## Über den Test

07.12.2023 | 13:20 – 13:50 (30min)

* TUWEL
* Multiple Choice
* Keine Teilpunkte
* Closed Booked

## Inhalt

1. Birds Eye View of the Architecture of a Large Language Model
2. Overview of the purpose of the different layers (till MHA)
3. Overview of how the layers transform their input (till MHA)
4. High-Level Overview of the training steps from nothing to RLHF
5. Basics of RL and PPO

## Birds Eye View of the Architecture of a Large Language Model

**Begriffe**

Transformer (Art von LLM welche wir benutzt haben):

🡪ein deep neural network für Text

🡪Attention-Mechanismus (besonders Multi-Head Attention – Fokus auf verschiedene Teile)
Positional Encoding 🡪 damit achtet es auf die Reihenfolge der Daten

## Overview of the purpose of the different layers (till MHA)

**Überblick - LLM**



1. **Inputs**: Eingabe von klarem Text zur Erkennung vom Kontext
2. **Input Embedding**: Der klare Text wir embedded / umgeformt in Zahlenwerte, um die Charaktere in Reihen von den Charakteren (also Worten) zu erkennen
3. **Positional Encoding**: Codiert die Zahlenwerte (von den Embeddings) und beachtet die Position der erkannten Sequenzen / also Worten

## Overview of how the layers transform their input (till MHA)

**Überblick**

1. Inputs / Klare Text Eingabe

Einfach der klare Text vom User. Er wird nicht transformiert oder geändert.

1. Tokenization / Encoding / Embedding

**Tokenizer:** Erstelllung einer Art Worterbuch: Jeder Charakter und Zeichen kriegt eine Zahl zugeordnet.

**Embedding:** konvergiert die Token vom Tokenizer in Vektoren 🡪 ähnliche Zusammensetzungen sind dadurch leichter zu erkennen. Hier erkennt das LLM die Bedeutung und Beziehung zwischen ganzen Wörten

1. Positional Encoding

Berücksichtig die Reihenfolge der Wörte innerhalb des Satzes. So wird der Kontext besser verstanden.

## High-Level Overview of the training steps from nothing to RLHF

**Begriffe**

RLHF: reinforcement learning for human feedback (Teil von PPO)

**Übersicht: Training Schritte**



(Inhalt der Blasen nicht wichtig, nur die Struktur des Prozesses)

* Training (Pre-Training): das Modell wird mit einen großen Datensatz trainiert
* Optimierung durch human feedback (SFT): Model wird „feinjustiert“ durch Beispiele von menschlichen Antworten
* Reinforcement Learning (RLHF): basierend auf diese „Feinjustierung“ wird das Model weiter trainiert
* Wiederholung /: Die drei oben genannten Schritte können wiederholt werden um Präferenzen für die Antworten zu setzten.

## Basics of RL and PPO

**Übersicht PPO**

1. **Agent**: interagiert mit der Umgebung (das LLM selbst)
2. **Policy/Critic**: von agent benutzt, um Entscheidungen zu treffen (Regelwerk)
3. **Judge/Baseline**: vergibt Punkte basierend auf die Interaktionen des Agenten
4. **(Ziel)**: die korrekten Parameter für das Regelwerk/Policy finden, um die meisten Punkte zu bekommen

**Begriffe (objective function)**

* πθ: Ist die Policy des agents.
* **πθ(at | st)**: Dies ist die stochastische Policy, die die Wahrscheinlichkeit angibt, mit der die Aktion at im Zustand st unter der Policy πθ gewählt wird.
* **πθold (at | st)**: Dies ist die Policy vor dem Update, bewertet an der Aktion at und dem Zustand st.
* ˆg: Dies ist der geschätzte Gradient der Policy, der für das Gradientenanstiegsverfahren verwendet wird.
* ˆEt[. . .]: Dies repräsentiert den empirischen Durchschnitt über eine endliche Anzahl von Stichproben.
* ˆAt: Dies ist ein Schätzer für die Advantage-Funktion zum Zeitpunkt t, der angibt, wie gut es ist, die Aktion at im Zustand st zu wählen.
* δ: Dies ist ein Schwellenwert, der die Größe der Policy-Änderung begrenzt.
* β: Dies ist ein Koeffizient, der in der Theorie hinter TRPO verwendet wird, um eine Strafe statt einer Beschränkung einzuführen, um die Optimierung zu vereinfachen.
* **∇θ log πθ(at | st)**: Dies ist der Gradient des Logarithmus der Policy πθ, bewertet an der Aktion at und dem Zustand st, bezüglich der Parameter θ.
* LP G(θ): Dies ist die Zielfunktion, deren Gradient der geschätzte Policy-Gradient ist.
* KL[**πθold (· | st), πθ(· | st)**]: Dies ist die Kullback-Leibler-Divergenz, was ein Maß für die Änderung der Policy ist.
* Dot-Product: Es erzeugt den attantion score & besagt, wie stark Tokens/Buchstaben in Sequenzen/Wörter
* Attention-Score: Der Wert über die Wichtigkeit eines Tokens in einer Sequenz

## Beispielsfragen



