碩 士 學 位 論 文

딥러닝 및 머신러닝 알고리즘 기반의 환율 예측 모형 설계 및 평가

嘉泉大學校 大學院

人工智能學科

人工智能 專攻

金 旭 玕

碩 士 學 位 論 文

딥러닝 및 머신러닝 알고리즘 기반의 환율 예측 모형 설계 및 평가

Design and Evaluation of Exchange Rate Prediction Model Based on Deep Learning and Machine Learning Algorithms

嘉泉大學校 大學院

人工智能學科

人工智能 專攻

金　　旭　　玕

碩 士 學 位 論 文

Design and Evaluation of Exchange Rate Prediction Model Based on Artificial Intelligence Neural Network and Machine Learning Algorithms

위 論文을 人工智能學 碩士學位 論文으로 提出함.

2020 年 12 月 4 日

嘉泉大學校 大學院

人工智能學科

人工智能 專攻

金 旭 玕

이 論文을 金 旭 玕의

工學碩士 學位論文으로 認准함

2020 年 12 月 4 日

審査委員長 황보 택근 ㊞

審 査 委 員 임 준 식 ㊞

審 査 委 員 최 창 ㊞

요 약

금융 서비스 시장에서는 금융 데이터 분석을 통하여 각 금융 시장의 지표를 예측하는 것에 대한 관심이 높아져 이와 관련된 다양한 연구가 활발하게 진행되고 있다. 특히, 인공지능 신경망 기반으로 금융 산업의 추세를 예측하는 연구가 각광 받고 있다. 한국은 기축통화 국가에 해당되지 않고 자원도 없으므로 거시경제지표 중에서 환율만큼 영향력이 큰 거시경제지표는 없다고 해도 과언이 아니다. 환율예측에 실패할 경우 수입업체는 물론, 수출업체까지 부도위기에 직면할 수 있고, 금융부실채권이 급증하는 등으로 국민경제 전체에 충격을 줄 수도 있다. 세계 환율 제도가 고정 환율 제도에서 변동 환율 제도로 이동한 후로 세계 각 나라의 환율의 변동성은 급증하고 있고, 이는 세계 경제의 불확실성을 초래한다. 환율 예측의 중요성이 높아지고 있는 가운데 실제로 환율 예측을 위해 사용할 수 있는 모형은 별로 개발된 것이 없는 게 현실이다. 본 연구는 위의 취지에 따라 우리나라 경제에 많은 영향을 끼치고 있는 경제 기초 변수인 원-달러 환율에 대한 장기 예측 모형을 개발하기 위함이다.

본 논문에서는 원 달러 환율에 영향을 주는 거시경제 변수들을 선택하여 5개의 학습 모델을 구축하여 환율 예측을 시도한다. 그 후 각 예측 모델과의 성능 비교 통해 가장 높은 예측 정확도를 보이는 모델을 제안하고 각 모델로 환율을 예측한 결과, 실제 값과의 오차는 있었지만 환율의 움직임을 파악하는 것이 가능하였다.

목 차

I. 서 론

1.1 연구의 배경 및 목적

환율, 즉 우리나라의 통화와 외국의 통화가 어떻게 교환되는지의 문제는 이미 우리 생활의 일부가 되었다. 환율의 변동은 한 나라의 국민경제와 국민생활에 대하여 중대한 영향을 미치고 있다. 오늘날의 환율변동은 그 자체로서 중요한 국제경제학적 요소일 뿐만 아니라 국제수지, 국내물가, 산업, 통화 및 금융, 외국자본도입에 따르는 원리금상환까지 광범위하게 그 영향이 파급되므로 현대 국제경제에 있어 중요한 화두가 되고 있다. 즉 환율정책의 결정은 그 나라 국제경쟁력의 조타수 역할을 한다고 볼 수 있다.

한국은 내수경제의 의존도 보다는 대외 의존도가 90% 이상인 수출주도형 경제이기 때문에 한국 경제구조는 환율이라는 개념에 굉장히 민감하다. 또한 환율은 주식, 채권, 부동산 등 금융 및 실물 자산에 많은 영향을 미치기 때문에 우리나라처럼 위같은 자산에 돈이 많이 묶여있는 구조를 보이는 나라는 환율의 상, 하락 폭에 따른 리스크에 항시 노출되어 있다고 할 수 있다. 산업 측면에서 원화환율이 하락하면 수출기업들은 같은 물량을 외국에 수출하여도 원화 표시 수출금액이 감소하므로 수출채산성이 악화된다. 반면에 해외 원자재를 많이 사용하여 제품을 생산하는 기업은 제품의 생산단가가 낮아져 가격경쟁력이 높아지게 된

다. 원가절감으로 수출이 확대되고 내수시장의 점유율이 높아져 이들 기업의 채산성이 향상된다. 따라서 국가경제와 기업의 경제활동에 있어서 환율이 중요한 척도이기 때문에, 환율 예측은 무엇보다도 중요하다.

흔히 환율을 예측하는 일은 어렵다고 말한다. 세계 여러 나라의 경제 흐름에 영향을 받기 때문이다. 경제 논리만이 환율에 영향을 미치는 것도 아니다. 각국의 정치, 군사, 사회 현상 등도 환율에 영향을 주기 때문에 기존에 제시된 많은 모형들은 한계점을 가지고 있었다. 본 논문에서는 환율에 영향을 주는 거시경제 변수를 최대한 활용하여 머신러닝과 딥러닝 알고리즘을 적용함으로써 예측 모델을 구축하고 각 모형들을 비교, 평가하여 가장 우수한 성능을 보이는 모형을 제안하고자 한다.

1.2 연구의 방법 및 구성

본 연구에서는 딥러닝 알고리즘 2개와 기계학습 알고리즘 3개를 이용하여 환율 예측을 시도하였다. 딥러닝 알고리즘은 순환신경망의 LSTM, 합성공 신경망(CNN)을 사용하였고, 기계학습 알고리즘에서는 XGBoost, SVM, Random Forest를 사용하였다. 입력변수로는 거시경제변수 요소인 KOSPI 지수, 경제성장률, 금리, 소비자물가지수, 다우존스 및 나스닥지수, WTI, 자본지수, 경상수지(상품수지, 서비스수지)로 선정하였다.

본 논문은 5개의 장으로 구성되어 있다. 1장에서는 연구의 배경 및 목적에 대하여 서술하고 2장에서는 구축하고자 하는 예측모형을 이해

하기 위해 딥러닝, 머신러닝 알고리즘에 대한 이론을 살펴보고, 환율에 영향을 주는 거시 경제 변수의 선정에 대한 당위성을 부여하기 위해 환율 결정 이론을 살펴본다. 3장에서는 5가지의 알고리즘을 이용하는 연구 모형 구축에 대해 서술하고 4장에서는 구축된 모형의 결과들의 예측 결과를 보여주도록 한다. 이후 5장에서는 이 연구의 결론과 향후 보완점 및 후행 과제를 서술토록 한다.

II. 연구의 이론적 배경

2.1 인공신경망을 이용한 시계열 지표 예측 연구

김재현(2002)은 인공신경망 모형을 구축하여 원-달러 환율 예측을 시도하였고, 이를 기존의 ARIMA 예측 모형과 비교 평가 하였다. 환율 예측과 환율 변동 방향의 예측에 있어서 기존 ARIMA 모형보다 인공신경망 모형이 더 우수한 예측 결과를 보여주었다[1].

김호현(2017)은 LSTM/GRU 모형에 Xavier 가중치 초기화 기법과 적합을 방지하기 위해 Dropout 비율을 조정하여 적용한 예측 모형을 구축하였다. 기존의 ARIMA 모형보다 우수한 예측력을 보여주었다[2].

강민영(2016)은 이미지,음성 인식 분야에서 높은 성능을 보여주는 합성곱 신경망(CNN)모형을 구축하여 기존 인공신경망 모형인 다층 퍼셉트론(MLP) 모형과 성능을 비교하였다. 그 결과 CNN모형의 예측력이 기존 인공신경망 모형보다 높은 예측 정확도를 보여주었다[3].

한나영(2011)은 인공신경망 모형을 활용해서 KOSPI 지수의 예측을 시도했다. 구축된 모형과 시계열분석, 다중회귀분석을 통해 예측성과를 비교하였다. 다중회귀분석의 경우에는 주가의 비선형적인 특성 때문에 정확한 주가예측이 이루어지지 않아 제외했고, 시계열분석의 이동평균법과 지수평활법으로 인공신경망 모델과 비교를 하였다. 예측성과는 전통적으로 인공신경망 모형의 성과를 측정하는데 쓰이던 RSME(Root Square Mean Error)로 비교하였는데, 그 결과 인공신경망이 시계열분석에 비해 더 높은 성과를 보였다[4].

이지훈(2017)은 기존의 MLP모델과 CNN 예측 모형, RNN 예측 모

형을 구축하여 3개의 예측 모형을 비교 평가하였다. 주가 예측에 대한 평균 정확도는 RNN이 가장 높게 나타남을 확인하였다[5].

배성완 · 유정석(2018)은 시계열 분석과 기계학습 모형 구축을 통해 부동산 가격지수에 대한 예측 정확도를 평가하였다. 그 결과 기계학습 모형은 상대적으로 시계열 분석보다 예측 정확도가 높게 나타났음을 확인할 수 있었다[6].

이태형(2019)은 아파트 가격지수만을 변수로 하는 일변량과 6개의 거시경제지표를 추가한 다변량으로 나누어 예측력을 비교하였다. 그 결과 일변량과 6개의 거시경제지표를 추가한 다변량으로 나누어 예측력을 비교하였다. 그 결과 일변량 분석의 경우 RNN과 LSTM 모형이 ARIMA 모형보다 우수한 결과를 보여주었고 다변량 분석의 경우도 LSTM의 예측력 다른 계량경제 모형보다 우수하다는 것을 확인시켜주었다. 특히 인공신경망의 예측력은 상대적으로 변동성이 크고 비선형성이 높은 자료에 대해 유용하다는 것을 확인하였다[7].

송대섭(2015)은 당일대비 익일 KOSPI지수 종가 등락여부를 예측하기 위해 딥러닝, SVM, LS-SVM 등의 기법을 이용하여 적중률을 비교하였다. 분석에 사용된 자료는 2007년~2009년 11월까지의 국내, 미국 주가지수 이고 각 기법에 앙상블을 접목시켜 추가로 모형을 구축 후 비교 하였다. 분석 결과 딥러닝 모형이 SVM, LS-SVM 모형에 비해 적중률이 높다는 것을 확인할 수 있었다[8].

2.2 거시경제 변수 분석

환율이란 개념은 자기나라 통화와 외국 통화간의 교환 비율을 나타낸다. 더 자세히 말하면 환율이 자유롭게 결정되는 변동환율제도하에서 환율은 외환시장의 수요와 공급에 의해 결정된다는 것이다. 예를 들어 원-달러 시장에서 미 달러화의 공급이 수요보다 많다면 원화 대비 미 달러화 가치가 하락하여 원-달러 환율이 하락한다[9]. 환율은 국민경제에 많은 영향을 미치는 가격 변수이다. 일반적으로 환율이 상승하면 수출 증가, 수입 감소를 통해 경상수지가 개선되고 생산 및 고용이 증가한가. 또한 수입물품가격이 상승하여 제조업 생산비용이 증가되고 물가가 상승한다. 외화대비 원화 가치가 하락하여 기업의 외채상환 부담이 가중된다. 외국인 관광객은 유리해지고, 내국인의 해외여행의 부담은 증가되는 효과가 있다. 환율을 결정하는 가장 근본적인 요인으로는 해당국가와 상대국의 물가수준 변동을 들 수 있다. 통화가치는 재화, 서비스, 자본 등에 대한 구매력의 척도이므로 결국 환율은 상대 물가수준으로 가늠되는 상대적 구매력에 의해 결정되기 때문이다.

장기적으로 환율에 영향을 미치는 또 다른 요소로 생산성의 변화를 들 수 있다. 예를 들어 한나라의 생산성이 다른 나라보다 더 빠른 속도로 향상(악화)될 경우 자국통화는 절상(절하)된다. 이는 생산성이 개선될 경우 재화생산에 필요한 비용이 절감되어 더 싼 값에 재화를 공급할 수 있게 되어 물가가 하락하고 통화가치는 올라가게 된다.

중기적 관점에서 보면 환율에 영향을 미치는 요인으로 대외거래, 거

시경제정책 등을 들 수 있다. 대외거래 결과 국제수지가 흑자를 보이면 외환의 공급이 늘어나므로 환율은 하락하고, 국제수지가 적자를 보여 외환의 초과수요가 지속되면 환율은 상승하게 된다. 통화정책 등 거시경제정책도 환율에 영향을 미친다. 통화정책을 긴축적으로 운용하면 통화 공급이 감소하여 외국의 통화량에 변화가 없다면 원화의 상대적인 공급이 줄어들어 환율이 하락(원화절상)한다.

단기적으로 환율은 외환시장 참가자들의 기대나 주변국의 환율 변동, 각종 뉴스 등에 따라 영향을 받는다. 첫째, 시장참가자들의 환율에 대한 기대가 변하게 되면 자기실현적(self-fulfilling)인 거래에 의해 실제 환율의 변동이 초래된다. 예를 들어 대부분의 시장참가자가 환율상승을 예상할 경우 환율이 오르기 전에 미리 외환을 매입하면 이익을 볼 수 있으므로 외환에 대한 수요가 증가하게 되어 실제 환율이 상승하게 된다.

둘째, 주요 교역 상대국의 환율 변동은 자국 통화가치에 많은 영향을 주게 된다. 예를 들어 수출경쟁관계에 있는 나라의 통화가 절하될 경우 자국의 수출경쟁력 약화로 인해 외환공급이 감소할 것이라는 시장기대가 형성되어 자국의 통화도 절하된다.

셋째, 각종 뉴스도 시장참가자들의 기대변화를 통해 단기 환율변동에 영향을 미치게 된다. 일례로 2010년 5월 천안함 침몰조사 결과가 발표되고 지정학적 위험이 부각되자 원/달러 환율이 일시적으로 큰 폭 상승하였다.

넷째, 은행의 외환포지션 변동도 환율에 영향을 미친다. 은행의 외환포지션(외화자산-외화부채)이 매도초과(외화부채> 외화자산) 혹은 매입초과(외화부채 < 외화자산)의 한 방향으로 크게 노출될 경우 포지션

조정을 위한 거래가 일어나고 그 결과 환율이 변동하게 된다. 예를 들어 은행의 선물환 포지션이 큰 폭의 매도초과를 보일 경우 환율변동에 따른 위험에 노출되지 않기 위해 현물환을 매입함으로써 환율이 상승하게 된다[10].

아래 [표 1-1]은 원화 환율에 영향을 미치는 거시 경제 변수들에 대해 정리 한 것이다. 이 변수들은 본 연구에서 각 모델의 구축의 입력 변수가 된다.

[표1-1] 환율에 영향을 미치는 거시 경제 변수

KOSPI 지수	환율이 상승할 때 증권시장에 가장 빠르게 반영되는 요소는 '외국인 투자자들의 동향' 이다. 외국인 투자자들은 환율의 움직임에 민감하다. 정확히는 국내 주식 시장에서 투자하고 있는 외국인 투자자들이다. 환율이 상승하면 보유하고 있던 시세차익을 통해 얻을 수 있는 양이 줄어들기 때문에 주식시장에서 외국인 투자자들의 매도세에 의해 투자 자본이 유출되면서 주가지수가 하락하는 경향이 있음. 이와 반대로 환율이 하락할 것으로 전망되어지면 외국인 투자 자본의 유입으로 인해 주가지수가 상승하는 경향이 있다.[11]
대한민국 경제성장률	한 국가의 경제성장률이 다른 국가에 비해 높으면 더 많은 수익을 얻기 위해 투자자금 유입이 활발해지고 안정적으로 많은 자금을 이용하여 더 높은 경제성장률을 기록할 수 있다. 또한 안정적으로 경제성장률을 유지하는 국가는 변동성이 확대되었을 경우 투자자금은 안전자산으로 판단된 국가로 이동한다. 결과적으로 자금의 유입이 높아진다는 것은 자국 화폐에 대한 수요가 높아지는 것으로 자국 화폐가치는 수요와 공급의 법칙에 따라 결정된다. 하지만 화폐가치의 상승은 수출경쟁력을

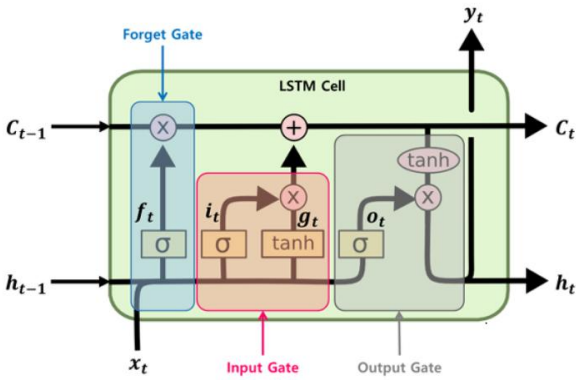
	낮추고 수입은 증가하여 다시 경제성장률을 하락시키는 순환구조를 가지고 있다.[12]
미국 중앙은행 정책금리	미국 연준 에서 결정하는 연방기금금리를 의미한다. 연준 에서 미국의 금리를 인상시키면 각국에 분포되어 있던 달러들이 미국 내 자본 시장으로 몰리게 되어 이는 곧 달러가 유출된 나라의 달러 공급의 부족을 초래하게 된다. 이는 원-달러 환율의 상승을 초래한다.
소비자 물가지수	수입 물가는 국내 소비자물가에 직접적인 영향을 끼치게 되며, 수입 물가는 환율에 직접적인 영향을 받게 된다. 환율 하락 시, 수입 제품의 가격이 변동하지 않아도 화폐의 가치 상승으로 인해 상품의 가격도 오르게 되고, 이에 따라 소비자 물가도 상승한다. 이처럼 환율 변동은 일부 수입물가에만 영향을 주는 것이 아니라 수입하는 모든 것의 가격에 영향을 미치게 된다. 결과적으로 소비자 물가지수에도 영향을 주게 되며 국가에서 중요한 물가상승률에 중요한 변수로 작용하게 된다. 즉, 수입 물가는 소비자물가에 직접적인 영향을 주고 수입 물가는 환율에 직접적으로 영향을 받는다. 환율이 상승하면 외국에서 들여오는 물건 가격이 변하지 않는다 하더라도 원화로 환산한 가격은 오르게 되고 수입물가지수는 오른다[13].
다우존스& 나스닥지수	미국의 시장 금리가 오르게 되면 각국에 흩어진 자본들이 자연스럽게 미국 시장으로 유입이 된다. 국내 자본시장의 자본도 달러도 환전되어 미국 시장으로 자본이 유출된다. 미국 내 유입된 자본은 미국 시장의 지수를 상승시키고 이는 국내지수의 하락 및 원-달러 환율의 상승을 초래한다.
WTI (국제유가지수)	일반적으로 미 달러화 가치와 국제 유가는 반대 방향으로 움직이는 경향이 있다. 이런 현상은 대부분 원유 거래가 달러화로 이뤄지고 있다는 점과 투자 시장에서 달러화와 원유가 대체 관계를 보이고 있기 때문에 나타난

	다. 유가와 원 달러 환율간의 음의 상관관계는 세계경기 호황 때 유가가 상승하고 수출호조로 원화 강세가 되는 현상으로 발생한다고 유추된다[14].
자본수지	자본수지는 각종 투자를 통해 들어오고 나가는 외화를 계산한 것으로 보면 된다. 외국인이 우리나라에 투자하게 되면 달러를 들고 오는 것이므로 외화가 유입된다. 반대로 외국인의 투자금액이 감소하게 되면 외화가 유출되는데. 이때 자본수지가 움직이게 된다. 마찬가지로 자국민이 해외에 투자하게 되어도 외화가 유출된다. 경상수지가 흑자로 돌아서게 되면 우리나라의 입장에서는 수입보다 수출이 많다는 의미이므로 경제적으로 좋은 상황이다. 그러나 주의해야할 점은 이렇게 되면 상대적으로 외화의 양이 많아지면서 외화의 가치가 떨어지고 환율이 하락한다. 경상수지가 계속해서 흑자가 나고 있다면, 해외에 많은 공장을 건설하고 부동산 자산을 사는 방법 등으로 국내에 쌓인 외화를 계속해서 소모시켜야 한다. 결과적으로 자본수지 자체는 적자가 나지만, 경상수지 흑자로 인해 환율이 하락하는 것을 막으면서 지속적인 수출 증대를 계속해서 유도할 수 있다. 즉, 자본수지 적자 = 외화 감소 = 원화 약세 = 환율 상승을 초래한다고 볼 수 있다.
경상수지 (상품수지)	경상 수지가 흑자이면 외화가 증가하여 원화가 강세흐름이 나타난다. 이는 환율 하락을 초래하며 수출 경쟁력이 떨어진다. 예를 들어 우리나라가 수출을 많이 하면 달러를 많이 벌어들였으므로 달러공급이 늘어나 달러가치는 하락하고 원화가치는 상승하여 환율이 강세를 보인다(하락한다).
경상수지 (서비스수지)	

2.2 모델 구축을 위한 알고리즘 분석

2.2.1 LSTM

순환신경망은 순차적이거나 길이가 가변적인 데이터에 적합한 딥러닝 모델이다. RNN은 재귀를 통한 정보전이 및 전파가 하나의 레이어로 제어되는 반면 LSTM은 [그림 2-1]과 같이 소실(Forget), 입력(Input), 갱신(Update), 출력(Output) 등 4가지 레이어가 서로 특별한 방식으로 정보를 주고 받을수 있게 되어 있다[15].



[그림 2-1] 4 Layers of LSTM

LSTM의 핵심은 셀상태(cell state)이다. 셀 상태는 LSTM 셀에서는 상태(state)가 두 개의 벡터 h_t 와 c_t로 나누어 진다는 것을 알 수 있다. h_t를 단기 상태(short-term state), c_t를 장기 상태(long-term state)라고 볼 수 있다. LSTM의 핵심은 네트워크가 장기 상태(c_t)에서 기억할 부분, 삭제할 부분, 그리고 읽어 들일 부분을 학습하는 것이다. 즉 셀 상태 c_t는 이전(t-n)부터 전달되어 오는 정보를 컨베이어 벨

트와 같은 역할을 수행한다. 이전 (t−n)이 증가하게 되더라도 "긴 기간의 의존성(long−term dependencies)"의 문제를 최소화하여 오차의 전파가 원활하게 이루어진다.

LSTM의 첫 단계로는 셀 상태로부터 어떤 정보를 버릴 것인지를 정하는 것으로, 시그모이드 함수에 의해 결정된다. 그래서 이 단계를 소실 레이어(Forget Layer)라고 부른다. 이 단계에서는 h_{t-1} 과 x_t를 받아서 0과 1사이의 값을 c_{t-1}에 보낸다. 그 값이 1이면 정보를 보존하고, 0이면 정보를 삭제한다. 소실 레이어는 식(1)에 의해 결정된다.

$$f_t = \sigma(W_f \cdot \ [h_{t-1}, x_t] + b_f) \tag{1}$$

다음 단계는 앞으로 들어오는 정보 중 셀 상태에 저장할 정보를 정하는 것이다. 시그모이드 함수로 구성된 입력 레이어는 어떤 값을 업데이트 할지 정한다. 그 다음 탄젠트 함수가 새로운 후보 값들인 $\widetilde{C}_t$라는 벡터를 만들고, 셀 상태에 더 할 준비를 한다. 이렇게 두 단계에서 나온 정보를 합쳐서 상태를 업데이트할 준비를 마친다. 입력 레이어는 식(2)에 의해 결정된다.

$$\begin{aligned} i_t &= \sigma(W_i \cdot \ [h_{t-1}, x_t] + b_i) \\ \widetilde{C}_t &= \tanh(W_C \cdot \ [h_{t-1}, x_t] + b_C) \end{aligned} \tag{2}$$

이후 과거 셀 상태인 C_{t-1}를 업데이트해서 새로운 셀 상태인 C_t를 만들 것이다. 이미 이전단계에서 벡터를 통해 어떤 값을 얼마나 업데이트

해야할지 정해놨으므로 실행만 하면 된다. 우선 이전 상태에 f_t를 곱해서 가장 첫 단계에서 소실하기로 정했던 것들을 실행한다. 그리고 난 후 $i_t * \widetilde{C}_t$를 더한다. 식(3)에 의해 결정된다.

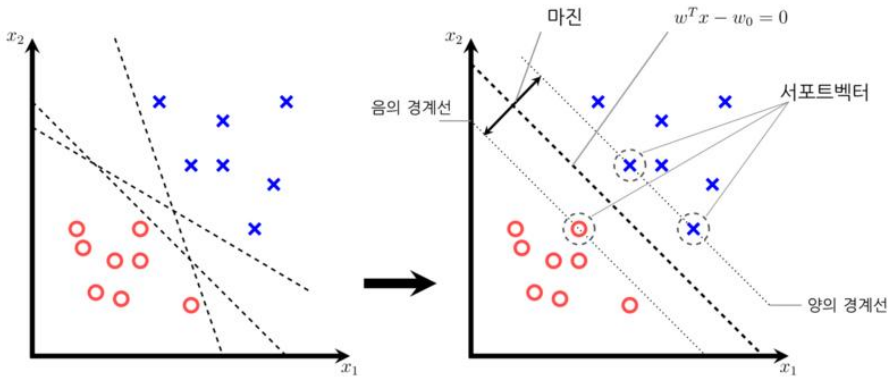
$$C_t = f_t * C_{t-1} + i_t * \widetilde{C}_t \tag{3}$$

마지막 단계는 출력 값을 정하는 것이다. 이 값은 셀 상태를 바탕으로 필터링 된 값이다. 먼저, 시그모이드 함수에 입력 데이터를 넣어 셀 상태의 어느 부분을 출력 값으로 내보낼 지를 정한다. 그 후 셀 상태를 탄젠트 함수로 계산하여 -1과 1사이의 값을 받은 후에 이전 단계에서 계산한 시그모이드 함수의 계산 값과 곱해준다. 이는 우리가 출력 값으로 보내고자 하는 부분만 내보낼 수 있게 해준다. 식(4)에 의해 결정된다.

$$\begin{aligned} o_t &= \sigma(W_o[h_{t-1}, x_t] + b_o) \\ h_t &= o_t * \tanh(C_t) \end{aligned} \tag{4}$$

2.2.2 SVM

SVM 모형은 선형이나 비선형 분류, 회귀, 이상치 탐색, 패턴인식 등 다목적 머신러닝 모델이다. SVM의 기본 개념은 집단사이의 마진(Margin)이 최대가 되도록 하는 초평면(Hyperplane)을 이용하는 것이다. 이때 초평면과 가장 가까운 벡터(Vector)를 서포트 벡터라 한다.



[그림 2-2] SVM 모델의 이해

위의 그림[2-2]를 보면 마진이란 선과 가장 가까운 양 옆 데이터와의 거리이다. SVM에서는 마진을 최대화하는 방향으로 최적화를 진행한다. 마진이 클수록 일반화 오차가 낮아지는 경향이 있고, 작을수록 모델은 과적합(overfitting)되기 쉽다. 따라서 마진이 클수록 좋은데 이때 마진에 걸쳐지는 데이터의 일부를 서포트 벡터라 한다. 즉 SVM은 마진을 최대화하는 분류 경계면을 찾는 기법이라고 정의할 수 있다.

본 논문에서는 손실함수를 도입하여 회귀식을 구성하는 SVR 모형을 구축한다. 일반 선형회귀 모델에서는 모형이 과적합 하게 되면 회귀 계수 W의 크기도 증가하기 때문에 추가적으로 제약을 부여하여 회귀계수의 크기가 너무 커지지 않도록 penalty식을 부여하여 계수의 크기를 제한하는 정규화 방법을 적용한다. SVR 손실함수 수식에 담긴 의미를 해석해보면, 회귀계수 크기를 작게 만들어 회귀식을 평평하게 만들되, 실제 값과 추정 값의 차이를 작도록 고려하는 선을 찾는 것이라 할 수

있다. 식(5)에 의해 결정된다.

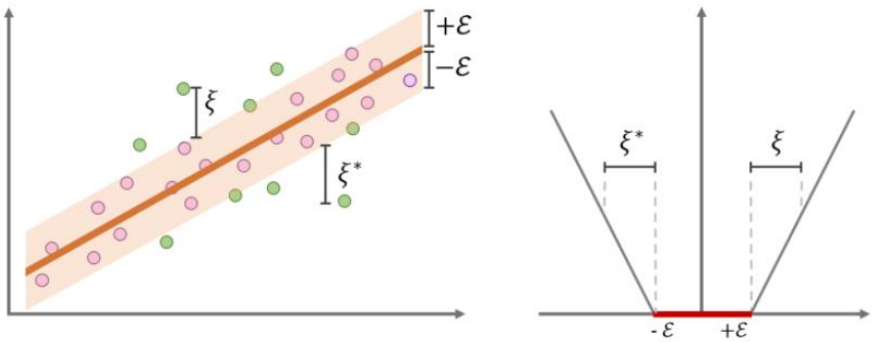
$$L_{SVR} = \min \overbrace{\| w \|^2}^{Robustness} + \underbrace{\lambda(\frac{1}{2}\sum_{i=1}^{n}(y_i - f(x_i))^2)}_{loss\ fucntion} \qquad (5)$$

식(5)에 ϵ −intensive 함수를 사용한 식(6)으로 표현한다.

$$L_{SVR} = \min \overbrace{\frac{1}{2}\| w \|^2}^{Robustness} + \underbrace{C\sum_{i=1}^{n}(\xi_i + \xi_i^*)}_{loss\ fucntion} \qquad \text{식(6)}$$

$$(w^T x_i + b) - y_i \leq \epsilon + \xi_i$$
$$yi - (w^T x_i + b) \leq \epsilon + \xi_i^*$$
$$\xi_i, \xi_i^* \geq 0$$

아래 [그림2-3]에서 ϵ는 회귀식 위아래 사용자가 지정한 값 ∝ 허용하는 노이즈 정도를 의미하고, ξ는 튜브 밖에 벗어난 거리(회귀식 위쪽)이고 ξ^*는 튜브 밖에 벗어난 거리(회귀식 아래쪽)을 의미한다.



[그림 2-3] ϵ -intensive 손실 함수

[그림2-3]의 오른쪽 그림에서처럼 튜브 내에 실제 값이 있다면 예측 값과 차이가 있더라도 용인해주기 위해 penalty를 0으로 주고, 튜브 밖에 실제 값이 있다면 C의 배율로 penalty를 부여한다. SVR은 데이터에 노이즈가 있다고 가정하며, 이러한 점을 고려하여 노이즈가 있는 실제 값을 완벽히 추정 하는 것 을 추구하지 않는다. 따라서 적정 범위(2ϵ) 내에서는 실제 값과 예측 값의 차이를 허용한다.

SVR도 커널함수를 사용하여 학습데이터를 특징 공간의 점으로 변화시킨 다음 특징공간에서 학습을 수행하게 된다. 간편하게 고차원공간으로 매핑하여 식(7)과 같이 비선형적인 회귀식을 도출할 수 있다. 이렇게 커널 트릭으로 활용할 수 있는 커널 함수는 다양하다. 대표적으로 RGB, Linear, Polynomial, Sigmoid 등이 있다.

$$\sum_{i=1}^{n}(\alpha_i^* - \alpha_i)\Phi(x_i) \quad \Rightarrow \quad f(x) = \sum_{i=i}^{n}(\alpha_i^* - \alpha_i)K(x_i x_j) + b \qquad \text{식(7)}$$

2.2.3 Random Forest

랜덤 포레스트를 위해서는 먼저 앙상블 기법에 대한 이해가 필요하다. 앙상블(ensemble)은 여러 모델들을 활용해 종합하여 예측을 하는 기법이다. 적당한 예측력(성능)을 가진 모델을 한 개만 활용하는 것보다 여러 모델을 가지고 종합해서 결정하면 더 나은 의사 결정을 할 수 있다는 개념이다. 본 논문에서는 앙상블 기법으로 배깅을 적용한다.

배깅(bagging)은 훈련 세트에서 중복을 허용하여 샘플링 하는 방식이다. 쉽게 말하면, 하나의 훈련 데이터를 가지고 여러 개의 훈련 데이터를 만들고, 이를 가지고 여러 모델을 만들어서 모델들을 종합하는 방식이다. 배깅에서는 훈련 세트에서 중복 허용을 통해 여러 샘플링 세트를 만들고, 이 샘플링들을 가지고 여러 개의 예측기(모델)를 훈련한다. 그 다음 예측할 때 이들 모델 예측의 평균을 계산해서 최종 예측을 하는 방식이다[16].

결정 트리 모델에 배깅(bagging) 기법을 적용한 앙상블 모델을, 특별하게 랜덤포레스트(Random Forest) 모델이라고 한다. 하나의 원본 데이터에서 랜덤하게 데이터들을 만들어 여러 개의 학습된 결정 트리 모델을 만들고 이를 종합하는 방식이다. 보통 랜덤 포레스트는 서로 다른 수십 개 혹은 수백 개의 결정 트리를 만들어서 사용한다. 트리를 여러 개 만들어서 숲을 만든다는 뜻이다.

랜덤 포레스트에서는 배깅 기법을 통해, 우선 훈련 데이터에서 중복을 허용한 여러 데이터 샘플을 만들어낸다. 이 때, 분산성을 더 키우기 위해 피처(독립변수)의 개수를 랜덤하게 줄인 데이터를 각각의 결정 트리 모델에 사용하는 기법도 적용하고 있다. 랜덤 포레스트의 가장 큰 장점은 앙상블 기법을 통해서 과적합(overfitting)문제를 감소시키고

또한 예측력이 올라간다는 것이다. 랜덤 포레스트는, 전 기간 데이터만 가지고 학습하는 것이 아니라 몇몇 시계열 구간과 몇몇 주요 변수들을 뺀 샘플들로 수백 개의 결정 트리를 만들어 종합하는 방식이다. 이를 통해 일부 사건에 과대 적합 되지 않은 범용적인 판단 모델을 더 잘 만들게 된다.

2.2.4 XGboost

XGBoost 모형은 약한 분류기를 순차적으로 개선해나감으로써 보다 강력한 분류기를 생성하는 트리 모형에 그래디언트 부스팅(gradient boosting)기법을 적용한 앙상블 알고리즘에서 병렬 학습이 지원되도록 구현한 라이브러리가 XGBoost이다[17]. 이 모형은 교차 검증(Cross validation)을 지원하고, 표준 GBM 경우 과대 적합 규제기능이 없으나 XGBoost는 자체에 과대 적합 규제 기능으로 강한 내구성 지닌다. XGBoost의 리프 노드는 가중치에 따라 임의의 상수 값을 가진다. 배깅 방식 을 사용하는 랜덤포레스트에서는 단순 평균 값을 비교하는 반면 XGboost는 손실함수와 트리 복잡도 함수로 구성 된 목적함수로 오류를 최소화하고 트리의 복잡도를 최소화하는 최적의 트리 조합을 찾아낸다[18].

아래 식(6)은 결정 트리에서의 앙상블 모형을 표현한 식이다. K는 결정트리의 개수, F는 모든 경우의 집합을 의미한다. f_k는 F 공간 안에서 k번째 결정트리이다.

$$y_i = \sum_{k=1}^{K} f_k(x_i), f_k \in F. \qquad \text{식(6)}$$

식(7)은 앙상블 모형에서 과적합 규제 기능을 추가한 XGBoost 모형이다. l은 모델의 예측 능력을 측정하는 손실 함수이고 Ω은 모델의 복잡도를 측정하는 정규화 방법이다.

$$Obj = \underbrace{\sum_{k=1}^{n} l(y_i, \widehat{y_i})}_{Training\ Loss} + \overbrace{\sum_{k=1}^{K} \Omega(f_k)}^{Regularization} \qquad \text{식(7)}$$

식(8)은 정규화식을 자세히 표현한 식이다. γT는 결정 트리의 노드 개수를 의미한다. 그 뒤의 이어진 수식은 노드의 점수를 의미한다.

$$\Omega(f) = \gamma T + \frac{1}{2}\lambda \| w \|^2 \qquad \text{식(8)}$$

XGBoost는 위와 식(7)의 목적함수를 $\Omega(f_t)$를 조절하면서 정규화 항과 손실함수를 최소화하여 오차를 줄여간다.

식(9)는 테일러급수 전개함으로 1차 미분과 2차 미분 값을 제공하여 오차를 정밀하게 분석하고 개별 리프의 최적화 가중치와 개별 트리의 성능을 빠르게 측정할 수 있다.

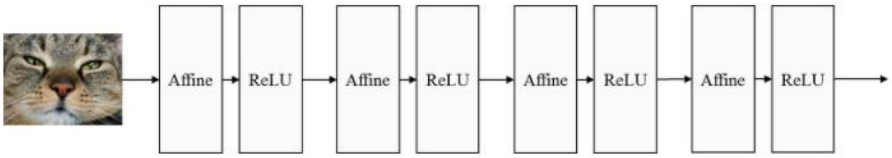
$$Obj = \sum_{i}^{n} [g_i f_i(x_i) + \frac{1}{2} h_i f_t^2(x_i)] + \Omega(f_t) \qquad \text{식(9)}$$

XGBoost는 각각의 트리 부스팅 단계 이후 새로 추가된 가중치를 계수로 조정하여 분산을 줄이는 shrinkage 기법으로 과적합을 방지한다. 그리고 RandomForest에서 사용되는 column sub-sampling을 적용하여 과적합을 방지하고 그 뿐만 아니라 병렬 처리 속도를 높인다[17].

2.2.5 CNN

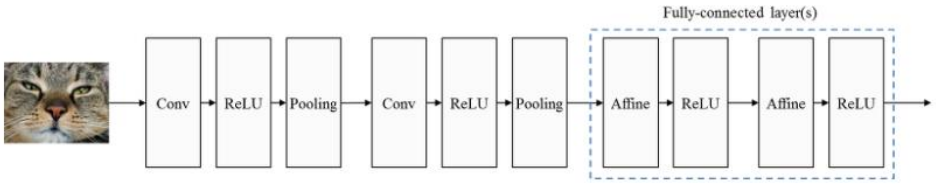
합성곱 신경망(CNN, Convolutional Neural Network)은 사물을 인식할 때 시각적 정보를 바탕으로 사물에 대한 특징을 추출한다는 것에 착안하여 개발된 알고리즘이다. 즉 3차원 데이터의 공간적 정보를 유지한채 다음 레이어로 정보 전달이 가능한 합성곱 연산을 사용하는 인공신경망의 한 종류이다[19].

기존의 인공신경망(Artificial Neural Network)구조는 인접하는 계층의 모든 뉴런이 결합되어 있는 완전연결(Fully Connected)로서, Affine 계층으로 구성되어 있다.



[그림 2-3] ANN 구조

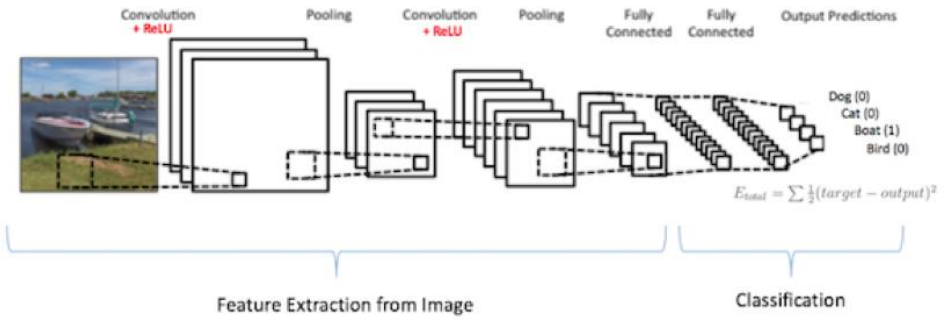
[그림 2-3]에서 이전 계층의 모든 뉴런과 결합된 형태의 Layer를 전결합 레이어라고 한다.



[그림 2-4] CNN 구조

[그림2-4]의 CNN에서는 합성곱 계층(Convolutional Layer), 풀링 계층(Pooling Layer)를 활성화 함수 앞뒤에 배치하게 된다. 이후 출력하게 되는 Layer에 도달하면 다시 전결합 레이어의 구조로서 결과 값을 보여준다.

전 결합 레이어는 1차원 데이터만 입력 받을 수 있기 때문에, 3차원 데이터를 평탄화해서 입력한다. 이 단계에서 3차원 데이터의 공간적 정보가 소실된다는 문제가 발생한다. 결과적으로 공간 정보 유실로 인한 정보 부족으로 인공 신경망이 특징을 추출 및 학습이 비효율적이고 정확도를 높이는데 한계가 있다. 반면 합성곱 계층은 형상을 유지하기 때문에 공간적 정보를 유지할 수 있다.

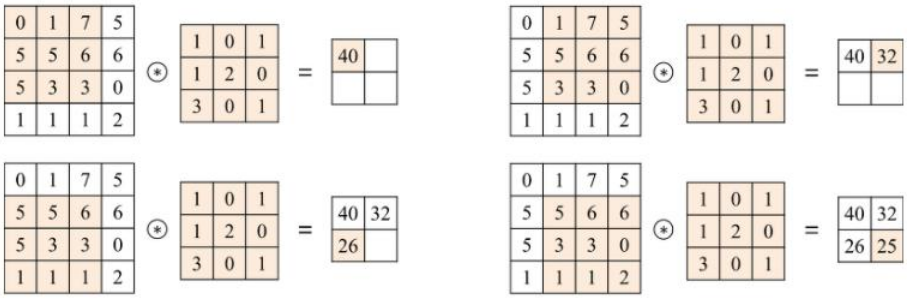


[그림 2-5] CNN 구조2

CNN은 [그림2-5]와 같이 특징을 추출하는 부분과 클래스를 분류하는

부분으로 나눌 수 있다. 추출 영역은 합성곱 레이어와 풀링 레이어를 여러 겹 쌓는 형태로 구성된다. 합성곱 레이어는 입력 데이터에 필터를 적용 후 활성화 함수를 반영하는 필수 요소이다. 합성곱 레이어 다음에 위치하는 Pooling Layer는 선택적인 레이어다. CNN 마지막 부분에는 이미지 분류를 위한 Fully Connected 레이어가 추가된다. 이미지의 특징을 추출하는 부분과 이미지를 분류하는 부분 사이에 이미지 형태의 데이터를 배열 형태로 만드는 Flatten 레이어가 위치한다.

합성곱 연산은 커널(kernel) 또는 필터(filter)라는 n×m 크기의 행렬로 높이(height)×너비(width) 크기의 이미지를 처음부터 끝까지 겹치며 훑으면서 n×m크기의 겹쳐지는 부분의 각 이미지와 커널의 원소의 값을 곱해서 모두 더한 값을 출력으로 하는 것을 말한다. 이때, 이미지의 가장 왼쪽 위부터 가장 오른쪽까지 순차적으로 탐색한다[20].



[그림 2-6] 합성곱 연산 순서

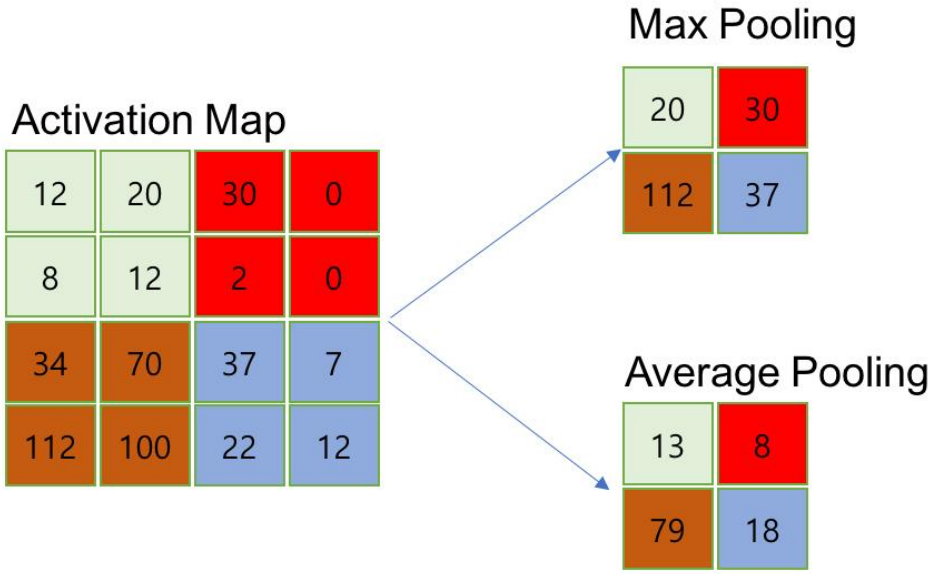
[그림2-6]은 입력 데이터는 (4,4)이고 필터는 (3,3)이다. 필터가 합성곱 계층의 가중치에 해당한다. 필터의 이동양을 의미하는 스트라이드(stride)를 설정한 후 스트라이드 값의 간격으로 이동해가면서 계산하게 된다. [그림 2-6]에서 입력 이미지에 대해 합성곱 연산을 수행하면

출력 이미지의 크기는 입력 이미지의 크기보다 작아지게 된다. 그러므로 합성곱 계층을 거치면서 이미지의 크기는 작아지게 되고, 이미지의 가장자리에 위치한 픽셀들의 정보는 점차 사라지게 된다. 이러한 문제점을 해결하기 위해 이용되는 것이 패딩(padding)이다. 입력의 가장자리에 지정된 개수의 폭만큼 행과 열을 추가해주는 것을 말한다. 좀 더 쉽게 설명하면 지정된 개수의 폭만큼 테두리를 추가하고 주로 값을 0으로 채우는 제로 패딩(zero padding)을 사용한다.

패딩과 스트라이드를 적용하고 입력의 크기와 커널의 크기만 알면 합성곱 연산의 결과인 특성맵(Feature Map)의 크기를 계산할 수 있다 식은 아래 식(10)에서 확인할 수 있다.

$$O_h = floor(\frac{I_w - K_h + 2P}{S} + 1) \qquad \text{식(10)}$$
$$O_w = floor(\frac{I_w - K_h + 2P}{S} + 1)$$

일반적으로 합성곱 레이어 다음에 풀링 레이어를 추가하는 것이 일반적이다. 풀링 레이어에서는 특성맵을 다운 샘플링하여 특성 맵의 크기를 줄이는 풀링 연산이 진행 된다. 풀링 연산에는 일반적으로 최대 풀링(max pooling)과 평균 풀링(average pooling)이 사용된다[20]. 풀링을 사용하면, 특성 맵의 크기가 줄어들어 특성 맵의 가중치의 개수를 줄여준다. 아래 [그림 2-7]은 풀링 예제이다.


Activation Map
12
20
30
0
8
12
2
0
34
70
37
7
112
100
22
12
Max Pooling
20
30
112
37
Average Pooling
13
8
79
18

[그림 2-7] 풀링 예제

Ⅲ. 딥러닝/머신러닝 알고리즘을 이용한 예측 모델 구축

3.1 본 연구에 대한 소개

인공신경망 기반 예측 모형은 많은 비선형 시계열의 모형 화에 매우 유용한 것으로 평가 된다. 따라서 금융시계열과 같이 비선형의 특징을 갖는 데이터에 적합하다. 이에 많은 시계열 데이터 연구에 딥러닝 및 머신러닝 기법을 적용한 연구가 활발히 진행되고 있다.

본 연구의 목적은 2010년 이후의 환율자료를 이용하여 환율 예측 모형을 딥러닝의 LSTM 모형, CNN 모형 2가지와 기계 학습 모형의 XGboost, SVM, Random Forest 알고리즘 3가지를 사용하여 총 5가지의 예측 모델을 구축하여 환율을 예측한 후, 성능 지표와 예측정확도로 평가하여 가장 우수한 성능을 보이는 모델을 선정하는 것이다.

3.2 학습 데이터의 수집

본 연구에서 사용하는 데이터는 한국은행 경제 통계 시스템에서 제공하는 데이터를 사용했으며, 환율데이터는 2010년 1월 1일부터 2019년 12월 31일까지의 일별 환율 기록이다. 구축하게 될 각 모델의 입력변수로는 [표 1-1]에서 제안한 변수들로 한국은행 경제 통계 시스템에서 제공받아 구성했다. 하지만 데이터마다 일별, 월별, 년별로 데이터의 제공주기가 다르기에 실험데이터를 위해서 일별로 통합해야 했다.

[표 1-2]는 데이터 수집을 통해 만들어진 원본 데이터의 일부분을 보여준다.

날짜	환율	코스피	경제성장률	연방금리	물가	다우	나스닥	원유	자본수지한	자본수지미
2019-12-31	1157.8	0	0	1.625	115.45	28538.44	8972.6	61.06	-64.2	-29
2019-12-30	1160.9	2197.67	0	1.625	115.45	28538.44	8972.6	61.06	-64.2	-29
2019-12-27	1161.2	2204.21	0	1.625	115.45	28538.44	8972.6	61.06	-64.2	-29
2019-12-26	1163.7	2197.93	0	1.625	115.45	28538.44	8972.6	61.06	-64.2	-29
2019-12-24	1161.7	2190.08	0	1.625	115.45	28538.44	8972.6	61.06	-64.2	-29
2019-12-23	1162.6	2203.71	0	1.625	115.45	28538.44	8972.6	61.06	-64.2	-29
2019-12-20	1164.5	2204.18	0	1.625	115.45	28538.44	8972.6	61.06	-64.2	-29
2019-12-19	1165.3	2196.56	0	1.625	115.45	28538.44	8972.6	61.06	-64.2	-29
2019-12-18	1166.7	2194.76	0	1.625	115.45	28538.44	8972.6	61.06	-64.2	-29
2019-12-17	1172.9	2195.68	0	1.625	115.45	28538.44	8972.6	61.06	-64.2	-29
2019-12-16	1171.7	2168.15	0	1.625	115.45	28538.44	8972.6	61.06	-64.2	-29
2019-12-13	1188.6	2170.25	0	1.625	115.45	28538.44	8972.6	61.06	-64.2	-29
2019-12-12	1193.3	2137.35	0	1.625	115.45	28538.44	8972.6	61.06	-64.2	-29
2019-12-11	1192	2105.62	0	1.625	115.45	28538.44	8972.6	61.06	-64.2	-29
2019-12-10	1189.3	2098	0	1.625	115.45	28538.44	8972.6	61.06	-64.2	-29
2019-12-09	1189.7	2088.65	0	1.625	115.45	28538.44	8972.6	61.06	-64.2	-29
2019-12-06	1190.2	2081.85	0	1.625	115.45	28538.44	8972.6	61.06	-64.2	-29
2019-12-05	1193.7	2060.74	0	1.625	115.45	28538.44	8972.6	61.06	-64.2	-29
2019-12-04	1186.2	2068.89	0	1.625	115.45	28538.44	8972.6	61.06	-64.2	-29
2019-12-03	1180.4	2084.07	0	1.625	115.45	28538.44	8972.6	61.06	-64.2	-29
2019-12-02	1180.3	2091.92	0	1.625	115.45	28538.44	8972.6	61.06	-64.2	-29

[표 1-2] 원본 데이터의 예

3.3 데이터 전처리

여러 개의 시트로 구성된 한 개의 엑셀 파일을 가지고 시트에 담긴 데이터를 합쳐 실험 데이터를 구성하였다. 일별, 월별, 연도별 데이터를 일별 데이터 기준으로 모든 데이터를 병합하였다. 환율이 제공되지 않은 날짜들은 제거 하였고, 데이터 기간의 단위가 다르기 때문에 일별로 병합할 경우 결측이 생기는데 평균값, 중앙값, 최빈값으로 대체하였으나 성능에 차이가 미비하여 이를 0으로 대체하였다. 입력변수들 간에 단위를 동일하게 맞춰주기 위해서 최소·최대 정규화 기법을 활용하여

모든 변수들의 범위를 0과1사이의 실수 값으로 변환한다.

최소-최대 정규화 기법은 모델에 투입될 모든 데이터 중에서 가장 작은 값을 0, 가장 큰 값을 1로 두고, 나머지 값들은 비율을 맞춰서 모두 0과 1 사이의 값으로 조정하는 것이다. 따라서 X라는 값을 정규화를 시킨다면 X는 식(5)의 수식을 사용하는 것을 확인할 수 있다[11].

$$X = \frac{x - x_{\min}}{x_{\max} - x_{\min}} \qquad \text{식(5)}$$

최소 · 최대 정규화 기법에는 단점이 있다. 이상치(outlier)의 영향이 크다는 점이다. 1차원일 때 그 데이터가 이상치일 수 있지만 다차원일 때 그 데이터를 제거하거나 변경하는 것은 큰 오류를 가져올 수 있다. 예를 들어, 환율 값만 봤을 때는 환율이 엄청 크게 증가한 하루를 제거하는 것이 예측 모델을 만드는 데에 있어서 좋을 수는 있지만, 본 논문에서는 입력데이터로 환율 및 다른 변수들(거시경제 변수)도 포함되어 있어 이상치의 영향은 고려하지 않았다.

원본 데이터는 10년 1월 1일부터 19년 12월 31일까지의 총 3652개다. 정규화 기법을 통해 전처리 과정을 거친 후 데이터 세트(data set) 2476개를 만들어 학습 세트는 (training set, 1737개, 2010.1.1. ~ 2016.12.27.) 70%로 할당하고 20%를 검증 세트(valid set, 491개, 2016.12.28. ~ 2018.12.28.)로, 10%를 테스트 세트(test set, 248개, 2018.12.29. ~ 2019.12.31)로 전처리를 진행했다.

아래 [표 1-3]은 [표 1-2]의 원본데이터에 최소 · 최대 정규화 기법을 적용 한 결과이다. [표 1-3]의 거시 경제 변수 외에 Feature Engineering의 파생변수를 추가한다. 시계열 데이터의 분석에서 사용

되는 지정 범위내의 평균을 계산하는 이동 평균(Moving Average)를 활용해서 기준일 이전 7일간의 환율 평균값을 사용한다. 기준일 과거 일주일전 모든 환율 값들을 변수로 추가 한다. (기준일 1일전 환율, 2일전 환율.., 7일전 환율 값).

[표1-3] 정규화 기법 적용 후 데이터

날짜	환율	코스피	성장률	연방금리	물가지수	다우	나스닥	원유	자본수지한	자본수지미
2019-12-31	0.58947	0	0.130435	0.666667	0.966299	1	1	0.341676	0.111499883	0.345368917
2019-12-30	0.601742	0.845847	0.130435	0.666667	0.966299	1	1	0.341676	0.111499883	0.345368917
2019-12-27	0.60293	0.848364	0.130435	0.666667	0.966299	1	1	0.341676	0.111499883	0.345368917
2019-12-26	0.612827	0.845947	0.130435	0.666667	0.966299	1	1	0.341676	0.111499883	0.345368917
2019-12-24	0.604909	0.842925	0.130435	0.666667	0.966299	1	1	0.341676	0.111499883	0.345368917
2019-12-23	0.608472	0.848171	0.130435	0.666667	0.966299	1	1	0.341676	0.111499883	0.345368917
2019-12-20	0.615994	0.848352	0.130435	0.666667	0.966299	1	1	0.341676	0.111499883	0.345368917
2019-12-19	0.619161	0.845419	0.130435	0.666667	0.966299	1	1	0.341676	0.111499883	0.345368917
2019-12-18	0.624703	0.844727	0.130435	0.666667	0.966299	1	1	0.341676	0.111499883	0.345368917
2019-12-17	0.649248	0.845081	0.130435	0.666667	0.966299	1	1	0.341676	0.111499883	0.345368917
2019-12-16	0.644497	0.834485	0.130435	0.666667	0.966299	1	1	0.341676	0.111499883	0.345368917
2019-12-13	0.711401	0.835293	0.130435	0.666667	0.966299	1	1	0.341676	0.111499883	0.345368917
2019-12-12	0.730008	0.82263	0.130435	0.666667	0.966299	1	1	0.341676	0.111499883	0.345368917
2019-12-11	0.724861	0.810418	0.130435	0.666667	0.966299	1	1	0.341676	0.111499883	0.345368917
2019-12-10	0.714173	0.807485	0.130435	0.666667	0.966299	1	1	0.341676	0.111499883	0.345368917
2019-12-09	0.715756	0.803887	0.130435	0.666667	0.966299	1	1	0.341676	0.111499883	0.345368917
2019-12-06	0.717736	0.801269	0.130435	0.666667	0.966299	1	1	0.341676	0.111499883	0.345368917
2019-12-05	0.731591	0.793144	0.130435	0.666667	0.966299	1	1	0.341676	0.111499883	0.345368917
2019-12-04	0.7019	0.796281	0.130435	0.666667	0.966299	1	1	0.341676	0.111499883	0.345368917
2019-12-03	0.678939	0.802124	0.130435	0.666667	0.966299	1	1	0.341676	0.111499883	0.345368917
2019-12-02	0.678543	0.805145	0.130435	0.666667	0.966299	1	1	0.341676	0.111499883	0.345368917
2019-11-29	0.674584	0.803621	0.695652	0.666667	0.95061	0.974045	0.955251	0.268335	0.111499883	0.345368917
2019-11-28	0.66152	0.815414	0.695652	0.666667	0.95061	0.974045	0.955251	0.268335	0.111499883	0.345368917
2019-11-27	0.655186	0.818974	0.695652	0.666667	0.95061	0.974045	0.955251	0.268335	0.111499883	0.345368917
2019-11-26	0.662312	0.816472	0.695652	0.666667	0.95061	0.974045	0.955251	0.268335	0.111499883	0.345368917

3.4 예측 모형의 훈련 및 검증

3.4.1 논문 실험 환경

본 논문에서는 구축하는 예측 모형의 실험 환경은 [표 1-4]와 같다.

HW	CPU	2.3GHz 8코어 9세대 Intel Core i9 프로세서
	GPU Server (네이버 클라우드)	V100 GPU 메모리 : 32GB vCPU : 8개 Memory Bandwidth : 900GB/s GPU boost Clock Speed : 1,530 MHz
	GPU Server RAM (네이버 클라우드)	90GB
	OS	Linux CentOS 7.3 64bit
SW	IDE	Jupyter Notebook
	Language	Python 3.6.9
	신경망 모형 구성	Keras 2,4.0
	기계학습 모형 구성	Tensorflow, XGboost, Sklearn
	전처리	sklearn

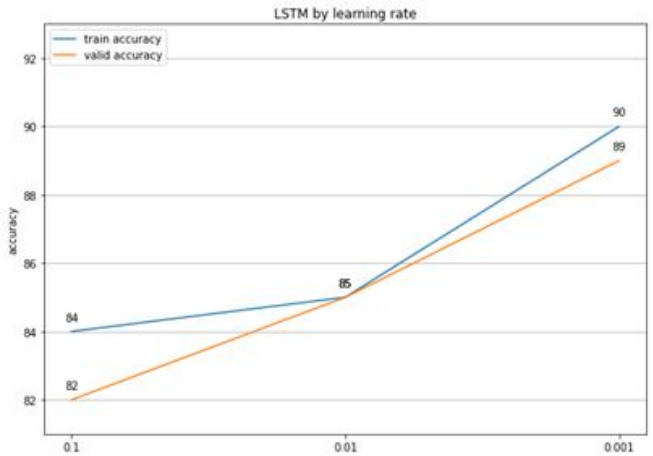
[표 1-4] 실험 환경

3.4.2 환율 예측 모델 훈련 및 검증

원/달러 환율 예측을 위해 구축된 다섯 가지의 예측 모델 각각의 초모수(hyper parameter)들의 값을 변화시키면서 가장 적합한 파라미터 값들을 선정하여 최적화된 5개의 모델을 구축하였다. 초모수는 알고리즘에 영향을 받지 않으며 최적의 모형을 결정하기 위해 자료의 학습 전에 미리 지정하는 상수이다[21].

3.4.2.1 LSTM 예측 모델

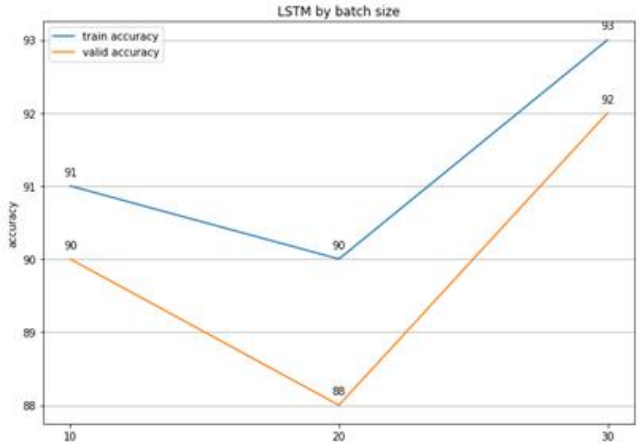
우선 learning rate를 변화시키면서 실험하였다. 학습속도를 0,1, 0.01, 0.001로 설정해서 실험한 결과 learning rate가 0.001 일 때 가장 좋은 성능을 보였다. [그림 2-8]은 learning rate 값 변화에 따른 정확도 그래프이다.



[그림 2-8] learning rate 변경 실험 그래프

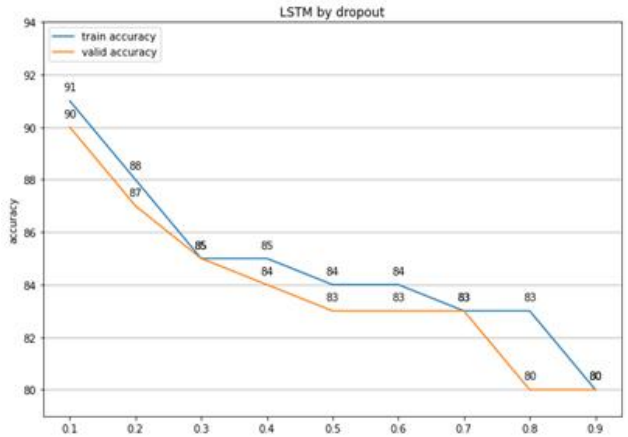
두 번째로 batch size를 변화시키면서 실험하였다. batch size를 10,

20, 30으로 설정해서 실험한 결과 batch size가 30일 때 가장 좋은 성능을 보였다. [그림 2-9]는 batch size 값 변화에 따른 정확도 그래프이다.



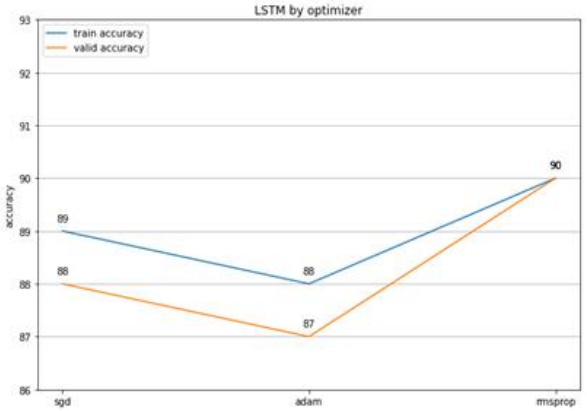
[그림 2-9] batch size 변경 실험 그래프

세 번째로 dropout를 변화시키면서 실험하였다. dropout 값을 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9로 설정해서 실험한 결과 dropout 값이 0.2일 때 가장 좋은 성능을 보였다. [그림 2-10]은 dropout 값 변화에 따른 정확도 그래프이다.



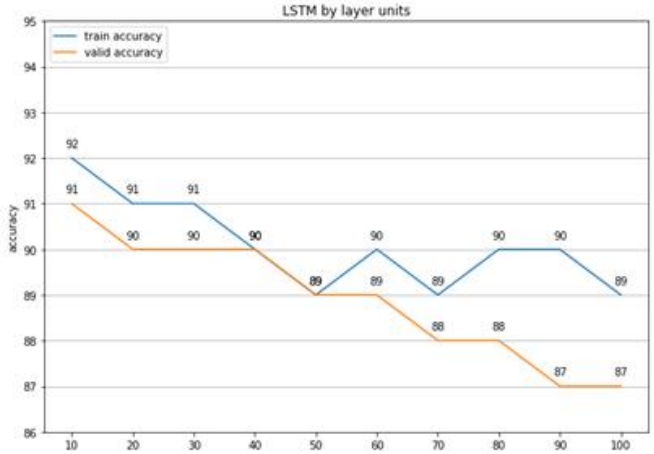
[그림 2-10] dropout 변경 실험 그래프

네 번째로 optimizer를 변화시키면서 실험하였다. otimizer의 알고리즘을 sgd, adam, rmsprop로 설정해서 실험한 결과 최적화 알고리즘을 rmsprop으로 적용할 때 가장 좋은 성능을 보였다. [그림 2-11]은 otimizer 알고리즘의 변화에 따른 정확도 그래프이다.



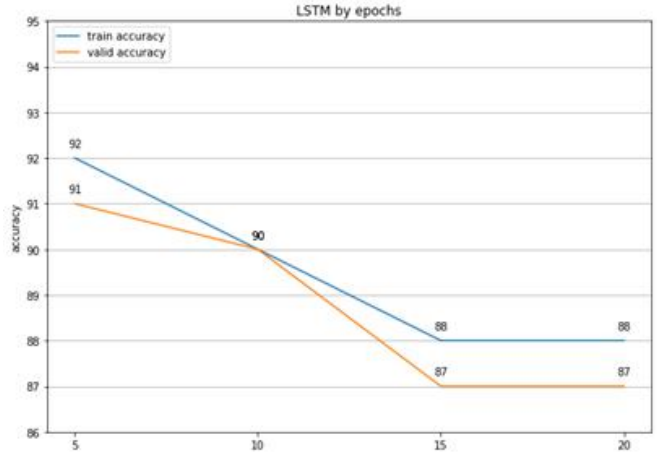
[그림 2-11] optimizer 변경 실험 그래프

다섯 번째로 레이어안의 뉴런 수를 변화시키면서 실험하였다. 뉴런의 값을 10, 20, 30, 40, 50, 60, 70, 80, 90, 100으로 설정해서 실험한 결과 뉴런의 수를 10으로 설정했을 때 가장 좋은 성능을 보였고 뉴런 수를 증가시키면 성능이 상대적으로 떨어지는 것을 확인할 수 있었다. [그림 2-12]은 뉴런수의 변화에 따른 정확도 그래프이다.



[그림 2-12] Layer unit 변경 실험 그래프

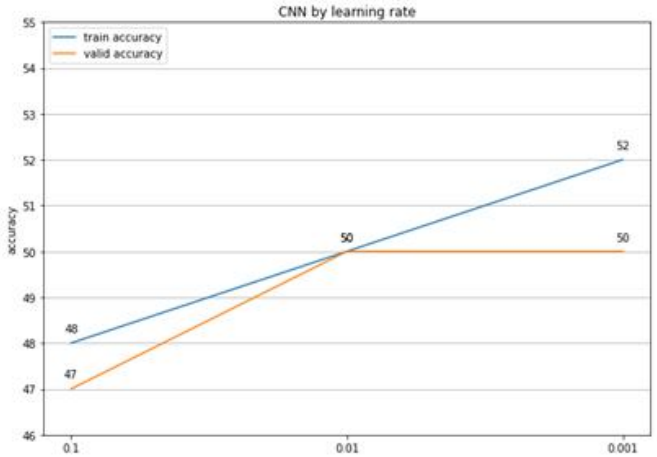
여섯 번째로 epochs를 변화시키면서 실험했을 때 epochs값을 5, 10, 15, 20으로 설정해서 실험한 결과 epochs를 5로 설정했을 때 가장 좋은 성능을 보였다. [그림 2-13]은 epochs 변화에 따른 정확도 그래프이다.



[그림 2-13] epochs 변경 실험 그래프

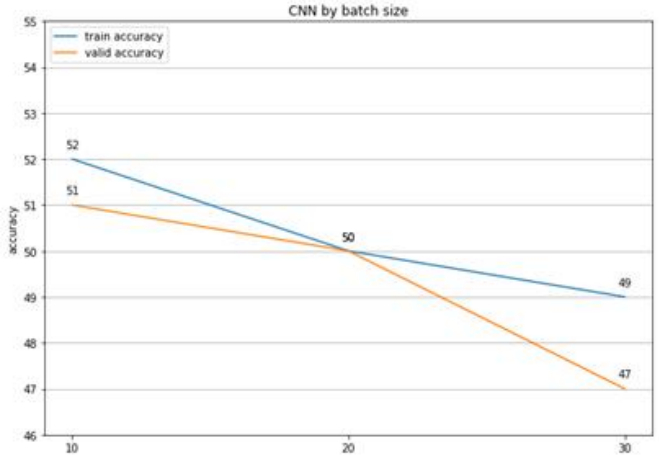
3.4.2.2 CNN 예측 모델

우선 learning rate를 변화시키면서 실험하였다. 학습속도를 0,1, 0.01, 0.001로 설정해서 실험한 결과 learning rate가 0.001 일 때 가장 좋은 성능을 보였다. [그림 2-14]은 learning rate 값 변화에 따른 정확도 그래프이다.



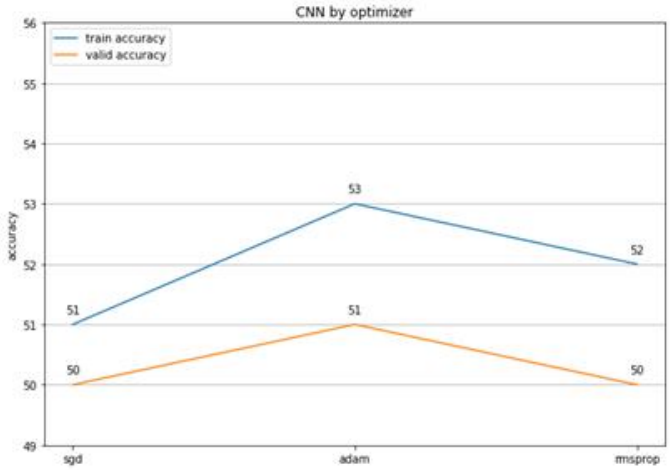
[그림 2-14] learning rate 변경 실험 그래프

두 번째로 batch size를 변화시키면서 실험하였다. batch size를 10, 20, 30으로 설정해서 실험한 결과 batch size가 10일 때 가장 좋은 성능을 보였다. 배치사이즈가 커질수록 검증데이터에서의 예측 정확도가 상대적으로 크게 떨어지는 것을 확인할 수 있었다. [그림 2-15]는 batch size 값 변화에 따른 정확도 그래프이다.



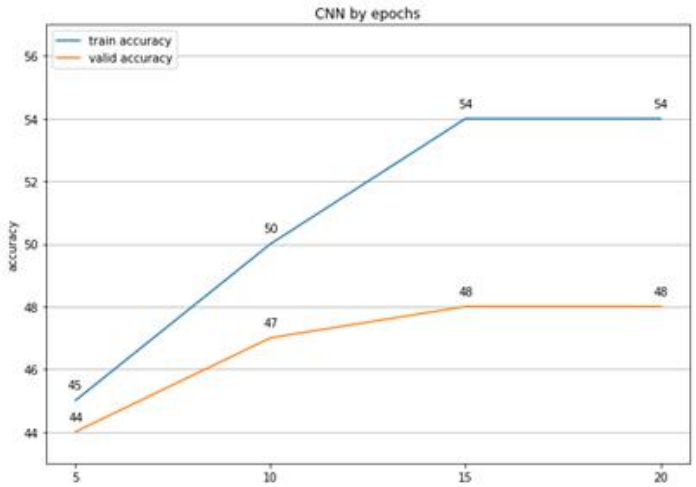
[그림 2-15] batch size 변경 실험 그래프

세 번째로 optimizer를 변화시키면서 실험하였다. otimizer의 알고리즘을 sgd, adam, rmsprop로 설정해서 실험한 결과 최적화 알고리즘을 rmsprop으로 적용할 때 가장 좋은 성능을 보였다. [그림 2-16]은 otimizer 알고리즘의 변화에 따른 정확도 그래프이다.



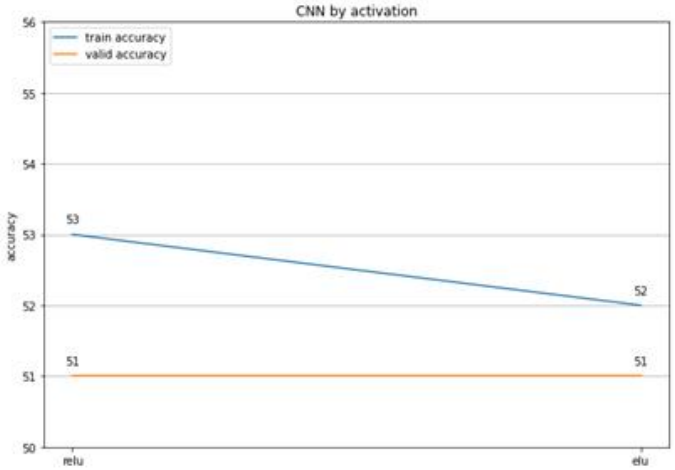
[그림 2-16] optimizer 변경 실험 그래프

네 번째로 epochs를 변화시키면서 실험했을 때 epochs값을 5, 10, 15, 20으로 설정해서 실험한 결과 epochs를 20으로 설정했을 때 가장 좋은 성능을 보였다. [그림 2-17]은 epochs 변화에 따른 정확도 그래프이다.



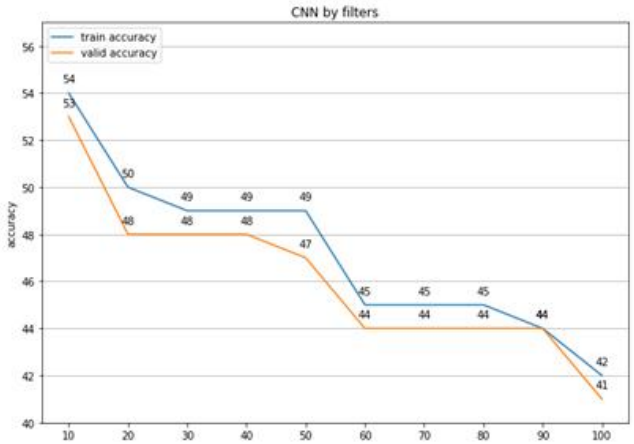
[그림 2-17] epochs 변경 실험 그래프

다섯 번째로 activation 함수를 변화시키면서 실험했을 때 활성화함수를 Relu, Elu로 설정해서 실험한 결과 활성화함수에서는 Elu와 Relu로 설정했을 때 확연한 성능차이를 보여주지 않았기에 Relu를 선택하였다. [그림 2-18]은 활성화 함수 선택에 따른 정확도 그래프이다.



[그림 2-18] activation 함수 변경 실험 그래프

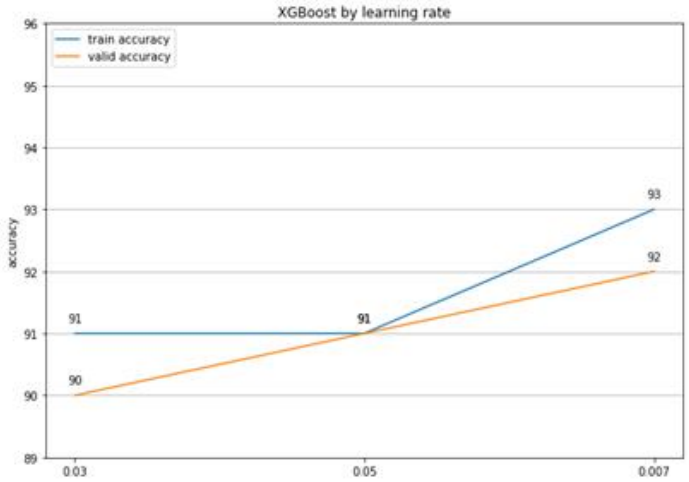
여섯 번째로 Filter를 변화시키면서 실험했을 때 Filter값을 10, 20, 30, 40, 50, 60, 70, 80, 90, 100으로 설정해서 실험한 결과 Filter를 100으로 설정했을 때 가장 좋은 성능을 보였다. [그림 2-19]는 Filter 변화에 따른 정확도 그래프이다.



[그림 2-19] Filter 변경 실험 그래프

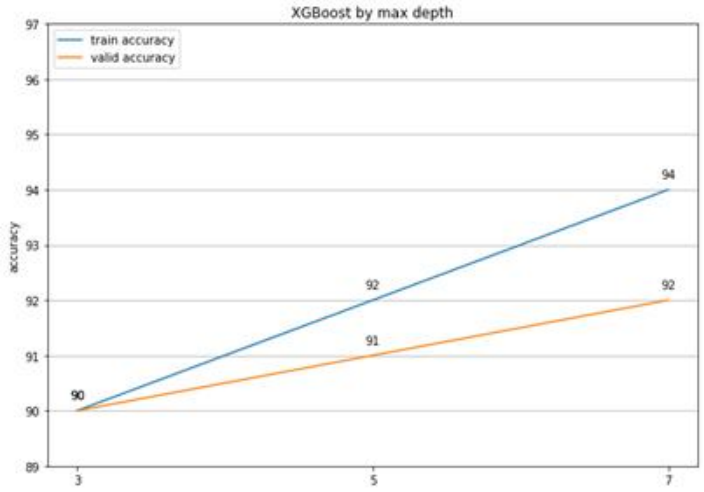
3.4.2.3 XGboost 예측 모델

우선 learning rate를 변화시키면서 실험하였다. 학습속도를 0,03, 0.05, 0.07로 설정해서 실험한 결과 learning rate가 0.07 일 때 가장 좋은 성능을 보였다. [그림 2-20]은 learning rate 값 변화에 따른 정확도 그래프이다.



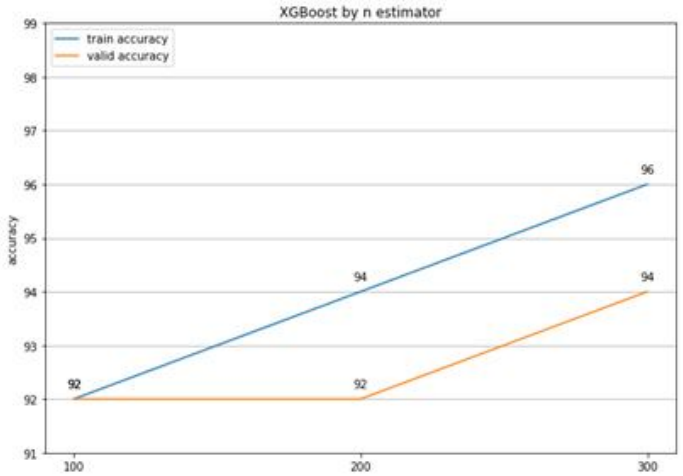
[그림 2-20] learning rate 변경 실험 그래프

두 번째로 max_depth로 실험하였다. max_depth는 Decisicion Tree의 최대 깊이를 나타낸다. max_depth의 값이 커질수록 복잡한 모델이 생성되며 훈련 데이터에 대해서 성능이 올라가지만 overfitting의 가능성이 존재한다. max_depth를 3, 5, 7로 설정해서 max_depth가 증가할수록 학습 성능은 계속 올라가는 것을 확인할 수 있었다. [그림 2-21]은 max_depth 실험 결과 그래프이다



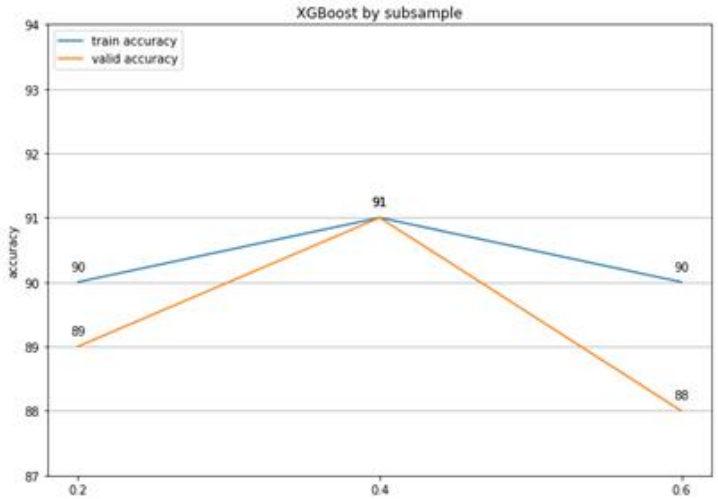
[그림 2-21] max_depth 변경 실험 그래프

세 번째로 n_estimator를 변화시키면서 실험하였다. n_estimator를 100, 200, 300으로 설정해서 실험한 결과 300으로 적용할 때 가장 좋은 성능을 보였다. [그림 2-22]은 n_estimator 변화에 따른 실험결과 그래프이다.



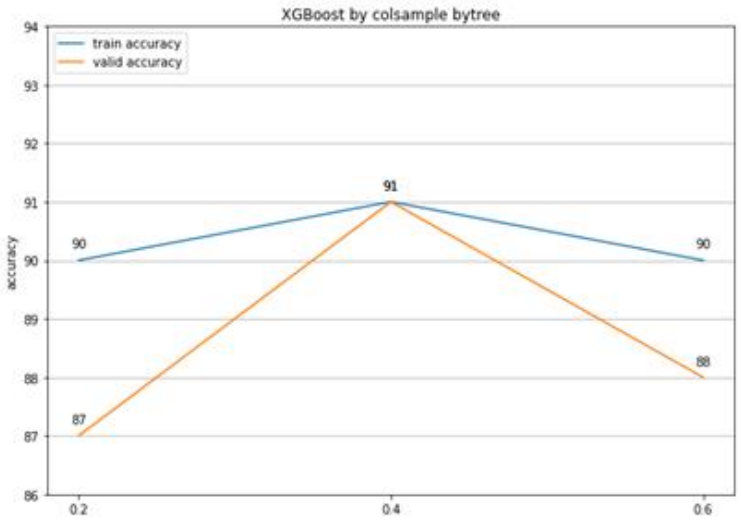
[그림 2-22] n_estimator 변경 실험 그래프

네 번째로 subsample를 변화시키면서 실험하였다. subsample의 값을 0.2, 0.4, 0.6으로 설정해서 실험한 결과 0.4으로 적용할 때 가장 좋은 성능을 보였고, 그 이후로는 정확도 오히려 떨어지는 것을 확인할 수 있었다. [그림 2-23]은 subsample 비율 변화에 따른 실험결과 그래프이다. subsample은 트리의 데이터의 비율을 의미하고 해당 값을 적게 주면 과대 적합을 방지하고 그에 따른 과소적합 이슈가 존재한다.



[그림 2-23] subsample 변경 실험 그래프

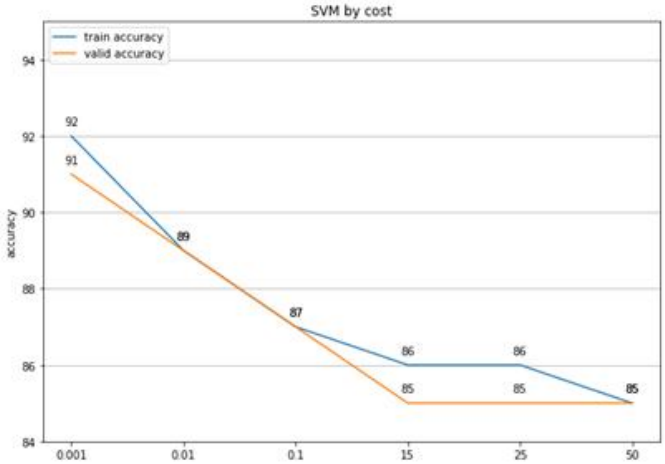
다섯 번째로 colsample_bytree의 비율을 변화시키면서 실험하였다. colsample_bytree를 0.2, 0.4, 0.6으로 설정해서 실험한 결과 0.4로 적용할 때 가장 좋은 성능을 보였고, 그 이후로는 정확도 오히려 떨어지는 것을 확인할 수 있었다. [그림 2-24]는 colsample_bytree 변화에 따른 실험결과 그래프이다.



[그림 2-24] colsample_bytree 변경 실험 그래프

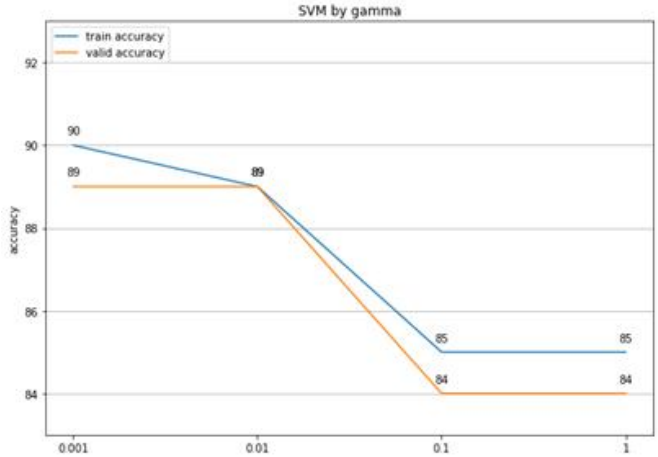
3.4.2.4 SVM 예측 모델

우선 C를 변화시키면서 실험하였다. C(제약조건의 강도)를 0.001, 0.10, 0.1, 10, 25, 50 로 설정해서 실험한 결과 C의 값이 50 일 때 가장 좋은 성능을 보였다. [그림 2-25]는 C의 값 변화에 따른 정확도 그래프이다.



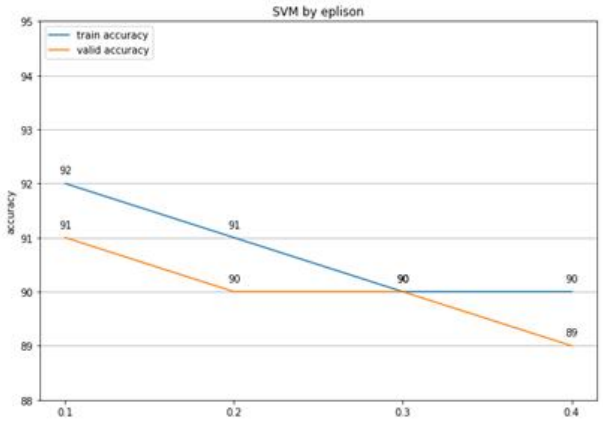
[그림 2-25] C 변경 실험 그래프

두 번째로 Gamma로 실험하였다. Gamma를 0.1, 0.01, 0.001, 0.0001로 설정해서 Gamma값이 감소할수록 학습 성능은 계속 올라가는 것을 확인할 수 있었다. [그림 2-26]은 Gamma 실험 결과 그래프이다



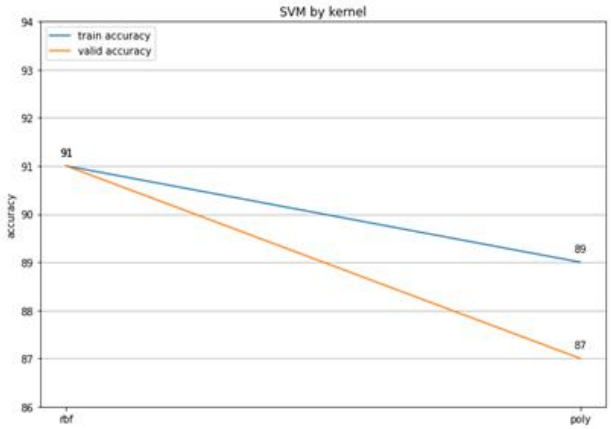
[그림 2-26] Gamma 변경 실험 그래프

세 번째로 epsilon으로 실험하였다. epsilon를 0.1, 0.2, 0.3, 0.5로 설정해서 epsilon값이 0.2까지는 성능에서 큰 차이가 없었으나 0.3부터는 학습 성능이 낮아지는 것을 확인할 수 있었다. [그림 2-27]은 epsilon 실험 결과 그래프이다



[그림 2-27] epsilon 변경 실험 그래프

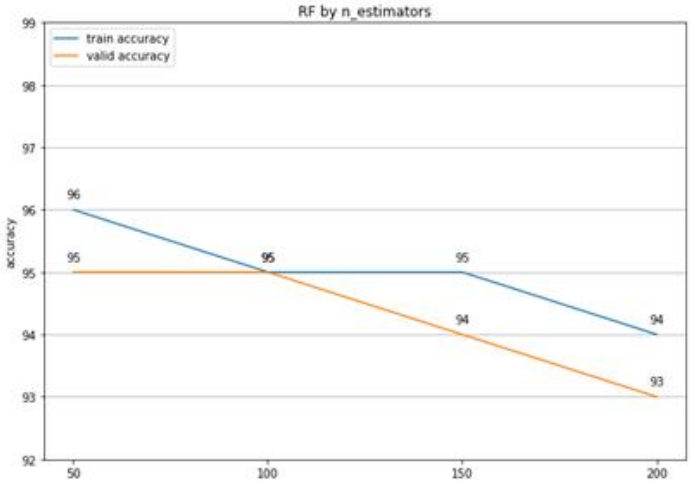
마지막으로 kernel로 실험하였다. kernel를 linear, rbf, poly로 설정해서 kernel 함수를 rbf로 설정했을 때 성능이 가장 높았다. [그림 2-28]은 kernel 실험 결과 그래프이다



[그림 2-28] kernel 변경 실험 그래프

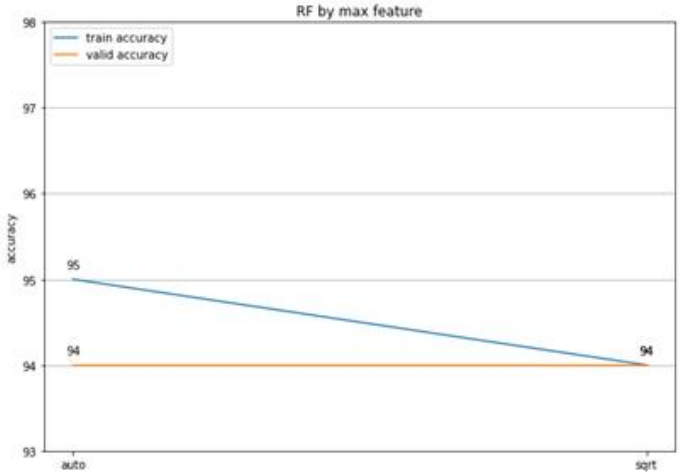
3.4.2.5 RF 예측 모델

첫 번째로 n_estimator를 변화시키면서 실험하였다. n_estimator를 50, 100, 150, 200으로 설정해서 실험한 결과 50으로 적용할 때 가장 좋은 성능을 보였다. [그림 2-28]은 n_estimator 변화에 따른 실험결과 그래프이다.



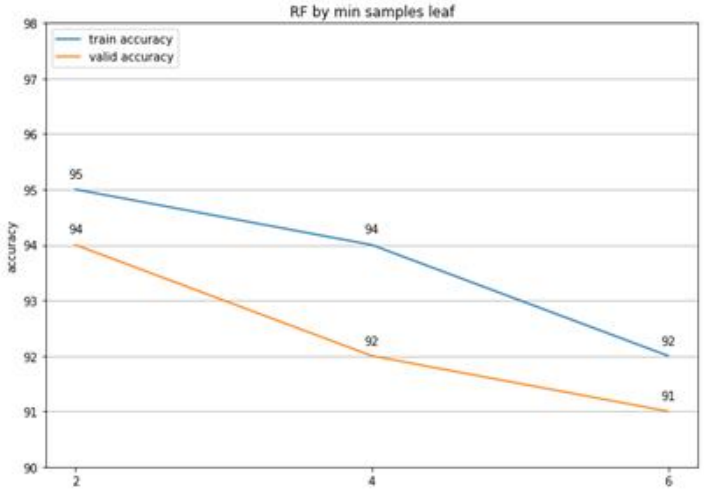
[그림 2-28] n_estimator 변경 실험 그래프

두 번째로 max_feature를 변화시키면서 실험하였다. max_feature를 auto, sqrt로 설정해서 실험한 결과 성능에서 큰 차이가 없어서 sqrt로 적용하였다. [그림 2-29]은 max_feature 변화에 따른 실험결과 그래프이다.



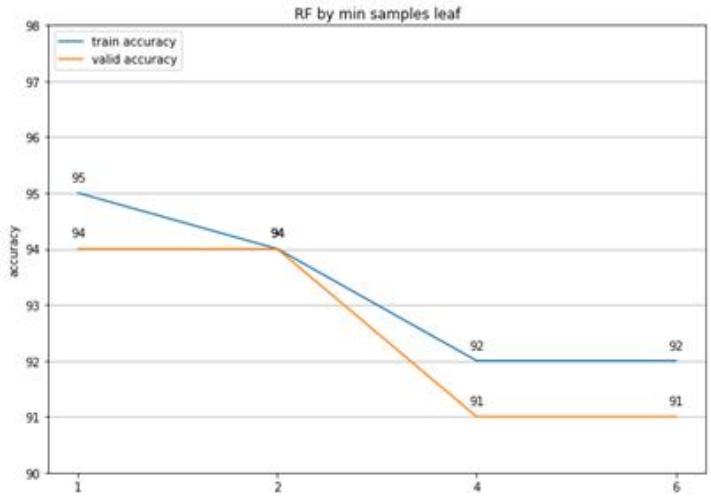
[그림 2-29] max_feature 변경 실험 그래프

세 번째로 min_samples_split를 변화해가면서 실험을 하였다. 해당 값은 2, 5, 7, 10으로 구성하여 실험한 결과 7로 설정하였을 때 가장 성능이 좋은 점을 확인할 수 있었다. [그림 2-30]은 해당 값의 변화에 따른 실험결과 그래프이다.



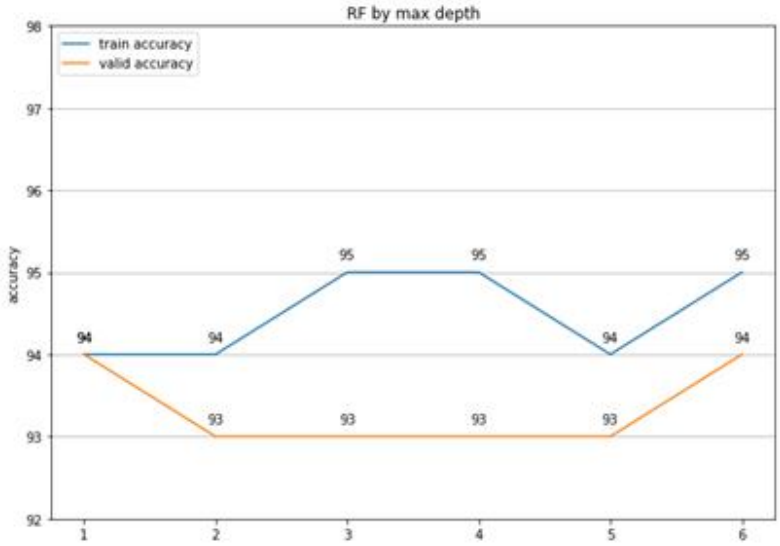
[그림 2-30] min_samples_split 변경 실험 그래프

네 번째로 min_samples_leaf를 변화해가면서 실험을 하였다. 해당 값은 1, 2, 4, 6으로 구성하여 실험한 결과 1로 설정하였을 때 가장 성능이 좋은 점을 확인할 수 있었다. [그림 2-31]은 해당 값의 변화에 따른 실험결과 그래프이다.



[그림 2-31] min_samples_leaf 변경 실험 그래프

네 번째로 max_depth를 변화해가면서 실험을 하였다. 해당 값은 1, 2, 3, 4, 5, 6으로 구성하여 실험한 결과 max_depth의 값이 4에서부터 성능은 큰 차이가 없음을 확인할 수 있었다. 본 모델에서는 해당 값을 5로 설정하였다. [그림 2-32]은 해당 값의 변화에 따른 실험결과 그래프이다.



[그림 2-32] max_depth 변경 실험 그래프

Ⅳ. 실험 결과 및 해석

4.1 LSTM 모델 실험 결과

검증 과정을 통하여 선정된 LSTM 모델의 매개변수는 아래와 같다. 아래 [표1-5]는 각 매개변수 별 실험 구간 및 실험을 통해 선정된 값이다.

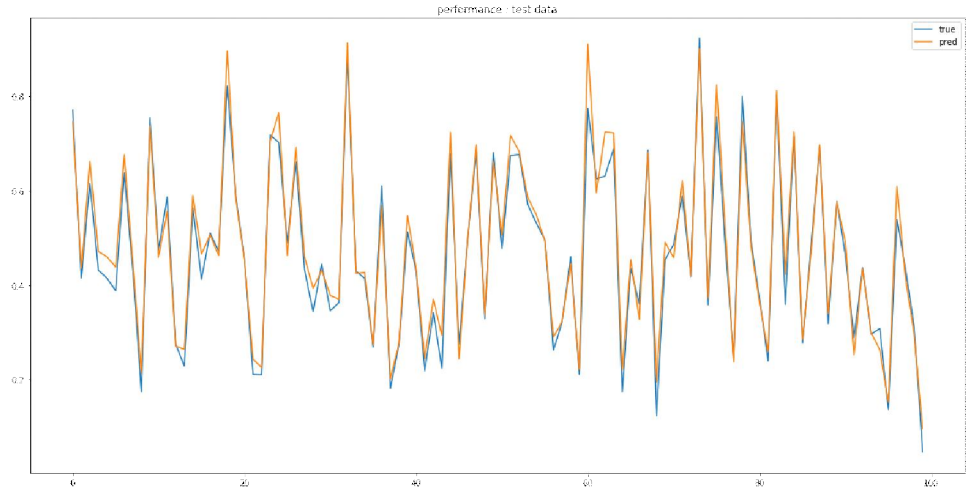
매개변수	실험 구간	설정 값
learning rate	0.1, 0.01. 0.001	0.01
Batch Size	1 ~ 30 (10단위)	30
epochs	5 ~20 (5단위)	5
first_neuron	1 ~ 100 (10단위)	30
dropout	0.1 ~ 0.9(0.1단위)	0.2
optimizer	SGD, RMSprop, Adam	Adam

[표 1-5] LSTM 모형 매개변수 선정 값

구축 모델의 성능 평가 지표는 평가 지표마다 장/단점이 존재한다. 그 단점에 의해서 잘못된 판단을 내릴 수 있기 때문에, 4가지의 회귀 평가 지표를 추출하였다. [표 1-6]는 LSTM 모델의 종합 성능 평가 지표, [그림 2-33]은 해당 모델의 예측 정확도를 시각화한 것이다.

모델명	RMSE	MAE	R^2 Score	MAPE
LSTM	0.034565	0.027392	0.959965	19.590139

[표 1-6] LSTM 의 성능 평가 지표 종합



[그림 2-33] LSTM의 Test data 예측 정확도 시각화

4.2 CNN 모델 실험 결과

CNN 예측 모형도 매개 변수인 배치 크기, 활성화 함수 종류, 필터 사이즈를 변경해가면서 가장 좋은 성능을 보여준 변수를 선택하였다. 아래의 [표1-7]은 최적의 파라미터를 찾기 위해 반복적으로 실험을 통해 선정된 값이다.

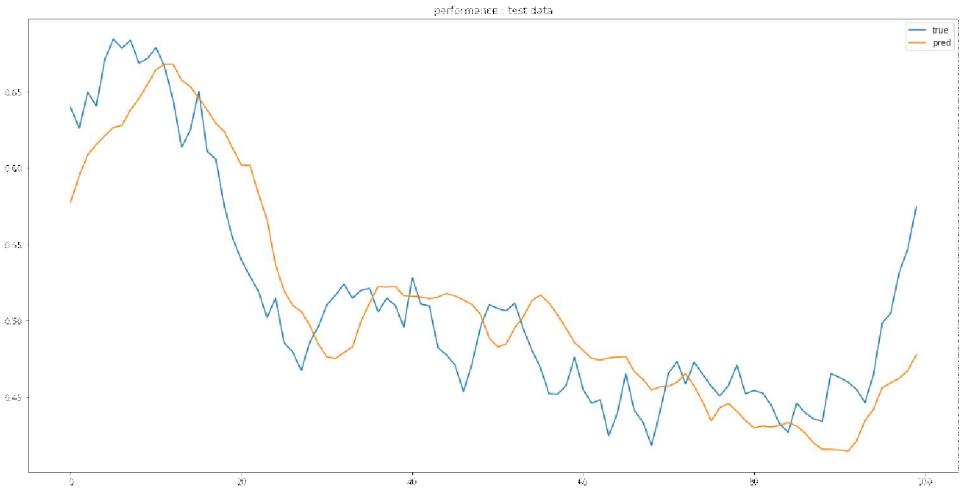
매개 변수	실험 구간	선정 값
learning rate	0.1, 0.01, 0.001	0.001
Batch Size	1 ~ 30 (10단위)	10
Epoch	5 ~ 20 (5단위)	20
Filter	1 ~ 100 (10단위)	100
activation	Relu, Elu	Relu
optimizer	Adam, SGD RMSprop, Adagrad	Adam

[표 1-7] CNN 모형 매개변수 선정 값

[표 1-8]은 CNN 모델의 종합 성능 평가 지표, [그림 2-34]는 해당 모델의 예측 정확도를 시각화한 것이다.

모델명	RMSE	MAE	R^2 Score	MAPE
CNN	0.066222	0.047213	0.754761	11.66548

[표 1-8] CNN 적용 모델의 성능 평가 지표 종합



[그림 2-34] CNN의 Test data 예측 정확도 시각화

4.3 XGboost 모델 실험 결과

해당 모형의 하이퍼 파라미터 변수인 Learning rate(학습 속도), Max_depth(트리의 깊이), N_estimator(생성할 의사결정트리의 개수), colsample_bytree(각 tree별 사용된 feature의 퍼센트 값. 높을수록 과적합), subsample(약한 학습기가 학습에 사용하는 샘플링 비율)를

변경해가면서 가장 좋은 성능을 보여준 변수를 선택하였다. 아래의 [표 1-9]는 최적의 파라미터를 찾기 위해 반복적으로 실험을 통해 선정된 값이다.

하이퍼 파라미터	실험 구간	선정 값
learning rate	0.03. 0.05, 0.07	0.07
Max_depth	3, 5, 7	7
N_estimator	100 ~ 300	300
subsample	0.2, 0.4, 0.6	0.6
colsample_bytree	0.2, 0.4, 0.6	0.4
slient	-	1
min_child_weight	-	4
nthread	-	4
objective	-	reg:linear

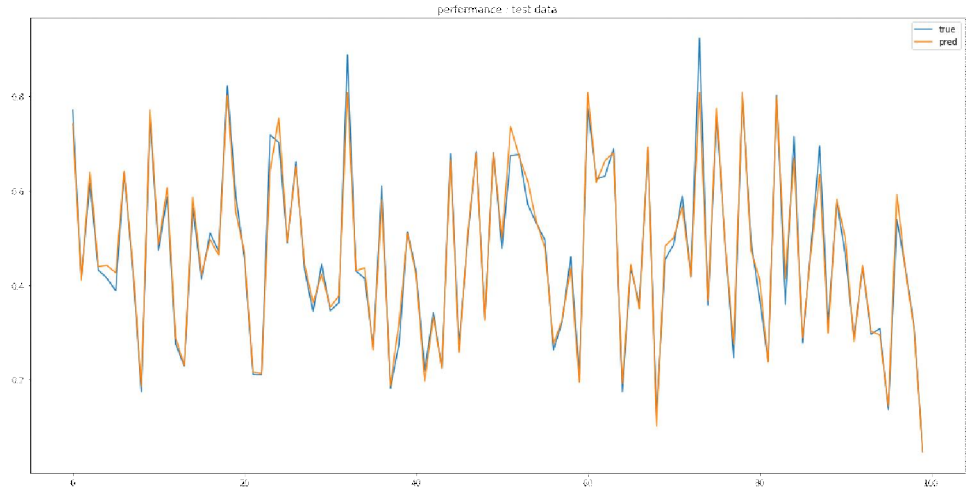
[표 1-9] XGboost 모형 매개변수 선정 값

이후 위의 선정 값으로 구축된 XGboost 모형을 가지고 3번의 반복 분할 및 훈련데이터 비율은 25%로의 설정으로 임의 분할 교차 검증 기법을 사용했다.

모델명	RMSE	MAE	R^2 Score	MAPE
XGboost	0.023015	0.016318	0.98225	13.450784

[표 1-10] XGboost 적용 모델의 성능 평가 지표 종합

위의 [표 1-10]은 XGboost 모델의 종합 성능 평가 지표, 아래 [그림 2-35]는 해당 모델의 예측 정확도를 시각화한 것이다.



[그림 2-35] XGboost 모델의 Test data 예측 정확도 시각

4.4 SVM 모델 실험 결과

해당 모형은 하이퍼 파라미터 튜닝 기법으로 그리드 탐색 기법을 적용하였다. 매개 변수인 kernel, Cost, Gamma를 변경해가면서 가장 좋은 성능을 보여준 변수를 선택하였다. 아래의 [표1-11]은 최적의 파라미터를 찾기 위해 반복적으로 실험을 통해 선정된 값이다.

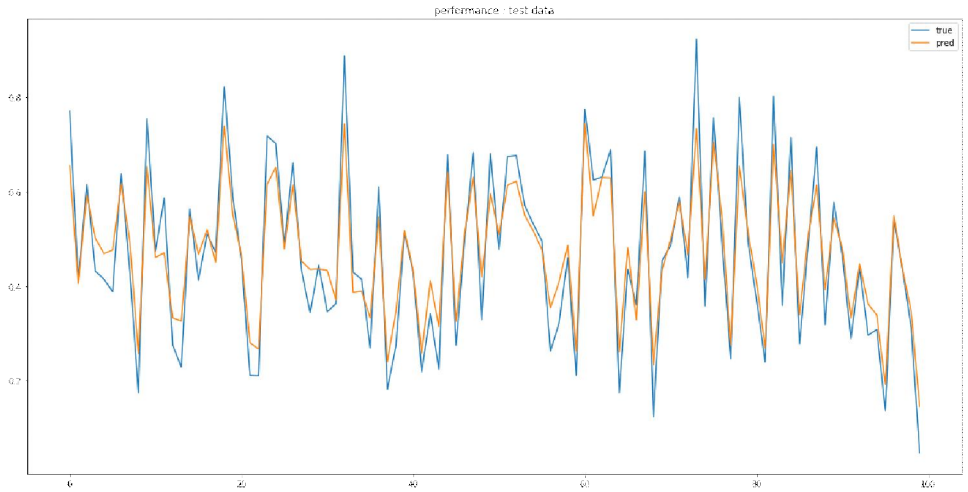
하이퍼 파라미터	실험 구간	선정 값
kernel	linear, rbf, poly	rbf
Cost	0.001, 0.10, 0.1 10, 25, 50	50
Gamma	0.1, 0.01, 0.001, 0.0001	0.01
Eplison	0.1, 0.2, 0.3, 0.5	0.1
shrinking	-	True
cache_size	-	200

[표 1-11] SVM 모형 매개변수 선정 값

[표 1-12]는 SVM 모델의 종합 성능 평가 지표, [그림 2-36]은 해당 모델의 예측 정확도를 시각화한 것이다.

모델명	RMSE	MAE	R^2 Score	MAPE
SVM	0.058341	0.051232	0.881231	37.456612

[표 1-12] SVM 적용 모델의 성능 평가 지표 종합



[그림 2-36] SVM 모델의 Test data 예측 정확도 시각화

4.5 Random Forest 모델 실험 결과

해당 모형의 하이퍼 파라미터 변수인 n_estimators(결정 트리의 개수를 지정), max_feature(최적의 분할을 위해 고려할 최대 특징 개수), min_samples_split(노드를 분할하기 위한 최소한의 샘플 데이터의 개수. 작게 설정 할수록 분할 노드가 많아져 과적합 가능성 증가), min_samples_leaf(리프노드가 되기 위해 필요한 최소한의 샘플 데이터의 개수. 과적합 제어 용도), max_depth(트리의 최대 깊이, 깊이가 깊어지면 과적합될 수 있으므로 적절히 제어 필요)를 변경해가면서 가

장 좋은 성능을 보여준 변수를 선택하였다. 아래의 [표1-13]은 최적의 파라미터를 찾기 위해 반복적으로 실험을 통해 선정된 값이다.

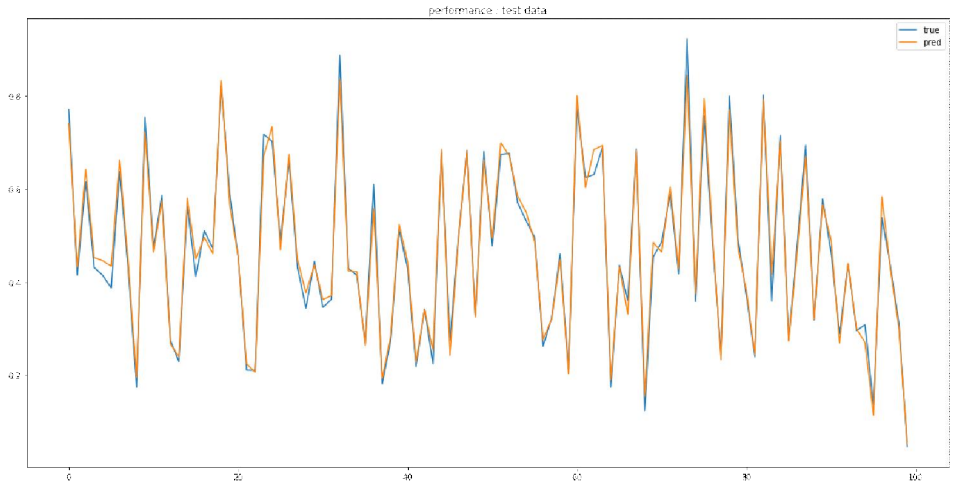
하이퍼 파라미터	실험 구간	선정 값
n_estimators	50, 100, 150, 200	50
max_feature	auto, sqrt	sqrt
min_samples_split	2, 5, 7, 10	7
min_samples_leaf	1, 2, 4, 6	1
max_depth	1, 2, 3, 4, 5, 6	5
bootstrap	True, False	False

[표 1-13] Random Forest 모형 매개변수 선정 값

이후 위의 선정 값으로 구축된 RF 모형을 가지고 3번의 반복 분할 및 훈련데이터 비율은 25%로의 설정으로 임의 분할 교차 검증 기법을 사용했다. 아래 [표 1-14]은 RF 모델의 종합 성능 평가 지표, 다음페이지 [그림 2-37]은 해당 모델의 예측 정확도를 시각화한 것이다.

모델명	RMSE	MAE	R^2 Score	MAPE
RF	0.022897	0.022982	0.978812	11.29114

[표 1-14] Random Forest 적용 모델의 성능 평가 지표 종합



[그림 2-37] RF 모델의 Test data 예측 정확도 시각화

4.6 실험 결과 분석

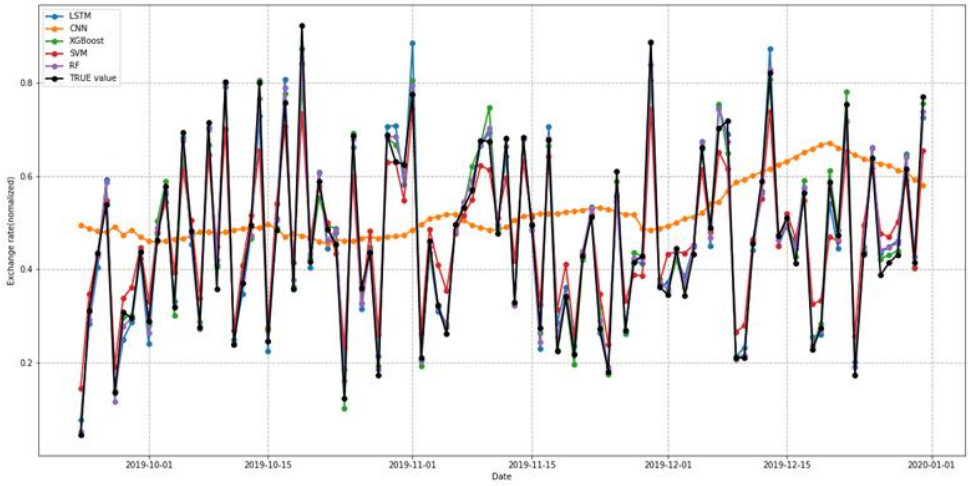
아래의 [표 1-16]는 본 논문에서 실험 결과에 대한 성능 평가 용도로 쓰일 지표들이다. 4가지의 지표(RMSE, MAE, R^2 score, MAPE)를 비교하여 구축된 모형의 학습의 적정성을 평가했다. 회귀의 평가를 위한 지표는 실제 값과 모델의 예측 값의 차이를 기반으로 한다. 회귀 평가지표 MAE, RMSE 는 값이 작을수록 모델 성능이 좋다. 값이 작을수록 예측 값과 실제 값의 차이가 없다는 뜻이다. 반면, R^2 score는 값이 클수록 성능이 좋다. MAPE는 퍼센트 값을 가지며 0에 가까울수록 회귀 모형의 성능이 좋다. 아래의 [표 1-15]의 종합 지표를 종합적으로 판단 시 XGBoost 예측 모형, RF 예측 모형, LSTM 예측 모형, SVM 예측 모형, CNN 예측 모형 순서로 상대적으로 더 정확한 예측성능을 보여주는 것을 확인할 수 있다. 아래의 [그림 2-38]에서는 총 5개의 예측 모형의 Test data set에서의 예측 흐름을 파악할 수 있고, CNN을 제외한 나머지 4개의 예측 모형은 환율의 흐름을 파악이 가능함을 확인할 수 있었다. 또한 [그림 2-39]에서는 각 모형의 예측 값의 편차 범위를 비교한 박스플롯(box plot)를 통해 모형 예측력의 상대적 정확성과 안정성을 확인할 수 있다. 편차 적은 모형은 그 모형의 예측하는 범위가 단조롭기 때문에 실제 값을 예측할 때 그 범위에 벗어나는 값들을 예측하지 못한다고 볼 수 있다.

모델명	RMSE	MAE	R^2 Score	MAPE
RF	0.023	0.023	0.98	11.29
LSTM	0.034	0.027	0.96	19.59
CNN	0.066	0.047	0.75	11.67
SVM	0.058	0.051	0.88	37.46
XGboost	0.023	0.016	0.98	13.45

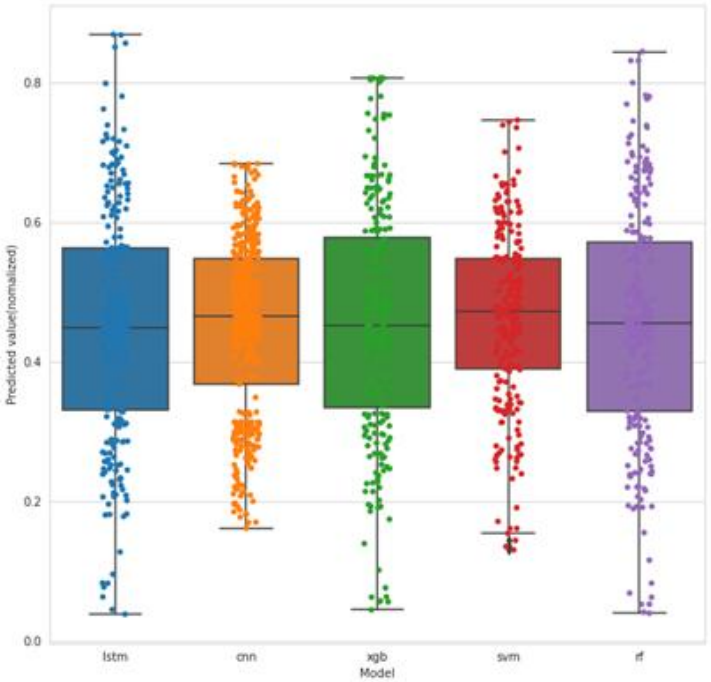
[표 1-15] 모델별 성능 평가 지표 종합

평가지표	설명
MAE	Mean Absolute Error $MAE = \frac{\sum \lvert y - \hat{y} \rvert}{n}$ 실제 값과 예측 값의 차이를 절댓값으로 변환한 평균 값이 작을수록 모델의 성능이 좋다.
RMSE	Root Mean Squared Error $\mathrm{SE} = \sqrt{\mathrm{MSE}} = \sqrt{\frac{1}{\mathrm{N}}\sum_{\mathrm{i}=1}^{\mathrm{N}}(\mathrm{y_i} - \hat{\mathrm{y}})^2}$ MSE 값은 오류의 제곱을 구하므로 실제 오류 평균보다 더 커지는 특성이 있어 MSE에 루트를 씌운 RMSE 값으로 평균값을 도출.
MAPE	Mean Absolute Percentage Error $MAPE = \frac{\sum \left\lvert \frac{y - \hat{y}}{y} \right\rvert}{n} * 100\%$ f가 제대로 추정되었는지 평가하기 위해 예측한 값이 실제 값과 유사한지 평가하는 척도가 필요하다. MAPE는 퍼센트 값을 가지며 0에 가까울수록 회귀 모형의 성능이 좋다고 해석할 수 있다. 0%~100% 사이의 값을 가져 성능 비교 해석에 용이하다
$R^2\,score$	$R^2 = 1 - \frac{SSE}{SST}$ 변수가 증가하면 증가할수록 $R^2\,score$ 증가한다.

[표 1-16] 회귀 평가 지표 설명



[그림 2-38] 각 모델의 Test data 예측 정확도 시각화



[그림 2-39] 각 모형의 예측값 편차 범위 비교

Ⅴ. 결론

전통적인 분석 도구인 시계열 분석은 사전에 정의된 모델을 기반으로 시계열의 과거와 현재의 종속 관계를 분석하고 미래를 예측하는 방법이다. 하지만 금융 시계열 데이터는 데이터에 영향을 주는 요소가 복잡하게 반영되어 있어 적절한 모델을 만들기가 어렵다. 많은 뉴런의 개수와 방대해진 연산량은 곧 과적합(Overfitting) 문제와 모형의 복잡도를 커지게 만든다.

본 연구는 금융 시계열 데이터 예측과 관련된 모형을 비교한 연구로서, 여러 거시 경제 변수를 이용한 원·달러 환율 예측에 있어서 머신러닝 방법과 딥러닝 학습 기법 활용 가능성을 확인하였다는 점에서 의의가 있다. 구축한 5개의 모형 중, 환율 예측에 가장 적합한 모델을 찾기 위해 각 모형이 가지고 있는 하이퍼 파라미터의 변수의 조정을 통해 연구를 수행하였으며 각 모형의 최적의 파라미터를 찾은 후 원-달러 환율 예측을 시도하였다. 그 결과 금융 시계열 데이터 분석에 머신러닝 알고리즘 RF모형, XGboost 모형 딥러닝 기법의 LSTM 모형이 시계열 데이터 분석에 효과가 있음을 확인할 수 있었다.

본 연구의 시사점은 다음과 같다. 환율, 주가, 지수 등 금융시계열 데이터는 상당히 많은 변수들과 인과관계를 형성하고 있다. 모델의 입력변수로 거시경제 변수를 추가했고, 이를 통해 시장상황이 일정한 추세거나 또는 불안정 시장 상황 속에 놓여 있을 때 환율의 의미 있는 예측이 가능했다. 이러한 점에서 기존 환율 예측 및 금융 시계열 데이터 예측 모형을 보완하거나 대체하는 역할을 할 수 있을 것으로 기대된다.

본 연구를 통해 딥러닝 학습이 더 우수한 성능을 보이지 않는 것을

확인할 수 있다. 딥러닝 학습 기법은 데이터의 숫자만큼이나 Feature가 풍부한 문제에 적합하다. 이는 분석 자료의 양, 종속 변수의 설정에 따라 분석결과가 달라질 수 있음을 의미하기에, 특정 모형이 우수하다고 단정 지을 수는 없다. RF, XGboost, LSTM은 결과 값이 산출되는 이유를 확인할 수 없고, 연구를 반복하면 조금씩 결과 값이 달라지는 이러한 점들은 인과관계를 중요시하는 과학에 있어서 큰 문제라고 할 수 있다.

딥러닝 학습이 금융 시계열 자료 예측에서 해결사가 되려면 보다 더 많은 데이터의 확보와 환율에 영향을 주는 변수들에 대한 추가 발굴을 통하여 지속적인 연구가 필요하다. 또한 기계학습 모형마다 다른 하이퍼 파라미터의 인과관계를 연구하여 모형을 최적화하기 위한 명확한 기준에 대한 연구가 필요하다. 추후 딥러닝 알고리즘을 사용한 예측 모형의 정확도를 높이는 연구를 지속할 예정이다.

참 고 문 헌

[1] 김재현 - 인공신경망 모형과 ARIMA 모형의 원/달러 환율예측성과 비교연구. 서강대학교 석사 논문(2002).

[2] 김호현 - LSTM/GRU 순환신경망을 이용한 시계열데이터 예측. 방송통신대학교 석사학위 논문(2017).

[3] 강민영 - CNN 기반 원 달러 환율예측 모형 연구. 서강대학교 석사학위 논문(2016)

[4] 한나영 – 인공신경망을 이용한 KOSPI예측에 대한 연구 : 금융위기 발생 전/후를 대상으로. 고려대학교 석사학위 논문(2011)

[5] 이지훈 – 딥러닝을 이용한 주가 예측 모델. 숭실대학교 석사학위 논문(2017)

[6] 배성완·유정석 - 머신 러닝 방법과 시계열 분석 모형을 이용한 부동산 가격지수 예측(2018). 『주택연구』, 26권 1호, 107-133.

[7] 이태형 - 인공신경망을 활용한 주택가격지수 예측에 관한 연구 : 서울 주택 가격 지수를 중심으로. 중앙대학교 박사학위 논문(2019)

[8] 송대섭 - 앙상블 딥러닝을 이용한 KOSPI 지수 등락 예측. 단국대학교 석사학위 논문(2015).

[9] 한국은행 국제국 외환시장팀 구종환 (2019.06). "환율 및 외환시장에 대한 이해" p.16

[10] 한국은행 홈페이지. 외환시장과 환율에 관한 정의

[11] Li, Haohao (2009) 환율과 주가의 상호관계에 대한 연구 : 두 차례 금융위기 이벤트를 중심으로 원달러환율과 KOSPI 종합지수에 대한 실증분석.

[12] 한국은행 금융경제연구원 김용복, 곽법준 (2009.04) 환율변동이 실물경제에 미치는 영향 -수출, 수입 및 투자부문을 중심으로.

[13] 국회예산정책처 연훈수 (2008.04) 환율변동이 국내물가에 미치는 영향.

[14] 가톨릭대학교 경제학과 교수 허 인, 경희대학교 국제학과 교수 안지연 - 유가과 원달러 환율의 관련성 및 원인분석.

[15] 서양모 - LSTM 모델기반 서울시 미세먼지 농도 예측 정확도 분석. 세종대학교 석사학위 논문.

[16] 랜덤포레스트 개념. 위키백과

[17] Chen, T. and Guestrin, C. XGBoost: A scalable tree boosting system. Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 785-794 (2016)

[18] 김형원 - 익스트림 그래디언트 부스팅 알고리즘에 기반한 축구 경기 예측. 서강대학교 석사학위 논문(2020).

[19] 김도현 - LSTM과 CNN을 이용한 단기 전력 수요 예측. 건국대학교 대학원 석사학위 논문.

[20] 유원준 - 합성곱 연산(Convolution operation). PyTorch로 시

작하는 딥 러닝 입문.

[21] 황승환 - 전문기관의 원/달러 환율예측력 분석: 시계열 모형 및 기계학습 LSTM 모형과의 비교 연구. 서강대학교 석사학위 논문 (2020)