המאמר היומי של מייק - 19.03.25  
EFFICIENTLY LEARNING AT TEST-TIME: ACTIVE FINE-TUNING OF LLMS

בתקופה האחרונה השיטה הכי פופולרית להתאמת מודלי שפה למשימה ספציפית היא למידה in-context או ICL. בגדול אנו מספקים למודל, בתוך הפרומפט, כמה דוגמאות לביצוע משימה והמודל ״לומד״ איך לבצע אותה ללא שום שינוי במשקליו. ICL מתאפשר עקב האופי האדפטיבי של הטרנספורמרים (מנגנון ה-attention בתוכו) המצליחים ״לעדכן את אופן החישוב שלו״ כפונקציה של קלט.

המאמר דן בשיטה אחרת לאדפטציה של מודל למשימה נתונה בזמן טסט(המאמר קצת מערבב את המושג של טסט ואינפרנס) המערב fine-tune קליל של המודל על סמך הפרומפט שמוזן אליו. להבדיל מ-ICL השיטה המוצעת (SIFT(Selects Informative data for Fine-Tuning כן משנה את משקלי המודל (מבצעת צעד אחד של מורד הגרדיאנט - gradient descent). למעשה SIFT (ד״א יש שיטה בשם כזה גם בעיבוד תמונה מהעידן לפני הרשתות) מציעה שיטה לבחירה של דוגמאות מהדאטהסט לפיין טיון של מודל עבור פרומפט נתון.

המחברים טוענים שבחירת דוגמאות הכי קרובות לפרומפט במרחב הלטנטי מבחינת מרחק קוסיין או מכפלה פנימית(nearest neighbors or NN) היא תת-אופטימלית ועלולה להביא דוגמאות מיותרות הפוגעות בביצועי פיין טיון. במקום לשלוף דוגמאות הדומות ביותר לפרומפט, SIFT בוחרת את אלו שמספקות את מירב המידע החדש, וכך משיגה התאמה טובה יותר של המודל עם מינימום חישובים נוספים.

הגישה המוצעת מערבת שיעורך רמת אי ודאות של תשובת המודל בהינתן הדוגמאות שבחרנו ל-FT (לאחר FT הכוונה). בפרק הבא אסביר למה זה חשוב בעצם.

הערכת אי-וודאות להנחיית FT ולמה זה בכלל חשוב כאן?

שיטות FT רבות מסתמכות על שליפת דוגמאות דומות בהתבסס על דמיון קוסיין או מרחק אוקלידי. אך גישה זו לוקה בחסר: היא אינה מבדילה בין דאטה רלוונטי לזה שמיותר. שתי דוגמאות דומות מאוד עשויות להכיל את אותו מידע, ולכן אחת מהן אינה תורמת לתוצאת FT. כדי לפתור זאת, המחברים מציעים שיטה להערכת אי-הוודאות של המודל בתשובתו לאחר FT . אם המודל בטוח מאוד בתשובתו אחרי FT, הוספת דוגמא לא תשפיע משמעותית. אך אם אי-הוודאות גבוהה, בחירה חכמה של דוגמאות יכולה לשפר את ביצועי המודל משמעותית ןהאתגר הוא למצוא את הדוגמאות הללו ביעילות.

מדידת דמיון במרחב הסמוי בעזרת פונקציית קרנל

כאמור הבסיס לשיטת הבחירה של SIFT הוא מדידת הדמיון בין דוגמאות במרחב לטנטי. כדי לכמת את הדמיון הזה, המחברים משתמשים בפונקציית קרנל - שהיא מוגדרת בתור מכפלה פנימית בין הייצוגים הלטנטיים של הדוגמאות. פונקציה זו מקבלת שני רצפים ומחזירה ציון דמיון—גבוה עבור סדרות דומות ונמוך עבור רצפים שונים. בעזרת פונקצית קרנל זו בונים מטריצה קרנל עבור הדוגמאות שנבחרו ל- FT והפרומפט עצמו. לאחר מכן מגדירי מודל דמה (surrogate model) שמטרו לשערך את ביצועי ה-LLM לאחר FT על הדוגמאות שנבחרו.

באמצעות מודל זה בונים (זה קצת כבד מתמטית) את השיערוך של אי וודאות של המודל אחרי הוספה של דוגמא x מהדאטהסט לסט הדוגמאות שעליהם יתבצע הטיוב. בסופו של דבר בוחרים דוגמא הממזערת את אי ודאות עבור הפרומפט ומוסיפים אותה לסט הדוגמאות זה.

במילים פשוטות הגישה המוצעות מאזנת בין שני שיקולים מנוגדים:

רלוונטיות: הדוגמאות הנבחרות צריכות להיות עדיין רלוונטיות לפרומפט.

גיוון: הדוגמאות אינן אמורות להכיל מידע חופף ומיותר.

במקום לבחור דוגמאות בבת אחת, SIFT בוחר כל דוגמה באופן הדרגתי, תוך שימוש בפונקציית קרנל כדי לקבוע את הערך המוסף שלה.

אם מועמד חדש דומה מדי לדוגמאות שנבחרו בעבר, הוא נדחה, מכיוון שהוא אינו מוסיף מידע חדש.

אם המועמד רלוונטי אך מכיל פרטים חדשים, הוא נבחר כדי להפחית את אי-הוודאות.

אם המועמד אינו קשור לפרומפט כלל, הוא נשאר מחוץ לתהליך.

https://arxiv.org/abs/2410.08020