

L U M B R E



GPU y entrenamiento a escala

Grado: 7

Materia: Deep Learning · Inteligencia Artificial

11/10/2026

Antes de empezar

El deep learning es ante todo álgebra lineal masiva y paralela, y ahí una GPU barre a una CPU. Sacar partido al hardware —lotes, precisión reducida, varias tarjetas— decide si un entrenamiento tarda horas o días.

QUÉ APRENDERÁS

- Aprovechar la aceleración por hardware, el dimensionado de lotes y la precisión mixta para entrenar modelos grandes de forma eficiente.

GPU y entrenamiento a escala

Que la GPU no espere

Una GPU multiplica miles de operaciones a la vez, perfecta para las matrices de una red. Pero tenerla no basta: hay que alimentarla con lotes grandes, mover los datos una sola vez y, si se puede, calcular en menos bits.

VIDEO

Precisión mixta en PyTorch

Vídeo en español que muestra el entrenamiento en fp16 para acelerar y ahorrar memoria en GPU.



Escanea para verlo en Lumbre

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

Por qué la GPU

Un entrenamiento es una sucesión de productos de matrices y convoluciones altamente paralelizables. La GPU, con sus múltiples núcleos, calcula esos bloques a la vez y acelera órdenes de magnitud frente a una CPU.

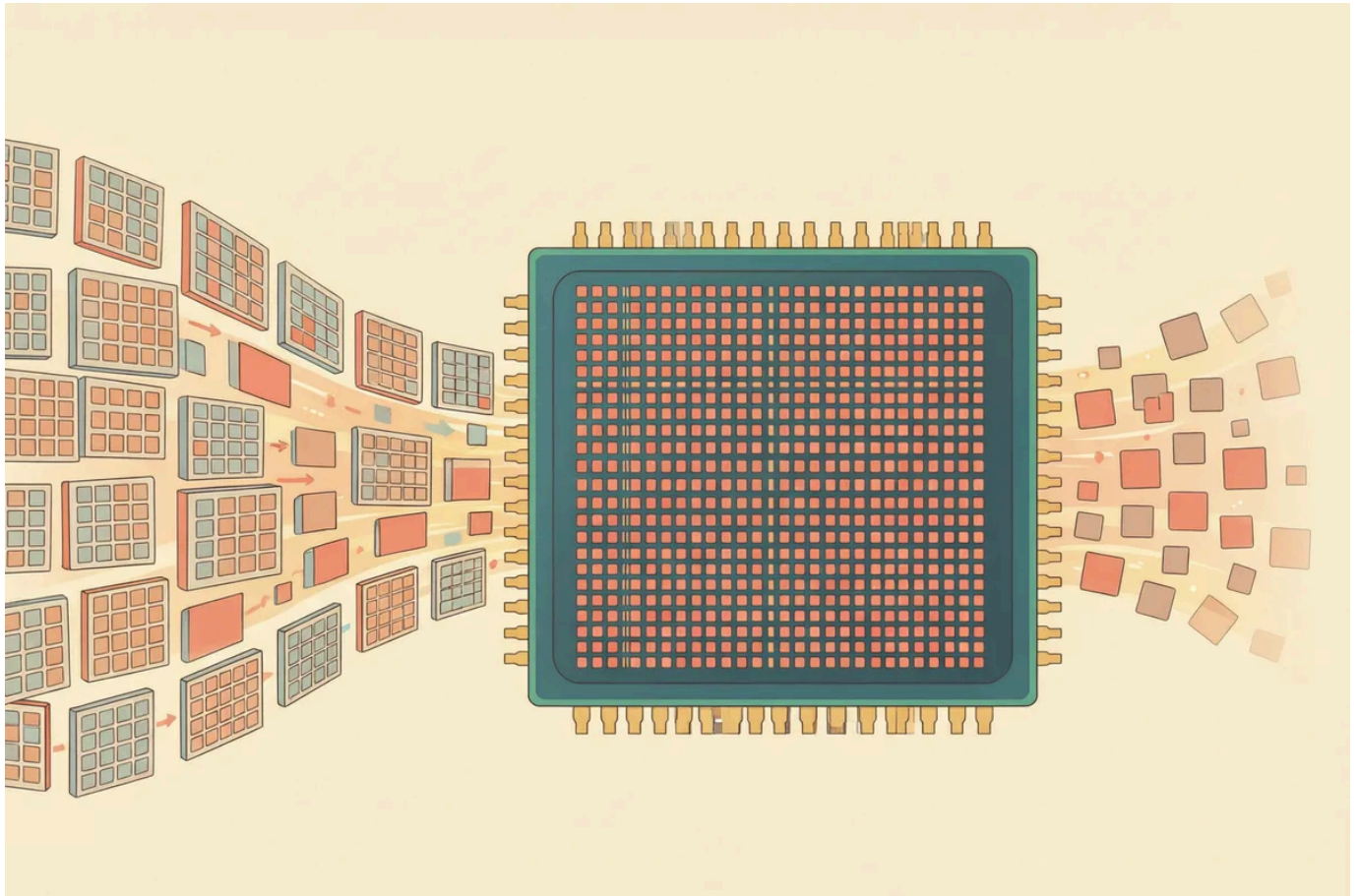
Alimentar la tubería

- Mover los tensores a la GPU y cargarlos bien evita ir y volver desde la CPU.

- Un lote demasiado grande deja de caber en memoria; uno demasiado pequeño infrautiliza los núcleos.
- Carga paralela de datos para que el cálculo no espere a leer.

El tamaño del lote

Lotes grandes dan gradientes más estables y mejor paralelismo, pero menos ruido regularizador y más memoria. Lotes pequeños generalizan a veces mejor y caben en tarjetas modestas. Es un equilibrio entre calidad, memoria y throughput.



La GPU reparte los productos de matrices entre sus núcleos; por eso barre a la CPU en el entrenamiento.

Memoria y su gestión

La memoria de la GPU limita modelo y lote. Concretar, no guardar el grafo donde no hace falta, o acumular gradientes entre micro-lotes permite entrenar redes que de otro modo no cabrían.

Precisión mixta

Calcular en fp16 o bf16 en vez de fp32 reduce memoria y multiplica el ancho de banda casi sin perder precisión, usando un guardián de escala para no desbordar. Es hoy el estándar para entrenar grande.

Paralelismo de datos

Con varias GPU, cada una procesa un sub-lote y se sincronizan los gradientes. Repartir el batch aganta más muestras por segundo casi linealmente mientras el intercambio de gradientes no sature.

Paralelismo de modelo

Cuando un modelo no cabe en una tarjeta, se parte por capas o por tensor entre dispositivos. Es más complejo y comunicativo, pero hace posibles redes gigantes.

Medir el throughput

De nada presume una GPU que se pasa el tiempo esperando datos. Medir muestras por segundo y uso de memoria revela si el límite es el cómputo, el ancho de banda o la tubería de entrada.

Errores frecuentes

- Usar un lote tan grande que desborda la memoria de la GPU y provoca fallos o lentos volcados.
- Olvidar mover modelo y datos al mismo dispositivo, de modo que la GPU queda infrautilizada.
- Activar precisión mixta sin escalado de pérdidas y entrenar con gradientes que se anulan o desbordan.

Ejemplo resuelto: exprimir una GPU sin romper la calidad

Tu entrenamiento cabe justo en una GPU y va lento. Lo aceleras por etapas midiendo en cada una:

1. Mide el throughput base en muestras por segundo con el lote actual.
2. Activa precisión mixta con su guardián de escala para liberar memoria.
3. Sube el lote hasta llenar la memoria sin desbordar y vuelve a medir.
4. Si la GPU aún no satura, mejora la carga paralela de datos.
5. Compara la pérdida y la validación antes de dar por bueno el cambio.

¿Para qué sirve en la realidad?

Saber exprimir el hardware decide si entrenar cuesta horas o días y cuánto se paga en la nube; es lo que permite experimentar a escala en lugar de limitarse a un único intento.

1. La precisión mixta acelera el entrenamiento sobre todo porque:

- ☐ elimina la retropropagación
- ☐ reduce memoria y aumenta el throughput de cálculo usando formatos de menos bits
- ☐ entrena con un solo ejemplo por paso
- ☐ reemplaza la GPU por la CPU

2. Duplicar el número de GPU en paralelo de datos reparte el lote entre tarjetas y sincroniza los gradientes.

- ☐ Verdadero
- ☐ Falso

ACTIVIDAD INTERACTIVA DISPONIBLE EN LUMBRE

3. A escala**4. Ordena la lógica para acelerar un entrenamiento en GPU:****Elementos:**

- asegurar que los datos llegan rápido a la GPU
- elegir un lote que llene la memoria sin desbordarla
- activar precisión mixta para ganar throughput
- escalar a varias GPU si el modelo o el tiempo lo piden

Datos	. . .
Lote	. . .
Precision	. . .
Tarjetas	. . .

GPU https://en.wikipedia.org/wiki/Graphics_processing_unit Procesador gráfico y su papel en el cómputo paralelo.

5. Une cada técnica de escalado con lo que consigue.

5.	calcular en fp16 o bf16 para ahorrar memoria	-----
5.	repartir el lote entre varias tarjetas	-----
5.	dividir un modelo enorme por capas entre dispositivos	-----
5.	medir las muestras por segundo	-----

Opciones: precision mixta · paralelismo de datos · paralelismo de modelo · throughput

Que la GPU no espere

Entrenamiento a escala

Hardware

miles de núcleos

memoria limitada

Lote

gradiente estable

llena la GPU

Precision

fp16 o bf16

guardián de escala

Paralelo

de datos

de modelo

El mapa mental interactivo es interactivo en la versión online.



6. Tu entrenamiento cabe justo en una GPU y va lento. ¿Probarías antes precisión mixta, un lote mayor o una segunda tarjeta? Justifica por qué ese orden.

Referencia curricular

Grado:	7
Materia:	Deep Learning · Inteligencia Artificial

Versión interactiva completa (con actividades, videos y autoevaluación):



<https://univ.lumbre.lat/object/cdia-dep-005-gpu-y-entrenamiento-a-escala>

Material educativo generado desde Lumbre · lumbre.lat