

## # Progetto Deep Learning: Ottimizzazione di TinyViT per Embedded

Questo repository contiene tutti i file e le cartelle relativi al progetto svolto per il corso di Deep Learning, il cui obiettivo stato l'ottimizzazione di un Vision Transformer (TinyViT) per il deployment su dispositivi embedded.

### ## Obiettivi del progetto

- Ridurre la complessit di TinyViT tramite:
  - Knowledge Distillation (da ResNet18)
  - Pruning (disattivazione dropout)
  - Esportazione in formato TFLite (float32)
- Adattare il modello al dataset CIFAR-10 (10 classi, immagini 32x32)
- Validare il funzionamento su ambiente embedded (simulazione CPU)

### ## Struttura del pacchetto consegna

...

PACCHETTO\_CONSEGNA/

|

- ARCHITETTURA\_MODELLO/ # Diagrammi, descrizioni e struttura architeturale dei modelli
- CODICE\_PROGETTO/ # Script Python organizzati per addestramento, verifica, esportazione
- FONTI\_PAPER/ # Riferimenti scientifici, paper in PDF o link utili
- MODELLO\_DISTILLATO\_PRUNATO/ # Checkpoint del modello student dopo distillazione e pruning
- MODELLO\_ORIGINALE/ # Versione originale del modello student pre-distillazione
- MODELLO\_TFLITE\_EMBEDDED/ # File .tflite esportato per dispositivi embedded (es. Android, Raspberry)
- SCREEN\_SHOT\_PROGETTO/ # Screenshot dell'interfaccia, risultati, inferenze
- |
- Giustificazione\_Scientifica\_TinyViT.pdf # Motivazione scientifica del progetto
- presentazione\_pechakucha.pdf # Presentazione finale (formato PechaKucha - 20 slide)
- presentazione\_pechakucha.pptx # Presentazione modificabile in PowerPoint

...

## ## Dipendenze principali

- torch, torchvision
- timm (per ViT e TinyViT)
- tensorflow (per validazione TFLite)
- ai-edge-torch (per esportazione TFLite da PyTorch)
- matplotlib, numpy

## ## Note tecniche

- La distillazione effettuata su CIFAR-10, sebbene TinyViT sia già stato distillato durante il pretraining su ImageNet-21k. Il fine-tuning distillato ha senso per l'adattamento al task specifico.
- La quantizzazione int8 non è stata possibile, in quanto TinyViT non è compatibile con TFLite quantization post-training standard.
- Il modello finale esportato in `.tflite` (float32) ed è pronto per inferenza su dispositivi embedded CPU.

## ## Autore

Davide Iannella

Progetto Deep Learning - Laurea Magistrale Ingegneria Informatica