An introduction to Machine Learning.

Demystification, foundations and state of the art.

Paul KRZAKALA

PhD student at Institut Polyechnique.

Machine Learning vs IA

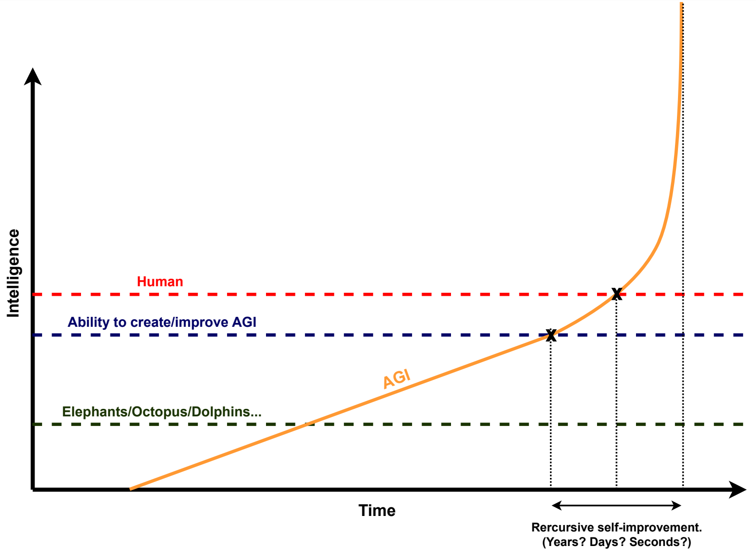

AT WORK
MAN, MACHINE LEARNING IS FUN.
AT A CONFERENCE
SO HERE'S OUR LATEST MACHINE LEARNING STUFF.
AT A PARTY
ME? I'M IN MACHINE LEARNING.
AT A FUNDING OPPORTUNITY
I WORK IN... ARTIFICIAL INTELLIGENCE.
smbc-comics.com

Artificial Intelligence (IA)

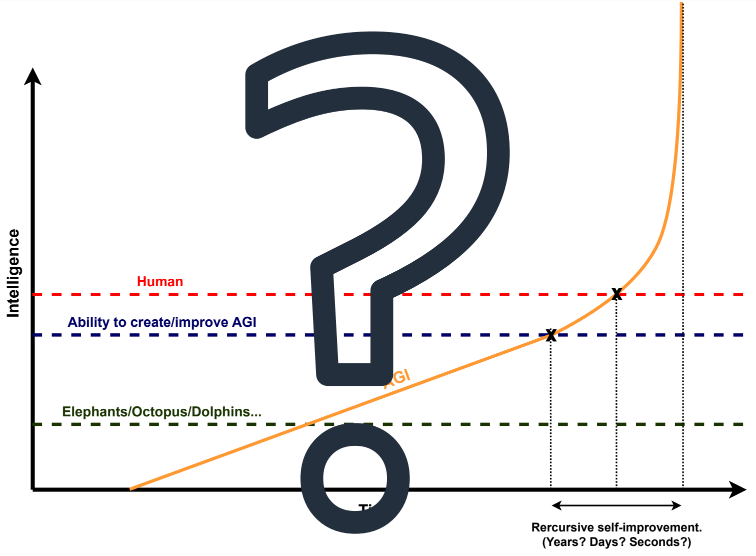
Artificial General Intelligence

- Artificial = human made (In Silico)
- Intelligence $\approx$ ability to solve problems
- General = Wide range of problems (comparable to humans)

The singularity



The singularity



"If its written in python its probably Machine Learning.

If its written in power point its probably AI."

Mat Velloso.

Machine Learning

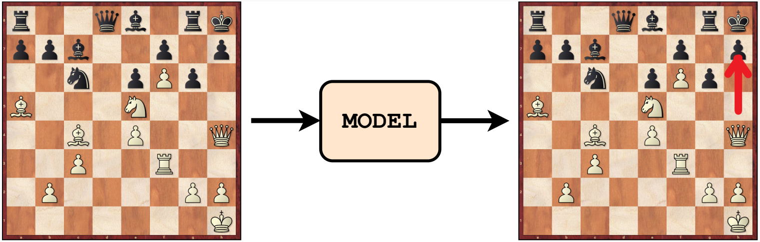
ML = The art of making computer learn to perform a given task.

The task is narrow, in practice it is formulated as finding a function.

The product is a ”model” which is a Artificial Weak Intelligence.

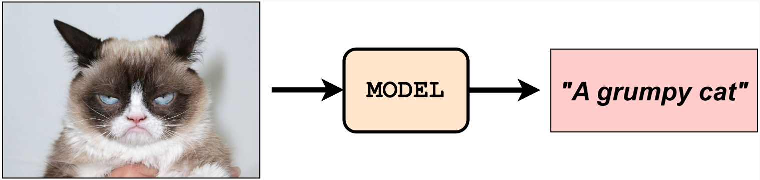
Some ML tasks:

Find a function that maps a chess position to the best move.



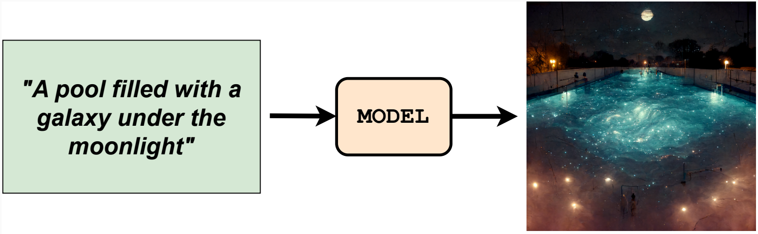
Some ML tasks:

Find a function that an image to a description.



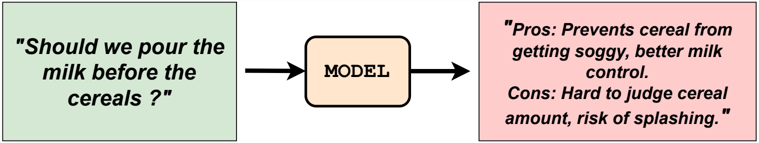
Some ML tasks:

Find a function that maps a description to an image.



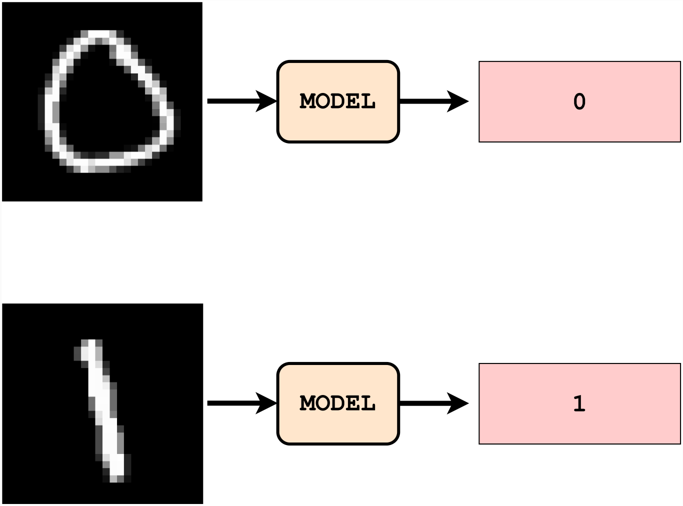
Some ML tasks:

Find a function that maps a prompt to a human-like answer.



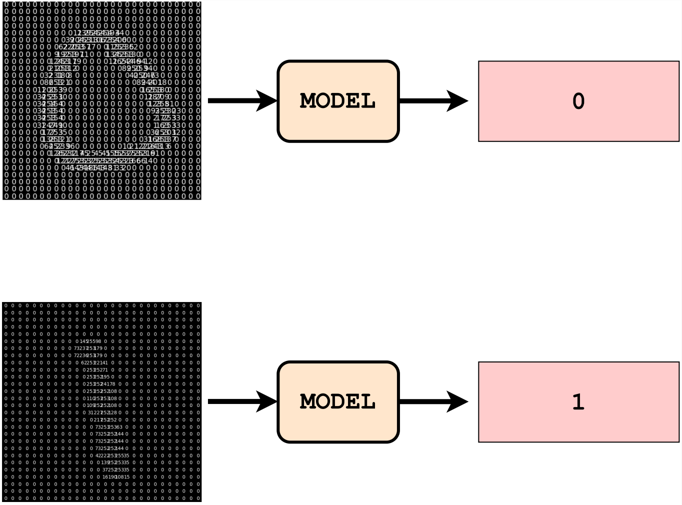
Note 1: complexity of the tasks

Find a function that maps an image to its label $\{0, 1\}$. (MNIST)



Note 1: complexity of the tasks

Find a function that maps an image to its label $\{0, 1\}$. (MNIST)



Note 2: LLMs are bluring the gap between ML and IA

LLM = Large anguage Model

Prompt: Here we have a book, 9 eggs, a laptop, a bottle and a nail. Please tell me how to stack them onto each other in a stable manner.

GPT-4:

One possible way to stack the objects onto each other in a stable manner is:

- Place the book flat on a level surface, such as a table or a floor. The book will serve as the base of the stack and provide a large and sturdy support.
- Arrange the 9 eggs in a 3 by 3 square on top of the book, leaving some space between them. The eggs will
form a second layer and distribute the weight evenly. Make sure the eggs are not cracked or broken, and handle them gently to avoid dropping them.
- Place the laptop on top of the eggs, with the screen facing down and the keyboard facing up. The laptop will fit snugly within the boundaries of the book and the eggs, and its flat and rigid surface will provide a stable platform for the next layer. Make sure the laptop is closed and turned off, and do not press any keys or buttons.
- Place the bottle on top of the laptop, with the cap facing up and the bottom facing down. The bottle will add some height and balance to the stack, and its cylindrical shape will reduce the contact area with the laptop. Make sure the bottle is empty and closed, and do not shake or tilt it.
- Place the nail on top of the bottle cap, with the pointy end facing up and the flat end facing down. The nail will be the final and smallest object in the stack, and its sharp tip will prevent it from rolling or sliding off the bottle. Make sure the nail is clean and straight, and do not touch or move it.

Note 2: LLMs are bluring the gap between ML and IA

LLM = Large Language Model

Prompt: Here we have a book, 9 eggs, a laptop, a bottle and a nail. Please tell me how to stack them onto each other in a stable manner.

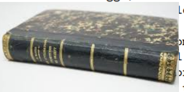
GPT-4:









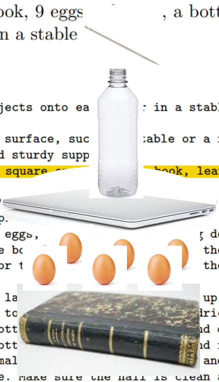



```
One possible way to stack th          :o each       in a stable manner is:

- Place the book flat on a level sur      such as a table       loor. The book will serve as the base of
the stack and provide a large and st      uppor
- Arrange the 9 eggs in a 3 by 3 square on top      book, leavi        :e bet      m. The eggs
will
form a second layer and distribute the weight evenly. Make sure tl          not c       r broken, and
handle them gently to avoid dropping them.
- Place the laptop on top of the eggs, with the screen facing down and the keyboar       up. The
laptop wi''        ,e boundaries of the book and the eggs, and its flat       gid surface
will p          the next layer.         losed a       ed off, and do
not pr
- Place          aptop, with the c          om facing down. The bottle
will add some          lance to the stack, and          l reduce the contact area
with the laptop. Make sure the bottle is empty and          or tilt it.
- Place the nail on top of the bottle cap, with the          the flat end facing down.
The nail will be the final and smallest object in the stack, and its sharp tip will prevent it from
rolling or sliding off the bottle. Make sure the nail is clean and straight, and do not touch or move
it.
```

LLMs are bluring the gap between ML and IA

LLM = Large Language Model



Prompt: Here we have a book, 9 eggs , a bottle and a nail. Please tell me how to stack them onto each other in a stable

GPT-4:

One possible way to stack the objects onto ea in a stable manner is:

- Place the book flat on a level surface, suc able or a floor. The book will serve as the base of the stack and provide a large and sturdy supp
- Arrange the 9 eggs in a 3 by 3 square book, leaving some space between them. The eggs will form a second layer and distribut the eggs are not cracked or broken, and handle them gently to avoid dropp
- Place the laptop on top of the eggs, down and the keyboard facing up. The laptop will fit snugly within the bo the eggs, and its flat and rigid surface will provide a stable platform for t the laptop is closed and turned off, and do not press any keys or buttons.
- Place the bottle on top of the la up and the bottom facing down. The bottle will add some height and balance to drical shape will reduce the contact area with the laptop. Make sure the bott d do not shake or tilt it.
- Place the nail on top of the bott d facing up and the flat end facing down. The nail will be the final and smal and its sharp tip will prevent it from rolling or sliding off the bottle. Make sure the nail is clean and straight, and do not touch or move it.

LLMs are bluring the gap between ML and IA

Recipe for an AGI ?

- Find a very complex task
- Train a very complex model to solve it

Did human followed the same path?

- Task: Survive/Reproduce/pass your DNA
- Model: brain, optimizer: natural selection

Foundations of the Machine Learning Paradigm

ML paradigms:

- Supervised
- Unsupervised
- Reinforcement Learning
- Generative modeling
- And others!

Supervised learning

An input space $\mathcal{X}$, an output space $\mathcal{Y}$ and a distribution $\mathbb{P}$ on $\mathcal{X} \times \mathcal{Y}$

$$(X, y) \sim \mathbb{P} \tag{1}$$

Minimize the expected risk :

$$R(f) = \mathbb{E}[\ell(f(X), y)] \tag{2}$$

$\ell : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}$ is a measure of the error (context dependant)

Supervised learning in practice

Bayes Optimal:

$$f^* = \arg\min_{f \in \mathcal{X}^{\mathcal{Y}}} \mathbb{E}[\ell(f(X), y)] \tag{3}$$

1) Restrict to an "Hypothesis space" $\mathcal{H} \subset \mathcal{X}^{\mathcal{Y}}$

$$\bar{f} = \arg\min_{f \in \mathcal{H}} \mathbb{E}[\ell(f(X), y)] \tag{4}$$

2) Only access to N samples $(X_i, y_i)_{i \in 1,N}$

$$\hat{f} = \arg\min_{f \in \mathcal{H}} \frac{1}{N} \sum_{i=1}^{N} \ell(f(X_i), y_i) \tag{5}$$

Supervised learning in practice

$$f^* = \arg\min_{f \in \mathcal{X}^{\mathcal{Y}}} \mathbb{E}[\ell(f(X), y)]$$

$$\bar{f} = \arg\min_{f \in \mathcal{H}} \mathbb{E}[\ell(f(X), y)]$$

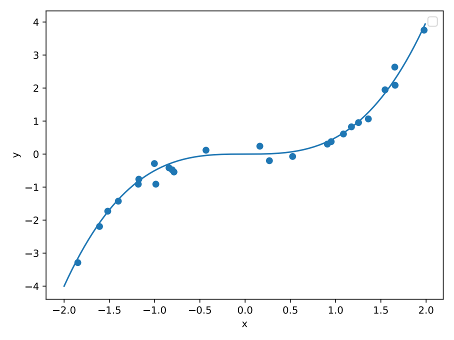
$$\hat{f} = \arg\min_{f \in \mathcal{H}} \frac{1}{N} \sum_{i=1}^{N} \ell(f(X_i), y_i)$$

- You typically do not control N, the more data the better.
- You are free to pick $\mathcal{H}$. But must choose carefully its "complexity" $C(\mathcal{H})$.

$$f^* \quad \underbrace{\Longleftarrow\!\!\!=\!=\!=\!=\!\!\!\Longrightarrow}_{\searrow \text{ with } C(\mathcal{H})} \quad \bar{f} \quad \underbrace{\Longleftarrow\!\!\!=\!=\!=\!=\!\!\!\Longrightarrow}_{\nearrow \text{ with } C(\mathcal{H})} \quad \hat{f} \tag{6}$$

An example:

The task: $N = 25$, $f^*(x) = x^3$, $\ell(y, y') = (y - y')^2$...



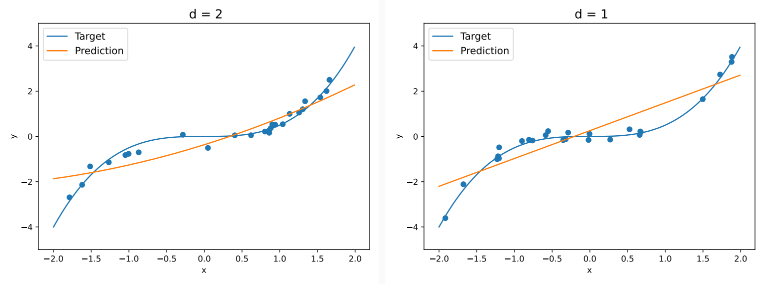
The model: $\mathcal{H}_d = \{x \mapsto \sum_{k=0}^{d} \alpha_k x^k / \alpha_1, ..., \alpha_k \in \mathbb{R}\}$

Complexity is well defined: $C(\mathcal{H}_d) = d$

Underfitting

If the hypothesis space is too small, the model won't be able to fit the training data. This is called underfitting.

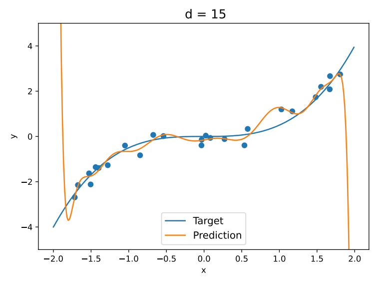
The model model weakly depends on the train data, there is too much biais.

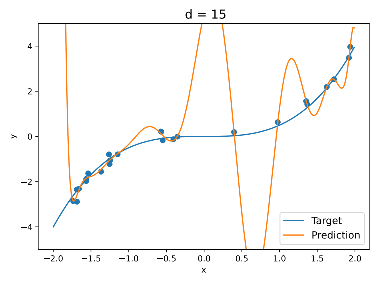


Overfitting

If the hypothesis space is too large, the model will perfectly fit the training data, including noise. This is called overfitting.

The model model depends too much on the train data, there is too much variance.





Biais-Variance Tradeoff

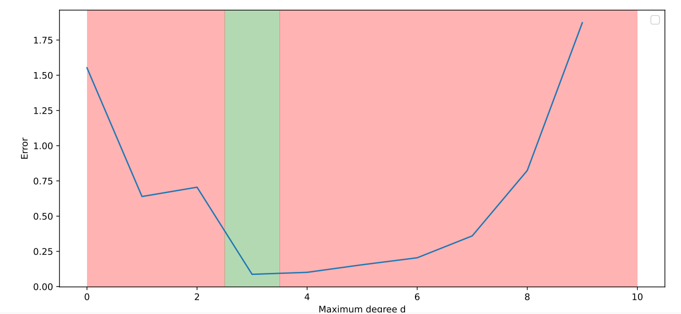


Figure 1: $R(\hat{f})$ vs $C(\mathcal{H})$

Regularization at the rescue

Regularization is a critical tool to control the complexity.

$$\mathcal{H}_d = \{x \mapsto \sum_{k=0}^{d} \alpha_k x^k / \alpha_1, ..., \alpha_k \in \mathbb{R}\} \qquad \text{(No Regularization)}$$

$$\mathcal{H}_{d,\lambda} = \{x \mapsto \sum_{k=0}^{d} \alpha_k x^k / \alpha_1, ..., \alpha_k \in [-\lambda, \lambda]\} \qquad \text{(Regularization } \lambda\text{)}$$

- $\mathcal{H}_d = \mathcal{H}_{d,+\infty}$
- $C(\mathcal{H}_{d,\lambda})$ with $\nearrow \lambda$

Regularization at the rescue

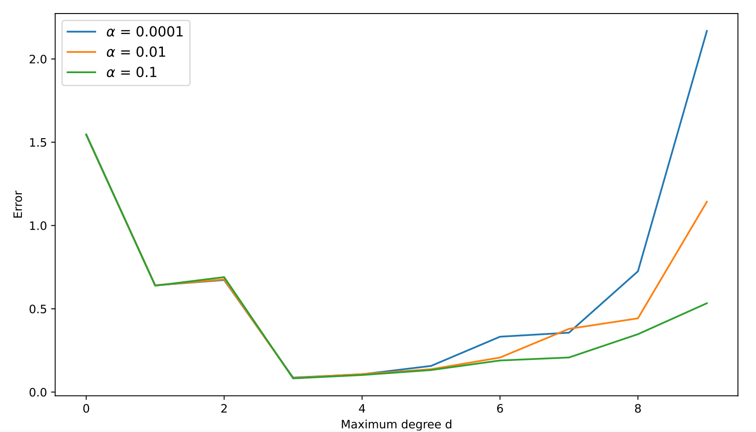
In practice regularization is done with an additional term to the loss

$$\hat{f} = \underset{f \in \mathcal{H}}{\arg\min} \frac{1}{N} \sum_{i=1}^{N} \ell(f(X_i), y_i) \qquad \text{(No Regularization)}$$

$$\hat{f}_\alpha = \underset{f \in \mathcal{H}}{\arg\min} \frac{1}{N} \sum_{i=1}^{N} \ell(f(X_i), y_i) + \alpha ||f||_2^2 \qquad \text{(Regularization } \alpha\text{)}$$

where $|| \sum_{k=0}^{d} \alpha_k x^k ||_2^2 = \sum_{k=0}^{d} \alpha_k^2$

Regularization at the rescue



Deep Learning.

Neural Networks

Neural Networks = type of hypothesis space.

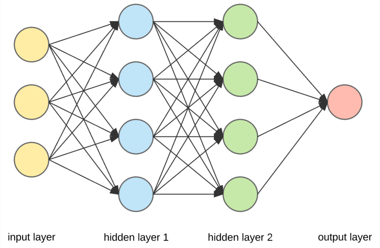
- You get a NN by stacking/composition of layers.

$$f(x) = \mathrm{L}_K \circ \ldots \circ \mathrm{L}_2 \circ \mathrm{L}_1(x) \tag{7}$$

- The most classical layer is:

$$\mathrm{L}_k(x) = \sigma(Ax) \tag{8}$$

A = learnable matrix, σ = non-learnable non-linear function.



Neural Networks advantages

Complexity

- Control complexity with depth (number of layers) and width (dimensions of hidden layers).
- Universal approximation theorems exists.

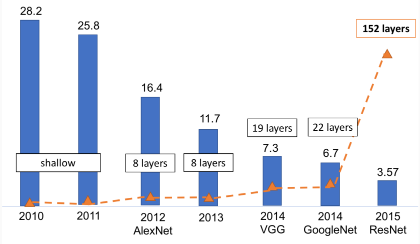
Optimization

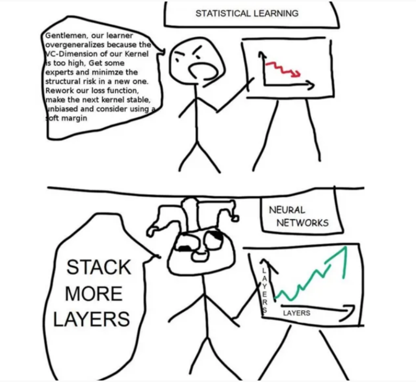
- Easy chain-rule computation of the gradient w.r.t. to parameters
- 1st order optimizer (gradient descent)

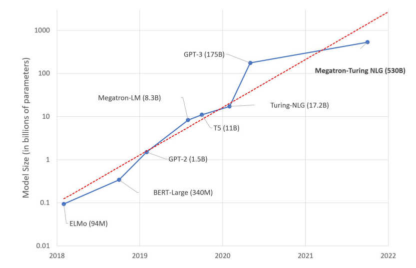
Design

- Easy to design variations (architecture), just design a new layer.
- Simply pick the architecture adapted to your task

Deep Learning: layers goes brrrr







Double descent: the dark matter of modern ML

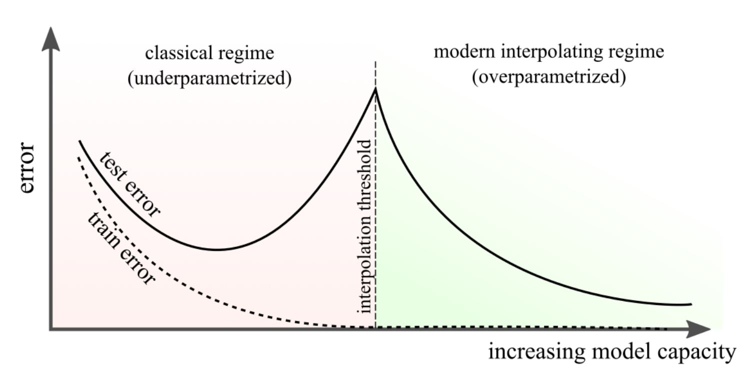


Figure 2: WTF is going on ???

The role of the optimization algorithm

Recall the 2 approximation at the foundation of ML:

$$f^* = \underset{f \in \mathcal{X}^{\mathcal{Y}}}{\arg\min}\, \mathbb{E}[\ell(f(X), y)]$$

$$\bar{f} = \underset{f \in \mathcal{H}}{\arg\min}\, \mathbb{E}[\ell(f(X), y)]$$

$$\hat{f} = \underset{f \in \mathcal{H}}{\arg\min}\, \frac{1}{N} \sum_{i=1}^{N} \ell(f(X_i), y_i)$$

We also need to take into account a third approximation:

$$f_{\mathtt{optim}} = ALG \left[\min_{f \in \mathcal{H}} \frac{1}{N} \sum_{i=1}^{N} \ell(f(X_i), y_i) \right] \tag{9}$$

Gradient Descent

Minimize:

$$\min_{\theta} \mathcal{L}(\theta) \tag{10}$$

Following:

$$\theta_{t+1} = \theta_t - \eta \nabla_\theta \mathcal{L}(\theta_t) \tag{11}$$

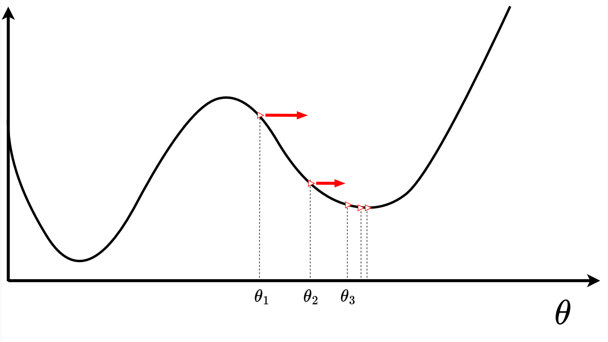
Stochastic gradient descent (SGD) = noisy gradient descent

$$\theta_{t+1} = \theta_t - \eta \Delta_t \tag{12}$$

where $E[\Delta_t] = \nabla_\theta \mathcal{L}(\theta_t)$

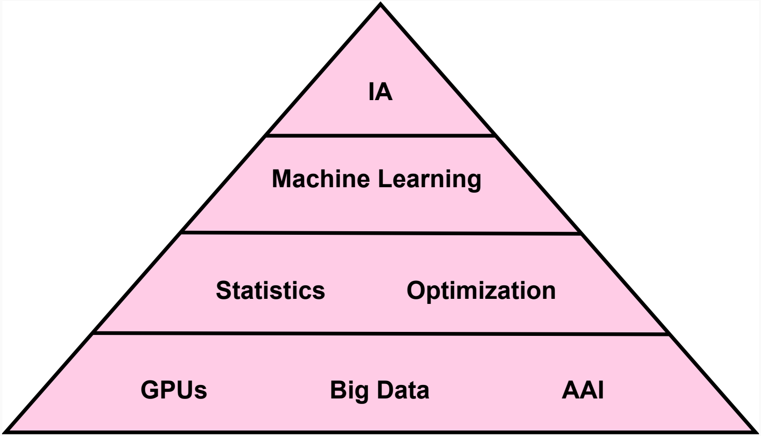
Possible explanations for double descent?

Intuition 1: gradient descent converges better in high dimension (less likely to be trapped in local minima)



Intuition 2: SGD provides implicit regularization.

ML foundation: Takeaway


IA
Machine Learning
Statistics
Optimization
GPUs
Big Data
AAI