****

人工智能与机器学习

实 验 报 告

|  |  |
| --- | --- |
| **学 院** | 网络空间安全学院 |
| **专 业** | 网络空间安全 |
| **班 级** | 23270312 |
| **学 号** | 23280343 |
| **学生姓名** | 王书利 |
| **教师姓名** | 黄娜娜 |
| **完成日期** | 2025/05/25 |
| **成 绩** |  |
| **实验五 模糊推理** | | |
| 1. **实验目的**   1. 掌握模糊推理的基本原理和实现过程；  2. 理解模糊集合、隶属度函数和模糊规则的定义方法；  3. 掌握模糊推理系统的构建和推理过程；  4. 学习如何应用模糊推理解决实际问题。 | | |
| 1. **实验内容**   模糊推理是一种模拟人类模糊思维的推理方法，广泛应用于控制系统、决策分析等领域。实验将实现一个简单的模糊推理系统，包括以下步骤：  1.模糊集合定义：确定输入输出的模糊集合及其隶属度函数  2.模糊规则建立：构建"如果-则"形式的模糊规则库  3.模糊推理：实现模糊关系的合成运算(Max-Min或Max-Prod)  4.去模糊化：采用重心法或其他方法将模糊输出转换为精确值。  **实验数据：**  给定输入模糊集合A'和模糊关系R：  A' = 1/1 + 1/2 + 0.5/3 + 0.2/4 + 0/5    = (1, 1, 0.5, 0.2, 0)  R = [0.5, 0.5, 0.5, 0.8, 1.0;     0.5, 0.5, 0.5, 0.8, 1.0;     0.5, 0.5, 0.5, 0.5, 0.5;     0.2, 0.2, 0.2, 0.2, 0.2;     0.0, 0.0, 0.0, 0.0, 0.0] | | |
| 1. **实验环境**   硬件配置：  CPU：AMD Ryzen 9 7945HX with Radeon Graphics @ 2.50 GHz  RAM：16GB  存储：1T SSD  显卡：NVIDIA GeForce RTX 4060 Laptop GPU  软件配置：  操作系统：Windows 11  编程语言：python 3.10.16  IDE：vscode  第三方库：numpy; matplotlib | | |
| 1. **主要操作步骤及实验结果记录**    1. 代码实现   1.定义模糊关系矩阵  R = np.array([      [0.5, 0.5, 0.5, 0.8, 1.0],      [0.5, 0.5, 0.5, 0.8, 1.0],      [0.5, 0.5, 0.5, 0.5, 0.5],      [0.2, 0.2, 0.2, 0.2, 0.2],      [0.0, 0.0, 0.0, 0.0, 0.0]  ])  2.实现Max-Min模糊推理  def fuzzy\_inference\_max\_min(A, R):      result = np.zeros(R.shape[1])      for j in range(R.shape[1]):          min\_vals = np.minimum(A, R[:, j])          result[j] = np.max(min\_vals)      return result  3.实现Max-Prod模糊推理  def fuzzy\_inference\_max\_prod(A, R):      result = np.zeros(R.shape[1])      for j in range(R.shape[1]):          prod\_vals = A \* R[:, j]          result[j] = np.max(prod\_vals)      return result  4.去模糊化  def defuzzification\_centroid(B):      x = np.arange(1, len(B) + 1)      return np.sum(B \* x) / np.sum(B) if np.sum(B) != 0 else 0   * 1. 推理计算及可视化输出        * 1. 结果分析   从结果来看，两个方法在这组数据上的合成输出完全一致，说明在当前输入 A 和关系矩阵 R 的配置下，两种推理方法对最终结果的影响差异不大。推理输出 B 的最后一个元素具有最高的隶属度（为 1.0），这与关系矩阵 R 的结构一致，因为 R 的最后一列拥有最大值，输入 A 中的前两个元素隶属度也较高，因此在合成过程中对最终结果的贡献较大。使用重心法进行去模糊化后，Max-Min 与 Max-Prod 方法得到相同的输出值 3.39，说明在这一情境下两种方法都能较好地反映输入信息的模糊关系，且输出具有较强的稳定性。若今后更换关系矩阵，特别是在 R 各列值差异较大的情况下，Max-Prod 与 Max-Min 的结果可能就会有更明显的差异，从而影响最终的去模糊化输出。因此，不同合成方法在不同应用场景下需要具体分析选择。 | | |
| 1. **实验分析总结及心得**   通过本次模糊推理实验，我对模糊集合的基本概念、模糊规则的表达方式以及推理过程中的运算逻辑有了更加直观和深刻的理解。实验中分别使用了 Max-Min 和 Max-Prod 两种常见的模糊合成方法，对比之后发现虽然在本次设定下两种方法的结果一致，但它们的运算机制和适用场景还是存在一定差异。Max-Min 强调的是对最小匹配值的保护，更适合处理约束性强的场景；而 Max-Prod 则更注重整体匹配强度的积累，更适合对输入强度有要求的系统。  在实验过程中，我还学习了如何用 Python 编程实现模糊推理，包括隶属度计算、矩阵合成、去模糊化等步骤，并通过可视化手段帮助理解中间结果的变化趋势。尤其是在绘制图像的时候，能够清晰地看到输入与输出之间的关系，让抽象的模糊理论变得更具象、更容易理解。虽然实验中使用的是较简单的 5 维模糊集合，但整个推理过程的逻辑是可以拓展到更复杂的实际系统中的。  此外，我还尝试修改模糊关系矩阵的结构，并观察不同推理方法下输出的差异，这个过程让我体会到模糊系统的灵活性以及建模时参数设置的重要性。通过这次实验，不仅加深了对模糊推理原理的掌握，也提高了动手编程和调试能力，为后续学习模糊控制系统等更高阶的课程打下了良好的基础。总的来说，这是一项结合理论与实践、具有挑战性但也很有收获的实验经历。 | | |