7. Deep Generative Models

המודלים שהוצגו בפרקים הקודמים הינם מודלים דיסקרימנטיביים, קרי הם מוציאים פלט על בסיס מידע נתון, אך לא יכולים ליצור מידע חדש בעצמם. בניגוד אליהם ישנם מודלים גנרטיביים, שלא רק לומדים להכליל את הדאטא הנלמד גם עבור דוגמאות חדשות, אלא יכולים גם להבין את מה שהם ראו וליצור מידע חדש על בסיס הדוגמאות שנלמדו. ישנם שני סוגים עיקריים מודלים גנרטיביים – מודלים המוצאים באופן מפורש את פונקציית הפילוג של הדאטא הנתון ובעזרת הפילוג מייצרות דוגמאות חדשות, ומודלים שלא יודעים לחשב בפירוש את הפילוג אלא מייצרים דוגמאות חדשות בדרכים אחרות. בפרק זה נדון בשני המודלים הפופולריים בתחום – VAE ו-GANs.

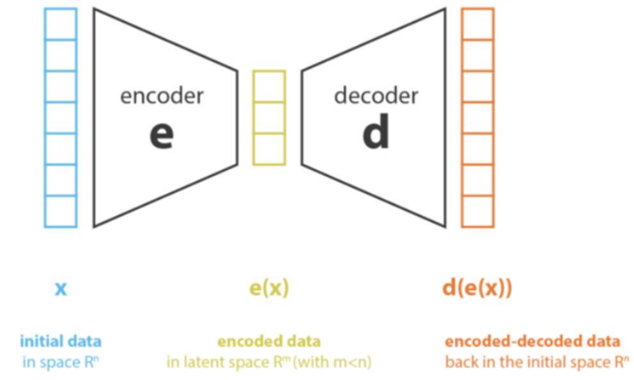
7.1 Variational AutoEncoder (VAE)

המודל הראשון הינו VAE, וכדי להבין אותו היטב יש להסביר קודם מהו Autoencoders, כיצד הוא עובד ומה החסרונות שלו.

7.1.1 Dimensionality Reduction

הרבה פעמים יש דאטא ממימד גבוה התלוי תלוי בהמון פרמטרים, כאשר לא כל הפרמטרים משפיעים בצורה זהה. לדוגמא – מחיר מניה של חברה מסוימת מושפע מהמון גורמים, אך ככל הנראה סכום ההכנסות של החברה משפיע על המניה הרבה יותר מאשר הגיל הממוצע של העובדים. כיוון שקשה לנתח דאטא ממימד גבוה ולבנות מודלים בעזרת דאטא כזה, הרבה פעמים מנסים להוריד את המימד של הדאטא תוך איבוד מינימלי של מידע. בתהליך הורדת המימד מנסים לקבל ייצוג חדש של הדאטא ממימד יותר נמוך, כאשר הייצוג הזה נבנה על יד הפרמטרים הכי משמעותיים, מתוך שאיפה שהוא ידמה כמה שיותר לדאטא המקורי. יש מגוון שיטות להורדת המימד של הדאטא, ורובן פועלות בשני שלבים – ראשית מעבירים את הדאטא לייצוג ממימד נמוך יותר (מימד זה נקרא גם המרחב הלטנטי), ולאחר מכן מחזירים אותו למימד המקורי עם פחות פרמטרים.

התהליך הדו-שלבי נראה כך: דאטא $x \in \mathbb{R}^n$ נכנס למערכת ועובר דרך encoder, ומתקבל $e(x) \in \mathbb{R}^m, (m < n)$. לאחר מכן התוצאה עוברת ב-decoder על מנת להחזיר את התוצאה למימד המקורי, ומתקבל $d\big(e(x)\big) \in \mathbb{R}^n$. אם לאחר התהליך מתקיים $x = d\big(e(x)\big)$ אז למעשה לא נאבד שום מידע בתהליך, ואם $x \neq d\big(e(x)\big)$ אז מידע מסוים אבד עקב הורדת המימד ולא היה ניתן לשחזר אותו בפענוח.



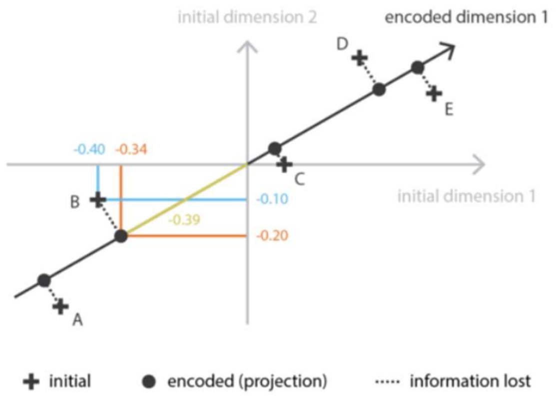
איור 7.1 ארכיטקטורת encoder ו-decoder.

המטרה העיקרית של כל השיטות להורדת מימד היא למצוא את זוג ה-encoders-decoders השומרים על מקסימום מידע בעת הקידוד, וממילא מביאים למינימום של שגיאת שחזור בעת הפענוח. אם נסמן בהתאמה E ו-D את כל הזוגות של encoders-decoders האפשריים, ניתן לכתוב את בעיית הורדת המימד באופן הבא:

$$(e^*, d^*) = \underset{(e,d) \in E \times D}{\arg\min} \ \epsilon\Big(x, d\big(e(x)\big)\Big)$$

כאשר $\epsilon\Big(x, d\big(e(x)\big)\Big)$ הוא ההפרש בין הדאטא המקורי לבין הדאטא המשוחזר.

אחת השיטות השימושיות להורדת מימד שאפשר להסתכל עליה בצורה הזו היא Principal components analysis (PCA). בשיטה זו מטילים דאטא ממימד n למימד m על ידי מציאת בסיס אורתוגונלי במימד החדש בו המרחק בין הדאטא המקורי לדאטא בייצוג החדש הוא מינימלי.



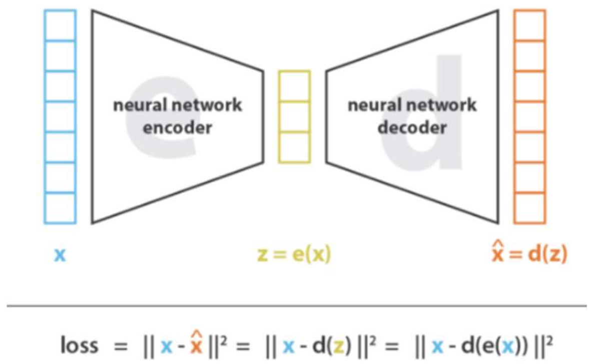
Point	Initial	Encoded	Decoded
A	(-0.50, -0.40)	-0.63	(-0.54, -0.33)
B	(-0.40, -0.10)	-0.39	(-0.34, -0.20)
C	(0.10, 0.00)	0.09	(0.07 0.04)
D	(0.30, 0.30)	0.41	(0.35, 0.21)
E	(0.50, 0.20)	0.53	(0.46, 0.27)

איור 7.2 דוגמא להורדת מימד בשיטת PCA.

במונחים של encoders-decoders, אלגוריתם PCA מחפש את ה-encoder שמבצע טרנספורמציה לינארית על הדאטא לבסיס אורתוגונלי במימד נמוך יותר, שיחד עם decoder מתאים יביא לשגיאה מינימלית. ניתן להוכיח שה-encoder האופטימלי קשור לווקטורים העצמיים של מטריצת ה-covariance של הדאטא, וה-decoder הוא השחלוף של ה-encoder.

7.1.2 Autoencoders (AE)

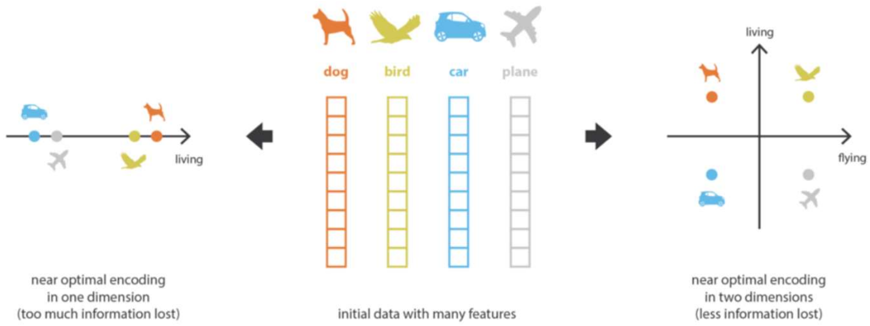
ניתן לקחת את המבנה של ה-encoder-decoder ולהשתמש ברשת נוירונים עבור קביעת הייצוג החדש ועבור השחזור. מבנה זה נראה Autoencoder:



איור 7.3 Autoencoder – שימוש ברשתות נוירונים עבור הורדת המימד והשחזור.

באופן הזה, הארכיטקטורה יוצרת צוואר בקבוק לדאטא שמבטיח שרק הפרמטרים העיקריים של הדאטא ייצגו את המידע במרחב הלטנטי ובעזרתם יתבצע השחזור. במקרה הפשוט בו בכל רשת יש רק שכבה חבויה אחת שאינה משתמשת שאקטיבציה לא לינארית, אז הורדת המימד תתבצע באופן לינארי, כלומר הרשת תחפש טרנספורמציה לינארית על הדאטא כך שיתקבל הייצוג הכי טוב ממימד נמוך יותר. בדומה ל-PCA, גם רשת כזו עובדת באופן לינארי, אך היא אינה בהכרח תייצג את הדאטא על ידי בסיס אורתוגונלי, בו הפרמטרים הם בלתי תלויים.

כעת נניח שהרשתות הן עמוקות ומשתמשות באקטיבציות לא לינאריות. במקרה כזה, ככל שהארכיטקטורה מורכבת יותר, כך הרשת יכולה להוריד יותר מימדים תוך יכולת לבצע שחזור ללא איבוד מידע. באופן תיאורטי, אם ל-encoder ול-decoder יש מספיק דרגות חופש, ניתן להפחית כל מימד לחד-מימד ללא איבוד מידע. עם זאת, הפחתת מימד דרסטית שכזו יכולה לגרום לדאטא המשוחזר לאבד את המבנה שלו. לכן יש חשיבות גדולה בבחירת מספר המימדים שבתהליך, כך שמצד אחד אכן יתבצע ניפוי של פרמטרים פחות משמעותיים ומצד שני המידע עדיין יהיה בעל משמעות. ניקח לדוגמא מערכת שמקבלת כלב, ציפור, מכונית ומטוס ומנסה למצוא את הפרמטרים העיקריים המבחינים ביניהם:



איור 7.4 דוגמא לשימוש ב-Autoencoder.

לפריטים אלו יש הרבה פיצ'רים, וקשה לבנות מודל שמבחין ביניהם על סמך כל הפיצ'רים. מעבר ברשת נוירונים יכול להביא לייצוג של כל הדוגמאות על קו ישר, כך שככל שפרט מסוים נמצא יותר ימינה, כך הוא יותר "חי". באופן הזה אמנם מתקבל ייצוג חד-מימדי, אבל הוא גורם לאיבוד המבנה של הדוגמאות ולא באמת ניתן להבין את ההפרדה ביניהן. לעומת זאת ניתן להוריד את המימד לדו-מימד ולהתייחס רק לפרמטרים "חי" ו"עף", וכך לקבל הבחנה יותר ברורה בין הדוגמאות, וכמובן שהפרדה זו היא הרבה יותר פשוטה מאשר הסתכלות על כל הפרמטרים של הדוגמאות. דוגמא זו מראה את החשיבות שיש בבחירת המימדים של ה-encoder.

7.1.3 Variational AutoEncoders (VAE)

ניתן לקחת את ה-AE ולהפוך אותו למודל גנרטיבי, כלומר מודל שמסוגל לייצר בעצמו דוגמאות חדשות. לשם כך יש לעמוד על המגבלות שלו וכיצד ניתן להתגבר עליהן. הרשתות של ה-AE מאומנות לייצג את הדאטא במימד נמוך שלוקח בחשבון את הפרמטרים העיקריים, ולאחר מכן לשחזר את התוצאה למימד המקורי, אך הן אינן מתייחסות לאופן בו הדאטא מיוצג במרחב הלטנטי. אם תוגרל נקודה כלשהו מהמרחב הלטנטי – קרוב לוודאי שהיא תהיה חסרת משמעות, כיוון שאמנם התקבל ייצוג אחר לדאטא, אך אין משמעות לייצוג הזה ללא השחזור. כדי לטפל במרחב הלטנטי, יש להשתמש ב-Variational AutoEncoders (VAE). בשונה מ-AE שלוקח דאטא ובונה לו ייצוג ממימד נמוך, VAE מנסה למצוא במרחב הלטנטי התפלגות שמתאימה לדאטא המקורי. לאחר מכן נדגמת נקודה מההתפלגות הנלמדת, ועליה מתבצע השחזור. באופן הזה, הלמידה דואגת לא רק להורדת המימד, אלא גם להתפלגות תחת המרחב הלטנטי. כאשר ההתפלגות טובה, ניתן בעזרתה גם ליצור דוגמאות חדשות, ובעצם מתקבל מודל גנרטיבי.

בפועל, ה-encoder מנסה לייצג את הדאטא המקורי באמצעות התפלגות נורמלית במימד נמוך יותר בעל תוחלת ומטריצת covariance: $e(x)\sim N(\mu_x,\sigma_x)$. התפלגות בעלת תוחלת ושונות דואגת לרגולריזציה של המרחב הלטנטי, כיוון שהתוחלת מפזרת את הדוגמאות סביב הממוצע ומזעור השונות דואג שלא יהיו במרחב החדש דוגמאות חסרות משמעות עקב שונות גדולה. באופן הזה המרחב הלטנטי מתאים לא רק לדוגמאות הנלמדות אלא יודע גם ליצור דוגמאות חדשות בעלות משמעות.

לאחר שהוצג המבנה הכללי של VAE, ניתן לתאר את תהליך הלמידה, ולשם כך נפריד בשלב זה בין שני החלקים של ה-VAE. נסתכל על ה-decoder כרשת שמקבלת המון דגימות מהסתברות מסוימת, לומדת את התוחלת והשונות ומייצרת דוגמא חדשה. ה-encoder מבצע את הפעולה ההפוכה – לומד תוחלת ושונות על בסיס הרבה דגימות. כעת נסמן את המרחב הלטנטי ב-z את הפרמטרים ה-decoder ב-θ, ואת הפרמטרים של ה-decoder ב-λ. כדי למצוא את הפרמטרים האופטימליים של הרשתות, נשתמש ב-log likelihood עבור פונקציית המחיר:

$$L(\theta)=\log\sum_i p(x_i;\theta)$$

אם נביא למקסימום את הביטוי הזה, נקבל את ה-θ האופטימלי, וכדי לעשות זאת יש להשתמש בקירוב הבא:

$$\log\sum_i p(x,z_i;\theta)=\log\sum_i q(z_i|x;\lambda)\frac{p(x,z_i;\theta)}{q(z_i|x;\lambda)}\geq\sum_i q(z_i|x;\lambda)\log\frac{p(x,z_i;\theta)}{q(z_i|x;\lambda)}$$

כאשר אי השוויון נובע מאי-שוויון ינסן, והביטוי הימני נקרא Evidence Lower BOund ($ELBO(\theta,\lambda)$). ההפרש בין שני הביטויים הוא המרחק בין שתי ההתפלגויות p,q והוא נקרא Kullback–Leibler divergence, ומסומן ב-$\mathcal{D}_{KL}$:

$$\log p(x;\theta)=ELBO(\theta,\lambda)+\mathcal{D}_{KL}(q(z|x;\lambda)\|p(z|x;\theta))$$

אם שתי ההתפלגויות שוות, אזי המרחק הוא 0 ומקבל שוויון: $\log p(x;\theta) = ELBO(\theta,\lambda)$. כזכור, מחפשים מקסימום לפונקציית המחיר $\log \sum_i p(x_i;\theta)$, וכעת בעזרת הקירוב ניתן לרשום:

$$L(\theta) = \log \sum_i p(x_i;\theta) \geq \sum_i ELBO(x_i,\theta,\lambda) = ELBO(\theta,\lambda)$$

$$L(\theta_{ML}) = \max_\theta L(\theta) \geq \max_\theta \max_\lambda ELBO(\theta,\lambda)$$

כעת ניתן בעזרת שיטת GD למצוא את האופטימום של הביטוי, וממנו לדעת את הפרמטרים האופטימליים של ה-encoder ול-decoder. נפתח יותר את ה-$ELBO(\theta,\lambda)$ עבור VAE.

$$ELBO(\theta,\lambda) = \sum_i q(z_i|x;\lambda)\log p(x,z_i;\theta) - \sum_i q(z_i|x;\lambda)\log q(z_i|x;\lambda)$$

את הביטוי $\log p(x,z_i;\theta)$ ניתן לפתוח לפי בייס $p(x,z_i) = p(x|z_i)\cdot p(z_i)$:

$$= \sum_i q(z_i|x;\lambda)(\log p(x|z_i;\theta) + \log p(z_i;\theta)) - \sum_i q(z_i|x;\lambda)\log q(z_i|x;\lambda)$$

$$= \sum_i q(z_i|x;\lambda)\log p(x|z_i;\theta) - \sum_i q(z_i|x;\lambda)(\log q(z_i|x;\lambda) - \log p(z_i;\theta))$$

$$= \sum_i q(z_i|x;\lambda)\log p(x|z_i;\theta) - \sum_i q(z_i|x;\lambda)\frac{\log q(z_i|x;\lambda)}{\log p(z_i;\theta)}$$

הביטוי השני לפי הגדרה שווה ל-$\mathcal{D}_{KL}(q(z_i|x;\lambda)\|p(z_i;\theta))$, לכן מתקבל:

$$= \sum_i q(z_i|x;\lambda)\log p(x|z_i;\theta) - \mathcal{D}_{KL}(q(z_i|x;\lambda)\|p(z_i))$$

הביטוי הראשון הוא בדיוק התוחלת של $\log p(x|z_i;\theta)$. תחת ההנחה ש-z מתפלג נורמלית, ניתן לרשום:

$$= E_{q(z|x;\lambda)}\log N\big(x;\mu_\theta(z),\sigma_\theta(z)\big) - \mathcal{D}_{KL}\big(N\big(\mu_\lambda(x),\sigma_\lambda(x)\big)\|N(0,I)\big)$$

כדי לחשב את התוחלת ניתן פשוט לדגום הרבה דוגמאות מההתפלגות $z|x\sim N\big(\mu_\theta(x),\sigma_\theta(x)\big)$, ומקבלים:

$$E_{q(z|x;\lambda)}\log N\big(x;\mu_\theta(z),\sigma_\theta(z)\big) \approx \log N\big(x;\mu_\theta(z),\sigma_\theta(z)\big)$$

ועבור הביטוי השני יש נוסחה סגורה:

$$\mathcal{D}_{KL}(N(\mu,\sigma^2)\|N(0,I)) = \frac{1}{2}(\mu^2 + \sigma^2 - \log\sigma^2)$$

המרחב הלטנטי z הוא בעל התפלגות $p(z)$, ממנו דוגמים נקודה באופן אקראי $z\sim p(z)$. לאחר מכן משחזרים את x בהינתן z, כלומר המוצא הוא בעל התפלגות $p(x|z)$, ולפי חוק בייס מתקיים:

$$p(z|x) = \frac{p(x|z)p(z)}{p(x)} = \frac{p(x|z)p(z)}{\int p(x|u)p(u)}$$

כעת נניח ש-z מתפלג נורמלית, התוחלת של $x|z$ היא פונקציה דטרמיניסטית של z והשונות של $x|z$ שווה לקבוע חיובי כפול השונות של z:

$$z\sim N(0,I) \rightarrow x|z\sim N(f(z),cI)$$

כדי לחשב את $p(z|x)$ יש צורך לדעת את $p(x)$, וכיוון שלא ניתן לחשב אותו יש לבצע קירוב.

למידה

7.2 Generative Adversarial Networks (GANs)