ELL881/AIL821

Large Language Models: Introduction and Recent Advances

Semester I, 2024-25

Mid-Semester Examination
IIT Delhi

Total Marks: 60
Time: 2 hours

- Answers should be brief and to the point. Writing unnecessary stuff will be penalized.
- Write proper justifications for every answer. Answers with no justifications will fetch *zero* marks (except for MCQs).

Section-A

Multiple Choice Questions (10 marks)

Each question carries 2 marks. MORE THAN ONE OPTION CAN BE CORRECT FOR EACH QUESTION. There is NO PARTIAL MARKING – only marking ALL options correctly will fetch marks.

1. Consider a unigram language model that estimates the probability of words based on their frequency in a training corpus. To address the problem of unseen words, various smoothing techniques are employed. Which of the following statement(s) regarding smoothing techniques is/are correct?

   (A) Laplace smoothing adds a constant value to all word counts to prevent zero probabilities.
   (B) Good-Turing smoothing redistributes the probability mass by adjusting only the counts of the unseen words.
   (C) Kneser-Ney smoothing considers the probability of word sequences based on the diversity of their contexts.
   (D) Smoothing techniques tend to increase perplexity in language models.

2. Word2vec and GloVe are popular word embedding models. Which of the following statement(s) accurately describes key differences between the two models?

   (A) Word2vec uses a predictive model based on context windows, while GloVe uses a co-occurrence matrix.
   (B) GloVe captures global statistical information about word co-occurrences, while Word2vec focuses on predicting local context.
   (C) Word2vec's embeddings are context-independent, while GloVe provides context-dependent embeddings.
   (D) Word2vec often employs negative sampling for efficient training, while GloVe does not require negative sampling.

3. Recurrent Neural Networks (RNNs) suffer from issues with long-term dependencies, which LSTMs and GRUs help address. Transformers introduce self-attention mechanisms that further mitigate these issues. Which of the following is/are true about LSTMs, GRUs, and Transformers?

   (A) Both LSTMs and GRUs use gating mechanisms to control information flow across time steps.
   (B) Transformers do not require sequential processing and hence can be parallelized more efficiently than RNNs.
   (C) LSTMs explicitly store both short-term and long-term dependencies through cell states, while GRUs only maintain a hidden state.
   (D) Self-attention mechanisms in transformers completely eliminate the need for positional encoding.

4. ELMo and BERT represent two different pre-training strategies for language models. Which of the following statement(s) about these approaches is/are true?

   (A) ELMo uses a bi-directional LSTM to pre-train word representations, while BERT uses a transformer encoder with masked language modeling.
   (B) ELMo provides context-independent word representations, whereas BERT provides context-dependent representations.
   (C) BERT's pre-training involves masking a random subset of tokens and predicting them, while ELMo predicts the next word in a sequence.
   (D) Both ELMo and BERT produce word embeddings that can be fine-tuned for downstream tasks.

5. Flash Attention and KV Caching are two methods designed to improve the efficiency of attention mechanisms in transformers. Which of the following statement(s) is/are correct about these methods?

   (A) KV Caching helps reduce the computational cost of attending to previously seen tokens in auto-regressive models by reusing key and value embeddings from previous time steps.
   (B) Flash Attention improves speed during attention computation by reducing the number of I/O operations between GPU SRAM and HBM.
   (C) KV Caching is primarily useful in auto-regressive generation, while Flash Attention optimizes both training and inference by reducing the complexity of attention computation.
   (D) Both Flash Attention and KV Caching reduce the memory footprint of transformers by sharing attention weights across layers.

Section-B

(50 marks)

Question 1: Statistical Language Models and Perplexity (8 marks)

(a) Suppose you have a bigram language model trained on a corpus with a vocabulary size of $V = 10000$. After training, the bigram counts $C(w_{i-1}, w_i)$ and unigram counts $C(w_{i-1})$ for some word pairs are:

Bigram	Count $C(w_{i-1}, w_i)$	Unigram Count $C(w_{i-1})$
(the, cat)	2500	5000
(the, mat)	800	5000
(cat, sat)	1200	3000
(sat, on)	900	1800
(on, the)	1800	2800

Compute the smoothed bigram probability $\hat{P}_{\text{Laplace}}(w_i|w_{i-1})$ for the bigram (the, cat) using Laplace smoothing. Show all calculations. (1.5 marks)

(b) Now, compute the perplexity of the bigram language model described in part (a) on the following sentence: '*the cat sat on the mat*', if the word '*the*' appears at the start of a sentence in 60% cases. Assume Laplace smoothing. Show all calculations. (2.5 marks)

(c) Write the expression for the adjusted count $\tilde{c}$ of a word w after Good-Turing smoothing. The original count of the word w in the corpus is c. Assume that there are α words appearing $(c-1)$ times, β words appearing c times and γ words appearing $(c+1)$ times in the corpus. The expression for $\tilde{c}$ should be in terms of c, α, β and γ only. (1 mark)

(d) What is the relation between the cross-entropy of a language model $\mathcal{M}$ and its perplexity on a sequence W consisting of N words? Clearly state the definition and expression for cross-entropy of a model on a probability distribution, as well as the assumptions you need to make to derive the relation with perplexity for a language model. (3 marks)

Question 2: Word Embeddings (5 marks)

Suppose you want to use the skip-gram model to learn **bigram embeddings** instead of word embeddings. Recall the basic skip-gram model:

$$P(\text{context} = y \mid \text{word} = x) = \frac{\exp(v_x^T c_y)}{\sum_{y' \in \text{vocab}} \exp(v_x^T c_{y'})}$$

You are going to learn a bigram vector for every bigram. That is, for a sentence like "the cat sat", you will form (word, context) pairs: $(b(\text{the cat}), u(\text{sat}))$, $(b(\text{cat sat}), u(\text{the}))$, where b and u denote bigram and unigram respectively (for maximal clarity). **Note that contexts are still unigrams**.

(a) What is the big-O runtime of computing the probability for a single bigram-context pair? Express the answer in terms of the vocabulary size $|V|$, the number of bigrams $|B|$ attested in the data, and the dimension of the vectors d. (2.5 marks)

(b) How many parameters are in the model? Express this in terms of the quantities in part (a). Your answer to this part should be exact. (2.5 marks)

Question 3: Neural Language Models (4 marks)

Assume that you are using an LSTM for performing the task of sentiment analysis. Recall the units of an LSTM cell are defined as:

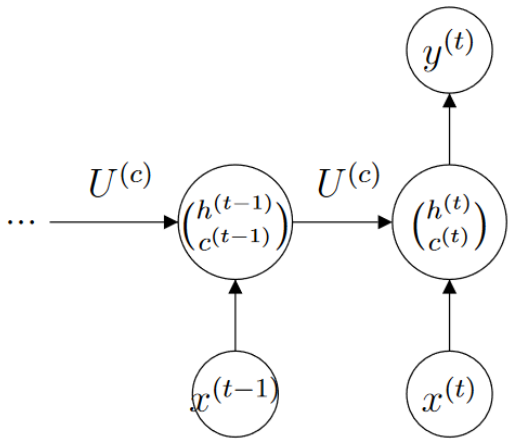
$$i_t = \sigma(W^{(i)} x_t + U^{(i)} h_{t-1})$$
$$f_t = \sigma(W^{(f)} x_t + U^{(f)} h_{t-1})$$
$$o_t = \sigma(W^{(o)} x_t + U^{(o)} h_{t-1})$$

$$\tilde{c}_t = \tanh(W^{(c)}x_t + U^{(c)}h_{t-1})$$
$$c_t = f_t \circ c_{t-1} + i_t \circ \tilde{c}_t$$
$$h_t = o_t \circ \tanh(c_t)$$

where the final output of the last LSTM cell is defined by:

$$\hat{y}_t = \text{softmax}(h_t W + b)$$

The final cost function J uses the cross-entropy loss. Consider an LSTM for two time steps, t and $t-1$.



(a) Derive the gradient $\frac{\partial J}{\partial U^{(c)}}$ in terms of the following gradients: $\frac{\partial h_t}{\partial h_{t-1}}$, $\frac{\partial h_{t-1}}{\partial U^{(c)}}$, $\frac{\partial J}{\partial h_t}$, $\frac{\partial c_t}{\partial U^{(c)}}$, $\frac{\partial c_{t-1}}{\partial U^{(c)}}$, $\frac{\partial c_t}{\partial c_{t-1}}$, $\frac{\partial h_t}{\partial c_t}$, and $\frac{\partial h_t}{\partial o_t}$.
Not all of the gradients may be used. You can leave the answer in the form of chain rule and do not have to calculate any individual gradients in your final result. (3 marks)

(b) Rather than using the last hidden state to output the sentiment of a sentence, what could be a better solution to improve the performance of the sentiment analysis task? (1 mark)

Question 4: Transformers (15 marks)

(a) Compare the computational complexity (time) of a vanilla Transformer vs an LSTM while generating sequences of length n at test time. (1.5+1.5 = 3 marks)

(b) Compute the number of matrix multiplications required to generate a sequence of length n by a single Transformer decoder (with only a single-head masked-attention layer and a MLP block; assume no Layer norm). Recompute the same when using a key-value cache. (3+2 = 5 marks)

(c) Can you use a BERT-like [CLS] token at the beginning of a sentence to get sentence representation with GPT-2? Explain your answer. (2 marks)

(d) Compute the number of trainable parameters for an attention layer with 6 attention heads for - (i) multi-head, (ii) multi-query, and (iii) grouped-query (with group-size 2) attention mechanisms. Assume the embedding dimension $d_{model} = 4$, the dimensions of each key, query and value vector to be identical $d_k = d_v = 3$. Also, assume that the embedding dimension d_{model} is preserved after the attention layer. [*Hint: The number of parameters in a trainable matrix* $A \in \mathbb{R}^{m \times n}$ *is* mn.] (1+1+1 = 3 marks)

(e) Show how Rotary Position Embedding (RoPE) captures the relative position between any two tokens in a sentence. (2 marks)

Question 5: Mixture of Experts (4 marks)

(a) How does the noisy top-k gating mechanism differ from the greedy expert selection? State the equations involved in the gating mechanism of both strategies. Also, compare the number of trainable parameter matrices required in each of the two (only for the gating mechanism; ignore other layers in the architecture). (2.5+1.5 = 4 marks)

Question 6: Scaling Laws (4 marks)

(a) Compare and contrast the Kaplan scaling laws with the Chinchilla scaling laws. You don't need to write the exact numerical value of any parameter associated with any of the laws; just state how they differ in the trends they propose. (2 marks)

(b) Argue the validity of the following statement with proper justification - *The emergent properties of LLMs show a sudden spike with increasing scale of the models irrespective of the evaluation metric used to quantify them.* (2 marks)

Question 7: Aligning LLMs with Human Preferences (10 marks)

(a) What is the primary difference between aligning LLMs with human preferences using *proximal policy optimization* (PPO) and *direct preference optimization* (DPO)? (1.5 marks)

(b) Write the loss function corresponding to the alignment process using PPO. Assume π_{ref} to be the instruction-tuned model, and π_θ to be the model we are aligning. Also, assume that we already have a trained reward model which assigns reward $r(x, y)$ for an output y corresponding to the input x. (1.5 marks)

(c) Now, from the PPO loss function in part (b), derive the loss function for DPO ($\mathcal{L}_{DPO}$). Clearly state the assumptions you make and the extra notations you use. (3 marks)

(c) Come up with an expression for $\nabla_\theta \mathcal{L}_{DPO}(\pi_\theta, \pi_{ref})$, which is the gradient of $\mathcal{L}_{DPO}$ with respect to the parameters of the model being aligned (θ). You don't need to further simplify any term involving $\nabla_\theta log(\pi_\theta(y|x))$. From the expression of $\nabla_\theta \mathcal{L}_{DPO}$, show that the gradient increases the likelihood of preferred outputs and decreases the same for dispreferred ones. (4 marks)