
Ampliación de Fundamentos de Hardware

Angel Fabrizio Ullaguari Yanza

2ºASIR

Práctica IA. Preentrenamiento



Objetivo	3
Requisitos	3
Configuración y prueba	3

Objetivo

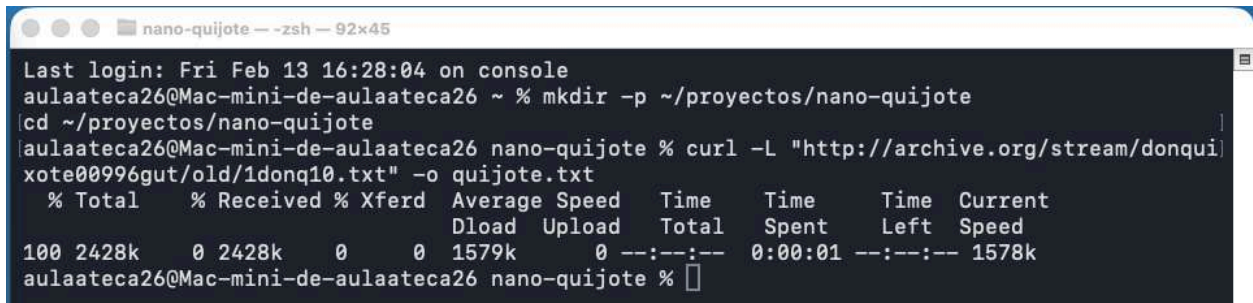
Entrenar un modelo de lenguaje muy pequeño desde cero, usando únicamente el texto de "Don Quijote de la Mancha". El modelo no sabrá nada del mundo ni del lenguaje moderno; Su único universo será el castellano del siglo XVII de Cervantes. Al final, deberá ser capaz de generar texto que imite ese estilo.

Requisitos

- Python.
- La librería PyTorch (pip install torch).
- El texto de "Don Quijote".

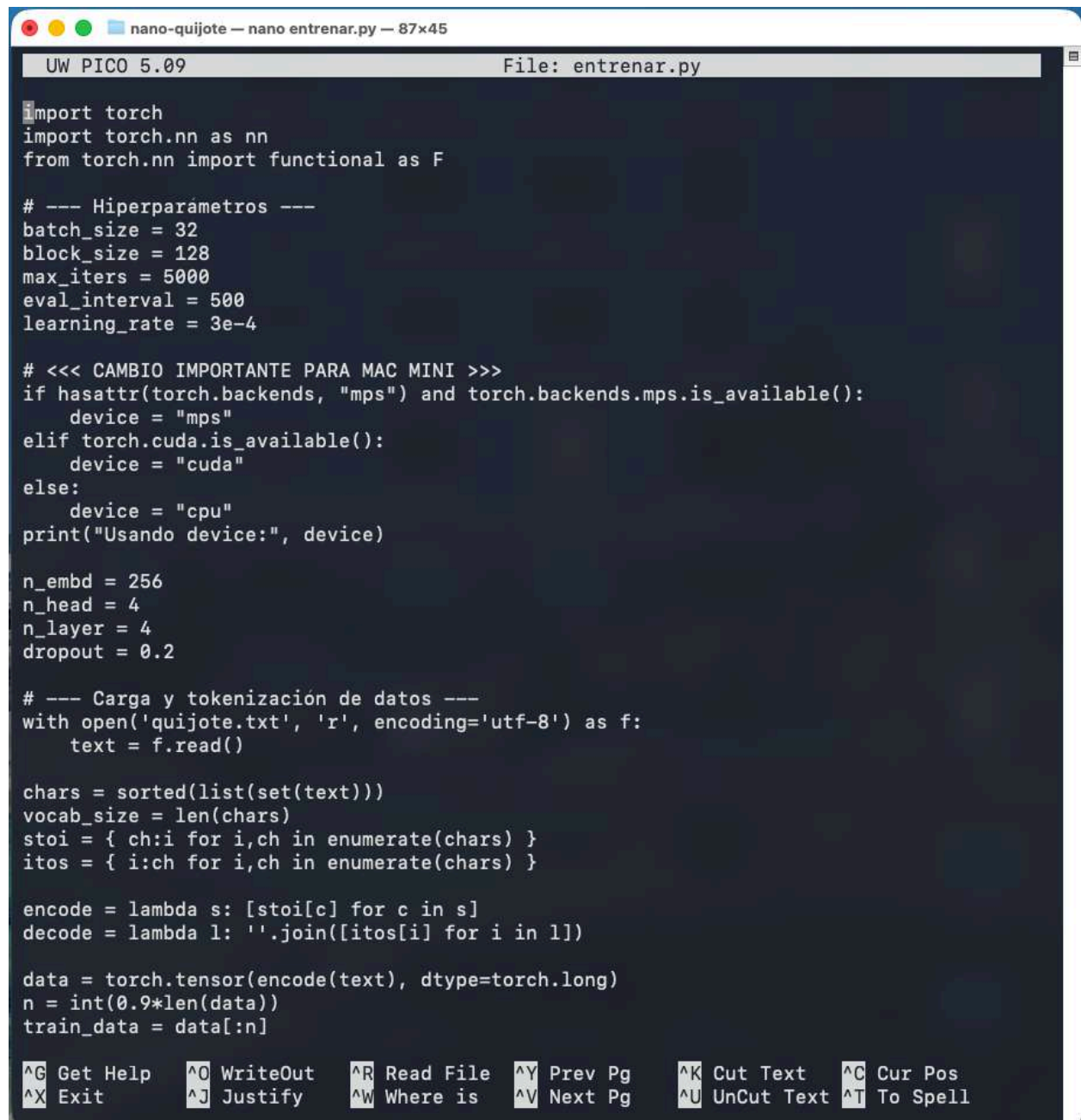
Configuración y prueba

Creemos la carpeta de nano-quijote, donde guardaremos el modelo y descargamos el texto de Don Quijote.



```
nano-quijote -- zsh -- 92x45
Last login: Fri Feb 13 16:28:04 on console
aulaateca26@Mac-mini-de-aulaateca26 ~ % mkdir -p ~/proyectos/nano-quijote
cd ~/proyectos/nano-quijote
aulaateca26@Mac-mini-de-aulaateca26 nano-quijote % curl -L "http://archive.org/stream/donqui
xote00996gut/old/1donq10.txt" -o quijote.txt
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left   Speed
100 2428k    0 2428k    0     0 1579k      0 --:--:--  0:00:01 --:--:-- 1578k
aulaateca26@Mac-mini-de-aulaateca26 nano-quijote %
```

Ahora creamos con nano el archivo entrenar.py e implantamos el código con el texto.



```
UW PICO 5.09 File: entrenar.py

import torch
import torch.nn as nn
from torch.nn import functional as F

# --- Hiperparámetros ---
batch_size = 32
block_size = 128
max_iters = 5000
eval_interval = 500
learning_rate = 3e-4

# <<< CAMBIO IMPORTANTE PARA MAC MINI >>>
if hasattr(torch.backends, "mps") and torch.backends.mps.is_available():
    device = "mps"
elif torch.cuda.is_available():
    device = "cuda"
else:
    device = "cpu"
print("Usando device:", device)

n_embd = 256
n_head = 4
n_layer = 4
dropout = 0.2

# --- Carga y tokenización de datos ---
with open('quijote.txt', 'r', encoding='utf-8') as f:
    text = f.read()

chars = sorted(list(set(text)))
vocab_size = len(chars)
stoi = { ch:i for i,ch in enumerate(chars) }
itos = { i:ch for i,ch in enumerate(chars) }

encode = lambda s: [stoi[c] for c in s]
decode = lambda l: ''.join([itos[i] for i in l])

data = torch.tensor(encode(text), dtype=torch.long)
n = int(0.9*len(data))
train_data = data[:n]
```

^G Get Help ^O WriteOut ^R Read File ^Y Prev Pg ^K Cut Text ^C Cur Pos
^X Exit ^J Justify ^W Where is ^V Next Pg ^U UnCut Text ^T To Spell

Guardamos los cambios y ejecutamos el archivo. Como se puede ver, devuelve el texto de Don Quijote, traduciendo el castellano del siglo XVII al inglés.

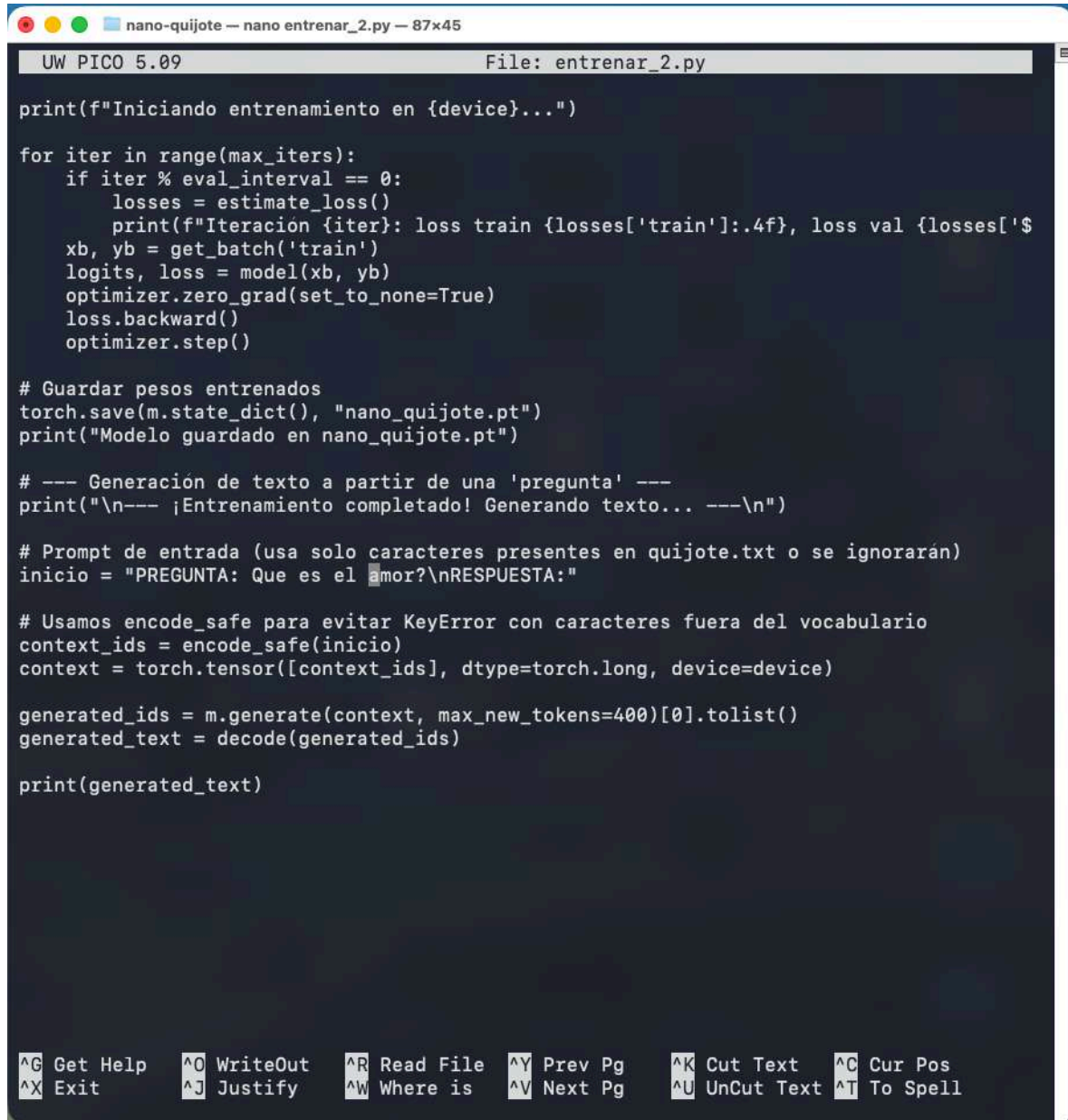
```
(venv-nano-quijote) aulaateca26@Mac-mini-de-aulaateca26 nano-quijote % nano entrenar.py
(venv-nano-quijote) aulaateca26@Mac-mini-de-aulaateca26 nano-quijote % cd ~/proyectos/nano-q
uijote
python entrenar.py

Usando device: mps
Iniciando entrenamiento en mps...
Iteración 0: loss train 4.6951, loss val 4.6886
Iteración 500: loss train 2.2466, loss val 2.2365
Iteración 1000: loss train 1.8719, loss val 1.8713
Iteración 1500: loss train 1.7045, loss val 1.7282
Iteración 2000: loss train 1.5994, loss val 1.6462
Iteración 2500: loss train 1.5296, loss val 1.5968
Iteración 3000: loss train 1.4791, loss val 1.5531
Iteración 3500: loss train 1.4339, loss val 1.5198
Iteración 4000: loss train 1.3948, loss val 1.5019
Iteración 4500: loss train 1.3694, loss val 1.4797

--- ¡Entrenamiento completado! Generando texto... ---

without
ducheded to him to a lous her expectites smade of all that stow Moriestor's
true, Seristiggles for Mancha (who encesty Princheding to
what, that you will was surm and very about yat to be so quited
pleace in good uponinguanced, I conce to very immenshinged to over
thit beaves, it was to Sparant Pettin, I had be scounced hide of the
serves like Don Quiteat contry. Don Fante that the whole head to
all put the certables; like no would, cart as which not are
seeing afterly plaing a didle mu
(venv-nano-quijote) aulaateca26@Mac-mini-de-aulaateca26 nano-quijote %
```

Creemos otro archivo llamado `entrenar_2.py`. Este sigue la estructura aprendida del texto de Don Quijote.



```
UW PICO 5.09 File: entrenar_2.py

print(f"Iniciando entrenamiento en {device}...")

for iter in range(max_iters):
    if iter % eval_interval == 0:
        losses = estimate_loss()
        print(f"Iteración {iter}: loss train {losses['train']:.4f}, loss val {losses['$
xb, yb = get_batch('train')
logits, loss = model(xb, yb)
optimizer.zero_grad(set_to_none=True)
loss.backward()
optimizer.step()

# Guardar pesos entrenados
torch.save(m.state_dict(), "nano_quijote.pt")
print("Modelo guardado en nano_quijote.pt")

# --- Generación de texto a partir de una 'pregunta' ---
print("\n--- ¡Entrenamiento completado! Generando texto... ---\n")

# Prompt de entrada (usa solo caracteres presentes en quijote.txt o se ignorarán)
inicio = "PREGUNTA: Que es el amor?\nRESPUESTA:"

# Usamos encode_safe para evitar KeyError con caracteres fuera del vocabulario
context_ids = encode_safe(inicio)
context = torch.tensor([context_ids], dtype=torch.long, device=device)

generated_ids = m.generate(context, max_new_tokens=400)[0].tolist()
generated_text = decode(generated_ids)

print(generated_text)

^G Get Help ^O WriteOut ^R Read File ^Y Prev Pg ^K Cut Text ^C Cur Pos
^X Exit ^J Justify ^W Where is ^V Next Pg ^U UnCut Text ^T To Spell
```


Y lo ejecutamos. Se puede ver que al realizar las 5000 iteraciones, el valor “loss” ha bajado de un 1.86 hasta un 1.47.

```
(venv-nano-quijote) aulaateca26@Mac-mini-de-aulaateca26 nano-quijote % cd ~/proyectos/nano-quijote
source venv-nano-quijote/bin/activate
python entrenar_2.py

Usando device: mps
Iniciando entrenamiento en mps...
Iteración 0: loss train 4.7645, loss val 4.7756
Iteración 500: loss train 2.2365, loss val 2.2108
e
Iteración 1000: loss train 1.8528, loss val 1.8652
Iteración 1500: loss train 1.6936, loss val 1.7298
Iteración 2000: loss train 1.5962, loss val 1.6468
Iteración 2500: loss train 1.5297, loss val 1.6062
Iteración 3000: loss train 1.4758, loss val 1.5616
Iteración 3500: loss train 1.4425, loss val 1.5331
Iteración 4000: loss train 1.4136, loss val 1.5052
Iteración 4500: loss train 1.3877, loss val 1.4793
Modelo guardado en nano_quijote.pt

--- ¡Entrenamiento completado! Generando texto... ---

PREGUNTA: Que es el amor?
RESPUESTA:

"Will there Riconfles, it what Sancho, "go tie,- it hour
light returned subdings in the king, that the judge musking sizal."

Fining in some datifide, "Need to the
"by who says!" said Sancho Mairallon's.

"Yexak it is proppared they deperds are and plet assom what
an
among; in the miserud is the gier of them Comilian, when he had . At the
othermour and without or in this own endugnifice him in
(venv-nano-quijote) aulaateca26@Mac-mini-de-aulaateca26 nano-quijote %
```