

PINN_o

Physics-Informed Neural Network Interactive Trainer

Manual de Usuario

Versión 1.0

Autores:

Angie Lorena Pineda Morales
Juan Sebastian Acuña Tellez
Pablo Patiño Bonilla

Universidad Distrital Francisco José de Caldas
21 de noviembre de 2025

Índice general

1. Introducción	2
1.1. ¿Qué es PINNo?	2
1.2. Características Principales	2
2. Instalación y Requisitos	3
2.1. Requisitos del Sistema	3
2.2. Instalación Paso a Paso	3
2.2.1. 1. Clonar el Repositorio	3
2.2.2. 2. Crear un Entorno Virtual (Recomendado)	3
2.2.3. 3. Instalar Dependencias	4
3. Guía de Uso	5
3.1. Opción A: Interfaz Web (Streamlit)	5
3.2. Opción B: Interfaz de Escritorio	5
3.3. Flujo de Trabajo Típico	5
3.3.1. 1. Pestaña: Learn PINNs (Educación)	5
3.3.2. 2. Pestaña: Data Explorer (Carga de Datos)	6
3.3.3. 3. Pestaña: Train & Config (Entrenamiento)	6
3.3.4. 4. Pestaña: Metrics Report (Resultados)	6
4. Solución de Problemas	7
4.1. Errores Comunes	7
4.1.1. Error: “x_tt is None”	7
4.1.2. La pérdida no baja (Loss estancado)	7
4.1.3. Interfaz Web no carga	7
A. Ecuaciones Físicas Implementadas	8
A.1. Oscilador Armónico Simple (SHO)	8
A.2. Oscilador Armónico Amortiguado (DHO)	8
A.3. Ecuación de Calor 2D (HEAT)	8

Capítulo 1

Introducción

1.1. ¿Qué es PINNo?

PINNo es una plataforma educativa y de investigación diseñada para entrenar, visualizar y experimentar con Redes Neuronales Informadas por la Física (PINNs).

A diferencia de las redes neuronales tradicionales que solo aprenden de datos observados, PINNo permite incorporar leyes físicas (expresadas como Ecuaciones Diferenciales) directamente en la función de pérdida del modelo.

1.2. Características Principales

- **Doble Interfaz:** Versión de escritorio (Tkinter) y versión web (Streamlit).
- **Entrenamiento Híbrido:** Capacidad de aprender simultáneamente de ecuaciones físicas y datos experimentales (.csv/.xlsx).
- **Visualización en Tiempo Real:** Gráficos de convergencia y mapas de calor dinámicos para la ecuación de calor 2D.
- **Arquitectura Flexible:** Modificación de hiperparámetros (capas, neuronas, funciones de activación) sin editar código.

Capítulo 2

Instalación y Requisitos

2.1. Requisitos del Sistema

Para ejecutar PINNo, necesita tener instalado **Python 3.10** o superior. Las dependencias principales incluyen:

- TensorFlow $\geq 2.12.0$
- NumPy, Matplotlib, Pandas
- Streamlit (para la interfaz web)
- Pytest (para pruebas unitarias)

2.2. Instalación Paso a Paso

2.2.1. 1. Clonar el Repositorio

Descargue el código fuente o clone el repositorio en su máquina local.

2.2.2. 2. Crear un Entorno Virtual (Recomendado)

Es buena práctica aislar las dependencias del proyecto.

Listing 2.1: Creación del entorno virtual

```
1 # En Windows (PowerShell)
2 python -m venv .venv
3 .venv\Scripts\activate
4
5 # En Linux / macOS
6 python3 -m venv .venv
7 source .venv/bin/activate
```

2.2.3. 3. Instalar Dependencias

Ejecute el siguiente comando en la raíz del proyecto:

```
1 pip install -r requirements.txt
```

Capítulo 3

Guía de Uso

PINNo ofrece dos formas de interactuar con el software. Ambas comparten el mismo núcleo de entrenamiento matemático.

3.1. Opción A: Interfaz Web (Streamlit)

Esta es la opción recomendada por su diseño moderno y gráficos interactivos.

Para iniciar, ejecute:

```
1 streamlit run gui_streamlit.py
```

El navegador se abrirá automáticamente en <http://localhost:8501>.

3.2. Opción B: Interfaz de Escritorio

Una aplicación de ventana nativa, útil si no desea usar el navegador.

Para iniciar, ejecute:

```
1 python run_desktop.py
```

3.3. Flujo de Trabajo Típico

El software está organizado en 4 pestañas principales. A continuación se describe el flujo de trabajo sugerido:

3.3.1. 1. Pestaña: Learn PINNs (Educación)

Si es nuevo en el tema, comience aquí. Encontrará explicaciones sobre:

- Diferencias entre Redes Neuronales Clásicas y PINNs.
- Significado de la función de pérdida ($Loss_{Total} = Loss_{Data} + Loss_{Physics}$).
- Glosario de hiperparámetros (Epochs, Learning Rate, etc.).

3.3.2. 2. Pestaña: Data Explorer (Carga de Datos)

(Opcional) Si tiene datos experimentales, puede cargarlos aquí.

- **Formatos soportados:** .csv o .xlsx.
- **Estructura requerida:**
 - Para Osciladores (SHO/DHO): Columnas t , x .
 - Para Calor (HEAT): Columnas x , y , t , u .

Al cargar un archivo, el sistema activa automáticamente el modo de **Entrenamiento Híbrido**.

3.3.3. 3. Pestaña: Train & Config (Entrenamiento)

Este es el panel de control principal.

1. **Seleccionar Problema:** Elija entre SHO (Oscilador Armónico), DHO (Amortiguado) o HEAT (Calor 2D).
2. **Configurar Arquitectura:** Puede modificar la profundidad de la red (capas), el ancho (neuronas) y la función de activación (se recomienda \tanh para física).
3. **Parámetros Físicos:** Ajuste variables como la frecuencia (ω) o la difusividad térmica (α).
4. **Iniciar:** Presione Start Training.

Visualización en Vivo:

- Verá la curva de pérdida disminuyendo (escala logarítmica).
- Para la ecuación **HEAT**, verá dos mapas de calor comparando la predicción vs. la solución analítica en tiempo real.

3.3.4. 4. Pestaña: Metrics Report (Resultados)

Al finalizar, diríjase a esta pestaña para:

- Ver estadísticas finales (Error mínimo, época final).
- **Exportar Gráficas:** Guarde la imagen de convergencia.
- **Exportar Modelo:** Guarde el archivo .keras para usarlo posteriormente en otros scripts de Python.
- **Exportar Datos:** Descargue el historial de pérdida en CSV.

Capítulo 4

Solución de Problemas

4.1. Errores Comunes

4.1.1. Error: “x_tt is None”

Causa: Está utilizando una función de activación no diferenciable dos veces (como ReLU estándar en algunas implementaciones) o el grafo de computación se ha roto. **Solución:** Cambie la función de activación a `tanh` o `swish` en la pestaña de configuración.

4.1.2. La pérdida no baja (Loss estancado)

Causa: Tasa de aprendizaje (Learning Rate) inadecuada o red muy superficial para el problema. **Solución:**

- Para problemas simples (SHO), intente $LR = 0.001$.
- Para problemas complejos (HEAT), aumente las capas ocultas a 6 y las neuronas a 64.

4.1.3. Interfaz Web no carga

Causa: El puerto 8501 puede estar ocupado. **Solución:** Ejecute Streamlit en otro puerto:

```
1 streamlit run gui_streamlit.py --server.port 8502
```


Apéndice A

Ecuaciones Físicas Implementadas

A.1. Oscilador Armónico Simple (SHO)

$$m \frac{d^2 x}{dt^2} + kx = 0 \quad \rightarrow \quad \ddot{x} + \omega^2 x = 0$$

A.2. Oscilador Armónico Amortiguado (DHO)

$$\ddot{x} + 2\zeta\omega\dot{x} + \omega^2 x = 0$$

Donde ζ es el coeficiente de amortiguamiento.

A.3. Ecuación de Calor 2D (HEAT)

$$\frac{\partial u}{\partial t} = \alpha \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right)$$

Donde α es la difusividad térmica.