Rapporto Tecnico

Ottimizzazione di TinyViT tramite Distillazione, Pruning e Quantizzazione per Sistemi Embedded
Autore: Davide Iannella
Corso: Progetto Deep Learning
Anno accademico: 2024–2025

1. Obiettivi del progetto

Il presente progetto ha l'obiettivo di ottimizzare un modello di tipo Vision Transformer (TinyViT) per l'esecuzione efficiente su dispositivi embedded (come Raspberry Pi, Android o microcontrollori compatibili con TensorFlow Lite). L'ottimizzazione è stata effettuata attraverso una pipeline composta da:
- Knowledge Distillation: trasferimento di conoscenza da un modello teacher (ResNet-18) a uno student (TinyViT).
- Pruning: semplificazione strutturale del modello mediante disattivazione di moduli non essenziali.
- Quantizzazione: esportazione del modello finale in formato `.tflite`, compatibile con architetture edge.

L'obiettivo finale è ottenere un modello leggero, preciso e riutilizzabile, adatto ad applicazioni reali embedded senza perdita di performance significative.

2. Descrizione dei modelli e tecniche utilizzate

Modello Teacher:
È stata utilizzata una ResNet-18, preaddestrata su ImageNet e fine-tuned su CIFAR-10 (10 classi). Si tratta di un modello leggero e ben adattabile al compito di classificazione.

Modello Student (TinyViT):
TinyViT (`vit_tiny_patch16_224`) è un Vision Transformer compatto, progettato per operare in ambienti con risorse limitate. La sua architettura è ibrida (CNN + Transformer) con attenzione locale e positional encoding ottimizzato.

Tecniche principali:
- Distillazione: Lo student apprende le predizioni del teacher attraverso una loss combinata (KLDivLoss per avvicinare le distribuzioni di output + CrossEntropyLoss sulle etichette).
- Pruning: disattivazione selettiva dei dropout nei Transformer block.
- Quantizzazione: conversione del modello a TFLite float32 con la libreria `ai-edge-torch`, rendendolo eseguibile su CPU embedded.

3. Dataset e Pipeline di Addestramento

Dataset:
È stato utilizzato CIFAR-10, un dataset composto da 10 classi di immagini a bassa risoluzione (32×32 pixel). Per compatibilità con ViT, le immagini sono state ridimensionate a 224×224.

Subset:
- Training set: 4000 immagini
- Test set: 1000 immagini
- Batch size: 32
- Ottimizzatore: Adam (lr = 3e-4)
- Epoche: 10

Pipeline:
1. Preprocessing CIFAR-10 e caricamento modelli
2. Distillazione student da teacher ResNet-18
3. Pruning (rimozione dropout)
4. Verifica test accuracy
5. Quantizzazione TFLite
6. Test su modello `.tflite`

4. Risultati e Verifica

Fase	Risultati principali
Distillazione	Accuracy test > 90%, training stabile
Pruning	Nessuna perdita significativa, modello più leggero
TFLite export	Modello funzionante in ambiente CPU, float32
Verifica TFLite	Predizioni corrette, output coerente con modello base

La verifica finale è stata condotta con script Python e TensorFlow Lite. È stato scelto un campione dal test set, ottenendo predizione corretta, visualizzazione della confidence Softmax e grafico con matplotlib. Il modello è considerato embedded-ready.

5. Problematiche incontrate

Durante lo sviluppo sono emerse alcune criticità:

- Bilanciamento difficile tra CE e KLDivLoss → Tuning empirico del peso nella funzione di loss.
- Pruning potenzialmente distruttivo → Disattivati solo dropout, evitando layer strutturali.
- Limitazioni TFLite int8 → Esportazione in float32 per mantenere compatibilità.
- Supporto GPU non disponibile in TFLite → Inferenza testata solo su CPU.

6. Conclusioni e sviluppi futuri

Il progetto ha dimostrato come sia possibile ottenere un modello Vision Transformer compatto, preciso ed eseguibile su dispositivi embedded attraverso una pipeline completa di ottimizzazione. Le tecniche applicate sono coerenti con la letteratura e replicabili in altri contesti edge-aware.

Prossimi sviluppi possibili:
- Addestramento completo su CIFAR-10
- Quantizzazione int8 con compatibilità ARM
- Deploy effettivo su Raspberry Pi o ambiente mobile

Il modello TinyViT distillato e prunato rappresenta un'ottima base per applicazioni reali di classificazione visiva embedded.