

בינה מלאכותית – מבוא והצגת בעיות כגרפים

מאת ניר אדר (UnderWarrior)

כשאנשים שומעים בימינו את המונח "בינה מלאכותית" בהרבה מקרים עולה תמונה שמצוירת לנו בסרטי הקולנוע – אולי זו התמונה של ארנולד שוורצנגר משחק כרובוט מהעתיד, או של הרובוטים המאיימים מסדרת סרטי המטריקס.

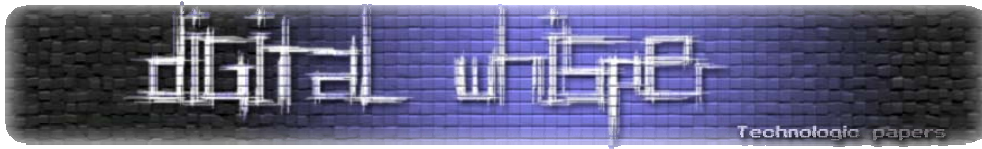
למרות שהתחום מתפתח ונחקר כל הזמן, אנחנו רחוקים מאוד מפיתוח מערכות שמזכירות את הרובוטים מהסרטים. עם זאת שימושים של הבינה המלאכותית אנו רואים במקומות רבים בעולם התוכנה: משחקי מחשב משתמשים בבינה מלאכותית כדי לדמות יריב המתמודד מול השחקן, מנועי חיפוש שולחים סוכנים המנסים להבין את הרשת ולעזור לנו לחפש מידע, תוכנות משרדיות שונות לומדות את ההרגלים שלנו ומתאימות את עצמן אלינו.

בסדרת המאמרים שמתחילה במאמר זה אני רוצה לדבר על תחום המחקר של הבינה המלאכותית, כמו שהוא בימים אלה. המאמרים מיועדים לקריאה עבור מתכנתים, אך לא בהכרח אקדמאים. אני אנסה להציג את העקרונות מאחורי האלגוריתמים הקיימים אך לא להכנס למתמטיקה הכבדה שמאחוריהם.

כדי לדבר על תחום הבינה המלאכותית נתחיל בהסבר מה זו בכלל בינה מלאכותית. בינה מלאכותית היא תחום מחקר במדעי המחשב, שהמטרה שלו היא פיתוח אלגוריתמים לתפיסה, הסקת מסקנות ולמידה, וזאת כדי לפתור בעיות מורכבות. שימו לב – הגדרה זו שונה משמעותית מהתפיסה הציורית שהסרטים מייצרים לנו. בעוד שהסרטים מציגים מכונות "חושבות", שניתן לקרוא להן אפילו אנושיות במובן מסויים, המחקר בתחום הבינה המלאכותית מתרכז ברובו במטרה שאפתנית הרבה פחות.

האם אי פעם יהיו מכונות שבאמת ידעו לחשוב? נושא זה מרתק את המדענים בכל העולם. המדען אלן טיורינג הציע ב-1950 מבחן שיגיד מתי מכונה תחשב לתבונית: מכונה תחשב לתבונית אם ייתן לאדם היושב בחדר סגור, לנהל שיחה באמצעות ממשק מחשב (Console) עם ישות שנמצאת בחדר השני, כאשר אותה ישות תהיה או אדם או מכונה, והאדם המשוחח לא יוכל לזהות האם מולו ניצב אדם או מכונה. תחרויות לא רשמיות, המכונות "תחרות טיורינג", נערכות מדי שנה כאשר המשתתפים מנסים להעמיד תוכנות שיעמדו במבחן. עד היום עוד לא נכתבה תוכנה שהצליחה לעבור את מבחן טיורינג.

נחזור לעולם המחקרי של המערכות הלומדות. אחד המושגים החשובים בעולם זה הוא מושג **הסוכן האינטליגנטי**. סוכן אינטליגנטי זו ישות התופסת את הסביבה שלה ופועלת עליה כדי להשיג מטרות שהוגדרו על ידי אדוניה. במשחק שחמט, למשל, הסוכן הוא המחשב המשחק מול השחקן. המטרה שלו היא לנצח את השחקן. סוכן שמנסה לפתור מבוך, המטרה שלו היא למצוא מסלול ליציאה (או אולי – את המסלול הקצר ביותר ליציאה). רובוט המחפש עצמים מסויימים בסביבתו הוא דוגמא נוספת לסוכן אינטליגנטי.



למה המושג של הסוכן האינטליגנטי כל כך חשוב? כי זהו הרכיב האינטליגנטי במערכת שלנו, ובשניה שתתפסו את הקונספט תוכלו לממש בינה מלאכותית מוגבלת בתוכנות שלכם, גם בלי להתעמק בכל התחום התיאורטי של הבינה המלאכותית במדעי המחשב.

נניח שיש לכם בעיה שברצונכם לפתור – לדוגמה הבעיה שהצגנו קודם – מציאת מסלול במבוך. ראשית אנחנו צריכים למצוא יצוג לעולם: נוכל לייצג את המבוך על ידי מערך – תא שיש בו 1 פירושו קיר שלא ניתן לעבור דרכו, תא שיש בו 0 זהו שביל שאפשר לעבור בו, וכן נתונים הקורדינטות של נקודת ההתחלה. הסוכן שלנו יהיה קטע קוד שיפעל כך:

כל עוד לא הגעת אל היציאה:

בצע צעד שיקרב אותך אל היציאה (צעד זה יכול להתבצע על ידי אלגוריתמים רבים ושונים)

ברגע זה הגדרנו סוכן אינטליגנטי שפותר את הבעיה – מציאת המסלול במבוך. האם הוא יצליח? תלוי באלגוריתם שנבחר כדי לבצע את הצעד. האם הוא יבצע את המטלה בזמן סביר? תלוי במבוך וכן תלוי באלגוריתם שנבחר. כאשר הקונספט של הסוכן האינטליגנטי ברור, נוכל לשפר את המיומנות שלנו בכתיבת תוכניות העוסקות בבינה מלאכותית על ידי לימוד של אלגוריתמים נוספים וטכניקות יעילות יותר.

הבינה המלאכותית עוסקת בתחומים שונים ובשאלות שונות:

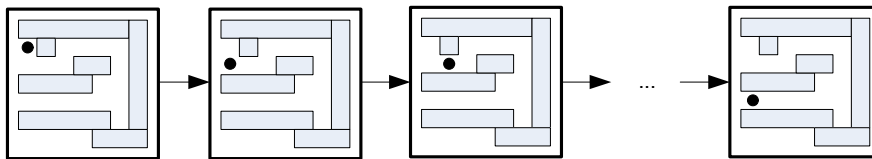
- ייצוג ידע – איך נייצג את הידע הרלוונטי לנו במחשב? במקרה של המבוך ייצגנו את המבוך על ידי מערך פשוט. לא תמיד התשובה תהיה כזו טריוויאלית. למשל – איך תייצגו במחשב את המשפט "אם יורד גשם, סימן שמעונן"?
- הסקת מסקנות – הסקת מסקנות חדשות מתוך ידע קיים.
- פתרון משחקים – למשל, תוכנת שחמט אינטליגנטית.
- למידה – איך נבנה סוכנים שישתפרו עם הזמן ועם ההתנסויות שעוברים עליהם?
- הבנת שפה טבעית – איך נגרום למחשב להבין טקסט כתוב? (חשוב במיוחד למנועי חיפוש). איך נגרום למחשב להבין דיבור? עדיין אין מחשב המסוגל לבצע משימות אלו באופן טוב.

נחזור אל הסוכן הפשוט שהגדרנו שמחפש דרכו במבוך, ונדבר טיפה על המימוש של הסוכן. מונח חשוב נוסף שיש צורך להציג הוא המונח **מצב המערכת**. מצב המערכת מתאר את המערכת ברגע נתון כלשהו. המטרה של הסוכן היא בעצם לעבור **ממצב התחלתי** – למשל, המצב בו הסוכן נמצא בנקודת ההתחלה של המבוך **למצב סופי** – הסוכן נמצא ביציאה מהמבוך.

בכל רגע נתון הסוכן בעצם מפעיל **אופרטור** על המצב. אופרטור הינו פעולה שהסוכן יכול להפעיל כדי להעביר את העולם ממצב למצב. אופרטור הוא פונקציה המקבלת מצב (המצב הנוכחי) ומחזירה מצב (המצב אחרי הפעלת הפעולה).

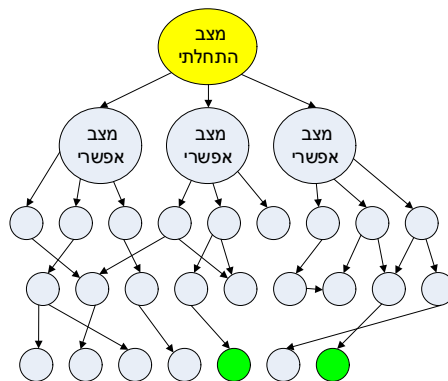
בהינתן בעיה אותה אנחנו רוצים לפתור, נפעל בצורה הבאה:

- נייצג את מצבי העולם האפשריים על ידי גרף מצבים.
- נייצג את הבעיה על ידי המצב הנוכחי ואת הפתרון הרצוי על ידי קבוצת מצבי מטרה.
- נפעיל אלגוריתם חיפוש למציאת מסלול בגרף המצבים מהמצב הנוכחי אל מצב מטרה.



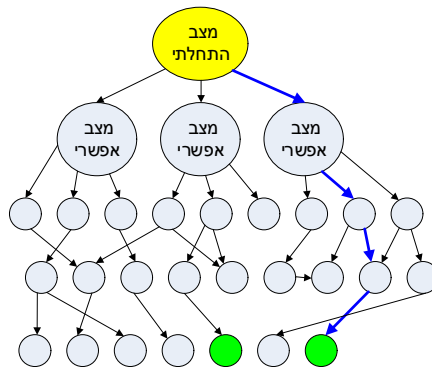
יצוג של המבוך. בכל מצב הסוכן נמצא במקום שונה במבוך

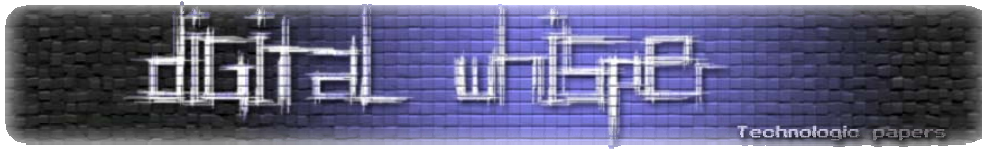
במקרה של המבוך הפשוט הזה הצעד בכל רגע היה התקדמות. במקרה של מבוך (או בעיה) מסובכת יותר נקבל גרף של מצבים:



כאשר המצבים המסומנים בירוק הינם מצבי המטרה.

הסוכן ימצא מסלול בין המצב ההתחלתי לאחד ממצבי המטרה:





כשנקבל בעיה שנרצה לפתור, נגדיר אותה על ידי גרף מצבים:

- נגדיר את קבוצת המצבים האפשריים (ואיך מייצגים אותם).
- הגדרת האופרטורים המעבירים ממצב למצב.
- בתוכנות מסובכות יותר נוכל להגדיר דברים נוספים – למשל מחיר על פעולות – יתכן שממצב מסויים אפשר לעבור לשני מצבים אחרים, אבל המחיר של מעבר אל כל אחד מהם שונה, והסוכן ינסה למצוא את המסלול הזול ביותר אל המטרה.

כמה נקודות על האלגוריתמים לחיפוש במרחב מצבים:

- האלגוריתמים מתחילים מהמצב ההתחלתי.
- בתחילה הגרף מוגדר רק בצורה לא מפורשת (כלומר, נתון לנו המצב ההתחלתי ואופרטורים המביאים אותנו למצבים הבאים המיידיים – אין לנו את כל הגרף בזכרון המחשב).
- אלגוריתמי החיפוש חושפים בהדרגה חלקים מהגרף והופכים אותו לגרף מפורש.
- **פתרון לבעיית חיפוש:** הפתרון לבעיית חיפוש היא סדרת אופרטורים המובילים מהמצב ההתחלתי למצב סופי.
- **פעולת פיתוח צומת:** הפעולה הבסיסית של אלגוריתמי החיפוש היא פעולת פיתוח צומת. פעולה זו מקבלת צומת ומחזירה את קבוצת הצמתים העוקבים.
- **אסטרטגיות חיפוש:** אסטרטגיית חיפוש מגדירה כיצד יש לחפש במרחב. כל אסטרטגיית חיפוש מבצעות סדרה של **פיתוחי צמתים**. האסטרטגיית נבדלות **בבחירת** הצומת הבא לפיתוח ובהחלטה אילו צמתים ישמרו בזיכרון.

כמו שניתן לראות, בכל אחד מנושאים אלה ניתן לדון רבות – איך נבחר את ייצוג המצבים? (ייצוג נכון של מצב הוא אחד מהגורמים הקריטיים לקביעת סיכויי ההצלחה של הסוכן!), איך הסוכן יחליט באיזה מהמצבים הבאים לבחור? איך נתמודד עם מגבלות של זכרון ומגבלות של זמן? במאמר הבא בסדרת המאמרים נציג מעט פרטים על מימוש של אלגוריתמים לחיפוש במרחב המצבים ונכיר את התכונות שלהם. אני מקווה שכבר אחרי מאמר זה השאלה "איך גורמים למחשב לפתור בעיות" ברורה לכם יותר.