**הפסל שבתוך מכונה: מדוע האופטימייזר שלכם אינו רק מכונית מרוץ, סקירה 518**

**מאמר היומי של מייק: 02.10.25  
Optimizers Qualitatively Alter Solutions And We Should Leverage This**

במשך רוב העשור האחרון, קהילת הלמידה העמוקה התייחסה לאופטימייזרים כמו למכוניות מרוץ. המשימה הייתה פשוטה: להגיע לתחתית משטח הלוס (loss landscape) מהר ככל האפשר. מדדנו אותם על פי מהירות: איטרציות, FLOPs, זמן ריצה. Adam היה מהיר יותר מ-SGD, וחגגנו (האמת WAdam די ניצח את כולם בזמן האחרון). ההנחה הסמויה, שנלקחה מעולם האופטימיזציה הקמורה (convex optimization) הנקי והצפוי, הייתה שהיעד נקבע מראש. כולנו נסענו לאותו עמק; לחלקנו פשוט היו מנועים טובים יותר.

המאמר שנסקור היום טוען כי המודל המחשבתי הזה אינו רק חלקי, אלא מטעה באופן יסודי. בשטח הפראי והלא-קמור של אימון רשתות נוירונים, האופטימייזר אינו מכונית מרוץ. הוא פסל. הנתיב שהוא חוצב במרחב הפרמטרים הרב-ממדי לא רק קובע את מהירות ההגעה; הוא קובע את עצם צורתו של הפסל הסופי. זוהי התובנה המרכזית ומשנת-התפיסה של המאמר. הבחירה באלגוריתם הלמידה אינה רק כלי להתכנסות, אלא מקור רב עוצמה ולא מנוצל מספיק של הטיה אינדוקטיבית (inductive bias). היא מעצבת באופן פעיל את האופי האיכותי של הפתרון שאנו מוצאים.

#### **אשליית הקמירות**

האובססיה שלנו למהירות ההתכנסות היא שריד לחרדות של עידן עבר. ספקנות מוקדמת לגבי רשתות נוירונים התרכזה בחשש להיתקע במינימה מקומית "רעה". גל של עבודות אמפיריות ותיאורטיות בתחילת שנות ה-2010 הרגיע לכאורה את החששות הללו, והצביע על כך שבמודלים גדולים מספיק, רוב נקודות המינימום המקומיות היו באיכות דומה. הנוף, המשטח נאמר לנו, היה "מתנהג יפה" באופן כללי.

נרטיב זה, על אף שהיה שימושי, עיוור אותנו לאמת עמוקה יותר של אי-קמירות: קיומם של *סוגים רבים ושונים של פתרונות טובים*. אם קיימות מספר נקודות מינימום נפרדות ובעלות ביצועים גבוהים, האלגוריתם שבו אנו משתמשים כדי לנווט בנוף הופך לגורם מכריע בקביעת הפתרון שאליו נגיע. אופטימייזר שלוקח צעדים חמדניים (כמו SGD) יעקוב אחר מסלול שונה וינחת באגן משיכה (basin of attraction) אחר מאשר אופטימייזר שמבין את עקמומיות משטח הלוס ואת הקשרים המורכבים בין פרמטרים (כמו שיטות מסדר שני כמו Shampoo ). הם לא רק נוסעים בדרכים שונות לאותה עיר; הם מגיעים למדינות שונות לחלוטין.

#### **האופטימייזר כמקור להטיה**

המאמר טוען שעלינו לראות באופטימייזר מנוף עיקרי לשליטה ב*אופן* שבו מודל לומד. כלל הלמידה הוא ביסודו מנגנון לייחוס אשמה (credit assignment): ההחלטה איזה ממיליארדי הפרמטרים ״יואשמו בשגיאה״. אופטימייזר פשוט מייחס את האשמה הזו באופן מקומי וקצר-רואי. שיטות מתוחכמות יותר, במיוחד אלו המשתמשות ב-preconditioners לא-אלכסוניים (המנצלים קשרים בין משקלי המודל למטרת אופטימיזציה), מבצעות את הייחוס הזה מתוך הבנה עשירה יותר של יחסי הגומלין בין הפרמטרים.

להבדל זה יש השלכות עמוקות. אופטימייזר שמתקן "תנועה לא יעילה" במרחב המשקלות במהלך אופטימיזציה, כאלו שבה עדכונים לפרמטרים שונים מבטלים השפעתם על הלוס, יעודד את הרשת למצוא ייצוגים דלילים יותר בתתי-מרחבים של המשקולות. הוא דוחף את הפתרון לעבר תת-מרחב בעל ממד נמוך יותר, תכונה רצויה ביותר למשימות כמו למידה continual שבהן מזעור ההפרעות הוא המפתח. זו אינה הטיה שאנו מוסיפים באמצעות מונח רגולריזציה מפורש; זוהי תכונה שנוצרת באופן טבעי מדינמיקת האופטימיזציה עצמה. כך תהליך הלמידה הופך לכלי לקידוד תכונות רצויות של הפתרון, כמו דלילות או רובסטיות, ישירות לתוך הפתרון.

המאמר מבסס רעיון מופשט זה באמצעות דוגמאות קונקרטיות ומשכנעות. בלמידת continual, למשל, אופטימייזר מסדר שני כמו Shampoo, שלוקח בחשבון את היחסים בין הפרמטרים, מוצא פתרונות עמידים יותר לשכחה קטסטרופלית (catastrophic forgetting) מאלו שנמצאים על ידי Adam. הוא עושה זאת על ידי למידת ייצוגים דחוסים וממוקדים יותר, ובכך משתמש בקיבולת של המודל ביעילות רבה יותר. זה אינו שינוי קטן; זהו פתרון שונה מבחינה איכותית, כזה שמתאים יותר לסביבה דינמית.

דוגמה נוספת ממשיגה מחדש טכניקה להשראת דלילות כמקדם מקדים מותאם אישית. על ידי תכנון אופטימייזר שמקשה על משקלים הקרובים לאפס לגדול (ובכך יוצר למעשה נקודות אוכף בנוף ההפסד), הלמידה מוטה לטובת פתרונות שבהם רק פרמטרים מעטים הופכים לגדולים. כאן, האופטימייזר מאט בכוונה את ההתכנסות בכיוונים מסוימים כדי להשיג תכונה מבנית רצויה' דלילות — פשרה שגישת "מכונית המרוץ" לעולם לא הייתה מאפשרת.

#### **יכולת ביטוי מול יכולת השגה (Expressivity vs. Reachability)**

אולי הנקודה החדה ביותר שהמאמר מעלה היא ההבחנה בין מה שמודל *יכול* לייצג לבין מה שהוא *יכול* ללמוד. במשך זמן רב התווכחנו על יכולת הביטוי התיאורטית של ארכיטקטורות, ושאלנו שאלות כמו: "האם מודל זה הוא turing complete?". לטענת המחברים, זהו תרגיל אקדמי במידה רבה אם אנו מתעלמים מאילוצי הלמידה.

ארכיטקטורה מגדירה יקום עצום של פונקציות אפשריות. אך האופטימייזר, יחד עם האתחול והדאטה, חוצב תת-קבוצה קטנה בהרבה, הניתנת להשגה, בתוך אותו יקום. רשת נוירונים רקורנטית עשויה תיאורטית להיות מסוגלת לייצג אלגוריתם מורכב, אך אם שיטות מבוססות גרדיאנט אינן יכולות למצוא את הפתרון הזה מנקודת התחלה אקראית, כוחה התיאורטי חסר משמעות. לכן, האופטימייזר מגדיר את *יכולת הביטוי האפקטיבית* של המודלים שלנו. הוא שומר הסף בין מה שניתן לייצוג לבין מה שניתן להשגה.

זה ממסגר מחדש את פרקטיקת תכנון המודלים. אנו רגילים להטמיע הטיות בארכיטקטורות. מאמר זה מציג טיעון משכנע לכך שזה תקף באותה מידה, ולפעמים אלגנטי יותר, להטמיע אותן באופטימייזרים שלנו. קיימת כאן דואליות שהתעלמנו ממנה במידה רבה.

המחברים קוראים לשינוי יסודי באופן שבו אנו מעריכים ומתכננים אלגוריתמי למידה. במקום לשאול "כמה מהר זה?", עלינו לשאול "איזה סוג של פתרון זה מייצר?". זוהי קריאה לנוע מעבר למסלול המרוצים ולהיכנס לסטודיו של הפסל.

https://arxiv.org/abs/2507.12224