

통합 다중 시뮬레이션에 의한 신경망 기반 주식 거래 시스템의 성능 최적화

이재원[†] · 오장민^{††}

요약

기계 학습 등 인공 지능 기법의 발전에 힘입어 지능형 주식 거래 시스템에 관한 많은 연구가 이루어져 왔다. 그러나 현실 주식 거래에서 적절한 거래 정책의 수립이 거래의 결과에 커다란 영향을 미치는 중요 요소로 작용하고 있음에도 불구하고, 기존의 연구에서는 예측 모듈의 예측 성능 향상에 주력하였거나, 거래 정책을 다룬 경우라도 예측 모듈에 종속적인 단순한 정책만을 제시하였다. 본 논문에서는 이러한 문제를 개선하기 위한 방안의 하나로, 신경망 기반 주식 거래 시스템의 구축을 위한 통합 개발 도구인 NNShell에서 채택하고 있는 ‘통합 다중 시뮬레이션’ 기법을 제안한다. 통합 다중 시뮬레이션 기법에서는 신경망의 출력 값과 거래 정책 인자들 간의 모든 조합에 대하여 다중의 거래 시뮬레이션을 수행하고, 다양한 성능 평가 척도에 의거하여 각 정책에 대한 거래 성능을 평가하며, 이에 기반하여 주어진 예측기의 특성에 맞는 고유의 최적 거래 정책을 수립한다. 제안된 기법의 효용성을 검증하기 위해, 한국 거래소 시장 및 코스닥 시장에서 수집한 데이터를 사용하여 수행한 거래 성능 비교 실험 결과를 제시한다.

키워드 : 주식 거래, 지능형 시스템, 다중 시뮬레이션, 거래 정책, 성능 최적화

Integrated Multiple Simulation for Optimizing Performance of Stock Trading Systems based on Neural Networks

Jae Won Lee[†] · Jangmin O^{††}

ABSTRACT

There are many researches about the intelligent stock trading systems with the help of the advance of the artificial intelligence such as machine learning techniques. Though the establishment of the reasonable trading policy plays an important role in the performance of the trading system, most researches focused on the improvement of the predictability. Also some previous works, which treated the trading policy, treated the simplified versions dependent on the predictors in less systematic ways. In this paper, we propose the ‘integrated multiple simulation’ as a method of optimizing trading performance of stock trading systems. The proposed method is adopted in the NNShell, a development environment for neural network based stock trading systems. Under the proposed ‘integrated multiple simulation’, we simulate the multiple tradings for all combinations of the neural network’s outputs and the trading policy parameters, evaluate the trading performance according to the various metrics, and establish the optimal policy for a given prediction module based on the resulting performance. In the experiment, we present the trading policy comparison results using the stock value data from the KOSPI and KOSDAQ.

Key Words : Stock Trading, Intelligent System, Multiple Simulation, Trading Policy, Performance Optimization

1. 서론

주가 예측 문제는 경제학, 물리학, 수학, 통계학, 전산학 등 여러 학문 분야의 관심이 되어왔다. 경영/경제학에서는

기본 분석(fundamental analysis) 및 기술 분석(technical analysis)에 대한 연구가 수행되었고[1][2], 물리/수학에서는 카오스 이론 등을 적용하였으며[3], 통계학 분야에서는 여러 가지 시계열(time series) 접근법을 사용하였다[4]. 이와 같이 오랜 기간 동안 많은 연구가 수행되었음에도 불구하고, 아직까지 주가에 대한 예측이 가능한지 그리고 실제 시장에서 유용성을 보장할 수 있는지에 대한 논란은 계속되고 있다. 그러나 최근 전산학의 인공지능(artificial intelligence)

※ 이 논문은 2005년도 성신여자대학교 학술연구조성비 지원에 의하여 연구되었음.

† 정회원 : 성신여자대학교 컴퓨터정보학부 교수

†† 정회원 : NHN 데이터마이닝랩 근무

논문접수 : 2006년 11월 13일, 심사완료 : 2007년 2월 23일

분야에서는 신경망(neural network)을 비롯한 다양한 기계 학습(machine learning) 기법들이 발전하면서, 매우 높은 수준의 통계적 신뢰도를 보이는 예측 결과들이 보고된 바 있으며[5-7], 이러한 예측 결과를 기반으로 지능형 주식 거래 시스템을 구축하기 위한 연구도 다수 수행되었다[8-10]. 그러나 주식 거래 시스템에 관한 기존의 연구에서는, 주로 주가의 예측을 위한 예측 방법론과 이에 기반한 거래 대상 종목의 선정(stock selection)에 주된 초점을 두었고, 미시적인 거래 정책(trading policy)의 선정에 관하여는 심도 있는 연구가 이루어지지 않았다. 강화 학습(reinforcement learning)에 기반한 연구들[10,11]에서 예측과 정책을 통합적으로 학습할 수 있는 학습 체계를 제안한 바 있으나, 역시 거래 정책의 선정을 독립적으로 다루지는 않았다.

본 논문에서는 거래 시스템의 최종 성능을 개선하기 위한 방안의 하나로 '통합 다중 시뮬레이션(integrated multiple simulation) 기법'(이하 'IMS 기법'이라 함)을 제안한다. IMS 기법은 신경망의 예측 결과와 다양한 미시 거래 정책 인자들을 통합적으로 고려하여 다중의 시뮬레이션을 수행하는 기법이다. 단기 예측에 적합한 기술 분석을 사용하는 신경망 기반 거래 시스템의 경우, 동일한 예측기(predictor)에 대하여 어떠한 거래 정책을 적용하는지에 따라 최종적인 거래 성능이 크게 달라질 수 있다. 또한 학습의 대상이나 학습 기간 등 예측기 고유의 특성에 따라 각 예측기에 적합한 최적 정책이 달라질 수 있다. 이러한 관점에서, IMS 기법은 많은 투자자들이 현실 거래에서 적용하고 있는 다양한 정책 관련 인자(parameter)들을 시스템에 도입하고, 개별 예측기의 특성에 따라 이 인자들의 값을 최적화함으로써, 최종적으로는 거래 시스템의 성능을 최적화하는 것을 그 목적으로 하고 있다.

IMS 기법은 신경망 기반 주식 거래 시스템의 구축을 위한 통합 개발 도구인 'NNShell'의 거래 정책 선정 모듈의 기능을 개선하기 위해 고안되었다. 다중 시뮬레이션의 결과로 생성된 후보 거래 정책 집합에 포함되는 모든 거래 정책들은 NNShell의 정책 정렬 기능에 의해 각 세부 성능 평가 척도의 관점에서 우수한 순서대로 정렬된다. 정렬 결과를 이용하여, 사용자는 자신의 투자 성향 및 선호하는 성능 평가 척도의 관점에서 우수하다고 판단되는 후보 거래 정책들을 비교 분석하고, 이에 근거하여 최종적인 최적 거래 정책을 선정할 수 있다.

논문의 구성은 다음과 같다. 2장에서는 IMS 기법이 가장 하고 있는 주가 변동 분석 방법론 및 거래 시스템의 개괄적 구성에 관하여 서술한다. 3장에서는 통합 다중 시뮬레이션의 개념을 소개하고, 시뮬레이션에 포함되는 거래 정책 인자들을 제시한다. 4장에서는 통합 다중 시뮬레이션의 결과를 활용하기 위한 평가 척도별 정렬 및 사용자 선호도의 반영을 포함한 최종 거래 정책 선정 과정을 설명한다. 5장에서는 두 가지 서로 다른 패턴에 대하여 학습한 예측기들에 대해 수행된 몇 가지 실험의 결과를 통해 IMS 기법의 효용성을 검증하고, 6장에서는 결론 및 추후 연구 과제에 관하여 논한다.

2. 배경

주가의 변동을 분석 혹은 예측하기 위해서는 일반적으로 기본 분석과 기술 분석의 두 가지 분석 방법론이 사용된다. 기본 분석은 특정 기업의 자산, 수익, 성장성 등 기업 내적 요인과 국내외 정치, 경제 상황 등 기업 외적 요인을 종합하여 주식의 이론적, 본질적 가치를 분석하고, 분석된 가치와 현재 주가를 비교하여 거래에 활용한다. 반면에 기술 분석은 날짜별 주가 변동 데이터 혹은 이로부터 계산된 값이나 지표(indicator)들만을 사용하여 주가를 분석하고 예측한다. 일반적으로 기본 분석은 중장기 예측에 적합하고, 기술 분석은 단기 예측에 적합한 것으로 알려져 있다.

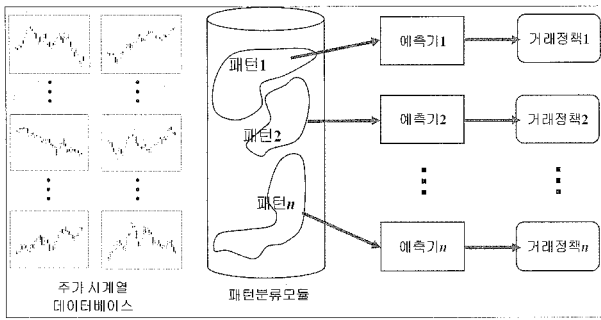
기본 분석은 주식의 가치를 평가하는 체계적인 틀을 제공할 수 있고 장기적 변동을 예측할 수 있다는 장점을 가지고 있으나, 부정확한 자료 및 통계적 오차로 인한 분석/예측의 오류 가능성이 있고 성장성 등 추상적 개념을 수치화하기 어렵다는 문제점이 있다. 이로 인해 기계 학습 기법을 적용하는 대부분의 주식 거래 시스템은 기술 분석에 기반하고 있다. 기술 분석에서는 투자자 집단의 군중 심리를 주가 변동의 주된 요인으로 간주하는데, 군중의 심리는 자연의 일부이므로 자연에서 발견할 수 있는 여러 패턴들과 유사한 패턴으로 주가의 추세가 형성될 것이므로, 주가 변동 차트(chart)의 패턴 및 파동 정보의 분석만으로도 주가의 변동을 예측할 수 있다는 입장을 취한다.

신경망 기반 주식 거래 시스템의 통합 개발 환경인 NNShell은 연구[9,11] 등을 위해 개발되었으며, 방대한 양의 주가 시계열 데이터에 대한 전처리(preprocessing), 시계열 패턴별 학습 데이터의 생성 및 학습 인자의 조절, 그리고 성능 평가를 위한 시뮬레이션 기능 등을 포함하고 있다. NNShell의 주가 데이터베이스에는 한국 주식 시장을 구성하는 거래소(KOSPI) 시장과 코스닥(KOSDAQ) 시장에 상장된 약 2,000 종목에 대한 1990년 1월 이후 현재까지의 데이터가 수집되어 있다. 각 원시 데이터(raw data, RD)는 7개의 필드로 구성되며 그 내용은 아래와 같다.

$$RD = (name, date, p_o, p_h, p_l, p_c, v)$$

여기서 *name*은 기업명, *date*는 날짜, p_o 는 해당 날짜의 거래 시작 시점의 가격인 시가(open), p_h 는 거래가 진행되는 동안 나타난 가장 높은 가격인 고가(high), p_l 는 가장 낮은 가격인 저가(low), p_c 는 거래가 종료된 시점에서의 가격인 종가(close), v 는 거래량이다. 이 원시 데이터로부터 가격 이동평균(moving average)을 비롯하여 기술 분석에서 사용되는 다양한 지표들을 계산하여 시스템의 학습과 검증에 사용한다.

본 논문에서 제안하는 IMS 기법은 NNShell을 사용하여 개발된 거래 시스템 중 연구[8]에서 사용한 거래 시스템의 기본 구조를 가정하고 있다(그림 1 참조). 기존의 전통적 거래 시스템들과는 달리 시계열 패턴별로 별도의 학습과 성능



(그림 1) 거래 시스템 구조

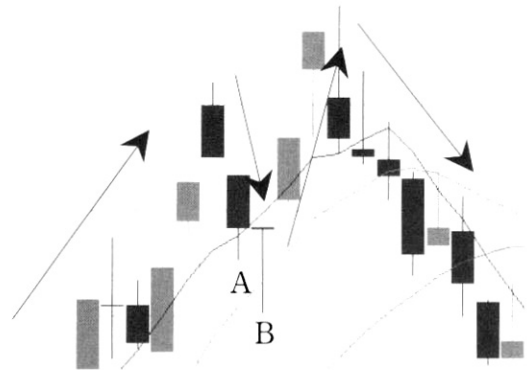
검증을 거친 독립된 다중 예측기(predictor)를 사용하며, 각 예측기의 특성에 따라 고유의 거래 정책을 채택할 수 있다.

3. 통합 다중 시뮬레이션

신경망 기반 주식 거래 시스템에서 신경망의 출력을 이용하는 가장 일반적인 방법은 특정 임계치 θ 이상의 신경망 출력(이하 ‘예측치’라 함)을 매수 신호로 간주하는 것이다 [12]. 즉, 예측과 거래의 대상이 되는 각 후보 종목에 대해 신경망 출력 o 를 구하고, 이 값이 예측 임계치 이상인 경우 해당 종목을 매수하게 된다. 신경망이 주가 변동에 대해 유의미한 예측력을 가지고 있다는 전제 하에, 적절한 예측 임계치의 선정이 거래 시스템의 최종 성능에 매우 큰 영향을 미치게 되는 것은 자명하다. 그러나 대부분의 기존 연구에서는 매우 간단한 시뮬레이션이나 통계적 검토를 통해 예측 임계치를 결정하여 사용하고 있다. 이는 기존의 연구들이 예측 결과의 활용보다는 예측력(predictability) 자체에 주된 초점을 두었기 때문인 것으로 판단된다.

그러나 실용적인 주식 거래 시스템의 구축이라는 관점에서, 최적 예측 임계치의 선정은 포함한 적절한 ‘거래 정책(trading policy)’의 선정이 높은 예측력의 확보 못지않게 중요한 의미를 가진다. 실제로 현실 거래에서 많은 일반 투자자들은 특정 종목의 주가가 오를 것이라는 기대나 예측에 근거하여 주식을 매수하지만, 각자 나름의 거래 정책을 가지고 거래에 임한다. 예로써, 어떤 투자자는 “주식을 매수하여 10퍼센트의 손실이 발생하면 추가적인 손실을 예방하기 위해 즉시 매도한다”와 같은 간단한 손절매(stop-loss) 거래 정책을 사용할 수 있다. 또 다른 어떤 투자자는, 상승세가 매우 강한 주식이라고 판단되면 전일 증가 대비 6% 이내의 가격이라면 비싼 가격에라도 추격 매수한다”와 같은, 매수 가격대에 대한 정책을 가지고 거래에 임할 수 있다. 이러한 정책들은 투자자마다의 개인적인 경험에 기반하는 경우가 많다. (그림 2)는 거래 정책의 중요성을 보여주는 극단적인 예로, 본 연구의 실험 과정에서 실제로 매수 신호가 발생하였던 종목의 차트이다.

A는 매수 신호가 발생한 날의 주가 변동 범위이고, B는 매수 신호에 의해 거래를 수행하게 되는 날의 주가 변동 범위이다. 이 경우 얼마나 낮은 가격에 주식을 매수하였는지,



(그림 2) 등락폭이 큰 주가 추세의 변동 사례

그리고 몇 퍼센트의 손절매율을 정하고 있었는지에 따라, B의 주가 범위 내에서 손실을 입은 채로 매도하였을 수도 있고, 정 반대로 이후 이틀간의 상승 기간 중 정해 놓은 목표 수익률을 달성한 채로 매도하였을 수도 있다.

본 논문에서 제안하는 IMS 기법은 앞서의 예에서 살펴본 바와 같이 실제 거래 상황에서 중요한 요소로 작용하는 다양한 거래 정책 인자(parameter)들을 통합적으로 고려하여 거래 정책을 수립함으로써 거래 시스템의 최종 성능을 향상시키기 위한 기법이다. 각 거래 정책 인자들의 모든 조합에 대해 다중의 가상 거래 시뮬레이션을 수행하며, 그 결과를 성능 평가 척도들(metrics)에 의거하여 분석하고, 시스템 사용자의 선호도(preference)에 따라 최종 거래 정책을 선정하게 된다. 파라미터 w 로 규정되는 예측 모형 f 가 있다고 하자. 또, 특정 날짜 t 에, f 에 대한 주식의 입력 표현을 x_t 로 하자. IMS 기법에서 다루는 거래 정책 인자는 다음과 같다:

3.1 예측임계치(prediction threshold): θ

식 (1)에서 제시한 바와 같이, 각 종목의 매수 신호 생성에 사용되는 임계치이다. 예측 임계치가 낮을 경우 거래 회수가 많아지는 반면 거래 당 수익률이 감소하는 경향이 있다. 예측값이 다음 조건을 만족하면 매수 신호가 발생한다.

$$f(x_t; w) \geq \theta \tag{1}$$

3.2 매수할인율(buy discount rate): α

매수 신호가 발생한 날의 증가 대비 매수 주문가(ask price)의 비율이다. 예를 들어, 매수할인율이 +3%인 거래 정책의 경우, 전일 증가가 10,000원인 종목에 대해서는 3%가 높은 10,300에 매수 주문을 내는 것으로 가정한다. 높은 매수할인율의 경우, 매수가 성립되지 않는 종목의 비율이 낮아져 거래 회수가 증가하지만, 손절매의 위험이 상대적으로 높아진다. 반대로 낮은 매수할인율의 경우, 손절매의 위험이 낮아지는 반면, 강한 상승세가 진행 중인 종목이 매수되지 않아 기회 수익을 잃게 될 위험은 상대적으로 높아진다.

매수신호가 발생하면, 매수할인율 α 를 이용하여, $t+1$ 날의 주문가는 $(1+0.01\alpha)p_o(t)$ 로 매수를 시도한다. 최종 매수

가격은 다음과 같이 결정된다. 만일 주문가가 저가보다 높게 되면 즉 $(1+0.01\alpha)p_c(t) \geq p_o(t+1)$ 라면 거래가 성립되고 매수가는 [식 2]와 같다.

$$p_{bid} = \min(p_o(t+1), (1+0.01\alpha)p_c(t)) \quad (2)$$

그렇지 않다면, 매수는 무효화된다.

3.3 손절률(stop-loss rate): β

매수 가격 대비 최대 허용 손실률이다. 예를 들어, 손절률이 10%인 거래 정책의 경우, 10,000원에 매수한 종목의 가격이 매수 당일로부터 보유기간 이내에 9,000원의 가격이 주가 범위에 나타나면 이 가격에 매도하는 것으로 가정한다.

3.4 최대보유일(maximum holding period): γ

매수 당일을 0일로 하여 보유 종목의 주식을 보유할 수 있는 최대 기간이다. 예를 들어, 최대보유일이 5일 거래 정책의 경우 매수 당일 이후 5일 후까지의 주가 범위에 이익실현율에 해당하는 가격과 손절률에 해당하는 가격이 모두 나타나지 않으면 최종일의 종가에 매도하는 것으로 가정한다.

3.5 이익실현율(target profit rate): δ

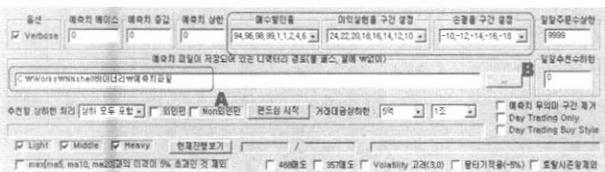
매수 가격(buy price) 대비 목표 수익률이다. 예를 들어, 이익실현율이 25%인 거래 정책의 경우, 10,000원에 매수한 종목의 가격이 매수 당일로부터 보유 기간 이내에 12,500원의 가격이 주가 범위에 나타나면 이 가격에 매도하는 것으로 가정한다.

매도 가격은 다음과 같이 계산된다.

$$p_{ask} = \begin{cases} (1-0.01\beta)p_{bid} & \text{if } \exists t' \text{ s.t. } p_l(t') \leq (1-0.01\beta)p_{bid} \\ (1+0.01\delta)p_{bid} & \text{if } \exists t' \text{ s.t. } p_h(t') \geq (1+0.01\delta)p_{bid} \\ p_c(t+\gamma) & \text{otherwise} \end{cases} \quad (3)$$

이 때, t' 는 $t+1 \leq t' \leq t+\gamma$ 의 범위를 갖는다. [식 3]의 의미는, 먼저 t' 의 저가가 손절가 보다 낮게 되면 손절가에 매도가 이뤄지며, 그렇지 않고, 고가가 이익실현가를 상회할 경우 이익실현가에 매도가 이뤄진다. 만일 최대 보유일에 도달하게 되면, 그 날의 종가에 매도가 이뤄진다.

(그림 3)은 IMS 기법을 구현한 NNShell의 사용자 인터페이스의 거래 정책 인자 선정 부분이다. 특정 예측기에 의해 예측된 전체 대상 종목의 날짜별 예측치 파일이 포함되어 있는 디렉토리를 지정하고(A), 각 거래 정책 인자들의



(그림 3) 거래 정책 인자의 설정을 위한 NNShell 인터페이스

가변 범위를 지정한 후(B), 시뮬레이션을 시작한다. 예측치 파일의 예측치 정보를 입력으로 하여 전체 종목의 전체 날짜에 대한 가격 변동 정보를 보관하고 있는 데이터베이스를 접근한 후, 지정된 범위 내의 모든 가변 인자 조합에 대해 다중의 가상 시뮬레이션이 수행된다.

4. 거래 정책의 선정

주어진 예측기에 대하여, 각 거래 정책별 시뮬레이션이 종료되면, 그 결과를 분석하여 최종 거래 정책을 선정하게 되는데 그 과정은 (그림 3)과 같다.

매우 많은 조합의 거래 정책에 대한 결과를 모두 살펴보는 것이 현실적으로 불가능하기 때문에 몇 가지 거래 성능 평가 척도별로 시뮬레이션 결과를 정렬하고, 정렬된 결과의 상위 정책들에 대하여, 사용자가 성능을 요약하는 평가 척도의 수치들과 세부 거래 내역을 검토한 후 사용자의 선호도에 맞는 최종 거래 정책을 선정한다.

거래 정책별 시뮬레이션 결과의 정렬에 사용되는 주요 성능 평가 척도는 다음과 같다.

매수가 발생한 거래 집합을 $T = \{(name_n, i_n, j_n) | n = 1, \dots, N\}$ 과 같이 표현하자. 이 때, N 은 총 거래 수, $(name_n, i_n, j_n)$ 은 각각 종목 이름, 거래 시작일, 거래 종료일을 의미한다.

4.1 거래당 수익률(profit rate per trade, PPT)

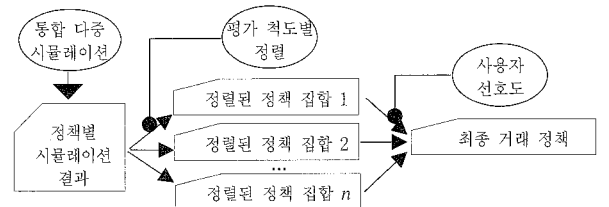
n 번째 거래의 수익률은 다음과 같다.

$$prof_n = \frac{p_{ask}(j_n) - p_{bid}(i_n)}{P_{bid,n}} \quad (4)$$

이 때, $p_{bid}(i_n), p_{ask}(j_n)$ 은 각각 [식 2, 3]를 이용해 구한다. 거래당 수익률 PPT는

$$PPT = \frac{100}{N} \sum_{n=1}^{N} prof_n \quad (5)$$

PPT 는 전반적인 거래의 성능을 직관적으로 표현할 수 있는 가장 일반적인 거래 성능 평가 척도이다. 그러나 거래회수가 작은 거래 정책의 경우에는 통계적 유의성(significance)의 관점에서, PPT 만으로 거래 정책의 성능을 평가하는 것이 무의미할 수도 있다. 따라서 PPT 를 평가할 때는 반드시 거래회수를 함께 고려하여야 한다.



(그림 4) 거래 정책의 선정 과정

4.2 적중률(hit rate, HR)

전체 거래 중 양의 수익률을 달성한 거래의 비율로, 다음과 같이 계산된다.

$$HR = \frac{100}{N} \sum_{n=1}^N I(\text{prof}_n > 0) \quad (6)$$

이때, $I(\cdot)$ 은 참이면 1을 돌려주는 지시자 함수(indicator function)이다. 양의 수익률을 달성한 거래는 이익실현율에 해당하는 가격이 나타나서 이익 실현을 한 거래와 최대 보유일이 경과하여 매도하였지만 양의 수익을 달성한 거래로 다시 구분 할 수 있으며, 적중률로 정책을 평가할 때는 이익 실현에 해당하는 거래의 비율을 동시에 참고하는 것이 타당하다.

4.3 수익금(total profit, TP)

각 거래당 자산 a 를 투자하였다고 가정했을 때, 전체 거래에 의해 달성되는 최종 수익금을 의미한다. 해당 시물레이션의 총 거래회수를 N 라 할때, 수익금 TP 는 [식 7]로 계산할 수 있다.

$$TP = a \times (0.01PPT) \times N \quad (7)$$

PPT 가 높은 거래 정책의 경우, 상대적으로 HR 이 낮아지며, 또한 작은 거래회수로 인해 통계적 신뢰도가 낮은 경우가 많다. TP 는 PPT 와 거래회수를 동시에 고려할 수 있는 평가 척도이지만, HR 은 반영되지 않는다.

(그림 5)는 통합 다중 시물레이션의 결과에 대한 정렬 사례를 보이고 있다. 예측 임계치를 0.25에서 출발하여 0.03씩 증가시키며 시물레이션을 수행한 후, HR 을 기준으로 정

렬한 상태이다. (A)에는 각 거래 정책의 인자를 요약하는 벡터와 성능 척도별 수치, 그리고 거래회수 및 평균 보유일 수 등의 보조 인자들이 출력된다. 특정 거래 정책에 대한 상세 거래 내역을 분석하고 싶은 경우, 정책에 해당하는 라인을 더블 클릭하여, (B)에 나타난 바와 같은 세부 거래 내역을 출력할 수 있다. 세부 거래 내역에는 날짜 순으로 모은 거래 종목의 이름, 거래 일자, 매수/매도 가격 등 거래에 관련된 상세 정보가 포함되고, 마지막 부분에는 월별 누적 수익 분포가 출력되어 월 단위의 기간별 성능을 관찰해 볼 수 있다.

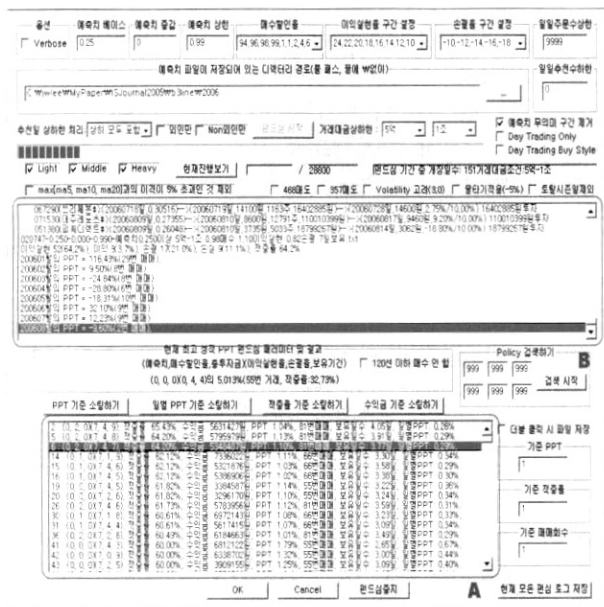
성능 평가 척도별 정렬 결과를 이용하여 최종 거래 정책의 선정에 사용자의 선호도를 반영하는 것에 관한 내용은 다음 장의 단계별 실험에 대한 설명 부분에 부연하여 서술하기로 한다.

5. 실험

본 연구에서 제안하는 IMS 기법의 효용성을 검증하기 위해, 서로 다른 두 가지 파동 패턴에 대한 예측을 수행하는 예측기들에 대해, 거래 정책을 선정하는 실험과 최종 성능을 검증하는 실험을 각기 수행하였으며, 실험의 절차는 다음과 같다. 먼저 최적 거래 정책을 선정하기 위해 '정책 선정 기간(policy selection period)'에 대해 통합 다중 시물레이션을 수행하였다. 이 결과로 생성된 후보 거래 정책들 중 각 평가 척도의 관점에서 최상위에 해당하는 정책들에 대한 세부 성능을 비교하여 최적 거래 정책을 선정하였으며, 이 정책의 성능에 대한 최종 검증을 위해 '정책 검증 기간(policy validation period)'에 대하여 시물레이션을 재차 수행하였다. <표 1>은 실험에 사용된 각 데이터 집합의 해당 기간을 요약하고 있다.

두 가지 예측기로는 일정 기간 주가 상승이 진행된 '상승 추세(bull trend)형'1) 패턴에 대한 예측을 수행하는 예측기 P_{bull} 과, 이와 반대의 추세가 진행된 '하락 추세(bear trend)형' 패턴에 대한 예측을 수행하는 예측기 P_{bear} 를 사용하였다. 두 예측기 모두 연구 [9]에서 제시한 기술 분석 지표들을 포함하는 총 127개의 입력 자질을 사용하여 학습하였다.

적절한 신경망의 구조를 선정하기 위하여 다양한 신경망 구조와 서로 다른 학습 기간들 간의 각 조합에 대해 학습된



(그림 5) 거래 정책들의 정렬 사례

<표 1> 실험 데이터 집합의 기간

데이터 집합	기간
신경망 학습용 데이터	1998년 1월~2000년 8월
신경망 학습 검증용 데이터	2000년 9월~2001년 12월
거래 정책 선정용 데이터	2002년 1월~2004년 6월
거래 정책 검증용 데이터	2004년 5월~2006년 8월

1) 상승 추세란 어디까지나 최근에 주가 상승이 이어졌다는 의미일 뿐, 이후 주가의 상승 및 하락 여부는 무관하며, 이는 하락 추세에 대해서도 마찬가지이다.

<표 2> 신경망의 구조와 학습 데이터에 따른 예측 성능 비교

1개 은닉층				2개 은닉층			
노드수	정확도1	정확도2	정확도3	노드수	정확도1	정확도2	정확도3
60	0.567	0.585	0.581	10x20	0.608	0.610	0.614
80	0.588	0.590	0.570	20x20	0.615	0.592	0.611
100	0.609	0.593	0.612	40x20	0.623	0.616	0.631
120	0.592	0.603	0.598	60x20	0.618	0.621	0.620
140	0.585	0.577	0.587	80x20	0.612	0.608	0.602

예측 성능을 측정하는 실험을 수행하였으며, 그 결과는 <표 2>와 같다.

여기서 정확도1은 <표 1>의 ‘신경망 학습용 데이터’를, 정확도2는 ‘거래 정책 선정용 데이터’를, 그리고 정확도3은 ‘거래정책 검증용 데이터’를 각기 학습 데이터로 사용하여 얻어진 학습 결과에 따른 예측의 정확도이다. 정확도 측정을 위한 검증 데이터로는 표 1의 ‘신경망 학습 검증용 데이터’를 공통으로 사용하였다. 각 조합에 대해, 예측치 상위 10%에 해당하는 동일 개수의 주식을 예측한 것으로 가정하였고, 예측한 날짜로부터 5일 이후의 가격이 예측 날짜의 가격보다 높은 경우에는 예측이 맞은 것으로, 그렇지 않은 경우에는 틀린 것으로 간주하였다. 전반적으로 1개 은닉층(hidden layer)을 사용한 경우보다 2개 은닉층을 사용한 경우가 성능이 우수하였으며, 그 중에서도 각 층에 40개와 20개의 노드를 둔 경우가 상대적으로 우수한 성능을 보인 것으로 판단되어 이를 최종 실험의 신경망 구조로 채택하였다. 1개 은닉층과 2개 은닉층 모두에서, 통계적으로 유의미한 수준의 학습 기간별 학습 성능의 편차는 나타나지 않았다.

<표 3>은 P_{bull} 의 정책 선정 기간에 대한 통합 다중 시물레이션의 결과를 요약한 것으로, 4장에서 제시한 세 가지 성능 평가 척도인 PPT , HR , TP 의 각 관점에서 가장 우수한 성능을 보인 거래 정책들($P_1 \sim P_3$)을 보이고 있다. BP는 기본 정책(baseline policy)으로, 매수할인율을 0%, 이익실현율과 손절률을 16%로 고정하고, 예측임계치와 최대보유일은 비교 대상이 되는 정책과 동일한 값으로 설정하는 정책이다.

<표 4>는 <표 3>의 거래 정책들의 세부 성능 평가 내역이다. HR 에 대하여는 목표 이익실현율을 달성한 ‘이익실현’과 이익실현율을 달성하지는 못했지만 양의 수익을 달성한 ‘이익’ 등으로 세부 구분하여 거래회수와 비율을 제시하였다. TP 는 거래 당 100,000원을 투자한 것으로 가정하여 산출한 수익금이다.

$P_1 \sim P_3$ 의 세 거래 정책 모두 기본 정책의 매수할인율인 0%보다 낮은 -2%의 매수할인율을 택하고 있다. 이것은 예측기 P_{bull} 의 경우, 전일 증가 대비 2% 낮은 가격에 주식을

<표 3> P_{bull} 의 정책 선정 기간에 대한 평가 척도별 우수 거래 정책

정책명	기준척도	예측 임계치	매수할인율	이익실현율	손절률	최대 보유일
P1	PPT	0.30	-2%	22%	14%	10
P2	HR	0.30	-2%	10%	16%	12
P3	TP	0.27	-2%	22%	14%	13
BP	.	.	0%	16%	16%	.

<표 4> P_{bull} 에 대한 우수 거래 정책들의 세부 성능

정책명	거래 회수	PPT	이익 실현	이익	손절	손실	TP
P1	76	4.22%	26(34.2%)	13(17.1%)	22(28.9%)	15(19.7%)	320,720
P2	76	3.68%	58(76.3%)	1(1.3%)	14(18.4%)	3(3.9%)	279,680
P3	113	3.14%	34(30.1%)	22(19.5%)	31(27.4%)	26(23.0%)	354,820
BP	89	2.89%	45(50.6%)	8(9.0%)	21(23.6%)	15(16.9%)	257,210

<표 5> P_{bull} 의 정책 검증 기간에 대한 성능

정책명	거래 회수	PPT	이익실현	이익	손절	손실	TP
P2	68	3.41%	49(72.1%)	4(5.9%)	14(20.6%)	1(1.5%)	231,880
BP	81	2.78%	38(47.0%)	9(11.1%)	18(22.2%)	16(19.8%)	225,180

매수하는 것이 각 성능 척도의 관점에서 최적이라는 의미이다. PPT 가 가장 높은 P1은 P2에 비해 이익실현율(22%)은 높고, 손절률(14%)은 낮다. 결과적으로 이익실현율은 P2에 비해 매우 낮지만, 최종 수익금인 TP 는 오히려 더 높은 것을 확인할 수 있다. 이것은 34.2%에 해당하는 이익실현 달성 거래들에서 큰 수익을 얻었기 때문인 것으로 판단된다. P2는 수익금이 P1에 비해 작지만, 76.3%라는 매우 높은 이익실현율을 보이고 있다. P3은 P1, P2에 비해 상대적으로 낮은 값인 0.27을 예측임계치로 사용하고 있다. 그 결과, 거래회수가 113회로 P1, P2에 비해 많은 대신 적중률 $HR^{(2)}$ 이 낮고 PPT 도 세 정책 중 가장 낮지만, 많은 거래회수로 인해 수익금은 가장 크다.

높은 PPT 를 선호한다면 P1을, 높은 이익실현 비율을 선호한다면 P2를 최적 거래 정책으로 선정할 수 있다. 또한 손절과 손실의 비율이 높아지는 위험을 감수하더라도 최종 수익금의 최대화를 선호하는 경우라면 P3을 최적 거래 정책으로 택할 수도 있다. 여기서는 각 정책들 간 수익금 차이에 비해, P2의 이익실현 비율이 두드러지게 높은 것을 근거로 하여 P2를 최적 거래 정책으로 선정하였다고 가정하기로 한다. 이러한 가정 하에 P2의 성능을 기본 정책 BP와 비교해 보면, 매수할인율이 0%인 BP에 비해 거래회수가 작음에도 불구하고, 이익실현 비율을 포함한 HR 과 PPT 의 측면에서 모두 BP보다 좋은 성능을 보임을 확인할 수 있다.

<표 4>는 최적 거래 정책의 성능을 최종적으로 검증하기 위해 정책 검증 기간에 대해 P2와 BP를 사용하여 수행한 시물레이션의 결과이다. 두 정책 모두 PPT 가 약간 낮아졌지만 전반적인 성능의 변화는 없는 것으로 판단되며, 이 결과에서도 P2가 BP에 비해 우수한 성능을 보임을 알 수 있다.

<표 6>은 하락추세형 패턴에 대한 예측기인 P_{bear} 의 정책 선정 기간에 대한 통합 다중 시물레이션 결과이다. 이 예측기의 경우 PPT 와 TP 의 측면에서 가장 우수한 거래 정책이 동일한 정책(P_4)이었다. P_{bull} 에 대한 실험 결과에서와는 반대로 두 정책 모두 +6%과 +4%의 매우 높은 매수할인율을

2) 이 표에서는 ‘이익실현’의 비율과 ‘이익’의 비율을 더한 값이다.

〈표 6〉 Pbear의 정책 선정 기간에 대한 평가 척도별 우수 거래 정책

정책명	기준 척도	예측 임계치	매수 할인율	이익 실현율	손절률	최대 보유일
P4	PPT/TP	0.38	+6%	24%	16%	17
P5	HR	0.38	+4%	10%	18%	12

〈표 7〉 Pbear에 대한 우수 거래 정책들의 세부 성능

정책명	거래 회수	PPT	이익 실현	이익	손절	손실	TP
P4	166	12.70%	106(63.9%)	20(12.0%)	28(16.9%)	12(7.2%)	2,108,200
P5	166	6.36%	148(89.2%)	0(0.0%)	16(9.6%)	2(1.2%)	1,055,760
BP	126	9.48%	100(79.4%)	3(2.4%)	20(15.9%)	3(2.4%)	1,194,480

〈표 8〉 Pbear의 정책 검증 기간에 대한 성능

정책명	거래 회수	PPT	이익 실현	이익	손절	손실	TP
P4	132	11.32%	82(62.1%)	12(9.1%)	24(18.2%)	14(10.6%)	1,494,240
BP	103	9.16%	80(77.7%)	3(2.9%)	15(14.6%)	5(4.9%)	943,480

택하고 있다. 즉, 거래 후보 종목들 중 더 많은 비율의 종목이 수익이 발생하는 종목이기 때문에 다소 높은 가격에 매수하더라도 매수하지 않는 것에 비해 최종 수익의 측면에서 유리함을 의미하는 것으로, P_{bull} 에 비해 P_{bear} 의 예측력이 높은 것에 기인한 것으로 판단된다.

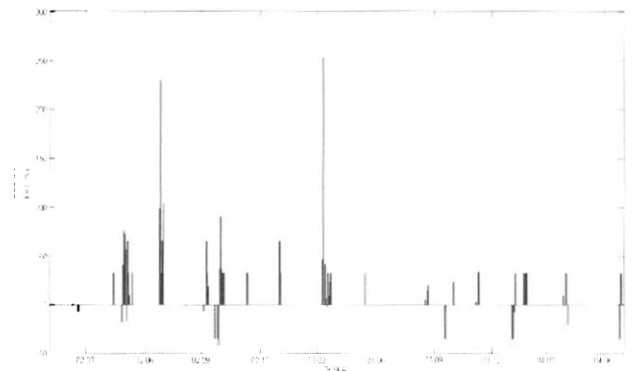
P_{bear} 에 대한 우수 거래 정책들의 세부 성능 평가 내역을 보이고 있는 <표 6>에 의하면, P_{bull} 의 경우에서와는 달리 PPT의 측면에서 가장 우수한 정책인 P_4 가 HR의 측면에서 가장 우수한 정책인 P_5 에 비해 두 배에 달하는 TP를 얻었다. 이것은 P_4 가 24%의 높은 이익실현율을 설정하고 있음에도 불구하고 75.9%의 높은 HR을 달성하고 있기 때문인 것으로 분석할 수 있으며, P_{bear} 에 대해서는 P_4 를 최적 거래 정책으로 사용하기로 한다. P_4 와 BP의 거래회수 차이는 매수 할인율의 차이로 인한 것이며, 결과적으로 TP의 차이에 반영된 것으로 해석할 수 있다. <표 8>은 P_4 와 BP의 정책 검증 기간에 대한 시뮬레이션 결과이다. <표 7>에서와 유사한 성능 평가 결과를 확인할 수 있다.

예측기 P_{bear} 에 대한 실험 결과는 기술 분석에 의한 주가 예측의 관점에서 매우 유의미한 결과로 볼 수 있다. 기술 분석에서 사용되는 다양한 파동 패턴 중 V자형 반등 패턴(그림 6) 참조), 즉 투자자들의 심리적 요인으로 인한 투매로 인해 단기간에 주가가 폭락한 패턴에 대해서는, 신경망에 의해 매우 높은 확률로 주가의 전환 시점을 포착해 낼 수 있음을 보이고 있기 때문이다.

다만, 이러한 실험 결과를 실제 거래에 활용함에 있어서는 유의해야 할 사항이 있다. P_{bear} 의 정책 선정 기간에 대하여, 각 날짜별로 거래의 수익률의 합을 보이고 있는(그림 7)의 경우, 이익실현에 해당하는 많은 거래들이 시장 전체의 폭락 시점³⁾에 밀집되어 분포하고 있음을 알 수 있다. 이는 거래 시스템의 성능 평가에 있어, 전체 거래 기간 중 세부



(그림 6) V자형 반등 패턴의 예



(그림 7) Pbear의 거래별 수익률과 손실률

구간 별 성능의 편차가 추가로 고려되어야 함을 시사하고 있다.

6. 결론

본 논문에서는 신경망 기반 주식 거래 시스템의 성능을 최적화하기 위한 방안의 하나로 IMS 기법을 제안하였다. IMS 기법은 기존의 연구에서 심도 있게 다루어지지 않았던 미시 거래 정책 인자들을 통합적으로 고려하여 다중의 시뮬레이션을 수행하는 기법으로, 기존의 단순 거래 정책에 비해 개선된 성능의 거래 정책들을 선정해 낼 수 있음을 실험을 통해 확인하였다. 또한, 이 기법에 의해, 예측 대상 및 예측력이 다른 개별 예측기의 고유한 특성에 따라 최적화된 예측기별 고유 거래 정책을 수립할 수 있음을 예시하였다. 실험 결과의 분석을 통해, 주가 예측의 주요 기법인 기술 분석에서 사용되는 여러 가지 파동 패턴 중, V자형 패턴이 전산학적 관점에서는 매우 유용한 패턴임을 확인할 수 있었다. 그러나 실제 거래에 제안된 기법을 적용함에 있어서는 세부 거래 구간별 성능의 편차 등이 고려되어야 하며, 위험 관리(risk management)도 추가로 고려되어야 한다. 이를 위해 거래 당 수익률에 대한 월별 분산 및 최대 손실(maxdraw) 구간에 관련된 척도 등을 기존의 평가 척도와 결합하여 다룰 수 있는 기법에 대한 추후 연구가 필요한 것으로 판단된다.

3) 2003년의 경우 3월부터 4월까지의 시기는 이라크전이 시작되는 시기로, 전쟁에 대한 투자자들의 불안 심리로 시장 전체가 폭락하였다가, 이라크전의 조기 종결 가능성이 제기되면서 반등이 시작되었다.

참고 문헌

- [1] S. Cottle, R. F. Murray, F. E. Block, and D. L. Dodd, *Graham and Dodd's Security Analysis*(5th edition), McGraw-Hill, 1988.
- [2] R. D. Edwards and J. Magee, *Technical Analysis of Stock Trends*, John Magee, Inc., 1974.
- [3] A. W. Lo, Long-Term Memory in Stock Market Prices, *Econometrica* 59(5).
- [4] S. M. Kendall and K. Ord, *Time Series*, Oxford, New York, 1997.
- [5] G. Armano, M. Marchesi, A. Murru, A Hybrid Genetic-neural Architecture for Stock Indexes Forecasting, *Information Sciences* 170(1), pp.3-33, 2005.
- [6] R. S. T. Lee, "iJADE Stock Advisor: An Intelligent Agent based Stock Prediction System using Hybrid RBF Recurrent Networks," *IEEE Transactions on Systems, Man, and Cybernetics, Part A:Systems and Humans* 34(3), pp. 421-428, 2004.
- [7] A. Fan, M. Palaniswami, "Stock Selection using Support Vector Machines," *Proceedings of the International Joint Conference on Neural Networks*, pp. 1793-1798, 2001.
- [8] J. O, J. Lee, J. W. Lee, and B. Zhang, "Adaptive Stock Trading with Dynamic Asset Allocation using Reinforcement Learning," *Information Sciences* 176, pp.2121-2147, 2006.
- [9] J. W. Lee, S. D. Kim, J. Lee, J. Chae, "An Intelligent Stock Trading System based on Reinforcement Learning," *IEICE Transactions on Information and Systems* E86-D(2), pp. 296-305, 2003.
- [10] J. Moody and M. Saffell, "Learning to Trade via Direct Reinforcement", *IEEE Transactions on Neural Networks*, 12(4), pp.875-889, 2001.
- [11] J. W. Lee, J. O, "A Multi-agent Q-learning framework for Optimizing Stock Trading Systems," *Proceedings of the International Conference on Database and Expert Systems Applications*, pp. 153-162, 2002.
- [12] J. S. Zirilli, *Financial Prediction using Neural Networks*, Thomson Computer Press, 1997.

이재원



e-mail : jwlee@sungshin.ac.kr

1990년 서울대학교 컴퓨터공학과 (학사)

1992년 서울대학교 컴퓨터공학과 (석사)

1998년 서울대학교 컴퓨터공학과 (박사)

1999~현재 성신여자대학교 컴퓨터정보

학부 교수

관심분야: 금융공학, 인공지능, 기계학습,

자연언어처리

오장민



e-mail: rupino11@naver.com

1997년 서울대학교 컴퓨터공학과 (학사)

1999년 서울대학교 컴퓨터공학과 (석사)

2006년 서울대학교 컴퓨터공학과 (박사)

2006~현재 NHN 데이터마이닝랩 근무

관심분야: 기계학습, 데이터마이닝,

금융공학