סקירה זו היא חלק מפינה קבועה בה שותפיי ואנוכי סוקרים מאמרים חשובים בתחום ה-ML/DL, וכותב גרסה פשוטה וברורה יותר שלהם בעברית. במידה ותרצו לקרוא את המאמרים הנוספים שסיכמנו, אתם מוזמנים לבדוק את העמוד שמרכז אותם תחת השם deepnightlearners.

לילה טוב חברים, היום אנחנו שוב בפינתנו deepnightlearners עם סקירה של מאמר בתחום הלמידה העמוקה. היום בחרתי לסקירה את המאמר שנקרא:

SafeDiffuser: Safe Planning with Diffusion Probabilistic Models

---

פינת הסוקר:

המלצת קריאה מיניב ומייק: קריאה מומלצת לאוהבי מודלים גנרטיבים שמבינים קצת בבקרה

בהירות כתיבה: **גבוהה**/בינונית/נמוכה וכאלא.

ידע מוקדם:

- מודלי דיפוזיה גנרטיביים (DDPM)
- יסודות של למידה באמצעות חיזוקים (reinforcement learning- RL)
- רקע בתכנון משימות עם שיטות RL

יישומים פרקטיים:

- רובוטיקה
- ניווט

פרטי מאמר:

לינק למאמר: כאן.

קוד: כאן

קולב: כאן

פורסם בתאריך: 21/12/2022 (v2, בארקיב)

הוצג בכנס: ICML 2022

תחומי מאמר:

- למידה מחיזוקים (RL)
- מודלי דיפוזיה גנרטיביים (DDPM)
- תכנון משימות על שיטות RL

כלים מתמטיים, מושגים וסימונים:

- הסרת רעש(denoising) הדרגית בתהליכי דיפוזיה גנרטיביים
- דגימה מונחית מסווג (classifier guided) במודלי דיפוזיה
- אופטימיזציה פונקציית התגמול (reward) בבעיות RL
- דגימה מסט של מסלולים (trajectories) כפונקציה של התגמול

מבוא:

<u>בעיות תכנון</u>

בעיות תכנון דורשות מציאה של סדרת פעולות כך שהסוכן ינוע ממצב התחלתי אל מצב סופי באופן רצוי (אופטימלי). בד"כ המשימה מנוסחת בהקשר של ניווט (כמו מצא לי מסלול מנק' א' לנק' ב'), אך לפעמים בעיות תכנון מופיעות גם בהקשרים אחרים כמו:

- מציאת סדרת טיפולים רפואיים שתמקסם את סיכויי ההחלמה של מטופל
- מציאת מדיניות להצגת פרסומות באתר שתמקסם את הלחיצות.

סקירה זו תתייחס בעיקר להקשרי הניווט, ולסוכן שלנו נתייחס כרובוט. את המטרה שעל הרובוט לבצע ניתן להגדיר באמצעות פונקציית תגמול (reward), כאשר מטרתו של הסוכן היא לבצע את הפעולות שיניבו לו תגמול מקסימלי. תכנון ניתן לבצע באמצעות למידה, למשל למידה מחיזוקים, או באמצעות אלגוריתמים קלאסיים מעולם הבקרה והתכנון.

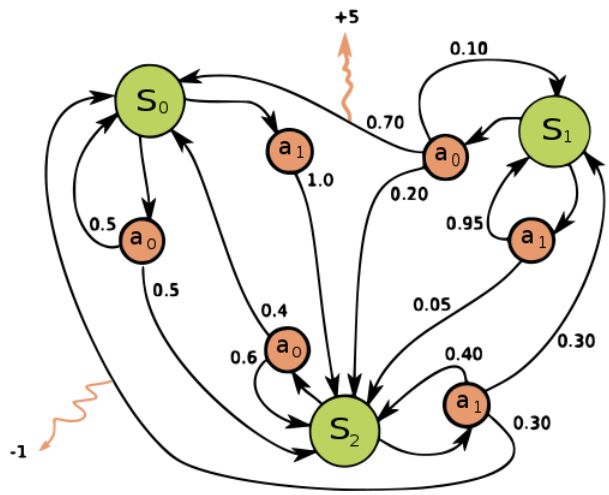
בניסוח מתמטי מצב s_i הוא וקטור המתאר את הסוכן והסביבה שלנו בנקודת זמן i והפעולה a_i היא וקטור המייצג פעולה של הסוכן באותה נק' זמן. התוצר המבוקש של תהליך התכנון הוא מסלול $\boldsymbol{\tau} = [s_0, a_0 \dots s_{t-1}, a_{t-1}, s_t]$ שזהו מסלול באורך t המכיל סדרה של מצבים ופעולות כך שהפעולה a_i מעבירה את הסוכן מהמצב s_i אל המצב s_{ti+1}, עד שנגיע למצב הסופי s_t.

מושגי יסוד של למידה עם חיזוקים:

סוכן: הישות בעולם שלנו שצריכה להחליט את הפעולות שעליה לבצע כדי להגיע אל היעד.

Reward: התגמול שהסוכן מקבל על פעולותיו השונות. המטרה בתכנון היא למקסם את התגמול המצטבר.

Markov Decision Process (MDP): תהליך סטוכסטי בזמן בדיד המתאר התקדמות של סוכן בין מצבים בסביבה מסוימת התקדמות התהליך תלויה בהסתברויות מעבר בין מצבים, אך גם בהחלטות המתקבלות ע"י סוכן.



דוגמה לתהליך MDP - בו המעבר ממצב s_i למצב s_j תלוי בהסתברויות המעבר המושרות מבחירת הפעולה a_i. שימו לב כי ההסתברויות תלויות אך ורק במצב הנוכחי ובפעולה שנבחרה - זוהי "הנחת המרקוביות".

תמצית מאמר:

מכיוון שהמאמר הנסקר הוא שדרוג של מאמר זה של סרגיי לוין האגדי ושותפיו, נתחיל את סקירתנו עם התיאור של הרעיונותיו העקריים ואז נמשיך עם הסבר על החידושים שהוצעו במאמר הנסקר. לנוחות נקרא למאמר זה המאמר המקורי. במאמר המקורי המחברים מנסים לפתור את בעיית תכנון המסלול לא באמצעות שיטות סטנדרטיות של RL החוזות מצב אחרי המצב אלא חוזים את כל המסלול יחד (trajectory). המסלול מוגדר כמערך (הסדר כמובן חשוב כאן)של זוגות (s_i, a_i),...N, i=0 כאשר s_i מסמן את הייצוג (שיכון או embedding) של המצב i ו- a_i היא הייצוג של הפעולה i. למעשה ניתן להתבונן בסדרה הזו בתור תמונה 2xN כאשר השורה הראשונה של "הפיקסלים" היא ייצוגי המצבים כאשר השנייה מהווה את ייצוגי הפעולות.

המחברים מציעים להגדיר את בעיית תכנון המסלול כבעיה בעיה גנרטיבית שמטרתה היא ליצור מסלולים "טובים" חדשים בהינתן דאטהסט של מסלולים בעלי תגמול כולל גבוה ומסלולים פחות טובים בעלי תגמול כולל נמוך יותר. כאמור במקום לבנות מסלול שלב אחרי שלב אנו מאמנים מודל לבנות את כולו כמקשה אחת. איך עושים זאת? כאמור בדומה למודלי דיפוזיה גנרטיבים מאמנים מודל היודע לנקות רעש במסלול המורעש בהדרגה. כלומר מרעישים את המסלולים במנות קטנות של רעש (איטרציות)

ומאמנים מודלי לחזות את הרעש המוסף בכל איטרציה. אז כאן עושים זאת לכל מסלול ובסוף המודל מייצר מסלול חדש מרעש טהור על ידי הורדה הדרגתית של הרעש ממנו.
במאמר **safediffuser** הכותבים ממשיכים את הקונספט צעד אחד קדימה - מעבר לתכנון המסלול, השיטה שהם מציעים מסוגלת לשלב אילוצים בתהליך התכנון. כלומר במאמר זה המטרה היא למנף גישה דיפוזיונית לבניית מסלול הסוכן במלאו **תוך כדי עמידה באילוצים הנדרשים.** שליטה זו חשובה מאוד בתרחישים בהם בטיחות היא קריטית - לא נרצה לייצר לרכב אוטונומי מסלול שמתעלם מתמרורים או נוסע נגד כיוון התנועה. הגישה נשענת על שילוב של צעדי "תיקון" במקביל לצעדי הדיפוזיה, שיובילו אותנו לתוצאות שבהכרח עונות על האילוץ תוך שמירה על קרבה ל-trajectory המתקבל ללא אכיפה של אילוצים (כמו שהוצע במאמר המקורי)..

תכנון באמצעות למידה

- בשנים האחרונות צצו מספר שיטות לתכנון מבוססות למידה, העיקריות בהן:

למידה מחיזוקים (Reinforcement Learning): בה מטרת האלגוריתם היא למצוא מדיניות שתמקסם אות Reward(חיזוק) מסוים.

- **למידה מחיקוי (Imitation Learning):** בעיית למידה מפוקחת (supervised learning) בה המודל מאומן לבנות מסלול דומה כמה שיותר למסלולים בדאטהסט הנתון לו.

ישנן מספר בעיות בשיטות אלו, כמו דרישה לאינטראקציות רבות עם הסביבה הנחוצות ללמידה של מדיניות טובה בלמידה מחיזוקים, יכולת הכללה בעייתית וקושי לגרום למודל לכבד אילוצים המערכת הרצוים. באפליקציות כמו נהיגה אוטונומית וניתוחים רפואיים לא נרצה להשתמש ברשת שמביאה תוצאות מעולות אם ישנו סיכון במקרי קצה מסוימים הרשת תחליט על צעד שבוודאות גבוהה יגרום לנזק. באלגוריתמים קלאסיים לרוב ניתן יחסית בקלות למנוע התנהגויות לא רצויות ע"י הגדרה ואכיפה של אילוצים רלוונטים (מהירות מקסימלית, איזורים אסורים, ועוד) אך ברשתות נוירונים זה עדיין די בעייתי.

למידה מחיזוקים (Reinforcement Learning)

בלמידה מחיזוקים הסוכן לומד מדיניות טובה תוך כדי אינטראקציה עם העולם, ודרך האינטראקציה יכול לנסות פעולות חדשות ולראות מה השפעתן על התגמול. בעיות RL בדרך מנוסחות בתור (Markov Decision Process (MDP - תהליך מרקובי בו המצב הבא ובתהליך תלוי אך ורק במצב הנוכחי ובפעולה הנוכחית. בהתאמה, כך גם מנוסחת המדיניות הנלמדת - בהינתן מצב מסוים המדיניות תחזיר את הפעולה הטובה ביותר (מבחינת התגמול הכולל) במצב זה.
כמו כן, ההחלטות מתבצעות בצורה סדרתית - בזמן הטסט הסוכן בכל נק' זמן יקבל החלטה על פעולה אחת, יבצע אותה, ורק אחרי שהגיע למצב הבא יחליט על הפעולה הבאה. סכמה זו אופיינית לאלגוריתמי Model free, שלא חוזים מראש איך הסביבה משתנה בעקבות הפעולה שנבחרה.
תהליך האימון מתבסס על שימוש אינטנסיבי בסימולטור, בו במשך מיליוני אינטראקציות של ניסוי וטעיה עם הסביבה הסוכן לומד את המדיניות האופטימלית - המיפוי בין המצב הנוכחי לפעולה שתניב את התגמול המצטבר המקסימלי.

Offline RL

אחת הבעיות המשמעותיות כיום בלמידה מחיזוקים בהקשרי רובוטיקה היא מציאת מדיניות(policy) אופטימלית על סמך דאטהסט קיים שנאסף מבעוד מועד, מבלי לבצע אינטראקציות נוספות עם הסביבה. בעיה זו צצה כאשר אינטראקציה עם הסביבה היא מאוד יקרה או מסוכנת, למשל שעות טיסה במזל"ט שעולות הרבה כסף או רכב אוטונומי שבשלב הלמידה יכול לסכן את שאר הנהגים אם ניתן לו לנסוע בעצמו לפני שהוא סיים את האימון.

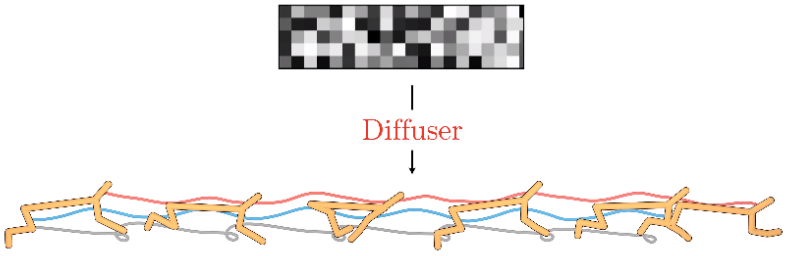
במסגרת offline RL צצים אתגרים חדשים - באימון אופליין אין יכולת לבצע אקספלורציה של פעולות חדשות. למשל, אם המדיניות הגיעה למצב מסוכן אין לנו אפשרות להתחיל לנחש פעולות אחרות ולבדוק אותן בסימולטור, כמו ב RL הסטנדרטי. במקרים כאלה הרשת תצטרך ללמוד לפעול אך ורק על פי הדאטהסט הקיים - ובמידה והמצב כלל לא קיים בדאטה, הרשת תצטרך ללמוד להכליל ממצבים דומים.

<u>מודלי דיפוזיה</u>

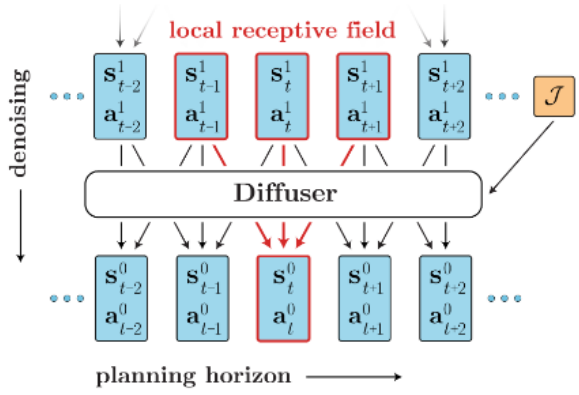
מודל דיפוזיה הוא סוג של מודל גנרטיבי שלומד ליצור פיסות דאטה חדשות מרעש (בד"כ גאוסי). מודלי דיפוזיה מאומנים באופן הבא: בתהליך הקדמי מוסיפים לפיסת דאטה מנות קטנות (איטרציות) של רעש עד שהדאטה הופכת להיות רעש טהור כאשר המודל מאומן לומד לנקות את רעש בהדרגתיות (מאיטרציה t+1 לאיטרציה t). במהלך אינפרנס (גנרוט) מתחילים מרעש טהור ומשתמשים במודל המאומן להסיר רעש בהדרגתיות עד קבלת פיסת הדאטה חדשה.

<u>תכנון בתור בעיית יצור תמונה</u>

אם נחשוב רגע על המסלול שאנו רוצים לייצר בתור רשימה של וקטורי ייצוג (embeddings) של מצב ופעולה, או בעצם מטריצה, ניתן להקביל את התהליך לייצור תמונה שאנחנו יודעים לעשות מעולה עם מודלי דיפוזיה.
למעשה בעיית בניית מסלול ניתן להציג בתור בעיית inpainting - בהינתן נקודת התחלה (העמודה הראשונה במטריצה, שמייצגת את s0), מצא את ההשלמה של המטריצה כך שנקבל מסלול רצוי. מספר העמודות במטריצה שנבקש הוא בעצם אורך המסלול, וכל עמודה שהמודל מייצר מייצגת מצב שנעבור בו ואת הפעולה שנעשה באותו זמן.
באותה מידה אם נרצה שהמסלול גם יסתיים במצב מסוים נבצע inpainting כאשר המצב הראשוני והסופי מקובעים והמודל משלים את היתר.



טכנית, המודל בו משתמשים במאמר מאוד דומה למודלי דיפוזיה המשמשים היום לגנרוט של סוגים רבים של דאטה ויזואלי כגון תמונות.



העובדה שהרשת המשערכת את הרעש הנוסף במודלי דיפוזיה משתמשת בפעולת קונבולוציה יוצרת כאן תבנית שונה מאוד מאלגוריתמי RL סטנדרטיים. כמו שהסברנו מקודם, באלגוריתמי RL לרוב ישנה הנחה מרקובית - כל מצב תלוי רק בקודמו (ולא בעבר הרחוק יותר)ובפעולה שנלקחה, ולכן התהליך מתואר כתהליך MDP. לעומת זאת, כאשר משתמשים במודלי דיפוזיה האינפורמציה הרלוונטית לכל מצב היא כל האינפורמציה שנכנסה ב-receptive field של הקונבולוציה, הכולל מצבים ופעולות שהיו לפני או אחרי המצב הנוכחי.
למה זה משנה בעצם? מודל הדיפוזיה מבצע אופטימיזציה על כל המסלול, מה שלרוב מביא תוצאות קוהרנטיות יותר בהשוואה למסלול המתקבל ב RL

Model Free או Model Based?

עוד אופיין חשוב של השיטה המוצעת הוא שבניגוד לאלגוריתמי RL אחרים הרשת לומדת באופן מרומז מודל של הסביבה. מודל הדיפוזיה מחזיר לנו כבר מההתחלה מסלול שלם שתוכנן t צעדים קדימה, לכן ניתן לומר שהרשת מבצעת תכנון אך במקביל לומד לחזות את הפעולות העתידיות. לעומת זאת באלגוריתמי RL המודל לומד לחזות אך ורק את הפעולה הכי טובה עבור כל מצב, אך לא תחזה לאן הפעולה תוביל אותנו.

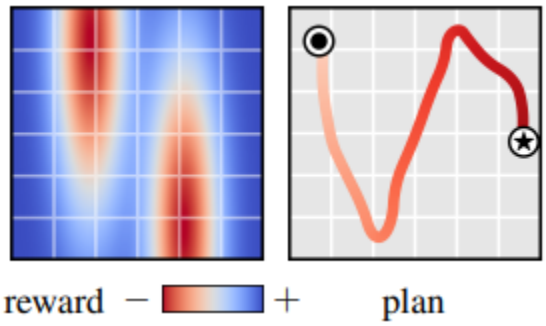
דאטהסט

מאיפה הדאטה? המודל מתאמן על אוסף מסלולים שבוצעו מראש והוקלטו. הדאטהסט יכול להיות אוסף הדגמות שבוצעו ע"י מומחה אנושי מבעוד מועד.
אופציה נוספת היא לשמור מסלולים שייצר אלגוריתם תכנון קלאסי. בד"כ אלגוריתמים קלאסיים יכולים להבטיח אופטימליות אך במחיר של זמן ריצה ארוך, לכן השימוש במודל דיפוזיה מאפשר לנו לשמור על זמן ריצה קבוע שמגיע לתוצאות קרובות מאוד למסלול האופטימלי.

אין דרישה על אורכי מסלול או איכותם. המודל יודע להכליל לאורכי מסלול שונים מסט האימון, מה שעדיין נחשב בעיה פתוחה באלגוריתמי state of the art offline RL.

שליטה באמצעות Reward

בשיטה שהגדרנו עד כה אין לנו שליטה על המסלול המתקבל מהמודל. הרשת מקבלת מסלולים מורעשים בתור מטריצה שמימדיה מייצגים את אורך המסלול שנרצה, וניתן במטריצה לקבוע מה יהיה המצב ההתחלתי והסופי, אבל זו כל השליטה שיש לנו על התהליך. בפועל במשימות מסוימות נרצה שליטה גדולה יותר על המסלול שנוצר - למשל ברכב אוטונומי נרצה מסלול שיצמד לנתיבים ולא יעלה על המדרכה. הדרך שבה המאמר מציע להתמודד עם אילוצים כאלו היא באמצעות reward.



בזמן ה-inference התגמול (reward) על מצבים מוזן למודל והוא לומד לבנות מסלול שממקסם את התגמול בזמן ביצוע ה inpainting. ניתן להרכיב את פונקציית התגמול איך שנרצה, מה שנותן הרבה גמישות בביצוע משימות שלא נראו בסט האימון.

אכיפת אילוצים בתהליך הדיפוזיה:

כעת נסביר על התרומה של מאמר ההמשך שאותו אנחנו סוקרים כאן:

SafeDiffuser: Safe Planning with Diffusion Probabilistic Models

התרומה העיקרית של המאמר היא הצגת מודלי safe diffusion, המאפשרים אכיפה של אילוצים על המסלולים המתקבלים ממודל הדיפוזיה.

<u>איפה צריך אכיפת אילוצים?</u>
ראשית, נתאר את הדוגמה הבאה:

נניח שנרצה לבנות מודל הדיפוזיה המתכנן מסלול נסיעה עבור רכב אוטונומי הנוסע בעיר. אספנו נתונים, אימנו את המודל, פרסנו אותו ברכב וקיבלנו מסלולים מעולים, אבל נתקלנו בבאג מסוכן - לפעמים בזמן פניה הרכב עולה על המדרכה.

קודם כל ננסה לענות על השאלה למה ש"באג" כזה יקרה מלכתחילה?
מתמטית, מודל הדיפוזיה יודע לבצע דגימה מהתפלגות. לאחר תהליך האימון המודל למד לייצר מסלול שיהיה סביר לתרחיש שביקשנו בהינתן המסלולים שראה בעבר. מכיוון שהתהליך הוא הסתברותי ייתכן שעבור תרחישים מסוימים שלא הופיעו הרבה בדאטה, המודל יחליט שעליה על המדרכה היא פעולה סבירה בהינתן המצבים הנתונים.

אבל למה בעצם לא להשתמש פונקציית התגמול שהצגנו מקודם? פתרון אפשרי אחד לסוגיה זו הוא

הקצאה של תגמול שלילי מאוד של מינוס אינסוף עבור איזור המדרכה. הוספת תגמול שכזה משנה את הניסוח המתמטי של מודל הדיפוזיה לדגימה מותנית - דגימה של מסלול בהינתן פונקציית התגמול נתונה. אבל גם אם נאמן את המודל באופן כזה בפועל לא ניתן "להתחייב" כי הוא לא יוביל את המכונית לאזור המדרכה. זה קורה עקב האופי ההסתברותי של יצירת המסלול באמצעות מודל דיפוזיה (הרי אנו מתחילים את בניית המסלול מדגימת הרעש, נכון). למעשה לא נוכל לשלול את הגעת המסלול לאזור האסור הזה ב 100% וזו בעיה מאוד רצינית.

אבל איך נרצה לאכוף אילוצים? הכותבים מציעים להשתמש ב- control barrier functions (CBFs). פונקציות CBF הן פונקציות המסמנות לנו את המצבים הבטוחים במרחב המצבים האפשריים שלנו. מתמטית נגדיר את הפונקציה h באופן הבא:

$$h(x_i) \geq 0 \Leftrightarrow x_i \, is \, safe$$

בתורת הבקרה המטרה לשימוש ב CBFs היא הוספת הפרעה ("תיקון") לפעולות של בקר מסוים כך שהמסלול המתקבל מהבקר יהיה בטיחותי באופן מוחלט ודטרמיניסטי, אך כמה שיותר קרוב למסלול המקורי שהיה אמור להתקבל ללא התיקון. הרעיון מאחורי השיטה הוא להשתמש במערכת קיימת שמספקת פתרונות טובים ולהוסיף לה את השינוי המינימלי כדי לפתור בעיות חדשות הכוללות אילוצים, מה שחוסך בתכנון של בקר חדש מאפס.
במקרה שלנו נרצה להכניס הפרעה ישירות בתהליך הדיפוזיה: בכל צעד דיפוזיה t נוסיף תיקון ל-Denoising של למטריצה שלנו כך שהתוצאה בצעד t+1 בהכרח תעמוד בדרישות הבטיחות. נקבל גם המסלול הסופי מובטח שיעמוד בדרישות הללו. השינוי שנכניס לצעד הדיפוזיה ימצא ע"י פתרון בעיית האופטימיזציה עם אילוצים (Quadratic Programming) הבאה:

$$\boldsymbol{u}^{j*} = \arg\min_{\boldsymbol{u}^j} ||\boldsymbol{u}^j - \frac{\boldsymbol{\tau}^j - \boldsymbol{\tau}^{j+1}}{\Delta\tau}||^2$$

תחת האילוץ שהמצב המתקבל מ u^{j*} הוא מבין המצבים הבטוחים, ז"א $h(\tau^{j+1}) \geq 0$.

הישגי מאמר:

המאמר מציג דרך לשלב מודלי דיפוזיה קיימים עם שיטות עדכניות מתורת הבקרה כדי לפתור בעיות תכנון תחת אילוצים. במאמר מוצגות התוצאות על מספר בנצ'מרקים מפורסמים מעולם הRL, ומראה תוצאות טובות תוך שמירה על אילוץ מסוים - למשל מניעת התנגשות בזרועות רובוטיות, הגדרת גובה קפיצה מקסימלי ברובוט hopper ועוד. לצפיה באנימציות אנו ממליצים לבקר את האתר.

באופן כללי, עולם התכנון חווה טרנד חדש של שימוש במודלי דיפוזיה לפתרון מגוון רחב של בעיות מורכבות במיוחד, כמו Multi Agent RL ו-offline RL על רובוטים פיזיים, ולמאמר זה בהחלט מגיע מקום של כבוד ביניהם.

#deepnightlearners

הפוסט נכתב על ידי יניב חסידוף ומיכאל (מייק) ארליכסון, PhD, Michael Erlihson

יניב עובד ברפאל בתור מהנדס אלגוריתמי ניווט. כמו כן סטודנט לתואר שני בתחום למידה עמוקה ורובוטיקה בטכניון.

מיכאל עובד בחברת הסייבר Salt Security בתור Principal Data Scientist. מיכאל חוקר ופועל בתחום הלמידה העמוקה, ולצד זאת מרצה ומנגיש את החומרים המדעיים לקהל הרחב.