机器学习、深度学习与 GPT

1. 机器学习（ML）是计算机科学的一个领域，使系统能够从数据中学习并提高性能，而无需显式编程。
2. 深度学习（DL）是 ML 的一个子领域，利用多层神经网络来模拟数据中的复杂模式。
3. 神经网络是受人脑启发的计算模型，由相互连接的节点（神经元）组成，按层次处理信息。
4. 训练数据是用于教导机器学习模型如何执行任务的标记或未标记数据集。
5. 监督学习涉及在标记数据上训练模型，每个示例都有一个输入和相关的正确输出。
6. 无监督学习使用未标记数据，允许模型在没有明确指示的情况下发现隐藏的模式或分组。
7. 强化学习（RL）通过奖励所需行为和惩罚不良行为来训练代理，使其做出决策。
8. 生成模型学习生成与其训练示例相似的新数据（例如，文本、图像）。
9. 判别模型专注于将输入分类到类别中或预测特定的结果。
10. 迁移学习允许在一个任务上训练的模型被重用或在相关任务上进行微调。
11. GPT（生成预训练变压器）是由 OpenAI 开发的一系列大型语言模型，可以生成类似人类的文本。
12. ChatGPT 是 GPT 的交互式变体，专门用于对话和遵循指令的任务。
13. 变压器架构在论文“Attention Is All You Need”中引入，通过依赖注意力机制，革命性地改进了自然语言处理。
14. 自注意力机制使模型在构建输出表示时对输入序列的不同部分进行加权。
15. 位置编码在变压器中帮助模型识别序列中的标记顺序。
16. 预训练是初始阶段，模型在大规模数据上学习一般特征，然后在特定任务上进行微调。
17. 微调是将预训练模型取出并使用较小的、特定于任务的数据集进行适应。
18. 语言建模是预测序列中下一个标记（单词或子单词）的任务，是 GPT 等模型的基础。
19. 零样本学习允许模型在没有明确训练示例的情况下处理任务，依赖于学到的一般知识。
20. 少样本学习利用有限数量的特定于任务的示例来指导模型的预测或行为。
21. RLHF（基于人类反馈的强化学习）用于使模型输出与人类偏好和价值一致。
22. 人类反馈可以包括排名或标签，指导模型的生成以获得更多的期望响应。
23. 提示工程是指精心设计输入查询或指令，以有效引导大型语言模型。
24. 上下文窗口是模型一次可以处理的最大文本量；GPT 模型具有有限的上下文长度。
25. 推理是训练模型在给定新输入时进行预测或生成输出的阶段。
26. 参数计数是模型容量的关键因素；较大的模型可以捕捉更复杂的模式，但需要更多的计算。

27. 模型压缩技术（例如，修剪、量化）在保持最小精度损失的情况下减少模型的大小并加快推理。
28. 变压器中的注意力头并行处理输入的不同方面，提高了表示能力。
29. 遮掩语言建模（例如，在 BERT 中）涉及预测句子中的缺失标记，帮助模型学习上下文。
30. 因果语言建模（例如，在 GPT 中）涉及基于所有先前标记预测下一个标记。
31. 编码器-解码器架构（例如，T5）使用一个网络对输入进行编码，另一个网络将其解码为目标序列。
32. 卷积神经网络（CNNs）在处理网格状数据（例如，图像）方面表现出色，通过卷积层。
33. 循环神经网络（RNNs）通过在时间步骤之间传递隐藏状态来处理序列数据，尽管它们可能会在长期依赖性方面遇到困难。
34. 长短期记忆（LSTM）和 GRU 是设计用于更好地捕捉长距离依赖性的 RNN 变体。
35. 批量归一化通过归一化中间层输出来稳定训练。
36. 丢弃是一种正则化技术，在训练过程中随机“丢弃”神经元，以防止过拟合。
37. 优化器算法（例如，随机梯度下降（SGD）、Adam 和 RMSProp）根据梯度更新模型参数。
38. 学习率是一个超参数，确定在训练过程中权重更新的剧烈程度。
39. 超参数（例如，批量大小、层数）是在训练前选择的配置设置，用于控制学习的进展。
40. 模型过拟合发生在模型过于熟悉训练数据，无法推广到新数据的情况下。
41. 正则化技术（例如，L2 权重衰减、丢弃）有助于减少过拟合并提高泛化能力。
42. 验证集用于调整超参数，而测试集评估模型的最终性能。
43. 交叉验证将数据分成多个子集，系统地进行训练和验证，以获得更健壮的性能估计。
44. 梯度爆炸和消失问题发生在深度网络中，使训练不稳定或无效。
45. 残差连接（跳跃连接）在 ResNet 等网络中帮助缓解梯度消失问题，通过数据路径的快捷方式。
46. 规模定律表明，增加模型大小和数据通常会带来更好的性能。
47. 计算效率至关重要；训练大型模型需要优化的硬件（GPUs、TPUs）和算法。
48. 伦理考虑包括偏见、公平性和潜在的伤害——ML 模型必须仔细测试和监控。
49. 数据增强通过人工扩展训练数据集来提高模型的健壮性（特别是在图像和语音任务中）。
50. 数据预处理（例如，标记化、归一化）对于有效的模型训练至关重要。
51. 标记化将文本拆分为标记（单词或子单词），这是语言模型处理的基本单位。
52. 向量嵌入将标记或概念表示为数值向量，保留语义关系。
53. 位置嵌入向模型添加有关每个标记位置的信息，以帮助变压器理解序列顺序。
54. 注意力权重揭示了模型如何在输入的不同部分之间分配注意力。

55. 束搜索是语言模型的解码策略，在每一步保留多个候选输出，以找到最佳的总序列。

56. 贪婪搜索在每一步选择最有可能的标记，但可能导致次优的最终输出。

57. 采样中的温度调整语言生成的创造性：较高的温度 = 更多的随机性。

58. Top-k 和 Top-p（Nucleus）采样方法将候选标记限制为 k 个最可能的或累积概率 p，平衡多样性和连贯性。

59. 困惑度衡量概率模型预测样本的能力；较低的困惑度表示更好的预测性能。

60. 精确度和召回率是分类任务的指标，分别关注正确性和完整性。

61. F1 分数是精确度和召回率的调和平均数，将两个指标平衡为一个值。

62. 精度是正确预测的分数，但在不平衡数据集中可能会误导。

63. ROC 曲线下的面积（AUC）衡量分类器在各种阈值下的性能。

64. 混淆矩阵显示真正例、假正例、假负例和真负例的计数。

65. 不确定性估计方法（例如，蒙特卡罗丢弃）衡量模型对其预测的信心程度。

66. 主动学习涉及查询模型最不确定的新数据示例，提高数据效率。

67. 在线学习在新数据到来时逐步更新模型，而不是从头开始重新训练。

68. 进化算法和遗传算法使用生物启发的突变和选择来优化模型或超参数。

69. 贝叶斯方法结合先验知识并根据新数据更新信念，有助于不确定性量化。

70. 集成方法（例如，随机森林、梯度提升）结合多个模型以提高性能和稳定性。

71. 装袋（Bootstrap Aggregating）在不同的数据子集上训练多个模型，然后平均它们的预测。

72. 提升方法逐步训练新模型以纠正先前训练模型所犯的错误。

73. 梯度提升决策树（GBDTs）在结构化数据中非常强大，通常优于简单的神经网络。

74. 自回归模型基于序列中的先前输出预测下一个值（或标记）。

75. 自编码器是一种神经网络，设计用于将数据编码为潜在表示，然后将其解码回去，学习压缩数据表示。

76. 变分自编码器（VAE）引入了概率性质，以生成与训练集相似的新数据。

77. 生成对抗网络（GAN）将生成器与判别器对抗，生成现实的图像、文本或其他数据。

78. 自监督学习利用大量未标记数据，通过创建人工训练任务（例如，预测缺失部分）。

79. 基础模型是大型预训练模型，可以适应各种下游任务。

80. 多模态学习整合来自多个来源（例如，文本、图像、音频）的数据，以创建更丰富的表示。

81. 数据标注通常是 ML 中最耗时的部分，需要仔细注解以确保准确性。

82. 边缘计算将 ML 推理带到数据源附近，减少延迟和带宽使用。

83. 联邦学习在分布式设备或服务器上训练模型，这些设备或服务器持有本地数据样本，而不交换它们。
84. 隐私保护 ML 包括差分隐私和同态加密等技术，以保护敏感数据。
85. 可解释人工智能（XAI）旨在使复杂模型的决策更易于人类理解。
86. 偏见和公平性在 ML 中需要仔细监督，因为模型可能会无意中学习和放大社会偏见。
87. 概念漂移发生在目标变量的统计属性随时间变化时，影响模型性能。
88. AB 测试比较两个或多个模型版本，以查看哪个在实际环境中表现更好。
89. GPU 加速利用图形卡上的并行计算，显著加快 ML 训练。
90. TPUs（张量处理单元）是 Google 为高效深度学习工作负载设计的专用硬件加速器。
91. 开源框架（例如，TensorFlow、PyTorch）提供 ML 模型开发的构建块和工具。
92. 模型服务是将训练模型部署的实践，以便它们可以处理实时或批量预测。
93. 可扩展性对于处理大型数据集或重负载至关重要，需要分布式训练和推理策略。
94. MLOps 将 ML 开发与操作实践结合起来，专注于可重复性、测试和持续集成。
95. 数据和模型的版本控制确保了实验的一致性和协作。
96. 部署策略（例如，容器、微服务）组织模型的打包和大规模服务方式。
97. 监控在部署后跟踪模型性能，监视性能下降或异常。
98. 重新训练和模型更新保持模型在新数据和变化条件下的最新状态。
99. 时间复杂度（O 表示法）衡量算法的运行时间如何随着输入大小的增加而扩展；O(1) 表示常数时间。
100. ML 的未来承诺越来越复杂和通用的模型，但必须解决伦理、社会和环境问题。