National University
of Computer & Emerging Sciences

Serial No: 202151

Please Tick (✓) Campus: CFD □ ISB ☑ KHI □ LHR □ PWR □ Semester: SP □ SU □ FA ☑ 2024

FINAL EXAM ANSWER BOOK

Course Code & Title: Generative AI

Roll No: 21i-2990 Section: A Student's Signature: Date: 3rd Jan 2025

Serial No. of continuation sheet if attached: ____ Total No. of Extra Sheets Used: ____ Invigilator's Signature: ____

(THIS ANSWER BOOK CONTAINS PAGE 1-16)

DO NOT OPEN THE ANSWER BOOKLET OR START UNTIL INSTRUCTED

Instructions:

1. Please ensure that the area in your threshold is free of any material classified as 'useful in the paper' or else there may a charge of cheating.
2. Read the question carefully to ensure clarity of context and understanding of meaning. Make assumptions whatever necessary, as neither the invigilator nor the teacher/examinar will address your queries or provide assistance in the examination hall.
3. Fit in all your answers in the answer booklet. You may use an extra sheet if required. If you do so, clearly mark question/part number on that page to avoid confusion.
4. Use only your own stationery and calculator. (If permitted by your teacher/examiner). If you do not have your own calculator, perform manual calculations.
5. Use only permanent ink pens. Only the questions attempted with permanent ink pen will be considered. Any part of paper done in lead pencil cannot be claimed for rechecking.
6. Ensure that you do not have any electronic gadget (like mobile phone, smart watch, ear buds etc.) with you.
7. Return your Question Paper along with the answer booklet (including extra sheets, if used) to the invigilator before leaving the exam venue.

Q./Part No.	Q-1	Q-2	Q-3	Q-4	Q-5	Q-6	Q-7	Q-8	Q-9	Q-10
Total Marks	50	30	5	5	10					
Obtained Marks	44	30	5	5	10					
CLO NO.										

Total Marks
100
94
Total Marks Obtained

Examiner/Course Teacher

Date

Part II

Q2 - Short Questions:

Q1. Context provides relevant information and patterns of data to help the model generate better outputs. It can include factual data or be examples of things like problems, solutions, question and answer formats, and chains of reasoning. This is able to help the model better understand how to address the user's prompt.

3

Q2. $$\text{Softmax}^{(z_i)} = \frac{e^{z_i}/T}{\sum_{j=0}^{2} e^{z_j}/T}$$

Probability for cat = $\text{Softmax}(3) = \frac{e^{3}/4}{(e^{3}/4)+(e^{2}/4)}$

$= 0.731$

3 Probability for dog = $\text{Softmax}(2) = \frac{e^{2}/4}{(e^{2}/4)+(e^{3}/4)} = 0.269$

Q3. For creative content, especially like poetry, we would increase the temperature. Temperature is used to increase probabilities of less likely outputs for the model. Thus, when we raise it, it allows the model to be more likely to use a wider variety of words, ideal for poetry and other creative writing tasks. However, we should ensure the temperature is not raised too high as it might cause model to not follow the theme and maybe even create non-sensical output.

3

Q4. number of layers = 6

number of attention heads per layer = 8

embedding size = 768

Total heads = $6 \times 8 = 48$

Thus, there are 48 attention heads.

$\frac{768}{8} = 96$ $\frac{768}{16} = 48$

3

Q5. I would choose sentence-Bert with a Bi-encoder siamese network. Each Bert encoder in the Bi-encoder can generate a single (classification) token to help categorize incoming queries into different classes. Furthermore, the Sentence-BERT network as a whole can compare our new query with queries in the database such as with a metric like cosine similarity. Sentence level tokens make for faster, easier comparison with predefined encodings.

3

Q6. Model quantization is a technique where we reduce the precision of ~~[illegible]~~ numbers within models to lower computation costs. It is often used in knowledge distillation, wherein we can quantize a larger teacher network into a student network with lower computational requirements.

3

Q7. The BERT model uses bi-directional context where tokens are able to see each other and themselves as opposed to a unidirectional context. Oftentimes, this is done by taking both left-to-right and right-to-left context, helping BERT gain a better understanding of the input to make semantic-capturing embeddings.

3

Q8. Autoregressive models are able to predict the next word in the sequence by looking at the previous words and context. If the models needs to generate multiple words, it will use past words to generate the next most likely word. Then this word is appended to the sequence, and the model repeats the word generation process.

Q9. I would prefer using RAG. Fine-tuning is a costly process and given the product catalog updates frequently, the fine-tuning costs would add up. Thus, RAG would be a better alternative as we would only need to update the vector database and provide that to our model. Thus, RAG is better as it is less costly and provides sufficiently decent results.

Q10. $A = [3, 4]$ $B = [4, 3]$

$$\cos(\theta) = \frac{A \cdot B}{\|A\| \|B\|} = \frac{24}{5 \times 5} = \frac{24}{25} = 0.96 \leftarrow \text{Ans}$$

$$A \cdot B = \begin{bmatrix} 3 \\ 4 \end{bmatrix} \cdot \begin{bmatrix} 4 \\ 3 \end{bmatrix} = 3 \times 4 + 3 \times 4 = 12 + 12 = 24$$

$$\|A\| = \sqrt{3^2 + 4^2} = \sqrt{9 + 16} = \sqrt{25} = 5$$

$$\|B\| = \sqrt{4^2 + 3^2} = \sqrt{16 + 9} = \sqrt{25} = 5$$

Long Questions:

Q5- Q1. x_t for forward diffusion:

$$x_0 = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \quad \epsilon = \begin{bmatrix} 0.1 & 0.2 \\ 0.3 & 0.4 \end{bmatrix}$$

$\alpha_t = 0.5$

$$x_t = \sqrt{a_t}\, x_0 + \sqrt{1-a_t}\, \epsilon$$

$$= \sqrt{0.5} \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} + \sqrt{1-0.5} \begin{bmatrix} 0.1 & 0.2 \\ 0.3 & 0.4 \end{bmatrix}$$

$$= \begin{bmatrix} \frac{\sqrt{2}}{2} & \sqrt{2} \\ \frac{3\sqrt{2}}{2} & 2\sqrt{2} \end{bmatrix} + \begin{bmatrix} \frac{\sqrt{2}}{20} & \frac{\sqrt{2}}{10} \\ \frac{3\sqrt{2}}{20} & \frac{\sqrt{2}}{5} \end{bmatrix}$$

$$= \begin{bmatrix} \frac{11\sqrt{2}}{20} & \frac{11\sqrt{2}}{10} \\ \frac{33\sqrt{2}}{20} & \frac{11\sqrt{2}}{5} \end{bmatrix}$$

$$x_t = \begin{bmatrix} 0.7778 & 1.5556 \\ 2.3335 & 3.1112 \end{bmatrix}$$

Q2. $L(\theta)$ for reverse diffusion:

$$L(\theta) = E_{x_0,t,\epsilon} [\|\epsilon - \epsilon_\theta(x_t, t)\|^2]$$

$$\downarrow$$

$$\epsilon = \begin{bmatrix} 0.1 & 0.2 \\ 0.3 & 0.4 \end{bmatrix}$$

5

$$\epsilon_\theta(x_t, t) = \begin{bmatrix} 0.0 & 0.1 \\ 0.3 & 0.5 \end{bmatrix}$$

$$\left\| \begin{bmatrix} 0.1 & 0.2 \\ 0.3 & 0.4 \end{bmatrix} - \begin{bmatrix} 0.0 & 0.1 \\ 0.3 & 0.5 \end{bmatrix} \right\|^2$$

$$= \left\| \begin{matrix} 0.1 & 0.1 \\ 0.0 & -0.1 \end{matrix} \right\|^2$$

$$= [(0.1 \times -0.1) - (0.1 \times 0.0)]^2$$

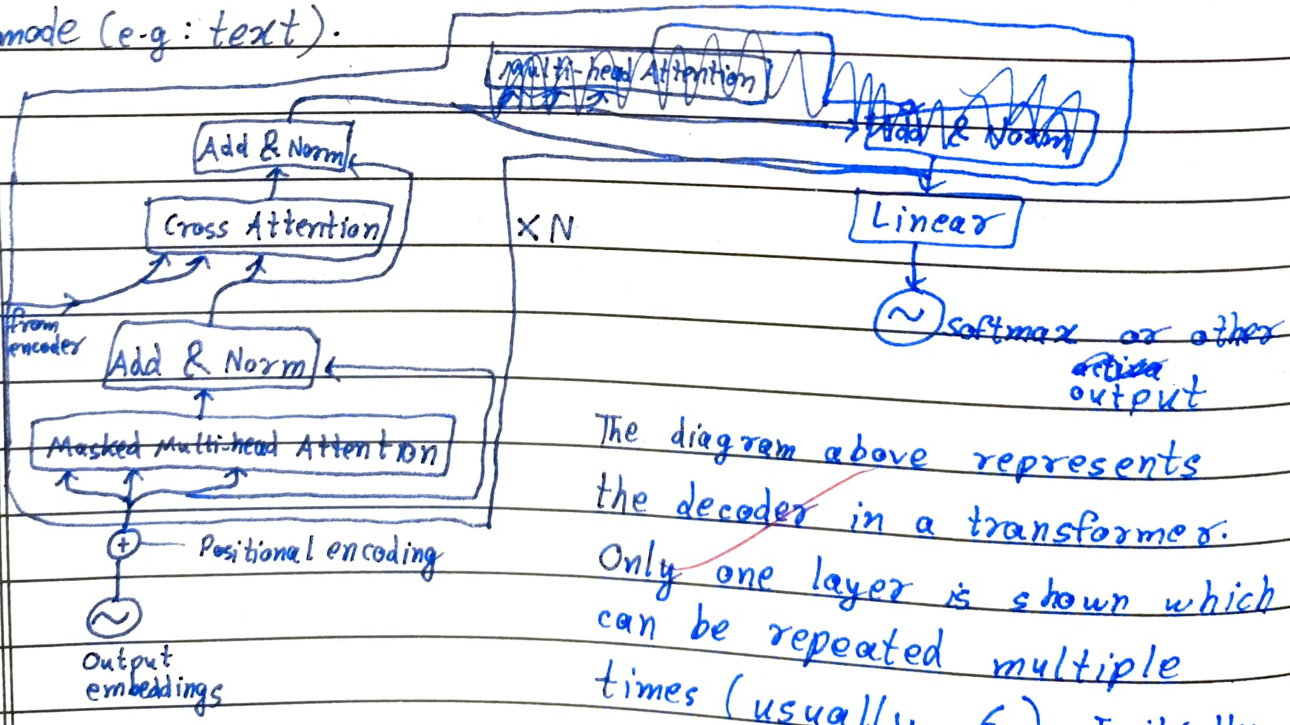
$$= [-0.01 - 0]^2$$

$$= 1 \times 10^{-4} = 0.0001$$

Q3- There are multiple ways and algorithms for training student and teacher models in knowledge distillation. If we only train the student model using a pre-trained teacher, it is offline distillation. If the teacher is trained online along with the student, it is called online distillation. If we use past layers and outputs of the model to help train future iterations, it is known as self-distillation. In terms of knowledge, if we look at soft outputs of teacher then it is response-based, ~~If~~ we can also look at activations of hidden layers, ~~it is relation based or~~ or means between teacher-student feature maps. In terms of actual distillation algorithms, we have adversarial distillation where a teacher, discriminator, and adversarial examples are used. we have cross-modal where teacher pre-trained on one mode of information (e.g: image) can help train student on another mode (e.g: text).

5

Q4.



The diagram above represents the decoder in a transformer. Only one layer is shown which can be repeated multiple times (usually 6). Initially, output embeddings are passed along with positional encoding.

If the decoder is just starting the predictions, usually only a start token embedding is provided. The output embeddings are passed forward to the masked multi-head attention layer which uses past sequence of output to help decoder generate next word. After attention~~s~~ outputs are calculated, they are sent to the Add and Norm layer which performs (instance) layer normalization. Residual connections from output embeddings are also passed here. The next step is for the decoder to carry out multi-headed cross-attention. To do this, Key and Value are provided from encoder output while decoder provides Query. Afterwards another Add & Norm layer processes the data while using ~~a~~ residual connections. The output can then be processed by ~~an~~ multiple more layers of the decoder ~~deco~~ and further processed with linear layers. Lastly, the final text output is calculated via an activation like softmax to find next probable word. The generated data can then be added to output and entire process repeated if we wish for sequence of outputs (sequence of words).