Machine Learning
for Many-Body Systems in Physics !

Michael Hecht

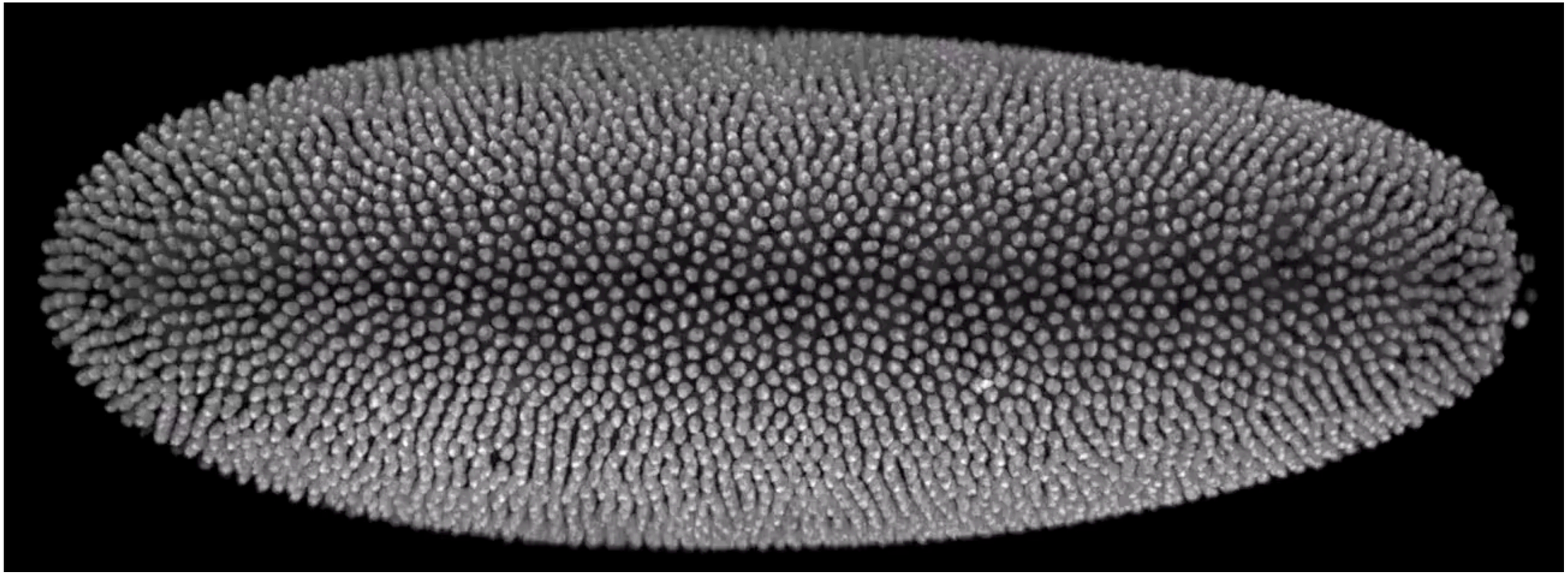
University Kiel, 2022

CASUS
CENTER FOR ADVANCED SYSTEMS UNDERSTANDING
www.casus.science

HZDR HELMHOLTZ ZENTRUM DRESDEN ROSSENDORF
TECHNISCHE UNIVERSITÄT DRESDEN
CBG Max Planck Institute of Molecular Cell Biology and Genetics
UFZ HELMHOLTZ Centre for Environmental Research
Uniwersytet Wrocławski

SPONSORED BY THE
Federal Ministry of Education and Research
STAATSMINISTERIUM FÜR WISSENSCHAFT KULTUR UND TOURISMUS
Freistaat SACHSEN

Motivating Example:
4th-order Bio-Physical PDE Models on 2D surfaces



Loic Royer

Ivo F. Sbalzarini

Drosophila Embryo development, by Loic Royer, MPI-CBG / CZ BioHub, San Francisco

center for systems biology dresden

CBG
Max Planck Institute of Molecular Cell Biology and Genetics

Gene Meyers

Marino Zerial

CASUS
CENTER FOR ADVANCED SYSTEMS UNDERSTANDING

www.casus.science

The Bottleneck

How to compute Multivariate Function Expansions that closely approximate the ground truth function ?

- **Geometric approximation rates of Fourier expansions for analytic periodic functions, FFTs**

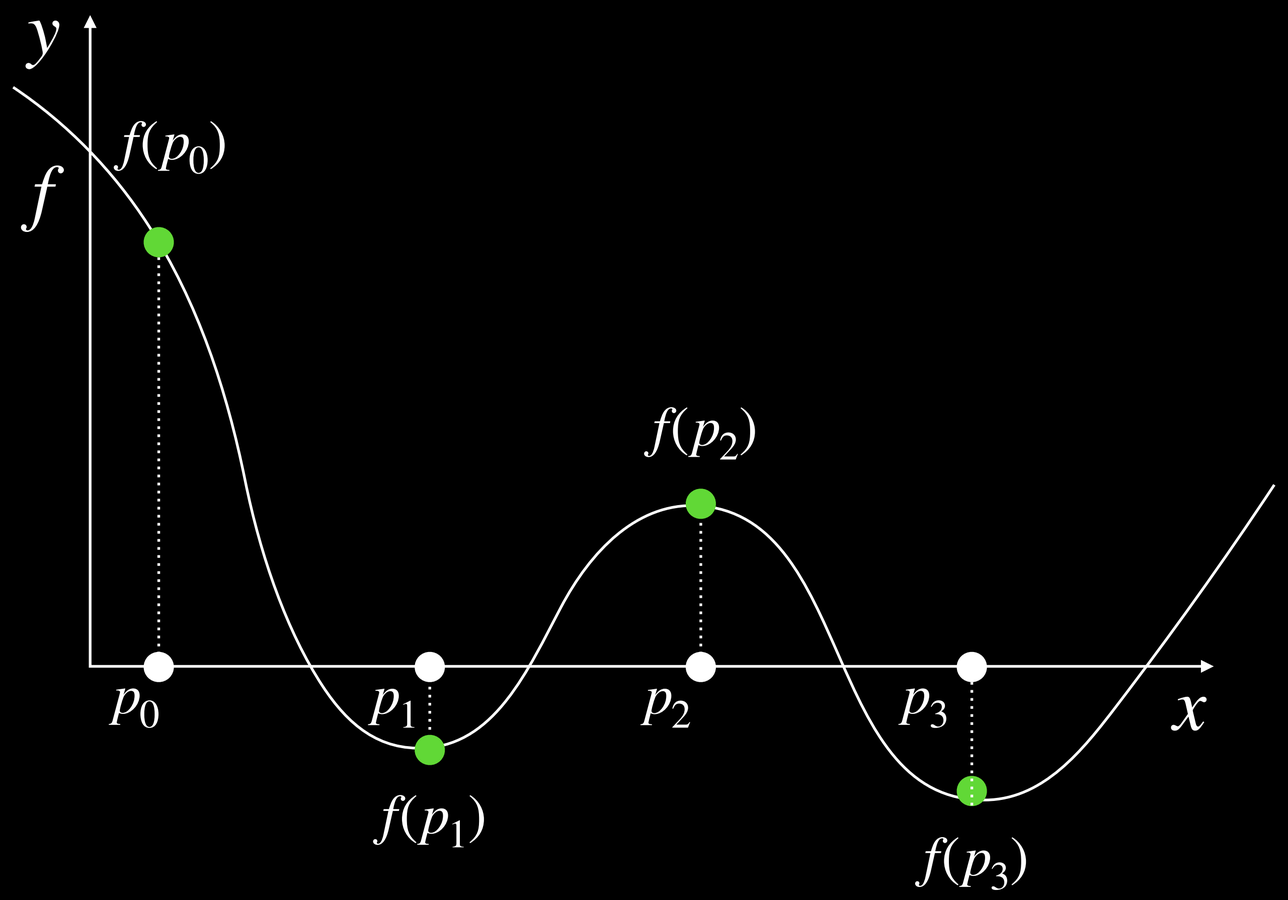
$$\theta(z) \approx \theta_n = \sum_{\|\alpha\|_\infty \leq n} c_\alpha e^{2\pi i \alpha \cdot z}, \qquad \|\theta - \theta_n\|_{C^0(\Omega)} = \mathcal{O}(r^{-n}), \qquad r > 1$$

- **Geometric approximation rates of the Coulomb / Gravitation field expansion in FMMs**

$$\phi(z) \approx \phi_n = Q \log(z) + \sum_{k=1}^{n} \frac{a_k}{z_k}, \qquad \|\phi - \phi_n\|_{C^0(\Omega)} = \mathcal{O}(z^{-n}), \qquad |z| > 1 .$$

Gauss, C. F. (1886). Theoria interpolationis methodo nova tractata Werke band 3, 265–327. *Göttingen: Königliche Gesellschaft der Wissenschaften*.
Cooley, J. W., & Tukey, J. W. (1965). An algorithm for the machine calculation of complex Fourier series. *Mathematics of computation*, *19*(90), 297-301.
Greengard, L., & Rokhlin, V. (1987). A fast algorithm for particle simulations. *Journal of computational physics*, *73*(2), 325-348.

1D Interpolation



Naïve Interpolation

Vandermonde Matrix

$$V_{n,P} = \begin{pmatrix} 1 & p_0 & \cdots & p_0^n \\ \vdots & \vdots & & \vdots \\ 1 & p_n & \cdots & p_n^n \end{pmatrix}, \quad P = \begin{pmatrix} p_0 \\ \vdots \\ p_n \end{pmatrix}, \quad F = \begin{pmatrix} f(p_0) \\ \vdots \\ f(p_n) \end{pmatrix}$$

$$C = V_{n,P}^{-1} \cdot F, \quad C = (c_0, \ldots, c_n)$$

$$Q_{f,n}(x) = c_0 + c_1 x + \cdots + c_n x^n$$

Runtime $\mathcal{O}(n^3)$ **Storage** $\mathcal{O}(n^2)$ **Evaluation** $\mathcal{O}(n^2)$

One of the most influential scientists of all time

Sir Isaac Newton
1643-1726

- **Mathematics**
- **Physics**
- **Optics**
- **Computational Sciences**

Philosophiæ Naturalis Principia Mathematica

Two of the most influential scientists of all time

Sir Isaac Newton
1643-1726

Philosophiæ Naturalis Principia Mathematica

Joseph-Louis Lagrange
1736-1813

Mécanique analytique

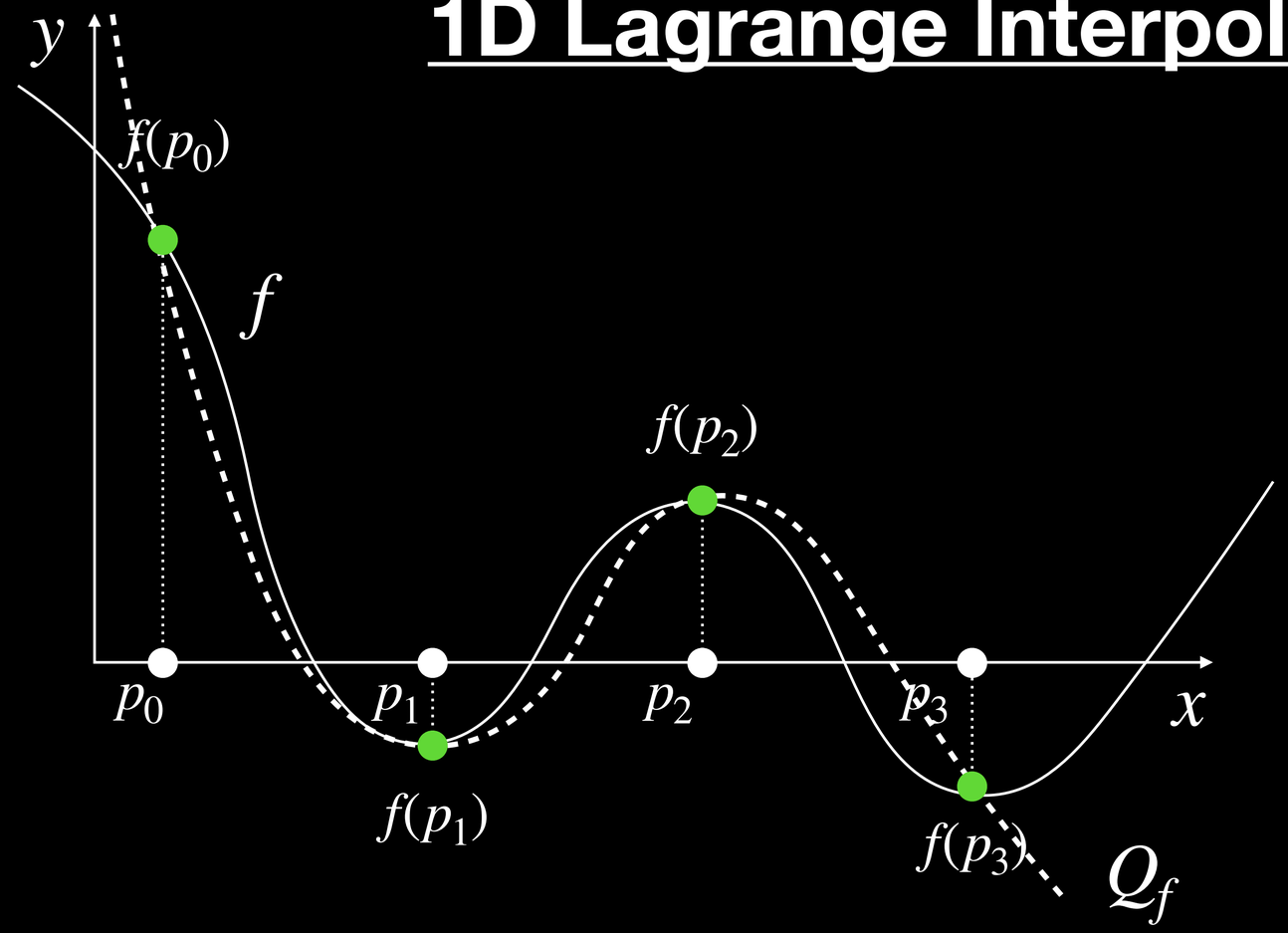
Divided Difference Scheme

$$[p_0]f := f(p_0)\,, \quad [p_i, \ldots, p_j]f := \frac{[p_i, \ldots, p_{j-1}]f - [p_{i+1}, \ldots, p_j]f}{x_j - x_i}\,, \; j \geq i$$

$$\begin{array}{ccccccccc}
[p_0]f & & & & & & & & \\
 & \searrow & & & & & & & \\
[p_1]f & \rightarrow & [p_0,p_1]f & & & & & & \\
 & \searrow & & \searrow & & & & & \\
[p_2]f & \rightarrow & [p_1,p_2]f & \rightarrow & [p_0,p_1,p_2]f & & & & \\
\vdots & \vdots & \vdots & \vdots & \vdots & \ddots & & & \\
 & \searrow & & \searrow & & & \searrow & & \\
[p_n]f & \rightarrow & [p_{n-1},p_n]f & \rightarrow & [p_{n-2},p_{n-1},p_n]f & \cdots & \rightarrow & [p_0 \cdots p_n]f
\end{array}$$

Runtime	$\mathcal{O}(n^2)$
Storage	$\mathcal{O}(n)$
Evaluation	$\mathcal{O}(n)$

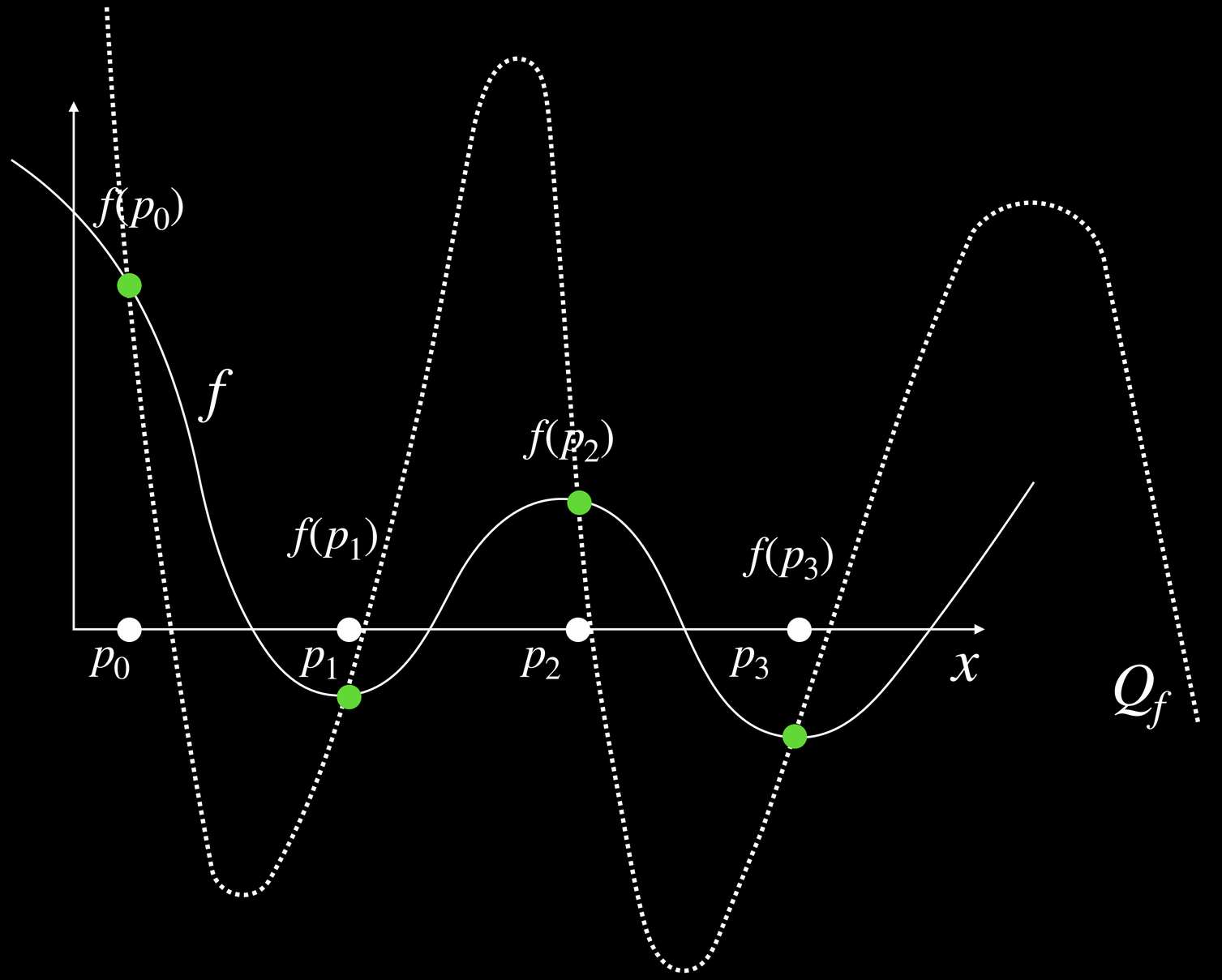
1D Lagrange Interpolation



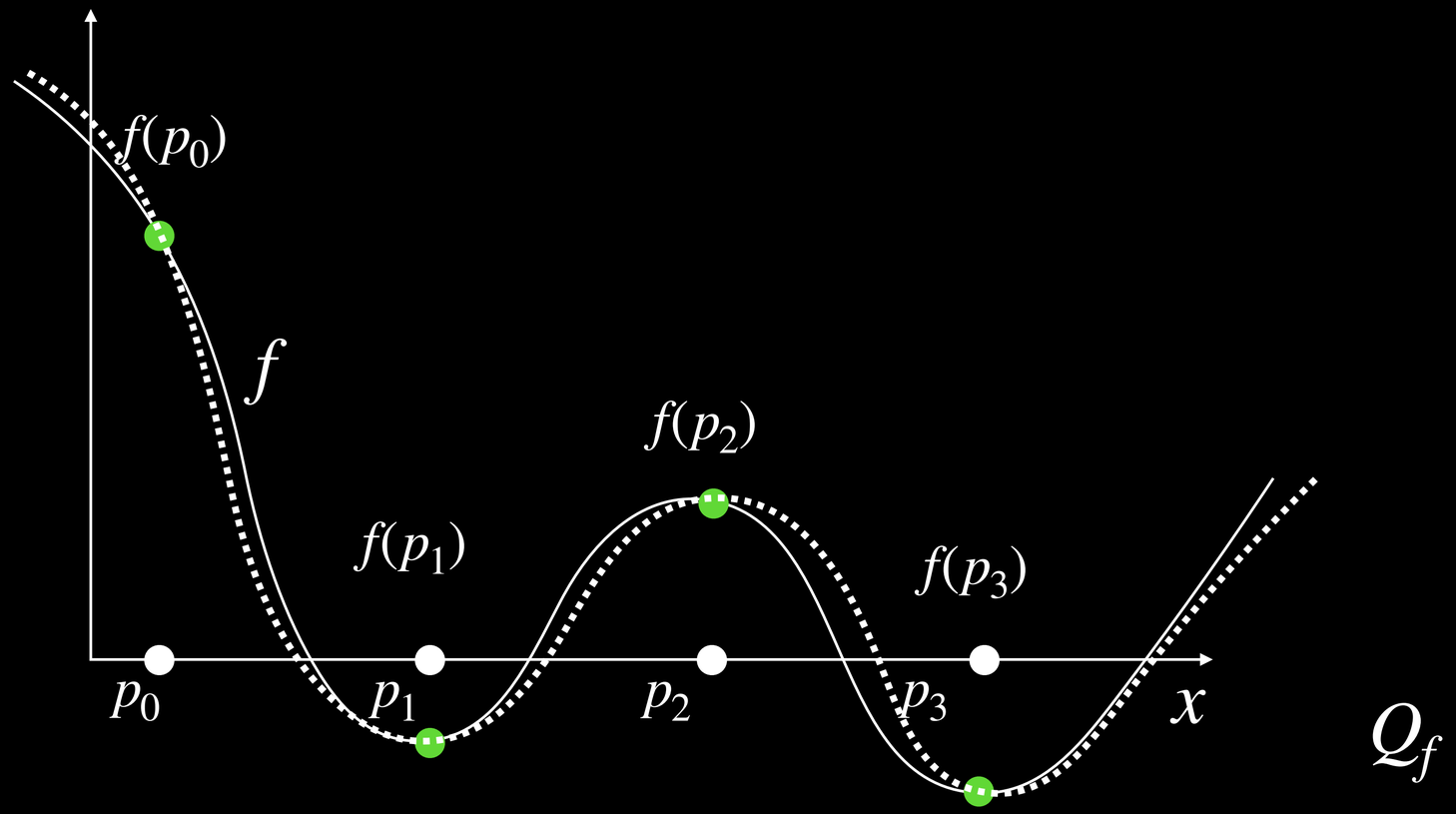
Runtime	$\mathcal{O}(n)$
Storage	$\mathcal{O}(n)$
Evaluation	$\mathcal{O}(n)$

$$Q_{f,3}(x) = \prod_{j=0}^{3}(x - p_j)\left(f(p_0)\frac{\omega_0}{x - p_0} + f(p_1)\frac{\omega_1}{x - p_1} + \sum_{i=2}^{n} f(p_i)\frac{\omega_i}{x - p_i}\right)$$

Runge's Phenomena



Runge's Phenomena - Can be avoided for Chebyshev Nodes



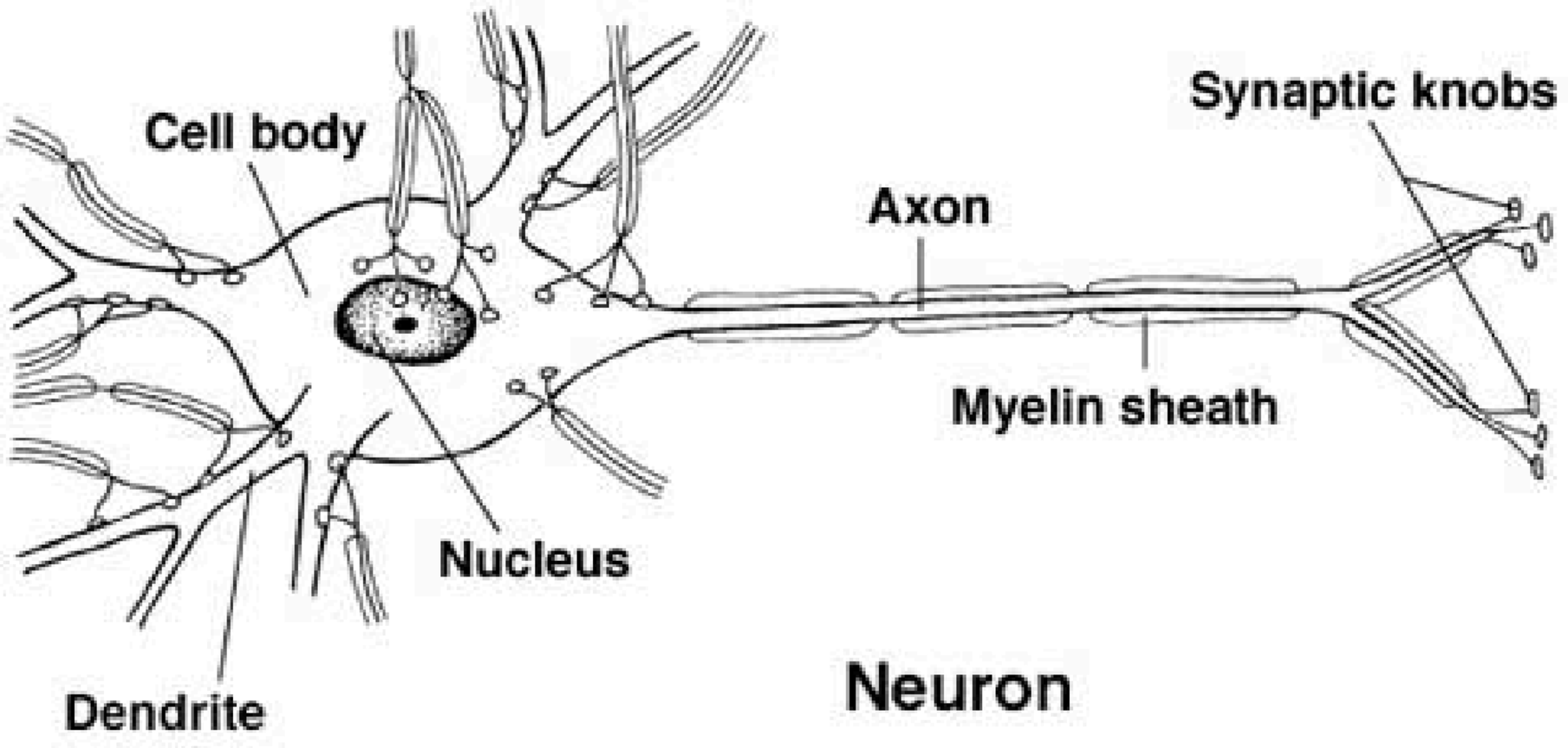
Computer … someone who computes

Complex computations

- physics & astrophysics
- engineering
- aeronautics
- economics
- etc

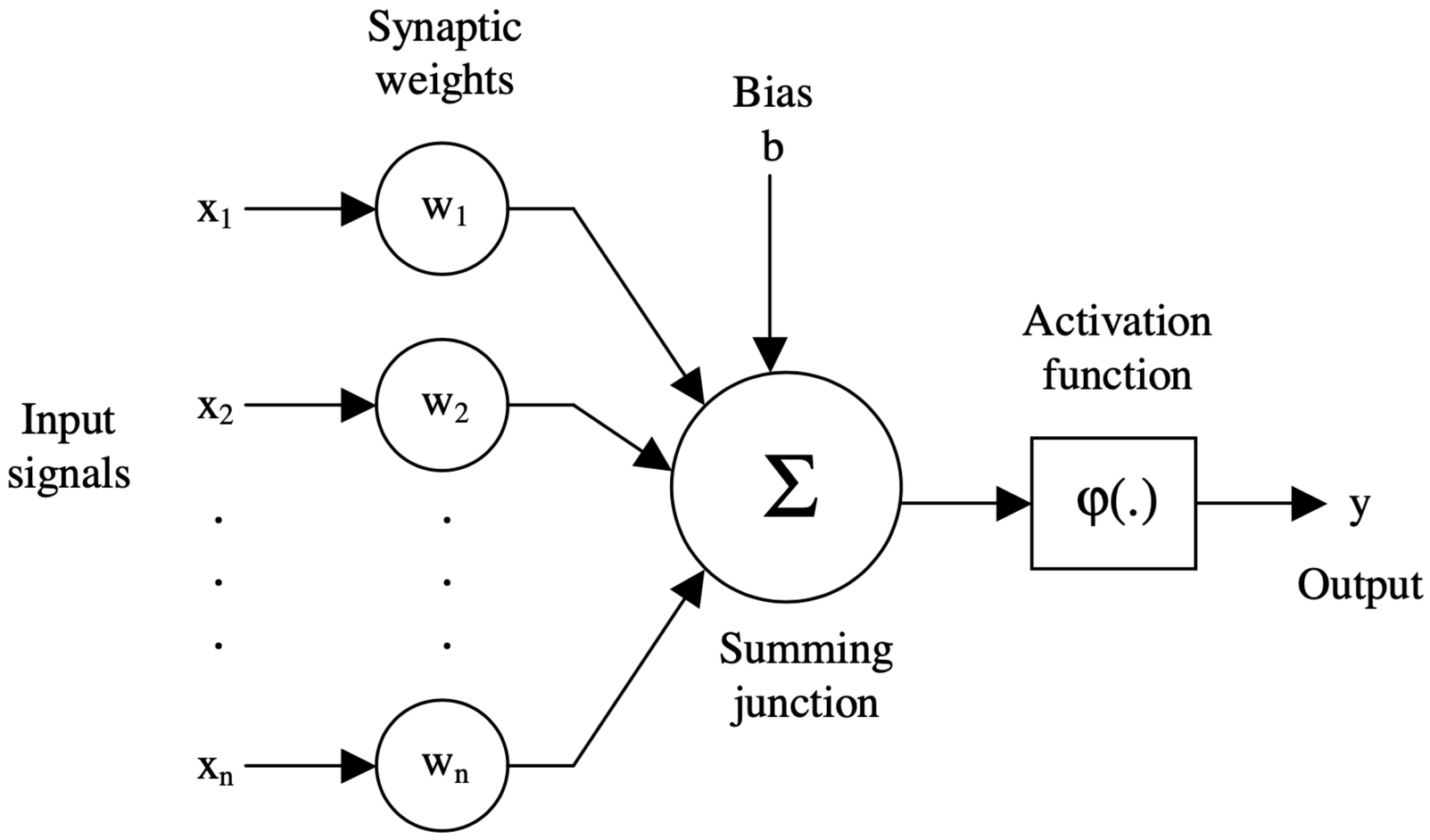
NACA (aeronautic) High Speed Computer Room (1949)

What are Neural Nets (NNs) ?



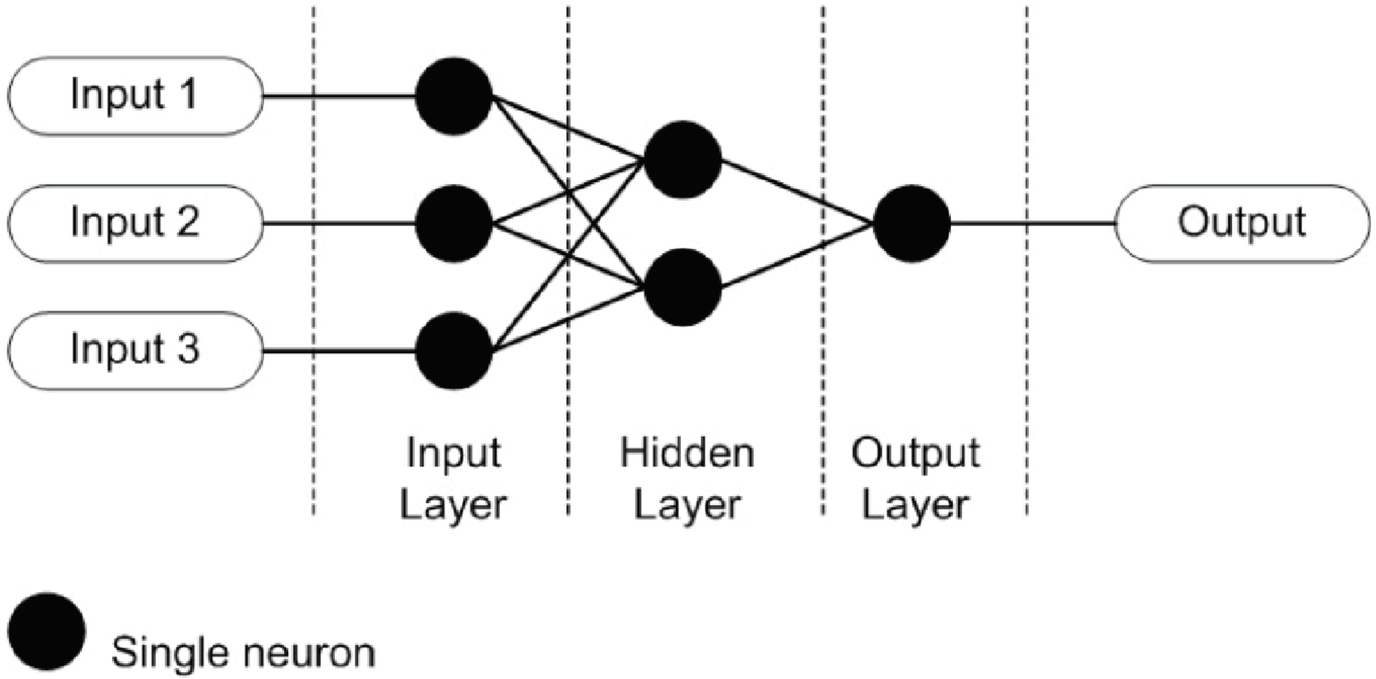
Csáji, B. C. (2001). Approximation with artificial neural networks. *Faculty of Sciences, Etvs Lornd University, Hungary*, *24*(48), 7.
Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT press.

What are Neural Nets (NNs) ?



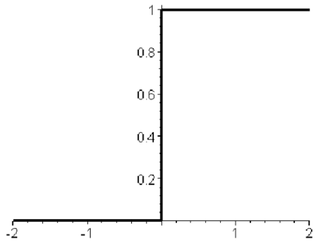
Csáji, B. C. (2001). Approximation with artificial neural networks. *Faculty of Sciences, Etvs Lornd University, Hungary*, *24*(48), 7.
Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT press.

Combination of the Neurons yields NNs



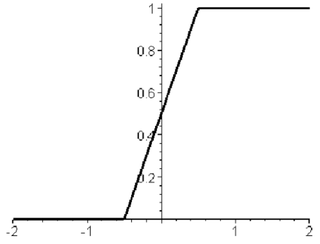
(1) *Threshold function*:

$$\varphi(x) = \begin{cases} 1 & x \geq 0 \\ 0 & x < 0 \end{cases}$$



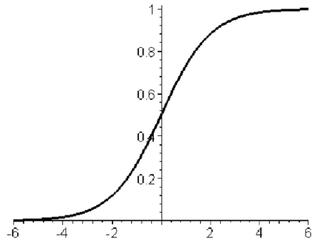
(2) *Piecewise linear function*:

$$\varphi(x) = \begin{cases} 1 & x \geq +\frac{1}{2} \\ x + \frac{1}{2} & \frac{1}{2} \geq x \geq -\frac{1}{2} \\ 0 & -\frac{1}{2} \geq x \end{cases}$$



(3) *Sigmoid function (logistic function)*:

$$\varphi(x) = \frac{1}{1+e^{-x}}$$



Neural Net Training

Given is a continuous function

$$u : \Omega = [-1,1]^m \longrightarrow \mathbb{R}$$

that is sampled on a set $P \subseteq \Omega$.

Consider a neural net

$$\hat{u} : \Omega \times \Theta \longrightarrow \mathbb{R}\,, \quad \hat{u} = \hat{u}(x, W, B)$$

depending on x and weights & bias

The mean square error (MSE) loss

$$\mathscr{L}(W, B) : \Theta \longrightarrow \mathbb{R}^+$$

is given by

$$\mathscr{L}(W, B) = \sum_{p \in P} \|u(p) - \hat{u}(p, W, B)\|^2 \approx \int_\Omega \|u(x) - \hat{u}(x, W, B)\|^2 d\Omega.$$

Neural Net Training is now given by loss minimisation due to stochastic gradient descent

Initial setting $W_0, B_0 \in \Theta$

$$(W_{k+1}, B_{k+1}) = (W_k, B_k) - \tau \nabla \mathscr{L}(W_k, B_k)$$

This is a non-convex & non-linear optimisation problem !!!

It's often simply termed Backpropagation

Automatic Differentiation of NNs

Chain Rule For Tensors

Given

Indices: $J = J_1 \times ... \times J_m \subset \mathbb{N}^m$ and $I = I_1 \times ... \times I_n \subset \mathbb{N}^n$

Functions: $g : \mathbb{R}(J) \to \mathbb{R}(I)$ and $f : \mathbb{R}(I) \to \mathbb{R}$

Partials

$\forall X \in \mathbb{R}(J) \forall j = (j_1, ..., j_m) \in J$ the following holds:

$$\partial_{X_j}(f \circ g)(X) = \sum_{i_1 \in I_1} ... \sum_{i_n \in I_n} (\partial_{(i_1,...,i_n)} f)(g(X)) \cdot (\partial_{(j_1,...,j_n)} g_i)(X)$$
$$= \sum_{i \in I} (\partial_i f)(g(X))(\partial_j g_i)(X) = \nabla f(g(X)) \cdot (\partial_j g_i)_{i \in I}$$

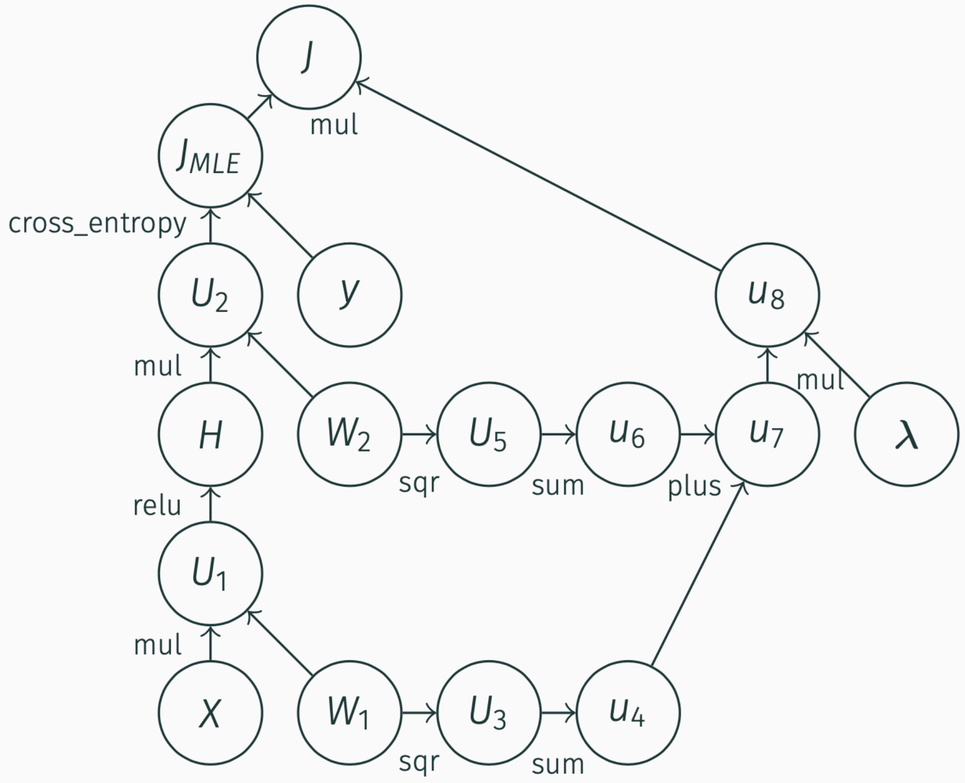
Gradient

$\mathcal{D}g : \mathbb{R}(J) \to \mathbb{R}(I)$, $(\mathcal{D}g)_{i,j} = (\partial_j g_i)$ $\forall (i, j) \in I \times J$ the following holds:

$$\nabla(f \circ g)(X) = \nabla f(g(X)) \cdot \mathcal{D}g(X) \quad \forall X \in \mathbb{R}(J)$$

MLP With One Hidden Layer (multilayer perceptron)

Computational Graph of Forward-Propagation



* Use minibatch SGD
* Op's correspond to tensors

tensors

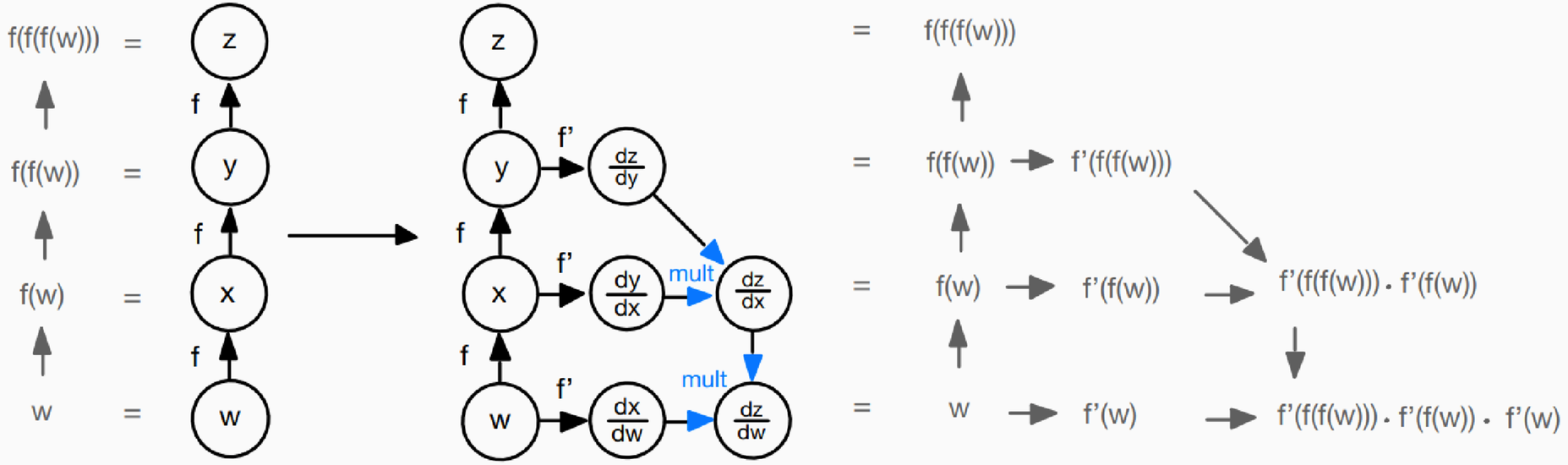
y	...	class labels
X	...	design matrix
W_1, W_2	...	weight

Objective

Compute $\nabla_{W_1} J$ and $\nabla_{W_2} J$

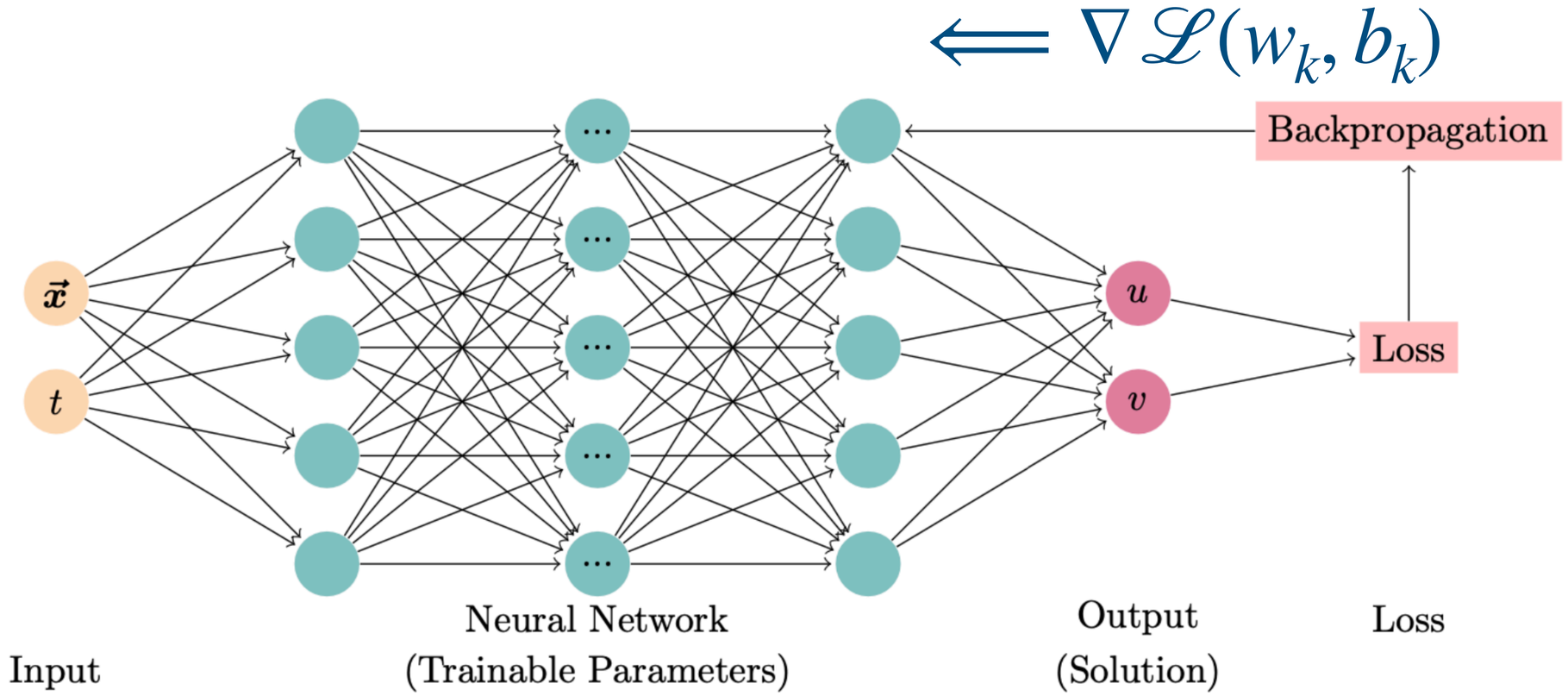
Symbol-To-Symbol Derivatives

Brief Example



BACKPROPAGATION:

Choose a stochastic mini-batch $(w_k, b_k) \subsetneq (W_k, B_k)$ and upgrade the weights and bias due to gradient descent.



Iterate this “training step” !

Physics Informed Neural Nets (PINNs)

We are seeking for a smooth function solving a PDE, e.g.

$$\Delta u + V(u) = 0 \quad \text{in} \quad \Omega = [-1,1]^m$$

and $u_{|\partial\Omega} = f$.

Consider a neural net

$$\hat{u} : \Omega \times \Theta \longrightarrow \mathbb{R}\,, \quad \hat{u} = \hat{u}(x, W, B)$$

depending on x **and weights & bias**

The mean square error (MSE) loss for this problem becomes

$$\mathscr{L}(W, B) : \Theta \longrightarrow \mathbb{R}^+$$

$$\mathscr{L}(W, B) = \sum_{p \in P} \|\Delta\hat{u}(p, W, B) + V(\hat{u}(p, W, B))\|^2 + \sum_{q \in Q} \|\hat{u}(q, W, B) - f(q, W, B)\|^2\,,$$

where $P \subseteq \Omega\,, Q \subseteq \partial\Omega$.

Can Neural Nets learn anything ?

Universal Approximation Theorem[1]

For any continuous function

$$f : \Omega \subseteq \mathbb{R}^m \longrightarrow \mathbb{R}\,,\ m \in \mathbb{N}$$

and $\varepsilon > 0$ there is a NN with one hidden layer such that:

$$\|f - \nu\|_{C^0(\Omega)} < \varepsilon.$$

No guarantee to derive ν by **Neural-Net-Training** !

Feed Forward ReLu Nets are splines[2]

Approximation Power of multivariate splines[3]

$f : \Omega \subseteq \mathbb{R}^m \longrightarrow \mathbb{R}\,,\ n+1$ differentiable. Then:

$$\mathrm{dist}(f, S_{n,\Delta}) \leq c(\Delta)\|D^{r+1}f\|_{C^0(\Omega)}|\,\Delta\,|^{r+1}$$

$S_{n,\Delta}$ piecewise polynomial functions of degree $n \geq 3r+1$, Δ mesh size.

Approximation is **guaranteed** for highly regular functions with **polynomial rate**

$$\|f - \sigma\|_{C^0(\Omega)} \in \mathcal{O}(\Delta^{r+1})\,, \sigma \in S_{n,\Delta}$$

[1] Hornik, K., Stinchcombe, M., & White, H. (1989). Multilayer feedforward networks are universal approximators. *Neural networks*, *2*(5), 359-366.
[2] Hansson, M., & Olsson, C. (2017). Feedforward neural networks with ReLU activation functions are linear splines. *Bachelor's Theses in Mathematical Sciences*.
[3] De Boor, C., & Höllig, K. (1988). Approximation power of smooth bivariate pp functions. *Mathematische Zeitschrift*, *197*(3), 343-363.

CASUS
CENTER FOR ADVANCED SYSTEMS UNDERSTANDING
www.casus.science

Can Polynomials learn anything ?

Stone-Weierstrass Approximation Theorem[1]

For any continuous function

$$f : \Omega \subseteq \mathbb{R}^m \longrightarrow \mathbb{R}\,, \;\; m \in \mathbb{N}$$

and $\varepsilon > 0$ there is a polynomial $Q \in \Pi_{m,n}$ of degree $n \in \mathbb{N}$ such that:

$$\|f - Q\|_{C^0(\Omega)} < \varepsilon.$$

No guarantee to derive Q by **Polynomial Interpolation** !

Polynomial Interpolation Theorem[2,3]

For any Sobolev function $f \in H^k(\Omega, \mathbb{R}), k > m/2$ and $\varepsilon > 0$ there is a polynomial interpolant $Q \in \Pi_{m,n}$ in proper chosen insolvent nodes $P \subseteq \Omega$ such that:

$$\|f - Q\|_{C^0(\Omega,\mathbb{R})} < \varepsilon\,, \quad Q(p) = f(p)\,, \forall\, p \in P$$

If in addition $f \in H^k(\Omega, \mathbb{R}), k > m/2$ is a **Trefethen function** then

$$\|f - Q\|_{C^0(\Omega)} \in \mathcal{O}_\delta(\rho^{-n})$$

While $|P| \in o(n^m)$ is of **sub-exponential size**, we may *lift the curse of dimensionality* for interpolation tasks of Trefethen functions.

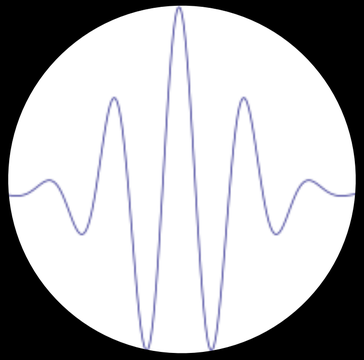
[1] Weierstrass, K. (1885). Über die analytische Darstellbarkeit sogenannter willkürlicher Functionen einer reellen Veränderlichen. *Sitzungsberichte der Königlich Preußischen Akademie der Wissenschaften zu Berlin*, *2*, 633-639.
[2] Hecht, M., Gonciarz, K., Michelfeit, J., Sivkin, V., & Sbalzarini, I. F. (2020). Multivariate Interpolation on Unisolvent Nodes. *arXiv preprint arXiv:2010.10824*.
[3] Trefethen, L. (2017). Multivariate polynomial approximation in the hypercube. *Proceedings of the American Mathematical Society*, *145*(11), 4837-4844.

CASUS
CENTER FOR ADVANCED SYSTEMS UNDERSTANDING
www.casus.science

Newton Interpolation

$$Q_f(x) = \sum_{i=0}^{n} c_i N_i(x)$$

- Numerically accurate & fast $\mathcal{O}(n^2)$
- Convergence to the ground truth $Q_{f,n} \xrightarrow[n\to\infty]{} f$
- Interpolant is easy to understand
- Allows further analysis/computation
- Only in 1D



Spline/Wavelet Interpolation

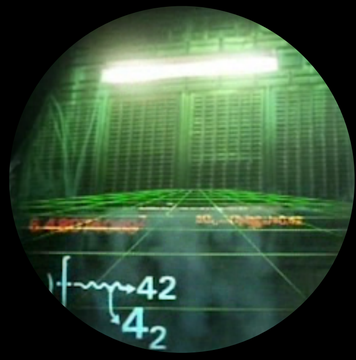
$$Q_f(x) = \sum_{p\in G} c_p \gamma_p(x)$$

- Numerically accurate & fast
- Convergence to ground truth $Q_{f,n} \xrightarrow[n\to\infty]{} f$
- Interpolant can be understood locally
- Allows further analysis/computation
- Feasible in low dimensions

Newton Interpolation

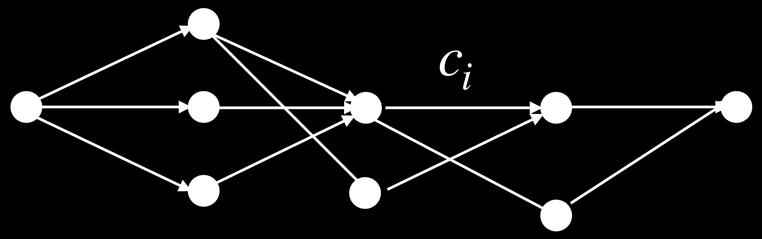
$$Q_f(x) = \sum_{i=0}^{n} c_i N_i(x)$$

- Numerically accurate & fast $\mathcal{O}(n^2)$
- Convergence to the ground truth $Q_{f,n} \xrightarrow[n\to\infty]{} f$
- Interpolant is easy to understand
- Allows further analysis/computation
- Only in 1D



Machine Learning

$$Q_f(x) =$$



- Numerically accurate & fast
- No Convergence to ground truth $Q_{f,n} \nrightarrow f$
- Interpolant is hard to understand
- Hampers further analysis/computation
- Feasible in high dimensions