Lectura 3.1: Redes Neuronales Informadas por Física

Introducción a las redes neuronales informadas por física
Especialización en Inteligencia Artificial - B52024

Outline

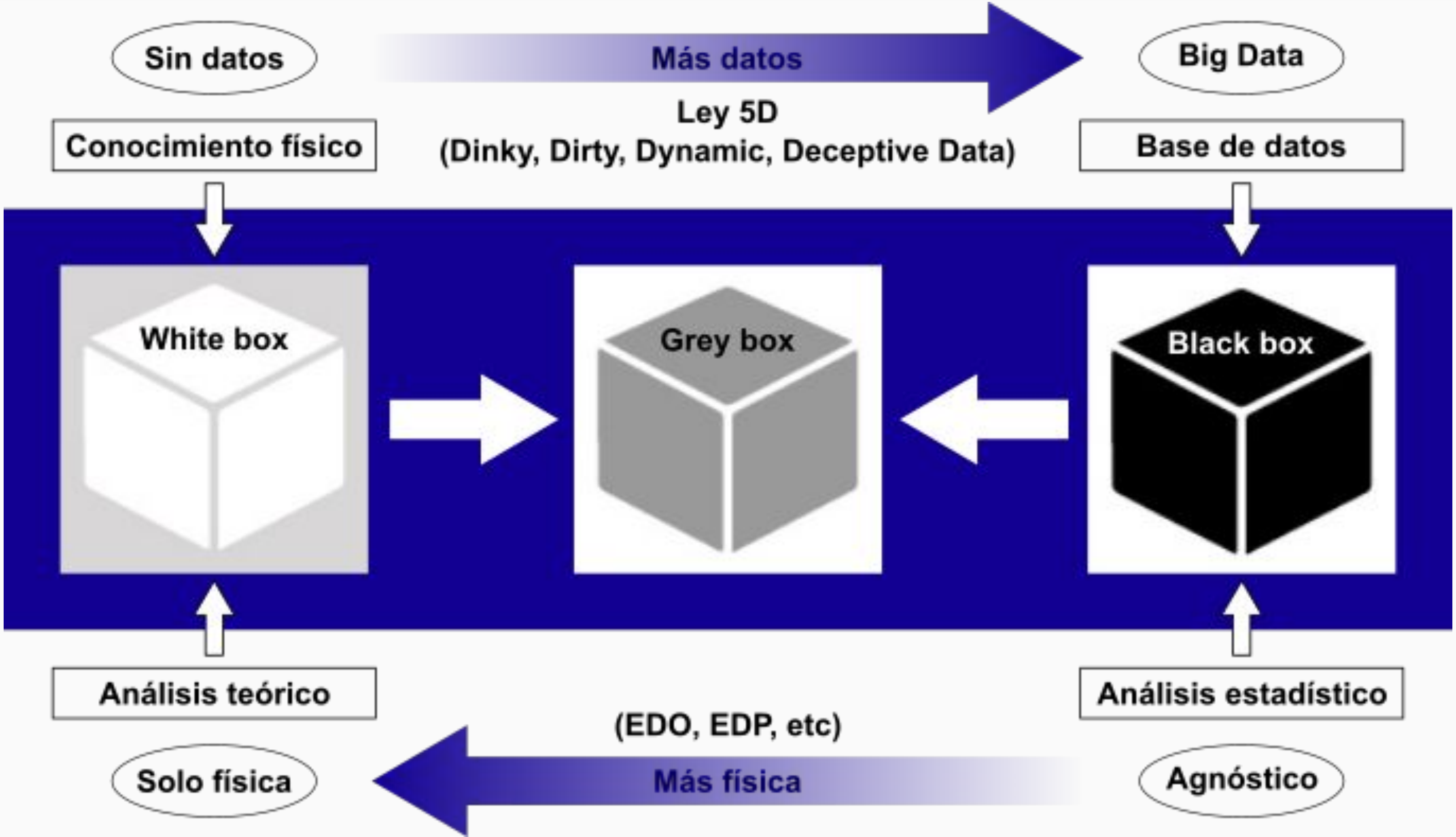
- Fundamentos teóricos
- Construcción de modelos vanilla PINN paso a paso
- Ejemplos de implementación

Paper fundacional

Physics-informed neural networks (PINN): A Deep Learning framework for solving forward and inverse problems involving nonlinear partial differential equations.

M. Raissi, P. Perdikaris, G. E. Karniadakis. *Journal of Computational Physics* 378, 686-707, 2019.

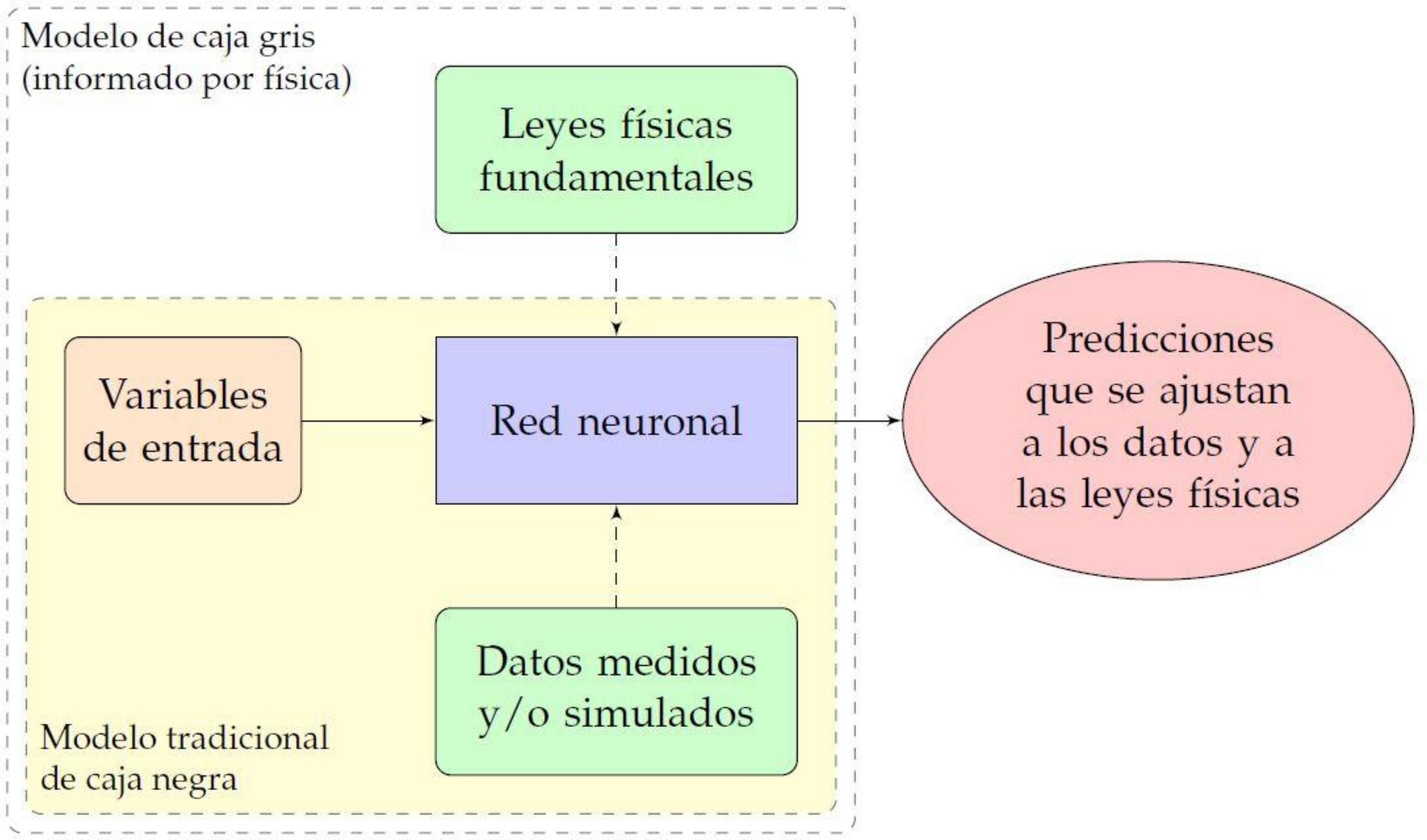
Datos + leyes físicas



Campos de aplicación

- Problemas directos
- Problemas inversos
- Problemas paramétricos
- Descubrimiento de física

Esquema elemental



Construcción de modelos vanilla PINN paso a paso

Planteo abstracto del problema a resolver

- Consideremos el siguiente PVIB:

$$\begin{aligned} u_t(\boldsymbol{x},t) + \mathcal{N}[u](\boldsymbol{x},t) &= 0, \quad \boldsymbol{x} \in \Omega,\ t \in [0,T], \\ \mathcal{B}[u](\boldsymbol{x},t) &= g(\boldsymbol{x},t), \quad \boldsymbol{x} \in \partial\Omega,\ t \in [0,T], \\ \mathcal{I}[u](\boldsymbol{x},0) &= h(\boldsymbol{x}), \quad \boldsymbol{x} \in \Omega \cup \partial\Omega, \end{aligned}$$

cuya solución (desconocida) es: $u : \Omega \times [0,T] \to \mathbb{R}$

- Construimos una red neuronal que aproxime a "u", representada como:

$$\hat{u} = f_{\boldsymbol{\theta}}(\boldsymbol{x},t) \qquad f : \Omega \times [0,T] \to \mathbb{R}$$

donde \theta son los parámetros ajustables de la red.

Residuos de las ED

- Si reemplazamos "u" por su aproximación "f", podemos reescribir el PVIB de la siguiente manera:

$$\mathcal{R}_{\mathcal{N}}(\boldsymbol{x}, t; \boldsymbol{\theta}) = \frac{\partial}{\partial t}[f_{\boldsymbol{\theta}}(\boldsymbol{x}, t)] + \mathcal{N}[f_{\boldsymbol{\theta}}](\boldsymbol{x}, t),$$
$$\mathcal{R}_{\mathcal{B}}(\boldsymbol{x}, t; \boldsymbol{\theta}) = \mathcal{B}[f_{\boldsymbol{\theta}}](\boldsymbol{x}, t) - g(\boldsymbol{x}, t),$$
$$\mathcal{R}_{\mathcal{I}}(\boldsymbol{x}, t; \boldsymbol{\theta}) = \mathcal{I}[f_{\boldsymbol{\theta}}](\boldsymbol{x}, t) - h(\boldsymbol{x})$$

- Los definidos son los "residuos" de las ED correspondientes → En general, distintos de cero
- Si se lograra que sean iguales a cero, la aproximación "f" satisfacería el PVIB… Cómo podemos proceder??

Residuos de las ED como funciones de pérdida

Recordemos la definición de riesgo: $J^*(\boldsymbol{\theta}) = \mathbb{E}_{(\boldsymbol{x},\mathrm{y})\sim p_{\mathrm{data}}} L(f(\boldsymbol{x};\boldsymbol{\theta}), y).$

$$\mathcal{L}^*(\boldsymbol{\theta}) = \underbrace{\int_{\Omega\times[0,T]} |\mathcal{R}_{\mathcal{N}}(\boldsymbol{x},t;\boldsymbol{\theta})|^2 \,\mathrm{d}\Omega\,\mathrm{d}t}_{\mathcal{L}^*_{\mathcal{N}}(\boldsymbol{\theta})} + \underbrace{\int_{\partial\Omega\times[0,T]} |\mathcal{R}_{\mathcal{B}}(\boldsymbol{x},t;\boldsymbol{\theta})|^2 \,\mathrm{d}\Gamma\,\mathrm{d}t}_{\mathcal{L}^*_{\mathcal{B}}(\boldsymbol{\theta})} + \underbrace{\int_{\Omega} |\mathcal{R}_{\mathcal{I}}(\boldsymbol{x},t;\boldsymbol{\theta})|^2 \,\mathrm{d}\Omega}_{\mathcal{L}^*_{\mathcal{I}}(\boldsymbol{\theta})}$$

No es posible resolver las integrales directamente, debemos emplear una aproximación discreta → Cuadraturas

Cuadraturas y puntos de colocación

- La aprox. numérica de integrales se realiza mediante reglas de cuadratura.
- En 1D podemos utilizar reglas de Simpson, Newton-Cotes o cuadraturas de Gauss-Legendre (también en 2D y 3D), que evaluan la integral en grillas regulares:

$$\int_{-1}^{1} f(x)\,dx \approx \sum_{i=1}^{n} w_i f(x_i),$$

Number of points, n	Points, x_i		Weights, w_i	
1	0		2	
2	$\pm\frac{1}{\sqrt{3}}$	±0.57735...	1	
3	0		$\frac{8}{9}$	0.888889...
	$\pm\sqrt{\frac{3}{5}}$	±0.774597...	$\frac{5}{9}$	0.555556...

En problemas de alta dimensionalidad, la aplicación repetida de tales reglas sufre de la CoD.

Cuadraturas y puntos de colocación: método Monte Carlo

- Método no determinístico que implica el muestreo aleatorio de puntos bajo diferentes estrategias: uniforme, estratificado, importancia, etc.

Integral exacta

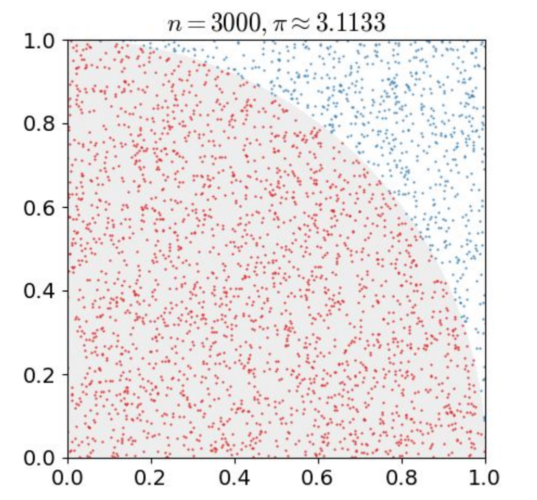
$$I = \int_{\Omega} f(\overline{\mathbf{x}})\, d\overline{\mathbf{x}}$$

Volumen

$$V = \int_{\Omega} d\overline{\mathbf{x}}$$

Muestreo aleatorio

$$\overline{\mathbf{x}}_1, \cdots, \overline{\mathbf{x}}_N \in \Omega,$$



Aproximación de la integral

$$I \approx Q_N \equiv V \frac{1}{N} \sum_{i=1}^{N} f(\overline{\mathbf{x}}_i) = V\langle f \rangle.$$

Estimación del error:

$$\delta Q_N \approx \sqrt{\mathrm{Var}(Q_N)} = V \frac{\sigma_N}{\sqrt{N}}$$

T. de los grandes números: $\lim_{N\to\infty} Q_N = I.$

Residuos evaluados en los puntos de colocación

- De esta manera, los residuos se deben expresar de manera discreta en base a los puntos de colocación muestreados aleatoriamente (muestras i.i.d.):

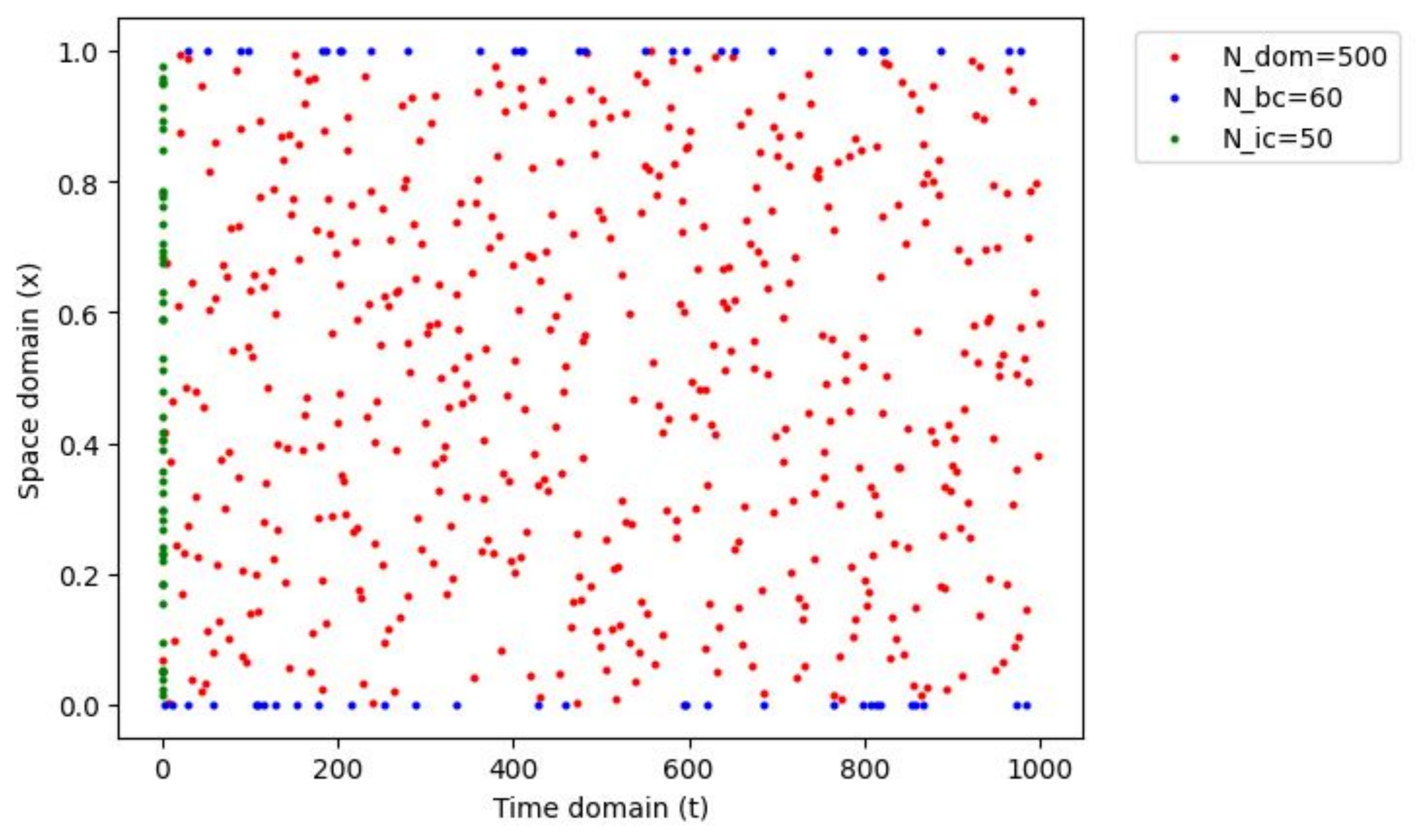
$$\mathcal{R}_{\mathcal{N}}(\boldsymbol{x}_i, t_i; \boldsymbol{\theta}) = \frac{\partial}{\partial t}[f_{\boldsymbol{\theta}}(\boldsymbol{x}_i, t_i)] + \mathcal{N}[f_{\boldsymbol{\theta}}](\boldsymbol{x}_i, t_i), \quad i = 1, \cdots, N_{\mathcal{N}}$$

$$\mathcal{R}_{\mathcal{B}}(\boldsymbol{x}_j, t_j; \boldsymbol{\theta}) = \mathcal{B}[f_{\boldsymbol{\theta}}](\boldsymbol{x}_j, t_j) - g(\boldsymbol{x}_j, t_j), \quad j = 1, \cdots, N_{\mathcal{B}}$$

$$\mathcal{R}_{\mathcal{I}}(\boldsymbol{x}_k, t_k; \boldsymbol{\theta}) = \mathcal{I}[f_{\boldsymbol{\theta}}](\boldsymbol{x}_k, t_k) - h(\boldsymbol{x}_k), \quad k = 1, \cdots, N_{\mathcal{I}}$$

$$X_{\mathcal{N}} = \begin{pmatrix} \boldsymbol{x}_1 & t_1 \\ \boldsymbol{x}_2 & t_2 \\ \vdots & \vdots \\ \boldsymbol{x}_{N_{\mathcal{N}}} & t_{N_{\mathcal{N}}} \end{pmatrix} \quad X_{\mathcal{B}} = \begin{pmatrix} \boldsymbol{x}_1 & t_1 \\ \boldsymbol{x}_2 & t_2 \\ \vdots & \vdots \\ \boldsymbol{x}_{N_{\mathcal{B}}} & t_{N_{\mathcal{B}}} \end{pmatrix} \quad X_{\mathcal{I}} = \begin{pmatrix} \boldsymbol{x}_1 & t_1 \\ \boldsymbol{x}_2 & t_2 \\ \vdots & \vdots \\ \boldsymbol{x}_{N_{\mathcal{I}}} & t_{N_{\mathcal{I}}} \end{pmatrix}$$

Ejemplo de ubicación de puntos de colocación



Función de pérdida empírica de PINN

- Residuos definidos en base a los puntos de colocación en cada término:

$$\mathcal{L}_j^*(\boldsymbol{\theta}) \approx \frac{1}{N_j}\sum_{i=1}^{N_j} \mathcal{R}_j(\boldsymbol{x}_i^j, t_i^j; \boldsymbol{\theta})^2 := \mathcal{L}_j(\theta), \quad j : \{\mathcal{N}, \mathcal{B}, \mathcal{I}\}$$

- Notar que también se podría incluir un término de ajuste de datos rotulados:

$$\mathcal{L}_{\mathcal{D}}(\boldsymbol{\theta}) = \frac{1}{N_{\mathcal{D}}}\sum_{i=1}^{N_{\mathcal{D}}} \left[f_{\boldsymbol{\theta}}(\boldsymbol{x}_i, t_i) - u(x_i, t_i)\right]^2$$

$$X_{\text{data}} = \left(\begin{array}{cc|c} \boldsymbol{x}_1 & t_1 & u_1 \\ \boldsymbol{x}_2 & t_2 & u_2 \\ \vdots & \vdots & \vdots \\ \boldsymbol{x}_{N_{\mathcal{D}}} & t_{N_{\text{data}}} & u_{N_{\mathcal{D}}} \end{array}\right)$$

Función de pérdida empírica de PINN

- Aprendizaje supervisado (black-box model):

$$\mathcal{L}(\boldsymbol{\theta}) = \mathcal{L}_{\mathcal{D}}(\boldsymbol{\theta})$$

- Aprendizaje no supervisado (PINN **sin** datos rotulados → white-box model):

$$\mathcal{L}(\boldsymbol{\theta}) = \mathcal{L}_{\mathcal{N}}(\boldsymbol{\theta}) + \lambda_{\mathcal{B}}\mathcal{L}_{\mathcal{B}}(\boldsymbol{\theta}) + \lambda_{\mathcal{I}}\mathcal{L}_{\mathcal{I}}(\boldsymbol{\theta})$$

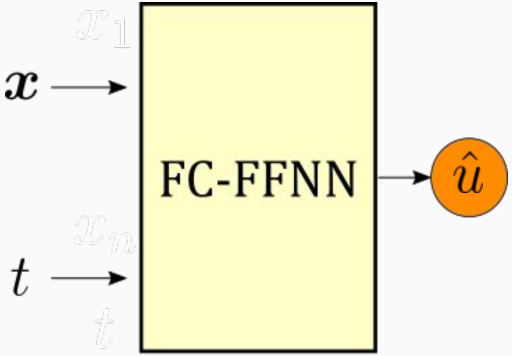
- Aprendizaje semi-supervisado (PINN **con** datos rotulados → grey-box model):

$$\mathcal{L}(\boldsymbol{\theta}) = \mathcal{L}_{\mathcal{N}}(\boldsymbol{\theta}) + \lambda_{\mathcal{B}}\mathcal{L}_{\mathcal{B}}(\boldsymbol{\theta}) + \lambda_{\mathcal{I}}\mathcal{L}_{\mathcal{I}}(\boldsymbol{\theta}) + \lambda_{\mathcal{D}}\mathcal{L}_{\mathcal{D}}(\boldsymbol{\theta})$$

Red neuronal feed-forward

$$X_{\mathcal{N}} = \begin{pmatrix} \boldsymbol{x}_1 & t_1 \\ \boldsymbol{x}_2 & t_2 \\ \vdots & \vdots \\ \boldsymbol{x}_{N_{\mathcal{N}}} & t_{N_{\mathcal{N}}} \end{pmatrix} \qquad X_{\mathcal{I}} = \begin{pmatrix} \boldsymbol{x}_1 & t_1 \\ \boldsymbol{x}_2 & t_2 \\ \vdots & \vdots \\ \boldsymbol{x}_{N_{\mathcal{I}}} & t_{N_{\mathcal{I}}} \end{pmatrix}$$

$$X_{\mathcal{B}} = \begin{pmatrix} \boldsymbol{x}_1 & t_1 \\ \boldsymbol{x}_2 & t_2 \\ \vdots & \vdots \\ \boldsymbol{x}_{N_{\mathcal{B}}} & t_{N_{\mathcal{B}}} \end{pmatrix} \qquad X_{\text{data}} = \left(\begin{array}{cc:c} \boldsymbol{x}_1 & t_1 & u_1 \\ \boldsymbol{x}_2 & t_2 & u_2 \\ \vdots & \vdots & \vdots \\ \boldsymbol{x}_{N_{\mathcal{D}}} & t_{N_{\text{data}}} & u_{N_{\mathcal{D}}} \end{array}\right)$$



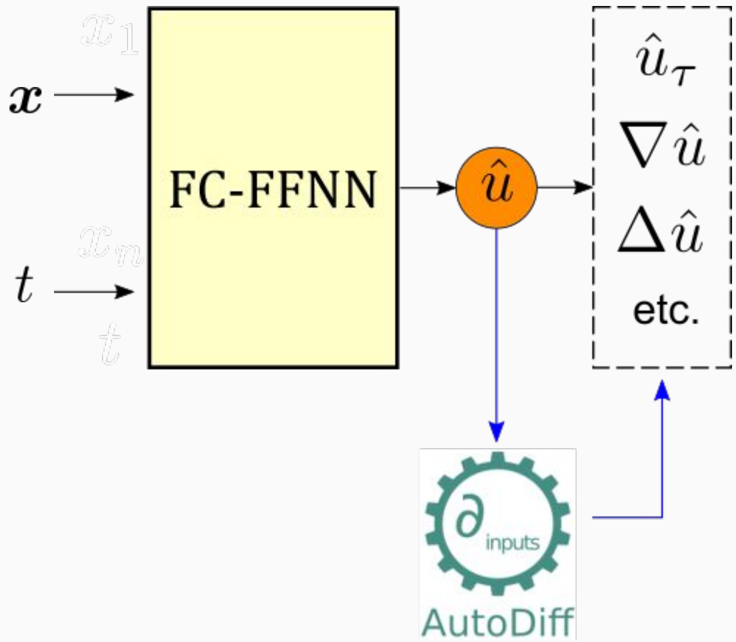
$$\hat{u} = f_{\boldsymbol{\theta}}(\boldsymbol{x}, t)$$

Cómputo de derivadas - Autodiff

$$\mathcal{N}[f_{\boldsymbol{\theta}}(\boldsymbol{x}_i, t_i)] \rightarrow \hat{u}_t,\ \hat{u}_{tt},\ \nabla\hat{u},\ \Delta\hat{u},\ \cdots$$

$$\mathcal{B}[f_{\boldsymbol{\theta}}(\boldsymbol{x}_i, t_i)] \rightarrow \nabla\hat{u},\ \Delta\hat{u},\ \cdots$$

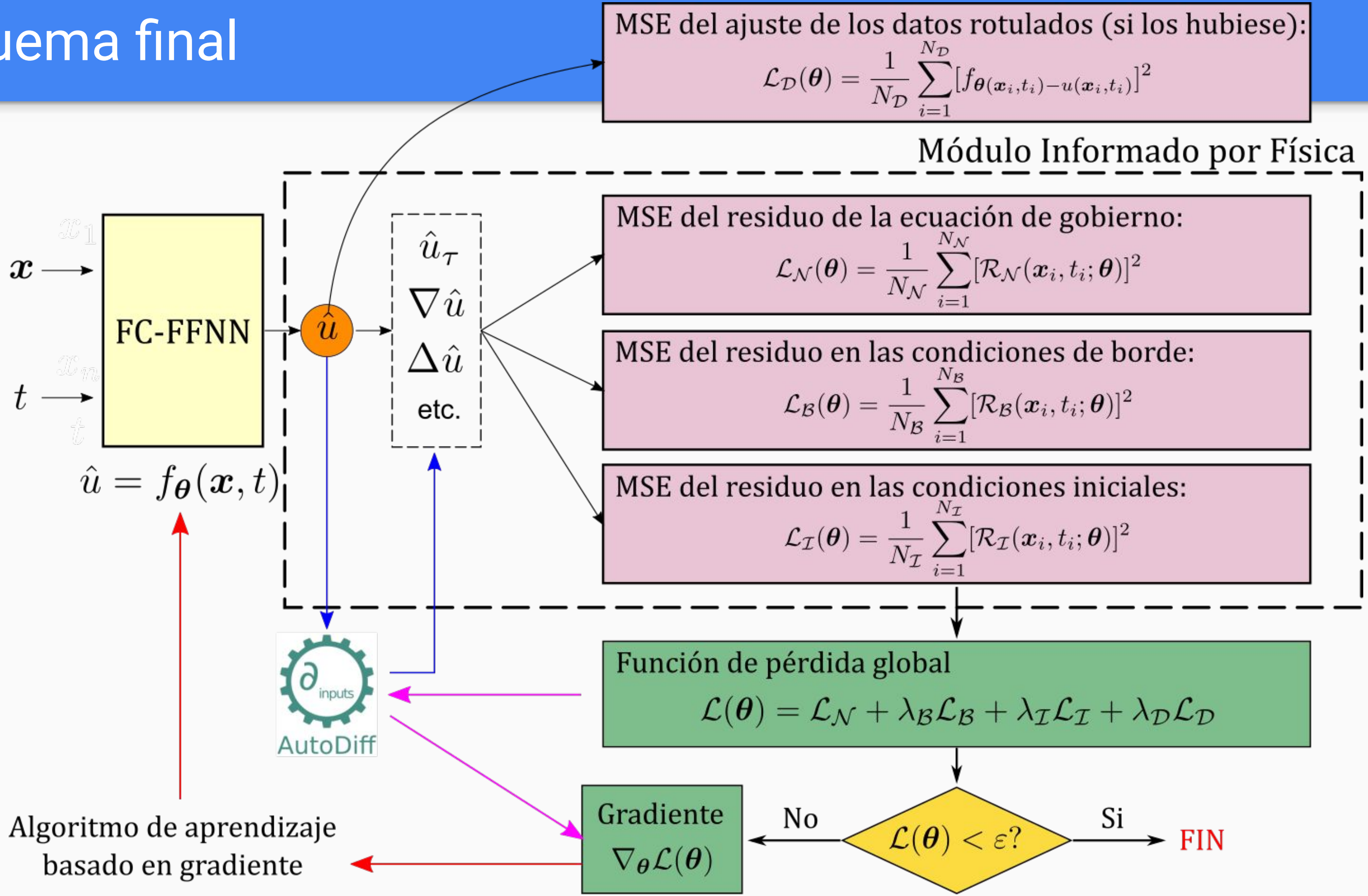
$$\mathcal{I}[f_{\boldsymbol{\theta}}(\boldsymbol{x}_i, t_i)] \rightarrow \hat{u}_t,\ \cdots$$



Autograd:

PyTorch: torch.autograd.grad(X, u, …)
TF: tape.gradient(u, X, …)

Esquema final



Algunos ejemplos de aplicación

Ecuaciones a resolver via PINN (ver en colab)

Ec. del calor 1D sin fuente térmica en régimen transiente:

$$\frac{\partial T}{\partial t} = \alpha \frac{\partial^2 T}{\partial x^2}, \quad x \in [0, L],\ t \in (0, t_f]$$
$$T(0,t) = T(L,t) = T_{\mathrm{bc}}, \quad t \in (0, t_f]$$
$$T(x,0) = T_0 \sin(\pi x / L) + T_{\mathrm{bc}}, \quad x \in [0, L]$$

Ecuación difusiva (lineal)

Ec. de Burgers viscosa 1D sin fuente en régimen transiente:

$$\frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} - \nu \frac{\partial^2 u}{\partial x^2} = 0, \quad x \in [-1, 1],\ t \in (0, t_f]$$
$$u(0,x) = -\sin(\pi x), \quad x \in [-1, 1]$$
$$u(-1,t) = u(1,t) = 0, \quad t \in (0, t_f]$$

Ecuación de convección-difusión (no lineal)

Visualización del loss landscape

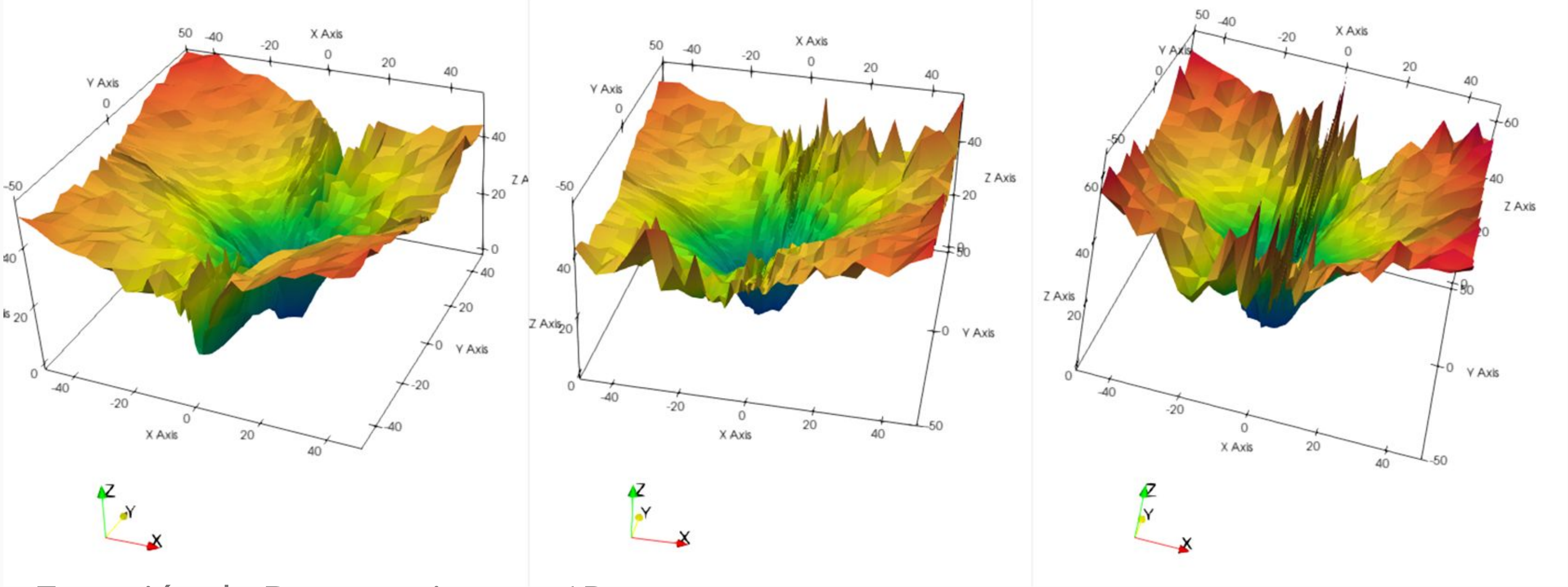
¿Cómo visualizar una gráfica de una función dependiente de N>>2 variables?

Receta:

- Fijar los valores de \theta → Punto fijo \theta*
- Elegir dos vectores dirección \delta y \eta (al azar o mediante PCA, por ej.)
- Construir y graficar la función $f(\alpha, \beta) = L\left(\theta^* + \alpha\delta + \beta\eta\right)$

Visualización del loss landscape

$\nu = 10^{-8}$ $\nu = 10^{-3}$ $\nu = 10^{-2}$



Ecuación de Burgers viscosa 1D

¿Dudas o preguntas?