מגישות: אריאל ברוך, מיכל בלבן

ת"ז: 205925704, 316327659

Intro to Deep Learning – תרגיל 2:

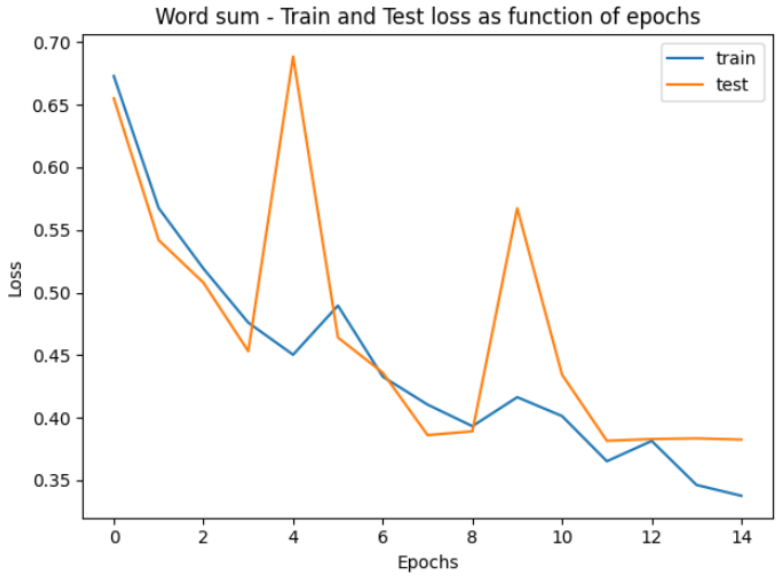
1. מימוש RNN ו-GRU :

- RNN – רשת הRNN שלנו ממומשת לפי הפסאודו קוד הבא מהמצגת:

```
For all t=0,1..len(x):
    ht=Sigmoid(Wx(x[t])+Wh(h[t-1]))
```

אבל לאחר שהשווינו עם פונקציות אקטיבציה אחרות, בחרנו להשתמש בReLU שנתן את התוצאות הטובות ביותר.
לאחר מכן אנחנו מעבירים את הפלט (הh האחרון) בשתי שכבות FC, אחת ReLU והשניה sigmoid ועל סמך התוצאה קובעים את הפרדיקציה למשפט.

ביצועי הרשת טובים יחסית אך פחות מאשר GRU שהוא מודל מורכב יותר. ניסינו לשנות את מימד שכבת ה- hidden state לערכים שונים שבטווח בין 64 ל128 וקיבלנו כי התוצאות הכי טובות הן עבור מספר נוירונים של 128.
למשל, להלן גרף המציג train loss וtest loss כפונקציה של מספרים הepochs עבור 128 נוירונים:



הדיוק אליו התכנסה הרשת הוא 0.81.
לשם השוואה, בהרצה עם 64 נוירונים הרשת התכנסה לדיוק של 0.608 וtest loss של 0.64 כלומר ההבדלים ניכרים מאוד.

- GRU – רשת הGRU מממומשת לפי הפסאודו קוד הבא:

```
For all t=0,1..len(x):

    zt=Sigmoid (Wzx(x[i])+Wzh(h[i-1]))

    rt= Sigmoid(Wrx(x[i])+Wrh(h[i-1]))
```

ht = Tanh(Wx(x[i])+Wh(rt* h[i-1]))

h =(1-z)* h[i-1]+zt*ht

ובאופן זהה לRNN מעבירים את הפלט (הh האחרון) בשתי שכבות FC, אחת ReLU והשניה sigmoid ועל סמך התוצאה קובעים את הפרדיקציה למשפט.
ביצועי הרשת טובים יותר מ- RNN, אבל הרשת עושה overfitting מאוד מהר אחרי 4 epochs, כפי שניתן לראות בגרפים הבאים בהם הloss של הtrain יורד מתחת לtest.
(עצם העובדה שהגענו לoverfitting הוא חיובי במובן מסוים, שכן הדאטא מספיק עשיר)
עם 128 נוירונים התופעה הloss של הtest עולה משמעותית לאחר מכן, כנראה כי יש יותר overfitting בגלל מספר הנוירונים הכפול.

לסיכום, הרצנו עבור הערכים הבאים למשך 4 epochs וקיבלנו את התוצאות הבאות עבור test:

Neurons	Accuracy	Loss	F1-score
64	0.8580	0.3085	0.9108
96	0.8648	0.3270	0.9364
128	0.8780	0.3054	0.9489

מסקנה: עבור כל הערכים קיבלנו תוצאות טובות עם 128 נוירונים, אך בהכרח כאשר מספר האיטרציות קטן (במקרה זה 4)

2. Word-wise FC layers –

מימשנו global average pooling , העברנו כל מילה ברשת FC לקבלת ניקוד לכל מילה, וכדי לקבל פרדיקציה לרגש המשפט ביצענו ממוצע על תוצאות כל המילים.

בבניית הרשת הfully connected הסתמכנו על מבנה הword embeddings. בייצוג בGlove וקטור ייצוג של מילה הוא וקטור ההסתברויות של כל מילה להופיע עם מילה אחרת, אבל באופן דחוס

למימד ששולחים. הוקטור בנוי בצורה שמילים בעלות משמעות דומה יהיו בעלות וקטור embedding דומה.

הרשת המורכבת מ-4 שכבות FC, אשר פועלות במקביל ובאופן ב"ת על כל המילים במשפט, ומחזירות תוצאה לכל מילה בשכבה הנקראת sub-score. להלן מבנה הרשת:

- 100 נוירונים עם אקטיבציית ReLU – המטרה בשכבה זו היא לחלק את וקטור הembedding של המילה למרכיבים שלו.
- 10 נוירונים עם אקטיבציית ReLU – שכבה זו מבצעת סכום ממושקל של 10 ממרכיבי המילה בembedding. (שורש של מס המרכיבים)
- 5 נוירונים עם אקטיבציית ReLU
- נוירון פלט ללא אקטיבצייה (כלומר, הפונקציה היא לינארית). ראינו כי כאשר משתמשים בפונקציית אקטיבציה, מילים בעלות ניקוד שלילי(מילים המביעות רגש שלילי) מתאפסות, מה שגרם לאותה השפעה של כל המילים בעלות הניקוד הנמוך על הניקוד הכללי של המשפט. אבל ראינו גם שללא אקטיבציה מילים אשר מביעות רגש שלילי כגון bad מקבלות ניקוד מאוד שלילי -70 בעוד שמילים אשר קיבלו ניקוד שלילי, אבל הן לא מילים שליליות באמת(כנראה הופיעו עם משפטים שליליים) קיבלו ניקוד שלילי חלש יותר -1. לכן כדי שכל מילה במשפט תוכל להשפיע על הסכום כמידת השליליות שהמודל נתן לה, החלטנו לא להפעיל אקטיבציה על השכבה האחרונה. היות ומפעילים sigmoid על הפרדיקציה הסופית, עדיין מקבלים בסוף הסתברות לרגש המשפט.

לאחר מכן הפעלנו ממוצע על הערך של כל המילים במשפט, ועל התוצאה הפעלנו את פונקציית הסיגמואיד כדי לקבל את הפרדיקציה לגבי המשפט כולו.

ביצועי המודל טובים אך פחות מאשר הביצועים של GRU (לאחר 15 איטרציות אימון ושימוש בארכיטקטורה המפורטת לעיל)

Accuracy	Loss	F1-score
0.842	0.3443	0.9211

משפט עליו המודל טעה:

"this film should be brilliant it sounds like a great plot the actors are first grade and the supporting cast is good as well and the main actor is attempting to deliver a good performance however it turned out to be pretty bad"

במשפט זה מורכב ברובו ממילים חיוביות, ורק הסוף שלו מביע רגש שלילי, ולכן, היות והמודל מתייחס לכל מילה בגפה, הניקוד של המשפט הזה הוא גבוה. ניתן לראות זאת בsub score הניתן למילה בכך שיש קפיצות בערכים שמתקבלים בין מילה למילה (בניגוד למודל בו עושים שימוש ב- self attention לדוגמא):

sentence prediction 0.73126096

Word number 0 is this sub_score: 2.8685884

Word number 1 is film sub_score: 0.09634846

Word number 2 is should sub_score: -8.681756

Word number 3 is be sub_score: 0.09634846

Word number 4 is brilliant sub_score: 49.580303

Word number 5 is it sub_score: 12.069657

Word number 6 is sounds sub_score: 6.3308616

Word number 7 is like sub_score: 0.09634846

Word number 8 is a sub_score: 3.5997794

Word number 9 is great sub_score: 48.41552

Word number 10 is plot sub_score: -16.008621

Word number 11 is the sub_score: 1.7708675

Word number 12 is actors sub_score: -16.252384

Word number 13 is are sub_score: 0.6113868

Word number 14 is first sub_score: 2.9071019

Word number 15 is grade sub_score: -29.00854

Word number 16 is and sub_score: 9.606562

Word number 17 is the sub_score: 1.7708675

Word number 18 is supporting sub_score: -3.5333467

Word number 19 is cast sub_score: -3.7669044

Word number 20 is is sub_score: 6.817626

Word number 21 is good sub_score: 27.969967

Word number 22 is as sub_score: 2.3094578

Word number 23 is well sub_score: 13.075874

Word number 24 is and sub_score: 9.606562

Word number 25 is the sub_score: 1.7708675

Word number 26 is main sub_score: -8.9449625

Word number 27 is actor sub_score: -13.685805

Word number 28 is is sub_score: 6.817626

Word number 29 is attempting sub_score: -24.284775

Word number 30 is to sub_score: 0.09634846

Word number 31 is deliver sub_score: 14.580749

Word number 32 is a sub_score: 3.5997794

Word number 33 is good sub_score: 27.969967

Word number 34 is performance sub_score: 11.354047

Word number 35 is however sub_score: 3.3614128

Word number 36 is it sub_score: 12.069657

Word number 37 is turned sub_score: -8.554055

Word number 38 is out sub_score: 0.09634846

Word number 39 is to sub_score: 0.09634846

Word number 40 is be sub_score: 0.09634846

Word number 41 is pretty sub_score: -7.832378

Word number 42 is bad sub_score: -67.24648

משפט אשר המודל צדק לגביו:

"this movie was so bad i coudn't wait to get out of the theatre"

משפט זה מכיל רק מילת תיאור אחת והיא בעלת רגש שלילי (קיבלה ניקוד נמוך בהתאם, וראו הדגשה) היות וכך המודל יכל לזהות שמדובר במשפט בעל רגש שלילי.

Word number 0 is this sub_score: 2.8685884

Word number 1 is movie sub_score: -5.6515465

Word number 2 is was sub_score: 0.09634846

Word number 3 is so sub_score: 7.461079

Word number 4 is bad sub_score: -67.24648

Word number 5 is i sub_score: 7.8979764

Word number 6 is coudn't sub_score: -2.011875

Word number 7 is wait sub_score: 0.09634846

Word number 8 is to sub_score: 0.788424

Word number 9 is get sub_score: 0.09634846

Word number 10 is out sub_score: 0.09634846

Word number 11 is of sub_score: 1.7708675

Word number 12 is the sub_score: -9.346082

Word number 13 is theatre sub_score: 0.46304414

3. Weighted word-wise FC layers

במודל זה השתמשנו באותה רשת fully connected מהסעיף הקודם רק שהפעם הוצאנו בשכבה האחרונה 2 ערכים, באחד השתמשנו כ sub score ובשני כמשקל כאשר אנו סוכמים את ניקוד כל המילים.

התוצאות של המודל הזה לא היו שונות מהרשת ללא המשקלים.

Accuracy	Loss	F1-score
0.836	0.3546	0.8285

ציפינו שהוספת המשקלים תשפר את הרשת ותאזן את ההשפעות של המילים אשר לא מביעות רגש ספציפי לעומת מילים אשר מביעות רגש באופן חזק, אבל התוצאות יצאו בערך אותו הדבר מבחינה מספרית (ערכי דיוק וloss) וגם הביצועים על המשפטים שבדקנו בסעיף קודם היו זהים.

4. Self-attention layer

מימשנו שכבת self-attention שמתייחסת לכל 5 מילים הכי קרובות בכל צד. לאחר הפעלת השכבה השתמשנו בשכבות שהוגדרו בסעיף 3 לקבלת הפרדיקציה.

להלן ביצועי הרצה של הרשת:

Accuracy	Loss	F1-score
0.828	0.3887	0.8259

כלומר, הביצועים טובים אך לא שיפור ביחס למודלים שבחנו קודם. ניתן דוגמא לפרדיקציה נכונה:

"this was a bad movie only for 10 seconds the rest from the beginning to the end was great"

זוהי ביקורת חיובית ואכן המודל חזה אותה ככזאת למרות שיש בה מילים שליליות, בניגוד למודלים מסעיפים 2,3 שמצאנו כי טעו בה וסיווגו אותה כביקורת שלילית.

המילה השלילית "bad" ורוב המילים בתחילת המשפט קיבלו score שלילי, אבל המשך המשפט קיבל score ניטרלי-חיובי מה שאיזן את הפרדיקציה הסופית להיות חיובית, יתכן שבזכות המאפיין של attention layers שמאפשר להתבונן בחלון סביב מילה וכך לקבוע את הפרדיקציה שלה, ולכן הייתה חלוקה מובהקת לחלק חיובי וחלק שלילי במשפט שניכרת בבלוק של מילים עם דירוג שלילי ולאחריו בלוק של מילים עם דירוג חיובי (בניגוד למודל word sum שבו כל מילה קיבלה score בנפרד).

להלן פירוט הscores:

Word number 0 is this sub_score: -18.518974

Word number 1 is was sub_score: -13.964624

Word number 2 is a sub_score: -8.453243

Word number 3 is bad sub_score: -20.500809

Word number 4 is movie sub_score: -12.02885

Word number 5 is only sub_score: -7.042422

Word number 6 is for sub_score: -2.4578335

Word number 7 is 10 sub_score: 0.5012445

Word number 8 is seconds sub_score: 0.43527746

Word number 9 is the sub_score: 0.34326753

Word number 10 is rest sub_score: 0.25650057

Word number 11 is from sub_score: 0.7721039

Word number 12 is the sub_score: 3.2634058

Word number 13 is beginning sub_score: 1.8405682

Word number 14 is to sub_score: 1.8003718

Word number 15 is the sub_score: 1.4710001

Word number 16 is end sub_score: 2.7864642

Word number 17 is was sub_score: 7.8923707

Word number 18 is great sub_score: 3.2624881

להלן דוגמא לטעות של המודל:

"this film should be brilliant it sounds like a great plot the actors are first grade and the supporting cast is good as well and the main actor is attempting to deliver a good performance however it turned out to be pretty bad"

למרות שהביקורת שלילית המודל חזה אותה כחיובי, ולפי הscore שניתן למילים ניתן לראות שהוא כן הצליח להבדיל בין הצדדים החיוביים והשליליים במשפט, אבל כיוון שהצד החיובי היה משמעותית יותר ארוך מהשלילי הוא הסתכם ליותר score שהשפיע על הפרדיקציה הסופית שנקבעה כחיובית.

Word number 0 is this sub_score: 6.75382

Word number 1 is film sub_score: 7.9254704

Word number 2 is should sub_score: 3.7875965

Word number 3 is be sub_score: 5.204723

Word number 4 is brilliant sub_score: 20.561337

Word number 5 is it sub_score: 16.745453

Word number 6 is sounds sub_score: 12.769249

Word number 7 is like sub_score: 7.8971047

Word number 8 is a sub_score: 10.604309

Word number 9 is great sub_score: 19.877983

Word number 10 is plot sub_score: 2.310594

Word number 11 is the sub_score: 4.938337

Word number 12 is actors sub_score: 7.8618913

Word number 13 is are sub_score: 8.77944

Word number 14 is first sub_score: 8.733805

Word number 15 is grade sub_score: -2.4377465

Word number 16 is and sub_score: 4.8943133

Word number 17 is the sub_score: 3.933766

Word number 18 is supporting sub_score: 6.924643

Word number 19 is cast sub_score: 5.0638723

Word number 20 is is sub_score: 3.0619466

Word number 21 is good sub_score: 13.874009

Word number 22 is as sub_score: 6.761789

Word number 23 is well sub_score: 10.03711

Word number 24 is and sub_score: 0.21717016

Word number 25 is the sub_score: -0.82072407

Word number 26 is main sub_score: 9.64172

Word number 27 is actor sub_score: 7.7832704

Word number 28 is is sub_score: 9.736155

Word number 29 is attempting sub_score: -1.0658919

Word number 30 is to sub_score: 3.7310195

Word number 31 is deliver sub_score: 10.317105

Word number 32 is a sub_score: 3.4869523

Word number 33 is good sub_score: 3.0969548

Word number 34 is performance sub_score: -2.2487247

Word number 35 is however sub_score: -2.0030382

Word number 36 is it sub_score: 3.1561756

Word number 37 is turned sub_score: -11.528616

Word number 38 is out sub_score: -12.095428

Word number 39 is to sub_score: -10.119557

Word number 40 is be sub_score: -7.476245

Word number 41 is pretty sub_score: -11.148399

Word number 42 is bad sub_score: -14.265566

<u>השוואה בין המודלים:</u>

ניתן לראות שהביצועים הטובים ביותר התקבלו עבור GRU, לאחר מכן שאר המודלים עם שכבות הFC שלא הראו הבדלים משמעותיים בaccuracy הכולל (אך כן ראינו מצאנו הבדלים בביצועים עבור משפטים ספציפים) ולבסוף RNN. עבור כל המודלים הבחנו בדוגמאות שסווגו בצורה שגויה בשל בעיות שנובעות מהנחות המודל על המשפט.

מבוא ללמידה עמוקה | תרגיל 2

מגישות: מיכל בלבן, אריאל ברוך | ת"ז: 205925704, 316327659

8 בינואר 2021

1

1.1 כתבו את הנגזרת הבאה: $\frac{\partial}{\partial x} f(x+y, 2x, z)$

נשתמש בכלל השרשרת:

$$\frac{\partial f}{\partial x} = \frac{\partial f}{\partial (x+y)} \cdot \frac{\partial (x+y)}{\partial x} + \frac{\partial f}{\partial (2x)} \cdot \frac{\partial (2x)}{\partial x} + \frac{\partial f}{\partial z} \cdot \frac{\partial z}{\partial x}$$

נשים לב כי:

$$\frac{\partial (x+y)}{\partial x} = 1, \quad \frac{\partial (2x)}{\partial x} = 2 \ , \frac{\partial z}{\partial x} = 0$$

לכן קיבלנו סהכ:

$$= \frac{\partial f}{\partial (x+y)} + 2\frac{\partial f}{\partial (2x)}$$

1.2 כתבו את כלל השרשרת בהתאמה לx עבור הפונקציה: $f_1(f_2(...f_n(x)))$

מכלל השרשרת:

$$\frac{\partial f_1(f_2(...f_n(x)))}{\partial x} = \frac{\partial f_1}{\partial f_2} \cdot \frac{\partial f_2}{\partial f_3} \cdot ... \cdot \frac{\partial f_n}{\partial x}$$

1.3 באיזו אקטיבציה נשתמש כדי לייצב את הגרדיאנט?

לייצב את הגרדיאנט משמעו למנוע מצב של גרדיאנטים נעלמים\מתפוצצים. הבעיה הראשונה עלולה לקרות בשימוש בפונקציות אקטיבציה חסומות בטווח בין $[0,1]$ או $[-1,1]$ (כמו $tanh$ ו$sigmoid$) הבעיה השניה עלולה לקרות כאשר משתמשים בפונקציה לא חסומה כמו $ReLU$. לכן כדאי במצב של גרדיאנטים נעלמים לשלב שימוש באקטיבציית $ReLU$ ובמצב של גרדיאנטים מתפוצצים לשלב פונקציות חסומות. זה בנוסף לפתרונות אחרים לבעיה כמו skip connections.

1.4 כתבו את הביטוי שמתקבל מההרכבה הבאה: $f_1(x, f_2(x, f_3, (..., f_{n-1}(x, f_n(x))..)$

נשתמש בכלל השרשרת:

$$\frac{\partial f_1(x, f_2(x, f_3, (..., f_{n-1}(x, f_n(x))..)}{\partial x} =$$

$$\frac{\partial f_1(x, f_2(x, f_3, (..., f_{n-1}(x, f_n(x))..)}{\partial x} \cdot \frac{\partial x}{\partial x} + \frac{\partial f_1(x, f_2(x, f_3, (..., f_{n-1}(x, f_n(x))..)}{\partial f_2(x, f_3(..., f_{n-1}(x, f_n(x))..)} \cdot \frac{\partial f_2(x, f_3(..., f_{n-1}(x, f_n(x))..)}{\partial x} =$$

נסמן בy_1 את הקלט של f_1, בy_2 את הקלט של f_2 וכו':

$$\frac{\partial f_1(y_1)}{\partial x} + \frac{\partial f_1(y_1)}{\partial f_2(y_2)} \cdot \left(\frac{\partial f_2(y_2)}{\partial x} \cdot \frac{\partial x}{\partial x} + \frac{\partial f_2(y_2)}{\partial f_3(y_3)} \frac{\partial f_3(y_3)}{\partial x} \right) =$$

$$= .... =$$

$$\frac{\partial f_1(y_1)}{\partial x} + \frac{\partial f_1(y_1)}{\partial f_2(y_2)} \cdot \frac{\partial f_2(y_2)}{\partial x} + ... + \frac{\partial f_{n-1}(y_{n-1})}{\partial f_n(y_n)} \cdot \frac{\partial f_n(y_n)}{\partial x}$$

1.5 הסבירו מה טוב לגבי הביטוי $f(x + g(x + h(x)))$ לעומת הרכבה פשוטה של f, g, h

במצב של הרכבת פונקציות, לפי כלל השרשרת אנחנו מכפילים הרבה ערכים האחד בשני. אבל כאשר השרשרת ארוכה ואנחנו מכפילים הרבה ערכים שקטנים מ1 אנחנו נמצאים בבעיה של vanishing gradients.
כאשר משתמשים בskip connections מסוג $ResNet$ אנחנו מעבירים גרדיאנטים משכבות חיצוניות לשכבות פנימיות יותר באופן ישיר וכך מפחיתים משמעותית את אפקט הדעיכה של הגרדיאנטים, כי השכבה העמוקה יותר מקבלת ערך שלא הוכפל הרבה פעמים ופוטנציאלית כבר קטן מאוד.

2

2.1 speech recognition

במקרה זה נשתמש בmany to many RNN כיוון שRNN מותאם להתמודדות עם בעיות מעולם הNLP (כלומר, אנחנו רוצים שתהיה לנו יכולת שמירת זכרון כלשהי ובכך ניצור הקשר בין חלקי המשפט השונים), והקלט הוא רצף של דיבור ללא הגבלה של מספר המילים\משפטים, והפלט בהתאם הוא הטקסט שמתאים לדיבור כלומר גם לא מוגבל באורכו.

2.2 answer questions

נשתמש גם במקרה זה בmany to many RNN כיוון שהקלט והפלט לא מוגבלים באורכם, והבעיה היא מעולם הNLP.

2.3 sentiment analysis

נשתמש בmany to one RNN במקרה בו ה sentiment שנרצה לזהות הוא חיובי או שלילי. הבעיה היא גם בעיית NLP לכן RNN מתאים ביותר לפתור אותה, והקלט הוא טקסט באורך משתנה והפלט עבורו הוא בינארי - האם הסנטימנט שבטקסט היה חיובי או שלילי. גם CNN יכול לעבוד טוב במקרה של sentiment analysis כאשר עושים embedding מתאים למילים ומייצגים אותם כוקטורים, בדומה לפיקסלים בתמונה.

2.4 image classification

נשתמש בCNN שזוהי הארכיטקטורה המתאימה ביותר לבעיות של קלאסיפיקציה ועיבוד של תמונות. משתמשים בקונבולוציות, כלומר slidint window שמעבירים על פני המטריצה (כלומר התמונה) ובכך מעבדים בכל שכבה של הרשת פיצ'רים שונים מתוך התמונה ברמה מורכבות הולכת ועולה.

2.5 single word translation

נשתמש בone to one RNN, שכן הבעיה מעולם הNLP, הקלט הוא באורך 1 ונוכל להניח שהפלט באורך אחד.

3

3.1

נניח שיש לנו $decoder$ מאומן אשר יודע להמיר וקטור מהמימד החבוי של ייצוג תמונה(low-dimensional latent space) לתמונה. נבנה רשת בצורה הבאה: נעביר את המשפט(word embeddings) מילה מילה דרך רשת RNN LSTM ולאחר מכן דרך רשת fully connected אשר תמיר את הייצוג שיצא מהRNN לייצוג ב low-dimensional latent space את הייצוג הזה נעביר ל$decoder$ ונקבל תמונה. המעבר בRNN יפרסר לנו את המשפט לייצוג אשר מתייחס למאפיינים השונים של התמונה, המעבר דרך הרשת הFC יסדר לנו את המימדים ויעביר לייצוג אשר ה$decoder$ יודע לפרש, וה$decoder$ יודע להמיר בין ייצוג תמונה בתור מאפיינים של סוג התמונות במאגר לתמונה אשר מתאימה לסוג התמונות במאגר.

3.2

באמצעות attention layer הרשת תסתכל על כל מילה רק מול האזור שהיא משפיעה/מדברת עליו, מה שאומר שלכל מילה הרשת תוכל להיות יותר מדויקת לגבי ההשפעה שלו על מבנה התמונה כי היא כבר לא מחפשת איפה הוא משפיע ביחס לכל התמונה אלא באזור קטן יותר, וכאשר הרשת תבנה את הוקטור שיעבור ל$decoder$ הattention weights יתנו לכל מילה משקל גבוה לאזור שהיא משפיעה עליו, כשסוכמים את כל המילים. ה $queries$ בattention layer שלנו יהיו לאיזה מארבעת האזורים בתמונה המילה שייכת) יכול להיות וקטור יחידה אשר מתייחס לאזור המתאים(ה$values$יהיו ייצוג של מילה כ4 וקטורי אזורים, ה$keys$ יהיו זהים ל$values$.