# Review 107: LoraHub: Efficient Cross-Task Generalization via Dynamic LoRA Composition 26.07.23 https://huggingface.co/papers/2307.13269

**Paper: https://arxiv.org/abs/2307.13269v3**

האם יצא לכם לכייל מודל שפה לכמה משימות בו זמנית מכמה דוגמאות נתונות לכל משימה? כלומר אתם כן רוצים לשנות את משקלי המודל אבל לא רוצים להשתמש ב-gradient descent בשביל לעשות זאת כי אין ברשותכם מעבד חזק. היום ב-#shorthebrewpapereviews מאמר שמציע שיטה לעשות את זה.  
  
המחברים משתמשים בכוכב העולה בעולם כיול #llms הנקרא LoRa. זו שיטה לכיול #llm שבמקום לאמן את המשקלים עצמם מואמנת למצוא את התוספות האופטימליות למשקלים (=מטריצות של דלטאות). בנוסף על כל תוספת מניחים שהיא בעלך רנק נמוך וניתן לתאר אותה כמכפלה של שתי מטריצות A ו- B בעלות רנק נמוך גם כן שמאפשר להקטין משמעותית את מספר המשקלים המכוילים ובכך להפוך אותו ליותר יעיל וקצר.   
  
אז בואו נניח של שיש לנו מטריצות A ו-B לכל משימה שאנו רוצים לכייל את מודל השפה והמטרה שלנו למצוא של מטריצת הדלטאות האופטימלית לכל המשימות יחד W. אז המחברים למעשה מתארים את W בתור מכפלה של הסכומים הממושקלים של מטריצות A לכל המשימות עם אותו סכום ממושקל של מטריצות B (עם אותן המשקלות). ואת המשקלות האלו אנו מאפטמים כדי לבנות את W שמתאימה לכל המשימות יחד עם כל הדוגמאות עבור המשימות האלו.   
מכיוון שמספר הפרמטרים המאופטמים הינו די קטן, המאמר משתמש בשיטה הנקראת Covariance Matrix Adaptive Evolution Strategies (CMA-ES) שהיא דורשת חישוב הגרדיאנט. זה שיטה שמחפשת במרחב הפרמטרים על ידי חישוב של פונקציית המחיר עבור סט של פרמטרים ואז מנסה לבחור את סט הפרמטרים הבא בכיוון שמקטין את פונקציית המחיר. בכך הם משתמשים בסוג של קורלציה משוערכת של כל פרמטר עם פונקציית המחיר (איך הוא משפיע על התוצאה). נזכיר שכאן פונקציית המחיר מודדת את הביצועים של הרשת עם מטריצת הדלטאות W על הדוגמאות מכל המשימות.