Maschinelles Lernen, Deep Learning und GPT

1. Maschinelles Lernen (ML) ist ein Bereich der Informatik, der Systemen ermöglicht, aus Daten zu lernen und ihre Leistung ohne explizite Programmierung zu verbessern.
2. Deep Learning (DL) ist ein Teilgebiet des ML, das mehrschichtige neuronale Netze nutzt, um komplexe Muster in Daten zu modellieren.
3. Neuronale Netze sind rechnerische Modelle, die vom menschlichen Gehirn inspiriert sind und aus miteinander verbundenen Knoten (Neuronen) bestehen, die Informationen in Schichten verarbeiten.
4. Trainingsdaten sind der etikettierte oder nicht-etikettierte Datensatz, der verwendet wird, um ein maschinelles Lernmodell zu lehren, wie es eine Aufgabe ausführt.
5. Supervisiertes Lernen beinhaltet das Training eines Modells an etikettierten Daten, bei denen jedes Beispiel einen Eingabewert und einen zugehörigen korrekten Ausgangswert hat.
6. Unsupervisiertes Lernen verwendet nicht-etikettierte Daten, sodass das Modell versteckte Muster oder Gruppierungen ohne explizite Anweisungen entdecken kann.
7. Verstärkendes Lernen (RL) trainiert Agenten, Entscheidungen zu treffen, indem erwünschte Verhaltensweisen belohnt und unerwünschte bestraft werden.
8. Generative Modelle lernen, neue Daten zu erzeugen, die ihren Trainingsbeispielen ähneln (z. B. Text, Bilder).
9. Diskriminative Modelle konzentrieren sich darauf, Eingaben in Kategorien zu klassifizieren oder spezifische Ergebnisse vorherzusagen.
10. Transfer Learning ermöglicht es einem Modell, das auf eine Aufgabe trainiert wurde, auf eine verwandte Aufgabe wiederverwendet oder feinabgestimmt zu werden.
11. GPT (Generative Pre-trained Transformer) ist eine Familie von großen Sprachmodellen, die von OpenAI entwickelt wurden und menschenähnlichen Text generieren können.
12. ChatGPT ist eine interaktive Variante von GPT, die für Konversations- und Anweisungsaufgaben feinabgestimmt ist.
13. Die Transformer-Architektur wurde im Papier „Attention Is All You Need“eingeführt und revolutionierte die Verarbeitung natürlicher Sprache, indem sie auf Aufmerksamkeitsmechanismen setzte.
14. Selbstaufmerksamkeitsmechanismen ermöglichen es dem Modell, verschiedene Teile der Eingabesequenz beim Erstellen einer Ausgaberepräsentation zu gewichten.
15. Positionale Kodierung in Transformern hilft dem Modell, die Reihenfolge der Tokens in einer Sequenz zu identifizieren.
16. Pre-training ist die Anfangsphase, in der ein Modell allgemeine Merkmale aus großen Datensätzen lernt, bevor es auf spezifische Aufgaben feinabgestimmt wird.

17. Feinabstimmung ist der Prozess, ein vorab trainiertes Modell zu nehmen und es an eine engere Aufgabe mit einem kleineren, aufgabenbezogenen Datensatz anzupassen.
18. Sprachmodellierung ist die Aufgabe, das nächste Token (Wort oder Subwort) in einer Sequenz vorherzusagen, was für GPT-ähnliche Modelle grundlegend ist.
19. Zero-shot Learning ermöglicht es einem Modell, Aufgaben ohne explizite Trainingsbeispiele zu bewältigen, indem es auf gelerntes allgemeines Wissen zurückgreift.
20. Few-shot Learning nutzt eine begrenzte Anzahl von aufgabenbezogenen Beispielen, um die Vorhersagen oder Verhaltensweisen des Modells zu leiten.
21. RLHF (Reinforcement Learning from Human Feedback) wird verwendet, um die Modellausgaben mit menschlichen Vorlieben und Werten in Einklang zu bringen.
22. Menschliches Feedback kann Ranglisten oder Etiketten umfassen, die die Erzeugung des Modells zu gewünschteren Antworten leiten.
23. Prompt Engineering ist die Kunst, Eingabeabfragen oder Anweisungen zu gestalten, um große Sprachmodelle effektiv zu leiten.
24. Kontextfenster bezieht sich auf die maximale Textmenge, die das Modell auf einmal verarbeiten kann; GPT-Modelle haben eine begrenzte Kontextlänge.
25. Inferenz ist die Phase, in der ein trainiertes Modell Vorhersagen trifft oder Ausgaben generiert, basierend auf neuen Eingaben.
26. Parameteranzahl ist ein Schlüsselfaktor für die Modellkapazität; größere Modelle können komplexere Muster erfassen, benötigen aber mehr Rechenleistung.
27. Modellkomprimierungstechniken (z. B. Pruning, Quantisierung) reduzieren die Größe eines Modells und beschleunigen die Inferenz mit minimalem Genauigkeitsverlust.
28. Aufmerksamkeitsköpfe in Transformern verarbeiten verschiedene Aspekte der Eingabe parallel, was die Darstellungsfähigkeit verbessert.
29. Masked Language Modeling (z. B. in BERT) beinhaltet das Vorhersagen fehlender Tokens in einem Satz, was dem Modell hilft, den Kontext zu lernen.
30. Kausales Sprachmodellieren (z. B. in GPT) beinhaltet das Vorhersagen des nächsten Tokens basierend auf allen vorherigen Tokens.
31. Encoder-Decoder-Architektur (z. B. T5) verwendet ein Netzwerk, um die Eingabe zu kodieren, und ein anderes, um sie in eine Zielsequenz zu dekodieren.
32. Faltungsneuronale Netze (CNNs) sind hervorragend für die Verarbeitung von gitterartigen Daten (z. B. Bilder) durch Faltungsschichten geeignet.
33. Rekurrente neuronale Netze (RNNs) verarbeiten sequentielle Daten, indem sie versteckte Zustände entlang der Zeitpunkte weitergeben, können jedoch mit langfristigen Abhängigkeiten kämpfen.

34. Long Short-Term Memory (LSTM) und GRU sind RNN-Varianten, die entwickelt wurden, um langfristige Abhängigkeiten besser zu erfassen.

35. Batch-Normalisierung hilft, das Training zu stabilisieren, indem sie die Ausgaben der Zwischenschichten normalisiert.

36. Dropout ist eine Regularisierungstechnik, die zufällig „Neuronen“während des Trainings entfernt, um Überanpassung zu verhindern.

37. Optimierer-Algorithmen wie stochastischer Gradientenabstieg (SGD), Adam und RMSProp aktualisieren die Modellparameter basierend auf Gradienten.

38. Lernrate ist ein Hyperparameter, der bestimmt, wie drastisch die Gewichte während des Trainings aktualisiert werden.

39. Hyperparameter (z. B. Batchgröße, Anzahl der Schichten) sind Konfigurationseinstellungen, die vor dem Training gewählt werden, um zu steuern, wie das Lernen abläuft.

40. Modellüberanpassung tritt auf, wenn ein Modell die Trainingsdaten zu gut lernt und nicht auf neue Daten generalisiert.

41. Regularisierungstechniken (z. B. L2-Gewichtsverfall, Dropout) helfen, die Überanpassung zu reduzieren und die Generalisierung zu verbessern.

42. Validierungsmenge wird verwendet, um Hyperparameter anzupassen, während die Testmenge die endgültige Leistung des Modells bewertet.

43. Kreuzvalidierung teilt Daten in mehrere Teilmengen auf, trainiert und validiert systematisch, um eine robustere Leistungsbewertung zu erhalten.

44. Gradient Exploding und Vanishing Probleme treten in tiefen Netzen auf, was das Training instabil oder ineffektiv macht.

45. Residualverbindungen (Übersprungverbindungen) in Netzen wie ResNet helfen, verschwindende Gradienten zu mildern, indem sie Datenpfade abkürzen.

46. Skalierungsgesetze deuten darauf hin, dass das Erhöhen der Modellgröße und der Daten im Allgemeinen zu einer besseren Leistung führt.

47. Recheneffizienz ist entscheidend; das Training großer Modelle erfordert optimierte Hardware (GPUs, TPUs) und Algorithmen.

48. Ethische Überlegungen umfassen Vorurteile, Fairness und potenziellen Schaden –ML-Modelle müssen sorgfältig getestet und überwacht werden.

49. Datenaugmentation erweitert künstlich Trainingsdatensätze, um die Robustheit des Modells zu verbessern (besonders bei Bild- und Sprachaufgaben).

50. Datenvorverarbeitung (z. B. Tokenisierung, Normalisierung) ist für ein effektives Modelltraining unerlässlich.

51. Tokenisierung teilt Text in Tokens (Wörter oder Subwörter) auf, die grundlegenden Einheiten, die von Sprachmodellen verarbeitet werden.

52. Vektorembeddings stellen Tokens oder Konzepte als numerische Vektoren dar, die semantische Beziehungen bewahren.

53. Positionale Einbettungen fügen Informationen über die Position jedes Tokens hinzu, um einem Transformer zu helfen, die Reihenfolge der Sequenz zu verstehen.

54. Aufmerksamkeitsgewichte zeigen, wie ein Modell den Fokus auf verschiedene Teile der Eingabe verteilt.

55. Beam Search ist eine Dekodierungsstrategie in Sprachmodellen, die mehrere Kandidatenausgaben auf jeder Stufe behält, um die beste Gesamtsequenz zu finden.

56. Greedy Search wählt das wahrscheinlichste Token auf jeder Stufe aus, kann jedoch zu suboptimalen Endausgaben führen.

57. Temperatur bei der Stichprobenentnahme passt die Kreativität der Sprachgenerierung an: höhere Temperatur = mehr Zufälligkeit.

58. Top-k- und Top-p (Nucleus)-Stichprobenmethoden beschränken die Kandidaten-Tokens auf die k wahrscheinlichsten oder eine kumulative Wahrscheinlichkeit p, um Vielfalt und Kohärenz auszugleichen.

59. Perplexität misst, wie gut ein Wahrscheinlichkeitsmodell eine Probe vorhersagt; niedrigere Perplexität deutet auf eine bessere Vorhersageleistung hin.

60. Präzision und Rückruf sind Metriken für Klassifikationsaufgaben, die sich auf Richtigkeit bzw. Vollständigkeit konzentrieren.

61. F1-Wert ist der harmonische Mittelwert von Präzision und Rückruf, der beide Metriken in einen einzigen Wert ausgleicht.

62. Genauigkeit ist der Anteil der korrekten Vorhersagen, kann jedoch in unausgeglichenen Datensätzen irreführend sein.

63. Fläche unter der ROC-Kurve (AUC) misst die Leistung eines Klassifikators über verschiedene Schwellenwerte hinweg.

64. Verwirrungsmatrix zeigt die Anzahl der wahren Positiven, falschen Positiven, falschen Negativen und wahren Negativen.

65. Unsicherheitsabschätzungsmethoden (z. B. Monte Carlo Dropout) messen, wie sicher ein Modell in seinen Vorhersagen ist.

66. Aktives Lernen fragt neue Dateneinträge ab, bei denen das Modell am wenigsten sicher ist, um die Dateneffizienz zu verbessern.

67. Online-Lernen aktualisiert das Modell schrittweise, wenn neue Daten eintreffen, anstatt von Grund auf neu zu trainieren.

68. Evolutionäre Algorithmen und genetische Algorithmen optimieren Modelle oder Hyperparameter mit bio-inspirierter Mutation und Selektion.

69. Bayesianische Methoden integrieren Vorwissen und aktualisieren Überzeugungen mit eingehenden Daten, was für die Unsicherheitsquantifizierung nützlich ist.

70. Ensemble-Methoden (z. B. Random Forest, Gradient Boosting) kombinieren mehrere Modelle, um die Leistung und Stabilität zu verbessern.

71. Bagging (Bootstrap Aggregating) trainiert mehrere Modelle an verschiedenen Teilmengen der Daten und gleicht dann ihre Vorhersagen aus.

72. Boosting trainiert iterativ neue Modelle, um Fehler zu korrigieren, die von zuvor trainierten Modellen gemacht wurden.

73. Gradient Boosted Decision Trees (GBDTs) sind leistungsfähig für strukturierte Daten und übertreffen oft einfache neuronale Netze.

74. Autoregressive Modelle sagen den nächsten Wert (oder Token) basierend auf vorherigen Ausgaben in einer Sequenz vor.

75. Autoencoder ist ein neuronales Netz, das entwickelt wurde, um Daten in eine latente Darstellung zu kodieren und dann zurück zu dekodieren, wobei es komprimierte Datenrepräsentationen lernt.

76. Variational Autoencoder (VAE) fügt eine probabilistische Wendung hinzu, um neue Daten zu generieren, die dem Trainingsdatensatz ähneln.

77. Generative Adversarial Network (GAN) stellt einen Generator gegen einen Diskriminator, um realistische Bilder, Text oder andere Daten zu produzieren.

78. Selbstüberwachtes Lernen nutzt große Mengen an nicht-etikettierten Daten, indem es künstliche Trainingsaufgaben erstellt (z. B. Vorhersagen fehlender Teile).

79. Foundation Models sind große vorab trainierte Modelle, die an eine Vielzahl von Downstream-Aufgaben angepasst werden können.

80. Multimodales Lernen integriert Daten aus mehreren Quellen (z. B. Text, Bilder, Audio), um reichere Darstellungen zu erstellen.

81. Datenetikettierung ist oft der zeitaufwendigste Teil des ML, der sorgfältige Annotationen für Genauigkeit erfordert.

82. Edge Computing bringt ML-Inferenz näher an die Datenquelle, um Latenz und Bandbreitenverbrauch zu reduzieren.

83. Federated Learning trainiert Modelle über dezentrale Geräte oder Server, die lokale Datensätze halten, ohne diese auszutauschen.

84. Datenschutz-ML umfasst Techniken wie differenzielle Privatsphäre und homomorphe Verschlüsselung, um sensible Daten zu schützen.

85. Erklärbare KI (XAI) zielt darauf ab, die Entscheidungen komplexer Modelle für Menschen interpretierbarer zu machen.

86. Vorurteile und Fairness im ML benötigen sorgfältige Überwachung, da Modelle gesellschaftliche Vorurteile unabsichtlich lernen und verstärken können.

87. Konzeptdrift tritt auf, wenn sich die statistischen Eigenschaften der Zielvariable im Laufe der Zeit ändern, was die Modellleistung beeinträchtigt.

88. AB-Test vergleicht zwei oder mehrere Versionen eines Modells, um zu sehen, welche in einer realen Umgebung besser abschneidet.

89. GPU-Beschleunigung nutzt paralleles Rechnen auf Grafikkarten, um das ML-Training erheblich zu beschleunigen.

90. TPUs (Tensor Processing Units) sind spezialisierte Hardwarebeschleuniger von Google für effiziente Deep-Learning-Arbeitslasten.

91. Open-Source-Frameworks (z. B. TensorFlow, PyTorch) bieten Bausteine und Tools für die Entwicklung von ML-Modellen.

92. Modellbereitstellung ist die Praxis, trainierte Modelle so zu implementieren, dass sie Echtzeit- oder Batch-Vorhersagen durchführen können.

93. Skalierbarkeit ist entscheidend, um große Datensätze oder hohen Verkehr zu bewältigen, was verteilte Trainings- und Inferenzstrategien erfordert.

94. MLOps kombiniert ML-Entwicklung mit Operationspraktiken, die sich auf Reproduzierbarkeit, Testen und kontinuierliche Integration konzentrieren.

95. Versionskontrolle für Daten und Modelle stellt sicher, dass Experimente konsistent verfolgt und zusammen gearbeitet wird.

96. Implementierungsstrategien (z. B. Container, Microservices) organisieren, wie Modelle im großen Stil verpackt und bereitgestellt werden.

97. Überwachung verfolgt die Modellleistung nach der Implementierung, um Verschlechterungen oder Anomalien zu überwachen.

98. Retraining und Modellaktualisierungen halten Modelle aktuell, wenn neue Daten und sich ändernde Bedingungen auftreten.

99. Zeitkomplexität (O-Notation) misst, wie sich die Laufzeit eines Algorithmus mit der Eingabegröße skaliert; O(1) bezeichnet konstante Zeit.

100. Zukunft des ML verspricht zunehmend raffinierte und allgemeine Modelle, muss jedoch ethische, soziale und umweltbezogene Überlegungen berücksichtigen.