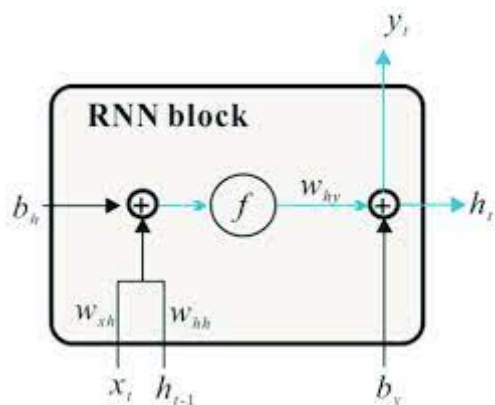


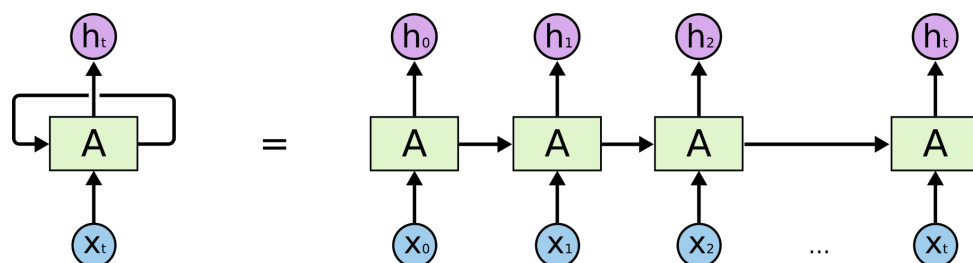
לפני שנדון בפתרונות שהוצעו לבעיה, נפתח בהצגה שלה. כאשר אנו מדברים על בעיה, עלינו להתייחס למורכבות (complexity) שלה. מורכבות של משימה הנשענת על ניתוח נתונים יכולה להתבטא בכמות המימדים, גודל הקלט, רעש, דגימות חריגות (outliers), ודפוסים מורכבים החבויים בדאטה. בהקשר זה, שפה טבעית היא בעיה עם מורכבות גבוהה, שכן היא מכילה תלות ארוכה וקצרת טווח בין מרכיבי הקלט השונים, סלנג, כפל משמעות (כדוגמת סרקזם), וסמנטיקה חבויה וגלויה. שפה מקיימת גם כללי תחביר ודקדוק, כאשר משמעות של מילה מסוימת תלויה בהקשר בו היא נאמרת, ועל כל אלו שפה אינה שומרת על רציפות באופן מוחלט. אלגוריתמים בעלי חוקיות מוגדרת מראש (rule-based) כדוגמת [Cyc](#) שניסה לפרמל פתרון לניתוח שפה באמצעות סט חוקים פנימיים, נכשלו בעקבות כך.

נדגים כעת את מורכבות ניתוח שפה, באמצעות בעיית התרגום, ואת הקושי שהיא מציפה. נניח ואנו מעוניינים לתרגם את המשפט הבא מאנגלית לעברית: "The dog sleeps" - "הכלב ישן". ניתן לתרגם כל מילה במשפט ללא תלות במילים האחרות. הכלל היחיד שעלינו להתחשב בו הינו שהמילה "the" תוסיף תחילית "ה" למילה שבאה אחריה. ולכן, נוכל לשמור מילון המכיל לכל מילה באנגלית את התרגום שלה בתוספת לחוקים שהצענו ולקבל תרגום תקין. אולם, בהינתן המשפט "We're all mad here" (עליסה בארץ הפלאות) תוצאת התרגום שבוצע בהתאם לחוקים אלו תהיה "כולנו את כל מטורף כאן". שורש הבעיה הוא תרגום המילה הנוכחית ללא התחשבות בהקשר (context), כלומר במהלך התרגום אנו חייבים להתחשב בתלויות בין המילה שאנו מתרגמים לבין המילים האחרות במשפט. ניתן להציע פתרון המשכלל את המילון שהצענו, ולשמור בנוסף על המילים והתרגום שלהם, גם צמדי מילים, אך זהו פתרון נקודתי לבעיה מבנית, ודי מהר נצטרך להרחיב את המילון לביטויים בעלי 3, 4 ויותר מילים. ולבסוף, נתכנס לכישלון המשותף לכל האלגוריתמים שניסו לפתור את הבעיה בצורת (rule-based), והוא חוסר היכולת למדל את חוקיות הבעיה בצורה יעילה.

הארכיטקטורה הראשונה שניסתה למדל את התלות בין מילים בקלט מתוארכת לסוף שנות השמונים של המאה הקודמת [2, 3], ונקראת RNN. הרעיון המרכזי מאחוריה הוא ניתוח הקלט הנוכחי, בתוספת למידע שהתקבל מהמילים האחרות במשפט. הרשת מקבלת מילה במשפט ומייצרת שני פלטים, הראשון הוא פלט המתאים למילה ביחס למשימה (תרגום, תקצור טקסט, יצירת טקסט וכו'), והשני הוא ההקשר שנצפה עד עתה ומייצג תמצות של כל המידע שהרשת קיבלה עד אז (בייצוג lossy). פתרון פשוט לבעיה זו הינו שרשור של יחידות חישוב כאלה, כך שכל יחידת עיבוד (שנקרא לה בלוק מטעמי נוחות החל מכאן) מקבלת קלט בזמן t (כלומר את הקלט במיקום ה- t במשפט/קטע טקסט) ואת ההקשר מזמן $t-1$ ומייצרת את הפלט המתאים לו. אולם, משפטים שונים מכילים מספר מילים שונה, ולכן, לא ניתן להחזיק מספר בלתי מוגבל של בלוקים כך שכל בלוק יתאים לקלט מסוימת במשפט. בנוסף בארכיטקטורה שכזו הבלוקים הראשונים יאומנו הרבה יותר מהבלוקים האחרונים, דבר הנובע מכך שבמשפטים, קצרים ממספר הבלוקים המירבי, ירופדו באפסים, והבלוקים האחרונים לא ישתתפו באופן מפורש בתהליך הלמידה. ולכן, הפתרון הינו הפעלה איטרטיבית של הרשת. כלומר שימוש בבלוק יחיד וכתוצאה מכך בסט משקולות משותף, כך שהרשת מקבלת בכל איטרציה קלט חדש ואת ההקשר שהתקבל במוצאה באיטרציה הקודמת. הייצוג הטורי מהווה פריסה (unfolding) של הרשת (ראה איור 2), בייצוג הקומפקטי: הרשת מקבלת קלט X_i ואת המצב הפנימי מהאיטרציה הקודמת שמתואר כ- H_{i-1} , ומוציאה פלט Y_i , בעוד שהמצב הפנימי H_i מוחזר בחזרה לרשת באיטרציה הבאה.



איור 1: מבנה רכיב RNN



איור 2: RNN בייצוג טורי (מימין) ובייצוג קומפקטי (unfolding)

אז כיצד משתמשים ברשת כזו לטובת ניתוח שפה? טרם השימוש מקודדים את המילים/חלקי מילים (word/subword embedding). פעולת הקידוד היא המרת מילה בשפה טבעית, לוווקטור בעל k ממדים, כאשר הייצוג הוווקטורי מקבץ מילים בעלות משמעות דומה תחת ערכים קרובים, ומרחיק מילים בעלות משמעות שונה אחת מהשנייה. [בקישור](#) הנ"ל ניתן לראות שיטות שונות לבצע זאת, כאשר הנפוצה ביותר היא שימוש ברשת נוירונים (לדוגמא word2vec). הבעיה באלגוריתמים אלו, הינה שקידוד מילים בעלות כפל משמעות יהיה זהה, דבר שמקשה על הלמידה. בהקשר זה, אחד החידושים שארכיטקטורת הטרנספורמרים הציגה היה שילוב שיכון (embedding) המילים כחלק מהארכיטקטורה, דבר שאפשר למידה של שיכון המילה ביחס למילים קרובות למילה הנוכחית, אלא גם מילים רחוקות ממנה (כלומר שיכוני מילים במרחב תלוי הקשר או contextualized embeddings).

על פניו, נראה שהרשת מספקת פתרון לבעיה, היא מסוגלת לאגור מידע ולמדל את התלות בין חתימות הזמן השונות. אז מדוע זה לא עובד? ישנן מספר סיבות האחריות לכך. הראשונה נעוצה במבנה הטורי של הרשת, שהוביל כמעט תמיד לדעיכת הגרדיאנט לאפס (vanishing gradient). כפי שהוזכר בהתחלה, במשימות ניתוח שפה הפלט עבור קלט מסוים תלוי בסדרת קלטים שקדמו לו, ולכן, בעת אימון הרשת המשקולות מעודכנות רק לאחר העברת הסדרה כולה. ברשתות איטרטיביות, יש לחשב את הגרדיאנט של המשקולות גם כן בסדר איטרטיבי אולם הפעם הוא הפוך. כדי לחשב את השפעת המשקולות על השינוי בשגיאה, עלינו לפעול בטור את הנגזרות החלקיות של כל מצב פנימי i ביחס למצב הפנימי $i-1$. מכיוון שהשינוי מחושב עבור אותו סט משקולות, הגרדיאנט יכול לדעוך או להתבדר לאורך האימון. ניתן להמחיש זאת באמצעות הערכים העצמיים של מטריצת המשקולות; במידה והם קטנים מ-1, הגרדיאנט ידעך מכיוון שמטריצת המשקולות מקטינה אותו בכל פעם שמתבצע עדכון, ובמידה והערכים העצמיים גדולים מ-1 הגרדיאנט יתבדר מכיוון שמטריצת המשקולות מגדילה אותו בכל איטרציה ([דעיכת הגרדיאנט ופתרונות RNN-ב](#)). הבעיה השנייה נובעת מהיחס בין גודל המצבים הפנימיים והיכולת שלהם לדחוס מידע, לבין גודל הסדרה שהרשת נדרשת לנתח. המצב הפנימי "דוחס" את כל המידע עד לנקודת הזמן הנוכחית, ולכן, כאשר

הסדרה עוברת סף מסוים בגודלה ומתחילות להיווצר תלויות ארוכות טווח בין מרכיביה, המצב הפנימי הופך להיות צוואר-בקבוק ברשת והיא מתחילה "לשכוח" פרטים שרחוקים מהקלט הנוכחי שלה. הבעיה האחרונה צצה במהלך השימוש ברשת לאחר האימון (inference). הרשת אינה מנצלת את מלוא הפוטנציאל החישובי שלה (ושל המכונה שהיא רצה עליה) כתוצאה מאופי קבלת הקלטים באופן טורי, דבר הגורר זמן הסקה איטי (כלומר latency גבוה) של המצבים הפנימיים במגוון משימות NLP. בעיה מהותית זו נפתרת בארכיטקטורת הטרנספורמים שנדבר עליה בהמשך.

נקודות מפתח לסיכום הפוסט:

- אלגוריתמים מבוססי חוקים (rule-based) נכשלו במשימות של ניתוח שפה טבעית עקב הקושי למדל בצורה אפקטיבית את הקשרים ששפה טבעית מכילה. ולכן, הפתרון הוא רשת שלומדת את הקשרים הללו.
- הרשת הינה יחידת עיבוד יחידה המקבלת את המצב הפנימי הקודם שלה כמצב הפנימי החדש בתוספת לקלט חדש, ולומדת ייצוג מתומצת של הקלט שראתה עד לנקודת הזמן הנוכחית.
- הרשת סובלת משני בעיות עיקריות: דעיכת גרדיאנט לאורך האימון, ואי יכולת ללמוד תלויות ארוכות טווח.
- במהלך השימוש ברשת(במהלך ההסקה), בגלל אופי המידע הטורי, הרשת לא מנצלת את מלוא המשאבים העומדים לרשותה, וכתוצאה מכך זמן עדכון המצב הפנימי איטי.