

**INSTITUTO POLITÉCNICO NACIONAL  
ESCUELA SUPERIOR DE CÓMPUTO**



## **Ejercicio de Lab10: Redes neuronales convolucionales**

**Alumnos:** Alvarado Sandoval Juan Manuel

Arzate Pérez Esdras Isaac

Boone Posada Cristian De Jesús

Cerda Zapata Eduardo

Delgado Acosta Luis Bernardo

Esponda Cervantes Alfredo Daniel

Franco Calderas Sergio Alberto

Hernández Jiménez Erick Daniel

Martinez Ayala Gerardo

Pérez Lucio Kevyn Alejandro

Sepúlveda Ramírez Carlos Emiliano

Vázquez Aguilar Alejandro

**Grupo:** 5BV1

# Redes neuronales convolucionales

## Introducción a las RN

Las redes neuronales artificiales (RNAs) son modelos computacionales inspirados en el funcionamiento del cerebro humano. Estas consisten en capas de neuronas artificiales que procesan datos mediante la transmisión de señales ponderadas, simulando cómo las neuronas biológicas activan impulsos eléctricos. Se utilizan ampliamente en tareas como clasificación, regresión, predicción, y modelado de relaciones complejas en datos. Una red neuronal básica contiene:

1. **Capa de entrada:** Recibe los datos.
2. **Capas ocultas:** Realizan cálculos a través de funciones de activación.
3. **Capa de salida:** Entrega los resultados.

## Redes Convolucionales

Las redes neuronales convolucionales (RNCs o CNNs, por sus siglas en inglés) son un tipo especializado de red neuronal diseñado para procesar datos que tienen una estructura de cuadrícula, como imágenes. A diferencia de las redes tradicionales, las RNCs utilizan operaciones de convolución para extraer características jerárquicas y espaciales de los datos de entrada.

Son especialmente útiles en visión por computadora, como en clasificación de imágenes, detección de objetos y segmentación.

Son similares a las redes neuronales multicanal, su principal ventaja es que cada parte de la red se le entrena para realizar una tarea, esto reduce significativamente el número de capas ocultas, por lo que el entrenamiento es más rápido. Además, presenta invarianza a la traslación de los patrones a identificar.

## Antecedentes

1. **Modelo de Neocognitrón (1980):** Propuesto por Kunihiko Fukushima, fue uno de los primeros modelos de red neuronal con características similares a las convoluciones. Inspiró el diseño de las RNCs modernas.
2. **Red de LeNet (1998):** Yann LeCun popularizó las CNNs con esta arquitectura, utilizada en el reconocimiento de dígitos escritos a mano.
3. **Avances recientes (2010s):** Con el crecimiento de datos (big data), avances en hardware (GPUs) y algoritmos de optimización, las CNNs comenzaron a dominar en tareas como el desafío ImageNet.

## Arquitectura

Una red convolucional es un tipo de red neuronal artificial multicapa que consta de capas convolucionales y de reducción alternadas, y al finalmente tiene capas de conexión total como una red perceptrón multicapa.

En la **convolución** se realizan operaciones de productos y sumas entre la capa de partida y los  $n$  filtros (o kernel) que genera un mapa de características. Las características extraídas corresponden a cada posible ubicación del filtro en la imagen original.

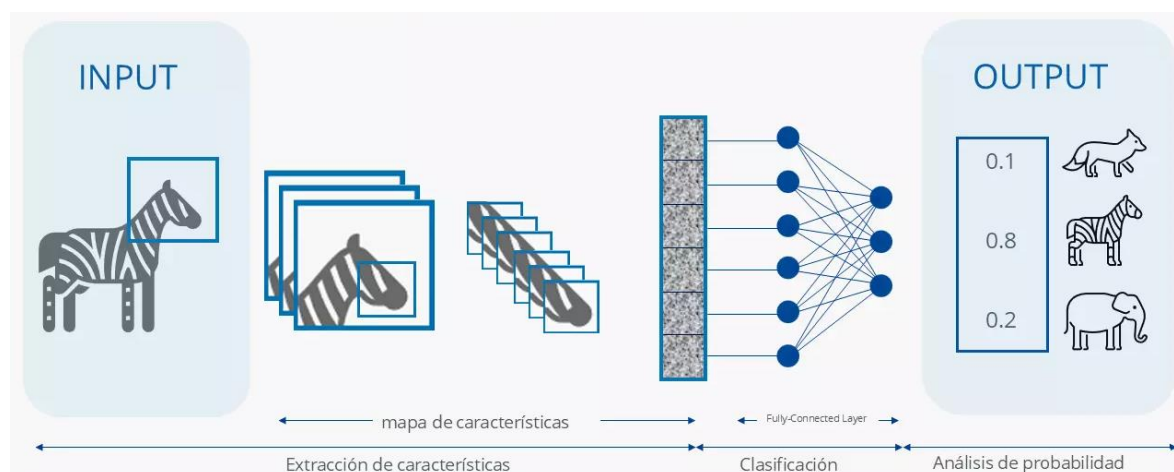


Fig.1 Arquitectura de una CNN

La ventaja es que el mismo filtro sirve para extraer la misma característica en cualquier parte de la entrada, con esto que consigue reducir el número de conexiones y el número de parámetros a entrenar en comparación con una red multicapa de conexión total.

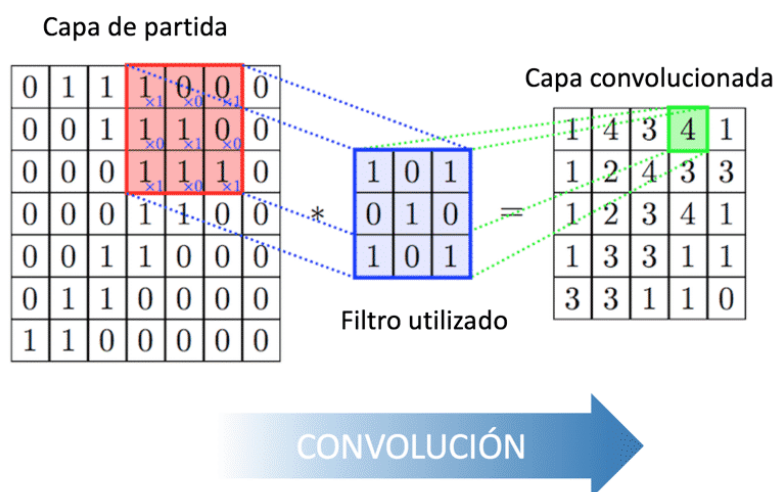
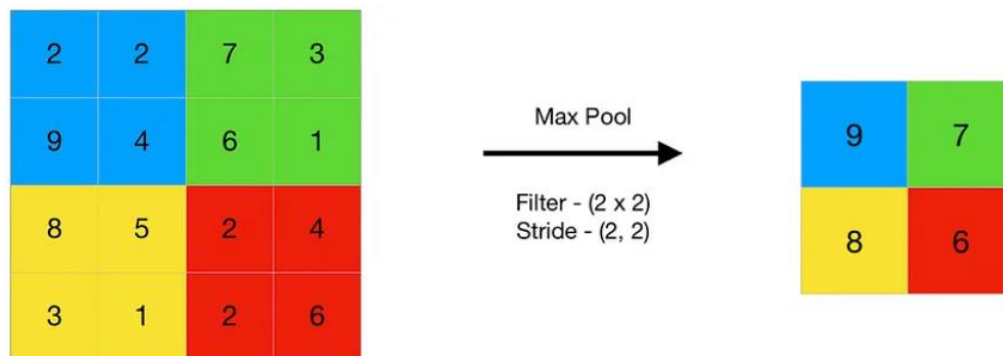


Fig.2 Proceso de convolución

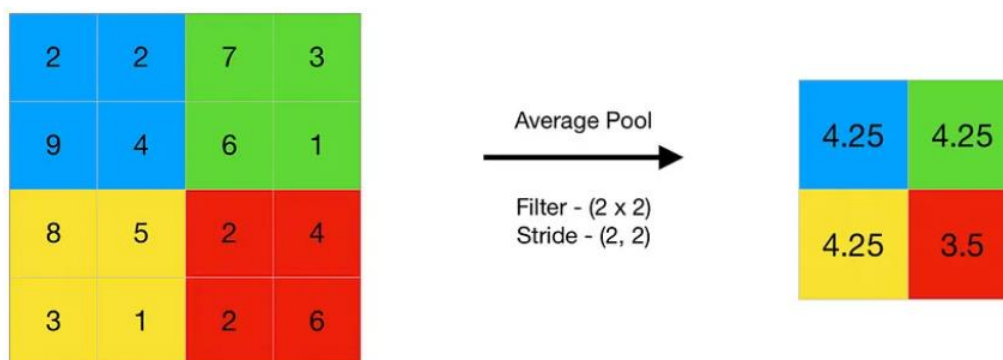
La operación de **pooling** se aplica entre dos capas de convolución. Esta recibe en la entrada las features maps formadas en la salida de la capa de convolución y su papel es reducir el tamaño de las imágenes y, a la vez, preservar sus características más esenciales. Entre las más utilizadas tenemos las siguientes:

1. **Max pooling:** Se selecciona el máximo de elementos de la región del mapa de características cubierta por el filtro, por lo tanto, el resultado después de la capa de agrupación máxima sería un mapa de características que contenga las características más destacadas del mapa de características anterior.



*Fig.3 Representación del Max pooling*

2. **Average pooling:** Calcula el promedio de los elementos presentes en la región del mapa de características cubierta por el filtro, mientras que el Max pooling proporciona la característica más destacada en un área particular del mapa de características, el Average pooling proporciona el promedio de las características presentes en un área.



*Fig.4 Representación del Average pooling*

3. **Global pooling:** Reduce cada canal del mapa de características a un único valor. Por lo tanto, un mapa de características  $n_h \times n_w \times n_c$  se reduce a un mapa de características  $1 \times 1 \times n_c$ . Esto es equivalente a utilizar un filtro de dimensiones  $n_h \times n_w$ , es decir, las dimensiones del mapa de características. Además, puede ser un Global Max pooling o un global Average pooling.

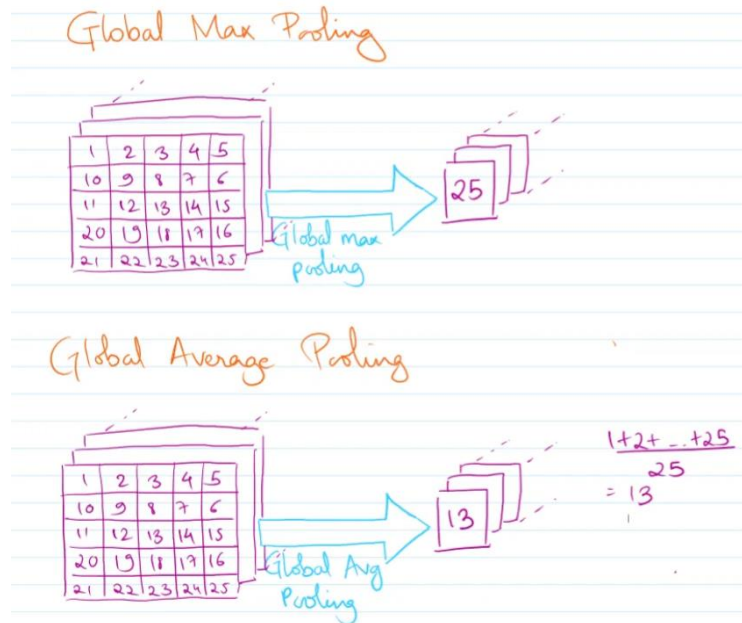


Fig.5 Representación del Global pooling

El proceso de **clasificación** se lleva a cabo después de las capas convolucionales y de pooling. La última capa de la red es una capa clasificadora, diseñada para tener tantas neuronas como clases se desean predecir. Cada neurona de esta capa representa una de las posibles categorías, y su salida indica la probabilidad de que la entrada pertenezca a esa clase específica.

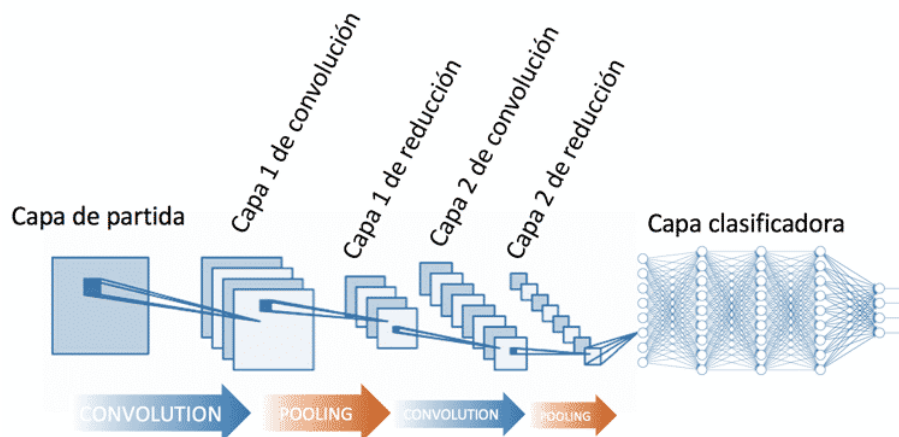


Fig.6 Representación de una CNN

## Ventajas de las redes neuronales convolucionales

1. **Eficiencia en la identificación de patrones visuales:** Las CNN están diseñadas para extraer automáticamente características relevantes de las imágenes, como bordes, texturas y formas, sin necesidad de ingeniería manual de características.
2. **Capa para procesar grandes cantidades de datos:** Las CNN son ideales para trabajar con datos voluminosos y de alta dimensión, como imágenes de alta resolución o secuencias de video.
3. **Capacidad de aprendizaje y mejora continua:** A medida que se entrenan con más datos, las CNN mejoran su rendimiento, identificando patrones más complejos y generalizando mejor en tareas diversas.
4. **Aplicaciones versátiles:** Son ampliamente utilizadas en campos como la visión por computadora, el procesamiento de video, la salud (diagnósticos médicos a partir de imágenes), la conducción autónoma y el reconocimiento facial.

## Desventajas de las redes neuronales convolucionales

1. **Requieren grandes cantidades de datos:** Para alcanzar un buen desempeño, las CNN necesitan datasets extensos y diversos, lo que puede ser un desafío en ciertos contextos donde los datos son limitados.
2. **Elevados costos computacionales:** Entrenar y ejecutar CNN puede requerir hardware especializado, como GPUs o TPUs, lo que puede incrementar los costos de implementación.
3. **Tiempos prolongados de entrenamiento:** Dado su tamaño y complejidad, las CNN pueden tomar días o incluso semanas para entrenarse adecuadamente en datasets grandes.
4. **Sensibilidad al sobreajuste:** Sin técnicas adecuadas como la regularización o el aumento de datos, las CNN pueden memorizar los datos de entrenamiento en lugar de generalizar, afectando su rendimiento en datos no vistos.
5. **Interpretabilidad limitada:** Aunque son efectivas, las CNN se consideran "cajas negras", lo que significa que puede ser difícil entender cómo y por qué toman ciertas decisiones.

## Parámetros e Hiperparámetros

Las CNN tienen varios parámetros que se ajustan durante el entrenamiento, como pesos y sesgos. Además, hay hiperparámetros importantes que deben definirse antes del entrenamiento:

1. **Tamaño del kernel:** Define el tamaño de la ventana para las operaciones de convolución.
2. **Stride:** Controla cuánto se desplaza el kernel en cada paso.
3. **Padding:** Determina si se añaden bordes al mapa de características para mantener su tamaño original tras la convolución.

## Funciones de Activación

Las funciones de activación son esenciales en las redes neuronales, ya que permiten modelar relaciones no lineales en los datos. Estas funciones transforman la salida de las neuronas de cada capa, lo que facilita la resolución de problemas complejos.

En las redes neuronales convolucionales se utilizan diferentes funciones de activación, entre las más comunes se encuentran:

1. **ReLU:** Define la salida como  $\text{ReLU}(x) = \max(0, x)$ . Es computacionalmente eficiente y ayuda a mitigar el problema de los gradientes desvanecientes.
2. **Leaky ReLU:** Es una variante de ReLU que introduce un pequeño gradiente para los valores negativos, lo que evita que las neuronas se "mueran" (es decir, que dejen de aprender debido a salidas constantes de 0).
3. **Sigmoide:** Aunque menos común en capas intermedias, esta función se utiliza en la capa de salida para tareas de clasificación multiclase. Convierte los valores en probabilidades entre 0 y 1.

## Optimización y Función de Pérdida

El aprendizaje de una CNN implica minimizar una función de pérdida que evalúa qué tan bien predice el modelo.

1. **Cross-Entropy Loss:** Utilizada comúnmente para tareas de clasificación. Penaliza predicciones incorrectas basándose en la probabilidad predicha para cada clase.
2. **MSE (Mean Squared Error):** Utilizada en tareas de regresión.
3. **IoU Loss (Intersection over Union):** Popular en tareas de detección de objetos, mide la superposición entre predicciones y etiquetas reales.

## ¿Cómo se entrena una red neuronal convolucional?

El entrenamiento de una CNN comienza con la preparación de los datos, los cuales deben estar organizados en conjunto de entrenamiento, validación y prueba. Es fundamental que los datos sean preprocesados adecuadamente, lo que incluye la normalización de los valores de los píxeles para garantizar que se encuentren en un rango estándar, como  $[0,1]$ . Además, se pueden aplicar técnicas de aumento de datos para incrementar la variabilidad del conjunto de entrenamiento y mejorar la capacidad de generalización de la red.

La arquitectura de la red se diseña considerando las necesidades del problema. Esto implica definir una capa de entrada que acepte imágenes del tamaño especificado y agregar bloques de convolución con número creciente de filtros para extraer características de manera jerárquica. Entre los bloques se incluyen capas de agrupación (pooling) que reducen dimensionalidad y retienen información relevante. Al final de la red, se añaden capas totalmente conectadas que combinan las características extraídas para generar las predicciones. La última capa depende del tipo de problema: para clasificación, una activación softmax; para regresión, una activación lineal.

Una vez definida la arquitectura, se configuran los parámetros del entrenamiento. Esto incluye seleccionar una función de pérdida adecuada (como entropía cruzada para clasificación o error cuadrático medio para regresión) y un optimizador eficiente, como Adam o RMSprop, ajustando la tasa de aprendizaje inicial. También se definen métricas para monitorear el desempeño, como la precisión o el error medio cuadrático.

Durante el proceso de entrenamiento, los datos de entrenamiento alimentan la red, que ajusta sus pesos en función de la función de pérdida. Al final de cada época, los datos de validación se usan para evaluar el rendimiento y prevenir el sobreajuste. Se monitorean gráficos de pérdida y métricas clave para identificar patrones de mejora después de varias épocas, se ajusta dinámicamente la tasa de aprendizaje para afinar el proceso de optimización.

Este ciclo se repite hasta que la red alcanza un rendimiento satisfactorio en el conjunto de validación. Posteriormente, se evalúan con los datos de prueba para confirmar su capacidad de generalización antes de pasar a la fase de predicción con datos nuevos.



## Optimización del Entrenamiento

El entrenamiento puede ser un proceso exigente, tanto en tiempo como en recursos. A continuación, se presentan estrategias para optimizar el proceso de entrenamiento:

1. **Ajuste dinámico del Learning Rate:** Una tasa de aprendizaje adecuada es crucial para la convergencia del modelo. Las estrategias dinámicas, como reducir la tasa de aprendizaje cuando la mejora en la pérdida se estabiliza, ayudan a afinar el modelo.
2. **Early Stopping:** Detener el entrenamiento automáticamente cuando el rendimiento en los datos de validación deje de mejorar durante varias épocas consecutivas. Esto previene el sobre-entrenamiento y ahorra recursos.
3. **Técnicas de regularización:**
  - Dropout: Desactiva aleatoriamente un porcentaje de neuronas durante el entrenamiento para evitar que el modelo dependa excesivamente de ciertas características.
  - L2 Regularization: Añade un término de penalización a los pesos grandes, promoviendo soluciones más simples y generalizables.
4. **Data Augmentation:** Genera variaciones artificiales de los datos de entrenamiento aplicando transformaciones como rotaciones, escalado, volteo y cambios de brillo. Esto mejora la capacidad del modelo para generalizar.

El entrenamiento de CNNs modernas puede ser intensivo, lo que hace que el hardware desempeñe un papel crucial para ahorrar recursos computacionales y tiempo de entrenamiento. Algunas recomendaciones para este aspecto son:

1. **GPUs y TPUs:**
  - Las Graphics Processing Units están optimizadas para cálculos paralelos, acelerando operaciones como convoluciones y multiplicaciones matriciales.
  - Las Tensor Processing Units de Google ofrecen una alternativa aún más rápida para entrenar redes profundas.
2. **Entrenamiento en la nube:** Plataformas como Google Cloud, AWS y Microsoft Azure permiten acceder a hardware especializado sin necesidad de adquirirlo, reduciendo los costos iniciales.
3. **Modelos optimizados para dispositivos Edge:** CNNs diseñadas para ejecutarse en hardware limitado, como teléfonos móviles y cámaras. Ejemplos incluyen MobileNet y EfficientNet, que ofrecen un equilibrio entre precisión y eficiencia computacional.

## Avances Recientes

1. **Vision Transformers:** Los transformers, originalmente diseñados para procesamiento de lenguaje natural, están siendo adaptados para visión por computadora. Vision Transformers dividen las imágenes en parches y aplican mecanismos de atención para analizar relaciones globales, superando a las CNNs en algunos casos.
2. **Arquitecturas Híbridas:** Modelos que combinan CNNs con RNNs o transformers para aprovechar lo mejor de cada arquitectura. Por ejemplo, en análisis de video, se usan CNNs para extraer características de cada fotograma y RNNs para modelar la temporalidad.

## Retos y Soluciones

1. **Sesgos en los Datos:** Las CNNs pueden amplificar sesgos presentes en los datos de entrenamiento. Soluciones como el aumento de datos equilibrado y las técnicas de fairness ayudan a mitigar este problema.
2. **Interpretación:** Las CNNs son percibidas como “cajas negras.” Herramientas como Grad-CAM y saliency maps facilitan la interpretación de sus decisiones, aumentando la confianza en aplicaciones críticas como salud y seguridad.
3. **Reducción del Consumo de Energía:** Redes ligeras como SqueezeNet y MobileNet están diseñadas para reducir el uso de energía sin sacrificar demasiada precisión, haciendo que las CNNs sean más sostenibles.

## Áreas de aplicación de las redes neuronales convolucionales

Antes de que existieran las CNN, se utilizaban métodos manuales y laboriosos de extracción de características para identificar objetos en imágenes. Las redes neuronales convolucionales ofrecen un enfoque más escalable para tareas de clasificación de imágenes y reconocimiento de objetos. Actualmente, las CNN se utilizan en diversas aplicaciones:

1. **Reconocimiento de imágenes y voz:** Reconocimiento automático de objetos o personas en imágenes y videos, por ejemplo, para servicios de etiquetado de fotos en smartphones, sistema de reconocimiento facial o asistentes de voz como Siri o Alexa.
2. **Diagnóstico médico:** Análisis de imágenes médicas gracias al reconocimiento de imágenes por IA para apoyar el diagnóstico, por ejemplo, en radiografías, tomografías computarizadas y resonancias magnéticas.
3. **Vehículos autónomos:** Reconocimiento de características de la carretera y obstáculos, por ejemplo, para coches autónomos.

4. **Redes sociales:** Las CNN se utilizan para moderar contenidos automáticamente y crear publicidad personalizada.
5. **Marketing y comercio minorista:** Por ejemplo, búsqueda visual y colocación de productos gracias al data mining.
6. **Procesamiento de Texto:** Tareas como análisis de sentimientos o clasificación de texto combinado con embeddings como Word2Vec o GloVe.
7. **Análisis de Señales:** Se aplican al análisis de datos de series temporales, como electrocardiogramas y señales de audio, para identificar patrones o anomalías.
8. **Bioinformática:** Se utilizan para predecir estructuras de proteínas, identificar secuencias genéticas o analizar imágenes médicas, como resonancias magnéticas y radiografías.

## Referencias

MathWorks. (n.d.). Train a convolutional neural network for regression. Recuperado de <https://la.mathworks.com/help/deeplearning/ug/train-a-convolutional-neural-network-for-regression.html>

Economipedia. (n.d.). Red neuronal convolucional. Recuperado de <https://economipedia.com/definiciones/red-neuronal-convolucional.html>

<https://www.datacamp.com/es/tutorial/introduction-to-convolutional-neural-networks-cnns>

IONOS. (n.d.). Convolutional Neural Network: Cómo funcionan las redes neuronales convolucionales. Recuperado de <https://www.ionos.mx/digitalguide/paginas-web/desarrollo-web/convolutional-neural-network/#:~:text=Actualmente%2C%20las%20CNN%20se%20utilizan,voz%20como%20Siri%20o%20Alexa>

DataCamp. (n.d.). *Introduction to Convolutional Neural Networks (CNNs)*. Recuperado el 20 de diciembre de 2024, de <https://www.datacamp.com/es/tutorial/introduction-to-convolutional-neural-networks-cnns>

IBM. (n.d.). *Basics of neural networks*. Recuperado el 20 de diciembre de 2024, de <https://www.ibm.com/docs/es/spss-modeler/saas?topic=networks-basics-neural>

ResearchGate. (n.d.). *Arquitectura de la red neuronal convolucional 3D (3D-CNN)* [Figura]. Recuperado el 20 de diciembre de 2024, de <https://www.researchgate.net/figure/Arquitectura-de-la-red-neuronal-convolucional-3D-3D-CNN-fig1-340023601>

Calvo, D. (n.d.). *Red neuronal convolucional*. Recuperado el 20 de diciembre de 2024, de <https://www.diegocalvo.es/red-neuronal-convolucional/>