DEEP LEARNING

Infografía Arquitecturas de Apredizaje Profundo

1. Redes Densas (MLP):

arquitecturas donde cada neurona de una capa está conectada con todas las neuronas de la capa siguiente. Son la base de los Multi-Layer Perceptron (MLP).

Modelo Matematico:

- **Capa Densa Unica:**

$$y = \phi(Wx + b)$$

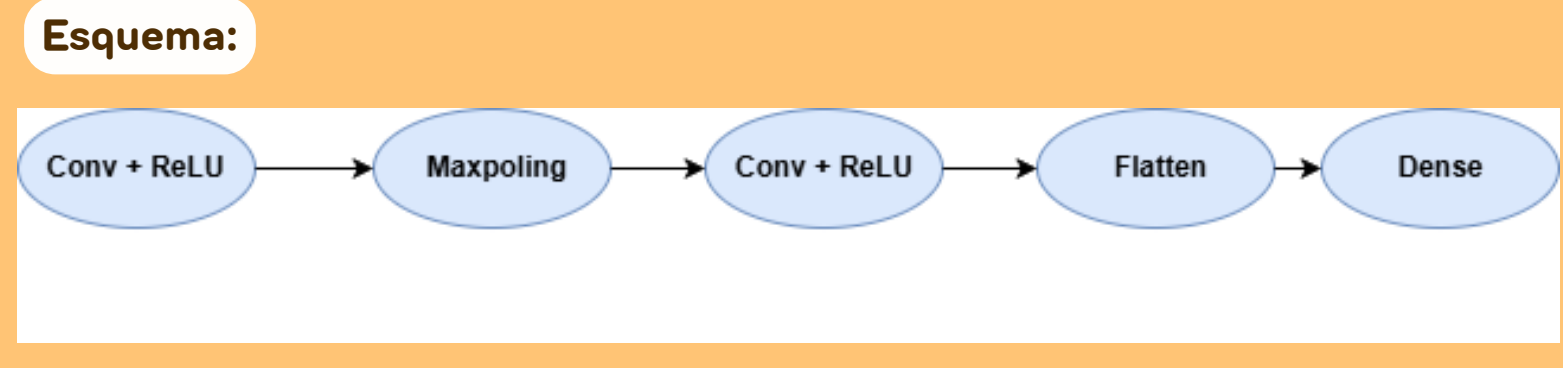
- **Capas Apiladas:**

$$a^{(l)} = \phi^{(l)}\left(W^{(l)}a^{(l-1)} + b^{(l)}\right)$$

¿Cómo funciona ?

- **Computar sucesivamente a^l hasta obtener y^l**
- **Calcular pérdida $L(y', y)$**
- **(backpropagation) $\nabla_W L$ y $\nabla_b L$**
- **Actualizar parámetros con un optimizador (SGD, Adam...).**
- **Iterar en minibatches hasta convergencia.**

Esquema:



Aplicaciones Representativas:

- **MLP básico para caracterizar imágenes pequeñas**
- **Clasificación binaria / multiclase en conjuntos tabulares.**
- **Regresión para aproximación de funciones y predicción de valores continuos (ej.: MLP con MSE)**
- **Como bloque final en arquitecturas más complejas CNN/RNN se usa capa totalmente conectada antes de la salida.**

2. Redes Convolucionales CNN:

Redes Convolucionales (CNN): arquitecturas diseñadas para procesar datos con estructura espacial (imágenes, señales), detectando patrones locales mediante filtros.

Modelo Matematico:

$$Y(i,j) = \sum_{m}^{\square}\sum_{n}^{\square} X(i+m, j+n)K(m,n)$$

- **Cada filtro aprende un patrón (bordes, texturas...).**
- **El resultado es un mapa de características.**

Pooling:

$$A = \phi(Y)$$

$$P(i,j) = \underset{(m,n)E2x2}{max} A(2i+m, 2j+n)$$

Reduce dimensionalidad y extrae características robustas.

¿Como Funciona?

- **Localización de patrones: cada filtro responde a bordes, texturas, colores.**
- **Profundización jerárquica: capas superiores detectan patrones más complejos.**
- **Pooling: hace la representación más robusta y comprimida.**
- **Head densa: toma las características y clasifica.**
- **Entrenamiento completo: se aplican backpropagation y optimizadores como Adam.**

Esquema:

Aplicaciones Representativas:

- Clasificación de imágenes
- Detección de objetos (YOLO, SSD).
- Segmentación semántica (U-Net).
- Reconocimiento facial.
- Análisis médico (radiografías, resonancias).
- Reconocimiento de gestos y video (3D CNNs).

3. Redes Recurrente (RNN/LSTM)

Las RNN son arquitecturas diseñadas para procesar datos secuenciales (texto, series de tiempo, audio). A diferencia de una red densa, una RNN mantiene un estado oculto que se actualiza en cada paso temporal.

Modelo Matematico:

- **Para una entrada secuencial x(t) :**

$$h = \phi\left(W_{xh}x_t + W_{hh}h_{t-1} + b_h\right)$$

$$y_t = W_{ht}h_t + b_y$$

LSTM

Una LSTM introduce un sistema de compuertas para controlar el flujo de información.

- **olvida la puerta**

$$f_t = \phi\left(W_f\left(h_{t-1}, x_t\right) + b_f\right)$$

- **Puerta de Entrada**

$$i_t = \phi\left(W_i\left(h_{t-1}, x_t\right) + b_i\right)$$

- **Estado candidato**

$$C'_t = tanh\left(W_C\left(h_{t-1}, x_t\right) + b_C\right)$$

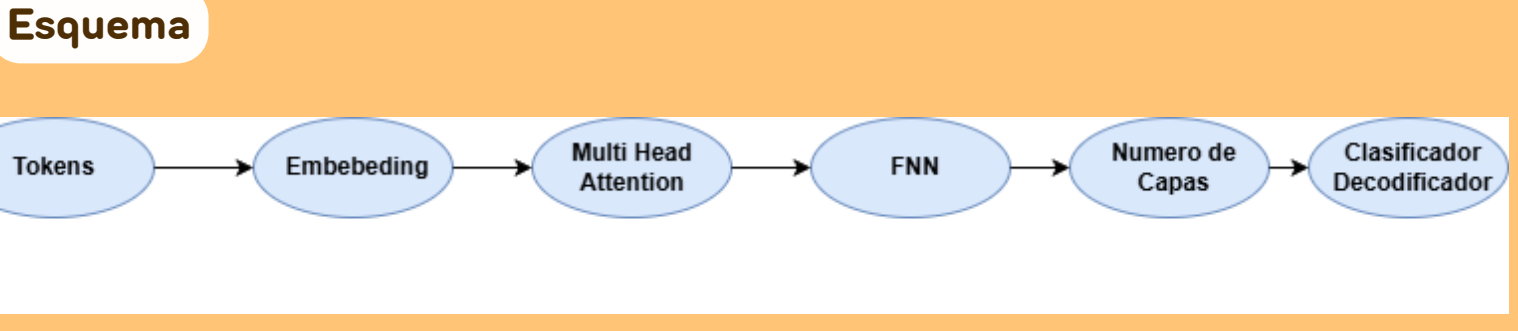
- **Actualización del estado de la celda**

$$C_t = f_t\left(C_{t-1}\right) + i_t\left(C'_t\right)$$

- **Puerta de salida**

$$o_t = \phi\left(W_o\left(h_{t-1}, x_t\right) + b_o\right)$$

Esquema:



Aplicaciones Representativas

- Modelos de texto
- Predicción de palabras
- Análisis de sentimiento
- Predicciones financieras
- Clima
- Señales biomédicas
- Reconocimiento de voz
- Modelado de secuencias acústicas

4. Transformers

diseñada para procesar secuencias sin recurrencia ni convoluciones. Su principio clave es el mecanismo de atención, que permite al modelo enfocarse dinámicamente en distintas partes de la secuencia.

Modelo Matematico:

$$Q = XW_Q \quad K = XW_K \quad V = XW_V$$

$$Attention\left(Q, V, K\right) = softmax\left(\frac{QK^T}{\sqrt[2]{d_k}}\right)(V)$$

Cada token genera tres vectores:

- **Q (query)**
- **K (key)**
- **V (value)**

Esto permite que cada palabra "mire" a todas las demás para obtener contexto

¿Como funciona?

- **Convertir tokens en embeddings.**
- **Añadir codificación posicional.**
- **Para cada capa:**
- **Aplicar self-attention → mezcla contextual de tokens.**
- **Pasar por FFN → transformación no lineal.**
- **En el decoder: aplicar masked attention y cross-attention.**
- **Pasar la salida final a un softmax para generar texto o clasificación.**

Esquema

Multi Head Attention

- El modelo usa varias cabezas de atención para capturar diferentes tipos de relaciones:
- Sintácticas
- Semánticas
- Dependencias de largo plazo

Posicional Encoding

Porque los Transformers no tienen estructura secuencial interna, usan codificación posicional para indicar el orden.

Flujo del Encoder

- Entrada → embeddings + posición.
- Self-attention: cada token consulta a todos.
- FFN punto a punto.
- Residuals + LayerNorm

Aplicaciones Principales

- Traducción automática
- Chatbots / modelos conversacionales (GPT)
- Análisis de sentimientos
- Resumen automático
- Generación de texto
- Clasificación de texto
- Búsqueda semántica
- Modelos de visión (Vision Transformer)

FNN

Su propósito es transformar la representación generada por la atención, añadiendo capacidad no lineal y proyectando la información a un espacio diferente donde se puedan capturar patrones más complejos.

Es una mini-red neuronal de dos capas totalmente conectadas:

1. Proyección hacia un espacio más grande (aumenta la dimensionalidad):
   - $x \rightarrow xW_1 + b_1$
2. No linealidad:
   - Se aplica ReLU, GELU u otra activación.
3. Proyección de regreso al tamaño original:
   - $h \rightarrow hW_2 + b_2$