המאמר היומי של מייק: 10.05.25  
EfficientQAT: Efficient Quantization-Aware Training for Large Language Models

אימון מודע לקווינטוט (Quantization-Aware Training, או QAT) הוא טכניקה שבה המודל לומד כבר בזמן האימון להתמודד עם מגבלות הקווינטוט שיופעלו עליו בזמן ריצה. מגבלות אלו מתבטאים בחישובים בדיוק נמוך יותר (למשל INT8 במקום FP32). עם QAT המודל מאומן תוך חיקוי של תהליך הקווינטוט, כך שבכל שלב באימון מדמים חישובים המדמים את עיבוד דאטה בדיוק מופחת. במהלך האימון שומרים על ייצוג מדויק לצורך חישוב גרדיאנטים, אך מוסיפים "הפרעה מבוקרת" בצורת קווינטוט קדימה ואחורה (quantization & dequantization) כדי לדמות את ההתנהגות של המודל לאחר ההפחתה בדיוק. כך, המשקלים והאקטיבציות מתאימים את עצמם באופן הדרגתי כדי להיות עמידים לשגיאות קווינטוט.

בניגוד לקווינטוט לאחר אימון (PTQ), אשר מתבצעת ללא התאמה של פרמטרי המודל, QAT מאפשר שמירה על ביצועים קרובים יותר למודל המקורי גם לאחר המעבר לייצוג מקוונטט. לרוב, משתמשים ב־"fake quantization" כדי לבצע כימות מדומה כחלק מגרף החישוב של המודל, תוך כדי שמירה על רזולוציה גבוהה לחישובי הגרדיאנטים. השיטה מאפשרת לפרוס מודלים על חומרה חסכונית כמו שבבים ניידים ו־Edge, מבלי לוותר על דיוק תחזיות.

דימוי קווינטוט (וגם dequantization) ב-QAT מתבצע באמצעות שני פרמטרים מאומנים עיקריים: האפס z (של הייצוג המקוונטט) וגורם סקיילינג s עבור יעד קווינטוט נתון (נגיד 8 ביט). אז המאמר מציע שני חידושים עיקריים. הראשון הוא אימון של s ו-z לכל בלוק טרנספורמר בנפרד (יחד עם משקליו). כלומר מתחילים מהבלוק הראשון מאמנים אותו יחד עם s ו-z שלו, מקפיאים אותם (s ו-z) וממשיכים ל- s ו-z של בלוק הבא. ד״א ניתן לאמן s ו-z שונים עבור השכבות השונות של בלוק הטרנספורמר (attention, FFN למשל).

החידוש השני הוא אימון מלא של כל הבלוקים יחד אחרי שאימנו אותם בנפרד בשלב הראשון. במהלך השלב השני z נותר קבוע ורק גורם הסקיילינג s מאומן.

זהו זה - ויש טענות לשיפור ביצועים כמובן…

https://arxiv.org/abs/2407.11062