ISSN: 2237-826X
ARTÍCULO | ARTIGO

Deep learning aplicado para la detección de hemorragias y tumores cerebrales

Deep learning aplicado para a detecção de hemorragias e tumores cerebrais

Mauricio Fernando Hidalgo Barrientos[1], Bryan Isaac Hayes Ortiz[2], Ignacio Delgadillo Vera[3], Manuel Goyo Escalona[4]

[1] *Universidad Finis Terrae, Providencia, Chile. ORCID: https://orcid.org/0000-0003-3191-3673*

[2] *Universidad Finis Terrae, Providencia, Chile. ORCID: https://orcid.org/0000-0003-2357-4800*

[3] *Universidad Técnica Federico Santa María (USM), Valparaíso, Chile. ORCID: https://orcid.org/0000-0003-0406-9833*

[3] *Universidad Técnica Federico Santa María (USM), Valparaíso, Chile. ORCID: https://orcid.org/0000-0001-7501-7315*

Autor para correspondência/*Mail to:* Mauricio Fernando Hidalgo Barrientos, mhidalgo@uft.cl

Recebido/*Submitted*: 31 de maio de 2021; **Aceito/*Approved*:** 27 de outubro de 2021

Resumen

Introducción: un problema que afecta a la salud en Chile se refiere a las patologías cerebrales, toma de exámenes y el alto tiempo de espera para la obtención delos resultados (retrasando el diagnóstico y tratamiento). Actualmente, los exámenes se envían al extranjero para ser procesados y el tiempo de espera juega en contra del paciente. Dada esta realidad, nuestro documento propone un modelo de deep learning para la predicción de imágenes cerebrales que permita obtener un diagnóstico previo, pero no definitivo, en virtud de disminuir el tiempo del proceso y, de ser necesario, priorizar a los pacientes cuya vida estaría potencialmente en riesgo. **Método:** el desarrolloutilizó un enfoque RAD iterativo y las imágenes se recogieron de Kaggle. Adicionalmente, se redimensiona el dataset para normalizar el tamaño y generamos nuevas imágenes utilizando "data augmentation". Las imágenes fueron procesadas en redes convolucionales, indagando en distintas configuraciones para la red, su optimizador y la función de activación, hasta llegar a un modelo que consideramos razonable. **Resultados:** con el modelo definitivo, los resultados superan el 80% de precisión en las predicciones y descubrimos que separar patologías (hemorragias y tumores) fue crucial para este resultado. **Conclusiones:** hemos logrado una herramienta de diagnóstico previo, pero se debe continuar la investigación en virtud de aumentar la precisión. Un próximo paso considera ampliar el dataset con imágenes de otras fuentes y separar el modelo para analizar patologías de forma independiente. Motivamos a seguir investigando ya que este tipo de apoyo puede contribuir a salvar vidas.

Palavras-chave: Deep Learning; Redes convolucionales; Aplicaciones prácticas; Bioinformática; Tumores cerebrales.

Resumo

Introdução: *um dos problemas que afeta a saúde no Chile refere-se às patologias cerebrais, à realização de exames e à longa espera pela obtenção dos resultados (atrasos no diagnóstico e tratamento). Atualmente, os exames são enviados ao exterior para serem processados e o tempo de espera joga contra o paciente. Dada a realidade, nosso documento propõe um modelo de deep learning para predição de imagens cerebrais que permite obter um diagnóstico prévio, mas não definitivo, em virtude de diminuir o tempo do processo e, se necessário, priorizar pacientes cuja vida estaria potencialmente em risco.* ***Método:*** *o desenvolvimento usou uma abordagem RAD iterativa e as imagens foram coletadas do Kaggle. Além disso, o conjunto de dados é redimensionado para normalizar o tamanho e geramos novas imagens usando "data augmentation". As imagens foram processadas em redes convolucionais, investigando diferentes configurações da rede, seu otimizador e a função de ativação, até chegarmos a um modelo que consideramos razoável.* ***Resultados:*** *Com o modelo definitivo os resultados ultrapassam 80% de acertos nas previsões e descobrimos que separar patologias (hemorragias e tumores) foi fundamental para este resultado.* ***Conclusão:*** *: alcançamos uma ferramenta de diagnóstico prévio, mas a pesquisa deve ser continuada em virtude do aumento da precisão. Uma próxima etapa é expandir o conjunto de dados com imagens de outras fontes e separar o modelo para analisar patologias de forma independente. Encorajamos mais investigação, uma vez que este tipo de apoio pode ajudar a salvar vidas.*

Keywords: *Deep Learning; Redes convolucionais; Aplicações práticas; Bioinformática; Tumores cerebrais.*

INTRODUCCIÓN

La identificación de patologías cerebrales es un tema que se puede resumir en "vida o muerte". La realidad en Chile nos sitúa en un promedio de 5.5 radiólogos por cada 100 mil habitantessegún lo indicado por la Sociedad Chilena de Radiología[1] y los servicios de salud no cuentan con un sistema eficiente para la entrega de resultados de exámenes imagenológicos.Por esta razón, las imágenes son enviadas a otros países, como Brasil, traduciendo el proceso en largos tiempos de espera para la obtención del informe radiológico. Lamentablemente, esto no ayuda cuando hay pacientes que, sin saberlo, están "contra el reloj" y, siendo concretos, un diagnóstico preliminar oportuno podría marcar la diferencia entre un tratamiento oportuno o un desenlace fatal.

Considerando este contexto, el presente trabajo muestra una solución informática para la automatización del reconocimiento de las patologías en las imágenes clínicas. Si bien no sería un diagnóstico definitivo al no ser

[1] https://www.sochradi.cl/

firmado por un médico, permitiría mitigar los tiempos asociados al proceso, ayudando a discriminar casos que pudieran ser de mayor riesgo para el paciente y permitir, de dicha manera, una priorización de una evaluación profesional. Para lograr lo anterior, hemos visto en el deep learning[2], al igual que otros autores, una herramienta eficaz para la predicción y detección de diversas condiciones en los humanos. Por ejemplo, Ari e Hanbay (2018) exploraron la clasificación de imágenes cerebrales para la identificación de tumores cancerígenos utilizando ELM-LRF y, en un ámbito relacionado, Cao et al. (2017) trabajaron en la detección de tumores mamarios en imágenes de ultrasonido mediante aprendizaje profundo.

Sin perjuicio de lo anterior, el trabajo en esta materia es aún incipiente por lo que nuestra investigación es novedosa al incorporar simultáneamente dos patologías asociadas a problemas cerebrales identificando cerebros saludables, presencia de hemorragias o tumores en una resonancia magnética de encéfalo. Además, en lo referente a la especialidad, las aplicaciones prácticas del Deep Learning son un foco interesante para desarrollo de soluciones bioinformáticas enfocadas en agilizar los procesos y a la contribución para salvar vidas.

CONCEPTOS PRELIMINARES

Deep Learning

Deep learning es un subcampo del aprendizaje automático (*machine learning*), el cual se concentra en el aprendizaje mediante capas sucesivas de representación. El concepto de profundidad (*depth*), no hace referencia a la profundidad del entendimiento obtenido, sino a la cantidad de capas o niveles de representación que contribuyen a la red (Chollet, 2017). Adicionalmente, el Deep learning está basado en el algoritmo de las redes neuronales, las cuales, bajo distintas arquitecturas, han demostrado tener un buen rendimiento en campos como visión por computadora, reconocimiento de voz, procesamiento de lenguaje natural, entre otros. Una definición formal fue introducida por McCulloch e Pitts (1943) aunque utilizaremos ladefinición más moderna mostrada por Petersen and Voigtlaender (2018) y que se expone en la Definición 1.

Definição 1

Sean d,s,L $\in \mathbb{N}$. Una red neuronal $\phi condimensindeentradad, dimensindesalidasyLcapasesunasecuencia$

$$\phi = ((W[1], b[1]), (W[2], b[2]), \cdots, (W[L], b[L]))$$

donde $n_0 = d, n_L = s, n_1, \ldots, n_(L-1) \in \mathbb{N}$ y cada $W^(l])$ es una matriz $n_l \times n_(l-1) y b^(l]) \in \mathrm{R}^(n_l)$.

Entonces, si ϕ es una red neuronal definida como se expresa anteriormente, K$\subset \mathbb{R}^d$ y σ: $\mathbb{R} \to \mathbb{R}$ es arbitraria, entonces se define la realización de con función de activación σ como la función $\mathbb{R}\sigma\phi$: $K \to \mathbb{R}^s$ como $\mathbb{R}\sigma(\phi)(x) = x^(L])$, donde $x^(L])$ sigue el siguiente esquema:

$x[0]x,$
$x[l]\sigma(W[l]x[l-1] + b[l]),$
$x[L]W[L]x[L-1] + b[L]$

donde $\sigma((v) = (\sigma((v_1), \sigma((v_2), \cdots, \sigma((v_m)) paracadav = (v_1, v_(2), \ldots, v_m)$

En consecuencia, y aunque se utilizó una función arbitraria (conocida como función de activación), popularmente se utilizan las funciones adoptadas del escritode Petersen, Raslan, e Voigtlaender (2020)como son ReLU, parametric ReLU y Sigmoid, entre otras. En pocas palabras, el adjetivo "deep" en deep learning se refiere al uso de múltiples capas, esto es cuando el número L en la representación anterior es grande, donde cada una de estas extrae una representación de alto nivel de las características intrínsecas en la data. Para hacer esto, la transformación implementada por una capa es parametrizada por sus pesos (valores W y b en la ecuación) donde entendemos como aprendizaje el hecho de obtener un conjunto de pesos para todas las capas de la red, de tal manera que dicha red podrá mapear ejemplos (entradas) a sus "targets" asociados. Para poder realizar esta tarea, se debe medir la diferencia entre las salidas obtenidas y las esperadas, dicha diferencia se alcanza mediante la función de pérdida o "loss function" en inglés. La función de pérdida toma las predicciones de la red y los targets esperados y calcula un valor de distancia entre éstos. Finalmente, este valor es utilizado para ajustar los pesos de las capas con el fin de disminuir el valor conseguido por la función de pérdida, a través de un optimizador, el cual implementa un algoritmo de retroalimentación o "*backpropagation*" en inglés. Un esquema resumido se presenta en la Figura 1.

[2]Deep learning es un subconjunto de machine learning (que a su vez es parte de la inteligencia artificial) donde las redes neuronales, algoritmos inspirados en cómo funciona el cerebro humano, aprenden de grandes cantidades de datos: https://www.ibm.com/cl-es/cloud/deep-learning.

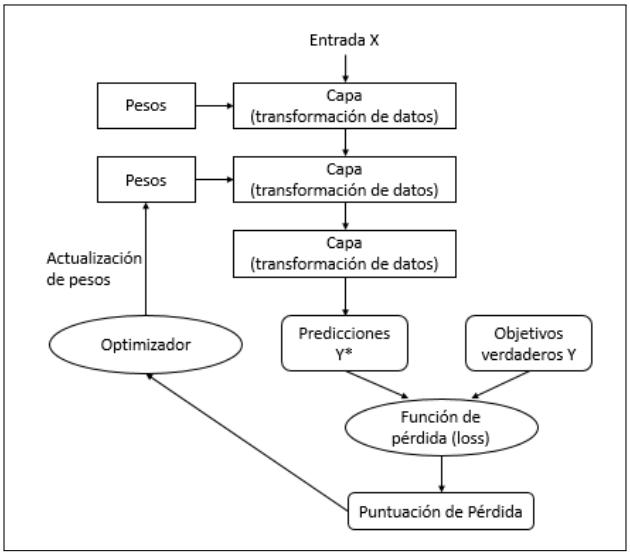

Figura 1. Retroalimentación en una red neuronal.
Fonte: Recuperado de "The "deep" in deep learning." de F.Chollet (2017).
Nota: La puntuación de pérdida se utiliza como señal de retroalimentación para ajustar los pesos.

Redes convolucionales

Las redes convolucionales fueron introducidas por Lecun (1989) y se refieren a un tipo de red neuronal diseñada para trabajar con entradas estructuradas en cuadrículas, las cuales tienen una fuerte dependencia espacial en la región local de la cuadrícula Aggarwal (2018). Un ejemplo de este tipo de entradas son las imágenes bidimensionales. En relación a sus capas, las redes convolucionales poseen, por lo general, 3 tipos de capas: Capa de convolución, capa de pooling y capa densa (O'Shea & Nash, 2015). La capa de convolución, según lo indicado en (Schmidhuber, 2014), genera una nueva matriz a partir de la obtenida por la entrada, mediante la aplicación de un filtro con el que va calculando los valores para cada cuadrante. El filtro puede poseer distintos valores dependiendo de qué proceso se le quiere realizar a la entrada, pero este filtro siempre va a crear un solo mapeo característico sin importar la profundidad. El resultado alcanzado en la salida es el de los productos puntos entre la entrada y el filtro. Un ejemplo (resumen) para una convolución entre una entrada de 7x7x1 y un filtro de 3x3x1 con un paso (*stride*) de 1 se presenta en la Figura 2.

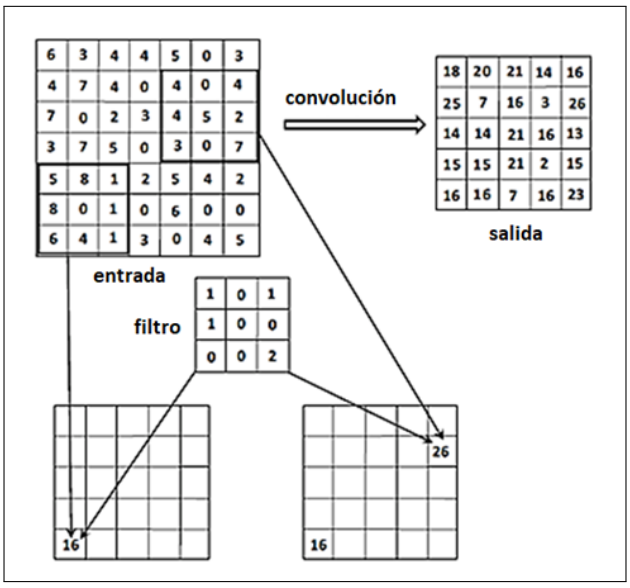

Figura 2. Proceso de convolución.
Fonte: Recuperado de "Neural Networks and Deep Learning a Textbook" de C. Aggarwal (2018).
Convolución entre una entrada de 7x7x1 y un filtro de 3x3x1 con un paso (*stride*) de 1.

La capa pooling (O'Shea & Nash, 2015) ejecuta el muestreo descendente del mapeo característico de la capa anterior, produciendo un nuevo mapeo con una resolución condensada, reduciendo de forma abrupta la dimensión espacial de la entrada.Sus propósitos son reducir el número de parámetros o pesos para disminuir el costo computacional y controlar el sobreajuste de la red (Gholamalinejad & Khosravi, 2020) y la capa densa o también conocida como fully connected layer (O'Shea & Nash, 2015). Esta es el conjunto de "últimas capas" (puede ser

una o más) que se encarga de producir el resultado que se espera y no es más que una red neuronal totalmente conectada. Ella toma de entrada el resultado obtenido de la capa pooling y, mediante procesos denominados forward propagation y backward propagation, va ajustando los pesos para conseguir una correcta predicción.

En problemas de clasificación, la función de activación de la última capa es generalmente una softmax que nos indica la probabilidad de pertenecer a alguna de las clases de estudio. Otras capas pueden utilizar otro tipo de función de activación, entre las que destaca ReLu, ya que ésta refleja de mejor modo el funcionamiento biológico de una neurona y alcanza mejor rendimiento comparada con la función de activación hiperbólica y sigmoid. A pesar de no ser diferenciable en cero y no ser totalmente lineal, ReLu permite también crear representaciones "ralas" (sparse) que favorecen el entrenamiento cuando el número de datos es escaso (Glorot, Bordes, & Bengio, 2010).

Transfer Learning

Tradicionalmente, los algoritmos de aprendizaje son diseñados para afrontar los problemas de manera independiente. Dependiendo de los requerimientos del caso y de la información disponible, un algoritmo es aplicado para entrenar el modelo para dicha tarea. Sin embargo, el proceso denominado "Transfer Learning" lleva el aprendizaje un paso más allá y "lo acerca" a la manera en que los humanos utilizan la información a través de las tareas. En términos simples, el proceso consiste en reutilizar un modelo existente y ya entrenado para una tarea relacionada, extendiéndose o adaptándolo para una nueva tarea (Sarkar & Bali, 2018).

Por su parte, Olivas (2009)lo definen como "la mejora del aprendizaje en una nueva tarea mediante la transferencia de conocimientos de una tarea relacionada que ya se ha aprendido", por lo que podemos inferir que de esta manera se obtiene un rápido progreso y un alto desempeño. Adicionalmente, esta técnica es ampliamente utilizada por su enfoque diverso y en Pan e Yang (2010) se define que el Transfer learning es "la idea de romper el paradigma de resolver un problema aislado y utilizar el conocimiento adquirido para enfocarlo en una tarea similar". Para simplificar la comparación, en la Figura 3 se puede apreciar la diferencia entre el Machine Learning "tradicional" y el Transfer Learning.

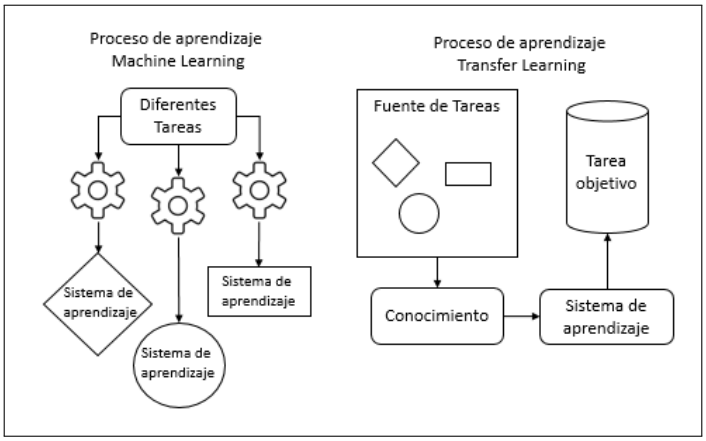

Figura 3. Proceso de aprendizaje machine learning y transfer learning.

Modelo VGG16

VGG16 es una red convolucional propuesta por Simonyan e Zisserman (2015) para la competición anual "The ImageNet Large Scale Visual Recognition Challenge", la cual incluye dos tareas: la primera es detectar objetos dentro de una imagen de 200 clases y la segunda trata de clasificar imágenes dentro de 1000 categorías. En el año 2014, Karen Simonyan y Andrew Zisserman de la Universidad de Oxford ganaron la competición con el modelo VGG16. Una imagen de dicha red se puede apreciar en la Figura 4. La entrada de la red es de tamaño 224, 224, 3. Luego, en cada bloque de convolución se aplica una convolución, una función de activación ReLU y una reducción de dimensionalidad vía capa de agrupación (*"pooling"*). Por último, se usa la función de activación softmax como salida de la red para determinar en cuál de las 1000 categorías pertenece la entrada.

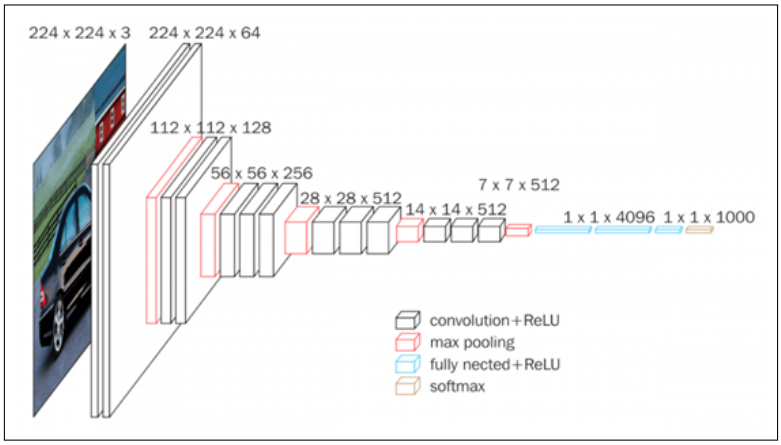

Figura 4. Ilustración estándar de una red VGG16.
Fonte: Recuperado de "VGG16 – Convolutional Network for Classification and Detection" deM. Hassan (2018)

METODOLOGÍA

El trabajo fue realizado bajo un enfoque de desarrollo rápido de aplicaciones (RAD) iterativo, donde se fue evolucionando el modelo y constó de tres etapas principales:

Obtención del dataset

1. Obtención de las imágenes: Las imágenes se consiguieron desde"Kaggle: Brain MRI Images for Brain Tumor Detection" (https://www.kaggle.com/navoneel/brain-mri-images-for-brain-tumor-detectionrecuperado el 12 enero de 2021) y desde el repositorio de imágenes médicas del "Cancer Imaging Archive: Brain-Tumor-Progression"[3].
2. Estandarización de las imágenes: Redimensionamiento de las imágenes a un valor común de tamaño 320x320.
3. Procesamiento vía Data augmentation: Generar imágenes a partir de las conseguidas para tener un dataset mayor, mediante las técnicas de rotación, inversión, desplazamiento, corte y ampliación.

Un mayor detalle se da en la subsección "Generalidades del Dataset" dentro del apartado "Experimentos y Resultados".

Implementación de la Red Neuronal

Se seleccionaron los parámetros iniciales para las pruebas de la red estableciendo una métrica simple determinada por el porcentaje de exactitud en las pruebas, para luego mejorar el modelo a través de uno o más ajustes en la cantidad de capas convolucionales, selección de optimizador y función de activación. Esta implementación fue evolucionando en cada nuevo prototipo en virtud de los resultados obtenidos en las pruebas.

Análisis de Resultados y Ajustes a la Red

Se analizaron los resultados derivados con la red creada y se definieron los ajustes pertinentes a los parámetros establecidos, con el objetivo de optimizar la red y poder obtener mejores porcentajes de precisión. Esto, en pocas palabras, contempló lo siguiente:

1. Análisis de precisión con la muestra de prueba.
2. Cambios en las capas de la red, estableciendo una o más opciones entre nuevas capas en la red, cambio de modelo, agregar capas convolucionales y/o capas de pooling o cambio de optimizador.
3. Reentrenamiento de la red.

Esta forma de trabajo permitió obtener una precisión de un 85% de exactitud en las pruebas, como se presentará en la sección "Resultados".

[3]https://wiki.cancerimagingarchive.net/display/Public/Brain-Tumor-Progressionrecuperado el 15 de octubre de 2020

FRAMEWORK Y HERRAMIENTAS PARA LA IMPLEMENTACIÓN DE LOS MODELOS

Considerando el avance que ha tenido el Deep Learning y las herramientas disponibles, se decidió utilizar Tensorflow y la API de Keras para Python 3.7. Los motivos para esta elección se basan en: 1) La documentación es abundante para las implementaciones y 2) Facilita la implementación de modelos y topologías complejas con pocas líneas de código (TensorFlow Core, 2020). Esto nos ayudó a enfocar los esfuerzos en el análisis y diseño por sobre la implementación. En el marco informativo de este trabajo, las herramientas utilizadas para la codificación fueron: Sequential de keras.models,Conv2D, Maxpooling2D, Dense y Flatten de keras.layers, optimizers de keras y VGG16 de keras.applications.vgg16.

Respecto al Hardware utilizado, se trabajó con un equipo ad-hoc que consta de un procesador AMD Ryzen 5 3600X 6-Core con 16GB de Memoria RAM, una Unidad SSD de 480GB, una unidad de HDD para el almacenamiento de las imágenes con una capacidad de 1TB y una GPU NVIDIA GeForce RTX 2060. Lo anterior nos permitió autonomía de herramientas como Google Colab y manejar nuestros experimentos en un entorno controlado.

Generalidades del experimento

Considerando que la identificación de presencia (o ausencia) de hemorragias y tumores en imágenes de resonancias magnéticas de cerebro es "terreno de investigación incipiente", como declaran Knoll et al. (2020); quienes realizaron un estudio sobre métodos de aprendizaje profundo para la reconstrucción de imágenes de resonancia magnética en paralelo, se decidió comenzar con un modelo simple que fue mejorando en virtud de los resultados alcanzados. De esta manera, la red convolucional creada inicialmente para probar el funcionamiento con las imágenes obtenidas consistió en una red con 1 capa convolucional utilizando 32 filtros de dimensión (2,2), un stride con dimensión de 1,1 y una entrada con dimensión de 320,320,3, para la cual se empleó una función de activación ReLu (Glorot et al., 2010), una capa flatten y una capa densa con 3 neuronas con función de activación Softmax (Petersen & Voigtländer, n.d.).

Este experimento sirvió para identificar puntos de mejora y verificar la eficiencia de nuestro Hardware. Luego, se utilizaron dos capas convolucionales 2D con 32 filtros como un parámetro arbitrario para la identificación de resultados base, optimizador "Adam[4]" y "categorical crossentropy" como función de pérdida. El diseño general de la red anteriormente descripta se puede apreciar en la Figura 5.

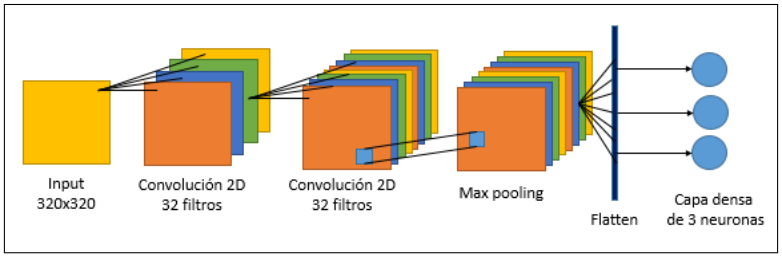

Figura 5. Ilustración del diseño general de la red implementada.

Sin embargo, después de realizar tres iteraciones se decidió explorar el Transfer Learning en virtud de optimizar el tiempo de entrenamiento para esta investigación.

EXPERIMENTOS Y RESULTADOS

Como se mencionó anteriormente, la configuración inicial de red neuronal convolucional generada consistió en una capa de convolución utilizando 32 filtros de dimensiones 2x2, un stride de (1,1) para leer imágenes de 320,320 pixeles, unida a una capa flatten seguida de una capa densa, obteniendo un 28% de precisión en su validación después de 30 épocas. Decidimos mantener la cantidad de épocas para tener un factor constante en las diferentes experiencias. En total se realizaron nueve modelos donde los tres primeros corresponden a una red ad-hoc y los seis siguientes fueron realizados a partir del Transfer Learning de VGG16. Un punto importante para el experimento guarda relación directa con el dataset utilizado y se expone en el apartado "Generalidades del Dataset", incluyendo en dicho espacio una reflexión sobre esto y la validez de los resultados.

Configuraciones previas al Transfer Learning

Las siguientes arquitecturas exploradas fueron combinaciones de diferentes cantidades de capas convolucionales variando el número de filtros y de capas de max pooling, obteniendo que al utilizar 2 capas convolucionales con

[4]"La optimización de Adam es un método del gradiente descendente estocástico que se basa en la estimación adaptativa de momentos de primer y segundo orden": https://keras.io/api/optimizers/adam/

32 filtros y 1 capa de pooling se consiguió un 30% de precisión. Dentro de las distintas combinaciones de capas convolucionales con 32 filtros y capas de pooling intercaladas entre las capas convoluciones, el mayor porcentaje de precisión fue de 34%. Posterior a estos resultados, se decidió – por experiencias previas en otras situaciones – realizar experimentos con Transfer Learning.

Configuraciones posteriores al Transfer Learning bajo optimizador Adam

Con la utilización del transfer learning empleando VGG16, sin reentrenamiento de sus capas, agregando una capa flatten y una capa densa que, al contar con 3 clases para clasificar – imágenes sanas, hemorragias y tumores–, se compuso con 3 neuronas, se logró conseguir un resultado de 30% de precisión. El VGG se cortó "justo antes" de la capa de salida de 1000 neuronas. De esta forma, al utilizar VGG16 se obtuvo un resultado del 32%. El notar que el aumento de la precisión fue bajo pensamos que estaría omitiéndose algo importante y, al ver nuestro dataset, sospechamos que el problema podría estar en el optimizador. Esto porqueAdam es un método de descenso de gradiente estocástico que se basa en la estimación adaptativa de momentos de primer y segundo orden y, en nuestro caso, las variaciones de magnitud parecían ser amplias. Eso fue crucial.

Configuraciones posteriores. El punto de quiebre: El optimizador

Como parte del aprendizaje de nuestro experimento notamos que las imágenes cerebrales cuentan con características particulares que harían muy probable la caída en un "punto silla". Luego, el utilizar Adam como optimizador fue un error (como se indicó anteriormente) y lo corregimos utilizando, en primera instancia, SDG para tener un punto de partida y/o comparación y luego cambiando a RMSprop (Keras, 2021). En concordancia con lo anterior, optamos por el que entregó mayor precisión, el cual fue RMSprop, ya que bajo los mismos parámetros generó mejores resultados. Si bien tanto SGD como RMSprop son optimizadores basados en Gradiente Descendiente, RMSprop equilibra el tamaño del paso, disminuyéndolo para gradientes grandes y aumentando el paso para gradientes pequeños. Lo antes descrito se ajustaba a nuestro dataset ya que posee imágenes donde los gradientes pueden variar ampliamente en cuanto a sus magnitudes, una particularidad de las imágenes cerebrales que "descubrimos" en un momento del análisis.

Dicho lo anterior, al utilizar el optimizador SDG se obtuvo un porcentaje de precisión del 70%, mientras que con el optimizador RMSprop fue alcanzado un porcentaje del 78%, bajo las mismas condiciones. Continuando con este optimizador y probando la red para el caso aislado de estudio exclusivo de imágenes de tumores, se consiguió un porcentaje de precisión del 85% y, para el caso del estudio de las imágenes de hemorragias,fue obtenido un porcentaje del 83%. En síntesis, logramos un modelo que clasificó en tres posibles resultados: Hemorragia, Tumor Cerebral o Ninguno de los anteriores. La Figura 6 muestra los resultados derivados en los diferentes experimentos y en la Tabla 1.

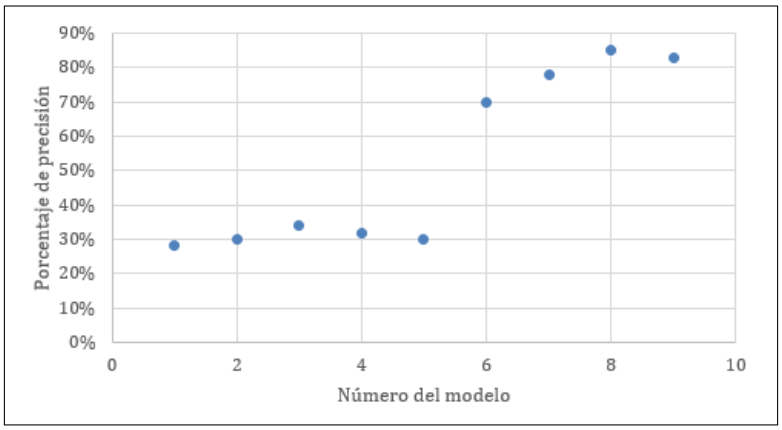

Figura 6. Porcentajes de precisión de los diferentes modelos.

N. de Modelo	Características generales	Precisión
Modelo 1	Una capa convolucional	28%
Modelo 2	Dos capas convolucionales y una capa de pooling	30%
Modelo 3	Seis capas convolucionales y dos capas de pooling	34%
Modelo 4	VGG16	32%
Modelo 5	VGG16 agregando seis capas convolucionales y dos capas de pooling y optimizador Adam	30%
Modelo 6	VGG16 manteniendo las seis capas convolucionales y dos capas de pooling y cambiando el optimizador por SGD	70%
Modelo 7	VGG16 manteniendo las seis capas convolucionales y dos capas de pooling y cambiando el optimizador por RMSprop	78%
Modelo 8	VGG16 manteniendo seis capas convolucionales y dos capas de pooling y optimizador RMSprop, pero solo se utilizaron imágenes sanas y de tumores	85%
Modeo 9	VGG16 manteniendo seis capas convolucionales y dos capas de pooling y optimizador RMSprop, pero solo se utilizaron imágenes sanas y de hemorragias	83%

Tabela 1. Cuadro resumen de resultados.

Generalidades del Dataset

El dataset utilizado, como se mencionó en la Metodología, comprende:

1. Imágenes obtenidas de Kaggle y de imágenes médicas "Cancer Imaging Archive". Dichas imágenes corresponden a dos fuentes diferentes donde se revisó manualmente que ninguna de ellas correspondiese a datos cruzados. En pocas palabras, las imágenes de Kaggle y/o Schamainda poseen datos en común. Sin perjuicio de lo anterior, esto nos llevó al siguiente paso.
2. La estandarización de las imágenes, realizada en el código fuente, fue relevante considerando que dicha reducción (en varios casos nuestras imágenes eran un poco mayores) podrían distorsionar el dataset y se optó por "no bajar tanto" la definición (llegando a 320x320). Si bien es cierto que el modelo VGG-16 fue originalmente entrenado con imágenes de 224x224px[5], el poder computacional de nuestro HW nos permitió trabajar sin inconvenientes con una dimensión 1.4 veces mayor. Sin perjuicio de lo anterior, el principal desafío de esta estrategia está en que a futuro se pueda "ignorar" para trabajar con los tamaños originales de las imágenes.
3. Nuestro dataset estuvo formado por 892 imágenes de cerebros sanos, 1400 imágenes de cerebros con hemorragias y 1186 imágenes de cerebros con tumores. Consideramos que esto es un dataset bastante balanceado, pero que pudo ser mayor (en cuanto a cantidad) por lo que usamos "Data Augmentation" para aplicar rotación, inversión, desplazamiento, corte y ampliación de nuestro dataset.
4. Respecto a la división del Dataset, siguiendo estándares generales obtenidos de la "literatura gris"[6] (y que vemos apropiados) se dividió nuestro dataset en 80% para entrenamiento y 20% de validación.
5. Finalmente, y para dejar público nuestro dataset en su totalidad (sin divisiones) en virtud de aportar a cualquier investigador que lo desee utilizar sin sesgar ello a nuestra división, los invitamos a conseguir el mismo enviando un correo electrónico a mhidalgo@uft.cl o utilizando este enlace https://n9.cl/wwpmk para su descarga.

Consideramos, de igual manera, que el dataset y su utilización es perfectible por lo que se menciona esto en el apartado de "Conclusiones y Trabajo Futuro".

CONCLUSIONES Y TRABAJO FUTURO

Los resultados obtenidos muestran que la implementación de deep learning a través de transfer Learning para el reconocimiento de presencia de hemorragia y tumores cerebrales en imágenes de resonancia magnética del cerebro, tiene una gran capacidad para poder detectar la presencia de éstos al haber logrado valores, como se aprecia en la Tabla 1, sobre el 80% en los experimentos.

Adicionalmente, es importante destacar que la experiencia de este trabajo nos pide poner mayor énfasis en un análisis de los dataset para detectar el optimizador más acorde para dicho grupo de imágenes puesto que, en esta oportunidad, los resultados obtenidos al momento de utilizar el optimizador Adam fueron muy inferiores a los

[5]https://keras.io/api/applications/vgg/
[6]Ejemplo: https://www.aprendemachinelearning.com/sets-de-entrenamiento-test-validacion-cruzada/

que se lograron con el optimizador RMSprop. Además, al analizar de manera independiente el funcionamiento de la red tanto para la detección de hemorragia como para los tumores, se alcanzaron valores de precisión mayores que para cuando la red es entrenada para la detección de ambos tipos de imágenes.Esto permite seguir explorando el trabajo de este documento a través de redes especializadas en una patología particular por sobre una sola red que intente "abarcar más" resultados posibles.

Sin perjuicio de lo anterior, es importante destacar que esta investigación aún posee puntos importantes a desarrollar por lo que si bien es cierto que los resultados conseguidos pueden ser considerados eficientes en un ambiente controlado, en este caso se está estudiando la detección de dos afecciones cerebrales las cuales pueden comprometer severamente la vida de una persona.Por esto, la precisión obtenida podría no ser adecuada para su uso en el actual campo médico como un "diagnóstico definitivo", pero sí se puede considerar un apoyo para discriminar casos que, posiblemente, podrían ser importantes de ser examinados por un especialista antes que otros. En términos simples, priorizar.

Además, consideramos que la precisión alcanzada puede ser mejorada mediante la obtención de un dataset más amplio que el utilizado durante este trabajo, pero nuestro país (Chile) presenta limitaciones legislativas al respecto, aunque se podrían mitigar "anonimizando" el dataset, sin embargo, de todas maneras, se deberán buscar nuevas fuentes para enriquecer el banco de imágenes y potenciar el entrenamiento de la red. Otro punto importante para el trabajo futuro está en la utilización de las imágenes sin modificar su tamaño y estudiar los efectos que esto podría llevar en cuanto a las predicciones en un entorno controlado. En lo venidero, consideramos importante hacer pruebas en nuestras propias configuraciones (redes neuronales propias) para determinar si otros optimizadores podrían tener un mejor desempeño en este tipo de imágenes de manera de identificar o dar mayor sustento a nuestro "punto de quiebre": El optimizador. Finalmente, podemos decir que este trabajo da un punto de partida para que futuras investigaciones puedan generar mayores niveles de precisión y con ello se pueda reducir el tiempo para diagnosticar casos críticos y agilizar el inicio de tratamientos vitales para los pacientes.

AGRADECIMIENTOS

Quisiéramos agradecer al Dr. Ricardo Ñanculef de la Universidad Técnica Federico Santa María (Chile) por habernos enseñado este fascinante campo del Deep Learning cuando fuimos estudiantes en su curso de Redes Neuronales Artificiales. Adicionalmente, extendemos nuestro agradecimiento a la Facultad de Ingeniería de la Universidad Finis Terrae (UFT, Chile) por el apoyo para la realización de este proyecto y, en especial, a su Directora, la Dra. Felisa Córdova, por fomentar la investigación como parte del plan estratégico de desarrollo dando lineamientos y abriendo puertas para nuestra realización académico-investigativa.

REFERÊNCIAS

Aggarwal, C. (2018). *Networks and deep learning a textbook.* Yorktown Heights, USA: Springer International Publishing.

Ari, A., & Hanbay, D. (2018). Deep learning based brain tumor classification and detection system. *Turkish journal of Electrical Engineering Computer Sciences*, *26*, 2275–2286. doi: 10.3906/elk-1801-8

Cao, Z., Lixin, D., Yang, G., Yue, T., Chen, Q., Fu, H., & Xu, Y. (2017). Breast tumor detection in ultrasound images using deep learning. *Lecture Notes in Computer Science*(10530), 121–128. doi: 10.1007/978-3-319-67434-6_14

Chollet, F. (2017). *Deep learning with python.* Shelter Island, New York: Manning Publications.

Gholamalinejad, H., & Khosravi, H. (2020). Pooling methods in deep neural networks, a review. *Computing Research Repository*. Recuperado de https://arxiv.org/abs/2009.07485

Glorot, X., Bordes, A., & Bengio, Y. (2010). Deep sparse rectifier neural networks. *Journal of Machine Learning Research*, *15*. doi: https://proceedings.mlr.press/v15/glorot11a/glorot11a.pdf

Hassan, M. (2018). *Vgg16: Convolutional network for classification and detection.* Recuperado de https://neurohive.io/en/popular-networks/vgg16/

Keras. (2021). *Keras.* Recuperado de https://keras.io/why_keras/

Knoll, F., Hammernik, K., Zhang, C., Moeller, S., Pock, T., Sodickson, D. K., & Akcakaya, M. (2020). Deep-learning methods for parallel magnetic resonance imaging reconstruction: A survey of the current approaches, trends, and issues. *IEEE Signal Processing Magazine*, *37*(1), 128-140. doi: 10.1109/MSP.2019.2950640

Lecun, Y. (1989). *Generalization and network design strategies.* Zurich: Elsevier.

McCulloch, W. S., & Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, *5*, 115–133. doi: 10.1007/BF02478259

Olivas, E. S. (2009). *Handbook of research on machine learning applications and trends: Algorithms, methods and techniques.* Hershey, PA, USA: IGI Publishing.

O'Shea, K., & Nash, R. (2015). Introduction to convolutional neural networks. *Computing Research Repository*. Recuperado de https://arxiv.org/abs/1511.08458

Pan, S. J., & Yang, S. J. (2010). A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, *22*(10). doi: 10.1109/TKDE.2009.191

Petersen, P., Raslan, M., & Voigtlaender, F. (2020). Topological properties of the set of functions generated by neural networks of fixed size. *Computing Research Repository*. Recuperado de https://arxiv.org/abs/1806.08459

Petersen, P., & Voigtländer, F. (s.d.). Optimal approximation of piecewise smooth functions using deep relu neural networks. *Neural networks: the official journal of the International Neural Network Society*(108), 296–330. doi: 10.1016/j.neunet.2018.08.019

Sarkar, D., & Bali, R. (2018). *Hands-on transfer learning with python.* Shelter Island, New York: Manning Publications.

Schmidhuber, J. (2014). Deep learning in neural networks: An overview. *Computing Research Repository*. Recuperado de https://arxiv.org/abs/1404.7828

Simonyan, K., & Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition. *Computing Research Repository*. Recuperado de https://arxiv.org/abs/1409.1556

TensorFlow Core. (2020). *TensorFlow Core.* Recuperado de https://www.tensorflow.org/tutorials?hl=pt-br

Hidalgo Barrientos, M. F., Hayes Ortiz, B. I., Delgadillo Vera, I., & Goyo Escalona, M. (2022). Deep learning aplicado para la detección de hemorragias y tumores cerebrales. *AtoZ: novas práticas em informação e conhecimento, 10(3)*, 1 – 10. Recuperado de: http://dx.doi.org/10.5380/atoz.v11i.81284