



PyTorch

Lasse Tristan Feldermann-Welkner,
Recommender Systems



Übersicht

1. Einführung
 - 1.1. Definition & Hintergrund
 - 1.2. Grundlegende Konzepte
2. Problematik und Theorie
3. Anwendungsfälle
4. Demo
5. Bewertung & Einordnung PyTorch
6. Diskussion/Fragen



Einführung - Definition & Hintergrund

- PyTorch ist eine Open-Source-Maschinenlernbibliothek, welche von dem Facebook-Forschungsteam für künstliche Intelligenz entwickelt wurde
 - Sie bietet eine dynamische Berechnungsumgebung für neuronale Netzwerke.
-
- PyTorch ist in Forschung und Wirtschaft weit verbreitt
 - Ermöglicht eine flexible und intuitive Implementierung von Deep Learning Modellen.



Einführung - Grundlegende Konzepte

Tensor:

- Ein Tensor ist eine multidimensionale Datenstruktur, in welcher im Kontext Recommender Systems zumeist Daten wie Nutzer- und Artikelmerkmale sowie Bewertungen
- Insbesondere zur schnellen und parallelen Verarbeitung der Daten bei umfangreichen Datensätzen sowie komplexen Modellen von Vorteil



Einführung - Grundlegende Konzepte

autograd (Automatische Differentiation)

- Die autograd-Funktionalität in PyTorch ermöglicht automatische Differentiation, was bedeutet, dass PyTorch automatisch den Gradienten von Tensoroperationen berechnet.
- Ermöglicht in fortgeschritteneren Modellen das Backpropagation Verfahren, wodurch Fehler im Modell durch das Netzwerk zurückverfolgt werden können



Einführung - Grundlegende Konzepte

Optimizer

- Ein Optimizer in PyTorch ist für die Anpassung der Modellparameter während des Trainings verantwortlich. Die Wahl des Optimierers und die Einstellung seiner Hyperparameter können einen erheblichen Einfluss auf die Konvergenz und Leistung des Modells haben.



Problematik & Theorie

Was sind Recommender Systems?

- Recommender Systems (Empfehlungssysteme) sind Softwareanwendungen, die personalisierte Empfehlungen für Produkte oder Inhalte generieren.
- Sie helfen Benutzern dabei, relevante Informationen oder Artikel zu finden, basierend auf ihren Präferenzen und dem Verhalten anderer Benutzer.



Problematik & Theorie

Warum sind sie wichtig?

- In einer Welt mit Informationsüberfluss sind Recommender Systems entscheidend, um die Benutzererfahrung zu verbessern.
- Sie erhöhen die Benutzerbindung, steigern Umsätze und ermöglichen personalisierte Inhalte.



Problematik & Theorie

- *Cold Start:*
 - Das "Cold Start"-Problem tritt auf, wenn ein neuer Benutzer oder ein neues Element (Produkt) dem System beitrifft.
 - Erläutere Strategien wie Content-basierte Empfehlungen für neue Benutzer.
- *Datenerhebung:*
 - Die Qualität und Quantität der Daten beeinflussen die Leistung von Recommender Systems.
 - Diskutiere die Herausforderungen bei der Beschaffung und Pflege relevanter Daten.
- *Datenverarbeitung:*
 - Die Verarbeitung großer Datensätze erfordert effiziente Algorithmen und Infrastruktur.
 - Betone die Bedeutung von Big Data-Technologien und skalierbaren Verarbeitungsmethoden.



Anwendungsfälle

- *1. Film- und Musikempfehlungen:*
 - Netflix verwendet PyTorch für die Verbesserung seiner Empfehlungsalgorithmen.
 - PyTorch ermöglicht die Integration komplexer Modelle für personalisierte Film- und Musikempfehlungen.
- *2. E-Commerce-Empfehlungen:*
 - Amazon nutzt PyTorch für die Optimierung von Produktempfehlungen.
 - Die Flexibilität von PyTorch ermöglicht die Integration von Deep-Learning-Modellen für personalisierte Einkaufsempfehlungen.
- *3. Forschung:*
 - Entwicklung von fortgeschrittenen neuronalen Netzwerkarchitekturen für die Bildverarbeitung
 - Generative Adversarial Networks (GANs) für Bildsynthese



Demo

<https://github.com/speedprogrammer9000/recommenderSystemsDemo>

(Link im Discord)



Bewertung und Einordnung von PyTorch

GPU-Beschleunigung:

- PyTorch unterstützt die nahtlose Nutzung von GPUs für beschleunigte Berechnungen.

Flexibilität bei der Modellanpassung:

- Forscher und Entwickler können das Modell während des Trainings anpassen, neue Schichten hinzufügen oder bestehende ändern, um verschiedene Ansätze zu testen und zu experimentieren.

Einfache Fehlerbehebung und Debugging:

- Entwickler können den Graphen schrittweise durchlaufen, um zu verstehen, wie die Modelle funktionieren, was besonders bei komplexen Recommender-Systemen von Vorteil ist.

Einfache Integration von Nicht-Neuronale-Netz-Komponenten:

- Recommender-Systeme können verschiedene Komponenten wie regelbasierte Filter, Gewichtungen oder externe Datenquellen integrieren. PyTorchs dynamische Berechnungsgraphenarchitektur erlaubt die einfache Integration solcher nicht-neuronaler Netzkomponenten in das Modell.



Bewertung und Einordnung von PyTorch

- Vorteile
 - Einfache Anwendbarkeit und große Community
 - Aktive Forschungsgemeinschaft
 - Gute Möglichkeiten für Debugging und Anpassungen
 - Effiziente Hardware Nutzung & Support für Cloudbasierte Systeme
- Nachteile
 - Weniger vorgefertigte Module als andere Frameworks (TensorFlow, Scikit-Learn)
 - Je nach individuellem Anwendungsfall gibt es möglicherweise andere Frameworks welche spezifische Probleme einfacher lösen



Fragen?



Quellen

<https://datascientest.com/de/pytorch>

<https://pytorch.org/docs/stable/>