ML, DL et GPT

1. L’apprentissage automatique (ML) est un domaine de l’informatique qui permet aux systèmes d’apprendre à partir des données et d’améliorer leurs performances sans programmation explicite.
2. L’apprentissage profond (DL) est un sous-domaine de ML qui utilise des réseaux neuronaux à couches multiples pour modéliser des motifs complexes dans les données.
3. Les réseaux neuronaux sont des modèles computationnels inspirés du cerveau humain, composés de nœuds interconnectés (neurones) qui traitent les informations par couches.
4. Les données d’entraînement sont l’ensemble de données étiquetées ou non étiquetées utilisé pour enseigner à un modèle d’apprentissage automatique comment effectuer une tâche.
5. L’apprentissage supervisé consiste à entraîner un modèle sur des données étiquetées, où chaque exemple a une entrée et une sortie correcte associée.
6. L’apprentissage non supervisé utilise des données non étiquetées, permettant au modèle de découvrir des motifs cachés ou des regroupements sans instruction explicite.
7. L’apprentissage par renforcement (RL) entraîne des agents à prendre des décisions en récompensant les comportements souhaités et en pénalisant les comportements indésirables.
8. Les modèles génératifs apprennent à produire de nouvelles données similaires à leurs exemples d’entraînement (par exemple, texte, images).
9. Les modèles discriminatifs se concentrent sur la classification des entrées en catégories ou la prédiction d’issues spécifiques.
10. L’apprentissage par transfert permet à un modèle entraîné sur une tâche d’être réutilisé ou affiné sur une tâche connexe.
11. GPT (Generative Pre-trained Transformer) est une famille de grands modèles de langage développés par OpenAI qui peuvent générer du texte ressemblant à celui des humains.
12. ChatGPT est une variante interactive de GPT, affinée pour la conversation et les tâches de suivi des instructions.
13. L’architecture Transformer a été introduite dans l’article « Attention Is All You Need », révolutionnant le traitement du langage naturel en s’appuyant sur des mécanismes d’attention.
14. Les mécanismes d’auto-attention permettent au modèle de pondérer différentes parties de la séquence d’entrée lors de la construction d’une représentation de sortie.
15. Le codage positionnel dans les Transformers aide le modèle à identifier l’ordre des jetons dans une séquence.
16. Le pré-entraînement est la phase initiale où un modèle apprend des caractéristiques générales à partir de grandes quantités de données avant d’être affiné sur des tâches spécifiques.

17. L’affinage est le processus consistant à prendre un modèle pré-entraîné et à l’adapter à une tâche plus spécifique en utilisant un ensemble de données plus petit et spécifique à la tâche.

18. La modélisation du langage consiste à prédire le jeton suivant (mot ou sous-mot) dans une séquence, fondamentale pour les modèles de type GPT.

19. L’apprentissage zéro-shot permet à un modèle de gérer des tâches sans exemples d’entraînement explicites, en s’appuyant sur des connaissances générales apprises.

20. L’apprentissage à quelques coups utilise un nombre limité d’exemples spécifiques à la tâche pour guider les prédictions ou comportements du modèle.

21. RLHF (Reinforcement Learning from Human Feedback) est utilisé pour aligner les sorties du modèle avec les préférences et valeurs humaines.

22. Les retours humains peuvent inclure des classements ou des étiquettes qui guident la génération du modèle vers des réponses plus souhaitées.

23. L’ingénierie des invites est l’art de formuler des requêtes d’entrée ou des instructions pour guider efficacement les grands modèles de langage.

24. La fenêtre de contexte fait référence à la quantité maximale de texte que le modèle peut traiter à la fois; les modèles GPT ont une longueur de contexte limitée.

25. L’inférence est l’étape où un modèle entraîné fait des prédictions ou génère des sorties à partir de nouvelles entrées.

26. Le nombre de paramètres est un facteur clé de la capacité du modèle; les plus grands modèles peuvent capturer des motifs plus complexes mais nécessitent plus de calculs.

27. Les techniques de compression de modèle (par exemple, élagage, quantification) réduisent la taille du modèle et accélèrent l’inférence avec une perte d’exactitude minimale.

28. Les têtes d’attention dans les Transformers traitent différents aspects de l’entrée en parallèle, améliorant la puissance représentative.

29. La modélisation du langage masqué (par exemple, dans BERT) consiste à prédire les jetons manquants dans une phrase, aidant le modèle à apprendre le contexte.

30. La modélisation du langage causal (par exemple, dans GPT) consiste à prédire le jeton suivant en fonction de tous les jetons précédents.

31. L’architecture Encoder-Decoder (par exemple, T5) utilise un réseau pour encoder l’entrée et un autre pour la décoder en une séquence cible.

32. Les réseaux de neurones convolutifs (CNN) excellent dans le traitement des données en grille (par exemple, images) via des couches convolutives.

33. Les réseaux de neurones récurrents (RNN) traitent les données séquentielles en passant des états cachés le long des étapes temporelles, bien qu'ils puissent avoir du mal avec les dépendances à long terme.

34. La mémoire à long terme (LSTM) et GRU sont des variantes de RNN conçues pour mieux capturer les dépendances à long terme.

35. La normalisation par lots aide à stabiliser l'entraînement en normalisant les sorties des couches intermédiaires.

36. Le dropout est une technique de régularisation qui « supprime » aléatoirement des neurones pendant l'entraînement pour prévenir le surapprentissage.

37. Les algorithmes d'optimisation comme la descente de gradient stochastique (SGD), Adam et RMSProp mettent à jour les paramètres du modèle en fonction des gradients.

38. Le taux d'apprentissage est un hyperparamètre qui détermine à quel point les poids sont mis à jour pendant l'entraînement.

39. Les hyperparamètres (par exemple, la taille du lot, le nombre de couches) sont des paramètres de configuration choisis avant l'entraînement pour contrôler comment l'apprentissage se déroule.

40. Le surapprentissage du modèle se produit lorsque le modèle apprend trop bien les données d'entraînement, échouant à généraliser à de nouvelles données.

41. Les techniques de régularisation (par exemple, la décroissance des poids L2, le dropout) aident à réduire le surapprentissage et à améliorer la généralisation.

42. L'ensemble de validation est utilisé pour ajuster les hyperparamètres, tandis que l'ensemble de test évalue la performance finale du modèle.

43. La validation croisée divise les données en plusieurs sous-ensembles, en entraînant et en validant systématiquement pour obtenir une estimation de performance plus robuste.

44. Les problèmes d'explosion et de disparition des gradients surviennent dans les réseaux profonds, rendant l'entraînement instable ou inefficace.

45. Les connexions résiduelles (connexions de saut) dans des réseaux comme ResNet atténuent les gradients disparus en raccourcissant les chemins de données.

46. Les lois d'échelle suggèrent que l'augmentation de la taille du modèle et des données conduit généralement à de meilleures performances.

47. L'efficacité de calcul est cruciale; l'entraînement de grands modèles nécessite un matériel optimisé (GPU, TPU) et des algorithmes.

48. Les considérations éthiques incluent le biais, l'équité et le préjudice potentiel—les modèles ML doivent être soigneusement testés et surveillés.

49. L’augmentation des données étend artificiellement les ensembles de données d’entraînement pour améliorer la robustesse du modèle (surtout dans les tâches d’image et de parole).

50. Le prétraitement des données (par exemple, la tokenisation, la normalisation) est essentiel pour un entraînement de modèle efficace.

51. La tokenisation divise le texte en jetons (mots ou sous-mots), les unités fondamentales traitées par les modèles de langage.

52. Les embeddings vectoriels représentent les jetons ou concepts sous forme de vecteurs numériques, préservant les relations sémantiques.

53. Les embeddings positionnels ajoutent des informations sur la position de chaque jeton pour aider un Transformer à comprendre l’ordre de la séquence.

54. Les poids d’attention révèlent comment un modèle distribue son attention sur différentes parties de l’entrée.

55. La recherche de faisceau est une stratégie de décodage dans les modèles de langage qui conserve plusieurs sorties candidates à chaque étape pour trouver la meilleure séquence globale.

56. La recherche avide choisit le jeton le plus probable à chaque étape, mais peut conduire à des sorties finales sous-optimales.

57. La température dans l’échantillonnage ajuste la créativité de la génération de langage: une température plus élevée = plus de randomisation.

58. Les méthodes d’échantillonnage Top-k et Top-p (Noyau) restreignent les jetons candidats aux k plus probables ou à une probabilité cumulative p, équilibrant diversité et cohérence.

59. La perplexité mesure à quel point un modèle de probabilité prédit un échantillon; une perplexité plus faible indique une meilleure performance prédictive.

60. La précision et le rappel sont des métriques pour les tâches de classification, se concentrant sur la correction et la complétude, respectivement.

61. Le score F1 est la moyenne harmonique de la précision et du rappel, équilibrant les deux métriques en une seule valeur.

62. L’exactitude est la fraction de prédictions correctes, mais elle peut être trompeuse dans les ensembles de données déséquilibrés.

63. L’aire sous la courbe ROC (AUC) mesure la performance d’un classificateur à travers divers seuils.

64. La matrice de confusion montre les comptes des vrais positifs, faux positifs, faux négatifs et vrais négatifs.

65. Les méthodes d’estimation de l’incertitude (par exemple, le dropout de Monte Carlo) évaluent à quel point un modèle est confiant dans ses prédictions.

66. L’apprentissage actif consiste à interroger de nouveaux exemples de données sur lesquels le modèle est le moins confiant, améliorant l’efficacité des données.

67. L’apprentissage en ligne met à jour le modèle de manière incrémentielle à mesure que de nouvelles données arrivent, plutôt que de réentraîner à partir de zéro.

68. Les algorithmes évolutionnaires et les algorithmes génétiques optimisent les modèles ou les hyperparamètres en utilisant des mutations et sélections inspirées de la biologie.

69. Les méthodes bayésiennes intègrent des connaissances a priori et mettent à jour les croyances avec les données entrantes, utiles pour la quantification de l’incertitude.

70. Les méthodes d’ensemble (par exemple, Random Forest, Gradient Boosting) combinent plusieurs modèles pour améliorer les performances et la stabilité.

71. Le bagging (Bootstrap Aggregating) entraîne plusieurs modèles sur différents sous-ensembles des données, puis fait la moyenne de leurs prédictions.

72. Le boosting entraîne itérativement de nouveaux modèles pour corriger les erreurs commises par les modèles précédemment entraînés.

73. Les arbres de décision boostés par gradient (GBDT) sont puissants pour les données structurées, souvent surpassant les réseaux neuronaux simples.

74. Les modèles autorégressifs prédisent la valeur suivante (ou jeton) en fonction des sorties précédentes dans une séquence.

75. Un autoencodeur est un réseau de neurones conçu pour encoder les données en une représentation latente, puis les décoder à nouveau, apprenant des représentations de données compressées.

76. Le variational autoencoder (VAE) introduit une torsion probabiliste pour générer de nouvelles données qui ressemblent à l’ensemble d’entraînement.

77. Le réseau antagoniste génératif (GAN) oppose un générateur à un discriminateur, produisant des images, du texte ou d’autres données réalistes.

78. L’apprentissage auto-supervisé tire parti de grandes quantités de données non étiquetées en créant des tâches d’entraînement artificielles (par exemple, prédire des parties manquantes).

79. Les modèles de base sont de grands modèles pré-entraînés qui peuvent être adaptés à une large gamme de tâches en aval.

80. L’apprentissage multimodal intègre des données provenant de multiples sources (par exemple, texte, images, audio) pour créer des représentations plus riches.

81. L’étiquetage des données est souvent la partie la plus chronophage de ML, nécessitant une annotation soigneuse pour l’exactitude.

82. Le calcul de bord rapproche l’inférence ML de la source de données, réduisant la latence et l’utilisation de la bande passante.

83. L’apprentissage fédéré entraîne des modèles à travers des dispositifs ou serveurs décentralisés contenant des échantillons de données locaux, sans les échanger.

84. L’apprentissage ML préservant la vie privée inclut des techniques comme la confidentialité différentielle et le chiffrement homomorphique pour protéger les données sensibles.

85. L’IA explicable (XAI) vise à rendre les décisions des modèles complexes plus interprétables pour les humains.

86. Le biais et l’équité en ML nécessitent une surveillance attentive, car les modèles peuvent apprendre et amplifier les biais sociétaux de manière involontaire.

87. Le décalage conceptuel se produit lorsque les propriétés statistiques de la variable cible changent au fil du temps, affectant les performances du modèle.

88. Le test A/B compare deux ou plusieurs versions d’un modèle pour voir laquelle fonctionne le mieux dans un environnement réel.

89. L’accélération GPU exploite le calcul parallèle sur les cartes graphiques pour accélérer considérablement l’entraînement ML.

90. Les TPU (Tensor Processing Units) sont des accélérateurs matériels spécialisés par Google pour des charges de travail d’apprentissage profond efficaces.

91. Les frameworks open-source (par exemple, TensorFlow, PyTorch) fournissent des blocs de construction et des outils pour le développement de modèles ML.

92. La mise en service de modèles consiste à déployer des modèles entraînés pour qu’ils puissent gérer des prédictions en temps réel ou par lots.

93. La scalabilité est cruciale pour gérer de grands ensembles de données ou un trafic important, nécessitant des stratégies d’entraînement et d’inférence distribuées.

94. MLOps combine le développement ML avec les pratiques opérationnelles, se concentrant sur la reproductibilité, les tests et l’intégration continue.

95. Le contrôle de version pour les données et les modèles assure un suivi d’expérimentation cohérent et une collaboration.

96. Les stratégies de déploiement (par exemple, conteneurs, microservices) organisent la manière dont les modèles sont empaquetés et servis à grande échelle.

97. La surveillance suit les performances du modèle après le déploiement, surveillant les dégradations ou anomalies.

98. La réentraînement et les mises à jour de modèle gardent les modèles à jour à mesure que de nouvelles données et conditions changeantes apparaissent.

99. La complexité temporelle (notation O) mesure comment le temps d’exécution d’un algorithme évolue avec la taille de l’entrée; O(1) désigne un temps constant.

100. L’avenir de ML promet des modèles de plus en plus sophistiqués et généraux, mais doit aborder les considérations éthiques, sociales et environnementales.