Deep Learning in a Nutshell, Convolutional Neural Networks, Implementation of Machine Learning Algorithms

Krisztian Buza

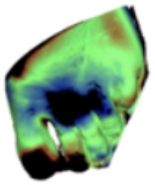
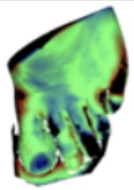
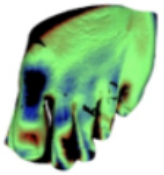
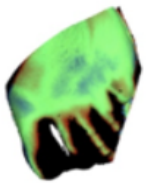
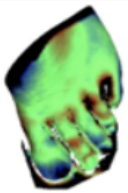
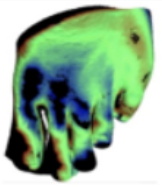
Department of Artificial Intelligence
Eötvös Loránd University
Budapest, Hungary

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

Announcement

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

Possible project / thesis topic: who made which statue?

	Sosikles Type (Capitolino)	Sciarra Type (Copenhagen)	Mattei Type (Tivoli)
Doryphoros (Pompei Statue)			
Diadoumenos (Athens)			
Doryphoros (Pompei Statue)			
Diadoumenos (Athens)			

Pheidias ? Polykleitos ? Krésilas ?

DORYPHOROS

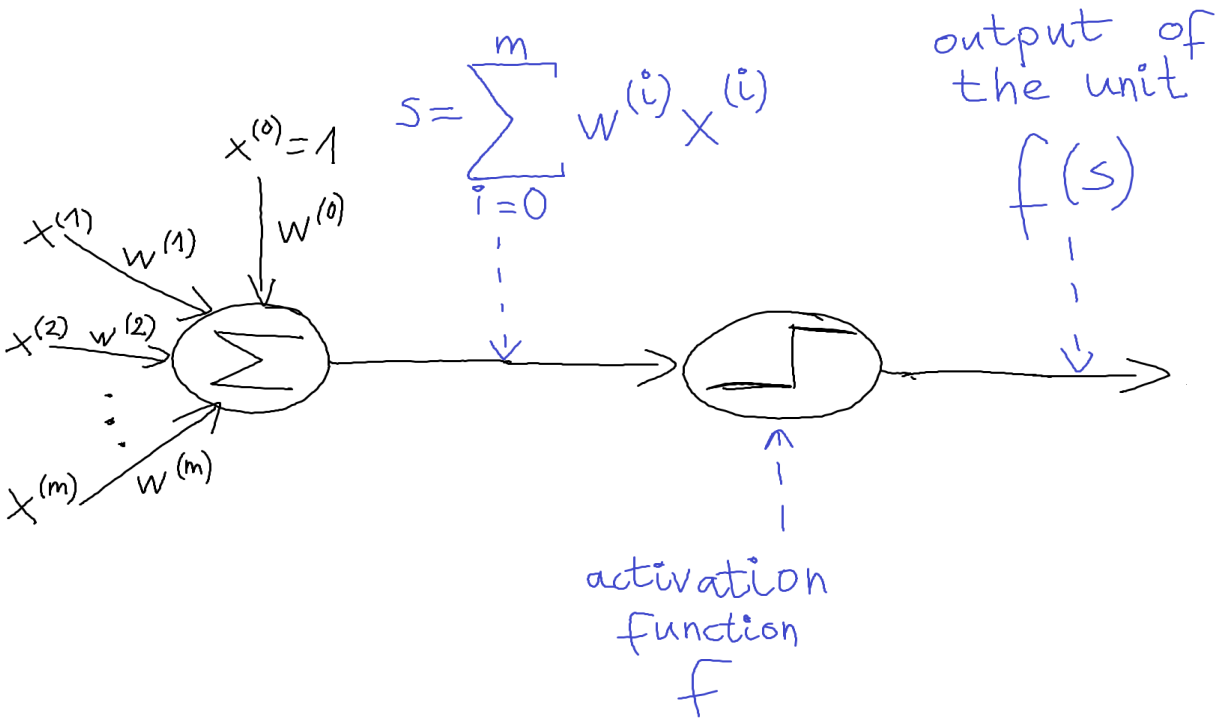
DIADOUMENOS

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

Deep Learning in a Nutshell

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

A Neural Unit



$x^{(1)}, x^{(2)}, \ldots, x^{(m)}$ = inputs of the unit

$w^{(1)}, w^{(2)}, \ldots, w^{(m)}$ = weights of $x^{(1)}, x^{(2)}, \ldots, x^{(m)}$

$w^{(0)}$ = bias weight

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

„What was wrong with backpropagation in 1986?“

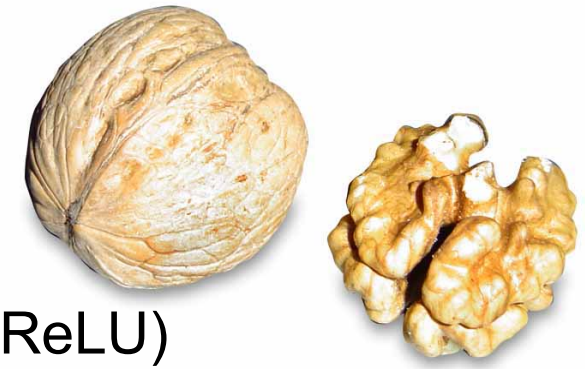
(Geoff Hinton, „Deep Learning“, May 22, 2015)

- Our labeled datasets were thousands of times too small.
- Our computers were millions of times too slow.
- We initialized the weights in a stupid way.
- We used the wrong type of non-linearity.

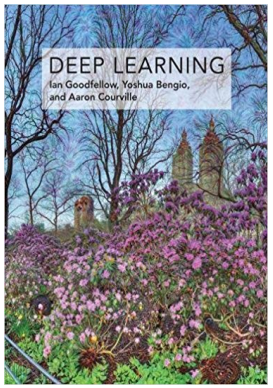
Image: https://upload.wikimedia.org/wikipedia/commons/3/34/Geoffrey_Hinton_at_UBC.jpg
Eviatar Bach [CC BY-SA 3.0 (https://creativecommons.org/licenses/by-sa/3.0)]

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

Deep Learning in a Nutshell



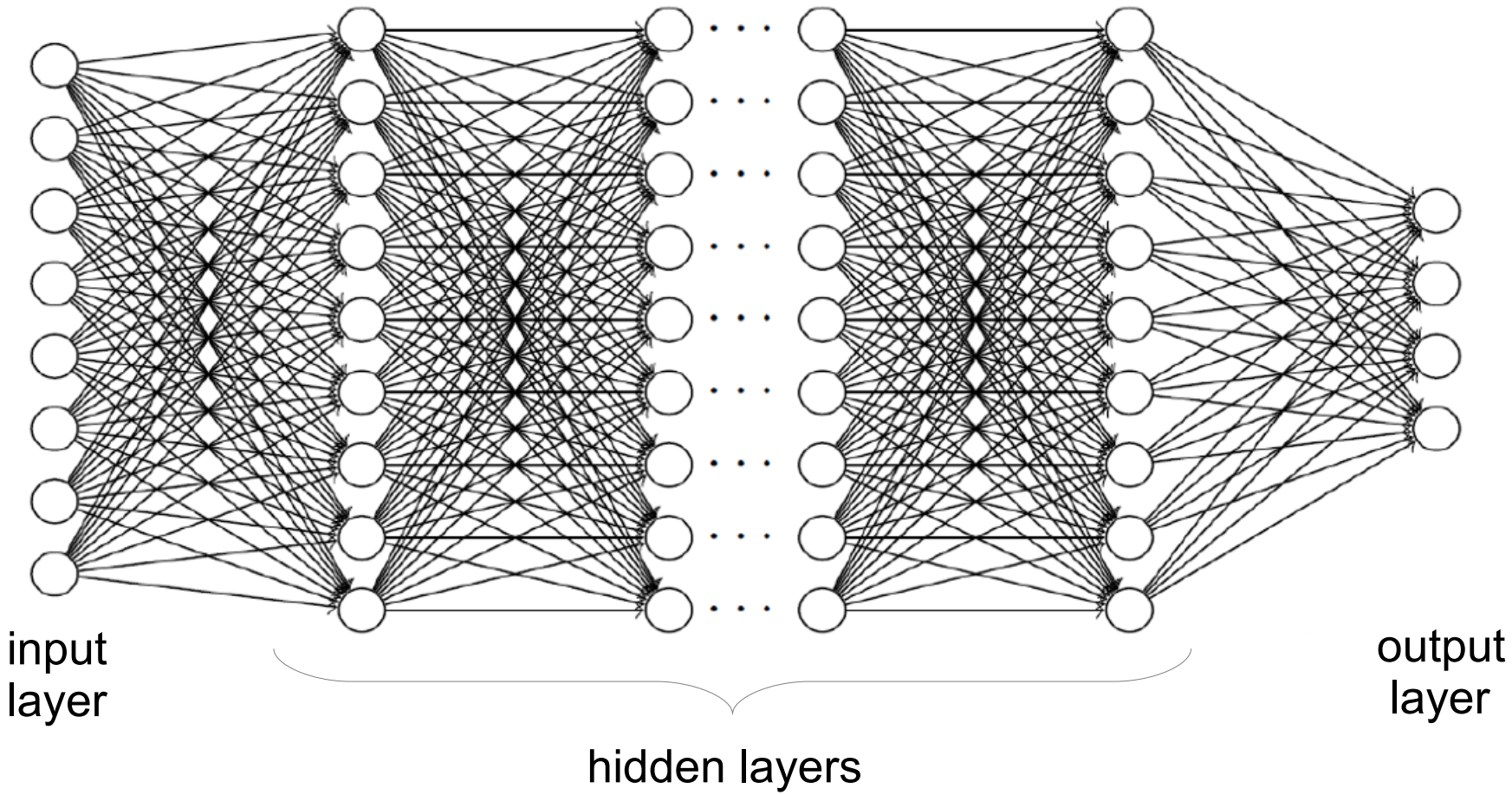
- Size and structure of the network:
  few layers → many layers
- Activation function: sigmoid → rectified linear unit (ReLU)
- Loss function: quadratic loss → cross-entropy
- Initialization of weights: random → (unsupervised) pre-training
- Size of training data, much more memory,
  distributed computation (HPC), GPUs, TPUs…
- New regularization techniques:
  “sparsity-enforcing” regularisation terms,
  drop-out, early stop,
  parameter sharing (convolutional networks)
- New optimization techniques (e.g. ADAM)
- Data augmentation



http://www.deeplearningbook.org

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

Deep Feed-Forward Neural Networks



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

Loss Function: Cross-entropy vs. Mean Squared Error

See **Figure 5** in

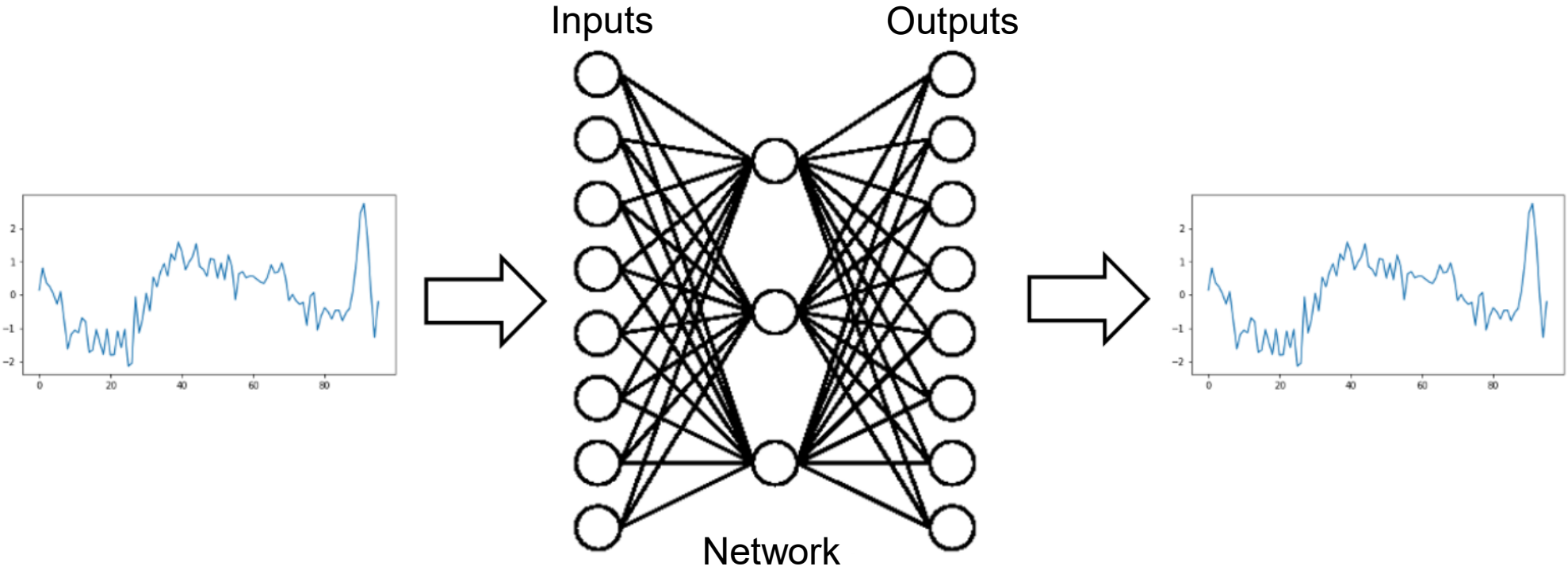
X Glorot, Y Bengio (2010):
Understanding the difficulty of training deep feedforward neural networks

http://proceedings.mlr.press/v9/glorot10a/glorot10a.pdf

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

Initialisation of the Weights

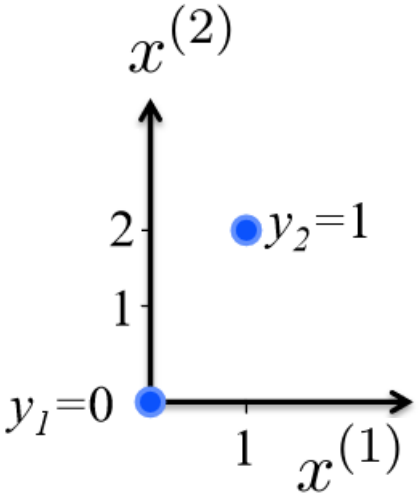
- Unsupervised pre-training: autoencoders
- Supervised pre-training (transfer learning):
  - Train a network for a different (but somehow related...) task
  - Re-use some of the weights
    (e.g. weights of the first few convolutional layers)



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

Sparsity-enforcing (encouraging) Regularisation

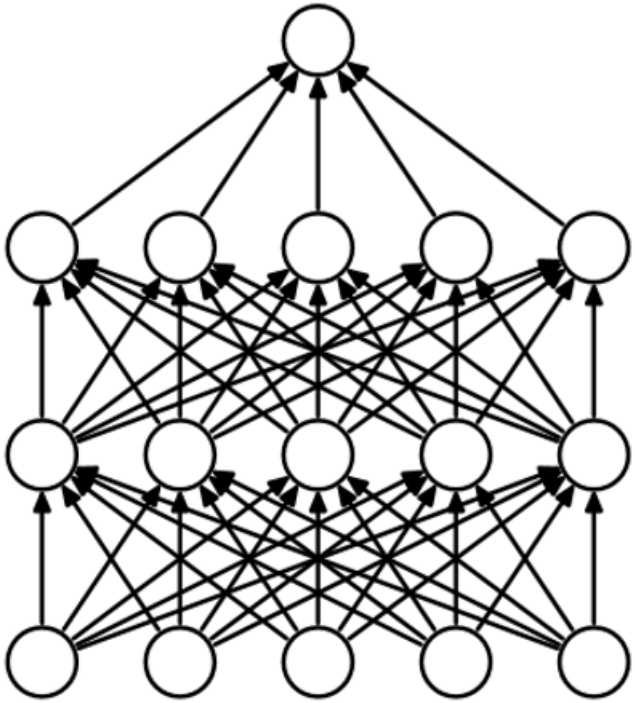
- In the example below, all the three models below have the same prediction performance (on training data)
- L_2 regularisation: $\sum_{i=1}^{p}(w^{(i)})^2$ L_1 regularisation: (sparsity-encouraging) $\sum_{i=1}^{p}|w^{(i)}|$



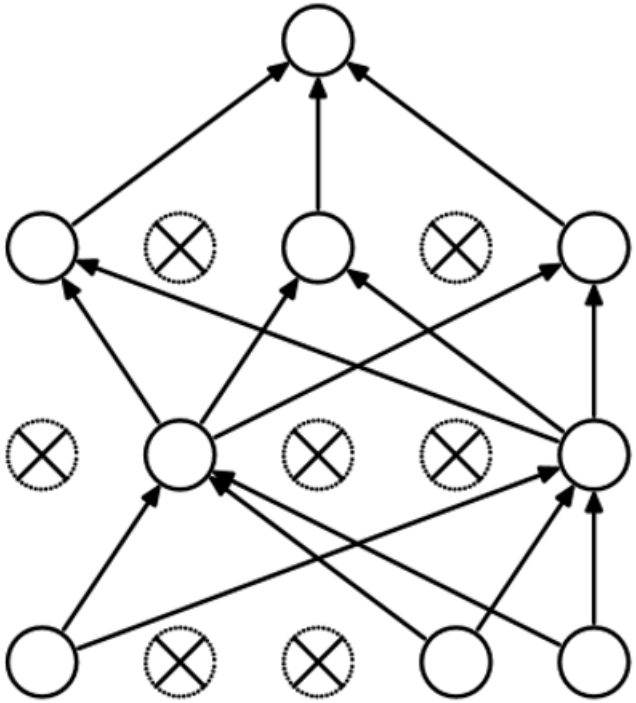
	$\sum_{i=1}^{p}(w^{(i)})^2$	$\sum_{i=1}^{p}\|w^{(i)}\|$
$\hat{y} = 1x^{(1)} + 0x^{(2)}$	$0^2 + 1^2 = 1$	$\|0\| + \|1\| = 1$
$\hat{y} = 0x^{(1)} + \frac{1}{2}x^{(2)}$	$0^2 + (\frac{1}{2})^2 = \frac{1}{4}$	$\|0\| + \|\frac{1}{2}\| = \frac{1}{2}$
$\hat{y} = \frac{1}{3}x^{(1)} + \frac{1}{3}x^{(2)}$	$(\frac{1}{3})^2 + (\frac{1}{3})^2 = \frac{2}{9}$	$\|\frac{1}{3}\| + \|\frac{1}{3}\| = \frac{2}{3}$

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

Dropout



(a) Standard Neural Net



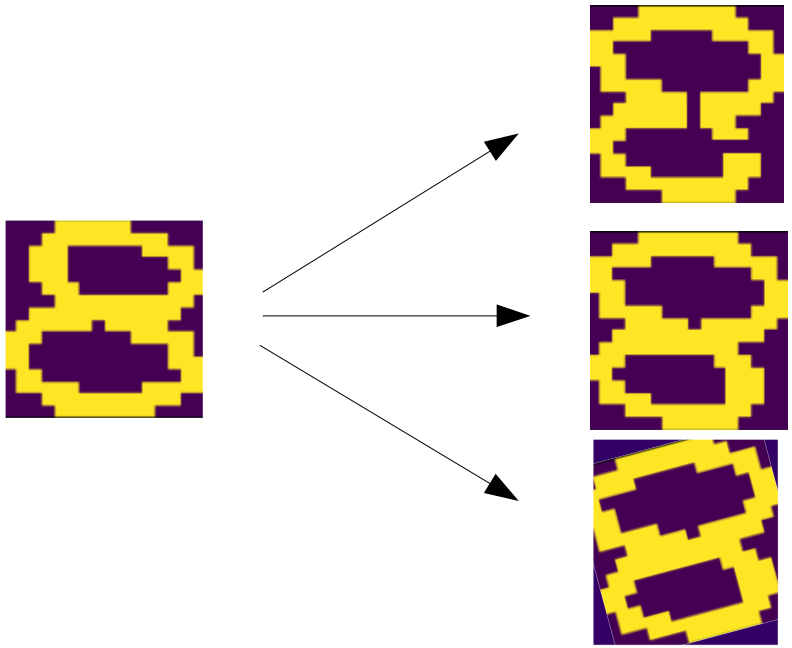
(b) After applying dropout.

Srivastava et al. (2014): Dropout: A Simple Way to Prevent Neural Networks from Overfitting, Journal of Machine Learning Research

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

Data Augmentation

- „Generate“ additional training data
  (usually from existing training instances)
- Images: add noise, rotate, zoom-in/zoom-out...

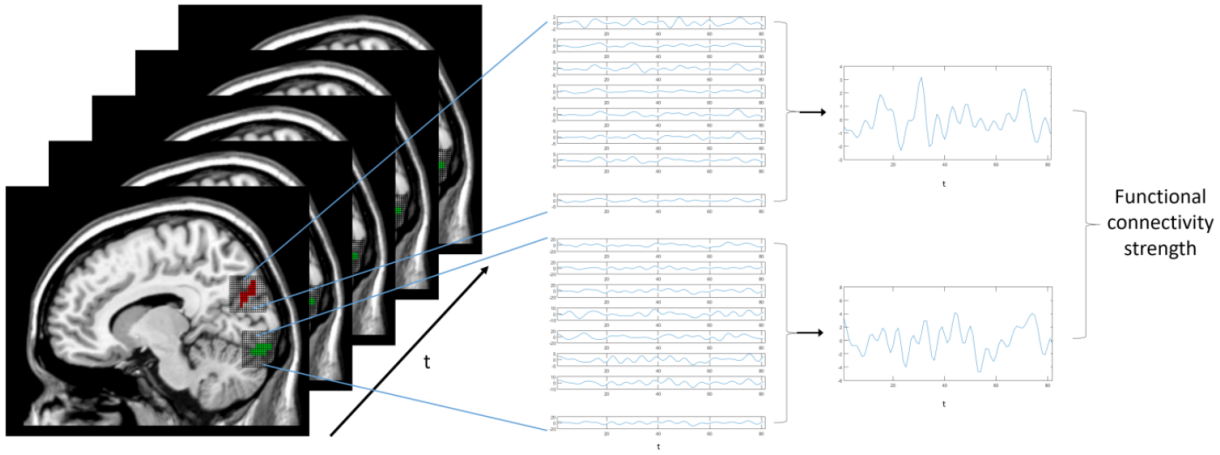


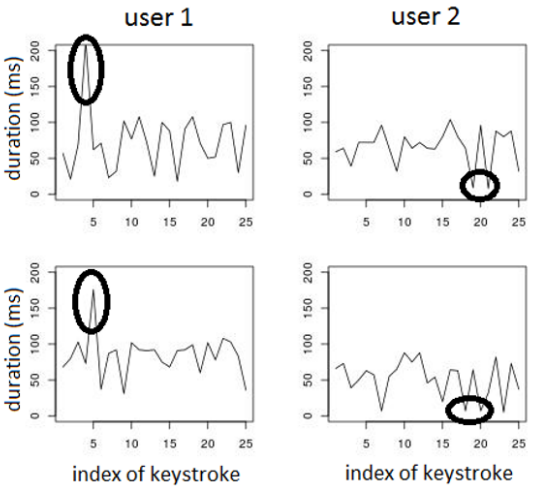
Krisztian Buza, buza@inf.elte.hu
http://biointelligence.hu

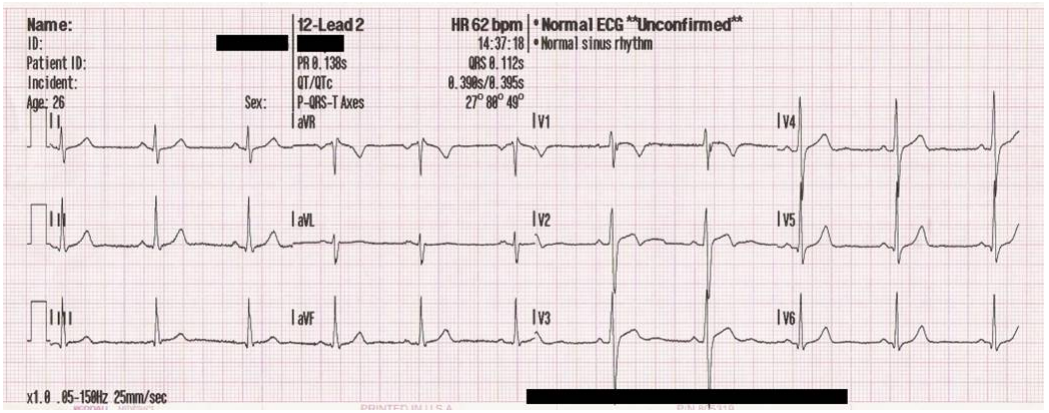
Convolutional Networks

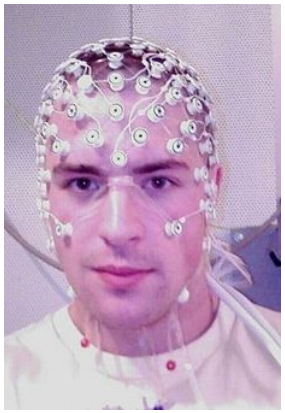
Krisztian Buza, buza@inf.elte.hu
http://biointelligence.hu

Time Series Classification – Examples









Images in the bottom, from left to right:
By MoodyGroove - 2007-01-24 (original upload date) Original uploader was MoodyGroove at en.wikipedia, Public Domain, https://commons.wikimedia.org/w/index.php?curid=5266589
By Thuglas at English Wikipedia - Transferred from en.wikipedia to Commons by Sreejithk2000 using CommonsHelper., Public Domain, https://commons.wikimedia.org/w/index.php?curid=10827060
By JSquish - Own work, CC BY-SA 3.0, https://commons.wikimedia.org/w/index.php?curid=16181727

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)

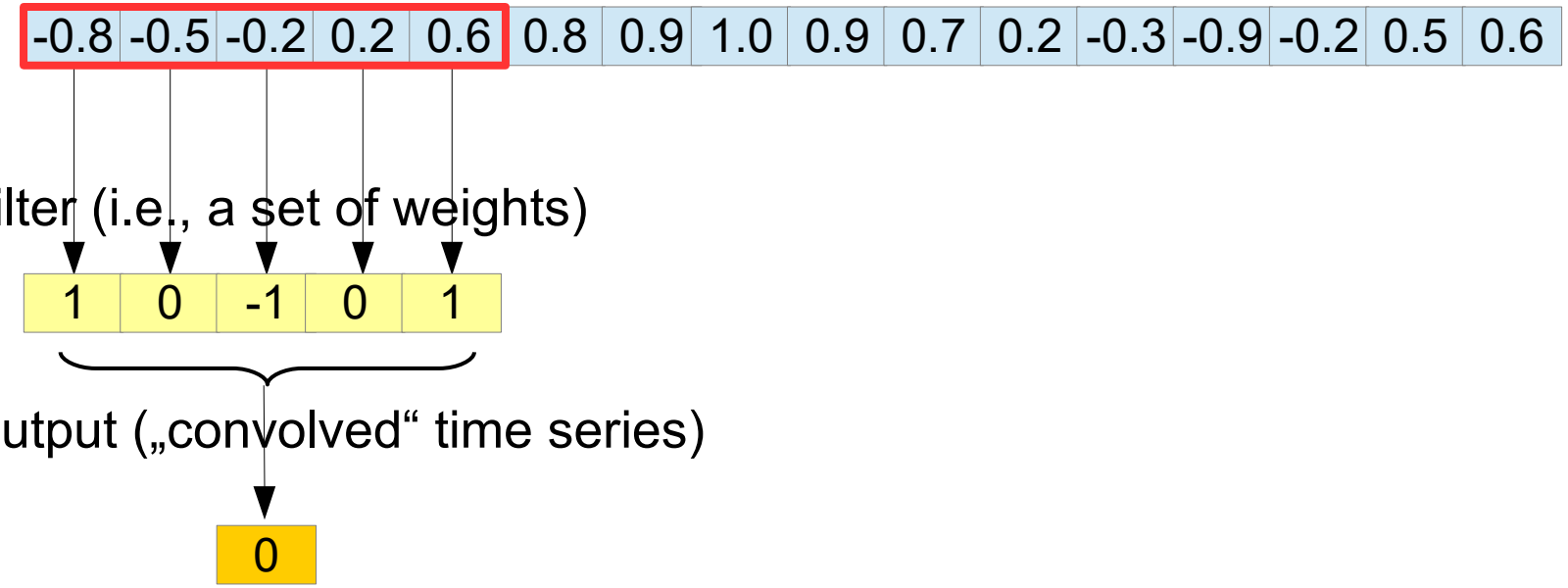
1	0	-1	0	1

* 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):



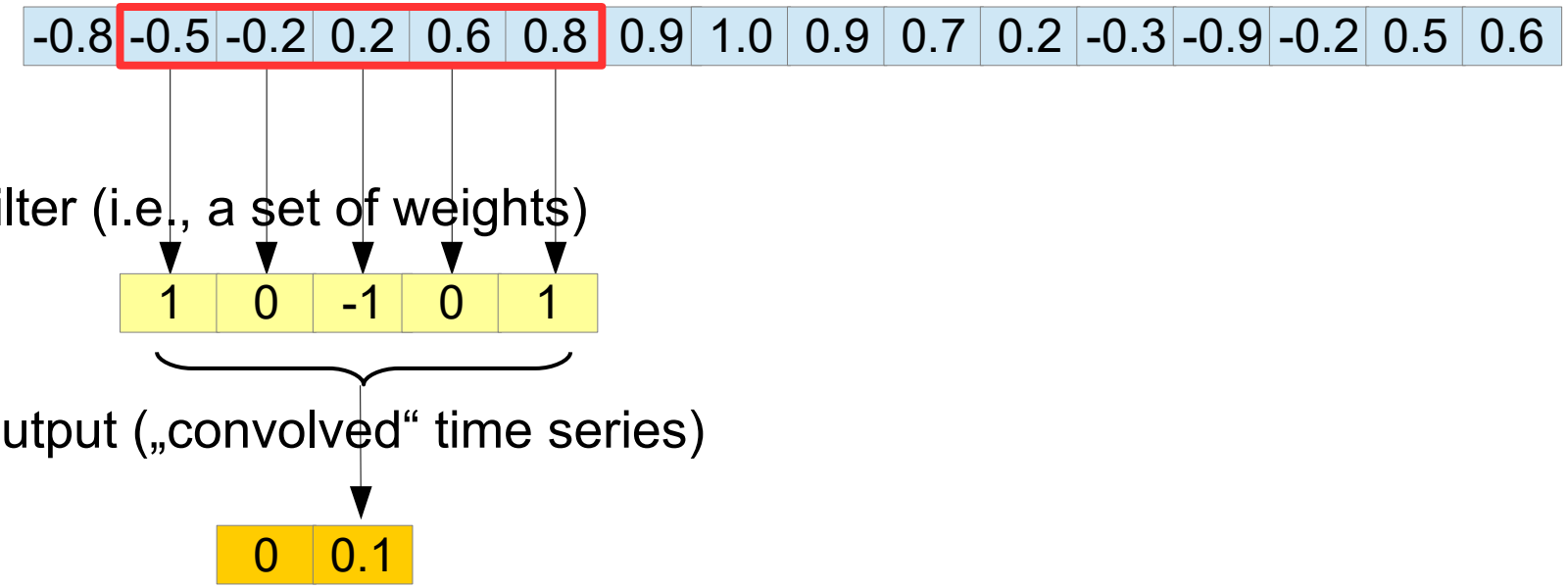
(-0.8) x 1 + (-0.5) x 0 + (-0.2) x (-1) + 0.2 x 0 + 0.6 x 1 = 0

* 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):



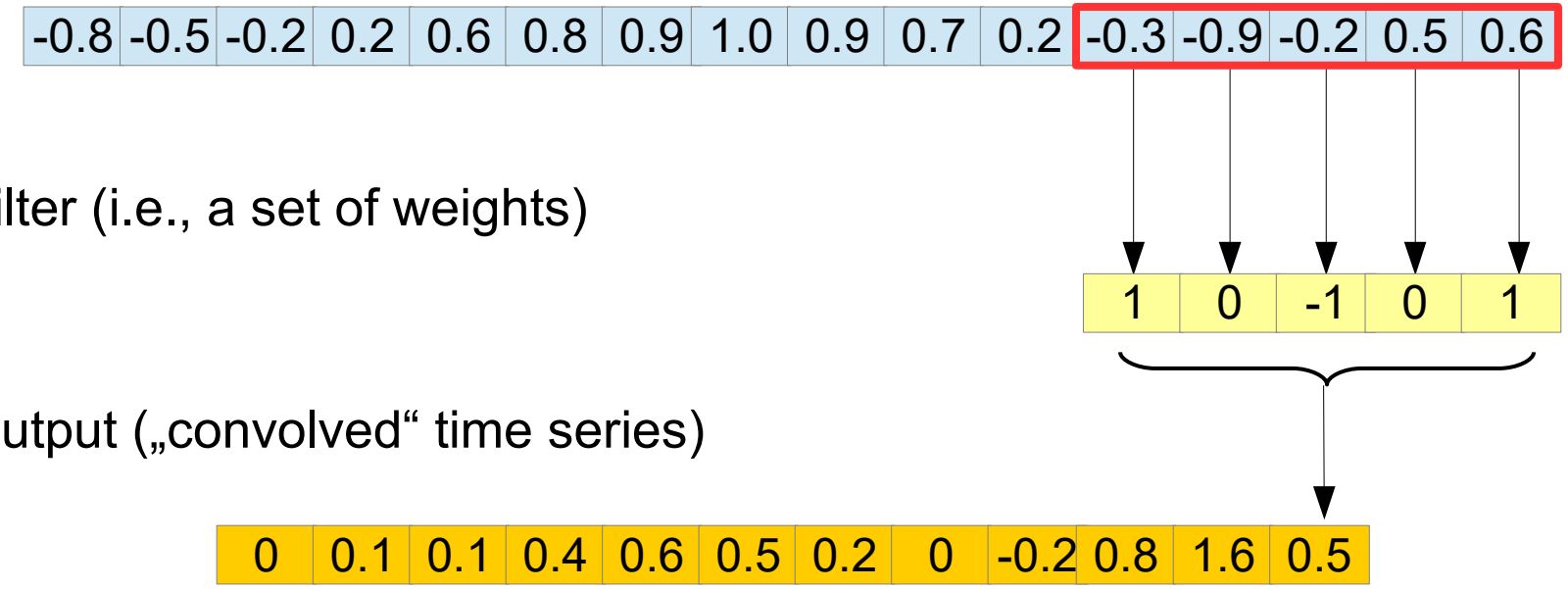
(-0.5) x 1 + (-0.2) x 0 + 0.2 x (-1) + 0.6 x 0 + 0.8 x 1 = -0.1

* 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

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	-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*

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)

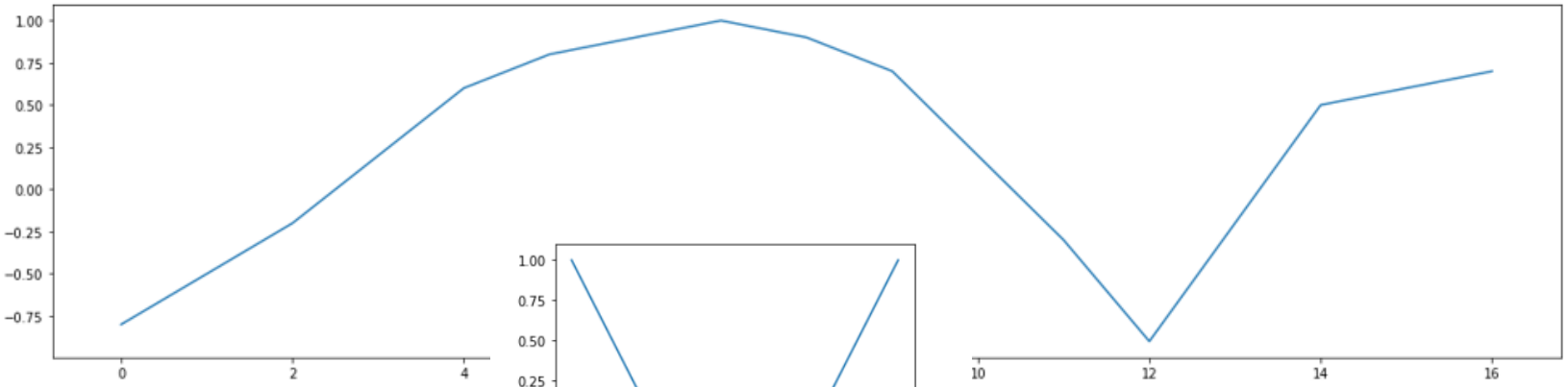
0.6	1.0	0.4	0.1	0.1	0.5	0.9	0.7	0.4	0	-0.4	0.5	1.4	0.7	-1.0	-0.3

* 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

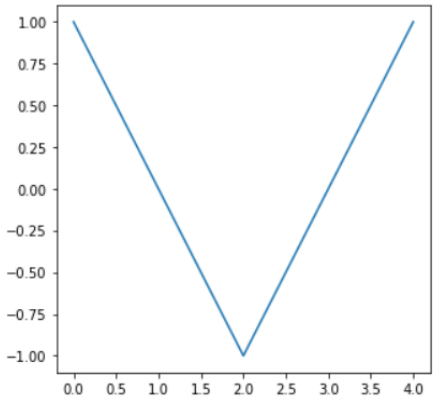


0	0

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

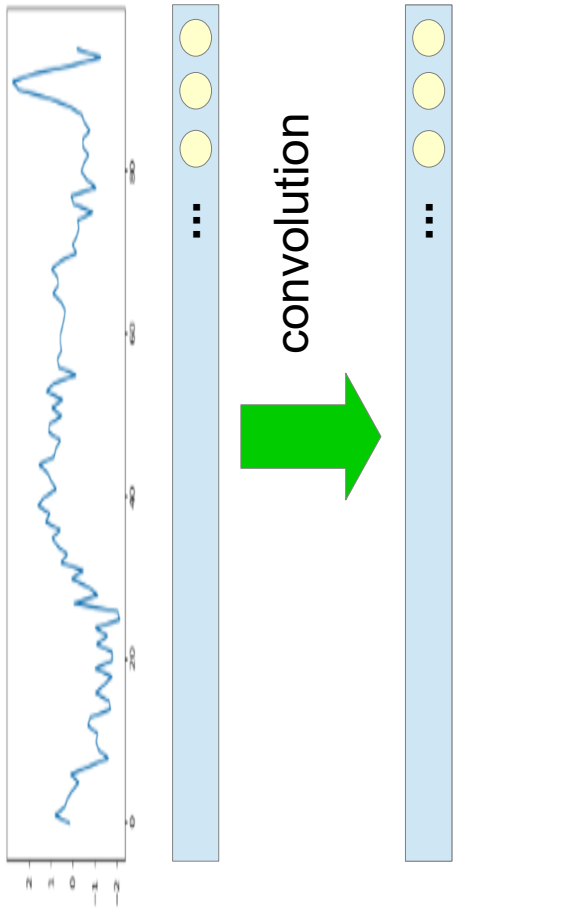
Max pooling

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.

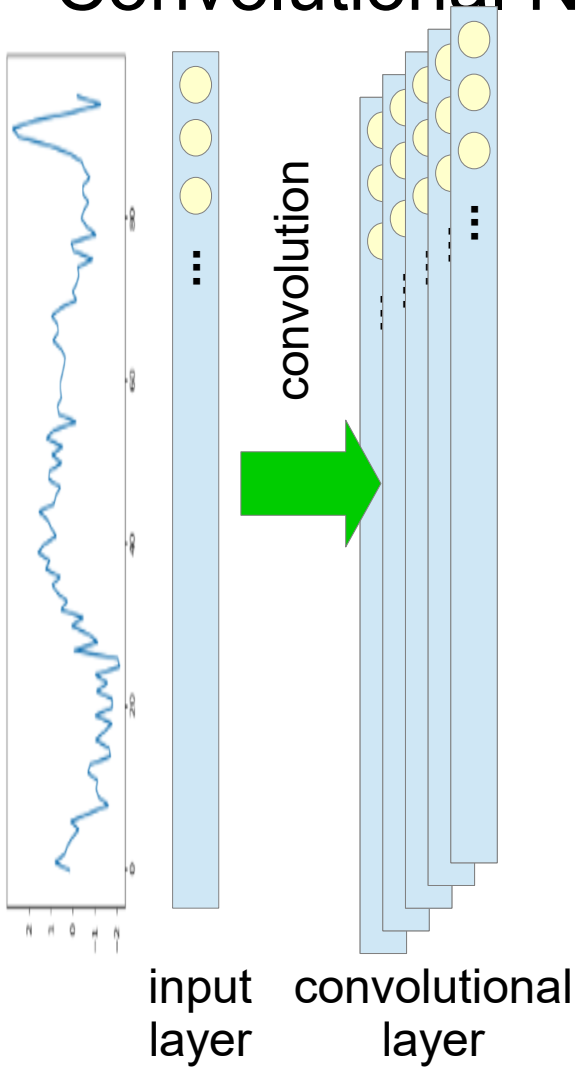
Krisztian Buza, buza@inf.elte.hu
http://biointelligence.hu

Convolutional Neural Networks



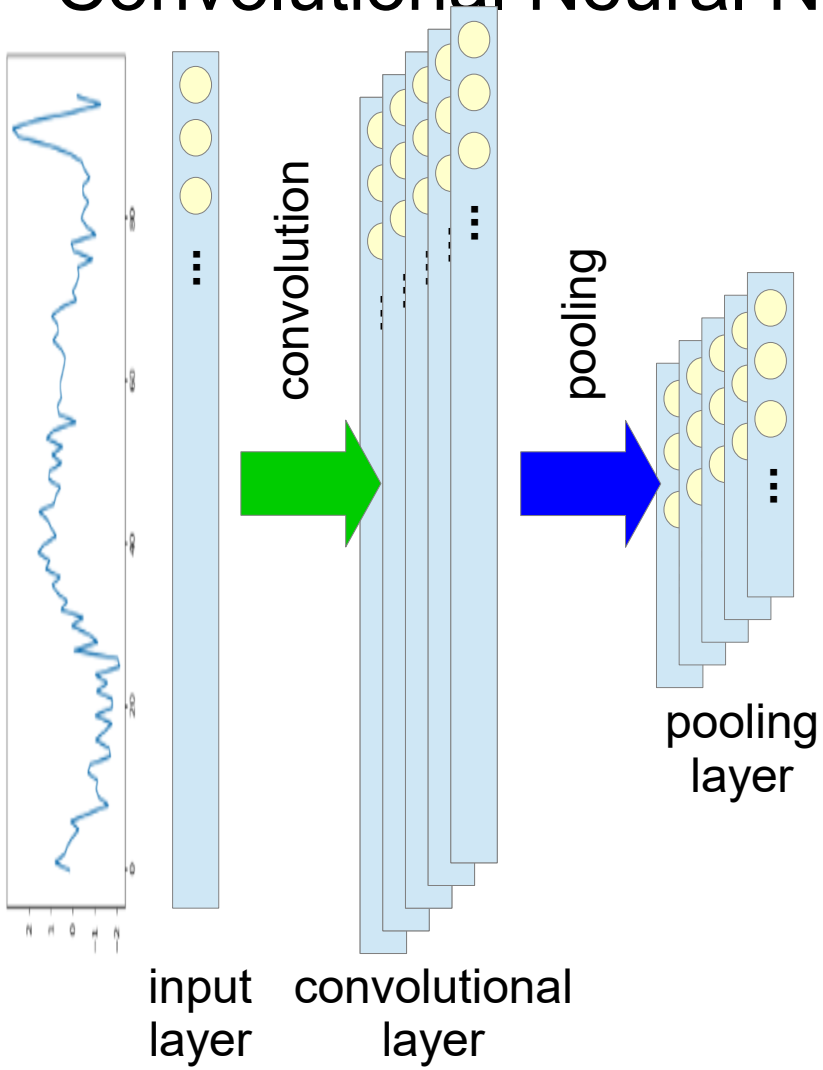
Krisztian Buza, buza@inf.elte.hu
http://biointelligence.hu

Convolutional Neural Networks



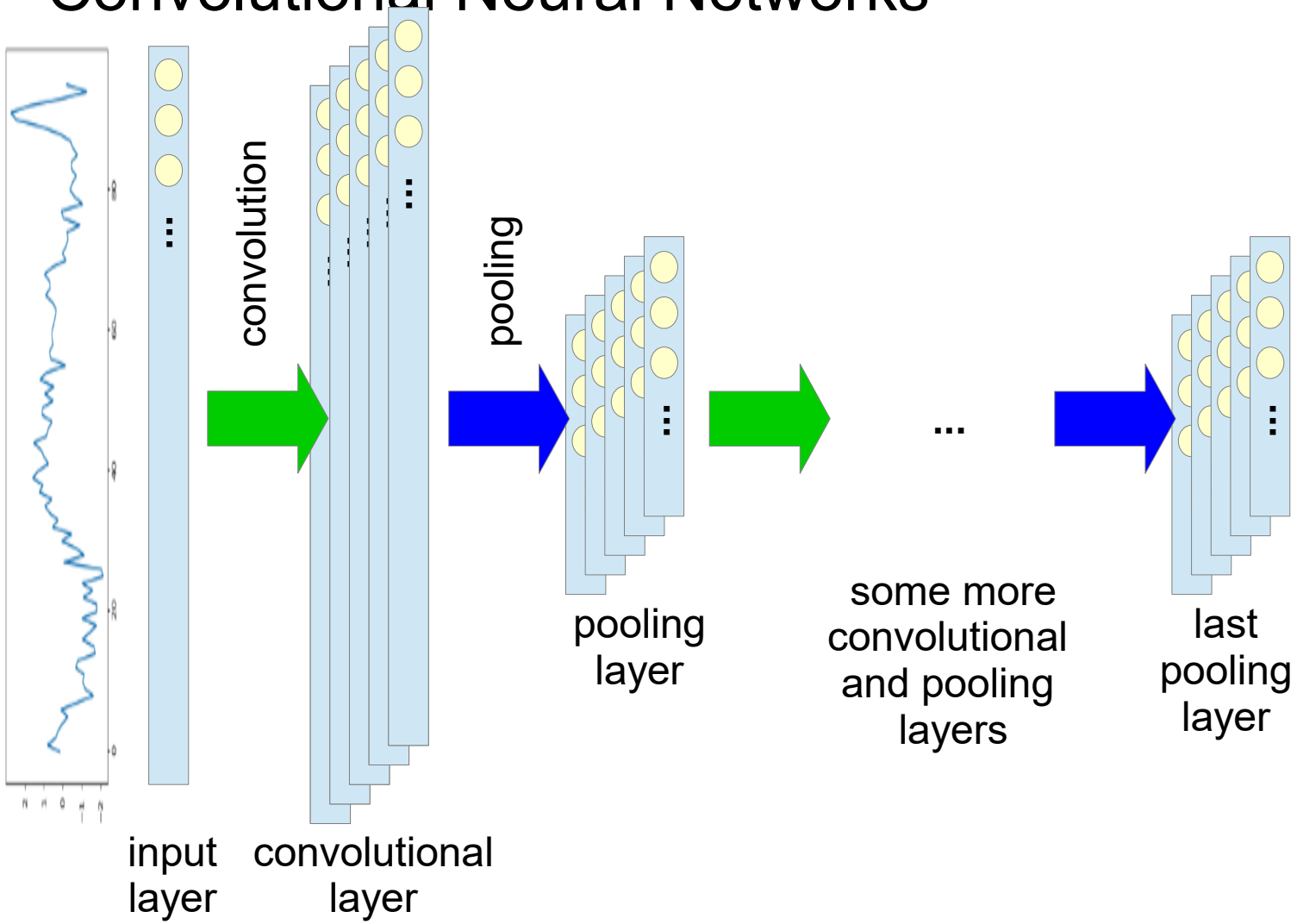
Krisztian Buza, buza@inf.elte.hu
http://biointelligence.hu

Convolutional Neural Networks



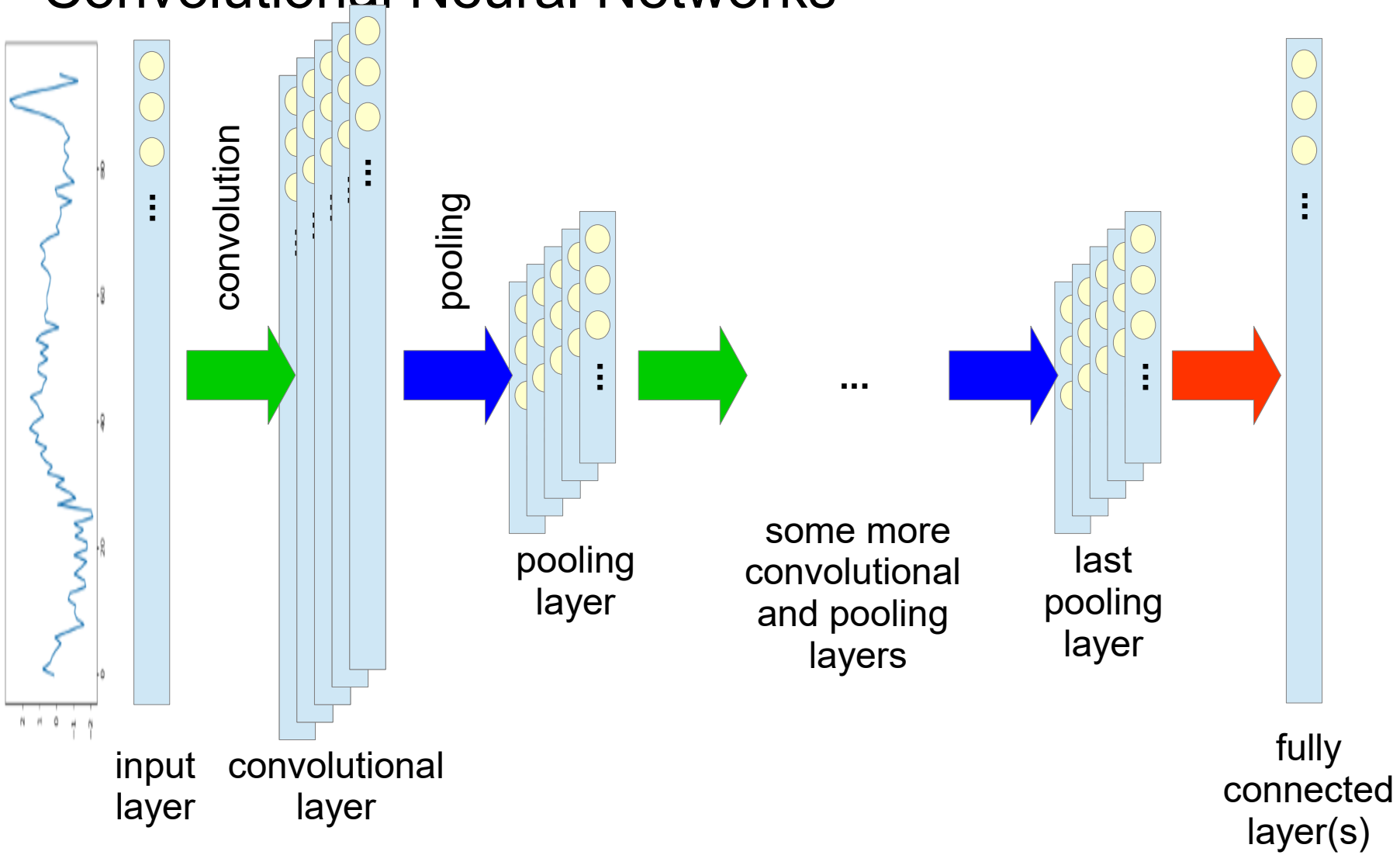
Krisztian Buza, buza@inf.elte.hu
http://biointelligence.hu

Convolutional Neural Networks



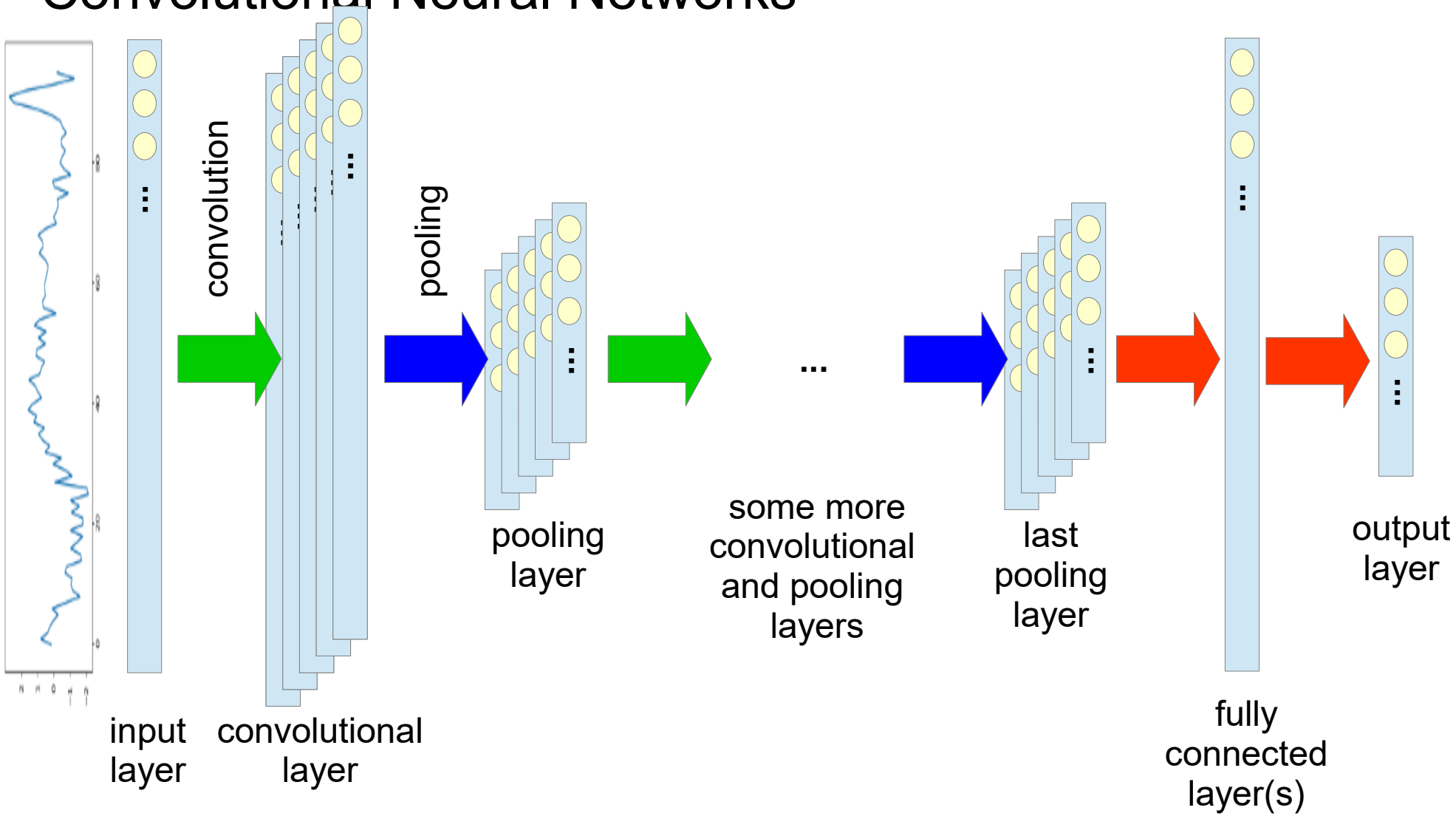
Krisztian Buza, buza@inf.elte.hu
http://biointelligence.hu

Convolutional Neural Networks



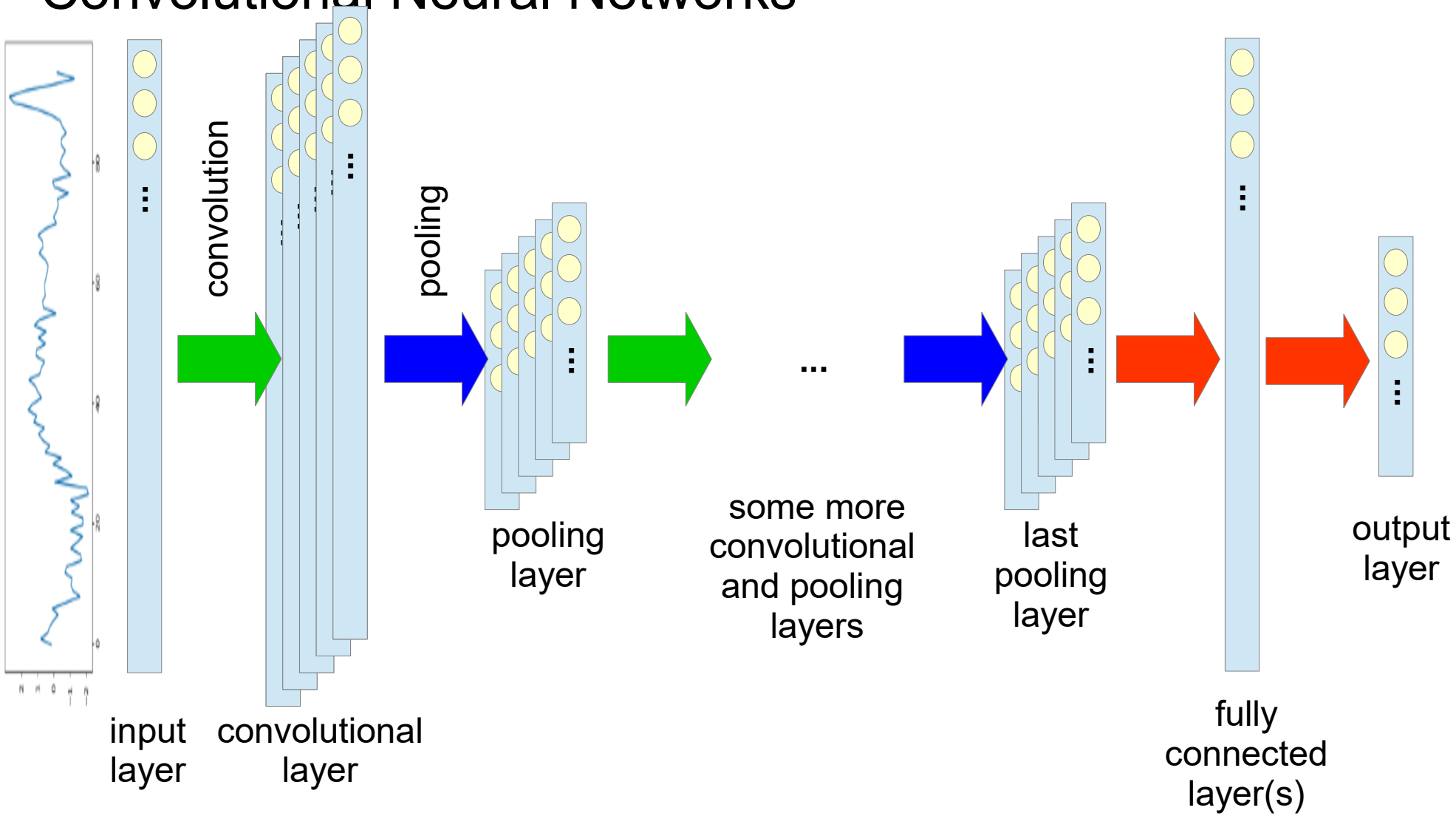
Krisztian Buza, buza@inf.elte.hu
http://biointelligence.hu

Convolutional Neural Networks



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

Convolutional Neural Networks



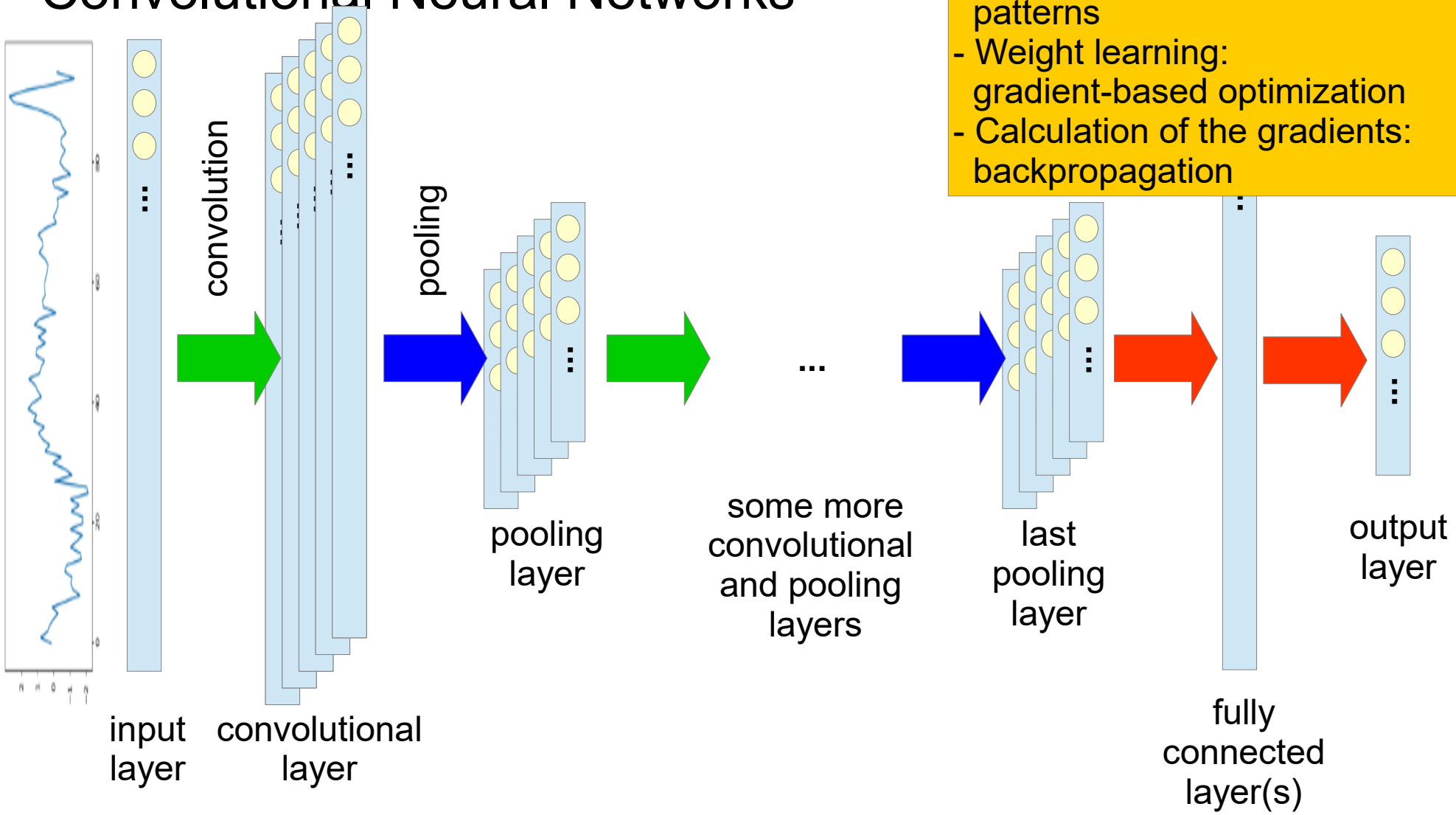
convolution (weight sharing)

pooling (no weights)

every unit is connected with every unit of the next layer

Convolutional Neural Networks

Remarks
- Subsequent convolutional layers detect more and more complex patterns
- Weight learning: gradient-based optimization
- Calculation of the gradients: backpropagation



convolution (weight sharing)

pooling (no weights)

every unit is connected with every unit of the next layer

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

Classification based on Local Patterns

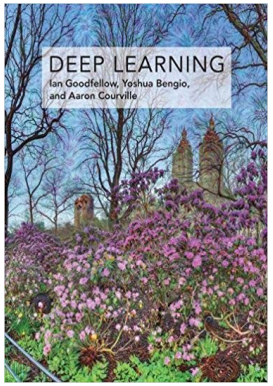
- Motif-based classification

  Buza, Schmidt-Thieme (2009): Motif-based classification of time series with Bayesian networks and SVMs, Advances in Data Analysis, Data Handling and Business Intelligence. Springer, Berlin, Heidelberg, pp. 105-114

- Shapelet-based classification

  Hills et al. (2014): Classification of time series by shapelet transformation, Data Mining and Knowledge Discovery, 28(4), pp. 851-881

- Convolutional Networks



Ian Goodfellow, Yoshua Bengio, Aaron Courville (2016): Deep Learning, http://www.deeplearningbook.org

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

Implementation of Machine Learning Algorithms

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

Options

- 1. Implement everything from scratch
- 2. Implement training and prediction algorithm using basic functionality of software libraries such as dot product, matrix multiplications, etc.
- 3. Implement the model using more advanced functionality of software libraries, such as calculation of the gradients, optimization algorithms, built-in functions to train and evaluate the model
- 4. Integrate an existing model into your system

Some Implementation issues

1. **Selection of the programming language, software libraries, development tools (IDE) and methods**

   Copyright, licence, unit tests, code review, coding and documentation standards, repository...

2. **Availability of labeled data, data preprocessing**

   This may be a technological and legal issue simultaneously.

3. **Optimization of computational costs, usage of RAM, energy consumption, availability of suitable hardware**

   Selection of data types, GPU, TPU, HPC, etc.

4. **Other issues**

   Safety („cover your ass“),
   compliance with law and ethical standards, etc.

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

Data Preprocessing (Data Cleaning)

- Handle Missing Values
- Elimination Duplicates
- Outlier Detection
- Handle Inconsistent Values

  (e.g. replace misspelled values)

TensorFlow

- State-of-the-art machine learning library
- Suited for deep learning
- https://www.tensorflow.org/

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

Programming Homeworks

Programming Homeworks

- 1-3 students are expected to work in a team
  (can be the same team as for the presentations, but it can be different as well)
- Three topics: Data Preprocessing (DP), Regression (R), Classification with Logistic Regression (C)
  each team is expected to solve ALL tasks from ALL these topics
- Deadline: 4st Nov, 24 pts, submission via e-mail to buza@inf.elte.hu
- You should deliver: code + contribution of team members like this:

Zoltan Horvath	25%	DP-1, DP-2, R-1...
Krisztian Buza	30%	R-1, R-2, C-2...
Zakarya Farou	45%	R-1, R-3, R-4, R-5, C-1...

- In the lecture on the 6th Nov, for each task T, a randomly chosen team will be selected. If the selected team delivered a solution for T, but can not explain their solution, team members' scores (for the entire programming homework) will be set to 0 (zero).

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

Programming Homeworks

- You will need
  - … a Python interpreter / IDE, such as
    - a command line interpreter
    - Anaconda Jupyter Notebook
    - Google Collab
  - … to install some libraries (numpy, scikit-learn...)
- Tasks are provided in two formats: html and ipynb
  - Feel free to use which is more convinient for you
- If you use the ipynb files, you may need to rename the **ipynb** files from *.html to *.ipynb (if your browser saves them as html files)
- **Please google for installation/usage instructions**

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

Programming Homework

- Most of the code is already prepared, you only need to complete some of the functions and execute the completed code.
- **Be honest**: the solution of similar tasks are available on the internet. Studying these solutions is allowed, because the goal of the homework is not that you suffer, but that you learn.
- **However**: you should **understand** the solutions that is submitted, and you should not ask anyone else (who is not a member of your team) to solve the tasks instead of you.

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

Summary

Summary

- Overview of the techniques known under the umbrella of „deep learning“
- New mathematical operation: Convolution
- Convolutional Networks
- Issues related to the implementation of machine learning algorithms

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

Essential Concepts

- GPU, TPU („AI chip“)
- Pre-training
  - Supervised pre-training
  - Unsupervised pre-training
- Drop-out
- Convolution
- Convolutional networks
- Data Augmentation
- Data Preprocessing, Data Cleaning