Clase 174 — Entrenamiento a escala con Vertex AI

Parte: 2 — Deep Learning · Fuente: Géron, cap. 19 § Running Large Training Jobs on Vertex AI + docs Vertex AI Training. Duración estimada: 65 min.

Clase 174 — Entrenamiento a escala con Vertex AI

> *Parte: 2 — Deep Learning · Fuente: Géron, cap. 19 § Running Large Training Jobs on Vertex AI + docs Vertex AI Training. Duración estimada: 65 min.*

Objetivo

Cierre del bloque de despliegue: lanzar training jobs a escala en Vertex AI (GCP managed) — cluster automático, GPUs/TPUs on-demand, hyperparameter tuning distribuido. Conocer alternativas: AWS SageMaker Training, Azure ML Jobs, y plataformas dedicadas a LLMs (Modal, Together AI).

Resultados de aprendizaje

Al finalizar, el estudiante podrá:

- Empaquetar un training script como Docker container.
- Lanzar un Custom Training Job en Vertex AI con gcloud ai custom-jobs create.
- Configurar HP tuning con Vertex AI Vizier.
- Usar TPU Pods para training distribuido extremo.
- Conocer alternativas cloud-agnostic (Modal, Together, RunPod).

Temas

- Custom container vs prebuilt container en Vertex.
- WorkerPoolSpec: master + workers + parameter servers.
- TPU pods para training a escala.
- Hyperparameter tuning con Vizier (Bayesian search).
- Costos: GPU/TPU hours.
- Alternativas: SageMaker, Modal, Together, RunPod, Lambda Labs.

Definiciones y características

- Custom Training Job: corre tu container en Vertex.
- WorkerPool: cluster spec (machine_type, count, accelerator).
- Vizier: black-box optimizer integrado para HP tuning.
- TPU: tensor processing unit, ASIC de Google.
- Spot instances / preemptible: ~70 % más barato pero pueden ser killed.

Dataset / recursos

- GCP account.
- Librerías: google-cloud-aiplatform.

Ejercicios

1. Dockerize: escribir Dockerfile con TF + tu training script.

2. Lanzar job: aiplatform.CustomJob(display_name='exp1', worker_pool_specs=[...]).run().
3. HP tuning: HyperparameterTuningJob con Vizier; 20 trials.
4. Multi-GPU spec: machine_type='a2-highgpu-4g' (4× A100).
5. TensorBoard integration: Vertex TB para monitor.

Homework verificable

Lanzar un training job de Fashion-MNIST en Vertex:

1. Containerizar.
2. Job de 1 GPU (n1-standard-4 + T4).
3. Logs y output a GCS.
4. Verificar accuracy similar a entrenamiento local.

Criterio de aceptación: el job termina con accuracy ≥ 0.87 y el modelo queda guardado en GCS.

Errores comunes

Síntoma / mensaje	Causa y cómo arreglar
Job falla al iniciar	Image no accesible. Fix: push a Artifact R
Olvidé apagar y factura $1000	Vertex jobs son time-limited pero HP tunin
Logs no aparecen	Tarda 1-2 min en stream. Fix: paciencia o
Multi-GPU pero job usa 1	TF no detecta GPUs en el container. Fix: i
Spot interrumpe a mitad	Aceptable para experimentos. Fix: checkpoi

Preguntas frecuentes

¿Cuándo Vertex vs local?

Local para iteración. Cloud cuando: dataset no entra en RAM local, training > 1 día, HP tuning con muchos trials, multi-GPU/TPU.

¿TPU vs GPU?

TPU brilla en modelos TF/JAX grandes (LLMs, transformers de visión). GPU es más universal. PyTorch en TPU funciona (vía XLA) pero menos óptimo.

¿Spot/preemptible?

Para training con checkpointing, ahorra 70 %. Para inference o jobs cortos, evitar.

¿Modal vs Vertex?

Modal: DX excelente, billing por segundo, ideal para LLMs e inference serverless. Vertex: más enterprise features (compliance, IAM, audit).

¿Together AI / Replicate cuándo?

Cuando solo necesitás API a modelos open-source preentrenados (LLMs, difusión). No para training desde cero.

Referencias

- Géron, cap. 19 — Running Large Training Jobs on Vertex AI.
- Vertex AI Training docs.
- SageMaker Training.
- Modal, Together AI, RunPod.

Siguiente parte

Clase 175 — Distribuciones: normal, binomial, Poisson, exponencial

Apéndice: notebook (primer bloque)

Primera celda ejecutable del notebook de la clase.

```
# Imports y configuración inicial
```

Archivos complementarios

- notebook.ipynb