다중 에이전트 Q-학습 구조에 기반한 주식 매매 시스템의 최적화

김 유 섭† · 이 재 원†† · 이 종 우†††

요 약

본 논문은 주식 매매 시스템을 위한 강화 학습 구조를 제시한다. 매매 시스템에 사용되는 매개변수들은 Q-학습 알고리즘에 의하여 최적화되고, 인공 신경망이 값의 근사치를 구하기 위하여 활용된다. 이 구조에서는 서로 유기적으로 협업하는 다중 에이전트를 이용하여 전역적인 추세 예측과 부분적인 매매 전략을 통합하여 개선된 매매 성능을 가능하게 한다. 에이전트들은 서로 통신하여 훈련 에피소드와 학습된 정책을 서로 공유하는데, 이 때 전통적인 Q-학습의 모든 골격을 유지한다. 실험을 통하여, KOSPI 200에서는 제안된 구조에 기반한 매매 시스템을 통하여 시장 평균 수익률을 상회하며 동시에 상당한 이익을 창출하는 것을 확인하였다. 게다가 위험 관리의 측면에서도 본 시스템은 교사 학습(supervised learning)에 의하여 훈련된 시스템에 비하여 더 뛰어난 성능을 보여주었다.

Optimization of Stock Trading System based on Multi-Agent Q-Learning Framework

Yu-Seop Kim† · Jae-Won Lee†† · Jong-Woo Lee†††

ABSTRACT

This paper presents a reinforcement learning framework for stock trading systems. Trading system parameters are optimized by Q-learning algorithm and neural networks are adopted for value approximation. In this framework, cooperative multiple agents are used to efficiently integrate global trend prediction and local trading strategy for obtaining better trading performance. Agents communicate with others sharing training episodes and learned policies, while keeping the overall scheme of conventional Q-learning. Experimental results on KOSPI 200 show that a trading system based on the proposed framework outperforms the market average and makes appreciable profits. Furthermore, in view of risk management, the system is superior to a system trained by supervised learning.

키워드 : Q-학습(Q-learning), 주식 매매(Stock Trading), 다중 에이전트(Multi Agent), 매수 신호(Buy Signal), 매수 주문(Buy Order), 매도 신호(Sell Signal), 매도 주문(Sell Order)

1. 서 론

금세기 들어 주식 시장에서의 투자자의 수는 날마다 증가하고 있으며, 이들의 매매를 돕는 지능형 의사 지원 시스템 역시 그 필요성이 날로 증가하고 있다. 이동 평균선과 같은 많은 기술적인 지표들이 경제 분야에서의 전문가들에 의하여 새로이 개발되어 왔다[1]. 또한, 통계적인 방법론이나 컴퓨터의 도움을 받는 기계 학습 방법론에 기반한 시스템들이 널리 개발되고 있다. 그러나 많은 수의 교사 학습에 기반한 시스템들은 하나의 통합된 구조에서 매매 정책을 고려하지 않으면서 예측을 최적화 한다는 한계를 보여주었다. 최근에는 대안으로써 강화 학습을 매매 시스템의 최적화에 응용하는 연구가 나타나고 있다[2, 3]. 강화 학습의 목적은 실제 일반적인 교사 학습의 목적인 오류 제곱(squares error)의 합을 최소화하는 것이 아니라, 학습 에이전트가 환경으로부터 최대의 평균 보상을 얻기 위한 최적의 정책을 획득하는 것이다.

본 논문은 다중 협업 에이전트로 구성된 강화 학습 구조를 제시하는데, 이 구조는 보다 더 효과적으로 예측 기준을 매매 정책과 통합시킬 수 있다. 매수와 매도 신호를 발생시키는 에이전트는 변환점 구조(turning-point structure)라 명명되는 행렬을 사용하는데, 이 행렬은 주가의 장기 의존 관계를 모델링한다. 그리고, 하룻동안의 가격 변동을 활용하기 위하여, 주문을 발생시키는 에이전트는 단기 정책을 최적화시킨다. 이때, 인공 신경망과 결합된 Q-학습 방법을 사용하여 에이전트를 학습하여 최적의 정책을 얻을 수 있도록 하였다. 게다가, 정규화 기법을 사용하여 값 근사화 모듈을 학습시켜서 매개변수들의 발산을 방지할 수 있었다. 실험에서는 실제 주식 매매 시스템을 제안된 구조를 사용하여 구현하였는데, 그 시스템은 시장 평균을 상회하는 성능을 보여주었으며 또한 고전적인 교사 학습 알고리즘으로 구현된 시스템보다 더 좋은 성능을 보여주었다.

2장에서는 강화 학습의 개념에 대하여 간략하게 요약하

† 종신회원 : 한림대학교 정보통신공학부 교수
†† 정 회 원 : 성신여자대학교 컴퓨터정보공학부 교수
††† 정 회 원 : (주)아이닉스소프트 기술이사
논문접수 : 2003년 9월 4일, 심사완료 : 2004년 3월 19일

고 현재까지 진행되어온 주식 시장 분석 방법에 대하여 설명한다. 3장에서는 본 시스템의 구조에 대하여 전체적으로 살펴보고 어떻게 협업 에이전트들이 서로 통신을 하는지에 대해서 설명한다. 또한 본 시스템을 위하여 개발된 학습 알고리즘의 보다 자세한 부분을 설명한다. 4장에서는 실험의 구성과 결과에 대하여 논하고 마지막으로 5장에서는 향후 방향과 함께 본 논문을 요약할 것이다.

2. 배경 지식

강화 학습은 목표 지향적인 학습과 의사 결정을 이해하고 자동화시키는 계산적인 접근 방법이다. 본 논문에서는 강화 학습에 대하여 Sutton[4]의 표기법에 따라서 소개할 것이다. 강화 학습 구조, 특히 마르코프 결정 과정(Markov decision process : MDP)에서는 하나의 에이전트가 있고, 이산 시간 과정 $t=0,1,2,\cdots, T$에 서로 상호작용하는 환경이 있다. 에이전트는 환경의 상태 $s_t \in S$에 기반하여 행동(action) $a_t \in A$를 정책 π로부터 선택한다. 만일 어떤 행동이 에이전트로부터 선택되었다면, 환경은 그 상태를 a_t에 반응하여 s_{t+1}로 변환하고 또한 보상(reward) $r_{t+1} \in R$을 발생시킨다.

만일 단일 단계(one-step) 상태 변이 확률과 단일 단계 기대 보상 모델이 유용하다면, 환경은 다음과 같이 완전하게 서술될 수 있다.

$$P^{a}_{ss'} = Pr\{s_{t+1}=s' \mid s_t=s, a_t=a\} \quad (1)$$

$$r^{a}_{s} = Pr\{r_{t+1} \mid s_t=s, a_t=a\} \quad (2)$$

여기서 $s, s' \in S$이고 $a \in A$이다. 여기서 에이전트의 목적은 최적의 정책을 학습하는 것인데, 이 때 정책이란 상태와 행동을 매핑시키는 것이다. 또한 이 정책은 결국 행동-값 함수인 $Q^{\pi}(s,a)$라 불리는 상태-행동 쌍 (s,a)로부터 기대 할인 미래 보상을 최대화시킨다. 시작 상태 s_0로부터 최종 상태 s_T까지 상호작용의 역사로 정의되는 에피소드 작업을 가지고 $Q^{\pi}(s,a)$는 다음과 같이 정의된다.

$$Q^{\pi}(s,a) = E_{\pi}\{r_t + \gamma r_{t+1} + \cdots + \gamma^{T-t-1} r_T \mid s_t=s, a_t=a\} \quad (3)$$

여기서 $0 \le \gamma \le 1$은 할인율(discount-rate)이다. 그리고 최적 정책 Q^{*}은 모든 s와 a에 대하여 다음과 같이 정의된다.

$$Q^{*}(s,a) = max_{\pi} Q^{\pi}(s,a) \quad (4)$$

현재 사용되고 있는 최적 정책을 학습하는 알고리즘은 매우 많은 종류가 있다. 그러나 백업이 어떻게 이루어지는가에 따라서 이들 알고리즘은 다음과 같이 크게 3가지 종류로 나누어 볼 수 있다. 첫째, 동적 프로그래밍을 사용하는 것이다. 동적 프로그래밍은 전체 백업을 사용하고 동시에 언제나 수렴하는 특징을 가지고 있다. 그러나 이것은 환경에 대한 정확한 모델이 있어야 하는 한계를 가지고 있다. 둘째로는, 몬테 카를로 기법이 있다. 이 기법은 전체 에피소드의 일부 샘플 백업만을 사용하여 어떠한 모델도 필요가 없으나 수렴을 하기 까지에는 엄청난 양의 에피소드가 있어야 하는 문제를 가지고 있다. 마지막으로, 일시적 차이(temporal difference : TD)를 사용하는 방법이 있다. 이 방법은 앞의 두 알고리즘을 적절히 혼합한 형태로서, n-단계 샘플과 currently learned value model[4, 5]로 불리는 부트스트래핑을 사용한다. 실제로, TD 방식은 다루기가 용이하기 때문에 널리 사용되고 있으나 반면에 수렴을 위해서는 많은 주의를 필요로 한다[6].

현재까지는 강화학습을 주식 시장 분석에 사용한 예가 많지 않았다. Neuneier는 몇 가지 가설과 단순화 가정을 시장의 특성에 적용시킴으로써 금융 시장을 MDP로 공식화하였다[2]. 그는 또한 [7]에서 알려진 위험 회피 성향에 선호 개념을 추가하여 Q-학습을 보완하였다. 그는 자산 분배에 보다 초점을 맞추어 연구를 진행하였는데, 여기서의 자산 분배란 DAX 또는 DM으로 자신의 포지션을 어떻게 변환시키는가에 대한 문제를 뜻하였다. Moody는 강화 학습 구조를 순환 인공 신경망으로 변환하였다[8, 9]. 정책은 이들의 구조안에서 시간에 걸친 역전달(back propagation) 방식으로 직접 갱신되는데, 이 방식은 순환 인공 신경망의 매우 유명한 학습 방법이다. 그는 훈련된 자산 배분기가 그것의 포지션을 S&P 500 또는 T-Bill 시장으로 성공적으로 변경함으로써 수익을 창출할 수 있음을 보여주었다. Xiu 또한 Q-학습을 이용한 포트폴리오 관리 시스템을 제시하였다[10]. 이 연구에서는 절대 수익과 상대 위험-적용 수익이라는 두개의 성능 함수를 사용하였기 때문에, 두개의 네트워크가 훈련 과정에서 사용되었다.

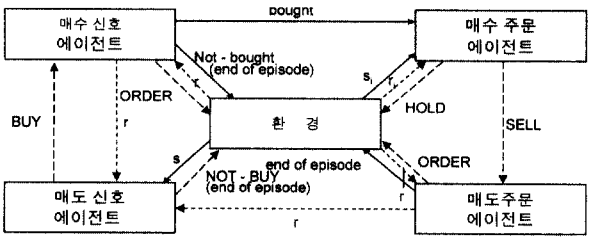
본 논문에서는 두 시장 사이의 자산 분배 보다는 하나의 주식 시장에서의 주식 매매에 대한 연구를 진행하였다. 또한 대부분의 주식 시장에 관한 강화 학습 공식들은 하나의 에이전트만을 구성하였으나 본 논문에서는 다중 에이전트를 사용하여 강화 학습을 공식화하였다.

3. 시스템의 구조 및 제반 알고리즘

이 장에서는 제안된 구조의 전체적인 골격과 본 구조에서 사용된 Q-학습 알고리즘에 대하여 보다 상세하게 설명한다.

3.1 전체 구조

본 논문에서 제안된 다중 에이전트 구조는 (그림 1)과 같다. 이 구조는 기본적으로 투자 이익의 극대화를 목표로 삼고 있는데, 이를 위해서는 주가의 전역적인 추세 뿐만 아니라 주식의 장중 가격 변동역시 고려하여야 한다. 각 에이전트들은 그들 자신만의 고유한 성취 목표를 가지고 있으며 학습 과정에서 에피소드를 공유하기 위하여 서로 상호작용을 일으키는데 각 에이전트의 역할은 다음과 같다.



(그림 1) 시스템 전체 구조

- 매수 신호 에이전트 : 이 에이전트는 매수 신호를 생성하기 위하여 주식의 장단기 상태 정보를 추정하여 예측을 실행한다.
- 매수 주문 에이전트 : 이 에이전트는 주식의 단기 정보를 추정하여 매수 호가를 결정한다. 여기서 매수는 주어진 가격에서 주식을 사는 주문을 말한다.
- 매도 신호 에이전트 : 이 에이전트는 매도 신호를 생성하기 위하여 주식의 상태와 관련한 장단기 정보와 현재의 수익을 추정함으로써 예측을 실행한다.
- 매도 주문 에이전트 : 이 에이전트는 주식의 상태와 관련한 단기 정보를 추정함으로써 매도 호가를 결정한다. 여기서 매도는 주어진 가격에서 주식을 파는 주문을 말한다.

매도 신호 에이전트는 매도를 위한 예측 뿐만 아니라 매매 정책을 전체적으로 고려한다는 점에서 매수 신호 에이전트와는 그 성격이 다르다. 그리고 매수/매도 주문 에이전트들은 예측을 실행하지 않는데, 그 이유는 그들의 유일한 목적은 매매의 성공률을 향상시키기 위하여 최적 주문 실행을 위한 정책을 제공하는데 있다.

3.2 상태, 행동, 보상

기계 학습에서 납득할만한 성능을 얻는데 있어서 가장 중요한 요인 중 하나는 입력 공간을 보다 효과적으로 표현하는 것이다. 특히, 강화 학습은 상태, 행동, 그리고 보상을 설계하는 것이 매우 중요한 요소이다. 본 논문에서는 이들 요소들을 다음과 같이 설계하여 사용하고 있다.

본 논문에서 우리는 변곡점 구조를 표현하는 이항 행렬을 적용하였다. 이 행렬은 5일 이동 평균선의 상하위 변곡점들에 기반하여 장기 의존관계를 요약한다. (그림 2)는 이 행렬의 한 예를 보여주고 있다. 이 행렬에서 열은 참조 날짜(reference day)로부터 변곡점 날짜의 전치(displacement)를 표현하고 있으며 동시에 행은 참조 날짜로부터 변곡점의 가격의 전치를 표현하고 있다. 표에서 'N'으로 표시된 슬롯은 KOSPI에 등록되어 있는 주식의 가격이 해당 항목에 결코 도달하지 못하기 때문에 적중될 수 없는 경우를 의미한다[1]. 변곡점을 위한 두 개의 행렬을 표현하는데 필요한 비트의 수는 305개가 되는데, 이 때 'N'으로 표시된 항목들은 제외된다. 이러한 임시 구조에 더하여, 매수 신호 에이전트는 <표 2>에서 보여지는 단기 기술 지표들을 주문 에이전트와 공유한다. 결국 매수 신호 에이전트의 상태는 353개의 이항 비트들로 구성이 된다. 매도 신호 에이전트는 몇 개의 비트를 더 가지고 있는데, 이들 비트들은 <표 1>에서 보여지는 것과 같이 주식의 보유 기간동안의 현재 수익률을 표현한다. 본 논문에서는 수익률은 +30%에서 -20% 사이의 값으로 임의 제한하였다. 그래서 주식을 보유하고 있는 동안, 수익률이 +30% 이상으로 증가하거나 -20% 이하로 감소하면 매도 신호 에이전트는 무조건 주식을 매도하도록 하였다.

	2	3	5	8	13	21	34	55	89		2	3	5	8	13	21	34	55	89
-89	N	N	N	N	0	0	0	0	0	-89	N	N	0	0	0	0	0	0	0
-55	N	N	0	0	0	0	0	0	0	-55	N	0	0	0	0	0	1	1	1
-34	N	0	0	0	0	0	1	1	1	-34	0	0	0	0	0	0	1	1	1
-21	0	0	0	0	0	1	0	1	1	-21	0	0	0	0	0	1	0	0	1
-13	0	0	0	0	0	0	0	0	1	-13	0	0	0	1	1	0	0	0	1
-8	0	0	0	0	1	0	0	0	1	-8	0	0	0	0	0	0	0	0	0
-5	0	0	0	0	0	0	0	0	0	-5	0	0	0	0	0	0	0	0	0
-3	0	0	0	0	0	0	0	0	0	-3	0	0	0	0	0	0	0	0	0
-2	0	0	0	0	0	0	0	0	0	-2	0	0	0	0	0	0	0	0	0
+2	0	0	0	0	0	0	0	0	0	+2	0	0	0	0	0	0	0	0	0
+3	0	0	0	0	0	0	0	0	0	+3	0	0	0	0	0	0	0	0	0
+5	0	0	0	0	0	0	0	0	0	+5	0	0	0	0	0	0	0	0	0
+8	0	0	0	0	0	0	0	0	0	+8	0	0	0	0	0	0	0	0	0
+13	0	0	0	0	0	0	0	0	0	+13	0	0	0	0	0	0	0	0	0
+21	0	0	0	0	0	0	0	0	0	+21	0	0	0	0	0	0	0	0	0
+34	0	0	0	0	0	0	0	0	0	+34	N	0	0	0	0	0	0	0	0
+55	0	0	0	0	0	0	0	0	0	+55	N	0	0	0	0	0	0	0	0
+89	N	N	N	0	0	0	0	0	0	+89	N	N	N	N	0	0	0	0	0

상위 변곡점 하위 변곡점

(그림 2) 변곡점 구조 사례('N' : not available)

〈표 1〉 매도 신호 에이전트가 현재 수익률을 표현하는데 필요한 추가 비트들(수익률은 100×(오늘의 종가-매수가)/매수가)

수 익	코 딩	수 익	코 딩
+(0~5)	00000001	-(0~3)	00010000
+(5~12)	00000010	-(3~7)	00100000
+(12~20)	00000100	-(7~12)	01000000
+(20~30)	00001000	-(12~20)	10000000

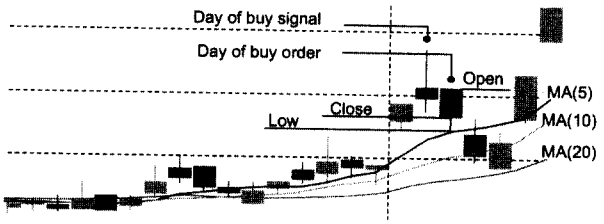
〈표 2〉 주문 에이전트의 상태 및 행동 코딩

상 태	비 트	행 동	코 딩
5일 이동평균 이격도	8	-12%	0000001
10일 이동평균 이격도	6	-7%	0000010
20일 이동평균 이격도	6	-3%	0000100
5일 이동평균 기울기(MA(5))	8	0%	0001000
10일 이동평균 기울기(MA(10))	6	+3%	0010000
20일 이동평균 기울기(MA(20))	6	+7%	0100000
몸 통	8	+12%	1000000

신호 에이전트들은 단지 두 종류의 행동만을 취한다. 매수 신호 에이전트는 NOT-BUY 또는 BUY 행동만을 취하

1) KOSPI(한국증권거래소)에서는 당일 주가의 변동폭이 (+15%, -15%)로 제한되어 있다.

고 매도 신호 에이전트는 HOLD 또는 SELL 만을 취한다. 그리고 보상의 계산은 다음과 같이 이루어진다. 매수 신호 에이전트는 그것이 NOT-BUY를 취하는 동안 0 수준의 보상을 받는다. 만일 그것이 BUY를 취한다면 보상의 계산은 매도 주문 에이전트가 주식을 매도할 때까지 유예된다. 주가의 일간 변화율은 매도 신호 에이전트가 HOLD를 취하는 동안 주어진다. 그러나 그것이 SELL을 취하게 되면 0 수준 보상이 주어지는데, 이는 신호가 그것이 시장에서 나간다는 것을 의미하기 때문이다. 그리고 매도가가 매도 주문 에이전트에 의하여 결정될 때, 매수 신호 에이전트는 보상으로 수익률을 받게 되는데 이는 매매 비용을 고려한 수익률이다.



(그림 3) 일간 가격 변동 사례

주문 에이전트는 <표 2>의 상태와 가능한 행동을 공유한다. 주어진 상태에 대하여, 매수 주문 에이전트는 최적의 호가를 생성하고자 하는데, 최적의 호가는 다음날의 주가변화의 범위내에서 최저가를 말한다. 매우 높은 변동성을 가지고 있는 주식을 매매할 때에는, 매수가가 매매의 최종 이익에 매우 심각한 영향을 미친다. (그림 3)은 이러한 부분의 사례를 보여준다. 여기서 최저 호가는 손절매의 위험성을 감소시켜준다. 매수 주문 에이전트의 보상은 다음과 같이 정의된다.

매수 신호 에이전트 :
1. 환경이 s를 생성한다.

$$a \leftarrow \begin{cases} argmax_a Q(s,a) & with\ prob.\ 1-\varepsilon, \\ random\ selection\ with\ prob. & \varepsilon \end{cases}$$

만일 a = NOT-BUY 이면 에피소드는 종료하고 1로 복귀
그렇지 않으면, 매수 주문 에이전트를 호출하고 다른 에이전트가 호출할 때까지 대기.
만일 호출이 매수 주문 에이전트로부터 오면
r ← 0
그렇지 않으면,

$$r \leftarrow ((1-TC)\times SP - BP)/BP$$

$$\delta \leftarrow r - Q(s,a);\quad \theta \leftarrow \theta + \eta \times \delta \times \nabla_\theta Q(s,a)$$

에피소드를 종료하고 1로 복귀

매수 주문 에이전트 :
2. 환경이 s를 생성한다.

$$a \leftarrow \begin{cases} argmax_a Q(s,a) & with\ prob.\ 1-\varepsilon, \\ random\ selection\ with\ prob. & \varepsilon \end{cases}$$

$$d = MA(5) + \frac{a}{100} \times MA(5) - Low$$

만일 d ≥ 0 이면,

$$r \leftarrow e^{-100\times d/Low};\ \delta \leftarrow r - Q(s,a);\ \theta \leftarrow \theta + \eta \times \delta \times \nabla_\theta Q(s,a)$$

BP를 가지고 매도 신호 에이전트를 호출, 여기서 BP = MA(5) + a/100×MA(5).
그렇지 않으면

$$r \leftarrow 0;\ \delta \leftarrow r - Q(s,a);\ \theta \leftarrow \theta + \eta \times \delta \times \nabla_\theta Q(s,a)$$

매수 주문 에이전트 호출.

* SP : 매도가, BP : 매수가, RC : 종가의 변화율, TC : 매매비용, MA(5) : 5일 평균가

(그림 4) 매수 관련 에이전트에 해당하는 Q-학습 알고리즘

$$r = \begin{cases} e^{-100*d/Low} & bid - price \geq Low \\ 0 & otherwise \end{cases} \quad (5)$$

여기서 $d = bid\text{-}price$(매수호가) $- Low$[2](저가)이다. 만일 d가 0이라면, 에이전트는 최대 보상을 받게 된다. 유사하게 매도 주문 에이전트의 보상은 $High$(고가)을 사용하여 정의된다.

$$r = \begin{cases} e^{100*d/High} & offer - price \leq High \\ 0 & otherwise \end{cases} \quad (6)$$

여기서 $d = offer\text{-}price$(매도호가) $- High$[3]이다.

3.3 학습 알고리즘

에피소드는 환경으로부터 랜덤으로 선택된 특정 일자에서 하나의 주식에 대하여 시작한다. 만일 매수 신호 에이전트가 NOT-BUY를 주어진 주식의 상태에 대한 응답으로써 선택한다면 에피소드는 종료하고 또다른 에피소드가 시작한다. 그러나 만일 매수 신호 에이전트가 BUY를 취한다면, 그 에이전트는 매수 주문 에이전트를 불러낸다. 매수 주문 에이전트는 동일한 거래일의 주식의 상태들을 참조하고 매수 호가를 결정한다. 만일 주식을 구입할 수 없다면 에피소드는 종료하고 매수 신호 에이전트를 0 수준의 보상을 부여하며 호출한다. 그리고 매도와 관련한 전체 알고리즘은 (그림 5)에서 볼 수 있다. 각각의 에이전트는 보상, 델타, 그리고 Q-값에 대한 방정식을 가지고 있다. 그러나 이들 모두는 편의를 위하여 동일한 표기법으로 표현된다. 시간 지수 역시 이와 마찬가지의 이유로 간략화되었다. 만일 매도 주문 에이전트가 그것이 제시한 가격으로 매도하는데 실패한다면, 에이전트는 종가로 매도를 한다. 또한 갱신 규칙과 함수 근사치는 일반적인 Q-함수에 기반한다.

그리고 상태-행동 쌍(2^{353}개)의 탐색 공간이 매수 신호 에이전트가 테이블에서 Q-값을 유지하기에 지나치게 크기 때문에, 본 논문에서는 Q-값의 근사화를 위하여 인공 신경망을 사용하였다. 이론적으로 Q-테이블의 단순 선형 근사화가 발산하는 경우가 발생한다. Baird는 이러한 근사화의 발산을 방지하기 위하여 약간의 이론적인 방향에 대하여 토의하였다[6]. 본 논문에서는 이러한 문제를 보다 간단하게 만들기 위하여 다음과 같은 정규화된 기울기 감소(regularized gradient descent)를 소개하였다.

$$\nabla_{\theta_i} \widetilde{Q}(s,a) = \begin{cases} \nabla_{\theta_i} Q(s,a) & \text{if } \theta_i = bias \\ \nabla_{\theta_i} Q(s,a) + \nu \times \theta_i & otherwise \end{cases} \quad (7)$$

여기서 ν는 가중치의 쇠퇴 정도를 제어하는 상수값이다. 정규화된 기울기 감소는 인공 신경망 학계에서는 매우 인

2) 다음날 최저가.
3) 다음날 최고가.

기있는 기법으로서 매개변수의 구성원들이 무한 증가하는 것을 방지할 수 있다.

매도 신호 에이전트 :
3. 환경이 s를 생성한다.

$$a \leftarrow \begin{cases} argmax_a Q(s,a) & with\ prob.\ 1-\varepsilon, \\ random\ selection\ with\ prob. & \varepsilon \end{cases}$$

만일 a = HOLD 이면,

$s' \leftarrow Action(s,a)$; $r \leftarrow RC$;

$\delta \leftarrow r + \gamma \times \max_{a'} Q(s',a') - Q(s,a)$; $\theta \leftarrow \theta + \eta \times \delta \times \nabla_\theta Q(s,a)$

3으로 이동

그렇지 않으면, 매도 주문 에이전트 호출

매도 주문 에이전트 :
4. 환경이 s를 생성한다.

$$a \leftarrow \begin{cases} argmax_a Q(s,a) & with\ prob.\ 1-\varepsilon, \\ random\ selection\ with\ prob. & \varepsilon \end{cases}$$

$$d = MA(5) + \frac{a}{100} \times MA(5) - High$$

만일 d ≤ 0 이면,

$r \leftarrow e^{-100 \times d/High}$; $\delta \leftarrow r - Q(s,a)$; $\theta \leftarrow \theta + \eta \times \delta \times \nabla_\theta Q(s,a)$

BP와 SP($MA(5) + \frac{a}{100} \times MA(5)$)를 가지고 매수 신호 에이전트 호출

그렇지 않으면,

$r \leftarrow 0$; $\delta \leftarrow r - Q(s,a)$; $\theta \leftarrow \theta + \eta \times \delta \times \nabla_\theta Q(s,a)$

BP와 SP(종가)를 가지고 매수 신호 에이전트 호출.

* SP : 매도가, BP : 매수가, RC : 종가의 변화율, TC : 매매비용, MA(5) : 5일 평균가

(그림 5) 매도 관련 에이전트에 해당하는 Q-학습 알고리즘

4. 실 험

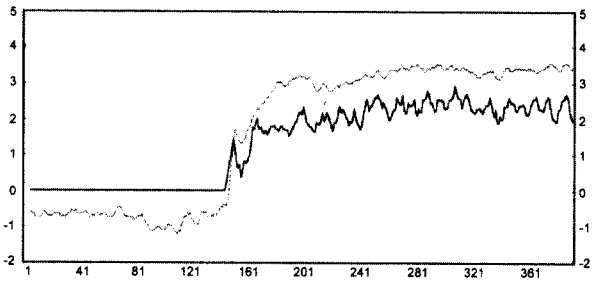
본 논문에서는 제안된 구조에 기반하여 구축된 주식 매매 시스템을 다른 매매 시스템과 비교하였는데, 비교 대상 시스템은 3장에서 설명된 것과 동일한 입력 공간을 가진 인공 신경망을 사용하여 교사 학습 방법으로 훈련되었다. 편의를 위하여 우리가 제안한 시스템을 MAQ이라 하였고 비교 대상이 된 시스템을 SNN이라 하였다.

<표 3>은 데이터 집합을 보여준다. MAQ의 네트워크 구조는 40×20개의 뉴런으로 구성된 2개의 은닉 계층을 가진다. 그리고 학습 비율의 감소 전략은 채택하지 않았고 모든 에이전트에 대하여 학습 비율을 $\eta = 0.05$로 고정시켰다. 또한 가중치 감축 상수는 약간의 기초적인 시도를 한 후에 $\nu = 0.2$로 고정하였다. 할인 요인(discount factor) γ은 0.95로 하였고 조사 요인(exploration factor) ε은 0.1로 하였다. 그리고 시스템은 20,000 에피소드 각각이 실험된 후에 검증 집합에 대하여 검증 작업을 거치게 하였다.

〈표 3〉 데이터 집합의 분할과 기타 명세

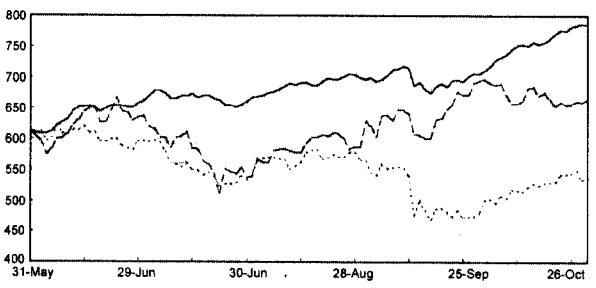
분 할	기 간	크 기
훈련 집합	1999년 1월~2000년 12월	32,019
검증 집합	2001년 1월~2001년 5월	6,102
테스트 집합	2001년 6월~2001년 10월	6,213

(그림 6)은 MAQ의 훈련 성능의 경향을 보여주고 있다. 2,800,000 에피소드가 실험되기 전에, 매매는 훈련 집합에 있는 20,000 에피소드 중에서 50회 정도로 구성되는데 검증 집합에 대해서는 구성하지 않는다. 이는 본 시스템이 오직 ε-정책의 랜덤 조사를 통해서만 매매를 시도하는 것을 말한다. 첫 번째 기간에서는, 매매는 약 -1.2%~-0.1% 정도의 손실을 유발하였다[4]. 그리고 2,800,000 에피소드 이후에는, 매매 횟수와 수익이 훈련 및 검증 데이터 집합에 대하여 점차로 증가하기 시작하였다. 이는 본 시스템이 탐욕 전략(greedy policy)에 의하여 주식을 매매하기 시작하였으며, 이들 매매로부터 수익이 발생하기 시작하였음을 보여준다. 그러나 5,900,000 에피소드 이후에는 약간의 의미 있는 평균 수익률의 하락이 검증 집합에서 보여지는데, 그렇기 때문에 이 시스템은 이 시점에서 훈련을 중단하고 테스트 집합에 적용되었다.



(그림 6) 20,000 에피소드 동안 유발된 평균 수익률

MAQ과 SNN 두 시스템 모두 5개의 종속매매자를 가지고 있는데 이들은 개별적으로 자신들이 속해있는 시스템의 전략에 따라서 그들의 자산을 매매한다. 매매 시스템은 주식을 평가하며 상위에 랭크된 매수 후보들을 수집한다. 이러한 후보 풀로부터 목표 주식이 랜덤하게 각각의 종속 매매자에게 분배가 된다. 몇단계의 매매를 거친 후에, 개별 종속 매매자의 투자금이 통합되고 다시 동일 금액으로 나누어 재분배한 후 매매는 계속된다. MAQ가 주문 에이전트의 결정을 따르는 반면에 SNN은 주식을 시가에 사고 판다.



(그림 7) SNN(대쉬선)과 MAQ(실선)의 수익률. 각 시스템은 보다 정확한 비교를 위하여 KOSPI(점선)가 612포인트일 때를 시작점으로 하여 매매를 시작하였음.

(그림 7)은 SNN과 MAQ의 테스트 집합에서의 성능을 보여주고 있다. 테스트 집합의 기간에서 시작일의 KOSPI 지수는 612 포인트였다. 전체적으로 본다면, MAQ는 시장

4) 한국 주식 시장에서 매매 비용은 0.5%이다.

지수와 SNN 모두 보다 뛰어난 성능을 보여준다고 볼 수 있다. 테스트 기간이 끝난 뒤에 MAQ의 자산은 28.26% 증가하여 785 포인트가 되었으나 SNN은 8.49% 증가하여 664 포인트가 되었을 뿐이었다.

위험 관리의 측면에서도 MAQ는 SNN보다 뛰어났다. 테스트 기간중에 KOSPI에는 두 번의 충격이 있었다. 하나는 6월과 7월 사이에 있었던 내부적인 충격으로 시장이 통상적인 침체를 맞은 것이었다. SNN의 수익률은 이 충격으로 인하여 심각하게 감소하였는데 MAQ은 상대적으로 적은 손실을 보게 됨으로써 이 충격을 견딜 수 있었다. 또 다른 충격은 외부 충격으로써 2001년 9월 11일에 있었던 비극이었다. 이 비극이 있고 나서 한주동안 한국 주식 시장은 패닉에 가까운 매도 현상을 겪게 되었다. 두 시스템 모두 이 짧은 기간동안 매우 심각한 손실을 보게 되었는데 9월이 끝나고 나서 시장은 다시 상승장으로 변환하게 되었다. 이 기간에 MAQ의 수익률은 꾸준하게 증가한 반면, SNN은 어느 정도 출렁거리는 모습을 보여주었다.

5. 결 론

우리는 본 논문에서 강화 학습하에서 체계화된 다중 협업 에이전트를 가진 주식 매매 시스템의 구조를 제안하였다. 또한 우리는 장기 주가 변동과 관련한 역사적인 정보를 요약하기 위하여 새로운 데이터 구조를 설계하였다. 본 시스템은 교사 학습으로 훈련된 시스템에 비하여 한국 주식 시장에서 높은 수익과 의미있는 위험 관리를 가능하게 하였다. 그러나, 이 시스템을 실제 주식 매매 시스템에 연동하여 활용하기 위해서는 추가적인 현실적 고려가 필요하다. 이러한 고려에는 포트폴리오의 개수, 개별 포트폴리오에 대한 자산의 분배, 그리고 주식 시장의 추세의 적용 등이 포함된다. 강화 학습은 매우 장래성이 좋은 방법임이 틀림 없으나 이러한 고려들을 실제 적용하는 것은 매우 복잡한 문제를 야기한다. 그래서 강화 학습을 이러한 고려들과 통합하여 공식화하는 것은 향후 연구 과제가 될 것이다.

참 고 문 헌

[1] S. M. Kendall and K. Ord, "Time Series," Oxford, New York, 1997.

[2] R. Neuneier, "Enhancing Q-Learning for Optimal Asset allocation," Advanced in Neural Information Processing System, 10, MIT Press, Cambridge, pp.936-942, 1998.

[3] J. Lee, "Stock Price Prediction using Reinforcement Learning," Proc. of the 6th IEEE International Symposium on Industrial Electronics, 2001.

[4] R. S. Sutton and A. G. Barto, "Reinforcement Learning : An Introduction," MIT Press, Cambridge, 1998.

[5] M. Jaakkola, M. Jordan and S. Signh, "On the Convergence of Stochastic Iterative Dynamic Programming Algorithms," Neural Computation, 6(6), pp.1185-2201, 1994.

[6] L. C. Baird, "Residual Algorithms : Reinforcement Learning with Function Approximation," Proc. of Twelfth International Conference on Machine Learning, Morgan Kaufmann, San Francisco, pp.30-37, 1995.

[7] R. Neuneier and O. Mihatsch, "Risk Sensitive Reinforcement Learning," Advances in Neural Information Processing Systems, 11, MIT Press, Cambridge, pp.1031-1037, 1999.

[8] J. Moody, Y. Wu, Y. Liao and M. Saffell, "Performance Functions and Reinforcement Learning for Trading Systems and Portfolios," Journal of Forecasting, 17(5-6), pp.441-470, 1998.

[9] J. Moody and M. Saffell, "Learning to Trade via Direct Reinforcement," IEEE Transactions on Neural Networks, 12(4), pp.875-889, 2001.

[10] G. Xiu and C. Laiwan, "Algorithm for Trading and Portfolio Management Using Q-learning and Sharpe Ratio Maximization," Proc. of ICONIP 2000, Korea, pp.832-837, 2000.

김 유 섭

e-mail : yskim01@hallym.ac.kr

1992년 서강대학교 전자계산학과(학사)

1994년 서울대학교 컴퓨터공학과(석사)

2000년 서울대학교 컴퓨터공학과(박사)

2000년~2001년 서울대학교 컴퓨터신기술 공동연구소 전문연구요원

2001년 (주)아이시티 연구소장

2001년~2002년 이화여자대학교 과학기술대학원 연구전임강사

2002년~현재 한림대학교 정보통신공학부 조교수

관심분야 : 전산금융, 자연언어처리, 기계번역, 데이터마이닝, 기계학습

이 재 원

e-mail : jwlee@cs.sungshin.ac.kr

1990년 서울대학교 컴퓨터공학과(학사)

1992년 서울대학교 컴퓨터공학과(석사)

1998년 서울대학교 컴퓨터공학과(박사)

1999년~현재 성신여자대학교 컴퓨터정보 공학부 전임강사

관심분야 : 전산금융, 인공지능, 기계학습, 자연어처리, 컴퓨터뮤직

이 종 우

e-mail : jwlee44@daisy.kw.ac.kr

1990년 서울대학교 컴퓨터공학과(학사)

1992년 서울대학교 컴퓨터공학과(석사)

1996년 서울대학교 컴퓨터공학과(박사)

1996년~1999년 현대전자산업 선임연구원

1999년~2002년 한림대학교 정보통신 공학부 조교수

2002년~2003년 광운대학교 컴퓨터공학과 조교수

2004년~현재 (주)아이닉스소프트 기술이사

관심분야 : 전산금융, 군집컴퓨팅, 분산 병렬시스템, 시스템 소프트웨어