**למידה עמוקה על מאיצים חישוביים פרויקט סוף**

**רקע:**

רשתות עמוקות ידועות ברגישותן להפרעות בעלות נורמה מוגבלת, המשבשות את הפלט באופן ניכר.

ניתן לנצל רגישות זו, וליצור הפרעה מכוונת, שמטרתה לשבש את פלט הרשת, על מנת לפגוע בתפקודה.

ישנם שני סוגים של התקפות, שמטרתן ליצור הפרעה כזו.

סוג אחד הוא התקפה ייעודית עבור אוסף קלטים נתון. כלומר, בהינתן אוסף הקלטים של הרשת, מטרת ההתקפה היא ליצור הפרעה שתשבש אך ורק את הפלטים עבור אותם הקלטים הנתונים.

הסוג השני הוא התקפה אוניברסלית. כלומר, בהינתן אוסף של קלטים עבור הרשת, מטרת ההתקפה היא ליצור הפרעה שתשבש קלטים שעדיין לא נראו, שהתפלגותם דומה לאלה שניתנו.

בפרויקט זה, מטרתנו היא לבצע התקפה אוניברסלית עבור מודל VO, שתפקידו להסיק את תנועתו של דאון (מיקומו וכיוון הסתכלותו), ע''פ רצף של תמונות הנצפות ע''י הדאון במהלך תנועתו.

דרגת החופש שלנו ביצירת ההפרעה, מוגבלת למשטח דו ממדי בגודל נתון, המהווה חלק מהמרחב התלת ממדי שבו הדאון נע. כלומר, אנו יכולים לשנות את כל הפיקסלים אך ורק על גבי אותו המשטח, בטווח ערכים שנע בין ערך האלבדו השחור, לבין ערך האלבדו הלבן של אותו הפיקסל.

איכות ההתקפה נמדדת באמצעות המרחק בין המיקום אליו מגיע הדאון ע''פ מודל ה-VO ללא ההפרעה, לבין המיקום אליו מגיע הדאון ע''פ מודל ה-VO בהתחשב בהפרעה.

לשם יצירת התקפה כזו, השתמשנו באלגוריתם אופטימיזציה, בשילוב עם פונקציית loss שעוצבה עבור הבעיה הספציפית הזו, שתפקידו היה למצוא הפרעה שתמקסם את איכות ההתקפה, ע''פ המדד התואר לעיל.

בנוסף לכך, השתמשנו בשיטות הכללה, שמטרתן להשיג תוצאות טובות עבור קלטים שטרם נצפו בזמן האימון.

בחנו את האלגוריתם במספר אופנים שיתוארו בהמשך, ומצאנו שאכן הושגו הפרעות משמעותיות, עבור קלטים שלא נעשה בהם שימוש במהלך האימון.

**הקדמה:**

בפרויקט זה נעזרנו בשני מאמרים.

אחד המאמרים הוא:

Nemcovsky, Y., Yaakoby, M., Bronstein, A.M., Baskin, C.: Physical passive patch adversarial attacks on visual odometry systems. arXiv preprint arXiv:2207.05729 (2022)

הבעיה הנחקרת במאמר זה, זהה לבעיה שלנו. במאמר זה, מנסים ללמוד כיצד לבצע התקפה, שתיצור משטח פיזי סטטי במערכת ניווט אוטונומית, ותשבש את יכולת הניווט של מודל VO.

עבודות שקדמו למאמר זה, שעסקו בהתקפות עבור מודלים רגרסיביים, דיברו לרוב על התקפות סטנדרטיות, שבהן ההפרעה מוכנסת ישירות לתוך התמונה המתקבלת (במקרה שלנו ע''י הדאון).

בניגוד לכך, במאמר זה, ההפרעה היא משטח פיזיקלי, המוכנס באופן סטטי למרחב התלת ממדי. כלומר, הוא נצפה באופן שונה ממקומות שונים במרחב. בעיה זו היא הרבה יותר ריאלית, שכן לא נדרשת גישה ישירה לקלט של המודל, אלא רק למרחב עצמו.

החלקים אותם ננסה לשפר בפתרון המוצע במאמר, הם שיפור פונקציית ה-loss, שתיקח בחשבון אלמנטים נוספים פרט למיקום הדאון, כמו הכיוון אליו הוא פונה, ושיפור האופטימייזר המוצג בו (pgd), ע''י החלפתו באופטימייזר שידע לשנות את הפרמטר אלפא באופן דינמי.

המאמר השני בו נעזרנו הוא:

**רעות: ...**

**שיטות:**

**השיטה המקורית:**

השיטה המקורית לפתרון הבעיה, היא שימוש באופטימיזציית pgd, בשילוב עם אחת מפונקציות ה-loss: .

הלמידה מתחילה מהפרעה רנדומלית, ובכל איטרציה מתעדכנת ע''י צעדי GD.

ישנן שתי פונקציות loss שמעורבות בתהליך הלמידה. אחת מהן מסומנת ב-, ובעזרתה מחושבים צעדי ה-GD, והשנייה מסומנת ב-, ומטרתה לבחון האם חל שיפור מהאיטרציה הקודמת בעקבות עדכון ההפרעה.

כמו כן, ישנן שתי קבוצות של מסלולים, שעל האחת מתבצע האימון, ועל השנייה מתבצעת בדיקת השיפור.

בכל איטרציה, עבור כל אחד מהמסלולים השייכים לקבוצת האימון, מחושב הגרדיאנט של הפעלת פונקציית ה-, על סדרת הפוזיציות (המיקומים במרחב, וכיווני ההסתכלות) של הדאון במהלך תנועתו ע''פ מודל ה-VO, ועל סדרת הפוזיציות ע''פ אותו המודל, כאשר הפעם נלקחת בחשבון ההפרעה שנלמדה באיטרציה הקודמת.

מסכמים את כל הגרדיאנטים אללו, על מנת לקבל את הגרדיאנט של סכום ה-loss-ים, ומעדכנים את ההפרעה ע''י הוספת אלפא פעמים סימן הגרדיאנט.

לאחר מכן, תוחמים את ההפרעה לטווח הערכים החוקיים, כך שכל פיקסל יקבל ערך בתחום שבין 0 ל-1.

בשלב הבדיקה, שנעשה לאחר שלב האימון בכל איטרציה, עבור כל אחד מהמסלולים השייכים לקבוצת הבדיקה, מחשבים את ה- על סדרת הפוזיציות של הדאון במהלך תנועתו ע''פ מודל ה-VO, ועל סדרת הפוזיציות ע''פ אותו המודל, כאשר הפעם נלקחת בחשבון ההפרעה שנלמדה באיטרציה הנוכחית.

אם התקבל שיפור, כלומר ערך ה- גבוה יותר מערכו הגדול ביותר עד כה, מעדכנים את ההפרעה המקסימלית להיות ההפרעה הנוכחית.

המידה ולא, ההפרעה המקסימלית תישאר כפי שהייתה לפני האיטרציה הנוכחית.

לבסוף, תוחזר ההפרעה המקסימלית שהתקבלה לאחר מספר האיטרציות המבוקש.

**השיפורים שלנו:**

ישנם מספר שיפורים שעשינו באלגוריתם המקורי (זה שכתוב בקובץ התרגיל).

אחד השיפורים הוא שכלול פונקציית ה- המקורית.

שלוש פונקציות ה-loss המדוברות המאמר (, אינן מנצלות האופן מקסימלי את המידע המצוי בסדרות הפוזיציות עליהן הן פועלות.

*ישנה קורלציה גבוהה בין השוני בכיוון אליו פונה הדאון בזמן נתון, והכיוון אליו הוא חושב שהוא פונה בהסתמך על ההפרעה, לבין השוני בין מיקום הדאון בנקודת הזמן הבאה והמיקום בו הוא חושב שהוא ימצא בנקודת הזמן הבאה בהסתמך על ההפרעה.*

*ניסינו לנצל קורלציה זו, ולהוסיף עוד גורם לפונקציית ה-, שיתן משקל לשוני שבין שני הכיוונים אללו. את המשקל עצמו כווננו במהלך הניסויים.*

*בנוסף לכך, ניסינו לנצל את הקורלציה שבין השוני של שתי הרשתות שמהן מתקבלים הכיוון והמיקום של הדאון, עם ובלי ההתחשבות בהפרעה, לבין ההפרש של שני המיקומים הסופיים. לכן, הוספנו עוד גורם לפונקציית ה- שמודד את ה-MSE בין שני אלה, וחישבנו את המשקל המתאים ע''י ניסויים.*

***אולי להוסיף פה עוד דברים על ה-loss בהתאם למה שיצא בניסויים.***

*שיפור נוסף שעשינו, הוא הרחבת יכולת ההכללה של האלגוריתם.*

*באופן בו האלגוריתם מומש, ביצועיו נמדדו ע''פ ערך ה- של קבוצת מסלולי ה-eval, שהיו מעורבים בתהליך הלמידה.*

*מכיוון שקבוצה זו הייתה מעורבת בתהליך הלמידה, והאלגוריתם בחר בהפרעה שממקסמת את ה- עבור קבוצה זו, ישנה זליגה של מידע מקבוצה זו למודל.*

*שיפרנו את הזליגה הזו ע''י חלוקה לקבוצה נוספת של טסט, שאינה מעורבת כלל בתהליך האימון, ולבסוף בהינתן ההפרעה הסופית, היא זו שמדדה את ביצועיה.*

*מכיוון שכמות המסלולים שניתנו לנו היא קטנה (בסך הכל 50), וחלקם משמשים בתור eval, וכעת חלק נוסף משמשים בתור test, יכולת בדיקה יחידה לתת מדד לכמה ההפרעה טובה היא מוגבלת. לכן מימשנו cross validation, כדי לנצל את הנתונים באופן מיטבי, ובכך לקבל תוצאות אמינות יותר, ע''י מיצוע התוצאות המתקבלות ע''י כל ה-folds. מכיוון שהקצאת data עבור הטסט מקטינה את ה-data עבור האימון ועלולה לפגוע בביצועים, בשלב יצירת ההפרעה הסופית, לאחר שבחרנו את כל הפרמטרים, אנו מבטלים את החלוקה הזו, ומאמנים על כל ה-data.*

*שיפור נוסף שעשינו, הוא שינוי פונקציית האופטימיזציה (pgd).*

***רעות: ... (צריך להוסיף גם את ה-early stopping ולכתוב עליו)***

***ניסויים:***

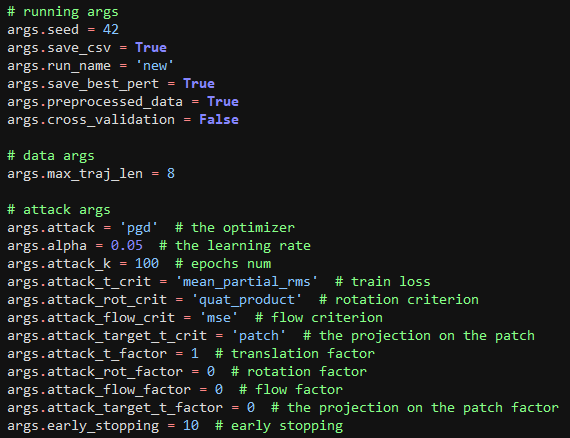
***הניסויים שביצענו בוחנים את הדברים הבאים:***

* *האם כדאי להשתמש ב-, או ב- בתור ה-.*
* *איזה משקל לתת לרכיב הסיבוב ב-loss.*
* *איזה משקל לתת לרכיב ה-hidden layer ב-loss.*
* *איזה פרמטר אלפא מתאים ביותר.*
* *האם כדאי להשתמש ב-apgd במקום ב-pgd.*

*את בחירת הפרמטרים נעשה בצורה גרידית. כלומר, במהלך ניסוי נכוונן פרמטר יחיד, ולאחר שנמצא את ערכו האופטימלי, נקבע אותו על ערך זה, ונריץ את הניסוי הבא בהתחשב באותו הערך.*

***קונפיגורציה ופרטי מימוש:***

*הקונפיגורציה הבסיסית (לפני כוונון הפרמטרים שיתבצע לאחר כל הרצה) היא זו:*



* *הניסויים יתבצעו על seed קבוע.*
* *ה-preprocessing של הקוד המקורי יהיה מופעל בכל הניסויים (פרט לניסוי הראשון), על מנת לקצר את זמן הניסוי.*
* *ה-cross validation מאריך את זמן הניסוי פי 5, ולכן נפעיל אותו רק אם יישאר מספיק זמן עד ההגשה, וכתלות בעומס על השרת. תפקידו של ה-cross validation, הוא לחלק את 5 תיקיות ה-data ל-3 תיקיות שמיועדות לאימון, תיקיה אחת לוולידציה ותיקיה אחת לטסט. האימון יתבצע ע''י תיקיות האימון והוולידציה (כפי שמתואר במאמר עבור pgd), והטסט יבחן את טיב התוצאות. חלוקה זו תתבצע 5 פעמים, כאשר בכל פעם יתחלפו תיקיות הטסט והוולידציה (ובהתאם לכן גם האימון), וטיב התוצאה יוגדר ע''י ממוצע הטסטים של כל חמש ההרצות.*
* *תפקידו של הפרמטר run\_name הוא להפריד בשמות התיקיות בין ריצה לריצה.*
* *הפרמטר max\_traj\_len קובע מהו מספר הפריימים בכל מסלול, ולכן ערכו יישאר 8 למשך כל הניסויים.*
* *הפרמטר attack קובע באיזה אופטימייזר משתמשים (pgd או apgd). האופטימייזר apgd ממומש באופן הבא: רעות: (להסביר פרטי מימוש ולהוסיף פרמטרים שרלוונטיים אם יש כאלה)*
* *הפרמטר alpha קובע קובע את ה-learning rate של האופטימייזר pgd.*
* *הפרמטר attack\_k יקבע את כמות האפוקים. בכל הניסויים נשאיר את הפרמטר הזה קבוע (100 אפוקים, אלא אם early stopping יעצור את הלמידה לפני)*
* *הפרמטר attack\_t\_crit קובע האם ה- מבוסס על או על . נשים לב שפרמטר זה לא אחראי על , שבמימוש שלנו תמיד מוגדר להיות ללא כל תוספת (סיבוב, זרימה וכו').*
* *הפרמטר attack\_rot\_crit קובע שנלקחת בחשבון הזווית שבין כיוון הסתכלות המוערך של הדאון עם ובלי ההפרעה. חישוב זה נעשה ע''י מכפלה סקלרית בכיווני ההסתכלות. הפרמטר הזה ישאר קבוע, ומידת השפעתו תהיה תלויה ב-attack\_rot\_factor. בתחילת ביצוע הניסויים, הפרמטר הזה לא ילקח בחשבון, מכיוון שערכו של attack\_rot\_factor הוא 0.*
* *הפרמטר attack\_flow\_crit קובע שנלקחת בחשבון שכבת הביניים של רשת ה-VO, שממנה נגזרים המיקום וכיוון ההסתכלות של הדאון. באופן קבוע ימדדה מרחק בין שכבת הביניים עם ההפרעה לזו ללא ההפרעה ע''י mse. בדומה ל-attack\_rot\_crit, גם פה מידת ההשפעה של הרכיב הזה תלויה בפרמטר attack\_flow\_factor.*
* *לא יודע אם להוסיף את ה-attack\_tartet\_t\_crit.*
* *תפקידו של הפרמטר early\_stopping הוא לקבוע לאחר כמה צעדים ללא שיפור תוצאות הטסט נעצור את האימון. בסיום כל אחד מהאפוקים, נבחנת ההפרעה החדשה על הטסט. במידה ולא חל שיפור במהלך early\_stopping צעדים רצופים, האימון יופסק.*

***מטריקה למדידת הביצועים:***

*המטריקה לפיה נמדוד את הביצועים היא ערך ה-rms של הטסט.*

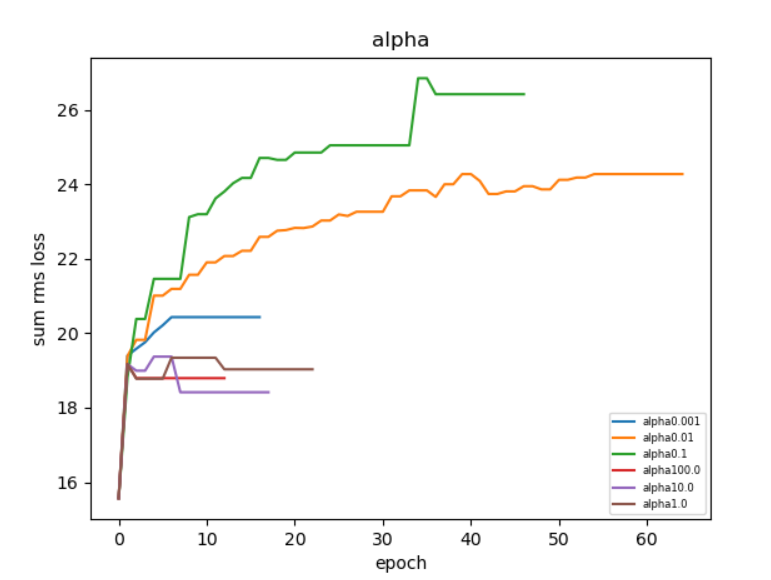
***תוצאות:***

***כיוונון הפרמטר אלפא:***

*מטרת הניסוי היא למצוא את ערך האלפא האופטימלי עבור האופטימייזר pgd.*

*בניסוי זה נבחן את ערכי אלפא מתוך הקבוצה .*

*תוצאות הניסוי מתוארות בתרשים הבא:*



*כפי שהיינו מצפים, עבור ערכים קטנים מדי של אלפא (0.01), הגרף יעלה בקצב איטי מאוד, ולכן ה-early stopping יעצור את ההתקדמות בשלב מסוים. ככל שמגדילים את ערך האלפא עד ל-0.1, קצב הלמידה הולך וגובר. כאשר אלפא גדול מדי (החל מ-1), הגרף מתחיל לזגזג, ונעצר בשלב מוקדם של הלמידה, שכן לא חלה התקדמות. בניסוי זה נבחר הערך אלפא להיות 0.1, והוא זה שישמש אותנו ביתר הניסויים.*

***בחירת ה-:***

מטרת הניסוי היא לבחון האם לבסס את פונקציית ה- על rms או על mean partial rms. נשווה את המימושים הבסיסיים של שתי פונקציות ה-loss אללו.

להוסיף ...

ע''פ תוצאות הניסוי, מתקבל ש...