Parallel MLP Inference Benchmark — Reporte (CPU)

Contexto
Se compara inferencia de un MLP (2 capas densas) usando:
- std::thread (paralelismo manual CPU)
- OpenMP (paralelismo CPU)
- AVX2 (SIMD con intrinsics/FMA)
- CUDA y MPI+CUDA (no ejecutable en este entorno)

Métricas:
- runtime_ms (ms)
- checksum (suma de todos los outputs) para validar correctitud.

Parámetros base:
- D=1024, H=2048, C=1000

2A) Correctness sanity check (CPU)
Orden por cada B: thread → openmp → avx2.

B	Impl	runtime (ms)	checksum
64	thread	68.464	-980.742942
64	openmp	60.796	-980.742942
64	avx2	27.028	-980.742932
256	thread	228.141	-3716.597040
256	openmp	233.613	-3716.597040
256	avx2	88.836	-3716.596763
1024	thread	931.781	-13699.002694
1024	openmp	956.182	-13699.002694
1024	avx2	358.932	-13699.002606

Conclusión 2A: thread y OpenMP coinciden exactamente; AVX2 difiere mínimamente por punto

---

2B) Performance vs Threads (B fijo=8192, variar P)
Fórmula: speedup(P) = time(P=1) / time(P)

Tiempos
P	thread ms	openmp ms
1	6072.505	5911.623
2	7075.293	6535.809
4	7136.380	6572.879
8	6813.282	6485.922
16	6922.903	6632.230

Speedups
P	speedup thread	speedup openmp
1	1.000	1.000
2	0.858	0.904
4	0.851	0.899
8	0.891	0.911
16	0.877	0.891

Observación 2B: no hay escalamiento (speedup < 1). Probable causa: entorno virtualizado/C

2C) Performance vs Problem Size (P=8 fijo)
B	thread ms	openmp ms
64	63.480	50.965
256	216.251	220.725
1024	834.970	1037.814
4096	3468.593	3216.288
8192	7040.069	6478.833
16384	13887.322	12980.510

Observación 2C: para B pequeños el overhead domina; para B grandes (4096+) OpenMP mejora

---

2D) SIMD vs OpenMP (B=8192)
Impl	Config	runtime (ms)	checksum
AVX2	single	2734.862	-117601.063413
OpenMP	P=8	6725.060	-117601.065306
OpenMP	P=16	6500.840	-117601.065306

AVX2 es ~2.4× más rápido que OpenMP en este entorno.

2E) CUDA vs MPI+CUDA (N/A en este entorno)
No ejecutable por:
- falta de ejecutable `mlp_cuda` (no hay nvcc/GPU)
- falta de `mpirun` (OpenMPI no instalado)

Se documentan observaciones esperadas: GPU-only mejora con B grande; MPI+CUDA solo conviene c

---

Conclusiones
1) Correctness validado (checksums consistentes; diferencias mínimas en AVX2 por punto flotan
2) No hubo speedup al aumentar threads (std::thread/OpenMP) para B=8192 en este entorno.
3) En B grandes, OpenMP mejora ligeramente vs thread (pero no escala con P).
4) AVX2 fue el mejor desempeño (~2.4× vs OpenMP).
5) CUDA/MPI no se midieron por limitaciones del entorno.