



Prototipo para la clasificación automática de imágenes histológicas para medir el riesgo de carcinoma ductal invasivo usando redes neuronales

Daniel José Jordán Ortega

Universidad El Bosque

Departamento de Matemáticas

Bogotá, Colombia

2019

Prototipo para la clasificación automática de imágenes histológicas para medir el riesgo de carcinoma ductal invasivo usando redes neuronales

Daniel José Jordán Ortega

Trabajo de grado presentado como requisito parcial para optar al título de:

Matemático

Directores:

PhD. Tomasso Benincasa

PhD. Diego Fernando Aranda Lozano

Universidad El Bosque

Departamento de Matemáticas

Bogotá, Colombia

2019

NOTA DE APROBACIÓN

El trabajo de grado titulado “**Prototipo para la clasificación automática de imágenes histológicas para medir el riesgo de carcinoma ductal invasivo usando redes neuronales**”, presentado por el estudiante Daniel José Jordán Ortega como requisito para optar al título de Matemático fue evaluado, calificado y aprobado por los jurados evaluadores.

Prof. Orlando López Cruz

Jurado

Prof. Efraín Patiño Blandón

Jurado

Prof. Carlos Puentes Morales

Jurado

“La Universidad El Bosque no se hace responsable de los conceptos emitidos por los estudiantes en su trabajo de grado, solo velará por el rigor científico, metodológico y ético de éste en aras de la búsqueda de la verdad y la justicia.”

AGRADECIMIENTOS

A Dios, Carola, Pao y Sarita por dejarme llegar.

RESUMEN

Este trabajo de grado se centra en el diseño e implementación de un prototipo basado en redes neuronales, específicamente de aprendizaje profundo (Deep Learning), capaz de clasificar de manera automática imágenes histológicas con Carcinoma Ductal Invasivo. El Carcinoma Ductal Invasivo, a veces denominado Carcinoma Ductal Infiltrante, es el tipo más común de cáncer de mama y alrededor del 80% de todos los casos de cáncer de mama son de esta naturaleza. Se utilizó para ello una variante de la red VVGNet 14 y redes neuronales convolucionales, con los cuales se logró un prototipo con una capacidad aproximada de 79% de sensibilidad y 79% de especificidad en el reconocimiento de imágenes histológicas con CDI.

Palabras claves: carcinoma ductal invasivo, redes convolucionales, imágenes.

ABSTRACT

This degree project focuses on the design and implementation of a prototype based on neural networks, specifically Deep Learning, capable of automatically classifying histological images with Invasive Ductal Carcinoma. Invasive Ductal Carcinoma sometimes called Infiltrating Ductal Carcinoma, is the most common type of breast cancer and about 80% of all cases belong to this class. A variant of the VVGNet 14 network and convolutional neural networks were used for this purpose, resulting in a prototype with an approximate capacity of 79% sensitivity and 79% specificity in the recognition of histological images with CDI.

Keywords: invasive ductal carcinoma, convolutional networks, images.

CONTENIDO

NOTA DE APROBACIÓN.....	3
AGRADECIMIENTOS.....	5
RESUMEN.....	6
ABSTRACT.....	7
FIGURAS.....	10
TABLAS.....	12
1. INTRODUCCIÓN.....	13
2. JUSTIFICACIÓN DEL PROYECTO.....	17
3. OBJETIVOS.....	17
3.1 Objetivo general.....	17
3.2 Objetivos específicos.....	17
4. METODOLOGÍA.....	18
5. CONSIDERACIONES ÉTICAS.....	18
6. PROBLEMA MÉDICO.....	19
6.1 ¿Qué es el cáncer?.....	19
6.2 ¿Qué diferencia hay entre una célula normal y una cancerosa?.....	20
6.3 Evolución del cáncer.....	21
6.4 ¿Qué causa el cáncer?.....	21
6.5 Tipos de cáncer.....	22
6.6 Grado de un tumor.....	26
6.7 Estadificación del cáncer.....	28
6.8 ¿Qué es el cáncer de mama?.....	29
6.9 ¿Existe predisposición al cáncer de mama?.....	32
6.10 Tipos de cáncer de mama.....	33
6.11 ¿Qué es el Carcinoma Ductal Invasivo o CDI?.....	33
7. ¿QUÉ SON LAS REDES NEURONALES?.....	36
8. MATEMÁTICAS DE LAS REDES NEURONALES.....	44
8.1 Red neuronal como clasificador binario.....	44
8.2 ¿Cómo aprenden las redes neuronales?.....	47
8.3 Red neuronal como clasificador múltiple.....	52

8.4 Red neuronal como predictor	53
8.5 Regla Delta o Descenso del Gradiente	54
8.6 Aprendizaje por lotes.....	56
8.7 Reconocimiento de patrones no lineales	57
8.8 ¿Cómo aprenden los perceptrones multicapa?	60
8.9 Redes neuronales convolucionales (CNN).....	63
9. DISEÑO E IMPLEMENTACIÓN DEL PROTOTIPO	67
10. RESULTADOS	71
11. CONCLUSIONES	73
12. RECOMENDACIONES	74
13. REFERENCIAS BIBLIOGRÁFICAS	75
14. CÓDIGO	77

FIGURAS

Fig.1. Evolución de las células normales a cancerosas.....	21
Fig.2. Morfología del seno.....	30
Fig.3. Morfología del seno.....	31
Fig.4. Evolución del CDI en el ducto de leche.....	34
Fig.5.Cáncer ductal in situ.....	35
Fig.6. Cáncer ductal invasivo.....	35
Fig.7. Funciones más comunes que pueden realizar las redes neuronales artificiales.....	37
Fig.8. Neurona biológica versus artificial.....	38
Fig.9. Proceso de vectorización de una imagen a colores a 3 canales RGB.....	40
Fig.10. Proceso de vectorización de una imagen en blanco y negro a 1 canal.....	41
Fig.11. Simplificación de la estructura de una red neural.....	42
Fig.12. Dos variables afectan el desempeño de una red neuronal.....	43
Fig.13. Vista gráfica de la función umbral o de activación.....	45
Fig.14. Línea recta que delimita los patrones de entradas.....	47
Fig.15. Perceptrón compuesto de una sola neurona.....	49
Fig.16. Perceptrón con múltiples salidas.....	52
Fig.17. Líneas de decisión en un perceptrón con 3 salidas.....	52
Fig.18. Modelo de neurona lineal.....	54
Fig.19. Descenso de gradiente de la función error.....	56
Fig.20. Neurona no lineal con función de activación.....	58
Fig.21. a) Tangente hiperbólica b) Sigmoide o logística c) Relu.....	59
Fig.22. Perceptrón multicapa.....	61
Fig.23. Modelo de una red convolucional.....	64
Fig.24. Ejemplo de un filtro y capa convolucional.....	64
Fig.25. Filtro recorriendo los tres canales de colores.....	65
Fig.26. Tipos de pooling.....	66

Fig.27. Arquitectura VGGNet 14 con 21 capas.....	68
Fig.28. Arquitectura en detalle de VGGNet 14 con entradas de 224x224x3.....	68
Fig.29. Arquitectura final del modelo CDI con entradas de 48x48x3.....	69
Fig.30. Proceso de entrenamiento de la red neuronal.....	71

TABLAS

Tabla 1. Datos de entradas.....	44
Tabla 2. Clasificación del TLU.....	45
Tabla 3. Precisión por clase.....	72
Tabla 4. Resultados globales.....	72

1. INTRODUCCIÓN

El cáncer de mama es un crecimiento anormal e incontrolable de las células mamarias usualmente como resultado de mutaciones en genes que controlan la proliferación y muerte celular. En la mayoría de los casos, estas mutaciones ocurren debido a eventos aún no plenamente entendidos con efectos acumulativos durante el tiempo de vida de la persona (Instituto Nacional de Cancerología ESE, 2013).

El cáncer de mama es en la actualidad la primera causa de cáncer en el mundo tanto en países desarrollados como en vía de desarrollo (Instituto Nacional del cáncer de los institutos nacionales de la salud de los Estados Unidos, 2016). Los principales órganos de cáncer primario en el género femenino son distribuidos así: cáncer de mama con 29%, cáncer de pulmón y bronquios 14%, cáncer de colon y recto con 9%, cáncer de cuerpo uterino 6%, cáncer de tiroides 5% y otros cánceres 37%. En los países desarrollados la incidencia del cáncer de mama corresponde a 122 casos por cada 100.000 mujeres (en descenso), con una mortalidad de 15,3 por cada 100.000 (en descenso), mientras que en Colombia la incidencia es de 35 por cada 100.000 (en ascenso), y la mortalidad es de 9,5 por cada 100.000 (en ascenso), lo que constituye al país como de riesgo intermedio, pero dado a que en las dos últimas décadas los niveles han aumentado, esta enfermedad se ha convertido en un problema de salud pública (Ramírez-Martínez CM, 2015).

El tipo más común de cáncer de mama es el Carcinoma Ductal Invasivo (CDI), a veces denominado carcinoma ductal infiltrante. Alrededor del 80% de todos los casos de cáncer de mama son de este tipo. Este tipo de cáncer resulta más común a medida que la mujer envejece. Aproximadamente dos tercios de las mujeres con CDI tienen 55 años o más. El CDI también afecta a los hombres (American Cancer Society, 2019).

El diagnóstico de CDI incluye una combinación de procedimientos, entre ellos, una exploración física y estudios por imágenes. En la exploración física el médico busca bultos en la mama que se noten al tacto. También se palpan los ganglios linfáticos bajo la axila y sobre la clavícula para determinar si hay inflamación u otras alteraciones atípicas.

De los estudios por imagen, la prueba más común es la mamografía que es una prueba que se obtiene por medio de rayos X en la mama. Esta prueba se lleva a cabo en mujeres aparentemente sanas para detectar indicios precoces de cáncer. Si se detecta la presencia de anomalías, tales como proyecciones salientes en forma de dedos, existe una gran probabilidad de la existencia de CDI y que éste haya invadido a otros tejidos. Para reunir más información acerca del área afectada, puede llevarse a cabo otras pruebas de imagen tales como ecografías (emisión de ondas sonoras), IRM o resonancia magnética (usa campos magnéticos y ondas de radio). Finalmente, si estos estudios revelan la presencia de anomalías, el protocolo médico indica la realización de una biopsia, que es la extirpación parcial o total del tejido de aspecto anormal para que un médico patólogo lo clasifique (Breastcancer.org, 2014).

En la ciudad de Medellín, por ejemplo, el 51% de los diagnósticos de CDI corresponde a estadios avanzados de cáncer (48% estadios avanzados y 3% en estadios metastásicos). Esta situación pone en evidencia la necesidad de implementación de estrategias clínicas e imagenológicas para diagnosticar en mayor proporción el cáncer de mama en estadios tempranos en las mujeres de nuestra población (Ramírez-Martínez CM, 2015).

Por otro lado, los métodos tradicionales utilizados por los patólogos para el análisis de imágenes histológicas son manuales y requieren un alto grado de experticia, esfuerzo y tiempo. En ocasiones llega a ser tan complejo el estudio de las imágenes, que surge

variabilidad de lecturas y posiciones encontradas entre los propios patólogos (Andrew Janowczyk, 2016)

En la actualidad existen técnicas matemático-informáticas modernas como las redes neuronales que se utilizan en estudios imagenológicos para el reconocimiento y diagnóstico de ciertos tipos de cáncer, tales como cáncer de piel (Andre Esteva, 2017) o cáncer de mama (Andrew Janowczyk, 2016).

Este trabajo se centra en el diseño e implementación de un prototipo basado en redes neuronales, específicamente de aprendizaje profundo (Deep Learning), capaz de clasificar de manera automática imágenes histológicas y su riesgo asociado ante la posible presencia de CDI.

Para la clasificación de datos, imágenes histológicas en este caso en particular, el prototipo a implementar usará el siguiente criterio: cero (0) para categorizar imágenes sin cáncer y uno (1) para imágenes con cáncer.

La forma de medir el riesgo ante la presencia o no de cáncer se basará en la capacidad predictiva del prototipo, es decir, de su precisión para el reconocimiento de la enfermedad. Este indicador se refiere al porcentaje de imágenes del conjunto de pruebas que pueden ser categorizadas correctamente independientemente de la clase a la que pertenece. La forma de cálculo es la siguiente:

$$\text{Precisión total} = \frac{\text{Total imágenes reconocidas}}{\text{Total imágenes de prueba}} = \frac{VP + VN}{P + N}$$

Para medir la bondad del prototipo, se usarán dos indicadores relacionados con la capacidad diagnóstica de los test médicos: sensibilidad y especificidad.

La sensibilidad indica la capacidad del prototipo para dar como casos positivos aquellos que realmente tienen cáncer, es decir, caracteriza la capacidad de la prueba para detectar la enfermedad en sujetos enfermos.

$$\text{Sensibilidad} = \frac{VP}{VP + FN}$$

La especificidad indica la capacidad del prototipo para dar como casos negativos los casos realmente sanos, es decir, caracteriza la capacidad de la prueba para detectar la ausencia de la enfermedad en sujetos sanos.

$$\text{Especificidad} = \frac{VN}{VN + FP}$$

donde VP es verdadero positivo, VN es verdadero negativo, P es positivos (con cáncer), N es negativos (sin cáncer).

2. JUSTIFICACIÓN DEL PROYECTO

- Técnica: complementar los procedimientos médicos actuales para la detección de CDI a través de redes neuronales.
- Social: diseñar un modelo matemático-informático de bajo costo que complemente el procedimiento médico regular para el diagnóstico de CDI.
- Institucional: culminar mi proyecto de grado de Matemáticas en la Universidad El Bosque.

3. OBJETIVOS

3.1 Objetivo general

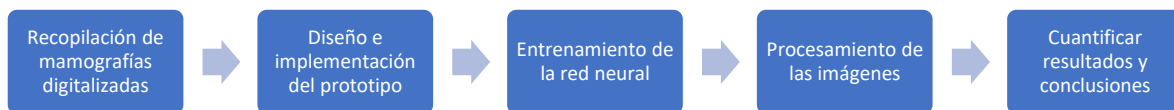
Diseñar e implementar un prototipo basado en redes neuronales que detecte la presencia de CDI de forma automática en imágenes histológicas.

3.2 Objetivos específicos

1. Diseñar e implementar un prototipo basado en redes neuronales que permita clasificar imágenes histológicas de CDI con una sensibilidad de al menos 75%.
2. Diseñar e implementar un prototipo basado en redes neuronales que permita clasificar imágenes histológicas de CDI con una especificidad de al menos 75%.

4. METODOLOGÍA

1. Recopilar imágenes histológicas de mamografías sanas y con CDI en formato digital.
2. Analizar y diseñar la topología de la red neuronal para clasificar las imágenes histológicas.
3. Implementar la red neuronal en un lenguaje multiplataforma de distribución gratuita.
4. Entrenar el prototipo haciendo uso de las imágenes digitalizadas y su diagnóstico.
5. Procesar las imágenes digitalizadas en el prototipo.
6. Obtener los valores de sensibilidad y especificidad del prototipo.
7. Obtener conclusiones y próximos pasos.



5. CONSIDERACIONES ÉTICAS

No son necesarios los datos personales de las pacientes cuyas mamografías son utilizadas en este proyecto.

6. PROBLEMA MÉDICO

6.1 ¿Qué es el cáncer?

Es una enfermedad donde algunas de las células del cuerpo empiezan a dividirse sin detenerse y se diseminan a los tejidos del derredor (Instituto Nacional del cáncer de los institutos nacionales de la salud de los Estados Unidos, 2016).

En estado normal, las células humanas crecen y se dividen para formar nuevas células, y cuando envejecen o se dañan, mueren para que células nuevas las remplacen. Sin embargo, ante la presencia de cáncer, este proceso ordenado se descontrola dando como resultado a un conjunto de células anómalas que se reproducen sin control. En la medida que estas células se hacen más anormales, su ciclo de vida cambia y sobreviven en lugar de morir. También se siguen reproduciendo cuando no son necesarias pudiendo llegar a formar masas que se denominan tumores. Los tumores pueden ser sólidos o no (los cánceres de la sangre, como las leucemias, no forman tumores sólidos).

Cuando un tumor se extiende a tejidos cercanos significa que es canceroso, y sus células pueden desprenderse y viajar a otros lugares del cuerpo a través del sistema circulatorio o linfático, dando origen a nuevos tumores. La propagación de las células de cáncer se denomina metástasis. Una característica importante de los tumores cancerosos o malignos es que pueden reproducirse de nuevo a pesar de haber sido removidos físicamente.

Cuando, por ejemplo, las células del cáncer de pulmón comienzan a crecer en los huesos, se sigue llamando cáncer de pulmón pues sus células son idénticas a las que se originaron en el pulmón. La enfermedad no es referida como cáncer óseo (cáncer en los huesos) salvo que se haya originado en los huesos (American Cancer Society, 2019).

Cabe destacar que el cáncer es una enfermedad genética, es decir, es causado por cambios en los genes que controlan la forma como funcionan nuestras células, en particular, la forma como crecen y se dividen.

Estos cambios pueden ser hereditarios o no, pudiendo aparecer como resultado de errores en la división celular o por daño del ADN ocasionado por exposición a ciertos elementos del ambiente, como por ejemplo, el humo de tabaco, la radiación, rayos ultravioleta, ingesta excesiva de carnes rojas, contacto con asbesto, entre otros. Los cambios genéticos en las células cancerosas son más comunes que en las células sanas.

6.2 ¿Qué diferencia hay entre una célula normal y una cancerosa?

Las células cancerosas son menos especializadas que las células normales, es decir, mientras las últimas tienen funciones muy específicas en el organismo, el foco de las primeras es dividirse sin pausa.

Cuando las células normales ya han envejecido, se inicia un proceso llamado apoptosis o muerte celular programada, en el cual el cuerpo se deshace de ellas. Sin embargo, las células cancerosas pueden ignorar este mandato genético para proseguir con su proceso de división. Además de dividirse sin control, las células cancerosas pueden inducir a las células normales a crear un microambiente beneficioso para el crecimiento de un tumor, como por ejemplo, la generación de vasos sanguíneos que transporten oxígeno y nutrientes.

Otra característica del cáncer es su capacidad para manipular al sistema inmunitario para que no elimine las células anómalas, con lo cual, éstas permanecen en el organismo.

6.3 Evolución del cáncer

Existen cambios en algunas células del cuerpo que no son considerados cáncer, pero que si no son tratados a tiempo pueden devenir en uno. Por ejemplo, cuando las células normales en un tejido se dividen más rápido de lo normal y se van acumulando; a este fenómeno se le conoce como hiperplasia.

Otro cambio que no es considerado cáncer es la displasia, que es la alteración en el tamaño, forma y organización celular. Sin embargo, cuanto más anormales se ven las células y el tejido, mayor es la posibilidad de que se forme cáncer. Un ejemplo común de displasia celular son los lunares atípicos o nevos displásicos. En la figura 1 se observa la evolución del cáncer:

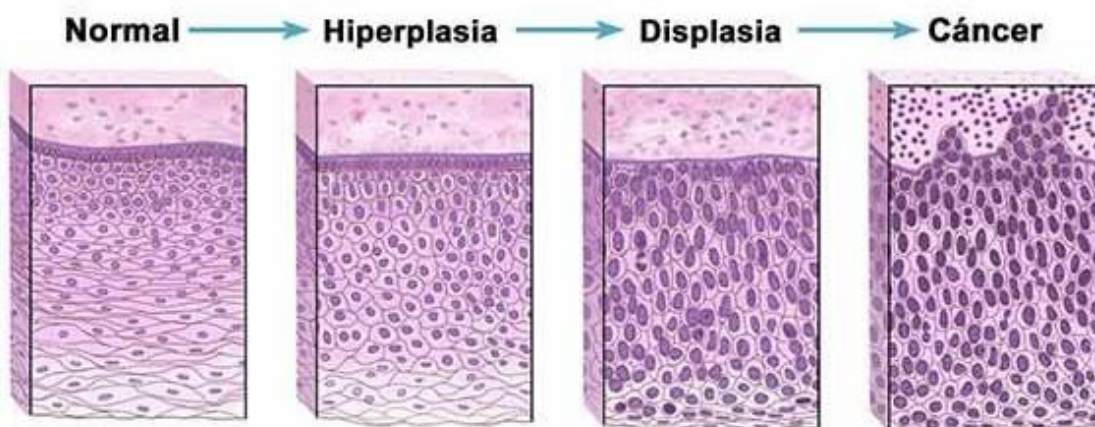


Fig.1. Evolución de las células normales a cancerosas (fuente: Terese Winslow, NIH).

6.4 ¿Qué causa el cáncer?

Dado que el cáncer es una enfermedad genética, la causa que lo origina tiene su génesis en la mutación de algunos genes, a saber:

- Los protooncogenes.
- Genes supresores de tumores.

Los protooncogenes son genes normales cuya función principal está asociada al crecimiento y la división celular. Cuando alguno de esos genes sufre una mutación, se altera su funcionalidad permitiendo que las células crezcan y sobrevivan cuando no deberían dando origen a la progresión tumoral.

Los genes supresores tienen como funcionalidad impedir la proliferación de tumores, pero cuando sufren mutaciones, esta capacidad se inactiva generando una pérdida de función que tiene un efecto recesivo en el gen.

Cuando un protooncogén muta se convierte en un oncogén, que es un gen anormal responsable de la transformación de una célula normal en una maligna que desarrollará un determinado tipo de cáncer. En condiciones normales las células sólo proliferan cuando reciben señales muy específicas que provienen de otras células vecinas. Los oncogenes, sin embargo, violan este esquema: ignoran todas las señales que reciben del exterior, y siguen sus propios esquemas de proliferación, invadiendo no sólo los espacios adyacentes, sino también sitios alejados del lugar de origen, a través del proceso de metástasis (Bruce Alberts, 2008).

6.5 Tipos de cáncer

A continuación se listan algunas de las categorías de cánceres más relevantes según el Instituto Nacional del Cáncer de los Estados Unidos:

- a) Carcinoma: son los tipos más comunes de cáncer. Se forman en las células epiteliales, las cuales cubren las superficies internas y externas del cuerpo. Los carcinomas que empiezan en diferentes tipos de células epiteliales tienen nombres específicos:

- El adenocarcinoma es un cáncer que se forma en las células epiteliales que producen fluidos o mucosidad. La mayoría de los cánceres de seno, de colon y de próstata son adenocarcinomas.
 - El carcinoma de células basales es un cáncer que empieza en la capa más baja o basal (en la base) de la epidermis, la cual es la capa exterior de la piel de una persona.
 - El carcinoma epidermoides o de células escamosas: el origen de este cáncer son células epiteliales que están debajo de la superficie exterior de la piel y tienen forma de escama. Estas células revisten muchos otros órganos, como el estómago, los intestinos, los pulmones, la vejiga y los riñones.
 - El carcinoma de células de transición es un cáncer que se forma en un tipo de tejido llamado epitelio de transición o urotelio. Este tejido se encuentra en el revestimiento de la vejiga, de los uréteres, en parte de los riñones y en algunos otros órganos.
- b) Sarcoma: son cánceres que se forman en el hueso y en los tejidos blandos, incluso en músculos, tejido adiposo (graso), vasos sanguíneos, vasos linfáticos y en tejido fibroso (como tendones y ligamentos).
- El osteosarcoma es el cáncer de hueso más común.
 - Los tipos más comunes de sarcoma de tejido blando son el leiomiomasarcoma, el sarcoma de Kaposi, el histiocitoma fibroso maligno, el liposarcoma y el dermatofibrosarcoma protuberante.
- c) Leucemia: son cánceres que empiezan en los tejidos que forman la sangre en la médula ósea y se caracterizan por no formar tumores sólidos. En vez de eso, un gran número de glóbulos blancos anormales se acumulan en la sangre y en la médula ósea

y desplazan a las células sanguíneas sanas, lo que disminuye la cantidad de plaquetas, glóbulos blancos y glóbulos rojos sanos. La concentración baja de células normales de la sangre puede hacer que el cuerpo lleve con dificultad oxígeno a los tejidos, que no controle las hemorragias o que no combata las infecciones. Hay cuatro tipos comunes de leucemia que se agrupan de acuerdo con la rapidez con la que empeora la enfermedad:

- Aguda: las células sanguíneas anormales son inmaduras (blastos). No pueden cumplir sus funciones normales y se multiplican rápido. La leucemia aguda exige un tratamiento oportuno y agresivo.
- Crónica: existen muchos tipos de leucemias crónicas. Algunas producen demasiadas células y otras, muy pocas. La leucemia crónica comprende células sanguíneas más maduras que se replican y acumulan muy lentamente y pueden funcionar con normalidad durante un tiempo.
- Linfoblástico: este tipo de leucemia afecta los linfocitos que forman el tejido linfoide o linfático.
- Mieloide: este tipo de leucemia afecta las células mieloides que dan origen a los glóbulos rojos, los glóbulos blancos y las células que producen plaquetas.

d) Linfoma: es un cáncer que empieza en los linfocitos (células T o células B). Estos son glóbulos blancos que combaten las enfermedades y que forman parte del sistema inmunitario. En el linfoma, los linfocitos anormales se acumulan en los ganglios linfáticos y en los vasos linfáticos, así como en otros órganos del cuerpo. Hay dos tipos principales de linfomas:

- Linfoma de Hodgkin: aunque puede comenzar en cualquier lugar, se origina con más frecuencia en los ganglios linfáticos de la parte superior del cuerpo como el

tórax, el cuello o debajo de los brazos. Con frecuencia se propaga por los vasos linfáticos de ganglio a ganglio. Pocas veces durante la etapa avanzada de la enfermedad puede invadir el torrente sanguíneo y propagarse a otras partes del cuerpo, como el hígado, los pulmones y/o la médula ósea.

- Linfoma no Hodgkin: es un grupo grande de cánceres que empiezan en los linfocitos. Los cánceres pueden crecer con rapidez o con lentitud y se pueden formar de células B o de células T.
- e) Mieloma múltiple: es un cáncer que empieza en las células plasmáticas, otro tipo de células inmunitarias. Las células plasmáticas anormales, llamadas células de mieloma, se acumulan en la médula ósea y forman tumores en los huesos de todo el cuerpo. El mieloma múltiple se llama también mieloma de células plasmáticas y enfermedad de Kahler.
- f) Melanoma: este cáncer empieza en los melanocitos que son células especializadas en producir melanina (el pigmento que da el color a la piel). La mayoría de los melanomas se forman en la piel, pero pueden formarse también en otros tejidos pigmentados, como en los ojos.
- g) Tumores de cerebro y de la médula espinal: hay diferentes tipos de tumores de cerebro y de la médula espinal. Estos tumores se llaman según el tipo de célula en donde se formaron. Por ejemplo, un tumor astrocítico empieza en las células del cerebro que tienen forma de estrella.
- h) Tumores de células germinativas: empiezan en las células que forman los espermatozoides o los óvulos.

- i) Tumores neuroendocrinos: se forman de células que secretan hormonas en la sangre como respuesta a una señal del sistema nervioso. Estos tumores pueden causar muchos síntomas diferentes.
- j) Tumores carcinoides: son un tipo de tumores neuroendocrinos de crecimiento lento que se encuentran generalmente en el aparato gastrointestinal (con más frecuencia en el recto y en el intestino delgado). Los tumores carcinoides pueden diseminarse al hígado o a otros sitios del cuerpo, y pueden secretar sustancias como serotonina o prostaglandinas y causar síndrome carcinoide.

6.6 Grado de un tumor

Es la descripción de un tumor que se basa en la forma como se ven al microscopio las células y el tejido que lo forma. Es un indicativo de la rapidez con la que probablemente crecerá y se extenderá. Si sus células y la organización del tejido se asemejan a las células y tejidos normales, el tumor se llama "bien diferenciado", y tienden a crecer y a extenderse a un paso más lento que los que son "indiferenciados" o "escasamente diferenciados", los cuales tienen células que se ven anormales y a los que les pueden faltar las estructuras de tejido normal.

En general, se asigna un grado de 1, 2, 3 o 4 dependiendo de qué tan anormal sean. En los de grado 1 y 2, las células tumorales y la organización del tejido tienen una apariencia cercana a la normal. Tienden a crecer y a diseminarse lentamente. Por el contrario, las células y el tejido de los tumores de grado 3 y 4 no se ven normales, y tienden a crecer y diseminarse con más rapidez que los tumores de un grado inferior.

Generalmente se usa el siguiente sistema de gradación:

GX: No es posible asignar un grado (grado indeterminado)

G1: Bien diferenciado (grado bajo)

G2: Moderadamente diferenciado (grado intermedio)

G3: Escasamente diferenciado (grado alto)

G4: Indiferenciado (grado alto)

El cáncer de seno (mama) tiene su sistema propio de gradación. El más usado es el sistema Nottingham (llamado también modificación de Elston-Ellis del sistema de gradación de Scarff-Bloom-Richardson). Este sistema da un grado a los tumores de seno basándose en las siguientes características:

- Formación tubular: qué tanto del tejido del tumor tiene estructuras normales de conductos de leche.
- Grado nuclear: una evaluación del tamaño y forma del núcleo en las células tumorales.
- Rango mitótico: cuántas células que se dividen están presentes, lo cual es una medida de la rapidez con la que el tumor está creciendo.

A cada una de las categorías se asigna una puntuación del 1 al 3; una puntuación de "1" significa que las células y el tejido del tumor se ven normal, y una puntuación de "3" significa que las células y el tejido se ven anormales en su mayoría. Las puntuaciones para las tres categorías se añaden luego, lo que da un resultado del 3 al 9. Hay tres grados posibles:

Puntuación total = 3–5: G1 (grado bajo o bien diferenciado).

Puntuación total = 6–7: G2 (grado intermedio o moderadamente diferenciado).

Puntuación total = 8–9: G3 (grado alto o escasamente diferenciado)

(American Joint Committee on Cancer, 2010).

6.7 Estadificación del cáncer

El estadio se refiere a qué tan grande es el tumor y si se ha extendido. El sistema TNM es el sistema de estadificación de cáncer de mayor uso:

La T se refiere al tamaño y extensión del tumor principal. La N se refiere a la extensión de cáncer que se ha diseminado a los ganglios (o nódulos) linfáticos cercanos. La M se refiere a si el cáncer se ha metastatizado.

Luego, habrá números después de cada letra que dan más detalles acerca del cáncer, por ejemplo:

TX: No puede medirse un tumor primario.

T0: No puede encontrarse un tumor primario.

T1, T2, T3, T4: Se refiere al tamaño y/o extensión del tumor principal. En cuanto más grande es el número después de la T, mayor es el tumor o tanto más ha crecido en los tejidos cercanos. Las T pueden dividirse todavía más para proveer más detalle, como T3a y T3b.

NX: no puede medirse el cáncer en los ganglios linfáticos cercanos.

N0: no hay cáncer en los ganglios linfáticos cercanos.

N1, N2, N3: se refiere al número y ubicación de los ganglios linfáticos que tienen cáncer. Cuanto más grande es el número después de la N, más son los ganglios linfáticos que tienen cáncer.

MX: No puede medirse la metástasis.

M0: El cáncer no se ha diseminado a otras partes del cuerpo.

M1: El cáncer se ha diseminado a otras partes del cuerpo.

Hay algunos tipos de cánceres donde las combinaciones TNM se agrupan en cinco estadios menos detallados:

Estadio	Significado
Estadio 0	Hay células anormales presentes pero no se han diseminado al tejido cercano. Se llama también carcinoma in situ, o CIS. El CIS no es cáncer, pero puede convertirse en cáncer.
Estadio I, II, III	Hay cáncer presente. En cuanto más grande es el número, mayor es el tumor y tanto más se ha extendido en los tejidos cercanos.
Estadio IV	El cáncer se ha diseminado a partes distantes del cuerpo.

(Instituto Nacional del cáncer de los institutos nacionales de la salud de los Estados Unidos, 2016)

6.8 ¿Qué es el cáncer de mama?

El cáncer de mama se origina cuando las células en el seno comienzan a crecer en forma descontrolada. Estas células normalmente forman un tumor que a menudo se puede observar en una radiografía o se puede palpar como una protuberancia. El tumor es maligno si las células pueden crecer invadiendo los tejidos circundantes o propagándose a áreas distantes del cuerpo. El cáncer de seno ocurre casi exclusivamente en las mujeres, pero los hombres también lo pueden padecer.

Este tipo de cáncer puede originarse en diferentes partes del seno, aunque la mayoría comienza en los conductos que llevan la leche hacia el pezón (cánceres ductales). Algunos se originan en las glándulas que producen leche (cánceres lobulillares) y algunos menos

comunes comienzan en otros tejidos del seno (ver figura 2). Estos cánceres se llaman sarcomas y linfomas, y en realidad no se consideran cánceres de seno.

Aunque muchos tipos de cáncer de seno pueden causar una protuberancia en el seno, no todos lo hacen. Muchos se encuentran en mamogramas de detección que pueden mostrar los tumores en una etapa más temprana, a menudo antes de que puedan palparse y presenten síntomas.

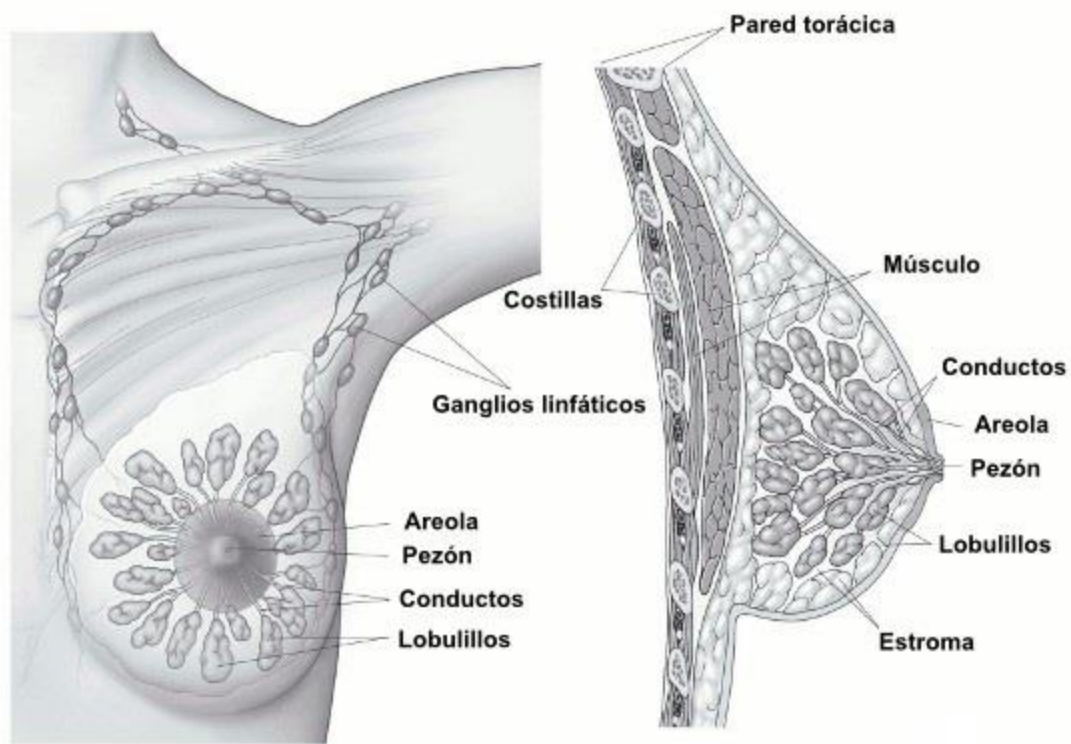


Fig.2. Morfología del seno (fuente: American Cancer Society).

El cáncer de seno se puede propagar cuando las células cancerosas alcanzan la sangre o el sistema linfático y llegan a otras partes del cuerpo. El sistema linfático es una red de vasos linfáticos que se encuentra en todo el cuerpo que conecta los ganglios linfáticos (células del sistema inmunitario). El líquido transparente dentro de los vasos linfáticos, llamado linfa,

contiene productos derivados de los tejidos y materia de desecho, así como células del sistema inmunitario. Los vasos linfáticos transportan líquido linfático fuera de los senos. En el caso de cáncer de seno, las células cancerosas pueden ingresar en los vasos linfáticos y comenzar a crecer en los ganglios linfáticos. En la figura 3 puede apreciarse la distribución de los ganglios linfáticos. La mayoría de los vasos linfáticos del seno drenan hacia:

- Los ganglios linfáticos localizados debajo del brazo (ganglios axilares).
- Los ganglios linfáticos que rodean la clavícula (ganglios linfáticos supraclaviculares e infraclaviculares).
- Los ganglios linfáticos (mamarios internos) en el interior del tórax y cerca del esternón.

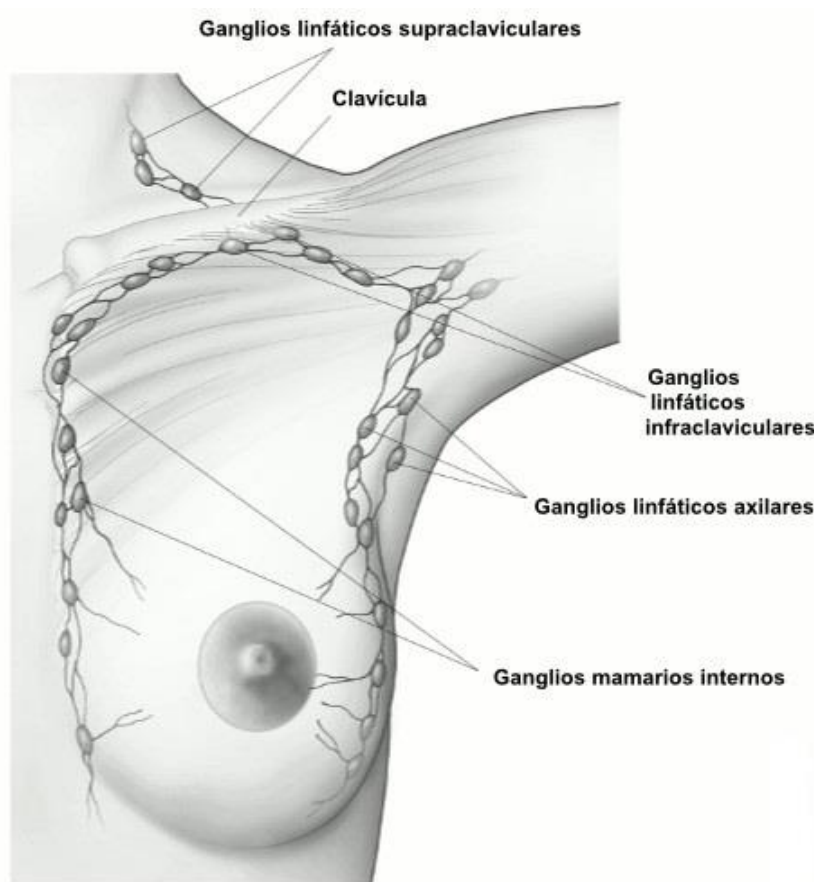


Fig.3. Morfología del seno (fuente: American Cancer Society).

Si las células cancerosas se han propagado a los ganglios linfáticos hay una mayor probabilidad de que se hayan desplazado a otras partes de su cuerpo. Mientras más ganglios linfáticos haya con células cancerosas del seno, mayor es la probabilidad de encontrar cáncer en otros órganos (American Cancer Society, 2019).

6.9 ¿Existe predisposición al cáncer de mama?

Actualmente se conocen varios genes que predisponen al cáncer de mama, dos de los cuales son BRCA1 y BRCA2, descritos como genes de cáncer de mama y ovario hereditarios. Ambos genes son supresores tumorales que comúnmente tienen la función de controlar el crecimiento y la muerte celular. Las proteínas BRCA1 y BRCA2 participan en la reparación del ADN y la recombinación homóloga, y probablemente en otros procesos celulares. El BRCA1 es un gen localizado en el cromosoma 17, involucrado en la regulación del ciclo celular, reparación del ADN dañado, mantenimiento de la estabilidad genómica y regulación de la transcripción. Las mutaciones de BRCA-1 están asociadas a la aparición de cáncer de mama en mujeres entre 40 y 50 años y también con el riesgo de padecer otros tumores como el de ovario. El gen que codifica la proteína BRCA-2 se aisló en el cromosoma 13. BRCA-2 está asociado a la aparición de cáncer a edades más avanzadas, entre 60 y 70 años, y en la población en general predispone a padecer cáncer de mama masculino, de ovario, vejiga, próstata y páncreas. Otros factores de riesgo para cáncer de mama son: historia familiar de cáncer de mama, ovario o endometrio, antecedentes personales de cáncer de mama, edad avanzada, el exceso de exposición a la radiación ionizante, el consumo de alcohol, así como el inicio temprano de la menstruación, nuliparidad o el primer parto retrasado, lactancia de corta duración, la menopausia tardía, obesidad posmenopáusica, extendido uso de

anticonceptivos orales, y el reemplazo de estrógeno prolongados (Karla Guzmán Santos, 2012).

6.10 Tipos de cáncer de mama

De acuerdo con la clasificación de la Organización Mundial de la Salud (OMS) existen dos tipos principales de cáncer de mama:

1. Carcinoma ductal (el más frecuente) que comienza en los ductos que llevan leche desde la mama hasta el pezón.
2. Carcinoma lobulillar que comienza en los lobulillos que producen la leche materna.

A su vez estos dos cánceres pueden clasificarse de la siguiente manera:

- In situ: no ha atravesado las uniones entre los tejidos.
- Invasivo o infiltrante: invade tejidos subyacentes vertiendo células cancerígenas a la sangre o linfa generando metástasis a través del cuerpo (Karla Guzmán Santos, 2012).

6.11 ¿Qué es el Carcinoma Ductal Invasivo o CDI?

El carcinoma ductal invasivo (CDI), a veces denominado carcinoma ductal infiltrante, es el tipo más común de cáncer de mama. Alrededor del 80% de todos los casos de cáncer de mama son carcinomas ductales invasivos.

Invasivo significa que el cáncer ha “invadido” o se ha propagado hacia los tejidos mamarios que lo rodean. Ductal significa que el cáncer comenzó en los conductos lácteos, que son las “tuberías” que transportan la leche desde los lobulillos productores de leche al pezón. Carcinoma hace referencia a todo cáncer que se origina en la piel o en otros tejidos que revisten órganos internos, como el tejido mamario. En conjunto, “carcinoma ductal invasivo” hace referencia al cáncer que ha atravesado la pared del conducto lácteo y ha comenzado a

invadir los tejidos de la mama. Con el tiempo, el carcinoma ductal invasivo puede propagarse hacia los ganglios linfáticos y posiblemente a otras áreas del cuerpo.

Aunque este carcinoma puede afectar a mujeres de cualquier edad, resulta más común a medida que la mujer envejece. Según la Sociedad Americana del Cáncer, aproximadamente dos tercios de las mujeres que son diagnosticadas con cáncer de mama invasivo tienen 55 años o más. El carcinoma ductal invasivo también afecta a los hombres (Breastcancer.org, 2014).

En la figura 4 se puede observar la evolución del CDI en el ducto de leche.

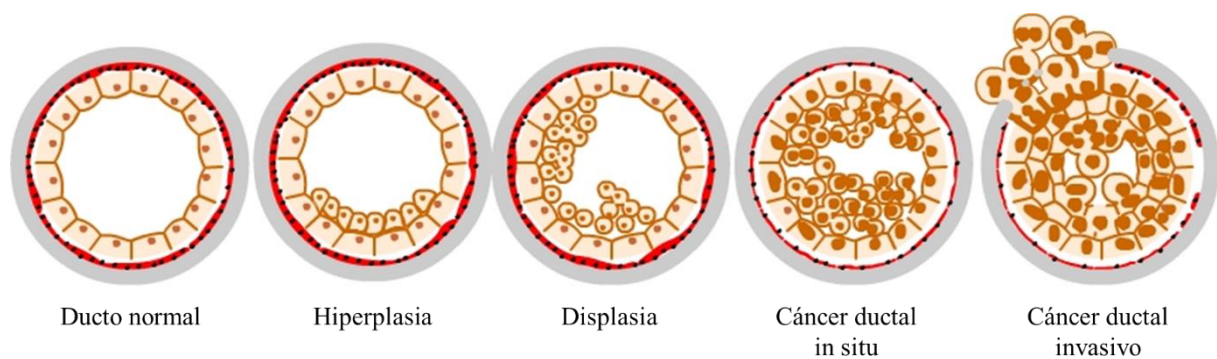


Fig.4. Evolución del CDI en el ducto de leche (fuente: rnceus.com).

En las siguientes dos imágenes histológicas (figuras 5 y 6) se puede apreciar la diferencia entre un cáncer ductal in situ (contenido) y cáncer ductal invasivo (infiltrante). En el primero, las células cancerosas aún no han traspasado las paredes del ducto y por tanto no hay metástasis. En la segunda, ya las células han infiltrado los tejidos sanos fuera del ducto y el riesgo a hacer metástasis es alto.

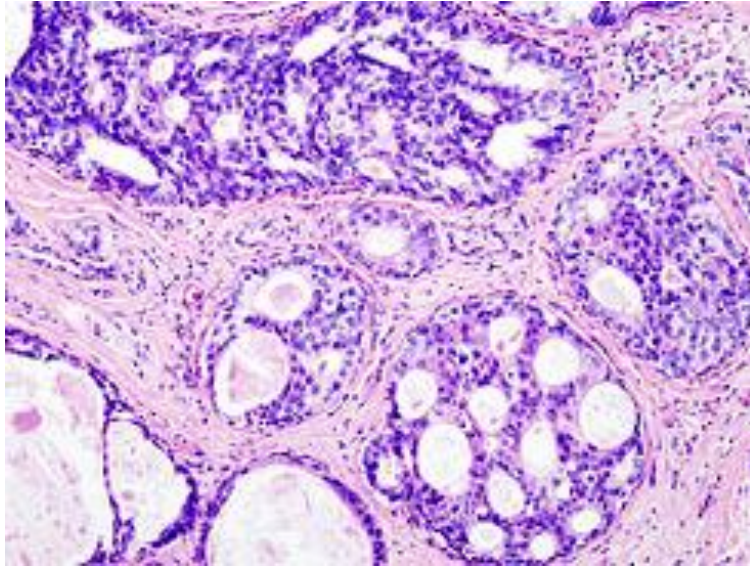


Fig.5. Cáncer ductal in situ (fuente: pathpedia.com).

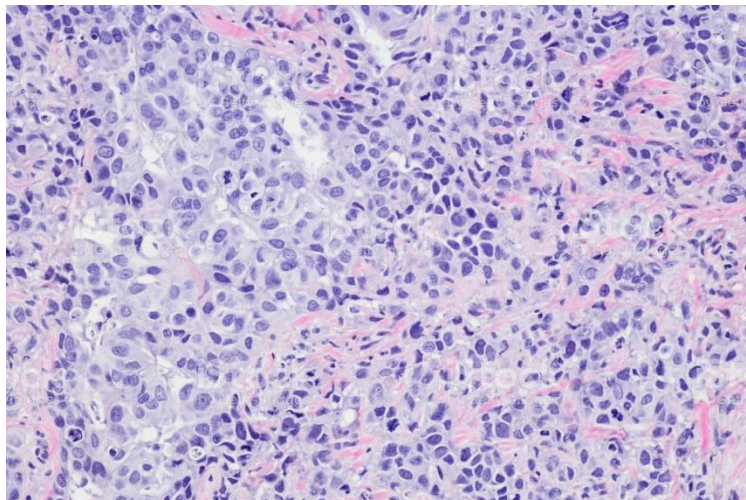


Fig.6. Cáncer ductal invasivo (fuente: pathpedia.com).

7. ¿QUÉ SON LAS REDES NEURONALES?

Las redes neuronales son un campo que ha evolucionado poderosamente en los últimos años y que tiene su origen en la neurobiología. Su principal característica es que intenta modelar algunos métodos básicos de procesamiento de la información que ocurren en el cerebro humano, y dado que este órgano desempeña tareas complejas, las redes neuronales también lo hacen.

Una definición práctica es que son una colección de neuronas artificiales interconectadas que aprenden de forma incremental a partir de su ambiente, que pueden capturar la esencia lineal o no lineal de los datos que reciben y son capaces de realizar predicciones confiables sobre nuevos entornos, aún con ruido o información parcial.

La neurona artificial es la unidad básica de cómputo que procesa datos localmente y es parte de una red global de otras neuronas. Estas neuronas forman una red de procesamiento paralelo cuya función está determinada por la forma en cómo está organizada, por la fuerza de conexión entre sus neuronas y el tipo de procesamiento de éstas.

Las redes neuronales artificiales realizan una variedad de tareas, tales como, predicción, aproximación de funciones, clasificación de patrones, agrupamiento de datos comunes en racimos, predicción de series temporales, entre otras (ver figura 7).

En ese sentido, las redes neuronales artificiales se asemejan al cerebro en dos aspectos:

1. El conocimiento es adquirido por la red a través de un proceso de aprendizaje.
2. La fuerza de interconexión entre las neuronas, conocida como peso sináptico, es usada para almacenar el conocimiento.

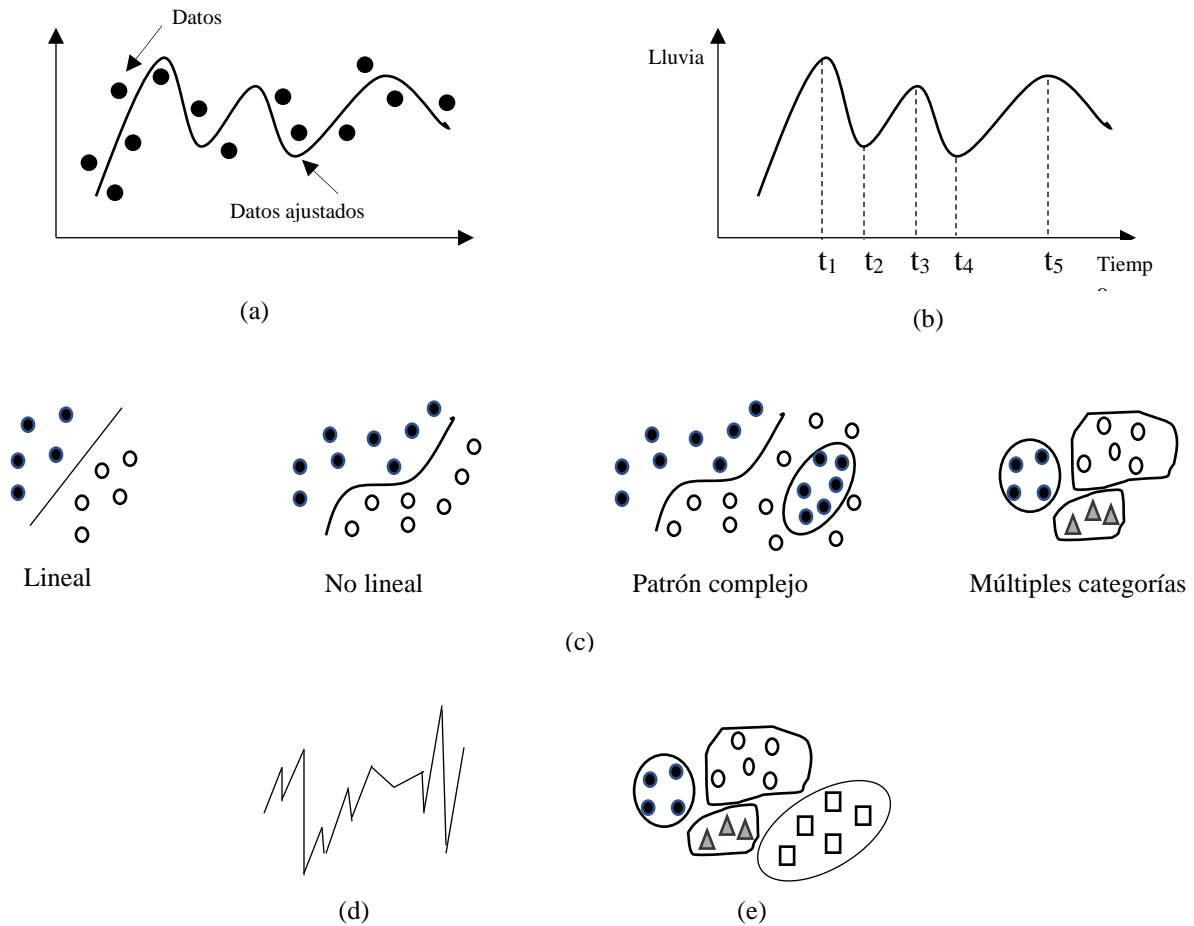


Fig.7. Algunas de las funciones más comunes que pueden realizar las redes neuronales artificiales: (a) aproximación de funciones, (b) predicción en series temporales, (c) clasificación de datos, (d) clasificación por señales, (e) agrupamiento en racimos (clustering) no supervisado (fuente: Sandhya Samarasinghe).

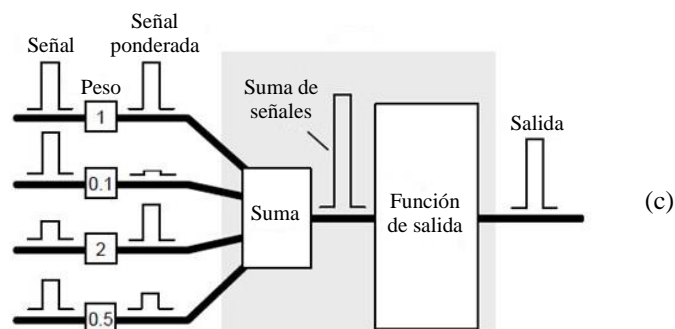
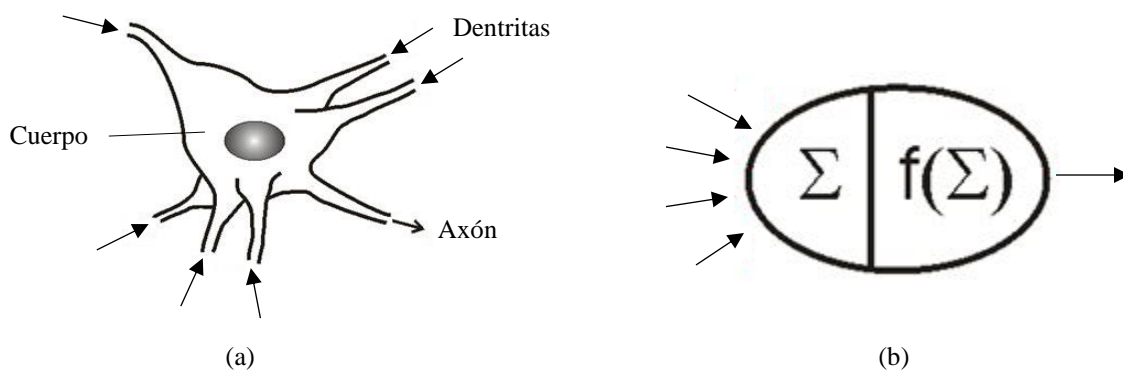
El cerebro recibe información de su entorno a través de los sentidos y los procesa para formar modelos internos que recrean el fenómeno externo. Esta información es gestionada localmente por las neuronas biológicas que forman grandes redes de procesamiento paralelo en la corteza prefrontal y luego envían una señal a miles de otras neuronas a las cuales están conectadas. Cuando este proceso de excitación neuronal se repite, aumenta la fuerza de conexión entre las neuronas y además se crean nuevas conexiones, lo cual deriva en la generación de una red masiva e interconectada donde se almacena la memoria. Se sabe que

la forma en cómo la memoria es almacenada depende de la “fuerza” de las conexiones entre neuronas.

La neurona biológica consta de tres partes principales: a) dentritas que reciben las señales de entrada y las ponderan de acuerdo con la fuerza de conexión; b) un cuerpo que acumula y procesa las señales de entrada; c) y un axón que transmite la señal de salida a las otras neuronas a las cuales está conectada.

Estas señales son eléctricas y generadas por la diferencia de concentraciones de iones de potasio (K^+) e iones de sodio (Na^+) que generan un potencial de acción el cual es transmitido de neurona en neurona a través de la liberación de neurotransmisores.

Este proceso biológico de cómputo es idealizado en un modelo matemático llamado neurona artificial (ver figura 8) que recibe las señales externas, las pondera y suma (Σ), y finalmente genera una señal de salida [$f(\Sigma)$].



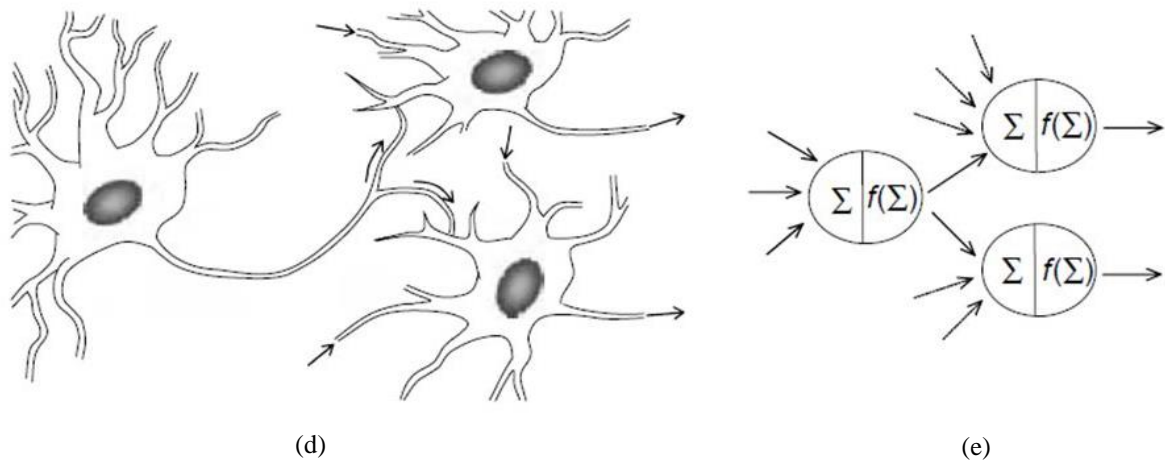


Fig.8. a) Neurona biológica, (b) neurona artificial, (c) proceso de cómputo en una neurona artificial, (d) red neuronal biológica, (e) red neuronal artificial (fuente: Sandhya Samarasinghe).

Las neuronas artificiales son estructuradas en forma de red y entrenadas para aprender por medio de un proceso de exposiciones repetidas a un conjunto de datos de entradas hasta que la red produce una salida correcta. Cada vez que la red recibe un conjunto de datos y los procesa, los pesos sinápticos entre neuronas son actualizados mediante un algoritmo cuya finalidad es lograr que toda la red genere la salida esperada. Los pesos de las neuronas son la forma en que como una red neuronal artificial almacena el conocimiento (imitando a las redes biológicas), y al proceso de ajustarlos se le conoce como aprendizaje.

Dado que una red neuronal artificial puede aprender, en particular ésta puede ser entrenada para que reconozca una imagen. Para ello, se la expone a múltiples fotografías de un objeto cualquiera para que actualice sus pesos sinápticos hasta que produzca la salida correcta: reconocimiento del objeto en cuestión. En la actualidad esta técnica está siendo utilizada para el diagnóstico de ciertos tipos de cáncer, tales como cáncer de piel (Andre Esteva, 2017) o cáncer de mama (Andrew Janowczyk, 2016).

En ciencias de la computación, una imagen es una matriz de números que corresponden a cada píxel de la imagen. Si la imagen es a color, existirán 3 matrices, una por cada canal de color: rojo, verde, azul (RGB por sus siglas en inglés). La mezcla de estos 3 canales produce toda la gama de colores que conocemos. Cada número describe la intensidad del color para un píxel, desde 0 (intensidad total del color) hasta 255 (ausencia total del color). Si la imagen es en blanco y negro, solo existirá una matriz de números (0 para el negro, 255 para el blanco, y el resto es una escala de grises).

Para hacer un uso más eficiente de estos datos numéricos, los modelos matemáticos de redes neuronales reorganizan la información de las matrices en un único arreglo de datos para luego procesarlos. A este procedimiento se le conoce como vectorización (ver figura 9):

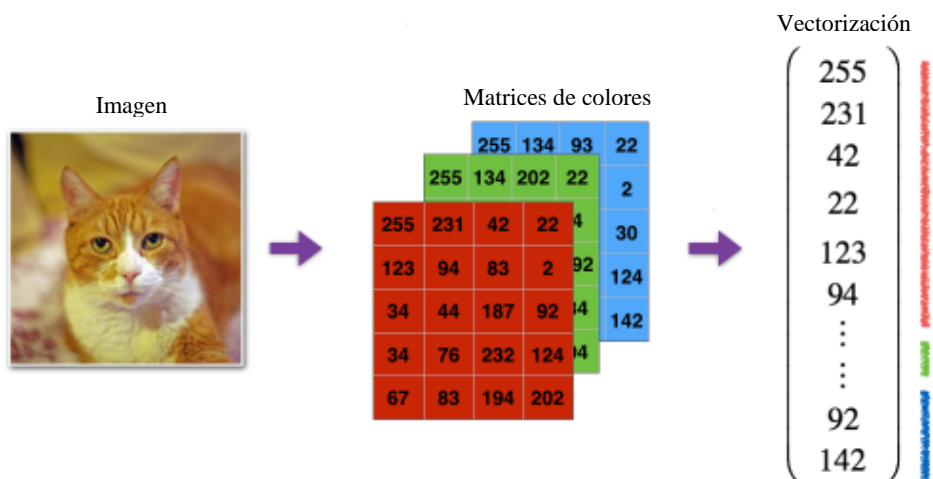


Fig.9. Proceso de vectorización de una imagen a colores a 3 canales RGB (fuente:Andrew Ng)

Si la imagen está en escala de grises se utiliza un sólo canal (matriz) de color como se indica en la figura 10:

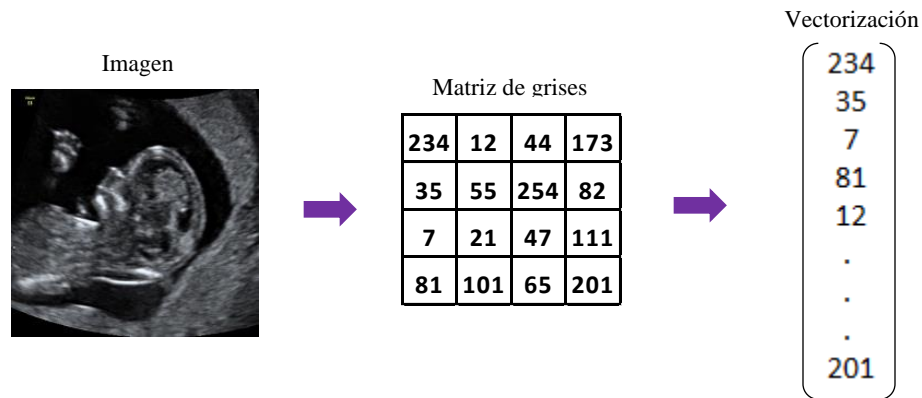


Fig.10. Proceso de vectorización de una imagen en escala de grises a 1 canal

Una vez que la data de cada imagen ha sido vectorizada, debe ser preprocesada para que la red neural trabaje mejor. A este proceso se le conoce como normalización o estandarización.

Finalmente, la red está lista para recibir la información y lo hace a través de unidades de entradas o de asociación (una por cada número del vector).

En general, las redes neurales artificiales se estructuran por capas de neuronas: la primera capa corresponde a las unidades de asociación que básicamente reciben la información (no son propiamente neuronas porque no realizan ningún procesamiento de la data). Luego están las capas internas compuestas por neuronas artificiales y los respectivos pesos sinápticos entre ellas. El número de capas internas y de neuronas depende de cuán grande y compleja sea la data de entrada; sin embargo, no existe una regla definitiva para calcular su número y depende de la pericia del modelador. Existe una heurística que sugiere utilizar tantas neuronas como $2/3$ de los datos de entrada. Por ejemplo, para imágenes en blanco y negro de 640×480 píxeles de resolución, se necesitarían 307.200 unidades de asociación (capa de entrada) y 204.800 neuronas en las capas internas. La capa final de la red es la de salida, que ofrece el resultado del procesamiento y puede contener una o varias neuronas dependiendo

del tipo de resultado que se quiera modelar. Para el caso particular de clasificación binaria, basta con una neurona de salida que producirá un cero (0) para una clase o un uno (1) en caso contrario. En la figura 11 puede apreciarse un ejemplo de red neuronal y sus capas:

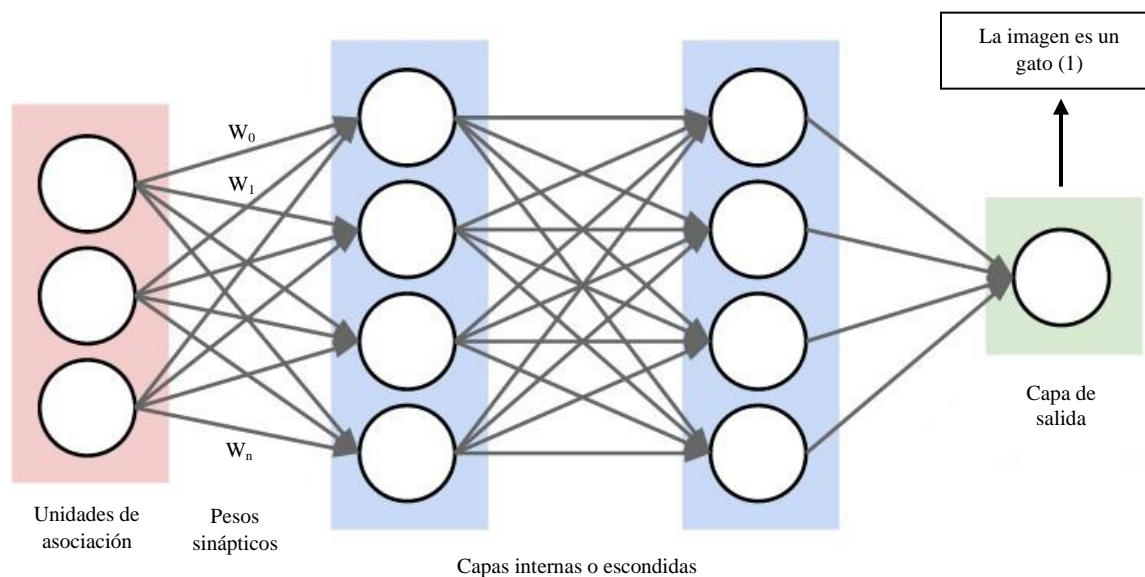


Fig.11. Simplificación de la estructura de una red neuronal con dos (2) capas internas de neuronas y una (1) neurona en su capa de salida.

Por ejemplo, para el entrenamiento de la red en este proyecto, se necesita conocer de antemano a qué categoría pertenece cada imagen histológica (con CDI o normal). Durante el proceso de aprendizaje la red recibe como insumo la imagen con su respectiva categoría y de esta forma puede actualizar sus pesos sinápticos. A este procedimiento se le conoce como aprendizaje supervisado.

Existen dos variables importantes que inciden directamente en la capacidad de predicción o clasificación de una red neuronal: la cantidad de data de entrenamiento (imágenes en el caso

de este trabajo) y el tamaño de la red (cantidad de capas internas). A mayor cantidad de cada una, mejor el desempeño de la red (ver figura 12).

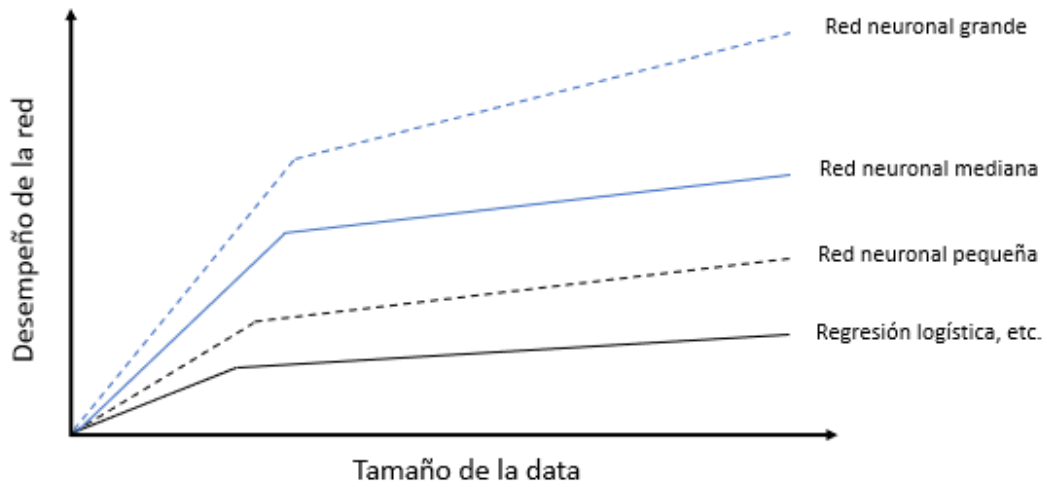


Fig.12. Dos variables afectan el desempeño de una red neuronal: a mayor cantidad de data de entrenamiento o del tamaño de la red, mejor será su desempeño.

8. MATEMÁTICAS DE LAS REDES NEURONALES

8.1 Red neuronal como clasificador binario

El modelo más simple de red neuronal se conoce como Unidad Lógica de Umbral (TLU por sus siglas en inglés) y fue propuesto por McCulloch-Pitts en 1940 como una aproximación muy sencilla de una neurona biológica. Utiliza una función umbral para convertir entradas reales en una única salida con valores cero (0) o uno (1). Esta característica convierte a este modelo en un clasificador. La idea principal es clasificar correctamente los patrones de entrada en dos categorías (0 o 1).

Para ejemplificar el modelo, se usarán dos entradas (x_1 y x_2) que serán ponderadas con los respectivos pesos de las unidades de asociación de la siguiente manera:

$$\Sigma = u = w_1x_1 + w_2x_2$$

Para simplificar aún más la explicación, se asumirán $w_1 = w_2 = 1$, por lo que $u = x_1 + x_2$. Se usarán los siguientes valores (tabla 1) de entrada asociados a sus respectivas categorías:

X1	X2	Categoría
0.2	0.4	0
0.2	0.7	0
0.8	0.1	0
1.0	0.7	1

Tabla 1. Datos de entradas

Asumiremos arbitrariamente un umbral de 1.5, por lo que la función de activación de la neurona se define de la siguiente manera:

$$f(\Sigma) = y = \begin{cases} 0 & \text{si } u < 1.5 \\ 1 & \text{si } u \geq 1.5 \end{cases}$$

En la figura 13 se puede ver la función umbral o de activación:

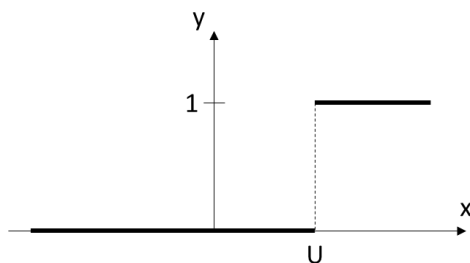


Fig.13. Vista gráfica de la función umbral o de activación. También se le conoce como función escalón.

Utilizando la neurona como un clasificador obtenemos los siguientes resultados (ver tabla 2):

Entradas	U	Y
(0.2, 0.4)	0.6	0
(0.2, 0.7)	1.0	0
(0.8, 0.1)	0.8	0
(1.0, 0.7)	1.7	1

Tabla 2. Clasificación del TLU

Se puede observar que la neurona clasificó correctamente los datos de entradas en sus respectivas categorías. Considerando el umbral utilizado $u=1.5$, se puede reformular de la siguiente manera:

$$u = x_1 + x_2 = 1.5 \quad \Rightarrow \quad x_2 = 1.5 - x_1$$

Graficando los datos de entradas y la línea recta $x_2 = 1.5 - x_1$ (ver figura 14):

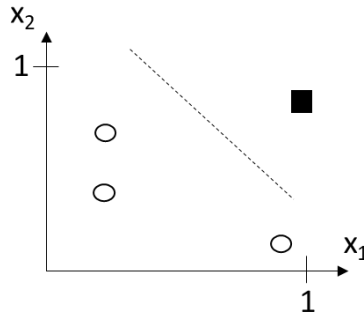


Fig.14. Línea recta que delimita los patrones de entradas y los clasifica geoméricamente

La línea recta que separa los cuatro patrones de entrada en realidad está delimitando dos espacios: uno por cada categoría (0 y 1), y lo hace linealmente.

Generalizando para dos entradas x_1, x_2 y un umbral θ se tiene la siguiente relación:

$$w_1x_1 + w_2x_2 = \theta$$

Despejando w_1x_1 obtenemos:

$$w_2x_2 = -w_1x_1 + \theta$$

Y despejando x_2 :

$$x_2 = -\left(\frac{w_1}{w_2}\right)x_1 + \left(\frac{\theta}{w_2}\right)$$

cuya forma general es:

$$x_2 = ax_1 + b$$

donde a y b son constantes. Esta ecuación describe una línea recta con pendiente a e intercepto b . Esta recta siempre va a poder separar cualquier par de entradas x_1 , x_2 de la neurona, lo cual convierte al modelo en un clasificador lineal binario. A esta recta se le conoce con el nombre de línea de decisión.

Este resultado puede generalizarse para cualquier número arbitrario n de entradas a la neurona. Para el caso de $n = 3$, ya no será una línea de decisión la que separe los patrones de entradas en categorías, sino un plano de decisión. Para $n > 3$, se denomina hiperplano de decisión.

Otra consecuencia de este análisis es que cualquier clasificación binaria que no pueda separarse a través de una superficie de decisión, no puede ser llevada a cabo por una TLU.

Adicionalmente, dado que los pesos de la neurona no se actualizan en ningún momento, los TLU no tienen capacidad para aprender.

8.2 ¿Cómo aprenden las redes neuronales?

En 1949, el psicólogo canadiense Donald Hebb propuso un mecanismo de aprendizaje para las neuronas, el cual consiste en actualizar los pesos entre ellas, lo que se conoce actualmente como aprendizaje Hebbiano. Sus tres postulados son los siguientes:

1. La información en una red se guarda en forma de pesos en las conexiones neuronales.
2. El cambio del peso entre dos neuronas x , y es proporcional al producto de sus valores de activación, es decir, a sus salidas.
3. El peso entre dos neuronas x , y se incrementa cuando se repite continuamente la activación entre ellas, es decir, la exposición repetida a un estímulo produce una mayor fuerza de conexión neuronal, por lo tanto, genera aprendizaje.

Estos postulados permiten formular matemáticamente el proceso del aprendizaje neuronal. Sean dos neuronas con valores de activación x , y , entonces si x excita (inhibe) a y , la fuerza de la conexión entre ellas aumenta (disminuye). Por lo tanto, el cambio del peso Δw es proporcional al producto de x e y . Esto puede expresarse de la siguiente manera:

$$\Delta w \propto x \cdot y$$

La proporcionalidad también puede escribirse de la siguiente manera:

$$\Delta w = \beta x \cdot y$$

El proceso de actualización del peso entre las neuronas puede expresarse a través de la siguiente ecuación:

$$w_{i+1} = w_i + \Delta w = w_i \pm \beta x \cdot y$$

donde β se denomina la tasa de aprendizaje, que es un valor real entre $[0,1]$ y determina la velocidad de aprendizaje.

Aplicando el principio Hebbiano, en 1958 el psicólogo estadounidense Frank Rosenblatt creó el primer modelo de neuronal artificial capaz de aprender, el cual era en esencia, un conjunto de TLU's que utilizaban el proceso de actualización de pesos. A este modelo lo llamó Perceptrón.

Rosenblatt también introdujo dos nuevos conceptos para complementar el principio de aprendizaje Hebbiano: aprendizaje supervisado y no supervisado.

- a) El aprendizaje supervisado utiliza un agente externo que asigna a cada patrón de entrada de la neurona una salida. Básicamente, se le presenta al perceptrón una salida

correcta para la entrada dada, y la red se encarga de ajustar sus pesos hasta que la diferencia entre el valor de activación calculado y la salida correcta sea aceptable.

- b) El aprendizaje no supervisado no utiliza agentes externos que presenten valores de salida para que la red haga los ajustes. A través de un proceso llamado aprendizaje competitivo, la red es capaz de ajustar sus pesos por sí misma.

El perceptrón más sencillo es un solo TLU que recibe múltiples entradas y para cada una de ellas debe existir una salida (ver figura 15). A las parejas de entrada-salida se le conoce como conjunto de entrenamiento. Los valores iniciales de los pesos asociados a cada entrada se escogen aleatoriamente. Exponer al perceptrón al conjunto de entrenamiento para que los pesos se ajusten y el modelo pueda aprender apropiadamente se llama entrenar una red neuronal.

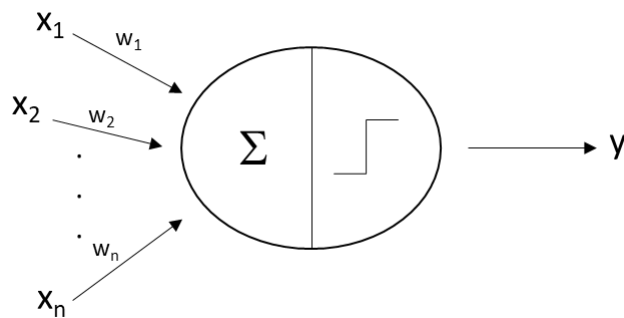


Fig.15. Perceptrón compuesto de una sola neurona

El algoritmo para entrenar a un perceptrón es el siguiente:

1. Los pesos se inicializan aleatoriamente.
2. El primer patrón de entrada es presentado y las entradas son ponderadas con sus correspondientes pesos.

3. La suma ponderada de pesos y entradas es pasada a la función de activación (función escalón con el umbral en el origen) para generar una salida.
4. Si la suma ponderada es mayor o igual a 0, entonces la salida 1, sino es 0.
5. La salida producida por la función de activación es comparada con la salida esperada. Si la clasificación producida por la neurona es errada, se ajustan los pesos del perceptrón de acuerdo con la ecuación de aprendizaje hebbiano hasta que se produzca la salida correcta.
6. Todo el proceso se repite hasta que el conjunto de entrenamiento completo se haya presentado y el aprendizaje haya culminado.

Este algoritmo puede verse matemáticamente de la siguiente manera: se tienen n entradas x_1, x_2, \dots, x_n , el cual representa un vector de entrada X de la forma (x_1, x_2, \dots, x_n) . Su correspondiente vector de pesos W es (w_1, w_2, \dots, w_n) . La suma ponderada se calcula

$$u = w_1x_1 + w_2x_2 + \dots + w_nx_n$$

La función de activación (escalón) que produce la salida y :

$$y = \begin{cases} 0 & \text{si } u < 0 \\ 1 & \text{si } u \geq 0 \end{cases}$$

El error entre la salida producida y la salida esperada se calcula:

$$Error = t - y$$

Donde t es la salida esperada. Los pesos se ajustan de la siguiente manera:

$$w_{i+1} = w_i + \beta \cdot X \cdot Error$$

Existen tres posibles alternativas para el ajuste de los pesos:

$$w_{i+1} = \begin{cases} w_1 & E = 0 & (t = y) \\ w_1 + \beta X & E = 1 & (t = 1, y = 0) \\ w_1 - \beta X & E = -1 & (t = 0, y = 1) \end{cases}$$

Si la tasa de aprendizaje es grande, entonces el entrenamiento es más rápido. Sin embargo, no necesariamente se logra la solución óptima, pues se podría oscilar alrededor de este valor sin llegar nunca a él. Por otro lado, si la tasa es muy pequeña, el entrenamiento puede hacerse muy lento.

Al finalizar el algoritmo, el vector de pesos W siempre es perpendicular a la línea de decisión que clasifica las entradas en categorías: si se tiene un vector de entrada $X = (x_1, x_2)$ y un vector de pesos $W = (w_1, w_2)$, entonces la suma ponderada es $u = w_1 x_1 + w_2 x_2$ que puede expresarse también en notación vectorial como $u = \|X\| \|W\| \cos(\theta)$, donde θ es el ángulo entre los dos vectores. Dado que el umbral es $u = 0$, entonces $u = \|X\| \|W\| \cos(\theta) = 0$.

Como $\|X\|$ y $\|W\|$ no son vectores nulos, entonces $\cos(\theta)$ tiene que ser igual a cero (0), por lo cual $\theta = \pm 90^\circ$. Luego, cualquier valor entre -90° y 90° entre los vectores X y W , generan un valor de activación $u \geq 0$ y por consiguiente $y = 1$, es decir, la única forma de que el vector X se clasifique en la categoría 1 es que θ no sobrepase los valores $\pm 90^\circ$. Si por el contrario el ángulo entre los vectores de entrada y peso sobrepasan los $\pm 90^\circ$, entonces $u < 0$ y se produce $y = 0$. Se puede concluir entonces que la línea de decisión debe ser perpendicular al vector W para no queden datos de entrada mal clasificados. Esto también quiere decir que la línea de decisión es única y, por lo tanto, la solución encontrada por el algoritmo es óptima.

8.3 Red neuronal como clasificador múltiple

Un perceptrón también puede ser utilizado como un clasificador para problemas que requieran múltiples categorías. Para ello, se utilizan tantas neuronas de salida (cuyos valores finales serán 0 o 1) como categorías se requiera clasificar (ver figura 16). El proceso de aprendizaje es el mismo que para el perceptrón simple.

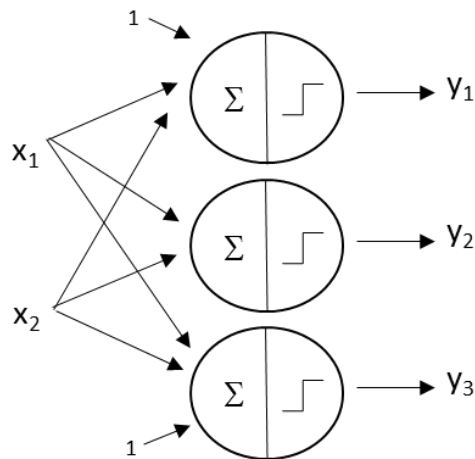


Fig.16. Perceptrón con múltiples salidas. Cada neurona es una categoría (fuente: Sandhya Samarasinghe)

En el ejemplo, el modelo neural clasifica tres categorías diferentes: (1,0,0), (0,1,0), (0,0,1). También se le añadió a cada neurona un peso adicional inicializado en 1 llamado sesgo. Cada neurona define una línea de decisión (ver figura 17) que separa cada clase de las demás.

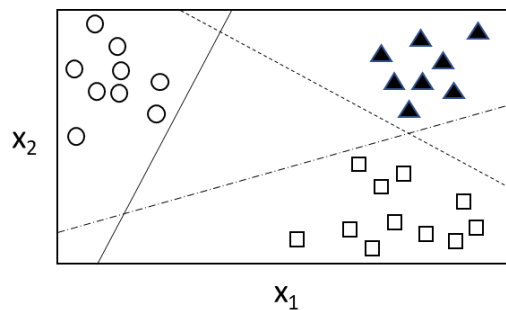


Fig.17. Líneas de decisión en un perceptrón con 3 salidas (fuente: Sandhya Samarasinghe).

Las ecuaciones correspondientes a cada línea de decisión quedarían de la siguiente manera:

$$b_1 + w_1^1 x_1 + w_2^1 x_2 = 0$$

$$b_2 + w_1^2 x_1 + w_2^2 x_2 = 0$$

$$b_3 + w_1^3 x_1 + w_2^3 x_2 = 0$$

Cabe destacar que, si la data inicial no está agrupada de forma lineal, el perceptrón no será capaz de clasificarla correctamente.

8.4 Red neuronal como predictor

Una red neural vista como clasificador solo tiene valores de salida 0 y 1, o combinaciones de ellos en caso de que existan varias salidas y_i . Por ejemplo: (1,0,0), (0,1,0,0), etc.

Es posible también que la red se comporte como un predictor, y para ello, es necesario linealizar sus neuronas eliminando la función de activación, pues de esta manera, la salida de la neurona tiene la siguiente forma:

$$u = \sum_{i=1}^n (w_i x_i + w_0) = w_1 x_1 + w_2 x_2 + \dots + w_n x_n + w_0$$

Donde w_0 corresponde al sesgo de la neurona y tiene valor igual a 1.

Luego, en lugar de usar la función de activación escalón (que es no lineal), hacemos $y = u$ y obtenemos la ecuación:

$$y = w_1 x_1 + w_2 x_2 + \dots + w_n x_n + w_0$$

que representa un modelo lineal. De hecho, es una regresión lineal y la salida de la neurona ya no se restringe a 0 y 1, sino que puede asumir cualquier valor real (ver figura 18).

En el caso de que la red tenga múltiples entradas y salidas, el modelo final es un sistema de ecuaciones lineales:

$$y_i = w_i^1 x_1 + w_i^2 x_2 + \dots + w_i^n x_n + w_0$$

donde y_i representa el i -ésimo valor aproximado y w_i^j es el peso correspondiente a la i -ésima entrada y la j -ésima variable dependiente.

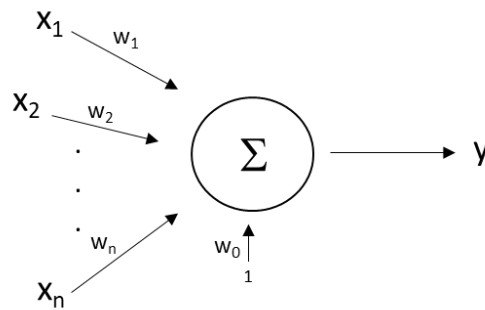


Fig.18. Modelo de neurona lineal. No usa función de activación (fuente: Sandhya Samarasinghe).

Este tipo de modelo neural lineal fue desarrollado por los estadounidenses Widros y Hoff en 1960, y lo llamaron ADALINE (Adaptive Linear Neuron), pues tiene la capacidad de corregir a la función de error utilizada durante el proceso de aprendizaje. A esta propiedad se le conoce como Regla Delta o Descenso del Gradiente.

8.5 Regla Delta o Descenso del Gradiente

La Regla Delta básicamente lo que hace es minimizar el error cometido durante el ajuste de los pesos durante el proceso de entrenamiento. Consideremos una neurona lineal con una sola entrada:

$$u = w_1 x$$

$$y = u = w_1x$$

Luego, el error E cometido corresponde a: $E = t - y = t - w_1x$

Sin embargo, la Regla Delta hace uso del concepto de error cuadrático, que es preferible pues elimina los posibles signos negativos. De esta forma:

$$SE = \frac{1}{2}E^2 = \frac{1}{2}(t - w_1x)^2$$

Otra ventaja importante de considerar una función cuadrática es que tiene forma de parábola, por lo tanto, tiene un solo punto mínimo local que también es mínimo global. Esta característica es deseable pues nos evita considerar múltiples mínimos locales. El punto mínimo de una parábola es su vértice lo que hace que el peso óptimo esté localizado justo allí. La Regla Delta lo que propone es alcanzar este punto a partir de un peso inicial dado y avanzando en la dirección opuesta al gradiente de la función (ver figura 19). El gradiente se calcula derivando la función error cuadrático SE y nos da la dirección en la cual el error crece más rápidamente:

$$\frac{dSE}{dw_1} = \frac{2}{2}(t - y)(-x) = -Ex$$

Dado que necesitamos ir en dirección contraria a la cual el gradiente crece más rápidamente, entonces el cambio del peso debe ser proporcional al negativo del gradiente, es decir, $\Delta w \propto Ex$ que también se puede expresar como $\Delta w = \beta Ex$. Por lo tanto, los pesos se ajustan de la siguiente manera:

$$w_0^{i+1} = w_0^i + \beta E$$

$$w_j^{i+1} = w_j^i + \beta x_j E$$

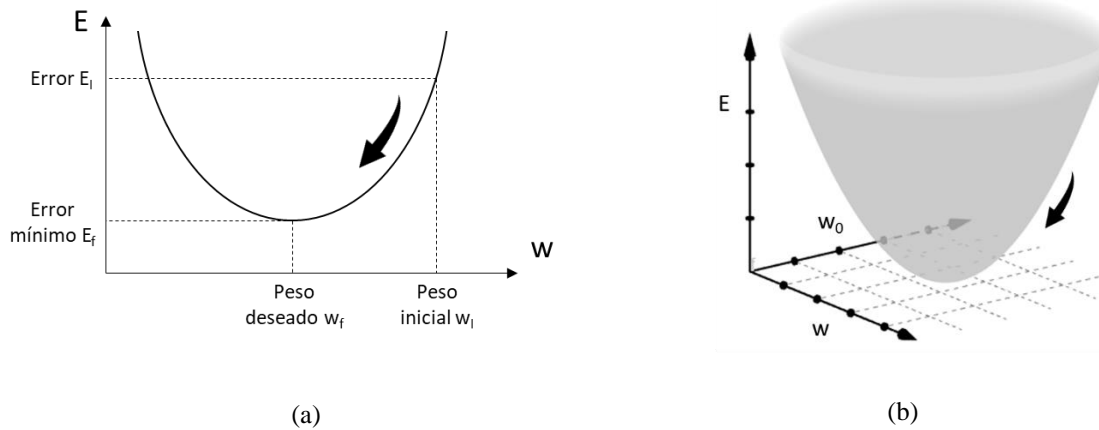


Fig.19. a) Descenso de gradiente de la función error en una dimensión: peso (W). b) Descenso del gradiente en la superficie error en dos dimensiones (peso W y umbral W_0)

8.6 Aprendizaje por lotes

En ocasiones, realizar el ajuste de los pesos para cada valor de entrada $\{x_1, x_2, \dots, x_n\}$ genera problemas en el entrenamiento. Esto se debe a que luego de ajustar los pesos para una entrada en particular x_i , el siguiente ciclo de entrenamiento para la entrada x_{i+1} puede cancelar el ajuste anterior realizado. La consecuencia inmediata es que el valor de los pesos para el vector W empieza a oscilar alrededor del valor óptimo pero nunca lo alcanza, generando clasificaciones o aproximaciones erróneas.

Para evitar este problema es preferible esperar a que todos los patrones de entrada hayan sido procesados y luego realizar un solo ajuste promedio a los pesos. A este proceso se le llama Aprendizaje por Lotes. El objetivo final es reducir el error promedio sobre todos los patrones de entrada. A esta función se le conoce como Error Cuadrático Medio y puede ser expresada por sus siglas en inglés de la siguiente forma:

$$MSE = \frac{1}{2n} \sum_{i=1}^n E_i^2$$

donde E_i representa el error correspondiente al i -ésimo patrón, y n es el número total de entradas. La idea principal es obtener el error del gradiente para cada entrada, sumarlos y obtener el promedio al final de haber procesado todo el lote. Con este error promedio entonces se realiza el ajuste de los pesos usando la regla delta. Luego de derivar la función MSE, obtenemos el cambio que debe hacerse a cada peso luego de una corrida completa a todas las entradas:

$$\Delta w_1 = \beta \left(\frac{1}{n} \sum_{i=1}^n E_i x_i \right)$$

8.7 Reconocimiento de patrones no lineales

Los modelos neurales vistos sólo son capaces de clasificar o aproximar patrones y funciones linealmente separables. Sin embargo, en la vida real, los problemas más abundantes son de naturaleza no lineal, y tanto el perceptrón como las TLU's son ineficaces para estos casos.

En la década de 1980, los psicólogos americanos McClelland y Rumelhart desarrollaron una serie de conceptos que dieron origen al perceptrón multicapa, el cual es una extensión del perceptrón, pero cuya estructura se basa en una o más capas de neuronas entre las unidades de asociación (entradas) y las neuronas de salida. A estas capas se les llama capas escondidas. Además, las neuronas pertenecientes a la(s) capa(s) escondidas usan una función de activación estrictamente no lineal.

Este modelo multicapa es tan poderoso que puede aproximar cualquier tipo de relación no lineal entre las entradas y las salidas (Samarasinghe, 2007), y por tal motivo, son llamados Predictores Universales. Esta ventaja proviene de las capas escondidas y del procesamiento no lineal de las funciones de activación que usa.

Un perceptrón multicapa está compuesto de una capa de asociación con entradas x_1, x_2, \dots, x_n , una o varias capas escondidas y una o varias neuronas de salida (ver figura 20.b). Cada neurona de las capas intermedias realiza dos procesamientos: la suma ponderada de los pesos con sus respectivos datos de entrada (Σ) y luego una transformación de esta suma a través de la función de activación no lineal (σ) como se aprecia en la figura 20.a. Si el objetivo del modelo es clasificar datos en categorías, por lo general se utilizan varias neuronas de salida (incluso una). Si el modelo va a aproximar, se utiliza una sola neurona de salida. Las salidas de las capas escondidas son pasadas a la siguiente capa escondida (si es que existe), y finalmente, la última capa escondida alimenta a la(s) neurona(s) de salida, que también realiza la suma ponderada, pero su función de activación puede o no ser lineal. Este proceso de entrenamiento se realiza hasta que la diferencia entre el valor de salida generado por la red y el valor deseado sea menor a un error estipulado.

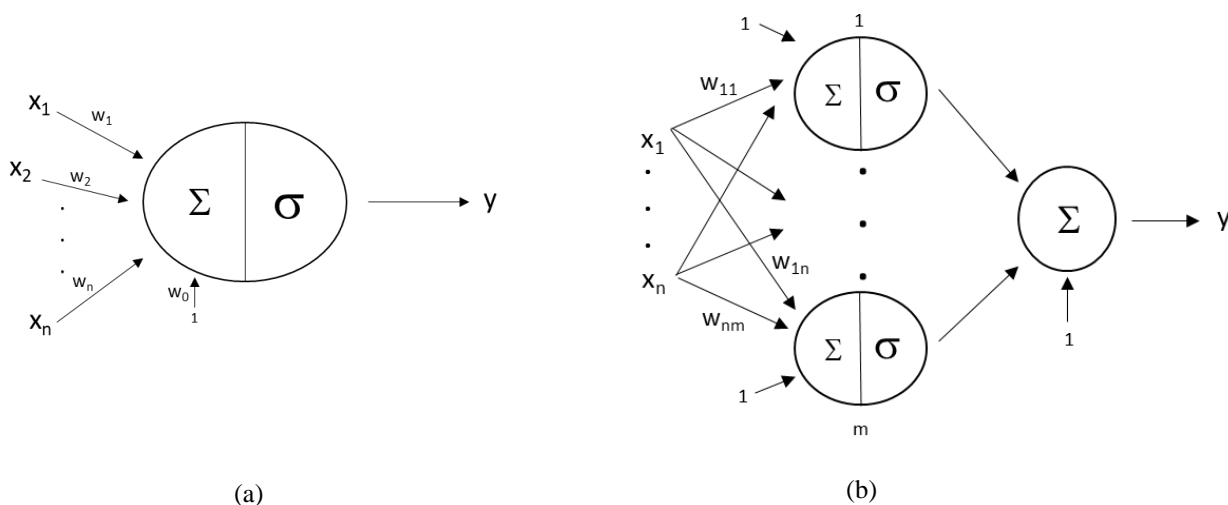


Fig.20. a) Neurona no lineal con función de activación σ . b) Ejemplo de perceptrón multicapa con n entradas, 1 capa escondida con m neuronas no lineales y una neurona lineal de salida (fuente: Sandhya Samarasinghe).

La salida de una neurona no lineal se calcula de la siguiente manera:

$$\sigma \left(\sum_{i=1}^n w_i x_i + b \right)$$

donde b es el sesgo. Luego existen varias funciones de activación (σ) comúnmente utilizadas para modelamiento no lineal: logística, tangente hiperbólica, relu, entre otras (ver figura 21).

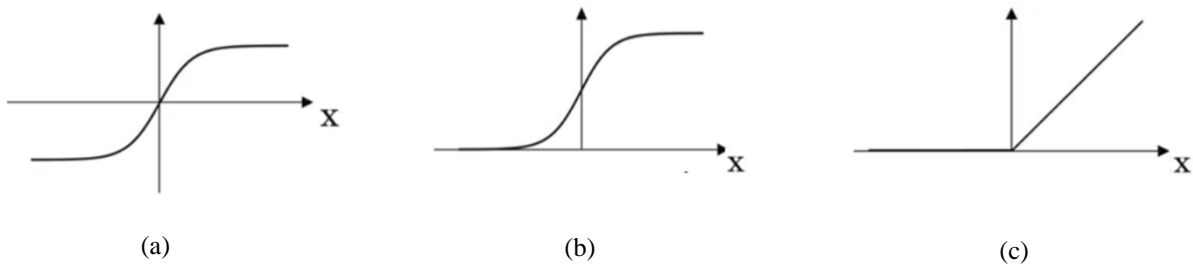


Fig.21. a) Tangente hiperbólica b) Sigmoide o logística c)Relu (Rectified Linear Unit)

$$a) y = \tanh x = \frac{1 + e^{-x}}{1 - e^{-x}}$$

$$b) y = \frac{1}{1 + e^{-x}}$$

$$c) y = \max(0, x)$$

Todas estas funciones tienen varias características deseables:

- a) Son no lineales.
- b) Intentan modelar el comportamiento de las neuronas biológicas.
- c) Son continuas.
- d) Son diferenciables.
- e) Están acotadas (casi siempre entre 0 y 1), excepto Relu.

Una propiedad importante para modelar la estructura de un perceptrón multicapa, es que cada neurona no lineal es capaz de modelar cualquier tipo de no linealidad siempre y cuando sea monótona (creciente o decreciente) (Samarasinghe, 2007).

8.8 ¿Cómo aprenden los perceptrones multicapa?

A medida que se entrena una red multicapa, las neuronas de las capas intermedias se organizan a sí mismas de tal modo que sus neuronas aprenden a reconocer distintas características del espacio total de entrada. Después del entrenamiento, cuando se les presente un patrón arbitrario de entrada que contenga ruido o que esté incompleto, las neuronas de las capas ocultas de la red responderán con una salida activa si la nueva entrada contiene un patrón que se asemeje a aquella característica que las neuronas individuales hayan aprendido a reconocer durante su entrenamiento.

Para lograr maximizar la eficacia de este proceso se utiliza un algoritmo llamado “Propagación hacia atrás”, que es un método de cálculo del gradiente para aprendizaje supervisado. El método emplea un ciclo propagación – adaptación de dos fases. Una vez que se ha aplicado un patrón a la entrada de la red como estímulo, éste se propaga desde la primera capa a través de las capas siguientes de la red, hasta generar una salida. La señal de salida se compara con la salida deseada y se calcula una señal de error para cada una de las salidas.

Las salidas de error se propagan hacia atrás, partiendo de la capa de salida, hacia todas las neuronas de la capa oculta que contribuyen directamente a la salida. Sin embargo las neuronas de las capas ocultas solo reciben una fracción de la señal total del error, basándose aproximadamente en la contribución relativa que haya aportado cada neurona a la salida original. Este proceso se repite, capa por capa, hasta que todas las neuronas de la red hayan recibido una señal de error que describa su contribución relativa al error total.

Usando como ejemplo la red neuronal de la figura 19, denotamos en forma general los errores de los pesos asociados a la neurona de salida como $\partial E/\partial b$ y los errores de los pesos respecto a las neuronas de la capa escondida como $\partial E/\partial a$. Dado que E (error cometido por la red respecto a la entrada x_i) no está directamente vinculado con b ni con a, se debe aplicar la regla de la cadena, pues E está directamente vinculado con z y éste con la sumatoria ponderada v, que sí depende de b. Por otra parte, v está vinculado con y (las salidas de las neuronas escondidas), que sí están directamente enlazadas con a (ver figura 22). Matemáticamente hablando, la utilización de la regla de la cadena para hallar la contribución relativa de cada peso al error cometido por la red es lo que se conoce como propagación hacia atrás.

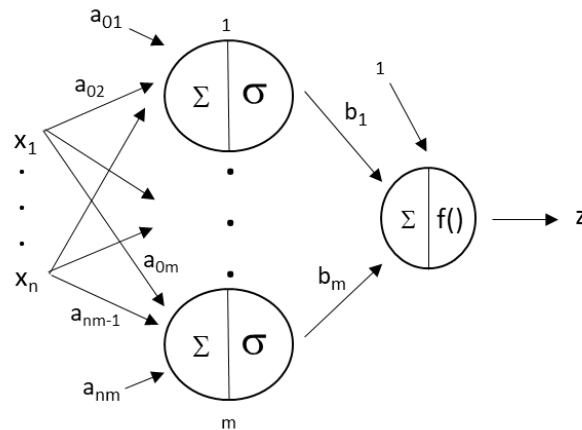


Fig.22. Perceptrón multicapa que será entrenado (fuente: Sandhya Samarasinghe).

De acuerdo con la regla de cadena, tenemos que el error cometido por b se calcula de la siguiente manera:

$$\frac{\partial E}{\partial b} = \frac{\partial E}{\partial z} \cdot \frac{\partial z}{\partial v} \cdot \frac{\partial v}{\partial b}$$

$$\text{Donde } \frac{\partial E}{\partial z} = z - t, \quad \frac{\partial z}{\partial v} = z(1 - z), \quad v = b_0 + b_1 y, \quad \frac{\partial v}{\partial b_1} = y, \quad \frac{\partial v}{\partial b_0} = 1$$

Por lo tanto,

$$\frac{\partial E}{\partial b_0} = (z - t)z(1 - z)$$

$$\frac{\partial E}{\partial b_1} = (z - t)z(1 - z)y$$

A su vez, el error cometido por a es:

$$\frac{\partial E}{\partial a} = \left(\frac{\partial E}{\partial z} \cdot \frac{\partial z}{\partial v} \cdot \frac{\partial v}{\partial y} \right) \cdot \frac{\partial y}{\partial u} \cdot \frac{\partial u}{\partial a}$$

$$\text{Donde } \frac{\partial v}{\partial y} = b_1, \quad \frac{\partial y}{\partial u} = y(1 - y), \quad \frac{\partial u}{\partial a_1} = x, \quad \frac{\partial u}{\partial a_0} = 1$$

Por lo tanto,

$$\frac{\partial E}{\partial a_1} = (z - t)z(1 - z)b_1 y(1 - y)x$$

$$\frac{\partial E}{\partial a_0} = (z - t)z(1 - z)b_1 y(1 - y)$$

A grandes rasgos el algoritmo de propagación hacia atrás se puede describir de la siguiente manera:

1. Calcular la salida de la red z_i a partir de uno de los conjuntos de valores de prueba x_i .
2. Comparar con la salida correcta t y calcular el error.
3. Calcular las derivadas parciales del error con respecto a los pesos de la red utilizando el concepto de la regla de la cadena.
4. Ajustar los pesos de cada neurona para reducir el error usando la regla delta.

5. Repetir el proceso varias veces por cada par de (entradas-salidas) de prueba.

8.9 Redes neuronales convolucionales (CNN)

Una red neuronal convolucional (CNN por sus siglas en inglés) es un tipo de red neuronal artificial donde las neuronas corresponden a campos receptivos de una manera muy similar a las neuronas en la corteza visual primaria de un cerebro biológico. Este tipo de red es una variación de un perceptrón multicapa, sin embargo, debido a que su aplicación es realizada en matrices bidimensionales, son muy efectivas para tareas de visión artificial, como en la clasificación y segmentación de imágenes, entre otras aplicaciones (Ciresan, Meier, Masci, Gambardella, & Schmidhuber, 2011).

Los perceptrones multicapas son capaces de clasificar y segmentar imágenes pero con una precisión relativamente baja en casos de alta complejidad gráfica, razón por la cual las CNN son ampliamente utilizadas para este fin (Saha, 2018).

La forma en cómo capturan la esencia espacial de una imagen es replicando el funcionamiento por capas de la corteza visual, donde las primeras reconocen características menos especializadas, y en la medida en que la información viaja hacia capas más profundas, se van detallando características más precisas. Este proceso se modela a través de filtros sucesivos que replican de manera simplificada las capas del cerebro (ver figura 23).

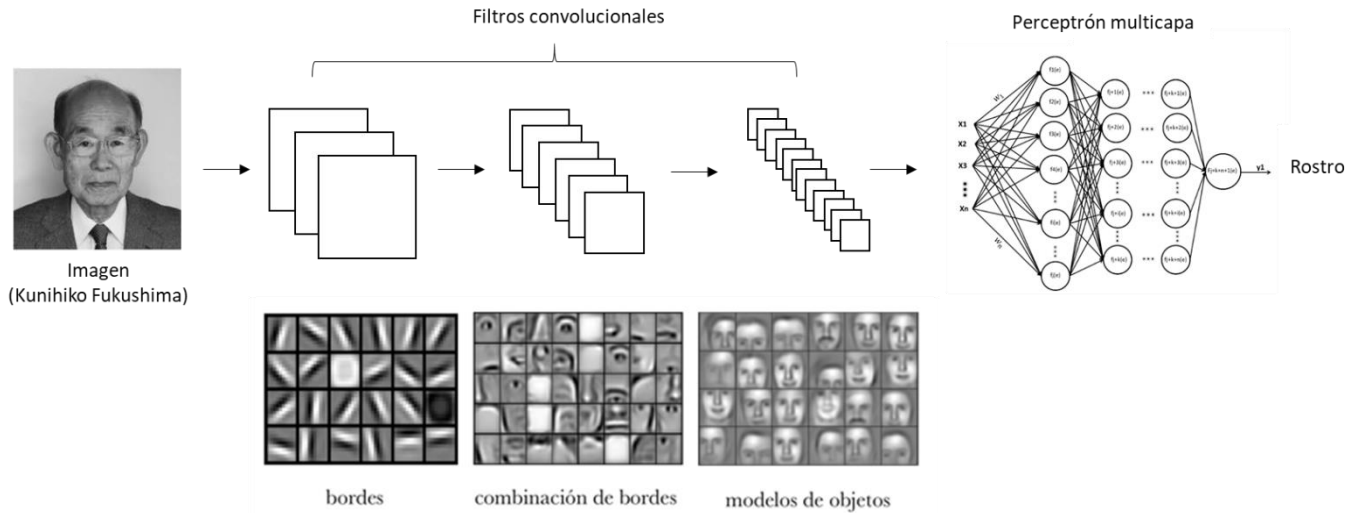


Fig.23. Modelo de una red convolucional con sus filtros y perceptrón multicapa clasificador.

El rol de una CNN es reducir la complejidad de las imágenes a una forma mucho más sencilla de procesar sin perder información necesaria para una buena clasificación.

Los filtros, también llamados kernels, son matrices bidimensionales que son operadas con la imagen de entrada o capa antecesora y producen una nueva matriz reducida con la información requerida según su especialización. La operación entre las matrices es un producto punto a punto y luego se suman todos los resultados (ver figura 24).

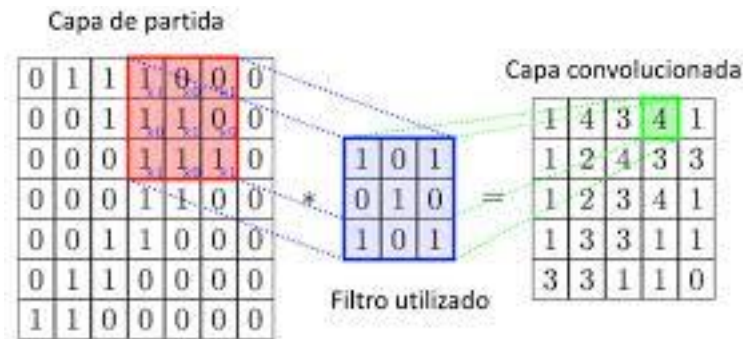


Fig.24. Ejemplo de un filtro y capa convolucional (fuente: Diego Calvo).

Para agregar no linealidad a la capa convolucional en ocasiones se suele usar una función de activación como ReLU luego del producto punto.

El filtro recorre toda la matriz antecesora para obtener la información completa incluyendo los canales de colores en caso de que existan (ver figura 25).

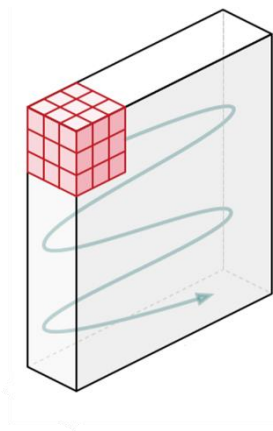


Fig.25.Filtro recorriendo los tres canales de colores (fuente: Francesco Visin Vincent Dumoulin).

Esta operación convolucional no se limita a una sola capa, aunque por convención, la primera capa es responsable de capturar las características menos especializadas de la imagen, tales como, bordes, color, orientación de los gradientes, sombras, entre otras. Sin embargo, al añadir más capas, se adapta a la red para que extraiga características más detalladas y pueda contener de forma distribuida toda la información de la imagen de la misma manera como lo hace nuestro cerebro.

Existe otro tipo de operación cuya única función es reducir la dimensión de la capa anterior. Se denomina capa de pooling y funciona de dos posibles formas (figura 26):

1. Max pooling: devuelve el valor máximo de la porción de la imagen sobre la que se posa el filtro.

2. Average pooling: devuelve el promedio de la porción de la imagen sobre la que se posa el filtro.

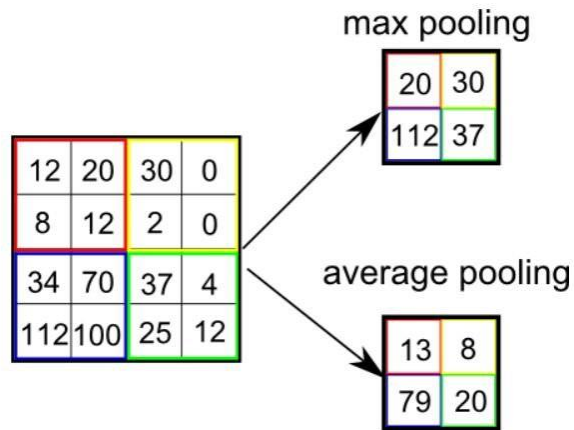


Fig.26. Tipos de pooling (fuente: Francesco Visin Vincent Dumoulin).

Dependiendo de la complejidad de la imagen varía el número de capas convolucionales y de pooling. A mayor número de capas, mayor la capacidad para capturar características muy especializadas, pero también aumenta el costo computacional.

Finalmente, después de haber extraído las características de la imagen, éstas son vectorizadas y pasadas a un perceptrón multicapa para efectos de clasificación.

Existen varios tipos de arquitecturas convolucionales ya estudiadas y predefinidas que ofrecen capacidades muy amplias para el reconocimiento de muchos tipos de imágenes.

Esas arquitecturas las podemos encontrar en las siguientes redes neuronales públicas: LeNet, AlexNet, VGGNet, GoogLeNet, ResNet, ZFNet.

9. DISEÑO E IMPLEMENTACIÓN DEL PROTOTIPO

Para preparar el prototipo es necesario conocer de antemano la cantidad de datos de entradas y dimensiones que se utilizarán para el entrenamiento de la red neuronal.

Las imágenes histológicas se obtuvieron del website Kaggle.com (<https://www.kaggle.com/paultimothymooney/breast-histopathology-images>). Kaggle es una comunidad científica que pertenece a Google LLC y permite a sus usuarios publicar grandes bases de datos, hacer uso de recursos académicos, ofrecer cursos en línea y programar competencias en el área de ciencia de datos.

La base de datos consta de 277.524 imágenes: 198.738 sin cáncer y 78.786 con cáncer, con un factor de aumento de 40x. Fueron curadas por los doctores Andrew Janowczyk y Anant Madabhushi del departamento de Ingeniería Biomédica de la Case Western Reserve University en Cleveland, USA.

Cada imagen tiene una dimensión de 50x50 píxeles, y en total, ocupan 5.8GB de espacio en disco. Dado que se necesita una cantidad considerable de memoria de acuerdo con estas dimensiones, todo el proceso de entrenamiento y validación posterior se realizó en la nube Azure de Microsoft, en una máquina virtual Linux (Ubuntu) preconfigurada con varias librerías de ciencia de datos, entre ellas: Keras, TensorFlow, PyTorch, Scikit-learn, Python, entre otras. El procesador utilizado fue una tarjeta gráfica standard NVIDIA K80 GPU y 8GB de memoria RAM.

Del total de imágenes, se tomó el 80% de estas para entrenamiento del prototipo. De esas imágenes de entrenamiento se tomó el 10% para validación de la red. El 20% restante del total de las imágenes se usó para pruebas.

Dado que este prototipo debe reconocer patrones gráficos de alta complejidad para el reconocimiento del CDI, se utilizó como base la arquitectura de la red neuronal convolucional VGGNet 14 (ver figura 27) que consta de 21 capas: 18 capas pertenecientes al modelo convolucional y 3 capas de redes neurales convencionales totalmente conectadas.

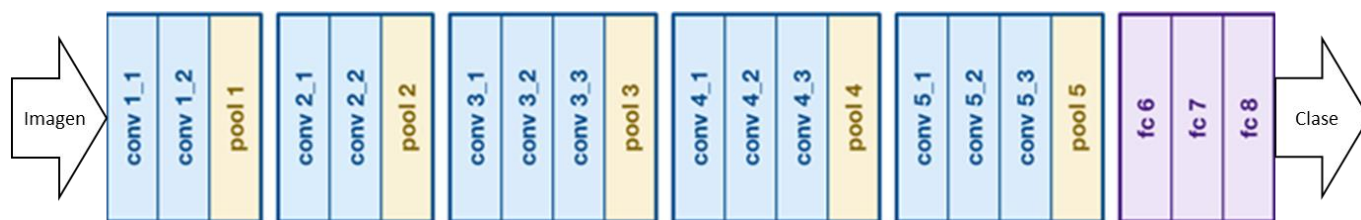


Fig.27. Arquitectura VGGNet 14 con 21 capas (fuente: GitHub).

Cabe resaltar que VVGNet utiliza como entrada imágenes de $224 \times 224 \times 3$ y hace 5 procesos de maxpooling. Sin embargo, las imágenes de la base de datos de CDI son de $50 \times 50 \times 3$ y no son compatibles con esta arquitectura (ver figura 28).

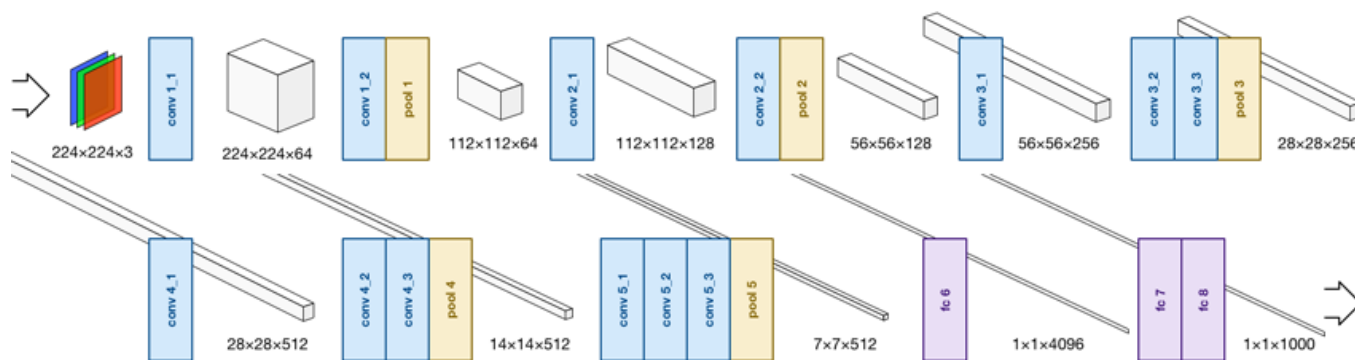


Fig.28. Arquitectura en detalle de VGGNet 14 con entradas de $224 \times 224 \times 3$ (fuente: GitHub).

Por tal motivo, se realizó una adaptación a una arquitectura compatible con las características de las imágenes que se utilizarán para el entrenamiento (ver figura 29).

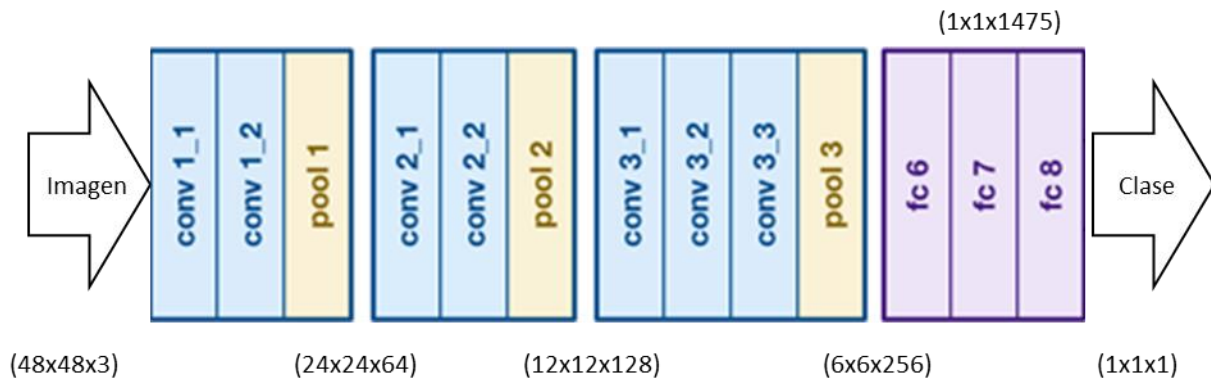


Fig.29. Arquitectura final del modelo CDI con entradas de 48x48x3.

Las imágenes se redimensionaron a 48x48 por canal ya que hay 3 procesos de Maxpooling que reducen las imágenes a la mitad y no harían factible el flujo con las dimensiones originales de 50x50: $50/2=25$ píxeles en el primer pooling y luego $25/2=12.5$ píxeles en el segundo pooling. Para lograr esta reducción, se eliminó un borde de 1 píxel de grosor en cada imagen que significa una pérdida total de menos del 8% por canal. Podemos asumir además que la información sensible en cada imagen no se encuentra en sus bordes.

$$Pérdida = \frac{(50x2 + 48x2)}{50x50} = 7,8\%$$

Todas las capas convolucionales hacen uso de kernels 3x3 y Maxpooling 2x2. Además, se usa un padding de 1.

La función de activación usada en cada capa es ReLU menos la capa final que usa Regresión Logística para clasificar las imágenes en 1 neurona: clase 0 (sin CDI) y clase 1 (con CDI).

La función de error utilizada es la del Error Cuadrático Medio (MSE).

El número de épocas usado fue 30, para la tasa de aprendizaje se usó 0,01 y el tamaño de los lotes para calcular los errores en el algoritmo de descenso del gradiente fue 50. Se hicieron pruebas con un número mayor de épocas pero no hubo cambio significativo alguno en los resultados. También se probaron tasas de aprendizaje más pequeñas que resultaron en entrenamientos mucho más lentos (mayores a 3 días) sin mejorar la precisión de la prueba, mientras que tasas mayores disminuyeron drásticamente la efectividad.

El lenguaje de programación y las librerías usadas fueron Python (para scripts e inicialización de variables), Keras (es una librería que se monta sobre TensorFlow para la construcción y entrenamiento de redes neuronales) y scikit-learn para el reporte final de resultados.

10. RESULTADOS

El proceso de entrenamiento de la red neuronal para las 277.524 imágenes se completó en 22 horas (aprox. 33min por época) en la máquina virtual Ubuntu con procesador standard NVIDIA K80 GPU.

La evolución de la precisión y el error cometido por el prototipo puede apreciarse en la figura 30.

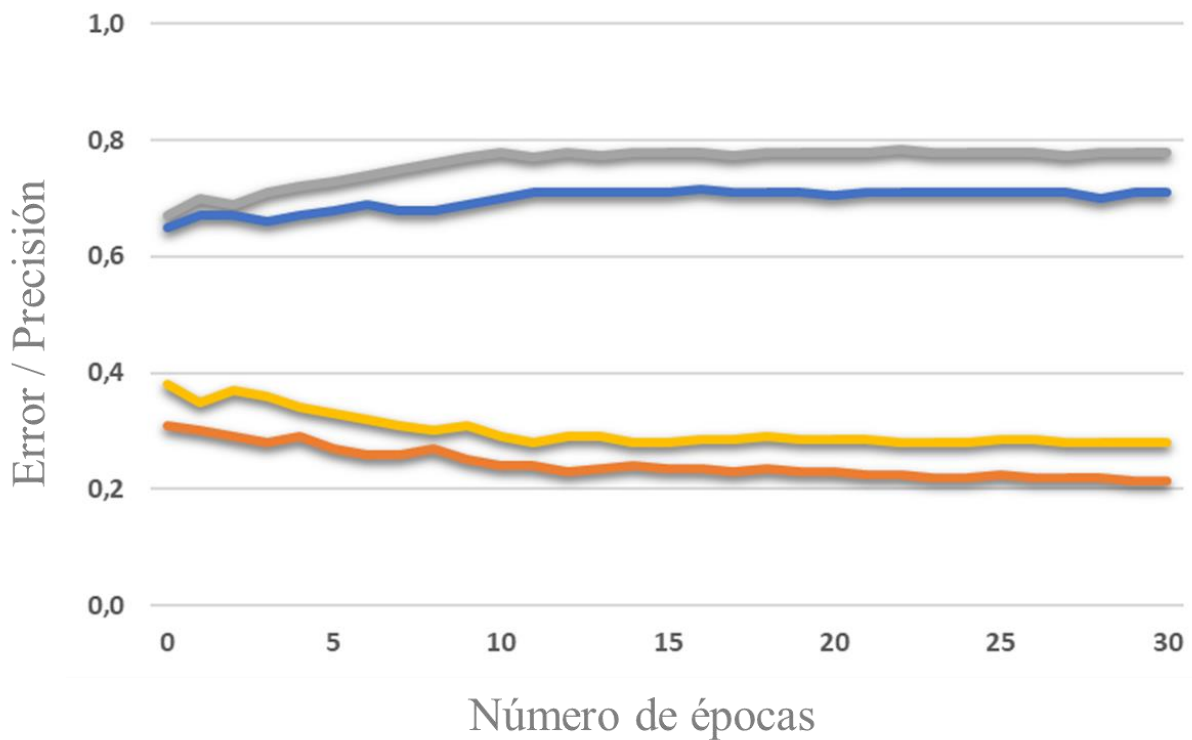


Fig.30. Proceso de entrenamiento de la red neuronal

Se puede notar que la convergencia es bastante rápida tanto en el error cometido como en la precisión de la red: a partir de la época 10 ya se hace estable para el resto del entrenamiento.

El entrenamiento de las imágenes de validación tiende a oscilar un poco más que todo el conjunto de entrenamiento, lo cual es normal porque su número es bastante menor. Así mismo, la precisión lograda es un poco menor y el error un poco mayor.

La precisión lograda por el prototipo para cada clase se puede observar en la tabla 3:

Precisión por clase	
Clase 0	0.87
Clase 1	0.64

Tabla 3. Precisión por clase

La precisión de la clase 0 implica que del número total de imágenes que se usaron para probar la red, ésta logró reconocer el 87% de las imágenes que pertenecen a dicha clase. En cambio, para la clase 1, la precisión llega al 64%.

Este resultado es lógico puesto que la proporción de imágenes pertenecientes a la clase 0 es más del doble que las de la clase 1 (198.738 sin cáncer y 78.786 con cáncer). Esto quiere decir que el prototipo aprendió a reconocer mucho mejor imágenes sin cáncer.

En cuanto a los valores totales, se calcularon los indicadores de precisión total, sensibilidad y especificidad tal como lo indica la tabla 4.

Resultados globales	
Precisión	0.7850
Sensibilidad	0.7916
Especificidad	0.7892

Tabla 4. Resultados globales

11. CONCLUSIONES

Considerando la pregunta problema, los objetivos planteados inicialmente y los resultados obtenidos durante el proyecto podemos concluir que:

Se logró implementar un prototipo con una sensibilidad del 79% y una especificidad del ~78%, superior a las esperadas en los objetivos iniciales del proyecto (75%). Además con una precisión del ~78%.

Se puede observar que la arquitectura de la red neuronal VGGNet 14 modificada a 3 grupos de capas convolucionales + un perceptrón de 3 capas totalmente conectadas utilizada para el desarrollo del prototipo funcionó adecuadamente y permitió resultados que sobrepasaron las expectativas iniciales.

El desbalance entre la cantidad de imágenes de entrenamiento con cáncer y sin cáncer afecta la precisión del prototipo a favor de la clase más numerosa, es decir, la red neuronal aprendió a reconocer mejor las imágenes histológicas sin cáncer que aquellas con cáncer.

Para este proyecto en particular, la tasa de aprendizaje que logró el mejor entrenamiento fue de 0,01. Tasas más pequeñas afectaron considerablemente el rendimiento del prototipo haciéndolo muy lento. Tasa mayores convergieron a precisiones mucho más bajas.

Finalmente, un número mayor de épocas no mejoró los resultados del prototipo.

12. RECOMENDACIONES

Como recomendaciones finales al proyecto tenemos:

1. Entrenar de nuevo la red neuronal haciendo uso de un conjunto de imágenes balanceado, donde el tamaño de ambas clases sea igual. Para ello, se puede usar las siguientes estrategias:
 - a. Disminuir la cantidad de imágenes de la clase 0 para que sea igual a la de la clase 1.
 - b. Usar técnicas de aumento (permutación de imágenes) para la clase 1, tales como rotación, flip vertical, flip horizontal, entre otras.
2. Considerar otro tipo de arquitecturas con el fin de verificar si se logra mejorar los resultados ya obtenidos.
3. Entrenar de nuevo el prototipo con imágenes modificadas a 56x56 px de resolución (3px en cada borde de ceros) para medir la precisión del prototipo versus las imágenes 48x48px.
4. Crear un nuevo prototipo que permita la ejecución de la red neuronal para nuevos casos sin la necesidad de entrenar de nuevo toda la red.
5. Revisar qué aspectos del prototipo sería necesario mejorar o añadir para que pueda considerarse como un potencial complemento en evaluaciones médicas del CDI.

13. REFERENCIAS BIBLIOGRÁFICAS

- American Cancer Society. (2019). *cancer.org*. Obtenido de <https://www.cancer.org/es>
- American Joint Committee on Cancer. (2010). *AJCC Cancer Staging Manual*. New York: Springer.
- Andre Esteva, B. K. (2017). Dermatologist-level classification of skin cancer with deep neural networks. *Nature*, 115-118.
- Andrew Janowczyk, A. M. (2016). Deep learning for digital pathology image analysis: A comprehensive tutorial with selected use cases. *Journal of Pathology Informatics*, 7-29.
- Breastcancer.org. (9 de julio de 2014). *Breastcancer.org*. Obtenido de <https://www.breastcancer.org/es>
- Bruce Alberts, A. J. (2008). *Biología molecular de la célula*. Barcelona: Omega.
- Ciresan, D., Meier, U., Masci, J., Gambardella, L. M., & Schmidhuber, J. (2011). Flexible, High Performance Convolutional Neural Networks for Image Classification. *Proceedings of the Twenty-Second international joint conference on Artificial Intelligence-Volume Volume Two 2*, (págs. 1237–1242). Manno-Lugano, Switzerland.
- Instituto Nacional de Cancerología ESE. (2013). *Guía de práctica clínica para la detección temprana, tratamiento integral, seguimiento y rehabilitación del cáncer de mama*. Bogotá: Colciencias.

Instituto Nacional del cáncer de los institutos nacionales de la salud de los Estados Unidos.

(5 de abril de 2016). *cancer.gov*. Obtenido de

<https://www.cancer.gov/espanol/cancer>

Karla Guzmán Santos, K. V. (2012). Carcinoma ductal infiltrante, el tipo de cáncer de mama más común. *iMedPub Journals Vol.8*.

Ramírez-Martínez CM, C.-R. J.-R.-R. (2015). Descripción clínica, anatomopatológica y de tratamiento de pacientes con cáncer de mama en una unidad de mastología de la ciudad de Medellín, Colombia. *Revista CES Medicina; 29(2)*, 181-190.

Saha, S. (15 de diciembre de 2018). *Medium*. Obtenido de

<https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53>

Samarasinghe, S. (2007). *Neural Networks for applied sciences and engineering: from fundamentals to complex pattern recognition*. New York: Auerbach Publications.

14. CÓDIGO

```
#####
#AUTOR: Daniel Jordan
#CARRERA: Matematicas
#ASUNTO: Arquitectura red neural
#ARCHIVO: redneural.py
#####

from keras.layers.convolutional import MaxPooling2D
from keras import backend as K
from keras.layers.normalization import BatchNormalization
from keras.layers.core import Dropout
from keras.layers.convolutional import SeparableConv2D
from keras.prototipos import Sequential
from keras.layers.core import Flatten
from keras.layers.core import Activation
from keras.layers.core import Dense

class REDNEURAL:
    @staticmethod
    def Crear(anchura, altura, profundidad, clases):
        TamTensor = -1
        prototipo = Sequential()
        forma = (altura, anchura, profundidad)

        if K.image_data_format() == "channels_first":
            forma = (profundidad, altura, anchura)
            TamTensor = 1
        prototipo.add(SeparableConv2D(64, (3, 3), padding="same",
            input_shape=forma)) #1era capa CONV
        prototipo.add(Activation("relu"))
        prototipo.add(BatchNormalization(axis=TamTensor))
        prototipo.add(MaxPooling2D(pool_size=(2, 2)))
        prototipo.add(SeparableConv2D(128, (3, 3), padding="same")) #2da capa
CONV
        prototipo.add(Activation("relu"))
        prototipo.add(BatchNormalization(axis=TamTensor))
        prototipo.add(SeparableConv2D(64, (3, 3), padding="same"))
        prototipo.add(Activation("relu"))
        prototipo.add(BatchNormalization(axis=TamTensor))
        prototipo.add(MaxPooling2D(pool_size=(2, 2)))
        prototipo.add(SeparableConv2D(256, (3, 3), padding="same")) #3era capa
CONV
        prototipo.add(Activation("relu"))
        prototipo.add(BatchNormalization(axis=TamTensor))
```

```

        prototipo.add(SeparableConv2D(128, (3, 3), padding="same"))
        prototipo.add(Activation("relu"))
        prototipo.add(BatchNormalization(axis=1))
        prototipo.add(SeparableConv2D(128, (3, 3), padding="same"))
        prototipo.add(Activation("relu"))
        prototipo.add(BatchNormalization(axis=1))
        prototipo.add(MaxPooling2D(pool_size=(2, 2)))
        prototipo.add(Flatten()) #Perceptron multicapa
        prototipo.add(Dense(256))
        prototipo.add(Activation("relu"))
        prototipo.add(BatchNormalization())
        prototipo.add(Dense(10))
        prototipo.add(Activation("sigmoid"))
    return prototipo

```

```

#*****
#AUTOR: Daniel Jordan
#CARRERA: Matematicas
#ASUNTO: Entrenamiento de la red
#ARCHIVO: entrenamiento.py
#*****

```

```

from TESIS import REDNEURAL
from imutils import paths
from keras.callbacks import LearningRateScheduler
from keras.preprocessing.image import ImageDataGenerator
from keras.utils import np_utils
from keras.optimizers import Adagrad
from sklearn.metrics import classification_report
from sklearn.metrics import confusion_matrix
import os
import numpy as np
import matplotlib.pyplot as plt
import argparse
import matplotlib
matplotlib.use("Agg")
import os

```

```

ORIGEN_DATA = "data/orig"
RUTA_CANCER = "datasets/idc"
RUTA_ENTREN = os.path.sep.join([RUTA_CANCER, "training"])
RUTA_VAL = os.path.sep.join([RUTA_CANCER, "validation"])
RUTA_PRUEBAS = os.path.sep.join([RUTA_CANCER, "testing"])
PORCENT_PRUEBAS = 0.8
PORCENT_VAL = 0.1

```

```

tasa = 1*(10**-2)

```

```

epocas = 30
lotes = 50
lineaComando = argparse.ArgumentParser() # Parsing linea de comando
lineaComando.add_argument("-p", "--plot", type=str, default="graf.png",
    help="directorio de la grafica")
argumentos = vars(lineaComando.parse_args())
rutaEntrenamiento = list(paths.list_images(config. RUTA_ENTREN)) #valida ruta de los
directorios
EntrenamientoCompleto = len(rutaEntrenamiento)
validacionCompleto = len(list(paths.list_images(config. RUTA_VAL)))
pruebasCompleto = len(list(paths.list_images(config. RUTA_PRUEBAS)))
categoriasEntrenamiento = [int(p.split(os.path.sep)[-2]) for p in rutaEntrenamiento]
#categoriza las clases
categoriasEntrenamiento = np_utils.to_categorical(categoriasEntrenamiento)
clases = categoriasEntrenamiento.sum(axis=0)
pesosClases = clases.max() / clases

motorEntrenamiento = trainAug.flow_from_directory(
    config. RUTA_ENTREN,
    class_mode="categorical",
    target_size=(48, 48),
    color_mode="rgb",
    shuffle=True,
    batch_size=lotes) #inicializa el generador del entrenamiento

motorValidacion = valAug.flow_from_directory(
    config. RUTA_VAL,
    class_mode="categorical",
    target_size=(48, 48),
    color_mode="rgb",
    shuffle=False,
    batch_size=lotes) #inicializa el generador del conjunto validacion

motorPruebas = valAug.flow_from_directory(
    config. RUTA_PRUEBAS,
    class_mode="categorical",
    target_size=(48, 48),
    color_mode="rgb",
    shuffle=False,
    batch_size=lotes) #inicializa el conjunto de pruebas

modelo = REDNEURAL.Crear(width=48, height=48, depth=3,
    classes=2)
opt = Adagrad(lr=tasa, decay=tasa / epocas)
modelo.compile(loss="mean_squared_error", optimizer=opt,
    metrics=["accuracy"]) #inicializa la red neural

```

```

actual = modelo.fit_generator(
    motorEntrenamiento,
    steps_per_epoch=EntrenamientoCompleto // lotes,
    validation_data=motorValidacion,
    validation_steps=validacionCompleto // lotes,
    class_weight=pesosClases,
    epochs=epocas) #actualiza parametros de la red

print("espere por favor...")
motorPruebas.reset()
indices = modelo.predict_generator(motorPruebas,
    steps=(pruebasCompleto // lotes) + 1)
indices = np.argmax(indices, axis=1)

print(classification_report(motorPruebas.classes, indices,
    target_names=motorPruebas.class_indices.keys()))

matrizConfusion = confusion_matrix(motorPruebas.classes, indices)
total = sum(sum(matrizConfusion))
acc = (matrizConfusion[0, 0] + matrizConfusion[1, 1]) / total
sensitivity = matrizConfusion[0, 0] / (matrizConfusion[0, 0] + matrizConfusion[0, 1])
specificity = matrizConfusion[1, 1] / (matrizConfusion[1, 0] + matrizConfusion[1, 1])
print(matrizConfusion) #resultados
print("acc: {:.4f}".format(acc))
print("sensitividad: {:.4f}".format(sensitivity))
print("especificidad: {:.4f}".format(specificity))
E = epocas
plt.style.use("ggplot")
plt.figure()
plt.xlabel("Epocas")
plt.ylabel("Error y precision")
plt.title("Resultados")
plt.savefig(argumentos["plot"])

```