⚡️🚀המאמר היומי של מייק -25.10.24: ⚡️🚀  
Amortized Planning with Large-Scale Transformers: A Case Study on Chess

מאמר די מעניין שגרם לדיונים רבים בנושא יכולות ריזונינג של מודלי שפה. אחרי שהעניינים קצת נרגעו הגעתי לסקורו בלי להתייחס יותר מדי לסוגיה הזו. המאמר למעשה אימן מודל שפה די צנוע מבחינת פרמטרים (עם הטרנספורמרים בפנים) לשחק שח. אזכיר שהמכונות הגיעו לרמת של בני אנוש בשחמט די מזמן (לדעתי לפני 30 שנה כאשר deep blue השאיר אבק לאלוף העולם דאז גארי קספרוב).

אז מה המחברים עשו בעצם? הם הורדו 10 מיליון משחק שחמט מאתר LiChess והשתמשו בכלי הנקרא StockFish לשערוך הסתברות ניצחון עבור מצב לוח נתון s. לאחר מכן הם הפכו את מצב הלוח ותיאור המהלך לטקסט (נראה די טבעי בסך הכל) ואימנו מודל שפה ״לשחק שח״. המחברים ניסו לעשות זאת בכמה דרכים:

אימנו את המודל לחזות את הסיכוי לניצחון בהינתן מצב הלוח s ומהלך a. כדי לעשות זאת הם חילקו סיכויי הניצחון לכמה בינים (זרים) ואימנו את המודל לחזות את הבין שבו נמצא הסיכוי ה-ground-truth. הם עשו את זה לא בצורה הרגילה (עם one-hot encoding של כל בין) אלא על ידי ״ריכוכו״ כלומר כל בין מקבל הסתברות משלו כאשר הבין ה-GD מקבל את ההסתברות הכי גבוה (נעשה לפי התפלגות גאוס ונקרא HL-Gauss)

אימנו את המודל את סיכוי הניצחון עבור מצב לוח נתון s באותה הצורה כמו ב 1.

אימון מודל לחזות את המהלך ה-GD של המשחק

בסוף המהלך נבחר כזה עם סיכוי לניצחון הגבוה ביותר. ויש תוצאות לא רעות.

האם זה מצביע על כך שהמודלים יודעים לעשות ריזונינג - לא יודע, מבטיח לחשוב על זה לעומק….

https://arxiv.org/pdf/2402.04494v2