



# Introduction to Machine Learning

Pavel Druzhdov

Internet of Things Group

# What is machine learning?

- “Field of study that gives computers the ability to learn without being explicitly programmed”.

Arthur Samuel, 1959.

- “A computer program is said to learn from experience  $E$  with respect to some class of tasks  $T$  and performance measure  $Q$  if its performance at tasks in  $T$ , as measured by  $Q$ , improves with experience  $E$ ”.

Tom M. Mitchell, 1997.

# Essentials

In machine learning we want to build an algorithm that will analyze *data* and extract useful *information* out of it.

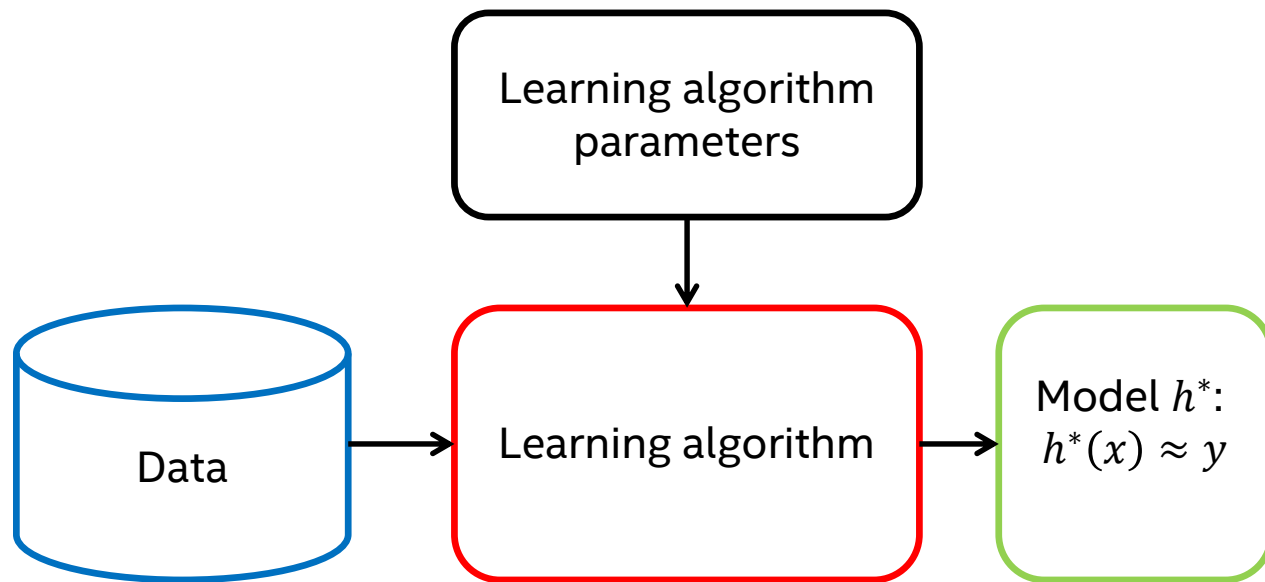
Depending on the type of data we have different machine learning:

- supervised learning,
- reinforcement learning,
- unsupervised learning,
- etc.

# Supervised learning

- Space of viable input values  $\mathcal{X}$ .
- Space of viable output (target) values  $\mathcal{Y}$ .
- Dataset  $\mathcal{D} \subset \mathcal{X} \times \mathcal{Y}$ ,  $|\mathcal{D}| = N$ .
- Find a function (a.k.a. model)  $h: \mathcal{X} \rightarrow \mathcal{Y}$  that is a good predictor of  $y \in \mathcal{Y}$  given  $x \in \mathcal{X}$ .

# Supervised learning



# Training and testing

One should clearly distinguish between:

- **Training** (a.k.a. learning or model fitting) phase.
  - Here we select the best model  $h^*$ .
  - As a rule is extremely computation (time) and memory consuming part especially in the context of deep learning. Often takes place in datacenters.
- **Testing** (a.k.a. prediction or inference or deployment or scoring) phase.
  - Here we evaluate a fixed function  $h^*$  at some (arbitrary) points  $x$ .
  - Require much more light-weight computations, but should work on embedded devices with limited computing power, memory and energy consumption and often in real-time.

# Feature space

- $x = (x_1, x_2, \dots, x_d) \in \mathcal{X}$ ,  
where  $x_j$  is a feature that describes some object property.
- $\mathcal{X} = P_1 \times P_2 \times \dots \times P_d$ ,  
where  $P_j$  is a set of viable values of feature  $j$ .
- $P_j$  (and consequently  $\mathcal{X}$ ) may have a difficult structure:
  - $P_j = \mathbb{R}$ ;
  - $|P_j| < \infty$  and  $P_j$  is ordered;
  - $|P_j| < \infty$  and  $P_j$  is unordered;
  - etc.

# Algorithms and models

Supervised learning algorithm is defined by:

- Class of functions  $\mathcal{H}$  to search  $h$  in.
  - Broder family of functions increases the chance to have a good approximation in, but makes it difficult to find it.
- Quality metrics  $Q$  (or loss  $\mathcal{L}$ ).
  - One should formalize what it means that one function is better than another and qualify the difference.
- Optimization algorithm.
  - Defines a search procedure in  $\mathcal{H}$  w.r.t.  $Q$ .
  - We want to maximize  $Q$  (or minimize loss  $\mathcal{L}$ ) on all viable data, which is not available or intractable.



# Empirical quality / loss

- As long as only a sample of all viable data is available, we can try to approximate model quality (or loss) using data at hand.
- Hence we have *empirical* quality  $Q(h, \mathcal{D})$  or *empirical* loss  $\mathcal{L}(h, \mathcal{D})$  estimates and an optimization problem:

$$h^* = \max_{h \in \mathcal{H}} Q(h, \mathcal{D}) = \min_{h \in \mathcal{H}} \mathcal{L}(h, \mathcal{D}).$$

# Model generalization

- How good is empirical quality estimate?
- How good is the model that optimizes empirical quality on some training set?
- It's easy to show that optimization of empirical quality on training set does not guarantee that model generalizes to the new data.

# Model generalization

- Let's build model  $h$  in the following way:

$$h(x) = \begin{cases} y_i & \text{if } \exists x_i \in \mathcal{X} \text{ and } x_i = x, \\ \text{random } y \text{ from } \mathcal{Y}. & \end{cases}$$

- $Q(h, \mathcal{D})$  is high, but this model makes a random guess on data points outside of the training datasets, hence  $Q(h)$  is low.

# Model generalization

- Conclusions

- Model generalization for the data unseen at training time is essential.
- Measure of model generalization is required.
- Generalization measure should be incorporated into the optimization objective somehow.

# Estimate of model generalization

- Having a dataset for the same task but independent from the training one, which is sampled under some reasonable constraints (i.i.d.), may provide us an unbiased estimate of model quality (not empirical!).
- This dataset is commonly referred to as *validation* or *test* one.

# Overfitting

- Situation when  $Q(h, \mathcal{D}_{train}) \gg Q(h, \mathcal{D}_{test})$  is called overfitting.
- Overfitting is one of the main concepts in machine learning.
- Overfitting occurs when:
  - We don't have enough data.
  - Our set of viable models  $\mathcal{H}$  is too broad, and model  $h^*$  is overcomplicated.

# Overfitting

$$y(x) = \frac{1}{1+25x^2}. \quad \mathcal{H} = \{a_0 + a_1x + a_2x^2 + \dots + a_nx^n\}. \quad \mathcal{L} = \frac{1}{\#D} \sum_{(x,y) \in D} (h(x) - y)^2.$$

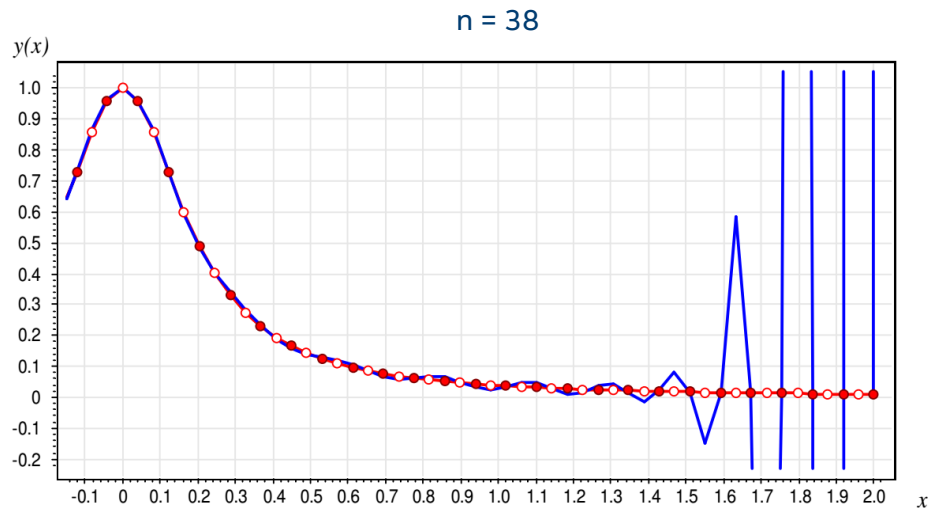
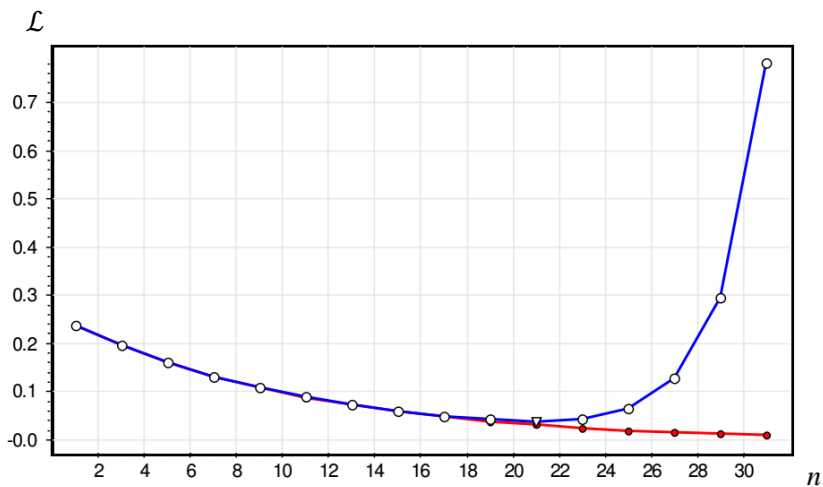


Image credits: <http://www.machinelearning.ru/wiki/images/f/fc/Voron-ML-Intro-slides.pdf>

# Regularization

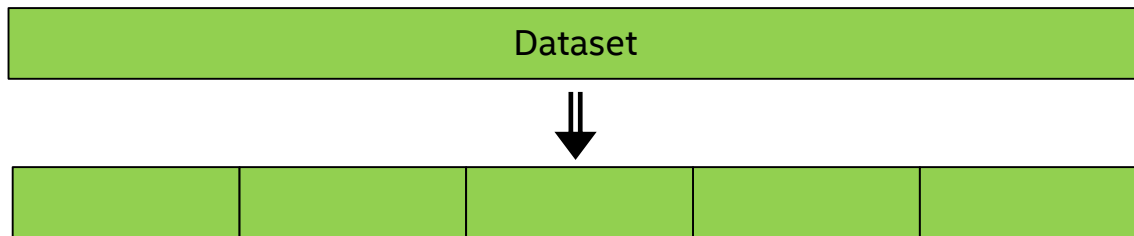
- Regularization is a common name for techniques that are intended to reduce generalization (test) error rather than simply reduce training loss.
- Example:

$$h^* = \max_{h \in \mathcal{H}} Q(h, \mathcal{D}) + \lambda \sum_{i=1}^n a_i^2 .$$

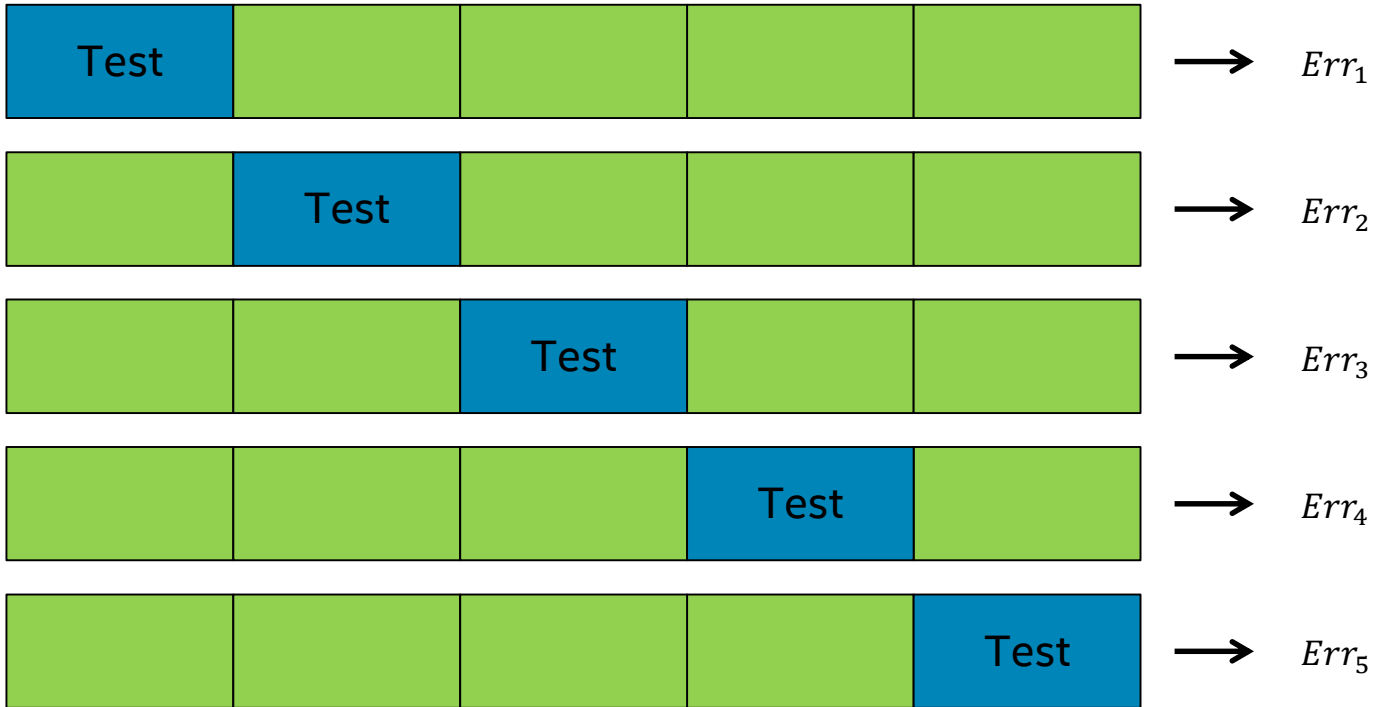


# Cross-validation

- The more data points are there in test dataset the more reliable model quality estimate is.
- The more data points are there in training dataset the more chances to fit a better model.
- There always is a tradeoff between sizes of train and test datasets.
- One of the popular ways to get better unbiased estimate of model quality is to use cross-validation: split dataset into train/test several times and average quality measures over this folds.



# Cross-validation



$$Err_{CV} = \frac{1}{k} \sum_{i=1}^k Err_i$$

# No free lunch

- Is there a single best machine learning algorithm?
- No. No machine learning algorithm is universally better than any other.
- All machine learning algorithms have the same error rate on a test dataset if it is averaged over all possible target functions.
  
- The goal is to understand what kinds of distributions are relevant to real-world tasks and what kinds of machine learning algorithms perform well under these assumptions.

# Beyond supervised learning

- Supervised learning algorithms require labeled data, which is expensive.
- For some tasks there is no single and well defined target, or it's not available at all.

# Reinforcement learning

- Situations when we know what we want, but don't know how to get it.
- E.g. sequential decision making.
  - It's hard to say what actions should some agent perform to reach the goal.
  - Every action has consequences that affect later actions.
  - It's hard to describe all successful sequences of actions.
- Direct supervision is not available. But allowing the agent to interaction with an environment (perform actions) we can get a feedback – receive a numerical reward signal (experience consequences of actions), which can guide learning.
- The goal is to estimate and maximize the long-term cumulative reward by finding the best policy for agents' behavior.

# Unsupervised learning

- We just want to find some useful / interesting associations in our data.
- No target feature, no reward.
- Often an ill-defined problem.
- Examples:
  - Clustering: group similar data points together and put dissimilar data points in different groups.
  - Dimensionality reduction: reduce feature space dimension without a loss of some useful dependencies / regularities.

# Applications

## Marketing and business.

- Churn prediction.  
Based on a user behavior predict if he/she wants to quit.
- Profit prediction.  
Given some parameters of the new product and state of the market, predict what will be the revenue from its sales.
- Sentiment analysis.  
Based on text of customer review determine if it is positive or negative.
- Market / customer segmentation.  
Based on customer profile and behavior find similar customer groups.
- Market basket analysis.  
Based on the history of customer purchases determine which products are often bought together.

# Applications

## Security.

- Fraud / anomaly detection.  
Based on some features of sample (bill, transaction, etc.) determine whether it is a valid sample or not.
- Identification / verification.  
Identify a user/customer based on his behavior.



# Applications

## Medicine.

- **Diagnosis.**  
Based on patients' test results determine the disease he/she has (if any).
- **Treatment planning.**  
Based on current patient state develop an optimal treatment plan.
- **Personalized medicine.**  
Based on medical test results of a particular patient develop an optimal personalized treatment plan.
- **Drug discovery.**  
Based on known effects of different drugs application and possibly tests of new (generated at algorithms' runtime) drugs, generate novel, more effective drugs.

# Applications

## Efficient search.

- Information retrieval.  
Based on the query and personalized search and click results rank search output.
- Natural language / image understanding.  
Get better understanding what is the semantic meaning of a text/image query and/or documents to do better matching.
- Recommendation systems (collaborative filtering).  
Based on personal user profile, its similarity to other users and their profiles recommend a product that user will like.

# Machine learning is everywhere

