המאמר היומי של מייק: 27.07.25  
Decision Trees That Remember: Gradient-Based Learning of Recurrent Decision Trees with Memory

עצי החלטה הם אבן יסוד בלמידת מכונה. הם אינטואיטיביים, חזקים, והכי חשוב, ניתנים לפירוש (interpretable). אפשר בקלות לעקוב אחר הלוגיקה של "אם-אז", ולהבין בדיוק כיצד הוא הגיע להחלטה. אבל יש להם חולשה בולטת: הם חסרי מצב (stateless). הם מתייחסים לכל דגימה כהתחלה חדשה, תוך התעלמות מוחלטת מהעבר. זה הופך אותם ללא כשירים לדאטה סדרתי, כמו סדרות עתיות, שפה, אודיו שבהם להיסטוריה יש חשיבות מכרעת.

המאמר מציע עצי החלטה רקורסיביים עם זיכרון (ReMeDe Trees), ארכיטקטורה חדשנית המתאימה לדאטה סדרתי. מודל זה שואף לגשר על הפער שבין היכולת לפירוש הגבוהה של עצי החלטה לבין יכולת המידול הטמפורלי של רשתות נוירונים RNNs שזה Recurrent Neural Nets. זהו ניסיון מעניין לקבל את הטוב משני העולמות, והביצוע הטכני הוא המקום שבו הקסם האמיתי קורה.

איך מעניקים זיכרון למבנה שתוכנן להיות חסר זיכרון? הפתרון של עצי ReMeDe הוא אלגנטי: המודל לא רק מבצע חיזוי; הוא גם מחליט כיצד לעדכן את מצב הזיכרון הפנימי שלו בכל שלב. זה מושג באמצעות מערכת עצים כפולה ייחודית. בכל צעד זמן, המודל לא משתמש בעץ אחד, אלא בשניים:

עץ הפלט T\_out: זהו ה"חזאי". הוא מקבל את נתוני הקלט הנוכחיים וגם את הזיכרון מהשלב הקודם כדי לייצר את החיזוי הסופי.

עץ עדכון המצב (T\_state​): זהו "כותב הזיכרון". הוא גם מסתכל על הקלט הנוכחי ועל הזיכרון הקודם, אך תפקידו הבלעדי הוא לחשב את מצב הזיכרון החדש שיועבר לצעד הזמן הבא.

מבנה עצים כפול זה מאפשר למודל ללמוד לוגיקות נפרדות ומתמחות לביצוע חיזויים לעומת זכירת מידע לעתיד. הפלט של עץ עדכון המצב הופך לזיכרון הקלט עבור איבר הבא בסדרה, ובכך יוצר זרימת מידע. זהו רעיון די חזק אבל האתגר האמיתי טמון באימון שלו. עצי החלטה מסורתיים נבנים באמצעות אלגוריתמים חמדניים (כמו CART) המשתמשים במדדים לא-גזירים (non-differentiable) כמו מדד Gini. אי אפשר להשתמש בהם בירידה בגרדיאנט (gradient descent). כדי לאמן את המערכת הרקורסיבית הזו מקצה לקצה, המודל כולו צריך להיות גזיר.

עצי ReMeDe פותרים זאת באמצעות טכניקה הנקראת ניתוב גזיר (differentiable routing). במהלך האימון, במקום לבצע פנייה "שמאלה או ימינה" באופן קשיח בכל צומת, המודל מבצע בחירה "רכה" והסתברותית. בכל פיצול, העץ מסתכל על תכונה (feature) ספציפית מהקלט ומשווה אותה לסף (threshold) נלמד. השוואה זו מוזנת לפונקציה מיוחדת המוציאה כפלט הסתברות, מספר בין 0 ל-1, לאיזה נתיב ללכת.

אם ערך התכונה גבוה בהרבה מהסף, ההסתברות ללכת ימינה מתקרבת ל-1. אם הוא נמוך בהרבה, ההסתברות ללכת שמאלה מתקרבת ל-1. אם הערך קרוב מאוד לסף, הבחירה אינה ודאית, וההסתברות מרחפת סביב 50/50. פרמטר מכריע של "טמפרטורה הפוכה" פועל כמו כפתור ביטחון: ככל שהאימון מתקדם, כפתור זה "מוגבר", מה שהופך את הפונקציה לרגישה יותר ומאלץ את ההסתברויות להתקרב לקצוות של 0 או 1. המשמעות היא שקלט אינו עוקב אחר נתיב בודד. במקום זאת, הוא "זורם" במורד כל הנתיבים האפשריים לכל העלים בו-זמנית. הפלט הסופי ומצב הזיכרון החדש מחושבים כממוצע משוקלל של כל ערכי העלים, כאשר המשקל של כל עלה הוא ההסתברות להגיע אליו.

מכיוון שהמערכת כולה, מהקלט, דרך הניתוב ההסתברותי ועד לפלט הסופי המבוסס על ממוצע משוקלל, היא כעת פונקציה חלקה וגזירה, ניתן לאמן אותה בדיוק כמו רשת נוירונים. המודל משתמש ב-(Backpropagation Through Time (BPTT, האלגוריתם הסטנדרטי לאימון רשתות RNNs, כדי לחשב את הגרדיאנטים של פונקציית לוס ביחס לכל פרמטרי המודל (ספי הפיצול וערכי העלים). זה מאפשר למודל ללמוד דפוסים טמפורליים מורכבים על פני סדרות ארוכות.

ה"עץ הרך" הזה מצוין לאימון, אבל אנחנו מאבדים את היתרון המרכזי של היכולת לפירוש. השלב האחרון והמבריק בתהליך הוא הקשחה (hardening). כפי שצוין, "פקטור הביטחון" (פרמטר β) "מוגבר" לאורך האימון. זה הופך את ההחלטות ה"רכות" ההסתברותיות לפחות ופחות מעורפלות. בסוף האימון, הן למעשה הופכות להחלטות "קשיחות" רלומר דטרמיניסטיות. התוצאה היא מודל סופי שהוא עץ החלטה סטנדרטי וניתן לפירוש עם כללי "אם-אז" קלאסיים. אפשר לבחון אותו ולהבין לא רק כיצד הוא מבצע חיזויים, אלא גם כיצד הוא בוחר לעדכן את הזיכרון שלו בהתבסס על הקלט שהוא רואה.

https://arxiv.org/abs/2502.04052