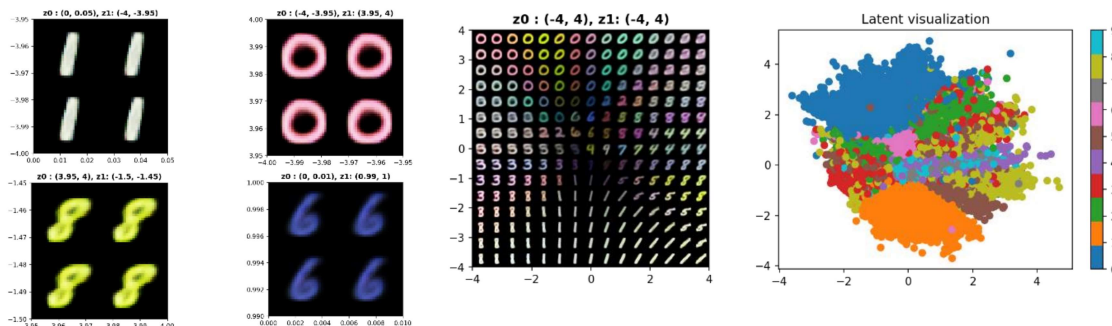


תרגיל בית 2

שאלה 1

VAE רציף

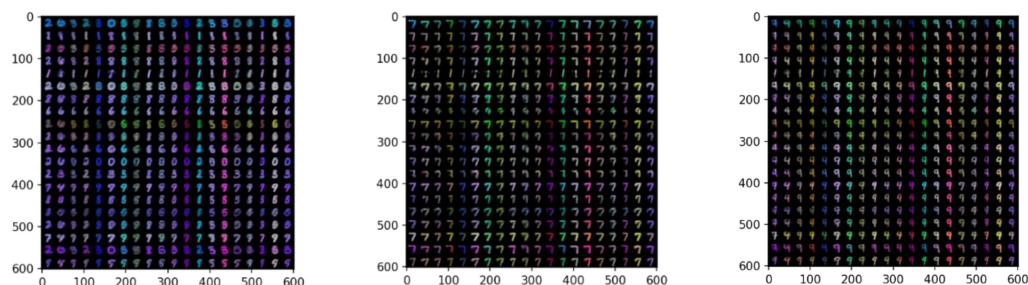
ארכיטקטורת המודל: הקלט הינו תמונה $(3 \times 28 \times 28)$ משוטחת לוקטור חד מימדי בגודל 2352. לאחר מכן יש 2 שכבות לינאריות: $2352 < 512$ ו $512 < 256$. בין כל שכבה מופעלת Relu. השכבה האחרונה מקבלת וקטור בגודל 256 וממירה אותו לוקטור בגודל 4: 2 לתוחלת ו 2 לשונות. בסך הגודל של z יהיה 4. לאחר מכן אני מפעיל reparametrization trick ומחזיר את וקטור הפלט (שגודלו 2) אל ה decoding. אתאר את שלב זה: 3 שכבות לינאריות. $2 < 256$, $256 < 512$ ו $512 < 2352$. פונקציית Relu מופעלת ב 2 השכבות הראשונות ובשכבת הפלט מופעל Sigmoid. הפלט בגודל הקלט. הדברים הבאים משותפים לכל המודלים: optimizer: ADAM עם $\text{learning rate} = 0.001$. פונקציית ה loss היא Binary cross entropy עם $\text{reduction} = \text{sum}$. כמות epochs: 30. Batch size: 120. הליך האימון: הליך אימון סטנדרטי שבו לכל batch חוזים את התמונות ומחשבים את ה loss שתואר לעיל בתוספת ערך ה KL שמחושב באופן זהה לנלמד בהרצאה.



משמאל רואים את התפלגות הצבעים והמספרים ובנוסף, 4 דוגמאות לשליטה על הצבע\ספרה. מימין זה המרחב הלטנטי.

VAE בדיד

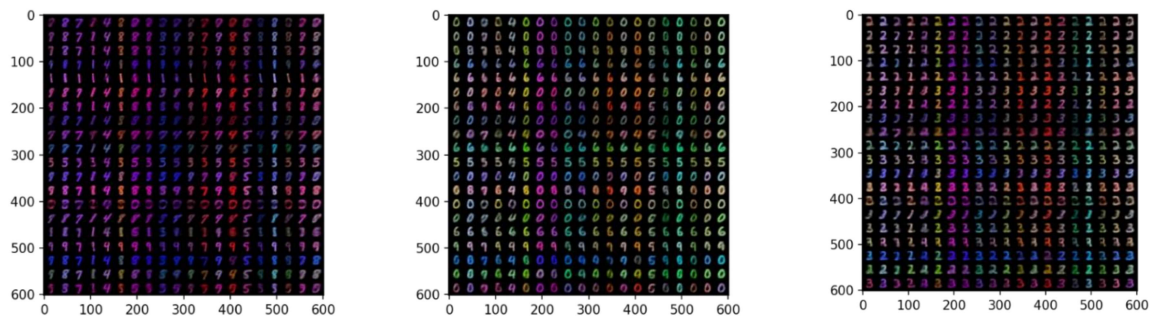
ארכיטקטורת המודל: שכבות קידוד ובנייה מחדש זהות למודל הרציף (6 בסך הכל). השוני היחיד הוא בפלט שכבת הקידוד השלישית $(K * N < 256)$ כאשר N זה מספר המשתנים ו K זה מספר הפיצ'רים. לאחר הקידוד נסתכל על הפלט כמטריצה בגודל $N \times K$ ונפעיל עליה gumbel softmax בדומה לנלמד בהרצאה. בנייה מחדש זהה למודל רציף. נחזיר את התמונה עם Softmax על המטריצה שתוארה קודם לכן. היפר פרמטרים: $N=3, K=20, \text{temperature} = 1.0, \text{hard} = \text{False}, \text{temp_min} = 0.5, \text{ANNEAL_RATE} = 0.0003$. להלן פלטים שממחישים שליטה בצבע או בספרה שיוצאת בפלט:



VAE משולב

ארכיטקטורת המודל: שילוב המודלים על ידי קנקוט וקטורי המרחב החבוי z . תמונת הקלט, אקודד אותה עד לגודל 256 באופן אחיד, ואז אקודד אותה פעמיים: ל z הרציף ול z הבדיד. ה z אותם אני מקנקט הם כבר אחרי הפעולות המתאימות לכל מודל (gumbel softmax או reparametrization trick). חלק הגנרטו זהה (3 שכבות לינאריות שתוארו במודלים הקודמים). בקריאת ה forward אחזיר את כל מה שרלוונטי לחישוב ה loss בשני המודלים: התמונה שנוצרה, kl של המודל הנורמלי והמטריצה q של המודל הבדיד (מגודל $N \times K$). גדלי ה z -ים וההיפר פרמטרים זהים למודלים.

הליך האימון: לכל batch ניצור תמונות ונחשב loss שהוא סכום של 3 גורמים: BCE loss ו-2 ה KL מהמודלים הקודמים. משום שכאן אין אנו לומדים רק מרחב רציף אלא גם בדיד, אין משמעות למרחב הלטנטי של החלק הרציף לבדו והוא מבולגן ואינו יעיל לדגימה. לכן על מנת לייצר את התמונות דגמתי z רציף מהתפלגות נורמלית $N(0,5)$ (מבוסס על המודל הרציף).



הבדלים בין המודלים: מודל vae רציף משתמש בהתפלגות נורמלית כהתפלגות הפריור למרחב הלטנטי z. מודל דיסקרטי לעומת זאת משתמש בכמה משתנים ובפיצ'רים שלהם על מנת לתאר את המרחב הלטנטי. מודל משולב לומד את המרחב הבדיד ואת המרחב הרציף של z. היתרון בשימוש במודל משולב הוא יכולתו ללמוד 2 מרחבי ייצוג על אותה תמונה וכתוצאה מכך ילמד דברים נוספים ועמוקים יותר שלא למד בכל מודל לחוד. מבחינה אמפירית המודל המשולב השיג את ה loss ביותר. בנוסף, הויזואליזציות הברורות ביותר היו של המודל המשולב. לאחר מכן המודל הדיסקרטי ולבסוף המודל הרציף.

שאלה 2

בחלק זה, בניתי 2 מדגמים של מספרים. בכל מדגם, כל ספרה צבועה בצבע שונה. מטרת המודל היא ללמוד כיצד לקחת ספרה בצבע מסוים ולהמירה לצבע שלה במדגם האחר. אתייחס ל-2 המדגמים כ מדגם A ומדגם B. ארכיטקטורת המודל: 2 ה discriminators בנויים באופן הבא: 2 שכבות קונבולוציה (1. ערוצי קלט, 32 ערוצי פלט, קרנל מגודל 3 על 3, 1 stride, 1 padding | 2. 32 ערוצי קלט, 64 ערוצי פלט, יתר הפרמטרים זהים לשכבה הקודמת). על כל שכבה מפעילים max pooling עם kernel בגודל 2 ו stride בגודל 2. לאחר מכן 2 שכבות לינאריות ($7*7*64 <$ 128 ו $128 < 1$) עם פונקציית Relu בין השכבות ופונקציית Sigmoid על הפלט. 2 ה generators בנויים באופן הבא: 3 שכבות קונבולוציה (1. 3 ערוצי קלט, 32 ערוצי פלט, קרנל מגודל 3, 1 stride, 1 padding | 2. 32 ערוצי קלט, 64 ערוצי פלט, קרנל מגודל 2 ו stride ו padding מגודל 1 | 3. 64 ערוצי קלט, 3 ערוצי פלט, קרנל מגודל 4 ו stride ו padding מגודל 1. על כל שכבה שאינה שכבת הפלט מופעלת פונקציית rel. optimizer: ADAM עם learning rate של 0.0003. פונקציית ה loss היא BCE. כמות epochs: 150. Batch size: 120

הליך האימון: לכל discriminator או generator יש מחלקה משלו ו optimizer משלו. בכל איטרציה דוגמים תמונה אמיתית מכל מדגם. מייצרים לייבלים שמייצגים תמונה אמיתית (1) ולייבלים לתמונה מזויפת (0). סדר האימון הוא: אימון discriminator של מדגם A, אימון generator ממחלקה B ל A, אימון discriminator של מדגם B ואז אימון generator של מדגם A למדגם B. אתאר כיצד מאומן discriminator: נותנים לו כקלט תמונה אמיתית מ A ותמונה מזויפת שהגנרטור של B ל A יצר. מחשבים loss לסיווג 2 התמונות בסכום ה loss של הסיווגים ומעדכנים את המשקולות. אתאר כעת כיצד מאומן generator: ה generator מייצר תמונה. את תמונה זו נכניס ל discriminator של מדגם המטרה ונתייחס אליה כתמונה אמיתית. מחשבים את ה loss ומעדכנים את המשקולות. להלן הפלט של שני ה generators:

