

UNIVERSIDAD DE SANTIAGO DE CHILE
FACULTAD DE INGENIERÍA
Departamento de Ingeniería Informática



**Clasificación Morfológica de Cabezas de Espermatozoides Humanos
Usando Redes Convolucionales**

Javier Andrés Vásquez Lizana

Profesor guía: Violeta Noemí Chang Camacho

Trabajo de titulación en conformidad a los
requisitos para obtener el título de Ingeniero
Civil en Informática

Santiago – Chile

2020

© Javier Andrés Vásquez Lizana , 2020



- Algunos derechos reservados. Esta obra está bajo una Licencia Creative Commons Atribución-NoComercial-Compartirlgual 3.0. Sus condiciones de uso pueden ser revisadas en: <http://creativecommons.org/licenses/by/3.0/cl/>.

RESUMEN

La clasificación morfológica de cabezas de espermatozoides, es un proceso realizado de forma manual por médicos expertos, con el objetivo de determinar el potencial fértil de los pacientes. Este proceso es subjetivo, ya que son los mismos médicos quienes observan las muestras y deciden si corresponde a una muestra normal de cabeza de espermatozoide, o tiene alguna anomalía.

En este trabajo se presenta la utilización de métodos basados en *deep learning*, para realizar la clasificación morfológica de las cabezas de espermatozoides humanos, de manera automática, para así dejar de lado la subjetividad de los expertos y entregar diagnósticos más certeros. En este trabajo, específicamente, se utilizan dos redes convolucionales para resolver esta problemática.

El *dataset* utilizado, corresponde a aproximadamente 1.300 imágenes de cabezas de espermatozoides humanos, que fueron clasificadas por tres expertos distintos de la Universidad de Chile, clasificándolas en cinco clases de formas distintas: normal, cónica, piriforme, pequeña y amorfa.

Los mejores resultados obtenidos en esta investigación, corresponden a una exactitud global en la clasificación de un 42% aproximadamente, en donde ambas redes convolucionales elegidas tuvieron problemas para identificar la clase amorfa de cabeza de espermatozoide, presente en el *dataset* utilizado en este trabajo. Estos resultados están muy por debajo de los obtenidos en la última publicación de clasificación morfológica de estas cabezas (año 2019), en donde se había obtenido una exactitud global de 62% en la clasificación morfológica.

Palabras Claves: Deep Learning; Redes neuronales convolucionales; Clasificación; Cabezas de espermatozoides humanos; Morfología

*Time passes, people move.
Like a river's flow, it never ends.
A childish mind will turn to noble ambition.
Young love will become deep affection.
The clear water's surface reflects growth.*

AGRADECIMIENTOS

Quiero agradecer principalmente a la mujer que me ha acompañado los últimos ocho años de mi vida, la cuál ha estado apoyándome incondicionalmente, incluso en los momentos más negros de mi vida personal y académica.

También debo agradecer a los amigos que forme en esta universidad, que me acompañaron durante los seis años que estuve en esta carrera, y mantuvieron una sonrisa en mi rostro a pesar de la terrible carga académica que este sistema nos impuso.

Por último, quiero agradecer a mi familia, especialmente a mi padre, quién falleció en el transcurso de este trabajo. A pesar de nuestras diferencias, él me dio un hogar y educación, por lo que siempre le estaré agradecido.

TABLA DE CONTENIDO

1	Introducción	1
1.1	Antecedentes y motivación	1
1.2	Descripción del problema	2
1.3	Solución propuesta	3
1.3.1	Hipótesis	3
1.4	Objetivos y alcance del proyecto	4
1.4.1	Objetivo general	4
1.4.2	Objetivos específicos	4
1.4.3	Alcances de la propuesta	4
1.5	Metodología y herramientas utilizadas	5
1.5.1	Metodología	5
1.5.2	Software y hardware requerido	7
1.6	Organización del documento	8
2	Marco Teórico	10
2.1	EspERMATOZOIDES humanos	10
2.2	Análisis de espermatozoides	11
2.2.1	Manual para procedimiento de análisis	11
2.2.2	Análisis morfológico de espermatozoides	13
2.3	Aprendizaje de máquinas	14
2.3.1	Aprendizaje supervisado	17
2.3.2	Aprendizaje no supervisado	18
2.4	Aprendizaje profundo	19
2.4.1	Función de costo	20
2.4.2	Optimizadores	20
2.4.3	Red neuronal convolucional	24
2.4.4	Aprendizaje por transferencia	31
2.5	Sesgo y Varianza	32
2.6	Aumento de datos	33
2.7	Validación cruzada	34
2.7.1	Métricas de evaluación	35
2.8	Resumen	38
3	Estado del arte	40
3.1	Estándar de oro para análisis morfológico de espermatozoides, asistido por computadoras (2017)	41
3.2	Clasificación morfológica automática de cabezas de espermatozoides humanos (2017)	42
3.3	Un nuevo método de aprendizaje profundo para la evaluación automática imágenes de espermatozoides humanos (2019)	43
3.4	Aprendizaje profundo para la clasificación de espermatozoides humanos (2019) . .	45
3.5	Clasificación morfológica, basa en aprendizaje profundo, de cabezas de espermatozoides humanos (2020)	47
3.6	Resumen	51
4	Elección de la red convolucional	52
4.1	Investigación sobre arquitecturas convolucionales	52
4.2	Resumen	59
5	Desarrollo de la investigación	60

5.1	Descripción de los datos	60
5.2	Aumento de los datos	63
5.2.1	Preprocesamiento y lectura de los datos	63
5.2.2	Limpieza de los datos	64
5.2.3	Proceso de aumento de datos	65
5.2.4	Aumento de datos equitativo	70
5.2.5	Aumento de datos dirigido	72
5.3	Adaptación de los modelos convolucionales escogidos	73
5.3.1	Adaptación de la red convolucional usada para clasificación de glóbulos rojos	73
5.3.2	Adaptación de la red convolucional de <i>ResNet50</i>	77
5.4	Afinamiento de parámetros para el entrenamiento	78
5.4.1	Elección del optimizador	79
5.4.2	Elección del número de épocas a entrenar	80
5.4.3	Elección de los mejores parámetros	81
5.5	Implementación de aprendizaje por transferencia	82
5.5.1	Pre-entrenamiento de las redes convolucionales escogidas	82
5.5.2	Modificación del mejor modelo convolucional obtenido	84
5.6	Resumen	85
6	Resultados experimentales	87
6.1	Validación cruzada	87
6.1.1	Separación del <i>dataset</i>	87
6.2	Cantidad de imágenes utilizadas	91
6.3	Primer experimento: entrenamiento desde cero	92
6.3.1	Protocolo de experimentación	92
6.3.2	Resultados del afinamiento de parámetros	94
6.3.3	Resultados de la clasificación morfológica	103
6.4	Segundo experimento: entrenamiento con aprendizaje por transferencia	108
6.4.1	Protocolo de experimentación	108
6.4.2	Pre-entrenamiento de las redes convolucionales	108
6.4.3	Resultados del afinamiento de parámetros	110
6.4.4	Resultados de la clasificación morfológica	118
6.5	Tercer experimento: modificación de la mejor red obtenida	121
6.5.1	Algoritmo de implementación	121
6.5.2	Modificaciones a realizar	122
6.5.3	Resultados congelando los primeros dos bloques de la red <i>RBC</i>	123
6.6	Resumen	127
7	Conclusiones	129
7.1	Dificultades encontradas	129
7.1.1	Objetivos específicos	130
7.1.2	Objetivo general	133
7.2	Trabajo a futuro	134
7.3	Observaciones finales	136
Glosario		137
Referencias bibliográficas		138
Apéndices		143
A Gráficos de afinamiento de datos		143
A.1	Experimento 1	143
A.2	Experimento 2	146

ÍNDICE DE TABLAS

Tabla 2.1 Ejemplo de matriz de confusión Fuente: Elaboración propia, 2020.	35
Tabla 3.1 Resumen: exactitud de la clasificación según Chang et al. (2017a). Fuente: Elaboración propia, 2020.	41
Tabla 3.2 Resultados de la clasificación según Chang et al. (2017a) con descriptores de Fourier. Fuente: Elaboración propia, 2020.	41
Tabla 3.3 Resumen de resultados de la clasificación según Chang et al. (2017b) Fuente: Elaboración propia, 2020.	42
Tabla 3.4 Matriz de confusión según Riordon et al. (2019), utilizando el conjunto de acuerdo parcial. Fuente: Elaboración propia, 2020.	47
Tabla 3.5 Resumen de resultados de la clasificación según Riordon et al. (2019). Fuente: Elaboración propia, 2020.	47
Tabla 3.6 Matriz de confusión resultante según Iqbal et al. (2020) con el conjunto de acuerdo parcial. Fuente: Elaboración propia, 2020.	49
Tabla 3.7 Resumen de resultados de la clasificación según Iqbal et al. (2020). Fuente: Elaboración propia, 2020.	49
Tabla 5.1 Escenarios de acuerdo en el conjunto de imágenes de espermatozoides Fuente: Elaboración propia, 2020.	61
Tabla 6.1 Imágenes por clases de forma de cabezas estudiadas, para el conjunto de evaluación y de entrenamiento Fuente: Elaboración propia, 2020.	91
Tabla 6.2 Separación del conjunto de entrenamiento, con aumento de datos equitativo Fuente: Elaboración propia, 2020.	91
Tabla 6.3 Separación del conjunto de entrenamiento luego de aumento de datos dirigido Fuente: Elaboración propia, 2020.	92
Tabla 6.4 Matriz de confusión promedio, y normalizada, de la clasificación realizada por la red <i>RBC</i> desde cero.	104
Tabla 6.5 Métricas de evaluación de la clasificación realizada por la red <i>RBC</i> , obtenida de la matriz de confusión normalizada	106
Tabla 6.6 Comparación de métricas finales del primer experimento, con respecto al estado del arte. Fuente: Elaboración propia, 2020.	107
Tabla 6.7 Matriz de confusión promedio, y normalizada, de la clasificación realizada por la red <i>RBC</i> con aprendizaje por transferencia.	119
Tabla 6.8 Métricas de evaluación de la clasificación realizada por la red <i>RBC</i> con aprendizaje por transferencia, obtenida de la matriz de confusión normalizada	120
Tabla 6.9 Comparación de métricas finales del segundo experimento con respecto al estado del arte. Fuente: Elaboración propia, 2020.	121
Tabla 6.10 Matriz de confusión promedio, y normalizada, de la clasificación realizada por la red <i>RBC</i> con sus primeros dos bloques congelados.	125
Tabla 6.11 Métricas de evaluación de la clasificación realizada por la red <i>RBC</i> con sus primeros dos bloques congelados, obtenida de la matriz de confusión normalizada.	126
Tabla 6.12 Comparación de métricas finales del experimento tres con respecto al estado del arte. Fuente: Elaboración propia, 2020.	127

ÍNDICE DE FIGURAS

Figura 1.1 Ejemplo de clasificación morfológica de cabezas de espermatozoides humanos	3
Figura 2.1 Proceso de espermiogénesis y estructura de un espermatozoide (Barría & Ramírez, 2014)	11
Figura 2.2 Ejemplo de aglutinamiento en muestras de espermatozoides (OMS, 2010b)	12
Figura 2.3 Tipos de formas de espermatozoides (OMS, 2010b)	14
Figura 2.4 Ejemplo de arquitectura de una red neuronal (Bre et al., 2017)	15
Figura 2.5 Entrenando una red neuronal	16
Figura 2.6 Ejemplo <i>SGD</i> con y sin <i>momentum</i> (Ruder, 2017)	22
Figura 2.7 Ejemplo de la aplicación del método de <i>Nesterov</i> (Ruder, 2017)	22
Figura 2.8 Comparación de error versus cantidad de épocas según optimizadores (Kingma & Ba, 2014)	24
Figura 2.9 Comparación de error versus cantidad de épocas según optimizadores (Wilson et al., 2018)	24
Figura 2.10 Características en reconocimiento de imágenes (Siegel et al., 2016)	25
Figura 2.11 Ejemplo de una capa convolucional de una <i>CNN</i>	26
Figura 2.12 Ejemplo de convolución en una imagen	27
Figura 2.13 Capa de activación <i>ReLU</i>	28
Figura 2.14 Ejemplo de función de agrupación <i>Max pooling</i>	29
Figura 2.15 Ejemplos de transformaciones hechas con <i>Keras</i> en <i>Python</i> (Sarin, 2019)	33
Figura 2.16 Ejemplo de curva <i>ROC</i>	38
Figura 3.1 Formas de cabezas de espermatozoides (Chang et al., 2017a)	40
Figura 3.2 Ejemplos de imágenes del <i>dataset</i> de <i>MHSMA</i> (Javadi & Mirroshandel, 2019)	44
Figura 3.3 Arquitectura de la red convolucional utilizada por Javadi & Mirroshandel (2019)	45
Figura 3.4 Red convolucional propuesta por Iqbal et al. (2020)	50
Figura 4.1 Formas de glóbulos rojos (Xu et al., 2017)	53
Figura 4.2 Estructura de red neuronal convolucional usada por Xu et al. (2017)	53
Figura 4.3 Formas de glóbulos rojos por patología (Angluin et al., 2016)	55
Figura 4.4 Red convolucional planteada por Angluin et al. (2016)	56
Figura 4.5 Ejemplo de red residual de 34 capas convolucionales (He et al., 2015)	58
Figura 5.1 Ejemplos de las clases en el conjunto de SCIAN-MorphoSpermGS	61
Figura 5.2 Ejemplos de imágenes diferentes en la misma categoría	62
Figura 5.3 Ejemplos de corrección de bordes negros	65
Figura 5.4 Ejemplos erróneos de aumento de datos	65
Figura 5.5 Ejemplo de aumento de datos realizado por Javadi & Mirroshandel (2019)	66
Figura 5.6 Ejemplos de trasladar una imagen	67
Figura 5.7 Ejemplos de reflexión vertical y horizontal	68
Figura 5.8 Ejemplos de rotación de una cabeza de espermatozoide	69
Figura 5.9 Ejemplos de cambio de brillo de una cabeza de espermatozoide	69
Figura 5.10 Arquitectura de red convolucional original de Angluin et al. (2016)	73
Figura 5.11 Arquitectura de la red convolucional Angluin et al. (2016) adaptada	76
Figura 5.12 Resumen arquitectura de <i>ResNet50</i> (Mahmood et al., 2020)	77
Figura 5.13 Arquitectura de <i>ResNet50</i> adaptada al experimento actual	78
Figura 6.1 Validación cruzada <i>shuffle split</i> realizada en este trabajo	89

Figura 6.2 Bloque de balanceo de clases para aumento de datos equitativo, con bolsas de datos	90
Figura 6.3 Bloque de balanceo de clases para aumento de datos equitativo, usando pesos en las clases	90
Figura 6.4 Error y exactitud en el entrenamiento de la red <i>RBC</i> con AD equitativo, usando bolsas de igual tamaño	95
Figura 6.5 Error y exactitud promedio en el entrenamiento de la red <i>RBC</i> con AD equitativo, usando pesos por clases	96
Figura 6.6 Error en el entrenamiento de la red <i>RBC</i> con AD dirigido, utilizando el optimizador C	97
Figura 6.7 Error y exactitud promedio en el entrenamiento de la red <i>RBC</i> con AD dirigido, utilizando el optimizador C	98
Figura 6.8 Error y exactitud promedio en el entrenamiento de la red <i>RBC</i> con AD dirigido	99
Figura 6.9 Error y exactitud en el entrenamiento de la red <i>ResNet50</i> con AD equitativo, usando bolsas de igual tamaño	100
Figura 6.10 Error y exactitud en el entrenamiento de la red <i>ResNet50</i> con AD equitativo, usando pesos por clases	101
Figura 6.11 Error y exactitud en el entrenamiento de la red <i>ResNet50</i> con AD dirigido	102
Figura 6.12 Curva <i>ROC</i> de la clasificación morfológica de cabezas de espermatozoídes, utilizando la red <i>RBC</i>	105
Figura 6.13 Error durante el pre-entrenamiento de la red <i>RBC</i> con el optimizador <i>Adam</i>	109
Figura 6.14 Error durante el pre-entrenamiento de la red <i>ResNet50</i> con el optimizador <i>SGD</i>	110
Figura 6.15 Error y exactitud en el entrenamiento de la red <i>RBC</i> pre-entrenada, con AD equitativo, usando bolsas de igual tamaño	112
Figura 6.16 Error y exactitud en el entrenamiento de la red <i>RBC</i> pre-entrenada, con AD equitativo, usando pesos por clases	113
Figura 6.17 Error y exactitud promedio en el entrenamiento de la red <i>RBC</i> pre-entrenada, con AD dirigido	114
Figura 6.18 Error y exactitud en el entrenamiento de la red <i>ResNet50</i> con AD equitativo, usando bolsas de igual tamaño	115
Figura 6.19 Error y exactitud en el entrenamiento de la red <i>ResNet50</i> con AD equitativo, usando pesos por clases	116
Figura 6.20 Error y exactitud en el entrenamiento de la red <i>ResNet50</i> con AD dirigido	117
Figura 6.21 Curva <i>ROC</i> de la clasificación morfológica de cabezas de espermatozoídes, utilizando la red <i>RBC</i> pre-entrenada	119
Figura 6.22 Arquitectura de la red convolucional Angluin et al. (2016) adaptada, separada en bloques	123
Figura 6.23 Error y exactitud promedio en el entrenamiento de la red <i>RBC</i> con los primeros bloques congelados	124
Figura 6.24 Curva <i>ROC</i> de la clasificación morfológica de cabezas de espermatozoídes, utilizando la red <i>RBC</i> con sus primeros dos bloques congelados.	126
 Figura A.1 Error y exactitud promedio en el entrenamiento de la red <i>RBC</i> con AD dirigido en el experimento 1	143
Figura A.2 Error y exactitud promedio en el entrenamiento de la red <i>RBC</i> con AD dirigido en el experimento 1	144
Figura A.3 Error y exactitud promedio en el entrenamiento de la red <i>RBC</i> con AD dirigido en el experimento 1	145
Figura A.4 Error y exactitud promedio en el entrenamiento de la red <i>RBC</i> con AD dirigido en el experimento 2	146
Figura A.5 Error y exactitud promedio en el entrenamiento de la red <i>RBC</i> con AD dirigido en el experimento 2	147

Figura A.6 Error y exactitud promedio en el entrenamiento de la red <i>RBC</i> con AD dirigido en el experimento 2	148
--	-----

CAPÍTULO 1. INTRODUCCIÓN

1.1 ANTECEDENTES Y MOTIVACIÓN

Aproximadamente el 15% de las parejas alrededor del mundo deben combatir contra la infertilidad (OMS, 2010a). Entre otras miradas de estudio, se han estudiado las causas de esta condición a través del análisis morfológico de los espermatozoides de los pacientes. Trabajos anteriores relacionan las tasas de embarazos con respecto al número de espermatozoides con una forma normal, en donde esta clasificación está basada en criterios morfológicos definidos, tanto para las cabezas, pieza media y colas de este tipo de células (Lee et al., 2002, p. 277).

Debido a lo anterior, el análisis de la forma de las cabezas de espermatozoides es una herramienta importante al momento de realizar un pronóstico acerca de la fertilidad de la persona, ya que a partir de este análisis, los médicos especialistas pueden obtener un diagnóstico sobre la cantidad de espermatozoides "normales" de la persona, y así sugerir alguna tecnología de reproducción asistida (Menkveld, 2013, p. 46). Estos tratamientos varían su complejidad dependiendo del grado de fertilidad del hombre. A mayor dificultad del proceso, se necesitan más expertos, materiales, recursos, etc. Por ejemplo, la fecundación *in vitro* es un proceso que se realiza en laboratorio, en donde el espermatozoide es implantado en el óvulo de forma artificial y una vez fecundado, es introducido mediante operación en el ovario de una mujer (Zegers-Hochschild et al., 2009).

En el presente, existen productos comerciales que tienen funcionalidades de análisis morfológico de espermatozoides, pero que no ofrecen funcionalidades de estudio para las cabezas de espermatozoides, que han sido definidas como amorfas por la OMS (2010b). Estos productos se venden como cajas negras, por lo que no dan espacio a estudios y desarrollos, para así lograr mejorar la clasificación morfológica de los espermatozoides. Estos programas comerciales, son también difíciles de obtener, debido a sus costos monetarios (Chang, 2015).

Es por esto que son los expertos los encargados de ver las imágenes de estos espermatozoides y clasificar la forma que estos tienen, a partir de sus conocimientos y experiencia. Esto significa que este diagnóstico está basado en la subjetividad de los especialistas que realizan el análisis visual. Por ejemplo, en la base de datos que se utiliza en este trabajo, se puede observar cómo en el etiquetado de las cabezas de espermatozoides hecho por tres expertos distintos y con amplia trayectoria en el área, existen diferencias de opinión al momento de distinguir la forma de las cabezas de los espermatozoides.

Se han publicado diversos trabajos con el objetivo de encontrar el mejor método asistido por computadora para la clasificación de cabezas de espermatozoides humanos. Este

mejor método es denominado como patrón de oro, o *gold standard* en inglés, y desde ahora en adelante cada mención de patrón de oro, se referirá a la definición anterior. Más adelante, en el Capítulo 3 se presentan resultados de algunos experimentos que han intentado resolver esta problemática, especialmente para categorizar las cabezas que son amorfas entre los grupos de imágenes.

1.2 DESCRIPCIÓN DEL PROBLEMA

El proceso de análisis y clasificación morfológica de las cabezas de espermatozoides es realizado de forma manual por expertos en el área de la andrología (Merriam-Webster, s.f), en donde los médicos observan las imágenes de las cabezas de estas células, y deciden bajo su experiencia y conocimientos, qué forma tienen.

A continuación se muestra como ejemplo el caso del *dataset* público de **SCIAND-MorphoSpermGS** el cual es un conjunto de imágenes de cabezas de espermatozoides, las cuales fueron etiquetados por tres expertos chilenos, de manera independiente. Este conjunto contiene un total de 1.854 imágenes en donde cada una contiene solo una cabeza de espermatozoide.

En este *dataset*, los tres expertos clasificaron solo clasificaron 384 cabezas con la misma forma, es decir, en menos del 25% del total de imágenes los expertos llegaron a un acuerdo común, mientras que las imágenes restantes, solo dos expertos estuvieron de acuerdo en la forma, o incluso, los tres expertos clasificaron las imágenes con formas distintas. En la Figura 1.1 se ve un ejemplo de algunas imágenes siendo clasificadas por los tres expertos distintos.

Este caso es un ejemplo real de la subjetividad que existe en este proceso de clasificación realizado por los expertos, los que, a pesar de tener experiencia en el campo, tuvieron problemas para decidir la forma de las cabezas de los espermatozoides presentes en el *dataset*.

Es debido a esto, que se han buscado formas de realizar este trabajo de manera automática y asistido por computadoras, para así eliminar la subjetividad e intentar alcanzar una prueba consistente para la clasificación morfológica de cabezas de espermatozoides humanos.

El análisis de las cabezas de espermatozoides se encuentra documentado en un manual de la Organización Mundial de la Salud, dedicado en su completitud al trabajo, manejo y experimentación de espermatozoides humanos (OMS, 2010b). Este documento detalla algunas formas que pueden presentar las cabezas de espermatozoides, aunque en este trabajo, el *dataset* utilizado solo tiene imágenes clasificadas en cinco formas distintas (normal; cónica; piriforme; pequeña y amorfá) y son estas las que los expertos han estado utilizando para clasificar imágenes

de muestras de pacientes (OMS, 2010b), y al mismo tiempo, son estas las que se busca clasificar de manera automática asistido por computadores.

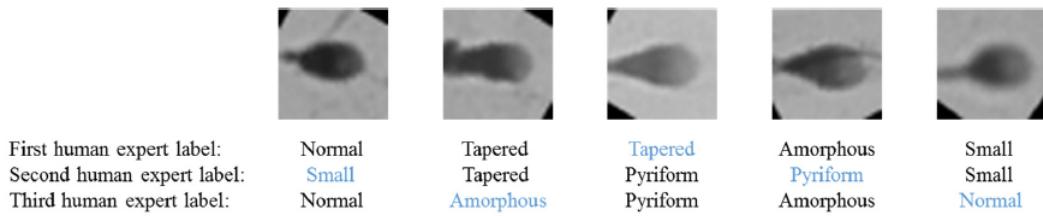


Figura 1.1: Ejemplo de clasificación morfológica de cabezas de espermatozoides humanos. En esta imagen, se observa distintas cabezas del dataset *SCIAN-MorphoSpermGS*, las cuales fueron clasificadas de manera independiente, por tres expertos distintos (Chang et al., 2017b)

1.3 SOLUCIÓN PROPUESTA

A partir de los problemas expuestos en el apartado anterior, es donde nace la solución propuesta en este trabajo de investigación. La alternativa que se ha escogido para la realización de este proyecto consiste en técnicas basadas en *deep learning* aplicadas al ámbito biomédico.

Particularmente se usarán como herramientas, dos redes neuronales convolucionales, las que fueron escogidas a partir de una investigación en la literatura, basándose en experimentos que ya han utilizado este tipo de redes en un contexto de clasificación según la morfología de objetos en imágenes. Estas redes convolucionales escogidas, no han sido probadas anteriormente en la clasificación morfológica de cabezas de espermatozoides humanos, por lo que se espera mejorar los resultados con alguna de estas implementaciones.

1.3.1 Hipótesis

- Aplicar métodos basados en *deep learning* permitirá mejorar la precisión en la clasificación morfológica de cabezas de espermatozoides humanos.

1.4 OBJETIVOS Y ALCANCE DEL PROYECTO

1.4.1 Objetivo general

El objetivo general del proyecto es evaluar el uso de métodos basados en *deep learning*, para mejorar la precisión en la clasificación morfológica de cabezas de espermatozoides humanos.

1.4.2 Objetivos específicos

El cumplimiento del objetivo general debe estar medido de alguna forma, como pasos a seguir, criterios o medidas de aceptación, entre otros, los cuales en conjunto forman los objetivos específicos del trabajo actual. Estos son los siguientes.

- Entrenar dos modelos de *deep learning* para clasificar morfológicamente cabezas de espermatozoides, usando el *dataset* público **SCIAN-MorphoSpermGS**, y evaluar su rendimiento con métricas similares a las utilizadas en el estado del arte.
- Distinguir los mejores parámetros de entrenamiento (número de épocas, tamaño del lote, optimizador y tasa de aprendizaje) para la arquitectura original de los dos modelos de *deep learning* elegidos.
- Aplicar y determinar el impacto de aprendizaje por transferencia en el problema de clasificación morfológica de cabezas de espermatozoides, usando los mismos dos modelos elegidos.

1.4.3 Alcances de la propuesta

En los objetivos específicos mencionados anteriormente, se han escogido dos tipos de redes convolucionales a partir de la literatura. Estas serán adaptadas y refinadas para su uso en este problema de clasificación morfológica de cabezas de espermatozoides humanos. Sin embargo, este trabajo no comprende la construcción desde cero de una nueva red neuronal convolucional, ni tampoco ayudará en la construcción o mejora del conjunto de imágenes de espermatozoides público que se va a utilizar.

1.5 METODOLOGÍA Y HERRAMIENTAS UTILIZADAS

1.5.1 Metodología

En este trabajo se utiliza el método científico, en donde en un comienzo, se ha planteado la pregunta sobre si es posible mejorar el estado del arte para la clasificación de cabezas de espermatozoides, seguido de una investigación para encontrar herramientas capaces de resolver este problema, en donde se ha llegado a la hipótesis planteada, en donde se utilizarán tecnologías de *deep learning* para intentar responder a la pregunta inicial.

La serie de objetivos específicos planteados corresponde al experimento, el cual tendrá como fin, buscar si la hipótesis planteada es verdadera o falsa. A partir de esto, se decidirá qué tipo de trabajos futuros podrán plantearse acerca del contexto de análisis y clasificación de cabezas de espermatozoides.

Para la aplicación de los objetivos específicos planteados, se presenta a continuación las actividades realizadas en esta investigación. Cada una de estas actividades están acompañadas de una etapa de investigación, debido a que las decisiones tomadas deben estar respaldadas por la literatura.

Extender el dataset público SCIAN-MorphoSpermGS.

Para esta tarea de extender el *dataset* de trabajo, se usa la técnica de aumento de datos, la cual deja de ser un proceso trivial y repetitivo en el caso de una aplicación como el análisis morfológico de cabezas de espermatozoides humano. Esto es debido a la sutil diferencia entre los objetos una clase respecto de otra (poca variabilidad inter-clases). Por lo mismo, la transformación de una imagen particular debe ser de tal forma, que no se pierdan las características morfométricas de la cabeza de espermatozoide contenida en ella. A ello hay que añadir la complejidad adicional de manejar el desbalanceo de clases presente en el *dataset* original.

Adaptar al menos dos modelos existentes de deep learning

Se utilizan en esta investigación, dos redes convolucionales distintas para clasificar las cabezas de espermatozoides, las que corresponden a la red convolucional utilizada para clasificar glóbulos rojos, y la red convolucional profunda, *ResNet50*, las que deben pasar por un

proceso de adaptación de la arquitectura al contexto de una nueva aplicación y conjunto de datos. Por ejemplo, las entradas de ambas redes convolucionales deben ser adaptadas para utilizar las imágenes de espermatozoides de 35×35 píxeles, y también se debe reacondicionar las salidas de estas redes, para que puedan clasificar las cinco clases de cabezas de espermatozoides utilizadas en el *dataset*.

Afinar parámetros para ambas arquitectura de red escogidas

Durante el entrenamiento, toda red convolucional utiliza un optimizador para recalcular los pesos de las neuronas al finalizar una época. Para afinar esta función optimizadora, se deberá investigar en la literatura sobre cuales son más aptos para utilizar en este caso de cabezas de espermatozoides. Además es preciso afinar valores de parámetros como por ejemplo, el tamaño de lote (cantidad de imágenes) y el número de épocas que se utilizarán para los experimentos. Todo este afinamiento se realiza metódicamente, usando un subconjunto pequeño del *dataset* (conjunto de validación) y buscando minimizar el error tanto en el entrenamiento como en la validación.

Reutilizar aprendizaje de otros experimentos al aplicar aprendizaje por transferencia

Al mismo tiempo, en esta investigación se utiliza la técnica de aprendizaje por transferencia, para mejorar la clasificación morfológica de las cabezas de espermatozoides. En este apartado, se vuelven a probar las redes convolucionales escogidas, pero esta vez se utilizan pesos pre-entrenados, los cuales serán escogidos a partir de otros trabajos pasados publicados. Se buscarán trabajos de clasificación con características similares a la clasificación de las cabezas de espermatozoides y se pre-entrenaran las redes con los conjuntos de estos trabajos pasados para generar un conjunto de pesos pre-entrenados a utilizar en la clasificación morfológica de las cabezas de los espermatozoides.

Afinar la estructura de las redes al modificar su arquitectura, e intentar mejorar la clasificación

Con la intención de intentar mejorar aún más los resultados de la clasificación, se ha decidido modificar de la arquitectura de alguno de los modelos elegidos y así entrenar nuevamente. Para esto, se ha decidido congelar algunas capas de la red convolucional que haya obtenido los mejores resultados en la clasificación morfológica de cabezas de espermatozoides humanos. Congelar las capas significa que los pesos de las neuronas no se actualizarán en el entrenamiento, y es una técnica que se ha frecuentemente en el aprendizaje por transferencia.

Diseñar múltiples experimentos en los cuales evaluar los modelos elegidos

Los resultados de la investigación están separados en tres experimentos distintos, los que corresponden al entrenamiento los modelos convolucionales desde cero (con pesos iniciales aleatorios), entrenamiento de los modelos con aprendizaje por transferencia (con pesos pre-entrenados) y finalmente, entrenar el mejor modelo obtenido de los experimentos anteriores, pero esta vez con la estructura convolucional modificada (congelar algunas de sus capas).

Evaluar modelos de redes convolucionales adaptadas usando validación cruzada, y escoger las mejores métricas de acuerdo al estado del arte

Todo este experimento está envuelto en la técnica de la validación cruzada, para así darle robustez a los resultados que se obtengas una vez entrenadas las redes convolucionales. Al mismo tiempo, se deberán escoger métricas de evaluación para poder interpretar los resultados de la clasificación de las imágenes. Ahora bien, esta implementación debe considerar que las clases no están balanceadas, ya que existen más cabezas de espermatozoides de clase amorfas en el conjunto inicial, por lo que habrá que balancear las muestras al aplicar la validación cruzada.

Finalmente, los resultados obtenidos será comparados con el estado del arte, el cual corresponde a los obtenidos en el trabajo de Riordon et al. (2019), el cuál ha realizado una clasificación morfológica de cabezas de espermatozoides, utilizando la red neuronal convolucional *VGG16*.

1.5.2 Software y hardware requerido

El algoritmo correspondiente al trabajo actual, es construido en un *notebook* con 8 GB de memoria *RAM*, el cual también tiene un procesador Intel I5 3210M, con 2 núcleos físicos y una frecuencia básica de 2,50 GHz. La computadora además posee una *GPU* integrada Intel HD Graphics 4000 de 2GB, y otra dedicada, modelo Nvidia Geforce GT 650M de 1GB. el sistema operativo utilizado es la distribución de Linux gratuita, *Ubuntu 18.04 LTS*. El algoritmo como tal fue construido en **Python 3.6**, utilizando principalmente las siguientes librerías. El manejo de *Python* y sus librerías es realizado por el gestor **Anaconda**, el cuál ayuda a simplificar la instalación de estas debido a que se encarga de manejar las versiones de las librerías y sus dependencias de manera automática y sencilla para el usuario.

En conjunto con este computador, también se utiliza un servidor dedicado, con el mismo sistema operativo *Ubuntu 18.04*, por lo que la instalación de librerías sigue el mismo procedimiento explicado, y tiene una *GPU* de modelo *Titan RTX* con 24GB de memoria.

- **Tensorflow y Keras** para la construcción y entrenamiento de la red convolucional, además del manejo de imágenes
- **Skicit** para análisis de datos
- **Numpy** para manejo de arreglos y cálculos matemáticos
- **Matplotlib** para generar gráficos asociados al trabajo
- **PIL** para leer archivos de imágenes.

1.6 ORGANIZACIÓN DEL DOCUMENTO

El documento está compuesto en cinco capítulos, contando esta introducción. Los capítulos siguientes se detallan a continuación.

El Capítulo 2 comprende el Marco Teórico relacionado a este trabajo, en donde se explicarán las bases del *deep learning*, las redes neuronales, las redes neuronales convolucionales, además de presentar definiciones acerca de los espermatozoides y el manual de análisis sobre estos, propuestos por la Organización Mundial de la Salud.

Por otro lado, en el Capítulo 3, se plantea el estado del arte para este problema, del cual se compararán los resultados del experimento actual, y una revisión de la literatura de cómo y por qué se escogió la red convolucional utilizada en este experimento.

Seguido de esto, en el Capítulo 4, se presenta la investigación realizada para escoger las redes neuronales convolucionales a utilizar en el problema actual, en donde se buscaron redes que se hayan usado en problemas similares al de clasificación morfológica de cabezas de espermatozoides.

El Capítulo 5 es en donde se muestra la investigación realizada para completar los objetivos específicos planteados. La aplicación de cada uno de estos objetivos, requiere tomar ciertas decisiones que deben estar respaldadas por la literatura, para así asegurar la robustez de los resultados.

El Capítulo 6 muestra los resultados obtenidos en el desarrollo de los experimentos definidos en el capítulo anterior, explicando los protocolos que se hicieron para llegar a estos, y comentando las tablas y gráficos obtenidos en el proceso de clasificación morfológica.

Finalmente, en el Capítulo 7 presenta las conclusiones del trabajo indicando el grado de logro obtenido en el trabajo, y comparándolo con los objetivos planteados en el inicio de este trabajo, y estableciendo trabajos futuros.

CAPÍTULO 2. MARCO TEÓRICO

En este capítulo se muestran los principales conceptos relacionados a los elementos de estudio, como los espermatozoides humanos y el estudio de su morfología, así como también se habla acerca de las herramientas utilizadas en el desarrollo de este trabajo.

2.1 ESPERMATOZOIDES HUMANOS

El espermatozoide es una de las células más especializadas que existen en la biología humana, debido a que es sumamente diferente al resto de células, al poseer una cabeza y una cola que le da motilidad a esta. A pesar de esto, este gameto pasa por distintas etapas de formación para alcanzar esta forma (González et al., 2018).

Espermatogénesis es el proceso conocido para la formación de las células de espermatozoides, la cual se compone de tres etapas distintas llamadas **espermatoцитogénesis, meiosis y espermiogénesis**. Cada una de estas etapas entrega como resultado la formación de espermatoцитos primarios, espermatoцитos secundarios y espermatozoides respectivamente. La diferenciación y formación del espermatozoide se genera en la última etapa, la que se puede ver en la Figura 2.1, y consiste en la condensación del núcleo, y la formación del cuello y el flagelo a partir del centriolo y las mitocondrias.

Un espermatozoide se encuentra formado, en primer lugar, por su cabeza, la cual está formada por una membrana plasmática, constituida por una doble capa de lípidos, separando el interior y exterior de la célula. La cabeza contiene en su extremo el acrosoma, cuya función es ayudar al gameto masculino a penetrar las paredes del óvulo, y también el núcleo de este, que contiene la mitad del material genético del hombre. En el otro extremo de la cabeza del espermatozoide se encuentra la parte media (cuello), que está formado en su inicio por el centriolo y un alto número de mitocondrias. Esta última parte es la encargada de dar energía al flagelo, que corresponde a la cola del espermatozoide, el cual permite la motilidad del gameto masculino, para que este se desplace de forma independiente en el tracto vaginal (Barría & Ramírez, 2014). El resumen de las partes se puede observar en la Figura 2.1, especialmente en las etapas *H* e *I* de la formación del espermatozoide que se ve en la imagen.

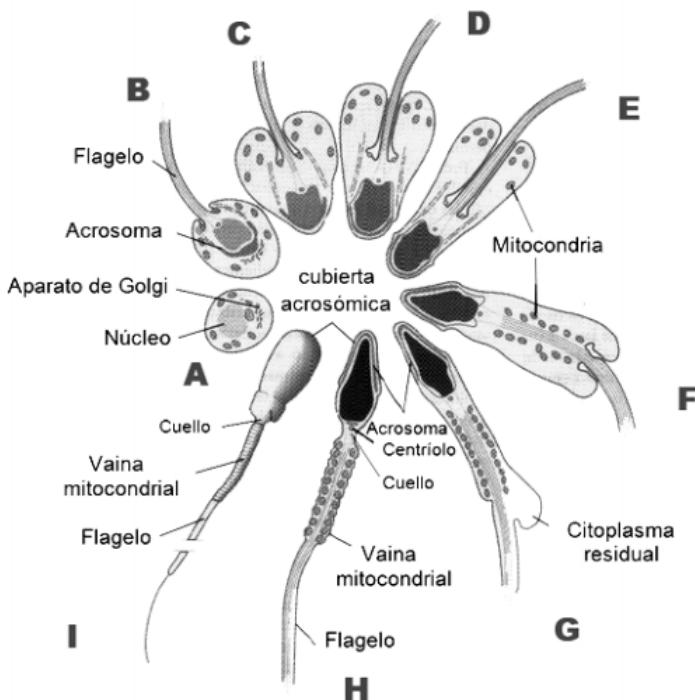


Figura 2.1: Proceso de espermiogénesis y estructura de un espermatozoide (Barría & Ramírez, 2014). El orden de la formación está dado por el orden de las letras, y en las últimas etapas se puede observar bien las partes como el cuello, acrosoma, núcleo y flagelo del espermatozoide.

2.2 ANÁLISIS DE ESPERMATOZOIDES

2.2.1 Manual para procedimiento de análisis

El análisis de espermatozoides está documentado en la quinta edición del manual de laboratorio para la evaluación y procesamiento de semen humano, publicado por la **Organización Mundial de la Salud** y es en este documento donde se detallan procedimientos como la captura de las muestras, mantención de estas, procesamiento de los datos en crudo, e incluso algunas pruebas que se pueden realizar en un laboratorio sobre estos, con un nivel de detalle bien fundamentado con trabajos anteriores. Por ejemplo, en la recolección de espermatozoides sobre pacientes, se habla de que la calidad de los espermatozoides puede cambiar disminuir si el paciente realizó su muestra cerca de los laboratorios, en comparación a hacerlo en su hogar (OMS, 2010b).

El segundo capítulo del manual de la OMS (2010b) habla acerca de los proced-

imientos estándar que se realizan al momento de trabajar e investigar con este tipo de células masculinas. A continuación se habla de algunos procedimientos que se realizan para el estudio de la forma de las cabezas de espermatozoides.

El análisis inicial corresponde a una observación macroscópica, con una magnificación al microscopio de $\times 100$, en donde se puede hacer una observación el aglutinamiento de los espermatozoides, y también de algunos elementos que no pertenecen a este tipo de células, como pueden ser restos de leucocitos, los que son descartados posteriormente en el análisis. El aglutinamiento es la forma en cómo los espermatozoides están presentes en una muestra, donde generalmente se encuentran pegados entre si en distintas partes de sus estructuras. En la Figura 2.2 se puede observar distintos niveles de aglutinamiento que pueden existir en una muestra.

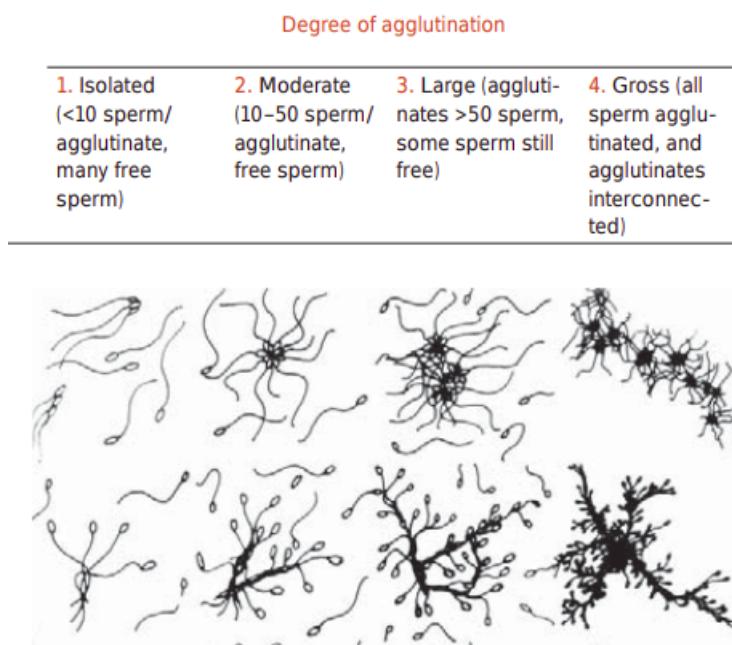


Figura 2.2: Ejemplo de aglutinamiento en muestras de espermatozoides (OMS, 2010b). El nivel de aglutinamiento es que tan pegadas entre sí se encuentran los espermatozoides, y los niveles de estos aumentan de izquierda a derecha en la figura.

El siguiente análisis (microscópico), trata de una observación de la muestra con una magnificación al microscopio de $\times 200$, y es en esta donde se realizan los análisis estándar y principales sobre los espermatozoides de un paciente. Este análisis estándar, se realiza previo a observar la morfología de cabezas de espermatozoides, y el manual habla acerca del análisis de la motilidad y vitalidad de este tipos de células.

El análisis de la motilidad tiene que ver con observar que tan capaz es un espermatozoide de desplazarse por si solo, lo cual está asociado a la forma de la cola de estas. Por otro lado, la vitalidad habla de la cantidad de espermatozoides que son capaces de fecundar un óvulo, y su estudio se basa en determinar si las muestras contienen o no su membrana celular

a partir de unas pruebas. El número de espermatozoides "vivos" en general es mayor a los que no pueden moverse (OMS, 2010b, p. 26).

2.2.2 Análisis morfológico de espermatozoides

El análisis morfológico de espermatozoides es una herramienta bastante importante al momento de identificar el nivel de fertilidad de un paciente. El trabajo de Eggert-Kruse et al. (1996) realizó un estudio sobre la tasa de embarazo versus la cantidad de espermatozoides con forma normal, el cual se llevó a cabo en Alemania y con muestras de 89 parejas escogidas de manera aleatoria. Dentro de los resultados, se observa que para las personas con un porcentaje de espermatozoides con forma normal mayor al 14%, se tiene una tasa de embarazo del 34,3% en sus parejas en los siguientes 12 meses, mientras que para menos del 14% de espermatozoides normales, la tasa de embarazo decreció hasta un 11,1%. Cabe destacar que del estudio se excluyeron resultados donde los hombres padecían de alguna infección o incluso los casos en que era la mujer quien presentaba alguna anomalía que produjera un incremento de la infertilidad. Esta investigación usaba la clasificación "normal" según ciertos criterios descritos por Kruger et al. (1988).

Actualmente, el manual de la Organización Mundial de la Salud, describe de manera resumida algunos pasos y métricas a utilizar al momento e observar una imagen de una célula de espermatozoide. Estas métricas también nacen a partir de las descripciones iniciales de Kruger et al. (1988), y se presentan distintas mediciones para identificar una forma normal, tanto para la cola, cuello y cabeza del espermatozoide.

Específicamente hablando de la forma de la cabeza de un espermatozoide, esta puede tener varios defectos, los que se pueden observar en la Figura 2.3. El manual destaca que, para considerar una forma normal de cabeza de espermatozoide, se considera que la cabeza debe ser de forma ovalada, además de tener un acrosoma bien definido que ocupe entre un 40% y 70% de la cabeza.(OMS, 2010b, p. 69)

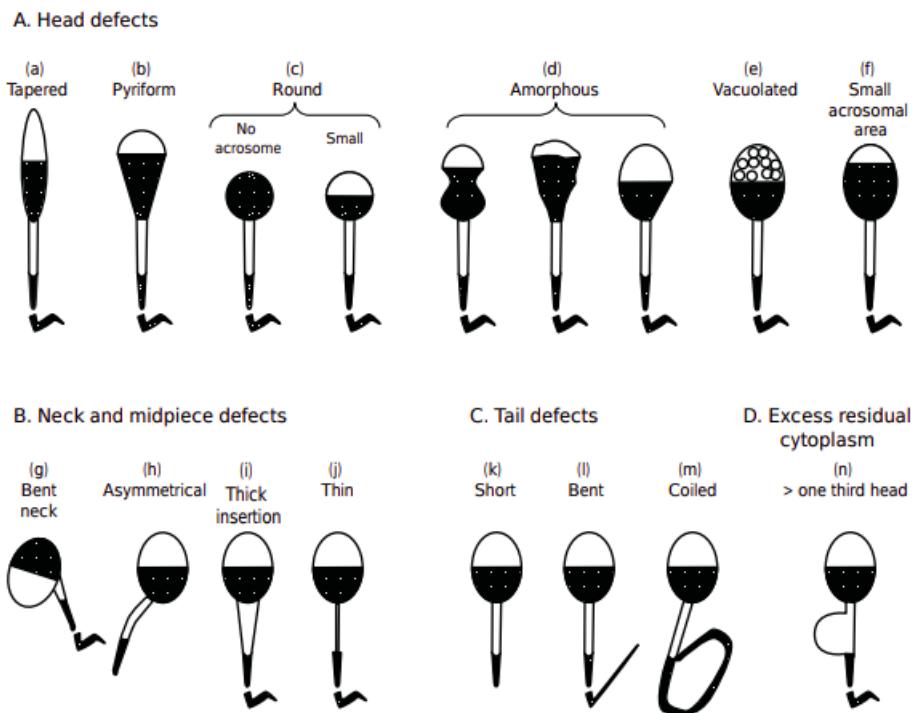


Figura 2.3: Tipos de formas de espermatozoides (OMS, 2010b). En la primera fila se muestra alteraciones en la cabeza del espermatozoide, y en la segunda alteraciones en otras partes de esta célula.

A continuación se presentaran temáticas relacionadas a la herramientas utilizadas en este experimento, para resolver al problema de la clasificación de cabezas de espermatozoides según su morfología.

2.3 APRENDIZAJE DE MÁQUINAS

Arthur Lee Samuel, pionero de los juegos informáticos y la inteligencia artificial, habla sobre el aprendizaje de máquinas (o *Machine Learning*) como un proceso realizado por computadores, el cual, es el equivalente al proceso de aprendizaje que realizan los animales y los seres humanos (Samuel, 1959). A partir de lo anterior, otros autores han parafraseado los dichos de Arthur Samuel, y han llegado a definir el aprendizaje de máquinas, como la habilidad que tienen los computadores para resolver problemas, sin decirles explícitamente qué pasos realizar (Koza et al., 1996, p. 153).

Por ejemplo, existe el algoritmo de clasificación y regresión llamado "K vecinos más cercanos" (*K-nearest neighbors*), el que estima el valor de la probabilidad o directamente la

probabilidad a posteriori de que un elemento pertenezca a cierta clase, a partir de los elementos ya existentes dentro del espacio de estudio. Esto quiere decir que, el aprendizaje se va realizando cada vez que un nuevo elemento entra en el espacio (Harrison, 2019).

Por otro lado, también se tiene el ejemplo de las redes neuronales, dentro del concepto de aprendizaje de máquinas, y es este el cual se habla a continuación, ya que es la base de las redes neuronales convolucionales, que es la herramienta utilizada en este experimento.

La unidad mínima de las redes neuronales, consiste en una neurona capaz de procesar la información que recibe, y se encuentra conectada con otras similares a esta (Schmidhuber, 2015). Una neurona recibe como entrada información del ambiente o desde otras neuronas, y al mismo tiempo, desprende el resultado de su procesamiento a otras neuronas o a la salida de la red.

Dependiendo de la cantidad de neuronas, la red se va volviendo cada vez más profunda y más compleja, por lo que su construcción y su uso depende del problema que se intentará resolver con esta (Schmidhuber, 2015). En la Figura 2.4, se puede observar como cada neurona (representadas como un círculo), se encuentran conectadas con otras de manera ordenadas. Las capas que no se encuentran en la entrada o salida de la red se conocen como **capas ocultas**, y la cantidad de estas son las que determinan la profundidad de la red neuronal. La capa de entrada recibe información cruda desde el ambiente, mientras que la capa de salida devuelve el resultado del procesamiento de la entrada, el que dependerá del problema.

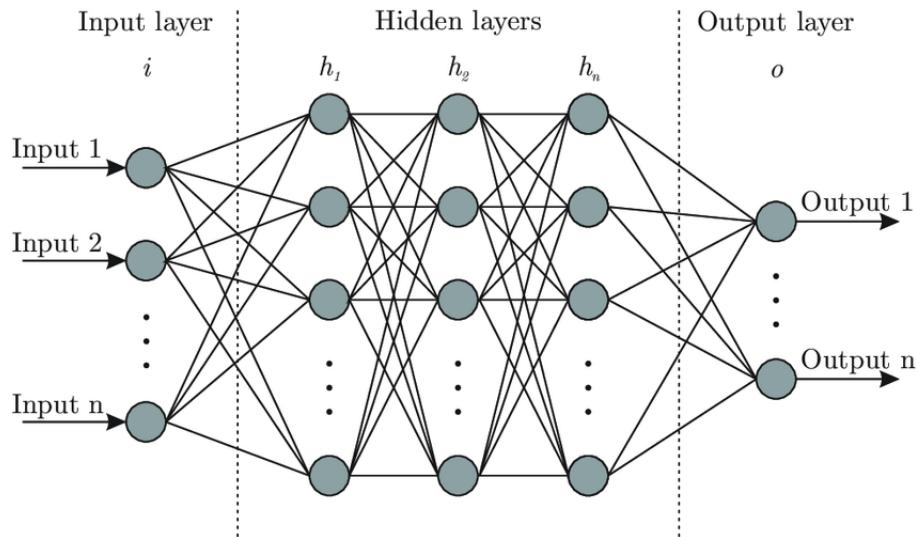


Figura 2.4: Ejemplo de arquitectura de una red neuronal (Bre et al., 2017). A la izquierda se muestra la capa de entrada, que recibe información el entorno, mientras que la siguen las capas ocultas, que se encuentran completamente conectadas entre sí, para finalizar con la capa de salida, que entrega el resultado del procesamiento de toda la red.

El entrenamiento corresponde a un proceso iterativo, en donde cada neurona comienza con un peso W_n , el cual inicialmente puede ser aleatorio, y en cada iteración, estos

se van actualizando para intentar alcanzar el error minino en la salida. Si se tomara una sola capa de una red neuronal, se tendría una imagen parecida a la Figura 2.5. Las neuronas y sus pesos son procesados en la función de entrada, en donde se multiplica la entrada de la neurona con su peso, y todas estas se suman para formar una combinación lineal. Luego de esto, la función de activación es aquella que entregará la salida de la red neuronal, por lo que actúa como un umbral. La función de entrada es lineal, así que la función de activación es la encargada de generalizar la salida para distintos problemas. En el caso de una regresión lineal, la función de activación sería de este mismo tipo, pero si se intenta resolver problemas categóricos, como clasificación binaria, entonces la función de activación tendría que ser una función que retorne solo uno de dos valores, dependiendo de la entrada.

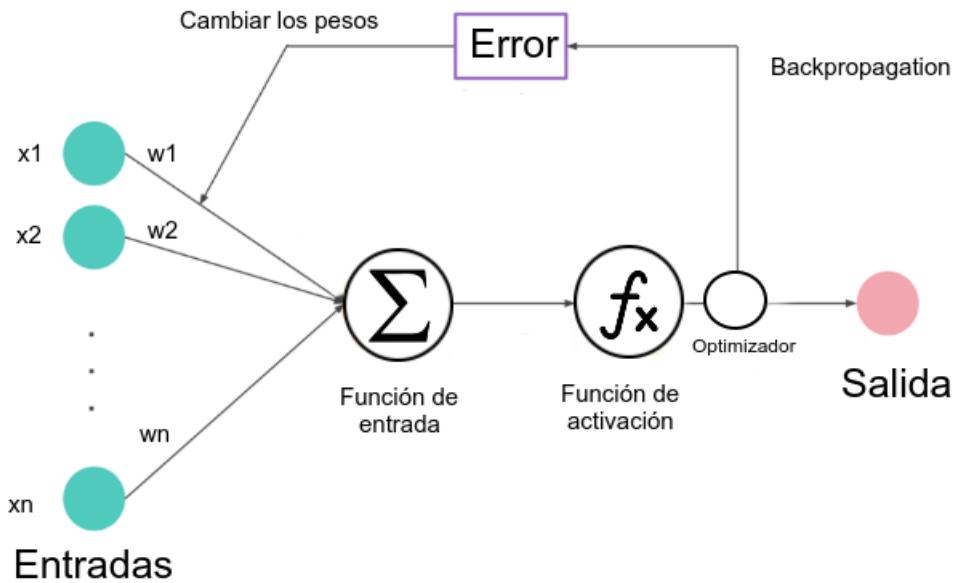


Figura 2.5: Entrenando una red neuronal. Cada red X_n tiene un peso W_n asociado, que son utilizados por la función de entrada y de activación. La función de activación es la encargada de entregar la salida de la red, por lo que de esta, se calcula el error utilizando funciones optimizadoras, y se recalculan los pesos W_n de las neuronas.

Fuente: Elaboración propia, 2020.

Durante el entrenamiento de los modelos neuronales, es posible que al ajustar los pesos, estos comiencen a buscar un mínimo local en vez de un mínimo global. Una solución a este inconveniente son las funciones optimizadoras, las que son algoritmos encargados de ajustar los pesos de las neuronas, para así intentar disminuir el error y alcanzar el mínimo global de este. Estos algoritmos utilizan un parámetro llamado tasa de aprendizaje o *momentum*, los que indican que tan grande será el ajuste de los pesos de las neuronas de la red Rocca (2019).

A pesar de la existencia de estos algoritmos, el mal uso de los valores como la tasa de aprendizaje, pueden perjudicar el entrenamiento. Por ejemplo, si la tasa de aprendizaje es

alta, el ajuste de los pesos puede llevar a que el error se exceda del mínimo global (*overshoot*), mientras que en el caso contrario, una tasa de aprendizaje muy baja hace que la búsqueda del mínimo global se vuelva lenta y tome más iteraciones llegar a esta.

Al final de una iteración, se calcula el error de la red, y se vuelven a calcular los pesos de las neuronas pasadas de modo de minimizar este mismo en las siguientes iteraciones. Si se vuelve a observar la Figura 2.5, el paso de recalcular los pesos de cada neurona se realiza a lo largo de todas las capas ocultas de la red, a través de un método de que conoce como **back propagation**.

El aprendizaje puede ser supervisado, o no supervisado, por lo que a continuación se explicará sus funcionalidades, especialmente para el aprendizaje supervisado, ya que es el utilizado durante este experimento.

2.3.1 Aprendizaje supervisado

Este tipo de aprendizaje se da cuando se ajusta la red para predecir salidas específicas, en respuesta a entradas específicas. Durante el entrenamiento, los datos se encuentran ordenados de a pares, indicando así cada entrada x y su correspondiente respuesta y . Es en esta etapa donde la red intenta buscar la función $f(x)$, para que cada entrada x_i , de como respuesta su correspondiente salida y_i (Reed & MarksII, 1999, p. 5). Luego de entrenar la red, esta utiliza esta función $f(x)$ sobre datos nuevos, para ver cual es la respuesta que tiene la red sobre información que no ha visto con anterioridad. En un caso "óptimo", la red habrá podido generalizar las características de los datos de entrada durante el entrenamiento, y así obtener la predicción correcta para todos los datos nuevos.

El ejemplo más simple de este tipo de aprendizaje, es una red neuronal que intenta predecir, a partir de una imagen, si el animal que esta en esta es un perro o un gato. Esto corresponde a un problema de clasificación, en donde durante el entrenamiento de la red, se le proporcionarán pares de imágenes y una etiqueta para la misma (gato o perro) y esta buscará generalizar las características correspondiente para ambas especies de animales. Finalmente, a la red se le mostrarán nuevas imágenes de gatos y perros que no haya visto el entrenamiento, e intentara clasificar si el animal presente en ellas es un perro o un gato.

Durante este tipo de aprendizaje, se sabe con cierta seguridad lo que está haciendo la red, como puede ser la clasificación de perros y gatos mencionados. Por lo mismo, es imposible que durante el entrenamiento, aparezcan clases nuevas a las existentes en las etiquetas, es decir, si se vuelve al ejemplo anterior, es imposible que la red clasifique algunas imágenes con animales que no sean perros ni gatos.

2.3.2 Aprendizaje no supervisado

En el aprendizaje no supervisado, a diferencia de su contraparte, la información no viene etiquetada o clasificada con anterioridad, si no que es el propio algoritmo, el encargado de encontrar la estructura de la información que va recibiendo. Es debido a esto, que este tipo de aprendizaje suele ser más complejo que el supervisado, además de que sin las etiquetas, es difícil saber si se están logrando los resultados deseados, por lo que es necesario consultar con expertos asociados al problema de estudio. A partir de la falta de las etiquetas predefinidas, el fin del aprendizaje no supervisado, es identificar cómo los datos de entrada se encuentran agrupados entre sí, según sus características (conocido como *clustering*) o bien, identificar cómo la información se encuentra distribuida en el espacio (conocido como *density estimation*) (Mishra, 2017).

Un ejemplo que sirve para diferenciarlo del aprendizaje supervisado, consiste en el agrupamiento de fotografías de distintos animales y sin una etiqueta predefinida. Durante el entrenamiento, la red debería de comenzar a aprender cuales características corresponden a cada especie y agruparlas, para finalmente haber separado aquellas que corresponden a imágenes de perros, gatos, conejos, etc.

En consecuencia, si a mitad de entrenamiento, se comienzan a introducir imágenes de nuevas especies de animales al modelo neuronal, este debe ser capaz de comenzar a generalizar las características de esta nueva especie y así poder agruparlas como el resto de especies del conjunto de datos.

Al mismo tiempo, existe otro tipo de aprendizaje que se considera como no supervisado, el cual es llamado aprendizaje por refuerzo. Este último consiste en entregarle reglas y recompensas a un agente, a medida que toma decisiones, y este último va aprendiendo que realizar a medida que va recibiendo las recompensas.

Un agente no es más que una entidad capaz de recibir información de su entorno, y tomar decisiones a partir de aquello. Durante el aprendizaje por refuerzo, existen reglas pre-establecidas, en conjunto con un número que indica la "recompensa" o el "castigo" que irá recibiendo el agente. Según el problema, aquellas acciones que ayuden a solucionar un problema, servirán de "recompensa" para el agente, mientras que en el caso contrario, servirá como "castigo". A medida que el agente va tomando las decisiones, se le irá recompensando o castigando según sus decisiones, para que, en el caso óptimo, aprender cuales son las tareas correctas que deba hacer en el entorno, y así solucionar a su problema Bakker (2002).

Este último tipo de aprendizaje es comúnmente utilizado en áreas como la robótica, como puede ser la exploración de terrenos, o incluso en los videojuegos (garychl, 2018).

2.4 APRENDIZAJE PROFUNDO

El aprendizaje profundo, o *Deep Learning*, es una parte del *Machine Learning*, que se basa principalmente en utilizar múltiples capas ordenadas, para así procesar la información cruda que reciben como entrada, con el objetivo de extraer características de alto nivel, clasificar la información, o reconocer patrones en esta (Deng & Yu, 2014, p. 217).

Un ejemplo del análisis y extracción de características de alto nivel, puede ser el reconocimiento de patrones en imágenes, lo que se puede utilizar para identificar rostros humanos, o incluso identificar mapas de atención, que consiste en identificar los lugares donde el ojo humano presta más atención dentro de una imagen (Shen et al., 2012).

En el aprendizaje de máquinas, existe un "cuello de botella" durante el entrenamiento para algoritmos poco profundos, y consiste en la cantidad de datos y los resultados de estos algoritmos. Existe un punto en que los resultados de estos algoritmos, frente a un problema, no mejoran a pesar de alimentar a los modelos con más y más datos. En contraste, los algoritmos profundos logran utilizar la gran cantidad de datos que se les proporciona, además de la capacidad de cómputo de los computadores modernos, para obtener mejores resultados y resolver problemas mucho más complejos (Brownlee, 2019c).

Algunos ejemplos de estos algoritmos profundos, son algunas herramientas que Google utiliza para proporcionar los servicios que se ocupan diariamente por personas alrededor de todo el mundo. Para el buscador de imágenes, es importante identificar dentro de las fotografías que existen en los sitios web, que el contenido de estas son los objetos que efectivamente el usuario está buscando. De la misma forma, también es posible identificar letras, números y palabras dentro de las imágenes. Este tipo de tareas son un ejemplo claro cuando se refiere a identificar características de alto nivel en la información que se le entregan a este tipo de algoritmos. (Dean, 2016)

Volviendo al contexto de este trabajo, las redes convolucionales son un ejemplo claro del aprendizaje profundo, debido a que su principal característica corresponde a la identificación de distintas características en los datos de entrada, en donde esta se especializa principalmente en el análisis de imágenes. Este tipo de redes trabaja con un gran número de neuronas, pero al mismo tiempo se encarga de disminuir la dimensionalidad de la entrada a medida que trabaja sobre esta, así como también, compartir información entre las neuronas vecinas. Más adelante se entrará en detalle acerca de este tipo de redes (Angluin et al., 2016).

Este tipo de aprendizaje sigue estando inmerso en los tipos de aprendizaje anteriormente definidos, los cuales son los supervisados y no supervisados. Las redes neuronales convolucionales trabajan de manera supervisada, por lo que el objetivo sigue siendo,

disminuir el error durante múltiples iteraciones de entrenamiento. A continuación se detalla un poco acerca de esta función de error.

2.4.1 Función de costo

En los problemas de optimización, lo que se busca es encontrar el máximo o el mínimo de una función objetivo. Para el contexto de las redes neuronales, que se ha presentado en las secciones anteriores, la función objetivo corresponde a encontrar el error mínimo en el entrenamiento, luego de ir calculando los pesos de la red al completar una iteración. Para el caso de minimizar la función objetivo, esta suele llamarse **función de costo** o **función de pérdida (loss function)** (Goodfellow et al., 2016, p. 79).

El cómo se calcula este error es algo que depende de la naturaleza del problema, en donde, en el caso de problemas de regresión, la forma más usada de calcular el error es usando la función del error medio absoluto, que se ve en la Ecuación 2.1, el que consiste en la sumatoria del valor absoluto de los errores, dividido por los n valores de entrada del problema de optimización (Willmott & Matsuura, 2005).

En el contexto de una red neuronal para realizar una clasificación, no se puede usar la misma función para determinar el error, ya que ahora se está trabajando con probabilidad de pertenencia de una clase. Para este tipo de problemas se usa la entropía cruzada como función de costo, la que se conoce como una medida de la diferencia entre dos distribuciones de probabilidad para una variable aleatoria dada o un conjunto de eventos, y esta permite medir el error entre dos distribuciones de probabilidad (Brownlee, 2019b).

$$EAM = \frac{\sum_{i=1}^n |e_i|}{n} \quad (2.1)$$

Cual sea la función utilizada para obtener el error, al finalizar una iteración del entrenamiento, se utilizará la técnica de *back propagation* para traspasar el error a todas las capas de la red, y así actualizar todos los pesos de la red neuronal, con el objetivo de volver a entrenar e intentar encontrar el mínimo de la función de costo.

2.4.2 Optimizadores

Los optimizadores son funciones encargadas de actualizar los pesos de una red neuronal al finalizar una iteración de entrenamiento. Existe varias implementaciones que ayudan

a resolver este problema de encontrar el mínimo de la función de pérdida, como pueden ser los algoritmos *SGD*; *RMSprop*; *Adam*; *Adadelta*; *Adagrad* y *Adamax* (Chollet et al., 2015), pero a continuación se presentan los utilizados durante los experimentos de este trabajo de investigación.

Gradiente descendente estocástico (SGD)

Antes de hablar del gradiente descendente estocástico, se hace mención al método el cual está basado, que se llama gradiente descendente. El gradiente descendente es un método para encontrar el mínimo de una función, el cual utiliza el gradiente (pendiente) de un punto al azar inicial de la función, para así, como dice su nombre, descender a lo largo del gradiente hasta llegar al punto mínimo de esta. El largo del paso al momento de ir descendiendo es lo que se conoce en estos métodos como la tasa de aprendizaje, en donde un valor pequeño puede volver lento este proceso, mientras que un valor muy alto puede hacer que al descender en la función de costo, se salte el mínimo que se desea encontrar, y el cálculo de los pesos entre en un ciclo en donde al final de cada iteración, nuevamente se llega a pasar del punto mínimo de la función(*overshoot*).

El método anterior tiene un problema, el cual consiste en que el cálculo del gradiente se realiza en cada punto, por lo que hace que este método sea costoso al momento de ejecutar una computadora, además de puede llegar a ser lenta la convergencia al mínimo global (Srinivasan, 2019).

Por otro lado, El método del gradiente descendente estocástico busca mejorar este ultimo inconveniente, ya que no calcula el gradiente en todos los puntos, si no que realiza un estimado al calcular el gradiente desde un subconjunto del total de la función. En comparación, el *SGD* ayuda bastante a converger en mejores mínimos locales, pero al mismo tiempo, puede que sobrepase el mínimo global de la función objetivo, como se mencionó con anterioridad con el problema del *overshooting* (Ruder, 2017).

Optimizaciones de SGD: Momentum

El método de *SGD* suele presentar algunos problemas al intentar moverse a través de la función. En algunas partes de la función objetivo, la convergencia hacia un mínimo local o global puede volverse lento, como se ve en la imagen de la izquierda de la Figura 2.6, en donde el paso del gradiente es pequeño en áreas donde la función se encuentra más aplanaada.

El *momentum* es un método que acelera la convergencia en una dirección al momento del cálculo del paso tomar a lo largo de la función de pérdida, como se puede observar en la imagen de la derecha de la Figura 2.6. En palabras sencillas, este método es comparable

a dejar caer una bola a través de una rampa, en donde al descender esta va acumulando *momentum* y convergiendo cada vez más rápido hacia el fondo de esta (Ruder, 2017).

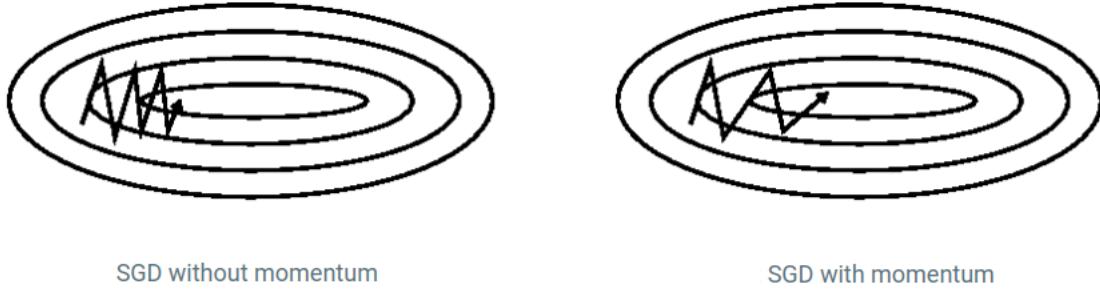


Figura 2.6: Ejemplo *SGD* con y sin *momentum* (Ruder, 2017). La flecha indica la dirección en que va apuntando el gradiente, en donde con *momentum*, se logra en menos iteraciones llegar al centro, que es el mínimo que se está buscando.

Optimizaciones de SGD: Nesterov

El *momentum* trae otro problema en su implementación. Al acelerar demasiado el paso del gradiente a lo largo de la función objetivo, es posible pasarse del mínimo local o global. El gradiente descendente de *Nesterov* busca solventar esta falencia del *momentum*, al calcular de manera prematura, dónde aproximadamente estará el siguiente punto de la función a posicionarse, estar luego de dar el paso con el gradiente descendente.

Según la Figura 2.7, se puede observar un ejemplo de la diferencia entre un paso hacia el mínimo global, utilizando *SGD* con y sin *Nesterov*, en donde se ve la característica de calcular un paso aproximado, y calcular la corrección de este, y así intentar llegar al mínimo de la función de costo. Esta mejora es la más utilizada cuando ese escoge la función *SGD* como optimizador.

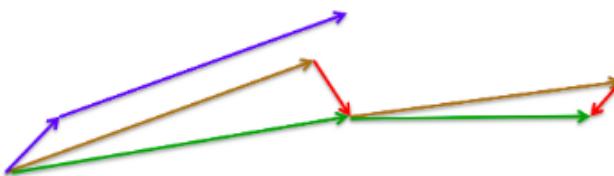


Figura 2.7: Ejemplo de la aplicación del método de *Nesterov* (Ruder, 2017). Las primeras flechas azules corresponden a dos pasos acelerados con *momentum*. Con *Nesterov*, primero se tiene un gran salto en dirección del gradiente en el punto actual (flecha de color café), en el cual se vuelve a calcular el gradiente y se obtiene una corrección del paso (flecha de color rojo).

Estimación de momentum adaptativo: Adam

La *Estimación de momentum adaptativo*, o *Adam*, es un método propuesto por Kingma & Ba (2014), el cual consiste en calcular tasas de aprendizajes ajustados a cada uno de los pesos de la red. La tasa de aprendizaje es calculada a través de los promedios de las magnitudes de gradientes pasados. Es debido a esta característica que este optimizador pertenece al tipo de optimizadores adaptativos, a diferencia del optimizador de gradiente descendente estocástico del que se ha hablado.

El cálculo utiliza tres parámetros, considerados como β_1 , β_2 y ϵ , en donde los dos primeros, son un factor de olvido para diferenciar cuantos gradientes pasados se utilizarán en el cálculo de la tasa de aprendizaje, mientras que el tercero corresponde a un factor de corrección para evitar que el método falle al dividir por cero en algunos cálculos de los nuevos pesos (Ruder, 2017).

El uso de este algoritmo optimizador generalmente es fácil de usar, debido a que, los valores por defecto de los parámetros de β_1 , β_2 y ϵ suelen funcionar en la mayoría de los casos Kingma & Ba (2014), en comparación al caso anterior del gradiente descendente estocástico, ya que al ir incluyendo elementos como *momentum* o *Nesterov* al método, significa entregar un nuevo parámetro de entrada a la función del gradiente descendente, lo que puede resultar costoso al intentar probar distintas instancias del método optimizador.

El algoritmo propuesto por Kingma & Ba (2014) recomienda el uso de ciertos valores para estos parámetros mencionados, los cuales han sido utilizados en la implementación de este método en librerías como *Caffe* o *Keras* para *Python*.

¿Cuál optimizador es mejor?

En redes convolucionales, el trabajo de Kingma & Ba (2014) realizó una comparativa de múltiples optimizadores, sobre el mismo conjunto de datos, y con la misma arquitectura convolucional. Los resultados de esta se pueden observar en la Figura 2.8, en gráficos de la función de costo a lo largo de las épocas de entrenamiento, y se puede ver que al llegar a 45 iteraciones, los optimizadores de *Adam* y *SGD* con *Nesterov*, son los que más se acercan a un mínimo global.

Sin embargo, se han mostrado casos en que el optimizador de *Adam* no generaliza bien las salidas en comparación con otros optimizadores, como se puede observar en la Figura 2.9, en el cuál se utilizó el mismo *dataset* de la Figura 2.8, pero con una red convolucional diferente y donde todos los optimizadores adaptativos, como *Adam*, no trabajan bien en comparación a

SGD, por ejemplo. Es por esto que la elección de un optimizador no es una tarea fácil, ya que depende del contexto y la complejidad del problema que se está trabajando (Wilson et al., 2018).

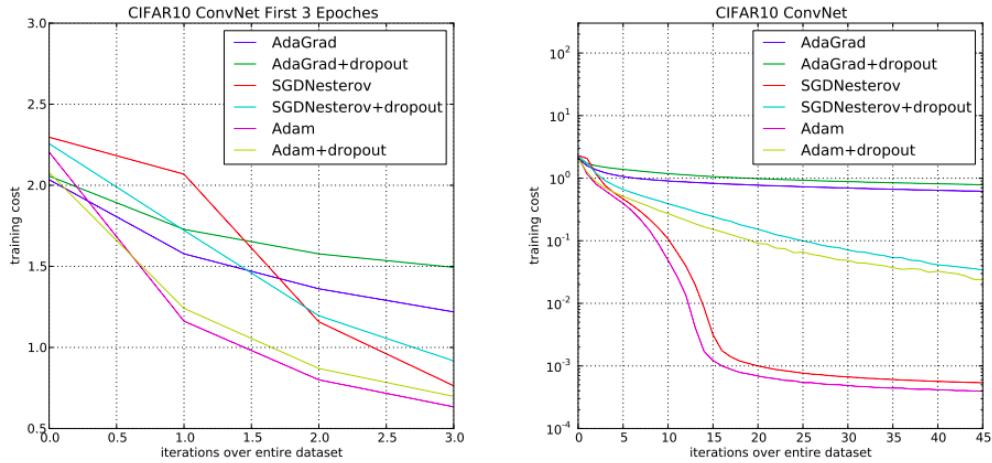
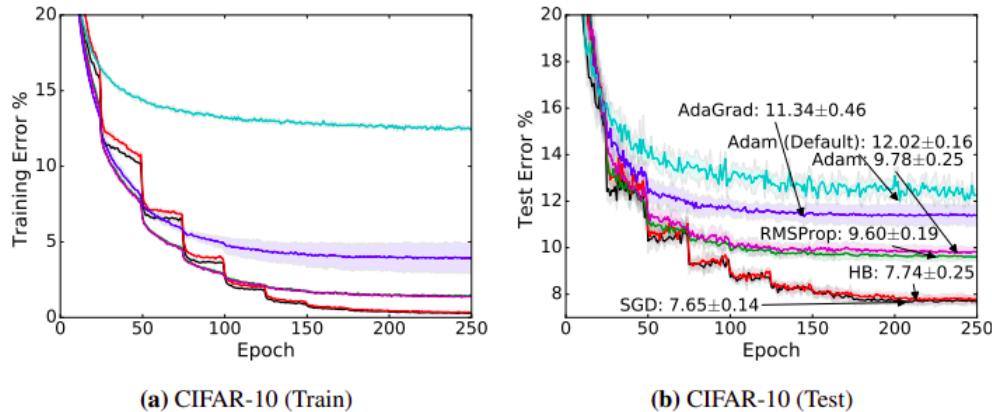


Figura 2.8: Comparación de error versus cantidad de épocas según optimizadores (Kingma & Ba, 2014). En el gráfico de la izquierda se muestra el error en las primeras tres épocas, mientras que en la derecha se muestran las primeras 45 épocas de entrenamiento.



2.4.3 Red neuronal convolucional

En la Figura 2.10 se puede observar reconocimiento de características, para identificar distintos tipos de imágenes, desde contornos (última fila), hasta características más

complejas como ojos y narices de una persona. Este nivel de abstracción es el utilizado para identificar el objeto que existe en una imagen.

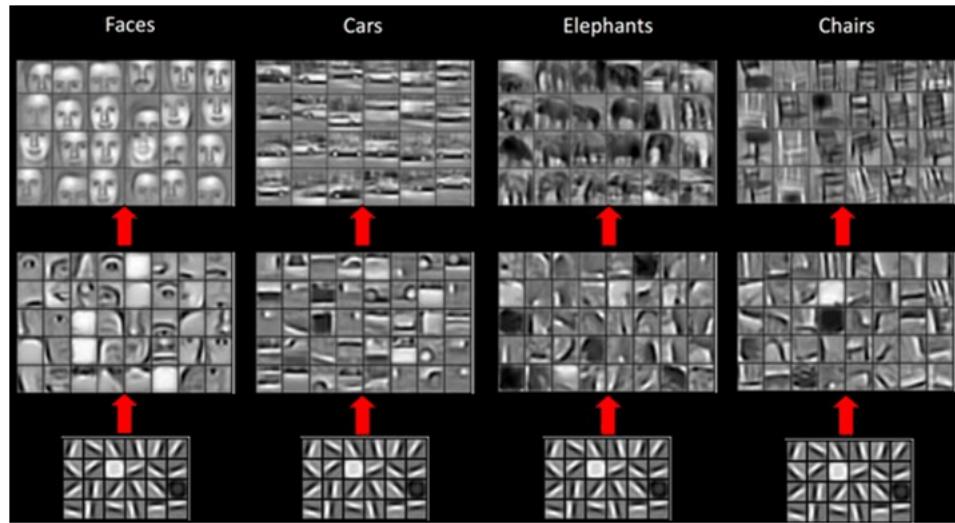


Figura 2.10: Características en reconocimiento de imágenes (Siegel et al., 2016). En la primera fila vemos los objetos originales de las imágenes, mientras que en la segunda y tercera fila se muestra la identificación de los bordes de estas imágenes.

Las redes neuronales convolucionales (o *CNN*), son un tipo de red neuronal dedicada especialmente al análisis de imágenes. Estas se caracterizan por tener un alto número de neuronas por capa, además de tener distintas etapas de procesamiento de una imagen, con el objetivo de poder identificar las características de estas (Angluin et al., 2016).

Una red neuronal estándar, como la que se ve en la Figura 2.4, se encuentra completamente conectada, por lo que por cada entrada existe una neurona en la primera capa de la red con su respectivo peso. Teniendo esto en cuenta, una imagen de 28×28 píxeles, en solo la primera capa tendría un total de 728 pesos, mientras que con un tamaño de 64×64 píxeles, se estarían alcanzando un total de 12.288 pesos. Esto no solo trae un problema de costo computacional, si no que al mismo tiempo puede llevar al *overfitting* de la red en el entrenamiento (O'Shea & Nash, 2015).

La red convolucional va disminuyendo la dimensionalidad de la entrada a través de sus capas, donde al mismo tiempo va identificando las características de la misma. En la Figura 2.11 se muestra una representación de una capa convolucional, la que tiene tres dimensiones, además de muchas neuronas que comparten la misma información de entrada. En la capa convolucional se ven múltiples filtros, y a lo largo, existen neuronas que están focalizadas para solo un sector de la imagen de entrada. Esto se conoce como conectividad local, y ayuda a disminuir problemas de *overfitting*, haciendo que la red generalice las características independiente del lugar y forma en que se presenten en la entrada, ya que existen un gran numero

de neuronas que comparten información parecida sobre ciertas regiones de la entrada (Angluin et al., 2016).

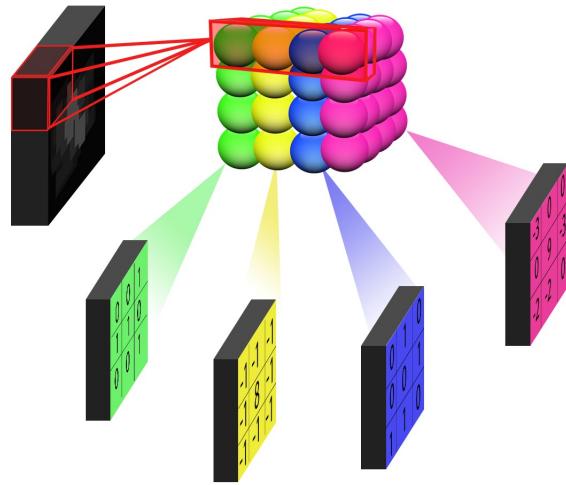


Figura 2.11: Ejemplo de una capa convolucional de una CNN. El sector marcado en rojo indica que todas esas neuronas tendrán información focalizada de una sola parte de la imagen. Al mismo tiempo, los distintos filtros convolucionales están marcados en distintos colores, para intentar identificar distintas características de la imagen.

Fuente: Elaboración propia, 2020.

Los procesos principales realizados por una red convolucional, se dividen en distintas capas, las que pueden tener distinto orden, pero todas comienzan con una capa de entrada para recibir los datos proporcionados, y terminan con una capa de salida, que tendrá distintas funciones dependiendo del problema en cuestión, como puede ser clasificación binaria, multiclasificación, etc.

A continuación se presentan distintos tipos de capas que se pueden utilizar en la red convolucional, los que se escogieron debido a que estos son los que se utilizaron en el desarrollo de este experimento.

Capa convolucional

La convolución es una operación matemática entre dos funciones, que da como respuesta una tercera función que indica como la forma de una función es modificada por la otra. Gráficamente, la convolución es tomar una función $f(t)$ y pasarlala a lo largo de otra función $g(t)$. Este comportamiento se describe matemáticamente en la Formula 2.2, en donde la integral está definida en torno a τ , que refleja la posición de una función que van cambiando al pasar una por sobre la otra. También es importante detallar que la función $g(t)$ está reflejada sobre el eje y al momento de hacer la convolución (Weisstein, 2020).

$$s(t) = (f * g)(t) = \int f(\tau)g(t - \tau)d\tau \quad (2.2)$$

En el contexto de las *CNN*, la convolución se aplica al pasar un filtro sobre la imagen que recibe como entrada. Este proceso es el que describirá algunas características pertenecientes en la imagen, como por ejemplo, la detección de bordes en alguna figura.

En la Figura 2.12, se muestra un ejemplo de este proceso convolucional, al aplicar un filtro de 3×3 sobre una imagen de entrada. El uso del filtro corresponde a un producto punto, en donde, siguiendo el ejemplo de la figura, se tiene la Formula 2.3.

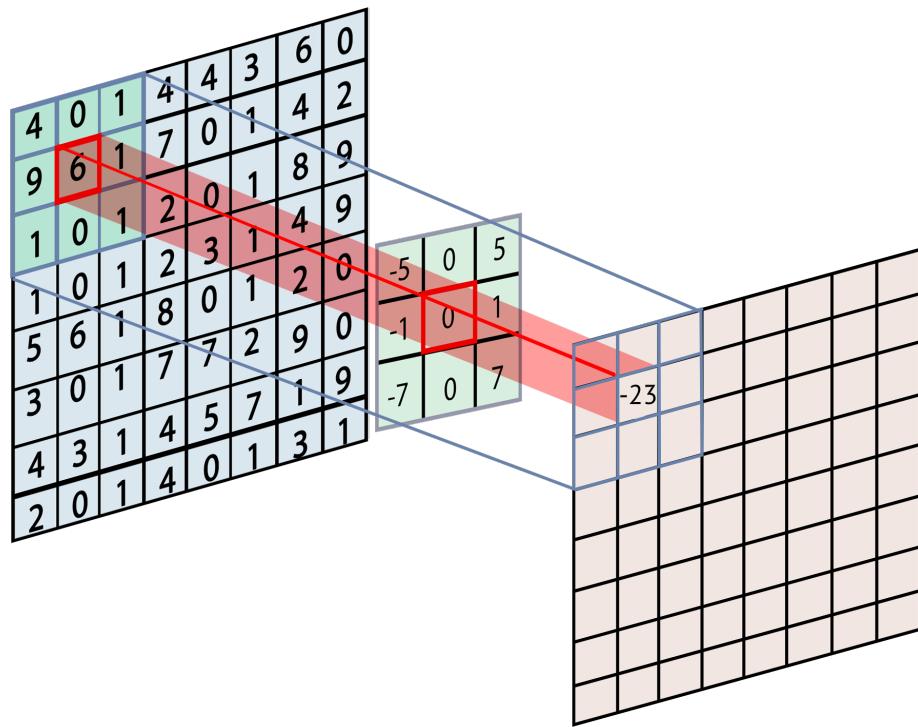


Figura 2.12: Ejemplo de convolución en una imagen. La matriz resultante corresponde al producto punto de los valores de la imagen original y el filtro convolucional, y el resultado va perdiendo dimensionalidad debido a esto.

Fuente: Elaboración propia, 2020.

$$(4 \times -5) + (0 \times 0) + (1 \times -5) + (9 \times -1) + (6 \times 0) + (1 \times 1) + (1 \times -7) + (0 \times 0) + (1 \times 7) = -23 \quad (2.3)$$

El filtro viaja a través de toda la imagen de entrada, como si fuera escaneandola, obteniendo así otro arreglo de menor dimensionalidad que el original, en donde se ven representadas algunas características de la imagen original. Estas características dependen del filtro que se utilice en la operación (Surma, 2019), los que son representados por los distintos

colores en la Figura 2.11.

Capa de activación

En estas capas se tiene como objetivo realizar una función de activación, como se habló en la subsección anterior de aprendizaje profundo. Anteriormente se hablaba de utilizar funciones lineales o categóricas para generalizar la salida de una red, pero en este caso, se intenta generalizar las características presentes en la imagen de entrada. Este tipo de capas no lineales suelen utilizarse, por convención, luego de cada capa convolucional, ya que la idea es ingresar no-linealidad a un sistema que acaba de realizar una operación lineal como lo es la convolución.

A continuación se detallan las funciones de activación *ReLU* y *Softmax*, debido a que son estas las funciones que se utilizaron en la construcción de las redes convolucionales que se utilizaron en este trabajo.

La función de activación ***ReLU***, que se caracteriza por dejar todos los valores negativos en 0, y mantener todos los valores positivos tal cual. En la Figura 2.13 se puede ver un ejemplo de esta función de activación sobre una entrada, la cual no disminuye su dimensionalidad en este caso. Se ha decidido mostrar esta función como ejemplo, ya que la red convolucional escogida en este trabajo utiliza este tipo de activación en su construcción, pero existen muchas otras funciones que se pueden aplicar en una red convolucional.

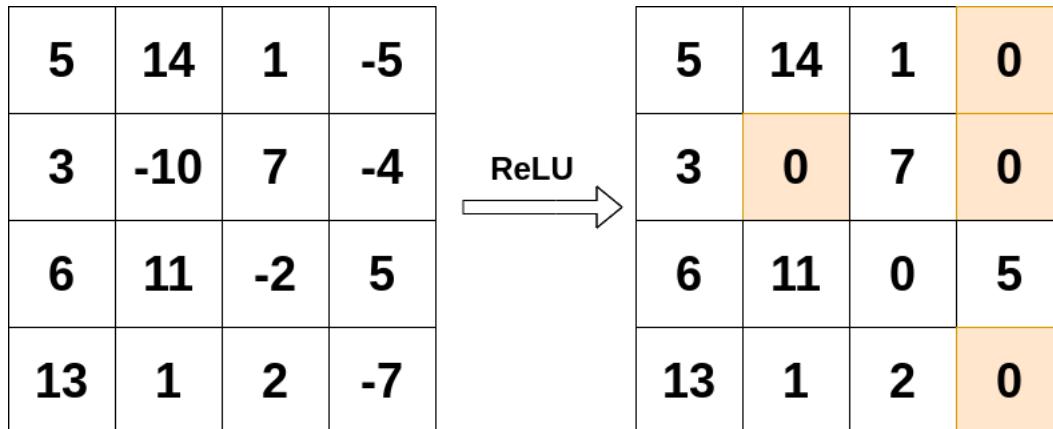


Figura 2.13: Capa de activación ***ReLU***. En este se ve como los valores negativos son convertidos en cero.

Fuente: Elaboración propia, 2020.

Otro ejemplo importante a mencionar como activación, es la función ***Softmax***, la cual generaliza las salidas en probabilidades. En el contexto de las redes convolucionales, esta se utiliza en conjunto con una capa completamente conectada para disminuir la dimensionalidad de

la red al número de clases del problema, y obtener la probabilidad de pertenencia de la entrada a una de las clases trabajadas. Esta función de activación se utiliza para problemas de clasificación, y la salida de esta depende de cada una de las clases, por lo que la suma de las probabilidades de todas las clases debe dar uno (o 100%) como resultado. Esto significa que esta activación entrega una etiqueta por entrada, por lo que se usa cuando las clases a trabajar son excluyentes, o sea, que una entrada no puede pertenecer a más de una etiqueta (Uniqtech, 2020).

Capa de pooling

Pooling se puede traducir al español como agrupación, y es exactamente esto lo que realiza una red convolucional en esta capa. Una función de *pooling* agrupará parte de la entrada en un solo valor, el que cumple la función de ser un resumen estadístico de los datos agrupados, y ayuda a que las características presentes en la imagen sean invariantes a cambios sobre esta. Esto quiere decir que, en el caso de querer identificar bordes de una figura en un conjunto de datos, la red podrá identificar estas características independiente de como se presenten los mismos bordes en otras imágenes del conjunto (como rotación, cambio de tamaño, o la presencia de esta característica en otra parte de una imagen).

La CNN trabajada en este experimento, utiliza la función **Max pooling** en la construcción de la red. Este función de agrupación, como dice su nombre, toma solo el valor mayor en un conjunto de datos de la entrada, descartando el resto. Un ejemplo de esto se puede ver en la Figura 2.14, en donde se ve como la dimensionalidad de la imagen disminuye al aplicar este tipo de función (Angluin et al., 2016).

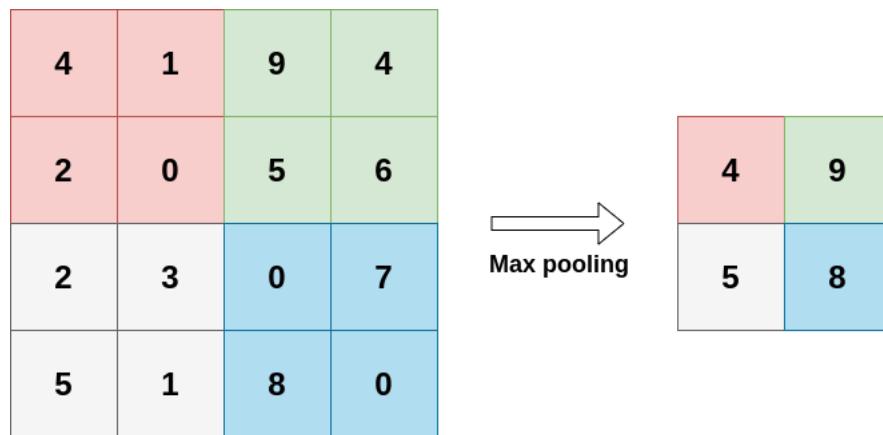


Figura 2.14: Ejemplo de función de agrupación **Max pooling**. Utilizando una activación de dimensión 2×2 , la función va escogiendo los mayores valores de la matriz
Fuente: Elaboración propia, 2020.

Capa completamente conectada

Una capa completamente conectada (*fully connected*) tiene este nombre, debido a que su función es aplanar la entrada y pasarl a una red completamente conectada, tal como se puede observar en la Figura 2.4. A esta altura, la dimensionalidad de la imagen de entrada ha sido disminuida dependiendo de las capas que tenga la red convolucional, además que el aplanamiento está definido por el tipo de problema en cuestión. Por ejemplo, para el reconocimiento de los gatos y perros, corresponde a una clasificación binaria, por lo que la salida de la red debiese por un lado tener la generalización de las características de un perro, y viceversa. Para este problema, la capa completamente conectada entonces aplana la imagen en dos nodos, donde cada uno corresponde a tener las características para cada animal (O'Shea & Nash, 2015).

Otras capas

Existen además otro tipo de operaciones que se pueden realizar en una red convolucional, que no tienen que ver directamente con el análisis de las características de la entrada, como pueden ser capas de normalización de los datos o de regularización.

La capa de *dropout* corresponde a esta última categoría, en donde se busca combatir con el *overfitting* durante el entrenamiento. La regularización consiste en penalizar el modelo durante el entrenamiento, de modo que este no memorice las entradas durante el entrenamiento de la red. Para el caso de una capa de *dropout*, lo que hace es eliminar de forma aleatoria, y con la misma probabilidad, algunas de las neuronas de la red convolucional, generando así ruido en el entrenamiento. Es este ruido el que finalmente ayudará a la red a no memorizar las entradas y prevenir el *overfitting* (Srivastava et al., 2014).

En las librerías como *Keras*, estas capas ya se encuentran implementadas, e incluso se encuentran adaptadas a distintos problemas, como puede ser análisis de imágenes en dos dimensiones y en tres dimensiones. Además, la librería entrega un resumen de las capas utilizadas en una red, y la dimensión de la entrada a lo largo de esta, lo cual es bastante útil al momento de identificar si una imagen es capaz de soportar todas las capas convolucionales sin perder el total de la imagen de entrada, luego de ir disminuyendo su dimensión (Chollet et al., 2015).

2.4.4 Aprendizaje por transferencia

En el aprendizaje de máquinas, existe la posibilidad de reutilizar algoritmos, sin necesidad de que los datos utilizados sean sobre el mismo problema para el cual estos algoritmos fueron utilizados originalmente. El aprendizaje por transferencia (o *transfer learning*) es una técnica que, tal como indica su nombre, intenta transferir conocimiento aprendido en un problema hacia otros, reutilizando algoritmos y modelos (Pan & Yang, 2010). De la misma forma que existe el aprendizaje supervisado y no supervisado, existen dos tipos de aprendizaje por transferencia, ya sea dependiendo si los datos se encuentran previamente etiquetados o no.

El aprendizaje por transferencia es una herramienta bastante útil, en escenarios en donde los datos no son los suficientes para crear modelos capaces de resolver a los problemas para los que fueron construidos (Pan & Yang, 2010). En el *dataset* de **SCIAN-MorphoSpermGS**, presentado en el Capítulo 1, se puede observar que el conjunto de imágenes de espermatozoides a utilizar, tiene alrededor de 3.000 imágenes, y además, solo se va a utilizar una porción de este conjunto para crear los modelos, en donde el número disminuye a aproximadamente 1.100 imágenes, por lo que este tipo de aprendizaje podría ayudar a crear los modelos que clasifiquen las cabezas de espermatozoides.

Un ejemplo simple de aprendizaje por transferencia, es intentar crear un clasificador que descubra si existen zarigüeyas en las fotografías, y esto se logra utilizando un modelo pre-entrenado, el cual fue entrenado con el propósito de clasificar imágenes que contenían mapaches. En este caso, las características de un animal pueden estar presentes en el otro, por lo que al reutilizar un modelo con conocimientos previos, se puede obtener mejores modelos clasificadores con respecto a crear modelos desde cero, utilizando pesos aleatorios en las neuronas de la red. Cuando se entrena un modelo desde cero, con pesos iniciales aleatorios, se conoce como *entrenamiento desde scratch* (Brownlee, 2017).

Para el caso de redes convolucionales, el aprendizaje por refuerzo se puede hacer de distintas formas. Primero que todo, se debe tener un modelo pre-entrenado, de donde se van a obtener los pesos de la red con el conocimiento de algún problema anterior. Luego de esto, los pasos a seguir pueden ser los siguientes (Chollet et al., 2015).

- Entrenar la red nuevamente con los el nuevo conjunto de datos
- Congelar algunas capas y entrenar el modelo con los datos nuevos. Congelar significa que las neuronas de la capa congelada no se van a volver a actualizar al terminar una época de entrenamiento.
- Agregar nuevas capas a la arquitectura, y entrenar el modelo con los datos nuevos

Cuando se han congelado algunas capas, un paso opcional que también puede realizar es descongelar las capas del modelo ya entrenado con aprendizaje con transferencia, y volver a entrenar en pocas iteraciones y con una tasa de aprendizaje pequeña. Este último paso se le conoce como *fine tuning*, y aunque puede disminuir el error del modelo, también puede provocar *overfitting* (Chollet et al., 2015).

2.5 SESGO Y VARIANZA

El término de sesgo (o *bias* en inglés) fue introducido por Mitchell (1980), y lo describe como la capacidad de un modelo para darle importancia a las características de la información, con el objetivo de generalizar mejor al momento de utilizar conjuntos de datos nuevos y más grandes. En *Machine Learning*, este término se utiliza de forma más precisa, y se refiere a un error de aprendizaje, en donde el modelo tiende a darle importancia a ciertas características por sobre otras, lo que genera resultados prejuiciosos al momento de utilizar un modelo entrenado sobre nueva información (Khetan, 2019). Un alto sesgo produce lo que se llama ***underfitting***, que significa que el modelo no da los resultados correctos debido a que no conoce las suficientes características como para generalizar los datos.

Por otro lado, la varianza (o *variance* en inglés) también es un error, y se considera como la cantidad en que la estimación de la función objetivo $f(x)$ puede cambiar, si es que en el entrenamiento se presentan datos nuevos al modelo, o sea, que tan sensible es el modelo a los cambios en el conjunto de entrenamiento. Un alto valor de varianza significa que el modelo esta memorizando las características de los datos del entrenamiento, incluido el ruido que pueda existir en estas. Este problema se conoce como ***overfitting***, ya que memorizar las características es lo contrario a generalizarlas, por lo que al momento de que el modelo se le presenten nuevos datos, no podrá predecir los resultados de manera correcta (Guanga, 2018).

Volviendo al ejemplo de clasificación de perros y gatos, un alto sesgo puede ocurrir al presentar más imágenes de perros en el entrenamiento, provocando que el modelo de más importancia a características de un canino, en comparación a las del gato. Al momento de clasificar el animal en nuevas imágenes, lo más probable es que el modelo identifique a los gatos de forma incorrecta, ya que no conoce las suficientes características para diferenciar una especie de la otra. Si se entrena la red por muchas iteraciones sobre un conjunto limitado de imágenes de perros y gatos distintos, se estaría generando un alto nivel de varianza, por lo que la clasificación de nuevas fotos de estos animales tendría un alto nivel de error, ya que el modelo solo sabe la especie correcta para las imágenes que ya conoce.

Para que la red tenga mejor precisión en la predicción sobre nuevos datos, lo que se

busca es disminuir los dos errores que se mencionaron. Sin embargo, al disminuir el sesgo, la varianza tiende a aumentar, y viceversa, por lo que se debe encontrar un punto mínimo entre los dos valores. Este problema se conoce como ***bias - variance tradeoff*** (Kohavi & Wolpert, 1997).

2.6 AUMENTO DE DATOS

El aumento de datos (*data augmentation* o *AD*), es otra técnica utilizada para combatir el *overfitting* durante el entrenamiento de una red, la cual consiste en incrementar la cantidad de datos durante el entrenamiento de la red, utilizando únicamente el mismo conjunto de datos que se vaya a usar en el entrenamiento. Una de las formas clásicas de realizar esta técnica cuando se trabaja con imágenes, es realizar transformaciones sobre estas, como pueden ser rotaciones, traslaciones, reflejar una imagen, entre otros (Perez & Wang, 2017). En la Figura 2.15 se puede ver un ejemplo de rotaciones en distintos grados sobre una misma imagen.



Figura 2.15: Ejemplos de transformaciones hechas con *Keras* en *Python* (Sarin, 2019). La misma imagen es transformada en cierta medida, generando nuevos datos.

El aumento de datos se puede realizar previo al entrenamiento y guardando la información en el disco duro, en casos donde la cantidad de imágenes es muy limitada, o por otro lado, se pueden generar imágenes nuevas en tiempo de entrenamiento, ahorrando así el espacio en disco y obteniendo nuevas imágenes en cada iteración del entrenamiento. Esta es una técnica ventajosa cuando se está trabajando con conjuntos limitados de datos, como pueden ser en el área de la medicina, en donde es más costoso obtener muestras e imágenes (Perez & Wang, 2017).

Hay que destacar que las imágenes generadas en este proceso debiesen de tener sentido para el problema en cuestión. Por ejemplo, para el caso de este trabajo, con un problema de clasificación según la forma del objeto en una imagen, realizar transformaciones como recortar o cambiar la razón de aspecto de esta, afecta directamente a la forma del objeto a clasificar, por lo que perdería el significado de la imagen.

2.7 VALIDACIÓN CRUZADA

Al momento de generar modelos estadísticos, es importante verificar el comportamiento de estos cuando se le presentan conjuntos de datos independientes, ya que los resultados no siempre serán los mismos con respecto a la información entregada para la formación de los modelos. Un ejemplo de esto, son el problema de *Overfitting* que ya se ha hablado con anterioridad, en donde los modelos estadísticos comienzan a memorizar las características de los datos de entrada, en vez de generalizar características, por lo que esto no podrán tener buenos resultados cuando se les presente nueva información.

La validación cruzada (o *cross validation*) es una técnica común para asegurar la robustez de los modelos generados, ya que, su objetivo es ir entrenando los modelos con una fracción de los datos, para luego evaluar como se van comportando con la información que aún no han visto, para así poner a prueba que tan bien han generalizado la información, y estimar el verdadero error de los modelos (Shaikh, 2018).

Existen distintas formas de separar los datos para realizar la validación cruzada, en donde la más común corresponde a la validación cruzada *k-fold*. En este, el conjunto de datos es separados en k particiones del mismo tamaño, y entrenado k veces también. En cada una de estas iteraciones de entrenamiento, se escogerán $k - 1$ particiones para entrenar el modelo de esta iteración, y la partición restante servirá para evaluar el modelo entrenado. De esta manera, se asegura que para el final de las k iteraciones, se ha evaluado el nivel predictivo de los modelos con distintos grupos de imágenes. Al final, suele utilizarse el promedio de las métricas de evaluación obtenidas en las k iteraciones (Pedregosa et al., 2011).

Una alternativa del *k-fold* anterior, en problemas de clasificación, consiste en realizar el mismo procedimiento, pero de manera estratificada. Esto es conocido como *stratified k-fold*, y a diferencia del *k-fold* anterior definido, al formar las k particiones de datos, cada clase de los datos tendrá una proporción cercana, a la que esa clase tiene en la totalidad de los datos. De esta manera, cuando se trabaja con datos desbalanceados, se puede asegurar que los modelos son entrenados y evaluados con muestras de todas las clases (Pedregosa et al., 2011).

Por otro lado, también es posible hacer este tipo de muestreo sin *k-fold*, si no con permutaciones aleatorias, es decir, se deja de partir el conjunto en k particiones iguales. En esta ocasión, se escoge una proporción en la cual se van a separar los conjuntos de entrenamiento y evaluación, y en cada iteración k , se toman muestras aleatorias, y con repetición, para cada uno de estos sub-conjuntos. En esta oportunidad, el tamaño del conjunto de entrenamiento y evaluación ya no dependen del valor k de iteraciones, y al mismo tiempo, las muestras para entrenamiento y/o validación pueden contener datos repetidos entre una iteración y otra. A esta

variación se le conoce como *shuffle split cross validation*, y al igual que en el caso anterior, también puede realizarse de manera estratificada, para asegurar la presencia de todas las clases en el entrenamiento y en la evaluación (Pedregosa et al., 2011).

2.7.1 Métricas de evaluación

De las iteraciones de validación cruzada, se obtienen medidas, o métricas de evaluación, las que generalmente se obtiene el promedio, para indicar los resultados obtenidos por los modelos.

Matriz de confusión

La matriz de confusión es un arreglo de dos dimensiones C_{ij} , para verificar de manera visual, la clasificación realizada por un modelo, en donde i y j son la cantidad de clases que se busca clasificar en el problema. En las filas de la matriz se encuentran la clasificación real de los datos, mientras que en las columnas se muestran los valores predichos por el modelo, aunque el orden puede cambiar según las bibliografía (Pedregosa et al., 2011).

En la Tabla 2.1 se observa un ejemplo de una matriz de confusión para clasificación binaria, en donde se ve la separación entre los datos que fueron clasificados correctamente, que son los verdaderos positivos y los verdaderos negativos, y los datos que fueron incorrectamente clasificados, los que se conocen como falsos positivos y verdaderos negativos. La suma de los valores de cada fila, corresponde a la cantidad de datos de esa clase, mientras que la suman de los valores de toda la matriz, corresponde al total de los datos de la muestra.

		Clase predicha	
		Clase A	No es clase A
Clase real	Clase A	Verdaderos positivos	Falsos negativos
	No es clase A	Falsos positivos	Verdaderos negativos

Tabla 2.1: Ejemplo de matriz de confusión
Fuente: Elaboración propia, 2020.

A partir de la matriz de confusión, y del ejemplo de la Tabla 2.1, se pueden obtener las siguientes métricas de evaluación por cada clase del problema:

- Sensibilidad: También conocida como tasa de verdaderos positivos (o *TPR* en inglés), mide la proporción de positivos que fueron clasificados como tal, lo cual se puede traducir a el porcentaje de datos de la clase A que efectivamente fueron clasificados en esta caso

(Parikh et al., 2008). En los trabajos del estado del arte (Chang et al., 2017b; Riordon et al., 2019; Iqbal et al., 2020), se usa principalmente esta métrica para comparar los resultados obtenidos de la clasificación morfológica de cabezas de espermatozoides humanos.

$$Sensibilidad = \frac{Verdaderos\ positivos}{Verdaderos\ positivos + Falsos\ negativos} \quad (2.4)$$

- Tasa de falsos positivos (o *FPR* en inglés): Está mide la proporción de casos negativos que fueron clasificados erróneamente. En el ejemplo actual, se considera como el porcentaje de datos que no fueron clasificados como la clase A, cuando efectivamente lo eran (Parikh et al., 2008).

$$Tasa\ de\ falsos\ positivos = \frac{Falsos\ positivos}{Falsos\ positivos + Verdaderos\ negativos} \quad (2.5)$$

- Especificidad: También conocida como tasa de verdaderos negativos (o *TNR* en inglés), mide la proporción de casos negativos que fueron identificados como tal, lo que, según el ejemplo anterior, la especificidad de la clase A se refiere al porcentaje de datos que no pertenecen a la clase A, y que fueron identificados correctamente (Parikh et al., 2008).

$$Especificidad = \frac{Verdaderos\ negativos}{Verdaderos\ negativos + Falsos\ positivos} \quad (2.6)$$

- Precisión: La precisión también se conoce como valor predictivo positivo, y se refiere a la proporción de clasificaciones positivas, entre los datos que son verdaderos positivos, o sea, Volviendo al ejemplo de la Tabla 2.1, de entre todos los datos que fueron clasificados como la clase A, la precisión consiste en el porcentaje de estos que fueron clasificados de forma correcta (Parikh et al., 2008).

$$Precisión = \frac{Verdaderos\ positivos}{Verdaderos\ positivos + Falsos\ positivos} \quad (2.7)$$

- Exactitud: La exactitud corresponde a la proporción de clasificaciones correctas para todas las clases, con respecto al total de los datos, por lo que en el ejemplo anterior, corresponde al porcentaje de las clasificaciones correctas que pertenecen y no pertenecen a la clase A, con respecto a la totalidad de los datos (Parikh et al., 2008). En los trabajos del estado del arte (Chang et al., 2017b; Riordon et al., 2019; Iqbal et al., 2020), también se compara la exactitud obtenida de la clasificación morfológica de cabezas de espermatozoides humanos, pero esta es obtenida como el promedio de la sensibilidad de todas las clases existentes en la clasificación.

$$Exactitud = \frac{Verdaderos positivos + Verdaderos negativos}{Total de datos} \quad (2.8)$$

Como cada métrica es calculada para cada clase, para problemas de clasificación con más de dos clases, se pude tomar la matriz de confusión y analizar cada clase como si fuera un problema de clasificación binaria, convirtiéndose en una matriz de $2x2$, en donde las filas corresponden a los valores reales (datos que si pertenecen a la clase A y los datos que no pertenecen a la clase A), y en las columnas se tendrán las predicciones realizadas (los datos que se clasificaron con la clase A, y los datos que no se clasificaron con la clase A).

Curva ROC

La curva *ROC* es herramienta para medir el rendimiento de un modelo clasificador. La curva *ROC* de representa en una gráfica de la tasa de falsos positivos, versus la tasa de verdaderos positivos, y esta es una curva de probabilidad, la cual se va formando con estos valores, a partir de distintas configuraciones de umbral. El área que se forma bajo la curva *ROC*, es lo que se conoce como área bajo la curva, y esta mide la capacidad del modelo de diferencias las dos clases que el modelo está tratando de clasificar (Narkhede, 2019).

Mientras más cerca la curva este del valor 1 en el eje *y*, mayor será el área bajo la curva, por lo que el modelo será mejor en diferenciar una clase por sobre la otra. Cuando la curva es una diagonal, significa que el área bajo la curva es de 0,5, indicando que el modelo no es capaz de diferenciar alguna de las dos clases. Por último, cuando la curva de acerca al 0 en el eje *y*, el área bajo la curva comienza a acercarse a 0, indicando que el modelo está clasificando de forma recíproca los datos (asigna la clase contraria para todos los datos) (Narkhede, 2019). Para clasificación de múltiples clases, la curva *ROC* suele ser una para cada una clase que el modelo intenta clasificar.

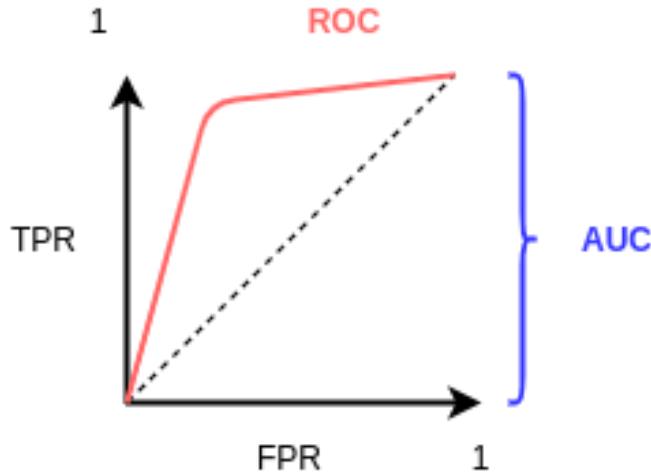


Figura 2.16: Ejemplo de curva *ROC*. Corresponde a la gráfica de la tasa de falsos positivos versus la tasa de verdaderos positivos. La linea roja representa la curva *ROC*, mientras que el área bajo la curva *ROC* es la denominada *AUC*. (*area under curve*. La linea punteada representa el *AUC* valor 0,5.

Fuente: Elaboración propia, 2020.

2.8 RESUMEN

El análisis morfológico de los espermatozoides es un paso importante al momento de diagnosticar el nivel de fertilidad de un hombre, en donde uno de estos análisis corresponde a la clasificación de defectos, según la forma de los espermatozoides. Este análisis de espermatozoides sigue guías construidas por expertos de la Organización Mundial de la Salud, y detallan el tipo de morfología que pueden tener distintas partes del espermatozoide humano, siendo una de ellas la cabeza de estas células.

El aprendizaje de máquinas consiste en la forma en que un computador puede realizar ciertos trabajos, sin decirle explícitamente que hacer, y es aquí donde se presenta un subconjunto de esta área, llamado aprendizaje profundo, el cual consiste en aquellos algoritmos capaces de realizar procesamiento y generalización de características de alto nivel, en donde un ejemplo de esto corresponde a las redes neuronales convolucionales.

Una red convolucional es un tipo de red neuronal que se utiliza generalmente para análisis de imágenes, por lo que se utilizará esta herramienta para el problema de clasificación de cabezas de espermatozoides según su forma. Este tipo de red tiene la característica de tener un aprendizaje supervisado, por lo que se utilizarán pares de imágenes y etiquetas para calcular el error y recalcular los pesos de la red en las siguientes iteraciones de estas.

La construcción de la red tendrá distintas capas, encargadas tanto de generalizar e identificar características, como también para regularizar el entrenamiento y prevenir el *overfitting* de la red. Otra técnica que se utiliza para prevenir este problema, es el aumento de datos, en donde se generan nuevas imágenes a partir de un conjunto original durante el entrenamiento, para así prevenir que la red memorice las entradas.

CAPÍTULO 3. ESTADO DEL ARTE

Dentro de la literatura, se escogieron unos documentos publicados en el año 2017, que se centran en la búsqueda de un *gold standard* para la clasificación de cabezas de espermatozoides asistida por computadora, publicados por Chang et al. (2017a) y Chang et al. (2017b). Estos trabajan sobre el mismo conjunto de datos y con las mismas etiquetas para la clasificación de estas células, que provienen del manual de laboratorio para el análisis de semen publicado por la OMS (2010b). Estas clases se pueden ver en la Figura 3.1, con ejemplos del *dataset* utilizados por los autores anteriores en sus experimentos.

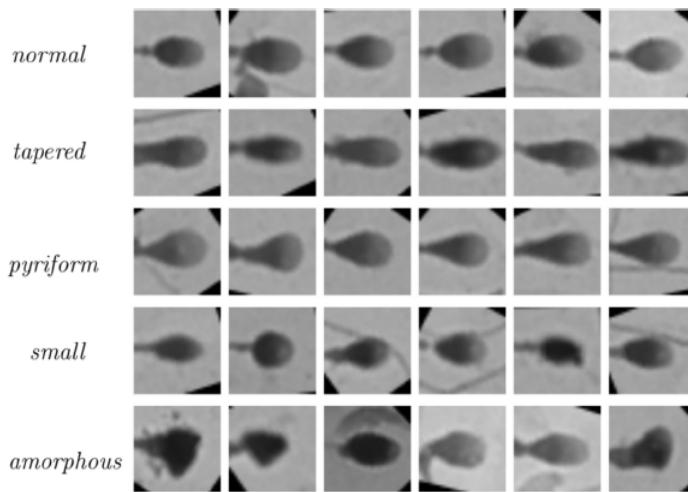


Figura 3.1: Formas de cabezas de espermatozoides (Chang et al., 2017a). Cada fila representa ejemplos de imágenes de una de las cinco clases del conjunto de datos.

Los datos obtenidos pertenecen al *dataset* llamado *SCI&AN-MorphoSpermGS*, el cual esta formado por 1.854 imágenes de células de espermatozoides observables y evaluables, las cuales fueron clasificadas por tres expertos chilenos, con mucha experiencia en este tipo de análisis, de forma manual e independiente. Los datos están separados en los siguientes tres sub-conjuntos:

- Conjunto base: corresponde a las 1.854 muestras categorizadas por los tres distintos expertos
- Conjunto de acuerdo parcial: tiene 1.132 muestras de espermatozoides en donde dos de los tres expertos llegaron a la misma clasificación.
- Conjunto de acuerdo total: finalmente este posee 384 muestras donde los tres expertos llegaron a la misma clasificación.

3.1 ESTÁNDAR DE ORO PARA ANÁLISIS MORFOLÓGICO DE ESPERMATOZOIDES, ASISTIDO POR COMPUTADORAS (2017)

En el trabajo de Chang et al. (2017a), los investigadores combinaron distintos métodos de clasificación, en conjunto con algoritmos descriptores de imágenes, que son los encargados de obtener la mayor información posible de las áreas de interés, que en este caso corresponde a las cabezas de los espermatozoides. Para este trabajo, se utilizó el conjunto de acuerdo parcial y mayoría de votos para decidir la clase "real" de cada una de las imágenes.

Los descriptores utilizan la forma de las zonas de una imagen, para así entregar la información correspondientes a los clasificadores. Los algoritmos de clasificación usados en este caso son *K-nearest neighbors*; *Naive Bayes Classifier*; *Decision Trees* y *Support Vector Machines*, mientras que por el lado de los descriptores, los que se utilizaron fueron *Hu Moments*; *Zernike moments* y *Fourier descriptors* Chang et al. (2017a). Los resultados de este experimento se presentan en la Tabla 3.1, donde se muestra el nivel de exactitud obtenido por cada clasificador con cada descriptor, mientras que en la Tabla 3.2 se presentan los resultados específicos para cada clase, usando el descriptor de Fourier.

Clasificador/Descriptor	Hu Moments	Zernike moments	Fourier Descriptors
1 - NN	0,39	0,40	0,40
Bayes	0,33	0,45	0,44
Decistion Trees	0,39	0,41	0,40
SVM	0,46	0,46	0,49

Tabla 3.1: Resumen: exactitud de la clasificación según Chang et al. (2017a).
Fuente: Elaboración propia, 2020.

Clasificador/ TPR	tpr(N)	tpr(T)	tpr(P)	tpr(S)	tpr(A)	Promedio
1 - NN	0,50	0,49	0,36	0,44	0,20	0,40
Bayes	0,51	0,45	0,58	0,78	0,10	0,44
Decistion Trees	0,44	0,51	0,35	0,45	0,25	0,40
SVM	0,57	0,68	0,53	0,54	0,15	0,49

Tabla 3.2: Resultados de la clasificación según Chang et al. (2017a) con descriptores de Fourier.
Fuente: Elaboración propia, 2020.

3.2 CLASIFICACIÓN MORFOLÓGICA AUTOMÁTICA DE CABEZAS DE ESPERMATOZOIDES HUMANOS (2017)

En el segundo documento, Chang et al. (2017b) utilizaron una clasificación en dos etapas para reducir la tasa de error en el proceso. La primera etapa consiste en decidir si las cabezas de espermatozoides caen en la categoría de amorfas o no. En el caso de que no sean amorfas, pasan a la segunda etapa, la cual corresponde a cuatro verificadores distintos para cada una de las clases restantes. En la segunda etapa, los cuatro verificadores buscan clasificar la célula en alguna de las otras cuatro clases contra la amorfía (normal v/s *amorphous*; *tapered* v/s *amorphous*; *small* v/s *amorphous*; *pyriform* v/s *amorphous*), entregando la etiqueta final del gameto observado. En ambas etapas se utilizan algoritmos descriptores de imágenes, seguidos de algoritmos de clasificación. Dentro de los descriptores, se ocuparon seis distintos, entre los cuales se incluye uno propuesto por los propios autores. Para la clasificación, se utilizó finalmente el método de *Support Vector Machine*. Los resultados de esta clasificación propuesta se presentan en la Tabla 3.3 en dos conjuntos de datos distintos.

Dataset/TPR	tpr(N)	tpr(T)	tpr(P)	tpr(S)	tpr(A)	Promedio
Dataset de acuerdo parcial	0,62	0,64	0,50	0,82	0,30	0,58
Dataset de acuerdo total	0,74	0,70	0,92	1,00	0,30	0,73

Tabla 3.3: Resumen de resultados de la clasificación según Chang et al. (2017b)
Fuente: Elaboración propia, 2020.

Las pruebas realizadas se hicieron en dos conjuntos distintos. El primero corresponde al *dataset* de acuerdo parcial, que corresponde al mismo utilizado en el trabajo anterior (Chang et al., 2017a), mientras que el *dataset* de acuerdo total, es un subconjunto del primero, en donde solo se escogieron las imágenes en donde los tres expertos escogieron la misma clase para una cabeza de espermatozoide.

Los resultados de la Tabla 3.3 muestran mejor desempeño en comparación a los observados en la Tabla 3.2. Utilizando el grupo de acuerdo total, la exactitud alcanzó un valor del 73%. Esto quiere decir que el método propuesto por los investigadores logra clasificar de forma correcta 3 de 4 imágenes de cabezas de espermatozoides dentro del conjunto de imágenes más sencillo de clasificar. Sin embargo, en el conjunto de acuerdo parcial, aquel en donde hubo desacuerdo entre los expertos, la exactitud solo llegó a un 58%, lo que significa un poco más de la mitad de aciertos en la clasificación en estas imágenes. Esto quiere decir que, si las imágenes son difíciles de etiquetar incluso para los médicos expertos, entonces la clasificación hecha por este algoritmo también baja su rendimiento al predecir una forma correcta de cabeza de espermatozoide.

Ahora bien, a través de la Tabla 3.2 y la Tabla 3.3, se puede observar también el detalle de clasificaciones correctas por clases de cabezas de las células, en donde para ambos experimentos se puede observar el mismo problema. En cualquiera de los dos experimentos, se puede observar la tendencia del método a fallar cuando intenta clasificar cabezas que son de tipo amorfas, donde en el primer trabajo solo se alcanza un 25% de *TPR* (tasa verdaderos positivos) usando árboles de decisión, mientras que en el segundo trabajo, solo se logró clasificar de forma correcta un 30% la misma clase. Esto quiere decir que ambos trabajos solo lograron clasificar de forma correcta, aproximadamente, una de cuatro cabezas de espermatozoides amorfas.

A partir de estos resultados, el haber alcanzado una exactitud del 73% aún deja espacio a mejoras en busca del *gold standard*, además que, en el caso de etiquetar muestras que en su mayoría contienen células amorfas de espermatozoides, las predicciones correctas serán mucho más bajas que este número.

3.3 UN NUEVO MÉTODO DE APRENDIZAJE PROFUNDO PARA LA EVALUACIÓN AUTOMÁTICA IMÁGENES DE ESPERMATOZOIDES HUMANOS (2019)

Por otro lado, existe el trabajo publicado por (Javadi & Mirroshandel, 2019), el cual consiste en clasificación morfológica de cabezas de espermatozoides humanos, utilizando redes neuronales convolucionales.

A diferencia de los trabajos mencionados anteriormente (Chang et al., 2017a,b), se optó por un proceso de clasificación binaria, para clasificar cabezas de espermatozoides normales y "anormales". Al mismo tiempo, la clasificación morfológica se hizo no solo a partir de la cabeza misma de los espermatozoides, si no que también se analizó la morfología de las vacuolas y los acrosomas.

En este trabajo, se utilizó el *dataset* público llamado *MHSMA*, el cual esta compuesto por aproximadamente, 1.500 imágenes de espermatozoides, obtenidas de 235 pacientes hombres con algún nivel de infertilidad, de las cuales todas se encuentran etiquetadas por expertos como normal y anormal para las cabezas, acrosomas y vacuolas.

Estas imágenes son de 1280×1024 píxeles, y contienen a todo el espermatozoide, no solamente la cabeza, por lo que se tuvo que marcar el área de interés de manera manual, terminando con imágenes de 128×128 píxeles con el área de interés, que corresponde a la cabeza de los espermatozoides, y esto se ve reflejado en la Figura 3.2.

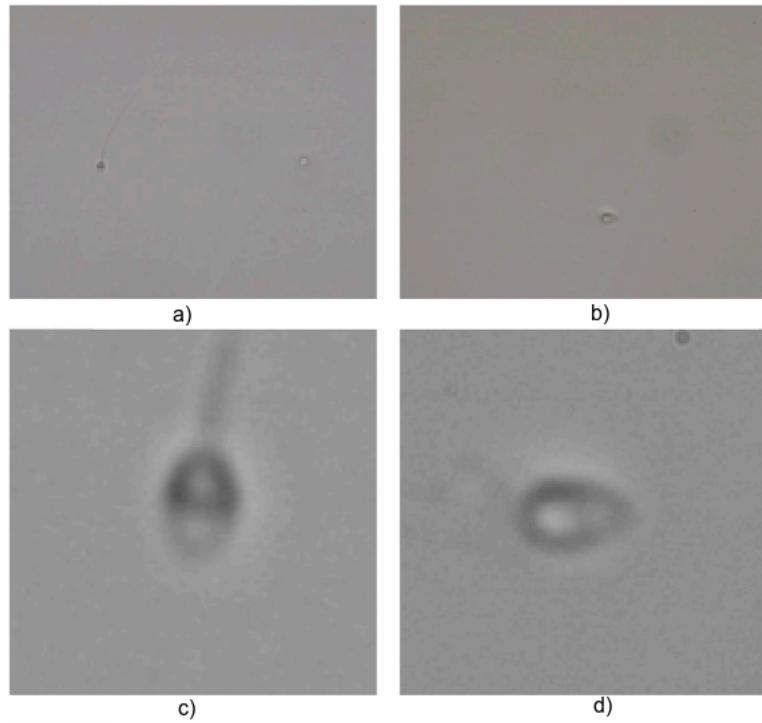


Figura 3.2: Ejemplos de imágenes del *dataset* de MHSMA (Javadi & Mirroshandel, 2019). En a) y b) se ven imágenes originales del *dataset*, mientras que c) y d) son el área de interés de a) y b) respectivamente, que se usaron en el análisis morfológico.

Al igual que con el *dataset* de SCIAN-MorphoSpermGS, las clases normales y anormales se encuentran desbalanceadas, y además, la cantidad de imágenes puede no ser suficiente para entrenar un modelo profundo como una red neuronal convolucional. Por lo mismo, se propuso realizar la técnica de aumento de datos, para así generar nuevas imágenes y poder tener mayor información al momento de entrenar los modelos.

En la Figura 3.3 se presenta las capas que conforman la arquitectura de la red neuronal convolucional propuesta en este trabajo, el que, solo tiene una profundidad de 12 capas, sin contar la entrada y la salida, además de que 8 de estas corresponden a capas convolucionales.

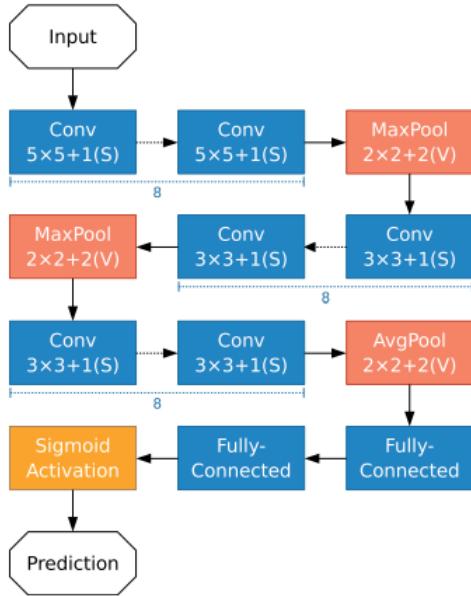


Figura 3.3: Arquitectura de la red convolucional utilizada por Javadi & Mirroshandel (2019). Principalmente se encuentra formada por capas convolucionales y de agrupación, terminando por las capas completamente conectadas y una función de activación sigmoidal, usada en clasificación binaria

Finalmente, en este trabajo, se logró obtener una exactitud 78,18%; 76,88%; y 88,31% para la clasificación morfológica, de los acrosomas, cabezas y vacuolas de los espermatozoides humanos. Estos resultados son el promedio simple de 5 iteraciones de validación cruzada. Al mismo tiempo, se logró obtener una sensibilidad promedio para la clase normal, de un 84,17%, 86,72%, 91,77% en la clasificación de los acrosomas, cabeza y vacuolas respectivamente, por lo que la capacidad de los modelos de reconocer efectivamente las formas "normales" fue bastante alta.

3.4 APRENDIZAJE PROFUNDO PARA LA CLASIFICACIÓN DE ESPERMATOZOIDES HUMANOS (2019)

El último trabajo de clasificación de cabezas de espermatozoides humanos, publicado antes del inicio de este experimento, es el escrito por Riordon et al. (2019), el cual enfrentó directamente la problemática de clasificación de cabezas de espermatozoides humanos, utilizando como herramienta las redes convolucionales, y no solo eso, si no que uno de los *datasets* que utilizó es el mismo ocupado por Chang et al. (2017b).

La red convolucional que utilizaron en Riordon et al. (2019) fue la **VGG16**, la que

se adaptó para soportar las imágenes de las cabezas de espermatozoides y para realizar una clasificación entre las cinco categorías de cabezas propuestas por la *OMS*, que se observan en la Figura 3.1.

VGG16 es una red convolucional profunda para reconocimiento de imágenes a gran escala, la cual ha alcanzado buenos resultados al ser puesta a prueba en desafíos como el llamado *ImageNet*, el cual corresponde a un conjunto de diez millones de imágenes de distintos tipos de objetos, el cual se espera clasificar cada una en 1.000 clases de objetos distintos. En este último conjunto de datos, la red convolucional *VGG16* ha alcanzado una exactitud del 92.7% (Simonyan & Zisserman, 2014).

La existencia de este trabajo ha sido de mucha sorpresa, ya que si bien la red *VGG16* ha sido entrenada con mil clases distintas, también es importante destacar que la clasificación de cabezas de espermatozoides es una tarea distinta a la clasificación hecha en *ImageNet*, ya que esta última presenta imágenes de la vida cotidiana en su mayoría, las cuales son diferentes en cuanto a forma, color, texturas y fondos, en comparación a lo que se puede presentar en una pequeña fotografía de una cabeza de espermatozoides.

El experimento de Riordon et al. (2019) utilizó validación cruzada en cinco iteraciones para evaluar la red, y antes de eso, utilizó la técnica de aumento de datos, no solo para aumentar la cantidad de imágenes en el entrenamiento, si no que también lo hizo para balancear las clases, ya que la categoría *amorphous*, tiene mucho mayor representatividad de imágenes que las otras cuatro clases.

Los resultados obtenidos utilizando el conjunto de acuerdo parcial de *SCIAND-MorphoSpermGS*, se ven reflejados en la Tabla 3.4, que se muestra la matriz de confusión del promedio de todas las iteraciones de validación cruzada que realizaron. En la Tabla 3.5 se han calculados algunas métricas como la exactitud y la tasa de verdaderos positivos, a partir de la matriz de confusión, para así realizar una mejor comparación con los resultados del trabajo anterior de Chang et al. (2017b), el cual se observan en la Tabla 3.3.

Con respecto a la clasificación en general, se logró aumentar la exactitud en la clasificación desde un 58% a un 62% aproximadamente, mientras que al comparar la tasa de verdaderos positivos entre las clases, existen algunas casos donde este experimento con redes convolucionales tuvo peores resultados con respecto al trabajo de clasificación con *SVM* de Chang et al. (2017b). Las clases *tapered* y *small* habían obtenido los mejores resultados en la clasificación usando *SVM*, pero con la red convolucional *VGG16* tuvo un poco más de dificultad al generalizar las características de estas formas de cabezas de espermatozoides. Por otro lado, las otras tres clases obtuvieron mejor resultados en la clasificación con la red convolucional, incluyendo la clase *amorphous*, en donde la mayor mejora fue la clase *pyriform*, en donde la tasa de verdaderos positivos aumento de un 50% a un 69% aproximadamente.

En general los resultados puede decirse que son bastante cercanos, y esto se ve reflejado en que la exactitud solo aumentó un 4% en general, e incluso se sigue teniendo el problema de generalizar las características de la clase *amorphous*, por lo que aún existe mucho espacio a mejora de estos resultados.

Este trabajo también realizó su experimento con el subconjunto en donde los tres expertos llegaron a la misma clasificación para una fotografía, pero en el documento no expusieron la matriz de confusión, pero si especificaron que con este conjunto, que tiene imágenes más fáciles de clasificar, obtuvieron una exactitud en la clasificación del 72%, el cual es un poco menor a los resultados obtenidos por Chang et al. (2017b), que se ven en la Tabla 3.3. A pesar de que los resultados mejoran cuando aumenta la calidad de los datos, se puede ver que la red convolucional *VGG16* no logra generalizar bien características en los casos que se pensaban más "sencillos" de clasificar, como las clases *small* y *tapered*, o el conjunto de acuerdo total.

Clase Verdadera / Clase Predicha	Normal	Tapered	Pyriform	Amorphous	Small
Normal	67,0	5,0	12,0	12,3	3,7
Tapered	10,5	56,6	1,8	20,9	10,1
Pyriform	11,2	1,4	68,9	16,5	1,9
Amorphous	18,5	10,3	15,5	38,3	17,5
Small	5,3	5,6	1,5	9,8	77,9

Tabla 3.4: Matriz de confusión según Riordon et al. (2019), utilizando el conjunto de acuerdo parcial.

Fuente: Elaboración propia, 2020.

Dataset/TPR	tpr(N)	tpr(T)	tpr(P)	tpr(S)	tpr(A)	Promedio
Dataset de acuerdo parcial	0,67	0,57	0,69	0,78	0,38	0,62

Tabla 3.5: Resumen de resultados de la clasificación según Riordon et al. (2019).

Fuente: Elaboración propia, 2020.

3.5 CLASIFICACIÓN MORFOLÓGICA, BASA EN APRENDIZAJE PROFUNDO, DE CABEZAS DE ESPERMATOZOIDES HUMANOS (2020)

Ahora bien, durante el mes de Mayo del año 2020, en medio del desarrollo de este trabajo, se ha publicado un nuevo trabajo sobre clasificación de cabezas de espermatozoides humanos, y esta vez utilizando redes convolucionales para realizar esta tarea. El experimento realizado por Iqbal et al. (2020), realizó la tarea de clasificación con dos conjuntos de datos distintos, de manera separada, en donde uno de estos corresponde al conjunto llamado *SCIAN-MorphoSpermGS*, el mismo utilizado por Chang et al. (2017b), y el experimento se realizó tanto como para el caso de acuerdo parcial y acuerdo total entre los expertos.

Los autores utilizaron una red convolucional de 53 capas, divididas en distintos bloques para que cada uno se encargue de generalizar ciertas características correspondientes a las cabezas de espermatozoides humanos, y la arquitectura final se puede observar en la Figura 3.4. Principalmente, se pueden identificar tres bloques convolucionales **A**, **B** y **C**, en donde el primero está compuesto de 9 capas convolucionales con el propósito de identificar características de la vacuola de la cabeza; en el segundo bloque se encuentran 12 capas para analizar el acrosoma del gameto, y en el último bloque se encuentran 18 capas para analizar el espacio sub-acrosomal de la célula en cuestión.

La arquitectura finalmente se encuentra formada por repeticiones del bloque **D**, el cual es una combinación de los tres bloques anteriores, seguido de un bloque **E**, en donde finalmente se termina en una capa completamente conectada para obtener la probabilidad de pertenencia de una imagen sobre las cinco clases de espermatozoides previamente definidas por la OMS (2010b).

Este trabajo realizó validación cruzada en cinco iteraciones de forma estratificada, mientras que para balancear las clases, utilizó la técnica de aumento de datos para generar un mayor número de imágenes en las clases con menor representatividad, y viceversa. Los resultados obtenidos en la clasificación, utilizando el conjunto de acuerdo parcial, se muestran en la Tabla 3.6, en forma de una matriz de confusión. Finalmente, para compararlos de mejor manera con los resultados obtenidos por Chang et al. (2017b), se presentan los resultados para ambos casos con la tasa de verdaderos positivos correspondiente, y la exactitud final en la clasificación. Los que se observan en la Tabla 3.7.

Al comparar estos últimos con la Tabla 3.3, se puede observar que el trabajo de Iqbal et al. (2020) mejoró los resultados en la clasificación utilizando redes convolucionales, en donde nuevamente se puede observar como los resultados son mejores con mejores datos (acuerdo total). En este caso también se puede ver como la tasa de verdaderos positivos también decrece en la categoría amorfa, pero se llegó a aumentar esta métrica desde un 30% (Chang et al., 2017b), a un 57% en el caso de acuerdo parcial, y un 72% para las imágenes de acuerdo total. Finalmente, la exactitud en la clasificación también aumentó, aproximadamente un 10% para ambos conjuntos de datos, por lo que en este experimento se ha logrado obtener mejores resultados utilizando este tipo de redes neuronales.

Clase Verdadera / Clase Predicha	Normal	Tapered	Pyriform	Amorphous	Small
Normal	70	3	3	20	4
Tapered	2	79	5	13	1
Pyriform	3	8	62	26	1
Amorphous	10	16	8	57	9
Small	10	2	1	16	71

Tabla 3.6: Matriz de confusión resultante según Iqbal et al. (2020) con el conjunto de acuerdo parcial.

Fuente: Elaboración propia, 2020.

Dataset/TPR	tpr(N)	tpr(T)	tpr(P)	tpr(S)	tpr(A)	Promedio
Dataset de acuerdo parcial	0,70	0,79	0,62	0,71	0,57	0,68

Tabla 3.7: Resumen de resultados de la clasificación según Iqbal et al. (2020).

Fuente: Elaboración propia, 2020.

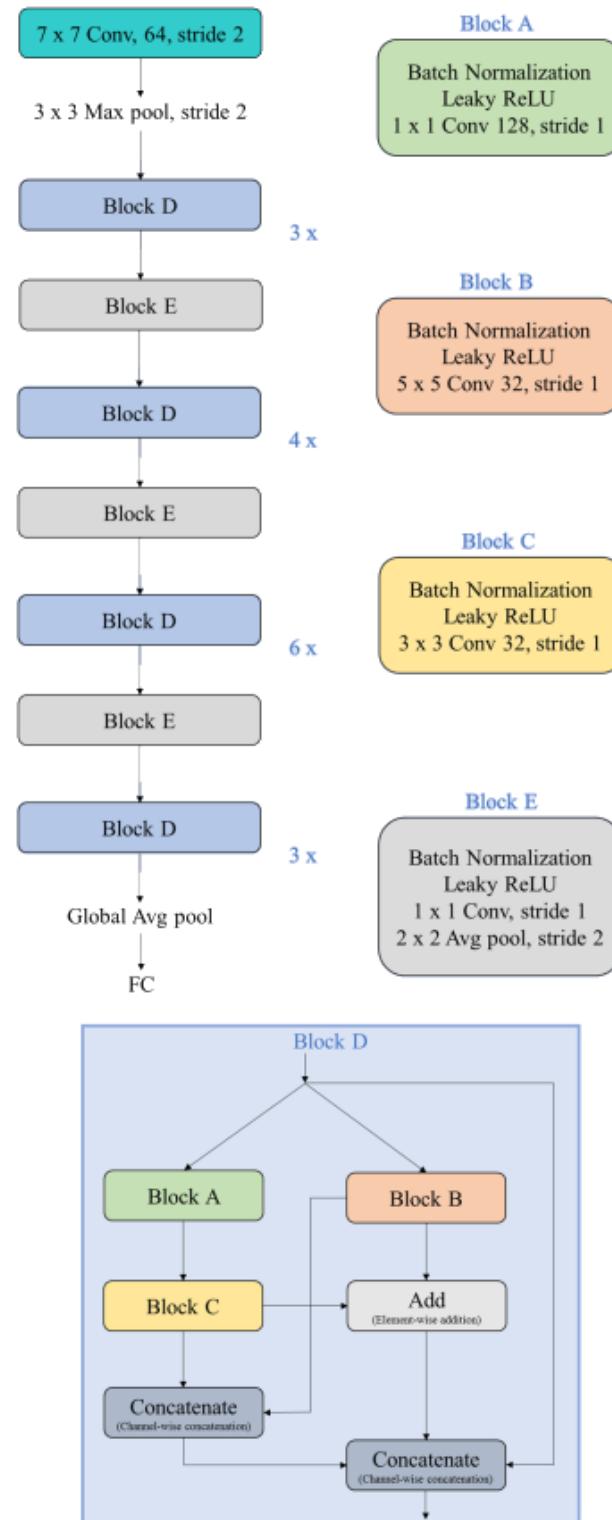


Figura 3.4: Red convolucional propuesta por Iqbal et al. (2020). El orden de las capas es representado por las flechas, y la estructura de los bloques se expresa de forma separada para facilitar la comprensión de la imagen.

3.6 RESUMEN

Utilizando el *dataset* de *SCIAN-MorphoSpermGS*, el trabajo de Chang et al. (2017b) intentó resolver el problema de clasificación de cabezas de espermatozoides, utilizando herramientas como *SVM* y Árboles de decisión, pero en ningún caso logró superar el 50% de exactitud en la clasificación de las cabezas. Seguido de esto, en el experimento de Chang et al. (2017b), se intentó clasificar las imágenes nuevamente con *SVM*, pero en dos procesos distintos de clasificación, para intentar discriminar de mejor manera la clase amorfa. La exactitud lograda alcanzó un 58% aproximadamente, pero la clase amorfa nuevamente presentó dificultades para generalizar sus características. El tercer trabajo presentado (Riordon et al., 2019), intentó resolver el problema, con el mismo *dataset*, y utilizando la red convolucional *VGG16* para clasificar las imágenes de espermatozoides humanos, pero solo logró aumentar la exactitud de la clasificación a un 62% aproximado. El trabajo más reciente (Iqbal et al., 2020), publicado en el transcurso del desarrollo de este trabajo, implementó una red convolucional propia, para discriminar las distintas partes de una cabeza de espermatozoide humano, logrando llegar a una exactitud del 68% en la clasificación de las imágenes.

CAPÍTULO 4. ELECCIÓN DE LA RED CONVOLUCIONAL

4.1 INVESTIGACIÓN SOBRE ARQUITECTURAS CONVOLUCIONALES

Como se mencionó en los objetivos, la creación desde cero de una red convolucional que haga la tarea de clasificación de imágenes, según la forma, es algo que no está contemplado, por lo que se ha estudiado en la literatura acerca de experimentos que realicen esta tarea utilizando redes convolucionales. Se espera que los documentos que traten este tema cumplan dos requisitos importantes. En primer lugar, que la red convolucional utilizada esté relacionada de alguna forma a la clasificación de cabezas de espermatozoides humanos, y segundo, que tenga una documentación completa de las capas de la *CNN* utilizada, o algún repositorio público con el código de la red que se haya usado en la clasificación, con el objetivo de replicar la misma red, pero ahora, en el etiquetado de las cabezas de los espermatozoides humanos.

El trabajo escrito por Xu et al. (2017) utiliza un tipo red convolucional, que discrimina principalmente por características morfológicas, para la clasificación de glóbulos rojos, en pacientes con la enfermedad de células falciformes. Los glóbulos rojos de los sujetos con esta enfermedad, tienen distintas formas que revelan características importantes de la célula misma, como su densidad, fragilidad, propiedades adhesivas, entre otros. Obtener la mejor forma de cuantificación y clasificación (*gold standard*) en este problema ayudaría a un mejor pronóstico de la enfermedad.

Actualmente existen múltiples algoritmos de detección de células, tanto para imágenes en dos y tres dimensiones. Sin embargo, la clasificación de estas es un problema más complicado en el caso de estos glóbulos rojos, debido a las formas heterogéneas que pueden presentar, las cuales se pueden ver en la Figura 4.1.

El *dataset* utilizado se obtuvo a partir de muestras de ocho pacientes con la enfermedad de anemia de células falciformes (o *sickle cell disease*) del Hospital General de Massachusetts y del Centro Médico de la Universidad de Pittsburgh, en acuerdo con la junta de revisión institucional. Las muestras de sangre se almacenaron a 4°C y se utilizaron dentro de los 3 días posteriores a la extracción de sangre (Xu et al., 2017). El conjunto esta compuesto por aproximadamente 7.000 imágenes, que se pueden observar como ejemplo en la Figura 4.1.

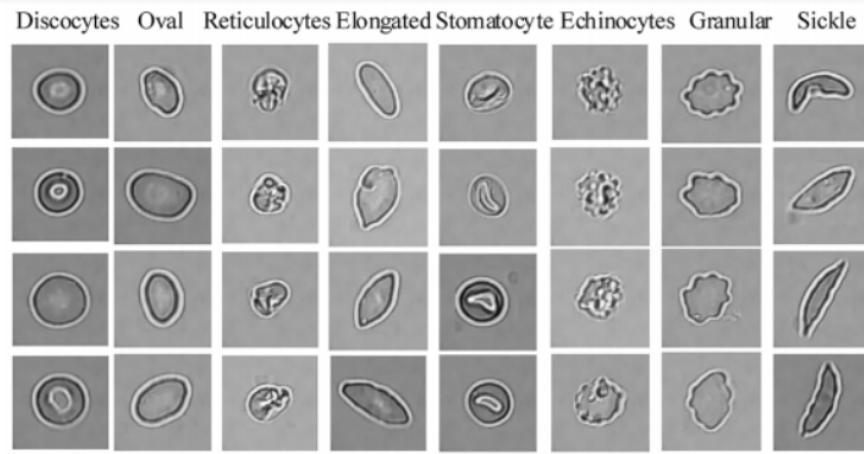


Figura 4.1: Formas de glóbulos rojos (Xu et al., 2017). Cada columna representa ejemplos de una de las clases de glóbulos rojos.

Los métodos utilizados en este trabajo son similares a los presentados en la parte de evaluación de espermatozoides, en donde se utiliza una técnica para obtener el área de interés, para luego realizar la clasificación con la red convolucional. El método de extracción de las imágenes de los glóbulos rojos viene acompañada de una etapa de normalización, en donde las imágenes de las células son ajustadas a un tamaño de 100x100 píxeles, llevadas al centro de la imagen y luego se ajusta el tamaño de estas.

La red convolucional que los autores utilizaron no tiene ningún nombre en específico, y se compone de tres capas convolucionales (C1, C3 y C5), tres capas de *pooling* (P2, P4 y P6) (submuestreo), dos de regularización o *dropout* (D7 y D9), y una capa completamente conectada (F8). El orden de las capas anteriores se pueden observar en la Figura 4.2

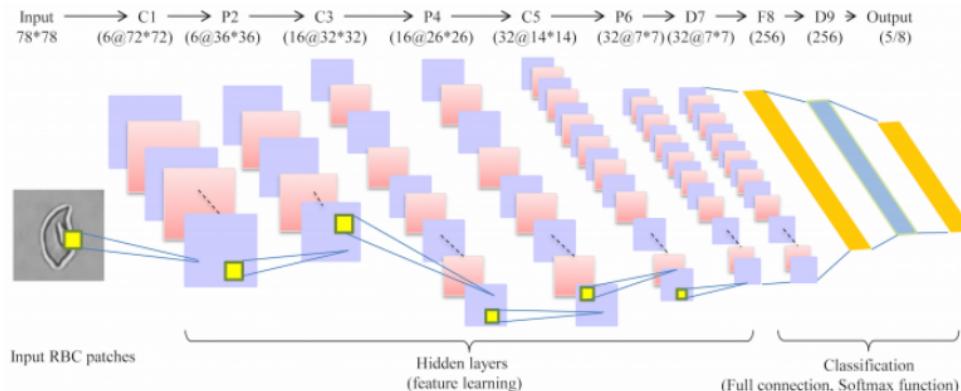


Figura 4.2: Estructura de red neuronal convolucional usada por Xu et al. (2017). En la parte superior está representada las capas de la red, compuesta por las capas convolucionales C, agrupación P, regularización D, y la completamente conectada F.

Se realizaron dos procesos de clasificación, uno con cinco clases y un segundo con ocho clases, en donde se utilizó validación cruzada de cinco iteraciones (*5 fold*), obteniendo así la media de las medidas de evaluación correspondientes.

En la primera clasificación, se juntaron formas de glóbulos rojos hasta obtener cinco clases distintas, las cuales son **Dic+Ovl; Ech; El+Sk; Grl y Ret** (abreviación de las formas vistas en la Figura 4.1). Los resultados muestran que la exactitud promedio en el entrenamiento de las cinco clases es del 91,01%, mientras que en la evaluación, esta fue de un 89,28%.

En el segundo proceso de clasificación, se utilizaron las ocho clases originales, tal como se ven en la Figura 4.1. Los resultados obtenidos, para el caso de la exactitud promedio en el entrenamiento, fue de un 89,69%, mientras que por el lado de la evaluación, se alcanzó una exactitud del 87,50%.

Por otro lado, se encontró otro documento que también realiza una clasificación de células de glóbulos rojos con respecto a su morfología. El trabajo de Angluin et al. (2016) usa una red convolucional para clasificar las imágenes en diez formas distintas, en donde cada una está relacionada con alguna patología que puedan tener los eritrocitos. La clasificación en este caso, también está basada en la forma de los glóbulos, y estos se pueden observar en la Figura 4.3, junto con la cantidad de imágenes existentes en el conjunto de datos por cada clase.

Para el preprocesamiento de las imágenes, se llevaron los glóbulos rojos al centro de las imágenes de 70x70 píxeles, obteniendo aproximadamente 750 muestras de este tipo de células. Luego de esto, se realizó un proceso de aumento de datos igualmente, para entrenar la red convolucional, alcanzando una cantidad aproximada de 4.000 imágenes.

Class	Example Picture	Number	Class	Example Picture	Number
Normal		203	Acanthocyte		33
Echinocyte		63	Target Cell		145
Dacrocyte		18	Stomatocyte		22
Schistocyte		152	Spherocyte		48
Elliptocyte		18	Overlap		46

Figura 4.3: Formas de glóbulos rojos por patología (Angluin et al., 2016), y cantidad de imágenes por clase en el conjunto de datos.

La red convolucional que los autores plantearon, está compuesta de Diecisiete capas incluyendo la entrada, de las cuales se encuentran nueve capas convolucionales, tres de regularización, tres de *pooling*, y una última completamente conectada. El detalle de cada capa y el orden se puede observar en la Figura 4.4.

Layer	Specification
Input Image	shape = (3 x 70 x 70)
2D Convolution	num nodes = 32, filter = (3 x 3), activation = ReLU
2D Convolution	num nodes = 32, filter = (3 x 3), activation = ReLU
2D Convolution	num nodes = 32, filter = (3 x 3), activation = ReLU
Max Pooling	pool size = (3 x 3), strides = (2 x 2)
Dropout	0.25
2D Convolution	num nodes = 64, filter = (3 x 3), activation = ReLU
2D Convolution	num nodes = 64, filter = (3 x 3), activation = ReLU
2D Convolution	num nodes = 64, filter = (3 x 3), activation = ReLU
Max Pooling	pool size = (3 x 3), strides = (2 x 2)
Dropout	0.25
2D Convolution	num nodes = 128, filter = (3 x 3), activation = ReLU
2D Convolution	num nodes = 128, filter = (3 x 3), activation = ReLU
2D Convolution	num nodes = 128, filter = (3 x 3), activation = ReLU
Max Pooling	pool size = (3 x 3), strides = (2 x 2)
Dropout	0.5
Dense Layer	10 dimensions, activation = softmax
Optimizer	Stochastic Gradient Descent
Loss function	Categorical Cross Entropy

Figura 4.4: Red convolucional planteada por Angluin et al. (2016). con la entrada en la parte superior, y siguiendo un orden descendente, hasta terminar con el optimizador y la función de pérdida utilizada.

En cuanto a los datos utilizado para la clasificación, se separó el *dataset* en un 80% y 20% para los conjuntos de entrenamiento y de pruebas respectivamente. Al mismo tiempo, el conjunto de entrenamiento también se separó en un 80% para un subconjunto de entrenamiento, y el restante 20% en un subconjunto de validación. Como la cantidad de imágenes es dispar entre clases (por ejemplo, 203 imágenes de glóbulos normales en comparación a 18 de tipo *Dacrocyte*), para la creación de los conjuntos mencionados en el párrafo anterior, se utilizó un muestreo estratificado.

Los resultados muestran una exactitud del modelo de un 97,59% en la predicción de alguna de las diez clases mostradas, lo cual es bastante alto. El algoritmo tuvo problemas en la clasificación para los tipos de glóbulos rojos que tenían menos cantidad de imágenes en la muestra.

A partir de los dos documentos anteriores mencionados, se ha decidido como objetivo adaptar y trabajar con la red convolucional propuesta por Angluin et al. (2016) para la realización de la clasificación de cabezas de espermatozoides según su morfología, debido a que los resultados fueron mejores en este trabajo, en comparación a los resultados de Xu et al. (2017). Además, la red convolucional usada en este caso está mejor documentada, y al mismo tiempo

existe un repositorio abierto en *Github* asociado a este documento, en donde se encuentra el código fuente con la arquitectura de esta *CNN*.

Desde otro punto de vista, en el estado del arte se ha visto como otros investigadores han utilizando redes convolucionales, con imágenes de contextos diferentes a la clasificación de espermatozoides, o alguna otra forma que se le parezca, como fue el experimento que utilizó la red *VGG16* para el problema de discriminar la forma de las cabezas de espermatozoides. Es por esta razón que se ha decidido utilizar otra red convolucional que se haya publicado y que no necesariamente tenga registros de clasificación según la forma de los objetos que se presentan en las imágenes.

En este caso se ha decidido utilizar la red convolucional propuesta llamada *ResNet*, la cual se encuentra ya implementada en la librería de *Keras* y *TensorFlow*, por lo que solo habría que adaptarla al problema actual y entrenar con ella. La versión de esta red convolucional a utilizar es una de las primeras, que lleva el nombre de **ResNet50**, la cual lleva este nombre porque su arquitectura comprende de 49 capas convolucionales y una completamente conectada. Se ha escogido esta debido a que se encuentra disponible en la versión de *Tensorflow* 12.0, que es la última versión de esta librería que es compatible con el hardware utilizado para la realización de este trabajo.

La red convolucional *ResNet* es también conocida como red convolucional residual (*residual network*), y comprenden una serie de arquitecturas que tienen la característica de ser profundas, o sea, que su construcción comprende de múltiples capas. En comparación, la red *VGG16* comprende 16 capas totales, entre convolucionales, entradas, activaciones, entre otras, mientras que la red específica de *ResNet50* comprende más de 170 capas, en donde 49 de estas son convolucionales, una completamente conectada, y el resto comprenden activaciones, agrupamientos y normalizaciones.

La profundidad de la red trae como consecuencia, en general, que la propagación del error se va perdiendo a lo largo de la red al volver a calcular los pesos de las neuronas al finalizar una época. Esto quiere decir que, las neuronas de las primeras capas podrán cambiar sus pesos de manera más lenta que las neuronas de las capas intermedias o las finales (He et al., 2015).

La característica principal de la red *ResNet* ayuda a combatir el problema anterior, y consiste en que las capas convolucionales no solo reciben los resultados de la capa anterior, si no que también reciben los resultados de capas anteriores, o incluso la entrada misma de la red. Es este concepto de entregar "residuos" a capas posteriores lo que le da el nombre a esta arquitectura. En la Figura 4.5, se observa la forma de una *ResNet* de 34 capas convolucionales residuales, en donde, la arquitectura deja de ser secuencial, como se pudo observar en el trabajo de Angluin et al. (2016) con su red convolucional expuesta en la Figura 4.4, si no que en esta

ocasión, existe información que se va entregando a capas posteriores (He et al., 2015).

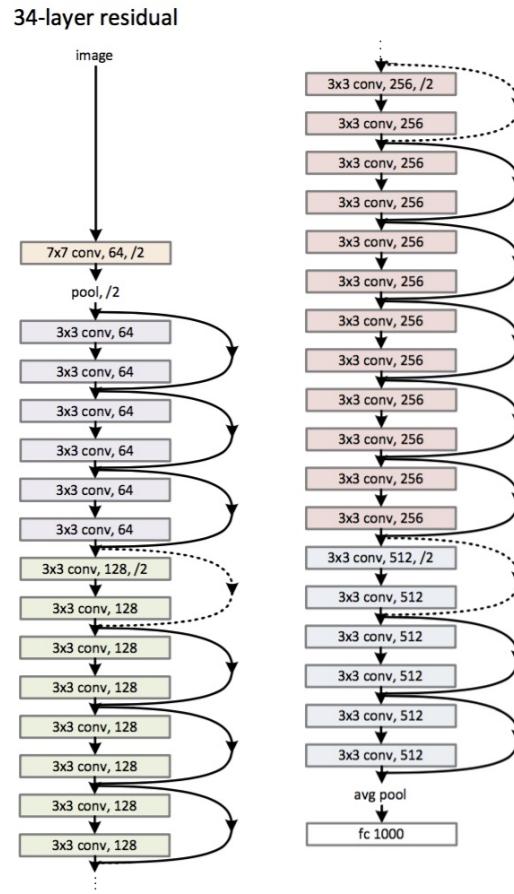


Figura 4.5: Ejemplo de red residual de 34 capas convolucionales (He et al., 2015), donde también se ve como el resultado de una capa, se va traspasando a etapas posteriores

Los resultados presentados muestran, que este acercamiento utilizando el concepto de los residuos, logró disminuir la tasa de error en comparación a arquitecturas secuenciales, o incluso a las redes *VGG16*. De hecho, en el desafío de *ImageNet*, se logró obtener el primer lugar con este tipo de arquitecturas, lo que ha demostrado que las redes profundas pueden obtener mejores resultados que su contraparte.

La *ResNet50* es la versión menos profunda implementada en *Keras*, con un total de 177 capas, pero de igual manera, es mucho más profunda que la red de Angluin et al. (2016) o la *VGG16*, lo cual hace que el entrenamiento sea mucho más lento. Debido a la longitud de la arquitectura de la *ResNet50*, que hace difícil colocar una imagen de las capas de la componente, pero la documentación de Chollet et al. (2015) hace énfasis que la implementación de esta sigue la publicación de He et al. (2015). En pocas palabras, la gran diferencia de su composición es la utilización de la técnica de los residuos que se ha hablado, y la aplicación de capas de normalización.

Finalmente, los experimentos se llevaran a cabo en las dos redes seleccionadas, en donde la primera corresponde a la red convolucional de Angluin et al. (2016), que fue utilizada para clasificar glóbulos rojos, y es poco profunda, y por otro lado, se tiene la red convolucional *ResNet50*, la que supero a *VGG16* en el desafío de clasificación de *ImageNet*, y que al contrario de la anterior, esta es más profunda en cuanto a capas se refiere.

4.2 RESUMEN

Se ha escogido utilizar la red convolucional propuesta por Angluin et al. (2016), debido a que realizó clasificación de imágenes de células de glóbulos rojos según la forma que estos pueden tener, obteniendo buenos resultados en la clasificación, considerando que también se tenían pocas imágenes para el experimento, y se tuvo que realizar aumento de datos para resolver este problema.

En segundo lugar, el trabajo de Riordon et al. (2019), utiliza la red *VGG16* pre-entrenada con imágenes del desafío de *ImageNet*, para clasificar las imágenes de cabezas de espermatozoides humanos, logrando mejorar un poco el estado del arte. Es por esta razón que se escogió como segunda arquitectura, la red convolucional *ResNet50*, ya que ha tenido mejores resultados que la *VGG16* en clasificación de imágenes, por lo que esto podría sugerir que los resultados de *ResNet50* también podrían ser mejores en la clasificación de cabezas de espermatozoides humanos.

CAPÍTULO 5. DESARROLLO DE LA INVESTIGACIÓN

Para el desarrollo del experimento, hay muchas decisiones que deben tomarse para resolver a los objetivos específicos planteados. Es por esto, que en este capítulo se presenta la investigación realizada, para el cumplimiento de estos. El orden de la investigación se basó en los objetivos específicos planteados en el Capítulo 1, comenzando por la descripción de las imágenes usadas para la clasificación de las cabezas de espermatozoides.

5.1 DESCRIPCIÓN DE LOS DATOS

El conjunto de datos que se utiliza como base en esta investigación, corresponde al *dataset* público llamado **SCIAN-MorphoSpermGS** (Chang et al., 2017a), el cual está formado por imágenes de cabezas de espermatozoides, obtenidas de muestras de semen de voluntarios entre 28 y 35 años, en el Laboratorio de Espermiogramas del Instituto de Ciencias Biomédicas (ICBM), de la Universidad de Chile, en Santiago de Chile. Este es el mismo conjunto de imágenes utilizado por los trabajos presentados en el estado del arte (Chang et al., 2017a,b; Riordon et al., 2019; Iqbal et al., 2020).

El *dataset* contiene 1.854 imágenes de cabezas de espermatozoides observables, evaluables y categorizadas dentro de una de las cinco siguientes clases de cabezas: normal; cónica (*tapered*); piriforme (*pyriform*); pequeña (*small*) y amorfa (*amorphous*), las que se pueden ver en la Figura 3.1. La clasificación manual de todas estas imágenes fue realizada de forma independiente, por tres expertos chilenos con reconocida experiencia en el análisis morfológico de cabezas de espermatozoides. En la Figura 5.1 se pueden observar ejemplos de cabezas de espermatozoides de cada clase, presentes en este conjunto de datos. A simple vista se puede observar que, para alguien que no es experto, clasificar estas imágenes no es una tarea sencilla.

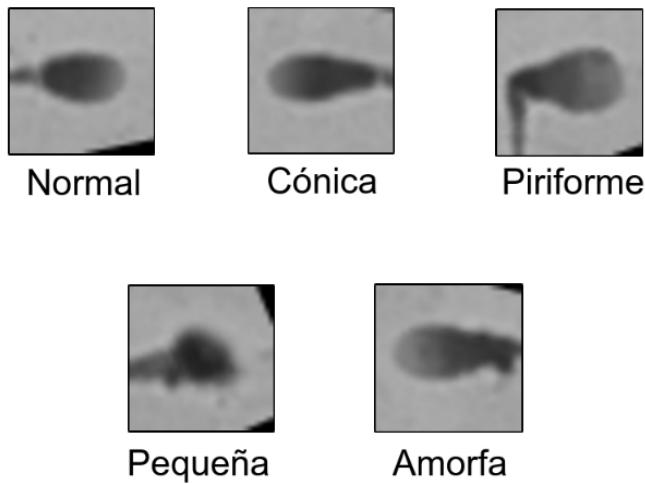


Figura 5.1: Ejemplos de las clases en el conjunto de **SCIAN-MorphoSpermGS**

Fuente: Elaboración propia, 2020.

Debido a que las muestras fueron evaluadas por tres personas distintas, esto permite que existan tres escenarios de acuerdo entre los expertos. El primero corresponde al caso base, compuesto por las 1.854 cabezas, las que fueron clasificadas en al menos una de las 5 clases distintas. El segundo es el caso en que dos o tres expertos llegaron a una misma clasificación para una cabeza (acuerdo parcial), y el último escenario corresponde a aquellas imágenes en las que los tres expertos llegaron a la misma conclusión (acuerdo total). Cada caso contiene a los siguientes, por lo que, el escenario de acuerdo total es un subconjunto del de acuerdo parcial, y el acuerdo parcial es un subconjunto del total de imágenes. La descripción total de las clases por escenario se ve reflejado en la Tabla 5.1. En el transcurso de este trabajo, se ha utilizado el conjunto de acuerdo parcial, en donde la etiqueta escogida para las imágenes de este último escenario, está definida por mayoría de votos de los expertos, por lo que solo se han utilizado 1.132 imágenes de este *dataset*.

Escenario/Clase	Normal	Cónica	Piriforme	Pequeña	Amorfa	Total
Al menos uno	175	420	188	152	919	1854
Acuerdo parcial	100	228	76	72	656	1132
Acuerdo total	35	69	7	11	262	384

Tabla 5.1: Escenarios de acuerdo en el conjunto de imágenes de espermatozoides

Fuente: Elaboración propia, 2020.

A partir de una observación inicial sobre conjunto de datos, se puede hacer un pequeño análisis acerca de las imágenes y las clases que lo componen. Primero que todo, se observa una baja variabilidad inter-clase, en donde, un ejemplo de esto se puede observar de inmediato en la Figura 5.1, al comparar en este caso, la similitud entre la cabeza de clase normal

y la cabeza de la clase cónica.

Al mismo tiempo, en las imágenes se puede observar que existe una alta variabilidad intra-clase, que se puede observar como ejemplo, en la Figura 5.2, en donde las tres imágenes correspondientes a la clase cónica, y tres imágenes de la clase amorfa, en donde se ven algunas diferencias entre imágenes de la misma forma de cabeza de espermatozoide, especialmente en el caso de la clase amorfa. La combinación de baja variabilidad inter-clase y alta variabilidad intra-clase caracterizan a un problema de clasificación automática como un problema complejo y de difícil solución.

Por otro lado, las imágenes de espermatozoides fueron obtenidas a partir de muestras observadas desde microscopio, por lo que el tamaño de las figuras solo alcanzan 35 píxeles de alto y ancho, además de contener ruido producto de la magnificación hecha para visualizar las muestras.

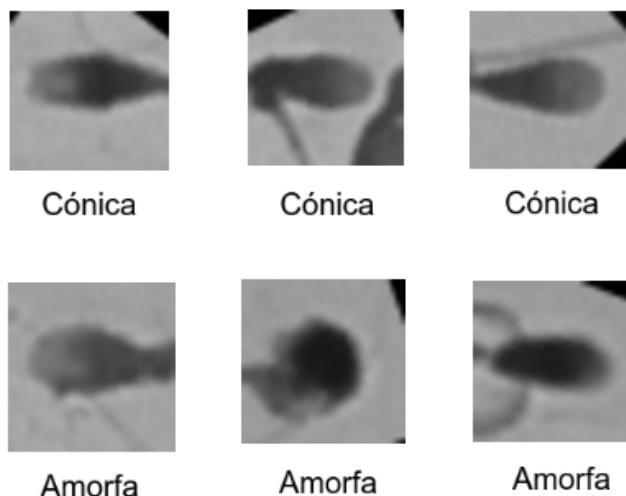


Figura 5.2: Ejemplos de imágenes diferentes en la misma categoría. Acá se observa como pueden existir diferencias incluso entre imágenes que pertenecen a la misma categoría.

Fuente: Elaboración propia, 2020.

También existen otros problemas que ya no tienen que ver con el contenido de la imagen, si no con la construcción del conjunto de datos. La cantidad total de imágenes no es la suficiente para entrenar modelos complejos con *deep learning*, lo que se debe principalmente al costo necesario para obtener y clasificar las muestras. Además, las clases se encuentran desbalanceadas, en donde la clase amorfa es más predominante que las otras cuatro.

5.2 AUMENTO DE LOS DATOS

5.2.1 Preprocesamiento y lectura de los datos

Antes de trabajar con las imágenes, es importante revisar el conjunto a utilizar, y cómo estas se encuentran etiquetadas por los expertos. Las fotos de las cabezas se encuentran en formato *TIF*, con una dimensión de 35×35 . Las imágenes se encuentran en formato *RGB*, lo que quiere decir que cada píxel de una imagen representa un arreglo de tamaño tres, el cual indica los valores de color rojo, verde y azul, dando como resultado el color real que el observador va a ver en este píxel. Teniendo esto en cuenta, finalmente los archivos de imágenes serán arreglos de tres dimensiones correspondientes a $35 \times 35 \times 3$.

La lectura de las imágenes se realiza utilizando la librería **PIL** disponible para *Python 3.6*, la cual se encarga de leer una imagen primero a un arreglo de 35×35 , para luego convertirlas a formato *RGB* y tener el arreglo final correspondiente a $35 \times 35 \times 3$. **PIL** realiza el manejo de estos arreglos con la librería **Numpy** por defecto.

El archivo de la clasificación hecha por los expertos se encuentra en un archivo de texto, en donde cada línea contiene el nombre de una imagen del conjunto; las clases escogidas por los tres expertos distintos, y finalmente la clase correspondiente a la mayoría de votos, todos separados de una tabulación. Lamentablemente, el nombre de las imágenes en este archivo de texto no es idéntico al que tienen los archivos de imágenes. El archivo de texto y el nombre de las imágenes tienen algunos caracteres distintos, como se muestren en el siguiente ejemplo.

- Texto: **p1-pl2-sample07/Sperm_11**
- Nombre imagen: **ch00_p1-pl2-sample7-sperm11**

A pesar de esto, se puede observar que la base de los nombres es la misma, por lo que al momento de leer el archivo de texto, se realiza un reemplazo de caracteres para asimilar el nombre de los archivos correctamente. Finalmente, el nombre de las imágenes con su respectiva clase es guardada en una estructura de diccionario que ofrece el lenguaje *Python*. En el conjunto de imágenes, no se encontró ningún archivo corrupto o con contenido inconsistente (imágenes que no fueran cabezas de espermatozoides) que fuera necesaria eliminar, por lo que la cantidad de imágenes a usar antes del aumento de datos, sigue siendo de 1.132.

5.2.2 Limpieza de los datos

En la misma Figura 5.2, se ven ejemplos de ciertos bordes negros presentes en las imágenes, especialmente en las esquinas de estos. Las redes convolucionales escogidas, deben generalizar las características de las formas de las cabezas de espermatozoides, pero estos píxeles negros no corresponden a características de la forma las cabezas de los espermatozoides. Considerando también que se hace un proceso de aumento de datos previo al entrenamiento, es importante eliminar estos píxeles no relevantes, antes de propagarse por las imágenes aumentadas durante el entrenamiento de los modelos. Esto se hace debido a que, la presencia de estos bordes negros puede hacer que los modelos generalicen estos bordes sobre una o más clases de cabezas, entregando malos resultados en la evaluación final de los modelos.

Esto se hace de forma automática con un algoritmo en *Python*, creado específicamente para este grupo de imágenes, en donde se buscan los píxeles más cercanos al color negro, y así rellenarlos con el color de fondo más cercanos a estos. Sin embargo, el rango de píxeles negros elegidos a reemplazar no es muy extenso, para así no intervenir en los píxeles correspondientes a las cabezas de espermatozoides humanos. Debido a esto, la corrección automática de los bordes negros no es perfecta en su totalidad, ya que aún después de utilizar el algoritmo, existen casos en que de todas formas existen algunos píxeles cercanos a negros en las esquinas, que no fueron corregidos.

Para compensar esto, se analizan de manera manual todas las imágenes del conjunto de acuerdo parcial luego de la corrección automática, y así observar cuales imágenes aún conservan bordes negros. Estas imágenes son corregidas de forma manual, llenando de la misma forma, los bordes negros con el color de los píxeles más cercanos. Si bien, es una tarea algo extensa, no es imposible de hacer, debido al moderado tamaño de este conjunto de imágenes.

En la Figura 5.3 se muestra un ejemplo de la corrección realizada de las imágenes de espermatozoides humanos, eliminando los bordes negros antes de trabajar sobre estas. Cabe destacar que ni las dimensiones ni los formatos de las imágenes deben ser alteradas durante este proceso, y lo mismo se puede decir de los píxeles correspondientes a las cabezas de espermatozoides humanos, ya que estos definen la morfología de estos.

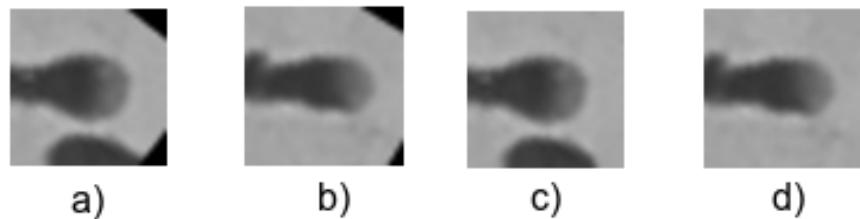


Figura 5.3: Ejemplos de corrección de bordes negros. Las imágenes a) y b) fueron corregidas para eliminar las esquinas negras, obteniendo c) y d) respectivamente.

Fuente: Elaboración propia, 2020.

5.2.3 Proceso de aumento de datos

La clasificación morfológica de las cabezas de espermatozoides depende de, valga la redundancia, la forma de estas, por lo que el aumento de datos es realizado de manera inteligente, para así no afectar la morfología de las cabezas de espermatozoides presentes en las imágenes. En el caso de alterar la forma de las cabezas de espermatozoides, se introducirían características que no pertenecen a ninguna clase, o incluso, podrían desaparecer características que sí corresponden a alguna de las cinco clases de cabezas de espermatozoides humanos. Por ejemplo, si se cambia la razón de aspecto o si se realiza un recorte de una imagen, se pierden características de la forma de la cabeza, como se puede observar en la Figura 5.4.

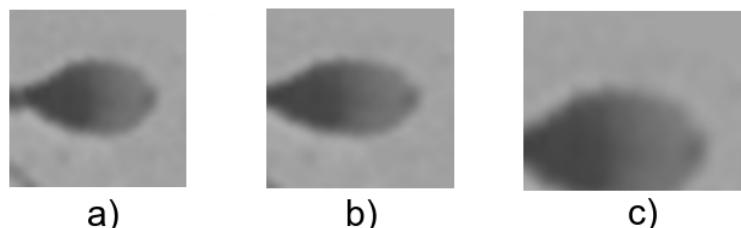


Figura 5.4: Ejemplos erróneos de aumento de datos. En a) se tiene la imagen original, en b) y c) se ven transformaciones de a) en donde se pueden perder partes de la cabeza del espermatozoides.

Fuente: Elaboración propia, 2020.

En la investigación, se ha encontrado el trabajo de Javadi & Mirroshandel (2019), en donde su problema consiste en identificar también, formas anormales de espermatozoides. En este caso, se refiere a la identificación de cabezas, vacuolas y acrosomas normales y anormales, a partir de imágenes de espermatozoides humanos. Los autores utilizaron una red neuronal convolucional para realizar esta tarea, sobre un *dataset* de alrededor de 1500 imágenes. Debido a

la poca cantidad de imágenes, se utilizó aumento de datos para entrenar la red convolucional que ellos propusieron, lo cual también se presentó como un problema, ya que este proceso tampoco debe afectar la forma de las imágenes de utilizar. Es por esto que se propuso las siguientes transformaciones, con el objetivo de evitar el problema anterior. En la Figura 5.5 se ve el resultado de las imágenes generadas en ese trabajo.

- Recorte aleatorio y zoom de imágenes
- Reflexión vertical
- Reflexión horizontal
- rotación de una imagen
- Ajuste de brillo

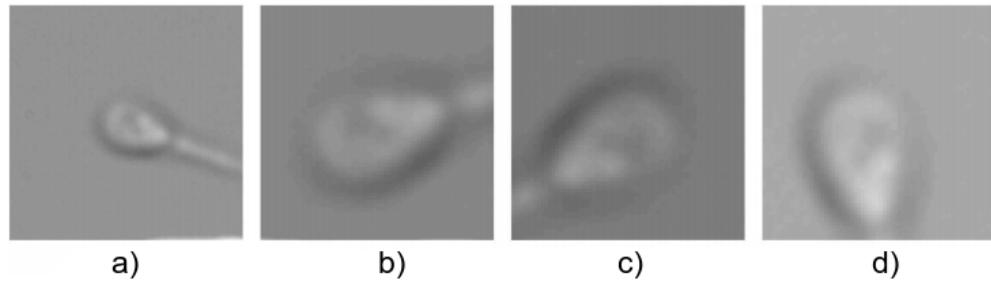


Figura 5.5: Ejemplo de aumento de datos realizado por Javadi & Mirroshandel (2019). La imagen a) es la muestra original, mientras que en b), c) y d) se aplicaron las transformaciones como rotación, cambio de brillo, rotación, entre otras.

El recorte aleatorio realizado se ha hecho en cierta medida para no perder la información de alguna parte de la cabeza del gameto masculino. Sin embargo, al comparar la imagen original de la Figura 5.5, con las que se observan en la Figura 5.1, se puede observar que las fotografías del conjunto *SCI/AN-MorphoSpermGS* se encuentran mucho más acotadas en comparación, por lo que para el experimento actual, no es adecuado realizar un recorte aleatorio de la imagen, ya que podría pasar a llevar partes del área de interés, que es la cabeza del espermatozoides. Es por esto que se ha decidido descartar esta transformación, por lo que finalmente en este trabajo se realizan cuatro transformaciones para el objetivo de aumento de datos.

Otro cambio que podría haberse realizado en las imágenes, es la inclusión de ruido en las fotografías, pero no se utilizó esto debido a que, ni en el trabajo de Angluin et al. (2016) ni de Javadi & Mirroshandel (2019) se realizó este tipo de transformaciones en el aumento de datos, y además, la librería de *Keras* trabaja al introducir ruido y error sobre las imágenes al

configurar capas que realizan esta operación en la arquitectura de la red convolucional. En este caso, Angluin et al. (2016) no utilizó capas de ruido *Gaussiano*, si no que utilizó capas de *dropout* para introducir error en el entrenamiento. Debido a esto, se ha descartado en esta etapa introducir ruido directamente en las imágenes, durante el aumento de datos.

Siguiendo la investigación para verificar cuáles otras modificaciones podrían hacerse sobre las imágenes, en el trabajo de Iqbal et al. (2020), que también utilizó redes convolucionales para clasificar la forma de las cabezas de espermatozoides del **dataset SCIAN-MorphoSpermGS**, se trasladaron las imágenes de forma vertical y horizontal en un 6% aproximadamente, durante el aumento de datos, para no perder información de la forma de las cabezas de espermatozoides. Durante esta transformación, los bordes generados al mover la imagen, corresponden a una media de los píxeles cercanos, para así mantener el tamaño de 35×35 píxeles de las imágenes. En la Figura 5.6 se observan ejemplos de mover imágenes un 6% de manera vertical y horizontal, donde se ve que no se han perdido partes del área de interés de la fotografía.

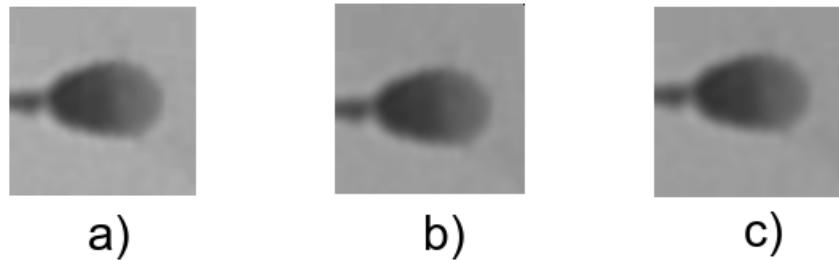


Figura 5.6: Ejemplos de trasladar una imagen. En a) se tiene una imagen original, mientras que en b) y c) se tienen ejemplos en donde se trasladó la imagen original, de manera horizontal y vertical respectivamente, en un 6% aproximadamente, con respecto a la dimensión original de 35×35 píxeles.

Fuente: Elaboración propia, 2020.

La reflexión vertical y horizontal corresponde a dar vuelta una imagen con respecto al eje horizontal y vertical respectivamente. Esta transformación mantiene las dimensiones de la figura original, por lo que no se ve afectada la forma de las cabezas de espermatozoides. En la Figura 5.7 se ve el resultado de aplicar esta operación sobre una imagen del conjunto de acuerdo parcial.

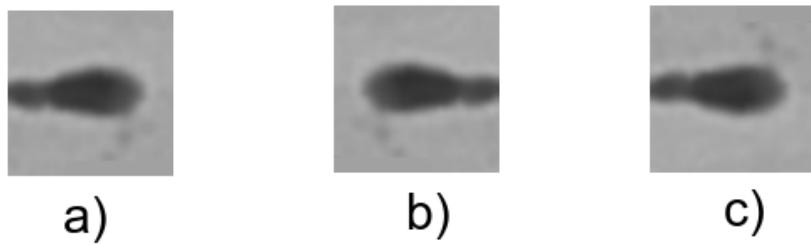


Figura 5.7: Ejemplos de reflexión vertical y horizontal. En a) se ve la imagen original, y en b) y c) se ve el resultado de una reflexión vertical y horizontal respectivamente.

Fuente: Elaboración propia, 2020.

La rotación de una imagen se realiza de manera aleatoria entre 0° y 360° . Al trabajar con imágenes cuadradas de 35×35 , la dimensión de estas no se ve afectadas al realizar esta operación con ángulos múltiplos de 90° , mientras que para el resto de los posibles ángulos, también es importante preservar la dimensión del dato, para utilizarla en la red convolucional. Al rotar en ángulos distintos a múltiplos de 90° , las esquinas de la foto original van a verse truncados, y nuevos píxeles vacíos aparecerán, dependiendo del ángulo de rotación. En la Figura 5.8 se puede observar como, al rotar una foto del espermatozoide, los bordes sobresalen la dimensión original, mientras que aparecerán píxeles sin color. Para terminar la rotación de forma correcta, los bordes que sobresalgan son descartados, mientras que los nuevos fondos son rellenados con información similar a la de los píxeles vecinos, que corresponden al color de fondo de las imágenes.

El último cambio corresponde al ajuste de brillo de la imagen, en donde se aclarará o oscurecerá una foto de espermatozoide de forma aleatoria y en cierta medida, de tal modo que no se pierdan los bordes de una cabeza de espermatozoide, tal como aparece en la Figura 5.9.

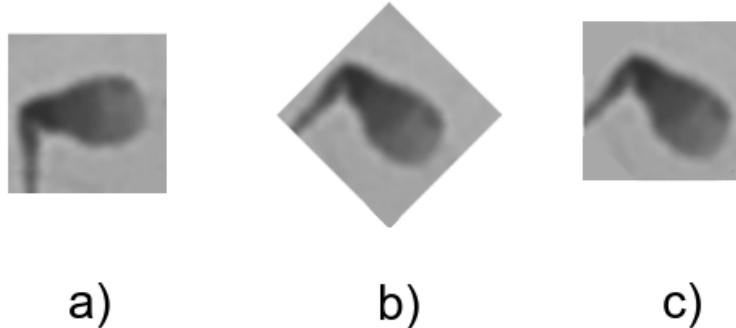


Figura 5.8: Ejemplos de rotación de una cabeza de espermatozoide. En a) se ve la imagen original, mientras que en b) se ve una rotación realizada, pero sin ajustar la dimensión, mientras que c) se tiene la misma rotación realizada en b), pero aplicando la dimensión de la imagen original a), cortando los bordes de exceso y rellenando los bordes faltantes, con los píxeles más cercanos.

Fuente: Elaboración propia, 2020.

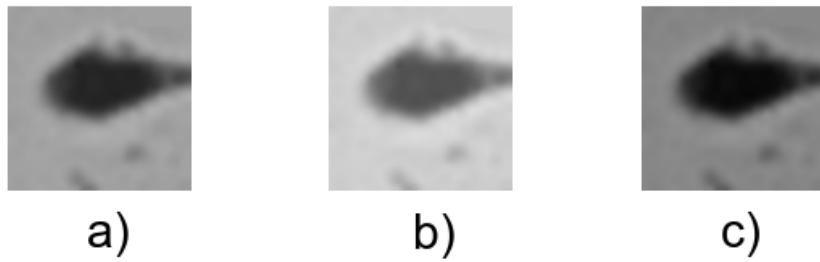


Figura 5.9: Ejemplos de cambio de brillo de una cabeza de espermatozoide. En a) se tiene la imagen original, mientras que en b) y c) se tiene el ajuste de aumento de brillo y de disminución de brillo, respectivamente.

Fuente: Elaboración propia, 2020.

Finalmente, el proceso completo de aumento de datos se realiza combinando todos los ejemplos anteriores, sobre el conjunto de imágenes que se utilizarán en el entrenamiento de la red. Cabe destacar que, la cantidad de imágenes a generar a partir de una sola puede afectar el entrenamiento de la red, ya que es posible que las imágenes generadas sean muy parecidas para el modelo a entrenar, utilizando solo las transformaciones escogidas. Al mismo tiempo, como se ha mencionado, las clases se encuentran desbalanceadas, por lo que tomar una decisión de la cantidad de imágenes a utilizar debe estar respaldado, ya que afecta directamente a los resultados del experimento.

Se han planteado dos formas distintas a utilizar este proceso de aumento de datos, con el objetivo de probar dos maneras independientes de sobrellevar este trabajo, y verificar cual de las dos formas, puede entregar mejores resultados al momento de entrenar las dos redes convolucionales elegidas.

5.2.4 Aumento de datos equitativo

La primera forma escogida, para realizar aumento de datos, es realizar la misma cantidad de transformaciones para todas las imágenes, independiente de la clase a la que pertenezcan. Esta forma de aumento de datos es la más común y rápida de utilizar, ya que solo hay que decidir la cantidad de imágenes nuevas que serán generadas por cada una de las imágenes originales. Como todas las imágenes se aumentan de igual forma, se ha patentado a esta estrategia como, aumento de datos **equitativo**. Un problema de realizar este tipo de aumento de datos, es que para este caso, el conjunto de datos seguirá estando desbalanceado después de realizar aplicar esta técnica.

Para este tipo de aumento de datos, se ha ido probando la cantidad de imágenes aumentadas, cada vez generando una imagen más, y revisando de forma manual, si las imágenes que se están generando entre sí son muy parecidas o no. Se fueron "marcando" las imágenes generadas, para saber cuales corresponden a la misma imagen original, y se observó que tan distintas o parecidas son entre si, ya sea en la orientación de las cabezas, la posición de estas, y el brillo de las imágenes.

Con este procedimiento, se llegó a decidir que a partir de diez imágenes generadas, estas comienzan a parecerse mucho, lo que podría generar *overfitting* durante el entrenamiento. Es por esto, que se van a aumentar las imágenes nueve veces, lo que, al incluir las imágenes originales, luego del aumento de datos, se tendrán diez veces la cantidad de imágenes para cada una de las cinco clases de cabezas de espermatozoides.

A continuación se presentan dos formas distintas para balancear las clases de las cabezas de espermatozoides. Estas se harán de manera independiente, por lo que se evaluará, cual de estas tendrá menos error al momento de la clasificación.

Balanceo de datos, utilizando bolsas de datos de igual tamaño

La primera estrategia planteada para resolver el problema de las clases desbalanceados, recomendada por la documentación de la librería de *Tensorflow*, consiste en utilizar la técnica de *undersampling*, la cual corresponde a disminuir la cantidad de datos de las clases

con mayor representatividad, para así tener la misma cantidad de datos aproximados para cada clase (Abadi et al., 2015).

Para aplicar esto, se construirán cinco bolsas de datos de tamaño m , donde m corresponde a la cantidad de imágenes que tiene la clase con menor representatividad, la que corresponde a la clase pequeña según la Tabla 5.1. Cada uno de las bolsas de datos, serán llenadas con imágenes de una sola clase, obtenidas de manera aleatoria, a partir de las imágenes aumentadas. Al final de esto, se unirán las bolsas nuevamente, se reordenaran de forma aleatoria, y se tendrá un conjunto de tamaño $5m$, balanceado, donde cada clase tendrá m cantidad de imágenes para entrenar la red convolucional. Cabe destacar que la clase con menor representatividad, no tendrá aleatoriedad al momento de hacer su bolsa de datos, ya que esta es del mismo tamaño que la cantidad de imágenes de esa clase.

Una desventaja de aplicar esta técnica de *undersampling*, es que se está reduciendo la cantidad de imágenes para entrenar los modelos. En este caso, se realizó aumento de datos equitativo con el objetivo de aumentar la cantidad de los datos y construir los modelos clasificadores, pero con esta forma de balanceo, hay una parte de los datos aumentados que no se estarán utilizando para entrenar los modelos al construir las bolsas de datos.

Balanceo de datos, utilizando pesos por clases durante el entrenamiento

La segunda estrategia elegida para balancear las clases de cabezas de espermatozoides, también fue obtenida de la documentación de la librería *Tensorflow*, y se basa en asignar "pesos" a cada una de las clases, antes del entrenamiento.

Los "pesos" a los que se refiere la documentación, es una forma de indicarle al modelo durante el entrenamiento, a qué clases deberá ponerle más atención. Esto consiste en agregar un multiplicador a las clases, al momento de obtener el error al finalizar una época, y recalcular los pesos de las neuronas de la red. Las clases con un gran número de datos recibirán un peso menor, mientras que para el caso contrario, las clases con menor representatividad, recibirán pesos más grandes (Abadi et al., 2015). Los pesos de cada una de las clases, se obtienen según la proporción de la cantidad de datos para cada una de las clases, con respecto al total de los datos. Esto se puede obtener de manera automática con la librería de *scikit-learn* para análisis de datos en *python*, en conjunto con todos los datos con sus etiquetas correspondientes (Pedregosa et al., 2011).

Una de las ventajas de realizar esto, es que, a diferencia del caso anterior, se estarán utilizando todas las imágenes aumentadas para el entrenamiento de la red. Sin embargo, al mismo tiempo es posible que los pesos no sean suficientes para balancear las clases, por lo que depende del problema (Abadi et al., 2015).

5.2.5 Aumento de datos dirigido

La siguiente forma en la que se ha decidido realizar aumento de datos, es realizar este procedimiento dependiendo de la cantidad de imágenes que tenga la clase originalmente, es decir, que para las clases con menos representatividad serán aumentadas en mayor cantidad, con respecto a aquellas imágenes que correspondan a las clases con mayor cantidad de imágenes. Debido a la característica anterior, a esta forma de aumento de datos, se le ha llamado aumento de datos **dirigido**, y desde ahora en adelante se le referirá de esta manera.

En la reciente publicación de Iqbal et al. (2020) sobre clasificación de cabezas de espermatozoídes humanos, se utilizó este aumento de datos dirigido para balancear las clases del *dataset*, logrando mejorar los resultados del trabajo anterior de Riordon et al. (2019). La cantidad de imágenes fueron aumentadas una gran cantidad de veces, y utilizando las mismas transformaciones definidas anteriormente, por lo que esto ha dado luz verde a realizar este misma estrategia de aumento de datos, para la investigación actual.

Las cinco clases de espermatozoídes fueron aumentadas de tal forma, que para cada clase, se tenían 6.000 imágenes aproximadamente para realizar el entrenamiento de los modelos, por lo que las imágenes de la clase piriforme y pequeñas, fueron aumentadas en una gran cantidad, en comparación a las imágenes de la clase amorfa. A partir de esto, el entrenamiento de los modelos se hizo aproximadamente, con 30.000 imágenes en total (Iqbal et al., 2020).

Debido a la publicación de este trabajo y a sus resultados, se decidió implementar este aumento de datos dirigido, en donde se aumentarán las imágenes para tener aproximadamente la misma cantidad de imágenes para entrenar ambas redes convolucionales escogidas en este experimento. Es importante mencionar que, el resultado de este aumento de datos resuelve también el problema del desbalanceo de clases, ya que se estará entrenando la red con cantidades similares de imágenes de cada una de las cinco clases.

Finalmente, se tienen tres estrategias de aumento de datos distintas, las cuales son, AD equitativo con balanceo con bolsas de igual tamaño, AD equitativo con balanceo a través de pesos por clase, y AD dirigido. Estas tres estrategias serán evaluadas en conjunto con las dos redes convolucionales escogidas, para ver cual es la que crea los mejores modelos, considerando el error durante el entrenamiento de estos.

5.3 ADAPTACIÓN DE LOS MODELOS CONVOLUCIONALES ESCOGIDOS

5.3.1 Adaptación de la red convolucional usada para clasificación de glóbulos rojos

La primera red convolucional a utilizar es la propuesta por Angluin et al. (2016) para su trabajo sobre clasificación de células de glóbulos rojos. Es por esto, que a esta red se le referirá como la red convolucional **RBC**. La construcción y entrenamiento de esta red fue hecha en *Python*, usando las librerías de *keras* y *Theano*, y esta fue presentada anteriormente en el Capítulo 4, pero en la Figura 5.10 se puede observar una representación más clara de esta arquitectura de red convolucional.

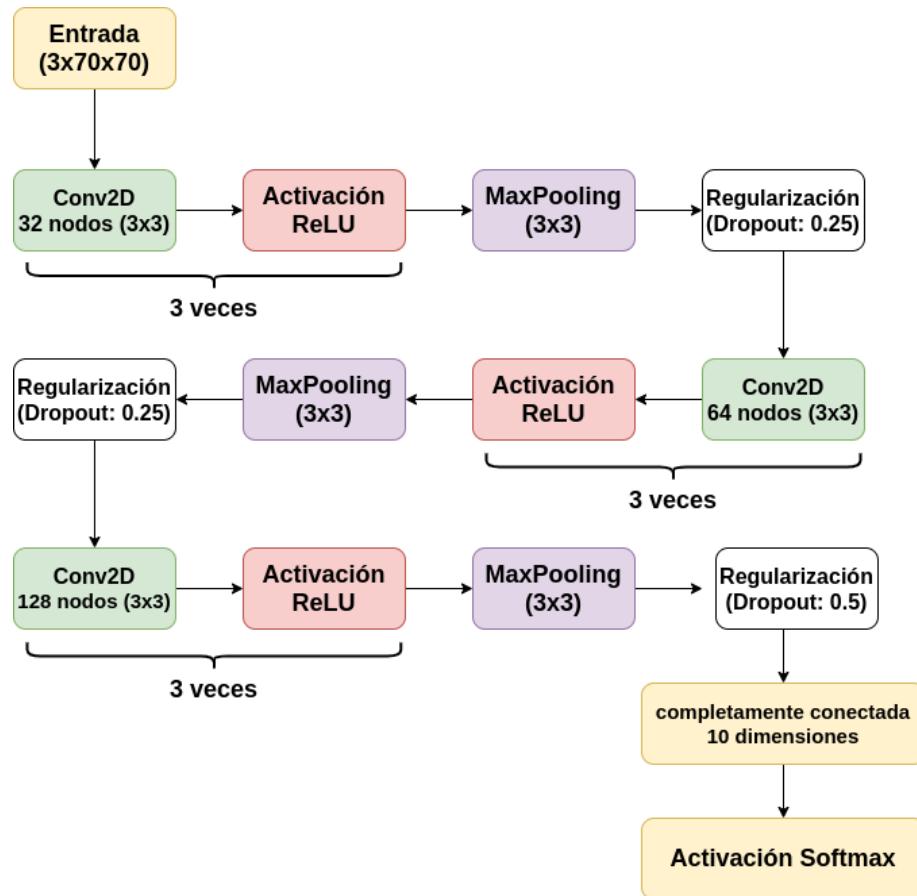


Figura 5.10: Arquitectura de red convolucional original de Angluin et al. (2016). El orden de las capas se encuentra de forma secuencial indicado por las flechas de color negro. Las capas de convolución y activación se repiten tres veces antes de seguir a la capa de *Max Pooling*.

Fuente: Elaboración propia, 2020.

Lo primero que se observa es que la entrada de la red corresponde a las imágenes de glóbulos rojos, y esta se encuentra en arreglos de $3 \times 70 \times 70$, ya que el tamaño de estas

imágenes es de 70×70 píxeles, y se encuentran en formato *RGB*. El ordenamiento de los arreglos de imágenes es importante, ya que la forma en que *Tensorflow* y *Theano* trabajan sobre estas es distinta. Este último utiliza como primera dimensión la cantidad de canales de la entrada (formato *RGB*), mientras que *Tensorflow* utiliza la última dimensión del arreglo de imagen para determinar la cantidad de canales. Las imágenes de glóbulos rojos siguen el ordenamiento de la librería *Theano*, pero en este caso, las fotografías de espermatozoides siguen el formato que trabaja *Tensorflow*, por lo que se ha adaptado el ordenamiento en que trabajan las capas convolucionales.

Se ha decidido utilizar la librería de *Tensorflow* para trabajar debido a que ya se ha trabajado usando este paquete en cursos anteriores, además de que, como ya se mencionó, el ordenamiento de las imágenes de espermatozoides es compatible con esta librería, y no así con *Theano*. Al mismo tiempo, el trabajo de Wu et al. (2019) realizó una comparativa en distintos frameworks de *deep learning*, incluyendo los paquetes de *Tensorflow* y *Theano*, realizando experimentos con las mismas máquinas y con los mismos conjuntos de datos. A partir de estos, se puede observar que el tiempo de entrenamiento puede variar entre ambas librerías, pero al fin y al cabo, en la evaluación de los modelos creados, se muestra que los resultados entre estas dos librerías son bastante cercanos, como aparece en los resultados del entrenamiento con una *GPU* sobre el conjunto de datos público, *MNIST*. Teniendo esto en cuenta, se puede decir que el cambio de librerías entre *Theano* y *Tensorflow*, para adaptar la red convolucional elegida, no afectará de forma negativa los resultados de este experimento de clasificación de cabezas de espermatozoides humanos.

La documentación de las capas convolucionales de *Keras*, detallan un parámetro para el ordenamiento de los arreglos de la entradas de la red, el cual se llama "*data_format*", el cual, para adaptarlo a este problema, fue cambiado desde "*channels_first*" a "*channels_last*", para así trabajar con las imágenes de cabezas de espermatozoides y la librería de *Tensorflow* (Chollet et al., 2015).

El trabajo de Angluin et al. (2016) realizó una clasificación multi-clase, en donde estas era excluyentes, osea que una foto de glóbulo rojo no puede pertenecer a más de una clase. Las dos últimas capas de la arquitectura propuesta corresponde a una capa completamente conectada de diez dimensiones, ya que el experimento trabajó con diez clases de glóbulos rojos, y una capa de activación *softmax*, para generalizar la salida de la red en probabilidades por cada una de las etiquetas.

En el trabajo actual, las clases de cabezas de espermatozoides son cinco, y también son excluyentes. Teniendo esto en cuenta, solo hace falta cambiar la dimensión de la capa completamente conectada a cinco, y mantener la misma función de activación, para así tener como resultado, la probabilidad de pertenencia de una foto de cabeza de espermatozoides a cada una de las cinco clases presentadas, donde finalmente se le asocia una única clasificación

correspondiente a la clase con mayor probabilidad de pertenencia.

La red propuesta utiliza imágenes de 70×70 píxeles, mientras que las muestras de espermatozoides son la mitad del tamaño. Como se ha mencionado, las capas convolucionales y de agrupamiento, disminuyen la dimensión de la imagen al trabajar sobre ellas. Este es un problema debido a que, al ingresar imágenes de la mitad del tamaño, estas no alcanzan a llegar al final de la red.

Una alternativa para solucionar esto, es eliminar alguna de las capas involucradas, pero esto lleva a modificar la forma de la red convolucional, y esto es algo que requiere una serie de experimentos adicionales, para evaluar distintas modificaciones a la red convolucional. Una segunda alternativa es adaptar las imágenes de espermatozoides a un tamaño de 70×70 píxeles, pero al expandir la imagen, se estaría vulnerando la restricción propuesta en el apartado de aumento de datos, ya que se podría estar afectando la forma de las cabezas de espermatozoides, y en consecuencia, las características morfológicas de cada una de las clases. Siguiendo esta misma lógica, se puede crear un marco de un solo color alrededor de las imágenes, para así llegar a tener el tamaño adecuado. La consecuencia de esto, es que muchas neuronas que vean el fondo nuevo de la fotografía, no podrán generalizar información de la forma de las cabezas de los gametos, y este problema se incrementa al considerar que ahora más de la mitad de los píxeles de la fotografía no tendrían información de la forma de estas células.

La alternativa por la que se optó, es la de disminuir el tamaño de los filtros de las capas convolucionales y de agrupamiento, para así, al introducir los arreglos de 35×35 estos alcancen a llegar a la última capa de la arquitectura propuesta. Las capas convolucionales y de agrupamiento ahora usan filtros de 2×2 en vez de 3×3 , y eso hace que una parte de la imagen sea analizada por una menor cantidad de neuronas, pero al fin y al cabo, se estaría evitando agregar neuronas que analicen partes de la fotografía que están fuera del área de interés. Finalmente, los cambios planteados se pueden observar en la Figura 5.11, en donde se puede observar que la arquitectura de la red sigue siendo la misma, solo que algunas configuraciones cambiaron para ingresar las imágenes de las cabezas de espermatozoides.

Como último detalle, el trabajo de Angluin et al. (2016) utilizó versiones antiguas de la librería de *Keras*, por lo que al intentar utilizarla con la versión actual, muchas de las capas de la red lanzaron error en la ejecución del programa. Algunos parámetros de las funciones de las capas convolucionales fueron cambiadas a tuplas, mientras que otros cambiaron de nombre, así que se tuvo que realizar este cambio para poder construir la *CNN*, siguiendo la documentación oficial de la librería (Chollet et al., 2015).

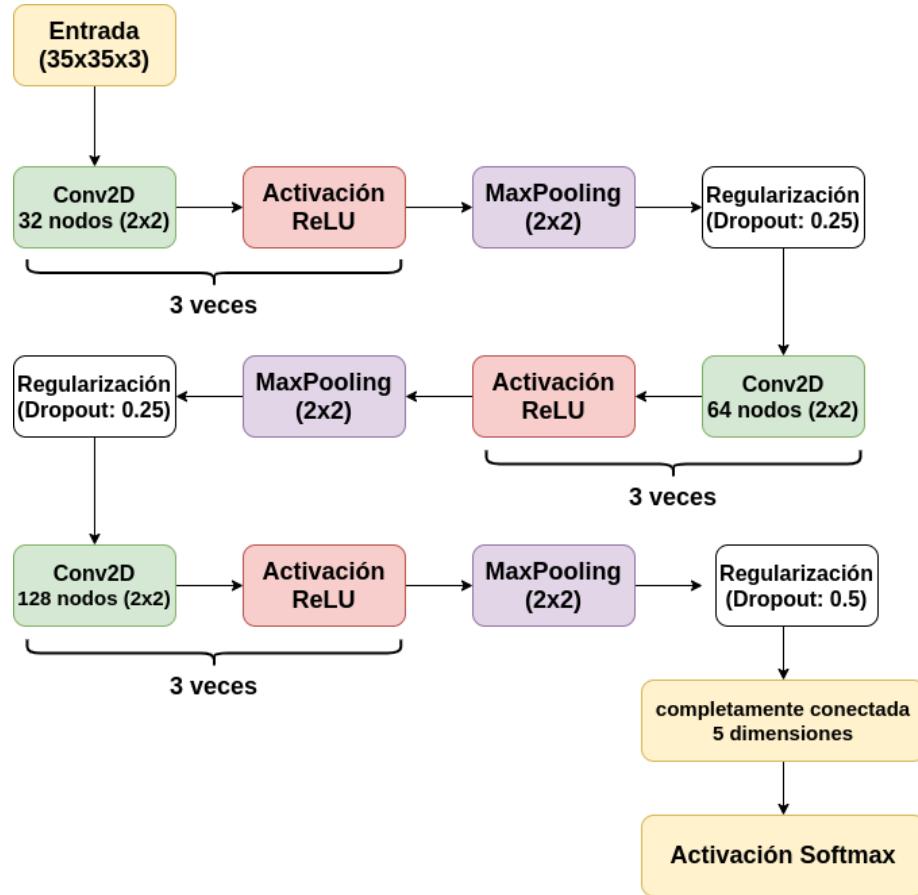


Figura 5.11: Arquitectura de la red convolucional Angluin et al. (2016) adaptada. Se han marcado de color verde aquellas capas que fueron modificadas, además de que la capa completamente conectada también fue modificada para clasificar las cinco clases de cabezas de espermatozoides.

Fuente: Elaboración propia, 2020.

La función de pérdida va acompañada de una función de activación, ya que para calcular el error, se necesita de la salida generalizada que proporciona la etapa de activación. Ya se mencionó que la función *softmax* es reutilizada porque el problema sigue siendo uno de clasificación multi-clase, y al momento de recalcular los pesos de la red en una iteración siguiente, se necesita la función encargada de calcular el error de la red al momento de terminar con una época. Considerando que el problema sigue siendo el mismo que en el trabajo de Angluin et al. (2016), no es necesario cambiar la función de perdida escogida, la cual es **entropía cruzada**. En la librería de Keras, esta función de pérdida se conoce como "*Categorical Cross-Entropy loss*" y se conoce como la función de perdida por defecto en problemas de clasificación (Brownlee, 2019a).

5.3.2 Adaptación de la red convolucional de *ResNet50*

Como ya se ha mencionado, la red *ResNet50* posee 50 capas convolucionales, y un total de 177 capas de profundidad en su arquitectura. En la Figura 5.12 se observa en resumen de la secuencia de capas presentes en la construcción de esta red. A partir de la entrada, existen algunos ciclos de capas convolucionales que se repiten a lo largo de la arquitectura, los cuales fueron agrupados para así intentar acaparar la longitud de esta arquitectura, para terminar en un extremo con la capa completamente conectada que contiene un total de 1.000 neuronas (Mahmood et al., 2020). Es necesario mencionar que en esta imagen no se aprecia la característica de entregar la salida de la convolución a capas posteriores, y tampoco se muestran las capas de agrupación, activación, y normalización que la componen.

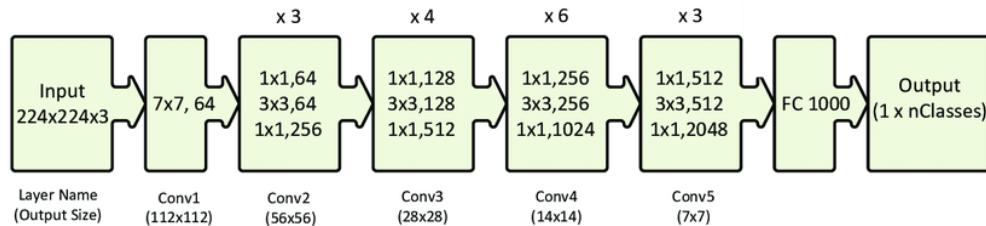


Figura 5.12: Resumen arquitectura de *ResNet50* (Mahmood et al., 2020). Los bloques azules sugieren etapas con distintas convoluciones, que se repiten la cantidad que aparece arriba de de estos. los filtros de las convoluciones tienen tamaños denotados por la forma $m \times m$. *FC* se refiere a la última capa que es la completamente conectada.

En esta ocasión no se está trabajando en la clasificación con el conjunto de imágenes del desafío *ImageNet*, el cuál esta etiquetado en 1.000 clases distintas, por lo que es necesario adaptar esta arquitectura según el problema actual.

Según el trabajo de Riordon et al. (2019), cuando utilizaron la red convolucional *VGG16* para clasificar las cabezas de espermatozoides humanos, lo que hicieron fue tomar un extremo de la red, correspondiente a la capa completamente conectada, agregar unas capas de regularización, y finalmente volver a agregar una capa completamente conectada, pero esta vez con la cantidad de neuronas necesarias para la clasificación de espermatozoides.

De la misma forma, se ha decidido reemplazar la última etapa de la red convolucional *ResNet50*, eliminando la capa completamente conectada de 1.000 neuronas, reemplazándola con una del mismo tipo, conectada solo con cinco neuronas, para así clasificar las cinco clases de cabezas de espermatozoides. Cabe destacar que, como ya se mencionó para el caso de la red neuronal de glóbulos rojos, esta última capa debe estar seguida de una capa de activación correspondiente al problema, la cual se ha escogido la función *Softmax*, para lograr así una clasificación en donde cada una de las cinco formas son independientes entre sí.

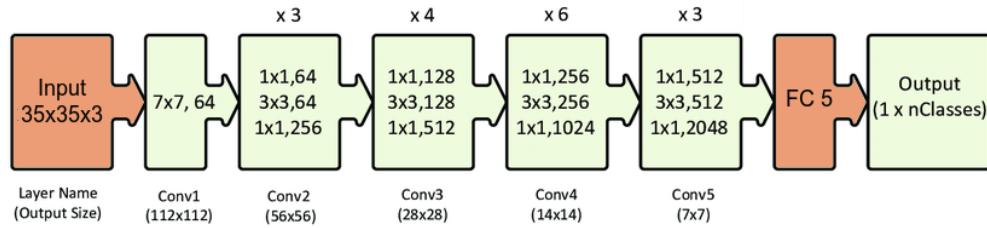


Figura 5.13: Arquitectura de *ResNet50* adaptada al experimento actual. El bloque en color naranja es el cual ha sido modificado, correspondiente a la capa completamente conectada, que ahora tiene una dimensión de 5 neuronas, para realizar la clasificación de las cinco clases de cabezas de espermatozoides humanos.

Fuente: Elaboración propia, 2020.

La librería de *Keras*, dentro de su implementación de la red convolucional *ResNet50*, tiene la opción de reemplazar estas capas finales, para así expandirla o adaptarla a otros problemas distintos al desafío de *ImageNet*. Al mismo tiempo, también se da la opción de utilizar la red con los pesos de sus neuronas pre-entrenados, con distintas bases de datos públicas, como la misma *ImageNet*. En el trabajo de Riordon et al. (2019), utilizaron la red *VGG16* con los pesos de sus neuronas pre-entrenadas con los resultados del mismo desafío, pero en este caso, se está trabajando con dos redes convolucionales distintas, las cuales para ser comparadas, deben ser trabajadas en igualdad de condición. Es por esto, que en este trabajo, se utilizarán ambas redes, pero no se utilizarán pesos pre-entrenados con las imágenes del desafío de *ImageNet*.

5.4 AFINAMIENTO DE PARÁMETROS PARA EL ENTRENAMIENTO

Con las arquitecturas convolucionales adaptadas para el problema de clasificación de cabezas de espermatozoides, hay que tener en cuenta que al momento de entrenarlas, existen algunos parámetros que aún no se encuentran definidos, y estos son la cantidad de épocas (o iteraciones) a entrenar, y el optimizador de esta, que se encarga de actualizar los pesos de la red dependiendo de los valores de error obtenidos al finalizar una época. Los optimizadores se encuentran implementados en la librería de *Keras*, entre los que se encuentran: *SGD*; *RMSprop*; *Adam*; *Adadelta* entre otros. Estos trabajan matemáticamente de forma distinta, por lo que para escoger un optimizador en este problema de clasificación de cabezas de espermatozoides, se ha decidido implementar aquellos que ya hayan sido utilizados en los problemas planteados en la revisión de la literatura presentada en este trabajo (Chollet et al., 2015).

5.4.1 Elección del optimizador

En el trabajo de Angluin et al. (2016) para clasificación de glóbulos rojos, se utilizó el optimizador *SGD* y el *Adam*, mientras que en el trabajo de clasificación de partes de espermatozoides normales y anormales de Javadi & Mirroshandel (2019) también utilizó *Adam* como optimizador al momento de entrenar su red convolucional. Este último trabajo no especifica que configuración de *Adam* utilizó en su trabajo (tasa de aprendizaje y sus constantes β_1 y β_2), mientras que en el experimento de Angluin et al. (2016), detalla de forma explícita que utilizó el optimizador *SGD* con *Nesterov*, y el optimizador *Adam* con sus parámetros por defecto, propuestos por Kingma & Ba (2014).

En consecuencia, se han planteado los siguientes optimizadores a entrenar las redes escogidas, los cuales serán evaluados, para ver cuál de estos obtiene mejores resultados.

- Optimizador **A**, *SGD* con parámetros utilizados por Angluin et al. (2016): El experimento de clasificación de glóbulos rojos utilizó esta configuración en primera instancia, por lo que se estudiará el entrenamiento con esta configuración, el cual utiliza una tasa de aprendizaje de 0,0001, un valor de *momentum* de 0,9, y aplicando la técnica de *Nesterov*.
- Optimizador **B**: Se utilizará el optimizador **A**, pero disminuyendo la tasa de aprendizaje a 0,000001, para verificar si no se está realizando *overshoot* al recalcular los pesos al finalizar una época.
- Optimizador **C**, *Adam* con parámetros por defecto: El trabajo de Javadi & Mirroshandel (2019) no menciona que configuración del optimizador de *Adam* utilizó al entrenar su red, mientras que, en el caso del trabajo de Angluin et al. (2016), si menciona que utilizó el optimizador de *Adam* con su configuración por defecto detallada por Kingma & Ba (2014). En consecuencia, también se utilizará la configuración por defecto del optimizador de *Adam* para este proceso de afinamiento de la red, el cual comprende de una tasa de aprendizaje del 0,001, mientras que los parámetros de β_1 , β_2 y ϵ , utilizan valores del 0,9, 0,999, y $1e - 08$ respectivamente (Chollet et al., 2015).
- Optimizador **D**: Se utilizará el optimizador **C**, pero disminuyendo la tasa de aprendizaje a 0,00001, de la misma forma que en el optimizador anterior, para compararlos en igualdad de condiciones.

5.4.2 Elección del número de épocas a entrenar

Por otro lado, la cantidad de épocas a entrenar una red convolucional también es un factor que cambia dependiendo del problema. Si se escoge un número pequeño, es posible llegar a tener *underfitting* en el modelo, por lo que la red no habrá generalizado suficientes características para reconocer las clases de cabezas de espermatozoides, o puede ocurrir el caso contrario, en donde el número de épocas es tan alto que, durante el entrenamiento, la red convolucional ha memorizado las imágenes de entrenamiento, generando *overfitting*, y no podrá diferenciar las clases de forma correcta cuando se le muestren imágenes nuevas de cabezas de espermatozoides.

Con el fin de decidir la cantidad de épocas a utilizar en el entrenamiento de la red, se evaluará este parámetro en los cuatro optimizadores seleccionados, y para las dos redes convolucionales a utilizar en este experimento. Para realizar esto, se entrenaron ambas redes convolucionales elegidas, durante 1.000 épocas, para así tener una amplia visión de como se comporta el error de los modelos.

Es importante mencionar que, como se estableció que se irán a evaluar tres estrategias de aumento de datos, es importante verificar la cantidad de épocas también para cada una de estas. Esto es porque, cada una de estas formas de realizar aumento de datos, utilizará una cantidad de imágenes distintas durante el entrenamiento, por lo que no se puede asegurar que la cantidad de épocas ideal sea la misma para las tres formas de *AD* definidas.

Para visualizar el error de los modelos durante el entrenamiento, se utilizó un 20% de los datos destinados para el entrenamiento, para así validar el error del modelo al finalizar cada época. Al finalizar todas las épocas de entrenamiento, se construirá el gráfico del comportamiento del error durante todo el entrenamiento, y así se podrá ver en qué momento se alcanzó el menor valor para la función de costo, que es el error del modelo al clasificar las cabezas de espermatozoides.

El principal criterio para decidir la cantidad de épocas a entrenar el modelo, consiste en observar el gráfico de error, y escoger el momento en donde el error sea más cercano a cero. Tampoco se puede escoger una época muy temprana ya que, podría generarse *underfitting* al no haber aprendido las suficientes características de las formas de cabezas de espermatozoides.

En este proceso, se usó 1.000 épocas para evaluar el comportamiento del error, a excepción de un caso específico, que corresponde a la red convolucional *Resnet50*, utilizando aumento de datos dirigido. El problema que se encontró con este caso, es que la red convolucional *Resnet50*, que ya tiene un entrenamiento lento debido a la cantidad de capas que la componen, se le sumó el hecho de que al hacer aumento de datos dirigido, se tienen 30.000 imágenes para entrenar los modelos, lo que provocó que cada época de entrenamiento tomaba más de 60

segundos, en comparación con los otros casos, que máximo tomaban cinco segundos por época. Debido a esto, solo para el caso de la red convolucional *ResNet50* con aumento de datos dirigido, se evaluó el error durante 100 épocas en vez de 1.000.

Se escogió utilizar en la mayoría de los casos, 1.000 épocas de entrenamiento, debido a que es más rápido probar esta cantidad de épocas una sola vez, en vez de ir probando pocas iteraciones e ir aumentando en cierto intervalo, como por ejemplo, comenzar entrenando con 200 épocas, luego ver qué sucede con 250 épocas, y así sucesivamente. Tampoco se entrenó por más de 1.000 épocas porque el tiempo de entrenamiento comenzaba a incrementarse más, considerando la cantidad de combinaciones a evaluar.

5.4.3 Elección de los mejores parámetros

Según lo anterior, se debe evaluar el número de las épocas de entrenamiento adecuadas para los cuatro optimizadores mencionados, para ambas redes convolucionales, y al mismo tiempo, para cada estrategia de aumento de datos. Es decir, que para este apartado de elección de parámetros, se evaluarán 12 casos posibles para cada red convolucional, al ir probando las combinaciones entre optimizadores y tipo de aumento de datos a realizar.

Se analizarán los gráficos de la función de error versus la cantidad de épocas, para los 24 casos posibles, y entre todos estos modelos generados, escoger uno o dos que hayan alcanzado el error más cercano a cero y la exactitud más alta con el conjunto de validación. Estos dos casos corresponderán a la mejor combinación entre optimizador, tipo de aumento de datos, y cantidad de épocas para las redes convolucionales *RBC* y *ResNet50*, por lo que serán estas configuraciones las utilizadas para crear los modelos clasificadores. Se compararán los resultados obtenidos al evaluar ambas redes convolucionales escogidas entre sí, y también contra el estado del arte mencionado en el Capítulo 3.

Los resultados se miden con el conjunto de validación, ya que este corresponde a las imágenes que no se han utilizado durante el entrenamiento, por lo que refleja mejor el comportamiento de los modelos con imágenes que no conoce, e incluso así se puede identificar cuando ocurren problemas como *overfitting* o *underfitting*.

5.5 IMPLEMENTACIÓN DE APRENDIZAJE POR TRANSFERENCIA

En el experimento de Riordon et al. (2019), donde utilizaron la red convolucional *VGG16* para la clasificación morfológica de cabezas de espermatozoides, los autores decidieron usar la red convolucional pre-entrenada con las imágenes del desafío *Imagenet*. Sin embargo, cuando se adaptaron las redes convolucionales *RBC* y *ResNet50*, se decidió que no se ocuparían pesos pre-entrenados con las imágenes de este desafío.

La principal razón por la cual se ha decidido no utilizar pesos pre-entrenados con *Imagenet*, es porque este conjunto de imágenes tiene alrededor de 1000 clases distintas de objetos a reconocer, como objetos comunes de la vida diaria, o incluso distintas razas de perros, pero no se ve ninguna semejanza o relación que pueda aportar estas clases, a la clasificación de las cabezas de espermatozoides humanos (Russakovsky et al., 2015).

Sin embargo, se ha planteado en los objetivos específicos de este experimento, la evaluación de aprendizaje por transferencia, por lo que, se usará un *dataset* distinto al de *Imagenet* para pre-entrenar las redes convolucionales *RBC* y *ResNet50*.

5.5.1 Pre-entrenamiento de las redes convolucionales escogidas

El experimento realizado por Angluin et al. (2016) realizó una clasificación morfológica de células de glóbulos rojos, con un *dataset* aumentado a más de 3000 imágenes. Como ya se mencionó en el Capítulo 4, se escogió adaptar esta red debido a la similitud del problema (clasificación morfológica) y de las imágenes a clasificar (células de glóbulos rojos y cabezas de espermatozoides respectivamente). A partir de este mismo razonamiento, es que se ha decidido, además, utilizar este *dataset* aumentado, para pre-entrenar las redes adaptadas de *RBC* y *ResNet50*, para luego entrenarlas con las imágenes de cabezas de espermatozoides, y así aplicar aprendizaje por transferencia.

Para la red convolucional de *RBC*, se conoce que los resultados de su trabajo, se obtuvieron utilizando el optimizador *SGD*, pero en este trabajo, el tamaño de los filtros de esta red convolucional fue disminuido para soportar las imágenes de 35×35 píxeles de las cabezas de espermatozoides humanos, por lo que se deberá probar nuevamente cual es el mejor optimizador para este pre-entrenamiento. Se utiliza la red adaptada y no la original, debido a que el tamaño de los filtros afecta directamente a la cantidad de neuronas por capa, y para realizar este aprendizaje por transferencia, la cantidad de los pesos debe ser la misma para reutilizar las redes.

Para decidir qué optimizador y cantidad de épocas a utilizar en el pre-entrenamiento,

se repetirá el mismo procedimiento realizado detallado en el afinamiento de parámetros, por lo que ambas redes escogidas serán entrenadas y validadas con un conjunto de imágenes separado, para así obtener los gráficos del error a lo largo de las épocas de entrenamiento.

Para realizar el aprendizaje por transferencia, las redes se van a reutilizar de un problema a otro, por lo que, las redes convolucionales utilizadas para entrenarse con las imágenes de glóbulos rojos, deben tener la misma estructura de capas, y tamaños de filtros, ya que son estos los que definen el número de neuronas y pesos de la red convolucional. A pesar de esto, la última capa de ambas redes, que corresponde a la completamente conectada, no puede ser la misma, por que está depende de la cantidad de clases del *dataset*.

El conjunto de imágenes de glóbulos rojos tiene **diez** clases distintas, por lo que las capas completamente conectadas deberán de tener un tamaño de diez neuronas, para pre-entrenar las redes con este *dataset*. En la Figura 5.11 y 5.13, se observan las redes adaptadas de *RBC* y *ResNet50* respectivamente, en donde, la penúltima capa en ambas figuras, corresponde a la completamente conectada, la cual será reemplazadas en ambos casos por una de tamaños de diez neuronas para realizar el pre-entrenamiento.

Entonces, una vez terminado el pre-entrenamiento, se utilizarán estos pesos correspondientes, para comenzar el entrenamiento de ambas redes convolucionales, en el contexto de la clasificación morfológica de cabezas de espermatozoides humanos. Sin embargo, como en el párrafo anterior se mencionó que la capa completamente conectada tiene que ser reemplazada, la cantidad de pesos para esta capa no es la misma para el problema de clasificación de los espermatozoides humanos.

La forma de resolver este problema, es descartar los pesos pre-entrenados para la capa completamente conectada, es decir que, al momento de entrenar los modelos para la clasificación de las cabezas de espermatozoides humanos, la única capa que tendrá pesos aleatorios en sus neuronas, será la capa completamente conectada, debido a que esta no es compatible con el *dataset* anterior de glóbulos rojos.

En las versiones más recientes de la librería *Tensorflow*, los métodos para cargar los pesos de una red a otra, tiene un parámetro definido para saltarse los pesos de las capas que no coinciden en el tamaño de neuronas ("*skip_mismatch*"). Sin embargo, en este trabajo se está utilizando la versión 2.0 de *Tensorflow*, debido a la compatibilidad de la librería con la *GPU* del computador de trabajo, y esta aún no tiene implementada esta lógica al transferir los pesos de una red convolucional a otra (Abadi et al., 2015).

Por esta razón, la forma de de pasar los pesos desde un problema a otro, será de manera manual, utilizando arreglos de la librería *Numpy* para *Python*, donde los pasos son los siguientes (aplica para ambas redes convolucionales).

1. Convertir los pesos pre-entrenados a un arreglo en *Numpy*

2. Eliminar la última fila de los pesos, que corresponde a la capa completamente conectada con 10 neuronas
3. Armar la red convolucional, sin contar la capa completamente conectada
4. Cargar los pesos pre-entrenados
5. agregar la capa completamente conectada con 5 neuronas y pesos aleatorios

Teniendo las redes pre-entrenadas, ya pueden partir a ser entrenadas para el problema de clasificación de cabezas de espermatozoides humanos. Esto da la posibilidad a dos experimentos posibles, los que consisten en entrenar los modelos **desde cero** (o *scratch*), con pesos iniciales aleatorios en las neuronas, o entrenar los modelos con pesos pre-entrenados con las imágenes de glóbulos rojos trabajados en Angluin et al. (2016), aplicando **aprendizaje por transferencia**.

Cada experimento deberá realizar su propio análisis de parámetros, es decir, si antes se habían establecido 24 escenarios posibles, de combinaciones entre optimizadores, aumento de datos, y cantidad de épocas, esto deberá aplicarse tanto para el experimento con entrenamiento desde cero, y entrenamiento con aprendizaje por transferencia, llevando a 48 la cantidad de configuraciones a evaluar el comportamiento del error para clasificar la forma de cabezas de espermatozoides humanos.

5.5.2 Modificación del mejor modelo convolucional obtenido

Considerando que en este trabajo se entrena los modelos desde cero, y con aprendizaje por transferencia, aún existe la posibilidad de, modificar la arquitectura convolucional de alguna de las capas que componen a las redes escogidas, agregando, eliminando, o incluso congelando algunas de estas. Realizar esto para las redes convolucionales de *RBC* y *ResNet50*, podría aumentar significativamente la cantidad de entrenamientos, considerando que se han establecido a 48 configuraciones las que se están evaluando.

Es por esto, que se escogerá la red convolucional que tenga los mejores resultados en la clasificación de cabezas de espermatozoides, ya sea la red *RBC* o *RBC*, con o sin aprendizaje por transferencia, y será esta la cuál se modificará su arquitectura, para intentar mejorar aún más la precisión en la clasificación de las imágenes.

En esta etapa, se ha decidido congelar algunas capas convolucionales, y volver a entrenar el mejor modelo ya entrenado con las imágenes de cabezas de espermatozoides. Los pesos congelados no se actualizarán al finalizar las épocas durante este nuevo entrenamiento,

manteniendo así el aprendizaje de las características de las formas de las cabezas. En el otro caso, las capas que no se encuentren congeladas, seguirán actualizando los pesos de sus neuronas en las épocas siguientes.

Para este caso final, se utilizará el mismo optimizador con el que se obtuvieron los resultados del modelo elegido, y la cantidad de épocas se determinará de la misma forma, al visualizar de forma gráfica el comportamiento del error a lo largo de las épocas de entrenamiento. Finalmente, se fija la cantidad de épocas en el valor donde se haya alcanzado el menor error posible, y se volverá a entrenar el modelo con la cantidad de épocas seleccionada, el cual se evaluará nuevamente para determinar si con estas modificaciones, se podría mejorar la precisión en la clasificación de las cabezas de espermatozoides humanos.

5.6 RESUMEN

Al utilizar el *dataset* de **SCIAN-MorphoSpermGS**, se trabajará con el conjunto de acuerdo parcial, para así decidir la clase de una imagen por mayoría de votos, a partir de la clasificación hecha por tres expertos distintos. Sin embargo, para utilizar este conjunto de imágenes, se debe realizar aumento de datos, ya que la cantidad de imágenes no es suficiente para crear buenos modelos. A partir de esto, se decidió implementar AD equitativo con balanceo con bolsas de datos de igual tamaño, AD equitativo utilizando pesos por clases para balancear el conjunto de datos, y AD dirigido.

Por otro lado, las redes convolucionales escogidas fueron adaptadas al problema en cuestión, en donde los modelos deben clasificar entre cinco clases distintas e independientes, de cabezas de espermatozoides. Para esto, se han modificado las capas completamente conectadas, tanto para la red de *RBC* y *ResNet50*, para poder realizar esta nueva clasificación, además de que en la primera de estas, se disminuyeron los tamaños de los filtros convolucionales, para que las imágenes de espermatozoides lograran cruzar toda la red convolucional, debido a que este conjunto solo tiene imágenes de 35×35 .

Para utilizar estas redes convolucionales, se deberán evaluar parámetros como los optimizadores para calcular los pesos de las redes, y la cantidad de épocas en el entrenamiento. Esto debe realizarse por cada red escogida, y por cada tipo de aumento de datos implementado, por lo que la cantidad de combinaciones a evaluar asciende a 24, las que serán evaluadas según el error durante el entrenamiento.

También se realizará aprendizaje por transferencia, al pre-entrenar ambas redes escogidas con el *dataset* de imágenes de Angluin et al. (2016), el cual corresponde a imágenes de células de glóbulos rojos, debido a que también se realizó clasificación morfológica, y además, los

glóbulos rojos tienen un cierto parecido a las cabezas de espermatozoides humanos, en cuanto a su forma se refiere. De esto, nace un segundo experimento, en donde además, se evaluaran las redes escogidas, pero utilizando los pesos de sus neuronas pre-entrenados, elevando a 48 la cantidad de casos a evaluar.

Cómo ultimo experimento, se escogerá el mejor modelo entre los obtenidos, ya sea de la red *RBC* o *ResNet50*, o desde cero o pre-entrenados. Luego, este modelo es modificado en su estructura, congelando sus capas, lo que consiste en no actualizar los pesos de las neuronas de estas capas. Finalmente, se vuelve a entrenar este modelo modificado para intentar mejorar la precisión en la clasificación morfológica de los espermatozoides humanos.

CAPÍTULO 6. RESULTADOS EXPERIMENTALES

En este capítulo se presentan los resultados obtenidos del procedimiento experimental expuesto en el Capítulo 5, los que se encuentran divididos en tres experimentos secuenciales. Los dos primeros corresponden a la evaluación de las redes convolucionales desde cero y con aprendizaje por transferencia, respectivamente, y el último consiste en evaluar el mejor modelo obtenido de los dos experimentos anteriores, pero con su arquitectura convolucional modificada, al congelar alguna de sus capas.

Los experimentos fueron realizados utilizando la técnica de validación cruzada, para asegurar la robustez de los resultados obtenidos, por lo que a continuación se presenta la implementación de esta técnica.

6.1 VALIDACIÓN CRUZADA

Se ha decidido utilizar validación cruzada *shuffle split* de **diez iteraciones** durante los experimentos, es decir, que en cada iteración, se separará el *dataset SCIAN-MorphoSpermGS*, en una proporción de 70% para los entrenamientos, y 30% para las evaluaciones. Además, las imágenes podrán repetirse en ambos sub-conjuntos, entre las iteraciones de esta validación cruzada.

Se decidió utilizar este tipo, por sobre *k-fold*, debido a que como el conjunto de acuerdo parcial tiene una cantidad limitada de imágenes, por lo que separar este *dataset* en *k* particiones, podría limitar el tamaño de los conjuntos utilizados para la evaluación de los modelos.

6.1.1 Separación del *dataset*

En cada iteración de validación cruzada, el *dataset* de 1.132 imágenes, será dividido en un 70% para el entrenamiento y un 30% para la evaluación, obteniendo así 792 y 340 imágenes, para el entrenamiento y evaluación respectivamente.

Estos sub-conjuntos generados deben cumplir con la condición de tener presente imágenes de todas las clases de cabezas de espermatozoides, por lo que el desbalanceo de clases vuelve a ser un problema acá. Si el subconjunto de evaluación no tiene imágenes de alguna clase (como la piriforme por ejemplo), no será posible obtener las métricas de evaluación con respecto a esa clase, o sea, que no se podrá saber cuál será el rendimiento de la red

convolucional al clasificar ese tipo de imágenes de espermatozoides en la práctica. En el caso opuesto, si es que en el conjunto de entrenamiento no existieran imágenes de alguna clase, la red convolucional no aprenderá las características de ese tipo de cabezas de espermatozoides, así que al momento de clasificar nuevas imágenes de espermatozoides humanos, no identificará a ninguna como la clase que no ha visto, ya que esta red trabaja con aprendizaje supervisado.

Para resolver a este problema, la formación de estos sub-conjuntos se realizó de manera **estratificada**. Esto quiere decir que, tanto en el conjunto de evaluación como en el de entrenamiento, la cantidad de imágenes presentes de cada clase será proporcional a la cantidad de imágenes por clase en el total del conjunto inicial, y así se puede asegurar que las cinco clases están presentes tanto al momento de entrenar como al momento de evaluar los modelos. Es importante destacar que los trabajos de Angluin et al. (2016) y Javadi & Mirroshandel (2019) también realizaron validación cruzada, y se encontraron con este problema al separar sus conjuntos de datos, el cual también resolvieron al formar los sub-conjuntos de forma estratificada.

Luego de la separación del *dataset*, se hace el proceso de **aumento de datos** solamente en el subconjunto de entrenamiento. En el Capítulo 5, se estableció que se implementarán tres estrategias distintas para aplicar aumento de datos, por lo que el tamaño del conjunto de entrenamiento, luego del aumento de datos, dependerá de esta.

El subconjunto de entrenamiento también será dividido previo al entrenamiento, en un 80% para entrenar los modelos, y en un 20% para realizar la validación de los modelos al finalizar cada época del entrenamiento. Esta validación consiste en probar las configuraciones de optimizador, aumento de datos, y tamaño de épocas, por lo que el gráfico de error utilizado para identificar las mejores configuraciones, se obtendrán a partir de este último 20%.

Luego de obtener las mejores configuraciones de parámetros para entrenar los modelos, se entrenarán las redes con estas configuraciones, obteniendo diez modelos distintos a partir de las diez iteraciones de la validación cruzada. De estos modelos, se obtendrán las métricas de evaluación correspondientes, y el resultado final será la media aritmética de las diez métricas anteriores.

En la Figura 6.1 se puede observar un diagrama, correspondiente al proceso de validación cruzada descrito. Antes del entrenamiento, existe un bloque de balanceo de datos, el cual solo estará presente cuando se utilice una estrategia de aumento de datos equitativa, debido que el aumento de datos **dirigido**, devuelve un conjunto ya balanceado. Cuando se utilice aumento de datos equitativo, con balanceo con bolsas de datos de igual tamaño, el bloque de balanceo de datos se ve reflejado en la Figura 6.2, mientras que para el caso de aumento de datos equitativo, usando pesos por clase para balancear los datos, el bloque representativo se ve en la Figura 6.3.

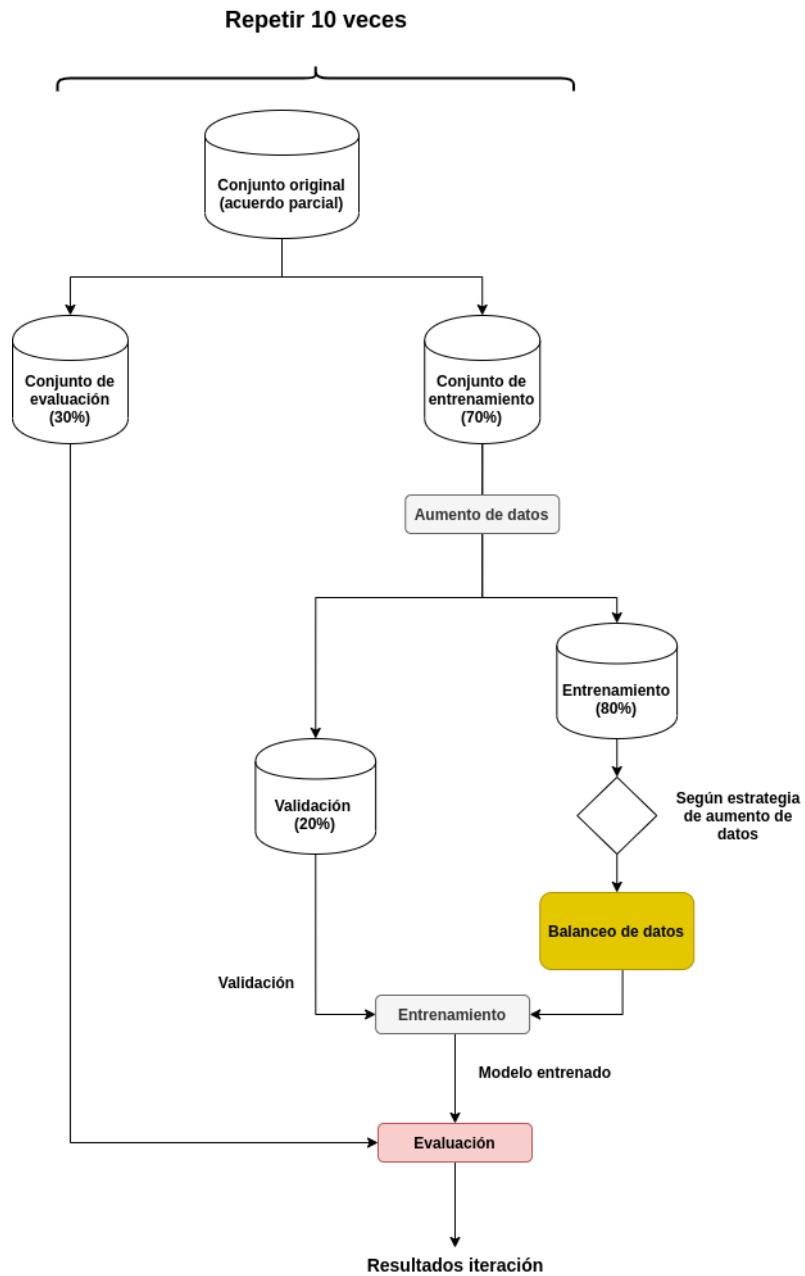


Figura 6.1: Validación cruzada *shuffle split* realizada en este trabajo. El orden está definido por las flechas, y este proceso se repetirá 10 veces. El bloque de balanceo de los datos dependerá de la estrategia de aumento de datos utilizada.

Fuente: Elaboración propia, 2020.

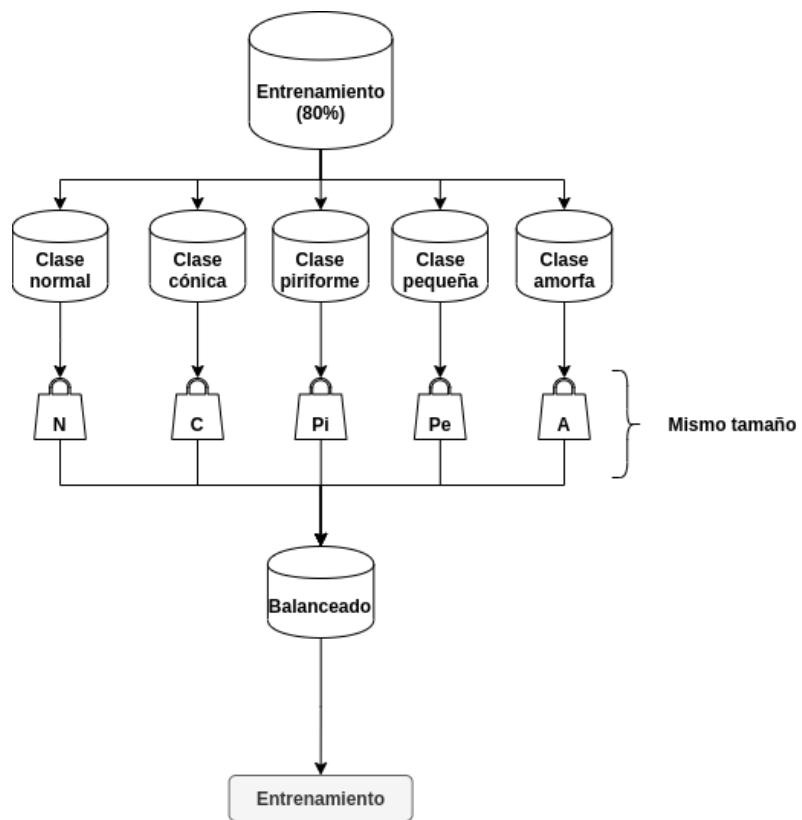


Figura 6.2: Bloque de balanceo de clases para aumento de datos equitativo, con bolsas de datos. Todas las bolsas tienen el mismo tamaño, por lo que al unirlas y reordenarlas, se tendrá un conjunto con el mismo número de imágenes en las cinco clases de cabezas de espermatozoides.

Fuente: Elaboración propia, 2020.

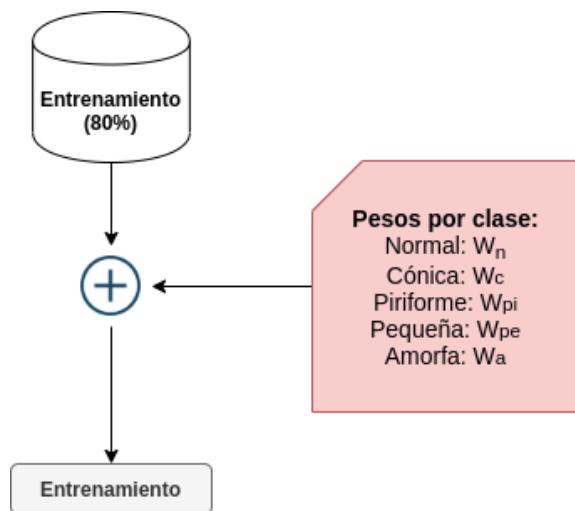


Figura 6.3: Bloque de balanceo de clases para aumento de datos equitativo, usando pesos en las clases. Los pesos se obtienen de manera externa, y se usa la totalidad de los datos aumentados.

Fuente: Elaboración propia, 2020.

6.2 CANTIDAD DE IMÁGENES UTILIZADAS

Considerando el procedimiento establecido para validación cruzada, en esta sección se define la cantidad de imágenes utilizadas por clases en todos los procedimientos. En la Tabla 6.1 se observa en primera instancia, la primera separación de los datos, que consiste en un 70% para el conjunto de entrenamiento, y un 30% para el conjunto que se utilizará en la evaluación de los modelos.

Conjunto/Clases	Normal	Cónica	Piriforme	Pequeña	Amorfa	Total
Original	100	228	76	72	656	1132
Evaluación (30%)	30	68	23	22	197	340
Entrenamiento (70%)	70	160	53	50	459	792

Tabla 6.1: Imágenes por clases de forma de cabezas estudiadas, para el conjunto de evaluación y de entrenamiento

Fuente: Elaboración propia, 2020.

A partir de esto, es donde la cantidad de imágenes comienza a diferir, dependiendo de la estrategia de aumento de datos elegida. Comenzando por el aumento de datos equitativo, se estableció que todas las imágenes del conjunto de entrenamiento serán aumentadas en la misma cantidad, independiente de las clases asociadas a la imagen. Luego de esto, se realiza la separación entre las imágenes que se utilizan para el entrenamiento mismo de los modelos, y las imágenes que se utilizan para la validación de los modelos, osea, las utilizadas para calcular las métricas al final de cada época de entrenamiento. Finalmente, la cantidad de imágenes utilizadas para este tipo de aumento de datos, se ve reflejada en la Tabla 6.2.

Conjunto/Clases	Normal	Cónica	Piriforme	Pequeña	Amorfa	Total
Conjunto de entrenamiento	70	160	53	50	459	792
Conjunto de entrenamiento (con AD equitativo)	700	1600	529	500	4589	7918
Validación (20%)	140	320	106	100	918	1584
Entrenamiento (80%)	560	1280	423	400	3671	6334

Tabla 6.2: Separación del conjunto de entrenamiento, con aumento de datos equitativo

Fuente: Elaboración propia, 2020.

Sin embargo, como ya se explicó en el Capítulo 5, se debe realizar un balanceo de las clases antes de entrenar los modelos. Para el caso de utilizar pesos por clases, el balanceo de las clases se realiza con estos factores al momento de entrenar la red convolucional, por lo que la cantidad de imágenes utilizadas, es exactamente la que se ve en la Tabla 6.2.

Por otro lado, para el caso del balanceo de las clases con bolsas de datos, se mencionó que se utilizarán bolsas de tamaño correspondiente a la clase con menor cantidad de imágenes para entrenar. Si se observa la última fila de la Tabla 6.2, la clase con menor cantidad

de imágenes es la pequeña con 400 datos, por lo que las bolsas armadas tendrán cada una 400 imágenes, elegidas de manera aleatoria, convergiendo finalmente a 2.000 imágenes para el entrenamiento (400 imágenes por cada clase), siendo esta estrategia de aumento de datos, la que menos imágenes tiene al momento de entrenar los modelos.

En el caso del aumento de datos dirigido, las imágenes son aumentadas en distintas cantidades, dependiendo de la clase de la imagen, en donde las imágenes de las clases con menor representatividad, serán aumentadas en mayor cantidad. En la Tabla 6.3 se observa la cantidad de datos utilizados con esta estrategia de aumento de datos, en donde se ve que las clases ya se encuentran balanceadas, listas para entrenar los modelos.

Conjunto/Clases	Normal	Cónica	Piriforme	Pequeña	Amorfa	Total
Conjunto de entrenamiento	70	160	53	50	459	792
Conjunto de entrenamiento (con AD dirigido)	7.786	7.509	7.717	7.544	7.795	38.351
Validación (20%)	1.557	1.502	1.543	1.509	1.559	7.670
Entrenamiento (80%)	6.229	6.007	6.174	6.035	6.236	30.681

Tabla 6.3: Separación del conjunto de entrenamiento luego de aumento de datos dirigido
Fuente: Elaboración propia, 2020.

6.3 PRIMER EXPERIMENTO: ENTRENAMIENTO DESDE CERO

6.3.1 Protocolo de experimentación

El paso a paso de como se llevo a cabo el entrenamiento y evaluación, de los modelos de las redes convolucionales *RBC* y *ResNet50* desde cero, se ve en el Algoritmo 6.1, desde la elección de parámetros, hasta el entrenamiento de los modelos con las mejores configuraciones obtenidas.

Algoritmo 6.1: Implementación del experimento 1: entrenar modelos desde cero

```
1 redes = RBC, ResNet50;
2 implementaciones de AD = equitativo con bolsas, equitativo con pesos, dirigido;
3 optimizador A = SGD con tasa aprendizaje de 0,0001;
4 optimizador B = SGD con tasa aprendizaje de 0,000001;
5 optimizador C = ADAM con tasa aprendizaje de 0,001;
6 optimizador C = ADAM con tasa aprendizaje de 0,00001;
7 optimizadores = optimizador A, optimizador B, optimizador C, optimizador D;
8 para cada red R en redes hacer
9   para cada estrategia AD en implementaciones de AD hacer
10    para Cada optimizador OP en optimizadores hacer
11      Implementando validación cruzada hacer
12        Entrenar R, usando AD y OP seleccionados;
13        Graficar error versus cantidad de épocas;
14        Graficar exactitud versus cantidad de épocas;
15    fin
16  fin
17 fin
18 fin
19 Evaluar las configuración con los gráficos de error y exactitud obtenidos;
20 Escoger la mejor configuración para la red RBC y Resnet50;
21 Seleccionar cantidad de épocas a entrenar para cada una de las configuraciones
seleccionadas;
22 para cada red R en redes hacer
23   para cada configuración escogida hacer
24     Implementando validación cruzada hacer
25       Entrenar la red con la configuración correspondiente y obtener los modelos;
26       Generar métricas de evaluación a partir de los modelos entrenados;
27   fin
28 fin
29 fin
30 obtener el modelo entrenado definitivo para la red RBC y ResNet50;
31 Evaluar el desempeño de la red entrenada con el conjunto de evaluación;
32 calcular métricas de evaluación comparando resultados del experimento contra estado del
arte;
```

6.3.2 Resultados del afinamiento de parámetros

Considerando que por cada red se probaron 12 configuraciones para cada red convolucional, sobre diez iteraciones de validación cruzada, solo se presentarán los gráficos promedios obtenidos de los mejores optimizadores, por cada estrategia de aumento de datos utilizada, para así no tener problemas con la cantidad de gráficos presentes en este capítulo.

El orden en que se muestran los resultados son, desde aquellas configuraciones con peores resultados, hasta las que obtuvieron los mejores, para así facilitar la lectura y el análisis realizado por sobre las mismas configuraciones.

Al mismo tiempo, como se explicó en el capítulo anterior, los resultados más importantes corresponden a los representados por el conjunto de validación (línea naranja en los gráficos), el que corresponde a la clasificación hecha por los modelos, en un conjunto de imágenes que no ha visto durante todo el entrenamiento, por lo que entregan una mejor visión del comportamiento de estos cuando se les presentan nuevas imágenes.

Red convolucional RBC

En la Figura 6.4, se ve un promedio de los resultados del error y exactitud en la clasificación de las imágenes, obtenidos de las 10 iteraciones de validación cruzada, y usando aumento de datos equitativo con bolsas de datos. El optimizador B (*SGD* con tasa de aprendizaje de 0,000001) fue el que obtuvo los mejores resultados para esta estrategia de aumento de datos, y aún así, los resultados que se observan en los gráficos son bastante malos.

Para empezar, el error se mantuvo casi constante a través de 1.000 épocas de entrenamiento, aumentando solo en pequeñas cantidades, lo que provoca que, la exactitud en la clasificación, sea muy cercana a 0 en la validación. Esto se debe a que, con las imágenes de entrenamiento, el error de clasificación también fue constante en todas las épocas, indicando así que el modelo no fue capaz de generalizar ninguna característica a medida que se le mostraban las imágenes al modelo.

Esto se debe principalmente, a la baja cantidad de imágenes de entrenamiento obtenidas de las bolsas de datos, utilizadas para balancear las clases de cabezas de espermatozoides (400 imágenes por cada clase, dando un total de 2.000). Con tan pocas imágenes por clase, los modelos no fueron capaces de obtener la información de la forma de las cabezas de espermatozoides, y esto quedará demostrado a medida que se muestren los siguientes gráficos del afinamiento.

Entrenamiento con optimizador B y AD con bolsas de datos

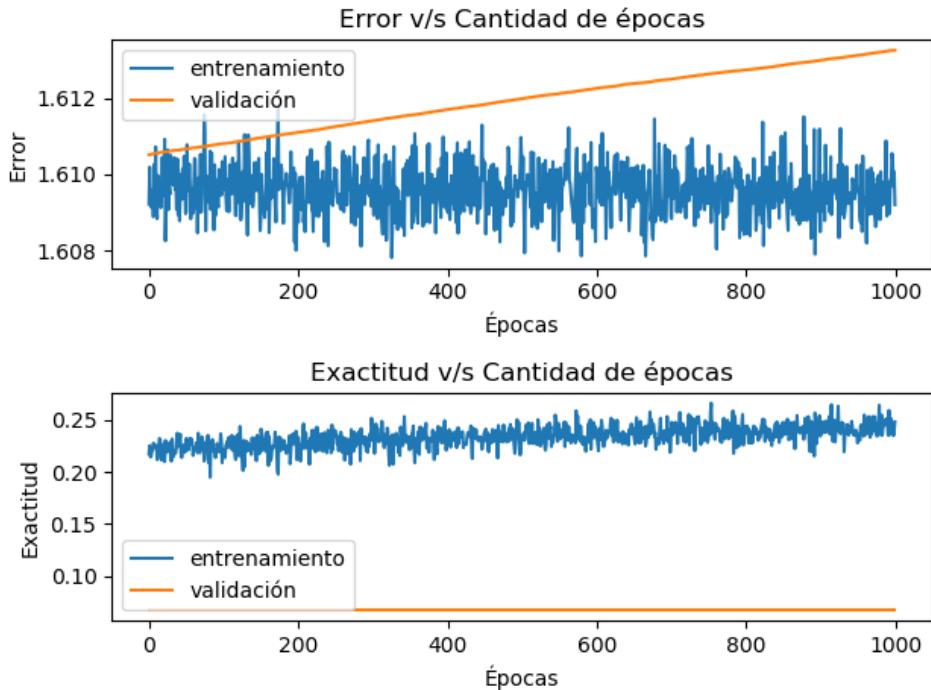


Figura 6.4: Error y exactitud en el entrenamiento de la red *RBC* con AD equitativo, usando bolsas de igual tamaño. El optimizador B corresponde a *SGD* tasa de aprendizaje de 0,000001. Los resultados en la validación corresponden a las imágenes que el modelo no ha visto durante el entrenamiento.

Fuente: Elaboración propia, 2020.

Para el caso del aumento de datos equitativo, utilizando pesos por clase para el balanceo de clases, el promedio de los resultados del entrenamiento se ven en la Figura 6.5, en donde el optimizador D (*ADAM* con tasa de aprendizaje de 0,00001) obtuvo los mejores resultados con esta estrategia de aumento de datos.

Se puede observar que con esta estrategia de aumento de datos, se logró disminuir el error de la validación, en comparación al caso anterior. Al mismo tiempo, la exactitud en la clasificación de las imágenes de validación, logró aumentar de manera lenta a lo largo de las 1.000 épocas, algo que tampoco se vio en el gráfico anterior.

Aumentando la cantidad de imágenes durante el entrenamiento, los modelos fueron capaces de generalizar solo algunas características de la forma de las cabezas de espermatozoides, ya que la exactitud en las imágenes de validación, solo alcanzó un 25% en promedio.

La cantidad de imágenes del entrenamiento demuestra ser un factor importante en la obtención de mejores resultados en la clasificación, y queda claro que la red convolucional *RBC* necesita de gran cantidades de imágenes, para reconocer las clases de las cabezas de espermatozoides humanos. Esto se refuerza incluso con los siguientes resultados, los que corresponden al afinamiento de parámetros con aumento de datos dirigido.

Entrenamiento con optimizador D y AD con pesos por clases

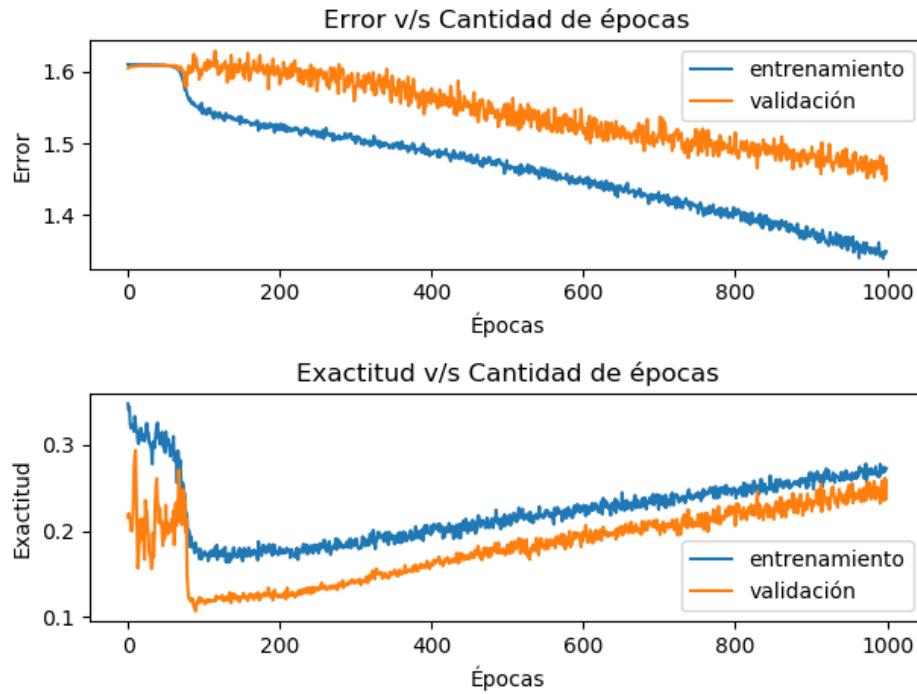


Figura 6.5: Error y exactitud en el entrenamiento de la red *RBC* con AD equitativo, usando pesos por clases. El optimizador D corresponde a *ADAM* con tasa de aprendizaje de 0,00001.

Fuente: Elaboración propia, 2020.

Los mejores resultados del afinamiento se obtuvieron utilizando la estrategia de aumento de datos dirigidos. Para este caso, primero hay que resaltar un optimizador que, obtuvo resultados bastante variados en las distintas iteraciones de validación cruzada.

En la Figura 6.6 se ve el error en el entrenamiento y validación, utilizando el optimizador C (*ADAM*), pero en dos iteraciones distintas de validación cruzada. Acá se ven dos casos opuestos en donde, el error en la validación se mantuvo estable a lo largo de las 1.000 épocas, obteniendo incluso valores de error cercanos a 0,70 en las primeras 200 épocas, y luego, en una siguiente iteración, en menos de 200 épocas, ya se puede observar un incremento considerable del error en la validación. Esto demuestra la existencia de *overfitting* en la clasificación, ya que el modelo ha memorizado características de las imágenes de entrenamiento, pero al presentarse nuevas imágenes (validación), no sabe a qué clase corresponde estas

cabezas de espermatozoides.

Es importante mencionar que durante todo el experimento 1, durante cada iteración de validación cruzada, los pesos iniciales de las neuronas de la red son aleatorios, por lo que es posible que esta diferencia de resultados esté dada por este factor. Para descartar la dependencia de los resultados con los pesos iniciales, se escogió realizar nuevamente entrenar los modelos con esta configuración, pero esta vez, se utilizaron los mismos pesos aleatorios iniciales en cada una de las iteraciones de validación cruzada, y estos pesos corresponden a los que se usaron al obtener el primer gráfico de la Figura 6.6.

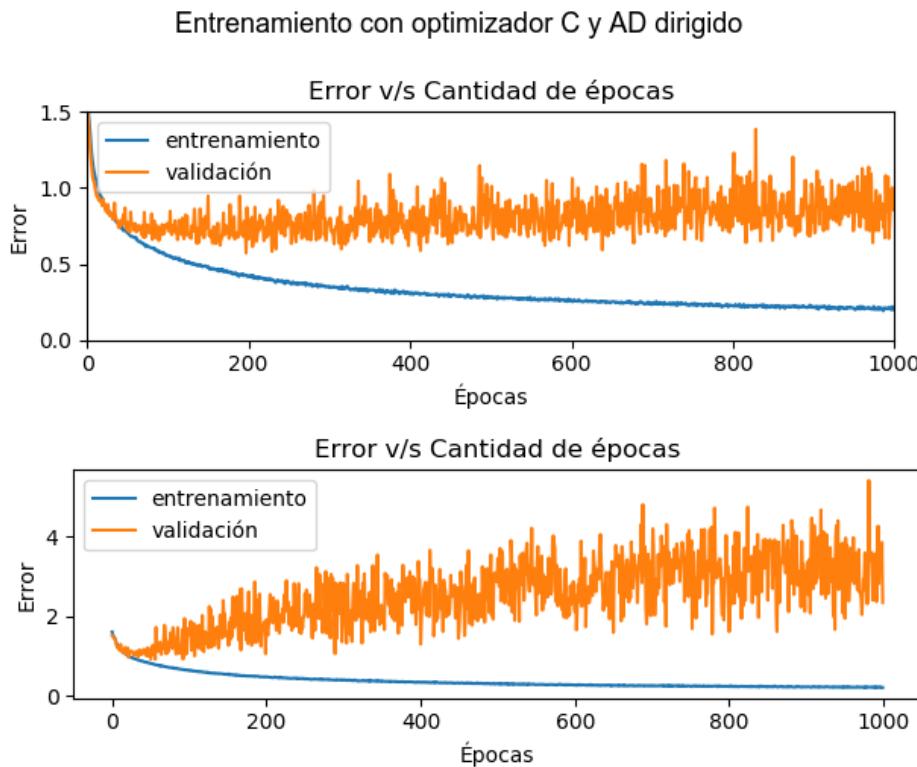


Figura 6.6: Error en el entrenamiento de la red *RBC* con AD dirigido, utilizando el optimizador C (*ADAM* con tasa de aprendizaje de 0,001). Ambos gráficos representan el error en la clasificación utilizando la misma configuración pero en iteraciones distintas de la validación cruzada. El optimizador C corresponde a *ADAM* con tasa de aprendizaje de 0,001.

Fuente: Elaboración propia, 2020.

A partir de esto, en la Figura 6.7, se observa finalmente el promedio de los resultados en la clasificación realizada con los mismos pesos aleatorios iniciales. El problema de la diferencia del error entre iteraciones persistió, por lo que los resultados se deben, principalmente, a las imágenes que se utilizaron en el entrenamiento, y no a los pesos iniciales. Esto quiere decir que, existen imágenes más fáciles de clasificar que otras.

A pesar de la diferencia entre los resultados obtenidos en algunas iteraciones de

validación cruzada, el error promedio obtenido alcanzo valores menores que utilizando el aumento de datos equitativo, alcanzando un error cercano a 1,0 y una exactitud en la validación del 50% aproximadamente.

Entrenamiento con optimizador C y AD dirigido

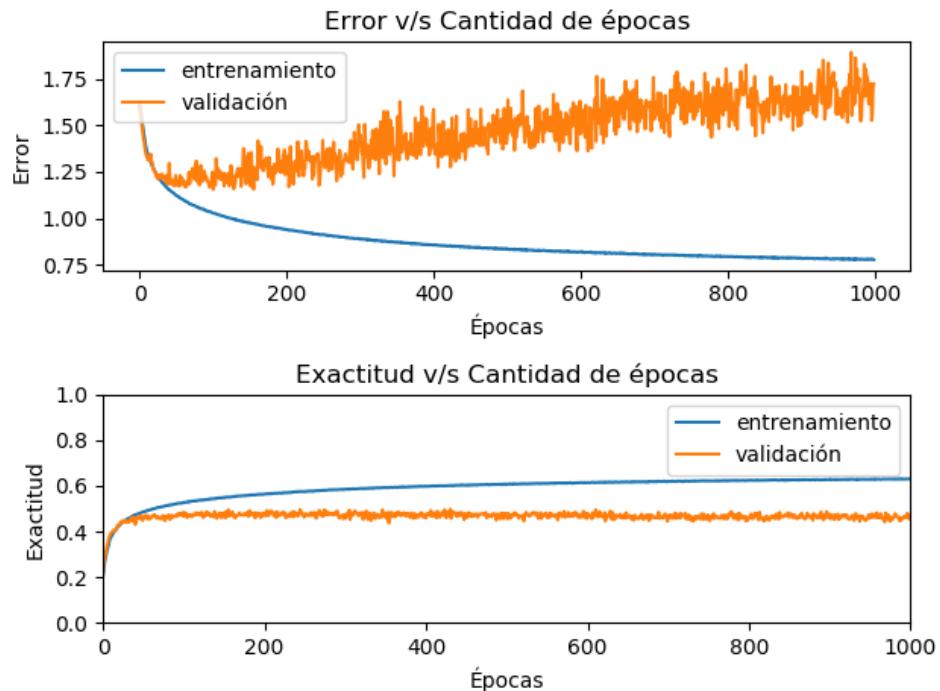


Figura 6.7: Error y exactitud promedio en el entrenamiento de la red *RBC* con AD dirigido, utilizando el optimizador C. Esto se obtuvo utilizando los mismos pesos aleatorios iniciales en todas las iteraciones de la validación cruzada. El optimizador C corresponde a *ADAM* con tasa de aprendizaje de 0,001.

Fuente: Elaboración propia, 2020.

A pesar de esto, los mejores resultados para el aumento de datos dirigido, se obtuvieron con el optimizador D (*ADAM* con tasa de aprendizaje de 0,00001), en donde el gráfico promedio obtenido del entrenamiento se ve en la Figura 6.8.

Para este caso, se logró disminuir el error en la validación durante todas las épocas de entrenamiento, alcanzando un error mínimo de 0,80, y lo mismo ocurrió con la exactitud, la que logró aumentar durante todas las épocas, obteniendo así en promedio una exactitud aproximada del 68% con las imágenes de validación.

Es importante destacar lo lento de este proceso, ya que estos resultados se obtuvieron luego de las 1.000 épocas de entrenamiento. En comparación, el trabajo de Angluin et al. (2016) de clasificación de las imágenes de glóbulos rojos, los resultados publicados se obtuvieron con solo 50 épocas de entrenamiento. Esto quiere decir que el problema de

clasificación de imágenes de espermatozoides humanos es mucho más complejo para los modelos de la red convolucional *RBC*

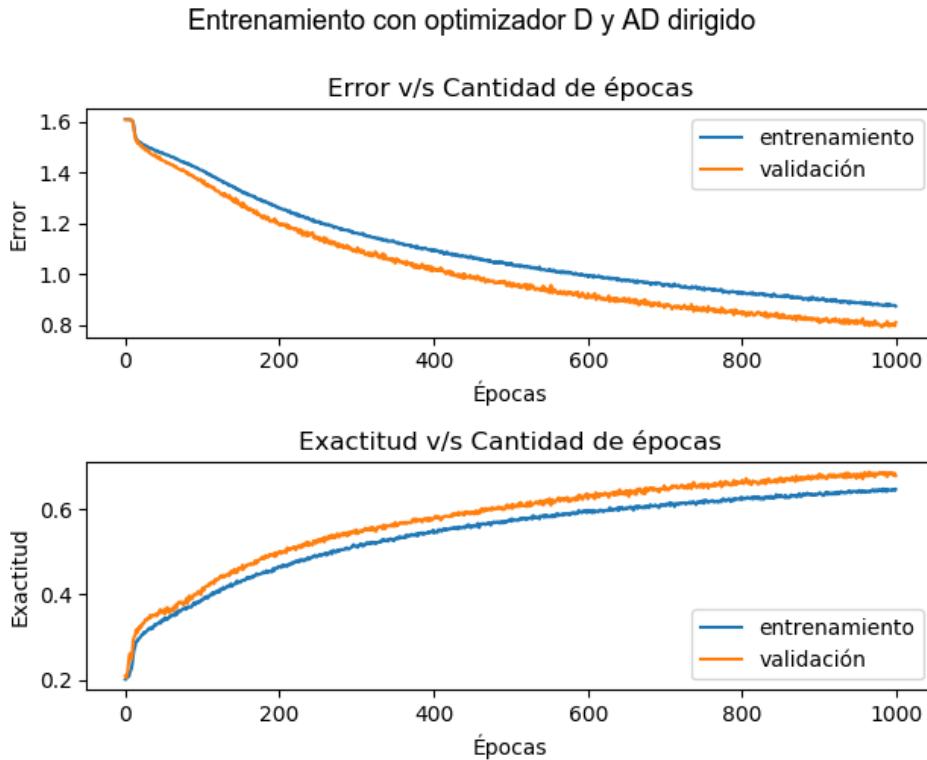


Figura 6.8: Error y exactitud en el entrenamiento de la red *RBC* con AD dirigido. El optimizador D corresponde a *ADAM* con tasa de aprendizaje de 0,00001.

Fuente: Elaboración propia, 2020.

Debido a los resultados presentados en la Figura 6.8, se utilizará en este experimento, la red *RBC* desde cero, con el optimizador D y 1.000 épocas para entrenar los modelos y clasificar las imágenes del conjunto de evaluación.

Red convolucional ResNet50

Utilizando la red convolucional *ResNet50*, nuevamente los peores resultados se obtuvieron utilizando el aumento de datos equitativo, con bolsas de datos para balancear las clases. En este caso, el optimizador B (*SGD* con tasa de aprendizaje de 0,000001) fue el que obtuvo los mejores resultados de error y exactitud en la validación para esta estrategia de aumento de datos, y la gráfica se encuentra en la Figura 6.9.

Para este caso se puede observar como el error en la validación solo aumenta durante las épocas, mientras que para la exactitud en la validación, se mantuvo por debajo del

20% durante todo el entrenamiento. Considerando que este es la configuración con mejores resultados de esta estrategia de aumento de datos, aún así el error y la exactitud en la validación, son los peores hasta el momento.

Esto solo refuerza la idea que, para la construcción de modelos complejos, se necesita una gran cantidad de datos, ya que en el caso de no tenerlos, se obtienen resultados como los presentes en la Figura 6.9

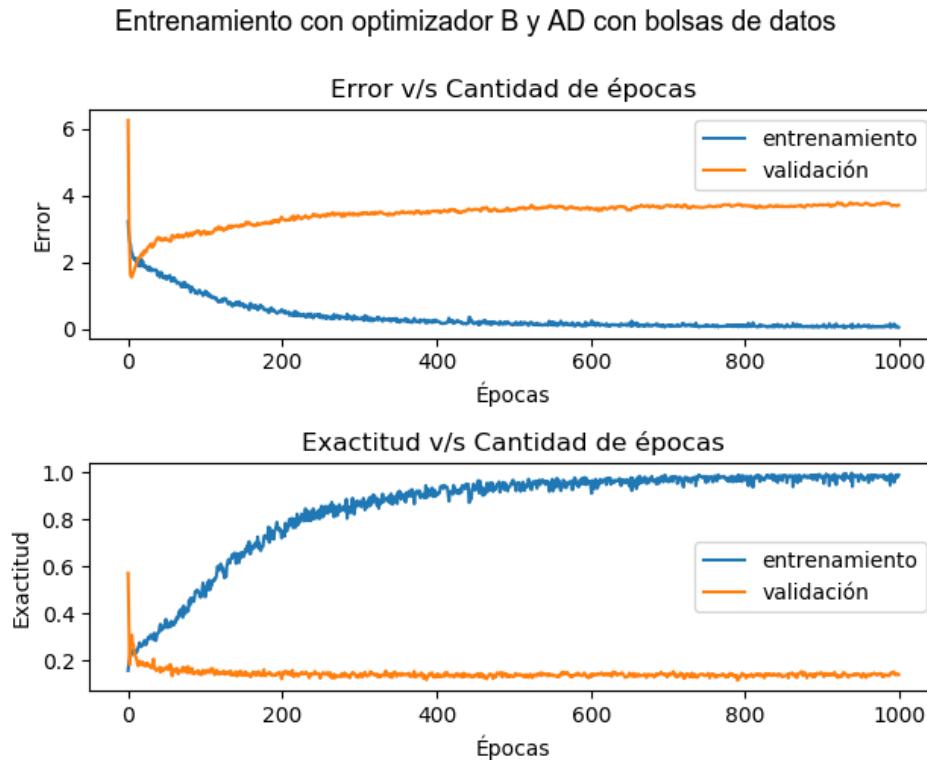


Figura 6.9: Error y exactitud en el entrenamiento de la red *ResNet50* con AD equitativo, usando bolsas de igual tamaño. El optimizador B corresponde a *SGD* con tasa de aprendizaje de 0,000001.

Fuente: Elaboración propia, 2020.

Utilizando el aumento de datos equitativo con pesos por clases, los resultados del entrenamiento mejoraron también para la red *ResNet50*. Los mejores resultados se obtuvieron esta vez con el optimizador A (*SGD* con tasa de aprendizaje de 0,0001), que se pueden ver en la Figura 6.10.

Para este caso, aunque el error tuvo valores cercanos a 2,0 durante el entrenamiento, es importante destacar que la exactitud en la clasificación del conjunto de validación logró ser de un 50% aproximadamente, lo cual no es menor si se compara con los resultados del caso anterior (la Figura 6.9).

A pesar del aumento de la exactitud promedio en la validación, se observa que el

error no logró disminuir en el entrenamiento. Esto se debe a que, como se ve en los resultados con las imágenes de entrenamiento, Se alcanzó un error 0 y una exactitud del 100% muy tempranamente, por lo que el modelo a partir de alcanzar este punto, no es capaz de generalizar más características, estancando el error y la exactitud en la clasificación de las imágenes del conjunto de validación.

Lo anterior se debe principalmente, a la complejidad de la red convolucional *ResNet50*, definida por la profundidad de esta. Esto hizo que la red encontrara un mínimo local en el error de manera muy rápida, sin poder escapar de este para llegar al mínimo global esperado. Más adelante, en el apartado de los resultados obtenidos con aprendizaje por transferencia, se retoma nuevamente este tópico, comprobando nuevamente que la complejidad de *ResNet50* es un factor en contra en este problema y el conjunto de imágenes.

Entrenamiento con optimizador A y AD con pesos por clases

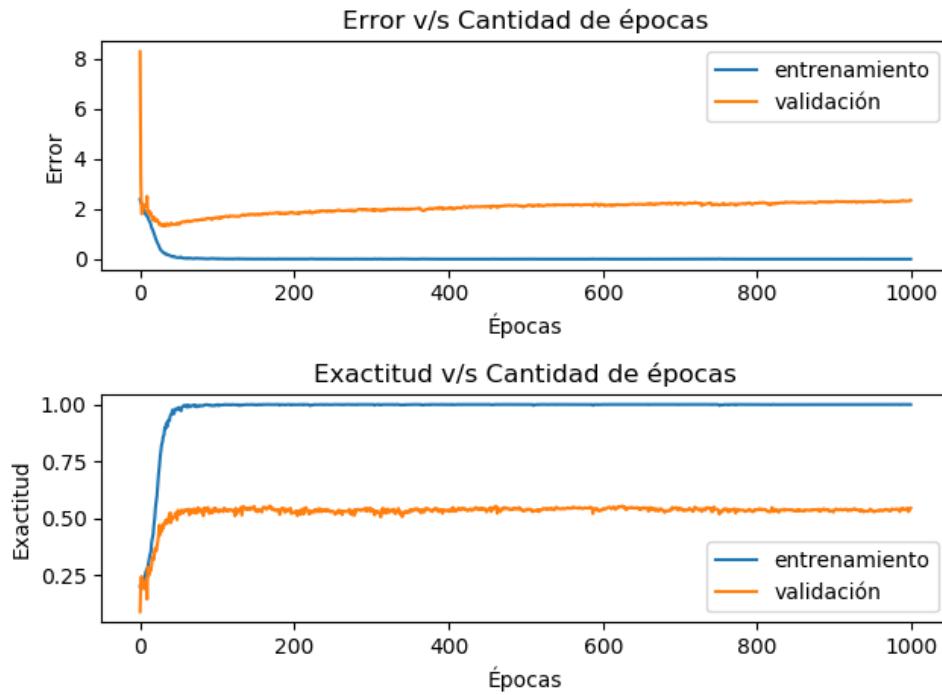


Figura 6.10: Error y exactitud en el entrenamiento de la red *ResNet50* con AD equitativo, usando pesos por clases. El optimizador A corresponde a *SGD* con tasa de aprendizaje de 0,0001.

Fuente: Elaboración propia, 2020.

Finalmente, los mejores resultados en el afinamiento de *ResNet50*, se obtuvieron nuevamente con el aumento de datos dirigido, y el mejor optimizador en este caso fue *SGD* (con tasa de aprendizaje de 0,0001), y el error y exactitud promedio se ve reflejado en la Figura 6.11.

Durante estos entrenamientos, se logró alcanzar una exactitud promedio de 58% aproximadamente, pero nuevamente presentando el problema establecido anteriormente, en

donde los modelos ya no puede mejorar los resultados en la validación, debido a que alcanzó un error 0 y exactitud 100% con las mismas imágenes de entrenamiento. Considerando el error 0, entonces los pesos de las neuronas del modelo dejan de actualizarse, estancando nuevamente el error y exactitud en la clasificación de las imágenes del conjunto de validación.

Incluso utilizando grandes cantidades de datos con el aumento de datos dirigido, el problema anterior persistió en el entrenamiento de los modelos, indicando así que la red convolucional *ResNet50* no pudo encontrar nuevas características a partir de las transformaciones de imágenes definidas en este trabajo, lo cual, se debe nuevamente, a la complejidad de esta red convolucional.

Considerando que los resultados vistos en el gráfico 6.11 siguen siendo que peores que los obtenidos con la red convolucional *RBC*, es que durante este primer experimento, se ha escogido utilizar solamente la red *RBC* para la clasificación morfológica de cabezas de espermatozooides humanos.

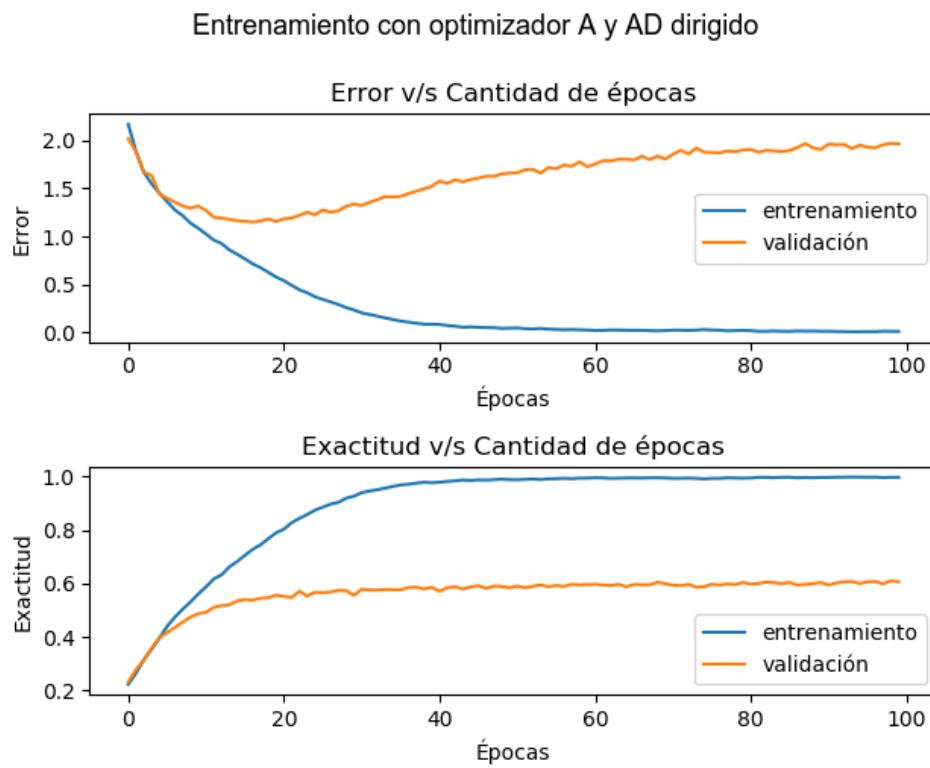


Figura 6.11: Error y exactitud en el entrenamiento de la red *ResNet50* con AD dirigido. El optimizador A corresponde a *SGD* con tasa de aprendizaje de 0,0001.

Fuente: Elaboración propia, 2020.

Comparaciones

Primero que todo, para ambas redes convolucionales elegidas, los resultados fueron mejores a medida que aumentaban las imágenes de entrenamiento. A pesar de que teóricamente, este tipo de modelos complejos de *deep learning* tienen mejores resultados con grandes cantidades de imágenes, hay que tener en cuenta que, las más de 30.000 imágenes de entrenamiento obtenidas en el aumento de datos dirigido, fueron generadas solamente a partir de 790 imágenes originales aproximadamente, por lo que a pesar de haber aumentado la cantidad de datos para entrenar los modelos, también era posible incrementar las probabilidades de tener *overfitting* al no tener suficientes imágenes distintas para poder generalizar las características de las cinco clases de cabezas de espermatozoides humanos.

Sin embargo, la red *ResNet50* alcanzó un límite de aprendizaje con las imágenes generadas, incluso con el aumento de datos dirigido, lo que significó un estancamiento en los resultados de esta red convolucional. En contraste, la red *RBC*, a ser menos profunda y menos compleja que su contra parte, no tuvo este inconveniente, por lo que logró obtener mejores resultados para el error y exactitud en la clasificación de las imágenes de validación.

De hecho, ocurrió justamente lo contrario durante el entrenamiento de la red convolucional *RBC*, en donde el aprendizaje fue tan lento, que le tomó a los modelos 1.000 épocas de entrenamiento para obtener los mejores resultados de error y exactitud. Esto quiere decir que, es posible que al aumentar la tasa de aprendizaje para esta red y optimizador, se puedan obtener mejores resultados, pero no se optó por esta opción, debido a que tendría que aplicarse esta configuración para todas las otras estrategias de aumento de datos, y también para la red *ResNet50*, aumentando aún más las configuraciones a probar y el tiempo de entrenamiento, ya que es necesario que ambas redes deben sean comparadas en igualdad de condición.

6.3.3 Resultados de la clasificación morfológica

Luego de diez iteraciones de validación cruzada, se obtuvieron diez matrices de confusión correspondiente a la clasificación morfológica de cabezas de espermatozoides humanos. En la Tabla 6.4 se puede observar, la matriz de confusión promedio, y normalizada, de las diez iteraciones, para los modelos entrenados correspondientes a la red convolucional *RBC*. Las clasificaciones correctas se encuentran en la diagonal de las matrices, que fue marcada de color amarillo, mientras que las filas indican los valores reales de las imágenes de evaluación, y las columnas son las clases predichas por los modelos.

De esta tabla, se puede la baja cantidad de clasificaciones correctas para las clases pequeña, teniendo solo 0,05 de verdaderos positivos, y siendo la más baja dentro de las cinco clases de formas de cabezas de espermatozoides, mientras que por otro lado, la clase con mayor cantidad de clasificaciones correctas, fue la amorfa, alcanzando 0,71 de verdaderos positivos.

Por otro lado, si se observa la columna de la clase amorfa, se ve que en general, los modelos han tendido a predecir las otras clases, con la forma amorfa de cabezas de espermatozoides, y esto se debe a que el modelo tiene un problema al identificar las primeras cuatro clases, en comparación de la clase amorfa de espermatozoides humanos.

En la Figura 6.12 se observa la curva *ROC* de la clasificación realizada por un modelo entrenado en una iteración de la validación cruzada. Aquí se observa nuevamente que existe un problema en la clasificación de la clase amorfa, lo cual aporta información respecto al problema visto en la columna de clasificación amorfa en la matriz de confusión.

En este caso, el modelo tiene bastantes problemas con esta última clase, teniendo un área bajo la curva no muy por encima del 0,5 (cercana a la diagonal), indicando que el modelo casi no tiene capacidad de separar esta clase de las otras, y es por esto que se ve en la matriz de confusión tantas predicción realizadas en esta clase. Ahora bien, viendo la curva de *ROC* de las otras cuatro clases de formas de cabezas de espermatozoides, se ve que tienen aproximadamente la misma área bajo la curva, por lo que la capacidad de identificar las otras cuatro clases, debiese ser similar.

		Clase predicha					Total real (# imágenes)
		Normal	Cónica	Piriforme	Pequeña	Amorfa	
Clase real	Normal	0,40	0,00	0,03	0,00	0,57	1 (30)
	Cónica	0,00	0,44	0,01	0,00	0,54	1 (68)
	Piriforme	0,00	0,00	0,39	0,00	0,61	1 (23)
	Pequeña	0,27	0,05	0,00	0,05	0,64	1 (22)
	Amorfa	0,09	0,14	0,06	0,01	0,71	1 (197)

Tabla 6.4: Matriz de confusión promedio, y normalizada, de la clasificación realizada por la red *RBC* desde cero. Los resultados han sido aproximados a la centésima.

Fuente: Elaboración propia, 2020.

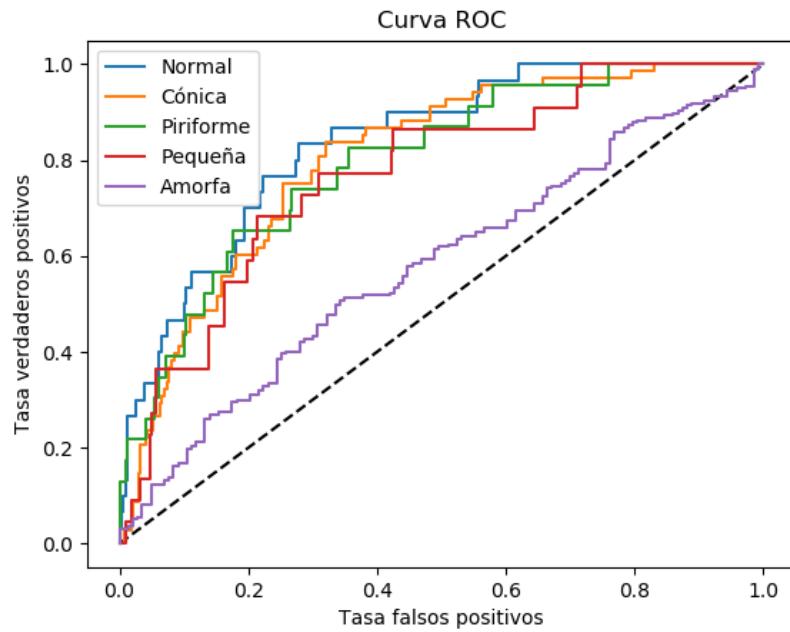


Figura 6.12: Curva *ROC* en una iteración de la clasificación morfológica de cabezas de espermatozoides, utilizando la red *RBC* con entrenamiento desde cero.

Fuente: Elaboración propia, 2020.

Como ya se ha presentado en el estado del arte ((Chang et al., 2017b), (Riordon et al., 2019) y Iqbal et al. (2020)), clasificar las cabezas de espermatozoides con la forma amorfa, ha sido un desafío importante, debido a las diferencias que existen en las imágenes de esta misma clase. Lamentablemente, en estos resultados expuestos, nuevamente se puede observar el problema con la clasificación de esta clase, al tener una baja capacidad de identificar estas cabezas de espermatozoides.

Es importante recordar que, el conjunto con el que se ha trabajado solamente tiene 1.132 imágenes de su totalidad, y en su mayoría, eran de la clase amorfa. Así como los resultados en el afinamiento, fueron mejorando a medida que se generaban más imágenes para el entrenamiento, se debe entender que estas imágenes aumentadas están limitadas por la cantidad original, así que las clases con menor representatividad son las más afectadas a la hora de que los modelos generalicen las características de esas clases.

Especialmente con esta red convolucional *RBC*, se puede observar un ejemplo de esto en la matriz de confusión anterior, en donde cantidad de aciertos obtenidos en la clase pequeña, fue la más baja entre las cinco clases estudiadas. Es importante mencionar esto debido a que los resultados presentados por el estado del arte mencionado no presentaron este inconveniente, por lo que la red *RBC* tiene un problema al intentar generalizar características con pocos datos.

De hecho, retomando un poco los gráficos de error y exactitud presentados en el afinamiento de parámetros, se vio el lento entrenamiento de la red *RBC*, lo que indicaba que quizás la red no es suficientemente compleja para este problema de clasificación, y finalmente acá, en los resultados con el conjunto de evaluación, se observa nuevamente esto en forma del sesgo en la clasificación hacia la clase amorfa, perjudicando especialmente a las clases con menor cantidad de imágenes.

Muy por el contrario, en el afinamiento de parámetros, se pudo observar como la red *ResNet50*, alcanzaba demasiado rápido el error mínimo en el entrenamiento, deteniendo el aprendizaje y obteniendo malos resultados con las imágenes que el modelo no ha visto durante el entrenamiento. Irónicamente, la red *ResNet50* fue demasiado compleja para este problema de clasificación, generando el *overfitting* en los modelos entrenados, demostrando otro ejemplo más del problema de la compensación entre sesgo y varianza.

Utilizando la matriz de confusión anterior, se presenta la Tabla 6.5 con las métricas de evaluación correspondientes a la clasificación realizada por la red *RBC*.

En primer lugar, se puede observar la baja precisión en la clasificación de la clase amorfa, indicando lo dispersa de la clasificación hecha por los modelos con respecto a esta clase, lo cual era esperado al ver la columna de la predicción de la clase amorfa en la matriz de confusión en la Tabla 3.4. Por otro lado, la especificidad de esta clase tampoco fue muy alta, debido a que los modelos también tuvieron hartos errores al clasificar las clases que no eran la forma amorfa, como se observa en la cantidad de predicciones hechas de forma incorrecta en la columna de la clase amorfa en la matriz de confusión.

La clase pequeña fue la clase con la peor cantidad de aciertos, y esto se ve reflejado en la sensibilidad de esta clase, que corresponde a las imágenes de clase pequeña que efectivamente fueron clasificados con esta forma.

Para el resto de las cuatro clases de formas de cabezas de espermatozoides, en general se observa una clasificación precisa y con buenos valores en la especificidad, pero que se contrasta con los bajos valores de sensibilidad. Esto quiere decir que, la clasificación realizada por los modelos, no fue tan dispersa, a excepción de la clase amorfa, y además, los mejores resultados fueron obtenidos al identificar las imágenes no pertenecientes a la clase, ya que los valores de especificidad fueron mejores que los obtenidos para la sensibilidad.

Métrica/Clase	Normal	Cónica	Piriforme	Pequeña	Amorfa	Promedio
Precisión	52.35%	70.74%	79.02%	81.74%	23.05%	61.38%
Sensibilidad	40.00%	44.12%	39.13%	4.55%	70.56%	39.67%
Especificidad	81.31%	89.42%	93.88%	99.48%	35.17%	79.85%

Tabla 6.5: Métricas de evaluación de la clasificación realizada por la red *RBC*, obtenida de la matriz de confusión normalizada. Los resultados han sido aproximados a la centésima.

Fuente: Elaboración propia, 2020.

Finalmente, en la Tabla 6.6, se observa una comparación de las métricas expuestas en el estado del arte presentado en este trabajo. En esta se ven reflejada la sensibilidad en la clasificación, y el promedio de estas, que corresponde a la exactitud global en la clasificación morfológica de cabezas de espermatozoides humanos.

Lo primero a destacar es la sensibilidad obtenida (o tasa de verdaderos positivos como se mencionó en el Capítulo 3), la cual para la clase amorfa es mucho mayor que en el estado del arte. Lamentablemente, este número se encuentra opacado principalmente por que los modelos tienen poca capacidad de separar la clase amorfa, vistos en la curva *ROC* y en la matriz de confusión normalizada, por lo que este resultado es la consecuencia de un sesgo hacia esta clase de cabeza de espermatozoide.

Para el resto de las clases, la mayor diferencia de ve en la sensibilidad de la clase pequeña con respecto a los otros trabajo, el cual es bastante bajo, por lo que el modelo no ha podido generalizar de manera correcta las características que sí corresponden a esta morfología de cabeza de espermatozoide. A partir de esto, se deduce finalmente que la red *RBC* no es lo suficientemente compleja para poder clasificar las formas de las cabezas de espermatozoides, cuando se tienen muy pocos datos.

En el contexto de este problema, es importante identificar la forma real de las cabezas de los espermatozoides de un paciente, para así diagnosticar el potencial fértil de este. Esta es justamente lo que se está midiendo con la sensibilidad, por lo que corresponde a las cabezas que fueron identificadas con la forma correcta, y es justamente la métrica que tuvo los peores resultados en la clasificación morfológica realizada con la red convolucional *RBC*.

Red/Sensibilidad	S(N)	S(C)	S(Pi)	S(Pe)	S(A)	Exactitud
Red convolucional <i>RBC</i>	0,40	0,44	0,39	0,05	0,71	0,40
Red convolucional <i>VGG16</i> (Riordon et al., 2019)	0,67	0,57	0,69	0,78	0,38	0,62
Red convolucional propuesta por Iqbal et al. (2020)	0,70	0,79	0,62	0,71	0,57	0,68

Tabla 6.6: Comparación de métricas finales del primer experimento, con respecto al estado del arte.

Fuente: Elaboración propia, 2020.

6.4 SEGUNDO EXPERIMENTO: ENTRENAMIENTO CON APRENDIZAJE POR TRANSFERENCIA

6.4.1 Protocolo de experimentación

El procedimiento es parecido al realizado en el experimento 1, solo que en esta oportunidad, se tiene que realizar el paso previo de pre-entrenamiento de las redes, con el conjunto de imágenes de glóbulos rojos (Angluin et al., 2016).

Algoritmo 6.2: Implementación del experimento 2: entrenar modelos con aprendizaje por transferencia

- 1 **para** cada red *R* **hacer**
 - 2 Construir arquitectura de *R* para clasificar las imágenes de glóbulos rojos;
 - 3 **para** cada optimizador *O* **hacer**
 - 4 Entrenar *R* con optimizador *O*, utilizando las imágenes de glóbulos rojos de Angluin et al. (2016) Graficar error versus cantidad de épocas;
 - 5 **fin**
 - 6 Elegir mejor optimizador para entrenar red *R*;
 - 7 Entrenar *R* con las configuraciones establecidas de optimizador y número de épocas;
 - 8 Obtener modelo pre-entrenado;
 - 9 **fin**
 - 10 Volver a construir *RBC* y *ResNet50* para la clasificación morfológica de cabeza de espermatozoides;
 - 11 Cargar los pesos pre-entrenados en ambas redes;
 - 12 Repetir Algoritmo 6.1 las redes *RBC* y *ResNet50* con pesos pre-entrenados;
-

6.4.2 Pre-entrenamiento de las redes convolucionales

Para la red convolucional de *RBC*, los mejores resultados del pre-entrenamiento, utilizando el optimizador de *ADAM*, se ven reflejados en la Figura 6.13, en donde se muestra como el error disminuyó durante las primeras 50 épocas de entrenamiento. A medida que aumentaron las épocas de entrenamiento, el error de la validación fue aumentando lentamente. Al mismo tiempo, se observa que el error con el conjunto de entrenamiento, disminuyó a casi cero, por lo

que la red ya no pudo seguir aprendiendo características de las imágenes luego de las primeras 50 épocas aproximadamente. Debido a esto, para pre-entrenar los pesos de la red *RBC*, se utilizará este optimizador, entrenando hasta la época 50.

Por otro lado, para la red convolucional *ResNet50*, los mejores resultados del pre-entrenamiento se observan en la Figura 6.14, obtenidos con el optimizador *SGD*. En este caso, el error disminuye rápidamente para mantenerse constante después. Al igual que para el pre-entrenamiento de la otra red convolucional escogida, el error durante el entrenamiento alcanzó a llegar a casi 0, por lo que es difícil para la red aprender más características de la forma de estas células, a partir de las imágenes proporcionadas durante el entrenamiento.

Como no se alcanza a percibir muy detalladamente el error, en la misma Figura se presenta un entrenamiento con una menor cantidad de épocas, solo para demostrar de mejor forma aproximadamente en qué momento el error deja de disminuir. Para este caso, se ha escogido pre-entrenar la red *ResNet50* con el optimizador *SGD* y durante 40 épocas.

Ambos casos de pre-entrenamiento se realizaron en varias iteraciones con la técnica de validación cruzada, en donde la tendencia del error fue aproximadamente la misma en las Figuras 6.13 y 6.14, por lo que se ha podido rescatar un caso con mejores resultados para el pre-entrenamiento de estas redes convolucionales.

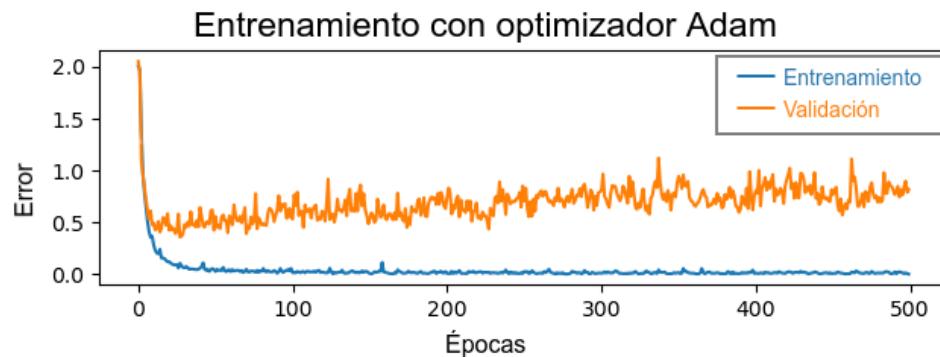


Figura 6.13: Error durante el pre-entrenamiento de la red *RBC* con el optimizador *Adam*.

Fuente: Elaboración propia, 2020.

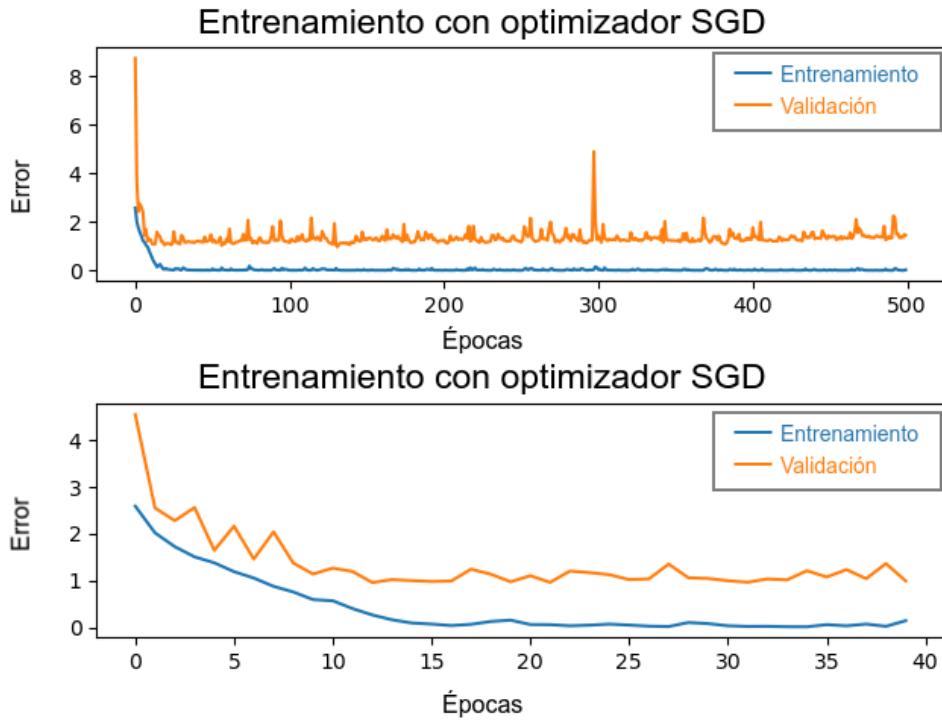


Figura 6.14: Error durante el pre-entrenamiento de la red *ResNet50* con el optimizador *SGD*. En este caso, la segunda fila corresponde a un entrenamiento con la misma red y optimizador, pero con menos cantidad de épocas, solamente para obtener mejor detalle del error.

Fuente: Elaboración propia, 2020.

6.4.3 Resultados del afinamiento de parámetros

Para el segundo experimento, se probaron nuevamente las doce combinaciones posibles de parámetros, para las dos redes convolucionales escogidas. Del mismo modo que en el primer experimento, se mostrarán los mejores parámetros por estrategia de aumento de datos, para no exponer tantos gráficos, considerando la cantidad de combinaciones que se probaron.

Red convolucional RBC

Nuevamente, se comienza por los resultados del entrenamiento, con la estrategia de aumento de datos equitativo, con bolsas de datos para balancear las clases. En la Figura 6.15 se observa el comportamiento del error y exactitud en el entrenamiento y validación, a lo largo de las 1.000 épocas en que fueron entrenados los modelos. Este gráfico además corresponde al obtenido con el optimizador B (*SGD* con tasa de aprendizaje de 0,000001), además de ser el promedio de las diez iteraciones de validación cruzada utilizado para el afinamiento de

parámetros.

La diferencia de los resultados del entrenamiento desde cero, con pesos aleatorios (Figura 6.4), se ve que los resultados tienden en un principio, a disminuir y aumentar el error y exactitud respectivamente, pero de forma inmediata, el error deja de disminuir, y la exactitud comienza a disminuir hasta tender a una exactitud igual o menor del 10% aproximadamente.

En este caso, en las primeras épocas, se ve una mejora en la exactitud con respecto al experimento 1, debido al aprendizaje previo de las características de las formas de los glóbulos rojos, pero a medida que avanzan las épocas, se ve como el error y exactitud en la validación, tienden a empeorar, incluso hasta alcanzar valores similares a los del afinamiento del experimento 1, observados en la Figura 6.4

A pesar de esto, el error en la validación, no disminuye más de 1.5, lo que no es una diferencia mayor con respecto al primero experimento, ya que al buscar un error cercano a 0, se espera que este pueda disminuir de 1,0 al menos, lo que no se logró con esta configuración.

En consecuencia, se puede decir que solo en las primeras épocas se pueden tener mejores resultados con aprendizaje por transferencia, con estos parámetros utilizados. Sin embargo, al igual que en el experimento 1, esta estrategia es la que ha entregado los peores resultados de error y exactitud en la validación, al probar las combinaciones de parámetros, por lo que no se utilizará aumento de datos equitativo, con bolsas de datos, en este experimento.

Entrenamiento con optimizador B y AD con bolsas de datos

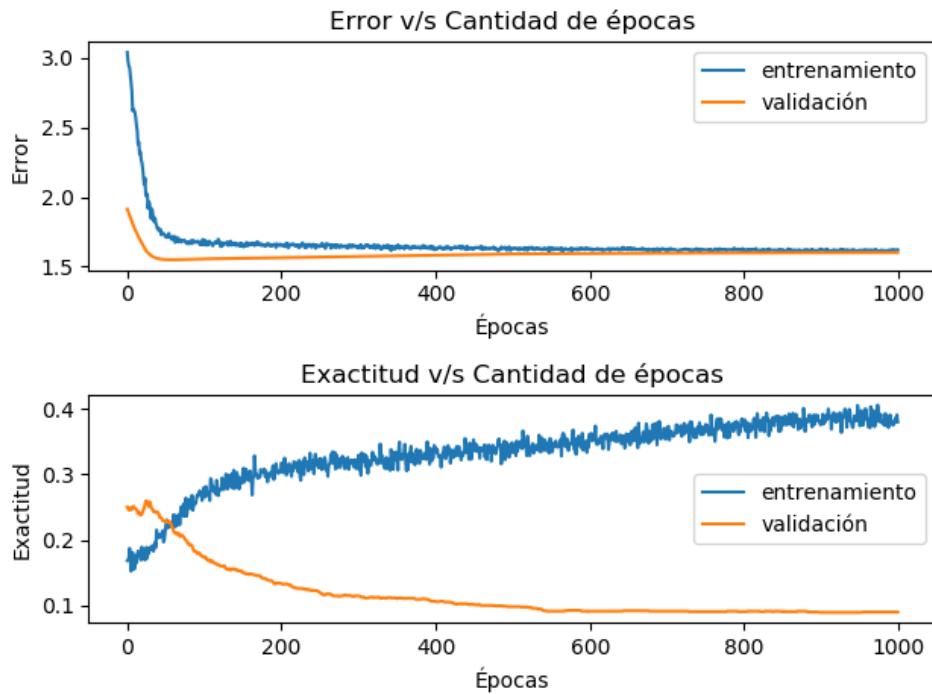


Figura 6.15: Error y exactitud en el entrenamiento de la red *RBC* pre-entrenada, con AD equitativo, usando bolsas de igual tamaño. El optimizador B corresponde a *SGD* con tasa de aprendizaje de 0,000001. Los resultados en la validación corresponden a las imágenes que el modelo no ha visto durante el entrenamiento.

Fuente: Elaboración propia, 2020.

Por otro lado, en la Figura 6.16 se ve los resultados de error y exactitud promedio en el entrenamiento y validación, luego de las iteraciones de validación cruzada, utilizando el aumento de datos equitativo con pesos por clases, y esta vez, el mejor optimizador para este caso fue el D (*ADAM* con tasa de aprendizaje de 0,00001).

Comparándolo con el gráfico anterior, se ve que el error al final de las 1.000 épocas es bien similar, aunque la exactitud en la validación, logra aumentar esta vez, a mas del 20%, en vez de disminuir. Sin embargo, a pesar de existir una mejora en esta métrica, el error sigue siendo alto y la exactitud sigue siendo baja, en comparación a otros resultados obtenidos en este mismo trabajo.

Comparando este gráfico con el primer experimento, con entrenamiento desde cero (Figura 6.5), se ve que el error inicial es menor utilizando los modelos pre-entrenados, pero la disminución del error es mucho más lenta, por lo que al final del entrenamiento, el error en la validación es similar a los obtenidos con estos parámetros en el experimento 1.

Para el caso de la exactitud en las imágenes de validación, se ve que existe un par de diferencias en el comportamiento de esta curva entre la Figura 6.5 y 6.16, pero al final del

proceso de entrenamiento, la exactitud final también es bastante similar, por lo que a lo largo del entrenamiento, nuevamente se ve que los pesos pre-entrenados, no tienen un impacto mayor sobre la clasificación morfológica de las cabezas de espermatozoides humanos.

Entrenamiento con optimizador D y AD con pesos por clases

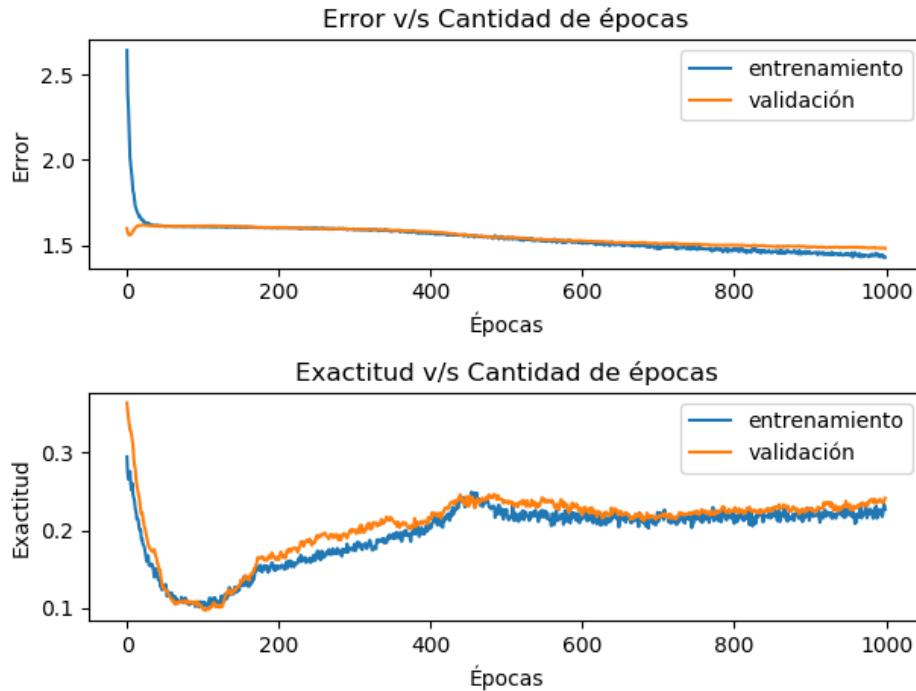


Figura 6.16: Error y exactitud en el entrenamiento de la red *RBC* pre-entrenada, con AD equitativo, usando pesos por clases. El optimizador D corresponde a *ADAM* con tasa de aprendizaje de 0,00001.

Fuente: Elaboración propia, 2020.

Los mejores resultados del afinamiento de parámetros para la red *RBC* pre-entrenada, nuevamente son con el aumento de datos dirigido y con el optimizador D (*ADAM* con tasa de aprendizaje de 0,00001). En la Figura 6.17 se ve el error y exactitud promedio en el entrenamiento y validación, utilizando estos parámetros. Cabe destacar que, utilizando la red *RBC* pre-entrenada, no se repitió el caso del experimento 1, en donde una combinación de parámetros obtuvo buenos y malos resultados en distintas iteraciones de validación cruzada.

Comparando estos resultados con los anteriores, se ve cómo el aumento de datos dirigido tiene una clara ventaja sobre en el error y exactitud en la validación, dejando en claro que la inclusión de esta estrategia de aumento de datos en este trabajo, ha sido una buena decisión al momento de entrenar los modelos.

Si se comparan estos resultados con los obtenidos en el afinamiento con los modelos desde cero (Figura 6.8), se ve cómo el error inicial en la validación en este caso es mejor, debido

al aprendizaje anterior, pero lamentablemente, la disminución del error ha sido más lenta que durante el entrenamiento desde cero, alcanzando un error en la validación de 0,98, y una exactitud del 0,60 aproximadamente, con las imágenes de validación.

Con esto, queda en claro que los mejores resultados obtenidos con la red *RBC* pre-entrenada, no superaron los mejores resultados de la red *RBC* desde cero. Se ha visto en este afinamiento de parámetros, como los pesos pre-entrenados tienen un impacto en el error en las primeras épocas de entrenamiento, lo cual tiene sentido, debido a que estos ya comienzan con un aprendizaje previo, pero a lo largo de las épocas, solo ha dificultado el aprendizaje de las formas de cabezas de espermatozoides, al tener una disminución más lenta del error, y lo mismo ocurrió con la exactitud en las imágenes de validación.

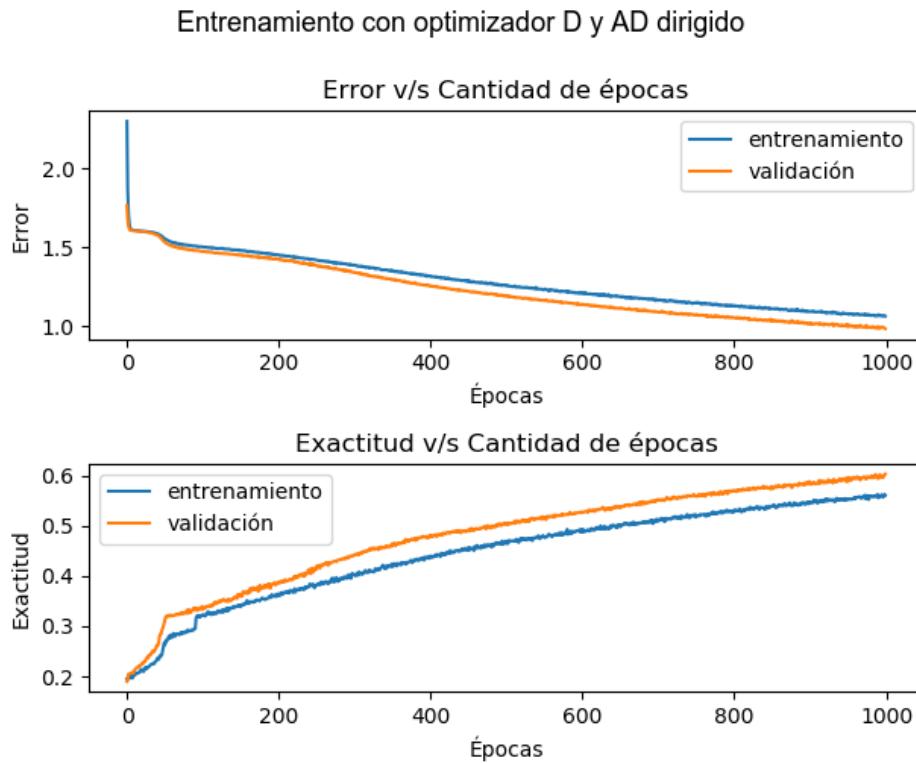


Figura 6.17: Error y exactitud en el entrenamiento de la red *RBC* pre-entrenada, con AD dirigido. El optimizador D corresponde a *ADAM* con tasa de aprendizaje de 0,00001.
Fuente: Elaboración propia, 2020.

Red convolucional ResNet50

Comenzando nuevamente con los resultados del afinamiento con aumento de datos equitativo, con bolsas de datos, el mejor optimizador para este caso corresponde al B (*SGD* con tasa de aprendizaje de 0,000001). En la Figura 6.18 se ve el promedio de error y exactitud de las

iteraciones de validación cruzada realizadas.

Como ocurrió durante el primer experimento, la red *ResNet50* tuvo los peores resultados generales entre todas las configuraciones de red, optimizadores, y estrategias de aumento de datos. Esto se repite en esta ocasión, con la red *ResNet50* pre-entrenada, en donde se ve cómo se dispara el error en la validación en las primeras épocas de entrenamiento, y la exactitud no alcanza a superar el 20% aproximadamente.

De hecho, se ve como el error en el entrenamiento, disminuyó hasta cero, y la exactitud, alcanzó el 100%, estancando el aprendizaje, y las métricas de error y exactitud en las imágenes de validación. Este problema de *overfitting*, ocurrió también durante el primer experimento, con aprendizaje desde cero, por lo que para este caso, los pesos pre-entrenados, no lograron solucionar este problema para *ResNet50*.

Entrenamiento con optimizador B y AD con bolsas de datos

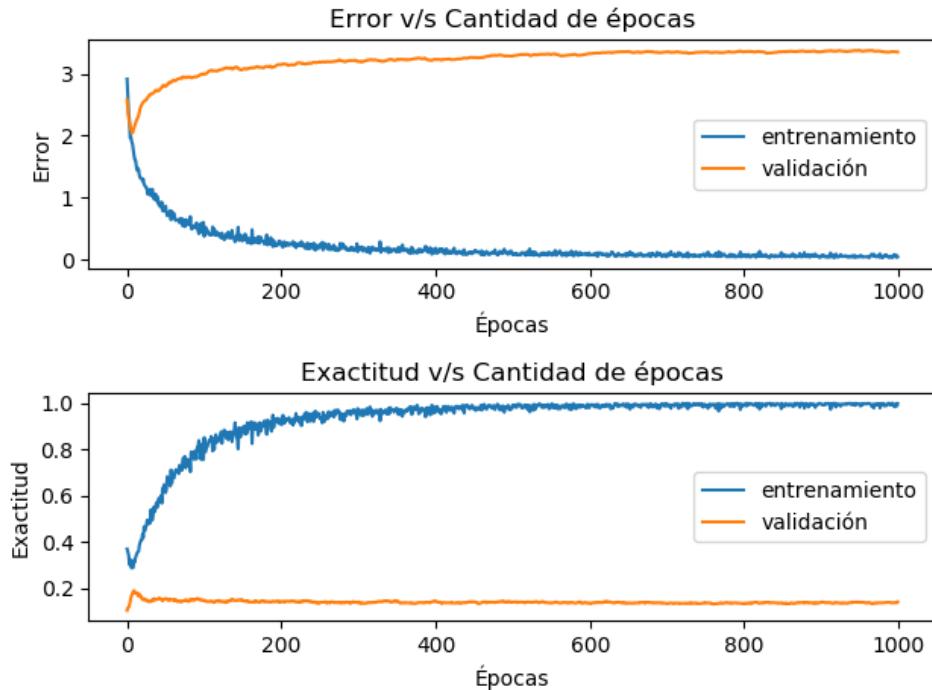


Figura 6.18: Error y exactitud en el entrenamiento de la red *ResNet50* con AD equitativo, usando bolsas de igual tamaño. El optimizador B corresponde a *SGD* con tasa de aprendizaje de 0,000001.

Fuente: Elaboración propia, 2020.

Luego, en la Figura 6.19 se ven los resultados en el entrenamiento de *ResNet50*, utilizando el aumento de datos equitativo con pesos por clases, además de usar el optimizador A (*SGD* con tasa de aprendizaje de 0,0001) para este caso.

Para esta combinación, la exactitud logró alcanzar un valor aproximado de 54%, lo

que en general es mucho mayor que el caso anterior, pero el error mínimo alcanzado en las primeras iteraciones, fue 1,55 aproximadamente, para las imágenes del conjunto de validación. Estas dos métricas, alcanzaron mejores resultados que los vistos con esta misma combinación de parámetros, en el primer experimento (Figura 6.10).

Esto quiere decir que los pesos pre-entrenados, tuvieron un efecto positivo en el entrenamiento de los modelos de *ResNet50*, en la clasificación morfológica de las cabezas de espermatocitos, pero aún así, nuevamente cae en el *overfitting*, al alcanzar un error muy cercano a 0 en las imágenes de entrenamiento, deteniendo el aprendizaje de la red nuevamente.

No hay mucho más que rescatar de estos resultados, ya que los resultados aún así son similares a los obtenidos en el experimento 1, por lo que, a pesar de tener una pequeña mejora en este caso, no es una diferencia considerable.

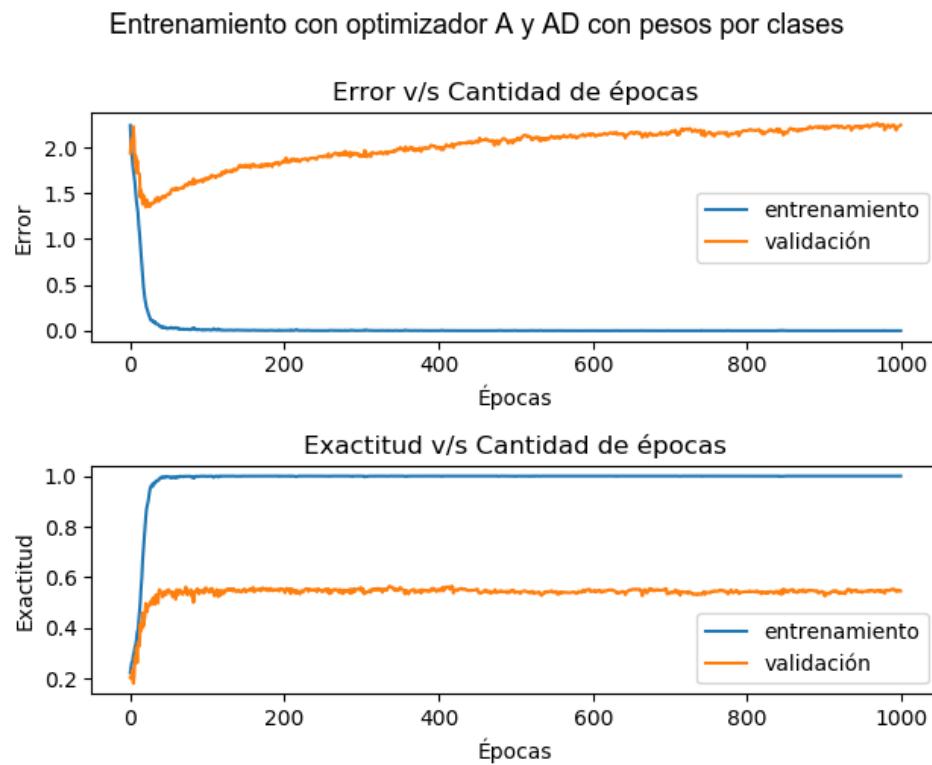


Figura 6.19: Error y exactitud en el entrenamiento de la red *ResNet50* con AD equitativo, usando pesos por clases. El optimizador A corresponde a *SGD* con tasa de aprendizaje de 0,0001.

Fuente: Elaboración propia, 2020.

Finalmente, los mejores resultados, como lo ha sido durante todo este trabajo, fueron obtenidos con el aumento de datos dirigido. En la Figura 6.20, se observa el error y exactitud promedio de *ResNet50*, a partir de las diez iteraciones de validación cruzada, y utilizando el optimizador A (*SGD* con tasa de aprendizaje de 0,0001).

Los resultados en este caso son bastante similares también en comparación al

entrenamiento de este caso, pero con los pesos aleatorios (Figura 6.11). El error mínimo y la exactitud máxima alcanzada, fue de aproximadamente 1,20 y de 58% respectivamente, y estos valores fueron alcanzados dentro de las primeras iteraciones aproximadamente.

A pesar de que la exactitud es cercana al 60%, el error es más alto que los obtenidos con la red convolucional *RBC*, tanto con entrenamiento desde cero, como con aprendizaje por transferencia. Además, sigue presentando el mismo problema de *overfitting* presente en todos los casos mostrados de esta red convolucional, estancando el aprendizaje por haber llegado a un error casi 0 con las imágenes de entrenamiento.

Además, los pesos pre-entrenados no dieron ningún beneficio para estos resultados con aumento de datos dirigido, lo que resulta inesperado, ya que este tipo de aumento de datos fue el que mejor resultados presentó en este trabajo. Se esperaba que los pesos pre-entrenados, mejoraran aunque sea un poco los resultados del entrenamiento para este caso en específico, pero no fue así.

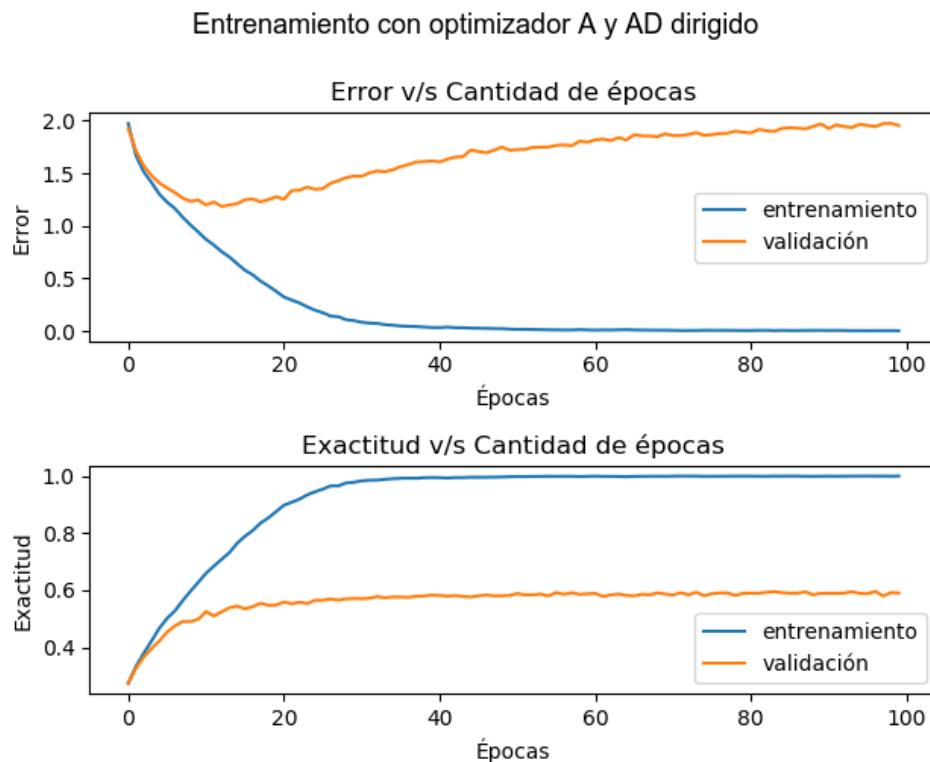


Figura 6.20: Error y exactitud en el entrenamiento de la red *ResNet50* con AD dirigido. El optimizador A corresponde a *SGD* con tasa de aprendizaje de 0,0001.

Fuente: Elaboración propia, 2020.

En cuanto a comparaciones entre *RBC* y *ResNet50*, no hay mucha diferencia entre lo observado en el experimento 1, en donde ya se ha establecido que *ResNet50* fue una red muy compleja para este problema de clasificación, al observar el *overfitting* presente en los gráficos

de entrenamiento durante el afinamiento de parámetros. Es debido a esto, que finalmente no se ha utilizado en ninguno de los dos experimentos, la red *ResNet50* para realizar la clasificación morfológica sobre el conjunto de evaluación.

Al mismo tiempo, los resultados para el aumento de datos dirigido y utilizando las redes pre-entrenadas, no mostraron ninguna mejora con respecto a los resultados obtenidos en el afinamiento del primer experimento, con aumento de datos dirigido. De hecho, para el caso de la red convolucional *RBC*, el entrenamiento se hizo más lento que utilizando pesos aleatorios, por lo que el aprendizaje previo de la morfología de las células de glóbulos rojos, no tuvo un impacto positivo sobre la clasificación morfológica de cabezas de espermatozoides humanos.

6.4.4 Resultados de la clasificación morfológica

Al igual que con el primer experimento, se realizaron diez iteraciones de validación cruzada para entrenar la red *RBC*, con aumento de datos dirigido, el optimizador D, y durante 1.000 épocas. Los diez modelos obtenidos fueron evaluados con el conjunto de evaluación separado al principio del procedimiento, obteniendo así las diez matrices de confusión correspondientes a la clasificación realizada por cada uno de los diez modelos entrenados.

En la Tabla 6.7, se ve como resultado, la matriz de confusión promedio de las diez iteraciones de validación cruzada, y al mismo tiempo, esta ha sido normalizada para establecer así las métricas y el análisis correspondiente.

A partir de los gráficos observados en el afinamiento de parámetros, los resultados obtenidos allí para este caso fueron peores que durante el primer experimento, por lo que los resultados expuestos en la matriz de confusión, también lo son.

El problema existente de los modelos, clasificando la mayoría de las clases con la clase amorfa, sigue estando presente en este caso, y en mayor cantidad, o sea, que el sesgo hacia la clase amorfa es mayor en este experimento. Esto quiere decir, que el pre-entrenamiento de los modelos con las imágenes de glóbulos rojos, no tuvo un impacto positivo, si no que todo lo contrario.

El efecto obtenido por el pre-entrenamiento de los modelos, fue un retroceso en el entrenamiento, es decir, los pesos de las neuronas estaban más alejados del mínimo global del error, que en comparación a los modelos entrenados con pesos aleatorios iniciales. Considerando además, que la red *RBC* tiene un aprendizaje muy lento sobre las características de la forma de las cabezas de espermatozoides, hizo que los modelos pre-entrenados no pudieran alcanzar las métricas obtenidas en el experimento 1.

De la misma matriz de confusión, se observa finalmente como las clasificaciones

correctas, disminuyeron en todas las clases, a excepción de la clase pequeña, pero la mejora de estos aciertos no es una diferencia importante, con respecto a la clasificación del experimento 1 en la Tabla 6.6 (la sensibilidad solo aumentó de 0,05 a 0,1), como para establecer que se trata de una mejora obtenida del pre-entrenamiento de los modelos, con el *dataset* de los glóbulos rojos.

En la Figura 6.21, se ve reflejada la curva *ROC* de la clasificación realizada en una iteración de validación cruzada. Como es de esperarse, se repite el problema del área bajo la curva de la clase amorfa, por lo que la habilidad de los modelos, de separar esta clase de las otras, no es muy buena, lo que se ve reflejado en la quinta columna de la matriz de confusión expuesta.

		Clase predicha					Total real (# imágenes)
		Normal	Cónica	Piriforme	Pequeña	Amorfa	
Clase real	Normal	0,18	0,04	0,03	0,02	0,73	1 (30)
	Cónica	0,01	0,46	0,01	0,00	0,51	1 (68)
	Piriforme	0,07	0,13	0,17	0,00	0,60	1 (23)
	Pequeña	0,03	0,07	0,01	0,1	0,79	1 (22)
	Amorfa	0,07	0,13	0,05	0,04	0,70	1 (197)

Tabla 6.7: Matriz de confusión promedio, y normalizada, de la clasificación realizada por la red *RBC* con aprendizaje por transferencia. Los resultados han sido aproximados a la centésima.

Fuente: Elaboración propia, 2020.

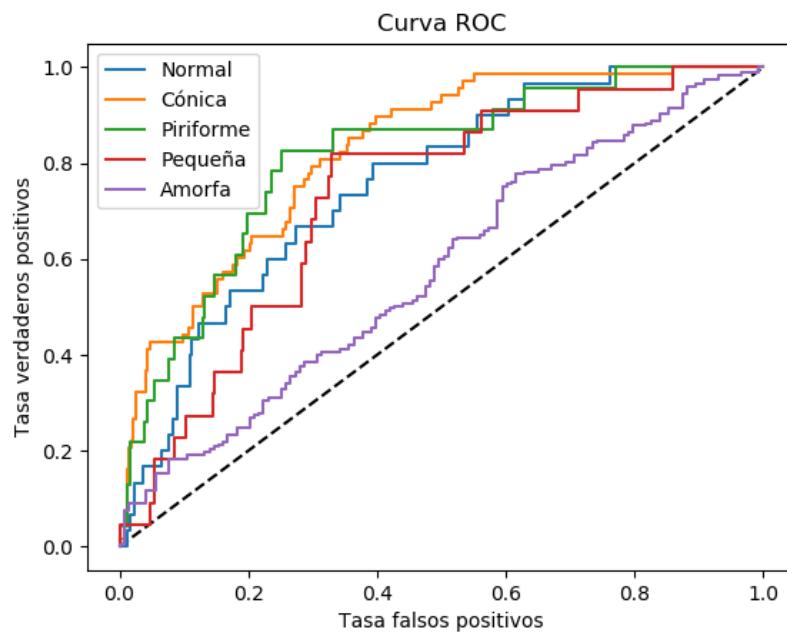


Figura 6.21: Curva *ROC* en una iteración de la clasificación morfológica de cabezas de espermatozoides, utilizando la red *RBC* con aprendizaje por transferencia.

Fuente: Elaboración propia, 2020.

De la matriz de confusión promedio, se obtienen las métricas expuestas en la Tabla 6.8, por cada una de las clases morfológicas de las cabezas de espermatozoides estudiadas en este trabajo.

En primera instancia, todas las métricas son peores que en el experimento 1, presentadas en la Tabla 6.5. Como los resultados son inferiores, nuevamente se hace presente la baja precisión y especificidad en la clasificación de la clase amorfa, y que tiene correspondencia con la quinta columna de la matriz de confusión, en donde se pudo ver que los modelos clasificaban muchas imágenes con la clase amorfa.

Por otro lado, como se mencionó, las clasificaciones correctas fueron peores en todas las clases menos la pequeña y la cónica, y esto se ve reflejada en la sensibilidad (o tasa de verdaderos positivos) en la clasificación realizada sobre estas clases. Especialmente, clases como la normal y la piriforme disminuyeron bastante la sensibilidad, por lo que los modelos identificaron menos imágenes correctas para estas clases, lo que es un retroceso con respecto al experimento 1. Además, la pequeña mejora en la sensibilidad de la clase pequeña no es importante, aumentando la sensibilidad desde 0,05 a 0,1, por lo que es difícil establecer si esto puede asociarse al pre-entrenamiento de los modelos convolucionales.

En general, todo lo que ya se ha mencionado acerca de los resultados del experimento 1, se vuelve a repetir en esta ocasión, pero con peores números. Por ejemplo, ya se ha hablado del sesgo hacia la clase amorfa, presente en los modelos de la red *RBC*, por lo que en este experimento, nuevamente se hace presente, así que el pre-entrenamiento realizado no pudo resolver este problema tampoco.

Métrica/Clase	Normal	Cónica	Piriforme	Pequeña	Amorfa	Promedio
Precisión	48.19%	54.22%	62.36%	58.21%	21.04%	48.80%
Sensibilidad	17.67%	46.03%	17.83%	10.00%	70.00%	32.30%
Especificidad	88.34%	74.82%	93.03%	95.48%	25.84%	75.50%

Tabla 6.8: Métricas de evaluación de la clasificación realizada por la red *RBC* con aprendizaje por transferencia, obtenida de la matriz de confusión normalizada. Los resultados han sido aproximados a la centésima

Fuente: Elaboración propia, 2020.

Finalmente, en la tabla 6.9, se tiene una comparación de los métricas expuestas, en comparación con las presentadas en los trabajos del estado del arte.

La sensibilidad, como se estableció en el experimento anterior, es la métrica a medir y comparar en este problema de clasificación morfológicas, pero al igual que en el experimento anterior, no se pudo superar la del estado del arte, así que el aprendizaje por transferencia aplicado en este experimento, no logró solucionar los problemas de la clasificación realizada por la red *RBC*.

De todas formas, este era un resultado que se esperaba desde la etapa de

afinamiento de los parámetros, ya que en los gráficos de error y exactitud con los conjuntos de validación, ya se mostraba un retroceso en estos resultados, en comparación a los resultados obtenidos en el experimento 1. En consecuencia, el impacto de aplicar pesos pre-entrenados de la clasificación morfológica de glóbulos rojos, en este problema de clasificación morfológica de cabezas de espermatozoides humanos, fue totalmente negativo.

Red/Sensibilidad	S(N)	S(C)	S(Pi)	S(Pe)	S(A)	Exactitud
Red convolucional <i>RBC</i> con aprendizaje por transferencia	0,18	0,46	0,18	0,10	0,70	0,32
Red convolucional <i>VGG16</i> (Riordon et al., 2019)	0,67	0,57	0,69	0,78	0,38	0,62
Red convolucional propuesta por Iqbal et al. (2020)	0,70	0,79	0,62	0,71	0,57	0,68

Tabla 6.9: Comparación de métricas finales del segundo experimento con respecto al estado del arte.

Fuente: Elaboración propia, 2020.

6.5 TERCER EXPERIMENTO: MODIFICACIÓN DE LA MEJOR RED OBTENIDA

6.5.1 Algoritmo de implementación

En el primer y segundo experimento, quedó claro que los mejores resultados hasta el momento en este trabajo, corresponden a los obtenidos con la red convolucional *RBC* desde cero, usando el optimizador *ADAM* con tasa de aprendizaje de 0,00001, y el aumento de datos dirigido. Es por esto, que se ha tomado esta arquitectura con sus modelos ya entrenados, para así realizar alguna modificación sobre esta y intentar mejorar la clasificación morfológica de las cabezas de espermatozoides humanos.

En este experimento, solo se utilizan los modelos entrenados con la combinación de parámetros mencionados en el párrafo anterior, en donde, utilizando los mismos pesos obtenidos luego de 1.000 épocas, se modifica la estructura de la red convolucional *RBC*, para luego volver a entrenar estos mismos modelos por más épocas. Estos entrenamientos se realizan de manera consecutiva, es decir, luego de entrenar un modelo de *RBC* por 1.000 épocas, se modifica la estructura de este y se vuelve a entrenar, ya que así se mantiene el mismo conjunto de imágenes para la evaluación y entrenamiento, ya que es importante medir el comportamiento de los modelos con imágenes que no han visto durante el entrenamiento.

Esta cantidad de épocas es el único valor desconocido en este experimento, ya que

la red, el optimizador, y el aumento de datos ya están pre-establecidos, por lo que los resultados son mucho más rápidos de obtener en este caso, al no tener que probar tantas combinaciones de parámetros. Por esto mismo, se entrena los modelos con estructura modificada por 1.000 épocas más, para obtener los gráficos de error y exactitud en la validación, y así determinar la cantidad de épocas necesarias (para llegar al error más cercano a 0) en la cual someter estos modelos para este experimento.

Con la cantidad de épocas establecida, finalmente se entrena los modelos modificados para poder así evaluarlos con el conjunto de imágenes de evaluación, y así obtener las métricas asociadas. Al igual que todos los experimentos hasta ahora, esto también se realiza con validación cruzada, por lo que los resultados que se muestran son el promedio resultante de las diez iteraciones realizadas.

6.5.2 Modificaciones a realizar

En la Figura 6.22, se presenta nuevamente la arquitectura convolucional de la red *RBC*. Esta vez se han marcado en color azul, tres bloques que se repiten a lo largo de esta red convolucional, los que contienen una capa convolucional, una de activación, una de agrupación, y una de regularización respectivamente.

Las modificaciones realizadas pueden ser variadas, desde eliminar estos bloques, reemplazarlos por otros, o congelarlos, es decir, que los pesos de las neuronas de estas capas no serán actualizados durante nuevas épocas de entrenamiento.

Para esta oportunidad, se ha decidido congelar las capas, según lo establecido en la documentación de la librería de *Tensorflow* (Abadi et al., 2015). Esto es una técnica que ya viene integrada en esta librería (y también en *Keras*), lo que facilita mucho más su implementación, en comparación a insertar nuevas capas al inicio o en el medio de una arquitectura convolucional.

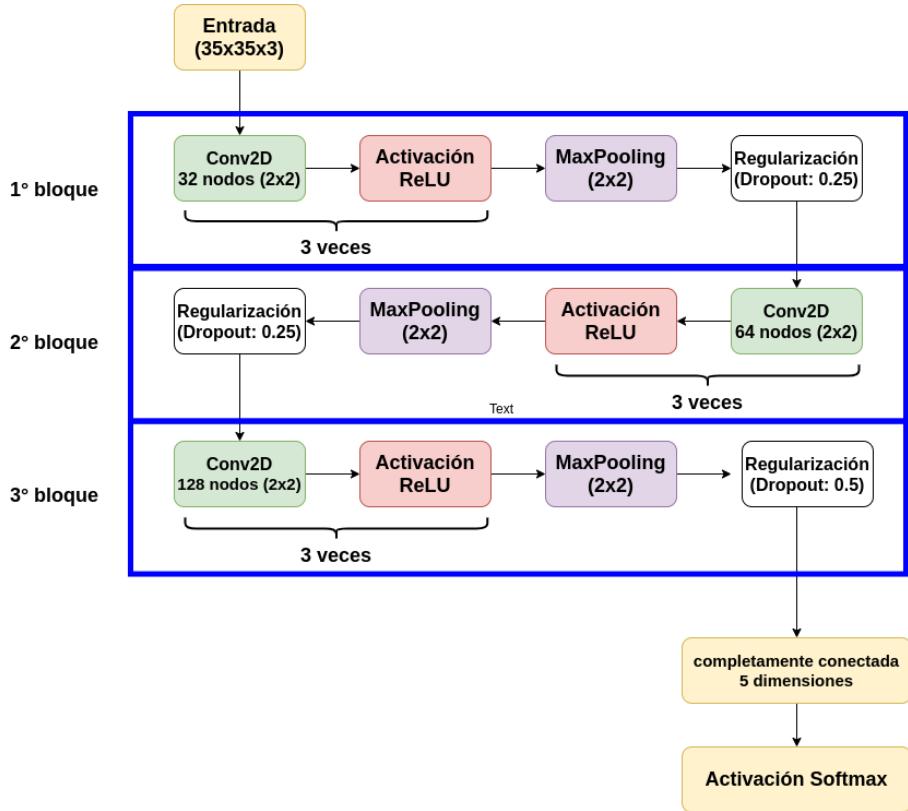


Figura 6.22: Arquitectura de la red convolucional Angluin et al. (2016) adaptada, separada en bloques. Se han marcado distintos bloques de la arquitectura, que corresponde a secciones repetidas a lo largo de esta red, para así identificar cuales son las capas que se van a congelar durante este experimento final.

Fuente: Elaboración propia, 2020.

6.5.3 Resultados congelando los primeros dos bloques de la red *RBC*

En esta sección, se entregan los resultados obtenidos al congelar las capas del primer y segundo bloque expuestos en la Figura 6.22. Esto quiere decir que, las primeras 16 capas no modificarán los pesos de sus neuronas en los nuevos entrenamientos.

En la Figura 6.23 se ve el comportamiento del error y exactitud (promedio de la validación cruzada), luego de haber entrenado por 1.000 épocas adicionales los modelos de la red *RBC* desde cero, pero con la modificación realizada en el congelamiento de los primeros dos bloques de su estructura.

En cuanto al error con las imágenes de validación, este fue disminuyendo hasta aproximadamente las 400 épocas de entrenamiento, aunque el error no logró disminuir de 0,8. Por otro lado, la exactitud en la validación no aumentó en gran cantidad, aunque existe una pequeña mejora. En este punto se puede observar de buena forma como ya el error en las imágenes de

validación, llega a un punto en que no vuelve a descender, y lo mismo ocurre con la exactitud. A partir de estos resultados, se ha escogido utilizar 400 épocas en este caso para entrenar los modelos con los primeros dos bloques congelados.

Entrenamiento con capas congeladas, optimizador D y AD dirigido

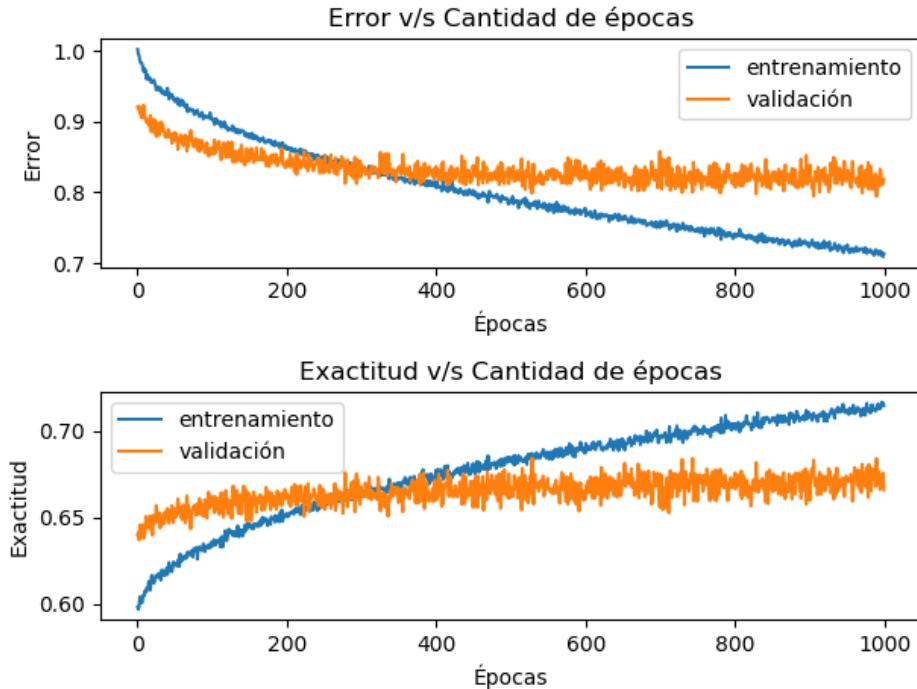


Figura 6.23: Error y exactitud en el entrenamiento de la red *RBC* con los primeros bloques congelados. El optimizador D corresponde a *ADAM* con tasa de aprendizaje de 0,00001.

Fuente: Elaboración propia, 2020.

Con la cantidad de épocas definida para este caso, se procede con el entrenamiento con esta misma cantidad, para luego evaluar los modelos con el conjunto de evaluación. En la Tabla 6.10 se tiene la matriz de confusión promedio y normalizada para la clasificación realizada, luego de entrenar los modelos por 400 épocas más pero con los primeros dos bloques de la red *RBC* congelados.

En el afinamiento de parámetros, se vio como la mejora de la exactitud y el error no fue importante, ya que estuvo estancada en un valor aproximado a un 68% de exactitud a lo largo de todo el entrenamiento, por lo que en la matriz de confusión, se ve reflejado el mismo comportamiento.

En primer lugar, las clases donde se presentaron mejoras en la cantidad de imágenes correctamente clasificadas, son en la clase normal, cónica, y en menor cantidad la piriforme y amorfa. Por otro lado, la clase pequeña no presentó mejoras en promedio en la cantidad de aciertos, por lo que a pesar de presentar una disminución en el error durante la elección de

la cantidad de épocas, los modelos nuevamente presentaron problemas para la clasificación de este tipo de morfología de cabezas de espermatozoides.

Con respecto a los resultados obtenidos en la matriz del experimento 1 (Tabla 6.4), no existe una gran diferencia entre los resultados obtenidos en esta oportunidad, a pesar de tener una pequeña mejora. De la misma forma, en la Figura 6.24 se como en la curva *ROC* de una iteración de validación cruzada, el modelo sigue teniendo el problema para separar la clase amorfa de las otras, el que se ve reflejado por el área bajo la curva de esta clase.

A partir de esto, los resultados de este trabajo demuestran que, al igual que en el estado del arte (Chang et al., 2017b; Riordon et al., 2019; Iqbal et al., 2020), la clase amorfa ha sido un problema constante al momento de tener buenos resultados en la clasificación morfológica de las cabezas de espermatozoides humanos.

		Clase predicha					Total real (# imágenes)
		Normal	Cónica	Piriforme	Pequeña	Amorfa	
Clase real	Normal	0,44	0,00	0,03	0,00	0,53	1 (30)
	Cónica	0,00	0,48	0,02	0,00	0,50	1 (68)
	Piriforme	0,00	0,00	0,41	0,00	0,59	1 (23)
	Pequeña	0,22	0,05	0,00	0,05	0,68	1 (22)
	Amorfa	0,09	0,14	0,05	0,00	0,72	1 (197)

Tabla 6.10: Matriz de confusión promedio, y normalizada, de la clasificación realizada por la red *RBC* con sus primeros dos bloques congelados. Los resultados han sido aproximados a la centésima.

Fuente: Elaboración propia, 2020.

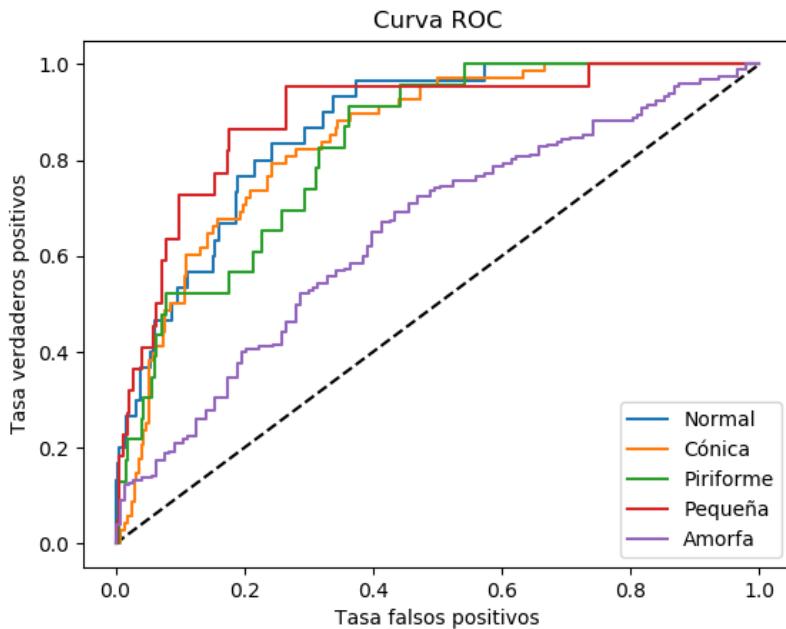


Figura 6.24: Curva *ROC* en una iteración de la clasificación morfológica de cabezas de espermatozoides, utilizando la red *RBC* con sus primeros dos bloques congelados.

Fuente: Elaboración propia, 2020.

Siguiendo la misma linea de los experimentos anteriores, se obtienen las métricas en la Tabla 6.11, los cuales han sido obtenidas de la matriz de confusión anterior.

La clase amorfa sigue teniendo los problemas de precisión y sensibilidad en la clasificación, algo que era previsto por la columna de predicción amorfa, y la curva *ROC* obtenida. Al mismo tiempo, no hubieron mejoras en la clasificación realizada por la clase pequeña, por lo que la red *RBC*, después de 1.400 épocas de entrenamiento, los modelos siguen teniendo este problema con esta clase, lo que da entender finalmente que esta red no puede generalizar las características de esta clase, al menos con las imágenes del *dataset* estudiado.

Cómo los aciertos mejoraron para la clase normal y cónica, también se ve esto reflejado en la sensibilidad de la clasificación de estas clases. Lamentablemente, no es una diferencia considerable, al menos no lo suficiente como para alcanzar los resultados del estado del arte.

Métrica/Clase	Normal	Cónica	Piriforme	Pequeña	Amorfa	Promedio
Precisión	57.81%	73.55%	79.16%	88.47%	23.71%	64.54%
Sensibilidad	43.67%	47.65%	40.87%	5.45%	71.57%	41.84%
Especificidad	83.86%	90.41%	93.99%	99.65%	37.41%	81.06%

Tabla 6.11: Métricas de evaluación de la clasificación realizada por la red *RBC* con sus primeros dos bloques congelados, obtenida de la matriz de confusión normalizada. Los resultados han sido aproximados a la centésima.

Fuente: Elaboración propia, 2020.

Finalmente, en la Tabla 6.12, se ven los resultados finales de la clasificación morfológica obtenida en este experimento, el cual, se ve que a pesar de haber mejorado un poco la sensibilidad promedio (exactitud), los resultados siguen estando muy por debajo de los obtenidos en el estado del arte.

Incluso después de tantas épocas de entrenamiento, la red *RBC* no logró mejorar las métricas con respecto al estado del arte. De hecho, el entrenamiento de esta red fue tan lento, que tomó 1.400 épocas solamente para lograr los resultados expuestos en la Tabla 6.11 y 6.12, demostrando nuevamente, como este problema de clasificación morfológica, es muy complicado para la red *RBC* escogida a partir de la literatura.

Red/Sensibilidad	S(N)	S(C)	S(Pi)	S(Pe)	S(A)	Exactitud
Red convolucional <i>RBC</i>	0,44	0,48	0,41	0,05	0,72	0,42
Red convolucional <i>VGG16</i> (Riordon et al., 2019)	0,67	0,57	0,69	0,78	0,38	0,62
Red convolucional propuesta por Iqbal et al. (2020)	0,70	0,79	0,62	0,71	0,57	0,68

Tabla 6.12: Comparación de métricas finales del experimento tres con respecto al estado del arte.

Fuente: Elaboración propia, 2020.

6.6 RESUMEN

Para asegurar la robustez de las métricas de evaluación, todos los experimentos se realizan con la técnica de validación cruzada, para así probar el entrenamiento y evaluación de la red con distintos grupos de imágenes. Finalmente, los resultados finales de los experimentos se obtienen del promedio de las diez iteraciones de validación cruzada.

En este trabajo, se realizaron tres experimentos distintos, los que corresponden a entrenar las redes *RBC* y *ResNet50* desde cero, con aprendizaje por transferencia, y luego modificando la arquitectura de la red con los mejores resultados. Luego del afinamiento de parámetros, quedó demostrado que para todos los experimentos, la red *ResNet50* es demasiado compleja para este problema de clasificación, al presentar *overfitting* en la mayoría de los gráficos de error en la validación obtenidos. Es debido a esto, que en todos los experimentos, finalmente se utilizó la red *RBC* para la clasificación morfológica sobre el conjunto de evaluación.

Al mismo tiempo, en el afinamiento de parámetros se vio como el aumento de datos dirigido demostró ser una estrategia favorable el entrenamiento, obteniendo los mejores resultados de error y exactitud en las imágenes de validación.

Luego, los resultados de la red *RBC* entrenada desde cero, con pesos aleatorios,

tuvo mejores resultados en la clasificación morfológica, en comparación a la misma red, pero con pesos pre-entrenados con las imágenes de glóbulos rojos. Esto quiere decir, que la información previa de las características de glóbulos rojos, no sirvieron para mejorar la precisión en la clasificación morfológica de las cabezas de espermatozoides humanos.

Es por esto, que se utilizó la red *RBC* entrenada desde cero, para modificar su arquitectura, e intentar mejorar aún más la clasificación morfológica. En este caso, se congelaron algunas capas que componen la red convolucional *RBC*, y se tomaron los modelos ya entrenados del experimento 1, para seguir entrenándolos con esta arquitectura modificada.

Finalmente, los resultados solo mejoraron una pequeña cantidad, especialmente en las clasificaciones correctas, por lo que la sensibilidad alcanzada en este último experimento (0,42 aproximadamente), a pesar de ser los mejores obtenidos en este trabajo, no alcanzaron a superar los del estado del arte de Riordon et al. (2019) e Iqbal et al. (2020).

CAPÍTULO 7. CONCLUSIONES

A modo de conclusión, se presentan las dificultades encontradas a lo largo de este trabajo, acompañado de los objetivos específicos y generales logrados. Por último, se plantean trabajos a futuro a considerar sobre el problema de la clasificación morfológica de cabezas de espermatozoides humanos.

7.1 DIFICULTADES ENCONTRADAS

Comenzando con dificultades del propio desarrollo de la investigación, uno de los primeros inconveniente fue la utilización del *dataset* público, **SCIAN-MorphoSpermGS**. Estas imágenes se encontraban en formato de imagen *.tif*, el cual presentó un error bien particular al momento de trabajar con la lectura de las imágenes.

Por algún motivo, al leer las imágenes utilizando *Python*, ocurría un efecto sobre el contraste de las imágenes al momento de convertirlas a un arreglo de enteros (los que representaban los colores de los píxeles de las imágenes). Al imprimir por pantalla las imágenes antes de convertirse en arreglos, se mostraban las imágenes tal cual como están publicadas, en cuanto al color y contraste, pero justo después de convertirlas en arreglo de enteros, para ser trabajadas con el aumento de datos, se volvían a mostrar por pantalla, y se notaba un cambio en el contraste de las imágenes.

Aún no queda claro el origen de esta modificación, ya que, al probar el código fuente, se determinó que ocurría específicamente al pasar las imágenes a los arreglos de enteros. Se intentó resolver este problema utilizando distintos tipos de datos en los arreglos, e incluso utilizando librerías específicas en la lectura de archivos *.tiff*.

Finalmente, utilizando la librería de *Matplotlib*, se logró convertir las imágenes en arreglos, sin alterar el contraste de esto. Como el aumento de datos debía ser realizado sobre las imágenes originales sin alteración, se tuvo que utilizar *Matplotlib* para guardar las imágenes aumentadas, en vez de las funciones que ya venían incorporadas dentro de la librería de *Keras*, solucionando este problema al fin y al cabo.

Por otro lado, desde un principio se planteó, que las redes convolucionales de *RBC* y *ResNet50*, iban a ser comparadas en igualdad de condición, por lo que se hizo un análisis en paralelo de ambas redes, en cuanto a su comportamiento con los distintos tipos de optimizadores y estrategias de aumento de datos. Esto causo un problema con la red *RBC*, la cual, desde un principio, al calcular la cantidad de épocas a entrenar esta red, se notó el lento aprendizaje sobre las características de la morfología de las cabezas de espermatozoides. Esto hizo que se probara

con una gran cantidad de épocas durante el entrenamiento, lo que fue bastante lento, además de que arrastró a la red *ResNet50* a probar con la misma cantidad de épocas.

De hecho, el entrenamiento de la red *ResNet50* fue tan lento, que para el aumento de datos dirigido, se tuvo que optar de manera obligada a entrenar por solo 100 épocas, siendo un caso excepcional durante el afinamiento de parámetros. A modo de ejemplo, entrenar *ResNet50* por 100 épocas, con el *dataset* con aumento de datos dirigido, tomó aproximadamente 12 horas, por cada uno de los cuatro optimizadores utilizados.

Siguiendo el hilo sobre la lentitud del entrenamiento, este procedimiento tuvo que realizarse en servidores dedicados, con una tarjeta de vídeo especializada para a este tipo de trabajo de análisis de imágenes. El problema fue que, a mitad del desarrollo, el servidor dedicado dejó de funcionar, lo que atraso los entrenamientos de las redes convolucionales. Considerando todos los parámetros probados para ambas redes, perder el servidor fue un golpe en el desarrollo de la investigación.

Luego de esto, una vez recuperado el servidor, se tuvo que compartir con otros investigadores que también necesitaban de estas máquinas, por lo que, aún después de recuperar el acceso a los servidores dedicados, no se pudo utilizar el 100% del tiempo para esta investigación.

Ahora bien, desde una perspectiva más personal del investigador, se esperaba que los resultados fueran mejores a los del estado del arte, especialmente luego de haber escogido estas dos redes convolucionales, de otros trabajos anteriores, con tan buenos resultados. A medida que se avanzó la investigación, y comenzaron a aparecer los gráficos y métricas de evaluación, la realidad mostró que ninguna de las dos redes fueron aptas para este problema de clasificación, lo que afectó mucho en la motivación en la realización de este trabajo en los últimos meses, y a esto se le suma, que durante el mismo desarrollo de este trabajo, se publicó una investigación sobre este mismo problema, que si logró mejorar el estado del arte utilizando estas herramientas de redes convolucionales.

Por último, y como un factor más externo, otra dificultad ha sido el mismo año 2020, el cual, con el contexto de la pandemia, ha afectado también la motivación y de esta investigación. Al mismo tiempo, se presentaron problemas personales más allá de la pandemia, durante los meses de abril y mayo, los que, aún a la fecha, han sido difíciles de superar.

7.1.1 Objetivos específicos

- Entrenar dos modelos de *deep learning* para clasificar morfológicamente cabezas de espermatozoides, usando el *dataset* público **SCIAN-MorphoSpermGS**, y evaluar su

rendimiento con métricas similares a las utilizadas en el estado del arte.

Las dos redes convolucionales utilizadas en esta investigación, fueron elegidas por haberse utilizado en trabajos pasados que tenían relación con el problema actual de clasificación morfológica de espermatozoides humanos (Angluin et al., 2016; Riordon et al., 2019). No solo eso, si no que ambas redes ya se habían utilizado en la clasificación multiclase, facilitando la adaptación de estas redes convolucionales, a la clasificación morfológica de las cinco cabezas de espermatozoides estudiadas en este trabajo.

Por otro lado, para entrenar las redes convolucionales, se debió también de limpiar y extender el *dataset* de trabajo, en donde se utilizaron tres estrategias distintas de aumento de datos, no solo para aumentar la cantidad de imágenes a utilizar en el entrenamiento, si no que también para balancear las clases. Estas estrategias, de igual manera, se escogieron debido a la documentación de trabajos anteriores y la documentación de las librerías utilizadas (Iqbal et al., 2020; Abadi et al., 2015).

Además, la misma documentación de las librerías de *Tensorflow* respaldaron el hecho de poder congelar capas de las redes, luego de haberlas entrenado, para así mejorar los resultados de la clasificación, por lo que esta alternativa también fue utilizada durante esta investigación.

Por lo tanto, considerando los puntos mencionados sobre la elección, adaptación y preparación del *dataset*, es que el entrenamiento de las redes convolucionales *RBC* y *ResNet50* se llevó a cabo de manera satisfactoria.

- Distinguir los mejores parámetros de entrenamiento (número de épocas, tamaño del lote, optimizador y tasa de aprendizaje) para la arquitectura original de los dos modelos de *deep learning* elegidos.

El entrenamiento de las redes convolucionales, debían tener una cantidad de épocas de entrenamiento adecuadas, para que los modelos no tuvieran problemas de *underfitting* o *overfitting* al realizar la clasificación morfológica de las cabezas de espermatozoides, con el conjunto de imágenes de evaluación. Entonces, cada configuración de parámetros distinta debió ser evaluada, no solo para obtener cuál era la que tenía mejores resultados, si no también para identificar la cantidad de épocas en la que finalmente deben entrenarse los modelos.

Dentro de la bibliografía se encontró dos principales optimizadores, los que corresponden a *SGD* con *Nesterov* y *ADAM*. Sin embargo, no fue posible establecer de manera inicial cual de estos era el que tendría mejores resultados para el problema de la clasificación morfológica de espermatozoides humanos.

Debido a esto, es que se evaluaron ambos optimizadores, utilizando un pequeño conjunto de imágenes en el entrenamiento, llamado conjunto de validación. Esta evaluación,

además, fue realizada para ambas redes convolucionales elegidas, ya que las redes deben ser comparadas en igualdad de condición.

El tamaño del lote (cantidad de imágenes) estuvo definido por las tres estrategias de aumento de datos utilizadas, por lo que se tuvieron tres tamaños de lote distintos, que se evaluaron en conjunto con los optimizadores, combinando cada optimizador con cada estrategia de aumento de datos utilizada.

No solo eso, si no que también se probaron dos tasas de aprendizajes distintas, aumentando más la combinatoria anterior de parámetros a utilizar para entrenar tanto la red *RBC* como la red *ResNet50*.

Finalmente, este proceso se repitió en el experimento 1 y 2, llegando a haber evaluado 48 configuraciones para los entrenamientos. En el experimento 3, solo se volvió a evaluar un optimizador y estrategia de aumento de datos, agregando una configuración más al total de configuraciones de parámetros evaluados.

Considerando la elección de ambos optimizadores, y todas las configuraciones evaluadas, es que este objetivo se considera completo.

- Aplicar y determinar el impacto de aprendizaje por transferencia en el problema de clasificación morfológica de cabezas de espermatozoides, usando los mismos dos modelos elegidos.

Para poder aplicar esta técnica, era necesario encontrar otro problema similar al de clasificación morfológica de cabezas de espermatozoides humanos, y que además, tuviera a disposición del conjunto de imágenes utilizados en ese trabajo.

Considerando que ya se ha establecido una gran similitud con el trabajo de clasificación morfológica de glóbulos rojos (Angluin et al., 2016) en objetivos anteriores, se decidió también, utilizar las imágenes de este trabajo de glóbulos rojos, para pre-entrenar las redes convolucionales elegidas en esta investigación.

En el repositorio del trabajo de Angluin et al. (2016), se encontró disponible las más de 3.000 imágenes ya clasificadas, que corresponde al conjunto de imágenes ya aumentado, según los números que se exponen en el mismo trabajo. Además, fue necesario pre-entrenar las redes con las mejores configuraciones, lo que no fue un problema, debido a que este mismo trabajo ya documentó que optimizador utilizó para entrenar la red *RBC*, por lo que solo se hicieron pruebas con la red *ResNet50* en este aspecto.

Traspasar los pesos pre-entrenados hacia los modelos de *RBC* y *ResNet50* fue un poco más difícil, debido a que la versión utilizada de *Tensorflow*, no tenía implementada las librerías para reemplazar pesos de las neuronas de las capas convolucionales, pero ignorando aquellas que no tuvieran el mismo tamaño. Hay que recordar que en este trabajo se adaptaron

las redes *RBC* y *ResNet50* para clasificar en 5 clases distintas, por lo que la dimensión de las últimas capas eran distintas entre el pre-entrenamiento y el problema actual, por lo que los pesos pre-entrenados no eran compatible en estas capas.

De todas formas, los pesos pre-entrenados estaban guardados en arreglos de la librería de *Numpy*, por lo que la solución a esto fue recorrer los arreglos y eliminar de manera manual, aquellos pesos que no coincidieran con los modelos a utilizar en la clasificación morfológica.

Finalmente, se pudo apreciar diferencias entre las métricas obtenidas de la clasificación morfológica de cabezas de espermatozoides humanos, entre el entrenamiento desde cero y el entrenamiento con aprendizaje por transferencia, por lo que este objetivo se considera exitoso.

7.1.2 Objetivo general

En este trabajo, se utilizaron dos redes convolucionales, como herramientas de *deep learning*, para la clasificación morfológica de cabezas de espermatozoides humanos. Hablando únicamente de los resultados obtenidos en este trabajo, las métricas en la clasificación morfológica, no fueron capaces de mejorar la precisión en la clasificación. Las mejores métricas de este trabajo, fueron obtenidas luego de entrenar la red convolucional *RBC* desde cero, para luego aplicar congelamiento de alguna de sus capas convolucionales.

En comparación a la clasificación realizada por la red convolucional *VGG16* (Riordon et al., 2019), la clasificación morfológica realizada por la red convolucional *RBC* fue un poco más exacta, lo que quiere decir, que la clasificación realizada por los modelos, tiende a ser más cercana a las clases reales de las imágenes, pero donde más problemas existe, es en las clasificaciones correctas, las que fueron bajas todas las clases menos la amorfá, afectando a las métricas como la sensibilidad y la precisión.

Este hecho ocurrió, principalmente, a que existió un sesgo en la clasificación, hacia la clase amorfá, por lo que muchas clasificaciones incorrectas, se hicieron sobre esta clase, obteniendo una clasificación mucho más dispersa y disminuyendo la precisión en la clasificación. En la misma Tabla 6.12, se pudo observar la alta sensibilidad de la clase amorfá, pero este es resultado del mismo sesgo en la clasificación, lo cual es evidenciado en las curvas *ROC* presentadas, demostrando la poca capacidad de los modelos de determinar la clase amorfá de las cabezas de espermatozoides.

En consecuencia, la baja cantidad de clasificaciones correctas, hizo que los resultados de la clasificación no superaran las métricas de la clasificación realizada por *VGG16*,

especialmente en la sensibilidad, en donde todas obtuvieron resultados más altos que los obtenidos por la red *RBC*, a excepción de la clase amorfa, pero esto es por el problema mencionado en el párrafo anterior.

Hay que dejar bien en claro también, que cuando se comenzó a investigar el problema de la clasificación morfológica de cabezas de espermatozoides humanos, durante el año 2019, el estado del arte más reciente en ese entonces correspondían a los obtenidos por Riordon et al. (2019), y así lo fue hasta el mes de mayo del 2020. El último trabajo publicado de Iqbal et al. (2020) presentó nueva información que, de haberse publicado con anterioridad, se habrían tomado otras decisiones, como por ejemplo, haber modificado la red convolucional de este trabajo, en vez de utilizar la red *ResNet50*. Sin embargo, de todas formas se utilizaron técnicas aplicadas en este trabajo, como lo fue la estrategia de aumento de datos dirigido.

Teniendo en cuenta lo anterior, los resultados que se buscaba superar siempre fueron los de Riordon et al. (2019), pero lamentablemente, a partir de los resultados publicados en el Capítulo 6, las redes convolucionales escogidas no tienen forma de competir contra la red *VGG16*, por lo que finalmente, las redes *RBC* y *ResNet50* no fueron capaces de superar al estado del arte, en la clasificación morfológica de cabezas de espermatozoides humanos.

7.2 TRABAJO A FUTURO

Considerando el conjunto de imágenes actual de **SCIAN-MorphoSpermGS**, ha quedado demostrado los problemas sobre la generalización de las características por parte de las redes convolucionales *RBC* y *ResNet50*. A pesar de que aún es posible probar con otros parámetros, como distintas tasas de aprendizaje o congelar otras capas de estas redes, con tantas opciones que ya se tomaron, ninguna fue capaz de mejorar la clasificación, de forma de mejorar el estado del arte (Iqbal et al., 2020).

A pesar de esto, es importante mencionar que no se pueden descartar las redes convolucionales para la clasificación morfológica de espermatozoides humanos. Que dos instancias de esta herramienta no hayan obtenido los resultados esperados, no significa que todas no puedan mejorar el estado del arte.

Desde la publicación del trabajo clasificación con *support vector machine* (Chang et al., 2017b), se han presentado ya dos trabajos, además de este mismo, sobre redes convolucionales para este problema de clasificación. Especialmente el último trabajo (Iqbal et al., 2020), el que implementó una arquitectura propia para resolver este problema, si logró mejorar el estado del arte, que para ese entonces, correspondían a la clasificación realizada con la red convolucional *VGG16* (Riordon et al., 2019).

Para próximos trabajos, puede tomarse como punto de partida la red convolucional propuesta por Iqbal et al. (2020), intentando modificar la arquitectura y ver si se pueden mejorar aún más la clasificación morfológica de espermatozoides humanos. Esta modificación contemplaría el congelamiento de algunas capas convolucionales, la agregación de nuevas capas o incluso, quitar algunas de las capas propuestas por este trabajo.

Esta modificación tiene un límite, ya que de este trabajo, se ha aprendido que las redes convolucionales profundas tienen malos resultados con el *dataset* de **SCIAN-MorphoSpermGS**, por lo que hay que tener cuidado de no convertir la red de Iqbal et al. (2020) en una red convolucional muy profunda, ya que podría ocurrir el mismo problema visto con los resultados obtenidos con la red *ResNet50*, en donde se encontró con el *overfitting* durante los entrenamientos.

Si bien es cierto que el código fuente correspondiente a la construcción de la red convolucional no se encuentra disponible, este trabajo es libre de descargar en internet, además de que este documento contiene en detalle la construcción de la red convolucional propuesta, por lo que es posible comenzar desde este artículo para modificar la red convolucional.

Por otro lado, a pesar de los malos resultados obtenidos durante el entrenamiento con la red convolucional *ResNet50*, también es posible implementar otra red convolucional residual, pero de menor complejidad. En otra librería de *machine learning* para *Python*, llamada *Pytorch*, se encuentran implementadas redes convolucionales residuales de 18 y 34 capas convolucionales, por lo que también es una alternativa, realizar la clasificación morfológica con estas arquitecturas, a pesar de que *ResNet50* haya demostrado malos resultados durante esta investigación.

Por otro lado, también es posible volver a realizar aprendizaje por transferencia, ya sea en la red *RBC* o la red propuesta por Iqbal et al. (2020), ya que a pesar de que las imágenes de glóbulos rojos no hayan ayudado en este problema de clasificación morfológica de cabezas de espermatozoides, no quiere decir que haya otro conjunto de imágenes que ayude a mejorar los resultados en la clasificación.

Al contrario, también es posible plantear nuevas arquitecturas para resolver este problema, aunque este es un proceso que requiere de más desarrollo e investigación, que en comparación a partir de una arquitectura ya planteada y probada.

Algo que se ha aprendido en este trabajo, es el problema mencionado de las redes complejas, por lo que, sea cual sea el camino a seguir, ya sea trabajar con la red de Iqbal et al. (2020), la red *RBC*, una red residual, o cualquier otra instancia de red convolucional, conviene que no sea una red muy profunda, ya que es muy probable que, trabajando con este *dataset*, vuelva a ocurrir problemas de *overfitting*, que fue lo que se vio al trabajar con *ResNet50*.

Finalmente, para futuros trabajos, se considera recomendable trabajar con un

enfoque más pequeño en las herramientas a utilizar, debido a que, en este trabajo se consideraron dos redes convolucionales distintas a utilizar, pero la cantidad de configuraciones posibles hizo que la cantidad de entrenamientos se extendieran demasiado, por lo que sería mejor centrarse en una sola red convolucional.

7.3 OBSERVACIONES FINALES

A pesar de los malos resultados obtenidos en la clasificación morfológica de cabezas de espermatozoides humanos, todo el desarrollo e investigación expuesto fue realizado mano a mano con el profesor guía de este trabajo, en conjunto además, con la bibliografía utilizada como base de las decisiones tomadas en el desarrollo de los experimentos.

Es por esto que, las redes *RBC* y *ResNet50* siempre fueron una buena elección a utilizar en este problema, y se probaron muchas configuraciones y técnicas posibles, para encontrar los mejores resultados que ambas redes pudieran entregar para este trabajo, terminando un trabajo de amplio desarrollo e investigación, superando incluso los tiempos propuestos en la propuesta de este trabajo.

GLOSARIO

Andrología: La andrología es la disciplina que se ocupa del diagnóstico y tratamiento de las enfermedades del sistema reproductivo masculino (Merriam-Webster, s.f).

ground truth: Según Cardoso et al. (2014), es el conjunto de medidas conocidas como las mas exactas en el momento. El nombre *ground truth* proviene de la información obtenida en una observación directa de lo que se estudió (*in situ*), dejando de lado la inferencia.

gold standard: En Medicina, el rendimiento de todo test diagnóstico se basa en su comparación con un *gold standard* (estándar de oro, patrón de oro, patrón de referencia). El estándar de oro es la técnica diagnóstica que define la presencia de la condición con la máxima certeza conocida (Salech et al., 2008).

Área de Interés (*region of interest*): Un área de interés es una porción de los datos de entrada, la cual contiene la información relevante a estudiar para un problema en específico. En este caso, el área de interés corresponde a las partes de las imágenes que contienen la cabezas de los espermatozoides humanos.

Tensorflow y Keras: Tensorflow es una librería dedicada al *machine learning*, y comprende módulos de alto y bajo nivel. Keras, es una librería dedicada a redes neuronales, y que ofrece módulos de alto nivel. En este caso, Keras trabaja como una envoltura sobre Tensorflow, el cual utiliza los módulos de redes neuronales que son más fáciles de implementar e importar, utilizando por debajo los paquetes de Tensorflow (Choudhury, 2019).

Aumento de datos: En el documento se le menciona con las siglas AD, y corresponde a la técnica utilizada para extender el *dataset* trabajado, al generar nuevos datos a partir de los originales.

REFERENCIAS BIBLIOGRÁFICAS

- Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., & Zheng, X. (2015). TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org/
URL <https://www.tensorflow.org/>
- Angluin, D., Olson, E., & Torres, R. (2016). Red Blood Cell Classification using Convolutional Networks. *Yale University*, (p. 21).
URL https://zoo.cs.yale.edu/classes/cs490/16-17a/tandon.devansh.dt373/final_paper.pdf
- Bakker, B. (2002). Reinforcement learning with long short-term memory. *Neural Information Processing Systems*.
- Barría, L., & Ramírez, M. (Eds.) (2014). *Ciencias Plan Común: Biología*. Chile: Cpech.
URL http://intranet-alumno.cepech.cl/wp-content/uploads/2013/04/BIOLOGIA-PLAN-COMUN_2014.pdf
- Bre, F., Gimenez, J., & Fachinotti, V. (2017). Prediction of wind pressure coefficients on building surfaces using Artificial Neural Networks. *Energy and Buildings*, 158.
- Brownlee, J. (2017). A Gentle Introduction to Transfer Learning for Deep Learning.
URL <https://machinelearningmastery.com/transfer-learning-for-deep-learning/>
- Brownlee, J. (2019a). How to Choose Loss Functions When Training Deep Learning Neural Networks. Library Catalog: machinelearningmastery.com.
URL <https://machinelearningmastery.com/how-to-choose-loss-functions-when-training-deep-learning-neural-networks/>
- Brownlee, J. (2019b). Loss and Loss Functions for Training Deep Learning Neural Networks. Library Catalog: machinelearningmastery.com.
URL <https://machinelearningmastery.com/loss-and-loss-functions-for-training-deep-learning-neural-networks/>
- Brownlee, J. (2019c). What is Deep Learning? Library Catalog: machinelearningmastery.com.
URL <https://machinelearningmastery.com/what-is-deep-learning/>
- Cardoso, J. R., Pereira, L. M., Iversen, M. D., & Ramos, A. L. (2014). What is gold standard and what is ground truth? *Dental Press Journal of Orthodontics*, 19(5), 27–30.
URL <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4296658/>
- Chang, V. (2015). *Segmentation and classification of human sperm heads towards morphological sperm analysis*. Ph.D. thesis, Universidad de Chile.
URL <http://repositorio.uchile.cl/handle/2250/136250>
- Chang, V., García, A., Hitschfeld, N., & Härtel, S. (2017a). Gold-standard for computer-assisted morphological sperm analysis. *Computers in Biology and Medicine*, 83, 143–150.
- Chang, V., Heutte, L., Petitjean, C., Härtel, S., & Hitschfeld, N. (2017b). Automatic classification of human sperm head morphology. *Computers in Biology and Medicine*, 84, 205–216.
- Chollet, F., et al. (2015). Keras. <https://keras.io>.

- Choudhury, A. (2019). TensorFlow vs Keras: Which One Should You Choose. Library Catalog: analyticsindiamag.com.
 URL <https://analyticsindiamag.com/tensorflow-vs-keras-which-one-should-you-choose/>
- Dean, J. (2016). Large-scale deep learning for building intelligent computer systems. (pp. 1–1).
- Deng, L., & Yu, D. (2014). Deep Learning: Methods and Applications. *Foundations and Trends® in Signal Processing*, 7(3–4), 197–387. Publisher: Now Publishers, Inc.
 URL <https://nowpublishers.com/article/Details/SIG-039>
- Eggert-Kruse, W., Schwarz, H., Rohr, G., Demirakca, T., Tilgen, W., & Runnebaum, B. (1996). Sperm morphology assessment using strict criteria and male fertility under in-vivo conditions of conception. *Human Reproduction (Oxford, England)*, 11(1), 139–146.
- garychl (2018). Applications of Reinforcement Learning in Real World.
 URL <https://towardsdatascience.com/applications-of-reinforcement-learning-in-real-world-1a94955bcd12>
- González, J., Ávalos, A., Vargas, K., & Herrera, J. (2018). *Recolección y manipulación seminal in vitro*. Universidad Autónoma Metropolitana, 1st ed.
 URL http://www.casadelibrosabiertos.uam.mx/contenido/contenido/Libroelectronico/recolección_manipulacion.pdf
- Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. MIT Press. Google-Books-ID: omivDQAAQBAJ.
- Guanga, A. (2018). Machine Learning: Bias VS. Variance. Library Catalog: becominghuman.ai.
 URL <https://becominghuman.ai/machine-learning-bias-vs-variance-641f924e6c57>
- Harrison, O. (2019). Machine Learning Basics with the K-Nearest Neighbors Algorithm.
 URL <https://towardsdatascience.com/machine-learning-basics-with-the-k-nearest-neighbors-algorithm-6a6e71d01761>
- He, K., Zhang, X., Ren, S., & Sun, J. (2015). Deep Residual Learning for Image Recognition. *arXiv:1512.03385 [cs]*. ArXiv: 1512.03385.
 URL <http://arxiv.org/abs/1512.03385>
- Iqbal, I., Mustafa, G., & Ma, J. (2020). Deep Learning-Based Morphological Classification of Human Sperm Heads. *Diagnostics*, 10(5), 325. Number: 5 Publisher: Multidisciplinary Digital Publishing Institute.
 URL <https://www.mdpi.com/2075-4418/10/5/325>
- Javadi, S., & Mirroshandel, S. A. (2019). A novel deep learning method for automatic assessment of human sperm images. *Computers in Biology and Medicine*, 109, 182–194.
- Khetan, V. (2019). Bias in machine learning algorithms. Library Catalog: towardsdatascience.com.
 URL <https://towardsdatascience.com/bias-in-machine-learning-algorithms-f36ddc2514c0>
- Kingma, D., & Ba, J. (2014). Adam: A method for stochastic optimization. *International Conference on Learning Representations*.
- Kohavi, R., & Wolpert, D. (1997). Bias Plus Variance Decomposition for Zero-One Loss Functions.
- Koza, J. R., Bennett, F. H., Andre, D., & Keane, M. A. (1996). Automated Design of Both the Topology and Sizing of Analog Electrical Circuits Using Genetic Programming. In J. S. Gero, & F. Sudweeks (Eds.) *Artificial Intelligence in Design '96*, (pp. 151–170). Dordrecht: Springer Netherlands.
 URL https://doi.org/10.1007/978-94-009-0279-4_9

- Kruger, T. F., Acosta, A. A., Simmons, K. F., Swanson, R. J., Matta, J. F., & Oehninger, S. (1988). Predictive value of abnormal sperm morphology in in vitro fertilization. *Fertility and Sterility*, 49(1), 112–117.
- Lee, R. K.-K., Hou, J.-W., Ho, H.-Y., Hwu, Y.-M., Lin, M.-H., Tsai, Y.-C., & Su, J.-T. (2002). Sperm morphology analysis using strict criteria as a prognostic factor in intrauterine insemination. *International Journal of Andrology*, 25(5), 277–280.
- Mahmood, A., Giraldo, A., Bennamoun, M., An, S., Sohel, F., Boussaid, F., Hovey, R., Fisher, R., & Kendrick, G. (2020). Automatic hierarchical classification of kelps using deep residual features. *Sensors*, 20, 447.
- Menkveld, R. (2013). Sperm morphology assessment using strict (tygerberg) criteria. *Methods in Molecular Biology (Clifton, N.J.)*, 927, 39–50.
- Merriam-Webster (s.f). Definition of ANDROLOGY.
 URL <https://www.merriam-webster.com/dictionary/andrology>
- Mishra, S. (2017). Unsupervised Learning and Data Clustering. Library Catalog: towardsdatascience.com.
 URL <https://towardsdatascience.com/unsupervised-learning-and-data-clustering-eeecb78b422a>
- Mitchell, T. M. (1980). *The need for biases in learning generalizations*. Department of Computer Science, Laboratory for Computer Science Research, Rutgers Univ.
- Narkhede, S. (2019). Understanding AUC - ROC Curve. Library Catalog: towardsdatascience.com.
 URL <https://towardsdatascience.com/understanding-auc-roc-curve-68b2303cc9c5>
- OMS (2010a). WHO | Mother or nothing: the agony of infertility.
 URL <https://www.who.int/bulletin/volumes/88/12/10-011210/en/>
- OMS (Ed.) (2010b). *WHO laboratory manual for the examination and processing of human semen*. Geneva: World Health Organization, 5th ed. OCLC: ocn646393549.
- O'Shea, K., & Nash, R. (2015). An Introduction to Convolutional Neural Networks. *arXiv:1511.08458 [cs]*. ArXiv: 1511.08458.
 URL <http://arxiv.org/abs/1511.08458>
- Pan, S. J., & Yang, Q. (2010). A Survey on Transfer Learning. *IEEE Transactions on Knowledge and Data Engineering*.
- Parikh, R., Mathai, A., Parikh, S., Chandra Sekhar, G., & Thomas, R. (2008). Understanding and using sensitivity, specificity and predictive values. *Indian Journal of Ophthalmology*, 56(1), 45–50.
 URL <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2636062/>
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12, 2825–2830.
- Perez, L., & Wang, J. (2017). The Effectiveness of Data Augmentation in Image Classification using Deep Learning. *arXiv:1712.04621 [cs]*. ArXiv: 1712.04621.
 URL <http://arxiv.org/abs/1712.04621>
- Reed, R., & MarksII, R. J. (1999). *Neural Smithing: Supervised Learning in Feedforward Artificial Neural Networks*. MIT Press. Google-Books-ID: 7C4TDgAAQBAJ.

- Riordon, J., McCallum, C., & Sinton, D. (2019). Deep learning for the classification of human sperm. *Computers in Biology and Medicine*, 111, 103342.
URL <http://www.sciencedirect.com/science/article/pii/S0010482519302112>
- Rocca, J. (2019). A gentle journey from linear regression to neural networks. Library Catalog: towardsdatascience.com.
URL <https://towardsdatascience.com/a-gentle-journey-from-linear-regression-to-neural-networks-68881590760e>
- Ruder, S. (2017). An overview of gradient descent optimization algorithms. *arXiv:1609.04747 [cs]*. ArXiv: 1609.04747.
URL <http://arxiv.org/abs/1609.04747>
- Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., & Fei-Fei, L. (2015). ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3), 211–252.
- Salech, F., Mery, V., Larrondo, F., & Rada, G. (2008). Estudios que evalúan un test diagnóstico: interpretando sus resultados. *Revista médica de Chile*, 136(9), 1208–1208.
URL https://scielo.conicyt.cl/scielo.php?script=sci_abstract&pid=S0034-98872008000900018&lng=es&nrm=iso&tlang=es
- Samuel, A. L. (1959). Some Studies in Machine Learning Using the Game of Checkers. *IBM Journal of Research and Development*, 3(3), 210–229. Conference Name: IBM Journal of Research and Development.
- Sarin, S. (2019). Exploring Data Augmentation with Keras and TensorFlow.
URL <https://towardsdatascience.com/exploring-image-data-augmentation-with-keras-and-tensorflow-a8162d89b844>
- Schmidhuber, J. (2015). Deep learning in neural networks: An overview. *Neural Networks*, 61, 85–117.
URL <http://www.sciencedirect.com/science/article/pii/S0893608014002135>
- Shaikh, R. (2018). Cross Validation Explained: Evaluating estimator performance. Library Catalog: towardsdatascience.com.
URL <https://towardsdatascience.com/cross-validation-explained-evaluating-estimator-performance-e51e5430ff85>
- Shen, C., Song, M., & Zhao, Q. (2012). Learning high-level concepts by training a deep network on eye fixations. In *in: Deep Learning and Unsupervised Feature Learning Workshop, In Conduction with NIPS, Lake Tahoe*.
- Siegel, C., Daily, J., & Vishnu, A. (2016). Adaptive Neuron Apoptosis for Accelerating Deep Learning on Large Scale Systems.
- Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv 1409.1556*.
- Srinivasan, A. V. (2019). Stochastic Gradient Descent — Clearly Explained !! Library Catalog: towardsdatascience.com.
URL <https://towardsdatascience.com/stochastic-gradient-descent-clearly-explained-53d239905d31>
- Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(56), 1929–1958.
URL <http://jmlr.org/papers/v15/srivastava14a.html>

- Surma, G. (2019). Image Classifier - Cats vs Dogs. Library Catalog: towardsdatascience.com.
 URL <https://towardsdatascience.com/image-classifier-cats-vs-dogs-with-convolutional-neural-networks-cnns-and-google-colabs-4e9af21ae7a8>
- Uniqtech (2020). Understand the Softmax Function in Minutes. Library Catalog: medium.com.
 URL <https://medium.com/data-science-bootcamp/understand-the-softmax-function-in-minutes-f3a59641e86d>
- Weisstein, E. W. (2020). Convolution. Library Catalog: mathworld.wolfram.com Publisher: Wolfram Research, Inc.
 URL <https://mathworld.wolfram.com/Convolution.html>
- Willmott, C. J., & Matsuura, K. (2005). Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance. *Climate Research*, 30(1), 79–82.
 URL <https://www.int-res.com/abstracts/cr/v30/n1/p79-82/>
- Wilson, A. C., Roelofs, R., Stern, M., Srebro, N., & Recht, B. (2018). The Marginal Value of Adaptive Gradient Methods in Machine Learning. *arXiv:1705.08292 [cs, stat]*. ArXiv: 1705.08292.
 URL <http://arxiv.org/abs/1705.08292>
- Wu, Y., Liu, L., Pu, C., Cao, W., Sahin, S., Wei, W., & Zhang, Q. (2019). A Comparative Measurement Study of Deep Learning as a Service Framework. *IEEE Transactions on Services Computing*, (pp. 1–1). ArXiv: 1810.12210.
 URL <http://arxiv.org/abs/1810.12210>
- Xu, M., Papageorgiou, D. P., Abidi, S. Z., Dao, M., Zhao, H., & Karniadakis, G. E. (2017). A deep convolutional neural network for classification of red blood cells in sickle cell anemia. *PLOS Computational Biology*, 13(10), e1005746.
 URL <https://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1005746>
- Zegers-Hochschild, F., Adamson, G., de Mouzon, J., Ishihara, O., Mansour, R., Nygren, K., Sullivan, E., & Vanderpoel, S. (2009). International Committee for Monitoring Assisted Reproductive Technology (ICMART) and the World Health Organization (WHO) revised glossary of ART terminology, 2009. *Fertility and Sterility*, 92(5), 1520–1524.
 URL <https://linkinghub.elsevier.com/retrieve/pii/S0015028209036887>

APÉNDICE A. GRÁFICOS DE AFINAMIENTO DE DATOS

A continuación se presentan resultados promedios obtenidos en el afinamiento de parámetros de la red *RBC*, con los optimizadores A, B y C, ya que fueron los que no alcanzaron los mejores resultados. Estos están dividido tanto para el experimento 1 como el experimento dos, y siguen el mismo formato de los gráficos expuestos en el Capítulo 6. Durante el experimento 3 no se probaron más optimizadores por lo que no existen gráficos de afinamiento de estos en este caso.

A.1 EXPERIMENTO 1

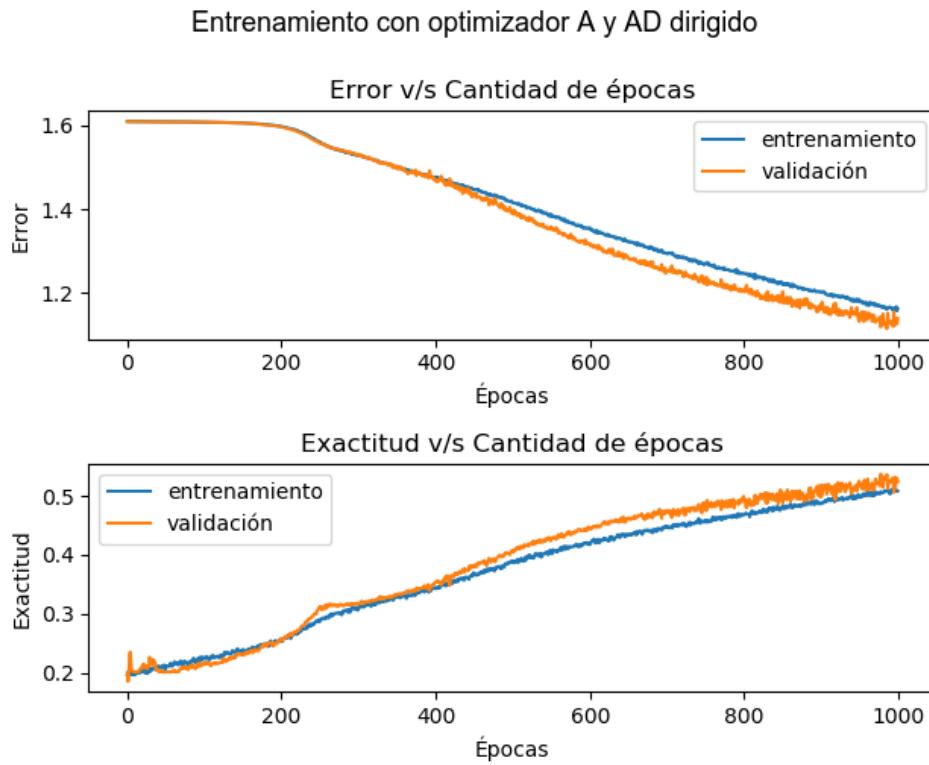


Figura A.1: Error y exactitud en el entrenamiento de la red *RBC* con AD dirigido en el experimento 1. El optimizador A corresponde a *SGD* con tasa de aprendizaje de 0,0001.

Fuente: Elaboración propia, 2020.

Entrenamiento con optimizador B y AD dirigido

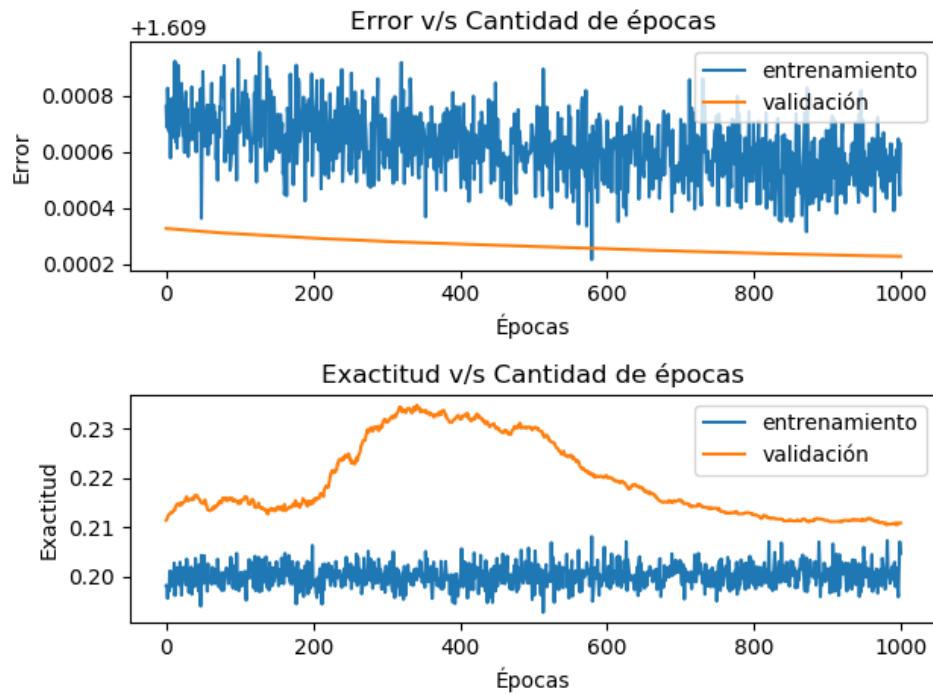


Figura A.2: Error y exactitud en el entrenamiento de la red *RBC* con AD dirigido en el experimento 1. El optimizador B corresponde a *SGD* con tasa de aprendizaje de 0,000001.

Fuente: Elaboración propia, 2020.

Entrenamiento con optimizador C y AD dirigido

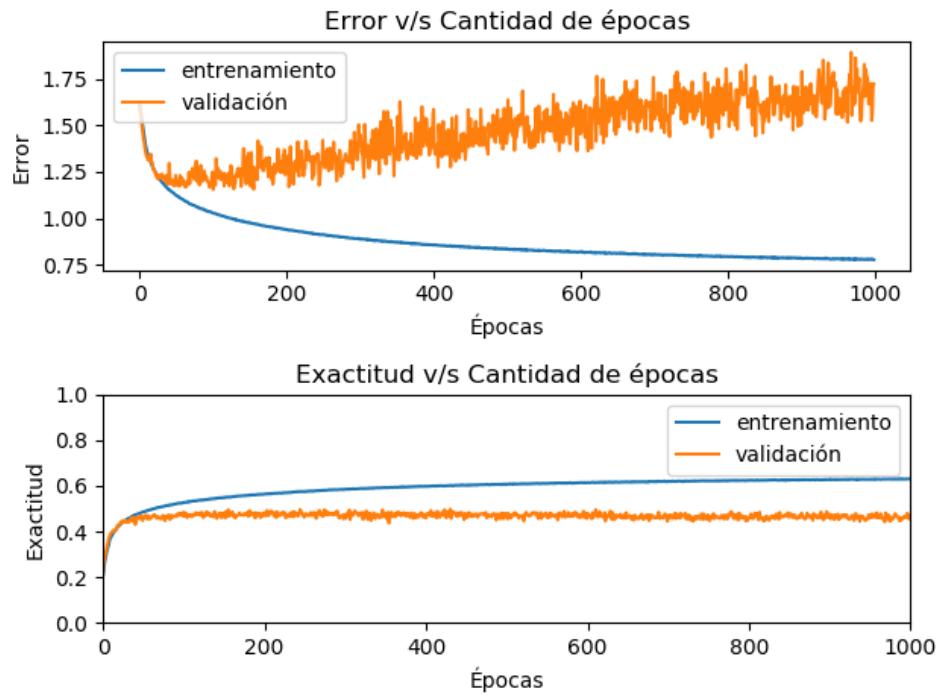


Figura A.3: Error y exactitud en el entrenamiento de la red *RBC* con AD dirigido en el experimento 1. El optimizador C corresponde a *ADAM* con tasa de aprendizaje de 0,001.

Fuente: Elaboración propia, 2020.

A.2 EXPERIMENTO 2

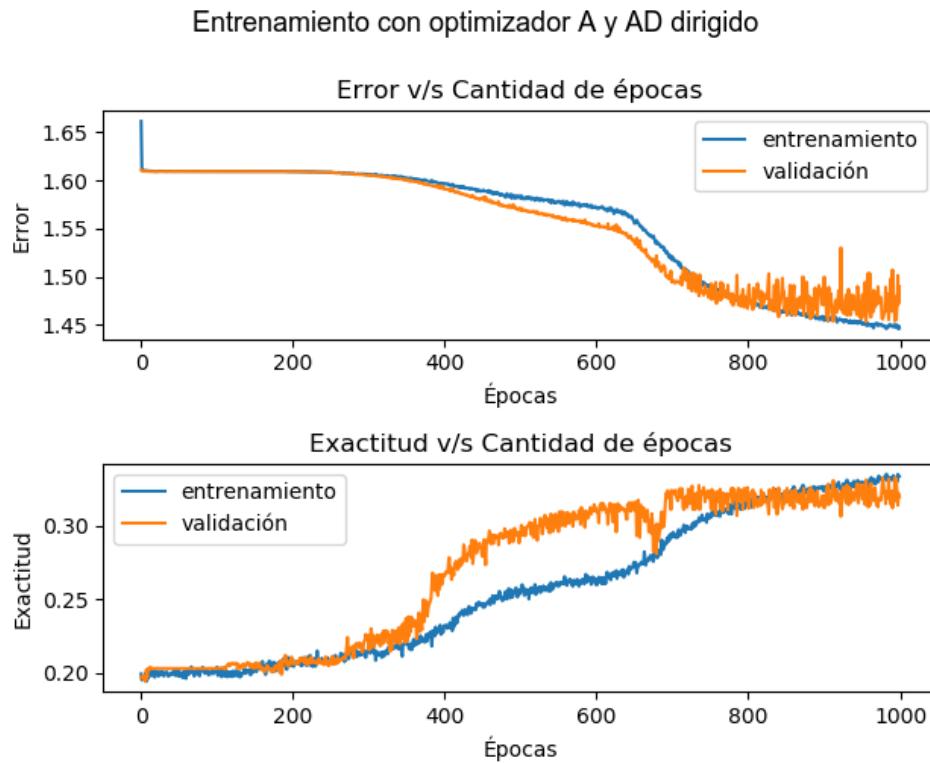


Figura A.4: Error y exactitud en el entrenamiento de la red *RBC* con AD dirigido en el experimento 2. El optimizador A corresponde a *SGD* con tasa de aprendizaje de 0,0001.

Fuente: Elaboración propia, 2020.

Entrenamiento con optimizador B y AD dirigido

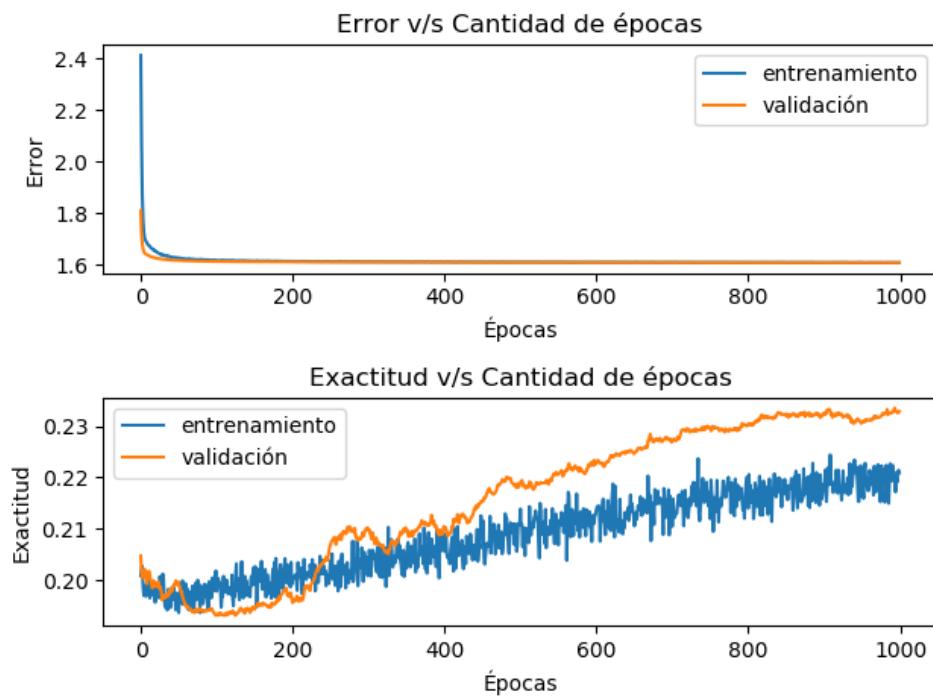


Figura A.5: Error y exactitud en el entrenamiento de la red *RBC* con AD dirigido en el experimento 2. El optimizador B corresponde a *SGD* con tasa de aprendizaje de 0,000001.

Fuente: Elaboración propia, 2020.

Entrenamiento con optimizador C y AD dirigido

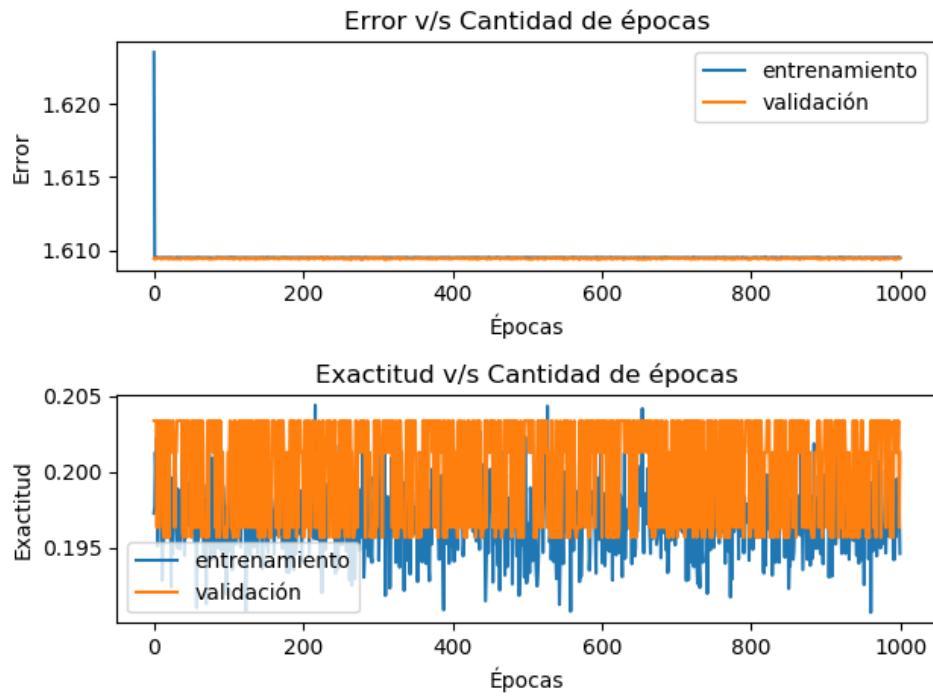


Figura A.6: Error y exactitud en el entrenamiento de la red *RBC* con AD dirigido en el experimento 2. El optimizador C corresponde a *ADAM* con tasa de aprendizaje de 0,001.
Fuente: Elaboración propia, 2020.