

# Técnicas de inteligencia artificial

Adriana Cervantes Castillo



## SEMANA 09: DEEP LEARNING

# Contenido

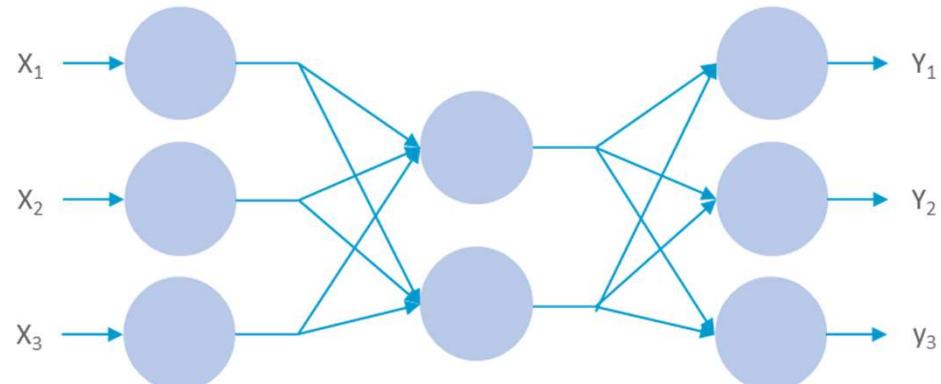
- Redes prealimentadas
- Redes neuronales recurrentes
- Redes autoencoders
- Redes neuronales convolucionales
- Redes generativas antagónicas

El funcionamiento de la red neuronal vendrá determinado por los siguientes parámetros:

- **Arquitectura de la red**, esto es, el número de capas, número de neuronas por capa y las conexiones entre neuronas entre diferentes capas.
- **Función de activación**: función signo, función escalón, etc.
- **Algoritmo de aprendizaje** determinado principalmente por la regla de aprendizaje para ajustar pesos

# Redes prealimentadas profundas

- Todas las neuronas de una capa están conectadas con todas las neuronas de la capa siguiente.
- No existe conexiones entre neuronas de la misma capa
- No existe retroalimentación entre capas.
- Las funciones de activación más utilizadas son las sigmoidales.
- Utiliza el *método del gradiente de error* para ajustar los pesos durante el entrenamiento:

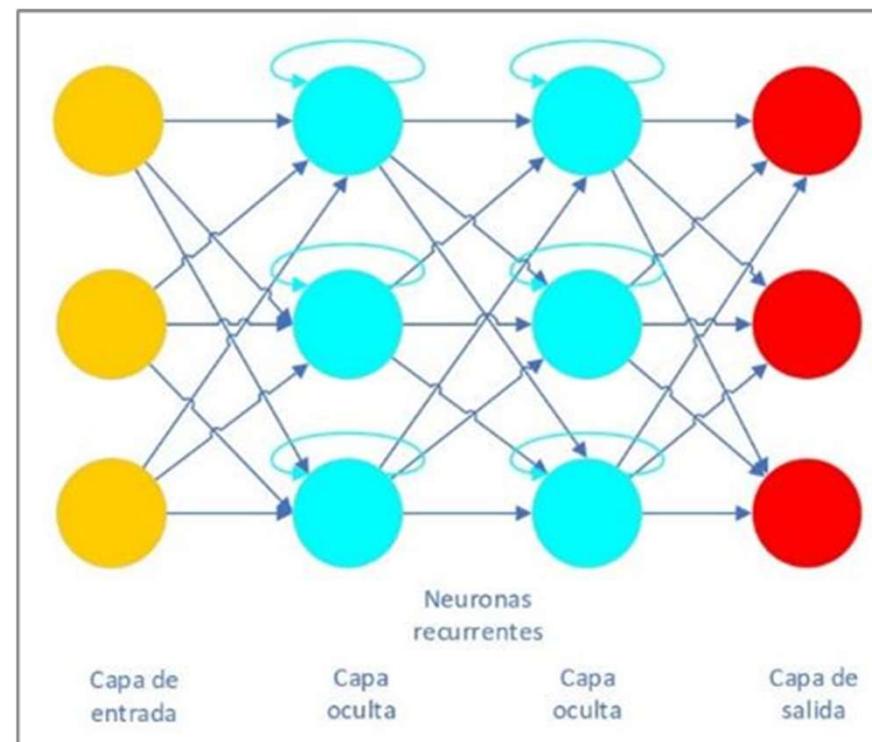


# Redes neuronales recurrentes

- Tienen neuronas específicas que incluyen una memoria en la que almacenan las entradas correspondientes a ciclos anteriores. De esta manera, son capaces de producir una salida dependiente de la entrada actual y de las entradas anteriores.

Memorias de Largo y Corto Plazo (*LSTM – Long and Short-Term Memories*)

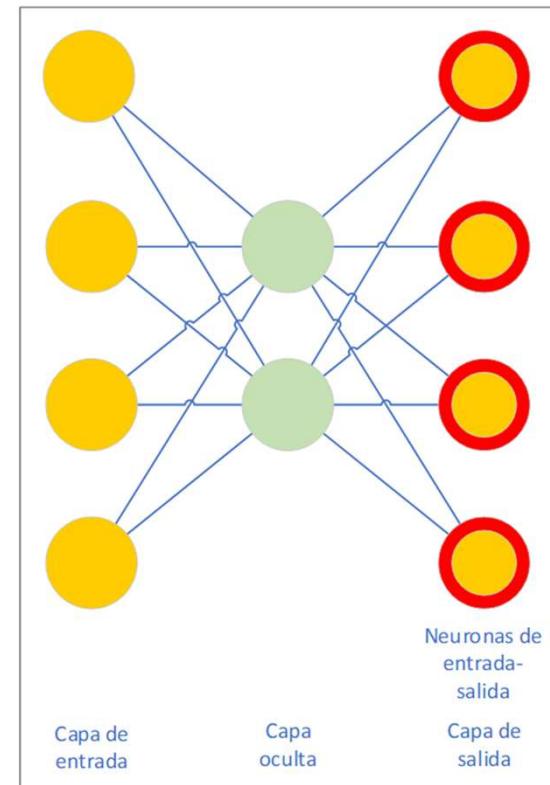
Procesan datos secuenciales, como series temporales, texto, audio, o cualquier otro tipo de datos donde el orden de las entradas sea importante.



# Autoencoders

- Los autocodificadores o *autoencoders* (AE) son redes neuronales simétricas en forma de reloj de arena en las que las capas ocultas son más pequeñas que las capas de entrada y de salida

De la entrada al código, el autocodificador actúa como un codificador (comprimiendo la información), y del código a la salida el autocodificador actúa como un decodificador.



# Redes generativas antagónicas (GANs)

Modelo que consta de dos redes neuronales que trabajan de manera competitiva:

## 1. Generador (Generator):

1. Su función es **crear datos falsos** que sean similares a los datos reales.
2. Toma un vector aleatorio como entrada (ruido) y lo transforma en un ejemplo generado, como una imagen.

## 2. Discriminador (Discriminator):

1. Su tarea es distinguir entre los datos reales (provenientes del conjunto de datos) y los generados (provenientes del generador).
2. Actúa como un "juez" que evalúa la autenticidad de los datos.

(OpenAI, 2025)

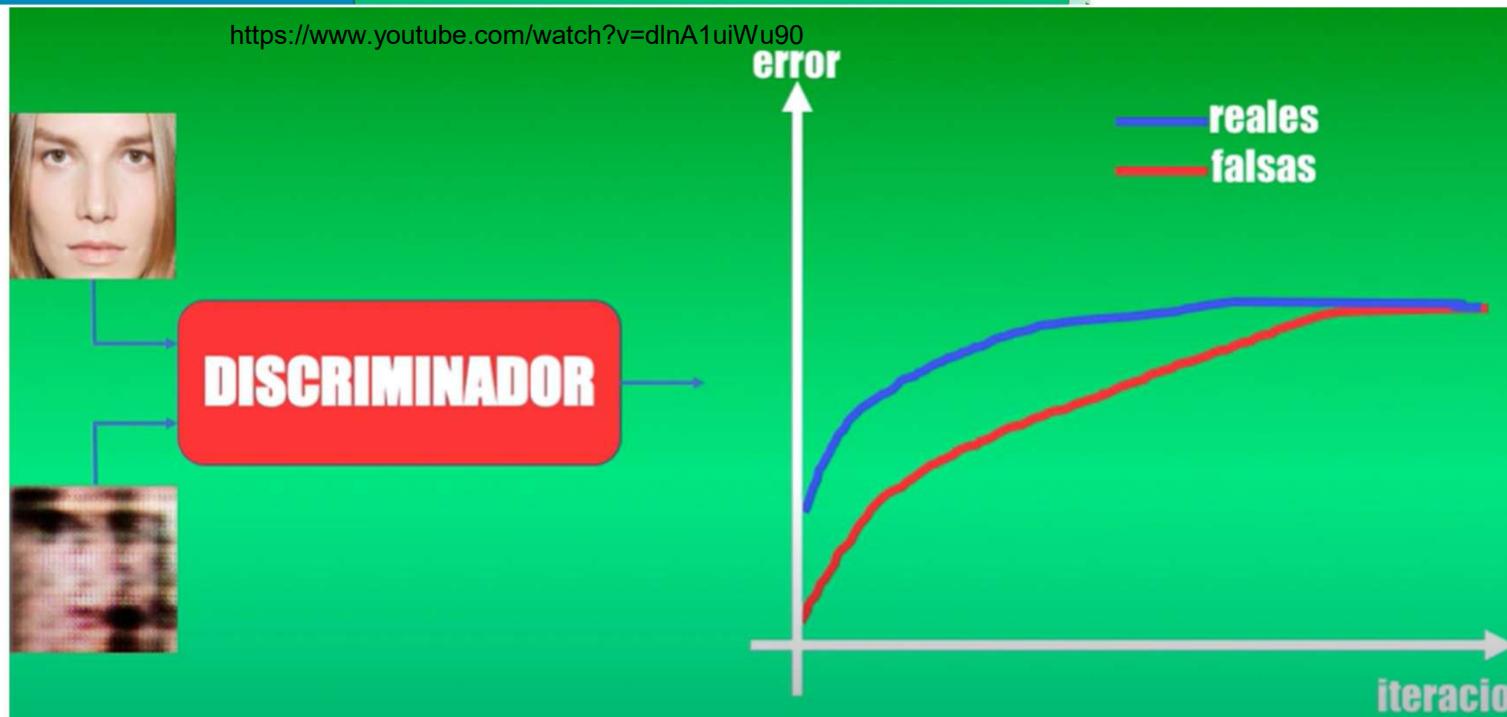


**generador**



**discriminador**

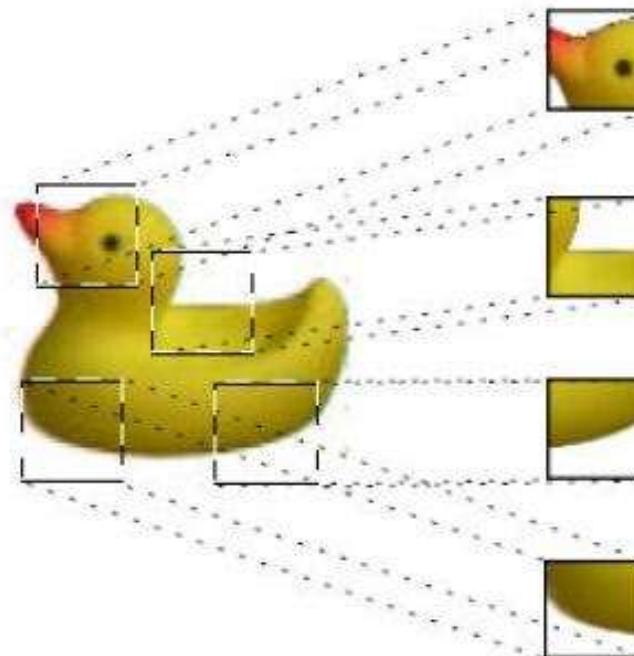
# GAN



# Redes neuronales convolucionales(CNNs)

- Inspiradas en la forma en que el cerebro humano procesa las imágenes visuales. Detectan patrones mediante **operaciones de convolución**.

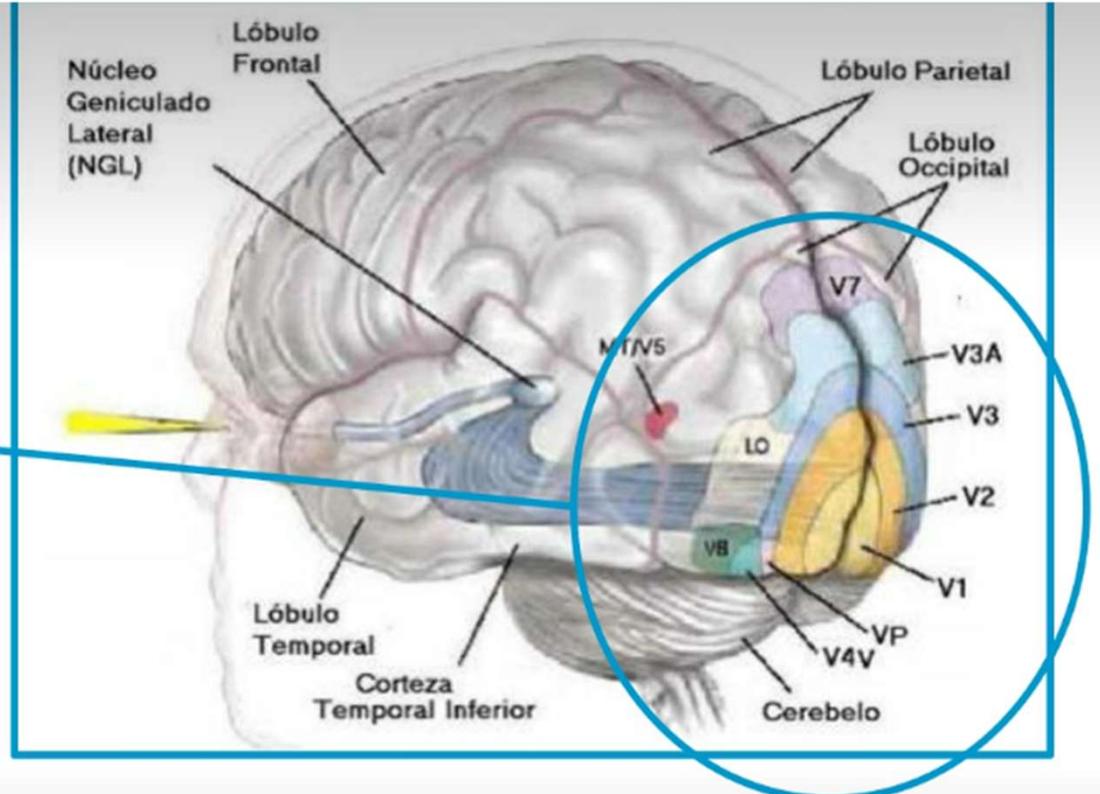
Imágenes → características

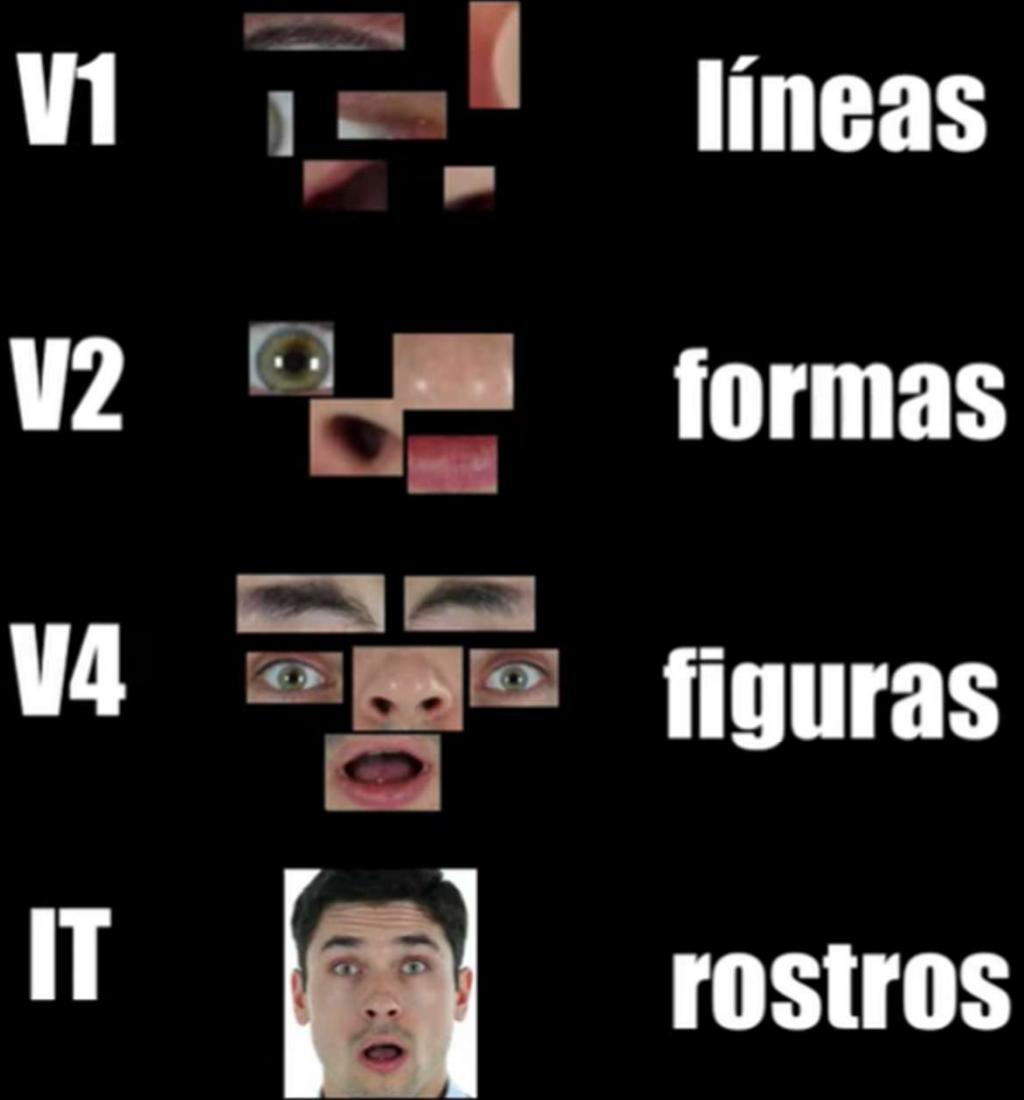


<https://recuperaciondeinformacionmultimodal.blogspot.com/2011/05/extraccion-de-caracteristicas.html>

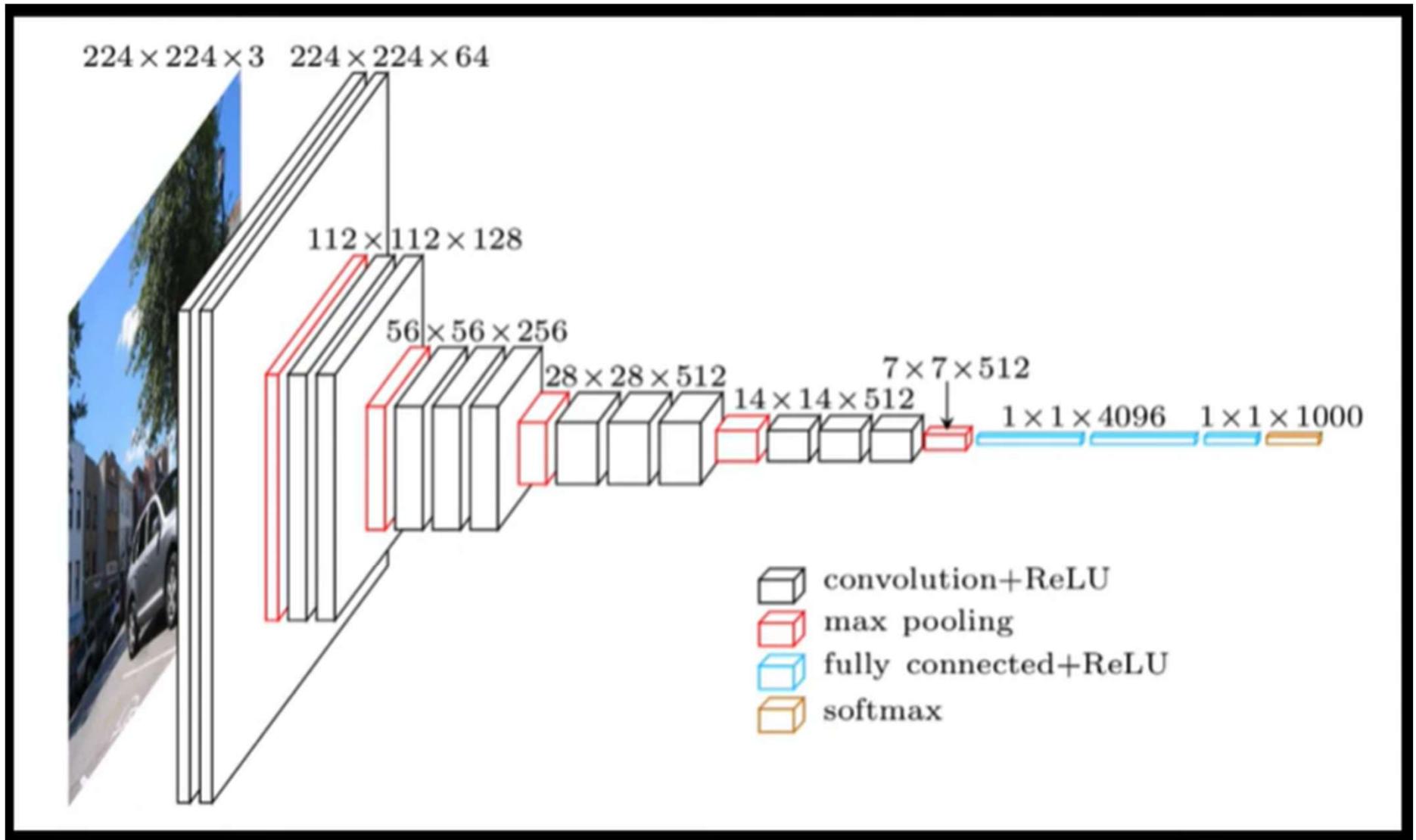
PROCESAMIENTO  
ELABORADO  
(CONSCIENTE)

Corteza  
visual





<https://www.youtube.com/watch?v=HyZFfBU0ADg>



<https://www.youtube.com/watch?v=HyZFfBU0ADg>

# Arquitectura

**Entrada:** píxeles de la imagen que va a ser analizada por la red.

**Capas convolucionales:** aplicación de filtros para capturar características representativas.

**Capas de muestreo:** o *pooling layers* que tratarán de reducir la imagen conservando lo esencial de su estructura espacial.

**Capas totalmente conectadas:** RNA regular conectada a la salida.

**Salida:** combinada y orientada a clasificación (*softmax* o *sigmoid*).

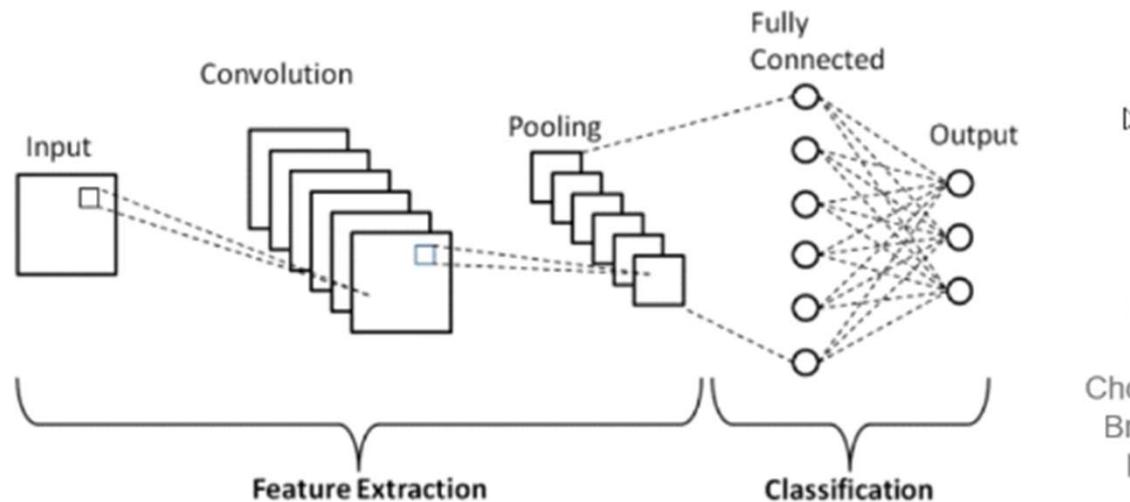
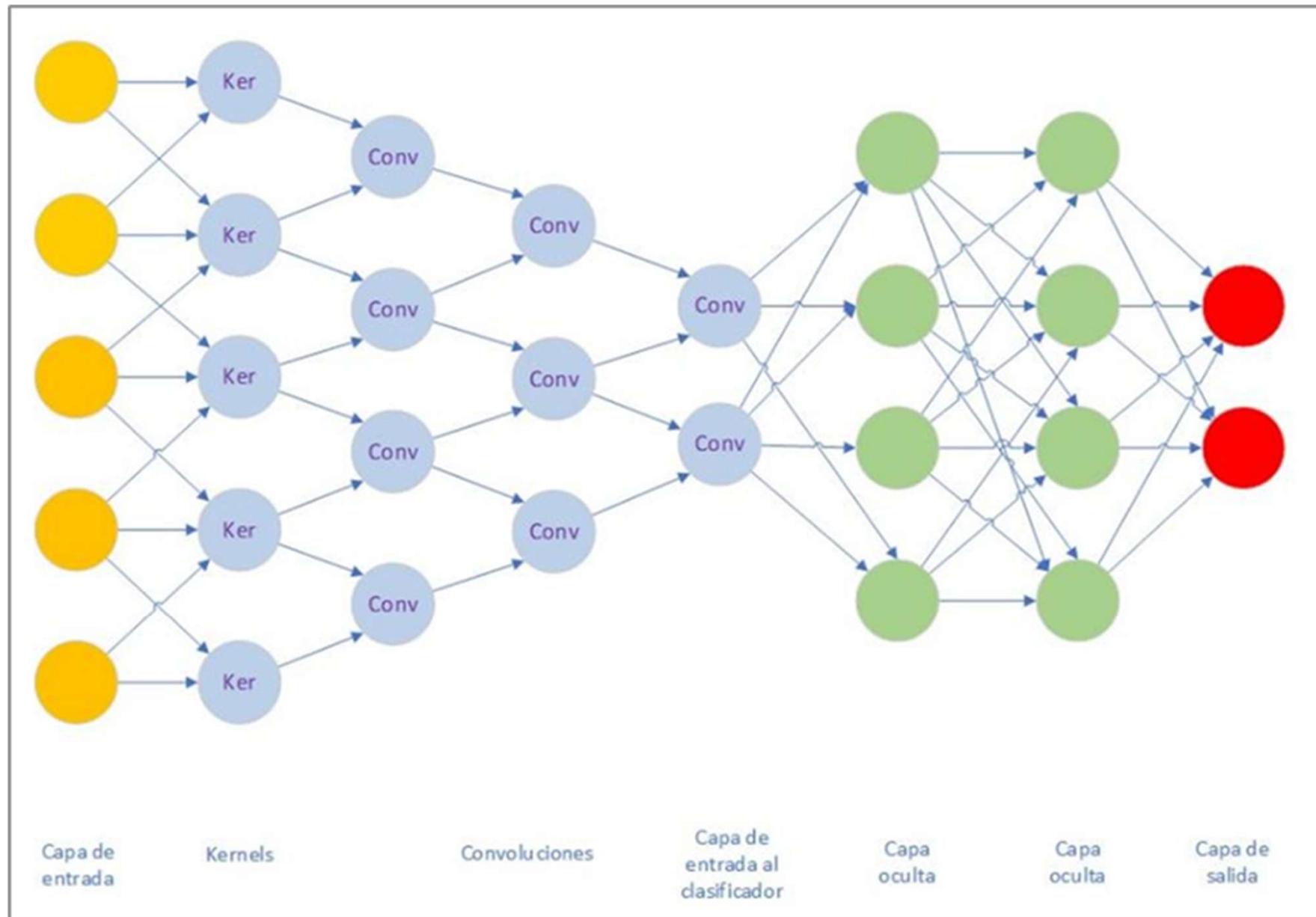
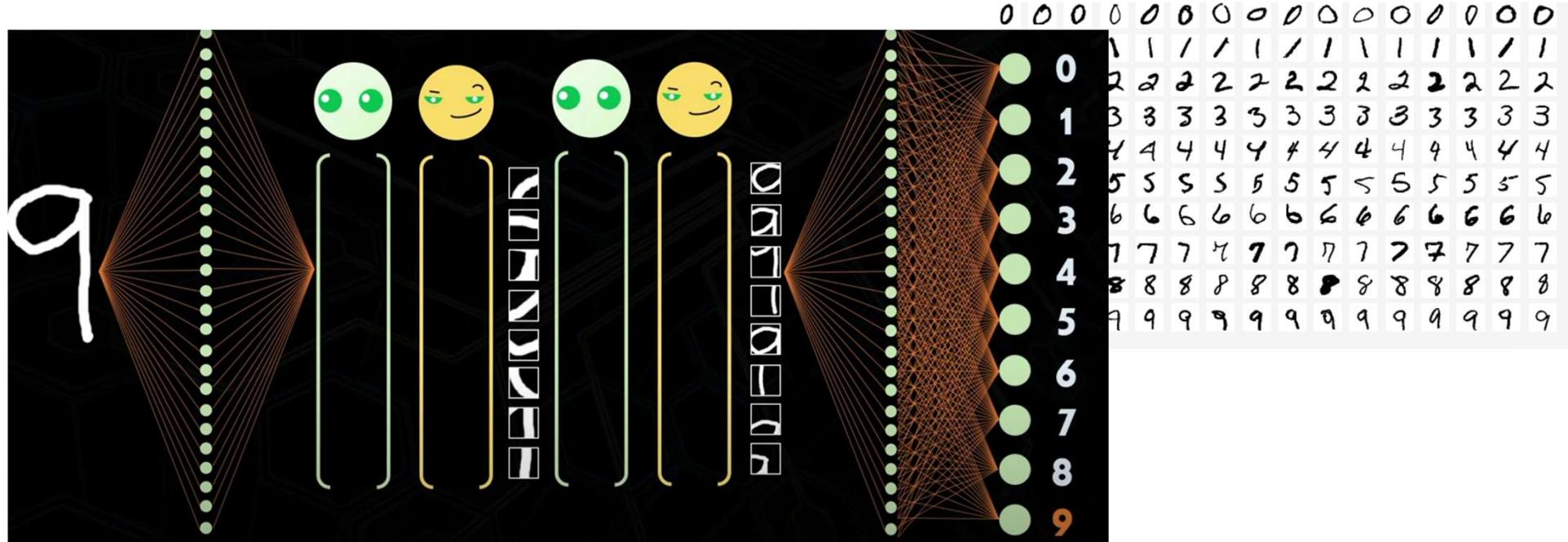


Figura 2. Arquitectura. Fuente:  
Dabholkar, S., Deshmukh, P.,  
Chormare, R. y Jawale, S. K. (2022).  
Breast Cancer Classification using  
Deep learning. [Research Gate](#).

# Red Convolucional



# Ejemplo de una red convolucional



```
modelo = tf.keras.Sequential([
    tf.keras.layers.Conv2D(32, (3,3), input_shape=(28,28,1), activation='relu'),
    tf.keras.layers.MaxPooling2D(2,2), #2,2 es el tamano de la matriz

    tf.keras.layers.Conv2D(64, (3,3), activation='relu'),
    tf.keras.layers.MaxPooling2D(2,2), #2,2 es el tamano de la matriz

    tf.keras.layers.Flatten(),
    tf.keras.layers.Dense(units=100, activation='relu'),
    tf.keras.layers.Dense(10, activation='softmax')
])
```

- <https://www.tensorflow.org/datasets/overview?hl=es-419>
- [https://www.youtube.com/watch?v=QLy8v6LL\\_4A](https://www.youtube.com/watch?v=QLy8v6LL_4A)
- <https://www.youtube.com/watch?v=SGoxsBgp3WM>
- <https://www.youtube.com/watch?v=4sWhhQwHqug&t=1128s>



Gracias por su atención

[www.unir.net](http://www.unir.net)