ML, DL y GPT

1. El Aprendizaje Automático (ML) es un campo de la ciencia de la computación que permite a los sistemas aprender de los datos y mejorar su rendimiento sin programación explícita.
2. El Aprendizaje Profundo (DL) es un subcampo de ML que utiliza redes neuronales multicapa para modelar patrones complejos en los datos.
3. Las Redes Neuronales son modelos computacionales inspirados en el cerebro humano, compuestos de nodos interconectados (neuronas) que procesan información en capas.
4. Los Datos de Entrenamiento es el conjunto de datos etiquetados o no etiquetados utilizado para enseñar a un modelo de aprendizaje automático cómo realizar una tarea.
5. El Aprendizaje Supervisado implica entrenar un modelo en datos etiquetados, donde cada ejemplo tiene una entrada y una salida correcta asociada.
6. El Aprendizaje No Supervisado utiliza datos no etiquetados, permitiendo que el modelo descubra patrones ocultos o agrupaciones sin instrucciones explícitas.
7. El Aprendizaje por Reforzamiento (RL) entrena a los agentes para tomar decisiones recompensando comportamientos deseados y penalizando los no deseados.
8. Los Modelos Generativos aprenden a producir nuevos datos similares a sus ejemplos de entrenamiento (por ejemplo, texto, imágenes).
9. Los Modelos Discriminativos se centran en clasificar entradas en categorías o predecir resultados específicos.
10. El Aprendizaje por Transferencia permite que un modelo entrenado en una tarea se reutilice o ajuste en una tarea relacionada.
11. GPT (Generative Pre-trained Transformer) es una familia de grandes modelos de lenguaje desarrollados por OpenAI que pueden generar texto similar al humano.
12. ChatGPT es una variante interactiva de GPT, ajustada para tareas de conversación y seguimiento de instrucciones.
13. La Arquitectura Transformer fue introducida en el artículo “Attention Is All You Need”, revolucionando el procesamiento del lenguaje natural al confiar en mecanismos de atención.
14. Los Mecanismos de Autoatención permiten que el modelo pondere diferentes partes de la secuencia de entrada al construir una representación de salida.
15. La Codificación Posicional en Transformers ayuda al modelo a identificar el orden de los tokens en una secuencia.
16. El Preentrenamiento es la fase inicial en la que un modelo aprende características generales a partir de datos a gran escala antes de ajustarse a tareas específicas.

17. El Ajuste es el proceso de tomar un modelo preentrenado y adaptarlo a una tarea más específica utilizando un conjunto de datos más pequeño y específico de la tarea.

18. El Modelado del Lenguaje es la tarea de predecir el siguiente token (palabra o subpalabra) en una secuencia, fundamental para modelos similares a GPT.

19. El Aprendizaje Cero Disparo permite que un modelo maneje tareas sin ejemplos de entrenamiento explícitos, confiando en el conocimiento general aprendido.

20. El Aprendizaje de Pocos Disparos aprovecha un número limitado de ejemplos específicos de la tarea para guiar las predicciones o comportamientos del modelo.

21. RLHF (Reinforcement Learning from Human Feedback) se utiliza para alinear las salidas del modelo con las preferencias y valores humanos.

22. La Retroalimentación Humana puede incluir clasificaciones o etiquetas que guíen la generación del modelo hacia respuestas más deseadas.

23. La Ingeniería de Prompts es el arte de elaborar consultas de entrada o instrucciones para guiar eficazmente a los grandes modelos de lenguaje.

24. La Ventana de Contexto se refiere a la cantidad máxima de texto que el modelo puede procesar a la vez; los modelos GPT tienen una longitud de contexto limitada.

25. La Inferencia es la etapa en la que un modelo entrenado hace predicciones o genera salidas dados nuevos ingresos.

26. La Cuenta de Parámetros es un factor clave en la capacidad del modelo; los modelos más grandes pueden capturar patrones más complejos pero requieren más computación.

27. Las Técnicas de Compresión de Modelos (por ejemplo, poda, cuantización) reducen el tamaño del modelo y aceleran la inferencia con una mínima pérdida de precisión.

28. Las Cabezas de Atención en Transformers procesan diferentes aspectos de la entrada en paralelo, mejorando el poder representativo.

29. El Modelado de Lenguaje Mascarado (por ejemplo, en BERT) implica predecir tokens faltantes en una oración, ayudando al modelo a aprender el contexto.

30. El Modelado de Lenguaje Causal (por ejemplo, en GPT) implica predecir el siguiente token basado en todos los tokens anteriores.

31. La Arquitectura Codificador-Decodificador (por ejemplo, T5) utiliza una red para codificar la entrada y otra para decodificarla en una secuencia objetivo.

32. Las Redes Neuronales Convolucionales (CNN) se destacan en el procesamiento de datos en forma de cuadrícula (por ejemplo, imágenes) a través de capas convolucionales.

33. Las Redes Neuronales Recurrentes (RNN) procesan datos secuenciales pasando estados ocultos a lo largo de los pasos temporales, aunque pueden tener dificultades con dependencias a largo plazo.

34. La Memoria a Largo Plazo (LSTM) y GRU son variantes de RNN diseñadas para capturar mejor las dependencias a largo plazo.

35. La Normalización por Lotes ayuda a estabilizar el entrenamiento normalizando las salidas de las capas intermedias.

36. El Dropout es una técnica de regularización que “elimina”aleatoriamente neuronas durante el entrenamiento para prevenir el sobreajuste.

37. Los Algoritmos Optimizadores como el Descenso de Gradiente Estocástico (SGD), Adam y RMSProp actualizan los parámetros del modelo en función de los gradientes.

38. La Tasa de Aprendizaje es un hiperparámetro que determina cuán drásticamente se actualizan los pesos durante el entrenamiento.

39. Los Hiperparámetros (por ejemplo, tamaño del lote, número de capas) son configuraciones elegidas antes del entrenamiento para controlar cómo se desarrolla el aprendizaje.

40. El Sobreajuste del Modelo ocurre cuando un modelo aprende los datos de entrenamiento demasiado bien, fallando en generalizar a nuevos datos.

41. Las Técnicas de Regularización (por ejemplo, decadencia de peso L2, dropout) ayudan a reducir el sobreajuste y mejorar la generalización.

42. El Conjunto de Validación se utiliza para ajustar los hiperparámetros, mientras que el Conjunto de Prueba evalúa el rendimiento final del modelo.

43. La Validación Cruzada divide los datos en múltiples subconjuntos, entrenando y validando sistemáticamente para obtener una estimación de rendimiento más robusta.

44. Los Problemas de Explosión y Desvanecimiento de Gradientes ocurren en redes profundas, haciendo que el entrenamiento sea inestable o ineficaz.

45. Las Conexiones Residuales (conexiones de salto) en redes como ResNet ayudan a mitigar los gradientes desvanecidos acortando las rutas de datos.

46. Las Leyes de Escalado sugieren que aumentar el tamaño del modelo y los datos generalmente lleva a un mejor rendimiento.

47. La Eficiencia Computacional es crítica; el entrenamiento de grandes modelos requiere hardware optimizado (GPUs, TPUs) y algoritmos.

48. Las Consideraciones Éticas incluyen el sesgo, la equidad y el daño potencial; los modelos de ML deben ser probados y monitoreados cuidadosamente.

49. La Augmentación de Datos expande artificialmente los conjuntos de datos de entrenamiento para mejorar la robustez del modelo (especialmente en tareas de imagen y voz).

50. El Preprocesamiento de Datos (por ejemplo, tokenización, normalización) es esencial para un entrenamiento de modelo efectivo.

51. La Tokenización divide el texto en tokens (palabras o subpalabras), las unidades fundamentales procesadas por los modelos de lenguaje.

52. Las Representaciones Vectoriales representan tokens o conceptos como vectores numéricos, preservando relaciones semánticas.

53. Las Representaciones Posicionales añaden información sobre la posición de cada token para ayudar a un Transformer a entender el orden de la secuencia.

54. Los Pesos de Atención revelan cómo un modelo distribuye la atención en diferentes partes de la entrada.

55. La Búsqueda de Haz es una estrategia de decodificación en modelos de lenguaje que mantiene múltiples salidas candidatas en cada paso para encontrar la mejor secuencia general.

56. La Búsqueda Avariciosa elige el token más probable en cada paso, pero puede llevar a salidas finales subóptimas.

57. La Temperatura en la muestra ajusta la creatividad de la generación de lenguaje: mayor temperatura = más aleatoriedad.

58. Los Métodos de Muestreo Top-k y Top-p (Núcleo) restringen los tokens candidatos a los k más probables o a una probabilidad acumulativa p, equilibrando diversidad y coherencia.

59. La Perplejidad mide cómo bien un modelo de probabilidad predice una muestra; una menor perplejidad indica un mejor rendimiento predictivo.

60. La Precisión y el Recall son métricas para tareas de clasificación, enfocándose en la corrección y la completitud, respectivamente.

61. La Puntuación F1 es la media armónica de la precisión y el recall, equilibrando ambas métricas en un solo valor.

62. La Exactitud es la fracción de predicciones correctas, pero puede ser engañosa en conjuntos de datos desequilibrados.

63. El Área Bajo la Curva ROC (AUC) mide el rendimiento de un clasificador a través de varios umbrales.

64. La Matriz de Confusión muestra las cuentas de verdaderos positivos, falsos positivos, falsos negativos y verdaderos negativos.

65. Los Métodos de Estimación de Incertidumbre (por ejemplo, Dropout de Monte Carlo) evalúan cuán confiado es un modelo en sus predicciones.

66. El Aprendizaje Activo implica consultar nuevos ejemplos de datos que el modelo tiene menos confianza, mejorando la eficiencia de los datos.

67. El Aprendizaje en Línea actualiza el modelo de manera incremental a medida que llegan nuevos datos, en lugar de volver a entrenar desde cero.

68. Los Algoritmos Evolutivos y los Algoritmos Genéticos optimizan modelos o hiperparámetros utilizando mutación y selección inspiradas en la biología.

69. Los Métodos Bayesianos incorporan conocimiento previo y actualizan creencias con datos entrantes, útiles para la cuantificación de la incertidumbre.

70. Los Métodos de Ensamble (por ejemplo, Random Forest, Gradient Boosting) combinan múltiples modelos para mejorar el rendimiento y la estabilidad.

71. El Ensamblaje (Bootstrap Aggregating) entrena múltiples modelos en diferentes subconjuntos de los datos, luego promedia sus predicciones.

72. El Boosting entrena iterativamente nuevos modelos para corregir errores cometidos por modelos previamente entrenados.

73. Los Árboles de Decisión de Gradient Boosting (GBDT) son poderosos para datos estructurados, a menudo superando a las redes neuronales simples.

74. Los Modelos Autorregresivos predicen el siguiente valor (o token) basado en salidas anteriores en una secuencia.

75. El Autoencoder es una red neuronal diseñada para codificar datos en una representación latente y luego decodificarlos, aprendiendo representaciones de datos comprimidas.

76. El Autoencoder Variacional (VAE) introduce un giro probabilístico para generar nuevos datos que se asemejan al conjunto de entrenamiento.

77. La Red Adversarial Generativa (GAN) enfrenta a un generador contra un discriminador, produciendo imágenes, texto u otros datos realistas.

78. El Aprendizaje Autosupervisado aprovecha grandes cantidades de datos no etiquetados creando tareas de entrenamiento artificiales (por ejemplo, predecir partes faltantes).

79. Los Modelos Fundamentales son grandes modelos preentrenados que se pueden adaptar a una amplia gama de tareas descendentes.

80. El Aprendizaje Multimodal integra datos de múltiples fuentes (por ejemplo, texto, imágenes, audio) para crear representaciones más ricas.

81. La Etiquetación de Datos es a menudo la parte más consumidora de tiempo del ML, requiriendo una anotación cuidadosa para la precisión.

82. El Cómputo en el Borde lleva la inferencia de ML más cerca de la fuente de datos, reduciendo la latencia y el uso de ancho de banda.

83. El Aprendizaje Federado entrena modelos a través de dispositivos o servidores descentralizados que mantienen muestras de datos locales, sin intercambiarlas.

84. El Aprendizaje Automático que Preserva la Privacidad incluye técnicas como la privacidad diferencial y la criptografía homomórfica para proteger datos sensibles.

85. La Inteligencia Artificial Explicable (XAI) tiene como objetivo hacer que las decisiones de los modelos complejos sean más interpretables para los humanos.

86. El Sesgo y la Equidad en ML necesitan una supervisión cuidadosa, ya que los modelos pueden aprender e amplificar sesgos sociales de manera involuntaria.

87. El Desplazamiento de Concepto ocurre cuando las propiedades estadísticas de la variable objetivo cambian con el tiempo, afectando el rendimiento del modelo.

88. La Prueba A/B compara dos o más versiones de un modelo para ver cuál funciona mejor en un entorno del mundo real.

89. La Aceleración por GPU aprovecha el cómputo paralelo en tarjetas gráficas para acelerar drásticamente el entrenamiento de ML.

90. Las TPUs (Unidades de Procesamiento de Tensores) son aceleradores de hardware especializados por Google para cargas de trabajo de aprendizaje profundo eficientes.

91. Los Frameworks de Código Abierto (por ejemplo, TensorFlow, PyTorch) proporcionan bloques de construcción y herramientas para el desarrollo de modelos de ML.

92. La Presentación de Modelos es la práctica de implementar modelos entrenados para que puedan manejar predicciones en tiempo real o por lotes.

93. La Escalabilidad es crucial para manejar grandes conjuntos de datos o tráfico pesado, requiriendo estrategias de entrenamiento e inferencia distribuidas.

94. MLOps combina el desarrollo de ML con prácticas operativas, centrándose en la reproducibilidad, las pruebas y la integración continua.

95. El Control de Versiones para datos y modelos asegura un seguimiento de experimentos consistente y la colaboración.

96. Las Estrategias de Implementación (por ejemplo, contenedores, microservicios) organizan cómo se empaquetan y sirven los modelos a escala.

97. El Monitoreo rastrea el rendimiento del modelo después de la implementación, vigilando degradaciones o anomalías.

98. El Retraining y las Actualizaciones de Modelos mantienen los modelos actualizados a medida que llegan nuevos datos y condiciones cambiantes.

99. La Complejidad Temporal (notación O) mide cómo escala el tiempo de ejecución de un algoritmo con el tamaño de la entrada; O(1) denota tiempo constante.

100. El Futuro de ML promete modelos cada vez más sofisticados y generales, pero debe abordar consideraciones éticas, sociales y ambientales.