CSCI 5922 - 001 Neural Networks and Deep Learning
Problem Set 4

1. **Transformers vs RNNs vs MLPs**

   a) Transformers and RNNs are similar because,
   - Both Transformers and RNNs handle the sequence to sequence data related problem statements such as language translation, speech recognition and text processing.
   - Both these models can be trained using a very large text datasets/corpus with a variety of word embeddings feeded into the model.
   - Both models understand the relationships between different words by passing the states through timestamps.

   b) Transformers and RNNs are different because,
   - RNNs struggle with the Long short term memory handling issue and in RNNs we cannot pass the input simultaneously but it is possible with Transformers with the help of self attention.
   - Transformers get the whole sentence at once but the RNNs will depend on the past words and sometimes it will miss important context required for predicting the next word.
   - RNNs handle different sequence lengths whereas the transformers use the positional encoding.

   c) Transformers are similar to MLPs because,
   - Both are fully connected architectures which contains weight vectors, activation functions and layers of neurons for learning the intricate patterns about the data.
   - Both of these models heavily rely on the mathematical computation of matrices so both of them are GPU intensive tasks.
   - Both of these models can be used for the classification task for example Transformers can be used in Classification of the text “Spam Detection” and MLP’s also classify ie.“Credit Card Fraud Detection”

   d) Transformers are different from MLPs, because
   - Transformers work with sequence to sequence text data (unstructured data) while the MLPs work well with the structured numerical data.
   - Transformers learns the contextual based relationships of generating a word / sequence of words but MLPs just maps the inputs to the outputs and they won’t even bother about the context.

- Transformers focus on the highly important words by means of self attention but in MLPs there is no ability like that.

2. **Self attention Equation**

a)

i. When computing the QK^T the values will become very large if the d_k ie. the dimension of the key vectors is high. This may cause the softmax function to produce very small gradient values, which makes the learning process very slow and leads to vanishing gradient problem. Dividing by $\sqrt{d_k}$ will scale down the values and keeps them in a stable range. This prevents the softmax function from producing overfitted probability values and it will also help in improving the training stability.

ii. The softmax activation function will convert the computed attention scores into probabilities ie. from the range of 0 to 1. The output of the softmax function produces a vector which sums to 1. This activation function will solve the problem of which relevant tokens should be considered more priority ie. helps in deciding the most relevant tokens. It will also help in a level of differentiation in which it amplifies the larger scores by suppressing the smaller scores which helps the self attention mechanism easier to capture the correlation between the input sequence.

b) The term equation (1) is referred to as the attention map/attention matrix because

- It will show how much attention each word in a sentence should give to other words in a sentence.
- It is called the attention map because it marks the important words and reduces the attention on the less relevant words.
- Each row represents a Query (word) and the column represents Key (Word) which forms the Attention Scores matrix. ie. The matrix multiplication of Query and keys forms attention matrix.
- This will also help in the understanding of which words should get importance in the given context.

c) For example in an example of Machine Translation of English to Hindi, if we use single headed attention the model will focus only one aspect of word at a time and will miss the important contexts.

For example: I love apple. Here ‘apple’ is represented as a fruit / company?
The single attention head will struggle to capture both of the meanings.

But the multi headed attentions will capture multiple meanings by means of different attention heads. And some heads will focus on the nearby words and other heads will capture the long range dependency words to help in improving the understanding.

Multiple heads will provide the wide range of perspectives, which prevents biasedness of relying on a single feature. It will also process the input simultaneously at the same time making the training process more efficient.

d) Transformer encoder will read the whole i/p sequence at once and it will understand how the different parts of the sentence relate to each other. It has layers that will help in learning the meaning and the context of the data.
The layers that it consists are self attention layers and feed forward neural networks which helps in learning the patterns of input data.
Example: Bidirectional Encoder Representations from Transformers [BERT] is used for the sentimental analysis of Food Reviews in Uber Eats. It helps the organisation to make investment and marketing decisions. It will process the entire text bidirectionally, it is well suitable for the understanding of context rich customer reviews.
Transformer decoder on the other end works step by step. It will predict based on the past word and the context from the encoder. ie. It will generate each and every word based on the previous words and the context is from the encoder.
The transformer decoder has self attention layer, encoder decoder attention layer and feed forward neural networks layers to understand both the previous knowledge as well as past knowledge from encoder.
Example: In Healthcare BioGPT which is created to write the medical reports. The decoder ensures the text flows such as summarizing the patient records and the test results which makes easier for the doctors to make decisions.

The encoders are best for analyzing the data and the decoders are best for generating new data.

Neural Networks & Deep Learning

problem set 4

3) i/p tokens:

1	-1	0	2
0	-1	2	1
-2	2	-1	0

Query weights W_Q

0	1
1	0
-0.5	0
1	1

key weights W_K

1	-0.5
0	1
0.5	-0.5
1	0.5

value weights W_V

1	0.5
-0.5	0
-1	2
0.5	-1

For i/p token 1 : $[1\ -1\ 0\ 2]$

Query Q_1 : $i/p1 \times W_Q = \underset{1\times4}{[1\ -1\ 0\ 2]} \underset{4\times2}{\begin{bmatrix} 0 & 1 \\ 1 & 0 \\ -0.5 & 0 \\ 1 & 1 \end{bmatrix}}$

$= [0-1+0+2 \quad 1+0+0+2]$

$Q_1 = [1 \quad 3]$

key K_1 : $i/p1 \times W_K = [1\ -1\ 0\ 2] \begin{bmatrix} 1 & -0.5 \\ 0 & 1 \\ 0.5 & -0.5 \\ 1 & 0.5 \end{bmatrix}$

$= [1+0+0+2 \quad -0.5-1+0+1]$

$K_1 = [3 \quad -0.5]$

Value V_1 : $i/p1 \times W_V = [1\ -1\ 0\ 2] \begin{bmatrix} 1 & 0.5 \\ -0.5 & 0 \\ -1 & 2 \\ 0.5 & -1 \end{bmatrix}$

$= [1+0.5+0+1 \quad \tfrac{1}{2}+0+0-2]$

$V_1 = [2.5 \quad -1.5]$

for i/p token 2 : $[0\ -1\ 2\ 1]$

Query Q_2 : $i/p1 \times W_Q = [0\ -1\ 2\ 1] \begin{bmatrix} 0 & 1 \\ 1 & 0 \\ -0.5 & 0 \\ 1 & 1 \end{bmatrix}$

$= [0-1-1+1 \quad 0+0+0+1]$

$Q_2 = [-1 \quad 1]$

key K_2 : $i/p2 \times W_K = [0\ -1\ 2\ 1] \begin{bmatrix} 1 & -0.5 \\ 0 & 1 \\ 0.5 & -0.5 \\ 1 & 0.5 \end{bmatrix}$

$= [0+0+1+1 \quad 0-1-1+\tfrac{1}{2}]$

$K_2 = [2 \quad -1.5]$

Value V_2 : $i/p2 \times W_V = [0\ -1\ 2\ 1] \begin{bmatrix} 1 & 0.5 \\ -0.5 & 0 \\ -1 & 2 \\ 0.5 & -1 \end{bmatrix}$

$= [0+\tfrac{1}{2}-2+\tfrac{1}{2} \quad 0+0+4-1]$

$V_2 = [-1 \quad 3]$

for i/p token 3 : $[-2\ 2\ -1\ 0]$

Query Q_3 : $i/p3 \times W_Q = [-2\ 2\ -1\ 0] \begin{bmatrix} 0 & 1 \\ 1 & 0 \\ -0.5 & 0 \\ 1 & 1 \end{bmatrix}$

$= [0+2+\tfrac{1}{2}+0 \quad -2+0+0+0]$

$Q_3 = [2.5 \quad -2]$

key K_3 : $i/p3 \times W_K = [-2\ 2\ -1\ 0] \begin{bmatrix} 1 & -0.5 \\ 0 & 1 \\ 0.5 & -0.5 \\ 1 & 0.5 \end{bmatrix}$

$= [-2+0-0.5+0 \quad 1+2+0.5+0]$

$K_3 = [-2.5 \quad 3.5]$

Value V_3 : $i/p3 \times W_V = [-2\ 2\ -1\ 0] \begin{bmatrix} 1 & 0.5 \\ -0.5 & 0 \\ -1 & 2 \\ 0.5 & -1 \end{bmatrix}$

$= [-2-1+1+0 \quad -1+0-2+0]$

$V_3 = [-2 \quad -3]$

Computing the attention scores

I/p token 1

we have to multiply the Query with all the keys.

$Q_1 \times K_1 = [1\ 3] \times [3\ -0.5]^T = 3 - 1.5 = 1.5$

$Q_1 \times K_2 = [1\ 3] \times [2\ -1.5]^T = 2 - 4.5 = -2.5$

$Q_1 \times K_3 = [1\ 3] \times [-2.5\ 3.5]^T = -2.5 + 10.5 = 8$

I/p token 2

$Q_2 \times K_1 = [-1\ 1] \times [3\ -0.5]^T = -3 - 0.5 = -3.5$

$Q_2 \times K_2 = [-1\ 1] \times [2\ -1.5]^T = -2 - 1.5 = -3.5$

$Q_2 \times K_3 = [-1\ 1] \times [-2.5\ 3.5]^T = 2.5 + 3.5 = 6$

I/p token 3:

$Q_3 \times K_1 = [2.5\ -2] \times [3\ -0.5]^T = 7.5 + 1 = 8.5$

$Q_3 \times K_2 = [2.5\ -2] \times [2\ -1.5]^T = 5 + 3 = 8$

$Q_3 \times K_3 = [2.5\ -2] \times [-2.5\ 3.5]^T = 6.25 - 7 = -13.25$

computing the attention weights & attention Maps

I/p token 1:

Attention weights $= softmax([1.5, -2.5, 8])$ → max is '8'

$$softmax(z_i) = \frac{e^{z_i - max(z)}}{\sum_{j=1}^{n} e^{z_j - max(z)}}$$

numer: $e^{z_i - max(z)} = [e^{1.5-8}, e^{-2.5-8}, e^{8-8}] = [e^{-6.5}, e^{-10.5}, e^{0}] = [0, 0, 1]$

denom: $\sum_{j=1}^{n} e^{z_j - max(z)} = e^{1.5-8} + e^{-2.5-8} + e^{8-8} = e^{-6.5} + e^{-10.5} + e^{0} = 1$

here $e^{-6.5} \approx 0$, $e^{-10.5} \approx 0$, $e^{0} = 1$

so, $softmax([1.5, -2.5, 8]) = (0/1, 0/1, 1/1)$

attention weights $= (0, 0, 1)$ → (aw_1, aw_2, aw_3)

Attention maps ⇒ attention weights × values

$aw_1 \times V_1 = 0 \times [2.5, -1.5] = [0\ 0]$

$aw_2 \times V_2 = 0 \times [-1\ 3] = [0\ 0]$

$aw_3 \times V_3 = 1 \times [-2\ -3] = [-2\ -3]$

I/p token 2:

Attention weights $= softmax([-3.5, -3.5, 6])$ → max is '6'

$$softmax(z_i) = \frac{e^{z_i - max(z)}}{\sum_{j=1}^{n} e^{z_j - max(z)}}$$

numer: $e^{z_i - max(z)} = [e^{-3.5-6}, e^{-3.5-6}, e^{6-6}] = [e^{-9.5}, e^{-9.5}, e^{0}] = [0, 0, 1]$

denom: $\sum_{j=1}^{n} e^{z_j - max(z)} = e^{-9.5} + e^{-9.5} + e^{0} = 1$

here $e^{-9.5} \approx 0$, $e^{0} = 1$

so, $softmax([-3.5, -3.5, 6]) = (0/1, 0/1, 1/1)$

attention weights $= (0, 0, 1)$ → (aw_1, aw_2, aw_3)

Attention maps ⇒ attention weights × values

$aw_1 \times V_1 = 0 \times [2.5, -1.5] = [0\ 0]$

$aw_2 \times V_2 = 0 \times [-1\ 3] = [0\ 0]$

$aw_3 \times V_3 = 1 \times [-2\ -3] = [-2\ -3]$

I/p token 3:

Attention weights $= softmax([8.5, 8, -13.25])$ → max is '8.5'

$$softmax(z_i) = \frac{e^{z_i - max(z)}}{\sum_{j=1}^{n} e^{z_j - max(z)}}$$

numer: $e^{z_i - max(z)} = [e^{8.5-8.5}, e^{8-8.5}, e^{-13.25-8.5}] = [e^{0}, e^{-0.5}, e^{-21.75}] = [1, 0.6, 0]$

denom: $\sum_{j=1}^{n} e^{z_j - max(z)} = e^{8.5-8.5} + e^{8-8.5} + e^{-13.25-8.5} = e^{0} + e^{-0.5} + e^{-21.75} = 1 + 0.6 = 1.6$

here $e^{0} = 1$, $e^{-0.5} = 0.6$, $e^{0} = 1$

so, $softmax([8.5, 8, -13.25]) = \left(\frac{1}{1.6}, \frac{0.6}{1.6}, \frac{0}{1.6}\right) = (0.62, 0.37, 0)$

attention weights $= (0.62, 0.37, 0)$ → (aw_1, aw_2, aw_3)

Attention maps ⇒ attention weights × values

$aw_1 \times V_1 = 0.62 \times [2.5, -1.5] = [1.55\ -0.93]$

$aw_2 \times V_2 = 0.37 \times [-1\ 3] = [-0.37\ 1.11]$

$aw_3 \times V_3 = 0 \times [-2\ -3] = [0\ 0]$

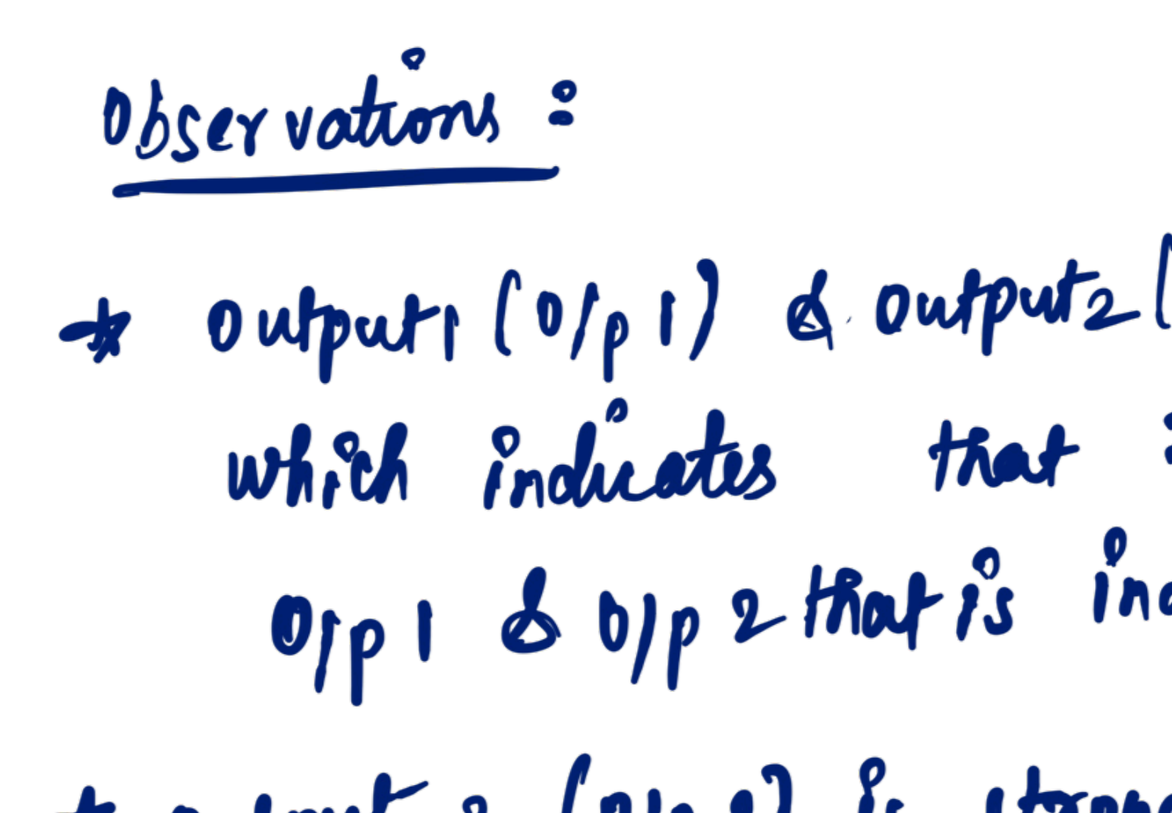
b) By using the attention maps got from (a)
The o/p tokens are :

for i/p token 1 : $[0\ 0] + [0\ 0] + [-2\ -3] = [-2\ -3]$

for i/p token 2 : $[0\ 0] + [0\ 0] + [-2\ -3] = [-2\ -3]$

for i/p token 3 : $[1.55\ -0.93] + [-0.37\ 1.11] + [0\ 0] = [1.18\ 0.18]$

c) Vizualization of attention map from (a)



d) **Observations :**

* output1 (o/p 1) & output2 (o/p 2) are strongly influenced by the Input3 (i/p 3) which indicates that Input 3 has the highest attention weight for both o/p 1 & o/p 2 that is indicated by the solid line

* output 3 (o/p 3) is strongly influenced by the Input1 (i/p 1) which indicates that i/p 1 has the highest attention weight for o/p 3 that is indicated by the solid line

* Output 2 (o/p 2) is slightly influenced by the Input 2 (i/p 2) which indicates that i/p 2 has the 2nd highest attention weight in the attention weight vector that is indicated by the dashed line.

4)

a) Computing the cross attention map

Query

$$Q = X_1 W_Q$$

$$= \begin{bmatrix} -1 & 1 & 0 \\ 0 & 1 & 0 \end{bmatrix}_{2\times3} \begin{bmatrix} 0 & -1 \\ 1 & 1 \\ -1 & 0 \end{bmatrix}_{3\times2}$$

$$= \begin{bmatrix} 0+1+0 & 1+1+0 \\ 0+1+0 & 0+1+0 \end{bmatrix}$$

$$Q = \begin{bmatrix} 1 & 2 \\ 1 & 1 \end{bmatrix}$$

Key

$$K = X_2 W_K$$

$$= \begin{bmatrix} -1 & 1 \\ 0 & 1 \\ 1 & 1 \\ 2 & 0 \end{bmatrix}_{4\times2} \begin{bmatrix} 1 & 0 \\ -1 & 1 \end{bmatrix}_{2\times2}$$

$$K = \begin{bmatrix} -2 & 1 \\ -1 & 1 \\ 0 & 1 \\ 2 & 0 \end{bmatrix}$$

Value

$$V = X_2 W_V$$

$$= \begin{bmatrix} -1 & 1 \\ 0 & 1 \\ 1 & 1 \\ 2 & 0 \end{bmatrix} \begin{bmatrix} -1 & 0 & 1 & 2 \\ 1 & -1 & 0 & -2 \end{bmatrix}$$

$$V = \begin{bmatrix} 2 & -1 & -1 & -4 \\ 1 & -1 & 0 & -2 \\ 0 & -1 & 1 & 0 \\ -2 & 0 & 2 & 4 \end{bmatrix}$$

Cross attention $(Q, K, V) = \text{Softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$

$$QK^T = \begin{bmatrix} 1 & 2 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} -2 & -1 & 0 & 2 \\ 1 & 1 & 1 & 0 \end{bmatrix}$$

$$QK^T = \begin{bmatrix} 0 & 1 & 2 & 2 \\ -1 & 0 & 1 & 2 \end{bmatrix}$$

Softmax

Softmax for row 1 of QK^T

Softmax $([0\ 1\ 2\ 2])$ → max is 2

$$\text{Softmax}(z_i) = \frac{e^{z_i - \max(z)}}{\sum_{j=1}^{n} e^{z_j - \max(z)}}$$

numer: $e^{z_i - \max(z)} = e^{0-2}, e^{1-2}, e^{2-2}, e^{2-2}$

$= e^{-2}, e^{-1}, e^{0}, e^{0}$

$= \left(\frac{1}{e^2}, \frac{1}{e}, 1, 1\right) = \left(\frac{1}{7.39}, \frac{1}{2.71}, 1, 1\right)$

denom: $\sum_{j=1}^{n} e^{z_j - \max(z)} = \frac{1}{e^2} + \frac{1}{e} + 1 + 1 = \frac{1}{7.39} + \frac{1}{2.71} + 1 + 1 = 2.50$

here $e^0 = 1, e^1 = 2.71, e^2 = 7.39$

so, Softmax $([0\ 1\ 2\ 2]) = \left[\frac{1}{7.39 \times 2.5}, \frac{1}{2.71 \times 2.5}, \frac{1}{2.5}, \frac{1}{2.5}\right]$

$= [0.05, 0.15, 0.40, 0.40]$

Softmax for row 2 of QK^T

Softmax $(-1\ 0\ 1\ 2)$ → max is 2

numer: $e^{z_i - \max(z)} = (e^{-1-2}, e^{0-2}, e^{1-2}, e^{2-2})$

$= (e^{-3}, e^{-2}, e^{-1}, e^{0}) \left(\frac{1}{19.90}, \frac{1}{7.39}, \frac{1}{2.71}, 1\right)$

denominator: $e^{-3} + e^{-2} + e^{-1} + e^{0} = \frac{1}{19.90} + \frac{1}{7.39} + \frac{1}{2.71} + 1 = 1.55$

Softmax $(-1\ 0\ 1\ 2) = \left[\frac{1}{19.90 \times 1.55}, \frac{1}{7.39 \times 1.55}, \frac{1}{2.71 \times 1.55}, \frac{1}{1.55}\right]$

$[0.03, 0.08, 0.24, 0.64]$

$$\text{Softmax}(QK^T) = \begin{bmatrix} 0.05 & 0.15 & 0.40 & 0.40 \\ 0.03 & 0.08 & 0.24 & 0.64 \end{bmatrix}$$

b) Output

o/p 1

$= [0.05 \quad 0.15 \quad 0.40 \quad 0.40] \times V$

$= [0.05 \quad 0.15 \quad 0.40 \quad 0.40]_{1\times4} \times \begin{bmatrix} 2 & -1 & -1 & -4 \\ 1 & -1 & 0 & -2 \\ 0 & -1 & 1 & 0 \\ -2 & 0 & 2 & 4 \end{bmatrix}_{4\times4}$

$= [-0.55, -0.6, 1.15, 1.1]$

o/p 2

$= [0.03 \quad 0.08 \quad 0.24 \quad 0.64] \begin{bmatrix} 2 & -1 & -1 & -4 \\ 1 & -1 & 0 & -2 \\ 0 & -1 & 1 & 0 \\ -2 & 0 & 2 & 4 \end{bmatrix}$

$= [-1.14 \quad -0.35 \quad 1.49 \quad 2.28]$

Cross attention map $= \begin{bmatrix} -0.55 & -0.6 & 1.15 & 1.1 \\ -1.14 & -0.35 & 1.49 & 2.28 \end{bmatrix}$

from the cross attention map

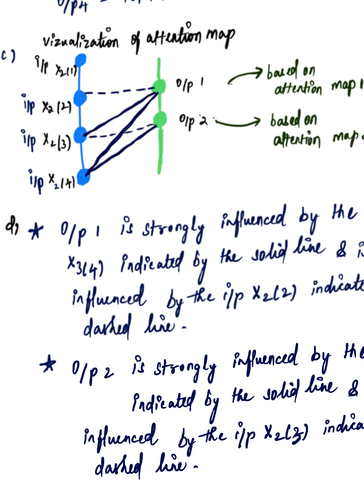
o/p 1 $= -0.55 - 1.14 = -1.69$

o/p 2 $= -0.6 - 0.35 = -0.95$

o/p 3 $= 1.15 + 1.49 = 2.64$

o/p 4 $= 1.1 + 2.28 = 3.38$

c) Visualization of attention map



d) ★ o/p 1 is strongly influenced by the i/p $X_{2(3)}$ & $X_{3(4)}$ indicated by the solid line & is slightly influenced by the i/p $X_{2(2)}$ indicated by the dashed line.

★ o/p 2 is strongly influenced by the i/p $X_{2(4)}$ indicated by the solid line & is slightly influenced by the i/p $X_{2(3)}$ indicated by the dashed line.