PINN

Tomas Escobar Juan Jose Garcia Manuel Santiago Martinez
Andrea Sánchez Santiago Uribe [1]

7 de marzo de 2025

[1] Universidad Nacional de Colombia, Sede Medellín

Tabla de contenidos

Redes neuronales

Las redes neuronales artificiales son modelos computacionales inspirados en la estructura y funcionamiento del cerebro humano. Están compuestas por unidades llamadas neuronas artificiales, organizadas en capas que procesan la información de manera jerárquica. Su entrenamiento se basa en la propagación de información y el ajuste de pesos mediante algoritmos de optimización.

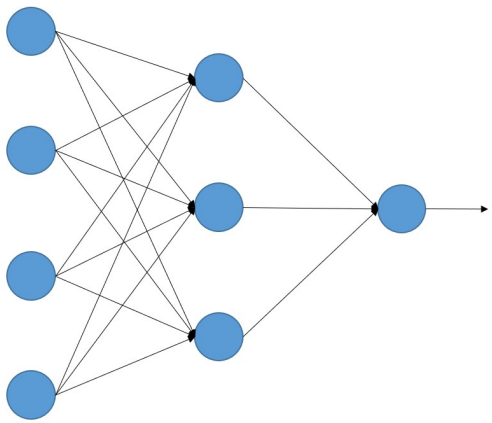
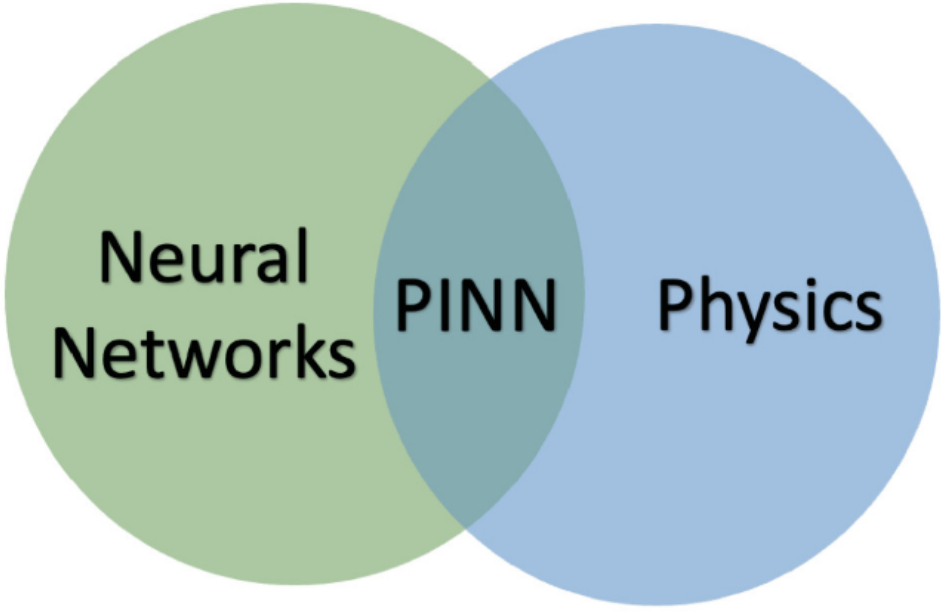


Figura: Red neuronal

PINNs - (Physics-Informed Neural Networks)



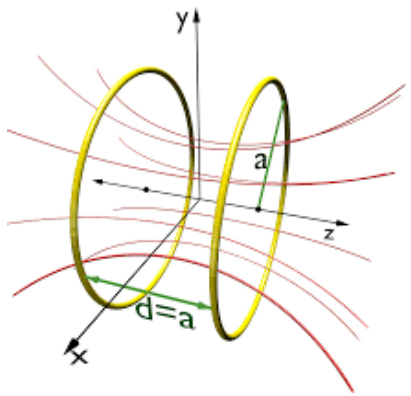
Tipo de red neuronal que incorpora principios físicos en su entrenamiento. Se utilizan para resolver ecuaciones diferenciales parciales de manera diferente a las convencionales, combinando el aprendizaje automático con el conocimiento previo de las leyes físicas.

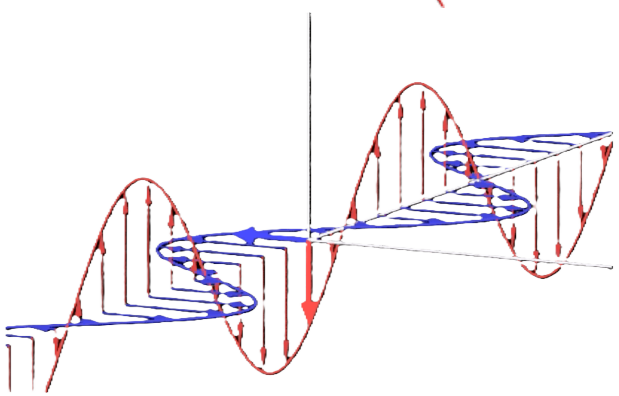
Física en las redes neuronales

La ecuación de Helmholtz, que surge naturalmente en óptica y electromagnetismo, es la base física de nuestro proyecto. Esta ecuación describe la propagación de ondas en estado estacionario y tiene la forma:

$$-\Delta u + \alpha u = f$$

donde u representa el campo (eléctrico o magnético), α es una constante relacionada con la frecuencia, y f es un término fuente.





Planteamiento del problema

Se busca resolver la ecuación de Helmholtz en un dominio $x \in (0, \pi)$ con condiciones de frontera en $x \in \{0, \pi\}$, es decir:

$$\begin{cases} -\Delta u + \alpha u = f, & x \in (0, \pi) \\ u = g, & x \in \{0, \pi\} \end{cases}$$

en donde u es la solución de la ecuación, α es una constante, f es una función fuente y g es una función de frontera.

Primer planteamiento

El primer planteamiento utiliza la función de pérdida de mínimos cuadrados, incluyendo las condiciones de frontera directamente en la función de pérdida. Esto significa que la función de pérdida calcula tanto el error de la PDE como el error de las condiciones de frontera.

$$\mathcal{L}(u_{NN}) = \frac{1}{N}\sum_{i=1}^{N}\Big[\lambda_0\left(-\Delta u_{NN}(x_i) + \alpha u_{NN}(x_i) - f(x_i)\right)^2 + \lambda_1\left(u_{NN}(0) - g(0)\right)^2 + \lambda_2\left(u_{NN}(\pi) - g(\pi)\right)^2\Big]$$

En donde λ_0, λ_1 y λ_2 son los parámetros de peso del modelo.

Segundo planteamiento

El segundo planteamiento propone una solución para el caso en que la frontera cumple $g(x) = 0$ para $x \in \{0, \pi\}$. En donde,

$$u_{NN} = NN(x) \cdot x \cdot (x - \pi)$$

Con función de pérdida

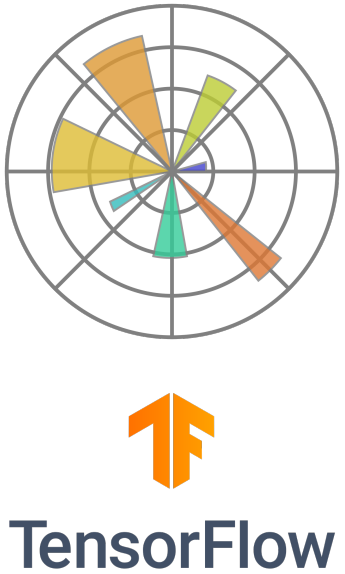
$$\mathcal{L}(u_{NN}) = \frac{1}{N} \sum_{i=1}^{N} [\, -\Delta u_{NN}(x_i) + \alpha u_{NN}(x_i) - f(x_i) \,]^2$$

Aquí las condiciones de frontera se manejan dentro del modelo de red neuronal en lugar de la función de pérdida. Esto se logra mediante el uso de una capa Lambda que ajusta la salida del modelo para cumplir con las condiciones de frontera.

Herramientas y tecnologías

Para la implementación de los modelos se utilizaron las siguientes herramientas:

- **Lenguaje de programación**: Python
- **Bibliotecas**:
  - TensorFlow y Keras para la construcción y entrenamiento de las redes neuronales
  - Numpy y Pandas para el manejo de datos y cálculos numéricos
- **Visualización**: Matplotlib para graficar los resultados y compararlos con la solución analítica



Métricas de evaluación

Para evaluar la precisión de los modelos, se utilizó el error H^1 como métrica principal, definido como:

$$\text{Error Relativo} = \frac{\frac{1}{N}\sum_{i=1}^{N}(u'_{NN}(x_i) - u'(x_i))^2}{\frac{1}{N}\sum_{i=1}^{N}(u'(x_i))^2}$$

donde $u_{NN}(x)$ es la solución aproximada por la red neuronal y $u(x)$ es la solución analítica.

Optimización y experimentación de hiperparámetros

Para analizar el impacto de los hiperparámetros en el desempeño de la PINNs, se realizaron **45 experimentos** utilizando una estrategia de búsqueda aleatoria en los siguientes rangos:

- Número de neuronas por capa: entre **4** y **10**.
- Número de capas ocultas: entre **4** y **5**.
- Número de puntos de integración: entre **500** y **3000**.
- Pesos de penalización para la ecuación diferencial (λ_0) y las condiciones de contorno (λ_1, λ_2): entre **1** y **10**.

Cuadro: Configuración base

Hiperparámetro	Valor
Neuronas	10
Capas	5
Puntos	2000
λ_0	5
λ_1	7
λ_2	7

Funciones testeadas

Queremos probar como funciona ambos planteamientos con diferentes funciones de prueba. Para esto, se probaron las siguientes funciones:

1. $u(x) = \sin(2x)$
2. $u(x) = \sin(2x)\cos\left(\frac{x}{2}\right)$
3. $u(x) = x^2(x-\pi)^2$

Resultados

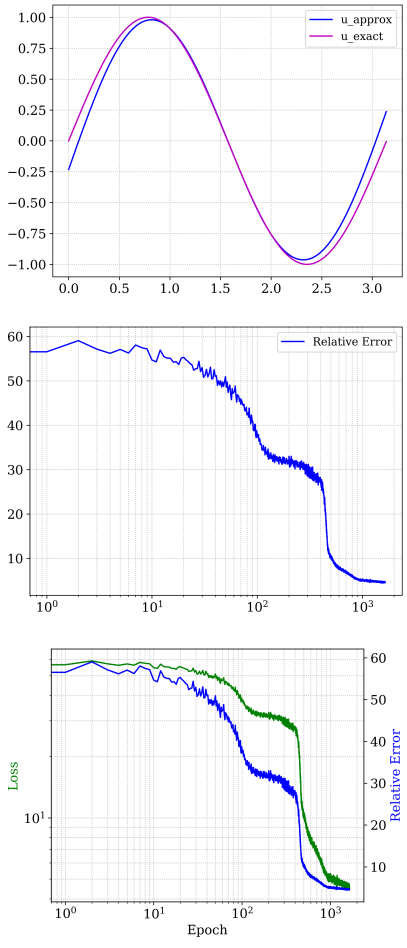


Figura: Resultados - Planteamiento 1

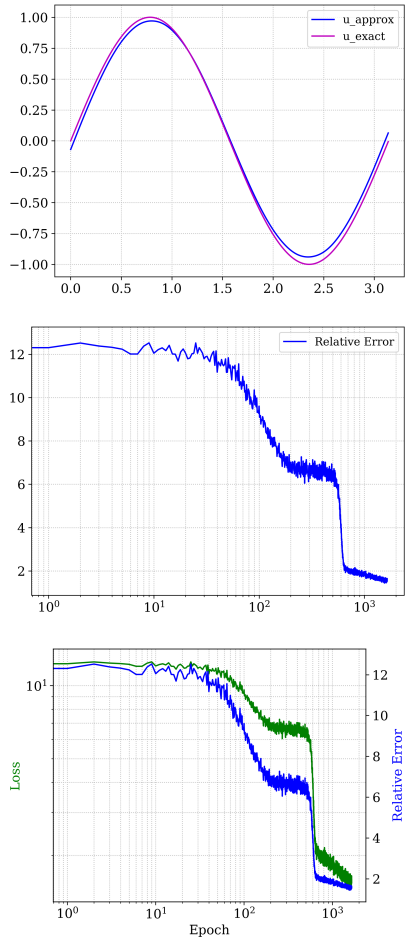


Figura: Resultados - Planteamiento 1

Resultados

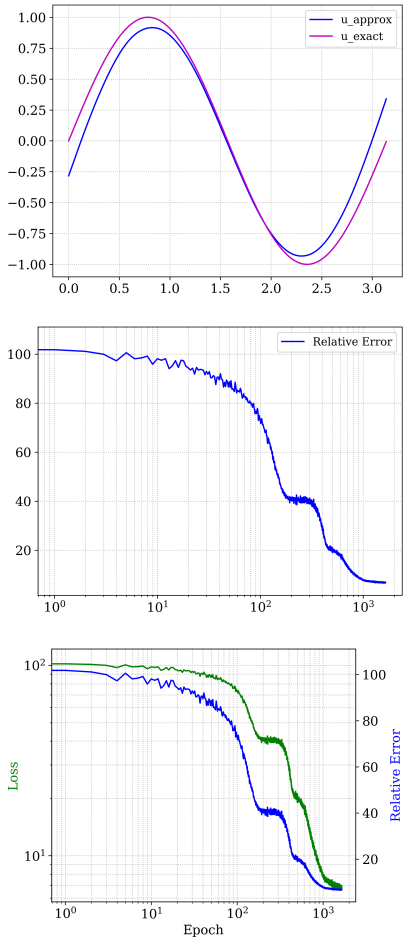


Figura: Resultados - Planteamiento 1

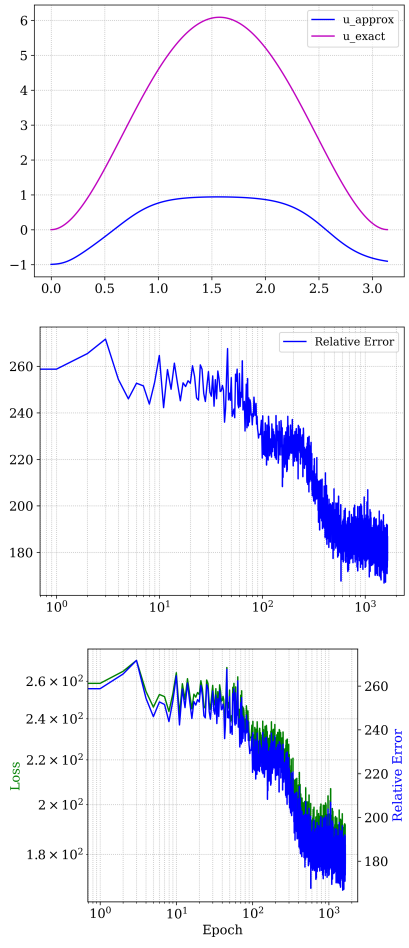


Figura: Resultados - Planteamiento 1

Resultados

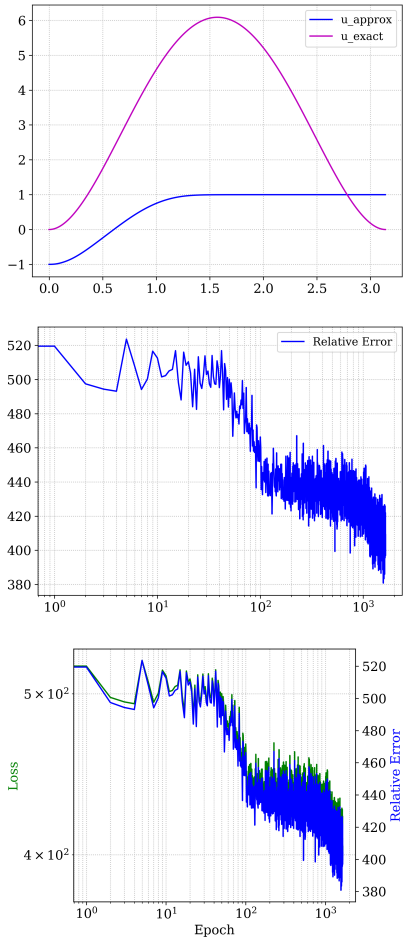


Figura: Resultados - Planteamiento 1

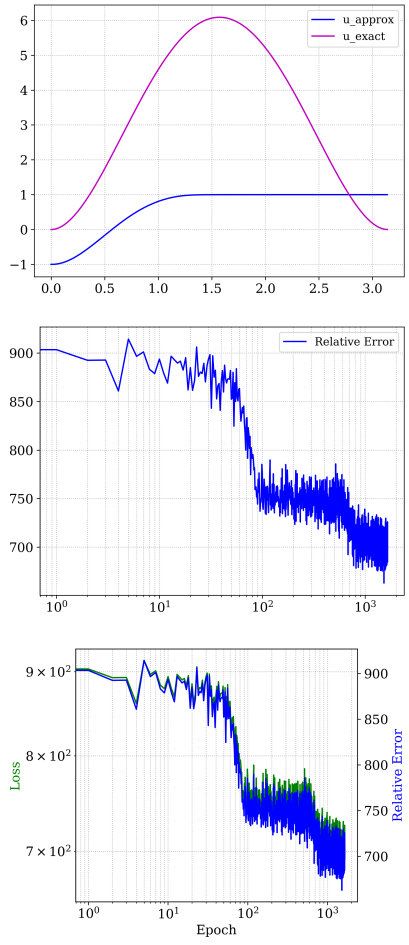


Figura: Resultados - Planteamiento 1

Resultados

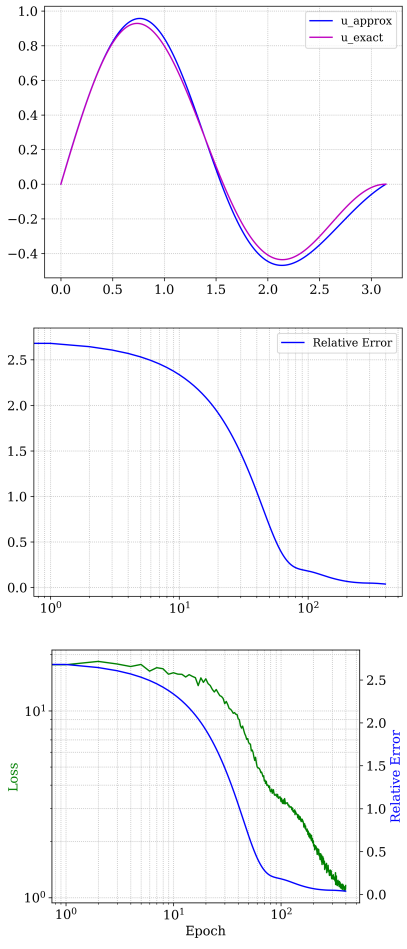


Figura: Resultados - Planteamiento 2

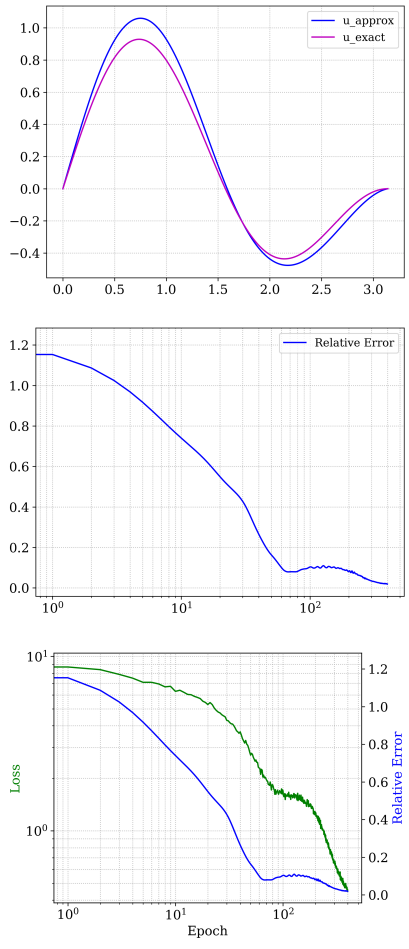


Figura: Resultados - Planteamiento 2