

# R-Trader : 강화 학습에 기반한 자동 주식 거래 시스템

(R-Trader: An Automatic Stock Trading System based on Reinforcement Learning)

이재원<sup>†</sup> 김성동<sup>\*\*</sup> 이종우<sup>\*\*\*</sup> 채진석<sup>\*\*\*\*</sup>  
(Jae Won Lee) (Sung Dong Kim) (Jongwoo Lee) (Jin Suk Chae)

**요약** 자동 주식 거래 시스템은 시장 추세의 예측, 투자 종목의 선정, 거래 전략 등 매우 다양한 최적화 문제를 통합적으로 해결할 수 있어야 한다. 그러나 기존의 감독 학습 기법에 기반한 거래 시스템들은 이러한 최적화 요소들의 효과적인 결합에는 큰 비중을 두지 않았으며, 이로 인해 시스템의 궁극적인 성능에 한계를 보인다. 이 논문은 주가의 변동 과정이 마르코프 의사결정 프로세스(MDP: Markov Decision Process)라는 가정 하에, 강화 학습에 기반한 자동 주식 거래 시스템인 R-Trader를 제안한다. 강화 학습은 예측과 거래 전략의 통합적 학습에 적합한 학습 방법이다. R-Trader는 널리 알려진 두 가지 강화 학습 알고리즘인 TD(Temporal-difference)와 Q 알고리즘을 사용하여 종목 선정과 기타 거래 인자의 최적화를 수행한다. 또한 기술 분석에 기반하여 시스템의 입력 속성을 설계하며, 가치도 함수의 근사를 위해 인공 신경망을 사용한다. 한국 주식 시장의 데이터를 사용한 실험을 통해 제안된 시스템이 시장 평균을 초과하는 수익을 달성할 수 있고, 수익률과 위험 관리의 두 가지 측면 모두에서 감독 학습에 기반한 거래 시스템에 비해 우수한 성능 보임을 확인한다.

**키워드** : 거래 시스템, 강화 학습, 예측, TD 알고리즘, Q 알고리즘, 신경망, 기술 분석

**Abstract** Automatic stock trading systems should be able to solve various kinds of optimization problems such as market trend prediction, stock selection, and trading strategies, in a unified framework. But most of the previous trading systems based on supervised learning have a limit in the ultimate performance, because they are not mainly concerned in the integration of those subproblems. This paper proposes a stock trading system, called R-Trader, based on reinforcement learning, regarding the process of stock price changes as Markov decision process (MDP). Reinforcement learning is suitable for joint optimization of predictions and trading strategies. R-Trader adopts two popular reinforcement learning algorithms, temporal-difference (TD) and Q, for selecting stocks and optimizing other trading parameters respectively. Technical analysis is also adopted to devise the input features of the system and value functions are approximated by feedforward neural networks. Experimental results on the Korea stock market show that the proposed system outperforms the market average and also a simple trading system trained by supervised learning both in profit and risk management.

**Key words** : trading system, reinforcement learning, prediction, TD algorithm, Q algorithm, neural network, technical analysis

· 본 연구는 2002년도 인천대학교 학술연구조성비 지원에 의하여 수행되었음

<sup>†</sup> 비회원 : 성신여자대학교 컴퓨터정보학부 교수

jwlee@cs.sungshin.ac.kr

<sup>\*\*</sup> 비회원 : 한성대학교 컴퓨터정보학부 교수

sdkim@hansung.ac.kr

<sup>\*\*\*</sup> 종신회원 : 광운대학교 컴퓨터공학부 교수

jwlee44@daisy.kwangwoon.ac.kr

<sup>\*\*\*\*</sup> 종신회원 : 인천대학교 컴퓨터공학과 교수

jschae@incheon.ac.kr

논문접수 : 2002년 5월 29일

심사완료 : 2002년 8월 9일

## 1. 서론

많은 잡음(noise), 비정상성(non-stationarity), 그리고 비선형성으로 인해 금융 시계열의 가격 변동이나 기대 수익을 예측하는 것은 매우 어려운 작업이다. 현재 많은 투자 전문가들은 시장 예측을 위해 기본 분석(fundamental analysis)이나 기술 분석(technical analysis) 기법들을 사용하고 있다. 그러나 특정 시점에 알려진 모든 정보가

그 시점의 가격에 이미 반영되어 있으므로, 미래 가격의 변동을 예측하는 것은 불가능하다고 주장하는 '효율적 시장 가설(EMH: Efficient Market Hypothesis)'[1]로 인해, 학계에서는 기존의 시장 분석 방법론들에 대해 회의적인 입장이 지배적이었다. 여전히 효율적 시장 가설을 완전히 기각할 수 있는 연구 결과가 보고되고 있지는 않지만, 최근 인공 지능 분야, 특히 기계 학습 분야의 연구자들은, 시장이 처음 생각했던 것만큼 효율적이지는 않아서 인공 지능 기법과 기술 분석 기법의 결합에 의해 시장 평균치를 초과하는 수익의 달성이 가능하다는 연구 결과들을 제시하고 있다.

시장 가격의 변동에 대한 예측력을 실제 투자 환경에서 활용하기 위해서는 예측의 수행과 함께 동적인 거래 전략(trading strategy)을 사용하는 거래 시스템이 구축되어야 한다. 이를 위한 접근법은 크게 두 가지로 분류할 수 있다. 하나는 전체 거래 문제를 두 개의 독립된 단계로 분리하는 것이다. 먼저 첫 단계에서는 예측 모형이 미래의 가격 변동에 대한 예측을 수행하고, 전략 단계에서 이 예측 정보를 실제 거래 신호(trading signal)로 변환하는 것이다. Chou 등은 예측 단계에 전통적인 기술 분석과 인공 지능 기법을 사용하였고, 거래 전략으로는 중단-전환(SAR: Stop and Reverse) 방법을 사용하였다[2]. Zhang 등은 다중 신경망으로 선물 거래(futures trading)를 예측하였고 수익률, 손절매율(stop loss ratio), 그리고 거래세(transaction cost) 등을 고려하는 독립적인 거래 전략을 사용하였다[3]. Kuo도 정량적(기술적) 요소와 정성적(기본적) 요소들을 결합한 예측에 신경망을 사용하였고, 거래 전략으로는 몇 가지 임계치와 거래 규칙을 정의하였다[4].

다른 하나의 접근법은 강화 학습에 의해 전체 거래 시스템을 직접 최적화하는 것이다. Neuneier 등은 몇 가지 가정과 시장 특성에 대한 단순화를 통해 금융 시장을 MDP로 모형화하였다[5]. 그는 또한 [6]에서 위험 회피를 선호하는 방향으로 Q-학습을 변형하여 적용하였다. Moody 등은 재귀 신경망(recurrent neural network)과 유사한 강화 학습 체계를 제안하였다[7][8]. 이 체계에서 정책은 널리 알려진 재귀 신경망의 학습 기법인 전후 시간을 고려한 역전파(backpropagation)에 의해 갱신되었다. 이 연구들은 학습된 자산 분배기(asset allocator)를 사용하여 S&P 500과 T-Bill 시장에서의 위치(position) 변경을 통해 수익을 얻을 수 있

음을 보였다. Xiu 등도 Q-학습을 이용한 포트폴리오 관리 시스템을 제안하였다[9]. 이 연구에서는 절대적 수익과 상대적 위험-고려(risk-adjusted) 수익이라는 두 가지 성능 함수를 사용하였으므로, 두 개의 신경망으로 학습을 수행하였다.

위에 기술한 기존의 강화 학습 기반 거래 시스템들은 모두 시장 종합 지수와 같은 단일 시계열을 대상으로 하였거나 몇 가지 투자 대상에 대한 자산 분배 문제를 다루었다. 반면에 이 논문에서 제안하고 있는 R-Trader는 수백 종목의 주식을 포함하고 있는 주식 시장에서의 포트폴리오 최적화를 목표로 한다. 또한 기존의 연구들은 입력 속성(input attribute)에 대해 비중 있게 언급하지 않았으나, 이 논문에서는 그랜빌(Granville)의 법칙과 일본식 봉차트(Japanese candlestick chart)에 기반한 입력 속성의 설계에도 비중을 둔다. R-Trader는 예측 문제에 적합한 TD 알고리즘을 사용하는 '매수 예측 모듈'과 제어 문제에 적합한 Q 알고리즘을 사용하는 '매도 모듈'로 구성된다. 매수 예측 모듈은 매수를 위한 예측만을 독립적으로 수행하며, 매도 모듈은 매도를 위한 예측과 거래 전략을 통합적으로 최적화한다. 2절에서는 강화 학습의 개요와 알고리즘을 소개하고, 3절에서는 R-Trader의 전체 구조를 설명하며, 4절과 5절에서는 종목 선정과 매도를 위한 강화 학습 모형을 기술한다. 마지막으로 6절에서는 실험 결과를 제시한다.

## 2. 강화 학습의 개요와 알고리즘

강화 학습은 목표 지향 학습(goal-directed learning)과 의사 결정 과정을 이해하고 자동화하는 기법이다[10]. 이 기법은 그림 1에서 보는 바와 같이 환경과의 직접적인 상호작용을 통한 행위자의 학습에 중점을 두고 있다. 점에서 다른 기법들과 구별된다.

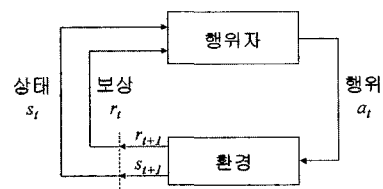


그림 1 행위자와 환경의 상호작용

각 시간 단계(time step)  $t$ 에서 행위자(agent)는 현재 상태(state)  $s_t$ 를 감지하고, 현재 행위(action)  $a_t$ 를 선택하여 수행한다. 환경은 응답으로 행위자에게 보상(reward)  $r_{t+1}=r(s_t, a_t)$ 를 주고 다음 상태  $s_{t+1}=\delta(s_t, a_t)$ 를 생성한

1) 일반적으로 시장 지수(market index)의 등락률을 의미하며, 여기서 시장 지수란 특정 시장에 상장된 모든 주식의 가격을 종합적으로 표현하는 수치이다.

다. 여기서 함수  $r$ 과  $\delta$ 는 환경의 일부로 행위자가 이것에 대해 반드시 알 필요는 없다. 마르코프 의사결정 프로세스에서 함수  $r(s_t, a_t)$ 와  $\delta(s_t, a_t)$ 는 과거의 상태, 행위와 무관하게 현재의 상태와 행위에만 의존한다. 상태의 집합을  $S$ , 행위의 집합을  $A$ 라고 할 때, 행위자의 목적은 현재 관찰된 상태  $s_t$ 에 기반하여 다음 행위  $a_t$ 를 선택하기 위한 정책  $\pi: S \rightarrow A$ (즉,  $\pi(s_t) = a_t$ )를 학습하는 것이다. 임의의 정책  $\pi$ 에 대하여, 상태-가치도(state-value)  $V^\pi(s)$ 는 특정 상태에서 출발하여 기대되는 미래 누적 보상이고, 행위-가치도(action-value)  $Q^\pi(s, a)$ 는 특정 상태-행위 쌍에서 출발하여 기대되는 누적 보상이다. 다음은 이에 대한 전형적인 정의이다.

$$\begin{aligned} V^\pi(s_t) &= Q^\pi(s_t, a_t) \\ &= r_{t+1} + \gamma r_{t+2} + \gamma^2 r_{t+3} + \dots \quad (1) \\ &= \sum_{i=0}^{\infty} \gamma^i r_{t+i} \end{aligned}$$

여기서  $v(0 < v < 1)$ 는 지연된 보상과 즉시(immediate) 보상의 비중을 결정하는 상수이다. '최적 정책(optimal policy)'  $\pi^*$ 는 모든 상태에 대해 상태-가치도를 최대화하거나 모든 상태-행위 쌍에 대해 행위-가치도를 최대화하는 정책이다:

$$\begin{aligned} \pi^* &= \underset{\pi}{\operatorname{argmax}} V^\pi(s), \text{ for all } s \in S \quad (2) \\ &= \underset{\pi}{\operatorname{argmax}} Q^\pi(s, a), \text{ for all } s \in S, a \in A \end{aligned}$$

$\pi^*$ 를 얻기 위해서는 그림 2와 같은 일반화된 정책 순환(GPI: Generalized Policy Iteration) 과정이 필요하다. 이 과정은 현재의 정책  $\pi$ 를 사용하여 각 상태의 가치도(value)를 계산함으로써 정책을 평가하고, 갱신된 가치도를 다시 정책의 개선에 사용하는 순환 과정이다.

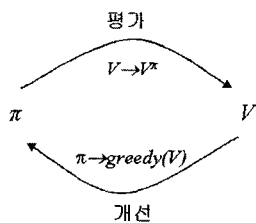


그림 2 일반화된 정책 순환 과정

다양한 강화 학습 알고리즘들은 GPI의 관점에서 분류할 수 있으며, 주된 차이점은 예측(평가) 문제에 대한 접근 방식이다. Monte Carlo 알고리즘과 TD 알고리즘은 환경에 대한 모델<sup>2)</sup>(model)이 주어지지 않은 경우에

도 경험(혹은 샘플링)에 의해 학습할 수 있다. TD 알고리즘은 또한 동적 프로그래밍(dynamic programming, DP)과 같이 기존의 예측에 부분적으로 기반하여 데이터를 갱신하는 부트스트래핑(bootstrapping) 알고리즘이다 [11]. 즉, 기존의 평가값(estimate)에 일정 정도 의존하는 알고리즘이다. 결국, TD 알고리즘은 Monte Carlo 알고리즘의 샘플링과 DP 알고리즘의 부트스트래핑을 결합한다. 특히 Monte Carlo 알고리즘의 경우 최종적인 결과(return)가 확정될 때까지 기다려야만 갱신이 가능한 것과 달리, TD 알고리즘은 하나의 시간 단계만 지나면 즉시 갱신이 가능하다. 따라서 TD는 주식 거래와 같은 연속(continuous) 작업이나 매우 긴 에피소드<sup>3)</sup>(episode)로 이루어진 작업에 적합하다[12]. 시간 단계  $t+1$ 에서 관찰된 보상  $r_{t+1}$ 과 평가값  $V(s_{t+1})$ 를 사용하여 즉시 목표값을 생성한다. 가장 단순한 형태의 TD 알고리즘인 TD(0)의 가치도 갱신은 다음과 같다:

$$V(s_t) \leftarrow V(s_t) + \alpha [r_{t+1} + \gamma V(s_{t+1}) - V(s_t)] \quad (3)$$

사실상 TD 갱신의 목표값은  $r_{t+1} + \gamma V(s_{t+1})$ 이다.

제어 문제(최적 정책의 획득)의 경우에는 TD(0)의 변형인 Q 알고리즘이 널리 사용된다. 다음의 Q 알고리즘 갱신 공식은 평가와 개선을 동시에 수행하며, 이를 통해 최적 정책을 찾아 나간다:

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha [r_{t+1} + \gamma \max_{a_{t+1}} Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t)] \quad (4)$$

### 3. R-Trader의 전체 구조

R-Trader의 전체 구조는 그림 3과 같다. 일별 주가 데이터베이스는 한국 증권거래소에 상장되어 있는 모든 주식의 일별 주가 데이터를 저장하고 있다. 여기서 일별 주가 데이터는 주가의 장중 변동을 요약한 시가, 고가, 저가, 종가 및 거래량으로 구성된다. 일별 주가 데이터베이스에는 거래량이 너무 작거나 종합주가지수와 현저히 다른 주가 변동 성향을 보이는 주식들이 다수 포함되어 있다<sup>4)</sup>. 이 논문에서 수행하는 학습과 모의 실험에서는 이러한 주식들을 배제하기 위해, 일정 기간 동안의 평균 거래량 검사 및 주가 변동 성향 분석을 통해 안정성이 있다고 판단되는 주식들의 데이터만을 사용한다.

매수 예측 모델은 TD 알고리즘으로 학습하며, 각 후보 주식에 대한 미래 누적 기대 수익을 수치로 표현하

2) 보상과 다음 상태에 대한 확률 분포

3) 에피소드는 행위자 전이(agent transition) 혹은 시간 단계의 환경된 집합이다.  
4) 일반적으로 이러한 성향을 보이는 주식은 투기적 거래의 대상인 경우가 많은 것으로 알려져 있으며 투자 시 여러 위험 요소를 감수해야 한다.

는 '예측'을 수행한다. 매수 전략은 이 예측치에 근거하여 추천 주식(매수 주식)을 선정하기 위한 임계치를 선정한다<sup>5)</sup>. 이 임계치 이상의 예측치를 가진 주식들이 추천되어 매수된 후 매도 모듈로 전달된다. 매도 모듈은 Q 알고리즘으로 학습하며, 거래에서의 이익을 극대화하기 위해 보유 중인 주식들의 향후 주가에 대한 예측과 수익률, 보유 기간 등 매도 전략에 관련된 요소들을 종합적으로 고려하여 매도 시기를 판단한다. 1일 단위로 각 주식의 보유(hold) 및 매도(sell) 여부를 결정하며, 주어진 상태에서 보유와 매도라는 '행위'에 대한 가치도를 비교하여 의사 결정이 이루어진다는 점에서 TD-학습을 수행하는 매수 예측 모듈과 다르다. 두 모듈 모두 입력 속성들의 상태 공간(state space)이 매우 크므로 신경망에 의한 함수 근사를 수행한다.

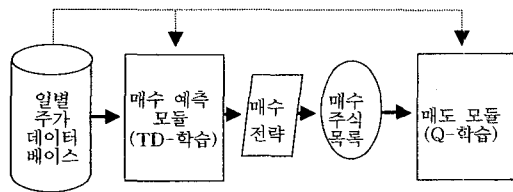


그림 3 거래 시스템의 구조

#### 4. 매수 예측 모듈

앞서 언급한 바와 같이, TD 알고리즘은 주어진 정책  $\pi$  하에서 각 상태의 가치도를 평가한다. [13]에서와 같이 행위<sup>6)</sup>가 환경(시장)에 영향을 미치지 않는다고 가정하면, 주식 시장을 환경으로 간주하여 과거 주가 데이터를 통해 가치도를 평가할 수 있으므로 학습 단계에서 실제 자산을 투자해볼 필요는 없다. 따라서 TD-학습의 관점에서 주식 시장을 모형화하기 위해 남은 문제는 상태와 보상을 정의하는 것이다.

##### 4.1 상태

입력 공간의 표현에 따라 기계 학습의 성능은 크게 달라질 수 있으며, 특히 강화 학습의 경우에는 상태의 표현이 매우 중요하다. 이 논문에서는 주가 변동 분석에 널리 사용되고 있는 기술 분석에 기반하여 상태를 설계한다. 주요 기술 분석 방법론으로는 다우 이론, 엘리오트 파동 이론, 패턴 분석, 그랜빌의 법칙, 진동자(oscillator), 일본

- 5) 매수 전략에는 매수가의 결정도 포함되어야 하지만, 이 논문에서는 추천일의 종가에 주식을 매수한다고 가정하는 단순한 방식을 채택한다.
- 6) 이 논문에서는 매수 매도 등 거래에 관한 의사 결정에 해당한다.

식 봉 차트 기법 등이 있다. 이 중 다우 이론은 중장기적 분석에 적합하지만 단기적인 주가 변동의 예측에는 부적합하며, 엘리오트 이론은 재귀적 정의 등 이론 자체의 복잡성으로 인해 프로그램화하기 힘들다. 또한 패턴 분석은 컴퓨터 분야의 몇몇 기존 연구에서 활용되었으나 패턴의 정의가 주관적인 요소를 많이 포함하고 있어 높은 성능을 기대하기 힘들다. 이 논문에서는 상호 연관성이 높고 수치화된 입력 속성으로의 표현이 용이하다고 판단되는 그랜빌의 법칙과 봉 차트 기법<sup>7)</sup>을 결합하여 사용한다.

그랜빌은 특정 기간 동안의 주가의 평균값인 이동평균(MA: Moving Average)의 관점에서 주가의 미래 추세를 설명하였다. 이동평균은 아래와 같이 정의할 수 있다.

$$MA_t^n = \left( \sum_{i=t-n+1}^t Close_i \right) / n \quad (5)$$

여기서  $Close_t$ 는 시간 단계(거래일)  $t$ 에서의 종가<sup>8)</sup>이다. 다음은 그랜빌이 제안한 8가지 원리(principle) 중 2가지 예이다[14]:

(A) 단기 이동평균선이 하강하여 장기 이동평균선에 근접하면 단기 이동 평균선은 반전하여 상승할 가능성이 있다. (그림 4(A) 참조)

(B) 단기 이동평균선이 하강하여 장기 이동평균선을 하향 돌파하였으나 단기간에 다시 상향 돌파하면 단기 이동평균선은 상승을 계속할 가능성이 있다. (그림 4(B) 참조)

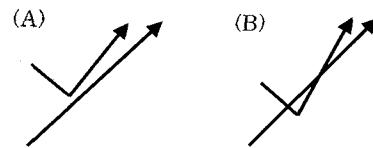


그림 4 그랜빌의 법칙

위의 원리들에서 알 수 있듯이 그랜빌의 법칙은 다분히 추상적인 표현을 사용하였으며, 실제로 이 원리들을 보다 일반화된 형태로 거래 시스템에 적용하기 위해서는 이동평균선의 기울기와 이동평균선간 거리를 표현하는 변수들의 정의가 필요하다. 이동평균선의 기울기  $g_t^n$ 와 이동평균선간 거리  $d_t^{m,n}$  ( $m > n$ )는 다음과 정의된다:

- 7) 일본식 봉 차트 기법은 서양식 차트 기법에 비해 하루 동안(장중) 발생한 가격 변화의 표현과 해석에 중점을 두는 기법이다.
- 8) 하루 동안의 주식 거래 중 마지막으로 체결된 거래의 가격이다. 아울러, 시가는 하루 거래 중 첫 거래의 가격을, 고가는 가장 높게 거래된 가격을, 그리고 저가는 가장 낮게 거래된 가격을 말한다.

$$g_t^n = (MA_t^n - MA_{t-1}^n) / MA_{t-1}^n \quad (6)$$

$$d_t^{m,n} = (MA_t^m - MA_t^n) / MA_t^m \quad (7)$$

일본식 봉 차트 기법[15]을 상태 표현에 활용하기 위해 그림 5의 몸통(body)  $body_t$ , 윗수염(upper shadow)  $us_t$ , 아랫수염(lower shadow)  $ls_t$ 와 전일 대비 증가 변화율(rate of change)  $rc_t$ 를 다음과 같이 정의한다:

$$body_t = (Open_t - Close_t) / Open_t \quad (8)$$

$$us_t = (High_t - \max(Open_t, Close_t)) / \max(Open_t, Close_t) \quad (9)$$

$$ls_t = (\min(Open_t, Close_t) - Low_t) / \min(Open_t, Close_t) \quad (10)$$

$$rc_t = (Close_t - Close_{t-1}) / Open_{t-1} \quad (11)$$

여기서  $Open_t$ ,  $High_t$ ,  $Low_t$ ,  $Close_t$ 는 각기  $t$ 에서의 시가, 고가, 저가, 그리고 종가이다.

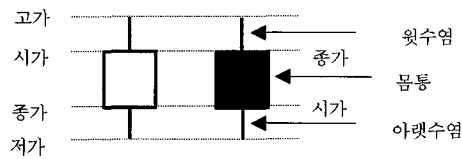


그림 5 일본식 봉의 예

이상의 정의에 기반하여 아래의 벡터를 매수 예측 모델의 상태로 정의한다:

$$\vec{\phi}_s^{TD} = (g_t^{120}, g_t^{60}, g_t^{20}, g_t^{10}, g_t^5, d_t^{120,60}, d_t^{60,20}, d_t^{20,10}, d_t^{10,5}, body_t, body_{t-1}, body_{t-2}, us_t, ls_t, rc_t, rc_{t-1}, rc_{t-2}) \quad (12)$$

#### 4.2 보상

대부분의 시계열 예측을 위한 감독 학습 알고리즘에서는 목표값(target)으로 다음의 식을 사용한다.

$$z(t+h) = y(t+h) \quad (13)$$

즉, 고정 예측 시간 간격(time horizon)인  $h$  시간 단계 후의 시계열 값을 목표값으로 사용한다. 그러나 주가 예측 문제에서 (13)과 같은 목표값을 사용하는 것은 부적절한 측면이 있다. 가격  $y$ 는 보통 긴 시간 주기 동안에는 급격하게 변화하며 주식의 종류에 따라서도  $y$  값이 크게 다를 수 있기 때문에 유효한 모델의 생성이 어렵다. 따라서 목표값으로 다음과 같은 상대적인 가격 변화, 즉 수익률을 사용하는 것이 바람직하다[16].

$$R_h(t) = 100 \cdot \frac{y(t) - y(t-h)}{y(t-h)} \quad (14)$$

(14)에서 시간 간격  $h$ 를 1일로 하면 시간 단계  $t$ 의 즉치 보상  $r_t$ 를 다음과 같이 정의할 수 있다.

$$r_t = 100 \cdot \frac{y_c(t) - y_c(t-1)}{y_c(t-1)} \quad (15)$$

여기서  $y_c(t)$ 는  $t$ 에서의 종가이다.

식 (15)를 보상으로 사용하여, 식 (3)에 의해 특정 주식의 각 시간 단계별 상태의 현재 추정 가치도를 계산할 수 있다. 이러한 추정 가치도는 할인율(discounting factor)  $v$ 에 따라 달라질 수 있다. 만일  $v=0$ 이라면, 다음 날의 주가 변동률을 나타내는 즉각적인 보상만이 반영될 것이다. 그러므로 단기 예측에는 0에 가까운 값을 가지는  $v$ 가 적합하다. 만일  $v=1$ 이라면 어떤 상태 후의 모든 보상은 그 상태로부터의 시간 경과에 관계없이 모두 동일하게 취급되므로  $v$ 가 1에 가까울수록 장기 예측에 적합하다.

주가 예측에 있어서는 현재 시간 단계의 가격과 일정 시간대 후의 가격 사이에 있는 중간 가격들은 모두 의미를 가진다. 따라서 할인율은 주가 예측 문제에 더 적합한 목표값의 표현을 가능하게 한다. 예를 들어, 주식 A와 B의 주가 변동을 보이는 그림 6을 보자. 가치도 함수 (1)에 의하면,  $v < 1$ 이고  $t > 4$ 에 대하여는 보상이 동일하다고 가정할 때, 시간 단계 0에서 A의 목표값이 B의 목표값 보다 크게 된다. 그러나 식 (14)에 의해 계산된 두 주식의 수익률은 시간 간격  $h$ 를 4로 설정했을 경우 모두 20.0이 된다.

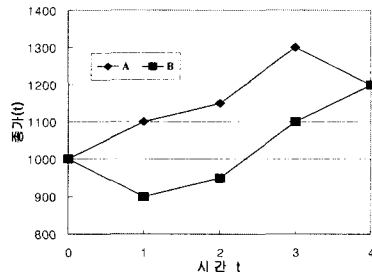


그림 6 시간  $t$ 의 변화에 따른 주가 변동

#### 4.3 신경망에 의한 함수 근사

4.2 절의 정의에 따르면, 주식의 상태 공간은 연속적이다. 따라서 이전에 경험한 상태로부터 경험하지 못한 상태로의 일반화가 필요하다. 기울기-감소 방식(gradient-descent method)에서는 시간  $t$ 에서의 가치도 근사 함수(approximate value function)  $V_t$ 를 고정된 수의 실수 값 요소(real-valued component)로 구성되는 열 벡터,  $\vec{\theta}_t = (\theta_t(1), \theta_t(2), \dots, \theta_t(n))^T$  ( $T$ 는 전치를 의미)를 매개변수 벡터로 갖는 함수로 표현한다. TD의 경우에는 이 벡터가 각각의 학습 예제를 사용하여 다음과 같이 갱신된다:

$$\vec{\theta}_{t+1} = \vec{\theta}_t + \Delta \vec{\theta}_t \quad (16)$$

여기서,  $\Delta \vec{\theta}_t = \eta [r_{t+1} + \gamma V_t(s_{t+1}) - V_t(s_t)] \nabla_{\vec{\theta}} V_t(s_t)$   
 가치도 함수  $V_t$ 가 다음과 같이 매개변수 벡터의 선형 함수라고 하자:

$$V_t(s_t) = \vec{\theta}_t^T \vec{\phi}_{s_t} = \sum_{i=1}^n \theta_t(i) \phi_{s_t}(i) \quad (17)$$

이 경우, 식 (16)의 기울기 항은 다음과 같이 시간  $t$ 에서의 상태 벡터(state vector) 자체가 된다:

$$\nabla_{\vec{\theta}} V_t(s_t) = \vec{\phi}_{s_t} \quad (18)$$

그러나 주식 시장에서의 입력과 출력은 비선형적이므로 선형적 방법은 주가 예측 문제에 적합하지 않다. 강화 학습에서 기울기 기반 함수 근사에 광범위하게 사용되는 비선형적 방식은 오류 역전파 알고리즘(error backpropagation algorithm)을 사용하는 다층 인공 신경망이다. 이 방식에서는 식 (16)의 기울기 항을 계산하기 위해 역전파(backpropagation) 과정을 사용한다.

**4.4 학습 알고리즘**

그림 7은 매수 예측 모듈의 TD-학습 알고리즘을 보인다. 안정성 검증을 거친 모든 주식들의 일별 주가 데이터 중 학습 기간에 해당하는 데이터로 학습을 수행한다<sup>9)</sup>. 특정 주식의 학습 기간 데이터를 하나의 에피소드 생성에 사용한다. 즉, 에피소드는 학습 기간의 첫날(first-day)에 시작하여 마지막 날(last-day)에 종료된다. 마지막 날의 경우에는 목표값  $r_{t+1} + \gamma V_t(s_{t+1})$ 을 0으로 하여 파라미터를 갱신한다. 일반적인 TD 학습과 달리 매수 예측 모듈의 TD 학습에서는 다음 상태가 행위와 무관하게 결정되며, 함수  $Next-state$ 가 다음 거래일의 상태를 반환한다. 보상은 증가 기준 일일 수익률이며, 앞서 기술한 신경망에 의한 함수 근사를 수행한다.

```

V(s)를 임의로 초기화
Repeat (for all stocks)
  t ← first-day
  Repeat
    s_{t+1} ← Next-state(t)
    r_{t+1} ← (Close_{t+1} - Close_t) / Close_t * 100
    δ_t ← r_{t+1} + γ * V(s_{t+1}) - V(s_t)
    θ_{t+1} ← θ_t + η * δ_t * ∇_{θ} V(s_t)
  t ← t + 1
until t > last-day
    
```

그림 7 매수 예측 모듈의 TD-학습 알고리즘

9) 나머지 기간의 데이터는 검증(validation)과 테스트(test) 데이터로 사용한다.

**5. 매도 모듈**

매도 모듈은 매수 예측 모듈 및 매수 전략(입계치)에 의해 매수된 주식들을 효과적으로 매도하는 역할을 담당한다. 매수 예측 모듈과 달리 매도 모듈에서는 행위-가치도를 평가하여 매도 여부를 결정한다. 또한 예측 관련 속성과 함께 수익률, 보유 기간 등을 표현하는 속성도 상태 표현에 포함되어야 한다. 따라서 매수 예측 모듈에 비해 예측 관련 입력 속성의 사용을 최소화하는 것이 불가피하다<sup>10)</sup>.

**5.1 상태, 행위, 그리고 보상**

매도를 위한 Q-학습에서 수익률 및 보유 기간은 매우 중요한 고려사항이다. 실제로 일반 투자자들의 경우, 주식을 매수하는 입장에서는 해당 주식의 주가가 향후 상승할 것인지의 여부가 가장 중요한 의사 결정 요인이지만, 주식을 매도하는 입장에서는 현재의 수익 혹은 손실의 정도가 더욱 중요한 결정 요인으로 작용하는 경우가 많다. 현재 수익률을 상태 표현에 포함시키는 한 가지 방법은 단순히 수익률에 해당하는 실수 값 속성 하나를 할당하는 것이다. 그러나 이 경우 미시적인 수익률의 차이를 효과적으로 학습에 반영할 수 없다. 따라서 이 논문에서는 여러 개의 이진 속성(binary attribute)으로 수익률을 표현하는 방식을 채택하며, 입력 속성간 일관성을 유지하기 위해 다른 모든 속성들도 이진으로 표현한다. 상태의 정의는 아래와 같다.

$$\vec{\phi}_s^Q = (p_{t,1}, \dots, p_{t,10}, h_{t,1}, \dots, h_{t,5}, g_{t,1}^5, \dots, g_{t,4}^5, d_{t,1}^{10,5}, \dots, d_{t,4}^{10,5}, body_{t,1}, \dots, body_{t,4}, rc_{t,1}, \dots, rc_{t,4}) \quad (19)$$

$p_{t,i}$ 는 보유 주식의 매수가 대비 수익률을 이진 표현으로 변환하는 속성들이며  $h_{t,i}$ 는 보유 기간을 표현하는 속성들이다. 표 1에 보여지는 바와 같이 수익률의 각 구간과 보유 기간은  $p_{t,i}$ 와  $h_{t,i}$ 에 사상된다. 나머지 속성들은 4.1에서 정의한 예측 관련 속성 중 단기(short-term) 속성들이다. 행위는 HOLD(보유)와 SELL(매도)의 두 가지가 가능하며, 보상은 매수 예측 모듈에서와 같이 일일 수익률이다.

표 1 수익률과 보유 기간의 이진 표현

수익률 (%)	$p_{t,1} \sim p_{t,10}$	수익률 (%)	$p_{t,1} \sim p_{t,10}$	보유 기간	$h_{t,1} \sim h_{t,5}$
(30, ∞)	1000000000	(-5, 0]	0000010000	1일	10000
(20, 30]	0100000000	(-12, -5]	0000001000	2~3일	01000
(12, 20]	0010000000	(-20, -12]	0000000100	4~6일	00100
(5, 12]	0001000000	(-30, -20]	0000000010	7~10일	00010
(0, 5]	0000100000	(-∞, -30)	0000000001	11일 이상	00001

10) 일반적으로 입력 속성의 수가 너무 많으면 효과적인 함수 근사가 불가능하다.

5.2 학습 알고리즘

그림 8은 매도 모듈의 Q-학습 알고리즘을 보인다. 전체 일별 주가 데이터 중 매수 예측 모듈에 의한 예측치(신경망의 출력)가 양의(positive) 임계치  $T_{train}$  이상인 데이터를 학습 데이터로 사용한다. 하나의 에피소드는 이 데이터로부터 특정 주식의 특정 날짜에 대한 데이터를 무작위로 선택함으로써 시작된다. 주식의 매수는 매수를 결정한 날의 증가에 이루어지는 것으로 가정한다.  $\epsilon$ -greedy에 의해 선택된 행위가 HOLD인 경우에는 가치도 갱신이 이루어지며, 선택된 행위가 SELL인 경우에는 에피소드가 종료된다. 가치도 갱신을 위해 먼저 Next-state 함수가 다음 상태를 반환한다. 동일 주식의 다음 거래일( $t+1$ )의 데이터를 사용하여 보상에 해당하는 증가 기준 일일 수익률을 계산하며, 이에 기반하여  $Q(s, a)$ 를 갱신한다.

```

Q(s, a)를 임의로 초기화
Repeat
  매수 종목 및 매수일  $t_b$ 를 무작위로 생성
   $t \leftarrow t_b$ 
  Repeat
     $a_t \leftarrow \begin{cases} \text{확률 } 1-\epsilon \text{로 } \operatorname{argmax}_a Q(s_t, a) \\ \text{확률 } \epsilon \text{로 무작위 선택} \end{cases}$ 
    If  $a = \text{HOLD}$ 
       $s_{t+1} \leftarrow \text{Next-state}(t)$ 
       $r_{t+1} \leftarrow \frac{\text{Close}_{t+1} - \text{Close}_t}{\text{Close}_t} \times 100$ 
       $\delta_t \leftarrow r_{t+1} + \gamma * \max_{a_{t+1}} Q(s_{t+1}, a_{t+1}) - Q(s_t, a_t)$ 
       $\theta_{t+1} \leftarrow \theta_t + \eta * \delta_t * \nabla_a Q(s_t, a_t)$ 
       $t \leftarrow t + 1$ 
    until  $a = \text{SELL}$ 
  
```

그림 8 매도 모듈의 Q-학습 알고리즘

6. 실험

이 절에서는 R-Trader의 성능을 평가하기 위한 실험 결과를 제시한다. 먼저 매수 예측 모듈의 예측력(predictability)을 테스트하고, 다음으로 감독 학습에 기반하여 구축된 거래 시스템과의 성능 비교 결과를 제시한다. 비교 대상이 되는 시스템은 R-Trader와 같은 입력 속성과 신경망을 사용하여 구축되었으며 S-Trader라는 명칭으로 부르기로 한다. 마지막으로 기존의 강화 학습 기반 거래 시스템과 R-Trader의 성능 비교를 통해 R-Trader의 의의에 관해 분석한다. 평균 거래량 및 주가 변동 성향 검사를 통해 안정성이 검증된 총 300 종목의 일별 주가 데이터로부터 표 2와 같은 실험 데이터를 구성하였다.

표 2 실험 데이터

학습 데이터		검증 데이터		테스트 데이터	
기간	데이터 수	기간	데이터 수	기간	데이터 수
1999년 4월 ~ 2000년 3월	73,200	2000년 6월 ~ 2000년 8월	18,900	2000년 9월 ~ 2001년 11월	91,800

매수 예측 모듈의 TD 학습은  $v$ 를 0.9로 하여 새로운 목표값이 계산될 때마다 즉시 갱신을 수행하는 온라인 갱신으로 진행하였다. 입력 예제의 TD 갱신을 위한 목표 값  $r_{t+1} + \gamma V(S_{t+1})$ 의 계산에 사용되는  $V(S_{t+1})$  항의 근사치를 구하기 위해 이전 예제들로부터 학습된 신경망의 가중치를 사용한다. 각 종목의 학습 기간에 해당하는 전체 데이터를 하나의 에피소드로 간주하였으므로 모두 300개의 에피소드가 사용되었으며, 검증 데이터에 대한 평균 수익률이 감소하기 시작할 때까지 이 에피소드들을 반복적으로 학습시켰다. 신경망은 각기 40개와 20개의 뉴우런을 포함하는 두개의 은닉층(hidden layer)을 사용하였다. 표 3은 테스트 데이터에 대한 TD 예측치의 분포를 보이며 그림 9는 TD 예측의 성능을 보인다.  $R(1) \sim R(20)$ 은 각기 1, 5, 10, 그리고 20일 후에 절삭된 평균 가치도(return)를 나타내며 다음의 공식을 사용하여 계산하였다:

$$R(n) = r_{t+1} + 0.9r_{t+2} + \dots + 0.9^{n-1}r_{t+n} \quad (20)$$

표 3 TD 예측치의 분포

구간 번호	가치도 범위	빈도수	구간 번호	가치도 범위	빈도수
1	(8, ∞)	249	6	(-2, 0]	23,688
2	(6, 8]	1,304	7	(-4, -2]	14,317
3	(4, 6]	4,537	8	(-6, -4]	6,982
4	(2, 4]	13,956	9	(-8, -6]	2,021
5	(0, 2]	25,045	10	(-∞, -8)	398

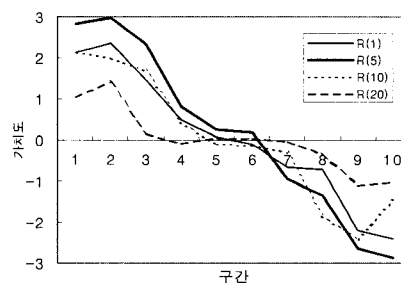


그림 9 TD 예측 성능

$R(5)$ 에 대한 예측 성능이 상대적으로 좋았는데, 이것

은 이 논문에서 사용한 입력 속성들이 기술 분석에 기반하고 있어 비교적 단기적인 주가 추세의 예측에 적합함을 의미한다고 볼 수 있다. 그러나  $R(1)$ , 즉 바로 다음 날의 주가에 대한 예측은 5일간의 미래 추세에 대한 예측보다 어려움을 알 수 있다. 가장 성능이 좋은  $R(5)$ 에 대한 예측의 경우에도, 거래세를 고려할 때, 구간 1~3만이 비교적 유용한 예측 성능을 보였다<sup>11)</sup>. 하지만 이 구간에 속하는 테스트 데이터의 수는 표 2에서 알 수 있듯이 전체 데이터 중 극히 일부에 불과하여, 기술 지표에 의한 주가 변동의 예측은 제한된 경우에 한하여 가능함을 확인할 수 있다.

표 4 R-Trader의 모의 실험 결과

$T_{test}$	최종 수익률(%)	평균 보유 종목수	평균 수익률(%)
3.0	49.65	10.00	1.79
3.5	49.65	10.00	1.79
4.0	60.59	9.53	2.07
4.5	68.32	9.11	2.44
5.0	56.43	8.04	2.65
5.5	37.16	6.12	2.79

표 4는 매도 모듈을 포함하여 R-Trader의 최종적인 성능을 평가하기 위한 모의 투자 실험 결과를 보인다. 매도 모듈은 매수 예측 모듈과 같은 구조의 신경망으로 같은 할인율을 사용하여 학습하였고,  $T_{train}$ 으로는 1.0을 사용하였다<sup>12)</sup>. 28만 에피소드를 학습한 이후 검증 데이터에 대한 수익률이 증가하기 시작하였고, 47만 에피소드 이후에 수익률이 감소하여 학습을 중단하였다. 매수 전략에 해당하는 매수 예측 임계치  $T_{test}$ 로 4.5를 사용한 경우가 가장 높은 최종 수익률을 보였다. 최대 매수 가능 종목수는 10개로 제한하였으며, 익일 매도가 결정된 종목수와 같은 수의 종목을 익일에 매수하는 방식으로 실험하였다. 임계치를 넘는 추천 종목의 수가 매수하고자 하는 종목수보다 많을 경우 예측치가 높은 순서대로 매수하였고, 반대의 경우에는 추천된 모든 종목을 매수하였다. 임계치 4.5 이하에서는 평균 보유 종목수의 증가가 평균 수익률 및 최종 수익률의 감소를 초래하였는데, 이는 예측 성능의 저하로 인해 거래세를 초과하는 수익의 달성이 불가능한 경우가 증가하기 때문인 것

로 해석할 수 있다. 따라서 매수 가능 종목수에 제한을 가하지 않더라도 최적 임계치에는 변화가 없을 것으로 판단된다. 임계치를 5.0과 5.5로 사용한 경우에는, 수익의 달성이 가능한 종목의 예측치가 임계치를 넘지 못하는 경우가 많아져서 최종 수익률이 감소한 것으로 보인다.

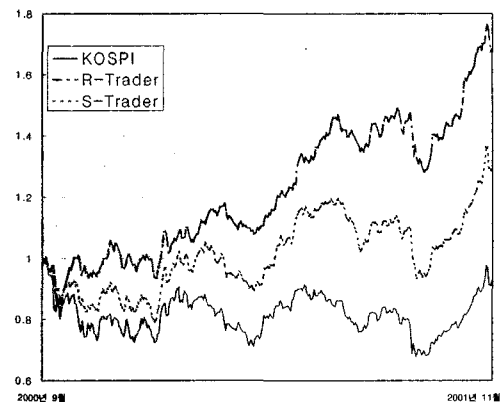


그림 10 R-Trader와 S-Trader의 성능 비교

그림 10은 최적 임계치를 사용한 R-Trader와 S-Trader의 성능을 비교한다. 시장 평균과의 비교를 위해 테스트 기간의 종합주가지수(KOSPI) 변화도 함께 제시하였다. 전반적으로 R-Trader는 종합주가지수와 S-Trader보다 높은 성능을 보였다. 테스트 기간 동안 종합주가지수는 6.98% 하락한 반면 R-Trader는 68.32%, S-Trader는 29.87%의 수익을 달성하였다. 비록 외부 충격에 해당하는 미국 9.11 테러 사건 직후의 폭락 장세에서는 R-Trader도 급격한 수익 감소를 보였지만, 시장 내부 요인에 의한 일반적인 하락 장세에서는 S-Trader에 비해 안정적인 성능을 보였다. 이것은 R-Trader가 위험 관리의 측면에서도 S-Trader에 비해 우수함을 의미한다.

마지막으로, 표 5는 강화 학습 기법을 사용한 기존 거래 시스템들과 R-Trader의 성능을 비교한다. 시스템마다 각기 다른 테스트 기간을 사용하였고, 투자 대상이나 방식에도 차이가 있어 직접적인 비교가 불가능하므로, 지수변동대비 연평균 수익률을 별도 산정하여 비교하였다. Neuneier가 제안한 시스템(T1)[12]은 독일 주식 시장에서의 모의 투자에 Q 학습을 사용하였다. Xiu의 시스템(T2)[8]과 R-Trader 모두 T1에 비해 높은 연평균 수익률을 보였다. T2는 독일 마르크화(DM)와 미 달러(USD) 간 환율 예측을 통한 외환 거래에 의해 수익을

11) 이 논문에서는 주가 상승에 대한 예측에 초점을 두고 있으므로 하락에 대한 예측 성능은 유용하다고 할 수 없다.

12) 매수 예측 모듈의 예측치가 1.0 이상인 예제만으로 학습하였음을 의미한다.



얻었는데, T1에서 사용한 독일 주가지수(DAX)와 환율이 모두 단일 시계열이라는 점에서 비교의 의의가 있다. 연평균 수익률만으로 보면 T2와 R-Trader의 성능이 비슷한 것으로 나타났지만, T2의 경우 마르크화로 환전 시에는 거래세를 적용하지 않았으므로 R-Trader에 비해 낮은 성능을 보이는 것으로 판단된다. 이 두 시스템과는 달리 R-Trader는 단일 시계열인 KOSPI 지수 대신 이 지수의 산정에 포함되는 개별 종목들의 주가에 대한 예측을 활용하였다는 점과, TD 예측 모듈을 매도 모듈과 분리함으로써 예측 성능을 증대하고 투자 자금의 분량을 신중적으로 조절한 점 등이 성능 개선에 기여한 것으로 보인다.

표 5 기존 거래 시스템과 R-Trader의 성능 비교

거래 시스템	지수	지수 변동률	테스트 기간	수익률	지수대비 수익률	지수대비 연평균 수익률
T1	DAX	+16.9	2년 9개월	41.0	24.1	8.2
T2	USA-DM	0	4개월	15.5	15.5	54.1
R-Trader	KOSPI	-7.0	1년 3개월	68.3	75.3	56.7

### 7. 결론

이 논문에서는 강화 학습에 기반한 자동 주식 거래 시스템인 R-Trader를 소개하였다. 주식 매매에 있어 매수의 관점과 매도의 관점이 다르다는 점에 착안하여, 종목 선정을 위한 예측에는 TD 알고리즘을 독립적으로 사용하였다. 매도 모듈은 기존의 강화 학습 알고리즘의 장점인 예측과 전략의 통합적 학습을 수행하기 위해 Q 알고리즘을 사용하였다. 강화 학습의 입력에 적합한 형태로 기술 분석 기법을 변형하여 활용하였으며, 신경망에 의한 가치도 근사를 수행하였다. 실험 결과는 제안한 시스템에 의해 시장 평균을 초과하는 수익의 달성이 가능하고 감독 학습에 기반한 시스템에 비해 우수한 성능을 달성할 수 있음을 보였다.

그러나, 보다 현실적인 주식 거래 시스템을 구현하기 위해서는 몇 가지 사항들이 추가로 고려되어야 한다. 시장 전반의 추세에 대한 예측을 통한 위험 회피 성능의 증대, 시스템에 의한 매수 매도가 시장에 영향을 미치지 않는다는 가정에 대한 검증, 즉 거래량 및 거래 대금과 시스템 성능의 상관 관계에 대한 분석 등이 필요하다. 또한 TD( $\lambda$ )와 같은 여타 강화 학습 알고리즘의 적용 및 유전자 알고리즘, 퍼지 이론, 그리고 카오스 이론 등과의 결합을 통한 시스템 성능의 향상도 추가적인 고려

사항이다.

### 참고 문헌

- [1] B. G. Malkiel, A Random Walk Down Wall Street, Norton, New York, 1996.
- [2] S. T. Chou, H. Hsu, C. Yang, F. Lai, "A stock selection DSS combining AI and technical analysis," Annals of Operations Research 75, pp.335-353, 1997.
- [3] B. Zhang, R. Coggins, M. A. Jabri, D. Dersch, B. Flower, "Multiresolution Forecasting for Futures Trading Using Wavelet Decompositions," IEEE Trans. Neural Networks, vol. 12, pp.765-775, 2001.
- [4] R. J. Kuo, "A Decision Support System for the Stock Market through Integration of Fuzzy Neural Networks and Fuzzy Delphi," Applied Intelligence, 6., pp.501-520, 1998.
- [5] R. Neuneier, "Enhancing Q-Learning for Optimal Asset Allocation," Advances in Neural Information Processing Systems 10, MIT Press, Cambridge, pp.936-942, 1998.
- [6] R. Neuneier, O. Mihatsch, "Risk Sensitive Reinforcement Learning," Advances in Neural Information Processing Systems 11, MIT Press, Cambridge, pp.1031-1037, 1999.
- [7] J. Moody, Y. Wu, Y. Liao, M. Saffell, "Performance Functions and Reinforcement Learning for Trading Systems and Portfolios," Journal of Forecasting, 17(5-6), pp.441-470, 1998.
- [8] J. Moody, M. Saffell, "Learning to Trade via Direct Reinforcement," IEEE Transactions on Neural Networks, 12(4), pp.875-889, 2001.
- [9] G. Xiu, C. Laiwan, "Algorithm for Trading and Portfolio Management Using Q-learning and Sharpe Ratio Maximization," Proceedings of ICONIP 2000, Korea, pp.832-837, 2000.
- [10] R. S. Sutton, A. G. Barto, Reinforcement Learning: An Introduction, MIT Press, Cambridge, 1998.
- [11] C. J. Watkins, Learning from Delayed Rewards. Ph.D. thesis, Cambridge University, 1989.
- [12] M. H. Kalos, P. A. Whitlock, Monte Carlo Methods, Wiley, New York, 1998.
- [13] R. Neuneier, "Optimal Asset Allocation Using Adaptive Dynamic Programming," Advances in Neural Information Processing Systems 8, Cambridge, MA: MIT Press, pp.953-958, 1996.
- [14] R. D. Edwards and J. Magee, Technical Analysis of Stock Trends, John Magee, Inc., 1974.
- [15] S. Nison, Japanese Candlestick Charting Techniques, New York, NY:New York Institute of

Finance, 1991.

- [16] T. Hellstroem, A Random Walk through the Stock Market, Ph.D. theis, University of Umea, Sweden, 1998.



이 재 원

1990년 2월 서울대학교 컴퓨터공학과 학사. 1992년 2월 서울대학교 컴퓨터공학과 석사. 1998년 8월 서울대학교 컴퓨터공학과 박사. 1999년 ~ 현재 성신여자대학교 컴퓨터정보학부 강의전담교수. 관심분야는 기계학습, 전산금융, 자연언어

처리



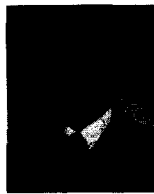
김 성 동

1991년 2월 서울대학교 컴퓨터공학과 학사. 1993년 2월 서울대학교 컴퓨터공학과 석사. 1999년 8월 서울대학교 컴퓨터공학과 박사. 1999년 8월 ~ 2001년 2월 서울대학교 컴퓨터신기술공동연구소 특별연구원. 1999년 8월 ~ 현재 한성대학교 컴퓨터공학부 전임강사. 관심분야는 데이터마이닝, 기계학습, 자연언어처리



이 중 우

1990년 서울대 컴퓨터공학과 졸업(학사). 1992년 서울대 컴퓨터공학과 석사과정 졸업(석사). 1996년 서울대 컴퓨터공학과 박사과정 졸업(박사). 1996년 ~ 1998년 현대전자산업(주) 과장. 1998년 ~ 1999년 현대정보기술(주) 책임연구원. 1999년 ~ 현재 한림대학교 정보통신공학부 조교수. 관심분야는 운영체제, 병렬/분산 운영체제, 클러스터 시스템, 전산금융



채 진 석

1990년 2월 서울대학교 컴퓨터공학과 학사. 1992년 2월 서울대학교 컴퓨터공학과 석사. 1998년 2월 서울대학교 컴퓨터공학과 박사. 1992년 3월 ~ 1997년 2월 서울대학교 공학연구소 조교. 1997년 7월 ~ 1998년 8월 한국학술진흥재단 부설 첨단학술정보센터 선임연구원. 1998년 8월 ~ 현재 인천대학교 컴퓨터공학과 조교수. 관심분야는 인터넷 소프트웨어, 마크업 언어, 한국어정보처리