LECTURER: TAI LE QUY

MACHINE LEARNING

UNSUPERVISED LEARNING AND FEATURE ENGINEERING

UNIT 6

AUTOMATED FEATURE GENERATION

STUDY GOALS

- Explain how to automatically generate **transformation features**.
- Understand how to automatically generate **aggregation features**.
- Analyze the **advantages and limitations** of the techniques used to automatically generate features.

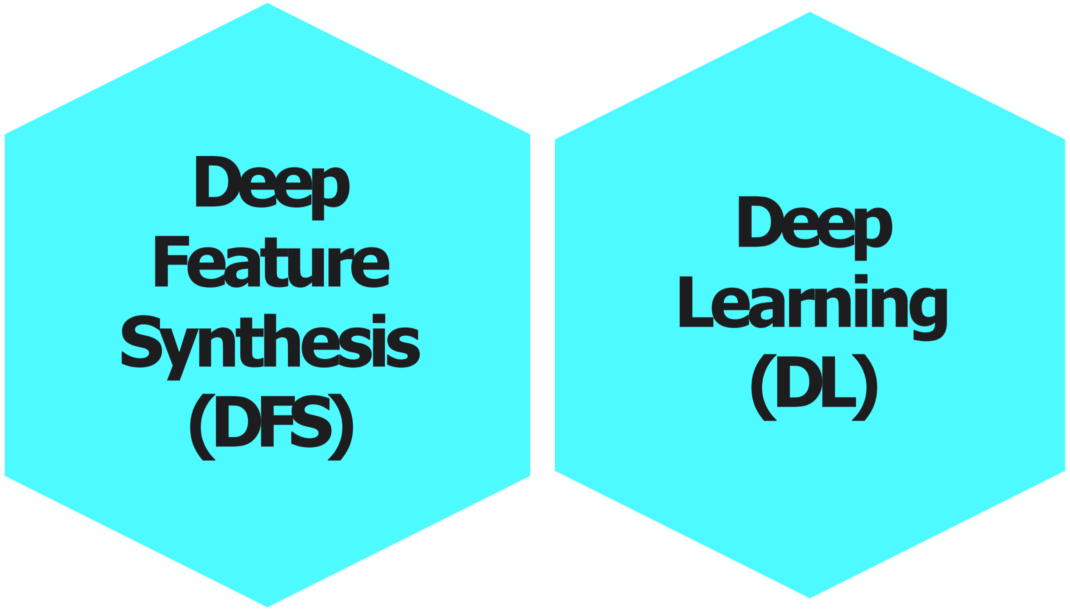
1. Explain the difference between **transformations** and **aggregations**.

2. Explain what is meant by the term "**complex features**".

3. Explain what **feature retrieval** is.

UNIT CONTENT

Image 1: Unit content - Automated Feature Generation



Source of the image: Müller-Kett, 2021.

- **Featuretools**:
  - Open source Python library framework for automatic generation of features
  - Transforms transactional and **relational datasets** into adapted feature matrices for machine learning
  - Works on a concept known as Deep Feature Synthesis (DFS)
  - DFS allows us to automatically create multiple features either as **transformations** or **aggregations**
  - **Transformations**: are done to one or more columns on a single table
  - **Aggregations**: using different primitives applied to several tables

https://www.featuretools.com/

TRANSFORMATION PRIMITIVES

Transformation Primitives in the Featuretools Library	
multiply_boolean	Element-wise multiplication of two lists of Boolean values
year	Determines the year value of a datetime
day	Determines the day of the month based on a datetime
weekday()	Returns the day of the week from a datetime value. Weeks start on Monday (day 0) and run through Sunday (day 6).
divide_by_feature	Divides a scalar by each value in the list
equal	Determines if values in one list are equal to another list

AGGREGATION PRIMITIVES

Aggregation Primitives in the featuretools Library	
all	Calculates if all values are 'True' in a list
std	Computes the standard deviation which is the dispersion relative to the mean value, ignoring `NaN`,
num_unique	Determines the number of distinct values, ignoring `NaN` values
n_most_common	Determines the `n` most common elements
mean	Computes the average for a list of values
num_true	Counts the number of `True` values
median	Determines the middlemost value in a list of values

- Tabular datasets into **derived feature matrices**
- **Transformations**
- **Aggregations**
- Example: Python's **featuretools**

Table 1: Feature matrix

t	sales	var(h=3)	max(h=7)	range(h=30)
1	234	1	234	30
2	321	2	321	32
3	323	3	323	24

Source of the text: Kanter & Veeramachaneni, 2015.
Source of the table: Müller-Kett, 2023.

Three major components:

Entities

- DataFrame Tables
- Must have a unique index identifying each row

Entitysets

- Multiple relational tables
- Hierarchical: Each relationship links an Entity parent to an Entity child

Primitives

- Aggregation operations (basic operations that are used to form new features across one entity or several entities.)
- E.g., applying the transformation primitive "AGE" to the feature or column Date of birth

Table 1: Feature matrix

t	sales	var(h=3)	max(h=7)	range(h=30)
1	234	1	234	30
2	321	2	321	32
3	323	3	323	24

Source of the table: Müller-Kett, 2023.

Primitive levels

- 1st depth: Mean
- 2nd depth: Max of means
- Complex features (depth >1)

Deep Feature Synthesis

- Automated multi-depth aggregations
- Based on defined entity relationships

Table 1: Feature matrix

t	sales	var(h=3)	max(h=7)	range(h=30)
1	234	1	234	30
2	321	2	321	32
3	323	3	323	24

Source of the table: Müller-Kett, 2023.

EXAMPLE

– Each customer orders a certain number of products and each product has a certain price.

Customer Table		
Customer_ID	Customer_name	Creation-date
C1	Martin	2018-08-15
C2	Julia	2020-05-05

Customer Orders	
Order ID	Customer ID
1	C1
2	C2
3	C1
4	C1
5	C2

Customer Payments	
Order_ID	Price
1	500
5	200
3	300
4	100
2	900

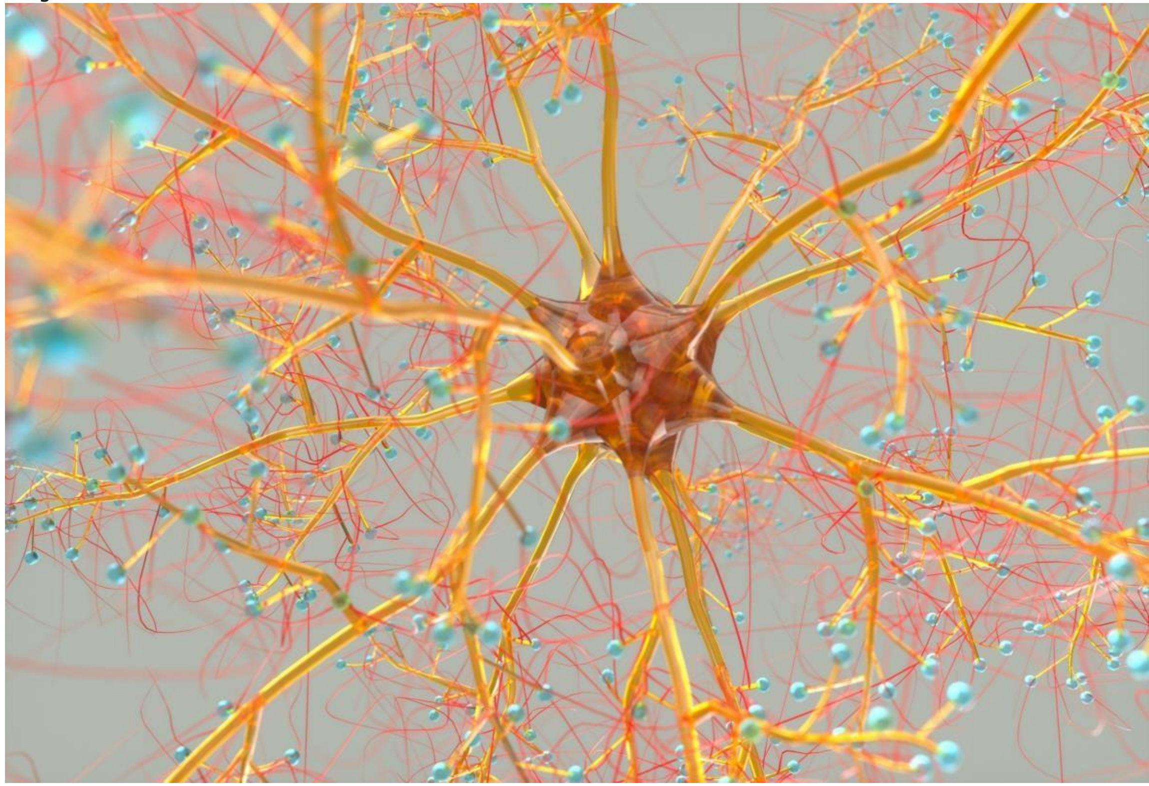
DFS is a concept allowing us to automatically generate new features from single and multiple entities (DataFrame).

FEATURE ENGINEERING VERSUS DEEP LEARNING

- Deep learning approaches, such as Convolutional Neural Network (CNN) are commonly used for classification tasks for images, text, and audio
- Kernel filters and pools, i.e., aggregations, are applied to the original input data → can also be used as input feature to other machine learning models → constituting a technique for automated feature generation
- These features are automatically extracted from raw data by matrix multiplication

- **Convolutional Neural Networks (CNN)**
- **Generate distinctive features** from input images
- **"Hidden" layers** in the network architecture
- **Feature retrieval**
  - **Extracting** this information to be used by **any machine learning algorithm**

Image 2: Neuron

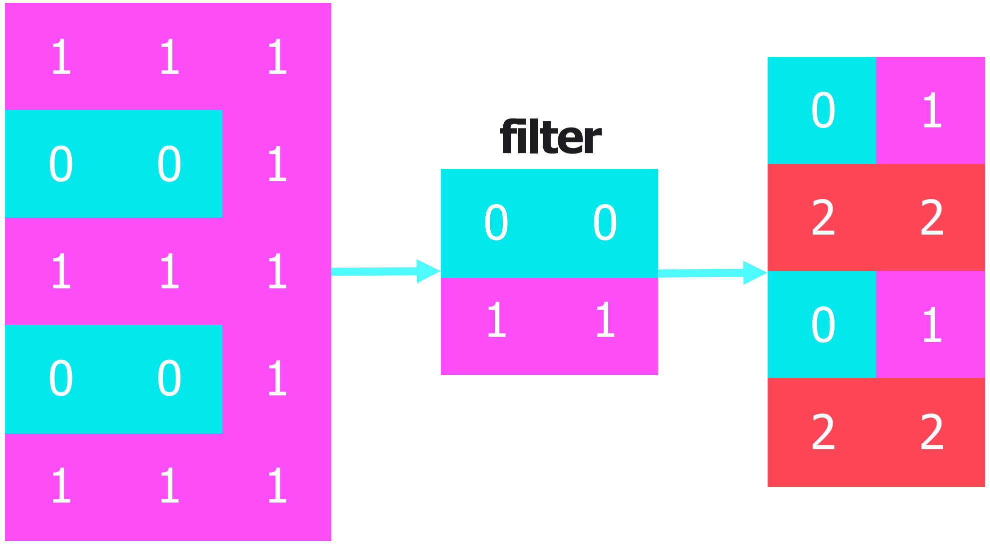


Source of the image: Microsoft Archive.

Filters

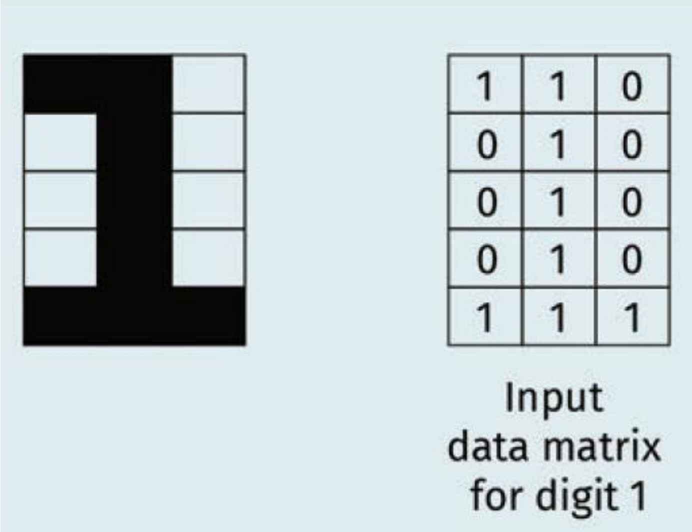
- **Kernel** functions
- Applied to **each image pixel**
- Considering **neighboring pixels**
- **Example**: Detecting vertical/horizontal lines

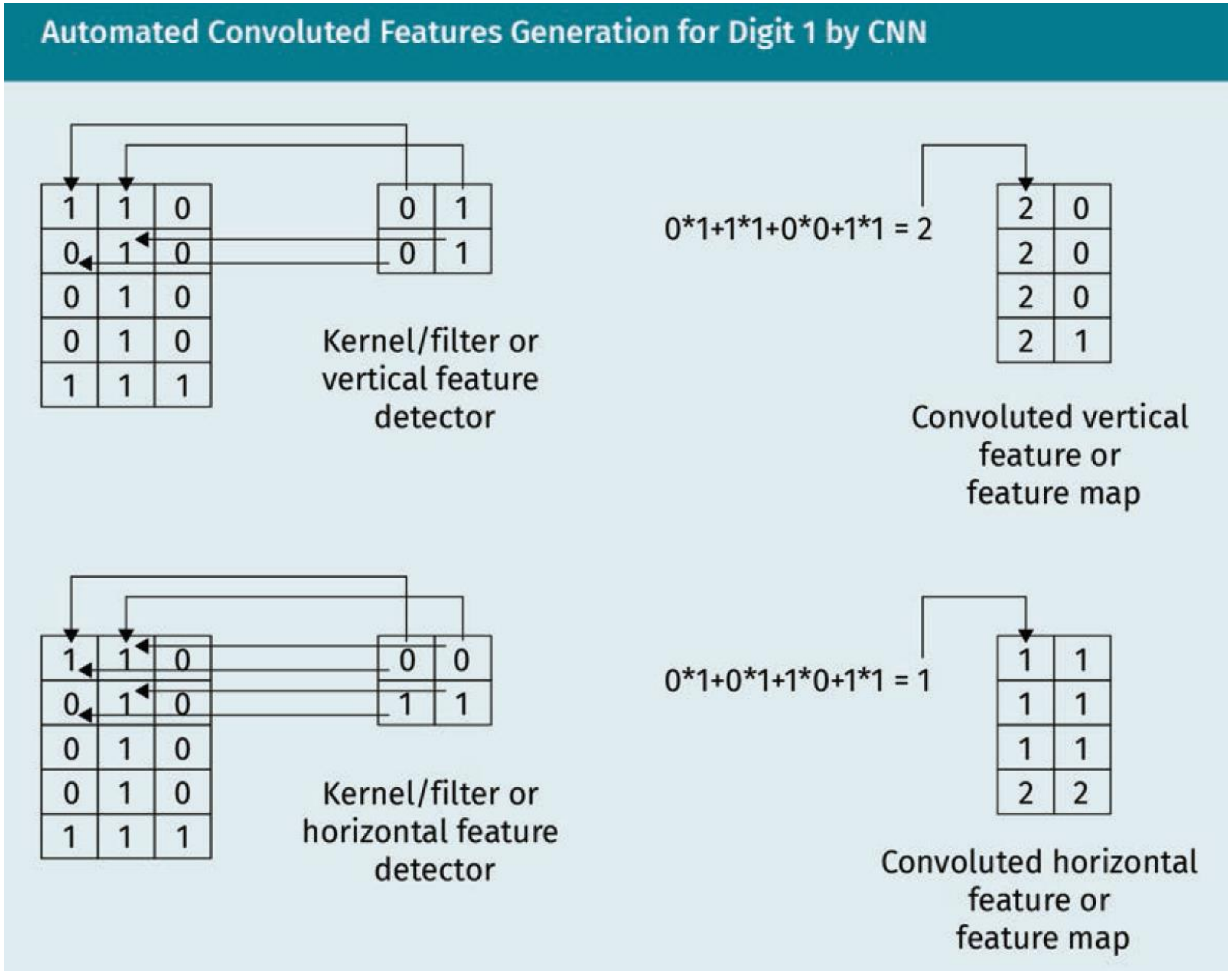
Image 3: Filter



Source of the table: Müller-Kett, 2023.

CONVOLUTIONAL NEURAL NETWORKS (CNN) – FILTER/KERNEL

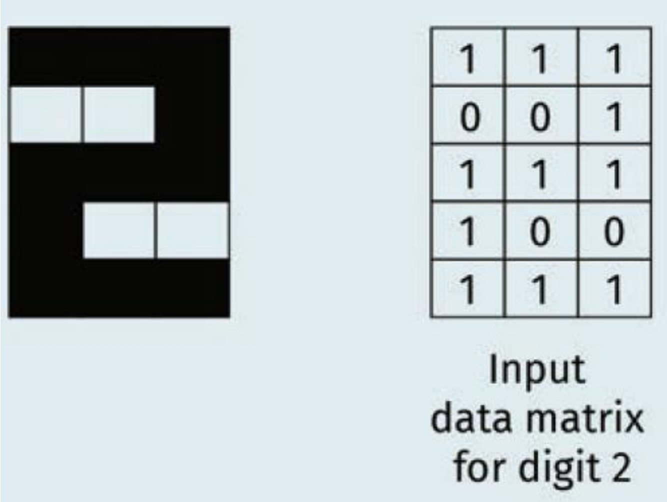


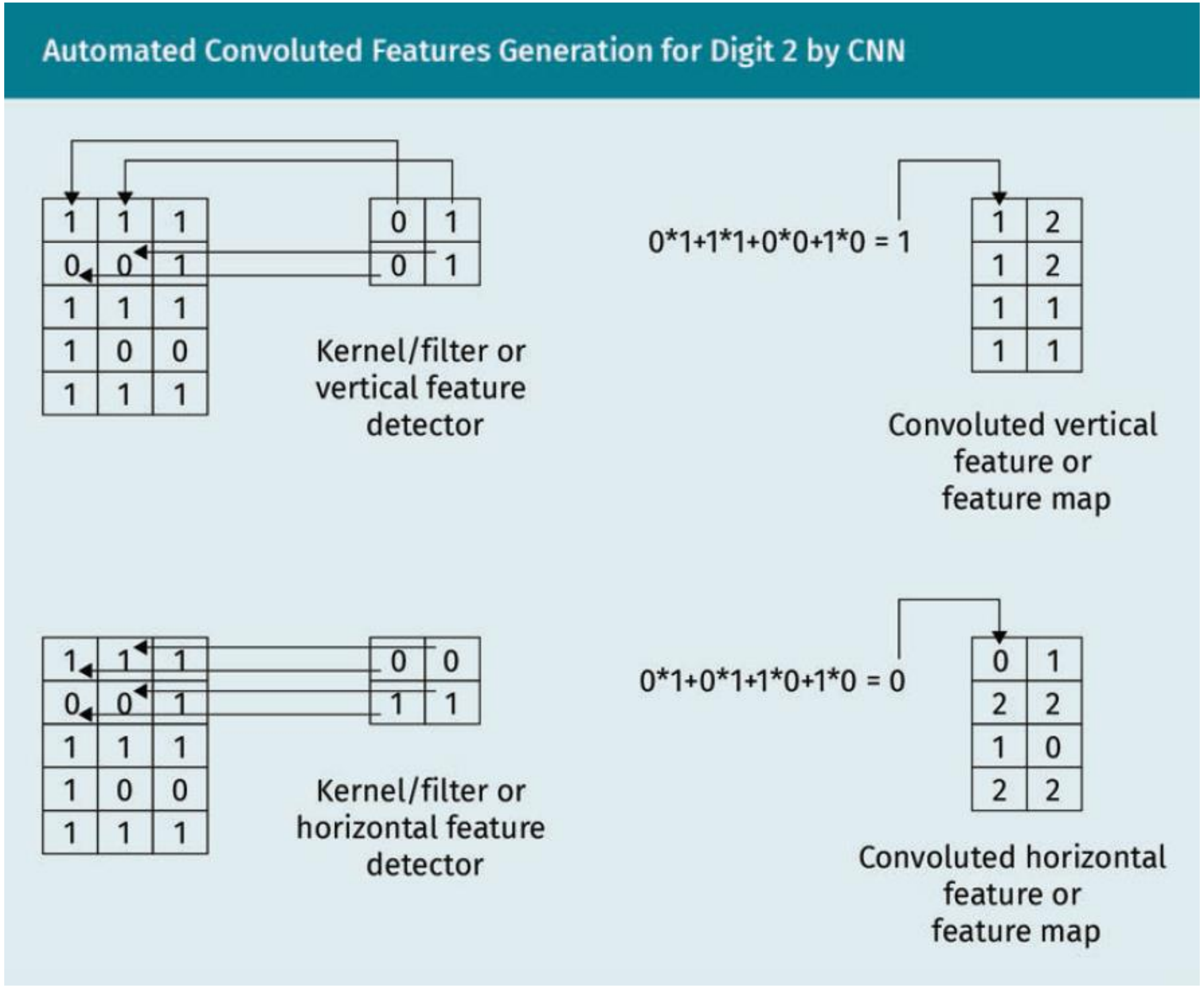


Vertical and horizontal filters are used by CNN in order to build the vertical and horizontal convolved features or **features map** for digit 1.

Source of the table: Müller-Kett, 2023.

CONVOLUTIONAL NEURAL NETWORKS (CNN) – FILTER/KERNEL

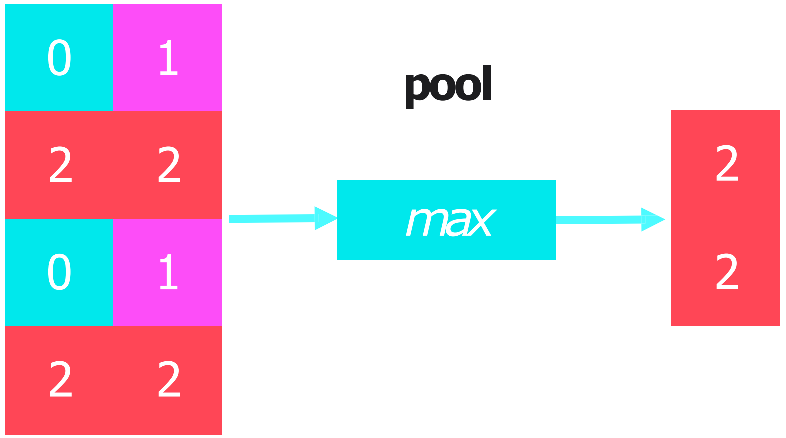




Pooling

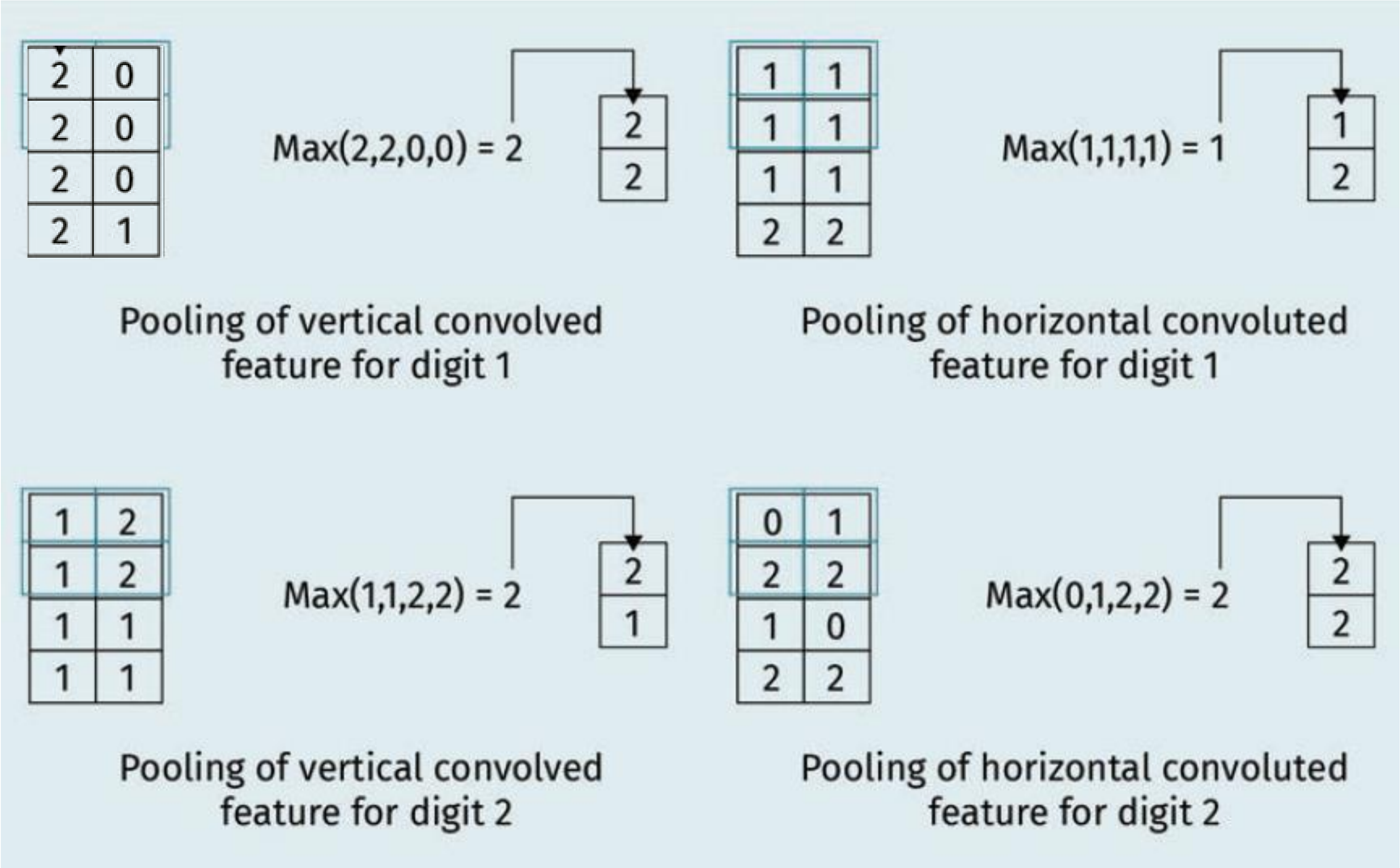
- **Aggregating** convoluted data
- Various pooling **functions**
  - Max
  - Min
  - Mean
  - etc.

Image 4: Pool



Source of the table: Müller-Kett, 2023.

CONVOLUTIONAL NEURAL NETWORKS (CNN) – POOLING



CNN can distinguish the digit 1 if the second element in the list is equal to 2 and the third element is equal to 1. It can distinguish the digit 2, if the second element in the list is equal to 1, and the third element is equal to 2.

- Explain how to automatically generate **transformation features**.
- Understand how to automatically generate **aggregation features**.
- Analyze the **advantages and limitations** of the techniques used to automatically generate features.

SESSION 6

TRANSFER TASK

A start-up that **sells sustainable products in smaller stores worldwide** has been very successful in recent years.

You as a Data Scientist and your team came up with a machine learning model **clustering similar products** (based on products, customers, stocks, tables). Although this clustering supports ordering and shipment, you and your team feel there is still unleashed potential, and the model **does not use all relevant information**. You have already generated several features **manually**, but this did not considerably improve the model's performance.

Discuss ways to **systematically** and **automatically generate additional features** from the existing data. Also, evaluate the **risks** in creating many more features and how these risks can be **mitigated**.

TRANSFER TASK
PRESENTATION OF THE RESULTS

Please present your results.
The results will be discussed in plenary.

1. Which one of the following operators is an example of a transformation primitive?

   a) max
   b) weekday
   c) min
   d) sum

2. Which of the following applies to a feature that was generated as the min(mean()) value?

a) It is of depth 1.
b) It is of depth 2.
c) It is not an interpretable feature.
d) It is not a complex feature.

3. In a convolutional neural network, kernel filters…
   a) … generate the feature map.
   b) … reduce the dimensionality of the feature map.
   c) … are assigned a probability to an input image.
   d) … flatten the feature map.

LIST OF SOURCES

Text

Kanter, J. M., & Veeramachaneni, K. (2015). Deep feature synthesis: Towards automating data science endeavors. 2015 IEEE international conference on data science and advanced analytics (DSAA) (pp. 1—10). IEEE.

Images

Müller-Kett, 2021.
Müller-Kett, 2023.
Microsoft Archive.

Table

Müller-Kett, 2023.

How did you like the course?

HOW DID YOU
LIKE THE COURSE?