Offline Reinforcement Learning with LLMs

N

Neel Desai

6 min read · Nov 16

Traditional Reinforcement Learning (RL) methods often require extensive interaction with the environment to learn effectively. This can be impractical or expensive in many real-world scenarios. LLMs, with their advanced few-shot learning capabilities, can extrapolate and learn effectively from limited datasets, addressing this critical constraint.

In this article, I will be summarizing a research paper which uses LLMs to address the above problem with RL methods. The paper is titled "Unleashing the Power of Pre-trained Language Models for Offline Reinforcement Learning", if you would like to read the entire paper you can find it here https://arxiv.org/abs/2201.11903.

Introduction

Imagine a computer program that learns to play a game, not by playing it directly, but by analyzing previously gathered gameplay data. This is the essence of offline reinforcement learning (RL), a fascinating area of machine learning. Unlike traditional RL, where an agent learns through direct interaction, offline RL uses pre-existing datasets, a bit like learning a sport by watching videos instead of playing. The challenge lies in dealing with situations where these datasets are sparse or limited.

Paper Summary

Open in app ↗
Search
Write
N

ingeniously leverages the capabilities of pre-trained Language Models (LMs) to enhance decision-making in situations where only pre-collected datasets are available, often characterized by limited data or sparse rewards.

Previous/Related Work

Transformers in Decision-Making

- Language to Games: Transformers have evolved from language processing to decision-making in games, notably through Decision Transformers (DT), where decision-making parallels sequence prediction in language.
- DT Enhancements: Researchers have refined DTs by incorporating various RL techniques and addressing data quality issues.

Large Language Models (LLMs)

- Versatile Abilities: LLMs are renowned for their extensive training on text data, enabling exceptional language understanding and generation. Their adaptability is showcased in few-shot and zero-shot learning.

- Fine-Tuning for Specific Tasks: LLMs are often fine-tuned to adapt to new tasks, maintaining their core capabilities while learning new skills.

LMs in Decision-Making

- Beyond Language Tasks: The application of LMs in decision-making is expanding. This involves using LMs for complex task planning and directly managing action sequences in various formats.
- Approaches and Integration: Research includes using LMs for separate planning and execution phases, and transforming diverse data types into LM-comprehensible formats.

Core Concept of LaMo

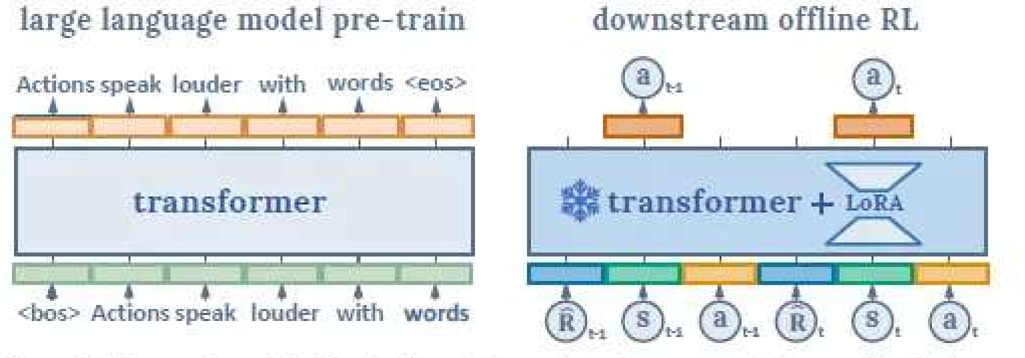
LaMo integrates Decision Transformers (DT) with advanced LMs. Decision Transformers are a type of model that approaches decision-making in sequential tasks (like playing a game) by treating it as a sequence prediction problem. LMs, on the other hand, are AI models trained on vast amounts of text data, excelling in understanding and generating human language. The fusion of these two models in LaMo aims to bring a nuanced understanding of LMs to the realm of motion control and decision-making in RL.

Key Innovations in LaMo

1. Integration of Pre-trained LMs with DT: LaMo starts with a well-established LM, GPT-2, which serves as the initial framework for the Decision Transformer. This integration is designed to transfer the rich linguistic and reasoning capabilities of LMs to the domain of decision-making in games and simulations.
2. Fine-Tuning with LoRA: LaMo employs a fine-tuning technique known as Low Rank Adaptation (LoRA), which adjusts only a small portion of the

LM's parameters. This selective fine-tuning ensures that the model retains its vast linguistic knowledge while adapting to the specific nuances of decision-making tasks.

3. Non-Linear MLPs for Embeddings: To bridge the gap between the language understanding and the specific requirements of decision-making tasks, LaMo replaces the standard linear embedding projections of LMs with more complex and capable Multi-Layer Perceptrons (MLPs). This change allows for a more nuanced processing of input data, vital for handling the complexities of RL tasks.
4. Auxiliary Language Prediction Loss: To maintain the LM's proficiency in language understanding and generation, LaMo incorporates an auxiliary language prediction task during training. This additional objective ensures that while the model learns decision-making strategies, it does not lose its inherent language capabilities.


large language model pre-train
downstream offline RL
Actions speak louder with words <eos>
a_{t-1}
a_t
transformer
transformer + LoRA
<bos> Actions speak louder with words
$\hat{R}_{t-1}$ s_{t-1} a_{t-1} $\hat{R}_t$ s_t a_t

The overview of LaMo. LaMo mainly consists of two stages: (1) pre-training LMs on language tasks, (2) freezing the pre-trained attention layers, replacing linear projections with MLPs, and using LoRA to adapt to RL tasks. We also apply the language loss during the offline RL stage as a regularizer.

Empirical Validation and Performance

The effectiveness of LaMo is rigorously tested across various tasks and environments, including MuJoCo simulations, Atari games, and a virtual Kitchen setup. The results demonstrate LaMo's superior performance in scenarios where rewards are sparse and data is limited, a common challenge in offline RL. Notably, LaMo shows significant improvements over traditional methods like CQL, IQL, TD3+BC, and even the standard Decision Transformer.

In dense-reward tasks, where data is abundant, LaMo's performance is competitive, although it does not always dominate. This illustrates the still-present strengths of traditional value-based methods in certain offline RL scenarios.

How are LLMs Used

1. Introduction to LLMs in Offline RL

- Role of LLMs: In offline reinforcement learning (RL), Large Language Models (LLMs) like GPT-2 or GPT-3 are employed to enhance decision-making capabilities. These models, originally designed for natural language processing tasks, possess a sophisticated understanding of sequences and predictions, making them valuable for sequential decision-making in RL.
- Challenges Addressed: Offline RL often struggles with limited data availability and the complexity of decision sequences. LLMs, with their advanced predictive capabilities and the ability to learn from sparse data, offer solutions to these challenges.

2. Integrating LLMs with Decision Transformers (DTs)

- Combination Strategy: The LaMo framework innovatively combines Decision Transformers with LLMs. DTs, which transform decision-making into a sequence prediction problem, are enhanced by the powerful sequence modeling capabilities of LLMs.

- Data Processing Adaptations: To effectively integrate LLMs, LaMo employs non-linear Multi-Layer Perceptrons (MLPs) instead of simpler linear projections. This modification allows for a more nuanced translation of RL tasks into a format understandable by the LLMs.

3. Fine-Tuning LLMs with Low-Rank Adapters (LoRA)

- Fine-Tuning Approach: Instead of fully retraining the LLMs for the RL context, LaMo uses LoRA, a method that adjusts only a small portion of the model's parameters. This targeted fine-tuning maintains the general language understanding capabilities of the LLM while adapting it for specific decision-making tasks.
- Balancing Generalization and Specialization: This approach ensures that the LLMs retain their broad language capabilities (useful for understanding diverse scenarios) while becoming specialized in the intricacies of specific RL tasks.

4. Incorporating Auxiliary Language Prediction Loss

- Dual Training Objective: LaMo employs an auxiliary language prediction loss during training. This means that while the model learns decision-making strategies, it also continues to train on language tasks.
- Preserving Language Abilities: This strategy ensures that the LLMs do not lose their language proficiency while being adapted to offline RL tasks. It also helps in preventing overfitting to specific RL scenarios.

5. Advantages of Using LLMs in Offline RL

- Enhanced Few-Shot Learning: LLMs are known for their few-shot learning capabilities. In offline RL, this translates to superior performance even with limited datasets, a significant advantage over traditional RL methods.

- Sequential Reasoning Power: The inherent ability of LLMs to understand and predict sequences lends itself well to decision-making in RL, especially in tasks where decisions are interdependent and complex.

Conclusion

LaMo stands out as a pioneering framework that effectively marries the advanced language understanding abilities of LMs with the strategic decision-making required in offline RL. Its prowess in sparse-reward tasks and exceptional performance in low-data scenarios underscore the potential of LMs in enhancing offline RL methodologies.

While LaMo marks a significant advancement, it's not without limitations. Its efficacy in dense-reward tasks, although commendable, is not always dominant. Moreover, the auxiliary language prediction loss, while instrumental in certain scenarios, doesn't universally boost performance across all tasks. This opens avenues for future research, particularly in leveraging language reasoning skills more effectively in offline RL and exploring the integration of even larger LMs.

In essence, LaMo paves the way for a more nuanced and data-efficient approach to offline RL, harnessing the power of language understanding to navigate the complex world of decision-making in virtual environments.

Written by Neel Desai

Edit profile

2 Followers

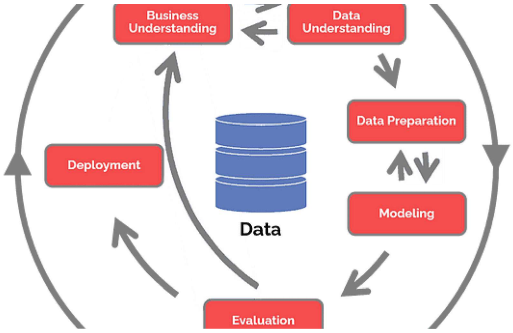
More from Neel Desai

Neel Desai

Using GPT4 to learn EDA, Data Cleaning and Data Preparation

Introduction

11 min read · Oct 27

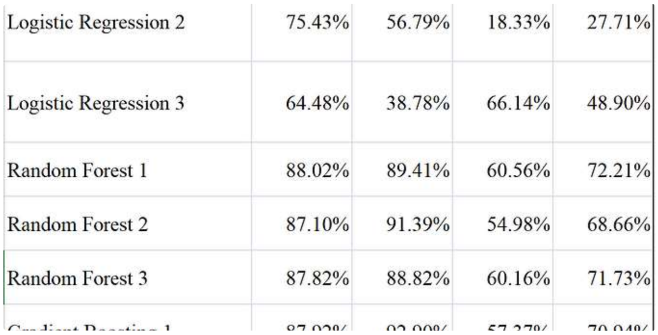


Neel Desai

Predicting Heart Attack Risk: A CRISP-DM Approach

We have a dataset which contains various health markers and the likelihood of the…

7 min read · Sep 22



Logistic Regression 2	75.43%	56.79%	18.33%	27.71%
Logistic Regression 3	64.48%	38.78%	66.14%	48.90%
Random Forest 1	88.02%	89.41%	60.56%	72.21%
Random Forest 2	87.10%	91.39%	54.98%	68.66%
Random Forest 3	87.82%	88.82%	60.16%	71.73%
Gradient Boosting 1	87.02%	92.90%	57.37%	70.94%

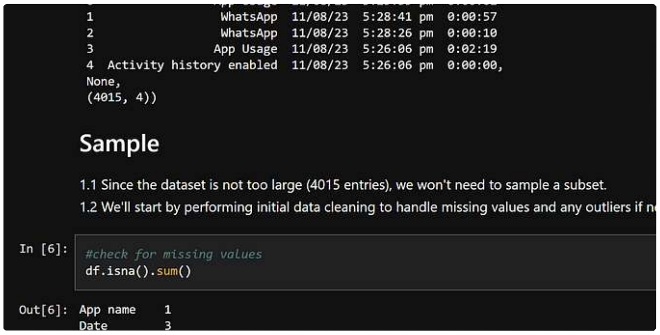
Neel Desai

Collaborative Analysis with ChatGPT: A Machine Learning…

Problem Statement

9 min read · Aug 30

8



```
1                  WhatsApp  11/08/23  5:28:41 pm  0:00:57
2                  WhatsApp  11/08/23  5:28:26 pm  0:00:10
3                 App Usage  11/08/23  5:26:06 pm  0:02:19
4  Activity history enabled  11/08/23  5:26:06 pm  0:00:00,
None,
(4015, 4))
```

Sample

1.1 Since the dataset is not too large (4015 entries), we won't need to sample a subset.

1.2 We'll start by performing initial data cleaning to handle missing values and any outliers if ne

```
In [6]:  #check for missing values
         df.isna().sum()

Out[6]:  App name    1
         Date        3
```

Neel Desai

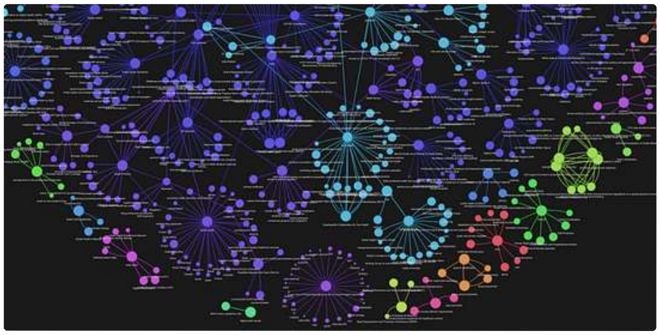
Unlocking App Usage Patterns: An Association Rule Mining Approac…

Dataset: Phone Usage Dataset

6 min read · Sep 22

See all from Neel Desai

Recommended from Medium



Rahul Nayak in Towards Data Science

How to Convert Any Text Into a Graph of Concepts

A method to convert any text corpus into a Knowledge Graph using Mistral 7B.

12 min read · Nov 9

2.8K 37

Gavin Li

Unbelievable! Run 70B LLM Inference on a Single 4GB GPU wi...

Large language models require huge amounts of GPU memory. Is it possible to ru...

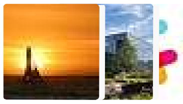
6 min read · Nov 18

728 13

Lists

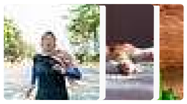
Staff Picks

516 stories · 468 saves



Stories to Help You Level-Up at Work

19 stories · 322 saves

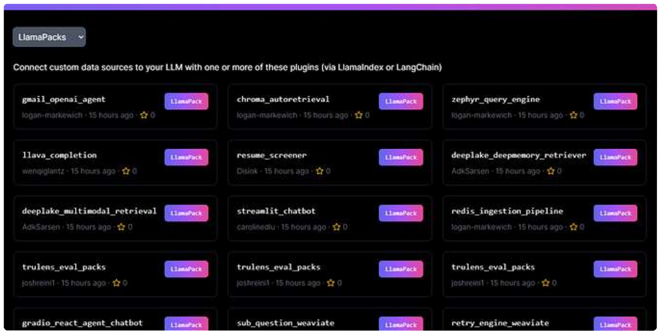


Self-Improvement 101

20 stories · 947 saves

Productivity 101

20 stories · 864 saves



LlamaPacks

Connect custom data sources to your LLM with one or more of these plugins (via LlamaIndex or LangChain)

gmail_openai_agent LlamaPack
logan-markewich · 15 hours ago · ☆ 0

chroma_autoretrieval LlamaPack
logan-markewich · 15 hours ago · ☆ 0

zephyr_query_engine LlamaPack
logan-markewich · 15 hours ago · ☆ 0

llava_completion LlamaPack
wenqiglantz · 15 hours ago · ☆ 0

resume_screener LlamaPack
Disiok · 15 hours ago · ☆ 0

deeplake_deepmemory_retriever LlamaPack
AdkSarsen · 15 hours ago · ☆ 0

deeplake_multimodal_retrieval LlamaPack
AdkSarsen · 15 hours ago · ☆ 0

streamlit_chatbot LlamaPack
carolinedlu · 15 hours ago · ☆ 0

redis_ingestion_pipeline LlamaPack
logan-markewich · 15 hours ago · ☆ 0

trulens_eval_packs LlamaPack
joshreini1 · 15 hours ago · ☆ 0

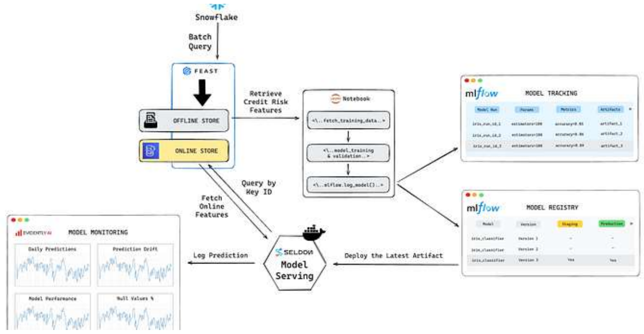
trulens_eval_packs LlamaPack
joshreini1 · 15 hours ago · ☆ 0

trulens_eval_packs LlamaPack
joshreini1 · 15 hours ago · ☆ 0

gradio_react_agent_chatbot LlamaPack

sub_question_weaviate LlamaPack

retry_engine_weaviate LlamaPack



Jerry Liu in LlamaIndex Blog

Introducing Llama Packs

Today we're excited to introduce Llama Packs 🦙📦 — a community-driven hub of...

4 min read · 5 days ago

Qwak

Building an End-to-End MLOps Pipeline with Open-Source Tools

MLOps Open Source: TL;DR

11 min read · Oct 18

84 1

Mahesh

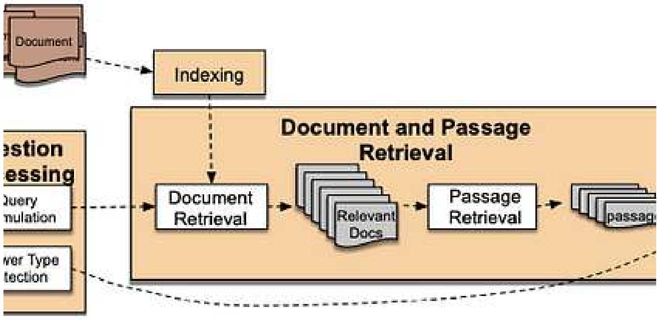
How to make a Recommender System Chatbot with LLMs

Make a session based apparel recommender system chatbot based on open source large...

13 min read · Nov 13

97 3

371 3



Krishna Yo... in Artificial Intelligence in Plain Engli...

Building a question-answering system using LLM on your private...

11 min read · Oct 5

337 6

See more recommendations