# EVOLUCIÓN DE LAS REDES NEURONALES

## **REDES NEURONALES RECURRENTES (RNN)**

Las **RNNs** surgieron para modelar **datos secuenciales**, es decir, aquellos en los que el orden importa. Algunos ejemplos son el lenguaje natural, series temporales y audio.

Las redes neuronales tradicionales (MLPs o CNNs) no podían capturar **dependencias a lo largo del tiempo**, ya que procesaban la información de manera independiente. Las **RNNs** introdujeron un mecanismo para mantener una **memoria** del pasado a través de **estados ocultos** que pasaban información de un paso de tiempo al siguiente.

Primeros usos clave de las RNNs:

1. **Reconocimiento de voz**: Modelaban patrones en ondas de sonido.
2. **Traducción automática**: Capturaban dependencias a largo plazo en textos.
3. **Predicción de series temporales**: Predecían valores futuros en datos financieros o meteorológicos.
4. **Modelado de lenguaje**: Generaban texto palabra por palabra basado en contexto previo.

### **LSTM /GRU**

A pesar de sus ventajas, las RNNs tenían problemas serios:

* **Exploding gradients y vanishing gradients**: Con largas secuencias, la información se degradaba y las redes dejaban de aprender correctamente.
* **Dificultad en capturar dependencias a largo plazo**: Costoso en cómputo y entrenamiento.

Para mejorar las RNNs, se crearon variantes como:

* **LSTMs (Long Short-Term Memory, 1997)**: Introdujeron una estructura de puertas para controlar mejor qué información recordar u olvidar.
* **GRUs (Gated Recurrent Units, 2014)**: Una variante simplificada de las LSTMs con desempeño similar.

Estas mejoras permitieron avances en:

* Traducción automática con **seq2seq** (2014, Sutskever et al.).
* Modelado de lenguaje con redes más profundas.
* Chatbots y asistentes virtuales más eficientes.

## **TRANSFORMERS (2017)**

A pesar de las mejoras, los LSTMs y GRUs seguían teniendo problemas:

1. **No paralelizables**: Cada paso depende del anterior, lo que hace que sean **lentas** en el entrenamiento.
2. **Difíciles de escalar**: Modelos como Google Neural Machine Translation (GNMT) usaban cientos de millones de parámetros y aún tenían problemas con largas dependencias.
3. **Costos computacionales elevados**: Especialmente con textos largos.

**Solución: "Attention is All You Need" (2017, Vaswani et al.)**

Google introdujo los **Transformers**, eliminando la necesidad de recurrencia y reemplazándola con **autoatención**.

Ventajas de los Transformers:

* **Paralelización total**: Permite entrenar grandes modelos mucho más rápido.
* **Captura de relaciones a largo plazo sin degradación**.
* **Mayor eficiencia y escalabilidad**: Puede manejar grandes corpus de datos.

Impacto inmediato en NLP:

* **BERT (2018)**: Mejora en comprensión de texto.
* **GPT (2018 en adelante)**: Creación de texto de manera fluida.
* **T5, Turing-NLG y otros modelos masivos**.

## **IA GENERATIVA Y LA EVOLUCIÓN DE TRANSFORMERS**

Con los **Transformers**, la IA generativa alcanzó un nuevo nivel.

### **IA GENERATIVA EN TEXTO:**

* **GPT-2 (2019):** Capaz de generar texto fluido y coherente.
* **GPT-3 (2020):** Mejora drástica en tamaño y calidad del texto generado.
* **ChatGPT / GPT-4 (2023+):** Conversaciones más naturales y capacidades multimodales.

### **IA GENERATIVA EN IMÁGENES:**

* **DALL·E (2021):** Creación de imágenes a partir de texto.
* **Stable Diffusion (2022):** IA generativa open-source para imágenes.

### **IA GENERATIVA EN AUDIO Y VIDEO:**

* **Jukebox (OpenAI, 2020):** IA generativa de música.
* **Synthesia / RunwayML:** Generación de video con IA.

### **TRANSFORMERS EN VISIÓN Y OTRAS ÁREAS:**

* **ViT (Vision Transformer, 2021):** Reemplaza CNNs en reconocimiento de imágenes.
* **Time Series Transformer:** Mejora en predicción de series temporales.

## **¿HACIA DÓNDE VA LA IA GENERATIVA?**

La evolución de los Transformers sigue transformando la IA

Tendencias futuras:

1. **Modelos Multimodales:** (Ej. GPT-4 Vision, Gemini) para trabajar con imágenes, texto y sonido juntos.
2. **Mejor eficiencia y menor consumo energético:** Transformers más eficientes como Mistral, Mixtral y modelos adaptados a dispositivos móviles.
3. **IA personalizada y adaptable:** Modelos más pequeños y ajustables para tareas específicas.
4. **Fusión con simbología y razonamiento lógico:** Para mejorar el pensamiento de IA.

# PROCESAMIENTO DEL LENGUAJE NATURAL

El **Procesamiento del Lenguaje Natural (PLN)** es una rama de la inteligencia artificial (IA) que permite a las máquinas comprender, interpretar y generar lenguaje humano. Su desarrollo ha revolucionado tecnologías como los asistentes virtuales, traductores automáticos y chatbots avanzados.

Desde los primeros métodos estadísticos hasta los modelos más sofisticados basados en **Transformers**, el PLN ha evolucionado significativamente, impulsado por avances en **tokenización, embeddings y autoatención**.

El **Procesamiento del Lenguaje Natural** ha pasado de técnicas simples de tokenización y embeddings estáticos a modelos avanzados de **Transformers** que pueden generar, comprender y traducir lenguaje con precisión casi humana.

El futuro del PLN está en la integración de modelos **multimodales** que combinen texto, imágenes, audio y video en un solo sistema.

Vamos a ver unos videos a modo de introducción y luego poder entender mejor todo:

* La Siguiente Gran Revolución: NLP (Procesamiento del Lenguaje Natural): <https://www.youtube.com/watch?v=cTQiN9dewIg> (15:11)
* **INTRO al Natural Language Processing (NLP) #1 - ¡De PALABRAS a VECTORES!:** <https://www.youtube.com/watch?v=Tg1MjMIVArc> (11:10)
* **INTRO al Natural Language Processing (NLP) #2 - ¿Qué es un EMBEDDING?:** <https://www.youtube.com/watch?v=RkYuH_K7Fx4> (15:00)
* **¿Qué es un TRANSFORMER? La Red Neuronal que lo cambió TODO!:** <https://www.youtube.com/watch?v=aL-EmKuB078> (18:41)
* **¿Por qué estas REDES NEURONALES son tan POTENTES? | TRANSFORMERS Parte 2**

<https://www.youtube.com/watch?v=xi94v_jl26U> (15:16)

* Lo que OpenAI NO quería que supieras sobre GPT4 - (De los MoEs a Mixtral): <https://www.youtube.com/watch?v=Sfnu5OmAITA> (17:58)

## **TOKENIZACIÓN: DIVIDIR EL LENGUAJE EN PARTES MANEJABLES**

El primer paso en PLN es convertir el texto en una forma comprensible para los modelos de IA. Esto se hace mediante la **tokenización**, que divide el texto en **unidades más pequeñas** llamadas **tokens**.

### **TIPOS DE TOKENIZACIÓN:**

1. **Tokenización basada en palabras**
   * "El gato negro" → ["El", "gato", "negro"]
   * Problema: No maneja bien palabras desconocidas (**OOV - Out of Vocabulary**).
2. **Tokenización basada en subpalabras** (Más eficiente y usada en Transformers)
   * Modelos como **BPE (Byte Pair Encoding), WordPiece y SentencePiece** dividen palabras en subunidades frecuentes.
   * Ejemplo: "increíblemente" → ["in", "cre", "íble", "mente"]
   * Ventaja: Maneja palabras desconocidas y reduce el vocabulario.
3. **Tokenización basada en caracteres** (Menos común en PLN, pero útil en algunos casos como idiomas con gran cantidad de caracteres, como el chino.)
   * "Hola" → ["H", "o", "l", "a"]
   * Problema: Pierde relaciones semánticas y requiere secuencias más largas.

En modelos de PLN modernos, los tokens no son necesariamente palabras completas, sino fragmentos optimizados para mejorar la eficiencia del aprendizaje automático.

## **EMBEDDINGS: REPRESENTACIÓN NUMÉRICA DEL LENGUAJE**

Una vez tokenizado el texto, las palabras o subpalabras deben convertirse en números para ser procesadas por un modelo. Aquí es donde entran los **word embeddings**, que representan palabras en un espacio matemático donde palabras con significados similares tienen vectores cercanos.

### **PRINCIPALES ENFOQUES DE EMBEDDINGS:**

1. **Word2Vec (2013, Google)**
   * Método basado en el contexto en el que aparece una palabra.
   * Problema: Una palabra tiene siempre la misma representación, sin considerar el contexto.
2. **GloVe (2014, Stanford)**
   * Basado en la coocurrencia de palabras en grandes corpus de texto.
   * Problema similar a Word2Vec: no es contextual.
3. **Embeddings contextuales (BERT, GPT, etc.)**
   * Las palabras se representan de manera diferente según el contexto.
   * Ejemplo:
     + "Banco" en "Me senté en un banco" → Vector A
     + "Banco" en "Fui al banco a retirar dinero" → Vector B
   * Modelos basados en **Transformers** usan esta técnica para mejorar la comprensión del lenguaje.

## **TRANSFORMERS: LA REVOLUCIÓN DEL PLN**

Antes de los **Transformers**, los modelos basados en **RNNs y LSTMs** tenían dificultades para manejar secuencias largas y no podían paralelizarse bien. En 2017, Google propuso los **Transformers** en el artículo **"Attention is All You Need"**, revolucionando el campo.

Diagrama

Descripción generada automáticamente

*La arquitectura Transformer original*

### **¿CÓMO FUNCIONA UN TRANSFORMER?**

* Utiliza **mecanismos de autoatención (self-attention)** para identificar relaciones entre palabras sin necesidad de recorrer el texto en orden secuencial.
* Es totalmente **paralelizable**, lo que permite entrenar modelos con miles de millones de parámetros.
* Puede procesar **oraciones enteras a la vez**, en lugar de procesarlas palabra por palabra como hacían los modelos anteriores.

### **MODELOS CLAVE BASADOS EN TRANSFORMERS:**

1. **BERT (2018, Google)**
   * Modelo bidireccional: aprende del contexto antes y después de una palabra.
   * Mejora tareas como la comprensión de texto y la búsqueda en Google.
2. **GPT-3/GPT-4 (2020-2023, OpenAI)**
   * Modelos generativos que pueden escribir textos, programar y responder preguntas con alto nivel de coherencia.
   * Utilizan un enfoque de **aprendizaje autoregresivo**: predicen la siguiente palabra basándose en las anteriores.
3. **T5 (2019, Google)**
   * Modelo multitarea que trata todas las tareas como problemas de generación de texto.
4. **DALL·E y Stable Diffusion (2021-2022)**
   * Modelos basados en Transformers para generar imágenes a partir de descripciones de texto.