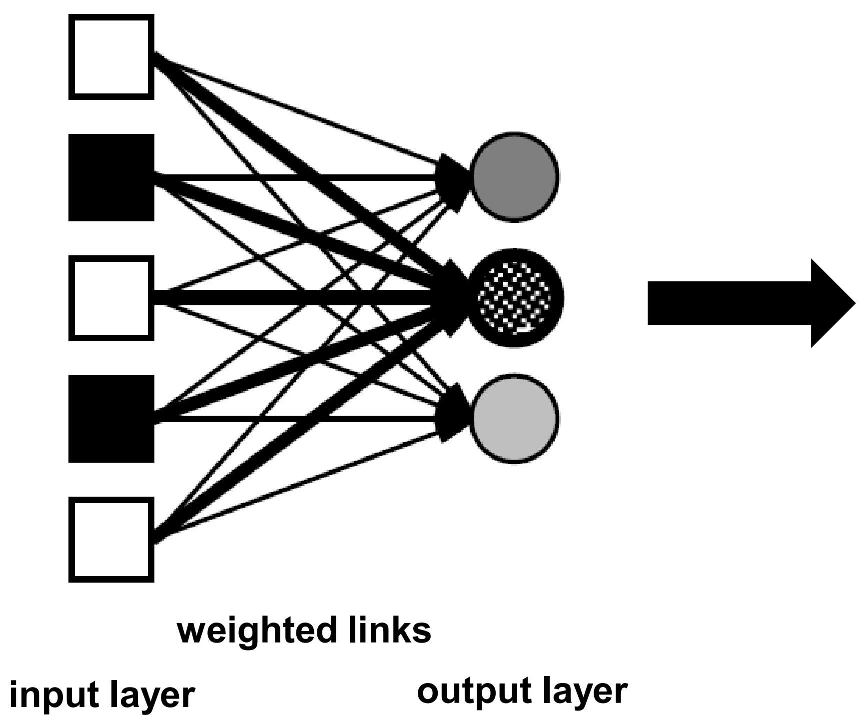
Perceptrons

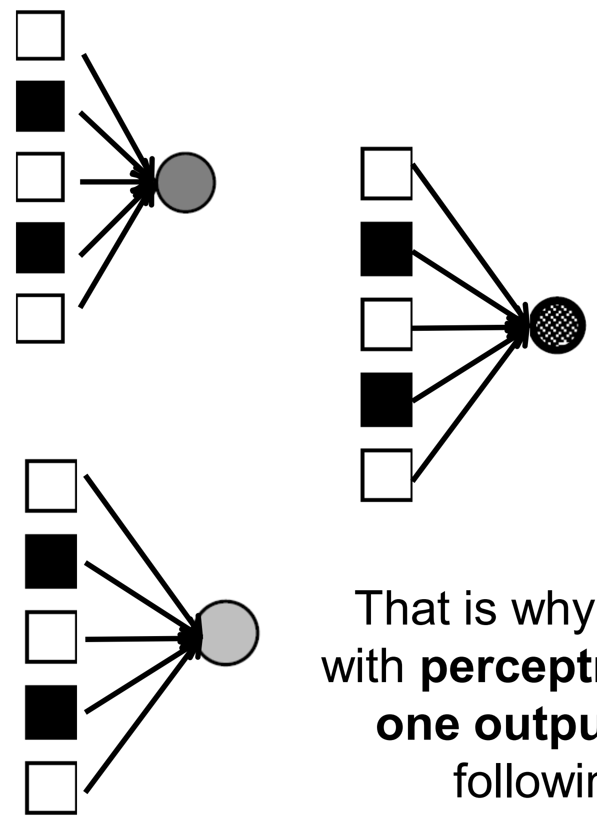
EFOP-3.4.3-16-2016-00009

A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

We use the concept „**perceptron**” to **one-layer feed-forward neural networks**.

Each multi-output perceptrons can be separated easily into a set of one-layer perceptrons, since **each weight (link) influences only one output**.

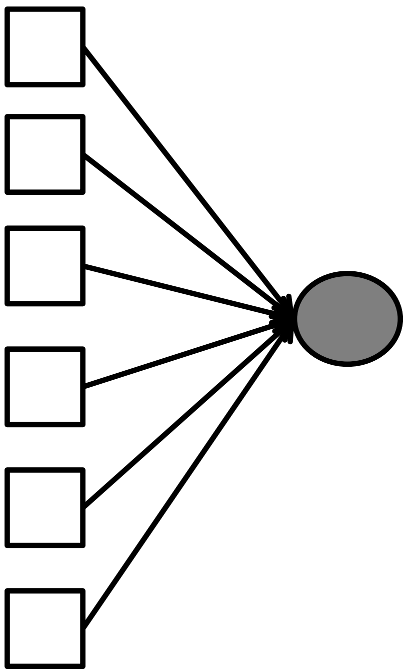




That is why we deal with **perceptrons with one output** in the followings.

EFOP-3.4.3-16-2016-00009
A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

Representation capability of perceptrons



One output of a perceptron is the result of the calculation of **exactly one neuron**.

We have seen, that, e.g., the logic AND, OR and NOT functions of the input can be calculated by one unit.

What about other functions?

Can we realize an n-input **majority function** by a perceptron?

/the output of majority function is 1 if and only if more than the half of the input values are 1/

EFOP-3.4.3-16-2016-00009
A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

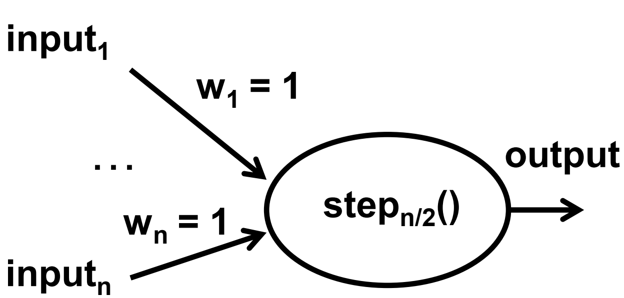
Representation capability of perceptrons

Can we realize an n-input **majority function** by a perceptron?

/the output of majority function is 1 if and only if more than the half of the input values are 1/

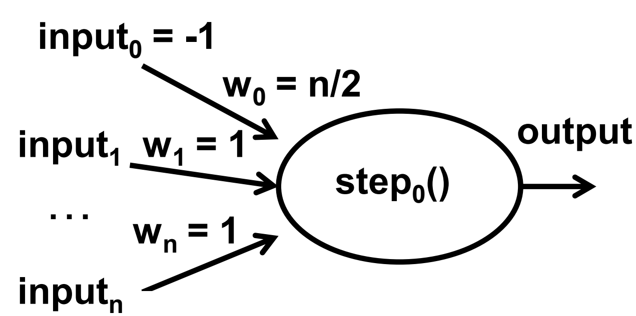
The answer is: **YES**

Possible **solutions** are:



$w_i = 1, \forall i = 1, \dots, n$

or



$w_i = 1, \forall i = 1, \dots, n$

Representation capability of perceptrons

We have seen a very simple realization of the n-input majority function by a **neural network** (**1 neuron + n weights**).

If we realize the same function by a **decision tree**, it requires a tree with **$O(2^n)$ nodes**.

Is neural network always a more efficient choice?
By the way, **is every function realizable by perceptrons**?

Of course, **NOT**!

Let us **discover the function class that is realizable** by perceptrons!

Representation capability of perceptrons

As one output value is the result of the calculation by one unit and the activation function operates like a switch, **any input's influence onto the output has only one direction (independently from the other inputs).**

It means, that, for **example**, supposing binary input and output values:

- if the output is 0 and the i^{th} element of the input vector is 0, and

- without modifying other input values, the change of this (i^{th}) input from 0 to 1 results in 1 on the output, then

→ **it means that the weight between the i^{th} input and the output is positive**

- there is no input vector which results in a 1 → 0 switch on the output if the i^{th} element of the input vector is changed from 0 to 1.

EFOP-3.4.3-16-2016-00009
A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

Representation capability of perceptrons

The consequences for the case of **continuous input values and binary output** values are that considering the **progress along the scale of any input continuously - into one direction -**, the **output value changes only once**.

Recall, that the calculation of the output of neuron *i* is as follows:

$$a_i = g(in_i) = g\left(\sum_{j=0}^{n} w_{j,i} * a_j\right) = \underline{\mathbf{W}}_i * \underline{\mathbf{a}}_i$$

Suppose, that the activation function is the step function:

$$step_t(x) = \begin{cases} 1, & if\ x \geq t \\ 0, & if\ x < t \end{cases}$$

Perceptrons

EFOP-3.4.3-16-2016-00009

A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

Representation capability of perceptrons

Putting them together and supposing that instead of the *t* threshold of the step function bias is used:

$$a_i = 1 \iff \sum_{j=0}^{n} w_{j,i} * a_j = \underline{\mathbf{W}}_i * \underline{\mathbf{a}}_i \geq 0$$

$\underline{\mathbf{W}}_i * \underline{\mathbf{a}}_i = 0$ separates the space into two parts (the one with 0 output and the other with 1 output)

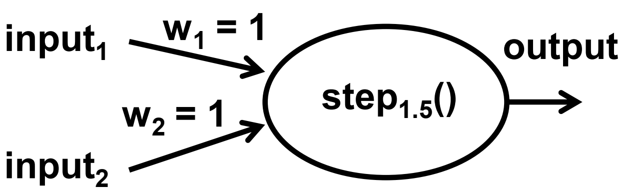
It defines a **hyperspace** (if the input has *n* dimensions, then it is a linear separator in an *n* dimensional hyperspace)

So, **perceptrons are able to represent only linearly separable functions**.

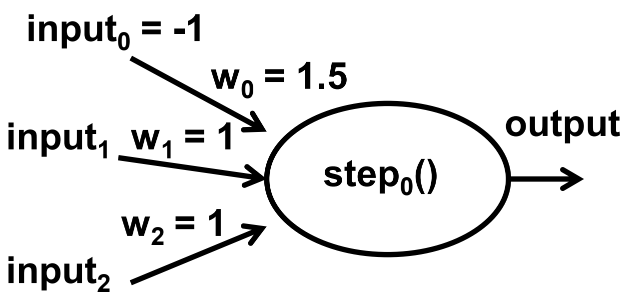
Perceptrons

Representation capability of perceptrons

Recall the 2-input **logical AND** function:



or



Here, the points of the 2-dimensional input space which give output value 1 are:

$\underline{\mathbf{W}}_i * \underline{\mathbf{a}}_i \geq 0$, in details: $\langle w_0, w_1, w_2 \rangle * \langle input_0, input_1, input_2 \rangle \geq 0$

It means in this case: $\langle 1.5, 1, 1 \rangle * \langle -1, input_1, input_2 \rangle \geq 0$

Perceptrons

EFOP-3.4.3-16-2016-00009

A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

Representation capability of perceptrons

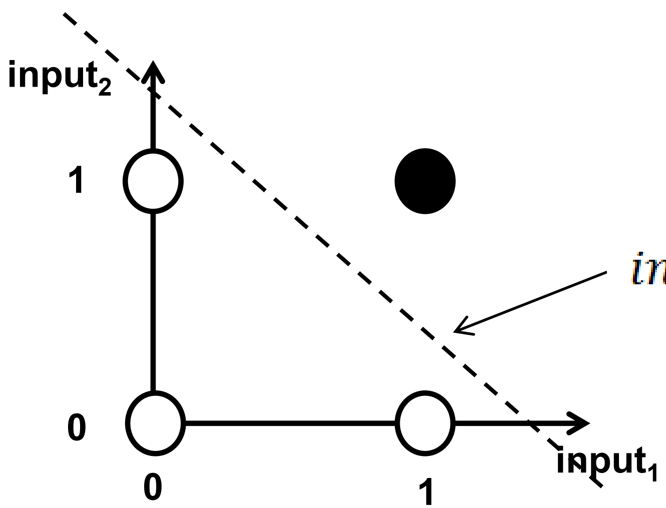
The **logical AND** function (cont.):

$$\langle 1.5, 1, 1 \rangle * \langle -1, input_1, input_2 \rangle \geq 0$$

$$-1 * 1.5 + 1 * input_1 + 1 * input_2 \geq 0$$

$$input_2 \geq 1.5 - input_1$$

Graphically:



The 2-dimensional linear separator line:

$$input_2 = 1.5 - input_1$$

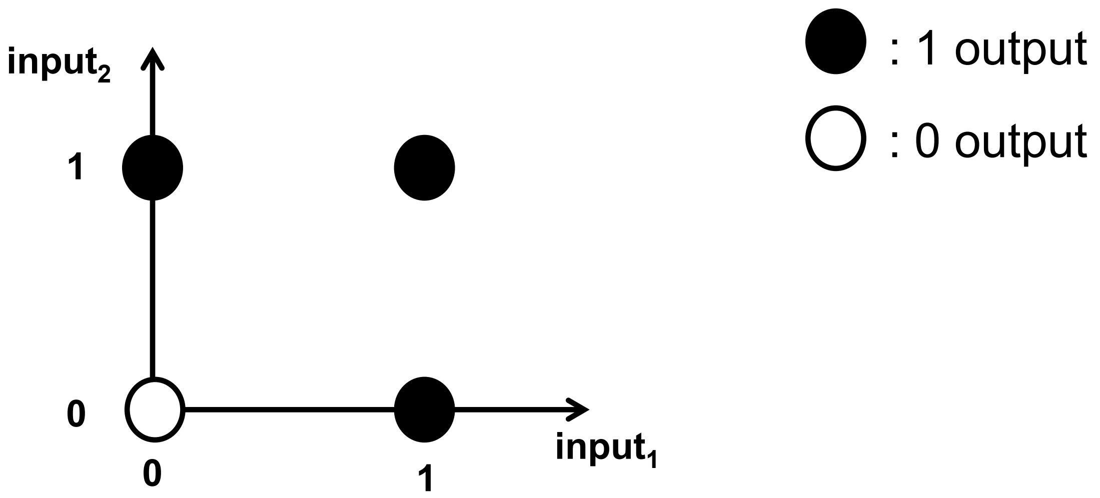
● : 1 output

○ : 0 output

EFOP-3.4.3-16-2016-00009
A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

Representation capability of perceptrons

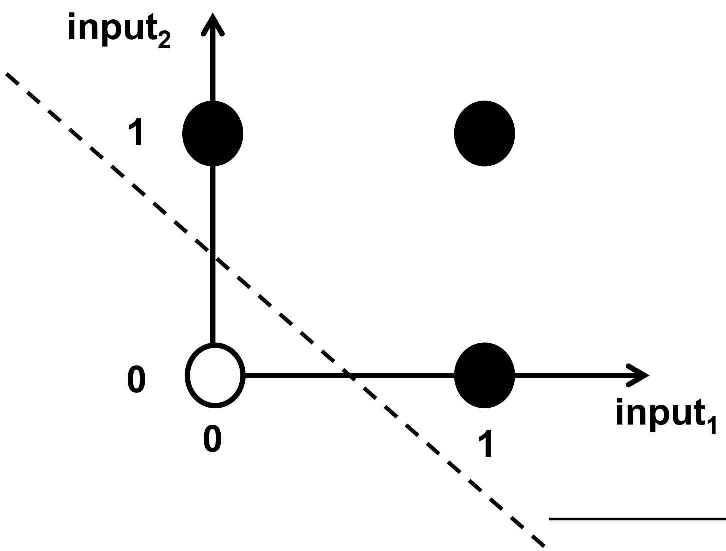
Let us do the same in the opposite way for the **logical OR** function (with 2 inputs):



Perceptrons

Representation capability of perceptrons

Let us do the same in the opposite way for the **logical OR** function (with 2 inputs):



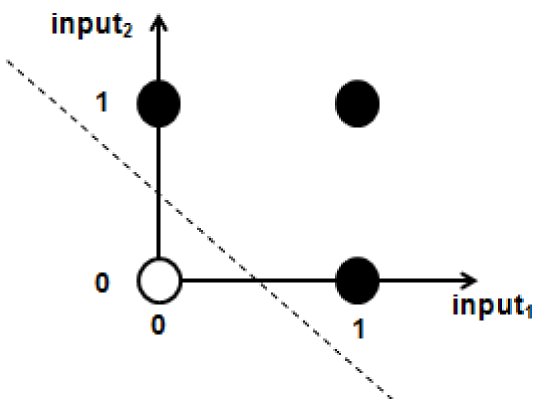
● : 1 output

○ : 0 output

The equation of a 2-dimensional linear separator line is:

$$input_2 = 0.5 - input_1$$

Representation capability of perceptrons



The 2-input **logical OR** function (cont.):

Based on the found separator line, **the points which result in 1 as output** are:

$$input_2 \geq 0.5 - input_1$$

If we order it to the format of the **positive part of the 0-thresholded step function** with the **product of the weight vector and the input vector**, we get:

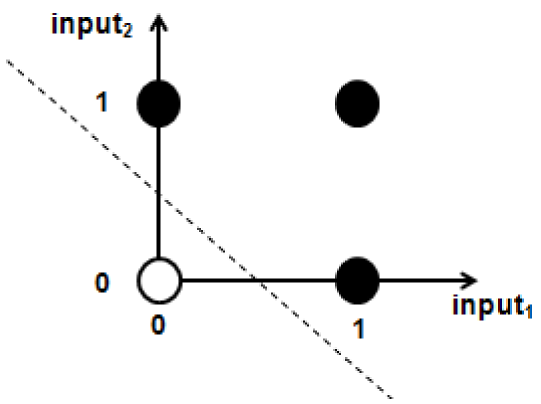
$$-0.5 + input_1 + input_2 \geq 0$$

$$\Downarrow$$

$$\langle 0.5, 1, 1 \rangle * \langle -1, input_1, input_2 \rangle \geq 0$$

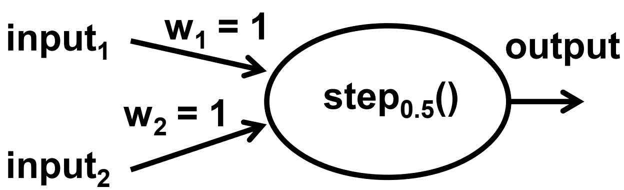
Representation capability of perceptrons

The 2-input **logical OR** function (cont.):

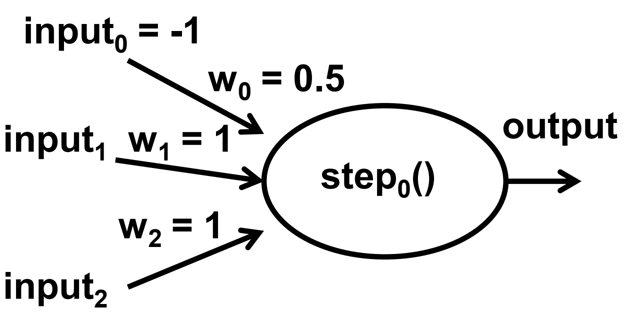


$$\langle 0.5, 1, 1\rangle * \langle -1, input_1, input_2\rangle \geq 0$$

So, the neural network will be, e.g.:



or



Representation capability of perceptrons

What about the **logical XOR** function?

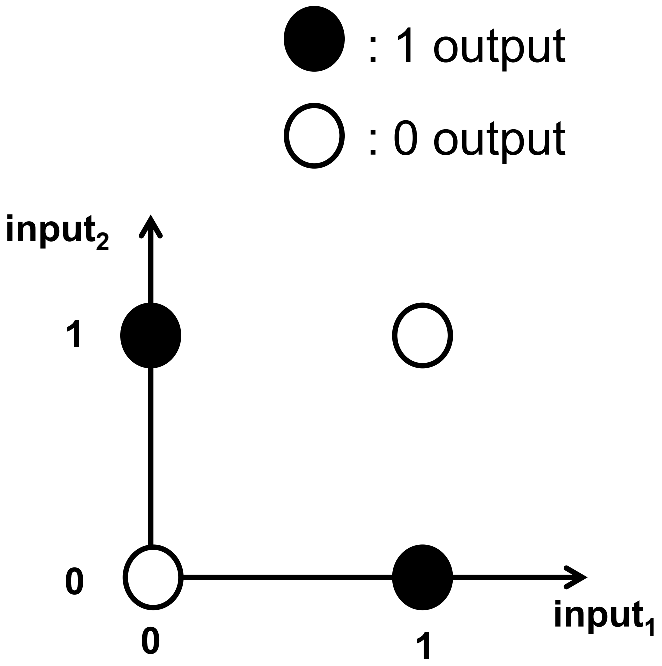
Perceptrons

EFOP-3.4.3-16-2016-00009
A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

Representation capability of perceptrons

What about the **logical XOR** function?

$input_1$	$input_2$	output
0	0	0
0	1	1
1	0	1
1	1	0



This function is **not linearly separable**!

Not representable by a perceptron!

Perceptron learning

It was proven, that **there exists perceptron learning algorithm that is able to teach any linearly separable function** of the inputs – if the training set is big enough.

The main idea behind that is the small **modification of the weights** in such a way that the **difference should decrease between the desired and the real output value**.
This process is repeated several times: the part of the modifications which **changes every weight of the network for each training example** is called:
epoch

In practice usually hundreds of epochs are applied during the training of a neural network.

The update of the weights includes a special variable: **α /learning rate/**
If α is higher, the network is changed more by the current example.

Perceptron learning

The algorithm (for a one-output perceptron)

Step 1. Let us **initialize all the weights** of the 1-layer neural network by a random number **/typically chosen from the interval [-0.5;0.5]/**

Step 2. Repeat

one epoch

Step 2.1. Loop for all the training examples (ex_i)

Step 2.1.1. Calculate the output (O_i^{calc}) for the input ($\boldsymbol{I_i}$) of ex_i

Step 2.1.2. Calculate the error for ex_i as follows:

$$err_i = O_i^{expected} - O_i^{calc}$$

Step 2.1.3. Update the weights between each concerned input (input$_j$) and the output:

$$w_j = w_j + \alpha * input_j * err_i$$

Loop end

until the stopping criteria is not fulfilled

Perceptrons

EFOP-3.4.3-16-2016-00009
A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

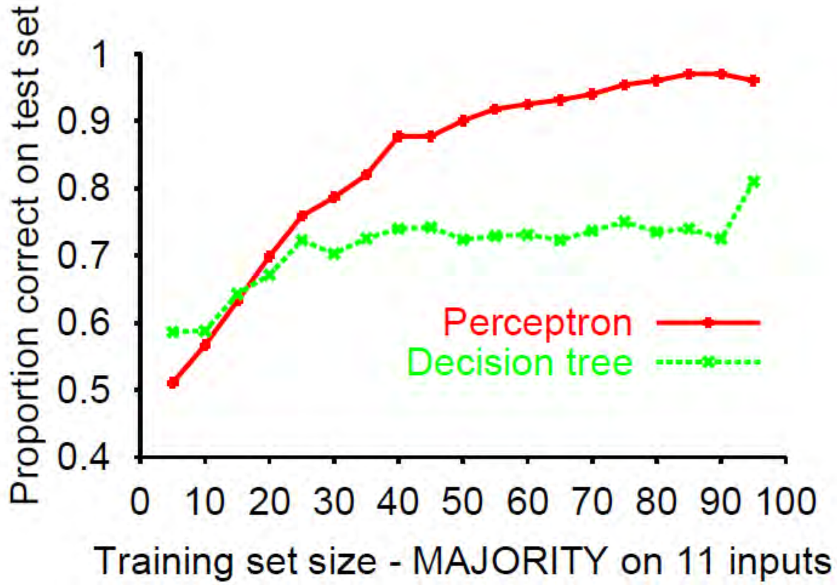
Perceptron learning

The presented perceptron learning algorithm realizes a **gradient descent-**based search in the space of the weights.

⬇

It **ensures the convergence** of the weights to the values that represent the desired – learnable /linearly separable/ – target function **/if α is not too high/**.

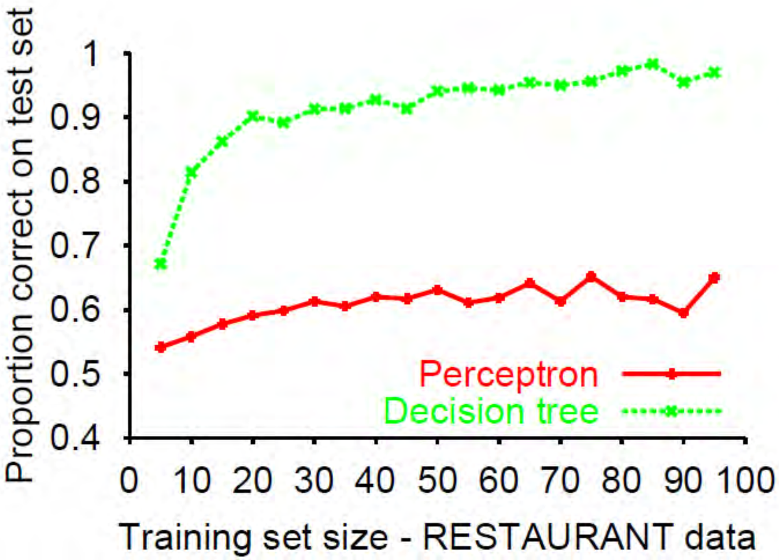
The **comparison** of the learning curves of a **perceptron and a decision tree** while learning the 11-input majority function:



EFOP-3.4.3-16-2016-00009
A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

Perceptron learning

However, the **restaurant example** of the „Decision tree learning”-related lesson **can not be learnt by a perceptron** (it is not linearly separable).
It is able only to find the **hyperplane that separates linearly the best** the examples of different classifications.
This fact is reflected by the following learning curves:



Perceptron learning

Other **difference between perceptron learning** (it is also valid for multi-layer neural networks) **and decision tree learning:**

One input of an example in case of decision trees may have only discrete values (e.g., the „Price" attribute of the restaurant example can has values „$", „$$" or „$$$"). If we intend to **describe the same input** in case of neural networks, we have two options:

Local encoding

One input unit represents one attribute. Different numbers are assigned to different attribute values, e.g., „$"=0.0, „$$"=0.5 and „$$$"=1.0.

Distributed encoding

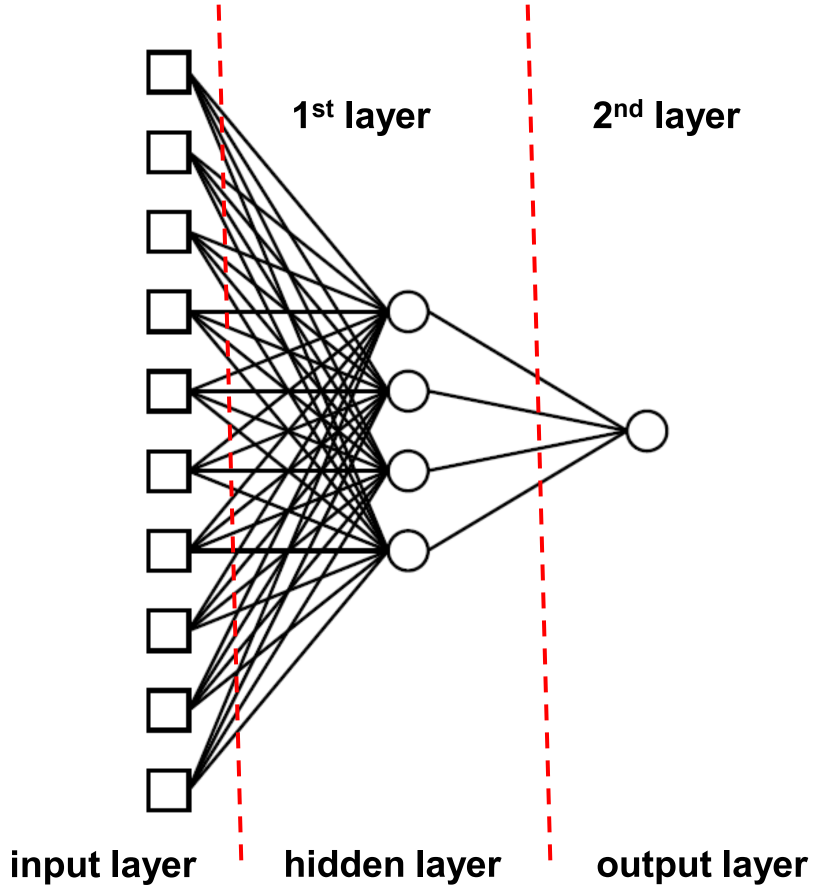
Each possible attribute value has its own input unit. This unit becomes active if and only if the example's input attribute has the value that is assigned to this unit.

MULTILAYER NEURAL NETWORKS

Multilayer Neural Networks

Multilayer neural networks have at least 2 layers, it means that they have **at least one hidden layer**.

An example:



EFOP-3.4.3-16-2016-00009

A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

Representation capabilities of multilayer neural networks

Neural networks with exactly one hidden layer (that is big enough) can represent **any continuous function** of the input.

Neural networks with more than one hidden layer (that are big enough) can represent **any function** of the input.

… and **recall**:

Perceptrons (neural networks without hidden layer) – if they are big enough – can represent **any linear separable function** of the input.

EFOP-3.4.3-16-2016-00009

A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

Example

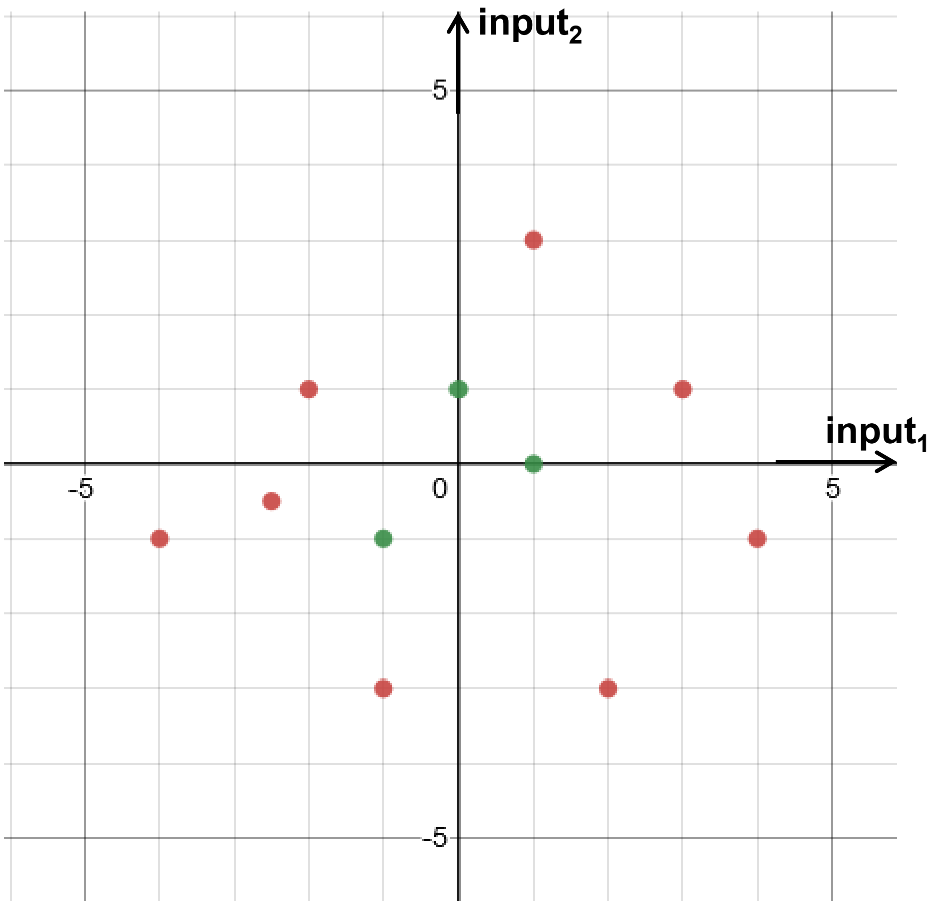
Design a neural network that is **consistent** with the following training set!

$input_1$	$input_2$	output
-4	-1	-1
-2.5	-0.5	-1
-2	1	-1
-1	-1	1
-1	-3	-1
0	1	1
1	0	1
1	3	-1
2	-3	-1
3	1	-1
4	-1	-1

Example

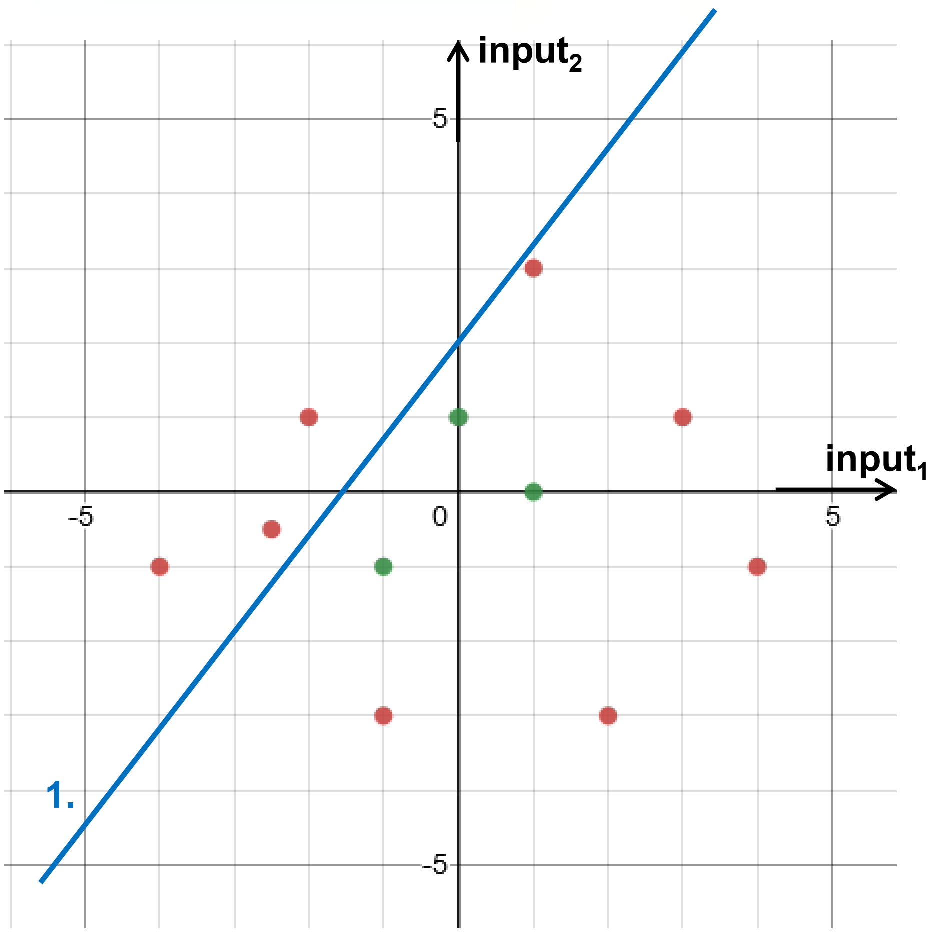
Design a neural network that is **consistent** with the following training set!

$input_1$	$input_2$	output
-4	-1	-1
-2.5	-0.5	-1
-2	1	-1
-1	-1	1
-1	-3	-1
0	1	1
1	0	1
1	3	-1
2	-3	-1
3	1	-1
4	-1	-1



Example

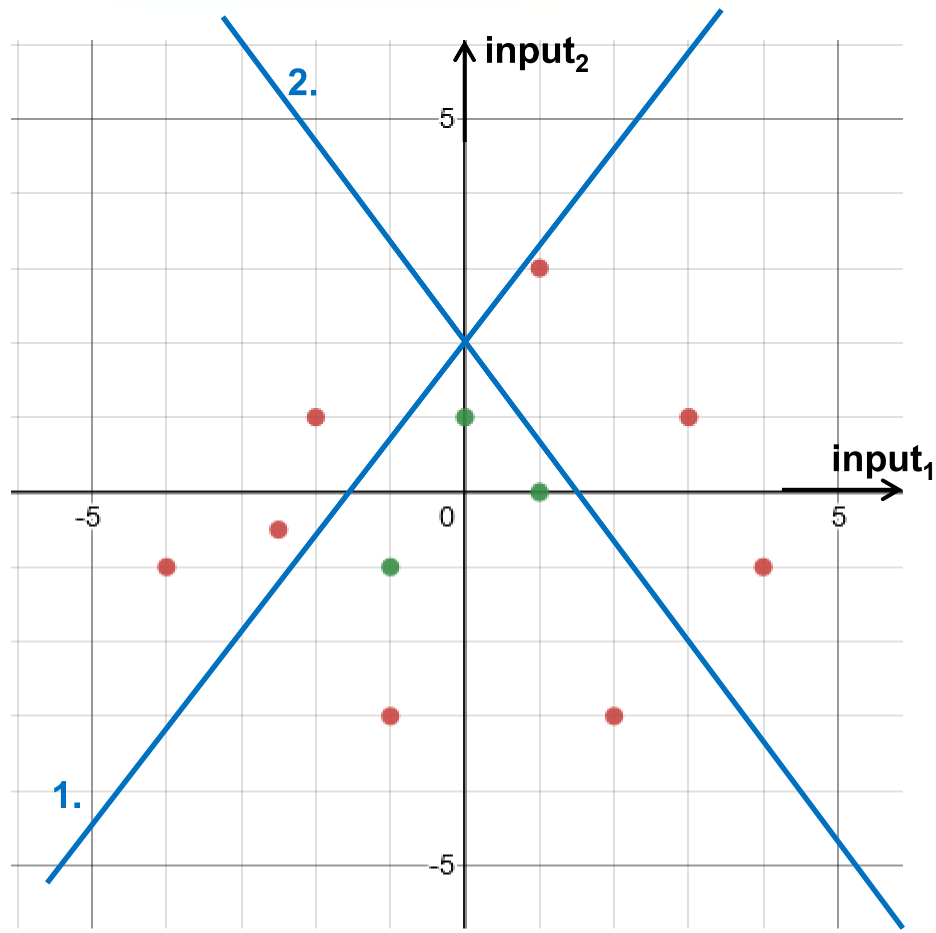
1.: $input_2 = \frac{4}{3} * input_1 + 2$



Example

1.: $input_2 = \frac{4}{3} * input_1 + 2$

2.: $input_2 = -\frac{4}{3} * input_1 + 2$

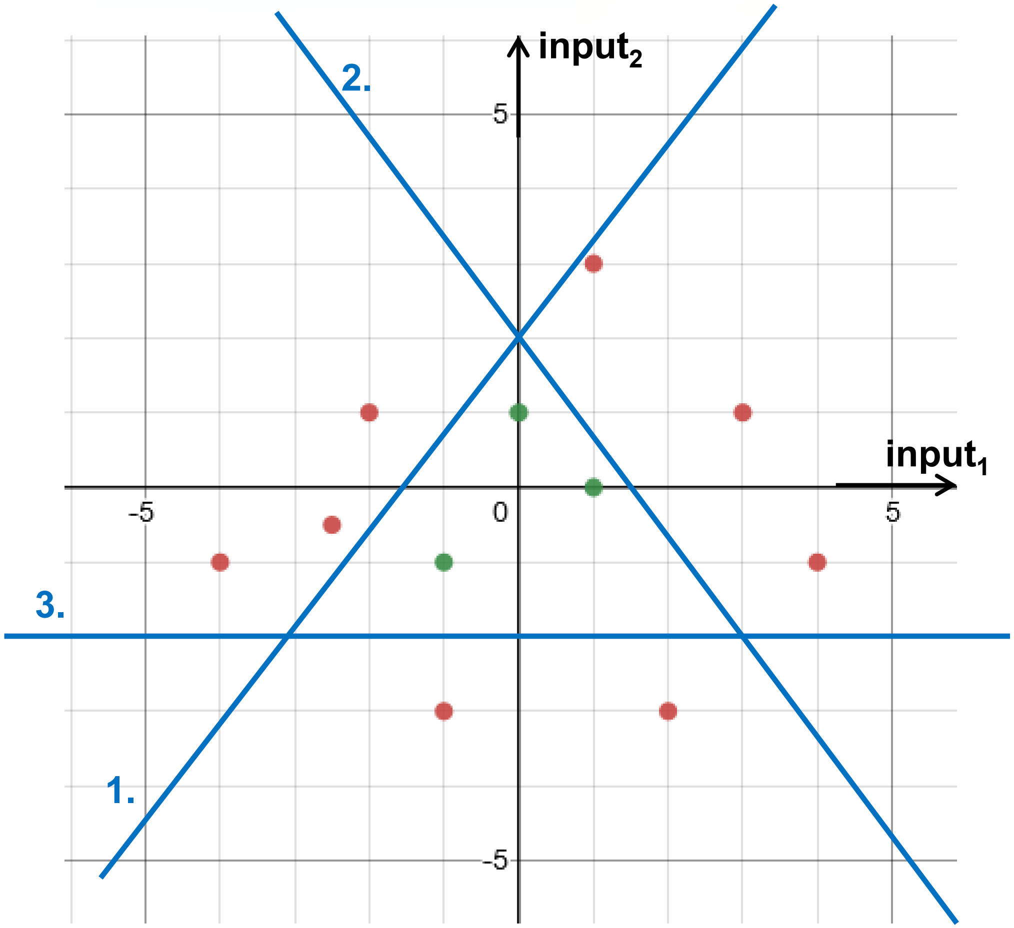


Example

1.: $input_2 = \frac{4}{3} * input_1 + 2$

2.: $input_2 = -\frac{4}{3} * input_1 + 2$

3.: $input_2 = -2$



EFOP-3.4.3-16-2016-00009
A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

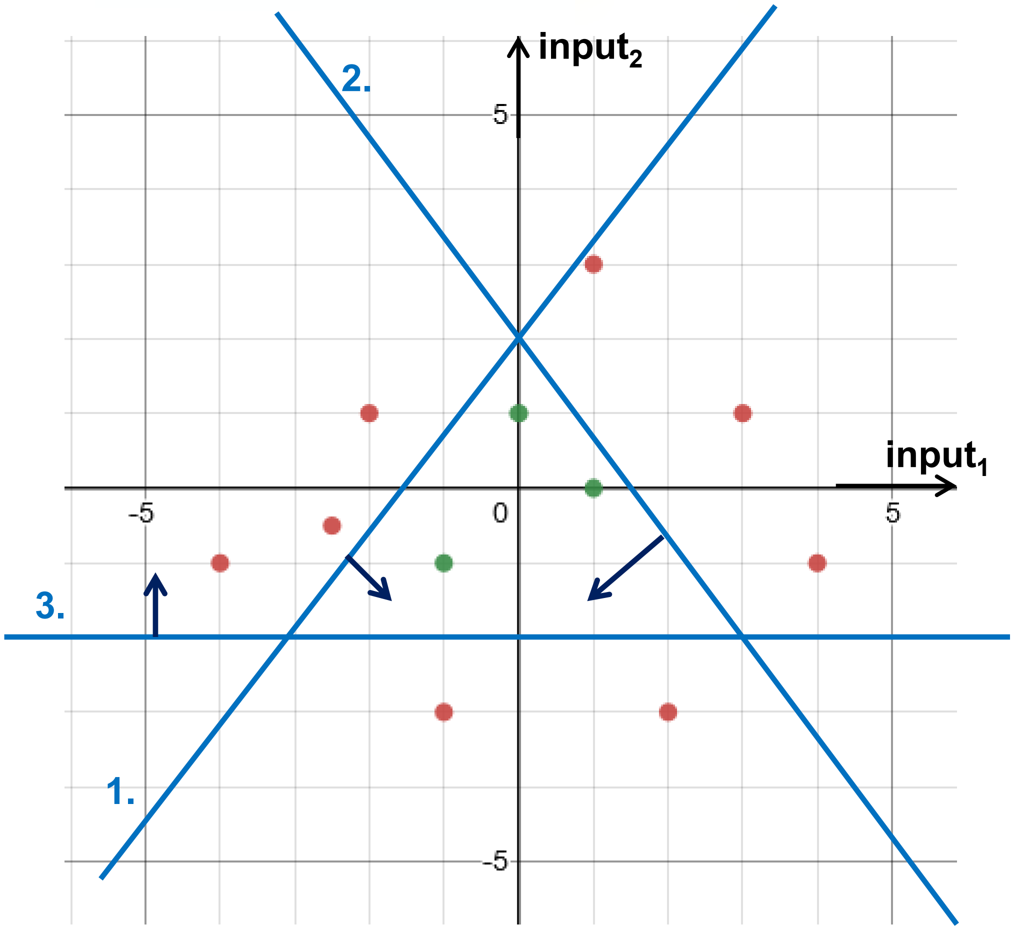
Example

1.: $input_2 < \frac{4}{3} * input_1 + 2$

2.: $input_2 < -\frac{4}{3} * input_1 + 2$

3.: $input_2 > -2$

Logical AND relation is between 1., 2. and 3.



EFOP-3.4.3-16-2016-00009
A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

Example

1.: $input_2 < \frac{4}{3} * input_1 + 2$ ➡ $0 < \frac{4}{3} * input_1 - input_2 + 2$

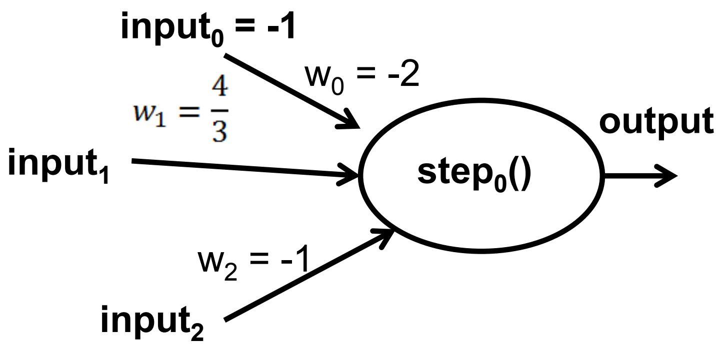
Since the result of this neuron has to be an input of a logical AND relation (0 or 1), we apply **step()** function as **activation function**.

$0 < \frac{4}{3} * input_1 - input_2 + 2$ ➡ The output will be 1, if the input function value is higher than 0.

So, the input function value – as the product of the input vector and the weight vector is:

$$\langle -1, input_1, input_2 \rangle * \langle -2, \frac{4}{3}, -1 \rangle$$

Thus, the appropriate neural network is:



Multilayer Neural Networks

EFOP-3.4.3-16-2016-00009

A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

Example

2.: $input_2 < -\frac{4}{3} * input_1 + 2$ ➡ $0 < -\frac{4}{3} * input_1 - input_2 + 2$

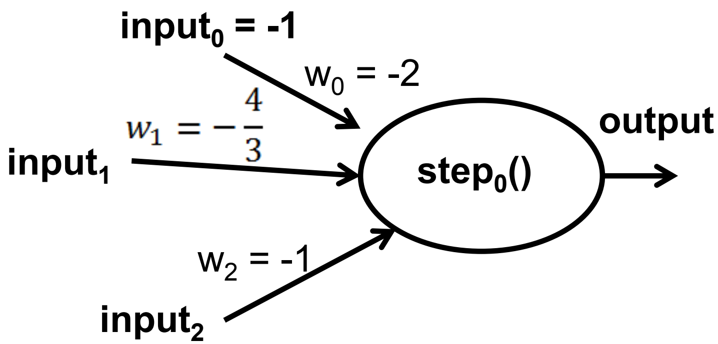
Since the result of this neuron has to be an input of a logical AND relation (0 or 1), we apply **step()** function as **activation function**.

$0 < -\frac{4}{3} * input_1 - input_2 + 2$ ➡ The output will be 1, if the input function value is higher than 0.

So, the input function value – as the product of the input vector and the weight vector is:

$$\langle -1, input_1, input_2 \rangle * \langle -2, -\frac{4}{3}, -1 \rangle$$

Thus, the appropriate neural network is:



EFOP-3.4.3-16-2016-00009
A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

Example

3.: $input_2 > -2 \quad \Rightarrow \quad 0 < input_2 + 2$

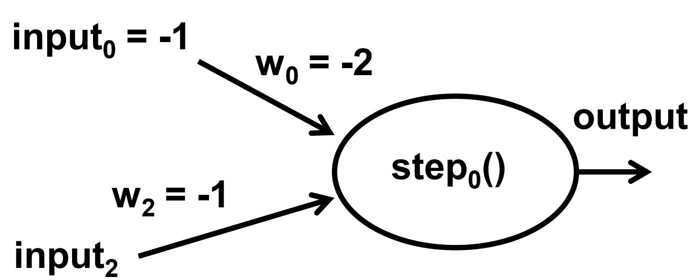
Since the result of this neuron has to be an input of a logical AND relation (0 or 1), we apply **step()** function as **activation function**.

$0 < input_2 + 2 \quad \Rightarrow$ The output will be 1, if the input function value is higher than 0.

So, the input function value – as the product of the input vector and the weight vector is:

$$\langle -1, input_1, input_2 \rangle * \langle -2, 0, -1 \rangle$$

Thus, the appropriate neural network is:



Example

As finally logical AND relation has to be created between 1., 2. and 3., the 2nd layer of the neural network has only one neuron that realizes a **3-input logical AND function**.

Since there are 2 input values of the examples plus there is the bias ($input_0$), the neurons that realize the 3 lines will be numbered as 3,4 and 5, so their output values are, respectively, a_3, a_4 and a_5. They are 0 or 1.

It means, that the 3-input logical AND function can be realized by a neuron that results in 1 if and only if:

$$a_3 + a_4 + a_5 > 2.5 \Longrightarrow a_3 + a_4 + a_5 - 2.5 > 0$$

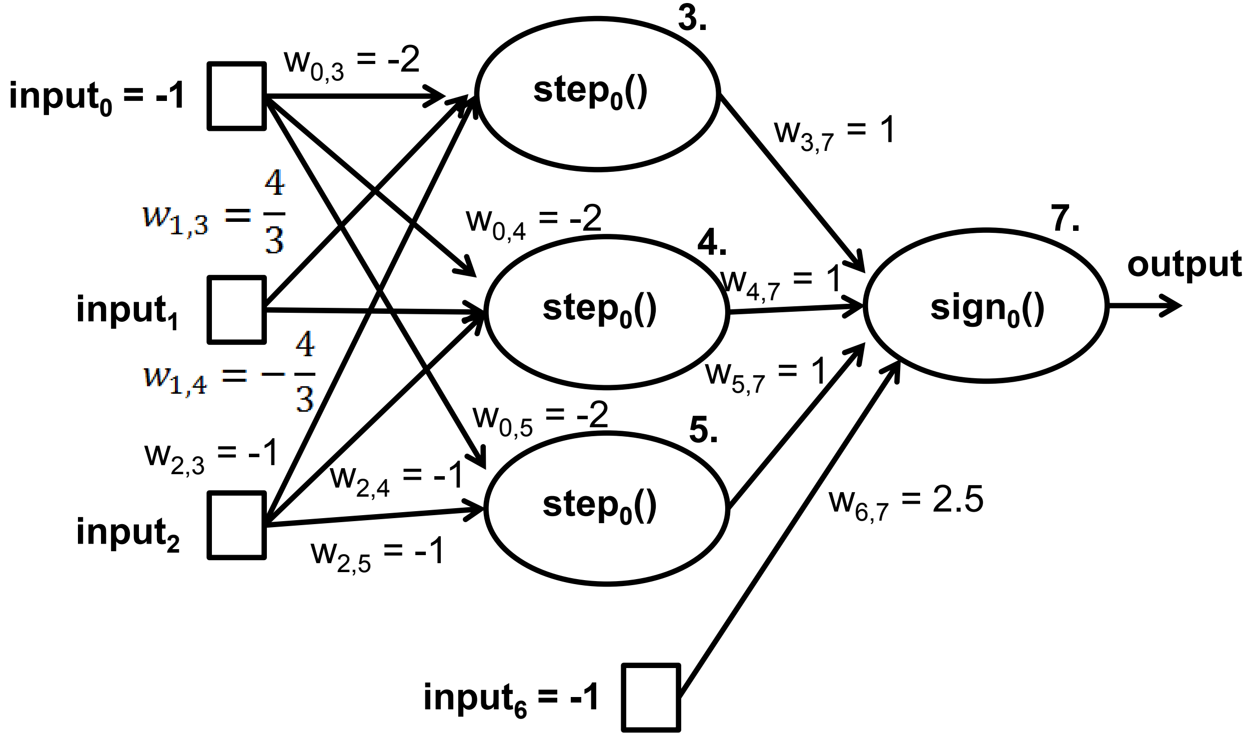
So, the input function value – as the product of the input vector and the weight vector is:

$$\langle -1, a_3, a_4, a_5 \rangle * \langle 2.5, 1, 1, 1 \rangle$$

Since the output values of the examples are 1 and -1, we apply **sign()** function as **activation function of the last layer**.

Example

Thus, **the solution** of the example is the following 2-layer neural network:



EFOP-3.4.3-16-2016-00009
A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

Training of multilayer feed-forward neural networks

The most popular algorithm for training the multilayer neural networks is **back-propagation**.

However, these methods are not effective and **does not guarantee the reaching of global optimum**.

The big differences between perceptrons and multilayer neural networks related to the learning process:

(1) Here, **weights can influence more than one output**.

(2) Moreover, **an error** at one output **may be originated from more than one weight**.

The **basic idea behind** back-propagation:

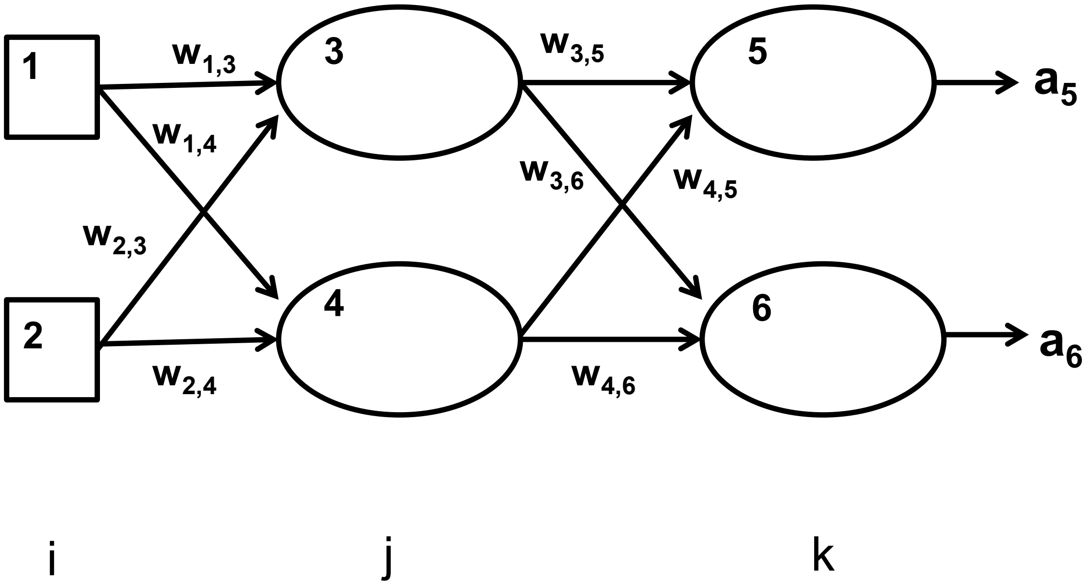
The output **errors are distributed between the weights** in such an extent, how much effect they have on the output.

Training of multilayer feed-forward neural networks

The method to calculate the updated weights - regarding one example - from epoch to epoch is demonstrated on the following 2-layer neural network:

In this example, the following parameter settings are applied:

- every initial weight is 0.2
- $\alpha = 0.5$
- every unit applies *sigmoid()*, as activation function */g()/*



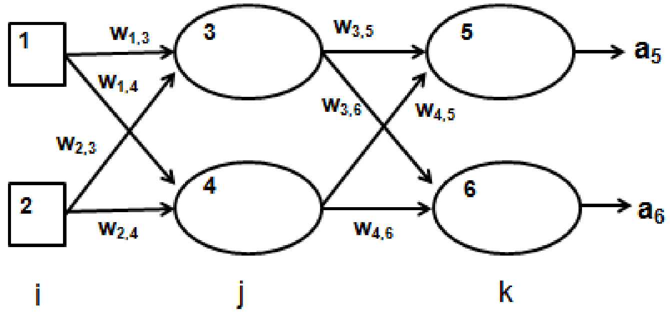
For making it more general, we will denote the layer as follows:

EFOP-3.4.3-16-2016-00009
A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

Training of multilayer feed-forward neural networks

For the calculations squared error is used, for the sake of simplicity **(*n* is the number of outputs)**:

$$E = \frac{1}{2} * \sum_{m=1}^{n} \left(output_m^{expected} - output_m^{calculated}\right)^2$$



We have to modify every weight proportionally to its role in its error. Mathematically, it is expressed by the partal derivative of the error with respect to the considered weight.

Its calculation for the weights between layer *j* and layer *k* is:

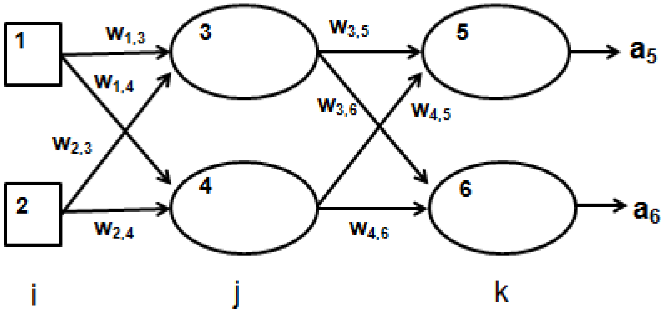
$$\frac{\partial E}{\partial w_{j,k}} =$$

EFOP-3.4.3-16-2016-00009
A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

Training of multilayer feed-forward neural networks

For the calculations squared error is used, for the sake of simplicity **(*n* is the number of outputs)**:

$$E = \frac{1}{2} * \sum_{m=1}^{n} \left(output_m^{expected} - output_m^{calculated}\right)^2$$



We have to modify every weight proportionally to its role in its error. Mathematically, it is expressed by the partal derivative of the error with respect to the considered weight.

Its calculation for the weights between layer *j* and layer *k* is:

$$\frac{\partial E}{\partial w_{j,k}} = -\frac{1}{2} * 2 * \left(output_k^{expected} - a_k\right) * \frac{\partial a_k}{\partial w_{j,k}} =$$

***k* is the index of the output that is influenced by $w_{j,k}$**

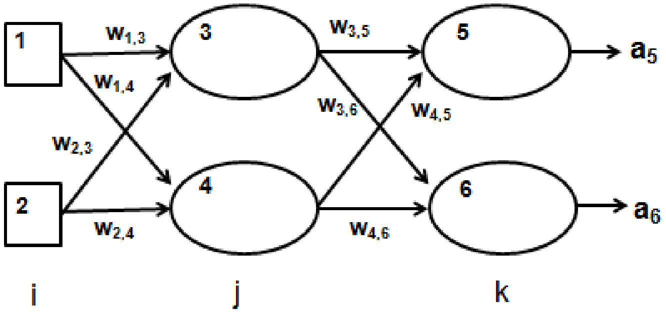
EFOP-3.4.3-16-2016-00009

A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

Training of multilayer feed-forward neural networks

For the calculations squared error is used, for the sake of simplicity **(*n* is the number of outputs)**:

$$E = \frac{1}{2} * \sum_{m=1}^{n} \left(output_m^{expected} - output_m^{calculated}\right)^2$$



$$\frac{\partial E}{\partial w_{j,k}} = -\frac{1}{2} * 2 * \left(output_k^{expected} - a_k\right) * \frac{\partial a_k}{\partial w_{j,k}} =$$

$$= -\left(output_k^{expected} - a_k\right) * \frac{\partial \left(g\left(in_k\right)\right)}{\partial w_{j,k}} =$$

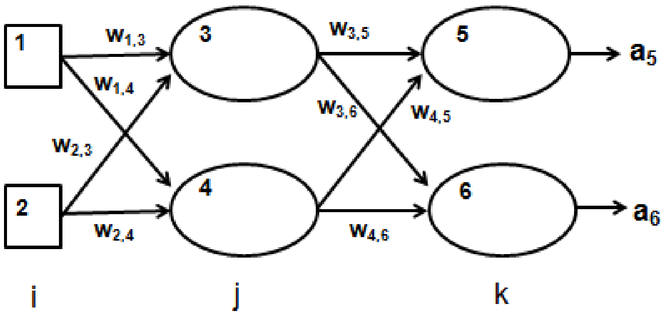
in_k is the input function value of node *k*

E.g., $in_5 = a_3 * w_{3,5} + a_4 * w_{4,5}$

Training of multilayer feed-forward neural networks

For the calculations squared error is used, for the sake of simplicity **(*n* is the number of outputs)**:

$$E = \frac{1}{2} * \sum_{m=1}^{n} \left(output_m^{expected} - output_m^{calculated} \right)^2$$



$$\frac{\partial E}{\partial w_{j,k}} = -\frac{1}{2} * 2 * \left(output_k^{expected} - a_k \right) * \frac{\partial a_k}{\partial w_{j,k}} =$$

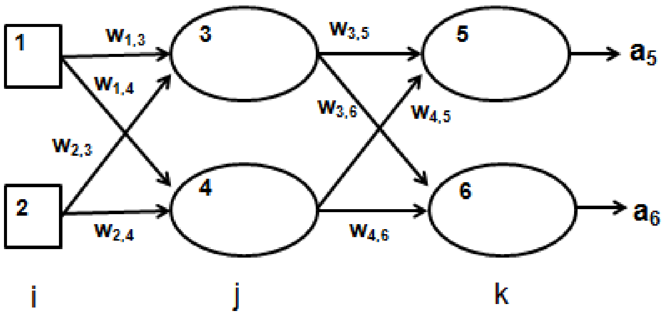
$$= -\left(output_k^{expected} - a_k \right) * \frac{\partial \left(g\left(in_k \right) \right)}{\partial w_{j,k}} =$$

$$= -\left(output_k^{expected} - a_k \right) * g'\left(in_k \right) * \frac{\partial \left(in_k \right)}{\partial w_{j,k}} =$$

Training of multilayer feed-forward neural networks

For the calculations squared error is used, for the sake of simplicity **(*n* is the number of outputs)**:

$$E = \frac{1}{2} * \sum_{m=1}^{n} \left(output_m^{expected} - output_m^{calculated}\right)^2$$

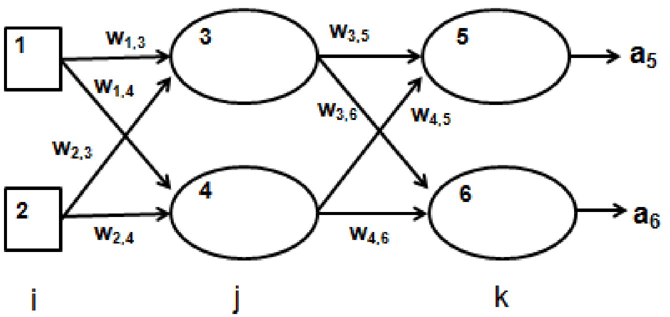


$$\frac{\partial E}{\partial w_{j,k}} = -\left(output_k^{expected} - a_k\right) * g'(in_k) * \frac{\partial (in_k)}{\partial w_{j,k}} =$$

$$= -\left(output_k^{expected} - a_k\right) * g'(in_k) * \frac{\partial \left(\sum_j w_{j,k} * a_j\right)}{\partial w_{j,k}} =$$

Training of multilayer feed-forward neural networks

For the calculations squared error is used, for the sake of simplicity **(*n* is the number of outputs)**:



$$E = \frac{1}{2} * \sum_{m=1}^{n} \left(output_m^{expected} - output_m^{calculated}\right)^2$$

$$\frac{\partial E}{\partial w_{j,k}} = -\left(output_k^{expected} - a_k\right) * g'(in_k) * \frac{\partial (in_k)}{\partial w_{j,k}} =$$

$$= -\left(output_k^{expected} - a_k\right) * g'(in_k) * \frac{\partial\left(\sum_j w_{j,k} * a_j\right)}{\partial w_{j,k}} =$$

$$= -\left(output_k^{expected} - a_k\right) * g'(in_k) * a_j$$

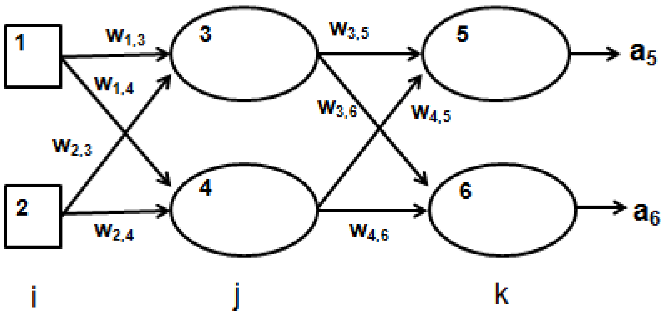
Multilayer Neural Networks

EFOP-3.4.3-16-2016-00009
A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

Training of multilayer feed-forward neural networks

For the calculations squared error is used, for the sake of simplicity **(*n* is the number of outputs)**:

$$E = \frac{1}{2} * \sum_{m=1}^{n} \left(output_m^{expected} - output_m^{calculated}\right)^2$$



$$\frac{\partial E}{\partial w_{j,k}} = \underbrace{-\left(output_k^{expected} - a_k\right) * g'(in_k)}_{\delta_k} * a_j$$

$w_{j,k}^{(x)}$: $w_{j,k}$ in the x. epoch

$$\boxed{w_{j,k}^{(2)} = w_{j,k}^{(1)} \pm \alpha * \delta_k * a_j}$$

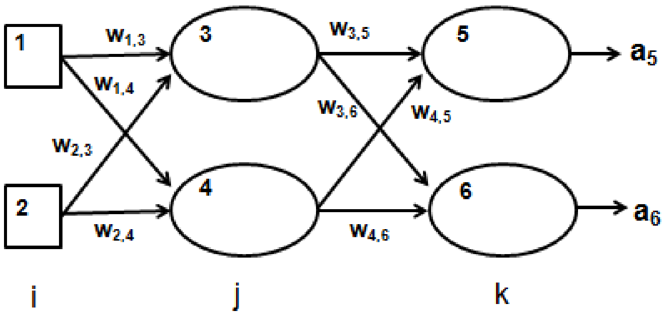
, where

$$\delta_k = \left(output_k^{expected} - a_k\right) * g'(in_k).$$

EFOP-3.4.3-16-2016-00009
A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

Training of multilayer feed-forward neural networks

The update rule for the weights between layer *i* and layer *j*:



Remember, that

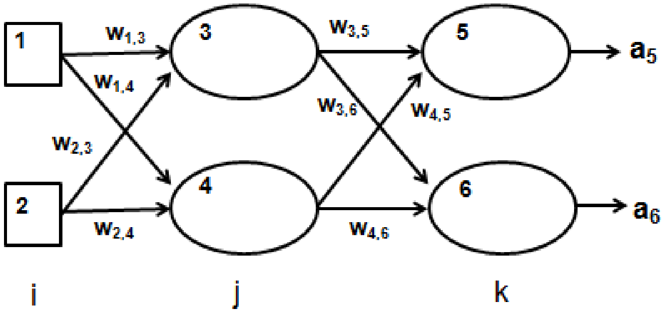
$$E = \frac{1}{2} * \sum_{m=1}^{n} \left(output_m^{expected} - output_m^{calculated}\right)^2$$

$$\frac{\partial E}{\partial w_{i,j}} =$$

EFOP-3.4.3-16-2016-00009
A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

Training of multilayer feed-forward neural networks

The update rule for the weights between layer *i* and layer *j*:



Remember, that

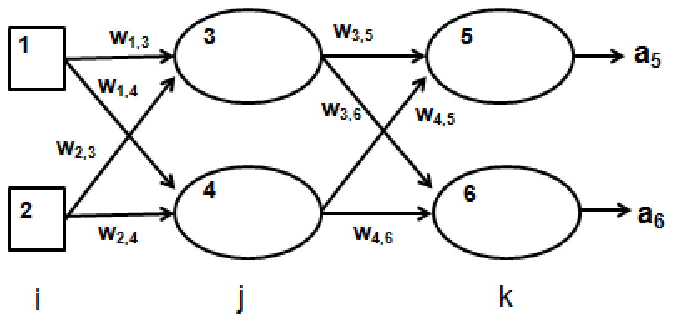
$$E = \frac{1}{2} * \sum_{m=1}^{n} \left(output_m^{expected} - output_m^{calculated}\right)^2$$

$$\frac{\partial E}{\partial w_{i,j}} = -\frac{1}{2} * 2 * \sum_k \left(output_k^{expected} - a_k\right) * \frac{\partial a_k}{\partial w_{i,j}} =$$

EFOP-3.4.3-16-2016-00009
A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

Training of multilayer feed-forward neural networks

The update rule for the weights between layer *i* and layer *j*:



Remember, that

$$E = \frac{1}{2} * \sum_{m=1}^{n} \left(output_m^{expected} - output_m^{calculated}\right)^2$$

$$\frac{\partial E}{\partial w_{i,j}} = -\frac{1}{2} * 2 * \sum_k \left(output_k^{expected} - a_k\right) * \frac{\partial a_k}{\partial w_{i,j}} =$$

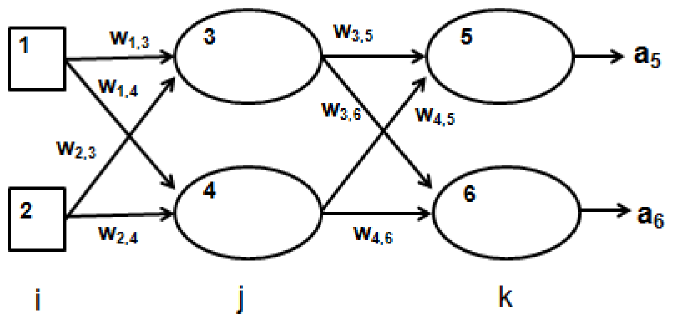
$$= -\sum_k \left(output_k^{expected} - a_k\right) * \frac{\partial g(in_k)}{\partial w_{i,j}} =$$

Multilayer Neural Networks

EFOP-3.4.3-16-2016-00009
A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

Training of multilayer feed-forward neural networks

$$E = \frac{1}{2} * \sum_{m=1}^{n} \left(output_m^{expected} - output_m^{calculated}\right)^2$$



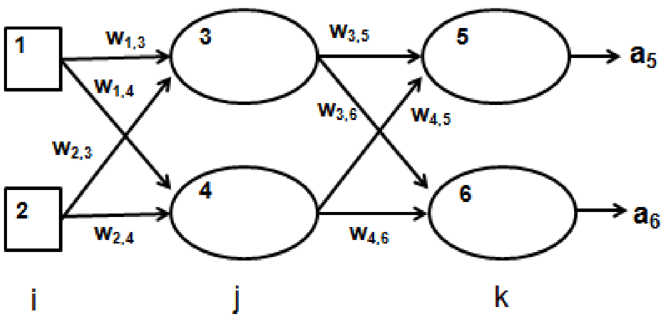
$$\frac{\partial E}{\partial w_{i,j}} = -\sum_k \left(output_k^{expected} - a_k\right) * \frac{\partial g(in_k)}{\partial w_{i,j}} =$$

$$= -\sum_k \left(output_k^{expected} - a_k\right) * g'(in_k) * \frac{\partial\left(\sum_j w_{j,k} * a_j\right)}{\partial w_{i,j}} =$$

EFOP-3.4.3-16-2016-00009
A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

Training of multilayer feed-forward neural networks

$$E = \frac{1}{2} * \sum_{m=1}^{n} \left(output_m^{expected} - output_m^{calculated}\right)^2$$



$$\frac{\partial E}{\partial w_{i,j}} = -\sum_k \left(output_k^{expected} - a_k\right) * \frac{\partial g(in_k)}{\partial w_{i,j}} =$$

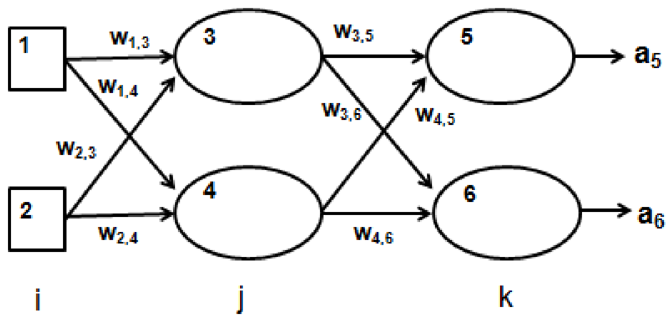
$$= -\sum_k \left(output_k^{expected} - a_k\right) * g'(in_k) * \frac{\partial \left(\sum_j w_{j,k} * a_j\right)}{\partial w_{i,j}} =$$

$$= -\sum_k \delta_k * w_{j,k} * \frac{\partial a_j}{\partial w_{i,j}} =$$

Remember: $\delta_k = \left(output_k^{expected} - a_k\right) * g'(in_k)$

Training of multilayer feed-forward neural networks

$$E = \frac{1}{2} * \sum_{m=1}^{n} \left(output_m^{expected} - output_m^{calculated}\right)^2$$



$$\frac{\partial E}{\partial w_{i,j}} = -\sum_k \delta_k * w_{j,k} * \frac{\partial a_j}{\partial w_{i,j}} =$$

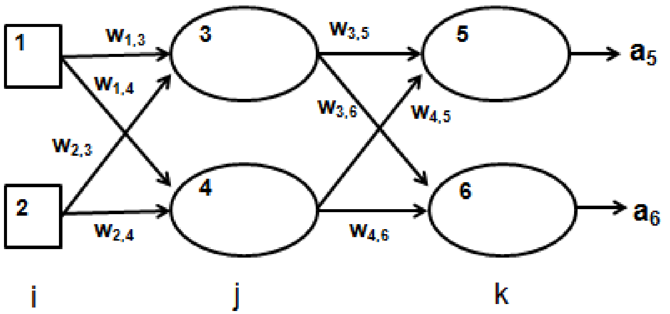
$$= -\sum_k \delta_k * w_{j,k} * \frac{\partial\left(g\left(in_j\right)\right)}{\partial w_{i,j}} =$$

Multilayer Neural Networks

EFOP-3.4.3-16-2016-00009
A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

Training of multilayer feed-forward neural networks

$$E = \frac{1}{2} * \sum_{m=1}^{n} \left(output_m^{expected} - output_m^{calculated}\right)^2$$



$$\frac{\partial E}{\partial w_{i,j}} = -\sum_k \delta_k * w_{j,k} * \frac{\partial a_j}{\partial w_{i,j}} =$$

$$= -\sum_k \delta_k * w_{j,k} * \frac{\partial\left(g\left(in_j\right)\right)}{\partial w_{i,j}} =$$

$$= -\sum_k \delta_k * w_{j,k} * g'\left(in_j\right) * \frac{\partial\left(\sum_i w_{i,j} * a_i\right)}{\partial w_{i,j}} =$$

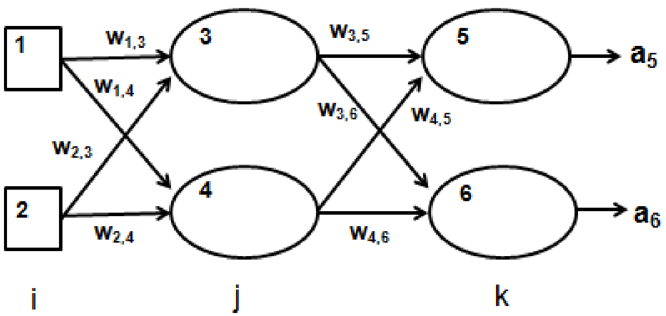
Multilayer Neural Networks

EFOP-3.4.3-16-2016-00009

A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

Training of multilayer feed-forward neural networks

$$E = \frac{1}{2} * \sum_{m=1}^{n} \left(output_m^{expected} - output_m^{calculated}\right)^2$$



$$\frac{\partial E}{\partial w_{i,j}} = -\sum_k \delta_k * w_{j,k} * g'\left(in_j\right) * \frac{\partial\left(\sum_i w_{i,j} * a_i\right)}{\partial w_{i,j}} =$$

$$= -\sum_k \delta_k * w_{j,k} * g'\left(in_j\right) * a_i$$

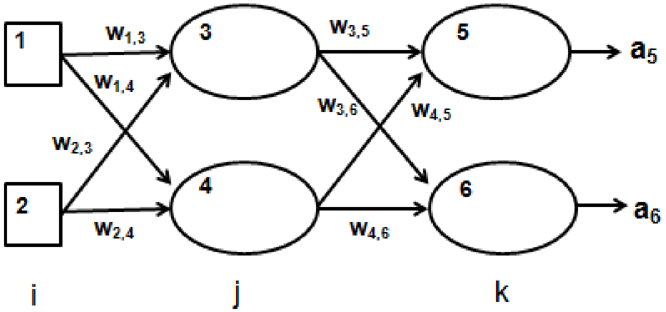
Multilayer Neural Networks

EFOP-3.4.3-16-2016-00009

A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

Training of multilayer feed-forward neural networks

$$E = \frac{1}{2} * \sum_{m=1}^{n} \left(output_m^{expected} - output_m^{calculated} \right)^2$$



$$\frac{\partial E}{\partial w_{i,j}} = \underbrace{-\sum_{k} \delta_k * w_{j,k} * g'(in_j)}_{\Delta_j} * a_i$$

$w_{i,j}^{(x)}$: $w_{i,j}$ in the x. epoch

$$\boxed{w_{i,j}^{(2)} = w_{i,j}^{(1)} \pm \alpha * \Delta_j * a_i}$$

, where $\Delta_j = \underbrace{\sum_{k} \delta_k * w_{j,k} * g'(in_j)}_{\text{weighted error for a node in layer } j}$

weighted error for a node in layer *j*

Training of multilayer feed-forward neural networks

It is worth to note that back-propagation requires the derivation of the activation function.

⬇

We can not use **step() or sign()** functions as activation functions, since they **are not continuous** functions, they can not be derivated.

If we apply sigmoid() function to be the activation function, it has the advantage, that:

$$sigmoid(x)' = sigmoid(x) * (1 - sigmoid(x))$$

EFOP-3.4.3-16-2016-00009

A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

THANK YOU FOR THE ATTENTION!

Reference:

Stuart J. Russel – Peter Norvig:
Artificial Intelligence: A Modern Approach,
Prentice Hall, 2010, ISBN 0136042597

http://aima.cs.berkeley.edu/

SZÉCHENYI 2020

MAGYARORSZÁG KORMÁNYA

Európai Unió
Európai Szociális Alap

BEFEKTETÉS A JÖVŐBE

EFOP-3.4.3-16-2016-00009

A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

ARTIFICIAL INTELLIGENCE

9-10. Robotics

Authors:

Tibor Dulai, Ágnes Werner-Stark

SZÉCHENYI 2020

MAGYARORSZÁG KORMÁNYA

Európai Unió
Európai Szociális Alap

BEFEKTETÉS A JÖVŐBE

A robot is …

EFOP-3.4.3-16-2016-00009

A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

“A programmable, multifunction manipulator designed to move material, parts, tools or specific devices through variable programmed motions for the performance of a variety of tasks”

(Robot Institute of America)

“An active artificial agent whose environment is the physical world”

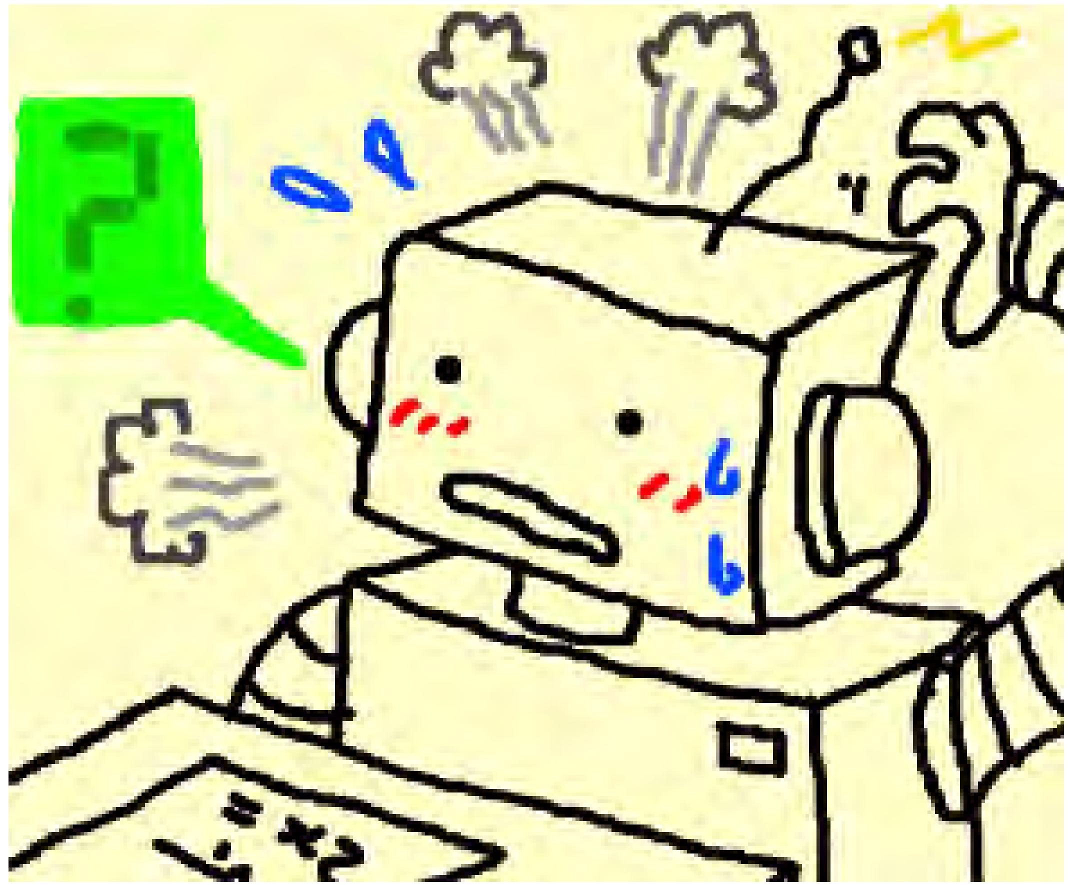
(Russel and Norvig)

The real (physical) world is …

EFOP-3.4.3-16-2016-00009

A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

- **inaccessible,**
- **nondeterministic,**
- **nonepisodic,**
- **dynamic,**
- **continuous.**



Where we are nowadays

https://www.youtube.com/watch?v=fRj34o4hN4I

https://www.youtube.com/watch?v=TxobtWAFh8o

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

Application areas

(1) Industrial production

Repeating tasks, usually without intelligence

(2) Material handling

Storing, transferring, moving materials (barcode helps in identification).
Challanges: irregular-shaped materials (pallet helps), strange structure (e.g., food).

Application areas

(3) Mobots (moving robots)

- collision avoidance
- 0-24 available
- reliable
- traceable
- does not loose any mail/pack

Application areas

EFOP-3.4.3-16-2016-00009

A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

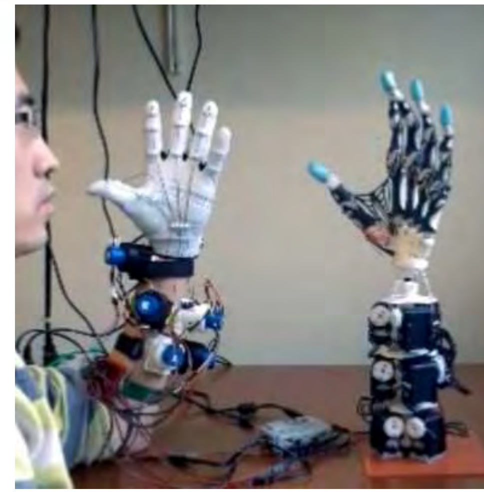
(4) Dangerous circumstances

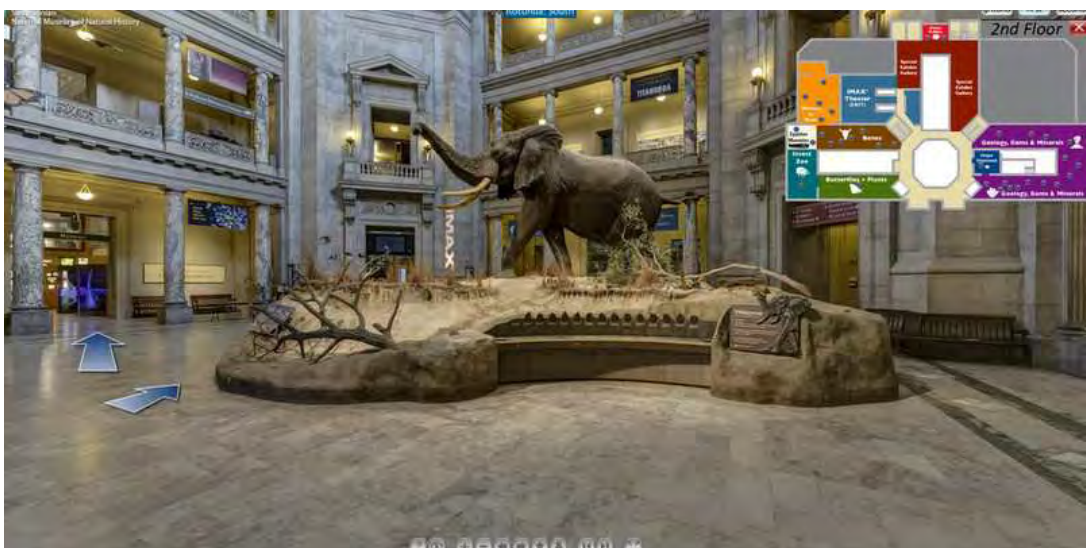
Earthquakes, fire, toxic smoke, radioactivity, toxic waste, mining, deep-sea work, biological hazard

Examples for challanges: coarse ground, lack of light, taking care of unconscious people

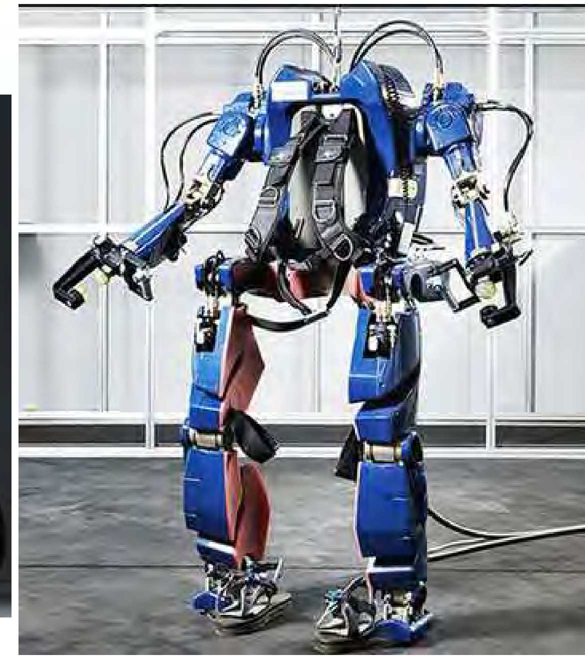
Application areas

(5) Telepresence, virtual reality (VR)





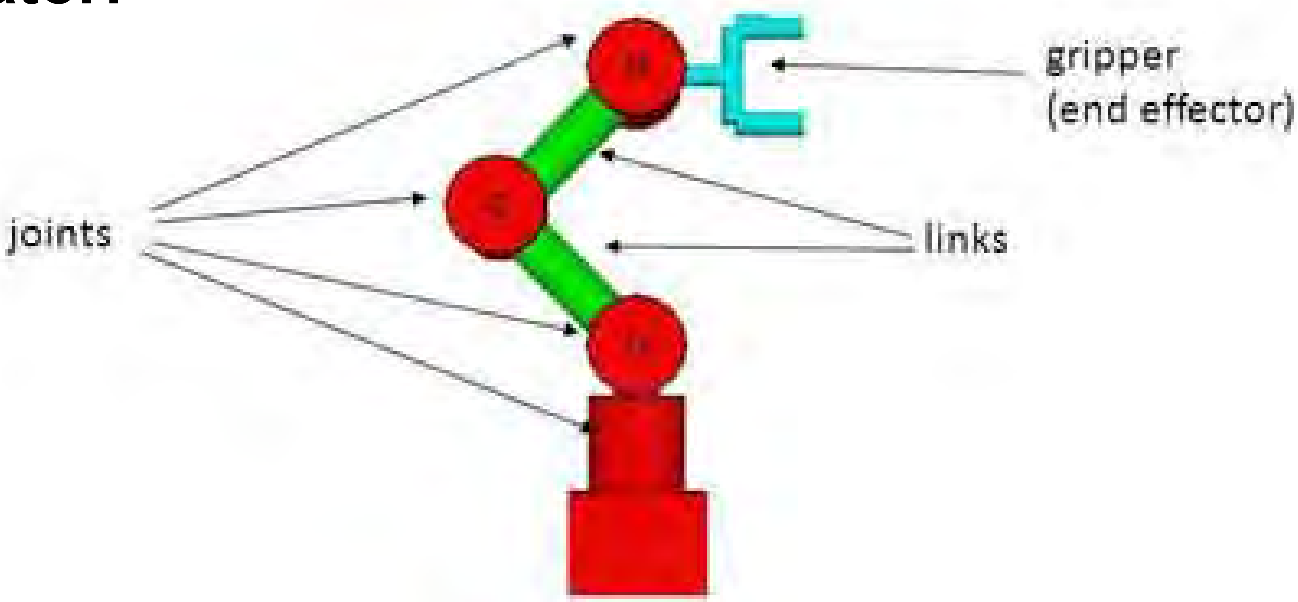
(6) Extension of human capabilities



Parts of a robot

EFOP-3.4.3-16-2016-00009
A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

Manipulator:



An **end effector** can be for example: gripper, sucker, screw-driver, torch, airbrush

Sensors can be, e.g., cameras, infrared sensors, radars, sound-locators, accelerometers

Effectors

Each part of a robot which has any effect on its environment by the control of the robot.

Actuators transform software commands to physical movements:

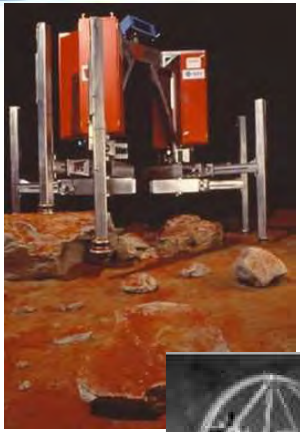
Electro-motors, hydraulic or pneumatic power cylinders

For the sake of simplicity we assume that one actuator is responsible only for one degree of freedom.

Effectors can be used for (1) locomotion or (2) manipulation.

Locomotion

EFOP-3.4.3-16-2016-00009
A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen



by legs: rarely used, e.g., for moving on irregular ground or between huge obstacles.

- **Statically stable:** can stop any time without falling (slow and energy-inefficient)
- **Dynamically stable:** stable only in motion (e.g., by hopping)

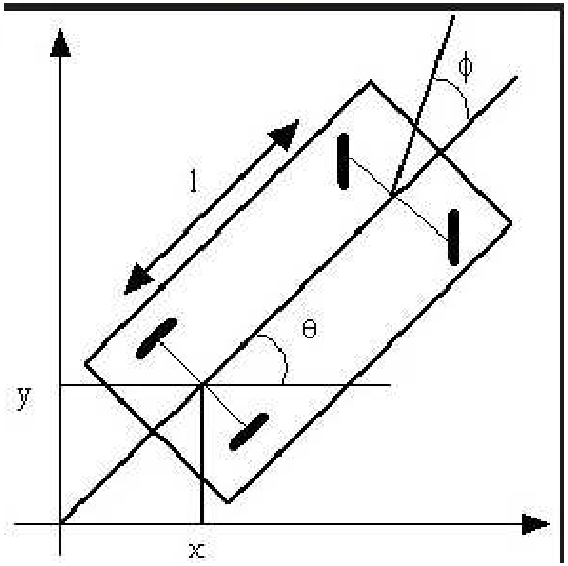
by wheels or special surface for locomotion: more effective and easier to control.

- **Holonomic robot:** the overall number of degree of freedom is equal to the number of controllable degree of freedom
- **Nonholonomic robot:** the overall number of degree of freedom is not equal to the number of controllable degree of freedom

Holonomic vs. Nonholonomic robots I.

EFOP-3.4.3-16-2016-00009

A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen



Overall number of degree of freedom: 3 (x, y, phi)

Controllable degree of freedom: 2 (driving, steering)

The higher the difference between the number of the overall and the controllable degree of freedom is, **the harder is to control** the robot.

e.g., truck trailer: 4 vs. 2

To build nonholonomic robots is easier.

The main difference is: does the robot need a minimal path for turning or is it able to turn around its own angle?

Holonomic vs. Nonholonomic robots II.

EFOP-3.4.3-16-2016-00009
A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

The shortest path between A and B:

Holonomic robots:

Turn towards B and go straight there

Nonholonomic robots:

Straight line segments connected to each other by arcs of miminal radius

Sometimes the **construction of the robot causes hardness in the planning of locomotion.**

Present control theory results make it possible to **drive and steer autonomous vehicles with arbitrary number of trailers of arbitrary sizes.**

Manipulation

EFOP-3.4.3-16-2016-00009

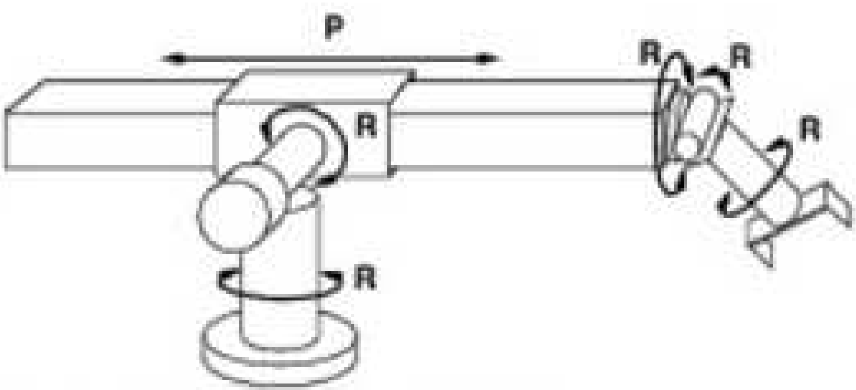
A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

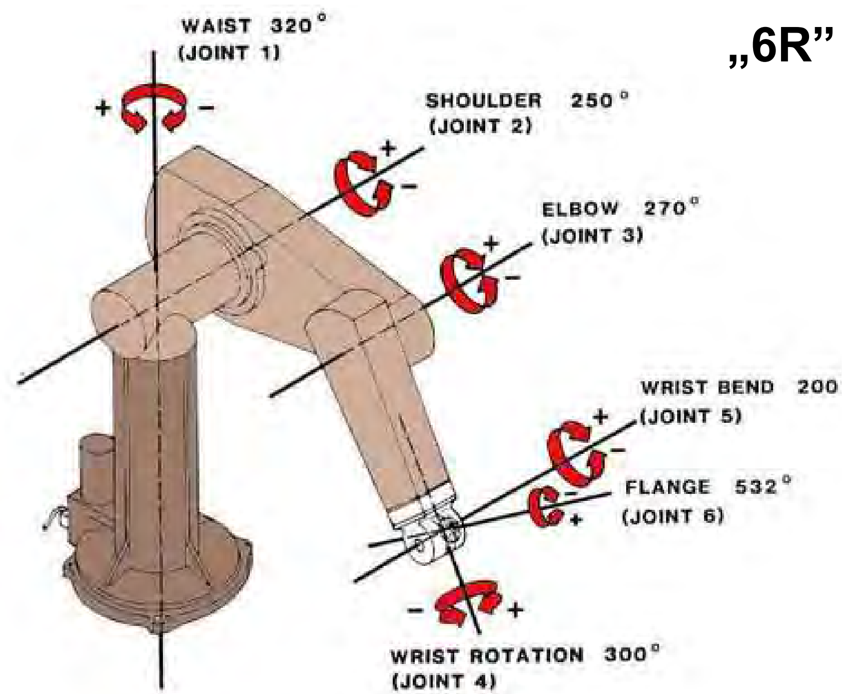
Manipulator: Effectors which move objects

Kinematics: Science which deals with the relation between the movement of actuators of a mechanism and the resulted movement of the different parts of the mechanism.

Manimulator movement types:

- rotary
- prismatic





„6R”

A robot needs at least 6 joints to reach arbitrary position in space.

Sensors – proprioceptive sensing

EFOP-3.4.3-16-2016-00009
A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

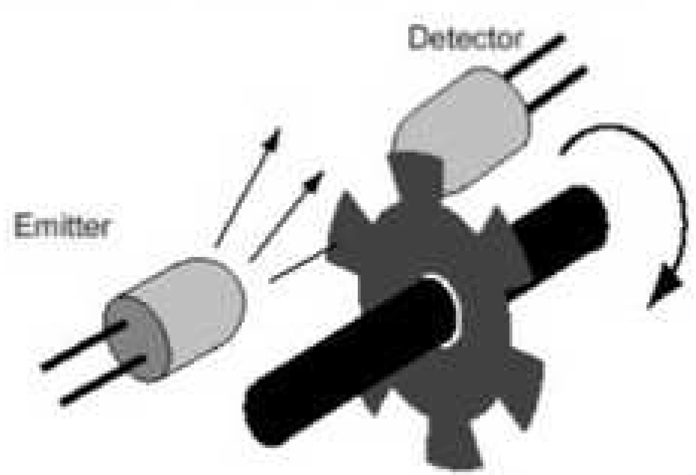
The robot „knows" what the state (location) of its joints is.

Encoders of the joints have information about their angle and size.

The change of the position can be calculated, e.g., by measuring the turn of the wheel or by counting the steps of a stepping motor.

⬇

E.g., slips can cause positioning failure



Precision can be improved for example by applying a magnetic compass or a gyroscope system or by accelerometer.

Sensors – Force sensors

EFOP-3.4.3-16-2016-00009
A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

Force can be adjusted by controlling the current of electromotors**.**

Usually **applied between the manipulator and the effector.**

Compliant motion: the robot moves along a surface while it maintains the connection with constant pressure

Sensors – Tactile sensing

By measuring the distortion of an elastic material

Tactile sensors are able to **sense oscillation**, too.

⬇

It **signs the need for adjusting force.**

Sensors – Sonar (Sound Navigation and Ranging)

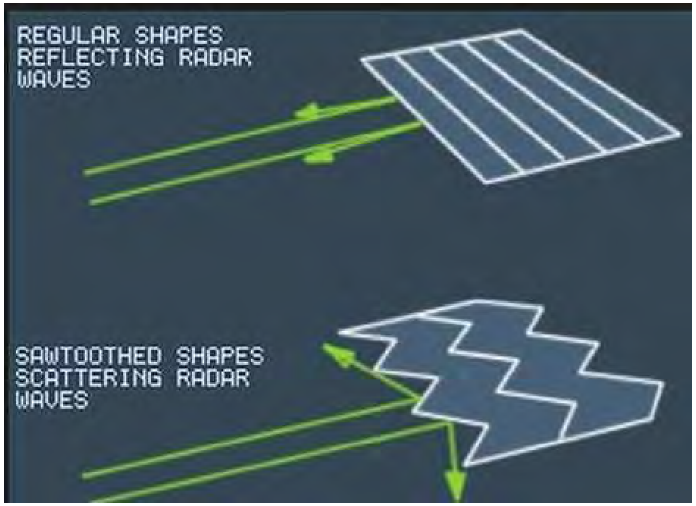
EFOP-3.4.3-16-2016-00009
A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

The impulses are about 50 kHz → 7 mm wavelength

The velocity of the sound is about 330 m/s

Problems:

- **delay**
- **beamwidth** (around 10° or more)
- **relative big wavelength** (smooth, mirroring surfaces)



⬇

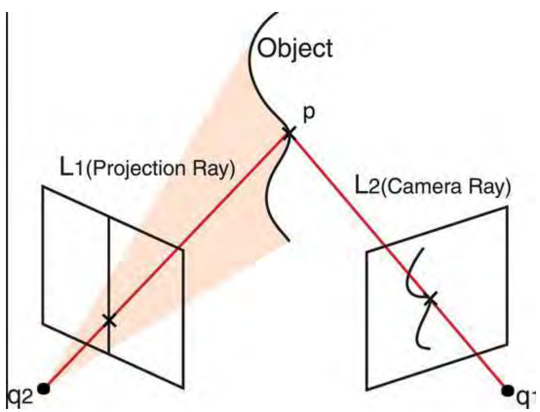
Bayesian (probabilistic) model
can be used

Sensors – Cameras

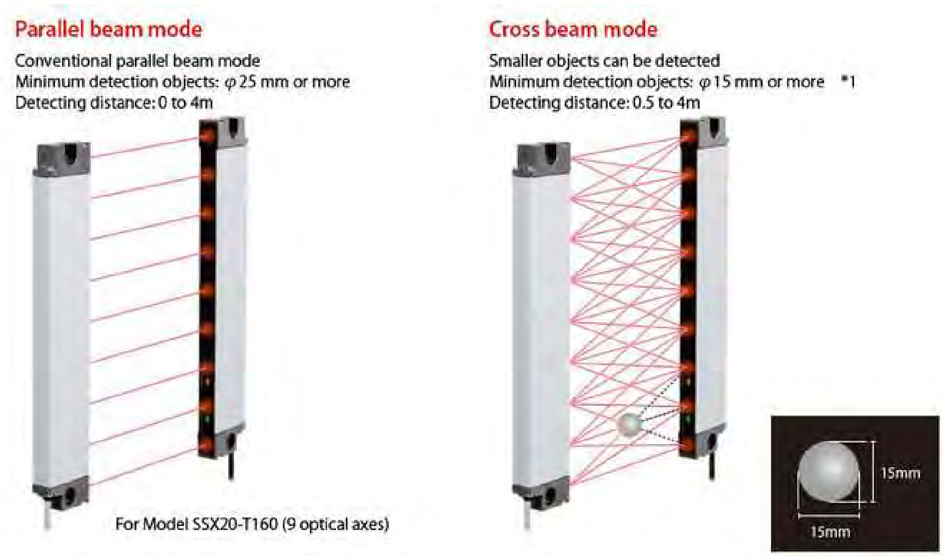
Domain constraints help **fill the gap between the real vision and the vision of robots.**

Robot vision (plus positioning) can be supported, e.g., by barcodes at predetermined places

Structured light sensors:



Applying the same for 3D image construction:



EFOP-3.4.3-16-2016-00009

A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

THANK YOU FOR THE ATTENTION!

Reference:

Stuart J. Russel – Peter Norvig:
Artificial Intelligence: A Modern Approach,
Prentice Hall, 2010, ISBN 0136042597

http://aima.cs.berkeley.edu/

SZÉCHENYI 2020

MAGYARORSZÁG KORMÁNYA

Európai Unió
Európai Szociális Alap

BEFEKTETÉS A JÖVŐBE

EFOP-3.4.3-16-2016-00009

A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

ARTIFICIAL INTELLIGENCE

11-12. Autonomous vehicles

Authors:

Tibor Dulai, Ágnes Werner-Stark

SZÉCHENYI 2020

MAGYARORSZÁG KORMÁNYA

Európai Unió
Európai Szociális Alap

BEFEKTETÉS A JÖVŐBE

What is an autonomous vehicle?

EFOP-3.4.3-16-2016-00009

A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

An autonomous vehicle (also known as a driverless vehicle, self-driving vehicle, robotic vehicle): is a vehicle that is capable of **sensing its environment and navigating without human input**.

Many such systems are evolving, however, till **2017 no cars permitted on public roads were fully autonomous**. They all require a human at the wheel who must be ready to take control at any time.

Autonomous cars use a variety of techniques to detect their surroundings, such as **radar, laser light, GPS, odometry and computer vision**.

The first attempts at truly autonomous cars

In the figure we can see **Navlab autonomous cars 1 through 5**. Navlab 1 (farthest in photo) was started in 1984 and completed in 1986. **Navlab 5** (closest vehicle), finished in 1995, **was the first car to drive coast-to-coast** (USA) autonomously.

Advantages of Driverless cars

- There would be **no bad drivers and less mistakes** on the roads.

- Travelers would be **able to journey overnight and sleep** for the duration.

- **Less traffic accidents** due to sensory technology.

- **Speed limits could be increased** to reflect the safer driving, shortening journey times.

- **Parking** the vehicle and **difficult maneuvering** would be **less stressful** and would require no special skills.

- There would be **no need for drivers' licenses** or driving tests.

- Efficient travel also means **fuel savings**, cutting costs.

- **Reduced need for safety gaps** means that **road capacities** for vehicles would be significantly increased.

Disadvantages of Driverless cars

EFOP-3.4.3-16-2016-00009

A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

- Driverless cars are very **expensive**, likely costing over $100,000.

- **Problems with certain types of weather**. Heavy rain interferes with roof-mounted laser sensors, and snow can interfere with its cameras.

- **Reading human road signs is challenging** for a robot.

- The **road system and infrastructure** would likely **need major upgrades** for driverless vehicles to operate on them.

- **Ethical problems** could arise which a machine might struggle to deal with. Faced with a choice between plowing into a group of schoolchildren or going off a bridge and killing all its passengers, what does the vehicle do? Should the vehicle always swerve to avoid animals in the road or always prioritize the safety and comfort of passengers?

- Human behavior such as **hand signals are difficult** for a computer **to understand**.

- **How would the police interact with driverless vehicles**, especially in the case of accidents or crimes?

Disadvantages of Driverless cars

The classification of automated vehicles

A formal **classification system for automated cars** has been proposed by the National Highway Traffic Safety Administration:

Level 0: **Driver has complete control** of vehicle at all times.

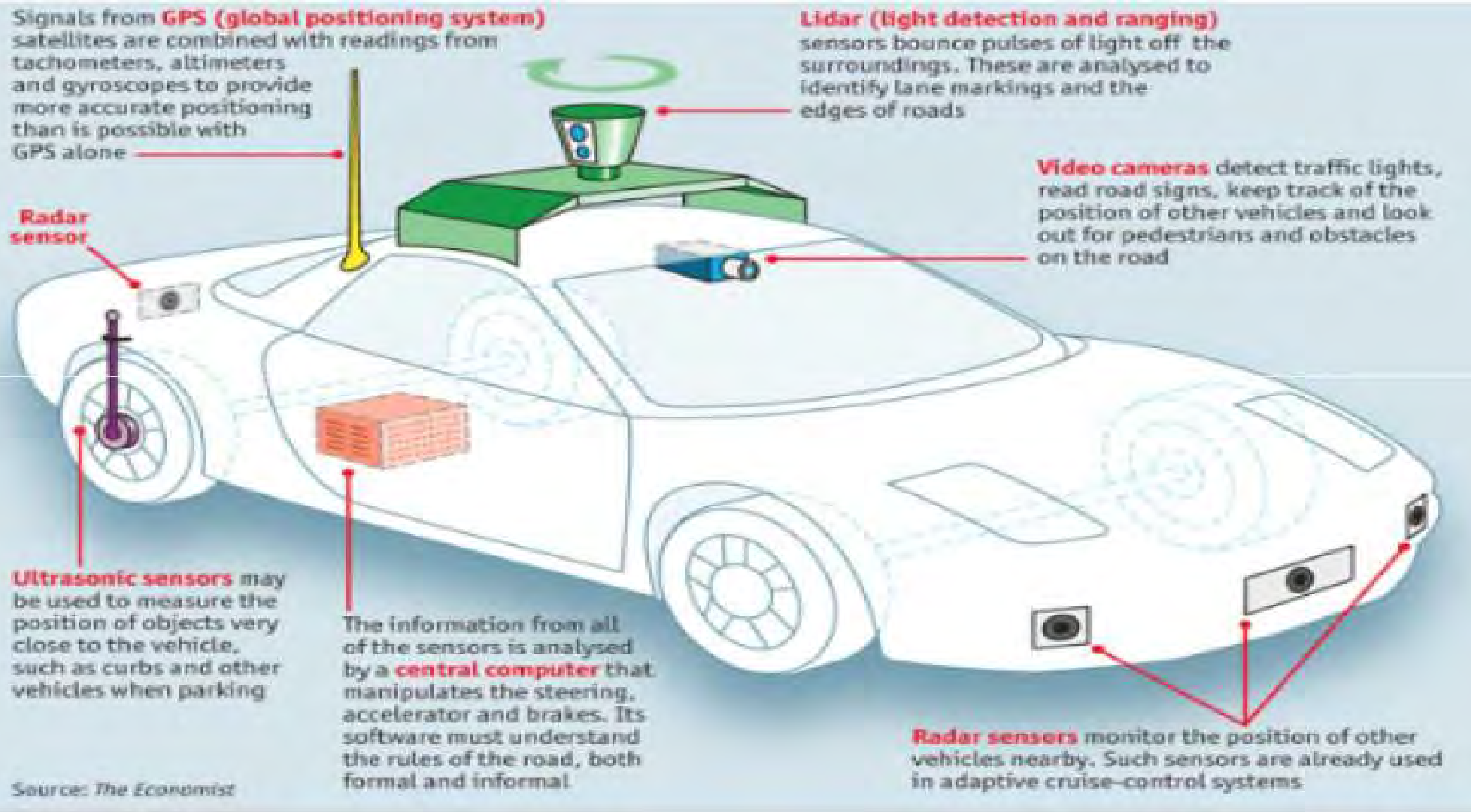
Level 1: **Some vehicle controls are automated**, e.g., automatic braking.

Level 2: **Two or more controls can be automated** at the same time, e.g., cruise control and lane keeping.

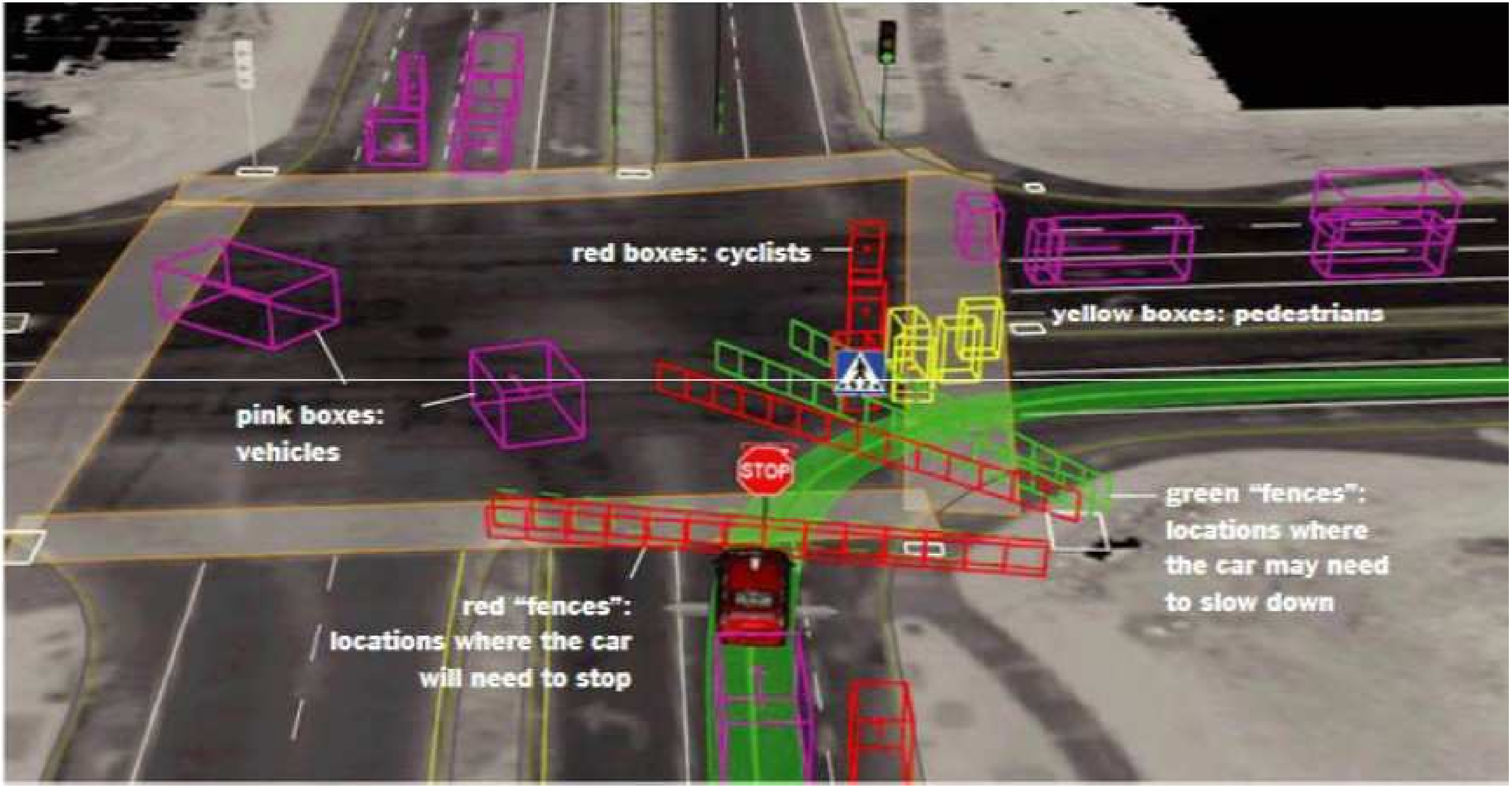
Level 3: The **driver can cede control** in certain circumstances.

Level 4: **Driver not expected to play any part** in the driving process at all.

How a car drives itself



What the car sees



Artificial intelligence in self-driving cars

1. **Sensing**: we need to enable self-driving cars to have **sensors that cover 360 degrees** using cameras and sensors for **redundancy** like radar scanners.

   Then the laser scanners take the data from all these sensors and go through a very powerful **high-performing computing device** in order to build an environmental model that can:

   - tell **where all the entities around the car are**, such as pedestrians, cyclists, signals, symbols, barriers, etc.

   - tell **what these entities are doing** or what their statuses are.

   - tell **about the path**, such as if this path is a highway.

EFOP-3.4.3-16-2016-00009
A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

2. Mapping:

Mapping **should not be confused with navigational map** (Google map). Instead, they are maps that are of high definition and they are very detailed.

Once you position your vehicle inside this map at a **very high accuracy (in the range of 10 cm)**, you know everything about the roadway, the drivable paths, the delimiter, etc., **the only thing that you don't know is where the road users are**.

These maps should reflect reality in a very short time, which means, the moment these environments change, the **map must be updated immediately**.

Today's maps are not updated at such a fast rate.

EFOP-3.4.3-16-2016-00009
A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

3. Driving Policy:

Self-driving car must understand driving policy, which is mostly about negotiation.

This skill does not come with naturally to us – it often takes both **driving lessons and driving experiences**. This is a **sophisticated negotiation** and since it is already difficult for humans, it is going to be even more difficult for computers.

Because we are talking about broad intelligence which involves understanding which vehicles to give way to, which vehicles to take the way from, what roads are often used to merge into traffic.

Furthermore, **driving policy changes according to locations** because traffic negotiation differs from city to city. This has to be taken into consideration when the driving policy is being fed to the system.

Neural networks in self-driving cars

The process of **identifying and classifying** objects from sensor data is known as **semantic segmentation** and autonomous vehicle depends on artificial neural networks for this process.

The **inputs are image data** and the **output is particular class of object**.

EFOP-3.4.3-16-2016-00009

A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

If we think about the range of inputs on an autonomous car, we might have data from the camera, radar, lidar, the road conditions, the humidity … **perhaps 10 highly-dimensional sources**. **With so many attributes a neural network would make sense**.

The process of training a neural network for semantic segmentation involves feeding it with numerous sets of **training data with labels** to identify key elements, such as cars or pedestrians. This data can be **generated from simulations** (providing they're accurate enough) **or captured from real-world footage**.

Self-driving cars are prime candidates for the use of neural networks. Neural networks could be used to **predict behavior based on a sequence of events**. It's not inconceivable, for instance, that a neural network could be taught to recognize that a ball bouncing into the road could be followed by a child.

In the autonomous car, one of the major tasks of a machine learning algorithm is the **continuous rendering of the surrounding environment and the forecasting the changes** that are possible to these surroundings.

These tasks are classified into 3 sub-tasks:

- The **detection** of an Object
- The **identification** of an Object or **recognition** object classification
- The Object **localization and prediction of movement**

The Machine Learning Algorithms used in self driving cars

EFOP-3.4.3-16-2016-00009

A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

The machine learning algorithms are loosely **divided into 4 classes**:

- **decision matrix** algorithms (e.g., AdaBoost: By adding the weak learners iteratively, AdaBoost creates a strong learner)

- **cluster** algorithms (e.g., K-means)

- **pattern recognition** algorithms (classification after edge/arc findings)

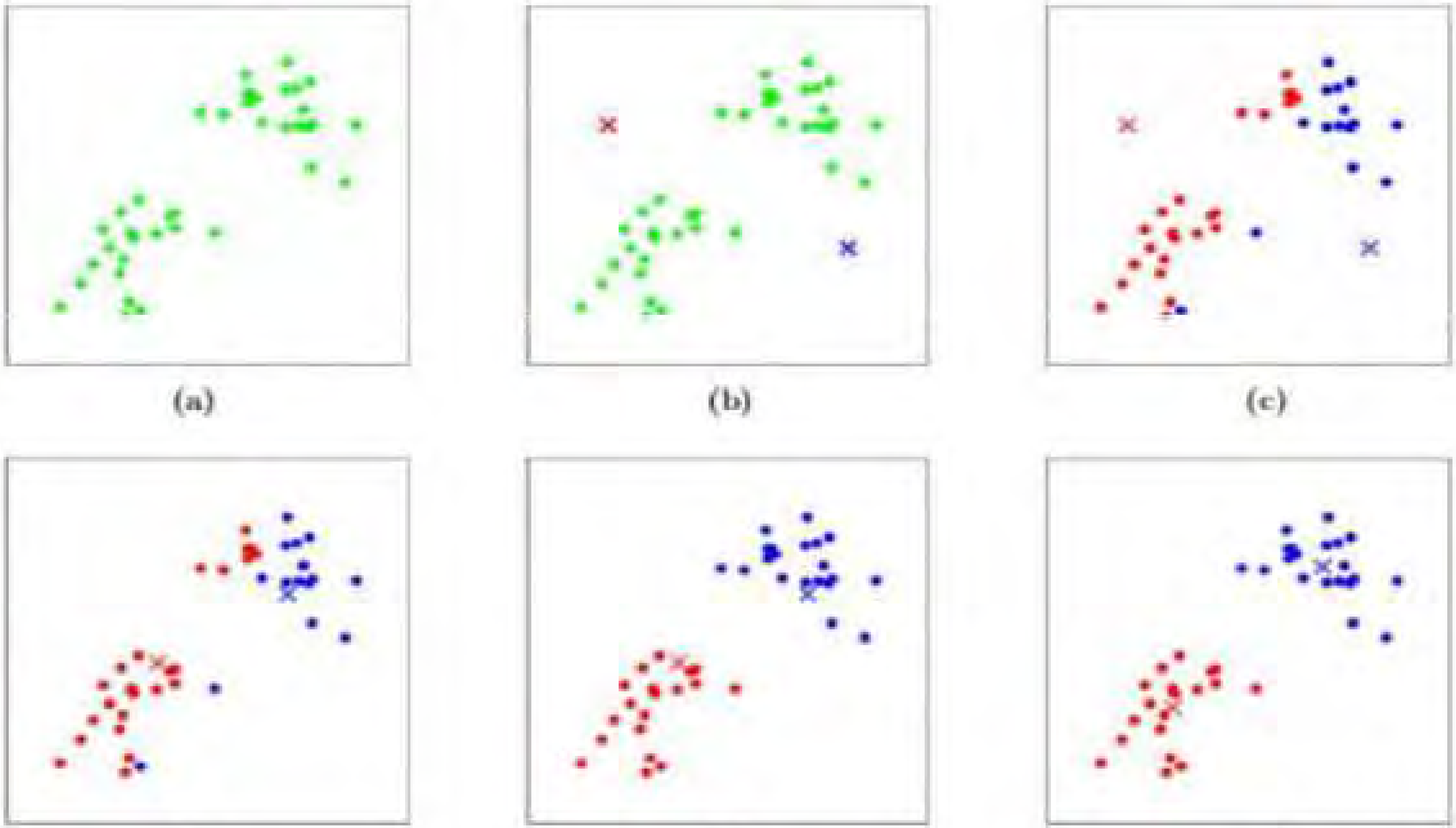
- **regression** algorithms

One category of the machine learning algorithms **can be utilized to accomplish 2 or more subtasks**. For instance, the regression algorithms can be utilized for object localization as well as object detection or prediction of the movement.

Cluster algorithms (e.g., K-means)

EFOP-3.4.3-16-2016-00009

A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen



(a) (b) (c) (d) (e) (f)

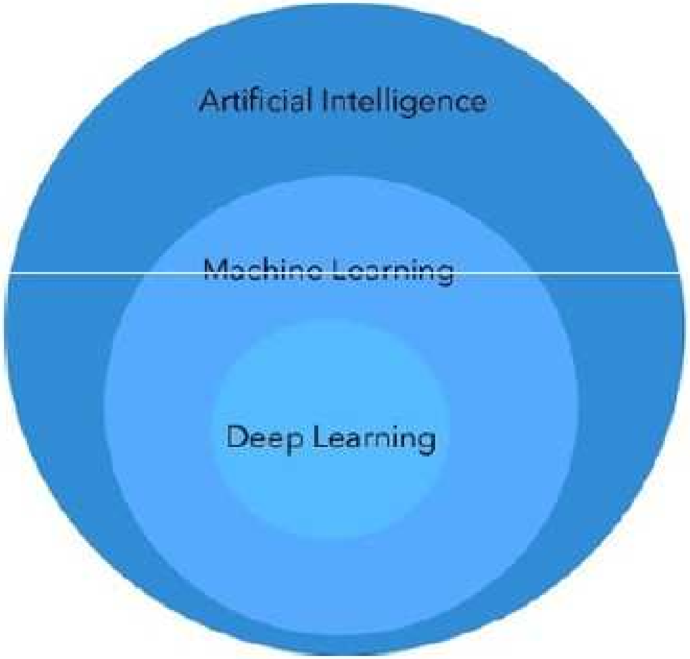
Deep learning

EFOP-3.4.3-16-2016-00009

A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

Deep learning (also known as **deep structured learning** or **hierarchical learning**) is part of a broader family of machine learning methods based on learning data representations, as opposed to task-specific algorithms.

Learning can be supervised, partially supervised or unsupervised.

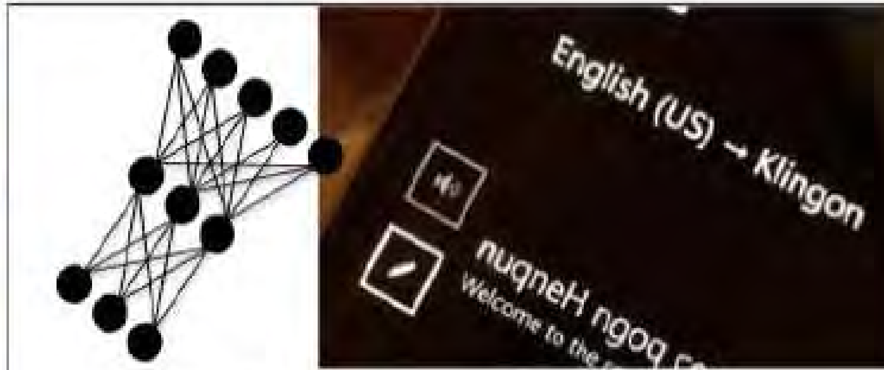


Deep learning examples from the practice

EFOP-3.4.3-16-2016-00009

A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

Image Classification, Object Detection, Localization, Action Recognition, Scene Understanding



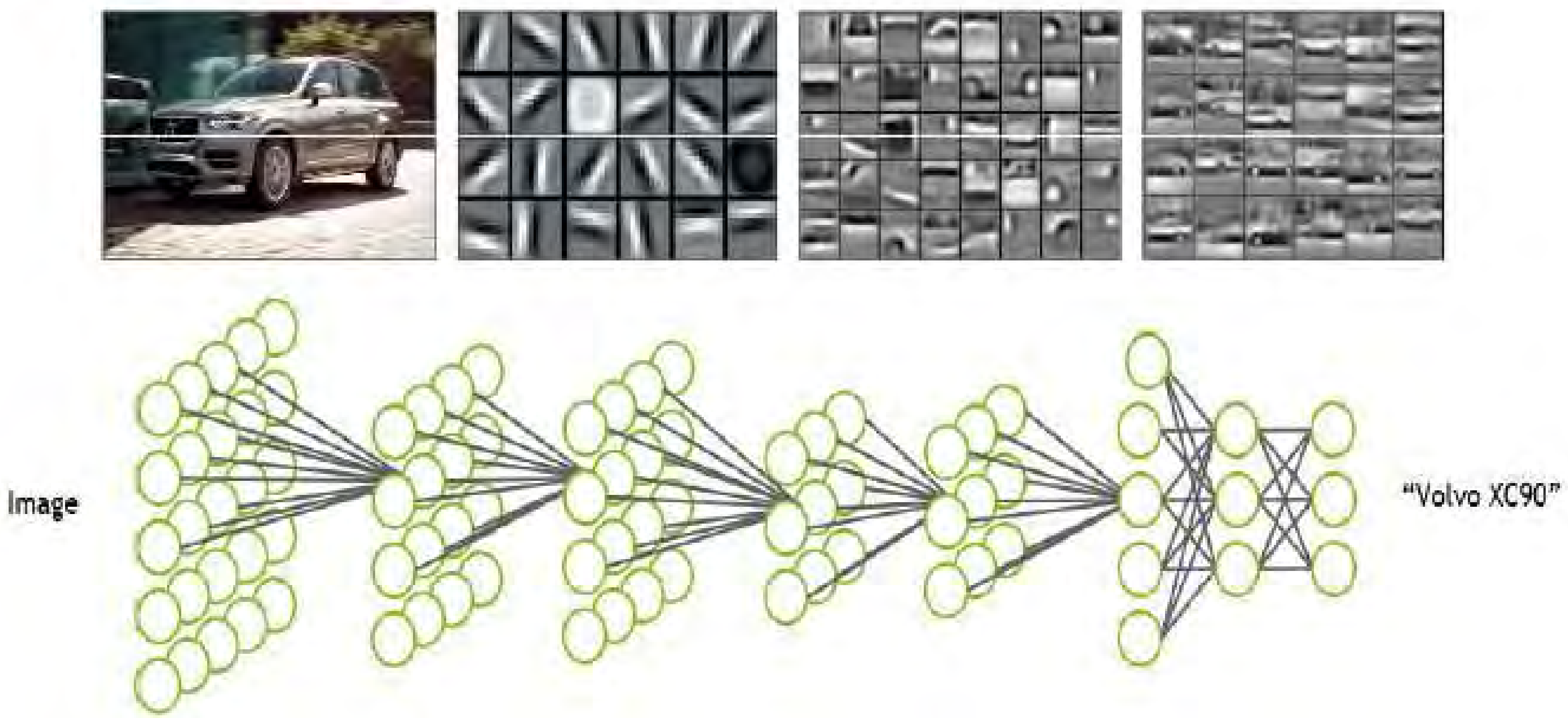
Speech Recognition, Speech Translation, Natural Language Processing

Pedestrian Detection, Traffic Sign Recognition

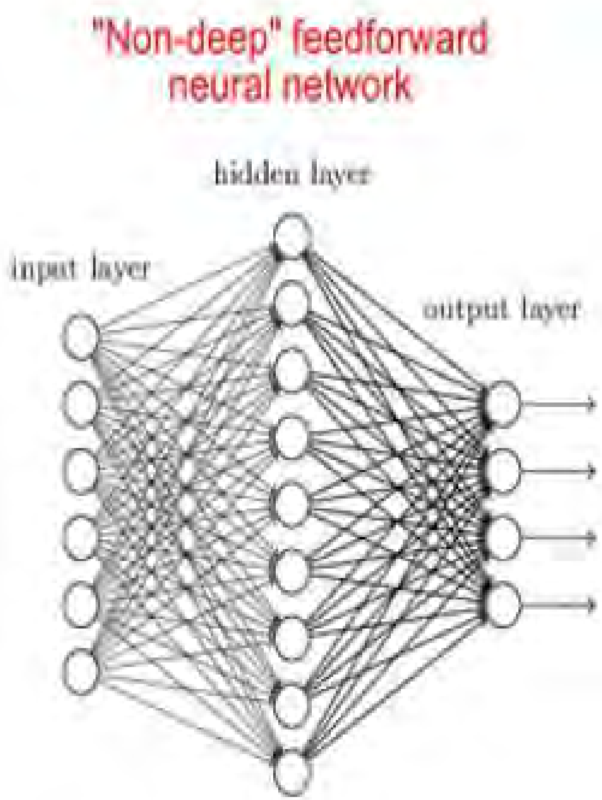
Breast Cancer Cell Mitosis Detection, Volumetric Brain Image Segmentation

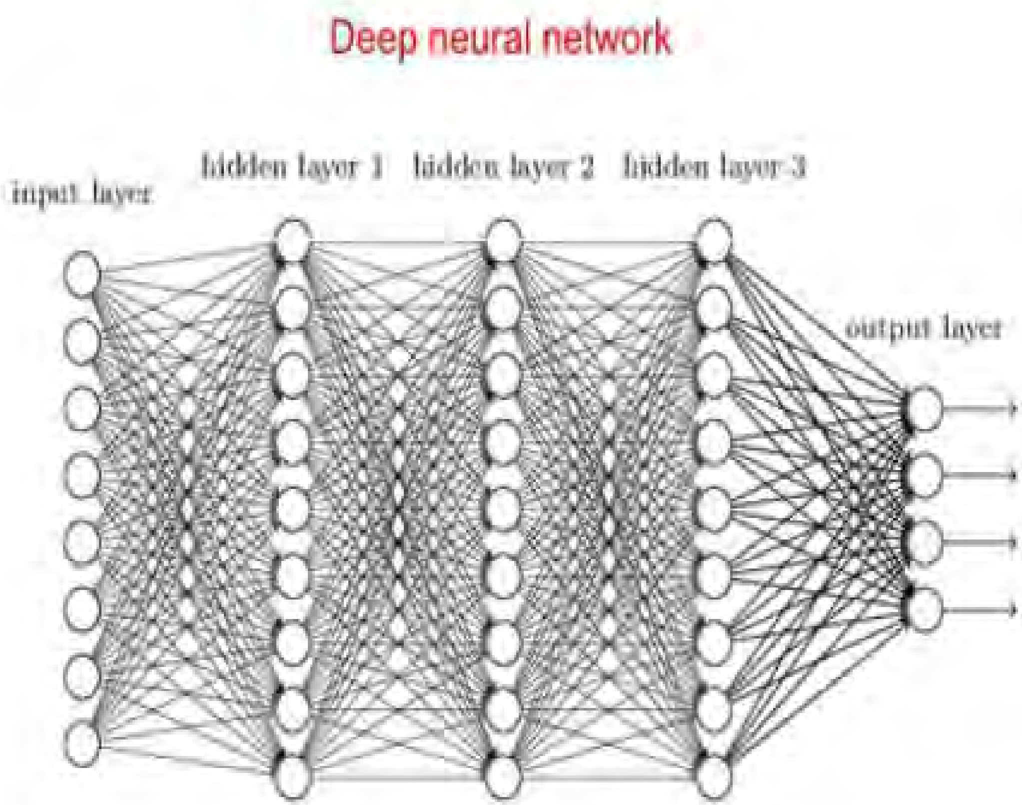
Deep neural networks

A **deep neural network** is a neural network with a certain level of complexity, a **neural network with more than two layers**. Deep neural networks use **sophisticated mathematical modeling** to process data in complex ways.



Shallow vs. deep neural networks





There are two main approaches.

First approach

- It uses semantic abstraction, where the problem of **autonomous driving is divided into several components**. These are algorithms that focus only on one part of the task. For example, one component can focus on pedestrian detection, another on detecting lane markings and a third one for detecting objects beyond the lanes.

- At the end, **these components are „glued together" into a master network** that makes the driving decisions. On the other hand, a network can be constructed in such a way that it detects and classifies multiple classes or even carries out semantic segmentation.

- The **advantages** of such a system, is the **lower tolerance for mistakes**, the ability to pinpoint the errors more easily and the capability to **manage unpredictable situation better**.

- Its **shortcomings**, however, are also big, since it **requires huge pre-work and complex programming**.

Second approach

- It is an **end-to-end learning** approach. This is where **the car actually teaches itself how to drive, based on a huge set of human driving data**.

- Although this approach also has big **shortcomings**, such as the **requirement of having a huge training data set** and the **difficulty to be trained and tuned properly**, it is **very promising** for the future of intelligent vehicles.

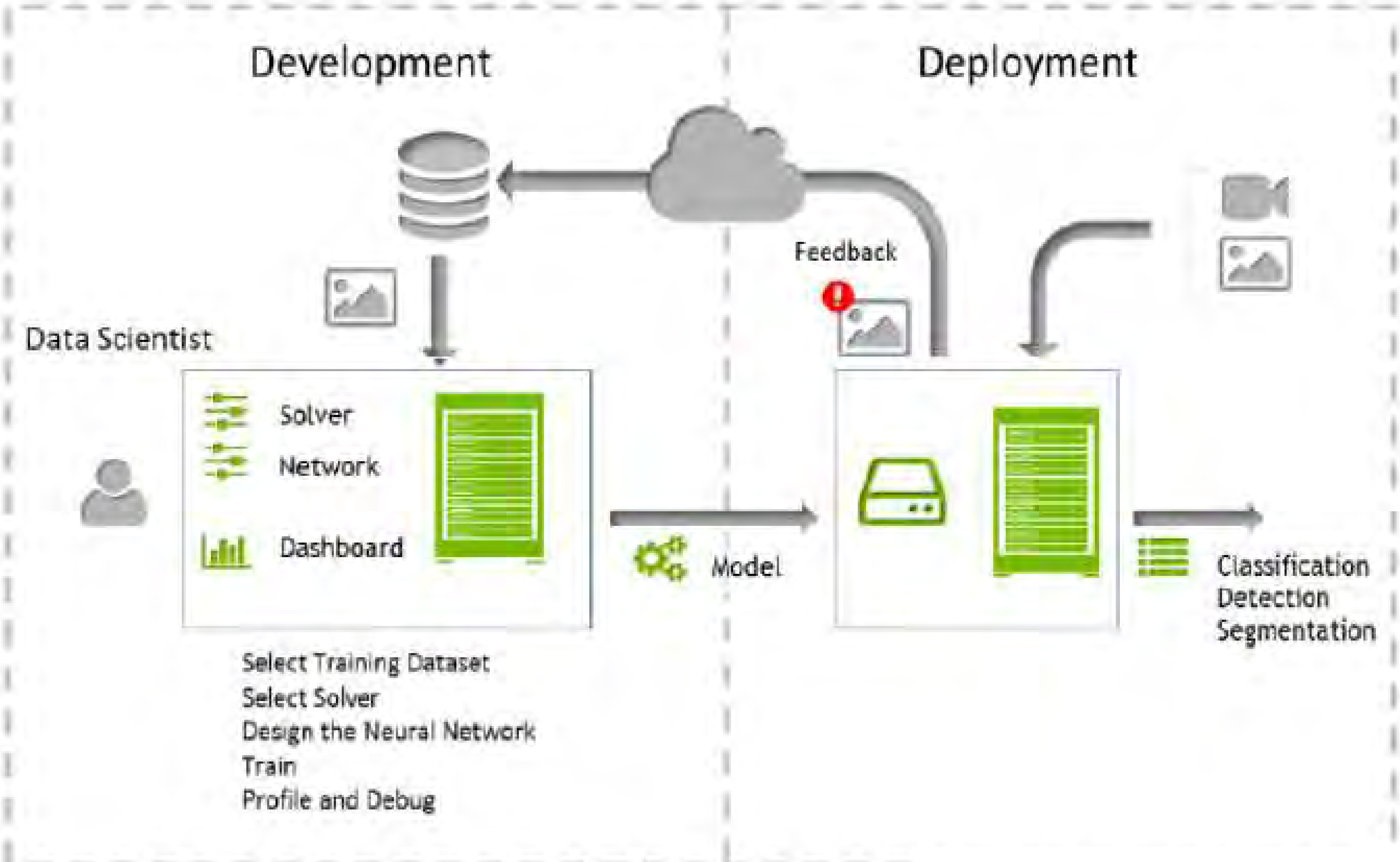
Development process for autonomous vehicles

EFOP-3.4.3-16-2016-00009
A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

source: Jan Wiegelmann: Deep learning for autonomous driving

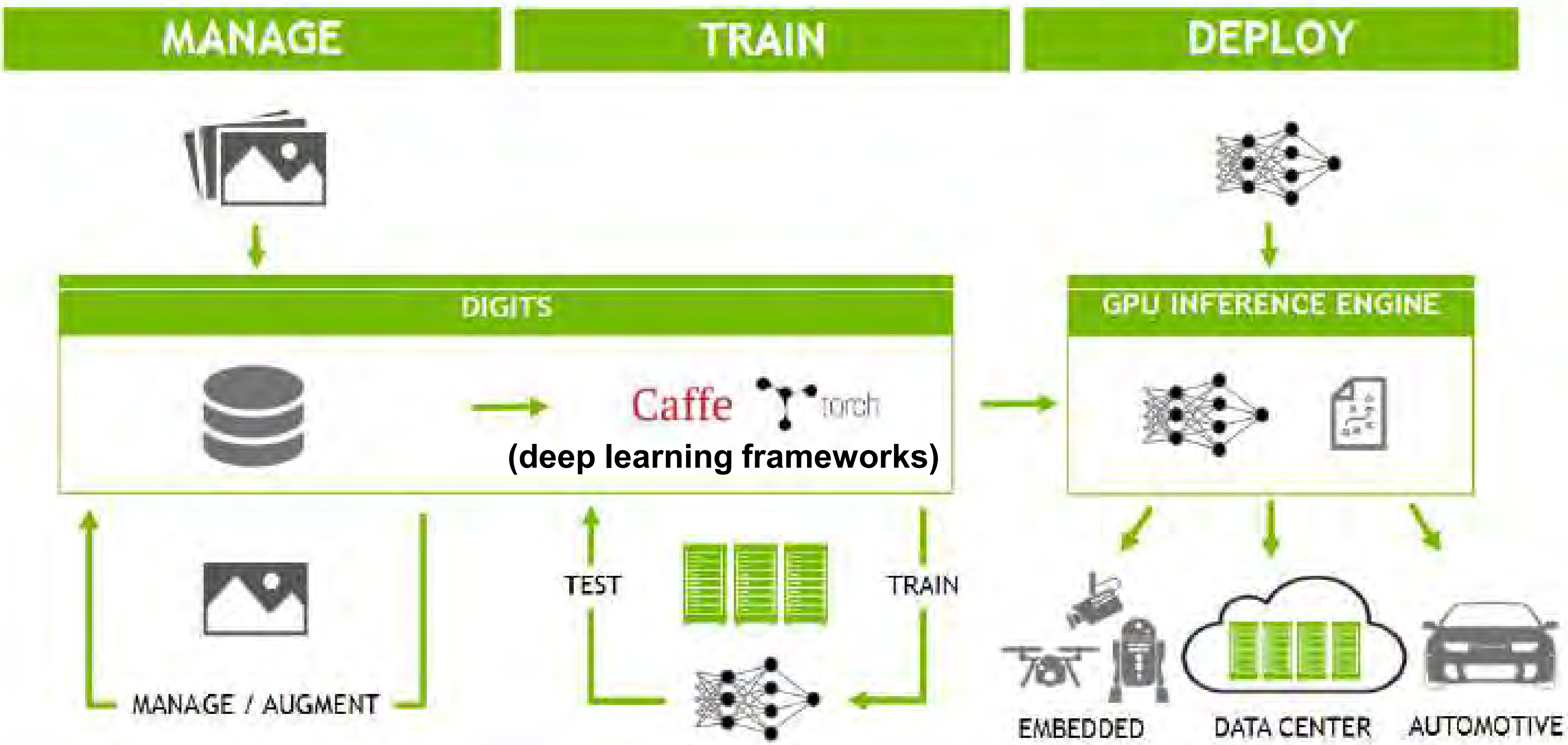
How cars see better and learn using deep learning

EFOP-3.4.3-16-2016-00009
A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen



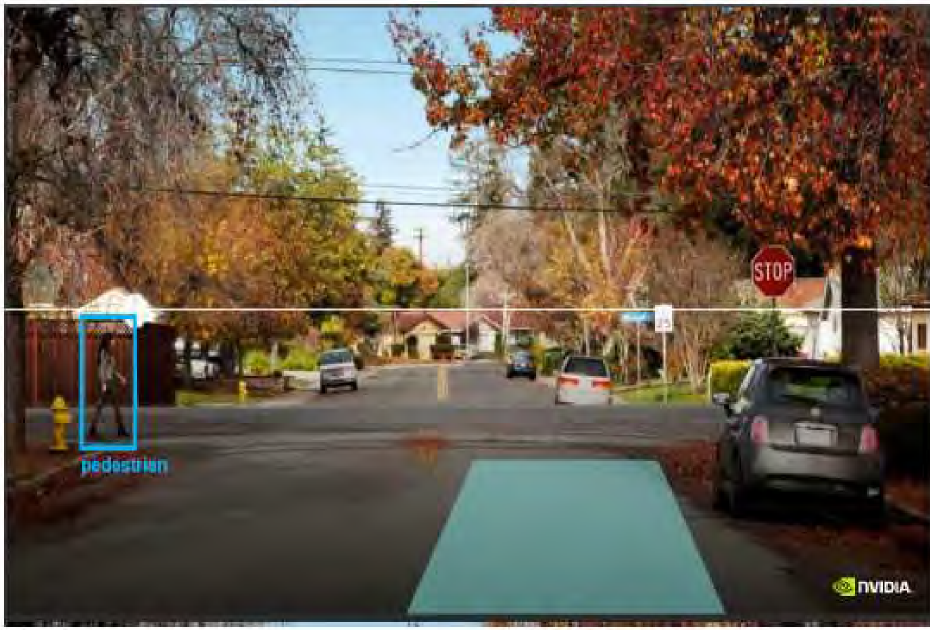
source: https://devblogs.nvidia.com

Complete deep learning platform



source: https://devblogs.nvidia.com

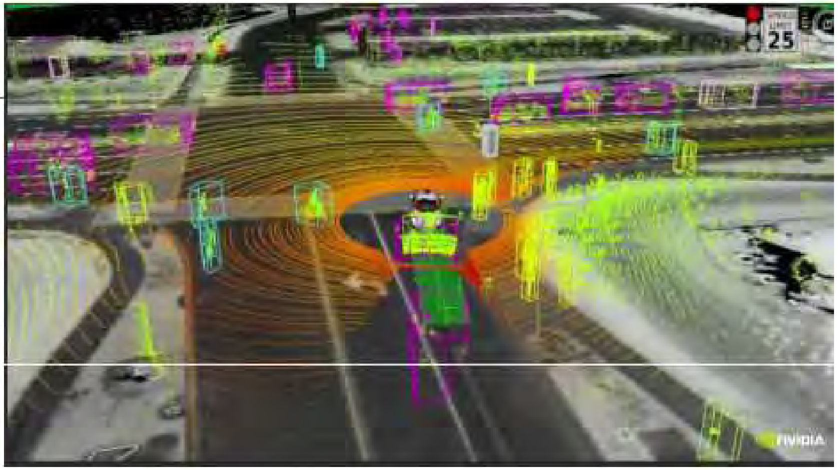
Deep learning inference



Deep neural networks are able to **recognize and classify the surrounding objects** …

Deep learning inference

EFOP-3.4.3-16-2016-00009
A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen



… **anywhere they appear** around the vehicle.

Autonomous car projects

Tesla Autopilot

Google self-driving car(Waymo)

Shortcomings and challanges of deep learning

EFOP-3.4.3-16-2016-00009
A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

Processing power

It is still a challenge to have a low cost **GPU** that operates within the **energy consumption** and other boundaries, such as **heat management**. Moreover, companies still struggle with **bandwidth** and **synchronization** issues.

Available training data

At least a **billion kilometers** of training data from realistic road scenarios are needed in order to make conclusions about safety of the vehicle. Moreover, the **data needs to be diverse** enough for it to be useful.

Safety

At present **safety assurance and verification methods are poorly studied** (**ISO 26262**, titled „Road vehicles – Functional safety”).

A prominent example of a safety failure is the 2016 Tesla auto-pilot accident, where the sensors of the vehicle were blended by the sun and the system failed to recognize the truck coming from the right, leading to the crash.

Future of autonomous vehicles

There are still **a lot of technological advancements** that need to be made to enable safe driverless capability (autonomous vehicles without a human driver).

The **capability of the perception systems** needs to increase, which means that sensor capabilities need to increase and the cost of sensors needs to come down in order to enable driverless vehicles.

Furthermore, the **creation and updating of high-definition maps** that are used by the autonomous vehicles need to become commonplace when we drive in the urban environments (off-highway).

One of the biggest barriers is that we still need to ensure **the human driver is „in the loop"**. This means that we need to address all of the **human-machine interaction issues** in order to provide safety in all circumstances.

EFOP-3.4.3-16-2016-00009

A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

THANK YOU FOR THE ATTENTION!

References:

http://www.deeplearningbook.org

https://selfdrivingcars.mit.edu

Combining Deep Learning with traditional algorithms in autonomous cars - Albin Falk & David Granqvist

www.theengineer.co.uk/ai-autonomous-cars

SZÉCHENYI 2020

MAGYARORSZÁG KORMÁNYA

Európai Unió
Európai Szociális Alap

BEFEKTETÉS A JÖVŐBE

EFOP-3.4.3-16-2016-00009

A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

ARTIFICIAL INTELLIGENCE

13-14. Ethical issues of AI

Authors:

Tibor Dulai, Ágnes Werner-Stark, Zsófia Bircher

SZÉCHENYI 2020

MAGYARORSZÁG KORMÁNYA

Európai Unió
Európai Szociális Alap

BEFEKTETÉS A JÖVŐBE

Content

EFOP-3.4.3-16-2016-00009
A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

- How much does AI influence our life?
- What ethic is?
- AI hype
- Is prejudice only human-related problem?
- OECD – better policies for better life
- FairML, LIME, Deon
- The economic background of AI
- AI’s effect on the labour markets
- Economical dangers

How much does AI infuence our life?

EFOP-3.4.3-16-2016-00009

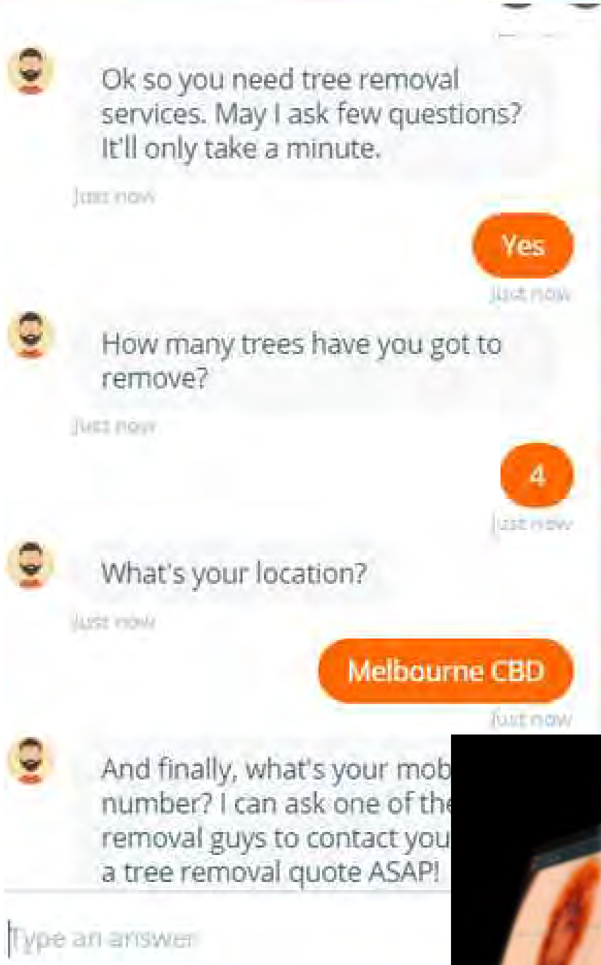
A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

- Social media
- Digital assistants
- Search on the Internet
- Business and services

How much does AI infuence our life?

EFOP-3.4.3-16-2016-00009

A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen



- Self-driving and parking vehicles
- E-mail and automated communication
- Healthcare

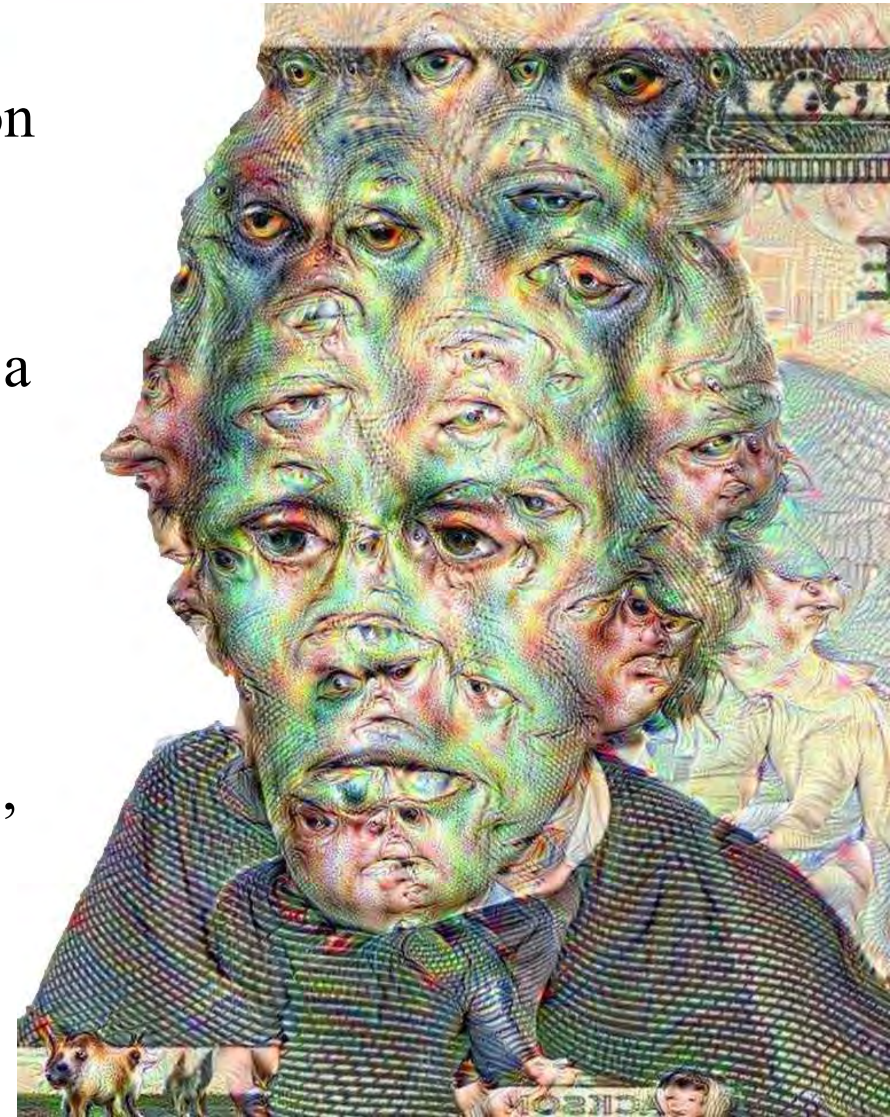
What ethic is?

EFOP-3.4.3-16-2016-00009
A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

- Its meaning: **rational**; **optimal and right decision made by common sense**.
- Ethic **has to be handled as a very important aspect** by everyone who uses or creates machine learning systems.
- It is necessary that **everyone in the industry takes responsibility**.
- Its **regulation can not be let to do by only politicians or** members of **some organizations**. People who can have real **infuence** on it are the **developers and engineers**.

Creativity of computers

- **AI is considered logical, rational and effective** because of its impact on the economy and technology.
- The field of **creativity includes culture, art and communication** in a developing society.
- It can help to **come up with new ideas**.
- Maybe its **creative influence is underestimated**: it can produce, e.g., **fake news**.



AI hype

A „**new arms race by AI**" was born between USA, Russia and China.

The AI hype resulted that **every company had to open suddenly an AI-division**, despite there are **too few people with real experiences** in this field.

What is against having a super-intelligent AI in the near future?

Most deep-learning systems are **highly data- and problem-specific**.

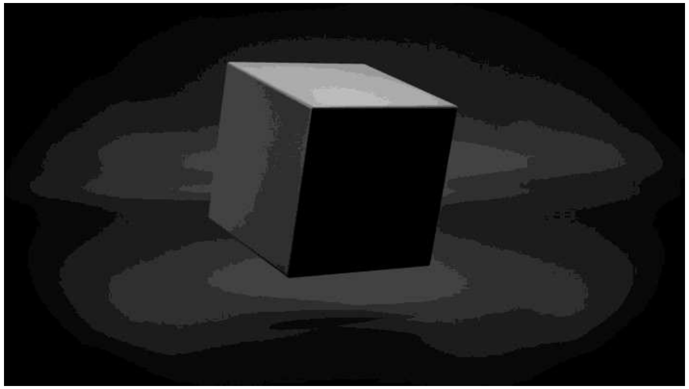
Can more data solve these problems? Due to the distribution of these problems and the likelihood of such data, the **bottleneck still remains**. That is, these systems do **not learn the global context**, **even if they are trained with more data**.

- Language systems can be trained without a specific task and they provide a huge amount of context. At the same time, **language is a dynamic, evolving system** that changes dramatically with time and the environment.

Is prejudice only human-related problem?

EFOP-3.4.3-16-2016-00009
A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

1. Actuarialists calculate **insurance prices** for years based on aggregated risk.
2. **Automatic model** by machine learning.
3. **Prices could have been reduced** by 10-20%.
4. The company **performed well** and the stock price performed well in the market.
5. In spite of all these, the system was eventually **banned in many states**.



Is prejudice only human-related problem?

EFOP-3.4.3-16-2016-00009
A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

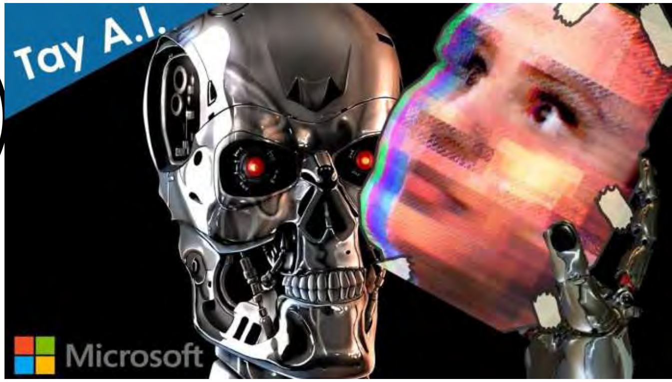
- Would **Amazon** change its **hiring process for women** so quickly if AI didn't show the distortion?
- AI highlights and helps you to understand a lot of things. There are three main reasons why **we can't slow it down** :
  - Globalization.
  - It can bring enormous benefits to the developing world.
  - It is strength to promote equality.
- AI can **illuminate unconscious bias** to help eradicate it.

Is prejudice only human-related problem?

Context

- We need to be aware of **why** we are trying **to do something** and **what its potential consequences** can be.
- If we use an algorithm, we essentially **automate decision-making**.
- Does it work the way we thought it would? Is there any **sign of bias**?

The Canadian government uses a **questionnaire** "to assess and mitigate the **risks associated with deploying an automated decision system**".



OECD – Better policies for better life

- Its principles **promote artificial intelligence**, which is
  - **innovative and reliable**
  - **respects human rights and**
  - **democratic values**
- These were adopted by OECD member states in May 2019, when they adopted the **OECD Council Recommendation on Artificial Intelligence**.
- The OECD AI Principles define AI with **practical and agile standards** that stand the test of time **in a rapidly evolving field**.
- They **complement existing OECD standards** in areas such as **privacy, digital security risk management and responsible business conduct**.

5 Recommnedations for Responsible Management of AI

EFOP-3.4.3-16-2016-00009

A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

- AI needs to **support people and the planet** by promoting growth, sustainable development and prosperity.
- AI systems must be designed to **respect the rule of law, human rights, democratic values and diversity**, and must include **appropriate safeguards** to ensure an appropriate fair society, for example by allowing for human intervention when it is necessary.
- There must be **transparency and responsible disclosure of AI systems** to ensure that people understand and challenge AI-based results.
- AI systems must **operate in a robust, secure manner** throughout their life cycle, and potential **risks must be continuously evaluated and managed**.
- **Organizations and individuals** developing, installing, or operating AI systems **should be made responsible** for the proper operation in accordance with the above principles.

The 5 OECD suggestions for the governments

EFOP-3.4.3-16-2016-00009
A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

- **Facilitate public and private R&D investment to drive innovation** in trusted AI.
- Promote accessible AI ecosystems through **digital infrastructure, technologies and mechanisms for sharing data and knowledge**.
- Provide a **policy environment** that paves the way for reliable AI systems to be deployed.
- **Empowering people with AI skills and supporting** employees for a **decent transition**.
- Promote responsible management of trusted AI through **cross-border and industry collaboration**.

AI4People - European organization

- **AI Global Mark of Compliance** (AIGMC) is a quality assurance label that **certifies the ethical use of AI**.
- **Vertical Business**: when the business is looking for a broad web and creates **products or services** that can be **attractive to** just about **anyone**,
- **AI4People is a multi-stakeholder forum** bringing together stakeholders involved in shaping the social impact of new applications of active innovation, **including**:
  - the **European Commission**,
  - the **European Parliament**,
  - **civil society organizations**,
  - the **industry**, and
  - the **media**.
- **Creating an ethical framework for a "good AI" society**, recognizing that AI is a powerful force that transforms our lives, interactions and environments, **is a key element** of this.

FairML, LIME, Deon

- There are **many tools** that can help us better **understand the persistence of bias** in algorithms.
- One of them (**FairML**) helps engineers **identify the extent to which algorithmic inputs can cause bias**. (Python)
- The **LIME** does not differ from FairML. Its purpose is to understand **why the algorithm makes its decisions** by interfering with the "input" and how it affects its outputs.
- **Deon**, which is more like a lighter, more developer-friendly version of an **algorithmic evaluation**. This is a command line tool that allows you to **add an ethics checklist** to your projects.

AI is more and more popular

EFOP-3.4.3-16-2016-00009

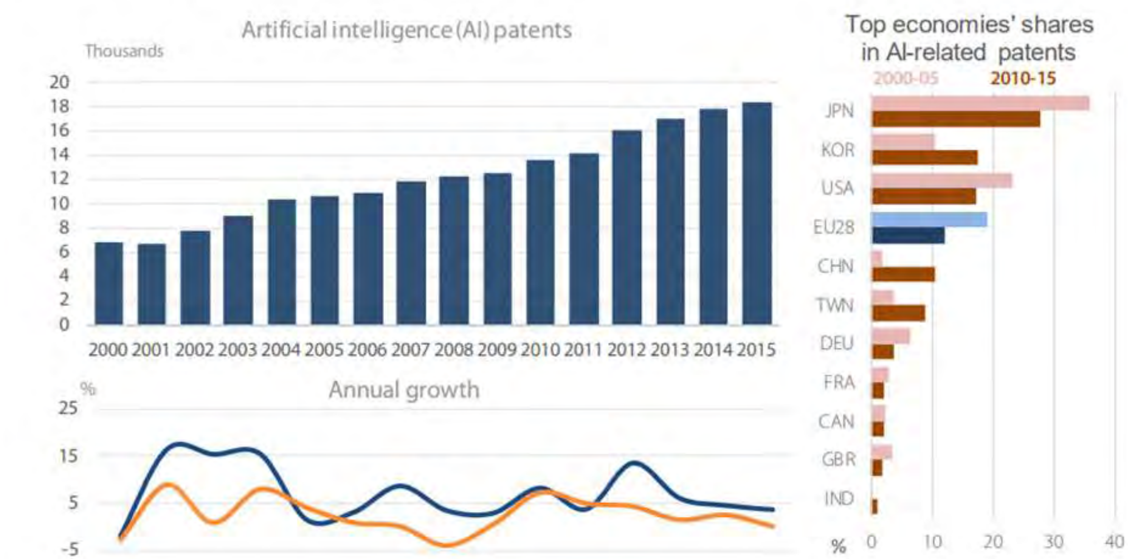
A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

- The number of AI patents show growth worldwide (see figure), with annual average growth rate of 6% between 2010 and 2015, higher than the annual growth rate observed for other patents.

Initially, the leading countries were Japan, South Korea and the United States, which together accounted for nearly two-thirds of patent applications for artificial intelligence.

South Korea, China and Chinese Taipei significantly increase AI patents compared to past results.

Figure 1 – AI patents worldwide, 2000-2015



Source: OECD, Science, Technology and Industry Scoreboard, 2017.

The economic background of AI

- Some warn that AI could lead to the creation of **super firms** that are **nodes of wealth and knowledge**, and thus have a detrimental effect on the wider economy.
- Experts also believe that AI is a "good" basis for **widening inequalities, reducing wages** and **reducing the tax base**.
- There is a global consensus that AI technologies have the potential to **revolutionize production** and **contribute to addressing major global challenges**.

The economic background of AI

- The World Intellectual Property Organization (WIPO) reports that most AI-related patents are in the field of:
  - telecommunications,
  - transport,
  - medicine,
  - personal tools that take part in human-computer interaction.
- The fastest growing applications of AI are:
  - smart cities,
  - agriculture,
  - e-government,
  - banking,
  - finance.

What can happen?

- In the near future AI can lead to **productivity gains based on automation of routine tasks**, which is likely to affect capital intensive sectors such as **manufacturing and transport**. This includes the widespread use of technologies such as **robots and autonomous vehicles**.
- The availability of **personalized and better quality AI-enhanced products and services** becomes even more important as this availability **increases consumer demand**, which in turn **generates more data**.
- It is estimated that AI could generate an **additional $13 trillion in economic revenue by 2030**, increasing global GDP by about 1.2% annually.

What can happen?

EFOP-3.4.3-16-2016-00009
A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

- AI is likely to **shock the labor markets** and require related **costs to handle labor market transitions**.
- Despite the progress made by AI, **certain areas** of the economy remain essential while **maintaining well-paid human labor**.
- AI **may even hold back future innovation by accelerating imitation**, which would limit the pay-off on innovation.
- According to a well-known productivity paradox, **lo productivity** is experienced in an age of **acceleratin technological development**.

Industry 4.0

- **AI is one of the bases** of the growing digitalisation of industry ("Industry 4.0").
- In future smart factories, **production processes would be interconnected**, and AI solutions would be essential to **connect machines, interfaces and components**.
- AI can lead to **breakthroughs in science**, which can even create completely **new, unpredictable job types**.
- Allows **small players and even individuals to complete a project** that is usually done by larger companies these days

⬇

The end result will be a **dumbbell-shaped economy**

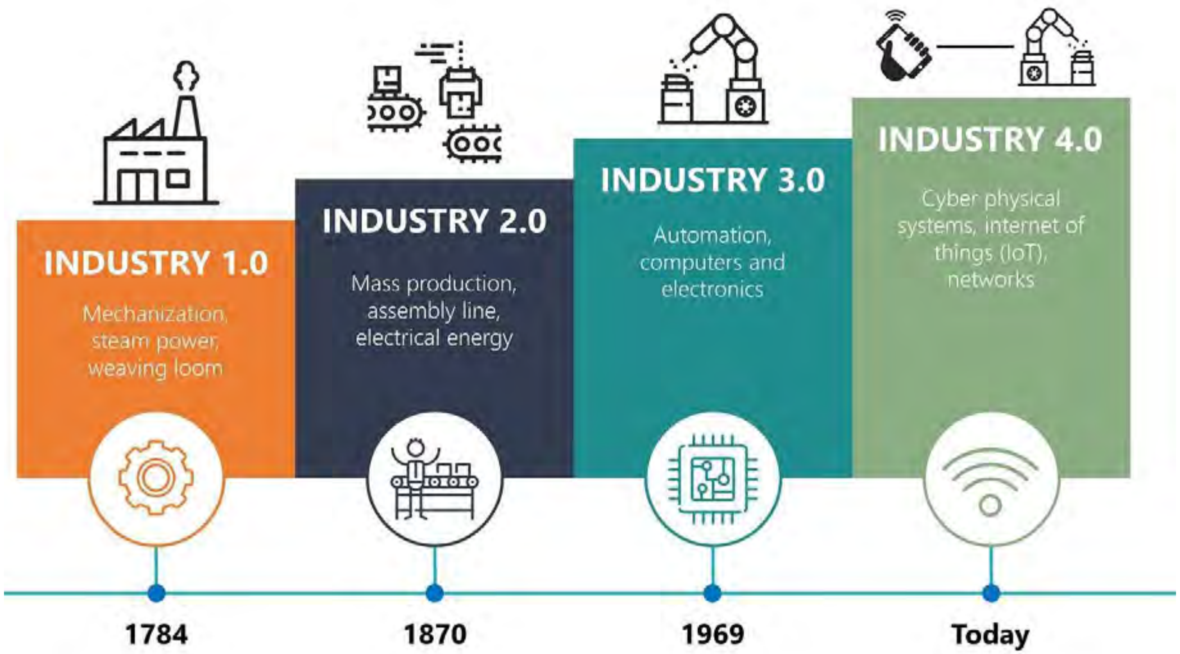
medium-sized businesses lose.

Impact of AI on labor markets

EFOP-3.4.3-16-2016-00009

A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

- Studying patterns from earlier industrial revolutions shows that **job destruction will be possibly stronger in the short and medium term,** while **job creation will last longer**.
- **Society** as a whole **would be much richer**, **but for many people, communities and regions**, technological change would only **exacerbate inequalities**.
- **Low-skilled automation** always **increases wage inequality**, and **high-skill automation always reduces it.**



Commentators

Bill Gates is one of the commentators who say robots who work for someone have to pay taxes

In 2017, the European Parliament rejected the idea of imposing a „robotic tax" on owners to support the retraining of workers who had to leave their jobs because of the robots.

Economic threats

- **If the employee does not benefit** from the economic benefits generated by AI, **then consumption may stagnate and limit growth**.

- Prerequisites for the success of AI's potential are the **development of education and relevant skills in work**, as well as the **funding of research** and **pooling of resources**.

EFOP-3.4.3-16-2016-00009

A felsőfokú oktatás minőségének és hozzáférhetőségének együttes javítása a Pannon Egyetemen

THANK YOU FOR THE ATTENTION!

SZÉCHENYI 2020

MAGYARORSZÁG KORMÁNYA

Európai Unió
Európai Szociális Alap

BEFEKTETÉS A JÖVŐBE