Practical Part

Note : results, and plots are shown in the colab notebook.

Task 1

Compare final results on validation set (without the sentences I added)

	RNN	GRU
Accuracy score	0.746	0.85
F1 score	0.7687	0.8414
ROC AUC score	0.8488	0.9233

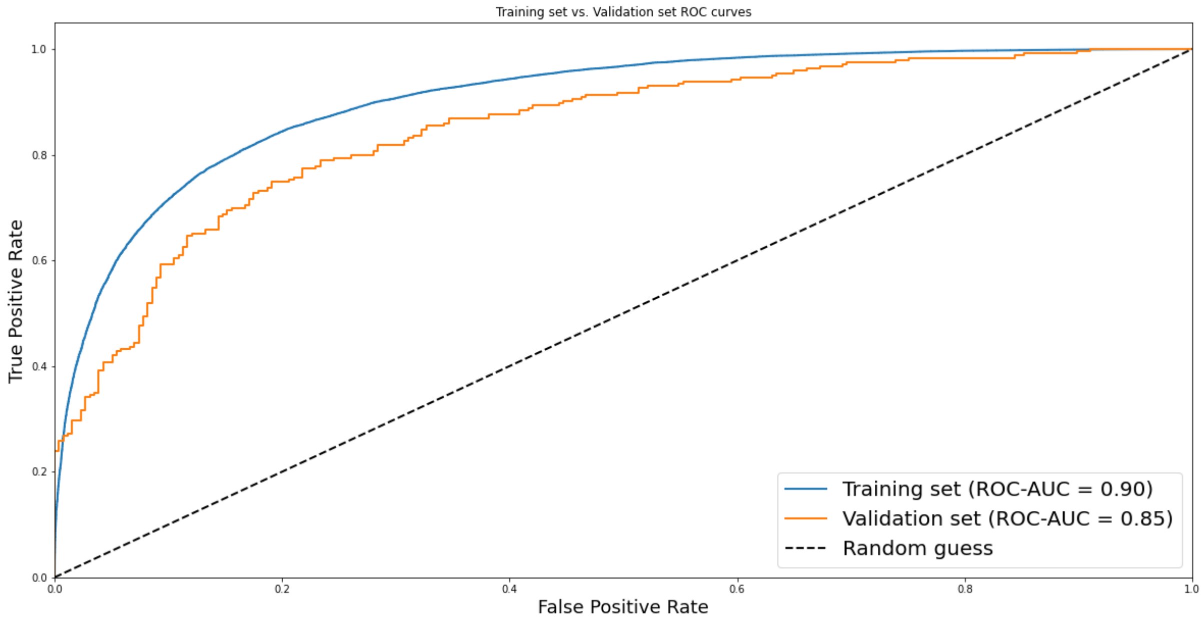
Compare final results on test set (the sentences I added)

	RNN	GRU
Accuracy score	0.6667	0.8333
F1 score	0.75	0.8571
ROC AUC score	0.625	1.0

RNN

1. Hidden-state dimension = 128 (best result for rnn)
   Results of Validation set:
   Accuracy score: 0.746
   F1 score: 0.7687
   ROC AUC score: 0.8488

   Results of Test set:
   Accuracy score: 0.6667
   F1 score: 0.75
   ROC AUC score: 0.625

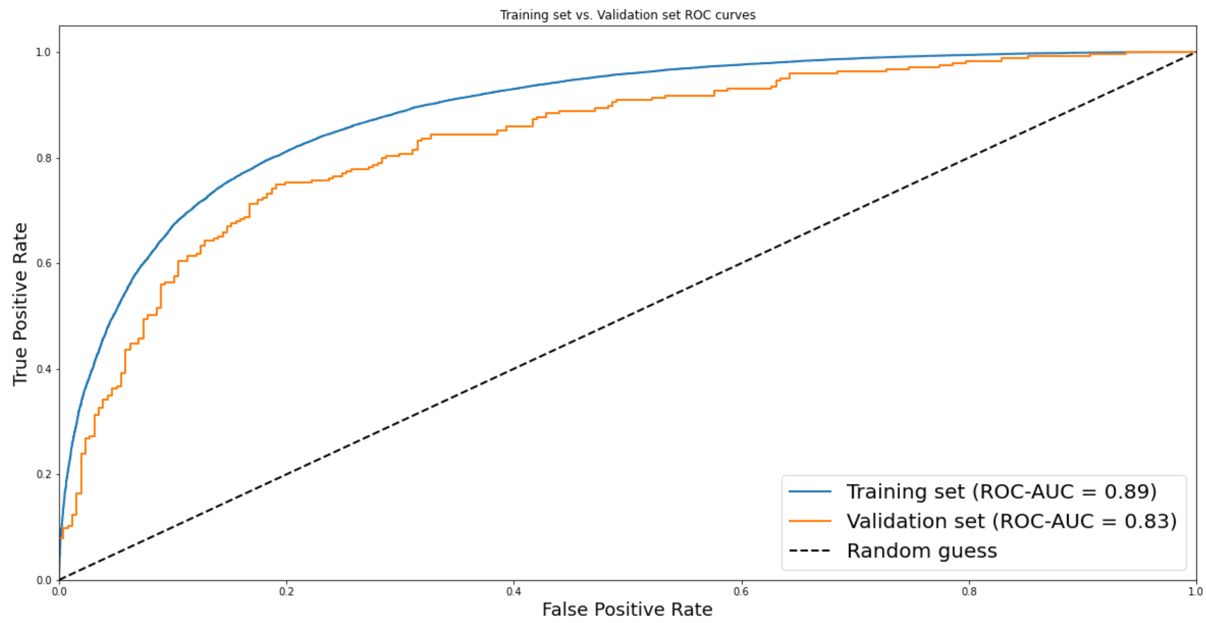


Example of my test set:

	Predictions		True Labels
['This', 'movie', 'is', 'amazing']	0.59350574	Positive	Positive
['I', 'cant', 'recommend', 'this', 'movie', 'highly', 'enough']	0.44046244	Negative	Positive
['I', 'loved', 'this', 'one']	0.5743084	Positive	Positive
['Boring', 'not', 'worth', 'the', 'time']	0.25164303	Negative	Negative
['I', 'had', 'expectations', 'I', 'thought', 'it', 'would', 'be', 'interesting', 'but', 'I', 'was', 'disappointed']	0.6438073	Positive	Negative

2. Hidden-state dimension = 64
   Results of Validation set:
   Accuracy score: 0.752
   F1 score: 0.7678
   ROC AUC score: 0.8345

   Results of Test set:
   Accuracy score: 0.5
   F1 score: 0.5714
   ROC AUC score: 0.625



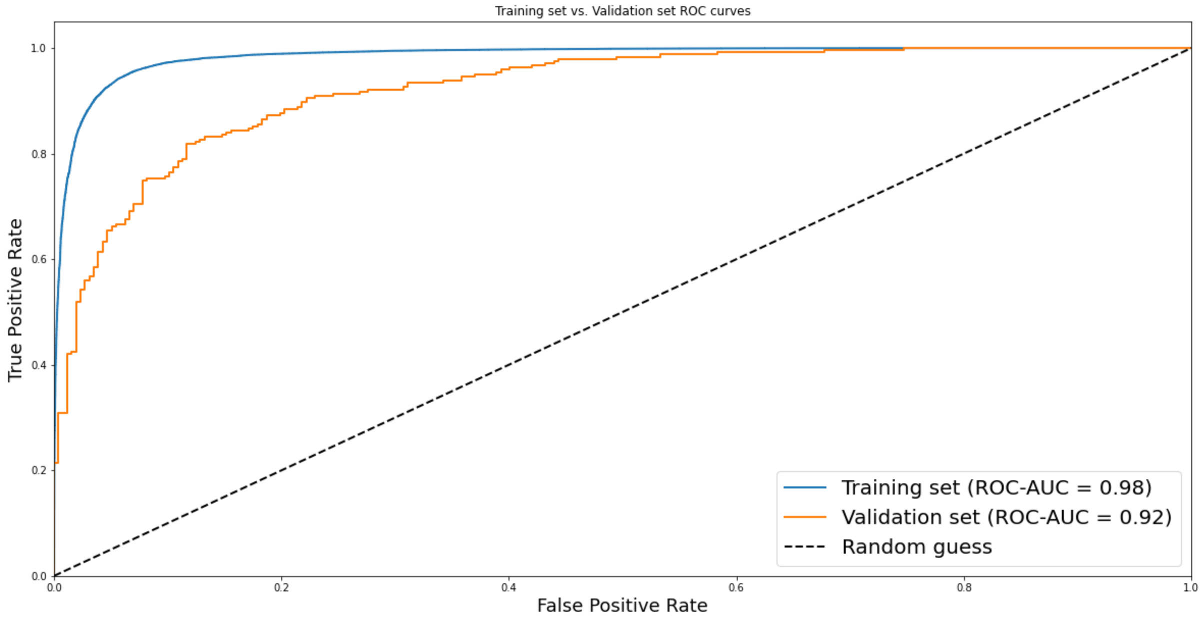
Example Predictions of the model with highest accuracy:

	Predictions		**True Labels**
['This', 'movie', 'is', 'amazing']	0.59350574	Positive	Positive
['I', 'cant', 'recommend', 'this', 'movie', 'highly', 'enough']	0.41895914	Negative	Positive
['I', 'loved', 'this', 'one']	0.4987315	Negative	Positive
['Boring', 'not', 'worth', 'the', 'time']	0.29164532	Negative	Negative
['I', 'had', 'expectations', 'I', 'thought', 'it', 'would', 'be', 'interesting', 'but', 'I', 'was', 'disappointed']	0.5592504	Positive	Negative

GRU

Hidden-state dimension = 64

Results of Validation set:
Accuracy score: 0.85
F1 score: 0.8414
ROC AUC score: 0.9233



Results of Test set:

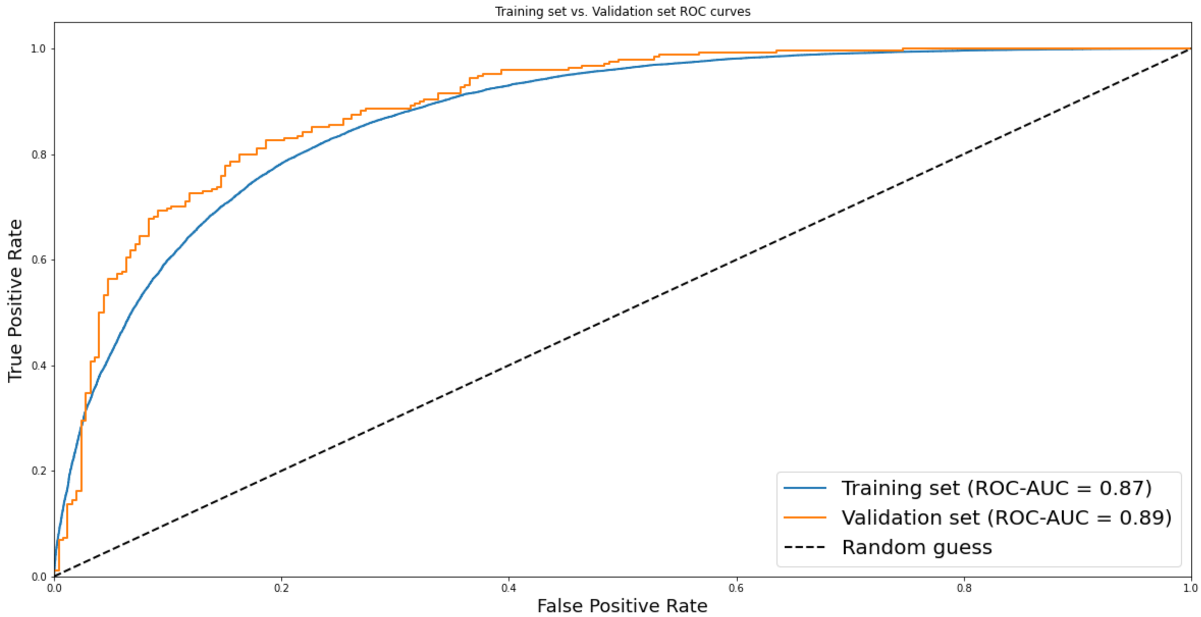
Accuracy score: 0.8333
F1 score: 0.8571
ROC AUC score: 1.0

Example Predictions:

	Predictions		**True Labels**
['This', 'movie', 'is', 'amazing']	0.97527677	Positive	Positive
['I', 'cant', 'recommend', 'this', 'movie', 'highly', 'enough']	0.00666757	Negative	Positive
['I', 'loved', 'this', 'one']	0.9776049	Positive	Positive
['Boring', 'not', 'worth', 'the', 'time']	0.00479398	Negative	Negative
['I', 'had', 'expectations', 'I', 'thought', 'it', 'would', 'be', 'interesting', 'but', 'I', 'was', 'disappointed']	0.00661436	Negative	Negative

Task 2

Results of Validation set:
Accuracy score: 0.808
F1 score: 0.8132
ROC AUC score: 0.8941



Results of Test set:
Accuracy score: 0.8333
F1 score: 0.8889
ROC AUC score: 0.875

Example Predictions:

<table>
<tr><th></th><th>Predictions</th><td rowspan="5">positive</td></tr>
<tr><td>This</td><td>0.00042231619590893339</td></tr>
<tr><td>movie</td><td>1.209144920721883e-05</td></tr>
<tr><td>is</td><td>0.0011918470263481l4</td></tr>
<tr><td>amazing</td><td>1.6441930711152963e-05</td></tr>
</table>

<table>
<tr><th></th><th>Predictions</th><th rowspan="8">Positive</th></tr>
<tr><td>I</td><td>1.6565861642447999e-06</td></tr>
<tr><td>cant</td><td>6.864743045298383e-05</td></tr>
<tr><td>recommend</td><td>3.7453403933795926e-08</td></tr>
<tr><td>this</td><td>0.00013868676614947617</td></tr>
<tr><td>movie</td><td>0.00010089482384501025</td></tr>
<tr><td>highly</td><td>0.00014803143858443946</td></tr>
<tr><td>enough</td><td>2.287507605558403e-08</td></tr>
</table>

<table>
<tr><th></th><th>Predictions</th><th rowspan="5">Positive</th></tr>
<tr><td>I</td><td>0.07947801053524017</td></tr>
<tr><td>loved</td><td>3.2623276638332754e-05</td></tr>
<tr><td>this</td><td>3.7453403933795926e-08</td></tr>
<tr><td>one</td><td>0.0016793844988569617</td></tr>
</table>

<table>
<tr><th></th><th>Predictions</th><th rowspan="6">Negative</th></tr>
<tr><td>boring</td><td>1.209144920721883e-05</td></tr>
<tr><td>not</td><td>5.591098783952475e-07</td></tr>
<tr><td>worth</td><td>7.018509495537728e-05</td></tr>
<tr><td>the</td><td>0.0017538771498948336</td></tr>
<tr><td>time</td><td>1.94239619304426e-05</td></tr>
</table>

	Predictions	
I	4.3294221541145816e-05	Positive (wrong prediction)
had	0.029746511951088905	
expectations	0.0005097261164337397	
I	0.0006210884894244373	
thought	3.7453403933795926e-08	
it	0.0005097261164337397	
would	0.0006115768919698894	
be	4.483520024223253e-05	
interesting	3.3501869438623544e-06	
but	0.00035350158577784896	
I	0.0005097261164337397	
was	0.00014803143858443946	
disappointed	0.0008100144332274795	

Consider the sentence : 'I had expectations, I thought it would be interesting, but I was disappointed'

In this example we noticed that this model classified the last sentence as positive review, although it's a negative one. The reason for this mistake could be the misunderstanding of the context. There are words like 'interesting' or 'expectations' that have positive meaning but when reading the words before and after, we can understand that actually the meaning is negative - the movie is not interesting, and the reviewer was disappointed.

Task 3

Let's compare task 2 and task 3 results :

	Task 2	Task 3
Accuracy score	0.8333	1.0
F1 score	0.8889	1.0
ROC AUC score	0.875	1.0

Now the final layer produces 2 values for each word and uses the additional scalar as a weight for summing the sub scores. This improved the test accuracy since this way the model can choose which sub score is more important. If a word is important, its weight is high and it will impact the final score of the whole sentence. This score affects the prediction that should be above 0.5 to be considered as positive.

For example the sentence : "This movie is amazing" is a positive review.
The word 'amazing' is the most important and its weight is higher compared to task 2.

	Task 2	**Task 3**
this	0.00042231	0.94605890
movie	0.00001209	1.82377719
is	0.00119184	1.04864975
amazing	**0.000064419**	**0.00756620**

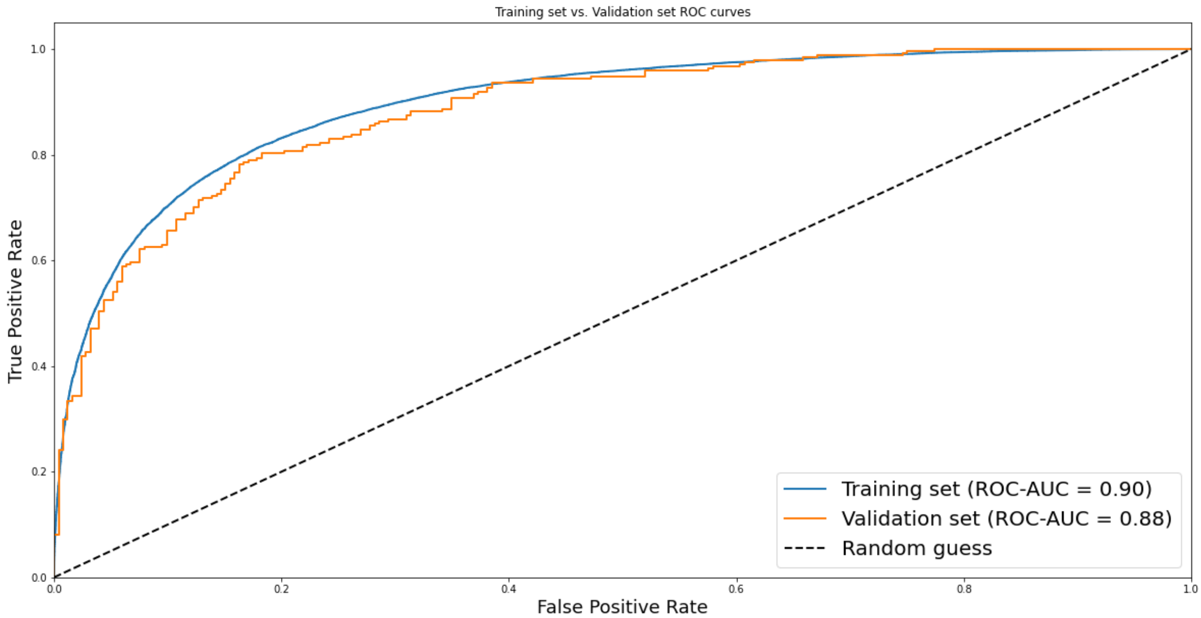
The sentence "I love this one" is a positive review.
The word 'loved' is the most important and its weight is higher compared to task 2.

	Task 2	**Task 3**
I	0.07947801	1.192771032
loved	**0.00003262**	**0.030170877**
this	3.745340e-08	1.992330312
one	0.0016793	0.400458993

The sentence "I thought it would be interesting but i was disappointed" is a negative review but task 2 marked this as a positive review.
The word 'disappointed ' is the most important and its weight is higher compared to task 2.

	Task 2	**Task 3**
disappointed	**0.000810014433227**	**1.1151685118675**

Results of Validation set:
Accuracy score: 0.806
F1 score: 0.8016
ROC AUC score: 0.884



Results of Test set:
Accuracy score: 1.0
F1 score: 1.0
ROC AUC score: 1.0

Example Predictions:

<table>
<tr><th></th><th>Predictions</th><th rowspan="5">positive</th></tr>
<tr><td>This</td><td>0.9460589078080375</td></tr>
<tr><td>movie</td><td>1.823777198791504</td></tr>
<tr><td>is</td><td>1.0486497581005096</td></tr>
<tr><td>amazing</td><td>0.007566207583295181</td></tr>
</table>

<table>
<tr><th></th><th>Predictions</th><th rowspan="8">Positive</th></tr>
<tr><td>I</td><td>1.2799495458602905</td></tr>
<tr><td>cant</td><td>1.835557758808136</td></tr>
<tr><td>recommend</td><td>1.9923303127288818</td></tr>
<tr><td>this</td><td>1.011188581585884</td></tr>
<tr><td>movie</td><td>0.8256320059299469</td></tr>
<tr><td>highly</td><td>0.2529417199548334</td></tr>
<tr><td>enough</td><td>0.6571245789527893</td></tr>
</table>

	Predictions	
I	1.1927710324525833	**Positive**
loved	0.030170877173077315	
this	1.9923303127288818	
one	0.4004589939304424	

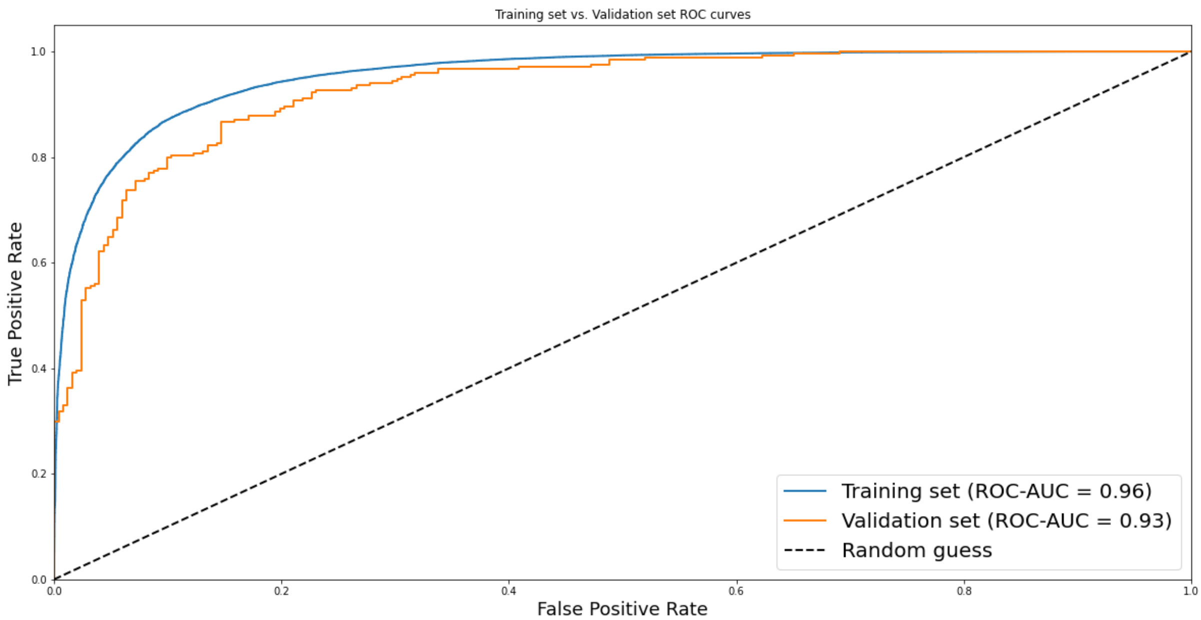
	Predictions	
boring	1.823777198791504	**Negative**
not	0.46824405749794096	
worth	0.2516891956979737	
the	1.1645564138889313	
time	0.3470552921498893	

	Predictions	
I	0.7417919631116092	**Negative**
had	0.6914840147946961	
expectations	0.35614229179918766	
I	0.46057990880217403	
thought	1.9923303127288818	
it	0.35614229179918766	
would	0.3617224246263504	
be	1.1369105577468872	
interesting	1.0108301350846887	
but	1.67457914352417	
I	0.35614229179918766	
was	0.2529417199548334	
disappointed	1.1151685118675232	

Task 5

Results of Validation set:

Accuracy score: 0.842
F1 score: 0.8384
ROC AUC score: 0.9292



Results of Test set:

Accuracy score: 1.0
F1 score: 1.0
ROC AUC score: 1.0
Example Predictions:

Example Predictions:

<table>
<tr><th></th><th>Predictions</th><th rowspan="5">positive</th></tr>
<tr><td>This</td><td>0.994184136390686</td></tr>
<tr><td>movie</td><td>0.6192785799503326</td></tr>
<tr><td>is</td><td>0.482318595983088</td></tr>
<tr><td>amazing</td><td>0.4971533864736557</td></tr>
</table>

	Predictions	
I	0.31996229849755764	
cant	0.7433578073978424	
recommend	0.9995439946651459	
this	0.6454012244939804	**Positive**
movie	1.860028624534607	
highly	1.079694151878357	
enough	0.2852265313267708	

	Predictions	
I	0.5464983582496643	
loved	0.9599781036376953	**Positive**
this	0.7846314311027527	
one	0.2868731766939163	

	Predictions	
boring	1.1153807938098907	
not	1.3042886853218079	**Negative**
worth	1.0846204161643982	
the	1.4919089674949646	
time	0.5084910094738007	

	Predictions	
I	0.19877048209309578	
had	0.6074221730232239	
expectations	0.9964616000652313	
I	0.5945707410573959	
thought	0.8110296130180359	
it	0.9962018728256226	
would	0.9965084791183472	Negative
be	1.0751320719718933	
interesting	0.29307248070836067	
but	0.9883694648742676	
I	0.9561741650104523	
was	1.2249450087547302	
disappointed	1.0710612535476685	

Discussion :

Attention Layer

1. Explain the results - how they differ from before.
   Let's compare task 3 and task 5 results :

	FC sub prediction score	FC sub prediction score + Attention layer
Accuracy score	0.806	0.842
F1 score	0.8016	0.8384
ROC AUC score	0.884	0.9292

As we can see the model accuracy score improved. Moreover when analyzing the prediction score per word , we can see that the Attention layer affects the subscore of close words.

For example the sentence : 'This movie is amazing'
Review is positive and predicted as positive.

	This	movie	is	amazing
Task 3	0.946	1.823	1.049	0.007
Task 5	0.994	0.619	0.482	0.497

The self-attention layer we added enables us to learn the correlation between the current words and the previous part of the sentence. Notice that the scores of words that have a relation are close.
For example:

Movie - which movie? this
Is - who? this movie
Amazing - who? this movie is

An apparent disadvantage of the model from task 3 is the incapability of the system to remember longer sequences. It could forget the earlier parts of the sequence once it has processed the entire sequence. The attention mechanism resolves this problem.

2. Describe the main principle difference in the predictions abilities that this layer adds to the network and how it can be seen in the results.

The attention model replicates the human attention mechanism and allows the network to focus on a single word from a sentence.
Self-attention is an attention mechanism relating different positions of a single sequence in order to compute a representation of the same sequence. It can help us to learn a correlation between the current words and the previous part of the sentence.

The sub score of each word is computed by measuring the distance between a query to each key. The network extracts interesting features from a sentence, if the distance between words is small , then they get high probability.

Deep Learning Ex2

1) כלל השרשרת

1c)

$$\frac{\partial}{\partial x} f(x+y, 2x, z) = \frac{\partial}{\partial x} f(u, v, q) = \frac{d}{dx} g(x,y) \cdot D_1 f + \frac{d}{dx} h(x) \cdot D_2 f + \frac{d}{dx} i(z) \cdot D_3 f$$

$$\begin{cases} u = g(x,y) = x+y & D_1 f = \frac{\partial f}{\partial u} \\ v = h(x) = 2x & D_2 f = \frac{\partial f}{\partial v} \\ q = i(z) = z, & D_3 f = \frac{\partial f}{\partial q} \end{cases}$$

$$= 1 \cdot D_1 f + 2 \cdot D_2 f$$

2) כלל השרשרת עבור

$$f_1(f_2(\ldots f_n(x)))$$

נגדיר

$$\begin{cases} f_{1,\ldots,n} = f_1 \circ f_2 \circ f_3 \circ \ldots \circ f_{n-1} \circ f_n \\ f_{i,\ldots,j}(x) = x \\ f_{i,\ldots,i} = f_i \end{cases}$$

כאשר אם $i>j$ אז"י

$$f'_{1,2,\ldots,n}(x) = f_1'(f_{2,\ldots,n}(x)) \cdot f_2'(f_{3,\ldots,n}(x)) \cdot \ldots \cdot f_{n-1}'(f_{n \ldots n}(x)) \cdot f_n'(x)$$

$$= \prod_{k=1}^{n} f_k'(f_{k+1,\ldots,n}(x))$$

לדוגמא עבור $n=3$:

$$f'_{1,2,3}(x) = f_1'(f_{2,3}(x)) \cdot f_2'(f_3(x)) \cdot f_3'(x)$$

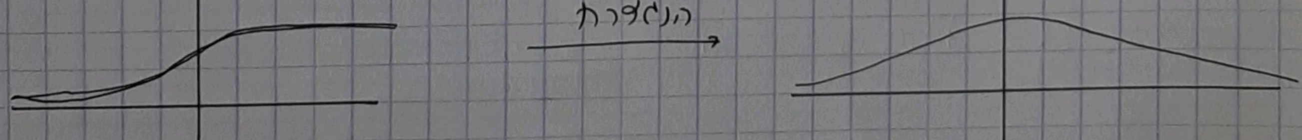
c) מצאו פונקציית אקטיבציה כדי להשתמש במבנה האחרון של G מהרכבה f_i כדי לייצב את הנגזרת שלה.

הפיתרון - שימוש בפונקציית אקטיבציה ReLU

ראשית נתייחס לבעיה של Vanishing Gradient -
האלגוריתם של Gradient Desent מנסה למזער את השגיאה ע"י שימוש בנגזרות ומהן מעדכנים את המשקולות.
מכפלות רבות של נגזרות קטנות יובילו לערכים מאוד קטנים שיהיה קשה לפעפע בשכבות הראשונות ברשת.
ע"פ כלל השרשרת הנגזרות של כל שכבה מוכפלות מהסוף להתחלה, כדי לחשב את הנגזרות של השכבות הראשונות.
מכיוון כך שמספר השכבות ברשת גדול יותר, ייתכן שהגרדיאנט יהיה קטן מאוד כך שהאימון של הרשת לא יהיה אופטימלי כי שכן המשקולות וה-bias של השכבות הראשונות לא יתעדכנו כנדרש - ויהיה קשה למצוא לאיזה כיוון הפרמטרים צריכים לזוז כדי למזער את השגיאה.
מכיוון שהשכבות הראשונות חשובות עבור זיהוי אלמנטים מרכזיים, ואם המשקולות שלהן לא יתעדכנו כנדרש, הביצועים של הרשת יהיו גרועים.

כעת נסביר מדוע פונקציית אקטיבציה של Sigmoid לא תפתור בעיה של Vanishing Gradient.

פונקציית ה-Sigmoid מכווצת את הערכים שמקבלת כך שטווחם בין 0 ל-1. כתוצאה מכך שינויים משמעותיים בקלט של ה-Sigmoid יגרמו לשינויים קטנים בפלט, כך שהנגזרת קטנה.
הגרף של הנגזרת של ה-Sigmoid נראה כך שככל שהערך מגדל ה-X, בערכו המוחלט, גדל יותר, כך הנגזרת מתקרבת ל-0.



כעת נניח שקיימות ברשת n שכבות (hidden layers) שמשתמשות בפונקציית אקטיבציה של sigmoid, ויש משהלך ה-back propagation, n נגזרות קטנות מוכפלות זו בזו, כך שהגרדיאנט קטן אקספוננציאלית ומפעפעים לשכבות הראשונות.

פונקציית ReLU לעומת זאת פשוט מאפסת ערכים שליליים, ולכן הנגזרות לא יהיו מאוד קטנות כמו במקרה של ה-sigmoid.

פונקציית ReLU: $f(X) = max\{0, X\}$ מתנהגת כמו פונקציה ליניארית למרות שהייא לא ליניארית) ומייצגת את הנגזרת של σ f_i.

$f_1(X, f_2(X, f_3(\dots f_{n-1}(X, f_n(X))))$ ג) נוכיח את כלל השרשרת על

* נסמן

$u_1 = X$
$v_1 = f_2(X, f_3(\dots f_{n-1}(X, f_n(X))))$

$$\Longrightarrow \frac{df_1}{dX} = \frac{dX}{dX} \cdot D_1 f_1 + \frac{d}{dX} f_2(X, f_3(\dots f_{n-1}(X, f_n(X))))$$ כפול
$$\cdot D_2 f_1$$

כאשר $D_1 f_1 = \frac{\partial f_1}{\partial u_1}, \quad D_2 f_1 = \frac{\partial f_1}{\partial v_1}$

* נסמן $u_2 = X, \quad V_2 = f_3(\dots f_{n-1}(X, f_n(X)))$

כאשר $D_1 f_2 = \frac{\partial f_2}{\partial u_2}, \quad D_2 f_2 = \frac{\partial f_2}{\partial V_2}$

$$\Longrightarrow \frac{df_2}{dX} = \frac{dX}{dX} \cdot D_1 f_2 + \frac{d}{dX} f_3(\dots f_{n-1}(X, f_n(X)))$$

בסה"כ נקבל $\frac{d}{dX} f_1(X, f_2(X, f_3(\dots f_{n-1}(X, f_n(X)))))$

$$= 1 \cdot \underline{D_1 f_1} + \underline{D_2 f_1} \cdot \Big[1 \cdot \underline{D_1 f_2} + f_3'(\dots f_{n-1}(X, f_n(X))) \cdot \dots \cdot$$
$$f_{n-1}'(X, f_n(X)) \cdot (1 \cdot \underline{D_1 f_{n-1}} + f_n'(X) \cdot \underline{D_2 f_{n-1}}))$$
$$\cdot \underline{D_2 f_2} \Big]$$

Skip Connection : $f(x+g(x+h(x)))$ ה)

נטען כי שימוש ב-Skip Connection יסייע לנו להתמודד עם בעיית ה-Vanishing Gradient.

זאת משום שבאופן זה הגרדיאנט יהיה מורכב מ-1 וערכו ישמר משכבות הראשונות.

כאשר נגזור את $f\circ g\circ h$ נקבל

$$\frac{\partial}{\partial x} f(g(h(x))) = f'(g(h(x)))\cdot g'(h(x))\cdot h'(x)$$

כך שאם הנגזרות קטנות, מכפלה שלהן תהיה מאוד קטנה.

כאשר נגזור את $f(x+g(x+h(x)))$ נקבל

$$\frac{\partial}{\partial x} f(x+g(x+h(x))) = f'(x+g(x+h(x)))\cdot(1+g'(x+h(x)))\cdot(1+h'(x))$$

$$=\left(f'(x+g(x+h(x)))+f'(x+g(x+h(x)))\cdot g'(x+h(x))\right)(1+h'(x))$$

$$= f'(x+g(x+h(x)))+f'(x+g(x+h(x)))\cdot g'(x+h(x))$$
$$+f'(x+g(x+h(x)))\cdot h'(x)$$
$$+f'(x+g(x+h(x)))\cdot g'(x+h(x))\cdot h'(x)$$

העיקרון של הskip connection שמאפשר לדלג על שכבות מסוימות ולהעביר אותן הלאה, יש שכבה אחת בקלט לשכבות העמוקות (ולא רק לשכבה הבאה) אינטואיטיבית, ניתן להעביר על כל שכבה ולהגיע לעצם עם פחות הרכבות פונקציונליות עם כל השכבות עליו להפעיל את הנגזרות החלקיות כפי כמו, הערכים שלהן הם מורכבים קטנים מ-1, ואז הגרדיאנט הולך וקטן מהרכבה של $f_1\circ f_2\circ\dots\circ f_n$.

כאשר משתמשים מפורשות בשיטת ה-back propagation, ע"י ניקוד שימו לב במקרה זה $g(\underline{x}+h(x))$ ו-$f(\underline{x}+\dots)$, הגרדיאנט יורכב מ-1 ולא יקטן כשמפעפעים לשכבות הראשונות.

2) כעת נתאר שיטה ארכיטקטורה שנשתמש בה לפתור את הבעיות הבאות:

א) Speech Recognition - Audio to text

נשתמש ב-GRU many to one שיקבל פיצ'רים מקטע האודיו ויוציא פלט של טקסט. משום שהקלט הינו אודיו נשתמש ב-CNN שתחלץ עבורנו פיצ'רים.
CNN מתאימה עבור קלט של אודיו שהינו מתמטי שכן משתנה כפונקציה של הזמן. במקרה של אותות שיש להן משמעות של זמן ומרחב יש סדר לערכים.

כפי שהסברתי בהרצאה LTI הינו קונבולוציה, ו-CNN הינו translation/time invariant שמשמע אותה פונקציה לאודיו אותו הדבר בכל מקרה, כלומר אותה פונקציה בכל זמן, ובכל מיקום.
אם ניקח דוגמת אודיו ונעשה פרמוטציה כלשהי, נקבל רעש שיהיה קשה לפענח כמו תמונות, משום שלאודיו יש משמעות של זמן ומרחב. ולכן של CNN תיתן מענה לכך.

ניתן להשתמש בPooling כדי לצמצם כל פיקסל שני, ולכן הורדת חלק מהנתונים מהרבה מסקנות שונות. להפחית סיבוכיות אורך האודיו מספיק גדול יותר ע"י שימוש בפיצ'רים קטנים.

מ) Answer Question

במקרה זה הקלט יכול להיות מספר משתנה של מילים, והפלט יכול להיות מספר משתנה של מילים ולכן נשתמש ב RNN many to many

כמו כן, נרצה להיות מסוגלים לתת משקל יותר גדול למילים מסוימות בשאלה שתורמות להן השפעה רבה יותר על התשובה שתינתן. נעשה זאת ע"י הוספה של שכבה של Attention, כשכבה ראשונה.
במשפט יש משמעות למיקום של כל מילה בקלט, וקיימת תלות בין מילה בקלט למילה בפלט במקרה של מענה על שאלות. כמו כן, לא ניתן להגיד שהפלט תלוי במילה אחת בכל רכיבי הקלט)
Attention layer תסייע לנו להחליט לאיזה רכיב מהקלט נתייחס באופן ספציפי עבור רכיב בפלט).

נ) Sentiment Analysis

אם במסגרת החלק המעשי מימשנו מסווגות עם מטרה לסווג איזה משפט יש לחוות עבור הדיוק הטוב ביותר הוא מסווג מסווגות חיוביות או שליליות.
מסווגים עם מודל של RNN, של GRU, מספר שכבות FC, מספר שכבות FC עבור sub score כאשר ה-GRU נתן דיוק גבוה יותר לעומת RNN, באופן משמעותי.
והוספת Attention layer הגיעה רק לתוצאות של GRU.

כפי שהוספנו במקרה של Answer questions, שכבת הAttention מאפשרת מתן משקל גבוה יותר למילים משמעותיות, ולכן, רלוונטי לתת מענה למשימה.

GRU מאפשר לבצע פרדיקציה מתוך ההיסטוריה, כאשר לא ברור כמה אחורה נדרש להסתכל. הרשת מקבלת את כל ההיסטוריה ומחליטה לאיזה חלק היא תתן משקל גבוה יותר.

במקרה זה מדובר ב- many to one שכן אנו מקבלים אוסף של מילים ומוציאים פלט האם חיובי או שלילי.
הערה: גם במסווגים הנוספים ניתן להחליף RNN ב-GRU שיפתור בעיית ה Vanishing Gradient
הערה: אנו מוסיפים שכבות FC בהמשך לפני מתן הפרדיקציה.

ס) Image Classification

נשתמש ב-CNN על מנת לבצע קלסיפיקציה לתמונות.
כפי שהתייחסנו לסיגנל של אודיו, גם לתמונה יש משמעות של זמן ומרחב.
הסדר של הפיקסלים בתמונה יוצר אובייקטים מסוימים שעל פיהם מתבצעת קלסיפיקציה, ככך שפרמוטציה כללית של הפיקסלים תייצר רעש, שיהיה קשה לפענח מן התמונות. משום שרשת CNN של הקונבולוציה הינה translation Invariant היא תתפוס על איזורים שונים מהתמונה בצורה אחידה

Trans. Inv משמר את הצורות של האובייקט, או הקטגוריה שלו, גם אם יש שינויים וסיבובים כמו המיקום של האובייקט בתמונה או הזווית ביחס לאובייקט)
כמו כן, הגישה כל פיקסל שני מאפשר זיהוי בסקאלות שונות.

ע) Single Word Translation

הערה: עבור קלט אקראי (גם מסווגים ע"י ו-ג') נשתמש ב-embedding Layers ייצוגים וקטורים על מנת לתרגם מילה כאשר לנקח קלט של מילה בשפה אחת ולהוציא פלט של התרגום של המילה לשפה שנייה, נשתמש ב-RNN one-to-one
נכניס את הקלט ל- encoder ואת הפלט של ה- encoder נכניס ל- decoder שיוציא את התרגום. ה- encoder יהיה Recurrent Layers, ה- decoder יהיה שכבות fully connected שימפו את ה encoded input לתרגום המתאים
לשכבות RNN יש אפשרות לזכור את ההיסטוריה, ובמקרה של קלט אקראי המילים הקודמות שמשמעותיות כדי להבין את הקונטקסט של מילה קיימת
הערה: לכן ניתן להשתמש ב- LSTM או בגרסה המתקדמת GRU במקום RNN
כדי לנטרל משמעות של Vanishing Gradient

3. Text-to-image:

a. Describe the architecture of a network that reads a sentence and generates an image based on this text. Do not address the question of how such a network should be trained, just explain why it should have the capacity to perform this task. You can assume that the images come from a restricted class of images e.g. faces and can be encoded (and decoded) in a low-dimensional latent space.

בהרצאה דיברנו על image captioning שבו הקלט הוא תמונה והפלט הוא משפט המתאר את התמונה, שזו למעשה הבעיה ההפוכה לבעיית text-to-image. ב-image captioning המטרה הסופית הייתה לייצר משפט ולמדנו שאפשר להשתמש במודל של LSTM שמחשב בהיסטוריה ומטפל בבעיה של vanising Gradient.

כמו כן למדנו כי ניתן להפוך את התמונה לוקטור קבוע כך שמדובר במקרה של one-to-many והדגשנו כי על מנת לתאר תמונה נתייחס ברוב המקרים לאובייקט שמבצע פעולה על אובייקט אחר, ולדוגמא היוצרה מורכבת פריטים. כלומר המילים במשפט לא מתייחסות לכל התמונה אלא לחלקים ספציפיים בה.

לאחר שהפכנו את התמונה לוקטור נחלץ פיצ'רים מהמפה ונעבדם ע"י שימוש ב attention layer דרכנו Hidden Vector שיידע מה נמצא איפה, ויכול לבצע פרמוטציה לסדר כי ומנות משפט. עשינו זאת ע"י חילוץ של feature vector מהתמונה שהוא שונה דיסקריפטור על אזורים בתמונה. באופן זה יכולנו למצוא את המילה המתאימה ביותר לאובייקט שמשויך לאיזור שחילצנו מהתמונה.

בשורה התחתונה ה-attention layer מסייע לנו לחלץ לאיזה רכיב בפלט כלומר לאיזה פיצ'רים מהתמונה להתייחס באופן ספציפי עבור רכיב הפלט - המשפט.

בבעיה במקרה שלנו (text to image) אנחנו רוצים להיות מסוגלים לזהות פיצ'רים משמעותיים ברכיב הקלט שבמקרה זה הוא משפט עבור רכיב הפלט שבמקרה הוא תמונה. בדומה למה שעשינו ב-image captioning נחלץ פיצ'רים מהמשפט באמצעות text embedding

נרצה לייצג תכונות ויזואליות מתוך הטקסט, כלומר נחלץ מהמשפט מילים שיאפשרו לנו מעבר להבנה של הקונטקסט גם יכולת לזהות מאפיינים ויזואלים. חילוץ הפיצ'רים המשמעותיים מתבטא גם בהורדת המימד. ה encoder ימפה תמונות ומשפטים כך שתמונה ותיאור שהם מתאימים "יוצגו ע"י וקטורים שהמכפלה הפנימית שלהם גדולה.

המשך

שהי: ה- Capacity על מנת לייצר תמונה מטקסט אנחנו צריכים והיא מסוגלת
להתמקד מהיסטוריה של מילה בשונה לא מהמחברת תיאור המבר משמעותי
לתמונה של המשפט כולו שכן תלויות בין המילים יוצרות את המשמעות של המשפט.
כמו כן אם המשפט ארוך ומתייחס למספר אובייקטים יש צורך לזכור את ההיסטוריה.

בנוסף, לבעיה זו יש אלמנט גנרטיבי - אדברסריאלי שכן עלינו לייצר תמונה ולכן
נשתמש ב GAN. יהיו לנו שתי רשתות אשר מתחרות אחת בשניה כך שהדיסקרימינטור
מאמן את הגנרטור שמנסה להטעות אותו. גנרטור יוצר מרעש תמונה שתהיה תואמת
למשפט שהתקבל, כאשר הוא ינסה להטעות את הדיסקרימינטור בשני אופנים - ינסה
לגרום לדיסקרימינטור לחשוב שהתמונה אמיתית וינסה לגרום לו לחשוב שהתמונה
מתאימה למשפט שהתקבל כקלט. אם הדיסקרימינטור חושב שהתמונה אמיתית אז נשלם
הקנס (הוא בודק התאמה בין משפט לתמונה ע"י חישוב ה- Similarity ומחזיר ציון
גבוה) כאשר יש התאמה.

b. assume that the latent codes are made of 4 vectors that correspond to four quadrants of an image. Explain how an attention layer can be used to allow such a network to better support regional descriptions in the input text. what would be the queries, keys and values?

באמצעות attention layer נוכל לבצע דיסקריפטורים עבור מילים משמעותיות מתוך המשפט שלפיהן
נרכיב את התמונה. כאשר שכבת attention נוכל לבחור באופן אוטומטי תנאים מרשת
המילים כדי לייצר חלקים שונים בתמונה, כאשר נרצה לבחור את המילים הרלוונטיות
ביותר עבורנו כדי לייצר את התמונה.

הקלט שלנו משפט מועבר ל LSTM שמייצר feature vectors אשר מתאימים
לדיסקריפטורים ששייכים לאיזורים שונים במשפט. אותם feature vectors נכנסים
לתוך ה Attentation model שיבנה תמונה על סמך המילים שבחר.

באמצעות Attentation layers נוכל להסיק באיזה סדר להתייחס לאיזה חלק במשפט
הכניסה האופטימלית. את ה- query נייצר מה hidden vector ולפי נוכל לבחור באיזה
קונטקסט אנחנו. הפונקציה שבוחרת לאיזה פיצ'ר יש להתייחס נעשית על פי הקונטקסט
והתמונה ובונה בהינתן הפיצ'רים המשמעותיים.

המרחק שנמדוד בין ה- key ל- query הוא מדד לקירבה ומחושב ע"י מכפלה
פנימית.
נשתמש ב Softmax ליצירת מפת הסתברויות על סמך המרחקים כך ששום המרחק קטן
אז ההסתברות שיבחר גבוהה וה- value יהיה משמעותי בקומבינציה הליניארית. במקרה
כזה נקבל ונבדוק שהמשפט והתמונה קרובים ולאחר מכן נחזיר את התמונה המתאימה.