MARTA

Machine le**AR**ning **T**utori**A**l

9-10 March 2023, Belgium

M. Boxho, T. Van Hoof, N. Valminck, L. Salesses, C. Sainvitu, T. Benarama

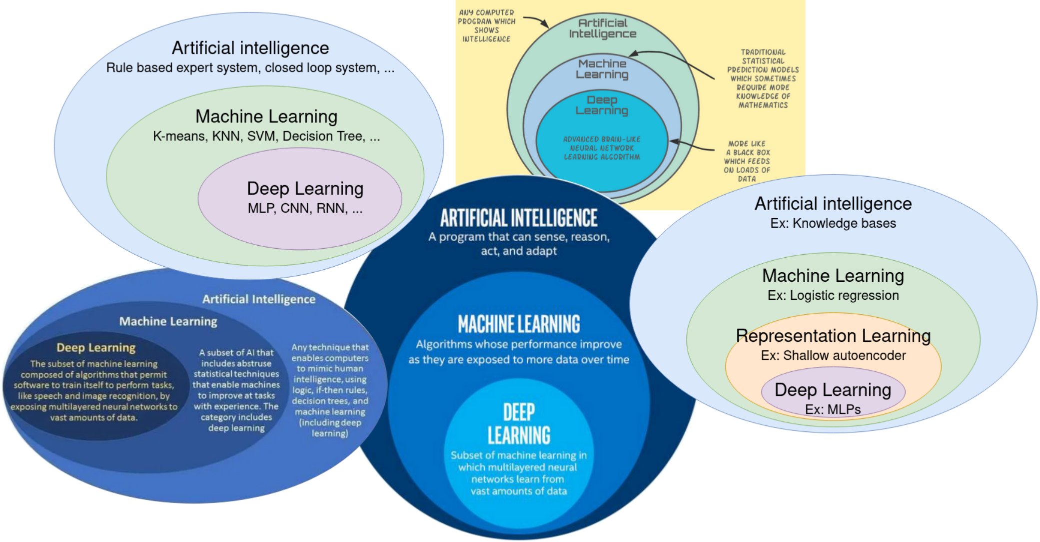
Cenaero

Contact: margaux.boxho@cenaero.be

Doc. ref.:

PROD-F-015-02

What is Machine Learning ?



PROD-F-015-02

Images extracted from [1, 2, 4, 5, 6]

Cenaero

What is Machine Learning ?

Machine Learning can be further classified in **three** categories:

Supervised | **Semi-supervised** | **Unsupervised**

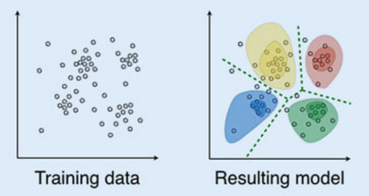
= Lion

= Cow

= Dog

= Labels

Uses a small set of **labeled** data and a larger set of **unlabeled** data, happy compromise between the two previous categories.



PCA and **POD** are also examples of unsupervised learning (=dimensionality reduction) and can also be performed with neural networks

What is Machine Learning ?

Examples of common Machine Learning algorithms:

- **Linear regression** is now part of ML techniques, even if it is used for many decades now
- **Logistic regression** is a supervised learning used to make predictions for categorical response variables
- **Clustering** is an unsupervised learning that identifies patterns in data to group them
- **Decision trees** are used both for regression problems (prediction of numerical values) and for classification (branching sequence of linked decisions)
- **Random forests**, the machine learning algorithm predicts a value or category by combining the results from several decision trees.
- **Neural networks** can be seen as "*the way the human brain works*"

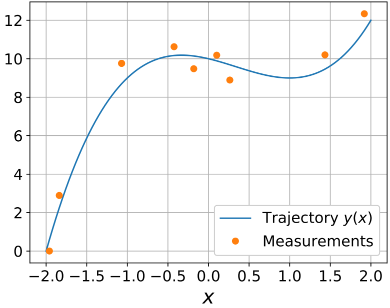
Cenaero

- **Speech recognition**
  - to conduct voice search (e.g., Siri)
  - to translate human speech into a written format
- **Computer vision**
  - facial recognition,
  - radiology imaging in healthcare, and
  - self-driving cars in the automotive industry.
- Applications into **Physics**
  - Space ablation model (**Tool**: LSTM)
  - Wall models and improvement of RANS models based on Direct Numerical Simulations (DNS) and Large Eddy Simulations (LES) data (**Tool**: CNN, MDN, MLP, ...)
  - Ice accretion model (**Tool**: well-chosen combination of MLP)
  - Prediction of the temperature evolution in additive manufacturing (**Tool**: Graph Neural Network)
  - ...

Let's start with a basic example of machine learning

Database

Assuming a phenomena y that is controlled by a single variable x (e.g. the trajectory a moving object with $y(x)$ the instantaneous position at time x). Let us construct a database of n pairs of values $\{(x_i, y(x_i)), i = 0, \ldots, n\}$, being a set of y measurements for different values of x.



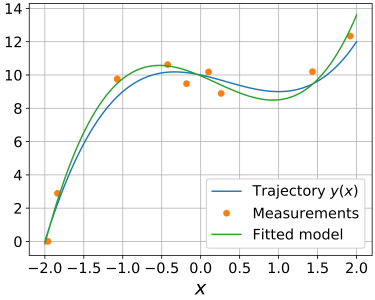
PROD-F-015-02

Cenaero

Let's start with a basic example of machine learning

Model

The database on itself is not useful. However, as an engineer, we are interested to predict the value of the phenomena y at any value of the control parameter x. We want to build a model $\hat{y}(x)$ that "*fits*" (i.e., according to a given risk) the values of the real phenomena $y(x_i)$ as recorded in the database but with generalization capabilities allowing to predict $\hat{y}$ for any x value.



Note: The model is here all polynomial functions of order 3 defined as

$$\hat{y} = p_3 x^3 + p_2 x^2 + p_1 x + p_0$$

PROD-F-015-02

Let's start with a basic example of machine learning

Method

1. **Database creation:** build, clean, and organize the available data in an adapted format
2. **Model definition:** the example above uses a polynomial fitting of order 3 such that we try to fit the four coefficients in $\hat{y} = p_3x^3 + p_2x^2 + p_1x + p_0$ according to a given risk
3. **Training the model on the database:** an optimization method is used to adjust the values of the four model parameters p_i such that $\hat{y}(x)$ is "*close*" (according to a given *metric*) to the measured value $y(x)$ for each of the point $(x, y(x))$ in the database
   - Define a minimization function $\mathcal{L}$ (e.g., MSE)
   - Setup an optimization method to adapt the four parameters (e.g., gradient descent):

   $$p_{i,t+1} = p_{i,t} - \gamma \left(\frac{\partial \mathcal{L}}{\partial p_i}\right)_t \quad \text{where } \gamma \text{ is the learning rate}\,.$$

   - Loop over the database to reach the desired model accuracy
   - Save the '*trained model*' for future use
4. **Test** the '*trained model*' on unseen data

PROD-F-015-02

More complicated example of machine learning

Remark

The previous example was an easy one-to-one relation. However, real-world problems are not as simple as that.

Question

How can an algorithm be implemented to recognise low-resolution handwritten digits (see figure on the right [3])?

PROD-F-015-02

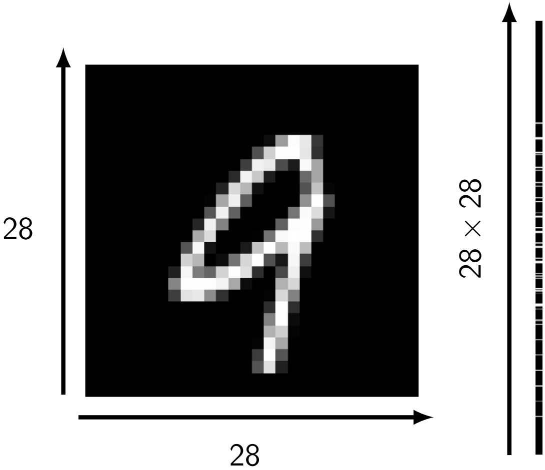
Digits classification using (deep) neural network

The human brain can quickly distinguish a three, a one, or a nine, but how can a machine make the same distinction as us? This is where deep learning appears as *almost miraculous*. Deep neural networks are large artificial neural networks (ANN) that claim to combine both

- automatic feature engineering, and,
- **universal approximation** capabilities.

Thanks to their hidden and activation layers, ANN are able to identify relevant features and their functional relationships with limited human intervention. However, the remaining **drawback** is the size of the database that needs to be large enough to capture the relevant phenomena. You will also have to deal with a list of **hyperparameters** to tune.

More complicated example of machine learning



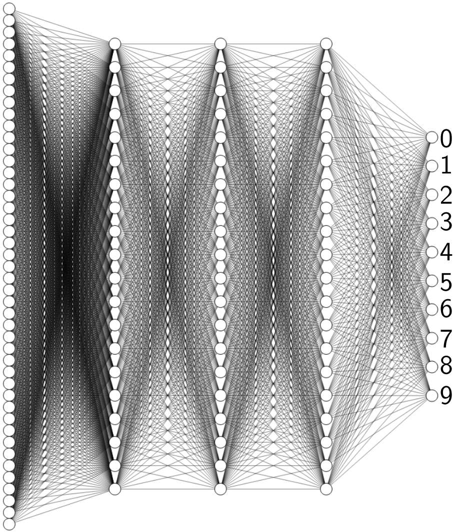


Image extracted from MNIST database and neural network generated on https://alexlenail.me/NN-SVG/

After this brief introduction about Machine Learning model, we can dive into the tutorial which is construct according to six sections:

1. The **first** section is dedicated to the `Pytorch` data structures. Through this section, we will see how to construct a tensor and how to perform operations on them.

After this brief introduction about Machine Learning model, we can dive into the tutorial which is construct according to six sections:

1. The **first** section is dedicated to the `Pytorch` data structures. Through this section, we will see how to construct a tensor and how to perform operations on them.
2. The **second** section is devoted to the construction of a neural network through the definition of `torch.nn.Module`.

Cenaero

After this brief introduction about Machine Learning model, we can dive into the tutorial which is construct according to six sections:

1. The **first** section is dedicated to the `Pytorch` data structures. Through this section, we will see how to construct a tensor and how to perform operations on them.
2. The **second** section is devoted to the construction of a neural network through the definition of `torch.nn.Module`.
3. The **third** section is focus on `torch.autograd` which is the automatic differentiation package. It is an essential package for our model optimization.

After this brief introduction about Machine Learning model, we can dive into the tutorial which is construct according to six sections:

1. The **first** section is dedicated to the `Pytorch` data structures. Through this section, we will see how to construct a tensor and how to perform operations on them.
2. The **second** section is devoted to the construction of a neural network through the definition of `torch.nn.Module`.
3. The **third** section is focus on `torch.autograd` which is the automatic differentiation package. It is an essential package for our model optimization.
4. The **fourth** section will discuss the optimization of a neural network (i.e., how to fit the parameters of the network based on the training data and the proper definition of a loss function)

After this brief introduction about Machine Learning model, we can dive into the tutorial which is construct according to six sections:

1. The **first** section is dedicated to the `Pytorch` data structures. Through this section, we will see how to construct a tensor and how to perform operations on them.
2. The **second** section is devoted to the construction of a neural network through the definition of `torch.nn.Module`.
3. The **third** section is focus on `torch.autograd` which is the automatic differentiation package. It is an essential package for our model optimization.
4. The **fourth** section will discuss the optimization of a neural network (i.e., how to fit the parameters of the network based on the training data and the proper definition of a loss function)
5. A neural network is nothing without its database. Hence, the **fifth** section is concerned with the definition of database (i.e., training, validation, and testing ones).

Cenaero

After this brief introduction about Machine Learning model, we can dive into the tutorial which is construct according to six sections:

1. The **first** section is dedicated to the `Pytorch` data structures. Through this section, we will see how to construct a tensor and how to perform operations on them.
2. The **second** section is devoted to the construction of a neural network through the definition of `torch.nn.Module`.
3. The **third** section is focus on `torch.autograd` which is the automatic differentiation package. It is an essential package for our model optimization.
4. The **fourth** section will discuss the optimization of a neural network (i.e., how to fit the parameters of the network based on the training data and the proper definition of a loss function)
5. A neural network is nothing without its database. Hence, the **fifth** section is concerned with the definition of database (i.e., training, validation, and testing ones).
6. The **sixth** section ... it's your turn.

Nomenclature[1]

- **Input** is a p-dimensional vector of features or descriptors.
- **Output** is the prediction of a model (e.g., a scalar value, a label, an image, a signal, ...).
- **Target** is the feature of a dataset about which you want to gain a deeper underdstanding.
- **Epoch** is a N/batch size training iterations, where N is the size of training database (i.e., number of training samples). A full training pass over the entire training set such that each sample of the training database is visited.
- **Batch** is a set of samples used in one training iteration.
- **Batch size** corresponds to the number of samples contained in a batch.
- **Learning rate** is a floating point which is multiplied to the gradient to adjust the weights and biases on each iteration.
- **Layer** is a set of neurons in a neural network. There exists three common layers: the input layer, the hidden layer and the output layer. A model underfits the training data if it has poor prediction abilities because it has not fully captured the training data complexity.

[1] Inspired by `https://developers.google.com/machine-learning/glossary$#$h`

- **Backpropagation** is an algorithm that implements the gradient descent based on the chain rule for a neural network.
- **Neuron** is the basic unit of computation in a neural network, aslo called node or unit.
- **Weights and Biases** are parameters learned during the training.
- **Hyperparameter** is a parameter whose value is used to control the learning process (e.g. the learning rate).
- **Loss function** measures how far the model's prediction is from its target value (i.e., its label) according to a given metric (e.g. MSE, Cross Entropy, ...).
- **Activation function** is the key ingredient to help neural network to learn nonlinear (complex) relationship between the features (=inputs) and the label (=targets).
- **Overfitting** A model overfits the training data if the predictions on it are so closely that it fails to make correct predictions on unseen data.
- **Underfitting** A model underfits the training data if it has poor prediction abilities because it has not fully captured the training data complexity.

[1] Inspired by `https://developers.google.com/machine-learning/glossary$#$h`

A. Abdullah.
AI vs ML vs DL Venn Diagram.
https://imgur.com/t/artificial_intelligence/kfDt6dZ.

S. Bhandarkar.
Deep Learning Demystified.
https://medium.com/@sanatbhandarkar75/deep-learning-demystified-4a7554a5c3ba.

N. Y. D. by an MIT license.
MNIST Handwritten Digits Recognition using PyTorch.
https://nvsyashwanth.github.io/machinelearningmaster/digits-mnist/.

R. G. Khalkar, A. S. Dikhit, A. Goel, and M. Gupta.
Handwritten text recognition using deep learning (CNN & RNN).
International Advanced Research Journal in Science, Engineering and Technology, 8(1), 2021.

A. Kumar.
AI – Three Different types of Machine Learning Algorithms.
https://vitalflux.com/ai-types-machine-learning-algorithms/.

H.-C. Lee and C.-W. Jung.
Anesthesia research in the artificial intelligence era.
Anesth Pain Med, 13:248–255, 2018.

Cenaero