עיבוד תמונה – תרגיל 5

אוריין חסידים ~ oryan.hassidim ~ 319131579

# מבוא

בתרגיל זה, נתרגל שימוש ברשתות נוירונים. המשימה אותה נבצע היא שימוש בהיפוך Generative Adversarial Networks (GAN inverting), כדי להחזיר תמונות עם גריעה מסויימת לתמונה באיכות המקורית. באופן ספיציפי, סוגי הגריעה איתם נעבוד הן תמונה חסרה על פי מסכה ותמונה בגווני אפור.

נקבל רשת שמאומנת לחולל (generate) תמונות של פרצופים מרעש התחלתי, וננסה למצוא את הרעשים שמביאים אותנו לתמונה הרצויה. נריץ כל פעם את הרעש במחולל, נמדוד את המרחק מהתמונה שקיבלנו כאשר מופעלת עליה הגריעה הרצויה, לתמונה אותה קיבלנו. כל פעם נשפר את הרעש לכיוון הרצוי כך שנתכנס לרעש האופטימלי שמוביל לתמונה הקרובה ביותר לתמונה הרצויה.

את המרחק בין תמונות לא נחשב כמרחק אוקלידי בין פיקסלים אלא נשתמש ברשת נוספת שתמפה כל תמונה של פנים לוקטור פיצ'רים, ונחשב על פיו.

אחרון והכי פחות משמעותי (Last And Least) (בעיני לפחות), הוא כמובן החלק הטכני של הפעלת הגריעה על הפלט של הרשת – חיתוך תמונה ע"פ מסכה, והמרת תמונה צבעונית לתמונה בגווני אפור.

# אלגוריתם

## המרת תמונה צבעונית לגווני אפור

השתמשנו בנוסחה מהתיעוד הרשמי של OpenCV: .

## חיתוך תמונה ע"פ מסכה

יצרנו עותק של התמונה, ובאינדקסים שרצינו לחתוך לפי המסכה שמנו (או , תלוי בקונבנציה).

## היפוך תמונה

* טעינת הרשת המאומנת stylegan2-ada מהאתר של Nvidia, ואתחול הסביבה להשתמש בכוח החישוב של ה-GPU;
* טעינת התמונה מהקובץ והתאמתה לממדים המתאימים לרשת;
* גריעת התמונה ע"פ אפשרות הגריעה שניתנה כפרמטר;
* הרצת האלגוריתם העיקרי – אופטימיזציה של וקטור קלט לרשת כך שנקבל תמונה קרובה ככל האפשר לתמונה הרצויה (לפני הגריעה), עבור היפר פרמטרים מתאימים. האלגוריתם יפורט להלן.
* שמירת התמונה ווקטור הרעש (וקטור הקלט) האחרונים מתהליך האופטימיזציה.

## אופטימיזציית וקטור הקלט לרשת

* אתחול:
  + דגימת וקטור z מהתפלגות נורמלית כוקטור במרחב Latent Space (מרחב Z);
  + מיפוי וקטור z לוקטור w ממרחב Intermediate Latent Space (מרחב W);
  + מציאת הממוצע וסטיית התקן וקטור z.
  + שמירת הרפרנסים לכל שכבות הרעש ברשת, ואתחול שלהם לרעש רנדומלי. בהמשך התהליך גם הם יעברו אופטימיזציה (רגולציה);
  + טעינת מזהה פיצ'רים VGG16 שנשתמש בו למדידת המרחק בין התמונות.
  + מציאת וקטור הפיצ'רים של התמונה הרצויה ע"פ VGG16.
  + אתחול וקטור w\_opt שאותו ננסה להביא לאופטימום.
  + אתחול מערך בו נשמור את w\_opt בכל שלב.
  + אתחול אופטימייזר Adam (שנראה שהוא אופטימייזר פופולרי ומאוד מרשים) שיעזור לנו בהמשך לעדכן את w\_opt ואת הרעשים.
* לולאה עבור כל שלב למידה:
  + חישוב קצב הלמידה לשלב זה (ע"פ הפרמטרים לפונקצייה ופונקציית רמפה) ועדכון האופטימייזר לקצב הלמידה הזה;
  + חישוב פלט הרשת עבור w\_opt הנוכחי;
  + גריעת התמונה לפי הפרמטר שניתן (ללא גריעה, גווני אפור, וחיתוך לפי מסכה) ושמירה של התמונה המקורית והתמונה הגרועה;
  + הצגת התמונה הנוכחית כל 20 צעדים, שמירת התמונה הנוכחית לקובץ כל 100 צעדים;
  + חישוב ה-reg\_loss כדי שנוכל למנוע התאמת יתר (עיין ‎נספח 1 מתחת, תמונה להמחשת Overfitting), ע"י סכימת ריבועי הקורלציה של הרעש עם הזזתו בשורה והזזתו בעמודה, וביצוע average pooling כפי שלמדנו בהרצאה, עד שממדי הרעש קטנים מ-8;
  + בדיקת וקטור הפיצ'רים של התמונה הנוכחית ע"פ VGG16 שהורדנו מקודם.
  + חישוב ה-loss – סכום של:
    - המרחק בין הוקטורים של ייצוגי התמונות ע"פ VGG16,
    - ה-reg\_loss שחישבנו מקודם כפול פקטור שניתן כפרמטר,
    - המרחק (הפעם L1) מהממוצע של מרחב ה-W, שחישבנו בשלב האתחול.
  + ביצוע שלב למידה של צעד אחד ב-Gradient Descent ע"י האופטימייזר;
  + הוספת ה-w\_opt הנוכחי למערך המוחזר;
  + נרמול מטריצות הרעש.
* החזרת מערך ה-w\_opt על שלביו.

# תוצאות

## 3.1 יישור תמונה

מימין התמונה המקורית, ומשמאל – החתוכה והמיושרת.

 

## 3.2 היפוך GAN

|  |  |  |
| --- | --- | --- |
|  |  |  |
| שלב 0/1000  סוריאליסטי משהו | שלב 200/1000  הוא קלט שיש הרבה שיער למטה | שלב 300/100  הופה צמח לי זקן |
|  |  |  |
| שלב 600/1000  יש גם כיפה ומשענת | שלב 1000/1000  וואלה מרשים. יש דמיון | התמונה המקורית |

להלן תרשים של עקומת הלמידה (הסרגל השמאלי מתאים ל-loss, הימני לאחרים):

## 3.3.2 צביעת תמונה

די ניסיתי לעקוב אחרי ההנחיות בהערות ולעשות מה שכתוב שם... לא הגדלתי ראש... גם ניסיתי שוב להשתמש בתמונה שלי, אבל הפעם...

|  |  |  |
| --- | --- | --- |
|  |  |  |
| מקורי | גווני אפור | צבוע |

ומסתבר שטעיתי בקוד כך שהוא ניסה להגיע להתכנס לתמונה האפורה. מרשים. ממשיכים לנסות:

|  |  |  |
| --- | --- | --- |
|  |  |  |
| הפעם הקוד כתוב נכון אבל התוצאה אדומה מדי | הפעם עם  initial\_noise\_factor=0.1  regularize\_noise\_weight=150000 | הפעם גם הוספתי  noise\_ramp\_length=0.5  מחריד. |

נראה שהזקן ו/או הכובע ו/או העיניים למטה (שהיו בכוונה כדי לבדוק) מאוד מפריעות לו. אז הלכתי על פרצוף שהוא בטוח בטוח יכיר – פרצוף מ-[thispersondoesnotexist.com (1024×1024)](https://thispersondoesnotexist.com/). הפעם התוצאה יפה דווקא:

|  |  |  |
| --- | --- | --- |
|  |  |  |
| מקורי | גווני אפור | צבוע |

הטיה מגדרית כבר אמרנו?

## 3.3.3 השלמת תמונה

גם פה, בקוד לגריעת התמונה לא השקענו המון מחשבה ודי הלכנו ישר... ניסיתי ללכת על משהו מרשים ולראות אם הוא מצליח להשלים חצי פנים:

|  |  |  |
| --- | --- | --- |
| תמונה שמכילה לבוש, זקן אדם, פני אדם, שפם  התיאור נוצר באופן אוטומטי |  |  |
| מקורי | חתוך | מושלם מחדש. נורא ואיום. |

נראה שהציפייה הייתה מוגזמת😅... עם דרישה יותר צנועה הגעתי לתוצאה יותר טובה, ועדיין לא ציפיתי שהקוד יצליח להשתלט על הטקסט, אבל הוא הבין פחות או יותר את הכובע אע"פ עדיין יש גליצ'...

|  |  |  |
| --- | --- | --- |
|  |  |  |
| מקורי | חתוך | מושלם מחדש |

# סיכום

הממצא המשמעותי ביותר עבורי הוא קודם כל שרוב הקסם ותהליכי הווּדוּ היו בעצם שורת קוד פשוטה למדי של קריאה ל-API מוכן – כמו בהרצת האופטימייזר שעשה את כל ה-GD בשבילנו, ובקבלת תמונה מוקטור z, וקבלת וקטור מ-VGG16.

בנוסף, כמובן התוצאות המרשימות אליהן המודל הצליח להגיע אכן ראויות לציון. כמו כן בעזרת עוד מודלים של בינה מלאכותית הצלחתי סה"כ אפילו להבין ברמה מרשימה למדי את הקוד המסופק, שאני חושב שזה הישג נאה.

מאידך, אכן ראינו שהתוצאות כן מוגבלות, ושישנה הטיה לגבי פנים מסגנון שהוא פחות מכיר, שזה משהו שאנחנו כבר יודעים על בכיות הטיה של בינה מלאכותית עוד מתחילת דרכה כמו בעיית מצלמות האבטחה ([מערכות לזיהוי פנים הובילו לשורה של מעצרי שווא בארה"ב (ynet.co.il)](https://www.ynet.co.il/digital/technews/article/r1vx001uk6))

תודה רבה על כל הקורס המעניין הזה!♥🤗

# נספחים

## תמונה להמחשת Overfitting



מ-[Oversimplifying Overfitting | Bed design, Weird beds, Unique bed design (pinterest.com)](https://www.pinterest.com/pin/684758318329981037/)

כן, אני יודע שזה מוכר ממש, סורי, אבל הייתי ממש חייב😅.