Machine learning, classification, artificial neural networks + two simple neuroans (Mc Culloch Pitts neuron, perceptron)

- Machine learning: Computer changes its knowledge structures (facts, rules, parameters, network structures, etc.) over time, due to experience, i.e. ongoing interaction with digital environment and humans.
- Feedback may be via examples (supervised learning), rewards or punishment (reinforcement learning), or indirect (unsupervised learning) – or, as always: hybrid
- Classification: Learn to assign one of a finite number of classes to an item. A typical machine learning task (supervised learning).
- Neural networks: Inspired by the human brain, networks of digital neurons, i.e. formula of varying complexity.
  - Simplest digital neuron: McCulloch Pitts Neuron, equally expressive as Boolean logic if inhibitory inputs are modelled. Binary input (n dimensions), binary output.
  - Perceptron: Weighted continuous input values (n dimensions), binary output when the decision function is a threshold over the sum of input values (linear n-dimensional decision plane).

So now we can decide for a given input vector, to which class it belongs

Limitation of what we concretely discussed:

- Two classes only (binary output)
- Linear separation only (perceptron function = threshold function over sum of input values)
- Separation function maybe not optimal

What is difficult about classification?

- What are good classes? Do we differentiate between the right problems?
- What is a good vector representation of the items that let‘s us differentiate between classes easily?
- What is a good algorithmic set-up to differentiate between clases – is it really a neural network?
- If yes – what is a suitable ANN architecture? What are good decision functions?
- **If we have all that – what are suitable parameters (weights)?**

We are discussing today just the last point – so we enter into the discussion under the assumption that we already have made many choices and that these have been reasonable. We have decided on two classes, a reasonable vector representation, and a single perceptron as algorithmic approach to classification, with a linear activation function.

WISSEN
TECHNIK
LEIDENSCHAFT

Machine Learning

Viktoria Pammer-Schindler

Introduction to Data Science and Artificial Intelligence

www.staff.tugraz.at/viktoria.pammer-schindler

ISDS
INSTITUTE OF INTERACTIVE SYSTEMS AND DATA SCIENCE

Remember: Classification Definition

Classification is the activity of deciding for a given input x, to which class/category the input belongs.

Given an input x

Determine $y = f(x)$

$y \in \{y_1 ... y_n\}$ – is one of a finite number of classes

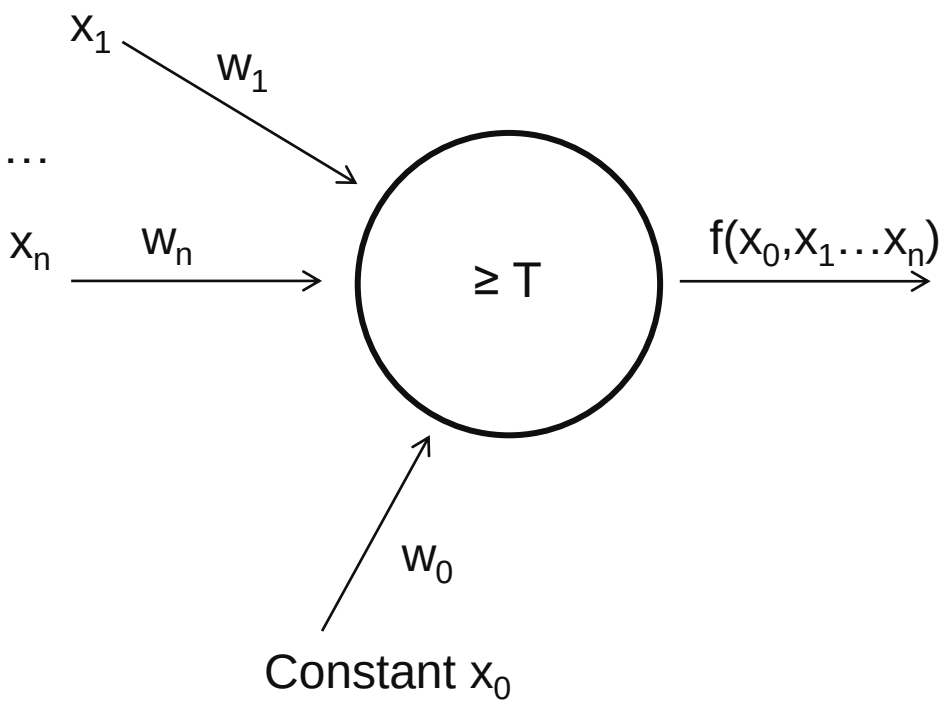
Typical machine learning task, when rules of classification become too complex

Is it sunny or raining? – A Classification Task

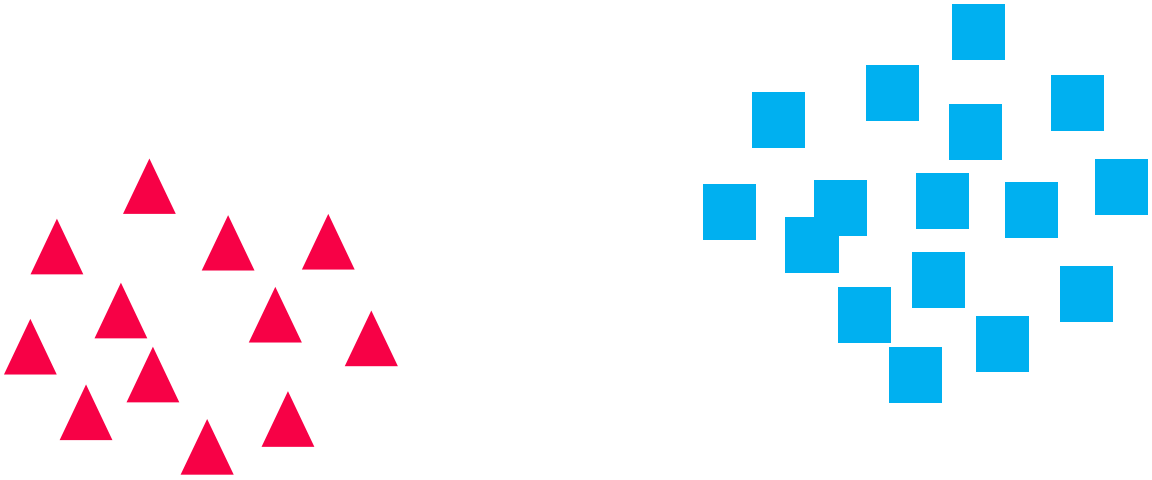
Es nieselt gerade.

Perceptron

$$f(x_0 \ldots x_n): w_0x_0 + w_1x_1 + \ldots w_nx_n \geq T$$



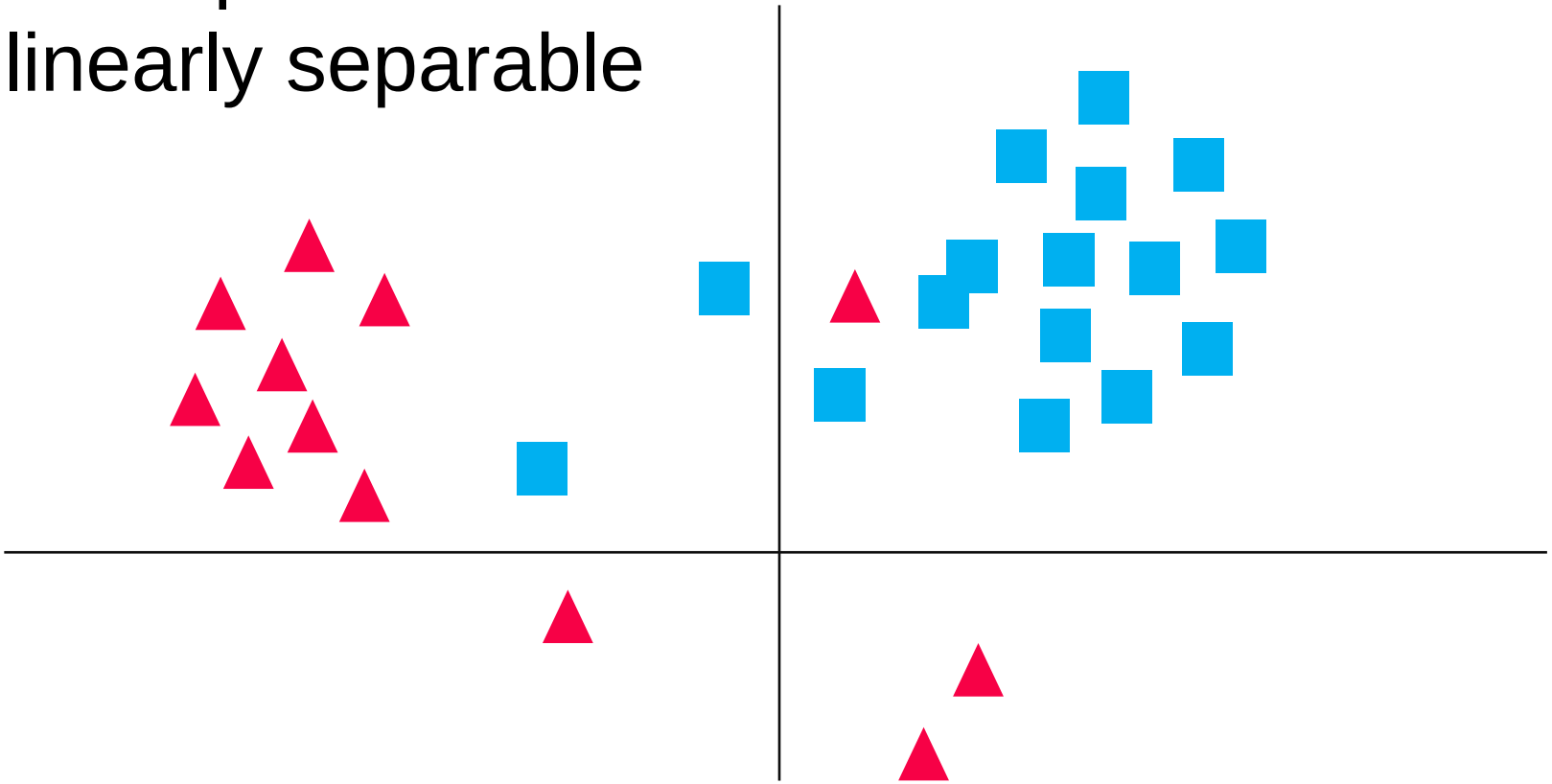
Example of a linearly separable classification task



Does a solution to the perceptron function exist?
Is it unique? Is it perfect?

$f(x_0 \ldots x_n)$: $w_0x_0 + w_1x_1 + \ldots w_nx_n \geq T$

Example of a classification task that is NOT linearly separable



Does a solution to the perceptron function exist?
Is it unique? Is it perfect?

$f(x_0 \ldots x_n)$: $w_0x_0 + w_1x_1 + \ldots w_nx_n \geq T$

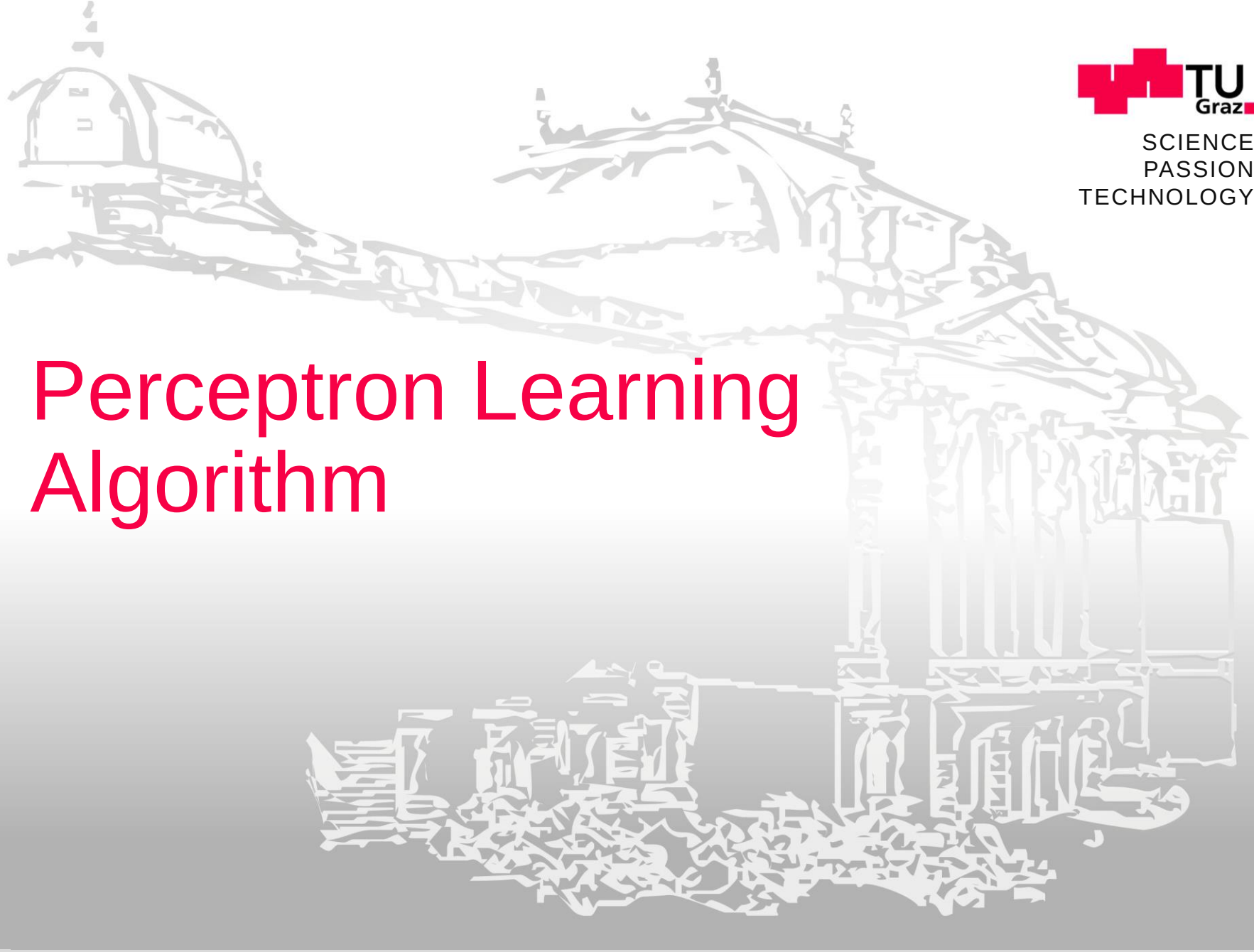
Machine learning for classification

We may not find the perfect solution to the perceptron function for all given examples.

Machine learning ≠ Equation solving

Remember:

- Given a **training set** of example pairs (x_1,y_1), (x_2,y_2) … (x_n,y_n) – *that's just a set of examples, not the full specification!*
- Where each y_i was generated by an unknown function $y=f_{TRUE}(x)$
  - Discrete number of values for y_i
- Discover a function f (**hypothesis**) that approximates the true function f_{TRUE} – *we may not even be close to right, as we have a lot of approximations on the way (vector representation, choice of examples, choice of ANN architecture, choice of cost function, etc), but the DECISION should be, overall, good enough.*



TU Graz

SCIENCE
PASSION
TECHNOLOGY

Perceptron Learning Algorithm

ISDS
INSTITUTE OF INTERACTIVE SYSTEMS AND DATA SCIENCE

Perceptron Learning Algorithm – Main Part

$f(x_0 \ldots x_n)$: $w_0x_0 + w_1x_1 + \ldots w_nx_n \geq T$

f is our hypothesis!

d_j= desired output of perceptron for example *j*

D = training set of example pairs $(\boldsymbol{x_j}, d_j)$

$y_j(t)$ = actual output of perceptron for example *j* at time *t*

r = learning rate

1. Initialize weights, typically to small values
2. For each example *j* in the training set D perform the following steps:
   1. $y_j(t) = (\mathbf{w}(t) * \mathbf{x}_j) \geq T$
   2. For all i=0…n: $w_i(t+1)=w_i(t) + r * (d_j-y_j(t))* x_{j,i}$

   Weights stays the same if the output $y_j(t)$ is the same as the desired output d_j

   Overall value needs to increase if actual output $y_j(t)$ is smaller than desired -> increase weights at positive inputs, decrease weights at negative inputs.

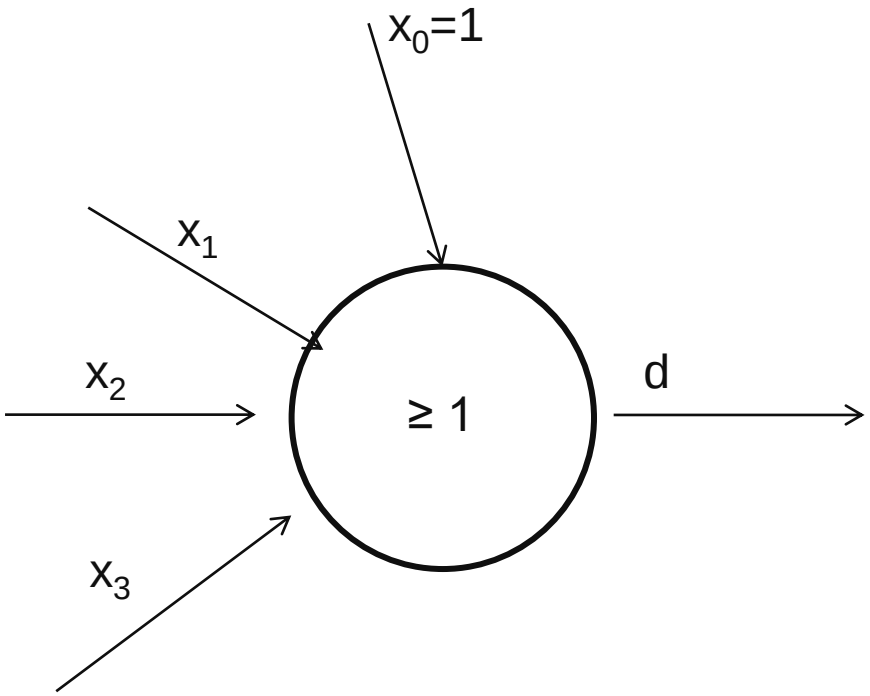
   Overall value needs to decrease if actual output $y_j(t)$ is larger than desired -> decrease weights at positive inputs, increase weights at negative inputs.

Perceptron Learning – Stopping Criteria

- Stop when the average error over seen examples is below a pre-defined threshold
  - useful in offline learning, and when classes are linearly separable (*in general: separable with the type of function we are using as f*)
- OR: stop after a pre-determined number of iterations
  - useful when separability is not certain
  - We need to think which weights should be returned? Last set of weights is not necessarily the best one

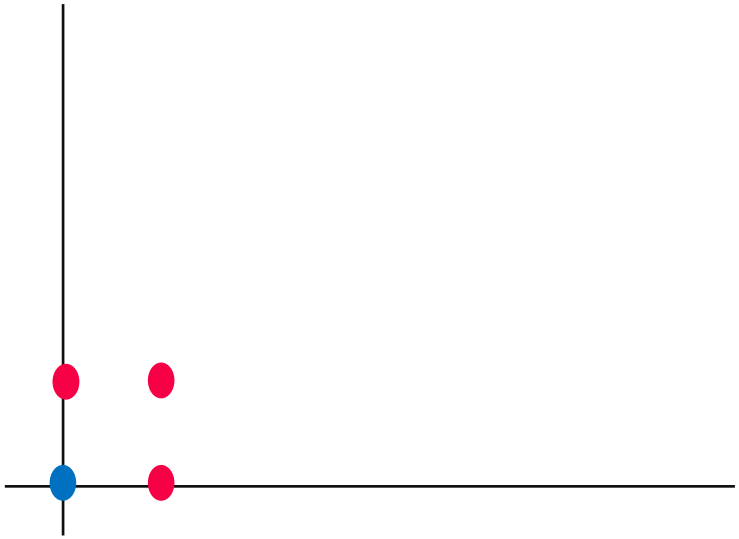
Example: Learning the OR function

Boolean OR: $f(x_0,x_1,x_2,x_3) = x_1$ OR x_2 OR x_3



x_1	x_2	x_3	d
1	1	1	1
1	1	0	1
1	0	1	1
1	0	0	1
0	1	1	1
0	1	0	1
0	0	1	1
0	0	0	0

2 dimensional OR function – is this linearly separable?



Example: Learning the OR function

Choice: $f(x_0, x_1, x_2, x_3) = w_0x_0 + w_1x_1 + w_2x_2 + w_3x_3 \geq 1$

x_0=*1*, ***r=0.5***, $w_i(0)$=*0*

Weight update formula from slide 13: $w_i(t+1)=w_i(t) + r * (d_j-y_j(t))* x_{j,i}$

t	Ex #	x_1	x_2	x_3	$w_0(t)$	$w_1(t)$	$w_2(t)$	$w_3(t)$	$y_{ex}(t)$	Correct (yes/no)
t=0	1	1	1	1						
t=1	2	1	1	0						
t=2	3	1	0	1						
t=3	4	1	0	0						

Discussion: Perceptron learning algorithm properties

+ Convergence towards a solution is guaranteed if and only if the training set is linearly separable.
– Doesn't slowly converge towards solution (jumps around), and may not find best solution

Variants

- Batch learning – change weights only after a batch of training examples have been seen.
- Keep the best already seen solution (=weights) in memory
- Voting between multiple perceptrons that are trained
- Include optimization criteria such as maximizing distance of separation function to both classes

Learning Goals

Understand

- The machine learning task “classification”, especially in comparison to (linear) equation solving.

Describe

- The perceptron learning algorithm

Be able to

- Carry out the perceptron learning algorithm

Recommended Reading

- https://ecee.colorado.edu/~ecen4831/lectures/NNet2.html - Short reading, highly recommended!

And two classic books:

- the AI classic book by Stuart Russell & Peter Norvig (http://myweb.sabanciuniv.edu/rdehkharghani/files/2016/02/Prentice-Hall-Series-in-Artificial-Intelligence-Stuart-Russell-Peter-Norvig-Artificial-Intelligence_-A-Modern-Approach-Prentice-Hall-2010.pdf)
- Another classic machine learning book by Tom Mitchell (here I took especially the Perceptron figure from): http://myweb.sabanciuniv.edu/rdehkharghani/files/2016/02/Prentice-Hall-Series-in-Artificial-Intelligence-Stuart-Russell-Peter-Norvig-Artificial-Intelligence_-A-Modern-Approach-Prentice-Hall-2010.pdf

Exercise 16

Example: Learning the OR function

Choice: $f(x_0, x_1, x_2, x_3) = w_0x_0 + w_1x_1 + w_2x_2 + w_3x_3 \geq 1$

*x_0=1, **r=0.1**, $w_i(0)$=0*

Weight update from slide 13: $w_i(t+1)=w_i(t) + r * (d_j-y_j(t))* x_{j,i}$

t	Ex #	x_1	x_2	x_3	$w_0(t)$	$w_1(t)$	$w_2(t)$	$w_3(t)$	$y_{ex}(t)$	Correct (yes/no)
t=0	1									
t=1	2									
t=2	3									
t=3	4									

Exercise 17

Example: Learning the OR function

Choice: $f(x_0, x_1, x_2, x_3) = w_0 x_0 + w_1 x_1 + w_2 x_2 + w_3 x_3 \geq 1$

x_0=1, r=0,5, ***$w_i(0)$=1***

Weight update from slide 13: $w_i(t+1)=w_i(t) + r * (d_j - y_j(t)) * x_{j,i}$

t	Ex #	x_1	x_2	x_3	$w_0(t)$	$w_1(t)$	$w_2(t)$	$w_3(t)$	$y_{ex}(t)$	Correct (yes/no)
t=0	1									
t=1	2									
t=2	3									
t=3	4									