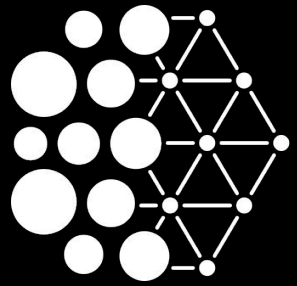
Quebec
Artificial
Intelligence
Institute



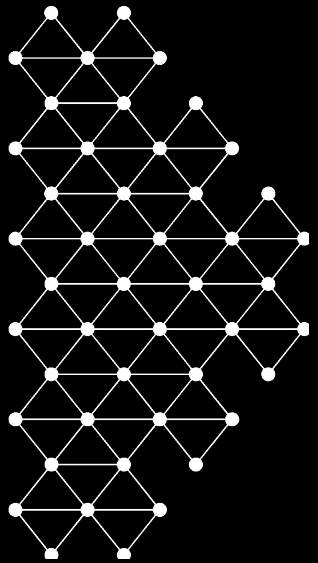
Mila

Machine learning and experimental protocol

Gaétan Marceau Caron
Applied research scientist, Mila
gaetan.marceau.caron@mila.quebec

Quebec
Artificial
Intelligence
Institute

Mila

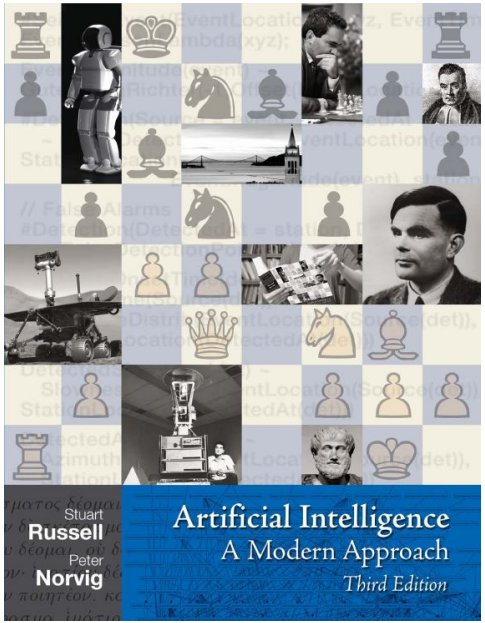


Artificial intelligence and machine learning

Artificial intelligence

- How to simulate **intelligence** on a machine?
- Computer anthropomorphism
- This research gives insights on humans, animals, and physical computers.
- Main question: are humans and animals Turing machines?
- Multi-disciplinary field of study.

1051 pages

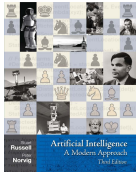


Source: http://aima.cs.berkeley.edu/cover.html

Used in over 1400 universities in over 125 countries.
The 22nd most cited computer science publication on Citeseer. (Published in 1994, last edition 2009)

Machine learning

- How to simulate **learning from examples** on a machine?
- Goal: *learning new algorithms from trial and errors*.
- It is the most active subfield of artificial intelligence in terms of **research** and **industrialization of new techniques**.

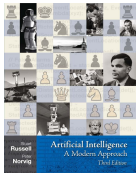


161/1051 pages

Part V: Learning

Deep learning

Set of machine learning techniques that implement the idea that **learning in high-dimensional space** occurs with **distributed representations** computed with **specific architectures**.



9/1051 pages

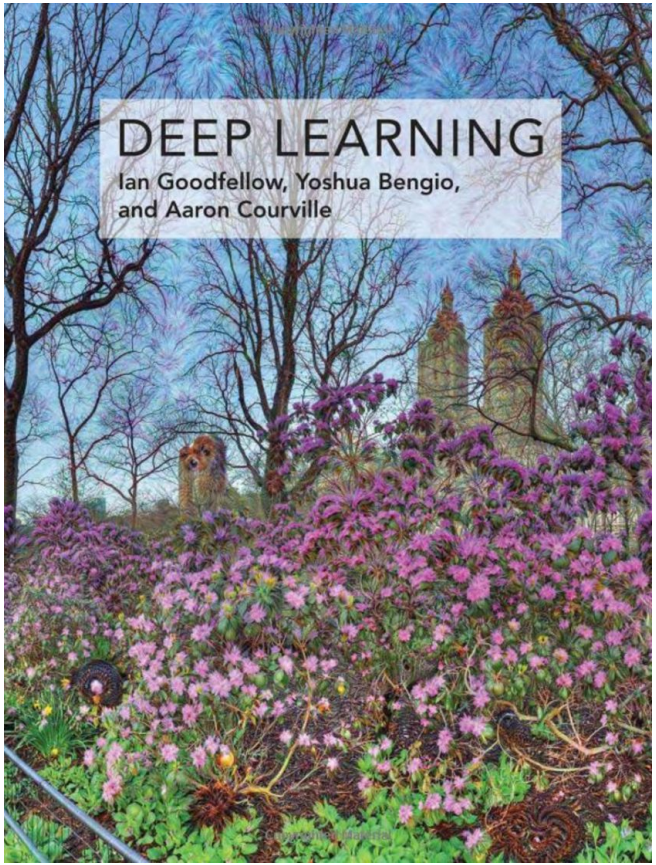
Part V: Learning

Chapter 18: Learning from Examples ... 693

Mila

Deep learning

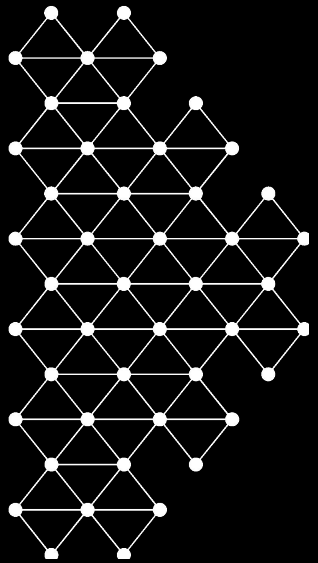
Set of machine learning techniques that implement the idea that **learning in high-dimensional space** occurs with **distributed representations** computed with **specific architectures**.



Source: https://www.deeplearningbook.org/

Quebec
Artificial
Intelligence
Institute

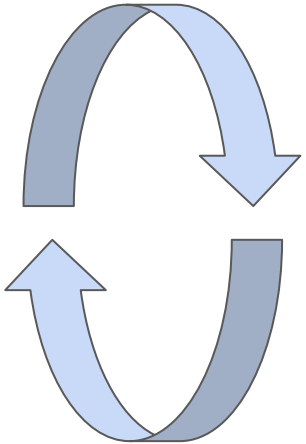
Mila



Machine learning methodology

Machine Learning methodology

1. Define a task to be done by the computer.
2. Define the performance measures used to evaluate the quality of the algorithm performing the task.
3. Gather the data required to train models.
4. Perform the experiments to find the best models.
5. Deploy the best models in production.
6. Retrieve more data and iterate.



Define the task

- A task description only concerns “*what*” the computer should do.
- At this stage, we should not describe “*how*” it should do it.
- The description of the task should be short and informal.
- Specify the input (domain) and the output (target) of the algorithm.

When do you need ML?

- Tasks for which it is hard to program how they should be solved.
- Tasks related to the analysis of complex data.
- Tasks that require the adaptivity of the algorithm over time.

Source: Adam Sherez, Unsplash

Examples of a task

- “Detecting **spam** emails.”
- “Restore colors in a B&W image.”
- “Detecting **violent** scenes in videos.”
- “Predicting the next failure of a system.”
- “Recommending new items in function of users’ preferences.”
- “Generating new molecules with given properties.”
- …

Realistic tasks

- Hard to know since it depends on:
  - current evidence gathered in research,
  - available data to support learning,
  - computational resources needed by the learning algorithm.
- Dedicate time to read **scientific literature** and attend **scientific conferences**.

Define the performance measures

A good performance measure should:

- evaluate if the algorithm successfully achieves a specific goal,
- be easy to interpret and communicate,
- be generic for evaluating different algorithms.

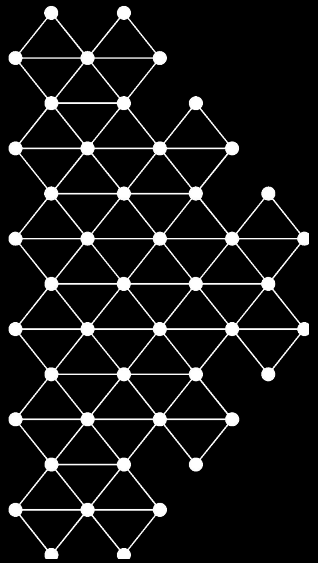
We must evaluate the performance measures in a **production context**.

Gather the data required to train a model

- Annotating data takes time and costs money.
- Never sacrifice quality for quantity.
- Always use mechanisms to ensure quality.
- Identify opportunities for data collection.

Quebec
Artificial
Intelligence
Institute

Mila



Experimental protocol

Hyperparameters

- An hyperparameter is a parameter of the learning algorithm as opposed to a model parameter,
- It cannot be optimized directly by the model optimizer,
- It is possible to use hyperparameter tuning algorithms,
  - Oríon (https://orion.readthedocs.io)
- Hyperparameter tuning is part of the ML expertise.

The experiment pipeline

1. Define an experiment,
2. Create the input pipeline,
3. Create visualization tools,
4. Create the training loop,
5. Create the hyperparameter tuning loop.

Source: Alex Kondratiev, Unsplash

What is an experiment?

An experiment is a complete specification of **all degrees of freedom** so that the training of a model is **reproducible**.

- Specification of the computers,
- Version of the libraries,
- Version of the dataset,
- Version of the training code,
- Model definition,
- Values of the hyperparameters,
- Values of the random seeds,
- ...

mlflow

					Parameters		Metrics		
	Date ▼	User	Source	Version	alpha	lambda	mae	r2	rmse
	2018-08-30 15:42:55	mlflow	R:train.R	da3f0a	1	1	0.638	0.03	0.857
	2018-08-30 15:42:50	mlflow	R:train.R	da3f0a	1	0.5	0.639	0.039	0.853
	2018-08-30 15:42:45	mlflow	R:train.R	da3f0a	1	0.2	0.617	0.153	0.804
	2018-08-30 15:42:40	mlflow	R:train.R	da3f0a	1	0	0.597	0.224	0.77
	2018-08-30 15:42:35	mlflow	R:train.R	da3f0a	0.5	1	0.639	0.039	0.853
	2018-08-30 15:42:30	mlflow	R:train.R	da3f0a	0.5	0.5	0.621	0.125	0.818
	2018-08-30 15:42:26	mlflow	R:train.R	da3f0a	0.5	0.2	0.616	0.169	0.794
	2018-08-30 15:42:21	mlflow	R:train.R	da3f0a	0.5	0	0.597	0.224	0.77
	2018-08-30 15:42:15	mlflow	R:train.R	da3f0a	0	1	0.617	0.158	0.801
	2018-08-30 15:42:09	mlflow	R:train.R	da3f0a	0	0.5	0.617	0.171	0.793
	2018-08-30 15:42:04	mlflow	R:train.R	da3f0a	0	0.2	0.618	0.178	0.788
	2018-08-30 15:41:50	mlflow	R:train.R	da3f0a	0	0	0.597	0.224	0.77

Source: https://www.mlflow.org/

Why do we need an experimental protocol?

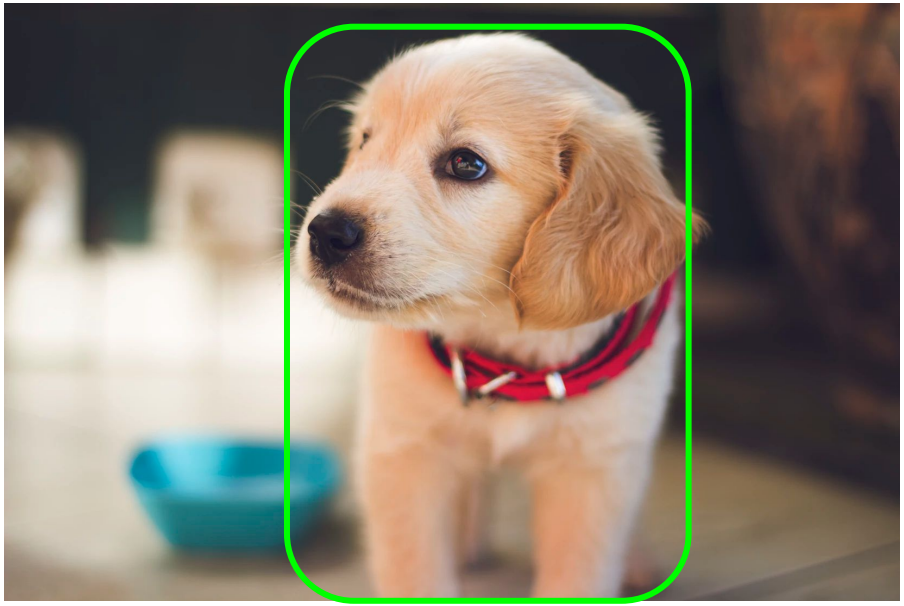
There are different ways to achieve a given goal, but many are considered “cheating” according to human.

For an AI, all means are good!

Source: Garmulewicz, Michał, Henryk Michalewski, and Piotr Miłoś. "Expert-augmented actor-critic for vizdoom and montezumas revenge." arXiv preprint arXiv:1809.03447 (2018).

Right for the wrong reasons

Is it a dog? Yes because...



Source: Berkay Gumustekin, Unsplash

Is it a dog? Yes because...

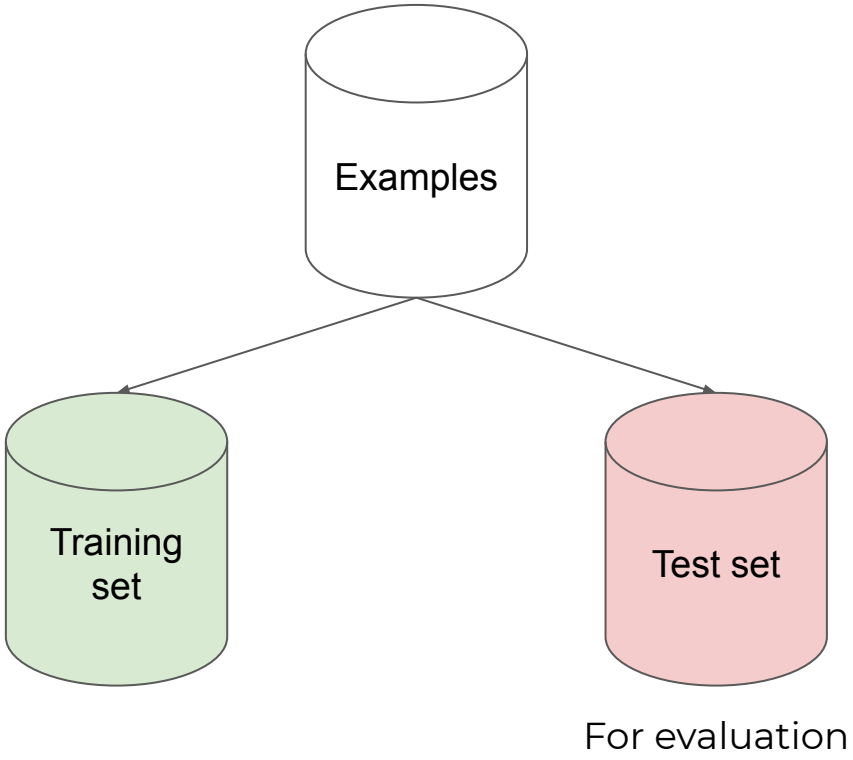
Is it a dog? Yes because...

Source: Berkay Gumustekin, Unsplash

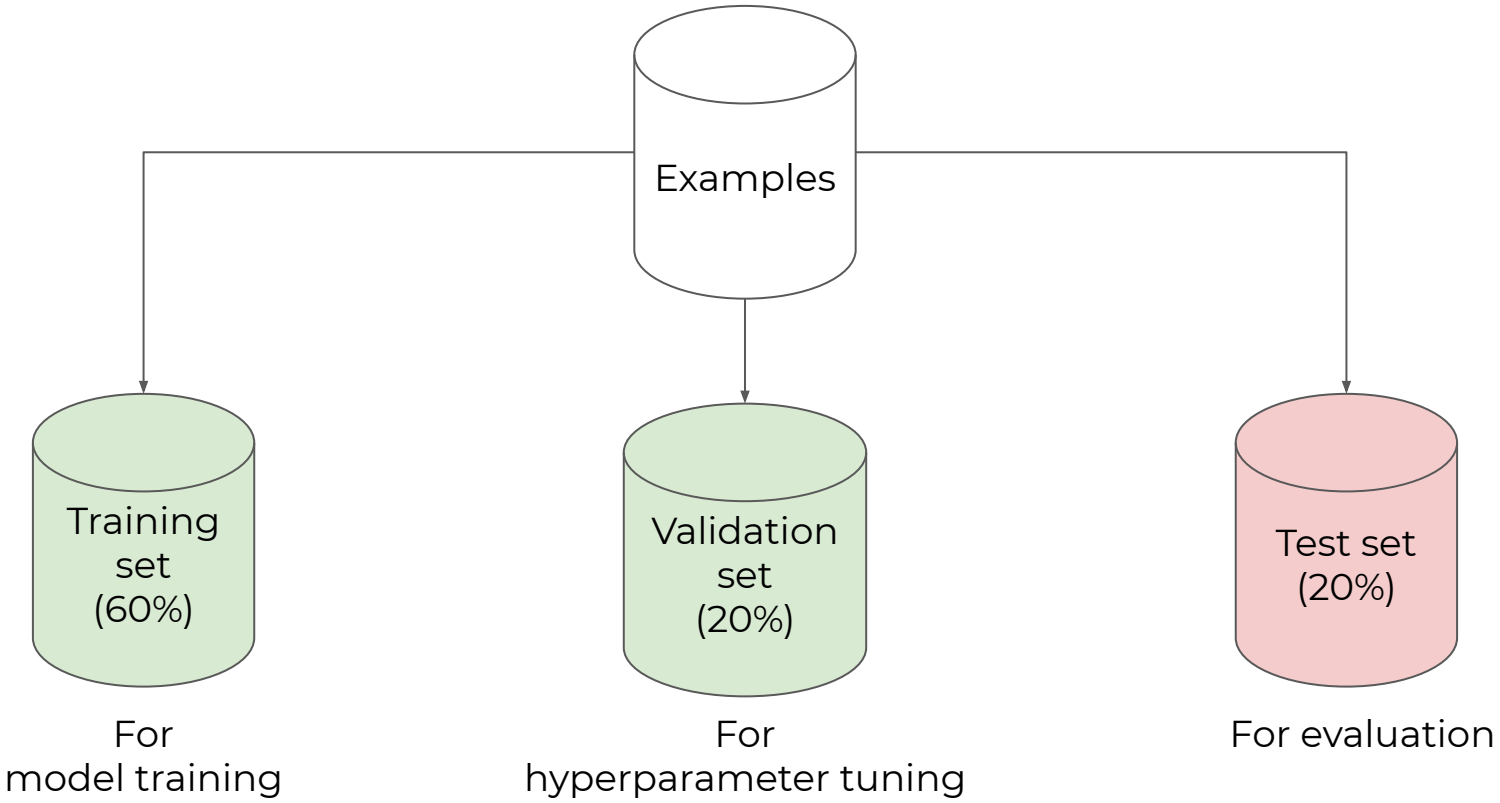
Saliency map for interpretation Source: Simonyan, Karen, Andrea Vedaldi, and Andrew Zisserman. "Deep inside convolutional networks: Visualising image classification models and saliency maps." (2013).

Experimental protocol

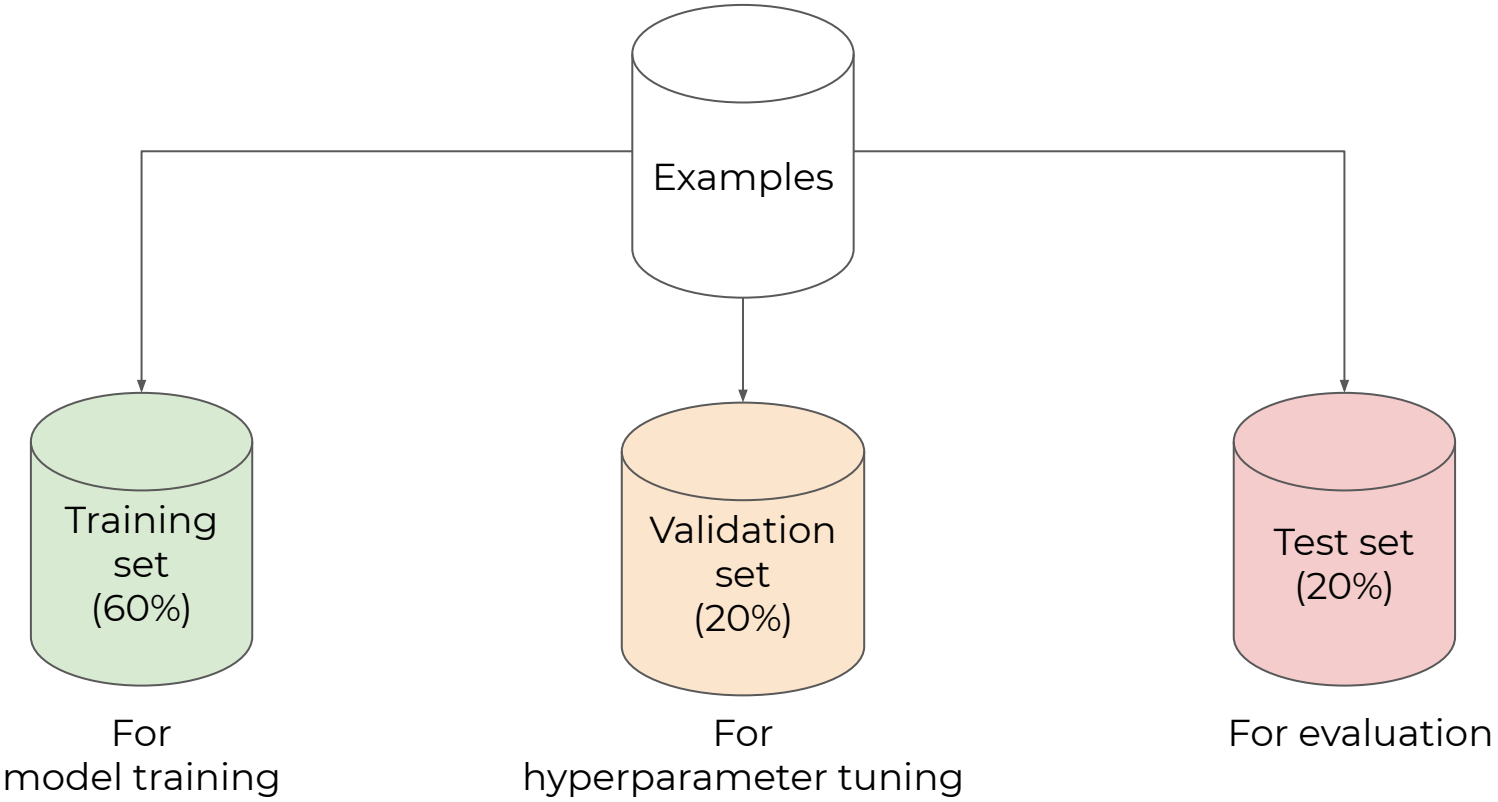
Never use a training example for evaluating an algorithm; otherwise the performance metric will be higher than in production.



How to choose the best hyperparameters?

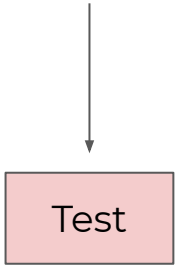


How to choose the best hyperparameters?



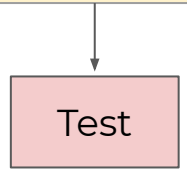
Cross-validation

Split	Exp. A	Exp. B	Exp. C
Train | Valid	Performance A1	Performance B1	Performance C1



K-fold cross-validation

Splits	Exp. A	Exp. B	Exp. C
	Performance A_1	Performance B_1	Performance C_1
	Performance A_2	Performance B_2	Performance C_2
	Performance A_K	Performance B_K	Performance C_K
	Average perf. A StdError perf. A	Average perf. B StdError perf. B	Average perf. C StdError perf. C



Deployment & engineering

How to detect and deal with prediction errors?

Other constraints must be respected:

- Latency
- Memory usage
- Computational power

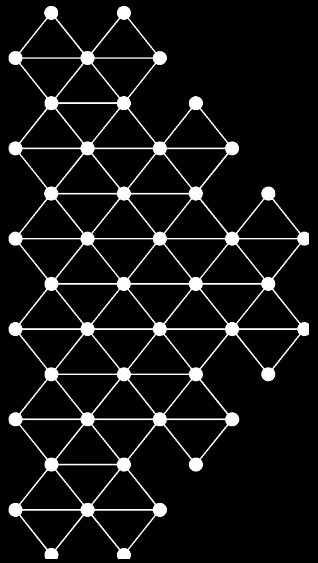
Source: Margaux Olverd, Unsplash

How the new algorithm interacts with other components?

- If too complicated to analyze, A/B testing can be an alternative.

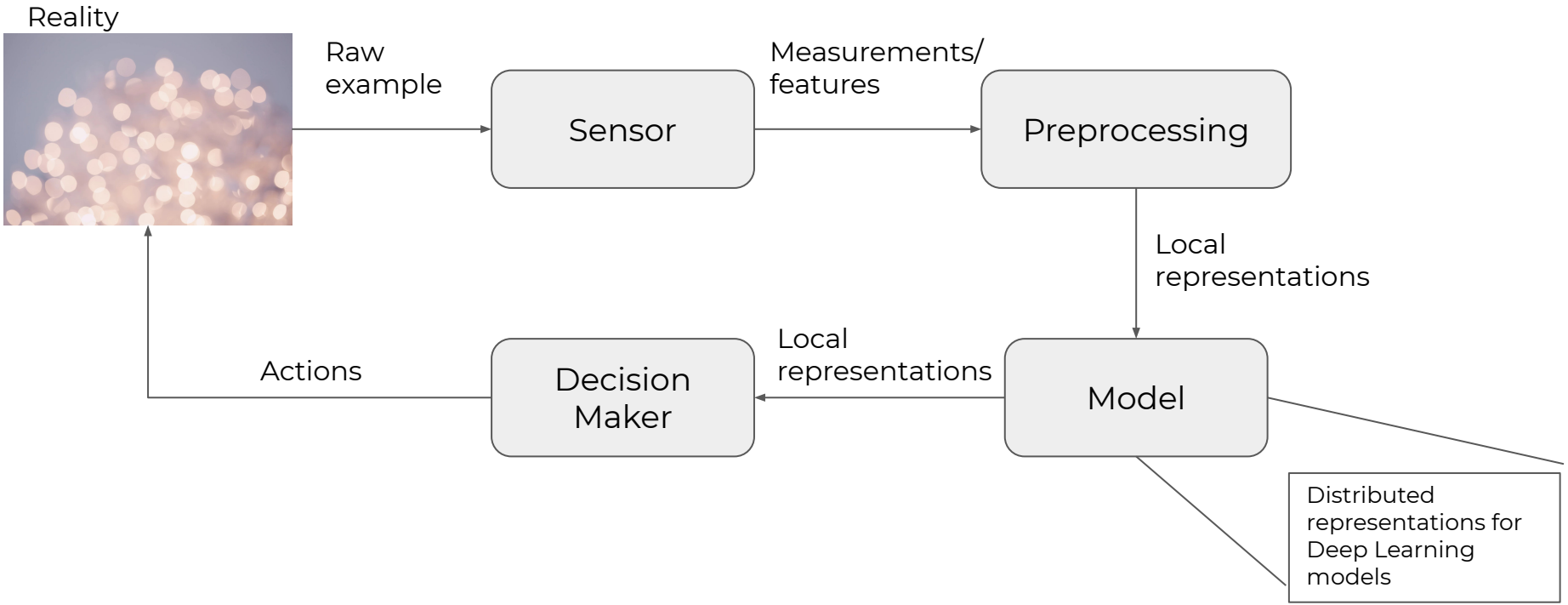
Quebec
Artificial
Intelligence
Institute

Mila

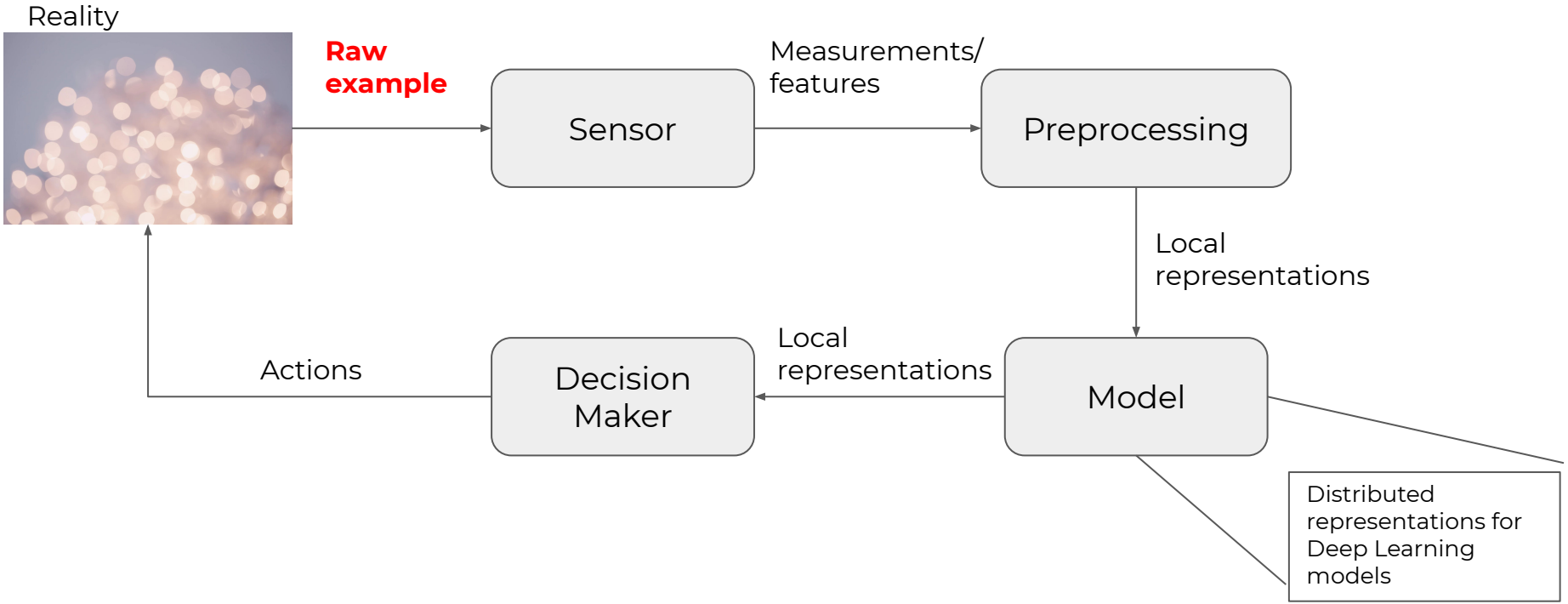


What is data?

The global picture



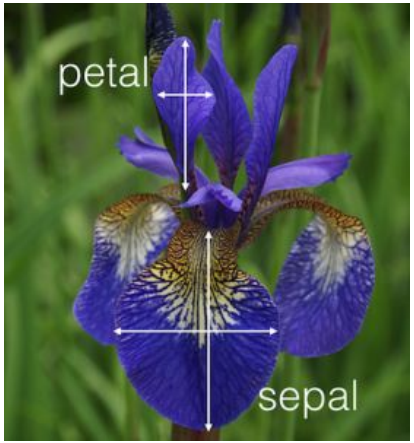
The global picture



The concept of *example*

Example: a set of data that is *self-contained*, *independent*, and *identically distributed*.

- **Self-contained**: all relevant features are present in the example.
- **Independent**: the generation of an example does not impact other examples.
- **Identically distributed**: the probability of seeing an example is fixed.

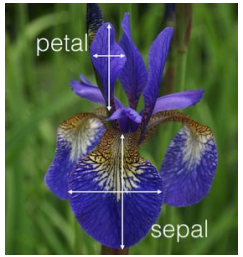


Source: Ronald Fisher, “The use of multiple measurements in taxonomic problems” (1936)

The concept of *example*

In supervised learning, we divide an example in 2 categories of variables:

$$z = (x, y)$$

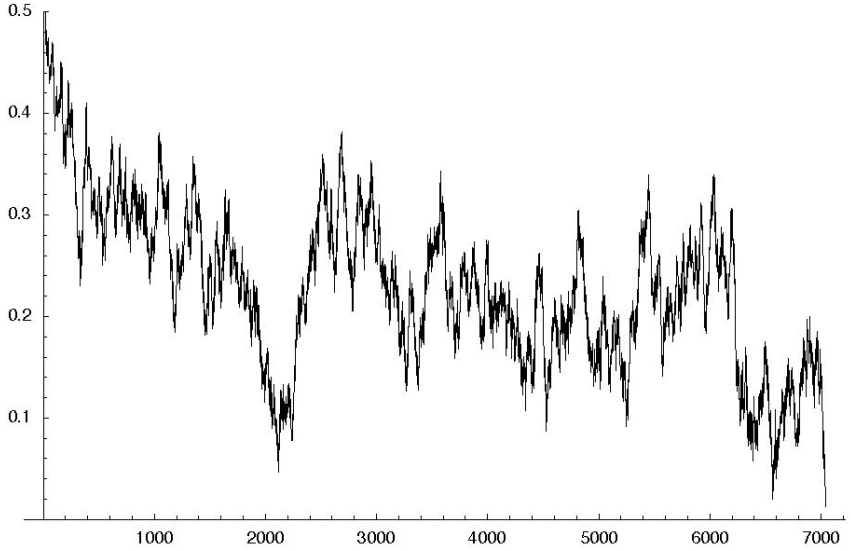


Input features

`Iris versicolor`

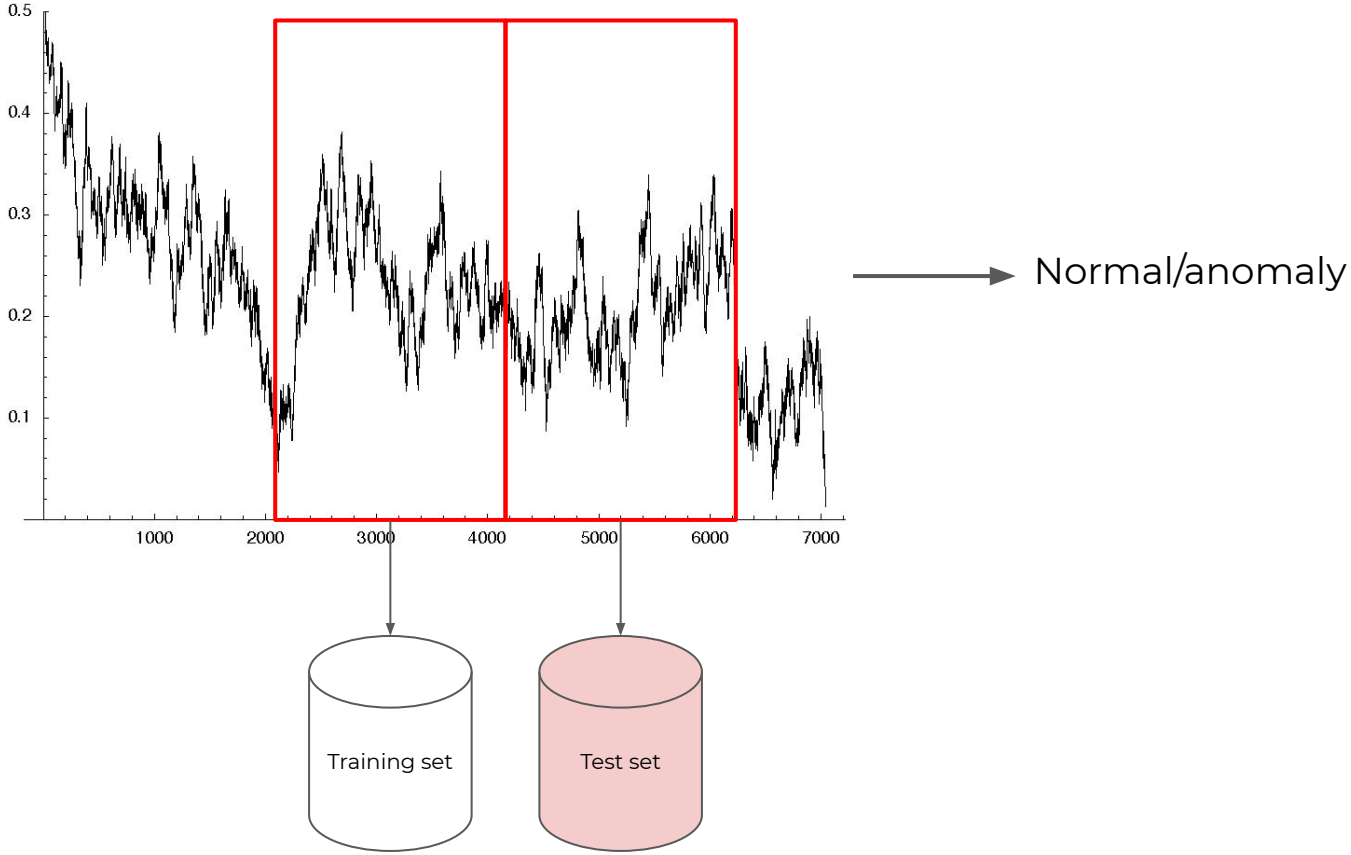
Output targets

Pitfall of independence

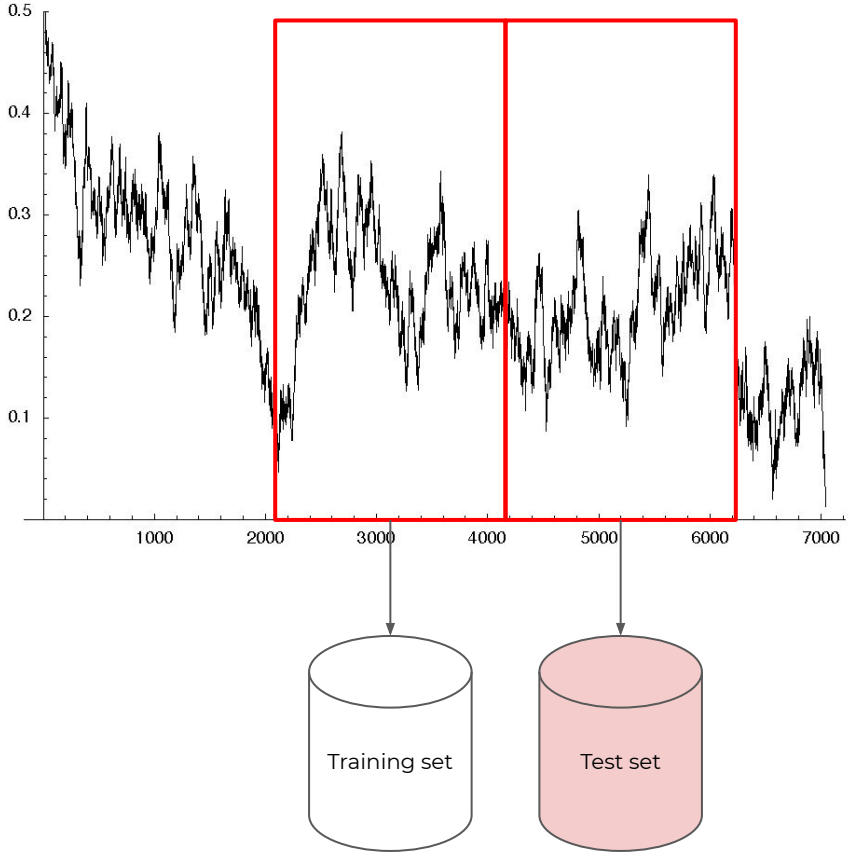


→ Normal/anomaly

Pitfall of independence

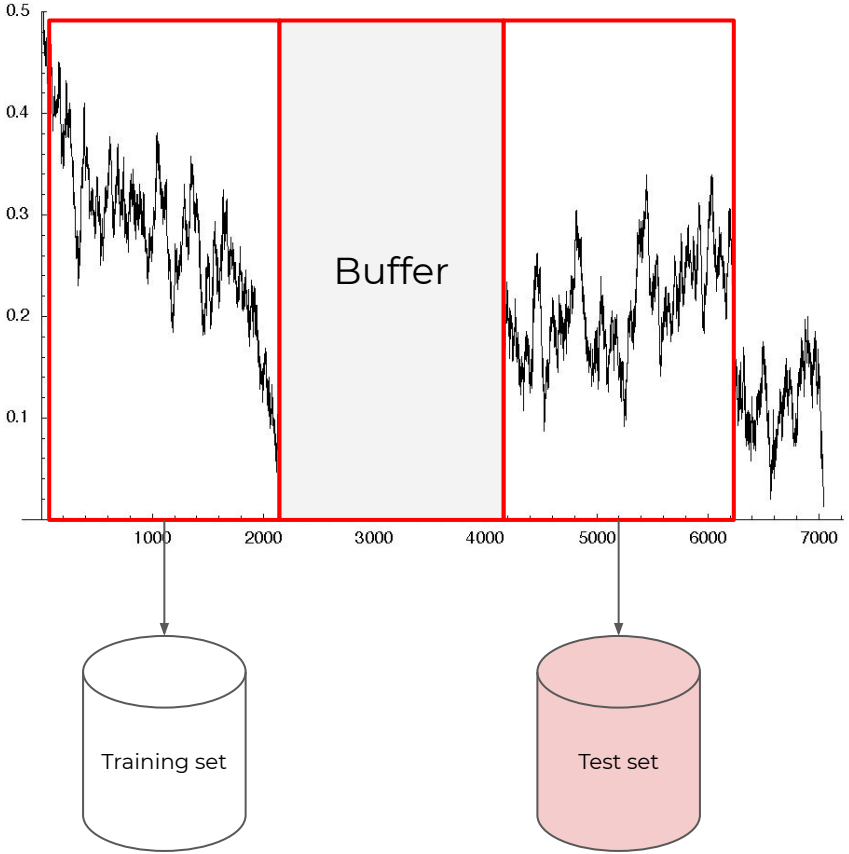


Pitfall of independence



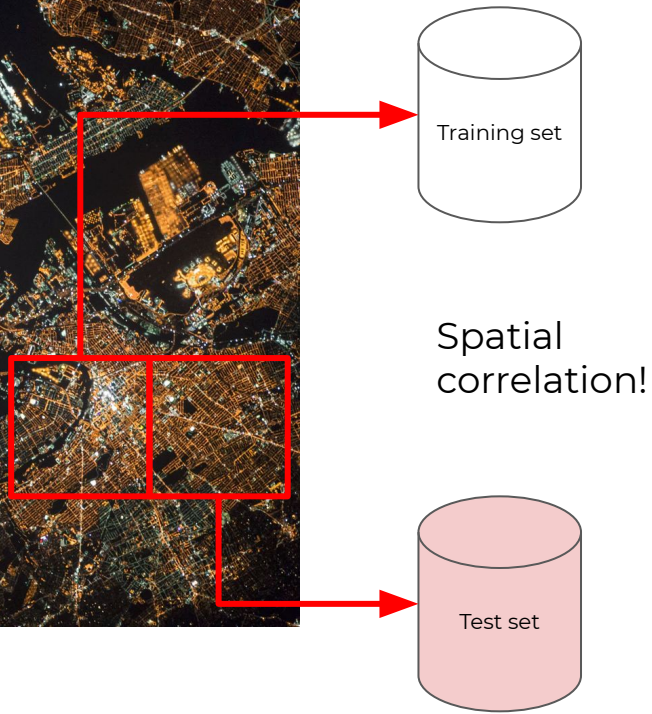
Temporal correlation!
Not independent.

Pitfall of independence

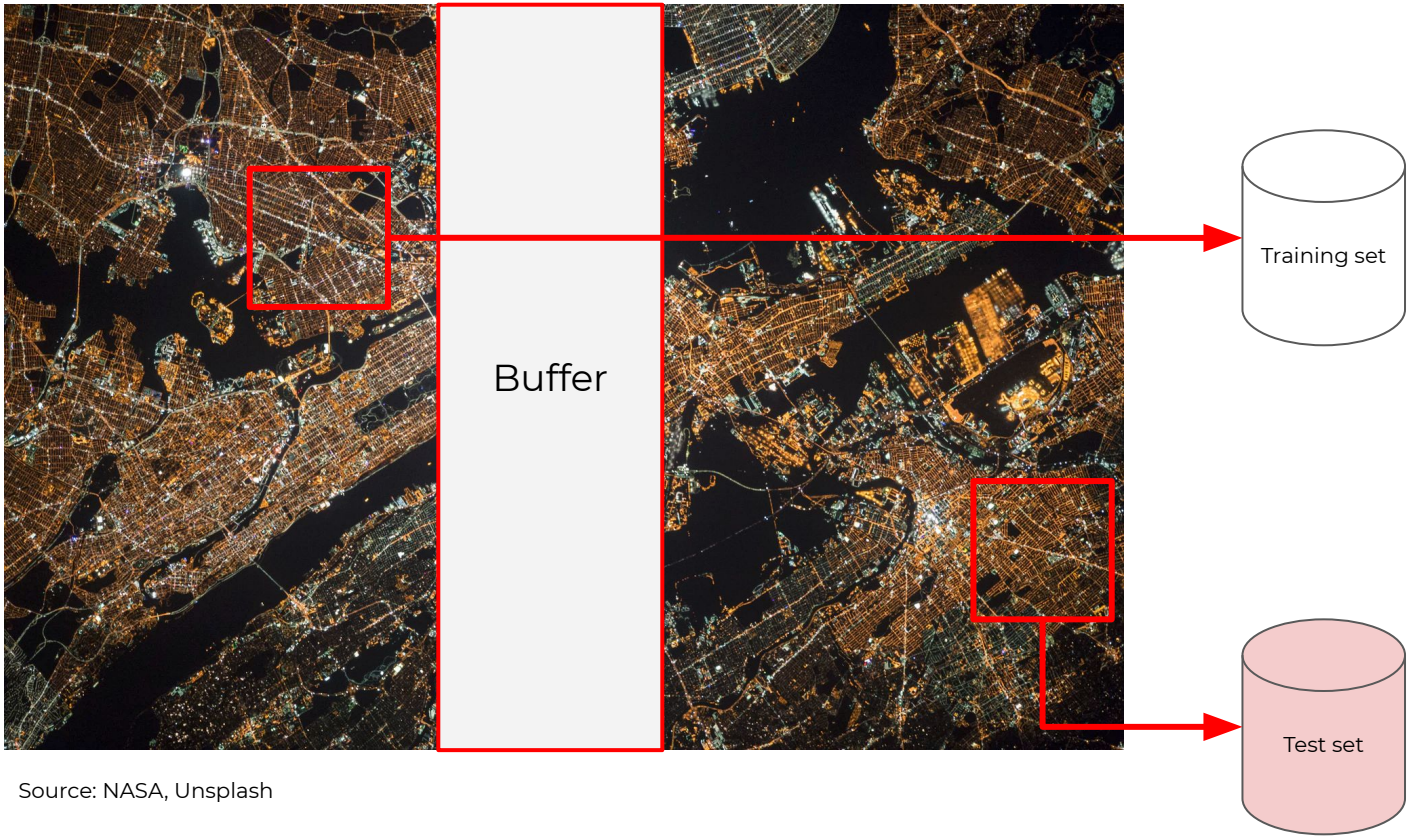


Pitfall of independence

Source: NASA, Unsplash



Pitfall of independence



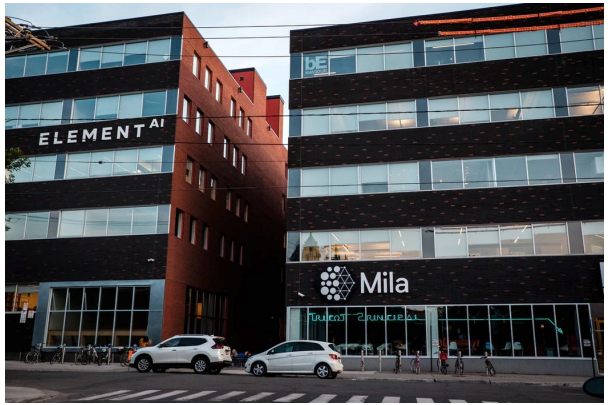
Source: NASA, Unsplash

Pitfall of identically distributed

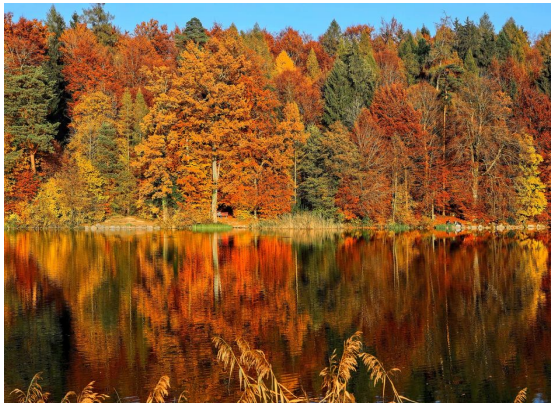
Non-stationary data generation process

$$P(x_1) \neq \cdots \neq P(x_6) \neq \cdots \neq P(x_{10})$$

Source: Joy Real, Unsplash

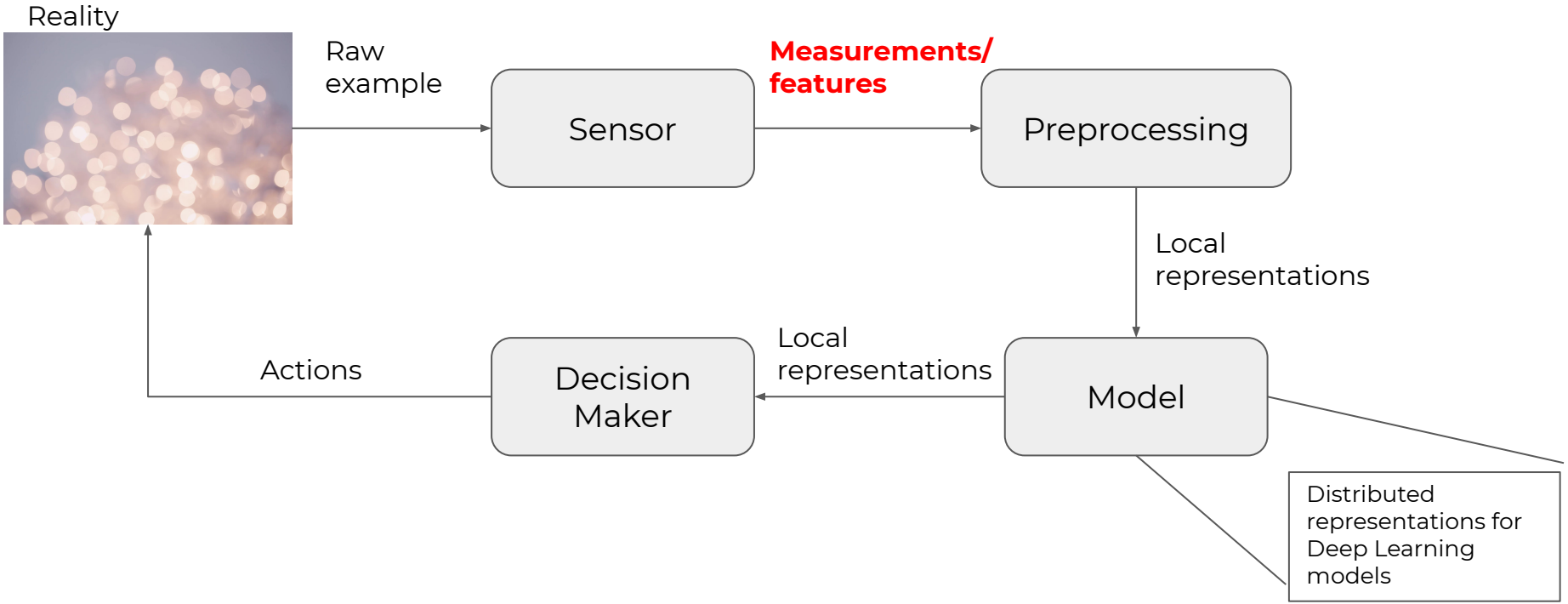


Source:Morgan Petroski, Unsplash



Source: Ricardo Gomez Angel, Unsplash

The global picture



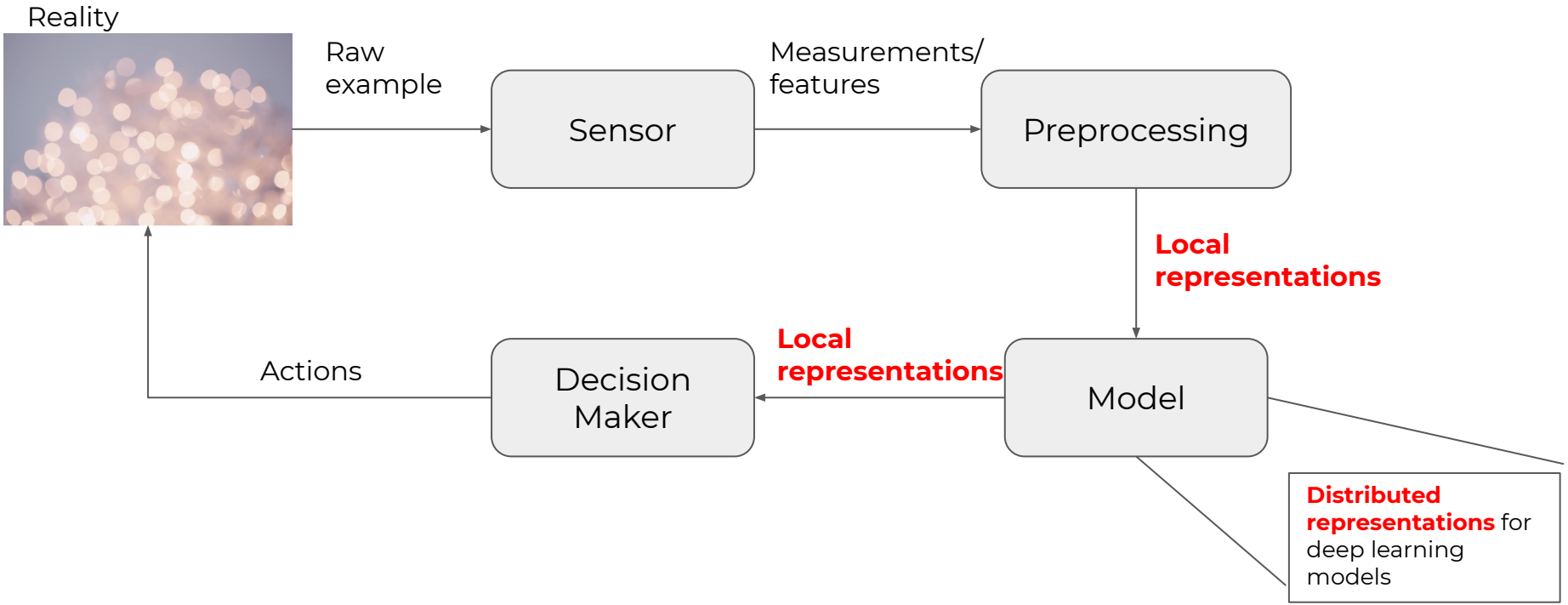
Specification of structured data

Structured data: adding **topology** to statistical data types

- **Text**: Number of sequences, maximum length, term frequencies, encoding (e.g. UTF-8).
- **Image**: Number of images, width, height, channel, encoding (e.g. PNG)
- **Video**: Number of video, width, height, channel, audio, subtitles, codec (e.g. MP4)
- **Graph**: Number of nodes and edges, features per node, feature per edges, adjacency matrix.
- **Dictionary**: Number of pairs (key, value), JSON structure

...

The global picture



The concept of *representation*

Representation: an implementation of the medium supporting data processing.

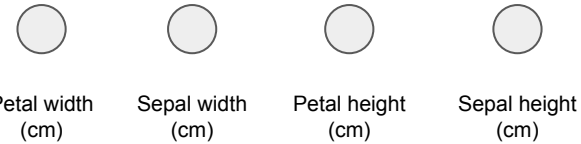
René Magritte, The Treachery of Images (1929)

The concept of *representation*

Representation: an implementation of the medium supporting data processing.

Local representation: one processing unit per concept. Easy to interpret. The inputs and outputs of a model are local representations.

René Magritte, The Treachery of Images (1929)



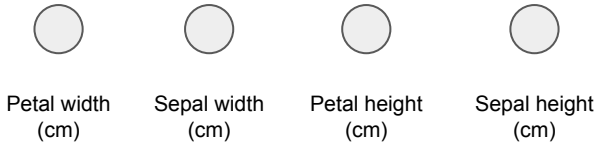
The concept of *representation*

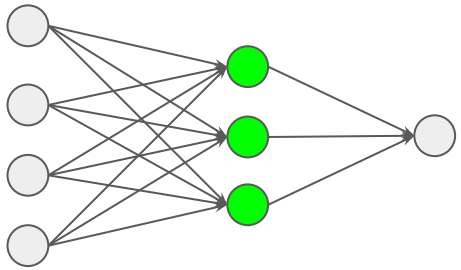
Representation: an implementation of the medium supporting data processing.

Local representation: one processing unit per concept. Easy to interpret. The inputs and outputs of a model are local representations.

Distributed representation: many processing units per concept and many concepts per processing units. Hard to interpret but efficient.

René Magritte, The Treachery of Images (1929)



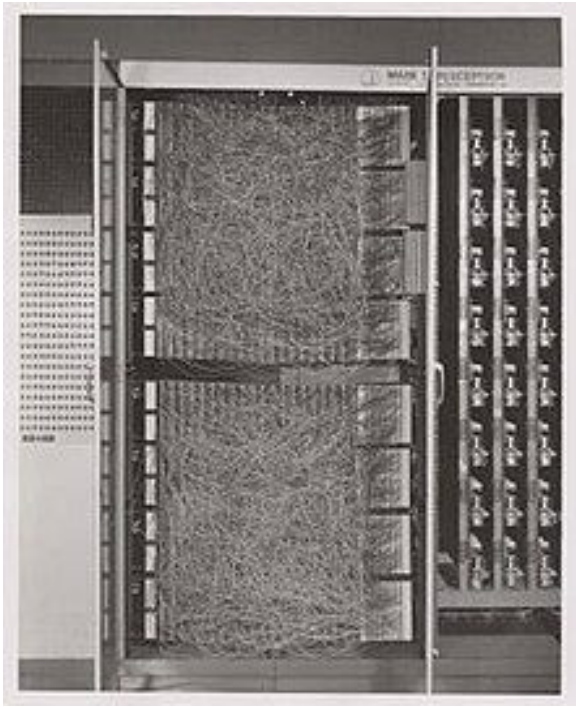


The concept of *representation*

Representation: an implementation of the medium supporting data processing.

Local representation: one processing unit per concept. Easy to interpret. The inputs and outputs of a model are local representations.

Distributed representation: many processing units per concept and many concepts per processing units. Hard to interpret but efficient.



Mark 1 Perceptron (1958)
Source: Wikipedia

Text representations

We define a **dictionary**: a finite set of categorical variables representing words.

Dictionary

{UNK: 0, “.”: 1, “we”: 2, “winter”: 3, “together”: 4, “live”: 5, “like”: 6, “in”: 7}

Corpus and tokenization

“We live in Montreal.” -> [“we”, “live”, “in”, UNK, “.”] ->[2, 5, 7, 0, 1]

“We live together.” -> [“we”, “live”, “together”, “.”] ->[2, 5, 4, 1]

“We like winter.” -> [“we”, “like”, “winter”, “.”] ->[2, 6, 3, 1]

Local representation: one-hot encoding

For nominal measurements (e.g. categorical variable).

The total number of elements must be known.

Example:

{UNK: 0, “.”: 1, “we”: 2, “winter”: 3, “together”: 4, “live”: 5, “like”: 6, “in”: 7}

“We live in Montreal.” -> [“we”, “live”, “in”, UNK, “.”] ->[2, 5, 7, 0, 1] ->

[[0 0 1 0 0 0 0 0], [0 0 0 0 0 1 0 0], [0 0 0 0 0 0 0 1], [0 0 0 0 0 0 0 0],

[0 1 0 0 0 0 0 0]]

Normalization

Min-max feature scaling:

$$x' = \frac{x - x_{min}}{x_{max} - x_{min}}$$

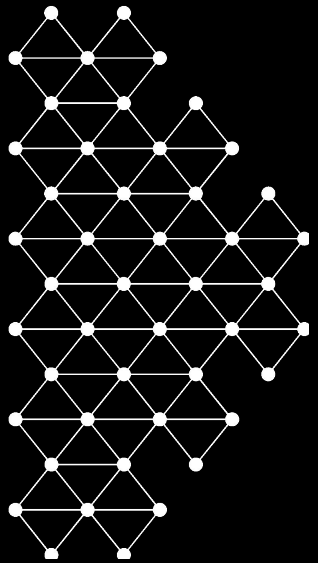
Standardization:

$$x' = \frac{x - \bar{x}}{\sigma_x}$$

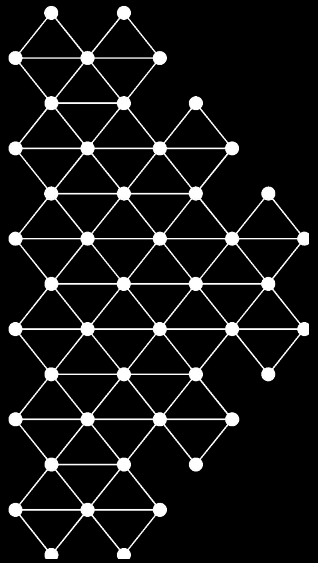
Take-home message

- Machine learning is a subfield of artificial intelligence.
- A methodology exists to prevent the learning algorithm from cheating.
- Using an experimental protocol with cross-validation is essential.
- Data should be independent and identically distributed.

Quebec
Artificial
Intelligence
Institute

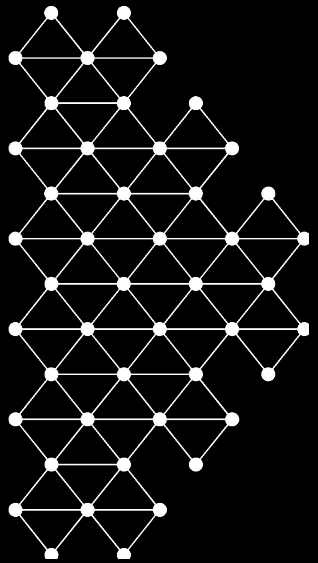


Quebec
Artificial
Intelligence
Institute



Appendix

Quebec
Artificial
Intelligence
Institute

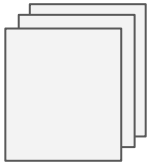


Introduction to probability

> “Probability theory is nothing but common sense reduced to calculation”.
>
> - Laplace (1819)

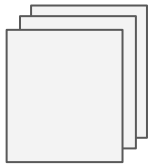
Data distribution

Suppose we have a finite set of examples $S = \{x_1, \ldots, x_N\}$

where $x_i =$  . With S, can we predict x_{N+1}?

In general, we cannot predict **the exact value** because examples are **independent** (e.g., rolling a dice).

Can we guess the probability of each possible value? (e.g., is the dice fair?)

$$P(x_{N+1} = \quad)$$


Example of probability

Suppose that all x_i are categorical features: $x_i \in \{1, \ldots, K\}$

The probability that x_i takes the value k : $P(x_i = k)$

Properties for all i :

- $\forall k,\ 0 \leq P(x_i = k) \leq 1$
- $\sum_{k=1}^{K} P(x_i = k) = 1$

Identically distributed

Several random variables $S = \{x_1, \ldots, x_N\}$ are identically distributed if, for each outcome k, they have the same probability:

$$\forall k,\ P(x_1 = k) = \cdots = P(x_N = k)$$

Joint probability

We can model a probability over the whole dataset $S = \{x_1, \ldots, x_N\}$

with the following properties:

1. $\forall k_1, \ldots, k_N,\ 0 \leq P(x_1 = k_1, \ldots, x_N = k_N) \leq 1$

2. $\sum_{k_1=1}^{N} \cdots \sum_{k_N=1}^{N} P(x_1 = k_1, \ldots, x_N = k_N) = 1$

Marginalization (Sum rule)

A marginal probability concerns only one variable: $P(x_i = k)$

The joint probability is linked to marginal probabilities by the sum rule:

$$\forall k_1, \ldots, k_{N-1}, \sum_{k_N=1}^{N} P(x_1 = k_1, \ldots, x_N = k_N)$$
$$= P(x_1 = k_1, \ldots, x_{N-1} = k_{N-1})$$

$$\forall k_1, \sum_{k_2=2}^{N} \cdots \sum_{k_N=1}^{N} P(x_1 = k_1, \ldots, x_N = k_N) = P(x_1 = k1)$$

Probability as frequencies

Suppose I have two coins A, B, each with outcomes {0, 1} (binary variables).

Each trial is an example $x_i = (a_i, b_i)$. We can count the number of outcomes.

	A=0	A=1	Total
B=0	24	8	32
B=1	36	12	48
Total	60	20	80

Probability as frequencies

Suppose we have two coins A, B, each with outcomes {0, 1} (binary variables). Each trial is an example $x_i = (a_i, b_i)$. We can count the number of outcomes and normalize (divide by the total number of trials).

	A=0	A=1	Total
B=0	3/10	1/10	2/5
B=1	9/20	3/20	3/5
Total	3/4	1/4	1

Probability as frequencies

We are estimating $P(A, B)$, but also $P(A)$ and $P(B)$.

	A=0	A=1	Total
B=0	3/10	1/10	2/5
B=1	9/20	3/20	3/5
Total	3/4	1/4	1

Mutual independence (Product rule)

Mutual independence is an assumption:

$$\forall k_1, \ldots, k_N, \; P(x_1 = k_1, \ldots, x_N = k_N) = \prod_{i=1}^{N} P(x_i = k_i)$$

Mutual independence (Product rule)

Mutual independence is an assumption that constraint the number of free parameters:

$$\forall k_1, \ldots, k_N, \underbrace{P(x_1 = k_1, \ldots, x_N = k_N)}_{K^N - 1 \text{ parameters}} = \underbrace{\prod_{i=1}^{N} P(x_i = k_i)}_{N(K-1) \text{ parameters}}$$

Probability as frequencies

Is A and B independent?

	A=0	A=1	Total
B=0	3/10	1/10	2/5
B=1	9/20	3/20	3/5
Total	3/4	1/4	1

Conditional probability

Definition:

$$\forall k_1, \ldots, k_N, P(x_1 = k_1, \ldots, x_i = k_i | x_{i+1} = k_{i+1}, \ldots, x_N = k_N)$$
$$= \frac{P(x_1 = k_1, \ldots, x_N = k_N)}{P(x_{i+1} = k_{i+1}, \ldots, x_N = k_N)}$$

Conditional probability

Definition:

given these variables equal these values

$$\forall k_1, \ldots, k_N, P(x_1 = k_1, \ldots, x_i = k_i | x_{i+1} = k_{i+1}, \ldots, x_N = k_N)$$

$$= \frac{P(x_1 = k_1, \ldots, x_N = k_N)}{P(x_{i+1} = k_{i+1}, \ldots, x_N = k_N)}$$

Renormalize w.r.t. what is known

Property:

$$\sum_{k_1=1}^{K} \cdots \sum_{k_i=1}^{K} P(x_1 = k_1, \ldots, x_i = k_i | x_{i+1} = k_{i+1}, \ldots, x_N = k_N) = 1$$

Probability as frequencies

Suppose $A = 0$, what is $P(B = 0|A = 0)$?

	A=0	A=1	Total
B=0	3/10	1/10	2/5
B=1	9/20	3/20	3/5
Total	3/4	1/4	1

Probability as frequencies

Suppose $A = 0$, what is $P(B = 0|A = 0)$?

	A=0
B=0	2/5
B=1	3/5
Total	1

Conditional probability and independence

Definition:

given these variables equal these values

$$\forall k_1, \ldots, k_N,\ P(x_1 = k_1, \ldots, x_i = k_i | x_{i+1} = k_{i+1}, \ldots, x_N = k_N)$$

$$= \frac{P(x_1 = k_1, \ldots, x_N = k_N)}{P(x_{i+1} = k_{i+1}, \ldots, x_N = k_N)}$$

$$= \frac{\prod_{j=1}^{N} P(x_j = k_j)}{\prod_{j=i+1}^{N} P(x_j = k_j)}$$

$$= \prod_{j=1}^{i} P(x_j = k_j)$$

Verify that $P(B|A) = P(B)$ in the previous example.

Independent and identically distributed

Definition of identically distributed:

$$\forall k_1, \ldots, k_N,\ P(x_1 = k_1, \ldots, x_N = k_N) = \prod_{j=1}^{N} P(x_j = k_j)$$

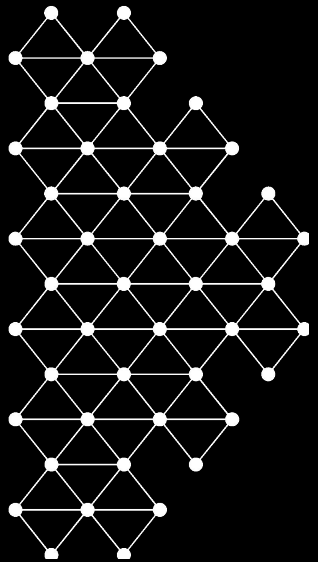
$$= \underbrace{\prod_{j=1}^{N} P(x_1 = k_j)}_{K \text{ parameters}}$$

This is the standard framework for probabilities as frequencies.

See *the law of large numbers*.

Source:Jordan Rowland

Quebec
Artificial
Intelligence
Institute



Probabilistic modeling

> “All models are wrong, but some are useful”.
>
> — Georges Box (1976)

Categorical distribution

Suppose that all x_i are categorical features: $x_i \in \{1, \ldots, K\}$

A categorical distribution is defined as a set of parameters $\{p_1, \ldots, p_K\}$ where $\forall i,\ 0 \leq p_i \leq 1,\ \sum_{i=1}^{K} p_i = 1$.

It is an **independent and identically distributed (iid)** model.

$\{p_1, \ldots, p_6\}$

```
In [2]: np.random.choice(a=6, size=20, p=[1/6, 1/6, 1/6, 1/6, 1/6, 1/6])+1
Out[2]: array([1, 6, 4, 6, 2, 6, 3, 3, 1, 4, 6, 2, 2, 3, 4, 2, 5, 3, 6, 1])
```

S

$K = 6$

Probabilistic modeling

In reality, we do not know the data generation mechanism; we only have the raw data:

```
Out[2]: array([1, 6, 4, 6, 2, 6, 3, 3, 1, 4, 6, 2, 2, 3, 4, 2, 5, 3, 6, 1])
```

By definition of an example, we use the following assumptions:

1. Independence,
2. Identically distributed.

Probabilistic modeling

Consequences:

1. the previous outcomes do not change the probability of the next outcome:

$$P(x_{N+1}|x_1, \ldots, x_N) = P(x_{N+1})$$

2. Each possible outcome are sampled identically from a categorical distribution with parameters $\{p_1, \ldots, p_K\}$

3. How can we infer the parameters from S ?

Maximum likelihood

Find $\{p_1, \ldots, p_K\}$ that maximizes $P(S) = P(x_1, \ldots, x_N)$

We obtain that:

$$p_k^* = \frac{N_k}{N}$$

Maximum likelihood estimator

Fraction of time the value k was observed.

Consequently, we can summarize an arbitrary large number of observations N with K parameters.

Limitation in high-dimensional space

- Suppose that an example x_i is a vector of n categorical variables.
- Also, suppose that $x_{ij} \in \{1, \ldots, K\}$.
- Thus, we have K^n possible vectors.
- By mapping each vector to a unique integer, we have $x_i \in \{1, \ldots, K^n\}$
- Example: a 28x28 binary image can take one of the $2^{(28 \times 28)} \approx 10^{236}$ possible images.

Limitation in high-dimensional space

In high-dimensional space, if we assign one parameter per possible data point and we use maximum likelihood as before: $p_k^* = \frac{N_k}{N}$

- Most of the parameters are zero
- Non-zero parameters are equal to $\frac{1}{N}$

Later, we will see parametric models where **each parameter depends on the entire dataset**. This idea is fundamental in learning representations.

Anomaly and outlier

We have estimated $\{p_1, \ldots, p_K\}$ from S and we observe $x_{N+1} = k$.

If $p_k \approx 0$ then it is a **rare event**.

We quantify information with $-\ln p_k$.

Rare event contains a lot of information, while events occurring with certainty contain no information at all.

How to interpret rare events depends on the application:

- Anomaly, outlier, measurement error,
- Novelty, ...

Missing value

If we decide that the new observation is a measurement error, we should replace it with another realistic value. To do so, we can sample from our probabilistic model:

$$x_{N+1} \sim \text{Categorical}(p_1, \ldots, p_K)$$

Supervised learning

In supervised learning, an example has the following structure:

$$P(S) = P((x_1, y_1), \ldots, (x_N, y_N))$$

We can assume a model and use maximum likelihood to find the parameters:

$$P(S) = \prod_{i=1}^{N} P(x_i, y_i) = \prod_{i=1}^{N} P_\theta(y_i|x_i)P(x_i)$$

Model parameters

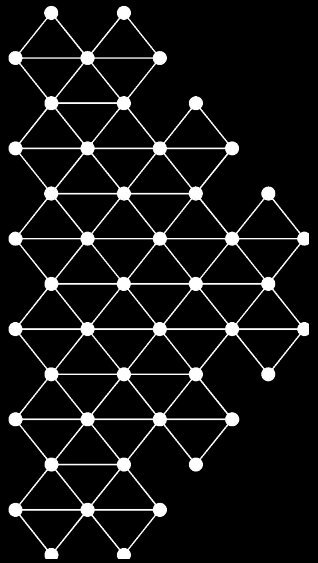
Discriminative model (easy)

Generative model (hard)

Summary

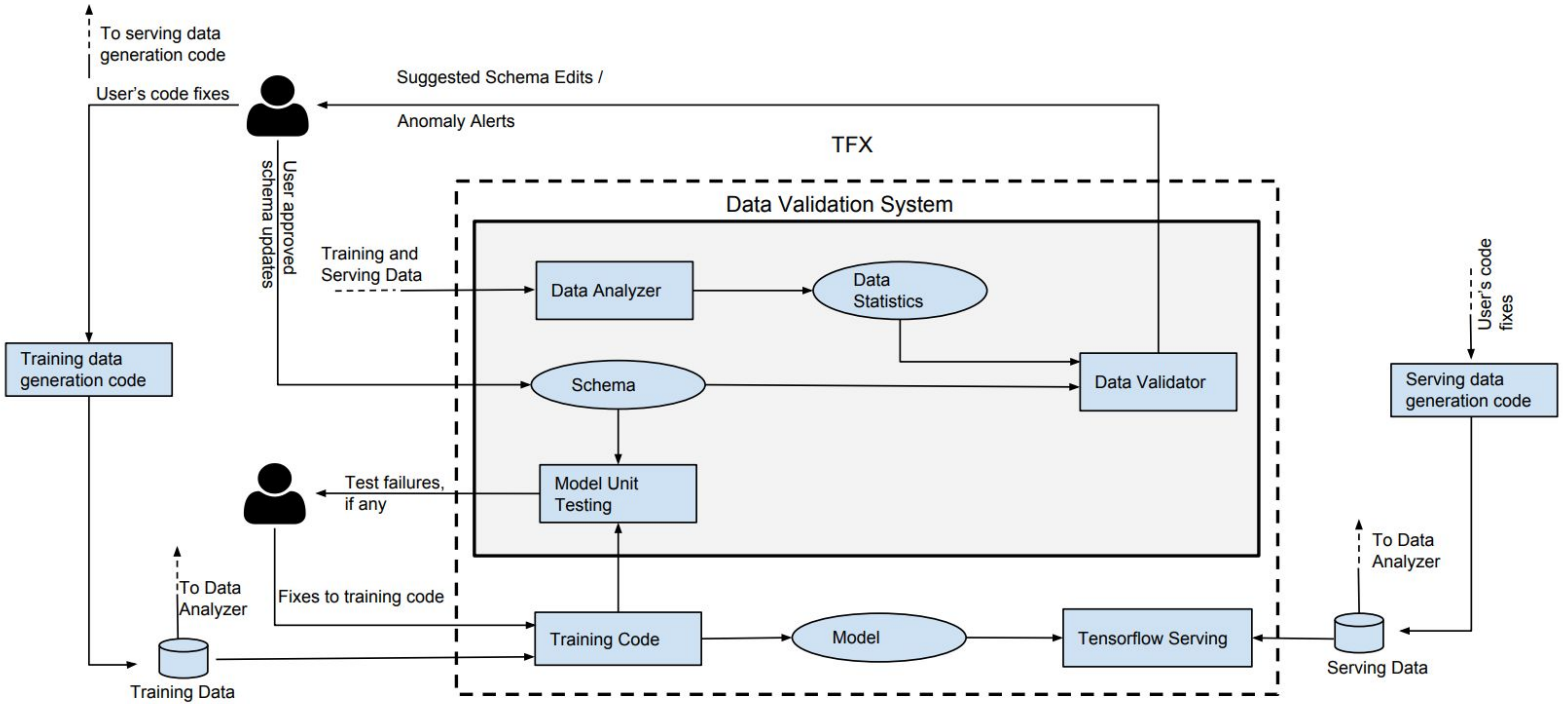
- Probability theory defines the tools to model uncertainty.
- A model implies (false) assumptions on the data.
- A model can summarize many observations with parameters.
- We can find proper parameters with maximum likelihood principle.
- A model can assign probabilities to new observations.
- A generative model can generate new observations.

Quebec
Artificial
Intelligence
Institute



Other topics

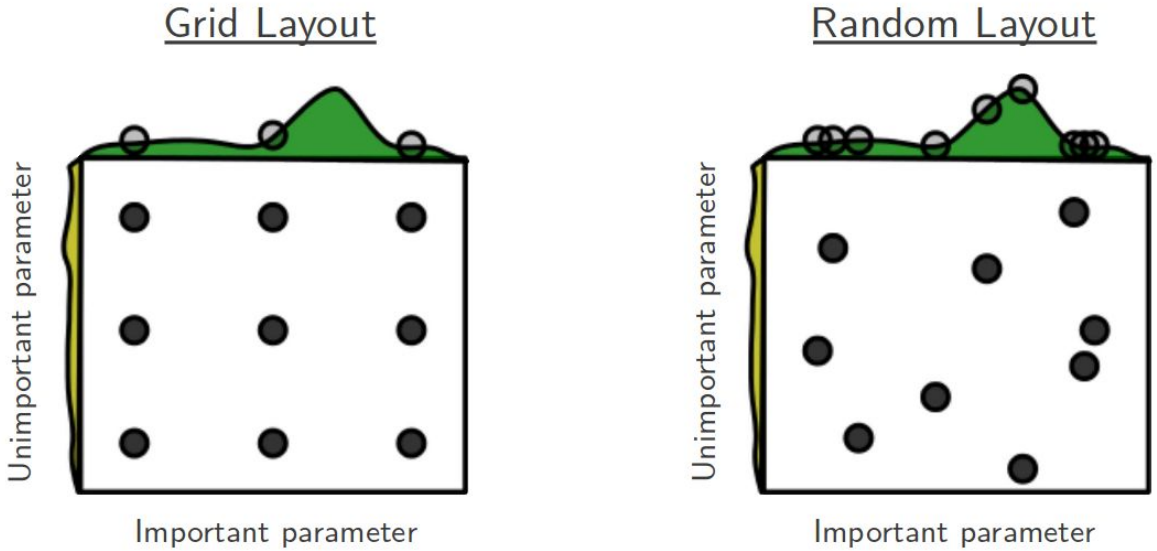
Toward ML engineering: data validation



Source: Data validation for machine learning Breck et al., SysML'19
https://github.com/tensorflow/data-validation

Hyperparameters

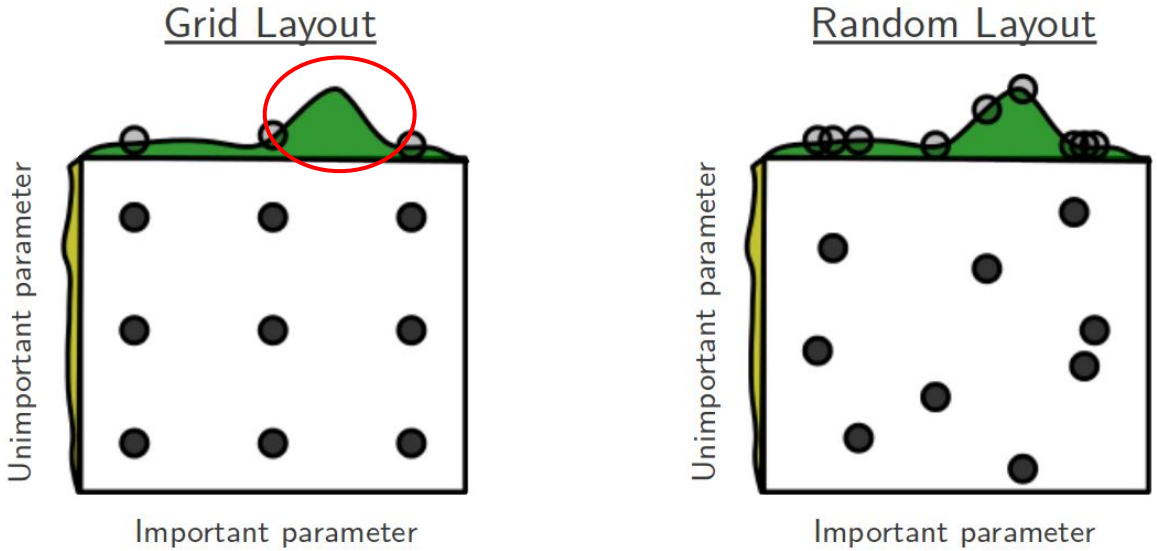
Random search with good candidates is already a strong baseline.



Source:Bergstra, James, and Yoshua Bengio. "Random search for hyper-parameter optimization." Journal of Machine Learning Research 13, no. Feb (2012): 281-305.

Hyperparameters

Random search with good candidates is already a strong baseline.



Source:Bergstra, James, and Yoshua Bengio. "Random search for hyper-parameter optimization." Journal of Machine Learning Research 13, no. Feb (2012): 281-305.

Input pipeline

The input pipeline is a ETL process:

- ***Extract***: read the data from a persistent storage
- ***Transform***: use the CPU to preprocess, do data augmentation and prepare the mini-batches
- ***Load***: transfer the data to an accelerator device such as GPU or TPU

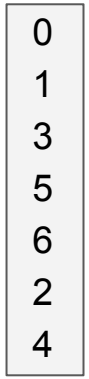
For new datasets, we need to code a *data loader* to perform these tasks.

Source: https://www.tensorflow.org/guide/performance/datasets

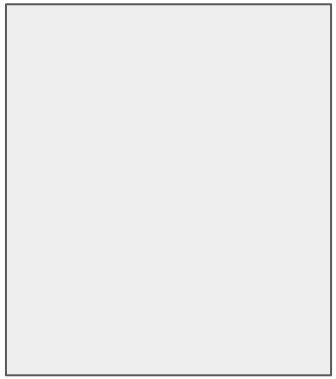
Statistical data type

Statistical data type	Level of measurement	Domain	Example and data type
Binary	Nominal	{0, 1}	“Missing/present” (uint8)
Categorical	Nominal	{1, ..., N}	“City”, “Blood type” (uint32)
Ordinal	Ordinal	Discrete, continuous	“Exam score” (uint8)
Count	Ratio	Nonnegative integer	“Number of persons” (uint8)
Real-valued additive	Interval	Continuous	“Temperature on Celsius scale” (float32)
Real-valued multiplicative	Ratio	Positive real value	“Price” (float32)

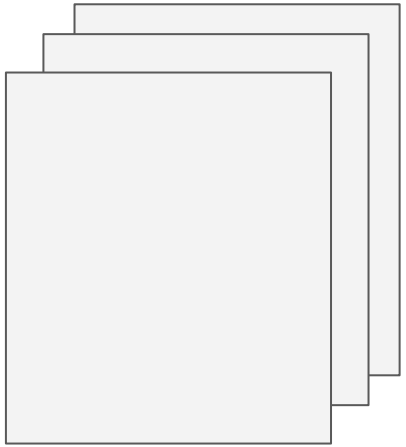
Tensor as a data structure for representations



1D
Vector



2D
Matrix



3D
Tensor

```
In [1]: import numpy as np

In [2]: A = np.random.randn(50000, 27, 27, 3)

In [3]: A.dtype
Out[3]: dtype('float64')

In [4]: A.shape
Out[4]: (50000, 27, 27, 3)

In [5]: A[10, 16, 16, 0]
Out[5]: -1.1260584851068263
```

N-D
Tensor

Convention: first dimension is used for examples.