Documentazione dei Parametri di Training

Questo documento descrive in dettaglio tutti i parametri di training utilizzati nel progetto di SVG captioning.

Modelli

Llama 3.1 8B Instruct

- **Hugging Face ID:** meta-llama/Llama-3.1-8B-Instruct
- **Dimensione:** 8 miliardi di parametri
- **Tipo:** Decoder-only, instruction-tuned
- **Contesto:** 8192 token
- **Quantizzazione:** 4-bit (nf4)
- **Job attivi:**
  - `llama_te` (2578401): 2 GPU, configurazione Test 3 Multi-GPU

Gemma 2 9B IT

- **Hugging Face ID:** google/gemma-2-9b-it
- **Dimensione:** 9 miliardi di parametri
- **Tipo:** Decoder-only, instruction-tuned
- **Contesto:** 8192 token
- **Quantizzazione:** 4-bit (nf4)
- **Specifiche:** Richiede implementazione 'eager' dell'attenzione
- **Job attivi:**
  - `gemma_te` (2578402): 2 GPU, configurazione Test 3 Multi-GPU

Parametri LoRA

Rank (r)

- **Descrizione:** Determina la dimensione della matrice di adattamento LoRA.
- **Valori testati:**
  - Test 1 & 2: r=4 (conservativo)
  - Test 3 & Test 3 Multi-GPU: r=16 (più espressivo)
- **Impatto:** Un valore più alto permette di catturare più informazioni ma aumenta il numero di parametri trainabili.

Alpha

- **Descrizione:** Controlla l'impatto dell'adattamento LoRA.
- **Valori testati:**
  - Test 1 & 2: alpha=8 (conservativo)
  - Test 3 & Test 3 Multi-GPU: alpha=32 (più impattante)
- **Impatto:** Un valore più alto aumenta l'effetto dell'adattamento.

Target Modules

- **Descrizione:** Moduli del modello a cui viene applicato LoRA.
- **Valori testati:**
  - Test 1 & 2: ["q_proj", "v_proj"] (solo query e value)
  - Test 3 & Test 3 Multi-GPU: ["q_proj", "v_proj", "k_proj", "o_proj"] (tutti i componenti dell'attenzione)
- **Impatto:** Più moduli target permettono un adattamento più completo ma aumentano i parametri trainabili.

Dropout

- **Descrizione:** Tasso di dropout applicato durante il training LoRA.
- **Valore:** 0.05 (costante in tutti i test)
- **Impatto:** Aiuta a prevenire l'overfitting.

Parametri di Ottimizzazione

Learning Rate

- **Descrizione:** Tasso di apprendimento per l'ottimizzatore.
- **Valori testati:**
  - Test 1 & 2: 2e-4
  - Test 3 & Test 3 Multi-GPU: 1e-5 (ridotto per stabilità)
- **Impatto:** Un valore più basso porta a una convergenza più lenta ma più stabile.

Batch Size

- **Descrizione:** Dimensione del batch per GPU.
- **Valori testati:**
  - Test 1: 1
  - Test 2: 4 (multi-GPU)
  - Test 3: 2 (ridotto per stabilità)
  - Test 3 Multi-GPU: 4 per GPU (batch effettivo 8 con 2 GPU)

- **Impatto:** Un batch size più grande permette un training più efficiente ma richiede più memoria.

Gradient Accumulation Steps

- **Descrizione:** Numero di step di accumulo del gradiente prima dell'aggiornamento dei pesi.
- **Valori testati:**
  - Test 1: 16
  - Test 2: 2 (ridotto per multi-GPU)
  - Test 3: 4 (aumentato per compensare il batch size ridotto)
  - Test 3 Multi-GPU: 1 (nessun accumulo, aggiornamento ad ogni batch)
- **Impatto:** Permette di simulare batch size più grandi con memoria limitata.

Weight Decay

- **Descrizione:** Regolarizzazione L2 per prevenire l'overfitting.
- **Valore:** 0.01 (costante in tutti i test)
- **Impatto:** Aiuta a prevenire l'overfitting penalizzando pesi grandi.

Warmup Ratio

- **Descrizione:** Percentuale di step di training dedicati al warmup del learning rate.
- **Valori testati:**
  - Test 1 & 2: 0.03
  - Test 3 & Test 3 Multi-GPU: 0.05 (aumentato per stabilità)
- **Impatto:** Un warmup più lungo aiuta a stabilizzare il training iniziale.

Optimizer

- **Descrizione:** Algoritmo di ottimizzazione.
- **Valore:** paged_adamw_8bit (costante in tutti i test)
- **Impatto:** Versione ottimizzata di AdamW che utilizza la quantizzazione a 8 bit per risparmiare memoria.

LR Scheduler

- **Descrizione:** Scheduler del learning rate.
- **Valore:** cosine (costante in tutti i test)
- **Impatto:** Decadimento cosinusoidale che riduce gradualmente il learning rate.

Parametri di Training

Epoche

- **Descrizione:** Numero di epoche di training.
- **Valore:** 3 (costante in tutti i test)
- **Impatto:** Determina quante volte il modello vede l'intero dataset.

Gradient Checkpointing

- **Descrizione:** Tecnica per risparmiare memoria durante il training.
- **Valore:** abilitato (costante in tutti i test)
- **Impatto:** Riduce il consumo di memoria a scapito di un leggero aumento del tempo di calcolo.

Early Stopping

- **Descrizione:** Ferma il training quando la metrica di validazione smette di migliorare.
- **Configurazione:**
  - Patience: 100 step
  - Min Delta: 0.0001
  - Metrica: loss di validazione
- **Impatto:** Previene l'overfitting fermando il training quando il modello smette di migliorare.

Salvataggio Checkpoint

- **Descrizione:** Configurazione per il salvataggio dei checkpoint durante il training.
- **Configurazione:**
  - Frequenza: ogni 100 step (Test 3 & Test 3 Multi-GPU) o 200 step (Test 1 & 2)
  - Limite totale: 3 (Test 1 & 2) o 5 (Test 3 & Test 3 Multi-GPU) checkpoint
  - Salvataggio del miglior modello: abilitato (in base alla loss di validazione)
- **Impatto:** Permette di riprendere il training e salvare il miglior modello.

Configurazioni DeepSpeed

Test 1 & 2

- **Configurazione:** ZeRO-2
- **Offload:** CPU (parametri e optimizer)
- **Precisione:** bf16
- **Impatto:** Permette di addestrare modelli più grandi distribuendo il carico su più GPU.

Test 3 & Test 3 Multi-GPU

- **Configurazione:** Disabilitato
- **Impatto:** Semplifica il debug e l'identificazione di problemi.

Parametri di Quantizzazione

Quantizzazione

- **Descrizione:** Tecnica per ridurre la precisione dei pesi del modello.
- **Configurazione:**
  - Tipo: 4-bit (nf4)
  - Compute dtype: bfloat16
  - Double quantization: abilitata
- **Impatto:** Riduce significativamente il consumo di memoria permettendo di caricare modelli più grandi.

Configurazione Hardware

Test 1

- **GPU:** 1x GPU (L40S 48GB o A40 48GB)
- **Memoria:** 32GB
- **CPU:** 8 core

Test 2

- **GPU:** 2x GPU (L40S 48GB o A40 48GB)
- **Memoria:** 32GB
- **CPU:** 8 core

Test 3

- **GPU:** 1x GPU (L40S 48GB o A40 48GB)
- **Memoria:** 32GB
- **CPU:** 8 core

Test 3 Multi-GPU

- **GPU:** 2x GPU (L40S 48GB o A40 48GB)
- **Memoria:** 48GB
- **CPU:** 8 core

Configurazione Software

- **Framework:** PyTorch 2.2.0 + Transformers 4.51.3 + PEFT 0.15.1

- **Ambiente:** Python 3.10 (svg_captioning_env)
- **Librerie aggiuntive:**
  - BitsAndBytes 0.41.3 (per quantizzazione)
  - DeepSpeed (per Test 1 & 2)
  - Weights & Biands (per tracking)

Evoluzione dei Test

Test 1 (Convergence)

- **Obiettivo:** Training fino a convergenza
- **Caratteristiche:** Single-GPU, parametri LoRA conservativi (r=4, alpha=8)
- **Problemi riscontrati:** Convergenza lenta, possibile underfitting

Test 2 (Multi-GPU)

- **Obiettivo:** Accelerare il training con multi-GPU
- **Caratteristiche:** 2 GPU, batch size aumentato (4), gradient accumulation ridotto (2)
- **Problemi riscontrati:** Loss sempre 0.0, grad_norm NaN

Test 3 (Semplificato)

- **Obiettivo:** Risolvere i problemi di training e migliorare la stabilità
- **Caratteristiche:** Single-GPU, parametri LoRA aumentati (r=16, alpha=32), DeepSpeed disabilitato, learning rate ridotto (1e-5)
- **Miglioramenti:** Debug avanzato, early stopping sulla loss di validazione, target modules estesi

Test 3 Multi-GPU

- **Obiettivo:** Combinare i miglioramenti del Test 3 con l'accelerazione multi-GPU
- **Caratteristiche:** 2 GPU per modello, parametri LoRA aumentati (r=16, alpha=32), DeepSpeed disabilitato, learning rate ridotto (1e-5)
- **Miglioramenti:** Maggiore velocità di training mantenendo la stabilità del Test 3
- **Job attivi:**
  - `llama_te` (2578401): Llama 3.1 8B su 2 GPU
  - `gemma_te` (2578402): Gemma 2 9B IT su 2 GPU

Job Attualmente in Esecuzione

Job di Training

1. **`llama_te` (2578401)**
   - **Modello:** Llama 3.1 8B Instruct
   - **GPU:** 2x (L40S 48GB o A40 48GB)
   - **Configurazione:** Test 3 Multi-GPU
   - **Parametri LoRA:** r=16, alpha=32, target=["q_proj", "v_proj", "k_proj", "o_proj"]
   - **Batch Size:** 4 per GPU (effettivo 8)
   - **Gradient Accumulation:** 1 (nessun accumulo)
   - **Learning Rate:** 1e-5
   - **Stato:** In attesa (PD)
2. **`gemma_te` (2578402)**
   - **Modello:** Gemma 2 9B IT
   - **GPU:** 2x (L40S 48GB o A40 48GB)
   - **Configurazione:** Test 3 Multi-GPU
   - **Parametri LoRA:** r=16, alpha=32, target=["q_proj", "v_proj", "k_proj", "o_proj"]
   - **Batch Size:** 4 per GPU (effettivo 8)
   - **Gradient Accumulation:** 1 (nessun accumulo)
   - **Learning Rate:** 1e-5
   - **Stato:** In attesa (PD)

Job di Valutazione

1. **`test_svg` (2578252)**
   - **Descrizione:** Test di rendering SVG con diverse librerie
   - **Tempo limite:** 24 ore
   - **Numero di campioni:** 5
   - **Stato:** In esecuzione (R)
2. **`ext_capt` (2578232)**
   - **Descrizione:** Valutazione di captioner esterni (BLIP, vit-gpt2, cogvlm)
   - **Tempo limite:** 24 ore
   - **Metriche:** BLEU-4, ROUGE-L, METEOR, CIDEr, CLIP Score
   - **Stato:** In esecuzione (R)

Analisi dei Parametri Trainabili con LoRA: Gemma 2 9B e Llama 3.1 8B

Gemma 2 9B IT

Configurazione LoRA

- **Rank (r)**: 16
- **Alpha**: 32
- **Target Modules**: ["q_proj", "v_proj", "k_proj", "o_proj"]
- **Dropout**: 0.05

Analisi Parametri

- **Parametri Totali**: ~9B
- **Parametri Trainabili**: ~4.5M (0.05% del totale)
- **Memoria GPU**: ~48GB (con quantizzazione 4-bit)
- **Batch Size**: 4 per GPU (totale 8 con 2 GPU)
- **Learning Rate**: 1e-5

Ottimizzazioni

- Implementazione attenzione: 'eager'
- Quantizzazione: 4-bit (nf4)
- Compute dtype: bfloat16
- Double quantization: abilitata

Llama 3.1 8B

Configurazione LoRA

- **Rank (r)**: 16
- **Alpha**: 32
- **Target Modules**: ["q_proj", "v_proj", "k_proj", "o_proj"]
- **Dropout**: 0.05

Analisi Parametri

- **Parametri Totali**: ~8B
- **Parametri Trainabili**: ~4.1M (0.051% del totale)
- **Memoria GPU**: ~48GB (con quantizzazione 4-bit)
- **Batch Size**: 4 per GPU (totale 8 con 2 GPU)
- **Learning Rate**: 1e-5

Ottimizzazioni

- Quantizzazione: 4-bit (nf4)
- Compute dtype: bfloat16
- Double quantization: abilitata

Confronto e Analisi

Efficienza Memoria

- Entrambi i modelli utilizzano quantizzazione 4-bit per ridurre il consumo di memoria
- La tecnica LoRA riduce i parametri trainabili a meno dello 0.1% del totale
- Il batch size è ottimizzato per l'utilizzo di 2 GPU

Performance Training

- Gemma 2 9B IT:
  - Loss di validazione: 0.310 (checkpoint migliore)
  - Convergenza più rapida
  - Implementazione 'eager' dell'attenzione
- Llama 3.1 8B:
  - Loss di validazione: 0.516 (checkpoint migliore)
  - Training più stabile
  - Architettura più semplice

Ottimizzazioni Comuni

- Early stopping con patience=100
- Weight decay=0.01
- LR scheduler di tipo cosine
- Warmup ratio=0.05
- Gradient checkpointing abilitato

Conclusioni

- Entrambi i modelli mostrano un'efficienza simile in termini di parametri trainabili
- Gemma 2 9B IT mostra una convergenza migliore ma richiede più memoria
- Llama 3.1 8B offre un buon compromesso tra performance e risorse
- La configurazione LoRA attuale (r=16, alpha=32) si è dimostrata efficace per entrambi i modelli