



# Unconventional applications of DL for the environment

Artificial Intelligence and  
Data Mining Methods in Ecology

University of Tuscia – Viterbo, July 21–25, 2025

Alex Falcon  
Beatrice Portelli  
University of Udine

**IR0000032 – ITINERIS, Italian Integrated Environmental Research Infrastructures System**  
(D.D. n. 130/2022 - CUP B53C22002150006) Funded by EU - Next Generation EU PNRR-  
Mission 4 "Education and Research" - Component 2: "From research to business" - Investment  
3.1: "Fund for the realisation of an integrated system of research and innovation infrastructures"



## Where We Left Off


Deep Learning: power, potential, and pitfalls

- ✓ We explored how DL can extract features automatically
- ✓ Saw powerful models: VGG, ResNet, Vision Transformers
- ✓ Discussed costs (data, compute) and transfer learning as a solution

⚠ Reminder: DL isn't always the answer...  
but when it works, it really works!

# Where We're Going Today

From models to imagination


 Group discussion:

How could ML/DL help with your data, your questions, your field?

 “Brave New Ideas”:

- Weird, ambitious, interdisciplinary uses of DL for environmental research
- Art, simulation, generation, behavior prediction... why not?

 Short wrap-up / feedback ... then exam time!

 Now that you know the tools, what could you build with them?

## By the end of today, we will...

- ✓ Connect ML/DL ideas with your own research problems
- ✓ Be exposed to unconventional or “crazy” applications of DL
- ✓ Share ideas & feedback with each other
- ✓ Close the course feeling inspired, not just informed ;)

(⚠ Also... Do a brief recap all together before the final test!!!)

## Day 1

Why AI for  
Environment?



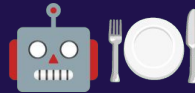
## Day 2

Best  
Practices



## Day 3

ML &  
Preprocessing



## Day 4

Deep  
Learning



## Day 5

Creative  
Application



# Kickoff: What Could You Build with ML/DL?

 Pick one or more to reflect on:

- What's a dataset you already have... or wish you had?
- Is there a messy, boring, or huge task you wish you could automate?
- Do you have an inkling there might be hidden patterns to uncover in your data?
- Can you imagine a DL model helping you simulate or visualize something in your field?
- What would be your “dream model”, no matter how ambitious or weird?

# Our (successful) weird example – Wastewater Pollution

# Our (successful) weird example – Wastewater Pollution

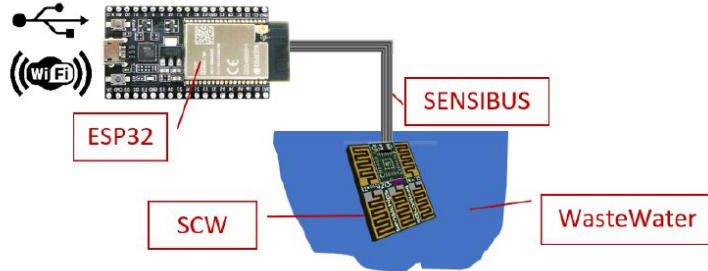
**Objective** Detecting presence of pollutants in WW

Acetic acid	$\text{CH}_3\text{COOH}$
Acetone	$\text{C}_3\text{H}_6\text{O}$
Ammonia	$\text{NH}_3$
Ethanol	$\text{C}_2\text{H}_5\text{OH}$
Formic acid	$\text{CH}_2\text{O}_2$
Hydrochloric acid	$\text{HCl}$
Hydrogen peroxide	$\text{H}_2\text{O}_2$
Phosphoric acid	$\text{H}_3\text{PO}_4$
Sodium hypochlorite	$\text{NaClO}$
Sulphuric acid	$\text{H}_2\text{SO}_4$

# Our (successful) weird example – Wastewater Pollution

**Objective** Detecting presence of pollutants in WW

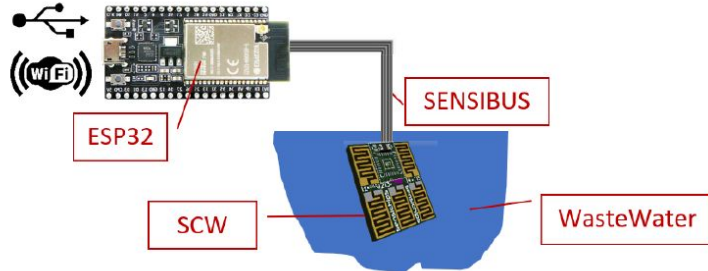
**Input Data** Multi-sensor chip  
 110 timeseries  
 176,000 timestamps



# Our (successful) weird example – Wastewater Pollution

**Objective** Detecting presence of pollutants in WW

**Input Data** Multi-sensor chip  
 110 timeseries  
 176,000 timestamps



- Platinum – Resistance (78 kHz, 200 Hz)  
Platinum – Capacitance (200 Hz)
- Gold – Resistance (78 kHz, 200 Hz)  
Gold – Capacitance (200 Hz)
- Copper – Resistance (200 Hz)  
Copper – Capacitance (200 Hz)
- Nickel – Resistance (200 Hz)  
Nickel – Capacitance (200 Hz)
- Silver – Resistance (200 Hz)  
Silver – Capacitance (200 Hz)

# Our (successful) weird example – Wastewater Pollution

## Objective

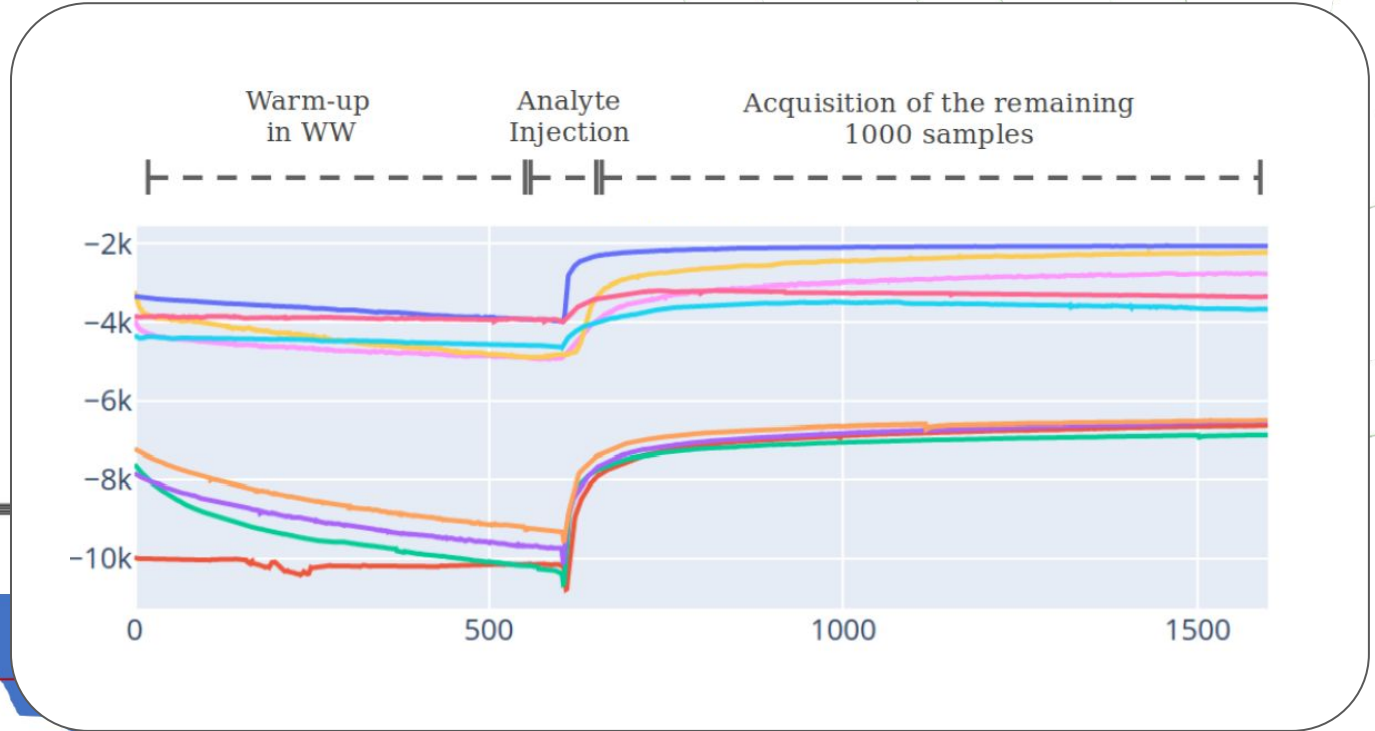
## Input Data

110 timeseries  
176,000 timestamps



ESP32

SCW



# Our (successful) weird example – Wastewater Pollution

 **Objective** Detecting presence of pollutants in WW

 **Input Data** Multi-sensor chip  
110 timeseries  
176,000 timestamps

 **Methods** ML and DL models,  
LLM model (T5)

# Our (successful) weird example – Wastewater Pollution

**Objective** Detecting presence of pollutants in WW

**Input Data** Multi-sensor chip  
110 timeseries  
176,000 timestamps

**Methods** ML and DL models,  
LLM model (T5)

AdaBoost  
Bagging  
KNN  
RF  
SVM

MLP  
T5

# Our (successful) weird example – Wastewater Pollution

**Objective** Detecting presence of pollutants in WW

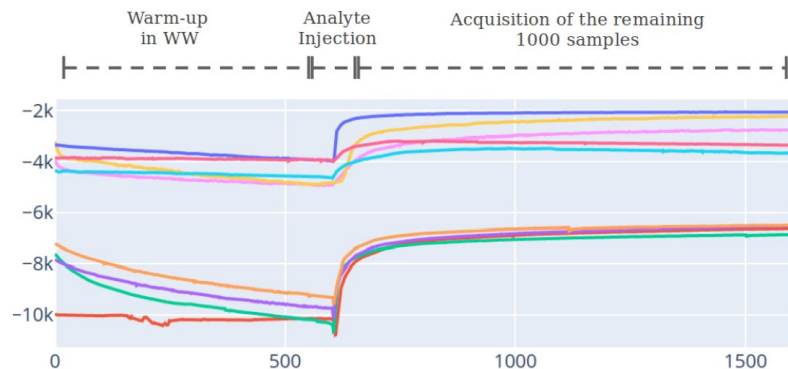
**Input Data** Multi-sensor chip

110 timeseries  
176,000 timestamps

**Methods** ML and DL models,  
LLM model (T5)

AdaBoost  
Bagging  
KNN  
RF  
SVM

MLP  
T5



# Our (successful) weird example – Wastewater Pollution

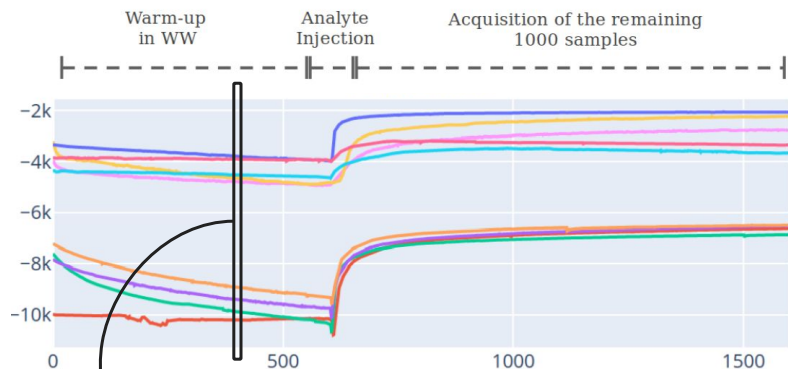
**Objective** Detecting presence of pollutants in WW

**Input Data** Multi-sensor chip  
110 timeseries  
176,000 timestamps

**Methods** ML and DL models, LLM model (T5)

AdaBoost  
Bagging  
KNN  
RF  
SVM

MLP  
T5



# Our (successful) weird example – Wastewater Pollution

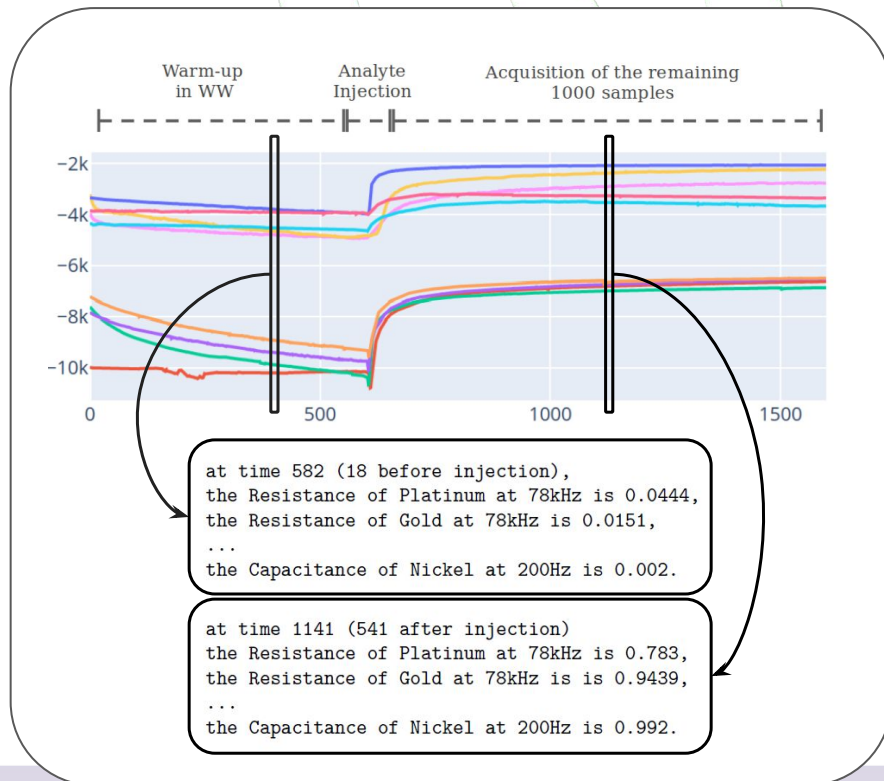
**Objective** Detecting presence of pollutants in WW

**Input Data** Multi-sensor chip  
110 timeseries  
176,000 timestamps

**Methods** ML and DL models, LLM model (T5)

AdaBoost  
Bagging  
KNN  
RF  
SVM

MLP  
T5



# Our (successful) weird example – Wastewater Pollution

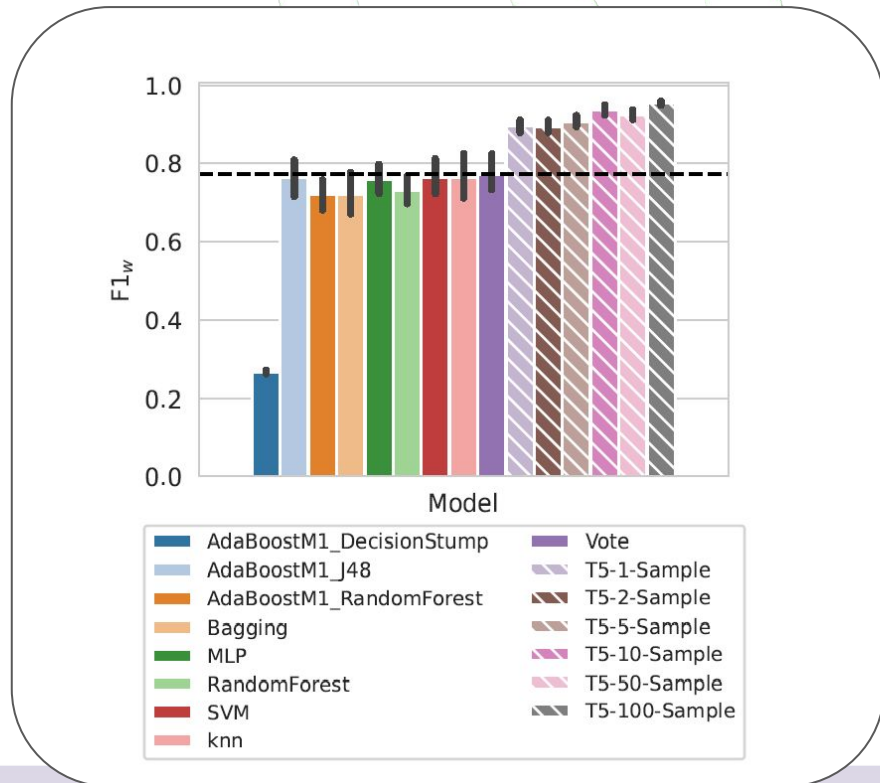
**Objective** Detecting presence of pollutants in WW

**Input Data** Multi-sensor chip  
110 timeseries  
176,000 timestamps

**Methods** ML and DL models, LLM model (T5)

AdaBoost  
Bagging  
KNN  
RF  
SVM

MLP  
T5



# Our (successful) weird example – Wastewater Pollution

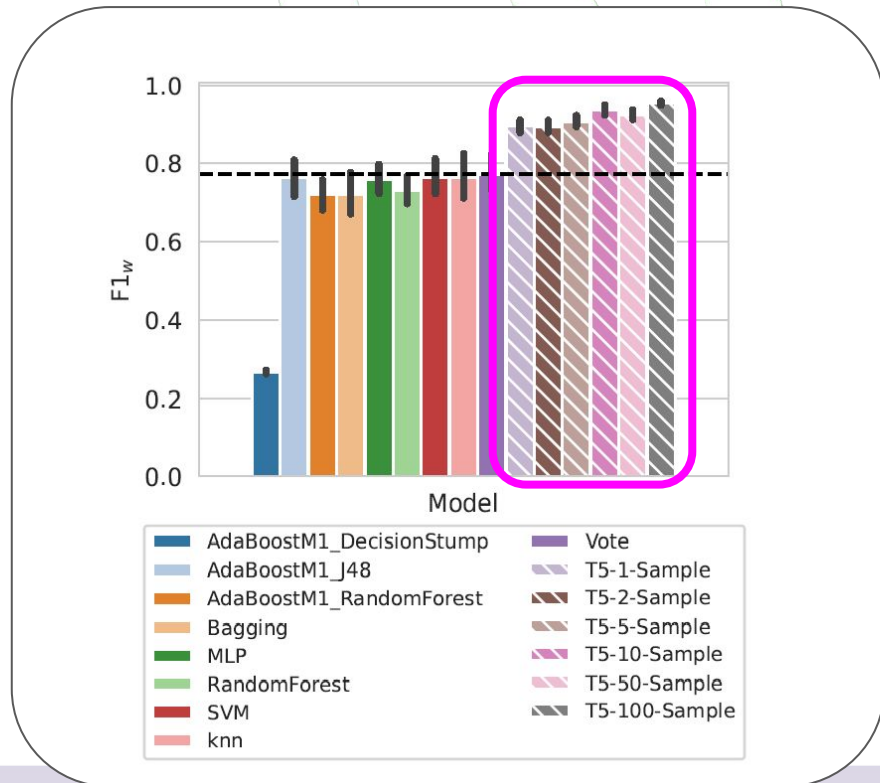
**Objective** Detecting presence of pollutants in WW

**Input Data** Multi-sensor chip  
110 timeseries  
176,000 timestamps

**Methods** ML and DL models, LLM model (T5)

AdaBoost  
Bagging  
KNN  
RF  
SVM

MLP  
T5



# Our (successful) weird example – Wastewater Pollution

**Objective** Detecting presence of pollutants in WW

**Input Data** Multi-sensor chip  
 110 timeseries  
 176,000 timestamps

**Methods** ML and DL models, LLM model (T5)

AdaBoost  
 Bagging  
 KNN  
 RF  
 SVM

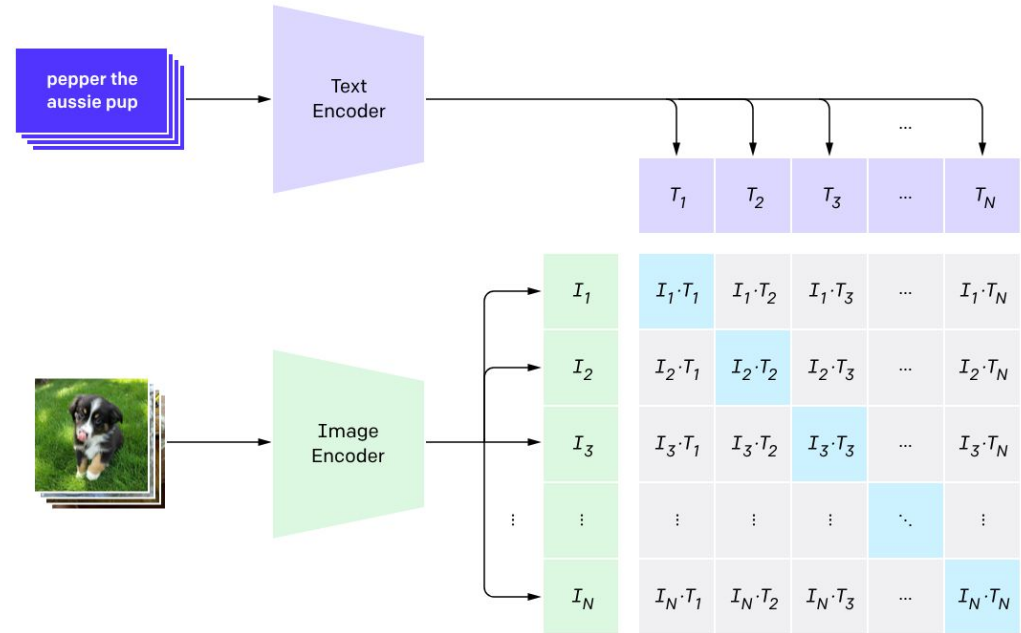
MLP  
 T5

	ACETICACID	1000	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	ACETONE	0	1000	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	AMMONIA	0	0	826	0	0	0	0	0	0	0	171	0	3						
	ETHANOL	0	0	0	995	0	0	5	0	0	0	0	0	0						
	FORMICACID	0	0	0	0	992	0	0	0	0	0	0	3	5						
Real	HYDROCHLORICACID	0	0	0	0	0	970	0	0	0	0	30	0							
	HYDROGENPEROXIDE	0	0	0	0	0	0	1000	0	0	0	0	0							
	PHOSPHORICACID	208	0	0	0	39	0	0	753	0	0	0	0							
	SODIUM_HYPOCHLORITE	0	0	0	0	0	0	0	0	1000	0	0	0							
	SULPHURICACID	0	0	0	0	0	2	0	0	0	0	998	0							
	SWW	0	0	0	0	0	0	0	0	0	0	0	1000							
	ACETICACID																			
	ACETONE																			
	AMMONIA																			
	ETHANOL																			
	FORMICACID																			
	HYDROCHLORICACID																			
	HYDROGENPEROXIDE																			
	PHOSPHORICACID																			
	SODIUM_HYPOCHLORITE																			
	SULPHURICACID																			
	SWW																			
	Predicted																			

# Our (less successful) weird example – Plant CLIP

# Our (less successful) weird example – Plant CLIP

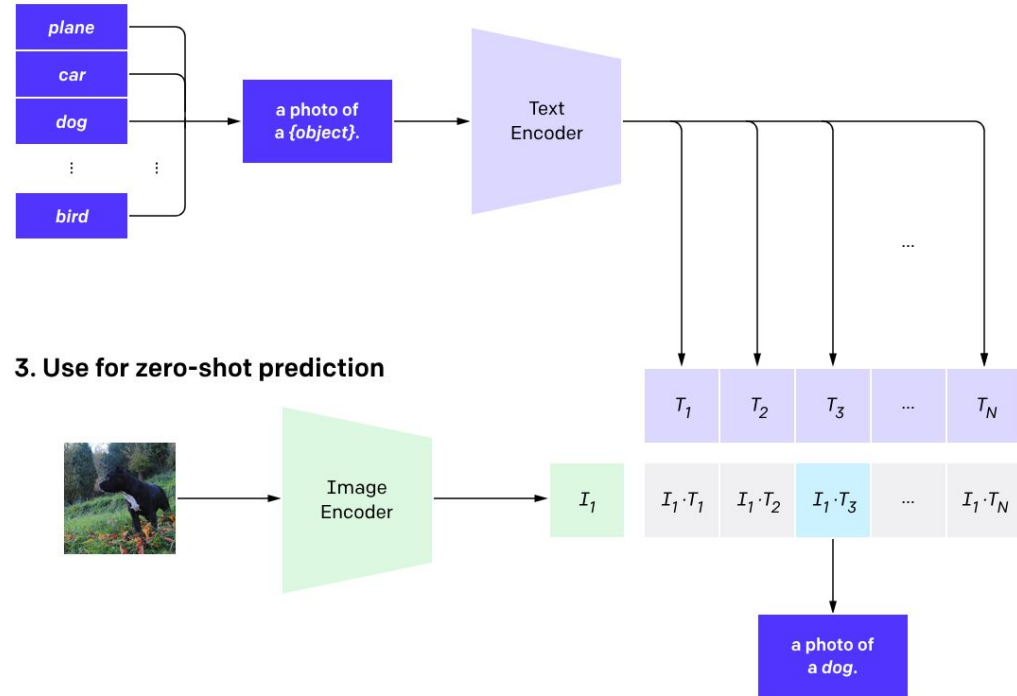
- CLIP is a famous multimodal model for connecting texts and images
- Texts are transformed into embeddings (abstract concepts)
- Images are also transformed into embeddings
- Text and image embeddings should be similar for similar things



# Our (less successful) weird example – Plant CLIP

For predictions:

- you have a new image
- you know all the possible output classes of the image (e.g., plane, car, dog, bird, ...)
- create the embedding of the photo
- create the embedding of all output classes (“This is a photo of a plane”, “This is a photo of a dog”, ...)
- check which text embedding is most similar to the photo embedding!



# Our (less successful) weird example – Plant CLIP

Apple  
Black Rot



Bell Pepper  
Bacterial



Blueberry  
Healthy



Cherry  
Powdery Mildew



Corn  
Gray Spots



Grape  
Black Rot



Potato  
Early Blight



Given an image of a leaf: classify which plant species and which disease (if any)

# Our (less successful) weird example – Plant CLIP

name	base_description	description
Apple	Simple, ovate, serrated margins, petiolate, alternate. Color: upper blade is dark/olive green, lower blade is pale green. The leaves turn yellow, orange, or red in autumn.	Image of Apple leaves. Appearance: Simple, ovate, serrated margins, petiolate, alternate. Color: upper blade is dark/olive green, lower blade is pale green. The leaves turn yellow, orange, or red in autumn.
Blueberry	Simple, elliptical, entire margins, petiolate, alternate. Color: green. The leaves turn purple or red in autumn.	Image of Blueberry leaves. Appearance: Simple, elliptical, entire margins, petiolate, alternate. Color: green. The leaves turn purple or red in autumn.
Cherry	Simple, ovate-acute, serrated margins, petiolate, alternate. Color: dark green with green/reddish petiole. The leaves turn red or purple in autumn.	Image of Cherry leaves. Appearance: Simple, ovate-acute, serrated margins, petiolate, alternate. Color: dark green with green/reddish petiole. The leaves turn red or purple in autumn.
Corn	Simple, linear-lanceolate, entire margins, sessile, alternate. Color: dark green. Leaves turn brown at maturity.	Image of Corn leaves. Appearance: Simple, linear-lanceolate, entire margins, sessile, alternate. Color: dark green. Leaves turn brown at maturity.
Grape	Simple, typically lobate (3 to 9 lobes) but sometimes deltoid, dentate margins, petiolate, alternate. Color: dark green. Leaves turn orange, purple or red in autumn.	Image of Grape leaves. Appearance: Simple, typically lobate (3 to 9 lobes) but sometimes deltoid, dentate margins, petiolate, alternate. Color: dark green. Leaves turn orange, purple or red in autumn.
Orange	Simple, ovate, crenated margins, petiolate, winged petiole, alternate. Color: glossy dark green.	Image of Orange leaves. Appearance: Simple, ovate, crenated margins, petiolate, winged petiole, alternate. Color: glossy dark green.

# Our (less successful) weird example – Plant CLIP

Leaf_scorch	Fungi	Diplocarpon earlianum	A non-infectious condition caused by environmental stress, leading to browning of leaf edges.	Plant affected by Leaf scorch. That is A non-infectious condition caused by environmental stress, leading to browning of leaf edges.
Tomato_mosaic_virus	Virus	Tomato Mosaic Virus	A viral disease resulting in mottled green and yellow leaves, stunted growth, and malformed fruits.	Plant affected by Tomato mosaic virus. That is A viral disease resulting in mottled green and yellow leaves, stunted growth, and malformed fruits.
Northern_Leaf_Blight	Fungi	Exserohillum turrcicum	A fungal disease causing large, cigar-shaped gray to tan spots on the leaves.	Plant affected by Northern Leaf Blight. That is A fungal disease causing large, cigar-shaped gray to tan spots on the leaves.
Cedar_apple_rust	Fungi	Gymnosporangium-juniper	A fungal disease causing premature defoliation, fruit distortion and abortion.	Plant affected by Cedar apple rust. That is A fungal disease causing premature defoliation, fruit distortion and abortion.
Leaf_Mold	Fungi	Fulvia fulva	A fungal disease prevalent in high humidity, causing pale green to yellow spots on the upper surfaces of leaves and a velvety growth on the undersides.	Plant affected by Leaf Mold. That is A fungal disease prevalent in high humidity, causing pale green to yellow spots on the upper surfaces of leaves and a velvety growth on the undersides.
Cercospora_leaf_spot Gray_leaf_spot	Fungi	Cercospora zeae maydis	A fungal disease causing small, round to irregular spots that are grayish to brown, primarily affecting leaves.	Plant affected by Cercospora leaf spot Gray leaf spot. That is A fungal disease causing small, round to irregular spots that are grayish to brown, primarily affecting leaves.
healthy			Healthy plant, which is free from any visible signs of diseases and pests.	Healthy plant, which is free from any visible signs of diseases and pests.

# Our (less successful) weird example – Plant CLIP

In the end... it didn't work as well as we hoped.

## Our (less successful) weird example – Plant CLIP

In the end... it didn't work as well as we hoped.

- The “CLIP” models still worked, but a little bit worse than “normal” models trained on images only
- The “CLIP” models didn't generalize better than image-only models

Nice try, but maybe the images were not enough or the descriptions needed to be more precise.

# The very final recap

# The very final recap

*What have we seen throughout this course?*

ML and DL can be powerful tools that learn from data how to solve complex problems – even when dealing with environmental sciences

# The very final recap

*What have we seen throughout this course?*

ML and DL can be powerful tools that learn from data how to solve complex problems – even when dealing with environmental sciences

Especially thanks to more and more data being collected

# The very final recap

*What have we seen throughout this course?*

ML and DL can be powerful tools that learn from data how to solve complex problems – even when dealing with environmental sciences

Especially thanks to more and more data being collected

However: heterogeneity, noisy, incomplete, **unlabelled** samples

# The very final recap

*What have we seen throughout this course?*

The importance of data splits: train, validation, test

How the train/val/test loop works

How to perform validation and testing properly when you have few data

# The very final recap

*What have we seen throughout this course?*

The importance of data splits: train, validation, test

How the train/val/test loop works

How to perform validation and testing properly when you have few data

*Feature engineering*, or how to extract useful information from your raw data

Tabular → statistical descriptors; images → shape, texture, color, local

# The very final recap

*What have we seen throughout this course?*

The importance of data splits: train, validation, test

How the train/val/test loop works

How to perform validation and testing properly when you have few data

*Feature engineering*, or how to extract useful information from your raw data

Tabular → statistical descriptors; images → shape, texture, color, local

Some limitations related to complexity, need for expertise, etc yet DL is **not** always the key to automatically perform this step

# The very final recap

*What have we seen throughout this course?*

What it means for a machine to “learn”: task, experience, performance

Machine Learning and the ability to automatically solve a problem, starting from useful features describing the input

# The very final recap

## What have we seen throughout this course?

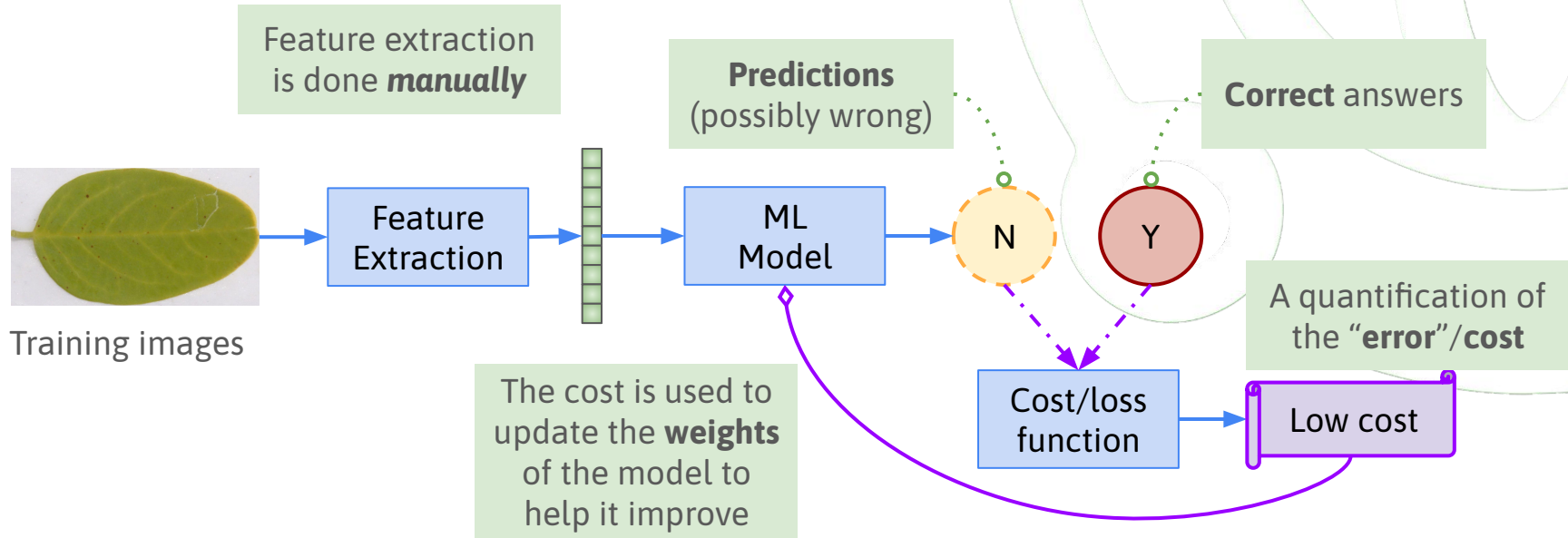
What it means for a machine to “learn”: task, experience, performance

Machine Learning and the ability to automatically solve a problem, starting from useful features describing the input

The main types of ML and when to use them:

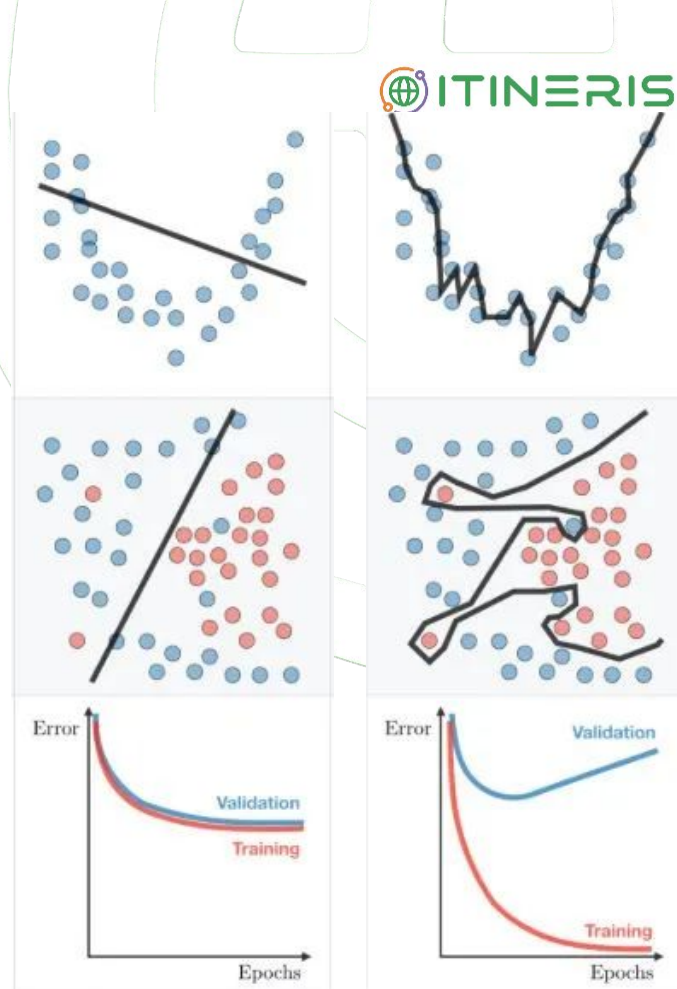
- **supervised** → learning to solve a task *from labelled examples* (i.e. each has a “correct” value we can learn from)
- **unsupervised** → *no labels: we use it to explore data and identify patterns*
- reinforcement → learning policies to achieve goal in a complex environment
- generation → the model learns to generate new content from input prompts

# Supervised ML



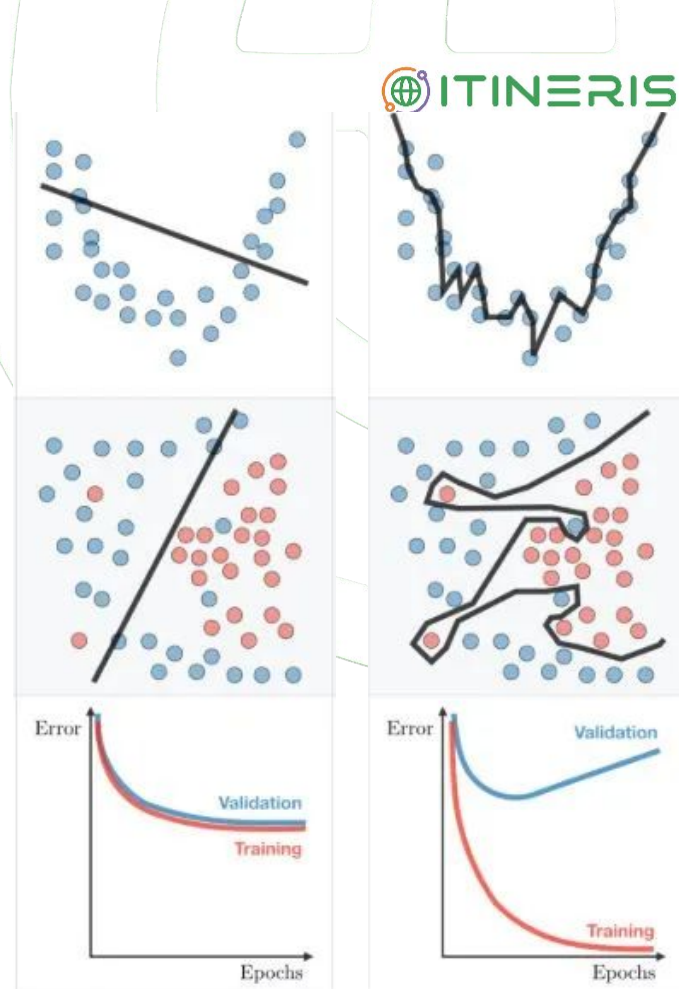
# Supervised ML - Check your loss!

- Always compare your model's loss (and/or performance) on the training and validation data



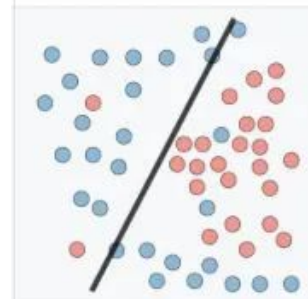
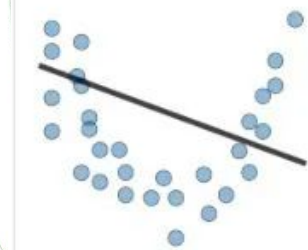
# Supervised ML - Check your loss!

- Always compare your model's loss (and/or performance) on the training and validation data
- Things you want to look out for:
  - **Underfitting:** you are not learning enough! Your model is **too simple** and you have low performance on both training and validation data
  - **Overfitting:** you are focusing on details! Your model is **too complex** and is “memorizing” the training set, including details which do not generalize on new data.



# Supervised ML - Check your loss!

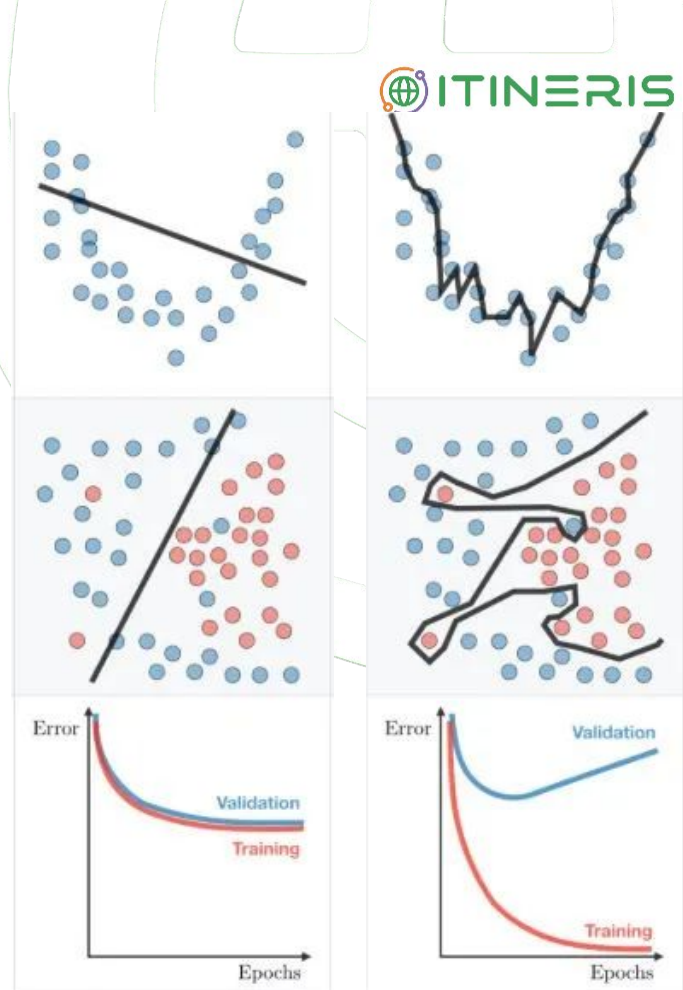
You evaluate your model on the *test set* and obtain a **bad value** for one metric.  
What can we say about the result?



# Supervised ML - Check your loss!

You evaluate your model on the *test set* and obtain a **bad value** for one metric.  
What can we say about the result?

You evaluate your model on the *test set* and obtain a **close-to-perfect value** for one metric.  
What can we say about the result?



# Supervised ML - A recap

- **Regression**

- We want to *predict a numerical value* of some kind
- E.g. crop yield prediction, carbon sequestered in trees, calories estimation from food image, ...

# Supervised ML - A recap

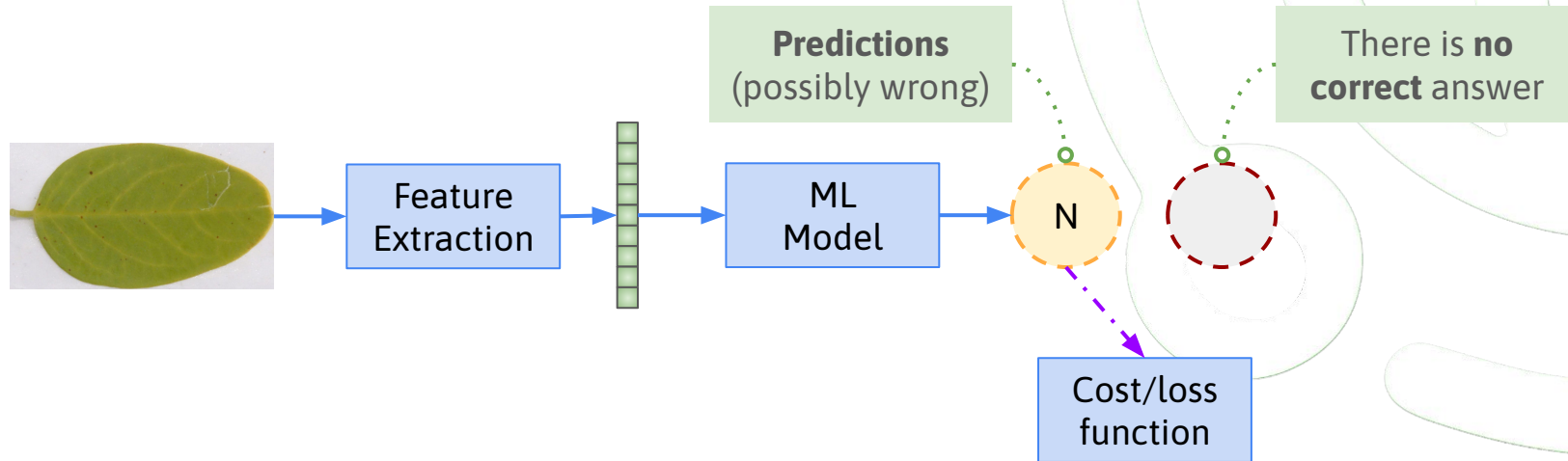
- **Regression**

- We want to *predict a numerical value* of some kind
- E.g. crop yield prediction, carbon sequestered in trees, calories estimation from food image, ...

- **Classification**

- **Binary** classification → whether the sample is *a member of the class or not*
  - E.g. healthy/unhealthy leaves, is the kiwi ripe or not, ...
- **Multiclass** classification → to *which one of some fixed classes* does my sample belong to
  - E.g. recognizing animal species, leaf species, which *disease* is affecting this plant, ...
- **Multilabel** classification → identify *all classes* to which my samples belongs to
  - E.g. which of these characteristics are valid for this animal, which *diseases* are affecting this plant, ...

# Unsupervised ML - The lack of supervision



Unsupervised ML is used when **no labels** are available.

It is used for **exploring the data** and see if there are **hidden patterns** we were not aware of.

# Unsupervised ML - A recap

- **Clustering**

- Can we automatically find **data groupings** in the data samples we have?

# Unsupervised ML - A recap

- **Clustering**

- Can we automatically find **data groupings** in the data samples we have?

- **Dimensionality Reduction**

- High dimensional (4+) data → *difficult to visualize, interpret, analyze*
- Can we reduce the dimensionality while preserving initial relations? (e.g. PCA)

# Unsupervised ML - A recap

- **Clustering**

- Can we automatically find **data groupings** in the data samples we have?

- **Dimensionality Reduction**

- High dimensional (4+) data → *difficult to visualize, interpret, analyze*
- Can we reduce the dimensionality while preserving initial relations? (e.g. PCA)

- **Manifold Learning**

- “Standard” dimensionality reduction is usually searching for linear relations
- Here we try to build some kind of “**nonlinear-sensitive**” PCA (e.g. TSNE)

# Unsupervised ML - A recap

- **Clustering**

- Can we automatically find **data groupings** in the data samples we have?

- **Dimensionality Reduction**

- High dimensional (4+) data → difficult to visualize, interpret, analyze
- Can we reduce the dimensionality while preserving initial relations? (e.g. PCA)

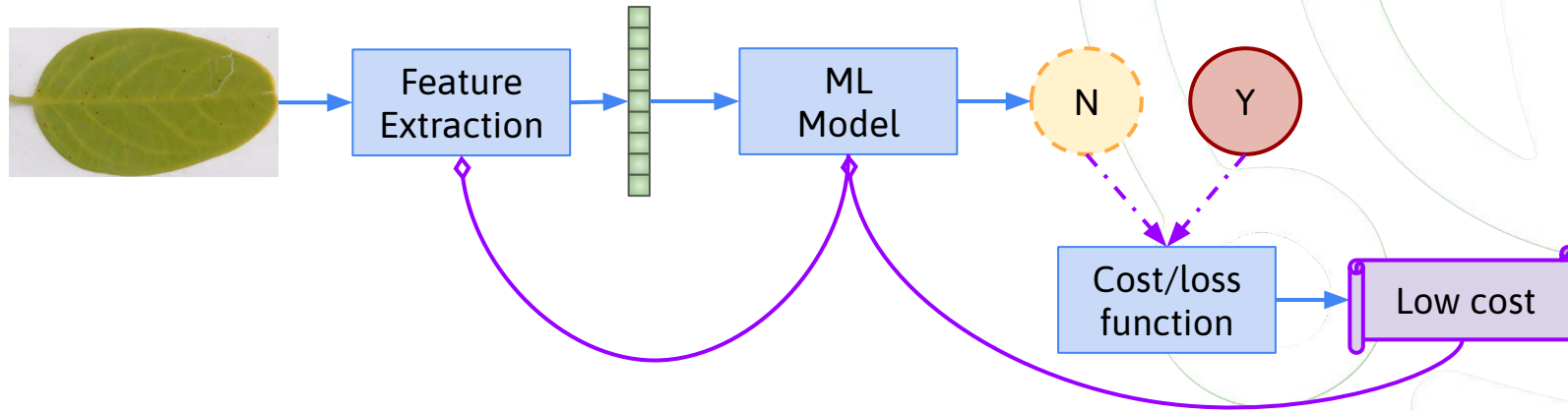
- **Manifold Learning**

- “Standard” dimensionality reduction is usually searching for linear relations
- Here we try to build some kind of **“nonlinear-sensitive”** PCA (e.g. TSNE)

- **Outlier/novelty/anomaly detection**

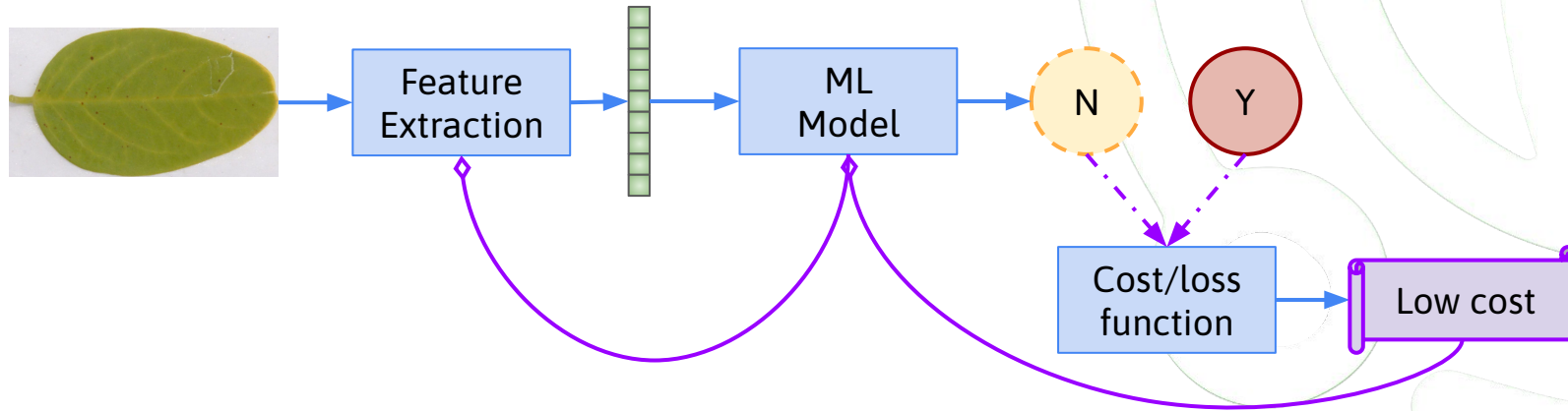
- Models learn what is **“normal”** and try to identify **“abnormal”** situations
- Outlier det. → polluted training data; novelty/anomaly det. → non-polluted

# Supervised Deep Learning



Main difference with ML “alone”: feature extraction is done **automatically** in DL.  
Think of convolutions!

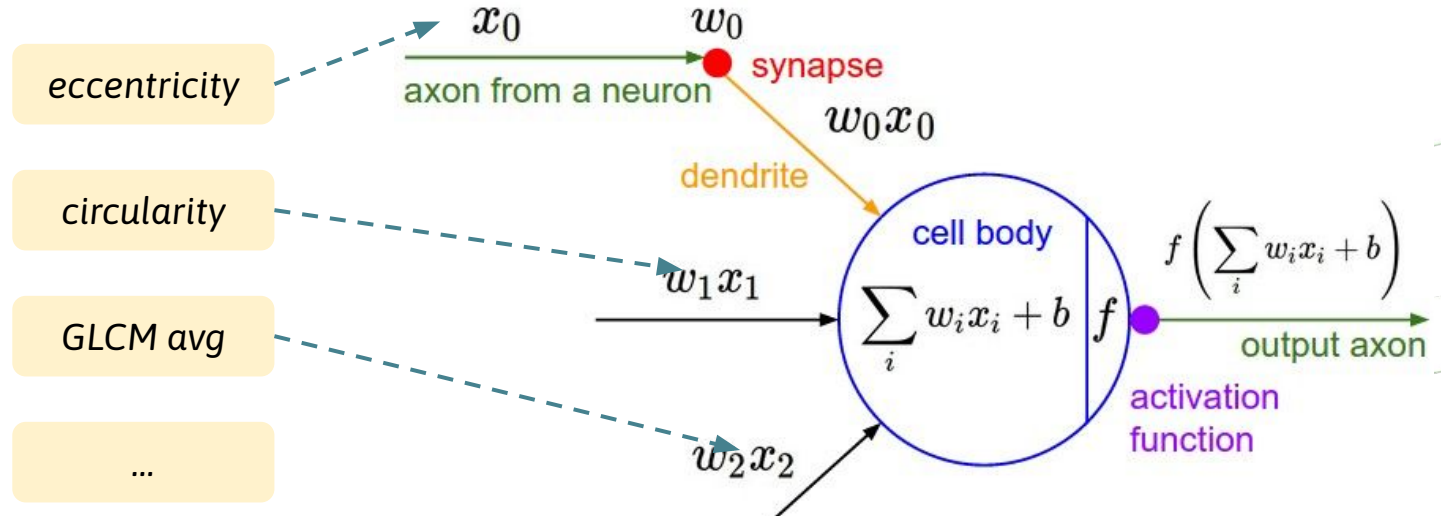
# Supervised Deep Learning



Main difference with ML “alone”: feature extraction is done **automatically** in DL.  
Think of convolutions!

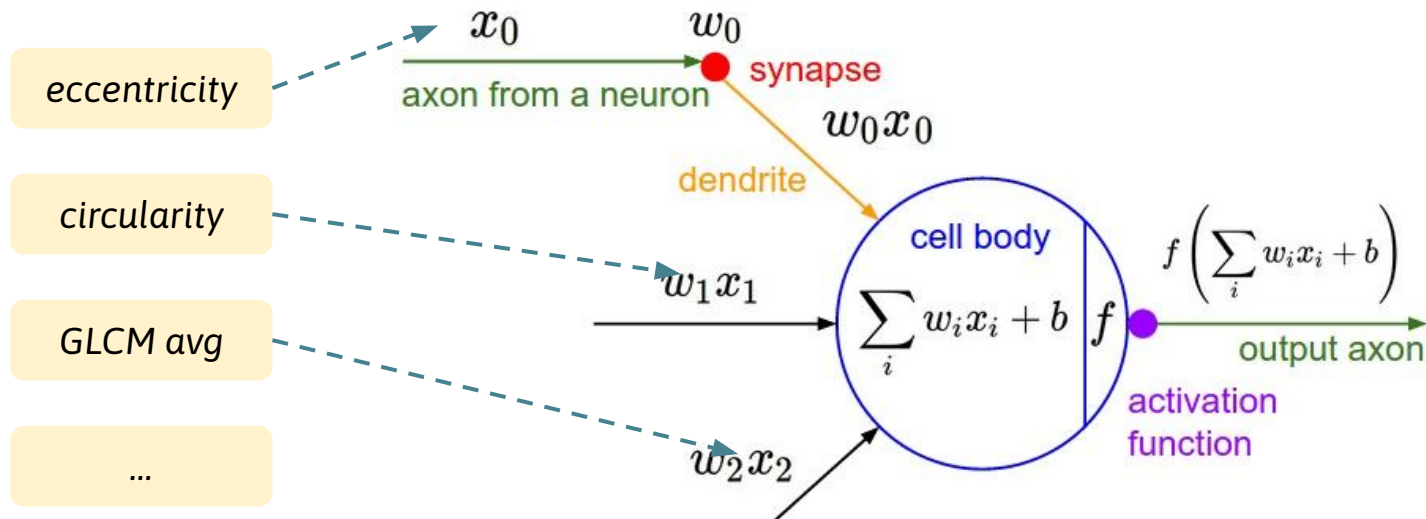
On top of the extracted features, we still have a “ML model” that learns to solve the problem. (e.g. MLP on top of convolutional features in VGGNet)

# Neurons and how they process inputs



Neurons learn some kind of “**weighted sum**” of the input features

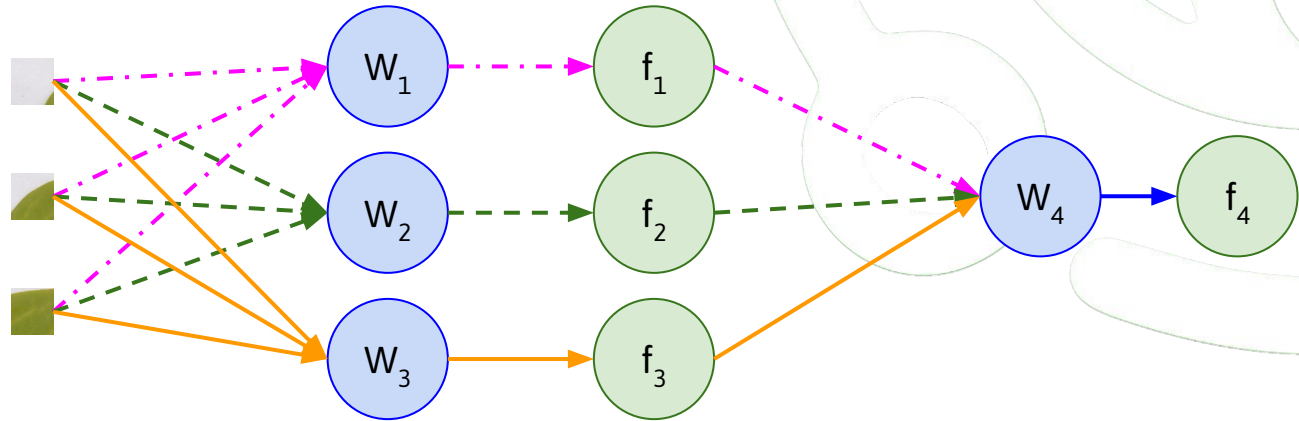
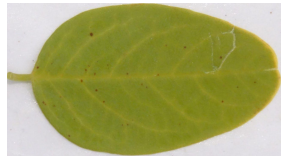
# Neurons and how they process inputs



Just a small note: manually extracted features here, it's still "ML"

Neurons learn some kind of "weighted sum" of the input features

# Multi-layer perceptron (MLP)



Neurons on their own are not effective at learning features from pixels

# The lack of spatiality



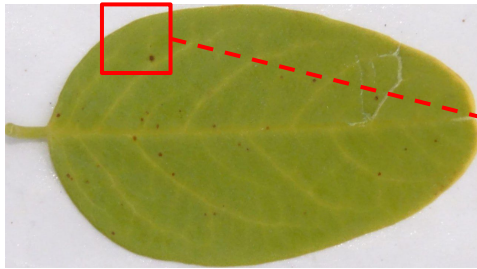
Key insight: the MLP has no direct access to spatial information, yet images make sense because pixels are arranged in some clear way, along both dimensions.



E.g. these small regions make (some) sense to us.

# Towards *Convolutional* neural networks (CNN)

CNNs are networks that use convolutions, i.e. small “filters” that sequentially scan the image and produce an output.



$p_{1,1}$	$p_{1,2}$	$p_{1,3}$
$p_{2,1}$	$p_{2,2}$	$p_{2,3}$
$p_{3,1}$	$p_{3,2}$	$p_{3,3}$

Imagine this is a 3x3 region

With convolutional filters we can **automatically** learn *local spatial features*.

$w_{1,1}$	$w_{1,2}$	$w_{1,3}$
$w_{2,1}$	$w_{2,2}$	$w_{2,3}$
$w_{3,1}$	$w_{3,2}$	$w_{3,3}$

Convolutional filter/kernel of size 3x3



$w_{1,1}p_{1,1}$	$w_{1,2}p_{1,2}$	$w_{1,3}p_{1,3}$
$w_{2,1}p_{2,1}$	$w_{2,2}p_{2,2}$	$w_{2,3}p_{2,3}$
$w_{3,1}p_{3,1}$	$w_{3,2}p_{3,2}$	$w_{3,3}p_{3,3}$

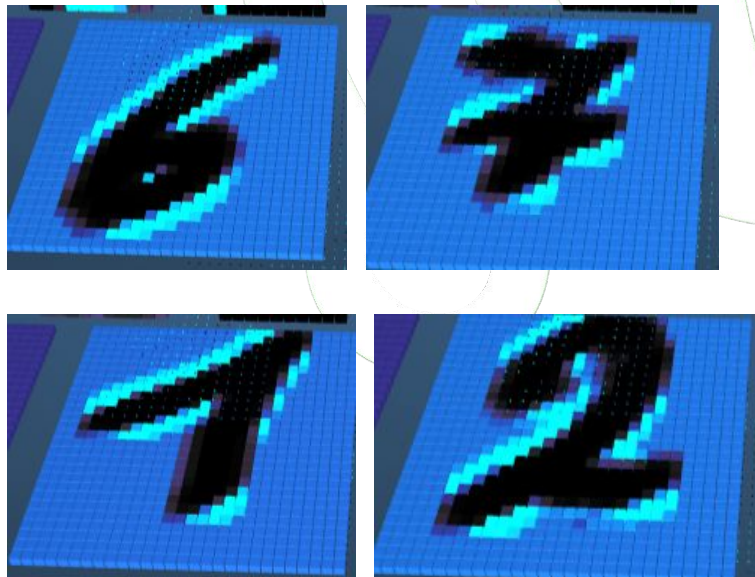
Then, output the sum:

$$w_{1,1}p_{1,1} + w_{1,2}p_{1,2} + \dots$$

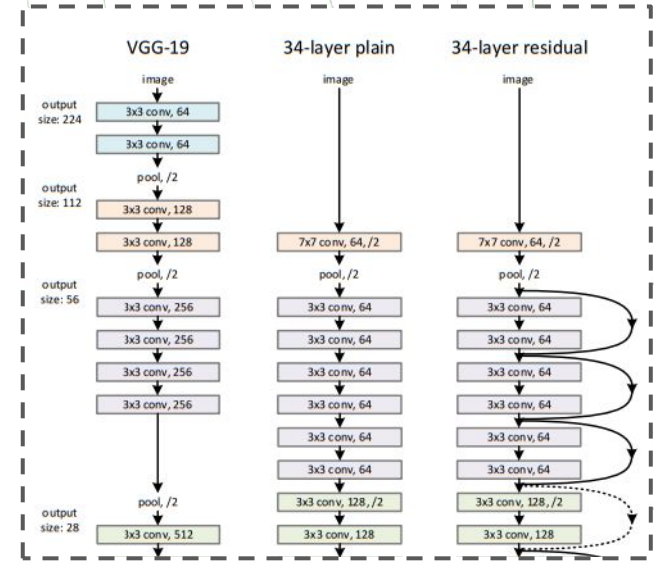
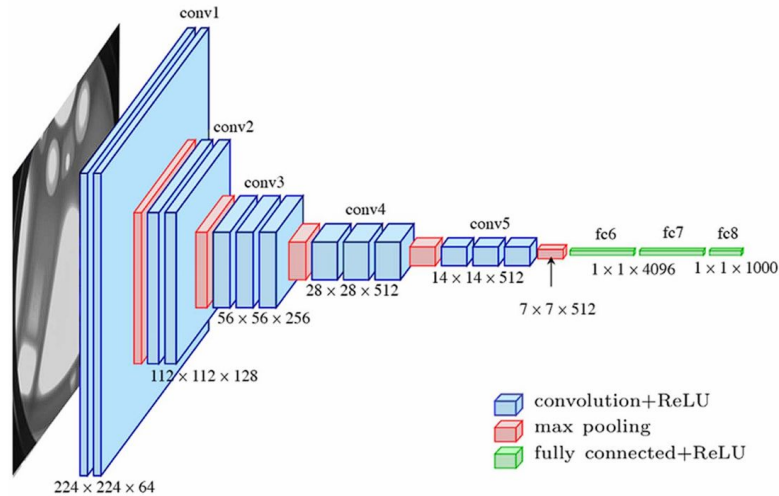
# What's inside these neurons?

1st level features seem to capture patterns in a fixed direction

If we visualize the **responses (feature maps)** on the images, we see that they seem to have learned some specific patterns... we can somehow interpret what is happening under the hood

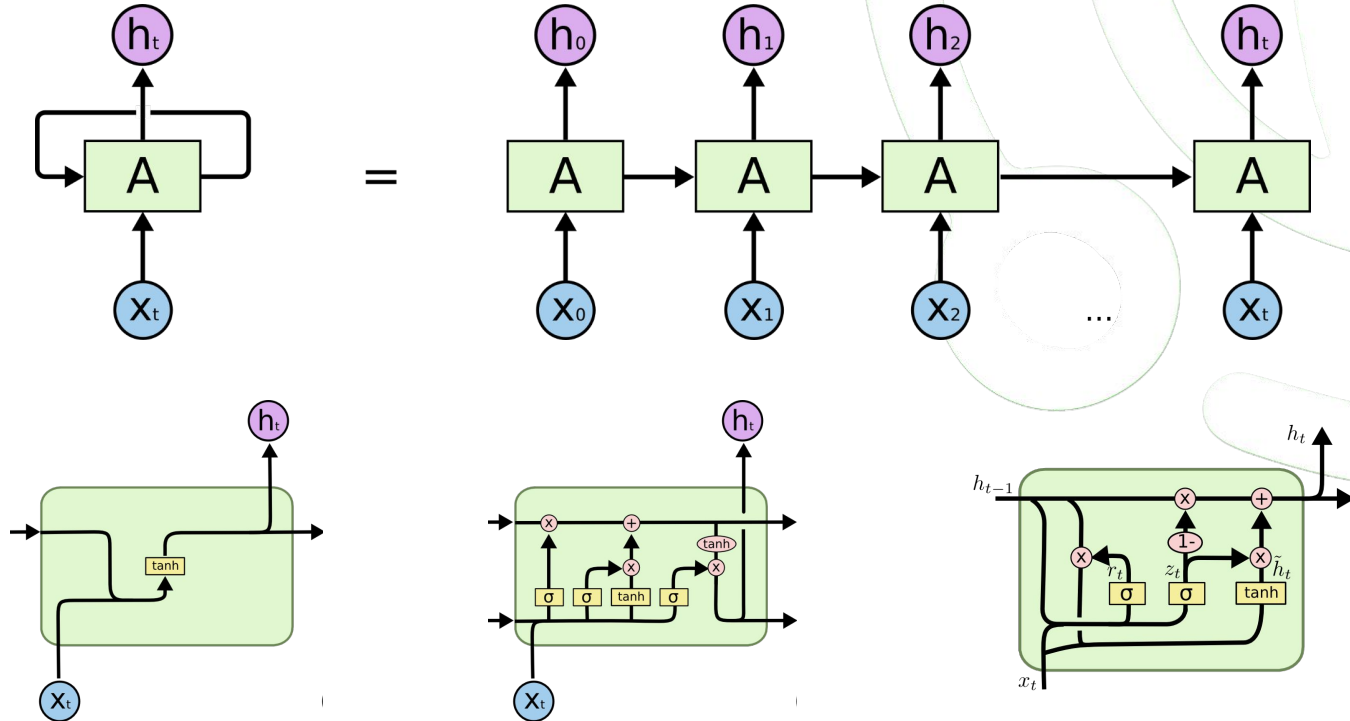


# Stacking more layers and tradeoffs

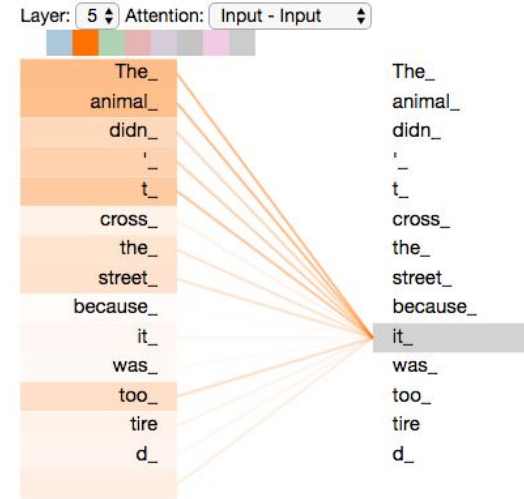
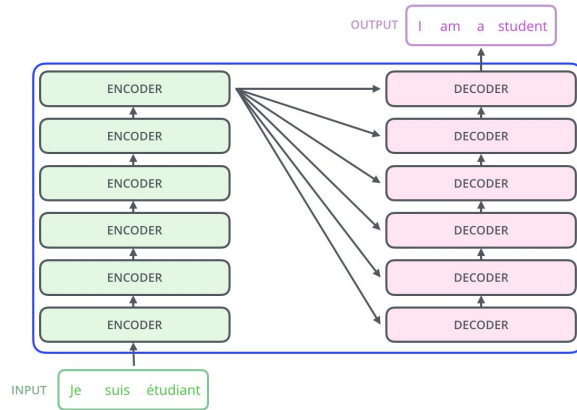


More layers → higher abstraction. Yet, problems with vanishing gradients.

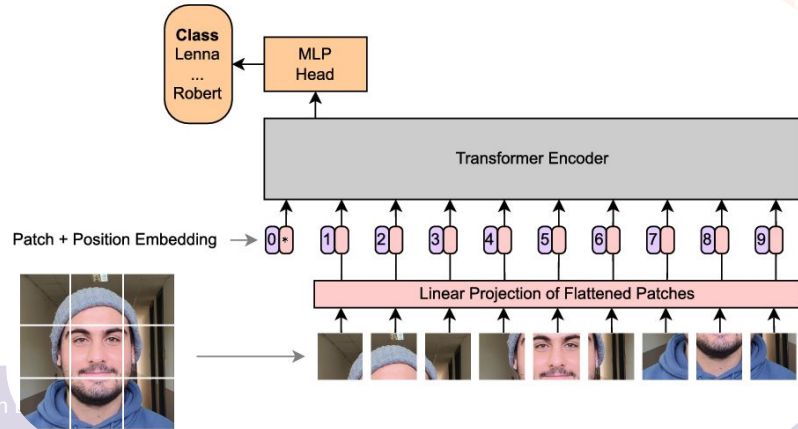
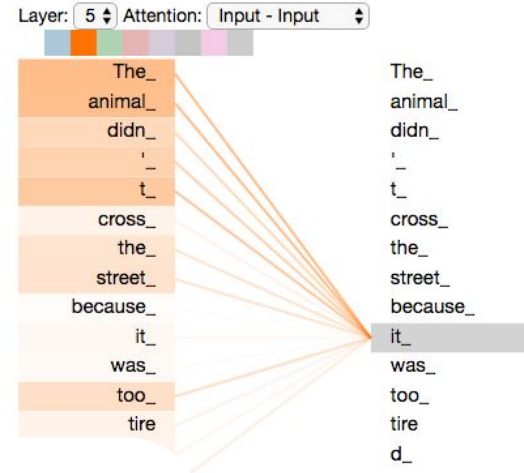
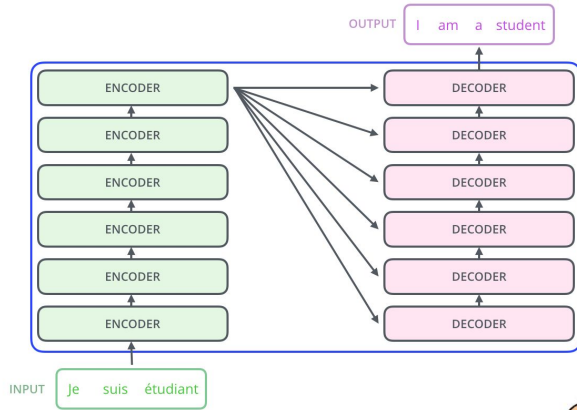
# Sequential data



# Transformers - text and visual data



# Transformers - text and visual data



# Limitations



Training DL models is costly



Not accessible to everyone



Environmental impact



Interpretability

- Not a good idea to start designing DL networks from scratch. *Always start from the literature, even on related problems.*
  - You may find code/models or some other starting points.
- Moreover, they are costly to train.
- DL is not always the answer: sometimes *simpler models can work much better*, e.g. on small datasets.
- DL models are *difficult to interpret*.



# THANKS!

**Alex Falcon**  
**Beatrice Portelli**

[falcon.alex@spes.uniud.it](mailto:falcon.alex@spes.uniud.it)

[portelli.beatrice@spes.uniud.it](mailto:portelli.beatrice@spes.uniud.it)

**IR0000032 – ITINERIS, Italian Integrated Environmental Research Infrastructures System**  
(D.D. n. 130/2022 - CUP B53C22002150006) Funded by EU - Next Generation EU PNRR-  
Mission 4 "Education and Research" - Component 2: "From research to business" - Investment  
3.1: "Fund for the realisation of an integrated system of research and innovation infrastructures"



Finanziato  
dall'Unione europea  
NextGenerationEU



Ministero  
dell'Università  
e della Ricerca

