Clase 114 — Optimizadores modernos: Lion, Sophia, Schedule-Free

Parte: 2 — Deep Learning · Fuente: Chen et al. (2023) Lion + Liu et al. (2023) Sophia + Defazio et al. (2024) Schedule-Free. Duración estimada: 75 min.

Clase 114 — Optimizadores modernos: Lion, Sophia, Schedule-Free

> *Parte: 2 — Deep Learning · Fuente: Chen et al. (2023) Lion + Liu et al. (2023) Sophia + Defazio et al. (2024) Schedule-Free. Duración estimada: 75 min.*

Objetivo

Conocer la nueva generación de optimizadores 2023-2024 que está reemplazando a AdamW en LLM training a escala: Lion (Google, signo del gradiente), Sophia (Stanford, segundo orden aproximado), Schedule-Free (Meta, sin LR scheduling). Saber cuándo justifican el cambio.

Resultados de aprendizaje

Al finalizar, el estudiante podrá:

- Aplicar Lion con LR 3-10× más chico que AdamW y weight_decay 3-10× más grande.
- Aplicar Sophia con estimación diagonal del Hessiano (Hutchinson sampling).
- Aplicar Schedule-Free (schedulefree.AdamWScheduleFree) sin warmup/cosine.
- Comparar memoria, velocidad y calidad final.
- Reconocer cuándo Lion supera AdamW (modelos grandes, ViT, CLIP) y cuándo no.

Temas

- Lion: update = sign(β·m + (1-β)·g). 1 buffer en lugar de 2.
- Sophia: pre-condicionador diagonal del Hessiano vía Hutchinson.
- Schedule-Free: aprende sin schedule explícito, sin warmup.
- Memory: Lion ahorra 50 % vs AdamW.
- Trade-off: Lion + LR alto explota fácilmente.

Definiciones y características

- Lion: signo del gradiente como dirección. β=0.9, β=0.99 típicos.
- Sophia: actualiza θ ← θ - η·clip(m/h, -ρ, ρ). Robusto a hessianas mal condicionadas.
- Schedule-Free: averaging interpolation entre z (live) y x (averaged); no necesita LR schedule.
- Memory cost: Adam 2× params, AdamW 2× params, Lion 1× params, Schedule-Free 2× params.

Dataset / recursos

- Fashion-MNIST o CIFAR-10 + ViT-Tiny.
- Librerías: torch, torch.optim, schedulefree (pip), implementaciones Lion/Sophia community.

Ejercicios

1. AdamW baseline: ViT-Tiny en CIFAR-10. LR=1e-3, wd=0.05.

2. Lion: misma red, LR=1e-4, wd=0.5. Comparar accuracy y memoria.
3. Sophia: con Hutchinson cada 10 steps. Comparar convergencia.
4. Schedule-Free: AdamWScheduleFree(lr=1e-3, warmup_steps=500). Sin cosine.
5. Memory: para modelo grande, medir VRAM con cada uno.

Homework verificable

Comparar 4 optimizadores en ViT-Tiny + CIFAR-100:

1. AdamW (baseline).
2. Lion.
3. Sophia.
4. Schedule-Free AdamW.

Reportar: accuracy final, wall-time, peak VRAM.

Criterio de aceptación: Lion debe ahorrar ≥ 30 % VRAM vs AdamW; al menos uno de los modernos iguala o supera AdamW en accuracy.

Errores comunes

Síntoma / mensaje	Causa y cómo arreglar
Lion con LR de AdamW → loss=nan	LR debe ser 3-10× menor. Fix: 1e-4 a 3e-4
Sophia muy lento	Hutchinson costoso. Fix: estimar hessiano
Schedule-Free + cosine schedule	Conflicto. Fix: NO usar scheduler externo.
Comparar wall-time sin igual cantidad de e	Trampa. Fix: mismo epochs o early-stopping
Lion con weight_decay bajo	Resultados peores. Fix: subir wd 3-10× res

Preguntas frecuentes

Lion o AdamW en 2026?

Para modelos < 100M params: AdamW sigue siendo default seguro. Para LLM training a escala (>1B), Lion ahorra mucha memoria y gana papers (Chen 2023).

Sophia en producción?

Aún experimental. Google reportó 2× speedup en LLM pretraining (770M). Vale probar.

Schedule-Free realmente funciona sin warmup?

Recomienda warmup corto (500-1000 steps). El resto sin cosine.

Implementaciones?

Lion: implementaciones community (lucidrains/lion-pytorch). Sophia: similares. Schedule-Free: schedulefree package oficial Meta.

Lion en CV / fine-tuning?

Sí — paper original lo demostró en ViT, CLIP, modelos de difusión. Especialmente bueno para fine-tuning grande.

Referencias

- Chen et al. (2023), Symbolic Discovery of Optimization Algorithms (Lion), NeurIPS.
- Liu et al. (2023), Sophia: A Scalable Stochastic Second-order Optimizer.
- Defazio et al. (2024), The Road Less Scheduled (Schedule-Free).
- schedulefree.

Siguiente clase

Clase 115 — Learning rate scheduling

Apéndice: notebook (primer bloque)

Lion (Google, 2023) = sign-based update, sin momentum de 2do orden. Sophia (Stanford, 2023) = aproximación diagonal del Hessian. Comparamos contra Adam manual sobre Rosenbrock.

```
import numpy as np
import matplotlib.pyplot as plt

np.random.seed(42)

# Rosenbrock: f(x,y) = (1-x)^2 + 100*(y-x^2)^2; mínimo en (1, 1)
def rosen(w):
    x, y = w
    return (1 - x)2 + 100  (y - x2)*2

def rosen_grad(w):
    x, y = w
    dx = -2(1 - x) - 400x(y - x*2)
    dy = 200(y - x*2)
    return np.array([dx, dy])
```

Archivos complementarios

- notebook.ipynb