Ampliación de Fundamentos de Hardware

Angel Fabrizio Ullaguari Yanza

2ºASIR

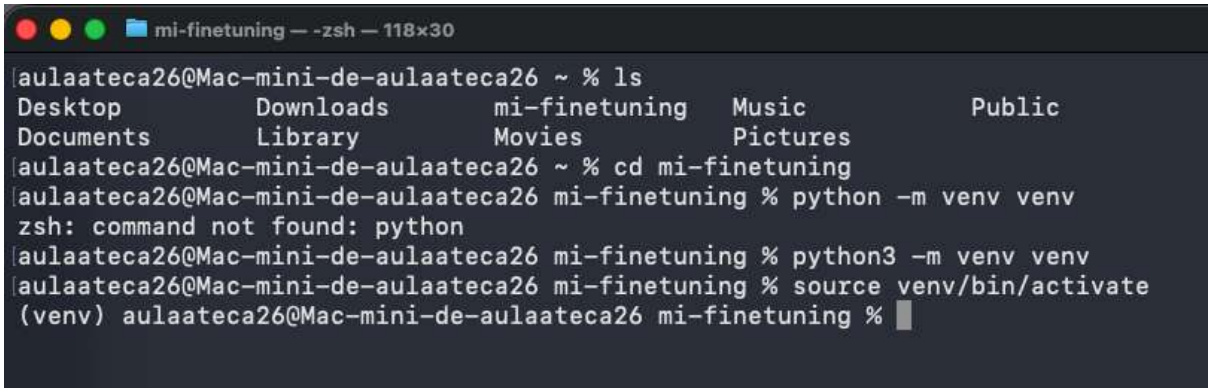
Práctica IA. Fine tuning

Práctica

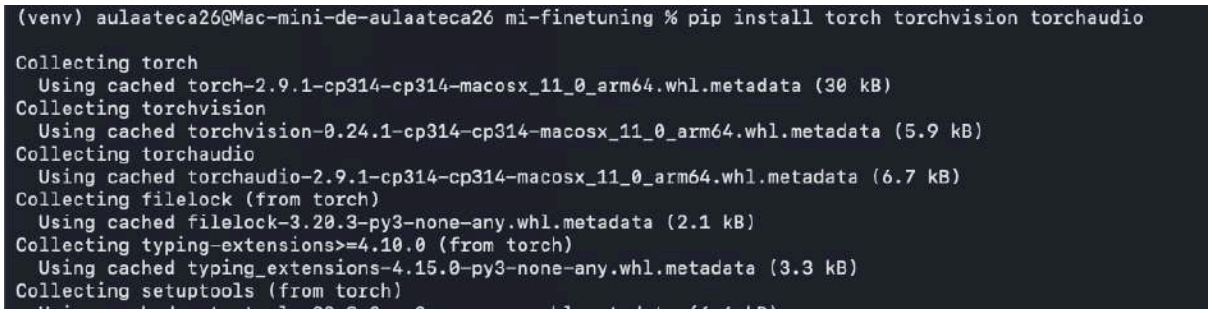
Configuración

Vamos a realizar un fine-tuning con el modelo Gemma-3-270M.

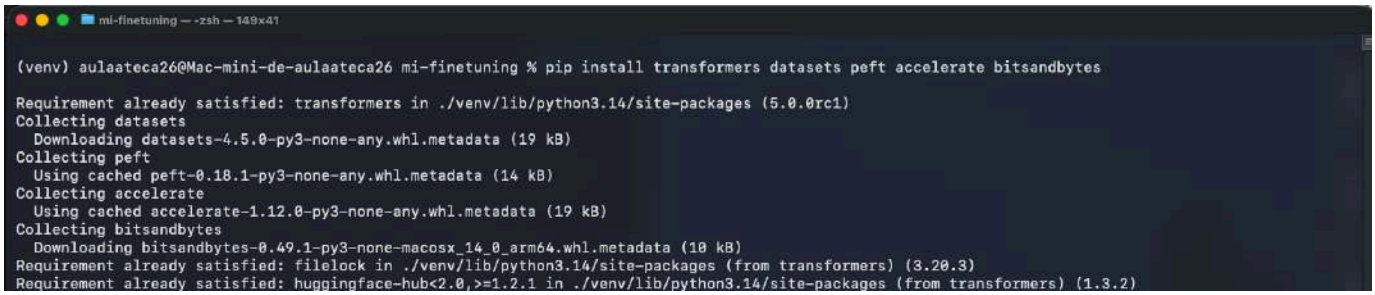
Primero, creamos la carpeta mi-finetuning para el proyecto. Luego, se crea un entorno virtual con python y lo activamos.



Ahora, se instala torch y sus dependencias para que gestionen el modelo.

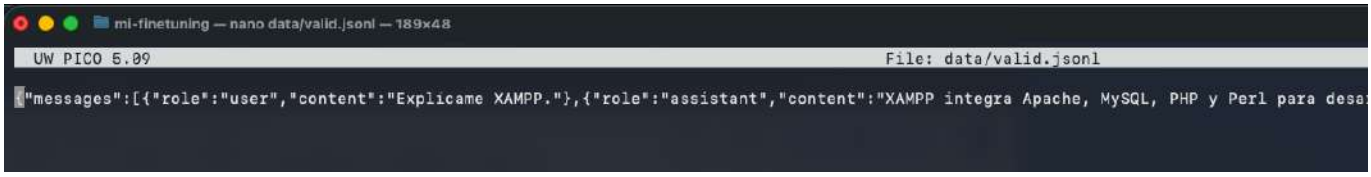


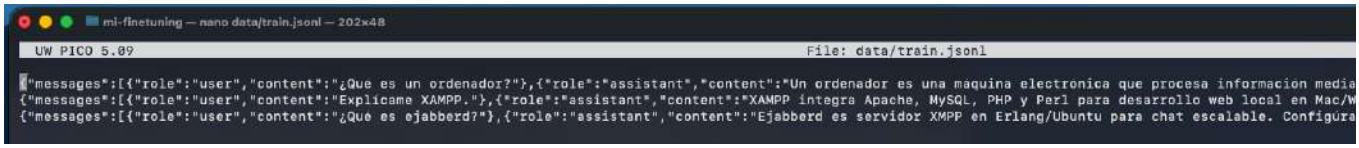
También hay que instalar transformers (biblioteca de Hugging Face) para manejar los modelos de lenguaje, además de peft (es una herramienta para usar LoRA, que permite entrenar modelos grandes con poca memoria), y bitsandbytes (permite reducir el peso del modelo).



En datasets editamos estos dos archivos.
train.jsonl (Entrenamiento): Son las preguntas y respuestas que la IA usa para estudiar. Es su libro de texto.

valid.jsonl (Validación): Son preguntas que la IA usa para hacerse un examen a sí misma y ver si está aprendiendo bien o si solo está memorizando.





Pruebas

Entrenamos a nuestro modelo.

```
(venv) aulaateca26@Mac-mini-de-aulaateca26 mi-finetuning % TOKENIZERS_PARALLELI
SM=false mlx_lm.lora --model mlx-community/gemma-3-270m-it-4bit --train --data
data --iters 100 --batch-size 1 --adapter-path checkpoints_ordenador --save-eve
ry 50

Loading pretrained model
Fetching 10 files: 100%|████████████████████| 10/10 [00:00<00:00, 128266.18it/s]
Download complete: : 0.00B [00:00, ?B/s]                | 0/10 [00:00<?, ?it/s]
Loading datasets
Training
Trainable parameters: 0.629% (1.688M/268.098M)
Starting training..., iters: 100
Calculating loss...: 100%|████████████████████████| 1/1 [00:00<00:00,  8.54it/s]
Iter 1: Val loss 9.081, Val took 0.119s
Iter 10: Train loss 4.417, Learning Rate 1.000e-05, It/sec 10.006, Tokens/sec 4
97.281, Trained Tokens 497, Peak mem 0.370 GB
Iter 20: Train loss 1.042, Learning Rate 1.000e-05, It/sec 27.101, Tokens/sec 1
336.075, Trained Tokens 990, Peak mem 0.370 GB
Iter 30: Train loss 0.308, Learning Rate 1.000e-05, It/sec 27.015, Tokens/sec 1
323.753, Trained Tokens 1480, Peak mem 0.370 GB
Iter 40: Train loss 0.245, Learning Rate 1.000e-05, It/sec 27.038, Tokens/sec 1
343.799, Trained Tokens 1977, Peak mem 0.370 GB
Iter 50: Train loss 0.201, Learning Rate 1.000e-05, It/sec 27.038, Tokens/sec 1
332.955, Trained Tokens 2470, Peak mem 0.370 GB
Iter 50: Saved adapter weights to checkpoints_ordenador/adapters.safetensors an
d checkpoints_ordenador/0000050_adapters.safetensors.
Iter 60: Train loss 0.162, Learning Rate 1.000e-05, It/sec 26.936, Tokens/sec 1
319.844, Trained Tokens 2960, Peak mem 0.379 GB
Iter 70: Train loss 0.132, Learning Rate 1.000e-05, It/sec 27.011, Tokens/sec 1
331.620, Trained Tokens 3453, Peak mem 0.379 GB
Iter 80: Train loss 0.099, Learning Rate 1.000e-05, It/sec 27.018, Tokens/sec 1
342.801, Trained Tokens 3950, Peak mem 0.379 GB
Iter 90: Train loss 0.069, Learning Rate 1.000e-05, It/sec 27.010, Tokens/sec 1
323.467, Trained Tokens 4440, Peak mem 0.379 GB
Calculating loss...: 100%|████████████████████████| 1/1 [00:00<00:00, 58.03it/s]
Iter 100: Val loss 0.029, Val took 0.019s
Iter 100: Train loss 0.045, Learning Rate 1.000e-05, It/sec 26.987, Tokens/sec
1330.449, Trained Tokens 4933, Peak mem 0.379 GB
Iter 100: Saved adapter weights to checkpoints_ordenador/adapters.safetensors a
nd checkpoints_ordenador/0000100_adapters.safetensors.
Saved final weights to checkpoints_ordenador/adapters.safetensors.
(venv) aulaateca26@Mac-mini-de-aulaateca26 mi-finetuning %
```

```
(venv) aulaateca26@Mac-mini-de-aulaateca26 mi-finetuning % TOKENIZERS_PARALLELISM=false mlx_lm.lora --model mlx-community/gemma-3-270m-it-4bit --train --data data --iters 1000 --batch-size 1 --adapter-path checkpoints_ordenador --save-every 50
```

Con la ayuda de nano, configuramos el archivo test_gemma.py para poder hablar con el modelo recién entrenado.