Relazione Progetto MLOps con Fashion-MNIST

Di Christian Dalla Valle VR500076

Introduzione

Un sistema di Machine Learning per essere effettivamente valido e duraturo deve adottare metodologie di ingegneria che assicurino ripetibilità, automatizzazione e ampliabilità. Proprio qui si inserisce il metodo MLOps, che trasferisce i concetti del DevOps nel percorso di vita dei modelli di Machine Learning. Il progetto si propone di realizzare una struttura MLOps in grado di amministrare tutte le fasi essenziali: dalla valutazione preliminare dei dati (EDA), all'istruzione del modello, fino alla verifica e alla messa in opera tramite la containerizzazione.

Dataset

Per questo progetto ho scelto di usare **Fashion-MNIST**, un dataset open source sviluppato da Zalando Research, spesso considerato un sostituto moderno del classico MNIST. Contiene 70.000 immagini in scala di grigi di dimensioni 28x28 pixel, suddivise in 10 classi di articoli di abbigliamento come scarpe, magliette, cappotti e borse. Il dataset è diviso in 60.000 immagini per il training e 10.000 per il test. Rispetto a MNIST, Fashion-MNIST è più complesso pur mantenendo un formato leggero e facilmente utilizzabile. Il dataset viene scaricato automaticamente attraverso la libreria **torchvision** al primo utilizzo.

Analisi Esplorativa dei Dati (EDA)

Prima di addestrare un modello, è importante comprendere la distribuzione dei dati. Lo script **eda.py** permette di eseguire un'analisi esplorativa, campionando 2000 immagini dal dataset e visualizzando la distribuzione delle classi. Vengono inoltre generate una griglia di immagini che aiuta a verificare la varietà del dataset e la qualità delle immagini e altri grafici (istogramma, grafico a barre, PCA 2D). Questa fase è fondamentale per individuare eventuali squilibri nelle classi o problemi nei dati, e sta alla base del corretto addestramento del modello.

Modello di rete neurale

Il modello implementato è una CNN (Convolutional Neural Network) custom, definita nella classe **SmallCNN**. L'architettura è volutamente semplice e adatta al dataset Fashion-MNIST.
Il modello è stato scritto manualmente in **PyTorch** e non utilizza modelli pre-addestrati.
Questa architettura permette di ottenere buoni risultati senza eccessiva complessità.

Training e Valutazione

Il file **train.py** si occupa dell'addestramento del modello e della sua validazione. Durante il ciclo di training, vengono eseguiti i seguenti passaggi:

- Forward pass e calcolo della loss.
- Backpropagation e aggiornamento dei pesi.
- Registrazione della loss media per ogni epoca.

In parallelo, si svolge una fase di valutazione sul set di test, dove si calcolano l'accuratezza e la loss media. Tutti i risultati vengono salvati in un file JSON chiamato **metrics.json**, mentre il modello addestrato viene archiviato in formato **PyTorch** (**final_tensor.pt**). Inoltre, i log del training sono disponibili in **train.log**. Per una valutazione rapida di un modello salvato, è possibile utilizzare anche lo script **evaluate.py.**

Struttura del progetto

Il progetto è stato organizzato seguendo le pratiche di sviluppo software. Il codice è racchiuso in un pacchetto Python (**src/mlops_fmnist**), i test sono contenuti in una cartella dedicata (**tests/**) e i task ricorrenti sono automatizzati tramite un **Makefile**. Tra i comandi disponibili: creazione ambiente virtuale, installazione dipendenze, training, valutazione, linting, test, build del pacchetto e gestione Docker.

CI/CD e Docker

Un aspetto fondamentale dell'approccio MLOps è l'automazione tramite Continuous Integration e Continuous Deployment. Il progetto utilizza GitHub Actions per:

- Installazione dipendenze e setup dell'ambiente.
- Linting e test unitari.
- Training rapido (1 epoca) per verifica.
- Build del pacchetto Python e dell'immagine Docker.

Il Dockerfile definisce un ambiente leggero basato su **python:3.11-slim** con tutte le dipendenze necessarie. Questo garantisce riproducibilità e portabilità, permettendo di eseguire training e valutazione in un container isolato.

Conclusioni

Questo progetto mi ha permesso di sperimentare le metodologie di un sistema MLOps, Partendo dall'analisi preliminare dei dati (EDA), passando per la fase di training e di test, arrivando alla containerizzazione e alla CI/CD.

Anche se si tratta di un'esercitazione semplificata, l'architettura può essere espansa e ampliata per contesti reali più articolati. Si potrebbe, ad esempio, ampliare il progetto aggiungendo AutoML, controllo dei modelli una volta rilasciati e rilascio in cloud. Per concludere, quest'iniziativa offre un solido punto di partenza per sviluppare capacità concrete nel campo MLOps.