

HERRAMIENTA:

Aprendizaje Maquina y Profundo vs. Redes Neuronales Convolucionales



Autores

M.I.C. Carlos Abraham Carballo Monsivais

I.S.C. Leticia Edith Trujillo Ballesteros

C.L.M.A. Sacbe García García



Hackathon Blockchain 2020

Elaboración 2020, Primera edición



HERRAMIENTA. Aprendizaje Maquina y Profundo vs Redes Neuronales Convolucionales

Contenido

| | | |
|------|--|---|
| 4. | Aprendizaje Maquina y Profundo | 4 |
| 4.1. | Aprendizaje de Máquina | 4 |
| 4.2. | Redes Neuronales Artificiales | 4 |
| 4.3. | Aprendizaje Profundo | 8 |
| 4.4. | Redes Neuronales Convolucionales | 9 |



4. Aprendizaje Máquina y Profundo

4.1. Aprendizaje de Máquina

Dentro del área de la inteligencia artificial se encuentra una subárea llamada aprendizaje de máquina (ML, del inglés *Machine Learning*), la cual tiene como objetivo enseñar a las máquinas a resolver ciertos tipos de problemas complejos, para resolver estos problemas primero se le tiene que mostrar algunos ejemplos para que puedan aprender a resolverlos. El aprendizaje de máquina puede servir para automatizar un gran número de procesos que pueden ser repetitivos para los humanos. Entonces, el aprendizaje de máquina puede ser utilizado para construir sistemas de inteligencia artificial que pueden funcionar en entornos difíciles del mundo real.

En el aprendizaje de máquina se aplica la probabilidad y estadística, enfocándose principalmente en estimar funciones complicadas. Los algoritmos que se utilizan en el aprendizaje de máquina se dividen en dos principales categorías: aprendizaje supervisado y no supervisado. Los algoritmos de aprendizaje supervisado consisten en aprender ciertas características de un conjunto de datos que la máquina recibe, en donde cada conjunto de características está asociada a una etiqueta proporcionada por un experto o supervisor. Entonces, el algoritmo debe aprender de las características y relacionarlas con su etiqueta correspondiente. Por lo tanto, una vez que la máquina aprenda de las características debe tener la capacidad de predecir un resultado correcto cuando se le proporcionen nuevos datos. Por otro lado, los algoritmos de aprendizaje no supervisado consisten en aprender de conjuntos de datos que contienen muchas características y no se le da ninguna etiqueta. El algoritmo debe de identificar las similitudes que existen entre los datos y separarlas en diferentes agrupaciones. En general, estos algoritmos consisten en dividir los conjuntos de datos en grupos que tienen características similares.

Una de las principales técnicas del aprendizaje no supervisado es *K-Means* el cual se utiliza para dividir conjuntos de datos que no tienen etiquetas como es el caso de las criptomonedas.

Por otro lado, se ha aumentado la eficiencia de los algoritmos de aprendizaje de máquina con técnicas de aprendizaje profundo, el cual ha permitido aprender de características más abstractas al utilizar redes neuronales profundas.

4.2. Redes Neuronales Artificiales

Las redes neuronales artificiales (ANN del inglés, *Artificial Neural Networks*) tratan de simular el proceso de decisión de las células nerviosas o neuronas. La simulación que se realiza es de célula por célula y se basa en los conocimientos neurofisiológicos de las neuronas biológicas. Esta simulación de una red neuronal biológica es una arquitectura informática que permite realizar operaciones matemáticas simples, tales como: suma, multiplicación y elementos lógicos.

La unidad básica de una ANN es la neurona, la cual recibe la información que va a procesar y devolverá una salida. Las operaciones que realiza la neurona están dadas por la siguiente fórmula:

$$y = wX + b$$



donde y representa la salida de la operación, w es el peso, X representa la entrada y b es el *bias* o sesgo, entonces la entrada se multiplica con el peso y finalmente se le suma el *bias*.

Una neurona puede recibir varias entradas y cada una de estas entradas tiene su propio peso. Además, a la salida de la neurona se le aplica una función de activación para introducir la no linealidad, como se puede observar en la Figura 1. La función de activación mantiene la salida de la neurona entre ciertos límites. Entonces, una neurona artificial está representada de la siguiente forma:

$$y = wX + b$$

Las redes neuronales artificiales (ANN del inglés, *Artificial Neural Networks*) tratan de simular el proceso de decisión de las células nerviosas o neuronas. La simulación que se realiza es de célula por célula y se basa en los conocimientos neurofisiológicos de las neuronas biológicas. Esta simulación de una red neuronal biológica es una arquitectura informática que permite realizar operaciones matemáticas simples, tales como: suma, multiplicación y elementos lógicos.

La unidad básica de una ANN es la neurona, la cual recibe la información que va a procesar y devolverá una salida. Las operaciones que realiza la neurona están dadas por la siguiente formula:

$$y = f \left(\sum_{i=1}^n w_i X_i + b \right)$$

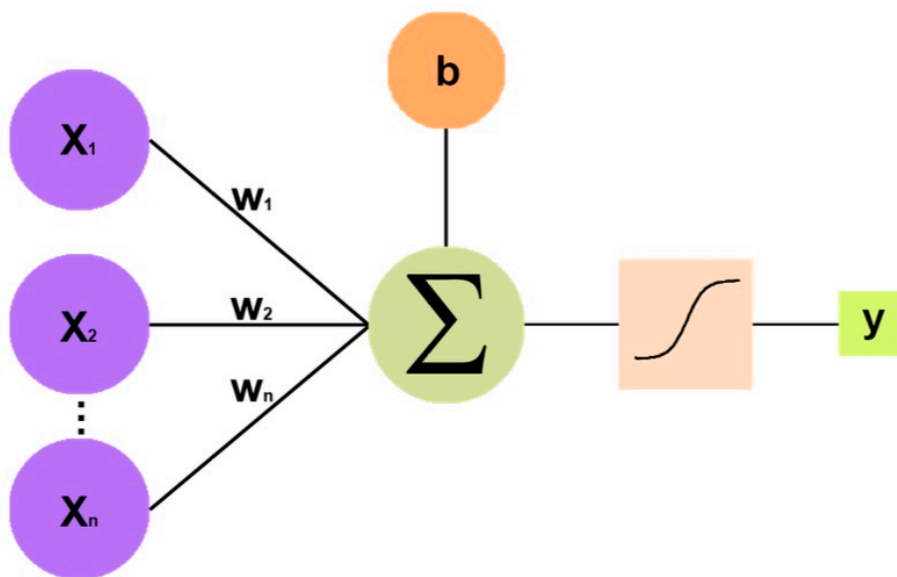


Figura 1. Enfoque del Aprendizaje Profundo



Existen diferentes tipos de funciones de activación, las más comunes son: *ReLU*, *Sigmoid*, *Tanh* y *Softmax*. En las redes neuronales modernas es comúnmente utilizada la función de activación de unidad lineal rectificadora (ReLU, del inglés *Rectified Linear Unit*).

La función *ReLU* transforma los valores negativos a 0 y los valores positivos pasan igual, es decir, solo se activa si los valores de entrada son positivos. Esta función de activación se comporta bien con imágenes, por lo tanto, tiene un buen desempeño cuando se trabaja con CNN. La función *ReLU* se define como:

$$ReLU(x) = \max \{0, x\}$$

La función de *Sigmoid* es utilizada en modelos de regresión logística y es la función más antigua. Esta función se satura cuando los argumentos son muy negativos o positivos, es decir, que la función se hace muy plana e insensible a pequeños cambios en la entrada. La función está acotada entre los valores 0 y 1. La función *Sigmoid* se define como:

$$Sigmoid(x) = \frac{1}{1 + e^{-x}}$$

La función *Tanh* es similar a la función de *Sigmoid*, pero con la diferencia de que está acotada entre los valores de -1 y 1. Esta función es utilizada cuando se tiene que predecir entre una opción totalmente diferente a otra y tienen un buen desempeño cuando se usa en redes recurrentes. La función *Tanh* se define como:

$$Tanh(x) = \frac{2}{1 + e^{-2x}} - 1$$

Finalmente, la función *Softmax* es utilizada en la capa final de la red neuronal y es una versión de la función logística pero para más de dos clases, por lo tanto, sirve para realizar clasificación multiclase. La salida de esta es un vector de probabilidades de cada una de las clases, por lo que la suma de todas las probabilidades es igual a 1. La función *Softmax* se define como:

$$Softmax(x)_j = \frac{e^{x_j}}{\sum_{i=1}^n e^{x_i}}$$



Las ANN están compuestas por estas neuronas y están conectas entre sí. Una ANN contiene múltiples capas de neuronas, principalmente de tres tipos de capas, las cuales son: la capa de entrada, oculta y de salida, como se muestra en la Figura 2.

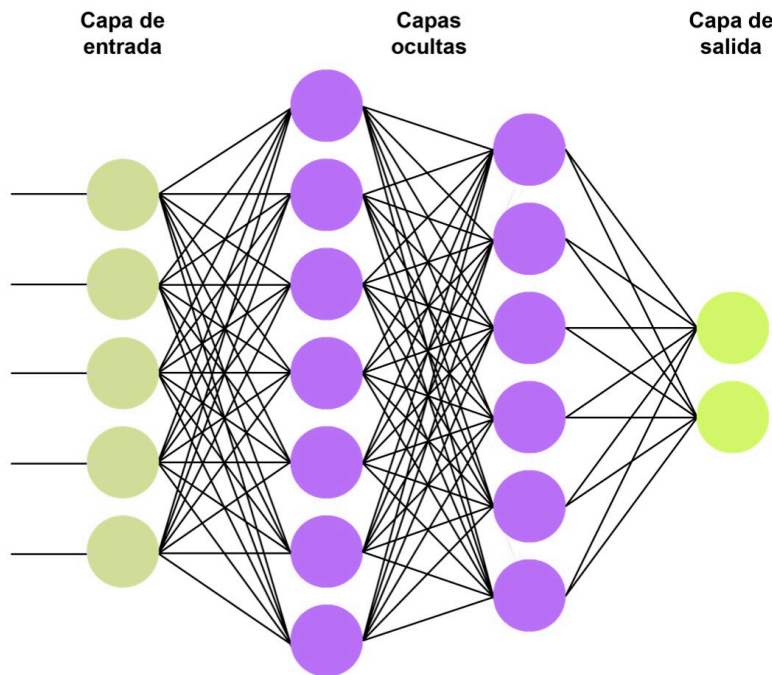


Figura 2. Enfoque del ANN

La capa de entrada es la primera capa de la ANN y es la encargada de recibir toda la información que se va a procesar. La capa oculta es la capa interna de la red, no tienen contacto directo con el entorno exterior y esta capa es donde se realiza el procesamiento de la información. Cabe mencionar que una ANN puede tener más de una capa oculta. Finalmente, la capa de salida proporciona la información generada por la ANN, es decir, provee la respuesta de la predicción. Cada una de las diferentes capas puede estar compuesta por más de una neurona, en donde cada neurona está representada por un círculo de la Figura 2.

Con los avances que han existido en las redes neuronales y la evolución tecnológica en los sistemas de procesamiento como las unidades de procesamiento gráfico han ayudado a que las redes neuronales sigan mejorando y realizando tareas cada vez más complejas.

Con estos logros han surgido nuevas técnicas mejoradas en donde a las redes neuronales se le han agregado un mayor número de capas ocultas para mejorar los resultados que no se pueden obtener con las redes neuronales simples. Con esto surge el aprendizaje profundo.

4.3. Aprendizaje Profundo

El aprendizaje profundo (DL, del inglés *Deep Learning*) es una especie de aprendizaje que se realiza por medio de redes neuronales artificiales profundas. Estas redes neuronales forman parte de la subárea del aprendizaje de máquina. El aprendizaje profundo se refiere a las arquitecturas que tienen varias capas ocultas para aprender diferentes características con múltiples niveles de abstracción. El aprendizaje profundo proporciona un mayor aprendizaje debido a la gran cantidad de capas y número de neuronas en cada una de las capas.

En la Figura 3, se pueden observar algunas diferencias entre el aprendizaje de máquina y el aprendizaje profundo, donde se muestra un fácil ejemplo de clasificación entre un perro y un gato. Primero se puede observar que se realiza la clasificación con técnicas del aprendizaje de máquina, en donde una persona experta en el contexto en el que se está trabajando selecciona el tipo de características que se van a extraer de los datos de entrada. Posteriormente, el experto ingresa las características de los datos a un algoritmo de clasificación como puede ser un árbol de decisiones el cual se encarga de realizar las predicciones cuando se ingresen nuevos datos.

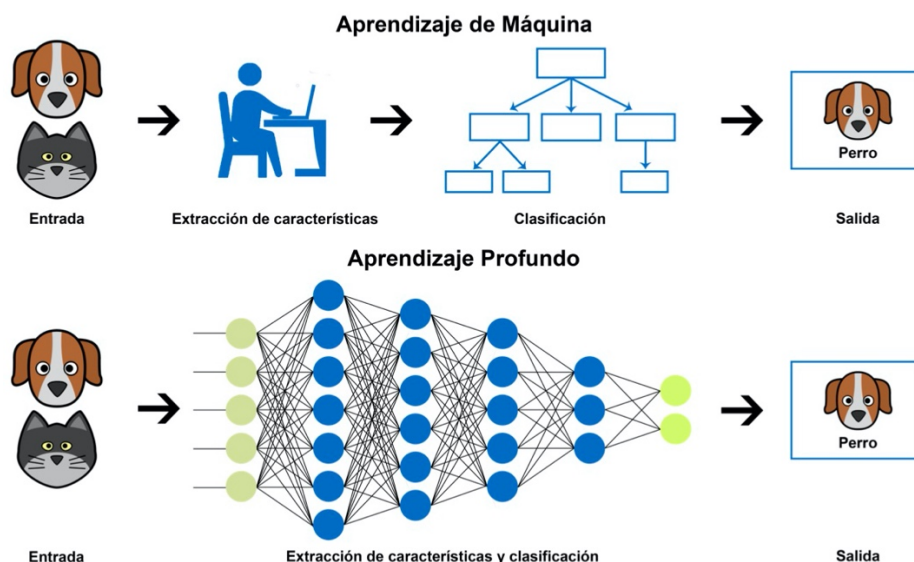


Figura 3. Enfoque del Aprendizaje Profundo

En cambio, con el aprendizaje profundo se ahorra la parte de la extracción de características, debido a que la red neuronal profunda se encarga de hacer este trabajo al aprender las características por sí misma. Entonces, los datos de entrada se ingresan tal cual a la red neuronal profunda. Por lo tanto, la red se encarga de extraer las características a través de todas las capas que tiene y a su vez realiza la clasificación. De igual manera se pueden generar predicciones cuando se ingresen nuevos datos. Las redes neuronales profundas se han vuelto populares por su gran aplicación y la cantidad de problemas que pueden llegar a resolver, tales como el reconocimiento de rostros donde utilizan redes neuronales convolucionales.



Una de las áreas que se ha visto muy beneficiada con estas técnicas es la medicina, dado que existe un gran número de aplicaciones para resolver problemas tales como: detección de lesiones mamográficas, segmentación en lesiones en el cerebro, clasificación de la retinopatía diabética, clasificación de lesiones cutáneas, supresión ósea de última generación en rayos X, entre otras aplicaciones más.

Otras de las aplicaciones que se han realizado es en el procesamiento del lenguaje natural y predicciones en series de tiempo en donde se utilizan redes neuronales recurrentes. También, se ha realizado la creación de nuevos datos a partir de datos mediante redes neuronales generativas adversarias (GAN, del inglés *Generative Adversarial Networks*), donde han creado imágenes de rostros de personas que no son reales. Una de las técnicas más conocidas en el área del aprendizaje profundo son las redes neuronales convolucionales, las cuales son empleadas en este proyecto BITMONEY.

Finalmente, en este proyecto BITMONEY se implementan los dos enfoques de aprendizaje supervisado y no supervisado. El aprendizaje no supervisado se aplica al momento de realizar el etiquetado de los datos de las criptomonedas, debido a que utiliza el algoritmo *K-Means*. El aprendizaje supervisado se realiza cuando se implementa una CNN para obtener la recomendación para los inversores.

4.4. Redes Neuronales Convolucionales

Las redes neuronales convolucionales (CNN, del inglés *Convolutional Neural Networks*) están diseñadas para trabajar con imágenes y se aplican a la visión por computadora. La convolución es una operación matemática que existe entre los píxeles de la imagen con un filtro o filtros de cierto tamaño también conocido como máscara o *kernel*, y que a su vez estos filtros pasan por toda la imagen. La convolución a una imagen se aplica de izquierda a derecha y de arriba hacia abajo. Como resultado de la convolución se obtiene una imagen distinta. En la Figura 4 se puede observar el proceso que realiza una convolución a una imagen. Se tiene una imagen de un tamaño de 5×5 en donde se le aplica un filtro de un tamaño de 3×3 . El filtro se coloca en la parte superior izquierda de la imagen y realiza la siguiente operación:

$$(5 * 0) + (10 * 1) + (2 * 0) + (1 * 1) + (2 * 1) + (4 * 1) + (5 * 0) + (4 * 1) + (3 * 0) = 21$$

El resultado de la operación es igual a 21, que formará parte del primer dato de la nueva imagen. Este proceso se repite por toda la imagen y el filtro se va recorriendo. Al final se obtiene una imagen de un tamaño de 4×4 . Un ejemplo de un filtro de convolución que se puede aplicar para obtener los bordes de los objetos que se encuentran en una imagen.

Una CNN puede estar compuesta por más de una capa de convolución, donde estas convoluciones se aplican a las imágenes de entrada de una red para extraer las características y a este conjunto de características se le conoce como mapa de características.



BITMONEY

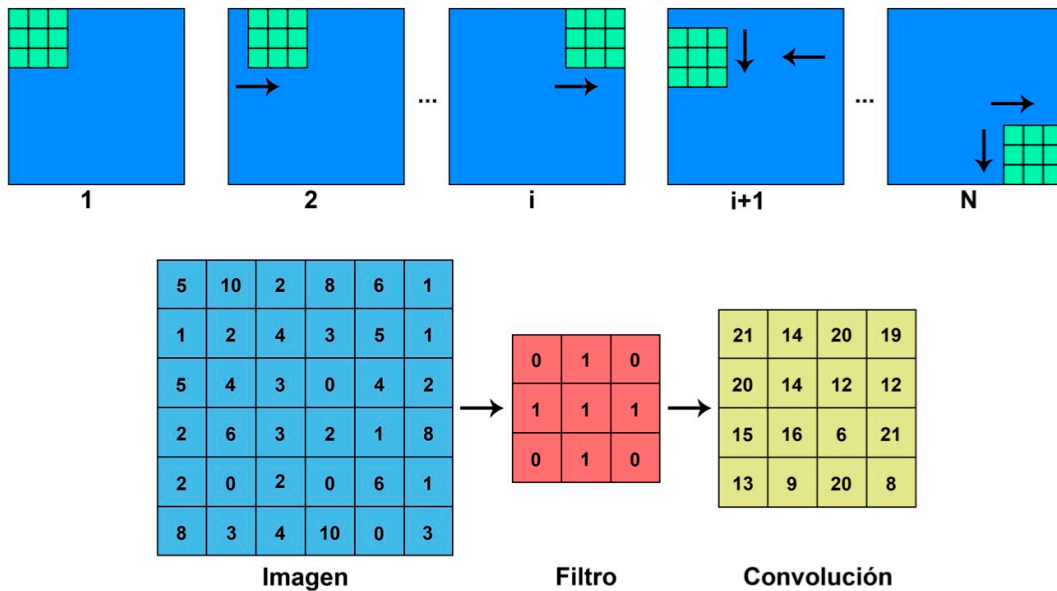


Figura 4. Funcionamiento de una convolución

Otra de las capas que se aplican en las CNN es la capa de submuestreo que consiste en reducir el tamaño del mapa de características y al mismo tiempo se reduce la complejidad de los cálculos. De igual manera que la capa de convolución, también es una operación matemática que se realiza de izquierda a derecha y de arriba hacia abajo con un filtro de un tamaño en específico.

Existen dos tipos principales de capas de submuestreo, las cuales son *max pooling* y *average pooling*. La capa de *max pooling* realiza el cálculo del valor máximo en cada parte donde se sitúa el filtro en el mapa de características. Por otro lado, la capa de *average pooling* calcula el promedio en cada parte donde se sitúa el filtro en el mapa de características.

En la figura 5 se muestra un ejemplo del funcionamiento de estos dos tipos de capas, primero se observa *max pooling* en donde se tiene el mapa de características de 4x4 de tamaño y se coloca el filtro de 2x2 de tamaño en la parte superior izquierda (cuadrícula en color rojo), entonces entre los datos que encerró ese filtro se selecciona el número mayor. Por ejemplo, los datos que abarco el filtro son el 55, 20, 45 y 80, por lo tanto, el valor máximo es el 80. Ese número 80 forma parte de un nuevo mapa. Eso se realiza sucesivamente por todo el mapa de características de izquierda a derecha y de arriba hacia abajo. Por otra parte, al mismo mapa de características se le aplica el *average pooling*, pero ahora los datos 55, 20, 45 y 80 se promedian dando como resultado 50. Es así como se realiza la reducción de los mapas de características generados por las capas de convoluciones, en donde se pasó de un mapa de 4 x 4 a un mapa de 2 x 2, es decir, se redujo a la mitad.

Generalmente una CNN está compuesta por una capa de convolución seguida de una capa de submuestreo y así sucesivamente hasta formar toda una arquitectura más compleja de capas y capas.

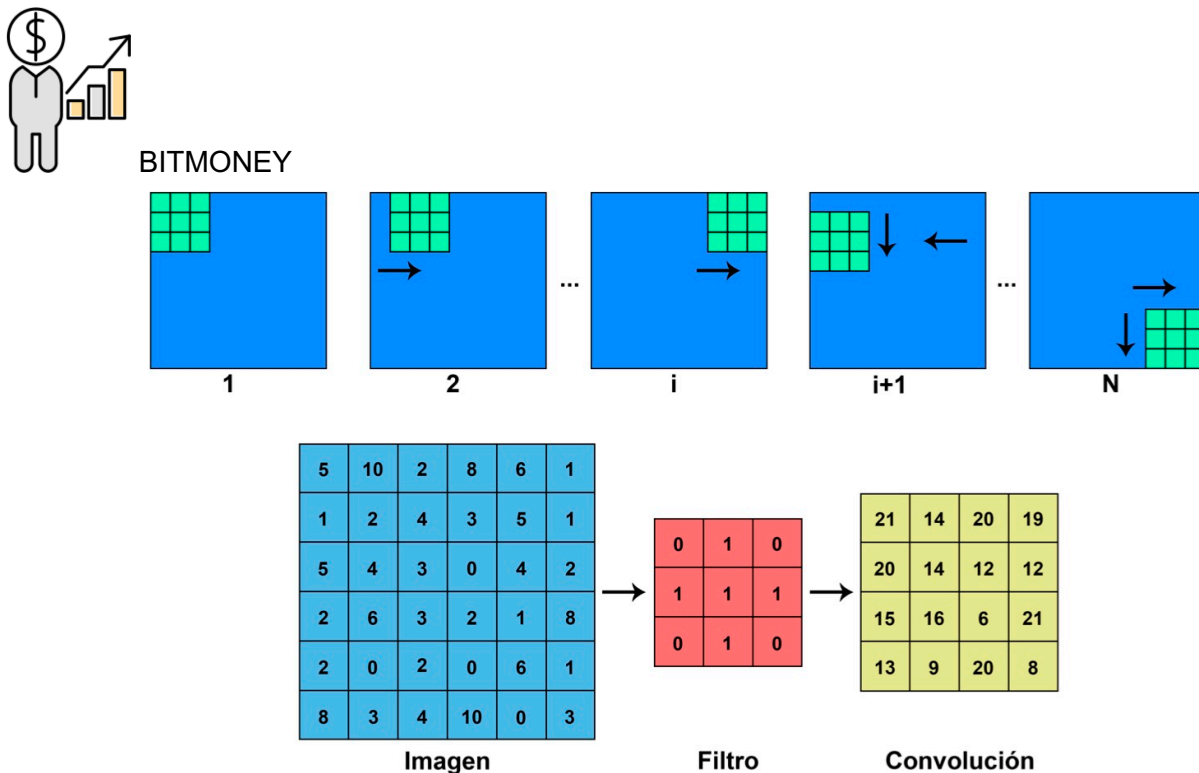


Figura 5. Proceso de submuestreo con *max pooling* y *average pooling* donde se puede observar que se aplica un filtro de 2×2 al mapa de características

Una vez realizadas las convoluciones y los submuestreos, y se obtienen los mapas de características finales, se aplica otro tipo de capa, llamada capa completamente conectada (FC, del inglés *Fully Connected*). Esta capa es similar a la capa de una red neuronal simple, donde cada neurona de esta capa es considerada un píxel de la imagen. Los mapas de características finales se aplanan para alimentar a la capa completamente conectada. En la Figura 6, tomando el ejemplo de la capa de *max pooling*, el mapa de características reducido de tamaño 2×2 se aplanan y se obtiene un vector de 4 dimensiones. Entonces, este vector alimenta a una capa completamente conectada y esta a su vez puede ir conectada con otra capa igual. Estas capas junto con la capa de salida que también es una capa completamente conectada se encargan de realizar la clasificación de la etiqueta de la imagen. Todo este proceso se repite para cada una de las imágenes del conjunto de datos para completar el entrenamiento de la CNN.

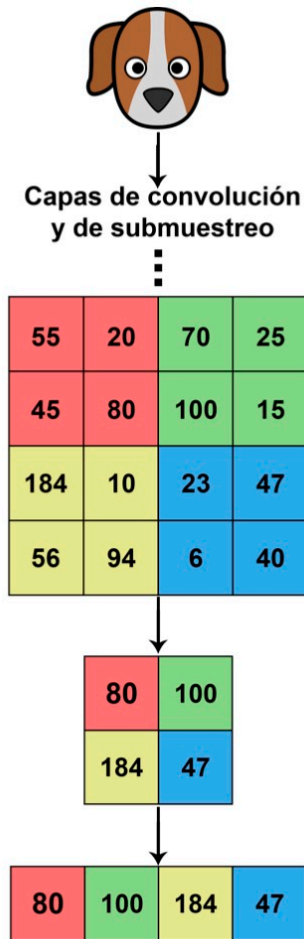
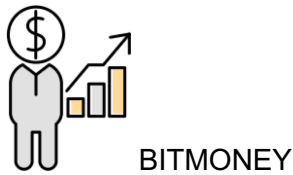


Figura 6. Aplanado de los datos de los mapas de características para alimentar a una capa completamente conectada.

Harley creó un visualizador interactivo de CNN que reconoce números manuscritos del 0 al 9. Este visualizador fue creado con el fin de mostrar el funcionamiento interno que ocurre en una CNN entrenada cuando se ingresa una nueva entrada. En la Figura 7, se muestra un ejemplo más claro del funcionamiento de una CNN. Esta CNN está compuesta por dos capas de convolución, dos capas de submuestreo, dos capas totalmente conectadas y finalmente la capa de salida. Entonces, se puede observar que cada vez que se realiza una convolución se obtienen características cada vez más abstractas.

Existen varias arquitecturas de CNN que a lo largo del tiempo han surgido y que han servido de base en muchos de los logros en el campo de la visión por computadora. LeNet-5 es una de las primeras arquitecturas reportadas en la literatura, es una arquitectura poco compleja, debido a que tiene solo dos capas de convolución y tres capas completamente conectadas.

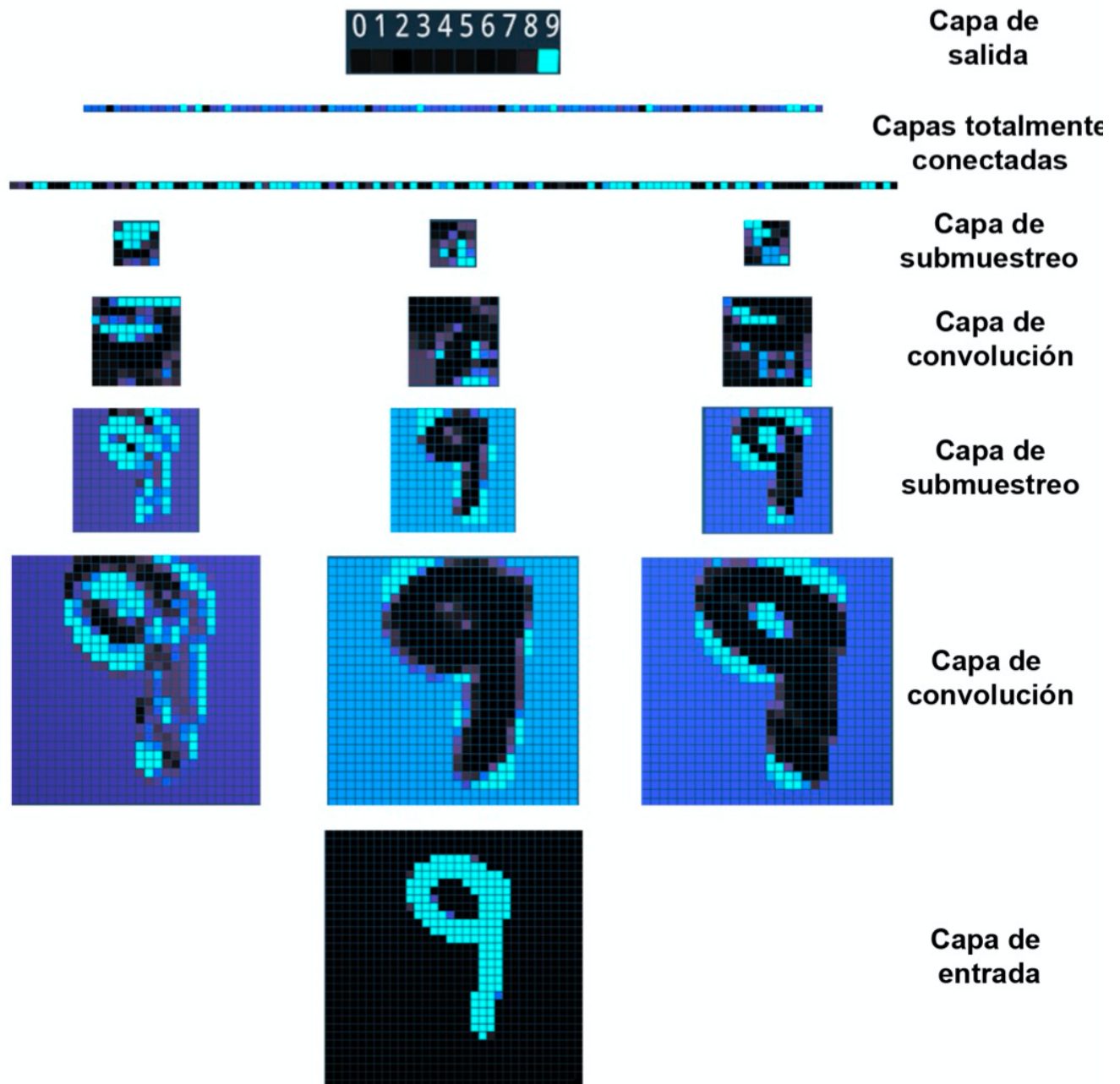
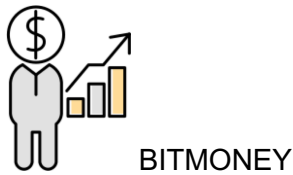


Figura 7. Visualización de una CNN entrenada con números manuscritos del 0 al 9 realizada, donde al ingresar un nuevo dato de entrada genera una predicción.



Después, surge otra arquitectura llamada AlexNet, la cual incrementa un poco su complejidad que la arquitectura de LeNet-5, es decir, es más profunda. La arquitectura de AlexNet consta de cinco capas de convolución y tres capas completamente conectadas.

Luego proponen una nueva arquitectura llamada VGG, la cual tiene diferentes números de capas, tales como once, trece, dieciséis y diecinueve capas de convolución y tres capas completamente conectadas. También existe la arquitectura de GoogleNet o también conocida como InceptionV1, la cual contiene veintidós capas. En esta CNN se usaron bloques densos, en los cuales existen capas convolucionales y se añaden para buscar mejorarlas. Posteriormente surge una nueva versión denominada InceptionV3, donde realizaron mejoras a la versión anterior con el fin de evitar cuellos de botella y hacer cálculos más eficientes. Por último, surge InceptionV4, con mejoras a InceptionV3, las principales son que se realizaron conexiones residuales para tener una mejor velocidad de entrenamiento.

Otra de las arquitecturas que han sido publicadas es ResNet, en donde contiene diferentes números de capas, como dieciocho, treinta y cuatro, cincuenta, ciento uno, y ciento cincuenta y dos. Con esta CNN se presenta un aprendizaje residual para ayudar a las redes que son muy profundas.