רשתות נוירונים לתמונות תרגיל 3

יואב שפירא 312492838

23 במאי 2023

חלק מעשי

שאלה 1 $Loss\ Saturation$

ארכיטקטורות:

- $Generator$: מורכב מ4 שכבות של $Strided\ Transpose\ Convolution$ עם קרנל בגודל 3. אחרי כל שכבה מה3 השכבות הראשונות יש שכבת $BatchNormalization$ ואקטיבציה $ReLU$. לאחר השכבה הרביעית אין $BatchNormalization$ והאקטיבציה היא $Sigmoid$.
  בשכבה הראשונה, האינפוט (וקטור בגודל 100) ממופה ל256 ערוצים לתמונה בגודל $4x4$.
  בשכבה השנייה ממופה ל128 ערוצים, בשלישית ל64 ערוצים וברביעית לערוץ 1.

```
Generator(
  (seq): Sequential(
    (0): ConvTranspose2d(100, 256, kernel_size=(3, 3), stride=(2, 2), bias=False)
    (1): BatchNorm2d(256, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (2): ReLU(inplace=True)
    (3): ConvTranspose2d(256, 128, kernel_size=(3, 3), stride=(2, 2), bias=False)
    (4): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (5): ReLU(inplace=True)
    (6): ConvTranspose2d(128, 64, kernel_size=(3, 3), stride=(2, 2), bias=False)
    (7): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (8): ReLU(inplace=True)
    (9): ConvTranspose2d(64, 1, kernel_size=(3, 3), stride=(2, 2), padding=(2, 2), output_padding=(1, 1), bias
    (10): Sigmoid()
  )
)
```

- $Discriminator$: מורכב מ3 שכבות של $Strided\ Convolution$, עם קרנל בגודל 4, ולבסוף שכבת FC שממפה לאאוטפוט - מספר יחיד שהוא ההתסברות שהתמונה היא אמיתית. אחרי כל שכבת קונובלוציה יש $BacthNormalization$ ואקטיבציה $LeakyReLU$ עם שיפוע של 0.2 לערכים שליליים. אחרי שכבת הFC יש שכבת אקטיבציה $Sigmoid$ כדי למפות את הערך האחרון להסתברות.
  בשכבה הראשונה האינפוט (תמונה בגודל $28x28$) ממופה ל32 ערוצים, בשכבה השנייה ל64 ערוצים, ובשכבה השלישית 128 ערוצים. לבסוף הFCממפה לערך יחיד.

```
Discriminator(
  (seq): Sequential(
    (0): Conv2d(1, 32, kernel_size=(4, 4), stride=(2, 2), padding=(1, 1))
    (1): BatchNorm2d(32, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (2): LeakyReLU(negative_slope=0.2, inplace=True)
    (3): Conv2d(32, 64, kernel_size=(4, 4), stride=(2, 2), padding=(1, 1))
    (4): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (5): LeakyReLU(negative_slope=0.2, inplace=True)
    (6): Conv2d(64, 128, kernel_size=(4, 4), stride=(2, 2), padding=(1, 1))
    (7): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True, track_running_stats=True)
    (8): LeakyReLU(negative_slope=0.2, inplace=True)
    (9): FlattenBatch()
    (10): Linear(in_features=1152, out_features=1, bias=True)
    (11): Sigmoid()
  )
```

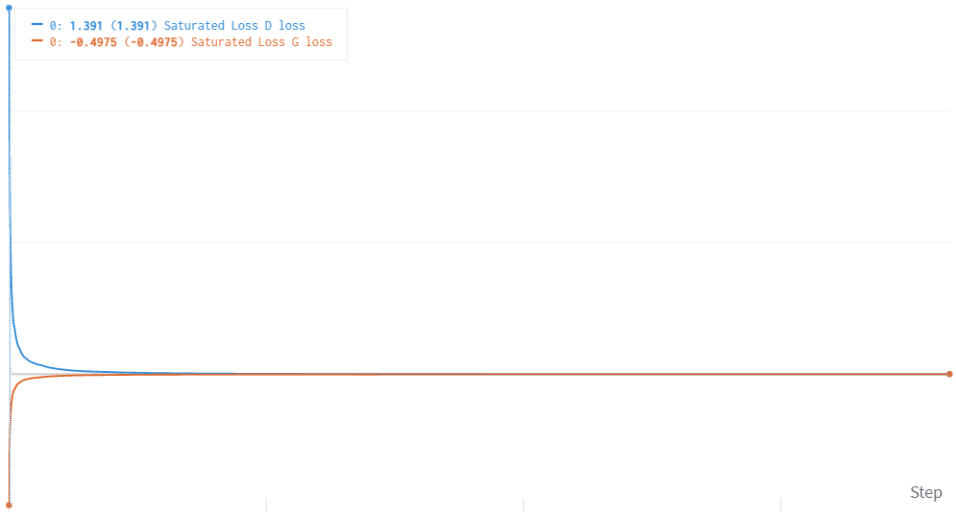
הבחירה לבנות את D קטן יותר מG נבעה מניסוי ותעייה: עבור D גדול יותר, המודל הגיע למצב $Mode\ Collapse$ לאחר מספר אפוקים של אימון. החלשה של D גרמה לו להתכנס לאט יותר, וכך אפשרה ללמידה להמשיך לאורך יותר זמן ללא $Mode\ Collapse$.

אימון:

השתמשתי באופטימייזר $Adam$ עבור שני המודלים, עם רגולריזציה (נורמת $l2$) עם ערך $\lambda = 0.5$.
אימנתי ב10 אפוקים, עם $lr = 0.002$.
על כל 3 באצ'ים של אימון לD אימנתי עם באץ' אחד את G.
הדאאט עליו אימנתי את המודלים הוא כל הדאטאסט של $MNIST$, גם הטריין וגם הטסט.

שאלה 1 פונקציות $Loss$:

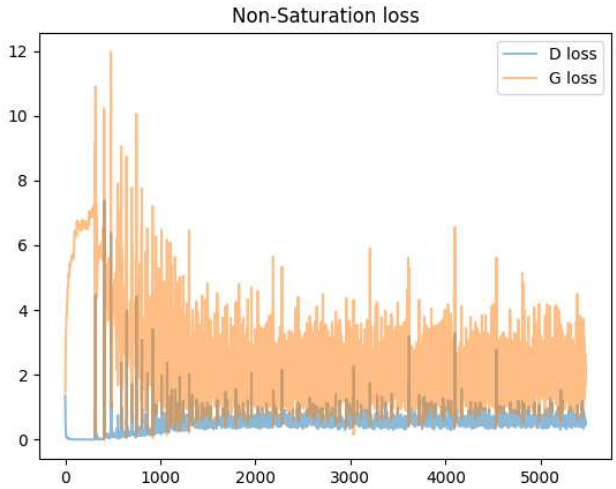
1. $Original\ GAN\ Cross-Entropy$: בשביל ה$loss$ הזה השתמשתי ב$criterion =$ $BCELoss$ כפונקציית ה$loss$ של D. כעת, G צריך למזער את השגיאה של $log\left(1 - D\left(G\left(z\right)\right)\right)$. כדי להשיג את זה השתמשתי באותה $BCELoss$ עם הפרדיקציות של D על התמונות המזוייפות, ועם $response$ של "תמונות מזוייפות", ומכיוון שזה $Binary\ Cross\ Entropy$ (מחושב על שני ערכים בלבד) נשיג את האפקט הרצוי.
הגרף של ה$Losses$ של שני המודלים מוצג כאן, בכתום G ובכחול D (הערכים במלבן הם ההתחלתיים, ואפשר לראות שהגרדיאנטים מתאפסים מהר):



התוצאות שהמודל הזה הניב הן תוצאות של רעש: G לא הצליח לייצר תמונות כי בכלל לא היה לא זמן ללמוד. לא צרפתי תמונה זה לא מעניין זה פשוט רעש.

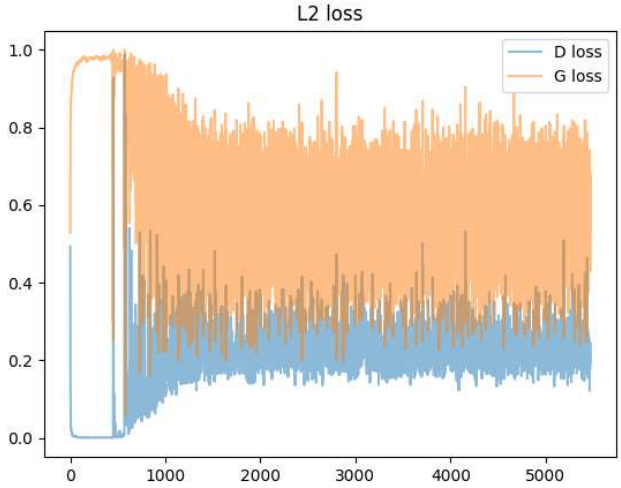
2. $Non-Saturatin\ Loss$: גם כאן השתמשתי ב$BCELoss$ כ$criterion$, רק שהפעם G צריך למקסם את $log\left(D\left(G(z)\right)\right)$, כלומר למזער את $-log\left(D\left(G(z)\right)\right)$, וכאן האפקט מתקבל על ידי מילוי הוקטור ה$response$ בלייבלים של "תמונות אמיתיות".

בגרפים של ה$Losses$ אפשר לראות שיש התכנסות למצב מסוים - המצב של $Nash\ Equalibirium$ - המודל D מאוד מוצלח בהתחלה, ולאט לאט הוא שוגה, והמודל G בדיוק הפוך, עד ששניהם יחסית מתאזנים כלומר מצליחים 'לבלבל אחד את השני'. בשיווי משקל אולטימטיבי, הדיוק של D יהיה 0.5, וניתן לראות שהוא מתקרב לשם.



התוצאות של המודל: לאחר 10 אפוקים G הצליח לייצר תמונות שנראות די טוב:

3. $L2\ Loss$: השתמשתי ב$MSELoss$, ובחישוב ה$loss$ של G שלחתי את הפרדיקציות של D על תמונות מזוייפות מG, עם לייבלים של "תמונות אמיתיות".
בגרף הזה אפילו יותר קל לראות את הכיוון לשיווי משקל נאש, כי השגיאה של שני המודלים חסומה בין 0 ל1:



התוצאות לא רעות:

פונקציית ה$Loss$ שהגיעה לסטורציה היא הפונקצייה הראשונה - המקורית. זה קורה בגלל שהנגזרת במקרה הזה $-\frac{1}{1-D(G(z))}$, צריך לשים לב שבשלבים ההתחלתיים G מייצר רק תמונות מזוייפות - ולD קל להבין את זה מאוד מהר. לכן, הגרדיאנטים בהתחלה יהיו מאוד קרובים ל-1, לעומת מצב של $Non-Saturation$ ובו הנגזרת שווה ל$-\frac{1}{D(G(z))}$, ולכן הגרדיאנטים בהתחלה יהיו קרובים ל$-\infty$. מכיוון ש$-\infty$ הרבה יותר רחוק מ0 מאשר -1, אנחנו מקבלים שעבור ה$loss$ הראשון הגרדיאנטים נעלמים מאוד מהר לעומת ה$loss$ השני. המצב של סטורציה הוא באמת מצב של $Vanishing\ Gradients$ בשלב התחלתי באימון של G.

שאלה 2 $Model\ Inversion$**:**

בשאלה זו לקחתי את המודל G שאומן עם $Non-Saturation\ Loss$, והקפאתי אותו בעזרת $eval()$ כדי שלא יתעדכן במהלך האופטימיזציה. הגרלתי z וקטור רעש, והגדרתי

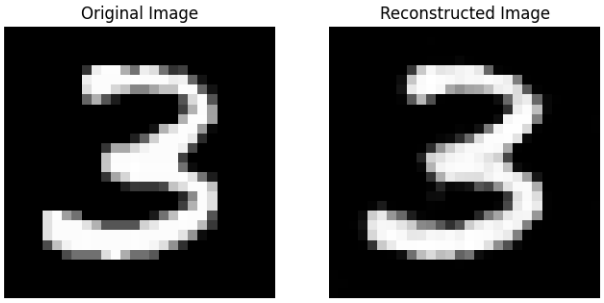
$Optimizer = Adam$ עם הפרמטרים של z. בעזרת $criterion = MSELoss$ חישבתי את ה$Loss$ של $|G(z) - I|$ עבור תמונה אקראית I מתוך הדאטאסט. לאחר 1000 איטרציות של GD קיבלתי z שמייצר את התמונה המקורית I יחסית טוב.

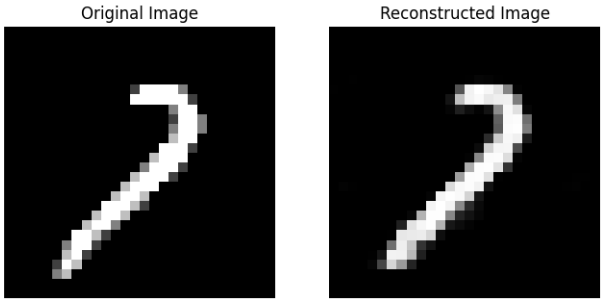
פיצ'רים ששוחזרו בהצלחה: המבנה הכללי של התמונה - התמונות המשוחזרות הגיעו לתוצאות כמעט מדוייקות מבחינת הצורה של הספרה המקורית.

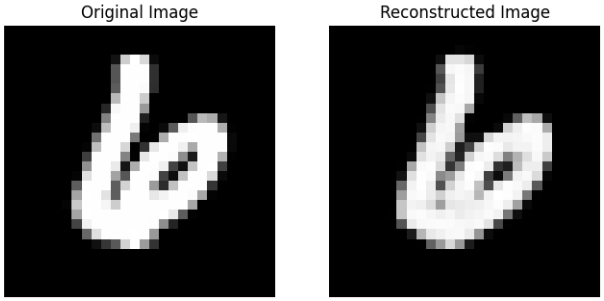
פיצ'רים שלא שוחזרו בהצלחה: הקצוות, הפרטים הקטנים - אפשר לראות בדוגמאות למטה, שהשוליים של הספרות מעט דהויות. בנוסף, זויות חדות היו נקודת מכשול מבחינת שחזור.

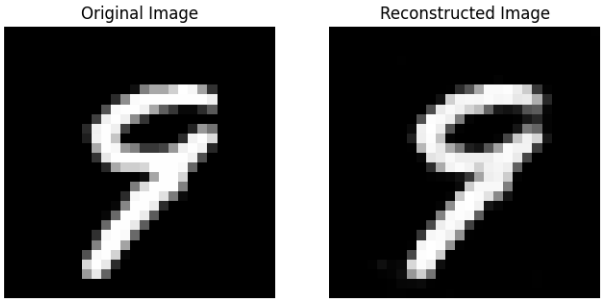
זה הגיוני כי הGAN הזה אומן משלב התחלתי, ובשלב ההתחלתי D מתבסס על פריטים מאוד גסים כדי ליצור קלסיפיקציה, ולכן G מייצר תמונות שהפרטים הגסים שלהם יטעו את D. בשלב מתקדם יותר, תאורטית היינו יכולים להמשיך לאמן את G עם מפריד D' שהוא יותר חזק מD ויודע להבחין בפרטים ספציפיים יותר כדי לגרום לG ללמוד את הפרטים הקטנים בתמונה.

דוגמאות:



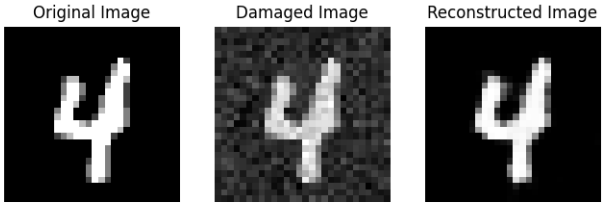


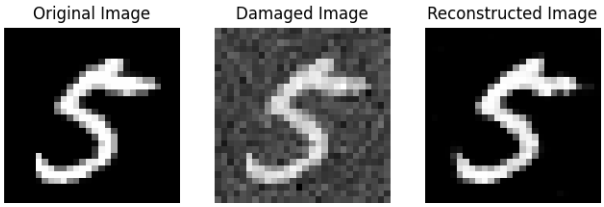


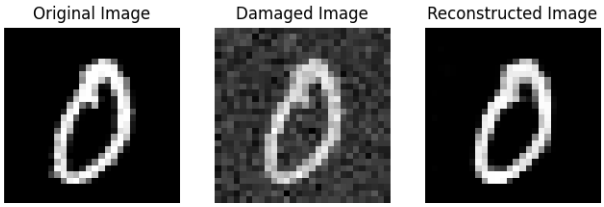


שאלה 3 $Image\ Restoration$**:**

1. $Denoising$: בשביל משימה זו בחרתי את פונקציית לוס $L2$, ואלה התוצאות:

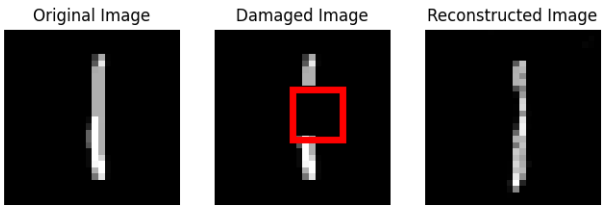


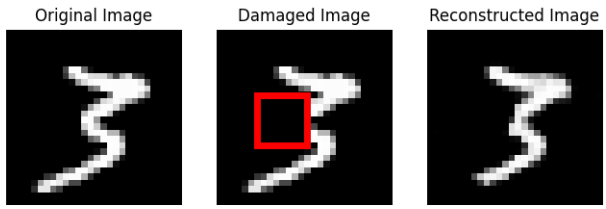


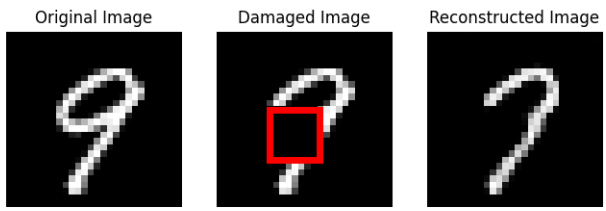


תוצאות מרשימות

2. $Inpainting$: למשימה זו בחרתי ב$L1$. המשימה הזו הייתה לו הרבה יותר קשה....
תוצאות:







כמו שניתן לראות לפעמים הוא עבד טוב ולפעמים לא. הדוגמא של הספרה "1" היא לא יצאת דופן - בספרה הזו הוא הצליח יחסית לשחזר טוב, וזה הגיוני כי צורה מאוד גסה ללא הרבה פרטים קטנים. בשחזור של הספרה "9" יש עוד דוגמא ל2 פרטים, שאחד מהם גס והמודל משחזר, ואחד מהם עדין והמודל לא משחזר: הרגל הספרה 9 היא פרט גס, התבנית של קו ישר חוזרת על עצמה בהרבה ספרות ולכן המודל מצליח לשחזר את הרגל של הספרה. לעומת זאת, החלק המעוגל של ה9 הוא פרט עדין, והמודל לא מצליח לשחזר אותו. ואכן, גם אם בן אדם היה מסתכל על התמונה הפגומה - זה די קשה להבין אם המשורר התכוון ל7 או ל9 או אפילו ל1..

חלק תאורטי

שאלה 1

נסמן את התפלגות המקור בP_x, והתפלגות פשוטה שידועה לנו בP_z , כך ש $x \sim P_x$, ונסמן בM את פונקציית החלפת המשתנה בין P_xלP_z, כך שמתקיים:

$$P_x(x) = \left|\frac{dM^{-1}(x)}{dx}\right| P_z\left(M^{-1}(x)\right) \quad , M(z) = x$$

כש $\frac{dM^{-1}(x)}{dx}$ הוא היעקוביאן של M^{-1}, ו$|\cdot|$ היא דטרמיננטה.

$GLOW$ משתמש בM בצורה דומה לGAN, אבל מניח עליו מספר הנחות, כדי שיוכל לחשב אותו ישירות. בהינתן דאטא, ותחת ההנחות הללו, אנחנו יכולים לאמוד את P_x על ידי חישוב הMLE בעזרת הנוסחה הנ"ל והדאטאסט.

ההנחות שמתקיימות תחת $GLOW$ הן:

1. כל אחת מהשכבות ברשת היא הפיכה וניתן לחשב אותה בצורה אנליטית ומדוייקת
2. הדטרמיננטה של היעקוביאן של כל שכבה ניתן לחישוב בצורה יעילה (עד $O\left(n^3\right)$). מכאן נגזר תנאי שכל היעקוביאנים צריכים להיות מטריצות משולשיות (או פחות, מבחינת סיבוכיות).
3. המימדים בין השכבות אינם משתנים (אחרת, הפונקציות יהיו לאו דווקא הפיכות)

נסמן את השכבות ברשת $h_1, h_2 ... h_k$, כך שכל השכבות הפיכות ומתקיים:

$$M = h_k \circ h_{k-1} \circ ... \circ h_1$$

נשים לב כעת, שמכך ש $det(A^{-1}) = det(A)^{-1}$ נובע ש:

$$P_x(x) = \left|\frac{dM(x)}{dx}\right|^{-1} P_z\left(M^{-1}(x)\right)$$

בנוסף, מכיוון ש M היא הרכבה של פונקציות הפיכות אז גם M היא הפיכה ומתקיים $M^{-1}(x_i) = z_i$ עבור i סמפל בדאטאסט. אנחנו כאן מדברים על כל ההתפלגות, ולכן נסמן פשוט:

$$P_x(x) = \left|\frac{dM(x)}{dx}\right|^{-1} P_z$$

כדי לאמוד את P_x, נמקסם את הנראות שלה על פני הדאטא. ה$log - likelihood$ נתון על ידי:

$$\begin{aligned}
log\left(P_x(x)\right) &= log\left(\left|\frac{dM(x)}{dx}\right|^{-1} P_z\right) \\
{}_1 &= log\left(\left|\frac{dM(x)}{dx}\right|^{-1}\right) + log\left(P_z\right) \\
{}_2 &= log\left(P_z\right) - log\left(\left|\frac{dM(x)}{dx}\right|\right) \\
{}_3 &= log\left(P_z\right) - log\left(\left|\frac{dh_k(z_{k-1}) \circ h_{k-1}(z_{k-2)}) \circ ... \circ h_1(x)}{dx}\right|\right) \\
{}_4 &= log\left(P_z\right) - log\left(\left|\frac{dh_k(z_{k-1})}{dh_{k-1}}\right| \cdot \left|\frac{dh_{k-1}(z_{k-2})}{dh_{k-2}}\right| ... \left|\frac{dh_1(x)}{dx}\right|\right) \\
{}_5 &= log\left(P_z\right) - \sum_{j=1}^{k} log\left(\left|\frac{dh_j(z_{j-1})}{dh_{j-1}}\right|\right)
\end{aligned}$$

כאשר 1 ו2 מחוקי לוגריתמים, 3 הצבה של M כהרכבה של פונקציות, 4 מכללי דטרמיננטה, ו5 מחוקי לוגריתמים. יש לשים לב שכל שכבה h_j פועלת על אינפוט מהשכבה הקודמת,

z_{j-1} ועל כן חישובי היעקוביאן בהתאם. כמו כן נגדיר את h_0 להיות פשוט $h_0 = x$ בשביל קונסיסטנטיות בסימונים.
אם כן, הMLE הוא:

$$MLE_{P_x} = argMax_{\{h_1 \dots h_k\}} \left\{ log\left(P_z\right) - \sum_{j=1}^{k} log\left(\left|\frac{dh_j(z_{j-1})}{dh_{j-1}}\right|\right) \right\}$$

הרכיב הראשון ניתן לחישוב כי אנחנו יודעים את P_z,והחישוב של הרכיב השני כמובן נעשה ע"י שיטות מבוססות גרדיאנט.
המקרה שונה בGAN ובGLO, שהם אינם ממדלים את ההתפלגות P_x אלא:

- GAN מנצל את **קיומו** של החלפת משתנה M שמחליף בין P_x לP_z, ורק משתמש באומדן שלו על מנת לייצר דגימות חדשות. בפרט, הוא לא מניח שום דבר לגבי M, ולא מחשב בעזרתו את P_x.
- GLO לא מנסה למדל את P_x, אלא מנסה לאפטם את $Range(G)$, בעזרת הוקטורים מה$latent\ space$, כלומר לגרום לתמונה של ה$Generator$ להיות הכי קרובה לדאטא סט, בעזרת מטריקות על z .

שאלה 2

בתחילת האימון של GAN, לD קל מאוד לזהות את התמונות המזוייפות שG מייצר (זה בסך הכל לקבוע האם תמונה היא רעש) והוא קובע $Decision\ plane$ מאוד חד - כלומר עבור התמונות המזוייפות הערך יהיה מאוד קרוב ל0, ועבור אמיתיות מאוד קרוב ל1. בנוסף, כל התמונות יהיו לו בבירור מזוייפות כי G עוד לא ידע לייצר תמונות טובות. באשר לG, אנחנו מעדכנים אותו לפי הגרדיאנטים של D, ולכן בשלב הזה אם D ייתכנס - משמע, הגרדיאנטים שלו יהיו 0 - אז G לא יוכל ללמוד ונגיע לסיטואצית הסטורציה. להלן ההסבר האנליטי:
אנחנו מעוניינים לחשב את $\frac{\partial L(D)}{\partial G(z)}$, כלומר לחשב את הגרדיאנט של הלוס של D, ביחס לG.
פונקציית הלוס היא כמו שלמדנו:

$$L\left(D\right) = log\left(D(x)\right) - log\left(1 - D\left(G\left(z\right)\right)\right)$$

כשx תמונות אמיתיות וz וקטורים של רעש ב$latent\ space$. בעקבות האמור לעיל לגבי מצב האימון ההתחלתי, נניח את ההנחה ש$D\left(G\left(z\right)\right) = 1 - \varepsilon$, כלומר D יודע לסווג כמעט את כל התמונות המוזייפות בהצלחה. נפתח את הביטוי:

$$\begin{aligned}
\frac{\partial L\left(D\right)}{\partial G(z)} &= \frac{\partial log\left(D(x)\right) - \partial log\left(1 - D\left(G\left(z\right)\right)\right)}{\partial G\left(z\right)} \\
{}_1 &= -\frac{\partial log\left(1 - D\left(G\left(z\right)\right)\right)}{\partial G\left(z\right)} \\
{}_2 &= -\frac{\partial log\left(1 - (1 - \varepsilon)\right)}{\partial G\left(z\right)} = -\frac{\partial log\left(\varepsilon\right)}{\partial G\left(z\right)} \\
{}_3 &= -\frac{1}{\varepsilon} \cdot \frac{\partial \varepsilon}{\partial G\left(z\right)} \\
{}_4 &= -\frac{1}{\varepsilon} \cdot 0 = 0
\end{aligned}$$

כאשר 1 נובע מכך ש$D(x)$ לא תלוי בG, 2 נובע מההנחה למעלה, 3 נגזרת של log, ו4 נובע מכך שε הוא קבוע ביחס ל$G(z)$.
כלומר, אם D התכנס במצב ההתחלתי - ל G לא תהיה השפעה על הגרדיאטנים שכבר התאפסו, והלמידה תיעצר - סטורציה.

שאלה 3

בשאלה הזו אנחנו עוסקים במצב של $Mode\ Collapse$.

1. הבעיה הפנימית $Min_G\{V(D,G)\}$ מאפטמת את G ביחס לD קבוע כלשהו. למעשה, היא מוצאת את המצב של G שגורם לתחזיות של D להיות מינימליות. הסבר: ניזכר בהגדרה של V כמו שהגדרנו בכיתה:

$$V(D,G) = \mathbb{E}_{p_{data}}\left[log\left(D(x)\right)\right] - \mathbb{E}_{p_z}\left[log\left(1 - D(G(z))\right)\right]$$

אם נתייחס למצב בו D קבוע, אז ישנו ערך קבוע של G עבורו מתקבל $D(G(z)$ מינימלי (ובך כפועל יוצא $\mathbb{E}_{p_z}\left[log\left(1 - D(G(z))\right)\right]$ מקסימלי ובכך לבסוף V מינימלי כמו שהבעיה הפנימית דורשת) והמשמעות היא שהדבר נכון **לכל** z, כלומר עבור כל אינפוט הוא יוציא את אותה התמונה שעבורה D יביא ערך מקסימלי. זוהי בדיוק הבעיה של $Mode\,Collapse$.

2. הבעיה החיצונית $Max_D\{Min_G\{V(D,G)\}\}$ מאפטמת את D שיביא ערך מקסימלי, בהינתן G שכבר התכנס, או לצורך העניין - G מושלם, שמייצר תמונות שנראות לגמרי אמיתיות. זו משימה קשה עבור רשת מוגבלת מבחינת פרמטרים, כי הבחנה בין תמונה אמיתית לתמונה מזוייפת מצויינת - תלויה בפרטים מאוד מאוד קטנים, ולצורך כך צריך D חזק ובפרט גדול. בנוסף, ככל שהרזולוציה של התמונה גדולה יותר כך צריך D עוד יותר גדול שיוכל לתפוס את כל הקשרים והפרטים בתמונה.

3. אני רואה בסנריו הזה 2 סכנות - כאמור, הראשונה היא $Mode\ Collapse$ כמו שהוסבר למעלה.
שנית, במקרה ש D לא מצליח להפריד טוב בין תמונות אמיתיות לתמונות מזוייפות, הגרדיאנטים שהוא ייצר יהיו בכיוון לא נכון עבור האימון של G, ובעצם הדבר יוריד את הביצועים הכלליים.