Universidad del Rosario

Una propuesta de neurona artificial: la Unidad NeuroVascular Artificial (UNVA)

Autor
Juan Camilo Ruiz Ortiz

Trabajo presentado como requisito para optar por el título de Profesional en Matemáticas Aplicadas y Ciencias de la Computación

Director, Tutor
Alexander Caicedo Dorado

Escuela de Ingeniería Ciencia y Tecnología
Matemáticas Aplicadas y Ciencias de la Computación (MACC)
Universidad del Rosario

Bogotá D.C. - Colombia
2021

«Para aquellos que no conocen las matemáticas, es difícil sentir la belleza de la naturaleza. Si quieres apreciarla, es necesario aprender el lenguaje en el que habla.»

Richard Feynman

Resumen

Las neuronas artificiales son un modelo computacional simplificado de cómo funcionan las neuronas biológicas presentes en el cerebro. Sin embargo, los modelos de las primeras neuronas artificiales se fundamentaron únicamente en el procesamiento de información proveniente de señales eléctricas, y no tuvieron en cuenta los cambios vasculares necesarios que permiten entregar nutrientes a las neuronas para que funcionen correctamente, en particular durante su activación eléctrica. Por lo tanto, en esta tesis se propone un nuevo modelo computacional que considera tanto el comportamiento eléctrico como el vascular. Para diseñar la nueva arquitectura, se revisaron las condiciones de estabilidad del descenso del gradiente. Este análisis nos permite definir cotas superiores para la tasa de aprendizaje. Una vez propuesta la arquitectura se evaluó su comportamiento comparado con algoritmos más tradicionales como la regresión lineal.

Índice general

Resumen V

Índice de figuras IX

Índice de cuadros XI

1. Introducción 1

2. Fundamentos Biológicos 5
2.1. El sistema nervioso . . . 5
2.2. La neurona . . . 6
2.3. El potencial de acción . . . 7
2.4. Acoplamiento neurovascular . . . 8
2.4.1. Resonancia mágnetica funcional . . . 8
2.4.2. Principios básicos del acoplamiento neurovascular . . . 9
2.4.3. Respiración celular . . . 10
2.4.4. La ley de Fick como modelo básico para describir el acoplamiento neurovascular . . . 11

3. Fundamentos de Neuronas Artificiales 13
3.1. Neurona de McCulloch-Pitts . . . 13
3.2. Perceptrón . . . 14
3.3. Adaline . . . 15
3.4. Neurona logística . . . 15
3.5. Reglas de aprendizaje . . . 16
3.5.1. Regla del perceptrón . . . 16
3.5.2. Descenso del gradiente . . . 17
3.5.3. Consideraciones de la función de error . . . 20
3.6. Redes Neuronales Artificiales . . . 21

4. Análisis de estabilidad del descenso del gradiente 23
4.1. Fundamentos teóricos de sistemas dinámicos . . . 23
4.2. Análisis de descenso del gradiente como sistema dinámico . . . 27
4.3. Derivación de las cotas de la tasa de aprendizaje . . . 31

5. La Unidad NeuroVascular Artificial (UNVA) 35
5.1. Algoritmo de aprendizaje de la UNVA . . . 38
5.1.1. Adaptaciones a partir de la regla del perceptrón . . . 38

5.1.2. Adaptaciones a partir del descenso del gradiente 41

6. Resultados **45**
6.1. Validación de resultados de cota de tasa de aprendizaje 45
6.1.1. Resultados con datos manuales 45
6.1.2. Resultados con datos Iris 50
6.2. Resultados de la UNVA 53
6.2.1. Resultados de UNVA con datos manuales 54
6.2.2. Resultados de UNVA para pronósticos con series de tiempo 56

7. Conclusión **61**
7.1. Resumen general 61
7.2. Trabajo futuro 62

Bibliografía **63**

Índice de figuras

2.1. Sistema nervioso 6
2.2. Esquema simplificado de una neurona 6
2.3. Evolución de potencial de acción en función del tiempo 7
2.4. Imagenes de resonancia magnética funcional desde distintos ángulos 9
2.5. Esquema de unidad neurovascular 10

3.1. Representación esquemática de la neurona de McCulloch-Pitts . . . 13
3.2. Representación esquemática del perceptrón 14
3.3. Representación esquemática de ADALINE 15
3.4. Esquema de neurona artificial generalizada 16
3.5. Ejemplos de separación lineal y no lineal 17
3.6. Evolución de función de costo en función de valor de tasa de aprendizaje 19
3.7. Esquema de red neuronal artificial 22

4.1. Evolución de modelo exponencial variando valor de constante a . . 24
4.2. Casos posibles de ubicacion de valores propios en plano complejo, de iquierda a derecha: estabilidad asintótica, inestabilidad, comportamiento oscilatorio) 27

5.1. Esquema de Unidad Neurovascular Artficial (UNVA) 37

6.1. Evolución de MSE con múltiples tasas de aprendizaje calculadas con cota $2N/\lambda_{max}$ y con datos manuales 46
6.2. Evolución de MSE con múltiples tasas de aprendizaje calculadas con cota $2N/\operatorname{tr}(X^T X)$ con datos manuales 47
6.3. Evolución de coeficientes para distintas tasas de aprendizaje η con datos manuales 48
6.4. Dispersiones entre variables de datos Iris 50
6.5. Evolución de MSE con múltiples tasas de aprendizaje para cota $2N/\lambda_{max}$ con datos Iris 51
6.6. Evolución de MSE con múltiples tasas de aprendizaje calculadas con cota $2N/\operatorname{tr}(X^T X)$ con datos Iris 52
6.7. Evolución de coeficientes para distintas tasas de aprendizaje η con datos Iris 52
6.8. Evolución de coeficientes de UNVA con $\eta = 0{,}65 \times 2N/\lambda_{max}[n]$. . 54
6.9. Evolución del flujo medio por épocas. 55
6.10. MSE de UNVA. 55

6.11. Evolución de MSE de UNVA calculado con varios multiplicadores a partir de cota $2N/\lambda_{max}$ y con datos manuales 56
6.12. Evolución de valor de apertura de Bitcoin 56
6.13. Errores de entrenamiento de regresion lineal y UNVA para series temporales 57
6.14. Evolucion de flujo (izquierda) y de la cota (derecha) en entrenamiento de UNVA para series de tiempo 58
6.15. Pronóstico de valor de apertura de Bitcoin con UNVA y con modelo autoregresivo 58

Índice de cuadros

6.1. Resultados con datos de pruebas y datos generados manualmente . . 49
6.2. Resultados con datos de pruebas y datos Iris 53

Lista de acrónimos

UNVA	Unidad NeuroVascular Artificial
ATP	Adenosine Triphosphate
Adaline	Adaptative Linear Neuron
SNC	Sistema Nervioso Central
SNP	Sistema Nervioso Periférico
fMRI	functional Magnetic Resonance Imaging
BOLD	Blood Oxygen Level Dependent
RNA	Redes Neuronales Artificiales
MSE	Mean Squared Error
CBF	Cerebral Blood Flow
UCI	Universidad de California en Irvine
SSE	Sum of Squared Errors
RNVA	Red NeuroVascular Artificial

Dedicada a mi familia, mi novia Alejandra y a los profesores que me apoyaron en todo este proceso.

Capítulo 1

Introducción

La inteligencia artificial es un área de las ciencias de la computación que estudia el desarrollo de sistemas y algoritmos que logren imitar algunas tareas desarrolladas por los humanos como aprender, adaptarse, razonar y autocorregirse [20]. La investigación en inteligencia artificial surgió y se desarrolló notablemente durante el siglo XX con trabajos como el de Alan Turing en el que se evalua la capacidad de las máquinas para pensar [37], pero también con el diseño de modelos computacionales inspirados en las neuronas; las cuales se consideran como la unidad celular básica del cerebro y permiten procesar la información que se recibe. Algunos precursores de esta línea de investigación incluyen a Warren McCulloch, Walter Pitts y Frank Rosenblatt, los cuales, inspirados en el comportamiento de una neurona biológica, propusieron arquitecturas computacionales que funcionaran como unidades neuronales artificiales de procesamiento de información.

Los modelos computacionales para unidades neuronales artificiales se fundamentan en que la información que procesan las neuronas consiste de señales provenientes de otras neuronas y de estímulos externos al cuerpo humano, los cuales al superar cierto umbral biológico producen la propagación de un potencial de acción. Sin embargo, estos modelos se basan en el comportamiento eléctrico de las neuronas, y no consideran el acoplamiento vascular, el cual regula el suministro de sustrato y oxígeno necesario para la generación energía. Esta energía generalmente se almacena en forma de adenosín trifosfato, o ATP por su sigla en inglés, y es necesaria para permitir la propagación del potencial de acción, y por ende para el procesamiento de la información en el cerebro. El control del suministro de oxígeno y nutrientes al cerebro se realiza mediante cambios localizados en el flujo/volumen de la sangre. El acoplamiento entre la actividad neuronal y su soporte vascular se conoce como acoplamiento neurovascular.

En esta tesis se busca generar un nuevo modelo de unidad neuronal artificial básica, inspirado en el acoplamiento neurovascular que ocurre naturalmente en el cerebro. Por lo tanto, los objetivos de esta tesis son:

- Formular un modelo matemático y computacional de una unidad neuronal artificial básica que considere no sólo en el procesamiento electrico de la información, sino que también integre el proceso vascular que sirve como soporte energético para las neuronas. Este modelo será llamado Unidad NeuroVascular Artificial (UNVA).

- Proponer algoritmos de aprendizaje para esta nueva estructura de modelo, usando como base el algoritmo de descenso del gradiente.
- Estudiar desde el punto de vista de sistemas dinámicos las condiciones de estabilidad del algoritmo propuesto. En particular, determinar las condiciones de la taza de aprendizaje para garantizar la convergencia del descenso del gradiente.
- Evaluar el rendimiento de un modelo de Unidad NeuroVascular Artificial en algunos ejemplos de aplicación.

Durante el desarrollo de esta tesis abordaremos cada uno de los objetivos planteados. Los capítulos de esta tesis se organizan de la siguiente manera:

- En el **capítulo 2** se describen los fundamentos biológicos acerca del sistema nervioso, el funcionamiento de las neuronas y la activación neuronal junto con el potencial de acción. Adicionalmente, se define el acoplamiento neurovascular, el cual establece una relación entre los cambios vasculares del cerebro y la actividad neuronal. Además se mencionan algunas de las herramientas que se han utilizado para estudiar la relación entre la actividad neuronal y cambios locales en el flujo de sangre. Por último, se explica los fundamentos de la respiración celular y se presenta la ley de Fick como base de algunos modelos matemáticos que se presentarán posteriormente.
- El **capítulo 3** presenta varias arquitecturas de neuronas artificiales tales como: la neurona de McCulloch - Pitts, el perceptrón, el ADALINE y la neurona logística. Además, se explican las reglas o algoritmos de aprendizaje como la regla de aprendizaje del perceptrón y el método del descenso del gradiente. Por último, se resumen algunas consideraciones acerca de los casos de uso de arquitecturas neuronales artificiales, y se introduce brevemente las redes neuronales artificiales.
- Antes de proponer la UNVA se consideró revisar las condiciones de estabilidad de los algoritmos de aprendizaje presentados en el capítulo 3. Por esta razón, en el **capítulo 4** se desarrollan primero las condiciones de estabilidad de un sistema de ecuaciones en diferencias lineal de primer orden, y seguido se analiza el descenso del gradiente desde el punto de vista de un sistema dinámico. Este análisis permite establecer bajo qué valores de tasa de aprendizaje se garantiza que el descenso del gradiente converja. Este capítulo es uno de los resultados más importantes de esta tesis.
- En el **capítulo 5** se formaliza la Unidad NeuroVascular Artificial (UNVA) y también se presentan las adaptaciones necesarias de los algoritmos de aprendizaje para entrenar esta nueva arquitectura, específicamente para el descenso del gradiente y la regla del perceptrón. Este capítulo es el centro de esta tesis.
- En el **capítulo 6** se presentan los resultados de validación sobre las condiciones de estabilidad determinadas en el capítulo 4 con varios conjuntos de datos. Se evalúa el comportamiento de una neurona regresora, equivalente a una regresión lineal, y el del modelo de neurona artificial propuesto, la UNVA.

- Por último, en el **capítulo 7** se resumen las conclusiones de la tesis, y se discute sobre posibles líneas de acción futuras para continuar con el desarrollo de este proyecto.

Capítulo 2

Fundamentos Biológicos

En este capítulo se explica los fundamentos biológicos del sistema nervioso, el cerebro y los fenómenos vasculares asociados a la actividad neuronal. Estos conceptos biológicos son importantes porque a partir de ellos se han formulado varios modelos de neuronas artificiales, y sirven como inspiración para la arquitectura propuesta en esta tesis.

El cuerpo humano está compuesto de varios sistemas que cumplen cada uno una función específica en el cuerpo, por ejemplo, el sistema circulatorio permite la circulación de la sangre en el cuerpo a través de las venas y las arterias junto con el corazón que bombea la sangre para distribuirla en el cuerpo, o el sistema digestivo encargado de absorber nutrientes y eliminar los desechos a través del tracto gastrointestinal, incluyendo la boca, el esófago, el estómago y los intestinos [43]. En particular, el sistema encargado de procesar la información capturada por los sentidos y de controlar las actividades motoras es el sistema nervioso.

2.1. El sistema nervioso

El sistema nervioso es un sistema complejo de los vertebrados cuya función principal es coordinar sus acciones y procesar la información sensorial detectada tanto internamente como externamente al cuerpo. Se compone de varios elementos como el cerebro, la médula espinal, los ganglios y las células llamadas neuronas, cuya función y estructura se explica más adelante.

Como se muestra en la Figura 2.1, el sistema nervioso está dividido en 2 partes principales:

- El sistema nervioso central (SNC), compuesto por el cerebro y la médula espinal, en el cual se llevan a cabo los procesos mentales necesarios para comprender la información recibida del exterior, y para transmitir los impulsos hacia los nervios y los músculos con el fin de dirigir sus movimientos [26].

- El sistema nervioso periférico (SNP), compuesto por partes del sistema nervioso fuera del cerebro y de la médula espinal como: ganglios y nervios, incluyendo los nervios craneales, los nervios espinales, los nervios periféricos y las uniones neuromusculares [6].

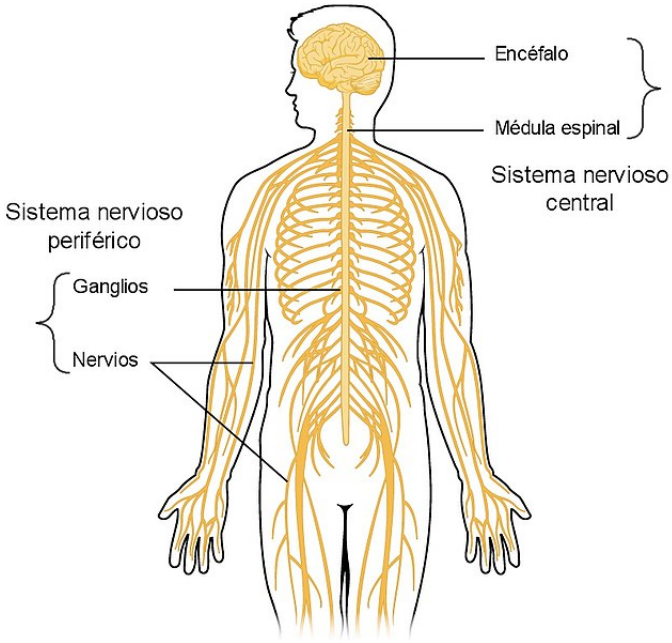
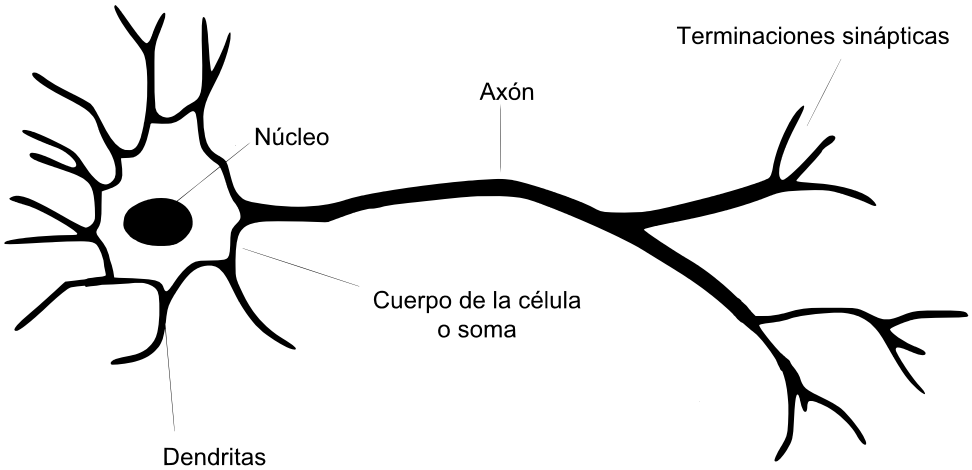


FIGURA 2.1: Sistema nervioso
Tomado de https://rea.ceibal.edu.uy/elp/los-aparatos-que-conforman-nuestro-cuerpo/sistema_nervioso.html

2.2. La neurona

El actor principal del sistema nervioso son las células llamadas neuronas, las cuales reciben estimulos del entorno, procesan la información recibida y la transmiten al sistema nervioso central desde el sistema nervioso periférico. Por medio de conexiones con otras neuronas, lo que se conoce como sinápsis [14], se transmite información de vuelta al sistema nervioso periférico [22].



Adaptado de https://upload.wikimedia.org/wikipedia/commons/thumb/4/48/Neurone_-_comment%C3%A9.svg/1200px-Neurone_-_comment%C3%A9.svg.png

FIGURA 2.2: Esquema simplificado de una neurona

Las neuronas se conectan entre ellas para transmitir mensajes que surgen de estímulos. En la figura 2.2 se muestra la composición de una neurona. Por un lado tenemos las dendritas, las cuales actúan como elementos que reciben un estímulo de otras neuronas, el soma, el cual es el cuerpo de la neurona, donde se encuentra su núcleo, y el axón, el cual es una especie de filamento por el cual se mueve el mensaje recibido por la neurona hacia sus terminaciones sinápticas. En estas terminaciones las neuronas se conectan con otras neuronas a través de lo que se conoce como sinápsis. Cuando la excitación de entrada en una neurona, recibida por las dendritas, supera cierto umbral, se dispara la propagación de un potencial de acción el cual, al llegar al terminal del axón ocasiona la liberación de neurotransmisores los cuales estimulan las dendritas de otras neuronas [1].

2.3. El potencial de acción

El potencial de acción se define como el cambio de potencial entre la membrana y el interior de la célula. Se produce por el desplazamiento de una corriente iónica que, en una neurona, al llegar al terminal del axón produce una descarga de neurotransmisores. Cuando estos neurotransmisores entran en contacto con las dendritas de otras neuronas pueden desencadenar el disparo de un potencial de acción en estas [21]. De esta forma, el potencial de acción de una neurona está determinado por la excitación en las dendritas. Esta excitación proviene de neurotransmisores de diferentes conexiones sinápticas. A continuación se muestra una gráfica que resume el comportamiento del potencial de acción.

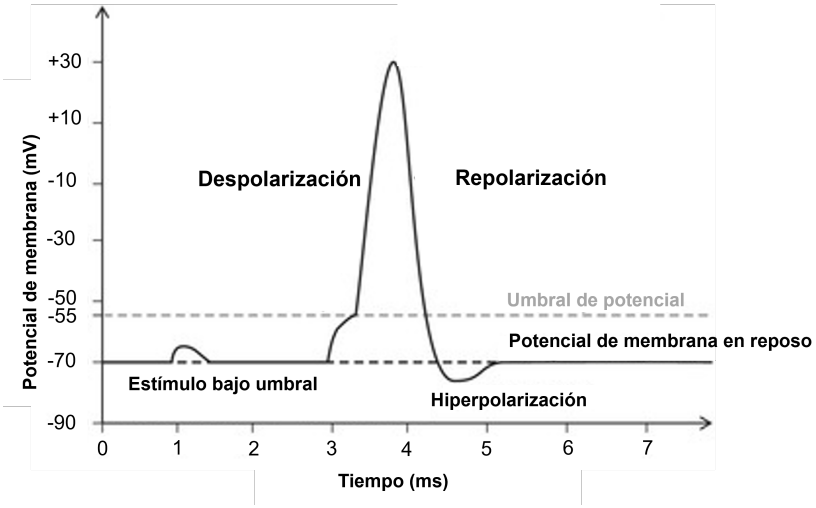


FIGURA 2.3: Evolución de potencial de acción en función del tiempo
Adaptado de [5]

En la figura 2.3 se muestra distintas fases del potencial de acción. Lo primero que vemos es que en reposo el voltaje del potencial de la membrana de la neurona es de -70 mV debido a que inicialmente la membrana de la neurona está permeada de iones de potasio K+ [19]. Luego, dependiendo de la amplitud del estímulo recibido, si se supera un umbral de -55mV aproximádamente entonces se libera el potencial

de acción. A esta fase se le conoce como Fase de Despolarización [21]. Durante esta fase se abren los canales iónicos de la membrana dejando entrar iones de sodio Na+. Una vez despolarizada, el voltaje de la membrana alcanza un máximo, a partir del cual se abren los canales de iones de K+, lo que causa que los iones de potasio salgan de la membrana y por ende disminuya el voltaje del potencial en la membrana. En general, la disminución del voltaje es tal que se llega a un voltaje menor que el del potencial en reposo hasta que finalmente se estabiliza a su valor inicial de reposo en -70 mV, esta es la fase de hiperpolarización.

El sistema nervioso juega un rol muy importante en el procesamiento de la información para controlar las actividades motoras y sensoriales del cuerpo. Adicionalmente, las neuronas son células fundamentales en el sistema nervioso debido a que son las encargadas de transmitir los mensajes químicos y/o eléctricos que reciben, debido a estímulos tanto externos como internos al cuerpo. Sin embargo, el funcionamiento de las neuronas no se restringe únicamente a cambios eléctricos, sino también a fenómenos vasculares como se explicará en la siguiente sección.

2.4. Acoplamiento neurovascular

Entender la actividad neuronal en el cerebro ha sido de gran interés para muchos investigadores, ya que si una persona presenta anomalías motoras o sensoriales estas pueden ser causadas por una falla en el sistema nervioso [35]. En consecuencia se han investigado distintas formas de identificar la actividad neuronal por medio de otras señales. Entre las técnicas principales desarrolladas se destaca las imágenes por resonancia magnética , las cuales son técnicas no invasivas que permiten obtener imágenes de alta calidad del cerebro con buen detalle acerca de su anatomía [28].

2.4.1. Resonancia mágnetica funcional

Una técnica para identificar las zonas del cerebro con actividad neuronal es la de imagen por resonancia magnética funcional, también conocida como fMRI por sus siglas en inglés. Este procedimiento es muy efectivo porque revela con buena precisión las regiones del cerebro que se activan cuando se realiza alguna tarea en específico [42] como lo vemos en la figura 2.4.

En la figura 2.4 se visualiza imágenes de resonancia magnética funcional desde arriba (imágenes superiores), de frente (abajo a la izquierda) y de lado (abajo derecha). Como se puede ver, hay una zona específica coloreada en amarillo. Esta zona corresponde a la sección del cerebro en la cual se detectaron cambios hemodinámicos y por lo tanto, se asume también que ahí existe actividad eléctrica.

La fMRI marca las regiones de interés a partir de la señal BOLD, *Blood Oxygen Level Dependent* en inglés, la cual indica cambios regionales en los niveles de oxigenación en la sangre. Esta técnica fue desarrollada por Seiji Ogawa en 1990 [27]. Estos cambios se generan por cambios en el flujo sanguíneo, el volumen de capilares u otro fenómeno hemodinámico. En particular, se asume que existe una alta correlación entre la actividad neuronal y la señal BOLD, debido a que las neuronas no tienen reservas internas de glucosa y oxígeno para su correcto funcionamiento.

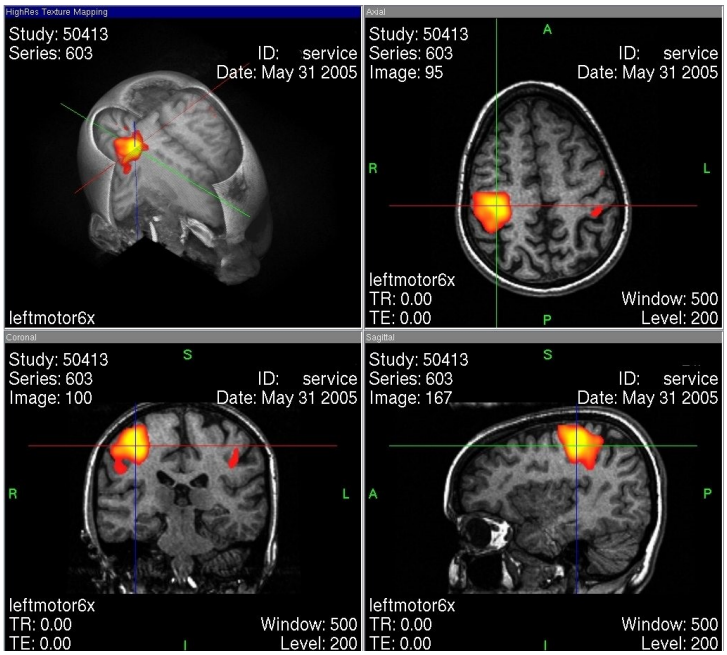


FIGURA 2.4: Imagenes de resonancia magnética funcional desde distintos ángulos

Tomado de https://commons.wikimedia.org/w/index.php?curid=1918499

[30]. En consecuencia, cuando las neuronas procesan información requieren de más nutrientes como el oxígeno y la glucosa, los cuales son entregados por el torrente sanguineo, y de esta forma aumenta la cantidad de oxyhemoglobina, la molécula de hemoglobina que se encuentra acoplada a una molecula de oxígeno, y se reduce la cantidad de deoxihemoglobina, la molecula de hemoglobina que ya ha entregado al tejido su molécula de oxígeno [29]. De esta forma cambia la relación oxi-deoxi hemogoblina, la cual es un marcador de actividad neuronal en las imágenes de resonancia magnética.

2.4.2. Principios básicos del acoplamiento neurovascular

Como vimos anteriormente, se puede asumir una fuerte relación entre la actividad neuronal y el aumento de oxigenación en la sangre, el cual está relacionado con el aumento de flujo sanguíneo en el cerebro. A este proceso se le conoce como *acoplamiento neurovascular* y se traduce en el aumento de flujo sanguineo en la región de activación para conservar el equilibrio entre el consumo de energía, debido a la activación, y la disponibilidad de sustrato, como oxígeno y glucosa, para la producción de adenosín trifosfato o ATP. El ATP es una molécula energética que se utiliza para sostener el metabolismo celular.

El acoplamiento neurovascular funciona a partir de un conjunto de células que forman la estructura representada en la figura 2.5.

En esta estructura se observa que, la unidad neurovascular se compone de distintas células tales como: las células endoteliales, las cuales cubren los vasos sanguineos

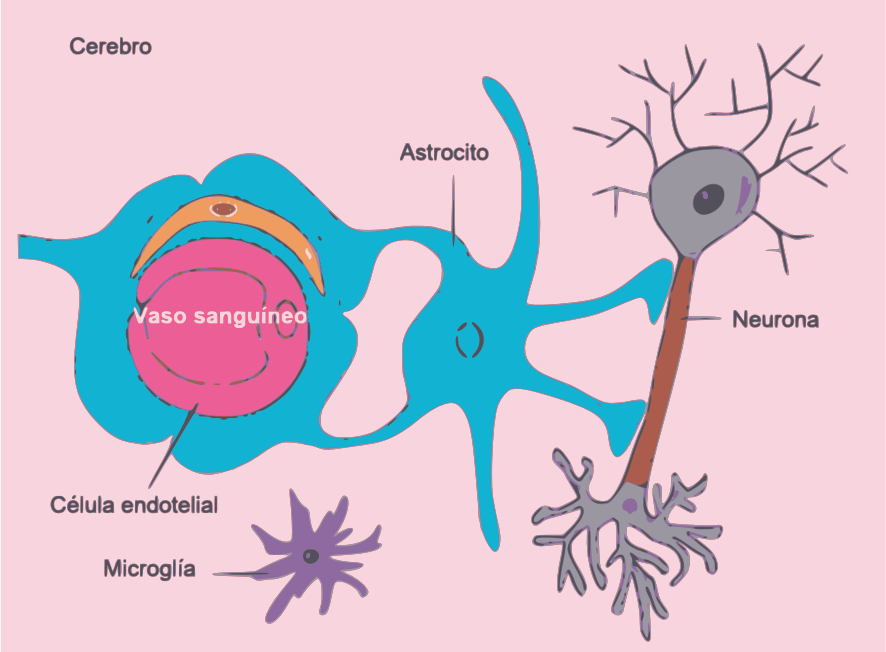


FIGURA 2.5: Esquema de unidad neurovascular
Tomado de https://www.scienceinschool.org/es/content/guardi%C3%A1n-del-cerebro-la-barrera-hematoencef%C3%A1lica,

y se encargan de regular los intercambios desde y hacia el exterior del sistema nervioso central; los astrocitos, los cuales ejercen muchas funciones esenciales en el sistema nervioso central sano, entre ellas, para identificar trastornos y patologías del SNC [36]; las microglías, células presentes en el SNC las cuales cumplen funciones imunológicas como la respuesta inflamatoria [2], y por suspuesto, las neuronas, cuya estructura y funcionamiento ya fue resumido anteriormente.

Ahora que ya se describió la estructura y funcionamiento del acoplamiento neurovascular, en el aporte de nutrientes a las neuronas para suplir la energía que requieren, se explicará cómo se genera esta energía por medio de la respiración celular.

2.4.3. Respiración celular

Anteriormente se mencionó que el oxígeno hace parte de lo que contiene la sangre para suministrarle energía a las células, pero todavía no se ha explicado como se crea esta energía. A continuación se describe de este proceso.

El oxígeno es fundamental para mantener la vida en el cuerpo humano y crear energía en forma de ATP por medio de la respiración aeróbica. De hecho, el ATP es una molécula de energía encontrada en las células de todos los seres vivos que actúa como el combustible de distintos procesos celulares, entre ellos la propagación del potencial de acción [8]. El ATP es generado por la respiración celular, la cual consiste en un conjunto de reacciones bioquímicas en las que compuestos orgánicos son degradados completamente, por oxidación, hasta convertirse en sustancias inorgánicas [41]. El oxígeno que proviene del aire inhalado, permite la oxidación de los compuestos orgánicos, luego pasa por los alvéolos y finalmente a la sangre a través

de los capilares que revisten las paredes alveolares. Este proceso es realizado por el orgánulo celular conocido como mitocondria [38] y proporciona energía aprovechable para la célula, principalmente en forma de ATP. La ecuación química (2.1) modela la interacción entre el oxígeno, la glucosa consumidos, y el ATP producido:

$$C_6H_{12}O_6 + 6O_2 \rightarrow 6H_2O + 6CO_2 + 38ATP. \tag{2.1}$$

Como se puede ver, la oxigenación en la sangre es fundamental para el correcto funcionamiento de las células. El medio de transporte del oxígeno es la sangre por lo cual el análisis del comportamiento del flujo sanguíneo resulta de gran interés para comprender diversos fenómenos vasculares. Por ejemplo, en el artículo de Dunn, Mythen y Grocott [9] se describe el transporte de oxígeno como un proceso dividido en 2 partes principales:

- La **convección**: movimiento de oxígeno a través del transporte en masa y que requiere energía derivada del bombeo de sangre del corazón.
- La **difusión**: movimiento pasivo del oxígeno definido por un gradiente de concentración [24] que atraviesa tejidos hasta las mitocondrias, pequeños compartimientos dentro de las células encargadas de transformar nutrientes, como la glucosa, en energía para las células a partir de la respiración celular [38].

2.4.4. La ley de Fick como modelo básico para describir el acoplamiento neurovascular

En general, los modelos matemáticos para describir la difusión de oxígeno en los tejidos pulmonares parten de las leyes de difusión de Adolf Fick [9]. La primera ley afirma que el flujo de partículas de una substancia se mueve desde un punto de alta concentración a un punto de baja concentración; los cambios espaciales de las partículas están gobernadas por la ecuación (2.2):

$$J = -D\frac{\partial C}{\partial x}. \tag{2.2}$$

donde J es el flujo difusivo el cual se mide en $mol/m^2.s$, D es el coeficiente de difusión en m^2/s, C es la concentración medida en mol/m^3 y x es la posición en m. La segunda ley de Fick relaciona los cambios del gradiente de concentración en función del tiempo junto con la aceleración de las concentraciones de una substancia en una dimensión espacial, como se expone en la ecuación (2.3):

$$\frac{\partial C}{\partial t} = D\frac{\partial^2 C}{\partial x^2}. \tag{2.3}$$

donde t representa el tiempo. Sin embargo, las leyes de Fick no brindan suficiente información sobre la relación entre los procesos eléctricos de las neuronas con los cambios vasculares. Además, aunque estas leyes describen principalmente los cambios de oxígeno en la sangre en el dominio espacial, en esta tesis se ignorara las

relaciones con los cambios espaciales dado que nos interesa más los cambios relacionados con el tiempo.

El procesamiento de información neuronal es un proceso que involucra tanto actividad eléctrica como vascular. Por lo tanto, vale la pena considerar ambos comportamientos si se quiere partir de estos conceptos biológicos para crear un modelo de neuronas artificiales. De esta forma, se pueden desarrollar nuevos modelos que representen mejor el procesamiento de la información que se realiza en nuestro cerebro.

Capítulo 3

Fundamentos de Neuronas Artificiales

Las redes neuronales artificiales (RNA) son un modelo matemático y computacional que está inspirado en una forma simplista y simplificada de cómo funcionan las neuronas en el cerebro [40]. Antes que se diseñaran las arquitecturas de redes neuronales, se formularon unos primeros modelos más simples de neuronas artificiales que simulan el comportamiento eléctrico de las neuronas biológicas.

3.1. Neurona de McCulloch-Pitts

La idea inicial de modelar matemáticamente una red neuronal surgió a partir de la creación de un modelo matemático de una neurona, específicamente en lo que se conoce como la neurona de McCulloch-Pitts en 1943 [25]. Este modelo fue propuesto por el neurofisiólogo Warren McCulloch (16 de noviembre de 1898 - 24 de septiembre de 1969) y el lógico Walter Pitts (23 de abril de 1923 - 14 de mayo de 1969) . Este modelo simplificado busca representar la actividad eléctrica que se observa en la neurona y se basa en el hecho de que el comportamiento de una neurona biológica depende de un conjunto de señales de excitación, provenientes de otras neuronas.

Si se asume que la neurona se comporta como un sistema lineal, y que cada una de las conexiones sinápticas tiene la misma probabilidad de disparar el potencial de acción en una neurona, el comportamiento de esta se puede representar por medio de la neurona de McCulloch-Pitts presente en la figura 3.1.

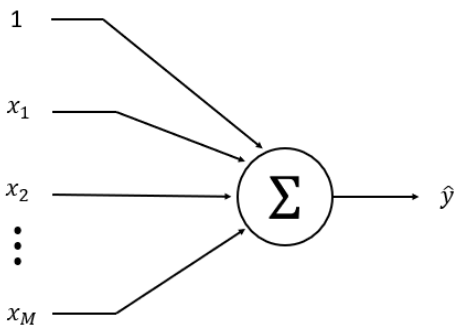


FIGURA 3.1: Representación esquemática de la neurona de McCulloch-Pitts

Matemáticamente el modelo de McCulloch-Pitts expresa las señales de excitación de la neurona como valores de entrada representados en un vector $\boldsymbol{x}$, el soma o cuerpo de la neurona representa la operación de adición y la salida es simplemente la suma de las entradas

$$y = \sum_{i=1}^{M} x_i \tag{3.1}$$

donde M representa el número de variables de entrada. Si la suma de las entradas es lo suficientemente "fuerte" y supera un cierto umbral entonces la salida de la neurona artificial es 1, de lo contrario es 0; equivalentemente, el potencial de acción se dispara si el conjunto de las señales de excitación supera un umbral.

3.2. Perceptrón

Aunque la neurona de McCulloch-Pitts ya representa de manera muy simplificada el funcionamiento de una neurona biológica, está muy restringida debido a que asume que las entradas del modelo son binarias, y que cada entrada tiene la misma importancia en la salida.
Por lo tanto, para relajar un poco las restricciones iniciales, el modelo se expande asignándole un peso a cada entrada de tal forma que la contribución de una entrada especifica en la salida es equivalente al valor de la entrada multiplicado por ese peso. A su vez, la salida es ahora una suma ponderada de los valores de entrada [34]. Además, para poder generalizar e introducir un comportamiento no lineal, esa suma ponderada corresponde a la entrada de lo que se denomina como la función de activación f. Por último, sea u un umbral y sea $z = \sum_{i=1}^{M} \omega_i x_i$, el modelo matemático de una neurona se puede expresar como:

$$\hat{y} = f(z) = \begin{cases} 1 \ , & \text{si } z > u \\ \\ 0 \ , & \text{de lo contrario} \quad . \end{cases} \tag{3.2}$$

Este modelo se conoce como el **perceptrón**, y fue introducido en 1958 por Frank Rosenblatt [32]. Este modelo es la unidad básica de las redes neuronales artificiales.

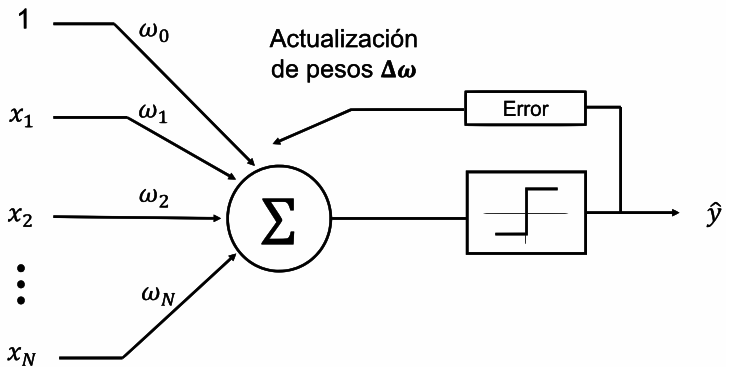
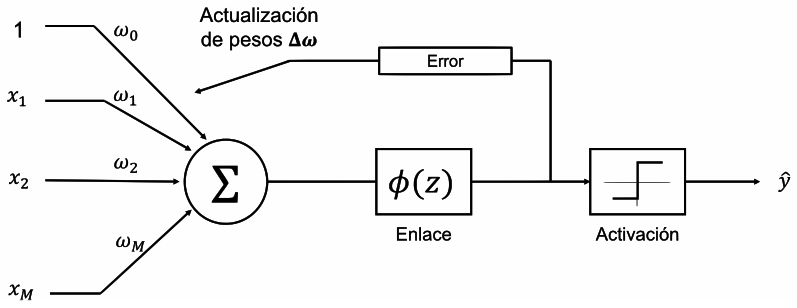


FIGURA 3.2: Representación esquemática del perceptrón

Cabe resaltar que en los años siguientes a la creación del modelo de McCulloch-Pitts, se presentaron nuevas arquitecturas de neuronas artificiales que buscan mejorar la flexibilidad de las arquitecturas anteriormente propuestas.

3.3. Adaline

El Adaline, o Adaptative Linear Neuron en inglés, es una arquitectura propuesta por Bernard Widrow y Ted Hoff en 1960 [44]. El Adaline es una variación del Perceptrón, ya que su modificación principal consiste en cambiar el punto desde el cual se calcula el error para actualizar los pesos. Adicionalmente, se define una función de enlace $\phi(z)$ la cual recibe como entrada el resultado de la suma ponderada de la neurona y su salida es la entrada a la función de umbralización. Además, a diferencia del Perceptrón cuya salida es 1 o 0, en el Adaline la salida es 1 o -1. Con estas modificaciones se obtiene una mayor flexibilidad en el entrenamiento con actualizaciones en los pesos más precisas ya, que la diferencia del error calculado no es solo 1, 0 o -1, sino que ahora puede ser cualquier valor real.



Adaptado de
https://vitalflux.com/adaline-explained-with-python-example/

FIGURA 3.3: Representación esquemática de ADALINE

3.4. Neurona logística

La neurona logística es otro tipo de neurona artificial cuya característica principal es que su función de activación es la función logística, que se define como:

$$\phi(z) = \frac{1}{1+e^{-z}} \quad . \tag{3.3}$$

Esta función se caracteriza por estar acotada inferiormente por 0 y superiormente por 1. Teniendo en cuenta que la salida del modelo de la neurona logística es 1 o 0, la selección de la función logística como función de enlace es conveniente, dado que es derivable y el gradiente es suave, ya que con la función de activación identidad (regresión) la suma ponderada $\sum_{i=1}^{M} \omega_i x_i$ puede tomar valores por fuera del intevalo [0,1], incluso con magnitud muy grande, lo que puede afectar el entrenamiento de los

pesos. Sin embargo, con la función logística la suma ponderada se escala a un valor entre 0 y 1, y por ende los cálculos realizados en el entrenamiento pueden mejorar los resultados finales.
Para resumir, el funcionamiento de las distintas propuestas de unidad neuronal básica artificial presentadas se pueden generalizar como muestra la figura 3.4:

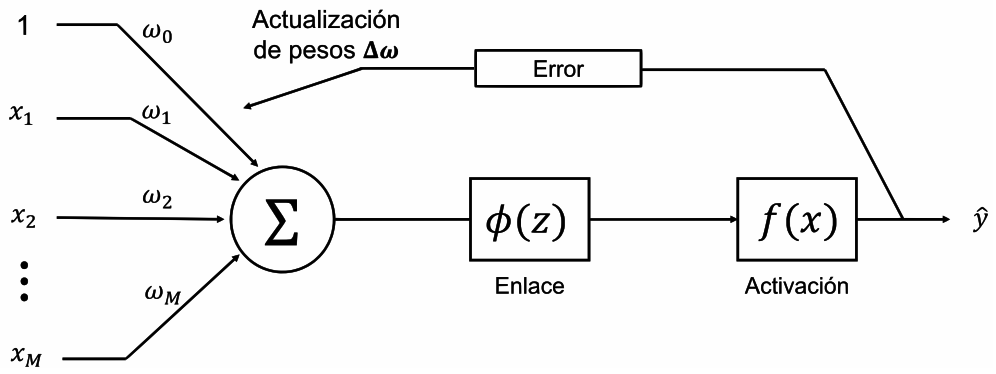


FIGURA 3.4: Esquema de neurona artificial generalizada

En la figura 3.4 vemos que hay un vector de entradas $\boldsymbol{x}$ multiplicado por un vector de pesos $\boldsymbol{\omega}$. El resultado de esta operación se pasa por una función de enlace $\phi(z)$ cuya salida es la entrada a la función de salida de $f(x)$. Finalmente, el resultado de la función f es la estimación $\hat{y}$ del modelo. Nótese que hasta el momento se ha descrito la salida como una variable binaria, es decir de clasificación. Sin embargo, dependiendo el problema en el que se quiera aplicar el modelo, clasificación o regresión, se puede cambiar el tipo de función ϕ (lineal, logística) y el de f (función escalón o lineal). Sin embargo, dependiendo el tipo de problema en el que se utilicen estas arquitecturas se debe elegir una regla de aprendizaje adecuada. En la siguiente sección se explican estas reglas de aprendizaje y la relación entre ellas.

3.5. Reglas de aprendizaje

Las reglas de aprendizaje son algoritmos diseñados para que diferentes modelos adapten sus pesos y se ajusten a la información disponible, de esta forma aprendan la estructura de dicha información. La formulación de estas reglas se fundamenta en dos áreas científicas. Por un lado, se parte de conceptos biológicos acerca del aprendizaje neuronal y de la formulación de un modelo matemático que simule el procesamiento de la información neuronal. Por otro lado, se utiliza la teoría matemática de optimización, en la que se define un método para encontrar mínimos de funciones, y en este caso, mínimos de la función de error de predicción.

3.5.1. Regla del perceptrón

El perceptrón es una arquitectura diseñada para problemas de clasificación binarias. Como se explicó en la sección 3.2, el perceptrón realiza una suma ponderada con un vector de coeficientes, $\boldsymbol{\omega}$, para estimar la salida. Los coeficientes $\boldsymbol{\omega}$ definen

un hiperplano de separación de los valores de entrada. En la figura 3.5 se muestran tres ejemplos de separación entre datos extraídos de dos clases con puntos en $\mathbb{R}^2$.

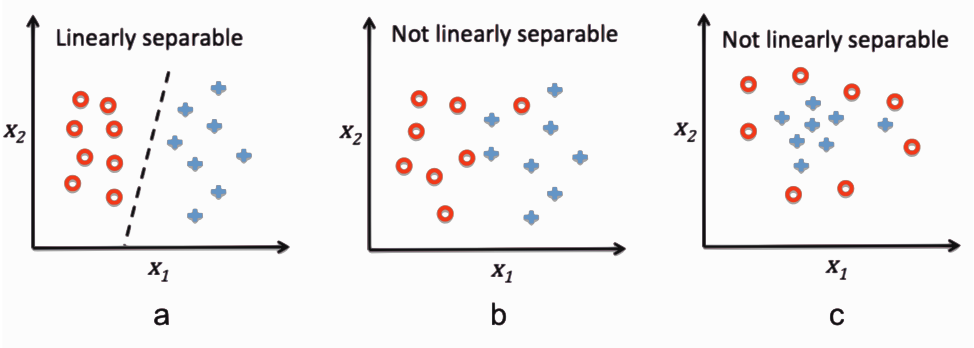


FIGURA 3.5: Ejemplos de separación lineal y no lineal
Adaptado de https://www.javiercancela.com/2017/03/19/python-machine-learning-ii/,

En el ejemplo a de la figura 3.5 se muestra un caso de separación lineal ya que se puede dibujar una recta que separe los puntos de la clase azul y roja, mientras que en los ejemplos b y c no hay forma de separar los puntos rojos de los azules con una recta. Cabe resaltar que en el dibujo b de la figura 3.5, el mejor hiperplano de separación de las clases tendría a lo sumo 3 puntos mal clasificados mientras que en el ejemplo c, si se utiliza una línea recta para separar los datos el error de clasificación será mayor que en el ejemplo b.

En términos matemáticos, un problema de clasificación entre dos clases se puede formular a partir de una matriz de datos $\boldsymbol{X} \in \mathbb{R}^{N\times M}$, donde N es la cantidad de filas de la matriz y M la cantidad de columnas, tal que la i-ésima fila de la matriz es un punto en $\mathbb{R}^M$. Además, se define un vector de sesgo $\boldsymbol{b} = \mathbb{1} \in \mathbb{R}^{N\times 1}$ el cual se concatena a la matriz $\boldsymbol{X}$ formando así la matriz diseño $\tilde{\boldsymbol{X}} = [\boldsymbol{b}|\boldsymbol{X}]$. Adicionalmente, se tiene un vector $\boldsymbol{y} \in \mathbb{R}^N$, el cual indica a qué clase pertenece cada punto. En general, se codifica las clases con 1 o 0. Luego, el vector de etiquetas estimado es $\hat{\boldsymbol{y}} = f(\boldsymbol{X}\boldsymbol{\omega})$ con $\boldsymbol{\omega} \in \mathbb{R}^{M+1}$.
Una vez calculadas las estimaciones, el perceptrón debe actualizar sus coeficientes $\boldsymbol{\omega}$ para mejorar sus predicciones por medio de la siguiente regla:

$$\boldsymbol{\omega} := \boldsymbol{\omega} + \Delta\boldsymbol{\omega} \tag{3.4}$$

$$\Delta\boldsymbol{\omega} = \tilde{\boldsymbol{X}}^T(\boldsymbol{y} - \hat{\boldsymbol{y}}) \tag{3.5}$$

El algoritmo de aprendizaje del perceptrón se presenta en el Algoritmo 1.

3.5.2. Descenso del gradiente

Una técnica muy utilizada en optimización es el algoritmo de descenso del gradiente. La idea detrás de este algoritmo es utilizar la dirección opuesta del gradiente

de la función considerada para así encontrar un punto mínimo [4].
En el aprendizaje de máquina, la función que se busca minimizar es la función de error de predicción, o función de costo, la cual depende de los parámetros de la neurona artificial $\boldsymbol{\omega}$. A esta función la llamaremos $J(\boldsymbol{\omega})$.
Para simplificar la descripción del problema consideremos un punto $\boldsymbol{x} \in \mathbb{R}^M$ y definamos:

$$\begin{aligned} \hat{y} &= h_{\boldsymbol{\omega}}(\boldsymbol{x}), \\ e &= (\hat{y} - y), \end{aligned} \tag{3.6}$$

donde $h_{\boldsymbol{\omega}}(\boldsymbol{x})$ es el modelo propuesto, $\hat{y}$ es la predicción del modelo y e es el error de aproximación.

Algoritmo 1: Entrenamiento de perceptrón

Result: Vector de pesos $\boldsymbol{\omega}$ entrenados
initialization;
$\omega \leftarrow$ inicialización aleatoria, $\forall \omega \in \boldsymbol{\omega}$;
$u \leftarrow 0{,}5$;
$\epsilon > 0$ máxima tolerancia del error;
$\boldsymbol{z} \leftarrow \boldsymbol{X\omega}$;
$\hat{\boldsymbol{y}} = (\hat{y_1}, ..., \hat{y_N})^T, \hat{y_i} = 1$ if $z_i > u$, else $\hat{y_i} = 0$;
$e \leftarrow J(\boldsymbol{\omega}) = \boldsymbol{(\hat{y} - y)}$;
while $e \geq \epsilon$ **do**
  $\boldsymbol{\Delta\omega} = \boldsymbol{X}^T\boldsymbol{(\hat{y} - y)}$;
  $\boldsymbol{\omega} := \boldsymbol{\omega} + \boldsymbol{\Delta\omega}$;
  $e \leftarrow J(\boldsymbol{\omega})$;
end

En los problemas de regresión, cuando se realiza una regresión lineal, el modelo propuesto es $h_{\boldsymbol{\omega}}(\boldsymbol{x}) = \boldsymbol{\omega}^T\boldsymbol{x}$ y por ende, se estima una variable independiente $\hat{y}$ como una suma ponderada de variables independientes x_i de la siguiente manera:

$$\hat{y} = \omega_0 + \omega_1 x_1 + ... + \omega_M x_M = \boldsymbol{\omega}^T\boldsymbol{x}. \tag{3.7}$$

Además, en los modelos de regresión lineal la estimación $\hat{y}$ es un número real y la función de error que se minimiza para una sola observación es

$$J(\boldsymbol{\omega}) = \frac{1}{2}(h_{\boldsymbol{\omega}}(\boldsymbol{x}) - y)^2 = \frac{1}{2}(\boldsymbol{\omega}^T\boldsymbol{x} - y)^2, \tag{3.8}$$

Si se considera N observaciones la función de error es igual a

$$J(\boldsymbol{\omega}) = \frac{1}{2N}(\boldsymbol{X\omega} - \boldsymbol{y})^T(\boldsymbol{X\omega} - \boldsymbol{y}) \tag{3.9}$$

Una vez definida una función de costo, se puede definir el cambio de los coeficientes $\boldsymbol{\omega}$ con las ecuaciones (3.10) y (3.11)

$$\omega_i := \omega_i + \Delta\omega_i, \tag{3.10}$$

$$\Delta\omega_i = -\eta \frac{\partial J(\boldsymbol{\omega})}{\partial \omega_i}, \tag{3.11}$$

donde la constante η se denomina la tasa de aprendizaje, la cual es un hiperparámetro del modelo de aprendizaje. Los hiperparámetros de los modelos de aprendizaje de máquina son muy importantes debido a que dependiendo de su valor el error de predicción de los modelos puede mejorar o empeorar considerablemente. Sin embargo, a diferencia de los parámetros internos del modelo, el valor óptimo de los hiperparámetros para minimizar la función de error no se calcula durante el proceso de aprendizaje del modelo [17].

Para analizar más en detalle los efectos de la tasa de aprendizaje, consideremos la evolución de la función de costo para diferentes valores de η tal como muestra la figura 3.6.

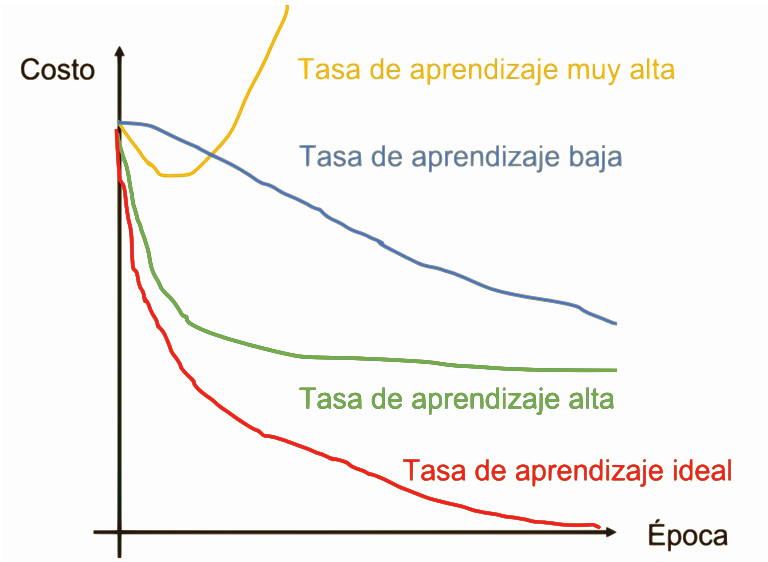


FIGURA 3.6: Evolución de función de costo en función de valor de tasa de aprendizaje

En la figura 3.6 se muestra la evolución de cuatro curvas de la función de costo, a medida que aumenta la cantidad de iteraciones de entrenamiento, o épocas, para distintas tasas de aprendizaje. Nótese que, aunque todas comienzan con el mismo error ya que se inicializa los coeficientes $\boldsymbol{\omega}$ con los mismos valores, la tasa de aprendizaje influye considerablemente en la evolución del error.
De hecho, para la curva amarilla se utilizó una tasa de aprendizaje muy alta y en consecuencia el algoritmo diverge. En el caso de la curva verde se utilizó una tasa de aprendizaje alta, menor que en el caso de la curva amarilla, y aunque se logra disminuir el error, el descenso del gradiente llega un punto que no puede encontrar el mínimo lo que hace que la función de costo converja de forma abrupta a un valor que no necesariamente es el mínimo de la función de costo, en este caso es probable que los valores de los parámetros oscilen. En tanto a la curva roja, se utiliza una tasa de aprendizaje menor que la verde y vemos que el error disminuye progresivamente e incluso converge a un valor menor. En la curva azul se utiliza la menor de las cuatro

tasas de aprendizaje y aunque se llega al mínimo de la función de error, así como con la gráfica roja, la cantidad de épocas para que esto pase es mucho mayor. Es importante que al aplicar el descenso del gradiente se evalúe la evolución del error para determinar si los hiperparámetros seleccionados garantizan convergencia del algoritmo.
Teniendo en cuenta las ecuaciones (3.10) y (3.11) el algoritmo de gradiente descendente se define de la siguiente manera [4]:

Algoritmo 2: Entrenamiento con descenso del gradiente

Result: Vector de pesos $\boldsymbol{\omega}$ entrenados
initialization;
$\omega \leftarrow$ valor aleatorio, $\forall \omega \in \boldsymbol{\omega}$;
Definir: u, η;
$\boldsymbol{z} \leftarrow \boldsymbol{X}\boldsymbol{\omega}$;
$\hat{\boldsymbol{y}} = (\hat{y_1}, ..., \hat{y_N})^T, \hat{y_i} = 1$ if $z_i > u$, else $\hat{y_i} = 0$;
$e \leftarrow J(\boldsymbol{z}) = \frac{1}{2N}(\boldsymbol{z} - \hat{\boldsymbol{y}})^T(\boldsymbol{z} - \hat{\boldsymbol{y}})$;
while $e \geq \epsilon$ **do**
  $\boldsymbol{\omega} := \boldsymbol{\omega} - \eta \vec{\boldsymbol{\nabla}} J(\boldsymbol{\omega})$;
  $\boldsymbol{z} \leftarrow \boldsymbol{X}\boldsymbol{\omega}$;
  $e \leftarrow J(\boldsymbol{z})$;

3.5.3. Consideraciones de la función de error

Si la función del costo es convexa, el uso del descenso del gradiente garantiza que se pueda llegar al mínimo global de esta función.
Para explicar más en detalle estas condiciones considere la función de la ecuación (3.9). Esta función se denomina error cuadrático medio, o MSE de la expresión en inglés Mean Square Error, la cual es convexa, lo que asegura que el descenso del gradiente pueda converjer a un mínimo global. Sin embargo, en el caso de la neurona logística, o equivalentemente para la regresión logística, esta función de costo no es convexa. En consecuencia, se define otra función de costo llamada entropía cruzada [7]. Recordemos que en el modelo logístico la función de enlace está dada por:

$$h_{\omega}(\boldsymbol{x}) = \frac{1}{1 + e^{-\omega^T \boldsymbol{x}}} \quad , \tag{3.12}$$

la cual tiene como rango de valores el intervalo (0,1).

Luego, para problemas de clasificación con dos clases la nueva función de costo se define como:

$$J(h_{\omega}(\boldsymbol{x}), y) = \begin{cases} -\log(h_{\omega}(\boldsymbol{x})) & , \text{ si } y = 1 \\ -\log(1 - h_{\omega}(\boldsymbol{x})) & , \text{ si } y = 0 \end{cases} \tag{3.13}$$

o equivalentemente:

$$J(h_\omega(\boldsymbol{x}), y) = -y \log(h_\omega(\boldsymbol{x})) - (1-y) \log(1 - h_\omega(\boldsymbol{x})) \tag{3.14}$$

La ecuación (3.14) indica que si $h_\omega(\boldsymbol{x})$ es cercana a 1 y $y = 1$ o si $h_\omega(\boldsymbol{x})$ es cercana a 0 y $y = 0$ entonces el costo tiende a 0, de lo contrario, el error crece y tiende a infinito. Por lo tanto, la función de costo para problemas de clasificación que utilicen la neurona logística está dado por:

$$J(\omega) = -\frac{1}{N} \sum_{i=1}^{N} = y_i \log(h_\omega(\boldsymbol{x_i})) + (1 - y_i) \log(1 - h_\omega(\boldsymbol{x_i})). \tag{3.15}$$

Cabe resaltar que se agrega un menos al inicio de la ecuación anterior para que la función sea convexa y no cóncava.

De los modelos de unidades neuronales artificiales presentados anteriormente se puede concluir que su funcionamiento permite encontrar una separación lineal entre dos clases. Sin embargo, esto es una limitación debido a que no siempre se cumple que las clases sean linealmente separables. Por lo tanto, los modelos de neuronas artificiales se extendieron más adelante con el objetivo de representar patrones más complejos, mediante la interconexión de varias neuronas ariticiales, lo que dió nacimiento a un nuevo tipo de arquitecturas de aprendizaje automático: las redes neuronales artificiales (RNA).

3.6. Redes Neuronales Artificiales

Las redes neuronales artificiales (RNA), o redes neuronales en abreviado, son el núcleo de una subárea de estudio del aprendizaje automático de máquina, llamado aprendizaje profundo. Las redes neuronales, al igual que las unidades neuronales artificiales, son arquitecturas inspiradas en la forma en la que las neuronas biológicas trasnmiten señales entre ellas [39]. Su estructura consiste en la interconexión entre neuronas artificiales tal que la salida de unas neuronas es la entrada de otras neuronas. En la figura 3.7 se ilustra un ejemplo de red neuronal:

Como se muestra en la figura 3.7 las redes neuronales se estructuran con varias capas de neuronas artificiales las cuales se dividen en 3 tipos: la capa de entrada la cual corresponde al vector de entradas $\boldsymbol{x}$; las capas ocultas, las cuales reciben información de capas anteriores; y la capa de salida, la cual es la capa final de la red neuronal. Entre más capas tenga la red neuronal se dice que es una red más profunda, de ahí el nombre de estudio dentro de la inteligencia artificial: el aprendizaje profundo. Las redes neuronales tienen distintas configuraciones dependiendo el problema en el cual se utilicen, por ejemplo, en la figura 3.7 hay solo tres capas ocultas pero este número puede variar, así como la cantidad de neuronas por capa y el tipo de arquitectura que determina como estas neuronas se organizan. Usualmente, este tipo de redes, en las que las conexiones son hacia adelante, se llaman redes feed-forward.

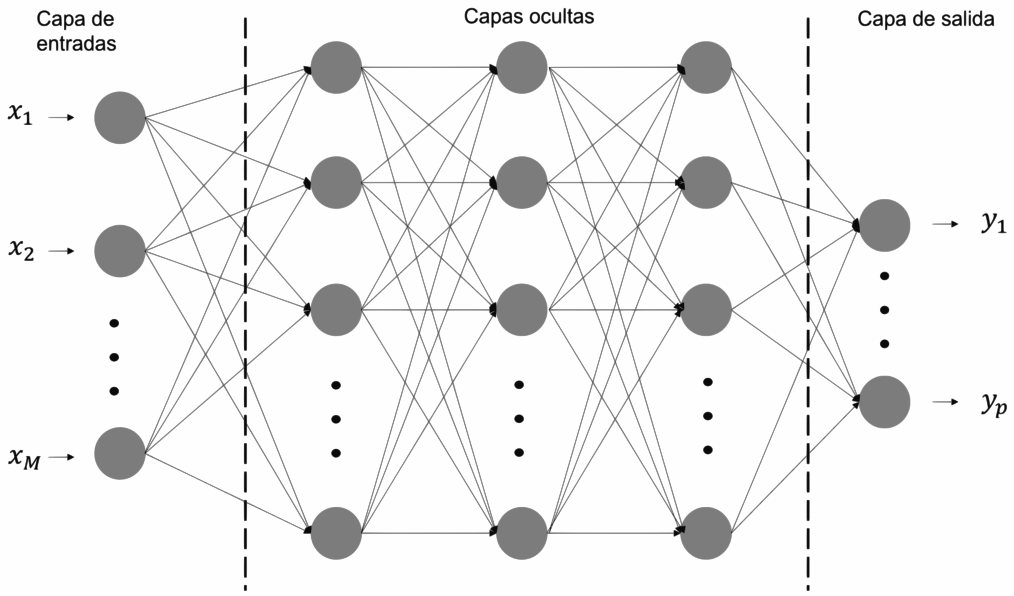


FIGURA 3.7: Esquema de red neuronal artificial

Sin embargo, las redes neuronales en las que la salida de las neuronas se retroalimenta como una nueva entrada se denominan redes neuronales recurrentes [33]. Este tipo de redes suelen utilizarse en problemas con series de tiempo y generalmente se utilizan para realizar pronósticos.

Capítulo 4

Análisis de estabilidad del descenso del gradiente

En el capítulo 3 se describieron distintas arquitecturas de neuronas artificiales, así como los algoritmos de aprendizaje que funcionan detrás de ellas. Estos algoritmos permiten la modificación de los pesos de la salida para que el modelo realice la mejor estimación posible. Aunque se explicó algunas condiciones necesarias sobre la tasa de aprendizaje para que el error converja hacia un mínimo aún no se ha presentado formalmente alguna condición para la tasa de aprendizaje que asegure esta convergencia. Por lo tanto, en este capítulo se explica cuales son las condiciones que debe cumplir la tasa de aprendizaje para garantizar la convergencia del algoritmo del descenso del gradiente. El análisis que se presenta a continuación surge de modelar el descenso del gradiente como un sistema dinámico.

4.1. Fundamentos teóricos de sistemas dinámicos

El análisis de sistemas dinámicos se basa en el estudio de fenómenos que evolucionan en el tiempo así como las relaciones de dichos fenómenos en instantes distintos [23]. Si estos fenómenos se describen en tiempo discreto, en instantes de tiempo fijos y separados entre ellos por una cantidad fija, entonces se modelan por medio de ecuaciones en diferencias, en el caso continuo estos modelos se describen por medio de ecuaciones diferenciales. La elección de si el fenómeno estudiado se modela en tiempo continuo o discreto depende del contexto del mismo. Para esta tesis, dado que las variables que cambian son los coeficientes ω los cuales evolucionan a medida que aumenta la cantidad de iteraciones de entrenamiento (épocas), sólo se hablará acerca de los modelos con ecuaciones en diferencias.
Las variables que evolucionan en el tiempo se denominan variables de estado. Uno de los modelos matemáticos más básicos de ecuaciones en diferencias es el modelo de crecimiento exponencial:

$$\begin{aligned} x[n+1] &= ax[n], \quad a \in \mathbb{R}, \\ x[0] &= x_0. \end{aligned} \tag{4.1}$$

Nótese que dependiendo del valor de la constante a, el comportamiento de la variable de estado x cambia considerablemente.

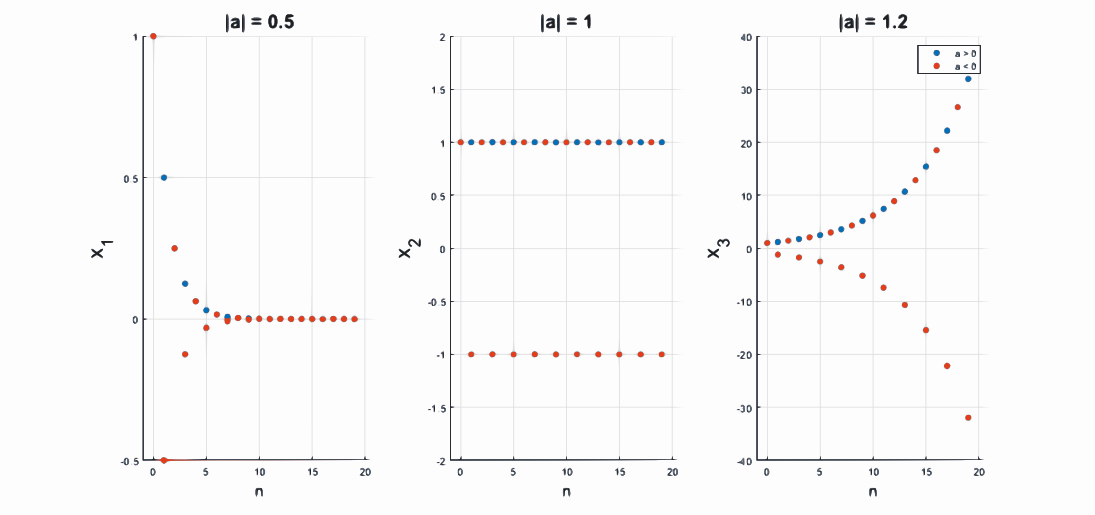


FIGURA 4.1: Evolución de modelo exponencial variando valor de constante a

En efecto, en la figura 4.1 se muestra los tres casos posibles de evolución. Primero, si $|a| < 1$ entonces x poco a poco converge a un valor finito, si $a = 1$, x se mantiene constante y si $|a| > 1$ se tiene que x diverge. Estas gráficas nos sugieren unas primeras condiciones de convergencia de los modelos en espacio de estado los cuales se detallarán más adelante.

Ahora, si se considera de manera general m variables de estado las cuales se relacionan entre ellas linealmente, con coeficientes constantes, entonces se puede obtener un sistema de ecuaciones en diferencias de primer grado con m ecuaciones como el siguiente:

$$\begin{aligned} x_1[n+1] &= a_{11}x_1[n] + \ldots + a_{1m}x_m[n] + b_1 \\ x_2[n+1] &= a_{21}x_1[n] + \ldots + a_{2m}x_m[n] + b_2 \\ &\vdots \\ x_m[n+1] &= a_{m1}x_1[n] + \ldots + a_{mm}x_m[n] + b_m, \end{aligned} \tag{4.2}$$

o de manera más simplificada:

$$\boldsymbol{x}[n+1] = \boldsymbol{A}\boldsymbol{x}[n] + \boldsymbol{b}, \tag{4.3}$$

donde

$$\boldsymbol{A} = \begin{bmatrix} a_{11} & \ldots & a_{1m} \\ \vdots & \ddots & \vdots \\ a_{m1} & \ldots & a_{mm} \end{bmatrix}, \quad \boldsymbol{x}[n] = \begin{bmatrix} x_1[n] \\ \vdots \\ x_m[n] \end{bmatrix}, \quad \boldsymbol{b} = \begin{bmatrix} b_1 \\ \vdots \\ b_m \end{bmatrix} \tag{4.4}$$

Dado que los sistemas dinámicos cambian en el tiempo resulta de interés conocer los puntos del sistema en los que las variables estudiadas se encuentran en un estado de reposo natural. Para comprender estos puntos, es necesario introducir los conceptos de **puntos de equilibrio** y **estabilidad** en sistemas dinámicos.
Por un lado, de acuerdo con la definición de Luenberger [23], un vector $\boldsymbol{x}^*$ es un *punto de equilibrio* de un sistema dinámico, si tiene la propiedad de que una vez el vector de variables de estado del sistema es igual a $\boldsymbol{x}^*$, entonces se mantiene igual a $\boldsymbol{x}^*$ para todo instante futuro. Equivalentemente, si se considera el sistema de la ecuación (4.3) se tiene que:

$$\boldsymbol{x}^* = \boldsymbol{A}\boldsymbol{x}^* + \boldsymbol{b}. \tag{4.5}$$

Con esta definición presentada ya se puede hablar formalmente de estabilidad.
Por otro lado, la estabilidad de un sistema dinámico, en general, se refiere a la estabilidad de sus puntos de equilibrio. De hecho, se dice que un punto de equilibrio es estable si, cuando el vector de estados se aleja ligeramente de un punto de equilibrio entonces este tiende a retornar al punto de equilibrio. Formalmente, se dice que un punto de equilibrio $\boldsymbol{x}^*$ es *asintóticamente estable* si para toda condición inicial el vector de estados tiende a $\boldsymbol{x}^*$ a medida que el tiempo avanza. De lo contrario, se dice que $\boldsymbol{x}^*$ es *inestable* si para alguna condición inicial el vector de estados diverge.

Para ilustrar mejor estos conceptos, considere un sistema lineal no hómogeneo como la ecuación (4.3) y suponga que $\boldsymbol{x}^*$ es un punto de equilibrio de este sistema. Entonces, se tiene que:

$$\boldsymbol{x}[k+1] - \boldsymbol{x}^* = \boldsymbol{A}\boldsymbol{x}[k] + \boldsymbol{b} - (\boldsymbol{A}\boldsymbol{x}^* + \boldsymbol{b}) = \boldsymbol{A}(\boldsymbol{x}[k] - \boldsymbol{x}^*). \tag{4.6}$$

Luego, sea $\boldsymbol{z}[k]$ el vector de estados igual a:

$$\boldsymbol{z}[k] = \boldsymbol{x}[k] - \boldsymbol{x}^*. \tag{4.7}$$

Por lo tanto, el vector de estados $\boldsymbol{x}[k]$ tiende a $\boldsymbol{x}^*$ si y solo si el vector $\boldsymbol{z}[k]$ tiende al vector $\mathbf{0}$ cuando k aumenta. Ahora el nuevo sistema a analizar es:

$$\boldsymbol{z}[k+1] = \boldsymbol{A}\boldsymbol{z}[k]. \tag{4.8}$$

Como vemos en la ecuación (4.8), el comportamiento del sistema está determinado por la matriz $\boldsymbol{A}$. Por lo tanto, la estabilidad del sistema está dada por la estabilidad de la matriz $\boldsymbol{A}$. Antes de continuar con el análisis de estabilidad es necesario mostrar que para el sistema lineal homogéneo de la ecuación (4.8) se puede determinar el comportamiento de las variables de estado por medio de la siguiente ecuación:

$$\boldsymbol{z}[k] = \boldsymbol{A}^k \boldsymbol{z}[0]. \tag{4.9}$$

donde $\boldsymbol{A}^k$ es la multiplicación de la matriz $\boldsymbol{A} \in \mathbb{R}^{m\times m}$ con ella misma k veces y $\boldsymbol{z}[0]$ es la condición inicial del vector de variables de estado. La demostración de la ecuación (4.9) se puede realizar muy brevemente por inducción matemática sobre k.
Con el resultado de la ecuación (4.9), se puede validar la estabilidad del sistema

a partir de los valores propios de $\boldsymbol{A}$. Adicionalmente, asumiendo que $\boldsymbol{A}$ es diagonalizable, se puede afirmar que:

$$\boldsymbol{A} = \boldsymbol{M}\boldsymbol{\Lambda}\boldsymbol{M}^{-1}, \tag{4.10}$$

donde $\boldsymbol{M}$ es una matriz no singular y además $\boldsymbol{\Lambda} \in \mathbb{R}^{m\times m}$ tal que:

$$\boldsymbol{\Lambda} = \begin{bmatrix} \lambda_1 & 0 & \dots & 0 \\ 0 & \lambda_2 & \dots 0 & \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \dots & 0 & \lambda_m \end{bmatrix},$$

y λ_i es un valor propio de $\boldsymbol{A}$ para todo $i \in \{1,\dots,m\}$. Entonces, se puede demostrar por inducción que:

$$\boldsymbol{A}^k = \boldsymbol{M}\boldsymbol{\Lambda}^k\boldsymbol{M}^{-1}. \tag{4.11}$$

Además se tiene que:

$$\boldsymbol{\Lambda}^k = \begin{bmatrix} \lambda_1^k & 0 & \dots & 0 \\ 0 & \lambda_2^k & \dots 0 & \\ \vdots & \vdots & \ddots & \vdots \\ 0 & \dots & 0 & \lambda_m^k \end{bmatrix}.$$

Recordemos que para que el sistema tenga estabilidad asintótica se debe cumplir que el vector $\boldsymbol{z}[k]$ tienda al vector $\mathbf{0}$, lo cual ocurre si la matriz $\boldsymbol{A}^k$ tiende a la matriz $\mathbf{0}$ cuando k tiende a infinito. En efecto, en la ecuación (4.11) se ve que $\boldsymbol{A}^k$ tiende a la matriz $\mathbf{0}$ si:

$$\lim_{k\to\infty} \lambda_i^k = 0 \quad \Leftrightarrow \quad |\lambda_i| < 1, \forall i \in \{1,\dots,m\}. \tag{4.12}$$

En conclusión, para que el sistema sea asintóticamente estable se debe cumplir que todos los valores propios de la matriz $\boldsymbol{A}$ del sistema se encuentren dentro el círculo unitario en el plano complejo. Por otro lado, si al menos un valor propio λ está por fuera del círculo unitario entonces el sistema diverge. En la figura 4.2 se muestra una gráfica que representa las situaciones posibles de los valores propios de la matriz $\boldsymbol{A}$ para estabilidad del sistema.

En la figura 4.2 se ilustra ejemplos de los casos posibles sobre los valores propios de la matriz $\boldsymbol{A}$ del sistema (primero estabilidad asintótica y segundo inestabilidad). Nótese que una última posibilidad es que un valor propio λ_i tenga magnitud igual a 1, es decir, el valor propio sea un punto sobre el círculo unitario. En este caso, el término λ_i^k de la matriz lambda se mantiene sobre el círculo unitario y por ende el sistema tendrá un comportamiento oscilatorio alrededor de un punto de equilibrio.

Ahora que ya se explicó brevemente algunos principios de sistemas dinámicos, estos principios son de gran utilidad para analizar con mayor claridad el algoritmo de gradiente descendente como sistema dinámico, del cual se obtendrá un resultado

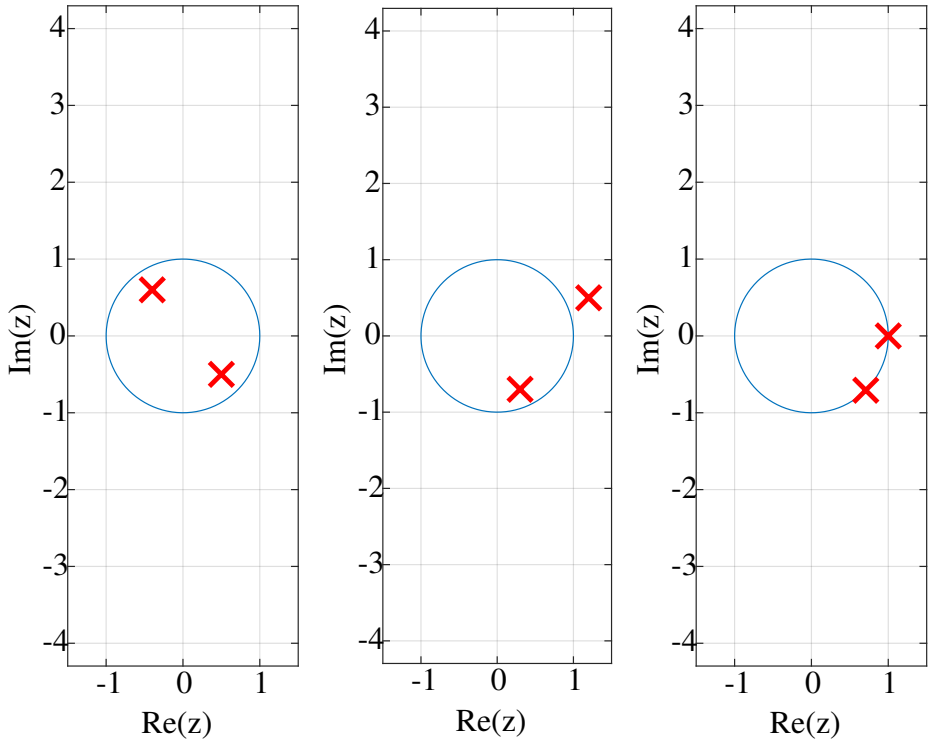


FIGURA 4.2: Casos posibles de ubicacion de valores propios en plano complejo, de iquierda a derecha: estabilidad asintótica, inestabilidad, comportamiento oscilatorio)

clave para asegurar la estabilidad durante el aprendizaje de arquitecturas de neuronas artificiales.

4.2. Análisis de descenso del gradiente como sistema dinámico

En el capítulo 3 se describió el funcionamiento del algoritmo de descenso del gradiente, en particular las ecuaciones (3.10) y (3.11) son las ecuaciones que explican cómo debe actualizarse el vector de pesos $\boldsymbol{\omega}$. Dado que los pesos se actualizan en cada época, este proceso se puede traducir en un sistema dinámico en el que las variables de estado son cada uno de los pesos. En esta sección, se explica cómo se obtiene el sistema dinámico que representa el cambio de los pesos ω el cual servirá para establecer condiciones de estabilidad del descenso del gradiente enfocándose en los valores posibles que puede tomar la tasa de aprendizaje.
Antes de continuar, es necesario resaltar que en esta tésis la notación $f(\boldsymbol{X\omega})$ se refiere a que la función f se aplica a cada uno de los términos del vector resultado de $\boldsymbol{X\omega}$, y análogamente, la notación $f(\boldsymbol{X})$ quiere decir que se aplica la función f

a cada una de las componentes de la matriz $\boldsymbol{X}$. Considere el paso a paso a paso del algoritmo de descenso del gradiente:

1. El vector de pesos $\boldsymbol{\omega}$ es inicializado aleatoriamente
2. Se calcula $\boldsymbol{X\omega}[n]$ y después se calcula una primera estimación $\hat{\boldsymbol{y}}[n] = f(\boldsymbol{X\omega}[0])$, donde f se generalizará como la combinación entre la función de enlace y la función de umbralización
3. Se calcula la función de costo y su derivada J.
4. Los pesos $\boldsymbol{\omega}$ se actualizan mediante las ecuaciones (3.10) y (3.11).

El proceso se repite hasta convergencia. Este proceso se puede modelar con las siguientes ecuaciones:

$$\boldsymbol{\omega}[n+1] = \boldsymbol{\omega}[n] - \eta \vec{\boldsymbol{\nabla}} J(\boldsymbol{\omega})[n] \tag{4.13}$$
$$\hat{\boldsymbol{y}}[n+1] = f(\boldsymbol{X\omega}[n]). \tag{4.14}$$

Para simplificar los cálculos considere inicialmente que el problema que se está resolviendo es de regresión. Adicionalmente, considere los siguientes cálculos para un único vector $\boldsymbol{x} \in \mathbb{R}^M$. Entonces, tenemos que:

$$J(\boldsymbol{\omega})[n] = \frac{1}{2}(\hat{y}[n] - y)^2 \tag{4.15}$$
$$\hat{y}[n] = f(\boldsymbol{\omega}^T \boldsymbol{x}) \tag{4.16}$$
$$\frac{\partial J}{\partial \omega_i} = \frac{\partial J}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial \omega_i} \quad \text{, para algún } i \in \{1, ..., M\}. \tag{4.17}$$

Calculemos las derivadas de la ecuación (4.17):

$$\frac{\partial J}{\partial \hat{y}} = \frac{2}{2}(\hat{y} - y) = (\hat{y} - y),$$
$$\frac{\partial \hat{y}}{\partial \omega_i} = f'(\boldsymbol{\omega}^T \boldsymbol{x}) x_i.$$

Por lo tanto:

$$\frac{\partial J}{\partial \omega_i}[n] = (\hat{y}[n] - y) f'(\boldsymbol{\omega}^T \boldsymbol{x}) x_i. \tag{4.18}$$

Ahora, si $\boldsymbol{X}$ es una matriz tal que:

$$\boldsymbol{X} = \begin{bmatrix} 1 & x_1^{(1)} & \dots & x_M^{(1)} \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_1^{(N)} & \dots & x_M^{(N)} \end{bmatrix},$$

entonces el vector de estimaciones es igual a

$$\hat{\boldsymbol{y}}[n] = f(\boldsymbol{X}\boldsymbol{\omega}[n]),$$

y además

$$\frac{\partial \hat{y}}{\partial \omega_i} = \boldsymbol{D}_{\boldsymbol{x}_i} f'(\boldsymbol{X}\boldsymbol{\omega}[n]),$$

con $\boldsymbol{D}_{\boldsymbol{x}_i} \in \mathbb{R}^{N\times N}$ tal que:

$$\boldsymbol{D}_{\boldsymbol{x}_i} = \begin{bmatrix} x_i^{(1)} & 0 & 0 & \dots & 0 \\ 0 & x_i^{(2)} & 0 & \dots & 0 \\ 0 & 0 & \ddots & \dots & \vdots \\ \vdots & \vdots & \vdots & x_i^{(N-1)} & 0 \\ 0 & 0 & \dots & 0 & x_i^{(N)} \end{bmatrix}$$

En consecuencia, la derivada del vector $\hat{\boldsymbol{y}}$ con respecto a cada uno de los pesos es la matriz jacobiana:

$$\begin{aligned} \boldsymbol{J}_{\hat{\boldsymbol{y}}} &= \begin{bmatrix} \frac{\partial \hat{\boldsymbol{y}}}{\partial \omega_0} & \dots & \frac{\partial \hat{\boldsymbol{y}}}{\partial \omega_M} \end{bmatrix} \\ &= \begin{bmatrix} \boldsymbol{D}_{\boldsymbol{x}_0} f'(\boldsymbol{X}\boldsymbol{\omega}[n]) & \dots & \boldsymbol{D}_{\boldsymbol{x}_M} f'(\boldsymbol{X}\boldsymbol{\omega}[n]) \end{bmatrix} \\ &= \boldsymbol{D}_{f'}\boldsymbol{X}. \end{aligned}$$

y la matriz $\boldsymbol{D}_{f'}$ igual a

$$\boldsymbol{D}_{f'} = \begin{bmatrix} f'(\boldsymbol{X}\boldsymbol{\omega}[n])^{(1)} & 0 & 0 & \dots & 0 \\ 0 & f'(\boldsymbol{X}\boldsymbol{\omega}[n])^{(2)} & 0 & \dots & 0 \\ 0 & 0 & \ddots & \dots & \vdots \\ \vdots & \vdots & \vdots & f'(\boldsymbol{X}\boldsymbol{\omega}[n])^{(N-1)} & 0 \\ 0 & 0 & \dots & 0 & f'(\boldsymbol{X}\boldsymbol{\omega}[n])^{(N)} \end{bmatrix}.$$

Finalmente, la derivada del costo con respecto a cada uno de los pesos de $\boldsymbol{\omega}$ es igual a:

$$\vec{\boldsymbol{\nabla}} J(\boldsymbol{\omega}) = \frac{1}{N} \boldsymbol{J}_{\hat{\boldsymbol{y}}}{}^{T} (\hat{\boldsymbol{y}}[n] - \boldsymbol{y}), \tag{4.19}$$

$$= \frac{1}{N} (\boldsymbol{D}_{f'}\boldsymbol{X})^{T} (\hat{\boldsymbol{y}}[n] - \boldsymbol{y}), \tag{4.20}$$

$$= \frac{1}{N} \boldsymbol{X}^{\boldsymbol{T}} \boldsymbol{D}_{f'}^{\boldsymbol{T}} (\hat{\boldsymbol{y}}[n] - \boldsymbol{y}). \tag{4.21}$$

Por lo tanto, generalizando la ecuación del gradiente decendiente, la actualización de los pesos es igual a:

$$\boldsymbol{\omega}[n+1] = \boldsymbol{\omega}[n] - \eta \vec{\boldsymbol{\nabla}} J(\boldsymbol{\omega}) \tag{4.22}$$

$$= \boldsymbol{\omega}[n] - \frac{\eta}{N} \boldsymbol{X}^T \boldsymbol{D}_{f'}^T (f(\boldsymbol{X}\boldsymbol{\omega}[n]) - \boldsymbol{y}). \tag{4.23}$$

De la bibliografía buscada, un autor que analiza el gradiente descendente como sistema dinámico es Pierre Baldi en su artículo *Gradient Descent Learning Algorithm Overview: A General Dynamical Systems Perspective* [3]. En este artículo se analiza algoritmos para redes neuronales desde el punto de vista de sistemas dinámicos, tanto discretos como continuos y también para distintas arquitecturas de red neuronal (feed-forward y recurrentes). De este artículo, las ecuaciones que son de interés para esta tesis son las ecuaciones (1.3) y (1.4) en las que se presenta un modelo aditivo de un solo compartimiento, es decir, una sola neurona artificial, de la siguiente manera:

$$\frac{du_i}{dt} = -\frac{u_i}{\tau_i} + \sum_j \omega_{i,j} f_j(u_j) + I_i, \tag{4.24}$$

$$\frac{du_i}{dt} = -\frac{u_i}{\tau_i} + \alpha_i f_i(\sum_j \omega_{i,j} u_j) + I_i. \tag{4.25}$$

Nótese que estas ecuaciones son exactamente las mismas a excepción de los términos $\sum_j \omega_{i,j} f_j(u_j)$ y $\alpha_i f_i(\sum_j f_j(\omega_{i,j} u_j))$. Además, según el autor, las dos ecuaciones anteriores son equivalentes bajo una simple transformación lineal, la cual no se detalla en esta tésis por simplicidad. Por lo tanto, teniendo en cuenta los siguientes puntos:

- sólo nos interesa el estudio de una sola neurona artificial (es decir $i = 1$) en las ecuaciones (4.24) y (4.25).
- de acuerdo con la notación que se tenía, $u_j = x_j, \quad j \in \{1, ..., M\}$.

se puede afirmar que:

$$\sum_j \omega_j f_j(u_j) = \alpha f(\sum_j f(\omega_j u_j)) \quad , \quad \alpha \in \mathbb{R}. \tag{4.26}$$

Además, como se cumple que

$$\sum_j \omega_j u_j = \sum_j \omega_j x_j = \boldsymbol{\omega}^T \boldsymbol{x} = \boldsymbol{x}^T \boldsymbol{\omega}.$$

Entonces se tiene que:

$$\boldsymbol{\omega}^T f(\boldsymbol{x}) = \alpha f(\boldsymbol{\omega}^T \boldsymbol{x}) \Leftrightarrow f(\boldsymbol{\omega}^T \boldsymbol{x}) = \frac{1}{\alpha} \boldsymbol{\omega}^T f(\boldsymbol{x}) \quad , \quad \alpha \neq 0. \tag{4.27}$$

Ahora, si consideramos toda la matriz de datos $\boldsymbol{X}$ y definimos $c = \frac{1}{\alpha}$ entonces:

$$f(\boldsymbol{X}\boldsymbol{\omega}) = c f(\boldsymbol{X}) \boldsymbol{\omega}[n] = \hat{\boldsymbol{y}}[n]. \tag{4.28}$$

Así, usando (4.28) en (4.23):

$$\begin{aligned}
\boldsymbol{\omega}[n+1] &= \boldsymbol{\omega}[n] - \eta\vec{\boldsymbol{\nabla}}J(\boldsymbol{\omega}[n]),\\
&= \boldsymbol{\omega}[n] - \frac{\eta}{N}(\boldsymbol{X}^T\boldsymbol{D}_{f'}^T(f(\boldsymbol{X}\boldsymbol{\omega}[n]) - \boldsymbol{y})),\\
&= \boldsymbol{\omega}[n] - \frac{\eta}{N}(\boldsymbol{X}^T\boldsymbol{D}_{f'}^T(cf(\boldsymbol{X})\boldsymbol{\omega}[n] - \boldsymbol{y})),\\
&= \boldsymbol{\omega}[n] - \frac{\eta}{N}c\boldsymbol{X}^T\boldsymbol{D}_{f'}^T f(\boldsymbol{X})\boldsymbol{\omega}[n] + \frac{\eta}{N}\boldsymbol{X}^T\boldsymbol{D}_{f'}^T\boldsymbol{y},\\
&= \left[\boldsymbol{I}_{M\times M} - \frac{\eta}{N}c\boldsymbol{X}^T\boldsymbol{D}_{f'}^T f(\boldsymbol{X})\right]\boldsymbol{\omega}[n] + \frac{\eta}{N}\boldsymbol{X}^T\boldsymbol{D}_{f'}^T\boldsymbol{y}.
\end{aligned}$$

Finalmente, definamos las matrices:

$$\boldsymbol{A} = \boldsymbol{I}_{M\times M} - \frac{\eta}{N}c\boldsymbol{X}^T\boldsymbol{D}_{f'}^T f(\boldsymbol{X}) \tag{4.29}$$

$$\boldsymbol{B} = \frac{\eta}{N}\boldsymbol{X}^T\boldsymbol{D}_{f'}^T\boldsymbol{y} \tag{4.30}$$

En consecuencia, la evolución de los pesos $\boldsymbol{\omega}$ se puede representar por medio del siguiente sistema dinámico:

$$\boldsymbol{\omega}[n+1] = \boldsymbol{A}\boldsymbol{\omega}[n] + \boldsymbol{B}, \tag{4.31}$$
$$\hat{\boldsymbol{y}}[n+1] = f(\boldsymbol{X}\boldsymbol{\omega}[n]). \tag{4.32}$$

Como se puede observar, la primera ecuación del sistema de ecuaciones planteado es un sistema lineal, por lo cual el sustento teórico de la sección 4.1. es fundamental para las derivaciones matemáticas que se explican en la siguiente subsección sobre la cota de la tasa de aprendizaje.

4.3. Derivación de las cotas de la tasa de aprendizaje

En la sección 1 se explicó que la estabilidad de un sistema de ecuaciones en diferencias de primer orden está dado por la estabilidad de la matriz $\boldsymbol{A}$ de la ecuación (4.3).

Primero, suponga que el problema que se está resolviendo es de regresión. Entonces, la función f se puede definir como la función identidad, es decir:

$$f : x \mapsto f(x) = x.$$

Por lo tanto, de la ecuación (4.28) se debe cumplir que:

$$f(\boldsymbol{X}\boldsymbol{\omega}[n]) = \boldsymbol{X}\boldsymbol{\omega}[n] = f(\boldsymbol{X})\boldsymbol{\omega} \Leftrightarrow c = 1. \tag{4.33}$$

Por lo tanto, dado que $f'(x) = 1$ entonces $\boldsymbol{D}_{f'} = \boldsymbol{I}_{N\times N}$, es decir $\boldsymbol{I}_{N\times N}$ es la matriz identidad de tamaño $N \times N$. Así, usando la ecuación (4.29) y (4.33) se tiene que:

$$\boldsymbol{A} = \boldsymbol{I}_{M\times M} - \frac{\eta}{N}\boldsymbol{X}^{\boldsymbol{T}}\boldsymbol{X}$$

Sabemos que $\boldsymbol{X}^{\boldsymbol{T}}\boldsymbol{X}$ es simétrica por lo cual sus valores propios son reales. Ahora, mostremos que la matriz $\boldsymbol{X}^{\boldsymbol{T}}\boldsymbol{X}$ es definida semipositiva. De esta forma, podremos establecer una cota para la tasa de aprendizaje de manera más sencilla a partir de los valores propios de la matriz $\boldsymbol{A}$.
Sea $\boldsymbol{v} \in \mathbb{C}^M$. La notación $\boldsymbol{v}^*$ se refiere al vector transpuesto conjugado de $\boldsymbol{v}$ y $\bar{\boldsymbol{v}}$ es el vector conjugado de $\boldsymbol{v}$. Para que $\boldsymbol{X}^{\boldsymbol{T}}\boldsymbol{X}$ sea definida semipositiva se requiere que:

$$\boldsymbol{v}^*\boldsymbol{X}^{\boldsymbol{T}}\boldsymbol{X}\boldsymbol{v} \geq 0.$$

Nótese que:

$$\boldsymbol{v}^*\boldsymbol{X}^{\boldsymbol{T}} = (\boldsymbol{X}\bar{\boldsymbol{v}})^T.$$

Luego,

$$\boldsymbol{X}\bar{\boldsymbol{v}} = \bar{\boldsymbol{z}} = \begin{bmatrix} \bar{z_1} \\ \vdots \\ \bar{z_N} \end{bmatrix},$$

con $\bar{z_i} = \sum_{j=1}^{M} x_{ij}\bar{v}_j$. Entonces se tiene que:

$$\boldsymbol{v}^*\boldsymbol{X}^{\boldsymbol{T}}\boldsymbol{X}\boldsymbol{v} = \bar{\boldsymbol{z}}^{\boldsymbol{T}}\boldsymbol{z} = \sum_{j=1}^{M} \bar{z_i}z_i.$$

Además, se cumple que

$$\forall z \in \mathbb{C}, |z|^2 = \bar{z}z \geq 0.$$

En consecuencia, $\boldsymbol{v}^*\boldsymbol{X}^{\boldsymbol{T}}\boldsymbol{X}\boldsymbol{v} = \bar{\boldsymbol{z}}^{\boldsymbol{T}}\boldsymbol{z} \geq 0$. Así, la matriz $\boldsymbol{X}^{\boldsymbol{T}}\boldsymbol{X}$ es definida semipositiva.

De lo anterior se puede concluir que cada uno de los valores propios de $\boldsymbol{X}^{\boldsymbol{T}}\boldsymbol{X}$ es mayor o igual a 0. Ahora, busquemos los vectores propios de $\boldsymbol{A}$. Para esto encontremos los valores propios de $\boldsymbol{X}^{\boldsymbol{T}}\boldsymbol{X}$. Estos vectores propios cumplen con la ecuación:

$$\frac{\boldsymbol{X}^{\boldsymbol{T}}\boldsymbol{X}\boldsymbol{v}}{N} = \frac{\lambda\boldsymbol{v}}{N} \quad , \quad \boldsymbol{v} \in \mathbb{R}^M. \tag{4.34}$$

Entonces,

$$\begin{aligned} -\frac{\eta}{N}\boldsymbol{X}^{\boldsymbol{T}}\boldsymbol{X}\boldsymbol{v} &= -\frac{\eta}{N}\lambda\boldsymbol{v}, \\ \Leftrightarrow \boldsymbol{v} - \frac{\eta}{N}\boldsymbol{X}^{\boldsymbol{T}}\boldsymbol{X}\boldsymbol{v} &= \boldsymbol{v} - \frac{\eta}{N}\lambda\boldsymbol{v}, \\ \Leftrightarrow (\boldsymbol{I}_{M\times M} - \frac{\eta}{N}\boldsymbol{X}^{\boldsymbol{T}}\boldsymbol{X})\boldsymbol{v} &= (1 - \frac{\eta}{N}\lambda)\boldsymbol{v}. \end{aligned}$$

Así, los valores propios de $\boldsymbol{A}$ son iguales a $1 - \frac{\eta}{N}\lambda_i$, siendo λ_i un valor propio de $\boldsymbol{X^T X}$ y η es la tasa de aprendizaje. Entonces, como $\eta > 0$ y $\lambda \geq 0$ para todo valor propio λ de $\boldsymbol{A}$ se tiene que $\frac{\eta}{N}\lambda \geq 0$. Por lo tanto, del resultado de la ecuación (4.12) se debe cumplir:

$$|1 - \frac{\eta}{N}\lambda| < 1. \tag{4.35}$$

Por propiedades de la función valor absoluto, de la ecuación anterior se puede escribir:

$$\begin{aligned} -1 < 1 - \frac{\eta}{N}\lambda_i < 1, \\ \Leftrightarrow -2 < -\frac{\eta}{N}\lambda_i < 0, \\ \Leftrightarrow 2 > \frac{\eta}{N}\lambda_i. \end{aligned}$$

Por último se tiene que:

$$\eta < \frac{2N}{\lambda_i} \tag{4.36}$$

para todo valor propio λ_i de $\boldsymbol{X^T X}$. Finalmente, como se debe asegurar que se cumpla la ecuación (4.36) es necesario buscar cotas superiores para el conjunto de valores propios de $\boldsymbol{X^T X}$. Por lo tanto, se propone dos cotas posibles: por un lado λ_{max}, es decir, el valor propio máximo, y por otro lado la traza de $\boldsymbol{X^T X}$ la cual se define de la siguiente manera:

$$\lambda_T = tr(\boldsymbol{X^T X}) = \lambda_1 + ... + \lambda_M.$$

Nótese que

$$\lambda_{max} < \lambda_T \Leftrightarrow \frac{2N}{\lambda_T} < \frac{2N}{\lambda_{max}}$$

Así, se propone dos cotas superiores posibles para la tasa de aprendizaje: $\frac{2N}{\lambda_{max}}$ y $\frac{2N}{\lambda_T}$, ambas válidas de acuerdo con la condición de la ecuación (4.36).

Para concluir, en este capítulo se formalizó matemáticamente los valores posibles de la tasa de aprendizaje del algoritmo de gradiente descendente. La validación de esta cotas teóricas se discutirá más adelante. Cabe resaltar que la derivación matemática se realizó principalmente para problemas de regresión por lo cual vale la pena revisar todo este estudio para problemas de clasificación.

Capítulo 5

La Unidad NeuroVascular Artificial (UNVA)

En el capítulo 2, se resumieron los principios biológicos de los cuales se parte para formular los modelos de unidades neuronales artificiales como los presentados en el capítulo 3 de esta tesis. Ahora que ya se presentó los fundamentos biológicos y mátematicos, en este capítulo se presenta la propuesta modelo de neurona artificial llamado Unidad Neuro-Vascular Artificial (UNVA)

Anteriormente, se describió el acoplamiento neurovascular como un comportamiento de las neuronas tal que cuando se activan hay un incremento en el consumo de energía. La energía es producida, seguido a la actividad neuronal, por la llegada de glucosa y oxígeno de pequeños vasos sanguíneos que se dilatan (hiperemia funcional) permitiendo un mejor flujo sanguíneo cerebral (CBF). Además, en el artículo de Huneau et al. [15] se presenta brevemente modelos descriptivos y explicativos desarrollados para neuroimagen funcional. Sin embargo, si bien existen técnicas ópticas para medir el nivel de oxigenación en la sangre como las imágenes de resonancia magnética (fMRI de sus siglas en inglés) a partir de la activación de regiones del cerebro, estas técnicas no son muy buenas para explorar a un nivel más profundo la estructura del cerebro. A pesar de esto, los métodos que miden la oxigenación en la sangre miden indirectamente la hiperemia funcional. Los modelos presentados en [15] sirven para proponer un primer modelo matemático que explique la relación entre la actividad eléctrica de las neuronas y los cambios del flujo sanguíneo cerebral.

De los 7 modelos presentados en [15], el artículo que presenta un modelo matemático que mejor se ajusta al objetivo de esta tésis es el de Riera et al. [31]. En el trabajo de Riera se propone un modelo dividido en 3 componentes principales, cada una con un sistema de ecuaciones diferenciales:

- El primer sistema de ecuaciones describe las variaciones temporales de los potenciales de la membrana en el soma de las neuronas. Por lo tanto, este sistema caracteriza los cambios eléctricos de la neurona.
- El segundo sistema describe el desencadenamiento de los movimientos vasculares causados por el requerimiento energético de las neuronas al activarse.

- El tercer sistema de ecuaciones explica los cambios vasculares con modelo extendido del globo [13].

En consecuencia, dado que el segundo sistema es el que mejor describe la relación entre el comportamiento eléctrico de la neurona y los cambios vasculares asociados al requerimiento energético debido a la activación neuronal, se utilizó ese sistema para diseñar la unidad vascular de la Unidad Neuro Vascular Artificial.

En el segundo sistema de ecuaciones propuesto en [31] hay 4 ecuaciones diferenciales de las cuales 2 nos interesan. Con estas 2 ecuaciones se modela la evolución del flujo sanguineo cerebral incluyendo también la demanda energética:

$$\frac{ds(t)}{dt} = (u(t-\tau_h) - u_0) - \frac{s(t)}{\tau_1} - \frac{f(t)-1}{\tau_2}, \tag{5.1}$$

$$\frac{df(t)}{dt} = s(t), \tag{5.2}$$

donde f es el flujo sanguineo cerebral, s es la derivada de f con respecto al tiempo, u_0 es la demanda básica de energía en reposo y $u(t)$ la demanda energética en el instante t. Note que τ, τ_1 y τ_2 son unas constantes genéricas del modelo pero la que más nos interesa es la del retardo en $u(t-\tau_h)$. En efecto, hay que tener en cuenta que la dilatación (o contracción) de capilares y vasos sanguineos es un proceso mecánico lento que toma tiempo comparado con la velocidad de la corriente electrica que se propaga mucho más rápido a lo largo de las neuronas.
Así, reemplazando la ecuación (5.2) en la ecuación (5.1) y simplificando las constantes se puede generalizar estas ecuaciones en el siguiente modelo:

$$f = c_1(u(t-\tau_h) - u_0) + c_2\frac{d^2f}{dt^2} + c_3\frac{df}{dt} + c_4. \tag{5.3}$$

Cabe resaltar que el modelo que se quiere plantear es un modelo en tiempo discreto computacional que simule una señal en tiempo discreto, no en tiempo continuo. Luego, si discretizamos la señal del flujo f y teniendo encuenta que el equivalente de la derivada de una señal en tiempo discreto se define de la siguiente manera:

$$f'[n] = \frac{f[n] - f[n-1]}{n-(n-1)} = f[n] - f[n-1], \tag{5.4}$$

entonces, a partir de la ecuación (5.3) se parte para proponer la siguiente ecuación que modela el flujo sanguineo de la unidad vascular:

$$\begin{aligned} f[n] = c_1(u[n-1] - u_0[n]) + \\ c_2(f[n] - 2f[n-1] + f[n-2]) + c_3(f[n] - f[n-1]) + c_4 \end{aligned}$$

que simplificando de nuevo se llega a:

$$f[n] = k_1(u[n-1] - u_0[n]) - k_2 f[n-1] + k_3 f[n-2] + k_4, \tag{5.5}$$

con $k_1, k_2, k_3, k_4 \in \mathbb{R}$.
La siguiente figura es un esquema de la Unidad Neurovasucular Artificial que incluye la estructura del perceptrón junto con la componente vascular inspirada en la ecuación (5.5).

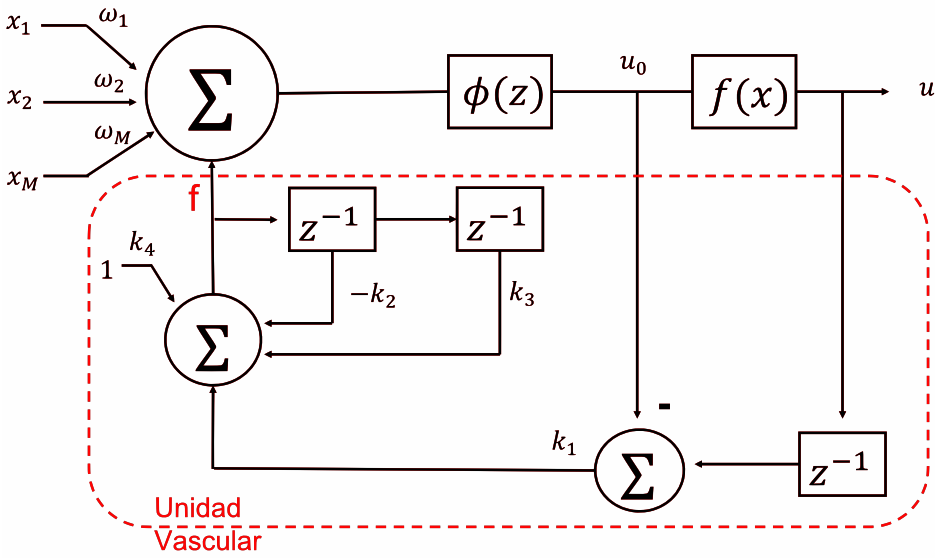


FIGURA 5.1: Esquema de Unidad Neurovascular Artficial (UNVA)

Por un lado, en la figura 5.1 se puede observar en la parte superior la arquitectura de la neurona artificial generalizada tal y como se describió en el capítulo 3. Adicionalmente, se interpretó la demanda básica de energía en reposo u_0 como el valor justo antes de la función de activación, así como se interpretó la demanda energética u como la salida de la neurona artificial ya que la activación eléctrica de la neurona se relaciona directamente con su demanda metabólica para funcionar correctamente. Por otro lado, en la parte inferior se representa la unidad vascular de la UNVA a partir de la ecuación (5.5). En la unidad vascular hay bloques de retardo para ilustrar que se toma términos con instantes anteriores a un instante n como $u[n-1]$, $f[n-1]$ y $f[n-2]$. Además, se muestran los coeficientes $k_i, i \in \{1,2,3,4\}$ de esta nueva arquitectura, los cuales al multiplicarse por sus términos respectivos permiten calcular el flujo sanguíneo cerebral f en el instante n. Así, el valor de la estimación $u[n]$ para un vector de entrada con la UNVA se puede reescribir de la siguiente manera:

$$\boldsymbol{u}[n] = h(\boldsymbol{z}[n]), \tag{5.6}$$

$$\boldsymbol{z}[n] = \boldsymbol{X}\boldsymbol{\omega} + \boldsymbol{f}[n] \quad , \quad h(x) = f(\phi(x)),$$

Por último, cabe resaltar que los pesos k_1, k_2, k_3 y k_4 deben actualizarse al igual que los coeficientes del perceptrón por medio de alguna regla de aprendizaje ya que el flujo es una entrada más en el modelo de la neurona artificial, luego el aprendizaje de los pesos debe ser tal que se minimice el error lo más posible. A continuación, se detalla el algoritmo de aprendizaje propuesto para la UNVA.

5.1. Algoritmo de aprendizaje de la UNVA

Antes de explicar el algoritmo de aprendizaje de la UNVA, recordemos el funcionamiento de las reglas de aprendizaje estudiadas. El objetivo final de estas reglas es modificar los pesos $\boldsymbol{\omega}$ tal que se logre minimizar una función de error o de costo de predicción.
Por un lado, existe la regla de aprendizaje del perceptrón la cual fue diseñada para problemas de clasificación dado que se usa en arquitecturas de neuronas artificiales cuya salida sólo puede ser 1 o 0 (originalmente 1 o -1). La actualización de los coeficientes se describe en las ecuaciones (3.4) y (3.5) , las cuales se recuerdan a continuación:

$$\begin{aligned} \boldsymbol{\omega} &\coloneqq \boldsymbol{\omega} + \Delta\boldsymbol{\omega}, \\ \Delta\boldsymbol{\omega} &= \eta \boldsymbol{X}^{\boldsymbol{T}}(\boldsymbol{y} - \hat{\boldsymbol{y}}). \end{aligned} \tag{5.7}$$

Nótese que en la ecuación (5.7) se agregó la tasa de aprendizaje η para generalizar la tasa a la cual se actualizan los pesos.
Por otro lado, se tiene el gradiente descendente, el cual es un algoritmo muy útil en optimización convexa debido a que permite encontrar el mínimo de una función a partir de la dirección opuesta del gradiente de la función. Las ecuaciones que describen el aprendizaje de los pesos $\boldsymbol{\omega}$ a partir de esta regla son la (3.10) y (3.11) las cuales se muestran a continuación:

$$\omega_i \coloneqq \omega_i + \Delta\omega_i, \tag{5.8}$$

$$\Delta\omega_i = -\eta \frac{\partial J(\boldsymbol{\omega})}{\partial \omega_i}. \tag{5.9}$$

Teniendo en cuenta ambos algoritmos de aprendizaje, a continuación se explica varios aspectos a tener presente para formular el algoritmo de aprendizaje de la UNVA.

5.1.1. Adaptaciones a partir de la regla del perceptrón

La arquitectura UNVA cuenta con unos coeficientes especiales k_1, k_2, k_3 y k_4 que la diferencian de las neuronas artificiales tradicionales. Sin embargo, hay que tener presente que se debe asegurar la estabilidad del aprendizaje con la nueva entrada f correspondiente al flujo sanguíneo. Para cumplir con esta estabilidad, se propone las siguientes modificaciones:

1. La estructura de la matriz $\boldsymbol{X}$ cambia en cada época n ya que se agrega nuevas columnas asociadas a la diferencia $u[n] - u_0[n-1]$ y al valor del flujo f en el instante $n-2$ y $n-1$. Luego, si se inicializa aleatoriamente los valores de los coeficientes k_1, k_2, k_3 y k_4 entonces la matriz $\boldsymbol{X}$ es ahora $\boldsymbol{X}[n] \in \mathbb{R}^{N\times(M+4)}$ y

es igual a:

$$\boldsymbol{X}[n] = \begin{bmatrix} x_{11} & \dots & x_{1M} & u_1[n-1] - u_1^0[n] & f_1[n-2] & f_1[n-1] & 1 \\ \vdots & \ddots & \vdots & \vdots & \vdots & \vdots & \vdots \\ x_{N1} & \dots & x_{NM} & u_N[n-1] - u_N^0[n] & f_N[n-2] & f_N[n-1] & 1 \end{bmatrix}.$$

Como podemos ver, ahora la matriz $\boldsymbol{X}[n]$ está dividida en dos partes que corresponden por un lado a las primeras M columnas con datos estáticos, y por otro lado, a los valores $u[n-1] - u^0[n]$, $f[n-2]$ y $f[n-1]$ para cada una de las N filas. Por último, la columna final de $\boldsymbol{X}[n]$ es una columna de unos la cual es multiplicada por peso k_4 del flujo. Por lo tanto, al ser igual a la columna de sesgo de la matriz original $\boldsymbol{X}$ se movió esa columna al final de la matriz para que sea más fácil de visualizar las 4 columnas que construyen el flujo. De esta forma, la entrada de la función h de acuerdo con la ecuación (5.6) es el vector $\boldsymbol{z}[n]$ igual a:

$$\boldsymbol{z}[n] = \boldsymbol{X}[n]\boldsymbol{\theta}, \tag{5.10}$$

y el vector $\boldsymbol{\theta}$ es igual a:

$$\boldsymbol{\theta} = \begin{bmatrix} \theta_1 & \dots & \theta_M & \theta_{M+1} & \theta_{M+2} & \theta_{M+3} & \theta_{M+4} \end{bmatrix}^T, \tag{5.11}$$

$$= \begin{bmatrix} \omega_1 & \dots & \omega_M & k_1 & k_2 & k_3 & k_4 \end{bmatrix}^T. \tag{5.12}$$

2. Recordemos de la sección 4.3 que para todo valor propio λ_i de la matriz $\boldsymbol{X}^T\boldsymbol{X}$ se debe cumplir:

$$\eta < \frac{2N}{\lambda_i} \quad , \quad \forall i \in \{1, ..., M\}.$$

Además, se propuso dos posibles cotas para asegurar la convergencia del algoritmo de aprendizaje: $\frac{2N}{\lambda_{max}}$ y $\frac{2N}{\lambda_T}$, con λ_{max} el máximo de los valores propios y λ_T la traza de la matriz $\boldsymbol{X}^T\boldsymbol{X}$. Aunque estos resultados se obtuvieron para el descenso del gradiente, se espera que para el algoritmo del perceptrón la condición se también se cumpla. Luego, teniendo en cuenta que la matriz $\boldsymbol{X}$ es ahora una matriz dinámica $\boldsymbol{X}[n]$ se debe validar que la condición se cumpla en cada una de las épocas considerando los valores propios de la matriz $\boldsymbol{X}[n]^T\boldsymbol{X}[n]$. De esta forma, la condición de convergencia del algoritmo de aprendizaje es la siguiente:

$$\eta < \frac{2N}{\lambda_i[n]} \quad , \quad \forall n \geq 0, \ i \in \{1, ..., M\}. \tag{5.13}$$

Así mismo, las cotas propuestas para la tasa de aprendizaje son $\frac{2N}{\lambda_{max}[n]}$ y $\frac{2N}{\lambda_T[n]}$ por lo cual, si en una iteración del algoritmo de aprendizaje, la tasa de aprendizaje es mayor que la cota, entonces la tasa debe reajustarse antes de que continue el aprendizaje de los pesos de la arquitectura.

3. Recordemos que para calcular $f[n]$ se necesita $f[n-1]$ y $f[n-2]$, por lo tanto, se necesita las condiciones iniciales $f[0]$ y $f[1]$. Para facilitar el cálculo

de la estimación $u[n]$, se puede inicializar las condiciones iniciales del flujo $f[0] = f[1] = 0$. De esta manera, el algoritmo de aprendizaje de la UNVA para clasificación es una modificación del algoritmo del perceptrón y se define en el algoritmo 3:

Algoritmo 3: Entrenamiento de UNVA con regla de perceptrón

Result: Vector de pesos $\boldsymbol{\theta}$ entrenados
initialization;
$\theta =$ valor_aleatorio, $\forall \theta \in \boldsymbol{\theta}$;
Definir: MAX_ITER, ϵ ; $n = 0$;
$\boldsymbol{z} = \boldsymbol{X}[n]\boldsymbol{\theta}$;
$\boldsymbol{u}[n] = (u_1[n], ..., u_N[n])^T, u_i[n] = h_\theta(z_i)$;
$e = J(\boldsymbol{\theta})$;
while $n < MAX_ITER$ ***and*** $e > \epsilon$ **do**
  Calcular cota $c[n]$ tal que se cumpla la condición de la ecuación (5.13);
  if $n < 2$ **then**
    $f[n] = 0$;
  else
    Continuar
  $\eta = 0.65c[n]$;
  $\Delta\boldsymbol{\theta} = \eta \boldsymbol{X}[n]^T(\boldsymbol{y} - \boldsymbol{u}[n])$;
  $\boldsymbol{\theta} = \boldsymbol{\theta} + \Delta\boldsymbol{\theta}$;
  $e = J(\boldsymbol{\theta})$;
  $n = n + 1$

Note que a la taza de aprendizaje η se asigna el valor $0{,}65 \times c[n]$ dado que se encontró de forma empírica que ese valor produce una convergencia acelerada. Del algoritmo 3 vale la pena resaltar los siguientes cambios:

- Se agregó las condiciones iniciales para las dos primeras épocas tal que

$$f[0] = f[1] = 0$$

  En consecuencia, los vectores de estimaciones $\boldsymbol{u}[0]$ y $\boldsymbol{u}[1]$ son iguales al producto de la parte estática de la matriz $\boldsymbol{X}[n]$ con el vector de pesos $\boldsymbol{\omega}$. Por lo tanto, en las dos primeras épocas, la estimación de la UNVA se calcula como la suma ponderada del perceptrón y desde la tecera época en adelante, el cálculo de la estimación cambia.

- Dado que se debe asegurar que la tasa de aprendizaje no sea mayor a una cota, la cual cambia por época debido a que la matriz $\boldsymbol{X}[n]$ cambia también, se debe buscar una forma de disminuir la tasa de aprendizaje. Por ejemplo, una posibilidad es dividir la tasa por 2 hasta que sea menor que la cota, si inicialmente no lo era. Este procedimiento se ejecuta por época de ser necesario, garantizando que la tasa disminuya sin ser menor o igual que 0. Sin embargo, una desventaja

de este procedimiento es que es posible que el decremento de η sea demasiado grande en una iteración por lo cual puede que el entrenamiento se demore más desde la siguiente iteración. Es por esto que vale la pena considerar algún tipo de decremento de la constante η que se asegure la condición de convergencia y que no afecte considerablemente el rendimiento de la arquitectura. Para esto se determinó que una posibilidad es multiplicar la taza de aprendizaje por una constante menor que uno. Empíricamente se obtuvo que un valor ideal de esta constante es 0.65 el cual se discute en el capítulo 6.

Ahora que se definió el algoritmo de aprendizaje de la UNVA a partir de la regla del perceptrón, vale la pena considerar la regla de aprendizaje de la UNVA a partir del descenso del gradiente.

5.1.2. Adaptaciones a partir del descenso del gradiente

En el caso del descenso del gradiente, la ecuación que define la regla de aprendizaje de los pesos de la arquitectura es:

$$\omega := \omega - \vec{\nabla} J(\omega). \tag{5.14}$$

Para la arquitectura UNVA, la regla se mantiene igual, sin embargo hay que tener en cuenta el valor de las derivadas parciales con respecto a los coeficientes k_i.

Considere por ejemplo que la arquitectura UNVA se usa en un problema de regresión lineal. Esto implica que las funciones

$$f(x) = \phi(x) = x, \tag{5.15}$$

y por ende la función de activación es igual a:

$$h(x) = x. \tag{5.16}$$

Además, de acuerdo a lo visto en el capítulo 3, la función de costo utilizada es el error cuadrático medio o MSE. Luego, con la UNVA, este costo es igual a:

$$J(\theta) = \frac{1}{2N}(u[n] - y)^2 = \frac{1}{2N}(\theta_1 x_1 + \ldots + \theta_M x_M + f[n] - y)^2. \tag{5.17}$$

Luego, las derivadas parciales del costo con la UNVA son iguales a:

$$\frac{\partial J}{\partial \theta_i} = \frac{1}{N}(u[n] - y)\frac{\partial u[n]}{\partial \theta_i} \quad , \quad \forall i \in \{1, \ldots, M+4\}. \tag{5.18}$$

Note que para calcular $\frac{\partial u[n]}{\partial \theta_i}$ se trata a los términos que multiplican los coeficientes k_i como constantes porque se calcularon en instantes pasados por lo cual el cálculo de la derivada parcial de $u[n]$ se simplifica tal que la derivada del costo J en

función de cada uno de los pesos es igual a:

$$\frac{\partial J(\boldsymbol{\theta})}{\partial \theta_i} = \begin{cases} \frac{1}{N}(u[n]-y)x_i & \forall i \leq M \\ \frac{1}{N}(u[n]-y)(u[n-1]-u_0[n]) & \text{si } \theta_i = k_1 \\ \frac{1}{N}(u[n]-y)f[n-1] & \text{si } \theta_i = k_2 \\ \frac{1}{N}(u[n]-y)f[n-2] & \text{si } \theta_i = k_3 \\ \frac{1}{N}(u[n]-y) & \text{si } \theta_i = k_4 \end{cases} . \tag{5.19}$$

De esta forma, ya podemos definir el algoritmo de aprendizaje de la UNVA partiendo de la regla del descenso del gradiente:

Algoritmo 4: Entrenamiento de UNVA con descenso del gradiente

Result: Vector de pesos $\boldsymbol{\theta}$ entrenados
initialización;
$\theta =$ valor_aleatorio, $\forall \theta \in \boldsymbol{\theta}$;
Definir: η;
$n = 0$;
$\boldsymbol{z} = \boldsymbol{X}[n]\boldsymbol{\theta}$;
$\boldsymbol{u}[n] = (u_1[n], ..., u_N[n])^T, u_i[n] = h_\theta(z_i)$;
$e = J(\boldsymbol{\theta})$;
while $e \geq \epsilon$ **do**
  if $n < 2$ **then**
    $f[n] = 0$;
  else
    Continuar
  $e = J(\boldsymbol{\theta})$;
  $\boldsymbol{\theta} = \boldsymbol{\theta} - \eta \vec{\boldsymbol{\nabla}} J(\boldsymbol{\theta})$;
  while $\eta \geq \frac{2N}{\lambda_T[n]}$ **do**
    Disminuir η;
  $n = n + 1$

Por último, así como se construyó la función de costo para regresión lineal de la UNVA también se puede construir la UNVA para problemas de clasificación. En este caso, a comparación de la regresión lineal, la función de enlace ϕ es igual a la función logística y además se utiliza como función de costo la entropía cruzada binaria de la ecuación (3.15) y se reemplaza $h_{\boldsymbol{\omega}}(\boldsymbol{x_i})$ por $u[n]$. Sin embargo, en este caso se debe analizar el sistema resultante y realizar menos simplificaciones. Esto no será parte de esta tesis debido a las restricciones temporales.

En conclusión, en este capítulo se definió la arquitectura llamada Unidad Neuro-Vascular Artificial (UNVA), la cual nace de un modelo del acoplamiento neurovascular y de algunas arquitecturas de neuronas artificiales. Adicionalmente, se redefinió las condiciones de la tasa de aprendizaje para convergencia del algoritmo de aprendizaje utilizado y se presentaron adaptaciones de la UNVA para aplicarla en problemas de clasificación y regresión. Ahora que se explicó la teoría detrás de la arquitectura propuesta, en el siguiente capítulo se probará la UNVA en distintos tipos de problemas y se analizará su comportamiento comparándola con las arquitecturas de unidades neuronales ya existentes.

Capítulo 6

Resultados

En este capítulo se presenta los resultados obtenidos de simulaciones del entrenamiento de las neuronas para validar los resultados teóricos acerca de las cotas superiores de la tasa de aprendizaje. También se evalúa los resultados de la UNVA al probarla con conjuntos de datos generados manualmente y con conjuntos de datos ya establecidos como los del repositorio de aprendizaje de máquina UCI [12].

6.1. Validación de resultados de cota de tasa de aprendizaje

A continuación se presenta la metodología con la que se valida los valores posibles que puede tomar la tasa de aprendizaje η. Para realizar las pruebas y simular un problema de regresión, se consideraron dos conjuntos de datos: el primero son datos generados manualmente y el segundo es a partir del conjunto de datos de flores Iris.

6.1.1. Resultados con datos manuales

Para el primero se creó una matriz $\boldsymbol{X} \in \mathbb{R}^{100\times 6}$ tal que la primera columna es un vector columna de 1s, es decir, es la columna de sesgo y las demás columnas son vectores aleatorios distribuidos aleatoriamente. Además, el vector de pesos $\boldsymbol{\omega}$ se inicializó aleatoriamente y se creó un vector $\boldsymbol{y}$ igual a:

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\omega} + \boldsymbol{\epsilon} \ , \ \boldsymbol{\epsilon} \in \mathbb{R}^{100} \tag{6.1}$$

Nótese que el vector $\boldsymbol{y}$ se creó de esta manera para asegurar que tenga una correlación lineal con las columnas de la matriz de datos $\boldsymbol{X}$ pero que la relación no sea tan directa al sumar el vector de ruido $\boldsymbol{\epsilon}$.
Segundo, los datos se separaron tal que 80 % de los datos son de entrenamiento y 20 % de prueba. Adicionalmente, se predefinió una cantidad de épocas de entrenamiento para que se controle manualmente si el algoritmo de entrenamiento presenta un sobreajuste (o subajuste) en el aprendizaje de los pesos $\boldsymbol{\omega}$. De igual modo, se definió varios valores de la tasa de aprendizaje η, incluyendo dentro de los valores posibles la cota así como también valores menores y superiores a ella.
Por último, se ejecutó el algoritmo de gradiente descendente y se analizó el comportamiento del error cuadrático medio así como la evolución de los coeficientes para

las distintas tasas de aprendizaje. Los resultados obtenidos del entrenamiento son los siguientes:

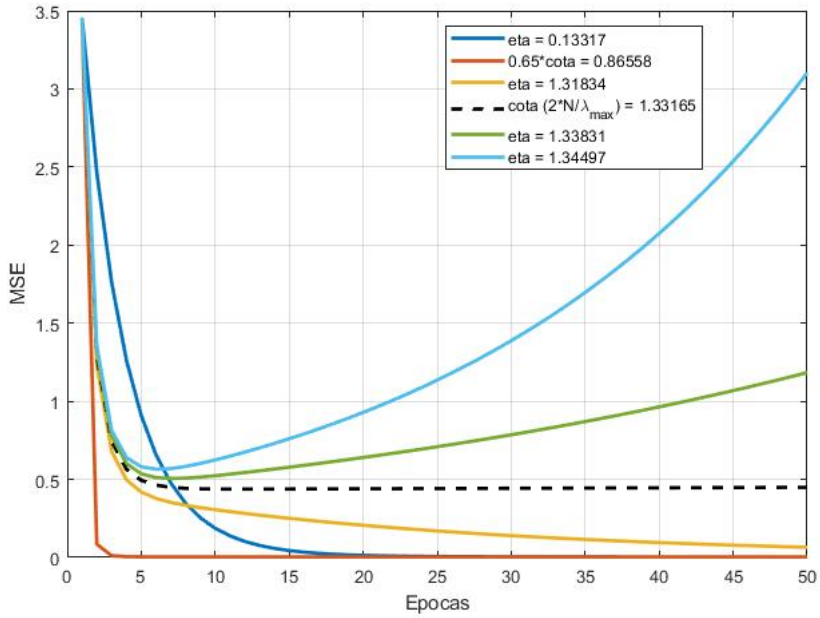


FIGURA 6.1: Evolución de MSE con múltiples tasas de aprendizaje calculadas con cota $2N/\lambda_{max}$ y con datos manuales

En la figura 6.1 se observa los resultados del error de entrenamiento con 6 tasas de aprendizaje: 3 son menores que la cota $\frac{2N}{\lambda_{max}}$ y 2 son mayores que la cota. Cabe resaltar que el valor N de la cota corresponde a la cantidad de datos de entrenamiento y no al total de datos de la matriz $\boldsymbol{X}$. Nótese que cada curva de la gráfica 6.1 tiene un comportamiento particular:

- La linea azul oscura corresponde a la menor tasa de aprendizaje η la cual es **no óptima** debido a que el error converge demasiado lento comparado con las otras curvas. Análogamente, aunque el error la curva amarilla converge inicialmente más rápido que en la curva azul oscuro, luego converge más lento por lo cual el entrenamiento también es lento.

- La linea roja corresponde a una tasa de aprendizaje **óptima** ya que es en la que más rápido converge el error. Cabe resaltar que esta tasa óptima es igual al 65 % de la cota y que este porcentaje para determinar si es una tasa óptima se obtuvo empíricamente.

- La linea punteada en negro está asociada al error calculado con la tasa de aprendizaje igual a la cota. Nótese que esta tasa no es conveniente pues aunque el error converge, no converge a un mínimo lo cual no asegura que se minimice el error y por ende las predicciones no sean tan cercanas a los valores esperados. Esto se debe a que los parámetros $\boldsymbol{\omega}$ oscilan entre valores que producen el mismo valor del costo.

- Las lineas azul claro y verde son curvas en las que la tasa es ligeramente superior a la cota (la linea verde tiene un aumento de 0.5 % con respecto a la cota) y como se ve en la gráfica 6,1 el error con estas tasas diverge, luego, estas tasas de aprendizaje no son buenas para el entrenamiento de estos modelos.

Por lo tanto, la gráfica 6.1 nos permite validar el resultado teórico sobre la cota $2N/\lambda_{max}$. Es importante recordar que esta cota no es tan restrictiva como la cota $2N/tr(X^TX)$ por lo cual es probable que para valores ligeramente superiores a $2N/\lambda_{max}$ el error diverja y para valores ligeramente inferiores el error converja. Para validar que con una cota más estricta como $2N/tr(X^TX)$ el comportamiento del error cambia, observe la figura 6.2.

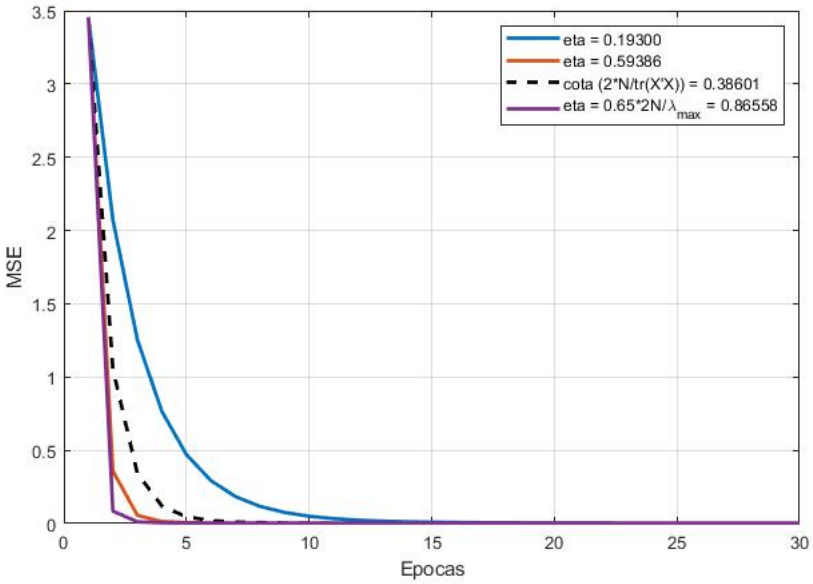


FIGURA 6.2: Evolución de MSE con múltiples tasas de aprendizaje calculadas con cota $2N/\operatorname{tr}(X^TX)$ con datos manuales

En la figura 6.2 hay 4 curvas: la azul representa la evolución del error con una tasa de aprendizaje menor a $2N/tr(\boldsymbol{X^TX})$, la negra punteada es del error calculado con la tasa η igual a $2N/tr(\boldsymbol{X^TX})$ y 2 cotas superiores a la cota calculada con la traza de $\boldsymbol{X^TX}$, las cuales aparecen en rojo, con $\eta = \frac{1}{0{,}65}2N/\operatorname{tr}(\boldsymbol{X^TX})$, y la morada con la mayor tasa de aprendizaje $(\text{con } \eta = 2N/\lambda_{max})$. Note que en el caso de η igual a la cota (linea negra), aunque el error converge en pocas iteraciones, el error converge aún más rápido con las tasas de aprendizaje superiores a la cota. En particular, nótese que la diferencia del error en la curva roja y la curva morada es muy pequeña por lo cual se considera que el valor $\eta = \frac{1}{0{,}65}\frac{2N}{tr(\boldsymbol{X^TX})}$, el cual tamién fue obtenido empíricamente, es otro valor óptimo para la tasa de aprendizaje.

Finalmente, otra información que ayuda a validar los resultados teóricos es el comportamiento de los pesos $\boldsymbol{\omega}$ al entrenarse. La figura 6.3 muestra su evolución en la fase de entrenamiento.

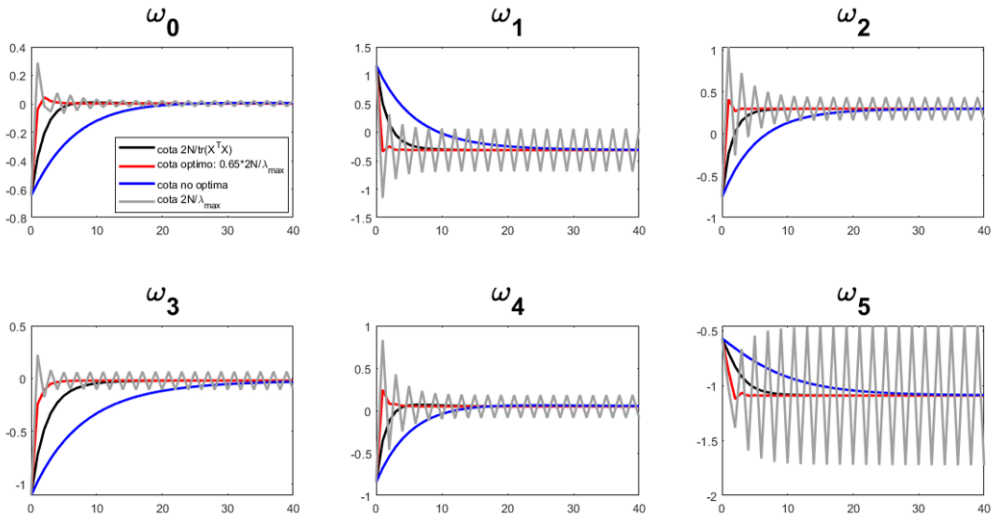


FIGURA 6.3: Evolución de coeficientes para distintas tasas de aprendizaje η con datos manuales

En la figura 6.3 se muestra el comportamiento de cada uno de los coeficientes de la regresión lineal con distintos valores de la tasa de aprendizaje, incluyendo tasas iguales a las 2 cotas propuestas y otras 2 con valor óptimo y no óptimo. Primero, podemos ver que con la tasa de aprendizaje igual a $\frac{2N}{\lambda_{max}}$ los coeficientes oscilan y se mantienen acotados. Este comportamiento es el esperado pues no es una cota muy pequeña y por ende el gradiente descendente no puede encontrar el mínimo de la función de costo pero tampoco aumenta el error de predicción.
Segundo, vemos que la curva roja converge rápidamente a un error mínimo por lo cual se consideró que la cota $0{,}65\frac{2N}{\lambda_{max}}$ es un valor óptimo para la tasa de aprendizaje.
Tercero, la curva negra representa la evolución de los coeficientes con $\eta = \frac{2N}{tr(\boldsymbol{X^T X})}$. Como es una cota más estricta para la tasa de aprendizaje, vemos que los pesos convergen a una lenta velocidad.
Finalmente, se dibujó en azul la curva asociada con una tasa de aprendizaje no óptima, demasiado pequeña, para comparar la velocidad de convergencia de los coeficientes con respecto al las otras tasas η. En consecuencia, vemos que el comportamiento de los coeficientes corresponde al comportamiento esperado de acuerdo con los resultados teóricos obtenidos en el capítulo 4.
Por último, validemos los resultados con los datos de prueba. Para comparar los resultados, se calcularon tres métricas de regresión para distintos multiplicadores de la cota:

1. ***SSE***: La Suma de Errores Cuadrados (SSE por sus siglas en inglés) representa la diferencia cuadrada entre la estimación del modelo y la variable respuesta [18], la cual es igual a:

$$SSE = \sum_{i=1}^{N} (\hat{y}_i - y)^2. \tag{6.2}$$

2. $\boldsymbol{R^2}$: También llamado *coeficiente de determinación*, el R^2 representa la proporción de la variación total de los valores y_i explicada por los predictores del modelo [18] y es igual a:

$$R^2 = \frac{\sum_{i=1}^{N}(\hat{y}_i - \bar{y})^2}{\sum_{i=1}^{N}(y_i - \bar{y})^2} = 1 - \frac{\sum_{i=1}^{N} e_i^2}{\sum_{i=1}^{N}(y_i - \bar{y})^2}, \tag{6.3}$$

con $e_i = (\hat{y}_i - y_i)^2$. Nótese que usualmente el valor de R^2 debe estar entre 0 y 1 tal que entre más cercano a 1 mejor es el modelo para predecir.
Nota: R^2 puede ser negativo si la suma de los errores cuadrados es mayor que la varianza de la variable respuesta $\boldsymbol{y}$.

3. $\boldsymbol{R^2}$ **ajustado**: cuando se considera más de un sólo predictor en regresión lineal el R^2 no es suficientemente preciso dado que al tener más datos con más columnas de la matriz $\boldsymbol{X}$ el coeficiente de determinación aumenta [10]. Por lo tanto, para compensar la cantidad de información adicional se calcula el R^2 ajustado:

$$R^2_{adj} = 1 - \frac{N-1}{N-M-1}(1 - R^2), \tag{6.4}$$

tal que N es la cantidad de filas y M cantidad de predictores de la matriz $\boldsymbol{X}$.

En la tabla 6.1 se resume los resultados con las métricas mencionadas:

η	**Multiplicador de cota** $2N/\lambda_{max}$	SSE	R^2	R^2 **ajustado**
0.13317	0.1	0.26061	0.99462	0.99319
0.86558	0.65	0.25718	0.99469	0.99328
1.3183	0.99	2.8256	0.94167	0.92612
1.3317	1	19.733	0.59267	0.48404
1.3383	1.005	52.996	-0.093975	-0.3857
1.345	1.01	141.6	-1.9231	-2.7026

CUADRO 6.1: Resultados con datos de pruebas y datos generados manualmente

Vemos que la tasa de aprendizaje con la que se obtuvo las mejores predicciones es con la que es igual a $0{,}65 \times 2N/\lambda_{max}$ pues se obtuvo el menor SSE y el mayor R^2 y R^2 ajustado. Además, aunque los valores para esta tasa de aprendizaje son muy cercanos a los de la tasa de aprendizaje igual a $0{,}1 \times 2N/\lambda_{max}$ cabe recordar que de acuerdo con la figura 6.1 el MSE converge al mínimo en aproximadamente 5 épocas con $\eta = 0{,}65 \times 2N/\lambda_{max}$ mientras que con $\eta = 0{,}1 \times 2N/\lambda_{max}$ el error converge al mínimo en aproximadamente 25 épocas por lo cual se considera que la tasa de aprendizaje óptima es igual a $\eta = 0{,}65 \times 2N/\lambda_{max}$. Por otro lado, note que para los valores de η mayores que la cota, la predicción es tan mala que el SSE es enorme, aproximádamente igual a 53 o mayor, comparado con las tasas de aprendizaje

menores que la cota, cuyo SSE es menor a 3, y además los coeficientes R^2 y R^2 son negativos lo cual valida que el error es tan grande que supera la distancia cuadrada de y_i a $\bar{y}$.

6.1.2. Resultados con datos Iris

El conjunto de datos Iris es quizás el conjunto de datos más conocido para aplicar algoritmos de aprendizaje de máquina. Estos datos fueron medidos por Ronal Fisher en 1936 en el cual el autor tomó 150 muestras de flores de la especie Iris. En los datos hay 3 variedades de flores Iris: virginica, versicolor y setosa, 50 datos por variación. Adicionalmente, se midió el ancho del pétalo, el largo del pétalo, el ancho del sépalo y el alto del sépalo, cada una de estas medidas en centímetros [12]. Aunque este conjunto de datos es ideal para un problema de clasificación, se ignoró la etiqueta que indica la variedad de la flor para concentrarse únicamente en las variables continuas. De esta forma, se puede plantear un problema de regresión en el que se utiliza 3 de las variables continuas como predictores y la variable restante es la variable respuesta del problema. Antes de continuar con los resultados observemos las correlaciones entre las 4 variables medidas, estas se muestran en la figura 6.4:

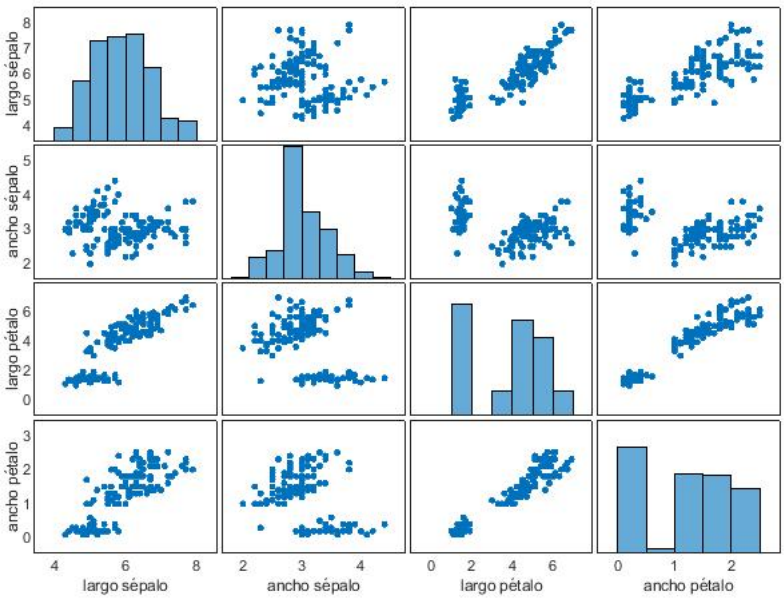


FIGURA 6.4: Dispersiones entre variables de datos Iris

En la figura 6.4 vemos que hay una correlación positiva entre el ancho del pétalo y el resto de variables. Luego, para el problema de regresión formulado se generó la matriz $\boldsymbol{X}$ con 150 filas y 3 columnas tal que las 3 columnas corresponden al largo del sépalo, el ancho del sépalo y el largo del pétalo y la variable $\boldsymbol{y}$ es el ancho del pétalo. Ahora veamos el error con distintas tasas de aprendizaje.

En la figura 6.5 vemos cuatro curvas asociadas a 4 tasas de aprendizaje distintas.

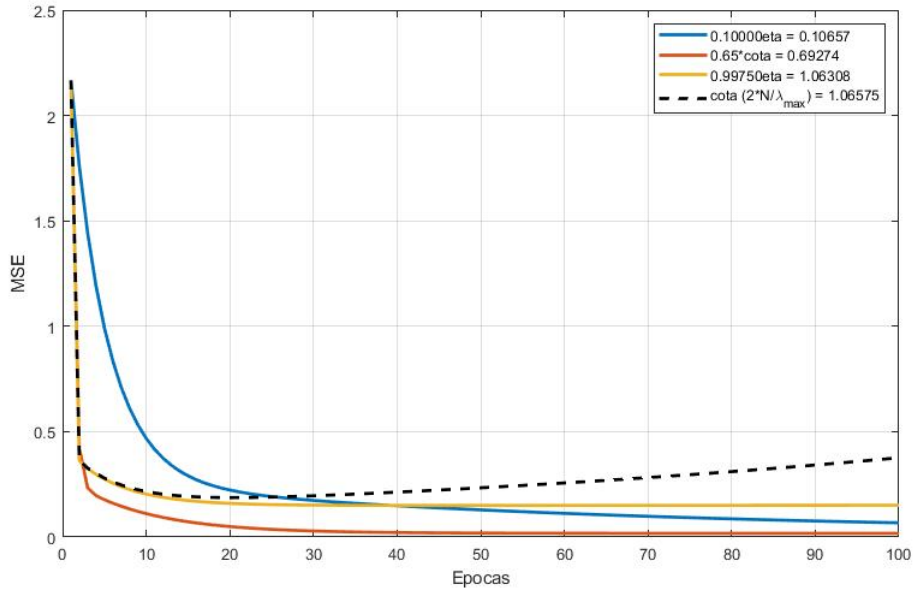


FIGURA 6.5: Evolución de MSE con múltiples tasas de aprendizaje para cota $2N/\lambda_{max}$ con datos Iris

.

- La curva azul representa el error con el menor η igual a $0{,}1 \times 2N/\lambda_{max}$ y así como ocurrió en el ejemplo con los datos manuales, este es el caso en el que el error converge más lento al mínimo.
- La curva roja es el error con $\eta = 0{,}65 \times 2N/\lambda_{max}$ y vemos nuevamente que es la curva que más rápido converge al mínimo.
- La curva amarilla representa el error con $\eta = 0{,}9975 \times 2N/\lambda_{max}$ y vemos que con este valor el MSE no converge al mínimo y se estanca en 0.2. Nótese que esto ocurre para un valor ligeramente menor a la cota teórica (debido a que η es igual a 0.9975 veces la cota teórica) mientras que cuando $\eta = 2N/\lambda_{max}$ (curva negra punteada) el error ya diverge. Esto se debe probablemente a errores numéricos al calcular la cota teórica lo que hace que el valor de la cota teórica para η sea ligeramente distinto a la cota numérica.

Ahora analicemos el comportamiento del error calculado con la cota $2N/tr(\boldsymbol{X}^T\boldsymbol{X})$.

En la figura 6.6 hay 4 curvas tales que la curva azul es la única en la que η es menor a la cota $2N/tr(\boldsymbol{X}^T\boldsymbol{X})$, las demás corresponden a $\eta = 2N/tr(\boldsymbol{X}^T\boldsymbol{X})$, la curva negra punteada, $\eta = \sqrt{2}\left(2N/tr(\boldsymbol{X}^T\boldsymbol{X}\right)$ (curva roja) y $\eta = 1{,}5049 \times 2N/tr(\boldsymbol{X}^T\boldsymbol{X})$ (curva morada). Vemos por un lado que casi no hay diferencia entre la curva roja y la curva negra y que en ambos casos el MSE converge a 0. Sin embargo, para la curva morada en la que el error se estanca y no converge a 0, cabe resaltar que el multiplicador de la cota (igual a 1.50490) el cual fue obtenido empíricamente, es muy cercano al multiplicador $1/0{,}65 \approx 1{,}5385$ el cual se analizó en la figura 6.2

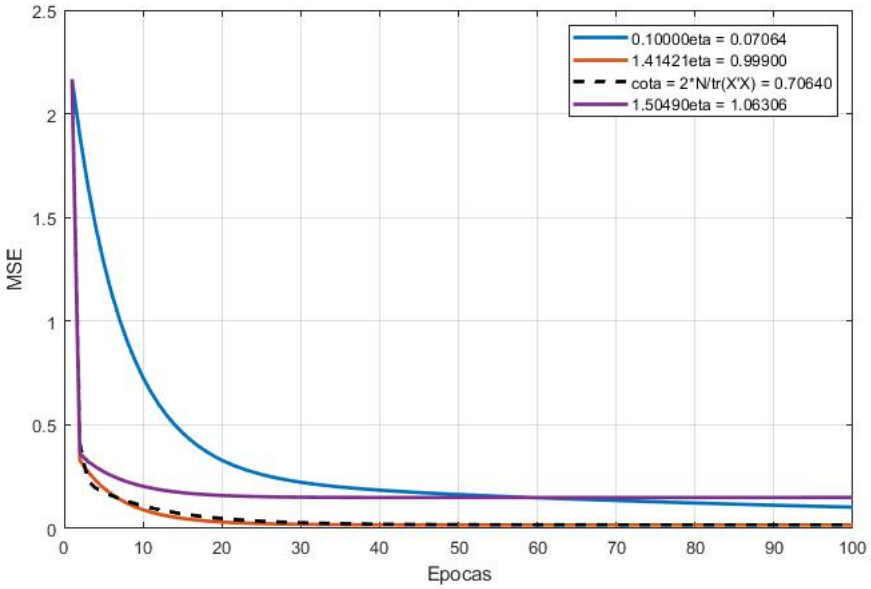


FIGURA 6.6: Evolución de MSE con múltiples tasas de aprendizaje calculadas con cota $2N/\operatorname{tr}(X^T X)$ con datos Iris

y con el cual se consideró que era la cota óptima. Por lo tanto, si se utiliza una tasa de aprendizaje calculada a partir de la cota $2N/\operatorname{tr}(\boldsymbol{X^T X})$ es preferible que la tasa sea menor que $\frac{1}{0{,}65} 2N/tr(\boldsymbol{X^T X})$ para evitar que el error diverja.

Por último estudiemos el comportamiento de los coeficientes en la figura 6.7.

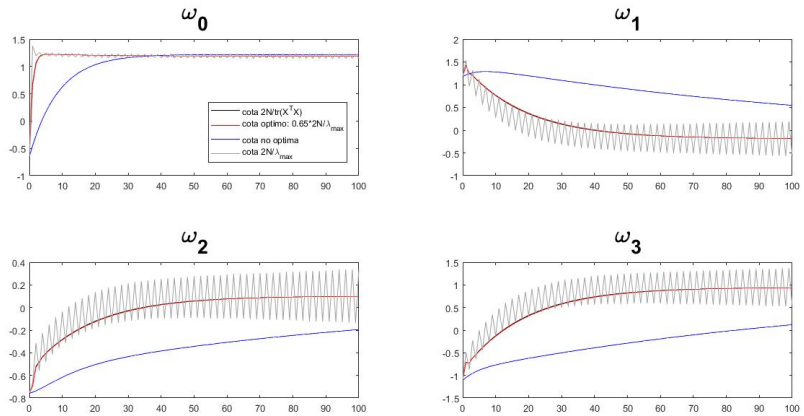


FIGURA 6.7: Evolución de coeficientes para distintas tasas de aprendizaje η con datos Iris

En la figura 6.7 vemos que los coeficientes con tasas de aprendizaje iguales a $2N/\operatorname{tr}(\boldsymbol{X^T X})$ y a $0{,}65 \times 2N/\lambda_{max}$ se comportan casi de la misma manera y convergen mucho más rápido que con la otras tasas. Por otro lado, al igual que en la

figura 6.3 vemos que cuando $\eta = 2N/\lambda_{max}$ los coeficientes oscilan lo que hace que el error no converja a 0 y por ende no se encuentra el mínimo de la función de costo debido a que la tasa de aprendizaje es muy grande. Por último, validemos los resultados con los datos de prueba.

η	**Multiplicador de cota** $2N/\lambda_{max}$	SSE	R^2	R^2 **ajustado**
0.10657	0.1	2.661	0.86544	0.85547
0.69274	0.65	1.2035	0.93914	0.93463
1.0631	0.9975	11.496	0.41864	0.37558
1.0657	1	29.464	-0.48998	-0.60035

CUADRO 6.2: Resultados con datos de pruebas y datos Iris

De la tabla 6.2 confirmamos nuevamente que la mejor tasa de aprendizaje es igual a $0{,}65 \times 2N/\lambda_{max}$ pues es la que presenta el menor SSE y los coeficientes de determinación R^2 y R^2 ajustado más altos.

A partir de los resultados encontrados, se puede concluir que para regresión lineal la tasa de aprendizaje óptima es igual a $0{,}65 \times 2N/\lambda_{max}$ dado que es un valor que es menor que la cota teórica derivada en el capítulo 4 y que además es la más rápida en minimizar el error de predicción así como también calcula las mejores predicciones comparado a otros valores de tasas de aprendizaje η. Ahora que se presentaron los resultados para una regresión lineal (o equivalentemente una neurona regresora) validemos si el comportamiento es similar para la UNVA regresora.

6.2. Resultados de la UNVA

En esta sección se presenta los resultados de la UNVA con dos conjuntos de datos. El primer conjunto de datos fue creado manualmente del mismo modo que en la subsección 6.1.1. El segundo conjunto contiene información del valor de apertura y de cierre de Bitcoin desde su inicio [11]. Aunque se podría plantear un problema de análisis multivariado en el cual se considera múltiples predictores como el valor de apertura, el valor de cierre o el volumen, el problema que se propone solo considera una sola variable. El objetivo de este problema es predecir el valor de apertura del Bitcoin en un día arbitrario a partir de los valores de apertura de los 5 días anteriores. Por lo tanto, este problema se puede modelar por medio de la ecuación:

$$\boldsymbol{x}_n = \theta_1 \boldsymbol{x}_{n-1} + \theta_2 \boldsymbol{x}_{n-2} + \theta_3 \boldsymbol{x}_{n-3} + \theta_4 \boldsymbol{x}_{n-4} + \theta_5 \boldsymbol{x}_{n-5} + \boldsymbol{\epsilon}_n, \tag{6.5}$$

o equivalentemente

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\theta} + \boldsymbol{\epsilon}, \tag{6.6}$$

tal que $\boldsymbol{y} = \boldsymbol{x}_n$, $\boldsymbol{X}$ es una matriz con 5 columnas en la que cada columna representa uno de los 5 retrasos considerados y $\boldsymbol{\epsilon}$ es el vector de error del modelo. La ecuación 6.5 indica que el modelo es autoregresivo [16], no obstante la ecuación

(6.6), sugiere que este problema se puede tratar como un problema de regresión lineal en el que se asume que la variable respuesta es igual a una combinación lineal de valores de entrada y por ende es posible utilizar la UNVA en este problema.

Antes de mostrar los resultados, cabe resaltar que para ambos conjuntos de datos el valor de la tasa de aprendizaje de la UNVA fue calculado únicamente a partir de la cota que incluye λ_{max} ya que de acuerdo con lo obtenido en la sección (6.1) un valor óptimo para la tasa de aprendizaje es $0{,}65 \times 2N/\lambda_{max}$.

6.2.1. Resultados de UNVA con datos manuales

Primero se ejecutó el entrenamiento de la arquitectura UNVA tal que para toda época $\eta = 0{,}65 \times 2N/\lambda_{max}$ Observemos el comportamiento de los pesos θ de la UNVA:

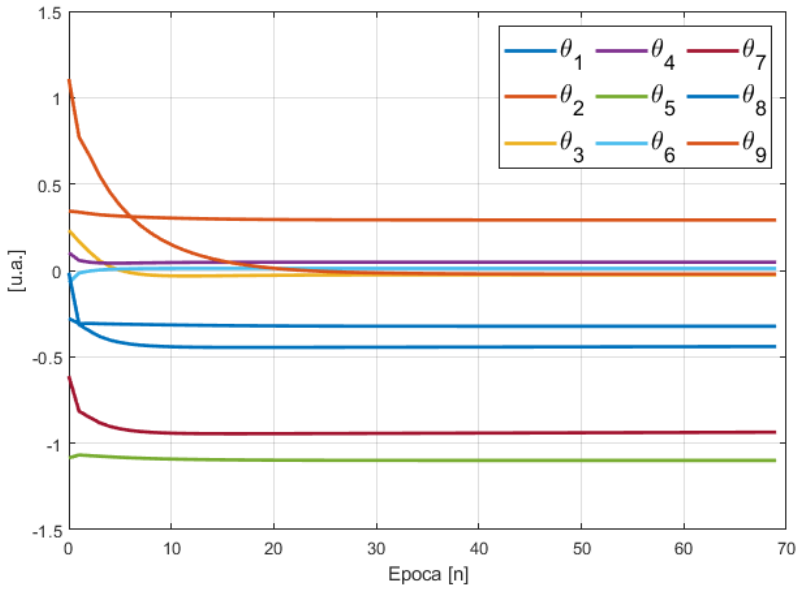


FIGURA 6.8: Evolución de coeficientes de UNVA con $\eta = 0{,}65 \times 2N/\lambda_{max}[n]$.

En la figura 6.8 vemos que todos los coeficientes convergen, en particular, incluyendo los pesos adicionales por lo cual lo más probable es que se minimizó globalmente la función de costo o se llegó a un punto en la función de costo que es un mínimo local. De no ser así el valor de los pesos divergería u oscilaciría. De hecho, la figura 6.9 revela que el flujo, el valor de retroalimentación de la UNVA, también converge por epocas y la figura 6.10 ilustra que el error se minimizó.

Al repetir la ejecución del algoritmo de descenso del gradiente con varias tasas de aprendizaje menores y mayores que la cota $2N/\lambda_{max}[n]$ se obtuvo las curvas de error presentadas en la figura 6.11. Cabe resaltar que aunque los coeficientes fueron inicializados aleatoriamente para cada una de las tasas de aprendizaje los errores ilustrados no comienzan en el mismo valor porque en la figura 6.11 se muestra el

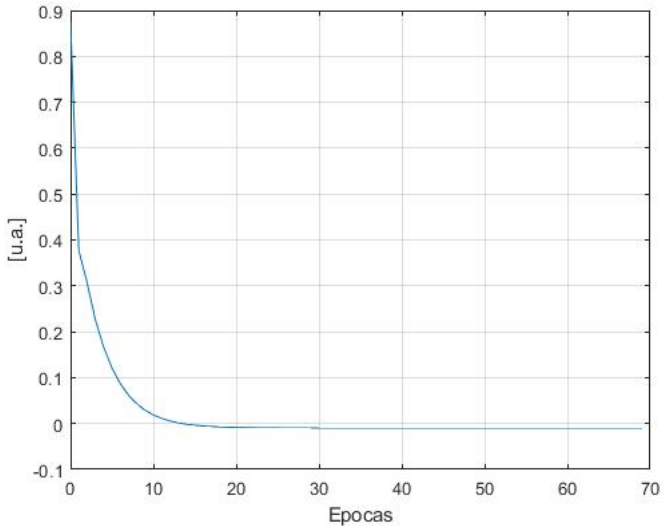


FIGURA 6.9: Evolución del flujo medio por épocas.

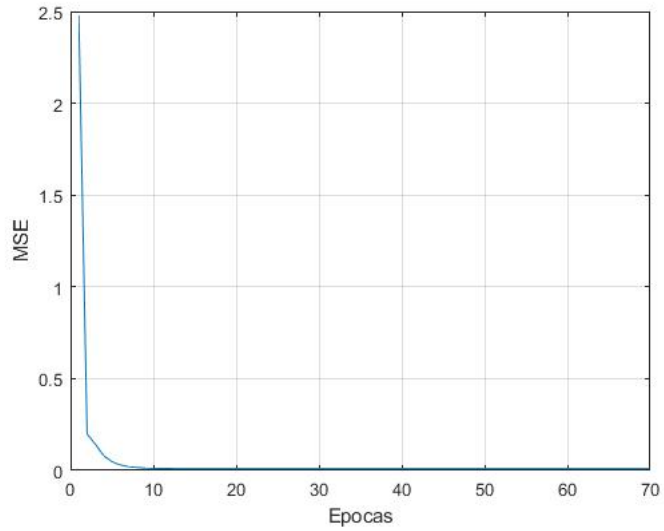


FIGURA 6.10: MSE de UNVA.

error promedio por época y no por fila y el cálculo del promedio hace que el primer valor sea distinto en cada uno de los entrenamiento.

Primero podemos ver que, a diferencia de los errores con regresión lineal, para todas las curvas el error converge hasta llegar a 0, incluyendo los casos en los que el multiplicador de la cota es mayor a 1. El estudio de por qué ocurre lo anterior se dejará como trabajo futuro. Sin embargo, nuestra hipótesis es que al ser un sistema dinámico y al cambiar las condiciones de convergencia en cada iteración, en este ejemplo se garantiza que no se inestabilice el sistema. En particular, vemos que en las 2 primeras iteraciones el error decrece muy rápido y desde la tercera iteración la convergencia del error es más suave. Esto se debe a que en las dos primeras iteraciones el error sólo se calcula modificando los primeros M pesos y desde la tercera

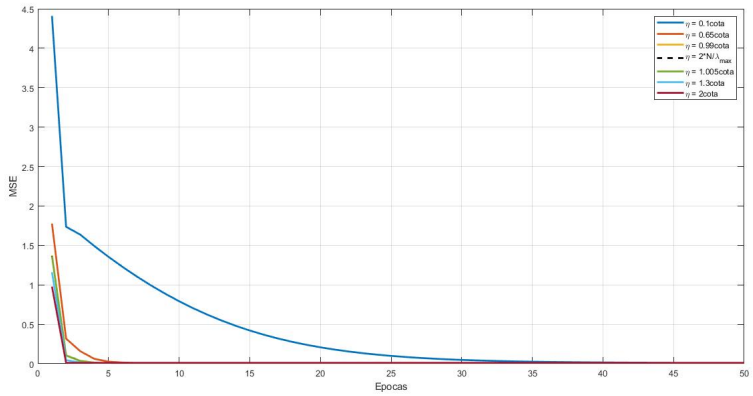


FIGURA 6.11: Evolución de MSE de UNVA calculado con varios multiplicadores a partir de cota $2N/\lambda_{max}$ y con datos manuales

época se modifican los $M + 4$ pesos de la UNVA.

6.2.2. Resultados de UNVA para pronósticos con series de tiempo

Como se mencionó anteriormente, se evaluó el comportamiento de la UNVA para un problema de pronóstico el cual tiene ciertas similaridades a los problemas con regresión lineal. Considere los datos de Bitcoin en la gráfica 6.12 extraídos de [11].

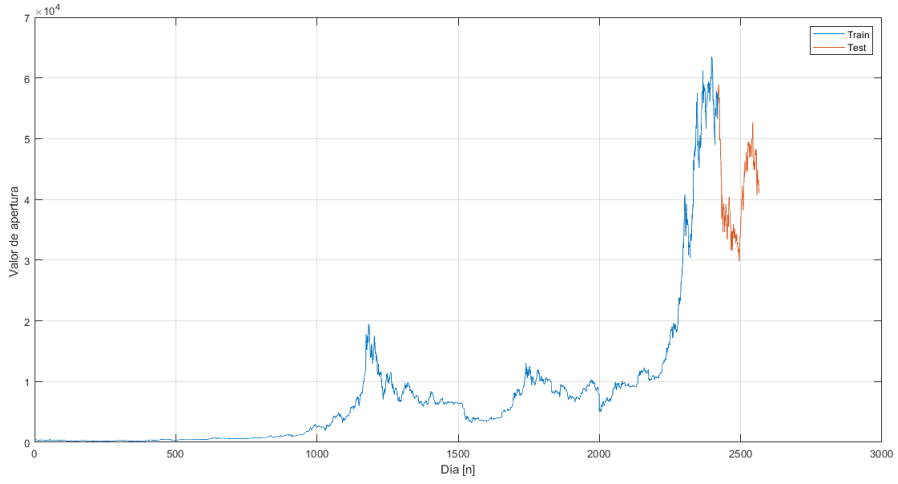


FIGURA 6.12: Evolución de valor de apertura de Bitcoin

Note que en la gráfica 6.12 se divide en 2 partes, la parte azul representa los datos de entrenamiento (94 % del total de datos) y la parte roja representa los datos de prueba (6 % del total). Cabe resaltar que se tomó una proporción grande del total de datos para cubrir parte del comportamiento inusual el cual se marca a partir del dato 2300 aproximadamente en el que el valor de apertura del Bitcoin crece considerablemente. Luego, desde la muestra 2450 aproximadamente el valor del Bitcoin se mantiene

entre 3×10^4 y 5×10^4, el cual es un conjunto de valores que ya ha observado el modelo al entrenarse y por ende, se tomaron estos últimos valores como datos de prueba.

Para el entrenamiento, se ejecutó el algoritmo de descenso de gradiente por 80 épocas. Cabe resaltar que aunque los primeros M coeficientes θ de la UNVA se inicializaron iguales a los M coeficientes ω de la regresión lineal, el error con el cual inicia cada modelo es distinto debido a los cuatro coeficientes adicionales que tiene la UNVA. Se obtuvo errores de entrenamiento similares con regresión lineal y con la UNVA de acuerdo con la figura 6.13. Vemos primero que ambos errores tardan menos de 10 épocas para converger al mínimo por lo cual no se puede entrenar por demasiadas épocas los algoritmos, de lo contrario el modelo estaría sobreajustado y los errores de prediccion serían muy grandes. Por este motivo, después de probar con distintas épocas de entrenamiento se determinó que con 80 épocas se evita el sobreajuste. Segundo, a pesar de que el error de la UNVA es menor al de la regresión lineal, la diferencia entre ambos errores es muy pequeña.

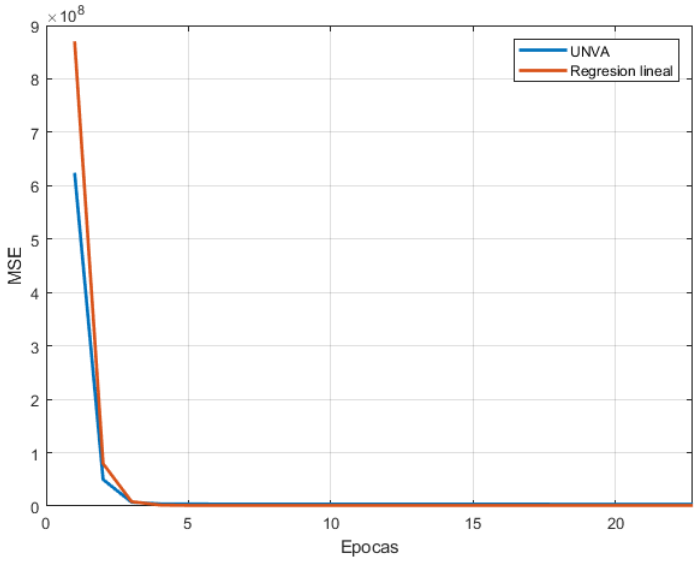


FIGURA 6.13: Errores de entrenamiento de regresion lineal y UNVA para series temporales

Ahora analicemos el comportamiento del flujo y la evolución de la cota lo cual se muestra en la figura 6.14. En la figura 6.14 vemos que inicialmente el flujo tiene un máximo relacionado con las dos primeras épocas pero desde la tercera época, la curva del flujo es mucho más suave e incluso converge a aproximadamente -0.169.

Adicionalmente, vemos que la cota inicialmente decrece muy rápido en las primeras 2 iteraciones pero después empieza a crecer lentamente aunque el valor de la cota no aumenta considerablemente pues como se ve en la gráfica, el orden del valor de la cota es de 10^{-9} es decir muy cercano a 0. Esto quiere decir que la tasa de aprendizaje debe ser aún más cercana a 0, lo que sugiere que el vector de coeficientes $\boldsymbol{\theta}$

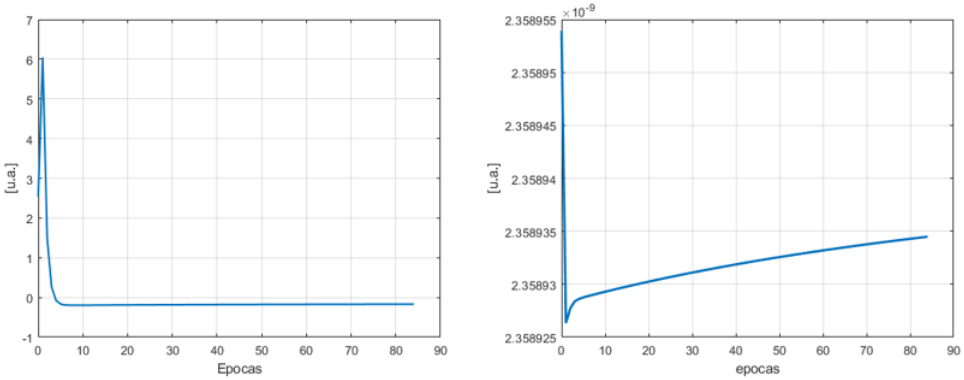


FIGURA 6.14: Evolucion de flujo (izquierda) y de la cota (derecha) en entrenamiento de UNVA para series de tiempo

no debe modificarse más. Este comportamiento puede explicar el por qué el modelo converge aún cuando se usa la cota como valor de η.

Por último, veamos los resultados de la UNVA obtenidos con los datos de prueba y comparemos el modelo autoregresivo tradicional. Recordemos que el modelo autogregresivo planteado considera que la predicción en el instante n depende de los valores en los instantes $n - i$ para todo $i \in \{1, 2, 3, 4, 5\}$. Además, como se utiliza predicciones pasadas para calcular la siguiente estimación y, que cada estimación tiene su respectivo error, el error de predicción se acumula a medida que se avanza en los instantes de tiempo y consecuencia, en los instantes futuros más lejanos en el tiempo las predicciones son muy alejadas de los valores verdaderos. Por esta razón, en la figura 6.15 se ilustra los pronósticos del valor de apertura del Bitcoin para las primeras 10 muestras.

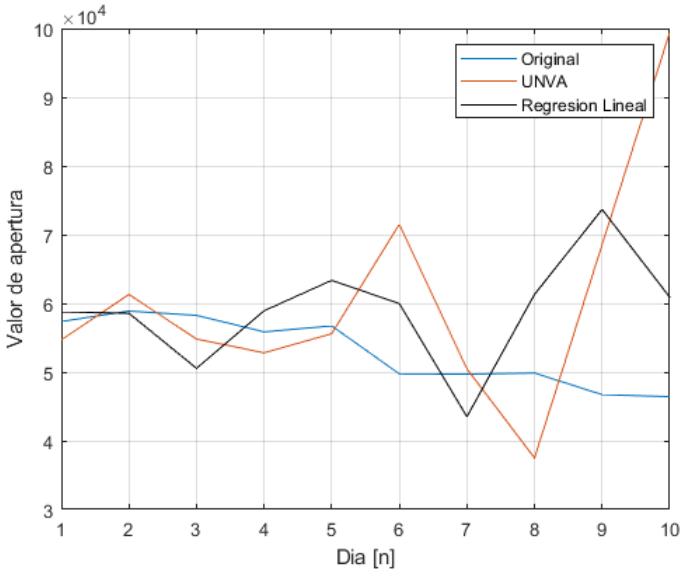


FIGURA 6.15: Pronóstico de valor de apertura de Bitcoin con UNVA y con modelo autoregresivo

En la figura 6.15 la curva azul representa los datos de prueba, la roja es del pronóstico de la UNVA y la negra es el pronóstico del modelo autoregresivo, entrenado del mismo modo que la regresión lineal. Primero, vemos que tanto para la UNVA como para el modelo autogregresivo tradicional el pronóstico empeora para los datos futuros más lejanos como lo explicamos anteriormente. Además, se puede observar que en las primeras 5 muestras en las que la serie de tiempo no tiene casi cambios, la UNVA tiene un mejor ajuste que el modelo tradicional autoregresivo. Sin embargo, desde la muestra 5 la variable respuesta disminuye, el cual es un cambio que también detecta el modelo autoregresivo pero que la UNVA no predice correctamente pues su predicción indica que el valor de apertura aumenta cuando en realidad disminuye. Finalmente, desde la sexta predicción, se puede ver que el error acumulado afecta demasiado la estimación final así como para el resto de predicciones futuras.

Capítulo 7

Conclusión

7.1. Resumen general

En esta tesis se partió primero de una revisión de los fundamentos necesarios para la arquitectura propuesta. Por un lado, se estudiaron los fundamentos biológicos entorno al sistema nervioso, su composición y su funcionamiento a partir de las células que permiten el procesamiento de información en el cerebro: las neuronas. Se explicó cómo el funcionamiento de las neuronas no sólo depende de la transmisión de señales eléctricas, el potencial de acción, entre las neuronas sino también de la energía proveniente del consumo de glucosa y oxígeno en la sangre. En particular, se presentó la relación entre los cambios vasculares en el cerebro y la actividad eléctrica de las neuronas, la cual se denomina el acoplamiento neurovascular. Por otro lado, también se revisaron modelos ya existentes de unidades neuronales artificiales como lo son la neurona de McCulloch-Pitts, el Perceptrón, el Adaline y la neurona regresora. De esta forma, el acoplamiento neurovascular y las unidades neuronales artificiales estudiadas constituyen la base teórica de la cual se parte para construir la Unidad NeuroVascular Artificial (UNVA).

Antes de formular la Unidad Neurovascular Artificial se determinó que era necesario evaluar las condiciones de estabilidad del algoritmo de entrenamiento, para reproducirlas en la construcción de la arquitectura propuesta. Por este motivo, en el capítulo 4 se presentaron las condiciones de convergencia de sistemas de ecuaciones en diferencias lineales y adicionalmente, se estudió la evolución de los pesos de la unidad neuronal artificial con el descenso del gradiente desde un punto de vista de sistemas dinámicos, apoyándose del análisis de Pierre Baldi [3]. Finalmente, la información del capítulo 4 nos permitió establecer una cota teórica de la tasa de aprendizaje utilizada en el descenso del descendiente

En el capítulo 5 se presentó el funcionamiento de la UNVA la cual nace principalmente del artículo de Riera et al. [31] en el que se modela por medio de ecuaciones diferenciales el acoplamiento neurovascular. En particular, se partió de un sistema de ecuaciones diferenciales que describe la relación entre el aumento de flujo sanguineo cerebral y la activación neuronal 5. Por último se adaptó el descenso del gradiente para la UNVA así como también se adaptó el criterio de la cota de la tasa de aprendizaje de este algoritmo para garantizar su estabilidad.

Finalmente, en el capítulo 6 se validó primero el resultado de las cota teórica con una regresión lineal en dos conjuntos de datos: el primer conjunto consiste en datos generados manualmente y el segundo conjunto es el de las flores Iris. Para ambos conjuntos de datos, se determinó empíricamente que un valor óptimo para la tasa de aprendizaje η es $0{,}65 \times 2N/\lambda_{max}$, donde N es la cantidad de datos y λ_{max} el valor propio más grande de la matriz $\boldsymbol{X}^{\boldsymbol{T}}\boldsymbol{X}$. Segundo, se probó la UNVA en dos conjuntos de datos: el primero es el mismo conjunto de datos generado manualmente utilizado para la regresión lineal y el segundo conjunto de datos consiste en la evolución del valor de apertura de Bitcoin. Este segundo conjunto de datos se eligió para comparar las pronósticos de la UNVA con los pronósticos de un modelo autoregresivo clásico el cual se entrenó del mismo modo que se entrenan los pesos de una regresión lineal. Al comparar ambas predicciones se determinó que la UNVA es más sensible a los cambios grandes de los valores comparado al modelo autoregresivo. Sin embargo, para los cambios de menor amplitud la UNVA se ajusta mejor a los datos verdaderos.

7.2. Trabajo futuro

Como trabajo futuro, hay varios elementos a tener en cuenta que no fueron explorados en esta tesis pero que complementan el trabajo realizado hasta ahora.

- Primero, el análisis de estabilidad se puede generalizar aún más si se considera que la función f en la ecuación (4.3) no es la identidad sino otra función de enlace como la función logística o la tangente hiperbólica. Además, el cambio de la función f también afecta el valor de la cota de la tasa de aprendizaje.

- Segundo, se debe profundizar en la función de costo de la UNVA y por ende en la actualización de los pesos de la UNVA al derivar los términos con índice temporal $u[n-1] - u_0[n], f[n-2]$ y $f[n-1]$.

- Se puede mejorar los resultados obtenidos de la cota teórica considerando conjuntos de datos con más filas y columnas. En tanto a los pronósticos con series de tiempo, vale la pena estudiar el comportamiento de la UNVA a partir de un modelo multivariado en vez de solo considerar un sólo predictor y sus retardos.

- Una vez formalizada la UNVA, se podría estudiar también la interconexión de múltiples UNVAs para obtener una arquitectura de mayor complejidad similar a las redes neuronales. Este nuevo tipo de arquitectura se denominaría Red NeuroVascular Artificial (RNVA).

Bibliografía

[1] F. Amenta, D. Zaccheo y W.L. Collier. «Neurotransmitters, neuroreceptors and aging». En: *Mechanisms of Ageing and Development* 61.3 (1991), págs. 249-273. ISSN: 0047-6374. DOI: https://doi.org/10.1016/0047-6374(91)90059-9. URL: https://www.sciencedirect.com/science/article/pii/0047637491900599.

[2] Sara Bachiller y col. «Microglia in Neurological Diseases: A Road Map to Brain-Disease Dependent-Inflammatory Response». En: *Frontiers in Cellular Neuroscience* 12 (2018), pág. 488. ISSN: 1662-5102. DOI: 10.3389/fncel.2018.00488. URL: https://www.frontiersin.org/article/10.3389/fncel.2018.00488.

[3] Pierre Baldi. «Gradient descent learning algorithm overview: A general dynamical systems perspective». En: *IEEE Transactions on neural networks* 6.1 (1995), págs. 182-195.

[4] Stephen Boyd y Lieven Vandenberghe. *Convex Optimization*. Cambridge University Press, 2004. DOI: 10.1017/CBO9780511804441.

[5] David Chambers, Christopher Huang y Gareth Matthews. «Nerve action potential and propagation». En: *Basic Physiology for Anaesthetists*. Cambridge University Press, 2015, 221–227. DOI: 10.1017/CBO9781139226394.051.

[6] Jasvinder Chawla y Ashmeet Singh Sahni. «Peripheral Nervous System Anatomy». En: *Medscape* (30 de jun. de 2016). visitado en 2-11-2021. URL: https://emedicine.medscape.com/article/1948687-overview.

[7] Xiaowei Chen, Samarjit Kar y Dan A Ralescu. «Cross-entropy measure of uncertain variables». En: *Information Sciences* 201 (2012), págs. 53-60.

[8] J. Dunn y M.H. Grider. *Phisiology, Adenosine Triphosphate*. visitado el 2 de mayo 2021. URL: \url{https://www.ncbi.nlm.nih.gov/books/NBK553175/}.

[9] J-OC. Dunn, MG. Mythen y MP. Grocott. «Physiology of oxygen transport». En: *BJA Education* 16.10 (mayo de 2016), págs. 341-348. ISSN: 2058-5349. DOI: 10.1093/bjaed/mkw012. eprint: https://academic.oup.com/bjaed/article-pdf/16/10/341/6690771/mkw012.pdf. URL: https://doi.org/10.1093/bjaed/mkw012.

[10] Jason Fernando. *R-Squared*. [Online; visitado el 21 de abril de 2021]. 2021. URL: https://www.investopedia.com/terms/r/r-squared.asp.

[11] Yahoo Finance. *Bitcoin USD (BTC-USD)*. [Online; visitado el 20 de octubre de 2021]. URL: https://finance.yahoo.com/quote/BTC-USD/history?p=BTC-USD.

[12] Ronald A Fisher. «The use of multiple measurements in taxonomic problems». En: *Annals of eugenics* 7.2 (1936), págs. 179-188.

[13] Karl J Friston y col. «Nonlinear responses in fMRI: the Balloon model, Volterra kernels, and other hemodynamics». En: *NeuroImage* 12.4 (2000), págs. 466-477.

[14] Sheriar G. Hormuzdi y col. «Electrical synapses: a dynamic signaling system that shapes the activity of neuronal networks». En: *Biochimica et Biophysica Acta (BBA) - Biomembranes* 1662.1 (2004). The Connexins, págs. 113-137. ISSN: 0005-2736. DOI: https://doi.org/10.1016/j.bbamem.2003.10.023. URL: https://www.sciencedirect.com/science/article/pii/S0005273604000410.

[15] Clément Huneau, Habib Benali y Hugues Chabriat. «Investigating Human Neurovascular Coupling Using Functional Neuroimaging: A Critical Review of Dynamic Models». En: 9 (2015), pág. 467. ISSN: 1662-453X. DOI: 10.3389/fnins.2015.00467. URL: https://www.frontiersin.org/article/10.3389/fnins.2015.00467.

[16] Rob J Hyndman y George Athanasopoulos. *Forecasting: principles and practice*. OTexts, 2018.

[17] Wu Jia y col. «Hyperparameter Optimization for Machine Learning Models Based on Bayesian Optimizationb». En: *Journal of Electronic Science and Technology* 17.1 (2019), págs. 26-40. ISSN: 1674-862X. DOI: https://doi.org/10.11989/JEST.1674-862X.80904120. URL: https://www.sciencedirect.com/science/article/pii/S1674862X19300047.

[18] Richard Arnold Johnson, Dean W Wichern y col. *Applied multivariate statistical analysis*. Vol. 6. Pearson London, UK: 2014.

[19] Aarushi Khanna. *Action Potential*. visitado el 25 de abril 2021. 2021. URL: \url{https://teachmephysiology.com/nervous-system/synapses/action-potential/}.

[20] Joost N Kok y col. «Artificial intelligence: definition, trends, techniques, and cases». En: *Artificial intelligence* 1 (2009), págs. 270-299.

[21] Lai-Wo Stan Leung y Chi-Yiu Conrad Yim. «Intrinsic membrane potential oscillations in hippocampal neurons in vitro». En: *Brain research* 553.2 (1991), págs. 261-274.

[22] H. Lodish y col. *Overview of Neuron Structure and Function*. Ed. por New York: W. H. Freeman. 2000. URL: https://www.ncbi.nlm.nih.gov/books/NBK21535/.

[23] D. G. Luenberger. *Introduction to dynamic systems : theory, models, and applications*. Wiley, 1979.

[24] John R. Manning. «Diffusion in a Chemical Concentration Gradient». En: *Phys. Rev.* 124 (2 1961), págs. 470-482. DOI: 10.1103/PhysRev.124.470. URL: https://link.aps.org/doi/10.1103/PhysRev.124.470.

[25] W. McCulloch y W. Pitts. «A Logical Calculus of the Ideas Immanent in Nervous Activity». En: *Bulletin of Mathematical Biology* 52.1, 2 (1943), págs. 99-115. DOI: https://doi.org/10.1007/BF02478259.

[26] Tim Newman. «All about the central nervous system». En: *MedicalNewsToday* (22 de dic. de 2017). visitado en 2-11-2021. URL: https://www.medicalnewstoday.com/articles/307076.

[27] S. Ogawa y col. «Brain magnetic resonance imaging with contrast dependent on blood oxygenation». En: *Proceedings of the National Academy of Sciences* 87.24 (1990), págs. 9868-9872. ISSN: 0027-8424. DOI: 10.1073/pnas.87.24.9868. eprint: https://www.pnas.org/content/87/24/9868.full.pdf. URL: https://www.pnas.org/content/87/24/9868.

[28] Mahan Mathur Omar Islam. *Brain Magnetic Resonance Imaging*. visitado en 17-04-2021. URL: https://emedicine.medscape.com/article/2105033-overview.

[29] Linus Pauling y Charles D. Coryell. «The Magnetic Properties and Structure of Hemoglobin, Oxyhemoglobin and Carbonmonoxyhemoglobin». En: *Proceedings of the National Academy of Sciences* 22.4 (1936), págs. 210-216. ISSN: 0027-8424. DOI: 10.1073/pnas.22.4.210. eprint: https://www.pnas.org/content/22/4/210.full.pdf. URL: https://www.pnas.org/content/22/4/210.

[30] Kevin A. Pelphrey. «Blood-Oxygen-Level-Dependent (BOLD) Signal». En: *Encyclopedia of Autism Spectrum Disorders*. Ed. por Fred R. Volkmar. New York, NY: Springer New York, 2013, págs. 465-466. ISBN: 978-1-4419-1698-3. DOI: 10.1007/978-1-4419-1698-3_550. URL: https://doi.org/10.1007/978-1-4419-1698-3_550.

[31] Jorge J Riera y col. «Nonlinear local electrovascular coupling. I: A theoretical model». En: *Human brain mapping* 27.11 (2006), págs. 896-914.

[32] Frank Rosenblatt. «The perceptron: a probabilistic model for information storage and organization in the brain.» En: *Psychological review* 65.6 (1958), pág. 386.

[33] Murat Sazli. «A brief review of feed-forward neural networks». En: *Communications, Faculty Of Science, University of Ankara* 50 (ene. de 2006), págs. 11-17. DOI: 10.1501/0003168.

[34] Manu Shaurya. *McCulloch-Pitts Neuron vs Perceptron model*. 2019. URL: https://medium.com/@manushaurya/mcculloch-pitts-neuron-vs-perceptron-model-8668ed82c36,visited2020-11-24.

[35] Roger P. Simon, David A. Greenberg y Michael J. Aminoff. «Deficiencias motoras». En: *Neurología clínica, 7e*. New York, NY: McGraw-Hill Education, 2015. URL: accessmedicina.mhmedical.com/content.aspx?aid=1119993943.

[36] Michael V Sofroniew y Harry V Vinters. «Astrocytes: biology and pathology». En: *Acta neuropathologica* 119.1 (2010), págs. 7-35.

[37] A. M. TURING. «I.—COMPUTING MACHINERY AND INTELLIGENCE». En: *Mind* LIX.236 (oct. de 1950), págs. 433-460. ISSN: 0026-4423. DOI: 10.1093/mind/LIX.236.433. eprint: https://academic.oup.com/mind/article-pdf/LIX/236/433/30123314/lix-236-433.pdf. URL: https://doi.org/10.1093/mind/LIX.236.433.

[38] MRC Mitochondrial Biology Unit. *What are mitochondria*. URL: http://www.mrc-mbu.cam.ac.uk/what-are-mitochondria,visited2020-11-24.

[39] Sun-Chong Wang. «Artificial Neural Network». En: *Interdisciplinary Computing in Java Programming*. Boston, MA: Springer US, 2003, págs. 81-100. ISBN: 978-1-4615-0377-4. DOI: 10.1007/978-1-4615-0377-4_5. URL: https://doi.org/10.1007/978-1-4615-0377-4_5.

[40] Wikipedia. *Red neuronal artificial*. URL: https://es.wikipedia.org/wiki/Red_neuronal_artificial,visited2020-11-24 (visitado 20-11-2020).

[41] Wikipedia. *Respiración celular — Wikipedia, La enciclopedia libre*. [Internet; descargado 3-noviembre-2021]. 2021. URL: \url{https://es.wikipedia.org/w/index.php?title=Respiraci%C3%B3n_celular&oldid=139277796}.

[42] Wikipedia contributors. *Functional magnetic resonance imaging — Wikipedia, The Free Encyclopedia*. [Online; accessed 17-Abril-2021]. 2021. URL: https://en.wikipedia.org/w/index.php?title=Functional_magnetic_resonance_imaging&oldid=1016553270.

[43] Wikipedia contributors. *List of systems of the human body — Wikipedia, The Free Encyclopedia*. https://en.wikipedia.org/w/index.php?title=List_of_systems_of_the_human_body&oldid=1019327234. [Online; accessed 22-April-2021]. 2021.

[44] DROW Wl. *An adaptive adaline neuron using chemical memistors*. Inf. téc. Standford Electronics Lavoratroy Technical Report, 1960.