



UNIVERSIDAD AUTÓNOMA DEL ESTADO DE
MÉXICO

FACULTAD DE CIENCIAS

**“Redes neuronales y su aplicación para
resolver la ecuación modificada de Korteweg
- de Vries”**

Por

Hugo Enrique Dapic Sánchez

Tesis para obtener el título de:

Físico

Asesor: Dr. Máximo Augusto Agüero Granados

Co-asesor: Dr. Leonid Serkin



Toluca, Estado de México, Septiembre 2024

Índice general

0.1. Agradecimientos	2
0.2. Resumen	11
1. Introducción general	13
1.1. Introducción	13
1.2. Antecedentes históricos	17
1.2.1. Inteligencia Artificial	20
1.2.2. Programación	25
1.2.3. Python	26
1.2.4. Redes neuronales	27
1.3. Objetivos	31
1.3.1. Objetivo general	31
1.3.2. Objetivos particulares	31
1.4. Desarrollo	32
1.4.1. Ecuaciones diferenciales parciales	32
1.4.2. Ondas	34

1.4.3.	Ecuaciones diferenciales parciales no lineales	43
1.4.4.	Ondas no lineales	45
1.4.5.	Dispersión	49
1.4.6.	Integrabilidad	51
1.4.7.	Método de dispersion inversa	51
1.4.8.	Solución exacta	53
1.4.9.	Aprendizaje de las redes neuronales	56
1.4.10.	Método PINN	57
1.4.11.	PINN´S	60
1.5.	Ejemplo número 2	68
1.6.	PINN del ejemplo número 2	69
1.7.	Conclusiones	74
1.8.	Referencias	76

Índice de figuras

1.1. Recreación del canal donde Russell observó el solitón	18
1.2. $u(x,t)$ es la evolución foto de pulso en un instante	33
1.3. Forma alterna de foto de pulso en el tiempo t_0	34
1.4. Onda estacionaria la cuadrícula va en función $f(x, t_0)$	35
1.5. Esquema de onda viajera	36
1.6. Fotografía de una onda mecánica en agua siendo vista su sombra	39
1.7. Onda no lineal sin dispersión propagándose donde se deforma su cresta, ya que su velocidad es mayor que las otras partes . .	48
1.8. Onda lineal propagándose que pierde su perfil al punto de desvanecerse mientras avanza en el tiempo	50
1.9. Solución graficada en tres dimensiones	55
1.10. Se muestra el flujo del algoritmo PINN. La idea principal del algoritmo PINN es imponer restricciones de información física durante el entrenamiento de la red neuronal para acelerar el entrenamiento de la red neuronal y mejorar la precisión del modelo.	58

1.11. Solución solitón de aprendizaje graficada	60
1.12. Funciones de peso y la ecuación a resolver	61
1.13. La estructura que llevó la optimización BFGS-B	62
1.14. Dominios que tiene, capas ocultas y condiciones	63
1.15. Solitón de los datos previos de aprendizaje hechos por la PINN	64
1.16. Forma en que se denota para las gráficas finales	65
1.17. Gráficas donde se compara en tres tiempos diferentes mostran- do así el aprendizaje de la PINN $t=0.03$, $t=0.05$ y $t=0.08$. . .	66
1.18. Gráfica de pérdidas en el entrenamiento por iteración	67
1.19. Solución graficada en tres dimensiones de sech	68
1.20. Código de datos previos de aprendizaje usado en ambos casos adaptado respectivamente	70
1.21. Solución solitón de aprendizaje de la ecuación (1.38) graficada	71
1.22. Soliton $\text{sech}(x-t)$ de los datos previos de aprendizaje hechos por la PINN	71
1.23. Gráficas donde se compara en tres tiempos diferentes mostran- do así el aprendizaje de la PINN $t=0.03$, $t=0.05$ y $t=0.08$. . .	72
1.24. Gráfica de pérdidas en el entrenamiento por iteración	73

0.2. Resumen

En esta tesis se estudia la ecuación modificada de Korteweg-De Vries la cual describe el comportamiento de redes armónicas, solitones acústicos de iones, modelos de solitones iónicos etc. siendo vital que tiene distintas formas de encontrar su solución, al ser una ecuación diferencial parcial no lineal permite que incluso acepte soluciones del tipo de onda viajera, lo cual fue lo que se tuvo que hacer después de analizar minuciosamente la teoría que se implica tanto matemáticamente como la física que se involucra, recordando que se esperaba una solución del tipo solitón.

De parte de la inteligencia artificial que en los últimos años ha sido de gran interés por parte de la comunidad científica, con el objetivo de resolver diferentes ecuaciones por medio de programas de computadora llegando al camino las redes neuronales que se comportan de manera semejante a una red neuronal natural tales como las del cerebro, sin embargo crear redes neuronales artificiales es una tarea que requiere de conocimientos de la programación, de la mente humana y también de la teoría matemática que se emplea en dicha área. Llegando a ser necesario saber cómo funcionan las diferentes herramientas de la programación.

Para que fuera posible utilizar una inteligencia artificial para resolver la ecuación modificada de Korteweg-De Vries, es necesario primeramente resolver la ecuación de manera analítica, luego de resolverla se crea un programa que grafique dicha solución la cual es una solución solitónica, después de que se

grafique la solución que es obtener los datos de cada punto en un dominio todo esto es hecho en Python. Simultáneamente utilizando PINN's que es el software utilizado para crear las redes neuronales, se crea un programa para generar dichas redes neuronales que a su vez utilizan diferentes optimizadores y librerías de Python para que se utilicen como conocimientos previos el primer programa de la solución analítica para que aprenda la red neuronal, después de que aprende se hace una comparación para ver la diferencia que hay sobre las redes neuronales con su aprendizaje y el método analítico. Finalmente se hace otra comparación de otro resultado menos adaptado a la red neuronal para así poder evaluar su estabilidad con respecto al uso de las redes neuronales para resolver ecuaciones diferenciales parciales no lineales.

Capítulo 1

Introducción general

1.1. Introducción

Muchas aplicaciones científicas e industriales requieren resolver ecuaciones diferenciales parciales (PDE) para describir los fenómenos físicos de interés. Se pueden encontrar algunos ejemplos en los campos de la aerodinámica, la astrodinámica, la combustión y muchos otros. En algunos casos excepcionales existe una solución analítica para las PDE, pero en la gran mayoría de las aplicaciones se debe calcular algún tipo de aproximación numérica. Las Redes Neuronales Artificiales (RNA) [1-5] son sistemas de procesamiento de la información cuya estructura y funcionamiento están inspirados en las redes neuronales biológicas. Las RNA constituyen una línea de investigación en Inteligencia Artificial (IA), la cual tiene como objetivo primario la construcción de máquinas inteligentes usando métodos de cibernética. Hoy en día existen

algunos enfoques basados en datos para resolver PDE con RNA. con inteligencia artificial (AI). En estos casos, se llevan a cabo muchas simulaciones para generar una gran cantidad de datos que luego se utilizan para entrenar RNA de forma supervisada. Este no es el enfoque que adoptamos, ya que nuestro objetivo es eliminar las costosas simulaciones del panorama. Nuestro enfoque no está supervisado y algunos trabajos que también siguen esta idea se pueden encontrar en la literatura. Recientemente, se presentaron las RNA profundas para estudiar las soluciones basadas en datos y el descubrimiento de parámetros de modelos físicos no lineales [6–23]. Particularmente, la técnica de redes neuronales informadas por la física (Physics Informed Neural Network, PINN) [8, 9, 10] se desarrolló como una aplicación prometedora de redes neuronales profundas para la solución numérica de ecuaciones diferenciales parciales no lineales (PDEs) [5–23]. Sin embargo, se ha reconocido que se necesitan procedimientos adaptativos para obligar a la red neuronal a ajustarse con precisión a las soluciones exactas de las PDEs. El método de PINN se basa en incrustar conocimientos previos sobre los sistemas físicos en una red neuronal de tal forma que restringe el espacio de posibles soluciones para abarcar sólo los tipos de soluciones que nos interesan.

Uno de los ejemplos de mayor importancia es la aplicación de PINN para estudiar las soluciones de la ecuación modifica de Korteweg - de Vries (mKdV)

$$v_t + 6\epsilon v^2 v_x + v_{xxx} = 0, \epsilon = \pm 1 \quad (1.1)$$

La ecuación mKdV aparece en muchos campos de física como redes anarmónicas [17], ondas de Alfvén, solitones acústicos de iones [18], modelos de congestión del tráfico [19], solitones acústicos iónicos [20], etc. Esta ecuación posee muchas propiedades notables como la transformación de Miura, las leyes de conservación, la transformación de dispersión inversa, soluciones N-solitónicas, soluciones de respiración tipo “breather”, transformación de Bäcklund, integrabilidad de Painlevé, transformación de Darboux, etc. [20–26].

En este trabajo de tesis estudiaremos los modelos de la ecuación modificada de Korteweg y de Vries a través de redes neuronales artificiales. Utilizamos el método de aprendizaje profundo de redes neuronales informadas por la física (PINN) para explorar las soluciones numéricas de estas ecuaciones.

1.2. Antecedentes históricos

En la mitad del siglo XIX, John Scott Russell fue comisionado por Channel Company para investigar una posible navegación usando propulsores de vapor en canales acuáticos. En 1834 descubrió en el Canal Unión en Herminston una onda solitaria, en 1844 publicó su hallazgo como Report Waves (reporte de ondas) a la Asociación Británica. En dicho documento relata que mientras realizaba pruebas con sus botes en el canal que mientras estaba observando el movimiento de un bote era jalado rápidamente a lo largo de un estrecho canal por unos caballos, de pronto se detiene el bote sin embargo la masa de agua se extiende hasta la proa del bote. Cabe mencionar que el canal de agua estaba bastante agitado, repentinamente la masa de agua empezó a moverse hacia delante con gran velocidad, tomando la forma de una gran elevación solitaria, redonda, suave y bien marcada. La cual continuó su curso a lo largo del canal visiblemente no cambió su forma ni disminuyó su velocidad, la siguió a caballo a una par de ocho o nueve millas sobre hora, sin ningún cambio en la forma unos treinta pies de largo y una amplitud de un pie y pie y medio. Al pasar un periodo de tiempo su amplitud fue disminuyendo gradualmente y a las dos millas de distancia la pierde en los recodos del canal. La cual llamó como Onda de Translación.

De su experiencia resaltó que una elevación solitaria que se propaga sin cambio alguno en su forma, dándole el nombre de "the great waves of translation" (gran onda de translación), pasando un tiempo se agregó el termino de onda



Figura 1.1: Recreación del canal donde Russell observó el solitón

solitaria, la razón de esto es por el tipo de movimiento de la onda que permanece sola. Este descubrimiento hizo que se interesara más, lo que lo llevó a hacer experimentos en el patio de su casa con un pequeño canal que mandó a construir.

Con la observación de Russell se descubre la existencia de ondas solitarias con un perfil localizado mientras que sus velocidades dependían de sus amplitudes, las ondas más altas viajaban más rápido que las ondas más bajas. Russell observó que las amplitudes, velocidades y formas mantienen sus características. Las ondas más altas atraviesan las más pequeñas como si no existieran, como todo tuvo sus críticas en la época debido que no contaban con la no linealidad y que la dispersión pudieran compensarse durante la evolución del sistema lo cual hacía que sus contemporáneos tuvieran sus falsas

especulaciones.

Las descripciones teóricas de los descubrimientos de Russell fueron hechas de manera independiente por Joseph Boussinesq (1871) y Lord Rayleigh (1876). Una de las principales contribuciones a la teoría de solitones consiste en la suposición de que el incremento de la velocidad de onda es compensando con el decremento asociado a la dispersión siendo la conservación de la onda.

En 1895 Diderick Johansen Korteweg y su estudiante Gustav de Vries, encontraron una ecuación (después llamada la ecuación KdV) que describe exactamente a la gran onda de translación de Russell, con el perfil clásico de tipo "campana" $sech^2(f(x, t))$. Ellos estudiaron las ecuaciones hidrodinámicas de Navier Stokes describiendo la propagación de ondas unidimensionales, las cuales su longitud es mayor que la altura del líquido. Korteweg y de Vries encontraron una ecuación dinámica aproximada que describe la propagación en una dirección de estas ondas.

Con el pasar del tiempo dicha ecuación se hizo conocer como la ecuación Korteweg de Vries, la cual ha sido utilizada en procesos dinámicos de gases y en física de plasmas. Desde ese tiempo hasta la actualidad el solitón descubierto por Russell ha estado jugando un papel muy valioso en los fenómenos naturales. Por mencionar más ejemplos uno de ellos es en la fisiología cuando se encontraron ondas que se transmitían de una forma muy parecida a solitones y también para la localización y movimiento de energía vibrante en las proteínas propuesto por Davitov (1985).

1.2.1. Inteligencia Artificial

La inteligencia artificial es una de las ciencias más nuevas de todas. Comienza después de terminar la segunda guerra mundial y se nombra hasta 1956, actualmente la inteligencia artificial (AI) tiene un gran campo de investigación que van desde los usos generales como el aprendizaje y la percepción hasta tareas muy específicas como ajedrez, demostrar teoremas matemáticos, escribir poesía, diagnosticar enfermedades. La AI sistematiza y automatiza tareas intelectuales por lo cual es potencialmente relevante en cualquier actividad humana siendo un campo universal.

La definición de la AI varia, algunos autores la definen con procesos de razonamiento y el pensamiento mientras que otros abordan el comportamiento, midiendo el éxito en términos de la fidelidad humana llegando incluso a buscar la racionalidad midiendo un concepto ideal de inteligencia dando un nuevo concepto de la racionalidad que es un sistema racional aquel que hace lo correcto porque sabe que es lo correcto. llegando a cuatro objetivos de sistemas que piensan como humanos, sistemas que piensan racionalmente, sistemas que se comportan como humanos y sistemas que actúan racionalmente que a su vez se separan en dos partes una centrada en lo humano aproximándose a una ciencia empírica envuelta de hipótesis y experimentos, mientras que la parte racional involucra una combinación matemática e ingeniería cada uno de estos grupos se ayudan mutuamente a pesar de todo.

Actuando humanamente fue con la prueba de Alan Turing (1950), estaba di-

señado para proveer una operación satisfactoria de definición de inteligencia. El sugiere una prueba basada en los indistinguible de entidades innegablemente inteligentes, la computadora pasa la prueba de un interrogador humano que hace las preguntas escritas no puede decir si las respuestas escritas provienen de una persona o no, si una computadora es inteligente pasa la prueba. La computadora debería poseer un procesamiento natural del lenguaje para comunicarse efectivamente, una representación del conocimiento que percibe ya sea para almacenar lo que sabe o lo que escucha, razonamiento automatizado que se relaciona con usar la información almacenada para contestar preguntas y crear nuevas conclusiones, aprendizaje automatizado para adaptar a nuevas circunstancias y detectar y extrapolar patrones. La prueba de Turing evade directamente la interacción física entre el interrogador y la computadora porque la simulación física de una persona es innecesaria para la inteligencia, sin embargo, la prueba total de Turing incluye una señal de video tal que el interrogador puede probar las habilidades perceptivas subjetivas, siendo como bien una oportunidad de que el interrogador pase físicamente los objetos a través de la .^{es}cotillaz aun así la computadora necesitaría visión para percibir objetos y robótica para manipularlos.

Pensando humanamente (un acercamiento de modelo cognitivo) debe ser principalmente para determinar cómo piensan los humanos, hay dos caminos para lograr esto en una computadora, la primera es a través de la introspección que se refiere a tratar de atrapar los propios pensamientos como vienen

y la segunda es a través de experimentos psicológicos. Una vez teniendo la suficiente y precisa información de la teoría de la mente, es posible expresar esto en un programa de computadora, si las entradas y salidas del programa se sincronizan sus comportamientos y estos corresponden con los comportamientos humanos es evidencia de que algunos mecanismos de los programas de computadora pueden ser operados en humanos. El campo interdisciplinario de la ciencia cognitiva trae consigo modelos computacionales de AI y técnicas psicológicas experimentales para intentar construir teorías precisas y probables de los trabajos de la mente humana. Los dos campos siguen desarrollándose mutuamente, especialmente en áreas de visión y de lenguaje natural.

Pensando racionalmente, para comprender más esta parte se debe remontar al filósofo griego Aristóteles que fue el primero que sentó la codificación del "correcto pensamiento que son un proceso de razonamientos irrefutables. Sus silogismos proveen patrones para la estructura de argumentos que siempre llevan a conclusiones correctas cuando se dan premisas correctas para ejemplificar se da un caso "La naranja es una fruta; todas las frutas cascara; por lo tanto, la naranja tiene cascara". Estas leyes de pensamiento son supuestas para gobernar las operaciones de la mente, su campo de estudio es nombrado lógica, dicha lógica ayuda para crear programas de AI. Hay dos obstáculos de este acercamiento, el primero es que no es fácil tomar un conocimiento informal estableciéndolo en términos de uno formal ya que requiere notación

lógica, ya que hay conocimientos menos del 100% de certeza. La segunda es que hay una gran diferencia entre resolver un problema en principio que hacerlo en práctica. Incluso algunos programas con una docena de hechos pueden fatigar los recursos computacionales de cualquier computadora a menos que tenga una guía de pasos de razonamiento para intentar primero. De cualquier forma, estos obstáculos se aplican en construir cualquier sistema de razonamiento computacional.

Actuando racionalmente, se espera que tengan otros atributos que los distingan de otros programas, tal que estas tengan operaciones de control autónomos percibiendo su medio, persistiendo sobre un periodo largo de tiempo, adaptado al cambio y siendo capaces de tomar otras metas. Un agente racional es aquel que actúa para lograr el mejor resultado o bien cuando hay incertidumbre el mejor resultado esperado.

El énfasis está sobre inferencias correctas, hacer correctas inferencias es a veces parte de un agente racional, porque un camino para actuar racionalmente es razonar lógicamente hasta la conclusión de que una acción determinada logrará los objetivos, y luego actuar sobre esa conclusión. Por otro lado, la correcta inferencia no es del todo racionalidad, porque aparecen situaciones donde no hay probabilidad de hacer lo correcto y todavía habrá algo que hacer. También hay formas de actuar racionalmente de las que puede que no impliquen inferencia.

El estudio de AI como diseño de agente racional tiene al menos dos ventajas.

En primer lugar, es más general que el enfoque de las "leyes del pensamiento", porque la inferencia correcta es simplemente uno de varios mecanismos posibles para lograr la racionalidad. En segundo lugar, es más susceptible a desarrollo científico que los enfoques basados en el comportamiento o el pensamiento humanos porque el estándar de racionalidad está claramente definido y es completamente general. Comportamiento humano, por otra parte, está bien adaptado a un entorno específico y es producto, en parte, de un proceso evolutivo complicado y en gran parte desconocido que todavía está lejos de producir perfección.

1.2.2. Programación

El programa de computadora es un conjunto independiente de instrucciones usado para operar una computadora con el fin de producir un resultado específico, algunas veces es un conjunto de programas se convierte en un software. Es importante conocer el lenguaje ensamblador ya que es uno de los primeros avances en la programación que fue la sustitución de símbolos en forma de palabras como ADD SUB, MUL, por los operadores binarios y los números decimales y etiquetas por las direcciones en memoria. Debido a que las computadoras solo pueden ejecutar programas en lenguaje de máquina, el conjunto de instrucciones contenido dentro de un programa en lenguaje ensamblador.

El proceso de escribir un programa o software se llama programación, mientras que al conjunto que puede usarse para construir un programa se llama lenguaje de programación.

Es vital saber que existen lenguajes de nivel alto los cuales usan instrucciones que se parecen a los lenguajes escritos, como el inglés, pueden ejecutarse en una variedad de tipos de computadora.

Los programas en un lenguaje de computadora se conocen como programas fuentes y código fuente. Una vez que se ha escrito un programa en un lenguaje de alto nivel también debe traducirse, como un programa ensamblador de bajo nivel, al lenguaje de máquina de la computadora en que se va a ejecutar. Un lenguaje interpretado es cuando cada declaración en un programa fuente

de alto nivel es traducido de manera individual y ejecutada inmediatamente después de la traducción, el lenguaje de programación usado se llama lenguaje interpretado y el programa que hace la traducción se interprete.

Cuando todas las instrucciones en un programa fuente de alto nivel son traducidas como una unidad completa antes que cualquier declaración sea ejecutada, el lenguaje de programación usado el programa que hace la traducción se llama compilador.

1.2.3. Python

Fue creado por Guido Van Rossum en los años 90, similar a Perl, pero con una sintaxis más limpia y código legible. Es un lenguaje interpretado, de tipado dinámico fuertemente, multiplataforma y orientado a objetos.

Su lenguaje interpretado se ejecuta utilizando un programa intermedio llamado interprete, en lugar de compilar el código a lenguaje máquina que pueda comprender y ejecutar directamente una computadora (lenguajes compilados). En si los lenguajes compilados son de ejecución más rápida y los lenguajes interpretados son más flexibles y más portables.

Al mencionar un tipado dinámico se refiere a que no es necesario declarar variables, sino que su tipo se determinará en tiempo de ejecución según el tipo de valor al que se asigne y cambia si se le asigna un valor de otro tipo.

El que sea fuertemente tipado es que no se permite tratar a una variable como si fuera de un tipo distinto al que tiene, más bien, se debe convertir de

forma explícita al nuevo tipo antes. Puede que en otros se pueda cruzar lo cual provoca errores.

Tratándose la multiplataforma en diferentes sistemas operativos y por las librerías específicas no usadas, no cambia mucho, al ser orientado a objetos provoca que los conceptos del mundo real relevantes se trasladan a clases y objetos en programa.

La ejecución va en una serie de interacciones entre los objetos, programas imperativos, funcional y orientada a aspectos con sintaxis cercana al lenguaje natural.

1.2.4. Redes neuronales

Físicos como John Hopfield (1982) usó técnicas de la mecánica estadística para analizar el almacenaje y la optimización de propiedades de redes, tratando de colecciones de nodos como colecciones de átomos. A mediados de los años 1980 cuando al menos cuatro diferentes. Los grupos reinventaron el algoritmo de aprendizaje de retro propagación encontrado por primera vez en 1969 por Bryson y Ho. El algoritmo se aplicó a muchos problemas de aprendizaje en informática y psicología, y la amplia difusión de los resultados en la colección Parallel Distributed. El procesamiento (Rumelhart y McClelland, 1986) causó gran entusiasmo.

Una neurona es una célula en el cerebro las cuales su principal función es la de recolectar, procesar y diseminar las señales eléctricas. La capacidad de

procesar información del cerebro es pensar para emerger primariamente de redes de tales neuronas. Algunos trabajos recientes de AI es crear redes neuronales artificiales. Se han desarrollado modelos detallados y realistas, tanto para neuronas como para sistemas más grandes. En el cerebro, lo que condujo al campo moderno de la neurociencia computacional. Por otro lado, Los investigadores de AI y estadística se interesaron por las propiedades más abstractas de las neuronas. redes, como su capacidad para realizar computación distribuida, tolerar entradas ruidosas, y aprender. Aunque ahora entendemos que otros tipos de sistemas, incluido el bayesiano, Las redes neuronales tienen estas propiedades, las redes neuronales siguen siendo una de las más populares y efectivas. formas de sistema de aprendizaje y son dignos de estudio por derecho propio.

Las redes neuronales poseen unidades o nodos, conectados por enlaces directos, un enlace puede venir de una unidad j a una unidad i sirviendo para propagar la activación a_j de j a i . Cada enlace tiene un peso numérico W asociado con eso, el cual determina la fuerza y señal de conexión. cada unidad i computa la suma de los pesos de entrada, luego se aplica una función de activación g a esta suma que deriva en la salida. También se incluye un peso de sesgo conectado a una entrada fija. La función de activación g es designada para conocer dos posibles deseos, primero para activarla cerca $+1$ cuando la correcta se introduce y la inactiva (cerca de 0) cuando el incorrecto se introduce; la segunda la activación necesita ser no lineal, en otras pala-

bras, la entera red neuronal colapsa a una simple función lineal. La función logística y la función sigmoidea, la función sigmoidea tiene la ventaja de ser diferenciable.

Estructuras de redes, hay dos tipos de estructuras de redes, las acíclicas y las cíclicas, las acíclicas representan una función de su entrada actual por lo cual no tiene un estado interno más que los propios pesos, las cíclicas se alimenta de salidas que regresan hacia sus propias entradas, es decir, que los niveles de activación de la red forman un sistema dinámico que puede alcanzar un estado estable o exhibir oscilaciones o incluso comportamiento caótico. Además, la respuesta de la red a un determinado La entrada depende de su estado inicial, que puede depender de entradas anteriores. Una red acíclica usualmente es usada en arreglos de capas, tal que cada unidad recibe una entrada solo inmediatamente de las unidades que preceden de la capa.

Las múltiples capas ocultas de las redes tienen la ventaja de que al agregar capas ocultas es que amplía el espacio de hipótesis que la red puede representar. Se piensa en cada unidad oculta como un perceptrón. que representa una función de umbral suave en el espacio de entrada. Luego, piense en una unidad de salida como una combinación lineal con un umbral suave de varias de estas funciones. Por ejemplo, agregando dos funciones de umbral suave opuestas y umbralizando Como resultado, se puede obtener una cresta". Combinando estas dos crestas al ángulo correcto de cada una, se obtiene un bulto.

Con más unidades ocultas de capas se pueden producir más bultos de diferentes tamaños en más espacios, incluso, con una única capa oculta suficientemente grande, es posible representar cualquier función de las entradas con precisión arbitraria; con dos capas, incluso funciones discontinuas se puede representar. Por supuesto, las redes neuronales son capaces de realizar tareas de aprendizaje mucho más complejas, aunque. Hay que decir que se necesita cierta cantidad de retoques para lograr que la estructura de la red sea correcta. y lograr la convergencia hacia algo cercano al óptimo global en el espacio de peso.

1.3. Objetivos

1.3.1. Objetivo general

Como objetivo principal de la tesis es la investigación de las soluciones de la ecuación modificada de Korteweg - de Vries aplicando el método de red neuronal informada por la física (PINN) y verificando que el método tiene una estabilidad y eficacia adecuadas.

1.3.2. Objetivos particulares

- 1) Estudiar la ecuación (1) y encontrar sus soluciones exactas que corresponden a soluciones tipo un solitón.
- 2) Estudiar el método de aprendizaje profundo PINN para las ecuaciones diferenciales parciales.
- 3) Representar la ecuación (1) en la forma apropiada para las PINNs con los parámetros de entrenamiento en una red neuronal profunda.
- 4) Usamos el método de aprendizaje profundo PINN y el lenguaje Python para aproximar las soluciones solitónicas.

1.4. Desarrollo

1.4.1. Ecuaciones diferenciales parciales

Una ecuación diferencial parcial es una ecuación que involucra una función de varias variables desconocidas y sus derivadas parciales las cuales tienen una función desconocida que depende de dos variables por ejemplo.

$$\frac{\partial u}{\partial t} + 3\frac{\partial u}{\partial x} = 2u$$

Dónde la función desconocida es u que depende de x y t . Puede haber funciones desconocidas que dependan de más variables e incluso de orden superior por ejemplo.

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} - u = \frac{\partial u}{\partial t}$$

Una solución de la ecuación diferencial parcial es una función $u=u(x,t)$ continua y diferenciable, así que tiene sentido calcular su derivada en el orden que es, es decir, que si es de primer orden se puede calcular su primera derivada, si es de segundo orden se puede calcular su segunda derivada parcial y estas a su vez sustituir en la ecuación convirtiéndola en una identidad, a esto se le conoce como una solución suave, también conocida como solución clásica o solución genuina.

También puede que se tengan soluciones que tengan discontinuidades en sus

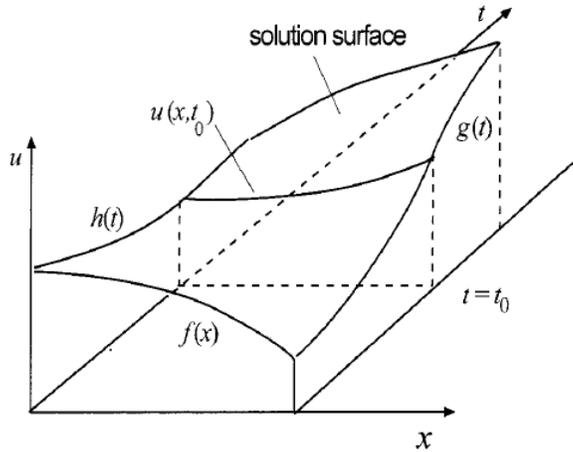


Figura 1.2: $u(x,t)$ es la evolución foto de pulso en un instante

derivadas, estas reciben el nombre de soluciones débiles.

Existe un dominio $D(x,t)$ donde el problema es definido, el cual es llamado dominio espacio-tiempo y cuando t representa el tiempo la ecuación diferencial parcial, es llamada ecuación de evolución.

Cuando las dos variables independientes son ambas variables espaciales, la ecuación diferencial parcial es llamada ecuación de equilibrio o de estado estacionario.

Graficamente una solución $u=u(x,t)$ es una superficie suave en el espacio tridimensional $x-t-u$ que se encuentra sobre el dominio D en el plano xt F y h son valores predeterminados de u condiciones iniciales y de frontera.

Una ecuación diferencial parcial tiene una infinidad de soluciones dependen-

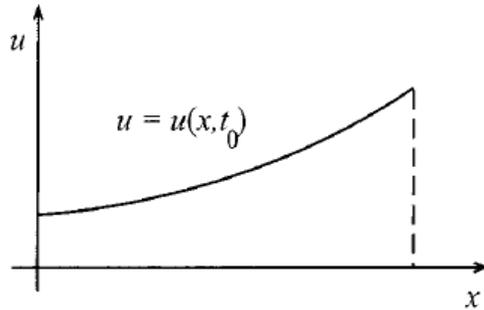


Figura 1.3: Forma alterna de foto de pulso en el tiempo t_0 de funciones arbitrarias.

1.4.2. Ondas

La ecuación diferencial parcial más simple es la ecuación de onda estacionaria

$$\frac{\partial u}{\partial t} = 0, u_t = 0$$

La cual se resuelve por teorema fundamental del cálculo siendo una solución $f(x)$ con f siendo función, generalmente una onda implica la propagación de una cierta perturbación siendo esta independiente del tiempo, ya que su perfil queda igual.

Para obtener la ecuación de movimiento de la onda se toma en cuenta dos

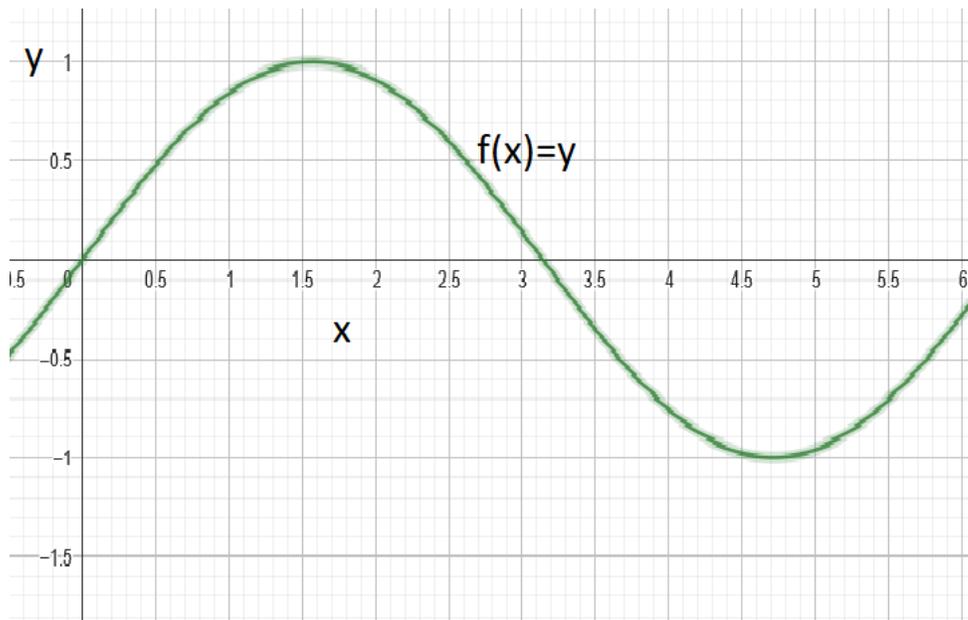


Figura 1.4: Onda estacionaria la cuadrícula va en función $f(x, t_0)$

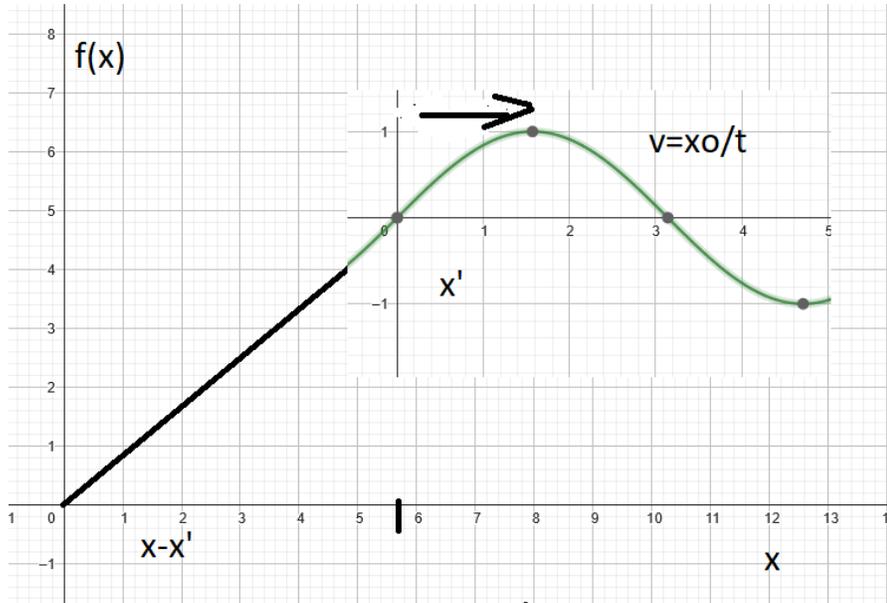


Figura 1.5: Esquema de onda viajera

aspectos, el analítico y el físico es decir

$$F(x) = \Psi(x)$$

Con el esquema 1.5 se puede deducir que

$$\frac{\partial u(x,t)}{\partial x} = \frac{\partial F}{\partial x'} \frac{\partial x'}{\partial x}$$

Siendo $\frac{\partial F}{\partial x'}$ un sistema fijo ya que la derivada parcial de x' con respecto de x se anula y $x' = x - vt$ y el sistema inercial es el lado izquierdo de la ecuación.

$$\frac{\partial \Psi}{\partial t} = \frac{\partial F}{\partial x'} \frac{\partial x'}{\partial t}$$

Ya que al derivar x' con respecto de t se obtiene que es igual a $-v$ convirtiendo la ecuación en:

$$\frac{\partial \Psi}{\partial t} = -v \frac{\partial \Psi}{\partial x'}$$

Por otro lado se tiene que

$$\frac{\partial \Psi}{\partial x} = \frac{\partial \Psi}{\partial x'}$$

Sustituyendo ambas ecuaciones de su lado derecho se obtiene

$$\frac{\partial \Psi}{\partial x} = \pm \frac{1}{v} \frac{\partial \Psi}{\partial t}$$

Al hacer sus segundas derivadas parciales de cada ecuación se obtiene el perfil de onda viajera, la cual es:

$$\frac{\partial^2 \Psi}{\partial x^2} = \frac{1}{v^2} \frac{\partial^2 \Psi}{\partial t^2}$$

Otra manera de obtener la ecuación de onda viajera es utilizando el sistema de cuerda discretizada la cual consiste en dividir la cuerda en distintos segmentos de cuerda de longitud pequeña, modelando su comportamiento por medio de sistemas de partículas acopladas.

Tomando la cuerda dividida en n segmentos y cada segmento representa

una partícula teniendo una posición q_j , las interacciones de cada segmento solamente tomará la partícula mas cercana de la posición j . Dicha partícula será influenciada por la $(j-1)$ esima y de la $(j+1)$ esima partículas.

Sumando la energía cinética y potencial por las aproximaciones de la ley de Hooke y considerando que todos los péndulos que surgen son de la misma masa, el lagrangiano será

$$L = T - U = \frac{1}{2}m \sum_j \left(m \left(\frac{dq_j}{dt} \right)^2 + \frac{k}{\delta} (q_{j-1} - q_j)^2 \right)$$

Donde k surge de la constante de los resortes y δ la separación de las posiciones aledañas. La ecuación de Lagrange de este sistema da cero, por lo que al usar el límite cuando δ tiende a cero se obtiene

$$\frac{\partial^2 q}{\partial t^2} - a^2 \frac{\partial^2 q}{\partial x^2} = 0, \text{ con } \frac{k}{m} = a^2$$

Una onda representa la forma en que se propaga una perturbación. La ecuación de onda viajera pertenece a la familia de ecuaciones hiperbólicas que estan relacionadas a líneas del espacio-tiempo siendo del estilo características.

La solución general de la ecuación de onda es una combinación lineal de dos perfiles de onda que se mueven en direcciones opuestas en la cuerda, semejante al esquema (1.5)

$$q = f(x - at) + g(x + at)$$



Figura 1.6: Fotografía de una onda mecánica en agua siendo vista su sombra

Tomando el caso de que el movimiento ondulatorio ocurre bajo una sola frecuencia $\nu = \omega/2\pi$ convirtiéndose en solo la parte real en la siguiente ecuación

$$q = A\cos(\kappa x - \omega t + \alpha) + B\sin(\kappa x + \omega t + \beta)$$

A y B son las amplitudes, α, β constantes arbitrarias que surgen de las condiciones iniciales, usualmente la siguiente expresión es la solución completa y compleja para la ecuación de onda.

$$q = ae^{i(\kappa x - \omega t)} + be^{i(\kappa x + \omega t)}$$

Mencionando un caso particular cuando el observador está junto a la onda y se mueve con cierta velocidad c constante, teniendo una fase θ y la cual es constante con respecto al observador. La velocidad de fase determina la velocidad para una oscilación por lo que se puede deducir que

$$\frac{d\theta}{dt} = \frac{\partial\theta}{\partial t} + \frac{dx}{dt} \frac{\partial\theta}{\partial x} = 0$$

Puede existir un caso en el que se relacione la frecuencia ω con el número de onda κ siendo

$$\omega = F(\kappa)$$

Llevando a que la velocidad de fase dependerá del número de onda κ tal que

$$c(\kappa) = \frac{\omega}{\kappa} = \frac{F(\kappa)}{\kappa}$$

Si la velocidad de fase es c constante, la onda no será dispersiva siendo $F(\kappa) = \kappa c$, la velocidad de fase si depende del número de onda, entonces, la onda se llama onda dispersiva. Cuando las ondas se propagan con una velocidad de fase constante en un tiempo cero, se puede representar la onda por medio de integrales de Fourier como una superposición de funciones sinusoidales ya que es un sistema lineal.

Al derivar con respecto de κ a la ecuación de dispersión es decir

$$\frac{d\omega}{d\kappa} = J(\kappa)$$

Se obtiene la velocidad de grupo, ya que posee dimensión de velocidad, esto se da porque cada parte envolvente de la oscilación puede ser visto como un grupo o un paquete de ondas que se mueven a cierta velocidad. Cuando las ondas se propagan sin dispersión la velocidad de grupo y la velocidad de fase son iguales, cuando son ondas dispersivas las velocidades de grupo son diferentes de la velocidad de fase pudiendo ser de mayor velocidad o menor velocidad. La única condición de que ocurran dichos fenómenos es que la longitud de onda sea mucho menor que la dimensión del promedio de paquete

de ondas. Recordando que la propagación de ondas su energía esta expresada en unidades de longitud.

1.4.3. Ecuaciones diferenciales parciales no lineales

El criterio más importante para la clasificación de las ecuaciones diferenciales parciales es distinguirla como lineales o no lineales.

Una ecuación diferencial parcial homogénea (las que son iguales a 0) es lineal si la suma de dos soluciones es solución y un múltiplo constante de una solución es una solución. En caso contrario será no lineal. La linealidad o no linealidad generalmente se define en términos de las propiedades del operador que define la ecuación diferencial parcial.

si $G(x, t, u, u_x, u_t, u_{xx}, u_{tt}, u_x t) = 0$ y esta se puede escribir de la forma

$$L_u = F, \text{ donde } F = F(x, t)$$

Por ejemplo:

$$U_{tt} - U_{xx} = 0$$
$$\left[\frac{\partial^2}{\partial t^2} - \frac{\partial^2}{\partial x^2} \right] [U] = 0$$

L es un operador en derivadas parciales por lo que $L[U] = 0$

L es lineal si $L(u + v) = L_u + L_v$ y $L(CU) = CL(U)$. La ecuación diferencial parcial es lineal si L es un operador lineal, en caso contrario no es lineal.

Siguiendo el ejemplo

$$U_t + UU_x = 0$$

$$LU = U_t + UU_x = 0$$

$$LC(U) = CU_t + CUC(U_x) = CU_t + C^2UU_x$$

El cual es distinto de

$$CL(U) = C[U_t + UU_x] = CU_t + CUU_x$$

Por lo que es no lineal.

El hecho de que el operador L sea lineal implica que la ecuación lineal homogénea tenga la propiedad de que si U_1, U_2, \dots, U_n son soluciones de la ecuación diferencial parcial entonces $U = C_1U_1 + C_2U_2 + \dots C_nU_n$ también es solución, lo cual esto es conocido como principio de superposición de soluciones.

Para las ecuaciones diferenciales parciales no lineales no se puede superponer soluciones. La mayoría de los métodos de solución de problemas lineales no son aplicables a las ecuaciones no lineales. Algunos métodos que se pueden desarrollar son:

1. Métodos de perturbación los cuales se obtienen soluciones aproximadas como un desarrollo en serie de algún parámetro.
2. Métodos de similitud o similaridad que se basa en que las ecuaciones diferenciales parciales y sus condiciones son invariantes bajo una familia de transformaciones la cual permite un cambio de variables el cual reduce la ecuación diferencial parcial a una ecuación diferencial ordinaria o se reduce el orden de la ecuación diferencial parcial.
3. Método de las características las ecuaciones hiperbólicas no lineales pueden ser analizadas con éxito en coordenadas características.

4. Transformaciones las cuales en algunas ocasiones es posible identificar transformaciones que cambien la ecuación diferencial parcial no lineal en una ecuación diferencial parcial simple.
5. Métodos numéricos que se basan en aproximaciones de varias iteraciones con cierto grado de exactitud.
6. Soluciones de ondas viajeras la cual consiste en buscar soluciones con propiedades especiales.
7. Soluciones de estado estacionario y estabilidad (independientes del tiempo)
8. Métodos de Ad Hoc que son métodos especiales para un cierto tipo de ecuaciones en derivadas parciales.

1.4.4. Ondas no lineales

En el caso de las ondas no lineales al momento de adicionar dos ondas no siempre da el resultado de que se sumen sus características, tales como su amplitud, fase y/o frecuencia, normalmente cuando son ondas no lineales la suma de dos elementos crea nuevos efectos como consecuencia de su actividad colectiva. Es una propiedad cooperativa y se manifiesta como resultado de fuerzas antagónicas.

Analizando el fenómeno lineal de transporte de ondas que se puede escribir mediante la ecuación lineal nuevamente escrita

$$\left(\frac{\partial}{\partial t} - a\frac{\partial}{\partial x}\right)\left(\frac{\partial}{\partial t} + a\frac{\partial}{\partial x}\right)q(x, t) = 0$$

Tomando solo dos factores para ser una ecuación lineal simple

$$\left(\frac{\partial}{\partial t} + a\frac{\partial}{\partial x}\right)q(x, t) = 0$$

Generalizando la ecuación anterior con una suposición no lineal del movimiento ondulatorio, lo cual convierte a la velocidad de fase dependa de la amplitud de onda haciendo que

$$\left(\frac{\partial}{\partial t} + a(q)\frac{\partial}{\partial x}\right)q(x, t) = 0$$

Suponiendo que la no linealidad sea más sencilla de tratar haciendo que la velocidad de fase depende de la onda de forma lineal, es decir que

$$a(q) = a_0 + \alpha q + \dots$$

Convirtiendo la ecuación no lineal en

$$\frac{\partial q}{\partial t} + a_0\frac{\partial q}{\partial x} + \alpha q\frac{\partial q}{\partial x} = 0$$

Proponiendo un cambio de variable tal que $x' = x - a_0t$, $t' = t$ con $q' = \alpha q$, convirtiendo la ecuación en una mas sencilla la cual es:

$$q_t + qq_x = 0$$

Siendo su solución del tipo de onda viajera cuando $a = a_0$

$$q(x, t) = g(x - a_0 t)$$

Al hacer las derivadas parciales de la solución q con respecto de x y t , se encuentran las velocidades proporcionales a la altura del perfil, cuando $q=0$, permanecen sin cambio ya que se realizó un cambio al sistema de referencia que se mueve con respecto a_0 . Por lo que en el avance del tiempo se comportará diferente el perfil de onda.

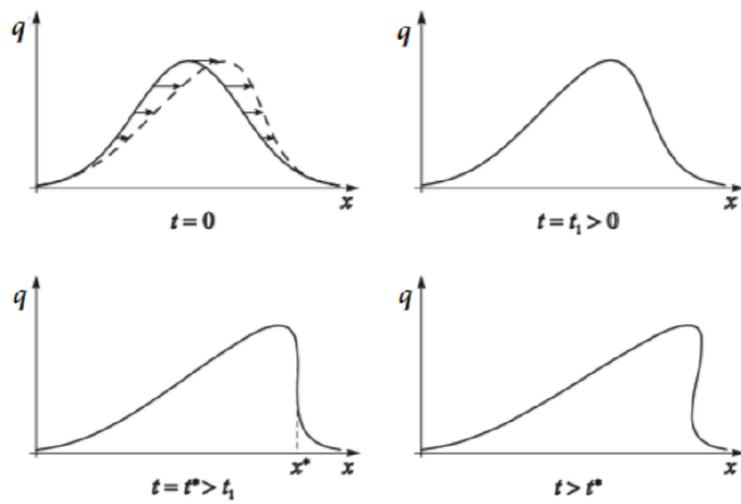


Figura 1.7: Onda no lineal sin dispersión propagándose donde se deforma su cresta, ya que su velocidad es mayor que las otras partes

Se aprecia que la inclinación del borde de la perturbación aumento en la figura (1.7). Dicho fenómeno es conocido como reforzamiento de onda. Con el tiempo el perfil de onda se va empinando más hasta que la derivada de q con respecto de x se hace infinita. Esto ocurre en el tiempo t^* . Luego la onda se inclina y su perfil se vuelve ambiguo lo cual la ecuación lineal simple ya no es viable.

La onda se vuelca muy semejante con las olas del océano en las playas especialmente, se puede apreciar que algunas partes de la onda viajan más rapido que otras creando un rompimiento más lejano de la onda creando a su vez ondas de choque.

1.4.5. Dispersión

Es un efecto que se encuentra en todos los medios causando un gran problema por la pérdida continua de información. El cual depende de la longitud de onda o su frecuencia, teniendo en cuenta que el paquete de ondas al inicio tiene la misma velocidad, al pasar un tiempo algunas ondas disminuyen su velocidad ocasionando que cada onda tenga una fase diferente haciendo que el paquete inicial se deforme y por ende se disperse. La dispersión depende en algunos casos del medio en el que se propague.

La velocidad de grupo es igual a la velocidad de fase, entonces el sistema es no dispersivo, caso contrario si sería dispersivo. Las relaciones de dispersión toman la forma de funciones complejas en su número de onda, las cuales las

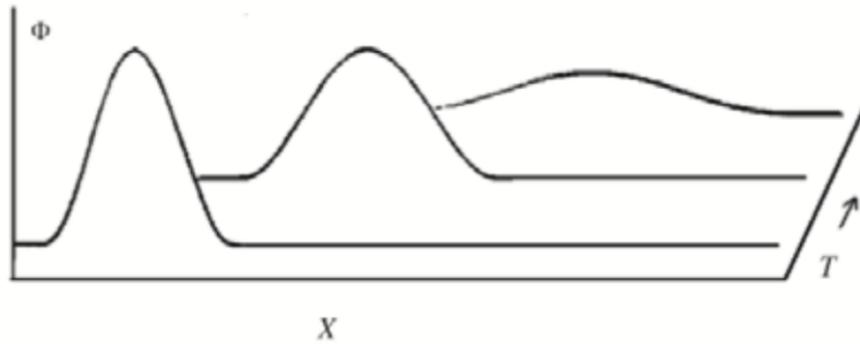


Figura 1.8: Onda lineal propagándose que pierde su perfil al punto de desvancerse mientras avanza en el tiempo

fuerzas dispersivas pueden ser compensadas por fuerzas opuestas no lineales en una ecuación diferencial donde estén presentes creando un balance que genera las soluciones solitónicas. Es importante mencionar que la dispersión no es igual a disipación de energía puesto que el contenido de energía puede permanecer constante sin importar que el pulso se disperse.

1.4.6. Integrabilidad

Un sistema integrable es aquel sistema que posee la representación de Lax (o dicho como la representación de la curvatura cero), que tiene un número contable de integrales de movimiento. Para investigar la dinámica de este sistema uno puede aplicar el método de transformada espectral inversa, problema de Riemann y el método de integración de la zona finita. A este tipo de integrabilidad se le llama integrabilidad-S. Hay otro tipo de integrabilidad que es la integrabilidad-N que se refiere a los sistemas dinámicos pudiendo ser resueltos por un cambio de variables convenientes o por un enfoque adecuado.

Dicho sistema integrable se le conoce como sistema integrable hamiltonico, para el cual es posible encontrar las variables canónicas de acción y del ángulo teniendo chance de poder reescribir el hamiltoniano del sistema en términos de estas variables.

1.4.7. Método de dispersion inversa

El método de dispersión inversa permite resolver ecuaciones diferenciales parciales no lineales que son totalmente integrables el cual consiste en encontrar o determinar un sistema de ecuaciones espectrales asociadas a la ecuación no lineal y con base en una solución particular generar nuevas soluciones de tipo solitón. Dichas ecuaciones espectrales son ecuaciones diferenciales lineales, de las que se obtienen pares de Lax, que al ser integradas precisamente tienen

como condición a la misma ecuación diferencial no lineal a resolver.

En las ecuaciones diferenciales parciales no lineales en las que se aplicó por primera vez este método fueron en la ecuación de Korteweg-de Vries y la de Sine-Gordon, estas ecuaciones admiten soluciones del tipo de onda solitónica. También se puede aplicar para resolver ecuaciones de orden superior como la ecuación de Einstein para gravitación (Belinski y Verdaguer, 2005).

La ecuación de Korteweg-de Vries $u_t + 6uu_x + u_{xxx}$ describe el movimiento de fluidos en medios de poca profundidad. Específicamente, describe la evolución en la transferencia de calor a través de fluidos (por convección). Cuando la dispersión de ondas de este flujo y la convección entran en un cierto equilibrio se forma una onda solitaria, admitiendo soluciones solitónicas.

1.4.8. Solución exacta

Para deducir la solución de tipo solitón, será que si la ecuación modificada de KdV ecuación (1.1), permite soluciones de ondas viajeras las cuales son:

$$v(x, t) = v'(z), z = x - ct,$$

donde c es una constante positiva y $v'(z)$, entonces debe ser continua y doble diferenciable en \Re

$$v_t = \frac{dv'}{dz} \frac{dz}{dt} = -c \frac{dv'}{dz} \quad (1.2)$$

Sabemos que

$$v_x = \frac{dv'}{dz} \frac{dz}{dx} = \frac{dv'}{dz}, v_{xx} = \frac{d^2v'}{dz^2}, v_{xxx} = \frac{d^3v'}{dz^3} \quad (1.3)$$

Sustituyendo las derivadas de v en la ecuación modificada de KdV, se obtiene

$$0 = v_t + 6\epsilon v^2 v_x + v_{xxx} = -c \frac{dv'}{dz} + 6\epsilon v'^2 \frac{dv'}{dz} + \frac{d^3v'}{dz^3} \quad (1.4)$$

Nótese que se vuelve una ecuación diferencial ordinaria.

integrando con respecto a z se obtiene

$$-cv' + \frac{6}{3}v'^3 + \frac{d^2v'}{dz^2} - B = 0 \quad (1.5)$$

Simplificando

$$-cv' + 2v'^3 + \frac{d^2v'}{dz^2} - B = 0 \quad (1.6)$$

B es una constante de integración, multiplicando por $\frac{dv'}{dz}$ se obtiene

$$-cv' \frac{dv'}{dz} + 2v'^3 \frac{dv'}{dz} + \frac{d^2v'}{dz^2} \frac{dv'}{dz} = B \frac{dv'}{dz} \quad (1.7)$$

Volviendo a integrar con respecto a z , se obtiene

$$\frac{-c}{2}v'^2 + \frac{2}{4}v'^4 + \frac{1}{2}\left(\frac{dv'}{dz}\right)^2 = Bv' + C. \quad (1.8)$$

Para encontrar las constantes B y C necesita tener que cuando $z \rightarrow \pm\infty$, por lo que

$v' \rightarrow 0$, $\frac{dv'}{dz} \rightarrow 0$ y $\frac{d^2v'}{dz^2} \rightarrow 0$ Para que las constantes B y C tomamos en cuenta que son cero, entonces.

$$-cv'^2 + v'^4 + \left(\frac{dv'}{dz}\right)^2 = 0 \quad (1.9)$$

Despejando $\left(\frac{dv'}{dz}\right)^2$

$$\frac{dv'}{v'\sqrt{c-v'^2}} = \pm dz \quad (1.10)$$

Se integrará ambas partes

$$\int \frac{dv'}{v'\sqrt{c-v'^2}} = z - z_0 \quad (1.11)$$

z_0 se tomará como el valor inicial. Integramos con tabla esta integral dando como resultado.

$$\int \frac{dv'}{v'\sqrt{c-v'^2}} = -\frac{1}{\sqrt{c}}\text{arcsech}\left(\frac{v'}{\sqrt{c}}\right) \quad (1.12)$$

Se obtiene una solución secante hiperbólica

$$v'(x, t) = \pm\sqrt{c}\text{sech}(\sqrt{c}(z - z_0)) \quad (1.13)$$

$$v(x, t) = \pm\sqrt{c}\text{sech}(\sqrt{c}(x - x_0 - ct)). \quad (1.14)$$

Para este trabajo de tesis se utiliza la solución exacta solo en su parte positiva.

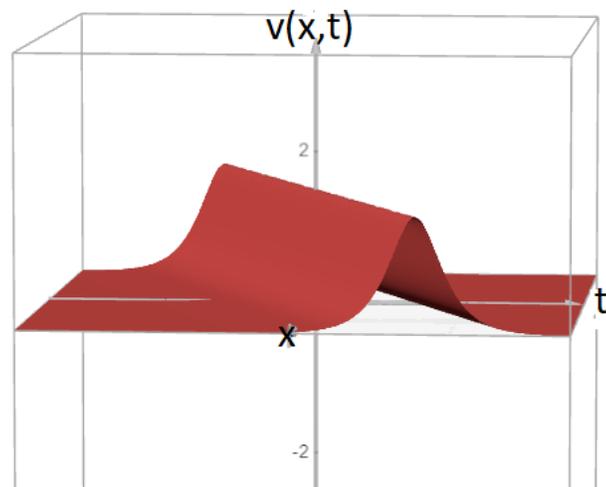


Figura 1.9: Solución graficada en tres dimensiones

1.4.9. Aprendizaje de las redes neuronales

Si se elige una red demasiado grande podrá memorizar todos los ejemplos formando una tabla de búsqueda grande la cual no necesariamente se generalizará bien las entradas que tienen que no se han visto antes, en otras palabras, como todos los modelos estadísticos, las redes neuronales están sujetas al sobreajuste cuando hay demasiados parámetros en el modelo. Si se atienden totalmente a todas las redes conectadas, las únicas opciones que se deben tomar son las que se refieren al número de capas ocultas y sus tamaños. El enfoque habitual es probar varios y quedarse con el mejor, se escoge la estructura de redes que da el más alto valor de predicción sobre los rangos de validación.

Si se considera que las redes no están completamente conectadas, entonces, se debe buscar algún método efectivo a través del gran espacio de posibles topologías de conexión. Después de que la red neuronal sea entrenada por primera vez, una teoría de la información identifica el enfoque de una selección óptima de conexiones que se pueden descartar. La red es reentrenada y si su eficiencia no decrementa entonces el proceso se repite. En suma, para remover conexiones y si es posible remover unidades que no contribuyen mucho al resultado.

1.4.10. Método PINN

La esencia de PINN es que su algoritmo es sin malla para aproximar soluciones a ecuaciones diferenciales parciales por transformar el problema, directamente resolviendo las ecuaciones que gobiernan en un problema de optimización de funciones de pérdida.

La metodología se basa estableciendo condiciones con condiciones de frontera, usualmente del tipo Dirichlet. Al resolver la parte real e imaginaria, la red neuronal profunda es usada para aproximar las soluciones y entonces, la ecuación de control es incorporada en la red. Se definen redes residuales, después con la teoría se usa para construir las leyes de conservación.

Justo en la figura 10 donde x y t representan las variables independientes de la solución $p(x, t)$, Ω_x y Ω_t son los dominios de estas dos variables independientes. El operador F representa la ecuación diferencial parcial y Φ es el parámetro dentro de ella. El operador P describe las condiciones de frontera. La forma en que se observan los datos se describe mediante el operador D . Entonces $p\theta$ es la función latente de la red neuronal profunda con el parámetro θ que se puede usar para aproximar la solución exacta $p(x,t)$ de la ecuación diferencial parcial (PDE).

El flujo de datos va escogiendo una red neuronal con ciertas capas, dichas capas tienen neuronas, la red pesa es inicializada con el algoritmo Xavier. La activación es la función tangente hiperbólica, la función de pérdida es optimizada por Adam y L-BFGS que son algoritmos. red neuronal profunda,

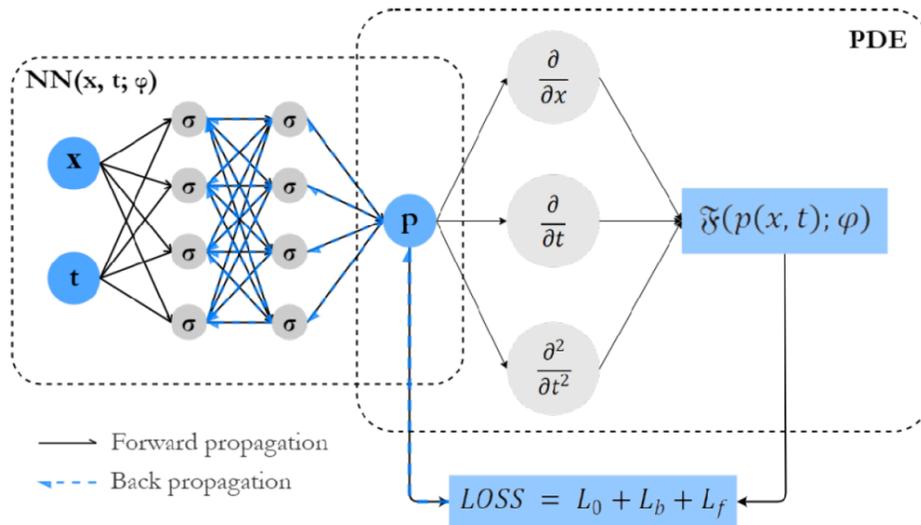


Figura 1.10: Se muestra el flujo del algoritmo PINN. La idea principal del algoritmo PINN es imponer restricciones de información física durante el entrenamiento de la red neuronal para acelerar el entrenamiento de la red neuronal y mejorar la precisión del modelo.

los parámetros de las funciones latentes $p\theta$ se pueden entrenar y actualizar constantemente. Al estudiar las soluciones basadas en datos de PDEs no lineales con las condiciones de frontera, los parámetros de las redes neuronales generalmente se aprenden minimizando la función de desviación estándar LOSS.

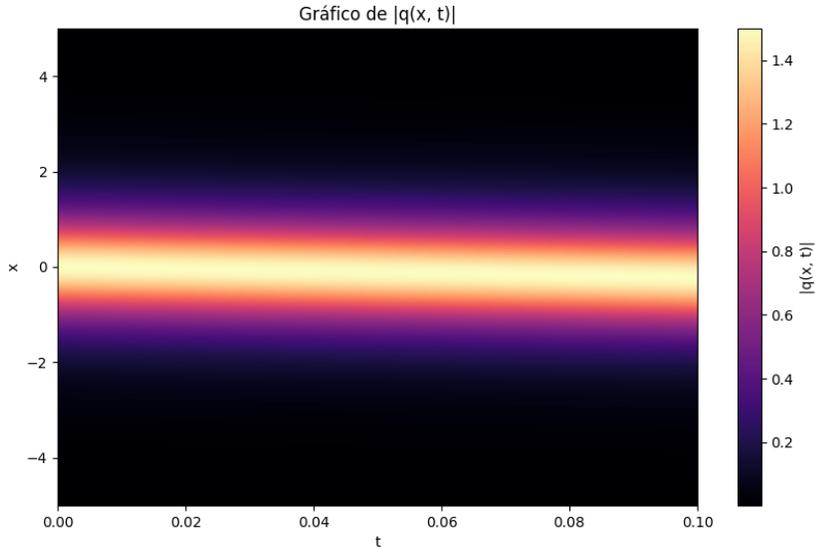


Figura 1.11: Solución solitón de aprendizaje graficada

1.4.11. PINN 'S

Primeramente, para que la PINN aprenda sobre la ecuación MKDV se tienen que establecer los datos previos, la forma en que esto se lleva a cabo es utilizando la solución (1.14) la cual será vital también para comprobar que la solución sea adecuada para el problema mencionado.

Para hacer los datos previos se hizo un programa en Python 3.8 en el que se grafica la ecuación (1.14) a través de definir el parámetro de x que va de $(-5,5)$ y de t que va de $(0,0.1)$, ambos dominios divididos en 200 puntos respectivamente. Se tuvo que contar la parte imaginaria de la solución ya que es una solución del tipo solitónica. Escogiendo una red neuronal con seis capas

```

def neural_net(self, X, weights, biases):
    num_layers = len(weights) + 1

    H = 2.0 * (X - self.lb) / (self.ub - self.lb) - 1.0
    for l in range(0, num_layers - 2):
        W = weights[l]
        b = biases[l]
        H = tf.tanh(tf.add(tf.matmul(H, W), b))
    W = weights[-1]
    b = biases[-1]
    Y = tf.add(tf.matmul(H, W), b)
    return Y

def net_u(self, x, t):
    X = tf.concat([x, t], 1)

    u = self.neural_net(X, self.weights, self.biases)
    u = u[:, 0:1]

    u_x = tf.gradients(u, x)[0]

    return u, u_x

def net_f(self, x, t):
    u, u_x = self.net_u(x, t)

    u_t = tf.gradients(u, t)[0]
    u_xx = tf.gradients(u_x, x)[0]
    u_xxx = tf.gradients(u_xx, x)[0]

    f = u_t + lambda_1*u**2*u_x + lambda_2*u_xxx

    return f

```

Figura 1.12: Funciones de peso y la ecuación a resolver

```

# Optimizers using TensorFlow 2.x
self.optimizer = dde.optimizers.tensorflow_compat_v1.scipy_optimizer.ScipyOptimizerInterface(
    self.loss, method='L-BFGS-B',
    options={'maxiter': 1000, 'maxfun': 1000, 'maxcor': 50, 'maxls': 50, 'ftol': 1.0 * np.finfo(float).eps})

#####
self.optimizer_Adam = tf.train.AdamOptimizer()
self.train_op_Adam = self.optimizer_Adam.minimize(self.loss)
self.gradients = tf.gradients(self.loss, self.weights)

# tf session
self.sess = tf.Session(config=tf.ConfigProto(allow_soft_placement=True,
                                             log_device_placement=True))

init = tf.global_variables_initializer()
self.sess.run(init)

self.losses = []

```

Figura 1.13: La estructura que llevó la optimización BFGS-B

ocultas, con un máximo de 100 neuronas y un mínimo de una, acomodadas de la siguiente forma [2,100,100,100,100,1]. Con 100 puntos iniciales (N_0), con 200 puntos de frontera (N_b) y 10,000 puntos de la región espacio-tiempo, al tener las condiciones solo queda resolver la parte real para la red neuronal profunda la cual se activa con una función tangente hiperbólica, la PINN por el error de la suma cuadrada de la ecuación de pérdida (1.15).

$$SSE = SSE_n + SSE_{n+1} \quad (1.15)$$

Luego con el optimizador L-BFGS que fue de 1000 iteraciones y el optimizador Adam de 400. Teniendo estos datos de entrenamiento a tiempo discreto la red neuronal logra aprender y con eso resolver la ecuación modificada KdV como se muestra en la imagen (1.15). Y el modelo aprendido comparado con el método exacto en la imagen (1.17)

```

# Domain bounds
lb = np.array([-5, 0])
ub = np.array([5, .1])

N0 = 100
N_b = 200
N_f = 10000
layers = [2, 100, 100, 100, 100, 1]

data = scipy.io.loadmat('C:\\Users\\thetr\\Downloads\\KdV_modificada82.mat')

t = data['t_values'].flatten()[:, None]
x = data['x_values'].flatten()[:, None]
Exact = data['q_values']

idx_x = np.random.choice(Exact.shape[0], N0, replace=False)
x0 = x[idx_x, :]
u0 = Exact[idx_x, 0:1]

X, T = np.meshgrid(x, t)

X_star = np.hstack((X.flatten()[:, None], T.flatten()[:, None]))
u_star = u0.T.flatten()[:, None]

idx_x = np.random.choice(x.shape[0], N0, replace=False)

idx_t = np.random.choice(t.shape[0], N_b, replace=False)
tb = t[idx_t, :]

X_f = lb + (ub - lb) * lhs(2, N_f)

model = PhysicsInformedNN(x0, u0, tb, X_f, layers, lb, ub)

```

Figura 1.14: Dominios que tiene, capas ocultas y condiciones

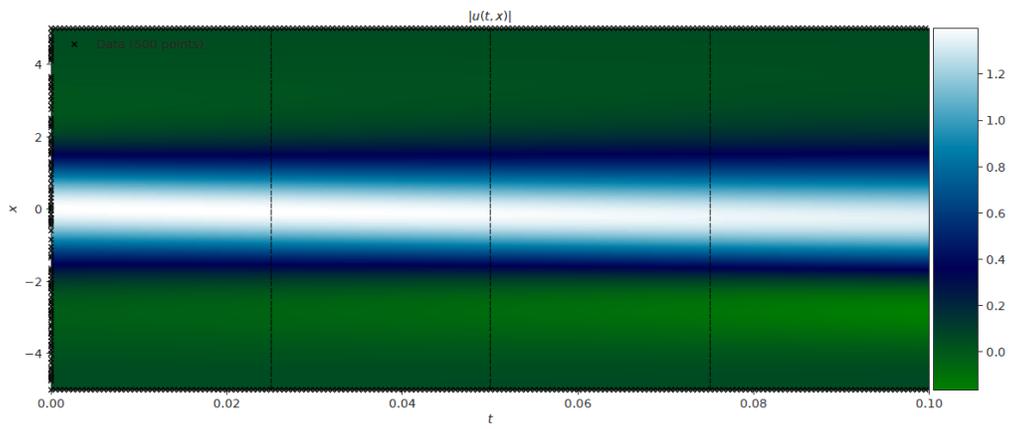


Figura 1.15: Solitón de los datos previos de aprendizaje hechos por la PINN

```

##### Row 1: h(t,x) slices #####
gs1 = gridspec.GridSpec(1, 3)
gs1.update(top=1 - 1 / 3, bottom=0, left=0.1, right=0.9, wspace=0.5)

ax = plt.subplot(gs1[0, 0])
ax.plot(x, Exact[:, 50], 'b-', linewidth=2, Label='Exact')
ax.plot(x, U_pred[50, :], 'r--', linewidth=2, Label='Prediction')
ax.set_xlabel('$x$')
ax.set_ylabel('$|u(t,x)|$')
ax.set_title('$t = %.2f$' % (t[50, 0]), fontsize=10)
ax.axis('square')
ax.set_xlim([-5.1, 5.1])
ax.set_ylim([-0.1, 1.1])

ax = plt.subplot(gs1[0, 1])
ax.plot(x, Exact[:, 100], 'b-', linewidth=2, Label='Exact')
ax.plot(x, U_pred[100, :], 'r--', linewidth=2, Label='Prediction')
ax.set_xlabel('$x$')
ax.set_ylabel('$|u(t,x)|$')
ax.axis('square')
ax.set_xlim([-5.1, 5.1])
ax.set_ylim([-0.1, 1.1])
ax.set_title('$t = %.2f$' % (t[100, 0]), fontsize=10)
ax.legend(loc='upper center', bbox_to_anchor=(0.5, -0.8), ncol=5, frameon=False)

ax = plt.subplot(gs1[0, 2])
ax.plot(x, Exact[:, 150], 'b-', linewidth=2, Label='Exact')
ax.plot(x, U_pred[150, :], 'r--', linewidth=2, Label='Prediction')
ax.set_xlabel('$x$')
ax.set_ylabel('$|h(t,x)|$')
ax.axis('square')

ax.set_xlim([-5.1, 5.1])
ax.set_ylim([-0.1, 1.1])
ax.set_title('$t = %.2f$' % (t[150, 0]), fontsize=10)

```

Figura 1.16: Forma en que se denota para las gráficas finales

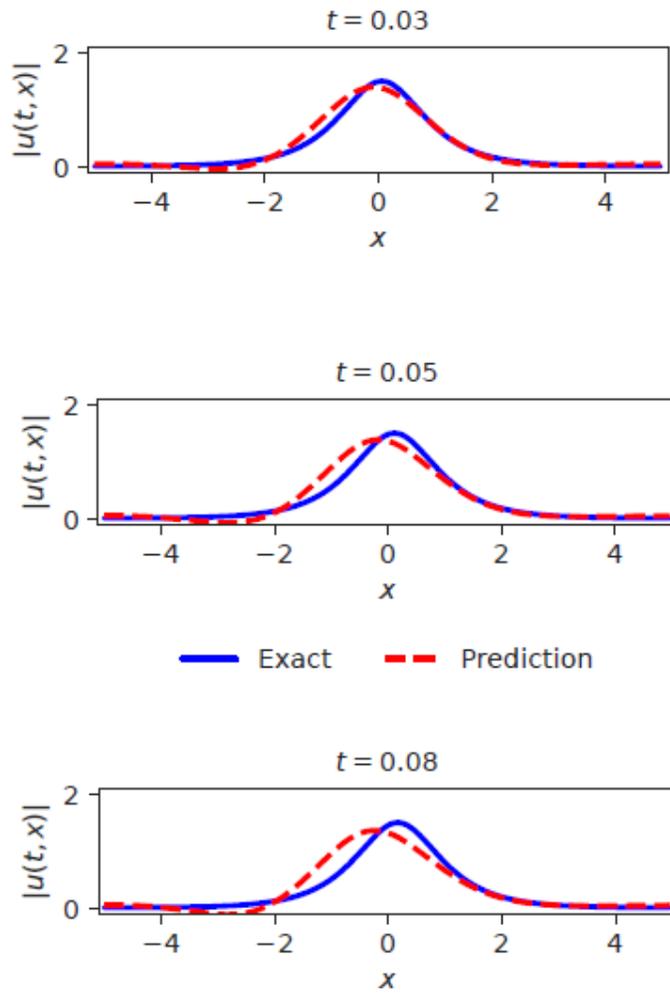


Figura 1.17: Gráficas donde se compara en tres tiempos diferentes mostrando así el aprendizaje de la PINN $t=0.03$, $t=0.05$ y $t=0.08$

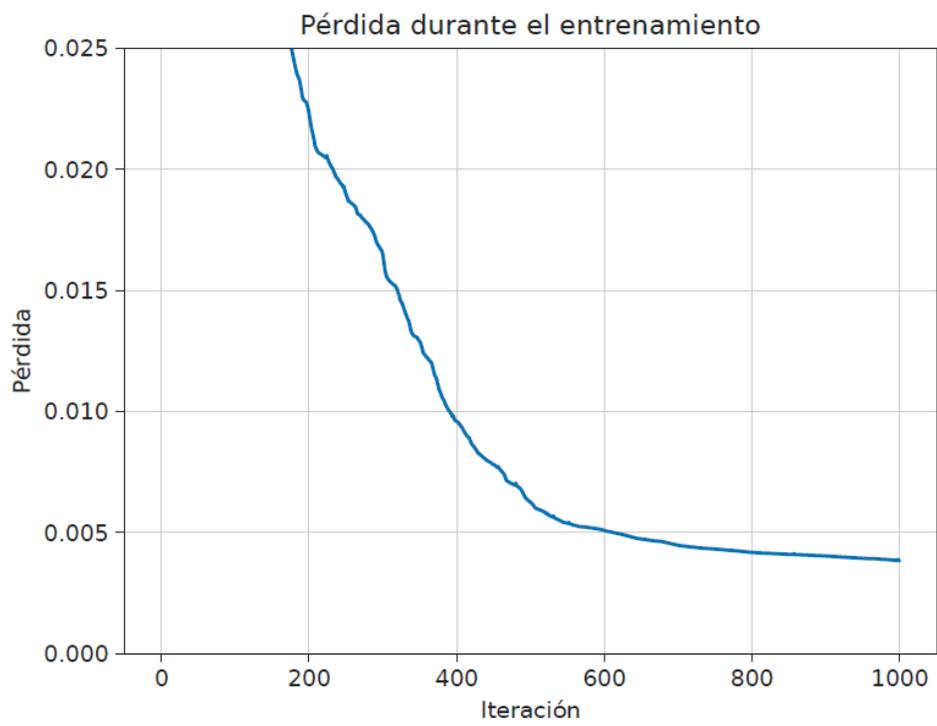


Figura 1.18: Gráfica de pérdidas en el entrenamiento por iteración

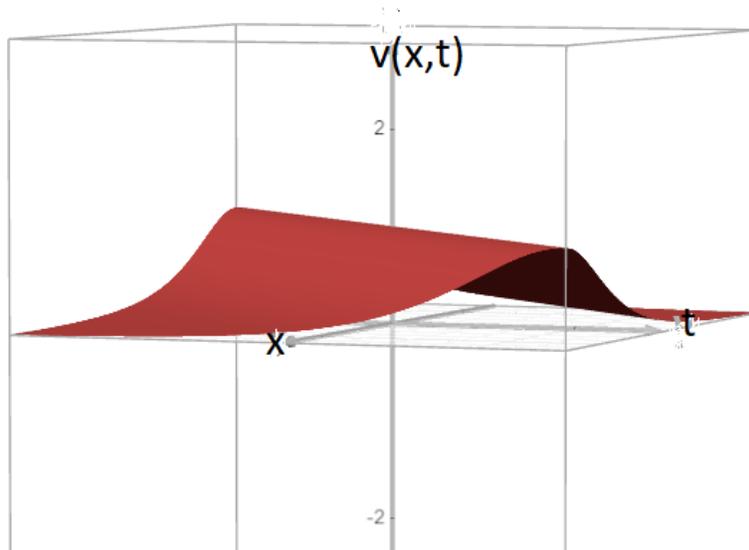


Figura 1.19: Solución graficada en tres dimensiones de sech

1.5. Ejemplo número 2

En esta sección se agrega otro resultado del tipo solución solitón, siendo este con $c=1$ y $x_0 = 0$, ya que al ser una ecuación diferencial parcial no lineal posee mas soluciones, donde se va a plantear de la siguiente manera. La solución exacta solo en su parte positiva al igual que en caso anterior.

$$v(x, t) = \operatorname{sech}(x - t) \quad (1.16)$$

1.6. PINN del ejemplo número 2

Para hacer los datos previos se hizo un programa en Python 3.8 en el que se grafica la ecuación (1.16) a través de definir el parametro de x que va de $(-5,5)$ y de t que va de $(0,0.1)$, ambos dominios divididos en 200 puntos respectivamente. Se tuvo que contar la parte imaginaria de la solución ya que es una solución del tipo solitónica viniendo de la ecuación en este caso de (1.16). Escogiendo una red neuronal con seis capas ocultas, con un máximo de 100 neuronas y un mínimo de una, acomodadas de la siguiente forma $[2,100,100,100,100,1]$. Con 100 puntos iniciales (N_0), con 200 puntos de frontera (N_b) y 10,000 puntos de la región espacio-tiempo, al tener las condiciones solo queda resolver la parte real para la red neuronal profunda la cual se activa con una función tangente hiperbólica.

Después con el optimizador L-BFGS que fue de 1000 iteraciones y el optimizador Adam de 400. Teniendo estos datos de entrenamiento a tiempo discreto la red neuronal logra aprender y con eso resolver la ecuación modificada KdV de manera igual que en el caso anterior. Se puede apreciar que la red neuronal tiene una funcionalidad muy viable para resolver ecuaciones diferenciales parciales no lineales a partir de soluciones que no están meramente adaptadas a su funcionalidad, tal como se ve en las gráficas de la figura (1.23), tiene un aprendizaje más que aceptable.

```

def q(x, t):
    return (np.cosh((x-t)/(np.sqrt(1))))**2
#era con 3

# Definir valores de x y t
x_values = np.linspace(-5, 5, 1000)
t_values = np.linspace(-0, .1, 1000)

# Calcular la función q(x, t)
Q = q(x_values[:, np.newaxis], t_values[np.newaxis, :])

# Graficar la función
plt.figure(figsize=(10, 6))
plt.imshow(Q, extent=[-0, .1, -5, 5], aspect='auto', cmap='magma')
plt.colorbar(Label='|q(x, t)|')
plt.xlabel('t')
plt.ylabel('x')
plt.title('Gráfico de |q(x, t)|')
plt.show()

# Guardar datos equidistantes en un archivo .mat
# Seleccionar puntos equidistantes de la función
x_indices = np.arange(0, len(x_values), 5)
t_indices = np.arange(0, len(t_values), 5)

# Crear matriz de datos para guardar
data = {
    'x_values': x_values[x_indices],
    't_values': t_values[t_indices],
    'q_values': Q[x_indices][:, t_indices]
}

```

Figura 1.20: Código de datos previos de aprendizaje usado en ambos casos adaptado respectivamente

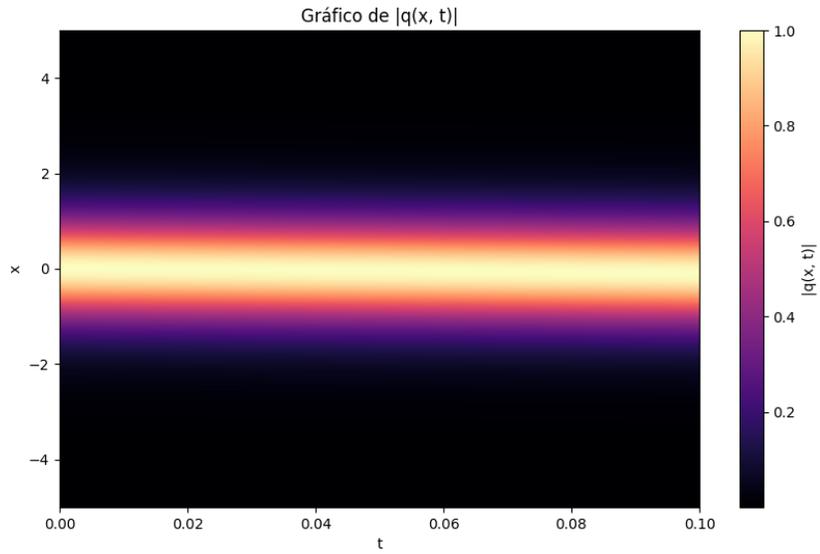


Figura 1.21: Solución solitón de aprendizaje de la ecuación (1.38) graficada

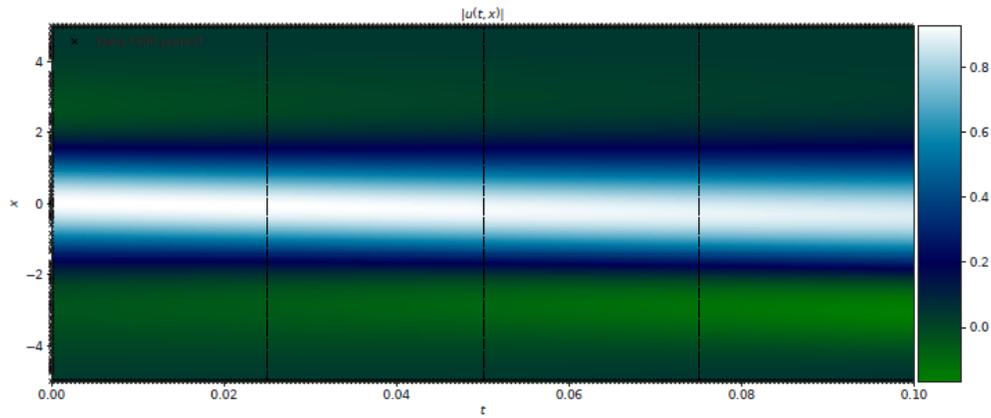


Figura 1.22: Soliton $\text{sech}(x-t)$ de los datos previos de aprendizaje hechos por la PINN

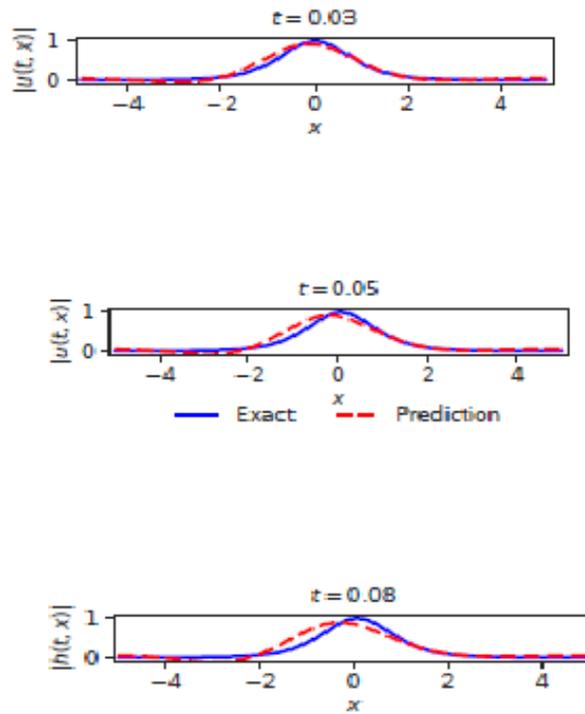


Figura 1.23: Gráficas donde se compara en tres tiempos diferentes mostrando así el aprendizaje de la PINN $t=0.03$, $t=0.05$ y $t=0.08$

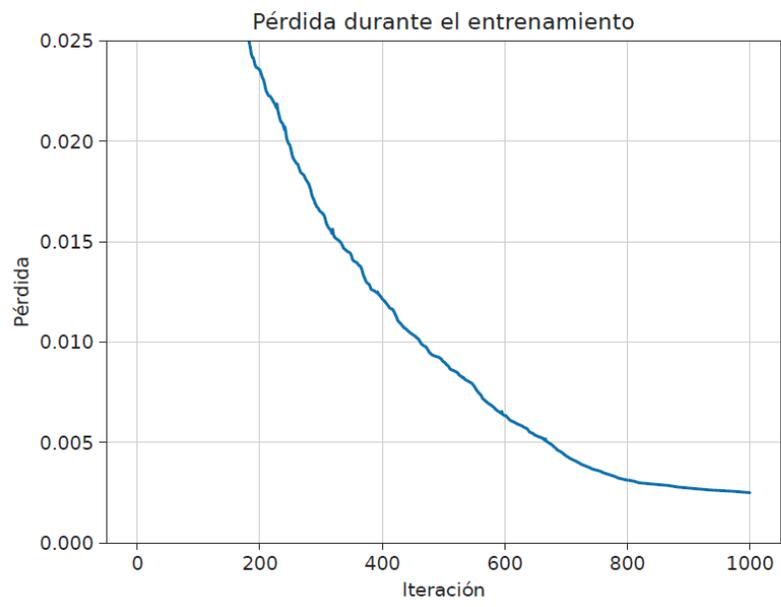


Figura 1.24: Gráfica de pérdidas en el entrenamiento por iteración

1.7. Conclusiones

El primero punto a mencionar fue al encontrar una solución analítica para la ecuación modificada KdV que se logró sin necesidad de implementar condiciones iniciales y de frontera, en el primer ejemplo, se usó la siguiente ecuación,

$$v(x, t) = 1,5 \operatorname{sech}[1,5x - 3,375t].$$

Donde c es igual a 1.5, $x_0 = 0$ y lo cual fue viable por la naturaleza de que la ecuación que si es integrable y al analizar su comportamiento se confirma que es del tipo solitón su solución, lo cual queda al graficarse se comprueba que tiene un comportamiento no lineal. Esto igualmente ocurre en el ejemplo número dos a pesar de que en ese caso $c=1$ lo cual es igual en sus valores de $x_0 = 0$, por eso tienen un comportamiento semejante.

En ambos ejemplos se utilizó el arreglo de la red neuronal de seis capas ocultas, con un máximo de 100 neuronas y un mínimo de una, acomodadas de la siguiente forma [2,100,100,100,100,1]. Con 100 puntos iniciales (N_0), con 200 puntos de frontera (N_b) y 10,000 puntos de la región espacio-tiempo. Con la única diferencia del valor c como se había mencionado.

Se aprecia que en la PINN se logra un aprendizaje perfectamente efectivo a pesar de que sean pocas iteraciones y poca pérdida por iteración en dicho proceso como se aprecia en la imagen (1.18) y (1.24). Teniéndose este trabajo se concluye que en un futuro la inteligencia artificial por redes neuronales que son de las primeras que hubo en dicho campo, es que la forma de tra-

bajar computacionalmente la física ayuda en bajar los costos energéticos y de cómputo al no tener la necesidad de usar super computadoras para conseguir un resultado semejante, incluso se podría decir que también servirá para resolver diferentes tipos de ecuaciones diferenciales parciales no lineales de manera más efectiva que con otro tipo de metodologías.

Al ser tan funcional una red neuronal es que como se vio anteriormente es lo amplia que puede ser para aprender distintas soluciones a pesar de que no sean adaptadas para la red, es decir que son soluciones más puras, logra aprender sin ningún inconveniente y no se limita a una sola solución.

Es importante mencionar que el campo de investigación sobre las PINN es algo reciente, por lo cual mejorando también los procesos de cómputo seguramente en un futuro las redes neuronales sean incluso más efectivas a largo plazo por las diferentes actualizaciones que se van haciendo sobre todo para la librería tensor Flow usada en Python, siendo usadas porque son de mayor accesibilidad al ser de uso libre, también, se utilizó una computadora común, incluso de bajos recursos para lograr el desarrollo de las redes neuronales al igual que los calculos anteriormente vistos, ya que los primeros trabajos que se hicieron fueron con su primera versión, actualmente se trabaja con la segunda versión el cual fue este caso.

1.8. Referencias

- 1 P.H. Winston, Artificial Intelligence Addison-Wesley, Massachusetts, USA, 1992.
- 2 Artificial Intelligence. A Modern Approach. Third Edition, S. Russell, P. Norvig, Editors, Pearson Education, New Jersey, 2010
- 3 <https://www.oracle.com/mx/artificial-intelligence/what-is-ai/>
- 4 C. M. Bishop, Pattern Recognition and Machine Learning, Springer, 2006.
- 5 Y. LeCun, Y. Bengio, G. Hinton, Deep learning, Nature 521, 436 (2015).
- 6 M. Raissi, Hidden physics models: Machine learning of nonlinear partial differential equations.
- 7 I. E. Lagaris, A. Likas, and D. I. Fotiadis, Artificial neural networks for solving ordinary and partial differential equations, IEEE Trans. Neural Networks 9, 987-1000 (1998).
- 8 M. Raissi, G. E. Karniadakis, Hidden physics models: machine learning of nonlinear partial differential equations, J. Comput. Phys. 357, 125-141 (2018)
- 9 M. Raissi, P. Perdikaris, G. E. Karniadakis. Physics-informed neural networks: A deep learning framework for solving forward and inverse

- problems involving nonlinear partial differential equations, *J. Comput. Phys.* 378 (2019) 686-707.
- 10 L. Lu, X. Meng, Z. Mao, and G. E. Karniadakis (2021), Deep xde: A deep learning library for solving differential equations, *SIAM Review*, 63(1), 208228, doi: 10.1137/19m1274067
 - 11 Z. Zhou and Z. Yan, Solving forward and inverse problems of the logarithmic nonlinear Schrödinger equation with PT-symmetric harmonic potential via deep learning, *Phys. Lett. A* 387 (2021) 127010.
 - 12 L. Wang and Z. Yan, Data-driven rogue waves and parameter discovery in the defocusing NLS equation with a potential using the PINN deep learning, arXiv: 2012.09984 (2020).
 - 13 L. McClenny and U. Braga-Neto (2020), Self-adaptive physics-informed neural networks using a soft attention mechanism. arXiv:2009.04544
 - 14 G. Marcucci, D. Pierangeli, C. Conti, Theory of neuromorphic computing by waves: machine learning by rogue waves, dispersive shocks, and solitons, *Phys. Rev. Lett.* 125, 093901 (2020).
 - 15 Z. Zhou, Zh. Yan (2021) Deep learning neural networks for the third-order nonlinear Schrödinger equation: Solitons, breathers, and rogue waves, *Commun. Theor. Phys.* 73 105006.

- 16 M. Nakajima, K. Tanaka, T. Hashimoto (2022) Neural Schrödinger Equation: Physical Law as Neural Network, arXiv:2006.13541
- 17 W. Ames, Nonlinear Partial Differential Equations, Academic Press, New York, 1967.
- 18 T.I. Karpman, H. Ono, J. Phys. Soc. Jpn. 34 (1973) 1073.
- 19 M.J. Ablowitz, H. Segur, Solitons and the Inverse Scattering Transform, SIAM, Philadelphia, 1981.
- 20 S. Watanabe, Ion acoustic soliton in plasma with negative ion, J. Phys. Soc. Jpn. 53 (1984) 950–956.
- 21 R. Hirota, Exact solution of the modified Korteweg-de Vries equation for the multiple collisions of solitons J. Phys. Soc. Jpn. 33 (1972) 1456–1458.
- 22 M. Wadati, The modified Korteweg-de Vries equation, J. Phys. Soc. Jpn. 34 (1973) 1289–1296.
- 23 S. Tanaka, Modified Korteweg-de Vries equation and scattering theory, Proc. Jap. Acad.48 (1972) 466–469.
- 24 T. Gesztesy, W. Schweiger and B. Simon, Commutation methods applied to the mKdV-equation, Trans. Amer. Math. Soc. 324 (1991) 465–525.

- 25 M. Wadati, The exact solution of the modified Korteweg–de Vries equation, *J. Phys. Soc. Jpn.* 32 (1972) 1681.
- 26 R. Miura, Korteweg–de Vries equation and generalizations, I: a remarkable explicit nonlinear transformation, *J. Math. Phys.* 9 (1968) 1202–1204.
- 27 <https://github.com/FelixEnriqueSH/PINNs>
- 28 Agüero Granados, M.A., Serkin V. N.(2020). *Introducción a la teoría de solitones*.(1°ed.). Ciudad de México.Ediciones y graficos Eón
- 29 Stuart J. Russell and Peter Norvig *Artificial Intelligence A Modern Approach* (2°ed.). Upper Saddle River, New Jersey. Pearson Education, Inc.
- 30 J. David Logan *An Introduction to Nonlinear Partial Differential Equations* (2° ed.). Hoboken, New Jersey. John Wiley and Sons, Inc.