UNIVERSITÉ LIBRE DE BRUXELLES

Everything you always wanted to know about ML (but were afraid to ask)

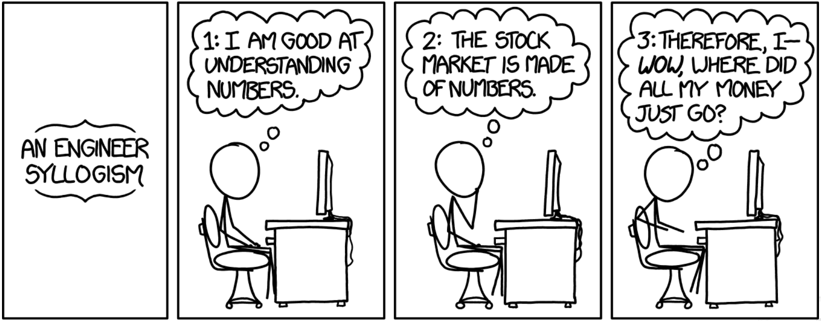
BSSM 2019

Jacopo De Stefani - jdestefa@ulb.ac.be

Université Libre de Bruxelles

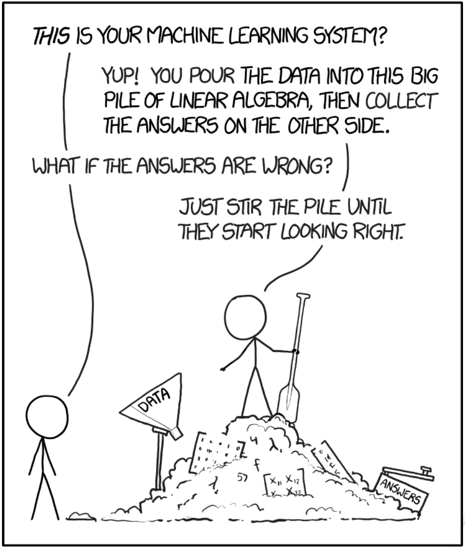
Tuesday 3rd September, 2019

Disclaimer



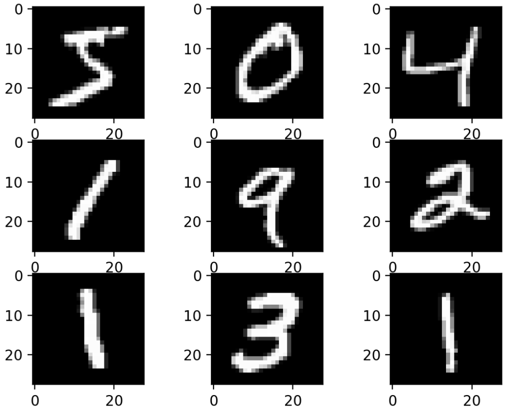
Relevant XCKD: 1570

Machine Learning?



Relevant XCKD: 1838

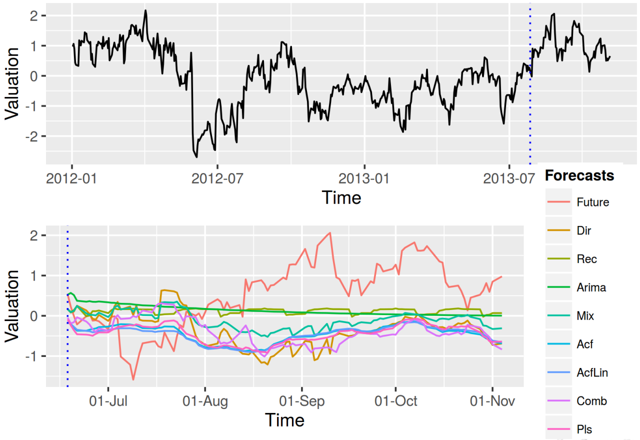
Some examples - Image classification



Some examples - Fraud Detection

Some examples - Villo availability prediction

Some examples - Time Series Analysis



What are the common points?

- Structured data
  - Often not the case in real-life problem
  - Preprocessing

ULB

UNIVERSITÉ LIBRE DE BRUXELLES

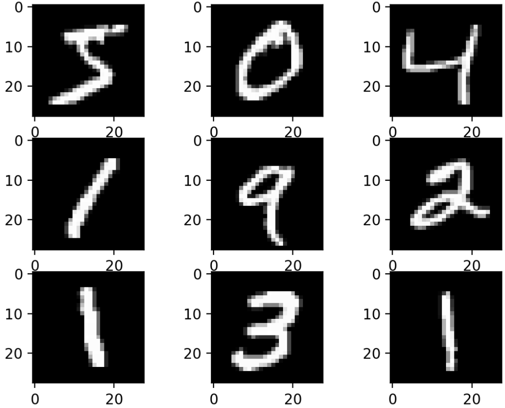
What are the common points?

- Structured data
  - Often not the case in real-life problem
  - Preprocessing
- Single output variable
  - Fraud Detection,Image classification: Discrete value $\Rightarrow$ **Classification**
  - Villo,TS: Continuous value $\Rightarrow$ **Regression**

What are the common points?

- Structured data
  - Often not the case in real-life problem
  - Preprocessing
- Single output variable
  - Fraud Detection,Image classification: Discrete value $\Rightarrow$ **Classification**
  - Villo,TS: Continuous value $\Rightarrow$ **Regression**
- Unknown Input/Output mapping
  - No available model
  - Data-driven

Some examples - Image classification



$$h_{IC} : \mathbf{X} \in \mathbb{R}^{32\times 32} \mapsto y \in \{0, \cdots, 9\}$$

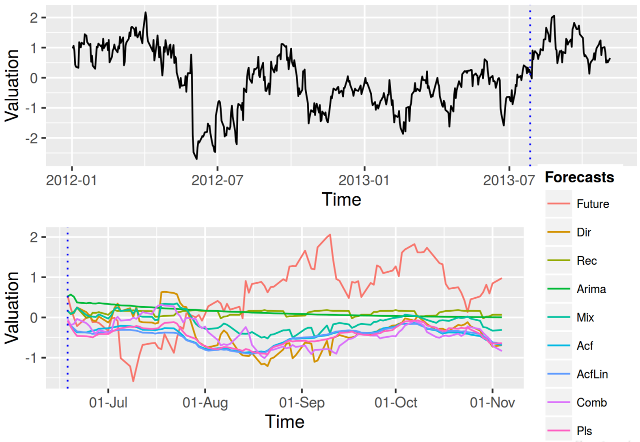
Some examples - Fraud Detection

$$h_{FD} :< ID, Country, Amount, Amount_{avg}, .. > \mapsto y \in \{0,1\}$$

Some examples - Regression

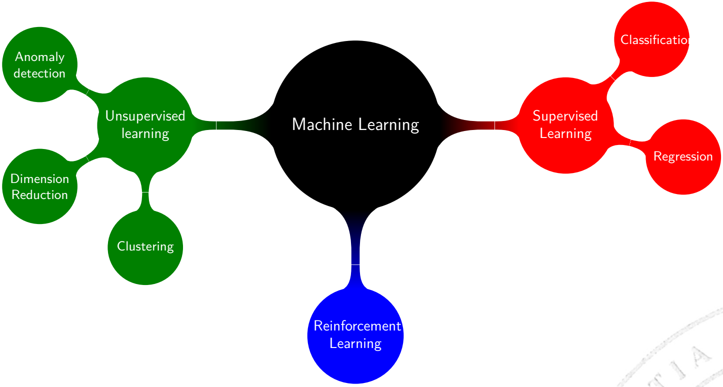
$$h_R :< Lat, Long, Weather, Day, \cdots > \mapsto y \in \mathbb{R}^+$$

Some examples - Time Series Analysis



$$h_{TS} : \mathbf{X} = [y_{t-d}, \cdots, y_{t-1}] \in \mathbb{R}^d \mapsto y = y_t \in \mathbb{R}$$

Machine Learning Overview



Machine Learning

Definition

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

T. Mitchell,1997

Supervised Learning Problem - ?

- **Input:** A vector of n random variables $\mathbf{x} \in \mathcal{X} \subset \mathbb{R}^N$, distributed according to an unknown probabilistic distribution $F_x(\cdot)$.
- **Target operator** $f : \mathbf{x} \mapsto y \in \mathcal{Y}$ according to an unknown probability conditional distribution $F_y(y|x = \mathbf{x})$.
- **Training set**: $D_n = \{< \mathbf{x_1}, y_1 >, \cdots, < \mathbf{x_1}, y_1 >\}$, drawn according to the joint input/output density $F_{xy}(\mathbf{x}, y)$.

Learning machine - ?

- Learning machine
  - **Hypothesis/Model:** $h(\cdot,\cdot) :< \mathbf{x}, \vartheta > \mapsto h(\mathbf{x}, \vartheta) \in \mathcal{Y}$
  - **Class of hypotheses:** $h(\cdot, \vartheta), \vartheta \in \Theta$
  - **Loss function:** $L(\cdot,\cdot) :< \mathbf{x}, y > \mapsto L(\mathbf{x}, y) \in \mathbb{R}$
  - **Learning algorithm:** $\mathcal{L} :< \Theta, D_n > \mapsto h(\cdot, \vartheta_n)$

Empirical risk minimization - ?

$$\vartheta_n = \vartheta(D_n) = \arg\min_{\vartheta \in \Theta} R_{emp}(\vartheta) \tag{1}$$

$$R_{emp}(\vartheta) = \frac{1}{n} \sum_{i=1}^{n} L(y_i, h(\mathbf{x_i}, \vartheta)) \tag{2}$$

$$\nabla J(\vartheta) = 0 \tag{3}$$

Machine Learning Process - ?

Preliminary phase

1. Problem formulation
2. Experimental design
3. Preprocessing step
   - Missing data
   - Feature selection
   - Outlier removal

Learning phase

1. Parametric identification
2. Model selection

Parametric identification - ?

The choice of an optimization algorithm depends on the form of:

$$J(\vartheta) = R_{emp}(\vartheta) = \frac{1}{n}\sum_{i=1}^{n} L(y_i, h(\mathbf{x_i}, \vartheta)) \qquad (4)$$

which in turns depends on:

- **Model:** $h(\cdot, \vartheta), \vartheta \in \Theta$
- **Loss function:** $L(y, h(\cdot, \mathbf{x}))$

Analytic solution - ?

For some specific cases (e.g. Linear regression)

$$h(\mathbf{x}, \vartheta) = \vartheta \mathbf{x} \tag{5}$$

$$L(y_i, h(\mathbf{x_i}, \vartheta)) = (y_i - h(\mathbf{x_i}, \vartheta))^2 \tag{6}$$

It exists a closed form solution:

$$\vartheta_n = (X^T X)^{-1} XY \tag{7}$$

Iterative search - ?

In order to minimize:

$$J(\vartheta) = R_{emp}(\vartheta) = \frac{1}{n}\sum_{i=1}^{n} L(y_i, h(\mathbf{x_i}, \vartheta)) \tag{8}$$

One need to solve:

$$\nabla J(\vartheta) = 0 \tag{9}$$

Several methods provides incremental solutions in the form:

$$\vartheta^{(\tau+1)} = \vartheta^{(\tau)} + \Delta\vartheta^{(\tau)} \tag{10}$$

Why $\nabla J(\vartheta) = 0$?

The first order derivative gives indication on the behaviour of the function:

$$\frac{\partial f}{\partial x} > 0 \Rightarrow \nearrow$$
$$\frac{\partial f}{\partial x} < 0 \Rightarrow \searrow$$

Critical points $x*$ are candidates for local minima/maxima

$$\frac{\partial f}{\partial x^*} = 0 \qquad (11)$$

1-**Dimensional**

The first order derivative gives indication on the behaviour of the function, **along each direction**:

$$\nabla f = \left[\frac{\partial f}{\partial x}, \frac{\partial f}{\partial y}, \frac{\partial f}{\partial z}, \cdots\right]$$
$$\nabla_i > 0 \Rightarrow \nearrow_i$$
$$\nabla_i < 0 \Rightarrow \searrow_i$$

Critical points $x*$ are candidates for local minima/maxima

$$\nabla f = \mathbf{0} \qquad (12)$$

n-**dimensional**

Gradient descent - ?

$$\vartheta^{(\tau+1)} = \vartheta^{(\tau)} + \Delta\vartheta^{(\tau)} \tag{13}$$

$$\Delta\vartheta^{(\tau)} = -\eta \nabla J(\vartheta^{(\tau)}) \tag{14}$$

figure/GradientDescent.jpg

Batch gradient descent

1: Initialize ϑ et η
2: **while** ! converged **do**
3: **for** $i \in \{1, \cdots, n_b\}$ **do**
4: $\vartheta^{(\tau+1)} = \vartheta^{(\tau)} - \eta \nabla J_i(\vartheta^{(\tau)})$
5: **end for**
6: **end while**

Standard gradient descent

1: Initialize ϑ et η
2: **while** ! converged **do**
3: **for** $i \in \{1, \cdots, n_b\}$ **do**
4: $\vartheta^{(\tau+1)} = \vartheta^{(\tau)} - \eta \frac{1}{b_i} \sum_{i=1}^{b_i} \nabla J_i(\vartheta^{(\tau)})$
5: **end for**
6: **end while**

Batch gradient descent

Stochastic gradient descent

```
Initialize ϑ et η
while ! converged do
    for i ∈ {1, ··· , n_b} do
        ϑ^(τ+1) = ϑ^(τ) − η (1/b_i) Σ_{i=1}^{b_i} ∇J_i(ϑ^(τ))
    end for
end while
```

$$\vartheta^{(\tau+1)} = \vartheta^{(\tau)} - \eta \frac{1}{b_i} \sum_{i=1}^{b_i} \nabla J_i(\vartheta^{(\tau)})$$

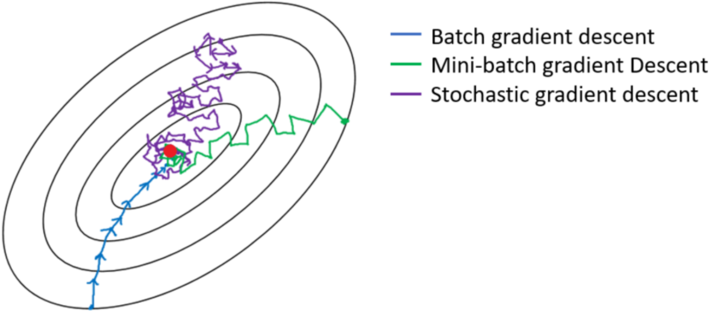
Batch gradient descent

```
Initialize ϑ et η
while ! converged do
    Shuffle training set
    for i ∈ {1, ··· , n_b} do
        ϑ^(τ+1) = ϑ^(τ) − η∇J_i(ϑ^(τ))
    end for
end while
```

$$\vartheta^{(\tau+1)} = \vartheta^{(\tau)} - \eta \nabla J_i(\vartheta^{(\tau)})$$

Stochastic gradient descent

Methods comparison



And many more...

- **Iterative methods**
  - Newton method
  - Levenberg-Marquardt
- **Stochastic gradient descent**
  - Momentum
  - AdaGrad
  - RMSProp
  - Adam
- **Meta-heuristics**
  - Random search
  - Genetic algorithm
  - Simulated annealing

Model selection - ?

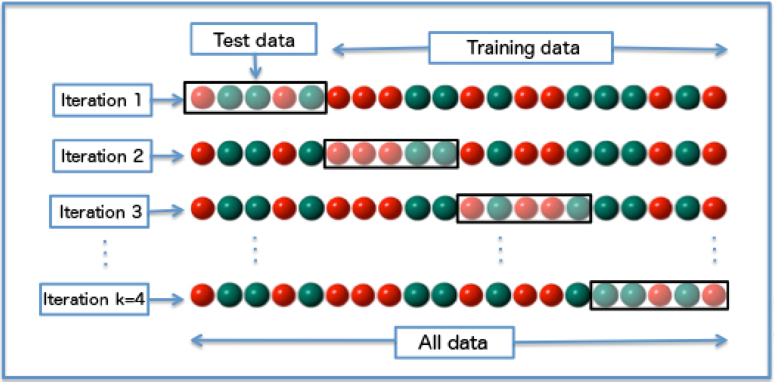
The selection of a model is usually perfomed by looking at its performance:

$$R_{ts}(\vartheta) = \frac{1}{n_{ts}} \sum_{i=1}^{n_{ts}} L(y_i, h(\mathbf{x_i}, \vartheta)) \tag{15}$$

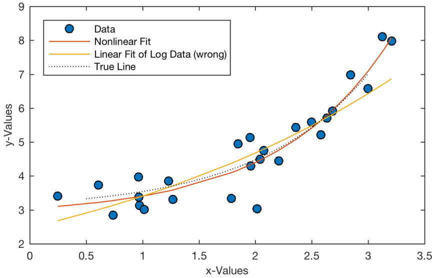
on unseen data:

$$D_{ts} = \{< \mathbf{x_{n+1}}, \mathbf{y_{n+1}} >, \cdots, < \mathbf{x_{n+n_{ts}}}, y_{n+n_{ts}} >\} \tag{16}$$

Cross Validation



Models - Linear model



$$y = \mathbf{w}\mathbf{x} = \sum_{i=1}^{n_w} w_i x_i$$

- **Parameters:** $\vartheta = \mathbf{w}$
- **Parametric identification:** Closed form / Gradient descent

Models - Logistic regression

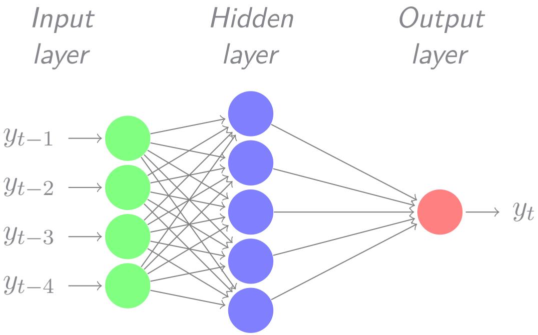
figure/LogisticRegression.png

- $\log_b \left(\frac{p}{1-p} \right) = \mathbf{wx}$
- $p = P(Y = 1) = \frac{1}{1+b^{-(w_0+w_1x_1+w_2x_2)}}$

- **Parameters:** $\vartheta = \mathbf{w}$
- **Parametric identification:** Maximum Likelihood Estimation + Gradient descent

ULB

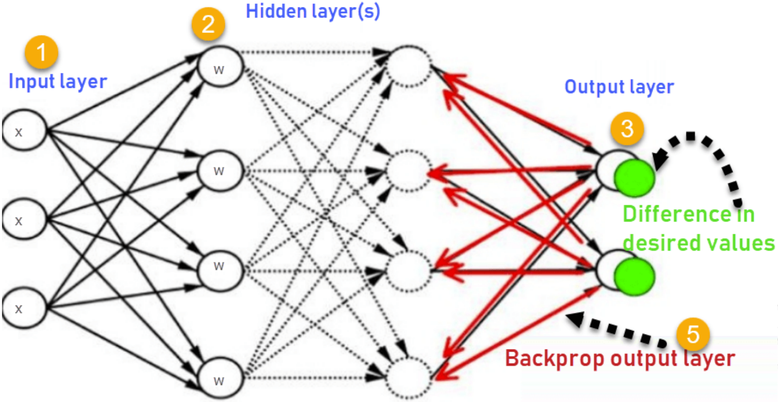
Models - ANN



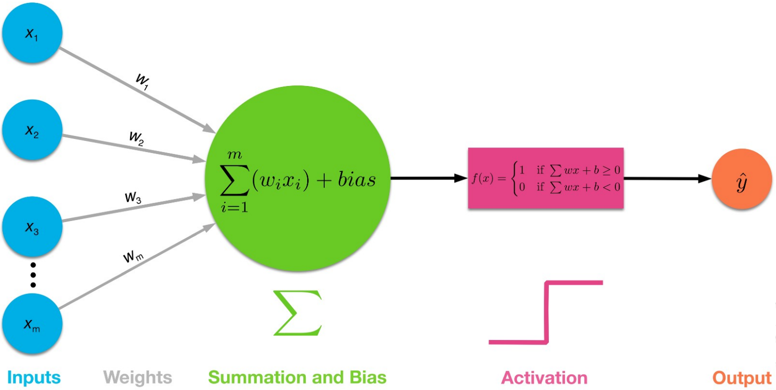
$$y = f\left(b_o + \sum_{j=1}^{|H|} w_{jo} \cdot g\left(\sum_{i=1}^{|I|} w_{ij}x_i + b_j\right)\right)$$

- **Parameters:** $\vartheta = [\mathbf{w_h}, \mathbf{w_o}]$
- **Parametric identification:** Gradient descent + Backpropagation

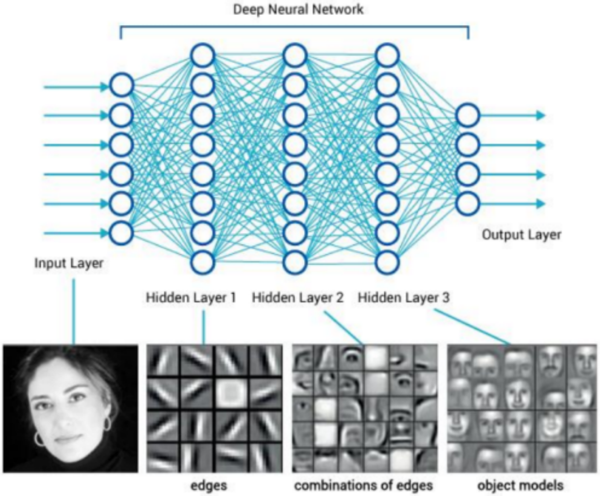
Backpropagation



Perceptron

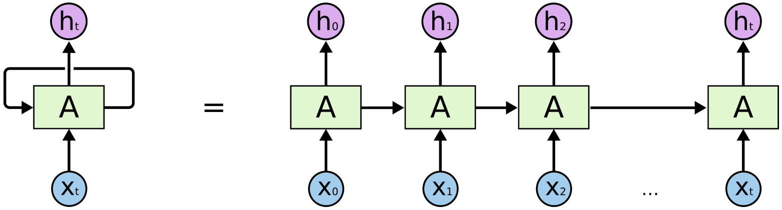

x_1
x_2
x_3
x_m
w_1
w_2
w_3
w_m
$\sum_{i=1}^{m}(w_i x_i) + bias$
$f(x) = \begin{cases} 1 & \text{if } \sum wx + b \geq 0 \\ 0 & \text{if } \sum wx + b < 0 \end{cases}$
$\hat{y}$
$\sum$
Inputs
Weights
Summation and Bias
Activation
Output

Deep Learning - Intuition

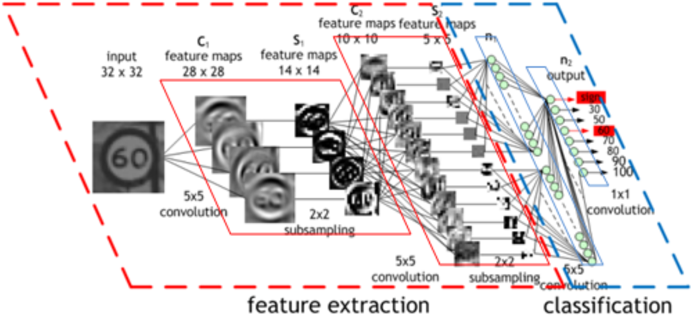


Demo : `http://playground.tensorflow.org/`

Deep Learning - RNN - Intuition



Deep Learning - CNN - Intuition



Demo: `https://cs.stanford.edu/people/karpathy/convnetjs/demo/mnist.html`

And many more...

- **Non-parametric methods**
  - Decision Trees
  - K-nearest neighbors
  - Radial Basis Functions
- **Network based**
  - CNN
  - Restricted Boltzmann Machines
- **Ensemble techniques**
  - Random Forests
  - Gradient Boosting

Wrap-up

- ML is not magic, but heavily relying on:
  - Linear algebra
  - Statistics
- Data is as important (if not more) than the model
- Data preprocessing can be as time consuming as parameter estimation / model selection
- Simpler is (often) better

figure/MLData.jpg

Thank you for your attention! Any questions/comments?

img/Questions.jpg

Bibliography I