

דף הבית



שם בית הספר: ישיבת תמר

שם העבודה: פרויקט למידת מכונה מכון שמיר

שם התלמיד: גלעד וינברגר

ת.ז. התלמיד: 216921403

שם המנחה: ליאל סנדלר, מירב צבירי

שם החלופה: למידת מכונה - deep learning

תאריך תחילת עבודה: 13/11/2024

תאריך הגשה: 1/6/2025

תוכן עניינים

דף הבית.....	1
תוכן עניינים.....	2
מבוא.....	3
רקע לפרויקט.....	3
קהל יעד.....	3
אופן פעולה של הפרויקט.....	3
תהליך המחקר.....	3
אתגרים מרכזיים.....	3
מבנה הפרויקט.....	4
איסוף, הכנה וניתוח הנתונים.....	4
בנייה ואימון המודל.....	4
יישום הפרויקט (ממשק משתמש).....	6

מבוא

רקע לפרויקט

הפרויקט הזה נוצר בשביל לעזור לחוקר ממכון שמיר, ד"ר ליאור גור. הוא חוקר עלי גפן וצריך לסווג אותם לפי מחלות. הפרויקט שלי עוזר לו לחשב כמה שטח לבן יש על כל עלה, וזה מקדם את המחקר שלו יותר מהר ויעיל.

קהל יעד

הפרויקט מיועד לחוקרים שחוקרים את אותו הדבר ורוצים לחסוך זמן בעבודה שלהם.

אופן פעולה של הפרויקט

המשתמש מעלה תמונה של עלה דרך ממשק משתמש, והמערכת מחזירה לו את כמות השטח הלבן על העלה.

תהליך המחקר

- המצב היום בשוק הוא שהרבה חוקרים לא יכולים להשתמש בכלי ניתוח ממוחשבים מתקדמים, אז הם עושים הרבה עבודה ידנית.
- החידוש בפרויקט הוא שהוא יכול לזהות כמה אחוז לבן יש על עלי גפן, וזה עוזר לזהות מחלה שנקראת כשותית הגפן.
- השתמשתי בטכנולוגיות שלא למדנו בבית הספר:
 - Django - זה Framework לפיתוח אתרי Full Stack בפיתוח, והוא עוזר ליצור אתרים בקלות. יש לו גם ממשק מנהל מובנה עם גישה למסד הנתונים.
 - יצירת מסד הנתונים נעשית בפיתוח, ומסד הנתונים הוא תיקייה של תמונות, ופרטי כל תמונה נתונים בשם שלה.

אתגרים מרכזיים

במהלך הפרויקט היו לי כמה אתגרים:

- למדתי על למידת מכונה.
- הייתי צריך להבין את המתמטיקה שמאחורי האלגוריתמים של למידת מכונה.

הפרויקט נותן פתרון לצורך בכלי סיווג ומיון שלא צריך הרבה עובדים, ויכול לעבוד מהר, יעיל ולחסוך זמן מחקר.

מבנה הפרויקט

איסוף, הכנה וניתוח הנתונים

מבנה הנתונים הוא תיקייה של כל התמונות השמורות בצורה שהאחוז הלבן של העלה נמצא בשם של הקובץ. לדוגמה: בקובץ "leaf_50.jpg" יהיו 50% לבן בתמונה. התמונות והערכים לדאטה נלקחו ממכון שמיר במחקרו של החוקר ליאור גור. אחרי יצירת התיקייה הראשונית של התמונות (כ-40 תמונות), כל תמונה משוכפלת 7 פעמים על ידי קוד פייתון - כל פעם עם הזזה ושיקוף בצורה שונה על מנת ליצור את כמות הדאטה המקסימלית האפשרית לאימון המודל. וכל התמונות משוכפלות לתיקייה נפרדת..

בנייה ואימון המודל

ארכיטקטורת רשת הניורונים (CNN - Convolutional Neural Network)

בפרויקט זה נבחרה ארכיטקטורה של רשת ניורונים מסוג CNN לצורך חיזוי אחוז השטח הלבן בעלי גפן. רשתות CNN הן סוג של רשתות ניורונים המצטיינות בעיבוד תמונות, והן בנויות משכבות שונות שתפקידן לחלץ מאפיינים רלוונטיים מהתמונה ולבצע את המשימה הסופית (במקרה שלנו, רגרסיה - חיזוי ערך מספרי רציף).

המודל הכללי הספציפי שבנית מורכב משני חלקים עיקריים:

1. שכבות לחילוץ מאפיינים (Features): חלק זה כולל סדרה של שכבות קונבולוציה (CNN) ושכבות איגום (Max Pooling Layers), ביניהן ממוקמות פונקציות הפעלה מסוג ReLU. שכבות הקונבולוציה לומדות פילטרים קטנים שמזהים דפוסים מקומיים בתמונה (כגון קצוות, פינות, מרקמים). שכבות ה-ReLU מוסיפות אי-ליניאריות לרשת, מה שמאפשר לה ללמוד פונקציות מורכבות יותר. שכבות האיגום מפחיתות את המימד המרחבי של מפות המאפיינים, ובכך מגבירות את העמידות של המודל לשינויים קטנים בתמונה (כגון תזוזה או שינוי קטן בקנה מידה).
2. שכבות רגרסיה (Regressor): חלק זה כולל סדרה של שכבות ליניאריות (Fully Connected Layers). לאחר ששכבות חילוץ המאפיינים מיצו את המידע הרלוונטי מהתמונה, הן משוטחות לווקטור חד-ממדי, אשר מוזן לשכבות הליניאריות. שכבות אלו מבצעות את החיזוי הסופי של אחוז השטח הלבן.

פירוט השכבות במודל:

1. שכבת קונבולוציה ראשונה: מקבלת תמונה צבעונית (3 ערוצים - RGB) ומפיקה 32 מפות מאפיינים. גודל המסנן הוא 3x3, הצעד הוא 1, והריפוד הוא 1 (שומר על גודל התמונה).
2. פונקציית הפעלה ReLU: שולטת על הפלט של שכבת הקונבולוציה הראשונה.
3. שכבת איגום ראשונה: מפחיתה את גודל מפות המאפיינים בחצי (גודל מסנן 2x2).
4. שכבת קונבולוציה שנייה: מקבלת 32 מפות מאפיינים ומפיקה 64 מפות מאפיינים (גודל מסנן 3x3, צעד 1, ריפוד 1).
5. פונקציית הפעלה ReLU: שולטת על הפלט של שכבת הקונבולוציה השנייה.

6. שכבת איגום שנייה: מפחיתה את גודל מפות המאפיינים בחצי.
7. שכבת קונבולוציה שלישית: מקבלת 64 מפות מאפיינים ומפיקה 128 מפות מאפיינים (גודל מסנן 3×3 , צעד 1, ריפוד 1).
8. פונקציית הפעלה ReLU: שולטת על הפלט של שכבת הקונבולוציה השלישית.
9. שכבת איגום שלישית: מפחיתה את גודל מפות המאפיינים בחצי.
10. שכבה ליניארית ראשונה: מקבלת את הפלט המשוטח של שכבות חילוץ המאפיינים (128 ערוצים בגודל 28×28 , סך הכל $28 * 28 * 128$ ניוונים) וממפה אותו ל-256 ניוונים.
11. פונקציית הפעלה ReLU: שולטת על הפלט של השכבה הליניארית הראשונה.
12. שכבה ליניארית שנייה (שכבת הפלט): מקבלת 256 ניוונים וממפה אותם לנירון יחיד, שמייצג את החיזוי של אחוז השטח הלבן (ערך בין 0 ל-1 שמייצג ערך בין 0 ל-100).

הסבר על שכבות ה-CNN:

- שכבת קונבולוציה (Conv2d): שכבה זו מבצעת פעולת קונבולוציה בין המסננים שלה (גרעינים) לבין התמונה (או מפת המאפיינים משכבה קודמת). כל מסנן לומד לזהות מאפיין ספציפי בתמונה, כגון קו אנכי, פינה, או מרקם מסוים. הפרמטרים של שכבה זו כוללים את מספר ערוצי הקלט, מספר ערוצי הפלט (מספר המסננים), גודל המסנן (בדרך כלל 3×3 או 5×5), הצעד (כמה המסנן זז בכל פעם), והריפוד (הוספת שוליים סביב התמונה כדי לשלוט על גודל הפלט).
- פונקציית הפעלה ReLU: פונקציה זו שולטת על הפלט של שכבת הקונבולוציה (או שכבה ליניארית) ומחליפה ערכים שליליים באפס. ערכים חיוביים נשארים כמו שהם.
- שכבת איגום (MaxPool2d): שכבה זו מפחיתה את גודל מפות המאפיינים על ידי בחירת הערך המקסימלי מתוך חלון קטן (בדרך כלל 2×2) שזז על פני המפה. פעולה זו מפחיתה את מספר הפרמטרים ברשת, מקלה על האימון, ומשפרת את העמידות של המודל לשינויים קטנים בתמונה.
- שכבה ליניארית (Linear): שכבה זו מבצעת טרנספורמציה ליניארית על הקלט שלה (כפל מטריצות וחבור וקטור). בשכבות האחרונות של רשת CNN, מפות המאפיינים המשוטחות מוזנחות לשכבות ליניאריות כדי לבצע את החיזוי הסופי.

בחירת ספריות תוכנה:

המודל נבנה בשפת Python תוך שימוש בספריית PyTorch. PyTorch מספקת כלים גמישים ויעילים לבנייה, אימון והערכה של מודלים של רשתות ניוונים. בנוסף, נעשה שימוש בספריות הבאות:

- NumPy: לביצוע פעולות מתמטיות בסיסיות ויעילות על מערכים.
- Matplotlib: להצגת גרפים של התקדמות האימון ותוצאות החיזוי.
- PIL (Pillow): לטיפול בתמונות (טעינה ושינוי גודל).
- torchvision.transforms: לביצוע טרנספורמציות שונות על התמונות לפני הכנסתן למודל (כגון שינוי גודל והמרה לטנסור).
- torch.utils.data.DataLoader ו-DataLoader: לניהול וטעינה יעילה של הנתונים במהלך האימון.

תהליך האימון

לאחר בניית ארכיטקטורת המודל, הוא מתאמן על מאגר הנתונים של תמונות עלי הגפן. תהליך האימון התבצע במשך מספר סבבים (epochs), כאשר בכל סבב המודל עובר על כל הנתונים. בכל איטרציה של האימון, המודל מקבל תמונה כקלט, מבצע חיזוי של אחוז השטח הלבן, ומשווה את החיזוי לערך האמיתי (הלייבל) שמופיע בשם הקובץ. ההפרש בין החיזוי לערך האמיתי מחושב באמצעות פונקציית Loss (במקרה זה, Mean Squared Error - MSE), ומטרת האימון היא למזער את ערך ה-Loss על ידי עדכון משקלי המודל באמצעות אלגוריתם אופטימיזציה (במקרה זה, Adam עם קצב למידה של 0.001). במהלך האימון, רמת הדיוק של המודל בחיזוי הולכת וגדלה, והשגיאה (Loss) הולכת וקטנה. תהליך האימון נמשך עד שמספר הסבבים המוגדר מראש (30 במקרה זה) מסתיים. במהלך האימון, נעשה שימוש בסרגל התקדמות (tqdm) כדי לעקוב אחר התקדמות האימון בכל epoch.

יישום הפרויקט (ממשק משתמש)

ממשק המשתמש בנוי בשפת פייתון עם Django Framework, כלי פשוט ונוח ליצירת אתרי Full-stack כולל חיבור מובנה מראש לקובץ SQLite המשמש כ-Database, וממשק ניהול מובנה הכולל גישה לכל הנתונים עם פעולות CRUD בסיסיות (Create, Read, Update, Delete).

בכניסה לכתובת האתר נכנסים לדף הבית עם אופציה להוספת תמונה, והחיזוי לתמונה מתבצע באמצעות המודל שנמצא בצד השרת של האתר. לאחר מכן התמונה עוברת את המודל, ותוצאת המודל מופקדת ב-100 (מספר בין 0 ל-1) < מספר בין 0 ל-100) ומוצגת על המסך.

מדריך למפתח

סיכום זום 12/12/2024:

המעבדה עוסקת במחלות צמחים (מחלות נוף, קרקע וכו'), הנגרמות ע"י פטריות ווירוסים. אנחנו נתעסק עם מחלות נוף (מחלות שנראות לעין)


בפרוייקט מתמקדים במחלת כשותית הגפן, הנגרמת על ידי דמוי-פטרייה כמו עובש-מים.

נקודות:

- לתקן את המבוא (לתאר את המצב הנוכחי ואת מטרת הפרוייקט והיתרונות בכך, לפני תיאור של למי הפרוייקט)

•

איפיון:

 MachinLearning-Project (1) (1) (1).pdf

מחווין:

 machinLearningProject_Finalt-5 (1).pdf