



### 저작자표시-비영리-동일조건변경허락 2.0 대한민국

이용자는 아래의 조건을 따르는 경우에 한하여 자유롭게

- 이 저작물을 복제, 배포, 전송, 전시, 공연 및 방송할 수 있습니다.
- 이차적 저작물을 작성할 수 있습니다.

다음과 같은 조건을 따라야 합니다:



**저작자표시.** 귀하는 원저작자를 표시하여야 합니다.



**비영리.** 귀하는 이 저작물을 영리 목적으로 이용할 수 없습니다.



**동일조건변경허락.** 귀하가 이 저작물을 개작, 변형 또는 가공했을 경우에는, 이 저작물과 동일한 이용허락조건하에서만 배포할 수 있습니다.

- 귀하는, 이 저작물의 재이용이나 배포의 경우, 이 저작물에 적용된 이용허락조건을 명확하게 나타내어야 합니다.
- 저작권자로부터 별도의 허가를 받으면 이러한 조건들은 적용되지 않습니다.

**저작권법에 따른 이용자의 권리는 위의 내용에 의하여 영향을 받지 않습니다.**

이것은 [이용허락규약\(Legal Code\)](#)을 이해하기 쉽게 요약한 것입니다.

[Disclaimer](#)

이 재 원 교수지도  
석사학위 청구논문

신경망 기반 주가 거래 시스템의  
성능 개선에 관한 연구

2008

성신여자대학교 일반대학원  
전산학과 인공지능시스템전공  
변 주 연

신경망 기반 주가 거래 시스템의  
성능 개선에 관한 연구

이 재 원 교수지도

이 논문을 석사학위 논문으로 제출함.

2007년 11월

성신여자대학교 일반대학원  
전산학과 인공지능시스템전공  
변 주 연

# 인 준 서

변 주 연 의 석사학위 논문을 인준함.

심사위원 (인)

심사위원 (인)

심사위원 (인)

성신여자대학교 일반대학원

# 논문 개요

주식 시장은 현대 자본주의 경제의 중심에 있다. 주가를 예측하려는 분야는 많은 분야에서 이루어져 왔으며 특히 컴퓨터 과학 분야에서의 연구는 성능 좋은 예측기계를 만들어 기술 분석에 집중해왔다. 그러나 아직까지 이러한 시스템들이 실제 주식시장에 적용되었을 때, 실험과 동일한 수준의 거래 성능을 유지할 수 있는지에 대해서는 연구가 부족하다. 또한 성능에 대한 평가 부족은 다시 기계가 제대로 학습 되었는지를 판단할 수 없다는 것이 가장 큰 문제이다. 이러한 시스템의 예측 성능을 보다 객관적이고 체계적으로 평가할 수 있어야 하며, 또한 이렇게 평가 된 부분을 토대로 시스템의 개선해야 할 것이다.

이에 본 논문에서는 기존의 단일 신경망 기반 주가 예측 거래 시스템들을 살펴보고, 본격적으로 기존 연구에서 개발, 활용된 신경망 기반 주가 예측 시스템의 개발도구인 'NNShell'을 소개하겠다. 그리고 이 예측 시스템의 성능을 살펴보겠다. 그리고 이에 기반하여 단일 신경망의 출력을 결합하여 성능을 개선하는 방법을 제시하고 이를 효과적으로 적용할 수 있도록 의사 결정 트리를 사용하여 예측의 성능을 개선하는 방안을 제시하고, 이를 증명하겠다.

# 목 차

## 논문개요

I. 서론 .....	1
II. 기반 지식 .....	4
2.1 개념과 구조 .....	4
2.2 발전 .....	6
2.3 활용 .....	10
2.3.1 기존 연구 .....	13
2.3.2 단일 신경망 기반 주식거래 시스템 .....	14
III. 다중신경망 .....	21
3.1 개념과 기술 .....	21
3.2 다중 신경망 기반 주식거래 시스템 .....	24
3.2.1 Counting 기법 .....	24
3.2.2 신경망 값의 합 .....	26
IV. 의사결정트리 .....	28
4.1 필요성 .....	29
4.2 분석 .....	30
4.2.1 분석과정 .....	30
4.2.2 성능평가 .....	33
VI. 결론 .....	38

## 표 목 차

[표 1] KOSPI 지수 주간 예측 - 신경망 모형 결과(%) .....	13
[표 2] 실험 데이터의 집합 .....	18
[표 3] 신경망의 구조와 학습데이터에 따른 예측 성능 .....	19
[표 4] 정책 선정 기간에 대한 평가 척도별 우수 거래 정책 .....	19
[표 5] 우수 거래 정책들의 성능평가 .....	20
[표 6] Counting 기법의 예 .....	25
[표 7] DT 결과의 1급의 수익률 .....	33
[표 8] 1급으로 선정된 종목의 정확도 .....	33
[표 9] DT 결과의 2급의 수익률 .....	34
[표 10] 2급으로 선정된 종목의 정확도 .....	35
[표 11] DT 결과의 3급의 수익률 .....	35
[표 12] 3급으로 선정된 종목의 정확도 .....	36
[표 13] DT 결과 각 급당 수익률/추천개수 .....	36
[표 14] 각 등급의 손익 거래량 .....	37

## 그림 목차

[그림 1] 신경망 층에 따른 결정 영역 .....	7
[그림 2] 신경망의 기본 구조 .....	8
[그림 3] 회귀 신경망 .....	9
[그림 4] 분류형 신경망의 기본 구조 .....	10
[그림 5] 예측형 신경망의 구조 .....	11
[그림 6] 거래 시스템의 구조 .....	15
[그림 7] 거래 정책의 선정 과정 .....	18
[그림 8] 각 신경망 결과에 대한 정규 분포표 .....	24
[그림 9] 예측치를 적용한 신경결합 기법의 결과 예제 .....	26
[그림 10] 다중 신경망의 모식도 .....	27
[그림 11] 의사 결정트리의 분석과정 .....	30
[그림 12] 입력 값 설정에 따른 DT .....	31
[그림 13] 의사결정트리의 결과 .....	32

## 식 목 차

[식 1] 델타규칙을 적용한 식 .....	4
[식 2] 신경망 안의 은닉층에서 가중치 조절 식 .....	8
[식 3] 원시 데이터 .....	14
[식 4] 매수 신호 발생 식 .....	16
[식 5] 거래 당 수익률의 수식 .....	17
[식 6] 최종 매수가격( $P_{bid}$ )과 매도 가격( $P_{ask}$ )의 수식 .....	17

# 제 1 장 서론

실세계의 정보환경과 기술이 급격히 발전하고 있음에 따라 모든 학문분야에서는 불확실성을 포함한 복잡한 현실적인 문제들을 가능한 빠르고 근접한 해를 얻고자 하는 노력들이 이루어지고 있다. 주가를 예측하는 문제는 여러 학문분야에서 관심이 되어 오고 있다. 일반적으로 경제학에서 언급하는 기본 분석(Fundamental Analysis)은 가치(Value)를 투자의 척도로 삼아 투자하는 방식이다. 이 분석의 중심은 수요와 공급이 바탕이 된 사회적이고 정치적인 힘을 분석함으로써 가치투자 방식을 지원한다. 분석가들은 경제 성장률, 이율, 인플레이션, 그리고 실업과 같은 경제요인과 산업 요인 및 기업요인 등을 광범위하게 검토하고 기업의 재무제표를 분석하여 기업의 수익이나 배당력을 계량화함으로써 주식의 장기적인 가치를 산출하고 이를 시장에서 형성되는 실제 주가와 비교하여 증권의 매수 또는 매도를 판단하는 지표로서 사용하는 방법으로 비교적 적절한 방법이나, 자신의 주관적인 잣대로 판단의 한계가 있으며, 또한 증권시장에 반영 여부도 알 수 없다는 단점이 있다. 수 분야에서는 카오스 이론을 적용하여 예측을 하며, 전산학 분야에서는 주가 변동의 시간흐름을 분석하기 위해 시계열(Time Series)접근법을 사용하기도 하였다.[1] 하지만 시장 환경이 변함에 따라 시장변수들은 많은 잡음(noise)을 갖게 되었으며 변동성을 예측하기 위하여 고려하여야 할 정성적, 정량적 변수들이 점차로 증가하고 있는 현실 속에서 이러한 시계열모형 자체는 변수의 분포에 대한 엄격한 가정을 요구하고 있기 때문에 시장 변수들을 모형 속에 직접 반영하기가 어렵다는 단점을 가지고 있다.[2] 이와 같은 계량적인 시계열 모형들의 단점을 보완하기 위해서 인공신경망(Artificial Neural Network), 유전자 알고리즘(Genetic Algorithm: GA)등의 인공지능 기법을 이용하여 시계열을 예측하는 방법에 많은 관심이 집중되고 있으며 최근 전산적인 접근 방법으로 신경망을 사용하였다.[3] 신경망은 인간의 정보처리 방식을 모방하여 적응 학습기능, 대규모 병렬처리, 그리고 함수 근사화 및 일반화의 장점을 갖고 있어 기존의 방법으로 해결하기 어려웠던 적응형 예측 및 제어, 음성·패턴 인식, 최적화 등 다양한 분야에서 매우 활발히 응용되고 있다. 특히 경제 예측분야에서도 연구가 활발히 진행되고 있는데, 주택가격지수에서 매우 높은 수준의

통계적 신뢰도를 보이는 결과들이 보고되었으며, 지능형 주식 거래 시스템을 구축하기 위한 연구도 실행되고 있다.[3] 주식을 예측하는 기존 연구에서는 각 주식에 대해 분리된 네트워크를 구축하는 방법과 네트워크를 공유하는 방법 모두 실험하여 성능을 평가하였으며 개인의 경제행위, 기업의 활동 등의 미시경제 변수 뿐 아니라 국내 총 생산, 물가, 실업률, 금리, 환율 등의 거시적인 경제 변수까지도 고려하였다. 그러나 연간 수익률로 시스템을 평가하는데 그쳤다.[4] 웹 게시판의 글을 이용하여 주가를 예측하는 알고리즘을 제안한 연구도 있다. 하지만 성능을 평가하는데 있어서 테스트 기간과, 한정된 종목으로 신뢰도를 주지 못하였다. 이러한 기존 연구에서 보여지 듯, 검증기간이 짧거나 장기적이지 못하고, 실세계 검증을 제대로 하지 않아 통계적인 시점에서 신뢰도가 부족하다. 또한 대량의 데이터로 검증한 경우라 할지라도 평균 수익률이나 거래 성공률 등 단순한 척도에 의존하였다. 실제 투자자들의 투자방식은 사람의 심리 등과 같이 단순한 성능 평가 척도로만 표현하기 어려운 문제가 있으므로 다양한 성능 평가 수단을 고려해야 한다.

본 논문에서는 코스피(KOSPI)와 코스닥(KOSDAQ)의 예측을 위하여 기존 ‘단일 신경망 기반 주식거래정책 시스템(Stock Trading System Based on Multiple Neural Networks)’의 성능을 개선한다. 이를 위해 다중 신경망 기반 주식거래 시스템 기반의 틀을 사용하여, 의사결정트리를 사용한 거래정책을 제안한다. 또한 성능 평가가 평가에서 그치는 것이 아니라 이 평가 수단에 의해 기존의 시스템의 최적화를 위한 수단으로 활용하는 것을 보인다.

본 논문 2장에서는 단일 신경망 기반 주가 거래 시스템이 등장 할 수 있었던, 단일 신경망에 대한 기본 지식과 그에 따른 활용, 그리고 대표적인 시스템인 ‘NNShell’에 대해 소개하고자 하며, 3장에서는 한 개의 신경망이 아닌 여러 개의 신경망을 사용하는 다중 신경망의 필요성과 기술에 대해 언급하고자 한다. 그리고 4장에서는 다중 신경망에 기반 하여 예측기의 성능을 개선하기 위해 단일 신경망의 출력을 의사결정 트리의 입력 값으로 사용하여 더 나은 종목 선정을 제안하고 그 성능을 보여준다. 그리고 마지막으로 5장에서는 본 논문의 결론을 제시한다.

## 제 2 장 기 반 지 식

2장에서는 패턴을 인식하여 주가를 예측하는 시스템의 주요한 알고리즘인 신경망의 발달을 역사적 순서에 따라 제안된 주요 모델을 일별해보고, 구조에 관하여 간략히 설명하고, 퍼셉트론을 기반으로 한 신경망을 사용하여 예측했던 기존의 시스템의 활용에 관하여 살펴본다.

### 2.1 개념과 구조

인간의 사고와 지능에 대한 인문을 오래전부터 품어왔다. 1943년 정신과 의사이자 신경 해부학자인 매컬록(McCulloch)와 수학자인 피츠(Pitts)에 의해 신경세포를 모델링한 뉴론(neuron)이라는 매우 단순한 기능을 가진 유닛들을 네트워크를 구성하여 가능성을 보였고, 이것을 토대로 “맥컬록-피츠 뉴런” 혹은 “임계 논리 유닛(Threshold Logic Unit : TLU)”를 고안하였다. 이 모델은 생물학적인 신경세포의 모양을 반영하였다. 생물학적인 신경세포는 수상돌기(dendrite), 축삭돌기(axon) 그리고 세포체(cell body or soma)로 이루어져있다. 수상돌기는 인접 뉴런으로부터 신경 흥분이 세포체로 입력되는 통로이며 축삭돌기는 가늘고 긴 신경섬유로 신경 흥분을 전달하는 역할을 한다. 세포체는 일정 시간동안 입력된 자극을 세포체 내에서 가중(weighed summation)되고 그 값이 임계치(역치)이상이 되면 뉴런이 활성화되어 축삭도리고 자극이 전달되는 매커니즘을 갖고 있다. 마찬가지로 신경망은 수상돌기는 입력 유닛으로 설정하고, 세포체의 가중 합은 각 유닛과 연결된 연결 가중치의 곱의 합으로 계산되고, 이것이 활성화 함수라 하는 함수에 의하여 출력 값을 얻도록 모델링 될 수 있다. 이러한 모델링을 구체적으로 설명하면, 각 유닛마다 가중치와 입력과 출력은 모두 변수로 설정하고, 각 변수는 정규화(normalization)을 거치게 된다. 즉, 입력 값은 수치저그로 설정될 것이고 이를 변수에 저장하는 과정이 입력을 받는 과정이 된다. 그리고 두 뉴런 간에 연결의 세기를 의미하는 가중치도 2차원 배열로 처리하면 입출력에 대한 각각의 연결 가중치를 설정할 수 있을 것이다. 세포체의 기능인 가중의 합도 연산자를 통하여 구현된다. 이렇게 구현된 신경망은 AND, OR, NOT과 같은 연산을 선형분리 하여 해결하였고, 폰-노이만 계열의

컴퓨터가 나오기 전에 이렇게 단순한 연결 구조만으로 가중치가 적절히 조정되면 컴퓨팅의 기본이 되는 소자들을 구현해 낼 수 있는 바로 이것이 신경세포를 모델링한 뉴론(neuron)이다.

## 2.2 발전

맥클로즈-피츠의 뉴런에서 보았듯, 연결강도가 신경망에 결정적인 역할을 한다. 이러한 연결강도는 학습에 의하여 연결강도를 최적화 시킬 필요가 있다. 이것을 헤브가 ‘행동의 조직’에서 이를 정리하는 학습규칙인 “헤브의 규칙(Hebbian learning)” 제안하였다. 이는 어떠한 신경 세포의 활성이 다른 신경세포가 활성화하는데 계속적으로 공헌한다면, 두 신경세포 간의 연결 가중치를 증가시켜 주어야 하는 것을 강조하였다. 이 규칙과 비슷하지만 연결가중치를 조절하는 방법은 다소 차이가 있는 델타 규칙(delta rule)은 퍼셉트론 신경망에 적용되어 놀라운 성능을 나타내었다. 델타 규칙의 적용으로 비로소 목적 패턴으로 신경망의 학습을 인위적으로 조절할 수 있게 되었다. 다음은 델타규칙을 식으로 나타낸 것이다.

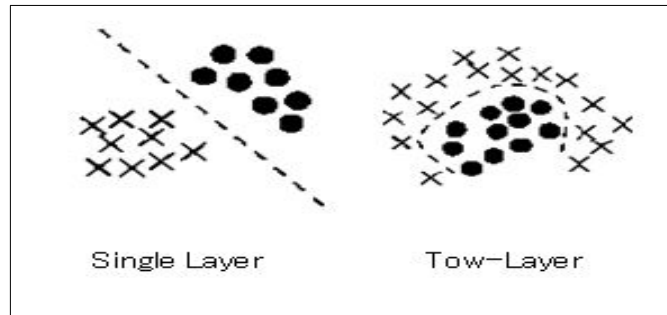
$$W_{ij}(\text{old}) = W_{ij}(\text{new}) + a a_i b_j$$

$$e_j = t_j - b_j$$

[식 1] 델타규칙을 적용한 식

$W_{ij}(\text{old})$  : 신경세포 i, j사이에 조절된 가중치,  $W_{ij}(\text{new})$  : 신경세포 i, j 사이에 연결 가중치,  $a$  : 학습률,  $a_i$  : 신경세포의 i의 활성값,  $b_j$  : 신경세포의 j의 활성값,  $e_j$  : 신경세포의 j의 오차,  $t_j$  : 목적 패턴의 출력층 신경세포 j에 대응하는 성분값

델타 규칙을 수렴한 신경망인 다층 퍼셉트론은 간단히 MLP(Multi-layer Perceptron)이라고 한다. MLP는 은닉층을 갖고 있으며, 층의 개수가 증가할수록 퍼셉트론을 형성하는 결정구역의 특성은 세밀해진다. 그림 1에서 보듯이, 한 개의 층으로는 선형으로 구역을 나눌 수 있기 때문에, xor 등의 간단한 문제에서도 이를 해결할 수 없다. 그래서 퍼셉트론은 여러 층을 연결하여 복잡한 영역을 곡면으로 나누어 결정영역을 구할 수 있게 하였다.



[그림 1] 신경망 층에 따른 결정 영역

각 유닛은 층을 갖고 입력층, 은닉층, 출력층을 이루는 네트워크가 된다. 각 유닛은 인접한 층의 모든 유닛들은 전 방향으로 연결되며, 각 유닛(o)은 하위층 유닛의 출력(x)과 연결된 링크의 가중치의 값의 합(weighted sum)을 계산한다. 출력층은 목표값(t)과 출력값(o)의 차에 학습율( $\eta$ )을 적용하여 차이( $\Delta w$ )만큼 하위 계층 노드의 링크 가중치 값을 조정한다. 조정된 링크 값으로 하위 층 유닛의 출력과 링크의 가중된 합을 계산하여 출력층의 목표값과 출력값의 에러율을 계산한다. 이 과정을 반복하여 예측 결과 값이 허용 오차 안으로 떨어지면 학습을 중단한다. 출력 결과는 0과 1사이의 범위를 갖는다.

---


$$w_i = w_i + \Delta w_i$$

$$\Delta w_i = \eta (t-o)x_i$$

$$o(x) = w \cdot x$$

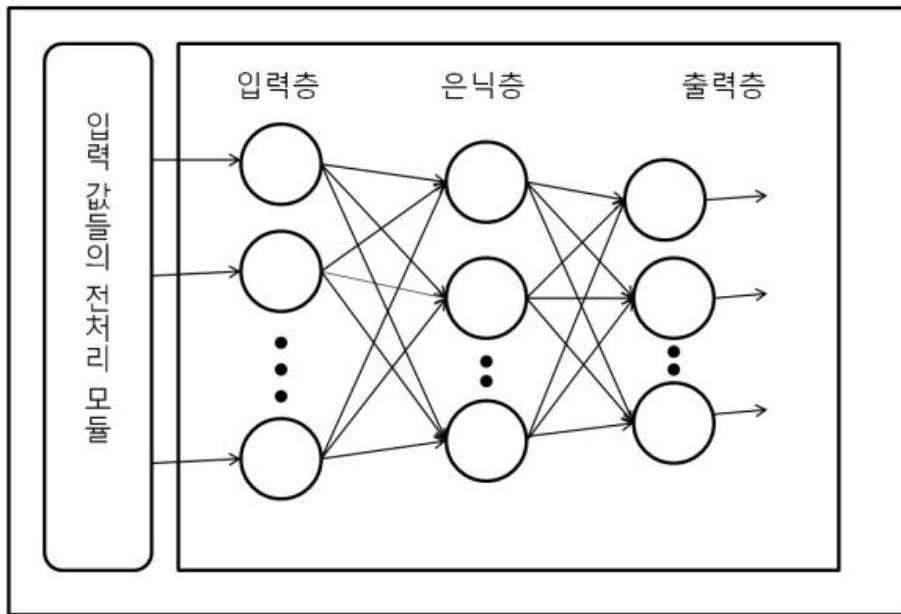

---

x : 하위층 출력의 유닛, w : 링크의 가중치 값

[식 2] 신경망 안의 은닉층에서 가중치 조절 식

출력층 유닛의 출력값은 전달 함수  $f(x)$ 에 의하여 계산되며 따라서 출력층의 에러는 출력값에 대한 전달 함수의 도함수인  $f'(x)$ 와 목표값과 출력값 차의 곱을 통하여 쉽게 구할 수 있다. 하지만 은닉층 유닛들은 목표 출력값이 없기 때문에, 이들의 에러 계산이 미묘해진다. 이 문제는 Rumelhard와 McClelland가 은닉층 유닛들의 에러를 계산하기 위한 방법[5]을 고안해냄으로써 해결되었다. 은닉층의 유닛이 높

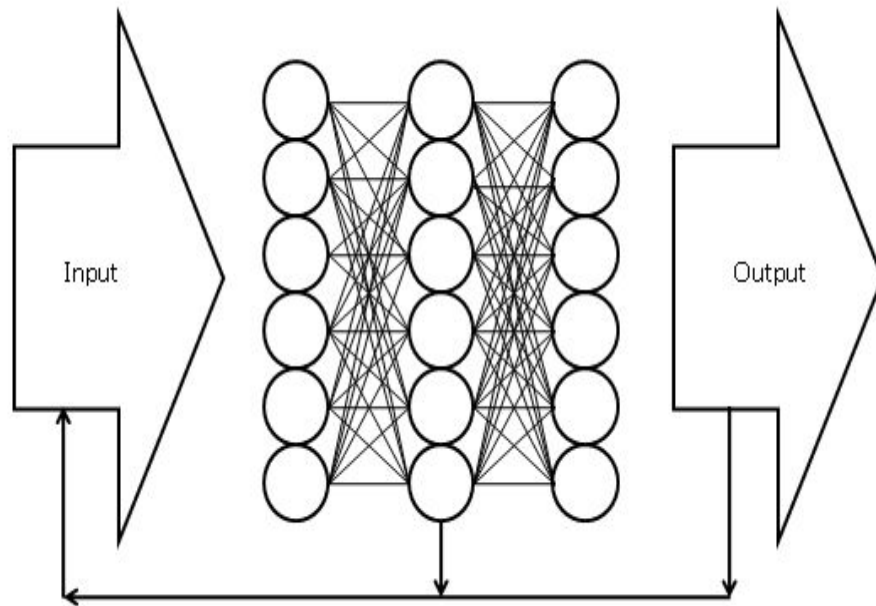
은 에러를 산출하는 출력층 유닛의 출력에 많은 기여를 한다면, 은닉층 노드 또한 높은 에러를 가져야 한다. 출력층 유닛들에 연결된 은닉층 유닛의 링크값들과 그것들에 대응되는 출력층 유닛의 출력과 합을 곱함으로써 은닉층 유닛의 에러를 측정할 수 있다.



[그림 2] 신경망의 기본 구조

$\eta$  : 학습율,  $t$  : 목표값,  $o$  : 출력값,  $i$  : 유닛번호

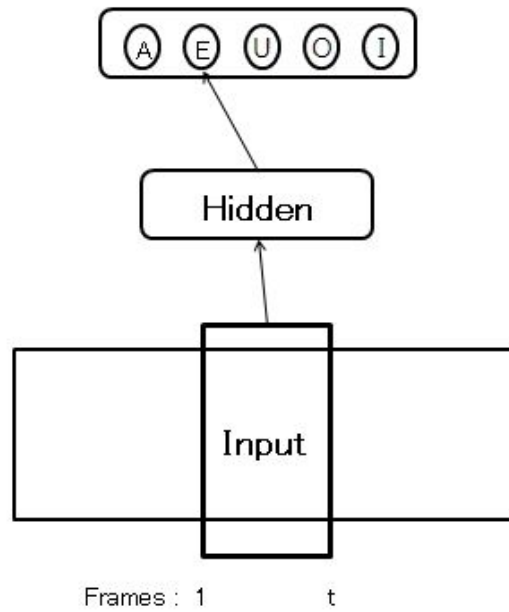
이렇게 완성된 신경망은 입력 패턴과 교사 패턴에 의해 주어지는 원하는 출력 패턴을 쌍으로 제시하면서 반복적인 학습을 하게 된다. 결국 신경망이 학습한다는 것은 전체 오차를 최소화 시키는 연결 강조의 최적화 문제로 생각할 수 있다. 또한 시계열 패턴 인식에 적용할 경우에 만족할 만한 성능을 보여주기 위해서는 동적 패턴인 음성이나 물체의 궤적을 단순한 시점에서의 출력을 계산한다고 해서 되는 것이 아니라, 공간적인 패턴과 동적인 입력의 시계열 패턴을 인식하고 식별할 수 있는 신경망 구조가 되어야 한다. 이와 같은 동적인 패턴을 모델링하기 위해서는 신경망 자체 내에서 주어진 입력 패턴에 대한 시변성을 흡수하는 방식으로 학습이 이루어지는 다이내믹스(dynamics)를 가져야 한다. 이것을 반영한 신경망이 회귀 연결을 갖는 그림 3에서 보여지는 회귀형 신경망이다.



[그림 3] 회귀 신경망

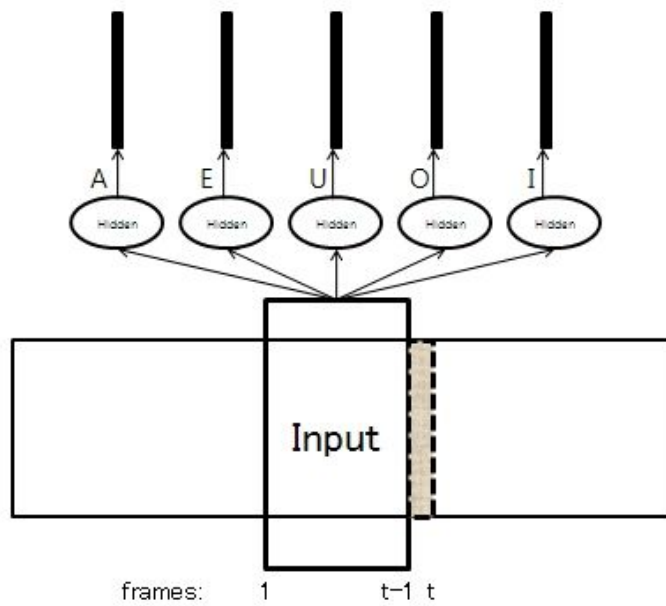
## 2.3 종류와 활용

신경망은 교사 패턴의 설정 방법에 따라서 예측 분류의 능력을 가지게 된다. 즉 어떠한 특징 벡터와 교사 패턴으로 이루어진 학습 데이터 집합을 이용한 반복적인 학습을 통하여 해당 신경망은 그림 4와 같이 패턴을 분류하는 능력을 갖게 된다. 주로 이러한 신경망은 정적 패턴에 대한 분류에 많이 이용된다. 예를 들면 음성 인식을 할 때, 분류형 신경망으로 자질로 표현되는 기준 음소 프레임에 대한 분류가 있다. 기본적으로 하나의 네트워크로 구성이 되며 학습 시에 사후 확률적 교사 신호를 사용할 경우, 인식 시에 출력 유닛 중에서 가장 출력 값이 높은 유닛을 인식 유닛으로 결정하게 되는 사후 확률의 추정에 의해 분류하는 것으로 인식할 카테고리 수만큼 출력 유닛이 필요하며 제한된 범위의 음성 인식에 사용되는데, 이러한 인식 대상 음절수를 확장하기 위해서는 별도의 교사신호를 설정하는 기술이 필요하게 된다.



[그림 4] 분류형 신경망의 기본 구조

그에 반해, 어느 정도 반복적인 특성을 갖는 동적 패턴의 경우에는 입력패턴과 교차 패턴을 동적 특성의 전, 후 패턴을 차례로 인가하는 방식으로 학습하여 해당 신경망의 주어진 입력에 대해 다음의 패턴을 출력하는 예측 능력을 갖게 된다.



[그림 5] 예측형 신경망의 구조

예측형 신경망은 인식 할 카테고리 수만큼 네트워크를 구성할 음성의 전 구간에 걸쳐 순차적으로 전후 프레임을 각각 입력신호와 교사 신호로 사용함으로써 학습된 각각의 네트워크가 뒤에 올 신호를 예측하도록 모델링한다.

기계를 학습시켜 주가를 예측하는 것은 이미 수년 전부터 연구되어지는 분야이다. 기계학습기법으로 신경망(neural network), 귀납적 학습법(inductive learning), 사례 기반 추론(case-based reasoning), 유전자 알고리즘(genetic algorithms), Support Vector Machine(SVM)등이 있다. 이 중 신경망은 높은 비선형성과 동적인 성질을 가진 문제에 통계적 기법보다 예측과 분류를 더 정확하게 수행한다. 또한 변화하는 환경에 쉽게 적응이 가능하며, 샘플의 크기, 변수의 수와 데이터의 분포에 대해서 덜 민감하여 잡음을 포함하고 있는 데이터도 이용이 가능하기에 가장 유력하게 사용이 되고 있다.[6]

### 2.3.1 기존연구

기존 연구[7]에서 보인, 3계층 구조를 가진 다층퍼셉트론(MLP)를 사용하고, 학습 알고리즘으로 역전파 학습방법을(Back-Propagation Algorithm)을 사용하여 한국의 주가 지수를 대표하는 KOSPI 200지수를 데이터로 선정하여 약 1400여개의 데이터를 학습으로 사용하였다. 단 IMF같이 특수한 경제기간 동안은 학습기간에서 제외시킨 모델의 예측능력을 살펴보겠다.[6] 이 연구에서는 신경망 노드수를 6, 10, 14, 18, 22로 하고, 학습률을 0.1로 정하였으며, 실험을 수행하여 살펴본 결과는 다음과 같다.

데이터구분	SET1	SET2	SET3	SET4	SET5
히든 노드수	10	14	14	6	14
학습용	55.5	62.0	60.7	57.0	57.3
평가용	56.0	62.8	56.0	55.4	60.4
검증용	55.0	56.4	56.4	53.0	51.3

[표 1] KOSPI 지수 주간 예측 - 신경망 모형 결과(%)

표 1에서 보면 알 수 있듯이, 노드수가 14일 때, 평가율이 집중적으로 높게 나타났으며, 노드수가 18개와 22개일 때 예측율이 높게 나타나지 않았다. 평가용의 예측율이 높은 5개 셋의 검증용 데이터의 예측율이 낮은 것이 51.3%, 높은 것은 56.4%로 나타났다. 이 연구에서 알 수 있는 것은 신경망 모형이 높은 성과를 보여주었다. 하지만, 입력 변수들이 적절한 변수인지에 관한 심층 연구가 부족하고, 학습데이터에 수(SET)에 한계가 있어 실세계에서 어느 정도의 효율을 보이는지에 대해 알 수 없다는 한계가 있다.

### 2.3.2 단일 신경망 기반 주식거래 시스템

본 절에서는 단일 신경망 기반 주식거래 시스템의 통합 개발 환경에서 만든 NNShell[8]을 소개하고자 한다. 이 시스템은 기술 분석에 기반을 두고 있으며, 날짜별 주가 변동 데이터와 이로부터 계산된 값이나 지표(indicator)들만을 사용하여 주가를 분석하고 예측한다. 우선 방대한 양의 주가를 처리하기 위해서 시계열 데이터에 대한 전처리(preprocessing)과정을 거쳤다. 시계열 패턴별 학습 데이터의 생성과 학습 인자의 조절을 하게 된다. NNShell의 데이터베이스에는 한국 주식 시장을 구성하는 거래소(KOSPI)시장과 코스닥(KOSDAQ) 시장에 상장된 약 2000 종목에 대한 1990년 1월 이후 현재까지의 데이터를 저장하였다. 각 원시 데이터(raw data)는 7개의 필드가 있다.

---

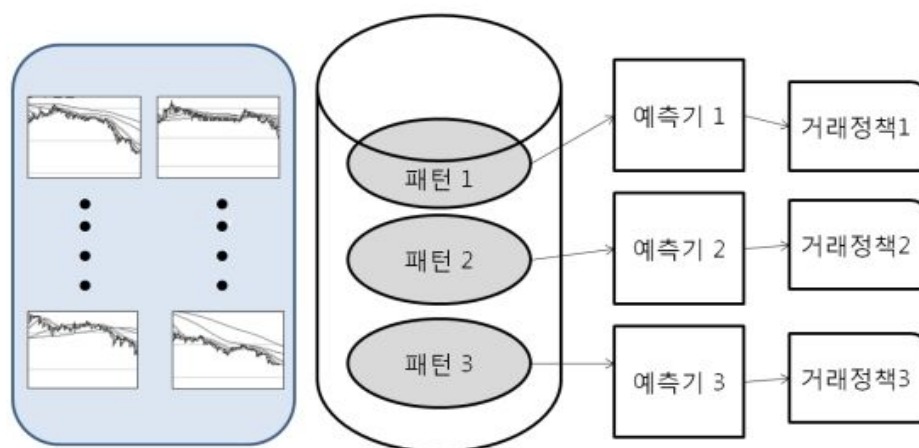

$$RD = (name, date, p_0, p_h, p_l, p_c, v)$$


---

#### [식 3] 원시 데이터

이 식에서 name은 기업명을 나타내고 date는 날짜,  $p_0$ 는 해당 날짜의 거래 시작가인 시가이다.  $p_h$ 는 거래가 진행되는 동안 가장 높은 가격인 고가이다.  $p_l$ 는 거래가 진행되는 동안 가장 낮은 가격인 저가이며,  $p_c$ 는 거래가 종료된 시점에서의 가격인 종가이며 마지막  $v$ 는 거래량이다. 이 데이터로부터 가격 이동평균(moving average)을 비롯하여 기술 분석에서 사용되는 다양한 경제적인 지표들과 함께 시스템을 학습시킨다.

NNShell을 사용하여 개발된 거래 시스템 중에서 사용한 거래 시스템의 기본구조를 가정하고 있다. 하지만 기존의 시스템과는 달리 시계열 패턴별로 별도의 학습과 성능 검증을 거친 독립된 예측기를 사용한다. 그리고 각 예측기의 특성에 따라 고유의 거래 정책을 채택할 수 있다. 다음 그림 6은 이러한 예측기들을 모은 거래 시스템의 구조이다.



[그림 6] 거래 시스템의 구조

신경망 기반 주식 거래 시스템에서 신경망의 출력을 이용하는 가장 일반적인 방법은 특정 임계치의 이상의 신경망 출력을 매수하라는 신호로 인식하는 것이다. 다시 말하면 각 후보 종목에 대해 신경망 출력을 구하고 이 값이 임계치( $\theta$ ) 이상의 경우 그 종목을 매수하게 된다. 신경망의 결과가 주가에 적절한 예측임을 가정한다면, 적절한 임계치의 설정은 시스템의 최종 설정에 큰 영향을 미친다고 볼 수 있다. 그러나 앞에서 보아 왔듯, 기존의 연구는 예측 결과의 활용 보다는 예측 능력에만 초점을 맞추었다. 그러나 실용성을 고려해 본다면, 최적 임계치의 설정을 포함한 적절한 ‘거래정책’의 선정은 예측의 정확성 못지않게 중요한 의미를 가진다.

예측에 필요한 임계치는 각 종목의 매수 신호 생성에 사용되며, 낮을 경우 거래 횟수가 많아지지만 거래 당 수익률이 감소하고, 너무 높게 잡으면 거래 횟수가 너무 줄어서 적당한 임계치를 찾아야 한다. 예측 값이 다음 조건을 만족하면, 매수 신호를 발생시킨다.

---

$$f(x_t; w) \geq \theta$$

---

[식 4] 매수 신호 발생 식

특정 날짜  $t$ 에 주식의 입력 표현을  $x_t$ 로 파라미터로  $w$ 로 규정한다. 보여지듯, 임계치의 값이 나온 각 종목을 저장하여, 거래 정책의 자료로 사용한다.

매수 신호가 발생 하였을 때, 증가 대비 주문가의 비율을 선정하게 되는데, 이것이 바로 매수 할인율(buy discount rate)이다. 만약 매수 할인율( $\alpha$ )이 +3%이면, 증가 10000인 종목에 대해서는 10300원에 매수 주문을 낸다. 높은 매수 할인율은 거래 성립이 쉽지만, 손절매의 위험이 상대적으로 높고, 반대로 낮으면 강항 상승세가 진행 중인 종목을 살 수 없어, 수익을 가질 수 있는 기회를 잃게 된다.

일단 매수 신호가 발생하면 그 다음날 즉,  $t+1$ 의 주문가는  $(1+0.01\alpha)P_c(t)$ 로 매수를 시도한다. 만약 주문가가 저가보다 높으면 거래가 성립될 것이다. 그렇지 않으면 매수는 성립하지 않게 된다.

매수 가격 대비 목표 수익률을 정하기 위해서는 이익실현율( $\delta$ )을 정해야 한다. 이익실현율(25%)인 거래 정책의 경우, 10000원에 매수한 종목의 가격은 12500원이 되면 이 가격에 매도한다. 하지만 언제까지 이익 실현을 위해 주식을 묵혀둘 수 없다. 이를 위해 최대 보유 일수를 정한다. 매수 당일 0일로 하여 보유 종목 주식을 보유할 수 있는 최대 기간인 최대 보유일( $\Upsilon$ ) 정한다. 예를 들어, 최대 보유일이 5일이면, 매수 당일 이후 5일 후까지 주가 범위 내에 이익실현율에 해당하는 가격이 나오지 않으면, 현 가격에 매도하는 것으로 가정한다. 하지만, 구매 후 떨어지는 종목에 대해서는 이익실현을 위해 최대 보유일까지 기다릴 수는 없다. 이것을 위해 손절률( $\beta$ )을 정해놓는다. 예를 들어 손절률이 10%인 거래 정책에 대해서는 10000원에 매수한 종목의 가격이 매수 당일로부터 보유기간 이내에 9000원의 가격이 주가 범위에 나타나면 이 가격에 매도하는 것으로 가정한다.

앞서 말한 것을 정리하면 실제 거래 시뮬레이션에 앞서 매수 할인율( $\alpha$ ), 손절률( $\beta$ ), 최대보유일( $\Upsilon$ ), 이익실현율( $\delta$ )을 정하게 된다. 주어진 예측기에 대해 각 거래 정책별 시뮬레이션을 하게 된다. 조합의 수는 셀 수 없을 정도로 많겠지만, 모든 것을 실험하기엔 현실적으로 불가능 하므로, 이 중 몇 가지에 대해 실행한 시뮬레이션을 보도록 하겠다.

매수가 발생한 거래 집합을  $T = \{(name_n, i_n, j_n) | n = 1, 2, \dots, N\}$ 과 같이 표현한다. 이 때,  $N$ 은 총 거래수이고, 괄호 안은 각각 종목 이름, 거래 시작일, 거래 종료일을 뜻한다.  $n$ 번째 거래의 수익률은 다음과 같이 나타낸다.

$$PPT(\text{거래당 수익률}) = \frac{100}{N} \sum_{n=1} \frac{P_{ask}(j_n) - P_{bid}(i_n)}{P_{bid,n}}$$

[식 5] 거래당 수익률의 수식

$P_{bid}(i_n)$ 는 매수 할인가를 지정한 이후의 실제 주문가이고,  $P_{ask}(j_n)$ 는 이익 실현을 계산하여 다음과 같은 식을 이용해서 구한다.

$$P_{bid} = \min(p_0(t+1), (1 + 1.01\alpha)p_c(t))$$

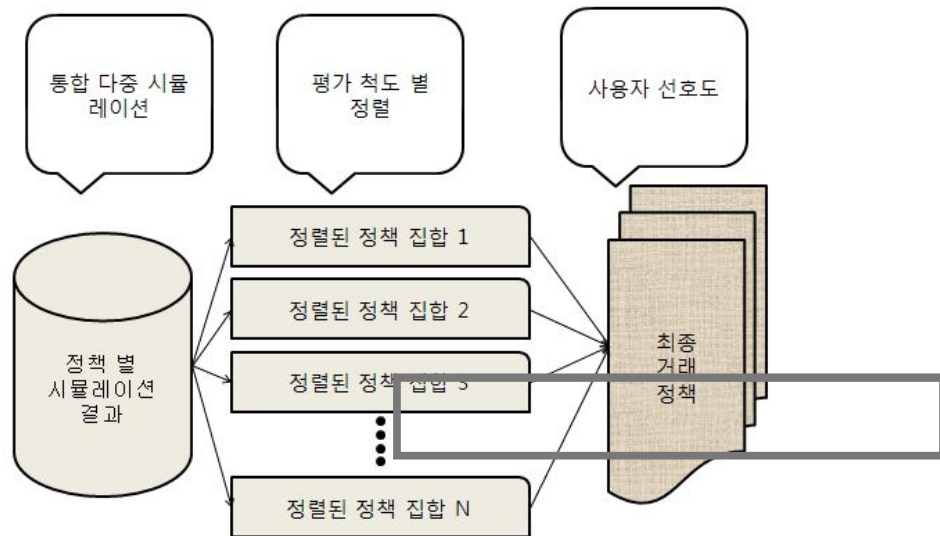
$$P_{ask} = \begin{cases} (1 - 0.01\beta)p_{bid} & \text{if } \exists t' \text{ s.t. } p_l(t') \leq (1 - 0.01\beta)p_{bid} \\ (1 + 0.01\delta)p_{bid} & \text{if } \exists t' \text{ s.t. } p_l(t') \leq (1 + 0.01\delta)p_{bid} \\ p_c(t + \gamma) & \text{otherwise} \end{cases}$$

[식 6] 최종 매수가격( $P_{bid}$ )과 매도 가격( $P_{ask}$ )의 수식

매수 할인율( $\alpha$ ), 손절률( $\beta$ ), 최대보유일( $\gamma$ ), 이익실현율( $\delta$ )을 뜻하며,  $t$ 는 오늘 날짜이다. 이렇게 구한 PPT는 전반적인 거래의 성능을 직관적으로 표현할 수 있는 가장 일반적인 거래 성능 평가 척도이다.

[그림 7] 거래 정책의 선정 과정

그림 7은 앞서 정한 거래 정책에 맞추어, 종목을 선정하고 시뮬레이션 하는 과정을 도식화 한 것이며, 이 과정을 실행하기 위해 NNShell시스템을 사용하여 성능평가를 실행한다. 그에 앞서, 예측하기 위한 파일 선정은 다음과 같다.



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

[표 2] 실험 데이터의 집합

위의 표 2는 정책을 검증하기 위해 각 데이터의 해당 기간을 표기하고 있다. 각 신경망은 기존 연구[8]에서 제시한 기술 분석 지표들을 포함하는 총 127개의 입력 자질을 사용하여 학습시킨 것이다.

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	0.558	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

[표 3] 신경망의 구조와 학습데이터에 따른 예측 성능

위의 표 3는 적정 신경망을 찾기 위해 신경망의 구조와 서로 다른 학습 기간들에

대한 학습결과이다. 표에 나와 있듯, 2개의 은닉층을 사용한 것이 1개의 은닉층을 사용한 것보다 성능이 우수하였으며, 그 중에서도 각 층에 40개와 20개의 노드를 둔 것이 가장 성능이 우수하여, 이 신경망을 기반으로 학습시킨 시뮬레이션을 보인다.

정책명	기준척도	예측 임계치	매수 할인율	이익 실현율	손절률	최대보유일
<b>P1</b>	PPT	0.30	-2%	22%	14%	10
<b>P2</b>	HR	0.30	-2%	10%	16%	12
<b>P3</b>	TP	0.27	-2%	22%	14%	13
<b>BP</b>	-	-	0%	16%	16%	-

[표 4] 정책 선정 기간에 대한 평가 척도별 우수 거래 정책

BP(Baseline Policy)는 기본정책이며, 평가 척도인 PPT와 HR, TP의 각 관점에서 가장 성능이 좋은 정책들(P1~P3)로 구성하여 성능을 평가한다.

정책명	거래횟수	PPT	이익실현	이익	손절	손실	TP
<b>P1</b>	76	4.22%	26%	13%	22%	15%	320,720
<b>P2</b>	76	3.68%	58%	1%	14%	3%	279,680
<b>P3</b>	113	3.14%	34%	22%	31%	26%	354,820
<b>BP</b>	89	2.89%	45%	8%	21%	15%	257,210

[표 5] 우수 거래 정책들의 성능평가

높은 PPT를 선호하면 P1을, 높은 이익실현 비율을 선호한다면 P2를 거래 선정 모델로 삼을 수 있다. [표 5]는 각 정책에 따른 거래 정책들의 성능 평가의 결과이다. 양의 수익을 달성한 이익을 세부 구분하여 거래 회수와 비율을 제시 하였다. TP는 거래 당 100,000원을 투자한 것으로 가정하여 산출한 수익금이다.

이러한 분석에서 예측기 별 고유 거래 정책을 수립할 수 있으며, 어느 정도의 수익을 낼 수 있다는 것을 보여주었다. 하지만 실제 거래 시에 제안된 가법을 적용할 때에는 구간 별 성능도 살펴봐야 하며, 각 신경망을 사용하여 얻은 이익실현보다 더 높은 실현을 위한 새로운 방안이 모색되어야 한다. 또한 손절 손실이 있을 경우, 실제 손절 금액을 낮추기 위한 위험 관리(risk management)도 고려해야 할 것이다.

## 제 3 장 다중 신경망

본 장에서는 신경망을 효과적으로 학습시켜 좀 더 신뢰도 높은 결과를 획득하기 위하여 상호 독립적으로 학습된 신경망의 결과를 의사결정트리 기법으로 결합하는 방법을 제시한다. 이론적으로는 주어진 문제를 해결할 수 있는 신경망을 항상 존재한다는 것이 증명되어 있지만, 현실적인 문제에서 단순한 구조의 신경망은 학습시키기 어려울 뿐만 아니라 학습된 신경망이 제대로 일반화하지 못하는 경향이 있다. 이를 해결하기 위해서 주어진 문제에서 특정 자질을 고려하여 결정을 내리도록 각 신경망을 학습시킨 후, 이들의 결과를 일종의 척도로 보고, 의사결정트리에 의해 통합하여 성능을 개선하는 방법을 제시하겠다. 그리고 제시된 방법의 유용성을 입증하기 위해, 실제 거래에 적용시키도록 하겠다.

### 3.1 개념과 기술

신경망은 패턴 분류 및 인식, 비선형 시스템의 식별과 제어, 그리고 로봇 제어 등과 같은 매우 다양한 응용 분야에서 활발히 사용되고 있다. 문제에 따라 적용되는 신경망의 모델은 매우 다양하게 존재하고, 그 중 교사 학습을 수행하는 대표적인 신경망으로는 2절에서 살펴본 다층 퍼셉트론 신경망을 들 수 있다. 실제 적용에 있어서 신경망은 대규모의 크기 및 매우 복잡한 문제의 해결을 요구한다. 따라서 신경망은 이러한 문제들에 대하여 유연하게 대처할 수 있는 능력을 지녀야 한다. 현재 널리 쓰이는 신경망 학습 알고리즘으로 가장 널리 쓰이는 에러 역전파 학습알고리즘이다. 가장 일반화 되었지만 단일 다층 신경망에서 학습시키므로 주어진 문제의 크기가 크고 복잡할수록 그리고 학습시간이 길어질 수 있다. 무엇보다도 가장 중요한 문제는 크고 복잡한 문제에서 단지 한 개의 신경망으로 좋은 결과를 낼 수 있는냐이다. 네트워크의 크기에 제한이 없고 무한한 양의 데이터가 제공되는 경우에는 주어진 문제를 해결할 수 있겠지만, 실제적으로는 사용할 수 있는 자원의 양은 한정되어 있기 때문에 실제 문제에 적용할 만큼 충분한 기능을 하도록 신경망을

학습시키기란 그리 쉬운 일이 아니다.

최근에는 이와 같은 문제를 극복하기 위해서 다양한 유형의 모듈화 신경망이 연구되고 있으며, 그중 일부는 동일한 입력에 대하여 서로 다른 조건하에서 여러 개의 신경망을 학습시킨 후 각 신경망의 결과를 적절하게 결합하여 최종결과를 내도록 하는 방법을 사용하였다. 이외에도 한 가지 문제를 해결하기 위하여 서로 다른 입력 값으로 어느 정도 이상의 신뢰도 있는 신경망 시스템을 여러 개 두어 신경망의 결과 값을 결합하는 연구[9]도 진행되었다. 단순히 출력 값의 평균을 사용하는 방법에서 부터 투표하는 방법, 그리고 Dempster-Shafer belief-based 방법, Supra Bayesian 방법, stacked generalisation 방법 등이 이러한 연구들의 결과이다. 다중 신경망을 사용하면, 하나의 신경망이 잘못된 결과를 도출하더라도 다른 신경망에 의해 보완될 수 있다. 그렇다면 각각의 신경망의 값을 어떻게 결합할 것인가가 성능을 결정한다. 이러한 신경망의 결합의 과의 형태에 따라 세 가지 레벨로 나눌 수 있다. 하나의 결과만을 출력하는 추상 레벨(Abstract level)과 순위 형태로 출력하는 순위 레벨(rank level), 그리고 순위와 더불어 신뢰 값까지 함께 출력하는 측정치 레벨(measurement level)이 있다.

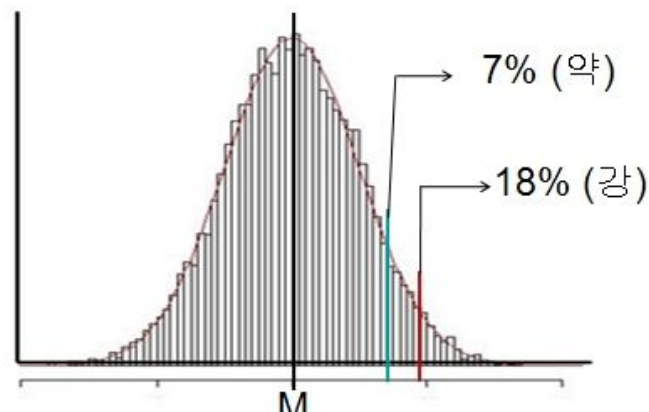
먼저 추상 레벨은 각 신경망이 하나의 결과만을 출력하는 방법으로 다수결 방법, 베이시안 방법 등이 있다. 이 방법의 경우, 제공되는 정보가 적기 때문에 신경망의 과거 성향이 결합에 크게 관여하게 된다는 장점이 있다. 순위 레벨은 각 신경망이 인식에 참여하는 모든 클래스의 순위를 결과로 출력 후, 모든 클래스들의 순위를 비교하여 결과를 나타내는 기법이다. 이 방법은 보다 많은 정보를 바탕으로 결합을 수행하고, 과거의 성향이 결합에는 거의 사용되지 않는다는 특성이 있다. Borda 함수, 가중화 Borda 함수, Condorcet 함수를 통한 결합이 순위 레벨에 속한다. 마지막으로 측정치 레벨의 방법은 각 인식기가 클래스들에 대한 순위와 더불어 각 순위에 대한 신뢰 값까지 함께 출력하는 방법이다. 그러나 각 인식기 별로 제공되는 신뢰 값이 일관성을 유지해야 한다는 어려움이 있다. 대표적인 방법으로 평균과 가중 평균 방법이 있다. 이러한 방법들 중 각 신경망 기반 시스템에 맞는 방법을 선택해야 신뢰성을 높일 수 있을 것이다.

## 3.2 다중 신경망 기반 주식거래 시스템

본 절에서는 기존 2.3.1 절에서 보인 단일 신경망 기반 주식거래 시스템의 성능을 높이기 위해 각각의 신경망에서 나온 출력 값을 결합하여 주가 거래 시스템의 성능을 높이는 기법을 소개한다.

### 3.2.1 Counting 기법

기존 연구[7]에서 다중 신경망 기반 주식거래 시스템은 각각의 신경망의 출력 값을 결합시키기 위해 Counting하는 기법이 추가되었다. 주가를 예측하기 위해서는 과거의 패턴 인식이 중요하다. 그러므로 과거 성향이 크게 관여하는 추상레벨을 적용하기에 적합하다. Counting기법은 각각의 신경망이 각 종목의 출력 값을 정규화 하여, 임계치 이상의 값을 내는 종목을 저장한다. 일정 이상의 값은 강한 신호로 인식하고, 좀 더 낮은 값의 이상은 약한 신호로 인식하여, 모든 신경망에서 나온 각각의 종목의 출력값의 신호를 세는 기법이다



[그림 8] 각 신경망 결과에 대한 정규 분포표

그림 8은 각각의 신경망마다 정규분포표이다. 이것을 각각의 신경망마다 적용한다. 각 종목에 관한 신호에 관하여 상위에서 18%까지 순위 내에 드는 출력을 강한 신호로 인식하고, 신호 '강'의 개수를 하나 늘린다. 상위 18% 이상 25% 이내의 신호는 신호 '약'의 개수를 하나 늘려 표를 만든 후, 이를 종합한다.

종목	종목 코드	강	약
아남정보기술	050320	4	5
SY	004530	4	5
인피트론	045470	4	4
미주레일	078940	3	6

[표 6] Counting 기법의 예

위의 표는 Counting 기법의 예이다. 각 종목에 관한 각각의 엔진에서 내는 신호를 하나씩 적립하여 종합한다. 이러한 방법은 별도의 계산 과정이나 기억 장소 없이 다중 신경망의 결과를 결합할 수 있다. 각 신경망이 공통으로 강하게 낸 신호의 종목은 각 신경망이 투표를 한 것이므로, 종목의 신뢰성을 높이는 효과를 가져오며, 기존 시스템보다 좀 더 나은 신뢰성을 갖게 된다. 하지만 각 신경망에 관계없이 동일한 기준을 적용하기 때문에 상대적으로 성능이 좋지 않은 신경망이 포함되어 있으면, 결합 결과에 영향을 미치므로, 이를 주의하도록 한다.

### 3.2.2 신경망의 값의 합

각각의 신경망의 출력 값을 결합시키기 위한 또 한 가지의 방법은 각각 신경망이 내는 출력 값의 합을 구하는 것이다. 이러한 방법은 측정치 레벨의 결합 방법이라고 볼 수 있다. 단일 신경망에서 KOSPI, KOSDAQ 종목마다 예측치의 값을 출력하는데, 임계치 이상의 값을 유효하다고 생각한다. 이렇게 유효한 값을 내는 신경망의 예측치의 값을, 각 종목마다 더하게 된다. 설혹, 한 신경망에서 옳지 못한 답을 예측하여도, 다른 신경망이 이러한 점을 보완해주어, 신뢰도를 높인다. 또한 높은 신뢰성의 신경망을 모아 결합하였기 때문에 측정치 레벨의 결합 방법이 유효하다.

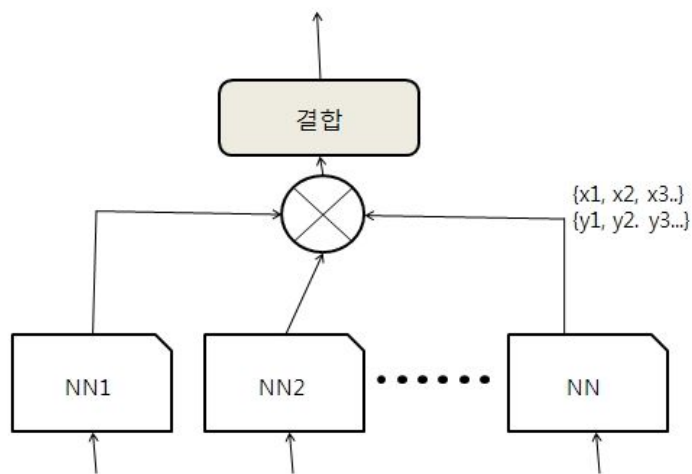
종목	종목코드	SUM
아남	050320	1.29
SY	004530	1.18

종목	코드	SUM	Kang	YAK
아남	050320	1.29	4	5
SY	004530	1.18	4	5

[그림 9] 예측치를 적용한 신경결합 기법의 결과 예제

그림 9는 위의 앞서 보여준 표에서 보여준 항목에 예측치가 추가되었음을 볼 수 있다. 각 신경망은 신뢰성이 이미 입증되었으므로, 이 예측치의 값 역시 신뢰성을 갖고 있으며, 여러 개의 신경망의 합을 사용하여, 성능을 높였다.



[그림 10] 다중 신경망의 모식도

위의 그림은 다중신경망의 모식도이다. 자세히 말하면 각각의 신경망에서 종목에 대한 일정 이상의 임계치 값 이상의 예측치를 x에 저장하여 공통된 종목을 NN1, NN2 등 모든 신경망의 값을 더한다. 또한 각 종목에 관한 강하고 약한 신호를 찾아내어 이것을 투표하여 종합적으로 숫자를 더한다. 이것에 대한 결과는 [그림 10]을 참조한다.

## 제 4 장 의사 결정 트리

주식을 사고, 파는 종목 선정에 있어서 기존 장에서 언급한 다중 신경 기반 주식 거래 시스템이 효과를 발휘한다. 하지만 실제 거래에 적용함에 있어서, 각 엔진에 대한 결과를 더한 예측 값과, 신경망에서 나온 시그널의 강·약의 개수가 효과적으로 적용되기 위해서는 이를 어떻게 적용해야 할지에 대한 방안이 필요하다. 다시 말해 마지막 결정에 대한 문제이다. 의사 결정을 위해서는 이러한 정보를 얻고, 평가한 다음 최선의 선택을 해야 한다. 이 모든 것은 매우 단순해 보이고 일반화된 논의이다. 그러나 주가의 종목선정과 같이 불확실성에 대한 문제가 복잡해지면 질수록 사람의 직관적인 의사결정은 도움이 되지 못한다. 그렇다면 불확실성 상황에서 ‘최선’이란 무엇을 의미하는지에 대한 문제가 제기된다.

의사결정트리(Decision Tree)는 의사 결정 규칙을 도표화 하여 관심 대상이 되는 집단을 몇 개의 소집단으로 분류(Classification)하거나 예측(Prediction)을 수행하는 계량적인 분석방법이다. 이러한 특징으로 DB마케팅, CRM, 시장조사, 광고 조사, 의학 연구, 품질 관리 등의 다양한 분야에서 활용되고 있으며, 구체적인 활용 예는 DM응답자 분석, 고객 타겟팅, 고객들의 신용 점수화, 캠페인 반응 분석, 고객 행동 예측, 고객 세분화, 시장 세분화, 신상품 수용도 분석, 광고 효과 측정, 상표 이미지 테스트 등 광범위한 범위에서 활용되고 있다. 본 절에서는 앞서 언급한 다중 신경망 기반 주식 거래 시스템을 종합한 결과를 의사결정 트리를 사용하여 기존 주식 거래 시스템의 성능을 개선하는 방안을 제시한다.

## 4.1 필요성

의사 결정 트리(DT)는 명목형 목표 변수를 기준으로 마디가 분화되는 분류 트리(Classification Tree)와 연속형 목표 변수를 기준으로 마디가 분화 되는 회귀 트리(Regression Tree)로 나눌 수 있다. 분류트리의 경우에는 목표 변수의 각 범주에 속하는 빈도(frequency)에 기초하여 분리가 일어난다. 예를 들어, 신용상태가 'Good'와 'Bad'의 값을 갖는 명목형 목표 변수를 기준으로 나이, 직업을 고려하여 전체 고객을 세분화 하는 과정을 설명해 준다. 첫번째 단계에서 목표 변수와 나이, 직업 등의 각각의 변수 간의 상관도가 가장 밀접한 변수를 선택하고, 선택한 변수의 범주를 다양하게 조합하여 가장 상관도가 높은 범주 조합을 선택하여 첫번째 분리를 한다. 직업보다 나이가 신용상태( 표 변수)와 상관도가 높으면 나이를 우선 선택하고 나이의 모든 범주 조합 중에 가장 상관도가 높은 조합을 선택한다. 의사결정트리 알고리즘에 따라 제곱 통계량(Chi-Square Statistics), 지니 지수(Gini Index), 엔트로피 지수(Entropy Index)등을 사용한다.[10]

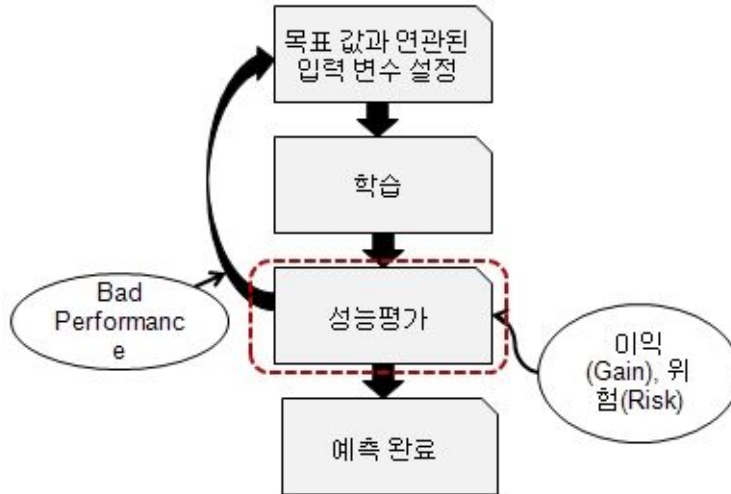
회귀나무(Regression Tree)의 경우에는 각 분리 마디별 목표 변수의 통계량 또는 분산의 감소량에 의해 마디를 분리한다. 목표 변수와 설명 변수들 간에 가장 큰 변수를 먼저 선택하여 분리한다.

의사결정트리는 계량적인 자료의 근거한 분석기법으로 사용자가 원하는 수준에 이르기까지 가지치기를 멈추지 않고 뻗어 나간다. 정지규칙이 나오기 전까지 분리를 하게 된다. 정지규칙은 최대 나무의 깊이, 자식의 마디의 최소 관측치 수, 또는 엔트로피 지수 등을 사용할 수 있다. 이렇게 생성된 의사 결정 트리는 여러 개의 신경망의 값의 종합을 좋은 등급과 좋지 않은 등급으로 분류를 하여, 최종 단계에서 주가 거래 종목을 선정하는 데 있어서 유용하게 쓰일 수 있으며, 신뢰도를 높일 수 있는 방법이 될 것이다.

## 4.2 분석

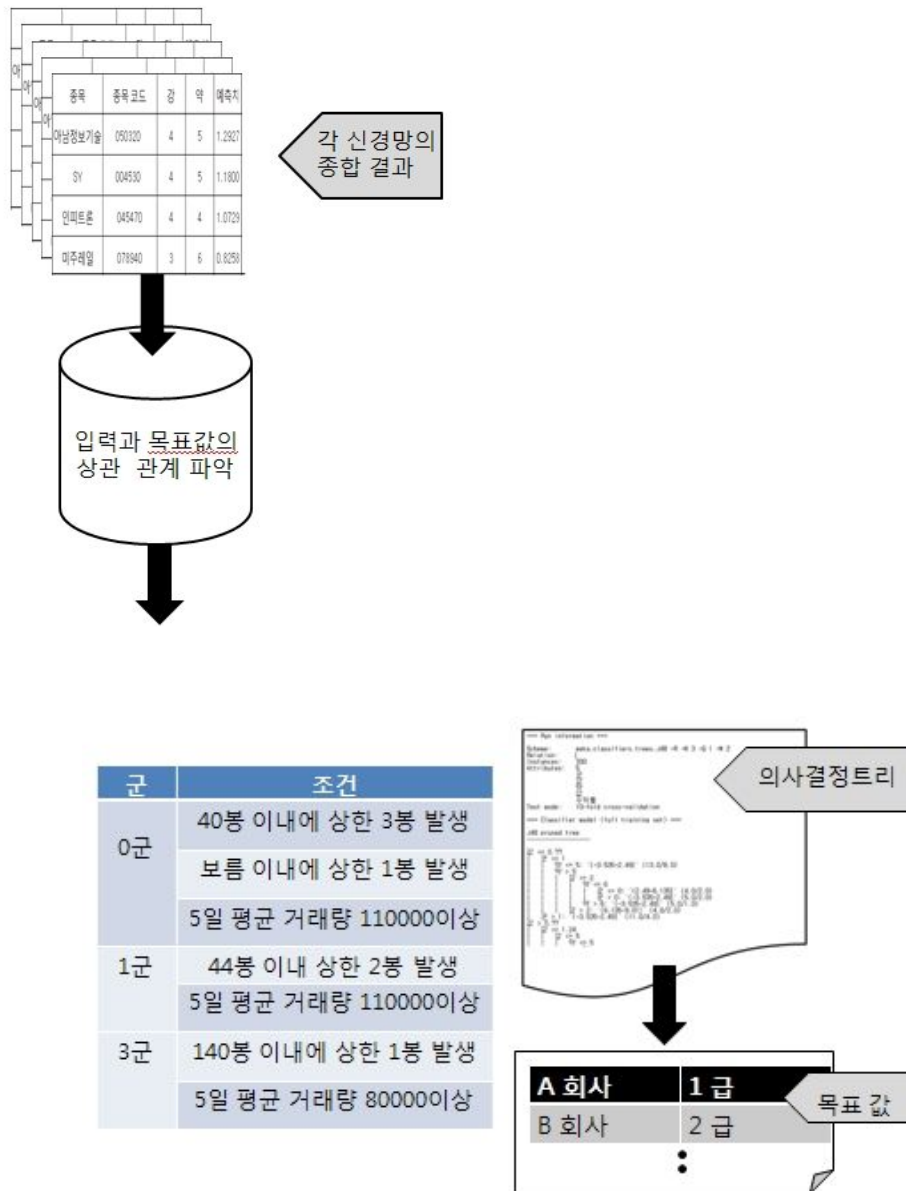
### 4.2.1 분석과정

의사 결정 나무의 분석 과정을 요약하면 다음과 같다.



[그림 11] 의사 결정트리의 분석과정

다중 신경망 기반 주식 거래 시스템의 목표는 각각의 신경망에서 나온 최종 후보 종목들에 대한 오르는 정도를 파악할 수 있는 것이다. 이 목표값을 위한 입력 값은 3.2절에서 설명하였고 기존연구[7]에서 이미 언급된 Counting값과 신경망 값의 합이다. 그리고 상한가를 많이 치는 주식 종목이 미래에도 상한가를 많이 친다는 가정 하에, 가장 많은 상한가를 치는 코스닥과 코스피의 종목을 각각 0군으로 설정한다. 좀 더 느슨한 조건을 주어 1군, 그리고 더 낮은 조건을 3군으로 각각 설정한다. 각각의 다른 조건의 군도 의사 결정에 영향을 미치는 입력의 자질로 선정하여 최종 종목의 급을 구분하게 될 것이다.

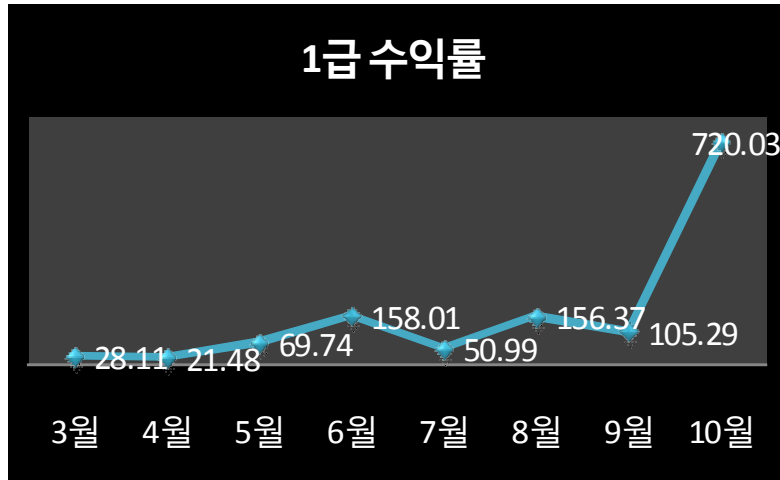


[그림 12] 입력 값 설정에 따른 DT

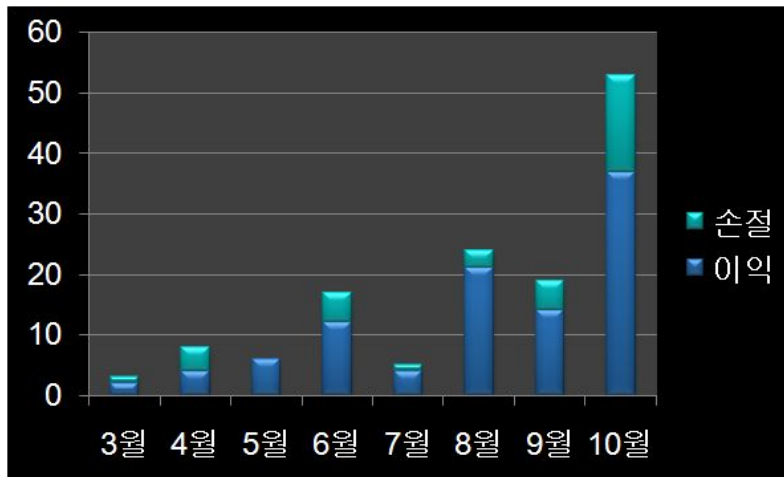
시뮬레이션의 결과가 좋지 않으면, 실험 설계가 불확실하다는 것이며 설계의 많은 부분을 다시 고려해야 하며 그림 11과 같이 목표 값과 그 값에 대한 입력 값을 재 설정한다.

그림 12는 기존 연구[7]에서 보인 단일 신경망 기반 주식 거래 시스템에서 제시한 모델의 각 신경망 출력을 종합하여, 각 종목의 신경망의 값의 합과 신경망 신호의 강약을 분석하여 Counting한 값, 그리고 주가의 고저를 분석하여 분류한 0, 1, 3군을 토대로 더욱 신뢰성 있는 주식 종목 선정을 제시하였다.



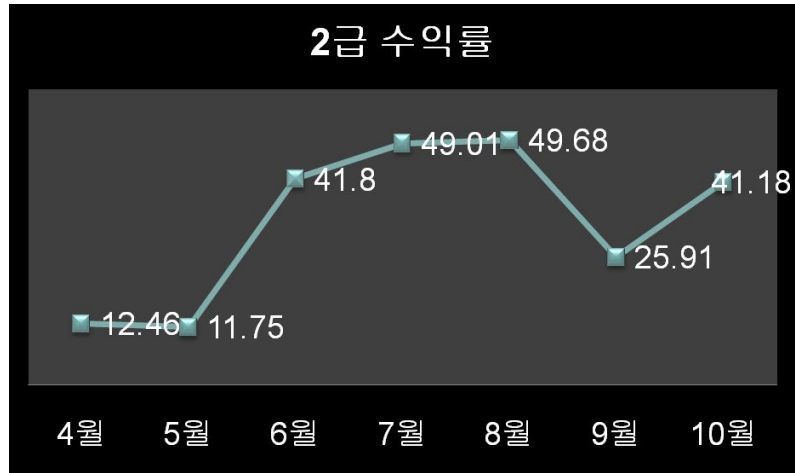


[표 7] DT 결과의 1급의 수익률



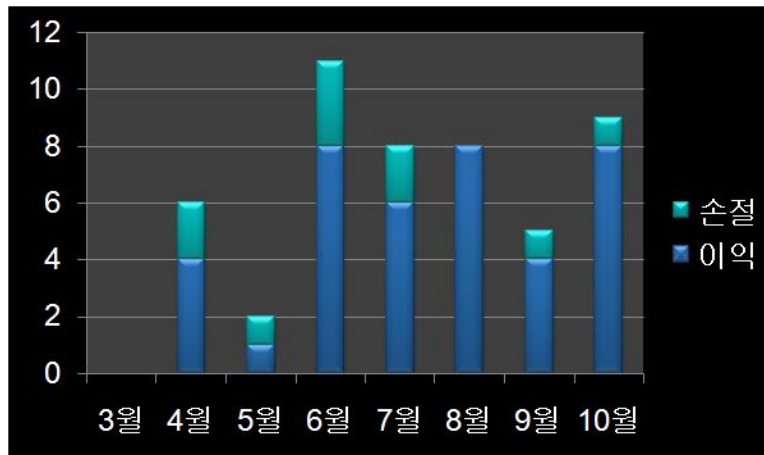
[표 8] 1급으로 선정된 종목의 정확도

위의 표 8은 의사결정 트리의 결과 중 가장 좋은 그룹으로 설정된 항목인 1급에 대해서 시뮬레이션을 실행했을 때이다. 2급과 1급 보다 낮은 추천 수임에도 불구하고 가장 높은 수익률을 보이며, 전체적인 시장이 하락한 7월에는 역시 같이 하락하였지만, 주가 시장이 상승세를 탔던 9월과 10월에는 70%이상의 신뢰성을 갖고, 최상의 실현을 하였다.



[표 9] DT 결과 중 2급의 수익률

위의 표는 DT의 결과 중 가장 두 번째로 높은 그룹으로 설정된 항목에 대해서 시뮬레이션을 실행했을 때이다. 전체적인 시장이 하락한 7월에는 역시 같이 하락하였지만, 시장이 상승세를 탔던 9월과 10월에는 65%이상의 신뢰성을 갖고, 최상의 실현을 하였다.

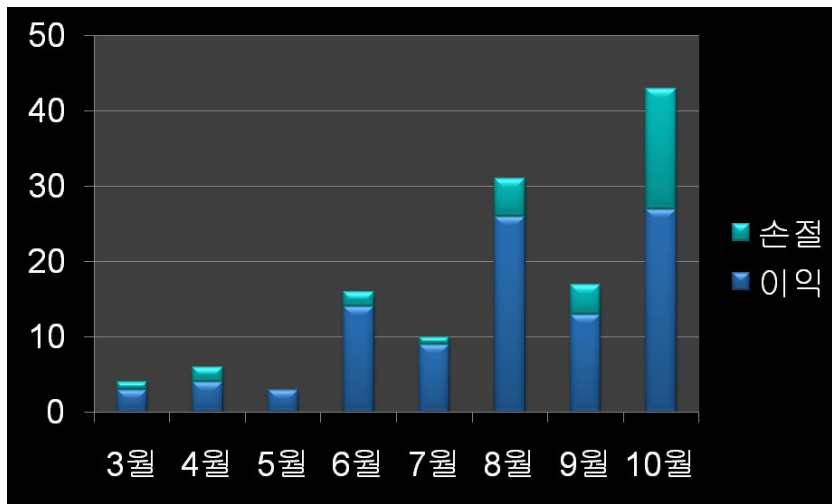


[표 10] 2급으로 선정된 종목의 정확도

하지만, 2급인만큼, 신뢰성에는 비슷하지만, 1급보다 높은 추천 수에도 불구하고 수익률의 정도는 1급과는 판이하게 차이를 볼 수 있다. 이는 DT에 의해 그려진 그래프가 급수에 따라 잘 나뉘었음을 보이는 중요한 부분이기도 하다.

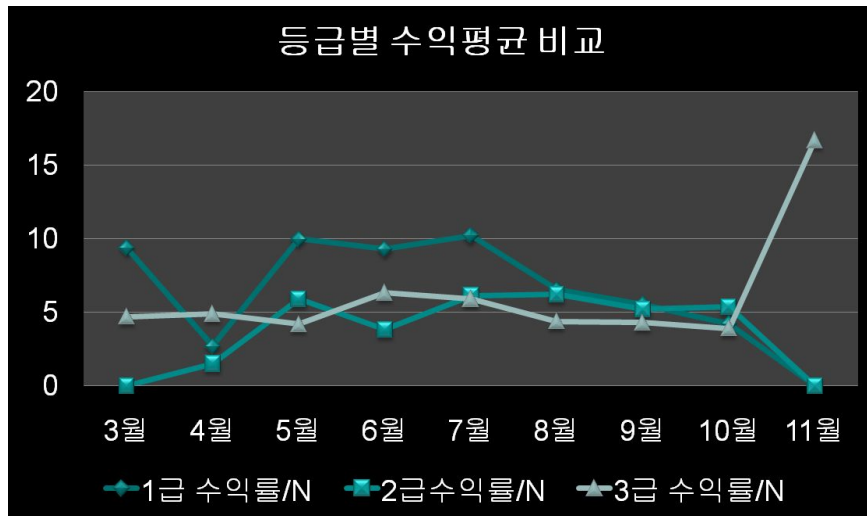


[표 11] DT 결과의 3급의 수익률



[표 12] 3급으로 선정된 종목의 정확도

위의 12번 표는 3급의 추천 종목에 관한 수익률이다. 매달의 수익률은 차이가 있지만 전체적으로 2급에 비해 높은 수익률을 보이는 이유는 더 나은 추천이 아니라, 낮은 제약으로 2급보다 더 많은 추천을 내었고, 그 추천 수만큼 더 많이 누적되었기 때문이다. 상승세를 탔던 타지 않았던, 일정 수준 이상의 수익을 내는 것을 볼 수 있다.



[표 13] DT 결과 각 급당 수익률/추천개수

위의 표는 각 급 당 수익률을 나타낸 표이다. 이 표에서 나타낸 세로 값은 각 수익률을 추천개수로 나눈 값이다. 다시 말해, 그 추천이 옳았든, 옳지 않았든 모든 추천 수로 나누며 이것은 각 급이 갖고 있는 추천에 대한 효율성을 볼 수 있다. 전체적으로 1급이 가장 높은 수치를 보이지만, 높은 수익률을 보이는 만큼 위험성을 갖고 있음을 볼 수 있다. 이 부분에 관해서는 개선해야 할 부분임을 알 수 있다. 하지만 2, 3 급에 비교해서는 좋은 효율을 갖고 있음을 알 수 있다.

등급	이익 거래량	무득실	손절 거래량	거래량
1등급	114	10	23	153
2등급	35	4	10	50
3등급	100	16	20	145
SUM	249	30	53	322
비율(%)	71.56%	8.62%	15.2%	

[표 14] 각 등급의 손익 거래량

위의 표에서 보듯이 1 등급에서 이익을 내는 거래량이 가장 많으며 또한 비율도 높음을 알 수 있다. 상대적으로 적은 추천 수가 나오는 2등급은 거래량자체가 적어 비교하기 힘들다. 마지막으로 이익이 상대적으로 작은 3등급은 비교적 많은 추천 수를 내었고, 이익률도 상당히 높음을 볼 수 있다. 본 논문에서는 등급에 따른 월별

수익률을 사용하여 거래율을 비교·분석 하였다. 이러한 자료를 토대로 결과적으로는 투자자에게 심리전에 흔들리지 않고 신뢰성 있는 주식 종목을 제공할 수 있는 새로운 메커니즘으로 유용하게 사용될 것으로 판단한다.

## 5 장 결론

최근 주가를 예측하는 일이 어렵고 복잡한 과정에 과학적이고 효율적인 방법을 제시하는 이론이 적용하고 있다. 특히, 신경망은 괄목할만한 성장이 있었으며 신경망 이론을 적용한 주가 예측에 관한 사례연구가 분석되어 수많은 성능 평가 실험이 제안되어 왔고, 성능을 증명하기 위해 다양한 평가 방법이 제시되었고, 시계열 접근법과 거래 시뮬레이션 접근법은 성능 평가가 그의 대한 예이다. 그러나 기존의 접근법은 현실적인 자료가 아닌, 일정 기간 혹은 몇 안 되는 주가 종목 아래에 이를 실행하여 성능을 무의미하게 평가하였고, 예측 분야 역시, 현존하는 주가의 모든 부분을 반영하지 못하여 현실에 적용하기에 그 한계를 드러내었다.

실세계의 복잡한 문제를 해결하기 위해서는 기존의 단일 신경망이 아닌, 일정 이상의 신뢰도를 갖는 복수 신경망 체계를 갖는 시스템의 필요하다. 그렇기 때문에 기존 '단일 신경망 기반 주식거래정책 시스템(Stock Trading System Based on Multiple Neural Networks)'의 성능을 개선하는 방안을 제시한다. 기존 단일 신경망의 장점을 모아 신뢰성을 높일 수 있도록, 각각의 신경망의 값을 의사결정트리를 이용하여 최적의 종목을 선택하였다. 성능이 개선됨을 실제 현실의 주가를 토대로 평가하여 증명하였다. 결과적으로는 분명 이전 보다 더 나은 신뢰성을 보이지만 앞으로 개선되어야 할 점도 보인다. 실제 투자를 하는 것에 있어서, 평균적으로 높은 신뢰도는 중요한 척도이지만, 단 한 개라도 신뢰도를 떨어뜨리게 하는 종목에 관하여 위험관리를 할 수 있도록 적용시켜 이러한 기능을 포함한 최적의 거래 시스템의 대한 연구가 필요할 것이라고 판단되어 진다.

# Abstract

## A Study of Performance Improvement for the stock Trading System Based on Neural Network

Byeon, Jue Yeon

Major in Computer Science

Graduate School

Sungshin Women's University

The stock market is the core of the modern economics in capitalism. The studies on predicting the stock prices are advanced in many research areas. Especially studies in computer science primarily use automatic learning methods, but those studies are short of whether their systems can show the performance reports when applied to the real stock market. The short of performance evaluation couldn't estimate whether the systems were reflected the stock market. In this thesis, the existing Stock Trading System based on the single neural network is examined carefully and to improve the performance, I introduce the new scheme using the sums of generating at the single neural networks and the classified the signals and to be applied the system, decision tree is used. Last, the simulation based on the real stock prices is conducted to demonstrate the performance of the new scheme.

## 참고 문헌

- [1] Sang-Bong Oh, Kun-Chang Lee, "A Neural Network-Driven Decision Tree Classifier Approach to Time Series identification"
- [2] Tae-Hyup Roh, Taek-Ho Lee, In-Goo Han , "Forecasting the Volatility of KOSPI 200 Using Neural Network-financial Time Series Model', 2005
- [3] Woon-Goo Chung, Sang-Youb Lee, "A Study on the Forecasting of the Apartment Price Index Using Artificial Neural Networks" pp51-64, 2007
- [4] Journana Ghosn, Yoshua Bengio, "Multi Task Learning for Stock Selection", pp947-952, 1998
- [5] Rumelhart, D, J. McClelland, "Paralled Distributed Processing", 1986
- [6] Yu-Il Kim, Eun-Kyoung Shin, Tae-Ho Hong, "Comparison of stock Price Index Prediction Performance Using Neural Networks and Support Vector Machine", 2004
- [7] Jae Won Lee, Jangmin O, "Integrated Multiple Simulation for Optimizing Performance of Stock Trading Systems based on Neural Networks", 2007
- [8] 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
- [9] Joon-Hyun Ahn, Sung-Bae, "Fusion of Evolutionary Neural Networks Speciated by Fitness Sharing", 2002
- [10] 오을임, 김구, "불확실성 상황에서의 의사결정 양상에 관한 실증적 연구", 2002