Introduction to Large Language Models
Assignment- 10

Number of questions: 10 **Total mark: 10 X 1 = 10**

QUESTION 1: [1 mark]

How do Prefix Tuning and Adapters differ in terms of where they inject new task-specific parameters in the Transformer architecture?

a. Prefix Tuning adds new feed-forward networks after every attention block, while Adapters prepend tokens.
b. Both approaches modify only the final output layer but in different ways.
c. Prefix Tuning learns trainable “prefix” hidden states at each layer’s input, whereas Adapters insert small bottleneck modules inside the Transformer blocks.
d. Both approaches rely entirely on attention masks to inject new task-specific knowledge.

Correct Answer: c

Explanation:

- **Prefix Tuning:** In this approach, we learn a sequence of “prefix” embeddings - trainable hidden states - that get prepended to the model’s internal representations at each layer. This means the main Transformer weights stay frozen, and the prefix acts like extra context or “virtual tokens” that the model attends to.

- **Adapters:** Adapters insert small, trainable modules (often feed-forward “bottleneck” layers) inside each Transformer layer, typically after the attention or feed-forward sub-layers. They are small enough to keep the main model weights mostly intact while still enabling fine-tuning for new tasks.

Therefore, the main difference is that Prefix Tuning adds trainable “prefix” embeddings at the input side of each layer, while Adapters add small modules inside the architecture.

QUESTION 2: [1 mark]

The Structure-Aware Intrinsic Dimension (SAID) improves over earlier low-rank adaptation approaches by:

a. Ignoring the network structure entirely
b. Learning one scalar per layer for layer-wise scaling
c. Sharing the same random matrix across all layers
d. Using adapters within self-attention layers

Correct Answer: b

Explanation:

- **Structure-Aware Intrinsic Dimension (SAID):** This method learns a small set of parameters - often including one scalar per layer - to scale or adjust each layer. Instead of ignoring the model's structure, it captures the global scaling behaviour across layers with minimal additional parameters.

- **Why not the others?**

  (a) SAID explicitly does not ignore the network structure; it leverages per-layer learnable parameters.

  (c) Sharing the exact same random matrix across all layers is a different approach (not specifically SAID's main method).

  (d) SAID is more about scalar scaling factors, not inserting adapter modules.

QUESTION 3: [1 mark]

Which of the following are correct about the extensions of LoRA?

a. LongLoRA supports inference on longer sequences using global attention
b. QLoRA supports low-rank adaptation on 4-bit quantized models
c. DyLoRA automatically selects the optimal rank during training
d. LoRA+ introduces gradient clipping to stabilize training

Correct Answer: b, c

Explanation:

- **QLoRA (b)** applies the LoRA idea to a quantized model, typically at 4-bit precision, enabling efficient fine-tuning with significantly reduced memory use.
- **DyLoRA (c)** is an approach that dynamically picks the optimal rank during training, thereby adapting the low-rank decomposition to the task at hand.

- **Why not (a) or (d)?**

  (a) "LongLoRA" is not a commonly recognized extension that specifically implements global attention for longer sequences as part of LoRA. (Some methods support longer-context inference, but that is not typically referred to as "LongLoRA.")

  (d) While "LoRA+" might be used informally in some contexts, the "+" here does not refer to a widely acknowledged official extension that only introduces gradient clipping.

QUESTION 4: [1 mark]

Which pruning technique specifically removes weights with the smallest absolute values first, potentially followed by retraining to recover accuracy?

a. Magnitude Pruning
b. Structured Pruning
c. Random Pruning
d. Knowledge Distillation

Correct Answer: a

Explanation:

- **Magnitude Pruning** removes weights whose absolute values are below a certain threshold (i.e., the smallest magnitudes). The rationale is that weights with small magnitudes contribute less to overall model outputs. After pruning them, one can optionally retrain (also called "fine-tuning" after pruning) to recover lost accuracy.

- **Structured Pruning (b)** removes entire groups of weights (e.g., entire filters or channels).

- **Random Pruning (c)** removes weights randomly, with no magnitude criterion.

- **Knowledge Distillation (d)** is a different approach, transferring knowledge from a teacher to a student model, not a direct pruning method.

QUESTION 5: [1 mark]

In Post-Training Quantization (PTQ) for LLMs, why is a calibration dataset used?

a. To precompute the entire attention matrix for all tokens.
b. To remove outlier dimensions before applying magnitude-based pruning.
c. To fine-tune the entire model on a small dataset and store the new weights.
d. To estimate scale factors for quantizing weights and activations under representative data conditions.

Correct Answer: d

Explanation:

- **Calibration Dataset:** In PTQ, you typically don't retrain the model. Instead, you gather a small "calibration set" of representative examples. By running these examples through the model, you observe the distribution of activations (and possibly weights). This helps you pick appropriate *scale* factors (or quantization parameters) so that the quantized model preserves accuracy as much as possible.

- Thus, it’s about **extracting distribution statistics** to set the quantization scale and zero points.

QUESTION 6: [1 mark]

Which best summarizes the function of the unembedding matrix W_U?

a. It merges the queries and keys for each token before final classification.
b. It converts the final residual vector into vocabulary logits for next-token prediction.
c. It is used for normalizing the QK and OV circuits so that their norms match.
d. It acts as a second attention layer that aggregates multiple heads.

Correct Answer: b

Explanation:

- The **unembedding matrix** (often the transpose of the embedding matrix – if weight sharing is enabled) takes the final hidden state (i.e., the final residual or contextual representation of each token) and maps it to a distribution over the vocabulary. That distribution is used to pick the next token.

- **Why not the others?**

  (a) Merging queries and keys is part of the attention mechanism, not the unembedding step.

  (c) Normalizing QK or OV circuits usually involves layer norms or other scaling parameters, not the unembedding.

  (d) A second attention layer is not typically called the “unembedding” layer.

QUESTION 7: [1 mark]

Which definition best matches an induction head as discovered in certain Transformer circuits?

a. A head that specifically attends to punctuation tokens to determine sentence boundaries
b. A feed-forward sub-layer specialized for outputting next-token probabilities for out-of-distribution tokens
c. A head that looks for previous occurrences of a token A, retrieves the token B that followed it last time, and then predicts B again
d. A masking head that prevents the model from looking ahead at future tokens

Correct Answer: c

Explanation:

- **Induction heads**: These specialized attention heads implement a pattern that can be described like: "If we see a token A repeated, we look back to see what token came after A the last time it appeared, and we hypothesize that token will appear again." This mechanism is behind repeating patterns or reusing local context the model has seen before.
- It is effectively a memory pattern that picks up repeated sequences in text.

QUESTION 8: [1 mark]

In mechanistic interpretability, how can we define 'circuit'?

a. A data pipeline for collecting training examples in an autoregressive model
b. A small LSTM module inserted into a Transformer for additional memory
c. A device external to the neural network used to fine-tune certain parameters after training
d. A subgraph of the neural network hypothesized to implement a specific function or behaviour

Correct Answer: d

Explanation:

- In **mechanistic interpretability**, a *circuit* is a **subgrap**h of a neural network—specific connections and components (e.g., heads, neurons, MLP layers)—that collectively implement a certain interpretable function. Researchers attempt to identify and visualize these circuits to understand *how* the model handles specific patterns or tasks.

QUESTION 9: [1 mark]

Which best describes the role of Double Quantization in QLoRA?

a. It quantizes the attention weights twice to achieve 1-bit representations.
b. It reinitializes parts of the model with random bit patterns for improved regularization.
c. It quantizes the quantization constants themselves for additional memory savings.
d. It systematically reverts partial quantized weights back to FP16 whenever performance degrades.

Correct Answer: c

Explanation:

- **Double Quantization** in **QLoRA** means not only are the main weights quantized, but the scaling factors (quantization constants) are themselves stored in a lower precision format to reduce memory usage further. It's essentially a second layer of quantization on top of the initial quantization scheme, yielding additional memory compression.

- **Why not the others?** There's no step of reinitializing with random bit patterns, nor switching to 1-bit, nor automatically reverting to FP16.

QUESTION 10: [1 mark]

Which of the following are true about sequence-level distillation for LLMs?

a. It trains a student model by matching the teacher's sequence outputs (e.g., predicted token sequences) rather than just individual token distributions.
b. It requires storing only the top-1 predictions from the teacher model for each token.
c. It can be combined with word-level distillation to transfer both local and global knowledge.
d. It forces the teacher to produce a chain-of-thought explanation for each example.

Correct Answer: a, c

Explanation:

- **Sequence-level distillation:** Instead of just matching the teacher's probability distribution at each token, the student is trained to mimic the teacher's entire output sequence (which can capture *global*, multi-token patterns).

    - **(a) True**: The student tries to match the teacher's full sequence outputs, not just per-token probabilities in isolation.

    - **(c) True:** Combining sequence-level with word-level distillation can yield a comprehensive approach where the student learns local token distribution and overall sequence structure.

- **Why not (b) or (d)?**

    - (b) Typically, sequence-level distillation can store more than top-1 predictions (e.g., entire sequences). Just top-1 might lose richer distribution information.

    - (d) It does *not* necessarily require chain-of-thought. Sequence-level distillation focuses on final outputs; it doesn't force an intermediate chain-of-thought explanation.
