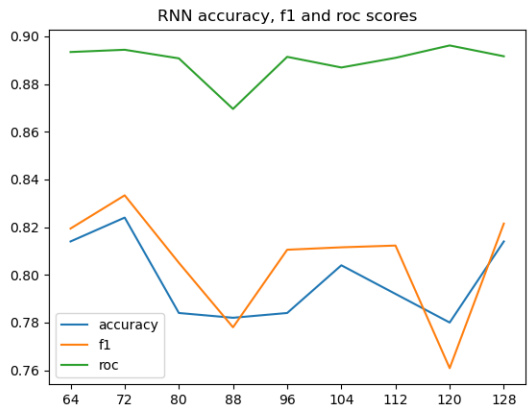
מבוא ללמידה עמוקה - תרגיל 2

עידן רפאלי ואנאל בן־סימון

8 בינואר 2021

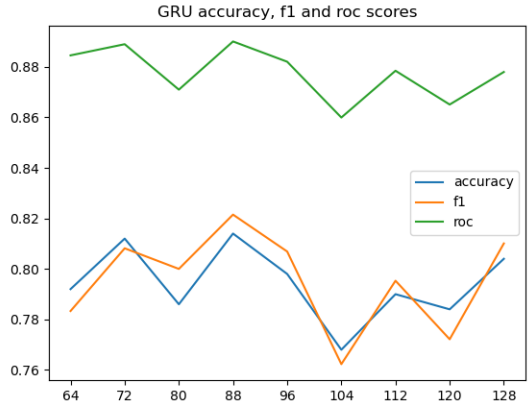
חלק תכנותי

1. לאחר שהרצנו את רשת Elman RNN הפשוטה על מימדים שונים בין 64 ל־128, קיבלנו את אחוזי הדיוק, ציוני F1 ו־ROC שניתן לראות בגרף הבא:



כפי שניתן לראות בגרף, אחוז הדיוק הגבוה ביותר, וכן ציון F1 ו־ROC הגבוהים ביותר, התקבלו כאשר המימד הוא 72. אחוז הדיוק המדויק כאשר המימד הוא 80 הוא 0.824.

לאחר שהרצנו את רשת GRU הפשוטה על מימדים שונים בין 64 ל־128, קיבלנו את אחוזי הדיוק, ציוני F1 ו־ROC שניתן לראות בגרף הבא:



כפי שניתן לראות בגרף, אחוז הדיוק הגבוה ביותר, וכן ציון F1 ו־ROC הגבוהים ביותר, התקבלו כאשר המימד הוא 88. אחוז הדיוק המדויק כאשר המימד הוא 64 הוא 0.814.

2. ניסינו מספר ארכיטקטורות שונות למודל FC עם ממוצע רגיל (ניתן לראות את כל הארכיטקטורות שבדקנו בקוד), וקיבלנו שהארכיטקטורה שמשיגה את אחוזי הדיוק הגבוהים ביותר (0.856), מבין הארכיטקטורות שבדקנו, היא כזו: רשת בעלת 5 שכבות חבויות שכל האקטיבציות שלהן הן Relu:

- שכבה חבויה 1 בגודל 100×75
- שכבה חבויה 2 בגודל 75×50
- שכבה חבויה 3 בגודל 50×50
- שכבה חבויה 4 בגודל 50×25
- שכבה חבויה 5 בגודל 25×25

להלן דוגמה בה הפרדיקציה של מודל FC עם ממוצע רגיל הטוב ביותר צודקת בתחזית שלה:

this movie is very good and interesting

זוהי ביקורת טובה, והמודל אכן מכריע על ביקורת זו Positive (ציון כללי של 0.6679857). להלן הציונים לכל מילה בביקורת:

this	-0.018179107
movie	-0.05298653
is	0.0583273
very	0.13450679
good	0.20255801
and	0.08384424
interesting	-0.25712827

המודל החליט להעניק למילים very ו־good ציון גבוה יחסית, שעזר כנראה להכריע את ההכרעה הסופית לגבי הביקורת. נשים לב כי אמנם המילה interesting רומזת על ביקורת טובה, אך המודל דווקא בחר להעניק לה ציון נמוך (אך זה לא שינה את ההכרעה שלו על הביקורת כולה).

להלן דוגמה בה הפרדיקציה של מודל FC עם ממוצע רגיל טועה בתחזית שלה:

this movie is not good at all

זוהי ביקורת רעה, אך המודל מכריע על ביקורת זו Positive (ציון כללי של 0.6416787). להלן הציונים לכל מילה בביקורת:

this	-0.018179107
movie	-0.05298653
is	0.0583273
not	-0.10613009
good	0.20255801
at	-0.008520244
all	-0.010323846

המודל החליט להעניק למילה good ציון גבוה יחסית, שעזר כנראה להכריע את ההכרעה הסופית לגבי הביקורת, אך הוא לא קישר את המילה not שהופיע לפניו למילה זו, שבעצם הופכת את משמעותה.

נשים לב שבשתי הדוגמאות, המודל החליט להעניק ציון גבוה (או נמוך) יחסית למילים שלא נראות משמעותיות וחשובות לצורך הקלסיפיקציה (למשל מילות קישור כמו and). מכיוון שהציון נקבע מסכום רגיל, המשקל של כל מילה זהה בחישוב הציון הסופי, ולכן למילים כאלו יש השפעה לא פחותה על הציון הסופי לעומת מילים יותר משמעותיות לצורך הקלסיפיקציה.

3. לאחר ששינינו את הרשת כך שהיא תכריע את הציון לכל מילה בצורה משוקללת (עם 2 פלטים במקום 1), קיבלנו שאחוז הדיוק עלה ל־0.866, וזה נובע לדעתנו מכך שהרשת יודעת כעת להבחין יותר טוב מהן המילים החשובות לצורך הקלסיפיקציה (כאלו עם משקל גבוה, כמו מילות תואר) ומהן המילים שפחות משמעותיות (כאלו עם משקל נמוך, כמו מילות קישור).

ניסינו מספר ארכיטקטורות שונות למודל FC עם ממוצע משוקלל (ניתן לראות את כל הארכיטקטורות שבדקנו בקוד), וקיבלנו שהארכיטקטורה שמשיגה את אחוזי הדיוק הגבוהים ביותר (0.866), מבין הארכיטקטורות שבדקנו, היא כזו: רשת בעלת 3 שכבות חבויות שכל האקטיבציות שלהן הן Relu:

- שכבה חבויה 1 בגודל 100×75
- שכבה חבויה 2 בגודל 75×50
- שכבה חבויה 3 בגודל 50×25

להלן דוגמה בה הפרדיקציה של מודל FC עם ממוצע משוקלל הטוב ביותר צודקת בתחזית שלה:

this movie is very good and interesting

זוהי ביקורת טובה, והמודל אכן מכריע על ביקורת זו Positive (ציון כללי של 0.8355199). להלן הציונים והמשקלים (לפני הפעלת Softmax) לכל מילה בביקורת:

word	sub-score	weight
this	2.9102345	−8.578235
movie	1.3391106	−9.815736
is	9.166944	−7.0207214
very	9.913178	−2.0933437
good	7.3660913	−1.7420683
and	13.439446	−3.345277
interesting	5.587928	−8.700354

המודל החליט להעניק למילים very ו־good ציון גבוה יחסית, וכן משקל גבוה יחסית למילים האחרות, מה שעזר כנראה להכריע את ההכרעה הסופית לגבי הביקורת. בנוסף, המודל אמנם העניק ציון גבוה יחסית למילות קישור כמו is ו־and, אבל הוא העניק להן משקל נמוך יותר, ולכן ההשפעה שלהן פחותה על הציון הסופי. המודל גם העניק משקל נמוך למילים שהן כנראה נפוצות באופן כללי בביקורות, כמו movie ו־this, ולכן ככל הנראה הן לא רומזות הרבה על אופי הביקורת. כן מעניין לציין שהמודל החליט להעניק ציון נמוך, וגם משקל נמוך יחסית למילה interesting, ואנחנו משערים שזה נובע מכך, שלפחות מהבדיקה שלנו, המילה מופיעה בהרבה ביקורות באופן כללי, וגם לאו דווקא בביקורות חיוביות או שליליות (למשל הרבה מופעים של not interesting).

להלן דוגמה בה הפרדיקציה של מודל FC עם ממוצע רגיל טועה בתחזית שלה:

this movie is not good at all

זוהי ביקורת רעה, אך המודל מכריע על ביקורת זו Positive (ציון כללי של 0.5629153). להלן הציונים לכל מילה בביקורת:

word	sub-score	weight
this	2.9102345	−8.578235
movie	1.3391106	−9.815736
is	9.166944	−7.0207214
not	−7.359451	−1.8712106
good	7.3660913	−1.7420683
at	4.559984	−13.329589
all	0.5402162	−4.6084385

המודל החליט להעניק למילה good ציון גבוה יחסית וגם משקל גבוה יחסית לאחרות, שעזר כנראה להכריע את ההכרעה הסופית לגבי הביקורת, אך הוא לא קישר את המילה not שהופיע לפניו למילה זו, שבעצם הופכת את משמעותה. גם כאן אפשר לראות שלמרות הציון הגבוה יחסית למילות הקישור is ו־at, המודל העניק להן משקל נמוך ולכן ההשפעה שלהן על הציון הסופי פחותה יותר.

לסיכום אפשר לראות שבשתי הדוגמאות, המשקל שניתן למילים שאינן משמעותיות (למשל מילות קישור כמו and, is) לצורך הקלסיפיקציה הוא נמוך יחסית, ולכן תת־הציון שלהן משפיע פחות על הציון הסופי של הביקורת, לעומת מילים משמעותיות יותר (למשל מילות תואר כמו good, horrible) שקיבלו משקל גבוה.

4. מומש בקוד

5. לאחר שהוספנו שכבת Self-Attention למודל מסעיף 3, אחוזי הדיוק שלנו עלו ל־0.874. לדעתנו זה נובע מכך שהרשת כעת יודעת למצוא את המילים הרלוונטיות לצורך הסיווג בצורה טובה יותר, ואף למצוא הקשרים בין מילים, כך שלמשל המילה not הופכת את המשמעות של המילה שמופיע לאחר מכן. אפשר לראות זאת בתוצאות בכך שהמודל מסווג את המשפט "This movie is not goot at all" בתור ביקורת שלילית, שזה אכן נכון, וזה משהו שהמודל מסעיף 3 לא הצליח לעשות.

להלן דוגמה נוספת בה הפרדיקציה של מודל FC עם שכבת Self-Attention וממוצע משוקלל צודקת בתחזית שלה:

this movie is bad and horrible

זוהי ביקורת רעה, והמודל אכן מכריע על ביקורת זו Negative (ציון כללי של 0.013739079). להלן הציונים לכל מילה בביקורת:

word	sub-score	weight
this	−12.14016	−1.3804729
movie	−16.946802	−1.8392323
is	−11.297283	−1.7930626
bad	−17.438004	−2.2817314
and	7.5154495	−2.7498808
horrible	−26.375221	−2.4981136

המודל החליט להעניק למילים bad ו־horrible ציון נמוך יחסית, ממה שעזר כנראה להכריע את ההכרעה הסופית לגבי הביקורת.
להלן דוגמה בה הפרדיקציה של מודל FC עם שכבת Self-Attention וממוצע משוקלל טועה בתחזית שלה:

the movie was mostly good but the end was disappointing

זוהי ביקורת שנחשבת לטובה ברובה, אך המודל מכריע על ביקורת זו Negative (ציון כללי של 0.009636521). להלן הציונים לכל מילה בביקורת:

word	sub-score	weight
the	1.6911106	−4.4560986
movie	−0.556497	−4.7698693
was	−0.083800495	−2.1263433
mostly	−5.183524	−1.9079744
good	2.4600856	−1.4575385
but	2.7267108	−2.679403
the	1.1359525	−5.0292974
end	4.6473975	−1.7039229
was	−1.4433919	−3.064852
disappointing	−11.622562	0.39833865

המודל החליט לתת למשפט סיווג שלילי, כנראה בגלל שהסוף של המשפט (ובפרט המילה disappointing שקיבלה ציון נמוך מאוד ומשקל גבוה יחסית) רומזות על ביקורת רעה, והוא לא הצליח להבין שלמעשה הביקורת היא טובה יחסית ברובה.

שאלות תאורטיות

1. כלל השרשרת:

(א) נחשב את הנגזרת:

$$\begin{aligned}\frac{\partial}{\partial x}f\left(x+y,2x,z\right) &= \frac{\partial f\left(x+y,2x,z\right)}{\partial x+y}\cdot\frac{\partial x+y}{\partial x}+\frac{\partial f\left(x+y,2x,z\right)}{\partial 2x}\cdot\frac{\partial 2x}{\partial x}+\frac{\partial f\left(x+y,2x,z\right)}{\partial z}\cdot\frac{\partial z}{\partial x}\\ &= \frac{\partial f\left(x+y,2x,z\right)}{\partial x+y}\cdot 1+\frac{\partial f\left(x+y,2x,z\right)}{\partial 2x}\cdot 2+\frac{\partial f\left(x+y,2x,z\right)}{\partial z}\cdot 0 =\\ &= \frac{\partial f\left(x+y,2x,z\right)}{\partial x+y}+2\cdot\frac{\partial f\left(x+y,2x,z\right)}{\partial 2x}\end{aligned}$$

(ב) נכתוב את כלל השרשרת ביחס ל־x:

$$\frac{\partial}{\partial x}f_1\left(f_2\left(\ldots f_n\left(x\right)\right)\right)=\frac{\partial f_1\left(f_2\left(\ldots f_n\left(x\right)\right)\right)}{\partial f_2\left(\ldots f_n\left(x\right)\right)}\cdot\frac{\partial f_2\left(f_3\left(\ldots f_n\left(x\right)\right)\right)}{\partial f_3\left(\ldots f_n\left(x\right)\right)}\cdot\ldots\cdot\frac{\partial f_n\left(x\right)}{\partial x}$$

(ג) נציע להשתמש בפונקציית האקטיבציה $\mathrm{Relu}\left(x\right)=\max\left(0,x\right)$. נשים לב, כי מהתשובה לסעיף ב', הנגזרת היא מכפלה של הרבה גורמים, בגלל כלל השרשרת. כל גורם הוא גזירה של פונקציית אקטיבציה, ובמקרה של Relu, היא תהיה 0 או 1 (כתלות בערך שמועבר לאקטיבציה), ולכן, במידה וכל הערכים, בכל השכבות יהיו חיוביים, נקבל גרדיינט שאינו אפס, ואפשר להשתמש בו כדי להגיע למינימום, ויש פחות סיכוי להתקל בבעיית הגרדיינט הנעלם. לעומת זאת, בפונקציות אקטיבציה אחרות, כגון סיגמאויד או tanh, הגרדיינטים שלהם חסומים בערכם, והם תמיד יהיו בין 0 ל־1 בערכם המוחלט, ומכפלה שלהם תהיה מספר קטן מאוד, ששואף ל־0. כאשר מחשבים את ערכי הגרדיינטים במחשב, הם יחושבו בפועל להיות 0, כלומר בעיית הגרדיינט הנעלם תתרחש בסיכוי גבוה הרבה יותר.

(ד) נגזור לפי כלל השרשרת:

$$\begin{aligned}\frac{\partial}{\partial x}f_1\left(x,f_2\left(x,f_3\left(\ldots f_{n-1}\left(x,f_n(x)\right)\right)\right)\right) &= \frac{\partial f_1}{\partial x}+\frac{\partial f_1}{\partial f_2}\cdot\left(\frac{\partial f_2}{\partial x}+\frac{\partial f_2}{\partial f_3}\cdot\left(\frac{\partial f_3}{\partial x}+\frac{\partial f_3}{\partial f_4}\cdot\left(\ldots\left(\frac{\partial f_{n-1}}{\partial x}+\frac{\partial f_{n-1}}{\partial f_n}\cdot\left(\frac{\partial f_n}{\partial x}\right)\right)\right)\right)\right)\\ &= \sum_{i=1}^{n}\left[\prod_{j=1}^{i-1}\left(\frac{\partial f_j}{\partial f_{j+1}}\right)\cdot\frac{\partial f_i}{\partial x}\right]\end{aligned}$$

(ה) נסמן $a = x + g(x + h(x))$, $b = x + h(x)$. נשים לב שהנגזרת של הביטוי לפי x היא:

$$\begin{aligned}
\frac{\partial f(x + g(x + h(x)))}{\partial x} &= \frac{\partial f(a)}{\partial a} \cdot \frac{\partial a}{\partial x} \\
&= \frac{\partial f(a)}{\partial a} \cdot \left(\left(1 + \frac{\partial g(b)}{\partial b} \right) \cdot \frac{\partial b}{\partial x} \right) \\
&= \frac{\partial f(a)}{\partial a} \cdot \left(\left(1 + \frac{\partial g(b)}{\partial b} \right) \cdot \left(1 + \frac{\partial h(x)}{\partial x} \right) \right) \\
&= \frac{\partial f(a)}{\partial a} \cdot \left(1 + \frac{\partial g(b)}{\partial b} + \frac{\partial h(x)}{\partial x} + \frac{\partial g(b)}{\partial b} \cdot \frac{\partial h(x)}{\partial x} \right) \\
&= \frac{\partial f(a)}{\partial a} + \frac{\partial f(a)}{\partial a} \cdot \frac{\partial g(b)}{\partial b} + \frac{\partial f(b)}{\partial b} \cdot \frac{\partial h(x)}{\partial x} + \frac{\partial f(a)}{\partial a} \cdot \frac{\partial g(b)}{\partial b} \cdot \frac{\partial h(x)}{\partial x}
\end{aligned}$$

נשים לב כי המחובר הראשון בשיוויון האחרון הוא הנגזרת של הפונקציה המקורית לפי a, והמחובר השני זה הנגזרת של הפונקציה המקורית כפול הנגזרת הפנימית שהיא g לפי b, וכן הלאה, עד שהמחובר האחרון הוא מכפלת הנגזרות של כל הרמות בהרכבה. כלומר הצלחנו לשמור לאורך כל תהליך הגזירה את ערכי הנגזרת של ההרכבות בשלבים הראשונים עד הסוף, ולא רק המכפלה שלהם, וכך הצלחנו להתגבר על בעיית הגרדיינט הנעלם.

.2

(א) עבור בעיית זיהוי דיבור, הרשת המתאימה ביותר לדעתנו היא one-to-many RNN (או לחילופין LSTM או GRU): הקלט מגיע פעם אחת בתור רצועת שמע, ובכל יחידת זמן הרשת תפלוט מילה אחר מילה את המילים שנאמרו בשמע.

(ב) עבור בעיית מענה על שאלות: הרשת המתאימה ביותר לדעתנו היא RNN (או לחילופין LSTM או GRU), כאשר אפשר להוסיף לה בהתחלה שכבת Self-Attention, כדי שהרשת תחפש את הקשר בין המילים הרלוונטיות בשאלה שמהם שהיא תוכל להסיק את התשובה. נבחין בין 2 מקרים:

i. אם מדובר במענה על שאלות עם תשובות סגורות (למשל מבחן אמריקאי, או השלמת מילה בודדת), הרשת RNN המתאימה היא מסוג many-to-one כי הקלט מגיע בתוך אוסף של מילים, והרשת תנתח את הקלט מילה אחר מילה, ובסופו של דבר תפלוט פלט יחיד שהוא התשובה שלה לשאלה.

ii. אם מדובר במענה על שאלות עם תשובות פתוחות (שמורכבות מכמה מילים), הרשת RNN המתאימה היא מסוג many-to-many כי הקלט מגיע בתוך אוסף של מילים, והרשת תנתח את הקלט מילה אחר מילה, ובנוסף היא תפלוט את התשובה שלה, המורכבת מכמה מילים, מילה אחר מילה.

(ג) עבור משימת Sentiment Analysis: הרשת המתאימה ביותר לדעתנו היא רשת עם שכבת Self-Attention (ואולי אפילו Multi-Head Self-Attention) ולאחר מכן מספר שכבות של FC. שכבת ה־Self-Attenton נועדה כדי שהרשת תבחר את המילים הרלוונטיות לסיווג מתוך כלל המילים בביקורת וגם למצוא הקשרים בין מילים (כך שהיא תבין למשל שאם המילה not מופיע לפני המילה good, אז היא למעשה הופכת את משמעותה). ראוי לציין שארכיטקטורה זו הביאה עבורנו את התוצאות הטובות ביותר מבין הארכיטקטורות שבדקנו גם בחלק התכנותי

(ד) עבור משימת זיהוי תמונות: הרשת המתאימה ביותר לדעתנו היא רשת קונבולוציה, כאשר אפשר להוסיף לה אחרי כן שכבת Attention (או Hard-Attention) ־ רשת הקונבולוציה תזהה, עם פילטרים שונים, כל מיני תבניות בתמונה, ושכבת ה־Attention תבחר את התבניות הרלוונטיות לצורך ההכרעה על האובייקט המופיע בתמונה.

(ה) עבור משימת תרגום מילה בודדת: הרשת המתאימה ביותר לדעתנו היא רשת מסוג Fully-Connected ־ במקרה הזה הקלט הוא מילה בודדת, והפלט הוא מילה בודדת, ולכן זה פחות הגיוני להשתמש ברשת RNN כלשהי, וגם שכבת Attention לא תעזור (כי כנראה שכל הקלט רלוונטי), רשת קונבולוציה לא מתאימה גם היא למשימה כמובן, ולכן לא נותר אלא להשתמש ברשת FC לצורך המשימה. נבחין כי מדובר במשימה פשוטה יחסית, שאפילו אין צורך ברשת כדי לפתור אותה, כי מספיק מילון פשוט לצורך תרגום בין מילים בודדות.

.3

(א) אנו מניחים כי קיימת רשת מסוג Auto-Encoder, שנקרא לה AE, שיודעת לקודד תמונות למרחב Latent קטן יותר, ולשחזר את התמונות ממרחב זה. נתאר את הארכיטקטורה של הרשת, שנקרא לה N, כך: הרשת N תקבל בתור קלט משפט (בתור מטריצה של מילים המקודדים על־ידי Word2Vec למשל), ותוציא כפלט וקטור השייך למרחב ה־Latent של רשת AE. לאחר מכן נקח את ה־Decoder מ־AE ונשתמש בו כדי לתרגם את הפלט של N לתמונה. הרשת N תכיל שכבת Attention, כדי שהיא תוכל לבחור ממשפט הקלט את המילים הרלוונטיות לצורך קידוד וקטור הפלט, ולאחר מכן תכיל מספר שכבות של רשת MLP.

(ב) אנו מניחים כי קיימת רשת מסוג Auto-Encoder, שנקרא לה AE, שיודעת לקודד תמונות למרחב Latent קטן יותר, כך שכל רבע מהקידוד מתייחס לרביע של התמונה, ולשחזר את התמונות ממרחב זה (כל רביע של התמונה משוחזר מהרבע המתאים בקידוד). נתאר את הארכיטקטורה של הרשת, שנקרא לה N, כך: הרשת N תקבל בתור קלט משפט (בתור מטריצה של מילים

המקודדים על־ידי Word2Vec למשל), ותוציא כפלט וקטור השייך למרחב ה־Latent של רשת AE (כלומר וקטור עם 4 חלקים, כל חלק מתייחס לרביע שונה בתמונה). לאחר מכן נקח את ה־Decoder מ־AE ונשתמש בו כדי לתרגם את הפלט של N לתמונה. הרשת N תכיל שכבת Attention, ולאחר מכן תכיל רשת LSTM, שתפקידה לייצר בכל פעם רבע מהקידוד של התמונה, כך שבסהכ הרשת תרוץ למשך 4 צעדים, והקידוד המלא יהיה שרשור של ארבעת הפלטים מכל צעד. **השאילתה (ה־query)** לשכבת ה־Attention תגיע מה־hidden state של ה־LSTM, ו**המפתחות (ה־keys)** ו**הערכים (ה־values)** יהיו המילים ממשפט הקלט (מיוצגים על־ידי הקידוד שלהם). הרעיון הוא שה־hidden state יקודד בתוכו איזה רביע של תמונה הוא הולך לצייר בצעד הבא, והוא מבקש משכבת ה־Attention את המילים הרלוונטיות במשפט לצורך ציור אותו רביע.