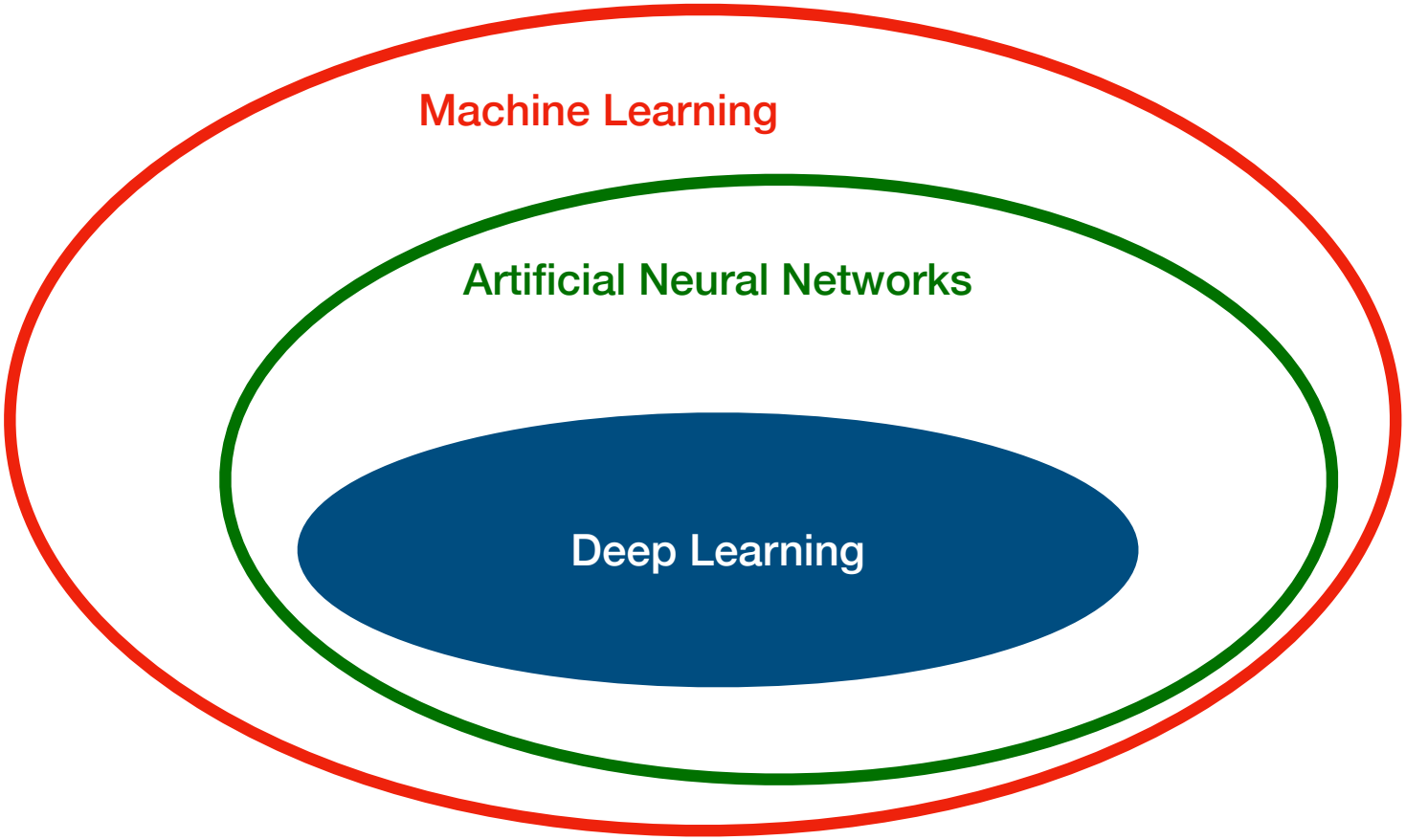
# Machine Learning

## A Practical Introduction with a bit of theory!

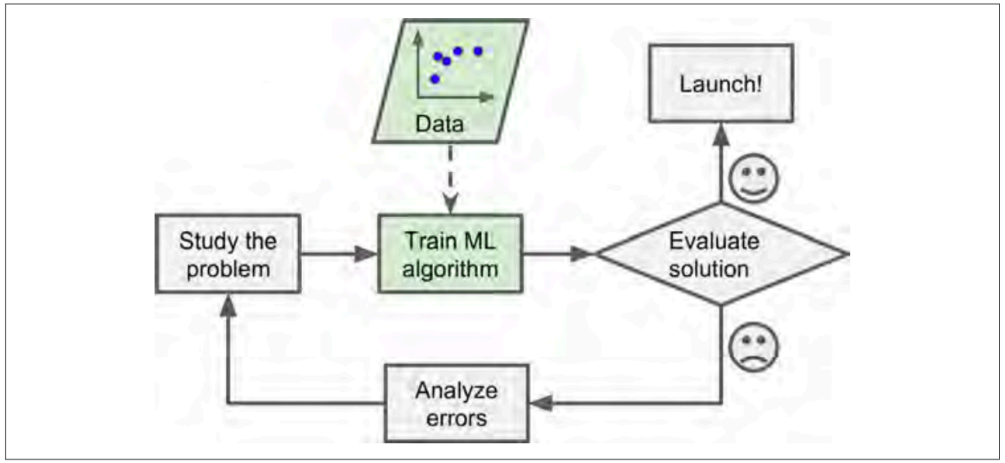
Pascal Germain, Hamid Mirisaee, Vera Shalaeva

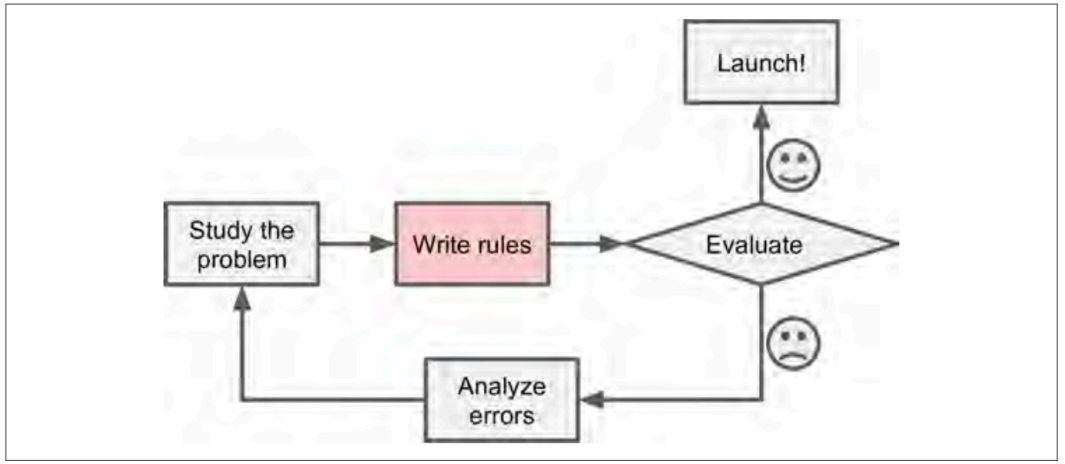
AI: Artificial Intelligence



# What is ML?

- Science of programming computers so that they can learn from data.
  - Spam filter: flag each email as “spam” or “not spam”.
    - Pattern-based (free, amazing, etc): many many rules —> complex.
    - ML-based: automatically learn which phrases are likely to be spam.



# Algorithms of ML

- Linear predictors
    - Linear regression
    - Support vector machines (SVM)
- Ensemble methods
    - AdaBoost
    - XGboost
    - Random Forest
- Neural Nets

# Types of ML systems (1)

1. Supervised: data points/instances/examples + labels/targets.
    - Classification (spam filter).
    - Predict target values (price of a car given its features, called predictors).
    - KNN, LR, Logistic Regression, SVM, DT, RF, NN.
2. Unsupervised: no labels.
    - Clustering: K-means, Hierarchical clustering, EM, etc.
    - Dimensionality Reduction: PCA, SVD, t-SNE, etc.
    - Association rule learning: Ecalt.
3. Semisupervised
4. Reinforcement Learning

# Types of ML systems (1)

1. Supervised: data points/instances/examples + labels/targets.
2. Unsupervised: no labels.
3. Semisupervised: partially labeled data
    - Usually mixture of supervised and supervised methods.
4. Reinforcement Learning: the system (agent) observes the environment, takes an action, gets a reward in return, and then learns a strategy (policy) to get the most reward over time.

# Types of ML systems (2)

1. Batch learning: gets the entire data, learns the model, then it is put into the production —> offline learning.
    - New data —> learning everything again.
    - Problem: what if you need to adapt rapidly and do not have time to retrain?
    - Learning rate: how fast we move towards the optimum point.
2. Online learning: the data is fed incrementally to the system (one-by-one or in form of small sets) —> efficient: once learned from data, just drop it.
    - Learning rate: how much we rely on the new data and forget the old one.

# Types of ML systems (3)

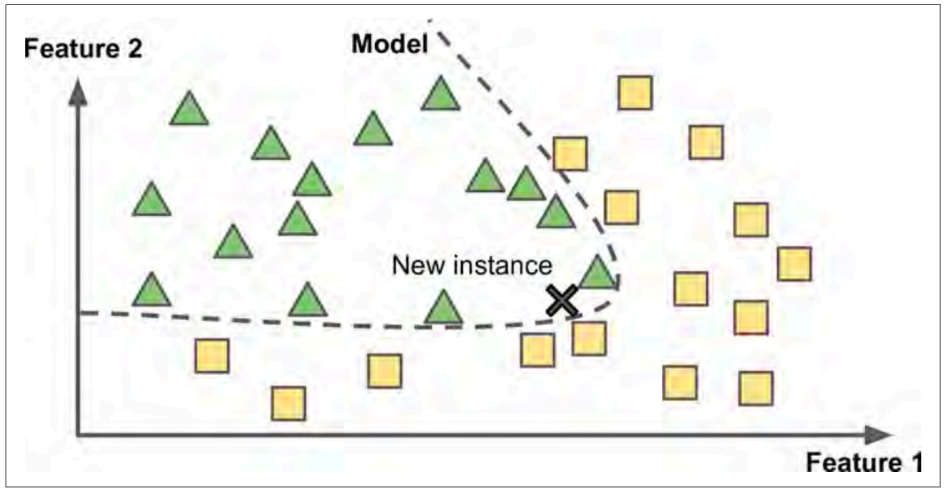
**Instance-based learning**

Tag new instances using their similarities to the know ones.

Spam filter: mark the emails similar to “already flagged as spam” as spam (using a similarity measure: #common words, etc).

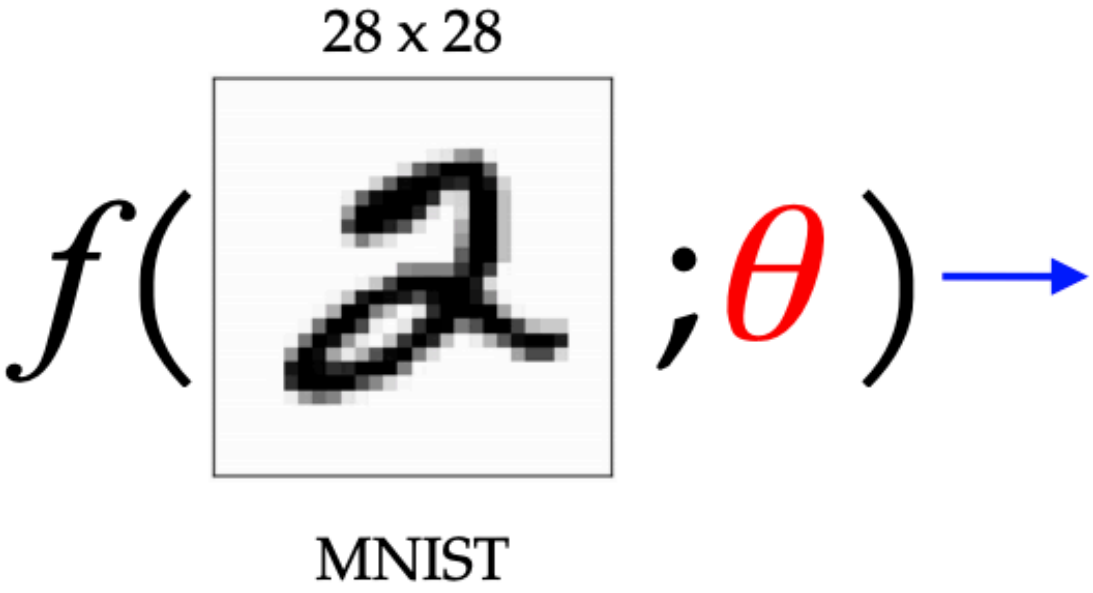
**Model-based learning**

Use the data to make a prediction model.



# Examples of ML applications

Character recognition




Vector of 10 scores, one per class

$\theta$ - parameters of the function

# Examples of ML applications

Image recognition

224 x 224 x 3

$f($  $;\theta)$ → Vector of 1000 scores, one per class

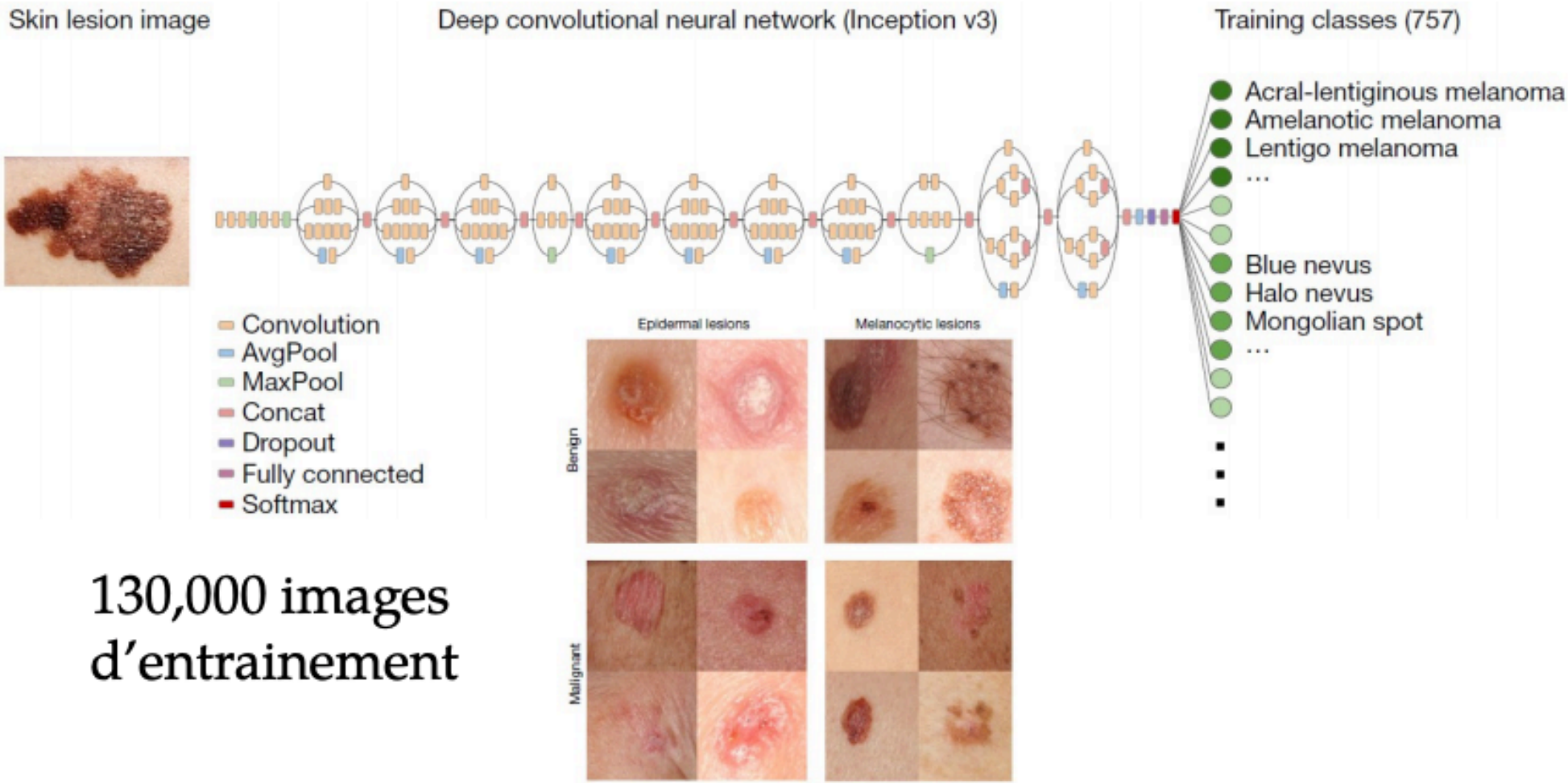
ImageNet

$\theta$ – parameters of the function

LETTER Nature, Février 2017
doi:10.1038/nature21056

# Dermatologist-level classification of skin cancer with deep neural networks

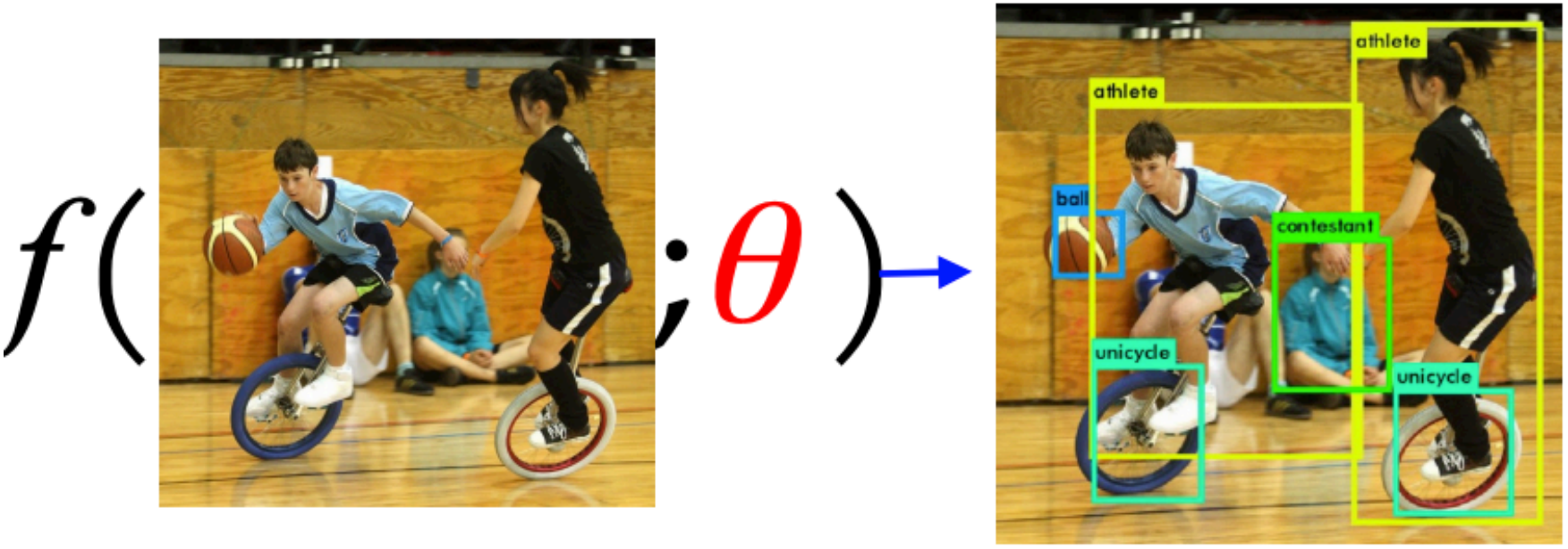
Andre Esteva[1]*, Brett Kuprel[1]*, Roberto A. Novoa[2,3], Justin Ko[2], Susan M. Swetter[2,4], Helen M. Blau[5] & Sebastian Thrun[6]




130,000 images d'entrainement

# Examples of ML applications

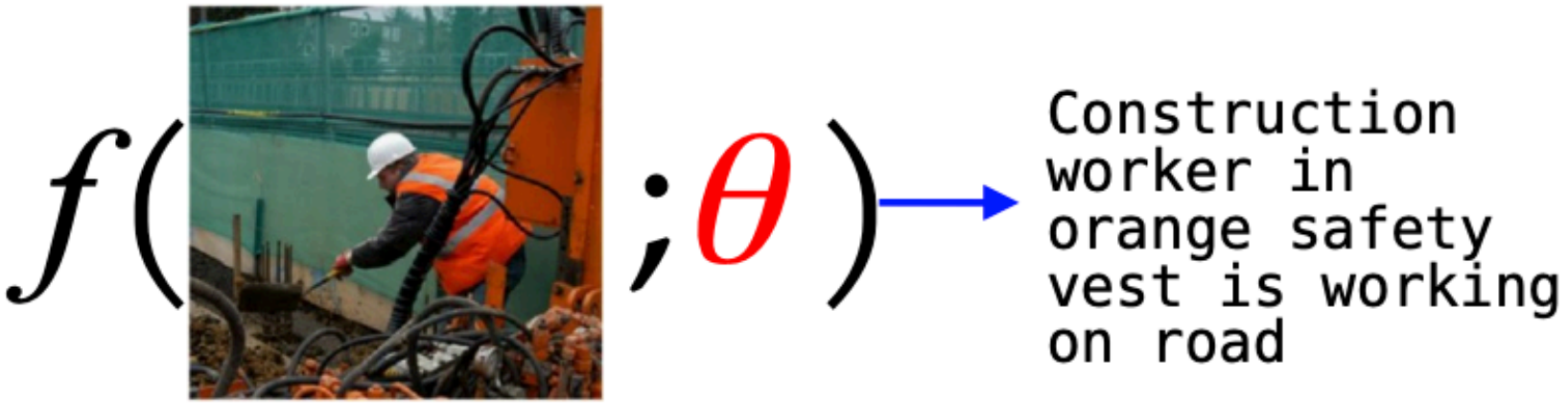
Object detection




Redmon and Farhadi, YOLO9000: Better, Faster, Stronger, CVPR 2017.

# Examples of ML applications

Image description




Karpathy and Fei-Fei, Deep Visual-Semantic Alignments for Generating Image Descriptions, CVPR 2015.

# Examples of ML applications

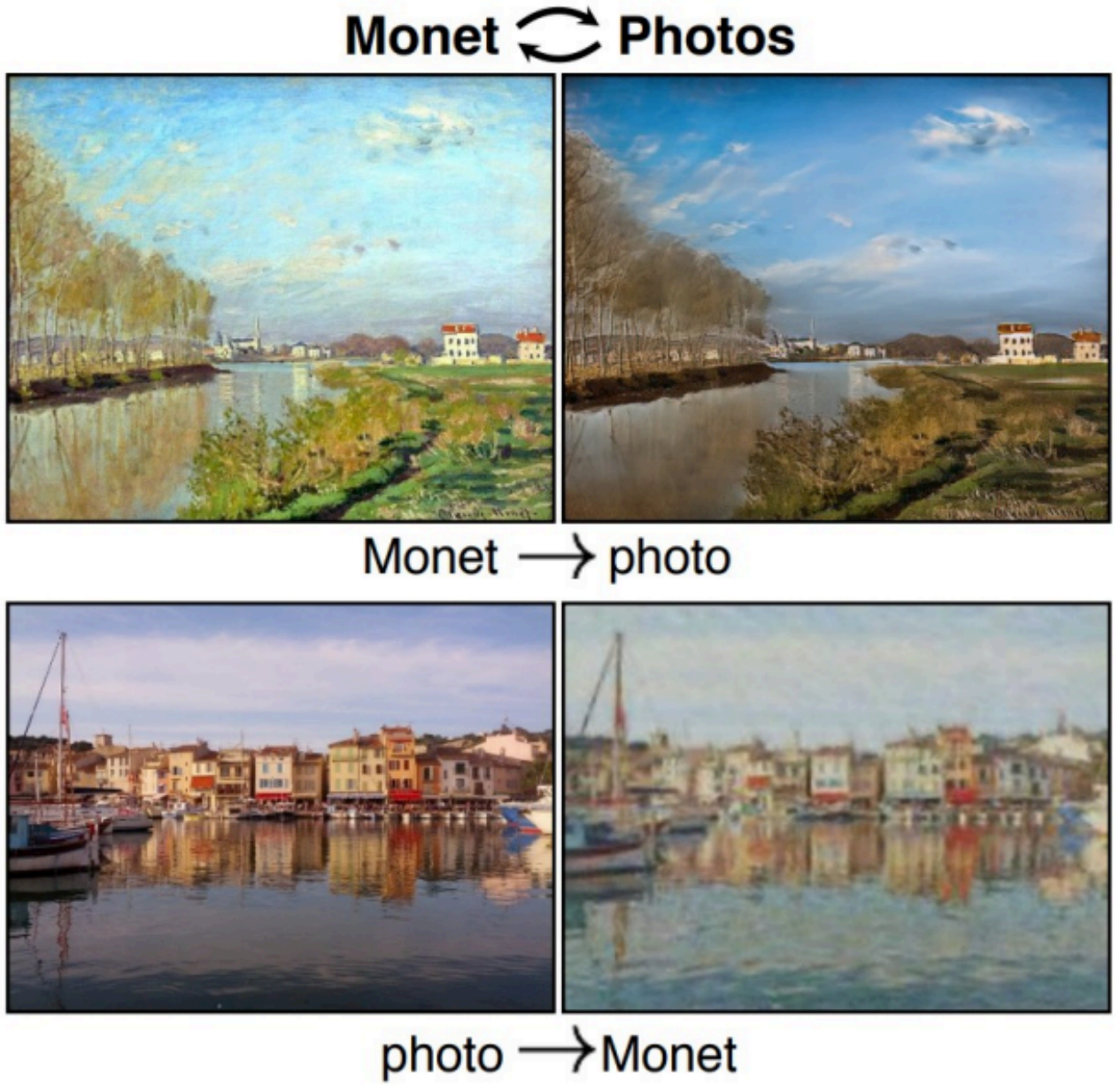
Voice recognition

# Examples of ML applications

Translation

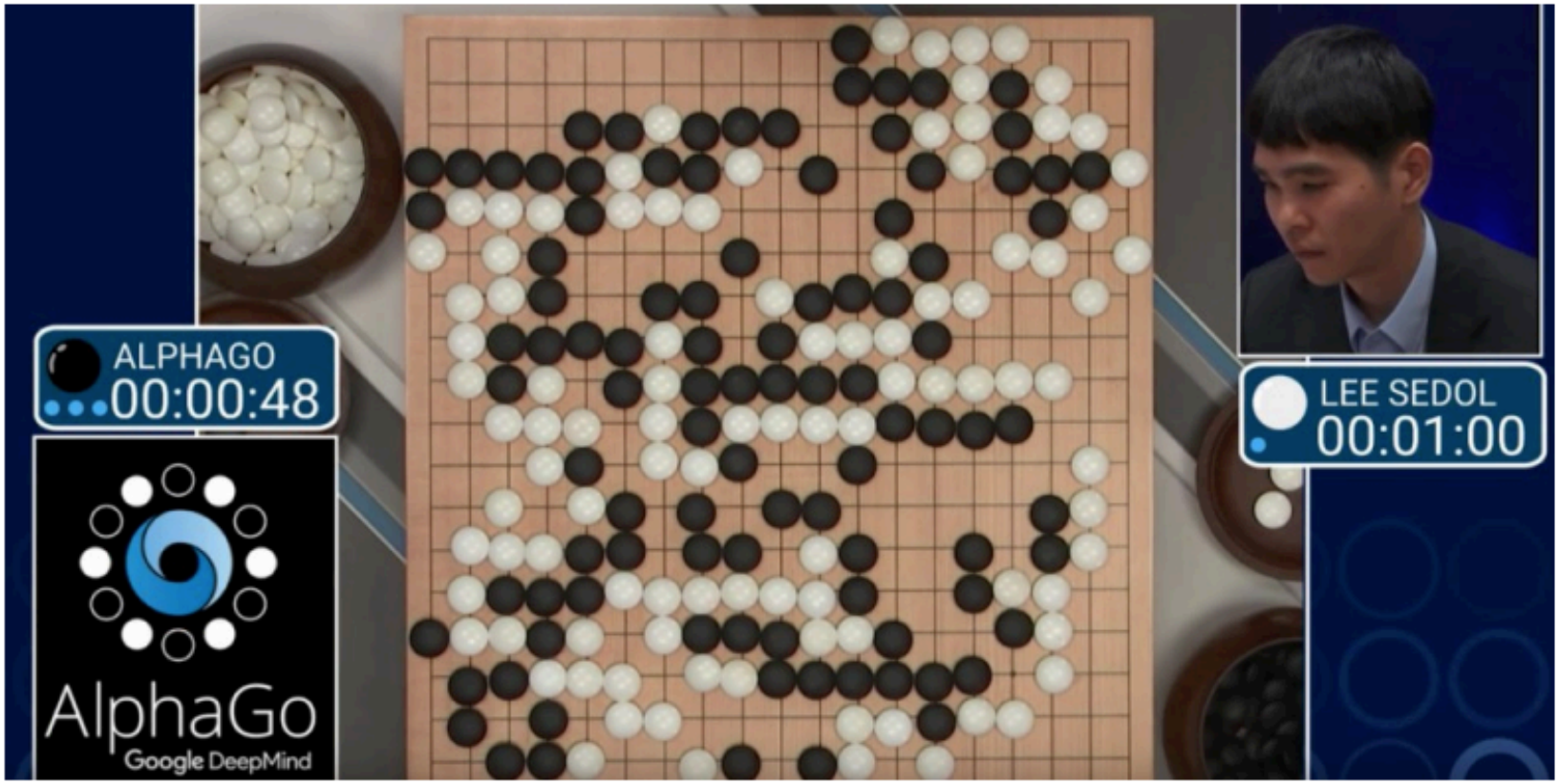
# Examples of ML applications

Style transfer



# Examples of ML applications

Games



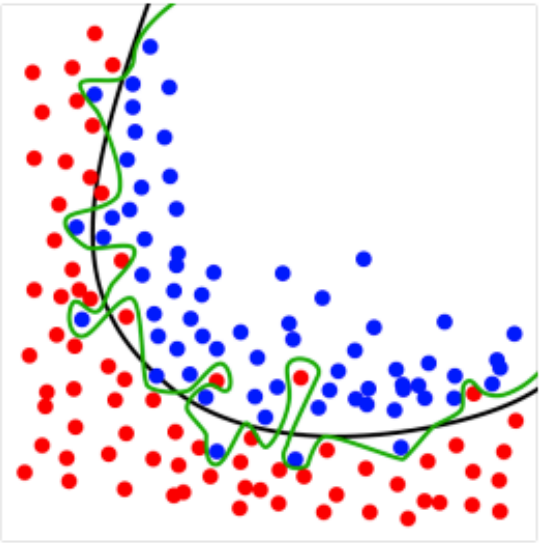
# Performance

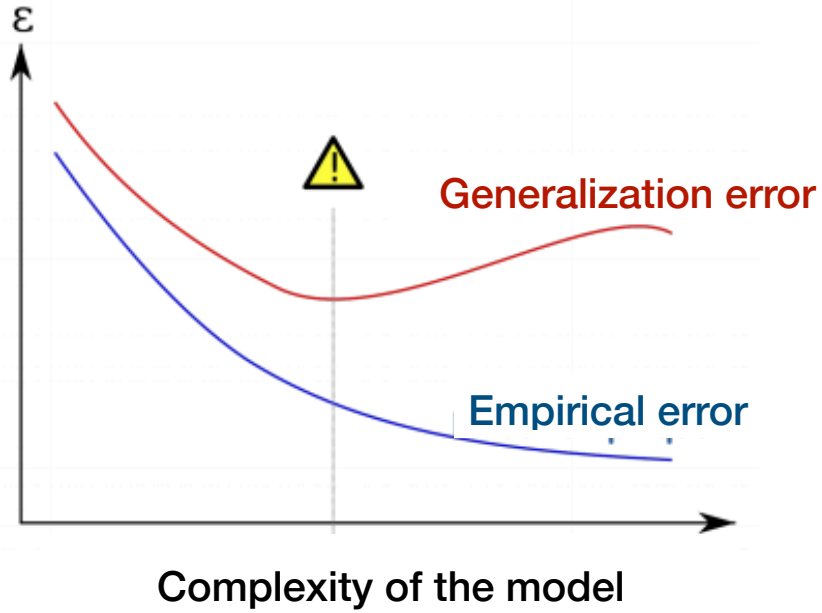
Loss functions:

- 0-1 loss
- Quadratic loss
- Hinge loss
- Logistic loss
- Cross entropy loss

# Challenges in ML

- Not enough training data.
- Not the right data (not representative, not good features, etc).
- High dimensionality of data, missing values.
- Underfitting: not enough training.
- Overfitting the data (the model does not generalize).





# Linear regression (LR)

Learning sample $S = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_n, y_n)\}, \mathbf{x}_i \in \mathbb{R}^d, y \in \mathbb{R}$

Linear predictor without a bias

$$f_{\mathbf{w}} = \mathbf{w} \cdot \mathbf{x}$$

Quadratic loss function

$$\min_{\mathbf{w}} \left[ \frac{1}{n} \sum_{i=1}^{n} (\mathbf{w} \cdot \mathbf{x}_i - y_i)^2 \right]$$

Find the minimum of the objective function

$$F(\mathbf{w}) = \frac{1}{n} \sum_{i=1}^{n} (\mathbf{w} \cdot \mathbf{x}_i - y_i)^2.$$

# Linear Regression (LR)

To find partial derivatives w.r.t. each parameter

$$\begin{aligned}\frac{\partial F(\mathbf{w})}{\partial w_k} &= \frac{1}{n}\sum_{i=1}^{n}\frac{\partial}{\partial w_k}(\mathbf{w}\cdot\mathbf{x}_i - y_i)^2 \\ &= \frac{1}{n}\sum_{i=1}^{n} 2\,(\mathbf{w}\cdot\mathbf{x}_i - y_i)\frac{\partial}{\partial w_k}(\mathbf{w}\cdot\mathbf{x}_i - y_i) \\ &= \frac{1}{n}\sum_{i=1}^{n} 2\,(\mathbf{w}\cdot\mathbf{x}_i - y_i)\frac{\partial}{\partial w_k}(w_k\, x_{i,k}) \\ &= \frac{1}{n}\sum_{i=1}^{n} 2\,(\mathbf{w}\cdot\mathbf{x}_i - y_i)\, x_{i,k}\,.\end{aligned}$$

Gradient

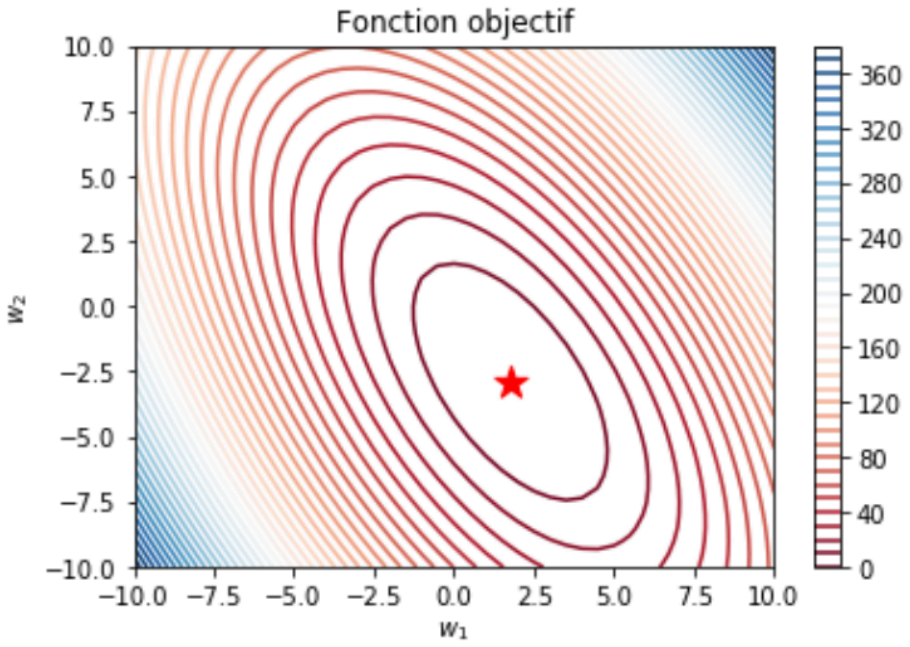
$$\nabla F(\mathbf{w}) = \begin{bmatrix}\frac{\partial}{\partial w_1} \\ \vdots \\ \frac{\partial}{\partial w_d}\end{bmatrix} = \frac{2}{n}\sum_{i=1}^{n}(\mathbf{w}\cdot\mathbf{x}_i - y_i)\,\mathbf{x}_i\,.$$

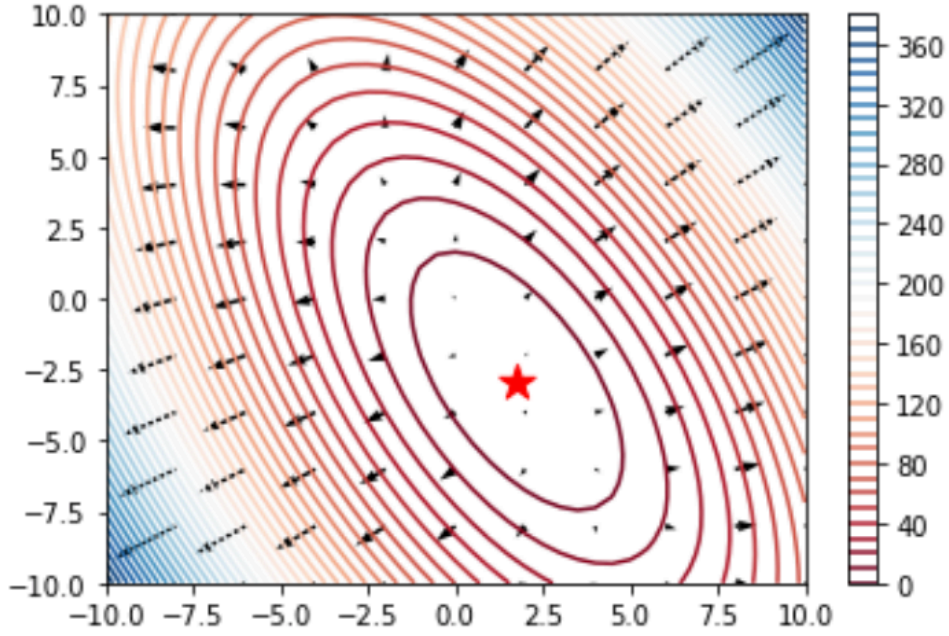
The minimum of a convex function is achieved when $\nabla F(\mathbf{w}) = 0.$

# Gradient descent

Example

$$\mathbf{X} = \begin{bmatrix} 1 & 1 \\ 0 & -1 \\ 2 & \frac{1}{2} \end{bmatrix}, \quad \mathbf{y} = \begin{bmatrix} -1 \\ 3 \\ 2 \end{bmatrix}$$



# Gradient descent

Algorithm

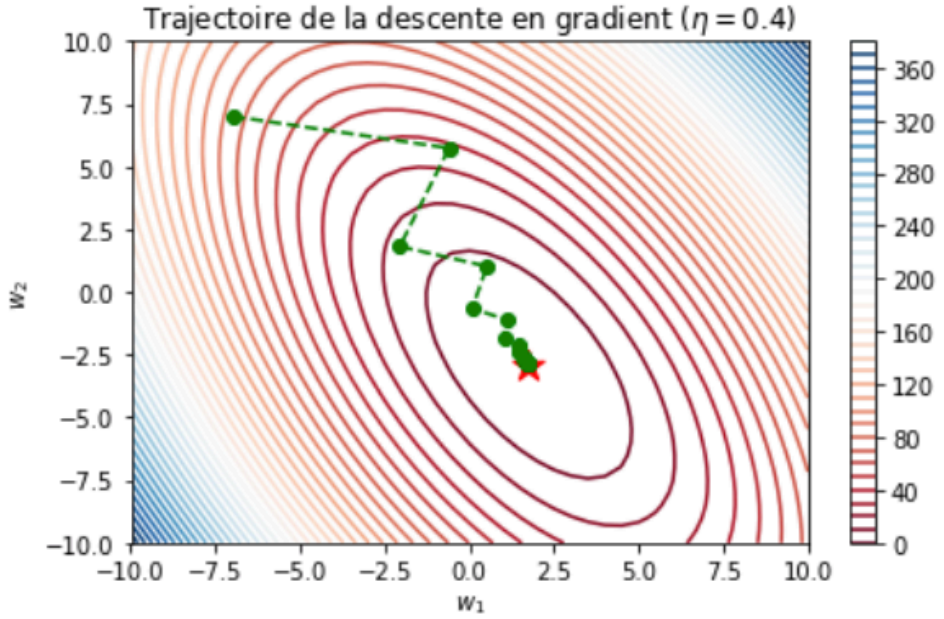
Gradient step $\eta$, number of iterations $T$

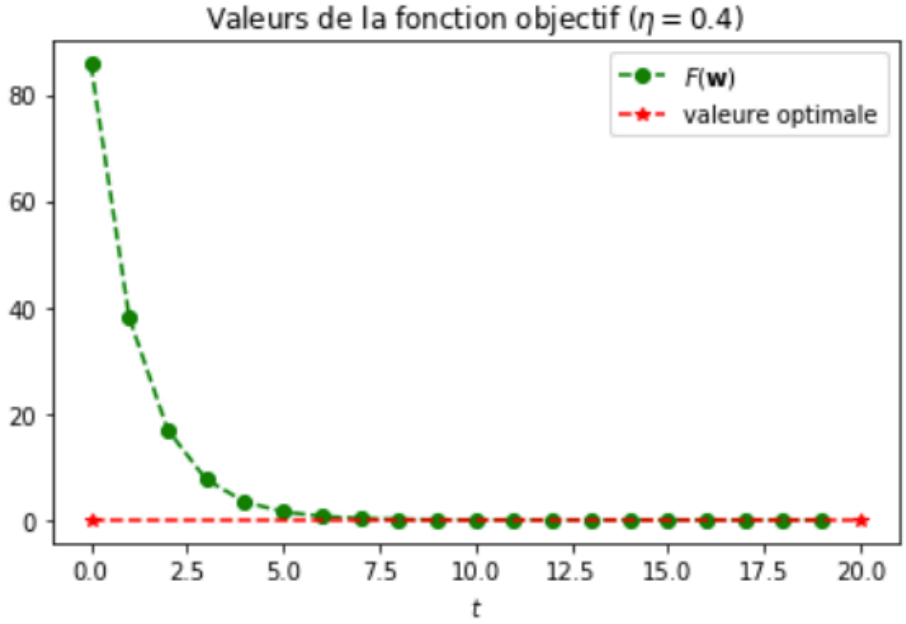
- Initialize $\mathbf{w}_0 \in \mathbb{R}^d$ randomly
- For $t$ from $1$ to $T$
  - $\mathbf{g}_t = \nabla F(\mathbf{w}_{t-1})$
  - $\mathbf{w}_t = \mathbf{w}_{t-1} - \eta \mathbf{g}_t$
- Return $\mathbf{w}_T$

# Gradient descent

Example

$\eta = 0.4$ $T = 20$

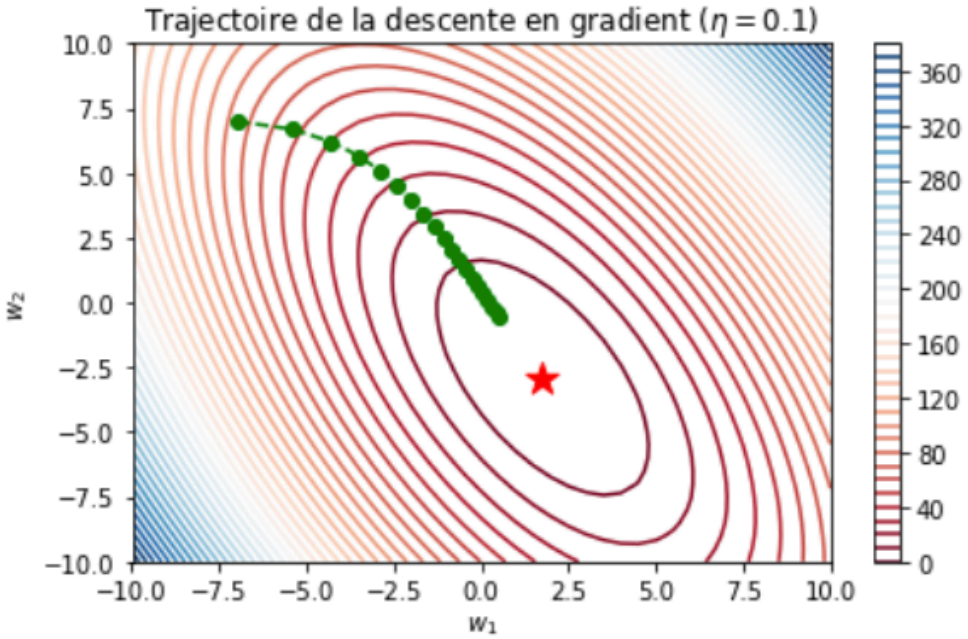


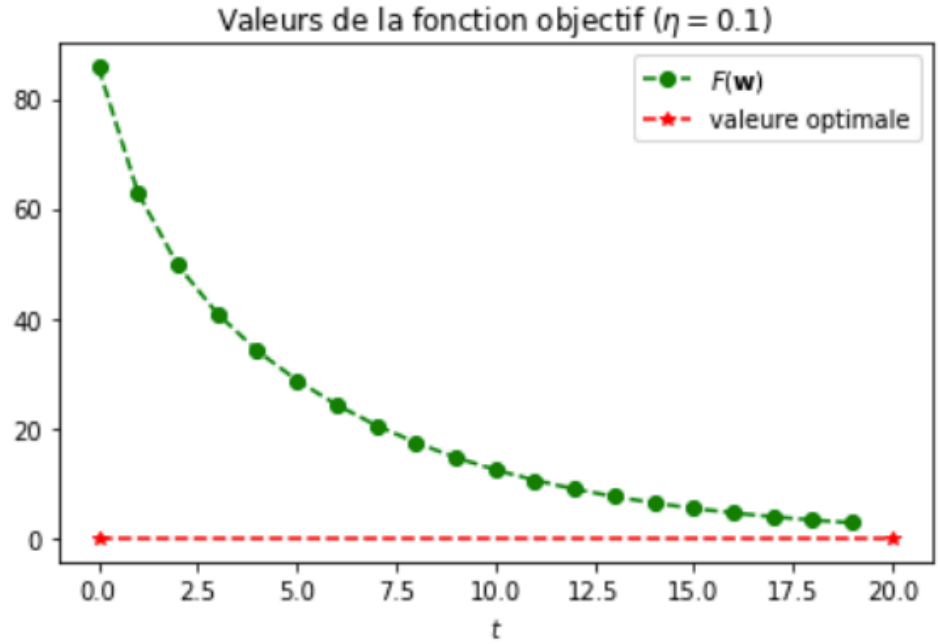



# Gradient descent

Example

$\eta = 0.1$ $T = 20$

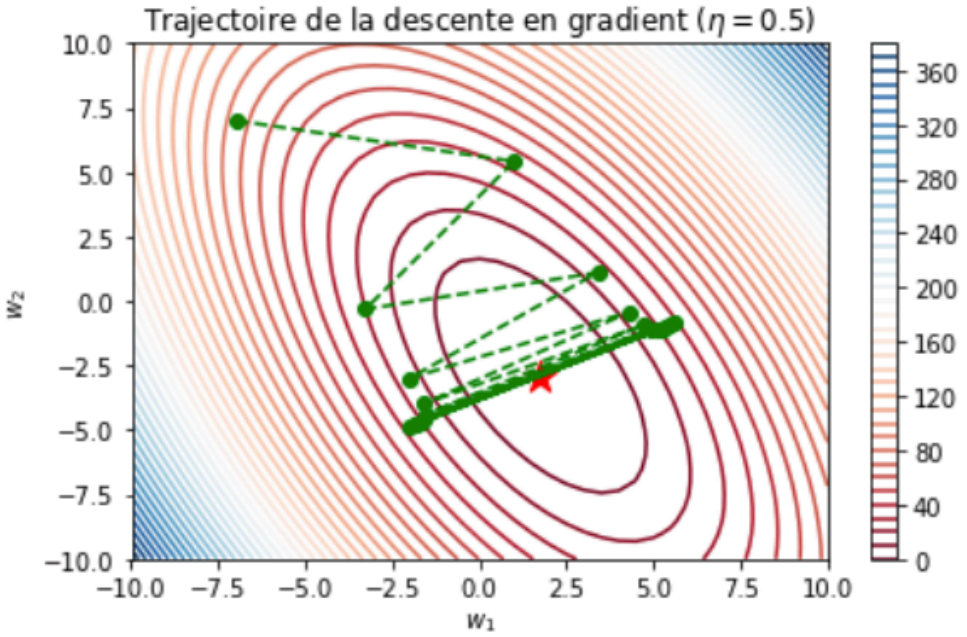


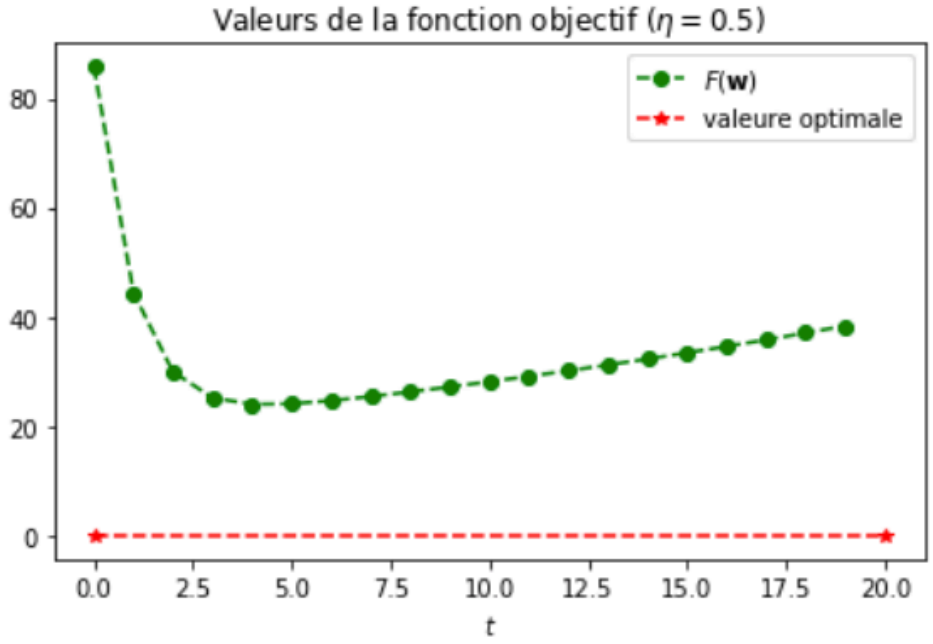



# Gradient descent

Example

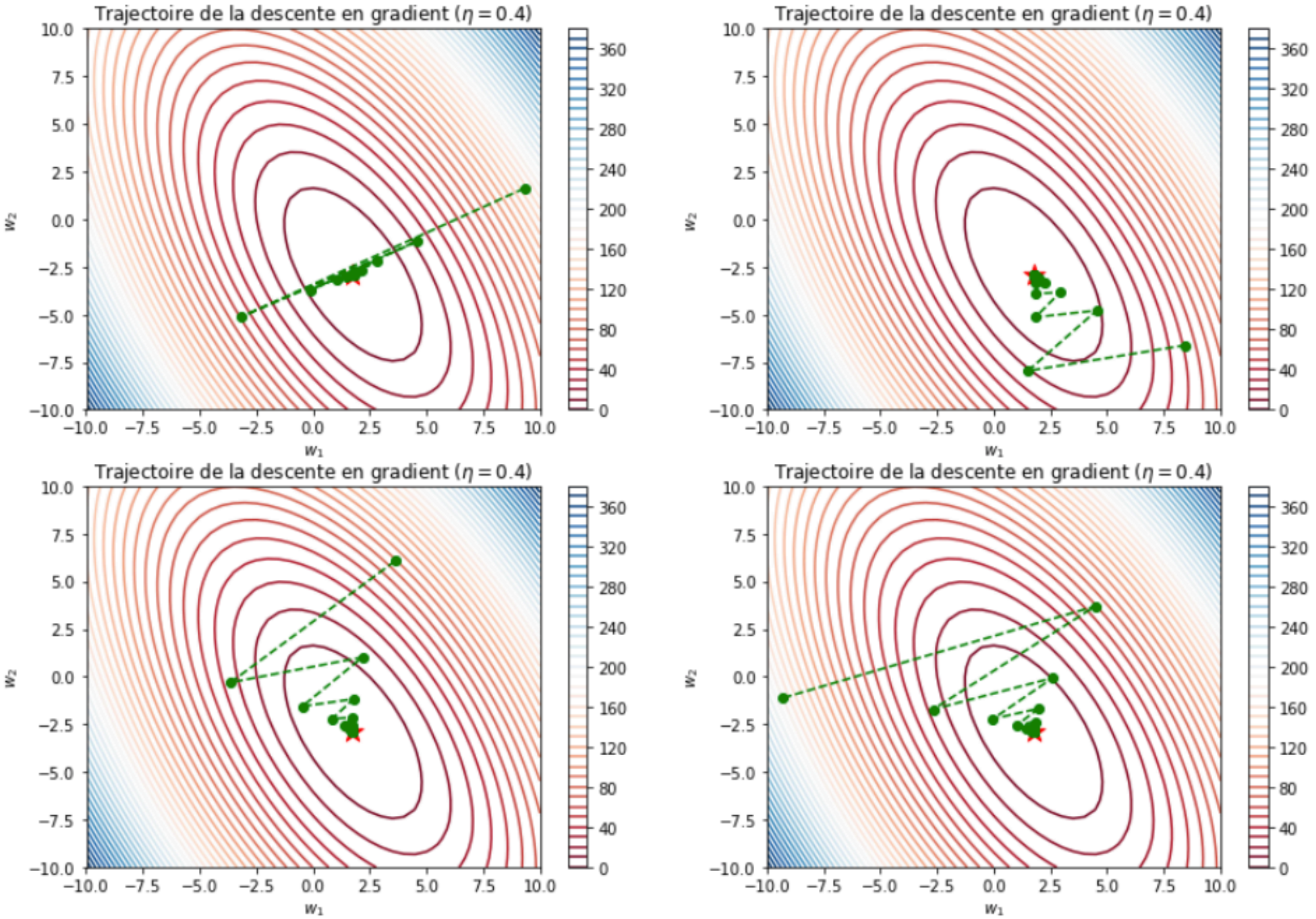
$\eta = 0.5$ $T = 20$



# Different initializations

$\eta = 0.4$ $T = 20$



# Stochastic gradient descent

Algorithm

$$F(\mathbf{w}) = \frac{1}{n}\sum_{i=1}^{n}(\mathbf{w}\cdot\mathbf{x}_i - y_i)^2 = \frac{1}{n}\sum_{i=1}^{n} F_i(\mathbf{w}),$$

where $F_i(\mathbf{w}) = (\mathbf{w}\cdot\mathbf{x}_i - y_i)^2$ and $\nabla F_i(\mathbf{w}) = 2(\mathbf{w}\cdot\mathbf{x}_i - y_i)\,\mathbf{x}_i$.

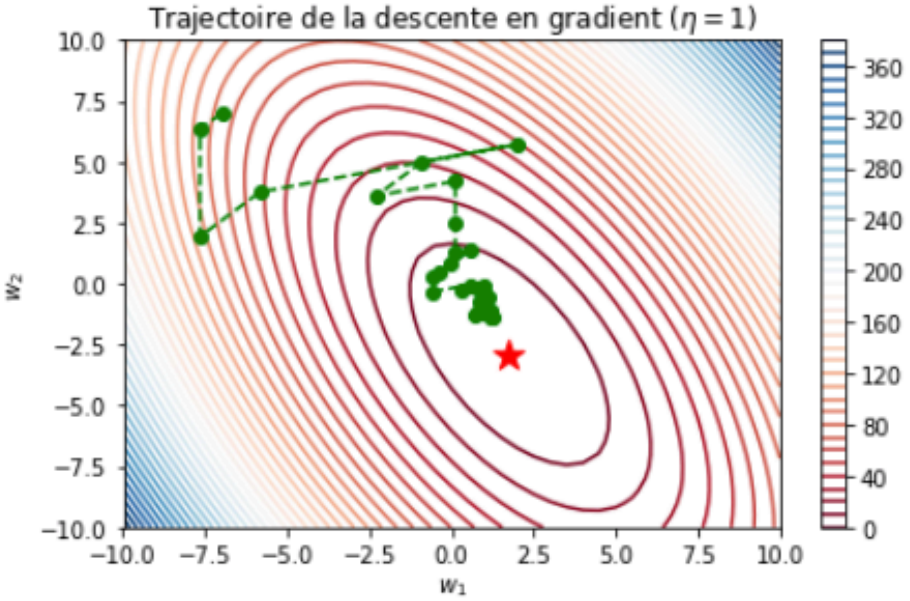
Gradient step $\eta$, number of iterations $T$

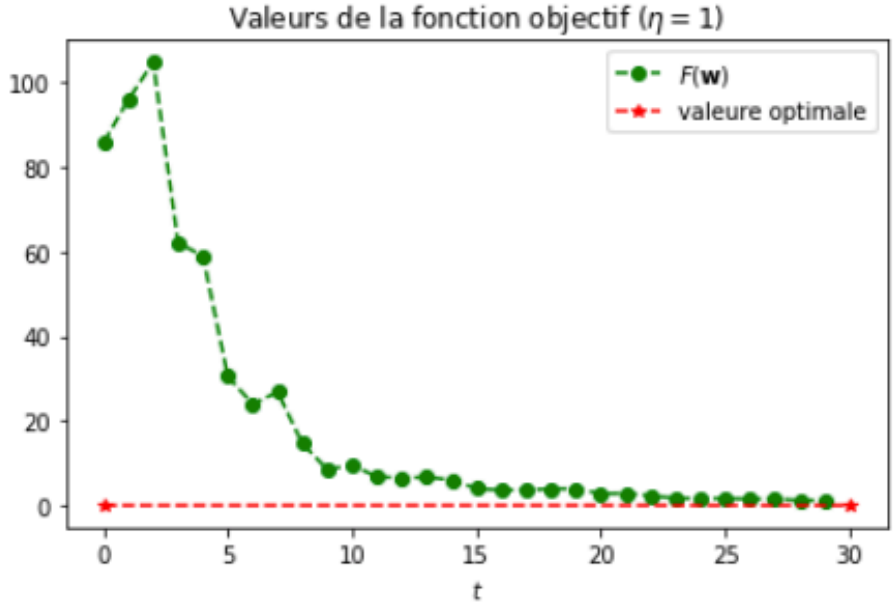
- Initialize $\mathbf{w}_0 \in \mathbb{R}^d$ randomly
- For $t$ from $1$ to $T$
  - Choose randomly $i \in \{1,\ldots n\}$
  - $\mathbf{g}_t = \nabla F_i(\mathbf{w}_{t-1})$
  - $\mathbf{w}_t = \mathbf{w}_{t-1} - \frac{\eta}{t}\mathbf{g}_t$
- Return $\mathbf{w}_T$

# Stochastic gradient descent

Example

$\eta = 1$ $T = 20$



# References

Machine Learning online course

**https://www.coursera.org/learn/machine-learning**

Deep Learning Book (Chapter 5)

**Ian Goodfellow and Yoshua Bengio and Aaron Courville**

**https://www.deeplearningbook.org/**