

פרויקט למידה חישובית

1. אלגוריתם להערכה:

Label Propagation

Reference - Iscen, Ahmet et al. "Label Propagation for Deep Semi-Supervised Learning." 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) (2019): 5065-5074.

תיאור האלגוריתם:

label propagation הוא אלגוריתם למידת מכונה מפקוח למחצה (Semi-Supervised learning). למידה מפוקחת למחצה מתייחסת לאלגוריתמים אשר מנסים לעשות שימוש בסט אימון מתויג ולא מתויג.

האינטואיציה של האלגוריתם label propagation הינה חיבור כל הצמתים שנוצרו בגרף על סמך המרחק שלהן, למשל מרחק אוקלידי. תווית אחת יכולה להפוך במהירות לדומיננטית בקבוצת צמתים אשר מחוברת בצפיפות, אך תתקשה באזור אשר מחובר בדלילות.

בתחילת האלגוריתם, תת קבוצה של צמתים אשר כוללת, תוויות מועברות לצמתים ללא, תוויות לאורך האלגוריתם. הצמתים נושאים, תווית המציינת את הקהילה אליה הם שייכים. התווית משתנה על סמך התוויות שיש לצמתים הסמוכים. כל צומת מאותחל עם תווית ייחודית, ואז מפוזרת ברשת. כתוצאה מכך, קבוצות אשר מחוברות בצפיפות מגיעות במהירות לתווית משותפת. האלגוריתם מגיע להתכנסות כאשר לכל צומת יש את התווית הרוב של שכניו. הוא מפסיק אם הגיע להתכנסות או הגיע למספר האיטרציות המוגדרות מראש.

יתרונות:

- מופקות תוויות על ידי התפשטות תוויות ולא על ידי תחזיות רשת.
- מבצעים label propagation על סט האימון במצב לא מקוון בעת אימון הרשת, כך ניתן יהיה להסיק מבלי לגשת לסט האימון המקורי.

חסרונות:

- לא מייצר פתרון ייחודי, אלא אוסף של פתרונות רבים.
- אם התיוג הראשוני שגוי, הדבר יכול להשפיע על תהליך התפשטות התוויות ותוויות שגויות עלולות להתפשט.

פסאודו קוד:

```
1. Initialize all labels of node in the network.
   1.1 for node x ->  $C_x(0) = x$ 
2. set iteration ->  $t = 1$ 
3. Randomly arrange the nodes in the network X.
4. For each  $x \in X$ 
   4.1 return the label that appears most frequently among neighbors.
5. If each node has a label that has the maximum number of neighbors
   5.1 Stop algorithm.
6. Else,  $t=t+1$  and go to (3).
```

המימוש שלנו:

שימוש בדאטה סט - Cifar100. חילקנו את הדאטה סט ל-20 דאטה סטים, כך שבכל דאטה סט יש 10 תוויות.

בכל הרצת Cross Validation, כאשר אנו מחלקות לסט אימון ובדיקה, אנו יוצרות מערך נתונים מתויגים ולא מתויגים עבור סט האימון על מנת להתאים את הנתונים למודל label propagate. אימון מודל label propagate על סט האימון, לאחר מכן אנו משתמשות בתוויות שהתקבלו מאימון המודל ובפרמטרים הטובים ביותר שהתקבלו מה-hyperparameters לאימון רשת ה-CNN ולהערכה.

רשת ה-CNN שלנו בנויה מ-13 שכבות ושימוש באופטימיזציה של SGD בהתאם למאמר.

2. הצעת שיפור לאלגוריתם:

במאמר נאמר שהם אינם משתמשים בנרמול עבור התמונות ולכן הם מקבלים ביצועים גרועים יותר מעבודות קודמות, לכן נשתמש בנורמליזציה עבור סט האימון והבדיקה על מנת לבדוק האם נקבל ביצועים טובים יותר. השתמשנו בפונקציה אשר מנרמלת את סט האימון והבדיקה לפי ממוצע וסטיית תקן של הרשומות.

3. בחירת אלגוריתם להשוואה:

מודל למידה עמוקה CNN עם VGG-19. השתמשנו עם ארכיטקטורת CNN של מודל מאומן מראש VGG-19 אשר נחשבת למודל ראייה מצוין עבור דאטה סט של תמונות, טענו גרסה מאומנת מראש של הרשת שהוכשרה על יותר ממיליון תמונות ממאגר ImageNet. במודל זה השתמשנו באופטימיזציה Adam וב-hyperparameters.

4. הערכת ביצועי המודלים:

השתמשנו ב-10 fold cross validation חיצוני כך שבכל fold אנו עושים חלוקה מחדש לסט אימון ומבחן.

בלולאה פנימית השתמשנו ב-3 fold cross validation על מנת לבצע אופטימיזציה לפרמטר ה-learning rate. עשינו hyperparameters באמצעות Random search.

Learning_rate: [0.01,0.001,0.0001]

מדדים:

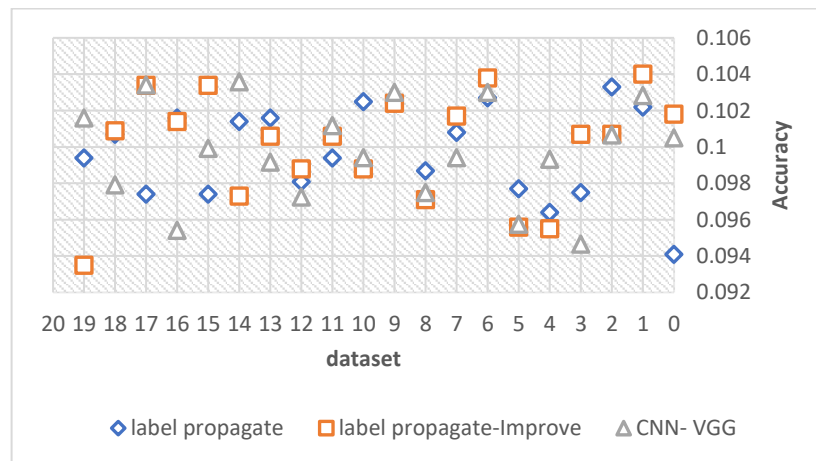
באמצעות המדדים הבאים הערכנו את ביצועי האלגוריתמים:

- (1) Accuracy - דיוק המודל
- (2) TPR
- (3) FPR
- (4) Precision
- (5) AUC - שטח תחת עקומת ROC
- (6) PR-Curve - שטח תחת Precision-Recall
- (7) זמן אימון
- (8) זמן ניבוי עבור 1000 תצפיות

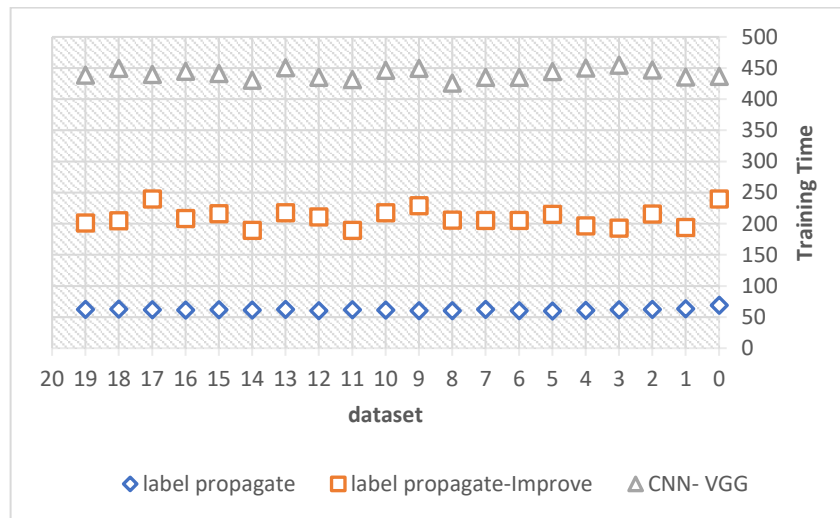
תוצאות:

בחנו את ממוצע המדדים אשר התקבלו עבור כל דאטה סט. (קבצי האקסל מצורפים לעבודה) נראה כמה תרשמים להשוואה בין המודלים השונים:

-Accuracy



-Training time



מסקנות:

- נראה כי לא קיים שינוי משמעותי בביצועים עבור שלושת האלגוריתמים. כנראה שאם היינו מגדילים את סט הנתונים אז היינו מקבלות תוצאות טובות יותר, הרי שאם מגדילים את סט האימון ביצועי המודל משתפרים וחיזוי המודל טוב יותר. היה לנו קושי בהתאמת סט הנתונים לעבודה ולכן לא הצלחנו לאתר סט נתונים גדול יותר שאיתו נוכל לעבוד ולחלק ל-20 דאטה סטים.
- זמן האימון של האלגוריתם השלישי (CNN- VGG) שלקחנו להשוואה היה ארוך בצורה משמעותית לעומת שאר האלגוריתמים.

5. בדיקת מובהקות סטטיסטית של התוצאות

מבחן פרידמן הינו מבחן סטטיסטי ללא פרמטרים. כדי לקבוע את אם ישנם הבדלים מובהקים סטטיסטית בין המודלים בהם השתמשנו נשתמש בממדד הביצוע – AUC.

תוצאות מבחן פרידמן:

`FriedmanchisquareResult(statistic=13.585714285714289, pvalue=0.8072844319082226)`

$\alpha = 0.05$, מכאן ש- $pvalue > \alpha$ ולכן לא נדחה את השערת האפס. השערת האפס של המבחן אומרת כי כל ה-classifiers זהים וכי לא קיים הבדל ביניהם.

קישור לעבודה ב-GitHub:

https://github.com/lironoskar/computational_learning_project