סקירת המאמר:

Reluplex: An efficient SMT solver for verifying deep neural networks

שפורסם ב-2017 על ידי Stanford AiSafety, כאשר המחבר הראשי הוא ד"ר גיא כץ, כיום באוניברסיטה העברית.

סקירה זו היא חלק ראשון מתוך שני חלקים, כאשר הסקירה השנייה תהיה על מאמר ההמשך – Marabou, שפורסם בשנה שעברה.

כתבתי בעבר על הנושא הכללי של אימות רשתות נוירונים – בלינק הזה.

הוצג בכנס: CAV'17

תחומי מאמר:

אימות רשתות נוירונים, adversarial examples.

כלים מתמטיים, מושגים וסימונים:

אלגוריתם סימפלקס (Simplex), בעיות SAT, adversarial examples.

תמצית מאמר:

המונח SAT מגיע מענף הלוגיקה, ומשמעותו – האם לבעיה מסוימת יש השמה נכונה שתפתור את הבעיה. למשל, הנוסחה "A or B" היא ספיקה (SAT), כי עבור A is true מתקבל ערך אמת עבור הנוסחה כולה. לעומת זאת, הנוסחה " A and not A" אינה ספיקה (UNSAT), כי לא ייתכן כי גם "A" וגם "not A" יהיו יחד בעלי הערך "אמת".

ניתן לקחת את הרעיון הזה וליישם אותו גם עבור רשת נוירונים. נניח ויש לנו תמונה של הספרה 1, בגודל של 28x28. כעת אימנו רשת נוירונים מסוימת, והיא אכן מפענחת בצורה נכונה את התמונה ובמוצא מתקבל 1. בשלב זה, ניקח את התמונה הזו (לשם הנוחות נסתכל עליה בוקטור באורך 784), ועבור כל פיקסל נאפשר רעש בגודל אפסילון. כעת נניח ונרצה שהרשת תיתן במוצא עבור התמונה המורעשת את הספרה 8. למעשה יצרנו פה בעיית SAT – האם עבור וקטור הכניסה פלוס רעש מסוים (שכמובן יכול להשתנות מפיקסל לפיקסל) יש השמה מסוימת, שהמשקלים של הרשת המאומנת יתנו במוצא 8. אם קיימת השמה, אז הבעיה היא SAT, ומצאנו דוגמא נגדית שהרשת טועה בפענוח שלה. אם הבעיה אינה SAT אלא UNSAT, אז עבור התמונה שהכנסנו בכניסה והרעש שאפשרנו – הרשת רובסטית ואינה טועה.

לאחר שהבנו את המשמעות של בעיות SAT והקשר שלהן לרשתות נוירונים, נותר למצוא דרך להכריע האם יש השמה או לא, כאשר ישנם שני אתגרים מרכזיים:

א. בעיות SAT הינן בעיות NP-שלמות.

ב. אמנם קיימים אלגוריתמים לקביעה האם לבעיה מסוימת יש השמה או לא, אך הם לא מתאימים לרשת נוירונים שבאופן כללי אינה לינארית.

המאמר מציע פתרון לאתגר השני על ידי הרחבה של אלגוריתם Simplex לרשתות נוירונים, ומספר דרכים להתמודד עם האתגר הראשון – כיצד להאיץ את פתרון בעיית ה-SAT.

אלגוריתם סימפלקס בא מעולם התכנות הלינארי, והוא נועד למצוא אופטימיזציה לבעיה בה יש מספר אילוצים לינאריים. בקצרה ניתן לומר שהשיטה לוקחת מספר אילוצים ומספקת אלגוריתם איטרטיבי למציאת האופטימום של כלל האילוצים (או קביעה כי אין אופטימום והבעיה אינה קמורה). ניתן להסתכל על וקטור הכניסה לרשת הנוירונים בתוספת הרעש כאוסף של אילוצים, הצריכים לקיים יחד קומבינציה כלשהיא עם משקלי הרשת על מנת להגיע למוצא מסוים. זה בעצם הרעיון המרכזי להתמודדות עם האתגר השני שהצגנו – כיצד לקבוע האם יש השמה מסוימת לכניסה והמוצא שקבענו. אך עדיין חסר דבר אחד – סימפלקס מתאים לבעיה בה כל האילוצים לינאריים, וזה לא המצב ברשת נוירונים. אך פה נכנסת ההרחבה (ממנה בא השם Reluplex) – ניתן לקחת כל צומת Relu ולפרק אותו לשני חלקים לינאריים. באופן הזה המבנה הכללי נהיה פתיר – יש וקטור כניסה מורעש, שעובר דרך משקלים שהם לינאריים למקוטעין וניתן להסתכל עליהם כאוסף של חלקים לינאריים, ובנוסף יש מוצא רצוי כלשהוא. את כל זה ניתן להכניס לתוך אלגוריתם סימפלקס ולבדוק האם יש השמה (ואז להכריע שהבעיה הינה SAT, ובעצם מצאנו דוגמא נגדית) או להכריע שאין השמה.

לגבי האצת החישוב – המחברים מציעים מספר דרכים – use of tighter bound derivation, conflict analysis, floating point arithmetic and under-approximation, ולא ארחיב עליהם כרגע.

מונחים נוספים השייכים לתחום שמוזכרים במאמר אם כי הם כלליים יותר: אלגוריתם Reluplex הינו Sound ו-Complete. המשמעות היא שהתשובה שהוא מחזיר היא בהכרח נכונה (Sound) ועבור כל בעיה שנזין לו, נקבל תשובה

(Complete). כמובן שזה דורש סיבוכיות גבוהה, אך זה מבטיח תשובות מהימנות עבור כל בעיה שנזין. ישנם אלגוריתמים אחרים שמעדיפים לוותר על הדיוק הזה לצורך האצת החישובים – למשל אלגוריתמים שמשתמשים בקירובים לא תמיד יהיו Sound, כיוון שיתכן שהם יחזירו תשובה אך היא תהיה שגויה.

תוצאות המאמר:

המחברים לקחו use-case מעניין הנקרא Airborne Collision Avoidance System X (ACAS X). זוהי בעצם משפחה של מערכות המלצה למניעת התנגשות בין מטוסים. עד לפני כמה שנים המידע היה שמור בטבלאות ענקיות. דבר זה התאפשר כיוון שמספר המצבים היה סופי (כמה מטוסים יש באזור, באיזה כיוון הם, מה המהירות שלהם וכו'). החיסרון היה במימד הגבוה של מרחב המצבים, שגם גרם לטבלאות להיות ענקיות, וגם הביא לתגובות איטיות יחסית של המטוסים, כיוון שנדרש זמן לחשב את המצב הנתון ולמצוא בטבלאות את התגובה הנדרשת. כמו כל דבר אחר בשנים האחרונות, גם לבעיה זו נבנתה רשת נוירונים שהייתה קומפקטית ומהירה. רשת זו נבדקה על ידי Reluplex – לקחו input אפשרי מסוים, הרעישו אותו מעט, ובדקו האם יש השמה כלשהי בטווח הרעש המותר, שיביא לפלט שגוי, כלומר לתגובה לא נכונה של המטוס עבור המצב הספציפי הקיים. הרשת מצאה השמות עבור כל מיני מקרים, והמשמעות של ההשמות האלו היא שיש דוגמאות בהן הרשת טועה – כיוון שהכניסו input מסוים, ושאלו האם רעש מסוים יכול להביא למוצא שגוי. כיוון שאלגוריתם האימות הכריע שהבעיה הינה SAT, אז בעצם יש השמה השקולה לדוגמא נגדית.

הישגי מאמר: ההישג העיקרי של המאמר הוא פריצת הדרך שהוא עשה בתחום של אימות רשתות. המאמר בעצם פתח את הדלת להרבה חוקרים עבור תחום זה, והיום כל מי שמתעסק וכותב מאמרים בתחום מזכיר את השיטה הזו ומשווה את עצמו עליה.

כמובן שהמאמר גם הצליח להראות מדוע יש צורך בתחום הזה, ושהאלגוריתם המוצע עובד, בעזרת ה-case study של ה-ACAS X.

יש שני חסרונות לשיטה, והמחברים מתמודדים איתם:

א. הסיבוכיות עבור כל שאלת SAT היא אקספוננציאלית. כפי שציינתי, המחברים הציעו מספר שיטות על מנת להאיץ את חישוב שאלת ההשמה.

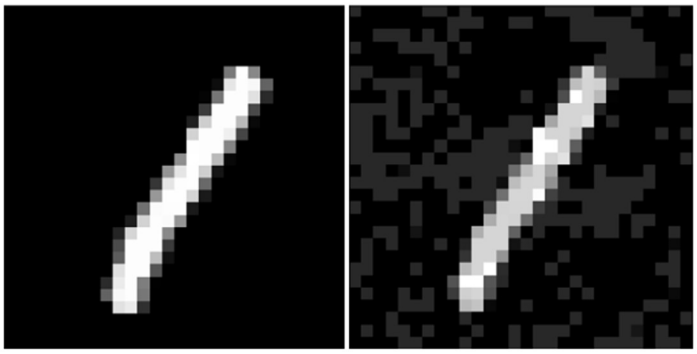
ב. האלגוריתם טוב רק לפונקציית הפעלה Relu. במאמר ההמשך Marabou המחברים מכלילים את האלגוריתם כך שיתאים לפונקציות נוספות.

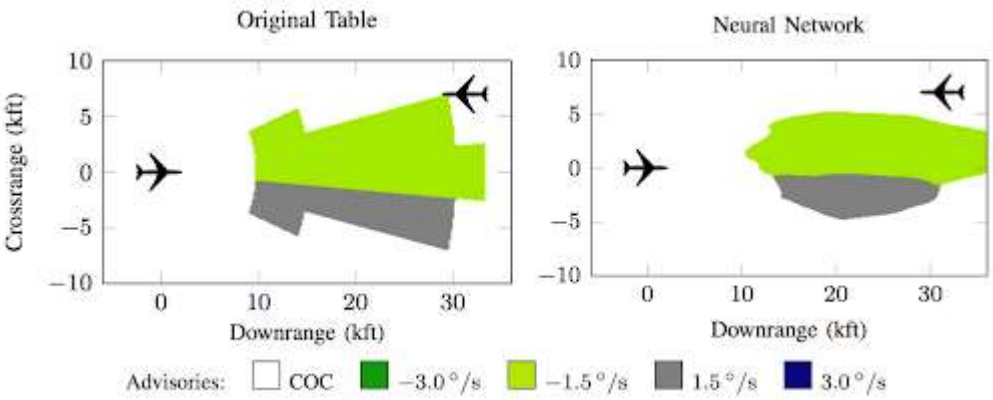
לינק למאמר: https://arxiv.org/abs/1702.01135

לינק לקוד: https://github.com/guykatzz/ReluplexCav2017

נ.ב. מאמר מאוד מעניין שניתן לראות בו בין היסודות של התחום בו הוא עוסק. אלגוריתמים רבים פותחו בעקבותיו, וגם אלגוריתם Reluplex עצמו עבר שיפורים והכללות.

בחלק הבא של הסקירה אדבר על מאמר ההמשך Marabou שמציע הכללות ושיפורים (למשל: פונקציות הפעלה, האצה בזמני החישוב ועוד).





#deepnightlearners