

BENEMÉRITA UNIVERSIDAD AUTÓNOMA
DE PUEBLA

FACULTAD DE CIENCIAS FÍSICO
MATEMÁTICAS



USO DE REDES NEURONALES
PARA LA RESOLUCIÓN DEL
PROBLEMA INVERSO EN EL
SISTEMA TOGGLE SWITCH DE
REDES DE REGULACIÓN GÉNICA

Tesis presentada para obtener el grado de Licenciado en
Física

Presenta: Marco Antonio Salazar Bengochea

Asesor: Dr. Jorge Velázquez Castro

Fecha: 01/25

FACULTAD DE CIENCIAS FISICO MATEMATICAS

Título: Uso de redes neuronales para la resolución del problema inverso en el sistema toggle switch de redes de regulación génica

Declaración de autoría: Declaro que los resultados que aparecen en este informe son producto de la investigación realizada por este autor, por lo que, tanto como mis tutores y yo nos hacemos responsables de ello. De igual forma autorizo a la Benemérita Universidad Autónoma de Puebla a hacer uso del contenido de esta tesis con la finalidad que estime conveniente.



Estudiante: Marco Antonio Salazar Bengochea
Matricula: 201315728
Carrera: Lic. Física



Dr. Javier Miguel Hernández L.
Presidente

COMITÉ



Dr. Enrique Varela Carlos
Secretario



Dr. Lucio Fide Repolledo H.
Vocal



Dr. Jorge Velázquez Castro
Asesor

Epigrafe

- *No vuelvas, no pienses en nosotros, no llames, no escribas, no te dejes engañar por la nostalgia. ¡Olvídanos! Si regresas, no quiero que vengas a verme, no te dejare entrar a mi casa, ¿Entendido?*
- *Gracias, por todo lo que has hecho por mí.*
- *Hagas lo que hagas, amalo, como amabas la cabina del Paradiso cuando eras niño.*

- Alfredo despide a Toto, Cinema Paradiso, 1988.

Dedicatoria

Dedico este trabajo de investigación que tanto me ha costado a las personas que siempre estuvieron conmigo apoyándome a su manera, mi hermanita Andrea Salazar Bengochea, mi jefa Martha Patricia Bengochea López y no menos importante por ponerlo al final a mi señor padre Marco Antonio Salazar de la Rosa, nunca me dejaron solo en este muy largo camino, entiendo que no les parecieran algunas de mis decisiones pero ahora entiendo que solo se preocupaban por mí y por eso se los agradezco con todo mi corazón los quiero mucho.

También me gustaría dedicarle esta tesis a mi familia, tanto materna como paterna, los quiero muchísimo a todos, disculpen mi ausencia.

Como mención especial, quisiera dedicarle esta Tesis a ese pequeño niño de 3 años que siempre quiso estudiar física. Aunque nuestro camino no nos permitió continuar con algunos de nuestros sueños, me gustaría decirle que lo estamos haciendo fenomenal y que lo estamos logrando a nuestro ritmo, aun hay un camino muy grande por delante y que no olvide que nunca hay que dejar de soñar.

Agradecimiento

Adicionalmente me gustaría agradecer a todas las personas que han pasado por mi vida y me han aportado muchísimas cosas que me han hecho ser la persona que soy en este momento. De esta manera me gustaría aclarar que seguramente hay muchísimas personas que pase por alto en este momento pero les aseguro que los tengo presente, muchísimas gracias por todo.

Le agradezco muchísimo a mi asesor Jorge, que me permitió la oportunidad de trabajar con el, dadas las circunstancias, no pude trabajar de una manera tan constante, pero por su comprensión y paciencia estoy infinitamente agradecido, espero poder seguir trabajando con usted.

Les agradezco infinitamente a mis hermanos de otras madres, Itzel muchísimas gracias por todo, ambos sabemos que si no me hubieras ayudado a salir del hoyo horrible en el que estaba no hubiera podido terminar esto, junto a esto te agradezco muchísimo Esme por ayudarme a entenderme mejor y a salir adelante. Alejandro te agradezco muchísimo por estar en las buenas y las malas, me has

ayudado muchísimo no solo en esto si no en muchísimas cosas más, espero algún día poder pagarte todo. Mis hermanos Jocelyn, Milton, José Luis y Bela gracias por aguantarme tanto tiempo, gracias por estar ahí cuando más los necesito no sé dónde estaría sin ustedes. Abraham te quiero muchísimo hermanito gracias por siempre preocuparte por mí y hacerme reír tanto. Ángel Daniel, gracias por escuchar mis penas siempre y por preocuparte siempre por mi familia, Fernando muchas gracias por ser mi amigo y aunque no lo creas (broma) tu me has explicado muchísimas mas cosas de las que yo podría haberte explicado a ti, Yoli muchas gracias por todo, disculpa la ausencia pero quiero que sepas que también estas presenten siempre. Los pinkys Rene, Abelardo, Isaac, Renel y Eldon, muchísimas gracias por hacer mas amena la universidad, sin ustedes me hubiera salido desde el primer día. A mi entrenador y mi amigo Jorge, muchísimas gracias por todo el apoyo, la constante presión para mejorar y las madrizas. Adrián muchísimas gracias por ser mi mejor amigo, me quede con él bebe del divorcio y no me arrepiento de nada, gracias por escucharme siempre. A la Famigochi por las aventuras y las lecciones de vida.

Como ya mencioné les agradezco infinitamente a todos las personas que han influido en mi persona, Edson, Max, Mauro, Chuy, Omar, Tacho, Mixtli, Apuch, Diana, Yessi, Lupita, Sando, Alhely, Hector, Bruno, Jair, Anthony, Memo, David, Cyn, Cynthia, Angel, Vicho, Pame, Bardo, Dominuku, Sergio, Armando, Joey, Carlo, Saneshy, Cesar, Cilene, Mario, Karen, Neri, Fabi, Mariana, Chuchin, Dani, Bere, Kike, Sergio, Itsayana, Alma, Karina, Santos, Dilan, Angel (hermane), Arnold (hermane), Gera, Rafa, a mis profesores de universidad, y todos los demás una disculpa si olvide mencionarte, pero muchas gracias.

Resumen

Palabras Clave: Redes neuronales, Toggle switch, Puntos estacionarios, Redes de regulación génica, Biología de sistemas.

Este proyecto de investigación se centra en resolver el problema inverso de regulación genética en particular el sistema toggle switch. Este problema consiste en identificar los parametros del sistema dinámico a través de los puntos estacionarios, que son estados de equilibrio del sistema, donde las tasas de cambio de las variables del modelo son nulas. Estos puntos son cruciales porque representan situaciones donde las variables biológicas alcanzan un estado de estabilidad, lo que a menudo refleja condiciones naturales o experimentales de interés.

Cuando diseñamos modelos matemáticos para representar sistemas biológicos, como redes de regulación génica o vías metabólicas, es crucial que estos modelos capturen adecuadamente el comportamiento del sistema en puntos estacionarios. Esto se debe a que, en la práctica experimental, es común medir o observar los puntos estacionarios del sistema, donde las concentraciones de las moléculas involucradas alcanzan valores constantes en el tiempo. Por lo tanto, si somos capaces de identificar los parámetros del modelo a partir de los puntos estacionarios observados en experimentos, podemos inferir la estructura y la dinámica del sistema biológico subyacente. Esto nos permite entender mejor cómo funcionan los procesos biológicos a nivel molecular y celular, y cómo responden a diferentes condiciones ambientales o perturbaciones externas.

El problema inverso no es trivial ya que una observación puede surgir de varios conjuntos de parámetros. Esto significa que diferentes combinaciones de parámetros pueden llevar a los mismos puntos estacionarios, lo que complica la tarea de determinar cuáles son los valores correctos. Además, en muchos casos no sabemos cuántos puntos estacionarios existen para un sistema dado, lo que añade otra capa de complejidad al problema. Abordar esta incertidumbre requiere métodos que puedan considerar múltiples soluciones posibles en base a datos experimentales y conocimientos previos del sistema.

El objetivo particular de este proyecto es usar redes neuronales para encontrar la solución al problema inverso, aprovechando las ventajas que estas ofrecen sobre los métodos clásicos. Las redes neuronales son adecuadas para sistemas complejos y no lineales debido a su capacidad para modelar relaciones compli-

cidas y capturar patrones intrincados en los datos. Aunque el entrenamiento de redes neuronales requiere una alta carga computacional, su capacidad de generalización es muy buena, permitiendo hacer predicciones precisas y robustas incluso cuando los datos de entrada son ruidosos o incompletos. Estas características hacen de las redes neuronales una herramienta poderosa para abordar el problema inverso en la regulación genética.

En particular, en el caso de mediciones de concentraciones de proteínas en equilibrio, la capacidad de identificar los parámetros del modelo a partir de los puntos estacionarios nos permite obtener información crucial sobre la cinética y la regulación de la expresión génica. Esto puede tener importantes implicaciones en áreas como la biología del desarrollo, la medicina y la ingeniería de tejidos, donde comprender cómo se regulan las proteínas en el tiempo y el espacio es esencial para entender fenómenos biológicos complejos y desarrollar intervenciones terapéuticas efectivas.

Abstract

Keywords: Neural networks, Toggle switch, Steady states, Gene regulatory networks, Systems biology.

This research project focuses on solving the inverse problem of genetic regulation, particularly in the context of the toggle switch system. This problem involves identifying the parameters of the dynamic system through the steady states, which are equilibrium points where the rates of change of the model variables are zero. Steady states are crucial because they represent situations where biological variables reach a stable state, often reflecting natural or experimental conditions of interest.

When designing mathematical models to represent biological systems, such as gene regulatory networks or metabolic pathways, it is essential that these models adequately capture the system's behavior at steady states. This is because, in experimental practice, it is common to measure or observe the steady states of the system, where the concentrations of the involved molecules reach constant values over time. Identifying the model parameters from the steady states observed in experiments allows us to infer the underlying structure and dynamics of the biological system. This, in turn, helps us better understand how biological processes function at the molecular and cellular levels and how they respond to different environmental conditions or external perturbations.

The inverse problem is non-trivial since an observation can arise from various sets of parameters. This means that different combinations of parameters can lead to the same steady states, complicating the task of determining the correct values. Additionally, in many cases, we do not know how many steady states exist for a given system, adding another layer of complexity to the problem. Addressing this uncertainty requires robust and sophisticated methods of analysis

and modeling that can consider multiple possible solutions and evaluate their plausibility based on experimental data and prior knowledge of the system.

A particular objective of this project is to use neural networks to find the solution to the inverse problem, leveraging the advantages they offer over classical methods. Neural networks are well-suited for complex and nonlinear systems due to their ability to model intricate relationships and capture complex patterns in the data. Although training neural networks requires high computational power, their generalization capability is excellent, allowing for accurate and robust predictions even when the input data is noisy or incomplete. These characteristics make neural networks a powerful tool for addressing the inverse problem in genetic regulation.

In particular, when it comes to measurements of protein concentrations at equilibrium, the ability to identify model parameters from steady states provides crucial information about the kinetics and regulation of gene expression. This has important implications in fields such as developmental biology, medicine, and tissue engineering, where understanding how proteins are regulated over time and space is essential for comprehending complex biological phenomena and developing effective therapeutic interventions.

Introducción

Antecedentes históricos

Introducción a la biología de sistemas, las redes de regulación génica, los sistemas dinámicos y la aportación de las redes neuronales a estas áreas.

La biología de sistemas es un campo interdisciplinario que estudia los sistemas biológicos mediante las interacciones entre sus componentes, enfocándose en la estructura y dinámica de las funciones celulares y de organismos completos. Aborda temas como la robustez de los sistemas biológicos y la estructura de las redes de regulación génica, siendo crucial para procesos como la diferenciación celular y la respuesta a estímulos.

Los sistemas dinámicos y las redes son campos estrechamente relacionados que estudian el comportamiento de sistemas interconectados a lo largo del tiempo. La relación entre sistemas dinámicos y redes puede entenderse de las siguientes maneras:

- **Sistemas Dinámicos en Redes:** En este contexto, la teoría de sistemas dinámicos se aplica para estudiar cómo los procesos evolucionan en redes complejas. La dinámica de los nodos individuales y las interacciones entre ellos son cruciales para modelar el comportamiento global del sistema. Ejemplos de esto incluyen el estudio de fenómenos de sincronización, la propagación de epidemias, la dinámica de opiniones y el flujo de información en redes.
- **Topología de la Red y Dinámica:** La estructura de una red, incluyendo su topología y patrones de conectividad, juega un papel significativo en la determinación de la dinámica de los procesos que ocurren en la red. Diferentes estructuras de red pueden llevar a comportamientos dinámicos distintos, como la emergencia de la sincronización en redes con patrones de conectividad específicos.
- **Redes Adaptativas:** Las redes adaptativas se refieren a sistemas donde

tanto la dinámica de la estructura de la red como la dinámica sobre la red se consideran simultáneamente. Esta interacción entre la evolución de la topología de la red y la dinámica de los procesos en la red puede llevar a comportamientos ricos y complejos.

- **Redes Dependientes del Tiempo:** Los sistemas dinámicos en redes dependientes del tiempo o temporales han ganado atención, donde la estructura de la red evoluciona con el tiempo. Comprender cómo la dinámica en estas redes en evolución se influyen mutuamente es un área clave de investigación.

Al estudiar las interacciones entre sistemas dinámicos y redes, los investigadores pueden obtener conocimientos sobre el comportamiento colectivo de sistemas interconectados, descubrir fenómenos emergentes y abordar desafíos del mundo real en diversos dominios. La integración de la teoría de sistemas dinámicos con la ciencia de redes ha llevado a una comprensión más profunda de los sistemas complejos y su dinámica. (Hirsch, M. W., Smale, S., & Devaney, R. L. 2013)

Las aplicaciones prácticas de la biología de sistemas van más allá del ámbito médico o teórico e incluyen campos como la epidemiología y la sincronización espontánea entre sistemas biológicos, con importantes implicaciones sociales desde una perspectiva biomatemática. El uso de técnicas computacionales, como las redes neuronales artificiales, ha mejorado significativamente el estudio de las redes de regulación génica, destacando la naturaleza interdisciplinaria y la importancia de este campo para el avance de la biología en el siglo XXI. Las redes neuronales, especialmente las redes neuronales recurrentes (RNN) y las redes neuronales de largo corto plazo (LSTM), son particularmente adecuadas para modelar y predecir sistemas dinámicos complejos debido a su capacidad para aprender patrones temporales y secuenciales en los datos. Estas técnicas pueden manejar grandes volúmenes de datos y capturar relaciones no lineales sin la necesidad de especificar un modelo exacto, lo que es crucial para entender la complejidad inherente a las redes de regulación génica. Sin embargo, el uso de redes neuronales también presenta desafíos, como la necesidad de grandes cantidades de datos para el entrenamiento, la alta carga computacional y la dificultad de interpretar los modelos resultantes. A pesar de estas limitaciones, las redes neuronales ofrecen una capacidad de generalización superior, permitiendo avances significativos en la biología de sistemas y la medicina personalizada, al mejorar nuestra comprensión y predicción de la dinámica génica. (Rainer, 2010; Kitano, H., 2022; Junqueira, 2012; Peña Jiménez, 2016; Kremling, 2014; Porter & Gleeson, 2016)

Contexto histórico y trabajos previos.

El desarrollo de la biología de sistemas y el estudio de las redes de regulación génica han experimentado avances significativos en las últimas décadas, siendo fundamentales para comprender la regulación génica y su implicación en el cán-

cer. Desde los descubrimientos pioneros de Watson y Crick sobre la estructura del ADN, hasta los hallazgos de Jacob y Monod sobre los operones en bacterias, y el trabajo de Davidson en la regulación génica durante el desarrollo embrionario, se ha progresado notablemente en este campo (Science History Institute Museum & Library, 2022; Jacob et al., 2005; ROY J. BRITTEN, 1969).

La biología molecular ha evolucionado hacia la biología de sistemas, que integra biología, matemáticas e informática para estudiar los sistemas biológicos de manera integral, centrándose en las interacciones entre sus componentes (Rainer, 2010). Dentro de este marco, los sistemas dinámicos han desempeñado un papel crucial en la modelización de sistemas biológicos complejos. Estos sistemas permiten capturar cómo las interacciones entre moléculas, células y organismos completos conducen a comportamientos observables a diferentes escalas de tiempo y espacio (Porter & Gleeson, 2016).

El concepto de "programa genético", ha sido fundamental para explicar la ontogenia de los organismos, mientras que la genética evolutiva del desarrollo ha abordado procesos genético-moleculares relacionados con transformaciones macroevolutivas (Gerald M. Rubin, 2000). La Evo-Devo ha desafiado las perspectivas convencionales sobre la regulación genética y el desarrollo, proporcionando nuevos entendimientos teóricos y empíricos sobre la complejidad biológica.

La interacción entre la biología de sistemas, las redes de regulación génica y Evo-Devo resalta la naturaleza interdisciplinaria de estos campos, fundamentales para el avance de la biología en el siglo XXI. (Love, 2015) Los desarrollos tecnológicos actuales, como la secuenciación genómica masiva y la medición del ARN mensajero, han proporcionado herramientas poderosas para abordar la complejidad de la regulación génica y tienen aplicaciones que van más allá de lo médico o teórico (Picelli, 2017).

Las técnicas computacionales han contribuido significativamente a la comprensión de la regulación génica y otros aspectos biológicos clave, permitiendo modelar las interacciones entre los componentes de las redes de regulación génica y analizar grandes conjuntos de datos experimentales sobre expresión génica (Abraham, 2002).

Estos avances han resultado en una mejor comprensión de la complejidad biológica y han abierto nuevas perspectivas para futuras investigaciones y descubrimientos. Diferentes metodologías informáticas han sido exitosas en este sentido, contribuyendo a despejar el camino hacia un mayor entendimiento de la biología (P, D, & H, 2019; V, N, & G. -J., 2015; T. H. H. Aldhyani, 2020).

Índice general

1. Contexto Teórico	1
1.1. Planteamiento del problema y objetivo de la investigación	1
1.2. Marco Teórico.....	2
1.2.1. Biología de Sistemas.....	2
1.2.2. Redes de regulación génica.....	4
1.2.3. Sistemas Dinámicos.....	6
1.2.4. Sistema Biestable.....	10
1.2.5. Toggle Switch.....	14
1.2.6. Problemas inversos	16
1.2.7. Redes neuronales y su aplicación.....	19
2. Metodología	25
2.1. Generación de datos	27
2.2. Solución al sistema de ecuaciones	29
2.3. Datos de entrenamiento, validación y prueba.....	30
2.4. Problema Directo	31
2.4.1. Definir y compilar el modelo	31
2.5. Problema inverso	33
2.5.1. Definir y compilar el modelo	33
3. Análisis de resultados	35
3.1. Generación de datos	35
3.2. Problema directo.....	36
3.3. Problema inverso	40

xvi	3.4. Conclusiones.....	42
-----	------------------------	----

Índice de figuras

1.1. Tipos de Atractores (Cáceres & O’Farrill, Chaos Culture, 2021) .	9
1.2. Clasificación de Puntos Críticos (Wiens, 2024).....	12
1.3. Ejemplos de interacciones representadas por nodos (Alon, 2007) .	15
1.4. Análisis de bifurcación (Miriam, 2017).....	17
1.5. Elementos que componen una red neuronal	20
1.6. Tipos de función de activación (Redacción KeepCoding, 2022).....	21
2.1. Arquitectura de la Red Neuronal	32
3.1. Solución al modelo toggle switch por fsolve.....	35
3.2. Red neuronal problema directo modelo base	36
3.3. Comparación predicción entre el modelo base y la variación con el mejor desempeño.....	37
3.4. Comparación curvas de aprendizaje entre el modelo base y la variación con el mejor desempeño	39
3.5. Resultados problema inverso	41

1

Contexto Teórico

1.1. Planteamiento del problema y objetivo de la investigación

El estudio de las redes de regulación génica resalta la complejidad de los organismos vivos, ya que implica comprender los intrincados mecanismos que regulan la expresión genética más allá del número de genes, esta dificultad se debe a la complejidad inherente del sistema biológico y a la falta de datos completos y claros. En este contexto, el modelo dinámico del interruptor de palanca, o toggle switch en inglés, es crucial porque permite representar estados estacionarios donde dos genes reguladores pueden alternar entre estados estables mutuamente excluyentes, capturando así dinámicas complejas como la estabilidad y la alternancia entre estados. Además, facilita el análisis de parámetros críticos en el problema inverso para entender cómo se establecen y mantienen los estados de expresión génica, proporcionando insights valiosos sobre la regulación génica en diferentes condiciones ambientales y patológicas.

Capturar adecuadamente el comportamiento del sistema en estos puntos estacionarios es esencial al diseñar modelos matemáticos para sistemas biológicos, ya que son comúnmente observados en experimentos y proporcionan información crucial sobre la cinética y la regulación de la expresión génica. Identificar los parámetros del modelo a partir de estos puntos estacionarios permite inferir la estructura y dinámica del sistema biológico subyacente, lo que mejora nuestra comprensión de los procesos biológicos a nivel molecular y celular, incluida su respuesta a diferentes condiciones ambientales o perturbaciones externas.

El problema inverso no es trivial ya que una observación puede surgir de varios conjuntos de parámetros. Esto significa que diferentes combinaciones de parámetros pueden llevar a los mismos puntos estacionarios, lo que complica la tarea de determinar cuáles son los valores correctos. Además, en muchos casos

no sabemos cuántos puntos estacionarios existen para un sistema dado, lo que añade otra capa de complejidad al problema. Abordar esta incertidumbre requiere métodos que puedan considerar múltiples soluciones posibles en base a datos experimentales y conocimientos previos del sistema.

Por lo que se plantea la siguiente pregunta:

¿Cómo podemos resolver los desafíos asociados con el problema inverso del modelo toggle switch en las redes de regulación génica?

Para encontrar solución a esta pregunta, se propone como hipótesis que el uso de métodos computacionales como una red neuronal podría abordar los desafíos que presenta la identificación de conjuntos de parámetros en el problema inverso del modelo toggle switch en las redes de regulación génica. Esto se debe a la capacidad de las redes neuronales para aprender y modelar relaciones complejas de manera general, entre las variables del sistema, lo que les permite inferir conjuntos de parámetros a partir de datos de puntos estacionarios con mayor precisión y eficacia en comparación con otros enfoques tradicionales.

1.2. Marco Teórico

1.2.1. Biología de Sistemas

La biología de sistemas es un enfoque interdisciplinario basado en métodos teóricos, experimentales y computacionales. Se centra en la modelización matemática para comprender las interacciones no lineales entre los componentes celulares. Esta modelización busca producir modelos predictivos que puedan ser probados mediante simulaciones y experimentos reales. La biología sintética aplica principios de ingeniería para diseñar y construir nuevos sistemas biológicos con funcionalidades novedosas, como circuitos genéticos modulares para controlar la expresión génica o detectar señales ambientales. La biología de sistemas también implica la integración de datos experimentales y modelos matemáticos para comprender y predecir el comportamiento de los sistemas biológicos complejos. Algunos de los temas clave en la biología de sistemas incluyen la dinámica de redes de genes y proteínas, la señalización celular, la homeostasis y la adaptación, y la evolución de sistemas biológicos. (Del Vecchio & M. Murray, 1975; Kremling, 2014)

Principios y conceptos de la biología de sistemas

Los principios fundamentales de la biología de sistemas incluyen la noción de que los seres vivos son sistemas complejos con interacciones dinámicas entre sus componentes, y que estas interacciones pueden modelarse y estudiarse mediante herramientas derivadas de la teoría del control y la ingeniería de sistemas. La

biología de sistemas también se concentra en comprender los circuitos genéticos y las redes moleculares dentro de las células, así como en capacitar para diseñar y construir sistemas biológicos con funciones específicas. Asimismo, reconoce el valor crucial de la composicionalidad; es decir, combinar módulos biológicos para crear sistemas más complejos con comportamientos predecibles. Estos conceptos son vitales para afrontar desafíos tanto en ingeniería biomolecular como en aplicaciones prácticas relacionadas con medicina, energía y medio ambiente. Algunos de los conceptos básicos de la biología de sistemas incluyen (Rainer, 2010; Kitano, H., 2022; Peña Jiménez, 2016; Kremling, 2014):

- **Sistemas biológicos:**

Los sistemas biológicos son complejos y están formados por múltiples componentes interconectados, como genes, proteínas, células y organismos completos. Estos sistemas muestran propiedades emergentes que no pueden entenderse completamente al estudiar sus componentes por separado. Pueden ser investigados en diversas escalas, desde lo molecular hasta lo organismal, e implican interacciones dentro y entre las células. La comprensión de la dinámica y regulación de estos sistemas es crucial para avances en medicina, biotecnología y bioingeniería. La biología de sistemas se centra en comprender cómo interactúan estos componentes para generar comportamientos emergentes a nivel celular y organizativo, utilizando enfoques integrados matemáticos, computacionales y experimentales. Estos sistemas exhiben características como retroalimentación, variabilidad, adaptabilidad y autorregulación.

- **Dinámica de sistemas:**

La dinámica de sistemas en biología se centra en el comportamiento de los sistemas biológicos a lo largo del tiempo, identificando patrones en datos experimentales y construyendo modelos matemáticos para describir este comportamiento. En la biología de sistemas, esto implica comprender cómo las señales y las interacciones moleculares afectan el comportamiento global de un sistema biológico, como las redes de regulación génica y las respuestas celulares a estímulos. Los modelos de dinámica de sistemas se construyen mediante diagramas de flujo que representan las relaciones entre variables y cómo cambian en el tiempo, permitiendo simular diferentes escenarios y predecir el comportamiento del sistema bajo diversas condiciones. Este estudio es esencial para comprender procesos como la diferenciación celular, la respuesta inmune y la enfermedad, y es fundamental para el diseño de terapias médicas, ingeniería de tejidos y biotecnología.

- **Regulación genética:**

La regulación génica es esencial para el control y coordinación de la expresión de genes en las células, lo cual influye en el desarrollo, la diferenciación celular, la respuesta a estímulos y la homeostasis del organismo. Su estudio es crucial para comprender cómo se establecen y mantienen los patrones

de expresión génica, así como para entender su implicación en enfermedades genéticas y trastornos. La manipulación de la regulación génica tiene aplicaciones en medicina, biotecnología e ingeniería de sistemas biológicos. La biología de sistemas se enfoca en comprender cómo las señales celulares afectan el comportamiento de los sistemas biológicos, donde la regulación génica desempeña un papel fundamental. Esta regulación permite que los genes se expresen en el momento y lugar adecuados, mediante diversos mecanismos como la regulación transcripcional, la estabilidad del ARN mensajero, la traducción y la degradación de proteínas. Los mecanismos de regulación pueden involucrar activación o represión génica, mediados por diversos factores como proteínas reguladoras, factores de transcripción, ARN no codificante y modificaciones epigenéticas.

■ **Modelización matemática:**

Esta herramienta es fundamental en la biología de sistemas, ya que permite representar y analizar cuantitativamente la dinámica de los sistemas biológicos. Los modelos matemáticos pueden ser utilizados para predecir cómo los sistemas biológicos responderán a diferentes estímulos y condiciones ambientales, y para identificar los puntos críticos de control en las redes de regulación génica y metabólica. Los modelos matemáticos también pueden ser utilizados para diseñar y optimizar sistemas biológicos sintéticos con funcionalidades específicas. Pueden ser de diferentes tipos, incluyendo modelos determinísticos y estocásticos, modelos de ecuaciones diferenciales y modelos de redes booleanas. Estos modelos pueden ser validados y refinados mediante la comparación con datos experimentales, lo que permite mejorar la precisión y la utilidad de los modelos. Aquí hay algunas maneras en las que se aplica el modelado matemático en la biología de sistemas:

- Representación de redes biológicas
- Simulación de dinámicas temporales
- Predicción de comportamientos emergentes
- Análisis de sensibilidad y robustez

■ **Señalización celular:**

La señalización celular se refiere a cómo las células se comunican entre sí para coordinar su comportamiento. La biología de sistemas se enfoca en comprender cómo se transmiten las señales celulares y cómo estas señales afectan el comportamiento de los sistemas biológicos.

1.2.2. Redes de regulación génica

Una red de regulación génica es un sistema complejo de interacciones entre genes y sus elementos como proteínas y otras moléculas dentro de una célula. Involucra las interacciones y los mecanismos como la regulación de la transcripción,

traducción y otros procesos. Las redes de regulación génica son vitales para el correcto funcionamiento de las células y organismos, y tienen un papel fundamental en el desarrollo, la diferenciación y la reacción a estímulos del entorno. (Domitilla Del Vecchio, 2018) (Kremling, 2014)

Importancia del control de expresión génica en la función y desarrollo de los organismos

Las células eucariotas regulan sus actividades mediante complejos mecanismos que involucran ácidos nucleicos y proteínas. Después de la traducción, las modificaciones postraduccionales de las proteínas contribuyen a su diversidad estructural y funcional. Cuando ciertos parámetros celulares son críticos, las células emplean múltiples mecanismos reguladores para garantizar un control preciso de sus actividades. Desde ajustes en la síntesis del ARN hasta modificaciones postraduccionales, cada etapa está finamente regulada. Estos procesos incluyen el atenuamiento del transcrito, selección alternativa del punto de maduración, control del extremo 3' mediante adición de poli-A, edición del ARN, transporte núcleo-citosol, localización específica del mRNA y regulación del inicio y la degradación del mRNA. Este sistema opera a varios niveles para garantizar la expresión génica adecuada, destacando la interconexión entre fases pre- y post-transcripcionales en la compleja danza celular. (Junqueira L. C., 2012; Alberts, 2002; Salazar Montes A. M., 2013; Salazar Montes, Sandoval Rodríguez, & Armendáriz Borunda, 2013)

Análisis y visualización de redes de regulación genética

Existen distintos métodos para analizar y visualizar las redes de regulación génica (Lillo Egaña, 2010), como:

1. Diagramas de Redes de Regulación Genética:

Utilizan nodos para representar genes y enlaces para mostrar las interacciones entre ellos, lo que permite visualizar las relaciones de activación o represión entre genes y proteínas.

2. Análisis de Expresión Génica:

Técnicas como microarrays y secuenciación de ARN (RNA-Seq) permiten medir los niveles de expresión de genes en diferentes condiciones, lo que proporciona información para construir y validar redes de regulación genética.

3. Modelado Computacional:

El modelado matemático y computacional se utiliza para simular y predecir el comportamiento de las redes de regulación genética, lo que ayuda a comprender cómo las interacciones entre genes dan forma a los fenotipos celulares.

4. Visualización de Datos de Interacción Genética:

Herramientas de visualización de datos, como Cytoscape, permiten representar gráficamente las interacciones entre genes, proteínas y moléculas, facilitando la comprensión de las redes de regulación genética.

5. Análisis de Redes de Coexpresión:

Identifica genes que muestran perfiles de expresión similares, lo que sugiere interacciones funcionales y regulación coordinada, contribuyendo a la comprensión de las redes de regulación genética.

6. Enfoques de Biología de Sistemas:

La integración de datos genómicos, transcriptómicos, proteómicos y metabolómicos permite construir modelos integrados de redes de regulación genética para comprender los sistemas biológicos a un nivel más completo.

1.2.3. Sistemas Dinámicos

Un sistema dinámico es un concepto en matemáticas y ciencia que describe cómo cambia un sistema con el tiempo. Los sistemas dinámicos pueden ser discretos o continuos, lineales o no lineales, y se modelan mediante ecuaciones diferenciales o mapas iterativos.

Características de los sistemas dinámicos

De acuerdo con los apuntes de los curso (Perez y Escobar, 2024; Bolós, 2024) ¹.

²

1. Estado del sistema:

El estado de un sistema dinámico se define por un conjunto de variables que describen completamente el sistema en un momento dado. Estas variables pueden ser posiciones, velocidades, concentraciones, etc., dependiendo del contexto del sistema.

2. Espacio Fase:

Es el espacio multidimensional en el que cada eje corresponde a una variable del sistema. El estado del sistema en cualquier momento es un punto en este espacio.

¹Katya Perez M. y Carla Escobar O., *Modelación en Variables de Estado, Capítulo 4, Curso: Ingeniería de Sistemas I*, apuntes de curso obtenidos de <https://virtual.usalesiana.edu.bo>, fecha de consulta: 19-6-24.

²Vicente J. Bolós, *Matemáticas para los Modelos Dinámicos, Apuntes, Curso Matemáticas para los Modelos Dinámicos*, <https://www.uv.es/vbolos/>, consulta: 19-6-24.

3. Evolución temporal:

La evolución temporal de un sistema dinámico describe cómo cambian las variables de estado con el tiempo. Dependiendo de si el sistema es discreto o continuo, esta evolución se describe mediante ecuaciones diferenciales o funciones iterativas.

- **Sistemas Dinámicos Discretos:**

En la dinámica discreta se estudia una magnitud X que toma valores en un conjunto discreto de instantes de tiempo $t_0, t_1, t_2, \dots$, que supondremos ordenado de menor a mayor, normalmente se considera $t_0 = 0, t_1 = 1, \dots$ para simplificar. Utilizando la notación X_n para referirnos al valor de X en el instante t_n .

La evolución de dicha magnitud está regida por una ley, una expresión en forma de ecuación en la que se relacionan los valores de X en diversos instantes, y en donde X hace el papel de incógnita. En general, cualquier expresión de la forma:

$$F(X_{n+k}, X_{n+k-1}, \dots, X_n, n) = 0$$

en donde F es una función dada y k es un número natural positivo, conforma un sistema dinámico discreto. De este modo cualquier expresión matemática equivalente recibe el nombre de ecuación en diferencias (ED). En este caso k es el orden de la ED.

Solución de una ED

Llamaremos solución de una ED, a toda sucesión $S = \{S_n\}_{n \in J}$, diremos que S_n es solución si $\forall n \in J, F(\{S_n\}) = 0$ y $\{S_n\} \in \text{dom}(F)$.

Normalmente las EDs tienen infinitas soluciones y para distinguir a una de ellas hay que añadir algunas condiciones adicionales. En concreto, dada una ED de orden k , si precisamos cuál es el valor de la magnitud X en k instantes consecutivos, se formula el llamado problema de valores iniciales (PVI). En este caso, si la ecuación está en forma explícita, puede asegurarse que existe una única solución.

- **Sistemas Dinámicos Continuos (Ecuaciones Diferenciales Ordinarias):**

En general, una ecuación diferencial ordinaria (EDO) es una expresión funcional que relaciona a una variable independiente $t \in \mathbb{R}$ con una función incógnita $X = X(t)$ y con algunas de sus derivadas.

$$x'(t) = ax(t)$$

Solución de una EDO

Si k es un número real, entonces la función $x(t) = ke^{at}$ es una solución cuando:

$$x'(t) = ake^{at} = ax(t)$$

La constante k está completamente determinada si definimos el valor de u_0 como solución para un punto específico t_0 . Supongamos que se requiere que una función $x(t)$ que satisfaga la ecuación diferencial también satisfaga $x(t_0) = u_0$. Entonces debemos tener $ke^{at_0} = u_0$, de modo que $k = u_0e^{-at_0}$. Por lo tanto, hemos determinado k y, por lo tanto, esta ecuación tiene una solución única que satisface la condición inicial especificada $x(t_0) = u_0$. Para simplificar, a menudo tomamos $t_0 = 0$; luego $k = u_0$. No hay pérdida de generalidad al tomar $t_0 = 0$, ya que si $u(t)$ es una solución con $u(0) = u_0$, entonces la función $v(t) = u(t - t_0)$ es una solución con $v(t_0) = u_0$.

4. Estabilidad:

La estabilidad analiza cómo responde un sistema a pequeñas perturbaciones en su estado inicial.

- Estabilidad Asintótica: Un estado es asintóticamente estable si el sistema tiende a volver a ese estado después de una perturbación.
- Inestabilidad: Un estado es inestable si pequeñas perturbaciones crecen con el tiempo.
- Estabilidad Neutra: Un estado es estable neutro si las perturbaciones ni crecen ni decrecen

5. Atractores:

De acuerdo con los apuntes de los curso (Barrigüete, 2024) ³.

Un atractor en un sistema dinámico es un conjunto de estados hacia los cuales el sistema evoluciona después de un tiempo suficientemente largo, independientemente de las condiciones iniciales, siempre que estas se encuentren dentro de una cierta región llamada la cuenca de atracción.

Tipos de Atractores

- Puntos Fijos (Atractores Estacionarios): Son estados donde el sistema no cambia con el tiempo. Puede ser un nodo atractivo (el sistema se dirige hacia él) o un nodo repulsivo (el sistema se aleja de él).
- Ciclos Límites (Atractores Periódicos): Son trayectorias cerradas en el espacio fases hacia las cuales el sistema converge y que representan un comportamiento periódico. Un ciclo límite atractivo hace que las trayectorias cercanas se dirijan hacia él con el tiempo.
- Atractores Torus (Toroidales): Son atractores que tienen una estructura de todo (dona), representando oscilaciones casi periódicas en múltiples frecuencias.

³José Luis Arroyo Barrigüete, *Introducción a los sistemas dinámicos, Capítulo 4*, apuntes de curso obtenidos de <https://frdelpino.es>, fecha de consulta: 19-6-24.

- **Atractores Extraños:** Tienen una estructura fractal y son comunes en sistemas caóticos. Atraen las trayectorias cercanas pero son extremadamente sensibles a las condiciones iniciales.

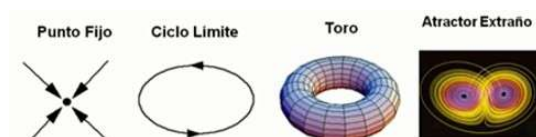


Figura 1.1: Tipos de Atractores (Cáceres & O'Farrill, Chaos Culture, 2021)

Cuenca de Atracción

La cuenca de atracción de un atractor es el conjunto de todos los puntos en el espacio de fases que eventualmente evolucionarán hacia ese atractor. Si un sistema tiene múltiples atractores, sus cuencas de atracción dividen el espacio de fases en regiones distintas.

6. Bifurcaciones ⁴.

Considerando la siguiente familia de ecuaciones diferenciales:

$$\dot{x} = f(x, a), x \in \mathbb{R}^n, a \in \mathbb{R}$$

donde $x \in \mathbb{R}$ es la variable de estado y $a \in \mathbb{R}$ es un parámetro real que no depende del tiempo t . Lo que se busca es analizar el comportamiento de las soluciones cuando se varía el parámetro a . Entonces dados dos valores de parámetros, existen dos posibilidades: que los sistemas queden topológicamente equivalentes o no.

Las bifurcaciones ocurren cuando un pequeño cambio en los parámetros de un sistema dinámico causa un cambio cualitativo en su comportamiento. Esto significa que el sistema puede pasar de un tipo de comportamiento a otro, como de un punto fijo a un ciclo límite, o de comportamiento periódico a caótico.

7. Diagramas de Fase

Un diagrama de fase es una representación gráfica que muestra las trayectorias de un sistema dinámico en el espacio de fases. Los diagramas de fase proporcionan una visión clara del comportamiento cualitativo del sistema, incluyendo sus puntos de equilibrio, ciclos límite y posibles comportamientos caóticos.

⁴Betancur, Hector, *Bifurcaciones básicas y formas normales*, Trabajo final de maestría, Universidad Nacional de Colombia Facultad de Ciencias Departamento de Matemáticas, 2010

Componentes de un Diagrama de Fase

- Ejes del Espacio de Fases: Cada eje representa una variable del sistema.
- Trayectorias o Órbitas: Curvas en el espacio de fases que muestran cómo evoluciona el estado del sistema con el tiempo. Cada punto en la trayectoria corresponde a un estado del sistema en un momento dado.
- Puntos Fijos (Equilibrios): Puntos en el espacio de fases donde el sistema no cambia con el tiempo. En un diagrama de fase, estos puntos se identifican como los lugares donde las trayectorias convergen o divergen.
- Vectores de Flujo: Representan la dirección del cambio en cada punto del espacio de fases.

1.2.4. Sistema Biestable

La teoría de la estabilidad se centra en analizar las soluciones de ecuaciones diferenciales y las trayectorias de sistemas dinámicos frente a pequeñas variaciones en las condiciones iniciales. Se han desarrollado múltiples criterios para determinar si una órbita es estable o inestable. En ciertas circunstancias favorables, el problema puede simplificarse a un tema bien estudiado que implica los autovalores de las matrices. En la práctica, se aplican varios criterios distintos para evaluar la estabilidad. (Angeli, 2004)

Estabilidad para sistemas lineales

Si existe una matriz $A \in \mathbb{R}^{n \times n}$ de forma que $\dot{x} = A x$ es un sistema lineal. Considerando el polinomio característico

$$p(\lambda) = \det(A - \lambda \cdot I)$$

Podemos verificar que:

1. Si todas las raíces de $p(\lambda) = 0$ tienen parte real negativa, todas las soluciones del sistema son asintóticamente estables.
2. Si al menos una de las raíces de $p(\lambda) = 0$ tiene parte real positiva, todas las soluciones son inestables.
3. Suponiendo que todas las raíces tienen parte real no positiva y que

$$\lambda_1 = i\sigma_1, \dots, \lambda_l = i\sigma_l$$

son todos los que tienen parte real 0. Sea κ_i la multiplicidad de $\lambda_i = i\omega_i$. Esto significa que el $p(\lambda)$ puede factorizarse como:

$$p(\lambda) = (\lambda - i\omega_1)^{\kappa_1} \dots (\lambda - i\omega_l)^{\kappa_l} Q(\lambda)$$

donde Q son las raíces con parte real negativa. Entonces, toda solución $x = x(t)$ es estable, pero no asintóticamente, si para todo $j = 1, \dots, l$ la matriz A tiene κ_j autovectores linealmente independientes del autovalor $\lambda_j = i\omega_j$. En otro caso, toda la solución es inestable. (Jiménez Jiménez, 2013; Hernández Paredes, 2015)

Existen otros criterios para demostrar la estabilidad de las soluciones para sistemas de ecuaciones lineales como el Criterio de Routh-Hurwitz (Valverde, 2004) con el cual podemos ubicar la posición de las soluciones en el plano complejo y determinar su estabilidad, también el Criterio del Círculo de Gershgorin (Varga, 2010; Curis, 1984) que es utilizado en algebra lineal para encontrar una cota de los valores propios de una matriz cuadrada.

Estabilidad en sistemas no lineales

En el análisis de estabilidad de sistemas no lineales, los métodos utilizados son más complejos y dependen del comportamiento específico, en su mayoría los sistemas son de naturaleza no lineal, pero es posible hacer aproximaciones para considerarlos como sistemas lineales en ciertas condiciones. Para los sistemas lineales, la estabilidad se puede analizar fácilmente a través de los puntos de equilibrio y las propiedades de los eigenvalores y eigenvectores de la matriz asociada al sistema. Sin embargo cuando se tratan de sistemas no lineales, estos métodos no tienen validez. (Turner, 1960; Hernández Paredes, Ampliación de Análisis Numérico - Tema 3. Resolución de ecuaciones no lineales., 2015)

Uno de los fenómenos que diferencian los sistemas lineales de los no lineales es la existencia de múltiples puntos de equilibrio aislado. En un sistema lineal la existencia de un punto de equilibrio aislado indica un único estado estacionario que atrae el estado del sistema independiente del estado inicial. En cambio, los sistemas no lineales pueden tener múltiples puntos de equilibrio, lo que significa que hay varios estados estacionarios posibles (DeJesus, 1987) y la convergencia a uno estable depende del estado inicial.

Para poder visualizar un sistema no lineal como uno lineal podemos considerar lo siguiente: Sea $\dot{x} = f(t, x, u)$ donde $x \in \mathbb{R}^n$ y representa los estados, $u \in \mathbb{R}_p$ y representa las variables de entrada.

El sistema no lineal puede ser aproximado mediante un sistema linealizado si (J. K. Hale, 1967):

1. El rango de operación es pequeño.
2. Las no linealidades son suaves.

3. Su dinámica se describe mediante un conjunto de EDL.

Pequeñas variaciones en las condiciones iniciales pueden generar pequeñas disparidades en el resultado, a menudo denominado caos. Un sistema estable eventualmente converge hacia un punto de equilibrio o una órbita, dependiendo de su dimensión (atractor). Por otro lado, un sistema inestable escapa de los trayentes. Un sistema caótico muestra ambos comportamientos simultáneamente. De esta manera, el sistema permanece restringido a una región específica dentro de su espacio estatal, pero no se dirige hacia un atractor fijo. (Grebogi, 1987; Anzures M. Juan, 2008)

Clasificación de los puntos críticos

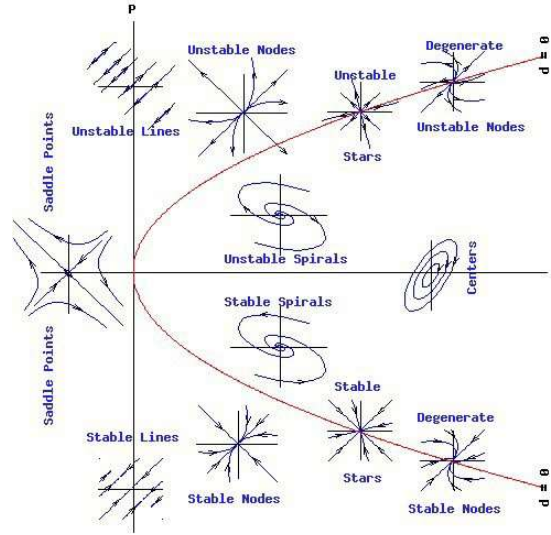


Figura 1.2: Clasificación de Puntos Críticos (Wiens, 2024)

Sea

$$\begin{aligned} p &= \text{tr}(A), \\ q &= \det(A), \\ d &= \text{disc}(A). \end{aligned}$$

La parábola $p^2 - 4q = 0$ es el lugar geométrico de puntos (p, q) para los cuales $d = 0$ [Wiens, 2024].

- Dentro del área cerrada por la parábola, si $p = 0$ y los valores propios son números complejos, se obtienen trayectorias en forma de espiral.

- Si $p = 0$, los eigenvalores son puramente imaginarios y las trayectorias rodean al punto fijo en forma de elipses o círculos.
- Si $p > 0$ y $q < 0$, ambos eigenvalores tienen partes reales positivas y el punto fijo es inestable.
- Si $p > 0$ y $q = 0$, ambos eigenvalores tienen partes reales negativas y el punto fijo es estable.
- Si $p < 0$ y $q = 0$, un eigenvalor es negativo y el otro es cero, por lo tanto las trayectorias son líneas inestables.
- Sin embargo, si $q < 0$, un eigenvalor es positivo y el otro es negativo, las trayectorias siguen un patrón de puntos de silla.
- Si $p = 0$ y $q = 0$: si A es la matriz nula, cualquier vector se convierte en un punto de equilibrio. Si A no es una matriz nula, entonces los eigenvectores asociados a eigenvalores cero forman una línea de puntos de equilibrio. El resto de las trayectorias son paralelas a esta línea y siguen direcciones opuestas a ambos lados de la línea de equilibrio. (Jiménez Jiménez, 2013; Wiens, 2024; GutiérrezG., 2016)

Biestabilidad

La biestabilidad es un fenómeno que ocurre en sistemas no lineales donde existen dos estados estables que coexisten. Estos dos estados estables pueden ser alcanzados dependiendo de las condiciones iniciales o de otros parámetros del sistema. En el contexto de la biología de sistemas, la biestabilidad es especialmente relevante en el estudio de las redes de regulación genética. La biestabilidad es la llave para entender el funcionamiento celular en los procesos de toma de decisiones en el ciclo celular, diferenciación celular y apoptosis.

Los sistemas biológicos se basan en reacciones químicas elementales mono y biomoleculares. El sistema mínimo biestable contiene dos reactivos y cuatro reacciones irreversibles (tres biomoleculares y un mono molecular). (Thomas, 2009; Veening, Klass Smits, & P. Kuipers, 2008)

Los interruptores biestables suelen ser activados por bucles de retroalimentación positiva en redes de transducción de señales. Por este motivo, se les denomina "interruptores de decisión". Un ejemplo que se discute con frecuencia es el control del punto crítico para regular la transición G1-S en el ciclo celular de mamíferos, donde recientemente se ha formulado una detallada ecuación diferencial ordinaria.

Un problema frecuentemente discutido es la condición necesaria (y/o suficiente) para que los sistemas diferenciales autónomos de biestabilidad puedan poseer múltiples estados estables solo bajo la presencia de bucles de retroalimentación positiva. Se argumentó también que la retroalimentación necesita algún tipo de 'no-linealidad' o 'ultra sensibilidad' para inducir biestabilidad. La teoría de redes de reacción química de Feinberg incluso proporciona condiciones necesarias

y suficientes para la biestabilidad, al restringirse a sistemas cinéticos especiales de acción masiva.

La identificación de las condiciones estructurales necesarias para cualquier comportamiento dinámico es la identificación de los sistemas mínimos correspondientes. (Thomas, 2009; Veening, Klass Smits, & P. Kuipers, 2008)

El sistema de reacción química biestable más pequeño (Thomas, 2009). Definimos el sistema químico más pequeño (que contiene solo reacciones mono- y biomoleculares, las reacciones reversibles se consideran como dos irreversibles) según los siguientes criterios en orden decreciente de importancia:

1. Número mínimo de reactantes
2. Número mínimo de reacciones
3. Número mínimo de términos en las EDO

1.2.5. Toggle Switch

El toggle switch o interruptor de palanca en el contexto de la biología de sistemas, esta compuesto por dos genes que se reprimen mutuamente. Este circuito biológico puede existir en uno de dos estados estables y puede cambiar entre estos dos estados bajo ciertas condiciones. En el estado 1 cuando el gen A esta activado, produce una proteína que inhibe (bloquea) al gen B, esto implica que el gen B esta apagado y no produce su proteína. En el estado 2 si el gen B esta activo produce su proteína que bloquea al gen A. Si hay una señal que activa el gen A (por ejemplo, un cambio en el entorno), el gen A empieza a producir su proteína. Esta proteína bloquea al gen B manteniendo lo apagado, de la misma manera ocurre cuando se cambia al estado 2. (Domitilla, 1975)

El toggle switch se puede interpretar como un sistema dinámico no lineal, porque su comportamiento y cambios de estado se pueden describir mediante ecuaciones diferenciales. En este contexto, cada gen y su respectiva proteína representan variables dinámicas cuyas concentraciones cambian con el tiempo. Las ecuaciones que modelan este sistema dinámico consideran las tasas de producción y degradación de las proteínas, así como la represión mutua entre los genes.

Miriam León nos menciona en su tesis (Miriam, 2017) que: “Los primeros dispositivos sintéticos exitosos es el interruptor genético alternante. El cual consiste en un conjunto de factores de transcripción que se reprimen mutuamente, lo que resulta en la existencia de dos estados fenotípicos distintos pero sin estado intermedio” (Lohmueller, 2007; Timothy S. Gardner, 2000)

Aun no existe un acuerdo sobre la estabilidad que puede lograr un interruptor y el método más adecuado para modelarlo. Diferentes enfoques conducen a conclusiones divergentes, lo que genera incertidumbre sobre qué comportamiento esperar incluso para un sistema simple como el interruptor, compuesto solo por dos genes. Hasta que su comportamiento pueda predecirse con precisión, el interruptor no podrá ser utilizado como componente constitutivo de sistemas

más grandes y complejos. Por lo tanto, será prácticamente imposible diseñar sistemas con comportamientos predecibles hasta ese momento. (Wilkinson, 2006; Veening, Klass Smits, & P. Kuipers, 2008; Miriam, 2017)

El modelado tiene como objetivo explicar los elementos y el funcionamiento del sistema bioquímico que se está estudiando. Esta técnica se utiliza para integrar información y datos experimentales, así como predecir el comportamiento del sistema. Las reacciones químicas acopladas son comúnmente empleadas para representar las redes de transcripción en biología de sistemas, ya que permiten describir de manera concisa un sistema y luego ser utilizadas en una variedad de métodos de simulación con interpretaciones específicas sobre la cinética química. (Wilkinson, 2006)

Representación grafica de sistemas bioquímicos

En un gráfico, los nodos simbolizan las distintas especies y las aristas indican la interacción entre estas especies. Por ejemplo, un factor de transcripción puede influir directamente en la transcripción de un gen. Una flecha al final de una arista indica activación, lo que significa que cuando el factor de transcripción se une al promotor, aumenta la tasa de transcripción del gen. En cambio, una línea plana perpendicular a la arista representa represión; es decir, cuando el factor de transcripción se asocia con el promotor disminuye la tasa de transcripción del gene. (R. Milo, 2002; Alon, 2007)

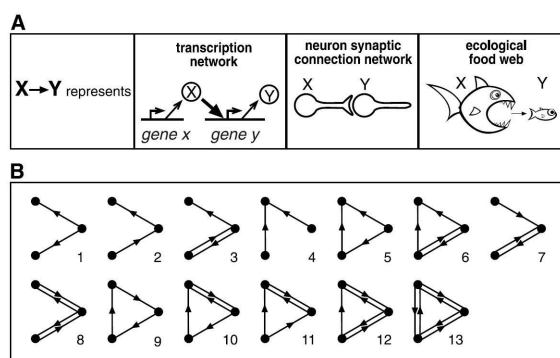


Figura 1.3: Ejemplos de interacciones representadas por nodos (Alon, 2007)
 (A) Ejemplos de interacciones representadas por aristas dirigidas entre nodos en algunas de las redes utilizadas para el presente estudio. Estas redes van desde la escala de biomoléculas (la proteína del factor de transcripción X se une a regiones de ADN reguladoras de un gen para regular la tasa de producción de la proteína Y), pasando por células (la neurona X está conectada sinápticamente a la neurona Y), hasta organismos (X se alimenta de Y). (B) Todos los 13 tipos de subgrafos conectados de tres nodos.

Formalismo de Hill

El formalismo de Hill es una herramienta matemática que se utiliza para describir las interacciones entre los componentes de un sistema bioquímico donde un activador o un represor está presente. La función de Hill es representada matemáticamente como una ecuación de potencia, donde la concentración del activador o represor se eleva a una potencia exponente llamado coeficiente de Hill. (Miriam, 2017; Hill, 1910)

$$\frac{dP}{dt} = V_{\max} \frac{x^n}{1+x^n}$$

El interruptor genético biestable

El interruptor genético sintético consiste en dos factores de transcripción que se reprimen mutuamente. (Miriam, 2017) Fue desarrollado por primera vez por Gardner, Cantor y Collins y consta de las siguientes ODE:

$$\begin{aligned}\frac{du}{dt} &= \frac{a_1}{1+v^\gamma} - u \\ \frac{dv}{dt} &= \frac{a_2}{1+u^\delta} - v\end{aligned}$$

donde u es la concentración del represor 1, v la concentración del represor 2, a_1 y a_2 denota las tasas de producción de los represores, y δ y γ son los parámetros que representan el grado de cooperatividad de la represión entre los represores. (Timothy S. Gardner, 2000)

En el trabajo de Miriam León (Miriam, 2017), podemos ver un análisis de bifurcación que es usado para determinar las propiedades de un sistema en un espacio de parámetros. Miriam León hace uso de PyDSTool, una paquetería de Python usada para el análisis de sistemas dinámicos, los parámetros los escoge del análisis del espacio fase están dentro del rango sugerido por Gardner, Cantor y Collins (2000) (Timothy S. Gardner, 2000), los parámetros a_1 y a_2 se establecen en 10, y δ , γ se establece en 2. (Figura 1.2)

1.2.6. Problemas inversos

Los problemas inversos son aquellos en los que se busca determinar las causas o condiciones que dieron lugar a un resultado o efecto dado. En el modelado matemático, los problemas inversos se utilizan para determinar los parámetros o condiciones iniciales que mejor se ajustan a los datos observados. Por ejemplo, en el modelado de sistemas dinámicos, como el interruptor genético biestable en la biología sintética. Implica inferir los parámetros del modelo a partir de datos experimentales para mejorar la comprensión del comportamiento del sistema. El enfoque de los problemas inversos en biología sintética es fundamental para avanzar en la capacidad de diseñar y manipular sistemas biológicos a medida.

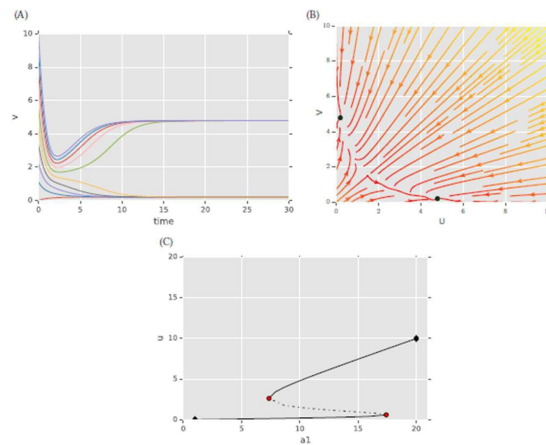


Figura 1.4: Análisis de bifurcación (Miriam, 2017)

- (A) El curso temporal del modelo simulado usando múltiples condiciones iniciales para v .
- (B) El diagrama vectorial del interruptor de Gardner muestra que hay dos estados estables en equilibrio en $(u, v) = (4, 791, 0, 208)$ y $(u, v) = (0, 208, 4, 791)$.
- (C) Un diagrama de bifurcación muestra que el sistema es biestable cuando $7 \geq a_1 \leq 17$.

Al comprender la relación entre los parámetros del modelo y los datos experimentales, se pueden realizar predicciones más precisas sobre el comportamiento del sistema, lo que es esencial para el desarrollo de aplicaciones prácticas en la ingeniería de sistemas biológicos.

Además, la resolución de problemas inversos en biología sintética puede proporcionar información valiosa sobre las limitaciones y las posibles mejoras en los modelos teóricos existentes, lo que a su vez puede impulsar el desarrollo de estrategias más efectivas para la ingeniería de sistemas biológicos. (Luaces Martínez, 2010)

Métodos y enfoques utilizados para resolver problemas inversos

Para resolver estos problemas, se recurre a una variedad de métodos y enfoques que incluyen (Lillo Egaña, 2010; O. Wolernhauer, 2004; Kremling, 2014):

1. Optimización: Los métodos de optimización, como el descenso del gradiente, algoritmos genéticos, algoritmos de enjambre de partículas, entre otros, se utilizan para encontrar los valores de los parámetros que minimizan la diferencia entre las observaciones experimentales y las predicciones del modelo.
2. Filtrado Bayesiano: El filtrado bayesiano, que incluye métodos como el

filtro de Kalman y el filtro de partículas, es utilizado para estimar estados no observados en sistemas dinámicos, considerando la incertidumbre y actualizando continuamente las estimaciones a medida que se obtienen nuevas observaciones.

3. Regularización: La regularización se emplea para controlar la complejidad del modelo y evitar el sobreajuste a los datos. Métodos como la regresión ridge, la regresión LASSO y otras técnicas de regularización son utilizadas para resolver problemas inversos en los que se busca encontrar parámetros óptimos.
4. Métodos de Identificación de Sistemas: Estos métodos se centran en la estimación de modelos a partir de datos experimentales, utilizando técnicas como la identificación de sistemas lineales, la identificación de sistemas no lineales, y métodos basados en la teoría de control.
5. Métodos de Reconstrucción de Imágenes: En el contexto de la tomografía y la reconstrucción de imágenes, se utilizan métodos como la transformada de Radón, la retroproyección filtrada, y algoritmos de reconstrucción iterativa para estimar la distribución de propiedades físicas a partir de datos de proyección.
6. Aprendizaje Automático: En problemas inversos complejos, se recurre a técnicas de aprendizaje automático, como redes neuronales, máquinas de vectores de soporte, y métodos de aprendizaje profundo, para estimar relaciones no lineales y realizar predicciones a partir de datos observados.

Método optimización descenso del gradiente

El método de descenso de gradiente es un enfoque iterativo para optimización en problemas diferenciales. Consiste en tomar pasos sucesivos en la dirección opuesta al gradiente, ya que esta dirección representa el mayor descenso. Por otro lado, seguir la misma dirección que el gradiente lleva al máximo local de la función, conocido como ascenso del gradiente. Este procedimiento se usa comúnmente para entrenar modelos de aprendizaje automático y redes neuronales (Lemaréchal, 2010)

Este método es descrito de la siguiente manera:

Sea $F(x)$ una función definida y derivable en una vecindad a un punto a , entonces $F(x)$ decrece rápidamente si se sigue la dirección opuesta al gradiente negativo de F en ese punto.

$$a_{t+1} = a_t - \gamma \nabla F(a_t)$$

para un tamaño de paso o tasa de aprendizaje lo suficientemente pequeño $\gamma \in \mathbb{R}^+$ $\gamma \nabla F(a_t)$ se resta de a_t por que se quiere avanzar contra la pendiente, hacia un mínimo local.

Es posible garantizar que se converja a un mismo mínimo local, bajo ciertas condiciones sobre la función F (convexo y GradF Lipschitz).

El tamaño del paso γ se le permite cambiar con cada iteración según diferentes métodos de ajuste, como por ejemplo el método del paso fijo, el método de búsqueda lineal o el método del paso adaptativo. Una γ muy pequeña ralentizaría la convergencia y una muy grande conduciría a un exceso y divergencia. (Barreto Salles & W. Anderson, 2008; Lemaréchal, 2010; Baldi, 1995)

1.2.7. Redes neuronales y su aplicación

Una red neuronal es un modelo matemático inspirado en la estructura y funcionamiento del cerebro humano, que se utiliza para realizar tareas de aprendizaje automático y reconocimiento de patrones. Está compuesta por un conjunto de nodos interconectados, llamados neuronas artificiales o unidades de procesamiento, que trabajan en conjunto para procesar información y realizar predicciones. Cada neurona artificial recibe una o varias entradas, que son procesadas mediante una función de activación no lineal, y produce una salida que se transmite a otras neuronas de la red. Las conexiones entre las neuronas se representan mediante pesos sinápticos, que determinan la fuerza y dirección de la influencia de una neurona sobre otra.

Las redes neuronales se entrenan mediante un proceso de aprendizaje supervisado o no supervisado, en el que se ajustan los pesos sinápticos para minimizar el error entre las predicciones de la red y los valores deseados. Una vez entrenada, la red neuronal puede utilizarse para realizar predicciones sobre nuevos datos de entrada. (Lillo Egaña, 2010; Nielsen, 2019)

Elementos que componen una red neuronal

Una red neuronal está compuesta por varios elementos fundamentales que permiten su funcionamiento. Algunos de estos elementos son: las neuronas artificiales (unidad de procesamiento), las conexiones entre las neuronas (pesos sinápticos), una función de activación, una función de pérdida y un algoritmo de optimización. (Wang, 2015; Munt, 2018; Mehrotra, 1996; Anderson, 1995)

- Función de entrada: Parcialmente la siguiente sección, aborda en su totalidad lo escrito en la cátedra impartida por los profesores Carlos Alberto Ruiz y Marta Susana Basualdo (Ruiz & Basualdo, 2001) y la cátedra impartida por el Dr. Jorge Velázquez (Velázquez Castro, 2022), la cual toma información proporcionada en el curso en línea de Michael Nielsen (Nielsen, 2019).

La función de entrada puede describirse como:

$$\text{input}_{(t+1)} = \text{input}_{(t)} \cdot \text{weight}_{(t)} * \dots * \text{input}_{(t-n)} \cdot \text{weight}_{(t-n)}$$

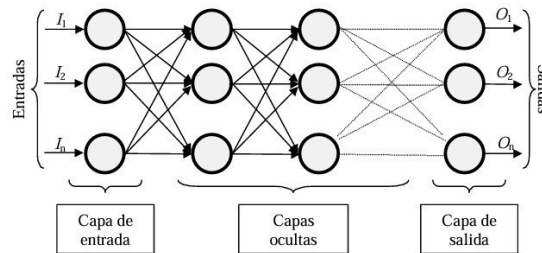


Figura 1.5: Elementos que componen una red neuronal

donde * representa al operador necesario, n es el número de entradas a la neurona a la neurona N_i y w_i al peso

Esta función toma todos los valores de entrada global y los combina en simples entradas. Los valores de entrada se multiplican por los pesos, que generalmente no están restringidos, cambian de acuerdo a la influencia de los valores de entrada.

Algunas funciones de entrada son las siguientes:

1. Sumatoria de las entradas pesadas.

$$\sum_j (n_{ij} w_{ij})$$

2. Productoria de las entradas pesadas.

$$\prod_j (n_{ij} w_{ij})$$

3. Máximo de las entradas pesadas.

$$\max_j (n_{ij} w_{ij})$$

- **Función de activación:** Las neuronas artificiales al igual que las neuronas biológicas pueden tener diferentes estados de activación, las cuales pueden tomar cualquier valor dentro de un conjunto determinado.

Esta función calcula el estado de actividad, transformando la entrada global menos un umbral θ_i en un estado de activación.

Las funciones de activación podrían ser:

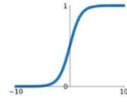
1. Lineal.
2. Sigmoidea.
3. Tangente hiperbólica.

- **Función de salida:** La función de salida determina que valor se transfiere a las neuronas vinculadas. Si la función de activación está por debajo de un umbral determinado, ninguna salida pasa a la neurona subsiguiente.

Activation Functions

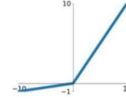
Sigmoid

$$\sigma(x) = \frac{1}{1+e^{-x}}$$



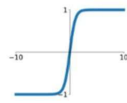
Leaky ReLU

$$\max(0.1x, x)$$



tanh

$$\tanh(x)$$

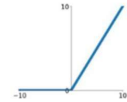


Maxout

$$\max(w_1^T x + b_1, w_2^T x + b_2)$$

ReLU

$$\max(0, x)$$



ELU

$$\begin{cases} x & x \geq 0 \\ \alpha(e^x - 1) & x < 0 \end{cases}$$



Figura 1.6: Tipos de función de activación (Redacción KeepCoding, 2022)

Arquitectura de una red neuronal

La distribución de las neuronas en una red, se realiza en capas.

- Entrada. Esta capa recibe la información.
- Oculta. Son neuronas internas que no tienen contacto con el exterior.
- Salida.

Tipos de neuronas

Estos se pueden clasificar de acuerdo a los valores que pueden tomar. La más comunes son:

- Neuronas Binarias.
- Neuronas reales.

Las neuronas binarias solamente pueden tomar valores dentro del intervalo 0, 1 o -1, 1, mientras que las neuronas reales pueden hacerlo dentro del rango [0, 1] o [-1, 1].

Tipo de Redes Neuronales

Podemos clasificarlas de acuerdo a dos criterios, la arquitectura y su método de aprendizaje.

Según el método de aprendizaje podemos encontrar redes de aprendizaje supervisado en el cual se debe proporcionar de manera manual los cálculos para los

patrones y las salidas deseadas. Por el otro lado están las redes de aprendizaje no supervisado, donde el desconoce el resultado y se basa en las observaciones de características, regularidades, correlaciones y categorías de los parámetros de entrada, sin verse influenciado por información exterior.

También existen las redes de aprendizaje reforzado, donde los resultados positivos son premiados y los errores son desechados.

Como se mencionó, también podemos clasificar las redes neuronales por su arquitectura, donde depende del número de capas, su organización y la disposición de las neuronas.

- Redes monocapa: la cuales podemos utilizar para clasificar patrones linealmente separables ya que únicamente cuenta con una capa.
- Redes multicapa, contiene capas ocultas entre la capa de entrada y de salida, esta es capaz de procesar información más a detalle y eliminar información irrelevante.
- Redes convolucionales, cuenta con diferentes capas y cada una de ellas esta entredas para realizar una tarea diferente.
- Redes recurrentes, este tipo de red cuenta con lazos de retroalimentación entre neuronas.

Entrenamiento

Machine Learning es el campo de estudio que por medio de la programación se les da la habilidad de aprender a las computadoras sin necesidad de un programa explícito. Tomar la experiencia E respecto a la tarea T y algún parámetro P , así su desempeño en T , medido por P , mejora con la experiencia E . (Géron, 2017) El aprendizaje es el proceso por el cual una red neuronal modifica sus pesos en respuesta a una información de entrada, de esta manera se puede afirmar que el proceso de aprendizaje ha terminado cuando los valores de los pesos permanecen estables. (Chechik, 1999; Pfeiffer, 2010)

- Aprendizaje supervisado.
En este el supervisor controla la salida de la red y en caso de que esta no coincida con la deseada, modificara manualmente los pesos de las conexiones.
 1. Corrección de error. Consiste en ajustar los pesos de las conexiones de la red en función de la diferencia entre los valores deseados y los resultados.
 2. Aprendizaje por refuerzo. Consiste en indicar mediante una señal de refuerzo si la salida obtenida es lo que se quiere, y en función de ellos se ajustan los pesos basándose en probabilidades.

3. Aprendizaje estocástico. Se realizan cambios aleatorios en los pesos y evaluar su efecto.

- Aprendizaje no supervisado.

No requieren influencia externa para ajustar los pesos, estas redes deben encontrar las correlaciones que se pueden establecer entre los datos que entran. En algunos casos la salida representa el grado de similitud entre la información que entra y la información que se le ha mostrado hasta entonces,

Este aprendizaje puede ser:

1. Hebbiano. Se pretende medir la familiaridad de los datos de entrada, las neuronas solo pueden tomar valores bilaterales (0,1) o (-1,1)
2. Competitivo o comparativo. Se orienta a la clasificación de los datos de entrada, determinando los patrones nuevos a partir de una clase reconocida previamente.

Aplicación de redes neuronales en la reconstrucción y análisis de redes de regulación genética

Las redes neuronales han demostrado ser una herramienta poderosa en la reconstrucción y análisis de redes de regulación genética. Su capacidad para modelar relaciones no lineales y realizar predicciones a partir de datos observados las hace especialmente útiles en este contexto. Algunas de las aplicaciones incluyen la identificación de interacciones entre genes, la predicción de la expresión génica y la reconstrucción de redes de regulación a partir de datos experimentales. (Peña Jiménez, 2016)

- Inferencia de interacciones entre genes: Las redes neuronales pueden utilizarse para inferir las interacciones entre genes a partir de datos de expresión génica u otros tipos de datos moleculares. Esto puede ayudar a identificar las relaciones causales entre los genes y comprender cómo se regulan los procesos biológicos.
- Predicción de la expresión génica: Las redes neuronales pueden utilizarse para predecir la expresión génica en diferentes condiciones o en diferentes tipos celulares. Esto puede ser útil para identificar genes que están regulados de manera similar y para comprender cómo se coordinan los procesos biológicos.
- Identificación de biomarcadores: Las redes neuronales pueden utilizarse para identificar biomarcadores que estén asociados con enfermedades o procesos biológicos específicos. Esto puede ser útil para el diagnóstico y tratamiento de enfermedades y para la identificación de nuevos objetivos terapéuticos.

- Análisis de redes de regulación genética: Las redes neuronales pueden utilizarse para analizar la estructura y dinámica de las redes de regulación genética. Esto puede ayudar a identificar los nodos críticos en la red y comprender cómo se propagan las señales a través de la red.

2

Metodologia

Teniendo como objetivo el resolver el problema inverso del toggle switch en redes de regulación génica, se abordó de manera secuencial dos aspectos, primero el problema directo, donde se discutieron las dificultades y limitaciones asociadas con los métodos tradicionales. A continuación el problema inverso, donde se destacaron los desafíos que presentan los métodos convencionales para abordar este problema y se justificará la necesidad de adoptar enfoques avanzados, como el uso de redes neuronales, para obtener soluciones más precisas y robustas.

Ya existen métodos tradicionales para resolver sistemas dinámicos, como:

- **Métodos analíticos:** Los cuales utilizan técnicas matemáticas para encontrar soluciones exactas a las ecuaciones diferenciales que describen el sistema. Ejemplos incluyen métodos de separación de variables, series de Fourier, transformada de Laplace, regularización de Tikhonov. Las limitaciones para este método para el caso del toggle switch radica en que solo es aplicable a sistemas simples o lineales.
- **Métodos numéricos:** Los métodos numéricos se componen de un número de pasos finitos que se ejecutan de manera lógica, mejorando aproximaciones iniciales a cierta cantidad. Métodos como el de Euler, Runge-Kutta, Newton-Raphson o Monte Carlo, requieren discretizar el tiempo, pueden ser computacionalmente costosos y sensibles a condiciones iniciales y pueden ser inestables.
- **Métodos de simulación:** Simulaciones Monte Carlo o basadas en agentes, simulaciones inversas o métodos bayesianos son utilizadas en sistemas complejos donde el comportamiento global emerge de la interacción de agentes individuales, sin embargo requieren una gran cantidad de simulaciones y son difíciles de validar y verificar.

Las redes neuronales representan una opción prometedora para resolver sistemas

dinámicos debido a su capacidad de aprender patrones temporales y secuenciales en los datos, lo cual es crucial en el problema inverso, donde una observación puede surgir de varios conjuntos de parámetros. Además, una red neuronal puede entrenarse con valores biológicamente aceptables y producir resultados también biológicamente aceptables, mientras que los métodos tradicionales podrían generar valores no válidos biológicamente. Las redes neuronales tienen el potencial de manejar sistemas dinámicos en múltiples dimensiones de manera eficiente. Del otro lado las limitaciones de las redes neuronales radican en la necesidad de grandes cantidades de datos de entrenamiento para aprender la relación, existe dificultad para interpretar cómo la red ha aprendido la relación entre las variables, podría existir un sobreajuste si no se maneja adecuadamente el entrenamiento si olvidar el consumo de recursos computacionales que implica entrenar una red neuronal.

De esta manera, concluimos que los los métodos clásicos para resolver problemas inversos son robustos y precisos para sistemas bien definidos y menos complejos, mientras que las redes neuronales ofrecen una potente herramienta para sistemas complejos y no lineales.

Una vez indentificadas correctamente los pros y los contras de los métodos tradicionales contra el uso de redes neuronales, procedimos a definir nuestro objetivo particular que consiste en modelar una red neuronal que pueda resolver el problema inverso.

Como ya se explico la metodología la dividimos en dos etapas, el problema directo y el problema inverso esto con la idea de que si era posible encontrar la función (red neuronal) que pudiera resolver el problema directo seria mas sencillo encontrar la funcion inversa para el caso del problema inverso.

El problema directo consistió en la búsqueda de puntos estacionarios a partir de los parámetros del sistema, mientras que el problema inverso implicó encontrar los parámetros a partir de los puntos estacionarios.

Para gestionar las correcciones del modelo de la red neuronal en ambas fases, se estableció un repositorio en GitHub y se utilizó Google Colab como entorno de programación y entrenamiento de la red neuronal (Repositorio Github, Salazar 2023, Jupyter Notebook, Salazar 2023).

En primer lugar, se importaron las bibliotecas necesarias para llevar a cabo una serie de operaciones, tales como la generación de datos aleatorios, la resolución de sistemas de ecuaciones no lineales, la manipulación de archivos CSV y el preprocesamiento de datos. Las bibliotecas utilizadas incluyeron: csv, os, numpy, pandas, matplotlib, scipy.optimize y sklearn.

Tomando el formalismo de Hill que vimos en nuestro marco teórico definimos nuestro sistema de ecuaciones diferenciales no lineales del modelo toggle switch para redes de regulacion génica. Este modelo describe un sistema biestable que presenta dos estados estacionarios distintos:

Listing 2.1: Sistema de ecuaciones Toggle Switch

```

def equations(x, beta, n, K1, K2, gamma1, gamma2,
             alfa):
    u, v = x
    du_dt = (beta / (1 + (v/K1) ** n)) - gamma1*u
    dv_dt = (alfa / (1 + (u/K2) ** n)) - gamma2*v
    return (du_dt, dv_dt)

```

2.1. Generación de datos

Dado que no se disponía de datos experimentales para el entrenamiento de la red neuronal, se optó por generar artificialmente parámetros para el sistema de ecuaciones no lineales. Se estableció la cantidad de datos a generar, almacenándola en la variable `num_iterations`. Dado el gran volumen de datos con el que se trabajaba, se implementó un método de escritura por bloques para optimizar el proceso.

Este método consiste en dividir los datos generados en partes más pequeñas, conocidas como *chunks* o bloques, antes de escribirlos en un archivo CSV. Esta técnica permite una manipulación más eficiente de los datos, facilitando su procesamiento y almacenamiento.

Los parámetros que se buscan generar son K , β , γ y α , los cuales poseen significados específicos y representan propiedades físicas fundamentales del sistema en estudio.

Los valores específicos de los parámetros K , β , γ y α en un modelo de "toggle switch" para redes de regulación génica pueden variar considerablemente según el sistema biológico que se esté modelando y el contexto de la aplicación. Estos parámetros dependen de las propiedades de los componentes específicos de la red génica y las interacciones entre ellos.

- K (Constante de saturación): Las constantes de saturación, K , suelen ser valores que reflejan la concentración crítica de una molécula reguladora para activar o inhibir la expresión génica. Estos valores pueden variar ampliamente según el sistema, pero en algunos casos, podrían estar en el rango de 0.1 a 10 unidades de concentración, dependiendo de la escala y las unidades específicas utilizadas en el modelo.
- β (Tasa de producción): Las tasas de producción, β , representan la velocidad a la que se producen las proteínas. Estas tasas pueden variar significativamente según el gen y las condiciones del entorno. Las tasas de producción pueden variar desde valores muy bajos, como 0.01 unidades por unidad de tiempo, hasta valores más altos, como 10 unidades por unidad de tiempo.
- γ (Tasa de degradación): Las tasas de degradación, γ , indican la velocidad a la que las proteínas se degradan o eliminan del sistema. Al

igual que con beta, estas tasas pueden variar ampliamente, pero podrían estar en el rango de 0.01 a 1 unidad por unidad de tiempo.

- α (Tasa de producción): Al igual que beta, alfa representa la tasa de producción de proteínas en el sistema. Sus valores también pueden variar según el contexto específico, pero estarían en el rango de 0.01 a 10 unidades por unidad de tiempo.
- N (Coeficiente de Hill): Representa la cooperatividad en la retroalimentación positiva, Puede ser igual o mayor que 1. un valor de n mayor que 1 indica cooperatividad positiva, lo que significa que la unión de una molécula a un sitio aumenta la afinidad de otras moléculas para unirse a ese sitio. Por otro lado, un valor de n igual a 1 indica no cooperatividad o ausencia de cooperatividad.

Es importante destacar que los rangos proporcionados para los parámetros son generales, y los valores precisos de estos parámetros pueden variar significativamente según la biología y las características específicas del sistema que se está modelando. Por lo tanto, se optó por generar valores dentro del rango de 0 a 1 para estos parámetros, para el caso del coeficiente de hill se consideraron los valores 2, 3 y 4 para asegurar cooperatividad. Esta elección permite una exploración amplia de las posibles configuraciones del sistema, sin imponer restricciones arbitrarias que podrían limitar su representación adecuada. Además, al trabajar con valores normalizados en este intervalo, se facilita la comparación y la interpretación de los resultados obtenidos durante el proceso de modelado y entrenamiento de la red neuronal.

Una vez que se generaron listas de valores aleatorios para cada uno de los parámetros K , β , γ y α los cuales se encuentran respectivamente en las listas K_values , $beta_values$, $gamma_values$ y $alfa_values$. Se procedió a eliminar cualquier valor duplicado de estas listas, manteniendo el orden de aparición de los valores únicos. Esto se logra mediante el uso de la función `np.unique()`. Después de esta operación, las listas no contienen valores duplicados, aunque podrían tener menos valores de los necesarios para alcanzar `num_iterations`.

Posteriormente, se verifica si la longitud de cada lista de valores es menor que `num_iterations`. Esto indica que no se generaron suficientes valores únicos en la primera fase. En caso de no haber suficientes valores únicos, se generan valores adicionales aleatorios para completar el espacio restante.

Luego, los nuevos valores generados se concatenan con las listas originales para asegurar que tengan al menos `num_iterations` valores únicos en total.

Listing 2.2: Ejemplo de generación aleatoria y eliminación de duplicados

```
K1_values = np.random.uniform(0, 1,
                               num_iterations)
K1_values = np.unique(K1_values)
while len(K1_values) < num_iterations:
```

```

additional_samples = num_iterations - len
    (K1_values)
new_K1_values = np.random.uniform(0, 1,
    additional_samples)
K1_values = np.concatenate((K1_values,
    new_K1_values))
K1_values = K1_values[:num_iterations]

```

2.2. Solución al sistema de ecuaciones

Una vez que generamos los valores para los parametros, se procedio a resolver el sistema de ecuaciones proponiendo 4 puntos lo suficientemente alejados para que apoyandonos del metodo fsolve este pueda hacer una aproximación a los puntos estacionarios. El método fsolve es una función en la biblioteca SciPy de Python que se utiliza para resolver sistemas de ecuaciones no lineales. Funciona mediante la búsqueda de raíces de una función dada, lo que significa encontrar los valores de las variables de entrada que hacen que la función sea igual a cero. (The SciPy community, 2008)

El resultado de fsolve es una lista de valores que representan la solución del sistema de ecuaciones. Estos valores son las raíces de la función y hacen que las ecuaciones sean aproximadamente igual a cero. En el código, los valores resultantes se asignan a las variables u y v.

Aunque el método fsolve ofrece ventajas significativas que pueden ser de gran utilidad para abordar nuestro problema, también es crucial ser conscientes de las posibles limitaciones que presenta. Entre estas desventajas se incluye su dependencia de condiciones iniciales adecuadas, ya que valores iniciales inapropiados pueden conducir a falta de convergencia o soluciones incorrectas. Además, la convergencia no está garantizada en sistemas complejos o mal condicionados, lo que puede requerir ajustes o el uso de técnicas avanzadas. Otra consideración importante es su eficiencia computacional, especialmente para sistemas grandes o altamente no lineales, donde los tiempos de cálculo pueden ser considerablemente prolongados. Por último, aunque fsolve selecciona automáticamente un método de solución, la elección del método puede influir en la velocidad y precisión de la solución, a veces requiriendo especificaciones manuales. Por lo tanto, mientras aprovechamos las ventajas que fsolve ofrece, es esencial abordar estas desventajas con precaución y considerar estrategias alternativas cuando sea necesario.

Listing 2.3: Solución del sistema de ecuaciones

```

for n in n_values:
    # Generar y dividir datos en chunks
    for i in range(num_iterations):
        K1 = K1_values[i]

```

```

K2 = K2_values[i]
gamma1 = gamma1_values[i]
gamma2 = gamma2_values[i]
alfa = alfa_values[i]
beta = beta_values[i]

# Resolver el sistema de ecuaciones para
# obtener u y v
initial_guesses = [(0.0, 0.0), (10.0,
0.0), (0.0, 10.0), (10.0, 10.0)]
solutions = []
for initial_guess in initial_guesses:
    solution = tuple(fsolve(equations,
        initial_guess, args=(beta, n, K1, K2
        , gamma1, gamma2, alfa)))
    solutions.append(solution)

```

Dado que el método `fsolve` proporciona solo una aproximación a los puntos estacionarios, se consideró la posibilidad de que las soluciones arrojadas estuvieran demasiado cercanas entre sí. Por lo tanto, se propuso una aproximación adicional a estas soluciones, teniendo en cuenta la distancia entre ellas. Esta aproximación adicional permitiría identificar soluciones que, aunque cercanas en términos de valores numéricos, pueden representar estados significativamente diferentes del sistema.

Se definieron tres funciones las cuales calculan la distancia entre puntos, agrupan puntos cercanos según un umbral de distancia y obtienen los puntos medios de estos grupos, los cuales consideramos como los puntos estacionarios definitivos.

Después se asignó un identificador de estabilidad el cual considerando que la derivada de las soluciones u y v son muy cercanas a cero, se calculan los eigenvalores y si estos son menores a cero se considera un punto estable de lo contrario se considera un punto inestable.

2.3. Datos de entrenamiento, validación y prueba

Una vez obtenidos los valores de los puntos estacionarios y evaluada su estabilidad, se procedió a la separación de los datos en conjuntos de entrenamiento, validación y prueba utilizando la función `train_test_split`. La proporción entre los conjuntos se configuró como un 90 % para entrenamiento, un 5 % para validación y un 5 % para prueba. Se asignaron identificadores de conjunto a cada uno de los conjuntos de datos: 0 para entrenamiento, 1 para validación y 2 para prueba. `train_test_split` es una función de la biblioteca `scikit-learn` en Python,

utilizada para dividir un conjunto de datos en dos o más conjuntos, comúnmente para propósitos de entrenamiento y evaluación de modelos de aprendizaje automático. (scikit-learn developers, 2007)
Quedando como resultado la siguiente distribución de datos considerando 100,000 iteraciones iniciales:

Cantidad de datos generados para entrenamiento: 683962

Cantidad de datos generados para validacion: 37998

Cantidad de datos generados para prueba: 37999

Cantidad de puntos estables: 419489

Cantidad de puntos inestables: 340470

2.4. Problema Directo

Para resolver el problema directo, se procedió a separar los datos en conjuntos de entrenamiento, validación y prueba. Esto agrupando los datos considerando los valores de los parametros y creando una lista que contiene todas las tuplas (uv) correspondientes a dichos valores para los parametros.

Ya que para cada grupo de datos no necesariamente existe la misma cantidad de tuplas o soluciones, se calculo la longitud máxima de la lista de tuplas para rellenar todas las listas hasta tener la misma cantidad de tuplas. Este relleno se hizo duplicando tantas veces fuera necesario el ultimo valor de su lista.

Separamos las características "X"(párametros) y etiquetas "Y"(u, v, split), donde split es el identificador de entrenamiento, validación o prueba.

Se convirtieron en arreglos numpy las listas de tuplas y se procedio a definir los conjuntos de datos dependiendo del valor de split, para los parametros X_train, X_val, X_test y para las etiquetas y_train, y_val, y_test.

Como nuestro objetivo final es modelar una red neuronal que pueda recibir 7 valores en su capa de entrada y devuelva 4 elementos que tenga dos componentes cada elemento, en su capa de salida, necesitamos jugar con la forma de los vectores de los datos. Buscamos convertir las listas en arreglos de la forma (n,4,2) para los valores de salida y (n,7) para los valores de entrada.

2.4.1. Definir y compilar el modelo

Definimos la arquitectura del modelo de aprendizaje automático utilizando TensorFlow y Keras. El modelo consistió en una red neuronal con una arquitectura básica de tipo feedforward, con una capa de entrada densa de 20 neuronas que espera una entrada de n_inputs igual a 7 parametros, utiliza una función de activación ReLU, los pesos se inicializan de manera uniforme, tiene una capa intermedia de 64 neuronas, una capa de salida de 2*n_outputs neuronas y una

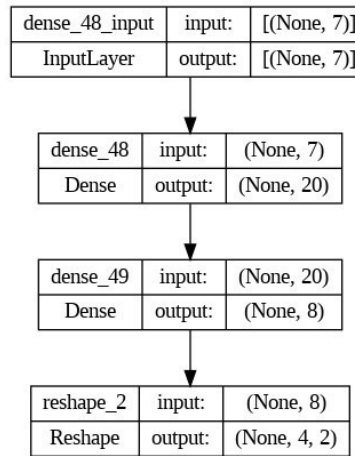


Figura 2.1: Arquitectura de la Red Neuronal

capa de remodelado la cual transforma la capa de salida en un tensor de (4,2). Además de que se utiliza una función de pérdida de error absoluto y el optimizador Adam, que es un algoritmo popular de optimización basado en gradiente estocástico.

Listing 2.4: Modelo de red neuronal

```
# Definir el modelo
def get_model(n_inputs, n_outputs):
    model = Sequential()
    model.add(Dense(20, input_dim=
        n_inputs, kernel_initializer='
        he_uniform', activation='relu'))
    model.add(Dense(2*n_outputs,
        kernel_initializer='he_uniform')
    )
    model.add(Reshape((4, 2)))
    model.compile(loss='mae', optimizer
        ='adam')
    return model
```

Se configura un callback llamado `checkpoint_callback` para guardar el mejor modelo durante el entrenamiento basado en la métrica de pérdida en el conjunto de validación. Luego, se entrena el modelo utilizando los datos de entrenamiento y validación, y la historia del entrenamiento se almacena en la variable `history`. De esta manera si se llega a interrumpir el entrenamiento del modelo por alguna situación externa se podría retomar el entrenamiento desde este llamado `checkpoint`.

Listing 2.5: Sistema de checkpoint

```
# Definir el callback para guardar el modelo
checkpoint_callback = ModelCheckpoint (
    file_path=os.path.join(
        google_drive_folder , '
        model_checkpoint.h5' ),
    # Ruta donde se guarda el modelo
    monitor='val_loss',
    # Metrica para controlar
    save_best_only=True,
    # Guardar solo el mejor modelo
    save_weights_only=False,
    # Guardar el modelo completo
```

Entrenamos el modelo durante 50 épocas con un tamaño de batch de 32. Cuando se termina de entrenar el modelo se hace una evaluación con los datos de validación y se calcula la pérdida. Después se realizaron predicciones usando los datos de prueba, se almacenaron los resultados en una variable llamada predictions y se guarda la configuración de modelo incluyendo detalles sobre las capas y el optimizador, en un archivo h5. Se registran métricas de evaluación en un archivo de texto llamado 'evaluation_metrics.txt'. Luego, se visualizan las predicciones del modelo mediante gráficos de dispersión que comparan los valores reales y las predicciones tanto para 'u' como para 'v'. Y finalmente se realizaron varios ajustes al modelo modificando la cantidad de capas intermedias, las funciones de activación, aumentando la cantidad de neuronas en cada capa, etc.

2.5. Problema inverso

Para el problema inverso se utiliza una metodología similar a la que se siguió para modelar la red para el problema directo.

En este caso cuando se hace la agrupación de los datos y la separación de los conjuntos. Se realiza agrupando los valores de (u, v, split), en lugar de los parametros como en el problema directo.

Para este caso necesitaríamos que nuestros arreglos de parametros fueran de la forma (n,7,2), donde el valor del 2 se obtiene de calcular la longitud máxima y mínima de las listas de parametros. De la misma manera se podemos observar que la forma de los arreglos de los puntos estacionarios es (n,2).

2.5.1. Definir y compilar el modelo

Usando una arquitectura similar al problema directo se define un modelo cambiando el tamaño de entrada y de salida, manteniendo la base igual.

Listing 2.6: Modelo de red neuronal problema inverso

```
# Definir el modelo
def get_model(n_inputs, n_outputs):
    model = Sequential()
    model.add(Dense(20, input_dim=
        n_inputs, kernel_initializer='
        he_uniform', activation='relu'))
    model.add(Dense(2*n_outputs,
        kernel_initializer='he_uniform')
    )
    model.add(Reshape((2,7)))
    model.compile(loss='mae', optimizer
        ='adam')
    return model
```

De igual manera inicialmente se entrena por 50 épocas y un tamaño de batch de 32. Y se realiza el mismo proceso de evaluación y predicción.

3

Análisis de resultados

3.1. Generación de datos

Usando la librería matplotlib, podemos visualizar los resultados de resolver las ecuaciones diferenciales no lineales por el método fsolve y realizando la aproximación extra que propusimos. El resultado nos arroja las siguientes gráficas:

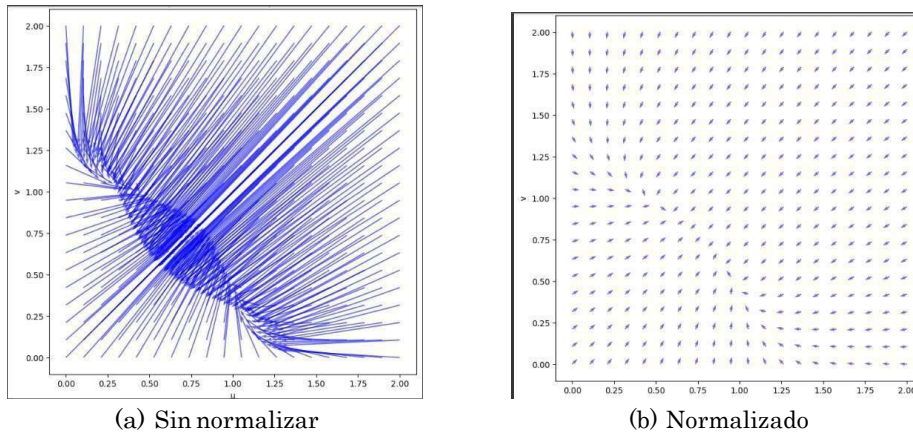


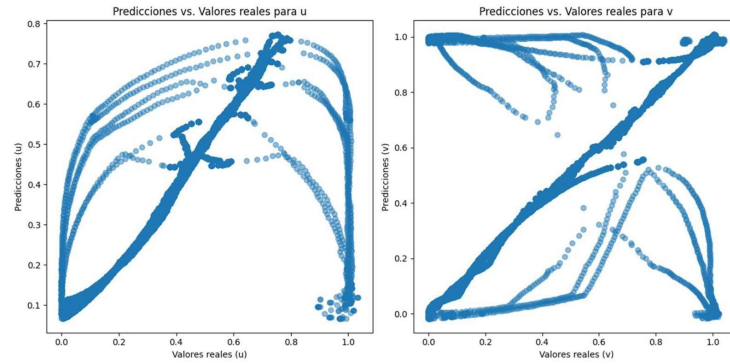
Figura 3.1: Solución al modelo toggle switch por fsolve

En el marco teórico, hemos examinado el análisis de bifurcación realizado por Miriam León en 2017. Al comparar los resultados obtenidos en nuestra investigación con la Figura 1.3 (B) presentada por Miriam, observamos similitudes significativas en el diagrama vectorial donde se reflejan la presencia de dos puntos estacionarios estables y uno inestable en el centro, lo cual concuerda con la teoría del modelo de toggle switch.

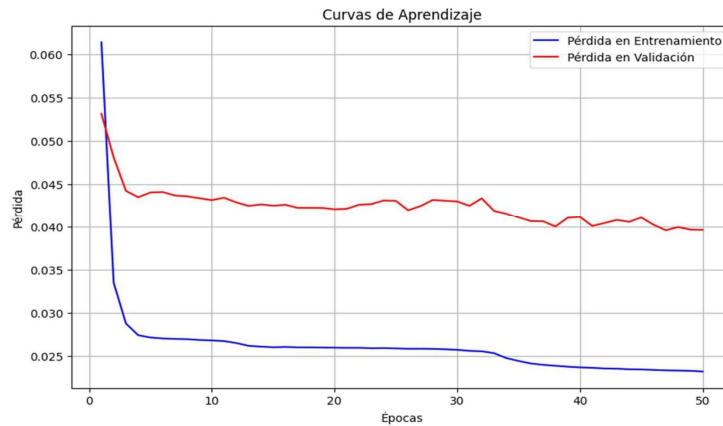
3.2. Problema directo

Para el problema directo, de igual manera haciendo uso de la libreria matplotlib, vamos a visualizar dos graficos, el primero corresponde a una comparación entre predicciones y valores reales para u y v . En la segunda grafica podemos visualizar las curvas de aprendizaje.

Se realizarón 8 variaciones y un modelo base. Para el modelo base podemos observar los siguientes resultados:



(a) Predicciones vs valores reales



(b) Curvas de aprendizaje

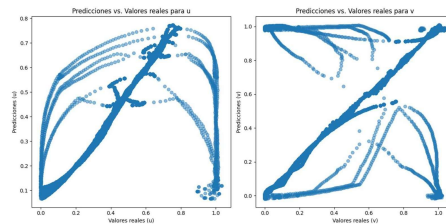
Figura 3.2: Red neuronal problema directo modelo base

Las variaciones con las que se probaron son las siguientes:

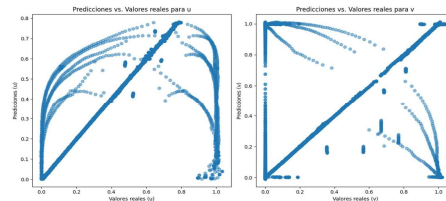
- Modelo base: Capa de entrada de 20 neuronas, utiliza una funcion de activación ReLU, los pesos se inicializan de manera uniforme, tiene una capa intermedia de 64 neuronas.
- Variación 1: Se agrego una capa intermedia con 64 neuronas, inicialización

uniforme y función de activación relu.

- Variación 2: Usando la variación 1 se agregó una capa intermedia extra con 32 neuronas, inicialización uniforme y función de activación relu.
- Variación 3: Se utilizó la arquitectura del modelo original y cambio la función de activación de relu a una sigmoide.
- Variación 4: Tomando la variación 1 se cambió la función de activación de todas las capas de relu a sigmoide.
- Variación 5: Tomando la variación 2 se cambió la función de activación de todas las capas de relu a sigmoide.
- Variación 6: A la variación 5 se le modifco la última capa intermedia pasando de 32 a 64 neuronas modificando la función de activación de la segunda capa intermedia de sigmoide a tanh y de la tercera capa intermedia de sigmoide a linear.
- Variación 7: Tomando la variación 6 se le agrego una capa intermedia extra con 32 neuronas, inicialización uniforme y función de activación linear. Con dos capas intermedias con función de activación relu.
- Variación 8: Se modifco la función de activación en la Variación 7 de una capa intermedia de tanh a sigmoide y se entrenó durante 100 épocas.



(a) Modelo Base



(b) Variación 8

Figura 3.3: Comparación predicción entre el modelo base y la variación con el mejor desempeño

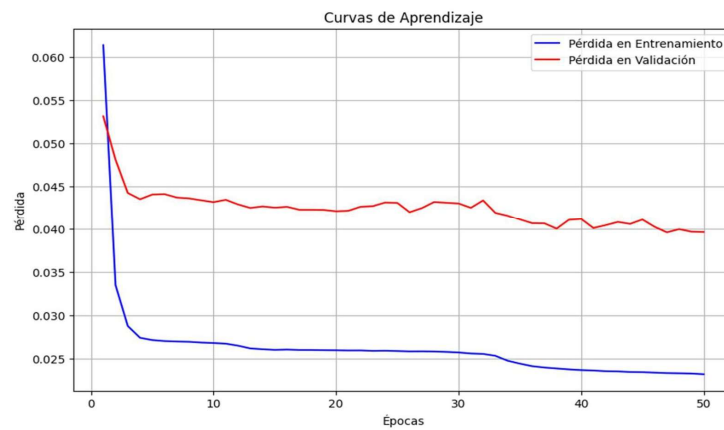
Comparando los resultados entre el modelo base y la variación más efectiva podemos concluir diversas cosas:

1. La adición de una capa intermedia extra con 32 neuronas permitió que la red neuronal aprendiera representaciones más complejas de los datos al proporcionar una mayor capacidad de modelado.
2. Al cambiar la función de activación de todas las capas a sigmoide, se observaron implicaciones importantes. Si bien la función de activación sigmoide es útil en problemas de clasificación binaria, su uso exclusivo puede limitar la capacidad de la red para aprender representaciones complejas debido al problema del desvanecimiento del gradiente. Esto destaca la importancia de elegir funciones de activación apropiadas según el contexto del problema.
3. Modificar la última capa intermedia y cambiar las funciones de activación resultó en una mejora significativa en la capacidad de capturar relaciones complejas en los datos. Al aumentar el número de neuronas y diversificar las funciones de activación, la red pudo capturar una gama más amplia de patrones y variabilidades en los datos.
4. La adición de otra capa intermedia extra con función de activación lineal demostró ser una estrategia efectiva para capturar relaciones no lineales en los datos. La función de activación lineal permitió a la red modelar relaciones lineales entre las características, lo que resultó beneficioso en la representación de ciertos tipos de datos.
5. Finalmente, la modificación de la función de activación a sigmoide y el aumento en el número de épocas de entrenamiento a 100 proporcionaron una perspectiva adicional sobre la adaptabilidad de la red. Este ajuste permitió a la red explorar más exhaustivamente el espacio de parámetros y aprender patrones más complejos en los datos, lo que condujo a una mejora en la eficacia del modelo.

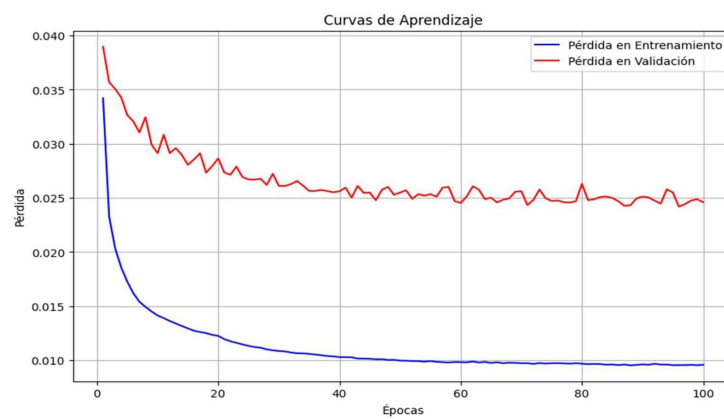
Al analizar detenidamente el proceso llevado a cabo y examinar las gráficas de los resultados, se observan ramificaciones fuera de la diagonal que representan los valores de predicción iguales a los valores reales. Se plantea la hipótesis de que estas ramificaciones podrían ser el resultado de que no todos los parámetros comparten la misma cantidad de puntos estacionarios. El ruido visible en las gráficas podría surgir del hecho de duplicar el último valor en las listas de tuplas hasta alcanzar el valor máximo, que en este caso es 4.

Para corroborar esta hipótesis, se sugiere la separación de los datos que comparten la misma cantidad de tuplas para los parámetros. Este enfoque podría proporcionar una comprensión más precisa de la relación entre los puntos estacionarios y las predicciones de la red neuronal.

Además, se consideró la posibilidad de modelar una red neuronal cuya capa de salida se ajustara dinámicamente en función de los valores de entrada. Este enfoque podría ofrecer una mayor flexibilidad y adaptabilidad del modelo a la complejidad y variabilidad de los datos.



(a) Modelo Base



(b) Variación 8

Figura 3.4: Comparación curvas de aprendizaje entre el modelo base y la variación con el mejor desempeño

3.3. Problema inverso

Para el caso del problema inverso, el objetivo de la red neuronal era encontrar los conjuntos de parametros, se propuso una arquitectura similar al modelo base del problema directo.

El modelo de red neuronal para el problema inverso es un modelo secuencial con una capa de entrada densamente conectada de 20 neuronas con función de activación ReLU, seguida por una capa oculta densamente conectada con el doble de neuronas que la capa de salida y sin función de activación especificada. Finalmente, hay una capa de salida que remodela los datos en una forma específica. El modelo se compila utilizando la función de pérdida de error medio absoluto (MAE) y el optimizador Adam.

Usando matplotlib podemos visualizar los resultados y sacar algunas conclusiones de estos:

1. Podemos observar que el modelo base funciona bastante bien para resolver el problema inverso, no hubo necesidad de tratar con mas variaciones ya que la perdida es minima, podria considerarse muy acertada.
2. La predicción de n no sigue una tendencia diagonal clara; en su lugar, observamos que para cada valor de n , la predicción varía verticalmente. Esto podría atribuirse a que las variaciones para valores donde $n < 2$ no son significativas, lo que implica que existen puntos estacionarios provenientes de los mismos conjuntos de parámetros.
3. La facilidad con la que el modelo base funciona muy bien radicara en que para los puntos estacionarios en nuestro data frame, unicamente corresponden dos conjuntos de parametros.
4. Por los resultados obtenidos podemos concluir que una red neuronal de tipo feedforward es una herramienta muy funcional para poder resolver el problema inverso para el modelo toggle switch en redes de regulación génica.

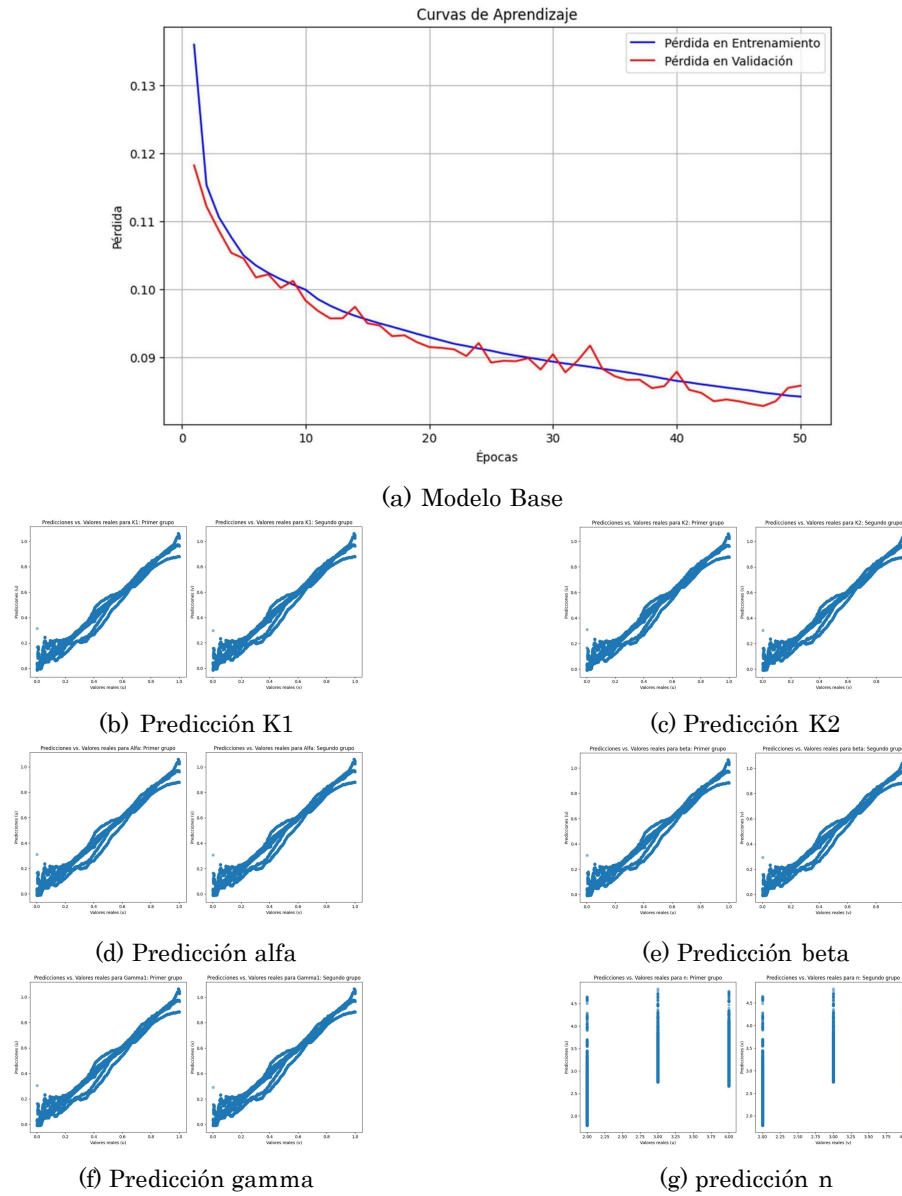


Figura 3.5: Resultados problema inverso

3.4. Conclusiones

Basándonos en la hipótesis de que una red neuronal es capaz de resolver el problema inverso del toggle switch, proponemos que este enfoque puede enfrentar y superar las diversas dificultades inherentes a este problema, tales como:

- El toggle switch es un sistema dinámico no lineal.
- La no trivialidad del problema ya que una observación puede surgir de múltiples conjuntos de parámetros.
- Dificultad de solución en múltiples dimensiones.
- Variación en los resultados dependiendo de las condiciones biológicas que se buscan.
- Sensibilidad a las condiciones iniciales.

Basados en los resultados observados durante la generación de datos, el modelado de la red neuronal, el entrenamiento y la obtención de resultados, podemos concluir que las redes neuronales tienen un potencial significativo para resolver problemas inversos en sistemas dinámicos no lineales.

Aunque en este estudio solo se trabajó con una dimensión, este potencial no se ve limitado debido a la versatilidad con la que las redes neuronales pueden ser aplicadas. Ajustando los valores de los parámetros, este enfoque podría ser extrapolado y aplicado en diferentes contextos y dimensiones, demostrando así su capacidad para abordar problemas complejos de modelado y predicción en biología y otros campos científicos.

Bibliografía

- [1] Abraham, T. H. (2002). (Physio)logical circuits: The intellectual origins of the McCulloch–Pitts neural networks. *The History of the Behavioral Sciences*, 3-25.
- [2] Alberts, B. J. (2002). *Molecular Biology of the Cell*. Garland Science.
- [3] Alon, U. (2007). Network motifs: theory and experimental approaches. *Nature Reviews Genetics*, 450-4361.
- [4] Anderson, J. A. (1995). *An Introduction to Neural Networks*. The MIT Press eBooks.
- [5] Barreto, A. da M. S., & Anderson, C. W. (2008). Restricted gradient-descent algorithm for value-function approximation in reinforcement learning. *Artificial Intelligence*, 454-482.
- [6] Angeli, D. S. (2004). Interconnections of Monotone Systems with Steady-State Characteristics. *Lecture Notes in Control and Information Science*, 135-154. doi:10.1007/978-3-540-39983-4_9
- [7] Anzurez M. Juan, P. G. (2008). Estabilidad de Sistemas No-lineales: Sistema de Nivel de Líquidos de Dos Tanques Interconectados. *RIEE&C*, 6-12.
- [8] Avery, O. T. (1944). Studies on the Chemical Nature of the Substance Inducing Transformation of Pneumococcal Types: Induction of Transformation by a Desoxyribonucleic Acid Fraction Isolated from Pneumococcus Type III. *Journal of Experimental Medicine*, 137-158.
- [9] Baldi, P. (1995). Gradient descent learning algorithm overview: a general dynamical systems perspective. *IEEE*, 182-195.
- [10] Beadle, G. W. (1941). Genetic Control of Biochemical Reactions in Neurospora. *Proceedings of the National Academy of Sciences*, 499-506.
- [11] Berg, J. T. (2015). *Biochemistry*. New York, NY: W.H. Freeman & Company.

- [12] Breitbart, R., & [et al.]. (1997). Transcriptional regulatory networks in *Saccharomyces cerevisiae*. *Science*, 53-56.
- [13] Chargaff, E. (1950). Chemical Specificity of Nucleic Acids and Mechanism of Their Enzymatic Degradation. *Experientia*, 201-209.
- [14] Chechik, G. M. (1999). Effective Learning Requires Neuronal Remodeling of Hebbian Synapses. *Neural Information Processing Systems*, 96-102.
- [15] Cooper, G. M., & Hausman, R. E. (2009). *La Célula*. Marban.
- [16] Curis, C. (1984). Some Applications of Linear Algebra. In: Linear Algebra. New York: Springer.
- [17] Darwin, C. (1859). *On the Origin of Species by Means of Natural Selection*. London: John Murray.
- [18] DeJesus, E. X. (1987). Routh-Hurwitz criterion in the examination of eigenvalues of a system of nonlinear ordinary differential equations. *Physical Review*.
- [19] Del Vecchio, D., & Murray, R. M. (1975). *Biomolecular Feedback Systems*. Oxfordshire: Princeton University Press Princeton Oxford.
- [20] Del Vecchio, D., & Qian, Y. (2018). Future systems and control research in synthetic biology. *Annual Reviews in Control*, 5-17. doi:10.1016/j.arcontrol.2018.04.007
- [21] Fee, T. M. (2011). Rudolf Carl Virchow Medical Scientist, Social Reformer, Role Model. *AJPH*, 2104-2105. doi:10.2105/AJPH.2005.078436
- [22] Fontana, W., Günter, W., & Leo W., B. (1994). Beyond Digital Naturalism. *Nature*, 23.
- [23] Franklin, R. E. (1953). Molecular Configuration in Sodium Thymonucleate. *Nature*, 740-741.
- [24] Rubin, G. M., & [et al.]. (2000). A Brief History of Drosophila's Contributions to Genome Research. *Science*, 2216-2218.
- [25] Géron, A. (2017). *Hands-On Machine Learning with Scikit-Learn and TensorFlow*. O'Reilly Media, Inc.
- [26] Grebogi, C. (1987). Chaos, Strange Attractors, and Fractal Basin Boundaries in Nonlinear Dynamics. *Science*, 632-638.
- [27] Griffith, F. (1928). The Significance of Pneumococcal Types. *Journal of Hygiene*, 113-159.
- [28] Gutiérrez G., A. C. (2016). Diagrama de bifurcación de sistemas lineales R^3 . *Revista de la Facultad de Ciencias*, 160-169.

- [29] Hernández Paredes, S. (2015). Ampliación de Análisis Numérico - Tema 3. Resolución de ecuaciones no lineales. Obtenido de Departamento de Matemática Aplicada y Estadística, Silvestre Paredes Hernández: <https://www.dmae.upct.es/~paredes/mn.htm>
- [30] Hernández Paredes, S. (2015). AMPLIACIÓN DE MATEMÁTICAS (GRADO EN TECNOLOGÍAS INDUSTRIALES) - Tema 3. Estabilidad de Ecuaciones Diferenciales. Obtenido de Departamento de Matemática Aplicada y Estadística, Silvestre Paredes Hernández: https://www.dmae.upct.es/~paredes/am_ti.html
- [31] Hershey, A. D. (1952). Independent Functions of Viral Protein and Nucleic Acid in Growth of Bacteriophage. *Journal of General Physiology*, 39-56.
- [32] Hill, A. (1910). The possible effects of the aggregation of the molecules of hæmoglobin on its dissociation curves. *The Journal of Physiology*, 1-7.
- [33] Ideker, T. L. (2001). A new approach to decoding life: systems biology. *Annual Review of Genomics and Human Genetics*, 343-372.
- [34] Institute., N. H. (2021). The Human Genome Project. Recuperado el 1 de Junio de 2023, de <https://www.genome.gov/panorama-general-del-proyecto-del-genoma-humano>.
- [35] Hale, J. K., & [et al.]. (1967). EXTENDED DYNAMICAL SYSTEMS AND STABILITY THEORY. *PNAS*, 405-409.
- [36] Hopfield, J. J. (1982). Neural networks and physical systems with emergent collective computational abilities. *National Academy of Sciences of the United States of America*, 2554-2558. doi:10.1073/pnas.79.8.2554
- [37] Jacob, F., & [et al.]. (1961). Genetic regulatory mechanisms in the synthesis of proteins. *Journal of Molecular Biology*, 318-356.
- [38] Jacob, F., David, P., Carmen, S., & Jacques, M. (2005). L'opéron : groupe de gènes à expression coordonnée par un opérateur [C. R. Acad. Sci. Paris 250 (1960) 1727-1729]. *Comptes Rendus Biologies*, 328, 514-520. doi:10.1016/j.crvi.2005.04.005.
- [39] Jiménez Jiménez, N. (23 de Septiembre de 2013). Ampliación de Matemáticas Ingeniería Industrial - Tema 6. Estabilidad de sistemas diferenciales. Obtenido de Nieves Jiménez Jiménez: <https://personal.us.es/niejimjim/ampmec.html>
- [40] Junqueira, L. C. (2012). *Biología Celular y Molecular*. Rio de Janeiro: GK.
- [41] Junqueira, L. C. (2012). Tecnología de la Biología Celular y Molecular: Algunos ejemplos. En L. C. Junqueira, *Biología Celular y Molecular* (págs. 24-36.). Rio de Janeiro: 9ª ed.

- [42] Kitano, H. (2002). Systems biology: a brief overview. *Science*, 295(5560), 1662-1664.
- [43] Kremling, A. (2014). *System Biology. Mathematical Modeling and Model Analysis*. Wiesbaden: CRC Press Taylor & Francis Group.
- [44] Latchman, D. (2005). *Gene Regulation*. London: Taylor & Francis.
- [45] van 't Veer, L. J., & [et al.]. (2002). Gene expression profiling predicts clinical outcome of breast cancer. *Nature*, 530-536.
- [46] Lemaréchal, C. (2010). Cauchy and the Gradient Method. *Documenta-Math.*, 251-254.
- [47] Levine, M., & [et al.]. (2003). Transcription Regulation and Animal Diversity. *Nature*, 147-151.
- [48] Lillo Egaña, D. H. (2010). INVESTIGACIÓN, MODELACIÓN Y RECONSTRUCCIÓN DE. FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS.
- [49] Lohmueller, J. N. (2007). Progress toward construction and modelling of a tri-stable toggle switch in *E. coli*. *IET Synthetic Biology*, 25-28.
- [50] Love, A. (2015). Evolutionary Developmental Biology: Philosophical Issues. *Springer eBooks*, 265-283. doi:10.1007/978-94-017-9014-7_13
- [51] Luaces Martinez, V. (210). Propuestas para la enseñanza de las matemáticas. En V. Luaces Martinez, *PROBLEMAS INVERSOS: LOS CASI OLVIDADOS DE LA MATEMÁTICA EDUCATIVA* (págs. 439-447). Comité Latinoamericano de Matemática Educativa.
- [52] Mehrotra, K. G. (1996). *Elements of Artificial Neural Networks*. The MIT Press eBooks.
- [53] Mendel, G. (1866). Experimentos sobre híbridos de plantas. *Actas de la Sociedad de Investigación Natural de Brno*.
- [54] Meselson, M., & Stahl, F. W. (1958). The replication of DNA in *Escherichia coli*. *Proceedings of the National Academy of Sciences*, 671-682. doi:10.1073/pnas.44.7.671
- [55] Miriam, L. (2017). Computational design and characterisation of synthetic genetic switches. *University College London*.
- [56] Morgan, T. H. (1910). Sex-limited inheritance in *Drosophila*. *Science*, 32(812), 120-122.
- [57] Mullis, K., & [et al.]. (1987). Specific synthesis of DNA in vitro via a polymerase-catalyzed chain reaction. *Methods in enzymology*, 335-350.

- [58] Munt, A. M. (9 de Febrero de 2018). Introducción a los modelos de redes neuronales artificiales. El Perceptrón simple y multicapa.
- [59] Nelson, D. (2017). DNA, RNA, and the Flow of Genetic Information. En D. Nelson, *Lehninger Principles of Biochemistry* (págs. 1024-1026). W.H. Freeman and Company.
- [60] Nielsen, M. (Diciembre de 2019). Neural Networks and Deep Learning. Obtenido de Neural Networks and Deep Learning: <http://neuralnetworksanddeeplearning.com/>
- [61] Wolernhauer, O., & Uhlenhof-Hoche, M. (2004). Modeling and simulation of intracellular dynamics: choosing an appropriate framework. *IEEE*, 200-207.
- [62] Orengo Ferriz, D. J. (2012). *Fundamentos de la Biología Molecular*. Barcelona: Editorial UOC.
- [63] Avery, O. T., MacLeod, C. M., & McCarty, M. (1944). STUDIES ON THE CHEMICAL NATURE OF THE SUBSTANCE INDUCING TRANSFORMATION OF PNEUMOCOCCAL TYPES: INDUCTION OF TRANSFORMATION BY A DESOXYRIBONUCLEIC ACID FRACTION ISOLATED FROM PNEUMOCOCCUS TYPE III. *J Exp Med*, 137-158. doi:10.1084/jem.79.2.137
- [64] P, S., D, Z., & H, Z. (2019). Behavior Action Mining. *IEEE Access*, 19954-19964. doi:10.1109/ACCESS.2019.2896141
- [65] Peña Jiménez, E. (2016). Modelación de redes de regulación genética con redes neuronales recurrentes desde una perspectiva bayesiana. *Universidad Autónoma de Nuevo León Facultad de Ciencias Físico Matemáticas*.
- [66] Pfeiffer, M. N. (2010). Reward-Modulated Hebbian Learning of Decision Making. *Neural Computation*, 1399-1444.
- [67] Picelli, S. (2017). Single-cell RNA-sequencing: The future of genome biology is now. *RNA Biology*, 637-650. doi:10.1080/15476286.2016.1201618
- [68] Pollack JR, I. V. (2002). Characterizing the physical genome. *Nature Genetics*, 515-521.
- [69] Ptashne, M. (2002). *Genes and Signals*. New York: Cold Spring Harbor Laboratory Press.
- [70] Milo, R., Shen-Orr, S., & Itzkovitz, S. (2002). Network Motifs: Simple Building Blocks of Complex Network. *Science*, 824-827.
- [71] Rainer, B. (2010). What is systems biology? *Frontiers in Physiology*, 1. doi:10.3389/fphys.2010.00009

- [72] Redacción KeepCoding. (17 de Agosto de 2022). ¿Qué es una función de activación en Deep Learning? Obtenido de KEEPCODING Tech School: <https://keepcoding.io/blog/funcionde-activacion-en-deep-learning/>
- [73] Britten, R. J., & Davidson, E. H. (1969). Gene Regulation for Higher Cells: A Theory. *Science*, 349-357.
- [74] Ruiz, C., & Basualdo, M. (Marzo de 2001). Facultad Regional Rosario - Catedras Informática Aplicada a la Ingeniería de Procesos – Orientación I. Obtenido de Facultad Regional Rosario: https://www.frro.utn.edu.ar/repositorio/catedras/quimica/5_anio/orientadora1/monograias/matich-redesneuronales.pdf
- [75] Salazar Montes, A. M. (2013). Regulación de la expresión genica. En A. M. Salazar Montes, *Biología Molecular: Fundamentos y aplicaciones en las ciencias de la salud* (pág. 66). México: Editorial McGraw-Hill Education.
- [76] Salazar Montes, A. M., Sandoval Rodríguez, A. S., & Armendáriz Borunda, J. S. (2013). *Biología Molecular. Fundamentos y aplicaciones en las ciencias de la salud*. Ciudad de México: McGRAW-HILL INTERAMERICANA EDITORES, S.A. de C.V.
- [77] Schena, M., Shalon, D., Davis, R. W., & Brown, P. O. (1995). Quantitative monitoring of gene expression patterns with a complementary DNA microarray. *Science*, 270(5235), 467-470. doi:10.1126/science.270.5235.467
- [78] Science History Institute Museum & Library. (28 de Julio de 2022). Education: Scientific Biographies. Obtenido de Science History Institute Web Site: <https://sciencehistory.org/education/scientific-biographies/james-watson-francis-crickmaurice-wilkins-and-rosalind-franklin/>
- [79] scikit-learn developers. (2007). *sklearn.model_selection.train_test_split*. Obtenido de scikit learn: https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.train_test_split.html
- [80] Shendure, J., & [et al.]. (2008). Next-generation DNA sequencing. *Nature biotechnology*, 1135-1145.
- [81] Sutton, W. S. (1902). The chromosomes in heredity. *Biological Bulletin*, 231-251.
- [82] Aldhyani, T. H. H., & [et al.]. (2020). Intelligent Hybrid Model to Enhance Time Series Models for Predicting Network Traffic. *IEEE Access*, 130431-130451. doi:10.1109/ACCESS.2020.3009169
- [83] The SciPy community. (2008). *scipy.optimize.fsolve*. Obtenido de SciPy v1.12.0 Manual: <https://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.fsolve.html>

- [84] Thomas, W. (2009). The smallest chemical reaction system with bistability. *BMC Systems Biology*.
- [85] Gardner, T. S., & [et al.]. (2000). Construction of a genetic toggle switch in *Escherichia coli*. *Nature*, 339-342.
- [86] Lee, T. I., & [et al.]. (2002). Transcriptional regulatory networks in *Saccharomyces cerevisiae*. *Science*, 799-804.
- [87] Trujillo Hernández, I. S., & Morales Huitzil, Y. (2021). *Metodología de la investigación*. Puebla, Puebla, México: Porrúa. Recuperado el 14 de enero de 2023.
- [88] Turner, L. R. (1960). SOLUTION OF NONLINEAR SYSTEMS. *Annals of the New York Academy of Sciences*, 817-827.
- [89] V, V., N, Z., & G.-J., Q. (2015). Differential Recurrent Neural Networks for Action Recognition. *IEEE International Conference on Computer Vision (ICCV)*, 4041-4049. doi:10.1109/ICCV.2015.460
- [90] Valverde, J. C. (2004). Stability of Continuous Systems by Routh-Hurwitz and Mathematica. *Journal of Computational Methods in Sciences and Engineering*, 125-134. doi:10.3233/jcm-2004-41-215
- [91] Varga, R. S. (2010). *Geršgorin and His Circles*. Springer Berlin Heidelberg.
- [92] Veening, J.-W., Klass Smits, W., & P. Kuipers, O. (2008). Bistability, Epigenetics, and Bet-Hedging in Bacteria. *Annual Review of Microbiology*, 193-210.
- [93] Velazquez Castro, J. (2022). Clases Redes Neuronales Otoño 2022. Obtenido de Clases Redes Neuronales Otoño 2022: <https://u.pcloud.link/publink/show?code=kZvojBVZzc2IPoO1JF8Cfy81EdmKHQI2T6V>
- [94] Vera Cruz, J. M., Salazar Montes, A. M., Guardado Mora, C., & Armendariz Borunda, J. (2013). Historia de la biología molecular. En J. M. Vera Cruz, A. M. Salazar Montes, C. Guardado Mora, & J. Armendariz Borunda, *Biología Molecular: Fundamentos y aplicaciones en las ciencias de la salud* (págs. 3-13.). México: Editorial McGraw-Hill Education.
- [95] Astbury, W., & O. (1938). Some Recent Developments in the X-Ray Study of Proteins and Related Structures. *Cold Spring Harbor Symposium on Quantitative Biology*, 266-272.
- [96] Wang, Z. L. (2015). Introduction to Neural Networks. *Qualitative Analysis and Control of Complex Neural Networks with Delays*, 1-36.
- [97] Watson, J. D. (1953). Molecular structure of nucleic acids: A structure for deoxyribose nucleic acid. *Nature*, 737-738.

- [98] Wiens, E. G. (29 de 1 de 2024). Systems of Linear Differential Equations. Obtenido de Egwald Mathematics: Linear Algebra: <https://www.egwald.ca/linearalgebra/lineardifferentialequationsstabilityanalysis.php>
- [99] Wilkinson, D. J. (2006). *Stochastic Modelling for Systems Biology*. New York: Chapman and Hall/CRC.
- [100] Salazar, B. Marco (2023). *Genetic_regulatory_system*. Obtenido de Genetic_regulatory_system.git: https://github.com/Ramkientonio/Genetic_regulatory_system.git
- [101] Salazar, B. Marco (2023). *Neural_Network*. Obtenido de Neural_Network.ipynb: <https://colab.research.google.com/drive/15u55ntAjQAU9uuCGAeDHI8xFOA9oM00x?usp=sharing>
- [102] Hirsch, M. W., Smale, S., & Devaney, R. L. (2013). *Differential Equations, Dynamical Systems, and An Introduction to Chaos*. Academic Press.