



Universidad Nacional Autónoma de México
Facultad de Química

Clasificación de receptores acoplados a proteína G
mediante una red neuronal

TESIS

que para obtener el título de:

QUÍMICO

presenta:

Arsenio Natahel Cruz Cardoso

Ciudad de México
8 de noviembre de 2021





Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

Jurado asignado

Presidenta

Profesora: Dra. Padilla Martinez Kira

Vocal

Profesor: M. en C. Cortés Arce Abel Humberto

Secretaria

Profesora: Dra. Domínguez Dueñas Laura

1° Suplente

Profesor: Dr. Torres Herrera Ulises

2° Suplente

Profesor: Dr. Franco Perez José Marco Antonio

Facultad de Química, edificio F

Dra. Laura Domínguez Dueñas

Asesora

Dr. Cesar Iván León Pimentel

Supervisor técnico

Arsenio Natahel Cruz Cardoso

Sustentante

*Dedicado a
mi familia*

y en especial a mis padres.

Agradecimientos

Esta tesis es producto de un gran esfuerzo, tanto de mi parte como de muchas más personas que me han apoyado a lo largo de la creación de este escrito. Antes que nada, me gustaría agradecer a mis padres que siempre me han apoyado sin importar que tan alocados puedan llegar a ser mis planes, siempre he contado con su incondicional ayuda para lograr mis metas. También son relevantes mis hermanos que nunca han dudado en darme soporte cuando lo necesito y que hacen mucho más amena mi vida y mis experiencias. Todos juntos hemos logrado cumplir muchos proyectos. Por otro lado, me gustaría agradecer a la Dra. Laura Domínguez Dueñas por brindarme un lugar en su grupo de trabajo para poder desarrollar el presente trabajo. También agradezco que me haya dejado madurar como científico y ser humano, así como la oportunidad de conocer a mis compañeros de laboratorio que nunca han dudado en echarme una mano cuando lo necesito y que son personas extraordinarias que constantemente me motivan a continuarme mejorando.

Otro agradecimiento que no puedo dejar pasar es al M. en C. Guillermo Goode-Romero que nunca me quedo mal cuando le pedí asesorías para aclarar dudas referentes a la presente investigación e incluso cuando requería una revisión de los primeros, y también los últimos, borradores de este escrito. Lo cual, para nada, no fue poco.

Referente a este escrito tampoco podría ignorar las contribuciones que el Dr. Cesar Iván León Pimentel aportó. Junto a Guillermo permitieron que el presente escrito fuera lo suficientemente coherente y entendible. Y en caso de no encontrarse con esta situación al leer este trabajo, me atribuyo la culpa.

Me sabe mal no poder explayarme para agradecer a todas las personas que me han permitido llegar a este punto, pero al menos les dedicaré algunas líneas. Quiero agradecerles a mis amistades que hicieron amena mi carrera y me motivaban en aquellos momentos en los que no quería continuar. Así como en aquellos, en que simplemente disfrutábamos de nuestras experiencias. No podría pasar por alto agradecerles a mis compañeros de diferentes laboratorios en los que me han permitido estar ya que ellos no sólo me ayudaron a desarrollarme como ser humano y científico, sino también, muchas veces se convirtieron en mis amistades. También le quiero agradecer a todos los investigadores que me brindaron la oportunidad de realizar estancias en sus laboratorios, en especial a la Dra. Patricia Guadarrama Acosta.

Por último, pero no menos importante, me gustaría agradecerle a la UNAM que me ha podido brindar muchas de mis mejores y peores experiencias en la vida, aparte de que ha contribuido a que sea la persona que soy. Y todo por tan sólo 25 centavos al año.

Índice general

| | |
|---|-----------|
| Agradecimientos | V |
| Índice general | 2 |
| Índice de figuras | 4 |
| 1. Introducción | 5 |
| 1.1. Preámbulo y resumen del proyecto. | 5 |
| 1.2. Objetivos | 6 |
| 1.2.1. Objetivo global | 6 |
| 1.3. Tesis e hipótesis | 6 |
| 2. Antecedentes teóricos | 7 |
| 2.1. Receptores acoplados a proteínas (GPCR) | 7 |
| 2.1.1. Características estructurales | 7 |
| 2.1.2. Efecto de los ligandos en la geometría de los GPCR y diferentes actividades de ligandos. | 10 |
| 2.1.3. Efecto de los ligandos en el arreglo de las hélices alfa. | 11 |
| 2.1.4. Familia de GPCR tipo A, familia de la Rodopsina | 12 |
| 2.1.5. Numeración de Ballesteros-Weinstein | 12 |
| 2.2. Representación de las proteínas | 12 |
| 2.2.1. Cadena principal de las proteínas. | 13 |
| 2.2.2. Mapas de distancias de las proteínas | 13 |
| 2.3. Redes neuronales artificiales | 15 |
| 2.3.1. La neurona | 16 |
| 2.3.2. Hiperparámetros | 18 |
| 2.3.3. Funciones de activación | 19 |
| 2.3.4. Función de pérdida y de costo | 20 |
| 2.3.5. Algoritmos de optimización | 21 |
| 2.3.6. Arquitectura de la red neuronal | 22 |
| 2.3.7. Entrenamiento de una red neuronal y tipos de entrenamiento | 22 |
| 2.3.8. Redes Neuronales Convolucionales | 22 |
| 2.4. Software empleado | 28 |
| 2.4.1. Python | 28 |
| 2.4.2. TensorFlow | 28 |
| 2.4.3. Keras | 29 |
| 2.5. Aceleración de redes neuronales mediante GPU | 29 |
| 2.6. Bases de datos | 29 |
| 2.6.1. GPCRdb | 29 |
| 2.6.2. RCSB PDB | 30 |
| 2.7. Antecedentes de propuestas. | 30 |
| 3. Metodología | 31 |
| 3.1. Obtención y limpieza de los datos | 31 |
| 3.1.1. Obtención de los datos estructurales. | 31 |
| 3.2. Prototipado de diferentes arquitecturas | 34 |
| 3.2.1. Variación de los hiperparámetros | 35 |
| 3.2.2. Aceleración de entrenamiento mediante GPU | 36 |

| | |
|--|-----------|
| 4. Resultados, discusión de resultados, conclusiones y perspectivas | 37 |
| 4.1. Resultados modelo 01 | 37 |
| 4.2. Resultados modelo 02 | 39 |
| 4.3. Resultados modelo 03 | 40 |
| 4.4. Resultados modelo 04 | 41 |
| 4.5. Resultados modelo 05 | 42 |
| 4.6. Pruebas con GPU. | 43 |
| 4.6.1. Resultados modelo 04* GPU | 43 |
| 4.6.2. Resultados modelo 05 GPU | 44 |
| 4.7. Resultados globales. | 45 |
| 4.8. Conclusiones | 46 |
| 4.9. Perspectivas | 47 |
| | |
| 5. Anexo I, tablas de resultados | 51 |
| 5.1. Resultados 01 | 51 |
| 5.2. Resultados 02 | 54 |
| 5.3. Resultados 03 | 57 |
| 5.4. Resultados 04 | 60 |
| 5.5. Resultados 05 | 63 |
| 5.6. Resultados 04* GPU | 66 |
| 5.7. Resultados 05 GPU | 68 |

Índice de figuras

| | |
|---|----|
| 2.1. Estructura de la rodopsina en 2I35, un GPCR de la familia A. | 8 |
| 2.2. Estructura de la rodopsina en 2I35, un GPCR de la familia A visto desde la cara extracelular. | 8 |
| 2.3. Estructura de un GPCR en la membrana. | 9 |
| 2.4. Curvas de respuesta ante diferentes actividades de ligandos | 11 |
| 2.5. Cadena principal de la proteína. Imagen de Jane S. Richardson. https://commons.wikimedia.org/wiki/File:Protein_backbone_PhiPsiOmega_drawing.jpg | 13 |
| 2.6. Nomenclatura $C\alpha$ y $C\beta$ | 13 |
| 2.7. Estructura de un aminoácido. | 13 |
| 2.8. Representación de 5 puntos tridimensionales. | 14 |
| 2.9. Mapa de distancias de la matriz 2.1 | 15 |
| 2.10. Mapa de distancias del GPCR 2PED, obtenido con nanoHUB https://nanohub.org/resources/contactmaps | 15 |
| 2.11. Esquema simplificado de una red neuronal. | 16 |
| 2.12. La neurona y sus componentes. | 17 |
| 2.13. Imágenes de diagonales a clasificar. | 17 |
| 2.14. Algunos ejemplos de funciones de activación: (a) tanh, (b) ReLU y (c) Sigmoidea | 20 |
| 2.15. Arquitecturas usadas. | 23 |
| 2.16. Ejemplo de una convolución con un filtro MaxPooling. | 24 |
| 2.17. Proceso de convoluciones en una red neuronal convolucional. | 25 |
| 2.18. Imagen a procesar por una red neuronal convolucional. Imagen de François Chollete | 26 |
| 2.19. Imágenes obtenidas tras 32 convoluciones. Imagen de François Chollete | 26 |
| 2.20. Imágenes obtenidas al aplicar un filtro 'max pooling'. Imagen de François Chollete | 26 |
| 2.21. Imágenes obtenidas tras 2 nuevas convoluciones. Imagen de François Chollete | 27 |
| 2.22. Imágenes obtenidas al aplicar un filtro 'max pooling'. Imagen de François Chollete | 27 |
| 2.23. Imágenes obtenidas tras 2 nuevas convoluciones. Imagen de François Chollete | 27 |
| 2.24. Imágenes obtenidas al aplicar un filtro 'max pooling'. Imagen de François Chollete | 28 |
| 2.25. Aminoácidos en las diferentes hélices. Imagen de https://gpccrdb.org/family/001/ | 30 |
| 2.26. Estructuras de las hélices visto desde la cara extracelular de la membrana. Imagen de https://gpccrdb.org/family/001/ | 30 |
| 3.1. Obtención y limpieza de los datos usados. | 31 |
| 3.2. Matrices de distancias generadas con los modelos, a) Modelo 01, b) Modelo 02, c) Modelo 03, d) Modelo 04 y e) Modelo 05. | 33 |
| 3.3. Manejo de archivos. | 34 |
| 3.4. Arquitectura de la red neuronal. | 35 |
| 4.1. Pérdidas por época obtenidas para modelo 01 en, a) el grupo de entrenamiento y b) el grupo de validación. | 38 |
| 4.2. Tasa de acierto por época obtenidas para modelo 01 en, a) el grupo de entrenamiento y b) el grupo de validación. | 38 |
| 4.3. Pérdidas por época obtenidas para modelo 02 en, a) el grupo de entrenamiento y b) el grupo de validación. | 39 |
| 4.4. Tasa de acierto por época obtenidas para modelo 02 en, a) el grupo de entrenamiento y b) el grupo de validación. | 39 |
| 4.5. Pérdidas por época obtenidas para modelo 03 en, a) el grupo de entrenamiento y b) el grupo de validación. | 40 |

| | |
|---|----|
| 4.6. Tasa de acierto por época obtenidas para modelo 03 en, a) el grupo de entrenamiento y b) el grupo de validación. | 40 |
| 4.7. Pérdidas por época obtenidas para modelo 04 en, a) el grupo de entrenamiento y b) el grupo de validación. | 41 |
| 4.8. Tasa de acierto por época obtenidas para modelo 04 en, a) el grupo de entrenamiento y b) el grupo de validación. | 41 |
| 4.9. Pérdidas por época obtenidas para modelo 05 en, a) el grupo de entrenamiento y b) el grupo de validación. | 42 |
| 4.10. Tasa de acierto por época obtenidas para modelo 04 en, a) el grupo de entrenamiento y b) el grupo de validación. | 42 |
| 4.11. Pérdidas por época obtenidas para modelo 04* GPU en, a) el grupo de entrenamiento y b) el grupo de validación. | 43 |
| 4.12. Tasa de acierto por época obtenidas para modelo 04* GPU en, a) el grupo de entrenamiento y b) el grupo de validación. | 44 |
| 4.13. Pérdidas por época obtenidas para modelo 05 GPU en, a) el grupo de entrenamiento y b) el grupo de validación. | 44 |
| 4.14. Tasa de acierto por época obtenidas para modelo 05 GPU en, a) el grupo de entrenamiento y b) el grupo de validación. | 44 |
| 4.15. Pérdidas por época obtenidas para los 5 modelos y las 2 variantes en GPU en el grupo de entrenamiento. | 45 |
| 4.16. Pérdidas por época obtenidas para los 5 modelos y las 2 variantes en GPU en, a) el grupo de entrenamiento (época 3 a 20) y b) el grupo de validación. | 45 |
| 4.17. Tasa de acierto por época obtenidas para los 5 modelos y las 2 variantes en GPU en, a) el grupo de entrenamiento y b) el grupo de validación. | 46 |

Capítulo 1

Introducción

1.1. Preámbulo y resumen del proyecto.

Los receptores acoplados a proteína G son una superfamilia de proteínas de membrana transductoras de señales. Estas proteínas se suelen referir en la literatura como GPCR, siglas de *G Protein-Coupled Receptors* y debido a su gran presencia en las células suelen ser blancos para fármacos. Por lo que conocer el estado de estas proteínas, con base en su estructura secundaria y terciaria debida a la interacción con los diferentes ligandos, puede facilitar el desarrollo de nuevos fármacos de una manera más económica y rápida.

Al consultar la literatura sólo se logro identificar un trabajo en el que se buscara abordar este problema [1]. En dicho trabajo (**sección 2.7**), se obtuvo una tasa de acierto del 72.1 % para clasificar GPCR de la familia A entre los estados funcionales activo e inactivo (**sección 2.1.2**).

En este proyecto se desarrolló un algoritmo de redes neuronales artificiales capaz de clasificar la actividad del ligando asociado al GPCR, donde los ligandos son sustancias que forman complejos con biomoléculas, en este caso proteínas. Para lograr construir este algoritmo solamente se usó la información estructural de diferentes GPCR. Este trabajo puede ayudar a mejorar el entendimiento en el cambio de estructura tridimensional de estas proteínas y cómo sus interacciones con diversos ligandos afectan su conformación y la modulación de señales celulares.

Se consideraron únicamente las estructuras de la familia más abundante de GPCR, que es la familia A. Las clasificaciones de estas proteínas se realizaron al tomar en cuenta sólo tres estados funcionales de los GPCR, los cuales son; agonista, antagonista y agonista inverso. Inspirados por el algoritmo de AlphaFold desarrollado por Google [2], que resuelve exitosamente el problema de predecir las estructuras secundarias y terciarias a partir de la estructura primaria de una proteína, y la forma en que se representaron las proteínas en dicho trabajo se construyó un algoritmo de redes neuronales convolucionales que clasifica mapas de distancias, una representación bidimensional de la estructura terciaria de una proteína, con base en la actividad del ligando asociado al GPCR.

Tener un algoritmo que pueda clasificar al estado funcional de estas proteínas de acuerdo a su actividad biológica, y solamente requiere la estructura tridimensional de las proteínas, es de relevancia en estudios computacionales para el desarrollo de nuevos fármacos, como lo pueden ser dinámicas moleculares. En dichos estudios es común que tras una simulación se desconozca cual es el estado funcional del GPCR modelado. Dado que el algoritmo construido puede determinar el estado funcional del GPCR en cada paso de una simulación, este también se puede usar como criterio para discernir en qué momento en una dinámica molecular la estructura de un GPCR cambia de estado. Por último, este proyecto servirá de base para desarrollar algoritmos que de acuerdo con la estructura deseada en un GPCR puedan proponer ligandos que lleven al GPCR a dicha estructura.

Para lograr cumplir con los objetivos planteados en este proyecto se usó un algoritmo de aprendizaje supervisado y se construyó una red neuronal convolucional, que después de entrenarla, ajustar su arquitectura y sus hiperparámetros permitió obtener una tasa de acierto de hasta el 82.27 % sobre el grupo de entrenamiento y del 92.86 % sobre el grupo de validación. El alto porcentaje de acierto mostrado por el algoritmo, permite validarlo como un método confiable para determinar el estado funcional de un GPCR. Otra ventaja del método desarrollado es la facilidad para implementarlo ya que sólo se requiere de una estructura tridimensional de la proteína, en formato PDB, lo cual es relativamente fácil de obtener en una dinámica molecular.

Como perspectivas a futuro se planean diversas estrategias para mejorar aun más la tasa de acierto. Siendo las opciones más viables buscar aplicar algoritmos de aumento de datos para incrementar el tamaño del grupo de datos con el que se entrena el algoritmo o consultar en la literatura por nuevas técnicas de redes neuronales convolucionales que puedan trabajar eficientemente con cantidades de datos reducidas, en el orden cientos de datos. Por último, cada mes se liberan nuevas estructuras de GPCR experimentales, usualmente obtenidas mediante cristalografía de rayos X, las cuales pueden ser usadas para mejorar el desempeño del algoritmo.

1.2. Objetivos

1.2.1. Objetivo global

Caracterizar los estados conformacionales de proteínas de la familia A de los Receptores Acoplados a Proteínas G, mediante la construcción de una red neuronal convolucional, para catalogar el estado funcional de los GPCR tipo A de acuerdo a su estructura tridimensional y la actividad farmacológica del ligando asociado al GPCR.

1.3. Tesis e hipótesis

1. **Podemos representar la estructura tridimensional de las proteínas como imágenes mediante mapas de distancias.**
2. **Existen arquitecturas de redes neuronales artificiales que son muy eficientes para clasificar imágenes, como lo son las redes neuronales convolucionales.**
3. **Las interacciones de un GPCR con un ligando afectan su conformación estructural, lo cual se ve representado en los mapas de distancias.**

Podemos representar a las proteínas como mapas de distancias, imágenes que representan proteínas, y a partir de estos mapas se puede adaptar la tecnología de reconocimiento de imágenes mediante redes neuronales convolucionales para clasificar proteínas de la familia A de los GPCR, con base en la actividad del ligando con el que el GPCR interactúa.

Capítulo 2

Antecedentes teóricos

2.1. Receptores acoplados a proteínas (GPCR)

Los GPCR son la superfamilia más grande y común de proteínas que se han encontrado presentes en los genomas de los mamíferos [3]. Las proteínas miembros, constan de una sola cadena polipeptídica y son proteínas transmembranales y transductores de señales que se encuentran relacionadas evolutivamente [4]. Estas proteínas cambian su estado conformacional, es decir la forma en que se encuentra plegada la proteína en su estructura terciaria, al interactuar con un ligando. Los GPCR son modulados por una gran variedad de ligandos que van desde moléculas inorgánicas, iones, hasta otras proteínas, lo que genera una señal intracelular al cambiar la geometría de la proteína [5]. Estos cambios afectan la bioquímica de la célula al generar una cascada de vías de señalización. Se han descrito 8 familias principales de GPCR, que han sido subclasificadas, usando las letras A a F, con base en la homología de sus secuencias de aminoácidos [6]. En este proyecto sólo se consideró a la familia clase A (familia de la rodopsina), que es la familia con una mayor cantidad de información estructural en las bases de datos consultadas en este proyecto. Siendo esto un punto decisivo en este trabajo ya que por lo general los algoritmos de redes neuronales requieren de varios miles de datos para aprender a realizar predicciones sobre un grupo de datos, aunque en este caso se pudo trabajar con unos pocos cientos de casos.

Dada su interacción en las células, su gran variedad en el genoma humano y su alta presencia en diversas células, los GPCR suelen ser un blanco idóneo para el desarrollo de nuevos fármacos [7, 8, 4]. Es por este motivo que se han realizado diversos esfuerzos por lograr identificar, caracterizar y clasificar a estas proteínas, junto a sus interacciones con diversos ligandos, mediante diferentes técnicas [4]. Estas técnicas abarcan desde las experimentales como la cristalografía de rayos X para determinar la estructura de la proteína o FRET, de las siglas para *Fluorescence Resonance Energy Transfer*, para determinar las interacciones proteína a proteína, hasta las computacionales como lo son la dinámica molecular que permite predecir como interactúan átomos o moléculas por un determinado tiempo.

Actualmente todavía quedan varios GPCR sin clasificar y otros, en los que aun no se entiende perfectamente su función en las células. Por lo que éste es todavía un campo relevante de investigación, debido a los beneficios de tener un mejor entendimiento de estas proteínas.

2.1.1. Características estructurales

Estructuralmente, estas proteínas consisten en una sola cadena de aminoácidos. Los GPCR al plegarse en su estructura terciaria forman 7 hélices- α , o TM de *TransMembranal helix*, que atraviesan la membrana plasmática de la célula de extremo a extremo y tienen una última hélice- α (TM7) que es más corta que las demás y que no termina de atravesar completamente la membrana celular [9, 10]. En la **figura 2.1** se representa una estructura de una rodopsina, un GPCR de la familia A, donde cada hélice transmembranal tiene un color diferente.

Figura 2.1: Estructura de la rodopsina en 2I35, un GPCR de la familia A.

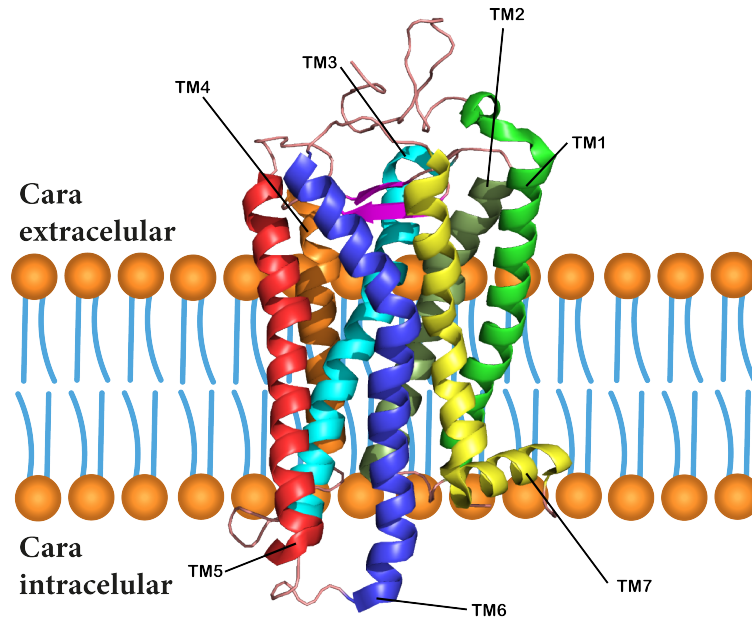
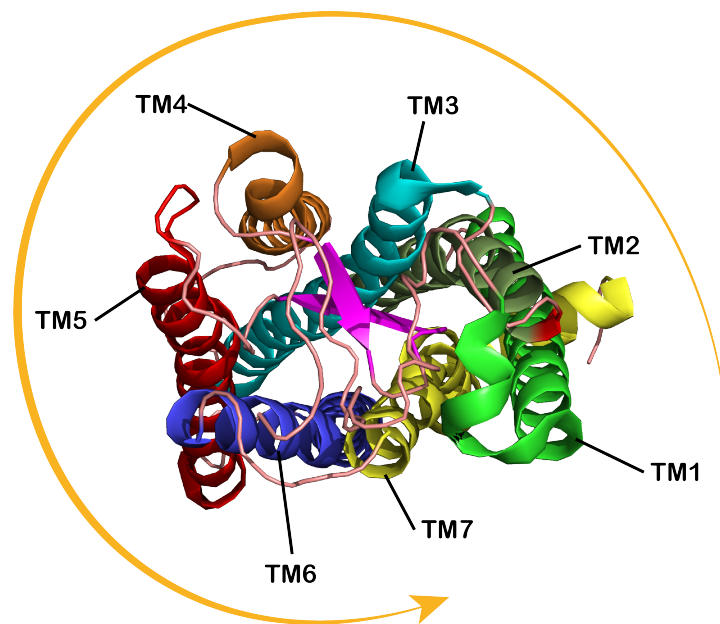


Figura 2.2: Estructura de la rodopsina en 2I35, un GPCR de la familia A visto desde la cara extracelular.

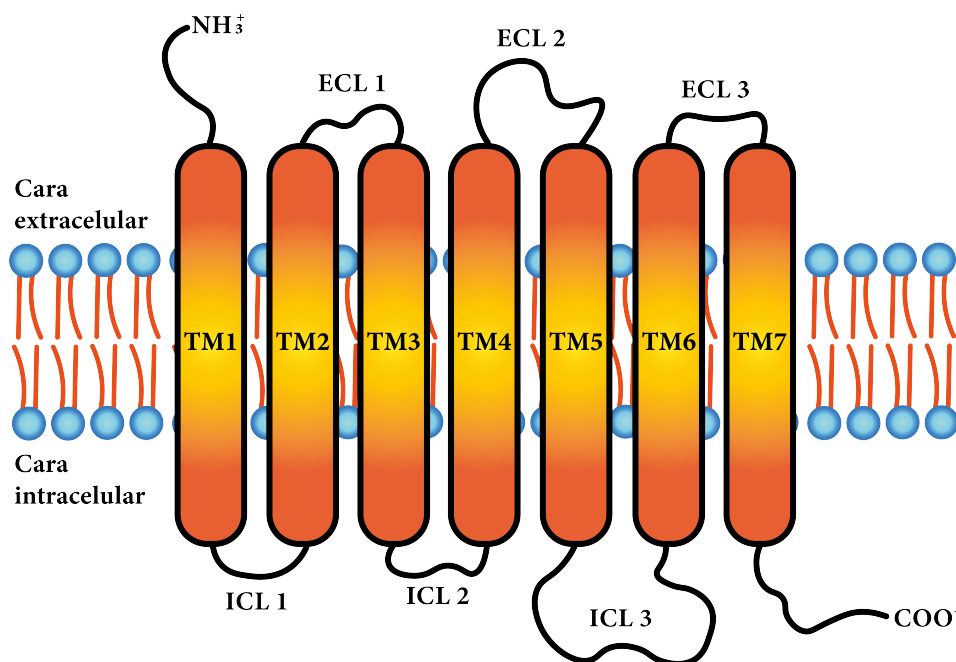


Estas hélices- α están unidas por 6 bucles, 3 intracelulares y 3 extracelulares. Con lo que podemos distinguir tres zonas en las hélices, una zona que corresponde al núcleo transmembrana, una zona intracelular y una extracelular. Estas zonas se encuentran representadas en la **figura 2.3**, donde el extremo amino terminal se encuentra en el exterior de la membrana celular y el extremo carboxilo terminal se encuentra en el citosol de la célula. Las 7 hélices tienen un arreglo en sentido antihorario vistas desde la cara extracelular, **figura 2.2** y se denominan TM1-TM7 en orden secuencial conforme aparecen en la cadena principal de la proteína. Los lazos o bucles extra e intracelulares que unen a las hélices pueden ser encontrados en la literatura como ECL1-ECL3 o ICL1-ICL3 (de ExtraCellular Loop e IntraCellular Loop respectivamente) [11], estos también se numeran en forma secuencial.

El ángulo del eje de la hélice- α , formado entre la recta de ajuste de los aminoácidos de cada hélice

medido con respecto al plano normal que representa a la membrana celular y que adquieren estas hélices- α , está estrechamente relacionado con su actividad biológica en la célula y se ve afectado por el ligando acoplado al GPCR [12].

Figura 2.3: Estructura de un GPCR en la membrana.



A pesar de tener una baja similitud entre sus secuencias de aminoácidos, los GPCR tienen varias similitudes estructurales [10]. Las principales regiones y sus correspondientes características en los GPCR son [12]:

1. **Región extracelular.** Las principales diferencias estructurales se encuentran en el dominio extracelular y particularmente en los sitios de unión al ligando y los bucles adyacentes. Se ha encontrado que el bucle extracelular 2 (ECL 2) es el más relevante en el cambio de actividad en los GPCR, a su vez es más grande que ECL 1 y ECL 3, por lo cual puede adquirir diferentes conformaciones [4].
2. **La región transmembrana y la zona de unión al ligando.** Es la región más conservada entre los GPCR, en comparación con los dominios intra y extracelulares, en especial para aquellos de una misma familia.
Una característica relevante es el ambiente químico sumamente conservado en la hélice TM7. Esta región está involucrada en cambios conformacionales clave asociados con la activación de estas proteínas. Los ligandos interactúan con los dominios transmembranales y extracelulares.
3. **La región intracelular.** Esta región se encuentra relativamente conservada, debido a la poca diversidad de las parejas con las que un GPCR se puede enlazar en el interior de la célula (Proteínas G, arrestinas, proteínas G acopladas a receptores de cinasas, etc.).

El sitio sobre el cual actúan los ligandos en los GPCR se conoce como sitio ortoestérico, cuyo significado es sitio correcto, y que representa la región de unión ligando-proteína.

Es relevante destacar que los GPCR, como su nombre lo indica, interactúan con proteínas de unión a GDP/GTP (Guanosin difosfato/Guanosin trifosfato) que se conocen como proteínas G. Estas proteínas son las que transmiten la señal del GPCR hacia el interior de la célula y se encuentran unidas a la membrana celular mediante enlaces covalentes, a la espera de que un GPCR interactúe con ellas.

Con base en sus diferencias estructurales, podemos clasificar a los GPCR en 6 familias principales [12]; familia de la rodopsina (Clase A o I), de la secretina (Clase B o II), de los receptores metabotrópicos de glutamato (Clase C o III), receptores de feromonas fúngicas (Clase D o IV), receptores de adenosín monofosfato cíclico (Clase E o V) y familia atípica de *Frizzled* (Clase F o VI).

Adicionalmente se considera una familia extra, a la que se le denomina GPCR huérfanos cuyos miembros son aquellos GPCR que no han podido ser todavía clasificados en alguna de las otras familias [13].

2.1.2. Efecto de los ligandos en la geometría de los GPCR y diferentes actividades de ligandos.

Los GPCR pueden presentar diversos estados funcionales que pueden cambiar con base en la actividad del ligando asociado.

Los GPCR experimentan diversos cambios conformacionales al pasar a su estado activo. Estos cambios buscan crear un espacio para que el ligando se pueda unir a la proteína entre las hélices transmembranales 3, 5, 6 y 7 [12]. Los tipos de proteínas activadas en una ruta de un GPCR dependen principalmente de la molécula mensajera (el ligando), el GPCR y la proteína G activada unida al GPCR. Buena parte de la complejidad de la señalización de un GPCR se debe a las siguientes características [12]:

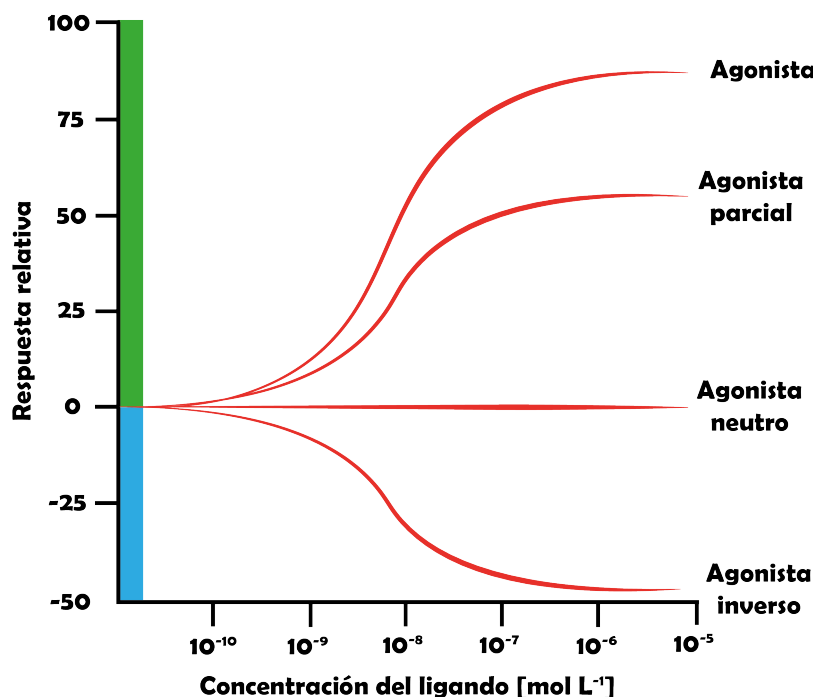
1. Un solo GPCR puede activar diferentes proteínas G, e incluso otro tipo de proteínas [14].
2. Muchos GPCR tienen un estado de actividad basal [15].
3. Un GPCR puede responder a diferentes tipos de ligandos [13].
4. La actividad de un GPCR se puede ver afectada por la composición de la membrana celular o su posición en dicha membrana con respecto a algunos compartimientos membranales [11].

Casi todas las moléculas en las células que están relacionadas con la generación de señales celulares presentan un nivel de actividad que se conoce como **actividad constitutiva** o **actividad basal**. Este es el nivel de señales generadas por una proteína, mediada por receptores, en ausencia de ligando debido al equilibrio dinámico entre los estados activos e inactivos del receptor. La mayoría de los GPCR se encuentran en un equilibrio dinámico entre su estado inactivo y activo [10]. Este equilibrio se desplaza hacia un lado o el otro en presencia de los diferentes tipos de ligandos con los que la proteína puede interactuar.

En un principio se manejaban solamente los conceptos de agonista y antagonista al momento de describir la acción de los ligandos sobre diferentes proteínas, donde un ligando agonista inducía o estabilizaba un estado “activo” en la proteína y un ligando antagonista tenía el efecto contrario. Actualmente se ha refinado ese concepto ya que se ha visto que al momento de estudiar a las proteínas *in vivo*, estas tenían un abanico de conformaciones entre los estados activo e inactivo generados por una amplia diversidad de ligandos, representados en la **imagen 2.4**, por lo que se tuvo que replantear la clasificación de los efectos de un ligando sobre las proteínas [8]. Las principales clasificaciones que actualmente se usan son:

1. **Agonista**. Estos ligandos inducen un estado activo sobre la proteína a la que se acoplan, con lo que aumentan al máximo la actividad de las señales asociadas a la proteína.
2. **Agonista parcial**. Son ligandos que por lo general inducen un estado que no es 100% activo. En estos casos la actividad de las señales debidas a la proteína, aunque no llegan a un máximo claramente se hacen muy elevadas y en algunas ocasiones particulares estos ligandos pueden inducir un estado completamente activo.
3. **Antagonistas**. En esta clasificación se encuentran los ligandos que bloquean la actividad de ligandos agonistas y desfavorecen la activación del receptor. Esto lo hacen al evitar que el GPCR se pueda activar al impedir que los ligandos agonistas se unan a la proteína.
4. **Agonistas inversos**. Son ligandos que disminuyen la actividad constitutiva de la señal asociada a la proteína al estabilizar una conformación inactiva.
5. **Antagonistas neutros**. Son antagonistas que inhiben los efectos agonistas pero que no intervienen en la actividad constitutiva de la señal de la proteína. De esta forma los ligandos agonistas no pueden interactuar con el GPCR y no lo pueden activar.

Figura 2.4: Curvas de respuesta ante diferentes actividades de ligandos



Adicionalmente los GPCR pueden ser regulados alostéricamente por moléculas que se unen a un sitio distinto al sitio ortoestérico (que significa sitio correcto en griego), el sitio ortoestérico corresponde a la región de unión entre ligandos agonistas, antagonistas y los descritos previamente. Estos reguladores alostéricos modulan la actividad de los ligandos ortoestéricos y por ende afectan el estado funcional de la proteína [8], y se pueden clasificar en:

1. **Moduladores alostéricos negativos (MAN, o por sus siglas en inglés NAM).**

Disminuyen la actividad de los ligandos ortoestéricos y también disminuyen la estabilidad relativa del estado activo basal.

2. **Moduladores alostéricos positivos (MAP, o por sus siglas en inglés PAM).**

Aumentan la actividad de los ligandos ortoestéricos, además pueden activar al receptor en ausencia de agonistas ortoestéricos.

Estos ligandos se conocen como ligandos alostéricos y pueden ser tanto endógenos como exógenos.

Los ligandos que interactúan en los sitios ortoestéricos y en el sitio alostérico se conocen como ligandos bitópicos, y pueden ser agonistas o antagonistas.

Cuando se empezó a conocer las estructuras de estas proteínas, no se conocía cómo era que los ligandos interactuaban con los GPCR para generar una respuesta. Fue hasta que los avances tecnológicos permitieron estudiar las interacciones ligando-proteína con técnicas como difracción de rayos X o resonancia magnética nuclear, que dichas estructuras se empezaron a elucidar. Aunado a esto, algunos estudios bioquímicos han indicado que el estado del GPCR se debe a movimientos en las hélices transmembranales [16, 17].

Se han usado diversas técnicas para representar las interacciones entre los ligandos y las proteínas, como es el caso de la simulación de dinámica molecular, búsquedas conformacionales, acoplamiento flexible, entre otros. Las técnicas descritas previamente han resultado favorables en la búsqueda de nuevas estructuras químicas que tengan actividad biológica, ya que las interacciones ligando-GPCR afectan la conformación de la proteína y mediante este cambio se generan diversas señales intracelulares.

2.1.3. Efecto de los ligandos en el arreglo de las hélices alfa.

A pesar de las diferencias entre los diferentes receptores, se ha visto que las conformaciones activas e inactivas de estas proteínas presentan características en común. Por ejemplo, el movimiento relativo entre la hélice 5 y 6 es fundamental para la unión a los ligandos, y el cambio a una conformación

activa por parte de la proteína. Por otro lado, los movimientos en las hélices 3 y 7 dependen más del tipo de receptor y ligando que tengamos presente y su rol en la activación de los GPCR no ha quedado completamente elucidado [10].

Otro factor que afecta el estado de activación de los GPCR es el rearrreglo de iones y moléculas de agua asociadas a la estructura de la proteína. También es relevante que el movimiento global de las hélices se ve acompañado de pequeños cambios en la geometría de los aminoácidos en las hélices. Estos cambios favorecen la interacción entre residuos específicos lo que ayuda a determinar el estado activo o inactivo de la proteína. Por ejemplo, en los GPCR tipo A, al usar la numeración relativa (sección 2.1.5), la Arginina 50 de la hélice 3 ($Arg^{3.50}$), que tiene un porcentaje de conservación del 96%, forma un puente iónico con un residuo vecino Aspartato o Glutamato ($Asp(Glu)^{3.49}$), que tienen un porcentaje de conservación del 68% y 20% respectivamente. Dicho enlace es característico de los estados inactivos de GPCR tipo A [10].

Estos cambios en los estados conformacionales de los GPCR son clave al momento de predecir el estado funcional de la proteína. Por esta razón la construcción de las redes neuronales planteadas en este trabajo consideró este punto al momento de ser estructuradas.

2.1.4. Familia de GPCR tipo A, familia de la Rodopsina

Esta familia consiste de aproximadamente 700 GPCR en humanos y se subdivide en 4 subgrupos principales ($\alpha - \delta$) que tienen una gran similitud en sus secuencias, siendo esta mayor al 25% entre los diferentes subgrupos. Dentro de cada uno de los subgrupos de la familia de la Rodopsina la similitud entre las secuencias correspondientes a ese subgrupo es mayor al 30% y frecuentemente tienen ligandos en común que generan el mismo efecto en la proteína [10]. Dentro de las secciones no homologas de estas proteínas, la mayor fuente de variaciones se ha encontrado mediante cristalografía de rayos X en los bucles de las zonas extracelulares que conectan a las diferentes hélices- α [10].

2.1.5. Numeración de Ballesteros-Weinstein

En este trabajo se busca representar a estas proteínas como datos numéricos que puedan ser procesados por una red neuronal, por lo cual conviene usar alguna abstracción de los aminoácidos en las proteínas. La principal información de interés es las coordenadas de los átomos en las siete hélices de cada proteína. Bajo este enfoque resulta muy útil usar la numeración establecida por Ballesteros-Weinstein [18, 19] donde se numeran los residuos de las hélices a partir del residuo más conservado en cada hélice, a cada residuo se le asigna una coordenada del tipo $n.m$ donde n es el número de la hélice y m es el número del residuo, siendo el más conservado $m = 50$, se suele escoger este número ya que por lo general las hélices alfa de los GPCR tienen menos de cien residuos. Los residuos hacia el extremo carboxilo aumentan la numeración de m en uno por cada aminoácido hacia el extremo y los residuos hacia el extremo amino disminuyen la numeración de m de la misma forma.

De acuerdo a lo reportado por Isberg [19] la **tabla 2.1** condensa los residuos más conservados por hélice para la familia de GPCR A.

Cuadro 2.1: Residuo más conservado en los GPCR de la familia A.

| Hélice | Residuo más conservado | Numeración m |
|--------|------------------------|----------------|
| TM1 | N (Asparagina) | 1.50 |
| TM2 | D (Ácido aspártico) | 2.50 |
| TM3 | R (Arginina) | 3.50 |
| TM4 | W (Triptófano) | 4.50 |
| TM5 | P (Prolina) | 5.50 |
| TM6 | P (Prolina) | 6.50 |
| TM7 | P (Prolina) | 7.50 |

2.2. Representación de las proteínas

Debido al diseño de las redes neuronales, cualquier grupo de datos usados para entrenar a estos algoritmos debe primero transformarse a una representación numérica que pueda abstraer de forma adecuada la información de los datos.

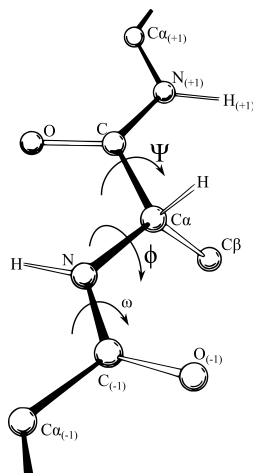
En esta sección se hace un breve repaso de algunos de los métodos usados para lograr dichas abstracciones y poder usar esos datos en el entrenamiento de la red neuronal.

2.2.1. Cadena principal de las proteínas.

Dado que una proteína es un polímero de aminoácidos donde hay una secuencia que se repite periódicamente ($N - C\alpha - C$), podemos representar a las proteínas con base en su cadena principal o columna vertebral, véase la **figura 2.5**, donde sólo se considera la unidad repetida entre los aminoácidos.

Figura 2.5: Cadena principal de la proteína. Imagen de Jane S. Richardson.

https://commons.wikimedia.org/wiki/File:Protein_backbone_PhiPsiOmega_drawing.jpg.



Para fines de este trabajo es importante destacar cuales son el $C\alpha$ y el $C\beta$. Ambas posiciones reciben su nombre de acuerdo a la nomenclatura usada en química orgánica para los grupos funcionales, en especial para el grupo carbonilo, **figura 2.6**. Donde el primer carbono al lado del grupo funcional será el α , luego lo seguirá el β y así sucesivamente. En el particular caso de los aminoácidos **figura 2.7** el $C\alpha$ corresponde al carbono unido al grupo funcional del aminoácido y el $C\beta$ corresponde al primer carbono del grupo funcional, en el caso particular de la glicina no hay $C\beta$.

Figura 2.6: Nomenclatura $C\alpha$ y $C\beta$.

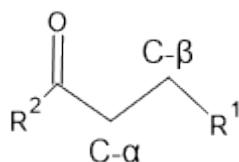
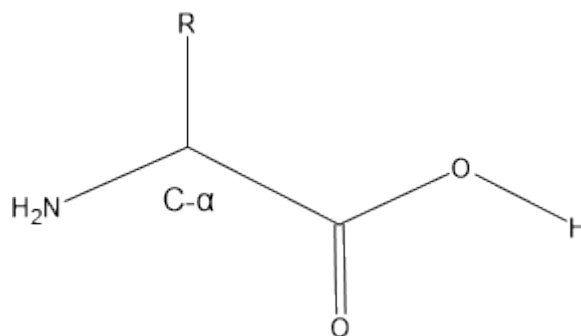


Figura 2.7: Estructura de un aminoácido.



2.2.2. Mapas de distancias de las proteínas

La información estructural de las proteínas se suele representar en un sistema de coordenadas cartesianas. Una forma alternativa de representar esta información es usar matrices bidimensionales

construidas con base en las distancias euclidianas entre diferentes elementos de las proteínas. Esta representación se conoce como **mapas de distancias** [20].

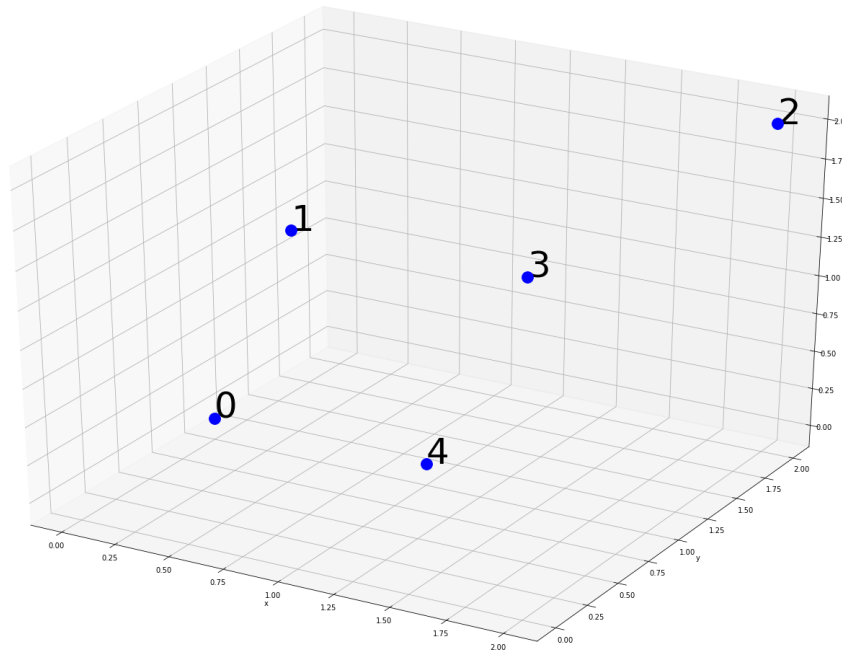
Para construir una matriz de distancias se debe de definir entre cuáles puntos se tomarán las distancias euclidianas. Tres opciones comunes para estos puntos son:

1. La distancia entre átomos específicos en la proteína. Conviene usar a los átomos de la cadena principal, como por ejemplo los $C\alpha$.
2. La distancia más corta entre los átomos del residuo i y los átomos del residuo j .
3. La distancia entre los centros de masa de los diversos residuos en la proteína.

Las matrices de distancias se construyen al medir el espacio entre cada elemento con respecto a todos los demás, por lo que en una lista de n elementos obtendremos una matriz de n^2 elementos.

Por ejemplo, consideremos los siguientes puntos en tres dimensiones mostrados en la **figura 2.8**

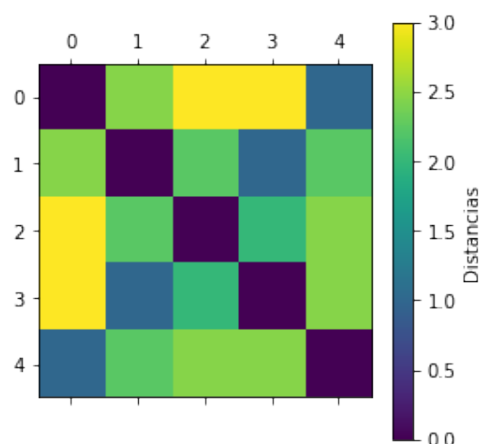
Figura 2.8: Representación de 5 puntos tridimensionales.



Ahora la matriz de distancias y su representación gráfica quedarían de las siguientes formas

$$Distancias = \begin{pmatrix} 0.00 & 2.45 & 3.00 & 3.00 & 1.00 \\ 2.45 & 0.00 & 2.24 & 1.00 & 2.24 \\ 3.00 & 2.24 & 0.00 & 2.00 & 2.45 \\ 3.00 & 1.00 & 2.00 & 0.00 & 2.45 \\ 1.00 & 2.24 & 2.45 & 2.45 & 0.00 \end{pmatrix} \quad (2.1)$$

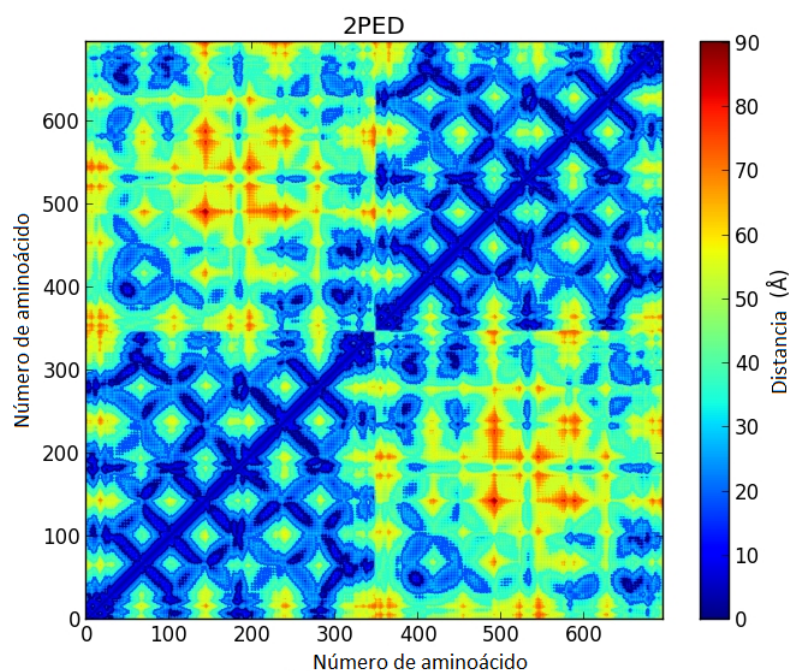
Figura 2.9: Mapa de distancias de la matriz 2.1



Por lo general un mapa de distancias para una proteína luce como el que se presenta en la **figura 2.10**, donde los ejes representan a la cadena de aminoácidos de la proteína.

Figura 2.10: Mapa de distancias del GPCR 2PED, obtenido con nanoHUB

<https://nanohub.org/resources/contactmaps>.



2.3. Redes neuronales artificiales

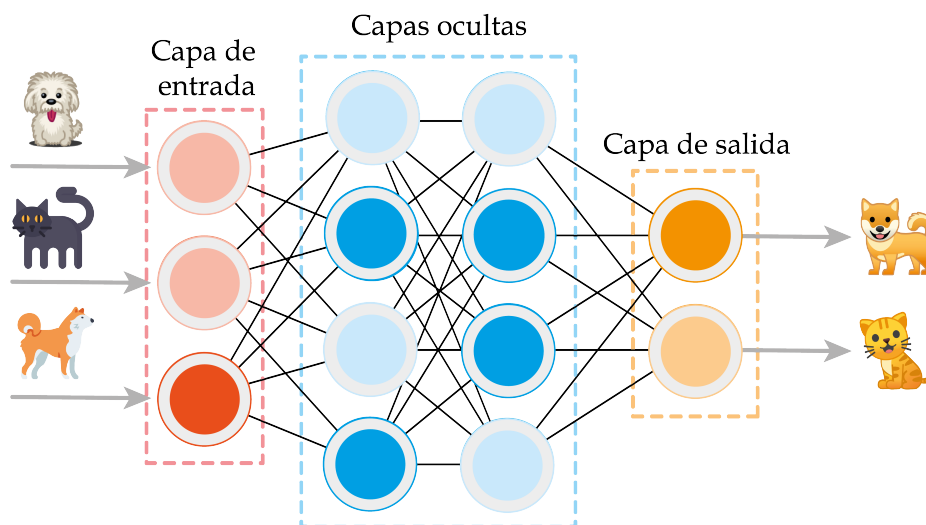
Las redes neuronales son modelos computacionales inspirados por las redes neuronales biológicas [21]. Estos modelos están conformados por unidades de cálculo o nodos, llamados neuronas e interconectadas entre ellas las cuales generan un valor de salida que puede ser interpretado para acciones posteriores ya sea por parte de una siguiente neurona, por un ser humano o un programa en una computadora si es el resultado final de la red neuronal.

Para realizar el aprendizaje de una red neuronal se busca optimizar una función de costo (**sección 2.3.4**), a través de varias iteraciones, conocidas como épocas de entrenamiento o simplemente épocas (**sección 2.3.7**), en las cuales la red neuronal propone una serie de parámetros para sus neuronas

(sección 2.3.1), evalúa la función de costo y mediante un algoritmo de optimización se ajustan los parámetros de las neuronas para obtener una mejor predicción. Este aprendizaje ha permitido resolver diversos problemas de clasificación de datos en diferentes categorías, como sería el caso de poder clasificar imágenes de gatos o perros como en la **figura 2.11**. También ha permitido identificar patrones en muestras de datos como imágenes o audio [22].

A continuación se presenta un esquema simplificado de una red neuronal, véase la **figura 2.11**. Al inicio de la red hay una entrada de datos, representada por la flecha de entrada, que pueden ser los datos de entrenamiento, o bien pueden ser las salidas de otra red neuronal. Cada neurona en la red está representada por un círculo, siendo los de color azul las neuronas de las capas ocultas. Las neuronas de las capas ocultas son aquellas neuronas en capas interiores de la red neuronal encargadas de abstraer la información que se le transfiere a la red neuronal. En el caso representado sólo se tienen dos capas ocultas. Los círculos naranjas corresponden a la capa de neuronas de entrada y los círculos amarillos simbolizan a las neuronas de salida. Las flechas grises representan el flujo de información en la red neuronal, donde se remarca que algunas veces las salidas de algunas neuronas son las entradas de otras. Al observar el número de neuronas en la capa final es posible determinar el número de categorías en las que se clasifican los datos de entrada, en este ejemplo la red neuronal es binaria ya que puede clasificar a los datos de entrada en dos categorías. Por último, las líneas grises entre los círculos describen las conexiones entre las neuronas, donde por lo general los valores de salida de una neurona se pasan a cada una de las siguientes neuronas en la capa posterior, estas conexiones pueden variar dependiendo del tipo de arquitectura de la red neuronal, **sección 2.3.6**.

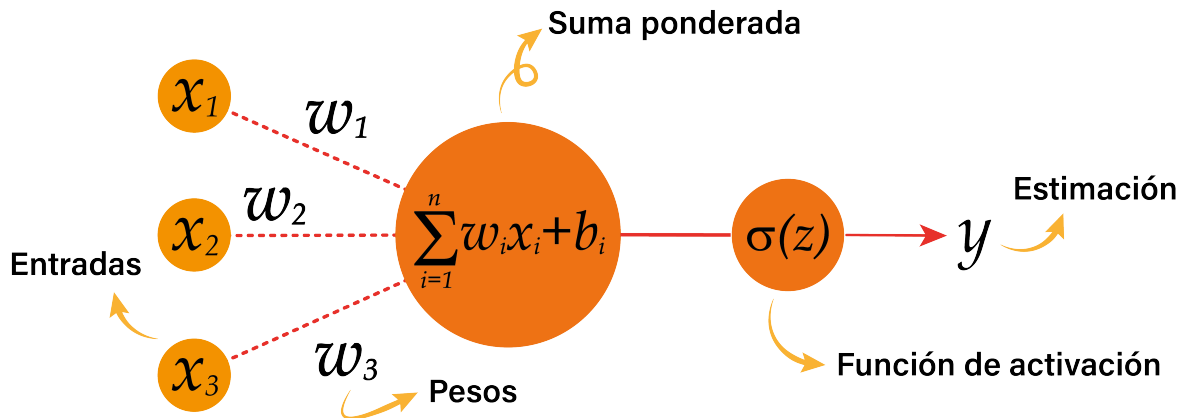
Figura 2.11: Esquema simplificado de una red neuronal.



2.3.1. La neurona

Lo que hace una red neuronal es clasificar un grupo de elementos, a través del procesamiento de representaciones numéricas de los elementos. Este proceso se logra al utilizar unidades de cálculo denominadas neuronas, siendo estas la unidad fundamental de las redes neuronales. Podemos ver una representación de una neurona en la **figura 2.12**.

Figura 2.12: La neurona y sus componentes.



Como vemos la neurona está compuesta de dos funciones matemáticas; **ecuación 2.2**, que representa una suma ponderada de todos los n valores de entrada, y la **ecuación 2.3** que corresponde a la función de activación, **sección 2.3.3**, que se le aplica a la función lineal z resultante.

$$z = \sum_{i=1}^n w_i x_i + b_i = W^T X + B \quad (2.2)$$

$$y = \sigma(z) \quad (2.3)$$

Donde:

w_i = Peso i

b_i = Sesgo i

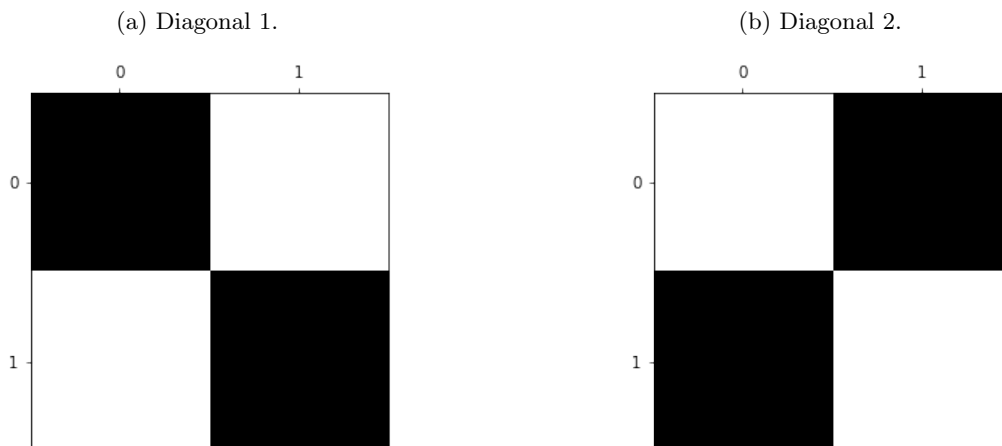
W^T = Vector transpuesto de pesos

B = Vector de sesgos

Los valores en x son la representación numérica de los datos de entrada y se pueden agrupar como un vector de características con una dimensionalidad l ($x \in \mathbb{R}^l$), donde l corresponde al tamaño del vector de datos de entrada de la neurona. Por otro lado, los valores de y se pueden agrupar en otro vector de dimensionalidad m ($y \in \mathbb{R}^m$), donde m corresponde al número de clasificaciones posibles para los datos usados. Es por eso que la **ecuación 2.2** también se puede representar como el producto de matrices.

Por ejemplo, pensemos que queremos clasificar imágenes de diagonales mediante redes neuronales y que los tipos de datos a clasificar pueden estar en dos categorías, como las de las **imágenes 2.13a** y **2.13b**, donde los valores sobre los ejes corresponden a las coordenadas y no al valor del pixel.

Figura 2.13: Imágenes de diagonales a clasificar.



Estás imágenes se pueden representar como matrices, donde el 1 significa que el píxel se encuentra prendido, blanco, y el 0 significa que se encuentra apagado, negro. La representación tanto en matriz como en vector para la primera diagonal sería

$$\begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \end{bmatrix},$$

mientras que para la segunda diagonal nos quedaría

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \end{bmatrix}.$$

Si se tuvieran tres ejemplos para entrenar una neurona, se tendría la siguiente ecuación:

$$\begin{bmatrix} w_1 & w_2 & w_3 & w_4 \end{bmatrix} \times \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 1 \end{bmatrix} = \begin{bmatrix} 0 \cdot w_1 + 1 \cdot w_2 + 1 \cdot w_3 + 0 \cdot w_4 & 1 \cdot w_1 + 0 \cdot w_2 + 0 \cdot w_3 + 1 \cdot w_4 & 1 \cdot w_1 + 0 \cdot w_2 + 0 \cdot w_3 + 1 \cdot w_4 \end{bmatrix},$$

donde el vector W^T representa los pesos de las diferentes entradas a la neurona y la matriz X representa una matriz donde cada columna corresponde a un caso de entrenamiento. Ahora aplicamos las sumas y obtenemos la siguiente matriz

$$\begin{bmatrix} \alpha & \beta & \gamma \end{bmatrix}.$$

Cuyos valores están definidos por las siguientes ecuaciones

$$\begin{aligned} \alpha &= 0 \cdot w_1 + 1 \cdot w_2 + 1 \cdot w_3 + 0 \cdot w_4 \\ \beta &= 1 \cdot w_1 + 0 \cdot w_2 + 0 \cdot w_3 + 1 \cdot w_4 \\ \gamma &= 1 \cdot w_1 + 0 \cdot w_2 + 0 \cdot w_3 + 1 \cdot w_4 \end{aligned}$$

Si le sumamos el vector B , que es un vector con los valores de b para cada neurona, obtendríamos

$$\begin{bmatrix} \alpha & \beta & \gamma \end{bmatrix} + \begin{bmatrix} b_1 & b_2 & b_3 \end{bmatrix} = \begin{bmatrix} z_\alpha & z_\beta & z_\gamma \end{bmatrix}.$$

Recordemos que cada columna representa un caso en el entrenamiento y en esas condiciones cada z representa un vector de $l \times 1$. El siguiente paso sería aplicar la función de activación

$$\mathbf{y} = \sigma(z) = \begin{bmatrix} \sigma(z_\alpha) & \sigma(z_\beta) & \sigma(z_\gamma) \end{bmatrix} = \begin{bmatrix} y_\alpha & y_\beta & y_\gamma \end{bmatrix},$$

donde cada y es un vector de $m \times 1$, que corresponde a un valor de predicción para cada entrada.

2.3.2. Hiperparámetros

Así como una red neuronal tiene una serie de parámetros, aquellos valores que la red neuronal va a configurar por su cuenta como w y b , también tiene un grupo de valores conocidos como hiperparámetros. Estos hacen referencia a todos los valores en la red neuronal que la persona que construye la red neuronal puede fijar y también se pueden modificar, tales como:

- Número de unidades ocultas. (sección 2.3.6)

- Número de capas. (**sección 2.3.6**)

- Valores asociados a los algoritmos de optimización (α, β, γ) . (**sección 2.3.5**)

- Arquitectura de la red neuronal. (**sección 2.3.6**)

- Épocas de entrenamiento. (**sección 2.3.7**)

- Función de activación. (**sección 2.3.3**)

- Algoritmo de optimización. (**sección 2.3.5**)

Aunque la mayoría de los términos previamente citados no tienen sentido en este momento para el lector, se explicarán a lo largo de las siguientes secciones.

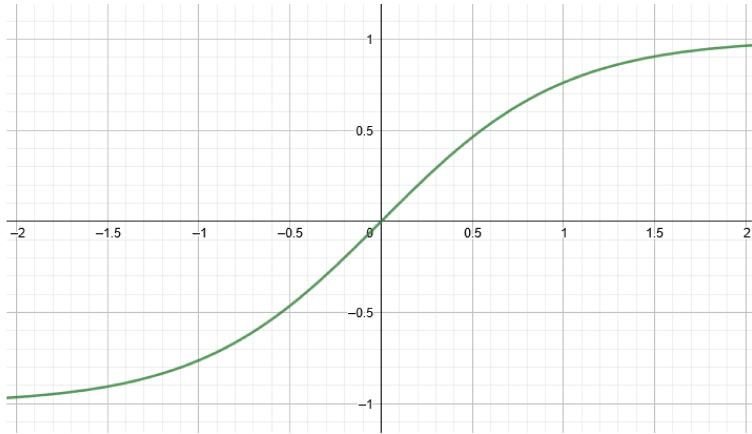
2.3.3. Funciones de activación

La función de activación en una neurona corresponde a la **ecuación 2.3**, que es una función no lineal y que busca generar una compuerta lógica para prender o apagar neuronas en la red. Sin ese paso, tendríamos una función lineal, **ecuación 2.2**, compuesta de funciones lineales que corresponderían a cada neurona en la red. Esto provocaría que toda la red neuronal colapsara y por lo tanto se comportaría como una sola neurona al dar como resultado una función lineal, con lo cual se perdería la capacidad de distinción característica de las redes neuronales [23].

La principal característica de las funciones de activación es que deben polarizar a los datos procesados, por lo que al pasar por la función de activación estos valores se van a aproximar hacia uno u otro valor. Los patrones menos relevantes para la red neuronal tenderán hacia un valor mientras que los datos y patrones de los que se puede generalizar para realizar una predicción tenderán hacia otro valor. Por ejemplo en una función de activación llamada *ReLU*, **figura 2.14 (b)**, (de las siglas en inglés para *Rectified Linear Unit*) los valores menos relevantes tienden hacia cero y en caso contrario se hacen positivos. Otras funciones de activación comunes son la sigmoidea, presentada en la **figura 2.14 (a)** y la tangente hiperbólica, representada en la **figura 2.14 (c)**

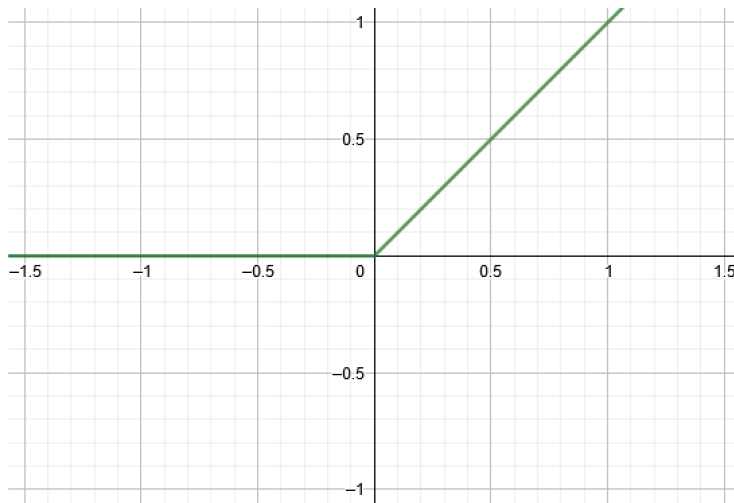
En este punto vale la pena destacar que al inicio de una clasificación en una red neuronal no sabemos cuáles son los datos relevantes para la red, no conocemos los parámetros. Estos valores serán obtenidos por la misma red al realizar numerosas iteraciones al momento de ser 'entrenada', un proceso que se describe en la **sección 2.3.7**, y evaluar sus predicciones obtenidas comparándolas con los valores 'reales' que le hemos dado a la red neuronal haciendo uso de las denominadas funciones de costo, que explicaremos en la siguiente sección.

Figura 2.14: Algunos ejemplos de funciones de activación: (a) tanh, (b) ReLU y (c) Sigmoidea



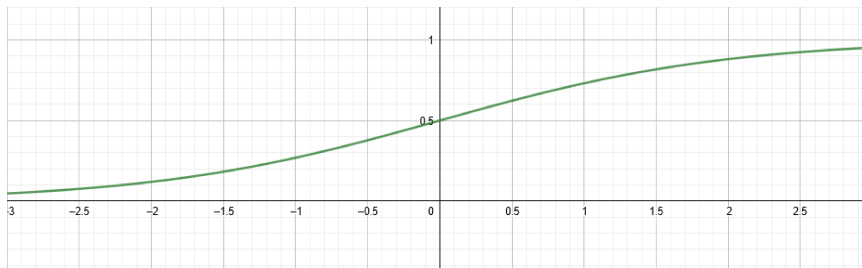
(a) Tangente hiperbólica (Tanh)

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



(b) Unidad rectificada lineal (Rectified Linear Unit, ReLU)

$$\sigma(z) = \frac{z + |z|}{2}$$



(c) Sigmoidea

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

2.3.4. Función de pérdida y de costo

Una vez que se ha descrito los elementos básicos que construyen nuestra red neuronal es posible ensamblarlos y entrenar a la red generada para después realizar predicciones. Dado que nuestra red neuronal, también denominada como modelo de aprendizaje, es supervisado podemos verificar qué tan cercanas son las predicciones y de la red neuronal de los valores verdaderos \hat{y} obtenidos de las clasificaciones de los datos de entrenamiento. Esta verificación la podemos realizar mediante la función de pérdida[21, 1], **ecuación 2.4**,

$$L(y, \hat{y}) = -[\hat{y} \cdot \log(y) + (1 - \hat{y})\log(1 - y)]. \quad (2.4)$$

Con base en la ecuación de la función de pérdida, **ecuación 2.4**, es posible formular la función de coste representada en la **ecuación 2.5**, que es el promedio de las funciones de pérdida del sistema y con la cual podemos evaluar el desempeño global de la red neuronal, donde m es el número total de muestras usadas.

$$J(y, \hat{y}) = \frac{1}{m} \sum_{i=1}^m L(y_i, \hat{y}_i). \quad (2.5)$$

El modelo anterior para calcular la pérdida también se conoce como pérdida por entropía cruzada binaria dado que aplica el concepto de la entropía en la teoría de la información (o entropía de Shannon) [24], **ecuación 2.6**, donde la entropía para un mensaje $H(x)$, es el valor medio de la cantidad de información de los diversos estados del mensaje, donde $p(x_i)$ es la probabilidad de ocurrencia de cada evento.

$$H(x) = \begin{cases} \sum_i^n p(x_i) \log_2 \left(\frac{1}{p(x_i)} \right) & \text{si } x \text{ es una variable discreta} \\ \int_i^n p(x_i) \log_2 \left(\frac{1}{p(x_i)} \right) dx & \text{si } x \text{ es una variable continua} \end{cases} \quad (2.6)$$

Para casos donde se contempla una clasificación binaria el modelo de entropía cruzada binaria es muy bueno, en este caso tenemos tres clasificaciones para los estados de los ligandos, por lo que es mejor usar un modelo de entropía cruzada categórica, **ecuación 2.7**, y que es muy similar a la entropía cruzada binaria. La ecuación que define a la entropía categórica es la siguiente

$$L_{EC} = - \sum_{i=1}^n \hat{y}_i \log y_i, \quad (2.7)$$

en este caso la función de coste sigue siendo el promedio de las funciones de pérdida.

El valor de la función de costo $J(y_i, \hat{y}_i)$ se aproximará a cero conforme la red neuronal aumente su exactitud en sus predicciones, dado que \hat{y} y y tendrán el mismo valor.

2.3.5. Algoritmos de optimización

Para disminuir el valor de la función de coste se implementan los algoritmos de optimización [21]. El principal algoritmo de optimización, y del cual parten una gran variedad de algoritmos, es el descenso del gradiente representado en las **ecuaciones 2.8**; Donde dada la función de coste buscamos encontrar valores de w y b que minimicen $J(w, b)$. Esto se logra al usar un algoritmo recursivo, lo que significa que se repetirá un determinado número de veces hasta cumplir una de las siguientes dos condiciones: La función de costo $J(w, b)$ llega a un valor umbral muy cercano a cero, o bien si cumple un cierto número de pasos. De esta forma los valores de w y b se optimizan de forma iterativa de acuerdo a la siguiente ecuación

$$\begin{aligned} w_i &:= w_{i-1} - \alpha \frac{\partial J(w, b)}{\partial w} \\ b_i &:= b_{i-1} - \alpha \frac{\partial J(w, b)}{\partial b}, \end{aligned} \quad (2.8)$$

aquí el parámetro α es conocido como la tasa de aprendizaje y nos indica qué tanto avanzamos en cada paso del algoritmo. Dicho en otras palabras ‘qué tan grande es el paso’. Este parámetro debe de ser seleccionado con cuidado ya que de ser muy pequeño hará que sea más costoso el cálculo de este algoritmo, aparte de disminuir la rapidez de convergencia. En caso de que α sea muy grande podemos dificultar la convergencia a valores mínimos de la función de coste.

Un problema que puede surgir al buscar un mínimo en la función de coste es quedarse atascados en un mínimo local en vez de un mínimo global, también se puede presentar que por el tamaño del paso el algoritmo nunca logre converger a un mínimo. Para lidiar con estos problemas se suelen aplicar técnicas adaptativas, para variar el parámetro de la tasa de aprendizaje en función del gradiente u otros hiperparámetros (los hiperparámetros son valores seleccionados en la construcción en la red neuronal, **sección 2.3.2**) creados específicamente para variar el valor de la tasa de aprendizaje.

En este trabajo se usó un optimizador que implementa el algoritmo Adadelta [25]. Este es un método de descenso del gradiente que se basa en la tasa de aprendizaje adaptativo por dimensión para resolver dos problemas que suelen surgir en los algoritmos de optimización:

1. La disminución de las tasas de aprendizaje a lo largo del entrenamiento.
2. La necesidad de una tasa de aprendizaje global que tiene que ser seleccionada manualmente.

Este algoritmo surge como una extensión de otro algoritmo de optimización llamado *Adagrad* [25] y su principal diferencia es que *Adadelta* adapta la tasa de aprendizaje α por lo que con este algoritmo la red neuronal sigue aprendiendo, al mejorar sus parámetros, aun cuando se han realizado muchas épocas.

2.3.6. Arquitectura de la red neuronal

Una vez que tenemos todos los elementos para construir la red neuronal y optimizarla para mejorar su capacidad de predicción, podemos cambiar la manera en que las neuronas están conectadas entre ellas. A esta forma de conectar neuronas se le conoce como arquitectura de la red neuronal. Existe una gran diversidad de arquitecturas y cada año se crean nuevas arquitecturas, Fjodor Van Veen ha hecho un gran trabajo manteniendo un recuento de estas arquitecturas y las ha juntado en lo que se conoce como un "zoológico de arquitecturas" [26] y basados en su infografía, presentamos las dos arquitecturas usadas en este trabajo en la **figura 2.15**.

La elección de la arquitectura varía en función del tipo de problema que se quiere solucionar. Por ejemplo *AlexNet CNN* [27], que es una red neuronal convolucional, se ha probado muy útil para clasificar imágenes. Existen muchas arquitecturas de redes neuronales, pero en este trabajo sólo se usaron 2 modelos y el modelo que dio los mejores resultados fue una red neuronal convolucional.

2.3.7. Entrenamiento de una red neuronal y tipos de entrenamiento

En el caso de una red neuronal, entendemos por entrenamiento al proceso de ejecutar diversas épocas, también conocidas como épocas de entrenamiento. Donde una época consiste en la evaluación de los parámetros (como los pesos w_i , el sesgo b , etc.) en las neuronas de la red neuronal (**sección 2.3.4**). Luego se optimizan los parámetros mediante el algoritmo de optimización usado en la red (**sección 2.3.5**) para actualizar dichos valores y nos permiten obtener nuevos valores para los parámetros que llevarán a una mejor capacidad de predicción de la red neuronal.

Aquí surge una disyuntiva, dependiendo de la cantidad y tipo de datos de los que disponemos podemos escoger entre tres opciones principales para los modelos de aprendizaje de máquina [28]:

1. **Entrenamiento supervisado**, es el que se realiza cuando contamos con un grupo de datos de entrenamiento previamente etiquetados y clasificados en las clases que nos interesa predecir. En este caso el modelo, la red neuronal, aprende a predecir los valores de los parámetros (w , b , etc.) al tratar de disminuir su error al comparar sus predicciones con las clasificaciones de los datos de entrenamiento y así obtener los valores óptimos de los parámetros, con los cuales obtener la mejor tasa de acierto posible.
2. **Entrenamiento no supervisado**, es el que se realiza cuando solamente tenemos los datos de entrenamiento sin clasificar, por lo que no contamos con ningún tipo de etiqueta con la cual diferenciar a los datos. Este tipo de algoritmos busca como separar a los datos provistos en un número de diferentes categorías, mediante la detección de patrones comunes en los ejemplos con los que el modelo trabaje. Cabe destacar que en estos casos por lo general se requiere grupos de datos muy grandes, del orden de decenas o cientos de miles de datos de entrenamiento.
3. **Entrenamiento por refuerzo**, en este caso se plantea un sistema acción-recompensa, en el que cada vez que el algoritmo acierte en su clasificación recibirá una señal que le indicará su acierto y optimizará su algoritmo para favorecer dichas señales de recompensa. En caso de que la predicción sea errónea, el algoritmo recibirá una señal que indique su error para que actualice sus parámetros al buscar aumentar el número de señales asociadas con recompensas.

En este trabajo, dado que se cuenta con una cantidad limitada de estructuras de GPCR, se optó por un modelo de entrenamiento supervisado.

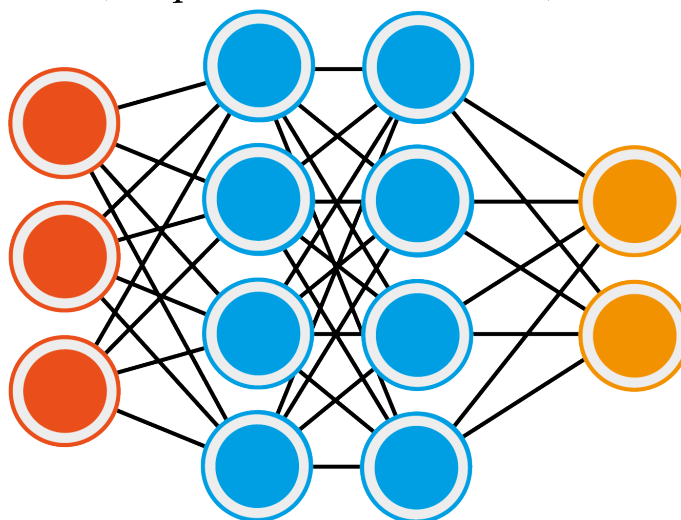
2.3.8. Redes Neuronales Convolucionales

Estas son una arquitectura de redes neuronales de avance o aprendizaje profundo que también entra en los algoritmos de inteligencia artificial. Al ser aprendizaje profundo estas arquitecturas constan de un gran número de capas de neuronas ocultas. Aunque el término aprendizaje profundo o en inglés *deep learning* es ampliamente usado, no hay una definición clara de en qué punto una red neuronal artificial se hace profunda. Jürgen Schmidhuber hace un estudio en el cual propone una medida denominada CAP, siglas de *Credit Assignment Paths*, que es una medida de las posibles

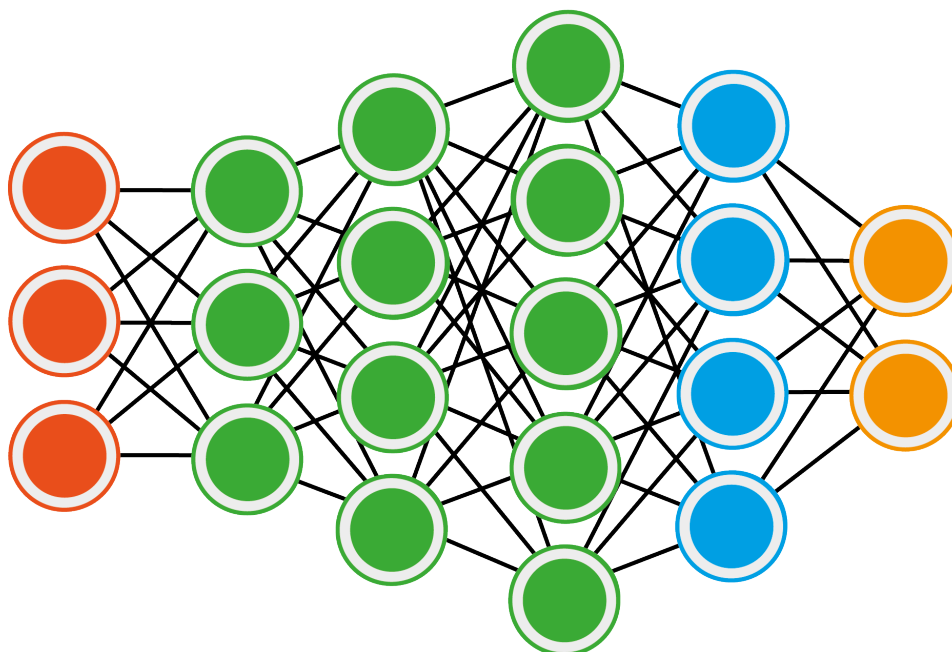
Figura 2.15: Arquitecturas usadas.

- Neurona de entrada
- Neurona oculta
- Neurona de salida
- Neurona convolucional con filtro

Red neuronal profunda
(*Deep Feed Forward, DFF*)



Red neuronal convolucional
(*Convolutional Neural Network, CNN*)



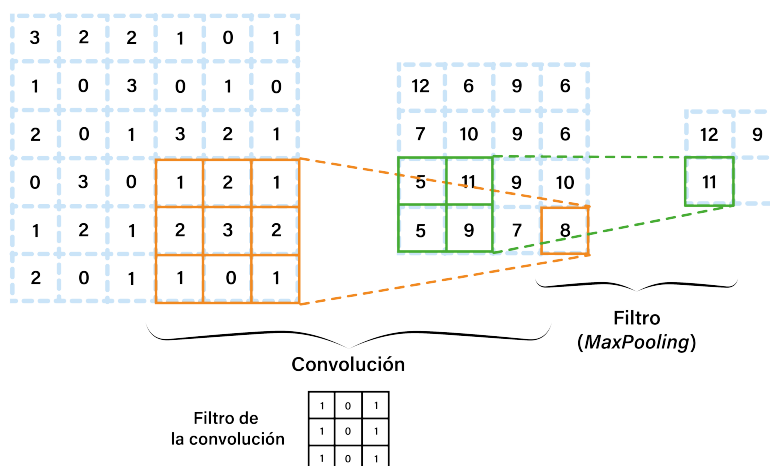
operaciones desde la capa de entrada hasta la capa de salida en una red neuronal, y con la cual se puede determinar si un modelo es profundo. En su artículo se proponen algunas arquitecturas con un CAP>4 que son ampliamente reconocidas como redes neuronales profundas [22]. Las redes neuronales convolucionales, o CNN de sus siglas en inglés *Convolutional Neural Networks*, se han mostrado muy útiles al momento de realizar clasificación de imágenes [21]. El núcleo de estas redes es un proceso llamado convolución, que consiste en aplicar una matriz llamada filtro o núcleo a una imagen (lo cual es otra matriz). Esto se puede hacer para extraer las características más relevantes de una imagen y en el proceso se puede disminuir el tamaño de la matriz, lo cual resulta bastante útil computacionalmente cuando se manejan grupos de cientos de miles de imágenes, ya que disminuye drásticamente la dimensión de los datos que se tendrán que calcular y por ende el proceso se acelera. Aunque también sirven perfectamente para trabajar con conjuntos más pequeños de datos, ya que las convoluciones permiten reconocer patrones en las imágenes y con esto mejorar la capacidad de predicción de una red neuronal.

En este tipo de redes neuronales se conserva la estructura jerárquica mientras la red aprende las representaciones de las características internas y las generaliza. Esto quiere decir que la red neuronal aprende a reconocer patrones en las imágenes. Por ejemplo, las llantas en un coche o las orejas puntiagudas de un gato, y generaliza los patrones detectados para realizar sus clasificaciones. Con esto se han podido resolver problemas de reconocimiento de objetos y visión de computadora [21]. Otra ventaja, debida a las convoluciones, es que los valores de entrada se hacen invariantes ante la rotación, inversión y otras operaciones de simetría. Lo cual permite una mayor flexibilidad al momento de procesar imágenes para la red neuronal ya que la red ahora puede detectar patrones dentro de las categorías a clasificar. En otras palabras, si nuestra red clasificara imágenes de perros, sería irrelevante la posición relativa del perro en la imagen. Situación que no necesariamente se cumple con otras arquitecturas de redes neuronales.

Diferentes capas en una Red Neuronal Convolucional (CNN de las siglas para *Convolutional Neural Networks*).

Las capas en una red neuronal convolucional se construyen a partir de mapas de imágenes, las matrices que representan a una imagen, y los filtros, lo cual se presenta en la **figura 2.16**, donde la imagen es la primera matriz, a la que se le aplicara una convolución con el filtro presentado en la imagen, después se pasara por otro filtro MaxPooling y se obtendrá una matriz final a la que denominaremos mapa de características. La forma en que las computadoras visualizan cualquier imagen es en realidad mediante una serie de matrices, una matriz para imágenes en blanco y negro y tres en caso de imágenes a color donde cada matriz representa un canal de colores RGB (Siglas de rojo, verde y azul en inglés), que con sus valores numéricos le indican a una computadora los valores que debe de adquirir cada píxel para en conjunto representar la imagen en un monitor.

Figura 2.16: Ejemplo de una convolución con un filtro MaxPooling.



Es de destacar que en una red neuronal convolucional los filtros se inicializan de forma aleatoria y la red neural aprende cuales son más eficientes para detectar patrones con base en el problema a resolver.

Para entender cómo interactúan los mapas de imágenes con los filtros supongamos que partimos del siguiente mapa de imagen:

$$\begin{pmatrix} 1.00 & 3.00 & 2.00 & 1.00 \\ 2.00 & 9.00 & 1.00 & 1.00 \\ 1.00 & 3.00 & 2.00 & 3.00 \\ 5.00 & 6.00 & 1.00 & 2.00 \end{pmatrix}, \quad (2.9)$$

y definimos dos filtros: Un filtro *Max Pooling* que regresa el valor máximo, fijamos su tamaño en 2×2 con un salto de 2 y otro filtro *Average Pooling* que regresa el valor promedio, el cual tendrá el mismo tamaño y salto que el filtro *Max Pooling* previamente definido. Al aplicar el filtro *Max Pooling* al mapa de imagen obtenemos la siguiente matriz

$$\begin{pmatrix} 9.00 & 2.00 \\ 6.00 & 3.00 \end{pmatrix}, \quad (2.10)$$

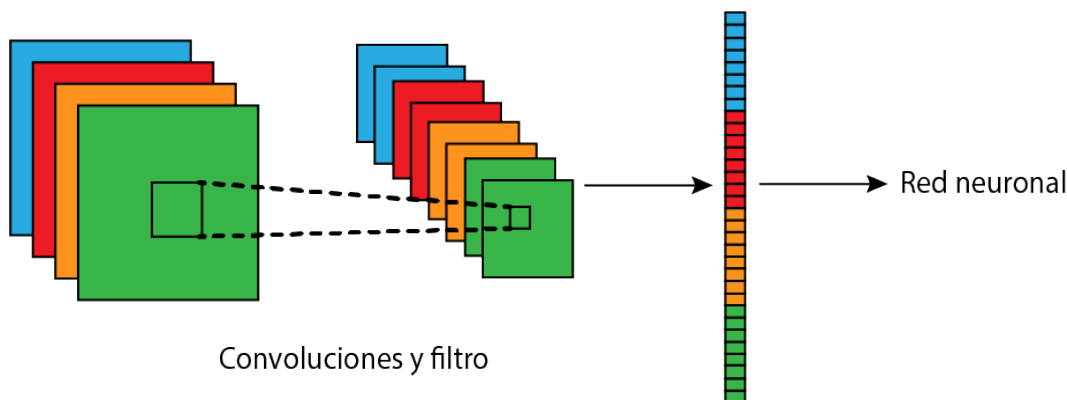
y al aplicar el filtro *Average Pooling* al mapa de imagen obtendremos la siguiente matriz

$$\begin{pmatrix} 3.75 & 1.25 \\ 3.75 & 2.00 \end{pmatrix}. \quad (2.11)$$

La imagen que se obtiene después de aplicar un filtro se llama mapa de características. Existen diferentes tipos de filtros y se pueden construir filtros que aprendan a reconocer patrones como bordes, siluetas o formas. Aunque en realidad la red neuronal es la que en sus diferentes iteraciones va construyendo los filtros que necesite para detectar patrones.

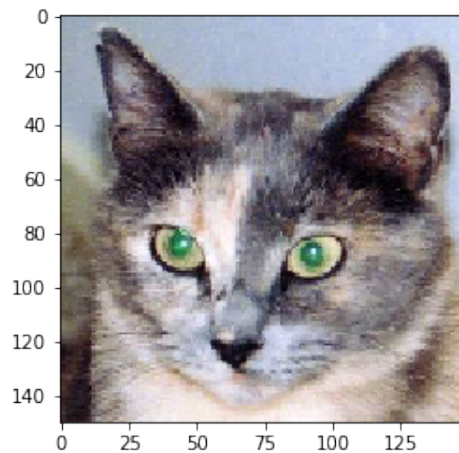
Cuando combinamos los distintos filtros con capas de convoluciones construimos la primera parte de una red neuronal convolucional. La siguiente parte transforma las imágenes obtenidas de los filtros en vectores, o matrices de $1 \times n$, para pasarlos a una red neuronal tradicional. Cada vez que se realiza una convolución la dimensión de la matriz disminuye, pero aumenta el número de matrices de acuerdo a los filtros aplicados. Por ejemplo, en la **imagen 2.17** partimos de 4 matrices de entrada y se aplican 2 filtros por convolución, luego se aplica un filtro que reduce la dimensionalidad de las imágenes con lo que se obtienen 8 matrices más pequeñas (4 matrices por 2 filtros), este proceso se suele realizar varias veces antes de convertir esas matrices en vectores y pasarlas a una red neuronal.

Figura 2.17: Proceso de convoluciones en una red neuronal convolucional.



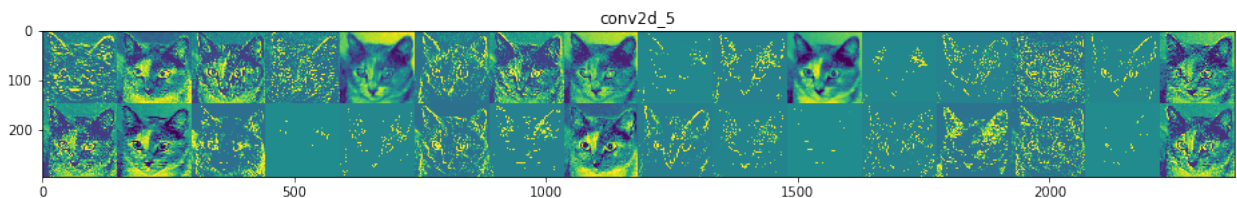
Cada filtro puede detectar diferentes características de las matrices originales y entre más convoluciones se apliquen, más abstractas serán las características abstraídas de las matrices originales. Para entender de una forma más clara el proceso de convolución podemos usar las imágenes del ejemplo de François Chollet presentado en el cuadernillo de Google Collaboratory [visualizing what convnets learn](#) [29] (consultar cuadernillo 5.4 de la primera edición) y el cual se describe a continuación. Partimos de una imagen de un gato

Figura 2.18: Imagen a procesar por una red neuronal convolucional.
Imagen de François Chollete



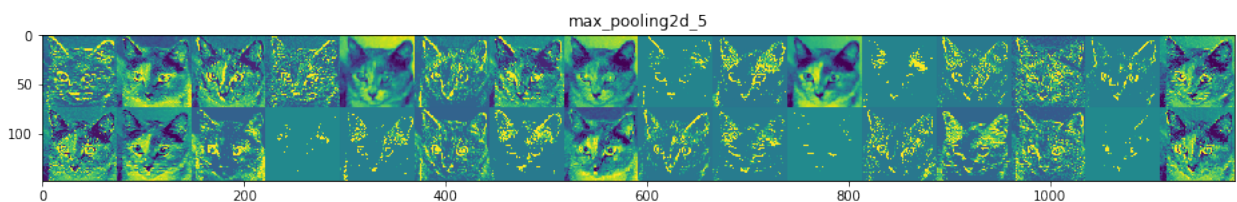
Aplicamos un filtro con 32 convoluciones en dos dimensiones, cuyos filtros son inicializados de forma aleatoria. Estas convoluciones aleatorias empezarán a detectar patrones y con el transcurso de las épocas de entrenamiento se optimizarán los filtros que generen las convoluciones que puedan destacar los patrones más significativos para lograr las clasificaciones deseadas. Al aplicar las convoluciones para una sola imagen se obtienen 32 nuevas imágenes, cada una de ellas es el resultado de una convolución (filtro) diferente, como se muestra en la **figura 2.19**.

Figura 2.19: Imágenes obtenidas tras 32 convoluciones.
Imagen de François Chollete



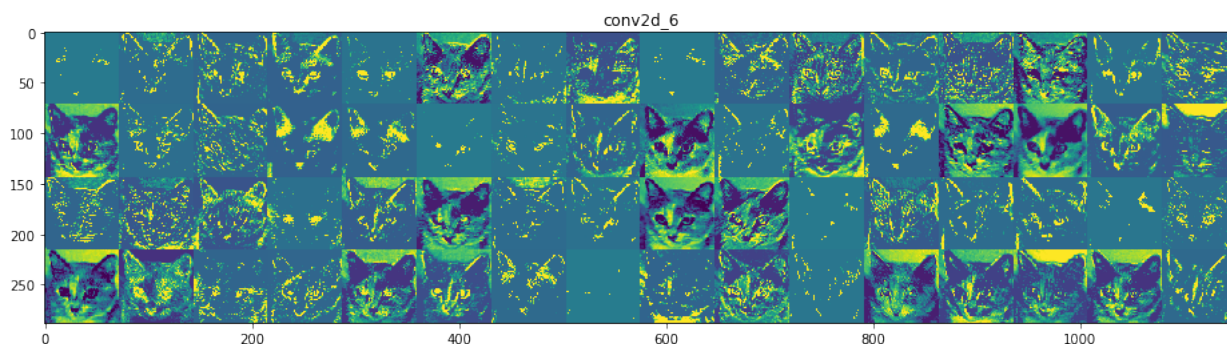
Después de cada convolución se suele aplicar un filtro que permite abstraer más la información y reducir la dimensionalidad de las imágenes. En este ejemplo se aplica un filtro *max pooling* a las convoluciones obtenidas y obtendríamos las imágenes de la **figura 2.20**.

Figura 2.20: Imágenes obtenidas al aplicar un filtro 'max pooling'.
Imagen de François Chollete



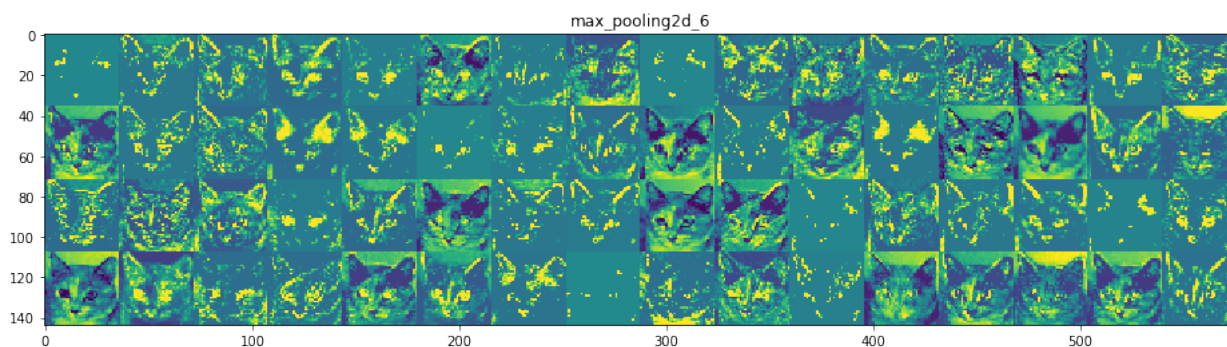
Por lo general se aplican varias capas de convoluciones y subsecuentes filtros que disminuyan la dimensionalidad de las imágenes para detectar patrones abstractos. En este ejemplo se aplican 3 capas de convoluciones y filtros *max pooling*. En la segunda capa se aplican 2 convoluciones nuevas, con lo que se dobla el número de imágenes, que se presentan en la **figura 2.21**,

Figura 2.21: Imágenes obtenidas tras 2 nuevas convoluciones.
Imagen de François Chollete



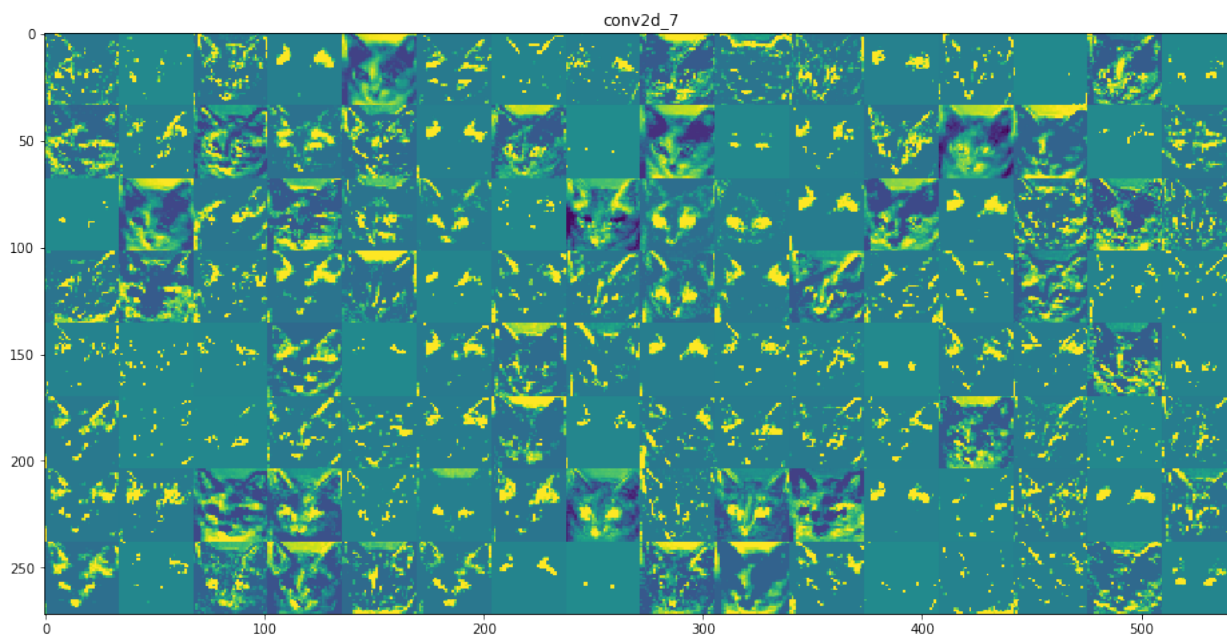
Nuevamente se aplica un nuevo filtro *max pooling* a los datos, como se muestra en la figura 2.22,

Figura 2.22: Imágenes obtenidas al aplicar un filtro 'max pooling'.
Imagen de François Chollete



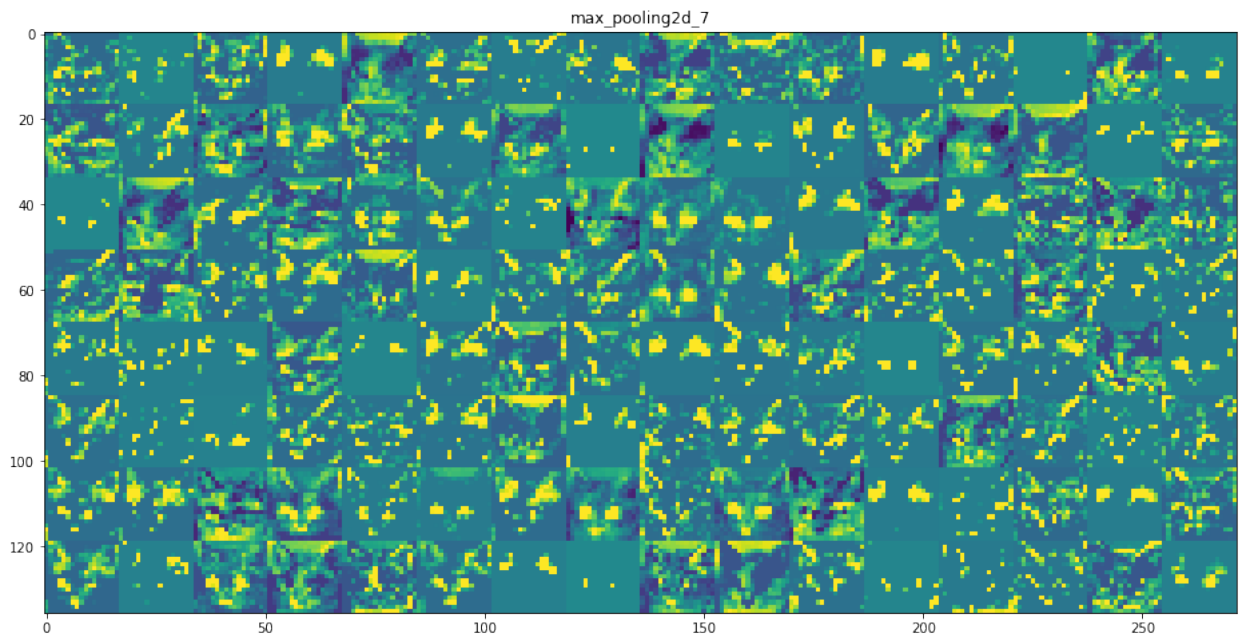
Por último se aplican dos nuevas convoluciones y en este punto podemos ver cómo los datos de las imágenes se empiezan a abstraer y cada vez dejan de parecer menos nuestra imagen de la que partimos, los resultados de la convolución se presentan en la **figura 2.23**,

Figura 2.23: Imágenes obtenidas tras 2 nuevas convoluciones.
Imagen de François Chollete



Por último aplicamos otro filtro *max pooling* y obtenemos las imágenes de la **figura 2.24**,

Figura 2.24: Imágenes obtenidas al aplicar un filtro 'max pooling'.
Imagen de François Chollete



Con este ejemplo podemos entender mejor como es que las convoluciones actúan para abstraer información de una matriz, que en este caso representa la imagen de un gato.

2.4. Software empleado

En este proyecto se programaron las redes neuronales mediante el uso de las siguientes herramientas

2.4.1. Python

Python es un lenguaje de programación de alto nivel, multiparadigma, esto último significa que Python soporta diferentes paradigmas de programación como lo son la programación orientada a objetos (usa metáforas de 'objetos' que interactúan entre sí), reflexiva (el programa puede observar y modificar sus estructuras de alto nivel, para ejecutar más rápido su código de bajo nivel), imperativa (se dan instrucciones claras de cómo resolver el problema) y funcional (se pueden declarar funciones). Python también es un lenguaje de código abierto; esto último siendo una de sus principales fortalezas ya que tiene una comunidad que aporta librerías y soluciones a diversos problemas.

Uno de los principales elementos en la filosofía de Python es KISS (*Keep it simple stupid*) por lo que se enfoca en ser lo más fácil de escribir y entender, lo que resulta ser una gran ventaja para científicos que quieren usar estas herramientas sin tener un conocimiento muy profundo respecto al funcionamiento de los lenguajes de programación o de las computadoras.

En este trabajo la simplicidad de Python permitió enfocar el trabajo en resolver el problema de clasificación de los GPCR.

2.4.2. TensorFlow

TensorFlow es una biblioteca de código abierto desarrollada por Google, inspirada en Theano, otra librería de Python optimizada para manipular y evaluar expresiones matemáticas, que ayuda en la diferenciación automática para aprendizaje automático al obtener las derivadas de los gradientes en los algoritmos de optimización. Esta librería usa grafos de flujo de datos computacionales para

representar redes neuronales complejas, los nodos en estos grafos representan operaciones y los nodos son los tensores de datos, siendo un tensor un arreglo de objetos de N dimensiones. Esta librería está basada en tensores y cuenta con capacidad de correr sus cálculos tanto en CPU (unidad de procesamiento central) como en GPU (unidad de procesamiento gráfico) [28].

2.4.3. Keras

Keras es una interfaz de programación de aplicaciones de alto nivel compacta, de composición de capas en redes neuronales profundas, con la cual se pueden implementar fácilmente distintas arquitecturas al crear modelos secuenciales definiendo las capas en la red neuronal y sus conexiones, aunando otras características de las redes neuronales, lo que facilita el diseño y prueba de prototipos, además se puede usar TensorFlow como motor para la diferenciación en las redes neuronales [21].

En un primer nivel usamos Python como lenguaje de programación para depurar los datos usados para entrenar la red neuronal, ya que los archivos obtenidos de las bases de datos que contenían la información de interés tenían mucha información que no era de interés y por lo tanto se tenía que aislar y presentar la información útil de una forma entendible para la red neuronal. La mayor parte de nuestra red neuronal se construyó mediante las herramientas de Keras con TensorFlow como motor para la diferenciación automática en los cálculos de los algoritmos de optimización.

2.5. Aceleración de redes neuronales mediante GPU

Antes de hablar acerca de cómo se puede acelerar el entrenamiento de una red neuronal mediante el uso de una GPU tenemos que definir que es una GPU. Una unidad de procesamiento gráfico, cuyas siglas en inglés serían GPU (*Graphics Processing Unit*), es un coprocesador que está optimizado para operar sobre matrices. Recordemos que las imágenes que vemos en una computadora en realidad son matrices que indican la intensidad de los colores RGB de un monitor.

Por la optimización que tienen para procesar matrices y tensores debido a su arquitectura, las GPUs son ampliamente usadas en inteligencia artificial para acelerar el entrenamiento de redes neuronales al grado de haber librerías construidas con esta intención como los son TensorFlow-gpu o CUDA de NVIDIA, una empresa de tarjetas gráficas.

Al usar este hardware podemos recortar el tiempo de entrenamiento de una red neuronal significativamente, con base en la potencia de la GPU, que por lo general pueden realizar de 10 a 15 operaciones por cada operación que un CPU, del inglés *Central Processing Unit* y que por defecto es donde se realizan las operaciones en una computadora, puede realizar [30].

Por lo general una red neuronal profunda puede llegar a los cientos o miles de millones de parámetros a entrenar, lo cual hace inviable realizar el entrenamiento de la red neuronal en un CPU y ahí es donde las GPU intervienen, ya que se puede usar más de una GPU al mismo tiempo para acelerar aun más los cálculos relacionados [31].

2.6. Bases de datos

2.6.1. GPCRdb

La base de datos de receptores acoplados a proteínas G cuyas siglas en inglés son GPCRdb es, como su nombre lo indica, una base de datos de los GPCR de código libre que se encuentra disponible en <http://www.gpcrdb.org/>. En esta base de datos se recopila la información estructural, mutaciones y datos relacionados con los que se pueden visualizar a los GPCR, además de tener diversas herramientas de análisis y visualización como lo son para ver los aminoácidos en cada hélice, **figura 2.25**, y la disposición de las hélices en la membrana celular vistas desde un plano horizontal, **figura 2.26**, siendo ambas imágenes obtenidas de dicha base de datos, y que tiene una interfaz de programación de aplicaciones compatible con Python, mediante la cual se pueden hacer solicitudes a la base de datos desde un programa de Python usando el protocolo de transferencia de hipertexto (HTTP). En este trabajo obtuvimos los datos de los ligandos de los receptores con esta herramienta, así como todos los nombres de las estructuras del grupo de GPCR tipo A con los que entrenamos a la red neuronal.

Figura 2.25: Aminoácidos en las diferentes hélices.
 Imagen de <https://gpcrd.org/family/001/>.

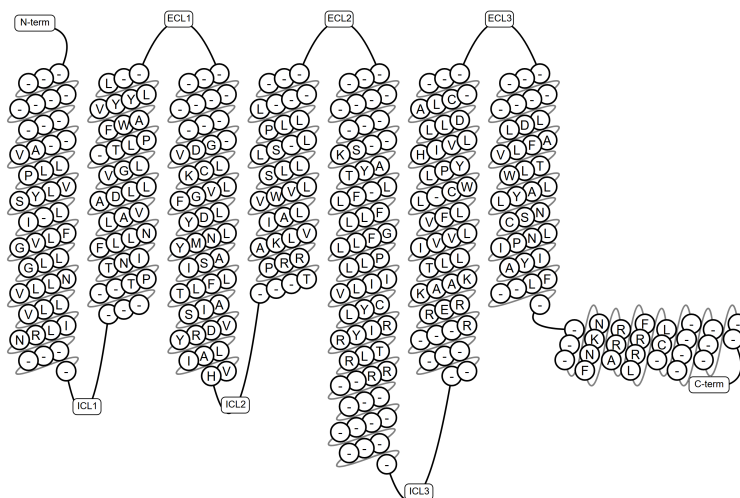
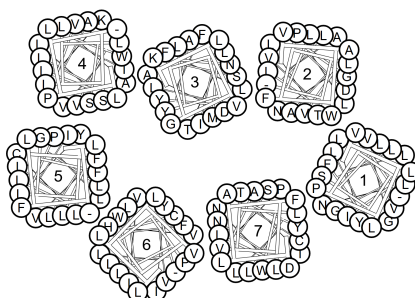


Figura 2.26: Estructuras de las hélices visto desde la cara extracelular de la membrana.
 Imagen de <https://gpcrd.org/family/001/>.



2.6.2. RCSB PDB

Esta base de datos disponible en <https://www.rcsb.org/> fue el primer sitio web de acceso libre en disponer digitalmente de datos biológicos y estructurales. En este portal podemos encontrar archivos de las estructuras tridimensionales de diversas proteínas, así como de una gran variedad de GPCR. Este portal también cuenta con una interfaz de programación de aplicaciones (cuyas siglas en inglés son API) compatible con Python.

2.7. Antecedentes de propuestas.

Al investigar en la literatura sólo se encontró un trabajo relacionado con el presente proyecto [1]. En dicho trabajo se estudiaron patrones en común de las estructuras tridimensionales de GPCR de la familia A, o familia de la Rodopsina, mediante algoritmos no supervisados de agrupamiento (también conocidos como *clustering*). Los resultados reportados en dicho trabajo logran una tasa de acierto del 72.1% para clasificar a los archivos de las proteínas entre los estados funcionales activo, inactivo e intermedio.

Capítulo 3

Metodología

Se dividió el proyecto en tres fases: 1) obtención de datos, 2) limpieza de los datos y 3) construcción y refinamiento de la red neuronal. Después de tener los datos de entrenamiento listos se realizó el prototipado de diferentes arquitecturas de redes neuronales, y por último se seleccionó a la arquitectura con la mejor tasa de acierto y con base en esta se realizó la variación de los hiperparámetros para mejorar su rendimiento y su tasa de acierto tanto en el grupo de entrenamiento como en el grupo de validación. Cada una de las fases del proyecto se describen en las siguientes secciones.

3.1. Obtención y limpieza de los datos

3.1.1. Obtención de los datos estructurales.

Con base en GPCRdb se obtuvo una lista de GPCR tipo A disponibles, de los que se tiene información estructural e información de sus ligandos asociados. Luego se descargó los archivos PDB de cada proteína de *RCSB PDB* mediante un código en Python programado para este proyecto.

Obtención de las etiquetas de los ligandos.

Después se desarrolló un programa en Python con el cual obtener la clasificación de los ligandos para las proteínas descargadas mediante el uso de la API de GPCRdb, que incluye información del tipo de ligando asociado a cada GPCR tipo A. Una vez que se tuvo en funcionamiento el programa, se aplicó en los archivos PDB obtenidos para los GPCR y con esto se completó la información de los GPCR con la que se trabajó en este proyecto.

Un diagrama general que muestra el procedimiento para construir la base de datos con la que se entrenó a las redes neuronales usadas se muestra en la **figura 3.1**

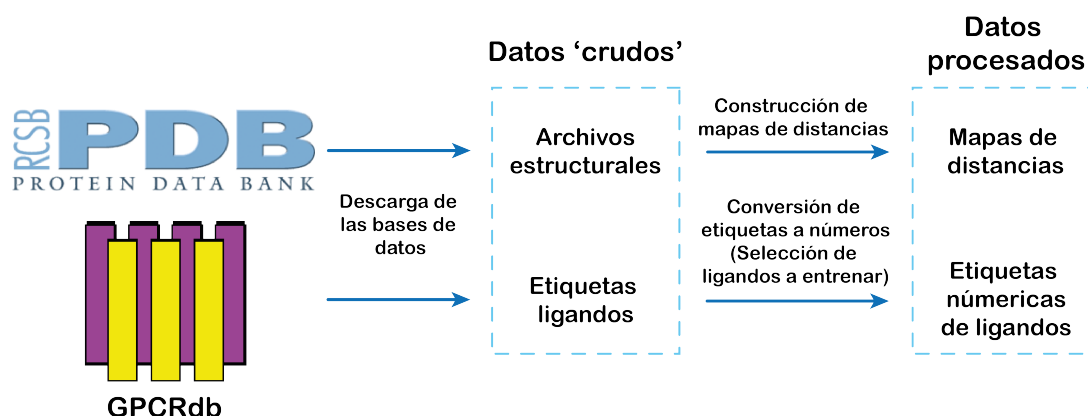


Figura 3.1: Obtención y limpieza de los datos usados.

Limpieza y preparación de datos para la red neuronal.

Después de obtener los datos, se tuvo que convertir la información seleccionada, representada en algunos casos por palabras como el caso de las etiquetas de sus ligandos, por información numérica que pudiera ser procesada por la red neuronal. Como se buscaba representar a los GPCR como imágenes, se optó por construir mapas de distancias usando los carbonos- α o β de la cadena principal de la proteína, cuyas coordenadas se obtuvieron de la información estructural provista en los archivos PDB de las proteínas. Se obtuvieron los datos de los aminoácidos correspondientes solamente a las hélices transmembranales y se hicieron cinco variantes para representar los mapas de distancias.

Para construir los mapas de distancias se tenía que seleccionar un punto de referencia que representara a cada aminoácido y con base en este, determinar las distancias entre los residuos contemplados para construir el mapa de distancias. Consultando la literatura se observó que una opción popular y sencilla era tomar el valor del carbono α , también se contempló probar con el carbono β de la cadena principal, **figura 2.5**.

Una vez que se determinó el centro del cual se obtendrían las distancias para cada aminoácido se prepararon los datos y se procesaron para obtener primero las matrices de distancias y con base en estas matrices construir los mapas de distancias en formato de imagen.

El algoritmo para construir los mapas de distancias fue el siguiente: 1) Se determinó que átomo sería considerado como centro de la proteína, 2) Se obtuvieron las coordenadas de todos los centros del paso previo para la proteína, dependiendo del modelo construido se usaron todos los aminoácidos o solo los de las 7 hélices transmembranales, 3) Se midió la distancia de cada aminoácido con respecto a los demás aminoácidos de la proteína, 4) Se construyó el mapa de distancias.

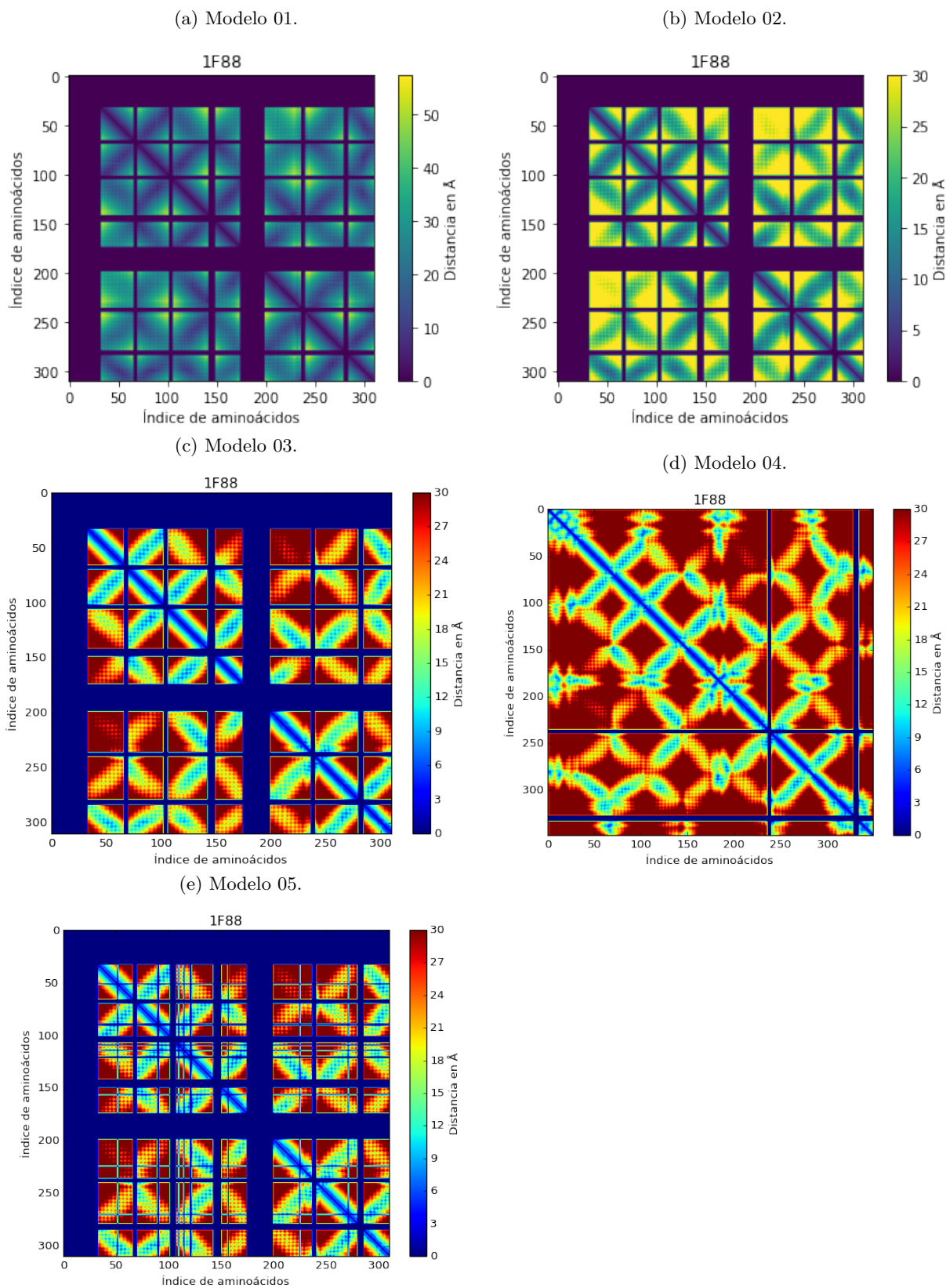
Se obtuvieron cinco diferentes modelos, o variantes, de mapas de distancias. Para el modelo 01 representado en la **figura 3.2a** se usaron los carbonos- α y sólo los residuos de las siete hélices transmembranales, aparte de que no se definió una distancia máxima, por lo que el color amarillo, que corresponde con el valor máximo en el mapa de distancias, podía representar diferentes distancias para diferentes proteínas. Luego se optó por poner un valor máximo para las distancias ya que en el modelo 01 el valor máximo variaba y eso imponía el color amarillo en las distancias máximas, por lo que las imágenes no estaban en la misma escala de colores. Con esa corrección en el modelo 02, representado en la **figura 3.2b** si dos píxeles en distintos mapas de distancias tienen el mismo color, entonces representan la misma distancia. Por último el modelo 03, **figura 3.2c** agrega más colores para enriquecer la información en las imágenes generadas. Al contar con más colores en las imágenes, se esperaba que el desempeño de la red neuronal fuera mejor.

Los modelos 04 y 05 son variantes del modelo 03. En el caso del modelo 04, **figura 3.2b** se consideraron todos los aminoácidos en la cadena de la proteína y no sólo los aminoácidos de las siete hélices transmembranales. Por último para el modelo 05 se reemplazaron los carbonos- α por los carbonos- β de los residuos en las 7 hélices transmembranales, **figura 3.2e**.

Los espacios al principio y en algunas partes de las imágenes correspondientes con una distancia cero se deben a la ausencia de información estructural en los diferentes archivos PDB que se usaron, a excepción de la diagonal central que corresponde a la distancia entre un aminoácido y el mismo aminoácido, por lo que se esperaba una distancia de cero angstroms.

De la base de datos usada (GPCRdb) sólo se obtuvieron las siguientes clasificaciones para los GPCR clase A: ['Agonista inverso', 'Agonista', 'Nulo', 'Antagonista', 'Agonista (parcial)', 'Agonista inverso (parcial)', 'PAM', 'NAM'], donde la categoría de nulo corresponde con GPCR que no cuentan con ligando o no se encuentra reportado y como carecían de esta información se descartaron esos archivos. Al hacer un recuento de la ocurrencia para cada uno de estos casos, a excepción de la categoría Nulo, en los datos de entrenamiento se obtuvo el **cuadro 3.1**.

Figura 3.2: Matrices de distancias generadas con los modelos, a) Modelo 01, b) Modelo 02, c) Modelo 03, d) Modelo 04 y e) Modelo 05.

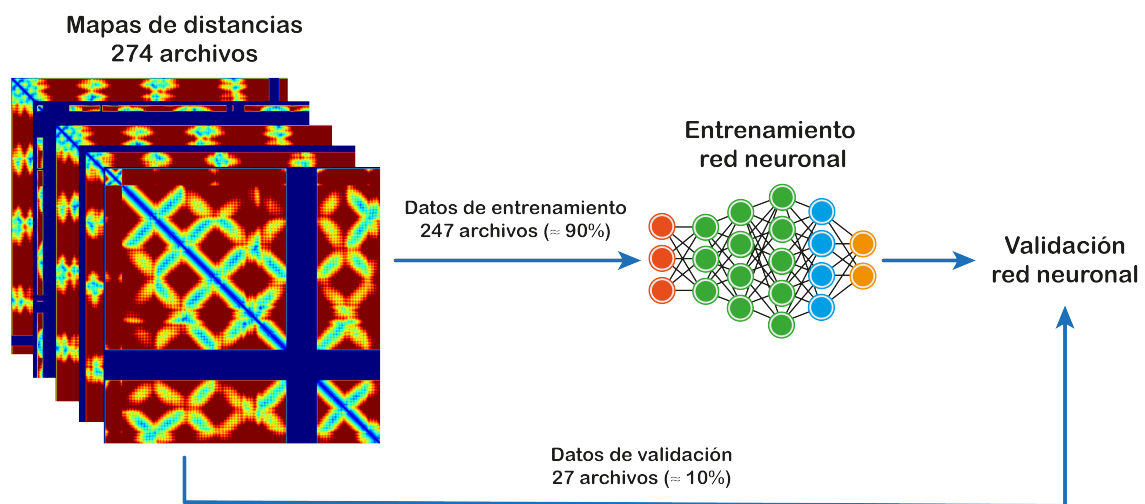


Cuadro 3.1: Incidencia de los ligandos en los GPCR a trabajar.

| Estado | Casos |
|----------------------------|-------|
| Agonista | 108 |
| Agonista (parcial) | 5 |
| Antagonista | 136 |
| Agonista inverso | 30 |
| Agonista inverso (parcial) | 1 |
| NAM | 1 |
| PAM | 2 |

Debido a que las clasificaciones más comunes fueron 'agonista', 'agonista inverso' y 'antagonista' se optó por trabajar sólo con esos tres grupos. Una vez que se tuvo la clasificación de los ligandos se prepararon los datos para alimentar las diferentes arquitecturas de redes neuronales que se probaron. Esa preparación consistió en separar a los 274 mapas de distancias de cada modelo en 2 categorías: un grupo de entrenamiento y uno de validación, como se muestra en la **figura 3.3**.

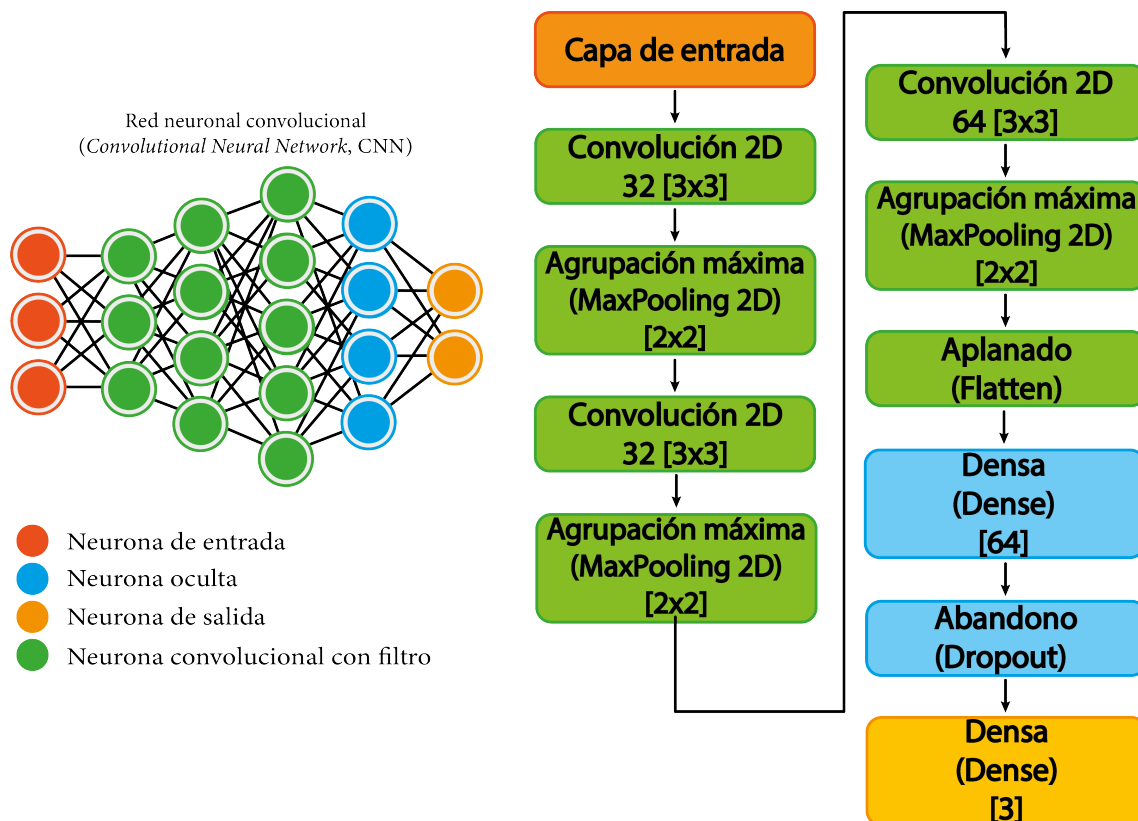
Figura 3.3: Manejo de archivos.



3.2. Prototipado de diferentes arquitecturas

Aunque se realizaron diversas pruebas de arquitecturas en esta parte del trabajo, todas eran redes neuronales convolucionales. Debido a la buena tasa de clasificación y la rapidez con la que se entrenaban las épocas en las pruebas preliminares, se optó por trabajar con modelos que tuvieran pocas capas de convoluciones y pocas capas ocultas de clasificación y el modelo que mejores resultados generó fue el presentado en la Figura 3.4. Los nombres de las capas describen qué procedimiento se aplicó, dónde: *Capa de entrada* es la primer capa de la red neuronal, aquella con la que se alimenta a la red neuronal con los datos de entrenamiento, *Convolución 2D* denota una capa convolucional 2D, *MaxPooling 2D* corresponde a un filtro convolucional MaxPooling en 2D, *Flatten* es la capa que convierte en vectores a las matrices, *Dense* es una capa oculta de neuronas y *Dropout* es una capa de regularización que apaga aleatoriamente neuronas en el entrenamiento para evitar una sobreestimación por parte de la red neuronal.

Figura 3.4: Arquitectura de la red neuronal.



3.2.1. Variación de los hiperparámetros

Una vez que se encontró una arquitectura con una buena capacidad de predicción, que en este trabajo fue definida como una tasa de acierto superior al 72.1% con base en lo reportado por [1], se inició la variación de hiperparámetros de la red neuronal para intentar mejorar el resultado de clasificación en el modelo construido. Los hiperparámetros que se variaron fueron:

- **Algoritmo de optimización.**
- **Número de neuronas.** Este parámetro se varió en múltiplos de 64 neuronas, las potencias de 2 ($2^6 = 64$) se usan ampliamente en la literatura aunque no es muy claro el motivo, algunos justifican que ayuda a manejar la información en los espacios de memoria en la computadora y de esta forma acelerar los cálculos [32], pero se usa ampliamente como una regla heurística. La última capa siempre tuvo 3 neuronas que corresponden a las 3 clasificaciones de los ligandos usadas en este trabajo. En un principio se procuró tener un modelo que empezaba con pocas neuronas y en cada capa aumentaba el número de neuronas hasta llegar al máximo y después disminuía hasta llegar a la última capa, la capa de salida. Lo anterior se pensó en conformidad con las arquitecturas tradicionales presentadas en la literatura [33]. Aunque después de las pruebas preliminares se observó que bastaba con una red neuronal que después de las convoluciones sólo requería una capa de 64 neuronas antes de la capa de salida.
- **Número de capas.** Originalmente se probó un modelo con tres capas, sin contar la capa 0, que es la capa de entrada de los datos; este parámetro se varió hasta llegar a una arquitectura de 2 capas de neuronas y 3 capas de convoluciones, sin contar a la capa 0.
- **Función de activación.** La función de activación que dio mejores resultados en las pruebas preliminares fue *ReLU*, véase **sección 2.3.3**, para las capas ocultas y las de las convoluciones, mientras que se escogió una activación *softmax* para la última capa, la encargada de la clasificación. Esta función regresa una probabilidad decimal para cada tipo de clasificación, donde la suma de estas probabilidades da 1.
- **Tamaño de los grupos de entrenamiento y evaluación.** Dado que se cuenta con pocos datos para el entrenamiento se terminó optando por usar el 90% de los datos de los que

disponíamos como datos de entrenamiento, el 10 % restante se usó como grupo de datos de evaluación.

- **Representación de los datos.** Con la intención de probar como afectaba a la red neuronal los datos con los que se le alimentaba se probaron 5 representaciones diferentes de los datos, estos modelos ya fueron descritos previamente y corresponden a las 5 variantes de imágenes con las que se alimentó a la red neuronal.

3.2.2. Aceleración de entrenamiento mediante GPU

Se aprovecho que se contaba con una tarjeta gráfica para ejecutar los cálculos, se trabajaron los modelos 04 y 05 con aceleración por GPU mediante TensorFlow.

En el caso del modelo 04 las imágenes procesadas eran demasiado grandes en terminos de memoria consumida y al momento de generar la red neuronal para estas imágenes con aceleración por GPU se consumía toda la memoria RAM (*Random Access Memory*) de la tarjeta gráfica y se interrumpía el entrenamiento. Por lo que se tuvieron que recortar los tamaños de las imágenes y se implementó una función que limpiaba la memoria RAM de la GPU tras cada entrenamiento, aparte de que se redujeron las pruebas a sólo 5. A este modelo lo denominamos **modelo 04***. Por la misma razón sólo se realizaron 6 pruebas en el modelo 05 GPU, al cual denominamos como **modelo 05***.

Para las pruebas con GPU la arquitectura y los modelos de entrenamiento fueron los mismos, salvo las excepciones mencionadas previamente en el manejo del tamaño de los mapas de distancias. Lo único diferente fue que los cálculos se ejecutaron en una unidad GPU en lugar de en una unidad CPU.

En este trabajo los entrenamientos se realizaron aproximadamente 10 veces más rápido disminuyendo las horas usadas en probar diferentes variantes de las arquitecturas en las redes neuronales prototipadas.

Capítulo 4

Resultados, discusión de resultados, conclusiones y perspectivas

Una vez que se procesaron los diferentes modelos de datos por la arquitectura de red neuronal seleccionada, correspondía comparar los valores obtenidos para poder determinar las diferencias entre los diferentes modelos usados y corroborar si se habían cumplido los objetivos propuestos en este trabajo.

Debido a que en algunos casos la red neuronal presentaba un sobreajuste por arriba de la época 20, se aplicó una *'callback'*, una función o procedimiento programado en Python que permite interrumpir el entrenamiento de la red neuronal en caso de satisfacer una condición, cuya tarea era parar el entrenamiento en caso de que al final de una época se lograra alcanzar una tasa de acierto del 91 % o mayor, sobre el conjunto de validación. Este valor se escogió con base en los resultados preliminares que se pudieron observar en las pruebas previas de diferentes arquitecturas para la red neuronal, ya que fue el mejor valor obtenido en dichas pruebas. En caso de no cumplir con dicha tasa de acierto, el entrenamiento se paraba tras 25 épocas. Se realizaron 7 pruebas, donde no se realizó ningún cambio y se observó si los resultados presentaban repetibilidad para cada modelo de los datos de entrenamiento usando la red neuronal. Se realizaron 7 pruebas por criterios estadísticos. De los resultados obtenidos para cada uno de los 5 modelos los datos se presentan en las siguientes secciones.

El criterio principal sobre el que se evaluó el desempeño de las redes neuronales en este trabajo fue la tasa de acierto en el conjunto de validación, ya que es un indicador de la capacidad que tiene la red neuronal para predecir acertadamente la clasificación de una imagen nueva que no fue usada para el entrenamiento de la red neuronal y que la red neuronal no conoce.

Las tablas donde se compilan los resultados obtenidos en los diferentes entrenamientos se encuentran en el capítulo 5, correspondiente con el Anexo I.

4.1. Resultados modelo 01

Este modelo se construyó al medir la distancia entre los carbonos- α de solamente las siete hélices transmembranales, con un abanico de colores entre azul y amarillo, aparte de que no se fijó una distancia máxima.

Los resultados para el modelo 01 se presentan en las **gráficas 4.1a, 4.1b, 4.2a y 4.2b**, así como en las **tablas 5.1-5.5**. Podemos ver que para el grupo de entrenamiento los mejores resultados se concentran en las últimas épocas, mientras que para el grupo de validación los mejores resultados se encuentran muy dispersos. Por ejemplo, en la época 9 de la sexta prueba se obtiene la tasa de acierto máxima para el grupo de validación con un valor de 85.71 % y uno de 78.46 %, mientras que para el experimento 4 se obtiene hasta la época 22 con un valor de 85.71 % en el grupo de validación y 91.06 % sobre el grupo de entrenamiento.

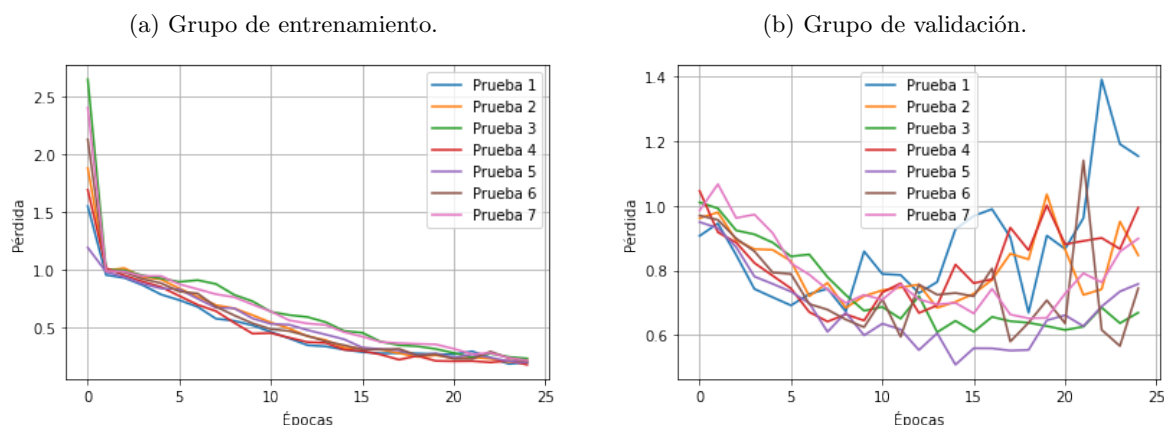
También podemos ver que los resultados para el grupo de validación tienen un comportamiento dispar y esto se debe a que el número de archivos usados en el grupo de validación es significativamente menor (27 archivos) que los usados en el grupo de entrenamiento (247 archivos).

Por otro lado, al analizar las gráficas de la función de pérdida podemos ver que la que corresponde al grupo de entrenamiento, **gráfica 4.1a**, se aproxima a cero conforme las épocas aumentan, lo

cual es un comportamiento esperado. Lo normal es que conforme aumentan las épocas la capacidad de predicción de la red neuronal mejora y como resultado la función de pérdida se aproxima a cero. Al analizar el mismo parámetro para el grupo de validación, **gráfica 4.1b**, podemos ver un comportamiento más caótico, en especial para la prueba 1, aunque en general todas las pruebas parecen tener una especie de punto de inflexión entre las épocas 10 a 15.

Otro detalle a destacar es que los valores evaluados para los diferentes modelos pueden ser muy distantes al principio de un entrenamiento. Esto se debe a que un entrenamiento inicializa los parámetros de la red neuronal de forma aleatoria, pero por la representación de los datos y la arquitectura usada en esos valores vana converger y se aproximaran, independientemente de pertenecer a experimentos diferentes.

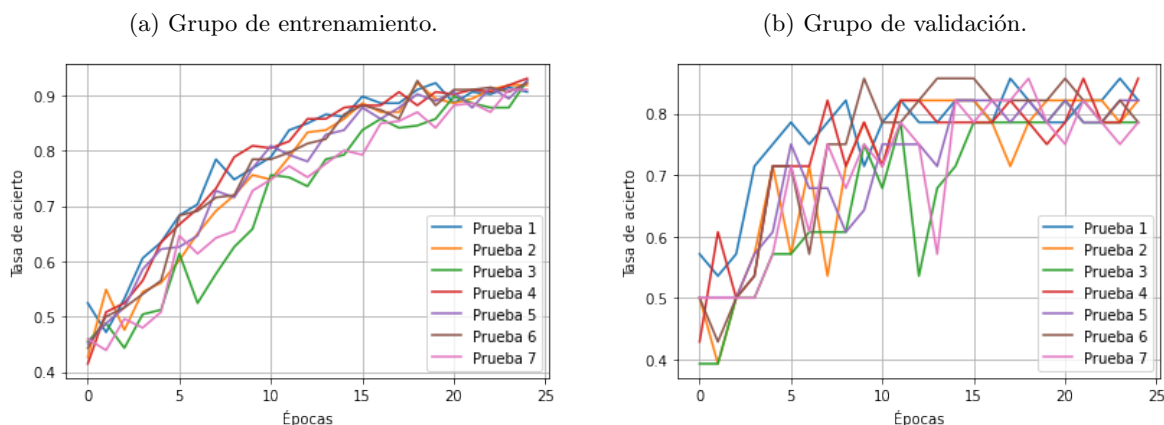
Figura 4.1: Pérdidas por época obtenidas para modelo 01 en, a) el grupo de entrenamiento y b) el grupo de validación.



Si analizamos las gráficas correspondientes a la tasa de acierto podemos ver que la correspondiente al grupo de entrenamiento, **gráfica 4.2a**, mejora su capacidad de predicción conforme aumentan las épocas de entrenamiento para todas las pruebas y viendo la **tabla 5.1** podemos ver que los valores máximos se encuentran un poco por encima del 91% e inician a aparecer tras la época 18.

Usando el mismo criterio podemos ver que los resultados para el grupo de validación, representado en la **gráfica 4.2b**, siguen un comportamiento similar. En este caso se puede ver un comportamiento con cambios abruptos en las líneas graficadas. Esto lo atribuimos al número limitado de datos con los que se disponía para evaluar las redes neuronales.

Figura 4.2: Tasa de acierto por época obtenidas para modelo 01 en, a) el grupo de entrenamiento y b) el grupo de validación.



El mejor resultado obtenido fue en la época 22 en la prueba 4, donde los resultados fueron; 85.71 % de acierto sobre el grupo de validación, 91.06 % de acierto sobre el grupo de entrenamiento, 0.8915 de pérdida en el grupo de validación y 0.2034 de pérdida sobre el grupo de entrenamiento.

Por último, los tiempos de entrenamiento, que se encuentran en la **tabla 5.5** oscilan en 5.42 segundos de cómputo por época, excepto para las últimas dos pruebas que toman casi el doble de

tiempo.

4.2. Resultados modelo 02

Este modelo se construyo midiendo la distancia entre los carbonos- α de solamente las siete hélices transmembranales, con un abanico de colores entre azul y amarillo, aparte de que se fijó una distancia máxima de 30Å.

Los resultados para el modelo 02 se presentan en las **gráficas (4.3a, 4.3b, 4.4a y 4.4b, así como en las tablas 5.6-5.10.** Respecto al comportamiento de las gráficas es bastante similar al que mostraban los entrenamientos para el modelo 01, a excepción de que la tasa de acierto en el grupo de validación llega a valores máximos en un menor número de épocas. Los valores obtenidos para la tasa de acierto son mayores, y se alcanzo el 89.29% de acierto, en comparación al 85.71 % del modelo 01 y todos las pruebas parecen oscilar sobre dicho valor hacia las ultimas épocas de la red neuronal.

Figura 4.3: Pérdidas por época obtenidas para modelo 02 en, a) el grupo de entrenamiento y b) el grupo de validación.

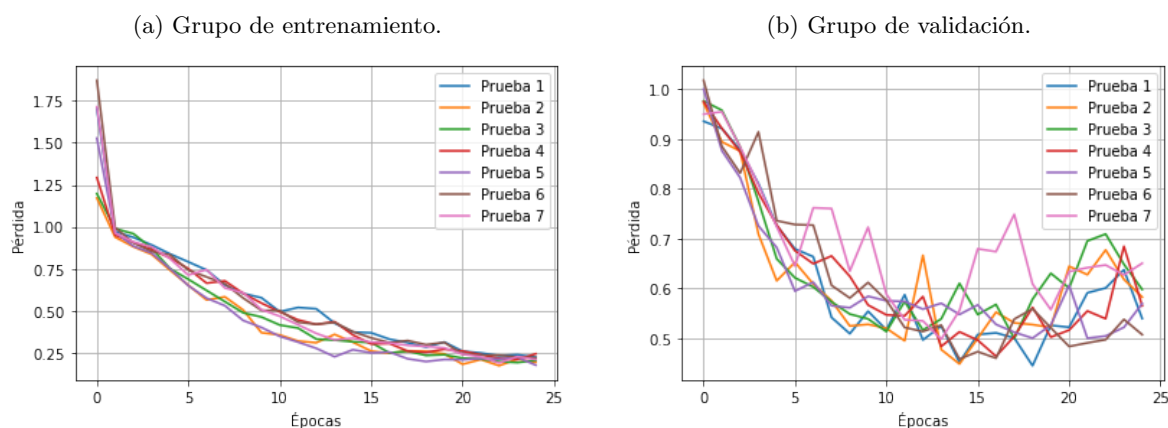
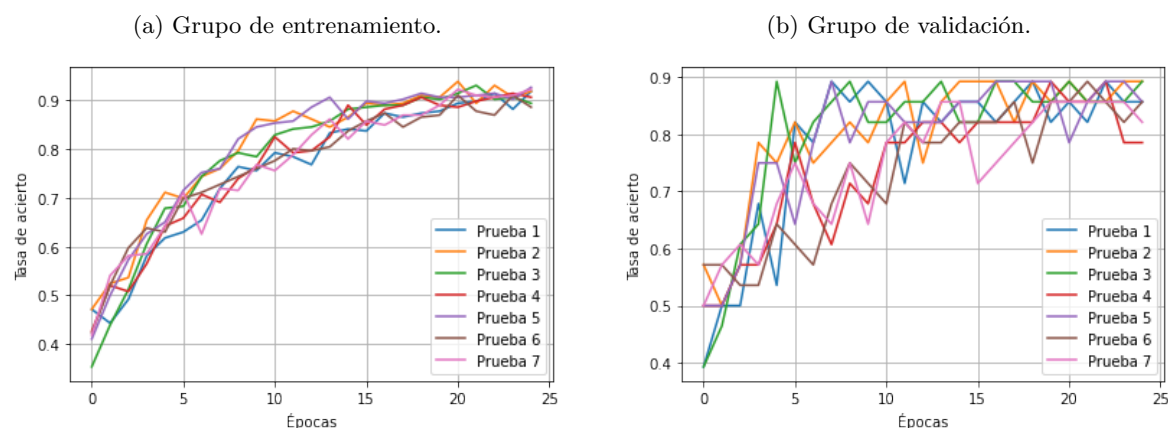


Figura 4.4: Tasa de acierto por época obtenidas para modelo 02 en, a) el grupo de entrenamiento y b) el grupo de validación.



El mejor resultado obtenido fue en la época 20 en la prueba 2, donde los resultados fueron; 89.29% de acierto sobre el grupo de validación, 89.02% de acierto sobre el grupo de entrenamiento, 0.5022 de pérdida en el grupo de validación y 0.2700 de pérdida sobre el grupo de entrenamiento. Con lo que podemos ver que todos los parámetros mejoran con respecto al mejor resultado del modelo 01.

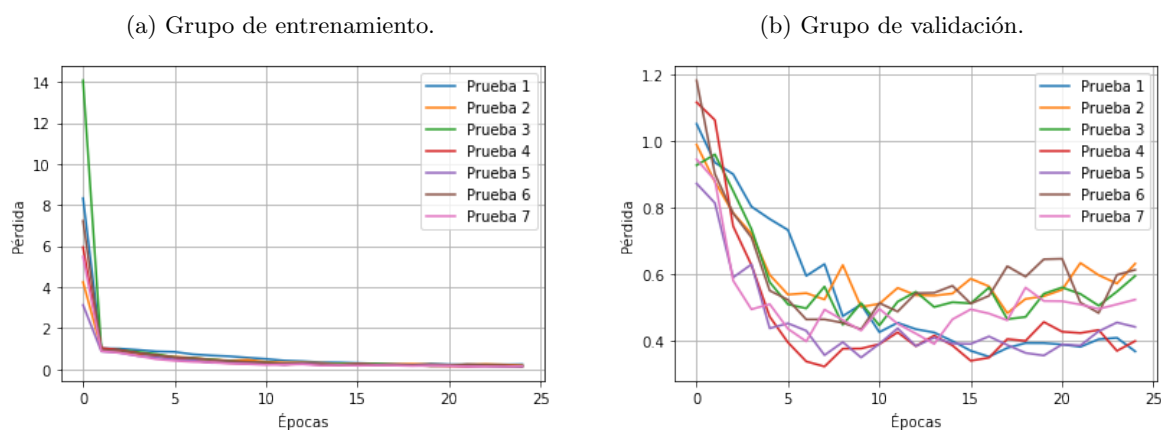
Al analizar los tiempos de entrenamiento, **tabla 5.10,** podemos ver que estos oscilan en 6 segundos por época.

4.3. Resultados modelo 03

Este modelo se construyó midiendo la distancia entre los carbonos- α de solamente las siete hélices transmembranales, con un abanico de colores correspondientes con los presentes en un arcoíris, aparte de que se fijó una distancia máxima de 30Å.

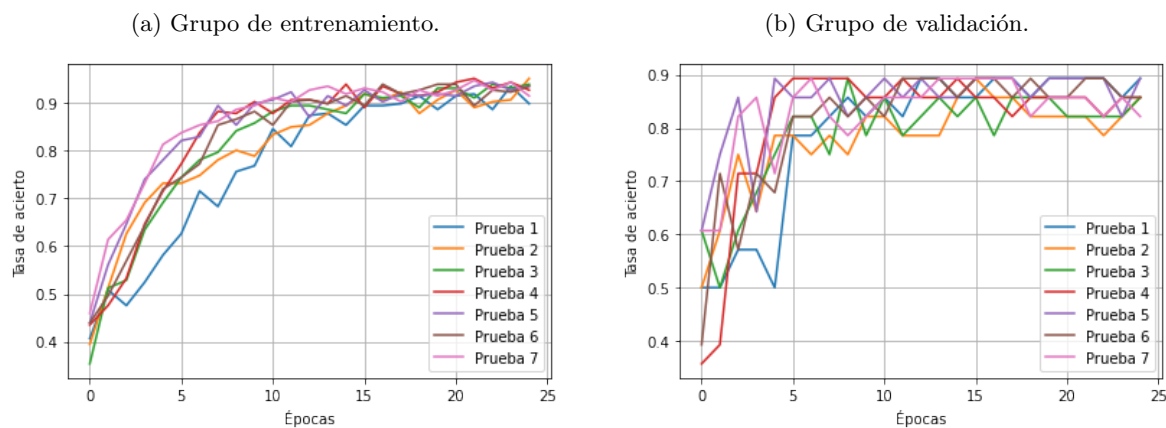
Los resultados para el modelo 03 se presentan en las **gráficas 4.5a, 4.5b, 4.6a y 4.6b**, así como en las **tablas 5.11-5.15**. En este modelo podemos notar un gran cambio en los comportamientos de los resultados con respecto a los dos anteriores. Primero podemos observar que en el grupo de entrenamiento, **gráfica 4.5a**, la función de pérdida alcanza valores más pequeños que los obtenidos en los modelos 01 y 02. Aunque en general todos los valores empiezan con valores relativamente grandes. Para el grupo de validación el comportamiento y valores obtenidos de la función de pérdida, **gráfica 4.5b**, son muy similares a los del modelo 02, aunque podemos notar que para las diferentes pruebas el comportamiento tiene menor variabilidad entre pruebas que en el caso del modelo 02, **gráfica 4.3b**

Figura 4.5: Pérdidas por época obtenidas para modelo 03 en, a) el grupo de entrenamiento y b) el grupo de validación.



Al analizar el comportamiento de la tasa de acierto, podemos ver que el rendimiento es mejor que la del modelo 01 para ambos grupos (el de entrenamiento y el de validación). También podemos ver que es ligeramente mejor que el desempeño del modelo 02, ya que alcanza más rápido los valores máximos y consistentemente se mantiene en estos, siendo 94.31 % para el grupo de entrenamiento y del 89.29 % para el grupo de validación.

Figura 4.6: Tasa de acierto por época obtenidas para modelo 03 en, a) el grupo de entrenamiento y b) el grupo de validación.



El mejor resultado obtenido fue en la época 23 en la prueba 5, donde los resultados fueron; 89.29 % de acierto sobre el grupo de validación, 94.31 % de acierto sobre el grupo de entrenamiento, 0.4294

de pérdida en el grupo de validación y 0.1417 de pérdida sobre el grupo de entrenamiento. Con lo que podemos ver que todos los parámetros, son similares a los obtenidos para el modelo 02.

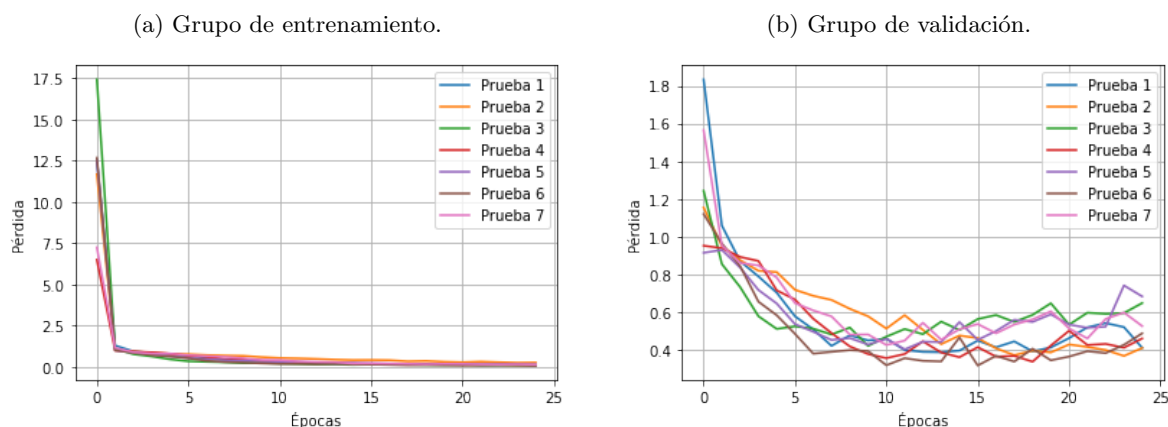
Al analizar los tiempos de entrenamiento, **tabla 5.15**, podemos ver que estos oscilan en 30 segundos por época, excepto para la prueba 2, donde se tardó en promedio 30 segundos por época.

4.4. Resultados modelo 04

Este modelo se construyó con base en el modelo 03, siendo el único cambio usar los carbonos- α de todos los aminoácidos presentes en la cadena principal de la proteína y no sólo los de las hélices transmembranales.

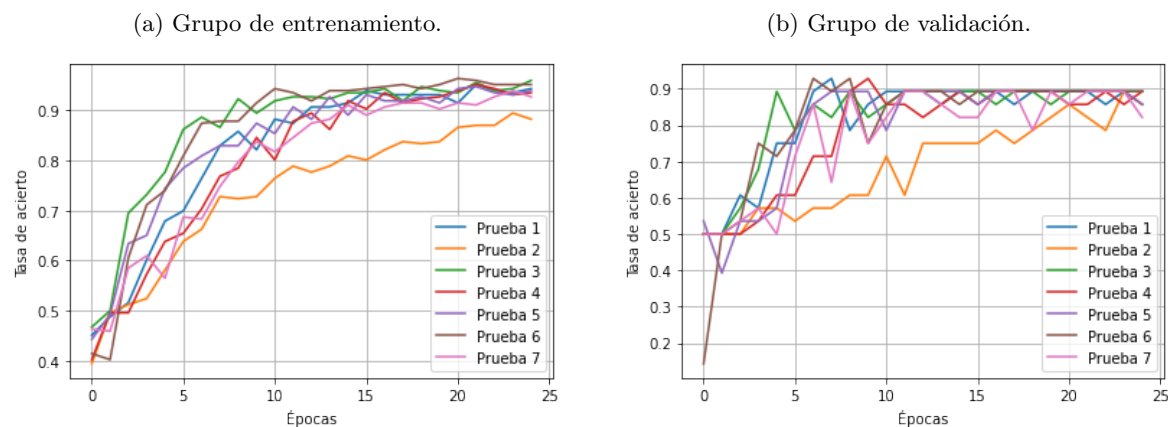
Los resultados para el modelo 04 se presentan en las **gráficas 4.7a, 4.7b, 4.8a y 4.8b**, así como en las **tablas 5.16-5.20**. El comportamiento de este modelo es muy similar al presentado en el modelo 03, en todos los parámetros analizados.

Figura 4.7: Pérdidas por época obtenidas para modelo 04 en, a) el grupo de entrenamiento y b) el grupo de validación.



Los elementos más relevantes son un comportamiento anómalo con respecto al de las otras pruebas de la tasa de acierto de la prueba 2, **gráficas 4.8a y 4.8b**, y que en este modelo se obtienen valores máximos del 92.86% de la tasa de acierto sobre el grupo de validación en las épocas 5 a 10 en diferentes pruebas, **gráfica 4.8b y tabla 5.17**.

Figura 4.8: Tasa de acierto por época obtenidas para modelo 04 en, a) el grupo de entrenamiento y b) el grupo de validación.



El mejor resultado obtenido fue en la época 7 en la prueba 6, donde los resultados fueron; 92.86% de acierto sobre el grupo de validación, 87.40% de acierto sobre el grupo de entrenamiento, 0.3794 de pérdida en el grupo de validación y 0.3815 de pérdida sobre el grupo de entrenamiento. En este

caso particular los parámetros relacionados con el grupo de validación mejoraron con respecto a los modelos anteriores, mientras que los parámetros asociados con el grupo de entrenamiento empeoraron, aunque siguen teniendo valores que pueden considerarse aceptables ya que los aciertos son mayores al criterio establecido de 72.1% y las pérdidas son relativamente cercanas a cero.

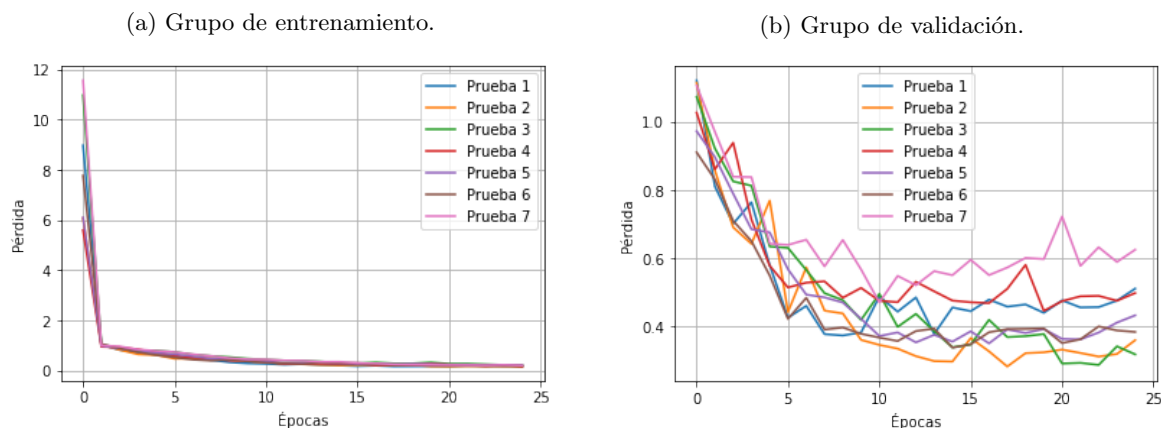
Al analizar los tiempos de entrenamiento, **tabla 5.20**, podemos ver que estos oscilan en 40 segundos por época. Esto se debe al aumento en la complejidad de los datos a trabajar, recordemos que los modelos 03, 04 y 05 amplían la gama de colores usados en los mapas de distancias, lo que resulto en un aumento en el tiempo de computo de dichas representaciones.

4.5. Resultados modelo 05

Este modelo se construyo con base en el modelo 03, siendo el único cambio usar los carbono- β de solamente los aminoácidos presentes en las hélices transmembranales.

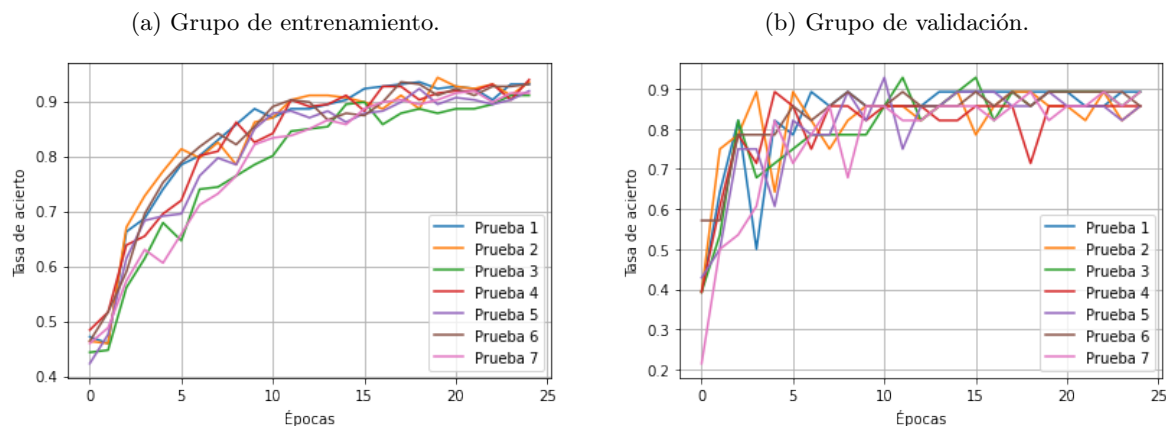
Los resultados para el modelo 05 se presentan en las **gráficas 4.9a, 4.9b, 4.10a y 4.10b**, así como en las **tablas 5.21-5.25**. El comportamiento de este modelo es muy similar al presentado en los modelos 03 y 04, en todos los parámetros analizados.

Figura 4.9: Pérdidas por época obtenidas para modelo 05 en, a) el grupo de entrenamiento y b) el grupo de validación.



Podemos ver en la **gráfica 4.10a** que el modelo tarda aproximadamente 10 épocas en alcanzar sus valores máximos sobre la tasa de acierto en el grupo de entrenamiento.

Figura 4.10: Tasa de acierto por época obtenidas para modelo 04 en, a) el grupo de entrenamiento y b) el grupo de validación.



El mejor resultado obtenido fue en la época 11 en la prueba 5, donde los resultados fueron; 92.86% de acierto sobre el grupo de validación, 87.80% de acierto sobre el grupo de entrenamiento, 0.3714 de

pérdida en el grupo de validación y 0.3550 de pérdida sobre el grupo de entrenamiento. Los resultados obtenidos en este modelo son muy similares a los obtenidos en el modelo 04, donde la tasa de acierto del grupo de entrenamiento y la pérdida del grupo de validación mejoran ligeramente con respecto al modelo 04 y la pérdida del grupo de validación empeora ligeramente.

La principal razón a la que atribuimos que la tasa de acierto sobre el grupo de validación sea mayor en el grupo de entrenamiento, es la muestra tan pequeña que representa el grupo de validación. Recordemos que sólo se trabajaron con 274 casos, de los cuales 27 se usaron para evaluar el modelo.

Por otra parte la forma en que se construyeron los mapas de distancias en el modelo 05 es poco práctico, ya que necesitamos definir la numeración de Ballesteros para poder construir los mapas de distancias. Este proceso no es difícil de realizar, pero puede entorpecer el procesamiento de datos nuevos, en especial en los casos en que el residuo central de alguna hélice no corresponde con el más conservado, véase **sección 2.1.5**

Al analizar los tiempos de entrenamiento, **tabla 5.25**, podemos ver que estos oscilan en 30.5 segundos por época.

4.6. Pruebas con GPU.

4.6.1. Resultados modelo 04* GPU

Los resultados para el modelo 05 se presentan en las **gráficas 4.11a, 4.11b, 4.12a y 4.12b**, así como en las **tablas 5.26-5.30**.

Los resultados obtenidos en el modelo 04* GPU son bastante similares a los obtenidos en el modelo 04, a pesar de los cambios aplicados al volver a dimensionar las imágenes usadas en el entrenamiento. La única mejora considerable es la drástica disminución de los tiempos de entrenamiento que oscilan por 2.5 segundos por época.

Figura 4.11: Pérdidas por época obtenidas para modelo 04* GPU en, a) el grupo de entrenamiento y b) el grupo de validación.

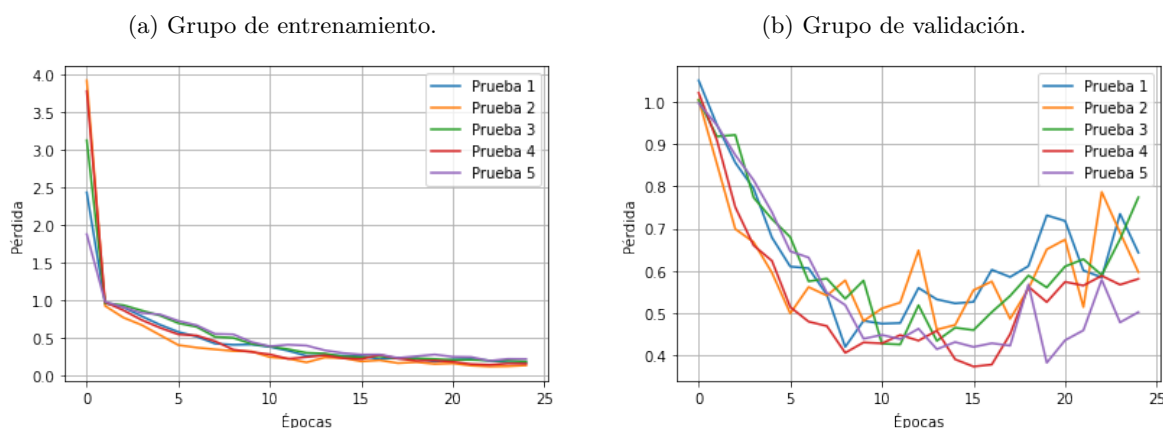
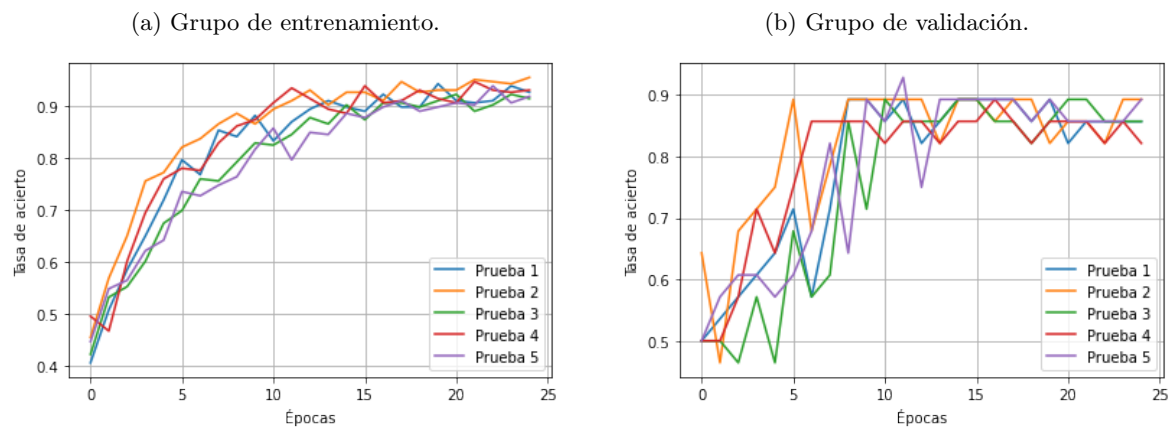


Figura 4.12: Tasa de acierto por época obtenidas para modelo 04* GPU en, a) el grupo de entrenamiento y b) el grupo de validación.



4.6.2. Resultados modelo 05 GPU

Los resultados para el modelo 05 se presentan en las **gráficas 4.13a, 4.13b, 4.14a y 4.14b**, así como en las **tablas 5.31-5.35**.

Nuevamente los resultados obtenidos por el modelo 05 GPU son bastante similares a los obtenidos en el modelo 05. La principal mejora son los tiempos de entrenamiento que disminuyen hasta los casi 1.5 segundos por época.

Figura 4.13: Pérdidas por época obtenidas para modelo 05 GPU en, a) el grupo de entrenamiento y b) el grupo de validación.

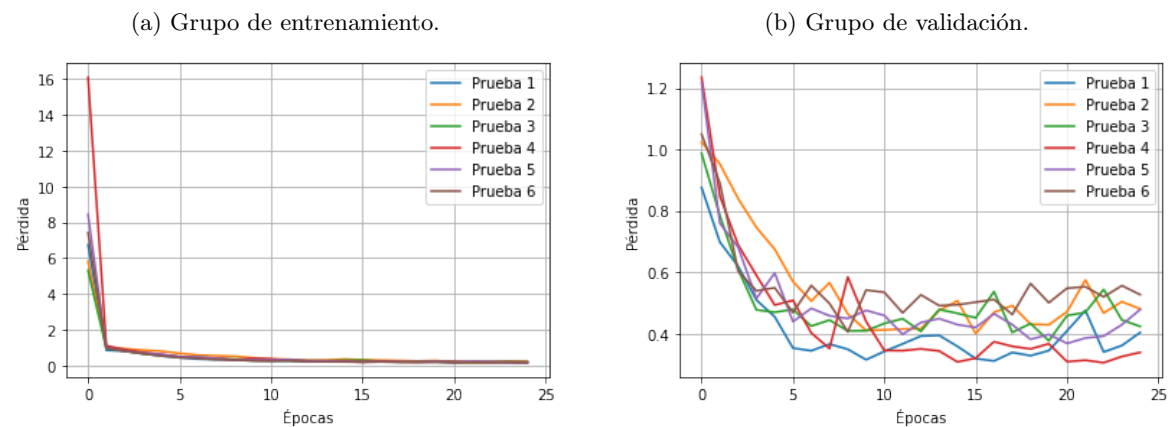
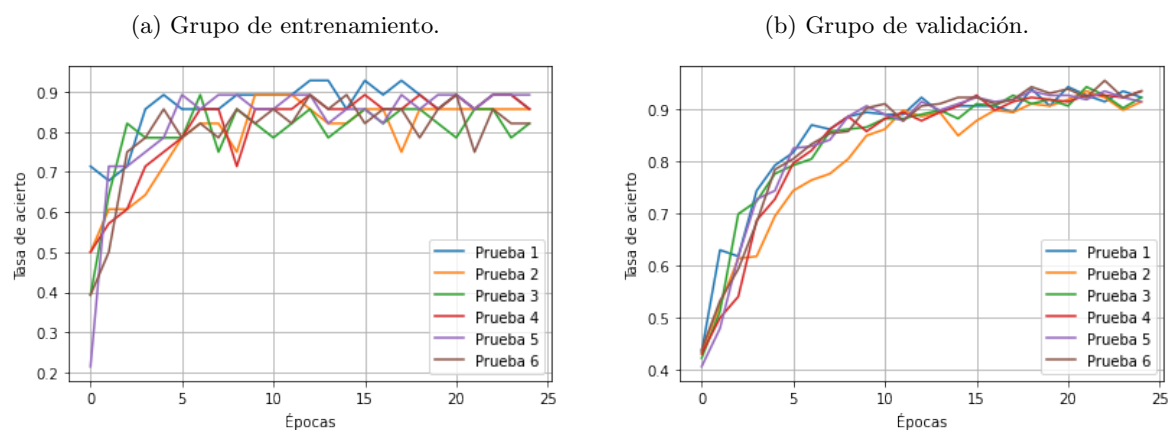


Figura 4.14: Tasa de acierto por época obtenidas para modelo 05 GPU en, a) el grupo de entrenamiento y b) el grupo de validación.



4.7. Resultados globales.

Los resultados obtenidos en las mejores redes neuronales para cada modelo, contemplando las pruebas realizadas usando GPU se encuentran resumidas en la **tabla 4.1**.

Cuadro 4.1: Resultados globales.

| Modelo | Acierto máximo (Entrenamiento) | Acierto máximo (Validación) | Pérdida mínima (Entrenamiento) | Pérdida mínima (Validación) | Tiempo por época [Segundos] | Época óptima y prueba |
|-----------------|--------------------------------|-----------------------------|--------------------------------|-----------------------------|-----------------------------|-----------------------|
| 01 | 91.06 % | 85.71 % | 0.2034 | 0.8915 | 5.42 ± 0.10 | 22 (4) |
| 02 | 89.02 % | 89.29 % | 0.2700 | 0.5022 | 6.15 ± 0.24 | 20 (4) |
| 03 | 94.31 % | 89.29 % | 0.1417 | 0.4294 | 29.84 ± 0.94 | 23 (5) |
| 04 | 87.40 % | 92.86 % | 0.3815 | 0.3794 | 42.96 ± 3.62 | 7 (6) |
| 04* | 79.67 % | 92.86 % | 0.4069 | 0.4382 | 2.70 ± 0.85 | 12 (5) |
| 05 | 87.80 % | 92.86 % | 0.3550 | 0.3714 | 31.19 ± 1.52 | 11 (5) |
| 05 (GPU) | 92.28 % | 92.86 % | 0.2075 | 0.3917 | 1.99 ± 1.99 | 13 (1) |

Las gráficas comparativas de los resultados globales se muestran a continuación

Figura 4.15: Pérdidas por época obtenidas para los 5 modelos y las 2 variantes en GPU en el grupo de entrenamiento.

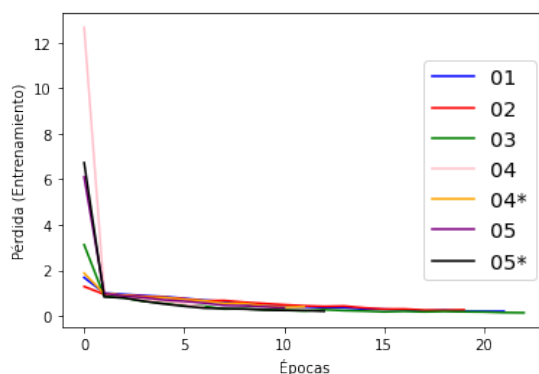
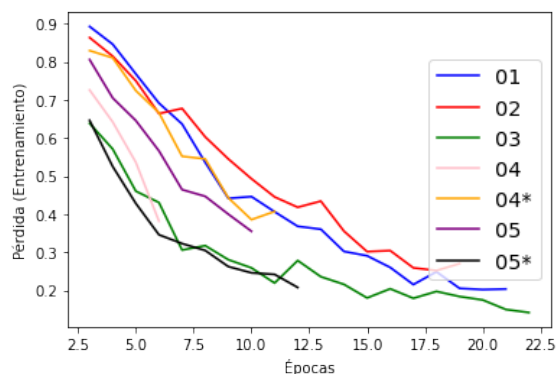


Figura 4.16: Pérdidas por época obtenidas para los 5 modelos y las 2 variantes en GPU en, a) el grupo de entrenamiento (época 3 a 20) y b) el grupo de validación.

(a) Grupo de entrenamiento.



(b) Grupo de validación.

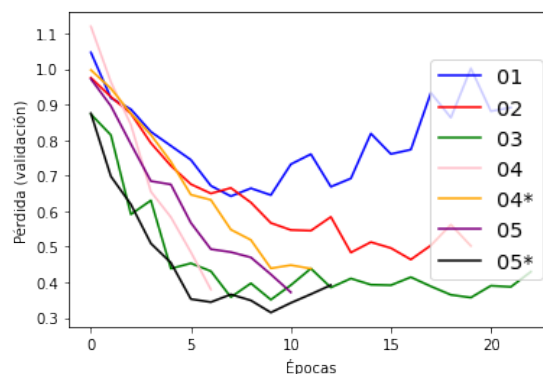
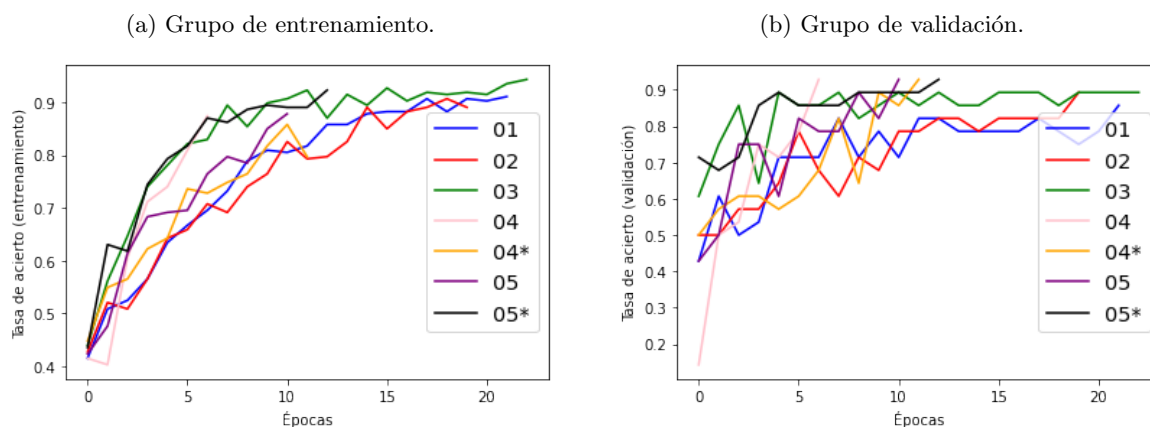


Figura 4.17: Tasa de acierto por época obtenidas para los 5 modelos y las 2 variantes en GPU en, a) el grupo de entrenamiento y b) el grupo de validación.



4.8. Conclusiones

Se logró construir una red neuronal que clasifica bien a los GPCR tipo A con una tasa de acierto de hasta el 92.85 % en ejemplos nuevos que la red neuronal no conocía, donde únicamente se trabajó con la estructura tridimensional de las proteínas. Este resultado es mayor que el obtenido en el trabajo de J. Schmidhuber [1], que corresponde con el 72.1 %. Aunque se usaron diferentes enfoques en ambos proyectos.

La representación de los datos de entrada afecta a la capacidad de clasificación de la red neuronal, así como el volumen de datos para decidir si el algoritmo puede o no ser supervisado. Estos factores se mostraron mucho más relevantes que la selección de hiperparámetros, en el sentido de como afectaron a la tasa de acierto en las diferentes pruebas realizadas a lo largo de este proyecto.

Los factores más relevantes que afectaron el desempeño del algoritmo desarrollado fueron:

1. **Aumentar el espectro de colores en los mapas de distancias.** Lo cual aumento la calidad de los datos de entrada y permitió que el algoritmo pudiera detectar de forma más eficiente los patrones relevantes para realizar las clasificaciones.
2. **Considerar todos los aminoácidos de la proteína.** En la literatura se reporta que alguno bucles intervienen en los mecanismos de cambio de estado funcional de los GPCR, por lo que usar esa información enriquece los datos empleados para entrenar al modelo [1].
3. **Cambiar los $C - \alpha$ por $C - \beta$.** Los $C - \beta$ se encuentran orientados en la misma dirección que el grupo funcional del aminoácido (**sección 2.2.1**). Lo cual parece ser relevante para determinar el estado funcional del GPCR.

Los mejores resultados se obtuvieron en el modelo 04 y el modelo 05 ya que con este último se obtuvieron los resultados más elevados de la tasa de acierto sobre el grupo de validación, también se obtuvo el mejor resultado general al contemplar las tasas de acierto sobre ambos grupos, el de validación y el de entrenamiento, así como los tiempos más breves por época de entre las redes neuronales que obtenían las mejores tasas de acierto.

Por otro lado, el modelo que resulta más fácil de implementar y que también tiene muy buenos resultados es el modelo 04, ya que sólo requiere los archivos PDB para realizar una clasificación, mientras que el modelo 05 requiere el archivo PDB y la numeración de Ballesteros-Weinstein.

También, se logró ejecutar los cálculos de las redes neuronales casi 16 veces más rápido al implementar aceleración por GPU en el caso del modelo 04.

Referente al objetivo se logró caracterizar a los estados conformacionales de las proteínas de la familia A de los GPCR para tres categorías diferentes con base en la actividad farmacológica de los ligandos. Esto se logró mediante el desarrollo e implementación de un algoritmo de clasificación del tipo red neuronal artificial y donde solo se usó la información de la estructura tridimensional así como la actividad farmacológica de los ligandos asociados al GPCR.

Por último, podemos afirmar que se logró afirmar la hipótesis planteada y es de destacar la importancia de haber desarrollado una herramienta confiable para determinar la estructura de GPCRs de la familia A, en tres categorías. Lo cual puede ayudar en el desarrollo de nuevos fármacos al combinarse con estudios de dinámica molecular.

4.9. Perspectivas

Conforme se liberen más archivos de las estructuras para los GPCR de la familia A podemos mejorar los resultados de la red neuronal.

Aunque en este proyecto se intentó usar la técnica de aumento de datos, los resultados obtenidos con este enfoque fueron malos ya que los algoritmos de entrenamiento convergían en una tasa de acierto de alrededor del 50% tras unas pocas épocas, menos de diez, lo cual nos llevó a descartar ese enfoque en esta parte del proyecto. Sin embargo, en un futuro se puede retomar esa estrategia para abordar la falta de suficientes imágenes para entrenar de una forma más eficiente a los modelos.

Al usar los resultados obtenidos en este proyecto se puede planear la extrapolación de los conocimientos generados en este proyecto a otras familias de GPCR o incluso de otras proteínas, podemos tratar un enfoque de transferencia de aprendizaje para adaptar la red neuronal construida a aquellas familias de proteínas que tengan muchos menos datos estructurales que la familia de GPCR tipo A.

Para mejorar el rendimiento de la red neuronal también se puede analizar el desempeño sobre diferentes subgrupos dentro de la familia de la Rodopsina de los GPCR y ver como se desempeña la red neuronal construida en este trabajo.

Bibliografía

- [1] Florian Koensgen y col. «Unsupervised Classification of G-Protein Coupled Receptors and Their Conformational States Using IChem Intramolecular Interaction Patterns». En: *Journal of Chemical Information and Modeling* 59.9 (2019), págs. 3611-3618. ISSN: 15205142. DOI: 10.1021/acs.jcim.9b00054.
- [2] Andrew W. Senior y col. «Improved protein structure prediction using potentials from deep learning». En: *Nature* 577.7792 (2020), págs. 706-710. ISSN: 14764687. DOI: 10.1038/s41586-019-1923-7. URL: <http://dx.doi.org/10.1038/s41586-019-1923-7>.
- [3] J Craig Venter y col. *The Sequence of the Human Genome Downloaded from*. Inf. téc. 2001. URL: www.sciencemag.org/cgi/content/full/291/.
- [4] Stefano Costanzi. «Modeling G protein-coupled receptors and their interactions with ligands». En: *Current Opinion in Structural Biology* 23.2 (2013), págs. 185-190. ISSN: 0959440X. DOI: 10.1016/j.sbi.2013.01.008. URL: <http://dx.doi.org/10.1016/j.sbi.2013.01.008>.
- [5] Anne Grahl y col. «A high-resolution description of β 1-adrenergic receptor functional dynamics and allosteric coupling from backbone NMR». En: *Nature Communications* 11.1 (2020). ISSN: 20411723. DOI: 10.1038/s41467-020-15864-y. URL: <http://dx.doi.org/10.1038/s41467-020-15864-y>.
- [6] Correspondence D E Gloriam y col. «GPCRdb: the G protein-coupled receptor database-an introduction». En: *British Journal of Pharmacology* 173 (2016), pág. 2195. DOI: 10.1111/bph.13509.
- [7] John P. Overington, Bissan Al-Lazikani y Andrew L. Hopkins. «How many drug targets are there?» En: *Nature Reviews Drug Discovery* 5.12 (2006), págs. 993-996. ISSN: 14741776. DOI: 10.1038/nrd2199.
- [8] Daniel Wacker, Raymond C. Stevens y Bryan L. Roth. «How Ligands Illuminate GPCR Molecular Pharmacology». En: *Cell* 170.3 (2017), págs. 414-427. ISSN: 10974172. DOI: 10.1016/j.cell.2017.07.009. URL: <http://dx.doi.org/10.1016/j.cell.2017.07.009>.
- [9] Vignir Isberg y col. «GPCRdb: An information system for G protein-coupled receptors». En: *Nucleic Acids Research* 44.D1 (2016), págs. D356-D364. ISSN: 13624962. DOI: 10.1093/nar/gkv1178.
- [10] Sven Jähnichen. «GPCR Structure-Function». En: *Gnu-Fdl* 7 (2006), págs. 531-556. DOI: 10.1146/annurev-pharmtox-032112-135923. Structure - Function. URL: <http://www.creative-biostructure.com/custom-mempro%E2%84%A2-gpcr-structure-function-services-22.htm>.
- [11] Daniel M. Rosenbaum, Søren G. F. Rasmussen y Brian K. Kobilka. «The structure and function of G-protein-coupled receptors». En: *Nature* 459.7245 (2009), págs. 356-363. ISSN: 0028-0836. DOI: 10.1038/nature08144. URL: <http://www.nature.com/articles/nature08144>.
- [12] Amit Kessel y Nir Ben-Tal. *Introduction to Proteins*. second edi. New York: Chapman y Hall/CRC, 2018. ISBN: 9781315113876. DOI: 10.1201/9781315113876. URL: <https://www.taylorfrancis.com/books/9781498747189>.
- [13] Brian K. Kobilka. «G protein coupled receptor structure and activation». En: *Biochimica et Biophysica Acta - Biomembranes* 1768.4 (2007), págs. 794-807. ISSN: 00052736. DOI: 10.1016/j.bbmem.2006.10.021.
- [14] Joël Bockaert y col. «GPCR interacting proteins (GIP)». En: *Pharmacology and Therapeutics* 103.3 (2004), págs. 203-221. ISSN: 01637258. DOI: 10.1016/j.pharmthera.2004.06.004.

- [15] William I. Weis y Brian K. Kobilka. «The Molecular Basis of G Protein-Coupled Receptor Activation». En: *Annual Review of Biochemistry* 87 (2018), págs. 897-919. ISSN: 15454509. DOI: 10.1146/annurev-biochem-060614-033910.
- [16] C. Hoffmann y col. «Conformational changes in G-protein-coupled receptors - The quest for functionally selective conformations is open». En: *British Journal of Pharmacology* 153.SUPPL. 1 (2008), págs. 358-367. ISSN: 00071188. DOI: 10.1038/sj.bjp.0707615.
- [17] Brian K. Kobilka y Xavier Deupi. «Conformational complexity of G-protein-coupled receptors». En: *Trends in Pharmacological Sciences* 28.8 (2007), págs. 397-406. ISSN: 01656147. DOI: 10.1016/j.tips.2007.06.003.
- [18] Weinstein H. Ballesteros JA. «Integrated methods for the construction of three-dimensional models and computational probing of structure-function relations in G protein-coupled receptors. Methods». En: *Methods in neurosciences* 25 (1995), págs. 366-428. ISSN: 00316997.
- [19] Vignir Isberg y col. *Generic GPCR residue numbers - Aligning topology maps while minding the gaps*. 2015. DOI: 10.1016/j.tips.2014.11.001.
- [20] Christopher Bystroff Mohammed J.Zaki. *Protein Structure Prediction*. Ed. por Mohammed J. Zaki y Christopher Bystroff. Totowa, NJ: Humana Press, 2008. ISBN: 978-1-58829-752-5. DOI: 10.1007/978-1-59745-574-9. URL: <http://link.springer.com/10.1007/978-1-59745-574-9>.
- [21] Navin Kumar Manaswi. *Deep Learning with Applications Using Python*. 2018. ISBN: 978-1-4842-3515-7. DOI: 10.1007/978-1-4842-3516-4.
- [22] Jürgen Schmidhuber. «Deep Learning in neural networks: An overview». En: *Neural Networks* 61 (2015), págs. 85-117. ISSN: 18792782. DOI: 10.1016/j.neunet.2014.09.003. arXiv: 1404.7828. URL: <http://dx.doi.org/10.1016/j.neunet.2014.09.003>.
- [23] Chigozie Nwankpa y col. «Activation Functions: Comparison of trends in Practice and Research for Deep Learning». En: (2018), págs. 1-20. arXiv: 1811.03378. URL: <http://arxiv.org/abs/1811.03378>.
- [24] Robert M Gray. *Entropy and information theory*. Second edi. Springer, 2011. ISBN: 9781441979704.
- [25] Matthew D. Zeiler. «ADADELTA: An Adaptive Learning Rate Method». En: (2012). arXiv: 1212.5701. URL: <http://arxiv.org/abs/1212.5701>.
- [26] ASIMOV. *THE NEURAL NETWORK ZOO THE ASIMOV INSTITUTE*. 2017. URL: <https://www.asimovinstitute.org/neural-network-zoo/><https://www.asimovinstitute.org/neural-network-zoo/><http://www.asimovinstitute.org/neural-network-zoo/> (visitado 08-03-2021).
- [27] Rui Wang, Jingwen Xu y Tony X. Han. «Object instance detection with pruned Alexnet and extended training data». En: *Signal Processing: Image Communication* 70.March 2018 (2019), págs. 145-156. ISSN: 09235965. DOI: 10.1016/j.image.2018.09.013. URL: <https://doi.org/10.1016/j.image.2018.09.013>.
- [28] Fabio Nelli y Fabio Nelli. *Deep Learning with TensorFlow*. 2018, págs. 349-407. ISBN: 9781484230954. DOI: 10.1007/978-1-4842-3913-1_9.
- [29] Nikhil Ketkar y Jojo Moolayil. *Deep Learning with Python*. 2021. ISBN: 9781617294433. DOI: 10.1007/978-1-4842-5364-9.
- [30] Andrew Shaw. *GCP — fast.ai course v3*. 2017. URL: https://course19.fast.ai/gpu_tutorial.htmlhttps://course.fast.ai/start_gcp.html (visitado 05-08-2021).
- [31] Jason Dsouza. «What is a GPU and do you need one in Deep Learning? — by Jason Dsouza — Towards Data Science». En: (2020), págs. 1-11. URL: <https://towardsdatascience.com/what-is-a-gpu-and-do-you-need-one-in-deep-learning-718b9597aa0d>.
- [32] Vincent Vanhoucke, Andrew Senior y Mz Mao. «Improving the speed of neural networks on CPUs». En: *Proc. Deep Learning and ...* (2011), págs. 1-8. ISSN: 9781450329569. URL: <http://research.google.com/pubs/archive/37631.pdf>.
- [33] Aurélien Géron. *Hands on Machine Learning with Scikit-Learn, Keras & TensorFlow*. Ed. por Nicole Tache Rachel Roumeliotis. Second Edi. Vol. 53. 9. O'Reilly, 2019, pág. 287. ISBN: 978-1-492-03264-9.

Capítulo 5

Anexo I, tablas de resultados

5.1. Resultados 01

Se resaltan en azul los valores máximos en las tablas de tasa de acierto y los mínimos en las tablas de perdida.

Cuadro 5.1: Tasa de acierto del grupo de entrenamiento para el modelo 01.

| Época | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|----------|--------|--------|--------|--------|--------|--------|--------|--------|--------|
| Prueba 1 | 0.5244 | 0.4715 | 0.5325 | 0.6057 | 0.6341 | 0.6829 | 0.7033 | 0.7846 | 0.7480 |
| Prueba 2 | 0.4268 | 0.5488 | 0.4756 | 0.5447 | 0.5610 | 0.6016 | 0.6504 | 0.6911 | 0.7195 |
| Prueba 3 | 0.4553 | 0.4878 | 0.4431 | 0.5041 | 0.5122 | 0.6138 | 0.5244 | 0.5772 | 0.6260 |
| Prueba 4 | 0.4146 | 0.5081 | 0.5244 | 0.5650 | 0.6341 | 0.6667 | 0.6951 | 0.7317 | 0.7886 |
| Prueba 5 | 0.4512 | 0.4878 | 0.5163 | 0.5854 | 0.6220 | 0.6260 | 0.6463 | 0.7276 | 0.7154 |
| Prueba 6 | 0.4431 | 0.5000 | 0.5163 | 0.5407 | 0.5650 | 0.6829 | 0.6911 | 0.7154 | 0.7195 |
| Prueba 7 | 0.4593 | 0.4390 | 0.4959 | 0.4797 | 0.5081 | 0.6463 | 0.6138 | 0.6423 | 0.6545 |

| Época | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 |
|----------|--------|--------|--------|--------|--------|--------|--------|--------|--------|
| Prueba 1 | 0.7683 | 0.7886 | 0.8374 | 0.8496 | 0.8659 | 0.8618 | 0.8984 | 0.8862 | 0.8862 |
| Prueba 2 | 0.7561 | 0.7480 | 0.7886 | 0.8333 | 0.8374 | 0.8577 | 0.8862 | 0.8699 | 0.8699 |
| Prueba 3 | 0.6585 | 0.7561 | 0.7520 | 0.7358 | 0.7846 | 0.7927 | 0.8374 | 0.8577 | 0.8415 |
| Prueba 4 | 0.8089 | 0.8049 | 0.8171 | 0.8577 | 0.8577 | 0.8780 | 0.8821 | 0.8821 | 0.9065 |
| Prueba 5 | 0.7683 | 0.8089 | 0.7927 | 0.7805 | 0.8293 | 0.8374 | 0.8780 | 0.8577 | 0.8780 |
| Prueba 6 | 0.7846 | 0.7846 | 0.7967 | 0.8130 | 0.8211 | 0.8659 | 0.8821 | 0.8740 | 0.8577 |
| Prueba 7 | 0.7276 | 0.7480 | 0.7724 | 0.7520 | 0.7764 | 0.8008 | 0.7927 | 0.8496 | 0.8537 |

| Época | 19 | 20 | 21 | 22 | 23 | 24 | 25 |
|----------|--------|--------|--------|--------|--------|--------|--------|
| Prueba 1 | 0.9106 | 0.9228 | 0.8862 | 0.9065 | 0.9024 | 0.9146 | 0.9065 |
| Prueba 2 | 0.9228 | 0.8943 | 0.8862 | 0.8943 | 0.9106 | 0.9187 | 0.9187 |
| Prueba 3 | 0.8455 | 0.8577 | 0.8984 | 0.8862 | 0.8780 | 0.8780 | 0.9268 |
| Prueba 4 | 0.8821 | 0.9065 | 0.9024 | 0.9106 | 0.9065 | 0.9187 | 0.9309 |
| Prueba 5 | 0.9024 | 0.8902 | 0.9065 | 0.8780 | 0.9146 | 0.8943 | 0.9268 |
| Prueba 6 | 0.9268 | 0.8821 | 0.9106 | 0.9106 | 0.9146 | 0.9065 | 0.9228 |
| Prueba 7 | 0.8699 | 0.8415 | 0.8821 | 0.8862 | 0.8699 | 0.9106 | 0.9106 |

Cuadro 5.2: Tasa de acierto del grupo de validación para el modelo 01.

| Época | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|----------|--------|--------|--------|--------|--------|--------|--------|--------|--------|
| Prueba 1 | 0.5714 | 0.5357 | 0.5714 | 0.7143 | 0.7500 | 0.7857 | 0.7500 | 0.7857 | 0.8214 |
| Prueba 2 | 0.5000 | 0.3929 | 0.5000 | 0.5714 | 0.7143 | 0.5714 | 0.7143 | 0.5357 | 0.7143 |
| Prueba 3 | 0.3929 | 0.3929 | 0.5000 | 0.5000 | 0.5714 | 0.5714 | 0.6071 | 0.6071 | 0.6071 |
| Prueba 4 | 0.4286 | 0.6071 | 0.5000 | 0.5357 | 0.7143 | 0.7143 | 0.7143 | 0.8214 | 0.7143 |
| Prueba 5 | 0.5000 | 0.5000 | 0.5000 | 0.5714 | 0.6071 | 0.7500 | 0.6786 | 0.6786 | 0.6071 |
| Prueba 6 | 0.5000 | 0.4286 | 0.5000 | 0.5357 | 0.7143 | 0.7143 | 0.5714 | 0.7500 | 0.7500 |
| Prueba 7 | 0.5000 | 0.5000 | 0.5000 | 0.5000 | 0.5714 | 0.7143 | 0.6071 | 0.7500 | 0.6786 |

| Época | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 |
|----------|--------|--------|--------|--------|--------|--------|--------|--------|--------|
| Prueba 1 | 0.7143 | 0.7857 | 0.8214 | 0.7857 | 0.7857 | 0.8214 | 0.8214 | 0.7857 | 0.8571 |
| Prueba 2 | 0.7857 | 0.7143 | 0.8214 | 0.8214 | 0.8214 | 0.8214 | 0.8214 | 0.7857 | 0.7143 |
| Prueba 3 | 0.7500 | 0.6786 | 0.7857 | 0.5357 | 0.6786 | 0.7143 | 0.7857 | 0.7857 | 0.7857 |
| Prueba 4 | 0.7857 | 0.7143 | 0.8214 | 0.8214 | 0.7857 | 0.7857 | 0.7857 | 0.7857 | 0.8214 |
| Prueba 5 | 0.6429 | 0.7500 | 0.7500 | 0.7500 | 0.7143 | 0.8214 | 0.8214 | 0.8214 | 0.7857 |
| Prueba 6 | 0.8571 | 0.7857 | 0.7857 | 0.8214 | 0.8571 | 0.8571 | 0.8571 | 0.8214 | 0.8214 |
| Prueba 7 | 0.7500 | 0.7143 | 0.7857 | 0.7500 | 0.5714 | 0.8214 | 0.7857 | 0.8214 | 0.8214 |

| Época | 19 | 20 | 21 | 22 | 23 | 24 | 25 |
|----------|--------|--------|--------|--------|--------|--------|--------|
| Prueba 1 | 0.8214 | 0.7857 | 0.7857 | 0.8214 | 0.8214 | 0.8571 | 0.8214 |
| Prueba 2 | 0.7857 | 0.8214 | 0.8214 | 0.8214 | 0.8214 | 0.7857 | 0.8214 |
| Prueba 3 | 0.7857 | 0.7857 | 0.8214 | 0.7857 | 0.7857 | 0.7857 | 0.7857 |
| Prueba 4 | 0.7857 | 0.7500 | 0.7857 | 0.8571 | 0.7857 | 0.7857 | 0.8571 |
| Prueba 5 | 0.8214 | 0.7857 | 0.8214 | 0.7857 | 0.7857 | 0.8214 | 0.8214 |
| Prueba 6 | 0.8214 | 0.8214 | 0.8571 | 0.8214 | 0.7857 | 0.8214 | 0.7857 |
| Prueba 7 | 0.8571 | 0.7857 | 0.7500 | 0.8214 | 0.7857 | 0.7500 | 0.7857 |

Cuadro 5.3: Pérdida del grupo de entrenamiento para el modelo 01.

| Época | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|----------|--------|--------|--------|--------|--------|--------|--------|--------|--------|
| Prueba 1 | 1.5458 | 0.9493 | 0.9243 | 0.8612 | 0.7807 | 0.7317 | 0.6735 | 0.5708 | 0.5523 |
| Prueba 2 | 1.8767 | 0.9900 | 1.0087 | 0.9322 | 0.9120 | 0.8328 | 0.7615 | 0.6883 | 0.6602 |
| Prueba 3 | 2.6449 | 0.9975 | 0.9895 | 0.9454 | 0.9204 | 0.8898 | 0.9050 | 0.8718 | 0.7776 |
| Prueba 4 | 1.6882 | 1.0099 | 0.9458 | 0.8925 | 0.8459 | 0.7680 | 0.6913 | 0.6366 | 0.5357 |
| Prueba 5 | 1.1871 | 0.9761 | 0.9311 | 0.8731 | 0.8329 | 0.8264 | 0.7440 | 0.6656 | 0.6614 |
| Prueba 6 | 2.1251 | 0.9857 | 0.9726 | 0.9154 | 0.8789 | 0.8068 | 0.7893 | 0.6753 | 0.5994 |
| Prueba 7 | 2.3997 | 0.9836 | 0.9791 | 0.9407 | 0.9411 | 0.8699 | 0.8267 | 0.7831 | 0.7573 |

| Época | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 |
|----------|--------|--------|--------|--------|--------|--------|--------|--------|--------|
| Prueba 1 | 0.5129 | 0.4562 | 0.4002 | 0.3410 | 0.3304 | 0.3026 | 0.2812 | 0.2704 | 0.2726 |
| Prueba 2 | 0.6015 | 0.5375 | 0.4861 | 0.4183 | 0.3850 | 0.3196 | 0.3153 | 0.3030 | 0.2716 |
| Prueba 3 | 0.7220 | 0.6307 | 0.6023 | 0.5852 | 0.5406 | 0.4646 | 0.4488 | 0.3715 | 0.3408 |
| Prueba 4 | 0.4413 | 0.4460 | 0.4062 | 0.3680 | 0.3604 | 0.3021 | 0.2907 | 0.2601 | 0.2151 |
| Prueba 5 | 0.5727 | 0.5254 | 0.5188 | 0.4721 | 0.4329 | 0.3923 | 0.3204 | 0.3073 | 0.2956 |
| Prueba 6 | 0.5305 | 0.4803 | 0.4635 | 0.4216 | 0.3745 | 0.3411 | 0.3028 | 0.3033 | 0.3092 |
| Prueba 7 | 0.6937 | 0.6336 | 0.5571 | 0.5292 | 0.5160 | 0.4540 | 0.4155 | 0.3690 | 0.3584 |

| Época | 19 | 20 | 21 | 22 | 23 | 24 | 25 |
|----------|--------|--------|--------|--------|--------|--------|--------|
| Prueba 1 | 0.2694 | 0.2639 | 0.2607 | 0.2878 | 0.2326 | 0.1782 | 0.1923 |
| Prueba 2 | 0.2393 | 0.2537 | 0.2285 | 0.2360 | 0.2231 | 0.1946 | 0.2185 |
| Prueba 3 | 0.3316 | 0.3084 | 0.2735 | 0.2363 | 0.2680 | 0.2394 | 0.2258 |
| Prueba 4 | 0.2486 | 0.2051 | 0.2020 | 0.2034 | 0.1919 | 0.2062 | 0.1716 |
| Prueba 5 | 0.2526 | 0.2694 | 0.2331 | 0.2615 | 0.2326 | 0.2022 | 0.1900 |
| Prueba 6 | 0.2536 | 0.2622 | 0.2227 | 0.2217 | 0.2860 | 0.2288 | 0.1992 |
| Prueba 7 | 0.3525 | 0.3481 | 0.3097 | 0.2636 | 0.2730 | 0.2315 | 0.2089 |

Cuadro 5.4: Pérdida del grupo de validación para el modelo 01.

| Época | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|----------|--------|--------|--------|--------|--------|--------|--------|--------|--------|
| Prueba 1 | 0.9072 | 0.9459 | 0.8482 | 0.7428 | 0.7171 | 0.6919 | 0.7273 | 0.7435 | 0.6731 |
| Prueba 2 | 0.9620 | 0.9801 | 0.8939 | 0.8662 | 0.8639 | 0.8294 | 0.7204 | 0.7618 | 0.6846 |
| Prueba 3 | 1.0110 | 0.9924 | 0.9242 | 0.9120 | 0.8866 | 0.8437 | 0.8498 | 0.7797 | 0.7241 |
| Prueba 4 | 1.0465 | 0.9184 | 0.8867 | 0.8234 | 0.7838 | 0.7444 | 0.6714 | 0.6419 | 0.6643 |
| Prueba 5 | 0.9500 | 0.9316 | 0.8724 | 0.7813 | 0.7586 | 0.7345 | 0.6988 | 0.6102 | 0.6698 |
| Prueba 6 | 0.9702 | 0.9569 | 0.8992 | 0.8594 | 0.7938 | 0.7891 | 0.6963 | 0.6789 | 0.6481 |
| Prueba 7 | 0.9859 | 1.0674 | 0.9626 | 0.9734 | 0.9159 | 0.8224 | 0.7872 | 0.7407 | 0.6997 |

| Época | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 |
|----------|--------|--------|--------|--------|--------|--------|--------|--------|--------|
| Prueba 1 | 0.8593 | 0.7893 | 0.7861 | 0.7301 | 0.7646 | 0.9279 | 0.9691 | 0.9900 | 0.9031 |
| Prueba 2 | 0.7210 | 0.7387 | 0.7475 | 0.7566 | 0.6845 | 0.7030 | 0.7280 | 0.7710 | 0.8521 |
| Prueba 3 | 0.6754 | 0.6871 | 0.6507 | 0.7211 | 0.6090 | 0.6442 | 0.6104 | 0.6564 | 0.6431 |
| Prueba 4 | 0.6451 | 0.7320 | 0.7604 | 0.6686 | 0.6920 | 0.8184 | 0.7607 | 0.7732 | 0.9329 |
| Prueba 5 | 0.5992 | 0.6354 | 0.6174 | 0.5536 | 0.6058 | 0.5083 | 0.5593 | 0.5589 | 0.5521 |
| Prueba 6 | 0.6245 | 0.7113 | 0.5948 | 0.7567 | 0.7249 | 0.7305 | 0.7203 | 0.8063 | 0.5799 |
| Prueba 7 | 0.7247 | 0.7097 | 0.7572 | 0.7106 | 0.6949 | 0.6999 | 0.6671 | 0.7435 | 0.6638 |

| Época | 19 | 20 | 21 | 22 | 23 | 24 | 25 |
|----------|--------|--------|--------|--------|--------|--------|--------|
| Prueba 1 | 0.6693 | 0.9080 | 0.8667 | 0.9623 | 1.3916 | 1.1915 | 1.1542 |
| Prueba 2 | 0.8345 | 1.0360 | 0.8650 | 0.7246 | 0.7420 | 0.9514 | 0.8471 |
| Prueba 3 | 0.6384 | 0.6278 | 0.6160 | 0.6260 | 0.6854 | 0.6369 | 0.6695 |
| Prueba 4 | 0.8628 | 1.0019 | 0.8813 | 0.8915 | 0.9008 | 0.8667 | 0.9945 |
| Prueba 5 | 0.5541 | 0.6457 | 0.6616 | 0.6276 | 0.6887 | 0.7351 | 0.7584 |
| Prueba 6 | 0.6384 | 0.7077 | 0.6362 | 1.1410 | 0.6163 | 0.5662 | 0.7448 |
| Prueba 7 | 0.6519 | 0.6534 | 0.7286 | 0.7917 | 0.7633 | 0.8587 | 0.8989 |

Cuadro 5.5: Tiempo de entrenamiento por época para el modelo 01 en segundos.

| Época | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|----------|---------|---------|---------|---------|---------|---------|---------|---------|---------|
| Prueba 1 | 6.4573 | 5.6146 | 5.7314 | 5.5262 | 5.5185 | 5.5160 | 5.4114 | 5.8203 | 5.6667 |
| Prueba 2 | 5.9277 | 5.4287 | 5.3966 | 5.4007 | 5.3842 | 5.3977 | 5.4003 | 5.4741 | 5.5236 |
| Prueba 3 | 5.7975 | 5.3517 | 5.4026 | 5.3742 | 5.3708 | 5.6069 | 5.3741 | 5.5121 | 5.3670 |
| Prueba 4 | 5.8451 | 5.4288 | 5.4356 | 5.3805 | 5.4119 | 5.3911 | 5.4324 | 5.4112 | 5.3888 |
| Prueba 5 | 5.7747 | 5.3670 | 5.3494 | 5.3599 | 5.3657 | 5.3894 | 5.3909 | 5.3871 | 5.3620 |
| Prueba 6 | 5.8580 | 5.5026 | 5.6361 | 5.5331 | 5.5823 | 5.6485 | 5.9241 | 6.7305 | 12.0896 |
| Prueba 7 | 16.7319 | 10.7424 | 12.5992 | 12.1473 | 11.9902 | 10.5238 | 11.8494 | 12.1172 | 12.7293 |

| Época | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 |
|----------|---------|---------|---------|---------|---------|---------|---------|---------|---------|
| Prueba 1 | 5.4952 | 5.4678 | 5.5542 | 5.4191 | 5.4232 | 5.4406 | 5.3987 | 5.3960 | 5.4280 |
| Prueba 2 | 5.4972 | 5.9556 | 5.5453 | 5.3819 | 5.4034 | 5.3623 | 5.3769 | 5.3981 | 5.4475 |
| Prueba 3 | 5.3717 | 5.3767 | 5.3870 | 5.5837 | 5.6704 | 5.4111 | 5.3911 | 5.4870 | 5.3562 |
| Prueba 4 | 5.3929 | 5.4141 | 5.3983 | 5.4291 | 5.4510 | 5.4206 | 5.3822 | 5.3916 | 5.3894 |
| Prueba 5 | 5.3592 | 5.4623 | 5.3547 | 5.3905 | 5.3622 | 5.3625 | 5.3818 | 5.3668 | 5.3526 |
| Prueba 6 | 12.2008 | 12.4863 | 11.6400 | 11.5019 | 11.4912 | 11.7599 | 11.6993 | 11.2810 | 11.7026 |
| Prueba 7 | 10.0683 | 12.6112 | 12.4225 | 11.4935 | 10.9023 | 11.9868 | 11.7674 | 11.7424 | 11.2391 |

| Época | 19 | 20 | 21 | 22 | 23 | 24 | 25 |
|----------|---------|---------|---------|---------|---------|---------|---------|
| Prueba 1 | 5.4390 | 5.3798 | 5.3644 | 5.3952 | 5.4663 | 5.3986 | 5.3422 |
| Prueba 2 | 5.4068 | 5.5069 | 5.4207 | 5.9011 | 5.4099 | 5.3931 | 5.3841 |
| Prueba 3 | 5.3704 | 5.3993 | 5.3642 | 5.3776 | 5.3695 | 5.3533 | 5.4003 |
| Prueba 4 | 5.3948 | 5.3812 | 5.3997 | 5.4029 | 5.4026 | 5.3953 | 5.4781 |
| Prueba 5 | 5.3661 | 5.3393 | 5.4013 | 5.4563 | 5.4575 | 5.3939 | 5.4001 |
| Prueba 6 | 12.1644 | 11.5285 | 12.1223 | 12.6153 | 12.1943 | 10.2778 | 12.4110 |
| Prueba 7 | 11.8219 | 12.0020 | 11.6195 | 12.0826 | 12.7542 | 12.4895 | 10.4794 |

5.2. Resultados 02

Se resaltan en azul los valores máximos en las tablas de tasa de acierto y los mínimos en las tablas de perdida.

Cuadro 5.6: Tasa de acierto en el grupo de entrenamiento por época para el modelo 02.

| Época | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|----------|--------|--------|--------|--------|--------|--------|--------|--------|--------|
| Prueba 1 | 0.4715 | 0.4431 | 0.4919 | 0.5813 | 0.6179 | 0.6301 | 0.6545 | 0.7195 | 0.7642 |
| Prueba 2 | 0.4715 | 0.5244 | 0.5366 | 0.6545 | 0.7114 | 0.6992 | 0.7439 | 0.7602 | 0.7967 |
| Prueba 3 | 0.3537 | 0.4390 | 0.5122 | 0.6057 | 0.6789 | 0.6829 | 0.7439 | 0.7764 | 0.7927 |
| Prueba 4 | 0.4268 | 0.5203 | 0.5081 | 0.5650 | 0.6423 | 0.6585 | 0.7073 | 0.6911 | 0.7398 |
| Prueba 5 | 0.4106 | 0.5000 | 0.5732 | 0.6260 | 0.6504 | 0.7154 | 0.7520 | 0.7602 | 0.8211 |
| Prueba 6 | 0.4228 | 0.5203 | 0.5976 | 0.6382 | 0.6301 | 0.6992 | 0.7114 | 0.7276 | 0.7439 |
| Prueba 7 | 0.4187 | 0.5407 | 0.5813 | 0.5854 | 0.6423 | 0.7114 | 0.6260 | 0.7195 | 0.7154 |

| Época | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 |
|----------|--------|--------|--------|--------|--------|--------|--------|--------|--------|
| Prueba 1 | 0.7561 | 0.7927 | 0.7846 | 0.7683 | 0.8333 | 0.8415 | 0.8374 | 0.8740 | 0.8659 |
| Prueba 2 | 0.8618 | 0.8577 | 0.8780 | 0.8618 | 0.8455 | 0.8659 | 0.8943 | 0.8902 | 0.8943 |
| Prueba 3 | 0.7846 | 0.8293 | 0.8415 | 0.8455 | 0.8577 | 0.8821 | 0.8862 | 0.8902 | 0.8902 |
| Prueba 4 | 0.7642 | 0.8252 | 0.7927 | 0.7967 | 0.8252 | 0.8902 | 0.8496 | 0.8821 | 0.8902 |
| Prueba 5 | 0.8455 | 0.8537 | 0.8577 | 0.8862 | 0.9065 | 0.8618 | 0.8984 | 0.8943 | 0.9024 |
| Prueba 6 | 0.7602 | 0.7764 | 0.8008 | 0.7967 | 0.8049 | 0.8374 | 0.8577 | 0.8740 | 0.8455 |
| Prueba 7 | 0.7683 | 0.7561 | 0.7886 | 0.8293 | 0.8618 | 0.8211 | 0.8577 | 0.8496 | 0.8699 |

| Época | 19 | 20 | 21 | 22 | 23 | 24 | 25 |
|----------|--------|--------|--------|--------|--------|--------|--------|
| Prueba 1 | 0.8740 | 0.8780 | 0.8943 | 0.8984 | 0.9146 | 0.8821 | 0.9187 |
| Prueba 2 | 0.9106 | 0.9065 | 0.9390 | 0.8943 | 0.9309 | 0.9065 | 0.9187 |
| Prueba 3 | 0.9065 | 0.9024 | 0.9146 | 0.9309 | 0.9024 | 0.9065 | 0.8943 |
| Prueba 4 | 0.9065 | 0.8902 | 0.8862 | 0.8984 | 0.9065 | 0.9146 | 0.9065 |
| Prueba 5 | 0.9146 | 0.9065 | 0.9065 | 0.9106 | 0.9146 | 0.9024 | 0.9268 |
| Prueba 6 | 0.8659 | 0.8699 | 0.9106 | 0.8780 | 0.8699 | 0.9106 | 0.8862 |
| Prueba 7 | 0.8699 | 0.8902 | 0.9228 | 0.9106 | 0.9065 | 0.9106 | 0.9228 |

Cuadro 5.7: Tasa de acierto del grupo de validación para el modelo 02.

| Época | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|----------|--------|--------|--------|--------|--------|--------|--------|--------|--------|
| Prueba 1 | 0.3929 | 0.5000 | 0.5000 | 0.6786 | 0.5357 | 0.8214 | 0.7857 | 0.8929 | 0.8571 |
| Prueba 2 | 0.5714 | 0.5000 | 0.5714 | 0.7857 | 0.7500 | 0.8214 | 0.7500 | 0.7857 | 0.8214 |
| Prueba 3 | 0.3929 | 0.4643 | 0.6071 | 0.6429 | 0.8929 | 0.7500 | 0.8214 | 0.8571 | 0.8929 |
| Prueba 4 | 0.5000 | 0.5000 | 0.5714 | 0.5714 | 0.6429 | 0.7857 | 0.6786 | 0.6071 | 0.7143 |
| Prueba 5 | 0.5000 | 0.5000 | 0.5714 | 0.7500 | 0.7500 | 0.6429 | 0.7857 | 0.8929 | 0.7857 |
| Prueba 6 | 0.5714 | 0.5714 | 0.5357 | 0.5357 | 0.6429 | 0.6071 | 0.5714 | 0.6786 | 0.7500 |
| Prueba 7 | 0.5000 | 0.5714 | 0.6071 | 0.5714 | 0.6786 | 0.7500 | 0.6786 | 0.6429 | 0.7500 |

| Época | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 |
|----------|--------|--------|--------|--------|--------|--------|--------|--------|--------|
| Prueba 1 | 0.8929 | 0.8571 | 0.7143 | 0.8571 | 0.8214 | 0.8571 | 0.8571 | 0.8214 | 0.8571 |
| Prueba 2 | 0.7857 | 0.8571 | 0.8929 | 0.7500 | 0.8571 | 0.8929 | 0.8929 | 0.8929 | 0.8214 |
| Prueba 3 | 0.8214 | 0.8214 | 0.8571 | 0.8571 | 0.8929 | 0.8214 | 0.8214 | 0.8929 | 0.8929 |
| Prueba 4 | 0.6786 | 0.7857 | 0.7857 | 0.8214 | 0.8214 | 0.7857 | 0.8214 | 0.8214 | 0.8214 |
| Prueba 5 | 0.8571 | 0.8571 | 0.8214 | 0.8214 | 0.8214 | 0.8571 | 0.8571 | 0.8929 | 0.8929 |
| Prueba 6 | 0.7143 | 0.6786 | 0.8214 | 0.7857 | 0.7857 | 0.8214 | 0.8214 | 0.8214 | 0.8571 |
| Prueba 7 | 0.6429 | 0.7857 | 0.8214 | 0.7857 | 0.8571 | 0.8571 | 0.7143 | 0.7500 | 0.7857 |

| Época | 19 | 20 | 21 | 22 | 23 | 24 | 25 |
|----------|--------|--------|--------|--------|--------|--------|--------|
| Prueba 1 | 0.8929 | 0.8214 | 0.8571 | 0.8214 | 0.8929 | 0.8571 | 0.8571 |
| Prueba 2 | 0.8929 | 0.8571 | 0.8929 | 0.8571 | 0.8571 | 0.8929 | 0.8929 |
| Prueba 3 | 0.8571 | 0.8571 | 0.8929 | 0.8571 | 0.8571 | 0.8571 | 0.8929 |
| Prueba 4 | 0.8214 | 0.8929 | 0.8571 | 0.8571 | 0.8929 | 0.7857 | 0.7857 |
| Prueba 5 | 0.8929 | 0.8929 | 0.7857 | 0.8571 | 0.8929 | 0.8929 | 0.8571 |
| Prueba 6 | 0.7500 | 0.8571 | 0.8571 | 0.8929 | 0.8571 | 0.8214 | 0.8571 |
| Prueba 7 | 0.8214 | 0.8571 | 0.8571 | 0.8571 | 0.8571 | 0.8571 | 0.8214 |

Cuadro 5.8: Pérdida del grupo de entrenamiento para el modelo 02.

| Época | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|----------|--------|--------|--------|--------|--------|--------|--------|--------|--------|
| Prueba 1 | 1.7024 | 0.9715 | 0.9362 | 0.8913 | 0.8378 | 0.7906 | 0.7421 | 0.6528 | 0.6022 |
| Prueba 2 | 1.1703 | 0.9378 | 0.8805 | 0.8349 | 0.7412 | 0.6516 | 0.5636 | 0.5824 | 0.5043 |
| Prueba 3 | 1.1968 | 0.9885 | 0.9583 | 0.8712 | 0.7505 | 0.6870 | 0.6203 | 0.5540 | 0.4872 |
| Prueba 4 | 1.2923 | 0.9494 | 0.9068 | 0.8634 | 0.8144 | 0.7500 | 0.6638 | 0.6773 | 0.6027 |
| Prueba 5 | 1.5270 | 0.9666 | 0.8843 | 0.8427 | 0.7509 | 0.6492 | 0.5746 | 0.5296 | 0.4416 |
| Prueba 6 | 1.8713 | 0.9948 | 0.9041 | 0.8527 | 0.8332 | 0.7414 | 0.7009 | 0.6602 | 0.5752 |
| Prueba 7 | 1.7141 | 0.9620 | 0.9093 | 0.8797 | 0.8097 | 0.7119 | 0.7441 | 0.6348 | 0.6070 |

| Época | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 |
|----------|--------|--------|--------|--------|--------|--------|--------|--------|--------|
| Prueba 1 | 0.5769 | 0.4938 | 0.5187 | 0.5110 | 0.4256 | 0.3724 | 0.3679 | 0.3301 | 0.3088 |
| Prueba 2 | 0.3687 | 0.3554 | 0.3188 | 0.3089 | 0.3586 | 0.3120 | 0.2591 | 0.2464 | 0.2603 |
| Prueba 3 | 0.4621 | 0.4144 | 0.3961 | 0.3307 | 0.3245 | 0.3147 | 0.3102 | 0.2483 | 0.2587 |
| Prueba 4 | 0.5448 | 0.4928 | 0.4451 | 0.4180 | 0.4341 | 0.3553 | 0.3016 | 0.3042 | 0.2590 |
| Prueba 5 | 0.4001 | 0.3468 | 0.3126 | 0.2755 | 0.2254 | 0.2670 | 0.2487 | 0.2532 | 0.2133 |
| Prueba 6 | 0.5014 | 0.5008 | 0.4309 | 0.4186 | 0.4299 | 0.3750 | 0.3377 | 0.3094 | 0.3214 |
| Prueba 7 | 0.5060 | 0.4668 | 0.4158 | 0.3644 | 0.3257 | 0.3355 | 0.3125 | 0.3078 | 0.2932 |

| Época | 19 | 20 | 21 | 22 | 23 | 24 | 25 |
|----------|--------|--------|--------|--------|--------|--------|--------|
| Prueba 1 | 0.2828 | 0.3130 | 0.2580 | 0.2477 | 0.2314 | 0.2371 | 0.2229 |
| Prueba 2 | 0.2576 | 0.2425 | 0.1806 | 0.2087 | 0.1719 | 0.2117 | 0.1918 |
| Prueba 3 | 0.2331 | 0.2374 | 0.2172 | 0.2118 | 0.1966 | 0.1909 | 0.2034 |
| Prueba 4 | 0.2519 | 0.2700 | 0.2590 | 0.2375 | 0.2158 | 0.2088 | 0.2425 |
| Prueba 5 | 0.1968 | 0.2090 | 0.2077 | 0.2152 | 0.2114 | 0.2221 | 0.1754 |
| Prueba 6 | 0.2975 | 0.3106 | 0.2405 | 0.2358 | 0.2340 | 0.2241 | 0.2195 |
| Prueba 7 | 0.2866 | 0.2782 | 0.2400 | 0.2249 | 0.1881 | 0.2262 | 0.2089 |

Cuadro 5.9: Pérdida del grupo de validación para el modelo 02.

| Época | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|----------|--------|--------|--------|--------|--------|--------|--------|--------|--------|
| Prueba 1 | 0.9360 | 0.9214 | 0.8785 | 0.8109 | 0.7263 | 0.6795 | 0.6638 | 0.5417 | 0.5091 |
| Prueba 2 | 0.9724 | 0.8947 | 0.8760 | 0.7093 | 0.6154 | 0.6527 | 0.6091 | 0.5753 | 0.5246 |
| Prueba 3 | 0.9767 | 0.9581 | 0.8842 | 0.7751 | 0.6595 | 0.6214 | 0.6031 | 0.5732 | 0.5485 |
| Prueba 4 | 0.9753 | 0.9217 | 0.8743 | 0.7917 | 0.7276 | 0.6758 | 0.6497 | 0.6652 | 0.6244 |
| Prueba 5 | 1.0004 | 0.8768 | 0.8226 | 0.7258 | 0.6823 | 0.5947 | 0.6126 | 0.5654 | 0.5613 |
| Prueba 6 | 1.0180 | 0.8855 | 0.8319 | 0.9145 | 0.7361 | 0.7284 | 0.7272 | 0.6062 | 0.5807 |
| Prueba 7 | 0.9502 | 0.9551 | 0.8843 | 0.8083 | 0.7240 | 0.6452 | 0.7620 | 0.7606 | 0.6349 |

| Época | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 |
|----------|--------|--------|--------|--------|--------|--------|--------|--------|--------|
| Prueba 1 | 0.5542 | 0.5168 | 0.5874 | 0.4965 | 0.5245 | 0.4514 | 0.5074 | 0.5108 | 0.5004 |
| Prueba 2 | 0.5278 | 0.5199 | 0.4948 | 0.6666 | 0.4772 | 0.4483 | 0.4998 | 0.5523 | 0.5308 |
| Prueba 3 | 0.5388 | 0.5133 | 0.5738 | 0.5164 | 0.5387 | 0.6103 | 0.5475 | 0.5678 | 0.4994 |
| Prueba 4 | 0.5666 | 0.5469 | 0.5450 | 0.5837 | 0.4835 | 0.5128 | 0.4961 | 0.4637 | 0.5036 |
| Prueba 5 | 0.5843 | 0.5752 | 0.5737 | 0.5584 | 0.5702 | 0.5477 | 0.5669 | 0.5278 | 0.5120 |
| Prueba 6 | 0.6120 | 0.5762 | 0.5216 | 0.5135 | 0.5263 | 0.4582 | 0.4729 | 0.4598 | 0.5384 |
| Prueba 7 | 0.7231 | 0.5885 | 0.5371 | 0.5348 | 0.4980 | 0.5555 | 0.6797 | 0.6740 | 0.7489 |

| Época | 19 | 20 | 21 | 22 | 23 | 24 | 25 |
|----------|--------|--------|--------|--------|--------|--------|--------|
| Prueba 1 | 0.4447 | 0.5259 | 0.5220 | 0.5912 | 0.6009 | 0.6377 | 0.5395 |
| Prueba 2 | 0.5273 | 0.5216 | 0.6446 | 0.6276 | 0.6774 | 0.6182 | 0.5827 |
| Prueba 3 | 0.5785 | 0.6304 | 0.6007 | 0.6955 | 0.7094 | 0.6485 | 0.5980 |
| Prueba 4 | 0.5619 | 0.5022 | 0.5163 | 0.5547 | 0.5391 | 0.6846 | 0.5650 |
| Prueba 5 | 0.5001 | 0.5265 | 0.6064 | 0.5000 | 0.5044 | 0.5219 | 0.5695 |
| Prueba 6 | 0.5591 | 0.5209 | 0.4834 | 0.4906 | 0.4970 | 0.5378 | 0.5072 |
| Prueba 7 | 0.6089 | 0.5574 | 0.6339 | 0.6418 | 0.6471 | 0.6270 | 0.6506 |

Cuadro 5.10: Tiempo de entrenamiento por época para el modelo 02 en segundos.

| Época | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|----------|--------|--------|--------|--------|--------|--------|--------|--------|--------|
| Prueba 1 | 6.8131 | 6.1166 | 5.8852 | 5.8742 | 5.9477 | 5.9826 | 5.8823 | 5.8977 | 5.8907 |
| Prueba 2 | 6.4361 | 5.9063 | 5.8744 | 5.8752 | 5.8705 | 5.9008 | 6.1137 | 5.9187 | 5.8758 |
| Prueba 3 | 6.6370 | 5.8145 | 6.0404 | 5.8036 | 5.6260 | 5.6232 | 5.6100 | 5.6264 | 5.6467 |
| Prueba 4 | 6.3529 | 5.9270 | 5.9077 | 5.9152 | 5.9177 | 5.9632 | 6.0065 | 6.0797 | 5.9729 |
| Prueba 5 | 6.6744 | 6.2435 | 6.1360 | 6.1024 | 6.1165 | 6.1934 | 6.2917 | 6.1369 | 6.0928 |
| Prueba 6 | 6.6301 | 6.1523 | 6.1471 | 6.1426 | 6.1397 | 6.9914 | 6.6652 | 6.3144 | 6.3166 |
| Prueba 7 | 6.1470 | 5.6679 | 5.6463 | 5.6497 | 5.6415 | 5.6744 | 5.6938 | 5.6141 | 5.6079 |

| Época | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 |
|----------|--------|--------|--------|--------|--------|--------|--------|--------|--------|
| Prueba 1 | 5.9148 | 6.0413 | 6.0968 | 5.9344 | 5.9286 | 5.9013 | 5.8762 | 5.9205 | 5.8732 |
| Prueba 2 | 5.9412 | 5.8433 | 5.9558 | 6.4445 | 6.2808 | 6.2266 | 5.9207 | 6.0224 | 5.9152 |
| Prueba 3 | 5.6082 | 5.7447 | 5.8062 | 5.7051 | 5.6258 | 5.6350 | 5.6365 | 5.6710 | 5.6299 |
| Prueba 4 | 6.0060 | 6.1377 | 6.5239 | 6.6989 | 6.3260 | 6.5057 | 6.2595 | 6.3708 | 6.1253 |
| Prueba 5 | 6.1137 | 6.1487 | 6.1958 | 6.0723 | 6.0768 | 6.0773 | 6.1159 | 6.0873 | 6.0853 |
| Prueba 6 | 6.3678 | 6.3130 | 6.1653 | 6.4016 | 6.1912 | 6.2101 | 6.2499 | 6.1389 | 6.1224 |
| Prueba 7 | 5.7008 | 5.7211 | 5.6058 | 5.5934 | 5.6029 | 5.5915 | 5.5995 | 5.6102 | 5.6149 |

| Época | 19 | 20 | 21 | 22 | 23 | 24 | 25 |
|----------|--------|--------|--------|--------|--------|--------|--------|
| Prueba 1 | 5.8972 | 5.9122 | 5.9058 | 6.0540 | 5.9718 | 5.8984 | 5.6921 |
| Prueba 2 | 5.8588 | 6.1885 | 5.5184 | 6.2927 | 6.5534 | 6.1546 | 5.5477 |
| Prueba 3 | 5.6805 | 5.6752 | 5.6418 | 5.7223 | 5.7900 | 5.6243 | 5.5191 |
| Prueba 4 | 6.0211 | 6.0120 | 5.9842 | 6.0020 | 5.9735 | 6.0010 | 5.8937 |
| Prueba 5 | 6.0660 | 6.0916 | 6.1382 | 6.1869 | 6.1003 | 6.0883 | 6.0002 |
| Prueba 6 | 6.1756 | 6.1776 | 6.2040 | 6.0992 | 6.1405 | 6.2249 | 6.4145 |
| Prueba 7 | 5.6110 | 5.6090 | 5.7344 | 5.6415 | 5.6248 | 5.6320 | 5.4948 |

5.3. Resultados 03

Se resaltan en azul los valores máximos en las tablas de tasa de acierto y los mínimos en las tablas de pérdida.

Cuadro 5.11: Tasa de acierto del grupo de entrenamiento para el modelo 03.

| Época | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|----------|--------|--------|--------|--------|--------|--------|--------|--------|--------|
| Prueba 1 | 0.4065 | 0.5081 | 0.4756 | 0.5244 | 0.5813 | 0.6260 | 0.7154 | 0.6829 | 0.7561 |
| Prueba 2 | 0.3943 | 0.5122 | 0.6260 | 0.6911 | 0.7317 | 0.7317 | 0.7480 | 0.7805 | 0.8008 |
| Prueba 3 | 0.3537 | 0.5122 | 0.5285 | 0.6341 | 0.6911 | 0.7439 | 0.7805 | 0.7967 | 0.8415 |
| Prueba 4 | 0.4350 | 0.4756 | 0.5325 | 0.6463 | 0.7154 | 0.7724 | 0.8374 | 0.8821 | 0.8780 |
| Prueba 5 | 0.4390 | 0.5610 | 0.6463 | 0.7398 | 0.7805 | 0.8211 | 0.8293 | 0.8943 | 0.8537 |
| Prueba 6 | 0.4390 | 0.4959 | 0.5691 | 0.6423 | 0.7195 | 0.7439 | 0.7724 | 0.8537 | 0.8659 |
| Prueba 7 | 0.4593 | 0.6138 | 0.6545 | 0.7317 | 0.8130 | 0.8374 | 0.8537 | 0.8618 | 0.8862 |

| Época | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 |
|----------|--------|--------|--------|--------|--------|--------|--------|--------|--------|
| Prueba 1 | 0.7683 | 0.8455 | 0.8089 | 0.8740 | 0.8780 | 0.8537 | 0.8943 | 0.8943 | 0.8984 |
| Prueba 2 | 0.7886 | 0.8333 | 0.8496 | 0.8537 | 0.8780 | 0.8943 | 0.9187 | 0.9065 | 0.9228 |
| Prueba 3 | 0.8577 | 0.8821 | 0.8943 | 0.8943 | 0.8862 | 0.8780 | 0.9187 | 0.9106 | 0.9146 |
| Prueba 4 | 0.9024 | 0.8780 | 0.9065 | 0.9065 | 0.8984 | 0.9390 | 0.8902 | 0.9350 | 0.9187 |
| Prueba 5 | 0.8984 | 0.9065 | 0.9228 | 0.8699 | 0.9146 | 0.8943 | 0.9268 | 0.9024 | 0.9187 |
| Prueba 6 | 0.8821 | 0.8537 | 0.9024 | 0.9065 | 0.8984 | 0.9146 | 0.8943 | 0.9390 | 0.9187 |
| Prueba 7 | 0.8943 | 0.9106 | 0.9024 | 0.9268 | 0.9350 | 0.9187 | 0.9309 | 0.9228 | 0.9024 |

| Época | 19 | 20 | 21 | 22 | 23 | 24 | 25 |
|----------|--------|--------|--------|--------|--------|--------|--------|
| Prueba 1 | 0.9146 | 0.8862 | 0.9146 | 0.9187 | 0.8862 | 0.9350 | 0.8984 |
| Prueba 2 | 0.8780 | 0.9065 | 0.9268 | 0.8902 | 0.9024 | 0.9065 | 0.9512 |
| Prueba 3 | 0.8902 | 0.9309 | 0.9309 | 0.9106 | 0.9390 | 0.9309 | 0.9390 |
| Prueba 4 | 0.9146 | 0.9228 | 0.9431 | 0.9512 | 0.9309 | 0.9431 | 0.9268 |
| Prueba 5 | 0.9146 | 0.9187 | 0.9146 | 0.9350 | 0.9431 | 0.9268 | 0.9350 |
| Prueba 6 | 0.9268 | 0.9390 | 0.9390 | 0.8943 | 0.9268 | 0.9228 | 0.9350 |
| Prueba 7 | 0.9268 | 0.9146 | 0.9268 | 0.9472 | 0.9268 | 0.9431 | 0.9146 |

Cuadro 5.12: Tasa de acierto del grupo de validación para el modelo 03.

| Época | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|----------|--------|--------|--------|--------|--------|--------|--------|--------|--------|
| Prueba 1 | 0.5000 | 0.5000 | 0.5714 | 0.5714 | 0.5000 | 0.7857 | 0.7857 | 0.8214 | 0.8571 |
| Prueba 2 | 0.5000 | 0.6071 | 0.7500 | 0.6429 | 0.7857 | 0.7857 | 0.7500 | 0.7857 | 0.7500 |
| Prueba 3 | 0.6071 | 0.5000 | 0.6071 | 0.6786 | 0.7500 | 0.8214 | 0.8214 | 0.7500 | 0.8929 |
| Prueba 4 | 0.3571 | 0.3929 | 0.7143 | 0.7143 | 0.8571 | 0.8929 | 0.8929 | 0.8929 | 0.8929 |
| Prueba 5 | 0.6071 | 0.7500 | 0.8571 | 0.6429 | 0.8929 | 0.8571 | 0.8571 | 0.8929 | 0.8214 |
| Prueba 6 | 0.3929 | 0.7143 | 0.5714 | 0.7143 | 0.6786 | 0.8214 | 0.8214 | 0.8571 | 0.8214 |
| Prueba 7 | 0.6071 | 0.6071 | 0.8214 | 0.8571 | 0.7143 | 0.8571 | 0.8929 | 0.8214 | 0.7857 |

| Época | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 |
|----------|--------|--------|--------|--------|--------|--------|--------|--------|--------|
| Prueba 1 | 0.8214 | 0.8571 | 0.8214 | 0.8929 | 0.8929 | 0.8571 | 0.8571 | 0.8929 | 0.8929 |
| Prueba 2 | 0.8214 | 0.8214 | 0.7857 | 0.7857 | 0.7857 | 0.8571 | 0.8929 | 0.8571 | 0.8571 |
| Prueba 3 | 0.7857 | 0.8571 | 0.7857 | 0.8214 | 0.8571 | 0.8214 | 0.8571 | 0.7857 | 0.8571 |
| Prueba 4 | 0.8571 | 0.8571 | 0.8929 | 0.8571 | 0.8571 | 0.8929 | 0.8571 | 0.8571 | 0.8214 |
| Prueba 5 | 0.8571 | 0.8929 | 0.8571 | 0.8929 | 0.8571 | 0.8571 | 0.8929 | 0.8929 | 0.8929 |
| Prueba 6 | 0.8571 | 0.8214 | 0.8929 | 0.8929 | 0.8929 | 0.8571 | 0.8929 | 0.8929 | 0.8571 |
| Prueba 7 | 0.8214 | 0.8571 | 0.8571 | 0.8571 | 0.8929 | 0.8929 | 0.8929 | 0.8929 | 0.8929 |

| Época | 19 | 20 | 21 | 22 | 23 | 24 | 25 |
|----------|--------|--------|--------|--------|--------|--------|--------|
| Prueba 1 | 0.8571 | 0.8929 | 0.8929 | 0.8929 | 0.8929 | 0.8571 | 0.8929 |
| Prueba 2 | 0.8214 | 0.8214 | 0.8214 | 0.8214 | 0.7857 | 0.8214 | 0.8571 |
| Prueba 3 | 0.8571 | 0.8571 | 0.8214 | 0.8214 | 0.8214 | 0.8214 | 0.8571 |
| Prueba 4 | 0.8571 | 0.8571 | 0.8571 | 0.8571 | 0.8214 | 0.8571 | 0.8571 |
| Prueba 5 | 0.8571 | 0.8929 | 0.8929 | 0.8929 | 0.8929 | 0.8214 | 0.8929 |
| Prueba 6 | 0.8929 | 0.8571 | 0.8571 | 0.8929 | 0.8929 | 0.8571 | 0.8571 |
| Prueba 7 | 0.8214 | 0.8571 | 0.8571 | 0.8571 | 0.8214 | 0.8571 | 0.8214 |

Cuadro 5.13: Pérdida del grupo de entrenamiento para el modelo 03.

| Época | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|----------|---------|--------|--------|--------|--------|--------|--------|--------|--------|
| Prueba 1 | 8.3313 | 1.0054 | 0.9960 | 0.9353 | 0.8615 | 0.8395 | 0.7210 | 0.6645 | 0.6233 |
| Prueba 2 | 4.2456 | 0.9776 | 0.8531 | 0.7431 | 0.6877 | 0.6003 | 0.5496 | 0.4640 | 0.4154 |
| Prueba 3 | 14.0685 | 0.9531 | 0.9361 | 0.8365 | 0.7311 | 0.5761 | 0.5238 | 0.4699 | 0.3913 |
| Prueba 4 | 5.9437 | 1.0286 | 0.9324 | 0.7755 | 0.6695 | 0.5461 | 0.4244 | 0.3253 | 0.3005 |
| Prueba 5 | 3.1245 | 0.8655 | 0.7929 | 0.6384 | 0.5708 | 0.4603 | 0.4304 | 0.3057 | 0.3173 |
| Prueba 6 | 7.2261 | 0.9485 | 0.8996 | 0.7868 | 0.6683 | 0.5803 | 0.5368 | 0.4504 | 0.3991 |
| Prueba 7 | 5.4827 | 0.8716 | 0.7932 | 0.6414 | 0.4806 | 0.4031 | 0.3494 | 0.3214 | 0.2684 |

| Época | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 |
|----------|--------|--------|--------|--------|--------|--------|--------|--------|--------|
| Prueba 1 | 0.5547 | 0.4948 | 0.4138 | 0.3806 | 0.3285 | 0.3190 | 0.2931 | 0.2661 | 0.2520 |
| Prueba 2 | 0.4523 | 0.3540 | 0.3177 | 0.3154 | 0.2977 | 0.2608 | 0.2346 | 0.2351 | 0.2387 |
| Prueba 3 | 0.3257 | 0.3077 | 0.2668 | 0.2767 | 0.2460 | 0.2565 | 0.2241 | 0.2526 | 0.2057 |
| Prueba 4 | 0.2647 | 0.2693 | 0.2233 | 0.2521 | 0.2235 | 0.1951 | 0.2338 | 0.2099 | 0.1751 |
| Prueba 5 | 0.2804 | 0.2589 | 0.2191 | 0.2782 | 0.2357 | 0.2156 | 0.1800 | 0.2040 | 0.1791 |
| Prueba 6 | 0.3505 | 0.3199 | 0.2744 | 0.2490 | 0.2342 | 0.2007 | 0.2441 | 0.1862 | 0.1829 |
| Prueba 7 | 0.2439 | 0.2022 | 0.2179 | 0.2343 | 0.1704 | 0.1878 | 0.1656 | 0.1810 | 0.1746 |

| Época | 19 | 20 | 21 | 22 | 23 | 24 | 25 |
|----------|--------|--------|--------|--------|--------|--------|--------|
| Prueba 1 | 0.2070 | 0.2660 | 0.2242 | 0.2223 | 0.2206 | 0.2154 | 0.2278 |
| Prueba 2 | 0.2508 | 0.2280 | 0.2069 | 0.2176 | 0.2382 | 0.2150 | 0.1820 |
| Prueba 3 | 0.2069 | 0.1628 | 0.1767 | 0.1870 | 0.1694 | 0.1326 | 0.1424 |
| Prueba 4 | 0.1803 | 0.1609 | 0.1435 | 0.1294 | 0.1571 | 0.1503 | 0.1310 |
| Prueba 5 | 0.1971 | 0.1837 | 0.1747 | 0.1497 | 0.1417 | 0.1333 | 0.1261 |
| Prueba 6 | 0.1912 | 0.1557 | 0.1645 | 0.2281 | 0.1934 | 0.1939 | 0.1534 |
| Prueba 7 | 0.1519 | 0.1659 | 0.1606 | 0.1310 | 0.1320 | 0.1240 | 0.1488 |

Cuadro 5.14: Pérdida del grupo de validación para el modelo 03.

| Época | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|----------|--------|--------|--------|--------|--------|--------|--------|--------|--------|
| Prueba 1 | 1.0516 | 0.9356 | 0.9006 | 0.8035 | 0.7663 | 0.7332 | 0.5956 | 0.6310 | 0.4748 |
| Prueba 2 | 0.9891 | 0.8780 | 0.7842 | 0.7222 | 0.5986 | 0.5395 | 0.5439 | 0.5248 | 0.6281 |
| Prueba 3 | 0.9276 | 0.9601 | 0.8503 | 0.7361 | 0.5767 | 0.5102 | 0.4980 | 0.5635 | 0.4489 |
| Prueba 4 | 1.1164 | 1.0631 | 0.7445 | 0.6282 | 0.4732 | 0.3959 | 0.3389 | 0.3236 | 0.3771 |
| Prueba 5 | 0.8727 | 0.8142 | 0.5909 | 0.6301 | 0.4382 | 0.4532 | 0.4310 | 0.3578 | 0.3969 |
| Prueba 6 | 1.1818 | 0.9013 | 0.7833 | 0.7110 | 0.5514 | 0.5239 | 0.4648 | 0.4651 | 0.4549 |
| Prueba 7 | 0.9451 | 0.8825 | 0.5813 | 0.4953 | 0.5108 | 0.4367 | 0.3990 | 0.4945 | 0.4630 |

| Época | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 |
|----------|--------|--------|--------|--------|--------|--------|--------|--------|--------|
| Prueba 1 | 0.5101 | 0.4268 | 0.4549 | 0.4357 | 0.4258 | 0.4002 | 0.3709 | 0.3531 | 0.3793 |
| Prueba 2 | 0.5026 | 0.5118 | 0.5596 | 0.5374 | 0.5365 | 0.5426 | 0.5871 | 0.5641 | 0.4840 |
| Prueba 3 | 0.5140 | 0.4471 | 0.5183 | 0.5479 | 0.5021 | 0.5166 | 0.5132 | 0.5607 | 0.4658 |
| Prueba 4 | 0.3776 | 0.3909 | 0.4262 | 0.3849 | 0.4167 | 0.3868 | 0.3409 | 0.3502 | 0.4061 |
| Prueba 5 | 0.3507 | 0.3917 | 0.4384 | 0.3850 | 0.4104 | 0.3929 | 0.3915 | 0.4141 | 0.3883 |
| Prueba 6 | 0.4353 | 0.5142 | 0.4885 | 0.5435 | 0.5448 | 0.5659 | 0.5124 | 0.5359 | 0.6246 |
| Prueba 7 | 0.4316 | 0.4969 | 0.4517 | 0.4224 | 0.3914 | 0.4653 | 0.4953 | 0.4827 | 0.4624 |

| Época | 19 | 20 | 21 | 22 | 23 | 24 | 25 |
|----------|--------|--------|--------|--------|--------|--------|--------|
| Prueba 1 | 0.3942 | 0.3940 | 0.3899 | 0.3827 | 0.4056 | 0.4099 | 0.3690 |
| Prueba 2 | 0.5264 | 0.5347 | 0.5540 | 0.6344 | 0.5979 | 0.5727 | 0.6324 |
| Prueba 3 | 0.4724 | 0.5420 | 0.5610 | 0.5415 | 0.5063 | 0.5485 | 0.5956 |
| Prueba 4 | 0.4007 | 0.4573 | 0.4281 | 0.4241 | 0.4334 | 0.3706 | 0.4002 |
| Prueba 5 | 0.3644 | 0.3567 | 0.3900 | 0.3868 | 0.4294 | 0.4558 | 0.4424 |
| Prueba 6 | 0.5932 | 0.6455 | 0.6470 | 0.5157 | 0.4843 | 0.5986 | 0.6138 |
| Prueba 7 | 0.5606 | 0.5203 | 0.5197 | 0.5096 | 0.4964 | 0.5102 | 0.5245 |

Cuadro 5.15: Tiempo de entrenamiento por época para el modelo 03 en segundos.

| Época | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|----------|---------|---------|---------|---------|---------|---------|---------|---------|---------|
| Prueba 1 | 27.9819 | 27.1325 | 26.6022 | 26.8327 | 26.8169 | 26.7179 | 26.5984 | 26.8504 | 27.0106 |
| Prueba 2 | 27.8495 | 27.1536 | 27.2289 | 27.0594 | 27.1576 | 27.2165 | 26.8720 | 27.0924 | 27.0316 |
| Prueba 3 | 30.6327 | 27.9969 | 28.5926 | 28.1032 | 28.0624 | 28.2303 | 28.0209 | 28.2965 | 27.9678 |
| Prueba 4 | 30.8979 | 28.5351 | 28.2455 | 28.5874 | 28.0474 | 28.3063 | 28.4032 | 28.4817 | 28.2150 |
| Prueba 5 | 33.9763 | 29.8468 | 29.8294 | 29.9367 | 29.8145 | 30.5099 | 29.4702 | 29.5563 | 29.5799 |
| Prueba 6 | 33.2899 | 29.7416 | 30.5861 | 29.8370 | 29.8381 | 29.8289 | 29.9052 | 29.9326 | 29.9145 |
| Prueba 7 | 33.0947 | 30.1097 | 29.6809 | 29.8576 | 30.1877 | 30.0492 | 29.9005 | 29.9893 | 29.7388 |

| Época | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 |
|----------|---------|---------|---------|-----------|---------|---------|---------|---------|---------|
| Prueba 1 | 26.7355 | 26.7598 | 26.6067 | 26.7021 | 26.4778 | 26.6006 | 26.5868 | 26.7428 | 26.6510 |
| Prueba 2 | 27.3462 | 27.1554 | 27.2081 | 27.0727 | 27.4939 | 27.0556 | 26.8345 | 27.0477 | 27.4655 |
| Prueba 3 | 28.2533 | 28.0154 | 28.2828 | 27.9257 | 28.1084 | 28.5870 | 28.5053 | 28.2029 | 28.1694 |
| Prueba 4 | 28.4082 | 28.2683 | 29.0648 | 1681.9956 | 29.0119 | 29.5819 | 29.3334 | 29.3843 | 29.3270 |
| Prueba 5 | 29.4625 | 29.4496 | 30.2159 | 29.5296 | 29.6378 | 29.4513 | 29.4218 | 29.5298 | 29.5327 |
| Prueba 6 | 29.7294 | 29.7364 | 29.9353 | 29.3180 | 29.2373 | 29.2174 | 29.3005 | 29.2687 | 29.7843 |
| Prueba 7 | 29.9986 | 29.8721 | 29.8483 | 30.3127 | 29.9534 | 30.6387 | 30.4604 | 30.6477 | 30.9470 |

| Época | 19 | 20 | 21 | 22 | 23 | 24 | 25 |
|----------|---------|-----------|---------|---------|---------|---------|---------|
| Prueba 1 | 26.4293 | 26.6399 | 26.5583 | 26.5975 | 26.6181 | 26.7030 | 25.8998 |
| Prueba 2 | 25.7075 | 2652.5776 | 27.9570 | 27.5066 | 27.5326 | 27.7458 | 26.7275 |
| Prueba 3 | 28.1101 | 28.5789 | 28.1655 | 28.0465 | 28.1940 | 28.1623 | 28.0239 |
| Prueba 4 | 29.5688 | 29.4554 | 29.5127 | 29.4356 | 29.4510 | 30.4287 | 29.2420 |
| Prueba 5 | 29.5494 | 29.4439 | 29.5790 | 29.5697 | 29.3131 | 29.5429 | 29.1974 |
| Prueba 6 | 28.9510 | 29.2895 | 29.3282 | 29.0043 | 29.9134 | 29.6404 | 29.0605 |
| Prueba 7 | 30.1325 | 30.1184 | 31.9687 | 31.0380 | 30.9983 | 30.8566 | 30.5050 |

5.4. Resultados 04

Se resaltan en azul los valores máximos en las tablas de tasa de acierto y los mínimos en las tablas de perdida.

Cuadro 5.16: Tasa de acierto del grupo de entrenamiento para el modelo 04.

| Época | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|----------|--------|--------|--------|--------|--------|--------|--------|--------|--------|
| Prueba 1 | 0.4512 | 0.4878 | 0.5163 | 0.6016 | 0.6789 | 0.6992 | 0.7642 | 0.8293 | 0.8577 |
| Prueba 2 | 0.3943 | 0.4959 | 0.5122 | 0.5244 | 0.5813 | 0.6382 | 0.6626 | 0.7276 | 0.7236 |
| Prueba 3 | 0.4675 | 0.5000 | 0.6951 | 0.7317 | 0.7764 | 0.8618 | 0.8862 | 0.8659 | 0.9228 |
| Prueba 4 | 0.4024 | 0.4959 | 0.4959 | 0.5732 | 0.6382 | 0.6545 | 0.7033 | 0.7683 | 0.7846 |
| Prueba 5 | 0.4431 | 0.4919 | 0.6341 | 0.6504 | 0.7439 | 0.7846 | 0.8089 | 0.8293 | 0.8293 |
| Prueba 6 | 0.4146 | 0.4024 | 0.6057 | 0.7114 | 0.7398 | 0.8089 | 0.8740 | 0.8780 | 0.8780 |
| Prueba 7 | 0.4634 | 0.4593 | 0.5854 | 0.6098 | 0.5650 | 0.6870 | 0.6829 | 0.7480 | 0.7967 |

| Época | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 |
|----------|--------|--------|--------|--------|--------|--------|--------|--------|--------|
| Prueba 1 | 0.8211 | 0.8821 | 0.8740 | 0.9065 | 0.9065 | 0.9146 | 0.9390 | 0.9309 | 0.9309 |
| Prueba 2 | 0.7276 | 0.7642 | 0.7886 | 0.7764 | 0.7886 | 0.8089 | 0.8008 | 0.8211 | 0.8374 |
| Prueba 3 | 0.8943 | 0.9187 | 0.9268 | 0.9268 | 0.9228 | 0.9350 | 0.9350 | 0.9431 | 0.9187 |
| Prueba 4 | 0.8455 | 0.8008 | 0.8780 | 0.8943 | 0.8618 | 0.9187 | 0.9024 | 0.9350 | 0.9146 |
| Prueba 5 | 0.8740 | 0.8537 | 0.9065 | 0.8821 | 0.9268 | 0.8902 | 0.9309 | 0.9187 | 0.9187 |
| Prueba 6 | 0.9146 | 0.9431 | 0.9350 | 0.9187 | 0.9390 | 0.9390 | 0.9431 | 0.9472 | 0.9512 |
| Prueba 7 | 0.8374 | 0.8171 | 0.8455 | 0.8740 | 0.8821 | 0.9106 | 0.8902 | 0.9065 | 0.9146 |

| Época | 19 | 20 | 21 | 22 | 23 | 24 | 25 |
|----------|--------|--------|--------|--------|--------|--------|--------|
| Prueba 1 | 0.9309 | 0.9309 | 0.9146 | 0.9512 | 0.9390 | 0.9350 | 0.9431 |
| Prueba 2 | 0.8333 | 0.8374 | 0.8659 | 0.8699 | 0.8699 | 0.8943 | 0.8821 |
| Prueba 3 | 0.9472 | 0.9390 | 0.9350 | 0.9553 | 0.9390 | 0.9431 | 0.9593 |
| Prueba 4 | 0.9228 | 0.9268 | 0.9390 | 0.9512 | 0.9431 | 0.9309 | 0.9350 |
| Prueba 5 | 0.9268 | 0.9146 | 0.9431 | 0.9472 | 0.9350 | 0.9309 | 0.9390 |
| Prueba 6 | 0.9431 | 0.9512 | 0.9634 | 0.9593 | 0.9512 | 0.9512 | 0.9512 |
| Prueba 7 | 0.9146 | 0.9024 | 0.9146 | 0.9106 | 0.9268 | 0.9390 | 0.9268 |

Cuadro 5.17: Tasa de acierto del grupo de validación para el modelo 04.

| Época | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|----------|--------|--------|--------|--------|--------|--------|--------|--------|--------|
| Prueba 1 | 0.5000 | 0.5000 | 0.6071 | 0.5714 | 0.7500 | 0.7500 | 0.8929 | 0.9286 | 0.7857 |
| Prueba 2 | 0.5000 | 0.5000 | 0.5000 | 0.5714 | 0.5714 | 0.5357 | 0.5714 | 0.5714 | 0.6071 |
| Prueba 3 | 0.5000 | 0.5000 | 0.5714 | 0.6786 | 0.8929 | 0.7857 | 0.8571 | 0.8214 | 0.8929 |
| Prueba 4 | 0.5000 | 0.5000 | 0.5000 | 0.5357 | 0.6071 | 0.6071 | 0.7143 | 0.7143 | 0.8929 |
| Prueba 5 | 0.5357 | 0.3929 | 0.5357 | 0.5357 | 0.5714 | 0.7857 | 0.8571 | 0.8929 | 0.8929 |
| Prueba 6 | 0.1429 | 0.5000 | 0.5357 | 0.7500 | 0.7143 | 0.7857 | 0.9286 | 0.8929 | 0.9286 |
| Prueba 7 | 0.5000 | 0.5000 | 0.5357 | 0.5714 | 0.5000 | 0.7143 | 0.8571 | 0.6429 | 0.8929 |

| Época | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 |
|----------|--------|--------|--------|--------|--------|--------|--------|--------|--------|
| Prueba 1 | 0.8571 | 0.8929 | 0.8929 | 0.8929 | 0.8571 | 0.8929 | 0.8929 | 0.8929 | 0.8571 |
| Prueba 2 | 0.6071 | 0.7143 | 0.6071 | 0.7500 | 0.7500 | 0.7500 | 0.7500 | 0.7857 | 0.7500 |
| Prueba 3 | 0.8214 | 0.8571 | 0.8929 | 0.8929 | 0.8929 | 0.8929 | 0.8929 | 0.8571 | 0.8929 |
| Prueba 4 | 0.9286 | 0.8571 | 0.8571 | 0.8214 | 0.8571 | 0.8929 | 0.8571 | 0.8929 | 0.8929 |
| Prueba 5 | 0.8929 | 0.7857 | 0.8929 | 0.8929 | 0.8929 | 0.8929 | 0.8571 | 0.8929 | 0.8929 |
| Prueba 6 | 0.7500 | 0.8571 | 0.8929 | 0.8929 | 0.8929 | 0.8571 | 0.8929 | 0.8929 | 0.8929 |
| Prueba 7 | 0.7500 | 0.8214 | 0.8929 | 0.8929 | 0.8571 | 0.8214 | 0.8214 | 0.8929 | 0.8929 |

| Época | 19 | 20 | 21 | 22 | 23 | 24 | 25 |
|----------|--------|--------|--------|--------|--------|--------|--------|
| Prueba 1 | 0.8929 | 0.8929 | 0.8929 | 0.8929 | 0.8571 | 0.8929 | 0.8571 |
| Prueba 2 | 0.7857 | 0.8214 | 0.8571 | 0.8214 | 0.7857 | 0.8929 | 0.8571 |
| Prueba 3 | 0.8929 | 0.8571 | 0.8929 | 0.8929 | 0.8929 | 0.8929 | 0.8929 |
| Prueba 4 | 0.8929 | 0.8929 | 0.8571 | 0.8571 | 0.8929 | 0.8571 | 0.8929 |
| Prueba 5 | 0.8929 | 0.8929 | 0.8929 | 0.8929 | 0.8929 | 0.8929 | 0.8571 |
| Prueba 6 | 0.8929 | 0.8929 | 0.8929 | 0.8929 | 0.8929 | 0.8929 | 0.8929 |
| Prueba 7 | 0.7857 | 0.8929 | 0.8571 | 0.8929 | 0.8929 | 0.8929 | 0.8214 |

Cuadro 5.18: pérdida del grupo de entrenamiento para el modelo 04.

| Época | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|----------|---------|--------|--------|--------|--------|--------|--------|--------|--------|
| Prueba 1 | 12.5079 | 1.3078 | 0.9544 | 0.8556 | 0.7555 | 0.6847 | 0.5677 | 0.4489 | 0.3679 |
| Prueba 2 | 11.6825 | 1.0609 | 0.9349 | 0.8809 | 0.8084 | 0.7811 | 0.7301 | 0.6983 | 0.6810 |
| Prueba 3 | 17.4085 | 1.1196 | 0.7831 | 0.6479 | 0.4730 | 0.3548 | 0.3232 | 0.2962 | 0.2452 |
| Prueba 4 | 6.4993 | 1.0033 | 0.9441 | 0.8966 | 0.8101 | 0.6912 | 0.6109 | 0.4874 | 0.4794 |
| Prueba 5 | 12.4870 | 1.0250 | 0.8798 | 0.7647 | 0.6569 | 0.5547 | 0.4762 | 0.4270 | 0.3775 |
| Prueba 6 | 12.6688 | 1.0237 | 0.8643 | 0.7258 | 0.6419 | 0.5360 | 0.3815 | 0.3068 | 0.2757 |
| Prueba 7 | 7.2449 | 1.1688 | 0.9067 | 0.8266 | 0.8209 | 0.7230 | 0.6703 | 0.6221 | 0.5369 |

| Época | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 |
|----------|--------|--------|--------|--------|--------|--------|--------|--------|--------|
| Prueba 1 | 0.3826 | 0.2754 | 0.3034 | 0.2633 | 0.2227 | 0.2082 | 0.1732 | 0.1739 | 0.1560 |
| Prueba 2 | 0.6053 | 0.5502 | 0.5282 | 0.4949 | 0.4540 | 0.4257 | 0.4299 | 0.4163 | 0.3510 |
| Prueba 3 | 0.2505 | 0.2215 | 0.1949 | 0.1628 | 0.1687 | 0.1442 | 0.1608 | 0.1633 | 0.1556 |
| Prueba 4 | 0.3811 | 0.3795 | 0.2945 | 0.2683 | 0.2661 | 0.2175 | 0.2041 | 0.1799 | 0.1974 |
| Prueba 5 | 0.3591 | 0.3223 | 0.2631 | 0.2717 | 0.2160 | 0.2556 | 0.2279 | 0.2015 | 0.1943 |
| Prueba 6 | 0.2287 | 0.1961 | 0.1976 | 0.1801 | 0.1956 | 0.1742 | 0.1674 | 0.1403 | 0.1142 |
| Prueba 7 | 0.4386 | 0.4213 | 0.3517 | 0.3208 | 0.3021 | 0.2205 | 0.2309 | 0.2172 | 0.2006 |

| Época | 19 | 20 | 21 | 22 | 23 | 24 | 25 |
|----------|--------|--------|--------|--------|--------|--------|--------|
| Prueba 1 | 0.1827 | 0.1833 | 0.1813 | 0.1280 | 0.1348 | 0.1639 | 0.1252 |
| Prueba 2 | 0.3728 | 0.3216 | 0.2960 | 0.3313 | 0.2950 | 0.2643 | 0.2746 |
| Prueba 3 | 0.1322 | 0.1279 | 0.1338 | 0.1222 | 0.1227 | 0.1053 | 0.1180 |
| Prueba 4 | 0.1901 | 0.1568 | 0.1410 | 0.1350 | 0.1395 | 0.1289 | 0.1176 |
| Prueba 5 | 0.1656 | 0.1685 | 0.1700 | 0.1557 | 0.1586 | 0.1558 | 0.1483 |
| Prueba 6 | 0.1269 | 0.1119 | 0.1043 | 0.1210 | 0.1236 | 0.1208 | 0.0967 |
| Prueba 7 | 0.1714 | 0.1965 | 0.1889 | 0.1661 | 0.1570 | 0.1667 | 0.1498 |

Cuadro 5.19: pérdida del grupo de validación para el modelo 04.

| Época | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|----------|--------|--------|--------|--------|--------|--------|--------|--------|--------|
| Prueba 1 | 1.8330 | 1.0581 | 0.8692 | 0.7887 | 0.7026 | 0.5774 | 0.5078 | 0.4207 | 0.4764 |
| Prueba 2 | 1.1540 | 0.9502 | 0.8750 | 0.8188 | 0.8120 | 0.7179 | 0.6861 | 0.6641 | 0.6171 |
| Prueba 3 | 1.2427 | 0.8543 | 0.7322 | 0.5777 | 0.5105 | 0.5242 | 0.5118 | 0.4809 | 0.5176 |
| Prueba 4 | 0.9517 | 0.9385 | 0.8922 | 0.8708 | 0.7152 | 0.6662 | 0.5645 | 0.4883 | 0.4178 |
| Prueba 5 | 0.9143 | 0.9290 | 0.8355 | 0.7168 | 0.6453 | 0.5361 | 0.4954 | 0.4519 | 0.4618 |
| Prueba 6 | 1.1196 | 0.9634 | 0.8432 | 0.6549 | 0.5830 | 0.4845 | 0.3794 | 0.3890 | 0.3981 |
| Prueba 7 | 1.5642 | 0.9527 | 0.8602 | 0.8472 | 0.7829 | 0.6472 | 0.6086 | 0.5768 | 0.4801 |

| Época | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 |
|----------|--------|--------|--------|--------|--------|--------|--------|--------|--------|
| Prueba 1 | 0.4496 | 0.4588 | 0.3987 | 0.3888 | 0.3871 | 0.3956 | 0.4486 | 0.4130 | 0.4442 |
| Prueba 2 | 0.5775 | 0.5124 | 0.5833 | 0.5052 | 0.4298 | 0.4753 | 0.4619 | 0.4075 | 0.3713 |
| Prueba 3 | 0.4203 | 0.4694 | 0.5098 | 0.4827 | 0.5493 | 0.5047 | 0.5628 | 0.5840 | 0.5476 |
| Prueba 4 | 0.3772 | 0.3555 | 0.3774 | 0.4446 | 0.3854 | 0.3604 | 0.4138 | 0.3636 | 0.3674 |
| Prueba 5 | 0.4276 | 0.4581 | 0.4020 | 0.4427 | 0.4406 | 0.5467 | 0.4526 | 0.4991 | 0.5587 |
| Prueba 6 | 0.3931 | 0.3177 | 0.3552 | 0.3413 | 0.3380 | 0.4649 | 0.3164 | 0.3678 | 0.3373 |
| Prueba 7 | 0.4810 | 0.4254 | 0.4484 | 0.5431 | 0.4535 | 0.5086 | 0.5363 | 0.4873 | 0.5337 |

| Época | 19 | 20 | 21 | 22 | 23 | 24 | 25 |
|----------|--------|--------|--------|--------|--------|--------|--------|
| Prueba 1 | 0.3915 | 0.4121 | 0.4617 | 0.5188 | 0.5415 | 0.5202 | 0.4098 |
| Prueba 2 | 0.3951 | 0.3859 | 0.4280 | 0.4147 | 0.3986 | 0.3673 | 0.4093 |
| Prueba 3 | 0.5847 | 0.6464 | 0.5331 | 0.5963 | 0.5905 | 0.5946 | 0.6476 |
| Prueba 4 | 0.3367 | 0.4221 | 0.5008 | 0.4251 | 0.4310 | 0.4109 | 0.4597 |
| Prueba 5 | 0.5477 | 0.5877 | 0.5336 | 0.5164 | 0.5224 | 0.7410 | 0.6827 |
| Prueba 6 | 0.4056 | 0.3440 | 0.3639 | 0.3923 | 0.3823 | 0.4262 | 0.4871 |
| Prueba 7 | 0.5608 | 0.6039 | 0.5128 | 0.4599 | 0.5633 | 0.5974 | 0.5255 |

Cuadro 5.20: Tiempo de entrenamiento por época para el modelo 04 en segundos.

| Época | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|----------|---------|---------|---------|---------|---------|---------|---------|---------|---------|
| Prueba 1 | 50.2675 | 40.8064 | 39.6241 | 40.9467 | 39.1497 | 38.2428 | 37.9803 | 39.4260 | 38.2507 |
| Prueba 2 | 52.2943 | 41.1532 | 41.4728 | 41.5566 | 41.5941 | 41.6929 | 42.0499 | 41.5630 | 41.4402 |
| Prueba 3 | 52.5813 | 41.1287 | 41.1251 | 40.8453 | 41.3952 | 43.6097 | 43.3231 | 41.5502 | 41.2113 |
| Prueba 4 | 51.5154 | 42.9207 | 41.5721 | 40.9425 | 42.3062 | 41.4084 | 42.0229 | 40.8589 | 40.4862 |
| Prueba 5 | 50.7506 | 41.9294 | 41.1781 | 41.1740 | 41.6597 | 41.5382 | 42.0200 | 41.8821 | 42.4164 |
| Prueba 6 | 51.0798 | 42.5448 | 41.5540 | 41.6229 | 41.9744 | 41.1181 | 40.8376 | 40.9248 | 40.8049 |
| Prueba 7 | 52.5458 | 42.9718 | 40.7053 | 41.1852 | 42.1442 | 42.1228 | 41.2773 | 41.6452 | 41.8317 |

| Época | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 |
|----------|---------|---------|---------|---------|---------|---------|---------|---------|---------|
| Prueba 1 | 38.3097 | 38.7796 | 38.0933 | 38.3027 | 39.1355 | 39.5271 | 41.0564 | 39.9728 | 40.3102 |
| Prueba 2 | 40.8008 | 41.4768 | 42.2821 | 41.4172 | 40.4074 | 41.7518 | 40.8051 | 40.5327 | 41.8214 |
| Prueba 3 | 41.2610 | 41.3747 | 41.1706 | 40.9429 | 41.0081 | 41.4840 | 41.4218 | 40.7354 | 40.4612 |
| Prueba 4 | 40.9443 | 40.5595 | 40.6416 | 40.7289 | 41.0268 | 41.0305 | 40.9717 | 40.6121 | 40.4876 |
| Prueba 5 | 41.7402 | 42.2911 | 42.1026 | 41.4877 | 41.0152 | 40.9570 | 42.5739 | 41.2563 | 41.6744 |
| Prueba 6 | 39.2069 | 38.1284 | 38.2746 | 38.2568 | 38.6036 | 39.3797 | 38.2258 | 39.5037 | 38.3709 |
| Prueba 7 | 40.8997 | 40.9019 | 40.8340 | 40.4167 | 40.2538 | 40.4267 | 39.0870 | 40.0741 | 38.7372 |

| Época | 19 | 20 | 21 | 22 | 23 | 24 | 25 |
|----------|---------|---------|---------|---------|---------|---------|---------|
| Prueba 1 | 40.1960 | 40.6049 | 40.3691 | 43.5361 | 41.6280 | 40.7186 | 39.9326 |
| Prueba 2 | 41.3853 | 41.2632 | 41.3336 | 42.2248 | 42.1328 | 42.3362 | 41.7377 |
| Prueba 3 | 40.9694 | 40.8116 | 41.0042 | 40.5736 | 40.9441 | 40.7608 | 40.9816 |
| Prueba 4 | 41.1523 | 40.5552 | 40.6313 | 41.0053 | 40.6405 | 40.8149 | 40.5847 |
| Prueba 5 | 41.6391 | 41.0281 | 41.4060 | 42.1677 | 41.9481 | 42.4044 | 41.8633 |
| Prueba 6 | 39.6271 | 41.5348 | 41.0200 | 40.4498 | 40.8512 | 41.2772 | 41.6770 |
| Prueba 7 | 37.9719 | 38.7900 | 39.2942 | 38.0678 | 37.8672 | 38.5282 | 38.0598 |

5.5. Resultados 05

Se resaltan en azul los valores máximos en las tablas de tasa de acierto y los mínimos en las tablas de pérdida.

Cuadro 5.21: Tasa de acierto del grupo de entrenamiento para el modelo 05.

| Época | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|----------|--------|--------|--------|--------|--------|--------|--------|--------|--------|
| Prueba 1 | 0.4715 | 0.4593 | 0.6626 | 0.6870 | 0.7398 | 0.7846 | 0.8008 | 0.8293 | 0.8577 |
| Prueba 2 | 0.4634 | 0.4593 | 0.6707 | 0.7276 | 0.7724 | 0.8130 | 0.7967 | 0.8252 | 0.7846 |
| Prueba 3 | 0.4431 | 0.4472 | 0.5610 | 0.6138 | 0.6789 | 0.6463 | 0.7398 | 0.7439 | 0.7642 |
| Prueba 4 | 0.4837 | 0.5163 | 0.6382 | 0.6545 | 0.6951 | 0.7195 | 0.8008 | 0.8089 | 0.8618 |
| Prueba 5 | 0.4228 | 0.4756 | 0.6138 | 0.6829 | 0.6911 | 0.6951 | 0.7642 | 0.7967 | 0.7846 |
| Prueba 6 | 0.4634 | 0.5163 | 0.5894 | 0.6951 | 0.7520 | 0.7886 | 0.8171 | 0.8415 | 0.8211 |
| Prueba 7 | 0.4593 | 0.4878 | 0.5732 | 0.6301 | 0.6057 | 0.6585 | 0.7114 | 0.7317 | 0.7642 |

| Época | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 |
|----------|--------|--------|--------|--------|--------|--------|--------|--------|--------|
| Prueba 1 | 0.8862 | 0.8699 | 0.8862 | 0.8862 | 0.8943 | 0.9024 | 0.9228 | 0.9268 | 0.9309 |
| Prueba 2 | 0.8618 | 0.8699 | 0.9024 | 0.9106 | 0.9106 | 0.9065 | 0.8984 | 0.8862 | 0.9106 |
| Prueba 3 | 0.7846 | 0.8008 | 0.8455 | 0.8496 | 0.8537 | 0.8943 | 0.8984 | 0.8577 | 0.8780 |
| Prueba 4 | 0.8252 | 0.8415 | 0.9024 | 0.8902 | 0.8943 | 0.9106 | 0.8821 | 0.9268 | 0.9268 |
| Prueba 5 | 0.8496 | 0.8780 | 0.8821 | 0.8699 | 0.8821 | 0.8618 | 0.8780 | 0.8821 | 0.8984 |
| Prueba 6 | 0.8537 | 0.8902 | 0.9024 | 0.8984 | 0.8659 | 0.8780 | 0.8740 | 0.8984 | 0.9350 |
| Prueba 7 | 0.8211 | 0.8333 | 0.8374 | 0.8496 | 0.8659 | 0.8577 | 0.8862 | 0.8984 | 0.9024 |

| Época | 19 | 20 | 21 | 22 | 23 | 24 | 25 |
|----------|--------|--------|--------|--------|--------|--------|--------|
| Prueba 1 | 0.9350 | 0.9228 | 0.9268 | 0.9228 | 0.9024 | 0.9309 | 0.9309 |
| Prueba 2 | 0.8862 | 0.9431 | 0.9268 | 0.9228 | 0.9309 | 0.9106 | 0.9350 |
| Prueba 3 | 0.8862 | 0.8780 | 0.8862 | 0.8862 | 0.8943 | 0.9106 | 0.9106 |
| Prueba 4 | 0.9024 | 0.9146 | 0.9187 | 0.9187 | 0.9309 | 0.9024 | 0.9390 |
| Prueba 5 | 0.9228 | 0.8943 | 0.9065 | 0.9024 | 0.8943 | 0.9024 | 0.9187 |
| Prueba 6 | 0.9309 | 0.9106 | 0.9228 | 0.9106 | 0.9268 | 0.9268 | 0.9309 |
| Prueba 7 | 0.8943 | 0.9024 | 0.9146 | 0.9187 | 0.8984 | 0.9146 | 0.9146 |

Cuadro 5.22: Tasa de acierto del grupo de validación para el modelo 05.

| Época | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|----------|--------|--------|--------|--------|--------|--------|--------|--------|--------|
| Prueba 1 | 0.3929 | 0.6429 | 0.8214 | 0.5000 | 0.8214 | 0.7857 | 0.8929 | 0.8571 | 0.8929 |
| Prueba 2 | 0.3929 | 0.7500 | 0.7857 | 0.8929 | 0.6429 | 0.8929 | 0.8214 | 0.7500 | 0.8214 |
| Prueba 3 | 0.3929 | 0.5357 | 0.8214 | 0.6786 | 0.7143 | 0.7500 | 0.7857 | 0.7857 | 0.7857 |
| Prueba 4 | 0.3929 | 0.6071 | 0.7857 | 0.7143 | 0.8929 | 0.8571 | 0.7500 | 0.8571 | 0.8571 |
| Prueba 5 | 0.4286 | 0.5000 | 0.7500 | 0.7500 | 0.6071 | 0.8214 | 0.7857 | 0.7857 | 0.8929 |
| Prueba 6 | 0.5714 | 0.5714 | 0.7857 | 0.7857 | 0.7857 | 0.8571 | 0.8214 | 0.8571 | 0.8929 |
| Prueba 7 | 0.2143 | 0.5000 | 0.5357 | 0.6071 | 0.8214 | 0.7143 | 0.7857 | 0.8571 | 0.6786 |

| Época | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 |
|----------|--------|--------|--------|--------|--------|--------|--------|--------|--------|
| Prueba 1 | 0.8571 | 0.8571 | 0.8571 | 0.8571 | 0.8929 | 0.8929 | 0.8929 | 0.8929 | 0.8929 |
| Prueba 2 | 0.8571 | 0.8571 | 0.8571 | 0.8214 | 0.8571 | 0.8929 | 0.7857 | 0.8571 | 0.8929 |
| Prueba 3 | 0.7857 | 0.8571 | 0.9286 | 0.8214 | 0.8571 | 0.8929 | 0.9286 | 0.8214 | 0.8929 |
| Prueba 4 | 0.8214 | 0.8571 | 0.8571 | 0.8571 | 0.8214 | 0.8214 | 0.8571 | 0.8571 | 0.8571 |
| Prueba 5 | 0.8214 | 0.9286 | 0.7500 | 0.8571 | 0.8571 | 0.8929 | 0.8929 | 0.8929 | 0.8571 |
| Prueba 6 | 0.8571 | 0.8571 | 0.8929 | 0.8571 | 0.8571 | 0.8571 | 0.8929 | 0.8571 | 0.8929 |
| Prueba 7 | 0.8571 | 0.8571 | 0.8214 | 0.8214 | 0.8571 | 0.8571 | 0.8571 | 0.8214 | 0.8571 |

| Época | 19 | 20 | 21 | 22 | 23 | 24 | 25 |
|----------|--------|--------|--------|--------|--------|--------|--------|
| Prueba 1 | 0.8929 | 0.8929 | 0.8929 | 0.8571 | 0.8571 | 0.8929 | 0.8929 |
| Prueba 2 | 0.8929 | 0.8571 | 0.8571 | 0.8214 | 0.8929 | 0.8214 | 0.8571 |
| Prueba 3 | 0.8571 | 0.8929 | 0.8929 | 0.8929 | 0.8929 | 0.8571 | 0.8929 |
| Prueba 4 | 0.7143 | 0.8571 | 0.8571 | 0.8571 | 0.8571 | 0.8571 | 0.8571 |
| Prueba 5 | 0.8571 | 0.8929 | 0.8571 | 0.8571 | 0.8571 | 0.8214 | 0.8571 |
| Prueba 6 | 0.8571 | 0.8929 | 0.8929 | 0.8929 | 0.8929 | 0.8929 | 0.8571 |
| Prueba 7 | 0.8929 | 0.8214 | 0.8571 | 0.8571 | 0.8929 | 0.8571 | 0.8929 |

Cuadro 5.23: pérdida del grupo de entrenamiento para el modelo 05.

| Época | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|----------|---------|--------|--------|--------|--------|--------|--------|--------|--------|
| Prueba 1 | 8.9748 | 1.0274 | 0.8243 | 0.7200 | 0.6314 | 0.5389 | 0.4845 | 0.4204 | 0.3293 |
| Prueba 2 | 6.0692 | 1.0286 | 0.8240 | 0.6479 | 0.6164 | 0.4728 | 0.4808 | 0.4526 | 0.4320 |
| Prueba 3 | 10.9817 | 1.0345 | 0.8946 | 0.7995 | 0.7939 | 0.7362 | 0.6369 | 0.5771 | 0.5336 |
| Prueba 4 | 5.5983 | 0.9474 | 0.9278 | 0.8422 | 0.7236 | 0.5876 | 0.5074 | 0.4775 | 0.4000 |
| Prueba 5 | 6.1015 | 0.9872 | 0.8759 | 0.8056 | 0.7049 | 0.6455 | 0.5660 | 0.4643 | 0.4467 |
| Prueba 6 | 7.7697 | 1.0587 | 0.8857 | 0.7308 | 0.6181 | 0.5217 | 0.4342 | 0.3935 | 0.4304 |
| Prueba 7 | 11.5653 | 0.9981 | 0.9618 | 0.8405 | 0.7850 | 0.7383 | 0.6471 | 0.5792 | 0.4977 |

| Época | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 |
|----------|--------|--------|--------|--------|--------|--------|--------|--------|--------|
| Prueba 1 | 0.2860 | 0.2687 | 0.2372 | 0.2616 | 0.2235 | 0.2040 | 0.1779 | 0.1966 | 0.1666 |
| Prueba 2 | 0.3752 | 0.3290 | 0.2797 | 0.2537 | 0.2289 | 0.2105 | 0.2207 | 0.2374 | 0.2085 |
| Prueba 3 | 0.4708 | 0.4308 | 0.3994 | 0.3866 | 0.3541 | 0.3020 | 0.2853 | 0.3226 | 0.2835 |
| Prueba 4 | 0.3886 | 0.3643 | 0.2653 | 0.2692 | 0.2742 | 0.2506 | 0.2704 | 0.2113 | 0.2007 |
| Prueba 5 | 0.3998 | 0.3550 | 0.2953 | 0.3030 | 0.3026 | 0.2864 | 0.2663 | 0.2801 | 0.2453 |
| Prueba 6 | 0.3643 | 0.3090 | 0.2835 | 0.2599 | 0.2994 | 0.2821 | 0.3001 | 0.2435 | 0.2025 |
| Prueba 7 | 0.4510 | 0.4433 | 0.3998 | 0.3917 | 0.3458 | 0.3473 | 0.3021 | 0.2803 | 0.2487 |

| Época | 19 | 20 | 21 | 22 | 23 | 24 | 25 |
|----------|--------|--------|--------|--------|--------|--------|--------|
| Prueba 1 | 0.1693 | 0.1762 | 0.1599 | 0.2018 | 0.1849 | 0.1881 | 0.1635 |
| Prueba 2 | 0.2238 | 0.1715 | 0.1638 | 0.1804 | 0.1758 | 0.1725 | 0.1624 |
| Prueba 3 | 0.2832 | 0.3230 | 0.2698 | 0.2713 | 0.2457 | 0.2197 | 0.2089 |
| Prueba 4 | 0.2321 | 0.2707 | 0.2338 | 0.2143 | 0.1801 | 0.2230 | 0.1632 |
| Prueba 5 | 0.2662 | 0.2575 | 0.2346 | 0.2348 | 0.2090 | 0.2271 | 0.2071 |
| Prueba 6 | 0.2003 | 0.1900 | 0.1984 | 0.1820 | 0.1829 | 0.1728 | 0.1839 |
| Prueba 7 | 0.2347 | 0.2272 | 0.2170 | 0.2151 | 0.2058 | 0.2221 | 0.2019 |

Cuadro 5.24: pérdida del grupo de validación para el modelo 05.

| Época | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|----------|--------|--------|--------|--------|--------|--------|--------|--------|--------|
| Prueba 1 | 1.1203 | 0.8099 | 0.7005 | 0.7634 | 0.5792 | 0.4252 | 0.4599 | 0.3771 | 0.3731 |
| Prueba 2 | 1.1129 | 0.8572 | 0.6901 | 0.6417 | 0.7685 | 0.4419 | 0.5727 | 0.4459 | 0.4377 |
| Prueba 3 | 1.0724 | 0.9215 | 0.8247 | 0.8120 | 0.6338 | 0.6298 | 0.5648 | 0.4971 | 0.4767 |
| Prueba 4 | 1.0261 | 0.8599 | 0.9381 | 0.7130 | 0.5773 | 0.5138 | 0.5279 | 0.5317 | 0.4837 |
| Prueba 5 | 0.9719 | 0.8960 | 0.7888 | 0.6845 | 0.6751 | 0.5671 | 0.4929 | 0.4847 | 0.4697 |
| Prueba 6 | 0.9098 | 0.8304 | 0.7091 | 0.6497 | 0.5487 | 0.4218 | 0.4835 | 0.3904 | 0.3965 |
| Prueba 7 | 1.1053 | 0.9702 | 0.8383 | 0.8373 | 0.6422 | 0.6385 | 0.6538 | 0.5754 | 0.6531 |

| Época | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 |
|----------|--------|--------|--------|--------|--------|--------|--------|--------|--------|
| Prueba 1 | 0.3819 | 0.4873 | 0.4429 | 0.4847 | 0.3776 | 0.4552 | 0.4443 | 0.4783 | 0.4576 |
| Prueba 2 | 0.3601 | 0.3454 | 0.3345 | 0.3125 | 0.2980 | 0.2967 | 0.3655 | 0.3266 | 0.2821 |
| Prueba 3 | 0.4187 | 0.4951 | 0.3983 | 0.4364 | 0.3853 | 0.3384 | 0.3465 | 0.4189 | 0.3684 |
| Prueba 4 | 0.5125 | 0.4749 | 0.4707 | 0.5305 | 0.5025 | 0.4753 | 0.4705 | 0.4679 | 0.5098 |
| Prueba 5 | 0.4220 | 0.3714 | 0.3820 | 0.3528 | 0.3750 | 0.3551 | 0.3853 | 0.3499 | 0.3907 |
| Prueba 6 | 0.3780 | 0.3668 | 0.3568 | 0.3860 | 0.3932 | 0.3374 | 0.3473 | 0.3826 | 0.3917 |
| Prueba 7 | 0.5666 | 0.4700 | 0.5477 | 0.5204 | 0.5617 | 0.5498 | 0.5949 | 0.5500 | 0.5723 |

| Época | 19 | 20 | 21 | 22 | 23 | 24 | 25 |
|----------|--------|--------|--------|--------|--------|--------|--------|
| Prueba 1 | 0.4640 | 0.4395 | 0.4760 | 0.4553 | 0.4563 | 0.4752 | 0.5105 |
| Prueba 2 | 0.3213 | 0.3237 | 0.3317 | 0.3218 | 0.3113 | 0.3188 | 0.3594 |
| Prueba 3 | 0.3711 | 0.3774 | 0.2909 | 0.2928 | 0.2868 | 0.3413 | 0.3178 |
| Prueba 4 | 0.5801 | 0.4452 | 0.4739 | 0.4879 | 0.4892 | 0.4750 | 0.4975 |
| Prueba 5 | 0.3804 | 0.3919 | 0.3630 | 0.3616 | 0.3817 | 0.4109 | 0.4321 |
| Prueba 6 | 0.3925 | 0.3936 | 0.3508 | 0.3618 | 0.3997 | 0.3878 | 0.3835 |
| Prueba 7 | 0.6004 | 0.5971 | 0.7219 | 0.5770 | 0.6316 | 0.5886 | 0.6245 |

Cuadro 5.25: Tiempo de entrenamiento por época para el modelo 05 en segundos.

| Época | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|----------|---------|---------|---------|---------|---------|---------|---------|---------|---------|
| Prueba 1 | 34.0148 | 30.9373 | 30.3723 | 30.1398 | 30.4514 | 30.5224 | 30.1990 | 30.0188 | 30.0193 |
| Prueba 2 | 34.0064 | 30.4619 | 30.5479 | 30.5585 | 30.4790 | 31.9454 | 32.2734 | 30.3462 | 30.4814 |
| Prueba 3 | 33.4509 | 30.2567 | 30.1635 | 30.3051 | 30.0790 | 30.2314 | 30.0850 | 30.2074 | 30.1006 |
| Prueba 4 | 40.8985 | 30.4233 | 30.2016 | 30.4246 | 30.1846 | 31.2620 | 31.0297 | 31.7170 | 30.5567 |
| Prueba 5 | 35.7058 | 30.4465 | 30.4395 | 31.3067 | 30.3618 | 30.9245 | 30.6016 | 30.7636 | 30.8833 |
| Prueba 6 | 34.0250 | 30.8195 | 31.6678 | 30.7807 | 30.7584 | 30.8432 | 31.0339 | 30.8348 | 30.6313 |
| Prueba 7 | 33.8194 | 29.9818 | 30.4822 | 30.2431 | 30.2114 | 30.1206 | 30.2472 | 30.1506 | 30.3780 |

| Época | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 |
|----------|---------|---------|---------|---------|---------|---------|---------|---------|---------|
| Prueba 1 | 30.1564 | 30.6890 | 30.2072 | 30.2662 | 30.3113 | 30.5734 | 29.9905 | 30.1972 | 29.9245 |
| Prueba 2 | 29.9688 | 30.0456 | 29.9359 | 29.9285 | 29.8882 | 30.0344 | 30.0961 | 30.1917 | 29.9982 |
| Prueba 3 | 30.1586 | 30.1644 | 32.1087 | 31.6789 | 32.5189 | 30.9256 | 32.2315 | 32.0370 | 31.1808 |
| Prueba 4 | 31.1029 | 30.6304 | 30.0056 | 30.3490 | 30.8915 | 30.4370 | 30.4350 | 30.2114 | 30.5326 |
| Prueba 5 | 31.0282 | 30.6676 | 30.6684 | 30.3104 | 30.6948 | 30.6519 | 30.8074 | 30.4498 | 30.8396 |
| Prueba 6 | 30.4679 | 30.1880 | 30.3831 | 31.2110 | 32.6539 | 32.0048 | 30.6092 | 30.6056 | 30.0616 |
| Prueba 7 | 30.0771 | 30.0808 | 30.0883 | 30.3705 | 30.0007 | 30.0429 | 30.0088 | 30.4777 | 30.3885 |

| Época | 19 | 20 | 21 | 22 | 23 | 24 | 25 |
|----------|---------|---------|---------|---------|---------|---------|---------|
| Prueba 1 | 30.0619 | 30.0122 | 30.3091 | 29.9860 | 32.0411 | 30.3173 | 29.8577 |
| Prueba 2 | 29.9612 | 30.1971 | 29.9878 | 29.8260 | 29.9136 | 29.7424 | 29.6564 |
| Prueba 3 | 31.3951 | 30.5522 | 29.1239 | 28.8510 | 28.8520 | 29.8821 | 29.8017 |
| Prueba 4 | 30.9385 | 30.5283 | 30.0896 | 30.3256 | 29.8925 | 30.1077 | 29.6105 |
| Prueba 5 | 30.8156 | 30.8895 | 30.5234 | 30.8128 | 30.7839 | 30.8616 | 30.2312 |
| Prueba 6 | 30.4102 | 31.1269 | 30.3261 | 30.3928 | 30.1899 | 30.2190 | 29.8090 |
| Prueba 7 | 29.9672 | 30.0587 | 30.2917 | 30.4980 | 30.4631 | 30.5089 | 29.9021 |

5.6. Resultados 04* GPU

Se resaltan en azul los valores máximos en las tablas de tasa de acierto y los mínimos en las tablas de perdida.

Cuadro 5.26: Tasa de acierto del grupo de entrenamiento para el modelo 04* GPU.

| Época | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|----------|--------|--------|--------|--------|--------|--------|--------|--------|--------|
| Prueba 1 | 0.4065 | 0.5081 | 0.5854 | 0.6504 | 0.7195 | 0.7967 | 0.7683 | 0.8537 | 0.8415 |
| Prueba 2 | 0.4553 | 0.5691 | 0.6504 | 0.7561 | 0.7724 | 0.8211 | 0.8374 | 0.8659 | 0.8862 |
| Prueba 3 | 0.4228 | 0.5325 | 0.5528 | 0.6016 | 0.6748 | 0.6992 | 0.7602 | 0.7561 | 0.7927 |
| Prueba 4 | 0.4959 | 0.4675 | 0.6016 | 0.6951 | 0.7602 | 0.7805 | 0.7764 | 0.8293 | 0.8618 |
| Prueba 5 | 0.4472 | 0.5488 | 0.5650 | 0.6220 | 0.6423 | 0.7358 | 0.7276 | 0.7480 | 0.7642 |

| Época | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 |
|----------|--------|--------|--------|--------|--------|--------|--------|--------|--------|
| Prueba 1 | 0.8821 | 0.8333 | 0.8699 | 0.8943 | 0.9106 | 0.8984 | 0.8902 | 0.9228 | 0.8984 |
| Prueba 2 | 0.8659 | 0.8943 | 0.9106 | 0.9309 | 0.9024 | 0.9268 | 0.9268 | 0.9106 | 0.9472 |
| Prueba 3 | 0.8293 | 0.8252 | 0.8455 | 0.8780 | 0.8659 | 0.9024 | 0.8740 | 0.9065 | 0.9065 |
| Prueba 4 | 0.8740 | 0.9065 | 0.9350 | 0.9146 | 0.8943 | 0.8862 | 0.9390 | 0.9065 | 0.9106 |
| Prueba 5 | 0.8171 | 0.8577 | 0.7967 | 0.8496 | 0.8455 | 0.8862 | 0.8780 | 0.8984 | 0.9106 |

| Época | 19 | 20 | 21 | 22 | 23 | 24 | 25 |
|----------|--------|--------|--------|--------|--------|--------|--------|
| Prueba 1 | 0.8984 | 0.9431 | 0.9106 | 0.9065 | 0.9106 | 0.9390 | 0.9268 |
| Prueba 2 | 0.9268 | 0.9309 | 0.9309 | 0.9512 | 0.9472 | 0.9431 | 0.9553 |
| Prueba 3 | 0.8984 | 0.9106 | 0.9228 | 0.8902 | 0.9024 | 0.9228 | 0.9146 |
| Prueba 4 | 0.9309 | 0.9146 | 0.9065 | 0.9472 | 0.9309 | 0.9268 | 0.9309 |
| Prueba 5 | 0.8902 | 0.8984 | 0.9065 | 0.9024 | 0.9390 | 0.9065 | 0.9187 |

Cuadro 5.27: Tasa de acierto del grupo de validación para el modelo 04* GPU.

| Época | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|----------|--------|--------|--------|--------|--------|--------|--------|--------|--------|
| Prueba 1 | 0.5000 | 0.5357 | 0.5714 | 0.6071 | 0.6429 | 0.7143 | 0.5714 | 0.7143 | 0.8929 |
| Prueba 2 | 0.6429 | 0.4643 | 0.6786 | 0.7143 | 0.7500 | 0.8929 | 0.6786 | 0.7857 | 0.8929 |
| Prueba 3 | 0.5000 | 0.5000 | 0.4643 | 0.5714 | 0.4643 | 0.6786 | 0.5714 | 0.6071 | 0.8571 |
| Prueba 4 | 0.5000 | 0.5000 | 0.5714 | 0.7143 | 0.6429 | 0.7500 | 0.8571 | 0.8571 | 0.8571 |
| Prueba 5 | 0.5000 | 0.5714 | 0.6071 | 0.6071 | 0.5714 | 0.6071 | 0.6786 | 0.8214 | 0.6429 |

| Época | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 |
|----------|--------|--------|--------|--------|--------|--------|--------|--------|--------|
| Prueba 1 | 0.8929 | 0.8571 | 0.8929 | 0.8214 | 0.8571 | 0.8929 | 0.8929 | 0.8929 | 0.8929 |
| Prueba 2 | 0.8929 | 0.8929 | 0.8929 | 0.8929 | 0.8214 | 0.8929 | 0.8929 | 0.8571 | 0.8929 |
| Prueba 3 | 0.7143 | 0.8929 | 0.8571 | 0.8571 | 0.8571 | 0.8929 | 0.8929 | 0.8571 | 0.8571 |
| Prueba 4 | 0.8571 | 0.8214 | 0.8571 | 0.8571 | 0.8214 | 0.8571 | 0.8571 | 0.8929 | 0.8571 |
| Prueba 5 | 0.8929 | 0.8571 | 0.9286 | 0.7500 | 0.8929 | 0.8929 | 0.8929 | 0.8929 | 0.8929 |

| Época | 19 | 20 | 21 | 22 | 23 | 24 | 25 |
|----------|--------|--------|--------|--------|--------|--------|--------|
| Prueba 1 | 0.8571 | 0.8929 | 0.8214 | 0.8571 | 0.8571 | 0.8571 | 0.8571 |
| Prueba 2 | 0.8929 | 0.8214 | 0.8571 | 0.8571 | 0.8214 | 0.8929 | 0.8929 |
| Prueba 3 | 0.8214 | 0.8571 | 0.8929 | 0.8929 | 0.8571 | 0.8571 | 0.8571 |
| Prueba 4 | 0.8214 | 0.8571 | 0.8571 | 0.8571 | 0.8214 | 0.8571 | 0.8214 |
| Prueba 5 | 0.8571 | 0.8929 | 0.8571 | 0.8571 | 0.8571 | 0.8571 | 0.8929 |

Cuadro 5.28: pérdida del grupo de entrenamiento para el modelo 04* GPU.

| Época | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|----------|--------|--------|--------|--------|--------|--------|--------|--------|--------|
| Prueba 1 | 2.4340 | 0.9588 | 0.9054 | 0.7863 | 0.6729 | 0.5770 | 0.5153 | 0.4218 | 0.4062 |
| Prueba 2 | 3.9235 | 0.9235 | 0.7679 | 0.6693 | 0.5355 | 0.4022 | 0.3686 | 0.3447 | 0.3234 |
| Prueba 3 | 3.1276 | 0.9607 | 0.9329 | 0.8618 | 0.7953 | 0.6934 | 0.6447 | 0.5121 | 0.4979 |
| Prueba 4 | 3.7822 | 0.9739 | 0.8650 | 0.7357 | 0.6327 | 0.5443 | 0.5292 | 0.4537 | 0.3415 |
| Prueba 5 | 1.8778 | 0.9794 | 0.9059 | 0.8293 | 0.8101 | 0.7241 | 0.6661 | 0.5517 | 0.5450 |

| Época | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 |
|----------|--------|--------|--------|--------|--------|--------|--------|--------|--------|
| Prueba 1 | 0.4100 | 0.3783 | 0.3264 | 0.2588 | 0.2683 | 0.2615 | 0.2556 | 0.2137 | 0.2359 |
| Prueba 2 | 0.3175 | 0.2413 | 0.2219 | 0.1716 | 0.2365 | 0.2287 | 0.1848 | 0.2001 | 0.1618 |
| Prueba 3 | 0.4184 | 0.3850 | 0.3468 | 0.2997 | 0.2896 | 0.2486 | 0.2423 | 0.2515 | 0.2239 |
| Prueba 4 | 0.3102 | 0.2782 | 0.2188 | 0.2420 | 0.2648 | 0.2273 | 0.2150 | 0.2724 | 0.2217 |
| Prueba 5 | 0.4427 | 0.3855 | 0.4069 | 0.3954 | 0.3323 | 0.2960 | 0.2754 | 0.2759 | 0.2317 |

| Época | 19 | 20 | 21 | 22 | 23 | 24 | 25 |
|----------|--------|--------|--------|--------|--------|--------|--------|
| Prueba 1 | 0.2256 | 0.1816 | 0.2086 | 0.2147 | 0.1952 | 0.1715 | 0.1657 |
| Prueba 2 | 0.1747 | 0.1510 | 0.1584 | 0.1284 | 0.1164 | 0.1199 | 0.1319 |
| Prueba 3 | 0.2274 | 0.2164 | 0.2036 | 0.2055 | 0.1968 | 0.1960 | 0.1835 |
| Prueba 4 | 0.1931 | 0.1963 | 0.1793 | 0.1488 | 0.1390 | 0.1558 | 0.1539 |
| Prueba 5 | 0.2535 | 0.2765 | 0.2464 | 0.2428 | 0.1923 | 0.2176 | 0.2162 |

Cuadro 5.29: pérdida del grupo de validación para el modelo 04* GPU.

| Época | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|----------|--------|--------|--------|--------|--------|--------|--------|--------|--------|
| Prueba 1 | 1.0512 | 0.9458 | 0.8565 | 0.7952 | 0.6789 | 0.6102 | 0.6061 | 0.5432 | 0.4196 |
| Prueba 2 | 1.0060 | 0.8531 | 0.6991 | 0.6680 | 0.5956 | 0.4990 | 0.5615 | 0.5413 | 0.5774 |
| Prueba 3 | 1.0044 | 0.9186 | 0.9221 | 0.7737 | 0.7228 | 0.6799 | 0.5751 | 0.5816 | 0.5338 |
| Prueba 4 | 1.0218 | 0.9092 | 0.7509 | 0.6602 | 0.6237 | 0.5141 | 0.4796 | 0.4691 | 0.4056 |
| Prueba 5 | 0.9972 | 0.9451 | 0.8745 | 0.8149 | 0.7391 | 0.6462 | 0.6316 | 0.5479 | 0.5189 |

| Época | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 |
|----------|--------|--------|--------|--------|--------|--------|--------|--------|--------|
| Prueba 1 | 0.4815 | 0.4747 | 0.4761 | 0.5595 | 0.5323 | 0.5228 | 0.5265 | 0.6025 | 0.5853 |
| Prueba 2 | 0.4810 | 0.5109 | 0.5248 | 0.6485 | 0.4607 | 0.4717 | 0.5540 | 0.5749 | 0.4860 |
| Prueba 3 | 0.5770 | 0.4275 | 0.4254 | 0.5184 | 0.4340 | 0.4652 | 0.4594 | 0.5022 | 0.5406 |
| Prueba 4 | 0.4303 | 0.4280 | 0.4478 | 0.4345 | 0.4581 | 0.3901 | 0.3731 | 0.3778 | 0.4499 |
| Prueba 5 | 0.4390 | 0.4480 | 0.4382 | 0.4627 | 0.4142 | 0.4311 | 0.4195 | 0.4283 | 0.4226 |

| Época | 19 | 20 | 21 | 22 | 23 | 24 | 25 |
|----------|--------|--------|--------|--------|--------|--------|--------|
| Prueba 1 | 0.6112 | 0.7316 | 0.7183 | 0.6007 | 0.5846 | 0.7351 | 0.6430 |
| Prueba 2 | 0.5576 | 0.6509 | 0.6741 | 0.5140 | 0.7871 | 0.6906 | 0.5968 |
| Prueba 3 | 0.5889 | 0.5606 | 0.6104 | 0.6276 | 0.5899 | 0.6750 | 0.7742 |
| Prueba 4 | 0.5624 | 0.5258 | 0.5739 | 0.5651 | 0.5882 | 0.5671 | 0.5809 |
| Prueba 5 | 0.5675 | 0.3820 | 0.4354 | 0.4589 | 0.5789 | 0.4778 | 0.5020 |

Cuadro 5.30: Tiempo de entrenamiento por época para el modelo 04* GPU en segundos.

| Época | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|----------|--------|--------|--------|--------|--------|--------|--------|--------|--------|
| Prueba 1 | 7.9560 | 2.6650 | 2.6851 | 2.6950 | 2.6803 | 2.6799 | 2.6902 | 2.6896 | 2.6900 |
| Prueba 2 | 5.7649 | 2.6748 | 2.6801 | 2.6602 | 2.6598 | 2.6848 | 2.6703 | 2.6500 | 2.6799 |
| Prueba 3 | 4.7672 | 2.2692 | 2.2505 | 2.2588 | 2.2548 | 2.2923 | 2.2504 | 2.2493 | 2.2871 |
| Prueba 4 | 5.0179 | 2.3050 | 2.2666 | 2.2991 | 2.2888 | 2.3000 | 2.2757 | 2.4945 | 2.3199 |
| Prueba 5 | 4.8939 | 2.3041 | 2.3534 | 2.3158 | 2.3508 | 2.4113 | 2.3191 | 2.5083 | 2.2844 |

| Época | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 |
|----------|--------|--------|--------|--------|--------|--------|--------|--------|--------|
| Prueba 1 | 2.6948 | 2.6701 | 2.6899 | 2.6900 | 2.6779 | 2.7169 | 2.6900 | 2.6698 | 2.7101 |
| Prueba 2 | 2.7402 | 2.7306 | 2.6767 | 2.4505 | 2.4380 | 2.3555 | 2.6762 | 2.7191 | 2.5920 |
| Prueba 3 | 2.2594 | 2.2264 | 2.2407 | 2.2751 | 2.2446 | 2.2999 | 2.2334 | 2.2587 | 2.2717 |
| Prueba 4 | 2.3604 | 2.4233 | 2.4988 | 2.5122 | 2.3529 | 2.6433 | 2.5905 | 2.4191 | 2.5403 |
| Prueba 5 | 2.3139 | 2.2739 | 4.0586 | 2.3006 | 2.3069 | 2.5360 | 2.3129 | 2.3534 | 2.2422 |

| Época | 19 | 20 | 21 | 22 | 23 | 24 | 25 |
|----------|--------|--------|--------|--------|--------|--------|--------|
| Prueba 1 | 2.6800 | 2.6899 | 2.6802 | 2.6899 | 2.6900 | 2.6800 | 2.5998 |
| Prueba 2 | 2.6229 | 2.4895 | 2.3004 | 2.2728 | 2.3210 | 2.2668 | 2.2934 |
| Prueba 3 | 2.3418 | 2.2902 | 2.2219 | 2.2975 | 2.4105 | 2.3492 | 2.3051 |
| Prueba 4 | 2.4330 | 2.5833 | 2.2819 | 2.2960 | 2.2941 | 2.2867 | 2.2645 |
| Prueba 5 | 2.3120 | 2.2976 | 2.2494 | 2.2846 | 2.3094 | 2.5772 | 2.2289 |

5.7. Resultados 05 GPU

Se resaltan en azul los valores máximos en las tablas de tasa de acierto y los mínimos en las tablas de perdida.

Cuadro 5.31: Tasa de acierto del grupo de entrenamiento para el modelo 05 GPU.

| Época | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|----------|--------|--------|--------|--------|--------|--------|--------|--------|--------|
| Prueba 1 | 0.4350 | 0.6301 | 0.6179 | 0.7439 | 0.7927 | 0.8171 | 0.8699 | 0.8618 | 0.8862 |
| Prueba 2 | 0.4350 | 0.5244 | 0.6138 | 0.6179 | 0.6951 | 0.7439 | 0.7642 | 0.7764 | 0.8049 |
| Prueba 3 | 0.4228 | 0.5122 | 0.6992 | 0.7236 | 0.7764 | 0.7927 | 0.8049 | 0.8577 | 0.8618 |
| Prueba 4 | 0.4309 | 0.5000 | 0.5407 | 0.6870 | 0.7276 | 0.7967 | 0.8211 | 0.8618 | 0.8862 |
| Prueba 5 | 0.4065 | 0.4797 | 0.6179 | 0.7276 | 0.7439 | 0.8252 | 0.8293 | 0.8415 | 0.8862 |
| Prueba 6 | 0.4390 | 0.5325 | 0.5935 | 0.6829 | 0.7846 | 0.8049 | 0.8333 | 0.8537 | 0.8577 |

| Época | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 |
|----------|--------|--------|--------|--------|--------|--------|--------|--------|--------|
| Prueba 1 | 0.8943 | 0.8902 | 0.8902 | 0.9228 | 0.8943 | 0.9065 | 0.9065 | 0.9065 | 0.8943 |
| Prueba 2 | 0.8496 | 0.8618 | 0.8984 | 0.8862 | 0.8943 | 0.8496 | 0.8780 | 0.8984 | 0.8943 |
| Prueba 3 | 0.8659 | 0.8821 | 0.8821 | 0.8902 | 0.8984 | 0.8821 | 0.9106 | 0.9065 | 0.9268 |
| Prueba 4 | 0.8577 | 0.8821 | 0.8943 | 0.8780 | 0.8943 | 0.9065 | 0.9268 | 0.8984 | 0.9146 |
| Prueba 5 | 0.9065 | 0.8902 | 0.8780 | 0.9146 | 0.8984 | 0.9106 | 0.9228 | 0.9146 | 0.9187 |
| Prueba 6 | 0.9024 | 0.9106 | 0.8780 | 0.9065 | 0.9106 | 0.9228 | 0.9228 | 0.9106 | 0.9187 |

| Época | 19 | 20 | 21 | 22 | 23 | 24 | 25 |
|----------|--------|--------|--------|--------|--------|--------|--------|
| Prueba 1 | 0.9390 | 0.9065 | 0.9431 | 0.9268 | 0.9146 | 0.9350 | 0.9228 |
| Prueba 2 | 0.9106 | 0.9065 | 0.9187 | 0.9350 | 0.9228 | 0.8984 | 0.9146 |
| Prueba 3 | 0.9106 | 0.9187 | 0.9065 | 0.9431 | 0.9268 | 0.9024 | 0.9228 |
| Prueba 4 | 0.9228 | 0.9187 | 0.9146 | 0.9268 | 0.9268 | 0.9228 | 0.9350 |
| Prueba 5 | 0.9350 | 0.9268 | 0.9268 | 0.9187 | 0.9350 | 0.9228 | 0.9146 |
| Prueba 6 | 0.9431 | 0.9309 | 0.9390 | 0.9228 | 0.9553 | 0.9228 | 0.9350 |

Cuadro 5.32: Tasa de acierto del grupo de validación para el modelo 05 GPU.

| Época | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|----------|--------|--------|--------|--------|--------|--------|--------|--------|--------|
| Prueba 1 | 0.7143 | 0.6786 | 0.7143 | 0.8571 | 0.8929 | 0.8571 | 0.8571 | 0.8571 | 0.8929 |
| Prueba 2 | 0.5000 | 0.6071 | 0.6071 | 0.6429 | 0.7143 | 0.7857 | 0.8214 | 0.8214 | 0.7500 |
| Prueba 3 | 0.3929 | 0.6429 | 0.8214 | 0.7857 | 0.7857 | 0.7857 | 0.8929 | 0.7500 | 0.8571 |
| Prueba 4 | 0.5000 | 0.5714 | 0.6071 | 0.7143 | 0.7500 | 0.7857 | 0.8571 | 0.8571 | 0.7143 |
| Prueba 5 | 0.2143 | 0.7143 | 0.7143 | 0.7500 | 0.7857 | 0.8929 | 0.8571 | 0.8929 | 0.8929 |
| Prueba 6 | 0.3929 | 0.5000 | 0.7500 | 0.7857 | 0.8571 | 0.7857 | 0.8214 | 0.7857 | 0.8571 |

| Época | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 |
|----------|--------|--------|--------|--------|--------|--------|--------|--------|--------|
| Prueba 1 | 0.8929 | 0.8929 | 0.8929 | 0.9286 | 0.9286 | 0.8571 | 0.9286 | 0.8929 | 0.9286 |
| Prueba 2 | 0.8929 | 0.8929 | 0.8929 | 0.8571 | 0.8214 | 0.8214 | 0.8571 | 0.8571 | 0.7500 |
| Prueba 3 | 0.8214 | 0.7857 | 0.8214 | 0.8571 | 0.7857 | 0.8214 | 0.8571 | 0.8214 | 0.8571 |
| Prueba 4 | 0.8571 | 0.8571 | 0.8571 | 0.8929 | 0.8571 | 0.8571 | 0.8929 | 0.8571 | 0.8571 |
| Prueba 5 | 0.8571 | 0.8571 | 0.8929 | 0.8929 | 0.8214 | 0.8571 | 0.8571 | 0.8214 | 0.8929 |
| Prueba 6 | 0.8214 | 0.8571 | 0.8214 | 0.8929 | 0.8571 | 0.8929 | 0.8214 | 0.8571 | 0.8571 |

| Época | 19 | 20 | 21 | 22 | 23 | 24 | 25 |
|----------|--------|--------|--------|--------|--------|--------|--------|
| Prueba 1 | 0.8929 | 0.8571 | 0.8929 | 0.8571 | 0.8929 | 0.8929 | 0.8571 |
| Prueba 2 | 0.8571 | 0.8571 | 0.8571 | 0.8571 | 0.8571 | 0.8571 | 0.8571 |
| Prueba 3 | 0.8571 | 0.8214 | 0.7857 | 0.8571 | 0.8571 | 0.7857 | 0.8214 |
| Prueba 4 | 0.8929 | 0.8571 | 0.8929 | 0.8571 | 0.8929 | 0.8929 | 0.8571 |
| Prueba 5 | 0.8571 | 0.8929 | 0.8929 | 0.8571 | 0.8929 | 0.8929 | 0.8929 |
| Prueba 6 | 0.7857 | 0.8571 | 0.8929 | 0.7500 | 0.8571 | 0.8214 | 0.8214 |

Cuadro 5.33: pérdida del grupo de entrenamiento para el modelo 05 GPU.

| Época | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|----------|---------|--------|--------|--------|--------|--------|--------|--------|--------|
| Prueba 1 | 6.7304 | 0.8403 | 0.7914 | 0.6460 | 0.5249 | 0.4281 | 0.3458 | 0.3227 | 0.3046 |
| Prueba 2 | 5.8001 | 1.0002 | 0.9233 | 0.8463 | 0.7851 | 0.6681 | 0.5699 | 0.5324 | 0.4984 |
| Prueba 3 | 5.2933 | 0.9399 | 0.7977 | 0.6516 | 0.5458 | 0.4809 | 0.4441 | 0.3792 | 0.3372 |
| Prueba 4 | 16.0956 | 1.0849 | 0.8828 | 0.7180 | 0.6049 | 0.4825 | 0.4304 | 0.3726 | 0.3037 |
| Prueba 5 | 8.4319 | 0.9726 | 0.7946 | 0.6859 | 0.6186 | 0.4696 | 0.4643 | 0.3920 | 0.3111 |
| Prueba 6 | 7.4066 | 0.9599 | 0.7904 | 0.6478 | 0.5245 | 0.4272 | 0.3854 | 0.3159 | 0.3125 |

| Época | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 |
|----------|--------|--------|--------|--------|--------|--------|--------|--------|--------|
| Prueba 1 | 0.2623 | 0.2459 | 0.2418 | 0.2075 | 0.1980 | 0.2120 | 0.2252 | 0.2195 | 0.2014 |
| Prueba 2 | 0.4166 | 0.3536 | 0.3283 | 0.2533 | 0.2829 | 0.3438 | 0.3048 | 0.2786 | 0.2594 |
| Prueba 3 | 0.3141 | 0.2843 | 0.2646 | 0.2553 | 0.2433 | 0.2732 | 0.2855 | 0.2182 | 0.1830 |
| Prueba 4 | 0.3898 | 0.3489 | 0.2736 | 0.2557 | 0.2486 | 0.2109 | 0.1822 | 0.2324 | 0.1916 |
| Prueba 5 | 0.2899 | 0.2941 | 0.3140 | 0.2368 | 0.2342 | 0.2102 | 0.1783 | 0.1937 | 0.1876 |
| Prueba 6 | 0.2397 | 0.2257 | 0.2497 | 0.2112 | 0.1979 | 0.1906 | 0.1763 | 0.1921 | 0.1679 |

| Época | 19 | 20 | 21 | 22 | 23 | 24 | 25 |
|----------|--------|--------|--------|--------|--------|--------|--------|
| Prueba 1 | 0.1812 | 0.1868 | 0.1544 | 0.1691 | 0.1828 | 0.1682 | 0.1737 |
| Prueba 2 | 0.2323 | 0.2598 | 0.1777 | 0.1728 | 0.1931 | 0.1929 | 0.1865 |
| Prueba 3 | 0.1958 | 0.2166 | 0.2080 | 0.1646 | 0.1890 | 0.2015 | 0.1826 |
| Prueba 4 | 0.1792 | 0.1946 | 0.1999 | 0.1780 | 0.1721 | 0.1859 | 0.1803 |
| Prueba 5 | 0.1821 | 0.2047 | 0.1669 | 0.2229 | 0.1928 | 0.1587 | 0.1532 |
| Prueba 6 | 0.1714 | 0.1709 | 0.1419 | 0.1710 | 0.1465 | 0.1559 | 0.1412 |

Cuadro 5.34: pérdida del grupo de validación para el modelo 05 GPU.

| Época | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|----------|--------|--------|--------|--------|--------|--------|--------|--------|--------|
| Prueba 1 | 0.8752 | 0.6978 | 0.6191 | 0.5091 | 0.4547 | 0.3525 | 0.3439 | 0.3656 | 0.3487 |
| Prueba 2 | 1.0230 | 0.9522 | 0.8396 | 0.7455 | 0.6748 | 0.5693 | 0.5059 | 0.5655 | 0.4639 |
| Prueba 3 | 0.9867 | 0.7841 | 0.6084 | 0.4771 | 0.4693 | 0.4784 | 0.4242 | 0.4439 | 0.4085 |
| Prueba 4 | 1.2342 | 0.8470 | 0.6886 | 0.5894 | 0.4929 | 0.5082 | 0.4020 | 0.3509 | 0.5842 |
| Prueba 5 | 1.2174 | 0.7589 | 0.6806 | 0.5140 | 0.5972 | 0.4388 | 0.4818 | 0.4578 | 0.4489 |
| Prueba 6 | 1.0492 | 0.8897 | 0.6031 | 0.5394 | 0.5491 | 0.4683 | 0.5564 | 0.4977 | 0.4054 |

| Época | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 |
|----------|--------|--------|--------|--------|--------|--------|--------|--------|--------|
| Prueba 1 | 0.3146 | 0.3418 | 0.3662 | 0.3917 | 0.3939 | 0.3581 | 0.3182 | 0.3106 | 0.3383 |
| Prueba 2 | 0.4099 | 0.4119 | 0.4148 | 0.4155 | 0.4750 | 0.5066 | 0.3998 | 0.4689 | 0.4911 |
| Prueba 3 | 0.4091 | 0.4328 | 0.4484 | 0.4072 | 0.4783 | 0.4657 | 0.4512 | 0.5365 | 0.4035 |
| Prueba 4 | 0.4394 | 0.3447 | 0.3442 | 0.3500 | 0.3432 | 0.3074 | 0.3199 | 0.3734 | 0.3582 |
| Prueba 5 | 0.4753 | 0.4588 | 0.3974 | 0.4358 | 0.4485 | 0.4290 | 0.4197 | 0.4644 | 0.4290 |
| Prueba 6 | 0.5417 | 0.5342 | 0.4672 | 0.5259 | 0.4909 | 0.4938 | 0.5020 | 0.5109 | 0.4620 |

| Época | 19 | 20 | 21 | 22 | 23 | 24 | 25 |
|----------|--------|--------|--------|--------|--------|--------|--------|
| Prueba 1 | 0.3278 | 0.3444 | 0.4089 | 0.4751 | 0.3398 | 0.3610 | 0.4034 |
| Prueba 2 | 0.4304 | 0.4284 | 0.4711 | 0.5744 | 0.4666 | 0.5036 | 0.4804 |
| Prueba 3 | 0.4325 | 0.3763 | 0.4585 | 0.4689 | 0.5437 | 0.4442 | 0.4230 |
| Prueba 4 | 0.3498 | 0.3667 | 0.3083 | 0.3130 | 0.3049 | 0.3251 | 0.3386 |
| Prueba 5 | 0.3810 | 0.3970 | 0.3678 | 0.3853 | 0.3915 | 0.4288 | 0.4782 |
| Prueba 6 | 0.5630 | 0.5003 | 0.5478 | 0.5524 | 0.5193 | 0.5560 | 0.5266 |

Cuadro 5.35: Tiempo de entrenamiento por época para el modelo 05 GPU en segundos.

| Época | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|----------|--------|--------|--------|--------|--------|--------|--------|--------|--------|
| Prueba 1 | 8.6200 | 1.4403 | 1.4198 | 1.4400 | 1.4297 | 1.4604 | 1.4401 | 1.4503 | 1.4500 |
| Prueba 2 | 2.2500 | 1.5319 | 1.4780 | 1.4101 | 1.4158 | 1.5140 | 1.4653 | 1.3897 | 1.3831 |
| Prueba 3 | 2.2698 | 1.4099 | 1.3786 | 1.4550 | 1.4699 | 1.4600 | 1.5003 | 1.4402 | 1.4108 |
| Prueba 4 | 2.5574 | 1.4385 | 1.4088 | 1.3849 | 1.4001 | 1.3901 | 1.3798 | 1.4001 | 1.4303 |
| Prueba 5 | 2.2000 | 1.3997 | 1.4200 | 1.3804 | 1.3801 | 1.5699 | 1.4999 | 1.5099 | 1.4904 |
| Prueba 6 | 2.2999 | 1.5200 | 1.5200 | 1.4397 | 1.4401 | 1.4404 | 1.4500 | 1.4599 | 1.4599 |

| Época | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 |
|----------|--------|--------|--------|--------|--------|--------|--------|--------|--------|
| Prueba 1 | 1.4498 | 1.4498 | 1.4404 | 1.4247 | 1.4252 | 1.4499 | 1.4599 | 1.4600 | 1.5927 |
| Prueba 2 | 1.3987 | 1.3957 | 1.3997 | 1.4099 | 1.4803 | 1.4397 | 1.4052 | 1.3952 | 1.4495 |
| Prueba 3 | 1.4490 | 1.5001 | 1.4400 | 1.4005 | 1.4193 | 1.4002 | 1.4299 | 1.3997 | 1.4202 |
| Prueba 4 | 1.4349 | 1.3999 | 1.4050 | 1.3799 | 1.4197 | 1.4154 | 1.4348 | 1.4501 | 1.4300 |
| Prueba 5 | 1.5100 | 1.5297 | 1.4902 | 1.4378 | 1.4401 | 1.4998 | 1.5305 | 1.4896 | 1.4800 |
| Prueba 6 | 1.4899 | 1.4801 | 1.4899 | 1.4801 | 1.4497 | 1.4500 | 1.4952 | 1.5174 | 1.4776 |

| Época | 19 | 20 | 21 | 22 | 23 | 24 | 25 |
|----------|--------|--------|--------|--------|--------|--------|--------|
| Prueba 1 | 1.5789 | 1.5214 | 1.4696 | 1.4152 | 1.5003 | 1.4396 | 1.3202 |
| Prueba 2 | 1.4451 | 1.4649 | 1.4499 | 1.4104 | 1.4098 | 1.4003 | 1.3247 |
| Prueba 3 | 1.4000 | 1.3998 | 1.3899 | 1.4002 | 1.4300 | 1.4399 | 1.3603 |
| Prueba 4 | 1.4499 | 1.4402 | 1.4297 | 1.4402 | 1.4749 | 1.4253 | 1.3598 |
| Prueba 5 | 1.4901 | 1.5550 | 1.5098 | 1.4702 | 1.4550 | 1.5097 | 1.4203 |
| Prueba 6 | 1.4600 | 1.4598 | 1.4453 | 1.4447 | 1.4301 | 1.5099 | 1.4198 |