DataInterview.com

85

Generative AI

Interview Questions 📚

*Seen in **ML Engineer** and **AI Engineer** interviews at FAANGs, startups and consulting firms*

Technical Foundation

1. Explain how the self-attention layer works in Transformer model work
2. Describe the backpropagation algorithm.
3. How does a single-layer perceptron differ from a multi-layer perceptron?
4. What is the purpose of an activation function in a neural network?
5. Explain the difference between weight initialization methods like Xavier and He initialization.
6. Describe the working of the dropout regularization technique.
7. How do pooling layers in CNNs work and why are they important?
8. Explain the concept of "depth" in a neural network.
9. How do LSTMs address the vanishing gradient problem?
10. Describe the difference between batch normalization and layer normalization.
11. What is the skip connection or residual connection in deep networks?
12. Compare and contrast feedforward networks with recurrent networks.
13. Explain the difference between one-hot encoding and word embeddings.
14. How does a max-pooling layer differ from an average pooling layer in a CNN?
15. What are the typical applications of autoencoders?
16. Explain the significance of the bias term in neural networks.
17. What are the potential issues with using a sigmoid activation function in deep networks?
18. How does a self-attention mechanism work in transformers?
19. What challenges arise when training very deep neural networks?
20. Describe the concept of "transfer learning" and its advantages.

Technical Foundation Cont’d

1. Explain the role of a validation set in model training.
2. Why might a neural network's training loss decrease while its validation loss increases?
3. Describe the challenges of training a deep network from scratch.
4. How can you handle imbalanced datasets in neural network training?
5. Explain the difference between stochastic gradient descent (SGD) and mini-batch gradient descent.
6. What is the adaptive learning rate, and why is it beneficial?
7. Describe the workings of the Adam optimizer and its advantages.
8. How do learning rate schedulers work, and why are they used?
9. Why do we shuffle the training data after each epoch?
10. Why is the learning rate considered one of the most important hyperparameters in neural network training?
11. Explain the momentum term in optimization algorithms.
12. What is the batch size in neural network training, and how does it affect convergence?
13. What are weight constraints, and how can they benefit training?
14. Explain the significance of the second moment in the Adam optimizer.
15. How does the RMSprop optimizer differ from vanilla SGD?
16. What are common symptoms of overfitting, and how can you diagnose them?
17. Describe the difference between global and local optima.
18. How do techniques like gradient clipping help in training?
19. What are the benefits of data augmentation in deep learning?
20. How does early stopping prevent overfitting?

Reinforcement Learning

1. What is reinforcement learning, and how does it differ from supervised and unsupervised learning?
2. Can you explain the concept of the Markov Decision Process (MDP) in the context of reinforcement learning?
3. What are the main components of a reinforcement learning agent?
4. How do you define the reward function in a reinforcement learning problem, and why is it important?
5. What is the difference between model-based and model-free reinforcement learning?
6. Can you explain Q-learning and how it is used in reinforcement learning?
7. What is the role of the discount factor in reinforcement learning algorithms?
8. How does the exploration-exploitation trade-off influence reinforcement learning agent performance?
9. What are policy gradient methods, and how do they differ from value iteration methods?
10. Explain the State (V) and Action-Value (Q) functions
11. How do you handle continuous action spaces in reinforcement learning?
12. What is deep reinforcement learning, and how does it integrate deep learning with reinforcement learning?
13. How do you ensure the convergence of a reinforcement learning algorithm?
14. What are the challenges of deploying reinforcement learning models in production environments?
15. How do multi-agent reinforcement learning systems work, and what are their applications?

Large Language Models

1. Define "pre-training" vs. "fine-tuning" in LLMs.
2. How do models like Stability Diffusion leverage LLMs to understand complex text prompts and generate high-quality images?
3. How do you train LLM models with billions of parameters?
4. How does RAG work?
5. How does LoRA work?
6. How do you train an LLM model that prevents prompt hallucinations?
7. How do you prevent bias and harmful prompt generation?
8. How does proximal policy gradient work in a prompt generation?
9. How does knowledge distillation benefit LLMs?
10. What's "few-shot" learning in LLMs?
11. Evaluating LLM performance metrics?
12. How would you use RLHF to train an LLM model?
13. What techniques can be employed to improve the factual accuracy of text generated by LLMs?
14. How would you detect drift in LLM performance over time, especially in real-world production settings?
15. Describe strategies for curating a high-quality dataset tailored for training a generative AI model.
16. What methods exist to identify and address biases within training data that might impact the generated output?
17. How would you fine-tune LLM for domain-specific purposes like financial and medical applications?
18. Explain the algorithm architecture for LLAMA and other LLMs alike.

LLM System Design

1. You need to design a system that uses an LLM to generate responses to a massive influx of user queries in near real-time. Discuss strategies for scaling, load balancing, and optimizing for rapid response times.
2. How would you incorporate caching mechanisms into an LLM-based system to improve performance and reduce computational costs? What kinds of information would be best suited for caching?
3. How would you reduce model size and optimize for deployment on resource-constrained devices (e.g., smartphones).
4. Discuss the trade-offs of using GPUs vs. TPUs vs. other specialized hardware when deploying large language models.
5. How would you build a ChatGPT-like system?
6. System design an LLM for code generation tasks. Discuss potential challenges.
7. Describe an approach to using generative AI models for creating original music compositions.
8. How would you build an LLM-based question-answering system for a specific domain or complex dataset?
9. What design considerations are important when building a multi-turn conversational AI system powered by an LLM?
10. How can you control and guide the creative output of generative models for specific styles or purposes?
11. How do vector databases work?
12. How do you monitor LLM systems once productionized?