人工智能期末复习

看到博弈论PPT第19页了

考纲

- **选择题** $(30分)$：15 个单选
  - 选择题范围为 PPT 内容 + 课本内容
- **计算、简答、推理题** $(70分)$：4 个大题，每个大题 $2 \sim 3$ 小问
  - 4 个大题分别为：机器学习、深度学习、深度学习、强化学习
  - 大题都是 PPT 内容

一、概述

- 人工智能的**定义**
  - 像人一样思考
  - 像人一样行动
  - 理性地思考
  - 理性地行动
- **图灵测试**
  - 一个人在**不接触对方**的情况下，通过一种特殊的方式**和对方进行一系列的问答。**如果在相当长时间内，他**无法根据这些问题判断对方是人还是计算机**，那么就可以认为这个计算机是智能的
- 人工智能的**发展历程**
  - 计算推理➡️知识表示➡️机器学习➡️深度学习
- 人工智能的**分类**
  - 弱人工智能
  - 强人工智能
  - 超人工智能

二、逻辑推理

2.1 命题逻辑

2.2 谓词逻辑

2.3 知识图谱推理

- **知识图谱**由**有向图**构成，被用来描述现实世界中实体及实体之间的关系
- 两个节点和连接边可表示为形如 `<left_node, relation, right_node>` 的三元组形式，也可表示为**一阶逻辑**（first order logic，FOL）的形式
- **归纳逻辑程序设计**（inductive logic programming，ILP）
  - ILP 使用**一阶谓词逻辑**进行知识表示，通过修改和扩充逻辑表达式对现有知识进行归纳，完成推理内容
  - **FOIL**（first order inductive learner）算法是 ILP 的代表性方法，通过**序贯覆盖**学习推理规则

$$FOIL\_Gain = \hat{m}_+ \cdot \left(\log_2 \left(\frac{\hat{m}_+}{\hat{m}_+ + m_-} \right) - \log_2 \left(\frac{m_+}{m_+ + m_-} \right) \right)$$

- **路径排序算法**（path ranking algorithm，PRA）
  - 将实体之间的关联路径作为特征，来学习目标关系的分类器
  - 属于**监督学习**
  - 流程
    - **特征抽取**：生成并选择路径特征集合。生成路径方法：随机游走（random walk）、BFS、DFS
    - **特征计算**：计算每个训练样例的特征值 $P(s \to t; \pi_j)$，表示从实体节点 s 出发，通过关系路径 π_j 达到实体节点 t 的概率。或表示是否存在这样一条路径，或表示路径出现的频次频率
    - **分类器训练**：根据训练样例特征值，为目标关系训练分类器。训练后可用于推理两个实体间是否存在目标关系

2.4 概率推理

- 贝叶斯网络
  - 用**有向无环图**表示
  - 贝叶斯网络中所有因素的联合分布等于所有节点的 $P(\text{节点}|\text{父节点})$ 的乘积

    $P(\text{多云}, \text{下雨}, \text{洒水车}, \text{路湿}) = P(\text{多云}) \cdot P(\text{洒水车}|\text{多云}) \cdot P(\text{下雨}|\text{多云}) \cdot P(\text{路湿}|\text{洒水车}, \text{下雨})$
- 马尔可夫网络
  - 用**无向图**表示

2.5 因果推理

- **辛普森悖论**
  - 在总体样本上成立的某种关系在分组样本上却不一定成立甚至相反
- 复杂数据中变量之间的关联关系
  - **因果关联**：数据中两个变量，如果一个变量是另一个变量的原因，则该两变量之间存在因果关联
  - **混淆关联**：数据中待研究的两个变量之间存在共同的原因变量（在辛普森悖论中，性别就是原因变量）
  - **选择关联**：数据中待研究的两个变量之间存在共同的结果变量
- 因果推理使用的模型
  - **潜在结果框架**
  - **结构因果模型**：使用因果图（有向无环图）

三、搜索求解

3.1 基本概念

- 搜索算法的**评价指标**
  - **完备性**：是否一定能找到一个解
  - **最优性**：找到的解是否是最优解

    - 时间复杂度和空间复杂度如何
    - 是否会陷入死循环
- 搜索算法分类
    - **有信息搜索（启发式搜索）**：贪婪最佳优先搜索，$A*$ 搜索（适用于单智能体的问题求解）
    - **无信息搜索**：深度优先搜索，广度优先搜索
    - **对抗搜索**：$Minimax$ 搜索，$alpha-beta$ 剪枝搜索（适用于多智能体的竞争环境）
    - **蒙特卡洛树搜索**：上限置信区间策略，蒙特卡洛树搜索（适用于无法完全遍历的大规模决策空间，找到近似最优解）

3.2 启发搜索

- **贪婪最佳优先搜索**
    - 评价函数$f(n)=$启发函数$h(n)$（完备但不是最优的）
- $A*$ **搜索**
    - 评价函数$f(n)=$起始节点到节点n代价$g(n)+$启发函数$h(n)$（完备且最优）
    - 良好的启发函数需要满足的性质为**可容性**和**一致性**
        - 可容性：启发函数不会过高估计从结点 n 到**终止结点**所应该付出的代价（即估计代价小于等于实际代价）
        - 一致性：对于任意节点 n，有 $h_n \leq c_{n,a,n'}+h_{n'}$，其中， $c_{n,a,n'}$ 表示节点 n 通过动作 a 到达其相应的后继节点 n' 的代价

3.3 对抗搜索

- **最小最大搜索（Minimax）**
    - 最大最小搜索是求解**对抗搜索问题**的基本算法
    - 该算法假设两名玩家在决策时总是理性地倾向于**最大化自己的得分**（最小化对方得分）
    - 执行了一个**完整的深度优先搜索**，时间复杂度过高

$$\text{minimax}(s)=\begin{cases}\text{utility}(s) & \text{if terminal\_test}(s)\\ \max_{a\in\text{actions}(s)}\text{minimax}(\text{result}(s,a)) & \text{if player}(s)=\text{MAX}\\ \min_{a\in\text{actions}(s)}\text{minimax}(\text{result}(s,a)) & \text{if player}(s)=\text{MIN}\end{cases}$$

- $alpha-beta$ **剪枝搜索**
    - 如果搜索树极大，则最大最小搜索的**开销巨大**，无法在合理时间内返回结果
    - 算法思想

$$\text{minimax}(s_0)=\max(\min(3,9,10),\min(2,x,y),\min(10,5,1))=\max(3,\min(2,x,y),1)=3$$

    - 为每个节点设置一个 α 值和一个 β 值，来判断该节点及其后继节点是否可被剪枝
    - 初始时 $[\alpha,\beta]=(-\infty,+\infty)$
        - 对于 MAX 节点，如果其后继节点得到收益大于当前的 α 值，则将 α 值更新为该收益
        - 对于 MIN 节点，如果其后继节点得到收益小于当前的 β 值，则将 β 值更新为该收益
    - 如果一个节点的 α 值和 β 值满足 $\alpha>\beta$ 的条件，则该节点尚未被访问的后继节点就会被剪枝，因而不会被智能体访问

3.4 蒙特卡洛搜索

- ϵ **- 贪心算法**
  - 在**探索与利用**之间进行平衡的搜索算法
    - 以 $1-\epsilon$ 的概率在过去 $t-1$ 次摇动赌博机摇臂的行动中**所得平均收益分数最高**的赌博机进行摇动
    - 以 ϵ 的概率**随机选择**一个赌博机进行摇动
- **上限置信区间算法（UCB1）**
  - 为每个动作的奖励期望计算一个估计范围，**优先采用估计范围上限较高的动作**
- **蒙特卡洛树搜索**
  - 找到**近似最优解**
  - 四个步骤
    - **选择**：算法从搜索树的根节点开始，向下递归选择子节点直到到达叶子结点或者到达还具有未被扩展的子节点的节点 L，向下递归选择的过程可由 UCB1 算法来实现，在递归选择过程中记录下每个节点被选择的次数和每个节点得到的奖励均值
    - **扩展**：如果节点 L 还不是一个终止节点，则随机扩展它的一个未被扩展过的后继边缘节点 M
    - **模拟**：从节点 M 出发，模拟扩展搜索树，直到找到一个终止节点
    - **反向传播**：用模拟所得结果回溯更新模拟路径中 M 及以上节点的奖励均值和被访问次数

四、机器学习

4.1 基本概念

- **机器学习种类**
  - **监督学习**
    - 从有标签的数据集中学习。即每个训练样本都有一个已知的结果（线性回归、逻辑回归、决策树、线性判别分析 LDA）
  - **无监督学习**
    - 从没有标签的数据集中学习（k 均值聚类、特征降维）
  - **半监督学习**
    - 在少量带标签和大量未带标签的数据集中学习
  - **强化学习**
    - 通过试错的方式，学习哪些行为可以获得奖励，哪些行为会导致惩罚

4.2 模型评估与参数估计

- **经验风险**（R_{emp}）：映射函数 f 在训练集上产生的损失

$$R_{\text{emp}} = \frac{1}{n}\sum_{i=1}^{n} Loss(y_i, f(x_i))$$

- **期望风险**（R）：真实风险、真实误差

$$R = \int_{x\times y} Loss(y, f(x))\, P(x,y)\, dxdy$$

    - 机器学习的目标是追求**期望风险最小化**
- 期望风险（R）与经验风险（R_{emp}）的关系

$$R \leq R_{\text{emp}} + err$$

- **结构风险最小化**
    - **防止过学习**，基于过学习时参数值通常都较大这一发现，在经验风险上加上表示模型复杂度的正则化项，在**最小化经验风险**与**降低模型复杂度**之间寻找平衡

$$R_{\text{srm}} = \frac{1}{n}\sum_{i=1}^{n} Loss(y_i, f(x_i)) + \lambda \mathcal{J}(f)$$

- 模型性能度量方法
    - **准确率**：$ACC = \frac{TP+TN}{P+N}$
    - **错误率**：$errorRate = \frac{FP+FN}{P+N}$
    - **精确率**：$precision = \frac{TP}{TP+FP}$
    - **召回率**：$recall = \frac{TP}{TP+FN}$
    - **综合分类率**：$F1\text{-}score = \frac{2 \times precision \times recall}{precision + recall}$
- 模型参数优化的两大思路
    - **频率学派**：**最大似然估计**（MLE）
        - 世界是**确定**的，有一个本体，这个本体的真值是不变的，我们的目标就是要找到这个真值或真值所在的范围
        - 存在唯一真值 θ
    - **贝叶斯学派**：**最大后验估计**（MAP）
        - 世界是**不确定**的，人们对世界先有一个预判，而后通过观测数据对这个预判做调整，我们的目标是要找到最优的描述这个世界的概率分布

$$P(\theta|X) = \frac{P(X|\theta) \times P(\theta)}{P(X)}$$

        - θ 是一个随机变量，符合一定的概率分布。在贝叶斯学派里有两大输入和一大输出，输入是**先验概率** $P(\theta)$ 和**似然概率** $P(X|\theta)$，输出是**后验概率** $P(\theta|X)$

4.3 监督学习

- **线性回归**
    - **回归问题**，**连续变量预测**
    - 输出可为任意实数
    - 线性回归对离群点非常敏感，导致模型不稳定，为了缓解这个问题可以考虑逻辑斯蒂回归（logistic regression）
- **逻辑回归**
    - **分类问题**，**二分类**（逻辑斯谛回归、多项逻辑斯谛回归）
    - 输出为概率值，范围在 $(0, 1)$ 之间
    - 是一种引入 $sigmoid$ 函数，非线性回归模型
- **决策树**
    - 通过**树形结构**来进行分类的方法

- 每个非叶子结点表示对分类目标在某个属性上的一个判断，每个分支代表基于该属性做出的一个判断，每个叶子结点代表一种分类结果
- 建立决策树的过程，就是**不断选择属性值对样本集进行划分**、直至每个子样本为同一个类别
- **构建决策树过程**
  - **信息熵**（entropy）可以用来衡量样本集合纯度，越大说明不确定性越大，纯度越低

$$E(D) = -\sum_{k=1}^{K} p_k \log_2(p_k)$$

有K个信息，p_k为第k个信息发生的概率

  - 划分样本集前后信息熵的减少量称为**信息增益**（information gain），用来衡量样本复杂度减少的程度

$$Gain(D, A) = E(D) - \sum_{i=1}^{n} \frac{|D_i|}{|D|} E(D_i)$$

D_i为子样本集包含的样本数量

  - **信息增益率**

$$info = -\sum_{i=1}^{n} \frac{|D_i|}{|D|} \log_2 \left(\frac{|D_i|}{|D|} \right)$$

$$Gain\_ratio = \frac{Gain(D, a)}{info}$$

  - 启发式算法：**先从候选划分属性中找出信息增益高于平均水平的，再从中选取增益率最高的**
  - **剪枝**是决策树对付“ **过拟合** ”的主要手段
    - 预剪枝：提前终止某些分支的生长
    - 后剪枝：生成一棵完全树，再回头剪枝

4.4 无监督学习

- 主要的无监督学习方法：聚类、降维、推荐系统
- **K 均值聚类**
  - 算法步骤
    - 选择 K 个点作为初始质心
    - 将每个点指派到最近的质心，形成 K 个簇
    - 对于上一步得到的结果，进行平均计算，得出新的质心
    - 重复上述步骤，直到质心不再发生变化
- **特征降维**
  - **维数灾难**：在涉及到向量的计算的问题中，随着维数的增加，计算量呈指数倍增长的现象
  - 高维使得学习算法的泛化能力变弱
  - **降维**：将训练数据中的样本从高维空间转换到低维空间
    - 不存在完全无损的降维
  - **主成分分析**（PCA）：通过将一个大的特征集转换成一个较小的特征集，这个特征集**仍然包含了原始数据中的大部分信息**，从而降低了原始数据的维数

        - 算法步骤
            - 数据标准化
            - 计算协方差矩阵
            - 计算协方差矩阵的特征值和特征向量
            - 选择主成分
            - 投影数据到低维空间

4.5 半监督学习

- 三个基本假设
    - **平滑性假设**
    - **聚类假设**
    - **流形假设**

五、深度学习

> Q&A：为什么以前深度学习不“深度”，现在却“深度”了呢？
>
> 1. **数据量的增加**：以前数据量相对较小，限制了复杂模型的训练效果；现在随着互联网的发展，数据量大幅增加，使得深度学习模型能够学习到更多的特征和模式
> 2. **计算能力的提升**：以前计算资源有限，训练深度模型的时间和成本高昂；现在GPU的广泛应用，使得进行大规模并行计算成为可能，从而加速了深度学习模型的训练
> 3. **算法和架构的创新**：以前深度学习模型设计和训练技巧不够成熟，容易出现过拟合；现在新算法（如卷积神经网络CNN、循环神经网络RNN、Transformers等）和训练技巧（如Batch Normalization、Dropout、学习率调度等）的提出，使得训练深度模型变得更加高效和稳定

5.1 前馈神经网络

- **前馈神经网络**（feedforward neural network，**FNN**）是一种最简单的神经网络，由**输入层**、**隐藏层**和**输出层**组成
    - 每层神经元只和相邻层神经元相连，即每层神经元只接受相邻前序神经层中神经元传来的信息，只给相邻后续神经层中神经元传递信息
    - 同一层的神经元之间没有任何连接，后续神经层也不向前序相邻神经层传递信息
    - 是目前最为广泛的神经网络之一
- **神经元**
    - 给定 n 个二值化输入数据 x_i 与连接参数 w_i，MCP 神经元对输入数据线性加权求和，然后将结果根据阈值 θ 进行二值化，得到输出：

$$y = \Phi\left(\sum_{i=1}^{n} w_i x_i\right)$$

    - 二值可能是 0 和 1，也可能是 -1 和 1。
- **激活函数**
    - 神经网络使用**非线性函数**作为激活函数，通过**对多个非线性函数进行组合**，来实现对**输入信息的非线性变换**
    - 激活函数必须是**连续可导**的

- 常用激活函数
  - $sigmoid$ 函数：

$$\text{原函数：}f(x)=\frac{1}{1+e^{-x}}$$
$$\text{导数：}f'(x)=f(x)(1-f(x))$$

    - 缺点：反向传播算法更新参数过程中易出现**梯度消失问题**（导数 <1）
  - $tanh$ 函数：

$$\text{原函数：}f(x)=\frac{1-e^{-2x}}{1+e^{-2x}}$$
$$\text{导数：}f'(x)=1-f^2(x)$$

    - 缺点：反向传播算法更新参数过程中易出现**梯度消失问题**（导数 <1）
  - $ReLU$ 函数：

$$\text{原函数：}f(x)=\max(0,x)$$
$$\text{导数：}0\text{或}1$$

    - **有效克服梯度消失问题**
  - $softmax$ 函数，一般用在**多分类问题**中：

$$f(x_i)=\frac{e^{x_i}}{\sum_{j=1}^{k}e^{x_j}}$$

    - 可将输出概率最大的作为分类目标
- **损失函数**（loss function）/ 代价函数（cost function）
  - **用来计算模型预测值与真实值之间的误差**
  - **均方误差损失函数**（MeanSquareError）：

$$\text{MSE}=\frac{1}{n}\sum_{i=1}^{n}(y_i-\hat{y}_i)^2$$

  - **交叉熵损失函数**，度量两个概率分布之间的差异

$$H(p,q)=-\sum_{x}p(x)\log q(x)$$

    - 旨在描绘通过概率分布 q （预测值）来表达概率分布 p （真实值）的困难程度
    - **交叉熵越小，两个概率分布越接近**
- **感知机模型**
  - **单层感知机**
    - 由一个输入层和一个输出层构成，输出层输出 -1 或 1
    - 可作为一种二类线性分类模型
    - 单层感知机构建损失函数来计算模型预测值与数据真实值之间的误差，通过修改权重最小化损失函数值，来优化模型参数
    - 单层感知机可被用来区分**线性可分**数据
  - **多层感知机**
    - 由输入层、输出层和**至少一层的隐藏层**构成
    - **相邻层之间全连接**

- **参数优化**
  - 神经网络参数优化是一种**监督学习**的过程
  - 模型会利用反向传播算法将损失函数计算所得误差从输出端出发，由后向前传递给神经网络中的每个单元，然后通过梯度下降算法对神经网络中的参数进行更新
  - **梯度下降**（gradient descent）

$$\nabla L = \begin{bmatrix} \frac{\partial L}{\partial w_1} \\ \frac{\partial L}{\partial w_2} \\ \vdots \\ \frac{\partial L}{\partial b_1} \\ \vdots \end{bmatrix}$$

    - 批量梯度下降：在整个训练集上计算损失误差
    - 随机梯度下降：在每个训练样本上计算损失误差，收敛速度慢，但有时可能找到更好的解
    - **小批量梯度下降**：选取训练集上小批量样本计算损失误差，最常用
- **误差反向传播**（error backpropagation，BP）

$$w_1^{new} = w_1 - \eta \times \frac{\partial L}{\partial w_1}$$

  - 在计算得到所有参数相对于损失函数 L 的偏导结果后，利用梯度下降算法，通过 $w_1^{new} = w_1 - \eta \times \frac{\partial L}{\partial w_1}$ 来更新参数取值。然后**不断迭代，直至模型收敛**，此时**损失函数减少到其最小值**

5.2 卷积神经网络

- **卷积**（convolution）是针对像素点的空间依赖性来对图像进行处理的一种技术（前馈神经网络处理图像时，由于模型参数巨大，不仅会占用大量计算机内存，也会模型的训练变得难以收敛）
- **卷积滤波的结果**在卷积神经网络中被称为**特征图**（feature map）
- **下采样**：对图像进行约减
- 在卷积操作时，经常会采用 padding（**填充**）和 striding（**步长**）两种方法

  假设被卷积图像大小为 $w \times w$，卷积核大小为 $F \times F$，上下左右四个边缘填充像素行 / 列数为 $P = \lfloor \frac{F}{2} \rfloor$，步长为 S，则被卷积结果的**分辨率**是 $\frac{w-F+2P}{S} + 1$

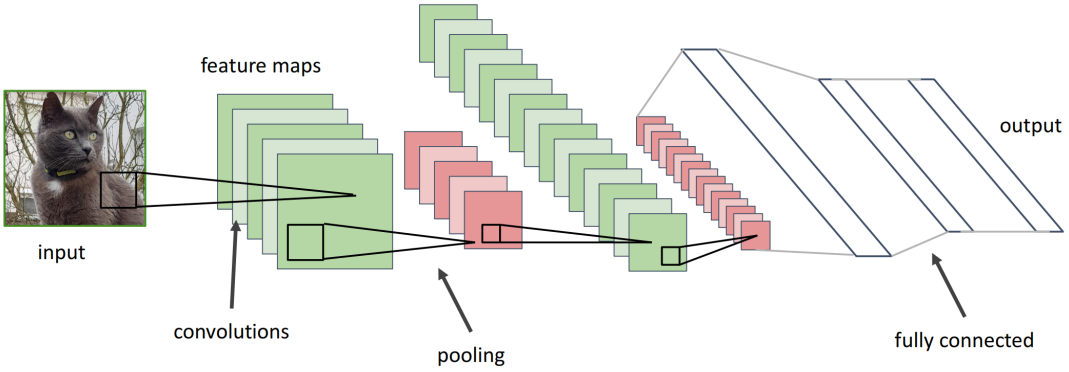
$$\lceil \frac{w + 2P - (F-1)}{S} \rceil = \lceil \frac{w + 2P - F + 1}{S} \rceil = \lfloor \frac{w - F + 2P}{S} + 1 \rfloor$$

- **池化**（pooling）
  - 可用某一区域子块的统计信息来刻画该区域中所有像素点呈现的空间分布模式，来替代区域子块中所有像素点取值
  - **池化操作对卷积结果特征图进行约减**，实现了下采样，同时保留了特征图中主要信息
    - **最大池化**
    - **平均池化**
    - $k - max$ **池化**：取前 k 个最大值，常用于自然语言处理
    - **全局池化**：更不容易过拟合，因为它没有需要优化的参数；对输入的空间平移更为稳健，因为它汇总了空间信息
- 卷积操作和全连接计算在算法上是可相互转换的，如全连接计算可等效于卷积核为 1×1 的卷积运算

- **卷积神经网络工作流程与结构**
  - 对于输入的海量标注数据，通过多次迭代训练，卷积神经网络在经过若干次卷积操作，对卷积所得结果进行激活函数操作和池化操作后，最终通过全连接层学习得到输入数据的特征表达，即分布式向量表达

$$[(Conv-ReLU)*N-Pooling?]*M-(FC-ReLU)*K-Softmax$$

  - 多个**卷积层**（$Conv$）后接激活函数（$ReLU$），然后是可选的**池化层**（$Pooling$），最后是**全连接层**（FC）和激活函数（$ReLU$），最终输出通过 $Softmax$ 进行分类



- **防止过拟合**的操作
  - 数据增强
  - Dropout
  - L1 和 L2 正则化
  - 早停法（提前终止）
- **防止欠拟合**的操作
  - 减少正则化
  - 增加训练时间

5.3 循环神经网络

- **循环神经网络**（recurrent neural network，RNN）是一类处理**序列数据**时采用的网格结构，能够刻画序列数据中存在的**时序依赖关系**
- 循环神经网络本质是模拟人所具有的记忆能力，在学习过程中记住部分已经出现的信息，并利用所记住的信息影响后续节点输出
- RNN 在自然语言处理如**语音识别**、**情感分析**、**机器翻译**等领域有重要应用
- Vanilla RNN （SimpleRNN）循环神经网络模型
  - 循环神经网络在处理数据过程中构成了一个**循环体**
  - 对于一个序列数据，在每一时刻 t，循环神经网络单元会读取当前输入数据 x_t 和前一时刻输入数据 x_{t-1} 所对应的隐式编码结果 h_{t-1}，一起生成 t 时刻的隐式编码结果 h_t

$$h_t=\Phi(W_x\cdot x_t+W_h\cdot h_{t-1}+b)$$

  - 激活函数一般可为 $sigmoid$ 或 $tanh$，使模型能够忘掉无关信息同时更新记忆内容
  - 训练方式为沿时间反向传播算法（BPTT）
    - 根据时刻 t 所得误差更新参数 W_x：

$$\frac{\partial E_t}{\partial W_x}=\sum_{i=1}^{t}\frac{\partial E_t}{\partial O_t}\frac{\partial O_t}{\partial h_t}\left(\prod_{j=i+1}^{t}\frac{\partial h_j}{\partial h_{j-1}}\right)\frac{\partial h_i}{\partial W_x}$$

  - 当**序列过长**时，循环神经网络也容易出现**梯度消失**或**梯度爆炸**的问题

    > - 梯度消失：梯度值随着层数的增加而迅速减小，最终趋近于零，导致靠近输入层的权重更新变得非常缓慢，甚至几乎不更新
    > - 梯度爆炸：在反向传播过程中，梯度值随着层数的增加而迅速增大，最终变得非常大，超出了神经网络的正常处理范围，从而导致模型参数更新不稳定，甚至训练失败

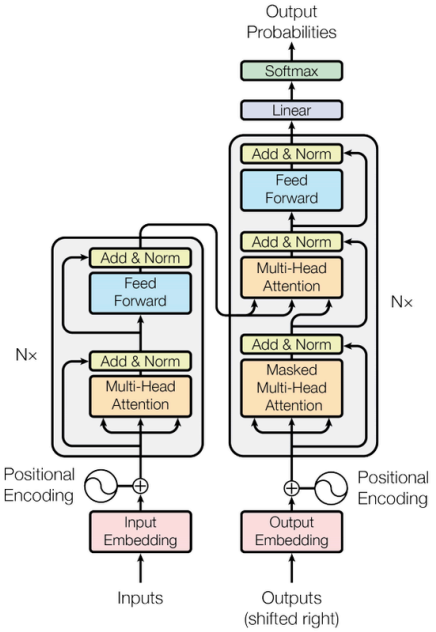
- 输入输出不同情况下循环神经网络结构：
  - **“ 多对多 ”**：输入和输出序列中包含多个单元（机器翻译）
  - **“ 多对一 ”**：输入序列数据中包含多个单元，输出序列数据中只包含一个单元（文本情感分类）
  - **“ 一对多 ”**：输入序列数据中只包含一个单元，输出序列数据中包含多个单元（图像描述生成）
- **长短时记忆网络**（LSTM）：引入了**内部记忆单元**和**门结构**来对当前输入信息以及前序时刻所生成的信息进行整合和传递，**避免了梯度消失问题**
  - 门结构：输入门、遗忘门、输出门
- **门控循环单元**（gated recurrent unit，GRU）：一种对 LSTM 简化的深度学习模型
  - **不再使用记忆单元**来传递信息，而是使用隐藏状态来传递，有**更快的计算速度**
  - 只包含更新门和重置门

5.4 Transformer

- **Transformer 模型**：引入 “ 自注意力 ” 来计算自然语言句子中单词与单词之间的概率关联
- **自注意力模型**工作流程：
  - 首先生成每个单词的**内嵌向量**（embedding vector），记为 w_i
    - **查询向量**：$q_i=W^q\times w_i$
    - **键向量**：$k_i=W^k\times w_i$
    - **值向量**：$v_i=W^v\times w_i$
  - 自注意力权重公式：

$$\mathrm{Attention}(Q,K,V)=\mathrm{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

  - 自注意力模型就是要挖掘单词 w_i 与其他单词在句子中因为上下文关联而具有的自注意力取值大小
- **Transformer 架构图**（以机器翻译的训练任务为例）：



- 模型的**输入**和**输出**
  - 输入：Inputs / Outputs（shifted right）
  - 输出：Output Probabilities
- **内嵌向量**，即词向量（Embedding）
- **位置编码**（Positional Encoding）
  - 表示单词出现在句子中的位置
- **编码器**（Encoder）
  - 由多个自注意力层和前馈神经网络组成
- **解码器**（Decoder）
- **多头注意力机制**（Multi-Head Attention）
  - 由多个 Self-Attention 组成
- **前馈神经网络**（Feed Forword）
  - 对每个位置的表示进行非线性变换，帮助模型进行更复杂的特征学习
- **残差连接**（Add）和**层标准化**（Norm）
  - 残差连接可以缓解梯度消失，改善梯度流动
  - 层标准化可以提高训练的稳定性和速度
- **线性层**（Linear）和 **softmax 层**
  - softmax 得到输出的概率分布，然后通过词典，输出概率最大的对应的单词作为预测输出

> Q&A：为什么我们不在编码器中使用带掩码的自注意力机制？
>
> 1. **完整访问输入序列**
>
> 编码器的自注意力机制允许每个位置的表示访问输入序列中的所有其他位置。这种设计使得模型能够充分利用输入序列的全局信息，捕捉词之间的依赖关系。因此，在编码器中，使用带掩码的自注意力会限制这种信息的获取。
>
> 2. **目标不需要逐步生成**

> 在解码器中，使用带掩码的自注意力机制是为了确保模型在生成序列时只能依赖于已生成的部分，以防止“窥视”未来的词。而编码器的任务是将整个输入序列编码成上下文表示，不涉及逐步生成，因此不需要这种限制。

- Transformer 模型**计算量**公式：

$$\text{总计算量} = l \cdot (12h^2 + 13h)$$
$$h\text{表示隐藏层的维度}, l\text{表示层数}$$

- **GPT-3 架构**：对 50275 个单词进行**词向量建模**，然后用 96 个堆叠的 **Transformer** 极尽其能对句子中每个单词与其他单词的**自注意力关联进行挖掘**，最后通过**前馈神经网络**对所得关联**进行非线性映射**

六、强化学习

6.1 强化学习问题定义

- 强化学习基本概念：
  - 智能体（agent），根据经验做出判断并执行动作
  - 环境（environment），智能体以外的一切
  - 状态（state），智能体对环境的一种理解和编码
  - 动作（action）
  - 策略（policy），当前状态执行某个动作的依据
  - 奖励（reward）
- 强化学习特点：
  - 交互式学习
  - 强化信息的延迟性
  - 无需先验知识
  - 增量式的在线学习
  - 可应用于不确定环境
- 与监督学习和无监督学习相比，强化学习基于评估，数据来源于时序交互，决策过程是序贯决策，目标是选择能够获取最大收益的状态到动作的映射
- **马尔可夫决策过程**
  - 马尔可夫性（Markov property）：下一刻的状态 X_{t+1} 只由当前状态 X_t 决定，与更早的所有状态均无关
  - 满足马尔可夫性的离散随机过程被称为马尔可夫链（Markov chain）
- **定义离散马尔可夫过程**
  - $\{S_t\}_{t=0,1,\ldots}$，可以定义状态转移概率

    $$P(S_{t+1} \mid S_t, A_t)$$

    其中 A_t 为在状态 S_t 下执行的动作
- **定义奖励函数**

$$R_{t+1} = R(S_t, A_t, S_{t+1})$$

描述从第 t 步状态转移采取动作 A_t 到第 $t+1$ 步状态的奖励

在每个时刻定义回报（return）来反映该时刻可得到的累加奖励

$$G_t = R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \cdots$$

其中 γ 称为折扣因子

一般情况下，初始状态和终止状态并不包括在马尔可夫决策过程定义中，可以添加虚拟的初始状态和终止状态，虚拟初始状态以一定概率转移到真正初始状态，真正终止状态以概率 1 转移到虚拟终止状态

智能体逐步采取行动，可得到一个状态序列 $(S_0, S_1, \ldots)$ 称为轨迹（trajectory），轨迹长度可以是无限的，也可以有终止状态 S_T
包含终止状态的问题叫分段（episodic）问题，此时从初始状态到终止状态的完整轨迹称为一个片段（episode）
不包含终止状态的问题叫持续（continuing）问题

强化学习问题定义

智能体选择动作的模型即策略函数，$\pi(s, a)$ 表示在状态 s 下采取动作 a 的概率
一个好的策略函数应该能够使智能体在采取了一系列行动之后可得到最佳奖励

价值函数（value function）

$$V_\pi(s) = E_\pi[G_t \mid S_t = s]$$

表示智能体在时刻 t 时处于状态 s 时，按照策略 π 采取行动时所得到的回报的期望。

动作 - 价值函数

$$q_\pi(s, a) = E_\pi[G_t \mid S_t = s, A_t = a]$$

强化学习可以转化为一个策略学习问题，给定一个马尔可夫决策过程 $MDP = (S, A, P, R, \gamma)$，学习一个最优策略 π^*，对任意 $s \in S$ 使得 $V_{\pi^*}(s)$ 值最大

贝尔曼方程 / 动态规划方程

$$V_\pi(s) = \sum_{a \in A} \pi(s, a) q_\pi(s, a)$$

可用动作 - 价值函数来表达价值函数

状态 s 的价值可用该状态下可采取所有动作而取得的期望价值来表述

$$q_\pi(s, a) = \sum_{s' \in S} P(s' \mid s, a) \left[R(s, a, s') + \gamma V_\pi(s') \right]$$

可用价值函数来表示动作 - 价值函数

在某个状态下执行某一个动作所得的价值可以通过执行该动作之后进入的所有状态获得的瞬时奖励和后续状态可取得价值的期望来表示。

价值函数贝尔曼方程：

$$V_\pi(s) = E_{a \sim \pi(s,\cdot)} E_{s' \sim P(\cdot|s,a)} \left[R(s, a, s'), \gamma V_\pi(s') \right]$$

价值函数取值和时间没有关系，只与策略 π、在策略 π 下从某个状态转移到后续状态所取得的回报以及在后续所得回报有关
V_π 由两个部分构成：执行当前动作所得到的瞬时奖励，在后续状态所能得到的回报的期望的折扣值

贝尔曼方程描述了价值函数或动作 - 价值函数的递推关系，是研究强化学习问题的重要手段

求解最优策略的一种方法是求解最优的价值函数或最优的动作 - 价值函数，即基于价值方法（value-based approach）

6.2 基于价值的强化学习

一种求解最优策略的思路：从一个任意的策略开始，首先计算该策略下价值函数，然后根据价值函数调整改进策略使其更优，不断迭代直到策略收敛。

通过策略计算价值函数的过程称为**策略评估**（policy evaluation）
通过价值优化策略的过程叫做**策略优化**（policy improvement）
策略评估和策略优化交替进行的强化学习求解方法叫做通用策略迭代（GPI），几乎所有强化学习方法都可以使用 GPI 来解释

策略优化定理

假设当前策略为 π，对应的价值函数和动作 - 价值函数为 V_π, q_π，则可以构造策略

$$\pi'(s) = \arg\max_a q_\pi(s, a)$$

此时 π' 不比 π 差

给定策略、价值函数和动作 - 价值函数，就可以通过该式计算得到更好或同等的策略

6.3 基于策略的强化学习

七、博弈论

7.1 基本概念

- **现代博弈论初步形成的标志**：1944 年，**冯•诺伊曼**和**奥斯卡•摩根斯特恩**（Oskar Morgenstern）合著的《**博弈论与经济行为**》一书出版
- 参与博弈的决策主体被称为**玩家**或**参与者**，通常被认为是完全理性的
- 在博弈中参与者能获得的与博弈相关的知识称为**信息**
- 博弈需要遵循一定的**规则**
- 参与者可以采取的行动方案称为**策略**
    - 策略必须是一整套在采取行动之前就已经准备好的完整方案
    - 一个参与者能采纳策略的全体组合形成其所拥有的**策略集**
    - 参与者也可以按照一定概率随机选择若干不同行动，称为**混合策略**，反之称为**纯策略**
- 参与者采取各自行动后形成的状态称为**局势**，不同局势下各个参与者所得到的利益或回报称为博弈的**收益**
    - 混合策略下收益为**期望收益**
- **博弈的分类**
    - 是否允许参与者之间合作：**合作博弈**、**非合作博弈**
    - 决策时间：**静态博弈**、**动态博弈**
        - 静态博弈中所有参与者同时决策，动态博弈中由规则决定先后且后者知道前者的行动
    - 对信息的了解程度：**完全信息博弈**、**不完全信息博弈**
        - 完全信息博弈中所有参与者都知道所有信息，不完全信息博弈中参与者只知道部分信息
        - 完美信息博弈：参与者知道所有参与者之前采取的行动
    - 总收益：**零和博弈**、**非零和博弈**
- **纳什均衡**：**博弈的稳定局势**

- **纳什定理**：**若参与者有限，每位参与者采取策略的集合有限，收益函数为实值函数，则博弈对抗必存在混合策略意义下的纳什均衡**
- **策梅洛定理**：**对于任意一个有限步的双人完全信息零和动态博弈，一定存在先手必胜策略、后手必胜策略或双方保平策略**

7.2 博弈策略求解

- **遗憾最小化算法**
  - 假设一共有 N 个玩家，玩家 i 在博弈中采取的策略为 σ_i
  - 一个策略组包含所有玩家策略，用 $\sigma = (\sigma_1, \sigma_2, \ldots, \sigma_{|N|})$ 表示
  - σ_{-i} 表示 σ 中除了 σ_i 之外的策略（即除去玩家 i 所采用的策略），$\sigma_{-i} = \{\sigma_1, \sigma_2, \ldots, \sigma_{i-1}, \sigma_{i+1}, \ldots, \sigma_N\}$
  - 玩家 i 对于所有其他玩家的策略组 σ_{-i} 的最佳反应策略 σ_i^* 满足如下条件：

$$u_i(\sigma_i^*, \sigma_{-i}) \geq \max_{\sigma_i' \in \Sigma_i} u_i(\sigma_i', \sigma_{-i})$$

  - 纳什均衡：策略组 $\sigma = (\sigma_1^*, \sigma_2^*, \ldots, \sigma_{|N|}^*)$ 是纳什均衡当且仅当对每个玩家 $i \in N$，满足如下条件：

$$u_i(\sigma^*) \geq \max_{\sigma_i' \in \Sigma_i} u_i(\sigma_1^*, \sigma_2^*, \ldots, \sigma_i', \ldots, \sigma_{|N|}^*)$$

  - 在博弈中，玩家 i 在第 T 轮次（每一轮表示一次博弈完成）采取策略 σ_i 的**累加遗憾值**定义如下：

$$Regret_i^T(\sigma_i) = \sum_{t=1}^{T}(u_i(\sigma_i, \sigma_{-t}) - u_i(\sigma^t))$$

  - **有效遗憾值**：

$$Regret_i^{T,+}(\sigma_i) = max(Regret_i^T(\sigma_i), 0)$$

  - > **整理到这里了。。。**
  -
  - 在博弈对决中，不同玩家在不同时刻会采取相应策略以及行动。策略 σ 下对应的行动序列 h 发生的概率表示为 $\pi_\sigma(h)$。于是，

$$\pi_\sigma(h) = \prod_{i \in N} \pi_i^\sigma(h)$$

    这里 $\pi_i^\sigma(h)$ 表示玩家 i 使用策略 σ_i 促使行动序列 h 发生的概率。除玩家 i 以外，其他玩家通过各自策略促使行动序列 h 发生的概率可表示为：

$$\pi_{-\sigma_i}(h) = \prod_{j \in N \setminus \{i\}} \pi_j^\sigma(h)$$

  - 对于每个玩家 $i \in N$，$u_i : Z \to \mathbb{R}$ 表示玩家 i 的收益函数，即在到达终止序列集合 Z 中某个终止序列时，玩家 i 所得到的收益。
  - 玩家 i 在给定策略 σ 下所能得到的期望收益可如下计算：

$$u_i(\sigma) = \sum_{h \in Z} u_i(h)\pi_\sigma(h)$$

7.3 博弈规则设计

八、安全与可信

8.1 人工智能伦理

8.2 人工智能安全性

- 攻击方法
  - **对抗攻击**（推理阶段）
    - 白盒攻击：预先知道模型参数，利用模型梯度信息制作对抗样本
    - 黑盒攻击：不知道模型参数，利用输入输出对制作对抗样本
  - **模型隐私窃取**（推理阶段）
  - **数据投毒攻击**（训练阶段）
  - **后门攻击**（训练阶段）
- 防御方法
  - 测试阶段的防御方法：在测试阶段引入随机性，可以缓解或消除对抗噪声的负面影响
  - 训练阶段的防御方法：将对抗样本以数据增强的方式纳入模型的训练过程中

8.3 人工智能可信性