Генеративные модели компьютерного зрения

1. ***Что такое генеративная модель в компьютерном зрении и какие задачи она решает? Чем генеративная модель отличается от дискриминативной?***

Генеративная модель создаёт новые данные (изображения), обучаясь на распределении реальных данных. Решает задачи синтеза, дорисовки, стилизации. Отличается от дискриминативной, которая классифицирует/разделяет данные.

2. ***Что называют латентным пространством?***

Латентное пространство — сжатое векторное представление данных, где каждая точка соответствует данным с определёнными признаками.

3. ***Что такое автоэнкодер и какова его структура?***

Автоэнкодер — нейросеть из энкодера (сжимает вход в латентный вектор) и декодера (восстанавливает данные из вектора).

4. ***Как работает декодер в автоэнкодере?***

Декодер преобразует латентный вектор обратно в данные, стремясь минимизировать разницу с оригиналом.

5. ***В чём отличие вариационного автоэнкодера (VAE) от обычного AE?***

VAE обучается на вероятностном латентном пространстве (с распределением), а не на детерминированных векторах, что позволяет генерировать новые данные.

6. ***Что делает регуляризация Кульбака–Лейблера (KL-divergence) в VAE?***

KL-дивергенция регулирует латентное пространство, приближая его к нормальному распределению для плавной генерации.

7. ***Почему выходные изображения VAE часто размыты?***

Из-за регуляризации и использования MSE-потерь, которые усредняют варианты.

8. ***Что такое генеративно-состязательная сеть (GAN)?***

GAN — генеративная модель, где генератор создаёт данные, а дискриминатор отличает их от реальных.

9. ***Из каких частей состоит GAN?***

GAN состоит из генератора и дискриминатора.

10. ***Какова цель дискриминатора в GAN?***

Цель дискриминатора — точно отличать сгенерированные данные от реальных.

11. ***Как происходит процесс обучения генератора и дискриминатора в классическом GAN?***

Генератор и дискриминатор обучаются одновременно в состязательном процессе: генератор улучшает подделку, дискриминатор — распознавание.

12. ***Какая функция потерь в классическом GAN?***

Минимаксная функция потерь:

min_G max_D [log D(x) + log(1 - D(G(z)))]

13. ***Объясните термин mode collapse. Для какого семейства генеративных моделей свойственен mode collapse?***

Mode collapse — генератор выучивает ограниченное разнообразие образцов. Свойственен GAN.

14. ***Почему GAN часто нестабилен при обучении?***

Нестабильность из-за дисбаланса между генератором и дискриминатором, сложности достижения равновесия.

15. ***Что такое conditional GAN? Как в conditional GAN добавляется условие (label, prompt и тд)?***

Conditional GAN — GAN с условием (класс, текст). Условие добавляется в вход генератору и/или дискриминатору через конкатенацию, эмбеддинги и т.д..

16. Что такое диффузионная модель (Denoising Diffusion Probabilistic Model)?

Диффузионная модель — генеративная модель, которая постепенно зашумляет данные, затем учится обратному процессу для генерации.

17. Как организован процесс добавления шума на изображение в диффузионной модели?

Шум добавляется постепенно за много шагов по расписанию (scheduler).

18. Как происходит обратная диффузия в DDPM? Что означает процесс «denoising»?

Обратная диффузия — процесс пошагового удаления шума из случайного шума для получения данных. «Denoising» — предсказание шума на каждом шаге.

19. Что делает Scheduler в диффузионных моделях?

Scheduler управляет расписанием добавления/удаления шума (скорость, степень).

20. Что делает Latent Diffusion Model? Почему латентные модели работают быстрее?

Latent Diffusion Model диффундирует в латентном пространстве (например, сжатом через VAE), а не в пикселях, что ускоряет обучение и генерацию из-за меньшей размерности.

21. Что делает автоэнкодер в LDM?

Автоэнкодер в LDM сжимает изображение в латентное представление и восстанавливает его. Диффузионный процесс происходит в этом латентном пространстве, что ускоряет генерацию.

22. Как используется текстовый энкодер в Stable Diffusion?

Текстовый энкодер в Stable Diffusion (обычно CLIP или T5) преобразует текстовый промпт в векторное представление (эмбеддинг), которое направляет диффузионный процесс для генерации соответствующего изображения.

23. Что такое CLIP?

CLIP — модель от OpenAI, которая обучается на парах "изображение-текст" для сопоставления их в общем векторном пространстве. Позволяет оценивать схожесть изображений и текстовых описаний.

24. Что делает guidance scale в Stable Diffusion?

Guidance scale в Stable Diffusion контролирует силу влияния текстового промпта на генерацию. Высокое значение усиливает соответствие промпту, но может снизить качество/разнообразие.

25. Что означает classifier-free guidance?

Classifier-free guidance — техника управления генерацией без отдельной классифицирующей модели. Использует разность предсказаний модели с промптом и без промпта для усиления условия.

26. Что такое prompt и как он влияет на результат?

Prompt — текстовое описание желаемого изображения. Он направляет генерацию через текстовый эмбеддинг, влияя на содержание, стиль и композицию.

27. Как оценить качество сгенерированных изображений (какие есть типы метрик, какие аспекты изображений они оценивают)?

* IS (Inception Score): Оценивает чёткость и разнообразие изображений (через классификатор Inception-v3).
* FID: Сравнивает распределения признаков сгенерированных и реальных изображений (ближе к реальности → лучше).
* Precision & Recall: Оценивают качество и покрытие мод распределения данных.
* User Studies (человеческая оценка).

28. Что такое IS (Inception Score)?

IS (Inception Score) — метрика, которая оценивает, насколько сгенерированные изображения разнообразны (высокая энтропия по классам) и при этом чётко определённы (низкая энтропия внутри одного изображения).

29. Что такое FID (Fréchet Inception Distance)?

FID (Fréchet Inception Distance) вычисляет расстояние между распределениями признаков реальных и сгенерированных изображений, извлечённых моделью Inception-v3. Чем ниже FID, тем ближе сгенерированные данные к реальным.

30. Что такое auto-regressive модели для изображений?

Auto-regressive модели для изображений генерируют изображение по пикселям или патчам последовательно, каждый раз предсказывая следующий элемент на основе уже созданных.

31. Что делает PixelCNN?

PixelCNN — это авторегрессивная модель, которая предсказывает распределение цвета каждого пикселя условно от всех предыдущих (левых и верхних) пикселей.

32. Почему PixelCNN трудоёмок при генерации?

PixelCNN трудоёмок, так как генерация происходит последовательно (пиксель за пикселем), а не параллельно для всего изображения сразу.

33. Что делает masked convolution в PixelCNN?

Masked convolution в PixelCNN использует маску, которая обнуляет веса для "будущих" пикселей (справа и снизу от текущего), чтобы обеспечить корректный авторегрессивный порядок генерации.

34. Как происходит генерация изображения по тексту в CLIP-conditioned модели?

В CLIP-conditioned модели CLIP используется для получения эмбеддинга текста (промпта), который затем направляет авторегрессивный или диффузионный процесс генерации изображения, стремясь максимизировать соответствие между сгенерированным изображением и текстом в пространстве CLIP.

35. Что означает inpainting в генеративных моделях?

Inpainting — задача заполнения пропущенных (удалённых) областей изображения правдоподобным содержанием с учётом контекста.

36. Как оценить разнообразие сгенерированных изображений?

* Вычисляя метрики (FID, Precision/Recall, покрытие латентного пространства).
* Визуальным осмотром на наличие повторяющихся паттернов.
* Анализируя дистанции между сгенерированными образцами.

37. Как бороться с mode collapse?

* Регулярные архитектуры (Wasserstein GAN с градиентным штрафом).
* Разнообразные и качественные данные для обучения.
* Использование мини-батчей в дискриминаторе (Mini-batch Discrimination).
* Применение диффузионных моделей, менее склонных к коллапсу.

38. Какие методы ускоряют inference диффузионных моделей?

* Улучшенные сэмплеры (DDIM, DPM-Solver).
* Сокращение числа шагов (меньше шагов дениойзинга).
* Кодек и работа в латентном пространстве (как в LDM/Stable Diffusion).
* Distillation (обучение студенческой сети за меньше шагов).

39. Что такое VQ-AR?

VQ-AR (Vector Quantized Autoregressive model) — модель, которая сначала квантует изображение в дискретные токены с помощью VQ-VAE, а затем генерирует последовательность этих токенов авторегрессивно (например, с помощью Transformer).

40. Почему изображения делят на патчи в VQ-AR?

В VQ-AR изображения делят на патчи для того, чтобы:

* Уменьшить длину последовательности токенов для авторегрессивной модели.
* Уловить локальные зависимости и структуры в изображении.
* Сделать обучение и генерацию вычислительно эффективнее.

41. Как ViT обучается классификации изображений?

ViT обучается классификации изображений путём:

* Разбиения изображения на патчи и их линейного проецирования в эмбеддинги.
* Добавления позиционных энкодирований и [CLS]-токена.
* Пропускания последовательности через Transformer-энкодер.
* Использования эмбеддинга [CLS]-токена для финальной классификации.

42. Чем ViT отличается от CNN?

Отличия ViT от CNN:

* Архитектура: ViT использует механизм внимания (self-attention) глобально, CNN — локальные свёртки.
* Индуктивные смещения: CNN имеет смещения к локальности и трансляционной инвариантности, ViT — минимальные, учит всё из данных.
* Требования к данным: ViT требует больше данных для обучения с нуля.

43. Что такое cross-attention в Stable Diffusion?

Cross-attention в Stable Diffusion позволяет текстовому промпту влиять на диффузионный процесс. В U-Net происходит "скрещивание" латентных представлений изображения и текстовых эмбеддингов: каждый элемент изображения "внимает" ко всем элементам текста.

44. Что такое нормализационные потоки?

Нормализационные потоки (NF) — генеративные модели, которые обучают обратимое и дифференцируемое преобразование между сложным распределением данных и простым (например, гауссовским).

45. Что означает «обратимость» преобразований в нормализационных потоках?

"Обратимость" в NF означает, что преобразование может быть выполнено в обе стороны без потери информации: $z = f(x)$ и $x = f^{-1}(z)$.

46. Как нормализационные потоки позволяют вычислять плотность вероятности данных?

NF вычисляют плотность вероятности используя формулу замены переменных: $p(x) = p(z) * |\det(J(f^{-1}))|$, где J — якобиан. Это возможно благодаря обратимости и известному якобиану.

47. В чём отличие Normalizing Flow от VAE и GAN?

Отличие NF от VAE и GAN:

* NF: Точное вычисление плотности, обратимость.
* VAE: Оценка нижней границы правдоподобия (ELBO), необратима.
* GAN: Неявное моделирование распределения, без прямого вычисления плотности.

48. Почему важно, чтобы преобразование в NF было дифференцируемым?

Дифференцируемость в NF важна для:

* Вычисления якобиана (необходимого для формулы плотности).
* Обучения методом обратного распространения ошибки.

49. Как такое RealNVP?

RealNVP — архитектура NF с аффинными преобразованиями масштаба и сдвига, где преобразования построены так, что их якобиан легко вычисляется (треугольная матрица).

50. Какие преимущества NF имеют по сравнению с VAE?

Преимущества NF перед VAE:

* Точное вычисление правдоподобия (не ELBO).
* Обратимость без потерь.
* Простой латентный код (произвольная выборка из простого распределения).

51. Каковы основные ограничения нормализационных потоков?

Основные ограничения NF:

* Вычислительная сложность из-за расчёта определителя якобиана.
* Ограниченная гибкость преобразований для сохранения обратимости.
* Трудности с масштабированием на очень высокие размерности.

52. Как генеративные модели помогают в задаче дополнения данных (data augmentation)?

Генеративные модели помогают в data augmentation создавая синтетические, но реалистичные образцы данных, расширяя тренировочный набор и улучшая обобщающую способность моделей.

53. В чём достоинство диффузионных моделей по сравнению с GAN?

Достоинство диффузионных моделей перед GAN:

* Более стабильное обучение (без проблемы коллапса мод).
* Высокое качество и разнообразие сгенерированных изображений.
* Лучший режим покрытия распределения данных.

54. Какие задачи можно решать при помощи VAE?

Задачи для VAE:

* Генерация новых данных.
* Сжатие данных (получение латентных представлений).
* Дениойзинг.
* Inpainting.
* Получение осмысленных латентных пространств для интерполяции.

55. В чём недостаток диффузионных моделей с точки зрения вычислительных затрат?

Высокие вычислительные затраты на inference, так как генерация требует многих шагов (сотни-тысячи) последовательных предсказаний нейросети.

Введение в NLP и генеративные модели NLP

56. Что такое механизм внимания (attention)?

Механизм внимания (attention) — это вычислительный механизм, который позволяет модели взвешивать важность разных элементов входной последовательности при обработке каждого отдельного элемента.

57. Зачем нужен attention в нейросетях?

Attention в нейросетях нужен для:

* Моделирования зависимостей независимо от расстояния между элементами.
* Фокусировки на релевантных частях входных данных.
* Улучшения передачи контекста.

58. Что означают термины Query, Key, Value?

Термины в attention:

* Query (Q) — "запрос": что мы ищем (текущий элемент).
* Key (K) — "ключ": что мы сравниваем с запросом (все элементы).
* Value (V) — "значение": информация, которая извлекается (взвешенная сумма значений).

59. Как вычисляется attention score?

Attention score вычисляется как скалярное произведение Query и Key (или другие функции сходства), нормированное для стабильности: `score = (Q * K^T) / sqrt(d_k)`

60. Что делает softmax в attention-механизме?

Softmax в attention преобразует attention scores в вероятностное распределение (веса в сумме = 1), определяя какой вес присвоить каждому Value.

61. Чем self-attention отличается от обычного attention?

Self-attention — это attention, где Query, Key и Value берутся из одного и того же источника (входной последовательности). Обычный (cross-)attention использует разные источники.

62. Что делает multi-head attention?

Multi-head attention выполняет attention параллельно в нескольких проекционных подпространствах, затем объединяет результаты, позволяя модели фокусироваться на разных типах зависимостей.

63. Почему несколько голов внимания повышают точность трансформерных моделей (по сравнению с одной головой внимания)?

* Несколько голов внимания повышают точность, потому что:
* Параллельно изучают разные типы зависимостей (синтаксические, семантические, дальние).
* Увеличивают представительную способность модели.

64. Что делает residual connection в трансформере?

Residual connection (остаточное соединение) добавляет вход слоя к его выходу: output = layer(x) + x. Это:

* Помогает градиентам течь через глубокую сеть (борьба с затуханием градиента).
* Сохраняет исходную информацию.

65. Как работает layer normalization?

Layer normalization нормализует активации по фичам для каждого примера отдельно (в отличие от batch norm). Стабилизирует обучение.

66. Что делает position encoding и зачем он нужен?

Position encoding добавляет информацию о позиции токенов в последовательности, поскольку механизм внимания сам по себе не учитывает порядок. Может быть синусоидальным или обучаемым.

67. Что делает feed-forward слой в трансформере?

Feed-forward слой в трансформере — это двухслойная полносвязная сеть с нелинейностью между слоями, применяемая к каждому токену отдельно. Добавляет нелинейную выразительность.

68. Из чего состоит encoder в классическом трансформере?

Encoder в классическом трансформере состоит из N одинаковых блоков, каждый из которых содержит:

* Multi-head self-attention + residual & norm
* Feed-forward network + residual & norm

69. Из чего состоит decoder в классическом трансформере?

Decoder в классическом трансформере состоит из N одинаковых блоков, каждый из которых содержит:

* Masked multi-head self-attention + residual & norm (для предотвращения "подглядывания" в будущее)
* Multi-head cross-attention (на выход энкодера) + residual & norm
* Feed-forward network + residual & norm

70. Что делает masked self-attention?

Masked self-attention в декодере маскирует (обнуляет) будущие токены при вычислении внимания, чтобы генерация была авторегрессивной (текущий токен зависит только от предыдущих).

71. Что делает cross-attention?

Cross-attention позволяет элементам одной последовательности (обычно запросам) взаимодействовать с элементами другой последовательности (ключами и значениями). Используется для связи между модальностями, например, текстом и изображением в Stable Diffusion.

72. Какова вычислительная сложность self-attention?

Вычислительная сложность self-attention — $O(n^2 d)$, где n — длина последовательности, d — размерность эмбеддинга. Квадратичность возникает из-за попарных взаимодействий.

73. Как трансформеры применяются для генерации текстов?

Трансформеры для генерации текстов (как GPT) используют авторегрессивную генерацию: на каждом шаге модель предсказывает следующий токен на основе предыдущих, с causal attention.

74. Что такое masked language modeling (MLM)?

Masked Language Modeling (MLM) — задача предсказания замаскированных (скрытых) токенов в предложении. Модель видит контекст с обеих сторон, что позволяет обучать двунаправленные представления (как в BERT).

75. Почему BERT использует двунаправленный attention?

BERT использует двунаправленный attention, потому что обучается на MLM: для предсказания замаскированного токена ему нужен контекст со всех сторон, что даёт более глубокое понимание языка.

76. Чем GPT отличается от BERT?

Отличие GPT от BERT:

* GPT: Decoder-only, causal attention, обучается на предсказании следующего токена (слева направо), лучше для генерации.
* BERT: Encoder-only, двунаправленное внимание, обучается на MLM, лучше для понимания (классификация, извлечение).

77. Что делает causal attention?

Causal attention (маскированное самовнимание) — механизм, где каждый токен внимает только к предыдущим токенам. Используется в авторегрессивных моделях для соблюдения причинно-следственного порядка.

78. Чем encoder-only отличается от decoder-only архитектуры?

Отличия архитектур:

* Encoder-only (BERT): Кодирует вход в представления, подходит для анализа (классификация, NER).
* Decoder-only (GPT): Авторегрессивно генерирует выход, подходит для генерации текста.
* Encoder-decoder (T5, BART): Кодирует вход, декодирует выход, подходит для задач "преобразования" (перевод, суммаризация).

79. Что делает encoder-decoder модель в NLP и какие названия таких моделей Вы знаете?

Encoder-decoder модель принимает входную последовательность, кодирует её, затем декодирует в выходную. Примеры: T5, BART, MarianMT (перевод).

80. Что такое prompt engineering?

Prompt engineering — это искусство формулировки промпта (текстового запроса) для получения нужного результата от языковой модели. Включает подбор слов, формата, примеров.

81. Что такое prompt tuning?

Prompt tuning — это обучение непрерывных (векторных) промптов при замороженных весах модели. Менее затратная адаптация, чем fine-tuning.

82. Что означает instruction-tuning?

Instruction-tuning — это дообучение модели на наборе задач, сформулированных как инструкции, чтобы научить её следовать указаниям и лучше обобщаться на новые задачи.