

# Clase 4 - Redes Neuronales y Fine-Tuning

Facundo González

18 de abril de 2024

# Outline

---

- ① Un poco más de redes neuronales
  - Feature Extraction
  - Redes Neuronales Convolucionales
  - Redes Neuronales Recurrentes
  - Autoencoders
  - Redes Generativas y Transformers
- ② Fine-Tuning
  - Fine-Tuning
- ③ Ejercicio Práctico
  - Definición del problema
  - Ejercicio

## ① Un poco más de redes neuronales

Feature Extraction

Redes Neuronales Convolucionales

Redes Neuronales Recurrentes

Autoencoders

Redes Generativas y Transformers

## ② Fine-Tuning

## ③ Ejercicio Práctico

# Algunas aplicaciones

---

Vimos un modelo de redes neuronales genéricas que sirve para resolver problemas variados. Pero hoy vamos a repasar algunas modificaciones o variaciones especializadas para algún problema en particular.

- Redes Neuronales Convolucionales: para computer vision.

# Algunas aplicaciones

---

Vimos un modelo de redes neuronales genéricas que sirve para resolver problemas variados. Pero hoy vamos a repasar algunas modificaciones o variaciones especializadas para algún problema en particular.

- Redes Neuronales Convolucionales: para computer vision.
- Redes Neuronales Recurrentes: para natural language processing (NLP).

# Algunas aplicaciones

---

Vimos un modelo de redes neuronales genéricas que sirve para resolver problemas variados. Pero hoy vamos a repasar algunas modificaciones o variaciones especializadas para algún problema en particular.

- Redes Neuronales Convolucionales: para computer vision.
- Redes Neuronales Recurrentes: para natural language processing (NLP).
- Autoencoders: reducción de dimensionalidad.

## Algunas aplicaciones

---

Vimos un modelo de redes neuronales genéricas que sirve para resolver problemas variados. Pero hoy vamos a repasar algunas modificaciones o variaciones especializadas para algún problema en particular.

- Redes Neuronales Convolucionales: para computer vision.
- Redes Neuronales Recurrentes: para natural language processing (NLP).
- Autoencoders: reducción de dimensionalidad.
- Redes Generativas y Transformers.

# Feature extraction

---

Para el análisis de imágenes con redes neuronales, es útil pensar en filtros que se le apliquen a la imagen previo al ingreso a la red.


Sería muy útil poder **extraer características** de la imagen, para pasarlas como input a la red neuronal.



# Feature extraction

Para el análisis de imágenes con redes neuronales, es útil pensar en filtros que se le apliquen a la imagen previo al ingreso a la red.

Sería muy útil poder **extraer características** de la imagen, para pasarlas como input a la red neuronal.



23	11	12	200	34	35	45	46	50	25
1	89	1	33	78	60	55	1	76	99
89	0	56	45	90	91	88	3	80	87
67	56	77	90	23	1	99	34	90	68
44	99	80	65	40	100	8	7	8	49
55	32	23	55	71	0	19	200	33	45
45	34	21	44	207	65	18	33	12	77
88	45	22	33	45	66	78	89	0	77
9	10	13	57	89	88	90	200	208	100
11	23	12	7	209	56	78	45	88	78

Tiene 4 patas

Tiene cola

Tiene dos orejas largas

Tiene ocico

Es bajito

Perro salchicha!

Features

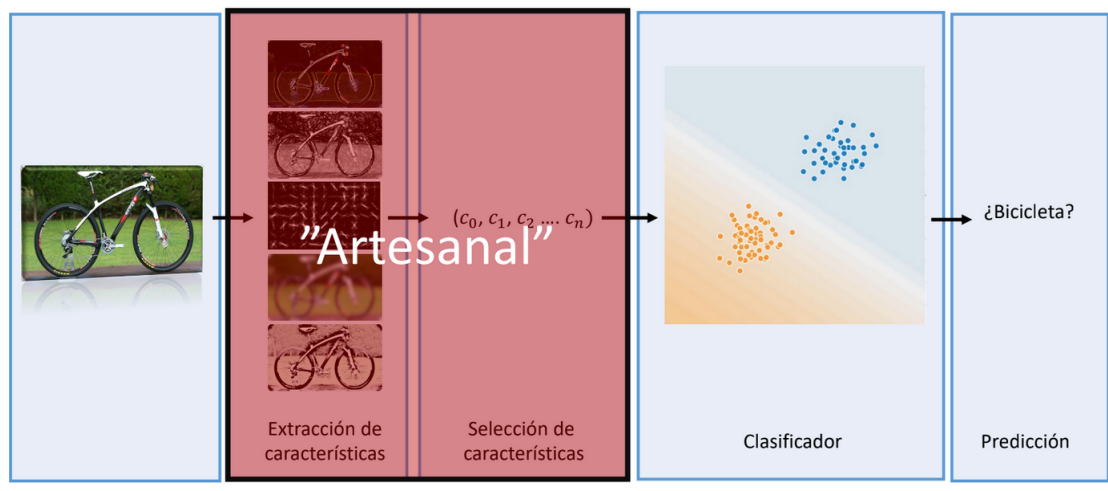
# Feature extraction

---

Conseguir estas características es una tarea manual, debemos definirlas a mano y poder extraerlas de las imágenes. Esta tarea es conocida como **feature extraction**, y es utilizada principalmente en imágenes.

Vamos a ver si podemos generalizar esto en un modelo que lo haga por nosotros.

# Feature extraction



# Redes Neuronales Convolucionales

---

Para resolver este problema y poder automatizar la tarea de extraer características, se crearon las redes neuronales convolucionales.

Son redes neuronales como las que vimos, pero con una capa, llamada **capa convolucional** que le va a aplicar filtros a las imagenes con el objetivo de que el input de la red tenga las características deseadas.

- **Capas convolucionales:** las primeras capas van a ser las que apliquen estos filtros, y las vamos a llamar capas convolucionales. Tienen como entrada la imagen entera (o alguna reduccion) y aplican una **convolución** sobre la imagen. Esto puede tener diferentes resultados, como detectar bordes en la imagen. Estas operaciones se aplican con *slicing windows* o *kernels*

# Redes Neuronales Convolucionales

---

Para resolver este problema y poder automatizar la tarea de extraer características, se crearon las redes neuronales convolucionales.

Son redes neuronales como las que vimos, pero con una capa, llamada **capa convolucional** que le va a aplicar filtros a las imagenes con el objetivo de que el input de la red tenga las características deseadas.

- **Capas convolucionales:** las primeras capas van a ser las que apliquen estos filtros, y las vamos a llamar capas convolucionales. Tienen como entrada la imagen entera (o alguna reduccion) y aplican una **convolución** sobre la imagen. Esto puede tener diferentes resultados, como detectar bordes en la imagen. Estas operaciones se aplican con *slicing windows* o *kernels*
- **Pooling layers:** son capas intermedias que tienen el objetivo de reducir la dimensión de la imagen tras aplicar una convolución.

# Redes Neuronales Convolucionales

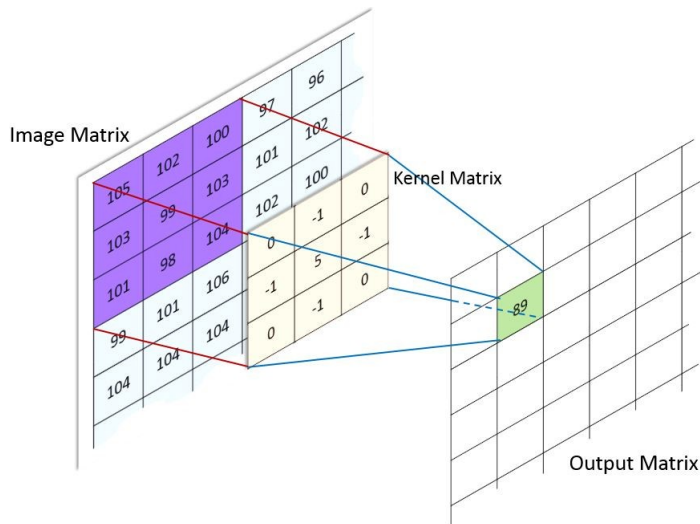


Figura: Aplicación de una convolución.

# Redes Neuronales Convolucionales

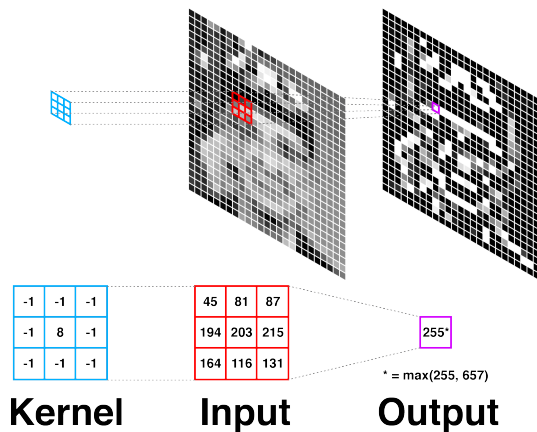


Figura: Aplicación de una convolución.

# Redes Neuronales Convolucionales

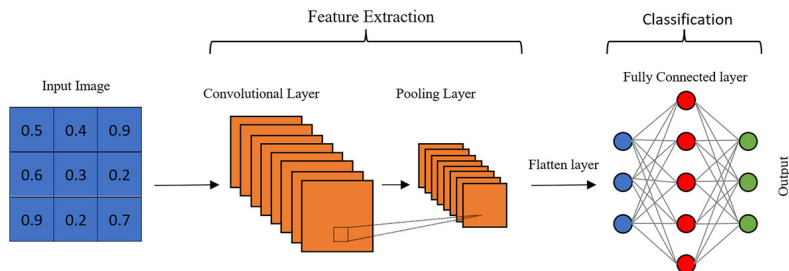


Figura: Red neuronal convolucional.



# Redes Neuronales Convolucionales

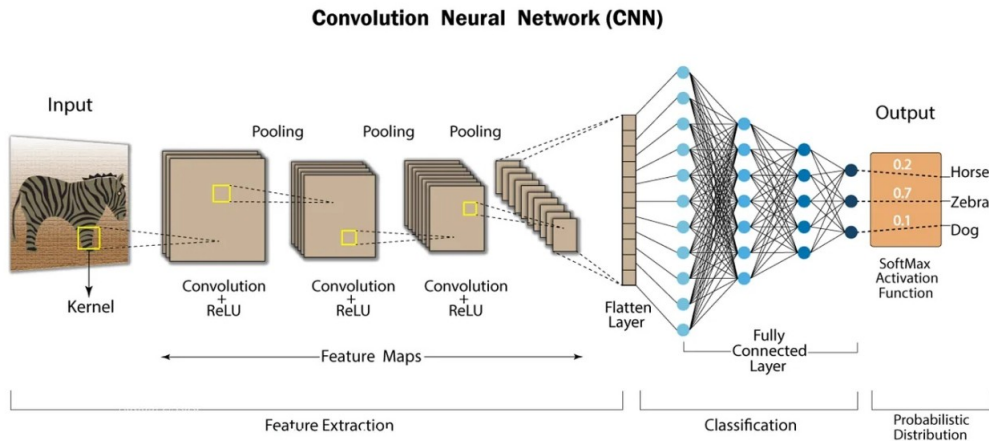


Figura: Red neuronal convolucional.

# Redes Neuronales Recurrentes

---

Las redes neuronales recurrentes es una arquitectura en donde las capas pueden retroalimentar neuronas de la capa anterior. El objetivo es poder tratar datos secuenciales (por ejemplo una secuencia de palabras) eficientemente. De alguna manera, podemos pensar que tienen “memoria”.

Uno de los problemas es el entrenamiento, ya que puede ser muy costoso en cómputo y memoria. Una estrategia es “desenrollar” la red y pensarla como una red neuronal común.

Es muy útil y utilizada para datos secuenciales, como por ejemplo texto o audio. La idea es que la secuencia de palabras anterior afecta a los pesos y el comportamiento al ingresar la siguiente palabra.

# Redes Neuronales Recurrentes

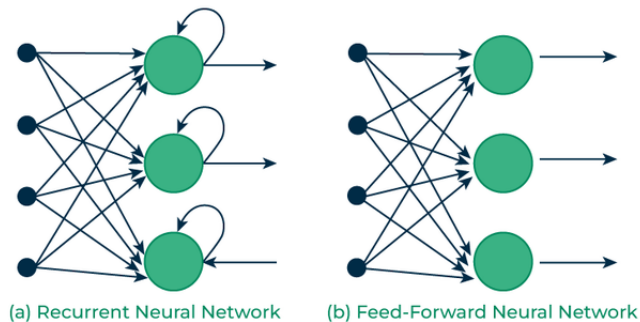


Figura: RNN vs FNN

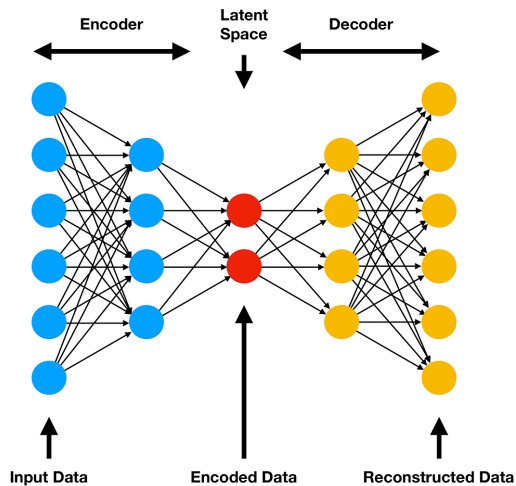
# Autoencoders

---

Otra de las aplicaciones útiles de redes neuronales, es reducir la dimensionalidad de algún dato. Para eso podemos implementar **autoencoders**, capas que reducen dimensionalidad y permiten codificar y decodificar.

Vamos a pensarlas como redes neuronales comunes, en donde queremos que aprendan a codificar un dato.

# Autoencoders



# Redes Generativas y Transformers

---

Otro tipo de red neuronal, muy de moda actualmente, son las redes neuronales generativas. Son modelos capaces de aprender de los datos de entrenamiento como para poder generar nuevos datos. También se las conoce como **Generative Adversial Network (GAN)**.

Lo que intentan hacer estas redes es aprender el comportamiento estadísticos de los datos para poder generar nuevos datos que no existen. Este tipo de Deep Learning entra dentro de la categoría de **aprendizaje no supervisado**, sin embargo existen enfoques dentro del aprendizaje supervisado.

- **Discriminator:** se utiliza otra red neuronal para evaluar qué tan realista es el dato generado por el modelo. El modelo es entrenado para mejorar su performance con respecto a este discriminador.

# Redes Generativas y Transformers

---

Otro tipo de red neuronal, muy de moda actualmente, son las redes neuronales generativas. Son modelos capaces de aprender de los datos de entrenamiento como para poder generar nuevos datos. También se las conoce como **Generative Adversial Network (GAN)**.

Lo que intentan hacer estas redes es aprender el comportamiento estadísticos de los datos para poder generar nuevos datos que no existen. Este tipo de Deep Learning entra dentro de la categoría de **aprendizaje no supervisado**, sin embargo existen enfoques dentro del aprendizaje supervisado.

- **Discriminator:** se utiliza otra red neuronal para evaluar qué tan realista es el dato generado por el modelo. El modelo es entrenado para mejorar su performance con respecto a este discriminador.
- **GPT:** General Pre-Trained

# Redes Generativas y Transformers

---

Otro tipo de red neuronal, muy de moda actualmente, son las redes neuronales generativas. Son modelos capaces de aprender de los datos de entrenamiento como para poder generar nuevos datos. También se las conoce como **Generative Adversial Network (GAN)**.

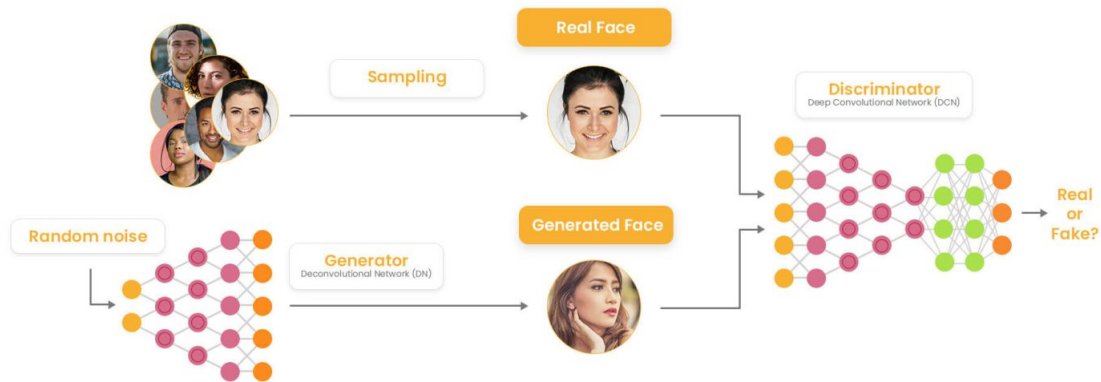
Lo que intentan hacer estas redes es aprender el comportamiento estadísticos de los datos para poder generar nuevos datos que no existen. Este tipo de Deep Learning entra dentro de la categoría de **aprendizaje no supervisado**, sin embargo existen enfoques dentro del aprendizaje supervisado.

- **Discriminator:** se utiliza otra red neuronal para evaluar qué tan realista es el dato generado por el modelo. El modelo es entrenado para mejorar su performance con respecto a este discriminador.
- **GPT:** General Pre-Trained

[ThisPersonDoesNotExist](#)



# Redes Generativas



# Transformer

---

Una arquitectura de redes neuronales que revolucionó las GAN's fueron los **transformers**, introducidos en 2017. [“Attention Is All You Need”](#) es el paper que introdujo este concepto. Los autores primero pensaron su aplicación en redes para traducir texto. Hoy en día son la arquitectura de muchos modelos generativos muy conocidos: **GPT: Generative Pre-trained Transformer**.

Primero se debe convertir el texto en representaciones llamadas **tokens**, que luego se transforman en vectores en alguna dimensión. Esto se llama hacer **embeddings**. Luego, cada token es contextualizado mediante una **capa de atención**. Lo interesante de esta implementación es que no necesita ser una RNN.

Para otro tipo de contextos, podemos generalizarlo de manera que el transformer aprende representaciones útiles de secuencias o conjuntos de datos. Podría ser audio o imágenes. Una de las grandes ventajas es que, a diferencia de las RNN, permite el entrenamiento en paralelo.

# Transformer

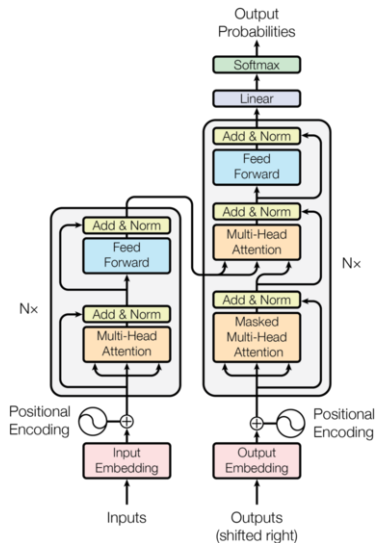


Figura: Arquitectura Transformer

# Self-Attention

Podemos pensar la atención como una matriz que relaciona las palabras de un input. Si tomamos la frase “El perro estaba en el salón, durmiendo tranquilo”, nos gustaría que tranquilo y perro tengan una relación fuerte.



Figura: Matriz de atención

- 1 Un poco más de redes neuronales
- 2 Fine-Tuning  
Fine-Tuning
- 3 Ejercicio Práctico

# Fine-Tuning

---

El **fine-tuning** es una estrategia que permite utilizar modelos pre-entrenados y realizar un entrenamiento particular sobre esos pesos.

Nos va a permitir utilizar modelos grandes ya pre-entrenados y hacer un entrenamiento para un caso particular. Para esta tarea vamos a necesitar datos etiquetados como si fuera un entrenamiento normal.

① Un poco más de redes neuronales

② Fine-Tuning

③ Ejercicio Práctico

Definición del problema

Ejercicio

# Definición del problema

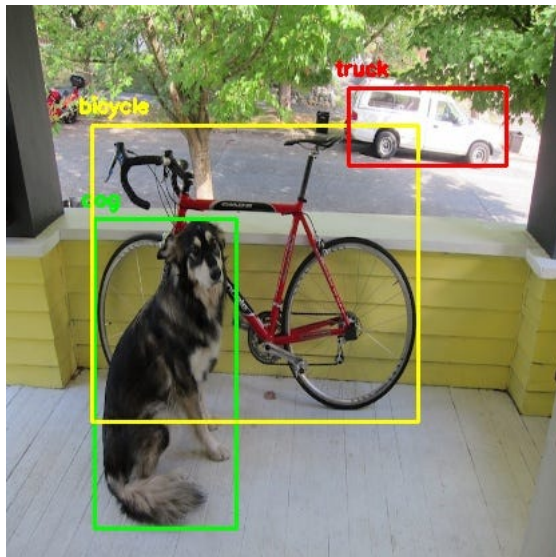
---

Vamos a querer un modelo que detecte comida en una imagen, y en particular vamos a querer que funcione en tiempo real.

YOLO: es un modelo de detección de imágenes en tiempo real. Dada una imagen (o video en tiempo real) detecta varias categorías junto con sus **bounding box**. ([Repositorio](#))



# YOLO



# YOLO

---

Además, el modelo permite hacer fine-tuning. Podemos entrenarlo con un conjunto de imágenes etiquetadas junto a sus bounding-boxes.

Para nuestro caso, necesitamos muchas imágenes de comida, etiquetada y con bounding box. El problema es cómo conseguirlas.

[GroundingDINO](#): es un modelo de detección de imágenes que permite buscar una etiqueta en lenguaje natural. A diferencia de YOLO, no funciona en tiempo real, pero nos va a permitir crear nuestro dataset etiquetado.

# Ejercicio

---

Nuestro objetivo entonces es:

- 1 Descargar un dataset de imagenes de comidas.
- 2 Etiquetarlo y generar un dataset con GroundingDINO.
- 3 Realizar el fine-tuning sobre el modelo YOLO con nuestro dataset.
- 4 Utilizarlo y ver si logramos nuestro objetivo.