# 完整设计文档

## 1. 项目概述

本项目旨在设计并实现一个基于深度强化学习（Deep Q-Learning, DQL）的自动化股票交易系统。通过结合量化交易的核心思想和技术因子提取，构建高效的智能交易策略。系统以股票历史数据为基础，训练强化学习代理，使其能够动态调整买入、卖出和持有策略，以实现收益最大化。

## 2. 系统模块划分

系统整体设计采用模块化开发，便于维护和扩展。各模块功能明确，涵盖从数据获取到模型训练及策略回测的全过程。

### 2.1 数据获取模块

* 功能描述：从 Yahoo Finance 提取股票的历史行情数据，包括开盘价、高价、低价、收盘价、成交量等。
* 输入参数：ticker（股票代码），start\_date（数据起始日期）。
* 输出结果：带有时间索引的 Pandas DataFrame，包含股票行情的基础数据。
* 关键函数：download\_daily\_data(ticker, start\_date)。

### 2.2 因子计算模块

* 功能描述：提取多种技术因子，覆盖动量类、波动率类、趋势类和成交量类等多维因子。
* 关键因子：RSI、MACD、KDJ、动量因子、布林带、金叉/死叉、ATR等。
* 输入数据：包括股票历史价格和成交量等。
* 输出结果：增加技术因子的 DataFrame，并保存为 alpha\_factors.xlsx。
* 关键函数：calculate\_alpha\_factors(data)。

### 2.3 数据预处理模块

* 功能描述：标准化因子数据以消除不同因子尺度的影响。提取用于模型训练的核心特征列。
* 输入数据：包含技术因子的 DataFrame。
* 输出结果：标准化后的因子数据和特征列名称列表。
* 关键函数：prepare\_data(data)。

### 2.4 强化学习代理模块

* 功能描述：实现 DQL 强化学习框架，包括动作决策、经验回放、模型训练等。
* 代理设计：状态空间（因子输入），动作空间（买入、卖出、持有），奖励函数（收益变化）。
* 神经网络结构：输入层（因子数）、两层隐藏层（128 和 64 个神经元）、输出层（动作空间）。
* 关键函数：DQLAgent 类及其方法（\_build\_model、act、replay）。

### 2.5 训练模块

* 功能描述：将数据划分为训练集和测试集，在训练集上优化 DQL 模型。
* 输入数据：训练集数据、因子特征和模型配置。
* 输出结果：训练好的 DQL 模型。
* 关键函数：train\_agent(data, features, initial\_balance, transaction\_cost, train\_split)。

### 2.6 回测模块

* 功能描述：在测试集上验证交易策略表现，计算组合价值变化，记录买入和卖出信号。
* 核心逻辑：包含买入、卖出、持有逻辑及交易成本计算。
* 输出结果：投资组合价值变化曲线、交易信号。
* 关键函数：backtest(agent, test\_data, features, initial\_balance, transaction\_cost)。

## 3. 系统工作流程

1. 数据获取：下载指定股票的历史数据。  
2. 因子计算：计算技术因子，生成用于训练和预测的特征数据。  
3. 数据预处理：标准化因子数据，提取核心特征。  
4. 模型训练：在训练集上优化 DQL 模型，动态调整交易策略。  
5. 策略回测：使用测试集验证策略表现，输出收益曲线和交易信号图。

## 4. 回测逻辑

初始条件：账户初始资金 100,000 美元，每笔交易成本为 0.1%。  
交易策略：包括买入、卖出、持有三种动作，奖励函数基于组合总价值变化。  
输出结果：投资组合价值曲线、买卖信号图。

## 5. 可扩展性与优化方向

1. 多资产支持：扩展到多个股票，构建多资产组合。  
2. 因子增强：增加行业轮动、宏观经济指标等非价格类因子。  
3. 多模型集成：将 DQL 与基于规则的策略结合。  
4. 优化奖励函数：引入风险调整后的收益（如 Sharpe 比率）。  
5. 超参数调优：自动化优化模型的学习率、隐藏层节点数等参数。

## 6. 专业术语定义

RSI：相对强弱指标，衡量价格上涨与下跌的动量强度。  
MACD：指数平滑移动平均线差分，用于捕捉趋势。  
KDJ：随机指标，用于超买超卖判断。  
布林带：上下轨范围内的价格波动范围。  
DQL：基于神经网络的强化学习算法，用于动态决策问题。