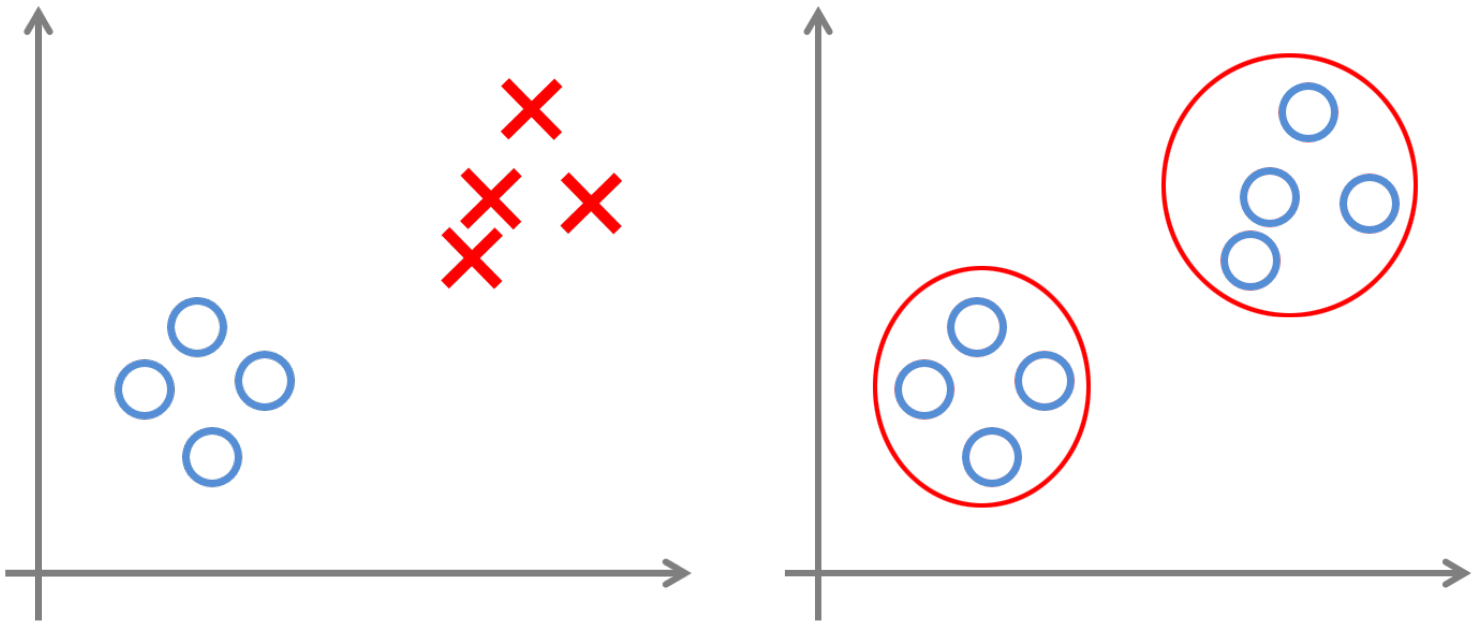
Neural Networks: Machine learning and Networks Characteristics

Andrzej Kordecki

Neural Networks (ML.ANK385 and ML.EM05): Lecture 03
Division of Theory of Machines and Robots
Institute of Aeronautics and Applied Mechanics
Faculty of Power and Aeronautical Engineering
Warsaw University of Technology

Table of Contents

Knowledge representation



Knowledge representation

The term "knowledge" in the definition of a neural network is present without an explicit description of what we mean by it:

> Knowledge refers to stored information or models used by a person or machine to interpret, predict, and appropriately respond to the outside world. (Fischler and Firschein, 1987)

Knowledge representation:

- How knowledge and facts about the world can be represented (especially in computer systems)?
- What kinds of reasoning can be done with that knowledge?

Knowledge representation

In real-world applications, it can be said that a good solution depends on a good representation of knowledge (Woods, 1986).

A major task for a neural network is to learn a model of the world (environment) in which it is embedded:

- The known world state, represented by facts about what is and what has been known (prior information).
- Observations of the world, obtained by means of sensors (measurements). Ordinarily, these observations are inherently noisy, being subject to errors due to sensor noise and system imperfections.

Knowledge representation

The sample (example, measurement) can be:

- labeled - example representing an input signal is paired with a corresponding desired response (i.e., target output).
- unlabeled - example representing a different realizations of the input signal all by itself.

A set of input–output pairs, with each pair consisting of an input signal and the corresponding desired response, is referred to as a set of training data, or simply training sample.

Knowledge representation

The design of a neural network may proceed as follows:

- An appropriate architecture is selected for the neural network and a subset of examples is then used to train the network by means of a suitable algorithm (learning phase),
- The recognition performance of the trained network is tested with data not seen before (testing phase).

The examples used to train a neural network may consist of both positive and negative examples.

Knowledge representation

Roles of Knowledge Representation (classification):

- Rule 1: Similar inputs from similar classes should usually produce similar representations inside the network, and should be classified as belonging to the same class,
- Rule 2: Items to be categorized as separate classes should be given widely different representations in the network.
- Rule 3: If a particular feature is important, then there should be a large number of neurons involved in the representation of that item in the network,
- Rule 4: Prior information and invariances should be built into the design of a neural network whenever they are available, so as to simplify the network design by its not having to learn them.

Knowledge representation

How to build prior information into Neural Network design?

Unfortunately, there are currently no well-defined rules for doing this.

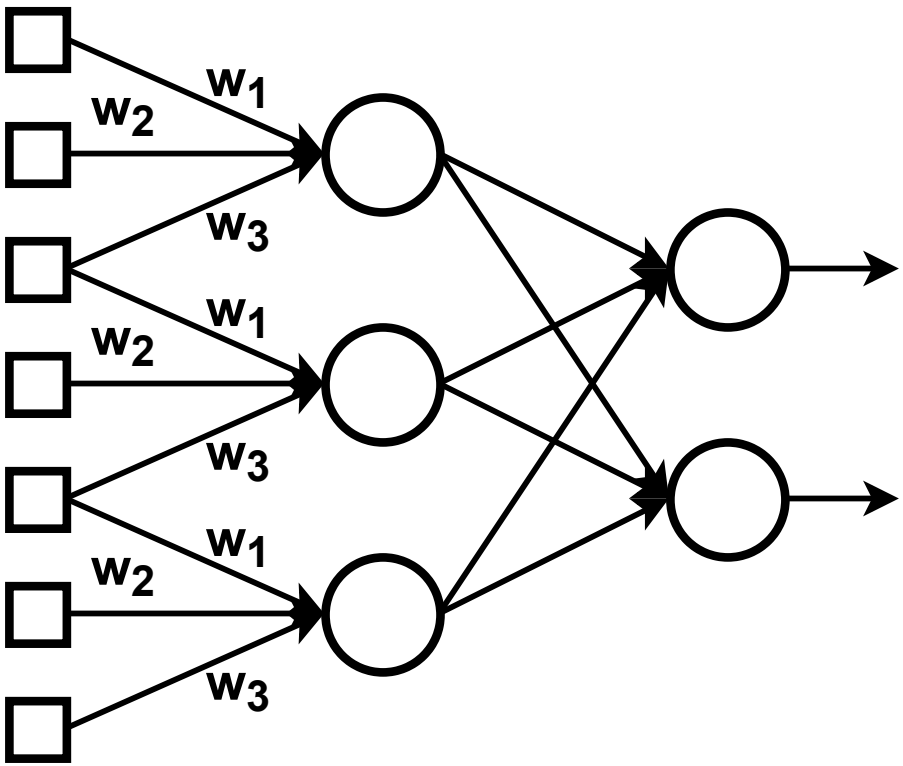
We have only some useful procedures e.g.:

- restricting the network architecture, which is achieved through the use of local connections known as receptive fields,
- constraining the choice of synaptic weights, which is implemented through the use of weight-sharing.

These two techniques, particularly the latter one, have a profitable side benefit: The number of free parameters in the network could be reduced significantly.

Knowledge representation

Example of combined use of a receptive field and weight sharing.



The convolution neural networks are designed on this idea.

Measurement experiment

Training data acquisition requires designing a measurement experiment

- Measurement experiment is the link between a theoretical argument, an neural network design and obtained results.
- The validity of inferences we draw from an experiment depend on the validity of the measures used.
- Measurement error reduces the power of your experiment: systematic and random measurement errors

Measurement procedure

Measurement procedure:

- Measurement procedures are developed as documents that allow operators to obtain results with uncertainty that does not exceed the target uncertainty.
- The detalization of measurement procedure is different, but it should be sufficiently detailed,

In recent years, there has been a tendency for reducing the share of measurement procedures-prescriptions and increasing the share of measurement procedures with a lower degree of detalization.

G. Nezhikhovskiy, Measurement procedure: from detailed description to technology, Journal of Physics: Conference Series, 1379, 2019

Sampling

Example of measurement errors:

- Sampling error occurs when the sample used is not representative of the whole population
- Quantization error is the difference between the analog signal and the closest available digital value at each sampling instant.

Sampling

Sampling is the reduction of a continuous-time signal to a discrete-time signal:

- Assignment of the discrete values,
- Loss of information,
- Sampling frequency,
- Reconstruction of a continuous function from samples is done by interpolation algorithms.

Aliasing is an effect that causes different signals to become indistinguishable after sampling process.

Nyquist - Shannon theorem

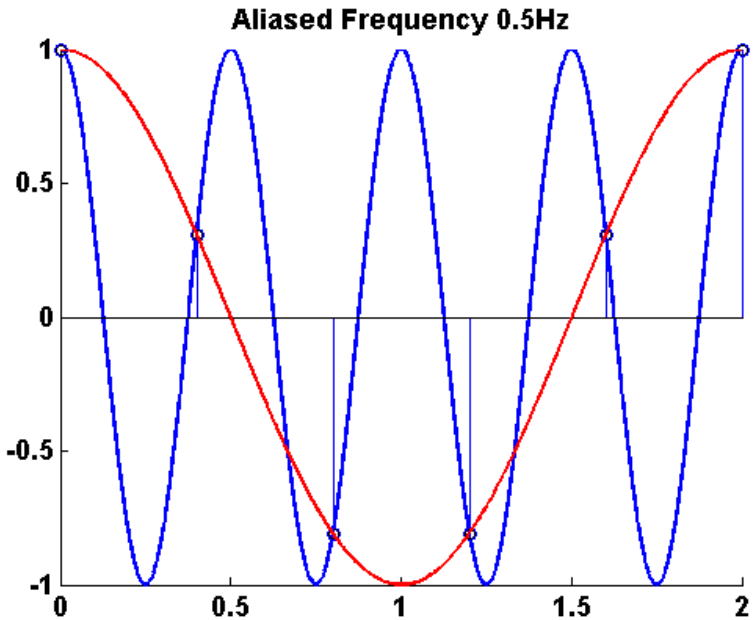
The Nyquist Theorem sampling theorem is a principle in the digitization of analog signals:

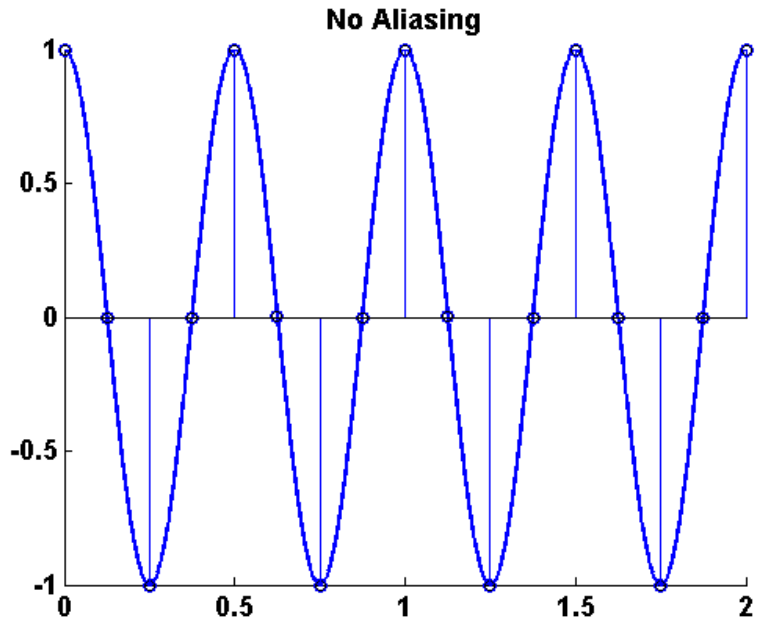
$$\text{Sampling frequency} > 2\times \text{ of maximum system } f_{max},$$

- Any analog signal consists of components at various frequencies. The highest frequency component in an analog signal determines the bandwidth of that signal.

Sampling

Example: System frequency $f = 2Hz$, sampling frequency $f_1 = 2.5Hz$ and $f_2 = 8Hz$





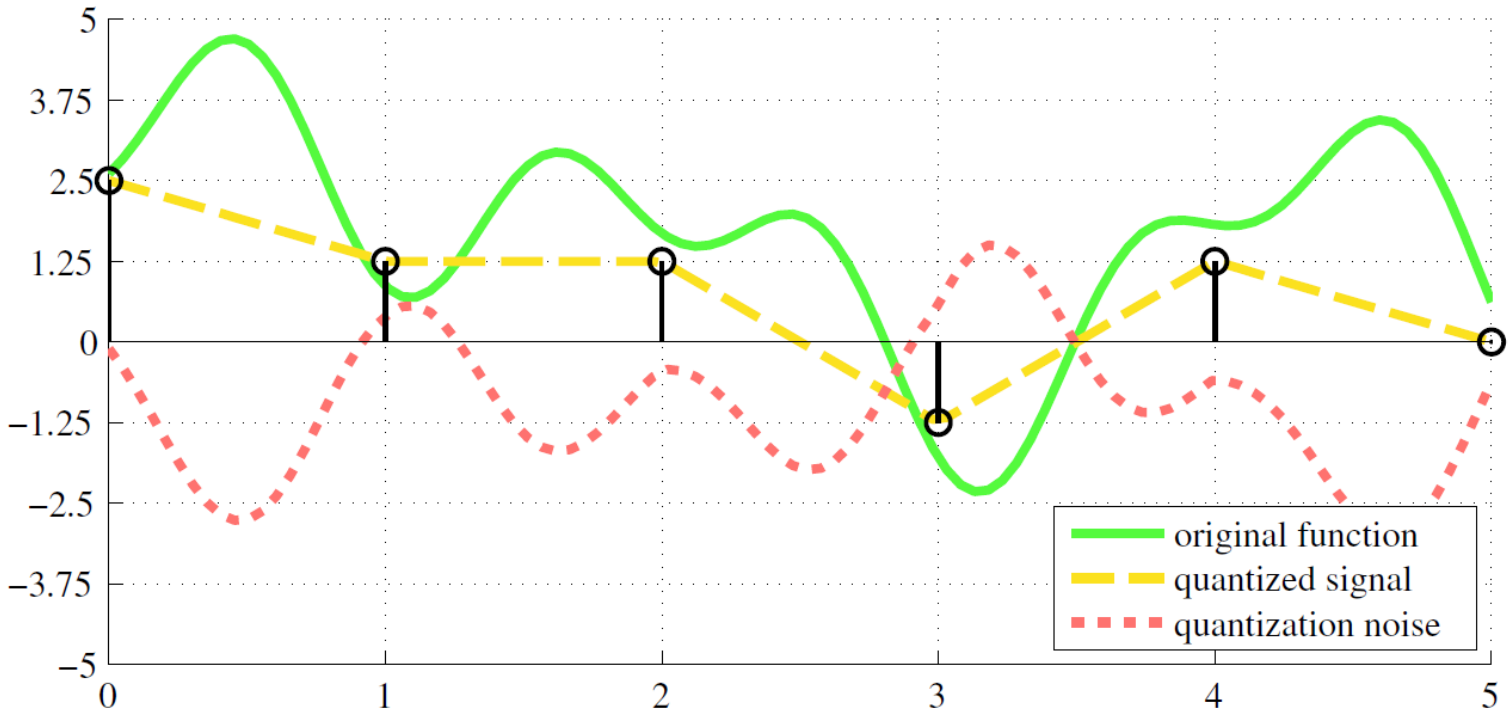
Quantization

Quantization is the process of mapping from a large set to smaller set:

- Assignment of the discrete values,
- Obtaining needed amount of data to represent the output,
- Loss of information,
- more levels - lower is its quantization noise power (quantization error).

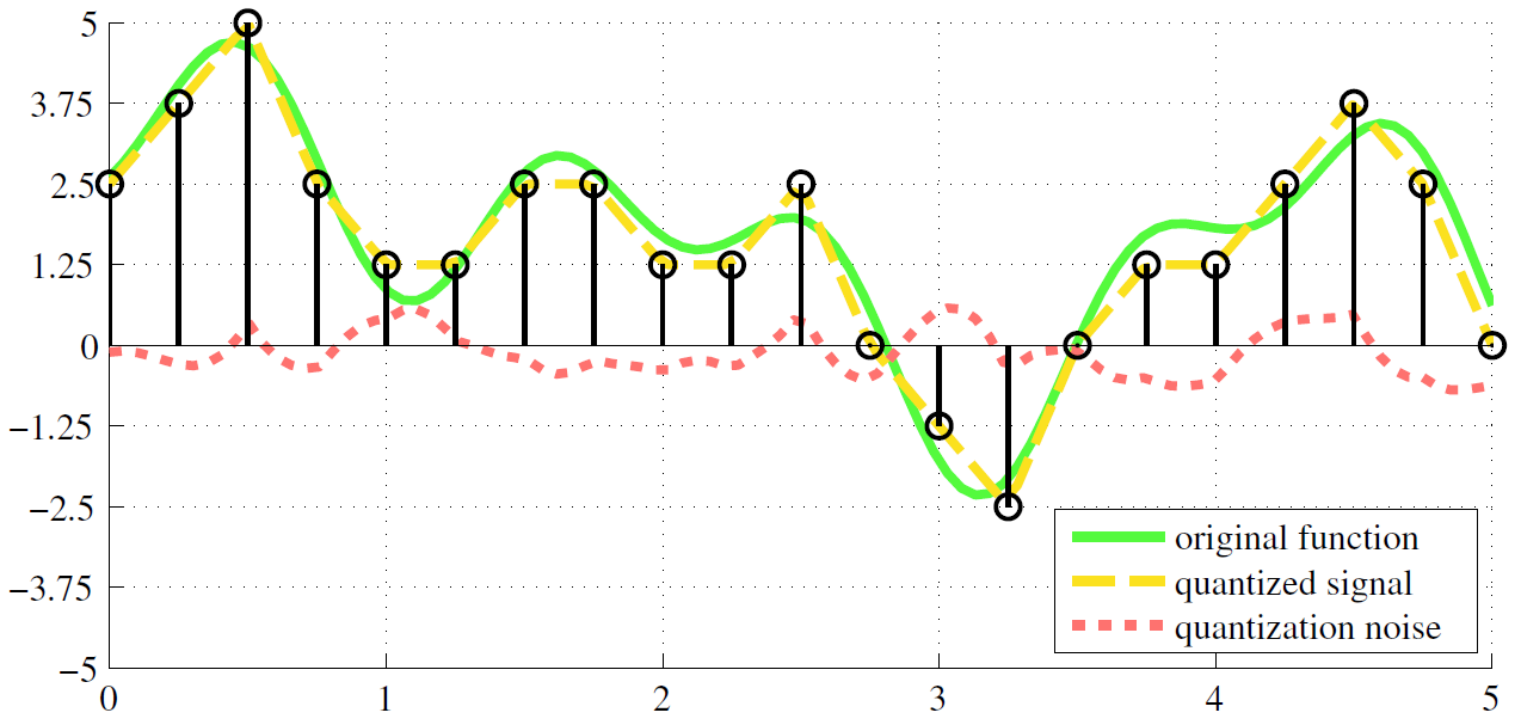
Sampling and Quantization

Example: Quantization for $T = 1s$, 3-bit converter



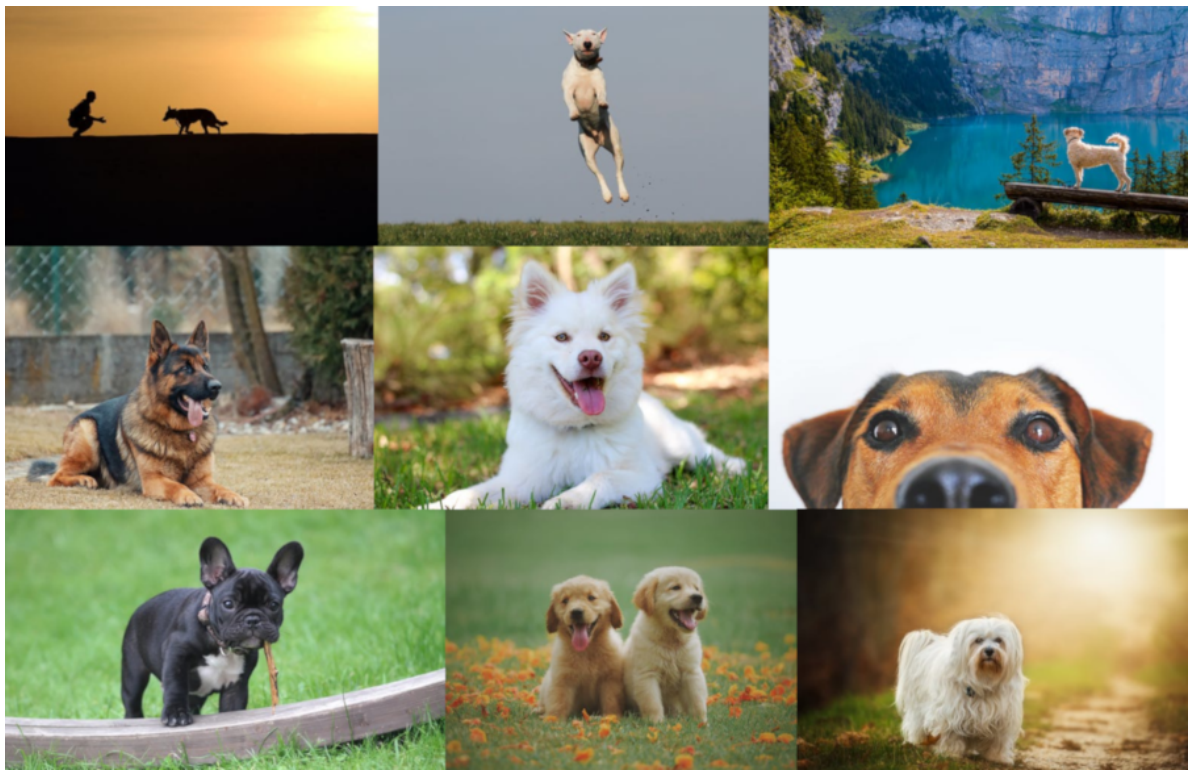
Sampling and Quantization

Example: Quantization for $T = 0.25s$, 3-bit converter



Images

Training data (input data) can have many different distortion.



Dogs in images: location, background, size, movement, occlusion, but also: age, breed, color and many others

Data augmentation

The goal of applying data augmentation is to increase the generalizability of the neural network.

Data augmentation is a method by which you can virtually increase the number of samples in your dataset using data you already have:

- data augmentation should not change the meaning of the samples,
- data augmentation applies also to labels,
- data augmentation cannot fully replace real data (specially in case of a small amount of data),
- the use of generated data can cause a drop of accuracy, but is it really a drop of accuracy?

Data augmentation

Data augmentation in neural networks training:

- Dataset generation by data expansion with use of data augmentation,
- In-place/on-the-fly data augmentation,
- Combining dataset generation and in-place augmentation.

Data augmentation methods for images:

- Image operation like: rotations, shifts, flips, size, cropping, whitening, contrast enchantment, adding noise, affine transformation and more,
- Image rendering by joining real objects and real backgrounds,
- Image rendering with use of 3D models.

Data augmentation

Example of data augmentation.

Conclusion

Training data in the neural networks:

- Knowledge of that kind of date NN need to perform the task and variety of its values,
- In case of deep learning, we need **a lot** of input data, e.g. CNN VGG19 have 144 millions parameters,
- Measurement is not enough. Data need to labeled by expert,
- Data normalization is important in NN generalization.

System Modeling

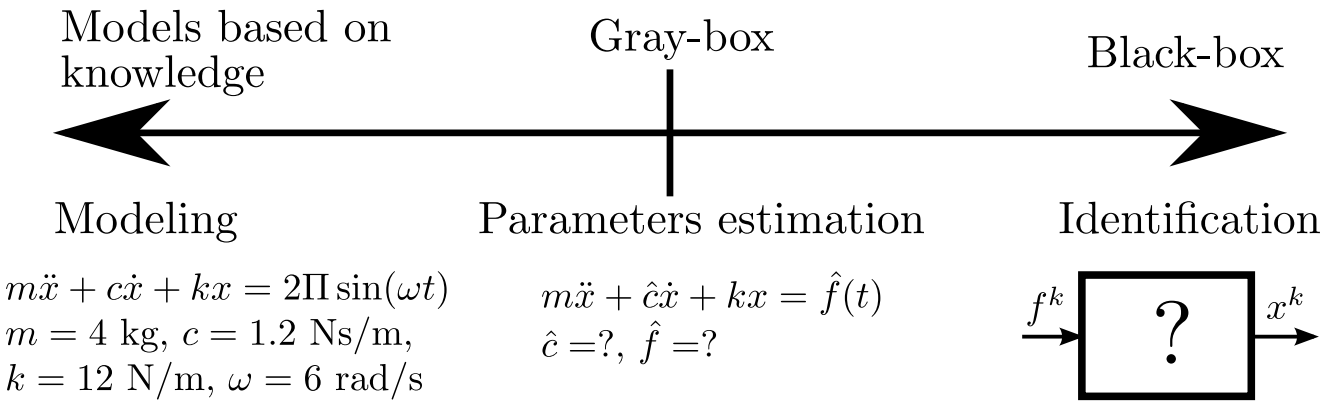
Static and Dynamic Modeling

A mathematical model is an abstract model that uses mathematical equations to describe the behavior of a system. The process of developing a mathematical model is termed mathematical modeling. We can divide mathematical models according to their structure:

- Dynamic model for time-dependent changes in the state of the system,
- Static model calculates the system in equilibrium (steady-state).

Depending on the available data, we can distinguish different types of model parameters identification.

Model Types



- Black-box
  - Lack of information about the model.
  - Long iterative procedure.
- White-box (knowledge-based model)
  - Complete information about the model.
  - Desired but often difficult to obtain.
- Gray-box
  - Partial information about the model.
  - The more we know about system, the better!

Function Approximation

The neural network can approximate an unknown input–output mapping in two ways:

- System identification. Example: The input–output relation of an unknown memoryless (time invariant) multiple input–multiple output (MIMO) system. We use the set of labeled examples to train a neural network as a model of the system. The goal is to adjust the free parameters of the network to minimize the difference between the outputs of the unknown system and the neural network in a statistical sense.

Function Approximation

The neural network can approximate an unknown input–output mapping in two ways:

- Inverse modeling. Example: The input–output relation of an unknown memoryless MIMO system. The requirement in this case is to construct an inverse model that produces the vector u in response to the vector y. The inverse system may thus be described by

$$u = f^{-1}(y)$$

where the $f^{-1}(.)$ is the inverse of $f(.)$.

State equations

In control engineering, a state-space representation is a mathematical model of a physical dynamic system as a set of input, output and state variables related by first-order differential equations or difference equations. In the case of a linear, continuous and stationary model:

$$\dot{x}(t) = Ax(t) + Bu(t)$$
$$y(t) = Cx(t) + Du(t)$$

where:

- x - state vector,
- y - output vector,
- u - input (control).

State equations

State equations for a discrete model:

$$\begin{cases} x(k+1) = Ax(k) + Bu(k) \\ y(k) = Cx(k) + Du(k) \end{cases}$$

- x(k) – state vector for time kT,
- u – input vector,
- y – output vector,
- A, B, C, D – relevant matrices.

State variables x unambiguously describe the current state of the system.

State equations

- Linear and deterministic system:

$$\begin{cases} x(k) = Ax(k-1) + Bu(k-1) \\ y(k) = Xc(k-1) + Du(k-1) \end{cases}$$

- Nonlinear and deterministic system:

$$\begin{cases} x(k) = \Phi(x(k-1), u(k-1)) \\ y(k) = \Psi(x(k-1), u(k-1)) \end{cases}$$

Φ, Ψ – nonlinear vector functions.

Any recursive network can be described using the last state equations of the above form, regardless of its complexity.

Equations Discretization

Model discretization:

- Input analog signal operation:
  - Sampling time T (most often fixed)
  - Signal quantization.
- Model equations discretization:
  - Approximation is used for all differentials.
  - We obtain algebraic equations.
  - There are called difference or recurrence equations.
  - There are many other schemes for solving Ordinary Differential Equations (ODE).
- Preparation for numerical calculations (code implementation), e.g. solve recursively by a computer.

Equations Discretization

Function differential $x(t)$:

$$\dot{x} = \lim_{\Delta t \to 0} \frac{\Delta x}{\Delta t} \quad \Delta x \text{ change of } x \text{ in time } \Delta t$$

Assuming small T we will introduce the Euler method:

- Forward rectangular rule:

$$\dot{x}(k) \approx \frac{x(k+1) - x(k)}{T}$$

- Backward rectangular rule:

$$\dot{x}(k) \approx \frac{x(k) - x(k-1)}{T}$$

Analytic solution

Example: Solve the initial value problem

$$\begin{aligned} 2\ddot{x}(t) + 3\dot{x}(t) + x &= 4 \\ \dot{x}(0) &= 1 \\ x(0) &= 1 \end{aligned}$$

Analytic solution of second-order linear ODE:

$$x(t) = e^{-t} - 4e^{-t/2} + 4$$

Tips: www.wolframalpha.com for problem:
2x'' + 3x' + x = 4, x(0) = 1, x'(0) = 1

Discrete solution

1. Problem

$$\begin{cases} 2\ddot{x}(t) + 3\dot{x}(t) + x(t) = 4 \\ \dot{x}(0) = 1 \\ x(0) = 1 \end{cases}$$

2. Order reduction

$$\begin{cases} y(t) = \dot{x}(t) \\ 2\dot{y}(t) + 3\dot{x}(t) + x(t) = 4 \\ \dot{x}(0) = 1 \\ x(0) = 1 \end{cases}$$

3. Approximation

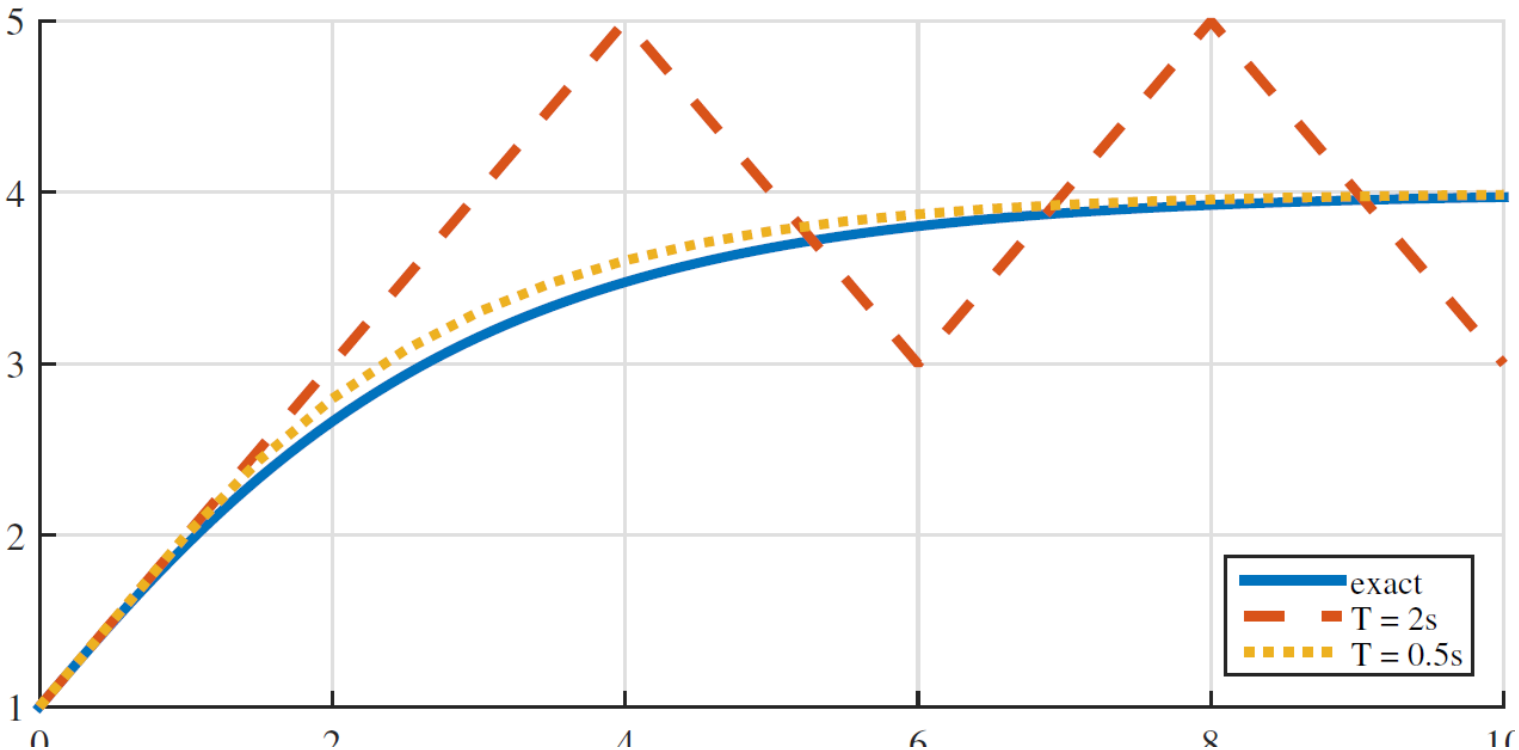
$$\begin{cases} y(k) = \frac{x(k+1)-x(k)}{T} \\ 2\frac{y(k+1)-y(k)}{T} + 3\frac{x(k+1)-x(k)}{T} \\ \quad +x(k) = 4 \\ \dot{x}(0) = 1 \\ x(0) = 1 \end{cases}$$

4. Finally

$$\begin{cases} x(k+1) = x(k) + Ty(k) \\ y(k+1) = y(k) + 2T \\ \quad -\frac{3}{2}x(k+1) \\ \quad +\frac{1}{2}(3-T)x(k) \\ \dot{x}(0) = 1 \\ x(0) = 1 \end{cases}$$

Discrete solution

Solution of the initial value problem using recurrence equations and Euler method.



Recursive Networks

This simple recursive networks network can be described as a dynamical system:

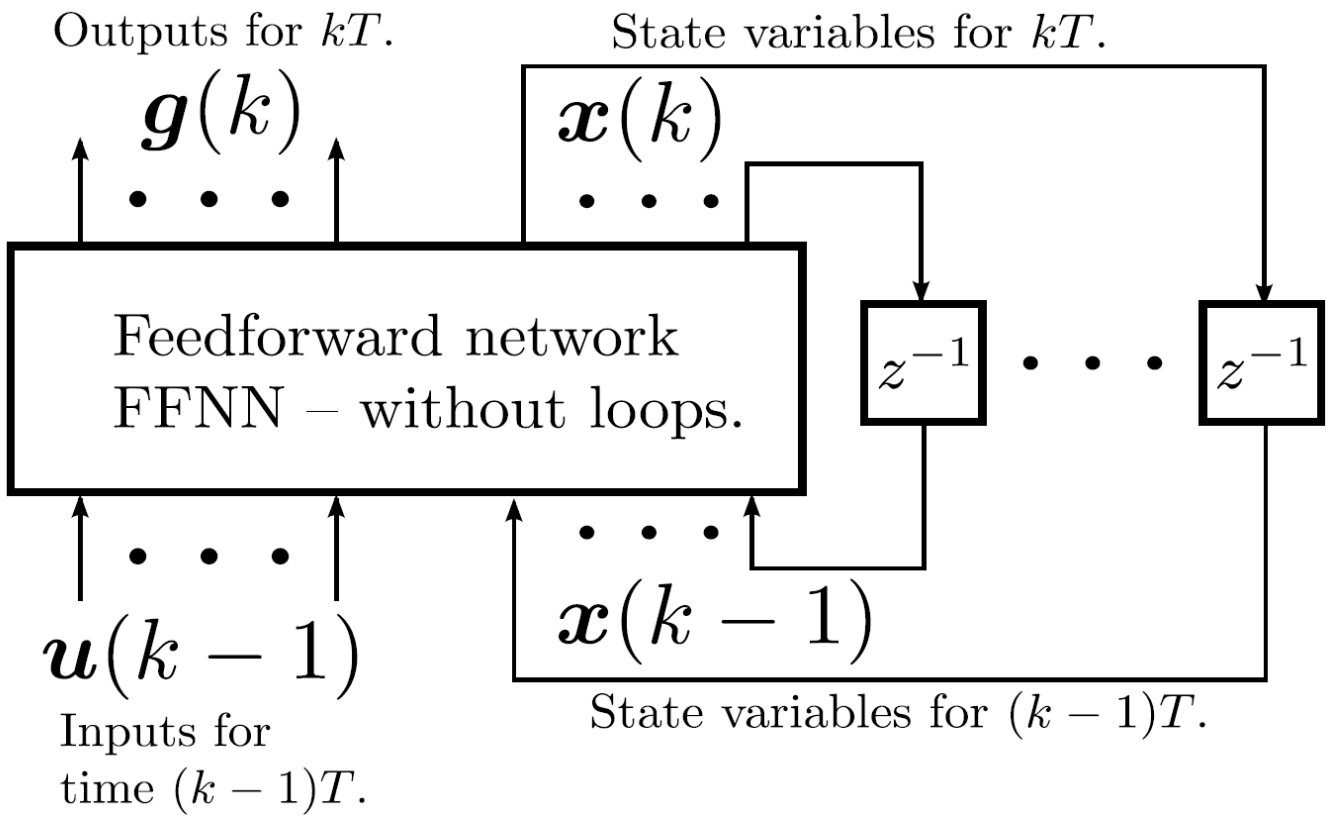
$$\begin{cases} x(k) = \sigma_x(W_{ih}u(k) + W_{uh}x(k-1)) \\ y(k) = \sigma_y(W_y x(k)) \end{cases}$$

- W_{uh}, W_{uh} and W_y - weight matrices,
- σ_x and σ_y - the hidden and output unit activation functions.

The state is defined by the set of hidden unit activations $h(t)$.

Recursive Networks

$$\begin{cases} x(k) = \Phi(x(k-1), u(k-1)) \\ y(k) = \Psi(x(k-1), u(k-1)) \end{cases}$$



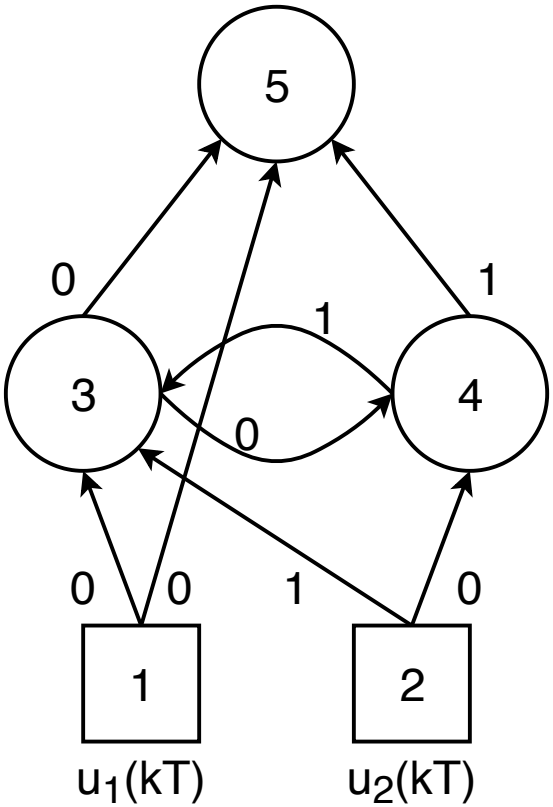
z^{-1} – unit delay

Recursive Networks

Recursive networks in terms of their properties as dynamical systems:

- Stability - concerns the boundedness of the network outputs and the response of the network outputs to small changes,
- Controllability - concerns whether it is possible to control system dynamic (network can be steerable to any desired state within a finite number of time steps),
- Observability - concerns whether it is possible to observe the results of the control applied (network state can be determined from a finite set of inputs/outputs).

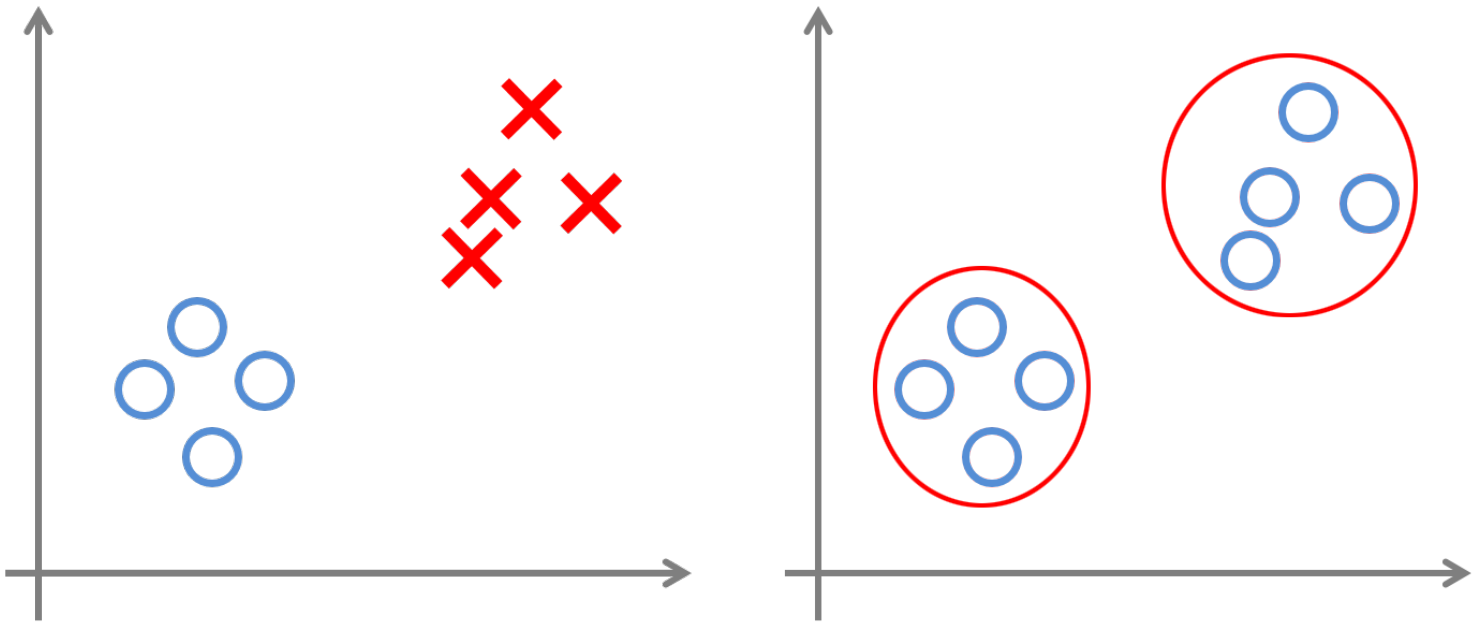
Recursive Network Example



- Sampling time T,
- Network operation for time kT,
- Often T is omitted.

Neuron	Outputs	Inputs
3	$y_3(kT)$	$u_1(kT),\ u_2[(k-1)T],\ y_4[(k-1)T]$
4	$y_4(kT)$	$u_2(kT),\ y_3(kT)$
5	$g(kT) = y_5(kT)$	$u_1(kT),\ y_3(kT),\ y_4[(k-1)T]$

Machine Learning



Introduction NN learning

One of the most powerful features of neural networks is their ability to learn and generalize from a set of training data. For this purpose, neural networks adapt the weights of the connections between neurons.

A neural network learns about its environment through an iterative process. There are three broad types of learning:

- Supervised learning (learning with teacher)
- Unsupervised learning (Learning without a teacher - learning with no help)
- Reinforcement learning (Learning without a teacher - learning with feedback)

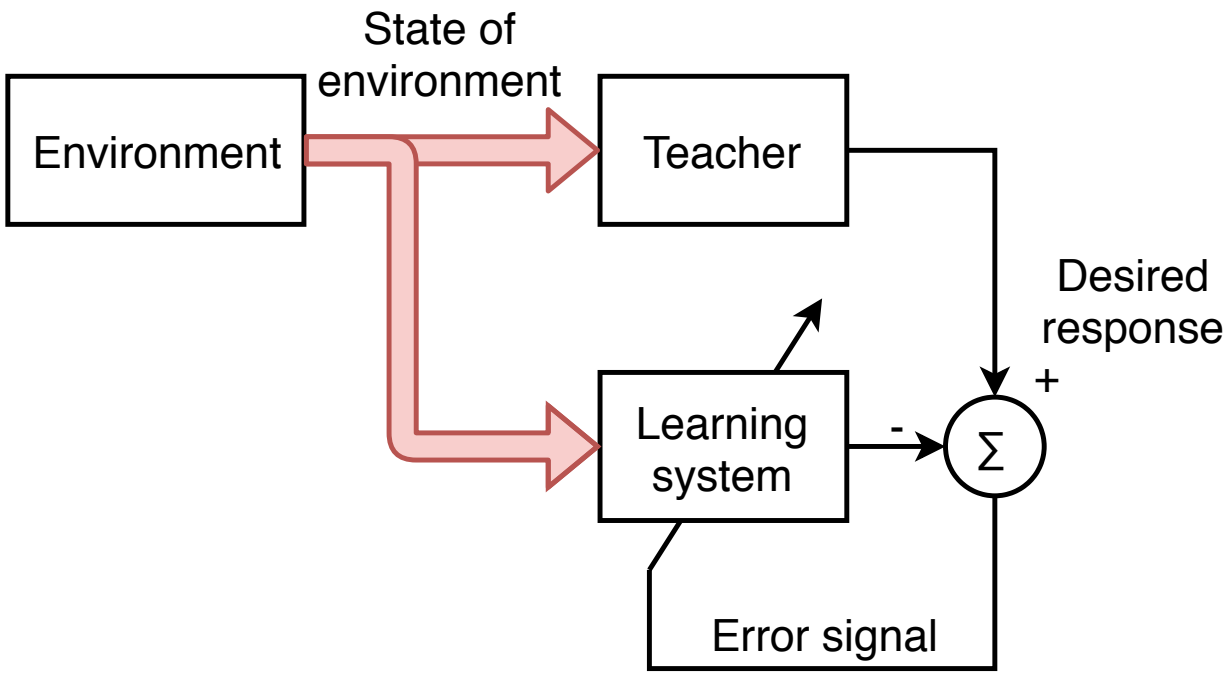
Supervised Learning

Characteristic of supervised Learning:

- the "correct answers" are known, each example consist of an input data and a desired output value (labeled data),
- analyzes the training data and produces an mapping function,
- optimal algorithm allows to determine the output values for unseen data,
- majority of practical machine learning problems use supervised learning.

Supervised Learning

The knowledge of environment is being represented by a set of input–output examples, but the environment is unknown to the neural network.



The aim of the neural network training is to emulate teacher.

Supervised Learning

Issues of supervised learning:

- trade-off between 'model generalization' and accuracy of fitting model to data,
- depends on the amount of training data and NN architecture (model complexity),
- noise and especially data outliers have a large impact of on the model parameters,
- trouble dealing with new information.

Supervised Learning

The effects of supervised learning split into two categories:

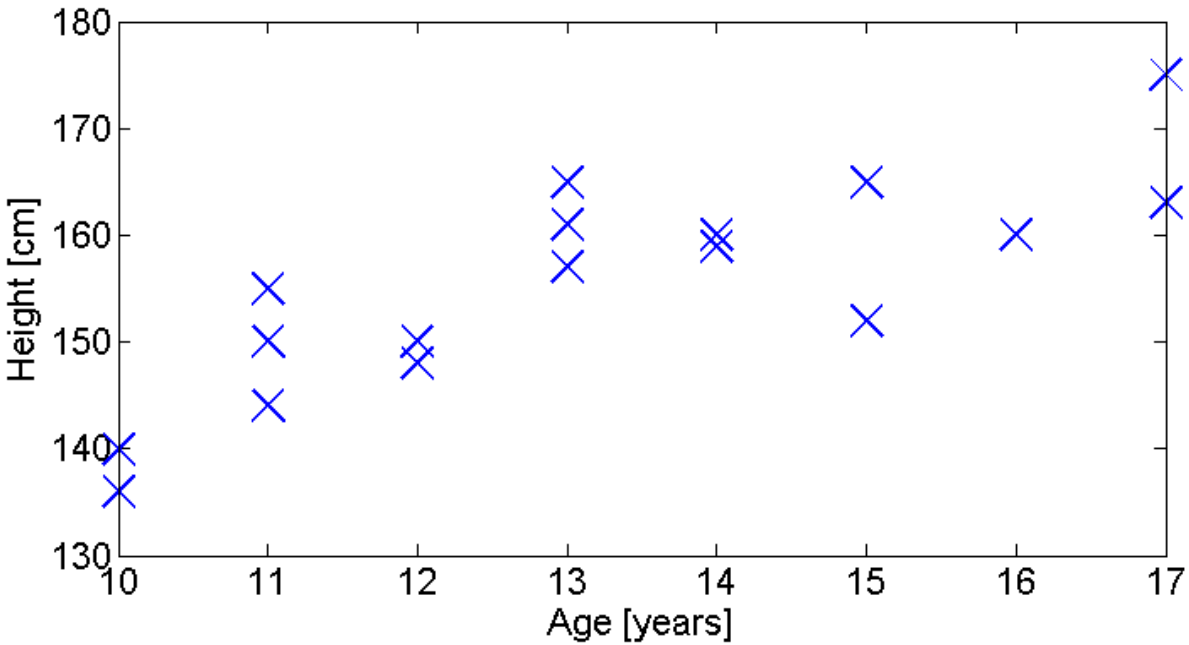
- Regression, the goal is to predict a continuous measurement for an data, i.e. temperature for next month,
- Classification, the goal is to assign a class from a finite set of classes to an data, i.e. an email is genuine or spam.

Examples of supervised machine learning algorithms:

- Linear and logistic regression (regression),
- Support vector machines (classification),
- Nearest neighbors, kNN (classification),
- Random forest (classification and regression).

Supervised Learning: regression

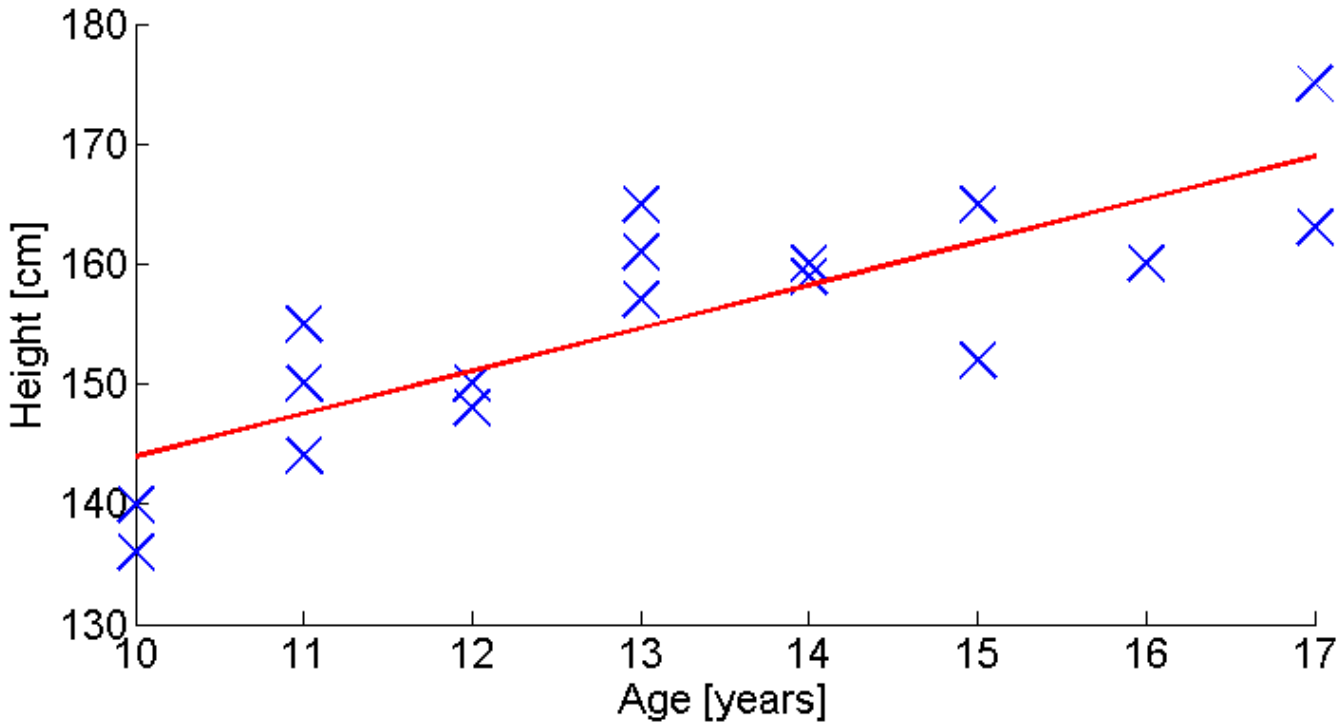
Example: age vs height



In supervised learning model should match data.

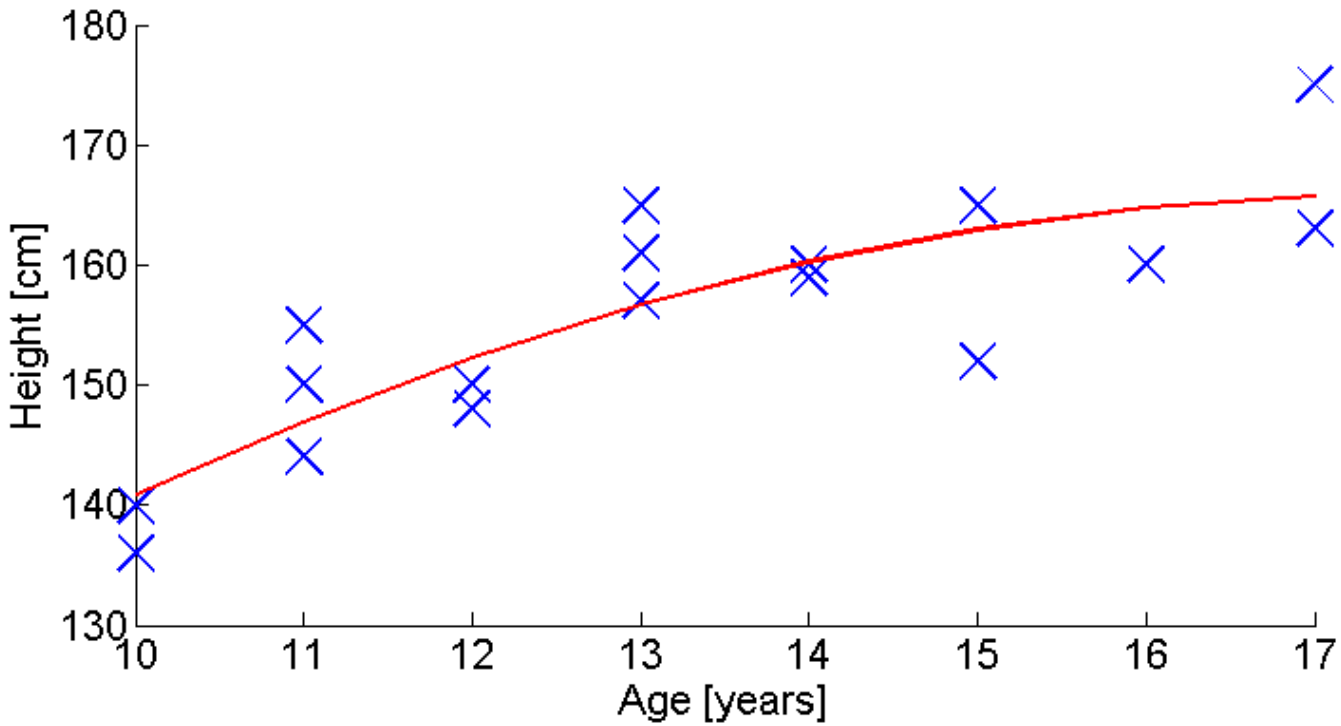
Supervised Learning: regression

Model: Linear function ?



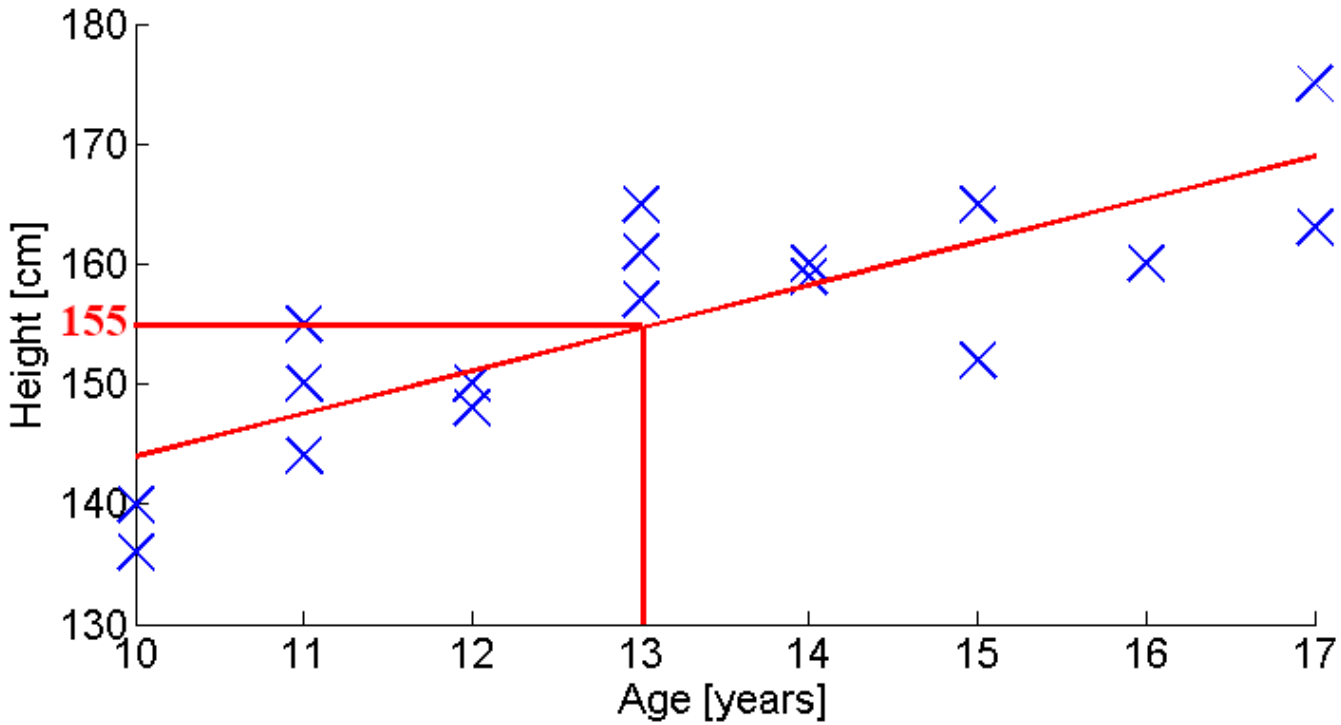
Supervised Learning: regression

Model: Quadratic function ?



Supervised Learning: regression

Goal: Provide the most accurate answer.

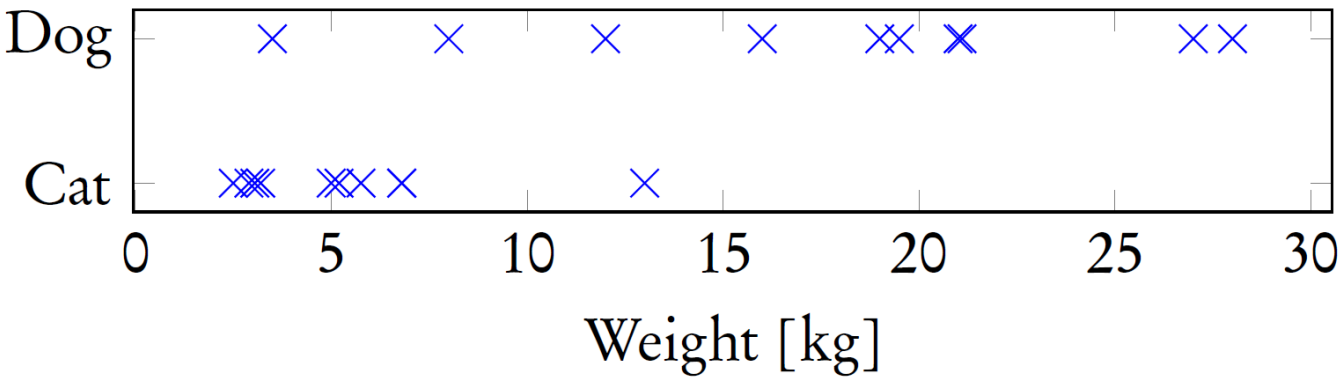


Supervised Learning: approximation

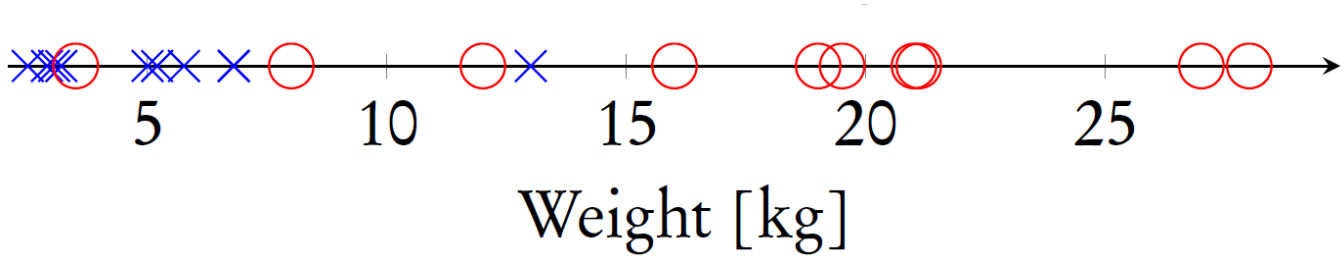
Example: Generate the images

Supervised Learning: classification

- Example: pets weight vs species

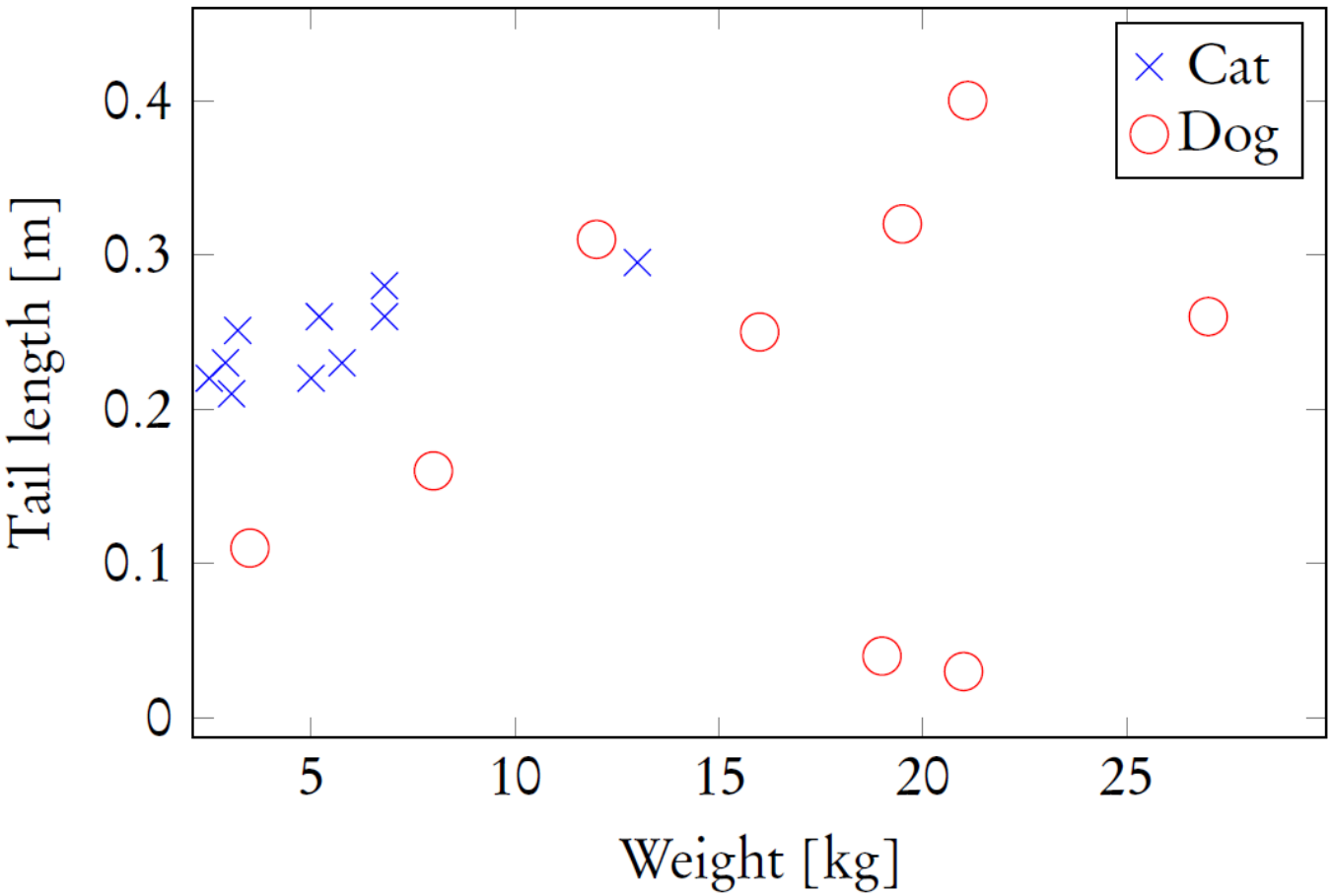


- Alternative form of presenting above data,



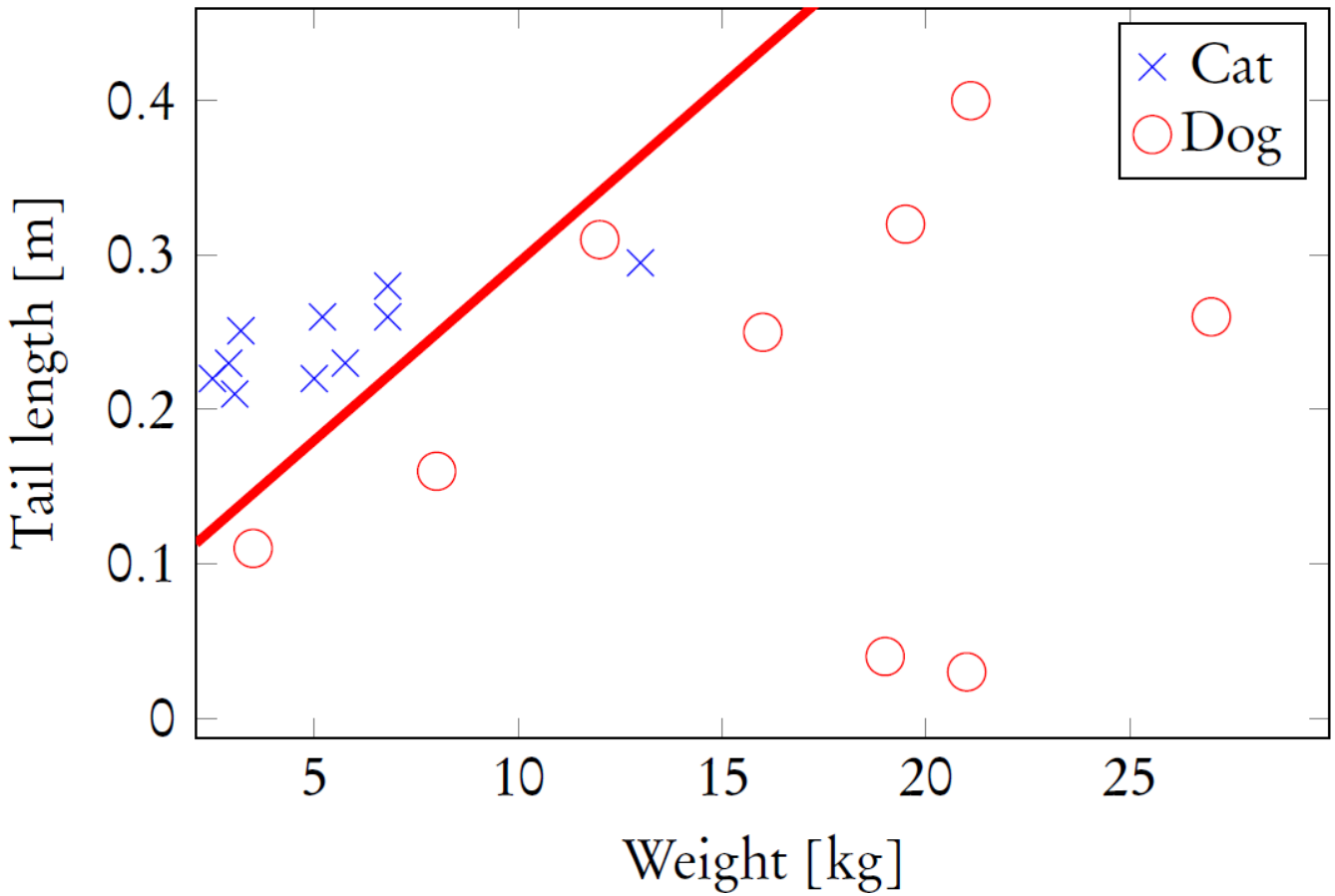
Supervised Learning: classification

Example: dog tail length vs weight



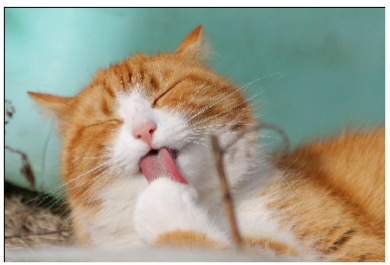
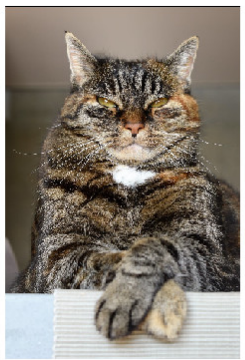
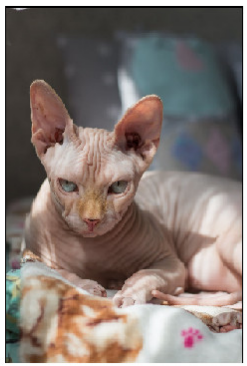
Supervised Learning: classification

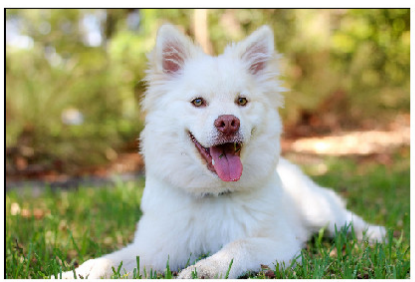
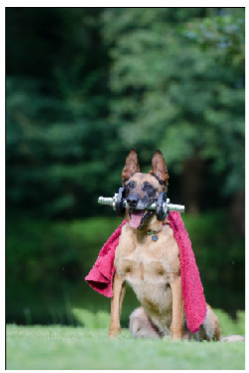
More features, as well as categories are always possible.

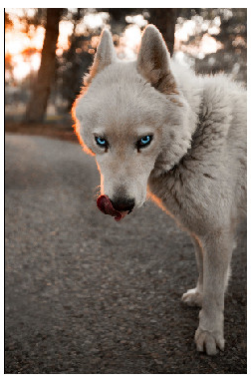


Supervised Learning: classification

Example: dog image vs cat image



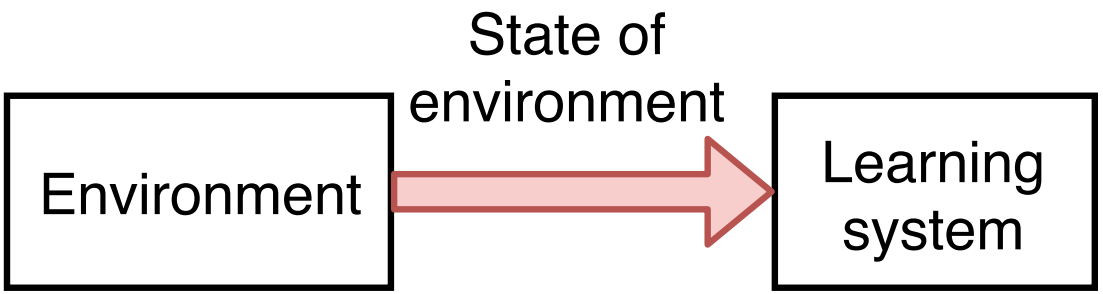

Unsupervised Learning

In unsupervised learning, there is no external teacher or critic to oversee the learning process:

- describe hidden structure from unlabeled data without additional knowledge about data relations,
- the goal is to find underlying structure or distribution in the data in order to learn more about the data itself,
- common use-cases are exploratory analysis (data mining) and dimensionality reduction,
- No evaluation of the accuracy of the output structure (Number of classes?).

Unsupervised Learning

The diagram of unsupervised learning is very simple.



Unsupervised Learning

The effects of unsupervised learning split into following categories:

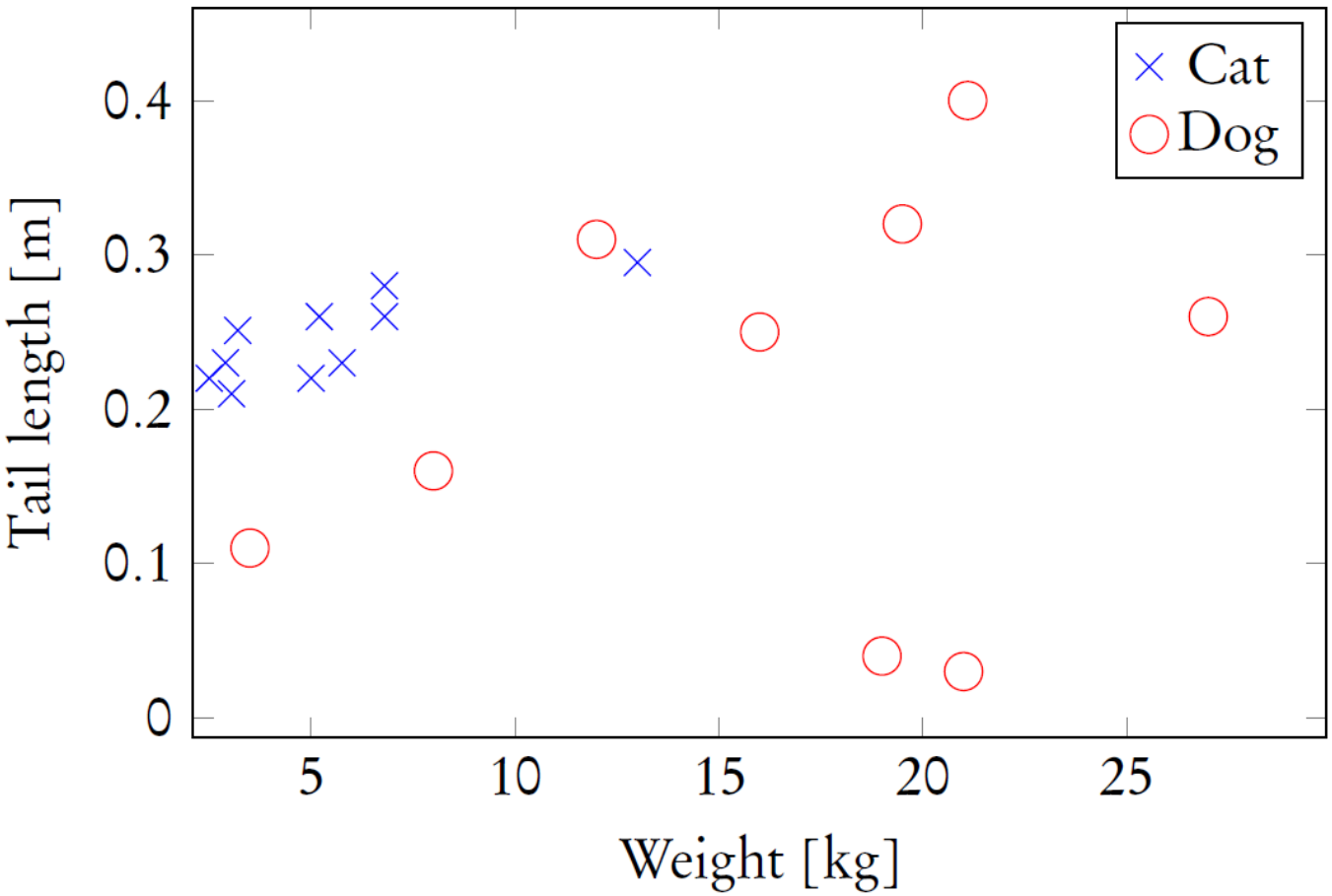
- Clustering - exploratory data analysis to find hidden patterns or groupings in data,
- Dimension reduction - compress the data while maintaining its structure and usefulness,
- Association rules - analysis rules that describe large portions of your data.

Examples of unsupervised machine learning algorithms:

- K-means (clustering),
- Principal component analysis, PCA (dimension reduction),
- Apriori algorithm (association rules).

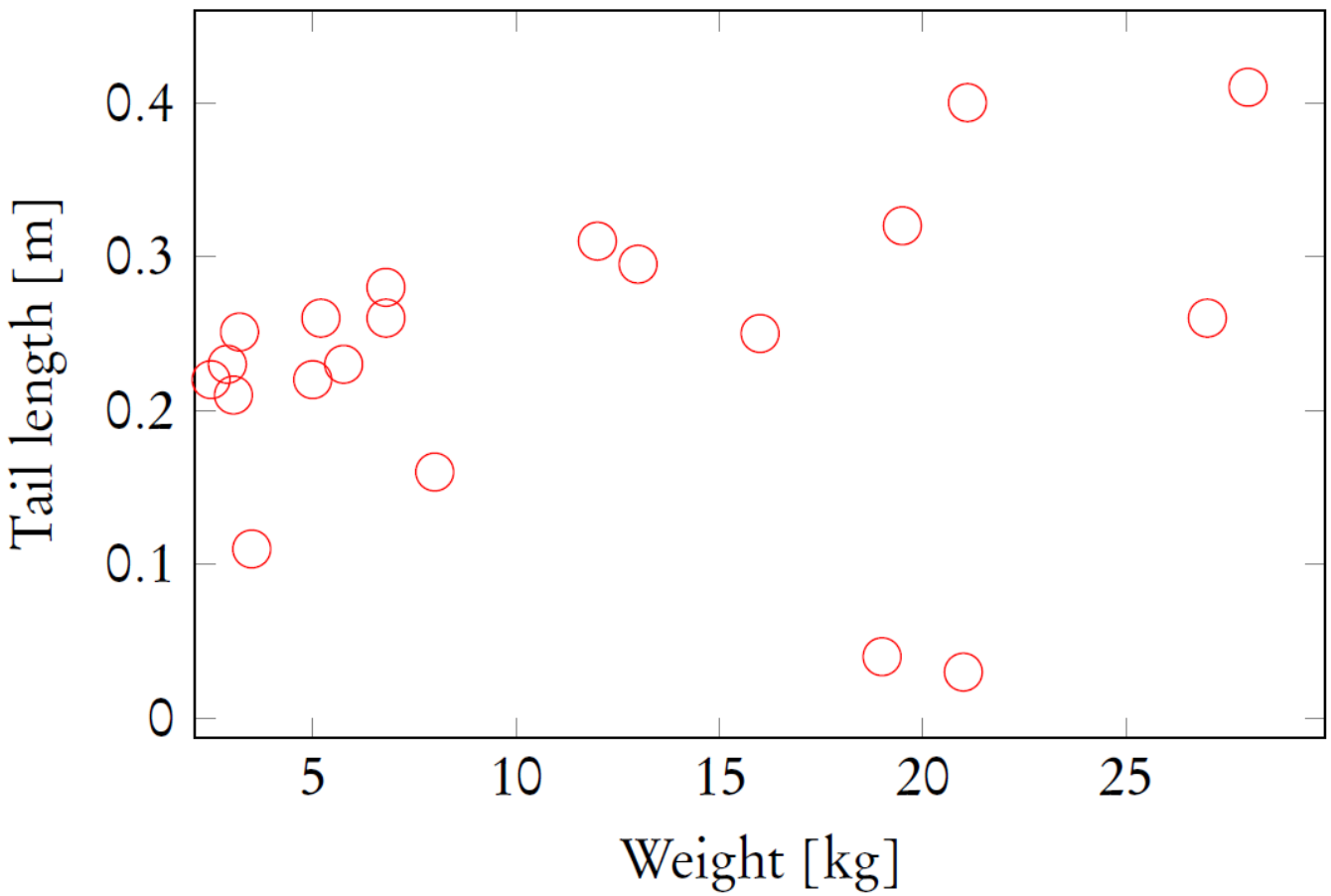
Unsupervised Learning: clustering

Example: dog tail length vs weight



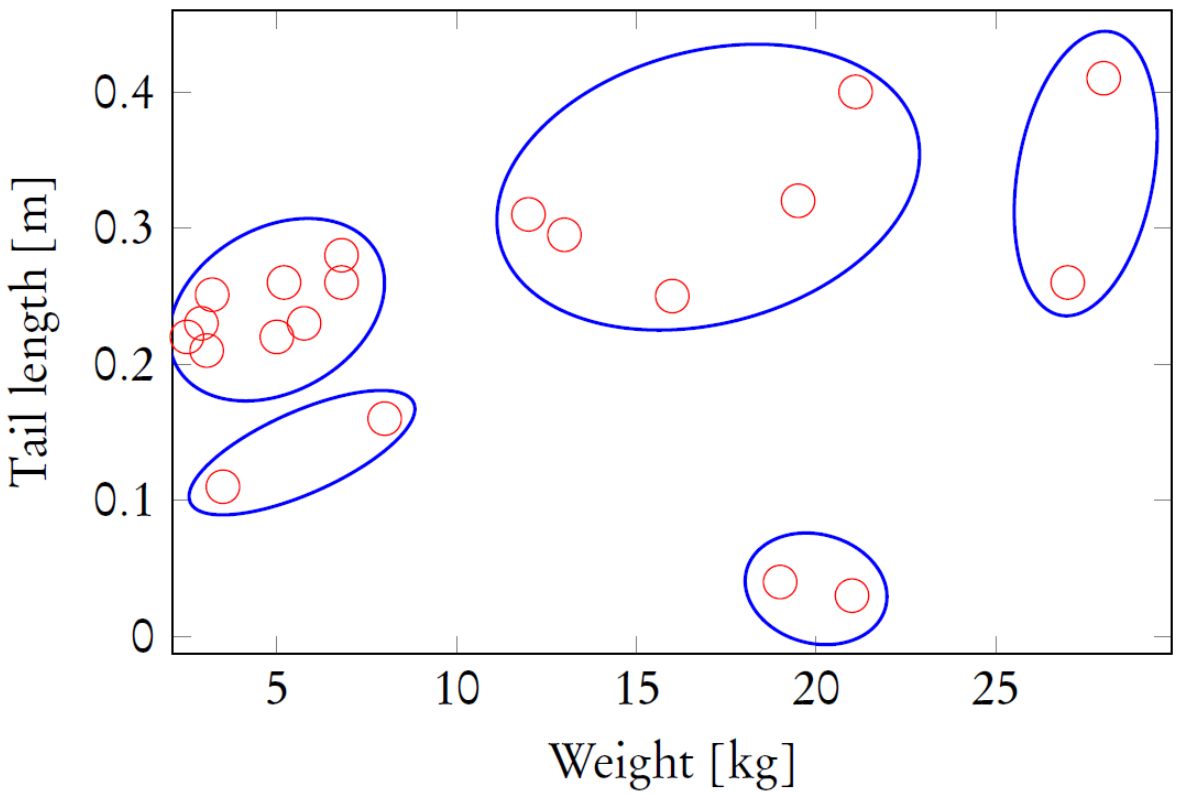
Unsupervised Learning: clustering

No categories (classes) or there is only one type of data.



Unsupervised Learning: clustering

Task of grouping a set of objects in such a way that objects in the same group are more similar to each other than to those in other groups.



Reinforcement Learning

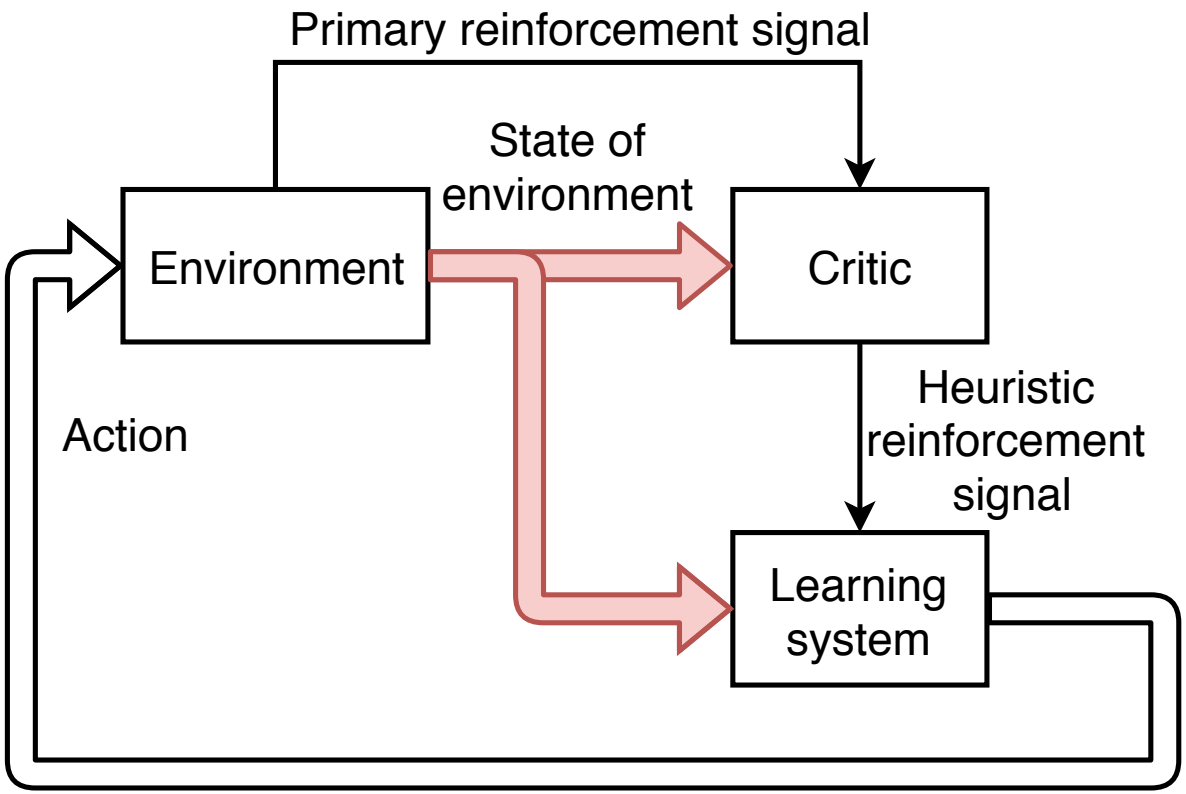
Characteristic of reinforcement Learning:

- Goal oriented - algorithm learns how to attain a complex objective over many steps,
- Optimization built on observation - judges actions by the results they produce,
- Sequential in terms of state-action pairs that occur one after the other,
- Solves the difficult problem of correlating immediate actions with the delayed returns they produce.

Reinforced learning algorithms allow to determine the best strategy in games (e.g. chess).

Reinforcement Learning

In reinforcement learning, the learning of an input–output mapping is performed through continued interaction with the environment in order to increase performance.



Reinforcement Learning

Reinforcement learning is particularly well-suited to problems that include a long-term versus short-term reward trade-off and can be used in large environments. Environment models:

- A model of the environment is known, but an analytic solution is not available,
- Only a simulation model of the environment is given,
- The only way to collect information about the environment is to interact with it.

Reinforcement learning converts planning problems to machine learning problems.

Conclusion

Machine learning:

- Different learning approaches are a result of the different goals of learning,
- Type of learning must fit the type of problem,
- Not all types of machine learning can be used or should not be used for specific problem,
- Neural networks can be used in all presented types of machine learning, but we need to assume appropriate network architecture.

Questions