

Physics-Informed Neural Networks

Taller teórico-práctico



David Ortiz-Puerta, Tabita Catalán, Tomás Banduc, Francisco Sahli

Universidad de Valparaíso
Pontificia Universidad Católica de Chile
Millennium Institute for intelligent Healthcare Engineering, ihealth

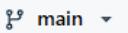
10 de enero, 2025



PONTIFICIA
UNIVERSIDAD
CATÓLICA
DE CHILE

 ihealth-pinns-summer-school Private

 Unwatch 1

 main  1 Branch  0 Tags

 Go to file 

 Add file 

 Code 

 dortiz5	Update README.md	d0b5369 · 9 minutes ago	 13 Commits
 data	Add files via upload	2 months ago	
 notebooks	Readme update	11 minutes ago	
 slides	Readme update	11 minutes ago	
 solutions	Add files via upload	2 months ago	
 LICENSE	Add files via upload	2 months ago	
 README.md	Update README.md	9 minutes ago	
 pinns-tutorial.yml	Add files via upload	2 months ago	

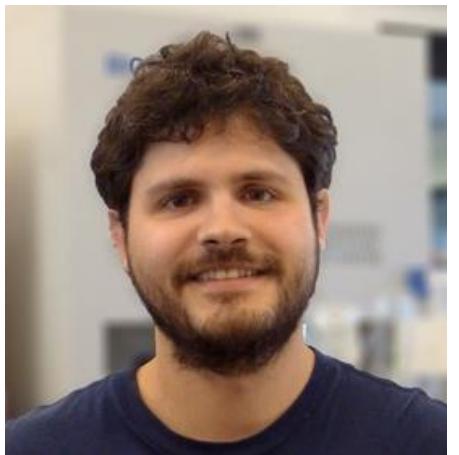
 README  MIT license  

Redes neuronales artificiales informadas por la física (PINNs)

Este repositorio contiene material para el taller sobre redes neuronales informadas por la física (PINNs) de la escuela de verano del Instituto Milenio [iHEALTH](#).



Taller teórico: 1.5 horas
Taller práctico: 2 horas



David Ortiz-Puerta

david.ortiz@uv.cl



Tabita Catalán

tabicm.nhg@gmail.com



Tomás Banduc

tomas.banduc.m@gmail.com



Francisco Sahli

fsahlic@uc.cl



Contenido del taller

1. Motivación
2. Aprendizaje de máquinas: *deep learning*
3. Modelos físicos: problemas directos e inversos
4. Un puente entre los modelos físicos y el aprendizaje
máquinas: teorema de aproximación universal
5. Redes neuronales informadas por la física
6. Pros y contras
7. Aplicaciones

Physics informed neural networks (PINNs)

Redes neuronales artificiales...
(Aprendizaje de máquinas)

... informadas por la física
(Modelos Matemáticos)

Physics informed neural networks (PINNs)

Redes neuronales artificiales...
(Aprendizaje de máquinas)

... informadas por la física
(Modelos Matemáticos)

Scopus query: “Machine learning” = 749,930 document results
Documentos publicados en el 2023: 159.206

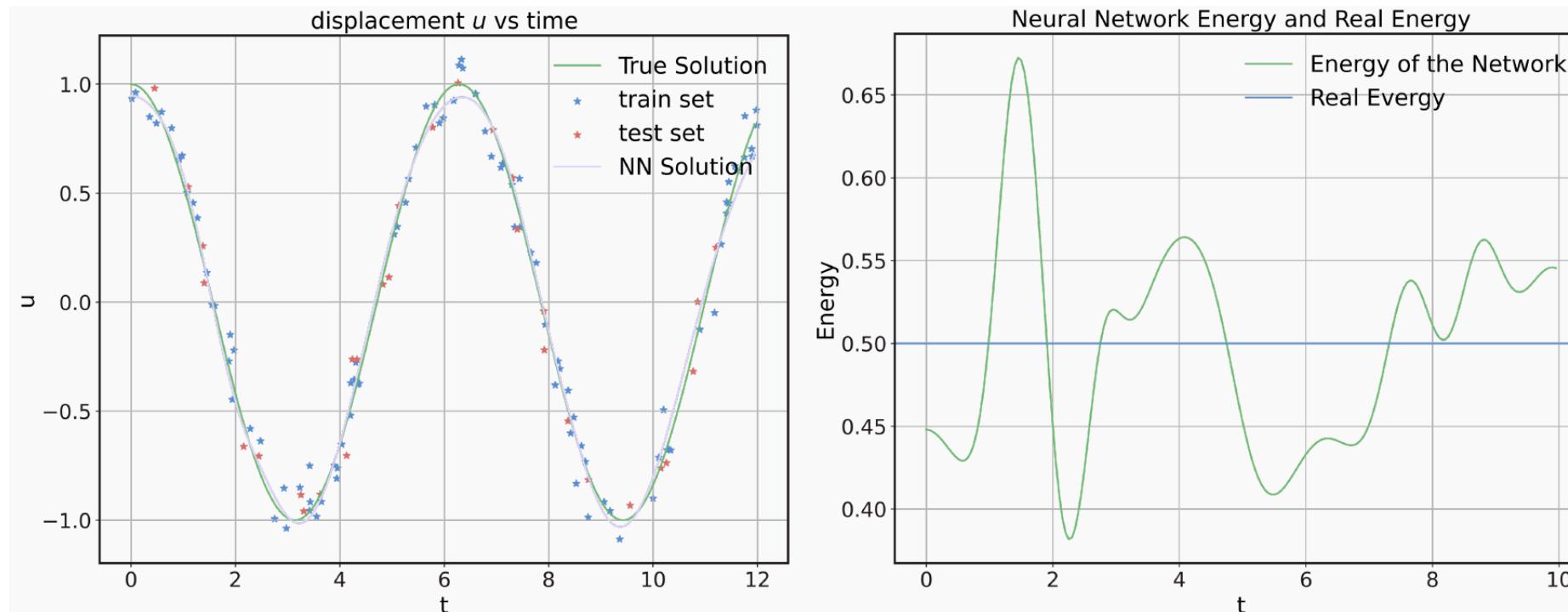
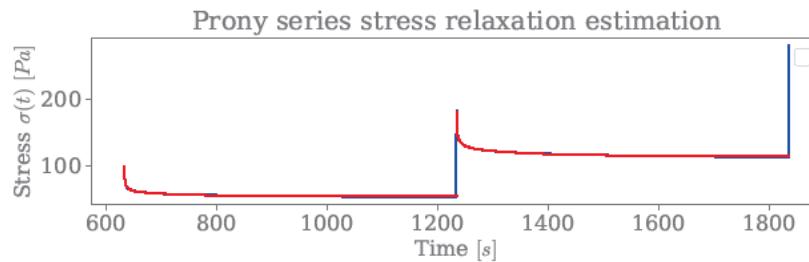
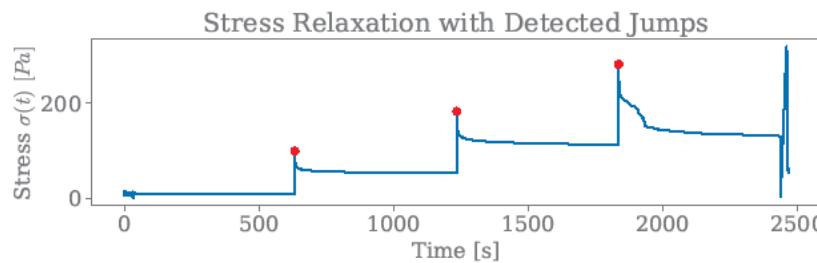


Imagen tomada del curso de
PINNs del profesor Pavlos
Protopapas, Harvard

¡No siempre funciona bien en modelos físicos!

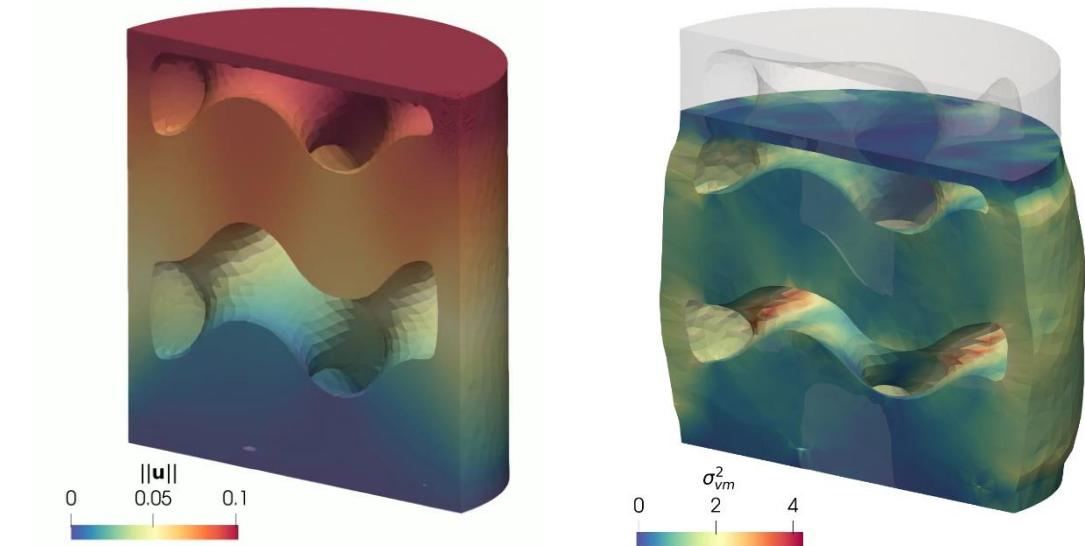
Physics informed neural networks (PINNs)

Redes neuronales artificiales...
(Aprendizaje de máquinas)



... informadas por la física
(Modelos Matemáticos)

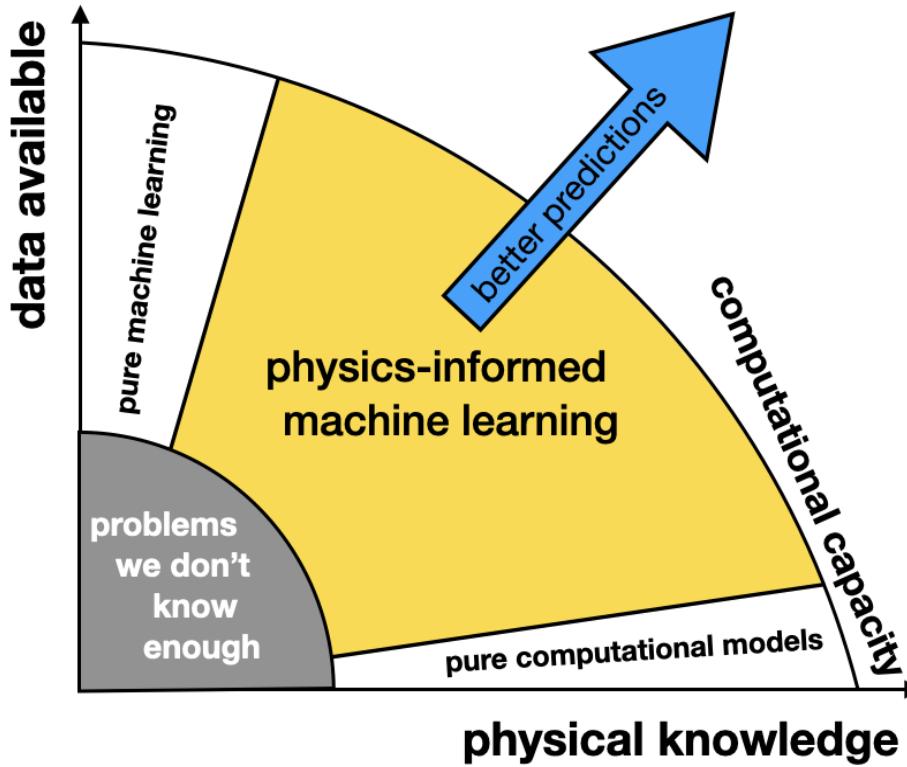
$$\Pi = \int_{\Omega} \psi(u) dx - \int_{\Omega} B \cdot u dx - \int_{\partial\Omega} T \cdot u ds$$
$$\min_{u \in V} \Pi$$



¡Integrar mediciones puede ser desafiante!

Combinación de modelos físicos y datos

¿cómo podemos sacar provecho de lo mejor de ambas áreas?



Aprendizaje de máquinas: *deep learning*

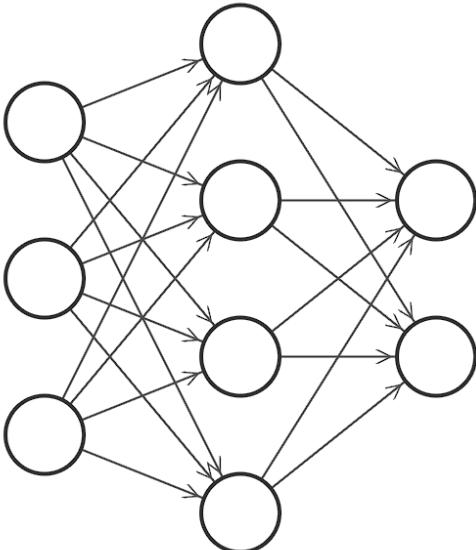
Aprendizaje de máquinas



Imagen tomada de internet

Entrada

x_i



Modelo

$NN(x_i, \theta)$

Probabilidad:
70% Muffin
30% chihuahua

Salida

$y_i = NN(x_i, \theta)$

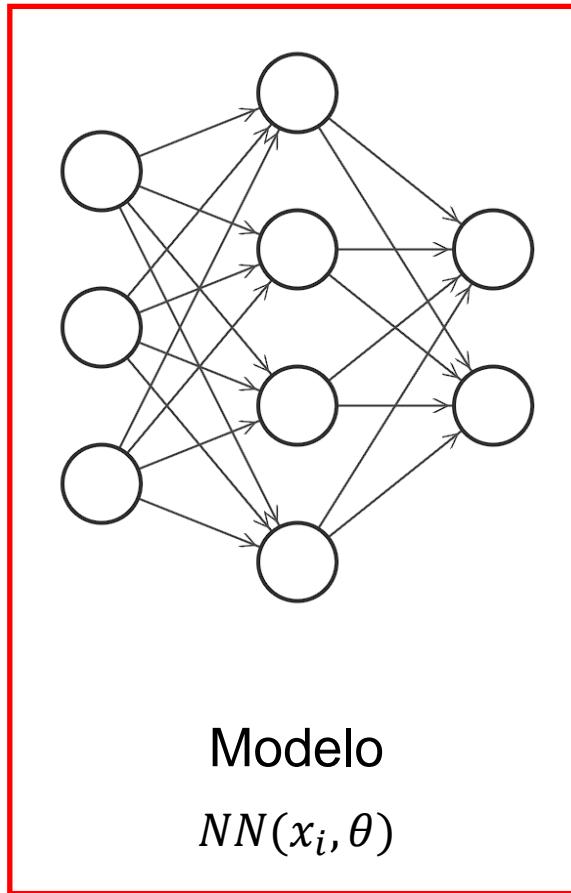
Aprendizaje de máquinas



Imagen tomada de internet

Entrada

x_i



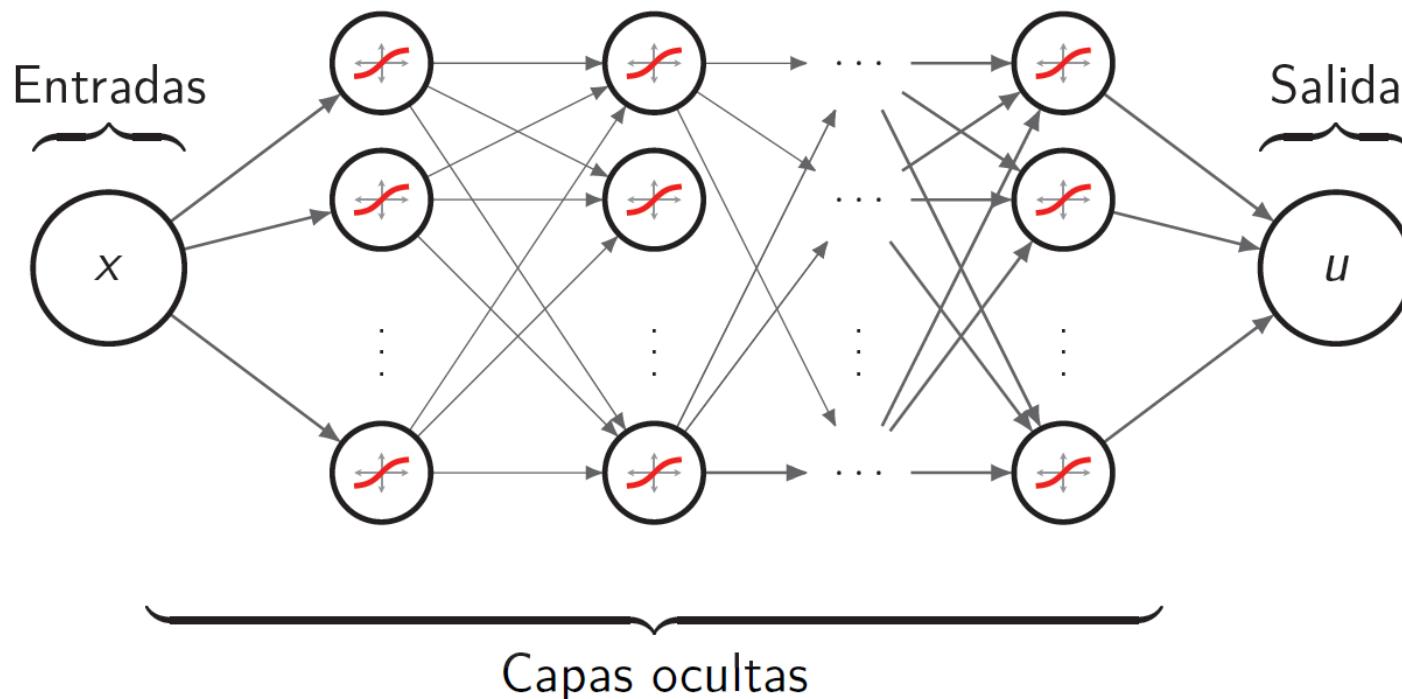
Probabilidad:
70% Muffin
30% chihuahua

Salida

$y_i = NN(x_i, \theta)$

Aprendizaje de máquinas: *deep learning*

Redes neuronales: modelos computacionales



Década de los 50's

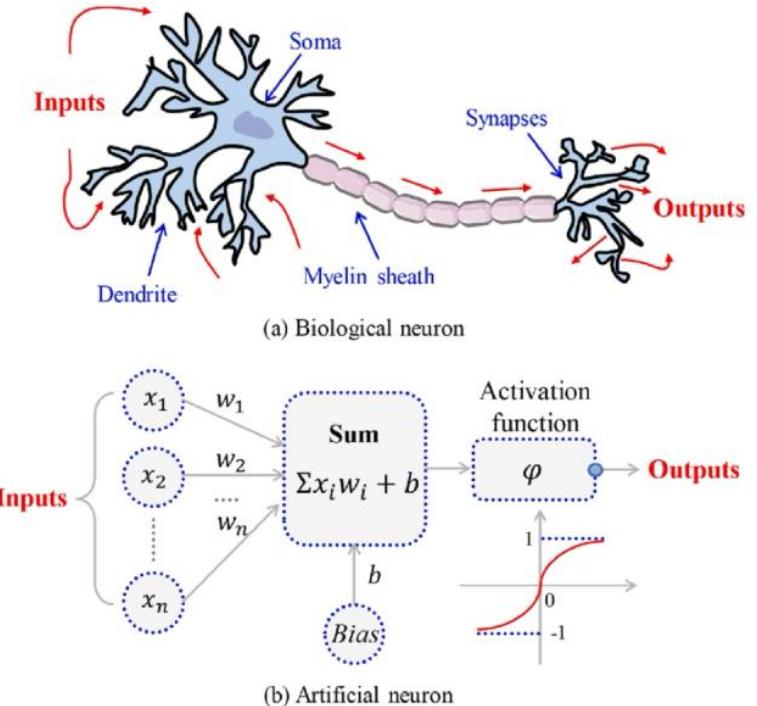
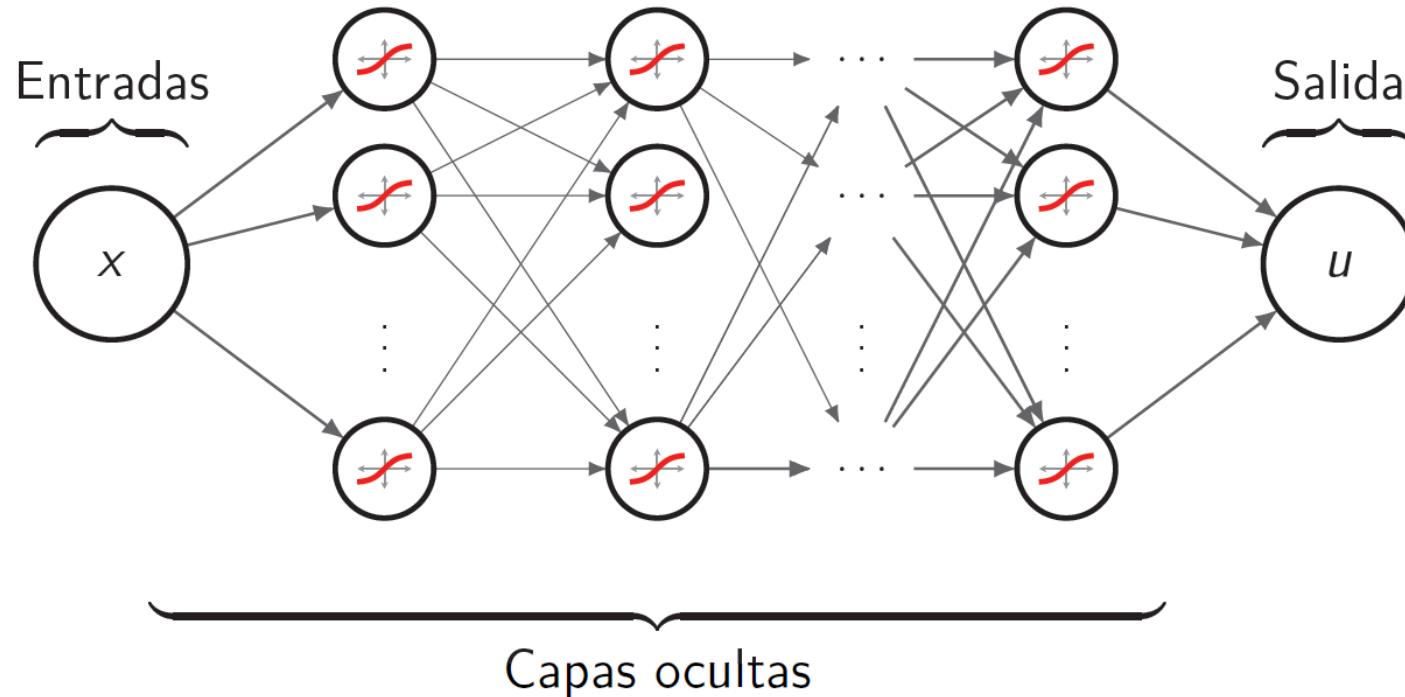


Imagen tomada de internet ([link](#))

Aprendizaje de máquinas: *deep learning*

Redes neuronales: modelos computacionales

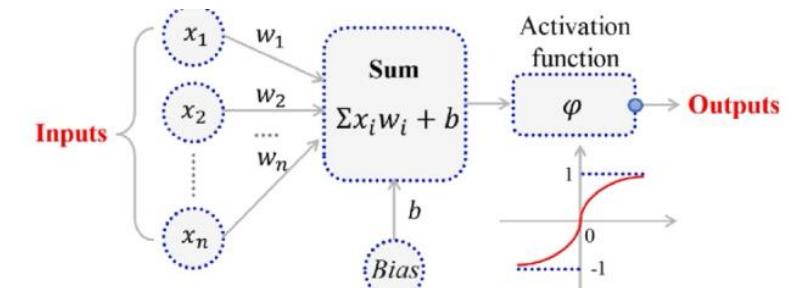


Funciones compuestas

$$f(\mathbf{x}; \boldsymbol{\theta}) = f_n(\dots f_2(f_1(\mathbf{x}; \boldsymbol{\theta}_1); \boldsymbol{\theta}_2); \dots; \boldsymbol{\theta}_n)$$

Pesos y sesgos, función de activación no lineal

$$\begin{aligned}\mathbf{h}_i &= f_i(\mathbf{h}_{i-1}; \boldsymbol{\theta}_i = \{\mathbf{W}_i, \mathbf{b}_i\}) \\ &= g(\mathbf{W}_i^T \mathbf{h}_{i-1} + \mathbf{b}_i)\end{aligned}$$

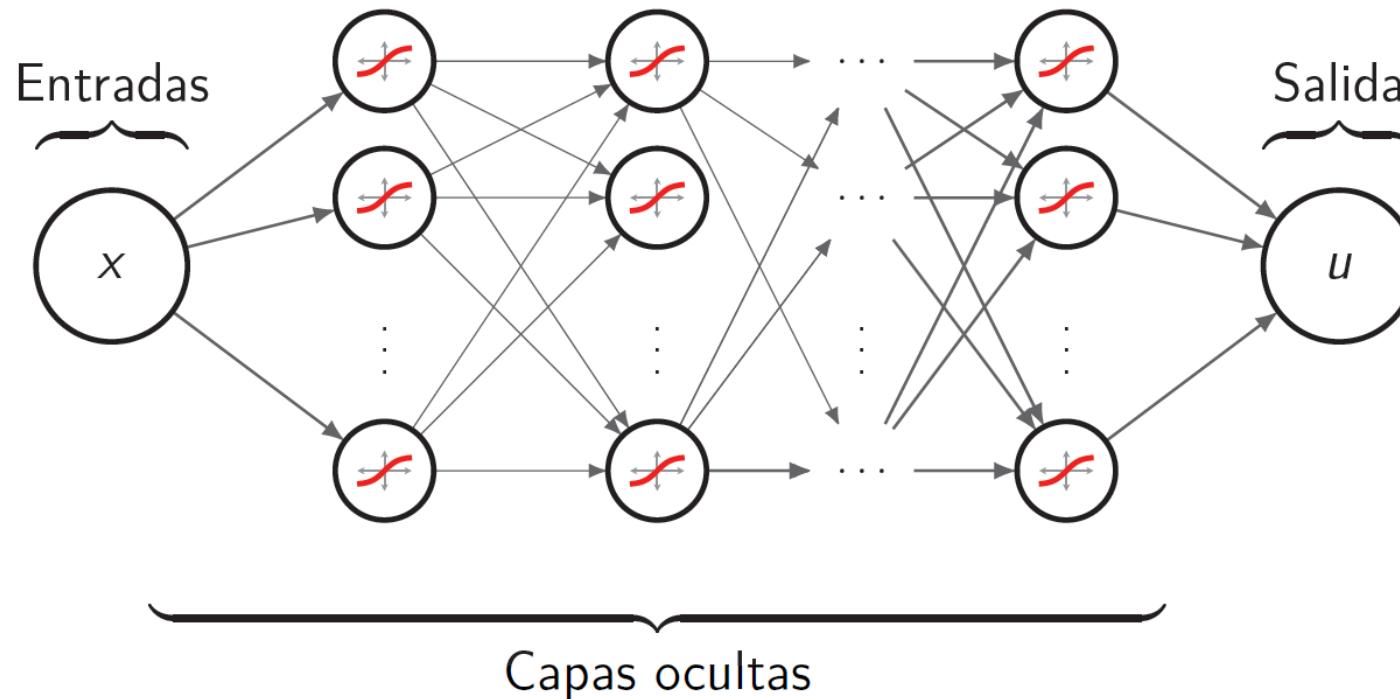


(b) Artificial neuron

Imagen tomada de internet ([link](#))

Aprendizaje de máquinas: *deep learning*

Redes neuronales: modelos computacionales



Función de pérdida

$$\mathcal{L}_{MSE}(\theta; \{x_i, y_i\}_{i=1}^N) = \frac{1}{N} \sum_i^N (y_i - f(x_i; \theta))^2$$

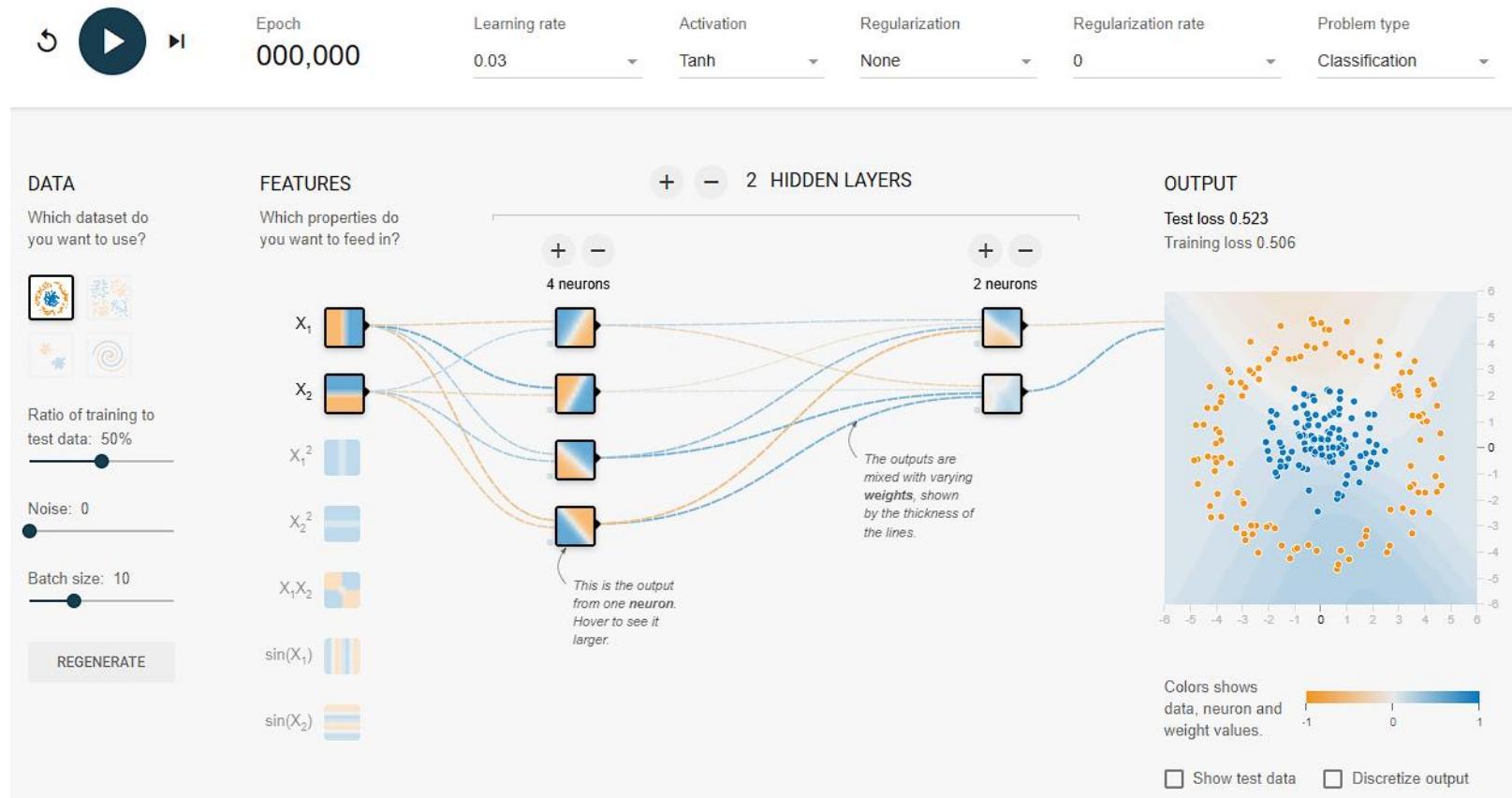
Método de entrenamiento:

Descenso de gradiente

Retropropagación

$$\theta \leftarrow \theta - \epsilon \nabla_{\theta} \mathcal{L}_{MSE}$$

Aprendizaje de máquinas: *deep learning*

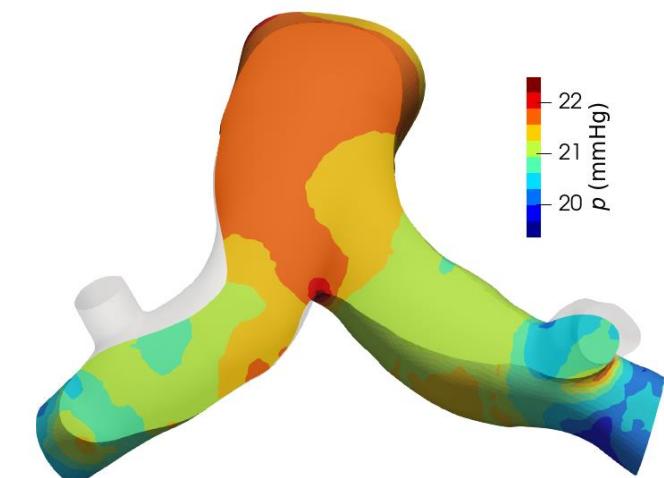
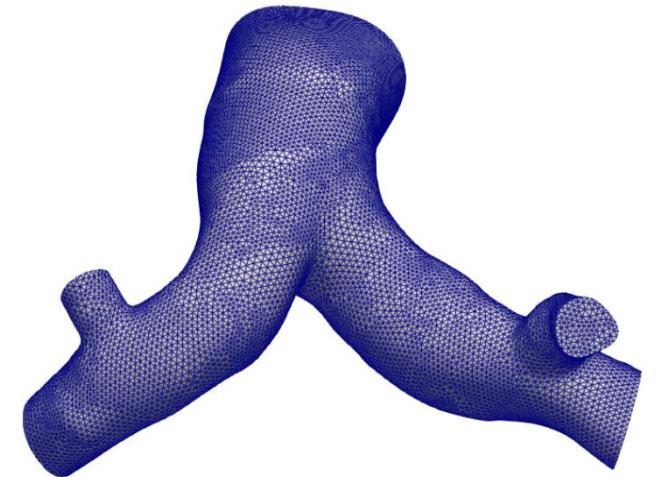
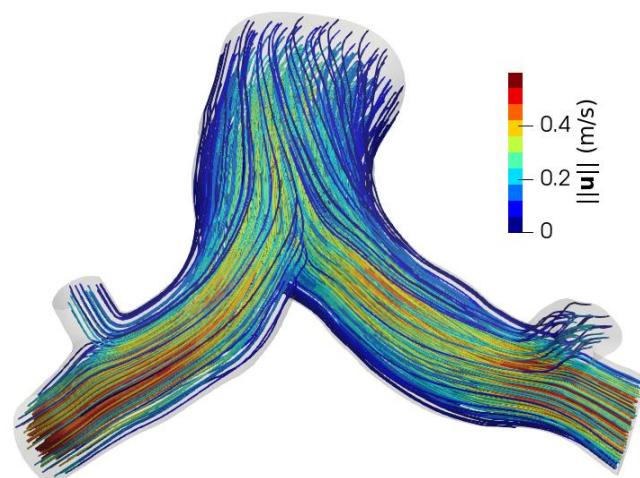


Modelos físicos: problemas directos e inversos

Modelos físicos (matemáticos)

- Desarrollar simulaciones computacionales
- Predecir el comportamiento futuro
- Experimentación in-silico
- ***Desarrollo de la medicina de precisión***

$$\mathbf{u}^* \cdot \nabla \mathbf{u}^* - \frac{1}{Re} \Delta \mathbf{u}^* + \nabla p^* = \mathbf{0}$$
$$\nabla \cdot \mathbf{u}^* = 0$$
$$BC : \quad \mathbf{u}^* = \mathbf{0}, \quad \frac{1}{Re} \frac{\partial \mathbf{u}^*}{\partial \mathbf{n}} - \mathbf{n} p^* = \mathbf{0}$$



Conceptos clave sobre modelos de ecuaciones diferenciales

- **Función:** $y = f(x)$

- **Derivada:** $\frac{dy}{dx} = f'(x)$

- **Derivada Parcial:**

Nombre	Descripción
$\mathbf{u}(x, y, z)$	Desplazamientos (Mecánica)
$T(t, x, y, z)$	Evolución temperatura (Transferencia de calor)
$c(t, x, y)$	Concentración (Química y biología)
$V(t)$	Potencial de acción (ECG, fisiología)

$$\frac{\partial u(x_1, \dots, x_i, \dots, x_n)}{\partial x_i} = \lim_{\Delta x \rightarrow 0} \frac{u(x_1, \dots, x_i + \Delta x, \dots, x_n) - u(x_1, \dots, x_i, \dots, x_n)}{\Delta x}$$

- **Operadores diferenciales:**

Gradiente

$$\nabla(\cdot) = \left[\frac{\partial(\cdot)}{\partial x_1}, \frac{\partial(\cdot)}{\partial x_2}, \dots, \frac{\partial(\cdot)}{\partial x_n} \right]$$

Divergencia

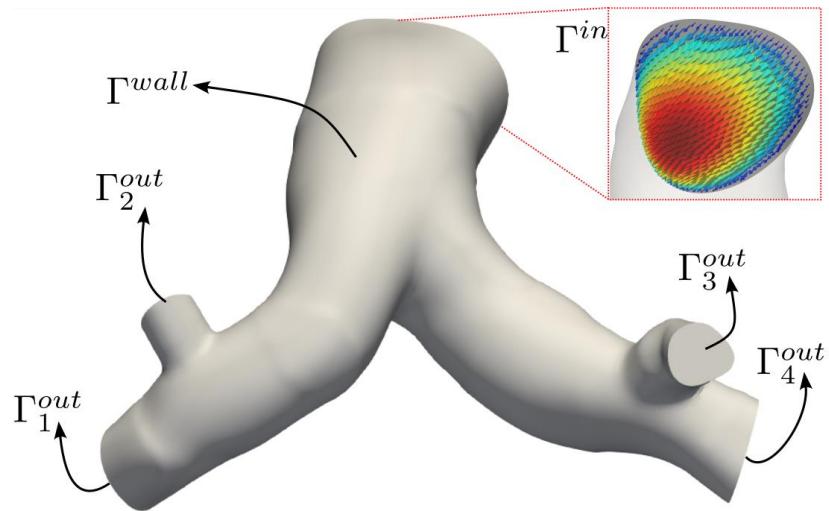
$$\operatorname{div}(\mathbf{a}) = \nabla \cdot \mathbf{a} = \frac{\partial a_1}{\partial x_1} + \dots + \frac{\partial a_n}{\partial x_n}$$

Laplaciano

$$\Delta(\cdot) = \nabla \cdot \nabla(\cdot) = \frac{\partial^2(\cdot)}{\partial x_1} + \dots + \frac{\partial^2(\cdot)}{\partial x_n}$$

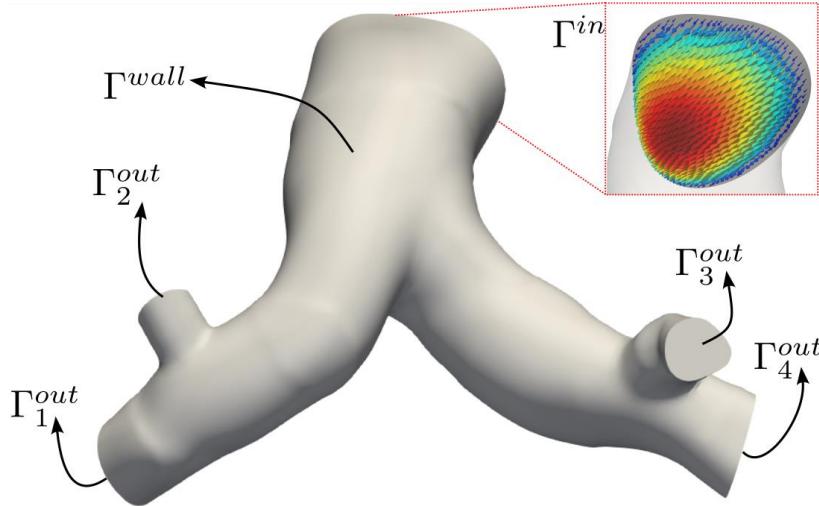
Ecuaciones diferenciales

- Dominio o geometría



Ecuaciones diferenciales

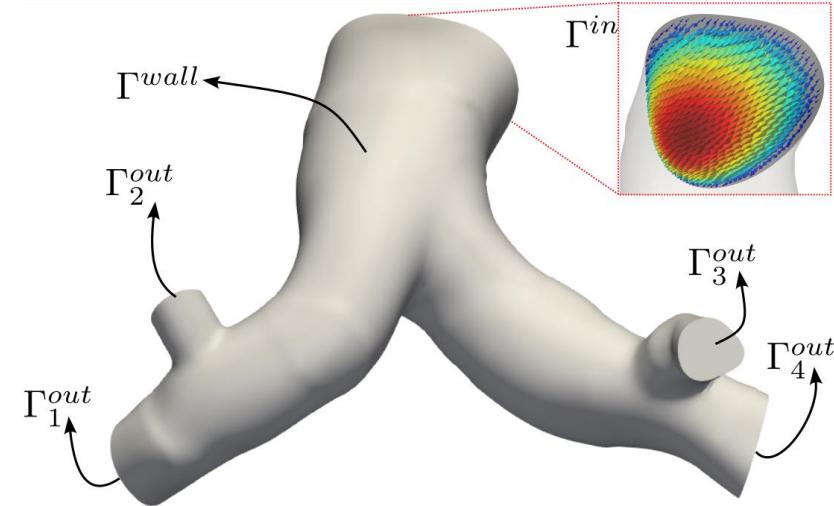
- Dominio o geometría
- Derivadas sobre una función desconocida



	Lineal	No Lineal
EDO	$m \frac{d^2x}{dt^2} + kx = 0$ (Sistema Masa-Resorte)	$\frac{dv}{dt} = v - \frac{v^3}{3} - w + I$ (Modelo de FitzHugh-Nagumo)
EDP	$\frac{\partial u}{\partial t} = \nabla \cdot (D \nabla u)$ (Ecuación de Difusión Anisotrópica)	$\frac{\partial u}{\partial t} = D_u \nabla^2 u - uv^2 + F(1-u)$ $\frac{\partial v}{\partial t} = D_v \nabla^2 v + uv^2 - (F+k)v$ (Modelo de Gray-Scott)

Ecuaciones diferenciales

- Dominio o geometría
- Derivadas sobre una función desconocida
- Condiciones iniciales y de frontera



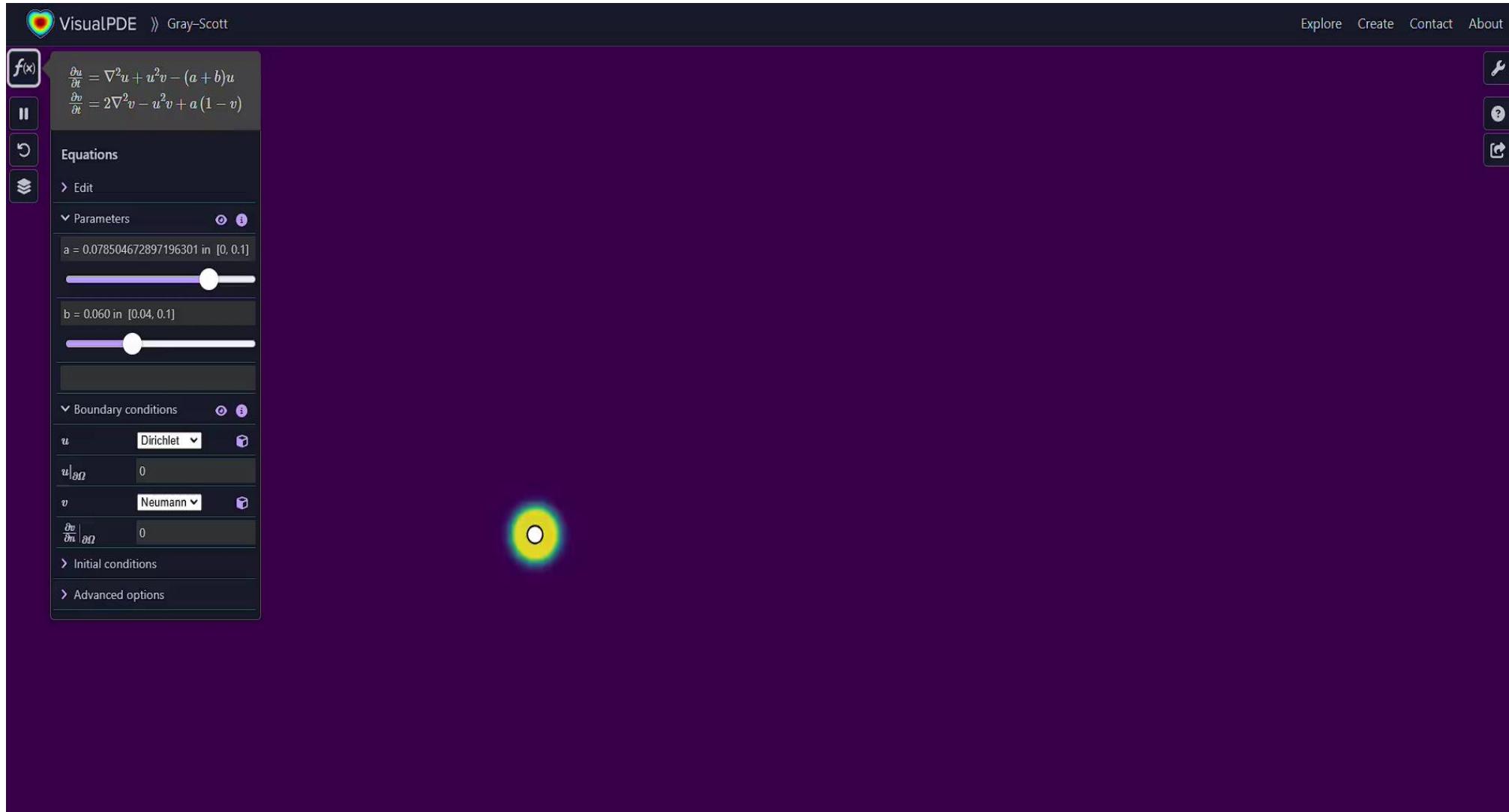
$$\frac{\partial u}{\partial t} = D_u \nabla^2 u - uv^2 + F(1-u) \quad \text{en } \Omega \times (0, T]$$

$$\frac{\partial v}{\partial t} = D_v \nabla^2 v + uv^2 - (F+k)v \quad \text{en } \Omega \times (0, T]$$

$$\begin{cases} u(x, t) = g(x, t) & (\text{Dirichlet}) \text{ en } \partial\Omega_D \times (0, T] \\ \frac{\partial u}{\partial n} = h(x, t) & (\text{Neumann}) \text{ en } \partial\Omega_N \times (0, T] \\ \frac{\partial u}{\partial n} + \alpha u = r(x, t) & (\text{Robin}) \text{ en } \partial\Omega_R \times (0, T] \\ u(x, 0) = u_0(x) & (\text{Condición Inicial}) \text{ en } \Omega \end{cases}$$

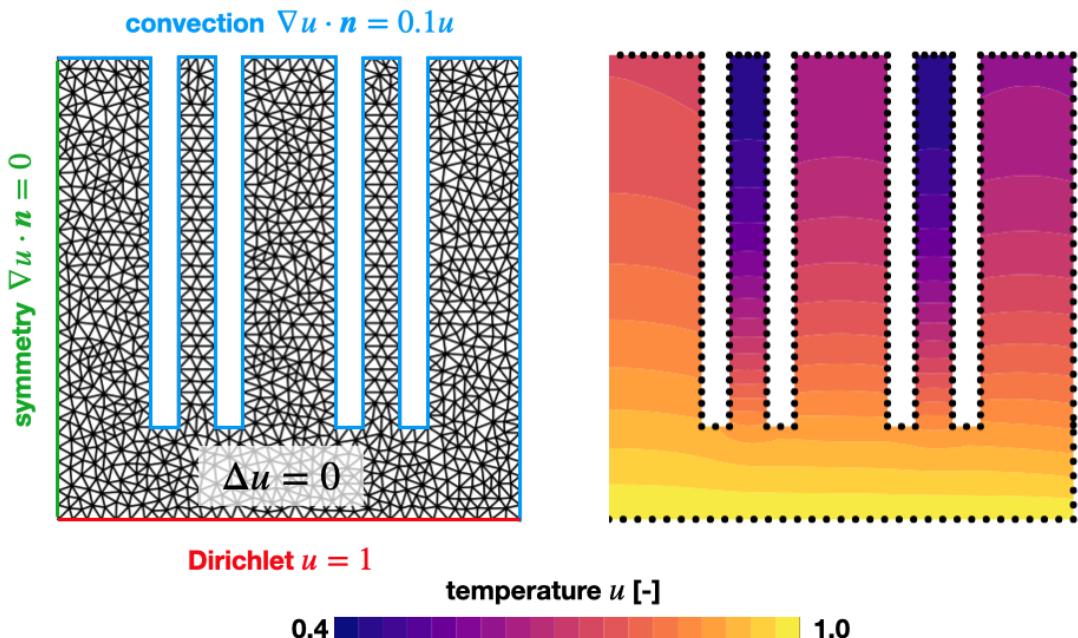
Muchos problemas con ED requieren soluciones numéricas

Modelo de Gray-Scott



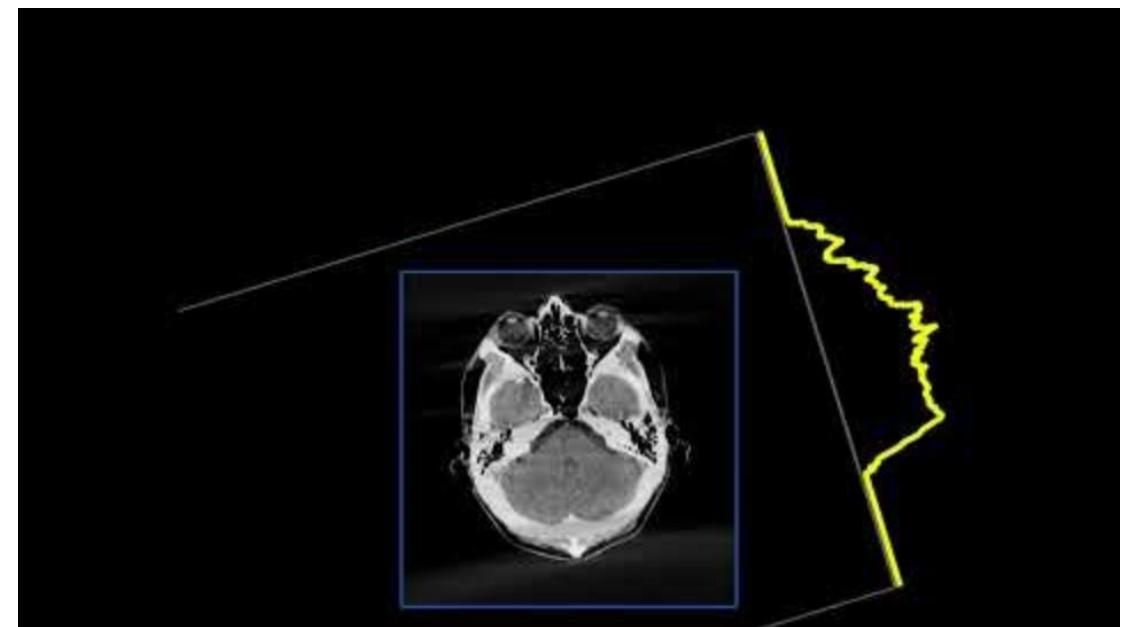
Problemas directos

De las causas a los efectos
(observaciones)



Problemas inversos

De los efectos (observaciones) a las causas



Reconstrucción de tomografía axial computarizada (TAC).
Por Samuli Siltanen

Un puente entre los modelos físicos y el aprendizaje máquinas: teorema de aproximación universal

Teorema de aproximación universal

Una red neuronal con una sola capa oculta y una función de activación no constante, acotada y continua σ , puede aproximar cualquier función continua f definida en un dominio cerrado y acotado Ω con una precisión arbitraria

Teorema de aproximación universal

Para cualquier función continua $f \in \mathcal{C}(\Omega)$ y $\epsilon > 0$, existe una red neuronal de la forma

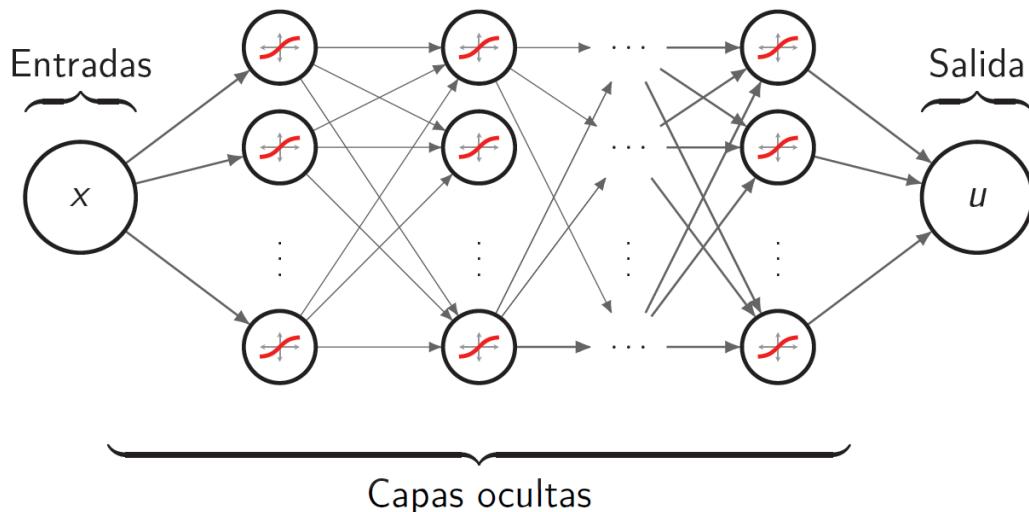
$$\phi(x) = \sum_{i=1}^N \alpha_i \sigma(w_i^T x + b_i)$$

Con parámetros α_i, w_i, b_i y un número de neuronas N en la capa oculta, tal que

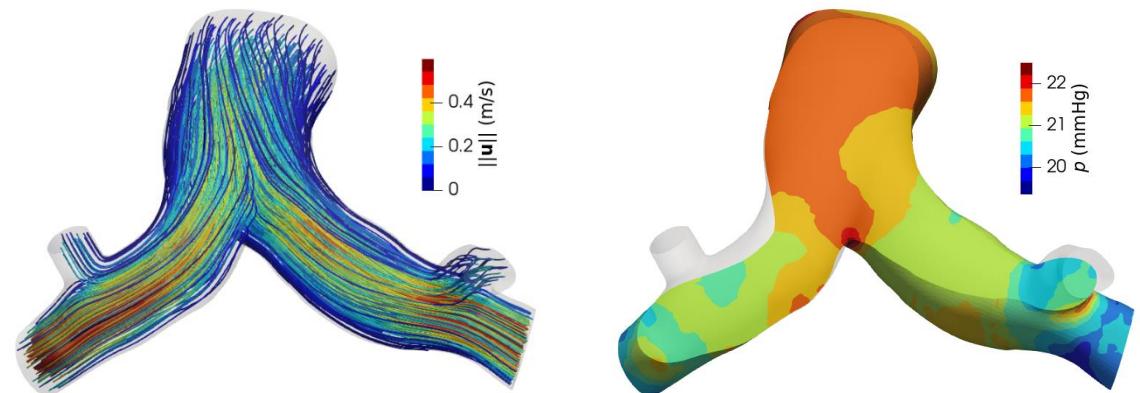
$$\| f - \phi \| < \epsilon \quad \text{en} \quad \Omega$$

Physics informed neural networks (PINNs)

Redes neuronales artificiales informadas por la física



$$\begin{aligned} \mathbf{u}^* \cdot \nabla \mathbf{u}^* - \frac{1}{Re} \Delta \mathbf{u}^* + \nabla p^* &= \mathbf{0} \\ \nabla \cdot \mathbf{u}^* &= 0 \\ BC : \quad \mathbf{u}^* = \mathbf{0}, \quad \frac{1}{Re} \frac{\partial \mathbf{u}^*}{\partial \mathbf{n}} - \mathbf{n} p^* &= \mathbf{0} \end{aligned}$$



Raissi, 2019 (PINNs)



Karniadakis, 2021 (PIML)



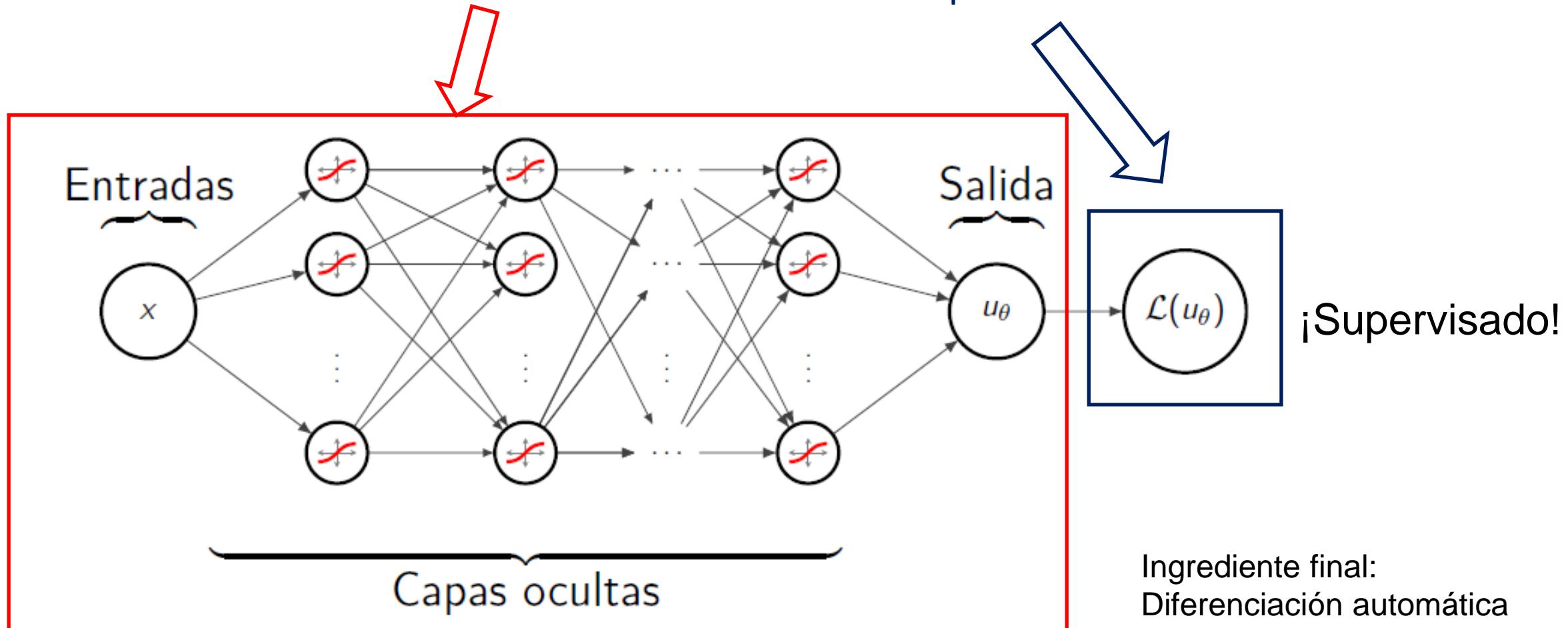
Expert's guide



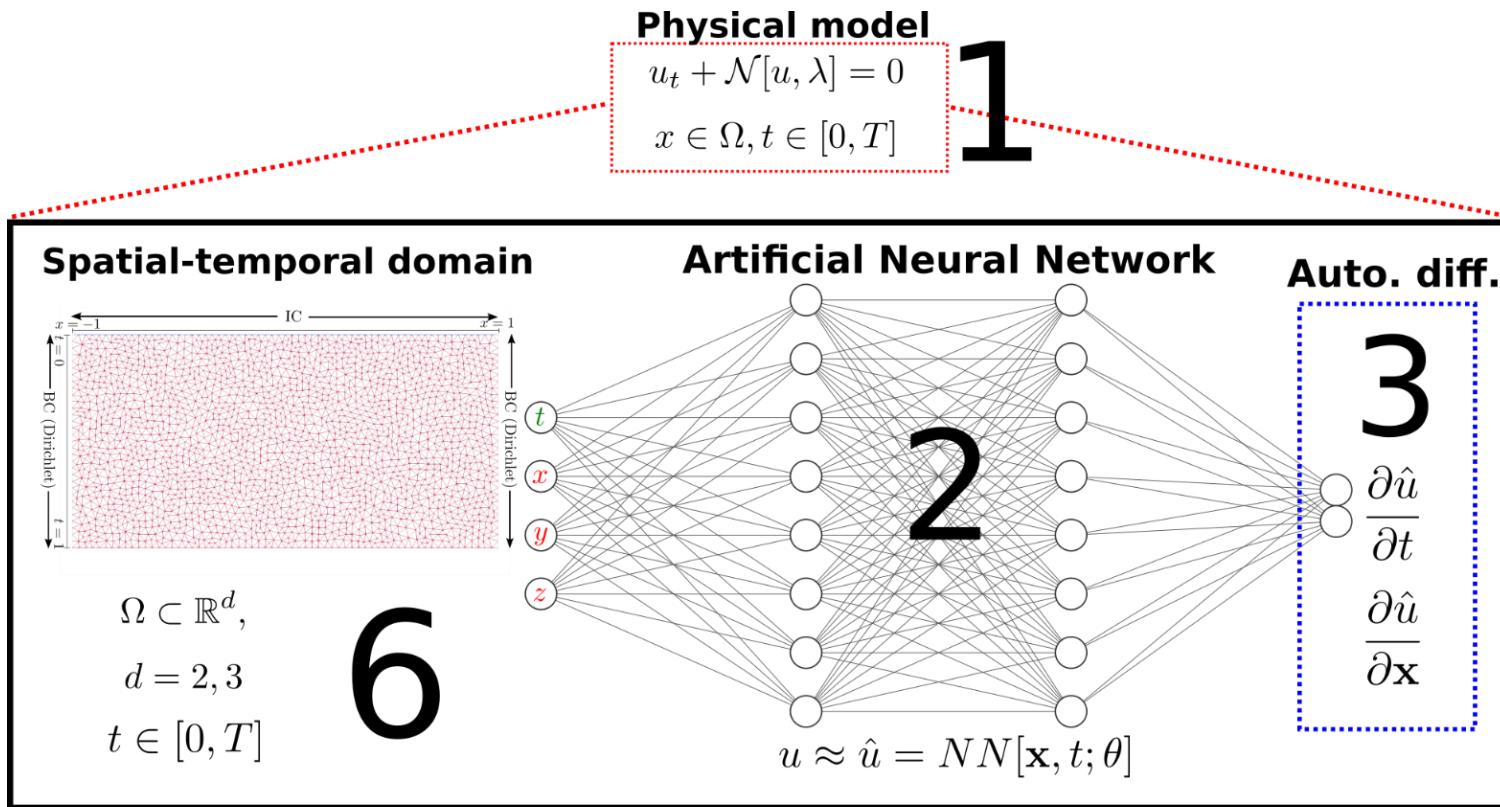
Ben Moseley

Physics informed neural networks (PINNs)

Redes neuronales artificiales informadas por la física



PINNs: esquema base



$$MSE_{Data}(\{u_i, x_i\}, \boldsymbol{\theta}) = \frac{1}{N} \sum_i^N (u_i - \hat{u}(x_i; \boldsymbol{\theta}))^2$$

$$MSE_{PDE}(\boldsymbol{\theta}, \lambda) = \frac{1}{R} \sum_i^R (u_t + \mathcal{N}[\hat{u}(x_i; \boldsymbol{\theta}); \lambda])^2$$

$$MSE_{BC}(\{g_i, x_i^b\}, \boldsymbol{\theta}) = \frac{1}{B} \sum_i^B (\nabla \hat{u}(x_i^b; \boldsymbol{\theta}) \cdot \mathbf{n} - g_i)^2$$

¡Trivia!

Las PINNs funcionan bien incluso cuando los datos contienen ruido.

Verdadero

Falso

Justificación: las PINNs integran restricciones físicas directamente en su función de pérdida durante el entrenamiento, lo que permite que el modelo ajuste sus predicciones no solo en función de los datos observados, sino también de las leyes físicas subyacentes. Este enfoque ayuda a filtrar el ruido presente en los datos, permitiendo que el modelo se enfoque en patrones coherentes con las restricciones físicas, mejorando la robustez de las predicciones incluso en escenarios con alta contaminación de ruido.

En las PINNs, las condiciones iniciales y de frontera se cumplen de forma exacta.

Verdadero

Falso

Justificación: aunque las PINNs incluyen las condiciones de frontera e iniciales en su función de pérdida para guiarlas hacia una solución consistente con estas restricciones, el cumplimiento no siempre es exacto. Esto ocurre porque las PINNs generan soluciones aproximadas al minimizar el error, lo que puede resultar en desviaciones, especialmente en geometrías complejas o cuando los datos son ruidosos. Además, la precisión depende de factores como la arquitectura de la red, el número de nodos de colocación y la convergencia del entrenamiento.

Resolver problemas inversos con PINNs requieren datos de las condiciones de frontera.

Verdadero

Falso

Justificación: una de las ventajas clave de las PINNs es su capacidad para manejar problemas inversos incluso cuando las condiciones de frontera no son completamente conocidas. Las PINNs pueden inferir estas condiciones a partir de datos observados y restricciones físicas (únicamente de la EDP), lo que las hace útiles en situaciones donde la información de contorno es incompleta o está ausente.

Si se modifica un parámetro del modelo matemático del problema (directo) a estudiar, es necesario volver a entrenar la PINN desde cero.

Verdadero

Falso

Justificación: existen estrategias avanzadas, como el **transfer learning**, que permiten reutilizar una PINN previamente entrenada al modificar un parámetro del modelo matemático. En estas estrategias, la función de pérdida (loss function) se ajusta para incluir el parámetro modificado como una variable entrenable, en lugar de reiniciar todo el entrenamiento desde cero. Esto es particularmente efectivo cuando se cuenta con datos observados que abarcan diferentes valores de los parámetros del modelo, permitiendo que la PINN ajuste sus predicciones a las nuevas condiciones sin descartar el conocimiento previo adquirido durante su entrenamiento inicial. Ver el siguiente artículo sobre esto: <https://arxiv.org/abs/2006.14372>

Luego del entrenamiento, la evaluación de las PINNs es muy rápida.

Verdadero

Falso

Justificación: Después del entrenamiento, la evaluación de las PINNs es muy rápida porque consiste únicamente en realizar cálculos hacia adelante (forward propagation) dentro de la red neuronal, lo cual es computacionalmente eficiente. A diferencia del entrenamiento, que implica resolver múltiples iteraciones de optimización y cálculos intensivos de gradientes, la evaluación no requiere recalcular ni ajustar parámetros, lo que permite obtener resultados en tiempo reducido.

Las PINNs funcionan únicamente con perceptrones multicapa.

Verdadero

Falso

Justificación: las PINNs no están limitadas a perceptrones multicapa (MLP). Aunque las MLP son una arquitectura comúnmente utilizada en PINNs debido a su capacidad para aproximar funciones continuas, también se pueden emplear otras arquitecturas, como redes convolucionales (CNN) o redes recurrentes (RNN), dependiendo del tipo de problema que se desea resolver. Estas alternativas pueden ser especialmente útiles para problemas específicos, como aquellos con datos de alta dimensionalidad o secuenciales. Esto demuestra la flexibilidad de las PINNs al adaptarse a diferentes arquitecturas de red.

Las PINNs permiten resolver problemas multifísicos incluso cuando no conocemos todos los modelos físicos.

Verdadero

Falso

Justificación: una de las fortalezas de las PINNs es su capacidad para integrar datos observados y restricciones físicas en la función de pérdida, lo que permite abordar problemas multifísicos incluso cuando los modelos físicos subyacentes no están completamente definidos. Al combinar datos y ecuaciones diferenciales conocidas, las PINNs pueden inferir comportamientos de sistemas complejos, llenando las lagunas en el conocimiento físico con patrones aprendidos de los datos. Esto las hace herramientas efectivas para resolver problemas en los que el modelo físico es parcial o incompleto.

Pros y contras

Pros

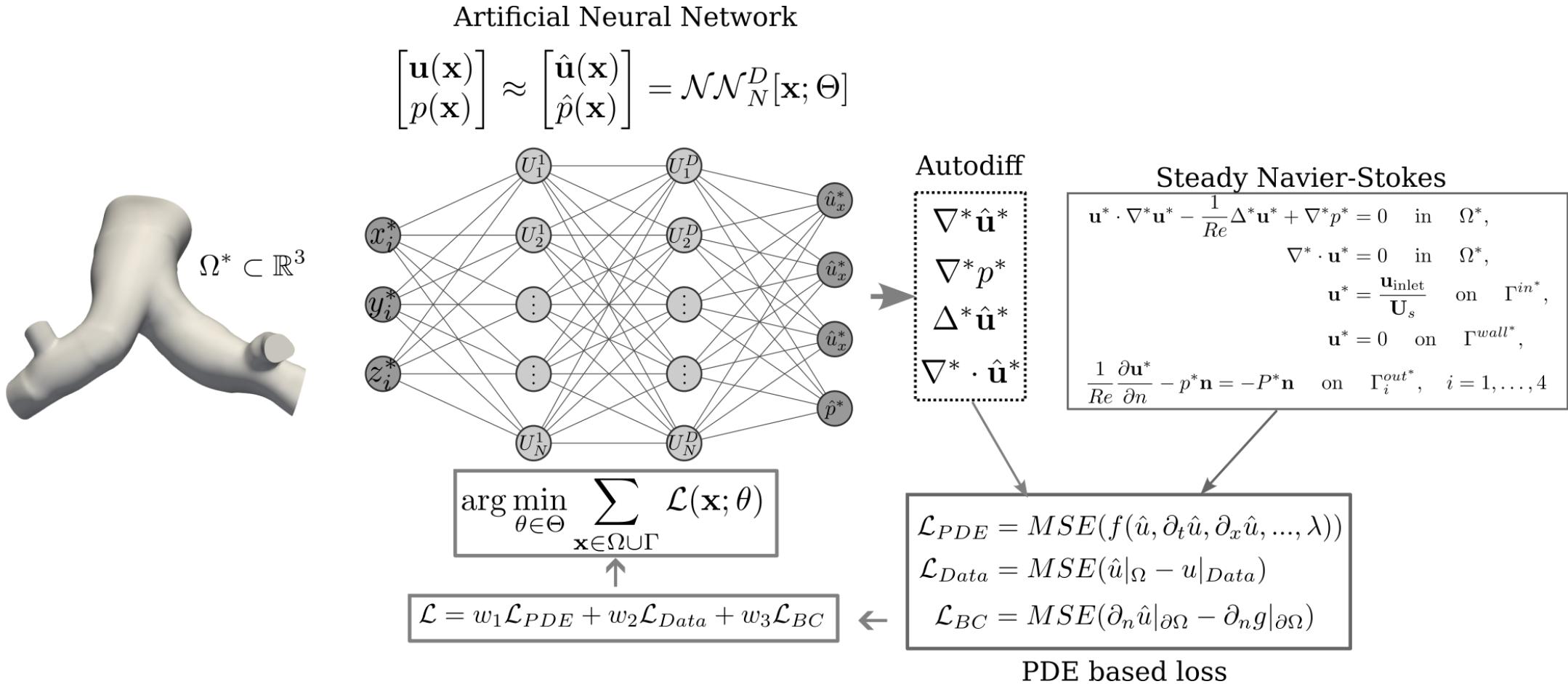
1. Sin malla (pero si sus nodos)
2. Funcionan bien en formulaciones mixtas (o acopladas)
3. Problemas directos e inversos, "descubrimiento" de modelos físicos escondidos
4. Funcionan bien en datos con ruidos
5. Puede eludir algunas condiciones que harían que el problema inverso esté mal planteado, como las condiciones de contorno.

Contras

1. Lentas de entrenar.
2. No cumple completamente las condiciones de frontera en geometrías irregulares
3. Difíciles de sintonizar. No hay una arquitectura definida como "la mejor"
4. En comparación con los métodos numéricos, No hay estimación del error a priori
5. Tiene sesgos espectrales

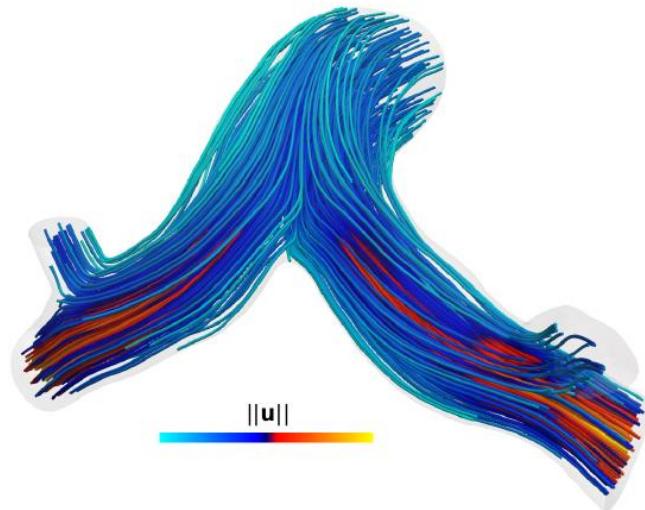
Aplicaciones

PINNs para el estudio de la presión arterial pulmonar

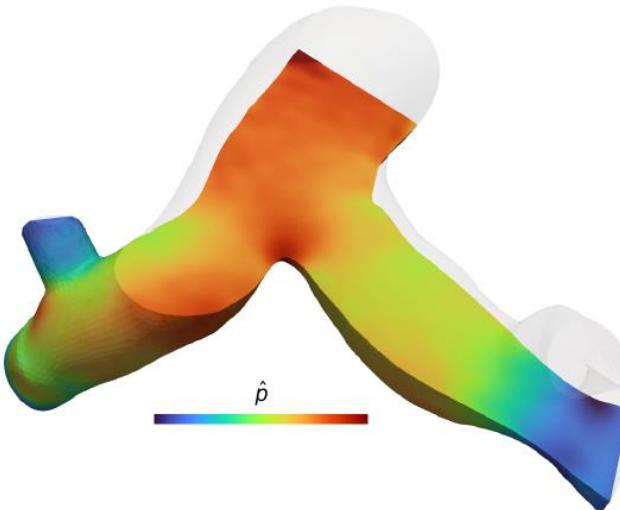
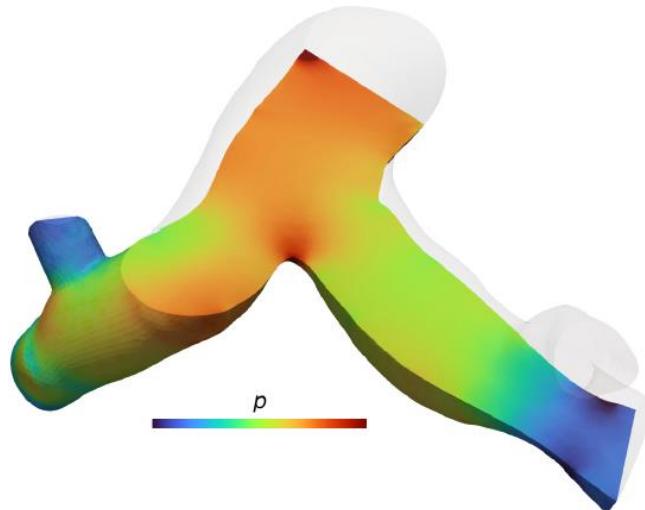
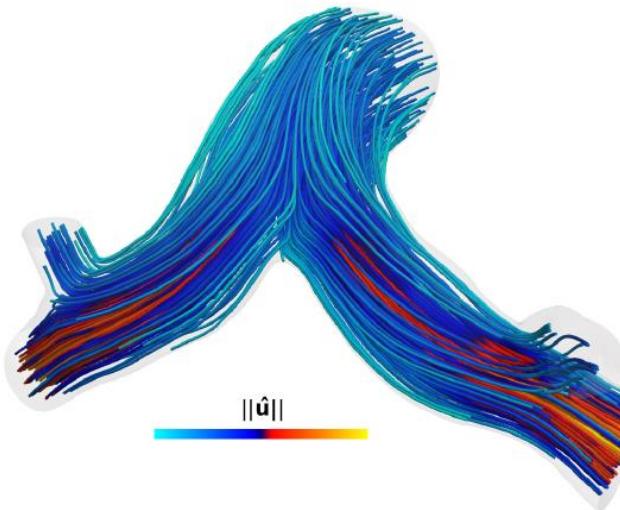


PINNs para el estudio de la presión arterial pulmonar

Solución numérica

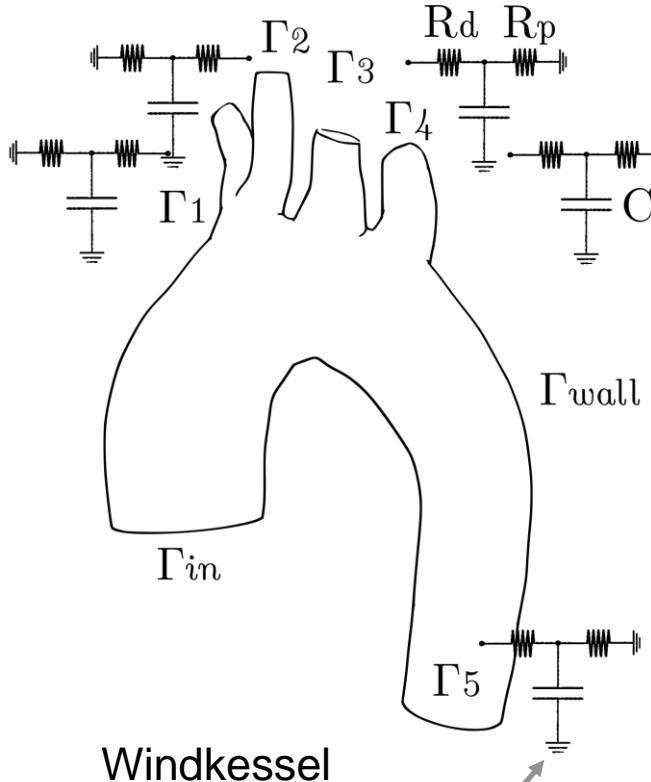


Solución PINN



PINNs para la estimación de parámetros de *Windkessel*

Modelo Directo



$$P_k = R_{p,k}Q_k + \pi_k$$

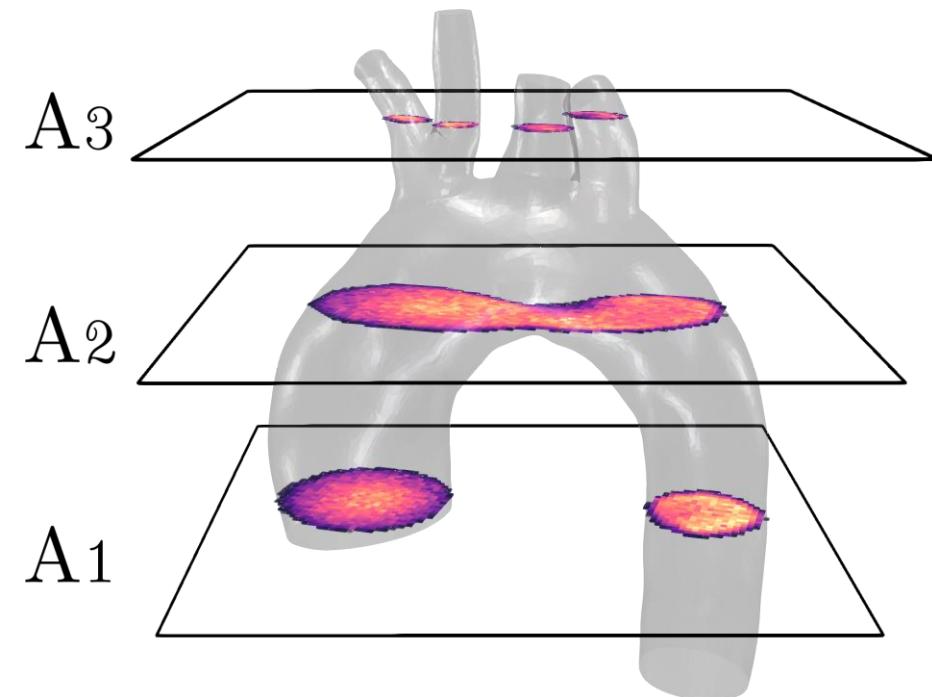
$$Q_k = \int_{\Gamma_k} \mathbf{u} \cdot \mathbf{n}$$

$$C_k \frac{d\pi_k}{dt} + \frac{\pi_k}{R_{d,k}} = Q_k$$

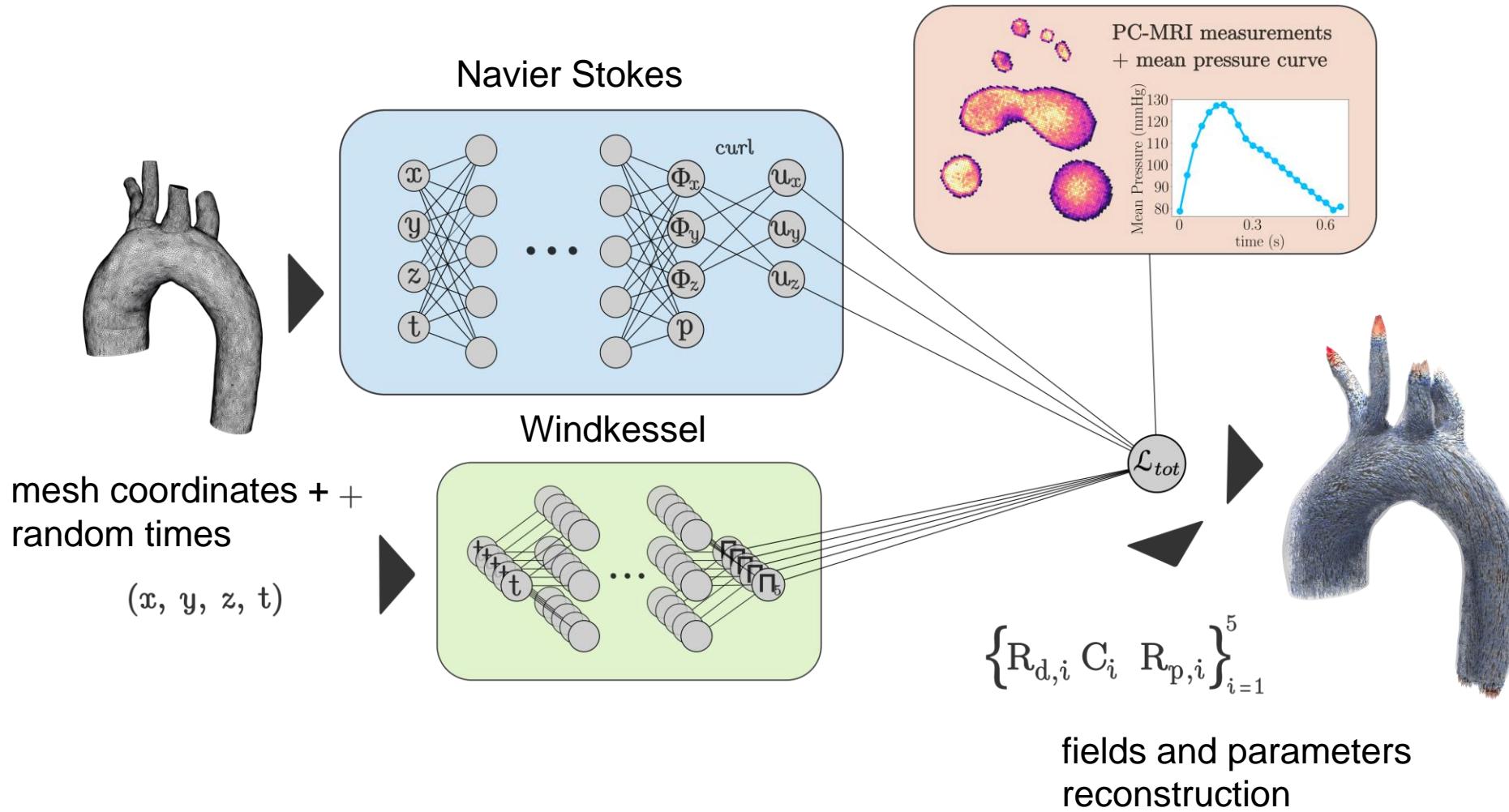
Navier Stokes

$$\frac{\partial \mathbf{u}}{\partial t} + \rho(\mathbf{u} \cdot \nabla) \mathbf{u} - \mu \Delta \mathbf{u} + \nabla p = 0 \text{ in } \Omega \times [0, T]$$
$$\nabla \cdot \mathbf{u} = 0 \text{ in } \Omega \times [0, T]$$
$$\mathbf{u} = \mathbf{u}_{inlet}(t) \text{ on } \Gamma_{in} \times [0, T]$$
$$\mathbf{u} = \mathbf{0} \text{ on } \Gamma_{wall} \times [0, T]$$
$$\mu \frac{\partial \mathbf{u}}{\partial n} - p \mathbf{n} = -P_k(t) \mathbf{n} \text{ on } \Gamma_k \times [0, T], k = 1, \dots, K$$

Mediciones: contraste de fase



time continuous physics-informed neural network



Resultados



ground truth



PINNs

velocity [cm/s]

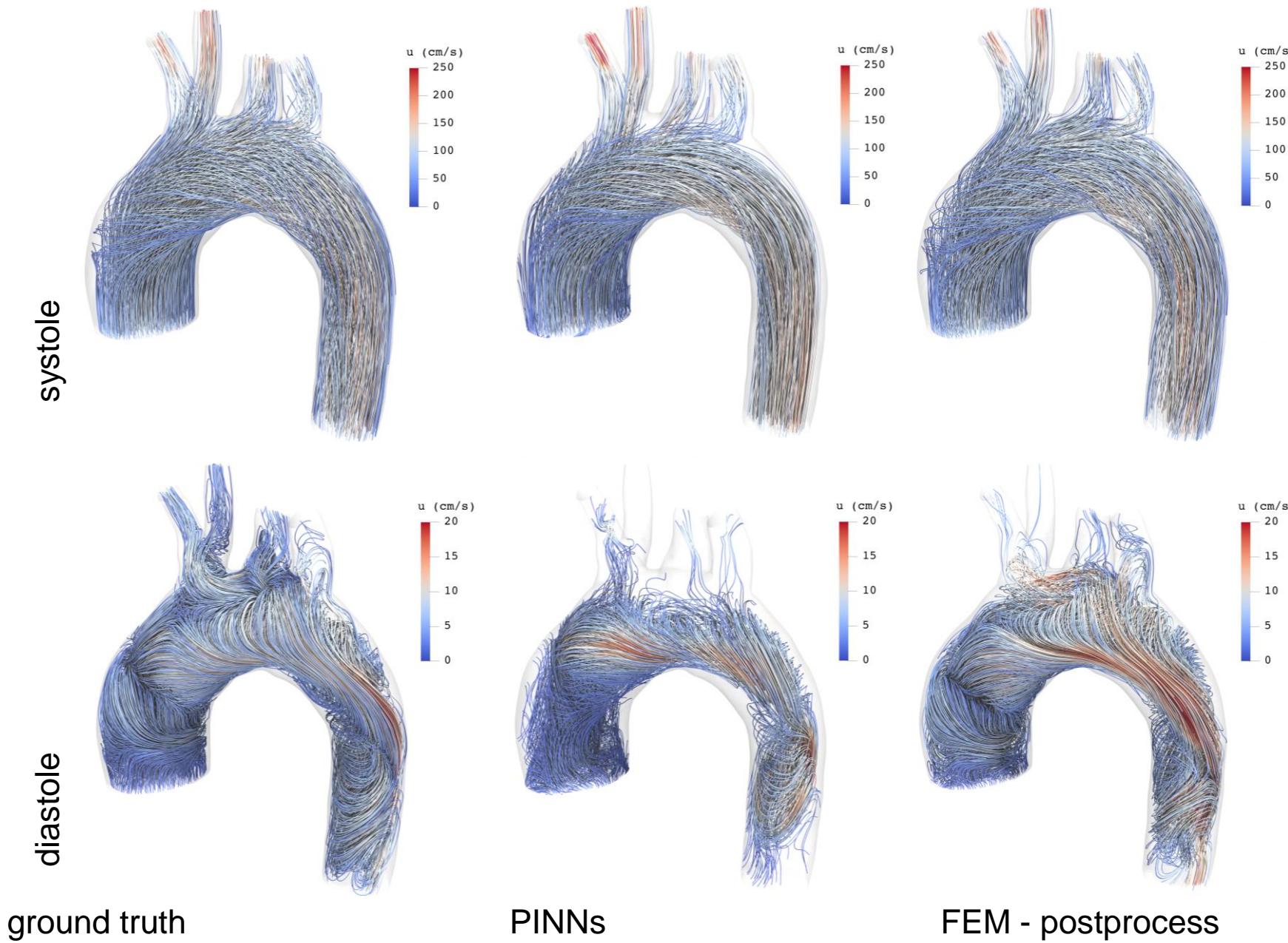
0

250

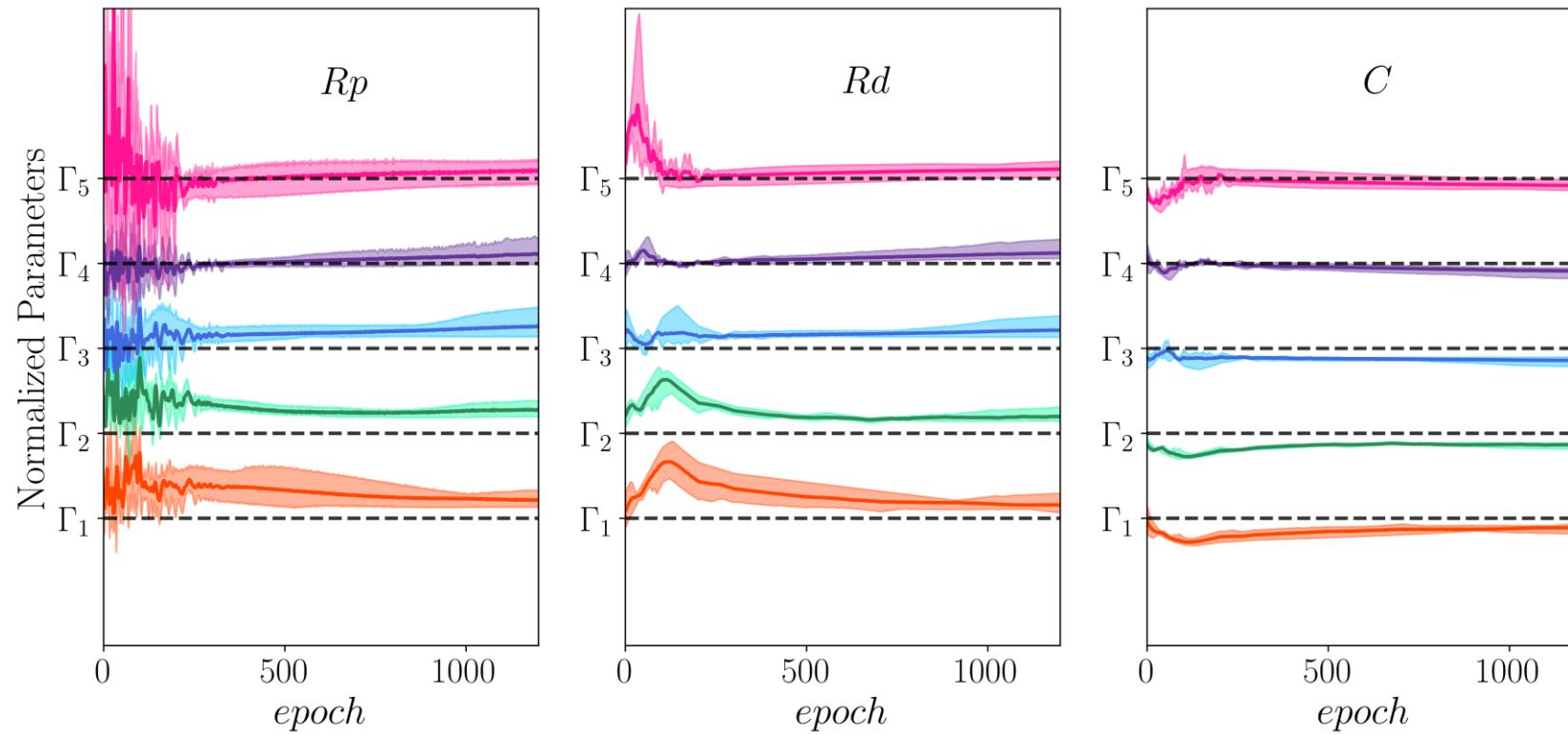


FEM - postprocess

Resultados



Estimación de parámetros



Conclusiones

Conclusiones

- Las PINNs ofrecen una metodología innovadora que combina modelos físicos y aprendizaje automático
- Las PINNs son particularmente útiles para abordar problemas inversos, multifísicos “incompletos”, y con datos ruidosos
- “Unas por otras”: eludir algunas restricciones, como las condiciones de frontera, para obtener soluciones aproximadas de problemas inversos mal planteados
- No existe una única arquitectura óptima para las PINNs, lo que requiere experimentación y ajustes específicos para cada problema

¡Hay muchísimo por hacer y descubrir!

¡Muchas gracias por su atención!

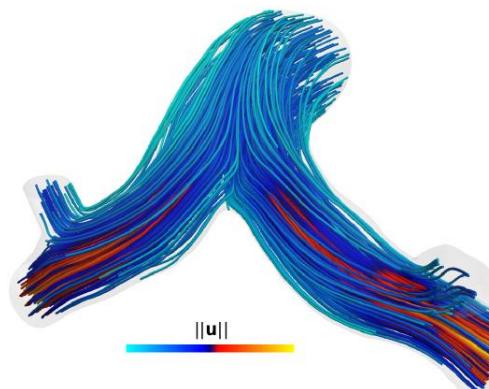
David Ortiz-Puerta

david.ortiz.puerta@gmail.com

david.ortiz@uv.cl



Solución numérica



Solución PINN

