Introduction To Machine Learning

Krisztian Buza

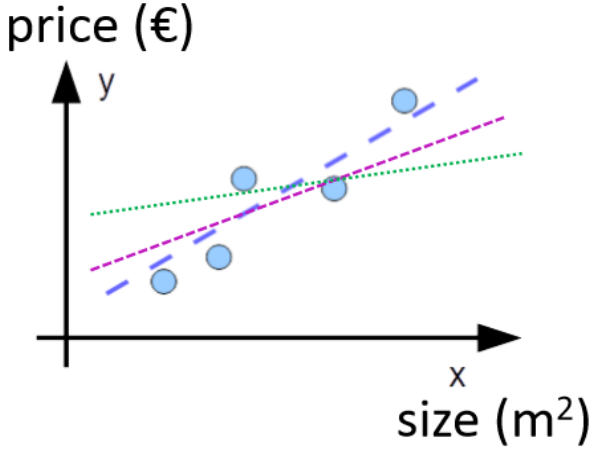
Department of Artificial Intelligence
Eötvös Loránd University
Budapest, Hungary

Krisztian Buza, buza@inf.elte.hu
http://biointelligence.hu

What is behind „Intelligent“ Behavior in Engineering Applications?

high computational power, sometimes coupled with a big database

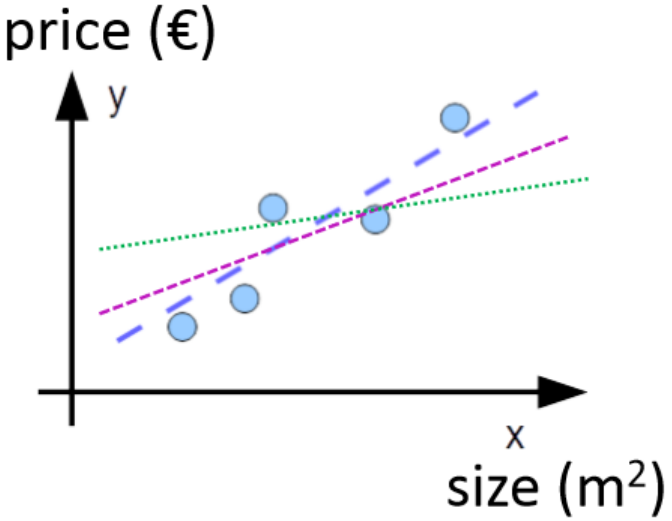
Mathematical model with many parameters

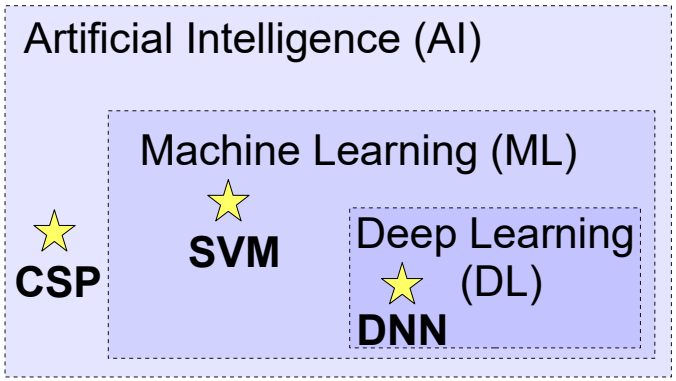


Rubic cube in the top left: from Wikipedia, author: „Alvaro qc“
[CC BY 3.0 (https://creativecommons.org/licenses/by/3.0)]
Server in the bottom right: from Wikipedia, Jfreyre – own work,
CC BY-SA 3.0, https://commons.wikimedia.org/w/index.php?curid=2817411

Krisztian Buza, buza@inf.elte.hu
http://biointelligence.hu

Machine Learning: an Extremly Simple Example



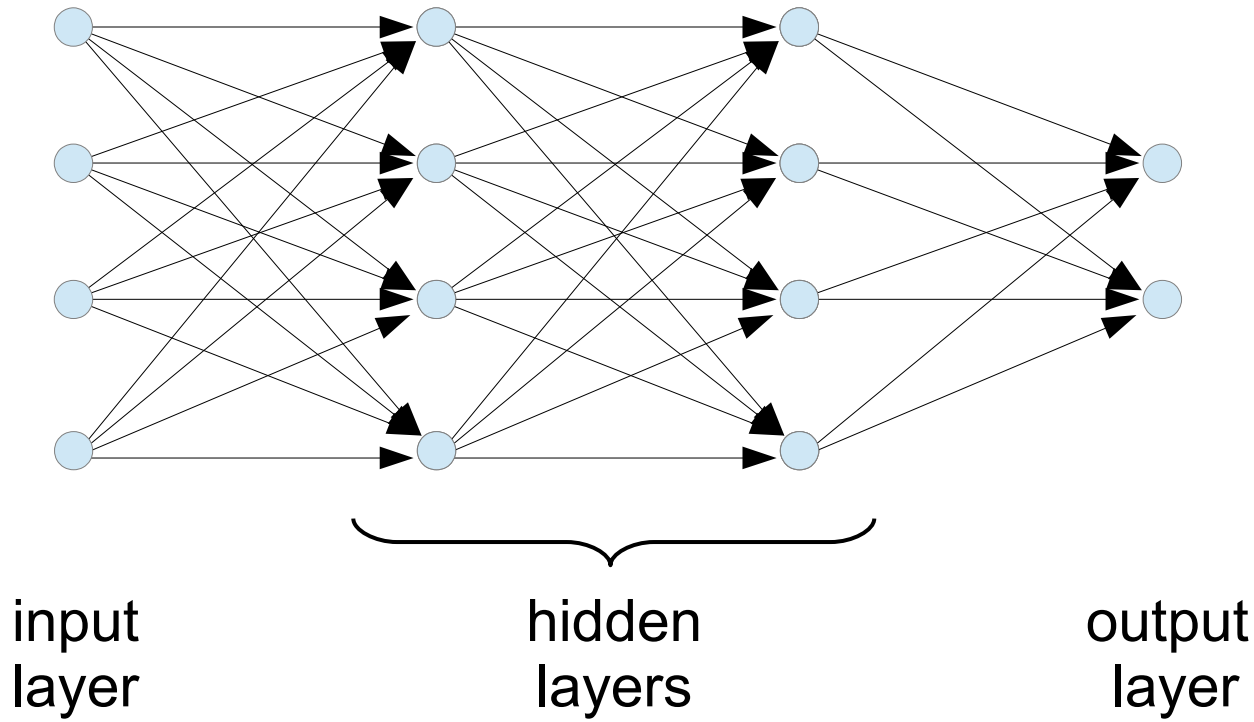


- Mathematical model:
  $y = m\,x + b$
- Parameters of the model: $m,\ b$
- *Learning* refers to finding the model parameters in an automated way
- This is true for more complex models as well (e.g. neural networks)
- In realistic applications, the number of model parametes is high (up to millions)
- Recent success stories: recognition of skin cancer, Go, self-driving cars...

Krisztian Buza, buza@inf.elte.hu
http://biointelligence.hu

How does a Neural Network Work?

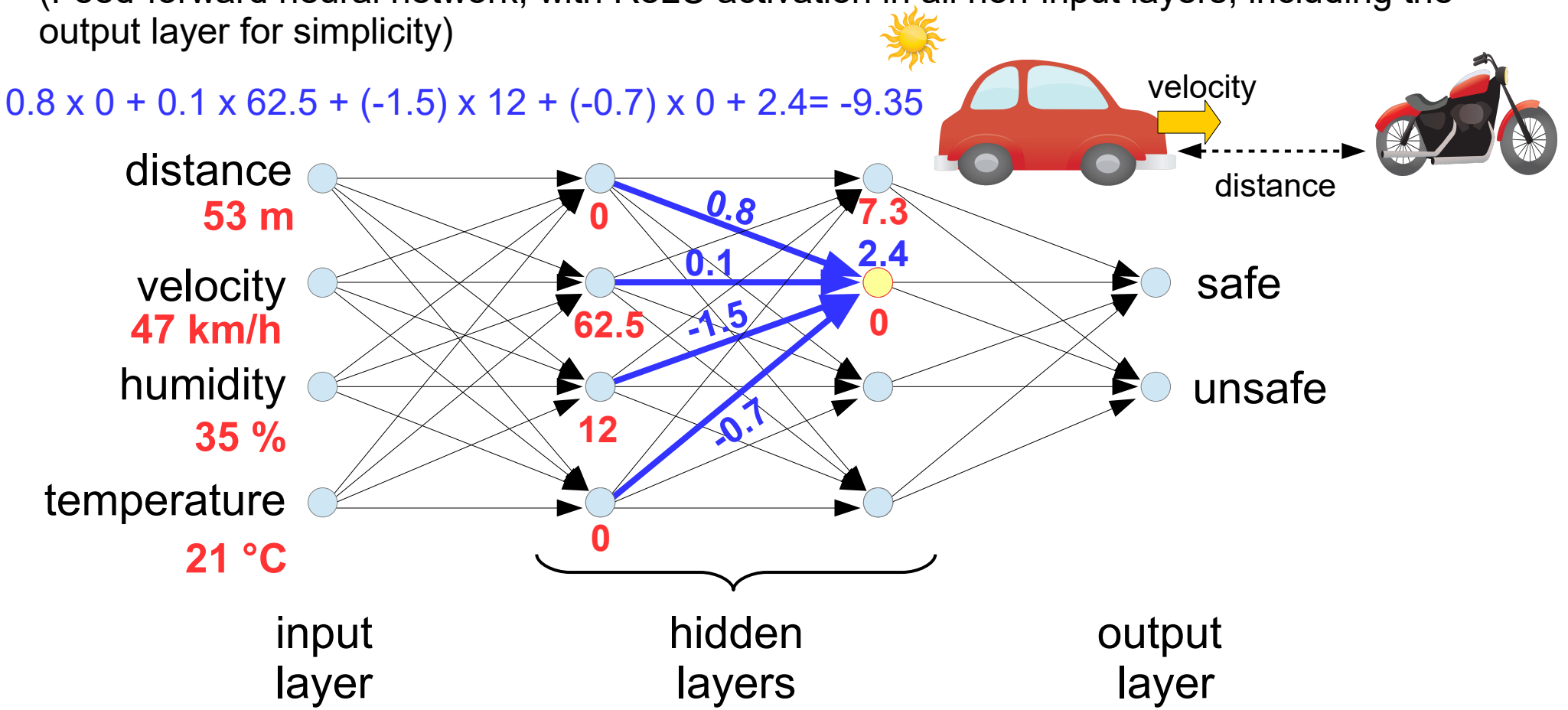
(Feed-forward neural network, with ReLU activation in all non-input layers, including the output layer for simplicity)



Krisztian Buza, buza@inf.elte.hu
http://biointelligence.hu

How does a Neural Network Work?

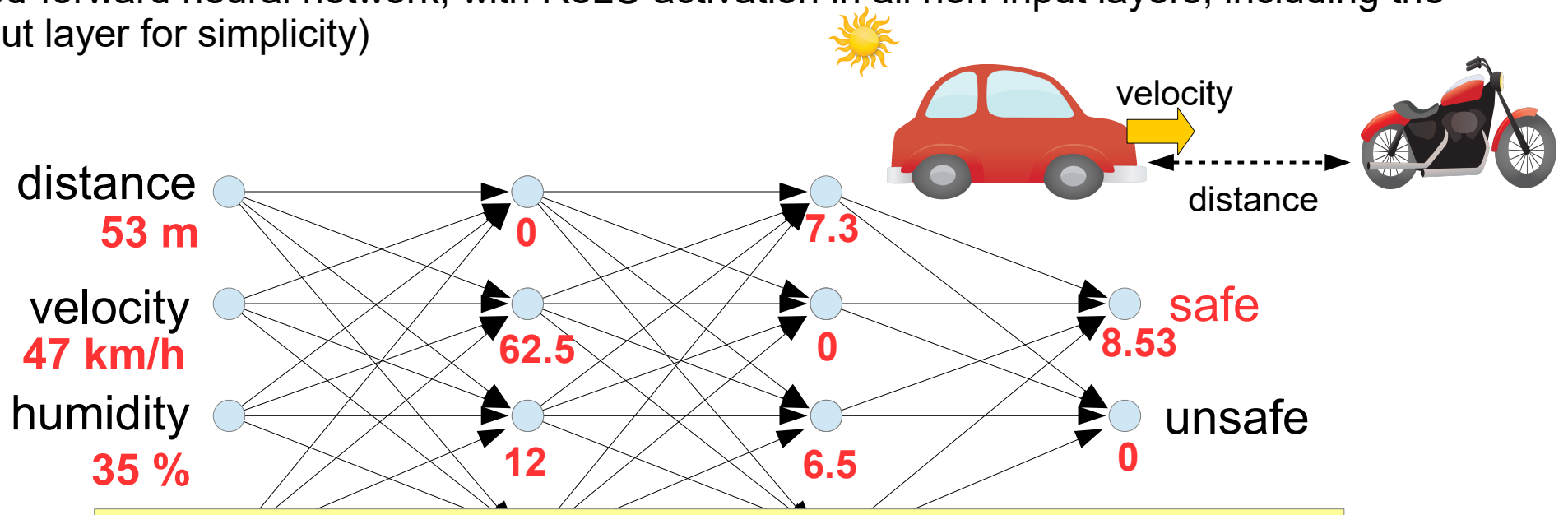
(Feed-forward neural network, with ReLU activation in all non-input layers, including the output layer for simplicity)



Krisztian Buza, buza@inf.elte.hu
http://biointelligence.hu

How does a Neural Network Work?

(Feed-forward neural network, with ReLU activation in all non-input layers, including the output layer for simplicity)



The weights (parameters) of the network are determined ~~by the learning algorithm, NOT by a human expert~~ by an optimization algorithm, such as batch gradient descent!

Krisztian Buza, buza@inf.elte.hu
http://biointelligence.hu

Slight modifications of street signs can fool machine learning algorithms

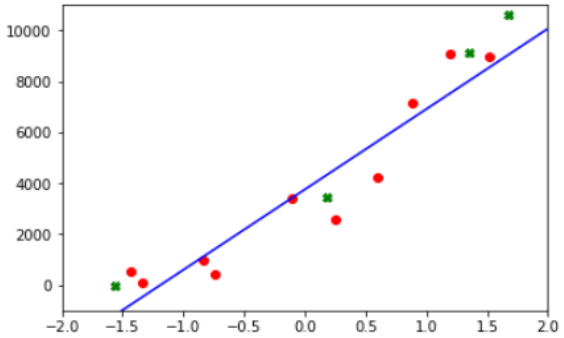
(This image is not from the paper, but it is a self-made photo about a real stop sign.)

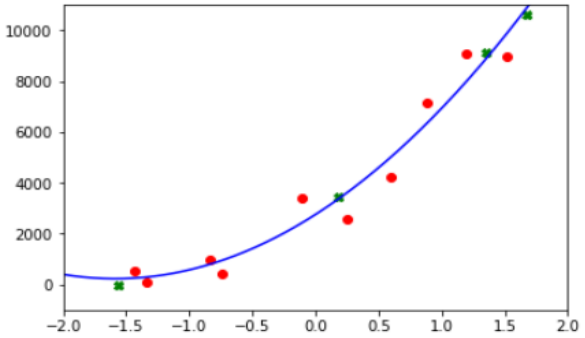
„The Stop sign is misclassified into our target class of Speed Limit 45 in 100% of the images taken according to our evaluation methodology.“

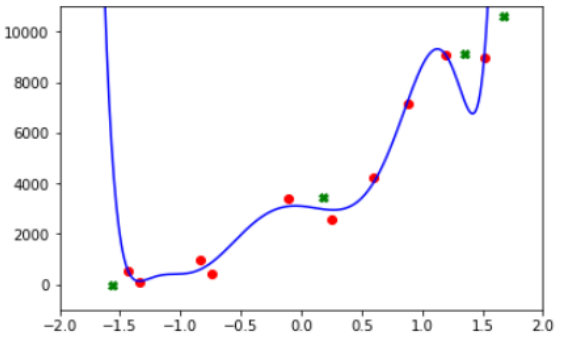
https://spectrum.ieee.org/cars-that-think/transportation/sensors/slight-street-sign-modifications-can-fool-machine-learning-algorithms

Krisztian Buza, buza@inf.elte.hu
http://biointelligence.hu

Root Mean Square Error of the Models







on training data:

1085.88	745.50	248.66

on new data (test data):

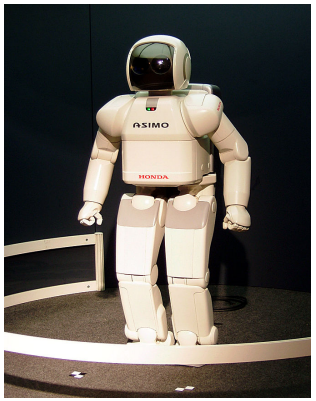
1202.28	209.73	13616.00

Krisztian Buza, buza@inf.elte.hu
http://biointelligence.hu

ANI versus AGI

Artificial Intelligence

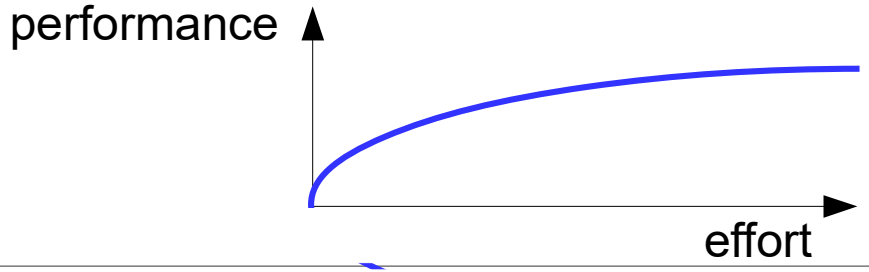
Artificial General Intelligence (AGI)



https://commons.wikimedia.org/wiki/File:HONDA_ASIMO.jpg

Artificial Narrow Intelligence (ANI)

- recognition of skin cancer
- self-driving cars
- playing chess, Go,...
- solving puzzles like the Rubic's cube
- detection of spam
- ...



Krisztian Buza, buza@inf.elte.hu
http://biointelligence.hu

Most important take-home-messages

- 1. Behavior that seems to be intelligent is based on a mathematical model and high computational power.
- 2. Machine learning refers for finding appropriate parameters of the model using a big dataset.
- 3. Most of the breakthroughs of the recent years belong to the category of „artificial narrow intelligence“.
- 4. You must use **new** data (=test data) to evaluate the performace of a model, such as the accuracy of a neural network.
- 5. Determining the appropriate values of hyperparameters is part of the training process, therefore, you are **not allowed** to use the test data to determine the appropriate values of hyperparameters.
- 6. Feed-forward neural networks with at least one hidden layer with non-linear activation are universal function approximators.

Krisztian Buza, buza@inf.elte.hu
http://biointelligence.hu

What to Expect in the Test?

Krisztian Buza, buza@inf.elte.hu
http://biointelligence.hu

How to Pass this Course?

- Quick tests (1 or 2 questions) both at the **beginning** as well as at the **end** of the lectures
  - Week 2 – 6, Week 9
  - Week 1: only at the end of the lecture
  - In total: 13 short test, 2 points each → 26 points
- Programming homework
  (week 8, team work, 1 team = 2 or 3 students) → 24 points
- Presentation about a topic related to social, economical or moral aspects of AI (week 11, team work, 1 team = 2 or 3 students, ≈10 minutes per team) → 20 points
- Test (week 13)
  - most essential concepts and algorithms related to the mathematical background of machine learning → 30 points

Krisztian Buza, buza@inf.elte.hu
http://biointelligence.hu

What to expect in the test?

- Basically, **everything** can appear in the test that we talked about during the lectures, however, while preparing, you may want to focus on these topics:
  - Calculation of (partial) derivatives, gradients
  - Dot product, matrix multiplication, convolution, max pooling
  - Calculation of RMSE (root mean sqaured error)
  - Constuction of a neural net that implements a boolean function
  - Theoretical questions, such as:
    - What is the difference between L_1 and L_2 regularisation?
    - ...stochastic gradient descent, (mini) batch gradient descent?
    - What is overfitting, drop-out...?

Krisztian Buza, buza@inf.elte.hu
http://biointelligence.hu

Derivatives

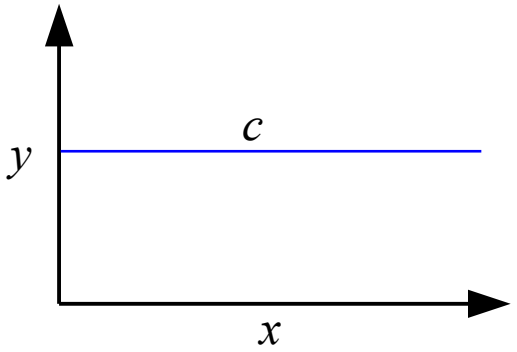
- The slope of function f (or $f(x)$)corresponds to its derivative, denoted as

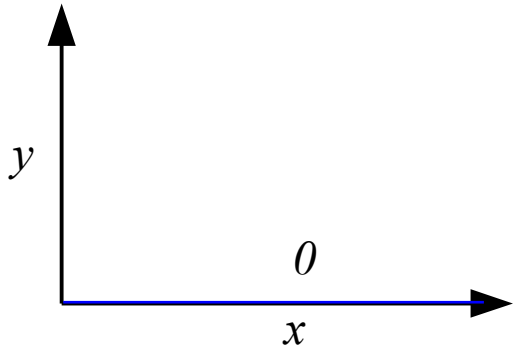
$$f', \quad \frac{d}{dx} f(x) \quad \text{or} \quad \frac{\partial f}{\partial x} \quad \text{(in case of multiple variables)}$$

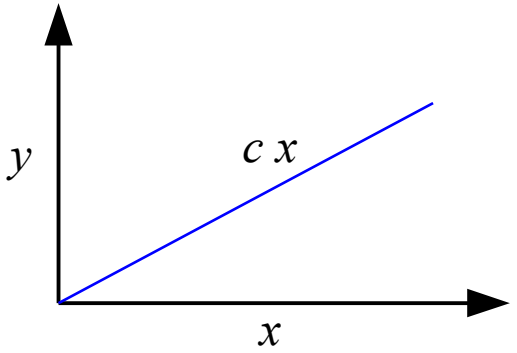
f	f'
c (constant)	0
$c\,x$ (where c is a constant)	c
x^2	$2x$
x^c (where c is a constant)	cx^{c-1}
$h(x) + g(x)$	$h'(x) + g'(x)$
$h(g(x))$	$h'(g(x))\,g'(x)$

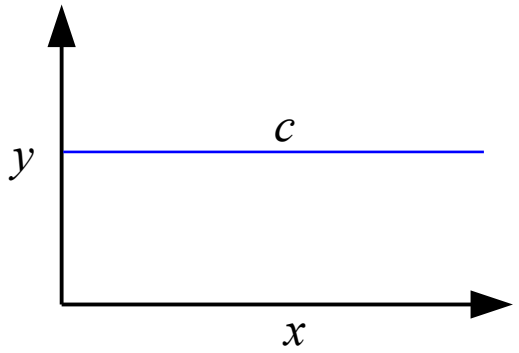
Krisztian Buza, buza@inf.elte.hu
http://biointelligence.hu

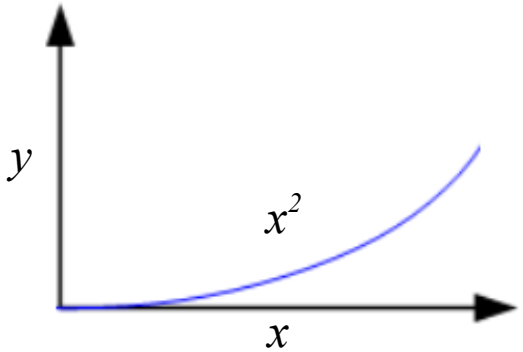
Illustration: Some Functions and Their Derivatives

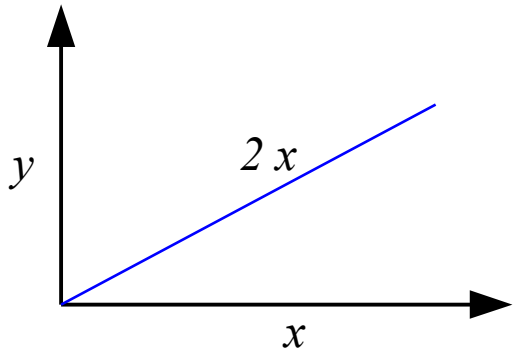












Can you justify that the derivative of x^2 is really $2x$?

Krisztian Buza, buza@inf.elte.hu
http://biointelligence.hu

Partial Derivatives in Case of m Variables

- Sum of Squared Errors: $E = \sum_{i=1}^{n} (\hat{y}_i - y_i)^2$

$$\hat{y}_i = w^{(1)} x_i^{(1)} + w^{(2)} x_i^{(2)} + \ldots + w^{(m)} x_i^{(m)} = \sum_{j=1}^{m} w^{(j)} x_i^{(j)}$$

- Partial Derivative w.r.t. $w^{(1)}$: $\frac{\partial E}{\partial w^{(1)}} = \sum_{i=1}^{n} 2 x_i^{(1)} (\hat{y}_i - y_i)$

...

- Partial Derivative w.r.t. $w^{(j)}$: $\frac{\partial E}{\partial w^{(j)}} = \sum_{i=1}^{n} 2 x_i^{(j)} (\hat{y}_i - y_i)$

Krisztian Buza, buza@inf.elte.hu
http://biointelligence.hu

Gradient

$$\nabla E = \left(\frac{\partial E}{\partial w^{(1)}}, \ldots, \frac{\partial E}{\partial w^{(m)}} \right)$$

Krisztian Buza, buza@inf.elte.hu
http://biointelligence.hu

Possible task

Calculate the derivative of this function w.r.t. w

$$f(w) = (w\,x + b)^3$$

Krisztian Buza, buza@inf.elte.hu
http://biointelligence.hu

Dot Product

Krisztian Buza, buza@inf.elte.hu
http://biointelligence.hu

Dot Product a.k.a. Scalar Product

$$\vec{w} = \mathbf{w} = (w^{(1)}, w^{(2)}, \ldots, w^{(m)})$$

$$\vec{x} = \mathbf{x} = (x^{(1)}, x^{(2)}, \ldots, x^{(m)})$$

$$\vec{w}\vec{x} = \mathbf{w}\mathbf{x} = \sum_{j=1}^{m} w^{(j)} x^{(j)}$$

Example:
Let us calculate the dot product of these vectors:

$\boldsymbol{v}_1 = (3, 4, 2, 0)$

$\boldsymbol{v}_2 = (5, 0, 4, 8)$

Krisztian Buza, buza@inf.elte.hu
http://biointelligence.hu

Dot Product a.k.a. Scalar Product

$$\vec{w} = \mathbf{w} = (w^{(1)}, w^{(2)}, \ldots, w^{(m)})$$

$$\vec{x} = \mathbf{x} = (x^{(1)}, x^{(2)}, \ldots, x^{(m)})$$

$$\vec{w}\vec{x} = \mathbf{w}\mathbf{x} = \sum_{j=1}^{m} w^{(j)} x^{(j)}$$

Example:
Let us calculate the dot product of these vectors:

$\boldsymbol{v}_1 = (3, 4, 2, 0)$
$\boldsymbol{v}_2 = (5, 0, 4, 8)$

$\boldsymbol{v}_1 \boldsymbol{v}_2 = 3\text{x}5$

Krisztian Buza, buza@inf.elte.hu
http://biointelligence.hu

Dot Product a.k.a. Scalar Product

$$\vec{w} = \mathbf{w} = (w^{(1)}, w^{(2)}, \ldots, w^{(m)})$$

$$\vec{x} = \mathbf{x} = (x^{(1)}, x^{(2)}, \ldots, x^{(m)})$$

$$\vec{w}\vec{x} = \mathbf{w}\mathbf{x} = \sum_{j=1}^{m} w^{(j)} x^{(j)}$$

Example:
Let us calculate the dot product of these vectors:

$\boldsymbol{v}_1 = (3, 4, 2, 0)$
$\boldsymbol{v}_2 = (5, 0, 4, 8)$

$\boldsymbol{v}_1 \boldsymbol{v}_2 = 3\text{x}5 + 4\text{x}0$

Krisztian Buza, buza@inf.elte.hu
http://biointelligence.hu

Dot Product a.k.a. Scalar Product

$$\vec{w} = \mathbf{w} = (w^{(1)}, w^{(2)}, \ldots, w^{(m)})$$

$$\vec{x} = \mathbf{x} = (x^{(1)}, x^{(2)}, \ldots, x^{(m)})$$

$$\vec{w}\vec{x} = \mathbf{w}\mathbf{x} = \sum_{j=1}^{m} w^{(j)} x^{(j)}$$

Example:
Let us calculate the dot product of these vectors:

$\boldsymbol{v}_1 = (3, 4, 2, 0)$
$\boldsymbol{v}_2 = (5, 0, 4, 8)$

$\boldsymbol{v}_1 \boldsymbol{v}_2 = 3\text{x}5 + 4\text{x}0 + 2\text{x}4$

Krisztian Buza, buza@inf.elte.hu
http://biointelligence.hu

Dot Product a.k.a. Scalar Product

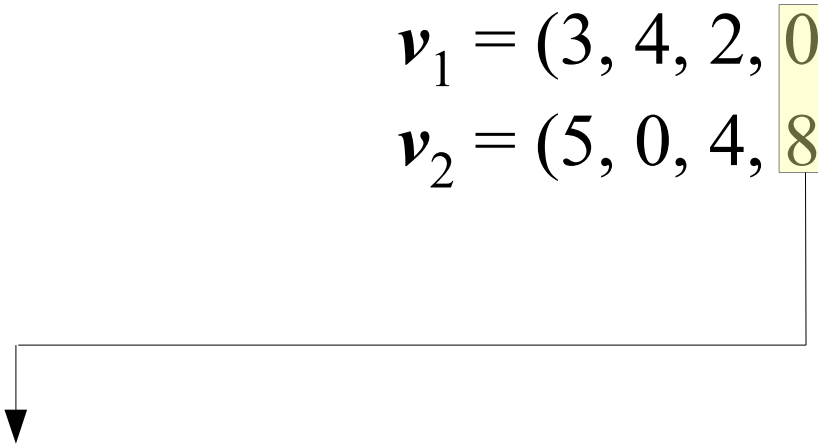
$$\vec{w} = \mathbf{w} = (w^{(1)}, w^{(2)}, \ldots, w^{(m)})$$

$$\vec{x} = \mathbf{x} = (x^{(1)}, x^{(2)}, \ldots, x^{(m)})$$

$$\vec{w}\vec{x} = \mathbf{w}\mathbf{x} = \sum_{j=1}^{m} w^{(j)} x^{(j)}$$

Example:
Let us calculate the dot product of these vectors:

$\boldsymbol{v}_1 = (3, 4, 2, 0)$
$\boldsymbol{v}_2 = (5, 0, 4, 8)$



$\boldsymbol{v}_1 \boldsymbol{v}_2 = 3\text{x}5 + 4\text{x}0 + 2\text{x}4 + 0\text{x}8$

Krisztian Buza, buza@inf.elte.hu
http://biointelligence.hu

Dot Product a.k.a. Scalar Product

$$\vec{w} = \mathbf{w} = (w^{(1)}, w^{(2)}, \ldots, w^{(m)})$$

$$\vec{x} = \mathbf{x} = (x^{(1)}, x^{(2)}, \ldots, x^{(m)})$$

$$\vec{w}\vec{x} = \mathbf{w}\mathbf{x} = \sum_{j=1}^{m} w^{(j)} x^{(j)}$$

Example:
Let us calculate the dot product of these vectors:

$\boldsymbol{v}_1 = (3, 4, 2, 0)$
$\boldsymbol{v}_2 = (5, 0, 4, 8)$

$$\boldsymbol{v}_1 \, \boldsymbol{v}_2 = 3\text{x}5 + 4\text{x}0 + 2\text{x}4 + 0\text{x}8 = 15 + 0 + 8 + 0 = 23$$

Krisztian Buza, buza@inf.elte.hu
http://biointelligence.hu

Multiplication of Matrices

Krisztian Buza, buza@inf.elte.hu
http://biointelligence.hu

Example: Multiplication of 2x2 matrices

$$\begin{pmatrix} 3 & 8 \\ 2 & 0 \end{pmatrix} \quad \begin{pmatrix} 0 & 2 \\ 5 & 7 \end{pmatrix}$$

Krisztian Buza, buza@inf.elte.hu
http://biointelligence.hu

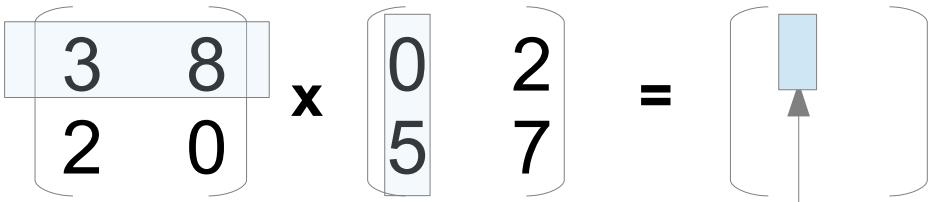
Example: Multiplication of 2x2 matrices

$$\begin{pmatrix} 3 & 8 \\ 2 & 0 \end{pmatrix} \times \begin{pmatrix} 0 & 2 \\ 5 & 7 \end{pmatrix} = \begin{pmatrix} & \\ & \end{pmatrix}$$

Krisztian Buza, buza@inf.elte.hu
http://biointelligence.hu

Example: Multiplication of 2x2 matrices

„row times column“

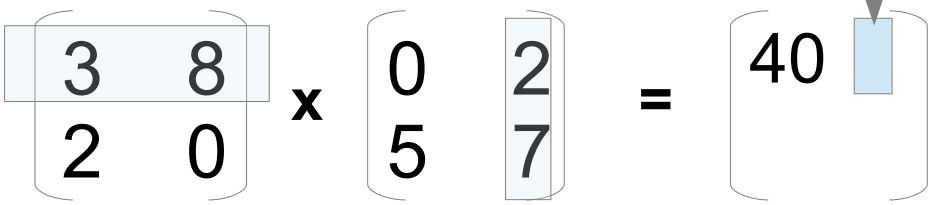


(3 x 0) + (8 x 5) = 0 + 40 = 40

Krisztian Buza, buza@inf.elte.hu
http://biointelligence.hu

Example: Multiplication of 2x2 matrices

„row times column“ (3 x 2) + (8 x 7) = 6 + 56 = 62



Krisztian Buza, buza@inf.elte.hu
http://biointelligence.hu

Example: Multiplication of 2x2 matrices

„row times column“

$$\begin{pmatrix} 3 & 8 \\ 2 & 0 \end{pmatrix} \times \begin{pmatrix} 0 & 2 \\ 5 & 7 \end{pmatrix} = \begin{pmatrix} 40 & 62 \\ \square & \end{pmatrix}$$

$$(2 \times 0) + (0 \times 5) = 0 + 0 = 0$$

Krisztian Buza, buza@inf.elte.hu
http://biointelligence.hu

Example: Multiplication of 2x2 matrices

„row times column“

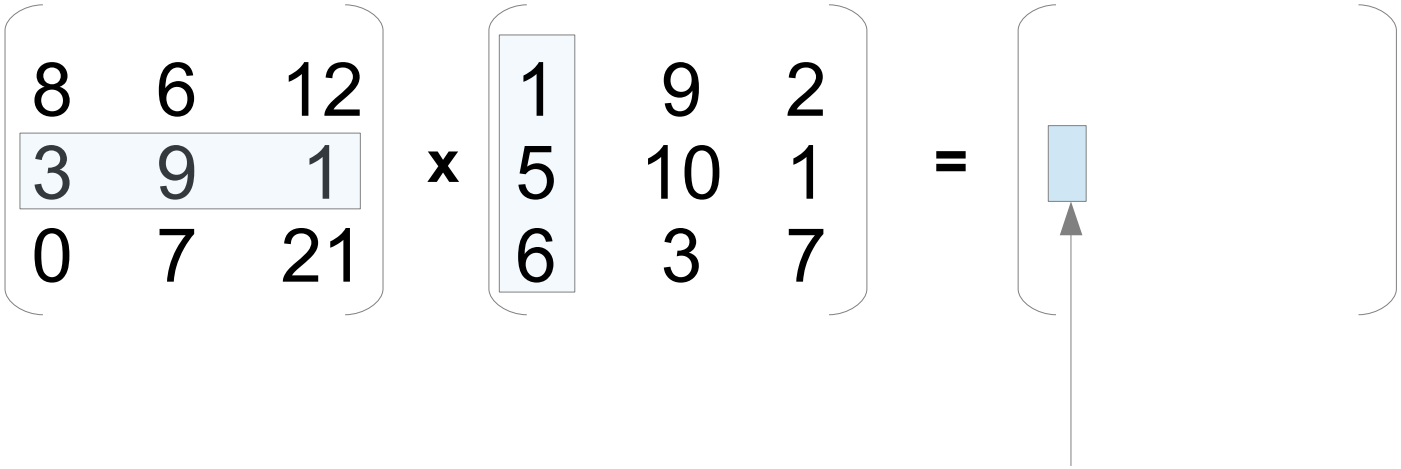
$$\begin{pmatrix} 3 & 8 \\ 2 & 0 \end{pmatrix} \times \begin{pmatrix} 0 & 2 \\ 5 & 7 \end{pmatrix} = \begin{pmatrix} 40 & 62 \\ 0 & \square \end{pmatrix}$$

$$(2 \times 2) + (0 \times 7) = 4 + 0 = 4$$

Krisztian Buza, buza@inf.elte.hu
http://biointelligence.hu

Example: Multiplication of 3x3 matrices

„row times column“

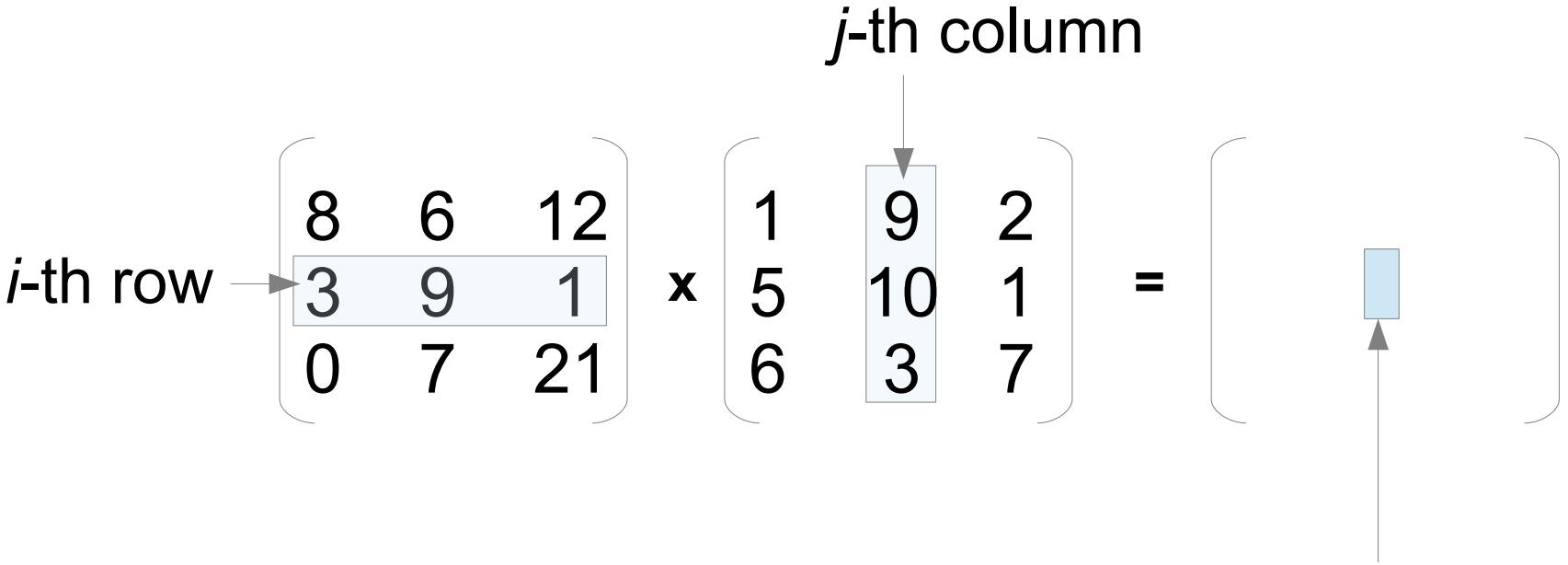


(3x1)+(9x5)+(1x6)=3+45+6=54

Krisztian Buza, buza@inf.elte.hu
http://biointelligence.hu

Example: Multiplication of 3x3 matrices

„row times column“



Element in the *i*-th row and *j*-th column:
product of the *i*-th row of the first matrix
and the *j*-th column of the second matrix

Krisztian Buza, buza@inf.elte.hu
http://biointelligence.hu

Convolution

Krisztian Buza, buza@inf.elte.hu
http://biointelligence.hu

Convolution*

Input of the convolution (time series):

-0.8	-0.5	-0.2	0.2	0.6	0.8	0.9	1.0	0.9	0.7	0.2	-0.3	-0.9	-0.2	0.5	0.6

* Remark: while being conceptually the same as traditional convolution in mathematics, the convolution used in neural networks is slightly different in terms of technical details.

Krisztian Buza, buza@inf.elte.hu
http://biointelligence.hu

Convolution*

Input of the convolution (time series):

-0.8	-0.5	-0.2	0.2	0.6	0.8	0.9	1.0	0.9	0.7	0.2	-0.3	-0.9	-0.2	0.5	0.6

Filter (i.e., a set of weights)

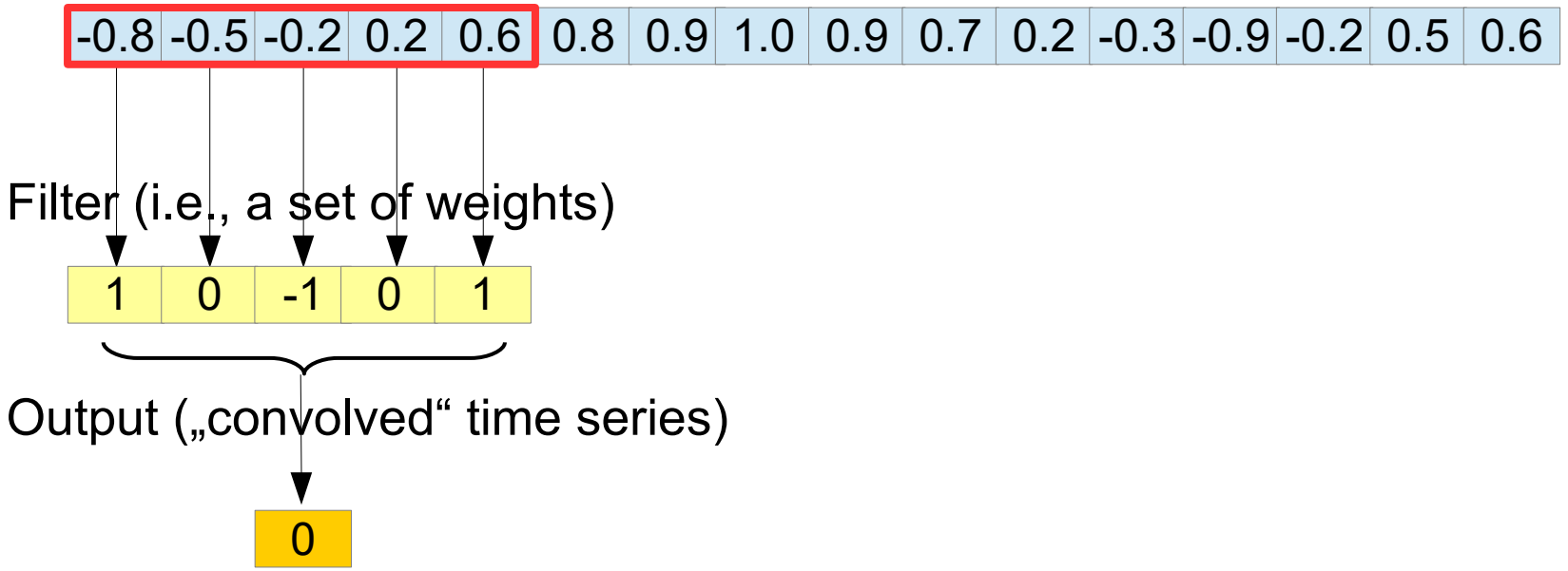
1	0	-1	0	1

* Remark: while being conceptually the same as traditional convolution in mathematics, the convolution used in neural networks is slightly different in terms of technical details.

Krisztian Buza, buza@inf.elte.hu
http://biointelligence.hu

Convolution*

Input of the convolution (time series):



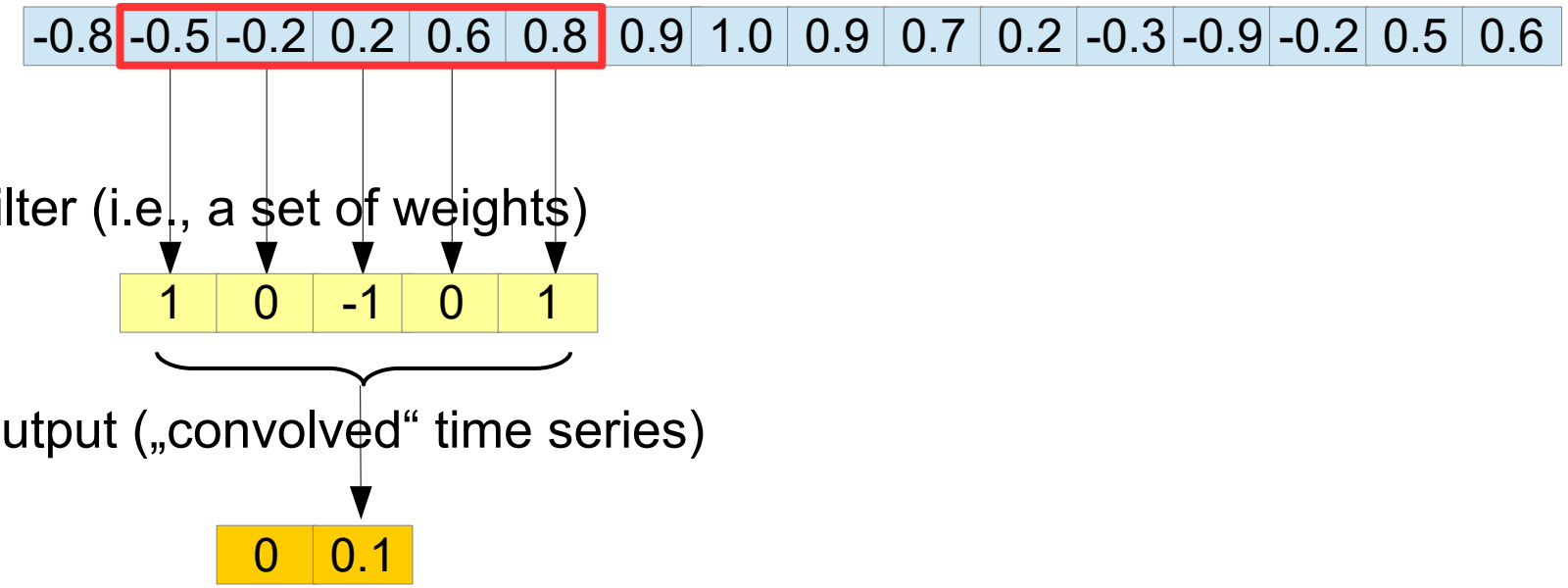
(-0.8) x 1 + (-0.5) x 0 + (-0.2) x (-1) + 0.2 x 0 + 0.6 x 1 = 0

* Remark: while being conceptually the same as traditional convolution in mathematics, the convolution used in neural networks is slightly different in terms of technical details.

Krisztian Buza, buza@inf.elte.hu
http://biointelligence.hu

Convolution*

Input of the convolution (time series):



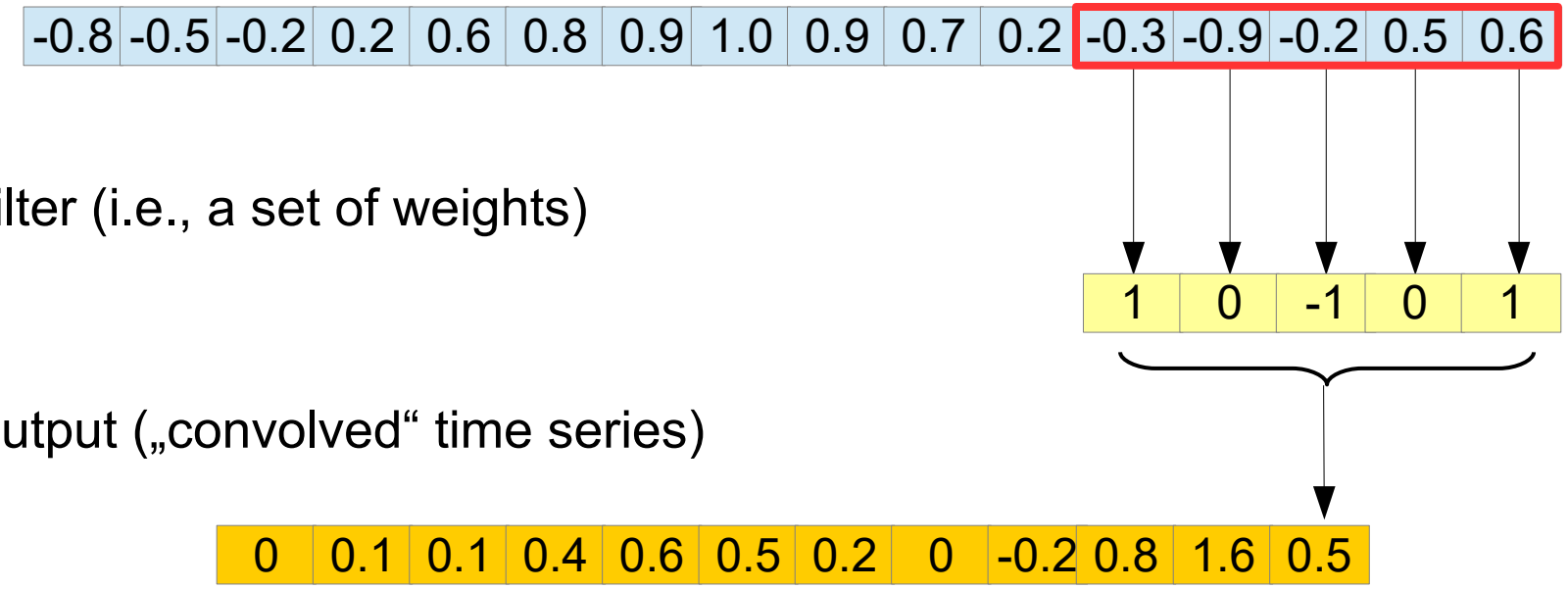
(-0.5) x 1 + (-0.2) x 0 + 0.2 x (-1) + 0.6 x 0 + 0.8 x 1 = -0.1

* Remark: while being conceptually the same as traditional convolution in mathematics, the convolution used in neural networks is slightly different in terms of technical details.

Krisztian Buza, buza@inf.elte.hu
http://biointelligence.hu

Convolution*

Input of the convolution (time series):



* Remark: while being conceptually the same as traditional convolution in mathematics, the convolution used in neural networks is slightly different in terms of technical details.

Krisztian Buza, buza@inf.elte.hu
http://biointelligence.hu

Convolution*

Input of the convolution (time series):

0	0	-0.8	-0.5	-0.2	0.2	0.6	0.8	0.9	1.0	0.9	0.7	0.2	-0.3	-0.9	-0.2	0.5	0.6	0	0
0	0	0.9	0.3	0.1	-0.2	0.5	0.3	0.1	0	0.2	-0.1	-0.2	0.4	0.5	0.5	0.6	0.3	0	0

Filter (i.e., a set of weights)

1	0	-1	0	1
0.5	0.3	0.1	-0.2	-0.3

Output („convolved“ time series)

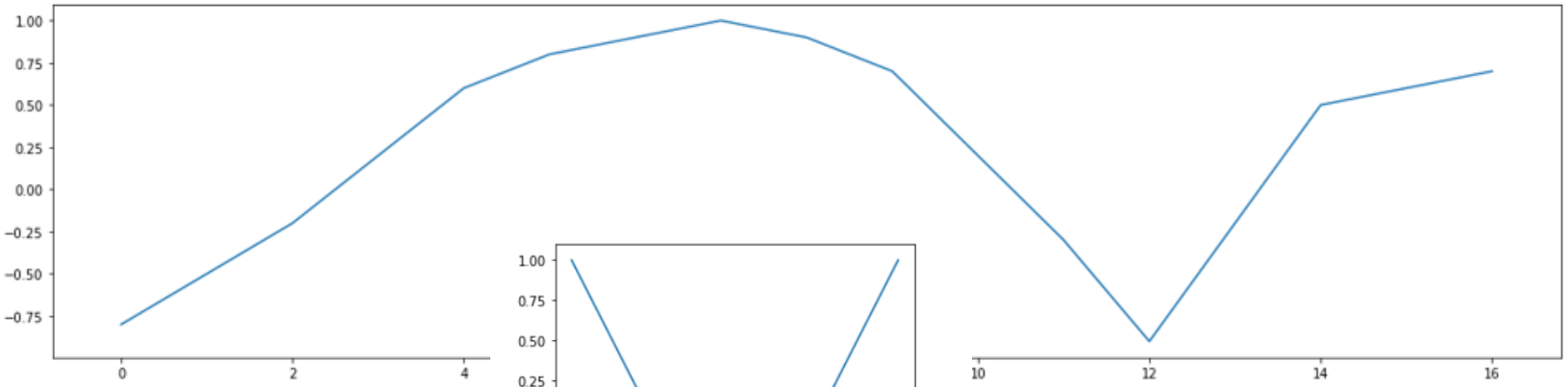
0.6	1.0	0.4	0.1	0.1	0.5	0.9	0.7	0.4	0	-0.4	0.5	1.4	0.7	-1.0	-0.3

* Remark: while being conceptually the same as traditional convolution in mathematics, the convolution used in neural networks is slightly different in terms of technical details.

Krisztian Buza, buza@inf.elte.hu
http://biointelligence.hu

Convolution*

In

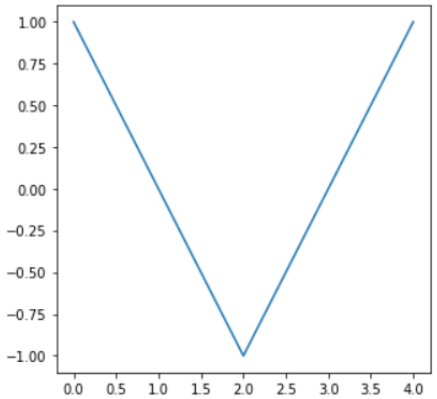


0 0

0 0

Fi

1 0 -1 0 1



Output („convolved“ time

0.6 0.7 0 0.1 0.1 0.4 0.6 0.5 0.2 0 -0.2 0.8 **1.6** 0.5 -1.4 -0.8

* Remark: while being conceptually the same as traditional convolution in mathematics, the convolution used in neural networks is slightly different in terms of technical details.

Krisztian Buza, buza@inf.elte.hu
http://biointelligence.hu

Convolution and Max Pooling*

Input of the convolution (time series):

0	0	-0.8	-0.5	-0.2	0.2	0.6	0.8	0.9	1.0	0.9	0.7	0.2	-0.3	-0.9	-0.2	0.5	0.6	0	0

Filter (i.e., a set of weights)

1	0	-1	0	1

Output („convolved“ time series)

0.6	0.7	0	0.1	0.1	0.4	0.6	0.5	0.2	0	-0.2	0.8	1.6	0.5	-0.7	-0.8

Max pooling

0.7	0.6	0.8	1.6

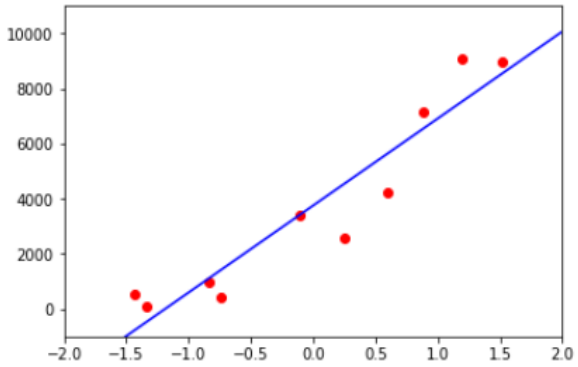
* Strictly speaking, max pooling has nothing to do with convolution, however, in convolutional neural networks (CNNs), the convolutional layer is often followed by a max pooling layer.

Krisztian Buza, buza@inf.elte.hu
http://biointelligence.hu

RMSE

Krisztian Buza, buza@inf.elte.hu
http://biointelligence.hu

Calculation of Root Mean Square Error



y	$\hat{y}$
549	-770
100	-470
976	1132
441	1433
3401	3435
2600	4537
4241	5638
7150	6539
9100	7541
9000	8542

n = 10

$$\frac{1}{10}((549-(-770))^2+(100-(-470))^2+$$
$$+(976-1132)^2+(441-1433)^2+$$
$$+(3401-3435)^2+(2600-4537)^2+$$
$$+(4241-5638)^2+(7150-6539)^2+$$
$$+(9100-7541)^2+(9000-8542)^2)$$

Krisztian Buza, buza@inf.elte.hu
http://biointelligence.hu

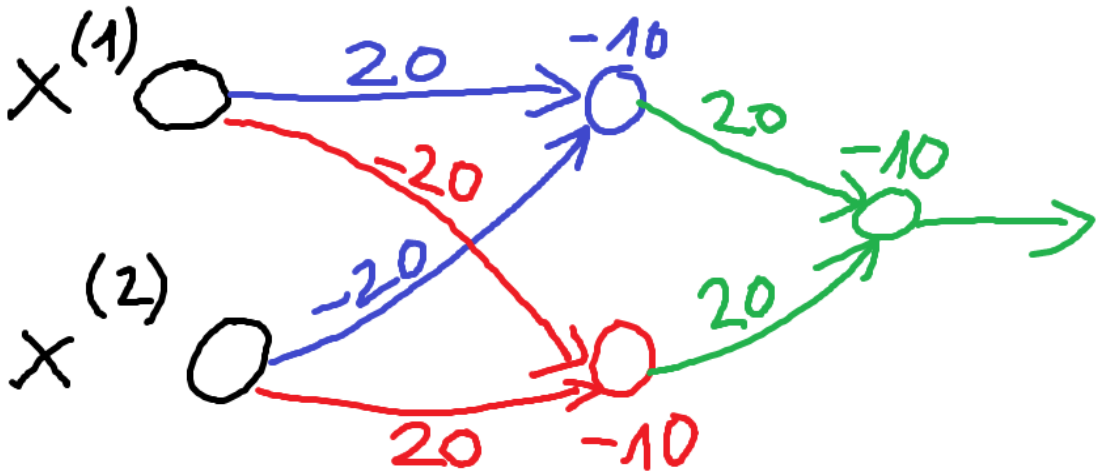
Construction of a Neural Network that Implements a Boolean Function

Krisztian Buza, buza@inf.elte.hu
http://biointelligence.hu

Neural Networks with One Hidden Layer

- Neural networks with one hidden layer with nonlinear activation function are **universal function approximators**.
- Illustration for the case of binary data: XOR

$x^{(1)}$	$x^{(2)}$	$x^{(1)}$ XOR $x^{(2)}$
0	0	0
0	1	1
1	0	1
1	1	0



Krisztian Buza, buza@inf.elte.hu
http://biointelligence.hu

Gradient Descent

Krisztian Buza, buza@inf.elte.hu
http://biointelligence.hu

How do Different Versions of Gradient Descent Find the Optimum?

Blue: gradient descent
Red: stochastic gradient descent