Low-Precision Floating-Point Formats

Análisis Numérico

Daniel Melo

Universidad Nacional De Colombia

September 30, 2026

Uso de la baja precisión en IA

¿Qué está pasando?

- **1985:** IEEE 754 estandariza fp32 y fp64.
- **Hoy:** GPUs e IA imponen formatos de 16 y 8 bits (*bfloat16, FP8*).
- **La Razón:** Menor uso de memoria y cálculos más rápidos.
- **El problema:** Perdemos rango y precisión muy rápido.

Comparativa rápida de formatos

Formato	Bits	Precisión (u)	Rango Máx.
fp64	64	10^{-16}	10^{308}
fp32	32	10^{-8}	10^{38}
fp16	16	10^{-4}	65,504
bfloat16	16	10^{-3}	10^{38}
FP8	8	10^{-1}	57,344

Comparativa

fp16: Buena precisión, pero se desborda fácil.
bfloat16: Gran rango, pero solo $\sim$2 decimales exactos.

Matemáticas en fp16

El estancamiento

- Al sumar números muy grandes con números muy pequeños, el pequeño se pierde.
- Las cotas teóricas de error dejan de funcionar.

Redondeo Estocástico

Redondeamos hacia arriba o abajo con probabilidad.

- Cancela errores en promedio.
- Evita que la suma se congele.

Ejemplo

Operación	Valor Exacto	Resultado
Sumar 1 (5000 veces)	5000	2048
Sumar $\sum 1/k$	∞	7.08

** En la serie armónica, el cálculo se congela por completo al llegar al término $k = 512$.*

Precisión Mixta

Estrategia para resolver $Ax = b$

1. **Trabajo pesado** $O(n^3)$**:** Factorizar $A \approx LU$ en **fp16** (Rápido).
2. **Corrección** $O(n^2)$**:** Calcular residuos y corregir en **fp64** (Preciso).
3. **Repetir** hasta converger.

$\Rightarrow$ Alcanza precisión de fp64 usando el motor rápido de fp16.

Impacto real

En GPUs NVIDIA (V100):

- Hasta $\sim$ **4**$\times$ más rápido que el solver tradicional en fp64.

Referencias:
Pranseh, Srikara (2019) – Low Precision Floating-Point Formats
Haidar et al. (SC18) – Benchmarks en GPU
Connolly, Higham & Mary (2021) – Stochastic Rounding