

Deep Learning for Image Analysis - Introduction

Thomas Walter, PhD

Centre for Computational Biology (CBIO)
MINES Paris-Tech, PSL Research University
Institut Curie, PSL Research University
INSERM U900

Overview

- 1 Introduction: Artificial Intelligence and Machine Learning
- 2 Design Principles of Machine Learning algorithms
- 3 Supervised Learning: Example algorithms
 - Nearest Neighbor classification
 - Random Forests
 - Linear Discriminant Analysis (LDA)
 - Support Vector Machines (SVM) and kernel methods
- 4 References

Overview

1 Introduction: Artificial Intelligence and Machine Learning

2 Design Principles of Machine Learning algorithms

3 Supervised Learning: Example algorithms

- Nearest Neighbor classification
- Random Forests
- Linear Discriminant Analysis (LDA)
- Support Vector Machines (SVM) and kernel methods

4 References

Definition of Artificial Intelligence and Machine Learning

- The definition of the term **intelligence** is highly controversial. Usually, one understands by intelligence the capacity of an individual to reason logically, to understand complexity, to learn more or less abstract concepts, to plan and to solve problems in varying conditions.
- **Artificial intelligence (AI)** is intelligence demonstrated by machines, in contrast to the natural intelligence displayed by humans and other animals.
- AI effect: *"AI is whatever hasn't been done yet"*.
- In 1956, AI became a field of research. AI can be broken down into many subfields: knowledge representation, planning, natural language processing, object manipulation (robotics), machine learning, ...
- **Machine Learning** is concerned with the technology that enables computer programs to improve their performance at a certain task by experience.

A short history of artificial intelligence

- The dream to create machines that can think and act has been present in literature and mythology since antiquity (e.g. the myth of "*Talos*" or "*Rossum's Universal Robots*" by Karel Čapek, 1920).
- AI as a scientific discipline took its beginning with the publication of Alan Turing in 1950 (Turing test). The question "*Can machines think?*" was asked scientifically.
- Other theoretical bases were developed by Wiener (cybernetics) and Shannon (information theory).
- The term "*artificial intelligence*" was coined in 1956, in the famous Dartmouth conference, where AI has been established as a field.

A short history of machine learning

- The theoretical foundations go back to the early 19th century (e.g. Bayesian theory and Least Squares).
- Even before machines came into play, there was a lot of interest in finding methods to derive rules from data (*data fitting*). These methods are part of what we call Machine Learning today (linear regression, Bayesian theory, Logistic regression, Linear discriminant analysis, Markov chains).
- In 1958, Rosenblatt published his *Perceptron*.

Machine Learning: basic definitions

- Machine Learning aims at predicting some output y from an input (or measurement) x :

$$y = f(x) \tag{1}$$

- In this formulation, Machine Learning aims at finding (learning) f from available data.
- The data that is used to learn f is called **training set**.
- In this general formulation, there is no particular limitation as to the mathematical nature of x and y . In many cases x is a P -dimensional vector and y a categorical or continuous output variable, but there are other settings, where x and / or y are more complicated objects, such as images or graphs.

Different settings in Machine Learning

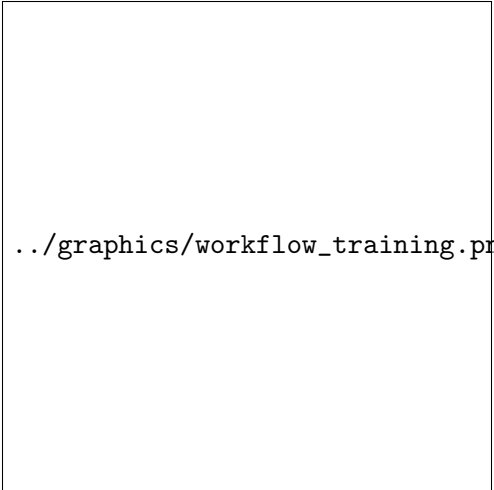
	Supervised	Unsupervised
y discrete	Classification	Clustering
y continuous	Regression	Dimensionality reduction

- In **supervised learning**, the training data contains both measurements x_i and the corresponding output variables y_i . Together, they build the training set T :

$$T = \{(x_i, y_i)\}_{i=1, \dots, N} \quad (2)$$

- In **unsupervised learning**, there are no annotations y_i . We aim at inferring **patterns** from the data (clusters, latent variables).

Training and prediction



`../graphics/workflow_training.png`

(a) Training a classifier

Objects

Features

Model

Classification
result

Objects and features

- Machine learning typically deals with objects outside the mathematical world (emails, images, genomes, cars, ...).
- The first step is therefore to find a suitable representation of the objects.
 - **feature engineering**: finding descriptors according to existing domain knowledge
 - **representation learning**: learning the descriptors together with the classifier
- In many cases the objects can be represented by a P -dimensional vector of features (or descriptors): $\mathbf{x} \in \mathbb{R}^P$.
- It can be convenient to map a feature vector to a higher dimensional space:

$$\phi : \mathbb{R}^P \rightarrow \mathbb{R}^Q \quad (3)$$

$$\mathbf{x} \rightarrow \phi(\mathbf{x}) \quad (4)$$

Training set and design matrix

- In the frequent case that objects can be described by a feature vector $\mathbf{x} \in \mathbb{R}^P$, we can represent the training set $T = \{(x_i, y_i)\}_{i=1, \dots, N}$ by a $N \times P$ **design matrix** \mathbf{X} and an output vector y .
- In the design matrix, each row corresponds to one sample, each column corresponds to one feature:

	feature 1	feature 2	...	feature P
Sample 1	0.23	1.30	...	0.01
Sample 2	0.42	1.15	...	-0.23
\vdots	\vdots	\vdots	\vdots	\vdots
Sample N	0.31	1.53	...	0.33

Example: classification of flowers



(a) *Iris setosa*



(b) *Iris versicolor*



(c) *Iris virginica*

- One of the oldest data sets in machine learning is the Iris data set, that was collected by the statistician and biologist Ronald Fisher in 1936 [?].
- There are 3 classes (different types of the Iris flower). For each class, there were 50 samples collected. For each sample, 4 characteristics were measured (lengths and widths of different parts of the plants).
- The task is thus to learn a rule to predict the type of the flower (y) from a 4-dimensional vector x of geometric measurements.

Example: classification of SPAM emails

Dear **thomas.walter@mines-paristech.fr**,

Your mailbox is almost full.

1969MB 2000MB

We noticed your E-mail account has almost exceed it's limit. And you may not be able to send or receive new messages until you re-validate,

[Click Here to Re-Validate.](#)

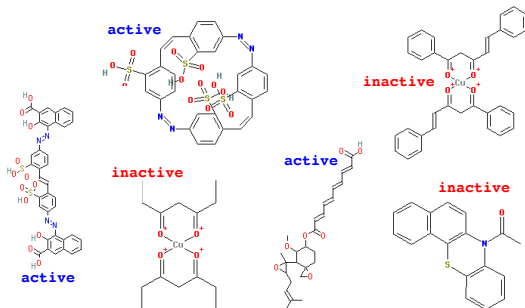
WARNING:

failure to re-validate your E-mail account. It will be permanently disable.

Thanks,
Account Service

- This is a binary classification problem: $y \in \{0, 1\}$ (0: junk, 1: not junk).
- The features can be constructed in the following way: for each email annotated by the user, the words are listed. An email is described as a vector of frequencies of these words.
- The system learns then a function that assigns to each vector of measured word frequencies the label y .

Example: classification of drugs



- Here, we want to classify molecules with respect to their efficiency against a disease (binary classification: a drug is efficient or not).
- An important question here is how to encode a molecule. One option is to define chemoinformatic features and obtain a vectorial representation of the molecule $\mathbf{x} \in \mathbb{R}^P$.

Overview

1 Introduction: Artificial Intelligence and Machine Learning

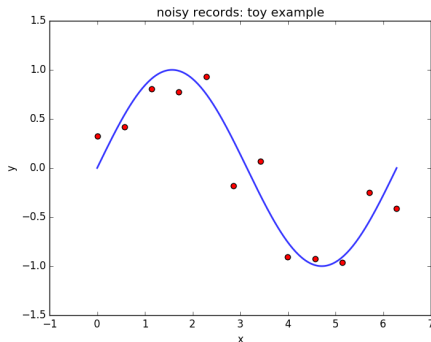
2 Design Principles of Machine Learning algorithms

3 Supervised Learning: Example algorithms

- Nearest Neighbor classification
- Random Forests
- Linear Discriminant Analysis (LDA)
- Support Vector Machines (SVM) and kernel methods

4 References

A simple example: polynomial curve fitting¹



- From a set of measured points (x_i, y_i) (red), we would like to build a model to predict the value y for any given x .
- The true function is $g(x) = \sin(x)$ (displayed in blue).
- The measurements y_i are noisy outputs of that function, i.e.

$$y_i = \sin(x_i) + \epsilon, \quad \epsilon \sim \mathcal{N}(0, 0.2) \quad (5)$$

¹Example adapted from [?]

A simple example: polynomial curve fitting

- We use the following polynomial model:

$$\begin{aligned}f(x) &= a_0 + a_1x + a_2x^2 + \dots + a_mx^m \\ &= \boldsymbol{\theta}^T \boldsymbol{\phi}(x)\end{aligned}\tag{6}$$

- Parameter vector: $\boldsymbol{\theta} = (a_0, a_1, \dots, a_m)^T$
- Here, we do not only use the measurement x . We actually map x to a higher dimensional space:

$$\begin{aligned}\boldsymbol{\phi} : \mathbb{R}^P &\rightarrow \mathbb{R}^Q \\ x &\rightarrow \boldsymbol{\phi}(x) = (1, x, x^2, \dots, x^m)^T\end{aligned}\tag{7}$$

In our example there is only one feature to start with: $P = 1$. But high dimensional feature mappings are also commonly used for $P > 1$.

- The model is linear in the parameters (but for $m > 1$ not in the inputs).

A simple example: polynomial curve fitting

- One classical approach is to minimize the least squared error between measured and predicted values:

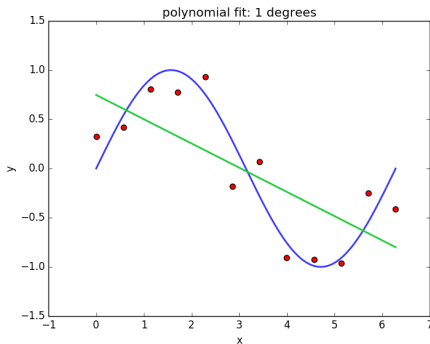
$$\begin{aligned}\min_{\theta} L(\theta) &= \min_{\theta} \sum_{i=1}^N (y_i - f(x_i))^2 \\ &= \min_{\theta} \sum_{i=1}^N (y_i - \theta^T \phi(x_i))^2\end{aligned}\quad (8)$$

- This can be achieved by setting the gradient with respect to θ to zero:

$$\nabla_{\theta} L = \left(\frac{\partial L}{\partial a_0}, \frac{\partial L}{\partial a_1}, \dots, \frac{\partial L}{\partial a_m} \right)^T = 0 \quad (9)$$

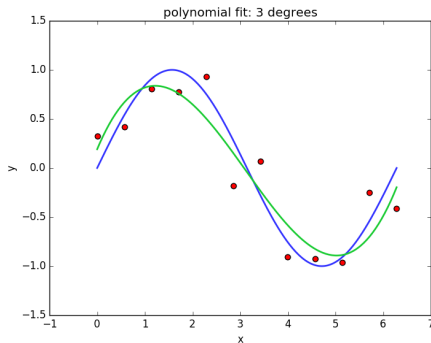
- Unlike for most optimization problems in this course, this leads to an analytical solution for θ . This is known as **linear regression**. For more details, we refer to [?].

Overfitting and underfitting



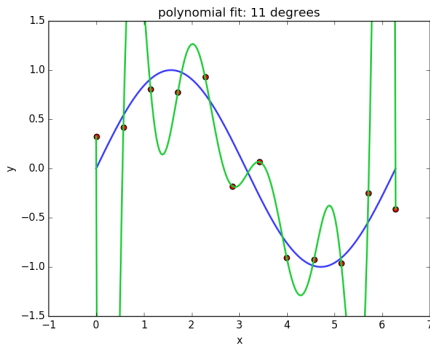
For $m = 1$, the model is linear in its inputs. The solution is not capable of modeling the measured data points; we get a poor approximation of the original function. The family of functions we have used was not complex enough to model the true data distribution. We also speak of **underfitting**.

Overfitting and underfitting



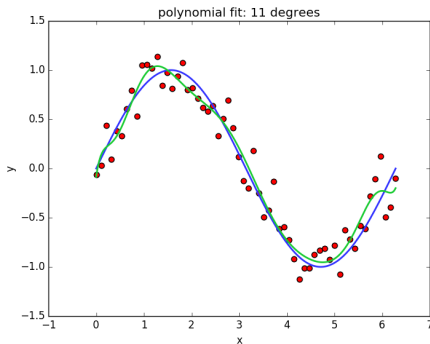
For $m = 3$, we obtain a solution that seems to be quite right: it is sufficiently complex to model the true data distribution, but not too complex to model the small variations which are due to noise.

Overfitting and underfitting



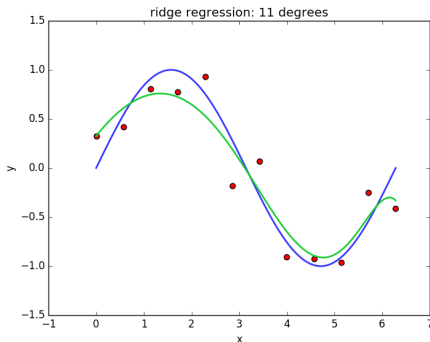
For $m = 11$, we obtain a solution that has zero error (the function passes through every point of the training set). But the coefficients with large absolute values that cancel each other precisely on the training points lead to a highly unstable function. We speak of **overfitting** and **poor generalization**.

Overfitting and underfitting



One way of reducing overfitting is to increase the number of samples. Even if the function is complex, it cannot be “too wild”, as it has to find a compromise between many training samples. This however implies the annotation (or measurement) of more samples.

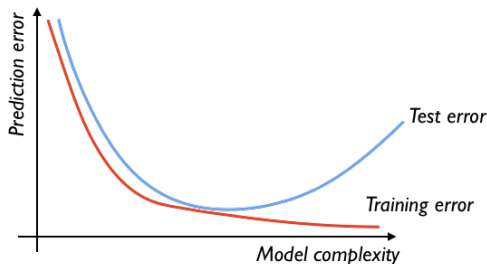
Overfitting and underfitting



Another way of preventing overfitting without increasing the number of samples, is to add a penalization term in the optimization procedure. This is also known as **regularization**:

$$L = \sum_{i=1}^N (y_i - \theta^T \phi(x_i))^2 + \lambda \|\theta\|^2 \quad (10)$$

Generalization: training and test error



- Supervised Learning aims at finding a function f that predicts an output value y from a measurement x for unseen data, i.e. for data that has not been used to find f .
- Machine Learning is much concerned with avoiding f to **memorize** the training set, i.e. to perform well on a training set but poorly on a test set.
- An important paradigm is that we must never evaluate the performance of our machine learning method on the data that has been used to train it.

Generalization: strategies

- Many ML algorithms can be written as an optimization problem:

$$\theta^* = \arg \min_{\theta} L(\theta) + \mathcal{R}(\theta) \quad (11)$$

Minimizing the loss $L(\theta)$ aims at finding the rule to reproduce the annotations in the training set, minimizing the regularization term $\mathcal{R}(\theta)$ aims at avoiding the model to adapt too much to the training data, leading to simpler models. We have seen the L_2 norm, but there are many other options for \mathcal{R} .

- Other regularization strategies include:
 - Model averaging (ensemble methods)
 - Artificial or actual increase of training data
 - Adversarial training

References I

- [Bishop, 2006] Bishop, C. M. (2006). *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag, Berlin, Heidelberg.
- [Fisher, 1936] Fisher, R. A. (1936). The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7(2):179–188.
- [Hastie et al., 2009] Hastie, T., Tibshirani, R., and Friedman, J. (2009). *The elements of statistical learning: data mining, inference and prediction*. Springer, 2 edition.