强化学习算法比较实验：LunarLander环境

项目目标

在本项目中，你将使用OpenAI Gym的LunarLander-v2环境，实现并比较五种强化学习算法。通过这个实验，你需要：

1. 理解不同强化学习算法在连续状态空间问题中的表现差异
2. 掌握将经典表格型方法应用于复杂环境的技术
3. 实现深度Q网络(DQN)及其关键组件
4. 分析算法在样本效率、策略质量和稳定性方面的差异

实验环境

```
import gym
env = gym.make('LunarLander-v2')
```

环境特性：

- 8维连续状态空间（坐标、速度、角度等）
- 4个离散动作（无操作、左引擎、主引擎、右引擎）
- 密集奖励系统（成功着陆+100~140分，坠毁-100分，燃料消耗-0.3分/帧）
- 每局最多1000时间步

实验要求

第一部分：算法实现（50分）

1. 表格型方法实现（需状态离散化）

- Monte Carlo控制（10分）
- SARSA（10分）
- SARSA(λ) with eligibility traces（10分）
- Q-learning（10分）

2. 深度强化学习实现

- DQN with experience replay（10分）

第二部分：对比实验（30分）

1. **训练曲线比较**（10分）
   - 绘制所有算法在训练过程中的平均奖励曲线
   - 横轴：训练episode数，纵轴：最近100局平均奖励
2. **最终策略评估**（10分）
   - 对每个算法运行100次测试episode
   - 记录以下指标：
     - 平均奖励
     - 着陆成功率（reward > 200）
     - 平均燃料消耗
3. **关键对比分析**（10分）
   - 对比MC和SARSA(λ)方法的样本效率
   - 分析SARSA与Q-learning的策略差异
   - 说明资格迹如何影响SARSA(λ)的学习速度
   - 讨论DQN相比表格型方法的优劣势

第三部分：创新探索（20分）

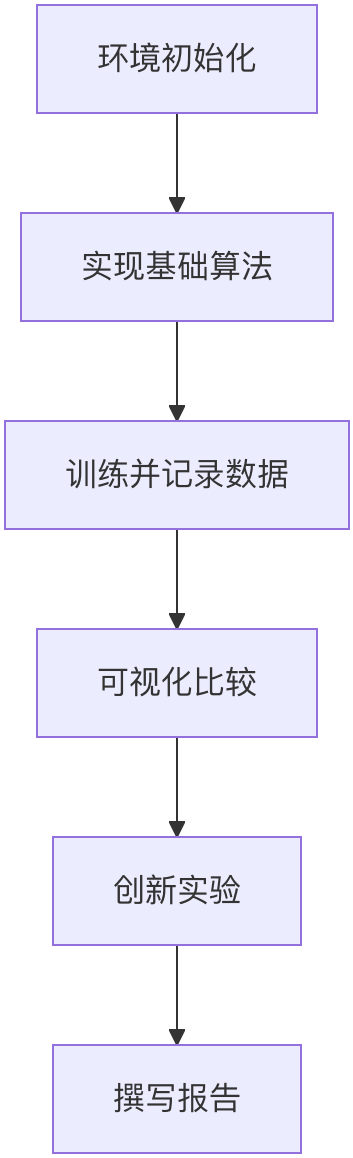
1. **状态离散化优化**（10分）
   - 设计至少两种不同的状态离散化方案
   - 比较其对表格型方法性能的影响
2. **DQN扩展**（10分）
   - 实现以下任一改进：
     - Double DQN
     - Dueling DQN
     - Prioritized Experience Replay
   - 分析改进效果

实验指南

1. 状态离散化建议

```
def discretize_state(state, bins=[8, 8, 6, 6, 6, 4]):
    """将连续状态离散化为6个关键维度"""
    features = [
        np.digitize(state[0], np.linspace(-1, 1, bins[0])),  # x坐标
        np.digitize(state[1], np.linspace(-1, 1, bins[1])),  # y坐标
        np.digitize(abs(state[2]), np.linspace(0, 2, bins[2])),  # x速度绝对值
        np.digitize(abs(state[3]), np.linspace(0, 2, bins[3])),  # y速度绝对值
        np.digitize(state[4], np.linspace(-1, 1, bins[4])),  # 角度
        np.digitize(state[5], np.linspace(-2, 2, bins[5]))   # 角速度
    ]
    return tuple(features)
```

2. 实验流程建议



3. 评估标准示例

算法	平均奖励	成功率	训练时间	超参数敏感性
MC控制	150±30	40%	25min	高
SARSA(λ=0.9)	180±25	65%	20min	中
DQN	220±50	80%	90min	低

交付要求

1. **代码提交**：

- 完整的Python实现（.py或.ipynb文件）
- 包含清晰的注释和函数说明

2. **实验报告**：
   - 算法实现细节（特别是状态离散化方法）
   - 实验结果图表（至少包含训练曲线和最终性能对比）
   - 关键发现与分析（回答第二部分的问题）
   - 创新实验的结果与讨论
3. **演示视频**：
   - 录制1分钟内的算法表现对比
   - 展示最佳和最差策略示例