PINNs para solucionar EDPs

Trabajo Final - Inteligencia Artificial

Amelia Hoyos[1,2], Yalena Sotelo[1] & Thomas Martinod[1,2]

[1]Ingeniería Física, Universidad EAFIT, Colombia.
[2]Ingeniería Matemática, Universidad EAFIT, Colombia.

22 de noviembre de 2025

1. Planteamiento del Problema

Este proyecto implementa *Physics-Informed Neural Networks* (PINNs) para resolver ecuaciones diferenciales parciales (EDPs) clásicas sin necesidad de datos etiquetados. A diferencia de los enfoques supervisados tradicionales, las PINNs incorporan las leyes físicas directamente en la función de costo mediante el residuo de la EDP, permitiendo entrenar la red con muy pocos datos o incluso sin datos de solución.

Se desarrolló un marco general capaz de resolver varias EDPs, incluyendo la ecuación de calor, difusión con término fuente, ecuación de Burgers, ecuación de onda amortiguada y Laplace en dos dimensiones. Cada caso se configura en `main_eq.ipynb`, donde se definen la solución exacta, el residuo físico y el dominio de integración.

El objetivo de este trabajo se compone de dos partes: (1) comprender el funcionamiento de las PINNs y (2) evaluar su desempeño comparando la solución aprendida con la solución analítica en distintos problemas.

2. Arquitectura del Modelo

El modelo implementado es una red neuronal totalmente conectada (MLP), definida en el archivo `PINN.py`[1]. Su estructura es:

- Entrada: 2 nodos (x, t). Como observación $x \in \Omega$ compacto de $\mathbb{R}^n$ y $t \in \mathbb{R}_0^+$. No obstante limitamos el número de variables espaciales tal que el número total de variables fuera 2 y así poder visualizar las gráficas.
- Capas ocultas: $[32, 32]$ neuronas con activación tanh.
- Salida: un único nodo que representa $u(x, t)$.

La característica diferenciadora de las PINNs es su **función de costo híbrida**, que combina:

1. **Pérdida en condiciones de frontera e iniciales:** $\mathcal{L}_{\text{BC}} = \|u_\theta(x_{\text{bc}}) - u_{\text{bc}}\|^2$.

[1] Los códigos del proyecto con los archivos mencionados se alojan en el repositorio WonderfulAme/PINNs

2. **Pérdida física dada por el residuo de la EDP:** $\mathcal{L}_{\text{PDE}} = \|\mathcal{N}[u_\theta]\|^2$, donde $\mathcal{N}$ es el operador diferencial de la ecuación $\mathcal{N}(u) = 0$. La norma que usamos en ambas funciones de pérdida es la norma euclidea estándar $\|\cdot\| = \|\cdot\|_2$.

Para calcular u_x, u_{xx}, u_t y otros términos derivados, la red utiliza **autodiferenciación**, integrando estos gradientes en el proceso de backpropagation. Esto actúa como una forma de "reparametrización funcional" que permite que los operadores diferenciales participen en el cálculo del gradiente.

3. Procedimiento de Entrenamiento

Todos los casos utilizan la misma rutina general:

- **Optimizador:** Adam
- **Tasa de aprendizaje:** 10^{-3}
- **Iteraciones:** entre $2 \cdot 10^4$ y $3 \cdot 10^4$ según la EDP
- **Tamaño de batch:** entrenamiento full-batch (todos los puntos)
- **Función de pérdida:** error cuadrático medio (MSE)
- **Muestreo:** Latin Hypercube Sampling para los puntos de colación.

Los datos de entrenamiento incluyen los puntos de frontera e iniciales x_{bd} obtenidos de la solución exacta, y los puntos de colación internos para forzar el cumplimiento de la EDP.
La pérdida total es la suma de las mencionadas: $\mathcal{L} = \mathcal{L}_{\text{BC}} + \mathcal{L}_{\text{PDE}}$.

4. Resultados

La clase `PINNSolver` fue validada en múltiples ecuaciones programadas en `main_eq.ipynb`. Gracias a su diseño modular, la misma arquitectura funciona para EDPs lineales, no lineales y de distinta dimensionalidad.

Como ejemplo representativo, se presentan los resultados obtenidos para la **ecuación de difusión**:

$$u_t - u_{xx} - f(x,t) = 0, \qquad f(x,t) = e^{-t}\left(\sin(\pi x) - \pi^2 \sin(\pi x)\right).$$

con solución analítica:

$$u(x,t) = e^{-t}\sin(\pi x),$$

Tras el entrenamiento, la red pudo recuperar con alta precisión la estructura de la solución analítica (ver Figura 1). Las métricas obtenidas fueron:

- Error absoluto máximo: $9{,}99 \times 10^{-2}$
- Error absoluto medio: $3{,}99 \times 10^{-2}$
- Error relativo L_2: $1{,}00 \times 10^{-1}$

Además de este caso, el solver se aplicó exitosamente a las ecuaciones del calor, difusión con fuente, Burgers, onda amortiguada, y Laplace en 2D. Esto muestra la capacidad de generalización del enfoque y su utilidad como método alternativo a los esquemas numéricos tradicionales.

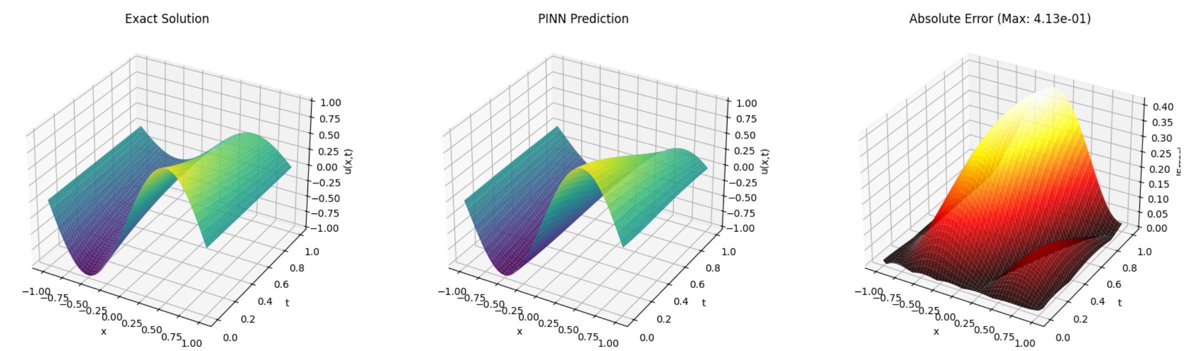

Figura 1: Solución exacta, solución PINN y error absoluto para la ecuación de difusión.

5. Conclusiones

Las PINNs ofrecen una herramienta flexible para resolver EDPs imponiendo directamente la física en la función de costo. La implementación desarrollada permite resolver un conjunto amplio de ecuaciones mediante un único modelo, logrando errores bajos y comportamientos suaves en todas las pruebas. Como trabajo futuro se propone incorporar técnicas de ponderación adaptativa, refinamiento automático de puntos de colación y dominios irregulares pero compactos para extender la aplicabilidad del método.