INDIAN INSTITUTE OF TECHNOLOGY KHARAGPUR

Stamp / Signature of the Invigilator

EXAMINATION (End Semester)										**SEMESTER (Spring)**			
Roll Number										**Section**		**Name**	
Subject Number	C	S	6	0	0	1	0	**Subject Name**		***Deep Learning***			
Department / Center of the Student											**Additional sheets**		

Important Instructions and Guidelines for Students

1. You must occupy your seat as per the Examination Schedule/Sitting Plan.
2. Do not keep mobile phones or any similar electronic gadgets with you even in the switched off mode.
3. Loose papers, class notes, books or any such materials must not be in your possession, even if they are irrelevant to the subject you are taking examination.
4. Data book, codes, graph papers, relevant standard tables/charts or any other materials are allowed only when instructed by the paper-setter.
5. Use of instrument box, pencil box and non-programmable calculator is allowed during the examination. However, exchange of these items or any other papers (including question papers) is not permitted.
6. Write on both sides of the answer script and do not tear off any page. **Use last page(s) of the answer script for rough work.** Report to the invigilator if the answer script has torn or distorted page(s).
7. It is your responsibility to ensure that you have signed the Attendance Sheet. Keep your Admit Card/Identity Card on the desk for checking by the invigilator.
8. You may leave the examination hall for wash room or for drinking water for a very short period. Record your absence from the Examination Hall in the register provided. Smoking and the consumption of any kind of beverages are strictly prohibited inside the Examination Hall.
9. Do not leave the Examination Hall without submitting your answer script to the invigilator. **In any case, you are not allowed to take away the answer script with you.** After the completion of the examination, do not leave the seat until the invigilators collect all the answer scripts.
10. During the examination, either inside or outside the Examination Hall, gathering information from any kind of sources or exchanging information with others or any such attempt will be treated as '**unfair means**'. Do not adopt unfair means and do not indulge in unseemly behavior.

Violation of any of the above instructions may lead to severe punishment.

Signature of the Student

To be filled in by the examiner

Question Number	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**Total**
Marks Obtained											

Marks obtained (in words)	**Signature of the Examiner**	**Signature of the Scrutineer**

CS60010 Deep Learning, Spring 2023–2024

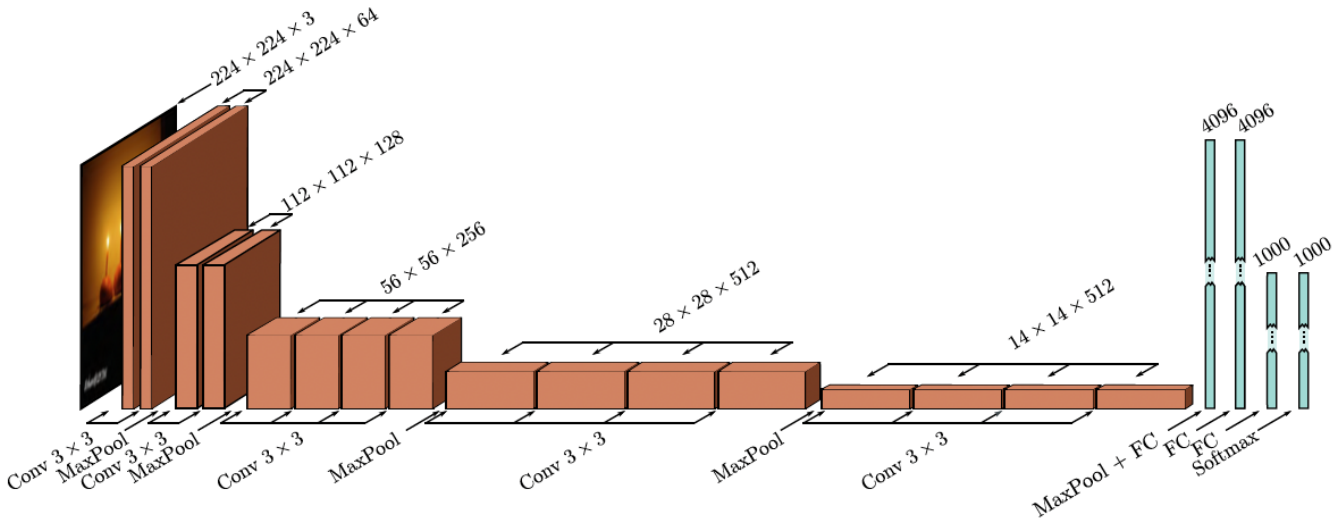
26-Apr-2024, 02:00pm–05:00pm **End-Semester Test** Maximum marks: 80

Instructions

- Write your answers in the respective space provided in the question paper itself. Be brief and precise. Please provide the final answer with related calculations.
- Answer all questions.
- There are no clarifications. In case of confusion, you can make a valid assumption, state that properly and proceed.

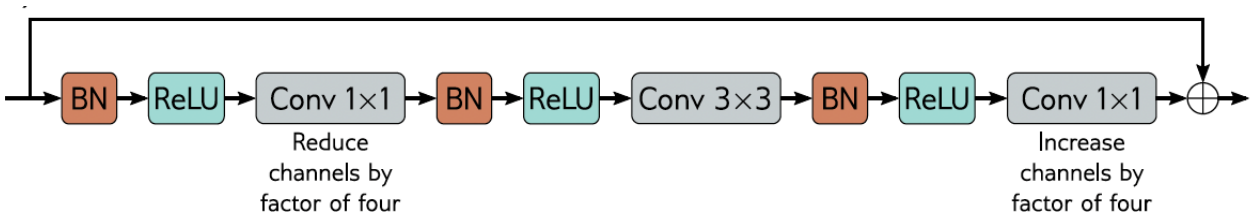
Questions start from the next page.

1. How many weights and biases are there at each convolutional layer and fully connected layer in the VGG architecture shown below? **(8)**



- Between the image and layer 1, there are $3 \times 64 \times 3 \times 3 = x$ weights and 64 biases.
- Between layer 1 and layer 2, there are $64 \times 64 \times 3 \times 3 = 36,384$ weights 64 biases
- Between layer 2 and layer 3, there are $64 \times 128 \times 3 \times 3 = 73,728$ weights and 128 biases.
- Between layer 3 and layer 4, there are $128 \times 128 \times 3 \times 3 = 147,456$ weights and 128 biases
- Between layer 4 and layer 5, there are $128 \times 256 \times 3 \times 3 = 294,912$ weights and 256 biases
- Between layer 5 and layer 6, there are $256 \times 256 \times 3 \times 3 = 589,824$ weights and 256 biases
- Between layer 6 and layer 7, there are $256 \times 256 \times 3 \times 3 = 589,824$ weights and 256 biases
- Between layer 7 and layer 8, there are $256 \times 256 \times 3 \times 3 = 589,824$ weights and 256 biases
- Between layer 8 and layer 9, there are $256 \times 512 \times 3 \times 3 = 1,179,648$ weights and 512 biases
- Between layer 9 and layer 10, there are $512 \times 512 \times 3 \times 3 = 2,359,256$ weights and 512 biases
- Between layer 10 and layer 11, there are $512 \times 512 \times 3 \times 3 = 2,359,256$ weights and 512 biases
- Between layer 11 and layer 12, there are $512 \times 512 \times 3 \times 3 = 2,359,256$ weights and 512 biases
- Between layer 12 and layer 13, there are $512 \times 512 \times 3 \times 3 = 2,359,256$ weights and 512 biases

2. Consider a convolutional residual block that contains a batch normalization operation, followed by a ReLU activation function, and then a 3×3 convolutional layer. If the input and output both have 512 channels, how many parameters are needed to define this block? Now consider a bottleneck residual block that contains three batch normalization/ReLU/convolution sequences. The first uses a 1×1 convolution to reduce the number of channels from 512 to 128. The second uses a 3×3 convolution with the same number of input and output channels. The third uses a 1×1 convolution to increase the number of channels from 128 to 512 (see Figure below). How many parameters are needed to define this block? **(6)**



The original block has $512 \times 512 \times 3 \times 3 = 2{,}359{,}296$ parameters for the convolution plus $512 \times 2 = 1024$ parameters for the BatchNorm. The bottleneck block has $512 \times 128 \times 1 \times 1 + 128 * 128 \times 3 \times 3 + 128 \times 512 \times 1 \times 1 = 278{,}528$ parameters for the convolution plus $512 \times 2 + 128 \times 2 + 128 \times 2 = 1536$ parameters for the BatchNorm.

3.

(a) Consider the problem of predicting the next word from a sequence of words using a fixed window neural language model (using the context of 4 words to predict the next word). Suppose, there are overall 50,000 words in your vocabulary, and each word is represented using a 300-dimension vector. Suppose the hidden layer has 50 dimensions, what are the total number of parameters that you will have to learn? Ignore the bias terms for this problem. **(2.5)**

Solution Input dimensions: 300 * 4 = 1200. Number of weights: 1200 * 50 + 50 * 50000 = 2,560,000

(b) Suppose you are using Bi-LSTM for NER problem, and there are 3 NER tags: $\{PER, LOC, ORG\}$, and you are using BIO tagging scheme. Assume that each word has a 300-dimension embedding, and the hidden state for the forward and backward LSTMs are 200 and 100 dimensions, respectively. How many total parameters will need to be trained? Ignore the bias terms. **(4.5)**

Solution No. of tags in BIO format: 3*2+1 = 7. In an LSTM, there are 4 U, W matrices. So, U, W parameters in forward: 4 * (200*200 + 200*300) = 400,000. U, W parameters in backward: 4 * (100*100 + 100*300) = 160,000. For the V matrix, hidden dimensions = 300, so no. of parameters = 300 * 7 = 2,100. Total: 562,100

(c) Suppose you are using BERT-base for topic classification of documents with 5 classes. How many task-specific parameters will you need to use? **(2)**

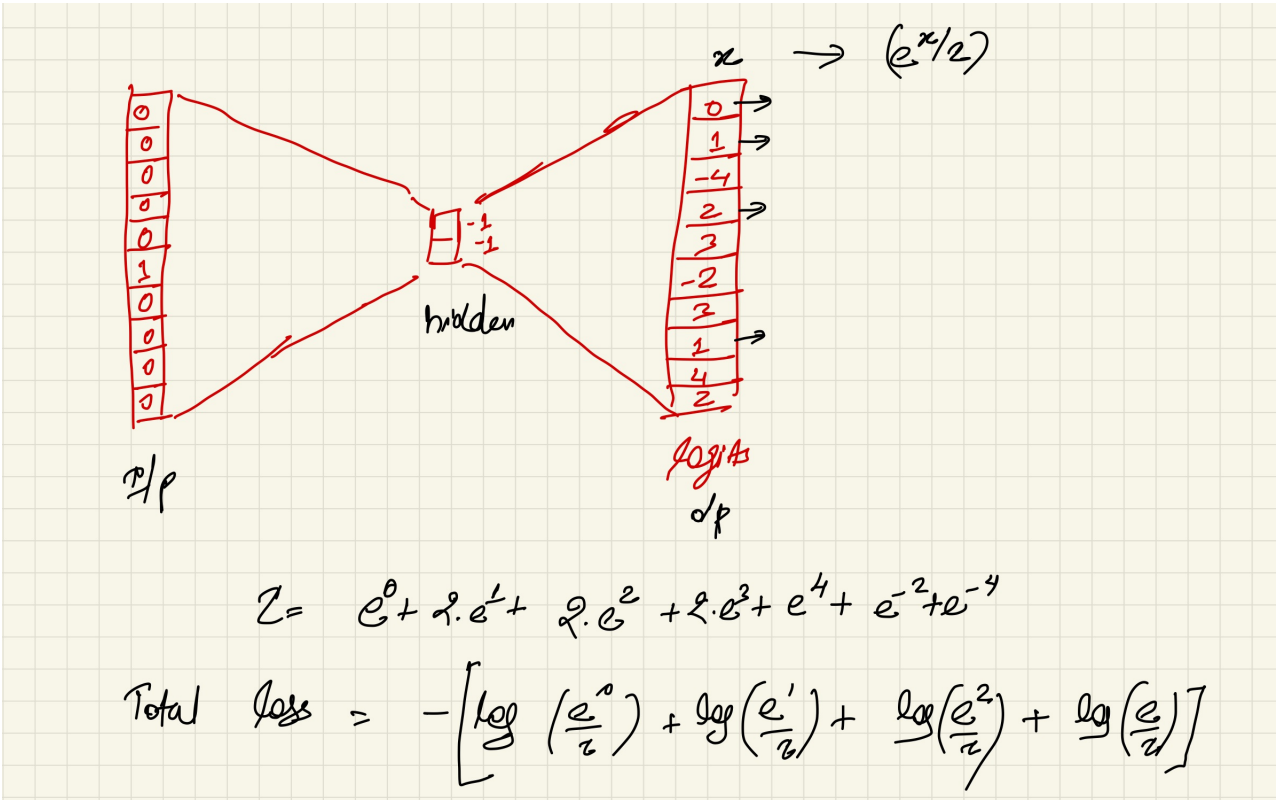
Solution 768 * 5 = 3,840

4. Suppose you are computing the word vectors using Skip-gram architecture. You have 10 words in your vocabulary, $\{hi, you, they, are, am, how, why, there, who, what\}$ in that order and suppose you have the window, 'hi there how are you' in your corpora. You use this window with 'how' as the center word and two words before and after the center word as your context. Also, suppose that for each word, you have 2-dim in and out vectors, which have the following value at this point given as follows:

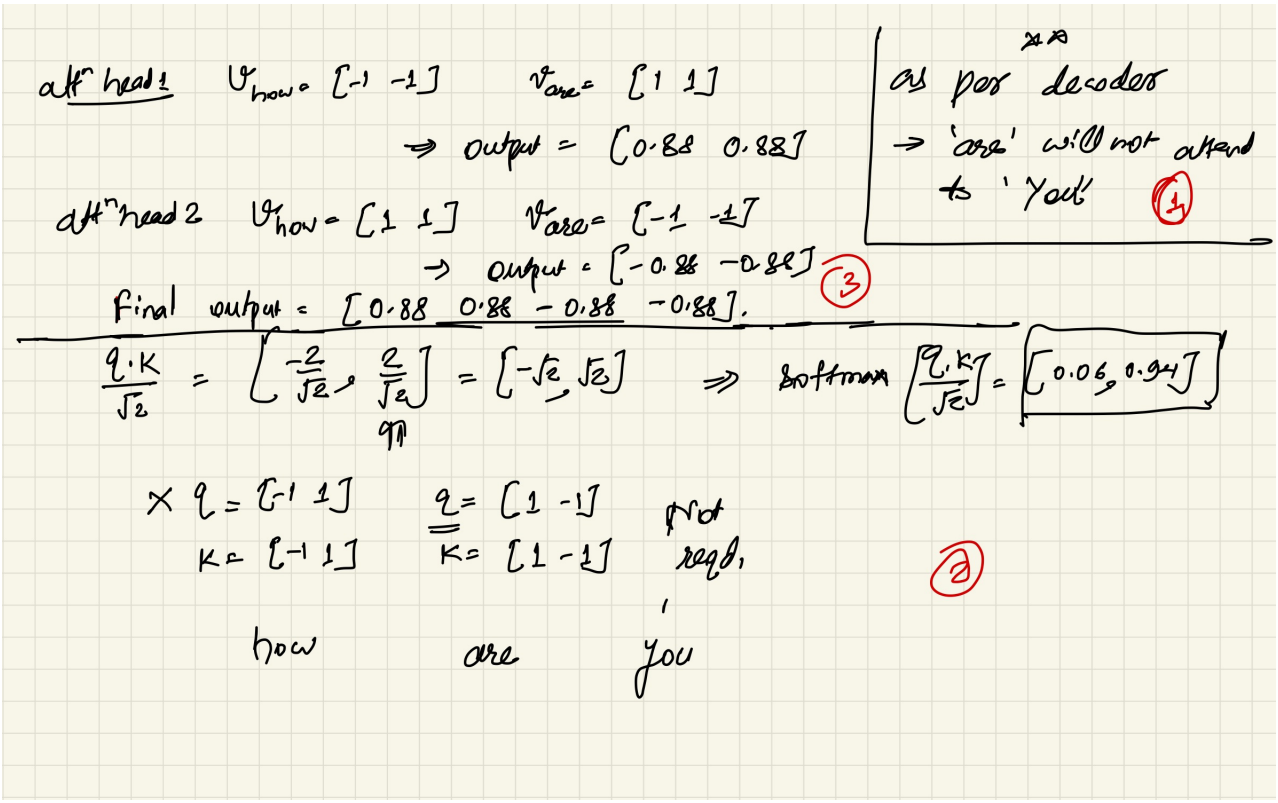
Word	In-vector	Out-vector
hi	(1, -1)	(-2, 2)
you	(-1, 2)	(1, -2)
they	(-2, -2)	(2, 2)
are	(1, 1)	(-1, -1)
am	(2, 1)	(-2, -1)
how	(-1, -1)	(1, 1)
why	(1, 2)	(-1, -2)
there	(2, -1)	(-2, 1)
who	(2, 2)	(-2, -2)
what	(1, 1)	(-1, -1)

Table 1: In and Out representations for words

What will the values at the input, hidden and output layer as per the Skip-gram architecture? What would be the total loss for this window? **(6)**



5. Suppose, you are training your transformer decoder with the Ground truth: 'how are you'. The embeddings for these words are the concatenation of in and out embeddings shown in Table 1. Suppose you are using multi-headed attention with $h = 2$. For both your attention heads, the first and third dimensions of the input define your query vector, and the second and fourth dimensions define your key vector. For the first attention head, the value is the in-vector, and for the second attention head, the value is the out-vector. What will be the output of the multi-headed self-attention block for the word 'are' (before applying the projection matrix W_0)? You are using the scaled dot vector for self-attention. Ignore the start-of-sequence token. **(6)**



6.

(a) Suppose you are using beam search decoding for generation with a beam size of 10. By the fourth step in the decoder, i.e., when you are generating the fourth word in the sequence, what is the number of hypotheses for which you will have computed the log probability? **(2)**

Solution First step: 10 values, Next steps: maximum 10*10 = 100 per step. So, total 310.

(b) Suppose your vocabulary is of size 10. At the time of decoding, the probability distribution by applying softmax to the logits is shown below. How will this distribution modify if a) you are using Top-k sampling with $k = 5$, b) you are using Top-p sampling with $p = 0.6$? Please show the resultant distributions. **(3)**

Vocab	
I	.01
you	.01
they	.01
are	.01
am	.01
how	.50
why	.10
where	.10
who	.15
what	.10

Top-K sampling K=5

[0 0 0 0 0 0.5 0.1 0.1 0.15 0.1]

↓ renormalize

[0 0 0 0 0 0.526 0.105 0.105 0.157 0.105]

Top-p sampling p=0.6

[0 0 0 0 0 0.5 0 0 0.15 0]

↓ renormalize

[0 0 0 0 0 0.77 0 0 0.23 0]

(c) During BART pretraining, the sentence can be corrupted through 'token masking' and / or 'text infiling', among other possible methods. Use an example sentence, "I went to the store" to show that these objectives can lead to different inputs to the encoder. Which of these objectives is closer to T5 pretraining? Show the corresponding input and output for T5 pretraining. **(3)**

Example

Token Masking

I/p I <MASK> to the store

Text infilling

I/p I ____ the store

* None from one tokens to be infilled

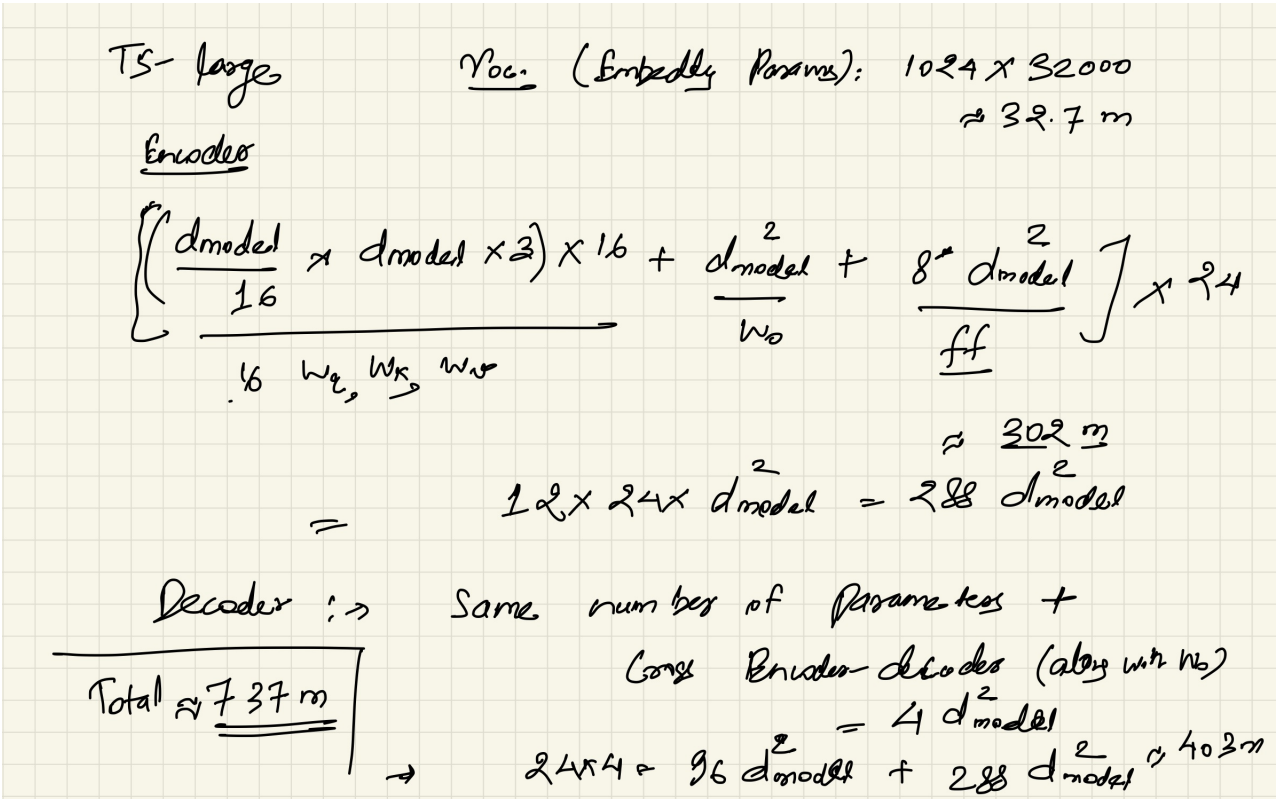
Text infilling is closer to T5 pretraining

T5 I/p I <X> the store

O/p <X> went to <Y>

7.

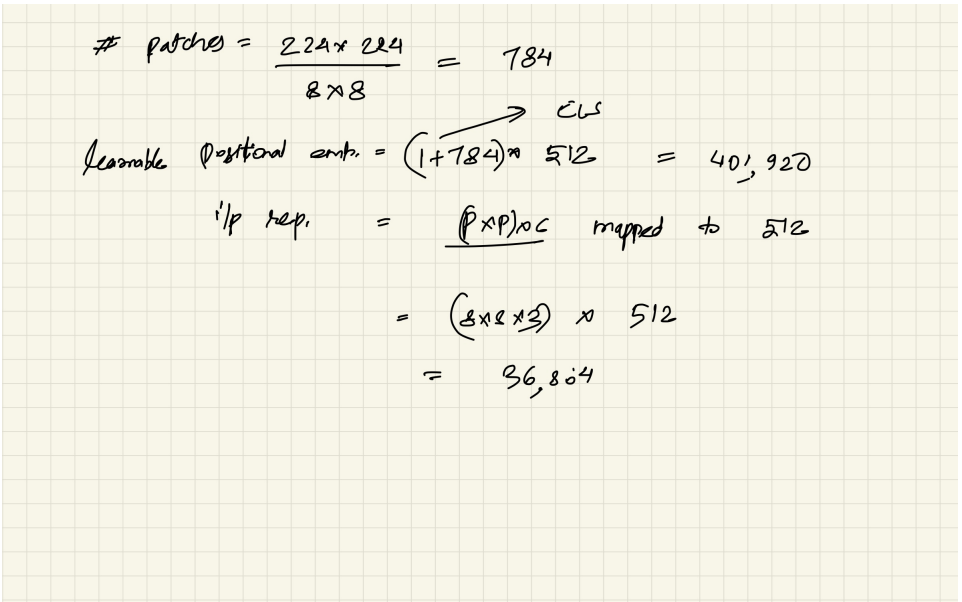
(a) Consider the T5-Large pretrained model, which has a model-dimensionality of 1024, and a vocabulary size of $32k$. It has 24 transformer layers, and has 16 attention heads (for all different attentions). The feed-forward network has 4096 nodes. How many parameters does T5-Large has? **(5)**



(b) While trying to train your LLM, you got 10^6 times more compute. You can use it to increase your model size by a factor of X, and/or increase your data (steps * batch size) by a factor of Y. What would be ideal values for X and Y? **(2)**

Solution $X = 10^4, Y = 10^2$

(c) Suppose you are using (Vision) Transformer to encode images of dimensions 224×224 using patches of size 8×8. Assume that the model dimensions are 512, and a learnable class token embedding is prepended. What will be the number of learnable parameters for the (a) positional embeddings, and (b) input representation? **(3)**



8.

(a) For the RLHF training, how is the reward model RM_ϕ trained? Which model is typically used? What data is required, and what is the loss function? **(3)**

8.(a) Typically RM_ϕ is initialized from the SFT model.

Data: Pairwise comparison data is required

s^w, s^l

s^w: winning sample; s^l: losing sample

$$\text{Loss } f^n = -E_{(s^w, s^l)}\left[\log \sigma\left(RM_\phi(s^w) - RM_\phi(s^l)\right)\right]$$

(b) While training with policy gradient, the weight update looks exactly like the normal cross-entropy loss, weighted by the reward score. But is there any other key difference? **(2)**

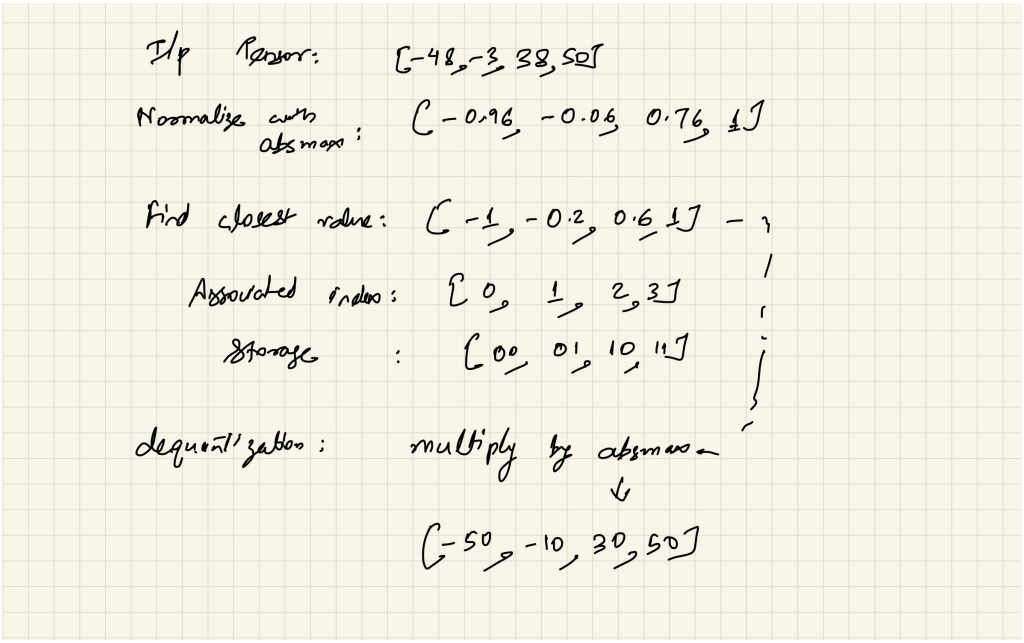
Solution The key difference is that the sentence in policy gradient is model-generated, while in cross-entropy loss is the ground truth.

(c) During RLHF training, the reward function $RS(s)$ is not exactly the reward model $RM_\phi(s)$, but has a penalty term. What is the penalty term, and what is the rationale behind using this? **(2)**

Solution Penalty term has the form: $-\beta log\frac{p_\theta^{RL}(s)}{p^{PT}(s)}$, where p^{PT} is the pretraining (SFT) model, from which RL model is initialized. The rationale of this term is to ensure that the wights do not diverge too much from the SFT model (avoid reward hacking)

9.

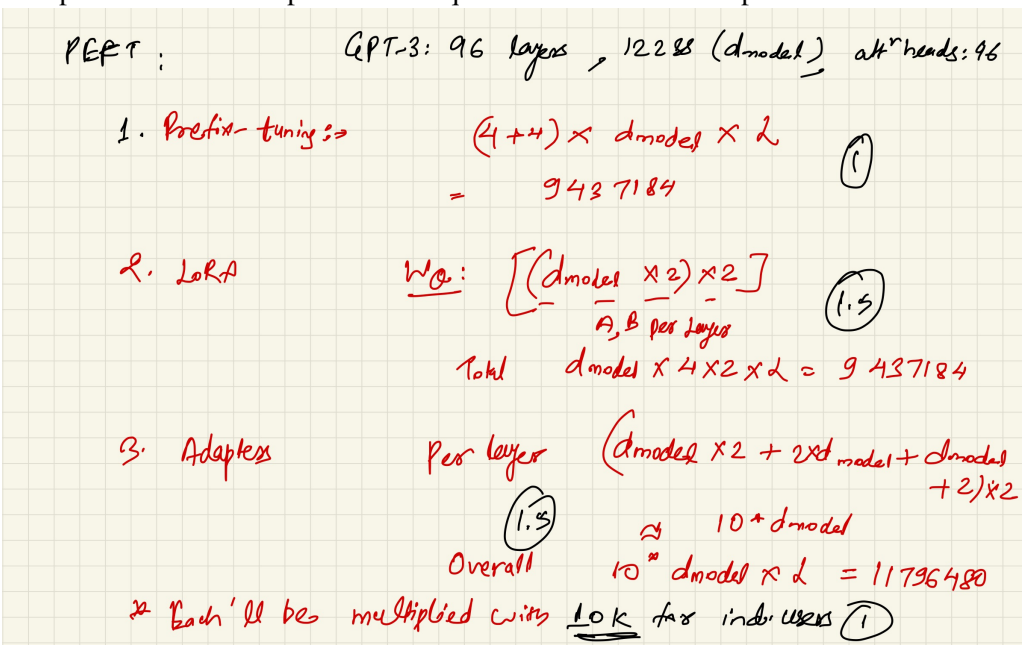
(a) Suppose you are using QLoRA to achieve 2-bit quantization with values: $\{-1, -0.2, 0.6, 1\}$, and your input tensor is $[-48, -3, 38, 50]$. What will you need to store in the quantized format. Also, what will be the output after dequantization? **(3)**



(b) Suppose you are using GPT-3 for the query auto-completion task, where the user types a prefix, and the system tries to complete this. You want to personalize this by fine-tuning individually for $10,000$ users of the system. You have the following options:

1. You can use prefix-tuning at all the layers with length of prefix as 4 and length of infix as 4
2. You can use LoRA for the query and key matrices at all the layers, with rank $r = 2$
3. You can use adapters (2 adapters per layer), with the projection dimension as 2

Compare the number of parameters required for each of these options. **(5)**



10.

(a) In the VAE, the encoder predicts a mean and standard deviation, rather than just a latent vector as in Autoencoder. However, this may just do local smoothing, with no continuity in the space. What is done to ensure the global smoothness? **(2)**

Solution To achieve global smoothness, a KL divergence loss between the distribution given by the encoder, and a standard normal distribution, is introduced.

(b) For the VAE model, derive the evidence-based lower bound in terms of the encoder and decoder parameters θ and ϕ. [Only derive the bound to maximize $log[Pr(x|\phi)]$, no need to simplify this.] **(4)**

$$\begin{aligned}\log[Pr(\mathbf{x}|\boldsymbol{\phi})] &= \log\left[\int Pr(\mathbf{x},\mathbf{z}|\boldsymbol{\phi})d\mathbf{z}\right] \\ &= \log\left[\int q(\mathbf{z})\frac{Pr(\mathbf{x},\mathbf{z}|\boldsymbol{\phi})}{q(\mathbf{z})}d\mathbf{z}\right], \qquad (17.14)\end{aligned}$$

We then use Jensen's inequality for the logarithm (equation 17.12) to find a lower bound:

$$\log\left[\int q(\mathbf{z})\frac{Pr(\mathbf{x},\mathbf{z}|\boldsymbol{\phi})}{q(\mathbf{z})}d\mathbf{z}\right] \geq \int q(\mathbf{z})\log\left[\frac{Pr(\mathbf{x},\mathbf{z}|\boldsymbol{\phi})}{q(\mathbf{z})}\right]d\mathbf{z}, \qquad (17.15)$$

where the right-hand side is termed the *evidence lower bound* or *ELBO*. It gets this name because $Pr(\mathbf{x}|\boldsymbol{\phi})$ is called the *evidence* in the context of Bayes' rule (equation 17.19).

In practice, the distribution $q(\mathbf{z})$ has parameters $\boldsymbol{\theta}$, so the ELBO can be written as:

$$\text{ELBO}[\boldsymbol{\theta},\boldsymbol{\phi}] = \int q(\mathbf{z}|\boldsymbol{\theta})\log\left[\frac{Pr(\mathbf{x},\mathbf{z}|\boldsymbol{\phi})}{q(\mathbf{z}|\boldsymbol{\theta})}\right]d\mathbf{z}. \qquad (17.16)$$

(c) In the diffusion model training, show that the reconstruction term $-log[Norm_{x_i}[f_1[z_{i1},\phi_1],\sigma_1^2 I]$ can be expressed as a mean squared loss to predict the noise using the decoder. **(4)**

Solution $-log[Norm_{x_i}[f_1[z_{i1},\phi_1],\sigma_1^2 I]$ can be simplified as follows (from loss functions, ignore the argmin and summation expressions

$$\begin{aligned}
\hat{\phi} &= \underset{\phi}{\text{argmin}}\left[-\sum_{i=1}^{I}\log\left[\frac{1}{\sqrt{2\pi\sigma^2}}\exp\left[-\frac{(y_i-\text{f}[\mathbf{x}_i,\phi])^2}{2\sigma^2}\right]\right]\right] \\
&= \underset{\phi}{\text{argmin}}\left[-\sum_{i=1}^{I}\left(\log\left[\frac{1}{\sqrt{2\pi\sigma^2}}\right]-\frac{(y_i-\text{f}[\mathbf{x}_i,\phi])^2}{2\sigma^2}\right)\right] \\
&= \underset{\phi}{\text{argmin}}\left[-\sum_{i=1}^{I}-\frac{(y_i-\text{f}[\mathbf{x}_i,\phi])^2}{2\sigma^2}\right] \\
&= \underset{\phi}{\text{argmin}}\left[\sum_{i=1}^{I}(y_i-\text{f}[\mathbf{x}_i,\phi])^2\right],
\end{aligned}$$

Solution Now, use the following definitions.

$$\mathbf{x} = \frac{1}{\sqrt{\alpha_t}}\cdot\mathbf{z}_t - \frac{\sqrt{1-\alpha_t}}{\sqrt{\alpha_t}}\cdot\boldsymbol{\epsilon}.$$

$$\mathbf{f}_t[\mathbf{z}_t,\boldsymbol{\phi}_t] = \frac{1}{\sqrt{1-\beta_t}}\mathbf{z}_t - \frac{\beta_t}{\sqrt{1-\alpha_t}\sqrt{1-\beta_t}}\mathbf{g}_t[\mathbf{z}_t,\boldsymbol{\phi}_t].$$

Solution for $t=1$, it simplifies to

$$\frac{1}{2\sigma_1^2}\left\|\mathbf{x}_i-\mathbf{f}_1[\mathbf{z}_{i1},\boldsymbol{\phi}_1]\right\|^2 = \frac{1}{2\sigma_1^2}\left\|\frac{\beta_1}{\sqrt{1-\alpha_1}\sqrt{1-\beta_1}}\mathbf{g}_1[\mathbf{z}_{i1},\boldsymbol{\phi}_1]-\frac{\beta_1}{\sqrt{1-\alpha_1}\sqrt{1-\beta_1}}\boldsymbol{\epsilon}_{i1}\right\|^2. \tag{18.37}$$

(d) In the diffusion model, why does the decoder model sample from $Norm_{Z_T}[0,I]$? [*Show that when the encoder is run for sufficiently long number of steps, it converges to this.*] **(2)**

Solution The diffusion kernel can be written as

$$\mathbf{z}_t = \sqrt{\alpha_t}\cdot\mathbf{x} + \sqrt{1-\alpha_t}\cdot\boldsymbol{\epsilon},$$

Solution where $\alpha_t = \prod_{s=1}^{T}(1-\beta_s)$. As $t\to\infty$, this converges to $Norm[0,I]$