

ollama e modelli hugging face

Ultimo aggiornamento: 2026-09-18

Ollama e modelli Hugging Face: installazione, download e test

Passi pratici per installare la CLI hf, scaricare i modelli GGUF, creare un Modelfile e testare velocità e modelli installati. Ambiente Linux con NVIDIA.

1. Installare la CLI Hugging Face (hf)

Se hf non è ancora presente:

```
curl -LsSf https://hf.co/cli/install.sh | bash
```

2. Vedere le dimensioni di un modello PRIMA di scaricarlo

Tecnica --dry-run: mostra i file e le dimensioni senza scaricare nulla.

```
# Elenca tutto il contenuto del repo
hf download \
  unsloth/NVIDIA-Nemotron-3.5-Lightning-30B-A3B-GGUF \
  --include "*.gguf" \
  --exclude "BF16/*" \
  --dry-run

# Forma corta equivalente
hf download \
  unsloth/NVIDIA-Nemotron-3.5-Lightning-30B-A3B-GGUF --dry-run

# Un altro modello di esempio
hf download \
  llmfan46/gemma-4-26B-A4B-it-ultra-uncensored-heretic-GGUF --dry-run
```

3. Scaricare un singolo file GGUF in una cartella locale

```
# modalità ad alte prestazioni xet
export HF_XET_HIGH_PERFORMANCE=1

hf download \
  unsloth/NVIDIA-Nemotron-3.5-Lightning-30B-A3B-GGUF \
  NVIDIA-Nemotron-3.5-Lightning-30B-A3B-MXFP4_M0E.gguf \
  --local-dir /localIA/llamacpp/models/Nemotron-3.5-Lightning-30B-A3B_MXFP4

# esempio Gemma 4 26B it qat
mkdir -p /localIA/llamacpp/models/gemma4-26b-a4b-qat

HF_XET_HIGH_PERFORMANCE=1 hf download \
  unsloth/gemma-4-26B-A4B-it-qat-GGUF \
  gemma-4-26B-A4B-it-qat-UD-Q4_K_XL.gguf \
  --local-dir /localIA/llamacpp/models/gemma4-26b-a4b-qat
```

4. Creare un modello Ollama da un GGUF locale (Modelfile)

Crea il file Modelfile puntando alla cartella di download:

```
FROM /localIA/llamacpp/models/gemma4-26b-a4b-qat/gemma-4-26B-A4B-it-qat-UD-Q4_K_XL.gguf

PARAMETER num_ctx 8192
PARAMETER num_batch 2048
PARAMETER num_predict 4096
```

Registra il modello in Ollama:

```
ollama create gemma4-q4xl-local \
-f /localIA/llamacpp/models/gemma4-26b-a4b-qat/Modelfile.ollama
```

5. Misurare la velocità di un modello

```
ollama run qwen3.6:35b-a3b --verbose
```

6. Usare llama-server in alternativa a Ollama

Installazione CLI llama.cpp:

```
curl -LsSf https://llama.app/install.sh | sh
llama --version
llama serve --list-devices
```

Per fermare Ollama e lasciare la GPU a llama-server:

```
sudo systemctl stop ollama
systemctl is-active ollama      # atteso: inactive
nvidia-smi                     # la GPU deve risultare libera
```

Esempio di avvio di llama serve con un GGUF locale:

```
FABLE_GGUF="/localIA/ollama_models/qwen36-fable/Qwen3.6-27B-Fable-Fus-711-UnHeretic-NM-DAU-NEO-MAX-NEO-Q4_K_M.
gguf"
```

```
llama serve \
-m "$FABLE_GGUF" \
--no-mmproj \
--host 0.0.0.0 \
--port 8080 \
--ctx-size 8192 \
--parallel 1 \
--flash-attn on \
--cache-type-k q8_0 \
--cache-type-v q8_0 \
--fit on \
--fit-target 1024 \
--batch-size 2048 \
--ubatch-size 512 \
--reasoning off \
--jinja \
--perf
```

7. Tabella di confronto modelli (riferimento)

#	Modello	Active	Dimensione	Note
1	Qwen3.6-28B REAP	~3B	17,3 GB	Baseline attuale
2	Gemma 4 26B-A4B-it	4B	17,0 GB	Miglior candidato non-Qwen
3	GLM-4.7-Flash 30B-A3B	3B	18,5 GB	Ragionamento e coding
4	Qwen3.6-35B-A3B completo	3B	22,3 GB	Probabile tetto qualitativo
5	GPT-OSS-20B	3,6B	12,1 GB	Riferimento di velocità
6	Nemotron 3.5 Lightning 30B-A3B	3B	25,5 GB	Esperimento più pesante

8. Verificare i modelli installati (e le loro capacità)

```
ollama list | awk 'NR > 1 {print $1}' |  
while read -r model; do  
    printf '%-25s ' "$model"  
    ollama show "$model" |  
    awk '/Capabilities/{p=1; next} p && NF==0{exit} p{printf "%s ", $1} END{print ""}'  
done
```

9. Vedere il Modelfile di un modello

```
ollama list  
ollama show qwen3.8:27b --modelfile
```