Introduction to Large Language Models
Assignment- 12

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

QUESTION 1: [1 mark]

Which statements correctly characterize “bias” in the context of LLMs?

1. Bias can generate objectionable or stereotypical views in model outputs.
2. Bias is always intentionally introduced by malicious data curators.
3. Bias can cause harmful real-world impacts such as reinforcing discrimination.
4. Bias only affects low-resource languages; high-resource languages are unaffected.

a. 1 and 2

b. 1 and 3

c. 2 and 4

d. 1, 3, and 4

Correct Answer: b

Explanation:

- **(1) True:** Model outputs can reflect harmful stereotypes if training data or modelling procedures contain biases.
- **(3) True:** Biased outputs may perpetuate discrimination or unfair treatment in real-world contexts.

- Statements (2) and (4) are not necessarily correct:

    - **(2) False:** Bias in data is often unintentional, reflecting existing societal or historical imbalances.
    - **(4) False:** Bias can affect any language; high-resource languages are not inherently immune.

QUESTION 2: [1 mark]

The Stereotype Score (ss) refers to:

a. The frequency with which a language model rejects biased associations.
b. The measure of how often a model’s predictions are meaningless as opposed to meaningful.
c. A ratio of positive sentiment to negative sentiment in model outputs.
d. The proportion of examples in which a model chooses a stereotypical association over an anti-stereotypical one.

Correct Answer: d

Explanation:

- **Stereotype Score (ss)** is a metric that measures how frequently the model picks a stereotypical continuation or association **instead of** a non-stereotypical or anti-stereotypical one.

- Essentially, it's a proportion (or fraction) of test items for which the model output aligns with the stereotype.

QUESTION 3: [1 mark]

Which of the following are prominent sources of bias in LLMs?

1. Improper selection of training data leading to skewed distributions.
2. Reliance on older datasets causing "temporal bias."
3. Overemphasis on low-resource languages causing "linguistic inversion."
4. Unequal focus on high-resource languages resulting in "cultural bias."

a. 1 and 2 only

b. 2 and 3 only

c. 1, 2, and 4

d. 1, 3, and 4

Correct Answer: c

Explanation:

1. **Improper selection of training data** (true) can lead to some groups or topics being over-represented, causing bias.
2. **Reliance on older datasets** (true) can introduce out-of-date or "temporal bias" that doesn't reflect current social norms or language usage.
3. "Overemphasis on low-resource languages" is not commonly described as "linguistic inversion"; typically the bias is the opposite — under-representation of low-resource languages.
4. **Unequal focus on high-resource languages** (true) can lead to cultural biases and poor performance or misrepresentations of underrepresented cultures.

QUESTION 4: [1 mark]

In the context of bias mitigation based on adversarial triggers, which best describes the goal of prepending specially chosen tokens to prompts?

a. To directly fine-tune the model parameters to remove bias
b. To override all prior knowledge in a model, effectively "resetting" it
c. To exploit the model's distributional patterns, thereby neutralizing or flipping biased associations in generated text
d. To randomly shuffle the tokens so that the model becomes more robust

Correct Answer: c

Explanation:

- **Adversarial triggers** are carefully crafted token sequences that, when prepended to the prompt, steer the model's output in a certain direction (e.g., reducing bias or toxicity). They work within the model's learned distribution rather than overriding its knowledge.
- They do *not* retrain the model; they exploit patterns in the existing parameters to mitigate biased outcomes.

QUESTION 5: [1 mark]

Which of the following best describes the "*regard"* metric?

a. It is a measure of how well a model can explain its internal decision process.
b. It is a measurement of a model's perplexity on demographically sensitive text.
c. It is the proportion of times a model self-corrects discriminatory language.
d. It is a classification label reflecting the attitude towards a demographic group in the generated text.

Correct Answer: d

Explanation:

- **Regard** is typically measured by classifying the *tone* of text toward a demographic group (e.g., "positive," "negative," or "neutral" regard).
- It's used to assess whether certain demographics consistently receive negative or disrespectful language.

QUESTION 6: [1 mark]
Which of the following steps compose the approach for improving response safety via in-context learning?

a. Retrieving safety demonstrations *similar* to the user query.

b. Fine-tuning the model with additional labeled data after generation.
c. Providing retrieved demonstrations as examples in the prompt to guide the model's response generation.
d. Sampling multiple outputs from LLMs and choosing the majority opinion.

Correct Answer: a, c

Explanation:

- One strategy for safe or polite generation with large language models is to retrieve "safety demonstrations" from a database of safe examples. Then you include these examples in the prompt to the LLM, showing it how to respond safely.

- Fine-tuning (b) is a different technique, not part of the described in-context learning approach.

- Majority vote (d) is also not typically a method described under "improving response safety via in-context learning."

QUESTION 7: [1 mark]

Which statement(s) is/are correct about how high-resource (HRL) vs. low-resource languages (LRL) affect model training?

a. LRLs typically have higher performance metrics due to smaller population sizes.
b. HRLs get more data, so the model might overfit to HRL cultural perspectives.
c. LRLs are often under-represented, leading to potential underestimation of their cultural nuances.
d. The dominance of HRLs can cause a reinforcing cycle that perpetuates imbalance.

Correct Answers: b, c, d

Explanation:

- (b) **True:** If the model sees far more data in certain HRLs, it might be overly biased or "overfit" to those languages' norms and perspectives.
- (c) **True:** LRLs often lack extensive corpora, so the model learns fewer details about these languages, risking lower performance and cultural misrepresentations.
- (d) **True:** The more a model focuses on HRLs, the more beneficial it appears to be for those languages, attracting further data, thus perpetuating imbalance.
- (a) is not correct: LRLs typically have lower performance metrics due to insufficient training data, not higher.

QUESTION 8: [1 mark]

The "Responsible LLM" concept is stated to address:

a. Only the bias in LLMs
b. A set of concerns including explainability, fairness, robustness, and security
c. Balancing training costs with carbon footprint
d. Implementation of purely rule-based safety filters

Correct Answer: b

Explanation:

- **Responsible LLM** research focuses on a broad range of ethical, social, and technical concerns:
    - Fairness & bias mitigation
    - Explainability & transparency
    - Robustness to adversarial inputs
    - Security & safe deployment

QUESTION 9: [1 mark]

Within the StereoSet framework, the ***icat*** metric specifically refers to:

a. The ratio of anti-stereotypical associations to neutral associations
b. The percentage of times a model refuses to generate content deemed hateful
c. A measure of domain coverage across different demographic groups
d. A balanced metric capturing both a model's language modelling ability and the tendency to avoid stereotypical bias

Correct Answer: d

Explanation:

- In the **StereoSet** framework, *icat* is designed to measure how well the model balances contextual accuracy (i.e., good language modelling) and reduced stereotyping.
- It's a combined metric that looks at correctness in typical language modelling tasks while also penalizing stereotypical responses.

QUESTION 10: [1 mark]

Bias due to improper selection of training data typically arises in LLMs when:

a. Data are selected exclusively from curated, balanced sources with equal representation
b. The language model sees only real-time social media feeds without any historical texts
c. The training corpus over-represents some topics or groups, creating a skewed distribution

d. All data are automatically filtered to remove any demographic markers

Correct Answer: c

Explanation:

- **Improper data selection** leads to over-representation of certain domains, topics, or demographic groups, causing the learned model to be skewed.

- Balanced data curation and filtering are actually methods to *reduce* bias. If data come only from certain communities or perspectives, the model lacks balanced coverage, and biases surface.

Introduction to Large Language Models

Week-4 Assignment

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

Question 1:

A one-hot vector representation captures semantic similarity between related words like "king" and "queen".

a) True
b) False

Answer: b
Solution*:* One-hot vectors are orthogonal; no similarity is encoded.

Question 2:

Which method is used to reduce the dimensionality of a term-context matrix in count-based word representations?

a) Principal Component Analysis
b) Matrix Inversion
c) Singular Value Decomposition (SVD)
d) Latent Dirichlet Allocation

Answer: c
Solution*:* SVD is used to obtain low-dimensional representations in latent semantic analysis.

Question 3:

Which property makes tf-idf a better representation than raw term frequency?

a) It is non-linear
b) It accounts for the informativeness of words
c) It penalizes longer documents
d) It uses hierarchical clustering

Answer: c
Solution*:* IDF downweights common terms like “the” and emphasizes rare but important ones.

Question 4:

What is the purpose of using negative sampling in Word2Vec training?

a) To reduce dimensionality of word vectors
b) To ensure gradient convergence
c) To balance class distribution in classification
d) To simplify softmax computation

Answer: d
Solution*:* Negative sampling avoids computing softmax over the entire vocabulary.

Question 5:

In skip-gram Word2Vec, the model:

a) Predicts a word given its context
b) Predicts the next sentence
c) Predicts surrounding context words given a target word
d) Learns n-gram frequencies

Answer: c
Solution*:* Skip-gram learns by predicting surrounding words given a center word.

Question 6:

Why does SVD-based word embedding struggle with adding new words to the vocabulary?

a) It uses online learning
b) It lacks semantic interpretability
c) It assumes word order
d) It is computationally expensive to retrain

Answer: d
Solution*:* New words require recomputing the entire decomposition.

Question 7:

Which of the following best describes the term “distributional hypothesis” in NLP?

a) Words with high frequency have greater meaning
b) Words are defined by their part-of-speech tags
c) A word’s meaning is characterized by the words around it
d) Words should be normalized before vectorization

Answer: c

Question 8:

In Word2Vec, similarity between word vectors is computed using Euclidean distance.

a) True
b) False

Answer: b
Solution*:* Similarity is computed using dot product or cosine similarity.

Question 9:

Which method solves the problem of OOV (Out-Of-Vocabulary) words better?

a) One-hot encoding
b) CBOW
c) Skip-gram with subsampling
d) FastText embedding

Answer: d
Solution*:* FastText builds embeddings using character n-grams and handles unseen words.

Question 10:

If the word "economy" occurs 4 times in a corpus, and "growth" appears in a window of 5 words around it 3 times, what is the entry for (economy, growth) in a term-context matrix?

a) 1
b) 2
c) 3
d) 4

Answer: c
Solution*:* It counts co-occurrences in the window — here, 3 times.

Introduction to Large Language Models
Assignment- 9

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

QUESTION 1: [1 mark]

In the knowledge-graph training pipeline that models P(o | s, r) with a softmax over all entities, what practical difficulty motivates the use of negative sampling?

a. The softmax is undefined for KG scores.
b. The denominator sums over all entities, which is computationally expensive.
c. The numerator requires the full adjacency list for each relation.
d. The scores must be normalized per relation rather than globally.

Correct Answer: b

Explanation:

- The softmax probability for predicting the object o given a subject s and relation r is defined as $P(o \mid s, r) = \frac{\exp(f(s,r,o))}{\sum_{o'} \exp(f(s,r,o'))}$.
- The **denominator** requires summing the exponentiated scores $exp(f(s, r, o'))$ over *all possible entities* o' in the entire knowledge graph.
- Knowledge graphs can contain millions of entities, making this summation computationally intractable or extremely **expensive** during training.
- **Negative sampling** is introduced as a technique to approximate this denominator by sampling a small subset of "negative" entities, rather than summing over all of them, thereby making the computation feasible.

Why not the others?

- (a) Softmax is well-defined as long as the scores $f(s, r, o)$ are finite real numbers.
- (c) The numerator only requires the score for the specific triple (s, r, o).
- (d) While normalization might be discussed in specific models, the core issue motivating negative sampling for this general softmax setup is the cost of the full denominator summation.

QUESTION 2: [1 mark]

Which statements correctly characterize the local closed-world assumption in KG training with negative sampling?

a. Any unobserved triple is treated as false for training purposes.
b. It is strictly correct because KGs are exhaustive.
c. It helps training but may mislabel genuinely missing positives as negatives.

d. It eliminates the need for development/test splits.

Correct Answer: a, c

Explanation:

- **(a) True:** Negative sampling works by taking a known positive fact (s, r, o) and corrupting it (e.g., replacing o with a randomly sampled entity o') to create a triple (s, r, o'). The training process then *assumes* this newly generated, unobserved triple is false (negative).
- **(c) True:** This assumption is made for practical training purposes. However, real-world knowledge graphs are known to be highly **incomplete**. Therefore, it's possible that a randomly generated triple (s, r, o') is actually a true fact that just happens to be missing from the KG. Treating it as negative during training is technically incorrect in such cases, but it's a necessary approximation.
- **(b) False:** Knowledge graphs are far from exhaustive or complete. The assumption is made *despite* this incompleteness.

(d) False: The local closed-world assumption is a *training* assumption. It does not replace the need for separate development and test sets to evaluate the model's generalization performance on unseen data.

QUESTION 3: [1 mark]

For discriminative training, why is it infeasible to enforce all constraints f(s, r, o) ≥ m + f(s′,r,o′) over every possible negative triple?

a. The number of possible facts is $O(E^2R)$, overwhelmingly larger than positives.
b. Because scores cannot be compared across relations.
c. Because margins must be tuned per entity.
d. Because negatives are always ambiguous.

Correct answer: a

Explanation:

- Discriminative training aims to ensure that the score of a true positive fact $f(s, r, o)$ is higher than the score of a negative fact $f(s', r', o')$ by at least a margin m.
- If a knowledge graph has E entities and R relations, the total number of possible triples (potential facts) is $E \times R \times E = E^2R$.
- Typically, only a very small fraction of these possible triples are actual true facts present in the KG.
- Trying to enforce the margin constraint against *every* possible negative triple would involve computing a loss term for nearly E^2R triples for *each* positive triple. This number is astronomically large for typical KGs, making it computationally **infeasible**.

This is why negative sampling is also used in discriminative training – to consider only a small subset of potential negatives for each positive.

QUESTION 4: [1 mark]

Which statement best describes score polarity in KG models?

a. Scores must always be larger for false triples.
b. Score polarity is fixed by the dataset.
c. Some models use higher scores for more plausible triples; others use lower, and probabilities/losses can be adapted accordingly.
d. Polarity only matters for RotatE.

Correct Answer: c

Explanation:

- KG completion models use a scoring function $f(s, r, o)$ to assign a real-valued score indicating the plausibility of a triple.
- Some models are designed such that a **higher** score means the triple is more likely to be true (e.g., models trained with softmax loss where the score is in the exponent).
- Other models, like TransE, use a distance-based score (e.g., ||s+r-o||) where a **lower** score indicates higher plausibility.
- The choice of polarity depends on the model design. The loss function (e.g., softmax cross-entropy vs. margin-based hinge loss) can typically be adjusted to work with either polarity.

QUESTION 5: [1 mark]

Compared to semantic interpretation (logical-form execution), a differentiable KGQA system:

a. Requires a hand-coded logical form for every question.
b. Cannot be trained end-to-end.
c. Provides complete interpretability of reasoning steps.
d. Learns dense question and graph embeddings and uses cross-attention to align them.

Correct Answer: d

Explanation:

- **Semantic Interpretation:** This approach first translates the natural language question into a formal, structured query. This query is then executed against the KG. It is generally more interpretable but harder to train end-to-end.
- **Differentiable KG QA:** This approach avoids creating an explicit logical form. Instead, it learns dense vector representations (embeddings) for both the question (e.g., using BERT) and the graph elements (e.g., using GCNs). **Cross-attention** mechanisms are then used to find correspondences between the question embedding and the graph embeddings to identify the relevant subgraph or answer entity. This allows for end-to-end training but is less interpretable. Option (d) accurately describes this differentiable approach.

QUESTION 6: [1 mark]

Which statements correctly describe filtered evaluation?

a. It removes candidates that are **true facts in train/dev** from the ranked list before scoring the test query.
b. It **increases fairness** by not penalizing the model for ranking another correct answer that happened to be in training data.
c. It always decreases MRR.
d. It affects measures like MRR and MAP.

Correct Answer: a, b, d

Explanation:

- **(a) True:** During evaluation for a query like $(s, r, ?)$, the model produces a ranked list of candidate objects $(o_1, o_2, \ldots)$. Filtered evaluation involves checking if any highly ranked candidates o_i are *already known* to be true answers for $(s, r, ?)$ in the training or development sets. If so, these known answers are removed from the list *before* calculating the rank of the actual *test* answer(s).
- **(b) True:** Without filtering, if the model correctly ranks a known training fact o_2 higher than the test fact, say o_6 , the rank of o_6 would be penalized (rank 6). Filtering removes o_2, potentially improving the rank of o_6 (e.g., to rank 5). This is considered **fairer** because the model shouldn't be penalized for retrieving other valid answers that simply weren't part of the test set.
- **(d) True:** By changing the effective rank of the correct test answer(s), filtered evaluation directly impacts rank-based metrics like Mean Reciprocal Rank (MRR) and Mean Average Precision (MAP).

(c) False: Filtering typically *increases* (or keeps the same) metrics like MRR and Hits@K because it removes "competing" correct answers that might have ranked higher than the target test answer, thus improving the target answer's effective rank.

QUESTION 7: [1 mark]

Which of the following best captures the motivation for KG completion?

a. KGs are complete; KG completion mainly compresses them.
b. Manual curation keeps KGs fully up-to-date.
c. KGs are useful but incomplete, so we learn embeddings and a scoring function to infer missing facts.
d. KG completion is only for alignment across languages.

Correct answer: c

Explanation:

- Knowledge graphs (KGs) are valuable resources for many applications like QA and search.
- However, they suffer from significant **incompleteness**; many true facts are missing. It's impossible to manually curate all human knowledge or keep it perfectly current.
- Therefore, **KG completion** is a crucial task. It aims to automatically **infer missing facts** by learning patterns from the existing KG structure, often by training **embeddings** for entities and relations and defining a **scoring function** to predict the likelihood of potential triples.

QUESTION 8: [1 mark]

Consider pairwise hinge/ReLU loss for discriminative training with margin m: max{ 0, m + f(s'$_k$, r, o'$_k$) − f(s, r, o) }. When does this loss become exactly zero for a given negative (s'$_k$, r, o'$_k$)?

a. When f(s, r, o) ≥ m + f(s'$_k$, r, o'$_k$)
b. When f(s, r, o) = f(s'$_k$, r, o'$_k$)
c. When f(s'$_k$, r, o'$_k$) ≥ m + f(s, r, o)
d. Only when m = 0

Correct answer: a

Explanation:

- The hinge/ReLU loss is defined as $max\{0, argument\}$.
- The loss is zero if and only if the $argument$ is less than or equal to zero.
- In this case, the argument is $m + f(s'_k, r, o'_k) - f(s, r, o)$.
- So, the loss is zero when: $m + f(s'_k, r, o'_k) - f(s, r, o) \leq 0$.
- Rearranging this inequality gives: $f(s, r, o) \geq m + f(s'_k, r, o'_k)$.

  This matches the desired condition for discriminative training: the score of the positive triple $f(s, r, o)$ should be greater than the score of the negative triple $f(s'_k, r, o'_k)$ by at least the margin m.

QUESTION 9: [1 mark]

Uniform negative sampling can introduce an extra bias unless you do which of the following when forming the sampled denominator?

a. Exclude the true object o from the denominator.
b. Normalize scores per relation type.
c. Sample only from entities not connected to s.
d. Always include the true object o in the denominator.

Correct Answer: d

Explanation:

- When approximating the softmax denominator $P(o\,|s,r) = \frac{\exp\,(f(s,r,o))}{\sum_{o'} \exp\,(f(s,r,o'))}$ using negative sampling, we replace the full sum $\sum_{o'} \exp\,(f(s,r,o'))$ with a scaled sum over K sampled negatives $\frac{E}{K} \sum_{o'' \in K} \exp\,(f(s,r,o''))$.
- However, this sampled sum doesn't necessarily include the *true object* o.

To get a better-behaved estimate (reducing potential bias in the denominator), the lecture notes explicitly state that one **must include o into the denominator by force**. This ensures the true positive score always contributes to the normalization term.

QUESTION 10: [1 mark]

Which of the following is the RotatE scoring function?

a. f(s, r, o) = ‖s+r−o‖²
b. f(s, r, o) = ‖s⊙r−o‖², where r lies on the unit circle element-wise
c. f(s, r, o) = $s^T R_r o$ with R_r orthonormal
d. f(s, r, o) = −⟨s, r, o⟩

Correct answer: b

Explanation:

- The RotatE model represents entities (s, o) and relations (r) as vectors in complex space C^D.
- Crucially, the relation embeddings r are constrained such that each element r_d has a magnitude of 1 (i.e., $|r_d| = 1$), representing a rotation in that complex dimension.
- The scoring function measures the squared distance between the rotated subject (s ⊙ r, where ⊙is element-wise complex multiplication) and the object (o): f(s, r, o) = ||s ⊙r - o||^2.
- This exactly matches option (b).
- Option (a) is the scoring function for TransE (often used without the square, just the norm).
- Options (c) and (d) represent other families of KG embedding models (e.g., bilinear models or factorization models like DistMult/ComplEx).

Introduction to Large Language Models

Week-1 Assignment

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

Question 1:

Which of the following best demonstrates the principle of distributional semantics?

a. Words that co-occur frequently tend to share semantic properties.
b. Each word has a unique, fixed meaning regardless of context.
c. Syntax determines the entire meaning of a sentence.
d. Distributional semantics is unrelated to word embeddings.

Correct Answer: a

Question 2:

Which of the following words is **least likely** to be polysemous?

a. Bank
b. Tree
c. Gravity
d. Idea

Correct Answer: c

Question 3:

Consider the following sentence pair:

Sentence 1: Riya dropped the glass.
Sentence 2: The glass broke.

Does Sentence 1 entail Sentence 2?

a. Yes
b. No

Correct Answer: b
Solution: Entailment is not guaranteed – the glass might not have broken.

QUESTION 4:

Which sentence contains a **homonym**?

a. He wound the clock before bed.
b. She tied her hair in a bun.
c. I can't bear the noise.
d. He likes to bat after lunch.

Correct Answer: d
Solution: "Bat" – sports equipment or animal, depending on usage.

Question 5:

Which of the following relationships are **incorrectly labeled**?

a. Car is a meronym of wheel.
b. Rose is a hyponym of flower.
c. Keyboard is a holonym of key.
d. Tree is a hypernym of oak.

Correct Answer: a
Solution: Wheel is a meronym of car, not the other way around.

Question 6:

_____ studies how context influences the interpretation of meaning.

a. Syntax
b. Morphology
c. Pragmatics
d. Semantics

Correct Answer: c

Question 7:

In the sentence, "After Sita praised Radha, she smiled shyly," who does "she" most likely refer to?

a. Sita
b. Radha
c. Ambiguous
d. Neither

Correct Answer: c

Question 8:

Which of the following statements is true?

(i) Word embeddings capture semantic similarity through context.
(ii) Morphological analysis is irrelevant in LLMs.
(iii) Hypernyms are more specific than hyponyms.

a. Only (i)
b. Only (i) and (iii)
c. Only (ii) and (iii)
d. All of the above

Correct Answer: a

QUESTION 9:

What issues can be observed in the following text?

On a much-needed #workcation in beautiful Goa. Workin & chillin by d waves!

a. Idioms

b. Non-standard English
c. Tricky Entity Names
d. Neologisms
Correct Answer: b,d

QUESTION 10:
In semantic role labelling, we determine the semantic role of each argument with respect to the
_____ of the sentence.
a. noun phrase
b. subject
c. predicate
d. adjunct
Correct Answer: c

Introduction to Large Language Models

Week-6 Assignment

Number of questions: 8 **Total mark: 6 X 1 + 2 X 2 = 10**

Question 1: [1 mark]

True or False:
RoPE uses additive embeddings like sinusoidal encoding.

Answer: False
Solution: Please refer to slides.

Question 2: [1 mark]

Which of the following is true about *multi-head attention*?

a. It increases model interpretability by using a single set of attention weights
b. Each head operates on different parts of the input in parallel
c. It reduces the number of parameters in the model
d. Heads are averaged before applying the softmax function

Answer: b
Solution: Each attention head processes different learned projections of the input, enabling the model to capture different features.

Question 3: [1 mark]

What is the role of the residual connection in the Transformer architecture?

a. Improve gradient flow during backpropagation

b. Normalize input embeddings

c. Reduce computational complexity

d. Prevent overfitting

Answer: a

Solution: Please refer to lecture slides.

Question 4: [1 mark]

True or False:
The feedforward network in a Transformer block introduces non-linearity between attention layers.

Answer: True

Solution: Please refer to lecture slides.

Question 5: [1 mark]

Fill in the blank:
The sinusoidal positional encoding uses sine for even dimensions and ___ for odd dimensions.

a. sine
b. cosine
c. tangent
d. None of these

Answer: b

Solution: Please refer to lecture slides.

Question 6: [1 mark]

Why is positional encoding added to input embeddings in Transformers?
a. To provide unique values for each word
b. To indicate the position of tokens since Transformers are non-sequential
c. To scale embeddings
d. To avoid vanishing gradients

Answer: b

Solution: Please refer to lecture slides.

Question 7: [2 marks]

You are given a self-attention layer with input dimension 512, using 8 heads. What is the output dimension per head?

a. 64
b. 128

c. 32
d. 256

Answer: a
Solution: Each head processes 512/8 = 64 dimensions

QUESTION 8: [2 marks]
For a transformer with d_{model} = 512, calculate the positional encoding for position p=14 and dimensions 6 and 7 using the sinusoidal formula:

$$PE(p,2i) = sin\left(\frac{p}{10000^{2i/d_{model}}}\right) \quad PE(p,2i+1) = cos\left(\frac{p}{10000^{2i/d_{model}}}\right)$$

a. $sin\left(\frac{14}{10000^{3/256}}\right), cos\left(\frac{14}{10000^{3/256}}\right)$
b. $cos\left(\frac{14}{10000^{6/256}}\right), sin\left(\frac{14}{10000^{7/256}}\right)$
c. $cos\left(\frac{14}{10000^{3/256}}\right), sin\left(\frac{14}{10000^{3/256}}\right)$
d. $sin\left(\frac{14}{10000^{3/512}}\right), cos\left(\frac{14}{10000^{3/256}}\right)$

Correct Answer: a

Solution:

For dimension 6, $PE(14,6) = sin\left(\frac{14}{10000^{6/512}}\right) = sin\left(\frac{14}{10000^{3/256}}\right)$

For dimension 7, $PE(14,7) = cos\left(\frac{14}{10000^{6/512}}\right) = cos\left(\frac{14}{10000^{3/256}}\right)$ _____
_____

Introduction to Large Language Models
Assignment- 11

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

QUESTION 1: [1 mark]

Assume that you build a document–term matrix M (rows: documents; columns: words) and take its thin SVD M = U Σ V^T. Which statement is most accurate for interpreting V in classical Latent Semantic Analysis (LSA)?

a. Columns of V (and rows of V^T) give low-dimensional word representations that capture co-occurrence similarity.
b. V gives only document embeddings; words are in U.
c. V and U are not orthonormal in LSA.
d. Σ can be ignored without affecting similarity.

Correct Answer: a

Explanation:

- **Latent Semantic Analysis (LSA)** uses Singular Value Decomposition (SVD) to factorize a document-term matrix M. The factorization is $\mathrm{M} = \mathrm{U}\,\Sigma\,\mathrm{V}^{\mathrm{T}}$.
- In this decomposition:
  - M is the $d \times w$ matrix (documents × words).
  - U is the $d \times k$ matrix whose rows represent documents in a lower-dimensional latent space.
  - Σ is a $k \times k$ diagonal matrix of singular values.
  - V^{T} is the $k \times w$ matrix whose *columns* represent words in the latent space. Equivalently, the *rows* of V^{T} (or columns of V are the word embeddings.
- The dot product between word vectors derived from V reflects their co-occurrence patterns across documents, capturing semantic similarity.
- Therefore, the columns of V (or rows of V^{T}) provide the low-dimensional word representations.

Why not the others?

- (b) is incorrect; V relates to words/terms, while U relates to documents.
- (c) is incorrect; the columns of U and V are orthonormal by the definition of SVD.
- (d) is incorrect; Σ contains singular values that scale the dimensions and are important for reconstructing the original matrix and weighting the importance of latent dimensions.

QUESTION 2: [1 mark]

Which statements correctly characterize the basic DistMult approach for knowledge graph completion?

a. Each relation r is parameterized by a full D×D matrix that can capture asymmetric relations.
b. The relation embedding is a diagonal matrix, leading to a multiplicative interaction of entity embeddings.
c. DistMult struggles with non-symmetric relations because score(s, r, o) = $a_s^T M_r a_o$ is inherently symmetric in s and o.
d. DistMult's performance is typically tested only on fully symmetric KGs.

Correct Answer: b, c

Explanation:

- **DistMult** is a tensor factorization model for knowledge graphs. It simplifies the general relation matrix M_r by constraining it to be **diagonal**. This leads to a score calculated as $f(s,r,o) = \sum_d s_d r_d o_d$, often written as <s,r,o>. This represents a **multiplicative interaction** between the corresponding elements of the subject, relation, and object embeddings. Statement (b) is correct.
- Because the score is calculated this way using a diagonal M_r, swapping s and o results in the same score: $\sum_d s_d r_d o_d = \sum_d o_d r_d s_d$. This means DistMult **cannot model asymmetric or anti-symmetric relations** effectively. Statement (c) is correct.
- Statement (a) is incorrect; DistMult uses diagonal matrices, not full matrices, for relations.
- Statement (d) is incorrect; while DistMult performs well on some benchmarks like FB15k and WN18, these datasets do contain non-symmetric relations. Its surprisingly good performance is partly attributed to the relative scarcity of asymmetric test queries where symmetry would be a major issue.

QUESTION 3: [1 mark]

Given a doc–term matrix M, what do M^TM and MM^T capture?

a. M^TM: word–word co-occurrence similarity across documents
b. MM^T: document–document similarity via shared terms
c. Both are identity matrices by construction
d. M^TM counts how often a word appears in the corpus total

Correct Answer: a, b

Explanation:

- Let M be a matrix where rows are documents (d) and columns are words (w).
- M^TM: The transpose M^T has words as rows and documents as columns. The element (i, j) of the product M^TM is the dot product of column i of M^T (word i's presence across documents) and column j of M (word j's presence across documents). This dot product effectively counts or measures the **co-occurrence of word i and word j across all documents**. This reflects word similarity. (a) is correct.
- MM^T: The element (i, j) of the product MM^T is the dot product of row i of M (document i's word content) and row j of M^T (document j's word content). This dot

product measures the **similarity between document i and document j based on the terms they share**. (b) is correct.
- (c) is incorrect; these products are generally not identity matrices unless M itself is orthonormal in a specific way, which is not true for typical doc-term matrices.
- (d) is incorrect; the diagonal elements of $\mathrm{M^TM}$ relate to how often a word appears across documents, but the off-diagonal elements capture co-occurrence, not just total counts.

QUESTION 4: [1 mark]

Which best describes the main advantage of using a factorized representation (e.g., DistMult, ComplEx) for large KGs?

a. It enforces that every relation in the KG be perfectly symmetric.
b. It ensures each entity is stored as a one-hot vector, simplifying nearest-neighbour queries.
c. It collapses the entire KG into a single scalar value.
d. It significantly reduces parameters and enables generalization to unseen triples by capturing low-rank structure.

Correct Answer: d

Explanation:

- A knowledge graph can be represented as a large, sparse 3D tensor X (subject × relation × object).
- Storing this tensor explicitly is inefficient. Factorization models assume this tensor has **low-rank structure**.
- Models like DistMult or ComplEx decompose this tensor into lower-dimensional embeddings for entities and relations.
- This **significantly reduces the number of parameters** needed compared to storing the full tensor. Instead of E ×R × E parameters, we store embeddings (e.g., E × D for entities, R × D for relations in DistMult).
- By learning these compact embeddings from observed triples, the model captures underlying patterns (the low-rank structure) and can **generalize** to predict the plausibility of **unseen triples** (KG completion).

Why not the others?

- (a) Is only true for specific models like DistMult, and it's a limitation, not the main advantage. ComplEx, for example, can handle asymmetry.
- (b) Factorization models learn *dense* embeddings, not one-hot vectors.
- (c) Factorization produces embeddings and a scoring function, not a single scalar.

QUESTION 5: [1 mark]

Which statement best describes the *reshaping* of a 3D KG tensor $X \in R^{|E|\times|R|\times|E|}$ into a matrix factorization problem?

a. One axis remains for subject, one axis remains for object, and relations are combined into a single expanded axis.
b. The subject dimension is repeated to match the relation dimension, resulting in a 2D matrix.
c. Each subject–relation pair is collapsed into a single dimension, while objects remain as separate entries.
d. The entire KG is vectorized into a 1D array and then factorized with an SVD approach.

Correct Answer: c

Explanation:

- One way to conceptualize the problem for matrix factorization is to reshape the tensor differently, and a common reshaping approach treats the task as predicting the object given a (subject, relation) pair.
- In this view, you can create a matrix where each row corresponds to a unique **(subject, relation) pair**, and each column corresponds to an **entity (potential object)**. The entries in this matrix would indicate the existence or score of the triple (s, r, o).
- This results in an $(|E| \times |R|) \times |E|$ matrix. Matrix factorization techniques can then be applied to this large, sparse matrix to find latent representations for (s, r) pairs and objects. Option (c) describes this reshaping where the (subject, relation) dimensions are effectively combined.

QUESTION 6: [1 mark]

SimplE addresses asymmetry by:

a. Using separate subject and object embeddings per entity and including inverse relations, with an averaged score over the two directions
b. Constraining relation vectors to unit modulus
c. Replacing dot-products by max-pooling
d. Removing inverse relations entirely

Correct Answer: a

Explanation:

- **SimplE** is designed to handle asymmetric relations better than models like DistMult.
- Its key idea is to learn **two different embedding vectors** for each entity e: one for when it acts as a subject (sub(e)) and one for when it acts as an object (obj(e)).
- It also explicitly introduces **inverse relations** (r^{-1}) for every relation r.
- The final score for a triple (s, r, o) is calculated as the **average** of the DistMult-style score in the forward direction <sub(s), rel(r), obj(o)> and the score in the inverse direction <sub(o), rel(r^{-1}), obj(s)>.
- This structure allows the model to assign different scores to (s, r, o) and (o, r, s), thus capturing asymmetry.

QUESTION 7: [1 mark]

Which of the following statements correctly describe hyperbolic (Poincare) embeddings for hierarchical data?

a. They map nodes onto a disk (or ball) such that large branching factors can be represented with lower distortion than in Euclidean space.
b. Distance grows slowly near the center and becomes infinite near the boundary, making it naturally suited for tree-like structures.
c. They require each node to be embedded on the surface of the Poincare disk of radius 1.
d. They can achieve arbitrarily low distortion embeddings for trees with the same dimension as Euclidean space.

Correct Answers: a, b

Explanation:

- **Hyperbolic spaces** (like the Poincaré disk model) have geometric properties different from Euclidean space. Specifically, the "volume" grows exponentially with radius, similar to how nodes in a tree multiply exponentially with depth.
- This property allows tree-like structures (hierarchies) to be embedded with much **lower distortion** compared to Euclidean space of the same dimension. Large branching factors can fit without "crowding". (a) is correct.
- In the Poincaré disk model, points are mapped inside a unit disk. The hyperbolic distance $d_H(x,y)$ between points increases rapidly as points approach the boundary (perimeter) of the disk, becoming infinite at the boundary itself. This naturally places root nodes near the center and leaf nodes near the periphery, mirroring tree structures. (b) is correct.
- (c) is incorrect; points are embedded *strictly inside* the unit disk, not on the surface.
- (d) is incorrect; while hyperbolic space achieves *lower* distortion for trees than Euclidean space of the same dimension, achieving arbitrarily low $(1+\epsilon)$ distortion might require increasing dimension or isn't guaranteed just by using hyperbolic space. Euclidean distortion for trees is much higher.

QUESTION 8: [1 mark]

Why might a partial-order-based approach (like order embeddings) be beneficial for modelling ‘is-a’ relationships compared to purely distance-based approaches?

a. They explicitly encode the ancestor–descendant relation as a coordinate-wise inequality or containment.
b. They can represent negative correlations (i.e., sibling vs. ancestor) more easily than distance metrics.
c. They inherently guarantee transitive closure of the hierarchy in the learned embedding space.

d. They do not rely on pairwise distances but use a notion of coordinate-wise ordering or interval containment.

Correct Answer: a, d

Explanation:

- Hierarchies represent a **partial order** (e.g., 'mammal' > 'dog', 'animal' > 'mammal').
- **Order embeddings** aim to directly model this partial order $\prec$ in the embedding space.
- Instead of just using distance, they define the relationship based on **coordinate-wise inequalities** (e.g., for cone embeddings, $x \prec y \Leftrightarrow u_x \geq u_y$ element-wise) or **region containment**. Statements (a) and (d) correctly describe this core idea.
- (b) is generally false; representing negative correlations (e.g., disjoint categories like 'fruit' vs. 'scientist') can actually be difficult for some order embeddings like cones. Box embeddings handle disjointness better. Distance-based methods can potentially model this via large distances.

(c) is not guaranteed; while the *goal* is to learn embeddings consistent with transitivity, the learned embeddings might still violate it depending on the training data and optimization.

QUESTION 9: [1 mark]

Which statement about box embeddings in hierarchical modelling is most accurate?

a. Each entity or type is assigned a single real-valued vector, ignoring bounding volumes.
b. Containment $I_x \subseteq I_y$ all dimensions encodes $x \prec y$.
c. They rely on spherical distances around a central node to measure tree depth.
d. They cannot be used to represent set intersections or partial overlap.

Correct Answer: b

Explanation:

- **Box embeddings** (or hyper-rectangle embeddings) represent each item x as a multi-dimensional box (or interval in each dimension), defined by its lower-left (b_x) and upper-right (h_x) corners .
- The hierarchical relationship $x \prec y$ (x is a descendant of y) is encoded by the geometric **containment** of the boxes: $x \prec y$ if and only if the box for x is fully contained within the box for y across all dimensions, i.e., $I_x[d] \subseteq I_y[d]$ for all d. Statement (b) is correct.
- (a) is incorrect; they use boxes (defined by min/max corners or center/offset), not single vectors.
- (c) describes hyperbolic embeddings more closely, not box embeddings.
- (d) is incorrect; the intersection of boxes is well-defined and can represent conjunctions or overlaps between concepts/types.

QUESTION 10: [1 mark]

For order embeddings with axis-aligned open cones:

a. Represent each item x by apex u_x; encode $x \prec y$ as $u_x \geq u_y$ (element-wise).
b. Positive loss encourages all dimensions to satisfy the order; negative loss enforces at least one dimension to violate it.
c. All cones (and their intersections) have the same measure in this construction.
d. This makes modeling negative correlation between sibling types difficult.

Correct Answer: a, b, c, d

Explanation:

All statements accurately describe properties or consequences of the open cone order embedding model presented:

- **(a) True:** Each item x is represented by the apex u_x of an infinite open cone extending towards $+\infty$ in all dimensions. The partial order $x \prec y$ is defined such that the cone for x must be contained within the cone for y, which translates to the element-wise inequality $u_x \geq u_y$
- **(b) True:** The loss functions are designed accordingly. For a positive example ($x \prec y$), the loss l_+ penalizes *any* dimension d where $u_y[d] > u_x[d]$, encouraging *all* dimensions to satisfy the constraint. For a negative example, the loss l_- requires *at least one* dimension d to violate the order condition (i.e., $u_y[d] > u_x[d]$ by some margin), becoming zero if *any* dimension satisfies this violation.
- **(c) True:** Unlike interval embeddings on a line where sub-intervals have smaller measures, these infinite cones (and their non-empty intersections) all have infinite volume (measure). This means the embedding doesn't inherently capture the notion that subtypes are "smaller" than supertypes.

(d) True: Because $x \prec y$ requires $u_x \geq u_y$ element-wise, it's hard to simultaneously model negative correlations. For instance, if 'fruit' and 'scientist' are both descendants of 'entity', their apex vectors would be element-wise greater than 'entity's apex. However, there's no easy way within this structure to enforce that the cones for 'fruit' and 'scientist' should be disjoint or negatively correlated, as they might overlap significantly in the embedding space.

Introduction to Large Language Models

Week-3 Assignment

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

Question 1:

In backpropagation, which method is used to compute the gradients?
a. Gradient descent
b. Chain rule of derivatives
c. Matrix factorization
d. Linear regression

Correct Answer: b
Solution: Backpropagation uses the chain rule of derivatives to calculate the gradients layer by layer.

Question 2:

Which of the following functions is **not differentiable at zero**?

a. Sigmoid
b. Tanh
c. ReLU
d. Linear

Correct Answer: c

Solution: ReLU is not differentiable at zero since the left and right limits of the derivative are not equal.

Question 3:

In the context of regularization, which of the following statements is true?

a. L2 regularization tends to produce sparse weights
b. Dropout is applied during inference to improve accuracy
c. L1 regularization adds the squared weight penalties to the loss function
d. Dropout prevents overfitting by randomly disabling neurons during training

Correct Answer: d
Solution: Dropout deactivates neurons randomly during training to prevent overfitting.

Question 4:

Which activation function is least likely to suffer from vanishing gradients?

a. Tanh
b. Sigmoid
c. ReLU

Correct Answer: c

Solution: Its gradient is 1 for positive input and 0 for negative input, so it allows gradients to flow effectively

---

Question 5:

Which of the following equations correctly represents the derivative of the sigmoid function?

a. $\sigma(x) \cdot (1 + \sigma(x))$
b. $\sigma(x)^2$
c. $\sigma(x) \cdot (1 - \sigma(x))$
d. 1 / (1 + e^x)

Correct Answer: c
Solution: The derivative of sigmoid $\sigma(x)$ is $\sigma(x)(1 - \sigma(x))$.

Question 6:

What condition must be met for the Perceptron learning algorithm to converge?

a. Learning rate must be zero
b. Data must be non-linearly separable
c. Data must be linearly separable
d. Activation function must be sigmoid

Correct Answer: c

Question 7:

Which of the following logic functions requires a network with at least one hidden layer to model?

a. AND
b. OR

c. NOT
d. XOR

Correct Answer: d
Solution: XOR is the classic example of a non-linearly separable function.

Question 8:

Why is it necessary to include non-linear activation functions between layers in an MLP?

a. Without them, the network is just a linear function
c. They prevent overfitting
d. They allow backpropagation to work

Correct Answer: a
Solution: Without non-linearity, stacking linear layers results in another linear function — limiting the model's expressiveness.

Question 9:

What is typically the output activation function for an MLP solving a binary classification task?

a. Tanh
b. ReLU
c. Sigmoid
d. Softmax

Correct Answer: c
Solution: For binary classification, the output is usually a single unit with a sigmoid activation.

Question 10:

Which type of regularization encourages sparsity in the weights?
a. L1 regularization
b. L2 regularization
c. Dropout
d. Early stopping

Correct Answer: a
Solution: L1 regularization encourages sparsity in the weights.

Introduction to Large Language Models
Assignment- 8

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

QUESTION 1: [1 mark]

In standard instruction tuning with a decoder-only LM, which tokens typically contribute to the next-token prediction loss?

a) Only the prompt tokens
b) Only the response tokens
c) Both prompt and response tokens
d) Neither; loss is computed at the sequence level only

Correct Answer: b

Explanation:

Instruction tuning trains a model to generate a desired response given an instruction (prompt). In a decoder-only model, the prompt and response are concatenated (e.g., [prompt_tokens] [response_tokens]). The model's objective is to predict the next token.

The loss (e.g., cross-entropy) is only calculated for the **response tokens**. The prompt tokens serve as the conditioning context, but we are not training the model to predict the prompt itself, only to predict the correct *answer* that should follow it.

QUESTION 2: [1 mark]

Why can using multiple instruction templates for the same task help?

a) It only increases the dataset size.
b) It regularizes the reward model.
c) It improves generalization by exposing the model to different phrasings of the instruction.
d) It ensures the same tokenization across tasks.

Correct Answer: c

Explanation:

The goal of instruction tuning is to teach the model to follow instructions in general, not just to memorize a few specific task formats.

By training on **multiple templates**—which are different phrasings of the same underlying instruction (e.g., "Based on the paragraph above, can we conclude..." vs. "Can we infer the following?") —the model is exposed to a wider variety of linguistic structures.

This "rephrasing... helps the model learn and generalize more effectively" to new, unseen instructions at test time.

 (a) is a side effect, but not the primary reason.

 (b) is incorrect; instruction tuning is separate from training a reward model.

 (d) is incorrect; different phrasings will likely have different tokenizations.

QUESTION 3: [1 mark]

As the model size grows, what happens to prompt length and initialization sensitivity in prompt tuning?

a) Both matter more.
b) Both matter less.
c) Length matters less but initialization matters more.
d) Initialization matters less but length matters more.

Correct Answer: b

Explanation:

The lecture slides present experimental results on "Prompt Tuning," which uses continuous/soft prompts.

1. **Prompt Length:** As the model size increases, the performance gap between different prompt lengths becomes very small and they converge to a similar high score .
2. **Initialization:** While smaller models are very sensitive to how the soft prompt is initialized (e.g., "Random Uniform" performs poorly), larger models achieve high performance regardless of the initialization method.

Therefore, as models scale, they become more robust, and both prompt length and initialization "matter less."

QUESTION 4: [1 mark]

Which of the following statement(s) is/are true about the POSIX metric for quantifying prompt sensitivity?

a) POSIX is independent of the correctness of the generated responses and captures sensitivity as a property independent of correctness
b) POSIX is a length-normalized metric
c) POSIX compares the generated responses against the ground-truth to quantify prompt sensitivity

d) POSIX captures the variance in the log-likelihood of the same response for different input prompt variations

Correct Answer: a, b, d

Explanation:

- **(a) True:** The POSIX metric measures the *stability* of a model's output probabilities in response to prompt variations, not whether those outputs are factually correct.
- **(b) True:** The POSIX formula explicitly includes the term $1/L_{y_j}$ for length normalization, to accommodate arbitrary response lengths.
- **(c) False:** The POSIX formula only compares the model's probabilities for its *own* generated responses (yj) given different prompts (xi and xj). It does not use a ground-truth label.
- **(d) True:** The core of the metric is the log of the likelihood ratio, which directly captures the relative-change in log-likelihood of a response yj when the prompt is changed from xj to an intent-aligned variant xi. This is a measure of variance in the model's confidence.

QUESTION 5: [1 mark]

Which statement is true about prompt sensitivity as captured by POSIX?

a) Larger models always have lower prompt sensitivity than smaller ones.
b) Larger models always have higher prompt sensitivity than smaller ones.
c) Prompt sensitivity decreases for models with a parameter count above a certain threshold.
d) Increasing parameter count does not necessarily reduce prompt sensitivity.

Correct Answer: d

Explanation:

The experimental results in the lecture slides demonstrate that the relationship between model size and sensitivity is not linear or guaranteed.

As we see, even in the case of Llama-2, a 13B model is not guaranteed to always have lesser prompt sensitivity than a 7B model. We can thus infer that an increase in parameter count does not necessarily decrease prompt sensitivity. This directly supports option (d).

QUESTION 6: [1 mark]

In training a reward model with pairwise preferences $(x,\ y^+, y^-)$, the Bradley-Terry style objective encourages:

a) Maximizing $r_\theta(x, y^-) - r_\theta(x, y^+)$
b) Minimizing the entropy of the policy
c) Maximizing $\log \sigma \left(r_\theta(x, y^+) - r_\theta(x, y^-)\right)$
d) Setting $r_\theta(x, y)$ equal to the log-probability under π_{ref}

Correct Answer: c

Explanation:

The goal of training a reward model (RM) is to teach it to assign a higher score (rθ) to the preferred response (y+) than to the rejected response (y−).

1. The **Bradley-Terry (BT) model** defines the probability that y+ is preferred as a function of the *difference* in their scores, passed through a sigmoid function (σ)
2. To train the RM, we use **Maximum Likelihood Estimation**, which aims to find the parameters θ that maximize the log-probability of the observed human preferences in our dataset.
3. This directly leads to the objective function: max ∑ log σ(r(x,y+)−r(x,y−)). This is exactly option (c).

Option (a) would do the opposite (prefer y−). Options (b) and (d) relate to the *policy optimization* phase, not the *reward model training* phase.

QUESTION 7: [1 mark]

Which of the following are recommended while performing REINFORCE-style policy optimization?

a) Use the log-derivative trick to obtain an unbiased gradient estimator.
b) Weight token-level log-probs by the advantage function to reduce variance.
c) Use importance weights and clip them when sampling from a fixed policy.
d) Avoid any clipping to preserve gradient magnitude.

Correct Answers: a, b, c

Explanation:

(a) **True:** The **log-derivative trick** is the core mathematical technique used to rewrite the policy gradient objective into an expectation, which allows us to approximate the gradient using samples from the policy .
(b) **True:** Standard REINFORCE has high variance. To reduce this, the gradient is weighted by the **advantage function** (At=Qt−Vt) instead of the full cumulative reward. This measures *how much better* an action was than the average, leading to more stable training .
(c) **True:** To improve sample efficiency, PPO (a REINFORCE-style algorithm) uses **importance weights** to allow for multiple gradient updates using

samples from an *old* policy . To ensure stability, these importance weights are **clipped** .

(d) **False:** This is incorrect. Clipping importance weights is a crucial part of PPO to prevent large, unstable gradient updates.

QUESTION 8: [1 mark]

Which method combines reward maximization and minimizing KL divergence?

a) REINFORCE
b) Monte Carlo Approximation
c) Proximal Policy Optimization
d) Constitutional AI

Correct Answer: c

Explanation:

Proximal Policy Optimization (PPO) is the specific algorithm used to optimize the combined objective . It uses a clipped surrogate objective that approximates this KL-constrained function, effectively balancing reward-seeking with policy stability.

 (a) REINFORCE is the general algorithm family, but PPO is the specific method that formally incorporates the KL constraint.

 (b) Monte Carlo Approximation is a *technique* used within these algorithms, not the method itself.

 (d) Constitutional AI is a method for *generating preference data* to train the reward model , not the policy optimization algorithm.

QUESTION 9: [1 mark]

Which of the following is the reason for performing alignment beyond instruction tuning in LLMs?

a) Instruction tuning guarantees safety on harmful queries.
b) Alignment can prevent outputs that a model might otherwise deem correct, but humans find unacceptable.
c) Alignment is only needed for small models.
d) Instruction tuning already optimizes a human preference model.

Correct Answer: b

Explanation:

Instruction tuning might produce both a helpful answer *and* a harmful one ("You should stop eating entirely..."). Instruction tuning *cannot* reliably prevent the harmful output.

Alignment (like RLHF) is the next step, which uses human preference data to fine-tune the model. Its specific purpose is to prevent certain outputs that the model assumes to be correct, but humans consider wrong (or harmful/unacceptable).

QUESTION 10: [1 mark]

Let π_θ be the probability of choosing token a_t in state s_t assigned by the current policy being optimized, π_k be that by the old/reference policy and $\epsilon > 0$ be the clip parameter. When the token-level advantage A_t is positive, PPO-CLIP maximizes which of the following expression at step t?

a) $\max\left(\frac{\pi_\theta}{\pi_k},\ 1-\epsilon\right) A_t$
b) $\max\left(\frac{\pi_k}{\pi_\theta},\ 1-\epsilon\right) A_t$
c) $\min\left(\frac{\pi_k}{\pi_\theta},\ 1+\epsilon\right) A_t$
d) $\min\left(\frac{\pi_\theta}{\pi_k},\ 1+\epsilon\right) A_t$

Correct Answer: d

Explanation:

The PPO-CLIP objective is designed to prevent the new policy (π_θ) from moving too far from the old policy (π_k) in a single update.

- When the advantage A_t is **positive**, it means the action a_t was *good*, and we want to *increase* its probability. This means we want to increase the ratio $r_t = \frac{\pi_\theta}{\pi_k}$.
- However, to ensure stability, we "clip" this increase. We don't want the ratio r_t to go higher than (1+ϵ).
- Therefore, the algorithm takes the **minimum** of the *actual* ratio (r_t) and the *clipped* ratio ((1+ϵ)). This clipped ratio is then multiplied by the positive advantage A_t.

The objective to maximize is: $\min\left(\frac{\pi_\theta(a_t|s_t)}{\pi_k(a_t|s_t)},\ 1+\epsilon\right) A_t\,(s_t, a_t)$. This corresponds exactly to option (d).

Introduction to Large Language Models
Assignment- 7

Number of questions: 7 **Total mark: 4 X 1 + 3 X 2 = 10**

QUESTION 1: [1 mark]

Why can a pre-trained BART model be fine-tuned directly for abstractive summarization?

a) Its encoder alone is sufficient.
b) It shares vocabulary with summarization datasets.
c) It uses a larger context window than BERT.
d) It already contains a generative decoder trained jointly during pre-training.

Correct Answer: d

Explanation:

- **Abstractive summarization** is a sequence-to-sequence (seq2seq) task that requires the model to *generate* new text (a summary) based on an input text (an article).
- **BART (Bidirectional and Auto-Regressive Transformer)** is explicitly designed as an **encoder-decoder** model. Its pre-training task involves corrupting an input (e.g., masking or deleting text) and training the model to reconstruct the original, clean text.
- This pre-training process jointly trains a **bidirectional encoder** (to understand the corrupted input) and an **autoregressive decoder** (to generate the clean output).
- Because BART already possesses this powerful, pre-trained generative decoder, it is perfectly suited for fine-tuning on other generative tasks like summarization, where it learns to map an article (via the encoder) to a summary (via the decoder).

Why not the others?

- (a) An encoder *alone* (like BERT) is not designed for text generation. It produces representations, which are typically used for classification or span-prediction tasks.
- (b) Vocabulary sharing is incidental and not the primary *architectural* reason BART is suitable for this task.
- (c) The context window size is a model hyperparameter and does not determine its ability to perform generative tasks.

QUESTION 2: [2 marks]

For pre-training of encoder-decoder models, which statement(s) is/are true?

a) The encoder attends bidirectionally to its whole input.
b) The decoder conditions on earlier decoder tokens and encoder outputs.
c) Unlabelled text is turned into a supervised task via a noising scheme.
d) Training relies on a next-sentence-prediction loss.

Correct Answer: a, b, c

Explanation:

- **(a) True:** The encoder part of an encoder-decoder model (like in BART or T5) is designed to be bidirectional. This allows it to build a rich representation of the input sequence by considering both left and right context for every token, which is crucial for understanding the "corrupted" input.
- **(b) True:** This describes the standard Transformer decoder mechanism. The decoder is **autoregressive**, so it uses a causal (masked) self-attention to look at previously generated tokens. It also uses **cross-attention** to look at the complete output from the encoder, allowing it to condition the generated sequence on the input sequence.
- **(c) True:** This is the core idea of pre-training models like BART and T5. We only have unlabelled text, so we create a "supervised" task. We apply a **noising function** (like masking, deleting, or permuting spans of text) to the input and train the model to "denoise" it, i.e., predict the original, clean text.
- **(d) False:** The Next Sentence Prediction (NSP) loss was a pre-training objective used for the original **BERT** (an encoder-only model) to help it understand sentence relationships. Encoder-decoder models like BART and T5 use denoising/span-corruption objectives, not NSP.

QUESTION 3: [2 marks]

Which attention mask(s) prevent(s) a token from looking at future positions?

a) Causal mask
b) Fully-visible mask
c) Prefix-LM mask
d) All of the above
e) None of the above

Correct Answer: a, c

Explanation:

- **(a) Causal mask:** This is the standard mask used in decoder-only models (like GPT). It's a triangular mask that ensures a token at position i can only attend to tokens at positions 1...i and *not* to any "future" tokens at positions i+1 ... N. This is essential for autoregressive generation.
- **(c) Prefix-LM mask:** This mask is a hybrid. It divides the sequence into a "prefix" (input) and a "suffix" (output). It allows **fully-visible** (bidirectional) attention over the prefix, but applies a **causal mask** to the suffix. Therefore, for any token in the suffix (the part being generated), it *is* prevented from looking at future positions within that suffix, just like a standard causal mask.
- **(b) Fully-visible mask:** This mask, used in encoders (like BERT), allows every token to attend to every other token in the sequence, including future ones. It does not prevent looking ahead.

QUESTION 4: [1 mark]

T5 experiments showed that clean and compact pre-training data can outperform a larger but noisier corpus primarily because:

A. Larger corpora overfit.
B**.** Noise forces the model to waste capacity on modelling irrelevant patterns.
C. Clean data has longer documents.
D. Compact data allows bigger batches.

Correct Answer: b

Explanation:

- The T5 paper introduced the "Colossal Clean Crawled Corpus" (C4), which was meticulously filtered to remove non-natural-language content like code, menus, "Lorem ipsum" placeholder text, and other web-page "noise."
- In experiments, the clean **C4** dataset produced better results on downstream tasks than the much larger **unfiltered** C4 dataset.
- **(B)** This is the correct reason. A model trained on noisy data must use a portion of its finite capacity (parameters) to learn patterns in the noise (e.g., how to predict JavaScript code or HTML tags). This "wasted capacity" is then unavailable for learning the useful patterns of natural language, leading to worse performance on downstream NLP tasks.
- **(A)** Overfitting is typically a concern for *small* datasets, not massive corpora.
- **(C) & (D)** The cleaning process and dataset size do not guarantee longer documents or bigger batch sizes; these are not the primary reasons for the performance difference.

QUESTION 5: [1 mark]

What makes sampling from an auto-regressive language model straightforward?

A. The model is deterministic.
B. The vocabulary is small.
C. Each conditional distribution over the vocabulary is readily normalised and can be sampled token-by-token.
D. Beam search guarantees optimality.

Correct Answer: c

Explanation:

- An auto-regressive language model, the probability of a text sequence X is a product of conditional probabilities: $P(X) = \prod P(x_i | x_1, \ldots, x_{i-1})$.
- **(C)** This property is what makes sampling possible. At each step i, the model takes the context (all previous tokens x1...xi−1) and produces a vector of logits (raw scores) for every token in the vocabulary. A **softmax** function is applied to these logits to create a complete, normalized probability distribution. Sampling is then the

straightforward process of picking one token from this distribution. This new token is appended to the context, and the process repeats.

- **(A)** The neural network's forward pass is deterministic, but the *sampling* process itself is stochastic (probabilistic) by definition.
- **(B)** The vocabulary is typically very large, not small.
- **(D)** Beam search is a decoding algorithm (a heuristic search to find a high-probability sequence), not a sampling method. It also does not guarantee finding the optimal (most probable) sequence.

QUESTION 6: [1 mark]

Why does **ELMo** build its input token representations from a **character-level CNN** instead of fixed word embeddings?

A. To reduce training time by sharing parameters
B. To avoid **UNK** tokens and generate representations for any string
C. To compress embeddings to 128 dimensions
D. To ensure the same vector for a word in every context

Correct Answer: b

Explanation:

- **(B)** The primary advantage of using a character-level CNN is to handle out-of-vocabulary (OOV) words. A model with fixed word embeddings (like word2vec) has a finite vocabulary; any word not in this vocabulary is mapped to a single "UNK" (unknown) token, losing all its meaning. By building representations from characters, ELMo can compose a unique vector for *any* word, including rare words, misspelled words, or new words, as long as it's made of known characters.
- **(A)** While CNNs do share parameters, this is a general property, not the specific reason for choosing them *over* fixed word embeddings in this context.
- **(C)** The lecture slides state the projection is to 512 dimensions, not 128.
- **(D)** This is the exact *problem* ELMo was designed to *solve*. ELMo's goal is to create *context-dependent* representations, whereas fixed embeddings (the alternative) *do* have the same vector for a word in every context.

QUESTION 7: (Numerical Question) [2 marks]

The **einsum** function in numpy is used as a generalized operation for performing tensor multiplications. Now, consider two matrices: $A = \begin{bmatrix} 2 & 8 \\ 4 & 3 \end{bmatrix}$ and $B = \begin{bmatrix} -9 & 9 \\ 0 & 11 \end{bmatrix}$. Then, what is the output of the following numpy operation?

numpy.einsum('ij,ij->', A, B)

Correct Answer: 87

Explanation:

The einsum notation 'ij,ij->' defines the operation:

1. **ij,ij**: The two inputs are both 2D matrices (indexed by i and j). Because the indices are identical for both inputs, this specifies an **element-wise multiplication** (also known as the Hadamard product).
2. **->**: The right side of the arrow is empty, which means the output should be a scalar (0-dimensional). This implies that we must sum over all indices that appear in the input but not the output (in this case, both i and j).

Therefore, the operation is an element-wise multiplication of A and B, followed by a sum of all the elements in the resulting matrix.

Step 1: Element-wise Multiplication (A⊙B)

$A \odot B = \begin{bmatrix} -18 & 72 \\ 0 & 33 \end{bmatrix}$

Step 2: Sum all elements

Sum = (−18) + 72 + 0 + 33 = 54+33 = 87

Introduction to Large Language Models

Week-2 Assignment

Number of questions: 8 **Total mark: 6 X 1 + 2 x 2 = 10**

Question 1:

Which of the following does **not** directly affect perplexity?

a. Vocabulary size
b. Sentence probability
c. Number of tokens
d. Sentence length

Answer: a

Question 2:

What is the goal of a probabilistic language model?

a. Translate sentences
b. Predict the next word in a sequence
c. Classify documents
d. Summarize text

Answer: b

Question 3:

Which equation expresses the chain rule for a 4-word sentence?
a. $P(w1, w2, w3, w4) = P(w1) + P(w2|w1) + P(w3|w2) + P(w4|w3)$
b. $P(w1, w2, w3, w4) = P(w1) \times P(w2|w1) \times P(w3|w1, w2) \times P(w4|w1, w2, w3)$
c. $P(w1, w2, w3, w4) = P(w1) \times P(w2|w1) \times P(w3|w2) \times P(w4|w3)$
d. $P(w1, w2, w3, w4) = P(w4|w3) \times P(w3|w2) \times P(w2|w1) \times P(w1)$

Answer: b

Question 4:

Which assumption allows n-gram models to reduce computation?

a. Bayes Assumption
b. Chain Rule
c. Independence Assumption
d. Markov Assumption

Answer: d

Question 5:

In a trigram language model, which of the following is a correct example of linear interpolation?

a. $P(w_i|w_{i-2},w_{i-1})=\lambda_1 P(w_i|w_{i-2},w_{i-1})$
b. $P(w_i|w_{i-2},w_{i-1})=\lambda_1 P(w_i|w_{i-2},w_{i-1})+\lambda_2 P(w_i|w_{i-1})+\lambda_3 P(w_i)$
c. $P(w_i|w_{i-2},w_{i-1})=\max(P(w_i|w_{i-2},w_{i-1}),P(w_i|w_{i-1}))$

d. $P(w_i|w_{i-2},w_{i-1})=P(w_i)P(w_{i-1})/P(w_{i-2})$

Answer: b

Question 5:
A trigram model is equivalent to which order Markov model?

a. 3
b. 2
c. 1
d. 4

Answer: b

Question 6:

Which smoothing technique leverages the number of unique contexts a word appears in?

a. Good-Turing
b. Add-k
c. Kneser-Ney
d. Absolute Discounting

Correct Answer: c
Explanation: Kneser-Ney uses continuation probability which counts the number of unique left contexts.

For Question 4 to 5, consider the following corpus:
<s> the sky is blue </s>
<s> birds fly in the sky </s>
<s> the blue birds sing </s>

QUESTION 4:
Assuming a bi-gram language model, calculate the probability of the sentence:
<s> birds fly in the blue sky </s>
Ignore the unigram probability of P(<s>) in your calculation.
a. 2/37
b. 1/27
c. 0
d. 1/36
Correct Answer: c
Solution:

P(<s> birds fly in the blue sky </s>) =
P(<s>) x P(birds | <s>) × P(fly | birds) × P(in | fly) × P(the | in) × P(blue | the) × P(sky | blue) × P(</s> | sky)

From the corpus:

P(birds | <s>) = Count(<s> birds) / Count(<s>) = 1 / 3

P(fly | birds) = Count(birds fly) / Count(birds) = 1 / 2

P(in | fly) = Count(fly in) / Count(fly) = 1 / 1

P(the | in) = Count(in the) / Count(in) = 1 / 1

P(blue | the) = Count(the blue) / Count(the) = 1 / 3

P(sky | blue) = Count(blue sky) = 0

P(<s> birds fly in the blue sky </s>) = 0

QUESTION 5:
Assuming a bi-gram language model, calculate the perplexity of the sentence:
<s> birds fly in the blue sky </s>

Please do not consider <s> and </s> as words of the sentence.
a. 271/4
b. 271/5
c. 91/6
d. None of these

Correct Answer: d
Solution:
As calculated in the previous question,
P(<s> birds fly in the blue sky </s>) = 0
Thus, Perplexity = undefined

Introduction to Large Language Models

Week-5 Assignment

Number of questions: 9 **Total mark: 8 X 1 + 1 X 2 = 10**

Question 1: [1 mark]

Which of the following best explains the vanishing gradient problem in RNNs?

a. RNNs lack memory mechanisms for long-term dependencies.
b. Gradients grow too large during backpropagation.
c. Gradients shrink exponentially over long sequences.
d. RNNs cannot process variable-length sequences.

Correct Answer: c
Solution: Please refer to slides.

Question 2: [1 mark]

In an attention mechanism, what does the softmax function ensure?

a. Normalization of decoder outputs
b. Stability of gradients during backpropagation
c. Values lie between -1 and 1
d. Attention weights sum to 1

Correct Answer: d
Solution:
The softmax is applied to attention scores to produce a probability distribution over encoder hidden states. This ensures the weights sum to 1.

Question 3: [1 mark]

Which of the following is true about the difference between a standard RNN and an LSTM?

a. LSTM does not use any non-linear activation.
b. LSTM has a gating mechanism to control information flow.
c. RNNs have fewer parameters than LSTMs because they use convolution.
d. LSTMs cannot learn long-term dependencies.

Correct Answer: b
Solution: Please refer to slides.

Question 4: [1 mark]

Which gate in an LSTM is responsible for deciding how much of the cell state to keep?

a. Forget gate
b. Input gate
c. Output gate
d. Cell candidate gate

Correct Answer: a

Solution:
The forget gate determines what fraction of the previous cell state should be retained in the current timestep.

Question 5: [1 mark]

What improvement does attention bring to the basic Seq2Seq model?

a. Reduces training time
b. Removes the need for an encoder
c. Allows access to all encoder states during decoding
d. Reduces the number of model parameters

Correct Answer: c
Solution:
Attention allows the decoder to consider all encoder hidden states dynamically.

Question 6: [1 mark]

Which of the following is a correct statement about the encoder-decoder architecture?

a. The encoder generates tokens one at a time.
b. The decoder summarizes the input sequence.
c. The decoder generates outputs based on encoder representations and its own prior outputs.
d. The encoder stores only the first token of the sequence.

Correct Answer: c
Solution:
The decoder uses both the encoder's output and its own previously generated tokens to produce the next output.

Question 7: [1 mark]

What is self-attention in Transformers used for?

a. To enable sequential computation
b. To attend to the previous layer's output
c. To relate different positions in the same sequence
d. To enforce fixed-length output

Correct Answer: c
Solution:
Self-attention allows each token to focus on all other tokens in the same sequence.

Question 8: [1 mark]

Why are RNNs preferred over fixed-window neural models?

a. They have a smaller parameter size.

b. They can process sequences of arbitrary length.

c. They eliminate the need for embedding layers.

d. None of the above.

Correct Answer: b

Solution: Please refer to lecture slides.

QUESTION 9: [2 marks]
Given the following encoder and decoder hidden states, compute the attention scores. (Use dot product as the scoring function)

Encoder hidden states: h1=[7,3], h2=[0,2], h3=[1,4]
Decoder hidden state: s=[0.2,1.5]

a. 0.42, 0.02, 0.56
b. 0.15, 0.53, 0.32
c. 0.64, 0.18, 0.18
d. 0.08, 0.91, 0.01

Correct Answer: a
Solution:
e1 = 7*0.2+3*1.5 = 5.9
e2 = 0*0.2+2*1.5 = 3
e3 = 1*0.2+4*1.5 = 6.2
$\alpha 1 = e^{5.9}/(e^{5.9} + e^{3} + e^{6.2}) = 0.42$

$\alpha 2 = e^{3}/(e^{5.9} + e^{3} + e^{6.2}) = 0.02$
$\alpha 3 = e^{6.2}/(e^{5.9} + e^{3} + e^{6.2}) = 0.56$

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.
