

## Intro to Deep Learning – תרגיל 3:

### חלק מעשי:

#### Transfer learning:

בחלק זה נתבקשנו לאמן על 50 דוגמאות בלבד שתי רשתות: האחת היא קלאסיפייר MLP והשני קלאסיפייר MLP על בסיס encoder שאומן בlatent spaces. שתי הרשתות נדרשות לחזות על פי תמונה מדאטאסט mnist את הספרה שכתובה בה בכתב יד. אנחנו מאמנים על מספר מועט של דוגמאות כיוון שבסעיף זה נראה כיצד האימון שבוצע בשלב הencoding הוא בעל משמעות לתוצאות הסופיות.

ניסינו ארכיטקטורות שונות כדי למצוא ארכיטקטורה מיטבית שתביא לביצועים הטובים ביותר עבור transfer learning:

Architecture	MLP accuracy	encoder+MLP accuracy
(softmax,10)	0.6404	0.5928
(relu,32)->(softmax,10)	0.6172	0.6363
(sigmoid,32)->(softmax,10)	0.6443	0.6862
<b>(sigmoid,64)-&gt;(softmax,10)</b>	<b>0.6501</b>	<b>0.7295</b>
(sigmoid,64)->(sigmoid,64)->(softmax,10)	0.5948	0.6649

אקטיבציה של sigmoid הביאה תוצאות יותר טובות עבור ארכיטקטורות שונות על פני relu, לכן נשארנו איתה. נבחין כי בכל הארכיטקטורות למעט הראשונה (רק שכבת softmax) הביצועים של transfer learning יותר טובים מאשר MLP בלבד, כאשר הארכיטקטורה הטובה ביותר היא שכבה של 64 נוירונים של sigmoid ואחריה softmax. אלו הן התוצאות להן ציפינו.

### Generative Adversarial Networks

בחלק זה עשינו מימוש לGAN בlatent spaces של encoder. 'יצרנו generator וdiscriminator לפי הארכיטקטורות הבאות:

Generator:

- הקלט מתפלג גאוסיאנית במימד 64 תוחלת 0 ושונות 1
- Relu,256
- Relu 512
- tanh 512
- sigmoid 256
- פלט ממימד 64 כלומר latent dim

discriminator:

- relu 64

- relu 128
- relu 256
- sigmoid 1 – הפרדיקציה אם התמונה מהגנרטור או מקורית.

ה generator אמור ליצור תמונות מתוך התפלגות התמונות בlatent space discriminator מבחין בין הדגימות שנוצרו ע"י generator לדגימות המקוריות. את הloss הגדרנו להיות optimizer binary crossentropy להיות Adam. האימון התבצע באופן הבא: בכל איטרציה דגמנו דגימות אמיתיות וייצרנו דגימות באמצעות generator, ואימנו את הGAN עליהן. לאחר האימון יצרנו ספרות שלמד ה generator לאחר האימון, ולהלן התוצאה:



התוצאות לא יצאו מאוד טובות, ונראה שחלקן קשה לזהות מה הספרה וחלק ניתן להבין.

לאחר מכן יצרנו אינטרפולציה בין שתי ספרות שונות. את האינטרפולציה מימשנו גם באמצעות AE וגם באמצעות GAN והשווינו בין התוצאות שהתקבלו. האינטרפולציה בפועל נעשית בין שני הוקטורים שכל אחד מהם מייצג ספרה אחרת, ומראה את שלבי האמצע ביניהם, ולהלן התוצאה:

GAN העליון AE תחתון



בAE – מעבר יותר הדרגתי כך שהשלבים באמצע נראים כמו "היבריד" של שתי הספרות  
 בGAN – ביצועים פחות טובים, ונראה שבכל שלב מנסה להתחייב לספרה אחרת

## Digit-Specific Generator

נתבקשנו ליצר מנגנון שבהנתן ספרה שהשתמש מבקש תוחזר כזאת ע"י מודל GAN. נשים לב כי הארכיטקטורה בה משתמשים כמעט זהה לחלוטין לסעיף קודם למעט העובדה שאנחנו מוסיפים את המידע של הספרה שאותה נרצה ליצור.

להלן התוצאה:

## חלק תיאורטי:

- שימוש בBERT לפעולת קונבולוציה** – נבחין כי יש הבדל מהותי בסוג הקישוריות שattention עושה, שיכול לקשר בין שני רכיבים מאוד רחוקים במשפט, לבין הקישוריות של קונבולוציה שתקשר בין פיקסלים סמוכים בלבד. על בעיה זאת נרצה להתגבר כדי שנוכל להשתמש בBERT על תמונות ולקבל תוצאות מניחות את הדעת. דרך עליה חשבנו היא להוסיף לכל פיקסל קידוד של מיקומו בתמונה, כלומר פיקסלים קרובים יקבלו ערכים קרובים ששכבת הattention יכולה לקשר ביניהם וכך להבין תמונה לפי האזורים שבה.  
**שימוש בBERT כ-FC layer** – בשכבת הattention שבBERT מתקבל וקטור קלט, והפלט (value) מחושב כקומבינציה קמורה של כל אחד מהפלטים האפשריים (כלומר לפי וקטור הסתברות שמחזירה שכבת softmax) וזה שונה באופן מהותי משכבת FC שבה כל קלט נמדד ביחס לעצמו. לשם כך, נוכל להכניס וקטור קלט שמכיל את אותו הערך בכל הקורדינטות. value יהיה אותו דבר לכל אחד מהkeys, ונקבל וקטור התפלגות אחיד ולכן נקבל פלט שהוא value המתאים ביותר לאותו הקלט.
- Regular GAN** – בשיטה זאת G מקבל תמונות מ class A ו-D נדרש להבדיל בין תמונות ש-G יצר לתמונות מ class B. אנחנו לא נצפה שיהיו הבדלים בביצועים ככל שההתאמה בין זוגות של תמונות תגדל, כיוון שעבור D אין קשר בין תמונות ממחלקה A למחלקה B והוא לא יבחין ביניהן על סמך הקשר, אין רלוונטיות לסוגי הדאטא השונים במקרה זה.
- Cycle GAN** – G1 ממפה ממחלקה A למחלקה B ו-G2 ממפה ממחלקה B למחלקה A. D1 מבדיל בין מה ש-G1 יצר לתמונות ממחלקה B ו-D2 מבדיל בין מה ש-G2 יצר לתמונות ממחלקה A. הloss מחושב על ידי עד כמה התמונה לאחר שני המעברים דומה לתמונה המקורית, כלומר יש tradeoff בין מיפוי נכון של G1 לתמונה ממחלקה B ליצירת תמונה שמכילה מספיק אלמנטים מקוריים ש-G2 תוכל לשחזר את התמונה המקורית בצורה מיטבית. כאשר המחלקות A ו-B שונות לחלוטין, יהיה מאוד קשה לשמור על שתי הדרישות הללו כי יהיה צורך לשנות הרבה דברים בתמונה ליצירת מעבר תקין. ככל שהמחלקות יהיו דומות יותר עם שינויים פחות משמעותיים כך הביצועים ישתפרו.
- Conditional GAN** – D צריך להבחין בין  $(I, G(I))$  כאשר  $I \in A$  לבין  $(I, J)$  כאשר  $I \in A, J \in B$ . נדרש שוב לייצר מתמונה ממחלקה A תמונה ממחלקה B. לכן, באופן דומה לסעיף קודם נשער כי ככל שהמחלקות יהיו דומות יותר כך התוצאות של G יהיו יותר טובות. בנוסף, כיוון שבכל מחלקה קיים איבר שמתאים לאיבר במחלקה אחרת, G ילמד לייצר מתוך תמונה של A את המקבילה שלה ב-B בהדרגה כי D יכיר את שניהם ו"ישתכנע" רק כאשר G יתכנס לתמונה המקבילה.

