למידה עמוקה

תרגיל 3

גל גרנות 315681593
ניר טבת 208731422

שאלה 1

סעיף א':

ארכיטקטורת הטרנספורמר יכולה לשמש להרבה סוגים של משימות סדרתיות (sequential tasks) בהן הקלט הוא סדרה ארוכה שלרוב מפורקת לקבוצות ומעובדת כך. ניתן להשתמש בטרנספורמרים לעיבוד שפה טבעית (NLP), סיווג אודיו, ראייה ממוחשבת וכו'.

סעיף ב':

1. **Positional Encoding:** מנגנון ה-attention לא נותן בפני עצמו משמעות למיקום של כל token בתוך הרצף המקודד, זאת בניגוד למודלים כמו RNN. זו בדרך כלל תכונה לא רצויה, לדוגמא בטקסט, מכיוון שיכולה לקודד משפטים שונים סמנטית אבל דומים בצורתם בצורה דומה/זהה. בשיטת ה- positional encoding נרצה לקודד לתוך הערך של ה-token גם מידע על מיקומו בתוך הסדרה – ניתן לעשות זאת למשל ע"י שילוב פונקציות טריגונומטריות בתהליך הקידוד, מכיוון שטבען הציקלי מאפשר להחזיר מידע על המיקום של ה-token.
2. **AddNorm:** מטרת רכיב ה-AddNorm היא לעזור ביציבות המודל בזמן אימון (למשל לטפל בבעיות גרדיאנט מתפוצץ/נעלם). ע"י residual connections (חיבור המוצא של שכבה אחת למבוא של שכבה מאוחרת יותר) בחלק ה-add ולאחר מכן נרמול התוצאות בחלק ה-norm, הרכיב עוזר לטפל בבעיות יציבוצ בהם נתקל לפעמים ברשתות עמוקות.
3. **Feed Forward:** רכיב המורכב בדר"כ מ-3 שכבות – לינארית, ReLU ואז לינארית שוב הפעולות על כל ה-tokens. הרעיון הוא בדרך כלל להשתמש בשכבה הראשונה בשביל להעלות מימד בצורה משמעותית כדי ללמוד יצוגים עשירים יותר של המידע הנכנס (למשל טקסט), ואז להשתמש בשכבה הלינארית השנייה בשביל להחזיר את המידע למימד המקורי שלו לאחר עיבוד (זה חשוב גם כדי שאח"כ נוכל להשתמש בו כ- residual connection ברכיב ה-AddNorm).

סעיף ג':

תפקיד המקודד הוא לקבל את סדרת הקלט בפורמט טקסט ולהעביר אותה ליצוג רציף כלשהו שיכול לעבור ברשת (לבצע גרדיאנטים וכו'). המפענח מייצר את סדרת הפלט, בדר"כ בצורה סדרתית לכל token, כשהוא משתמש בקלט כפי שעובד ע"י הטרנספורמר והמקודד. כמו כן נשתמש במנגנון מיסוך של tokens עתידיים על מנת לוודא שהפלט נקבע ע"י tokens שהמפענח "ראה" כבר.

סעיף ד':

נחשב את המימדים:

$$X \in \mathbb{R}^{T\times d}, W_Q \in \mathbb{R}^{d\times D}, W_K \in \mathbb{R}^{d\times D} \Longrightarrow Q = XW_Q \in \mathbb{R}^{T\times D}, K = XW_K \in \mathbb{R}^{T\times D}$$

$$QK^T = Q_{T\times D}K_{D\times T} \sim \Theta_{T\times T}$$

במכפלה הזו נצטרך להכפיל T^2 פעמים וקטורים באורך D ולכן סה"כ נקבל T^2D מכפלות. בצורה דומה למכפלה השנייה:

$$X \in \mathbb{R}^{T\times d}, W_V \in \mathbb{R}^{d\times M} \Longrightarrow V = XW_V \in \mathbb{R}^{T\times M}$$

$$(QK^T)_{T\times T}V_{T\times M} \sim \Theta_{T\times M}$$

במכפלה הזו נצטרך להכפיל TM פעמים וקטורים באורך T ולכן נקבל סה"כ T^2M מכפלות. ביחד עם הקודם:

$$\#\text{multiplications} = T^2(D+M)$$

סעיף ד':

נחשב את האיבר ה-i בסדרה:

$$V_i' = \frac{\sum_{j=1}^{T}\mathrm{sim}(Q_i, K_j)V_j}{\sum_{r=1}^{T}\mathrm{sim}(Q_i, K_r)} = \frac{\sum_{j=1}^{T}\exp\left(\frac{Q_iK_j^T}{\sqrt{D}}\right)V_j}{\sum_{r=1}^{T}\exp\left(\frac{Q_iK_r^T}{\sqrt{D}}\right)} = \sum_{j=1}^{T}\frac{\exp\left(\frac{Q_iK_j^T}{\sqrt{D}}\right)}{\sum_{r=1}^{T}\exp\left(\frac{Q_iK_r^T}{\sqrt{D}}\right)}V_j$$

$$= \sum_{r=1}^{T}\mathrm{softmax}\left(\frac{Q_iK_j^T}{\sqrt{D}}\right)V_i$$

כאשר הביטוי ב-(2) מתקבל עבור הפעלת ה-softmax איבר איבר.

שאלה 2

נתבונן בביטוי הראשון:

$$E[F(u_l)|u_l] = E[W_l \cdot \mathrm{ReLU}(u_l) + b_l|u_l] = \mathrm{ReLU}(u_l)E[W_l|u_l] + E[b_l|u_l] = 0$$

כדי שהביטוי יתקיים נרצה שתוחלת המשקולות ותוחלת הביאסים תהינה אפס.

כדי שגם הביטוי השני יתקיים, נשים לב שנוכל לאתחל את המשקולות של לאפס בתוספת רעש גאוסי בעל תוחלת אפס ושונות 1 כך ששונות המשקולות היא 1 והתוחלת אפס. את הביאסים נאתחל לאפס.

שאלה 3

נחשב את הגודל המבוקש בעזרת BPTT:

$$\frac{\partial l_T}{\partial W[i,j]} = \sum_{t=1}^{T} \frac{\partial l_T}{\partial v_T} \cdot \frac{\partial v_T}{\partial v_t[i]} \cdot \frac{\partial v_t[i]}{\partial W[i,j]} = (*)$$

$$\frac{\partial v_T}{\partial v_t} = \prod_{\tau=t+1}^{T} \frac{\partial v_\tau}{\partial v_{\tau-1}} = \prod_{\tau=t+1}^{T} D_\tau W$$

$$\frac{\partial v_t[i]}{\partial W[i,j]} = D_t \cdot \frac{\partial (W v_{t-1} + B x_t)}{\partial W[i,j]} = D_t v_{t-1}$$

כאשר D_τ היא מטריצה אלכסונית:

$$D_\tau = Diag\big(\Phi'(u_\tau)\big) = Diag\big(\Phi'(W v_{\tau-1} + B x_\tau)\big)$$

וערכה:

$$D_\tau = \begin{cases} I, & x < px \\ p \cdot I, & otherwise \end{cases}$$

סה"כ נקבל:

$$(*) = \sum_{t=1}^{T} \left[\frac{\partial l_T}{\partial v_T} \left(\prod_{\tau=t+1}^{T} D_\tau W \right) D_t v_{t-1} \right][i] \cdot \frac{\partial v_t[i]}{\partial W[i,j]} = \sum_{t=1}^{T} \left[\frac{\partial l_T}{\partial v_T} \left(\prod_{\tau=t+1}^{T} D_\tau W \right) D_t v_{t-1} \right][i] \cdot D_t v_{t-1}$$

.2

משאבי חישוב גדלים:

- משאבי החישוב הנדרשים ל-BPTT גדלים ככל שאורך הsequence גדל. הסיבה לכך היא שהרשת חייבת לאחסן את האקטיבציות והגרדיאנטים עבור כל שלבי הזמן, מה שמוביל לשימוש גבוה בזיכרון.
- כדי להתגבר על כך, ניתן להשתמש ב-BPTT מקוצר, שבו מתבצע חישוב גרדיאנטים לאחור רק עבור מספר קבוע של שלבי זמן (במקום על כל הsequence). דבר זה מפחית את השימוש בזיכרון ואת זמן החישוב.
- בגרסה המקוצרת יש שגיאות קירוב מכיוון שמתעלמים מגרדיאנטים של תלויות רחוקות.

התפוצצות/דעיכת גרדיאנטים:

- ברשתות RNN, הגרדיאנטים המחושבים במהלך BPTT יכולים לגדול באופן אקספוננציאלי (התפוצצות נגזרות) או לדעוך לכמעט אפס (דעיכת נגזרות) כאשר הן מועברות אחורה לאורך שלבי זמן רבים. זה הופך את האימון ללא יציב ומקשה על למידת תלויות לטווח ארוך.
- כדי לפתור זאת ניתן להשתמש בחיתוך נגזרות. קובעים סף, וגרדיאנטים שעוברים את הסף הזה יוקטנו. גישה נוספת היא שימוש בארכיטקטורות כמו LSTM או GRU, שתוכננו להתמודד עם בעיית דעיכת הנגזרות על ידי שמירה על נגזרות יציבות יותר באמצעות מנגנוני שערים.
- כמו כן, חיתוך נגזרות מתייחס רק להתפוצצות נגזרות, ודגמי LSTM/GRU מורכבים יותר ודורשים משאבים חישוביים רבים יותר מאשר רשתות RNN רגילות.