Guía de estudio - Redes Neuronales Convolucionales

¡Hola! Te damos la bienvenida a esta nueva guía de estudio.

¿En qué consiste esta guía?

En la siguiente guía revisaremos los conceptos de las redes neuronales convolucionales y su aplicación.

Las redes neuronales convolucionales (CNN) son una herramienta poderosa para el procesamiento de información visual. Estas redes, inspiradas en la estructura y funcionamiento del sistema visual humano, han revolucionado la forma en que los computadores pueden interpretar y comprender imágenes. En esta guía de estudio nos sumergimos en el fascinante mundo de las CNN, explorando en detalle su arquitectura, sus procesos principales de convolución y pooling y analizaremos un ejemplo aplicado paso a paso que ilustra la implementación práctica de estos conceptos.

Comenzaremos explicando los fundamentos de las CNN, destacando su arquitectura única diseñada para extraer y aprender patrones jerárquicos en datos visuales. Se abordará la estructura fundamental de las capas convolucionales y como cada una contribuye a la capacidad de la red para reconocer características específicas en imágenes, desde bordes más simples hasta conceptos más abstractos.

Posteriormente profundizaremos en el núcleo de las CNN, explicando en detalle el proceso de convolución. Mostraremos cómo esta operación esencial permite a la red aprender y detectar características claves en imágenes mediante la aplicación de filtros convolucionales. Comprenderemos cómo se produce la convolución a lo largo de las capas y cómo esta técnica facilita la extracción de características significativas en diferentes niveles de abstracción.

Luego se revisará el proceso de pooling en el contexto de las CNN. Veremos como las capas de pooling desempeñan una tarea de reducción de la resolución de los mapas de características, preservando la información esencial y mejorando la eficiencia computacional de la red. Se examinarán diferentes tipos de pooling como el max pooling y el average pooling, y evaluaremos el impacto en el rendimiento de la red.

Terminaremos aplicando todos estos conceptos en un ejemplo aplicado paso a paso. A través de un caso práctico, se implementará una CNN desde cero, destacando la importancia de cada componente en la construcción y entrenamiento de la red.

¡Vamos con todo!

Tabla de contenidos

¡Comencemos!

Redes neuronales convolucionales (CNN)

Las redes neuronales convolucionales (CNN) representan una categoría especializada de modelos de aprendizaje profundo diseñadas para procesar y analizar datos visuales, como imágenes y videos. Su arquitectura se inspira en la organización jerárquica del sistema visual biológico, lo que las hace particularmente eficientes en tareas de reconocimiento visual y clasificación de patrones. Las CNN se componen de capas convolucionales que aplican filtros para extraer características locales, habitualmente seguidas de capas de pooling que disminuyen la resolución del mapa de características y conservan la información relevante. Este diseño permite a las CNN capturar desde detalles simples hasta detalles complejos. La capacidad de compartir parámetros a lo largo de la imagen, gracias a las operaciones convolucionales, otorga a las CNN una ventaja en la detección de patrones invariantes a la traslación, haciendo que sean ampliamente utilizadas en tareas como reconocimiento facial, clasificación de objetos y segmentación de imágenes. Su versatilidad y éxito en una variabilidad de aplicaciones las convierten en una herramienta esencial en el campo de la visión por computador y el procesamiento de datos visuales.

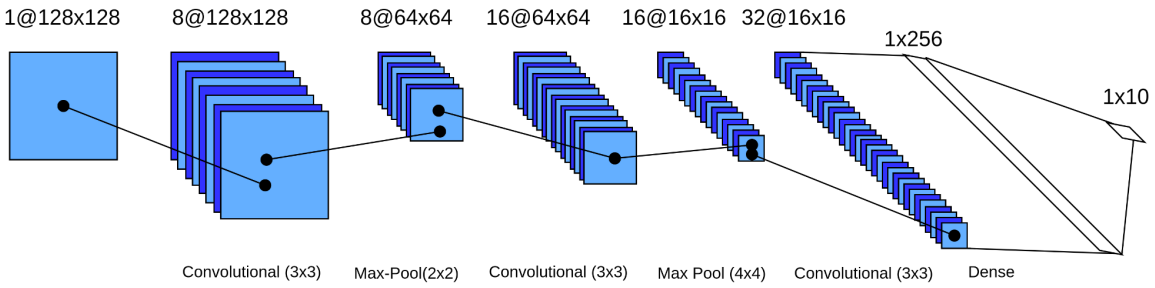

Figura 1. Arquitectura típica de una red neuronal convolucional (CNN)
Fuente: Desafío Latam

En las redes neuronales convolucionales (CNN) su input serán principalmente imágenes, en la Figura 1 se muestra una CNN siguiendo un patrón habitual. Este está compuesto por: Capa INPUT (imagen) -> Capa CONVOLUCIÓN -> Operación de POOLING -> Capa CONVOLUCIÓN -> Operación POOLING -> Capa CONVOLUCIÓN, etc. Este patrón se repite muchas veces hasta concluir en una capa con datos en una dimensión, luego con este vector de características conectamos con una red neuronal Feed-Forward Fully Connected definiendo cantidad de capas y neuronas por capas como se hace habitualmente, en el que debemos asignar a la última una cantidad de neuronas que depende del problema a resolver.

Al observar la Figura 1, vemos que las capas convolucionales no modifican el tamaño de la imagen. Esto no es algo que suceda en todos los casos, sin embargo, lo habitual es que las capas convolucionales modifiquen poco los tamaños de la imagen o mapa de características. Cuando estas operaciones se realizan más cerca del input, su objetivo es

extraer características generales que existan en la imagen, y a medida que se aplican en un nivel de profundidad mayor de la red entonces estarán capturando características cada vez con mayor nivel de abstracción. Por otro lado, las operaciones de Pooling vienen a resumir las características representadas hasta entonces en el nivel de profundidad de la red en la que se apliquen, reduciendo el ancho y alto de la imagen tanto como se haya modelado.

Este tipo de Red se ve bastante diferente a las que hemos visto hasta ahora, y uno de los componentes que no vemos son las funciones de activación (la Figura 1 no muestra funciones de activación). Sin embargo, estas están presentes y se aplican antes de generar cada capa. Debido a la alta demanda de cálculos computacionales que se requieren y a los buenos resultados obtenidos, es que se utiliza para esto la función de activación ReLU. Otro componente que no vemos en la Figura 1 es el bias, que en CNN estará presente participando antes de aplicar la función de activación.

Es importante también mencionar que cuando hacemos operaciones de pooling no contaremos a esto como una capa. A medida que avanzamos en la profundidad en CNN tendremos cada vez mapas de características más pequeñas, pero una cantidad mayor de canales (profundidad del mapa de características). Al llegar al final de las capas de Convolución y operaciones de Pooling tendremos un vector de características que será conectado a una red neuronal Feed-Forward Fully connected con tantas capas como lo necesitemos modelar, habiendo definido la cantidad de neuronas de su última capa de acuerdo al problema a resolver.

Los parámetros que debe aprender nuestra red CNN corresponden a filtros en terminología en el campo de visión computacional junto con los parámetros que están asociados a la red FFNN al final de CNN, típicamente necesitaremos estimar millones de estos.

Convolución

Una imagen se representa como un tensor típicamente de tres dimensiones: alto, ancho y profundidad. Esta última almacena información de los canales asociados al color de la imágen, por ejemplo, un canal para almacenar intensidad de rojo, otro para verde y un tercero para azul.

La convolución es un proceso en el cual aplicamos una matriz a cada píxel de una imagen multiplicando elemento a elemento. Esta operación se conoce en el campo de la visión por computador como **aplicar un filtro** a una imagen, y tiene como objetivo capturar alguna característica presente en la imagen.

Filtro

Un filtro o kernel es una matriz bidimensional de números que se utiliza para realizar operaciones locales en una imagen. Este filtro se desliza a lo largo de la imagen, multiplicando sus valores con los valores correspondientes de la región de la imagen en la

que se encuentra. La operación resultante es una combinación ponderada de los píxeles de la región, y se utiliza para resaltar características específicas o realizar transformaciones en la imagen original.

La matriz del filtro actúa como una *ventana* que se desplaza por la imagen. Cada elemento de la matriz del filtro tiene un peso asociado y contribuye al cálculo final del nuevo valor del píxel en el mapa de características resultante. El filtro que se utilice determinará qué tipo de característica se extraerá de la imagen.

Por ejemplo, si aplicamos el filtro que se muestra en la figura a la imagen el resultado será una nueva imagen donde se resaltan sus bordes, entonces diremos que este filtro estará capturando información con respecto a los bordes de ella.

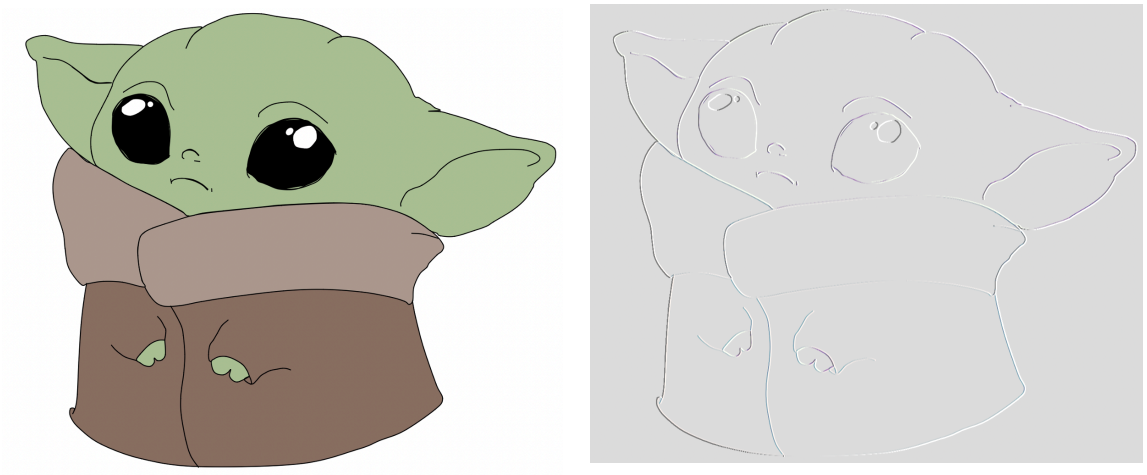

Figura 2. Filtro para capturar bordes de la imagen
Fuente: DesafíoLatam

La operación de ir pasando el kernel como si este fuera una ventana sobre la imagen se puede apreciar en la Figura anterior, en la que representamos una imagen en escala de grises, es decir, con solo un canal y se muestra en términos de los valores de intensidad de gris en cada uno de sus píxeles.

En este ejemplo vamos pasando un kernel de 3x3 sobre la imagen, se puede apreciar que debido a la dimensión del kernel es que se perderán la primera y última fila de la imagen, como también la primera y última columna. Para solucionar esto se puede agregar a la imagen píxeles extras en todo el contorno, con valores habitualmente iguales a cero. A esto se le conoce como el **PADDING,** en nuestro ejemplo usamos cero ya que no tenemos PADDING.

Por otro lado, la decisión de cuánto se moverá la ventana en cada paso del kernel se denomina **STRIDE;** en este ejemplo tenemos STRIDE igual a 1. Si aplicamos un PADDING de 1 y STRIDE de 1, entonces el resultado sería un mapa de características con las mismas dimensiones que la imagen original.

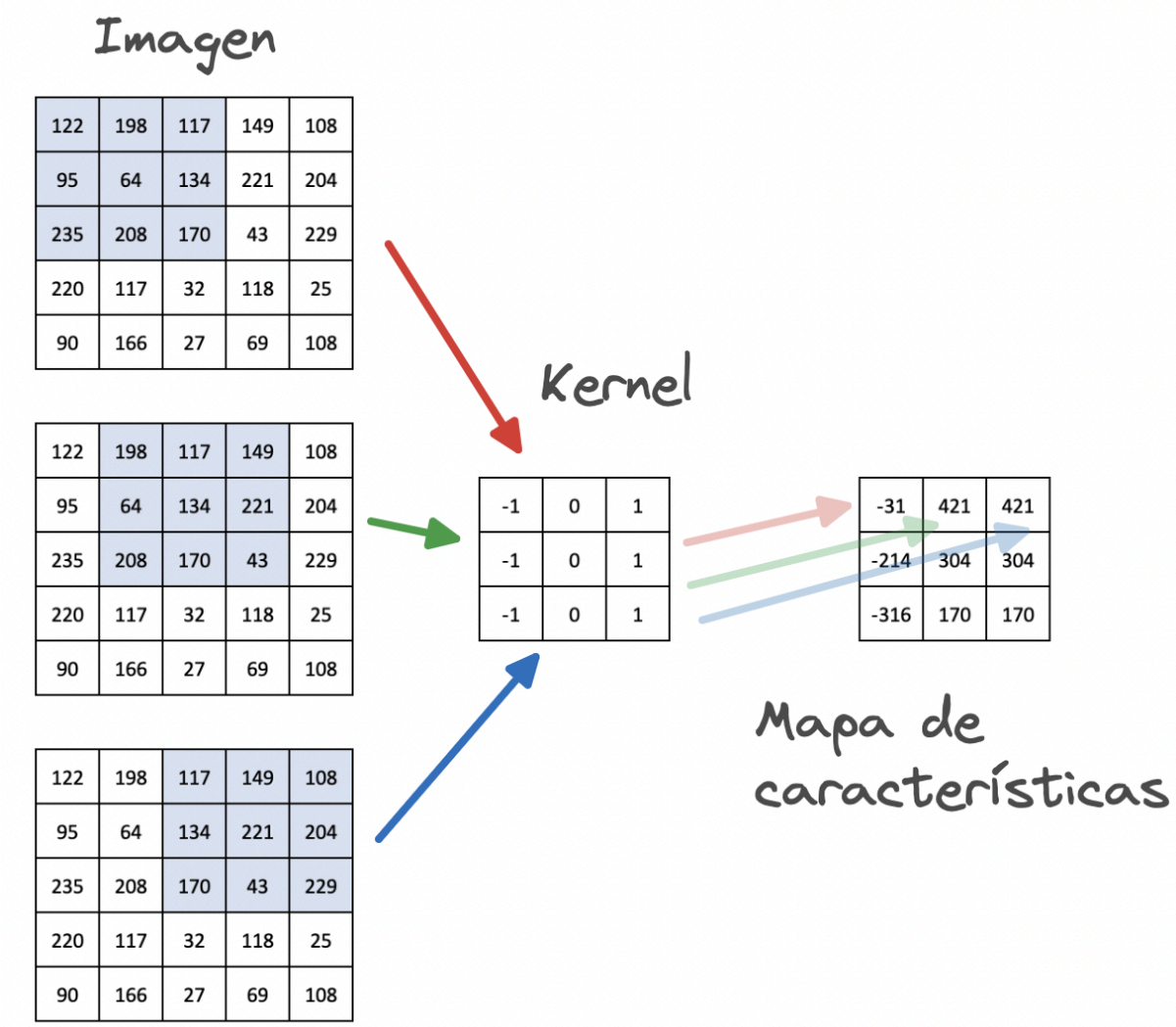

Figura 3. Operación de convolución
Fuente: Desafío Latam.

Las capas convolucionales en una red neuronal se construyen al aplicar un filtro sobre cada píxel de la imagen de entrada, de forma que los valores del kernel serán los parámetros o pesos que la red tendrá que aprender y se definen para cada capa convolucional una cantidad de filtros de acuerdo a la arquitectura que se esté desarrollando. Es necesario considerar que si la imagen cuenta con dos o más canales, entonces por cada canal tendremos un kernel que la red debe aproximar. Por ejemplo, si nuestra imagen de entrada es de 16x16 con tres canales, padding 2 y stride 1, si diseñamos una capa convolucional con 10 filtros de 5x5 entonces se necesitan aprender 10x5x3 parámetros + un bias por cada kernel.

Pooling

La operación de **Pooling**, o también conocida como **submuestreo**, es una técnica comúnmente utilizada en redes convolucionales para reducir el tamaño de los mapas de características de las salidas de las capas convolucionales y, al mismo tiempo preservar las características más relevantes.

El Pooling toma un mapa de características y aplica sobre este una matriz de dos dimensiones con cierto tamaño cuadrado para todos los valores correspondientes. Al poner esta matriz (también llamada **filtro**) sobre el mapa de características, Pooling resume todos estos valores del mapa de características con un sólo valor. La decisión de cómo resumir esta información se realiza de dos diferentes formas: tomando el promedio de todos los valores correspondientes (**Average Pooling)** o resumir usando el valor máximo de todos los valores del mapa de característica correspondiente, (**Max Pooling)**.

En la Figura 4 se muestra la operación de calcular un Máx Pooling y un Average Pooling ambos para filtro de 3x3 y un **STRIDE** de 1.

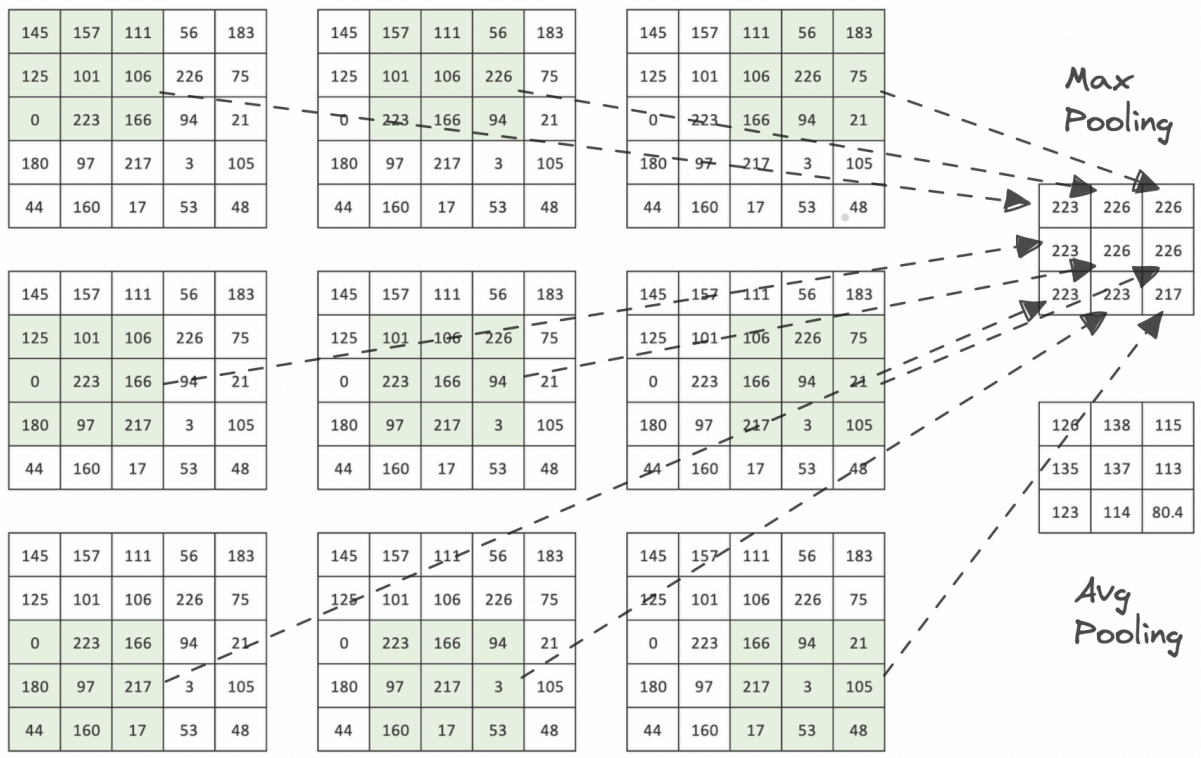

Figura 4. Operación de Máx Pooling y AVG Pooling
Fuente: Desafío Latam

Red Neuronal Convolucional AlexNet

En el año 2012 aparece la primera arquitectura de red neuronal convolucional conocida como AlexNet (desarrollada por Alex Kriszhevsky), que muestra ser mejor que otras técnicas clásicas empleadas para la clasificación de imágenes o reconocimiento de objetos. Esta red neuronal fue sometida al popular set de imágenes IMAGENET, que es considerado uno de los más desafiantes actualmente.

El **ILSVRC** o *ImageNet Large Scale Visual Recognition Challenge* es una competición que se desarrolla una vez al año organizada por el equipo de ImageNet, en la que diferentes equipos de investigación ponen a prueba sus algoritmos de reconocimiento visual. El conjunto de datos para usar en esta competición consta de 1000 clases o categorías diferentes, contabilizando aproximadamente de 1.2 millones con imágenes de tamaño 256x256.

En el año 2012, AlexNet logró un error del 16% (métrica top 5), logrando superar el 25% de error que se había conseguido hasta entonces.

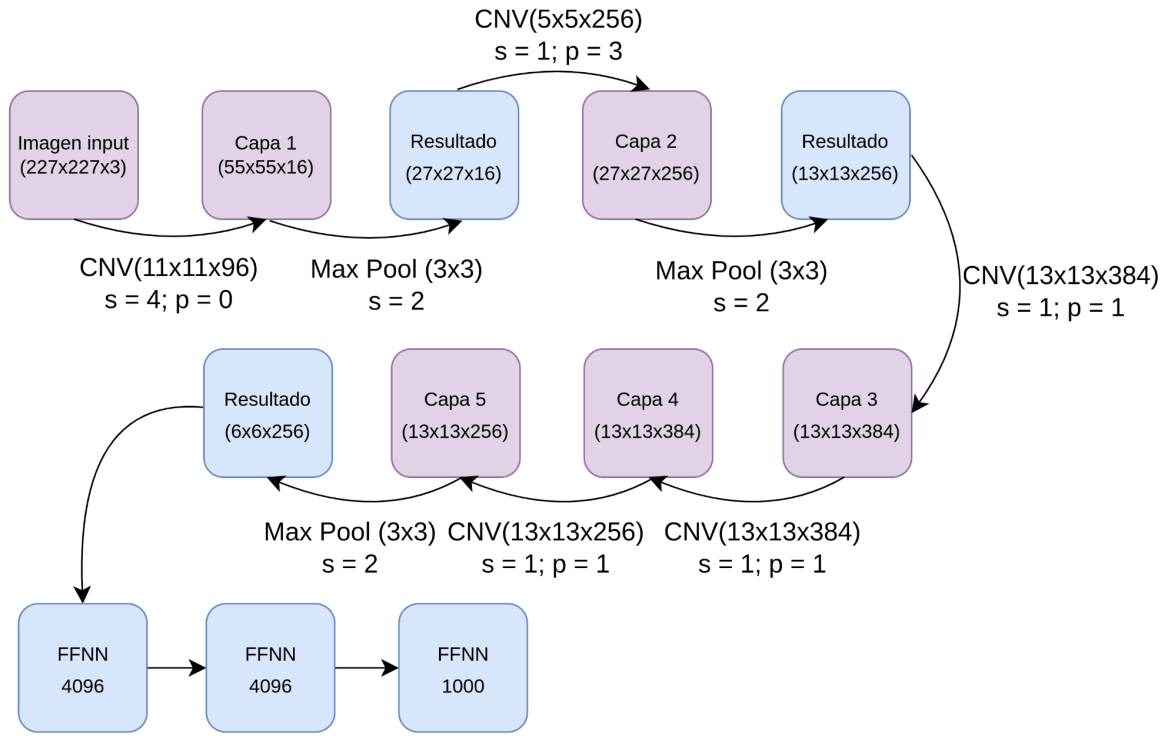

Figura 5. Arquitectura de AlexNet (año 2012)
Fuente: Desafío Latam

En la Figura 5, vemos que AlexNet se compone de 8 capas en total, con cinco de ellas de tipo convolucional y tres de tipo Densa. En cuanto a la cantidad de parámetros, estamos en el orden de 62 MM.

Luego de AlexNet se han implementado otras redes que han ido mejorando el error agregando ciertas mejoras. Entre ellas figuran:

- VGG-16, VGG-19
- Inception (GoogleNet)
- RestNet
- DenseNet

Actividad guiada: Desarrollo Red Neuronal AlexNet y aplicación.

Puedes analizar la aplicación de una red Neuronal AlexNet para la identificación de dígitos escritos a mano. Para ello, revisa el archivo 01 - Red CNN

Red Neuronal Convolucional VGG-16

Después del éxito de AlexNet se comenzó a mirar con mejores ojos a las redes neuronales para clasificación de imágenes, y producto de esto nace VGG-16 que supera a AlexNet con un 7.7% de error, en top-5 para la competencia **ILSVRC** del año 2015. Esta arquitectura plantea que las arquitecturas de redes neuronales convolucionales no deben tener como hiper parámetro los tamaños de los kernel en cada capa, sugiriendo dejarlo fijo en 3x3 junto con fijar el tamaño de max pooling en 2x2.

En la Figura se muestra la arquitectura VGG-16. Podemos observar que posee 16 capas, con 13 de tipo convolucional y tres Fully Connected. La cantidad de parámetros a estimar en esta red es de 138 MM.

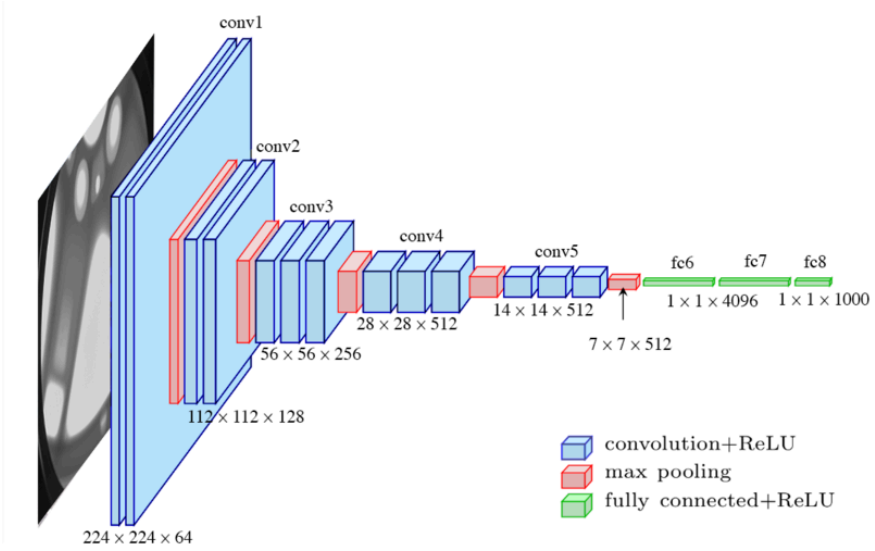

Figura 6. Arquitectura red neuronal VGG-16
Fuente: Automatic Localization of Casting Defects with Convolutional Neural Networks.

Redes convolucionales profundas

En general, esperamos que con redes neuronales más profundas podremos mejorar el rendimiento en la clasificación a costa de perder capacidad de generalización. Sin embargo, en la práctica nos encontraremos con que esto no ocurre.

Lo anterior se puede explicar considerando que con redes más profundas tendremos mayor capacidad de aproximación, pero para lograr sacar provecho de esta capacidad y mejorar el rendimiento tendremos que aplicar métodos que permitan entrenar eficientemente la red para poder aprovecharla. De esta forma vamos avanzando en arquitecturas CNN mejores cada vez, ya que vamos incorporando pequeños cambios que permiten mejorar la capacidad de entrenamiento en la red neuronal.

Actividad guiada: Desarrollo de Redes convolucionales

Puedes analizar la aplicación de una red Neuronal AlexNet para la identificación de dígitos escritos a mano. Para ello, revisa el archivo 01 - Red CNN

Implementación de redes neuronales convolucionales usando Keras y Tensorflow

La implementación de redes neuronales convolucionales seguirá la misma estructura que hemos usado hasta ahora en Redes Neuronales Feed-Forward Fully Connected. Lo nuevo es que debemos definir capas para realizar convoluciones llamadas **Conv2D** en Keras, y operaciones usando objetos de tipo capa llamados **MaxPooling2D** y/o **AvgPooling2D**.

Implementaremos redes neuronales convolucionales para realizar clasificaciones de imágenes que pertenecen al conjunto CIFAR-10 *Canadian Institute for Advanced Research*. Este dataset posee 60,000 imágenes, de 32x32x3 con canales RGB en que tenemos 50,000 para entrenamiento y el resto (10,000) para prueba, consta de 10 clases diferentes mutuamente excluyentes. Para verlo, utiliza el archivo 01 - Redes convolucionales

Preguntas de proceso

Reflexiona:

- ¿Qué es una convolución y para qué sirve?
- ¿Cual es el objetivo de la operación de Máx Pooling?
- ¿Mencione alguna problemática en la cual se puede aplicar CNN?
- ¿Cómo se relacionan las Redes Neuronales Convolucionales con las Redes Fully Connected?

Referencias bibliográficas

Deep Learning, Ian Goodfellow and Yoshua Bengio and Aaron Courville 2016.

M. Ferguson, R. Ak, Y. -T. T. Lee and K. H. Law, "Automatic localization of casting defects with convolutional neural networks," *2017 IEEE International Conference on Big Data (Big Data)*, Boston, MA, USA, 2017, pp. 1726-1735, doi: 10.1109/BigData.2017.8258115.

Alex Krizhevsky, Ilya Sutskever, Geoffrey E. Hinton, “ImageNet Classification with Deep Convolutional Neural Networks,” 2012 Published in Advances in Neural Information Processing Systems (NIPS 2012).