ES 691
Mathematics for Machine Learning

with

Dr. Naveed R. Butt

@

GIKI - FES

- So far we’ve seen the magic of learning in
  - Neurons, organisms, algorithms, and materials

- But what is it that makes ML so *special and powerful*?
- And have we seen examples of "*mathematics that learns*" before?
- Concerns, Limitations and Open Problems in ML

What Makes ML So Special?

- Although there is no one specific answer, and may vary from ML to ML, some broader aspects include…

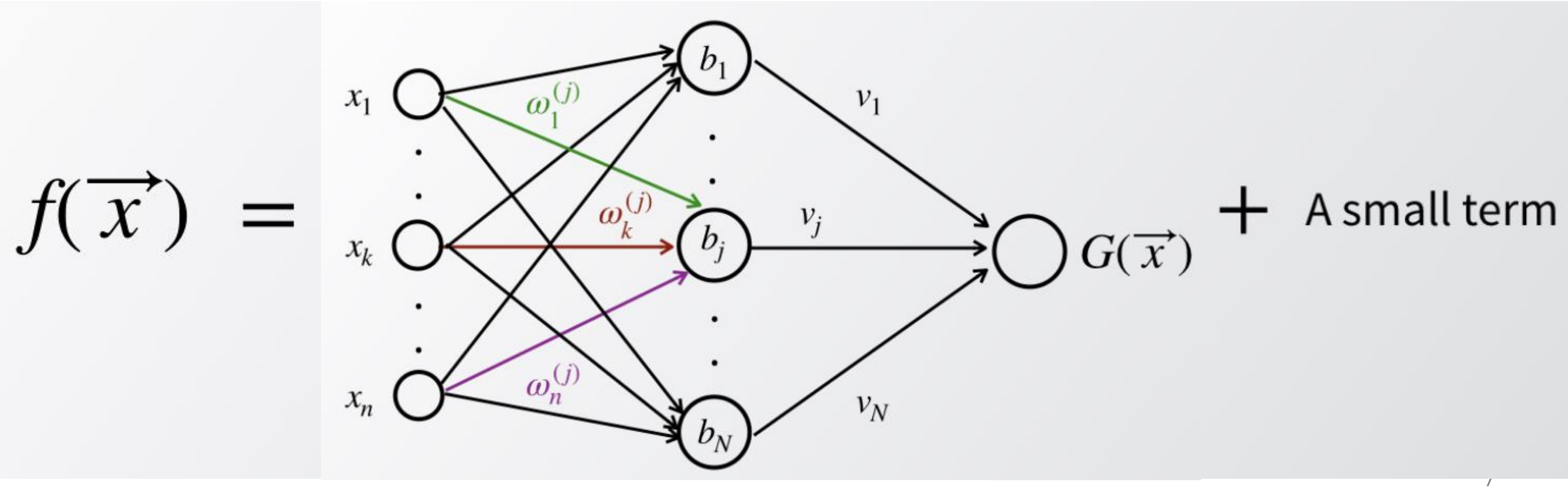
The Power of Universal Approximation…

The Power of Universal Approximation…

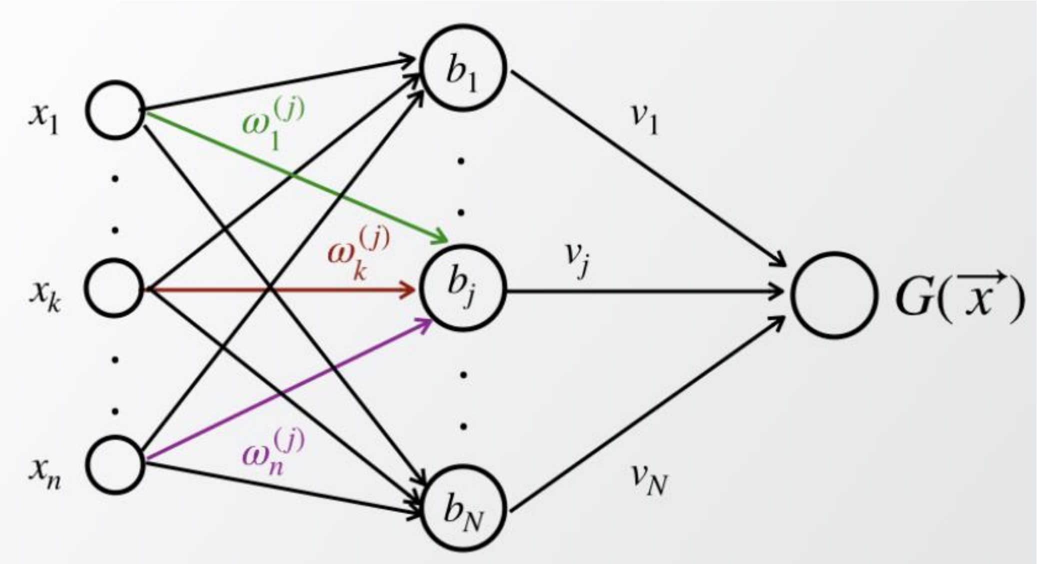
- NN with 1 hidden layer can represent:
  - any bounded continuous function (to arbitrary ε)
    - Universal Approximation Theorem [Cybenko 1989]
  - any Boolean function (exactly)

The Power of Universal Approximation…

- NN with 1 hidden layer can represent:
  - any bounded continuous function (to arbitrary ε)
    - Universal Approximation Theorem [Cybenko 1989]
  - any Boolean function (exactly)



The Power of Universal Approximation…



Theorem (Cybenko)

Let σ be any continuous discriminatory function. Then finite sums of the form

$$G(x) = \sum_{j=1}^{N} \alpha_j \sigma(w_j^T x + b_j), \quad \text{where } w_j \in \mathbb{R}^n, \ \alpha_j, \ b_j \in \mathbb{R}$$

are dense in $C(I_n)$.

In other words, given any $\varepsilon > 0$ and $f \in C(I_n)$, there is a sum $G(x)$ of the above form such that

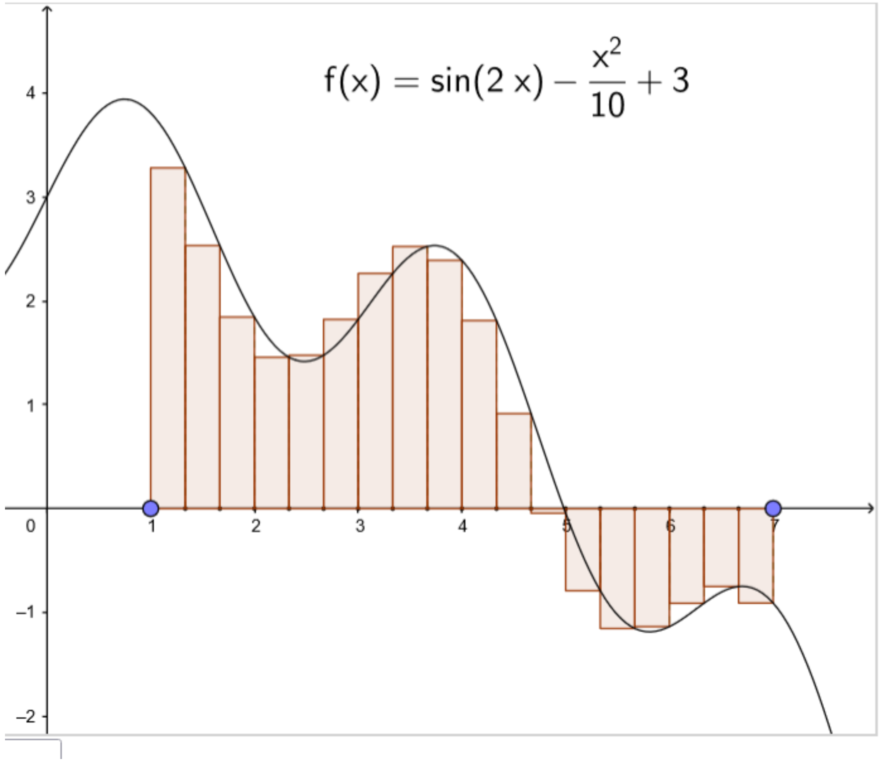
$$|G(x) - f(x)| < \varepsilon, \quad \forall x \in I_n$$

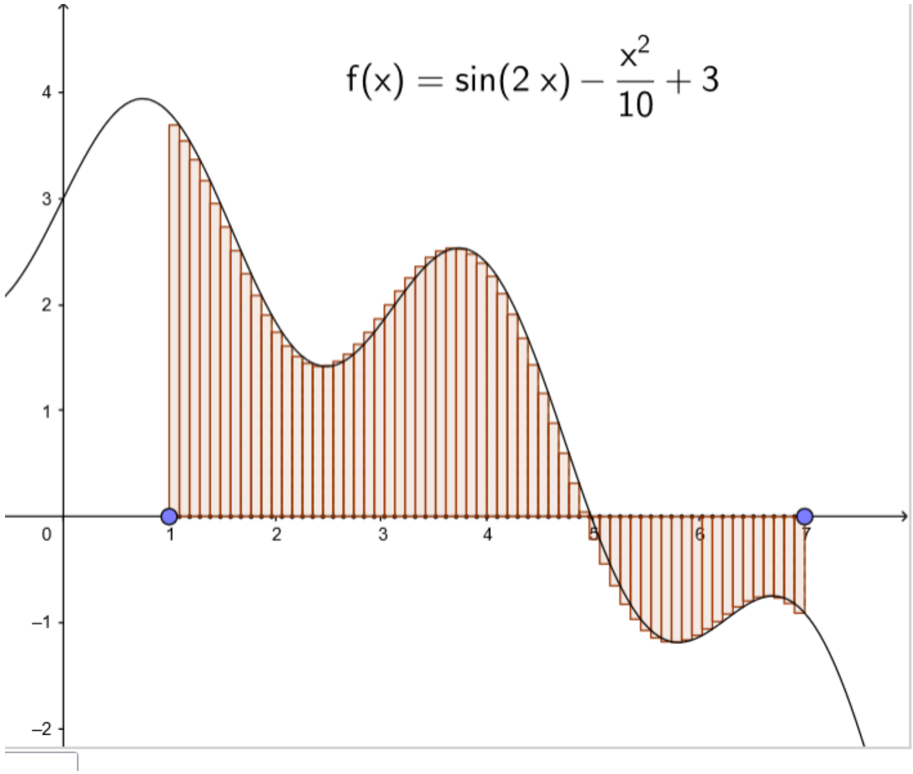
But Why?

But Why?

We all know about piece-wise linear approximations.

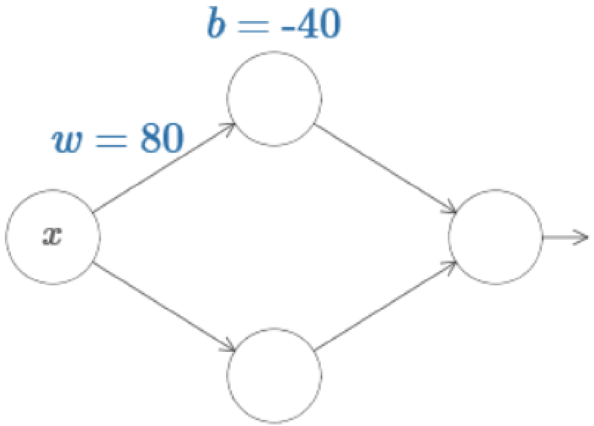
But Why?

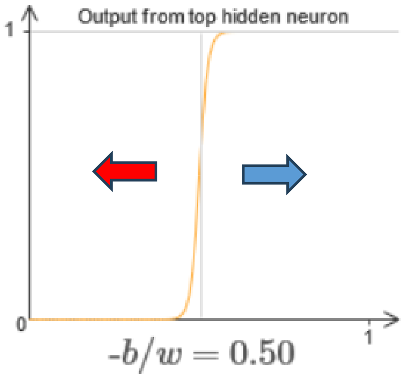




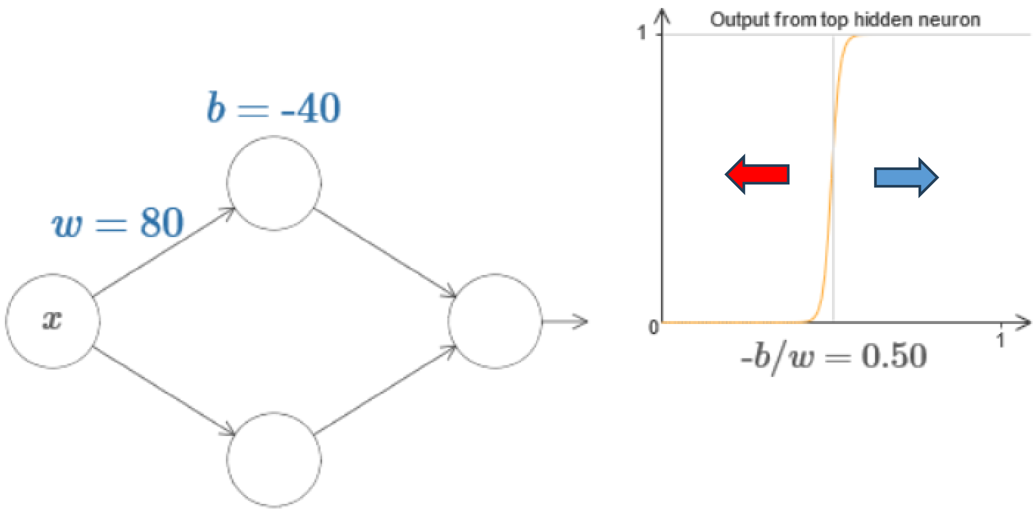
We all know about piece-wise linear approximations.

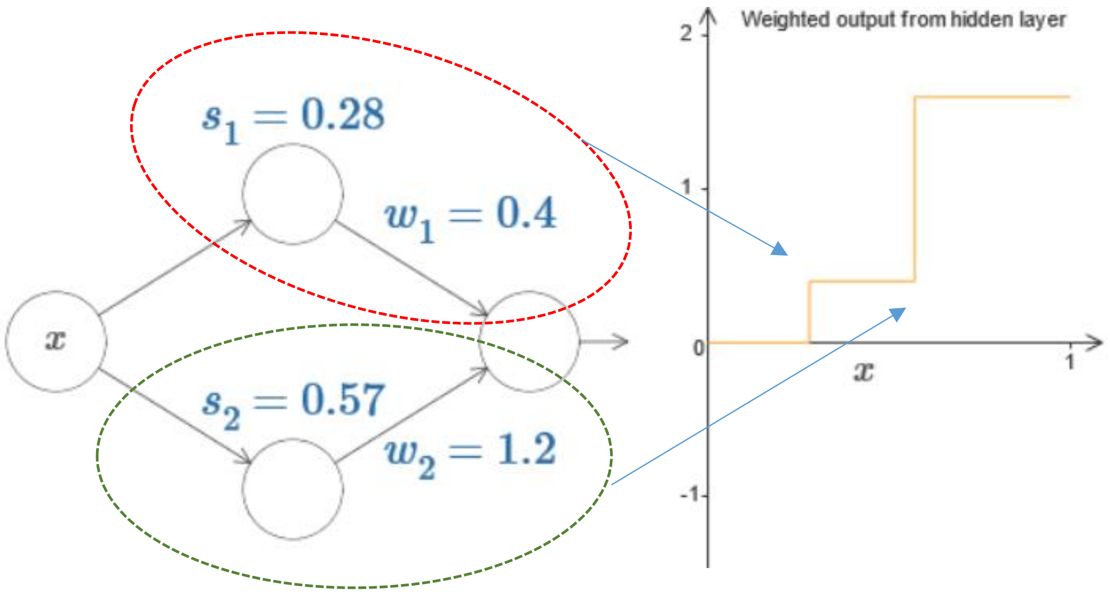
..we could split any continuous function into several regions and approximate its value in each region by height of a rectangle and still get arbitrarily close to the function values (as long as we have sufficient number of regions)!



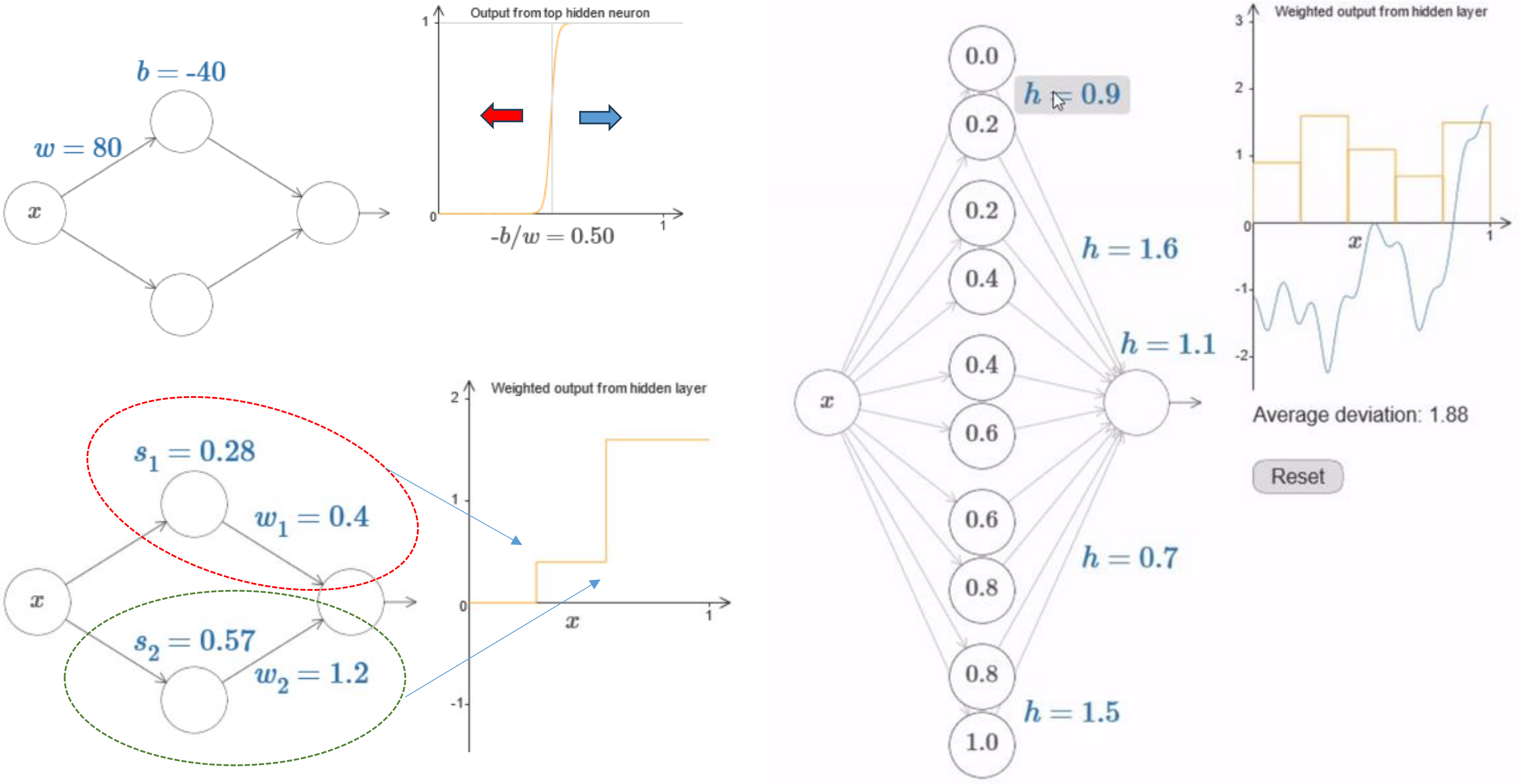


Source: neuralnetworksanddeeplearning.com (a MUST VISIT Interactive page by Michael Nielsen to see all this in action).





Source: neuralnetworksanddeeplearning.com (a MUST VISIT Interactive page by Michael Nielsen to see all this in action).



Source: neuralnetworksanddeeplearning.com (a MUST VISIT Interactive page by Michael Nielsen to see all this in action).

- But Universal Approximation is only part of the story…
  - Why?

- But Universal Approximation is only part of the story…
  - Why? Because we have so many other universal approximators…
  - E.g.,

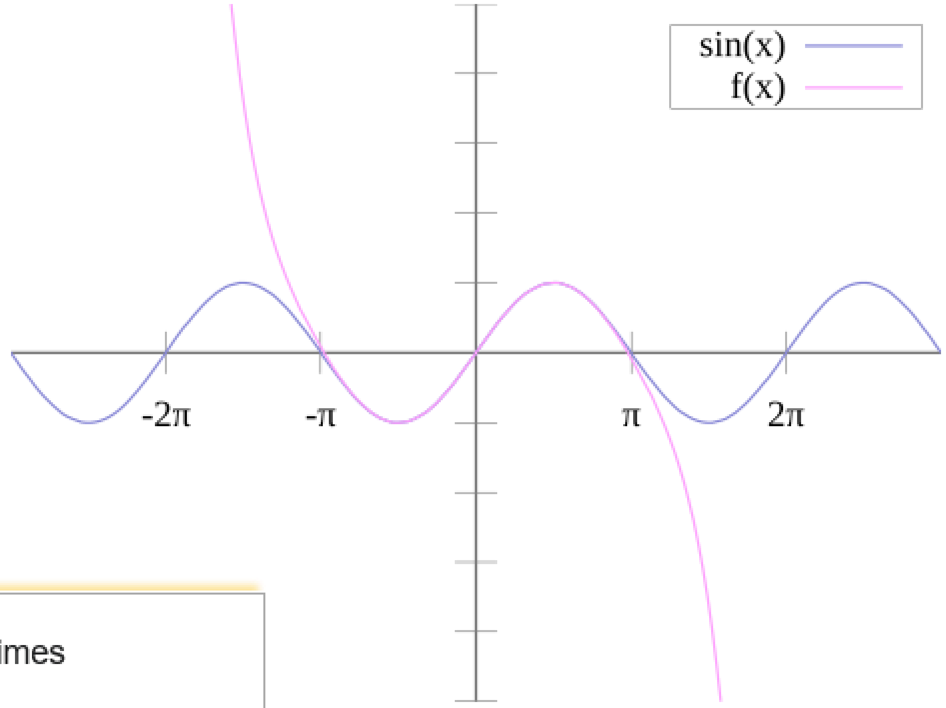
- But Universal Approximation is only part of the story…
  - Why? Because we have so many other universal approximators…
  - E.g., Polynomials, Power Series, Fourier Series…

…other “Universal” Approximators

$$P(x) = a_n x^n + a_{n-1} x^{n-1} + a_{n-2} x^{n-2} + \cdots + a_2 x^2 + a_1 x + a_0$$

…other "Universal" Approximators

$$P(x) = a_n x^n + a_{n-1} x^{n-1} + a_{n-2} x^{n-2} + \cdots + a_2 x^2 + a_1 x + a_0$$



Weierstrass Approximation Theorem

- If f(x) is a continuous real-valued function on [a, b] then for any $\varepsilon > 0$, then there exists a polynomial P_n on [a, b] such that

$$|f(x) - P_n(x)| < \varepsilon$$

for all $x \in [a, b]$.

Taylor's theorem[4][5][6] — Let $k \geq 1$ be an integer and let the function $f : \mathbf{R} \to \mathbf{R}$ be k times differentiable at the point $a \in \mathbf{R}$. Then there exists a function $h_k : \mathbf{R} \to \mathbf{R}$ such that

$$f(x) = f(a) + f'(a)(x-a) + \frac{f''(a)}{2!}(x-a)^2 + \cdots + \frac{f^{(k)}(a)}{k!}(x-a)^k + h_k(x)(x-a)^k,$$

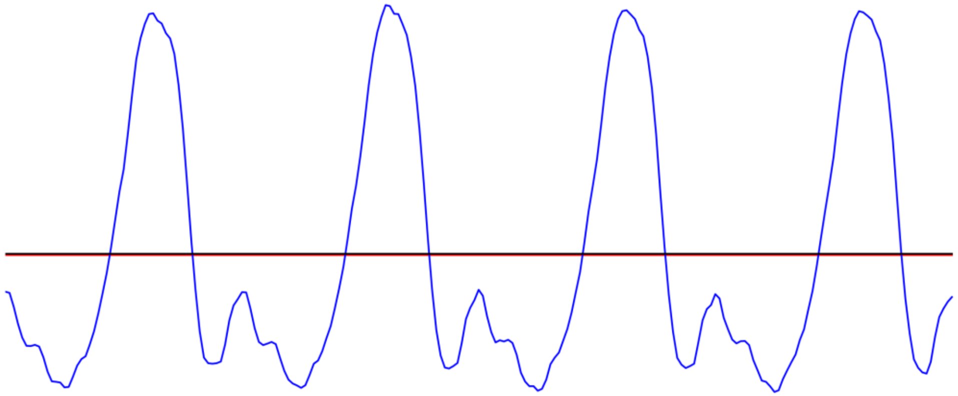
and

$$\lim_{x \to a} h_k(x) = 0.$$

…other "Universal" Approximators

Fourier series, sine-cosine form

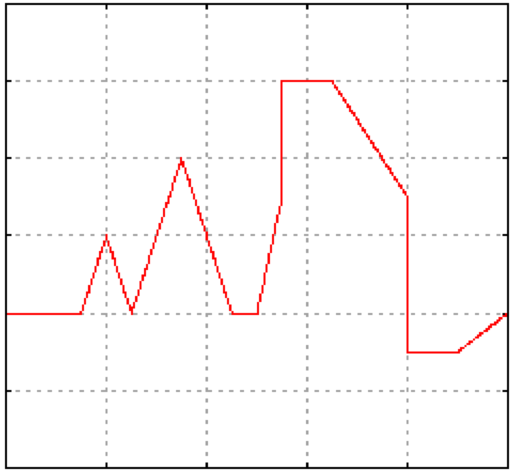
$$s_N(x) = A_0 + \sum_{n=1}^{N}\left(A_n \cos\left(2\pi \tfrac{n}{P} x\right) + B_n \sin\left(2\pi \tfrac{n}{P} x\right)\right) \quad \textbf{(Eq.2)}$$



…other "Universal" Approximators

Fourier series, sine-cosine form

$$s_N(x) = A_0 + \sum_{n=1}^{N} \left(A_n \cos\left(2\pi \tfrac{n}{P} x\right) + B_n \sin\left(2\pi \tfrac{n}{P} x\right) \right) \quad \textbf{(Eq.2)}$$



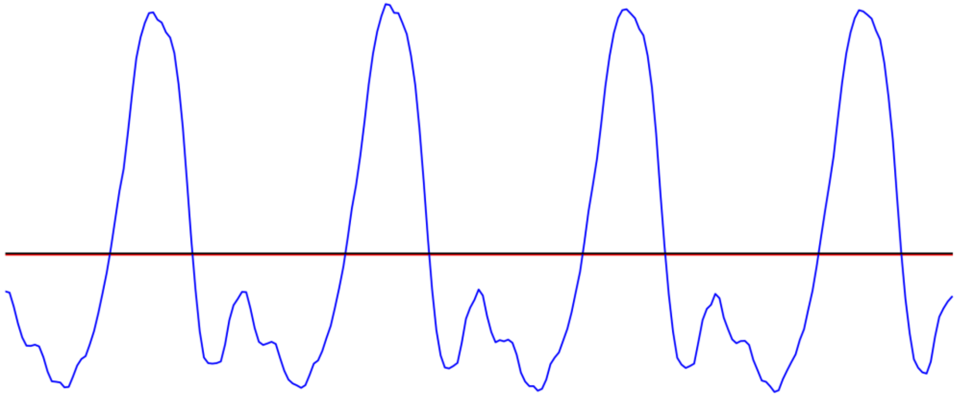
Inverse transform

$$f(x) = \int_{-\infty}^{\infty} \hat{f}(\xi)\, e^{i2\pi\xi x}\, d\xi, \quad \forall\, x \in \mathbb{R}. \quad \textbf{(Eq.2)}$$

Eq.2 is a representation of $f(x)$ as a weighted summation of complex exponential functions.

Fourier transform

$$\hat{f}(\xi) = \int_{-\infty}^{\infty} f(x)\, e^{-i2\pi\xi x}\, dx. \quad \textbf{(Eq.1)}$$



- In fact, approximation (ability to memorize and fit available data) alone is not sufficient.

- Power of “generalization” (interpolation, extrapolation, fitting to new data, learning of general class) is critical, and modern ML seems to do exceptionally well in this.

The Power of OVERPARAMETRIZATION… (and how it may help "Generalization")

Occam's Razor

The Power of OVERPARAMETRIZATION…

(and how it may help "Generalization")

Occam's Razor

Pluralitas non est ponenda sine necessitate

- "Plurality should not be posited without necessity."

In Mathematical Modeling Language: "Model parameters should not be increased without necessity".

The Power of OVERPARAMETRIZATION… (and how it may help "Generalization")

Occam's Razor

Pluralitas non est ponenda sine necessitate

- "Plurality should not be posited without necessity."

In Mathematical Modeling Language: "Model parameters should not be increased without necessity".

But…

- Overparameterization seems to help Neural Networks
  - Why?

Open question, with several theories…

The Power of OVERPARAMETRIZATION… (and how it may help "Generalization")

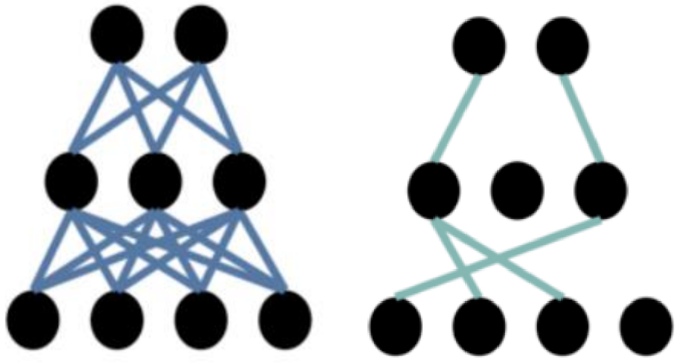
Hypothesis: Overparameterization Leads to Sparsity

- Overparameterization lets the learning algorithm choose the best options (parameters) among data-dependent couplings and essentially "discard" the rest (leading to sparsity).

The Power of OVERPARAMETRIZATION… (and how it may help "Generalization")

Hypothesis: Overparameterization Leads to Sparsity

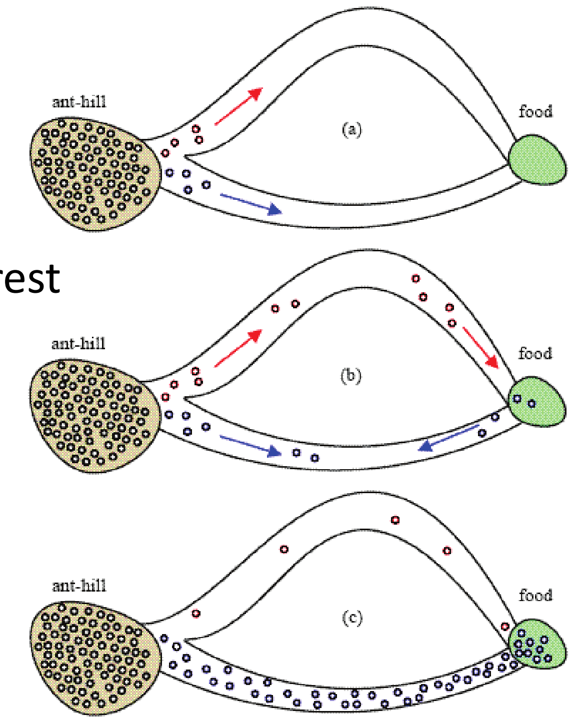
- Overparameterization lets the learning algorithm choose the best options (parameters) among data-dependent couplings and essentially "discard" the rest (leading to sparsity).



The original "dense" network (left) and its "pruned" subnet (right) both give very similar performance if subnet initialized with same weights that original network was initialized with when it successfully learned.

The Power of OVERPARAMETRIZATION…

(and how it may help "Generalization")

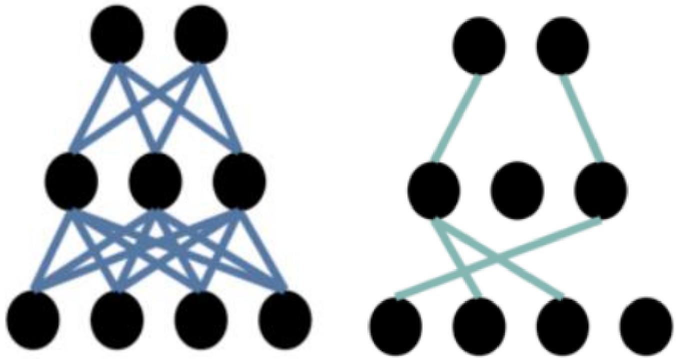


Hypothesis: Overparameterization Leads to Sparsity

- Overparameterization lets the learning algorithm choose the best options (parameters) among data-dependent couplings and essentially "discard" the rest (leading to sparsity).

Lottery Ticket Hypothesis

- Random initializations lead to some initializations that are fastest to train towards optima (recall Ant Colony, random directions taken at first).



The original "dense" network (left) and its "pruned" subnet (right) both give very similar performance if subnet initialized with same weights that original network was initialized with when it successfully learned.

The Power of OVERPARAMETRIZATION… (and how it may help "Generalization")

Overparameterization and Random Initializations Help Weight-Update Algorithm

- Gradient Descent has been shown to provably optimize overparametrized NNs.

The Power of OVERPARAMETRIZATION…

(and how it may help “Generalization”)

Overparameterization and Random Initializations Help Weight-Update Algorithm

- Gradient Descent has been shown to provably optimize overparametrized NNs.

Algorithmic Stability Hypothesis

- Algorithms that do not vary too much with slight changes in training datasets, generalize better (possibly indicating that they have learned underlying distributions rather than just the data).

The Power of OVERPARAMETRIZATION…

(and how it may help "Generalization")

Manifold Hypothesis

- Many high-dimensional data sets that occur in the real world actually lie along low-dimensional latent manifolds inside that high-dimensional space.
- As a consequence, many data sets that appear to initially require many variables to describe, can actually be described by a comparatively small number of variables.
- Machine learning models only have to fit relatively simple, low-dimensional, highly structured subspaces within their potential input space (latent manifolds).
- Within one of these manifolds, it is always possible to interpolate between two inputs, that is to say, morph one into another via a continuous path along which all points fall on the manifold.

The Power of OVERPARAMETRIZATION…

(and how it may help "Generalization")

Manifold Hypothesis

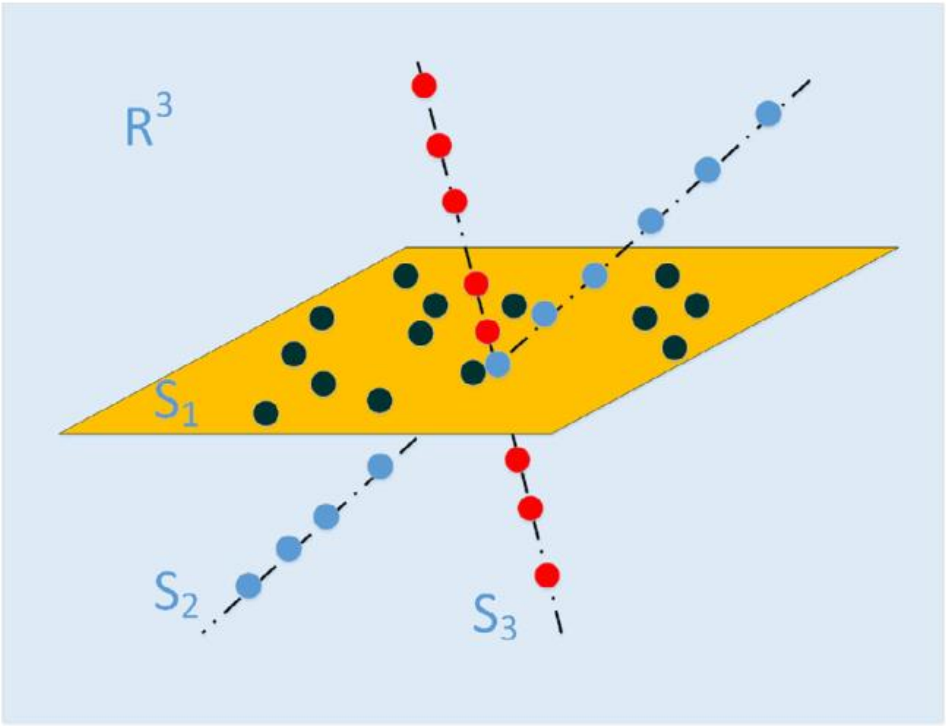


Fig. 1. Example of high-dimensional data lying in low-dimensional subspaces. It is seen that rather than uniformly distributed in the 3-dimensional space, these data points lie on the union of two lines and one plane.

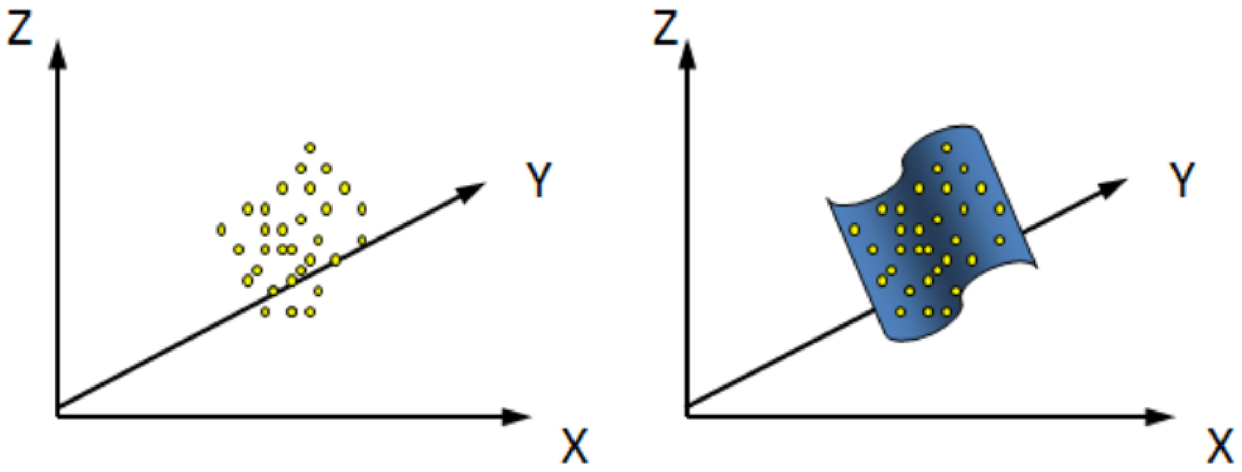


Fig. 2. Data is often embedded in (lies on) a lower-dimensional structure or manifold. It should be possible to characterize the data and the relationship between individual points using fewer dimensions, if we were able to measure distances on the manifold itself instead of in Euclidean space.

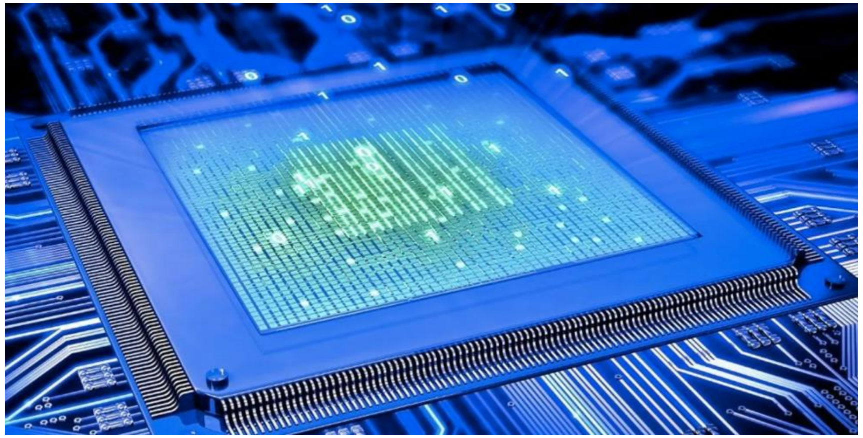
- Of course, we could get ourselves in trouble (e.g., computationally) with heavy overparameterization!!
    - Next, let’s look at some other aspects that help in this regard.

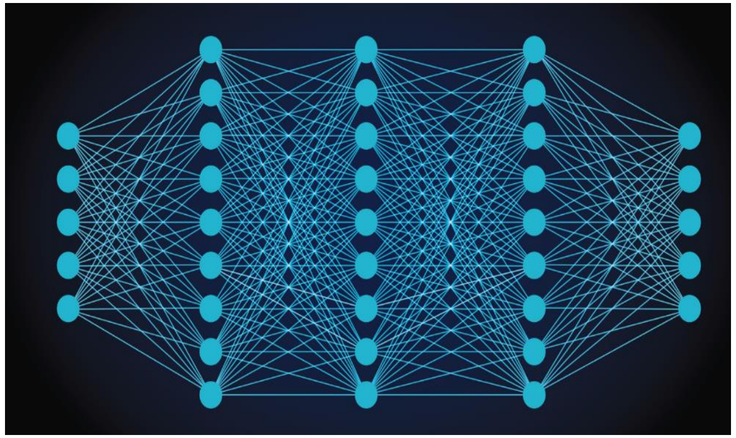
The Power of LOTS of (REPEATED) SIMPLE UNITS with SIMPLE RULES

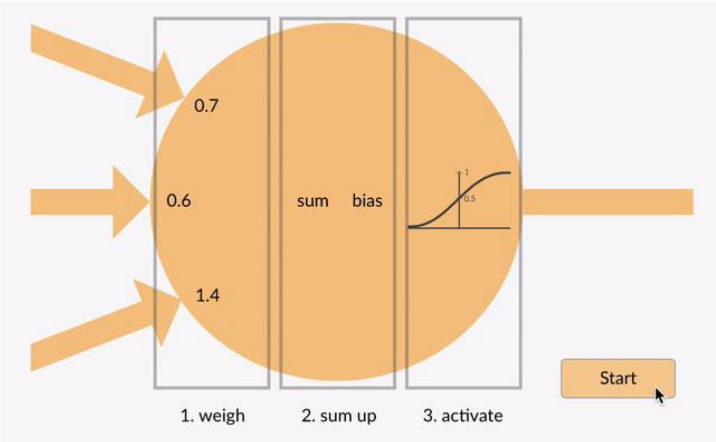
(with localized “gating” of information…)

The Power of LOTS of (REPEATED) SIMPLE UNITS with SIMPLE RULES

(with localized “gating” of information…)

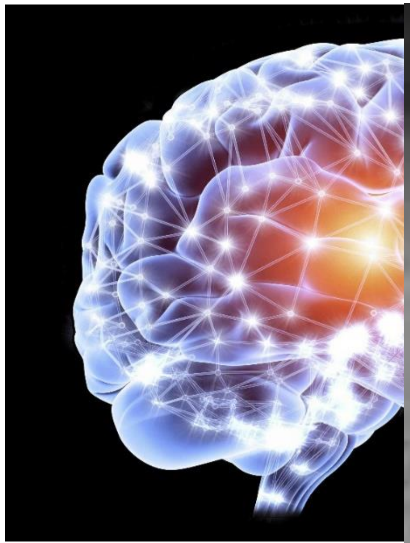


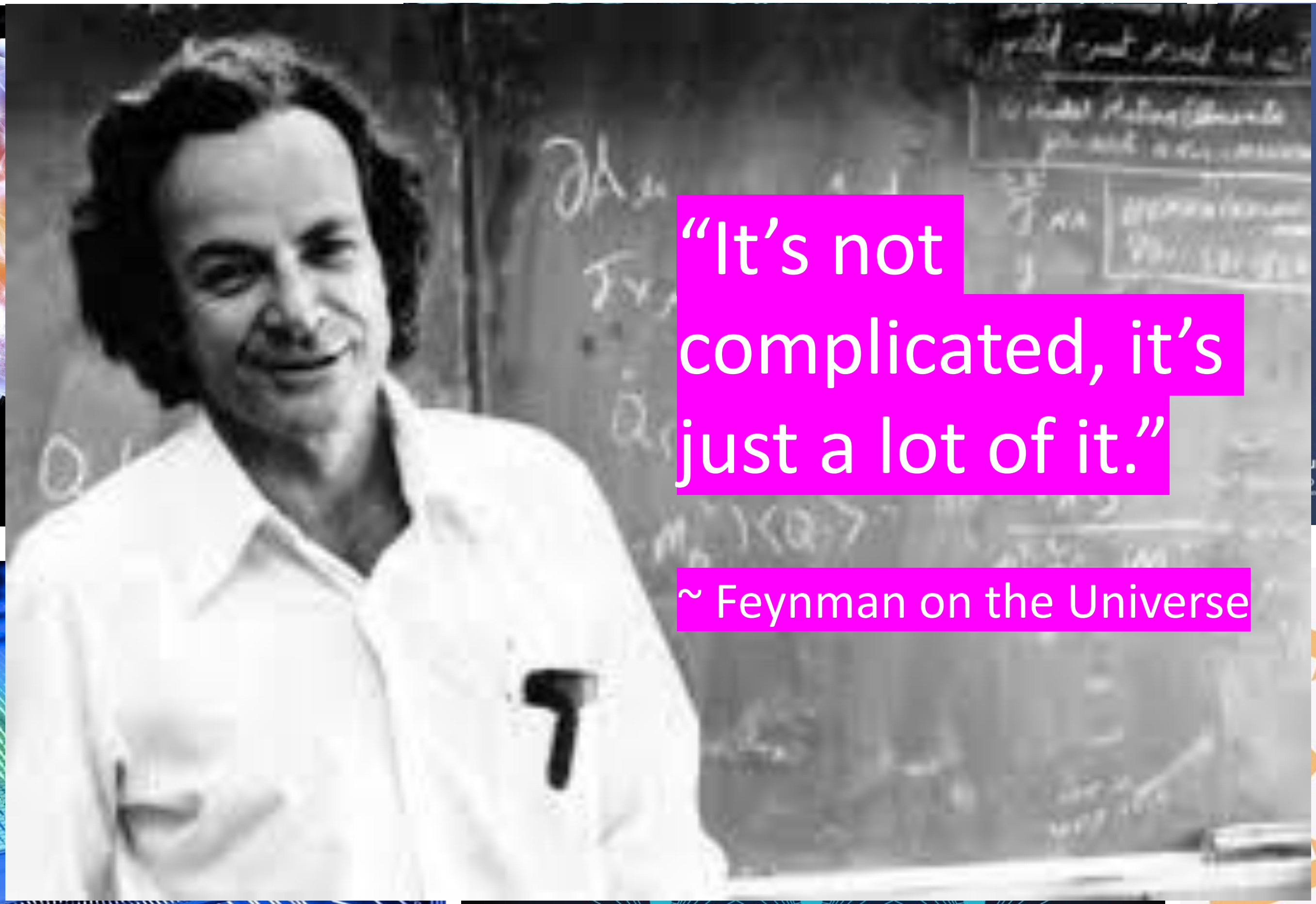




The Power of LOTS of (REPEATED) SIMPLE UNITS with SIMPLE RULES

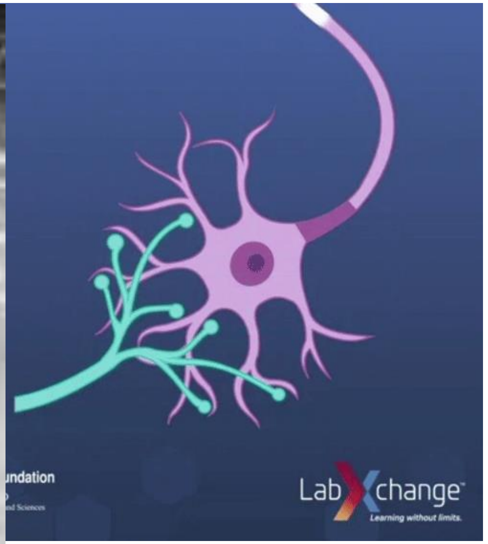
(with localized “gating” of information…)

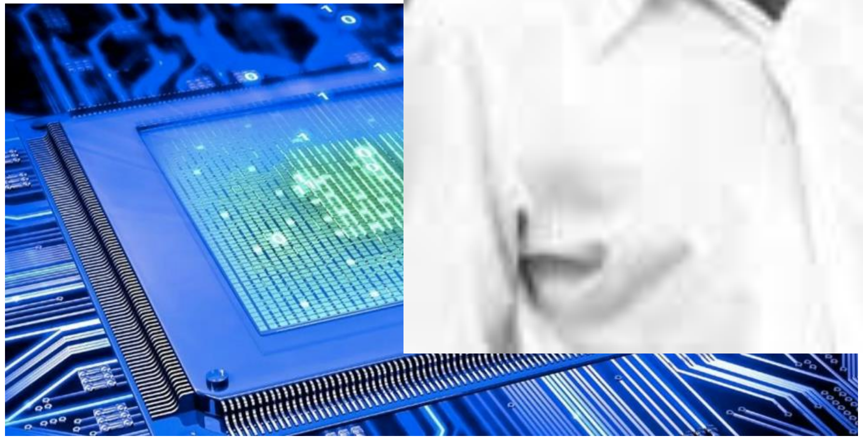


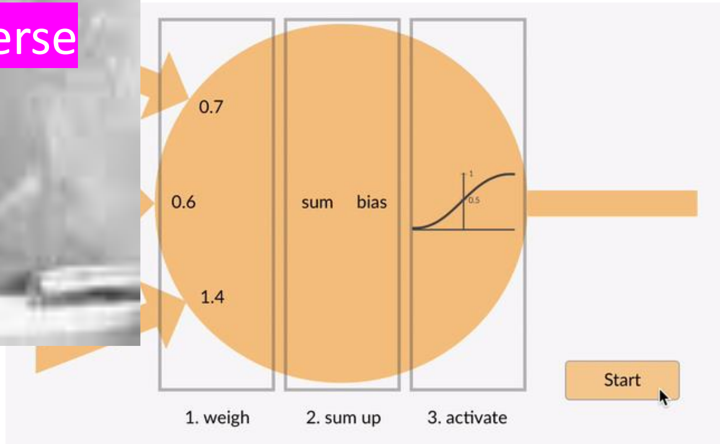


“It’s not complicated, it’s just a lot of it.”

~ Feynman on the Universe







Advantage of Having More Parameters (e.g., Weights) to Play With…

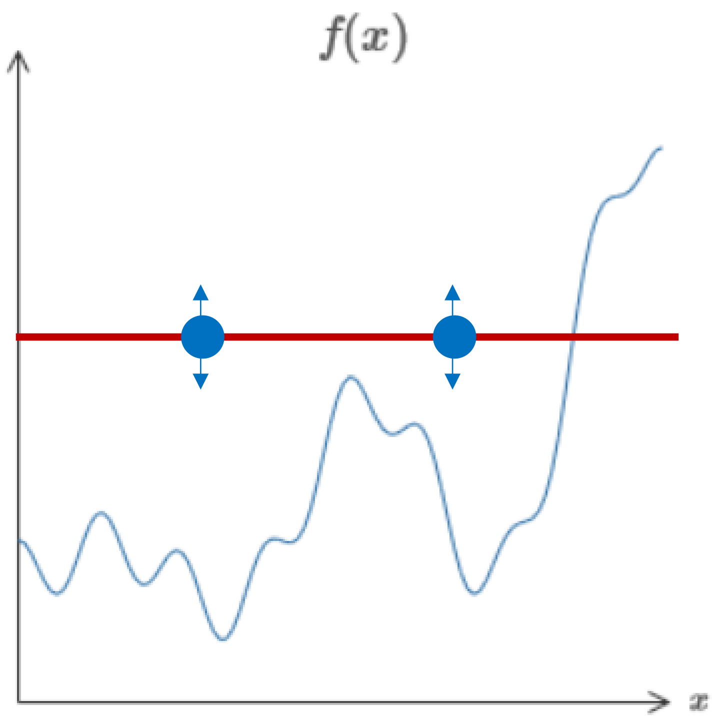
(without exploding computations…)

Advantage of Having More Parameters (e.g., Weights) to Play With…

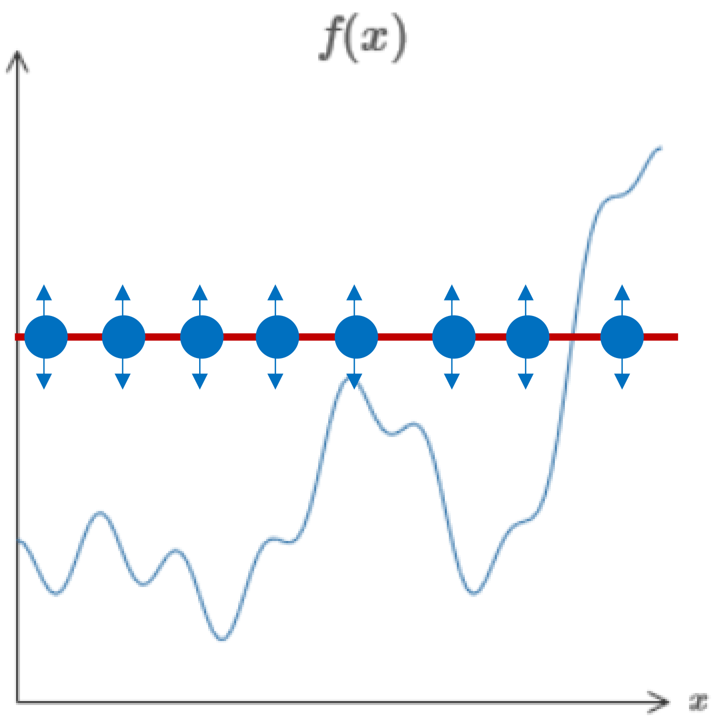
(without exploding computations…)

Advantage of Having More Parameters (e.g., Weights) to Play With…

(without exploding computations…)

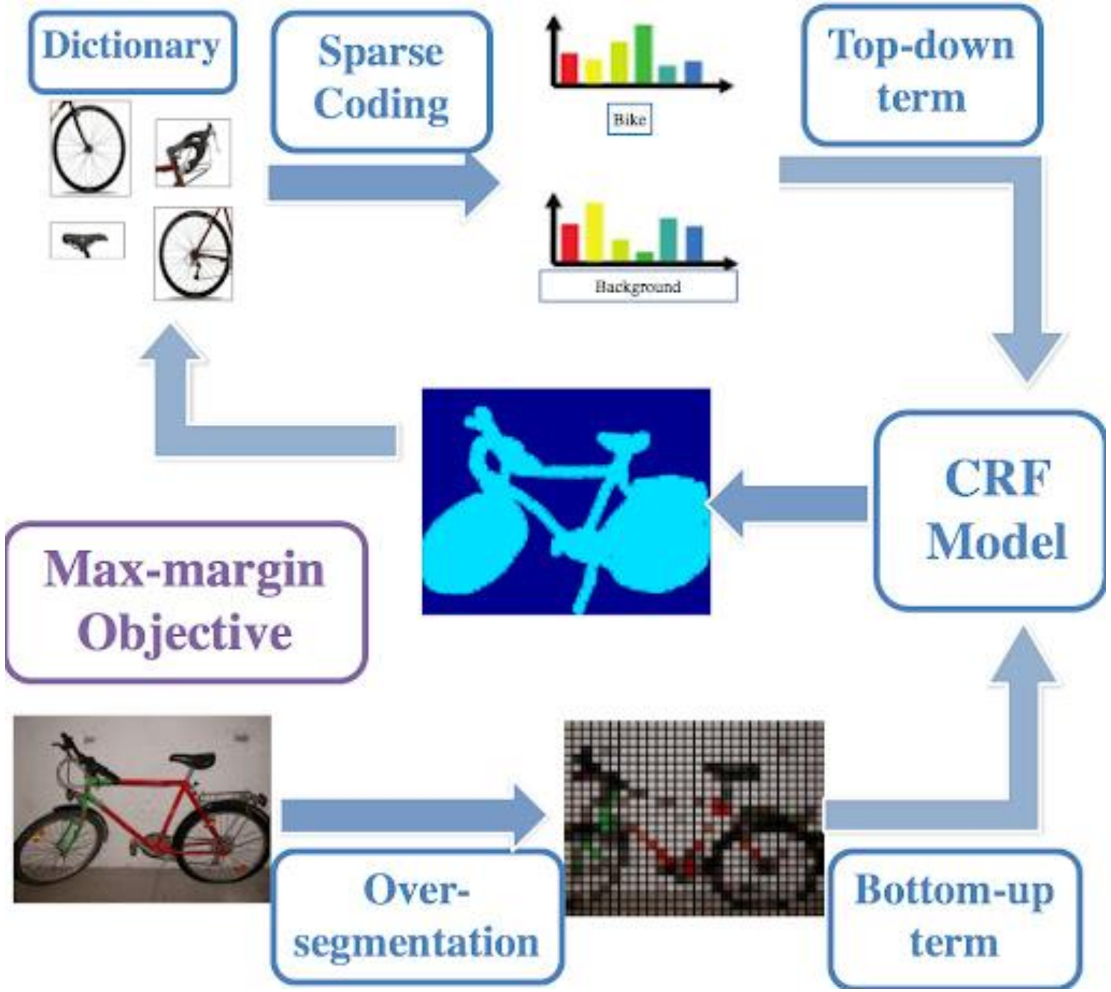


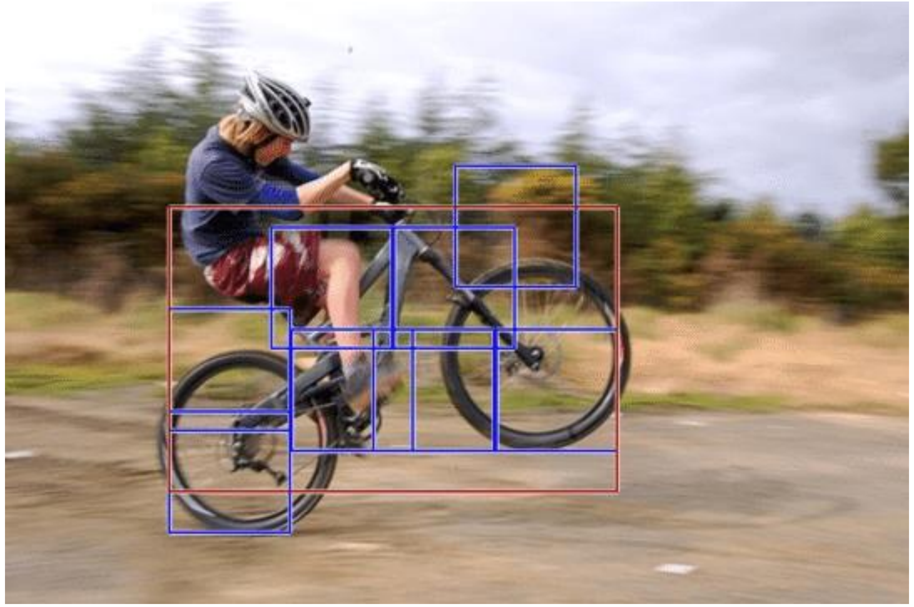
Vs.



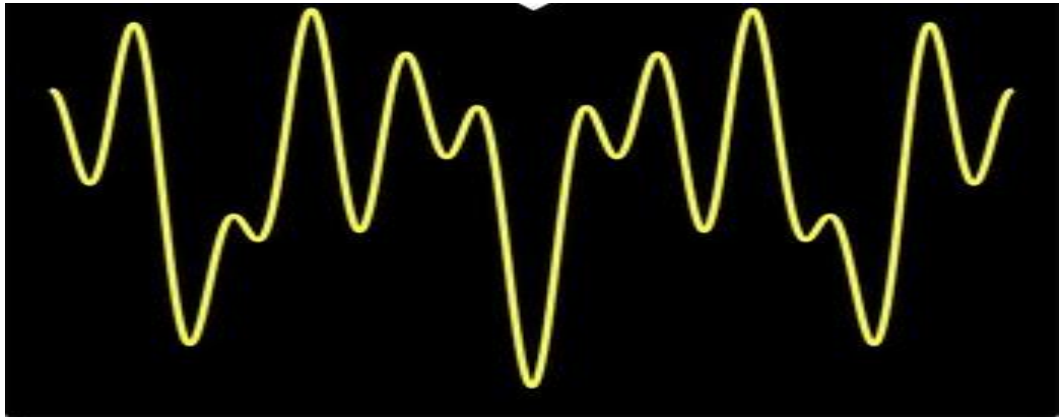
The Power of Bases (e.g., Features) and Their <u>Weighted</u> Combinations…

The Power of Bases (e.g., Features) and Their <u>Weighted</u> Combinations…

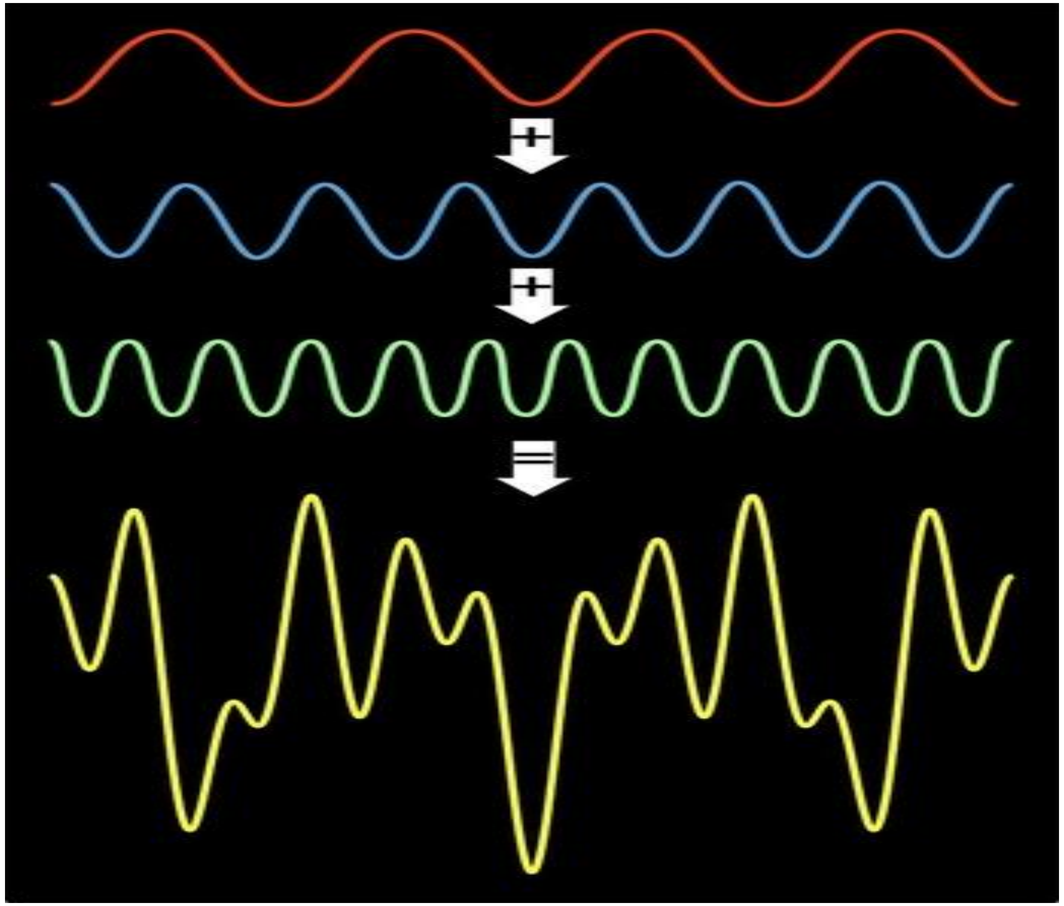




The Power of Bases (e.g., Features) and Their <u>Weighted</u> Combinations…

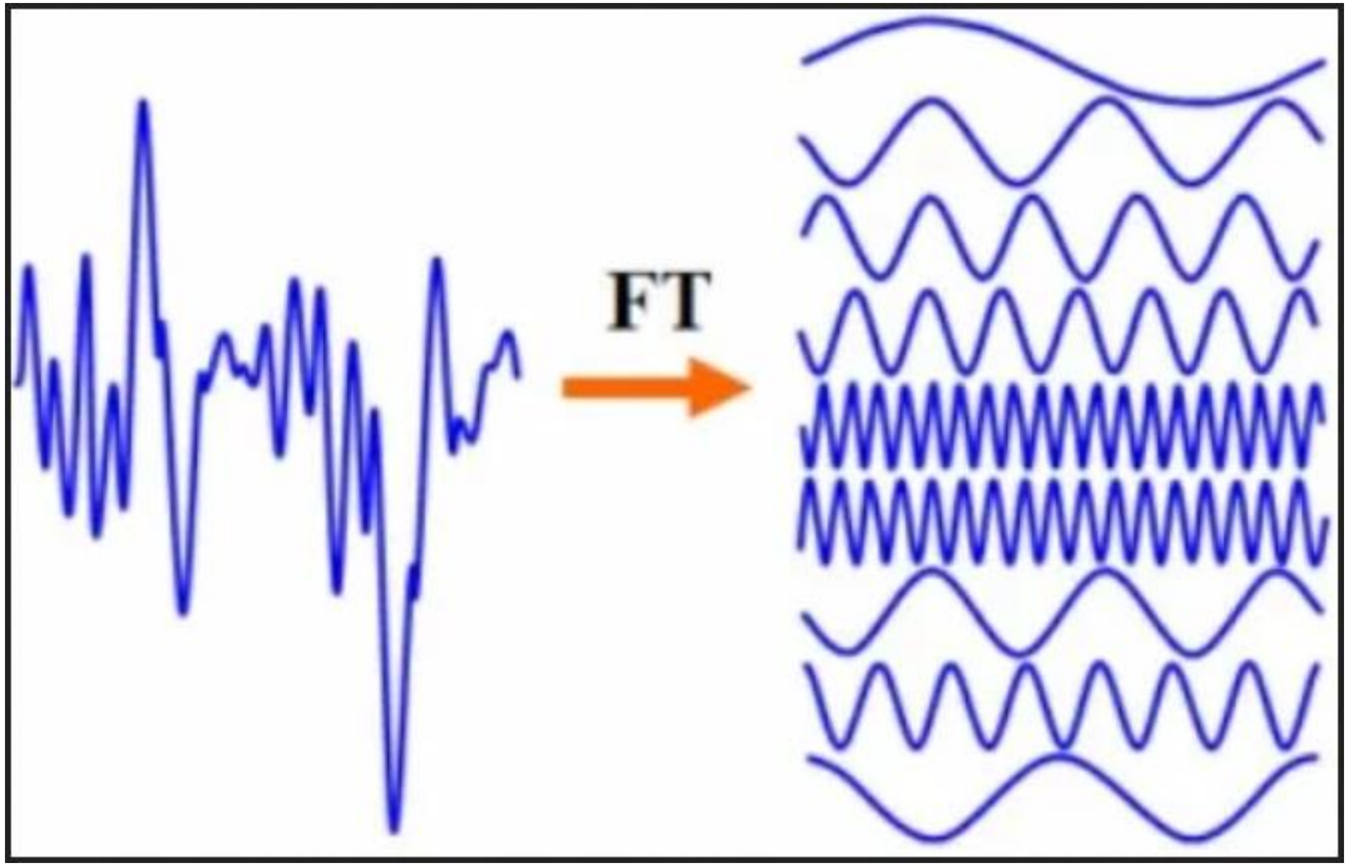


The Power of Bases (e.g., Features) and Their Weighted Combinations…

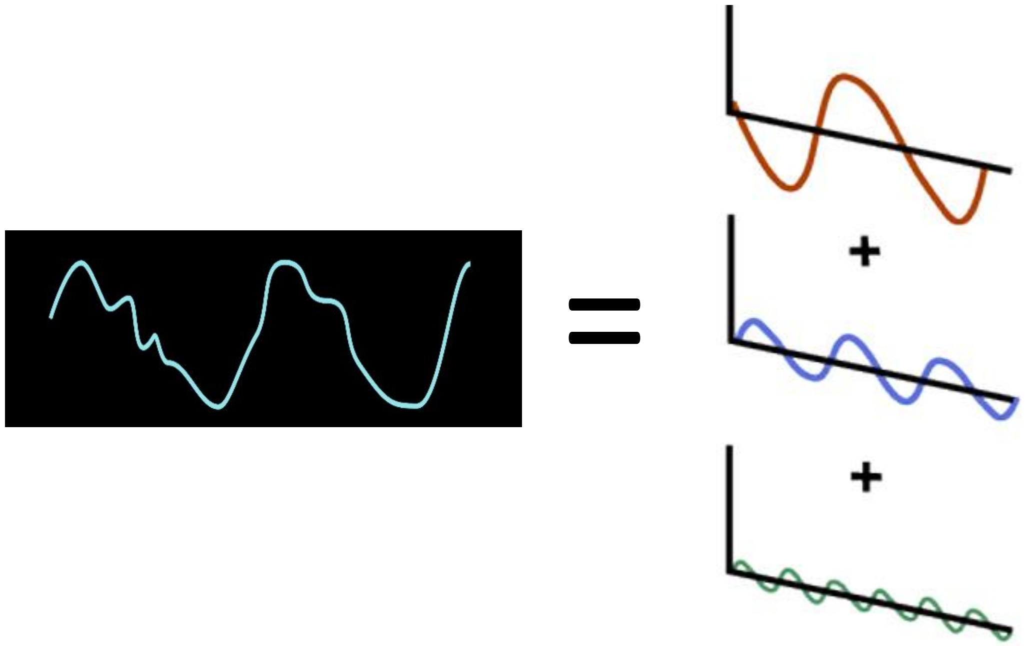


The Power of Basis (e.g., Features) and Their Weighted Combinations…

Q. Can we write functions as weighted sums of sinusoids (features)?

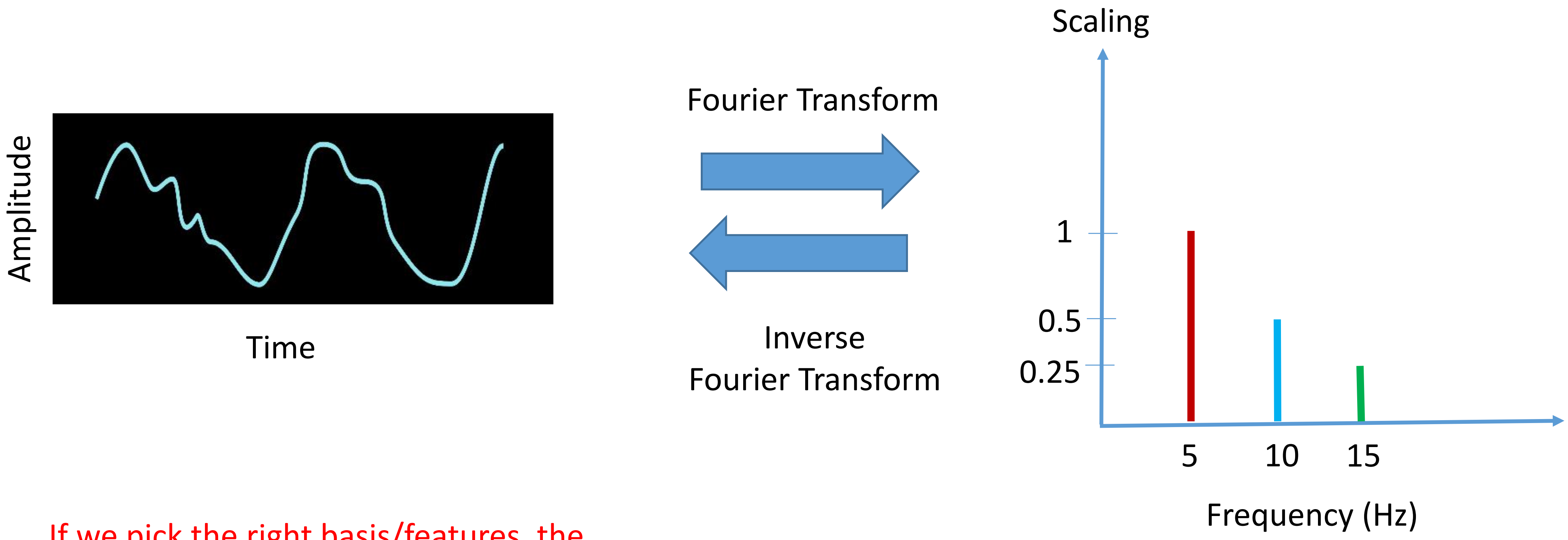


The Power of Basis (e.g., Features) and Their Weighted Combinations…



Ingredient (sinusoid frequency)	Amount (scaling)	Process
f_1	1	**Add all**
f_2	0.5	
f_3	0.25	

The Power of Basis (e.g., Features) and Their <u>Weighted</u> Combinations…



If we pick the right basis/features, the problem could become very sparse!

Now to Our Second Question…

- In our learning of mathematics, have we seen "mathematics that learns" before?

Roots of a Polynomial…

Analytical
(Direct Solution)

Solve $x^3 + 4x + 8 = -2x^2$ analytically (symbolically).

$x^3 + 2x^2 + 4x + 8 = 0$ Standard form

$x^2(x+2) + 4(x+2) = 0$ Factor by grouping

$(x^2 + 4)(x+2) = 0$

$x^2 + 4 = 0 \; or \; x + 2 = 0$ zero product property

$x^2 = -4 \; or \; x = -2$

$x = \pm 2i \; or \; x = -2$

So if we can factor into linear and quadratic factors, we can find the exact values of all real and complex roots.

Roots of a Polynomial…

Analytical (Direct Solution) Vs. **Algorithmic (Iterative/"Learning" Solution)**

Solve $x^3 + 4x + 8 = -2x^2$ analytically (symbolically).

$x^3 + 2x^2 + 4x + 8 = 0$ Standard form

$x^2(x+2) + 4(x+2) = 0$ Factor by grouping

$(x^2+4)(x+2) = 0$

$x^2 + 4 = 0 \text{ or } x + 2 = 0$ zero product property

$x^2 = -4 \text{ or } x = -2$

$x = \pm 2i \text{ or } x = -2$

So if we can factor into linear and quadratic factors, we can find the exact values of all real and complex roots.

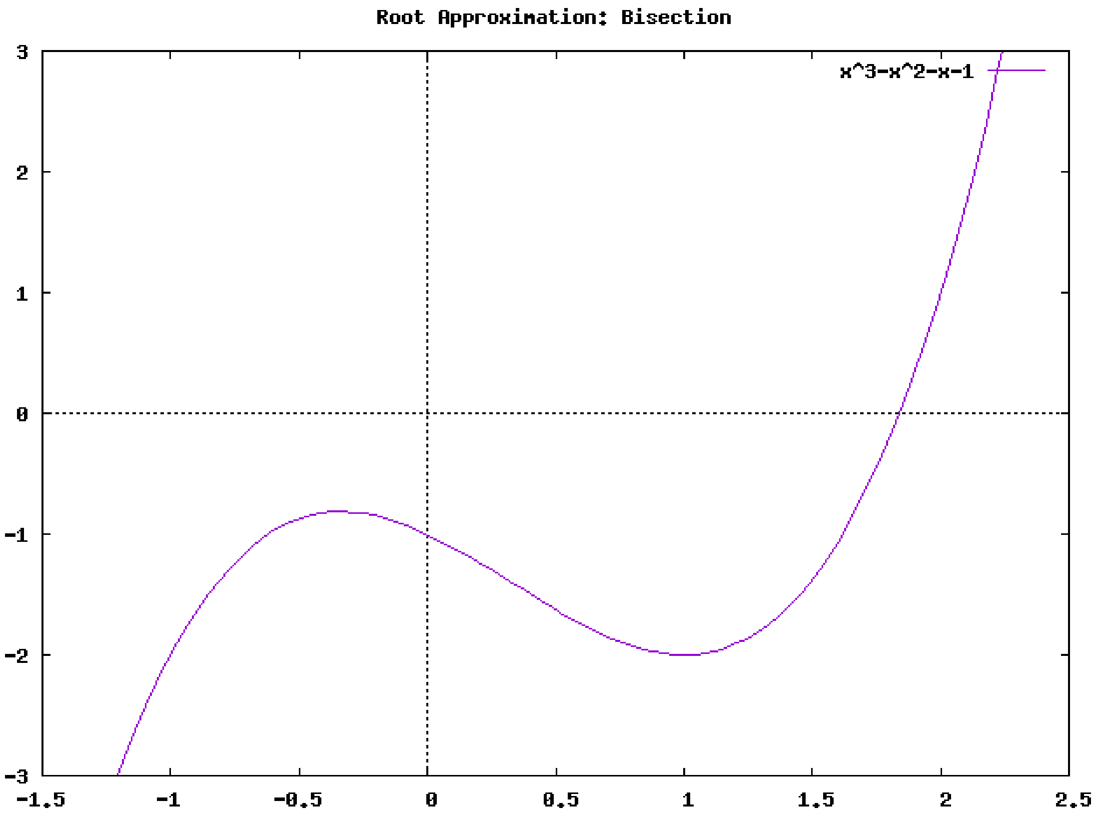
Calculate a and b

1. **Procedure** bisection method (a,b,ϵ)
2. $c = \frac{a+b}{2}$
3. Compute derivative of $f(x)$ denoted as $\acute{f}(x)$
4. **While** $|a-b| \geq \epsilon$ and $\acute{f}(x) \neq 0$ **do**
5. **If** $\acute{f}(a) \times \acute{f}(c) < 0$ **then**
6. $b \leftarrow c$
7. **Else**
8. $a \leftarrow c$
9. $c \leftarrow \frac{a+b}{2}$
10. **Return** a or b or c

Negative value indicates that a and c are on opposite sides of the root.

Which one is easier to teach a machine? (hint: the one with simple repeated steps).

Roots of a Polynomial…



Algorithmic
(Iterative/"Learning" Solution)

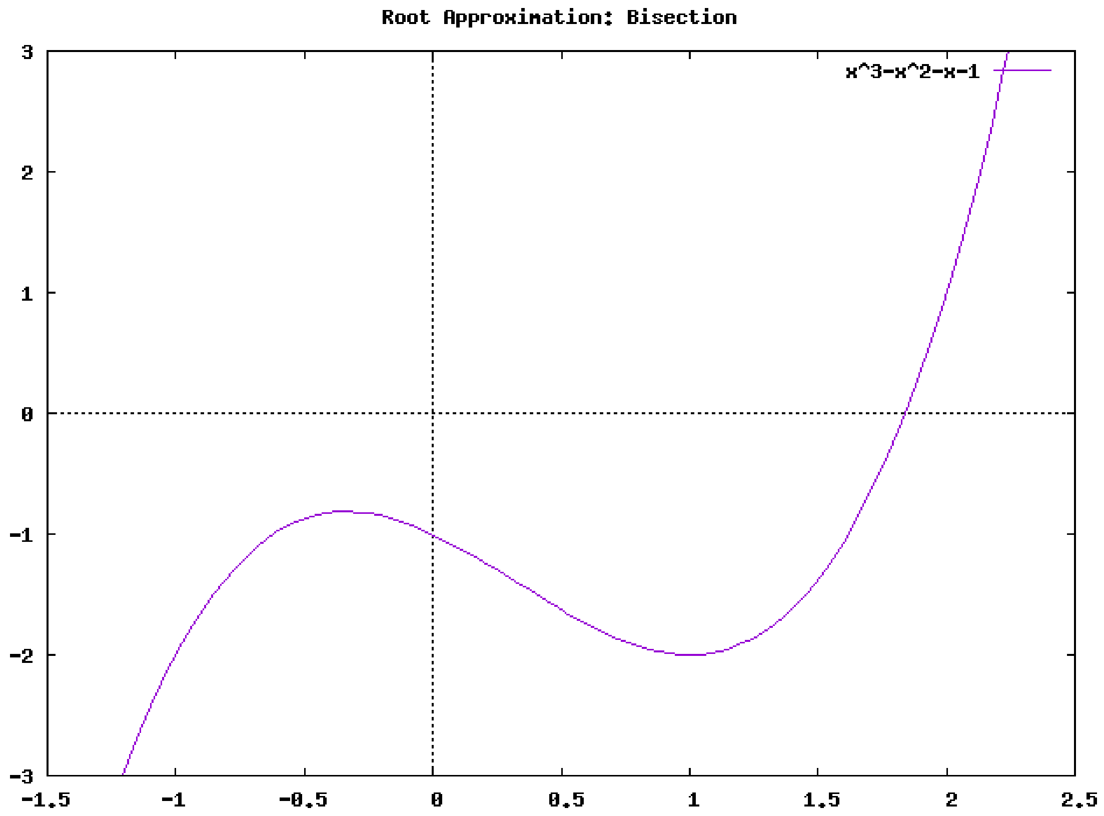
Calculate a and b

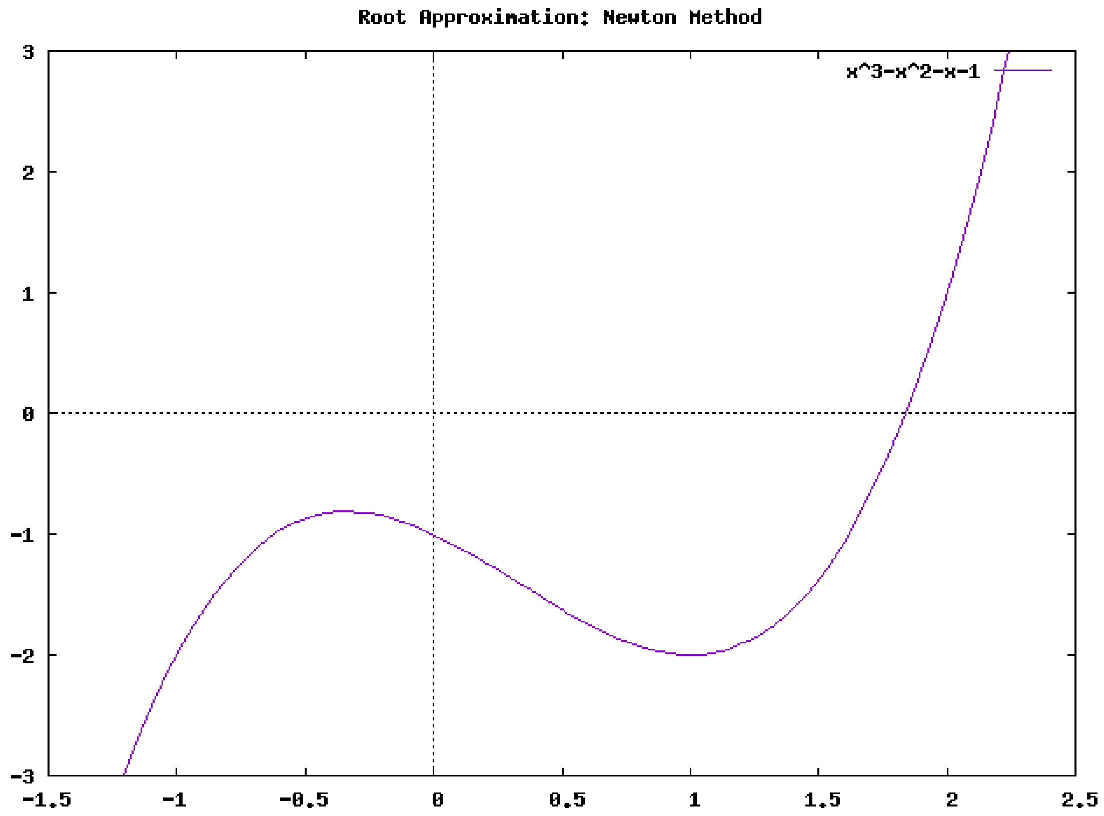
Procedure bisection method (a,b,ϵ)
$c = \frac{a+b}{2}$
Compute derivative of $f(x)$ denoted as $\acute{f}(x)$
While $|a - b| \geq \epsilon$ and $\acute{f}(x) \neq 0$ **do**
If $\acute{f}(a) \times \acute{f}(c) < 0$ **then**
$b \leftarrow c$
Else
$a \leftarrow c$
$c \leftarrow \frac{a+b}{2}$
Return a or b or c

Negative value indicates that a and c are on opposite sides of the root.

Roots of a Polynomial…

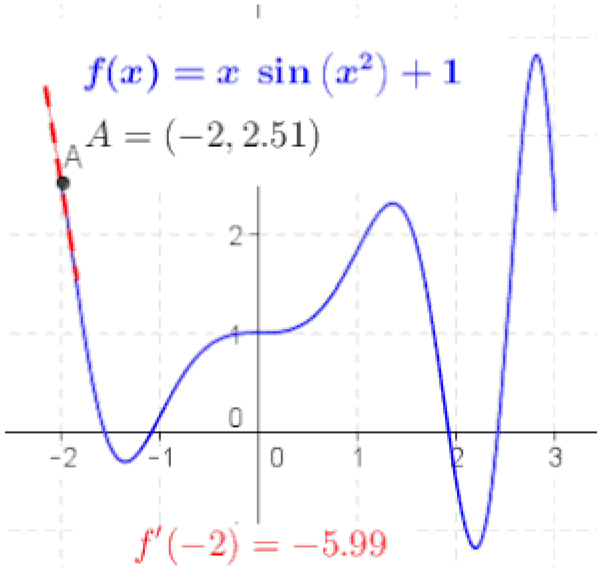
…even smarter way would be to use knowledge of function local behavior (e.g., gradient) to plan your next move!





Finding Optima...

Analytical
(Direct Solution)



Find derivative, and solve for x

$$f'(x) = \sin(x^2) + 2x^2 \cos(x^2) = 0$$

Finding Optima...

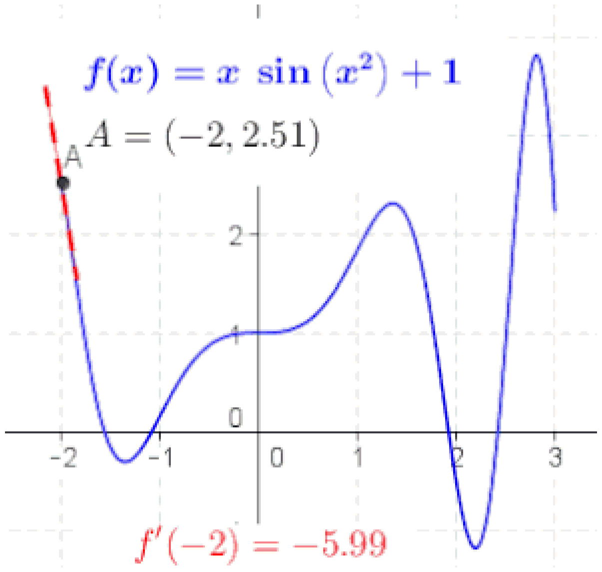
Analytical
(Direct Solution)

$$f(x,y) = -x^4 + 4(x^2 - y^2) - 3$$

Find gradient

$$\nabla f = \begin{bmatrix} \frac{\partial}{\partial x}(-x^4 + 4(x^2 - y^2) - 3) \\ \\ \frac{\partial}{\partial y}(-x^4 + 4(x^2 - y^2) - 3) \end{bmatrix} = \begin{bmatrix} -4x^3 + 8x \\ -8y \end{bmatrix}$$

Solve system of equations $\nabla \mathbf{f} = \mathbf{0}$ for x and y

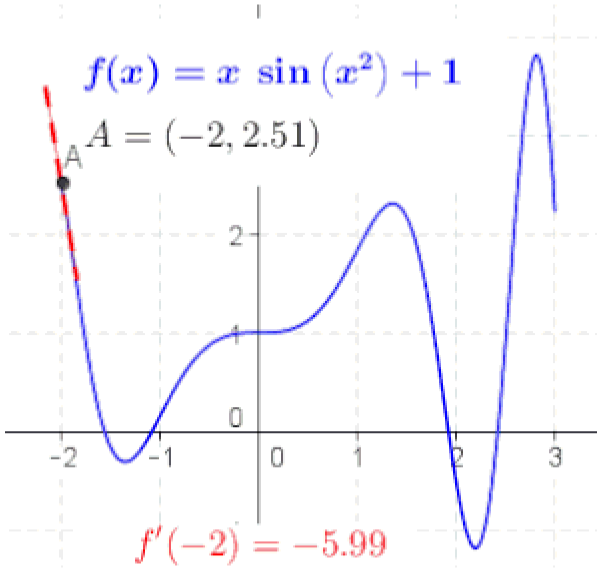


Find derivative, and solve for x

$$f'(x) = \sin(x^2) + 2x^2 \cos(x^2) = 0$$

Finding Optima...

Analytical
(Direct Solution)



Find derivative, and solve for x

$$f'(x) = \sin(x^2) + 2x^2\cos(x^2) = 0$$

$$f(x,y) = -x^4 + 4(x^2 - y^2) - 3$$

Find gradient

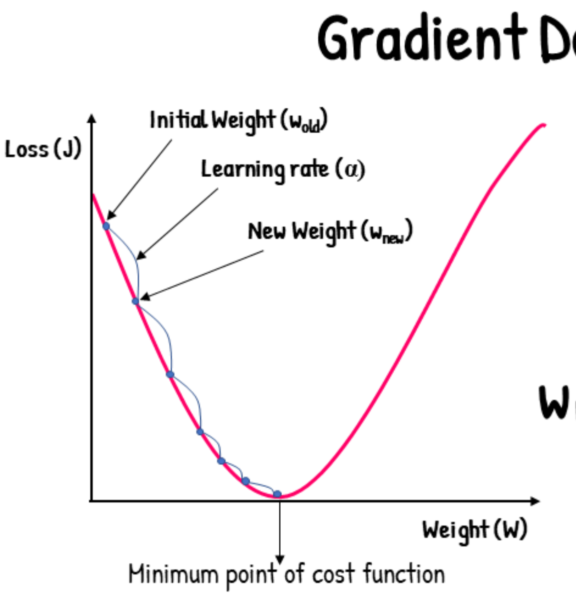
$$\nabla f = \begin{bmatrix} \frac{\partial}{\partial x}(-x^4 + 4(x^2 - y^2) - 3) \\ \\ \frac{\partial}{\partial y}(-x^4 + 4(x^2 - y^2) - 3) \end{bmatrix} = \begin{bmatrix} -4x^3 + 8x \\ -8y \end{bmatrix}$$

Solve system of equations $\nabla \mathbf{f} = \mathbf{0}$ for x and y

Imagine having to do so for a function of thousands or millions of variables!

Finding Optima…

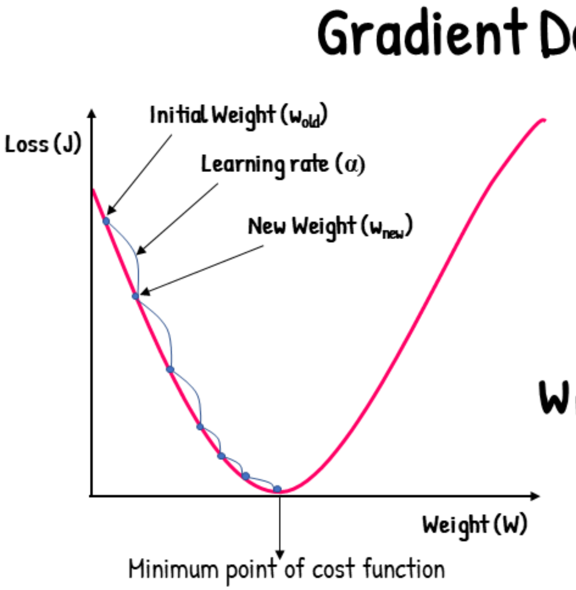
Algorithmic
(Iterative/"Learning" Solution)



$$w_{new} = w_{old} - \alpha \frac{\delta J}{\delta w}$$

Finding Optima…

Algorithmic
(Iterative/"Learning" Solution)



$$w_{new} = w_{old} - \alpha \frac{\delta J}{\delta w}$$

Algorithm 2: Gradient Descent

input : $f : \mathbb{R}^n \to \mathbb{R}$ a differentiable function
$\mathbf{x}^{(0)}$ an initial solution

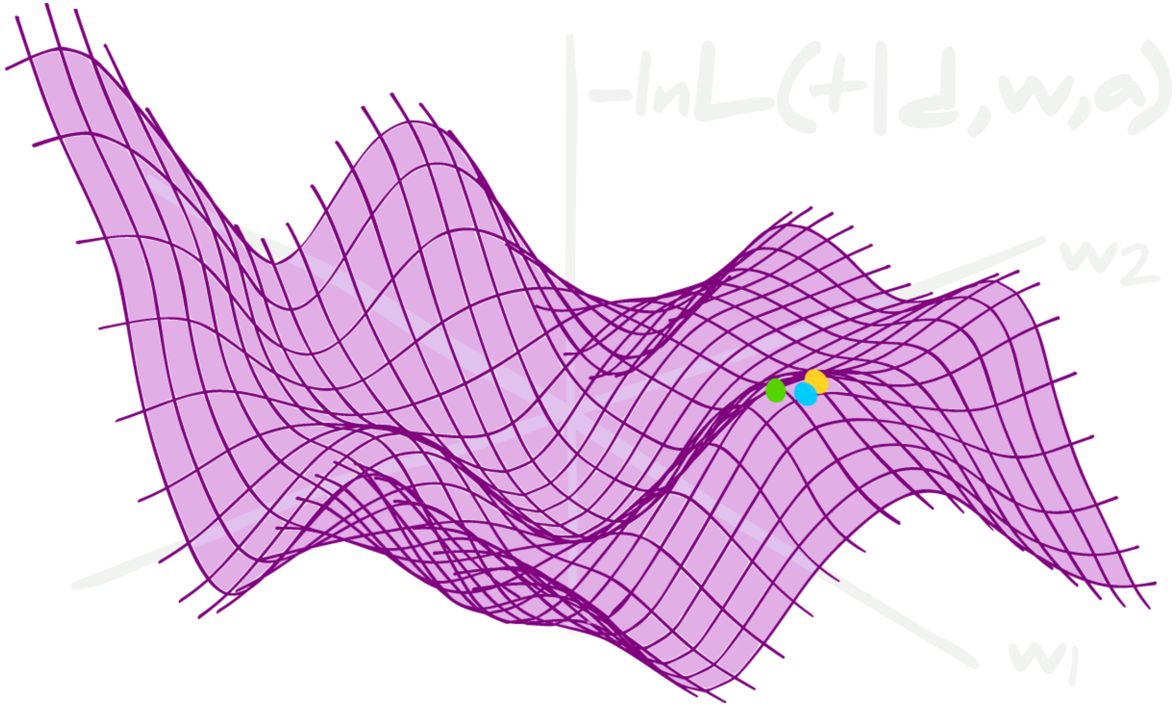
output: $\mathbf{x}^{\star}$, a local minimum of the cost function f.

begin
  $k \leftarrow 0$;
  while STOP-CRIT **and** $(k < k_{max})$ **do**
    $\mathbf{x}^{(k+1)} \leftarrow \mathbf{x}^{(k)} - \alpha^{(k)} \nabla f(\mathbf{x})$;
    with $\alpha^{(k)} = \arg\min_{\alpha \in \mathbb{R}_+} f\left(\mathbf{x}^{(k)} - \alpha \nabla f(\mathbf{x})\right)$;
    $k \leftarrow k + 1$;
  return $\mathbf{x}^{(k)}$
end

Also approximated numerically!

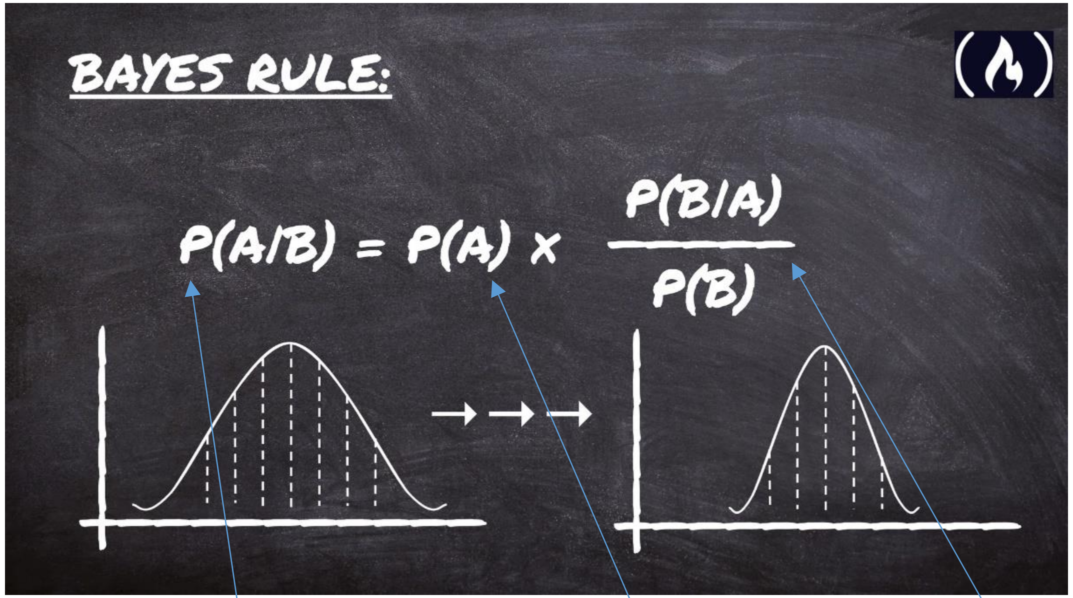
Finding Optima...

Algorithmic
(Iterative/"Learning" Solution)



We could try multiple initializations to avoid local minima.

Guess Who…? “Learning” Probabilities

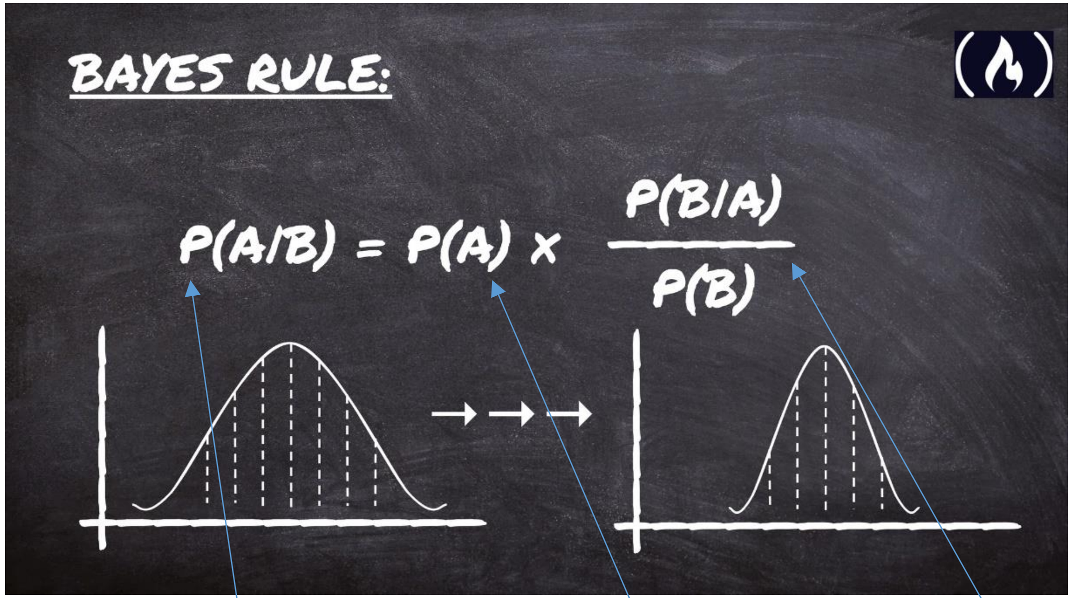


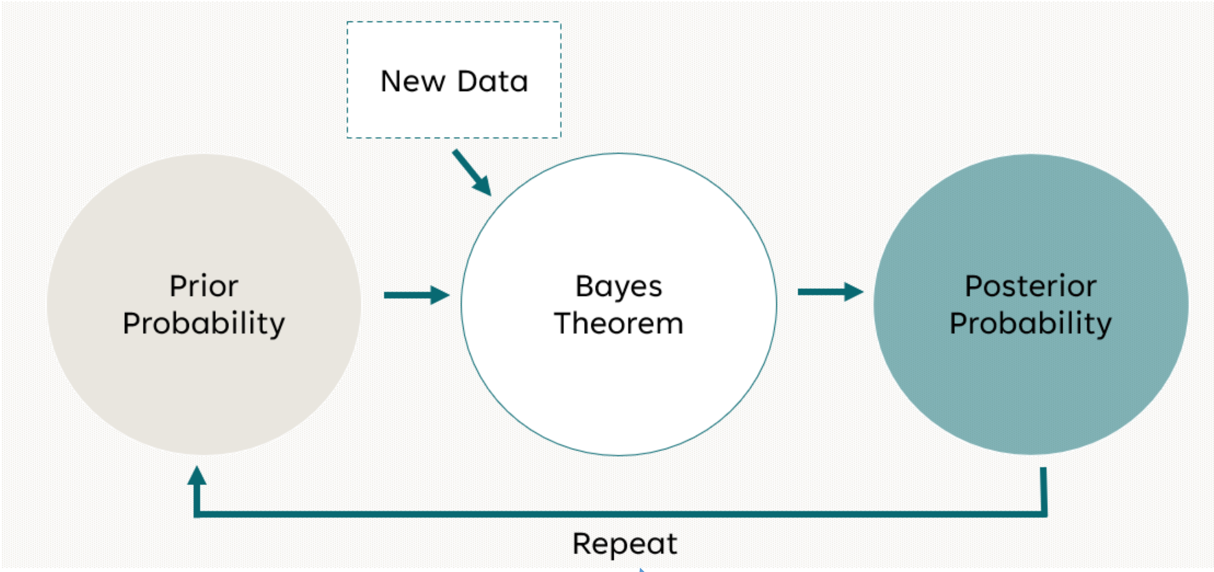
Learned/Updated probability of event given new data/information

Initial probability of an event (assumed or based on past data)

Statistics of new data and its statistical relevance to event of interest

Guess Who...? "Learning" Probabilities





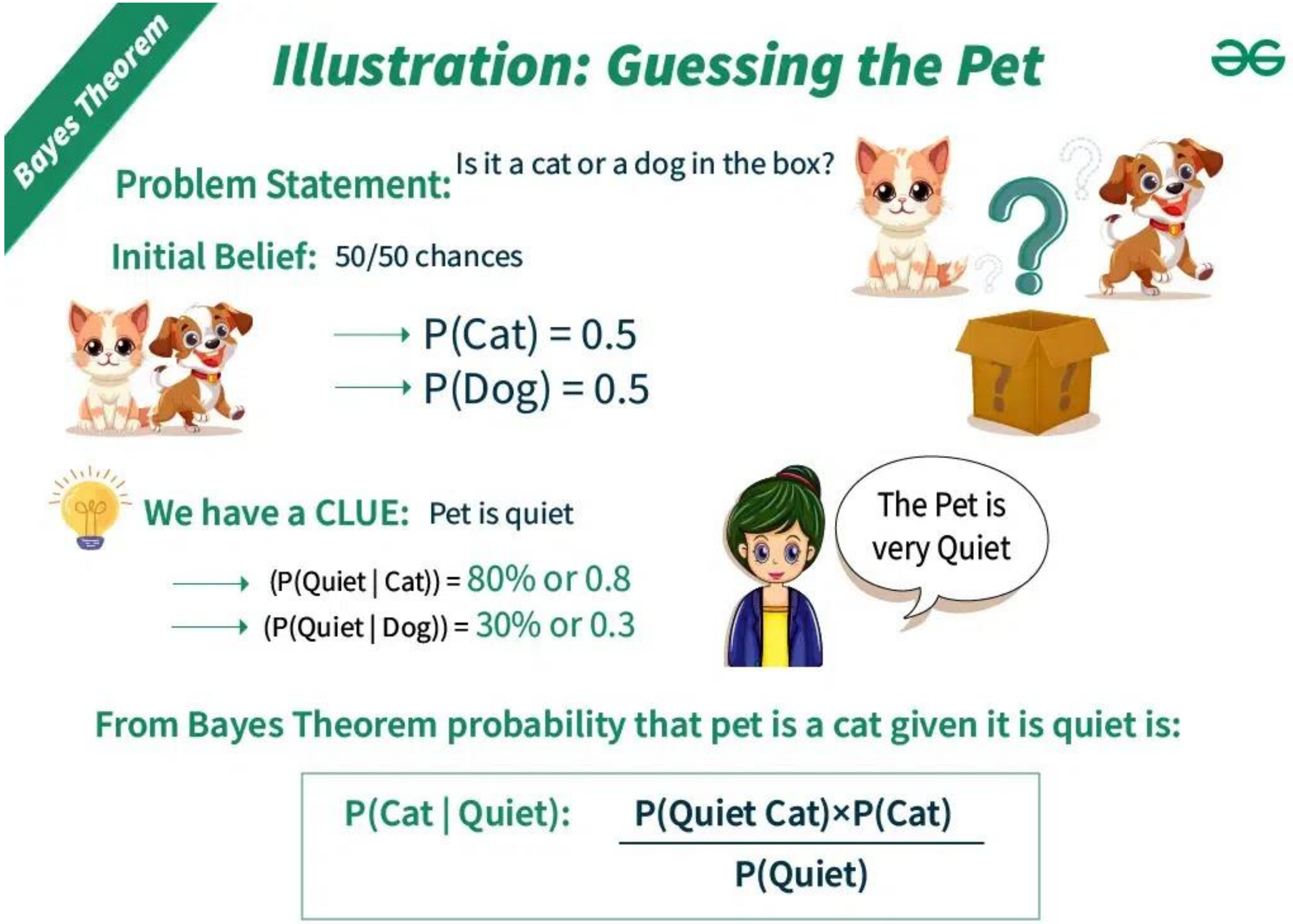
Learned/Updated probability of event given new data/information

Initial probability of an event (assumed or based on past data)

Statistics of new data and its statistical relevance to event of interest

Iterative "learning" as new data arrives

Guess Who…? “Learning” Probabilities



The Good, the Bad, and the Ugly…

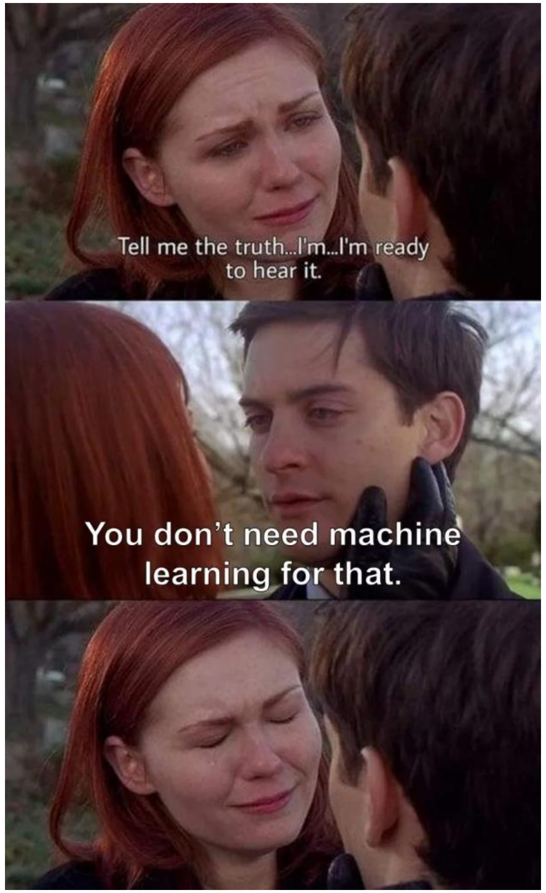
ML Concerns, Limitations, and Open Problems

The Good, the Bad, and the Ugly…

ML Concerns, Limitations, and Open Problems

- Data Acquisition
- Heavy Reliance on Data Quality
- Investment of Time & Resources
- Difficulties in Interpretation
- High Susceptibility to Errors & Biases
- Concerns of Data Privacy
- Ethical Concerns

1. They Can be an Overkill…



2. Data Hungry, Hardware Hungry, Power Hungry…

FEEDING THE AI BEAST —

Power-hungry AI is putting the hurt on global electricity supply

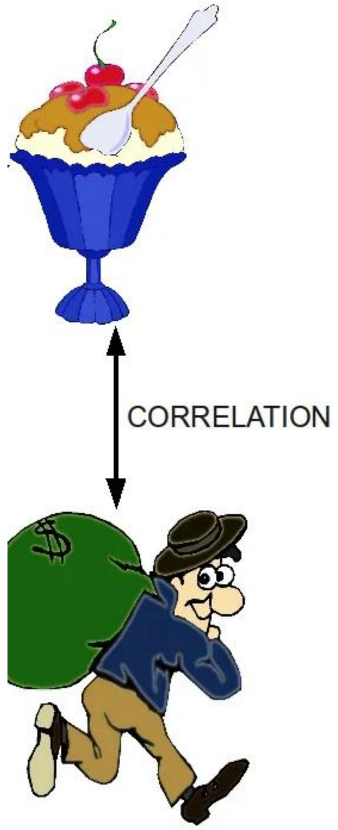
Data centers are becoming a bottleneck for AI development.

CAMILLA HODGSON, FINANCIAL TIMES - 4/17/2024, 6:55 PM

Even on small organizational scale, reliable training requires good deal of reliable data.

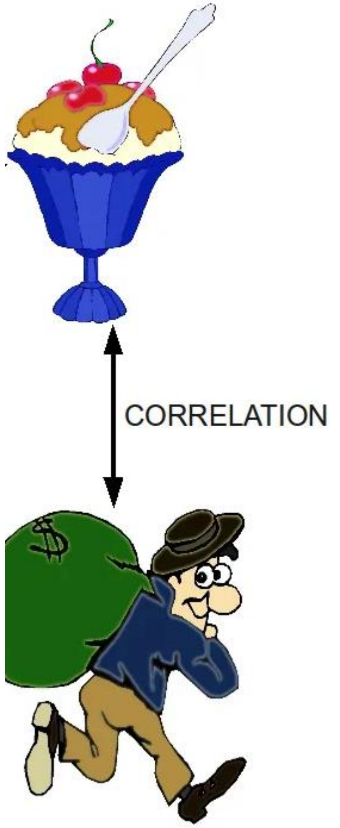
3. Do Not Capture Causal Relations…

ML generally works on statistical relations.



3. Do Not Capture Causal Relations…

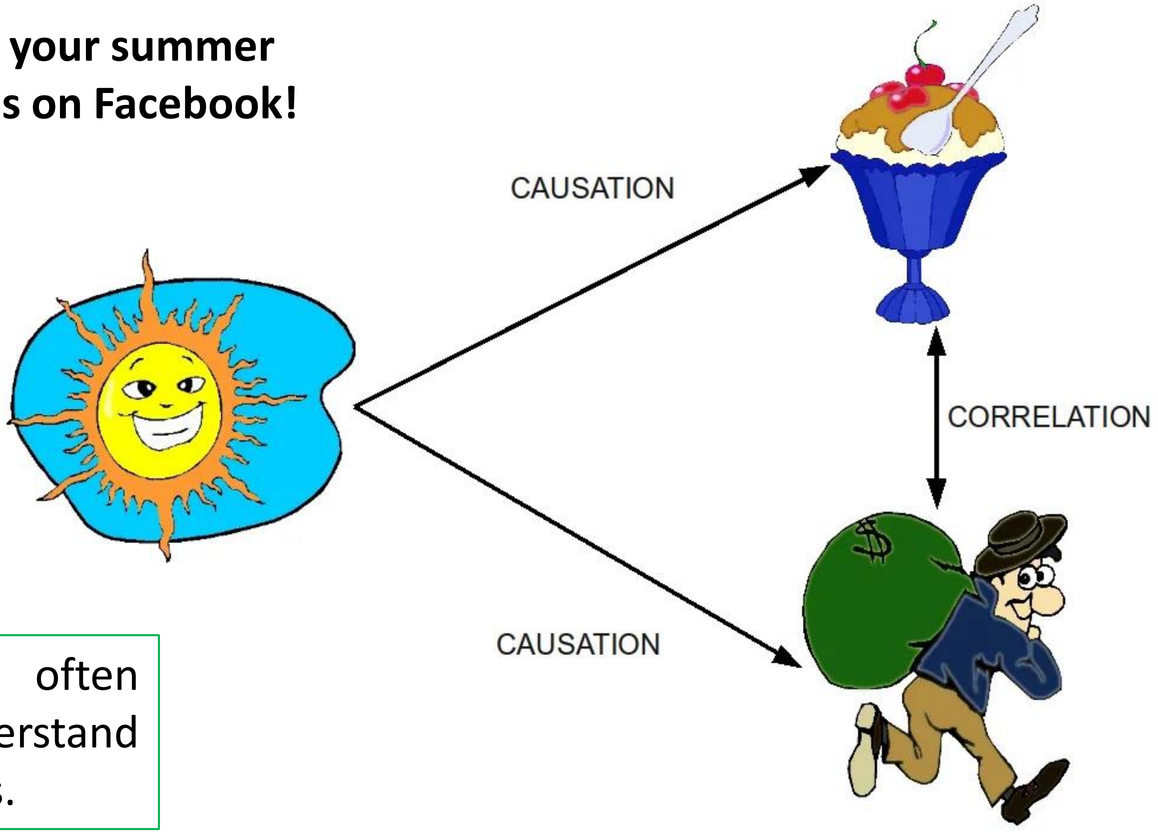
ML generally works on statistical relations.



Stop eating Ice-cream!

3. Do Not Capture Causal Relations…

Don’t post your summer travel plans on Facebook!



But causal relations often needed to truly understand and work on problems.

4. Bias, Reproducibility, and Verifiability…

Hard to identify ethical biases in training data and in results provided by a big ML algorithm.

4. Bias, Reproducibility, and Verifiability…

Hard to identify ethical biases in training data and in results provided by a big ML algorithm.

Example: Biased "unchallengeable" decisions could worsen economic disparity.

4. Bias, Reproducibility, and Verifiability…

Hard to identify ethical biases in training data and in results provided by a big ML algorithm.

A Survey on Bias and Fairness in Machine Learning

NINAREH MEHRABI, FRED MORSTATTER, NRIPSUTA SAXENA, KRISTINA LERMAN, and ARAM GALSTYAN, USC-ISI

Reuters

Insight - Amazon scraps secret AI recruiting tool that showed bias against women

- ML was trained to find "good CVs" using CVs of highly successful people in Silicon Valley (which are mostly men – for reasons other than competence).
- It started rejecting CVs with "feminine" language.

4. Bias, Reproducibility, and Verifiability…

Hard to verify and reproduce operation of large models.

nature

Explore content ⌄ About the journal ⌄ Publish with us ⌄ Subscribe

nature › news feature › article

NEWS FEATURE | 05 December 2023

Is AI leading to a reproducibility crisis in science?

Scientists worry that ill-informed use of artificial intelligence is driving a deluge of unreliable or useless research.

4. Bias, Reproducibility, and Verifiability…

Hard to verify and reproduce operation of large models.

Why bother?

- Facilitates collaboration and review processes
- Ensures continuity of work and knowledge exchange retains
- Provides opportunity for future evaluations
- Verification of results for hidden biases can help reduce them

nature

Explore content ⌄ About the journal ⌄ Publish with us ⌄ Subscribe

nature › news feature › article

NEWS FEATURE | 05 December 2023

Is AI leading to a reproducibility crisis in science?

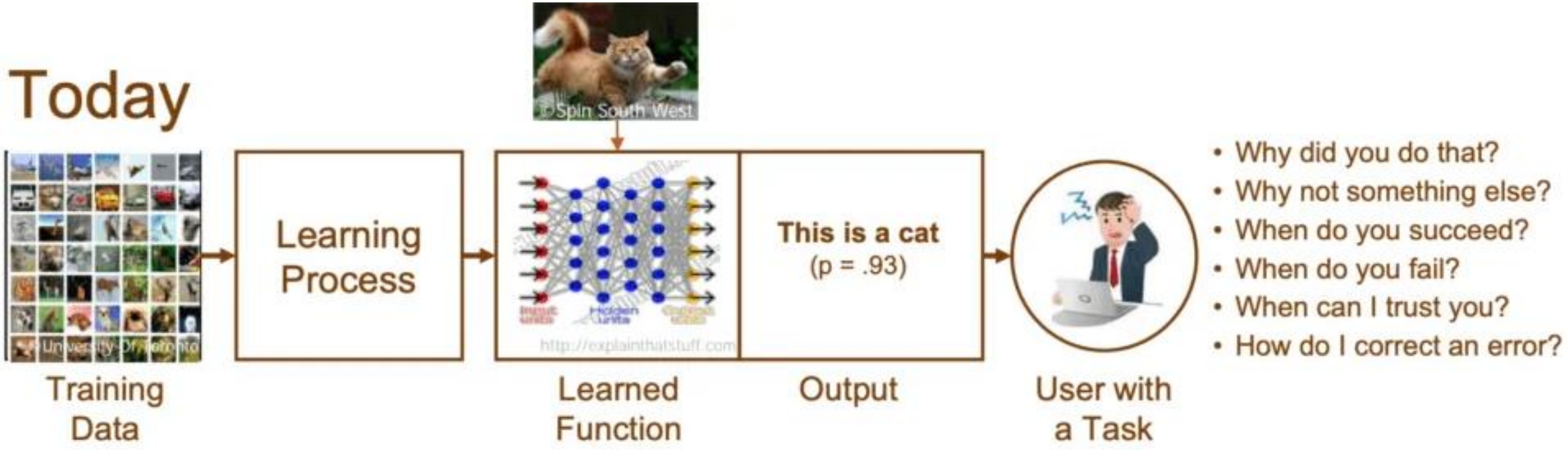
Scientists worry that ill-informed use of artificial intelligence is driving a deluge of unreliable or useless research.

5. Black Box – Explainability Issues…

ML learned “features” are not always aligned with what we consider features.

5. Black Box – Explainability Issues…

ML learned "features" are not always aligned with what we consider features.

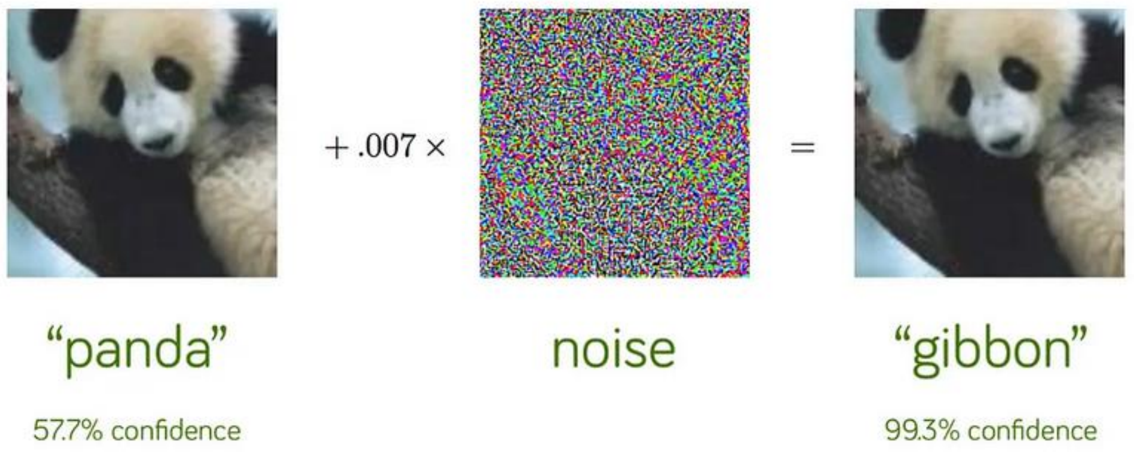


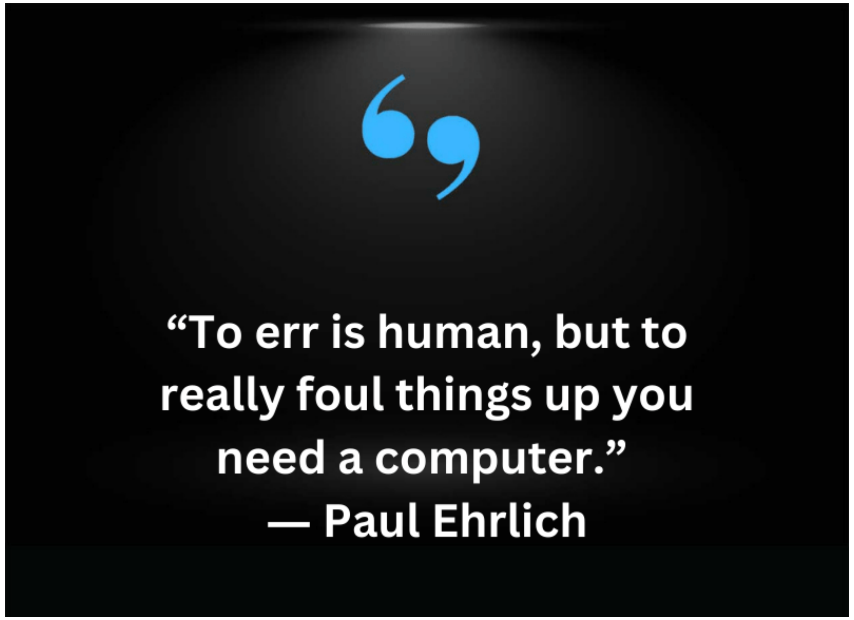
6. Vulnerabilities

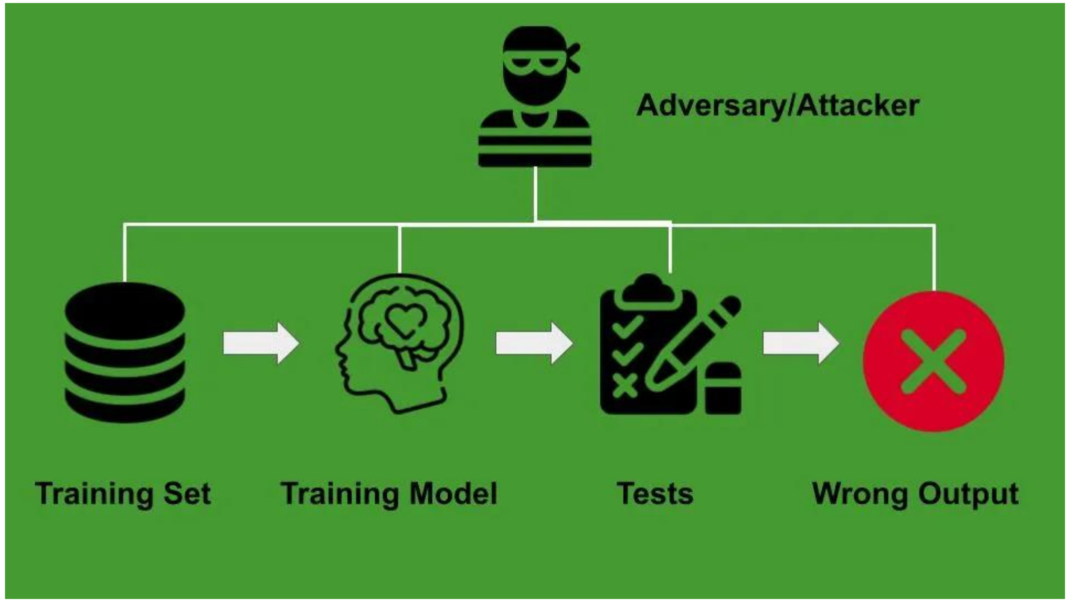
Deep ML, in particular, is prone to adversarial attacks and errors in data.

6. Vulnerabilities

Deep ML, in particular, is prone to adversarial attacks and errors in data.







7. [Some] Open Problems

- Why does machine learning (particularly, deep learning) work so well?
  - We saw some recent hypotheses. Can you add more or prove these?

7. [Some] Open Problems

- Why does machine learning (particularly, deep learning) work so well?
  - We saw some recent hypotheses. Can you add more or prove these?
- How to make the learning process more efficient?
  - Work with lesser data, smaller model (pruning), fewer hyper-parameters

7. [Some] Open Problems

- Why does machine learning (particularly, deep learning) work so well?
  - We saw some recent hypotheses. Can you add more or prove these?
- How to make the learning process more efficient?
  - Work with lesser data, smaller model (pruning), fewer hyper-parameters
- How can we include learning of causal relations?
  - New field: Causal ML

7. [Some] Open Problems

- Why does machine learning (particularly, deep learning) work so well?
  - We saw some recent hypotheses. Can you add more or prove these?
- How to make the learning process more efficient?
  - Work with lesser data, smaller model (pruning), fewer hyper-parameters
- How can we include learning of causal relations?
  - New field: Causal ML
- How to best include any known physical laws (PDEs etc.) in the training process?
  - New field: Physics Informed Neural Networks (PINNs)

7. [Some] Open Problems

- How to make ML reproducible, verifiable, and bias-free?

7. [Some] Open Problems

- How to make ML reproducible, verifiable, and bias-free?
- How to make ML more explainable?
  - New field: Explainable AI (XAI)

7. [Some] Open Problems

- How to make ML reproducible, verifiable, and bias-free?
- How to make ML more explainable?
  - New field: Explainable AI (XAI)
- How to reduce vulnerabilities and defend against attacks?

Questions?? Thoughts??