UNIVERSITÀ
DEGLI STUDI
DI PADOVA

DIPARTIMENTO
DI INGEGNERIA
DELL'INFORMAZIONE

DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE

CORSO DI LAUREA IN INGEGNERIA INFORMATICA

Progetto Programmazione di Sistemi Embedded: Mobile AI

Autori
Alessandro Girlanda, Andrea Mutti, Umberto Bianchin

ANNO ACCADEMICO 2023-2024

Indice

Elenco delle figure

Capitolo 1

NNAPI

Capitolo 2

PyTorch Mobile

2.1 PyTorch & PyTorch Mobile

PyTorch[2] è un framework di deep learning open source, sviluppato inizialmente da *Meta AI* e ora parte della *Linux Foundation*. Progettato con Python, viene usato per creare reti neurali e per progetti di apprendimento automatico, combinando la libreria di machine learning **Torch**[6] con un'API di alto livello basata su Python. Torch è famosa, specialmente nel campo del Deep Learning, per fornire tool semplici e flessibili insieme a performance elevate; uno dei sui punti salienti è il grande supporto per le GPU, che contribuisce ad un allenamento più efficiente dei modelli di deep learning. PyTorch fornisce innanzitutto un pacchetto Python per funzionalità ad alto livello, come l'elaborazione dei **tensori**[1] ed inoltre un così detto **TorchScript**, che permette di creare modelli da PyTorch che possono poi venire salvati e caricati in un processo dove non c'è alcuna dipendenza di Python.

PyTorch Mobile[4], introdotto per la prima volta nel 2019 alla *PyTorch Developer Conference*, si riferisce ad un set di librerie e funzionalità fornite da PyTorch che permettono allo sviluppatore di eseguire un modello PyTorch direttamente su dispositivi mobili, come smartphone e tablet.

2.2 Caratteristiche Principali

Le caratteristiche principali di PyTorch, così come scritto sul sito ufficiale[5], sono:

- Disponibile per iOS, Android e Linux;

[1] Array multidimensionale utilizzato per memorizzare dati. Nel campo del Machine Learning vengono usati per rappresentare e manipolare input, pesi e output.

- Fornisce API per comuni compiti di pre-processing e integrazione necessari ad incorporare Machine Learning nelle applicazioni mobile;
- Supporta la libreria XNNPACK per le CPU Arm e integra QNNPACK per i kernel a 8 bit quantizzati;
- Fornisce un efficiente interprete mobile per Android e iOS;
- Supporterà a breve backend hardware come GPU, DSP e NPU.

XNNPACK[1] è una libreria altamente ottimizzata per accelerare le operazioni di reti neurali convoluzionali (CNN) e altre operazioni di reti neurali su hardware mobile, mentre QNNPACK[3] (Quantized Neural Network PACKage) è progettata per accelerare le reti neurali quantizzate[2] su hardware mobile.

2.3 Da un modello PyTorch a PyTorch Mobile

Il tipico flusso dalla creazione del modello in PyTorch all'implementazione sul dispositivo mobile può essere visionato in figura 2.1; di seguito verranno spiegati i vari step.

2.3.1 Costruire un modello in PyTorch

2.3.2 Quantizzazione

2.3.3 Scripting e Tracing del modello

2.3.4 Ottimizzazione

[2]La quantizzazione è un processo che riduce la precisione dei numeri usati nei calcoli di una rete neurale, da 32-bit a 8-bit o meno, riducendo così l'uso della memoria e migliorando le prestazioni senza sacrificare significativamente l'accuratezza.

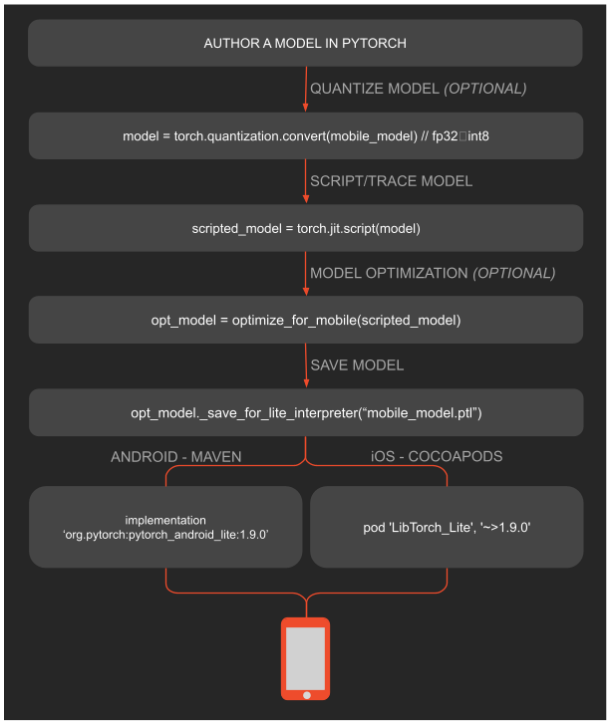

Figura 2.1: Workflow dal training al rilascio di un modello su piattaforma mobile. Fonte [5]

Capitolo 3

TensorFlow

TensorFlow è una libreria open source per l'apprendimento automatico, il calcolo numerico e altre attività di analisi statistica e predittiva. Questo tipo di tecnologia, sviluppata e rilasciata da Google nel novembre 2015, rende l'implementazione di modelli di machine learning più semplice e veloce per gli sviluppatori, assistendo nel processo di acquisizione dei dati, nella formulazione di previsioni su larga scala e nel successivo affinamento dei risultati.

Lo scopo principale di TensorFlow è la creazione e l'addestramento di reti neurali, che possono essere utilizzate per moltissime applicazioni, quali:

- Classificazione delle immagini;
- Elaborazione del linguaggio naturale;

Figura 3.1: Diagramma dei punteggi di utilizzo di vari framework nel 2018

Capitolo 4

PyTorch Mobile vs TensorFlow Lite

Bibliografia

[1] Marat Dukhan. *Accelerating TensorFlow Lite with XNNPACK Integration*. 2020. URL: https://blog.tensorflow.org/2020/07/accelerating-tensorflow-lite-xnnpack-integration.html.

[2] IBM. *Cos'è PyTorch?* 2024. URL: https://www.ibm.com/it-it/topics/pytorch.

[3] Hao Lu Marat Dukhan Yiming Wu. *QNNPACK: Open source library for optimized mobile deep learning*. 2018. URL: https://engineering.fb.com/2018/10/29/ml-applications/qnnpack/.

[4] Sujatha Mudadla. *PyTorch Mobile*. 2023. URL: https://medium.com/@sujathamudadla1213/pytorch-mobile-a5dc9cabe511.

[5] *PyTorch Mobile: End-to-end workflow from Training to Deployment for iOS and Android mobile devices*. URL: https://pytorch.org/mobile/home/.

[6] *Torch (machine learning)*. 2024. URL: https://en.wikipedia.org/wiki/Torch_(machine_learning).