FUNDAMENTOS | MÓDULO 02

Fundamentos

Unidad 1: qué hace un LLM, pesos y tokens, alucinación, contexto, y la transición de modelo a asistente (prompt e historial).

Semana 01 | Martes 11 de agosto | 09:40-10:50 | clase

LECTURA COMPLETA

Handbook

Fundamentos

Unidad 1: qué hace un LLM, por qué alucina, capas (modelo / asistente / agente / harness), riesgos y declaración de uso de IA. Modelos, límites, capas y control humano.

Antes, hay que entender qué son. Las palabras agente, AI y LLM están de moda y a menudo se usan de forma incorrecta. En esta lectura se apilan capas: modelo (predice el siguiente token), asistente (responde turno a turno) y agente (decide en un bucle). El harness (ejecuta y acota) y el detalle del agente se profundizan más adelante.

Agenda

1. Modelo: pesos, tokens, alucinación, contexto (de entrada y de salida).
2. Asistente, agente y harness: modelo → asistente → agente → harness.
3. Riesgos y ética: sicofancia, privacidad, declaración de uso.
4. Cierre: portafolio del día.

Modelo y límites

Por detrás de un chatbot hay un modelo. La dinámica conversacional es de asistente (eso viene más adelante). El modelo no piensa como nosotros: no tiene memoria propia, no tiene intención y no verifica el mundo solo.

¿Qué es un modelo (LLM)?

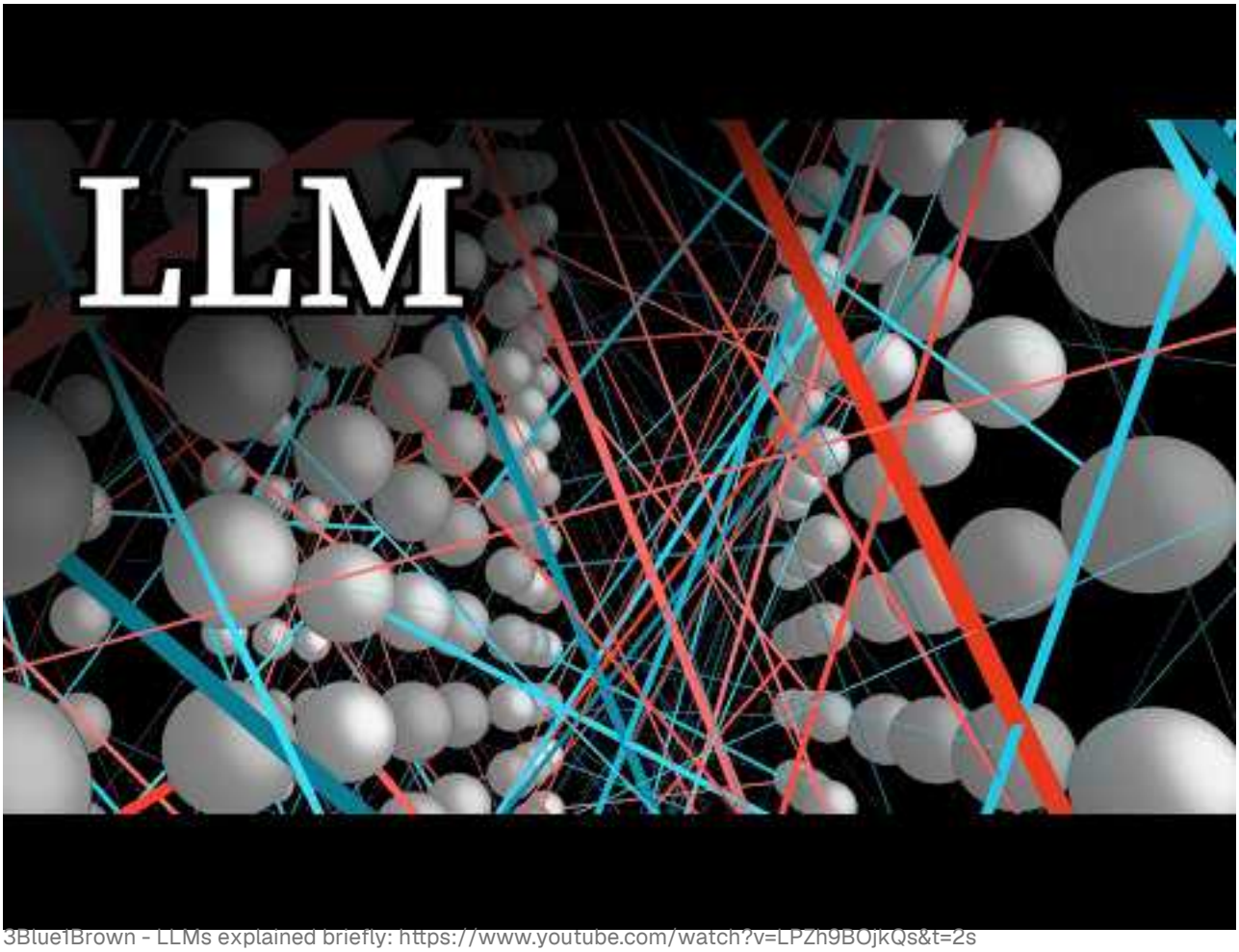
Un LLM (Large Language Model) es un modelo de lenguaje que predice la continuación más probable para un texto, a partir de un contexto inicial.

Un LLM predice la siguiente palabra

Imagina un texto incompleto: "París es una ciudad de _____". Hay varias continuaciones factibles (Francia, Texas, Europa, arte...). El modelo asigna probabilidades; no elige con certeza absoluta. Por eso la misma pregunta puede dar respuestas distintas.

- Dado un texto, propone una continuación factible palabra a palabra.
- Esto se repite hasta completar el texto.
- No elige una sola palabra con certeza: asigna probabilidades.

Video de apoyo: 3Blue1Brown - Large Language Models explained briefly (https://www.youtube.com/watch?v=LPZh9BOjkQs&t=2s).



3Blue1Brown - LLMs explained briefly: https://www.youtube.com/watch?v=LPZh9BOjkQs&t=2s

Entrenamiento y pesos

- Dado un texto, se oculta la última palabra y el modelo predice.
- Se compara la predicción con la palabra real. A mayor error, más se ajustan los pesos.
- Los pesos son números que el entrenamiento ajusta poco a poco a partir de miles de ejemplos.
- En el entrenamiento se ajustan billones de veces, hasta completar texto nuevo de forma que haga sentido.

Tokens: la conexión entre pesos y texto

1. Parte la entrada (input) en tokens (palabra, trozo o signo).
2. Los tokens se convierten en números (lista de valores) para el modelo.
3. Se elige el siguiente token y se agrega al contexto.
4. Se repite el proceso con el nuevo contexto (entrada + nuevo token).

La tokenización no siempre es división por palabras ni por espacios: algunas palabras se parten, otras se juntan, otras se ignoran.

Ejercicio práctico: prueba el tokenizer de OpenAI (https://platform.openai.com/tokenizer). Otro ejemplo de generación en vivo: TokenProbe (Columbia).

TokenProbe: Visualizing LLM Learning in Real-Time (https://tokenprobe.cs.columbia.edu/).

¿Qué es el contexto?

En un LLM, el contexto es todo lo que el modelo tiene a la vista en ese instante para predecir el siguiente token: la instrucción que le entregaste y lo que ya escribió en la misma respuesta. Tiene un tamaño limitado

(medido en tokens): lo que no entra, el modelo no lo puede usar.

Durante el curso vamos a hablar mucho de contexto. Según el tema (modelo, prompt, documentos, conversación, agente...), la palabra apunta a algo parecido, pero no siempre es exactamente lo mismo.

Lo importante a recordar:

- Entrenamiento: ajustar pesos con ejemplos.
- Tokenización: dividir y convertir a números.
- Contexto en LLMs: lo que el modelo puede ver y usar para predecir.

Frontera irregular (jagged frontier)

¿Qué debería ser más fácil: resumir un párrafo o contar cuántas "r" hay en strawberry? Los modelos fallan en tareas que parecen triviales porque no "cuentan" letras: trabajan con tokens, no con caracteres. strawberry no se ve como s-t-r-a-w-b-e-r-r-y, sino como trozos ya armados.

- Capacidades altas en tareas que a nosotros nos parecen difíciles.
- Fallas en otras que nos parecen triviales.
- Los modelos están diseñados para generar respuestas que suenen legítimas.

Por eso siempre es importante verificar la respuesta del modelo.

Alucinación y verificación

Si inventa una cita a un paper, ¿está "mintiendo"? El modelo completa un patrón que suena legítimo.

- En el entrenamiento predice el siguiente token a partir de texto visto.
- Datos raros (una cita, una fecha) dejan poca señal: inventar y acertar pueden sonar igual de bien.
- Si se evalúa solo por "acertó / falló", adivinar gana al "no sé".

Alucinar es rellenar con fluidez cuando falta contexto para predecir.

Cómo evitamos la alucinación

Verificando la respuesta. Si el modelo cita un paper, búscalo por título o identificador. Si da un número, contrástalo con la fuente original o con otra búsqueda. Más adelante en el curso veremos verificación sistemática.

Actividad: auditar citas

En un solo chat de LLM Arena (Direct Chat, sin búsqueda web), corre estas preguntas. Luego verifica cada cita: ¿existe? ¿el detalle cuadra?

Herramienta: lmarena.ai (https://lmarena.ai/).

- Derecho: cita tres fallos de la Corte Suprema de Chile (rol, año, sala y enlace) sobre responsabilidad civil por daños de sistemas de IA, con cita textual de cada fallo.
- Medicina veterinaria: cita completa y DOI de Contreras et al. (2023), Efficacy of ivermectin nanoemulsion against canine demodicosis in Chile, en Veterinary Parasitology.
- Historia: cita completa (archivo, fondo, volumen y foja) de tres documentos del Archivo Nacional Histórico de Chile que prueben que Bernardo O'Higgins prohibió las imprentas privadas en 1818, con transcripción textual.

De modelo a asistente

Manejo de conversación

En el ejercicio de citas el modelo ya no es solo una entrada y una salida: se maneja como conversación. Ahí aparece la diferencia entre modelo y asistente.

- El modelo recibe una entrada y genera una salida. Cada llamada es independiente.
- El asistente recibe una entrada, genera una salida y puede recibir una nueva entrada.
- El modelo no "recuerda" turnos anteriores por sí solo. En cada iteración es una nueva conversación.
- El asistente arma el contexto con el historial de la conversación.
- El modelo solo predice texto plausible a partir de lo que recibe como entrada.

Cuando abren ChatGPT, Claude u OpenWork, ¿usan el modelo o un asistente? Ambos: el modelo es la capa más baja; sobre eso se construye el asistente. El agente (decide en un bucle) y el harness (ejecuta y acota) aparecen como capas siguientes.

1. Modelo
Motor de predicción. Texto entra → continuidad plausible sale. Puede proponer una herramienta a usar; no la ejecuta.

2. Asistente
Producto conversacional sobre un modelo: una prompt inicial + historial de mensajes (asistente y usuario). Patrón turno a turno: tú pides → responde → tú decides el siguiente paso.

¿Qué es un prompt?
El texto de entrada que le das al asistente en un turno: una pregunta, una instrucción o un ejemplo.

¿Qué es el historial de conversación?
La secuencia de mensajes (tuyos y del asistente) que el producto vuelve a incluir en cada nuevo turno. El modelo no "recuerda": el asistente reenvía ese arreglo de textos como parte del contexto.

Con prompt e historial combinados, lo que el asistente le envía al modelo es una lista de mensajes con roles; el modelo genera el siguiente mensaje del asistente:

```
[ { "role": "assistant", "content": "Hola. ¿En qué te ayudo?" }, { "role": "user", "content": "¿Qué es un token?" }, { "role": "assistant", "content": "Una unidad de texto que el modelo procesa." }, { "role": "user", "content": "Explica qué es un token en una frase, con un ejemplo." }]
```

Riesgos, ética y cierre
La agenda de la sesión cierra con riesgos y ética (sicofancia, privacidad, declaración de uso de IA) y con el portafolio del día. El punto de anclaje es el mismo que en alucinación: verificar, acotar qué datos entregas y declarar el uso de IA cuando corresponda.

- Sicofancia: el modelo puede acomodarse demasiado a lo que quieres oír.
- Privacidad: no pegues datos sensibles en chats públicos o sin control de retención.
- Declaración de uso: deja registro de cuándo y cómo usaste IA en entregas del curso.
- Portafolio del día: registra una situación, acción, evidencia y reflexión breve de esta clase.

Spoiler - próxima clase
Ya vimos contexto en el modelo y cómo el asistente arma historial y prompts. La próxima clase abre:

- ¿Cómo se usa el contexto dentro de un asistente?
- ¿Existe alguna forma de controlar el comportamiento del asistente a lo largo de la conversación?
- ¿Cómo podemos empezar a darle más herramientas para que pueda resolver problemas más complejos?

- ¿Cuándo deja de ser un asistente y se convierte en un agente?

RECURSOS

Materiales

- 3Blue1Brown - Large Language Models explained briefly: https://www.youtube.com/watch?v=LPZh9BOjkQs&t=2s
- OpenAI Tokenizer: https://platform.openai.com/tokenizer
- LLM Arena: https://lmarena.ai/
- Programa oficial IIC103G: https://catalogo.uc.cl/index.php?tmpl=component&option=com_catalogo&view=programa&sigla=IIC103G
- OpenWork | documentación oficial: https://openworklabs.com/docs/start-here/get-started