⚡️🚀המאמר היומי של מייק 05-06.09.24: ⚡️🚀

EAGLE: Speculative Sampling Requires Rethinking Feature Uncertainty

EAGLE-2: Faster Inference of Language Models with Dynamic Draft Trees

חדי זכרון מביניכם אולי שמו לב כי לא פרסמתי סקירה יומית אתמול (בד״כ אני לא מפרסם סקירה בימי ראשון שבהם אני מקליט פודקאסט). אתמול היה לא חמישי ולא פרסמתי סקירה כי הכנתי לכם סקירה כפולה להיום. היום נסקור שני מאמרים שהשני מהם הוא שכלול של הראשון.

שני המאמר הם בנושא של speculative decoding או SpDe (הקיצור הומצא על ידיי). SpDe זו דרך להאיץ דגימה (גנרוט טקסט) ממודלי שפה. כמו שאתם זוכרים הגנרוט ממודלי שפה מתבצע באופן אוטורגרסיבי כמו טוקן לאחר טוקן. כמובן שזה יכול להיות איטי בטח עבור מודלים עצומים בעלי מאות מיליארדי פרמטרים.

האם ניתן להאיץ את תהליך הגנרוט - התשובה היא כן ובשנתיים האחרונות נעשה מחקר מאוד רציני בנושא והוצעו מספר שיטות שבאמצעותם ניתן להגיע לקצב דגימה גבוה יותר. SpDe היא משפחת שיטות להאצת קצב גנרוט באמצעות שימוש במודל קל (וחלש יותר) בנוסף למודל היעד (שאותו אנחנו מעוניינים להאיץ כאמור).

שיטות SpDe מבוססות על אובזרבציה כי מהלך הגנרוט טוקן אחרי טוקן צוואר בקבוק הוא העברת הדאטה מהזיכרון SRAM המהיר (אך קטן) של יחידת החישוב של GPU לבין זיכרון DRAM הגדול אך איטי יותר. ולא החישוב עצמו. נובע מכך שניתן לנצל את צוואר הבקבוק הזה ולבצע יותר חישובים (עבור יותר טוקנים) בזמן שהדאטה מטייל בין SRAM ל-DRAM.

הבעיה לממש את הגישה הזו בצורה נאיבית נובעת מאופן אוטורגרסיבי של הגנרוט ממודל שפה שלא מאפשר לבצע את החישובים עבור חיזוי של יותר טוקנים באותו הזמן. שיטות SpDe עוקפות את המכשול הזה על ידי הוספת מודל קטן ומהיר יותר שיאפשר למודל הגדול לחזות כמה טוקנים באותו הזמן.

איך זה עובד? אנו חוזים כמה טוקנים עם המודל הקטן L\_s ולאחר מכן ״מתקנים״ את החיזוי עבור הטוקנים שנדגמו עם המודל הגדול כאשר ה״תיקון״ עבור כל הטוקנים שנחזו על יד L\_s מתבצע באותו הזמן. כלומר L\_s חוזה הסתברויות עבור k טוקנים רצופים (בהינתן הטקסט שכבר גונרט), הם מוזנים (יחד עם הקשר) למודל הגדול L\_b והוא חוזה את ההסתברויות p\_i עבור k טוקנים אלו באותו הזמן.

לאחר מהם מבצעים משהו דומה למה שעושים ב-rejection sampling ו״מתקנים״ הסתבריות אלו כך שיתאימו להתפלגות של המודל הגדול L\_l. לאחר מכן יש שלב של rejection שבו אנו מחליטים האם אנחנו מקבלים או לא מקבלים את הטוקנים שנדגמו על יד המודל הקטן L\_s. זה מתבצע טוקן טוקן (חישוב מהיר מאוד) וכל הטוקנים שבאים לפני הטוקן הראשון t שקיבל reject מתקבלים (נכנסים לטקסט המגונרט) ואיטרציית דגימה חדשה מתחילה מ-t.

ניתן להוכיח שדגימה כזו היא בעלת אותה ההתפלגות של הטוקנים כמו מודל היעד L\_l. יש כאן כמה שאלות חשובות על איך לבחור מודל קטן L\_s, בין כמה טוקנים לדגום איתו כל פעם במטרה למקסם את קצב הדגימה. עכשיו השאלה האם ניתן לבחור מודל L\_s כך שהוא גם יהיה מהיר מאוד וגם איכותי מספיק כך שמספר הטוקנים שנדגמו איתו יתקבלו לרוב על ידי L\_l. מודל כזה עתיד להגביר את מהירות הדגימה האפקטיבית מ- L\_l.

זה בדיוק מה שהמאמר Eagle מציע. הוא מציע לקחת מודל קטן ולאמן אותו להיות L\_s, כלומר לתפור אותו למשימה שהוא מיועד. בשביל זה עבור מודל L\_l נתון לוקחים מודל טרנספורמר רדוד ומאמנים אותו לחזות לא רק את הטוקן הבא (ההסתברות שלו) אלא גם הייצוג בשכבה אחרונה של L\_l לפני שכבת החיזוי (כלומר יש כאן בעיית רגרסיה). החיזוי מתבצע בהינתן הטוקנים הקודמים (הייצוגים שלהם) וגם הייצוג מהשכבה האחרונה של הטוקנים הקודמים. מכיוון שהמודל L\_s הוא קטן ומהיר אנו חוזים איתו כמה סדרות טוקנים (עבור הקשר נתון) על ידי בחירה של כמה טוקנים(אך מספר קבוע כל פעם, נגיד 3) בעלי הסתברות הגבוה ביותר כל פעם. כלומר להקשר נתון חוזים כמה המשכים עבורו - בונים סוג של עץ חיזוי עבור הטוקנים. ככה יותר טוקנים רצופים עשויים לא לקבל reject מ- L\_l אחר כך.

לאחר שאימון L\_s הסתיים לוקחים אותו ומבצעים אינפרנס בצורה די דומה ל-SpDe עם שכלול של מנגונן ה-reject.

מה בעצם Eagle2 משכלל את מנגנון בניית עץ החיזוים על ידי L\_s על ידי בחירת סדרות בעלי הסתברות כוללת מקסימלית. סדרות השונות עם Eagle2 יכולות להיות בעלות אורך שונה כמובן (הכל מסתמך על ההסתברות הכוללת של הסדרה). ככה נוצרות סדרות בעלות פוטנציאל גבוה יותר להתקבל על ידי L\_l.

היה ארוך - מקווה שלא איבדתי אותכם….

https://arxiv.org/pdf/2401.15077

https://arxiv.org/pdf/2406.16858