Can a Neural Network Learn the Laws of Physics?

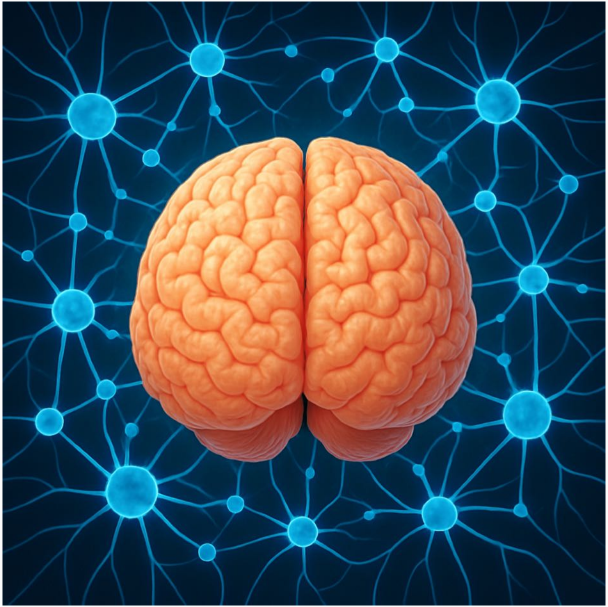
A Swinging Pendulum Example

Alexander Heinlein[1]

Wiskunde D-dag, TU Delft, The Netherlands, June 6, 2025

[1]Delft University of Technology

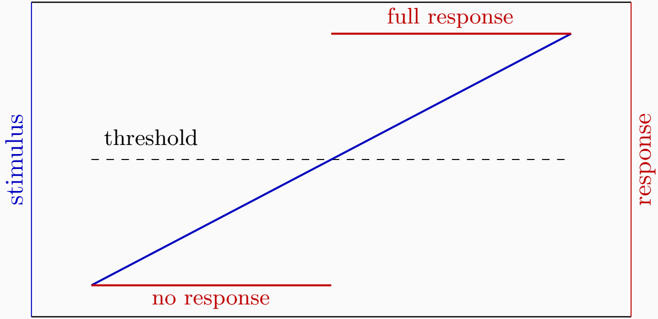
Artificial Intelligence and Neural Networks

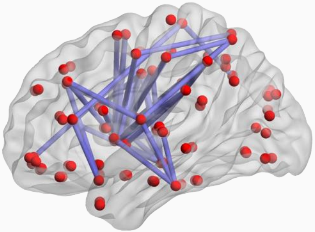


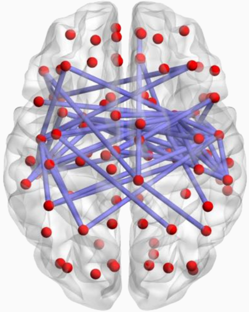
- Modern AI uses **artificial neural networks**, built on mathematics: matrix and vector operations.
- **Hopfield** and **Hinton** won the **2024 Nobel Prize in Physics** for pioneering artificial neural network models; they employ **principles of physics** to motivate the machine learning models.

Artificial Intelligence and Neural Networks

- **Artificial neural networks (ANNs)** or simply **neural networks (NNs)** are machine learning models loosely inspired by the structure of **biological brains**.
- **Biological neurons** respond only when inputs (stimulus) exceed a certain threshold.







Machine Learning

Wikipedia

*"Machine learning (ML) is a field of study in artificial intelligence concerned with the development and study of statistical algorithms that **can learn from data** and generalise to unseen data, and thus perform tasks **without explicit instructions**."*

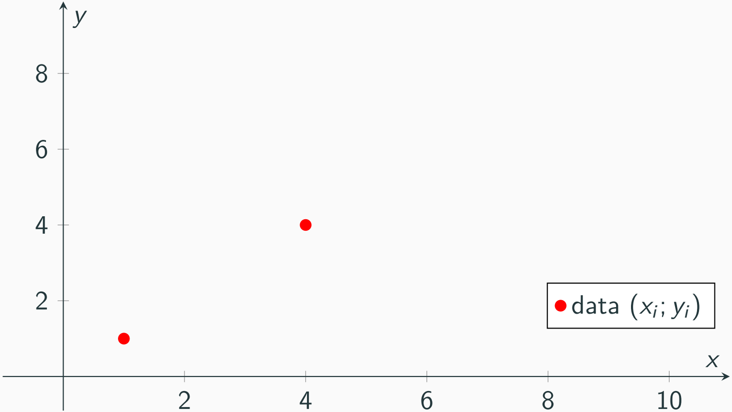
https://en.wikipedia.org/wiki/Machine_learning

Mitchell (1997)

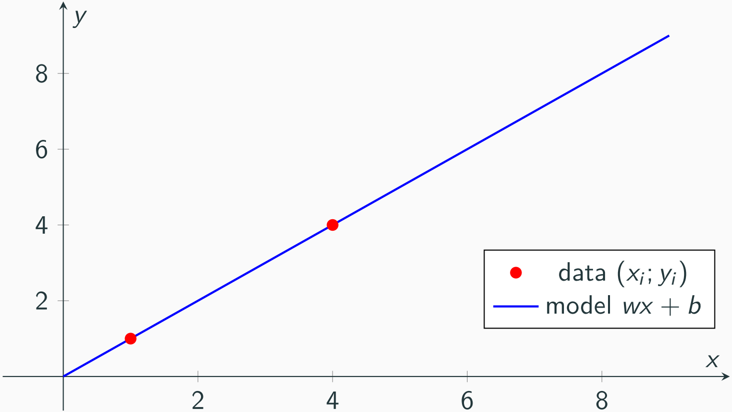
"A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P if its performance at tasks in T, as measured by P, improves with experience E."

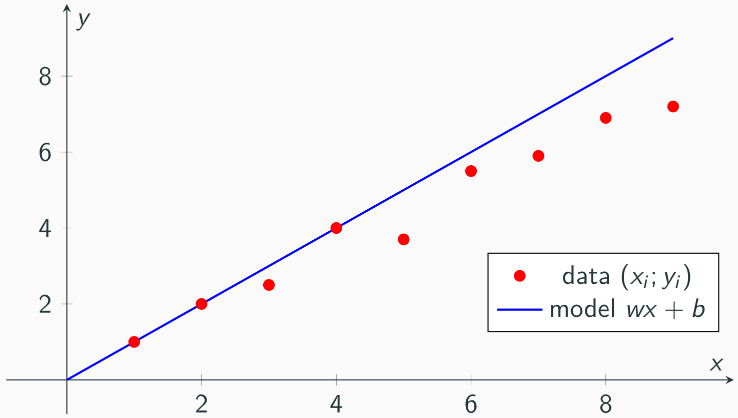
— Mitchell, T. (1997). *Machine Learning*

Learning From Data

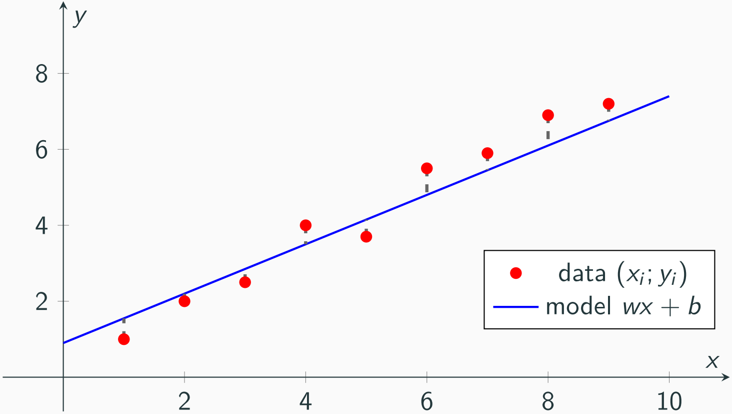


Learning From Data



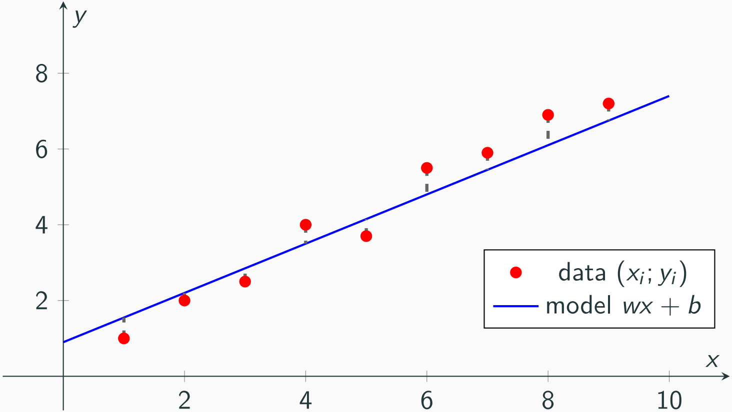

y
8
6
4
2
2
4
6
8
x
data $(x_i; y_i)$
model $wx + b$

Linear Regression Fit



$$\text{Minimize} \quad \sum_{i=1}^{n} (y_i - (wx_i + b))^2$$

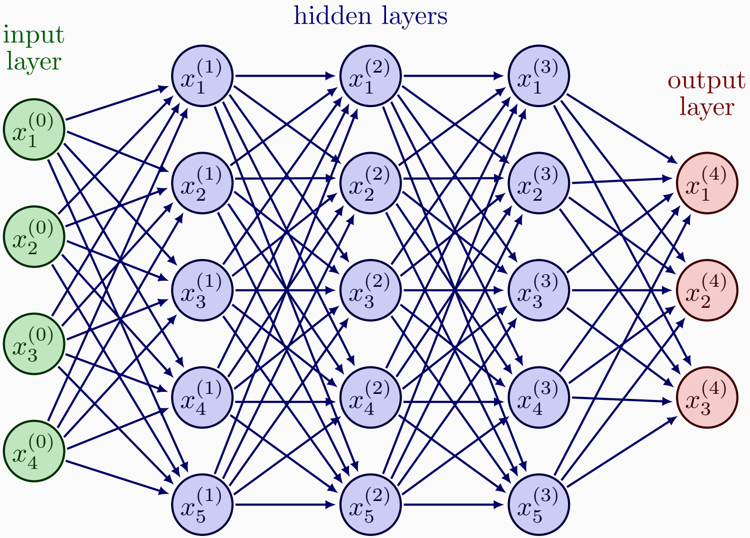
Linear Regression Fit



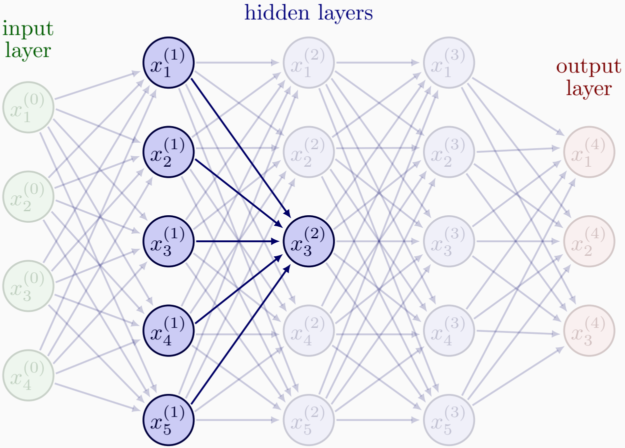
Minimize $\sum_{i=1}^{n} (y_i - (\mathbf{w}x_i + \mathbf{b}))^2$

"learnable parameters"

Structure of an Artificial Neural Network Model

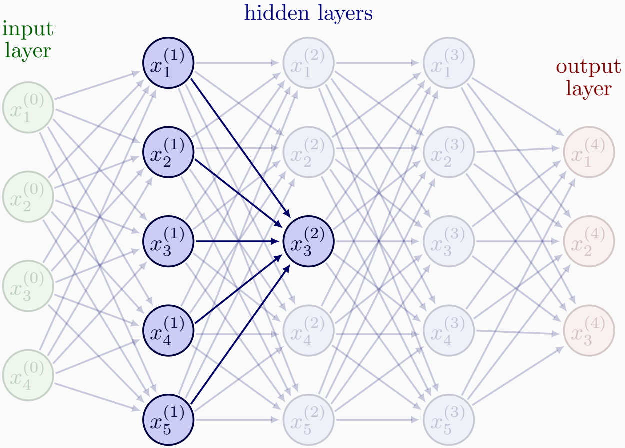


Structure of an Artificial Neural Network Model



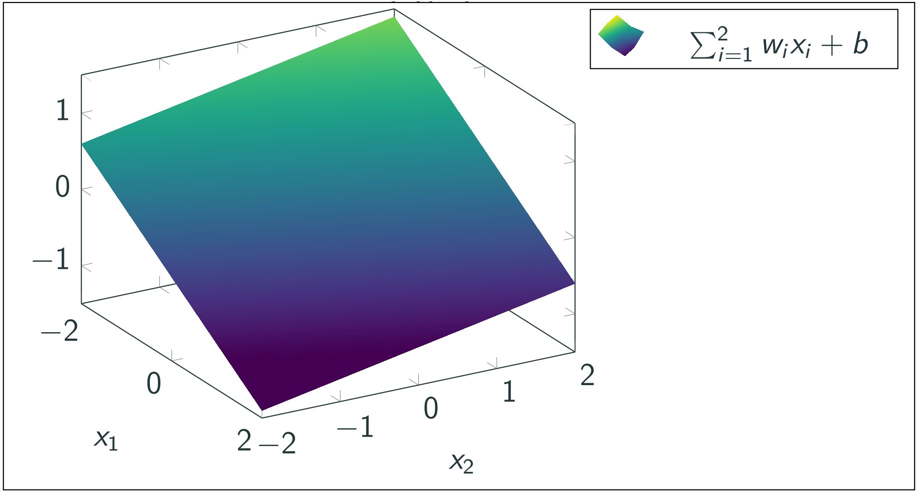
$$x_3^{(2)} = \sigma\left(\sum_{i=1}^{5} w_i^{(2)} x_i^{(1)} + b_3^{(2)}\right)$$

Structure of an Artificial Neural Network Model



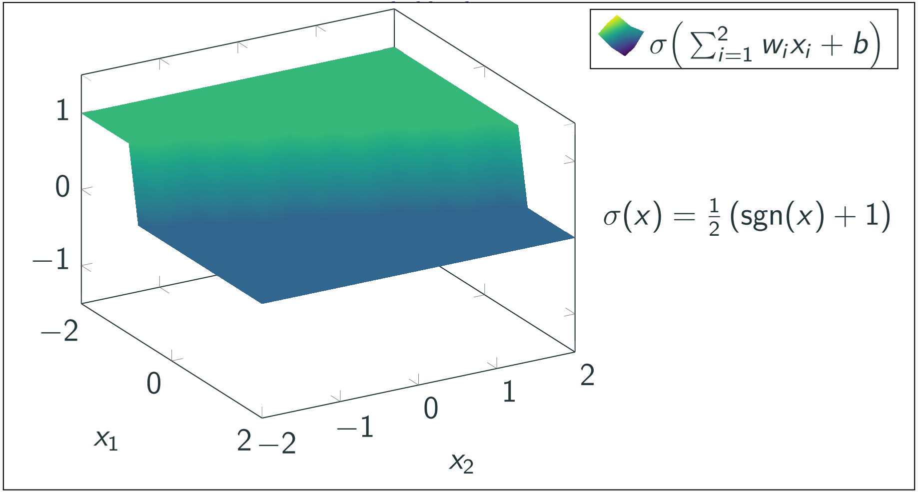
$$x_3^{(2)} = \underbrace{\sigma}_{\text{activation}} \left(\sum_{i=1}^{5} \underbrace{w_i^{(2)}}_{\text{weight}} x_i^{(1)} + \underbrace{b_3^{(2)}}_{\text{bias}} \right)$$

Structure of an Artificial Neural Network Model



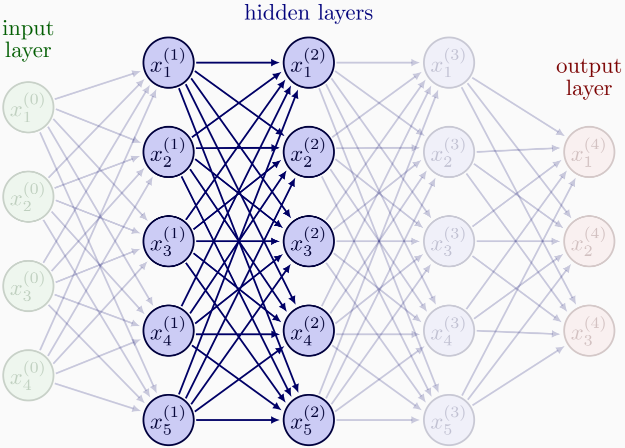
$$x_3^{(2)} = \underbrace{\sigma}_{\text{activation}} \left(\sum_{i=1}^{5} \underbrace{w_i^{(2)}}_{\text{weight}} x_i^{(1)} + \underbrace{b_3^{(2)}}_{\text{bias}} \right)$$

Structure of an Artificial Neural Network Model



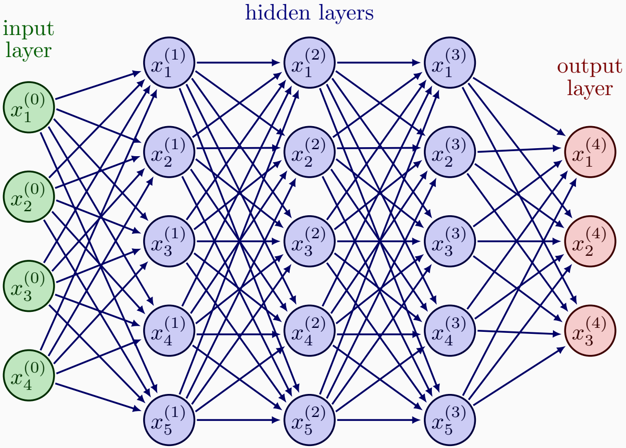
$$x_3^{(2)} = \underbrace{\sigma}_{\text{activation}} \left(\sum_{i=1}^{5} \underbrace{w_i^{(2)}}_{\text{weight}} x_i^{(1)} + \underbrace{b_3^{(2)}}_{\text{bias}} \right)$$

Structure of an Artificial Neural Network Model

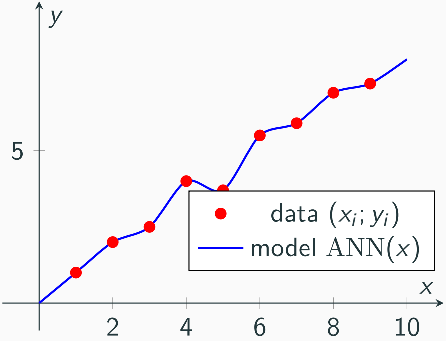


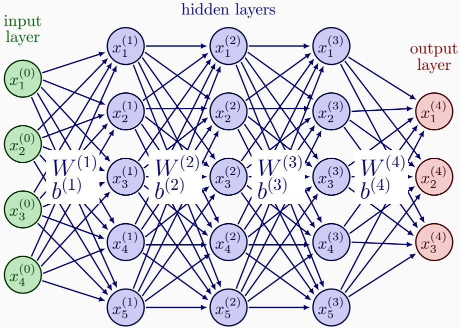
$$x^{(2)} = \sigma\left(W^{(2)}x^{(1)} + b^{(2)}\right)$$

Structure of an Artificial Neural Network Model



$$x^{(j+1)} = \sigma\left(W^{(j+1)}x^{(j)} + b^{(j+1)}\right)$$





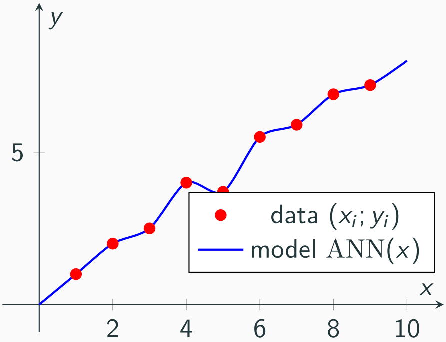
Minimize

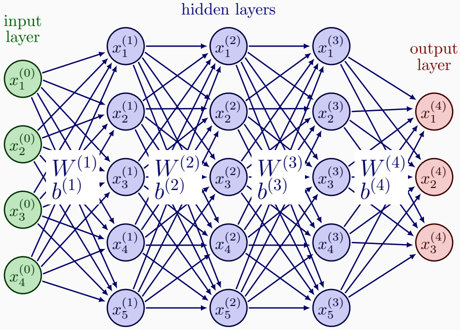
$$\sum_{i=1}^{n} (y_i - \mathrm{ANN}(x_i))^2$$

with respect to

weights $W^{(j)}$ and **biases** $b^{(j)}$.

Training a Neural Network





Minimize

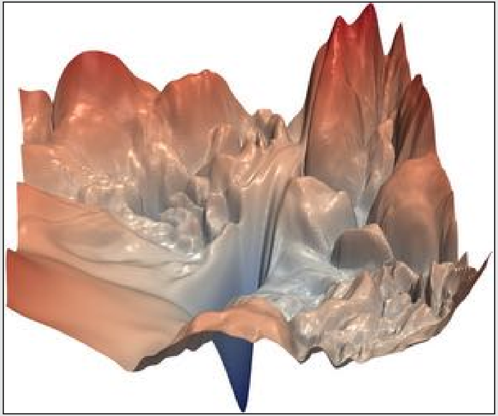
$$\underbrace{\sum_{i=1}^{n} (y_i - \mathrm{ANN}(x_i))^2}_{\text{loss function}}$$

with respect to

weights $W^{(j)}$ and **biases** $b^{(j)}$.

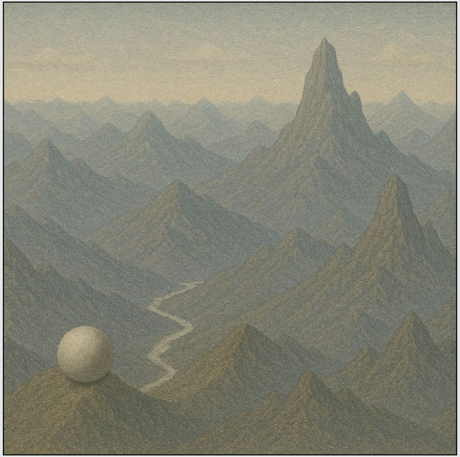
Rolling a Ball Down a Hill

Visualizing the Loss Landscape



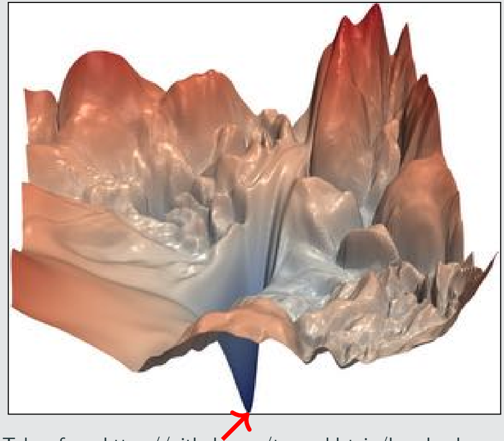
Taken from https://github.com/tomgoldstein/loss-landscape

Training Process



Rolling a Ball Down a Hill

Visualizing the Loss Landscape

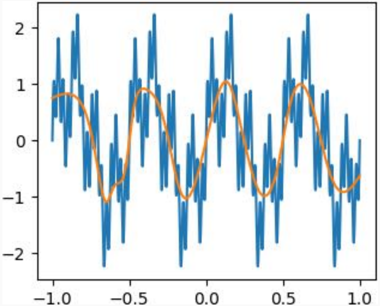


Taken from https://github.com/tomgoldstein/loss-landscape

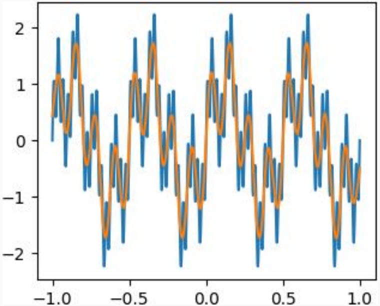
Training Process

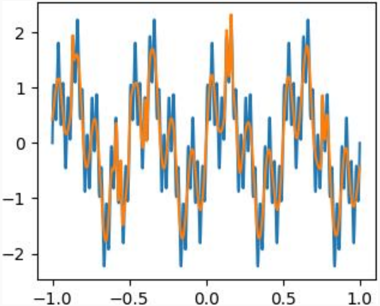
- Follow the **steepest descent**
- We can adjust the **mass of the ball** to avoid getting stuck

3 000 steps

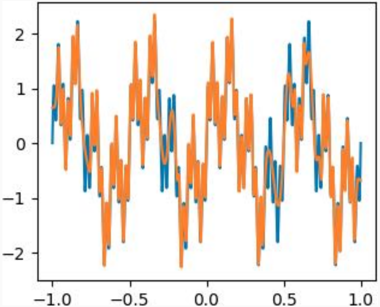


25 000 steps



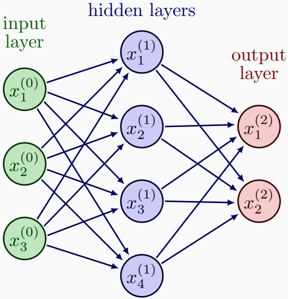


100 000 steps



400 000 steps

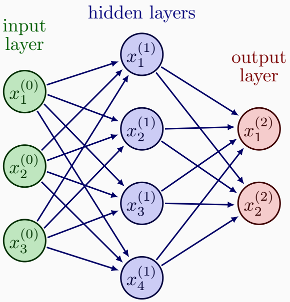
The Power of Nonlinearity



If we **remove the activation** σ, the model is given by:

$$x^{(2)} = W^{(2)} x^{(1)} + b^{(2)} \qquad x^{(1)} = W^{(1)} x^{(0)} + b^{(1)}$$

The Power of Nonlinearity



If we **remove the activation** σ, the model is given by:

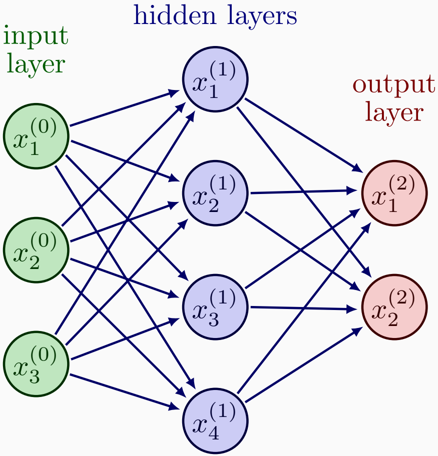
$$x^{(2)} = W^{(2)}x^{(1)} + b^{(2)} \qquad x^{(1)} = W^{(1)}x^{(0)} + b^{(1)}$$

We obtain a **linear model**:

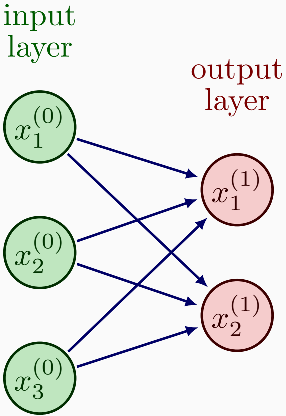
$$x^{(2)} = W^{(2)}\left(W^{(1)}x^{(0)} + b^{(1)}\right) + b^{(2)} = \underbrace{W^{(2)}W^{(1)}}_{:=W}x^{(0)} + \underbrace{W^{(2)}b^{(1)} + b^{(2)}}_{:=b} = Wx^{(0)} + b$$

The Power of Nonlinearity

Without σ



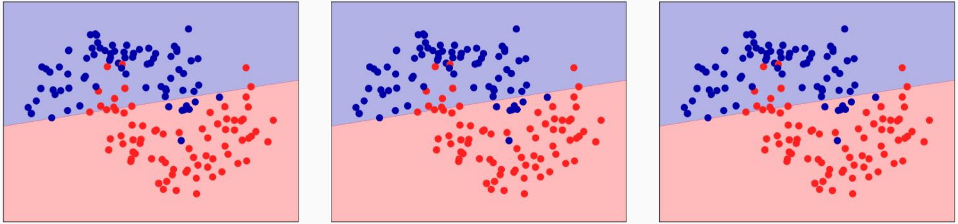
=



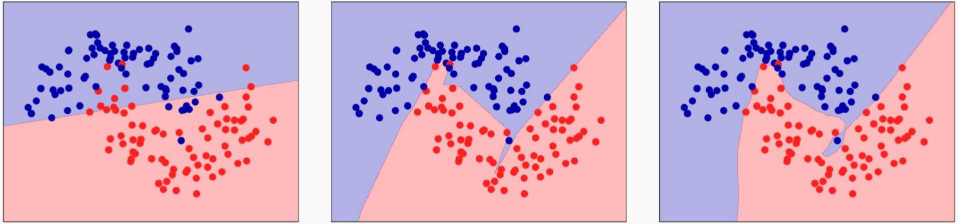
The Power of Nonlinearity

For a classification example:

no σ



$\sigma(x) = \max\{0, x\}$



1 neuron | 10 neurons | 100 neurons

Intermediate Summary

- An artificial neural network consists of layers: each applies a **linear transformation**

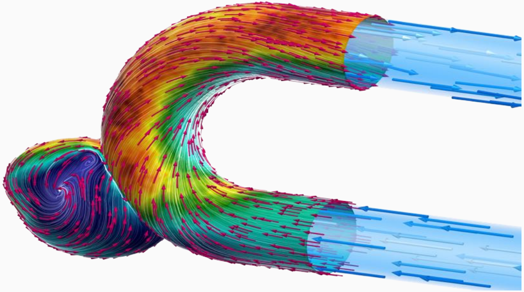
$$Wx + b$$

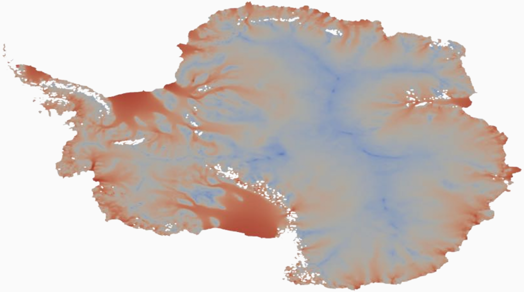
followed by a **nonlinear activation**

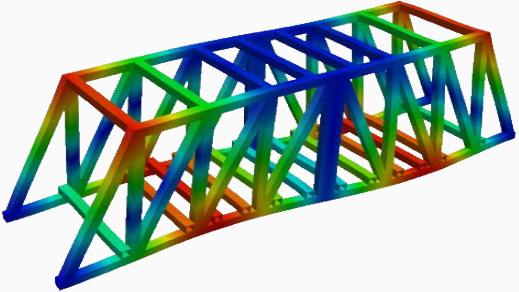
$$\sigma$$

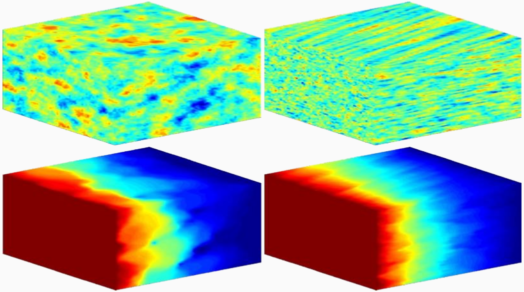
- **Nonlinearity** gives neural networks their power
- Training: **adjusting** W and b
- Training is **complex**, but can be seen as rolling a ball down a hill

Can Neural Networks Learn Physics?



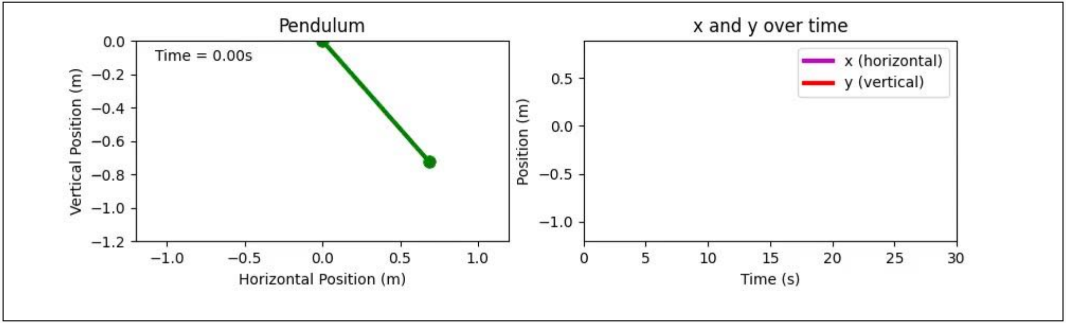




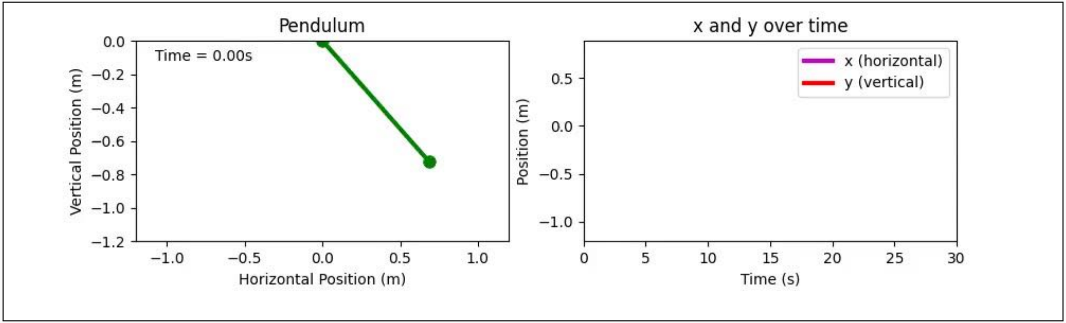


A Swinging Pendulum — Undamped

Let us consider a “simple” example!

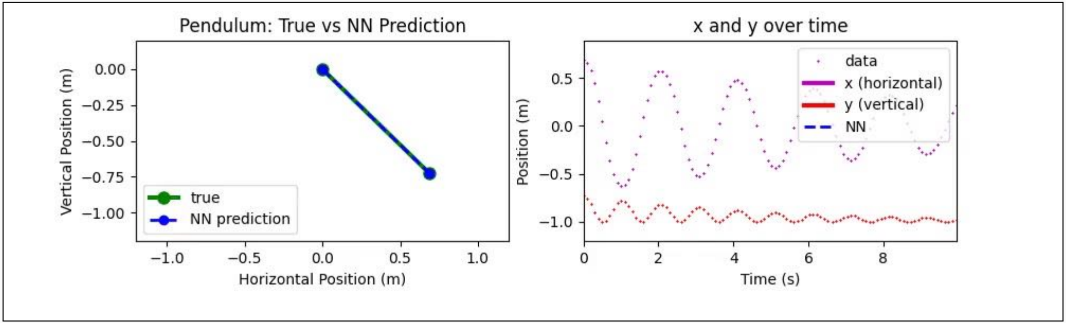


Let us consider a “simple” example!



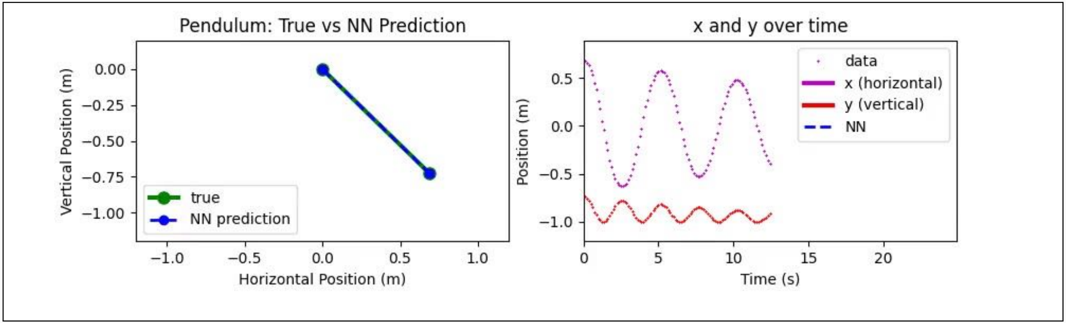
Learning the Pendulum Motion Using Observations

$$\text{minimize} \quad \sum_{i=1}^{n} (\theta_i - \mathrm{ANN}(t_i))^2$$



Did the Neural Network Understand the Physics?

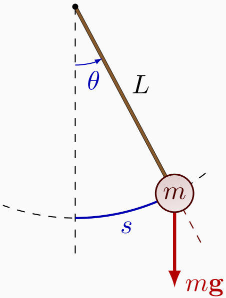
$$\text{minimize} \quad \sum_{i=1}^{n} (\theta_i - \mathrm{ANN}(t_i))^2$$

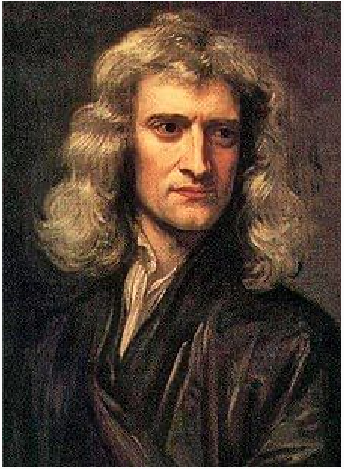


$$\mathbf{F} = ma$$

- F: net force acting on a body
- m: mass of the body
- a: acceleration of the body

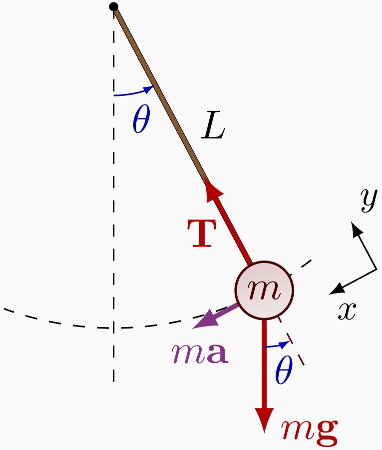
Pendulum



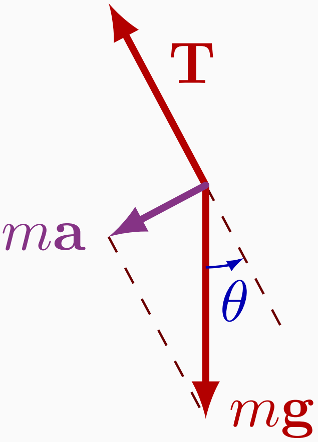


Sir Isaac Newton (1643–1727) formulated the laws of motion and gravity.

Pendulum

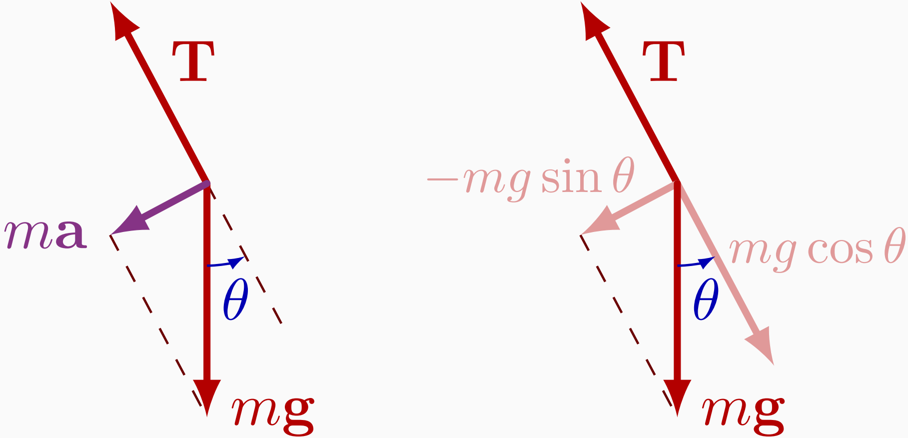


θ
L
$\mathbf{T}$
m
y
x
$m\mathbf{a}$
θ
$m\mathbf{g}$

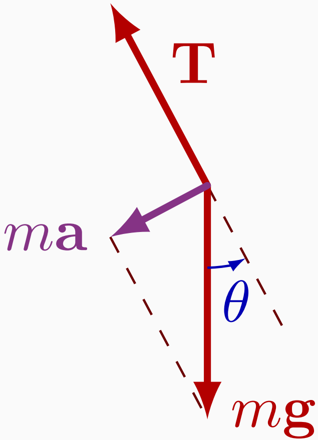


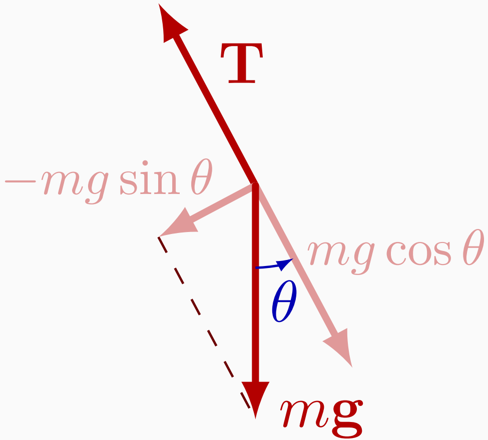
$\mathbf{T}$
$m\mathbf{a}$
θ
$m\mathbf{g}$

Pendulum


$\mathbf{T}$
$m\mathbf{a}$
θ
$m\mathbf{g}$
$\mathbf{T}$
$-mg\sin\theta$
$mg\cos\theta$
θ
$m\mathbf{g}$

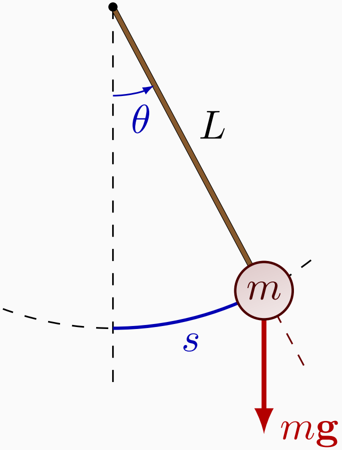
Pendulum





$$\Rightarrow ma = -mg\sin\theta \Rightarrow a = -g\sin\theta$$

Pendulum



We have

$$\begin{aligned} s(t) &= L\theta(t) \\ \Rightarrow v(t) &= L\theta'(t) \qquad \textit{(velocity)} \\ \Rightarrow a(t) &= L\theta''(t) \qquad \textit{(acceleration)} \end{aligned}$$

Hence, we obtain

$$\begin{aligned} a(t) &= -g \sin\theta(t) \\ \Rightarrow \theta''(t) &= -\frac{g}{L} \sin\theta(t) \end{aligned}$$

Newton's Second Law of Motion and the Pendulum

The change of the angle θ follows the equation

$$\theta''(t) + \frac{g}{L} \sin \theta(t) = 0.$$

If we add **damping** λ (coefficient of friction) **relative to the velocity**, we obtain the equation:

$$\theta''(t) + \lambda\theta'(t) + \frac{g}{L} \sin \theta(t) = 0.$$

The change of the angle θ follows the equation

$$\theta''(t) + \frac{g}{L} \sin \theta(t) = 0.$$

If we add **damping** λ (coefficient of friction) **relative to the velocity**, we obtain the equation:

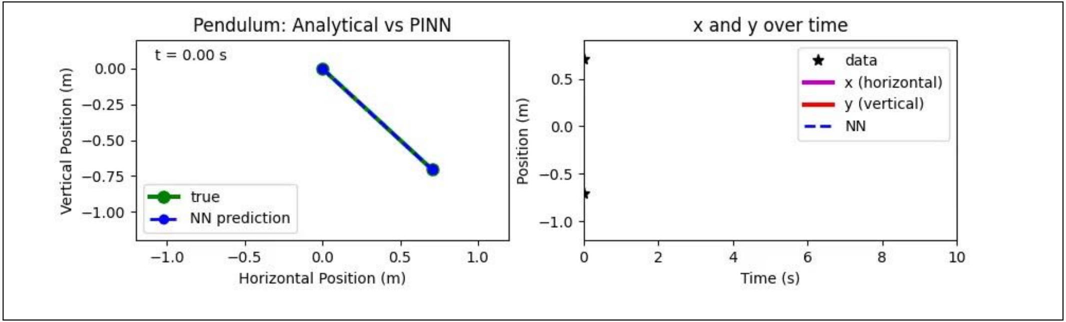
$$\theta''(t) + \lambda\theta'(t) + \frac{g}{L} \sin \theta(t) = 0.$$

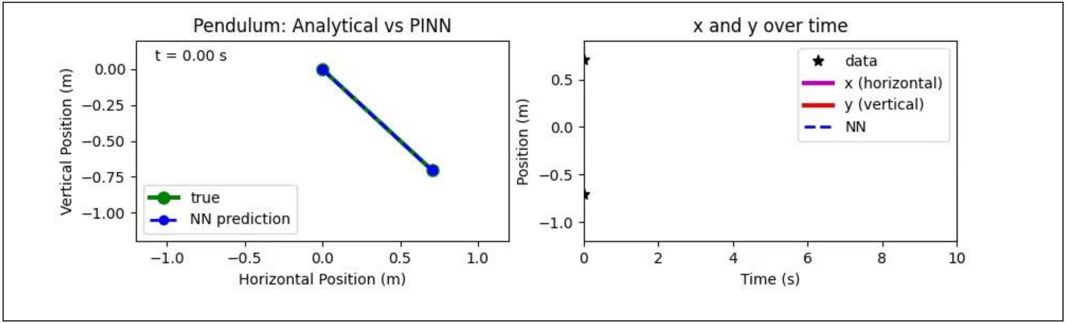
Where no data

$$\text{minimize} \quad \sum_{i=1}^{n} (\theta_i - \mathrm{ANN}(t_i))^2$$

is given, employ instead:

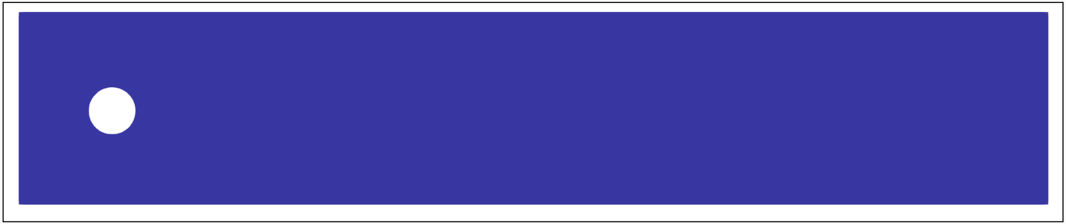
$$\text{minimize} \quad \sum_{i=1}^{n} \left(\mathrm{ANN}''(t) + \lambda \mathrm{ANN}'(t_i) + \frac{g}{L} \sin \mathrm{ANN}(t_i) \right)^2$$





This approach is called **physics-informed neural networks (PINNs)**!

Current Research – More “Challenging” Problems



Learn More About . . .

Deep learning

- **Courses**
  - **TU Delft**: https://ocw.tudelft.nl/courses/ai-skills-introduction-to-unsupervised-deep-and-reinforcement-learning/subjects/module-4-introduction-to-deep-learning/
  - **Stanford University**: https://cs230.stanford.edu/

Physics-informed neural networks

- **Blog posts**
  https://benmoseley.blog/my-research/so-what-is-a-physics-informed-neural-network/
- **Videos**
  - https://www.youtube.com/@CAMLabETHZurich/videos
  - https://www.youtube.com/@Eigensteve

. . . and more!

Thank you for your attention!

Questions?