⚡️🚀המאמר היומי של מייק 14.08.24: ⚡️🚀

Jumping Ahead: Improving Reconstruction Fidelity with JumpReLU Sparse Autoencoders

אתמול סקרנו מאמר שהשתמש בגישת SAE או Sparse AutoEncoders כדי לחדור ל״מחשובותיו״ של מודל שפה גדול דרך האקטיבציות של הנוירונים שלהם. הנחת היסוד במאמר היתה כי נוירונים ״מגיבים״ לכמה קונספטים שונים וניתן לאמן SAE רדוד מאוד (שכבה אחת בדקודר ושכבה אחת באנדוקר) כדי להגיע לוקטור דליל המקודד (נדלק) קונספט אחד בלבד כלומר disentanglement של הפיצ'רים לנוירונים ייעודיים.

כמאמר יש באנקודר של SAE שכבה לינארית אחת עם פונקציית אקטיבציה הנקראת JumpReLU שראיתי אותה בפעם הראשונה במאמר הזה. פונקציה הזו היא בעצם הזזה של ReLU בציר X ובציר y בפרמטר t נלמד (במאמר זה נקרא טטה). הטענה במאמר שזה מאפשר ללמוד את הייצוג הדליל של דאטה על ידי האנקודר יותר טוב של פונקציית ReLU בגלל שהוא מאפשר לאפס את הקטיבציות בצורה ״נלמדת יותר מ-ReLU".

עכשיו נשאלת השאלה איך אנחנו אוכפים דלילות על ייצוג הדאטה (אחרי האנקודר). בעבודות קודמות השתמשו ב-L1 בשביל כך אך כאן המחברים משתמשים באותה JumpReLU כדי להפוך את איפוס האיברים בייצוג יותר נלמד. ושימו לב ש- JumpReLU בא עם פרמטר נלמד הזה לזה של האנקודר עצמו שזה עוזר לאכוף דלילות על הייצוג.

יש עוד טריק אחד קטן ולא מאוד מהותי במאמר הנקרא Kernel density estimation או KDE. אם אתם זוכרים KDE עוזר לנו לשערך(כלומר לקרב) פונקצית צפיפות בהינתם דאטהסט של נקודות באמצעות פונקציית קרנל. פונקציית קרנל יכולה להיות גאוסית למשל ומטרתה לשערך את פונקציית הצפיפות לנקודות לא ידועות על ידי קירובה בין הנקודות בדאטהסט (בדומה לספליין). אז המחברים משתמשים בטריק הזה כדי לשערך את JumpReLU בנקודה t שבה היא לא גזירה.

מאמר נחמד בנושא די חשוב שאמשיך לסקור כנראה גם בעתיד…

https://arxiv.org/pdf/2407.14435