PRIMA LEZIONE

PyTorch, sviluppato da Facebook AI Research (FAIR), è un framework cruciale nel campo dell'Intelligenza Artificiale e del Deep Learning. Utilizzato da giganti tecnologici come OpenAI, Tesla, Amazon, Meta e Microsoft, PyTorch offre strumenti avanzati per lo sviluppo e l'implementazione di modelli complessi.

I tensori, elementi fondamentali di PyTorch, sono strutture dati che rappresentano diversi livelli di astrazione, dai semplici scalari alle matrici multidimensionali. Questi tensori sono essenziali per gestire dati complessi e eseguire operazioni matematiche avanzate in parallelo, soprattutto quando supportati da dispositivi come GPU e TPU, che accelerano significativamente il calcolo.

Nel contesto dei dispositivi hardware, le GPU, prodotte principalmente da Nvidia e AMD, sono ottimizzate per il calcolo parallelo e trovano largo impiego nel training e nell'inferenza delle reti neurali. Le TPU, sviluppate da Google, offrono una potenza eccezionale con un consumo energetico ridotto, focalizzandosi specificamente sulle operazioni I/O intensive necessarie per il Deep Learning.

La configurazione di PyTorch per sfruttare le GPU richiede l'installazione di driver aggiornati, del CUDA Toolkit e della libreria cuDNN, che ottimizza le operazioni di Deep Learning su GPU. Verificare l'installazione corretta è cruciale per garantire prestazioni ottimali durante lo sviluppo e l'esecuzione dei modelli.

In sintesi, PyTorch non solo semplifica lo sviluppo di applicazioni AI complesse, ma anche sfrutta appieno il potenziale dei tensori e dei dispositivi hardware avanzati per raggiungere risultati di alto livello nel machine learning e nel deep learning.

SEDONCA LEZIONE

Il documento intende guidare attraverso il processo di preparazione dei dati per l'uso in modelli di machine learning e deep learning, delineando una serie di passaggi cruciali.

1. **Premessa e Identificazione del Problema**: Prima di tutto, è essenziale chiarire l'obiettivo desiderato e identificare il tipo di problema da affrontare, che potrebbe essere regressione, classificazione, object detection, o altri.
2. **Raccolta dei Dati**: I dati possono provenire da varie fonti come file CSV, raccolte di immagini, dispositivi di misura, video, o sintetizzati per esperimenti. In tutti i casi, è fondamentale che i dati siano rappresentati numericamente, tipicamente come tensori.
3. **Codifica e Decodifica Numerica**: I dati in input e output per i modelli devono essere codificati e decodificati come tensori, poiché le reti neurali operano solo su dati numerici.
4. **Manipolazione dei Dati**: I tensori richiedono spesso pre-processamenti come normalizzazione, standardizzazione, bilanciamento dei campioni (oversampling, undersampling), e data augmentation per migliorare la qualità dei dati e l'efficacia dell'addestramento.
5. **Dataset e Divisione dei Campioni**: Una volta manipolati, i dati sono organizzati in dataset divisi in training, validation, e test set. Questa divisione è cruciale per addestrare, validare e testare l'efficacia del modello.
6. **Etichettatura dei Dati**: Nei dataset supervisionati, ogni campione è associato a un'etichetta che indica l'output desiderato del modello. Questa etichetta è fondamentale per l'addestramento e la valutazione del modello.
7. **Approcci di Addestramento**: A seconda del tipo di apprendimento (supervisionato o non supervisionato), il processo di addestramento varia significativamente, con il secondo che richiede alla rete di scoprire i pattern autonomamente.
8. **Utilizzo e Verifica**: Infine, i dataset vengono utilizzati per addestrare modelli di rete neurale che, una volta addestrati, vengono testati sui dati di test per verificare la loro efficacia.

Questi passaggi sono fondamentali per garantire che i dati siano pronti per l'addestramento di modelli AI e DL, assicurando al contempo che le prestazioni dei modelli siano ottimali e generalizzabili.

3.5