תרגיל 3 ברשתות נוירונים

אופיר בירקה - 316389410
נריה אוחנה - 304933575

חלק מעשי

כדי לפתור את השאלות בחלק המעשי, ניסינו בהתחלה להשתמש בשכבה אחת של FC, שתעביר וקטור באורך 100 (Z) אל וקטור באורך 784 שמייצג תמונה של 28X28.
כאשר השתמשנו בארכיטקטורה זו כ Autoencoder קיבלנו תוצאות טובות עם מספר נמוך של איטרציות, והמספרים נראו מדוייקים.
אך כאשר הכנסנו את השכבה של ה FC בשאלה C במקום הגנרטור, הGAN לא הצליח לייצר מספרים בצורה טובה. הבנו ששכבה אחת לא מספיקה בשביל GAN ולכן הוספנו שכבות של קונבולוציה והשתמשנו בארכיטקטורה כבדה יותר. כתוצאה מכך מספר הפרמטרים גדל משמעותית והיה צורך באימון ארוך יותר כדי לראות תוצאות. ראינו שהתוצאות משתפרות עם התקדמות האימון, אך האימון היה ארוך ואנו מאמינים שאם היינו יכולים לאמן עם כוח חישוב יותר חזק התוצאות היו מדוייקות אפילו יותר..

שאלה 1
הארכיטקטורה שהשתמשנו בה כללה:

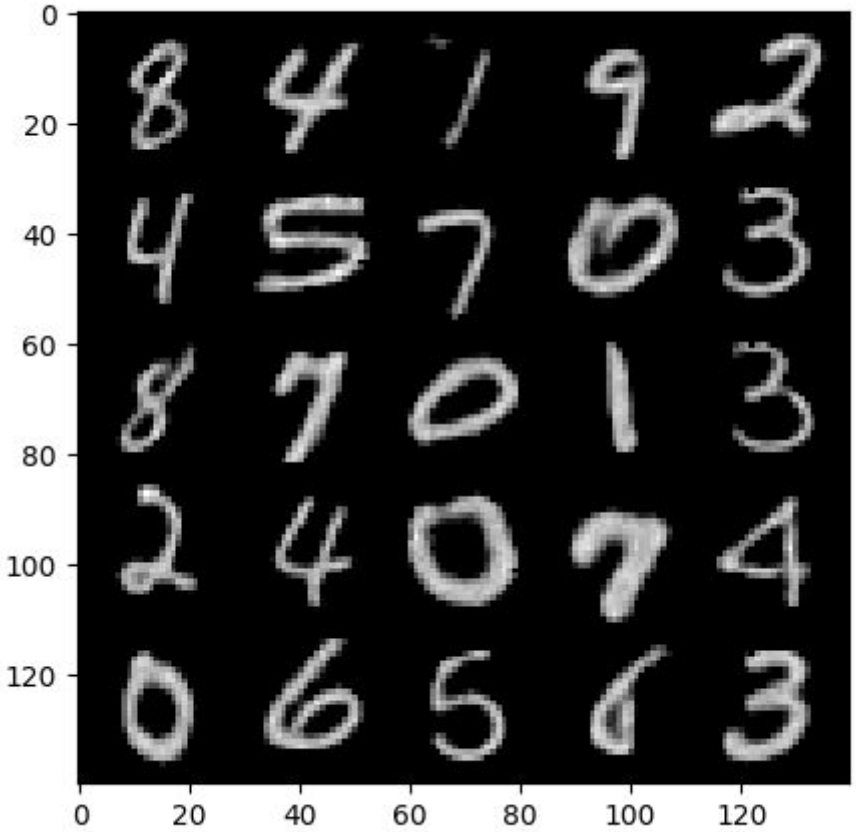
Encoder:

- קלט של תמונה בגודל 28X28
- שכבת קונבולוציה עם 4 ערוצי פלט, ללא פדינג (SAME) , ואחריה RELU
- מקס-פולינג
- שכבת קונבולוציה עם 8 ערוצי פלט, ללא פדינג (SAME) , ואחריה RELU
- מקס-פולינג
- שכבת קונבולוציה עם 16 ערוצי פלט, ללא פדינג (SAME) , ואחריה RELU
- מקס-פולינג
- שכבה של FC מוקטור של 4X4X16 לוקטור של 100

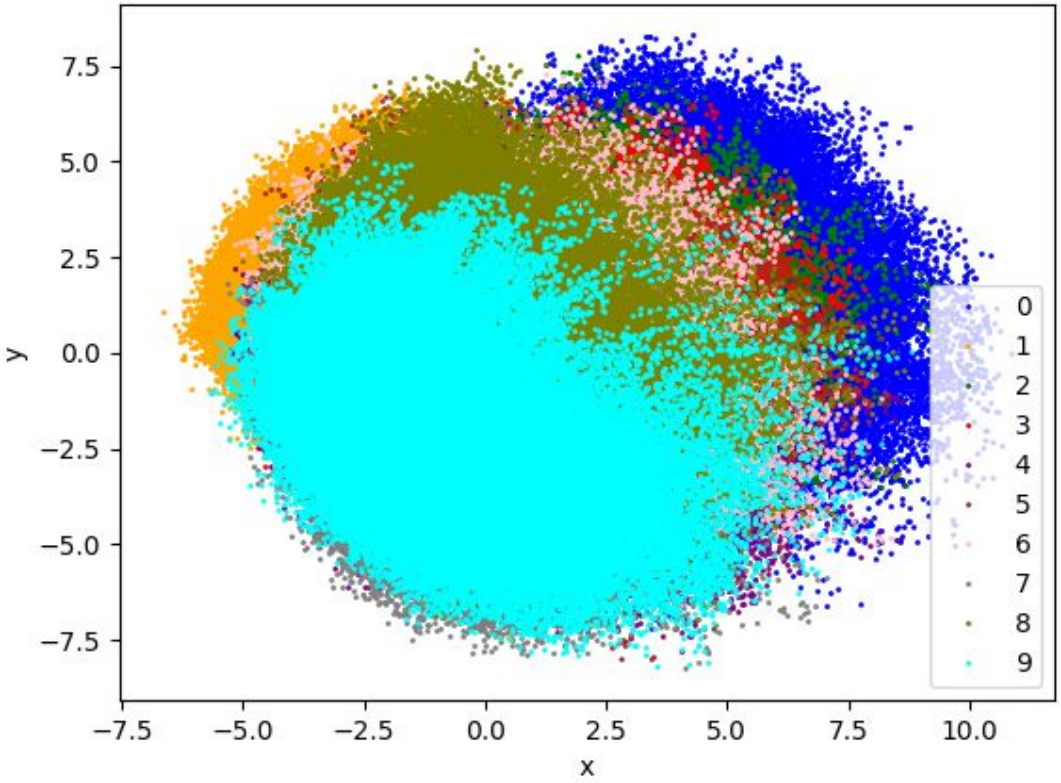
Decoder:

- קלט באורך 100
- שכבה של FC שמעבירה וקטור של 100 אל צורה של [14,14,8]
- שכבת קונבולוציה עם 8 ערוצי פלט, ללא פדינג (SAME) , ואחריה RELU
- שכבה של FC שמעבירה לצורה של [28,28,4]
- שכבת קונבולוציה עם ערוץ פלט אחד, ללא פדינג (SAME) , ואחריה RELU
- פלט של תמונה בגודל 28X28

לאחר אימון של הארכיטקטורה על MNIST, קיבלנו שעבור מספר תמונות של ספרות קיבלנו את הפלט הבא:



הורדה של הוקטורים בעזרת PCA נתנה לנו:

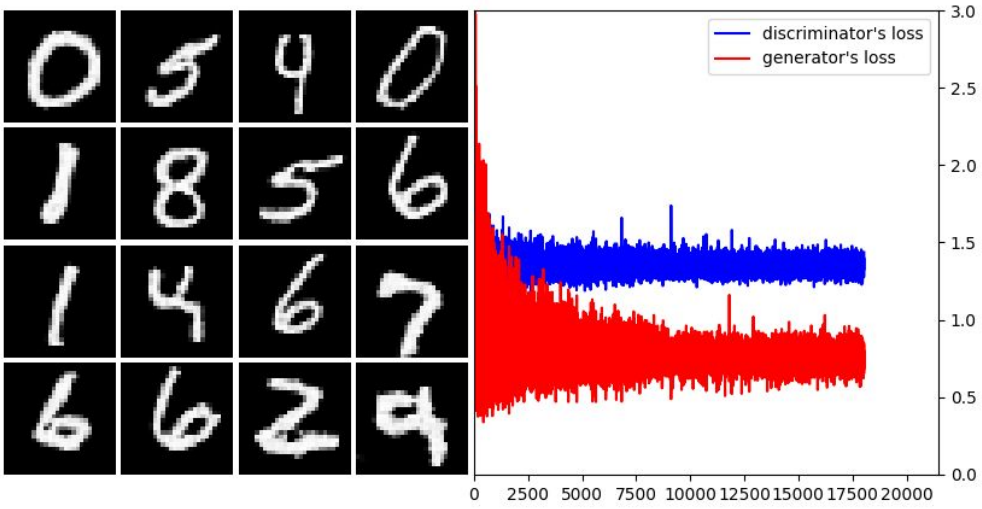


ניתן לראות שיש הפרדה של הספרות, אך גם איזורים של חפיפה, שצפויים כיוון שהורדנו 10 קבוצות למימד 2
.

שאלה 2

סעיף א

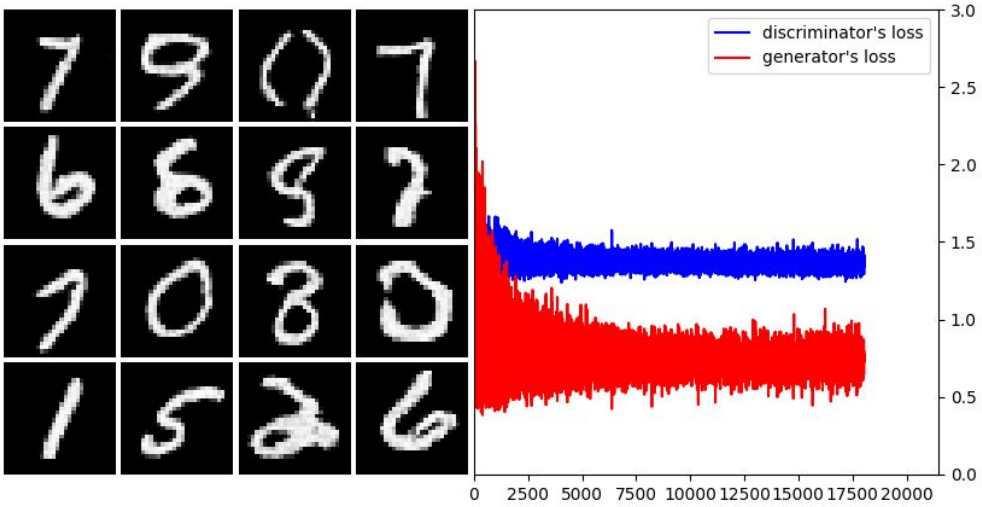
כאשר ייצרנו וקטורים של Z בדגימה אחידה: uniform([-1,-1]) קיבלנו:



סעיף ב

כאן השתמשנו ב Gaussian mixture סביב 10 נקודות בטווח [-1:1] וקיבלנו:

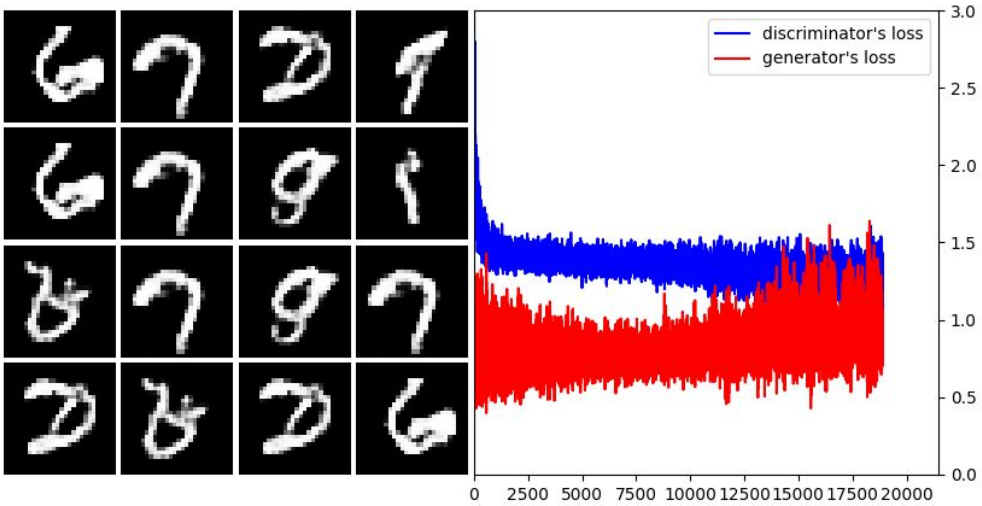
ו



כאן אנו רואים ששני ה LOSS קצת יותר יציבים ביחס לגרף הקודם (ביחס לציר ה Y יש פחות קפיצות).
כנראה שכאשר דוגמים סביב 10 מרכזים מכניסים קצת יותר יציבות מאשר לדגום רנדומלית על פני כל הטווח.

סעיף ג

כדי לדגום וקטורים של MNIST לאחר שעברו ב Autoencoder יצרנו קובץ בשם ModuleForQ2 שעובר על MNIST, לוקח תמונות ומריץ על ה Autoencoder , ויוצר מהם קובץ ששומר את הוקטורים.
לאחר מכן השתמשנו בקובץ זה כדי לקבל וקטורים של Z במהלך האימון. התוצאה בגרף הבא.
מהסתכלות על האיכות של המספרים שהתקבלו, נראה כי המספרים פחות מדוייקים ולכן ניסיון זה לא שיפר את התוצאה.

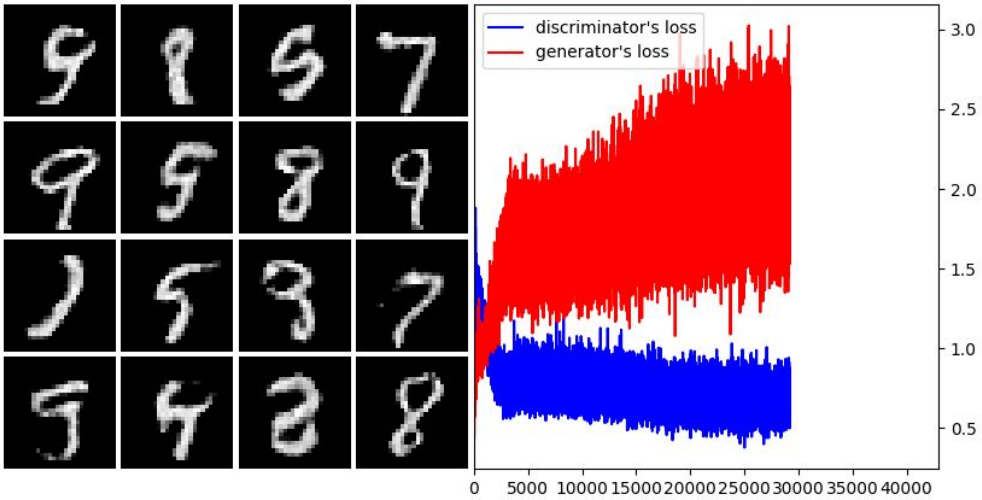


<u>שאלה 3</u>

החלפנו את ה generator ב Autoencoder המאומן משאלה 1. השתמשנו באפשרות לשמור מודל משאלה 1 עם כל המשקלים, ובשאלה זו שחזרנו חלק מהשכבות מתוך המודל השמור, כך שהשכבות של ה generator קיבלו את הערכים מתוך ה Autoencoder .

כמו שהזכרנו קודם, ניסינו בהתחלה להשתמש בשכבה אחת של FC. אנו מניחים שלצורך משימה זו צריך להוסיף יותר שלבים ולכלול קונבולוציה, כי המשימה של יצירת תמונות ב GAN מורכבת יותר ומצריכה יותר פרמטרים ואת האפשרות להסתכלות לוקאלית עם קונבולוציות.

לאחר שהוספנו שכבות נוספות וקונבולוציה קיבלנו:



<u>חלק תאורטי</u>

<u>שאלה 1</u>

סעיף א

נסמן I=I+N כאשר N~laplace(0,b) כלומר N(x)=1/2b*exp(-|x|/b) . וביטוי זה הוא בעצם גם(p(Ic|I, כי אנו מחפשים את ההסתברות לתמונה מורעשת בהינתן תמונה, כלומר ההסתברות לרעש.

נשתמש בחוק בייס ונציב:

p(I|Ic)=p(Ic|I)*p(I)/p(Ic)=p(I)/(p(Ic)* 2b*exp(-|x|/b))

סעיף ב

נרצה לתת ביטוי לפונקציית ההתפלגות המצטברת של שני ההתפלגויות ונרצה להשוות ביניהן וכך נמצא את פונקציית המעבר בין ההתפלגויות:

Let's define $F_1(x) = \int_0^x (s)ds$ (the uniform distribution)
And let's define $F_2(y) = \int_0^y (0.5 + s)ds$ (the second distribution)
We want to find a mapping function between a given x to y
$F_1(x) = x$
$F_2(y) = 0.5y + 0.5y^2$
So let's compare them
$0.5y + 0.5y^2 = x$
$y = \frac{-1+\sqrt{1+8x}}{2}$

לדוגמא: עבור x=⅜ נקבל y=½ וניתן לוודא שהשטחים תואמים (כלומר פונקציית ההתפלגות המצטברת זהה).

סעיף ג

נתאר אלגוריתם לדגימה בשיטת Gibbs:

1. נאתחל t=0
2. נדגום x0 מהקטע [0,1] בהסתברות אחידה
3. לולאה עד t=M
    a. הגדר t=t+1
    b. נדגום x1 בהסתברות אחידה מהקטע [x0^2-1, -x0^2+1]
    c. נדגום x0 בהסתברות אחידה מהקטע [x1^2-1, -x1^2+1]
4. נחזיר את (x0,x1)

האלגוריתם דגימה המתקבל הוא אי פריק כיוון שאפשר תוך מספר סופי של צעדים להגיע מכל מקום במעגל לכל מקום במעגל, וזאת הסקנו בעזרת האבחנה הבאה:

מנקודת המרכז(0,0) ניתן להגיע לכל נקודה במעגל היחידה תוך צעדים - למשל אם נרצה להגיע לנקודה (x,y) פשוט נבחר את הערכים האלו.

לכן גם ההפך הוא הנכון - מכל נקודה במעגל ניתן להגיע לנקודת המרכז תוך 2 צעדים לכל היותר.

ולכן כתוצאה מכך, תוך לכל היותר 4 דגימות (דגימה לx, דגימה לy, דגימה לx, דגימה לy) ניתן להגיע מכל מקום לכל מקום במעגל.

שאלה 2

ננסח מודל שבו I זו תמונה, h זה ייצוג של התמונה על ידי וקטור כך שניתן לקבל את I מ h בצורה דטרמיניסטית. ו c זה המחלקה של התמונה, אזי נוכל לייצג את המודל כך:

$$P(I,c,h)=P(h)*P(I|h)*P(c|I)=P(h)*P(c|I)$$

השוויון הראשון מגדיר את המודל, והשוויון השני מתקיים כיוון שניתן לקבל את I מ h בצורה דטרמיניסטית, אז ניתן להשמיט את $P(I|h)$

כדי להשתמש במודל נאמן רשת CNN על dataset גדול של תמונות - נניח ImageNet.

את הביטוי $P(c|I)$ נוכל לקבל ישירות מה CNN, אם נעביר תמונה I ונקבל תיוג c .

את הביטוי $P(I|h)$ נוכל לקבל על ידי הפיכת השכבות ב CNN (עם שימוש ב deconvolution), וכך עבור וקטור קלט h נוכל לקבל את התמונה עצמה.

החלק החופשי היחיד במודל הוא בחירת h, אותה נוכל לדגום מהתפלגות גאוסיאנית. ואחרי שנדגום את h, נוכל לקבוע ממנו את התמונה ואח"כ את התיוג, כך שבחירת h היא החלק החופשי היחיד במודל $P(I,c,h)$.

שאלה 3

א. נבנה autoencoder (נבחר ארכיטקטורה מסוימת). נאמן את הautoencoder בצורה הבאה, נכניס תמונות טבעיות, ואת פונקציית הloss היא מרחק L2 בין הoutput לinput (התמונה שהכנסנו). כאשר נרצה לזהות משהו חריג בתמונה, נכניס אותה לautoencoder ונבדוק את הloss בין תמונת הפלט לתמונת הקלט, אם השגיאה גדולה מידי נחזיר שהתמונה חריגה.

המחשבה מאחורי האלגוריתם, כאשר אנחנו מכניסים לautoencoder תמונות טבעיות ורוצים למזער את ההפרש בין תמונת הפלט של הautoencoder לתמונה שהכנסנו, מה שיקרה זה, שהlatent vector שילמד יהיה אופטימלי בשחזור תמונות טבעיות, וכאשר תכנס תמונה לא טבעית, הautoencoder יוציא תמונה שיחסית טבעית לתמונה שהכנסנו, ולכן השגיאה ביניהן תהיה גדולה וכך נדע שהתמונה שהוכנסה היא חריגה.

ב. נבנה autoencoder. נאמן את הautoencoder בצורה הבאה, נכניס תמונות טבעיות, ואת פונקציית ה loss היא מרחק L2 בין הoutput לinput (התמונה שהכנסנו). כשנרצה להוריד עיניים אדומות, נחזיר את תמונת הפלט של הautoencoder.

המחשבה מאחורי האלגוריתם, היא כאשר אנחנו מאמנים, הlatent vectors יתנו ייצוג לעיניים בתמונות של ה training data. וכאשר נכניס תמונה עם עיניים אדומות, בlatent space נחליף את הייצוג הפגום של העיניים בתמונה, בייצוג הנכון של העיניים שהautoencoder אומן עליהן.

ג. נבנה autoencoder. נאמן את הautoencoder כך, בהינתן dataset של תמונות, לכל תמונה נוסיף רעש/טשטוש ונכניס את זה לautoencoder, ואת פונקציית נגדיר כמרחק L2 בין תמונת הפלט של ה autoencoder לתמונה המקורית.

המחשבה מאחורי האלגוריתם היא שכשאר אנחנו מאמנים את הautoencoder עם תמונות פגומות לייצר תמונות טובות, הlatent vectors שהוא לומד יהיו לא רגישים לפגמים בתמונות. ולכן בסוף האימון כשנכניס תמונה פגומה, הencoder ייצג את התמונה ב latent space (ושם הפגמים בתמונה לא יהיו מיוצגים) ואז ה decoder יוכל לשחזר את התמונה ללא הפגמים.

שאלה 4

סעיף א

נבנה גנרטור שמקבל תמונה ומוציא תמונה. כלומר הקלט והפלט שלו באותו גודל, ובאמצע שכבות קונבולוציה.

נבנה דיסקרימינטור שמקבל תמונה וקובע האם היא תמונה טבעית או תמונה מורעשת (על פי הרעש שמוגדר באמצעות הפונקציה C(I|J)).
ניקח מאגר תמונות טבעיות מאותה התפלגות של התמונות לפני הרעש, ומאגר נוסף של תמונות מורעשות.
ניתן לגנרטור תמונה מורעשת ונצפה שיוציא ממנה תמונה טבעית. במהלך האימון הדיסקרימינטור יצטרך להבדיל בין תמונות שייצר הגנרטור לבין תמונות ממאגר התמונות הטבעיות. בדומה לאימון של GAN, נגדיר את הלוס כך שישאף לתת ערך גבוה לתמונות אמיתיות ונמוך לתמונות של הגנרטור.
נאמן את שתי הרשתות במקביל בדומה לGAN, כך לאט לאט הגנרטור יהפוך את התמונות לתמונות טבעיות עד שהדיסקרימינטור לא יצליח להבדיל ביניהם לאמיתיות.
כדי לשמור על קשר בין התמונה שמקבל הגנרטור לתמונה שהוא מוציא, ניתן להוסיף רגולריזציה שממזערת את ההפרש בין התמונה המורעשת לתמונה שהוא מוציא וכך השינויים שלו יהיו קטנים וישאירו את התמונה דומה למעט הסרת הרעש כך שהתמונה תראה טבעית יותר. ניתן גם להשתמש ברגולריזציה שראינו בתרגיל הראשון שגורמת לו ליצור תמונות עם טרנספורם פוריה דומה לשל תמונה טבעית וכך לשפר את ההתכנסות לכיוון תמונות טבעיות.

סעיף ב

התוצאות שמקבלים מGAN זהות לתוצאות שנדגמות באופן אחיד, כיוון שאם מניחים שהGAN מתכנס, כלומר הdiscriminator מבחין בהסתברות 0.5 בין תמונות שהgenerator מייצר לבין תמונות מהdataset. ודגימה באופן אחיד כמו בMCMC מייצרת דגימות מההתפלגות הרצויה (כמו בdataset), לכן הdiscriminator של הרשת הזו לא יוכל להבחין בין תמונות שמיוצרות ע"י הgenerator לבין תמונות מMCMC.