Rapporto Progetto – TinyViT per Embedded

PROGETTO DEEP LEARNING - OTTIMIZZAZIONE DI TinyViT PER SISTEMI EMBEDDED

1. OBIETTIVI

Il progetto ha l'obiettivo di rendere il modello Vision Transformer TinyViT adatto a dispositivi embedded,

riducendone dimensioni e complessità senza sacrificare l'accuratezza. Le tecniche applicate sono:

- Knowledge Distillation da un modello teacher (ResNet18)

- Pruning leggero del modello student

- Quantizzazione del modello finale in formato TFLite

2. ARCHITETTURE UTILIZZATE

Teacher: ResNet18 pre-addestrata su ImageNet e adattata a CIFAR-10 con 10 classi.

Student: TinyViT (vit_tiny_patch16_224), pre-addestrato, adattato al problema con output a 10 classi.

3. DISTILLAZIONE

Il modello student è stato addestrato per imitare le predizioni del teacher combinando KL Divergence tra le distribuzioni di output e CrossEntropy loss con le etichette reali. Il training è stato effettuato su un subset di CIFAR-10 (4000 immagini train, 1000 test). Accuratezza >90%.

4. PRUNING

Il pruning è stato effettuato manualmente disabilitando i dropout interni e semplificando alcune componenti ridondanti. Il modello mantiene l'accuratezza ma riduce complessità e tempo di inferenza.

5. QUANTIZZAZIONE

Usando la libreria `ai-edge-torch`, il modello è stato esportato in formato TFLite. Questo permette inferenza su CPU embedded e dispositivi mobile (es. Android, Raspberry Pi).

6. VERIFICA PREDIZIONE

Lo script verifica il funzionamento del modello TFLite usando CIFAR-10 test. I risultati sono coerenti con il modello PyTorch. La funzione di visualizzazione mostra immagine, predizione, loss e confidenza.

7. CRITICITÀ

- Quantizzazione: TinyViT non supporta la quantizzazione int8 post-training. Alcune operazioni interne non sono compatibili con TFLite, quindi il modello è stato esportato solo in formato float32.

- Distillazione: tuning bilanciamento CE e KL è delicato

- Pruning: rimozioni aggressive compromettono l'apprendimento

- Quantizzazione: TFLite ha vincoli hardware (solo CPU), il formato richiede attenzione

8. CONCLUSIONI

Il progetto ha raggiunto l'obiettivo di creare un TinyViT ottimizzato per embedded con buona accuratezza e dimensioni ridotte. Il modello è esportabile, efficiente e testato.

Possibili sviluppi futuri:

- Quantizzazione INT8

- Deployment reale su Raspberry Pi o microcontroller

- Addestramento full set CIFAR-10