SINTEF

A BRIEF INTRODUCTION TO ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

Ben and Bjørn Magnus

SINTEF Digital

Bjørn Magnus Mathisen

Senior Researcher
PhD in AI and ML, NTNU
SINTEF Digital
Software Engineering, Safety and Security
Trondheim

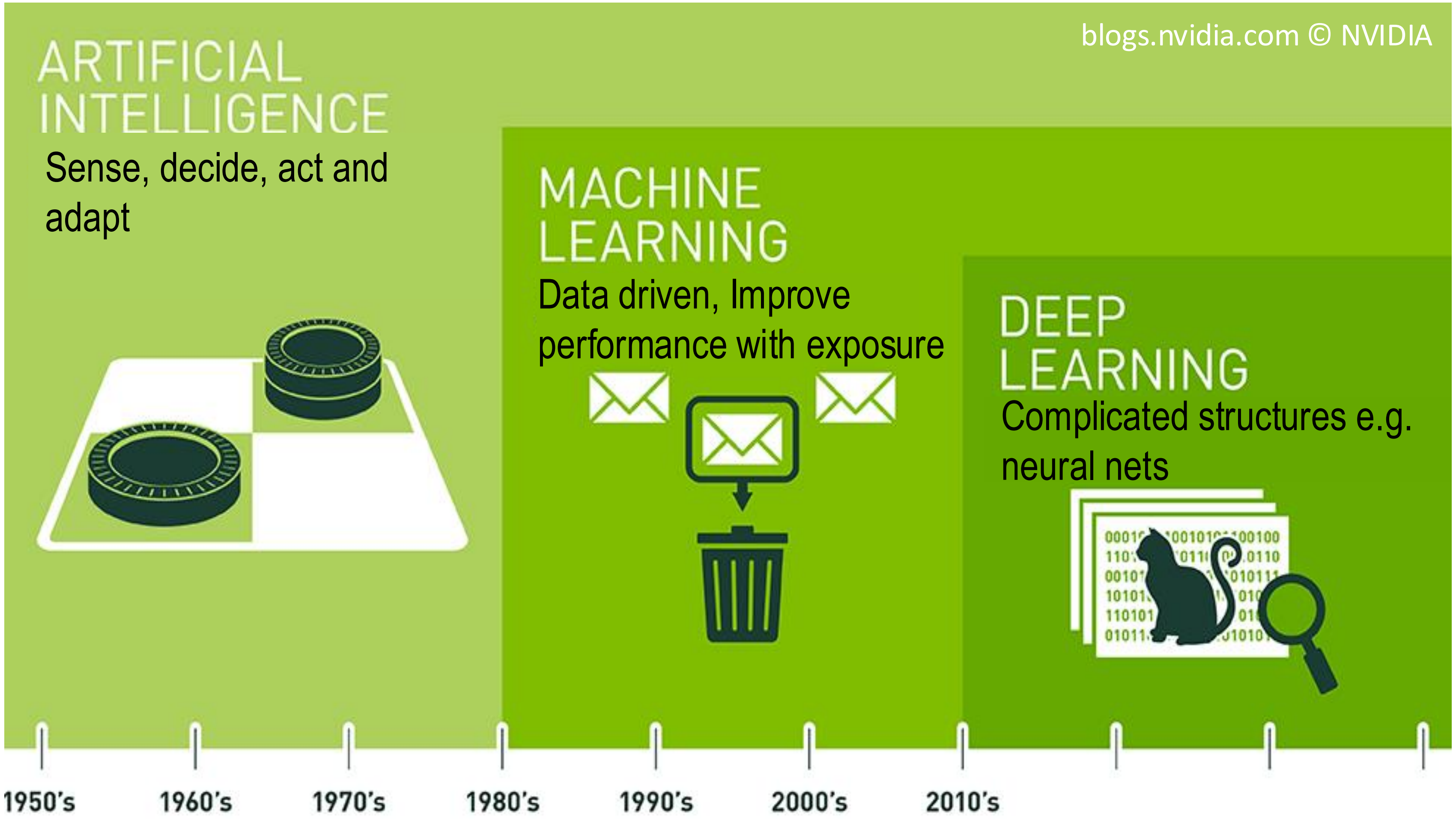
Ben Tapley

Research Scientist
PhD in Applied Mathematics, NTNU
SINTEF Digital
Analytics and Artificial Intelligence group
Oslo

Schedule

- Introduction
- Overview of machine learning types
- ML concepts using neural networks
- Break
- Other supervised ML models
- Probabilistic ML
- Unsupervised ML

AI - Machine learning – Deep learning



SINTEF

Machine learning

- Everything* that happens in the world can be described by mathematical functions!
- Some of these functions are simple.
- Some are more complex.
- As computers get more advanced, we can solve more problems with them.
- While we might never understand some real-world functions, we can observe their effects by recording ***data***.
- Using this data, we can guess/approximate complex functions.

An A.I.-Generated Picture Won an Art Prize. Artists Aren't Happy.

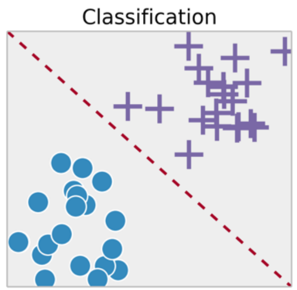
"I won, and I didn't break any rules," the artwork's creator says.

"I couldn't believe what I was seeing," he said. "I felt like it was demonically inspired — like some otherworldly force was involved."

https://www.nytimes.com/2022/09/02/technology/ai-artificial-intelligence-artists.html

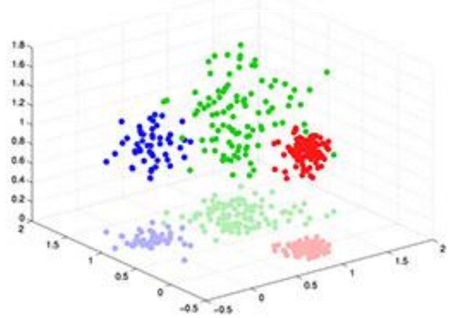
SINTEF

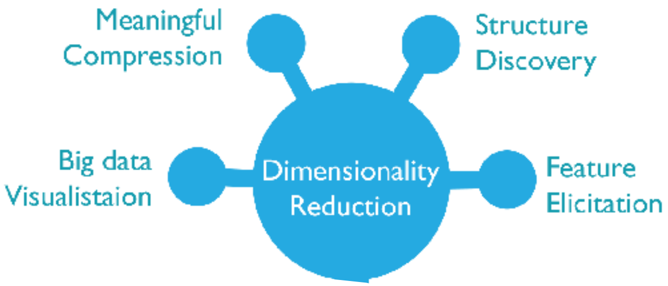
towardsdatascience.com

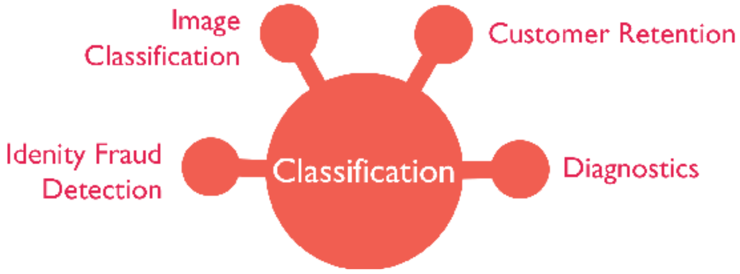


towardsdatascience.com

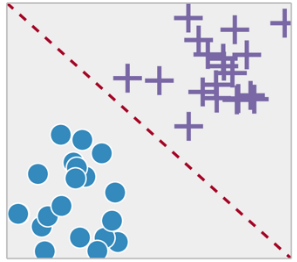
Dimensionality reduction



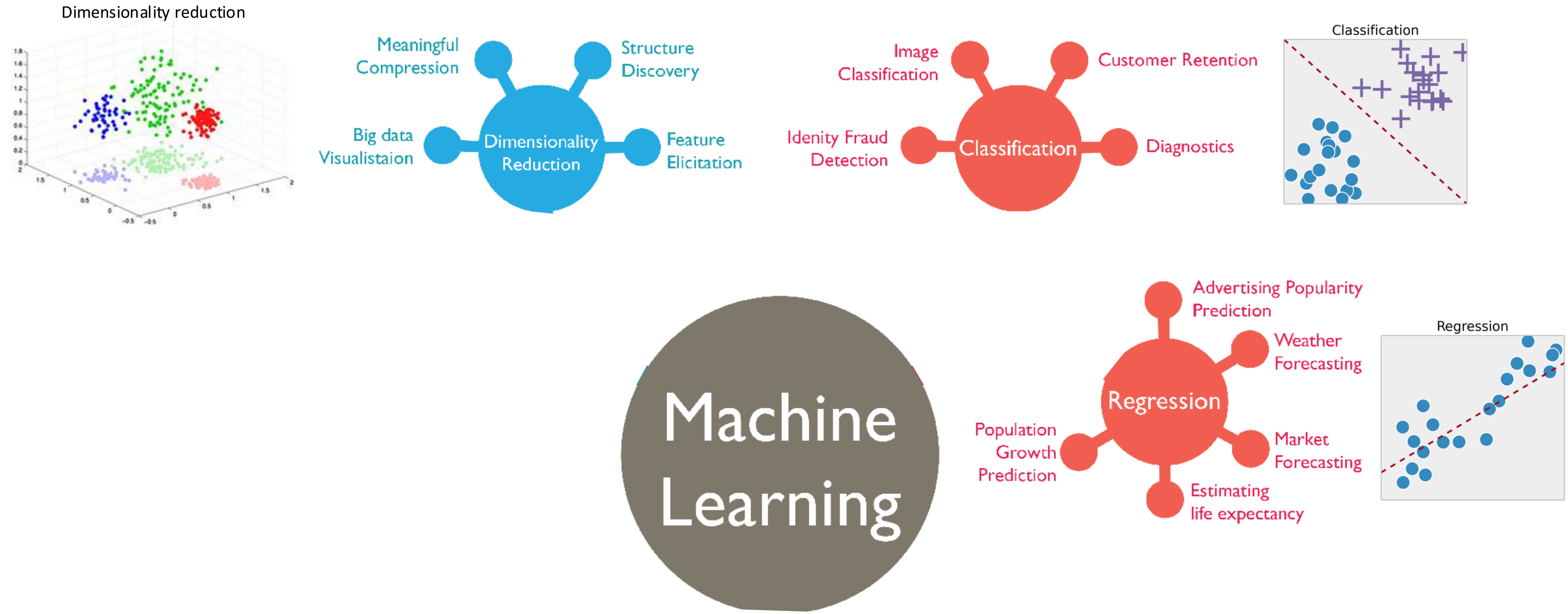




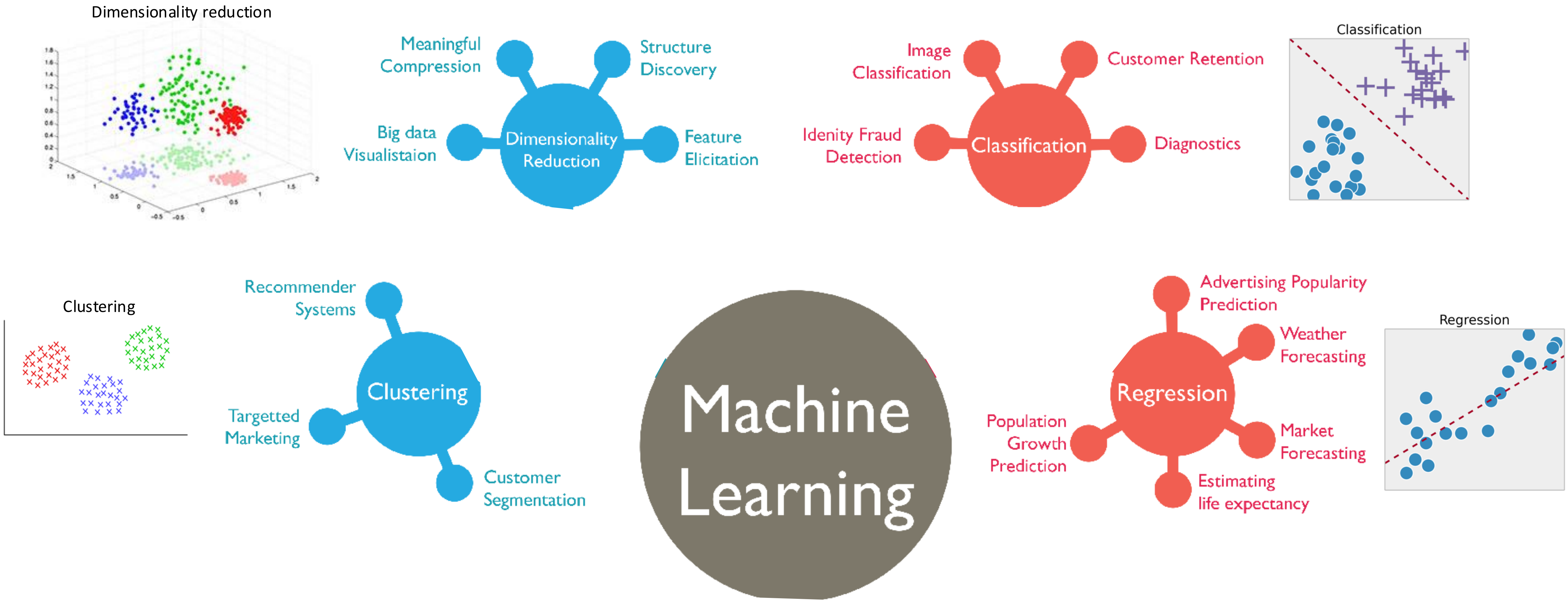
Classification



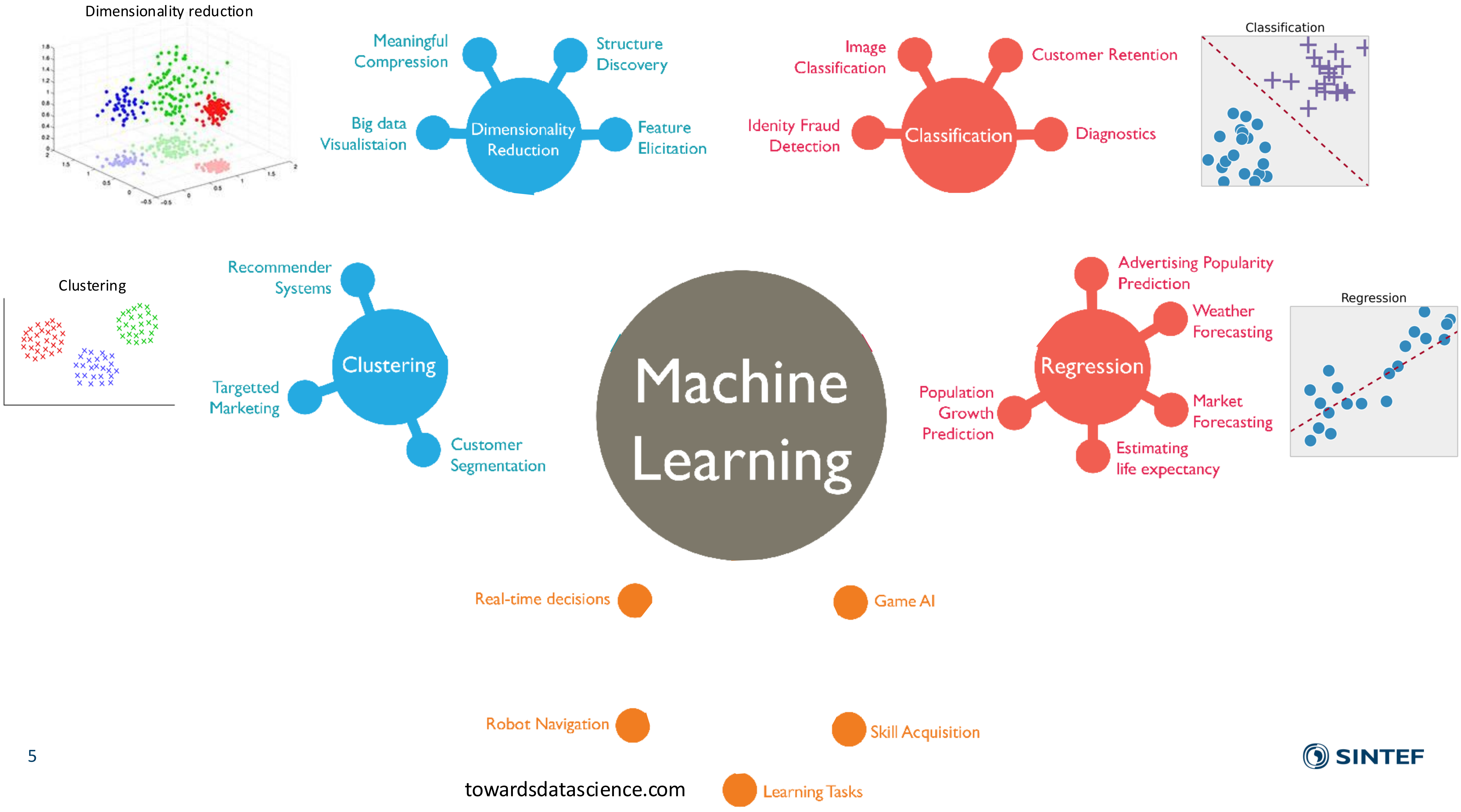
towardsdatascience.com



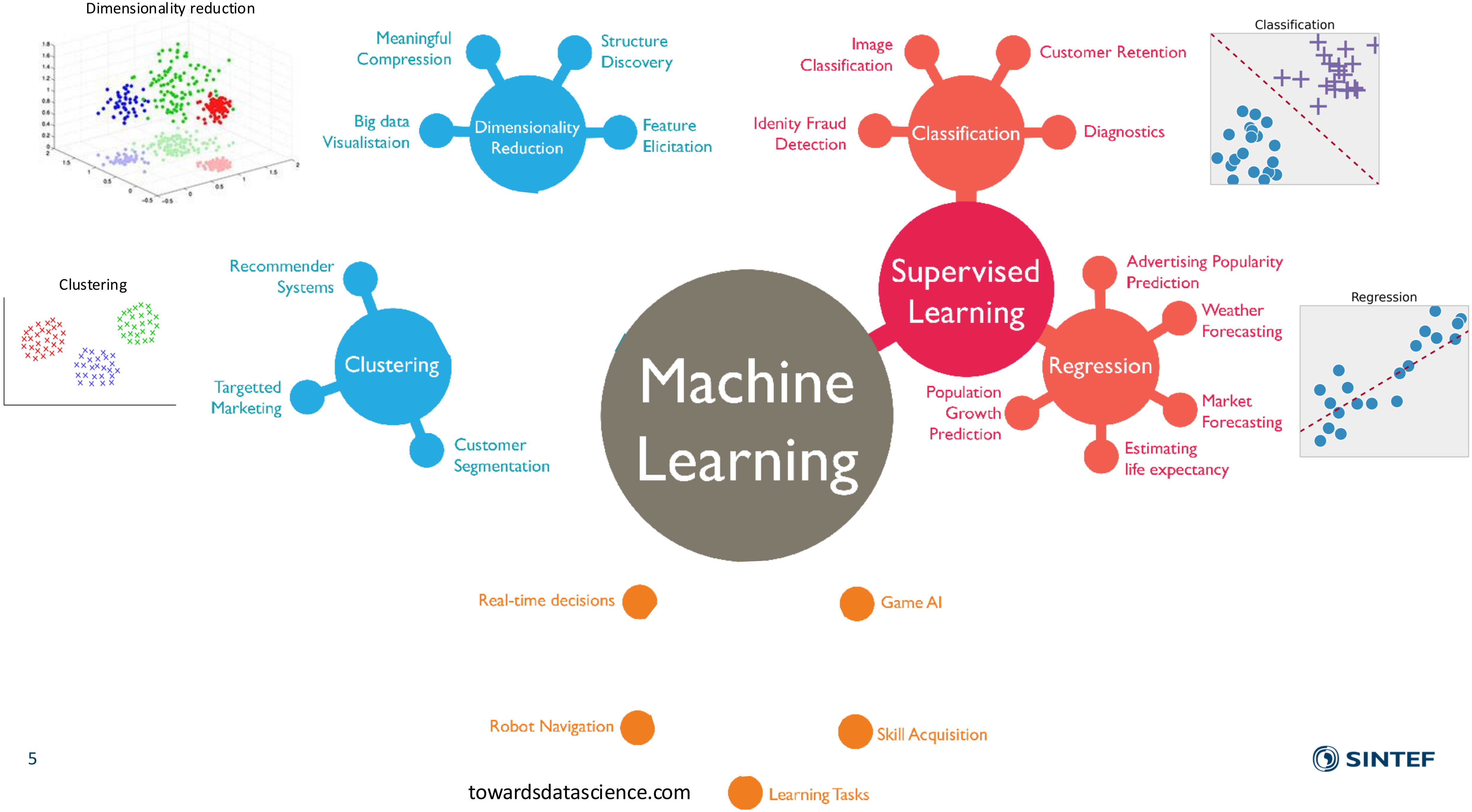
towardsdatascience.com



towardsdatascience.com

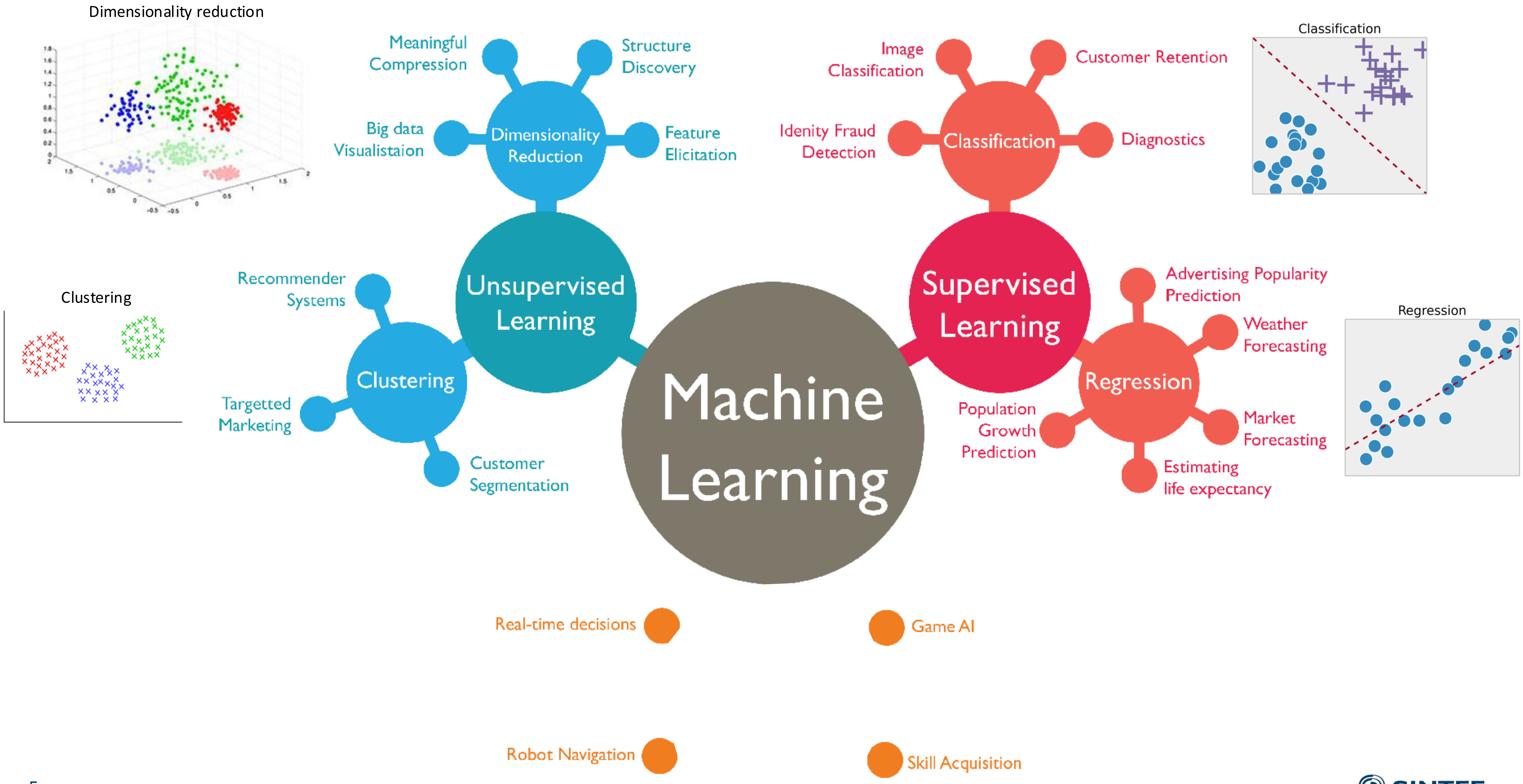


towardsdatascience.com



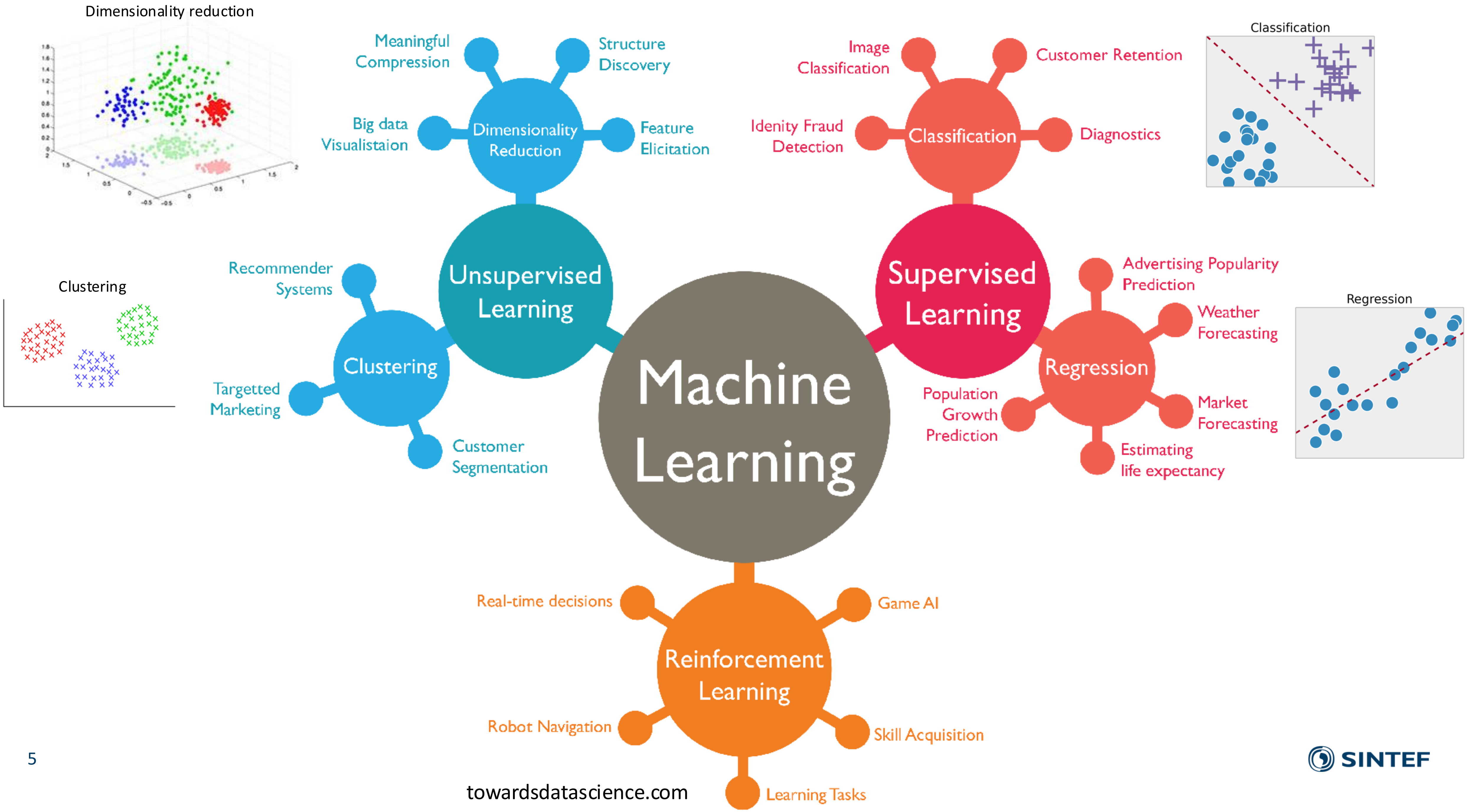
towardsdatascience.com

SINTEF



towardsdatascience.com

Learning Tasks



towardsdatascience.com

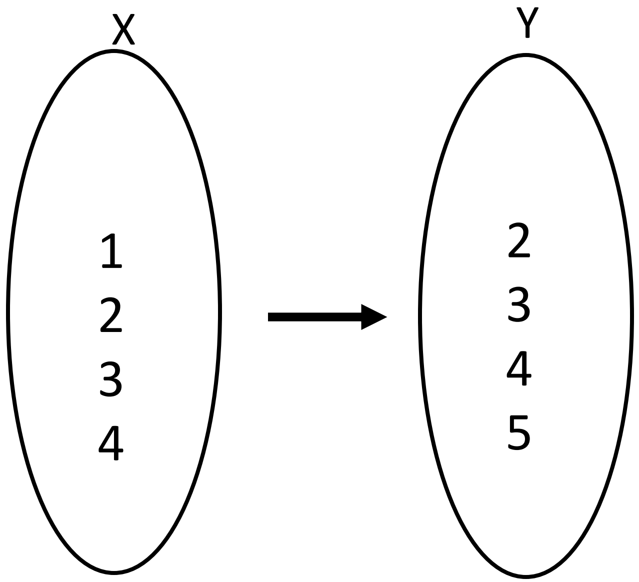
Supervised learning

- When the dataset includes **labels**
- Given enough examples, we can learn complex relationships between the input data and the labels

Supervised learning

- When the dataset includes **labels**
- Given enough examples, we can learn complex relationships between the input data and the labels

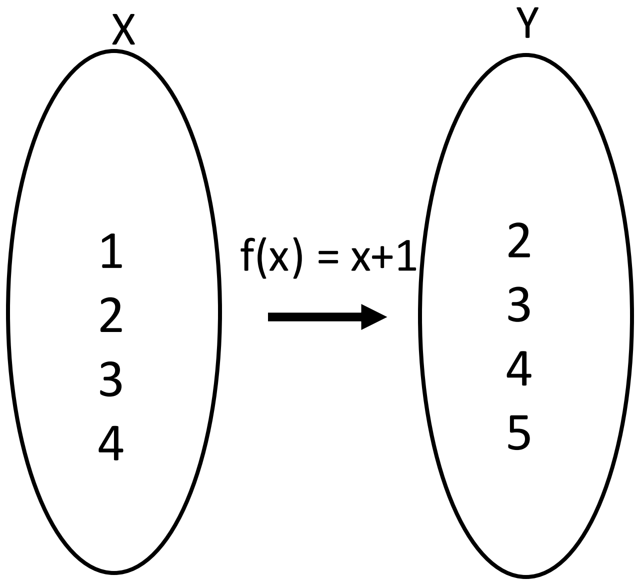
A simple function: find function f(x), such that f(1)=2, f(2)=3 , f(3)=4 , f(4)=5



Supervised learning

- When the dataset includes **labels**
- Given enough examples, we can learn complex relationships between the input data and the labels

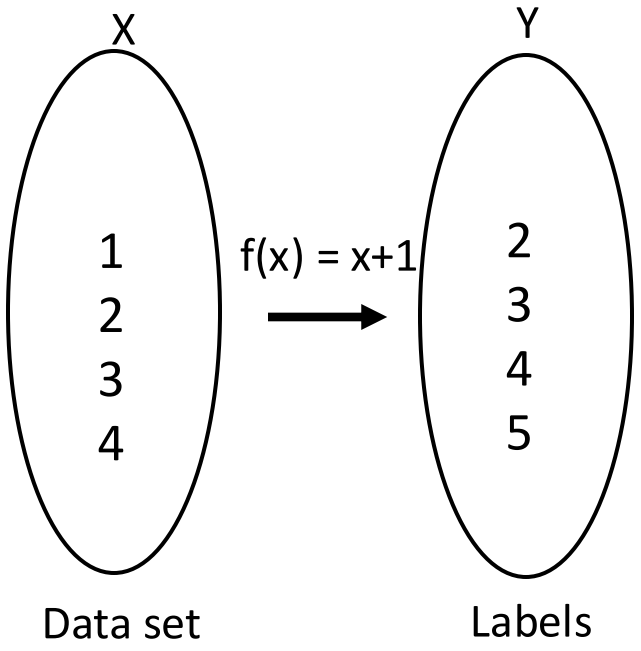
A simple function: find function f(x), such that f(1)=2, f(2)=3 , f(3)=4 , f(4)=5



Supervised learning

- When the dataset includes **labels**
- Given enough examples, we can learn complex relationships between the input data and the labels

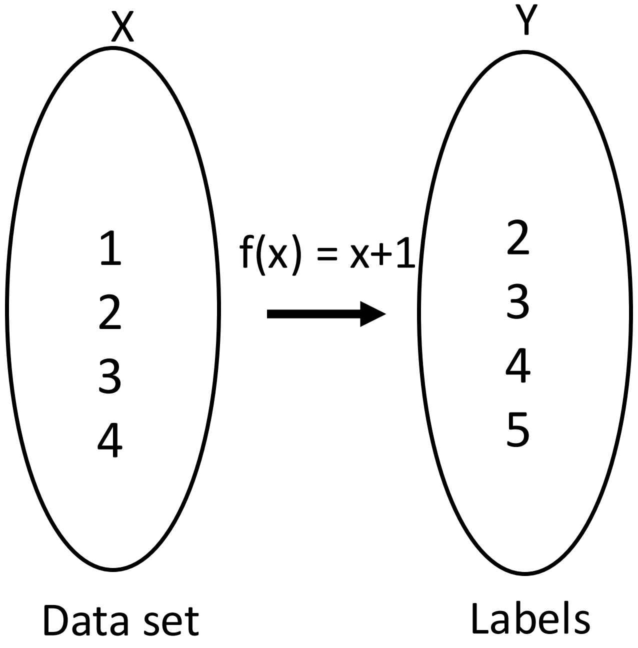
A simple function: find function f(x), such that f(1)=2, f(2)=3 , f(3)=4 , f(4)=5



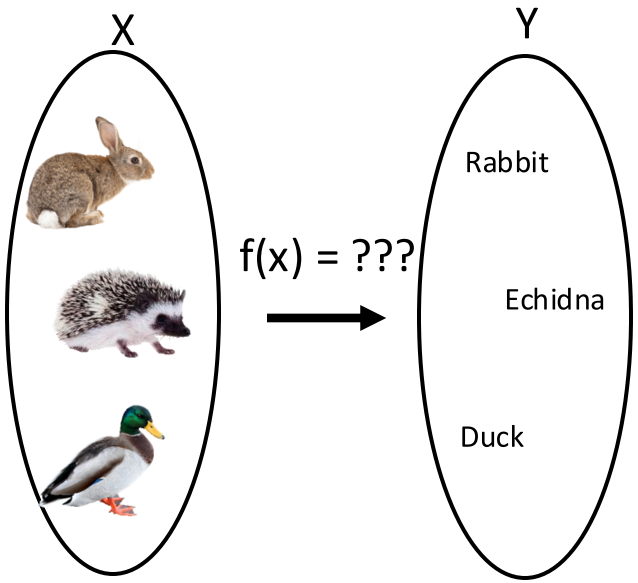
Supervised learning

- When the dataset includes **labels**
- Given enough examples, we can learn complex relationships between the input data and the labels

A simple function: find function f(x), such that f(1)=2, f(2)=3 , f(3)=4 , f(4)=5

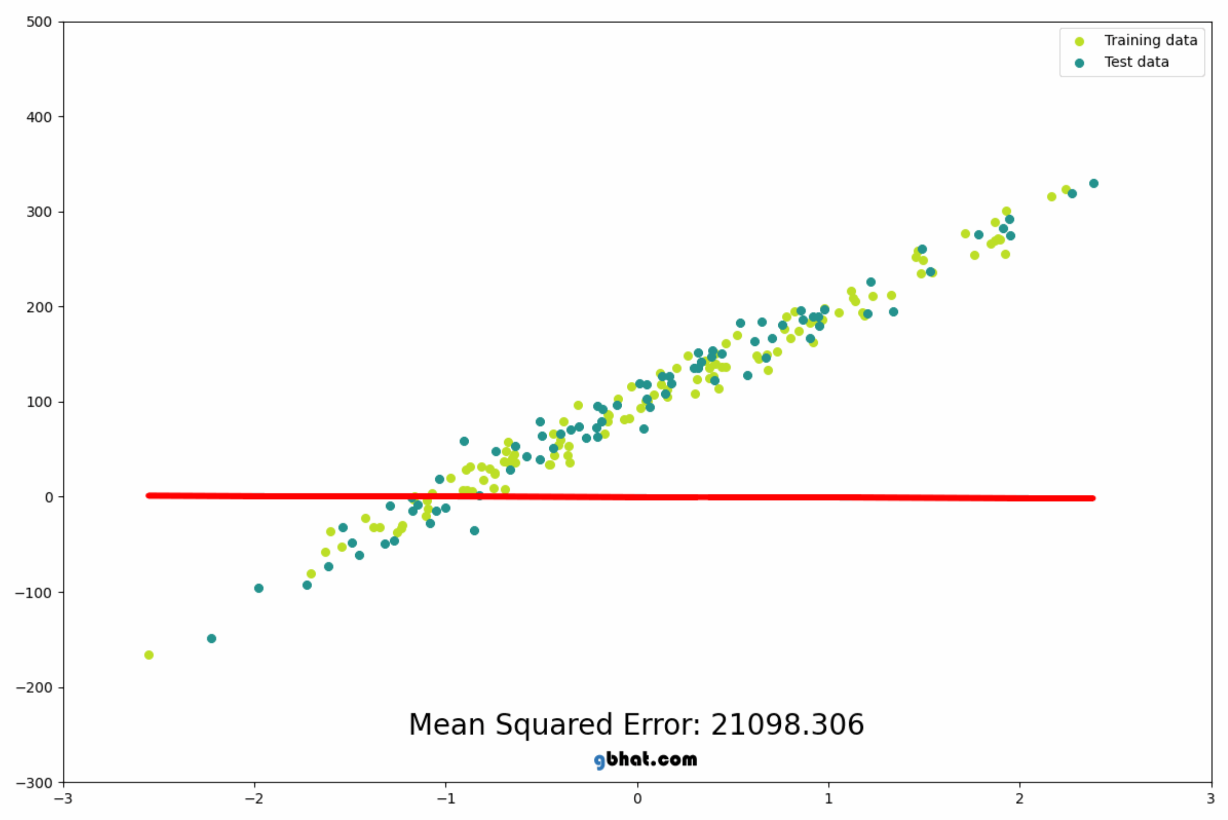


A complex function: identify the species from a picture



Example: regression

Y (labels)

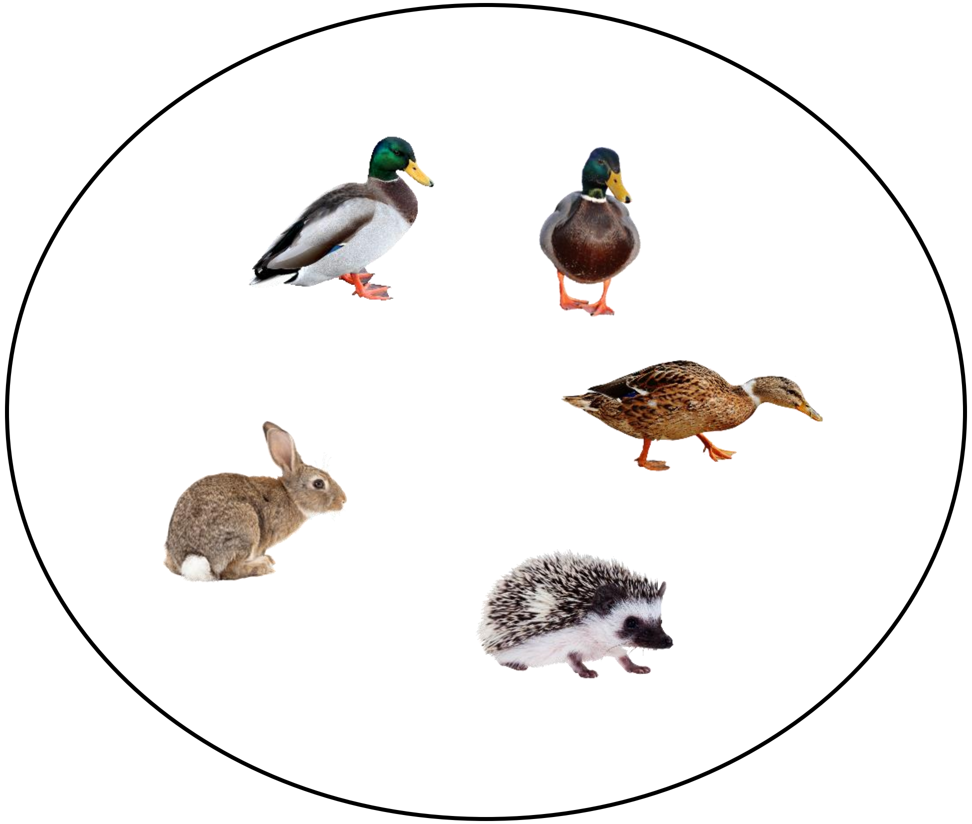


X (input data)

SINTEF

Unsupervised learning

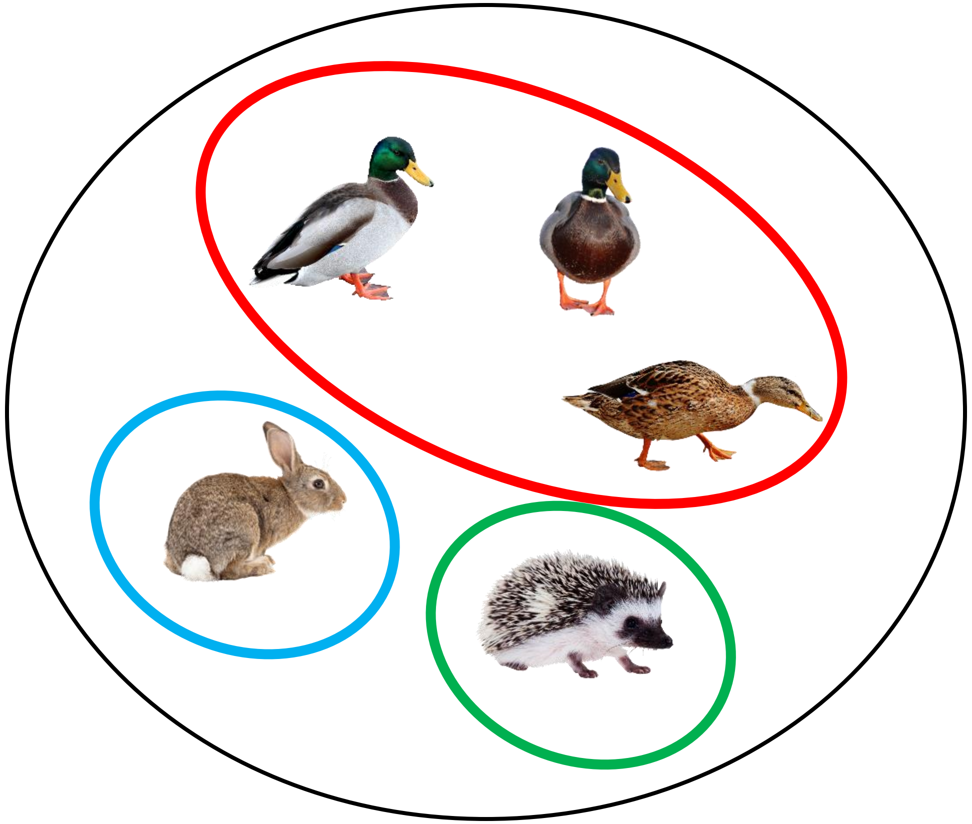
- When the dataset has **no labels**
- We want to identify patterns in the data
- Unsupervised -> let the algorithm decide how to label the data



Data set

Unsupervised learning

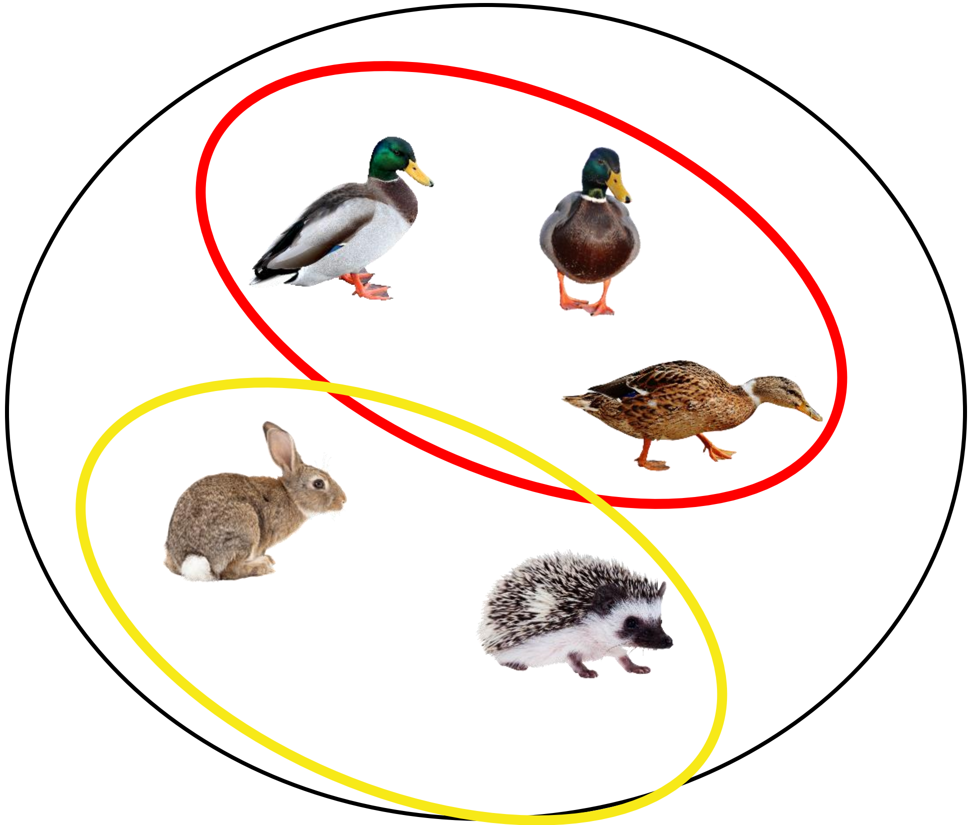
- When the dataset has **no labels**
- We want to identify patterns in the data
- Unsupervised -> let the algorithm decide how to label the data



Data set

Unsupervised learning

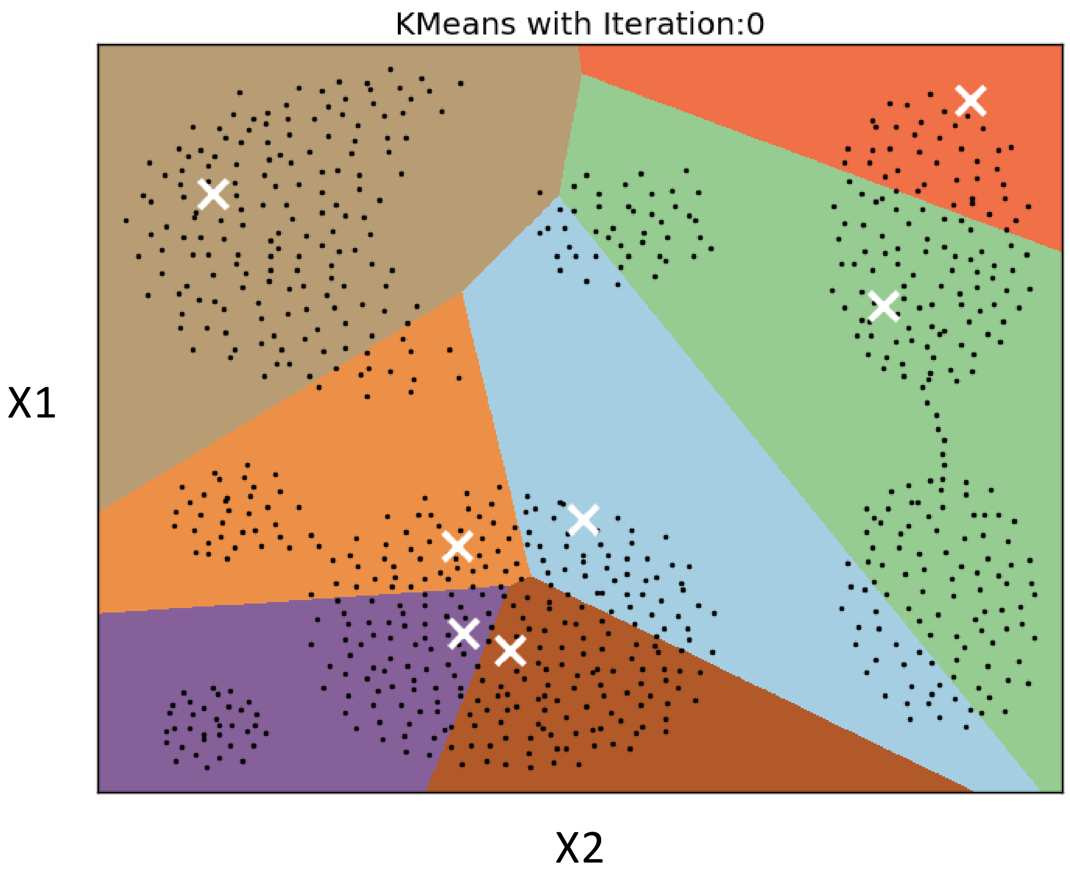
- When the dataset has **no labels**
- We want to identify patterns in the data
- Unsupervised -> let the algorithm decide how to label the data



Data set

SINTEF

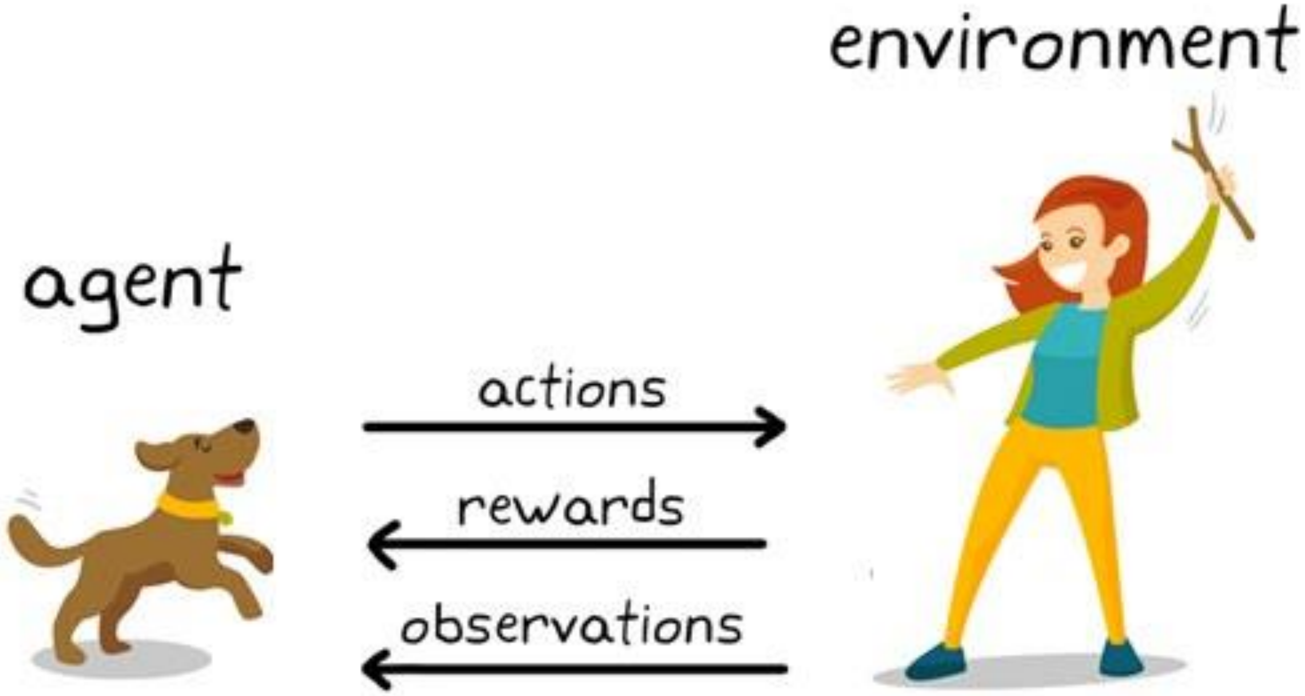
Example of unsupervised learning: clustering



Algorithm learns how to label the data y (the colours)

Reinforcement learning

- Similar to the way humans learn
- No data!
- Instead:
  - an environment
  - a way to explore and interact with the environment
  - learn from mistakes and reward good actions



SINTEF

Example: learning to play a video game

SINTEF

Reward the model for good performance

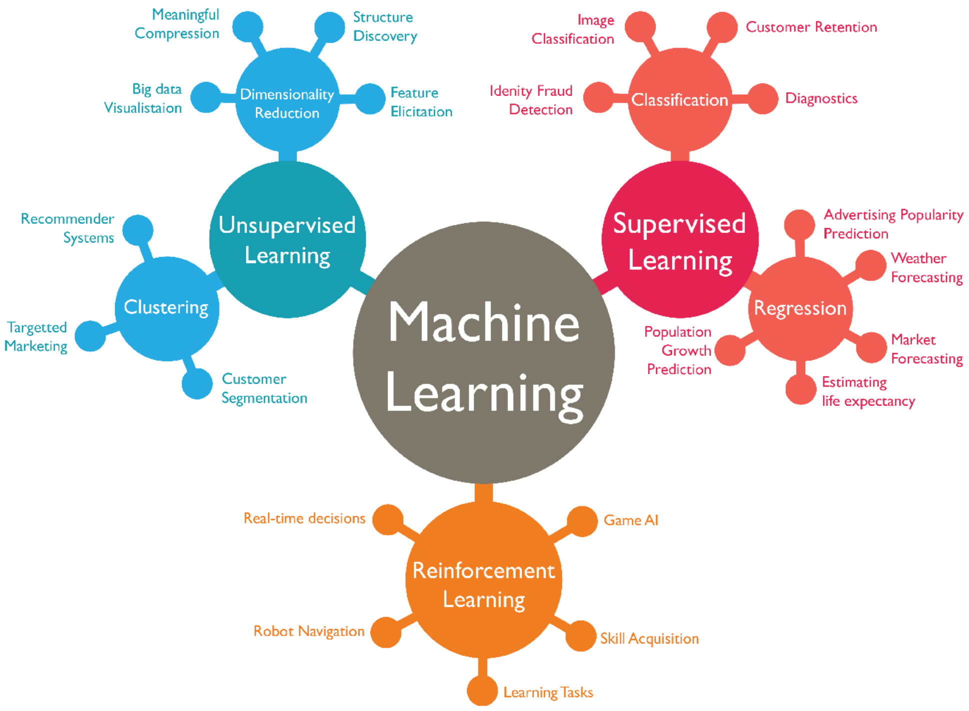
Let the model optimise its decision making process


Run #1050
Exploration

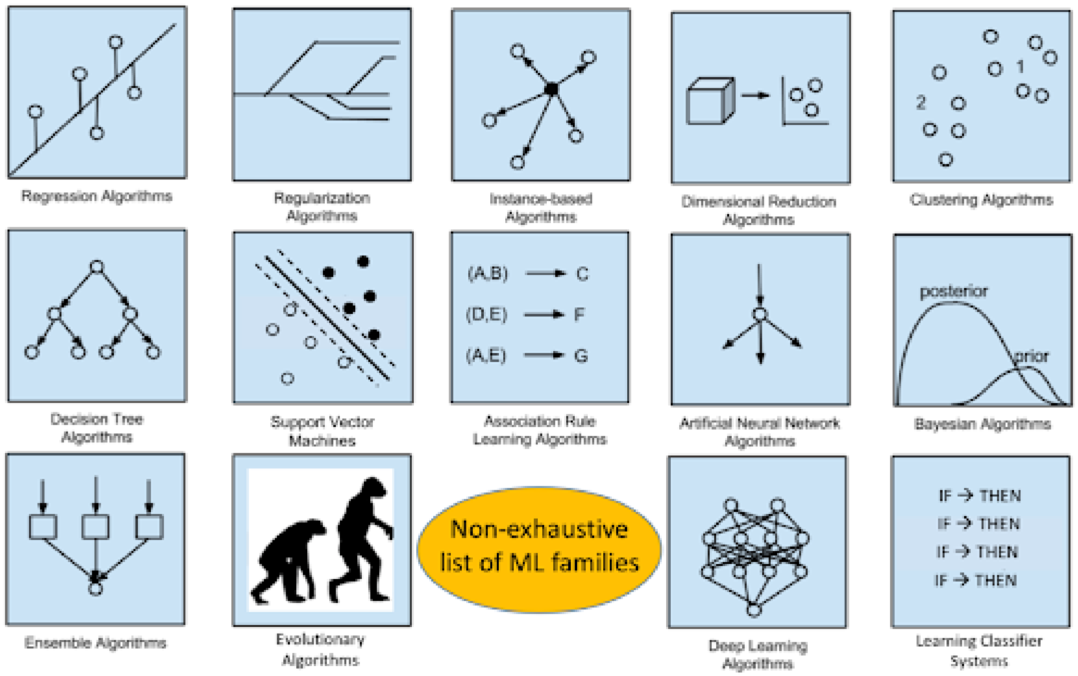

Runs #15k-16k
Best : 480m
Training time : 21h
makeagif.com

Quick summary

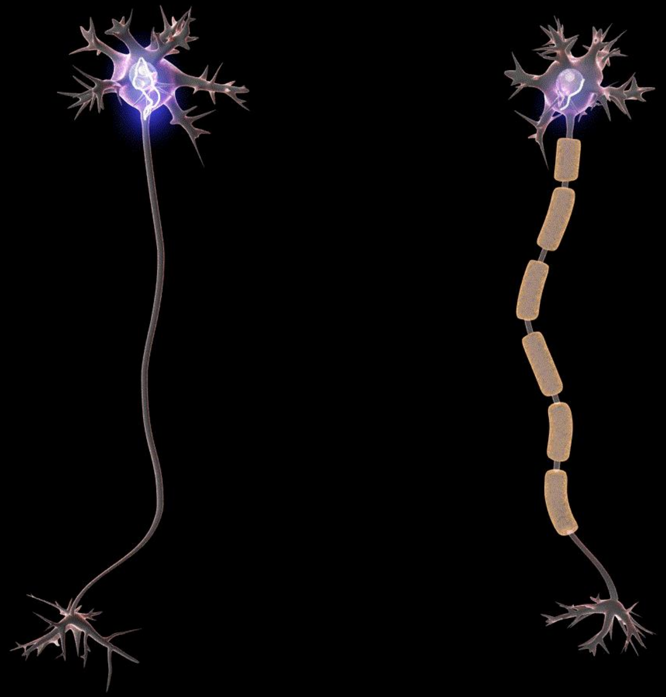
We have covered **what** ML it is:



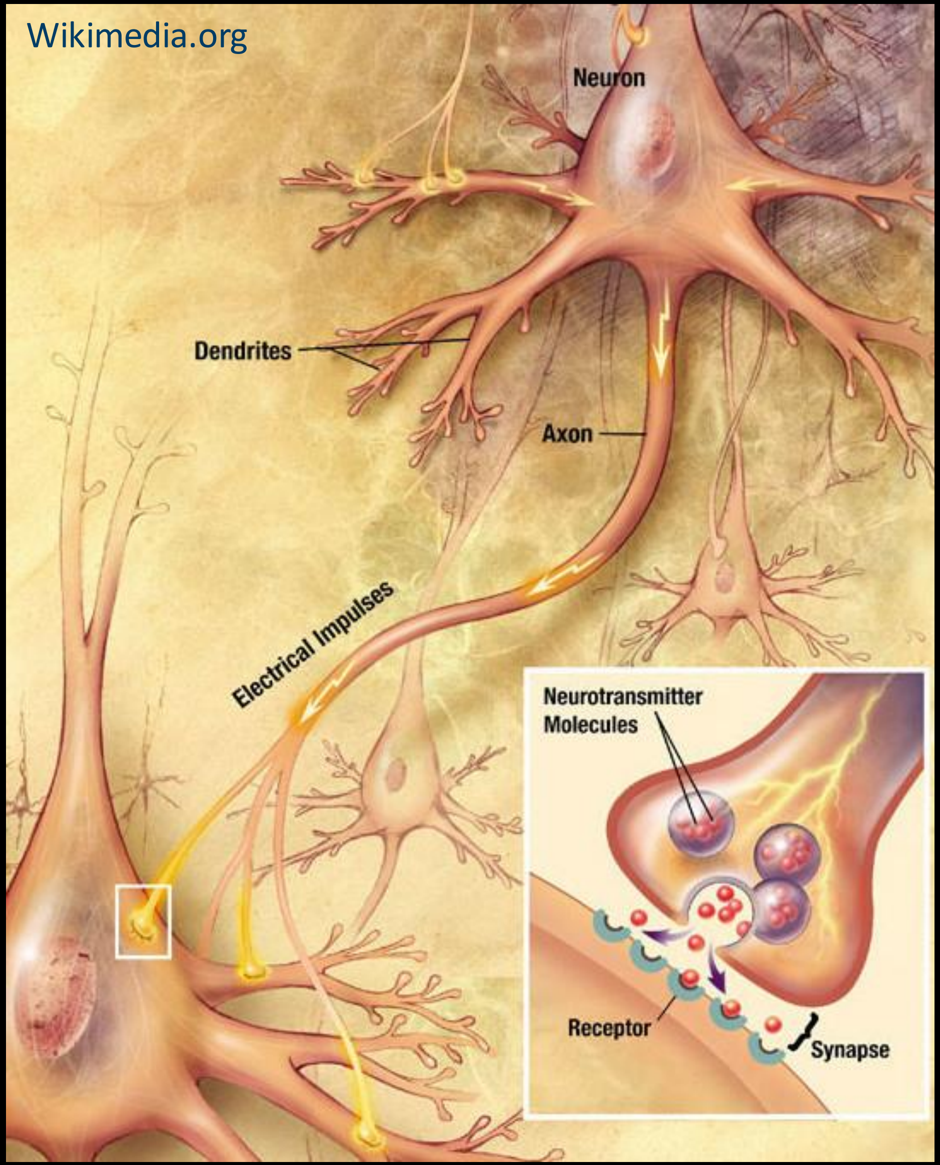
But **how** do ML models work in practice?



Neural networks



nia.nih.gov



Neural Networks

Neural networks are inspired by the brain. They can approximate complex functions!

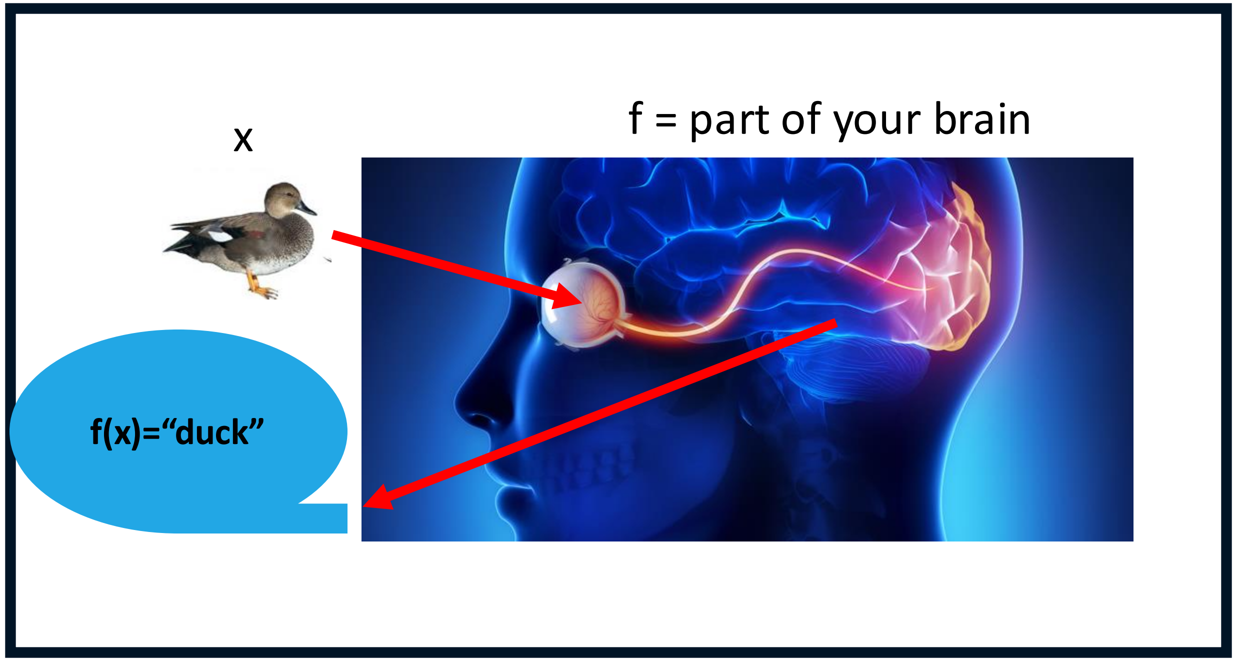
$$y = f(x)$$

Neural Networks

Neural networks are inspired by the brain. They can approximate complex functions!

y = f(x)

Biological neural network

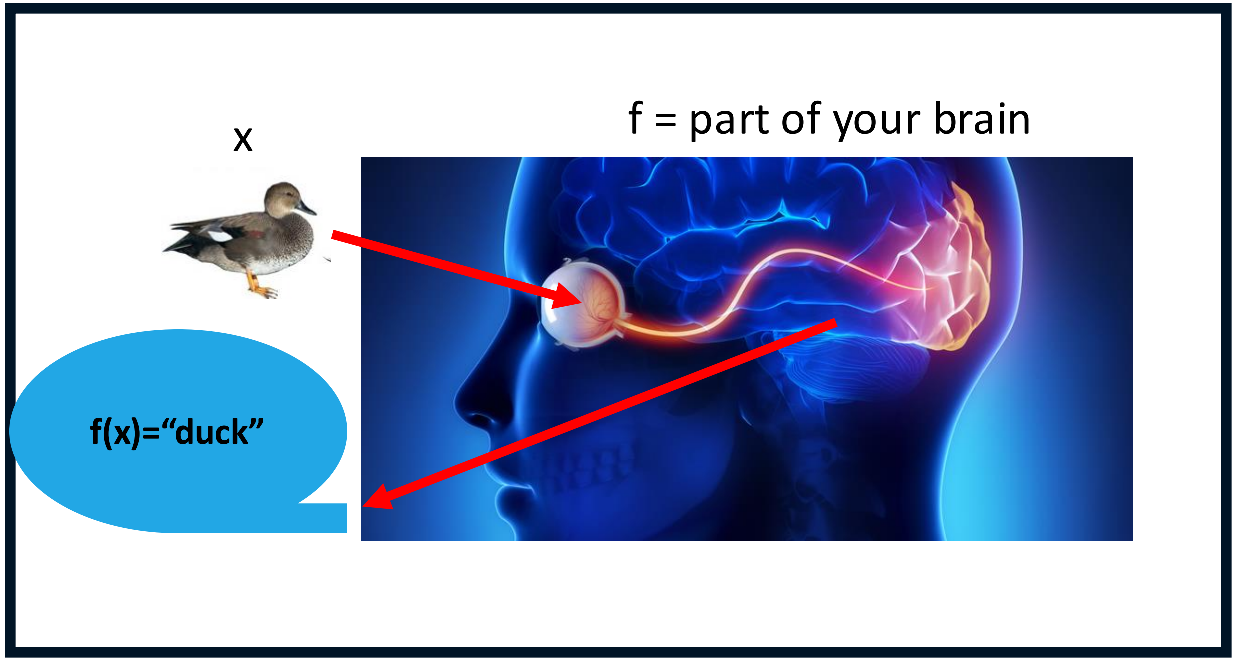


Neural Networks

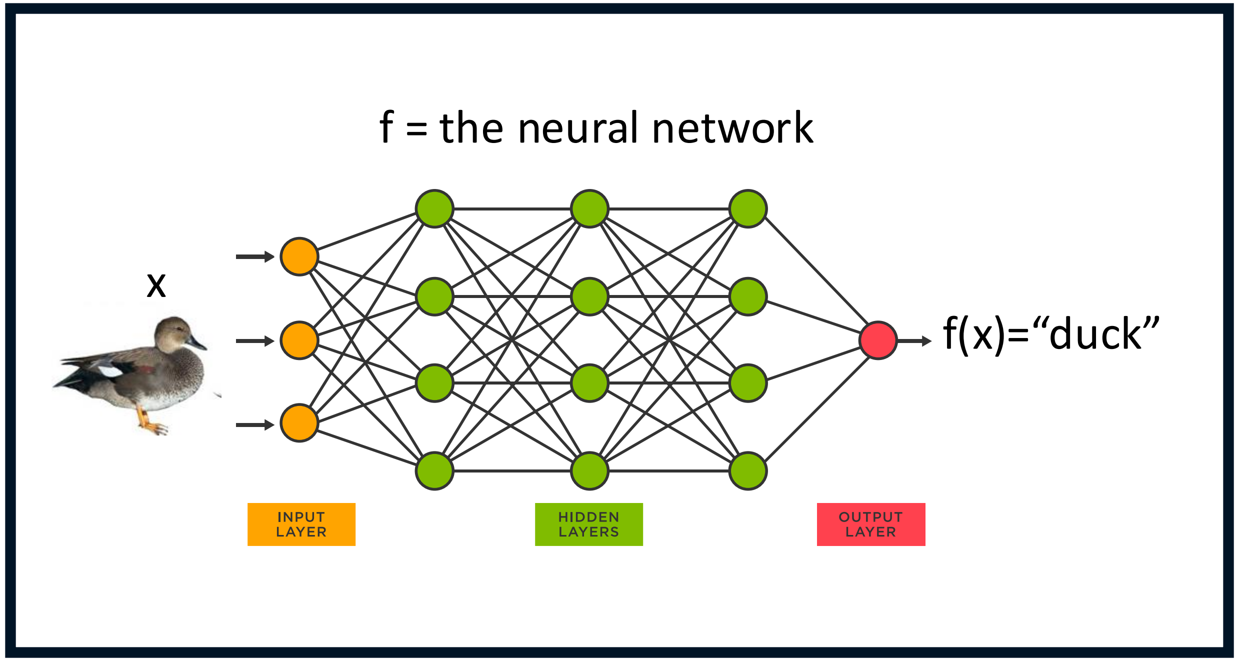
Neural networks are inspired by the brain. They can approximate complex functions!

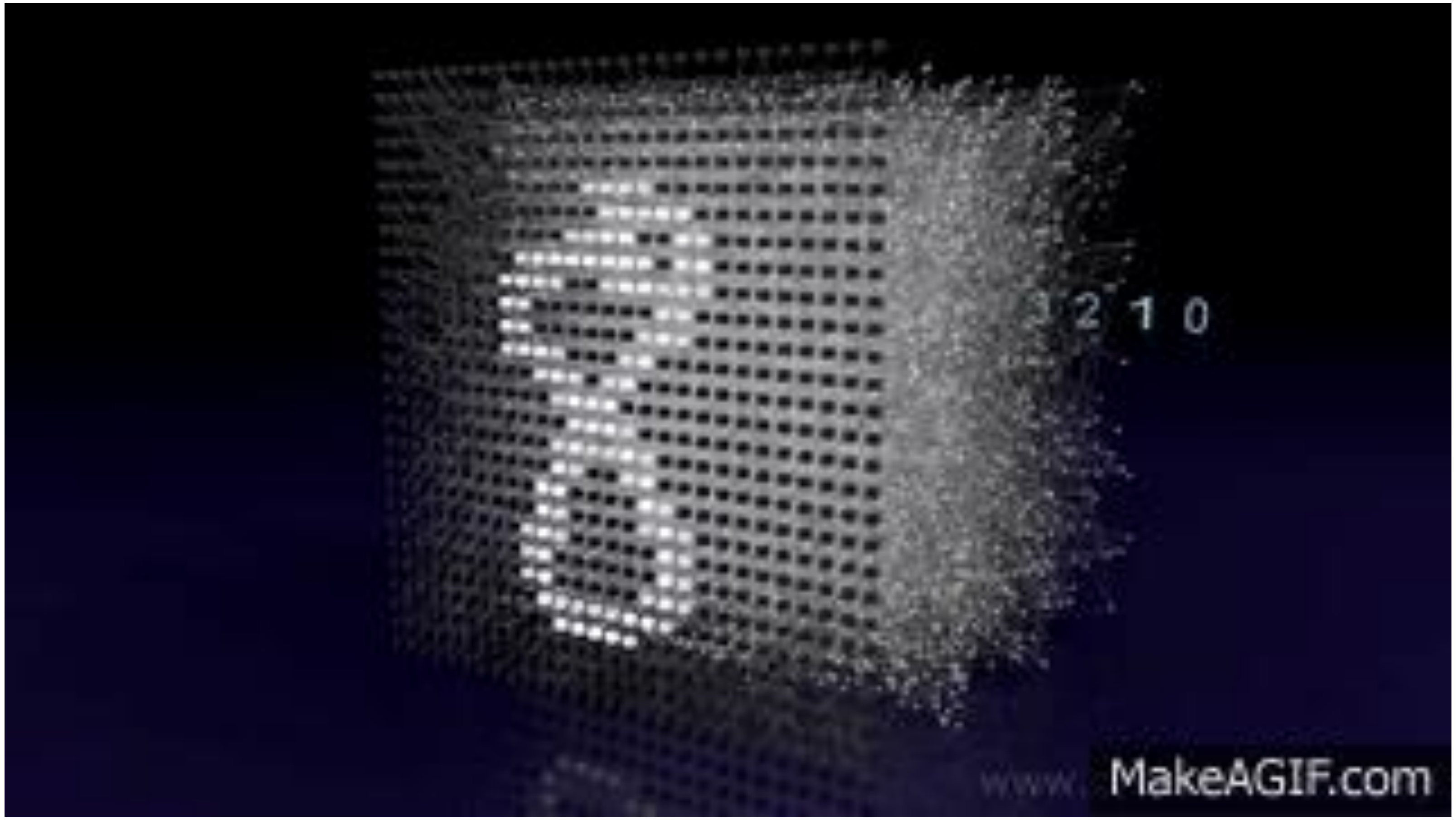
y = f(x)

Biological neural network



Artificial neural network



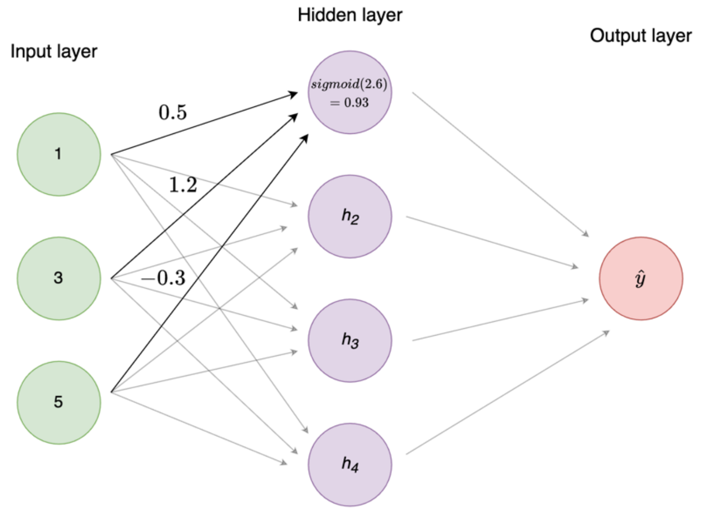

3 2 1 0
www.MakeAGIF.com

The Role of Free Parameters

- Neural networks contain many free parameters that control the output of the function
- The goal is to tune these parameters to give the desired output

$$f(x) = \sum_{i=1}^{m} \alpha_i \sigma(w_i^T x)$$

- the "free parameters" are the α_i and w_i



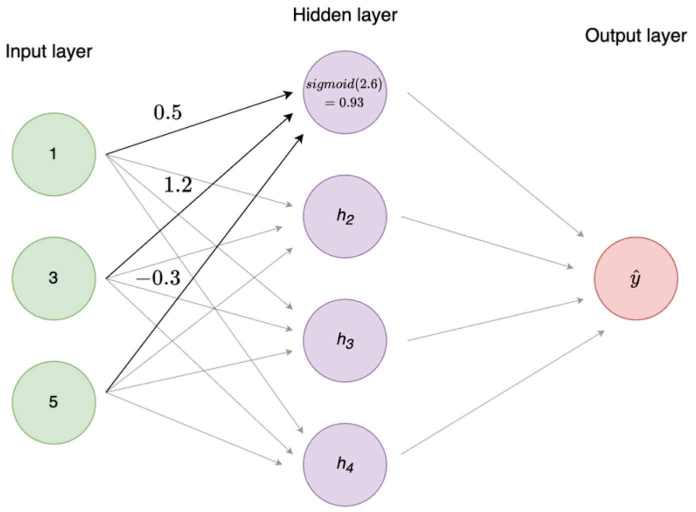
SINTEF

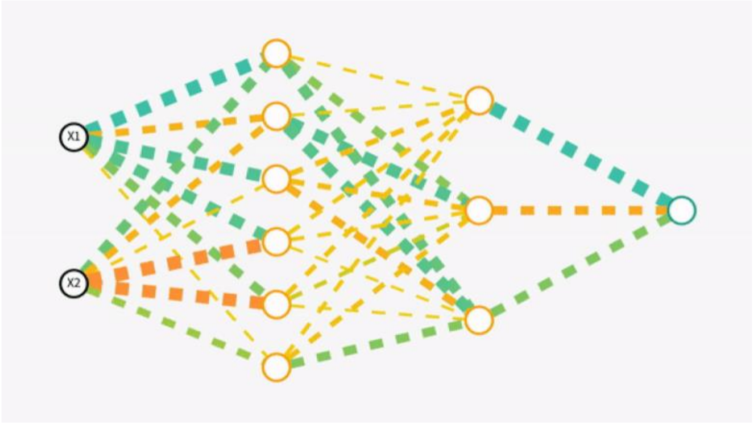
The Role of Free Parameters

- Neural networks contain many free parameters that control the output of the function
- The goal is to tune these parameters to give the desired output

$$f(x) = \sum_{i=1}^{m} \alpha_i \sigma(w_i^T x)$$

- the "free parameters" are the α_i and w_i





Why do they work?

Universal approximation theorem

"There always exists a neural network that can approximate any function"

Why do they work?

Universal approximation theorem

"There always exists a neural network that can approximate any function"

Given a function g(x). Could be unknown or in the form of data $(x_i, g(x_i))$

Then there exists some set of parameters (the α_i and w_i) such that f(x) can be arbitrarily close to any function.

But how to find these parameters?

Why do they work?

Universal approximation theorem

"There always exists a neural network that can approximate any function"

Given a function g(x). Could be unknown or in the form of data (x_i, $g(x_i)$)

Then there exists some set of parameters (the α_i and w_i) such that f(x) can be arbitrarily close to any function.

$$f(x) = \sum_{i=1}^{m} \alpha_i \sigma(w_i^T x) \quad \rightarrow \quad ||f(x) - g(x)|| < \epsilon, \quad \forall \epsilon > 0$$

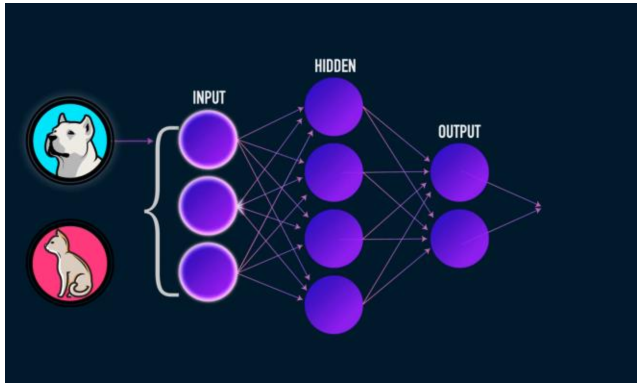
But how to find these parameters?

Training: Loss and gradient descent

Idea: change the parameters to minimise the error on the data set

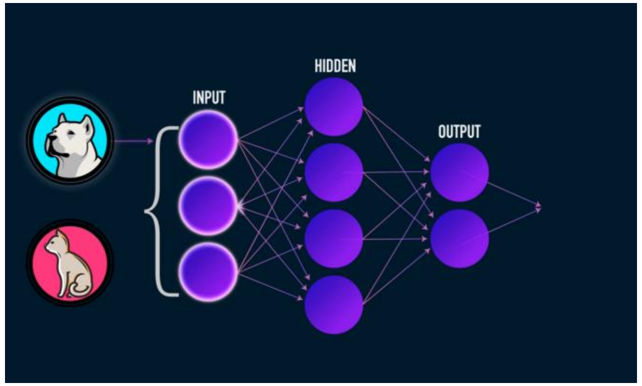
Training: Loss and gradient descent

Idea: change the parameters to minimise the error on the data set



Training: Loss and gradient descent

Idea: change the parameters to minimise the error on the data set

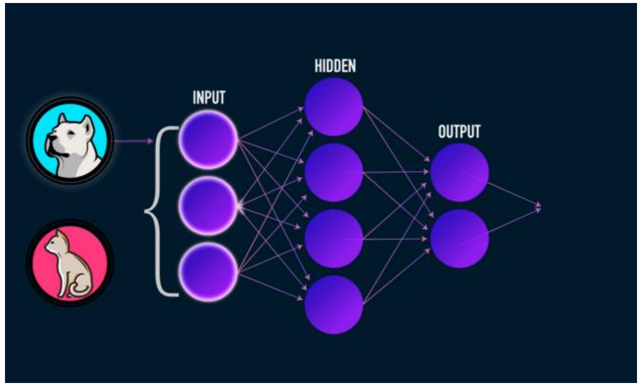


$$loss = \sum_i \|f(x_i) - y_i\|^2 = \text{“prediction”} - \text{“data labels”}$$

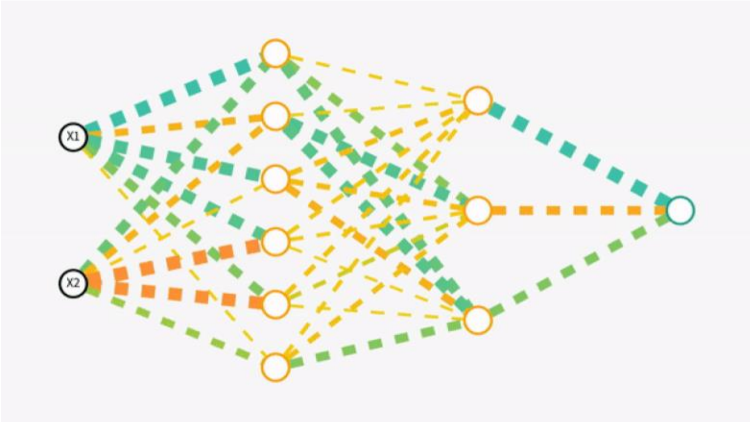
SINTEF

Training: Loss and gradient descent

Idea: change the parameters to minimise the error on the data set



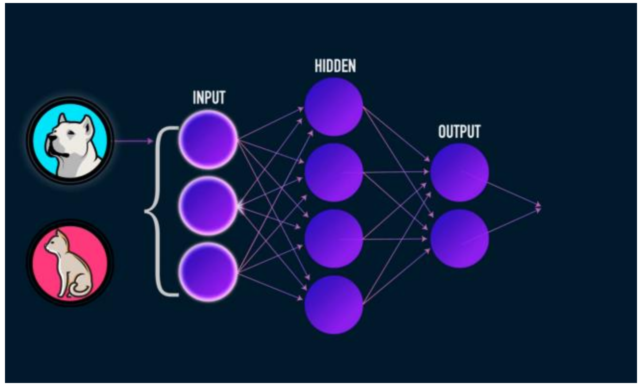
$$loss = \sum_i \|f(x_i) - y_i\|^2 = \text{“prediction”} - \text{“data labels”}$$



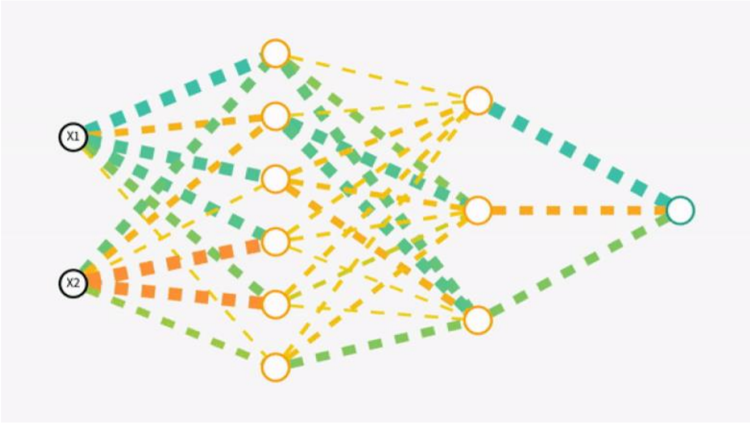
SINTEF

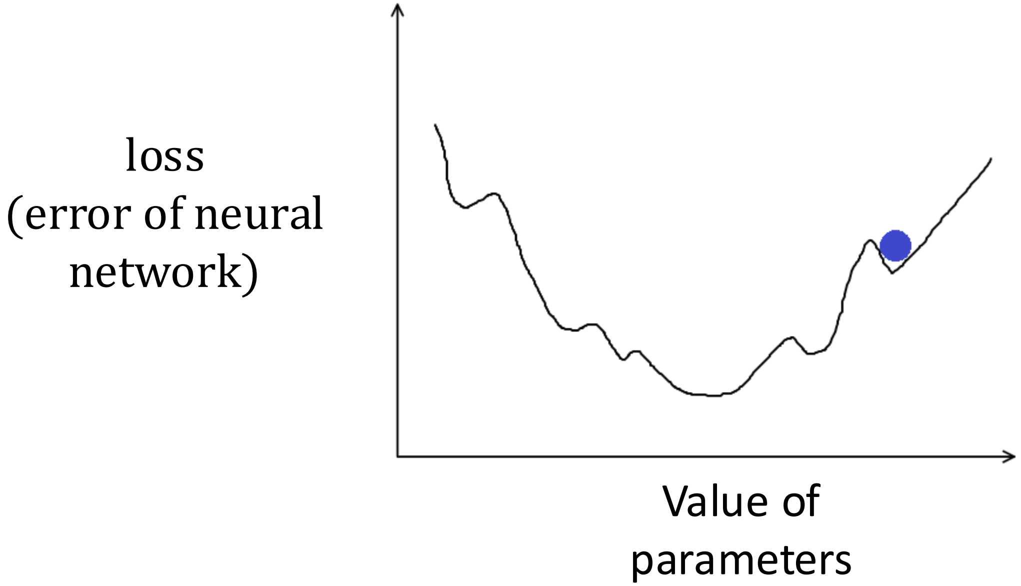
Training: Loss and gradient descent

Idea: change the parameters to minimise the error on the data set



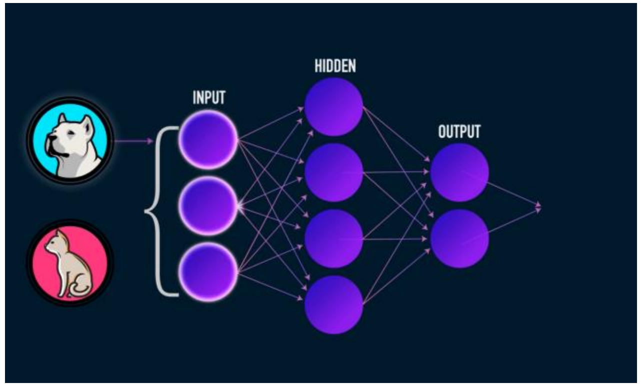
$$loss = \sum_i \|f(x_i) - y_i\|^2 = \text{"prediction"} - \text{"data labels"}$$



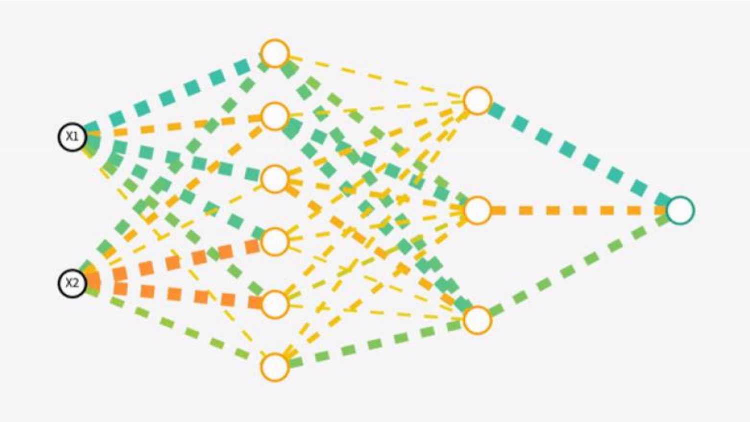


Training: Loss and gradient descent

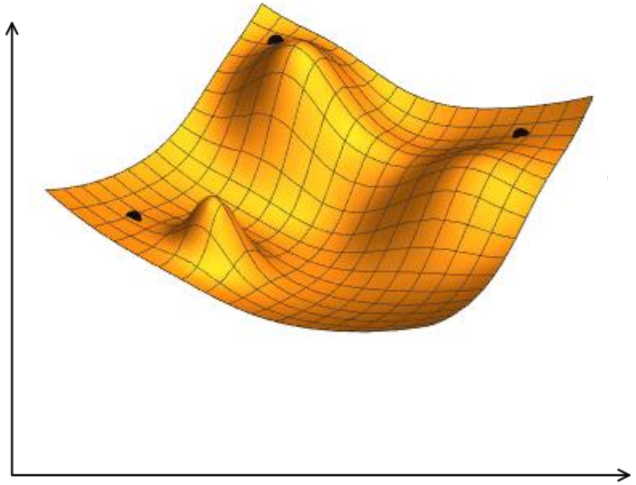
Idea: change the parameters to minimise the error on the data set



$$loss = \sum_i \|f(x_i) - y_i\|^2 = \text{“prediction”} - \text{“data labels”}$$



loss
(error of neural
network)



Value of
parameters

Example: linear regression

Fit a function

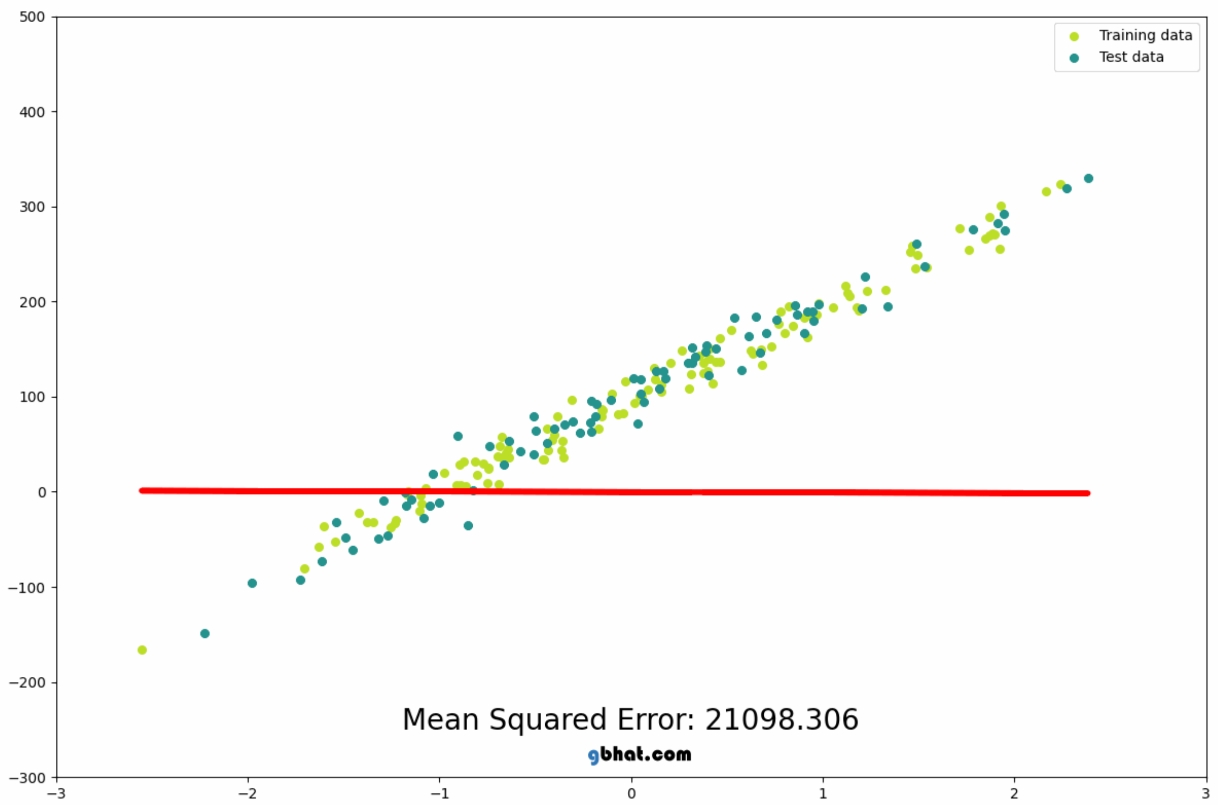
$$f(x) = mx + c$$

to the data.

That minimises the MSE loss:

$$loss = \sum_i \|f(x_i) - y_i\|^2$$

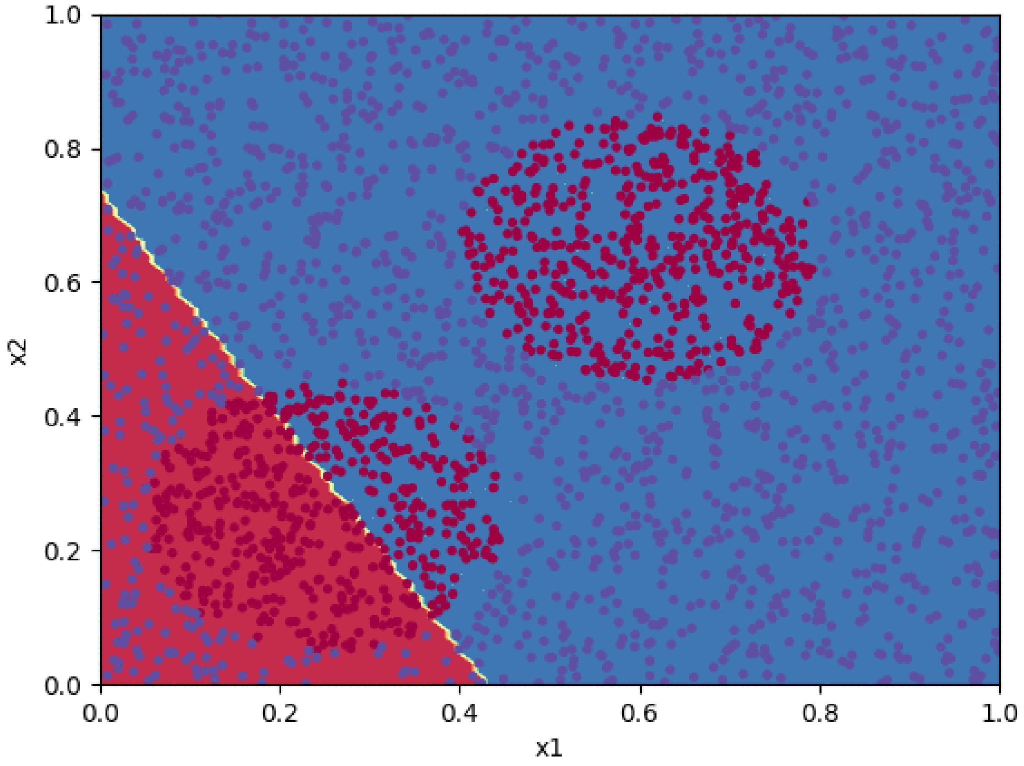
y (labels)

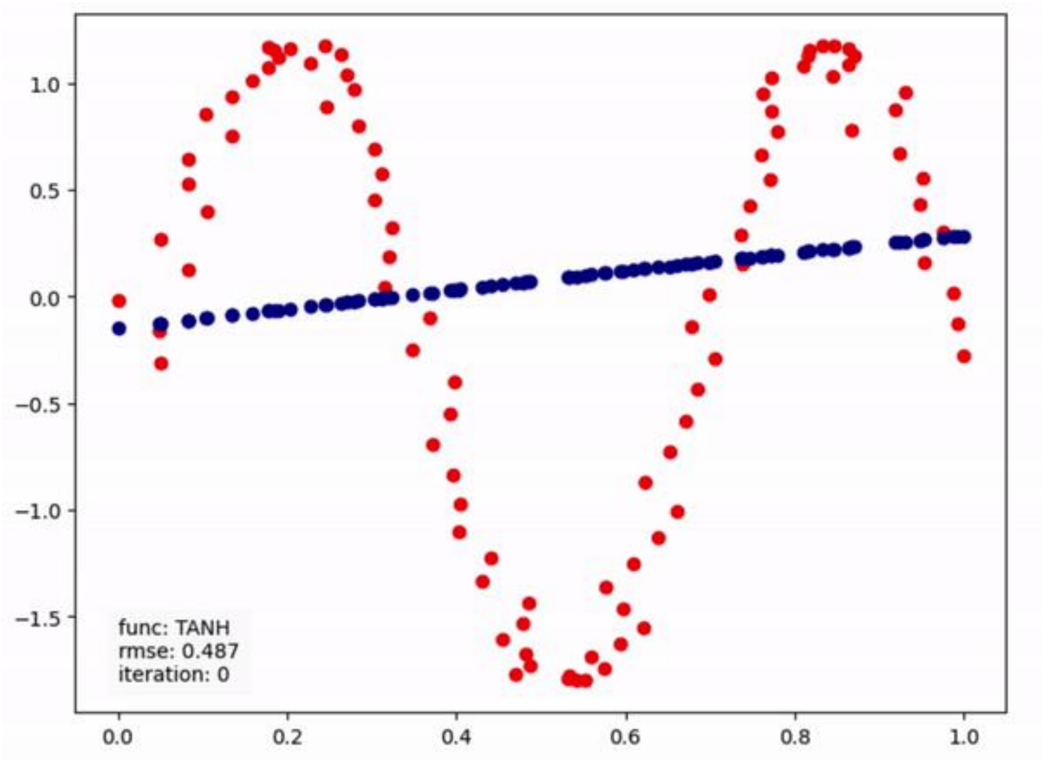


x (input data)

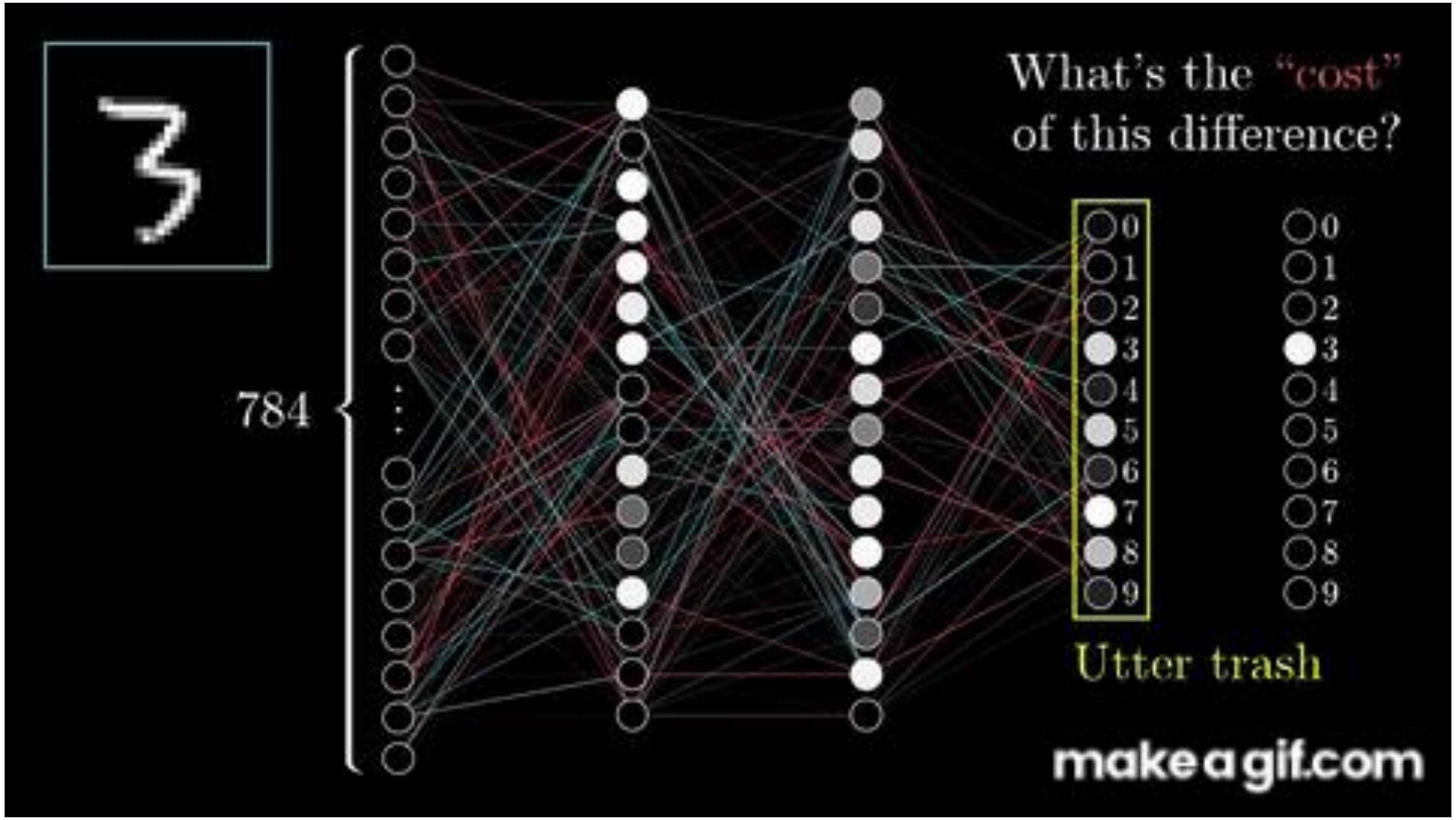
SINTEF

… same for neural networks





Training a neural network



SINTEF

Quick summary

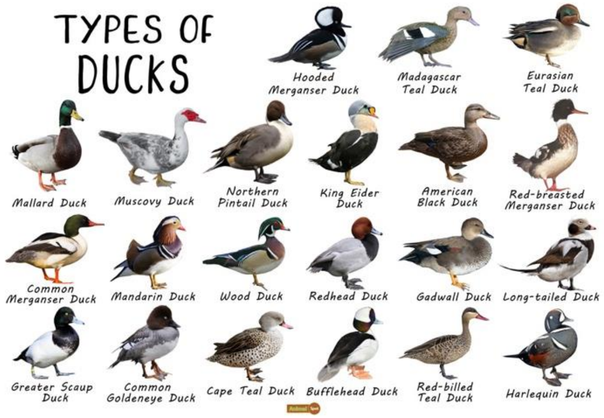
Error

Choice of free parameters

SINTEF

Quick summary

1. Given a labelled data



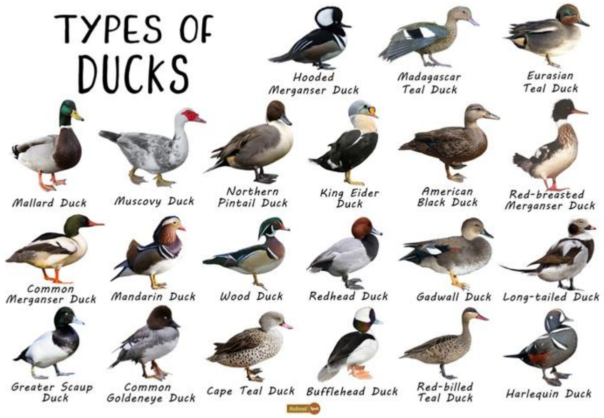
Error

Choice of free parameters

SINTEF

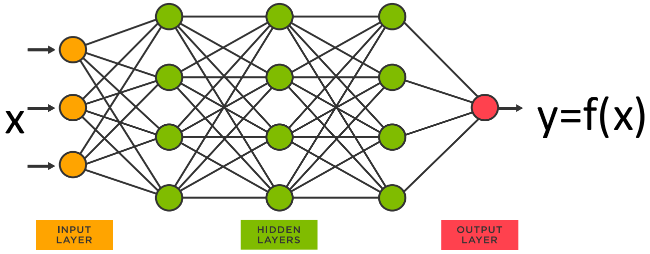
Quick summary

1. Given a labelled data



2. Choose an ML model, initially, with random parameters

f = the neural network

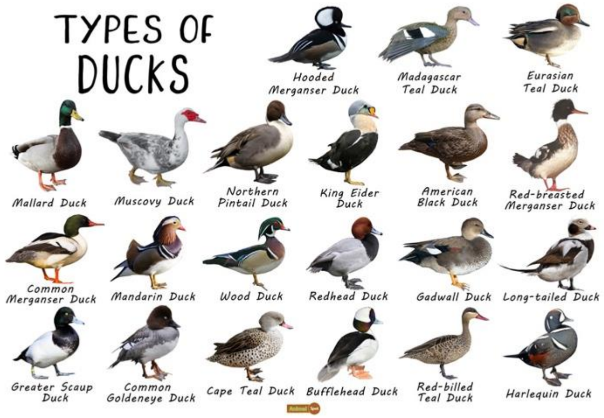


Error

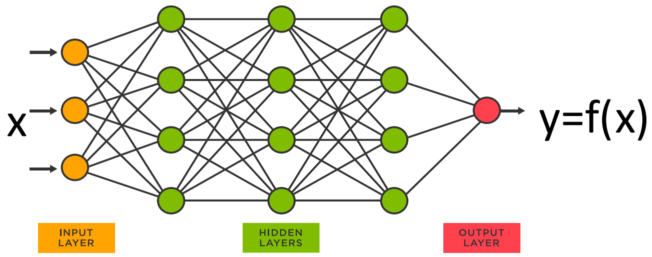
Choice of free parameters

Quick summary

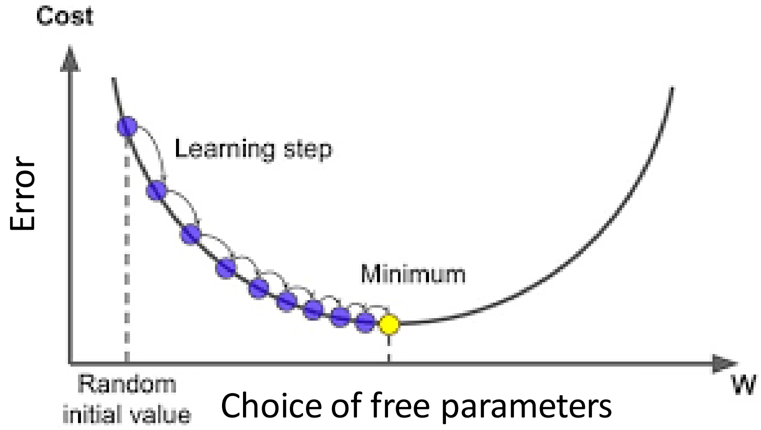
1. Given a labelled data



2. Choose an ML model, initially, with random parameters

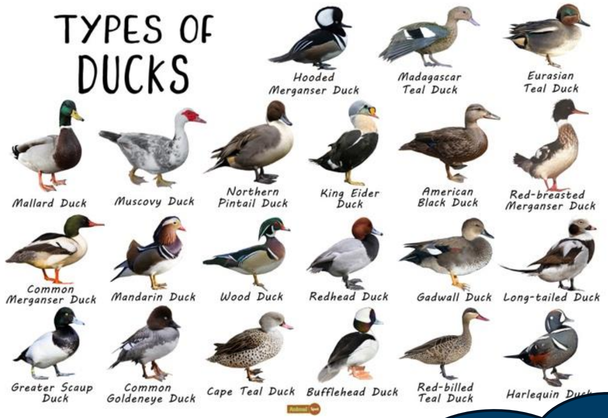


3. Adjust the parameters using gradient descent so the network matches the data

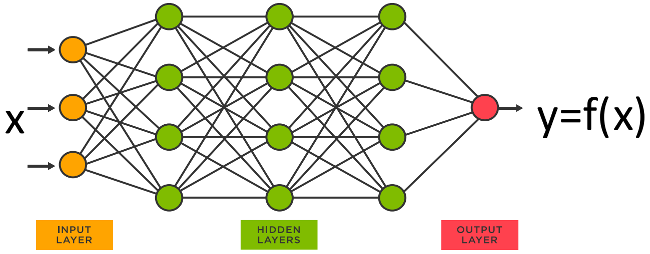


Quick summary

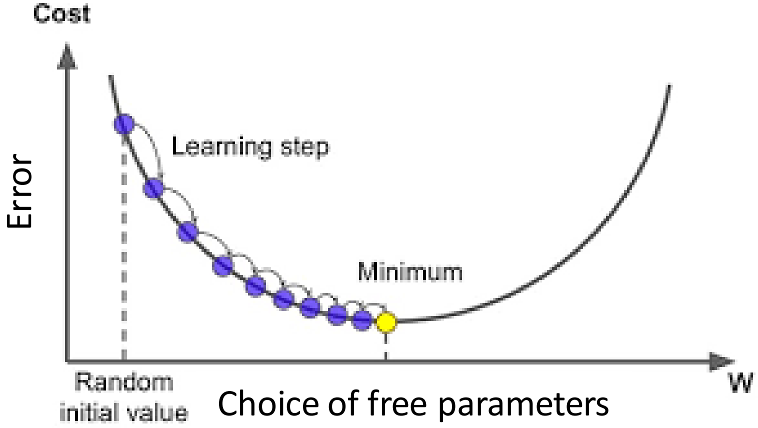
1. Given a labelled data



2. Choose an ML model, initially, with random parameters



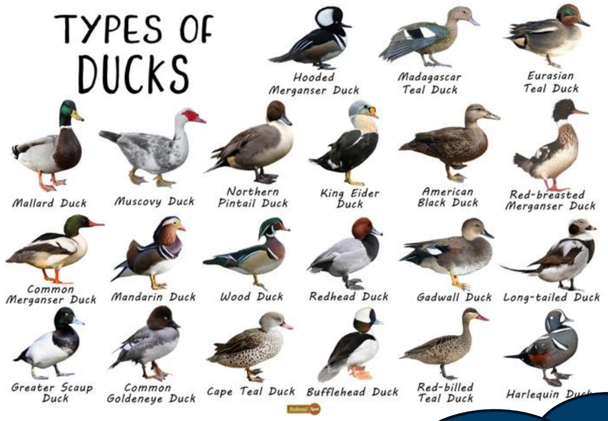
3. Adjust the parameters using gradient descent so the network matches the data



Many ML models are universal approximators!

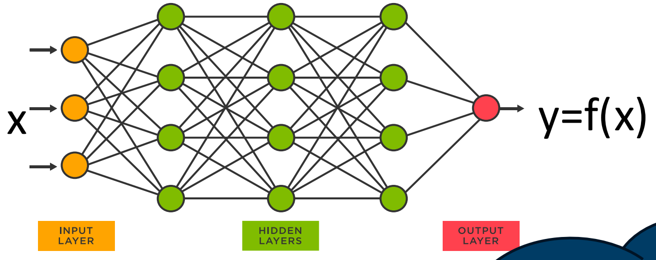
Quick summary

1. Given a labelled data

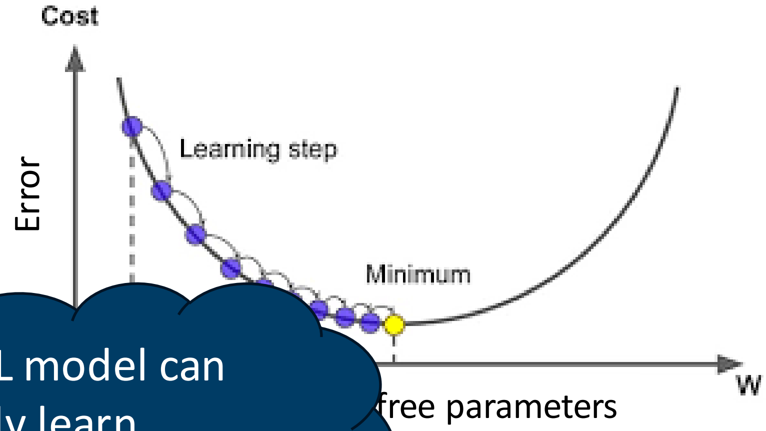


2. Choose an ML model, initially, with random parameters

f = the neural network



3. Adjust the parameters using gradient descent so the network matches the data

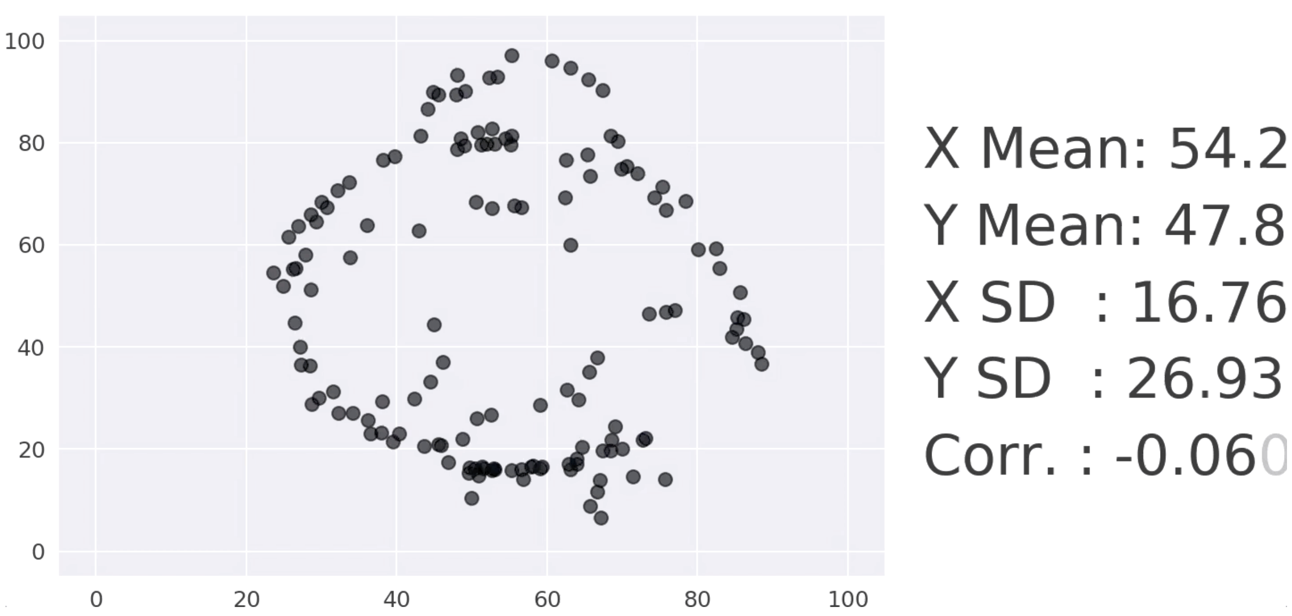


Many ML models are universal approximators!

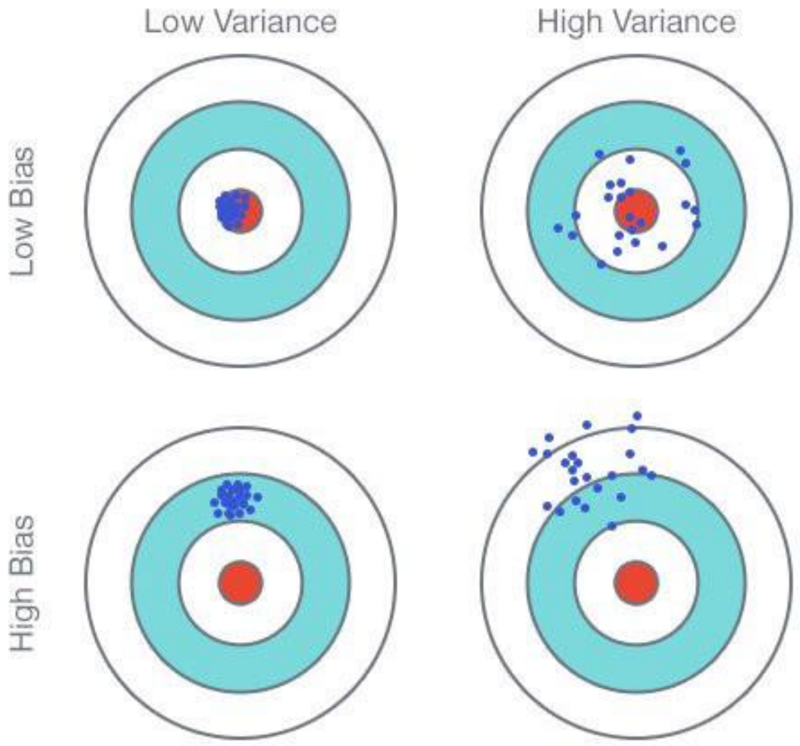
Your ML model can only learn information already in the data!

Data and feature engineering

- Your ML model can only learn information already in the data!
- Data cleaning, feature engineering/selection can have a bigger effect on model performance



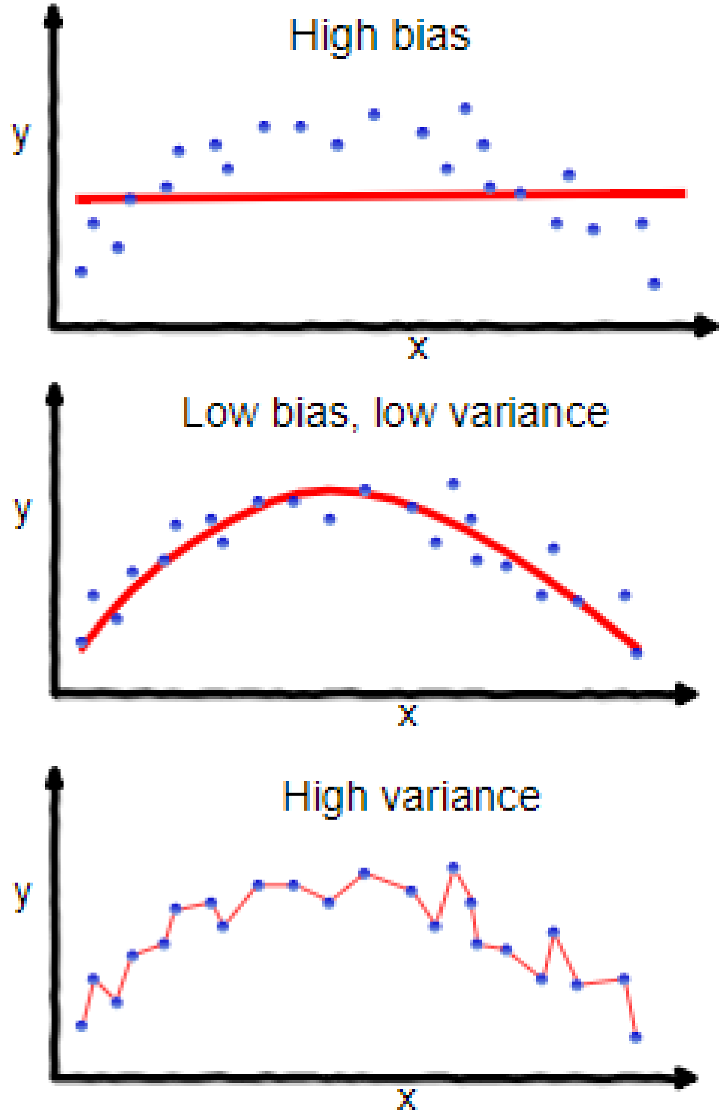
Bias-variance trade off



Under fitting

Good balance

Over fitting



An example of bias

Predicted: Wolf
True: Wolf

Predicted: Husky
True: Husky

Predicted: Wolf
True: Wolf

Predicted: Wolf
True: Wolf

Predicted: Husky
True: Husky

Predicted: Wolf
True: Husky

Source: Ribeiro, Singh, Guestrin, (2016); *"Why Should I Trust You?" Explaining the Predictions of Any Classifier*

An example of bias

Predicted: Wolf
True: Wolf

Predicted: Husky
True: Husky

Predicted: Wolf
True: Wolf

Predicted: Wolf
True: Wolf

Predicted: Husky
True: Husky

Predicted: Wolf
True: Husky

Source: Ribeiro, Singh, Guestrin, (2016); *"Why Should I Trust You?" Explaining the Predictions of Any Classifier*

An example of bias

Predicted: Wolf
True: Wolf

Predicted: Husky
True: Husky

Predicted: Wolf
True: Wolf

Predicted: Wolf
True: Wolf

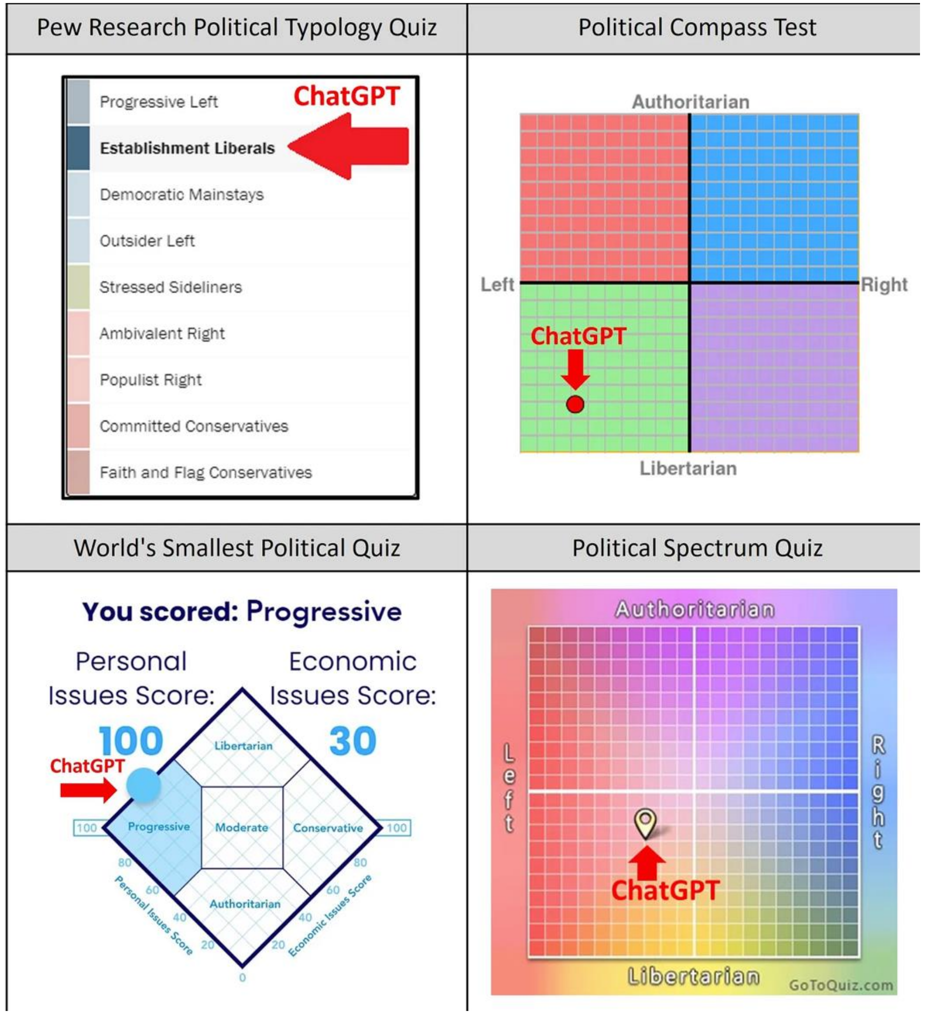
Predicted: Husky
True: Husky

Predicted: Wolf
True: Husky

Explainable AI!!

Source: Ribeiro, Singh, Guestrin, (2016); *"Why Should I Trust You?" Explaining the Predictions of Any Classifier*

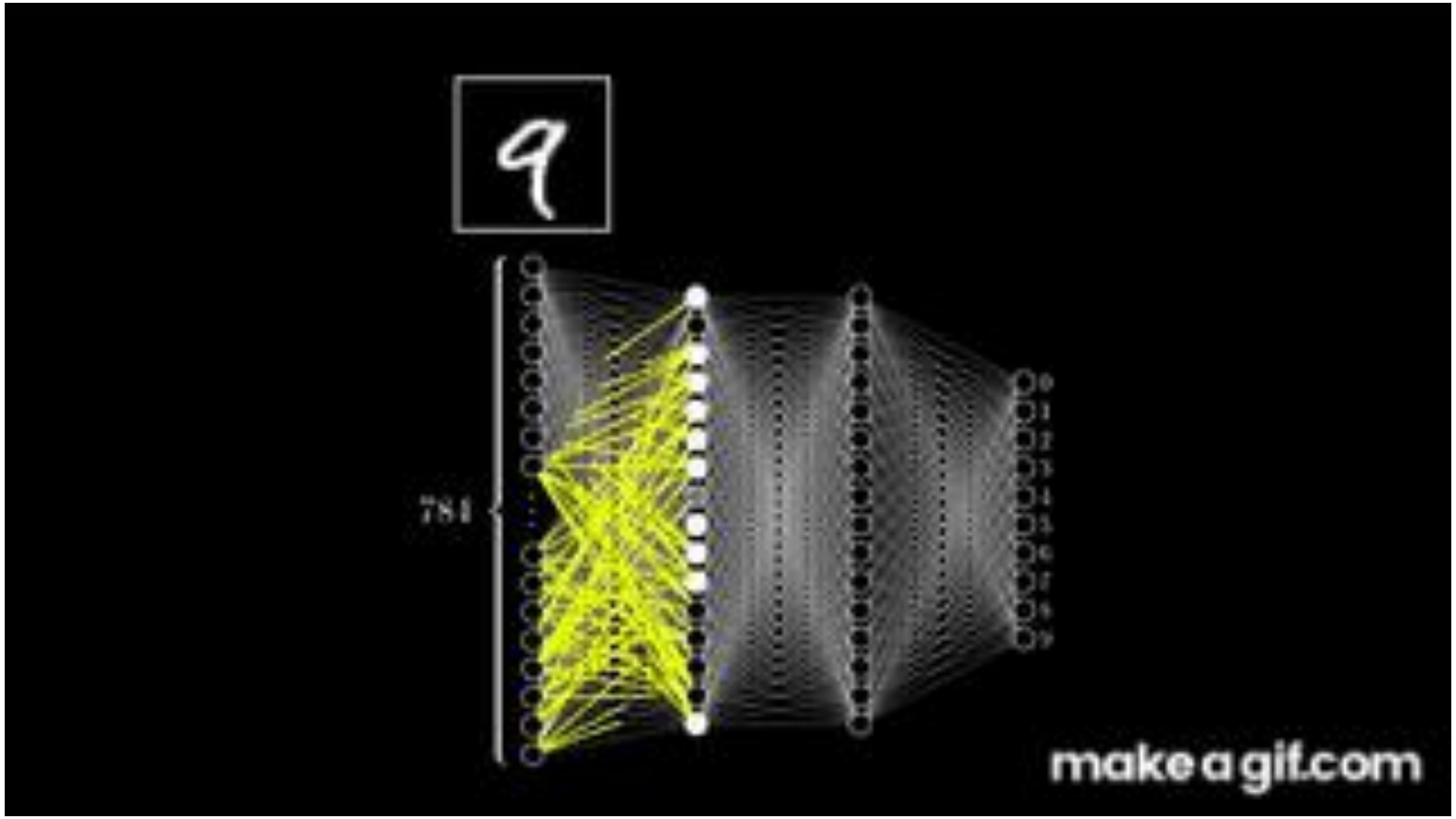
Bias in ChatGPT



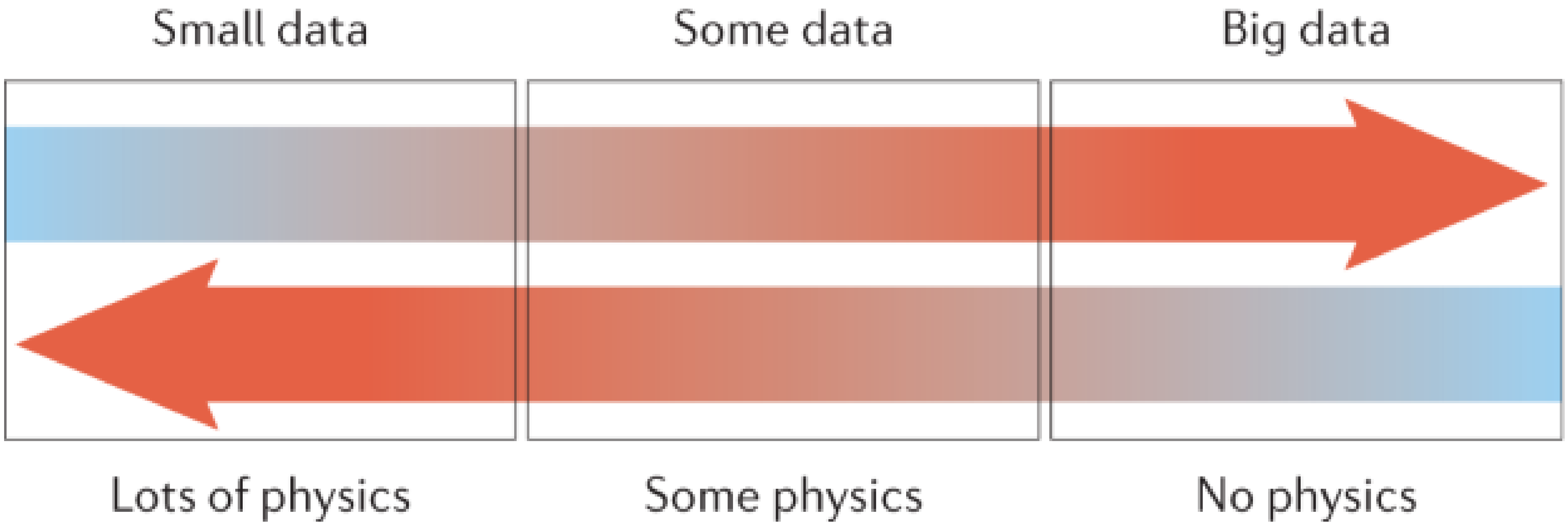
SINTEF

Deep learning

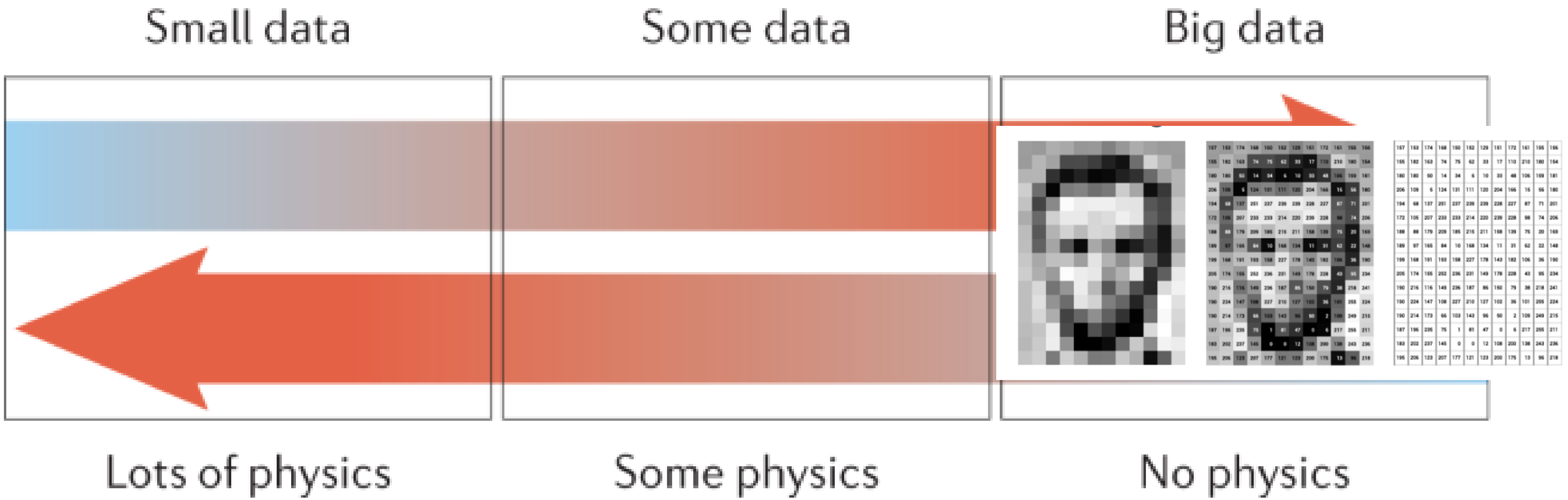
Deeper neural nets allows them to learn more complex functions!



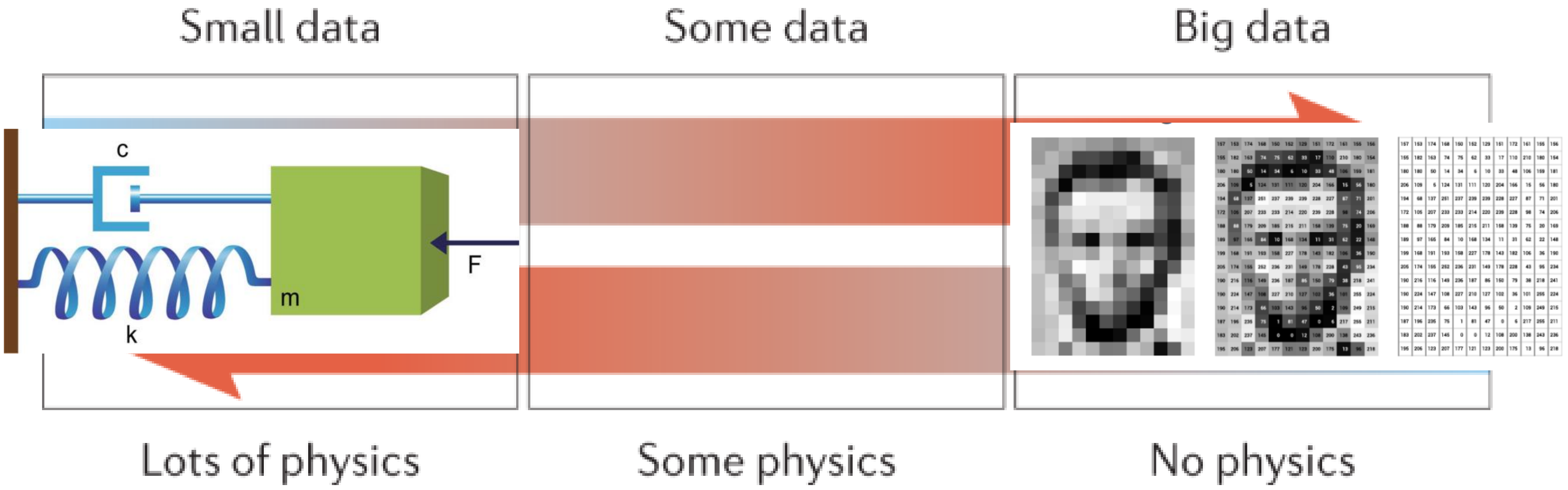
Hybrid-machine learning



Hybrid-machine learning

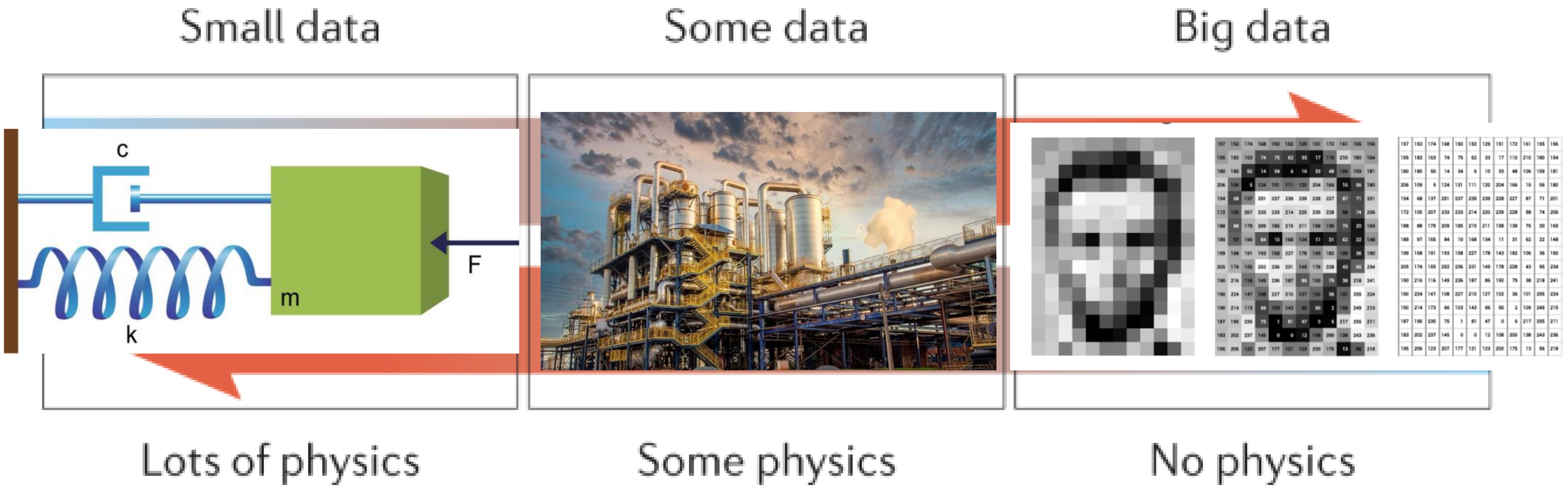


Hybrid-machine learning

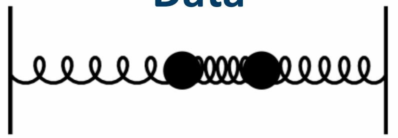


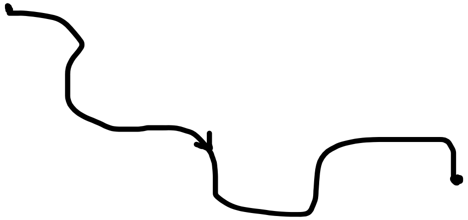
SINTEF

Hybrid-machine learning

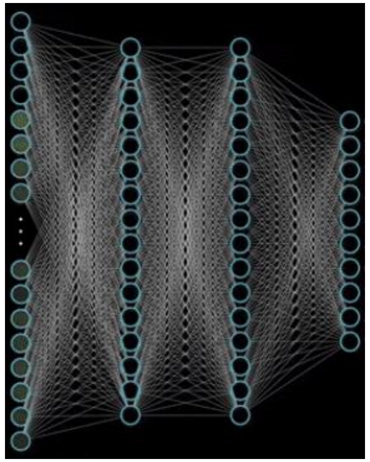


Data

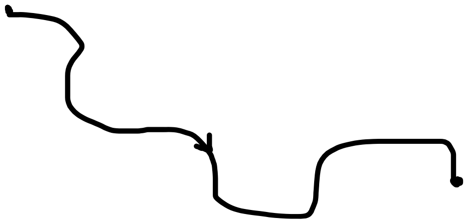


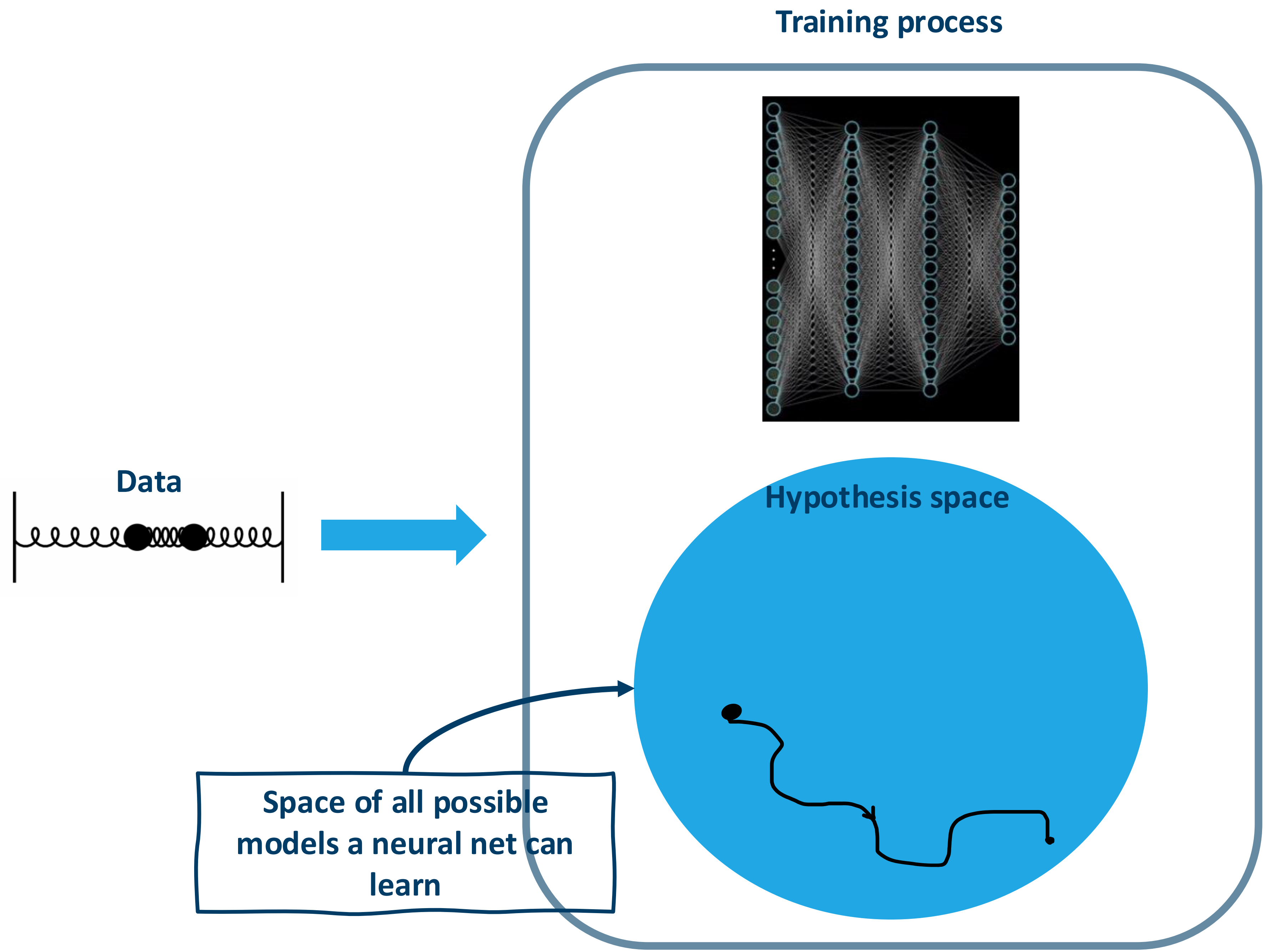


Training process

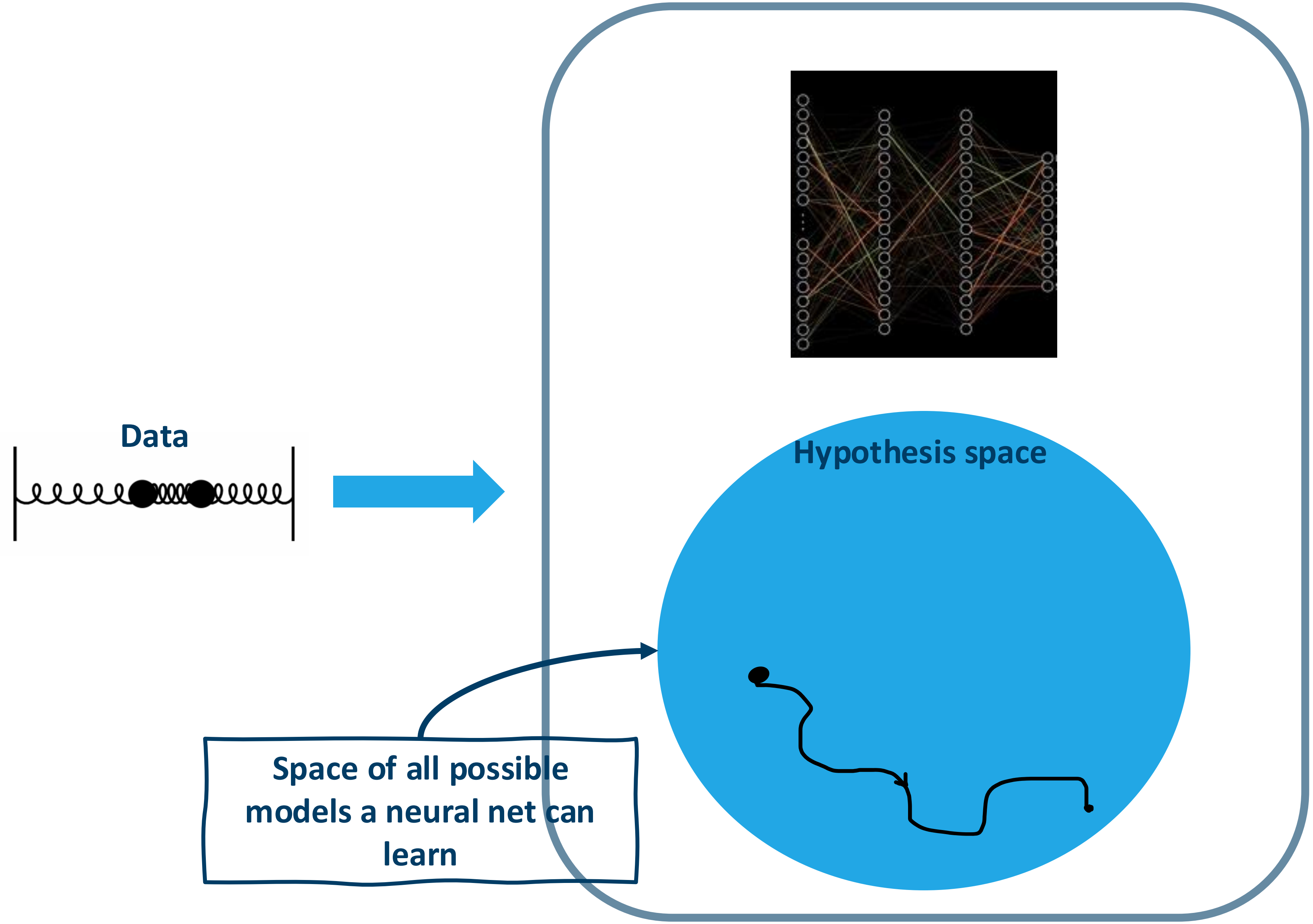


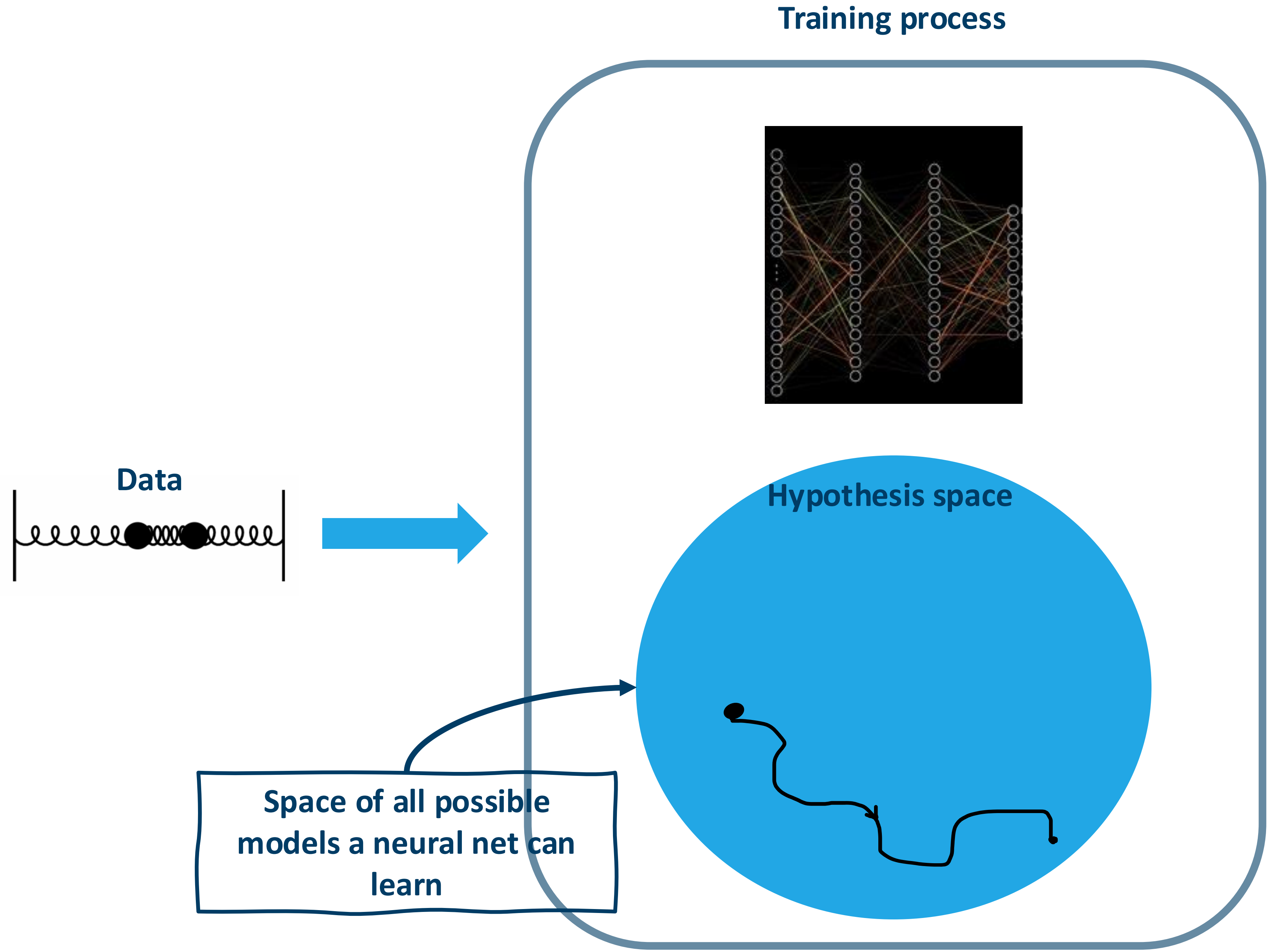
Data

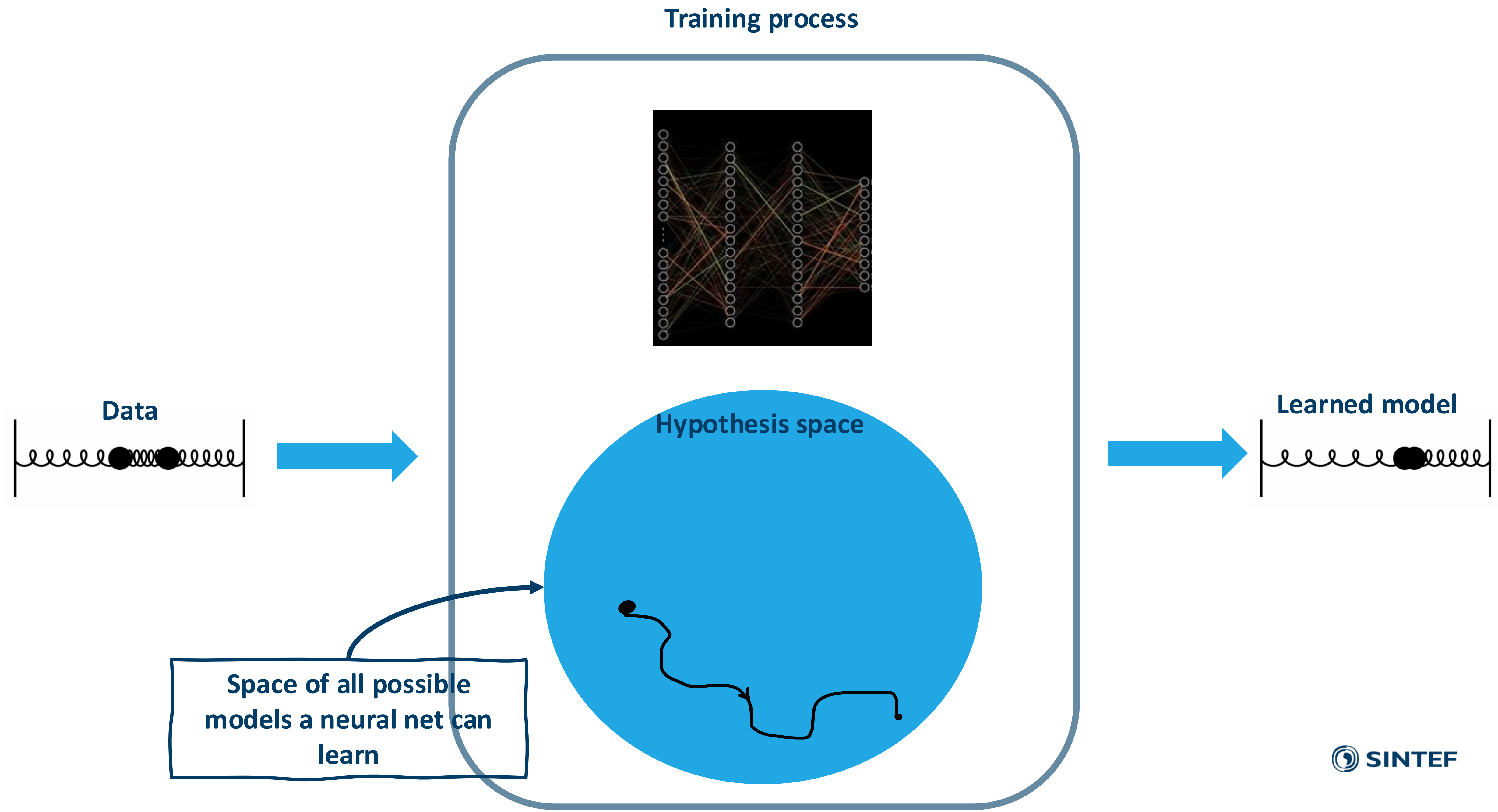


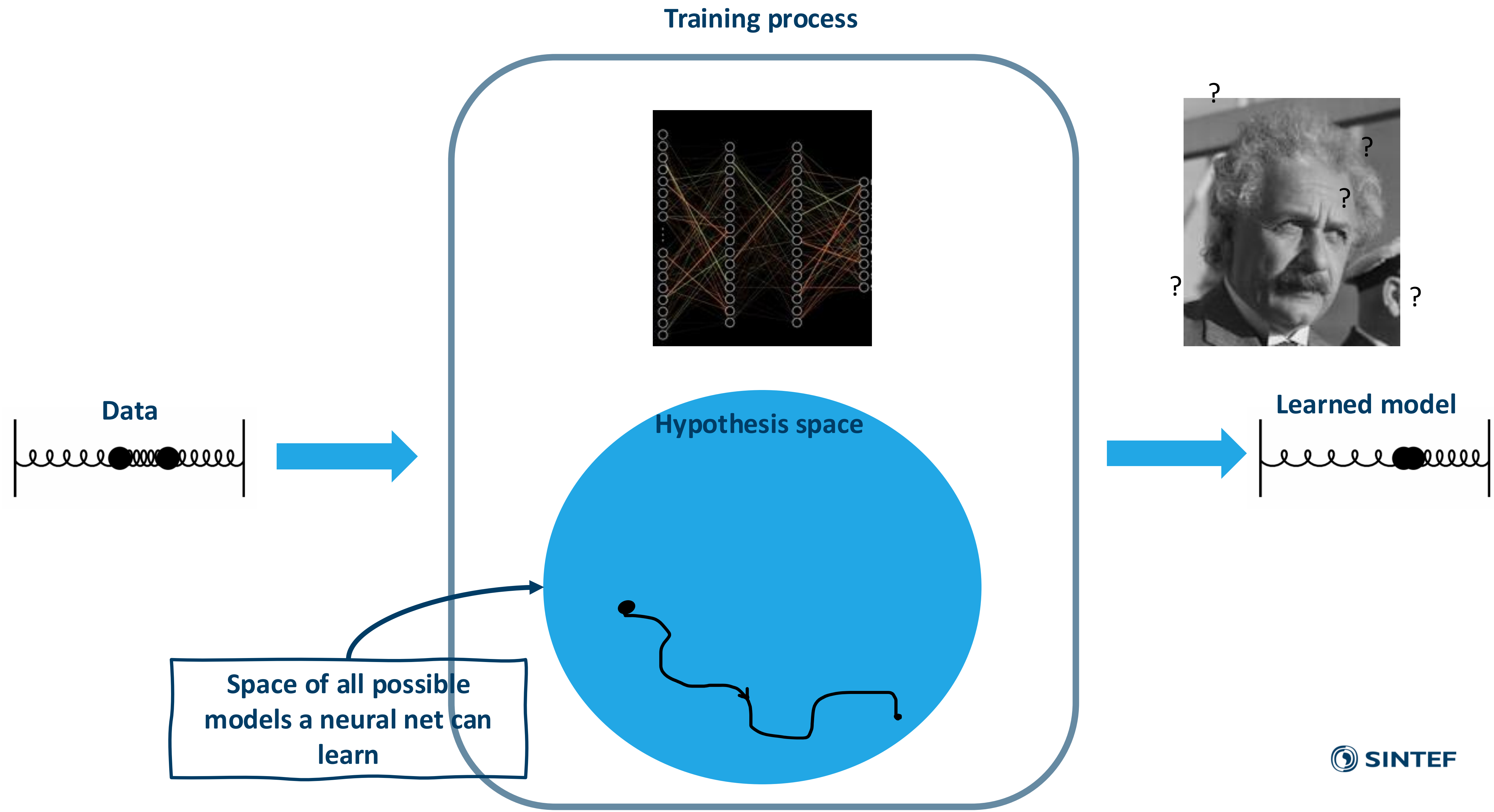

Training process
Data
Hypothesis space
Space of all possible models a neural net can learn

Training process

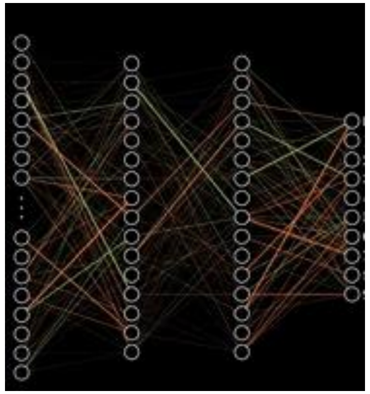


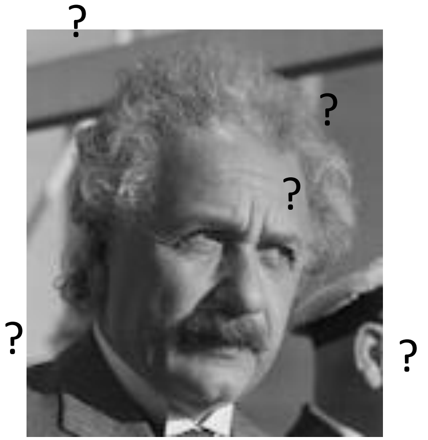

Training process
Data
Hypothesis space
Space of all possible models a neural net can learn


Training process
Data
Hypothesis space
Learned model
Space of all possible models a neural net can learn

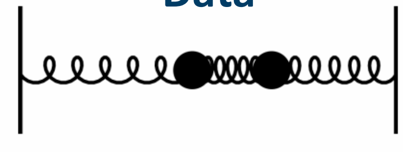

Training process
Data
Hypothesis space
Learned model
?
?
?
?
?
Space of all possible models a neural net can learn

Training process

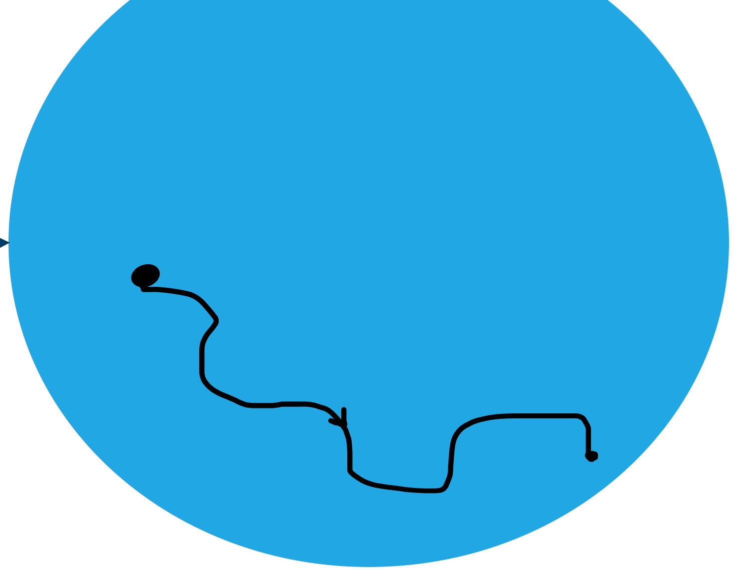




Data

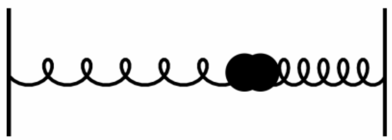


Hypothesis space



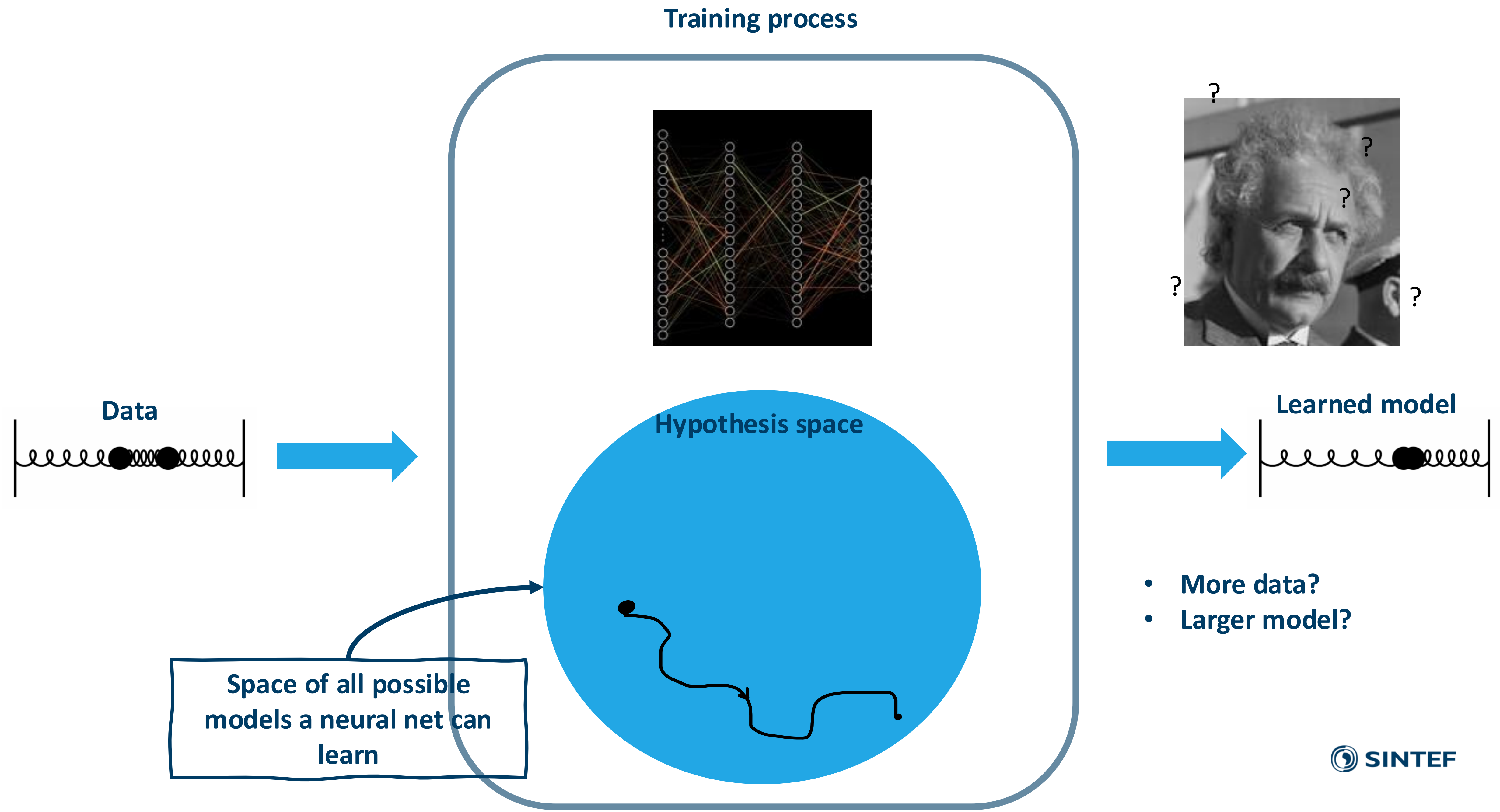
Space of all possible models a neural net can learn

Learned model



- More data?

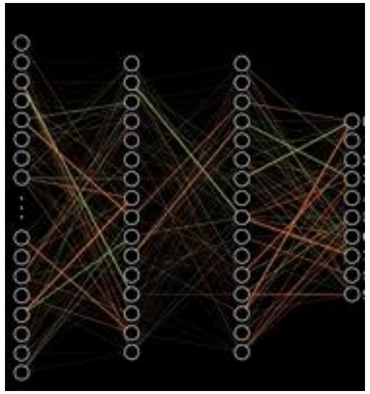
SINTEF

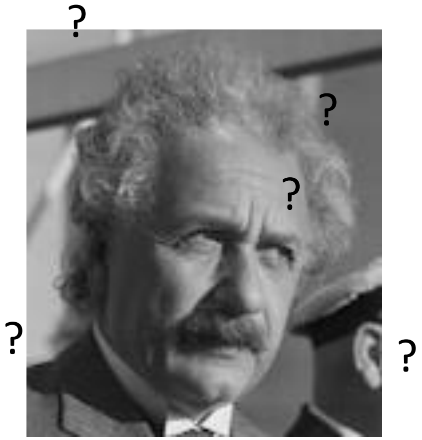


Training process
Data
Hypothesis space
Learned model
Space of all possible models a neural net can learn
?
More data?
Larger model?

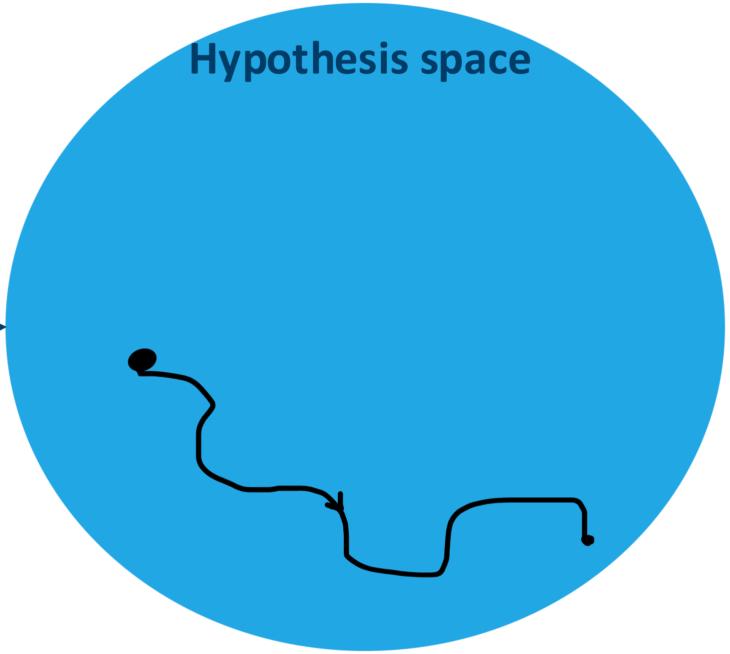
SINTEF

Training process



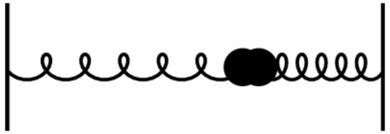


Data



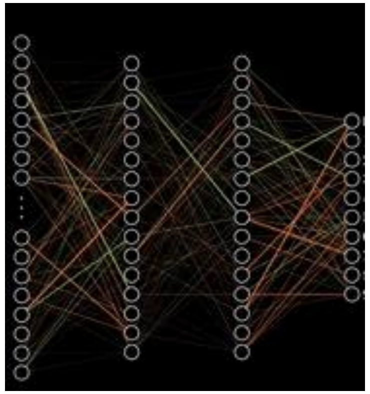
Space of all possible models a neural net can learn

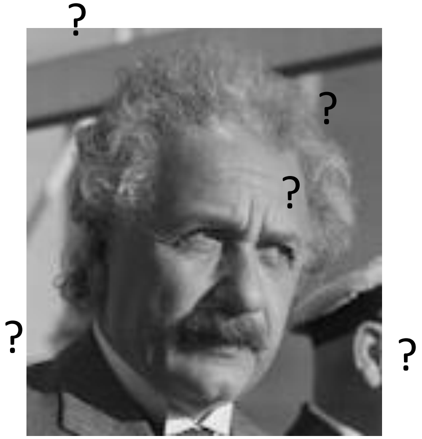
Learned model



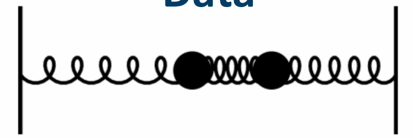
- More data?
- Larger model?
- More compute?

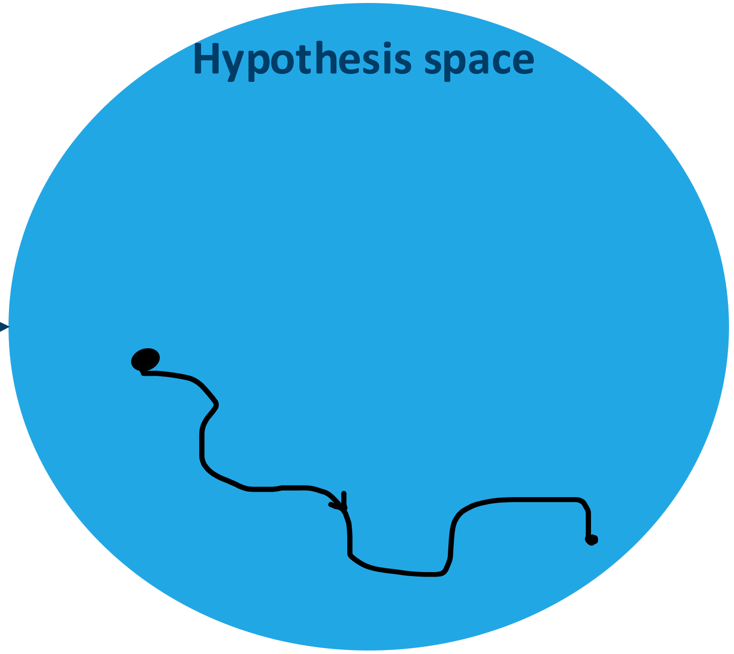
Training process



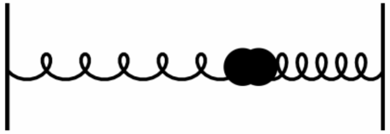


Data





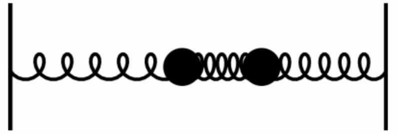
Learned model

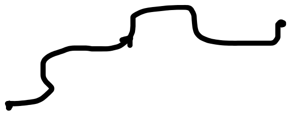


Space of all possible models a neural net can learn

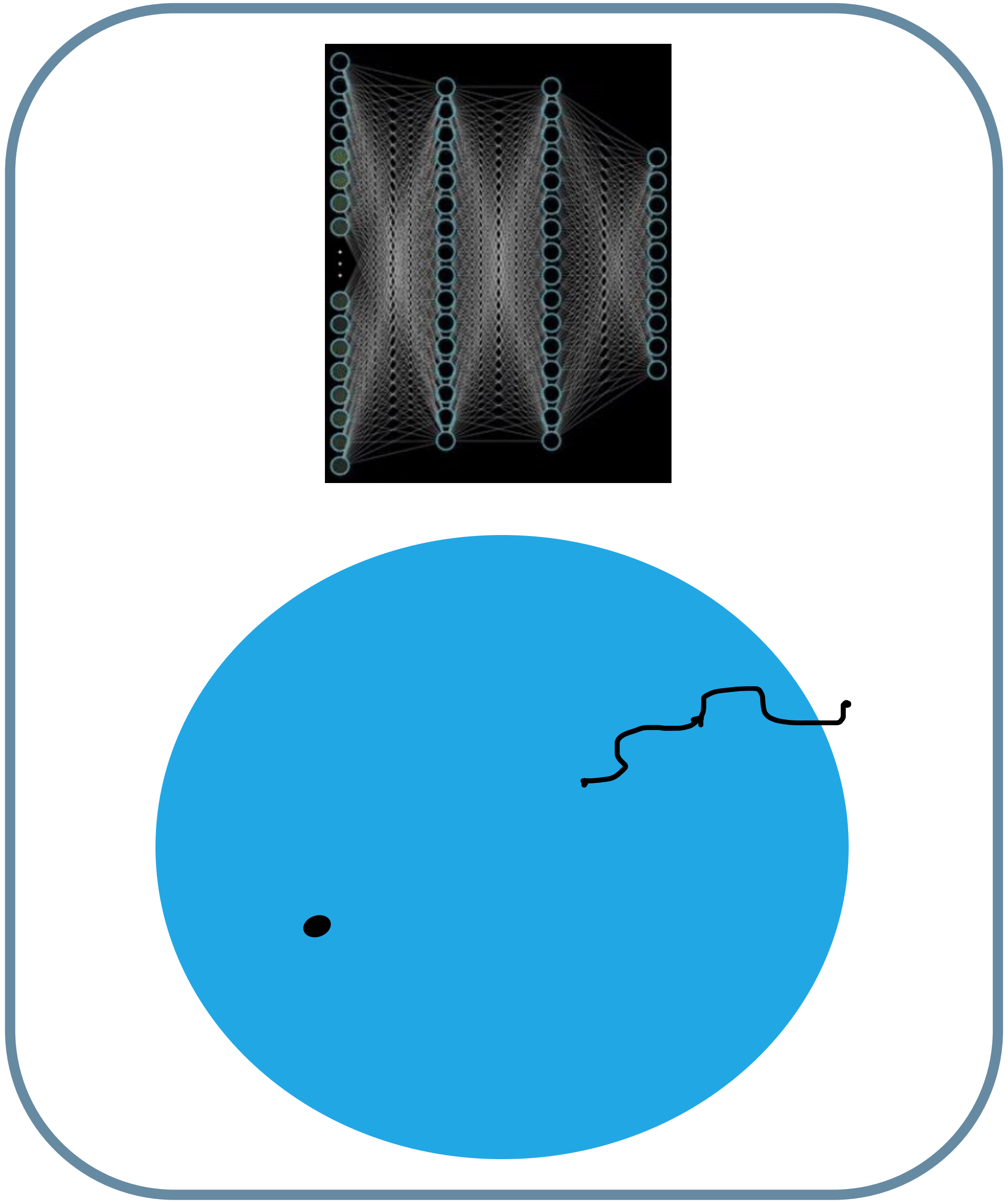
- More data?
- Larger model?
- More compute?
- More data scientists?

Data

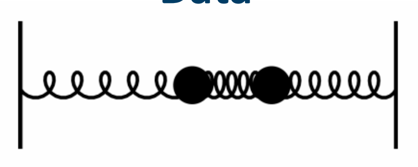




Training process



Data

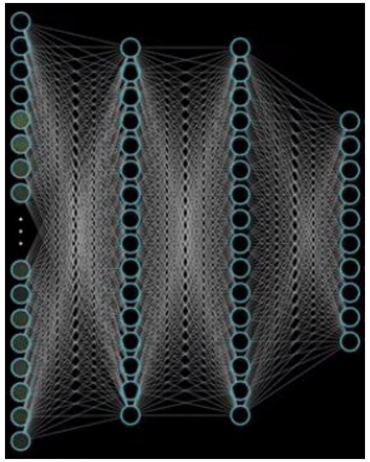


$$\delta \int_{t_1}^{t_2} L\, dt = 0$$

Training process $\dot{x} = (S(x) - R(x))\nabla H(x) + f(x,t)$

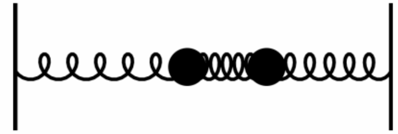
$$\dot{q}_i = \frac{\partial H}{\partial p_i}$$

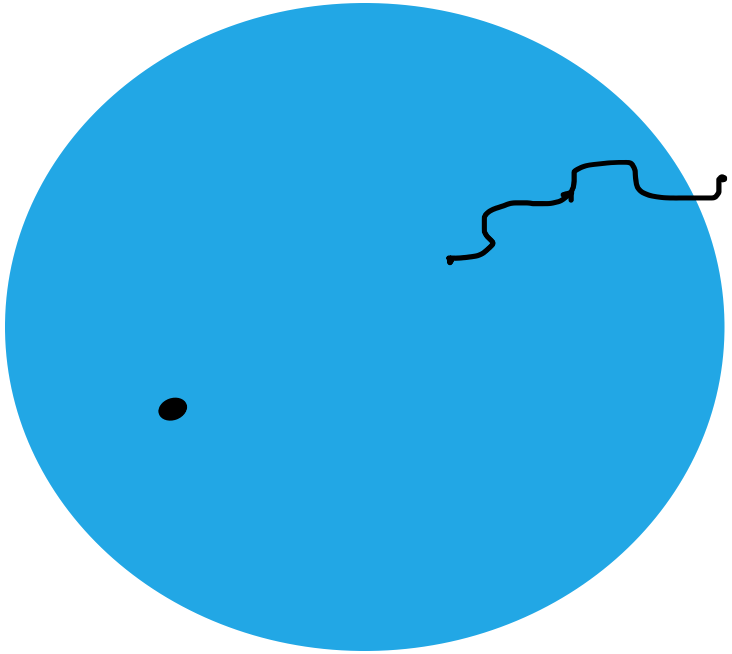
$$\dot{p}_i = -\frac{\partial H}{\partial q_i}$$



$$\omega = \sum_{i=1}^{n} dq_i \wedge dp_i,$$

Data





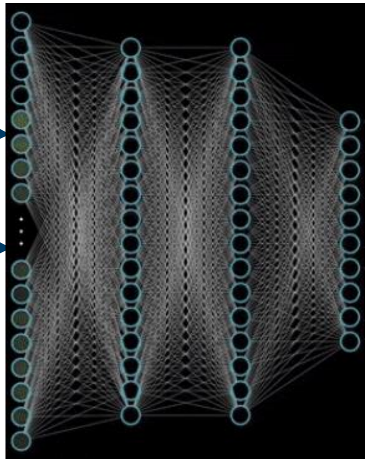
$$\delta \int_{t_1}^{t_2} L\,dt = 0$$

Training process $\dot{x} = (S(x) - R(x))\nabla H(x) + f(x,t)$

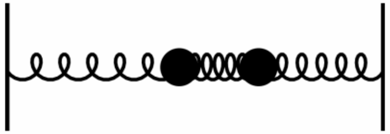
$$\dot{q}_i = \frac{\partial H}{\partial p_i}$$

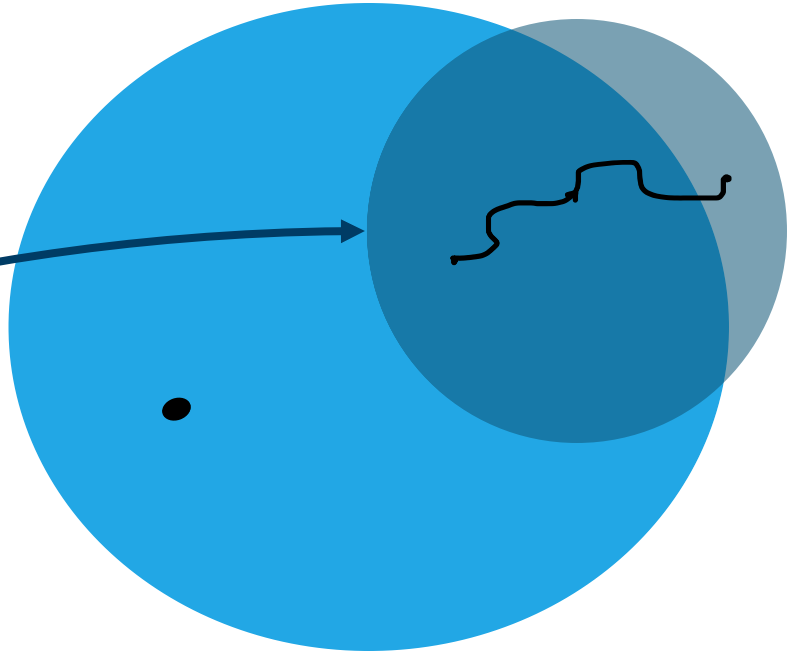
$$\dot{p}_i = -\frac{\partial H}{\partial q_i}$$

$\omega = \sum_{i=1}^{n} dq_i \wedge dp_i,$

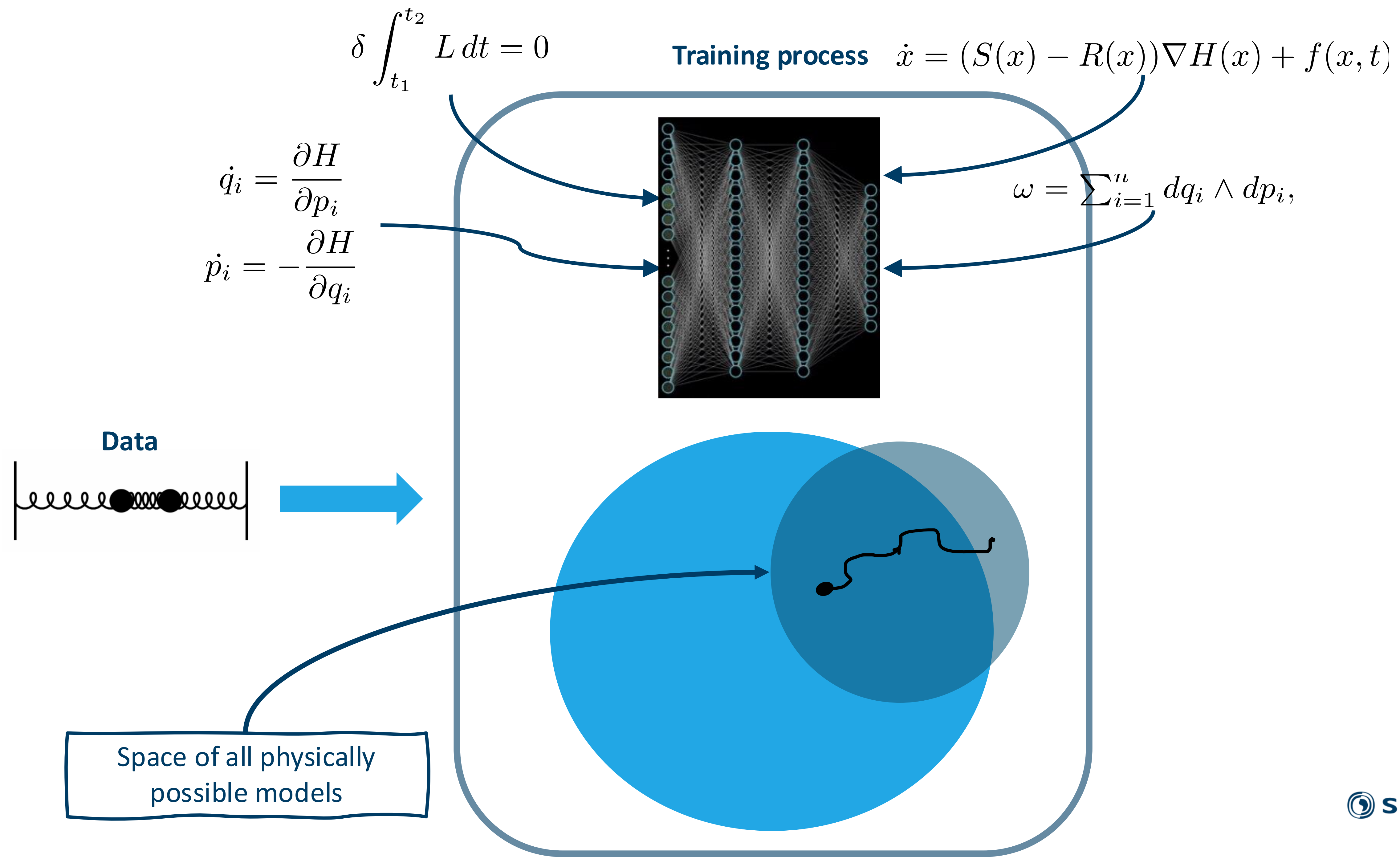


Data





Space of all physically possible models



$\delta \int_{t_1}^{t_2} L \, dt = 0$
Training process
$\dot{x} = (S(x) - R(x))\nabla H(x) + f(x,t)$
$\dot{q}_i = \frac{\partial H}{\partial p_i}$
$\dot{p}_i = -\frac{\partial H}{\partial q_i}$
$\omega = \sum_{i=1}^{n} dq_i \wedge dp_i,$
Data
Space of all physically possible models

$$\delta \int_{t_1}^{t_2} L\, dt = 0$$

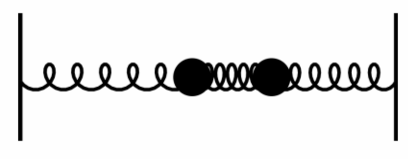
Training process $\dot{x} = (S(x) - R(x))\nabla H(x) + f(x,t)$

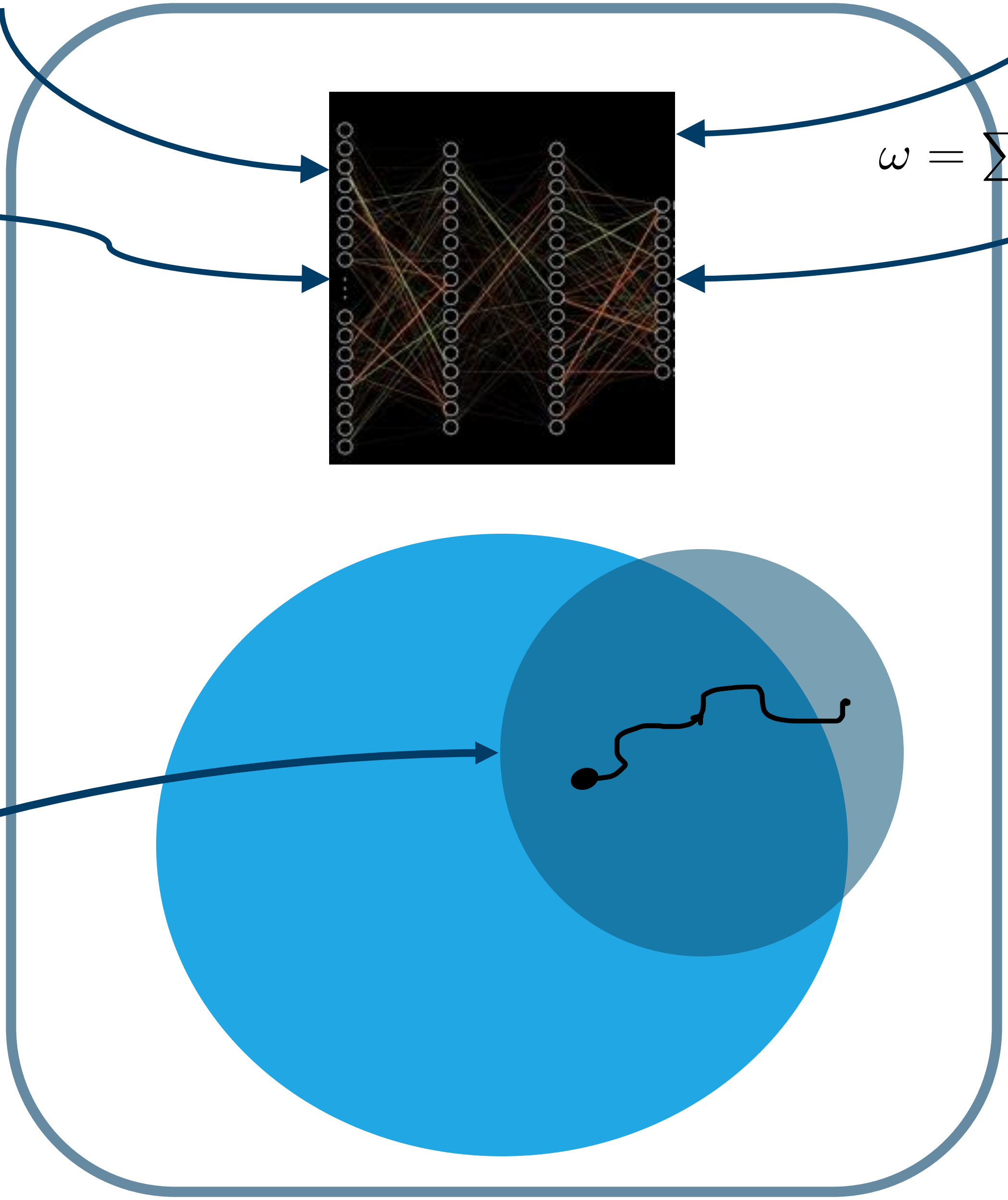
$$\dot{q}_i = \frac{\partial H}{\partial p_i}$$

$$\dot{p}_i = -\frac{\partial H}{\partial q_i}$$

$$\omega = \sum_{i=1}^{n} dq_i \wedge dp_i,$$

Data





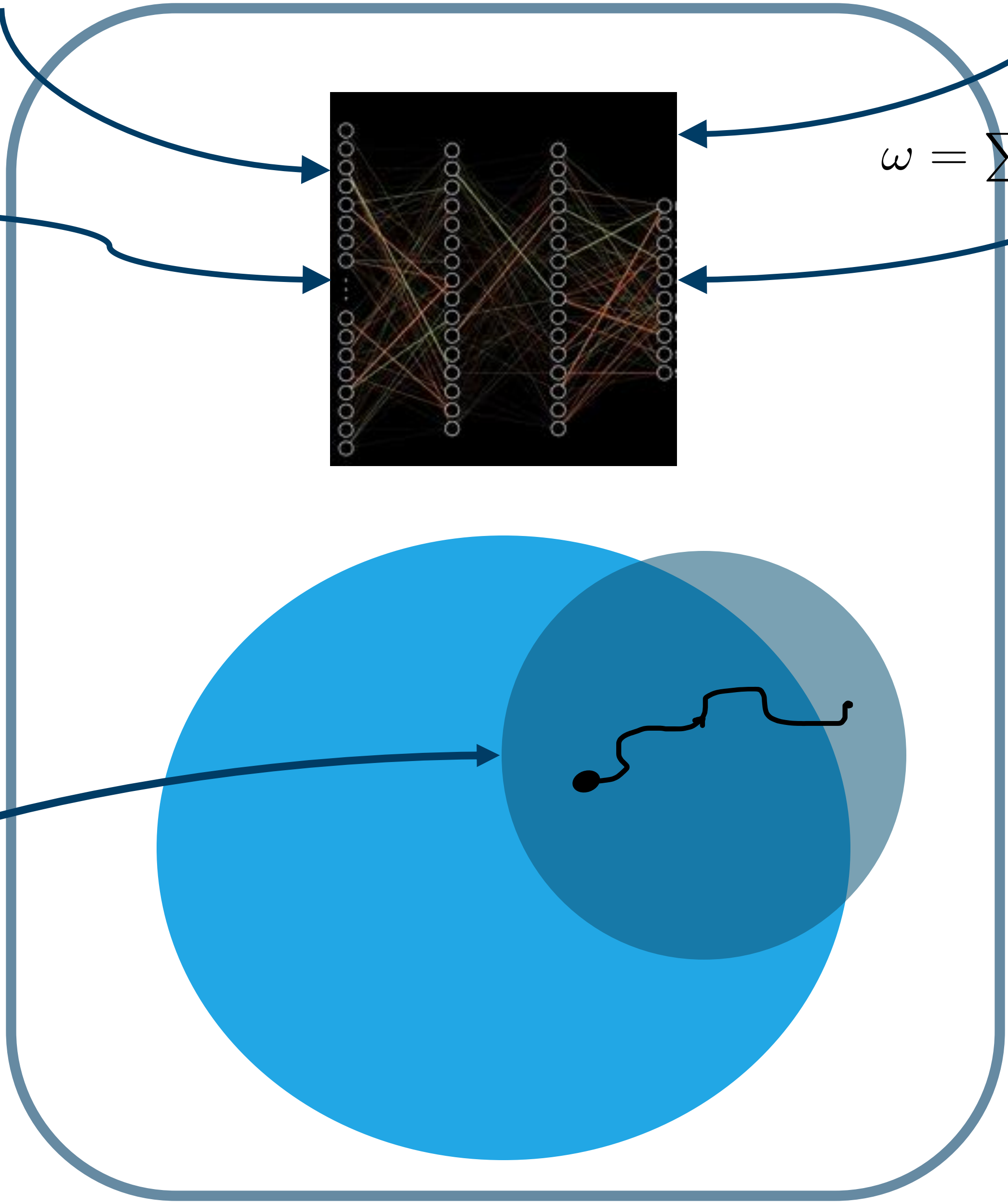
Space of all physically possible models

$$\delta \int_{t_1}^{t_2} L \, dt = 0$$

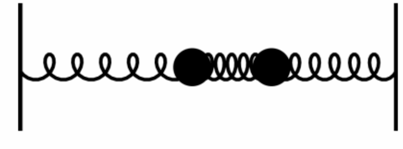
Training process $\dot{x} = (S(x) - R(x))\nabla H(x) + f(x,t)$

$$\dot{q}_i = \frac{\partial H}{\partial p_i}$$
$$\dot{p}_i = -\frac{\partial H}{\partial q_i}$$

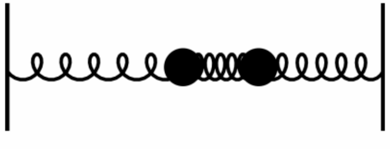
$$\omega = \sum_{i=1}^{n} dq_i \wedge dp_i,$$



Data



Learned model



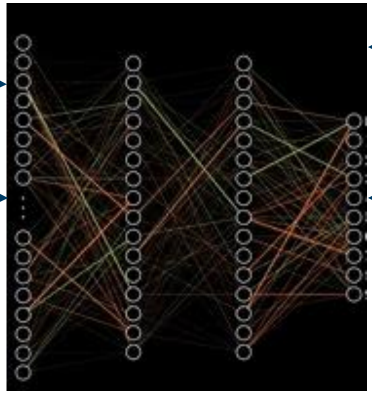
Space of all physically possible models

$$\delta \int_{t_1}^{t_2} L\,dt = 0$$

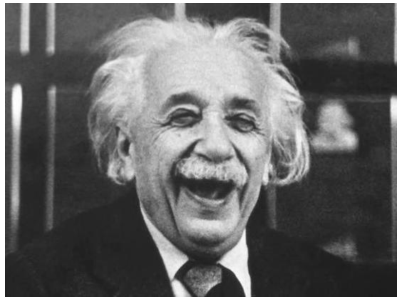
Training process $\dot{x} = (S(x) - R(x))\nabla H(x) + f(x,t)$

$$\dot{q}_i = \frac{\partial H}{\partial p_i}$$

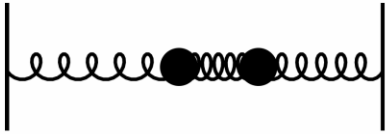
$$\dot{p}_i = -\frac{\partial H}{\partial q_i}$$

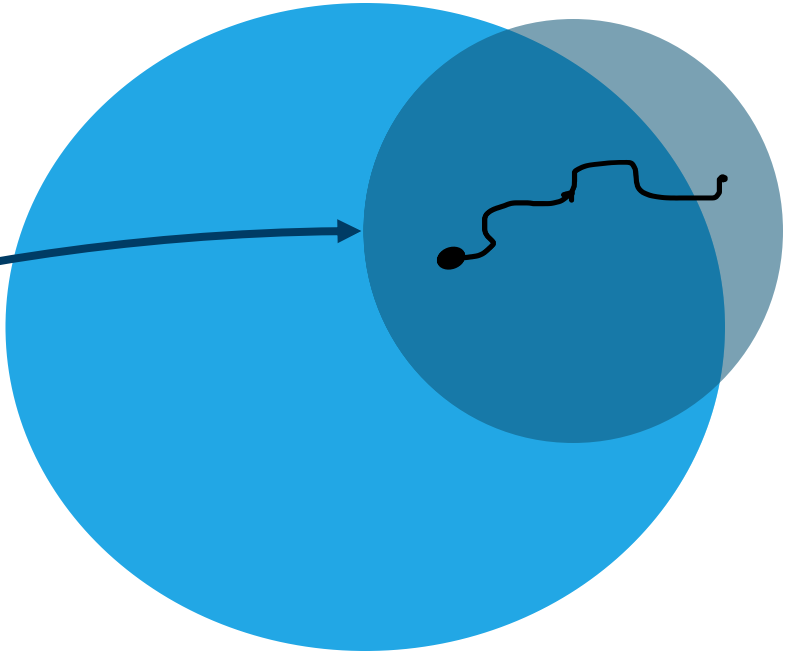


$$\omega = \sum_{i=1}^{n} dq_i \wedge dp_i,$$

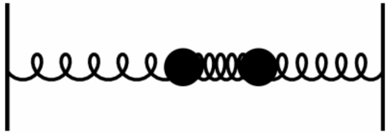


Data





Learned model

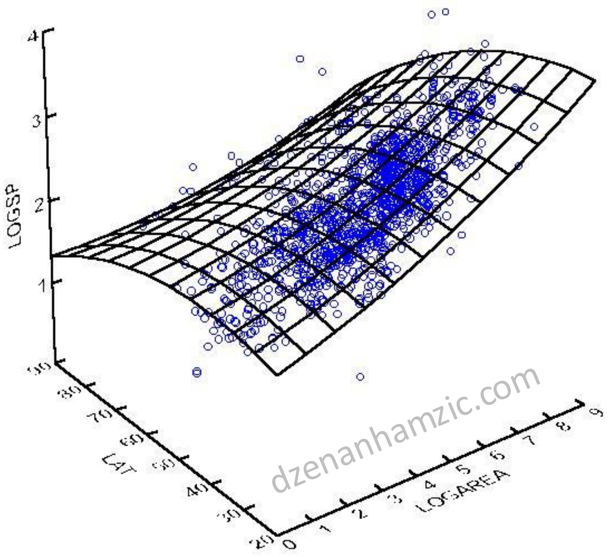


Space of all physically possible models

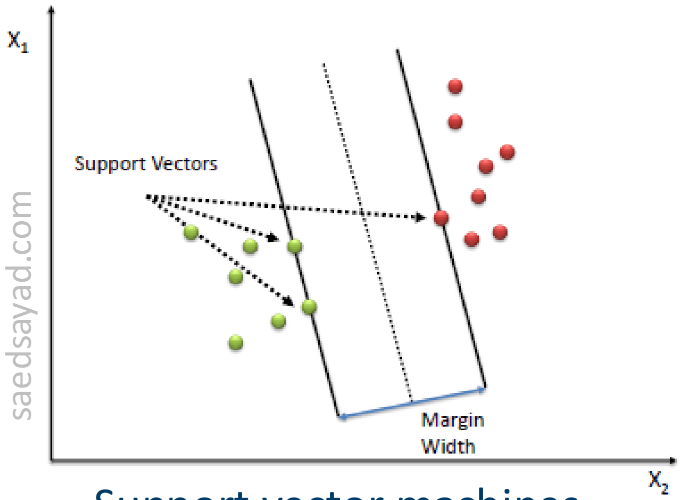
SINTEF

COFFEE BREAK

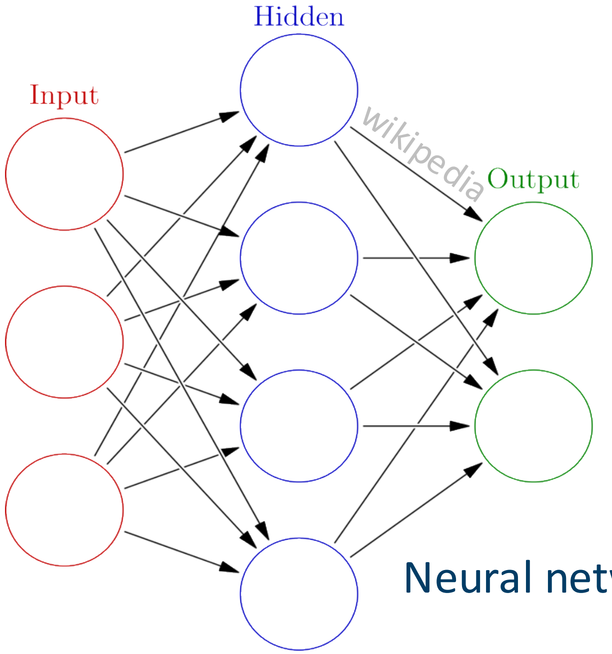
Examples of methods



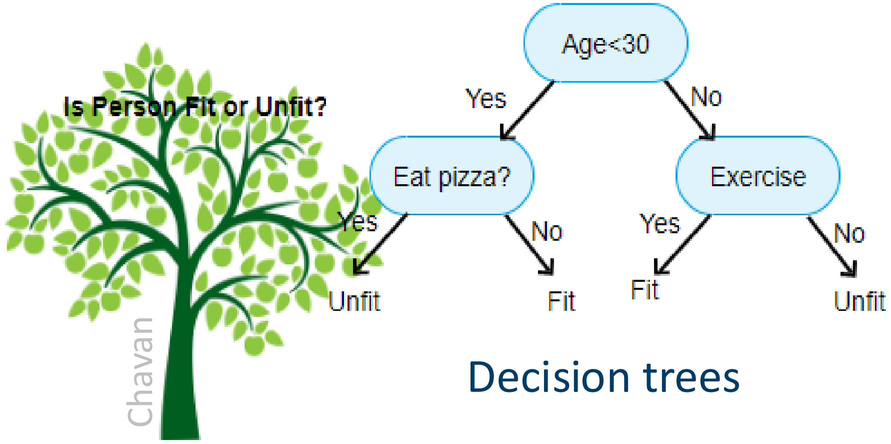
(Non-)linear regression



Support vector machines



Neural networks

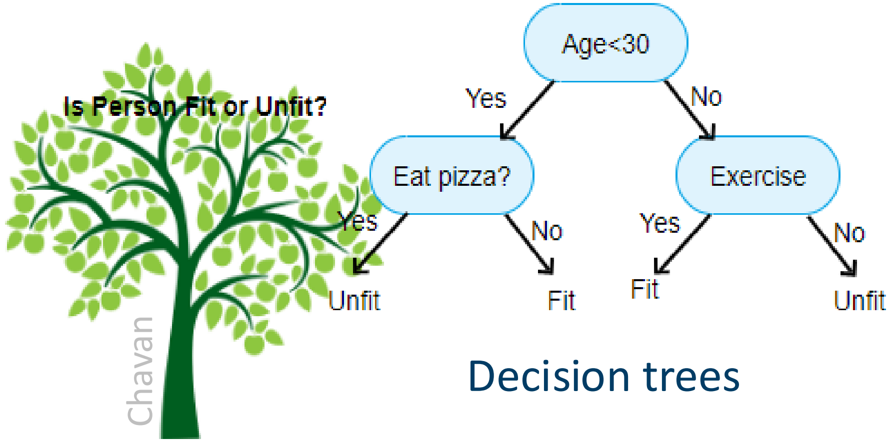


Decision trees

SINTEF

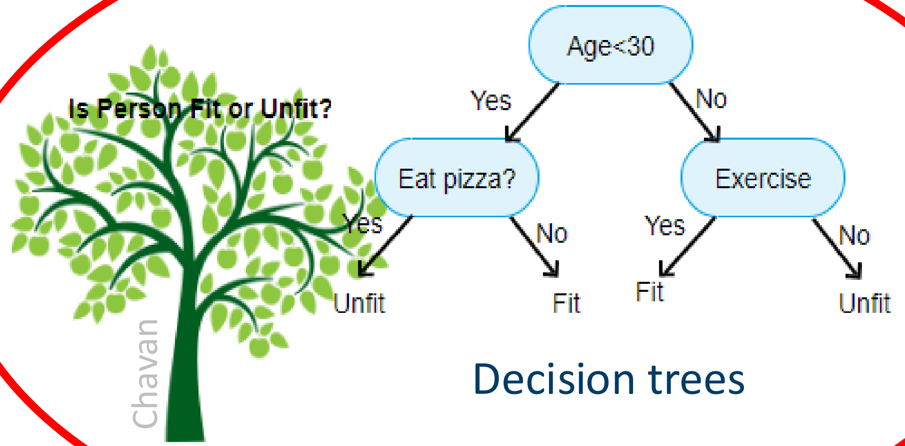
Examples of methods

(Non-)linear regression



Decision trees

Examples of methods

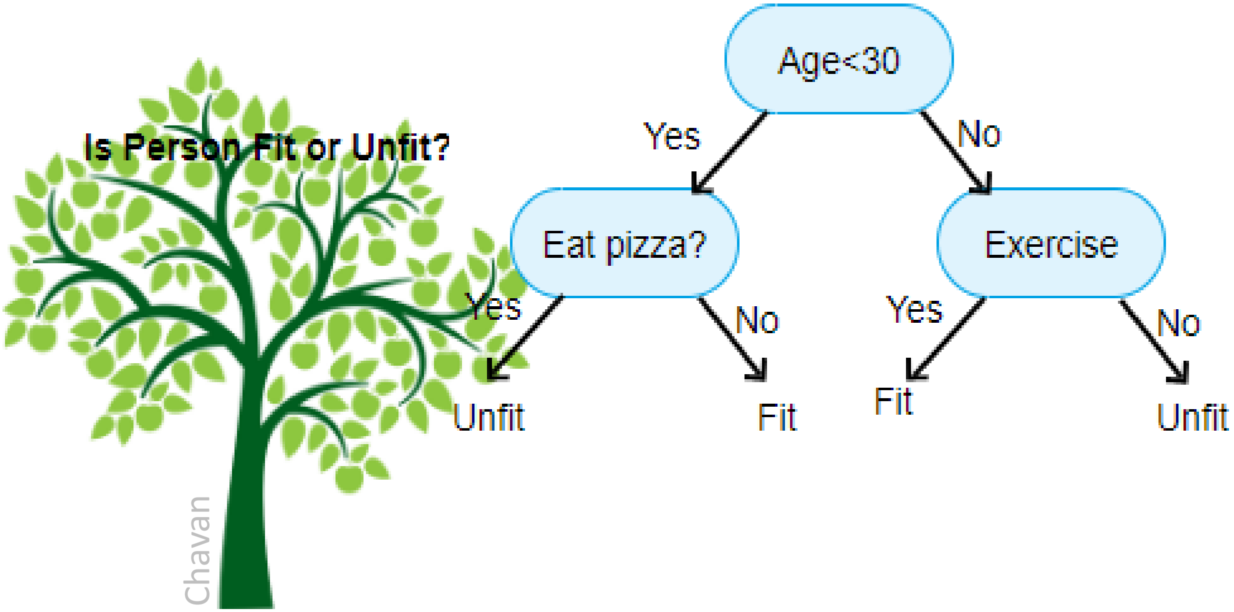


Decision trees

(Non-)linear regression

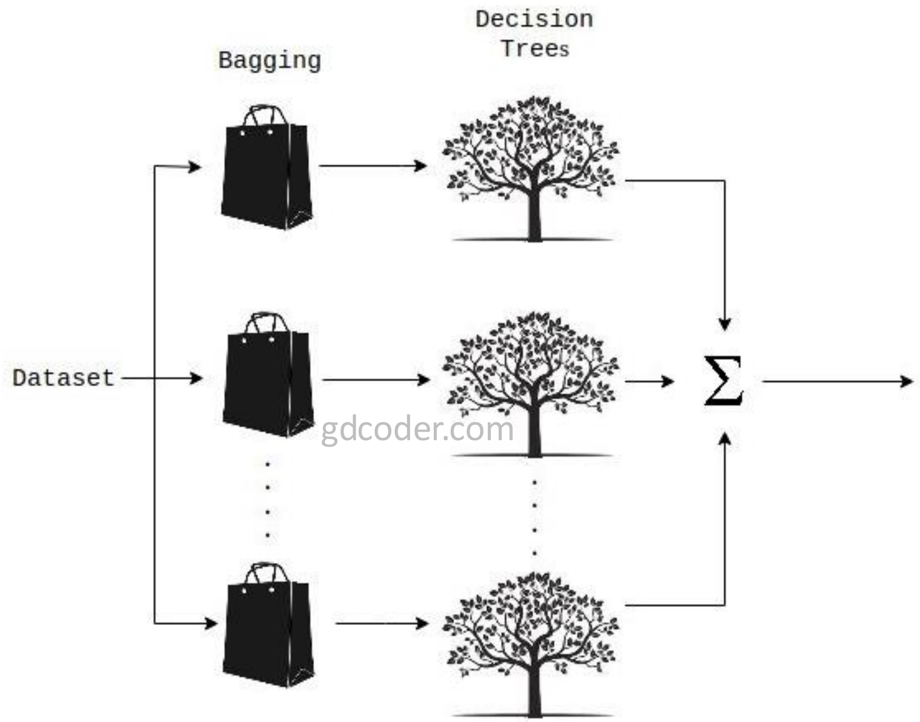
SINTEF

Decision trees

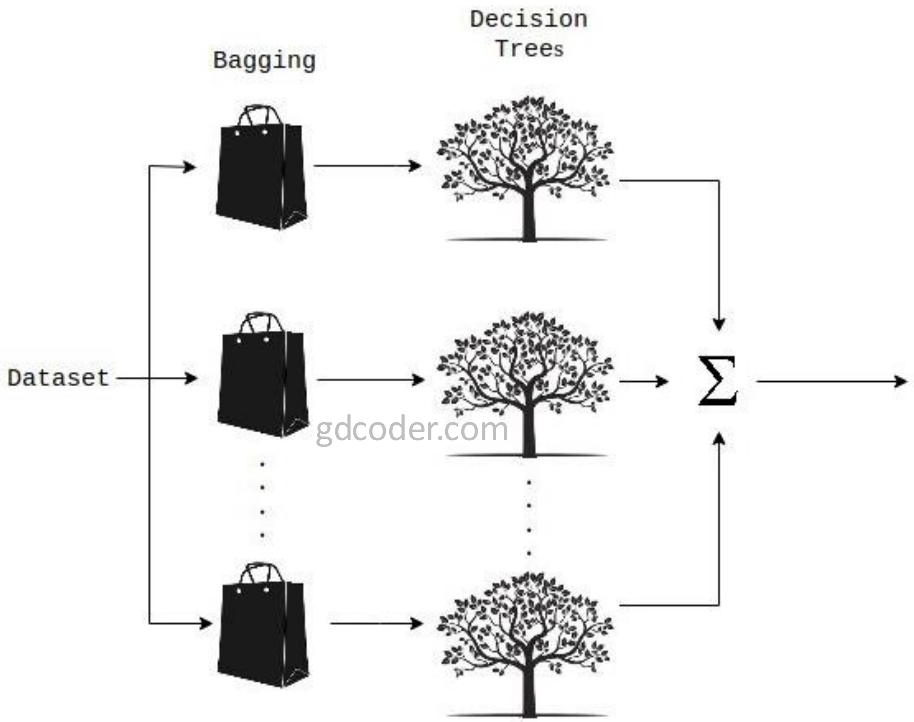


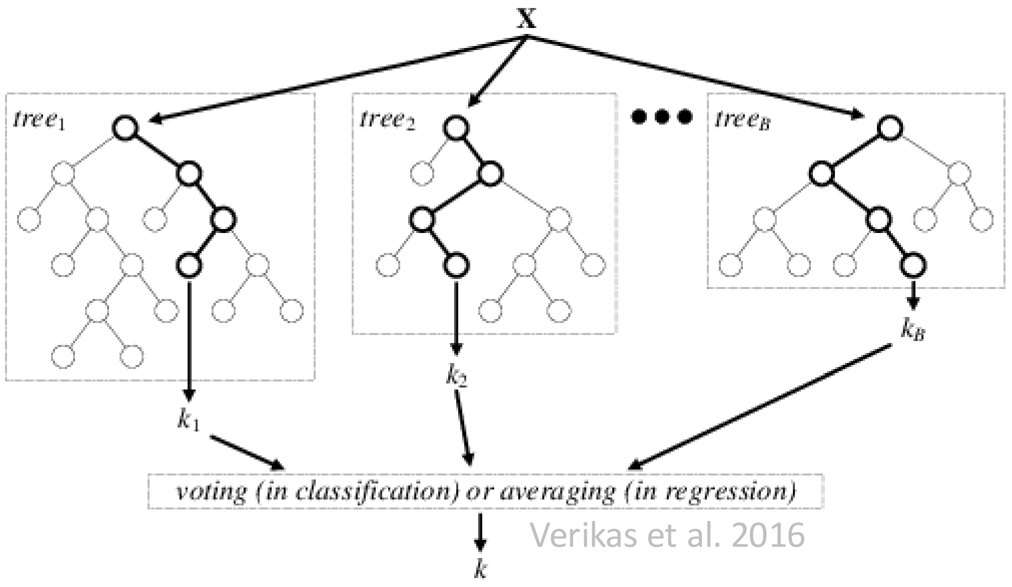
Bagging and random forest

Bagging and random forest



Bagging and random forest

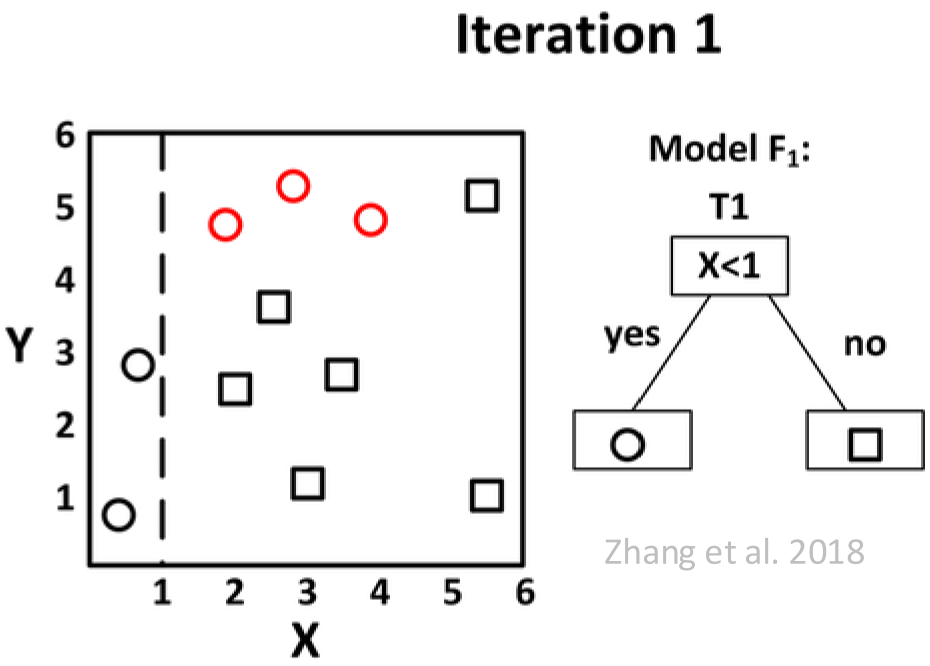




Verikas et al. 2016

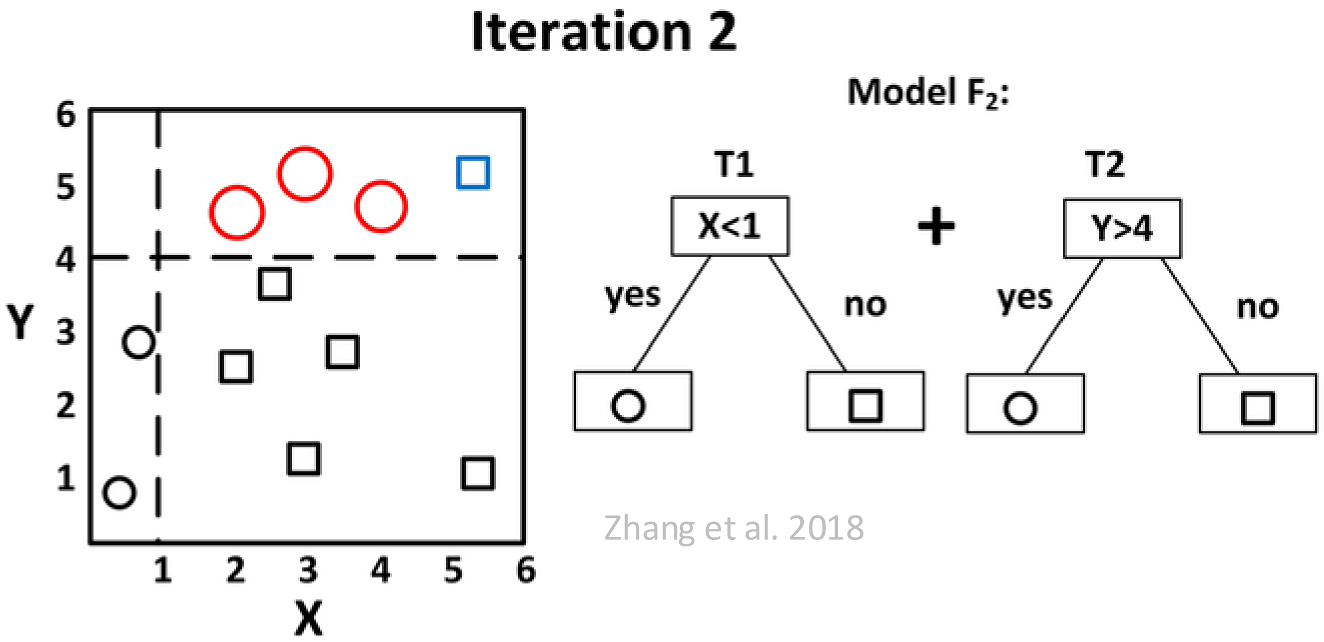
SINTEF

Boosting



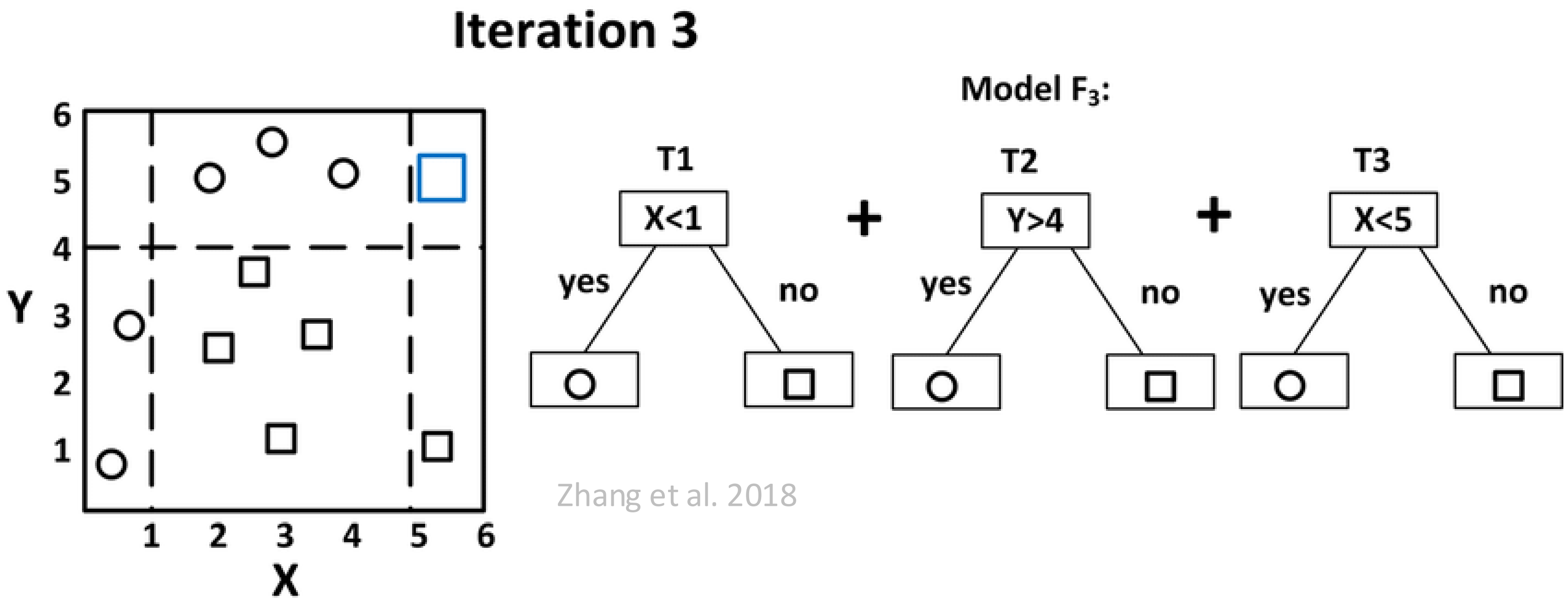
Zhang et al. 2018

Boosting



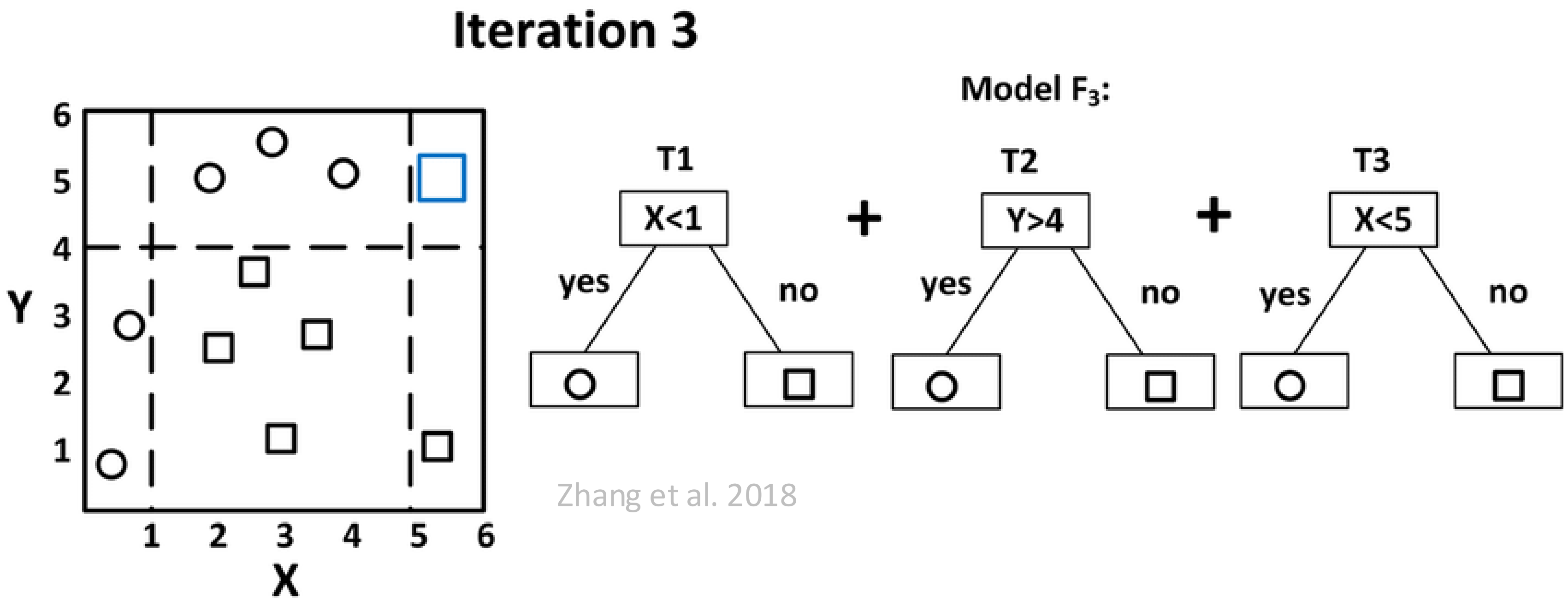
Zhang et al. 2018

Boosting



Zhang et al. 2018

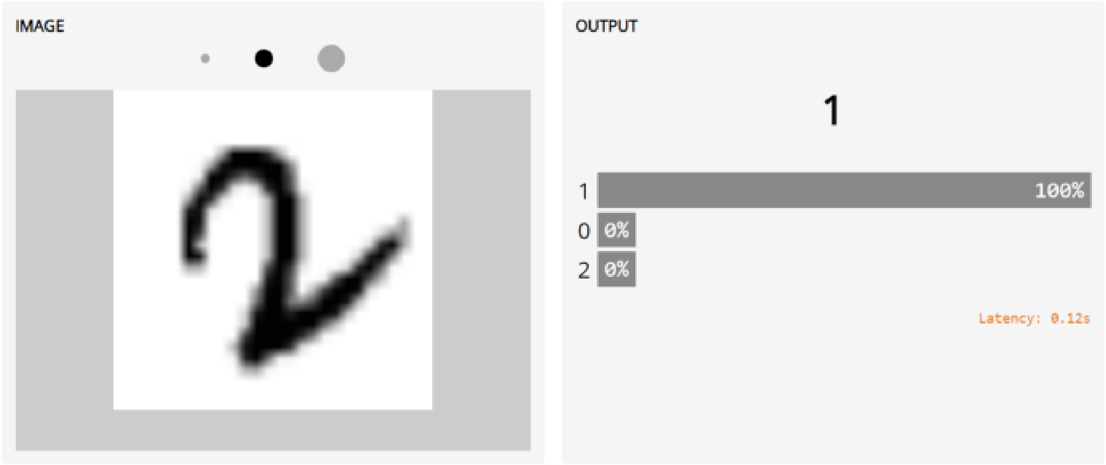
Boosting



Zhang et al. 2018

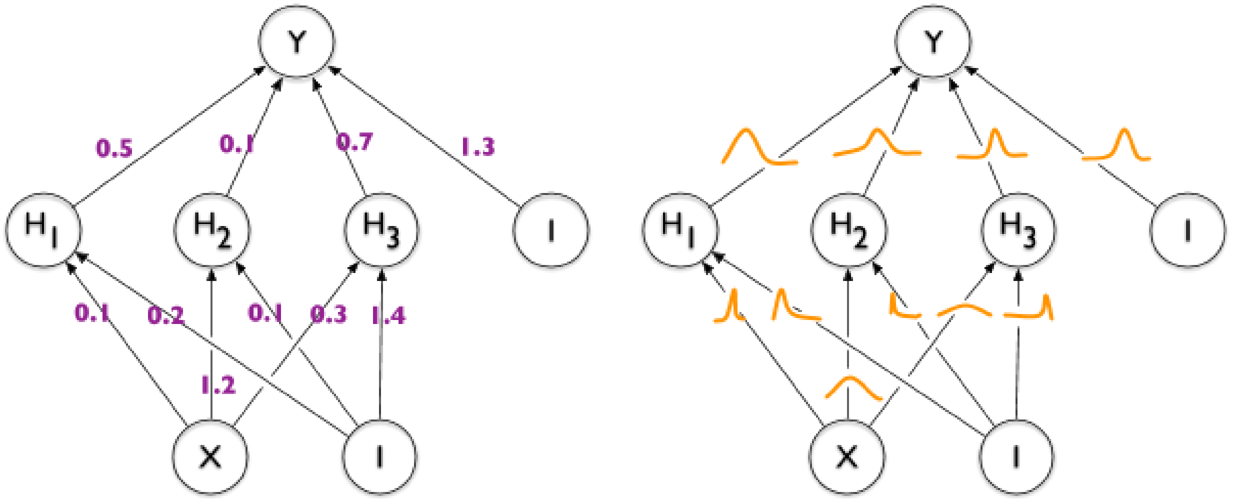
XGBoost (XAI)

Failures happen



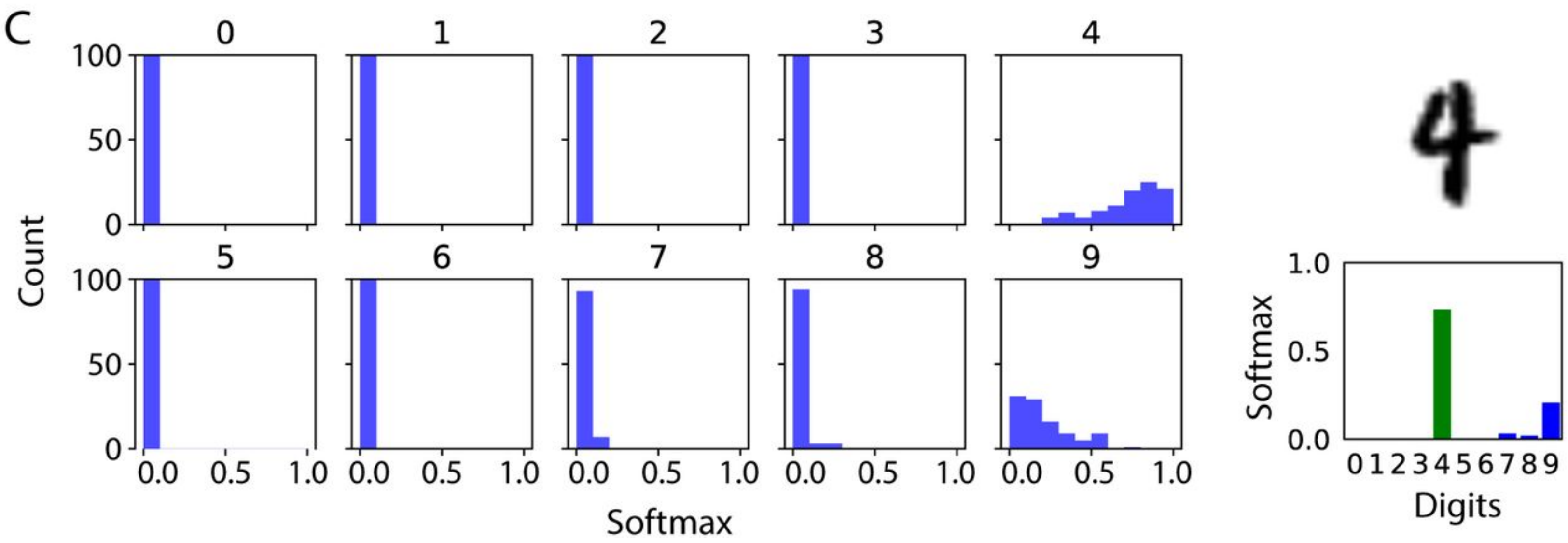
https://towardsdatascience.com/fixing-your-machine-learning-models-failure-points-e3ec0a047895)

Probabilistic ML



Concept of Bayesian neural networks (from https://sanjaykthakur.com/2018/12/05/the-very-basics-of-bayesian-neural-networks/)

Probabilistic ML



Bayesian neural networks classifying FashionMNIST and MNIST

SINTEF

SINTEF

Digital Akademi

UNSUPERVISED LEARNING 101

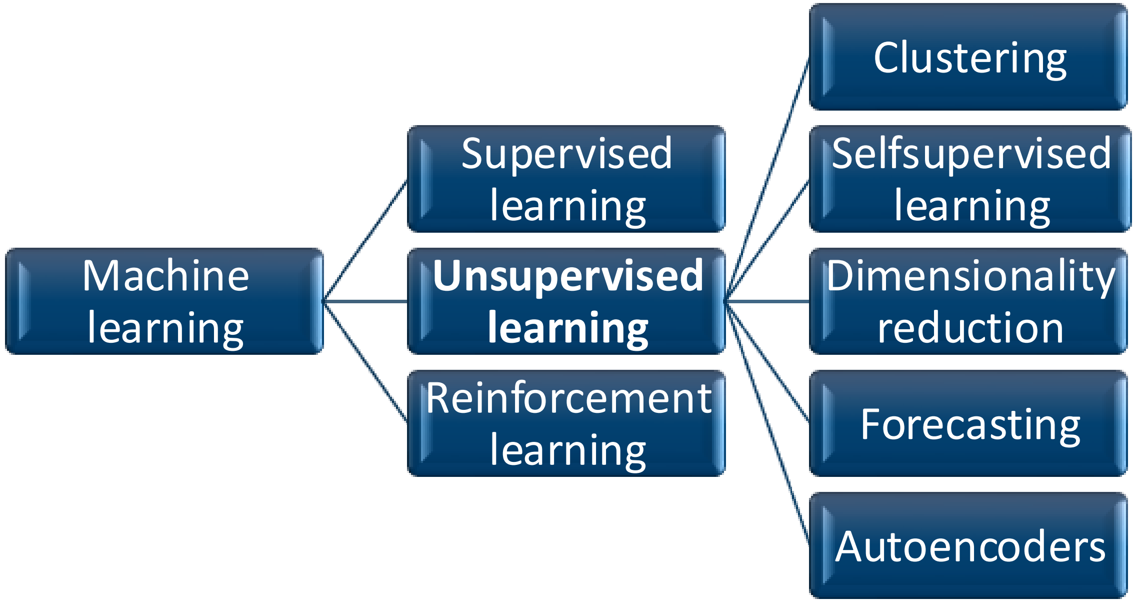
Bjørn Magnus Mathisen

Katarzyna Michałowska

Unsupervised learning

- Unlabelled data
- Finding patterns in the data
- Making the data more meaningful

	sepal_length	sepal_width	petal_length	petal_width	~~species~~
0	5.1	3.5	1.4	0.2	~~setosa~~
1	4.9	3.0	1.4	0.2	~~setosa~~
2	4.7	3.2	1.3	0.2	~~setosa~~
3	4.6	3.1	1.5	0.2	~~setosa~~
4	5.0	3.6	1.4	0.2	~~setosa~~



Supervised learning is the cherry!

Clustering

Grouping objects to simultaneously obtain:

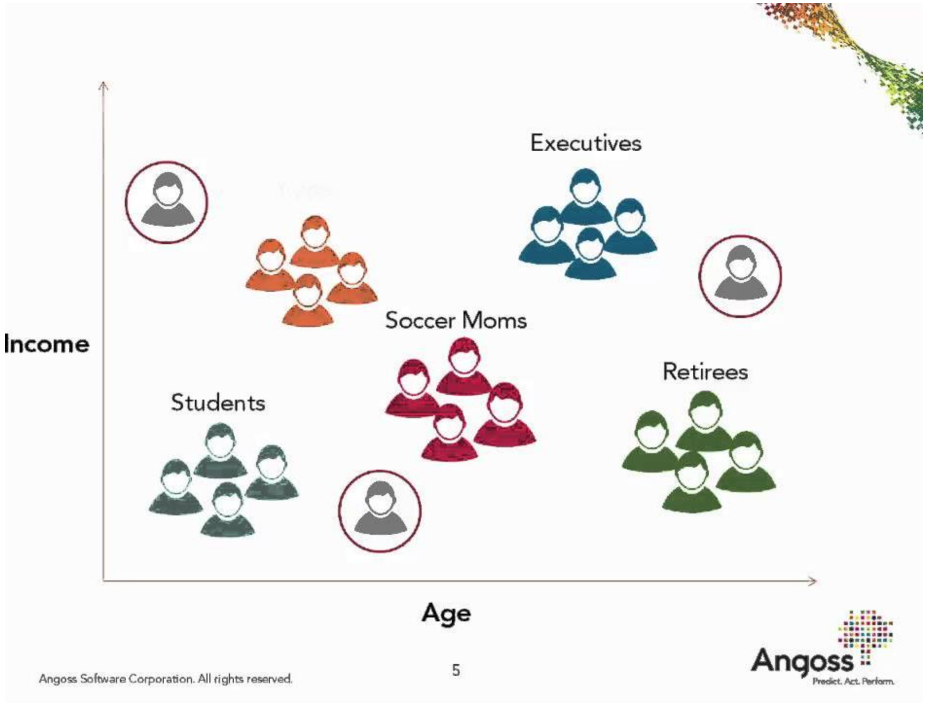
1. **Similar** objects in the same group
2. **Dissimilar** objects separated into different groups

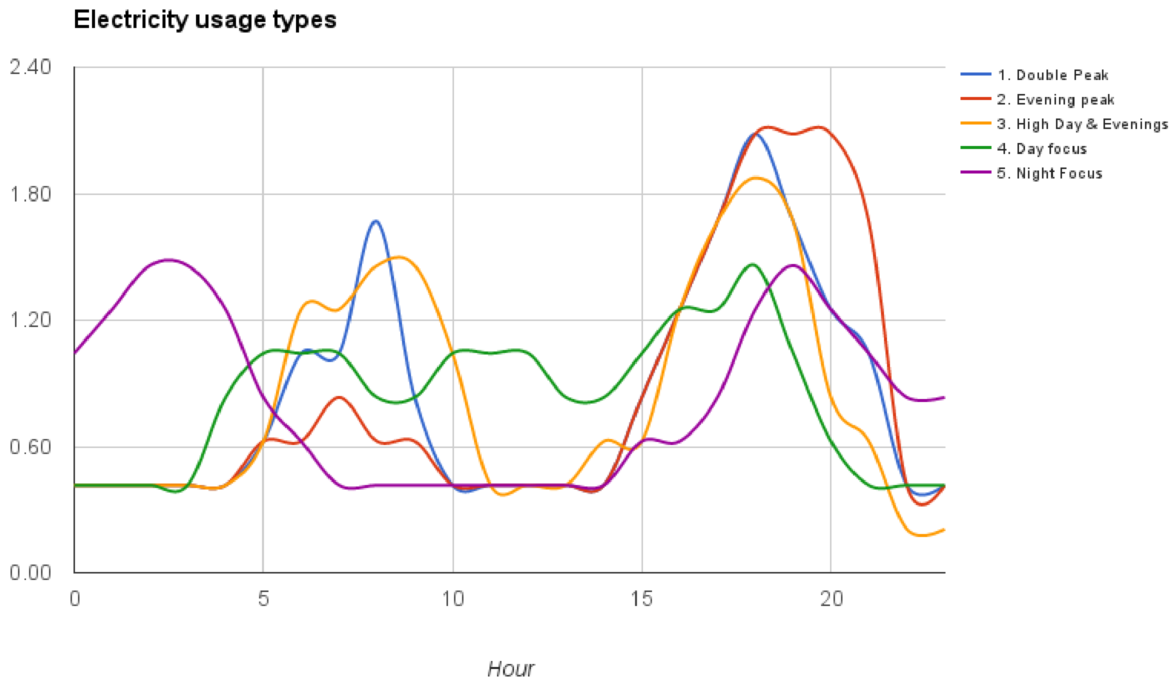
(a) (b) (c)

SINTEF

Source: J. Hu, J. Pei, Subspace multi-clustering: a review, 2017.

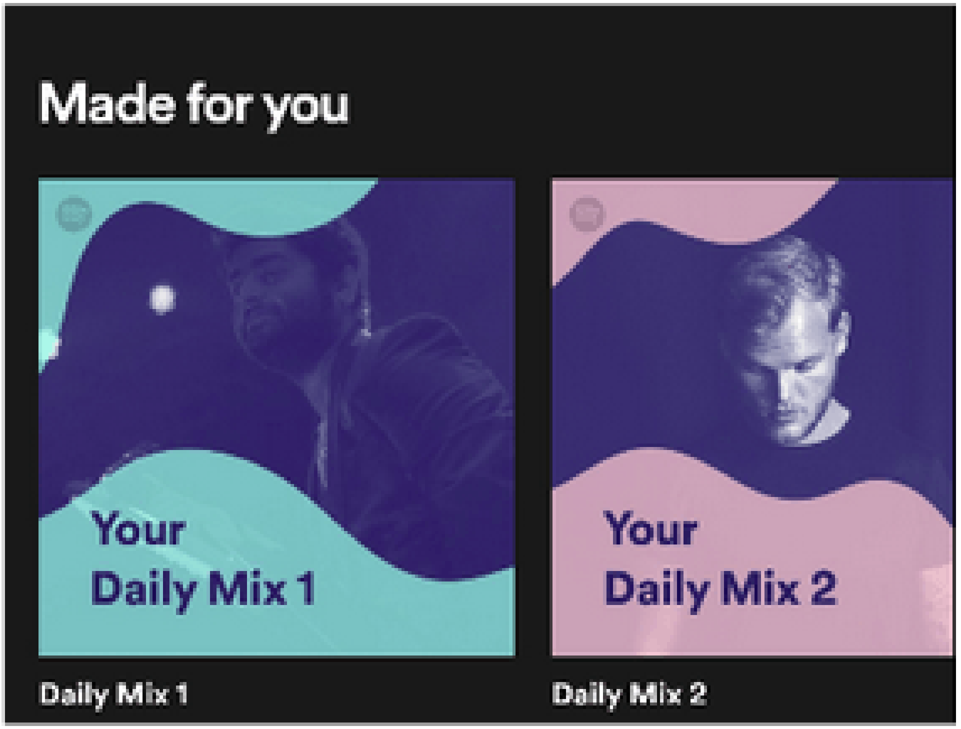
Clustering example: Consumer segmentation

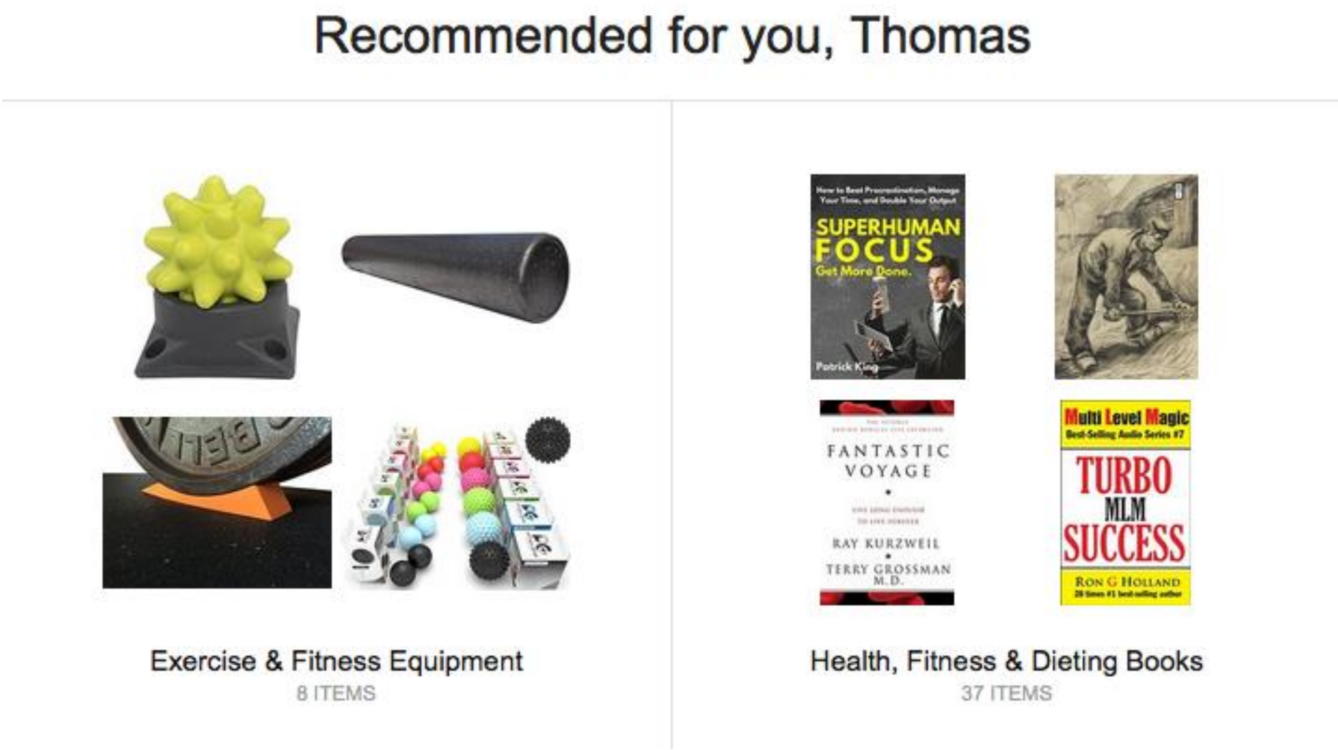




SINTEF

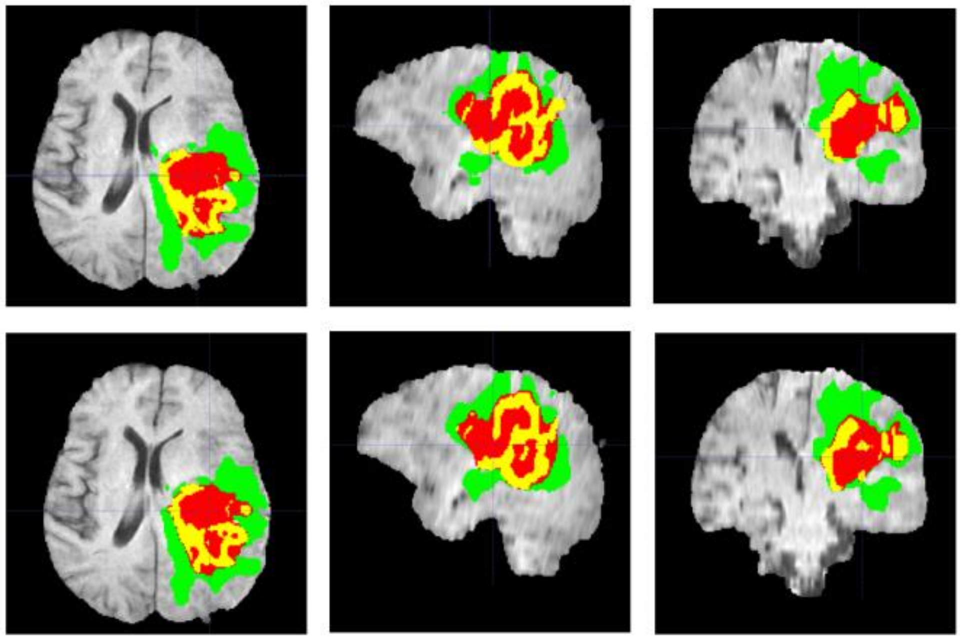
Clustering example: Recommender systems





SINTEF

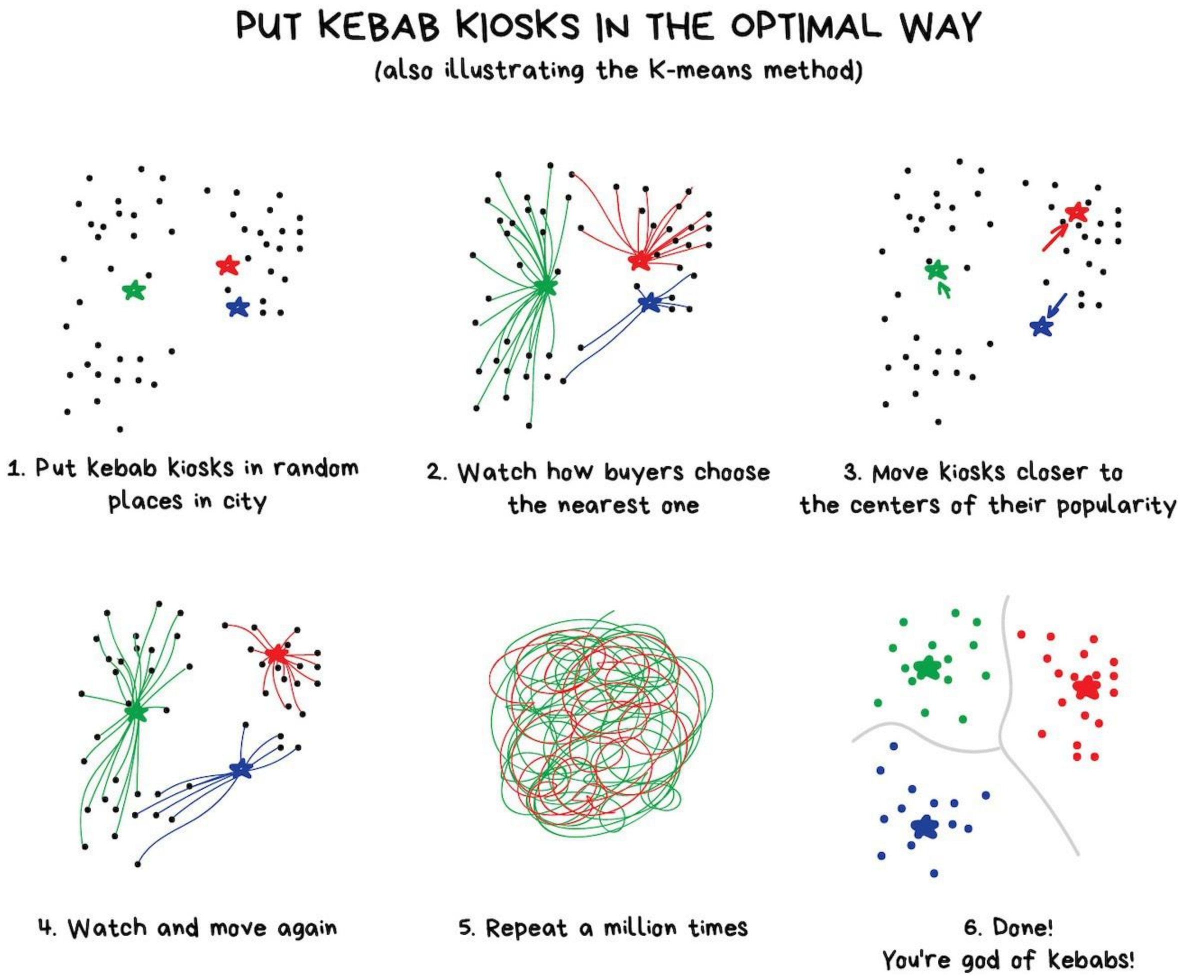
Clustering example: Image segmentation



Clustering: Other applications

- Document segmentation;
- Taxonomy;
- Gene expression clustering;
- Social network analysis;
- Denoising;
- Anomaly detection…

Clustering: Algorithm K-means



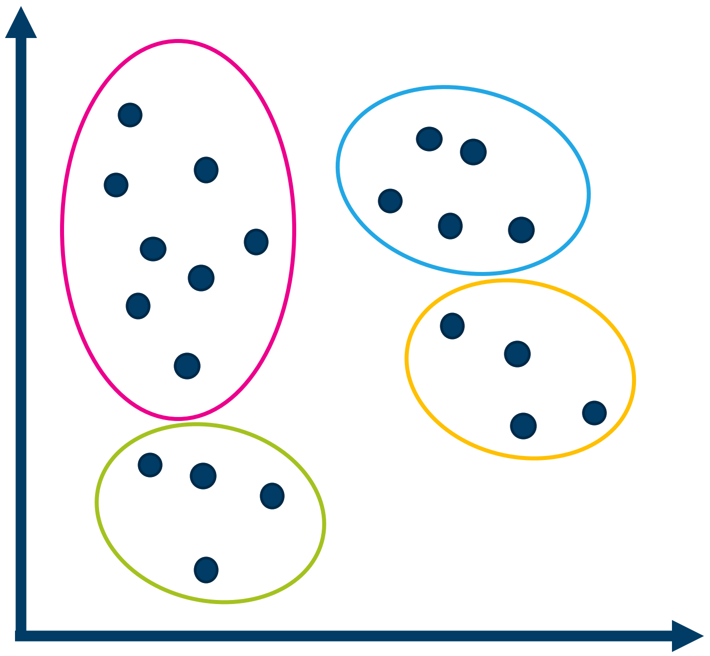
kiosk = cluster centroid
buyer = observation
(x,y) position of a buyer = features describing an observation

vas3k.com/blog/machine_learning/?ref=hn

Clustering: Optimal number of clusters

The 'best' number of clusters depends on the application

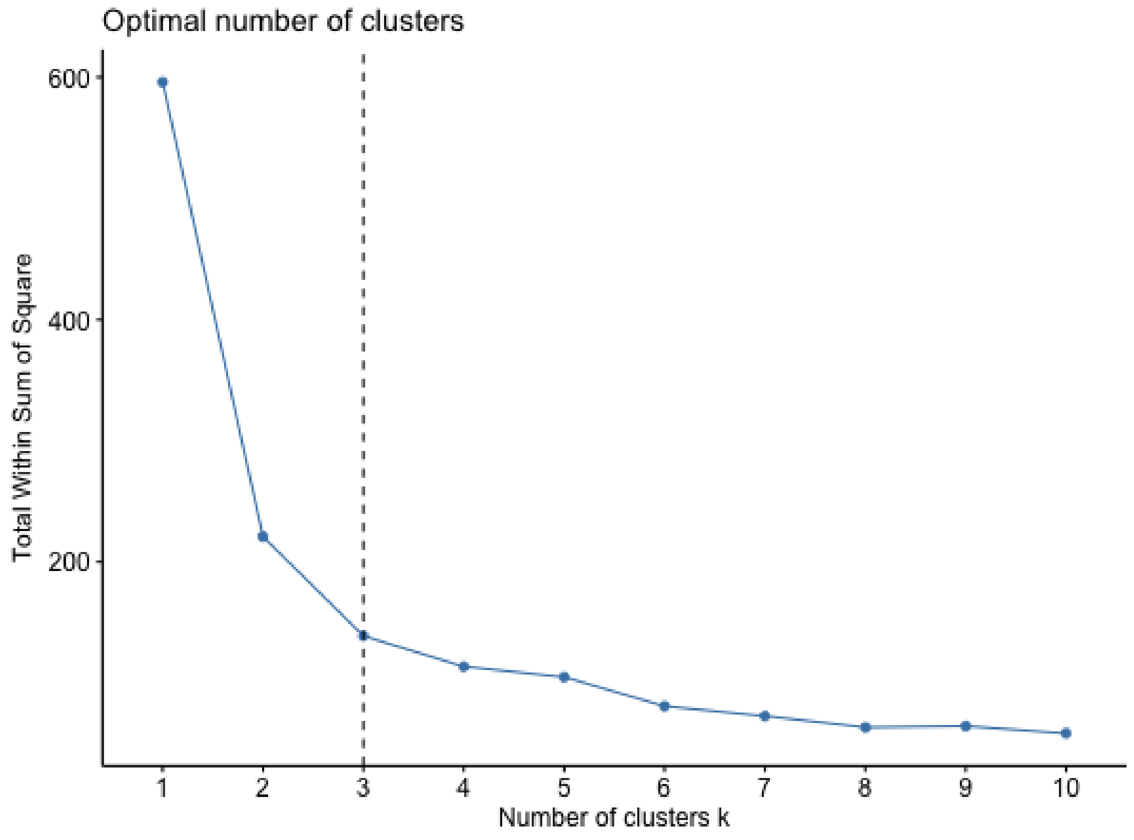
Think about:
- Geographical regions of different sizes
- Taxonomic families
- Etc.



SINTEF

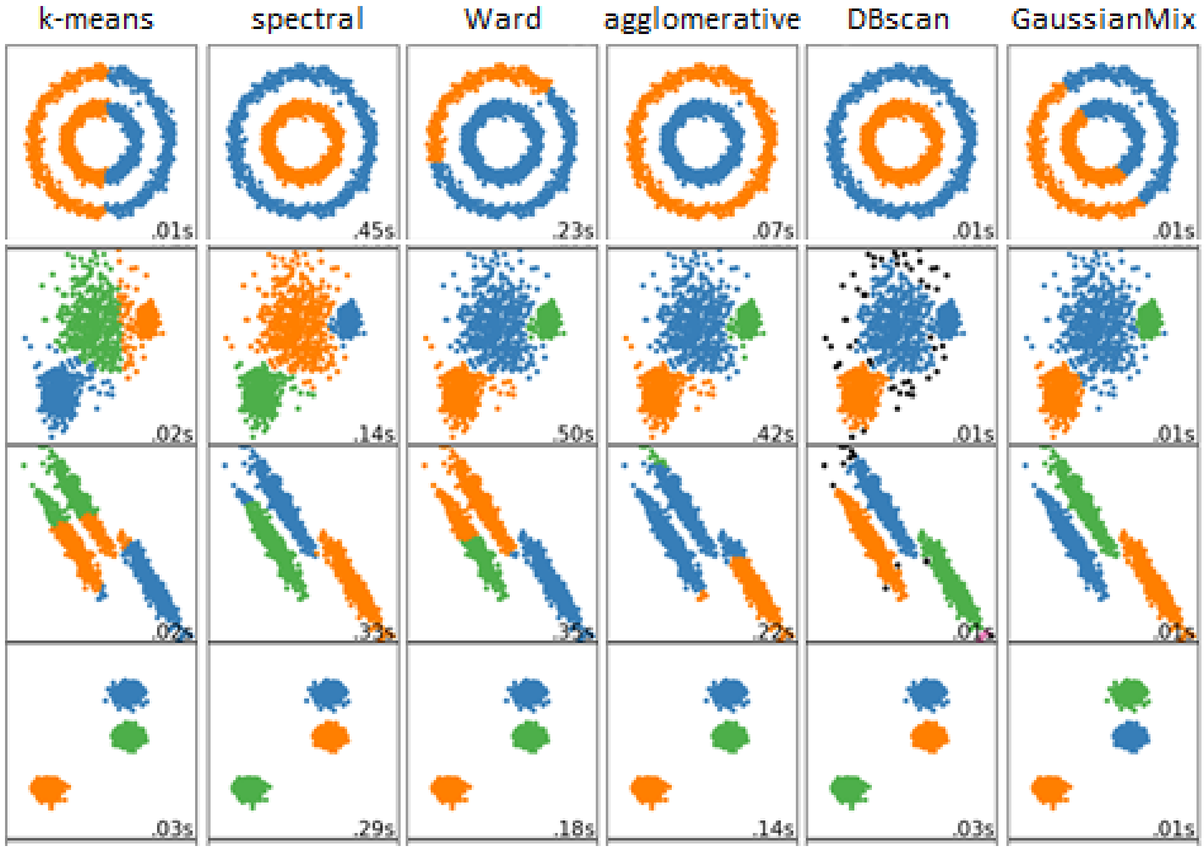
Clustering: Optimal number of clusters

- No ground truth = no ideal answer!
- Based on:
  - How well the clusters are separated
    -> Maximize the distances
  - How similar are the observations within clusters
    -> Minimize the distances
- Elbow method: How tight are the clusters?



SINTEF

Clustering: Selection of methods



scikit-learn.org/stable/modules/clustering.html#overview-of-clustering-methods

No Free Lunch Theorem!

- No algorithm can perform ideally on all data

Selfsupervised learning

Selfsupervised learning

Input

Output

Where is Paris ? In ____

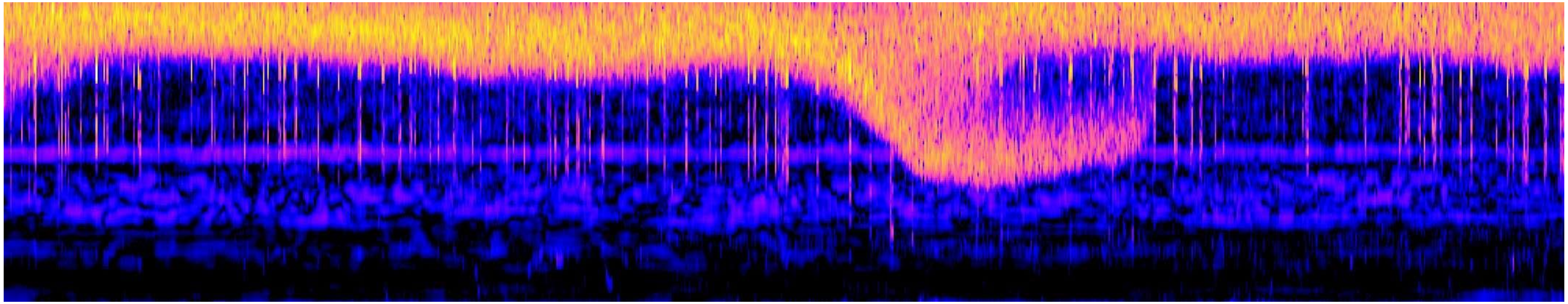
Original «prompt»

France
the
a
most

Selfsupervised learning

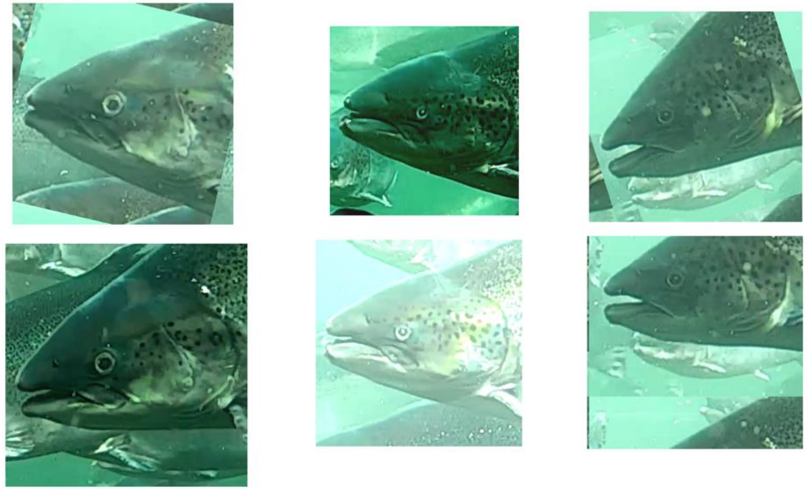
- Text
- Pictures
- Timeseries
- Video
- Even..

Selfsupervised learning



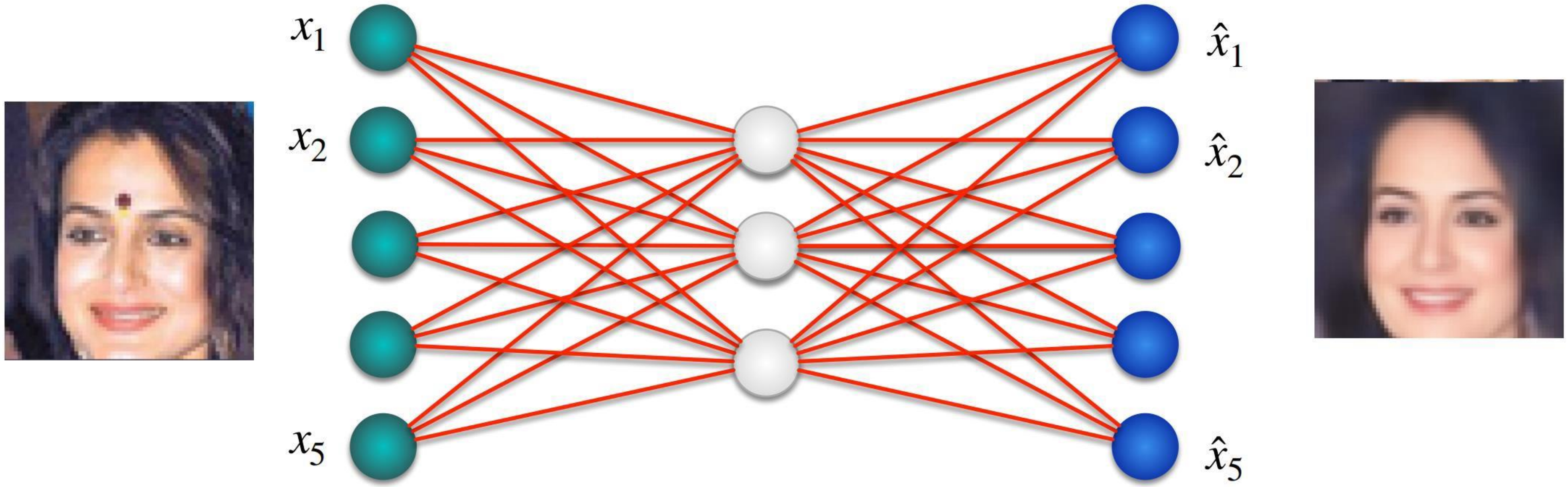
Echobert. Måløy et al.

Selfsupervised learning – cheat codes



Fishnet. Mathisen et al.

Honorable mentions - VAE

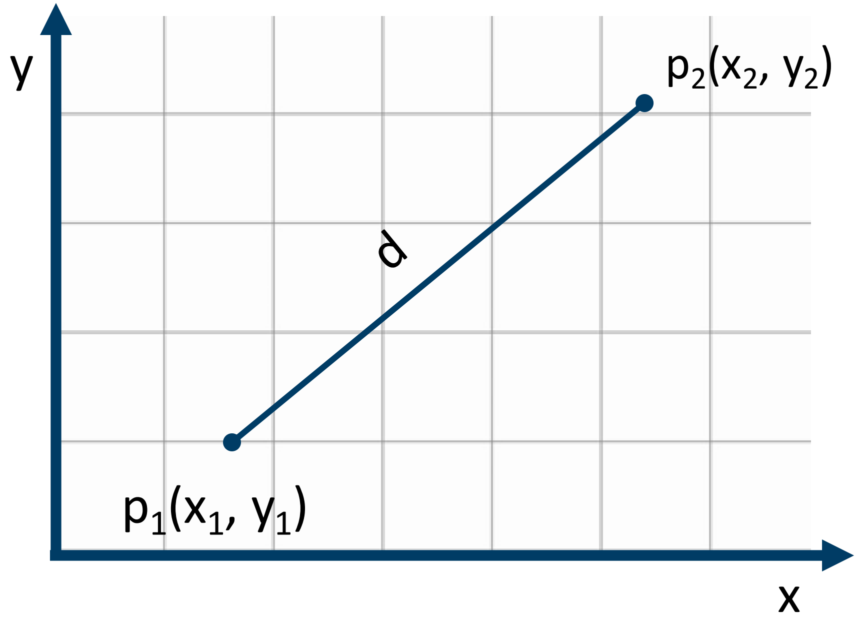


Face image source: Tolstikhin et al., ICLR 2018

How to measure similarity?

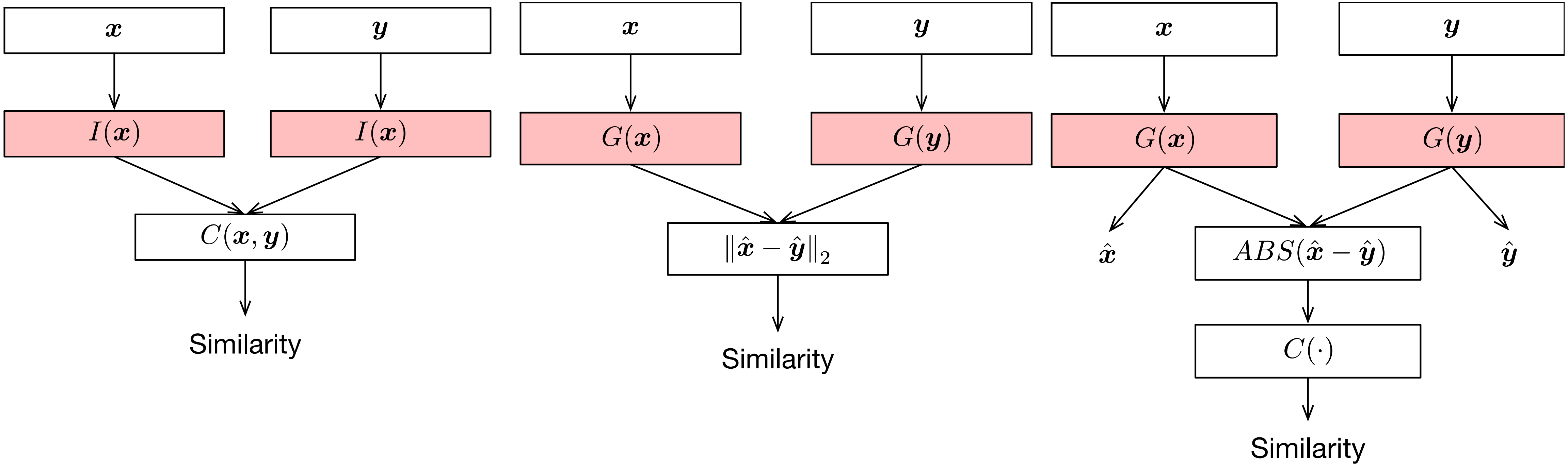
- Limited to *description* of observations
- Similarity between observations is defined using inter-observation distance measures or correlation-based distance measures

e.g.: Euclidean distance



$$d = \sqrt{(x_2 - x_1) + (y_2 - y_1) + \cdots}$$

How to learn the measure of similarity?



Mathisen, B.M., Aamodt, A., Bach, K. *et al.* Learning similarity measures from data.
Prog Artif Intell **9**, 129–143 (2020). https://doi.org/10.1007/s13748-019-00201-2

Other usages for SSL / similarity

- Embedding search (CLIP/RAG)
- ML assisted data-exploring
- Re-identification (faces, fishes, signatures)

Implementations

Keras

PYTORCH

Take home messages

- Each context is different
- The model is the data!
- Experimentation is key
- Lots of tricks lies in the preprocessing and data exploration
  - Not all features are important
  - Finding the correct method is an art

Gatys Ecker, Bethge 2015

Take home messages

- Each context is different
- The model is the data!
- Experimentation is key
- Lots of tricks lies in the preprocessing and data exploration
  - Not all features are important
  - Finding the correct method is an art

Gatys Ecker, Bethge 2015

SINTEF

Technology for a better society

Projects

- **SESAR EU Exploratory research PROJECT SynthAIr - Improved ATM automation and simulation through AI-based universal models for synthetic data generation**
- **Subzerospace**