

Quizzes

Natural Language Processing (2025)

This document contains the quizzes for the lectures, along with correct solutions and brief explanations. The explanations were drafted using ChatGPT in January 2025 and then manually reviewed and corrected.

Unit 3

Lecture 3.1

1. Which of the models mentioned in the slide on Decoder-based language models has the largest context length?

- GPT

Incorrect. OpenAI's o3 model (release: 2025-01) has a context size of 200K tokens.

- Gemini

Correct. Gemini 2.0 (release: 2025-01) has a context size of 1M tokens.

- DeepSeek

Incorrect. DeepSeek R1 (release: 2025-01) has a context size of 128K tokens.

2. What are adapters not typically used for?

- To adapt a pretrained language model to a new computing hardware.

Correct. Adapters are designed to fine-tune models for new tasks or languages without retraining the entire model, not for adapting to different hardware.

- To adapt a pretrained language model to a new task.

Incorrect. Adapters are commonly used to fine-tune models for specific tasks efficiently.

- To adapt a pretrained language model to a new language.

Incorrect. Adapters can be employed to extend a model's capabilities to additional languages.

3. What speedup is quoted for the H200 architecture over the H100 architecture when it comes to inference with the Llama 2 70B model?

- 1.9

Correct. The H200 architecture offers a 1.9x speedup over the H100 for Llama 2 70B model inference.

- 1.6

Incorrect.

- 1.4

Incorrect.

4. What task is addressed in the SWE-bench benchmark?

- resolving GitHub issues

Correct.

- strategic workflow extrapolation

Incorrect.

- answering factual questions about Sweden

Incorrect.

5. Which were the three best-performing models on the Chatbot Arena leaderboard?

- Gemini, ChatGPT, DeepSeek

Correct. (As of the date when the slides were produced.)

- ChatGPT, DeepSeek, Step-2

Incorrect.

- DeepSeek, Llama 3, o1

Incorrect.

Lecture 3.2

1. How does Adam solve the zig-zagging problem?

- It keeps averages of past gradient directions.

Correct.

- It keeps averages of past gradient magnitudes.

Correct. While momentum-based optimisers keep track of past gradient magnitudes to set adaptive learning rates, the main source of the zig-zagging problem are gradient directions.

- It keeps averages of past gradient similarities.

Incorrect. Adam does not compute averages based on gradient similarities; it focuses on magnitudes and squared gradients.

2. What is the total norm as referred to in the context of gradient clipping?

- The norm of the vector containing all parameter norms.

Correct.

- The norm of the vector containing all parameters.

Incorrect. Gradient clipping concerns the gradients' norms, not the parameters' norms.

- The sum of the norms of all parameter vectors.

Incorrect. In gradient clipping, the total norm is computed over all gradient norms as if they were concatenated into a single vector.

3. In the learning rate scheduler example, what is the learning rate after 280B tokens?

- 0.006

Incorrect. This learning rate is too high for the given point in training.

- 0.0006

Incorrect.

- 0.00006

Correct.

4. Consider a case of batch accumulation over three micro-batches with sizes [800, 1000, 600] where the summed micro-batch losses are [960, 1300, 900]. What is the loss over the full accumulation?

- 1.23

Incorrect.

- 1.32

Correct. The total loss is calculated by dividing the sum of all micro-batch losses by the sum of all micro-batch sizes:

$$\frac{960 + 1300 + 900}{800 + 1000 + 600} = \frac{3160}{2400} \approx 1.3167$$

Rounding to two decimal places gives 1.32.

- 1.33

Incorrect.

5. To which of the following would we typically not apply weight decay?

- the weights of the layer norms

Correct. Weight decay is generally not applied to the weights of layer normalisation layers, as these parameters are crucial for maintaining the normalised distribution of activations.

- the weights of the linear layers

Incorrect. Applying weight decay to the weights of linear layers is a common regularisation practice to prevent overfitting.

- the weights of the attention layers

Incorrect. Weight decay is often applied to the weights within attention mechanisms to promote generalisation.

Lecture 3.3

1. Assuming a token/byte ratio of 0.75 for English text, how many tokens would we expect to be in the text-only version of the latest Common Crawl dump?

- 600 billion

Incorrect.

- 6 trillion

Correct. The text extracted from the latest Common Crawl dump (2024-51) comprises approximately 7.37 TiB. Converting terabytes to bytes and multiplying with 0.75 gives approximately 6 trillion tokens.

- 67 trillion

Incorrect.

2. Looking at the slide Impact of filtering, which set of filters is the second-best set?

- C4 Filters

Correct.

- FineWeb

Incorrect.

- Gopher

Incorrect.

3. Why is deduplication so important in dataset curation for pretraining LLMs?

- It prevents overfitting, enhances data diversity, and reduces computational costs.

Correct. Deduplication ensures that models do not memorise repeated data, promotes exposure to diverse information, and reduces unnecessary computational load.

- It ensures that the model memorises frequently occurring text patterns for better accuracy.

Incorrect. Memorising repeated patterns can lead to overfitting and reduced generalisation.

- It increases the total dataset size, allowing the model to train on more tokens.

Incorrect. Deduplication reduces the dataset size by removing redundant data, but this is beneficial for training efficiency and model performance.

4. What model is used in the educational quality classifier in FineWeb-EDU?

- linear regression

Correct. The educational quality classifier is a simple linear regression model built upon embeddings obtained via the Snowflake-arctic-embed architecture.

- Transformer

Incorrect. While Transformers are used to produce the *input* to the classifier (using the Snowflake-arctic-embed architecture), the classifier itself uses a linear regression model.

- recurrent neural network

Incorrect.

5. For the largest ablation model considered, how many points of average accuracy is FineWeb-EDU better than standard FineWeb?

- 2 points

Correct.

- 4 points

Incorrect.

- 6 points

Correct.

Lecture 3.4

1. What is the estimated computational cost for the smallest GPT-2 model and a dataset consisting of 2.5B tokens?

- 15,000,000,000

Incorrect.

- 310,000,000,000,000,000

Incorrect.

- 1,860,000,000,000,000,000

Correct. The computational cost C for training a language model can be estimated using the formula $C = 6 \times P \times T$, where P is the number of parameters and T is the number of tokens. For the smallest GPT-2 model with 124 million parameters and a dataset of 2.5 billion tokens:

$$C = 6 \times 124 \times 10^6 \times 2.5 \times 10^9 = 1.86 \times 10^{18} \text{ FLOPs}$$

2. In the slide “Performance improves smoothly with scale”, what does each point of the thick black line in the leftmost plot correspond to?

- the model with the lowest test loss for a specific compute budget

Correct. Each point on the thick black line represents a model that achieves the lowest test loss given a particular compute budget, illustrating the optimal trade-off between model size and training duration.

- the model with the lowest test loss for a specific number of parameters

Incorrect. The plot focuses on compute budgets rather than solely on the number of parameters.

- the model with the lowest test loss for a specific training set size

Incorrect. The plot is centered on compute budgets, not directly on training set sizes.

3. An 8x A100 system can do approximately 64 exaflops per 24 hours. How many tokens can we expect to train a small GPT-2 model (124M parameters) on during that time?

- 8.6T

Incorrect.

- 860B

Incorrect.

- 86B

Correct. Given the computational capacity ($C = 64 \times 10^{18}$ FLOPs) and the model size ($P = 124 \times 10^6$), we can estimate the number of tokens T as

$$T = \frac{C}{6 \times P} = \frac{64 \times 10^{18}}{6 \times 124 \times 10^6} \approx 86 \times 10^9 \text{ tokens}$$

This equates to approximately 86 billion tokens.

4. Based on the results from the Chinchilla paper, approximately how many tokens should we train on per model parameter in a compute-optimal model?

- 6

Incorrect.

- 20

Correct. Looking at the slide “Compute-optimal models”, the optimal model size of 63B parameters is achieved for 1.4T tokens. This is approximately 20 tokens per parameter.

- 100

Incorrect.

5. Based on the results from the Chinchilla paper, which of the following models was least compute-optimal?

- Gopher (280B)

Incorrect. While Gopher was not compute-optimal, Megatron-Turing NLG is even less so.

- GPT-3 (175B)

Incorrect. GPT-3 also deviated from compute-optimal training, but not to the extent of Megatron-Turing NLG.

- Megatron-Turing NLG (530B)

Correct. Megatron-Turing NLG, with 530 billion parameters, was significantly undertrained relative to the compute-optimal guidelines suggested in the Chinchilla paper.

Lecture 3.5

1. How many trainable parameters does the GPT-3 model have?
 - 117 B
Incorrect.
 - 175 B
Correct.
 - 1542 B
Incorrect.
2. What was the main insight that came with GPT-1?
 - Next word prediction is an effective pre-training strategy for many tasks in NLP
Correct. GPT-1 demonstrated that unsupervised pre-training using next word prediction could significantly improve performance across various NLP tasks.
 - Even 117-M-parameter models can be effectively trained on suitable hardware
Incorrect. While GPT-1 had 117 million parameters, the main insight was related to the effectiveness of unsupervised pre-training, not hardware capabilities.
 - Masked language modelling is more important than next sentence prediction
Incorrect. This insight is associated with BERT, not GPT-1.
3. Consider the Winograd example in the slide on zero-shot learning. Which of the following would be the matching prompt for a next-word distribution where $p(demonstrators) > p(councilmen)$ and $they = demonstrators$?
 - The city councilmen refused the demonstrators a permit because they advocated violence.
Correct. In this sentence, *they* refers to *demonstrators*.
 - The demonstrators attacked the city councilmen because they refused them a permit.
Incorrect. Here, *they* refers to *councilmen*, not *demonstrators*.
 - The city councilmen refused the demonstrators additional permits because they were too expensive.
Incorrect. In this context, *they* refers to *permits*, not *demonstrators*.
4. How is in-context learning different from zero-shot learning?

- The model can learn new tasks from examples.

Correct. In-context learning involves providing the model with examples within the input context, enabling it to adapt to new tasks without parameter updates.

- The model has more than one attempt at predicting the next word.

Incorrect. Both in-context and zero-shot learning involve single forward passes for predictions.

- The model can update its gradients based on the words in the context of the prediction.

Incorrect. In-context learning does not involve gradient updates; the model adapts based on the input context alone.

5. In the slide on prompt engineering, how much better was the LLM-designed prompt compared to the best human-designed prompt?

- 3.3 points

Correct. The LLM-designed prompt outperformed the best human-designed prompt by 3.3 points, indicating a measurable improvement in performance.

- 3.3 percent

Incorrect. A 3.3 percent improvement from 78.7 (the accuracy of the best human-designed prompt) would correspond to an accuracy of 81.3, not 82.0 (the accuracy obtained with the LLM-designed prompt).

- 3.3 times

Incorrect. The improvement was not a multiple of 3.3 times but an increase of 3.3 points.

Lecture 3.6

1. Assistant models are trained in several stages. Which of the following stages does not involve the language modelling objective?

- reward modelling

Correct. Reward modelling focuses on learning to predict human preferences and does not use the language modelling objective, which involves next-token prediction.

- unsupervised pre-training

Incorrect. Unsupervised pre-training directly uses the language modelling objective to predict the next token in a sequence.

- instruction fine-tuning

Incorrect. Instruction fine-tuning still uses the language modelling objective but incorporates task-specific instructions to guide responses.

2. Consider a customer support chatbot with a female avatar. Which of the following outputs would be an example showing the limitation of language modelling as an objective in instruction fine-tuning?

- I'm a woman so I just don't understand.

Correct. This output is okay from a language modelling perspective but demonstrates a failure in aligning the model to avoid generating biased or inappropriate statements.

- Not I understand sorry sorry.

Incorrect. While grammatically incorrect, this example reflects a language modeling failure, not a limitation in the objective.

- Can you repeat that? I am not sure I understand.

Incorrect. This is an appropriate and coherent response, showing no clear limitation of the language modeling objective.

3. What is the purpose of reward models in LLM training?

- to act as a proxy for immediate human feedback

Correct. Reward models are trained to predict human preferences and serve as a proxy for human feedback during reinforcement learning.

- to suggest suitable payment for human annotators

Incorrect. Reward models are unrelated to annotator compensation.

- to increase the computational efficiency of the training process

Incorrect. Reward models are used to align the model's behaviour with human preferences, not to improve computational efficiency.

4. For a well-aligned LLM, which of the following is the most plausible reward model loss for the two completions from the moon landing example?

- 0.11

Correct. A low loss (e.g., 0.11) indicates that the reward model successfully distinguishes between the preferred and non-preferred completions.

- 0.69

Incorrect. A loss around 0.69 suggests random guessing, indicating the reward model is not distinguishing between the outputs effectively.

- 2.30

Incorrect. A higher loss like 2.30 implies the model is failing to align with human preferences.

5. What is the goal of policy gradient methods?

- maximise the probability of generating outputs with high reward

Correct. Policy gradient methods aim to optimise the model's parameters to increase the likelihood of generating outputs that receive higher rewards.

- maximise the reward of generated outputs

Incorrect. While related, policy gradient focuses on increasing the probability of high-reward outputs rather than the absolute reward.

- minimise the perplexity of the generated outputs

Incorrect. Minimising perplexity is associated with language modelling, not reinforcement learning using policy gradients.