

Convolutional Neural Networks

Dr. Mauricio Toledo-Acosta

Diplomado Ciencia de Datos con Python

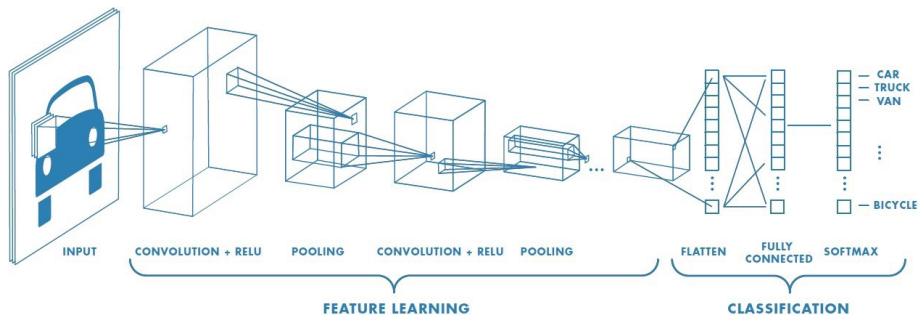
Table of Contents

1 Convolutional Neural Networks

- Convolution Kernels
- Pooling Layers

2 Further Applications

CNN



Source

Why CNNs?

- Una imagen no es más que una matriz de valores de píxeles. **¿Por qué no basta con aplanar la imagen y alimentar la MLP para tareas de clasificación?**

Why CNNs?

- Una imagen no es más que una matriz de valores de píxeles. **¿Por qué no basta con aplanar la imagen y alimentar la MLP para tareas de clasificación?**
- Para imágenes muy básicas, este enfoque puede exhibir un desempeño razonable al realizar tareas de clasificación, pero **tendría poca precisión cuando se trata de imágenes complejas que tienen dependencias entre píxeles.**

Why CNNs?

- Una imagen no es más que una matriz de valores de píxeles. **¿Por qué no basta con aplanar la imagen y alimentar la MLP para tareas de clasificación?**
- Para imágenes muy básicas, este enfoque puede exhibir un desempeño razonable al realizar tareas de clasificación, pero **tendría poca precisión cuando se trata de imágenes complejas que tienen dependencias entre píxeles.**
- **Una CNN es capaz de capturar las dependencias espaciales de una imagen mediante la aplicación de filtros.** La arquitectura se ajusta mejor al conjunto de datos de la imagen gracias a la reducción del número de parámetros implicados.

CNN

Convolutional Neural Networks son muy similares a las redes neuronales ordinarias:

- Estan hechas de neuronas que aprenden pesos y sesgos.

CNN

Convolutional Neural Networks son muy similares a las redes neuronales ordinarias:

- Estan hechas de neuronas que aprenden pesos y sesgos.
- Cada neurona recibe una entrada, realiza un producto punto y tiene una activación no lineal.

CNN

Convolutional Neural Networks son muy similares a las redes neuronales ordinarias:

- Estan hechas de neuronas que aprenden pesos y sesgos.
- Cada neurona recibe una entrada, realiza un producto punto y tiene una activación no lineal.
- La red recibe las imágenes como conjuntos de píxeles en un lado y produce scores de clases en el otro.

CNN

Convolutional Neural Networks son muy similares a las redes neuronales ordinarias:

- Estan hechas de neuronas que aprenden pesos y sesgos.
- Cada neurona recibe una entrada, realiza un producto punto y tiene una activación no lineal.
- La red recibe las imágenes como conjuntos de píxeles en un lado y produce scores de clases en el otro.
- La red tiene una función de pérdida.

CNN

Convolutional Neural Networks son muy similares a las redes neuronales ordinarias:

- Estan hechas de neuronas que aprenden pesos y sesgos.
- Cada neurona recibe una entrada, realiza un producto punto y tiene una activación no lineal.
- La red recibe las imágenes como conjuntos de píxeles en un lado y produce scores de clases en el otro.
- La red tiene una función de pérdida.
- Los pesos se actualizan con descenso de gradiente u otros optimizadores.

Layers of a CNN

Una red CNN se compone principalmente de tres tipos de capas:

- Capas convolucionales (convolutional).

Layers of a CNN

Una red CNN se compone principalmente de tres tipos de capas:

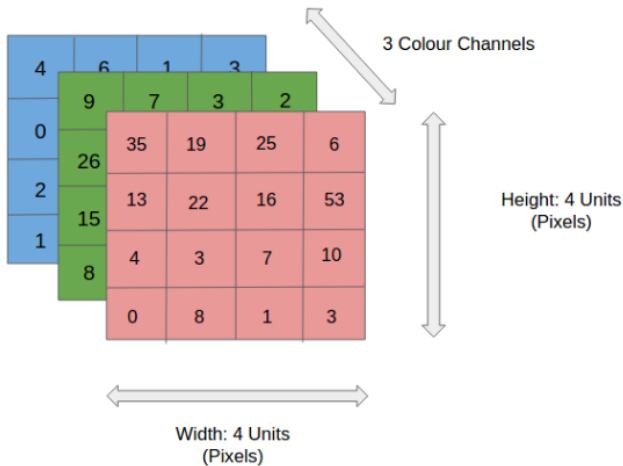
- Capas convolucionales (convolutional).
- Capas de pooling (pooling)

Layers of a CNN

Una red CNN se compone principalmente de tres tipos de capas:

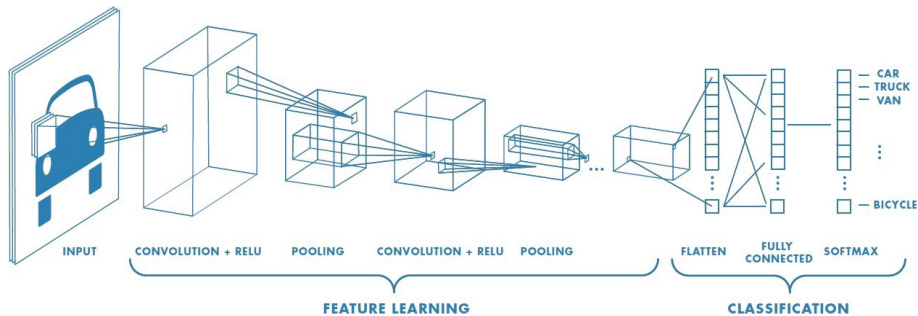
- Capas convolucionales (convolutional).
- Capas de pooling (pooling)
- Capas totalmente conectadas (fully connected).

CNN



Source

CNN



Convolution Kernels

- También llamada matriz de convolución o máscara.

Convolution Kernels

- También llamada matriz de convolución o máscara.
- Esta matriz es utilizada para transformar los valores de la imagen por medio de los valores del kernel.

Convolution Kernels

- También llamada matriz de convolución o máscara.
- Esta matriz es utilizada para transformar los valores de la imagen por medio de los valores del kernel.
 - Es cuadrada y pequeña (3×3 , 5×5).

Convolution Kernels

- También llamada matriz de convolución o máscara.
- Esta matriz es utilizada para transformar los valores de la imagen por medio de los valores del kernel.
 - Es cuadrada y pequeña (3×3 , 5×5).
 - Cuanto más grande es la matriz, más información local se pierde.

Convolution Kernels

- También llamada matriz de convolución o máscara.
- Esta matriz es utilizada para transformar los valores de la imagen por medio de los valores del kernel.
 - Es cuadrada y pequeña (3×3 , 5×5).
 - Cuanto más grande es la matriz, más información local se pierde.
- Permite efectos de *área* como desenfoque, nitidez y detección de bordes.

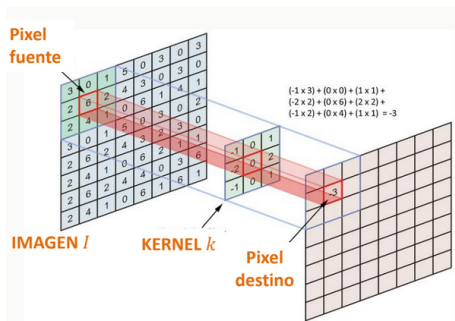
Convolution Kernels

- También llamada matriz de convolución o máscara.
- Esta matriz es utilizada para transformar los valores de la imagen por medio de los valores del kernel.
 - Es cuadrada y pequeña (3×3 , 5×5).
 - Cuanto más grande es la matriz, más información local se pierde.
- Permite efectos de *área* como desenfoque, nitidez y detección de bordes.
- No es una multiplicación de matrices.

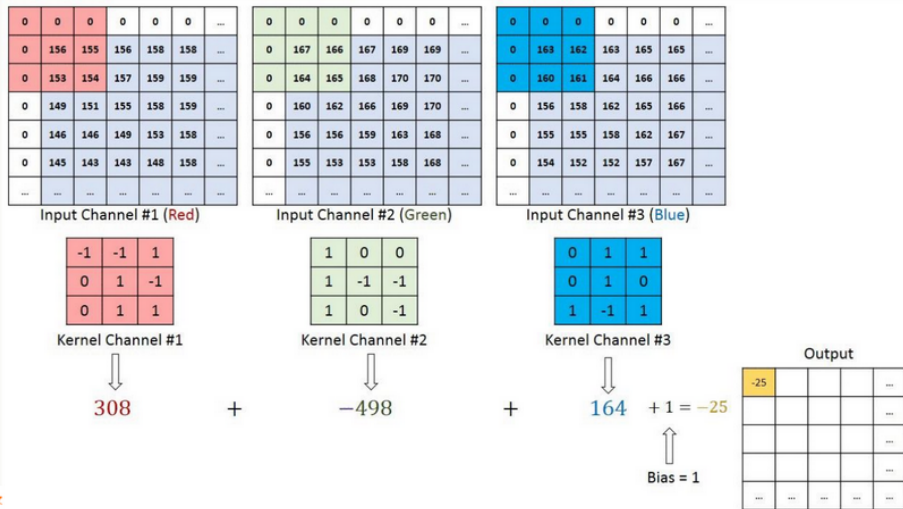
Convolution Kernels

Al aplicar el kernel de convolución k a una entrada (i, j) de la imagen I , esta entrada se transforma en

$$I_{i,j} = \sum_{x,y=1}^n I_{x-i,y-j} k_{x,y}$$



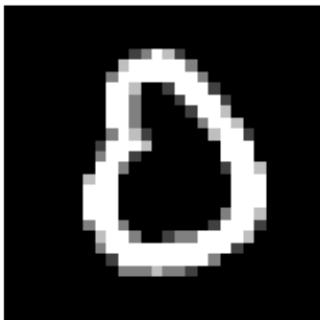
Convolution



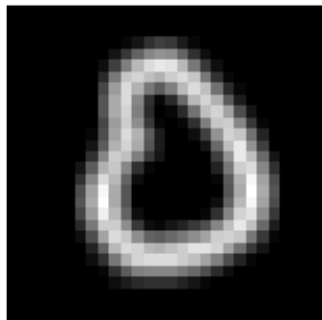
Convolution: Examples

$$K = \frac{1}{9} \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}$$

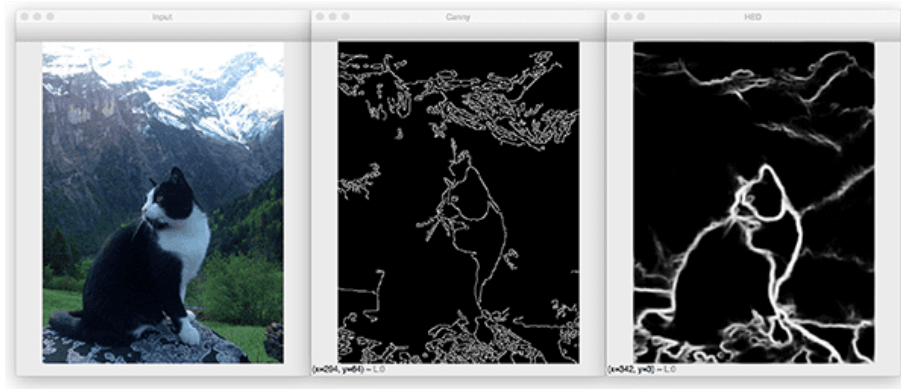
Before



After



State of the Art: Edge Detection



Holistically-Nested Edge Detection

Pooling Layers

- Uno de los principales objetivos de las CNN es aprender los filtros (kernels).

Pooling Layers

- Uno de los principales objetivos de las CNN es aprender los filtros (kernels).
- Los filtros sirven para detectar rasgos discriminativos en imágenes.

Pooling Layers

- Uno de los principales objetivos de las CNN es aprender los filtros (kernels).
- Los filtros sirven para detectar rasgos discriminativos en imágenes.
- El problema con este enfoque es que el proceso es que es sensible a la ubicación donde se encuentren estos rasgos.

Pooling Layers

- Uno de los principales objetivos de las CNN es aprender los filtros (kernels).
- Los filtros sirven para detectar rasgos discriminativos en imágenes.
- El problema con este enfoque es que el proceso es que es sensible a la ubicación donde se encuentren estos rasgos.
- Una solución a esto es subsamplear estas salidas para hacerlas más robustas al cambio de posición en la imagen.

Pooling Layers

- Uno de los principales objetivos de las CNN es aprender los filtros (kernels).
- Los filtros sirven para detectar rasgos discriminativos en imágenes.
- El problema con este enfoque es que el proceso es que es sensible a la ubicación donde se encuentren estos rasgos.
- Una solución a esto es subsamplear estas salidas para hacerlas más robustas al cambio de posición en la imagen.
- Aquí es donde entran las capas de **pooling**. El pooling se aplica después de una capa de convolución.

Pooling

La operación de pooling consiste en subsamplear la imagen de entrada. Esta operación se especifica, en vez de aprenderse.

Pooling

La operación de pooling consiste en subsamplear la imagen de entrada. Esta operación se especifica, en vez de aprenderse. Las dos maneras típicas que se usan son:

- Average Pooling: Calcula el valor promedio para cada porción de la imagen.
- Maximum Pooling (Max Pooling): Calcula el valor máximo para cada porción de la imagen.

Pooling

La operación de pooling consiste en subsamplear la imagen de entrada. Esta operación se especifica, en vez de aprenderse. Las dos maneras típicas que se usan son:

- Average Pooling: Calcula el valor promedio para cada porción de la imagen.
- Maximum Pooling (Max Pooling): Calcula el valor máximo para cada porción de la imagen.

El tamaño de la operación de pooling es más pequeña que el tamaño de la imagen; casi siempre es de 2×2 píxeles con un paso de 2 píxeles. En este caso, se reduce el tamaño a la mitad. Por ejemplo, una capa de pooling aplicada a una imagen de 6×6 resultará en una salida de 3×3 .

Pooling

Max Pooling

29	15	28	184
0	100	70	38
12	12	7	2
12	12	45	6

2 x 2
pool size

100	184
12	45

Average Pooling

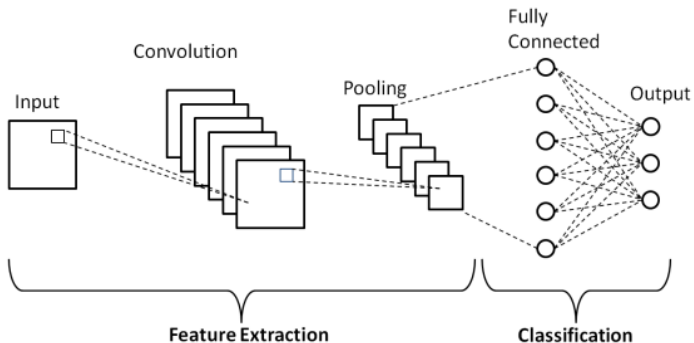
31	15	28	184
0	100	70	38
12	12	7	2
12	12	45	6

2 x 2
pool size

36	80
12	15

Example

Consideremos el siguiente ejemplo de CNN



Example

- La capa INPUT $[32 \times 32 \times 3]$ tendrá los valores de los pixeles de la imagen, en este caso la imagen es de tamaño 32×32 , con tres canales R, G, B.

Example

- La capa INPUT $[32 \times 32 \times 3]$ tendrá los valores de los pixeles de la imagen, en este caso la imagen es de tamaño 32×32 , con tres canales R, G, B.
- La capa CONV calculará la convolución con cada filtro. Esto resultará en un volumen de imágenes $[32 \times 32 \times 6]$ si es que usamos 12 filtros. A cada salida aplicamos la activación RELU, seguimos teniendo $([32 \times 32 \times 6])$.

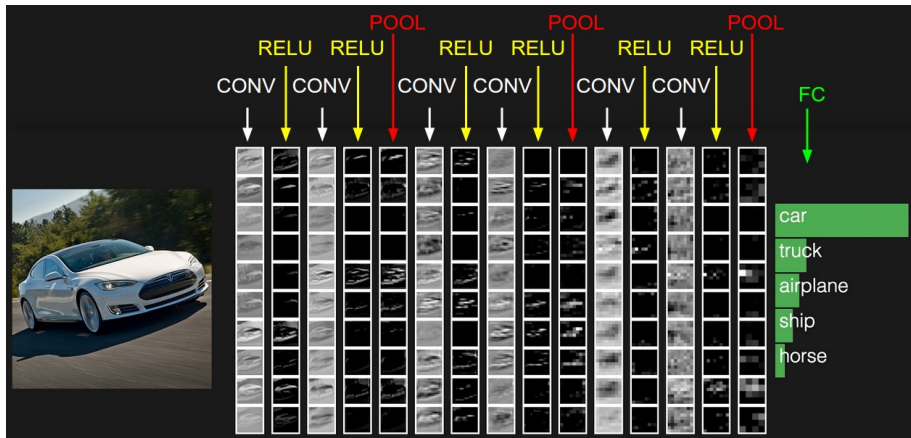
Example

- La capa INPUT $[32 \times 32 \times 3]$ tendrá los valores de los pixeles de la imagen, en este caso la imagen es de tamaño 32×32 , con tres canales R, G, B.
- La capa CONV calculará la convolución con cada filtro. Esto resultará en un volumen de imágenes $[32 \times 32 \times 6]$ si es que usamos 12 filtros. A cada salida aplicamos la activación RELU, seguimos teniendo $[32 \times 32 \times 6]$.
- La capa POOL subsampla a lo largo del ancho y largo de las imágenes, el resultado es de tamaño $[16 \times 16 \times 6]$.

Example

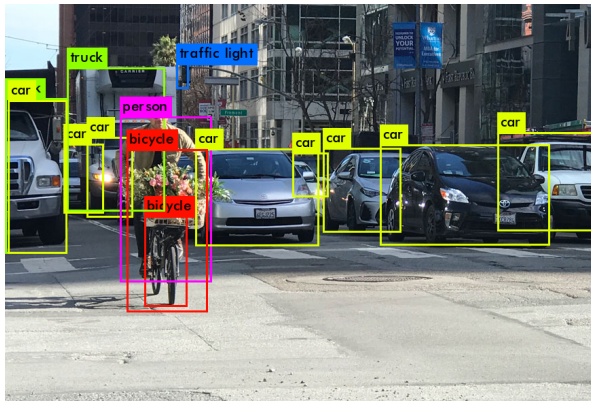
- La capa INPUT $[32 \times 32 \times 3]$ tendrá los valores de los pixeles de la imagen, en este caso la imagen es de tamaño 32×32 , con tres canales R, G, B.
- La capa CONV calculará la convolución con cada filtro. Esto resultará en un volumen de imágenes $[32 \times 32 \times 6]$ si es que usamos 12 filtros. A cada salida aplicamos la activación RELU, seguimos teniendo $([32 \times 32 \times 6])$.
- La capa POOL subsamplea a lo largo del ancho y largo de las imágenes, el resultado es de tamaño $[16 \times 16 \times 6]$.
- El volumen de datos anterior se aplana y entra a la red FC, la cuál calculará los scores de clase resultando en un volumen de tamaño $[N_{\text{on}} \times 3 \times 1]$.

Classification



Real-Time Object Detection

YOLO: You only look once



Video demo
Source

Table of Contents

1 Convolutional Neural Networks

- Convolution Kernels
- Pooling Layers

2 Further Applications

Natural Language Processing

Yesterday I had a very good day with my son

Text message

Yesterday had very good day son

Text without stop-words

$$\begin{pmatrix} 0.23 \\ -0.32 \\ \dots \\ 0.66 \end{pmatrix} \begin{pmatrix} 0.52 \\ 0.12 \\ \dots \\ -0.07 \end{pmatrix} \quad \begin{pmatrix} -0.42 \\ 0.33 \\ \dots \\ -0.17 \end{pmatrix} \begin{pmatrix} -0.42 \\ -0.58 \\ \dots \\ -0.19 \end{pmatrix} \begin{pmatrix} -0.32 \\ 0.53 \\ \dots \\ 0.09 \end{pmatrix}$$

Word representations

$$\begin{pmatrix} 0.23 & 0.52 & -0.42 & -0.42 & -0.32 \\ -0.32 & 0.12 & 0.33 & -0.58 & 0.53 \\ \dots & \dots & \dots & \dots & \dots \\ 0.66 & -0.07 & -0.17 & -0.19 & 0.09 \end{pmatrix}$$

Matrix representation for the text

Natural Language Processing

