המאמר היומי של מייק: 08.05.25  
Memory Layers at Scale

המאמר הזה משך את עיניי כי מופיעה בו המילה ״memory" בהקשר מודלי שפה. כבר היום כשאתם מדברים עם ChatGPT, קלוד ומודלים אחרים אתם לא מדברים רק עם מודל שפה אלא עם מערכת שלמה הכוללת בעצמה שכבות של זיכרון (למשל ממומשים כראג(Retrieval Augmented Generation) או קאשים). המאמר מציע שכבה לרשת נוירונים שהיא מנגנון של זכרון שניתן לשמור בה וגם לאחזר ממנה בהתאם לשאילתה.

למעשה שכבת זכרון זו די דומה לבלוק טרנספורמר אבל להבדיל ממנו אנו מאחזרים ממנו רק מה שרלוונטי לשאילתה המיוצגת על ידי וקטור q. כלומר במקום לשלב את כל הוקטורים האפשריים שיש לנו בזכרון אנו בוחרים k אלו שהם ה״מתאימים ביותר לוקטור השאילתה q". המנגנון הזה קצת דומה למנגנון MoE (שזה Mixture of Experts) כאשר אנו בוחרים להפעלים תת-מטריצות של שכבות FFN. ההבדל בין השיטה המוצעת ל-MoE נעוץ בעובדה כי ב-MoE אני המומחים (experts) הם קבועים (תת-מטריצות מוגדרות מראש של שכבת ה-FFN המלאה) וכאן ניתן לבחור כל שילוב של עמודות של שכבת ה-FFN.

לוקטור שאילתה נתונה q אנו בוחרים את הוקטורים הקרובים אליו ביותר מהזיכרון. K וקטורים בעלי ציון דמיון הגבוה ביותר נבחרים, משולבים עם מטריצת ערכים V (המאמר לא מפרט איך בדיוק ועקב גם אני ״חושד״ במכפלה רגילה). לאחר מכן מכפילים את את התוצאה במכפלה של q במטריצה נלמדת W1 שמוכפלת בפלט של מנגנון ה-attention שנמצא לפני בלוק המאחזר מהזכרון, שעליו מופעלת אקטיבציית silu (שהפכה להיות מאוד פופולרית לאחרונה). לאחר מכן מכפילים את התוצאה במטריצה נלמדת W2.

מכיוון שאנו רוצים לשמור הרבה מאוד וקטורים בזכרון המכפלות שלהם עם q עלולים להיות כבדים מבחינה חישובית. כמו שמקובל הים המחברים ״מחלקים את הזכרון״ בין כמה gpus ואז בודקים את הדמיון בכל אחד מהם בנפרד ואז משלבים את התוצאות כדי לבחור את וקטורי הזכרון הדומים ביותר. כמובן שהמטריצות בכל gpu כמובן קטנות יותר ממטריצת הזכרון הגדולה וגם וקטור q מחולק לכמה תת-וקטורים בין ה-gpus.

שכבה זו יכולה להיות משובלת עם בלוקי טרנספורמרים במודלי שפה אבל אני גם לא רואה שום בעיה לשלבם עם שכבות אחרות כמו ממבה. מאמר נחמד וקליל (על הדרך גיליתי שיטה מעניינת לאחזור יעיל מהזיכרון המבוזר על כמה gpus).

https://arxiv.org/abs/2412.09764