2022년도 연구용역보고서

딥러닝(Deep Learning) 방법론을 활용한 GDP 예측모형 연구

- 2022. 08. -

이 연구는 국회예산정책처의 연구용역사업으로 수행된 것으로서, 보고서의 내용은 연구용역사업을 수행한 연구자의 개인 의견이며, 국회예산정책처의 공식 견해가 아님을 알려드립니다.

연구책임자

창원대학교 정수관

딥러닝(Deep Learning) 방법론을 활용한 GDP 예측모형 연구

- 2022. 08. -

연구책임자 정 수 관 (창원대학교 경제학과 부교수)
연 구 원 표 동 진 (창원대학교 경제학과 조교수)
연 구 원 허 등 용 (경북대학교 식품자원경제학과 조교수)

이 연구는 국회예산정책처의 연구용역사업으로 수행된 것으로서, 보고서의 내용은 연구용역사업을 수행한 연구자의 개인 의견이며, 국회예산정책처의 공식 견해가 아님을 알려드립니다.

국회예산정책처
National Assembly Budget Office

<차례>

<요 약> ····· 1
1. 연구 목적 ····· 3
가. 연구 배경 및 연구의 목적 ····· 3
(1) 대내외 경제 환경의 불확실성 증대 ····· 3
(2) 경제 예측모형의 진화: 기계학습(machine learning) 기반 모델의 확산 ····· 4
(3) 전통적 모형과 비전통적 모형의 결합 ····· 5
2. 연구내용 및 방법 ····· 6
가. 연구내용 ····· 6
(1) 선행연구 ····· 6
(2) 예측모형 기본 구조 ····· 11
(3) 모듈 2: 결측치(missing value) 추정 ····· 12
(3-1): Ragged-Edge 문제 ····· 12
(3-2): EM 알고리즘 ····· 14
(3-3): 동태요인모형(Dynamic Factor Model) ····· 16
(4) 모듈 3: LSTM/GRU 알고리즘을 이용한 학습 및 예측 ····· 19
(4-1): 인공신경망 개관 ····· 19
(4-2): 순환신경망 (Recurrent Neural Network) ····· 21
(4-3): LSTM (Long-Short Term Memory) 알고리즘 ····· 22
(4-4): GRU (Gated Recurrent Unit) 알고리즘 (Cho et al., 2014) ····· 24
나. 분석 자료 및 학습자료(training data) 구조 ····· 33
(1) 분석 자료 ····· 33
(2) 인공신경망 학습 자료 (training data) 구조 ····· 35
(3) 예측을 위한 새로운 입력 변수 생성 ····· 36
(4) 인공신경망 구조 설정 ····· 37
다. 예측력 비교 검증 ····· 43
(1) 벤치마크 모형 ····· 43
(2) 표본 외 예측 ····· 44
3. 주요 결과 ····· 46
(1) 코로나-19 기간(2020년) 제외 표본 외 예측 결과 ····· 46
(2) 코로나-19 기간(2020년) 포함 표본 외 예측 결과 ····· 48
4. 결론 ····· 58

5. 참고문헌 ····· 61
[부록 1: 혼합주기 자료 예측모형 결과] ····· 66
(1) 모형 셋업 ····· 66
(2) 추가 자료 ····· 71
(3) 표본 외 예측 결과 (2020년도 제외) ····· 72
(4) 코로나-19 팬데믹 기간 모형의 예측 양상 ····· 78
[부록 2: 나우캐스팅(nowcasting) 변수 활용 모형 결과] ····· 81
[부록 3: 인공신경망에 대한 이해] ····· 84
가. 생물학적 뉴런과 생물학적 신경망(Biological Neural Network; BNN) ····· 84
나. 인공뉴런(Artificial neuron)과 인공 신경망 (Artificial Neural Network; ANN) ····· 84
다. 퍼셉트론 ····· 85
라. 신경망 ····· 90
마. 신경망 시계열 분석 ····· 99
바. RNN (Recurrent Neural Network) ····· 102
사. LSTM (Long Short-Term Memory) ····· 106
[부록 4: 모형 사용 매뉴얼 및 함수 설명] ····· 112

<표 차례>

<표 1> 모듈별 기능 ····· 11
<표 2> 예측모형에 사용된 설명 변수 ····· 33
<표 3> GRU 모형 초매개변수(hyperparameters) 설정값 ····· 38
<표 4> LSTM 모형 초매개변수(hyperparameters) 설정값 ····· 39
<표 5> GDP 성장률 표본 외 예측 (2020년도 제외): RMSE ····· 51
<표 6> 초월 기준 RMSE (2020년도 제외) ····· 51
<표 7> 중월 기준 RMSE (2020년도 제외) ····· 51
<표 8> 말월 기준 RMSE (2020년도 제외) ····· 51
<표 9> 종합 RMSE (2020년도 제외) ····· 52
<표 10> GDP 성장률 표본 외 예측 (2020년도 제외): MAE ····· 52
<표 11> 초월 기준 MAE (2020년도 제외) ····· 52
<표 12> 중월 기준 MAE (2020년도 제외) ····· 53
<표 13> 말월 기준 MAE (2020년도 제외) ····· 53
<표 14> 종합 MAE (2020년도 제외) ····· 53

<표 15> GDP 성장률 표본 외 예측 (2020년도 포함): RMSE ····· 54
<표 16> 초월 기준 RMSE (2020년도 포함) ····· 54
<표 17> 중월 기준 RMSE (2020년도 포함) ····· 54
<표 18> 말월 기준 RMSE (2020년도 포함) ····· 55
<표 19> 종합 RMSE (2020년도 포함) ····· 55
<표 20 > GDP 성장률 표본 외 예측 (2020년도 포함): MAE ····· 56
<표 21> 초월 기준 MAE (2020년도 포함) ····· 56
<표 22> 중월 기준 MAE (2020년도 포함) ····· 56
<표 23> 말월 기준 MAE (2020년도 포함) ····· 57
<표 24> 종합 MAE (2020년도 포함) ····· 57
<표 25> 문헌 내 모형과 예측오차 비교 ····· 59
<표 26> 2022년 2분기 한국 경제성장률 예측치 (단위: %) ····· 59
<표 27> 혼합주기 모형 설명 변수 ····· 71
<표 28> 혼합주기 모형 GDP 성장률 표본 외 예측(2020년도 제외): RMSE ····· 74
<표 29> 초월 기준 혼합주기 모형 RMSE (2020년도 제외) ····· 74
<표 30> 중월 기준 혼합주기 모형 RMSE (2020년도 제외) ····· 75
<표 31> 말월 기준 혼합주기 모형 RMSE (2020년도 제외) ····· 75
<표 32> 종합 기준 혼합주기 모형 RMSE (2020년도 제외) ····· 75
<표 33> 혼합주기 모형 GDP 성장률 표본 외 예측(2020년도 제외): MAE ····· 76
<표 34> 초월 기준 혼합주기 모형 MAE (2020년도 제외) ····· 76
<표 35> 중월 기준 혼합주기 모형 MAE (2020년도 제외) ····· 76
<표 36> 말월 기준 혼합주기 모형 MAE (2020년도 제외) ····· 77
<표 37> 말월 기준 혼합주기 모형 MAE (2020년도 제외) ····· 77
<표 38> 통계청 발표 속보성 사회경제지표 관련 변수 ····· 81
<표 39> 속보성 지표 포함 모형 표본 외 예측: ····· 83
<표 40> 속보성 지표 포함 모형 예측오차: RMSE ····· 83
<표 41> 속보성 지표 포함 모형 예측오차: MAE ····· 83

<그림 차례>

<그림 1> 예측 모형 구조 ····· 12
<그림 2> Ragged-Edge 문제 예시 ····· 13
<그림 3> 결측치가 채워진 상태의 분석 자료 예시 ····· 18

<그림 4> Multilayer Perceptron 구조 ····· 19
<그림 5> 순환신경망(Recurrent Neural Network) 구조 ····· 21
<그림 6> LSTM 구조 ····· 23
<그림 7> RNN, LSTM, GRU 구조 비교 ····· 24
<그림 8> 과소적합(underfitting)과 과대적합(overfitting) ····· 25
<그림 9> 학습, 시험, 검증 데이터 분리 ····· 26
<그림 10> 드롭아웃을 적용하지 않은 경우와 적용한 경우 작동 원리 ····· 29
<그림 11> GRU 구조 ····· 31
<그림 12> GRU의 망각 게이트(forget gate) 구조 ····· 31
<그림 13> GRU의 업데이트 게이트 구조 ····· 32
<그림 14> 순환신경망 학습자료(training data) 구조 ····· 36
<그림 15> 예측을 위한 새 입력 변수(new instance) 생성 예시 ····· 37
<그림 16> 경사 하강법 (Gradient Descent) 과정 ····· 40
<그림 17> 학습률에 따른 경사 하강법 과정 ····· 41
<그림 18> 2020년도 예측 양상: 롤링윈도우 예측 ····· 79
<그림 19> 2020년도 예측 양상: 축차예측 ····· 79
<그림 20> 생물학적 뉴런 ····· 84
<그림 21> 뇌 신경망과 인공 신경망 구조 ····· 85
<그림 22> 입력 신호가 두 개인 퍼셉트론 예시 ····· 86
<그림 23> 퍼셉트론 진리표 예시 1: AND 게이트 ····· 87
<그림 24> 퍼셉트론 진리표 예시 2: NAND 게이트 ····· 88
<그림 25> 퍼셉트론 진리표 예시 3: OR 게이트 ····· 88
<그림 26> 퍼셉트론 진리표 예시 4: XOR 게이트 ····· 88
<그림 27> 퍼셉트론 시각화 (왼: OR게이트 조건, 우: XOR게이트 조건) ····· 89
<그림 28> XOR 퍼셉트론 ····· 90
<그림 29> 신경망 구조 예시 ····· 91
<그림 30> 활성화 함수 형태: 시그모이드, 하이퍼볼릭 탄젠트, ReLu (왼쪽부터) ····· 92
<그림 31> 순전파 학습과 역전파 학습방향 및 과정 ····· 94
<그림 32> 실제 값과 예측값 차이 ····· 95
<그림 33> 손실함수의 도함수와 매개변수 관계 ····· 97
<그림 34> 시각화와 벡터장 ····· 97
<그림 35> 경사하강법, 확률적 경사하강법, 미니배치(mini batch) 방법 원리 ····· 98

<그림 36> Timestep에 따른 순환뉴런 흐름 ····· 100
<그림 37> 순환층과 Timestep에 따른 순환층 흐름 ····· 100
<그림 38> Timestep에 따른 메모리 셀 흐름 ····· 101
<그림 39> 일대다, 다대일, 다대다, 동기화 다대다 구조 (시계방향) ····· 102
<그림 40> RNN 다대다 구조 ····· 103
<그림 41> 자동검색어 완성 예시 ····· 104
<그림 42> RNN 모델에서의 가중치 ····· 104
<그림 43> RNN 순전파 학습과 역전파 학습 ····· 106
<그림 44> RNN과 LSTM 출력구조 차이점 ····· 106
<그림 45> LSTM 계층 구조 ····· 107
<그림 46> LSTM 망각 게이트 구조 ····· 107
<그림 47> LSTM 새로운 기억 셀과 입력 게이트(input gate)구조 ····· 108
<그림 48> LSTM 출력 게이트 구조 ····· 110
<그림 49> LSTM 역전파 학습 ····· 111
<그림 50> Anaconda 다운로드 및 설치 ····· 112
<그림 51> 주요 라이브러리 설치 1 ····· 114
<그림 52> 주요 라이브러리 설치 2 ····· 115
<그림 53> 편집기 (IDE) 선택 ····· 116
<그림 54> 편집기(IDE) 예시: Spyder ····· 117
<그림 55> 편집기(IDE) 예시: Jupyter Notebook ····· 118
<그림 56> 편집기에서 함수 실행 ····· 119

<요 약>

❑ 대내외 거시경제적 충격이 지속적으로 발생하는 상황에서 경기변동 최소화를 위한 재정정책 및 통화정책의 선제적인 계획 수립·수행은 현재 거시경제 상황에 대한 면밀한 평가의 기초 위에서 가능함

○ 하지만 국민총생산(GDP) 집계 시차 존재라는 구조적 제약 존재로 인해, 관련 분야에서는 나우캐스팅(nowcasting)이라는 새로운 접근방법을 통해 현 거시경제상황에 대한 평가 및 단기 경제성장률 예측에 대한 연구가 활발히 이루어지고 있음

- 본 연구는 4차 산업혁명시대의 핵심적인 기술로 평가받고 있는 딥러닝(deep learning) 알고리즘을 단기 경제성장률 예측모형에 적용하여, 그 예측력을 평가하는 것을 목표로 하고 있음. 구체적으로, 시계열 예측에 특화되어 있는 장단기기억 (Long-Short Term Memory; LSTM) 및 GRU (Gated Recurrent Unit) 알고리즘을 단기 경제성장률 예측 모형에 적용함

- 본 연구는 예측에 사용할 자료상에 존재하는 결측치를 동태요인모형(Dynamic Factor Model; DFM)을 통해 추정하였고, 관측가능한 자료와 추정된 자료를 딥러닝 알고리즘의 입력변수로 사용함

❑ LSTM 예측모형은 벤치마크 예측모형 AR(1)에 비해 상대적으로 낮은 예측오차를 보이나, DFM 모형과는 예측력에 있어서 큰 차이를 보이지 않는 것으로 관측됨

○ 2020년도를 제외한 표본 외 당분기 성장률 축차(recursive) 예측에서 LSTM 예측 모형의 예측오차(RMSE)는 약 0.48을 기록하였으며, GRU 예측 모형의 예측오차(RMSE)는 약 0.58을 나타냄

- 2020년도를 제외한 표본 외 당분기 성장률 축차예측에서 LSTM 예측 모형의 예측오차(MAE)는 약 0.38, GRU 예측 모형의 예측오차(RMSE)는 약 0.49를 나타냄

○ 2020년도를 제외한 표본 외 당분기 성장률 롤링윈도우 (rolling window) 예측에서 LSTM 예

측 모형의 예측오차(RMSE)는 약 0.48을 기록하였으며, GRU 예측 모형의 예측오차(RMSE)는 약 0.54를 나타냄

- 2020년도를 제외한 표본 외 당분기 성장률 롤링 윈도우 예측에서 LSTM 예측 모형의 예측오차(MAE)는 약 0.37을 기록하였으며, GRU 예측 모형의 예측오차(RMSE)는 약 0.44를 나타냄

❑ 본 연구에서 제시하는 모형은 벤치마크 모형 대비 예측력을 획기적으로 개선시키지는 못하였지만, 거시경제적 위기 상황인 2020년도(코로나-19 팬데믹) 우리나라 경제성장률 경로에 대한 전반적인 흐름을 잘 포착하는 것으로 관찰되었음

○ 모형의 지속적인 보완을 통해 우리나라 단기 경제성장률 예측 모형을 고도화시킬 수 있을 것으로 기대함

○ 예측 모형의 지속적인 보완 및 개선이 가능하도록 모형과 관련한 API (Application Programming Interface) documentation을 첨부함

1. 연구 목적

가. 연구 배경 및 연구의 목적

(1) 대내외 경제 환경의 불확실성 증대

❑ 글로벌 지정학적 위험 증대, 인플레이션 압력 가중에 따른 전세계적인 고금리 정책으로 전환 등 우리나라를 둘러싼 대내외적 경제환경의 불확실성은 크게 증가하고 있음

○ 지난 2년간의 전례 없는 팬데믹 상황은 우리나라 거시경제 및 글로벌 경제에 심대한 부정적 영향을 미쳤음

- 코로나 바이러스의 재확산 가능성 대두, 러시아-우크라이나 전쟁은 원유를 비롯한 원자재 가격 상승을 주도하면서 인플레이션이 상방압력이 점차 가중되고 있고, 이에 대응하기 위해 연준을 비롯한 각국 중앙은행의 유동성 회수 움직임이 이미 시작되고 있는 시점

❑ 정책당국 및 각국 중앙은행은 경기변동을 최소화하기 위한 다양한 정책들을 도입하고 있으나, 이런 거시경제정책 및 통화정책의 수립 및 실행은 현재 우리나라가 처해있는 거시경제적 상황에 대한 정확한 진단이 선행되어야 함

○ 한 국가의 거시경제적 성과를 요약해서 나타내고 있는 국내총생산(GDP)에 대한 선제적이고 정확한 예측은 관련 정책의 효과적인 수립 및 정책평가에 필수적인 요소라 할 수 있음

- GDP에 대한 선제적인 예측의 중요성에도 불구하고 관련 데이터 수집의 구조적인 제약에 직면

- 분기별 GDP에 대한 속보치는 다음 분기 첫째 달에 발표되고 있으며, 속보치는 다음 분기 말에 발표되는 등 발표 시점에 시차가 존재함에 따라 현재 거시경제 상황에 대한 실시간 평가에 어려움 존재

(2) 경제 예측모형의 진화: 기계학습(machine learning) 기반 모델의 확산

❑ GDP 예측의 적시성 문제를 해결하기 위하여 관련 분야에서는 월별 자료를 이용한 분기별 GDP 또는 분기별 거시경제 변수에 예측에 대한 연구가 활발히 이루어졌음

○ 보통 문헌에서 혼합주기 (mixed frequency) 모형이라 일컬어지는 일련의 모형들은 발표 주기 및 자료의 발표 시점의 차이에 따른 문제를 해결하면서 GDP 예측력을 높일 수 있는 모형 개발에 초점을 맞추고 있음 (e.g., Astveit et al., 2011; Barhoumi et al. 2008; Clements and Galvao, 2009; Foroni and Marcellino, 2012; Giannone et al., 2008; Nunes, 2005; Mittnik and Zadronzy, 2004; Parigi and Schiltzer, 1995; Rathjens and Robins, 1993; Zheng and Rossiter, 2006; 강창구, 2010; 김덕파 외, 2020, 김윤영, 2010; 김윤영·박준용, 2007; 최병재·한진현, 2014; 황선웅·박문수, 2012; 황종률, 2012)

- 특히 동적요인모형(Dynamic Factor Model; DFM)은 nowcasting에 있어서 이러한 혼잡주기 데이터의 문제를 다루기 위한 대표적인 방법론으로 활발하게 사용되어 왔음 (Stock and Watson, 2002)

- DFM을 이용한 GDP 예측과 관련된 대표적인 연구로는 Matheson (2011), Marcellino and Schumacher (2010), Chernis and Sekkel (2017), 오현희 (2021)를 들 수 있음

❑ DFM은 nowcasting이 직면한 여러 가지 기술적인 문제들을 해결할 수 있는 강점이 있기는 하지만, 다른 분야에서 인상적인 예측력을 보여주고 있는 딥러닝(deep learning) 방법론의 한 형태인 인공신경망(Artificial Neural Network; ANN) 또는 기타 기계학습(machine learning) 알고리즘에 기반한 새로운 방법론을 이용한 예측의 움직임이 서서히 확대되고 있음

○ 이런 배경에는 DFM을 포함한 전통적인 계량모형들이 가지고 있는 구조적 한계 (즉 설명변수들의 선형관계에 의존한 예측)를 기계학습 알고리즘을 통해 일부 극복할 수 있다는 장점과, 많은 변수를 효과적으로 관리할 수 있다는 점을 들 수 있음

○ 특히 여러 기계학습 알고리즘 중 인공신경망(ANN) 모델의 적용이 비교적 활발히 진행

- ANN에 기반한 알고리즘은 '자동 번역', '이미지 인식', '검색엔진 추천 기능', '자율주행' 등을 가능케 하는 핵심적인 요소임

❑ 다양한 예측 분야에서 두각을 나타내고 있는 딥러닝(deep learning) 방법론에 기반한 GDP 예측모형 개발을 통해 정책당국자는 전통적인 계량모형에 기초한 예측 모형과 상호보완적으로 활용할 필요가 있음

○ 딥러닝이 주로 적용되어온 영역인 '음성인식', '이미지 인식', '자율주행'과 같은 분야가 거시경제 시스템 동일한 대상으로 볼 수 없어, 어떤 접근법이 예측력 관점에서 더 우수한지 사전적으로 예단할 수 없음

○ 다만, 본 연구를 통해 딥러닝 방법론에 기반 우리나라 GDP 예측모형의 한계 효용에 대한 평가가 이루어질 필요가 있음

(3) 전통적 모형과 비전통적 모형의 결합

❑ 본 연구의 목적은 전통적인 계량경제학적 방법론과 기계합습(machine learning)에 기반한 알고리즘의 방법론적 결합을 통해 우리나라 경제성장률 단기 예측 모형을 개발하는데 있음

○ DFM 모형과 같은 전통적인 계량모형과 인공신경망과 같은 기계학습 관련 알고리즘의 장점을 모두 가지고 있는 예측모형의 예측력을 점검

○ 또한 예측 관련 실무자가 자신의 재량 하에 탄력적으로 모형 운영을 가능케 하는 인터페이스가 탑재된 예측모형 개발이 목표임

2. 연구내용 및 방법

가. 연구내용

(1) 선행연구

❑ 기계학습 알고리즘을 이용한 예측은 광범위한 분야에서 진행되고 있고, 그 중 거시경제 예측과 관련성이 높은 연구는 아래와 같음

○ 문헌에서는 전통적인 계량모형과 기계학습 기반의 예측 모형과의 비교가 주로 이루어지고 있으며, 특정 접근 방법이 더욱 우수하다는 결론을 내릴 수 있는 명확한 실증적 증거는 부족함

○ 다만, 비교적 과거 연구의 경우 초기 단계의 기계학습 기반 예측 모형을 사용하였고, 데이터의 축적 및 알고리즘의 선진화로 인해 최근 연구에 있어서는 기계학습 기반의 예측모형의 예측력이 상대적으로 더욱 우수하다는 실증적 증거가 많이 보고되고 있음

❑ Swanson and White (1997)

○ 미국의 주요 9개 거시경제 변수 예측을 위한 다양한 모형간의 예측력 비교를 실시

- 연구에서 고려되고 있는 모형은 적응적[1] 단변량(adaptive univariate) 모형, 비적응적 단변량(nonadaptive univariate) 모형, 적응적 비선형(adaptive nonlinear) 모형, 비적응적 다변량(nonadaptive multivariate) 모형, 전문가 예측치, 인공신경망 기반 예측 모형임

- 분기별 실업률, 산업생산, 명목 GDP, 기업의 수익성, 민간 소비지출, 재고투자, 순수출이 설명변수에 포함됨

- 적응적 벡터자기회귀모형(VAR)이 비교군 예측모형 대비 가장 우수한 예측력을 가

1) 본 연구에서 적응적 모형이란 의미는 매 예측시점마다 예측 모형의 세부 구조(specification)를 특정 기준 하에 시변적으로 조정한다는 의미임.

지고 있는 것으로 보고하고 있음

❑ Biau and D'Elia (2012)

○ Random Forest[2] 알고리즘을 사용하여 유로지역의 당분기 GDP 예측

- 기업과 가계의 설문조사 자료 (구체적으로 Joint Harmonized European Business and Consumer Survey)를 중심으로 설명변수를 구축하여 예측모형에 활용

- AR모형과의 표본 외 예측비교를 통해 Random Forest 기반 예측 모형이 상대적으로 예측성과가 우수함을 보여줌

- 구체적으로, Random Forest 기반 예측모형의 예측오차(RMSE)는 0.66, AR모형의 예측오차(RMSE)는 0.8로 보고함

❑ Tiffin (2016)

○ Elastic Net Regression[3]과 Random Forest 알고리즘을 사용하여 레바논의 당분기 GDP 성장률 예측

- 표본 외 예측에서 거시경제적 위기 상황이라고 할 수 있는 기간 (즉, Arab Spring 및 Syrian Crisis)은 제외하였으며, 월별 주기 자료를 분기 주기 자료로 전환하여 예측에 활용

- Elastic Net Regression 기반 모형과 Random Forest 모형의 예측치를 결합하는 앙상블 기반의 예측 모형도 구축

2) 의사결정나무(Decision Tree) 기반의 앙상블(ensemble) 모형으로, 변수와 관측치의 임의 선택을 통해 여러 개의 의사결정나무의 예측치를 결합하는 형태의 대표적인 기계학습 알고리즘 중의 하나임.

3) Elastic Net Regression은 일반적인 회귀분석 과정에서 모형의 모수의 L1 norm과 L2 norm에 대한 추가적인 패널티를 부여하여 주요 모수들을 추정하는 분석 기법.

- Elastic Net Regression의 예측오차(RMSE)는 0.69, Random Forest의 예측오차(RMSE)는 1.17, 두 알고리즘의 결합한 앙상블 모형의 RMSE를 0.7로 보고함

❑ Jung et al. (2018)

○ Elastic Net, SuperLearner[4], 순환신경망(recurrent neural network)[5] 알고리즘을 사용하여 주요 선진국 및 개발도상국의 1분기 후 경제성장률 예측 모형을 구축

- SuperLearner에는 베이지안 일반 선형모형(Bayesian generalized linear model), Elastic Net, Random Forest 등 총 10개 하위 알고리즘들이 포함되어 있음

- 예측에 활용한 설명변수는 IMF의 World Economic Outlook DB에 포함되어 있는 거시경제 관련 변수들과, 서베이 기반 기업/소비자심리지수 변수, 주가지수, 국제 원자재 가격 등이 포함되어 있음

- RNN 모형의 경우 예측오차(RMSE)가 선진국 (미국, 영국, 독일, 스페인)의 경우 0.7~3.21를 기록하였으며, 개발도상국 (멕시코, 필리핀, 베트남)의 경우 1.15~1.83로 보고됨

- 반면, SuperLearer 알고리즘의 경우 예측성과를 크게 향상시키는 것으로 보고함. 선진국(미국, 영국, 독일, 스페인)의 경우 예측오차(RMSE)가 0.17~0.33 수준을 기록하였으며, 개발도상국(멕시코, 필리핀, 베트남)의 경우 예측오차(RMSE)가 0.54~0.62 범위로 보고됨

- 기계학습 기반 예측모형이 IMF의 경제성장률 전망치(World Economic Outlook)에 비해 더 높은 예측성과를 보여주는 실증적 증거를 제시함

4) SuperLearner 알고리즘은 앙상블(ensemble) 기법의 하나로, 여러 기계학습 알고리즘에 기반에 둔 예측치를 결합하여 최종적인 예측치를 산출하는 기법.
5) 순환신경망에 대한 자세한 내용은 [부록 3] 참조.

❑ Medeiros et al. (2021)

○ LASSO[6] 및 Random Forest 알고리즘을 통해 미국의 인플레이션율 예측

- FRED-MD DB내 월별 거시경제 관련 자료 예측에 활용

- 부스팅(boosting)[7] 알고리즘을 이용해 변수 및 모형 내 설명 변수의 시차(lag) 선택

- Random Forest 알고리즘이 가장 우수한 예측성능을 가지고 있는 것으로 보고하고 있으며, 경제의 불확실성에 증대되는 시기, 즉 비즈니스 사이클 상 수축 기간에서 예측력이 상대적으로 더욱 우수한 것으로 보고

- 기계학습 기반 모형의 예측오차(RMSE)는 0.73~0.82 수준을 기록

○ Tkacz (2001)

- 가장 단순한 형태의 인공신경망(neural network)을 활용하여 분기 경제성장률 및 연간 경제성장률 예측

- 분기별 과거 경제성장률, M1 및 M2 증가율, 수익률 스프레드, 주가수익률을 주요 설명변수로 활용함

- 인공신경망 기반 예측모형의 경우 1분기 이후 경제성장률 예측오차(RMSE)는 1.78~2.23을 기록

- 연간 성장률 예측에서는 신경망 모형이 벤치마크 모형(선형회귀모형 및 시계열모형)에 비해 상대적으로 우수한 예측력을 가지고 있는 것으로 보고되나, 분기 성장

6) 일반적인 선형회귀 모형에 L1 norm 패널티를 적용하여 모형의 모수를 추정하는 기법. L1 norm 패널티에 대한 자세한 내용은 <참고 3> 참조.
7) weak learners의 결합을 통해 최종적인 예측치를 산출하는 일종의 앙상블 기법.

률 예측에서는 벤치마크 모형에 비해 예측력에 큰 개선을 가져오지 못함을 보고하고 있음

❑ Egler et al. (2006)

○ 순환신경망(RNN) 모형과 MS-AR(Markov Switching Autoregressive) 모형을 미국의 분기 인플레이션에 대한 예측력을 비교

- 순환신경망의 경우 과거 인플레이션(2분기 이전) 자료를 주요 설명변수로 활용

- MS-AR 모형이 순환신경망 모형에 비해 상대적으로 더욱 우수한 예측력을 가지고 있는 것으로 보고

❑ Nakamura (2005)

○ 단순한 인공신경망 모형과 AR모형과의 미국의 인플레이션 예측 성과를 비교

- 과거 인플레이션율이 주요 설명변수로 활용

- 인공신경망 모형이 AR모형에 비해 예측력이 더욱 우수한 것으로 보고

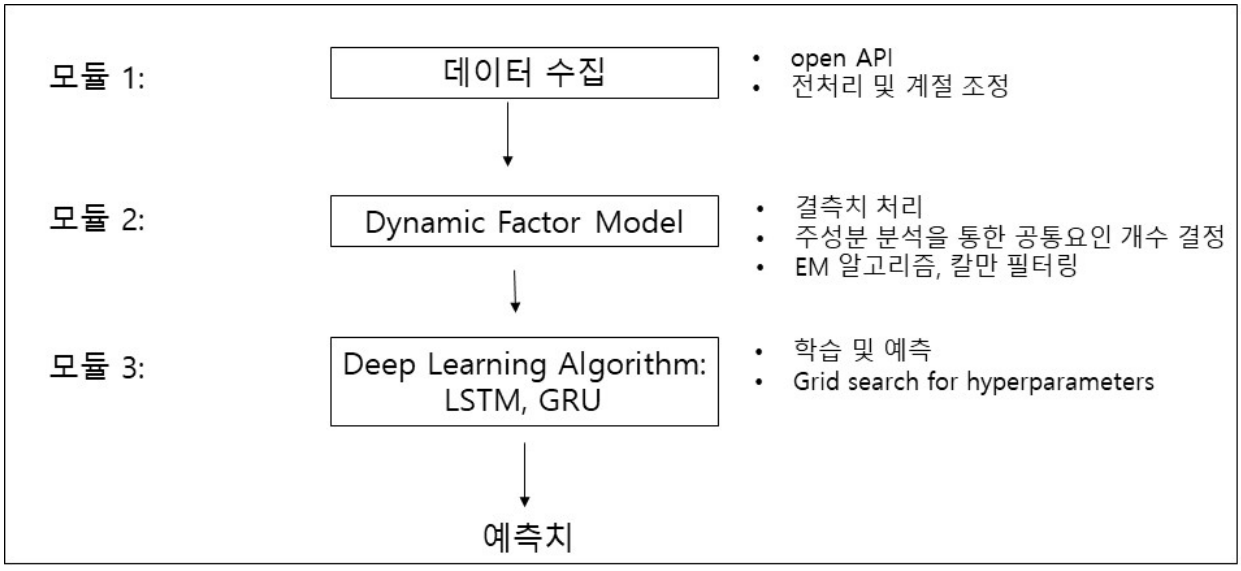
(2) 예측모형 기본 구조

❑ 동 연구에서 제시하는 경제성장률 예측모형은 <표 1>과 같이 3개의 하위 모듈로 구성

<표 1 > 모듈별 기능

	기능	관련 함수명
모듈 1	- 예측 모형에 사용할 자료의 수집 및 수집된 자료의 전처리 및 계절조정 수행 - 분석자가 필요한 자료에 대한 명세서를 제공하면 한국은행 Open API를 통해 자료의 수집 및 전처리를 자동적으로 처리하여 산출물을 제시 - 모듈 1에서 생성된 산출물은 다음 단계인 모듈 2의 입력으로 작용	- get_raw_data - get_vintage_data - transform
모듈 2	- 공통요인 개수 결정: 주성분분석(PCA; Principal Component Analysis)을 통해 주어진 예측 활용 자료의 차원 축소. Kaiser (1960) 기준을 적용하여 유효한 공통 요인 개수 결정 - 수집된 자료의 발표시차 및 발표주기의 상이함에 따른 결측치에 대한 추정치를 제공하는 역할을 수행 - 결측치 문제를 해결하기 위한 다양한 방법론들이 제시되어 있지만, 본 연구에서는 수집된 변수에 동태요인모형(Dynamic Factor Model)을 상정하고, 이에 EM 알고리즘을 적용하여 주요 동태요인모형의 주요 모수 및 자료상 존재하는 결측치를 추정 - 모듈 2에서 생성된 산출물은 모듈 3의 입력으로 작용	- get_factor_num - df_model - fill_missing
모듈 3	- 모듈 2로부터 입력을 받아 딥러닝 알고리즘 기반의 GDP성장률 예측치 산출을 위한 학습데이터(training data) 생성, 학습된 모형을 바탕으로 예측치 도출 - 순환신경망(Recurrent Neural Network)의 일종인 LSTM(Long-Short Term Memory) 및 GRU(Gated Recurrent Unit) 인공신경망을 적용 - LSTM 및 GRU 신경망의 적정 구조 설정(i.e., hyperparameter tuning)을 위해 초매개변수(hyperparameters) 공간상에서 grid search 수행	- get_training_data - train_network - train_gru_network - get_new_instance - make_prediction - get_optimal_hyperparam

<그림 1> 예측 모형 구조



(3) 모듈 2: 결측치(missing value) 추정

(3-1): Ragged-Edge 문제

❑ 혼합주기(mixed frequency) 자료를 이용한 단기 GDP 예측에 있어서 가장 먼저 대두되는 이슈는 데이터상의 결측치(missing value)에 대한 처리임

○ 민간 소비지출 증가율과 같은 분기 자료의 경우 월별 자료가 존재하지 않으며, 월별 자료의 경우에도 자료마다 발표 시차에 있어서 차이가 존재

- 가령, 설문 기반의 월별 자료(예: 소비자신뢰지수)의 경우 월말 기준으로 발표 시차 없이 획득 가능하지만, 노동시장과 관련된 자료(예: 실업률)의 경우 발표 시차가 1개월, 산업별 생산지수의 경우 발표 시차가 2개월로서, 개별 예측시점에 사용가능한 자료는 시점별(초월, 중월, 말월)로 상이한 구조를 가짐

- 분기 자료인 국내총생산 증가율[8], 소비증가율 등과 같이 국민계정에 포함된 변수

8) 한국은행은 특정 분기의 경제성장률을 속보치와 잠정치로 나누어서 발표하고 있음. 속보치는 해당 분기 종료 후 1개월 후, 잠정치는 2개월 뒤에 발표되고 있음. 경제성장률 속보치와 잠정치 데이터베이스에 대한 접근이 가능하지 않아, 본 연구에서는 분기별 확정치 자료에 개별 자료의 발표 시차를 고려하여 관찰가능 데이터(vintage dataset)를 구성함.

의 경우 약 2개월 발표 시차를 두고 확인 가능함

○ 따라서 특정 예측 시점에서 단기 예측모형 구축에 사용할 수 있는 자료는 아래 <그림 2>와 같이 변수별로 상이한 형태의 결측치 패턴을 보이게 됨

- 관련 문헌에서는 이를 Ragged-Edge 문제라고 지칭

<그림 2> Ragged-Edge 문제 예시

DATE	GDP	CONSUMPTION	PINVEST	CINVEST	EXPORT	PROD	PRODMAN	PRODSER	PRODMIN	MAFSHIP	...	IMP	BSI_SALES
2019-09	0.5	0.6	0.6	-2.3	4.0	109.0	108.0	108.4	78.8	103.7	...	38741473.0	76.0
2019-10	NaN	NaN	NaN	NaN	NaN	109.0	106.9	108.7	80.3	102.9	...	41396517.0	77.0
2019-11	NaN	NaN	NaN	NaN	NaN	109.4	107.5	109.5	84.0	100.9	...	40729215.0	80.0
2019-12	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	43687430.0	81.0
2020-01	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	79.0

주: 2020년 1월 말 기준 관측 가능 데이터(vintage data). 지면 제약으로 인해 과거 자료는 2019년 9월까지만 표시. 단, 분기자료의 경우 해당 분기말의 월에 표시. GDP, CONSUMPTION, PINVEST EXPORT는 분기 자료이며, 이외의 변수는 월별 자료임. NaN은 결측치를 나타냄.

❑ 관련 문헌에서는 혼합주기 및 결측치 문제를 해결하기 위해 다양한 방법이 시도되었음

○ 교량방정식(bridge equation) 모형은 이런 문제를 해결하기 위한 대표적인 방법으로 과거에 많이 활용되었음. 주로 동 모형 하에서는 분기 변수(quarterly variable)가 종속변수이며 월별자료가 독립변수로 상정됨 (e.g., Trehan, 1989; Parigi and Schlitzer, 1995;. Parigi and Colinelli, 2007; Runster and Sedillot, 2003, Kitchen and Monaco, 2003; Diron, 2008)

- 주기(frequency) 일치를 위해 월별자료를 분기별로 단순 평균하여 예측모형에 활용

- 가령 월별 자료에 결측치가 존재할 경우, 개별 자료에 대한 하위 모형(auxiliary model)을 바탕으로 결측치를 추정하여 분기 자료로 전환 후 예측모형에 활용

- 하지만, 이러한 접근방법은 월별 자료의 평균치를 이용한다는 관점에서 정보 손실 발생 가능하며, 이상치(outlier)에 의해 월별 자료의 평균치에 큰 왜곡이 일어날 수 있어 예측모형의 정확도에 부정적인 영향을 미칠 수 있음

○ 혼합주기 자료의 주기 일치를 위한 평균치를 통한 분기 자료로의 전환이 필요하지 않는 MIDAS(Mixed Data Sampling) 기법도 제안되었음

- 저빈도(low frequency) 자료의 변화를 설명함에 있어 아래 식 (1)과 같이 고빈도(high frequency) 자료에 distributed lag polynomial을 적용하여 회귀분석 모형에 사용함

$$y_t = \beta_0 + \beta_1 B(L^{1/m};\theta) x_t^{(m)} + \epsilon_t^{(m)}$$
$$B(L^{1/m};\theta) = \sum_{k=0}^{K} B(k;\theta) L^{k/m}$$

..... (1)

○ 최근엔 Nowcasting 영역에서 동태요인모형(Dynamic Factor Model)에 기반한 E-M(Expectation-Maximization) 알고리즘을 통해 혼합주기 문제와 결측치 문제를 해결하려는 접근방법이 활발하게 진행되고 있음 (e.g., Stock and Watson, 2002; Schumacher and Breitung, 2008; Mariano and Murasawa, 2003; Angelini et al., 2008, Banbura and Modugno, 2010 Mitchell et al., 2005; Proietti, 2008; Banbura et al. 2010)

※ 동 연구도 단일 방정식 모형이 아닌, 각 변수들의 결합 확률분포(joint probability distribution)를 고려한 EM 알고리즘 및 동적요인모형(Dynamic Factor Model)을 통해 결측치 추정

(3-2): EM 알고리즘

○ EM 알고리즘이란 데이터의 결측치 문제가 발생했을 때 해당 확률 변수의 확률분포

와 관련된 모수(parameters)를 추정 가능케 함. EM 알고리즘은 이러한 결측치의 문제뿐만 아니라, 애초부터 관측 불가능한 latent 변수의 추정도 가능케 함

- 설명의 편의상 두 개의 확률 변수 $Z = (X, Y)$ 고려하자. X와 관련된 자료만 관측 가능하고, 확률변수의 결합 확률 분포는 아래 식 (2)와 같이 주어졌다고 가정하면,

$$Z \sim N(\mu, Q)$$
$$\mu = \begin{bmatrix} \mu_x \\ \mu_y \end{bmatrix}, \; Q = \begin{bmatrix} \sigma_x^2 & \sigma_{xy} \\ \sigma_{xy} & \sigma_y^2 \end{bmatrix}$$

..... (2)

- EM 알고리즘은 이러한 기본적인 구조 하에서 두 가지 과정의 반복을 통해 두 확률변수 확률분포의 모수($\theta = \mu, Q$)를 추정

- (STEP 1): E-STEP이라고 일컬어지는 첫 번째 단계에서는 관측 가능한 데이터 X와 이전 단계(j-1)의 모수($\theta^{(j-1)}$)를 바탕으로 우도함수(likelihood function)의 기대치를 추정. 구체적으로,

$$\begin{aligned} J(\theta; \theta^{(j-1)}) &:= E[l(\theta; X, Y) \mid X, \theta^{(j-1)}] \\ &= \int l(\theta; X, y) p(y \mid X, \theta^{(j-1)}) dy \end{aligned}$$

..... (3)

- (STEP 2): M-STEP이라고 일컬어지는 두 번째 과정은 첫 번째 스텝에서 추정된 기대 우도함수를 극대화시키는 모수를 도출. 즉,

$$\theta^{(j)} := \arg\max J(\theta; \theta^{(j-1)})$$

..... (4)

- EM 알고리즘은 위 두 과정을 모수인 θ가 특정 값으로 수렴할 때까지 반복적으로 수행

(3-3): 동태요인모형(Dynamic Factor Model)

○ 결측치 추정을 위해 EM 알고리즘을 적용하기 위해서 예측에 사용할 변수들의 확률과정(stochastic process)에 대한 가정이 필요함

- 이를 위해 본 연구에서는 예측에 사용할 변수들의 확률과정이 동태요인모형(Dynamic Factor Model. 이하 DFM)을 따르는 것으로 상정함. 즉, 여러 변수 간 공통요인(f)이 존재한다고 한다면,

$$
\begin{aligned}
x_t &= \Lambda f_t + \epsilon_t \\
f_t &= A_1 f_{t-1} + \cdots + A_p f_{t-p} + u_t \\
u_t &\sim i.i.d.\, N(0,\, Q) \\
\epsilon_{it} &= \alpha_i \epsilon_{it-1} + e_{it} \\
e_{it} &\sim i.i.d.\, N(0,\, \sigma_i^2)
\end{aligned}
\qquad \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots (5)
$$

- 식 (5)에서 $y_t = (y_{1t},\, \cdots,\, y_{nt})$는 예측에 사용할 변수들로 구성된 벡터이며, Λ는 공통요인이 각 변수에 미치는 영향을 나타내는 factor loading 행렬이며, Q는 공통요인의 공분산 행렬임

○ 모듈 2의 동태요인모형에 적용되는 공통요인 개수는 주성분분석(Principal Component Analysis)을 통해 결정

- 주성분분석은 다변량 자료들의 차원을 축소하는 기법으로, 설명변수들의 선형결합으로 이루어진 주성분요인이 주어진 자료들의 분산을 극대화하는 선형결합과 관련된 가중치를 찾는 문제로 요약할 수 있음

- 주성분분석상에서 추출된 공통요인은 주어진 자료의 수만큼 존재하게 되는데, 모든

요인이 중요한 의미를 가지는 것이 아니므로, 주성분의 적정 개수를 결정할 필요 존재

- 문헌에서는 적정 주성분 개수를 결정하는 방법으로 Kaiser (1960) 기준[9], 설명력 기준, Scree 도표, Parallel 분석방법 등이 있는데, 본 연구에서는 Kaiser (1960)을 사용하여 매 예측시점에 주어진 자료의 적정 주성분 개수를 결정

- [부록 1]의 모형에서는 주성분분석이 아닌 임의의 공통요인을 부여한 모형의 결과 및 모형의 세부적인 내용을 담고 있으므로, 구체적인 내용은 [부록 1]을 참조

○ EM 알고리즘을 통해 예측에 사용할 변수들의 확률과정과 연관되어 있는 모수를 추정하기 위해, 월별 자료의 DFM에 대한 구조를 바탕으로 해당 변수들의 역학구조를 상태공간모형(state space model; 이하 SSM)으로 표현할 수 있음[10]

○ 식 (5)를 아래 식 (6)과 같은 상태공간모형(SSM)으로 나타낼 수 있음

$$\begin{aligned} x_t &= \mu + Z(\theta)s_t \\ s_t &= T(\theta)s_{t-1} + \eta_t \end{aligned} \qquad \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \quad (6)$$

- 상기한 SSM에서 상태변수 벡터(state variable vector)는 공통요인과 변수들의 idiosyncratic shock으로 이루어져 있음 (단, 아래 식은 $p = 1$인 경우를 나타냄)

$$s_t = (f'_t,\ \epsilon_{1t},\ \cdots,\ \epsilon_{nt})'$$

- 동태요인모형의 상태공간모형에서의 구체적인 구조는 식 (7), 식 (8)과 같음

9) 추정된 공분산 행렬에서 도출된 각 주성분의 고유값(eigenvalue)이 1 이상인 주성분 요인만을 유효한 공통요인으로 설정하는 기준임.
10) EM 알고리즘 E-STEP 과정의 기대 우도함수 계산에 칼만 필터링(Kalman Filtering) 알고리즘을 사용.

(Measurement Equation)

$$x_t = \mu + \begin{bmatrix} \Lambda & I_n \end{bmatrix} \begin{bmatrix} f_t \\ \epsilon_t \end{bmatrix} \qquad \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \quad (7)$$

(Transition Equation)

$$\begin{bmatrix} f_t \\ \epsilon_t \end{bmatrix} = \begin{bmatrix} A_1 & 0 \\ 0 & diag(\alpha_1, \cdots, \alpha_n) \end{bmatrix} \begin{bmatrix} f_{t-1} \\ \epsilon_{t-1} \end{bmatrix} + \begin{bmatrix} u_t \\ e_t \end{bmatrix} \qquad (8)$$

○ 앞서 상정한 상태공간모형에 EM 알고리즘을 적용하여 모형의 주요 모수들을 추정

- 추정된 모수들을 바탕으로 자료상에 존재하는 결측치에 대한 추정 가능

- [부록 1]의 모형 내 분기 자료의 경우 분기 말월은 결측치로 인식하지 않았으며, 이외의 월은 결측치로 상정하였음

- 아래 <그림 3>은 위 과정을 통해 결측치가 추정되어 예측 모형의 학습에 활용할 데이터셋에 포함된 예시를 보여줌

<그림 3> 결측치가 채워진 상태의 분석 자료 예시

DATE	GDP	CONSUMPTION	PINVEST	CINVEST	EXPORT	PROD	PRODMAN	PRODSER	PRODMIN	MAFSHIP	...	IMP	BSI_SALES
2019-09	0.500000	0.600000	0.600000	-2.300000	4.000000	-0.003663	0.013048	-0.011921	-0.092075	-0.001927	...	-0.091587	76.0
2019-10	0.827655	0.641431	1.091518	0.043155	2.766504	0.000000	-0.010237	0.002764	0.018857	-0.007744	...	0.066286	77.0
2019-11	0.338828	0.420444	-0.669273	-0.945222	1.939587	0.003663	0.005597	0.007333	0.045047	-0.019628	...	-0.016251	80.0
2019-12	0.632463	0.378089	-0.512424	1.596009	0.269640	-0.000239	-0.003277	-0.000721	-0.014191	0.002320	...	0.070115	81.0
2020-01	0.574645	0.615698	-0.377313	0.654843	0.963885	0.000703	-0.001929	0.001866	0.005544	-0.003664	...	-0.038610	79.0

주: 지면 제약으로 인해 과거 자료는 2019년 9월까지만 표시.

(4) 모듈 3: LSTM/GRU 알고리즘을 이용한 학습 및 예측

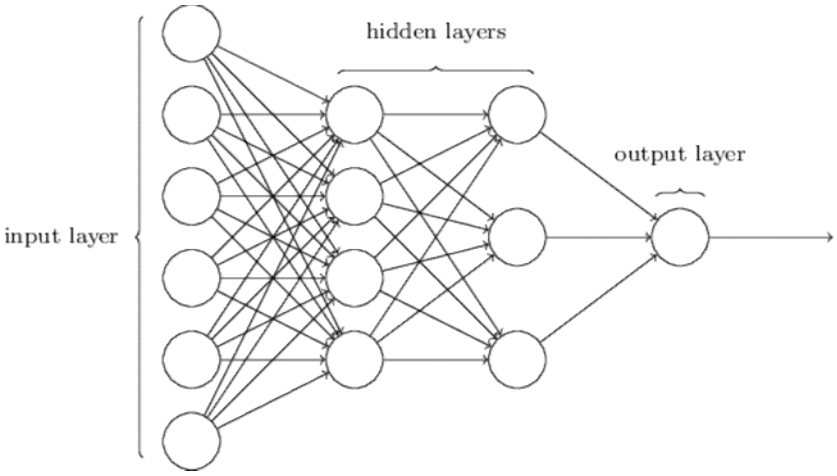
(4-1): 인공신경망 개관[11)]

❑ Multilayer Perceptron(MLP)는 인공신경망 모형 중 가장 원시적인 형태의 기계학습 방법론임

○ MLP와 같은 인공신경망(Artificial Neural Network; ANN) 모형은 전통적인 선형 계량 모형에 비해 변수들 간의 복잡한 상호관계를 더 잘 잡아낼 수 있는 것으로 알려져 있음

○ MLP는 기본적으로 아래 <그림 4>와 같이 입력층(input layer), 출력층(output layer), 은닉층(hidden layer)의 구조로 이루어져 있음

<그림 4> Multilayer Perceptron 구조



○ 입력층은 예측에 사용하고자 하는 개별 설명 변수들의 관측치를 입력 값으로 가지고 은닉층의 개별 노드(node)[12)]로 정보를 전달하는 기능을 수행함

11) 인공신경망 및 순환신경망에 대한 자세한 설명은 [부록 3: 인공신경망에 대한 이해] 참고.
12) 뉴런(neuron)으로 불리기도 함.

○ 입력 값들이 은닉층의 개별 노드(node)로 전달될시 가중치를 부여받게 되며, 입력 값들의 가중 합들은 활성화 함수(activation function)에 의해 출력 값으로 전환됨

- 가령 첫 번째 은닉층의 노드(node) 개수가 4개라고 한다면, 첫 번째 은닉층에서의 입력 값들의 가중 합은 다음과 같이 정의될 수 있음 (단, $w_i^{(j)}$는 j번째 노드가 i번째 입력 값에 부여하는 가중치)

$$z = (z_1, z_2, z_3, z_4) = (\sum_{i=1}^{d} w_i^{(1)} x_i, \sum_{i=1}^{d} w_i^{(2)} x_i, \sum_{i=1}^{d} w_i^{(3)} x_i, \sum_{i=1}^{d} w_i^{(4)} x_i) \quad \cdots\cdots (9)$$

○ 따라서 첫 번째 은닉층에서의 출력값(y^1) 벡터는 활성화 함수 f가 주어진다면 아래와 같이 계산될 수 있음

$$y^1 = f(z) = (f(z_1), f(z_2), f(z_3), f(z_4)) \quad \cdots\cdots (10)$$

○ 첫 번째 은닉층(hidden layer)에서 도출된 출력 값은 다음 은닉층(hidden layer)으로의 입력 값(input) 역할을 하게 되며, 앞과 동일한 방식으로 가중치를 부여받으며 신경망 네트워크 상에서 전방(feedforward)으로 정보를 전달하며 처리

○ MLP는 학습 데이터(training data)를 통해 각 층마다 존재하는 정보에 부여하는 가중치를 찾아나가는 과정이며, 구체적으로는 Backpropagation Through Time (BPTT) 알고리즘[13]을 통해 모형의 파라미터 추정

○ MLP는 특정 시점에 예측에 사용할 설명변수만 명확히 정의된다면 쉽게 학습을 통해 모형을 추정하고, 학습된 모형을 바탕으로 예측치를 쉽게 산출할 수 있는 장점이 있지만,[14] 시계열(time series) 자료상에 존재할 수 있는 시간의존성(temporal dependency)을 고려하지 못해 시계열 자료의 예측에 있어서 한계가 존재

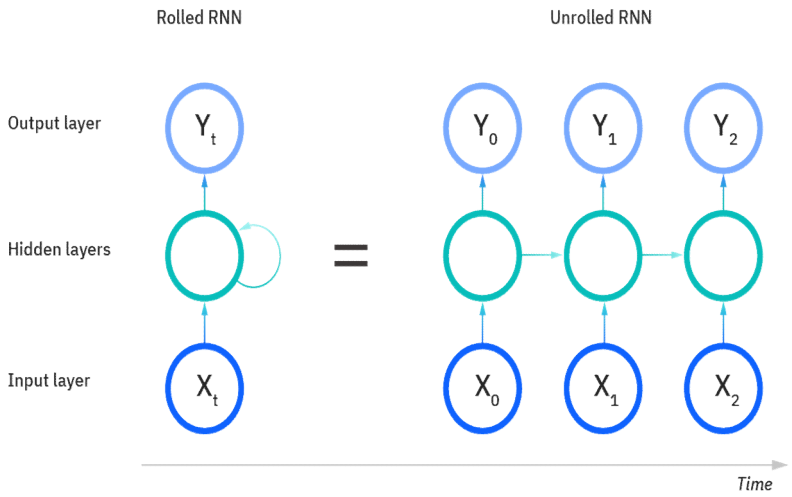
13) BPTT 알고리즘에 관한 자세한 내용은 Hecht-Nielsen (1992) 참조.
14) 예측에 사용할 설명변수의 수가 많다고 하더라도 모형의 추정에 큰 어려움이 존재하지 않음.

(4-2): 순환신경망 (Recurrent Neural Network)

❑ MLP가 가지고 있는 시계열 예측 측면에서의 약점을 보완하는 알고리즘의 대표적인 것이 순환신경망(Recurrent Neural Network; RNN)임

○ RNN의 경우 기본적인 구조는 MLP와 유사하지만, 가장 큰 차이점은 이전 시점(timestep)에서 신경망을 통해 도출된 값(y_{t-1})이 다음 timestep에서 다른 설명변수(x_t)와 함께 하나의 추가적인 입력 값(input)으로 다시 활용된다는 점임

<그림 5> 순환신경망(Recurrent Neural Network) 구조



출처: IBM

○ 이전 timestep에서의 출력 값이 현재 timestep에서의 입력 값의 역할을 하기 때문에 추가적인 가중치 파라미터(W_y)가 산출 값 계산에 사용됨

$$y_t = \phi(W_x x_t + W_y y_{t-1} + b) \quad \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \quad (11)$$

❑ RNN은 이전의 모든 timestep과 관련된 정보에 대한 기억(memory)을 담고 있는 상태 변수(state variable; h_t)를 도입할 수 있는데, 이를 기억세포(memory cell)라 지칭함

○ 기억세포는 이전 timestep의 기억세포(memory cell)와 현재 timestep의 입력 값(x_t)에 의해 결정되는 구조

○ 가령 Elman (1990)의 경우 상태변수와 출력 값 사이의 관계를 아래와 같이 모형화

$$h_t = \sigma_h(W_h x_t + U_h h_{t-1} + b_h)$$
$$y_t = \sigma_y(W_y h_t + b_y)$$

····· (12)

단, W_h, W_y, U_h는 입력 정보에 대한 가중치, b는 bias term, σ는 활성화 함수

○ RNN은 짧은 기간의 데이터의 규칙성(regularity)을 포착하는데 효율적이지만, 긴 기간에서의 패턴을 잡아내기에는 한계가 있음: timestep이 진행될 때마다 먼 과거 시점의 정보에 대한 것은 거의 사라지게 됨 (i.e., short memory)

(4-3): LSTM (Long-Short Term Memory) 알고리즘

❑ RNN이 가지고 있는 단기 기억(short memory) 단점을 극복하고, 모델 추정에 있어서 효율성을 증대시키기 위해 대안으로 제시된 것이 Long Short-Term Memory (LSTM) 알고리즘임[15] (Hochreiter, 1997)

○ LSTM의 cell memory는 단기 기억(short memory)을 위한 상태변수(h_t)와 장기 기억(long memory)을 위한 상태변수(c_t)로 나뉘어져 있음

○ LSTM의 cell은 상호작용하는 4가지 층(layer)으로 구분됨: main layer, forget gate, input gate, output gate

15) LSTM에 대한 자세한 내용은 [부록 3] 참조.

- main layer: 현재 timestep에서 주어진 입력 값(x_t)과 이전 timestep에서 생성된 단기 상태변수(h_{t-1})로부터 출력 값을 산출해 내는 역할

- forget gate: 장기 상태변수에 포함된 정보의 일부분을 제거하는 역할

- input gate: main layer에서 생성된 출력 값을 장기 상태변수에 얼마만큼 반영할지를 결정하는 역할

- output gate: 장기 상태변수와 현재 timestep의 출력 값에서 어느 정도 정보를 취할지 결정하는 역할

<그림 6> LSTM 구조

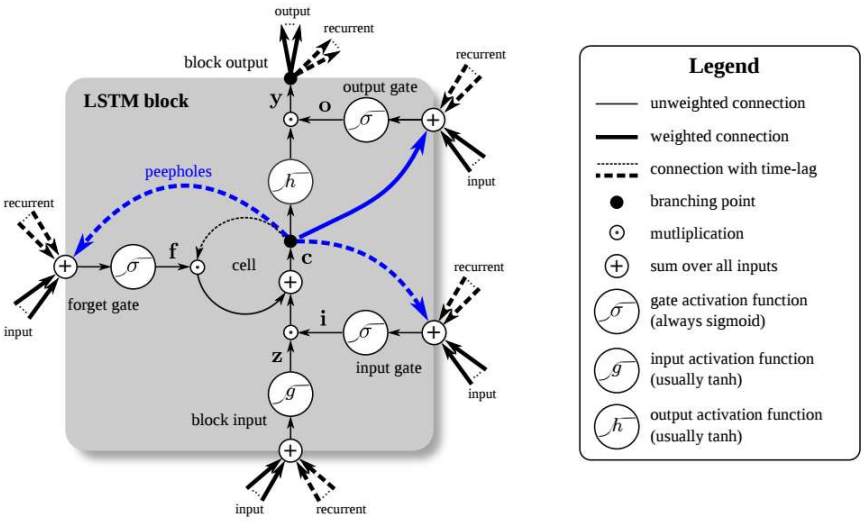


그림 출처: Nvidia Developer

○ LSTM 모형 알고리즘의 기본적인 구조는 다음과 같음

$$
\begin{aligned}
i_t &= \sigma(W_{xi}x_t + W_{hi}h_{t-1} + b_i) \\
f_t &= \sigma(W_{xf}x_t + W_{hf}h_{t-1} + b_f) \\
o_t &= \sigma(W_{xo}x_t + W_{ho}h_{t-1} + b_o) \\
g_t &= \tanh(W_{xg}x_t + W_{hg}h_{t-1} + b_g) \\
c_t &= f_t \odot c_{t-1} + i_t \odot g_t \\
y_t &= h_t = o_t \odot \tanh(c_t)
\end{aligned}
\qquad \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots (13)
$$

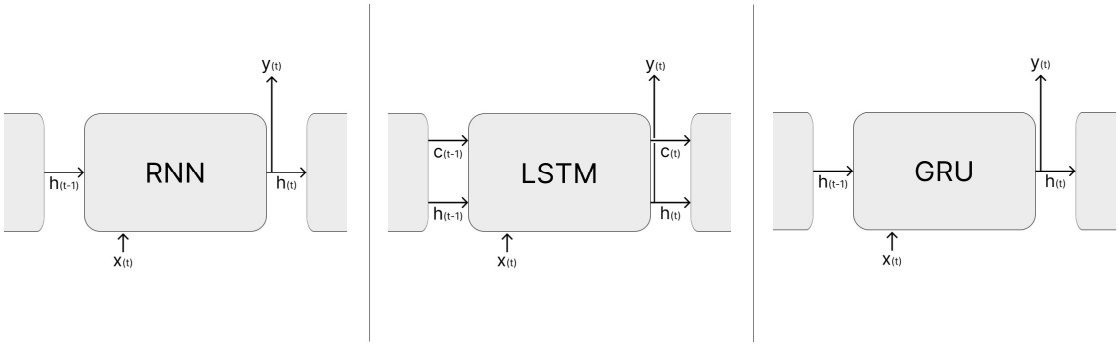
단, $\odot$는 element-wise multiplication operator, W는 가중치 행렬, σ는 sigmoid function, b는 각 층(layer)별 bias term.

❑ LSTM은 매개변수가 많아서 학습 속도가 느리며, 과적합(overfitting)[16] 문제가 발생하기 쉬움. 또한 Dropout[17]을 구현하기 상대적으로 어려우며, 그에 따라 결과 값이 민감하게 변하는 단점이 존재함

(4-4): GRU (Gated Recurrent Unit) 알고리즘 (Cho et al., 2014)

(4-4-1) 기존 신경망 모델과의 차이점

<그림 7> RNN, LSTM, GRU 구조 비교



❑ GRU는 게이트가 추가된 RNN 모델이라는 점에서 LSTM과 동일하지만, LSTM의 학습 시간이 오래 걸린다는 단점을 매개변수 개수를 줄이는 것을 통해 보완한 모델. LSTM 보다 구조가 상대적으로 간단해서 메모리 셀을 2개 사용하는 LSTM과 달리, GRU는 장기, 단기 과거 정보가 합쳐진 하나의 메모리 셀(memory cell, h_t)만 사용

16) 과적합에 대한 자세한 내용은 <참고 1> 참조.
17) Dropout은 과적합을 방지하는 규제화(regularization)의 하나로서, 규제화에 대한 자세한 내용은 <참고 3> 참조.

<참고 1: 과적합 (Overfitting)>

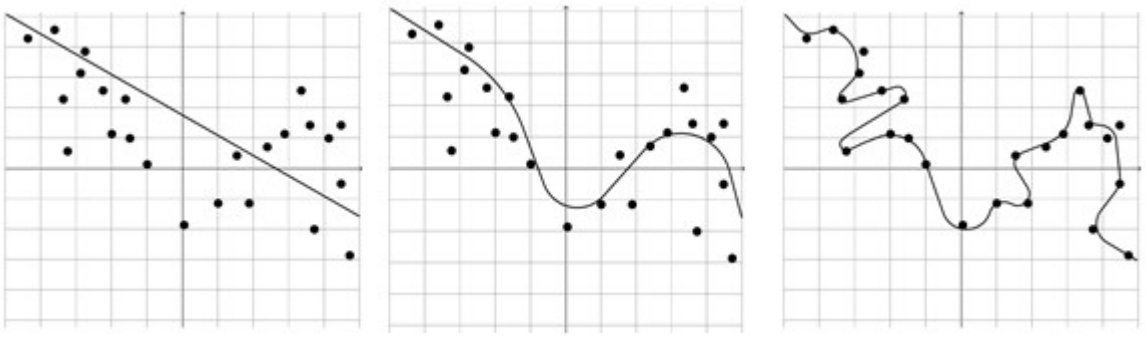
○ 학습 데이터(training data)에 치중하여 최적화된 상태로 모델의 범용성이 낮아지면 발생하는 문제로, 학습에 사용하지 않은 새로운 데이터에 대한 설명력이 낮아짐을 지칭

- 주로 학습 데이터가 적거나, 매개변수가 많고 유연하게 표현할 수 있는 모델은 과적합 문제가 발생할 가능성이 높음

○ <그림 8>의 중앙의 그래프가 가장 적합하게 학습된 예측 함수라고 라고 한다면, 우측의 그래프는 학습 데이터에만 치중한 예측 함수로, 학습 데이터(training data)에 대한 예측의 정확도는 아주 높지만, 새로운 시험데이터(test data)에 대한 예측력이 낮을 가능성이 매우 높음

- 좌측 그래프와 같이 주어진 데이터의 패턴에 대해 단순 선형함수로 예측함수를 상정한다면 학습 데이터와 시험 데이터에 대한 예측 정확성이 모두 낮을 가능성이 높고, 이를 경우 과소적합(underfitting)이라 지칭함

<그림 8> 과소적합(underfitting)과 과대적합(overfitting)



주: 왼쪽에서 오른쪽으로 이동할수록 과적합의 문제가 심화되는 경향을 나타냄

<참고 2: 기계학습에서의 데이터 구분>

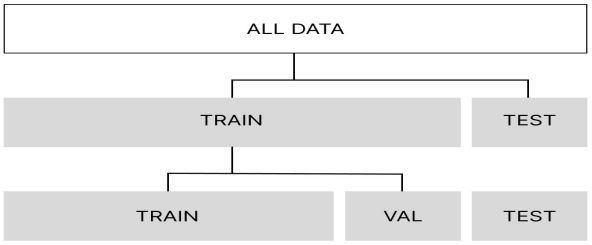
❑ 기계학습(machine learning) 과정에서는 학습 모델의 학습과 성능을 평가하기 위해 주어진 데이터를 여러 부분 집합으로 분할하여 학습 및 검증에 활용. 아래 <그림 9>와 같이 크게 3개의 하위 데이터 세트로 구분

○ 학습 데이터 (training data) : 본 데이터를 모델에 투입하여 학습을 진행하며, 반복적인 가중치 업데이트를 통해 최적의 파라미터 매개변수를 결정. 학습데이터는 검증 및 시험 데이터 사이의 교집합을 가지지 않음

○ 검증 데이터 (validation data) : 학습 데이터로 사용되지는 않지만, 학습 과정에 일부 관여하여 학습데이터로 학습된 모델의 성능을 검증하기 위한 데이터. 서로 다른 초매개변수(hyperparameter)를 가진 모델 중에서 가장 성능이 높은 모델을 선정하기 위한 목적

○ 시험 데이터 (test data) : 학습 과정에 관여하지 않으며 모형의 최종 성능 평가. 검증 데이터를 거쳐 선정된 모델이 최종적으로 적절한 모델인지 다시 한 번 테스트하는 것이 목적임. 새로운 데이터가 사용하거나 추가되었을 때, 범용적으로 예측하는 모델을 만들기 위해 범용성을 평가 목적. 일반적으로 데이터 분할 비율은 학습:검증:시험 = 6:2:2이 사용됨. 비유하자면 훈련 데이터는 교과서, 검증 데이터는 모의고사, 시험 데이터는 수능 시험으로 볼 수 있음

<그림 9> 학습, 시험, 검증 데이터 분리



<참고 3: 규제화(regularization)>

(1) 과적합 규제

❑ 학습 데이터에 지나치게 적합하게 되면 시험 데이터에 대해서 낮은 예측 성능을 보이고, 예측함수 내의 특정 변수에 대한 가중치(weight) 즉, 기울기 값이 커짐. 학습 과정시 학습 데이터의 실제 값과 예측 값의 차이인 손실 함수 값의 최소화뿐만 아니라, 과적합 또한 고려해야 함. 과적합을 개선하기 위해 학습모형에 존재하는 파라미터에 대한 일종의 제약을 부과하는 것을 규제화라고 일컬음. 규제를 적용한 손실함수는 아래 (식) 14와 같이 표현

$$Min(RSS(W) + \alpha \parallel W \parallel_p)$$

..... (14)

○ α는 규제 계수[18]으로 학습 데이터에 대한 적합 정도와 손실함수의 기울기 W의 크기 제어함. $\parallel W \parallel_p$는 가중치 벡터의 p-norm이며, 이 값이 커질수록 예측 함수가 비선형적인 복잡한 선으로 표현되므로 과적합 문제에 노출될 가능성이 커짐. p는 L_2과 L_1에 따라 정의됨

$$\parallel W \parallel_p = \left(\sum_{i=1}^{n} \mid W_i \mid^p \right)^{\frac{1}{p}}$$

..... (15)

○ 규제가 없거나 작게 설정하면, α와 $\alpha \parallel W \parallel_p$는 0 또는 0에 가까운 값이 되므로 RSS만 기존 손실함수와 동일하거나 비슷한 $Min(RSS(W))$가 됨. 하지만 규제를 크게 설정하면, 규제 계수인 α와 $\alpha \times \parallel W \parallel_p$은 ∞에 가까운 값이 되므로 최소화된 손실함수를 달성하기 위해서는 기울기 W를 작게 만들어야 함. 즉 기울기의 값이 지나치게 커지지 않도록 과적합을 제어

(2) L_2 규제(Ridge)와 L_1 규제(Lasso), $L_1 L_2$ 규제(Elastic Net)

❑ L_2 규제는 기울기의 값의 영향도를 줄이기 위해 α라는 매개변수를 통해 $\| W \|_p$에 패널티를 부여하는 방식. L_2 규제는 아래 식 (16)의 $(\| W \|_2)^2$ 항으로 정의되며 가중치 행렬의 제곱의 합 형태로, L_2의 기울기는 $2W$

$$Min(RSS(W) + \alpha(\| W \|_2)^2)$$

..... (16)

○ α는 주요 매개변수로 규제 강도를 결정. α 값에 따라 예측 성능이 민감하게 반응하는데, 적절한 α값은 실험적으로 탐색해야 하므로 시간 비용이 크다는 단점 존재. α의 값을 너무 크게 설정하면 W값이 0에 수렴하게 되어 언더피팅이 발생할 수 있으며, α의 값이 너무 작게 설정하면 오버피팅 개선에 대한 영향도가 미미함

❑ L_1 규제 또한 규제 계수를 적용하여 과적합을 개선한다는 점은 L_2 규제와 동일하지만, 제곱의 합인 $(\| W \|_2)^2$이 아닌 절댓값의 합 $\| W \|_1$이라는 점에서 다르며 L_1의 기울기는 $sign(W)$. 예측 영향도가 높은 [19]feature의 기울기는 1, 낮은 feature의 기울기는 0으로 결정하여 변수를 선택하는 방식

$$Min(RSS(W) + \alpha \| W \|_1)$$

..... (17)

○ L_2 규제와 L_1 규제에 따라 추정 결과가 달라지는데, L_2는 기울기가 전체적으로 절대값이 작게 추정되는 결과를 갖고, L_1는 대다수의 기울기는 0이 되지만 일부 파라미터는 0이 아닌 값이 되는 결과

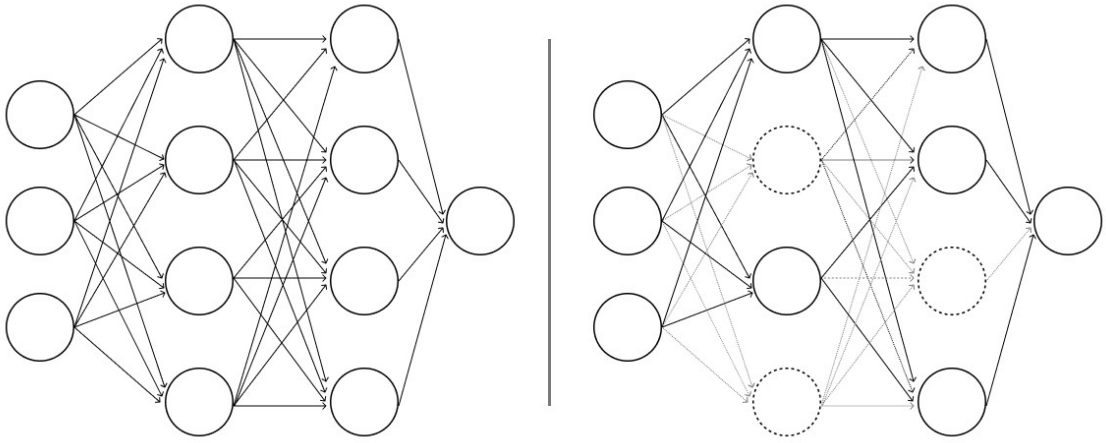
❑ $L_1 L_2$ 규제는 L_1 규제와 L_2 규제를 결합한 방식으로 L_1 규제에서 과감한 feature 선택을 완화하기 위해 L_2 규제를 추가. 이 방식은 feature 개수를 줄이면서 기울기 크기를 제어하기 때문에 주로 차원이 높은(즉, 설명변수의 개수가 많은) 데이터에 주로 사용됨

$$Min(RSS(W) + \alpha_2 (\| W \|_2)^2 + \alpha_1 \| W \|_1) \quad \cdots\cdots (18)$$

(3) 드롭아웃(Dropout)

❑ 신경망 모델에서 인접한 은닉층끼리 서로 연결되어 있는 노드들의 연결을 일정 확률에 따라 끊는 방식으로 과적합을 해결. 매 훈련마다 무시될 노드를 확률에 따라 무작위로 선정하고 나머지 뉴런에 신호를 전달하며 드롭아웃 비율은 일반적으로 10~50% 사이로 설정

<그림 10> 드롭아웃을 적용하지 않은 경우와 적용한 경우 작동 원리



(4) 순환신경망에서의 규제(Regularization)

❑ 순환 신경망 내 파라미터에 대한 규제를 적용해 과적합을 제어해 모델 성능 향상 목적. L_2와

L_1 규제를 기반으로 kernel, 편향(bias), 출력(activity)에 패널티를 적용되어 손실 함수에 영향을 미침

$$Min(\frac{1}{N}\sum_{i=1}^{N}(y_i-(w_0+w_1x_i))^2+\alpha \parallel W\parallel_p)$$

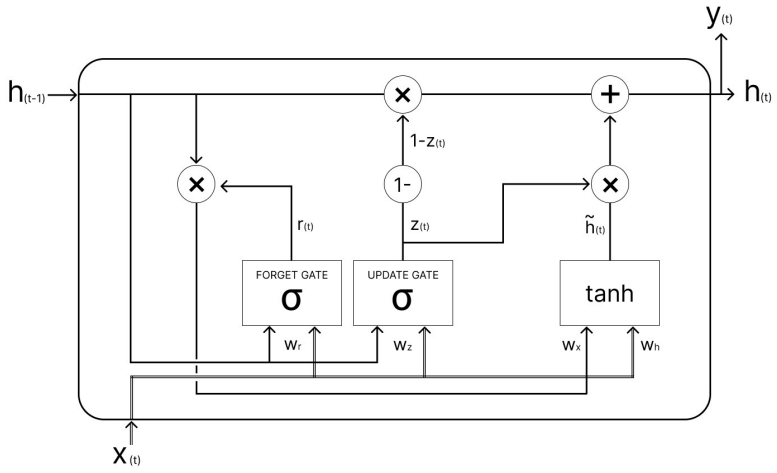
..... (19)

○ kernel regularizer는 kernel(커널)에 규제 계수를 추가하는 방식으로, 커널은 연결 가중치 행렬의 다른 표현이며 손실 함수의 기울기 w_1값의 크기 조정하고 w_0에는 영향을 미치지 않음

○ bias regularizer는 bias term을 규제하는 것으로, 구체적으로 위 식에서 절편인 w_0의 값을 0에 가깝게 제어하는 방식

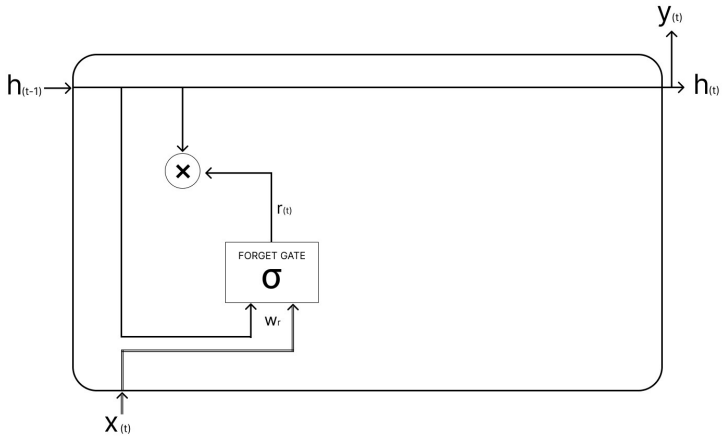
○ activity regularizer 는 층(layer)의 출력을 규제하여 $w_0+w_1x_i$ 값을 작게 조정하는 방식으로 출력값 자체를 0에 가깝게 정규화하기 위한 목적으로 사용

<그림 11> GRU 구조



(4-4-2) GRU 구조

<그림 12> GRU의 망각 게이트(forget gate) 구조

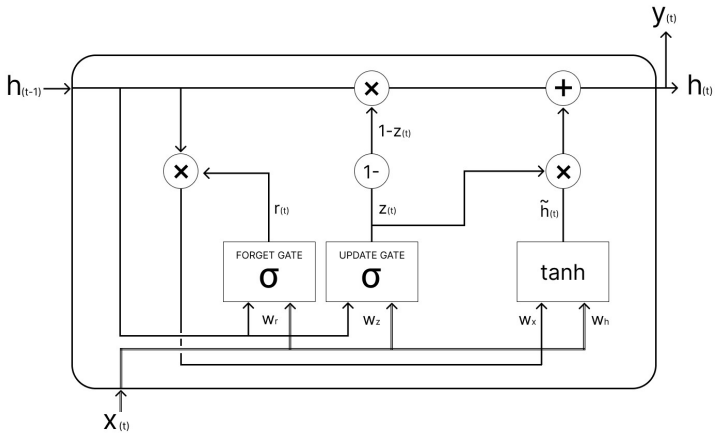


❑ 망각 게이트(forget gate)는 과거 기억정보(즉, 메모리 셀; h_{t-1}) 초기화 비율을 결정하는 게이트. 일부 과거 기억정보를 망각하기 위해 sigmoid 활성화 함수를 거침. 동 게이트에서 출력값이 0에 가까울수록 현재 시점 메모리 셀(h_t)에 이전 시점의 메모리(h_{t-1})는 적게 반영됨

$$r_t = \sigma(W_{xr} x_t + W_{hr} h_{t-1} + b_r) \quad \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots (20)$$

18) 정규화항, 벌칙항으로도 지칭된다.
19) 설명변수

<그림 13> GRU의 업데이트 게이트 구조



❑ 업데이트 게이트(update gate)는 LSTM의 망각 게이트와 입력 게이트가 결합된 형태로, 두 게이트의 기능을 제어기 z_t를 통해 수행함. 동 게이트 제어기의 출력 z_t에 따라, $(1-z_t)\odot h_{t-1}$ 를 통해 과거 기억정보를 일부 삭제, $z_t\odot\widetilde{h}_t$를 통해 현재 시점의 새로운 정보를 추가함

❑ $\widetilde{h}_t$는 현시점 메모리 셀 (h_t)이 최종출력되기 전의 임시 메모리 셀의 값으로, 망각 게이트를 통과한 과거정보(즉, $r_t\odot h_{t-1}$) 및 현재시점의 feature input을 이용하여 계산. h_t는 최종적으로 출력되는 현재 timestep의 메모리 셀로서, 이전 timestep의 메모리 셀과 $\widetilde{h}_t$의 가중평균임

$$z_t = \sigma(W_{xz}x_t + W_{hz}h_{t-1} + b_z) \qquad \cdots\cdots (21)$$

$$\widetilde{h}_t = \tanh(W_x x_t + W_h(r_t\odot h_{t-1}) + b)$$

$$h_t = (1-z_t)\odot h_{t-1} + z_t\odot\widetilde{h}_t \qquad \cdots\cdots (22)$$

※ 본 연구는 시계열 자료 예측에 특화되어 있는 LSTM 및 GRU 알고리즘을 모듈 3에 적용하여 GDP성장률 예측 모형에 적용[20)]

나. 분석 자료 및 학습자료(training data) 구조

(1) 분석 자료

❑ 연구에 사용한 데이터의 기간은 2001년부터 2020년까지로 한정함

○ 분석기간은 코로나-19 팬데믹 기간을 포함하고 있어 동 연구에서 구축한 모형이 거시경제적 위기 상황을 잘 포착할 수 있는지 검증

– 다만, 코로나 팬데믹 기간을 제외한 예측 결과도 포함하였음

❑ 예측모형에 사용된 주요 설명변수 집합은 기존 관련 연구 (e.g, 오현희, 2021; 김덕파·강성진, 2020)와 국내총생산과 연관성이 높을 것으로 기대되는 월별 자료를 포함[21)]

○ 모형에 사용된 설명변수의 세부 내용은 아래 <표 2>와 같음

<표 2> 예측모형에 사용된 설명 변수

변수명	변수 설명	LAG	DIFF	FREQ
PROD	산업생산지수 (전산업; 농림업제외)	2	1	M
PRODMAN	산업생산지수 (제조업)	2	1	M
PRODSER	산업생산지수 (서비스업)	2	1	M
PRODMIN	광업생산지수	2	1	M
MAFSHIP	제조업출하지수	2	1	M
MAFINVEN	제조업재고지수	2	1	M
INTENSITY	제조업가동률지수	2	0	M
LEADINDEX	경기선행지수 (순환변동치)	2	0	M
COINDEX	경기동행지수 (순환변동치)	2	0	M
RETAILSALES	소매판매지수	2	1	M
RETAILSALESD	소매판매지수 (내구재)	2	1	M
RETAILSALESND	소매판매지수 (비내구재)	2	1	M
INVESTIND	설비투자지수	2	1	M
CONSAMT	건설기성액 (불변)	2	0	M

20) 본 연구에서 LSTM 및 GRU는 Tensorflow Core v2.9.1 버전을 사용하여 구현함. 자세한 내용은 https://www.tensorflow.org/api_docs/python/tf/keras/layers/LSTM 및 https://www.tensorflow.org/api_docs/python/tf/keras/layers/GRU 참조.

21) 분기 자료를 포함한 모형은 [부록 1]을 참조.

CPI	소비자물가지수	1	1	M
PPI	생산자물가지수	1	1	M
IPI	수입물가지수	1	1	M
EXPI	수출물가지수	1	1	M
CPI_EXAGRI	소비자물가지수 (농산물석유제외)	1	1	M
CPI_EXFOODS	소비자물가지수 (식료품에너지제외)	1	1	M
EMP_NUM	취업자수	1	1	M
UEMPR	실업률	1	0	M
EMPR	고용률	1	0	M
ECONPART_RATE	경제활동참가율	1	0	M
M1END	M1말잔	2	1	M
M2END	M2말잔	2	1	M
LFEND	LF 말잔	2	1	M
EXP	수출금액 (통관기준)	1	1	M
IMP	수입금액 (통관기준)	1	1	M
BSI_SALES	매출 BSI (실적기준, 전산업)	0	0	M
BSI_ENV	업황 BSI (실적기준, 전산업)	0	0	M
BSI_MAF_EX	수출 BSI (제조업)	0	0	M
BSI_MAF_INTENSITY	가동률 BSI (제조업)	0	0	M
BSI_MAF_DOMESTIC	내수판매 BSI (제조업)	0	0	M
BSI_MAF_ENV	업황 BSI (제조업)	0	0	M
SENT	경제심리지수 (순환변동치)	0	0	M

주: DIFF = 로그 차분 여부. FREQ = 발표 주기. LAG = 발표시차

○ <표 2>의 「DIFF」는 예측 모형에 해당 변수가 사용될 때 차분(difference) 적용 여부를 나타내는 변수임. DIFF = 1이면 로그 차분(log difference) 후 예측 모형에 사용한 것을 나타내며, DIFF = 0이면 로그차분 없이 원자료 그대로 사용되었다는 것을 의미함

○ <표 2>의 「LAG」는 해당 자료의 발표 시차를 나타냄. 월말 기준으로 분기자료의 경우 2개월의 시차를 가지며, 월별 자료의 경우 발표시차가 최대 2개월에서 발표시차가 없는 변수까지 다양하게 포함됨

- 모형은 월말 기준 발표시차가 없는 설문 기반의 소비자신뢰지수(CSI) 및 기업경기실사지수(BSI) 변수들을 추가함

- 소비자신뢰지수(CSI)/기업경기실사지수(BSI)와 같은 심리기반의 변수들은 현재 및 향후 경기 방향에 대한 선제적인 정보를 제공하는 것으로 알려져 있고, 향후 소비자들의 지출행태 및 기업들의 투자지출에 대한 부분적인 정보를 가질 것으로 기대하여 모형에 포함[22)]

○ <표 2>의 「FREQ」는 해당 자료의 발표주기를 나타내며, M은 월별자료를 나타냄

○ 동 연구에서는 계절조정된(seasonally-adjusted) 자료를 사용하였으며, 계절조정된 원자료가 존재하지 않는 경우 X-11 알고리즘[23)]을 통해 계절조정을 거쳤음

(2) 인공신경망 학습 자료 (training data) 구조

❑ LSTM을 포함한 순환신경망에서 가장 핵심적인 요소라고 할 수 있는 것은 학습 과정에서 특정 시점의 목표변수를 설명하는데 있어서, 과거 어느 시점까지의 자료를 이용할 것인가라는 점임

○ 이는 관련 문헌에서 timestep이라고 지칭되는데, timestep의 크기가 크면 클수록 학습 자료 상에서 더 많은 과거 자료까지 이용하여 순환신경망을 학습한다는 의미

- <그림 5>에서 나타나있듯이 각 $t-1$ 시점의 timestep에서 인공신경망을 통과한 산출값이 다음 t시점의 timestep의 입력 값으로도 작용하기 때문에, timestep은 순환신경망의 정확도에 큰 영향을 미치는 요소임

○ 본 연구에서는 GDP성장률의 해당 분기 말월을 기준으로 각 설명변수의 과거 10개월 이전 값까지 포함하도록 학습 자료를 구축 (grid search를 통해 결정)

---

22) 학계에는 이에 대한 반론도 역시 존재함. 소비자신뢰지수는 현재 및 향후 경기흐름에 대한 적절한 정보를 제공하기에는 그 표본수가 제한적이고 과거 역사적 경험을 보면 소비자신뢰지수 하락기에도 소비자들의 지출이 줄어들지 않는 상황이 발생하였음. 가령, 과거 2020년 상반기 팬데믹 기간 동안 우리나라 소비자신뢰지수는 1분기에는 하락 후 2분기부터는 점차 상승세로 돌아섰지만, 실제 2분기에 최근 10년 내 최저 분기 성장률(-3%)을 기록함.

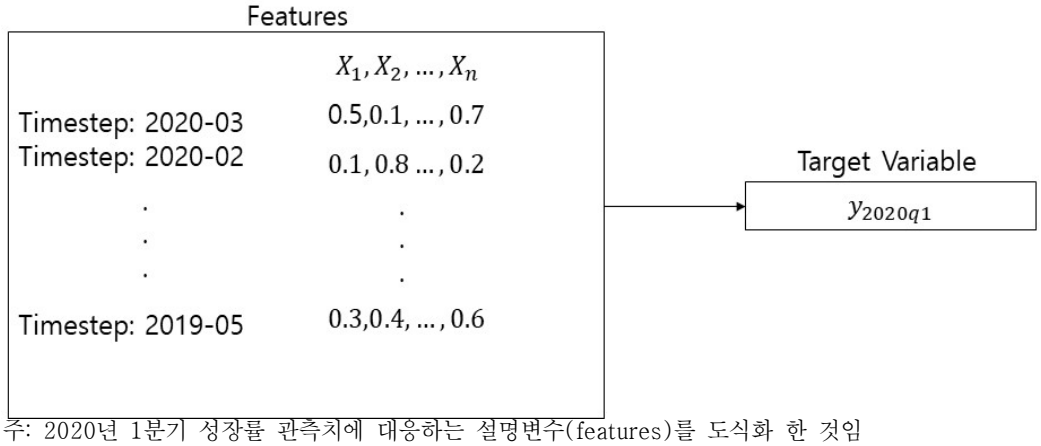
23) X-11 알고리즘의 기본 구조는 원 시계열을 트렌드(T), 계절성(S), 임의성(I) 등 세 가지 구성요소로 분해 및 추정하여 계절성 부분을 원시계열에서 제거.

- 가령, 2020년 6월말이 예측시점이라고 한다면, 관측 가능한 가장 최근의 GDP성장률 자료는 2020년 1분기 자료임

- 2020년 1분기의 말월인 3월을 기준으로 설명 변수들의 과거 10개월치 자료를 순환신경망의 학습과정에 입력(input)으로 사용

- 예측시점에서 관측 가능한 모든 분기별 GDP성장률에 해당하는 학습자료를 위와 같은 방식으로 구축

<그림 14> 순환신경망 학습자료(training data) 구조



주: 2020년 1분기 성장률 관측치에 대응하는 설명변수(features)를 도식화 한 것임

(3) 예측을 위한 새로운 입력 변수 생성

❑ 학습 자료를 바탕으로 인공신경망 학습이 종료되면 (즉, 인공신경망의 주요 파라미터들의 추정), 추정된 인공신경망 모형을 바탕으로 새롭게 주어진 입력 값에 대응하는 예측치를 산출

○ 즉, 예측치를 산출하기 위해서는 예측 대상이 되는 분기의 입력 변수에 대한 정보가 필요

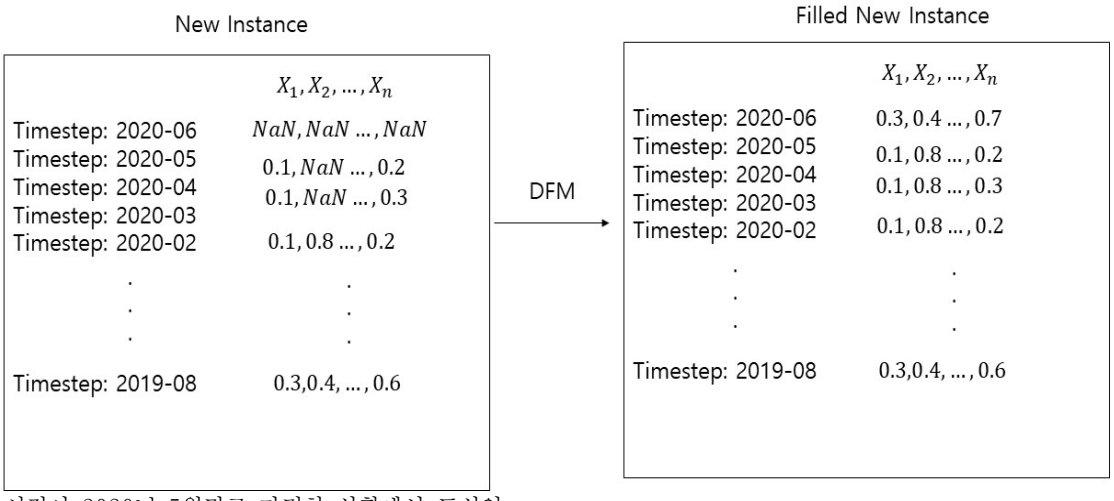
- 인공신경망 모형의 학습에 분기 말월을 기준으로 설명변수의 과거 10개월 치의 자료를 사용하였기 때문에, 예측대상이 되는 분기 말월까지의 월별자료의 입력값 생성 필요

- ragged-edge 문제로 인해 예측 대상 분기 말 월까지의 결측치 문제가 대두됨

- 동 연구에서는 예측에 사용할 새로운 입력 값의 결측치가 발생하는 경우, 모듈 2의 DFM 모형을 통해 추정하여 대용치(proxy)로 사용

- 가령, 아래 <그림 15>의 경우 현재 예측시점이 2020년 5월말이라고 가정했을 때, 해당 분기인 2020년 2분기 월말(6월)까지의 설명변수 값을 DFM으로 추정하여 새로운 입력 값(filled new instance)으로 사용

<그림 15> 예측을 위한 새 입력 변수(new instance) 생성 예시



주: 예측시점이 2020년 5월말로 가정한 상황에서 도식임.

(4) 인공신경망 구조 설정

❑ 심층신경망에는 대부분의 기계학습(machine learning) 알고리즘에서와 마찬가지로 분석자에 의해 결정되어야 하는 초매개변수(hyperparameters) 존재

○ 본 연구에서 상정한 순환신경망의 초매개변수는 다음과 같음

- 은닉층(hidden layer) 개수
- 각 은닉층 내 unit 개수
- 신경망의 주요 파라미터 업데이트를 위한 학습률(learning rate)

- 활성화 함수 (activation function)
- 규제화(regularization) 관련 초매개변수: 드랍아웃(dropout) 비율[24]

○ 아래 초매개변수의 공간에서 grid search를 통해 전체 샘플 기간(2001/01~2020/12)을 대상으로 한 검증 데이터(validation data)[25]에서 RMSE(Root-Mean-Squared-Error)를 최소화시키는 초매개변수를 선정 (<표 3>, <표 4> 참조)

- 은닉층 개수 공간: $H = \{1,2,3,4,5\}$
- 각 은닉층 내 unit 개수 공간: $U = \{5,6, \cdots, 100\}$
- 학습률 공간: $L = \{0.1, 0.01, 0.001, 0.0001\}$
- 활성화 함수 공간: $F = \{ReLU, \tanh, sigmoid\}$
- 드롭아웃(dropout) 비율 공간 : $D = \{0.0, 0.1, ..., 0.4\}$

<표 3> GRU 모형 초매개변수(hyperparameters) 설정값

초매개변수명	수치
은닉층 개수	2
첫 번째 은닉층 내 Unit 개수	10
첫 번째 은닉층 활성화 함수	sigmoid
첫 번째 은닉층 드롭아웃 비율	0.4
두 번째 은닉층 내 Unit 개수	75
두 번째 은닉층 활성화 함수	sigmoid
두 번째 은닉층 드롭아웃 비율	0.4
학습률	0.1

24) 본 연구에서는 신경망의 학습시 L2 규제화(regularization)를 적용하였으며, 드롭아웃(dropout) 비율은 초매개변수에 대한 grid search를 통해 설정함.
25) 검증 데이터의 비율은 0.25로 설정함.

<표 4> LSTM 모형 초매개변수(hyperparameters) 설정값

초매개변수명	수치
은닉층 개수	2
첫 번째 은닉층 내 Unit 개수	8
첫 번째 은닉층 활성화 함수	sigmoid
첫 번째 은닉층 드롭아웃 비율	0.2
두 번째 은닉층 내 Unit 개수	21
두 번째 은닉층 활성화 함수	sigmoid
두 번째 은닉층 드롭아웃 비율	0.4
학습률	0.1

○ Tensorflow 내 신경망 학습의 세부 알고리즘에는 임의성(randomness)이 일부 작용하기 때문에, 동일한 데이터와 신경망 구조 하에서라도 학습마다 조금씩 상이한 결과치가 도출될 수 있음

- 딥러닝의 실제 적용시 학습 과정을 반복적으로 실행하여, 이러한 예측치들의 경험적 분포(empirical distribution)를 통해 최종적인 예측치 도출하는 것이 바람직함 (가령 경험적 분포 기반의 평균 또는 중위값 등)

- 본 연구는 시간적 제약 및 결과의 재현성(reproducibility)을 위해 모든 학습 과정에 동일한 random seed를 부여한 단일 시행 시뮬레이션 결과를 담고 있음

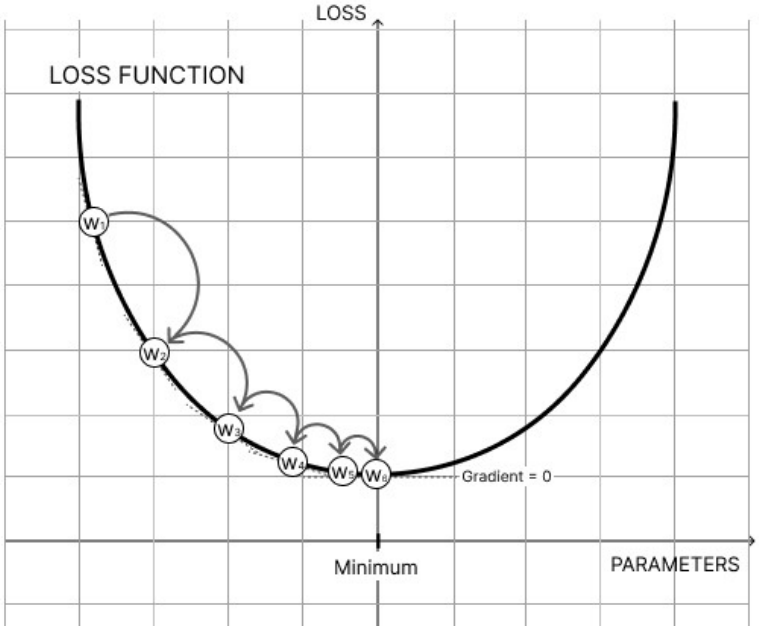
<참고 5: 학습률 (learning rate)>

❑ 학습률이란 학습과정에서 모형의 예측 오차를 얼마만큼 빠르게 학습모형의 매개변수에 반영하여 업데이트 해나갈지를 결정하는 매개변수임

❑ 기계학습 과정에서 모형 내 최적의 가중치를 찾아나가는 방식은 경사하강법(gradient descent)이 사용됨

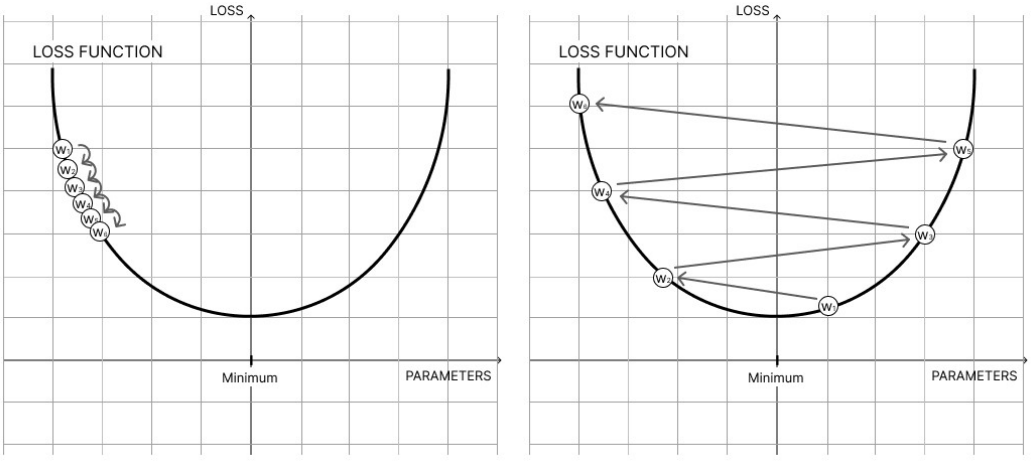
❑ 경사 하강법은 최적화 알고리즘으로서, 손실함수(loss function)를 최소화하기 위해 반복해서 매개변수(parameter)[26]를 업데이트하는 과정임. 손실함수[27]를 최소화하는 매개변수를 찾기 위한 가장 빠른 길인 그래디언트(gradient) 방향을 탐색

<그림 16> 경사 하강법 (Gradient Descent) 과정



○ 위 그래프에서 보면, 먼저 매개변수 벡터에 대한 손실함수의 현재(w_1) 그래디언트를 계산하고 그래디언트가 감소하는 방향으로 이동하여, 그래디언트가 0이 되는 최솟값 지점(w_6) 도달함. 다음 스텝으로 넘어갈 때의 크기를 학습률(learning rate)라고 하고, 학습률이 클수록 더 크게 이동함

<그림 17> 학습률에 따른 경사 하강법 과정



○ 학습률을 너무 작게 설정하면 최솟값 탐색까지 학습을 반복해야 하므로 시간이 오래 걸리며, 너무 크게 설정하면 수렴이 아닌 발산하게 되어 최솟값을 지나쳐 탐색에 실패할 수 있음

❑ 손실 함수 최소화가 목적이므로, 손실 함수로부터 경사 하강법 수식 전개. 그래디언트를 구하기 위해 손실 함수를 기울기와 편향인 파라미터 θ에 대해 편미분 계산

$$RSS(\theta) = \frac{1}{N}\sum_{i=1}^{N}(y_i - \theta x_i)^2$$ 28)

$$\frac{\partial}{\partial w_1} RSS(\theta) = \frac{2}{N}\sum_{i=1}^{N}(y_i - \theta x_i)x_i$$

..... (23)

○ 위의 편도 함수식을 모든 파라미터에 대한 편미분을 한번에 계산하기 위해 벡터로 표현

$$\nabla_{\theta} RSS = \begin{bmatrix} \frac{\partial}{\partial_{\theta_0}} RSS(\theta) \\ \frac{\partial}{\partial_{\theta_1}} RSS(\theta) \\ \vdots \\ \frac{\partial}{\partial_{\theta_n}} RSS(\theta) \end{bmatrix} = \frac{2}{N} X^T (y - \theta X)$$ [29]

..... (24)

○ 경사하강법 공식은 아래와 같이 표현

$$\theta_{i+1} = \theta_i - a \nabla_{\theta} RSS(\theta) = \theta_i - \frac{2}{N} a X^T (y - \theta X)$$

..... (25)

○ θ_i는 주어진 현재값으로, 초기값은 θ_0이 됨. $a \nabla_{\theta} RSS(\theta)$ 은 탐색방향이며, a는 탐색방향으로 얼마나 θ_i를 업데이트할 것인지 결정하는 학습률. θ_{i+1}는 업데이트된 값으로, 업데이트를 통해 최소 손실 함숫값을 갖는 θ_{i+1}을 찾는 원리로 작동함

26) 매개변수의 대표적인 예시로는 가중치, 편향 등이 있다.
27) 비용함수의 대표적인 예시로는 MSE, Cross-entropy 등이 있다. 손실함수와 동일하다.
28) 단순화를 위해 기울기와 편향인 파라미터 θ로 치환한다.
29) T는 전치를 의미한다.

다. 예측력 비교 검증

(1) 벤치마크 모형

❑ LSTM 및 GRU 인공신경망 기반 예측 모형과 아래 주요 벤치마크(benchmark) 모형들의 예측력을 비교·검증

① AR (Autoregressive) Model

- 예측시점별 GDP성장률에 대한 분기 자료를 바탕으로 AR(p) 모형을 설정하여 예측치 산출. 단, 예측치 산출에 있어서 초월, 중월 및 말월에 따라 관측할 수 있는 과거 GDP성장률에 대한 정보가 상이한 점을 고려하여 예측치를 산출

- 가령, 발표 시차로 인해 초월/중월의 경우 지난 분기 GDP성장률에 대한 정보가 존재하지 않기 때문에 당분기 경제성장률을 산출하기 위해서는 다음 분기의 예측치를 산출하도록 설정함

- AR 모형의 lag order는 1로 설정

② Dynamic Factor Model (DFM)

- [부록 1]의 <표 27>의 설명변수에 분기별 GDP 성장률 자료를 추가하여 DFM 모형을 구축 후, 각 예측시점별로 GDP성장률에 대한 전망치를 산출

- [부록 1]의 혼합주기 모형은 Factor loading 구조상 GDP 성장률의 하위 그룹은 다른 분기 변수와 동일하게 'REAL(실물)'로 설정하였음

(2) 표본 외 예측

❑ 표본 외 예측(out-of-sample forecasts)을 통해 개발된 모형의 예측력을 벤치마크 모형들과 비교함

○ 예측력 평가기준은 관련 문헌에서 광범위하게 사용되고 있는, 제곱근평균자승오차(Root-Mean-Squared-Error; RMSE)와 평균절대오차(Mean-Absolute-Error; MAE) 두 가지 기준을 사용

$$RMSE = \sqrt{\frac{\sum_{t=1}^{T}(\hat{y}_t - y_t)^2}{T}}$$

$$MAE = \frac{\sum_{t=1}^{T}\left|\hat{y}_t - y_t\right|}{T}$$

····· (26)

단, T는 표본 외 예측 기간의 샘플 사이즈, $\hat{y}_t$는 예측치.

○ 예측 시점별(초월, 중월, 말월) RMSE와 MAE를 시산하여 정보 양의 변화에 따른 예측력 변화 양상을 분석

○ 표본 외 예측기간은 2015년 1월부터 2020년 12월까지로 설정함

- 축차 예측(recursive forecasts; RF)과 롤링 윈도우 예측(rolling window forecasts; RWF) 두 가지 방식을 사용하여 표본 외 예측(out-of-sample forecasts)의 RMSE와 MAE를 시산

- RWF는 딥러닝 모형의 학습에 사용할 학습 데이터(training data) 샘플의 크기를

계속 일정하게 유지한 상태에서 표본 외 예측을 해나가는 것인 반면, RF는 학습 데이터의 크기가 예측 시점이 미래로 갈수록 계속 확대되는 특징을 가지고 있음

- 시계열 자료의 구조적 특징(가령, stationarity 여부)에 따라 두 가지 예측 방식의 정확도가 달라질 수 있기 때문에 두 가지 방식을 비교 (기존 관련 연구에서는 롤링 윈도우 예측방식이 전망성과를 개선시키는 것으로 나타났음; Swanson and White, 1997; Tashman, 2000, 최병재·한진현, 2014)

3. 주요 결과

(1) 코로나-19 기간(2020년) 제외 표본 외 예측 결과

(RMSE 기준)

❑ 당분기 GDP성장률 표본 외 예측에서 축차예측 기반 LSTM 모형의 RMSE는 약 0.48, GRU의 RMSE는 0.58를 나타냄 (<표 9> 참조)

○ 롤링윈도우 예측의 경우 GRU의 RMSE는 약 0.54, LSTM의 RMSE는 약 0.48을 기록하여, 전반적으로 당분기 경제성장률 예측에는 LSTM 기반 모형이 상대적으로 우수한 것으로 평가됨

- 당분기 성장률 예측오차의 제곱항에 대한 Diebold and Mariano (1995) 검정 결과를 보면 GRU 모형과 DFM 모형 간 유의한 차이가 존재하나, LSTM과 DFM 간의 유의한 차이는 존재하지 않음

- LSTM 모형과 DFM 모형은 전 예측시계(forecasting horizon)에 걸쳐 예측력에 통계적으로 유의미한 차이를 보이지 않음

- 단, LSTM 모형의 축차예측은 당분기 및 1분기 후 경제성장률 예측력에 있어서 AR(1) 모형과 유의미한 차이를 보임

○ 모든 예측 모형에 있어서 예측 시계가 길어질수록 예측오차가 확대되는 경향을 보이고 있으며, 3분기 이후 GDP성장률에 대한 예측오차는 AR모형과도 큰 차이를 보이지 않는 것으로 관찰됨

- 단, LSTM 모형은 축차 예측에서 각 예측시점별 상대적으로 먼 예측시계에서도 AR(1) 모형과 통계적으로 유의미한 차이를 보임

- GRU 모형의 경우 예측시점별로 벤치마크 모형과 예측력에 있어서의 통계적으로 유의한 차이가 존재하나, 예측력이 악화되는 방향으로 예측력의 차이를 보임

○ GRU, LSTM 기반 모두 예측 시점(초월, 중월, 말월)에 따라 예측 오차가 크게 변동하지 않는 것으로 관찰됨

- 단, 예측시점이 말월인 경우 LSTM, GRU 모두 예측오차가 중월 시점의 예측에 비해 더욱 확대되는 경향을 보이고 있음

(MAE 기준)

❑ <u>당분기 GDP성장률</u> 표본 외 예측에서 전반적으로 LSTM 모형의 축차예측 MAE는 약 0.38, GRU의 MAE는 약 0.49를 나타냄 (<표 14>참조)

○ 롤링윈도우 예측의 경우 GRU 모형의 MAE는 약 0.44, LSTM 모형의 MAE는 약 0.37을 기록했으며, LSTM 모형은 AR(1) 모형과 예측력에 통계적으로 유의미한 차이가 존재하나 GRU모형은 예측력이 악화되는 방향으로 예측력의 정확성에 차이가 존재함

- 단, LSTM 모형의 경우 MAE 기준 DFM 모형의 예측력과 통계적으로 유의미한 차이를 보이고 있지 않음 (3분기 후 롤링윈도우 예측치 제외)

- GRU 기반 예측의 경우 벤치마크 모형인 DFM의 예측오차보다 전 예측시계에 걸쳐 상대적으로 더욱 높게 나타나는 경향이 관찰되며, 축차예측의 경우 당분기, 2분기 후, 3분기 후 성장률 예측치에 있어서 DFM과 예측력에 있어서 유의한 차이를 보임

- LSTM 기반 예측의 경우 벤치마크 모형인 DFM의 예측오차와 전 예측시계에 걸쳐 크게 다르게 나타나지 않은 것으로 관찰되지만, 3분기 후 성장률 예측치에는 DFM 대비 예측력에 유의한 차이를 보임

○ 모든 예측 모형에 있어서 예측 시계가 길어질수록 예측오차가 확대되는 경향을 보이고 있으며, GRU 기반 예측의 경우 롤링윈도우 예측에서 미래 시점일수록 AR(1) 모형보다 더 큰 예측 오차를 나타냄

- 3분기 후 성장률에 대한 축차예측의 경우 AR(1) 및 DFM과의 유의한 차이를 보임

○ 예측 시점(초월, 중월, 말월)에 따라 예측 오차가 크게 변동하지 않는 것으로 관찰됨

- RMSE 경우와 마찬가지로, 예측시점이 말월인 경우 LSTM, GRU 모두 예측오차가 중월 시점의 예측에 비해 더욱 확대되는 경향을 보이고 있음

- LSTM 모형의 경우, 각 예측시점별로 당분기 및 1분기 후 성장률 예측에 있어서 AR(1) 모형 대비 예측력을 개선시키는 방향의 결과를 보여주고 있음

※ 코로나-19 기간(2020년)을 제외한 표본 외 예측에서 딥러닝 기반 모형의 예측오차와 DFM 모형 간의 예측오차에 있어서 유의적인 차이를 보이고 있지 않지만, LSTM 모형이 GRU 모형에 비해 상대적으로 예측력이 우수한 것으로 나타남. 따라서 GRU 기반 모형보다 LSTM 기반의 예측모형을 예측에 활용하는 것이 더욱 적절해 보임

(2) 코로나-19 기간(2020년) 포함 표본 외 예측 결과

(RMSE 기준)

❑ 코로나-19 기간을 포함한 <u>당분기 GDP성장률</u> 표본 외 예측에서 전반적으로 축차예측 기반 LSTM 모형의 RMSE는 약 0.99, GRU의 RMSE는 약 0.87를 나타냄 (<표 19> 참조)

○ 롤링윈도우 예측의 경우 GRU의 RMSE는 약 0.92, LSTM의 RMSE는 약 0.98을 나타냄

- 다만, LSTM의 경우 1분기 이후 성장률 예측에 있어서 모형 간에 예측오차에 큰 차이가 존재하지 않지만, 당분기 성장률 축차예측에 있어서 AR(1) 모형과 유의한 차이가 존재함

- 즉, 코로나-19 (2020년) 기간을 포함하면 모형간의 예측력에 큰 차이는 존재하지 않는 것으로 평가됨

- GRU 모형은 AR(1) 모형보다 예측오차가 컸으나, 두 모형 간 예측오차 차이에

통계적 유의성은 발견되지 않음

○ 당분기 성장률 예측의 경우, 예측 시점이 말월로 진행될수록 GRU 모형 및 LSTM 모형의 예측오차가 낮아지는 경향이 존재함

- 초월 축차예측 기준, LSTM 모형의 3분기 이후 성장률에 대한 예측력은 AR모형과 대비해서 유의미한 차이를 보이고 있고, 전반적으로 딥러닝 기반 모형과 벤치마크 모형 간의 예측력에 있어서 유의미한 차이를 보이고 있지 않음

- 말월 롤링윈도우 예측 기준, GRU 모형의 1분기 이후 성장률에 대한 예측력은 DFM모형과 대비해서 유의미한 차이를 보이고 있지만, RMSE 차이는 크지 않음

(MAE 기준)

❑ <u>당분기 GDP성장률</u> 표본 외 예측에서 전반적으로 축차예측 기준 LSTM 모형의 MAE는 약 0.63, GRU의 MAE는 약 0.65를 나타냄 (<표 24> 참조)

○ 롤링윈도우 예측의 경우 GRU 모형의 MAE는 약 0.65, LSTM의 MAE는 약 0.64을 기록하였으며, GRU 기반 모형과 LSTM 모형과의 예측성과가 유사함. 또한 롤링윈도우 예측과 축차예측의 예측 오차도 큰 차이가 존재하지 않음

- 다만, LSTM 모형 축차 예측의 경우 GRU 및 DFM 기반 모형에 비해 예측 시계가 길어질수록 가장 낮은 예측 오차를 기록하였으며, 전 예측 시계에 걸쳐 0.62~0.68 수준의 안정적인 예측성과를 보여주고 있음

- LSTM 모형은 벤치마크 모형 대비 통계적으로 유의한 차이는 당분기 및 3분기 이후 예측치에서만 발견할 수 있음

- GRU 기반 예측 모형의 경우 1분기 이후 예측에서는 0.76 수준의 MAE를 보여주고 있고, 벤치마크 모형의 예측력과 통계적으로 유의한 차이를 만들어내지 못함

○ 예측시점이 초월에서 말월로 갈수록 GRU 모형 및 LSTM 모형의 예측오차가 낮아지는

경향이 존재함

○ 초월 기준 축차예측의 경우, LSTM은 전 예측 시계에 걸쳐 0.64~ 0.68 수준의 안정적인 MAE를 기록하였으며, GRU 모형은 1분기 이후 성장률 예측에서 가장 높은 MAE를 기록하는 것으로 나타남

- 다만, 예측모형 간 초월 기준 예측오차 차이에는 통계적 유의성은 발견되지 않음

○ 중월 기준 축차예측의 경우, LSTM 기반 모형은 전 예측 시계에 걸쳐 0.62~ 0.68 수준의 MAE를 보이고 있으며, GRU 기반 모형은 1분기 이후 성장률 예측에서 다른 예측모형과 비교하여 가장 낮은 예측성과를 보여주는 것으로 관찰됨

- 중월 기준, 당분기 성장률 예측에 있어서 DFM의 예측성과가 가장 우수한 것으로 나오나, LSTM 모형과의 예측력 차이에 통계적 유의성은 발견되지 않음

○ 말월 축차예측 기준, LSTM 모형은 전 예측 시계에 걸쳐 0.61~ 0.68 수준의 MAE를 보이고 있으며, GRU 모형은 0.60~0.80 범위의 MAE를 보여줌

- 단, GRU 모형의 예측력은 벤치마크 모형과 비교해서 유의적인 차이가 없으나, LSTM 모형의 당분기, 1분기 후, 2분기 후 성장률 예측오차는 AR모형과 유의적인 차이를 나타냄

- 말월 롤링윈도우 예측 기준, LSTM 모형의 당분기 성장률 예측력은 AR(1) 모형과 비교하여 우수한 것으로 평가됨

※ 코로나-19 기간(2020년)을 포함한 표본 외 당분기 성장률 예측에서 DFM 모형과 LSTM 모형 간의 예측력에 차이는 존재하지 않지만, LSTM은 AR모형에 비해 예측력이 상대적으로 좋은 것으로 평가됨. 또한 롤링윈도우 예측 기준, 예측시계가 멀어질수록 LSTM 기반 예측 모형의 예측성과가 DFM 모형 대비 더욱 우수한 것으로 나타남

<표 5> GDP 성장률 표본 외 예측 (2020년도 제외): RMSE

(a) 초월

<표 6> 초월 기준 RMSE (2020년도 제외)

	GRU	LSTM	DFM	AR(1)
축차 예측				
y(t)	0.6069**	0.4767	0.5416	0.5124
y(t+1)	0.7126*	0.6411**	0.6396	0.7299
y(t+2)	1.0745*	1.0551	1.0301	1.1408
y(t+3)	1.1628	1.1098**	1.0995	1.1771
롤링윈도우 예측				
y(t)	0.5567	0.4784	0.5222	0.4912
y(t+1)	0.6444	0.6098**	0.6278	0.6944
y(t+2)	0.9937	1.0078	1.0005	1.1010
y(t+3)	1.1144	1.0822**	1.0835	1.1464

(b) 중월

<표 7> 중월 기준 RMSE (2020년도 제외)

	GRU	LSTM	DFM	AR(1)
축차 예측				
y(t)	0.5758*	0.4665	0.3926	0.5124
y(t+1)	0.6983	0.6441**	0.6468	0.7299
y(t+2)	1.0743*	1.0541	1.0191	1.1408
y(t+3)	1.1611	1.1098**	1.1086	1.1771
롤링윈도우 예측				
y(t)	0.5109	0.4558	0.4005	0.4912
y(t+1)	0.6567	0.6424**	0.6394	0.6944
y(t+2)	1.0164	1.0273	0.9879	1.1010
y(t+3)	1.0993	1.0752	1.0980	1.1464

(c) 말월

<표 8> 말월 기준 RMSE (2020년도 제외)

	GRU	LSTM	DFM	AR(1)
축차 예측				
y(t)	0.5695	0.4834**	0.4977	0.5737
y(t+1)	0.7172	0.6445**	0.6673	0.7115

y(t+2)	1.0527	1.0438	1.0151	1.1376
y(t+3)	1.1532*	1.1116**	1.1165	1.1776
롤링윈도우 예측				
y(t)	0.5485	0.4921	0.4823	0.5401
y(t+1)	0.6604	0.6265	0.6697	0.6863
y(t+2)	0.9943	0.9828	0.9818	1.0993
y(t+3)	1.0948	1.0945	1.0941	1.1470

(d) 종합

<표 9> 종합 RMSE (2020년도 제외)

	GRU	LSTM	DFM	AR(1)
축차 예측				
y(t)	0.5843*	0.4756**	0.4814	0.5336
y(t+1)	0.7094	0.6432**	0.6514	0.7238
y(t+2)	1.0672*	1.0510	1.0214	1.1398
y(t+3)	1.1590	1.1104	1.1082	1.1773
롤링윈도우 예측				
y(t)	0.5391*	0.4757	0.4710	0.5080
y(t+1)	0.6539	0.6264	0.6459	0.6917
y(t+2)	1.0015	1.0061	0.9901	1.1005
y(t+3)	1.1028	1.0840	1.0919	1.1466

주1: GRU: Gated Recurrent Unit. LSTM: Long-Short Term Memory. DFM = Dynamic Factor Model, AR = Autoregressive Model.

주2: *: Diebold-Mariano (1995) 검정 결과 DFM 모형과의 예측오차 제곱항의 차이가 10% 유의수준에서 유의함을 지칭.

주3: **: Diebold-Mariano (1995) 검정 결과 AR(1) 모형과의 예측오차 제곱항의 차이가 10% 유의수준에서 유의함을 지칭.

주4: ***: Diebold-Mariano (1995) 검정 결과 DFM 및 AR(1) 모형과의 예측오차 제곱항의 차이가 10% 유의수준에서 유의함을 지칭.

<표 10> GDP 성장률 표본 외 예측 (2020년도 제외): MAE

(a) 초월

<표 11> 초월 기준 MAE (2020년도 제외)

	GRU	LSTM	DFM	AR(1)
축차 예측				
y(t)	0.5146**	0.3776	0.4151	0.4106
y(t+1)	0.5456*	0.4620***	0.4759	0.5318
y(t+2)	0.6969*	0.6456**	0.6280	0.7063

y(t+3)	0.7762	0.6920**	0.7204	0.7554
롤링윈도우 예측				
y(t)	0.4662	0.3745	0.4057	0.4028
y(t+1)	0.4919	0.4466***	0.4962	0.5115
y(t+2)	0.6230**	0.6323**	0.6227	0.6810
y(t+3)	0.7686	0.6761**	0.7114	0.7350

(b) 중월

<표 12> 중월 기준 MAE (2020년도 제외)

	GRU	LSTM	DFM	AR(1)
축차 예측				
y(t)	0.4844*	0.3716	0.3269	0.4106
y(t+1)	0.5356	0.4672**	0.4806	0.5318
y(t+2)	0.7099*	0.6452***	0.6075	0.7063
y(t+3)	0.7606	0.6855**	0.7006	0.7554
롤링윈도우 예측				
y(t)	0.3992	0.3562	0.3451	0.4028
y(t+1)	0.4939	0.4691**	0.4918	0.5115
y(t+2)	0.6503*	0.6305**	0.5953	0.6810
y(t+3)	0.7272	0.6686***	0.7173	0.7350

(c) 말월

<표 13> 말월 기준 MAE (2020년도 제외)

	GRU	LSTM	DFM	AR(1)
축차 예측				
y(t)	0.4680	0.3762**	0.3742	0.4838
y(t+1)	0.5695**	0.4669**	0.5013	0.5128
y(t+2)	0.6645*	0.6339***	0.6060	0.7028
y(t+3)	0.7796*	0.6955	0.7174	0.7557
롤링윈도우 예측				
y(t)	0.4454	0.3745**	0.3614	0.4545
y(t+1)	0.4915	0.4587***	0.5307	0.5025
y(t+2)	0.6242	0.6081***	0.5869	0.6770
y(t+3)	0.7165	0.7053	0.7319	0.7368

(d) 종합

<표 14> 종합 MAE (2020년도 제외)

	GRU	LSTM	DFM	AR(1)
축차 예측				
y(t)	0.4890***	0.3751**	0.3721	0.4350
y(t+1)	0.5502	0.4654	0.4859	0.5255
y(t+2)	0.6904*	0.6416*	0.6138	0.7051
y(t+3)	0.7721***	0.6910**	0.7128	0.7555
롤링윈도우 예측				
y(t)	0.4369*	0.3684**	0.3707	0.4201
y(t+1)	0.4924	0.4581	0.5062	0.5085
y(t+2)	0.6325	0.6236	0.6016	0.6797
y(t+3)	0.7375	0.6833***	0.7202	0.7356

주1: GRU: Gated Recurrent Unit. LSTM: Long-Short Term Memory. DFM = Dynamic Factor Model, AR = Autoregressive Model.

주2: *: Diebold-Mariano (1995) 검정 결과 DFM 모형과의 예측오차 절대값의 차이가 10% 유의수준에서 유의함을 지칭.

주3: **: Diebold-Mariano (1995) 검정 결과 AR(1) 모형과의 예측오차 절대값의 차이가 10% 유의수준에서 유의함을 지칭.

주4: ***: Diebold-Mariano (1995) 검정 결과 DFM 및 AR(1) 모형과의 예측오차 절대값의 차이가 10% 유의수준에서 유의함을 지칭.

<표 15> GDP 성장률 표본 외 예측 (2020년도 포함): RMSE

(a) 초월

<표 16> 초월 기준 RMSE (2020년도 포함)

	GRU	LSTM	DFM	AR(1)
축차 예측				
y(t)	0.9355	1.0194	0.9846	1.1000
y(t+1)	1.2106	1.0414	1.1859	1.1025
y(t+2)	1.1242	1.0511	1.1584	1.0910
y(t+3)	1.1216	1.0472**	1.1226	1.0940
롤링윈도우 예측				
y(t)	0.9714	1.0347	0.9824	1.0642
y(t+1)	1.2456	1.0726	1.1983	1.0762
y(t+2)	1.1270	1.0732	1.1609	1.0695
y(t+3)	1.1150	1.0417	1.1363	1.0685

(b) 중월

<표 17> 중월 기준 RMSE (2020년도 포함)

	GRU	LSTM	DFM	AR(1)
축차 예측				
y(t)	0.8567	0.9930	0.8081	1.1000
y(t+1)	1.1718	1.0451	1.0868	1.1025
y(t+2)	1.1070*	1.0508	1.0803	1.0910
y(t+3)	1.1472	1.0540	1.0912	1.0940
롤링윈도우 예측				
y(t)	0.9620*	0.9523	0.7919	1.0642
y(t+1)	1.0666	1.0442	1.0904	1.0762
y(t+2)	1.0473	1.0640	1.0865	1.0695
y(t+3)	1.0520	1.0341	1.1047	1.0685

(c) 말월

<표 18> 말월 기준 RMSE (2020년도 포함)

	GRU	LSTM	DFM	AR(1)
축차 예측				
y(t)	0.8030**	0.9660	0.7490	1.1523
y(t+1)	1.0774	1.0415	1.1527	1.1098
y(t+2)	1.1368	1.0358**	1.1561	1.0976
y(t+3)	1.1434	1.0579	1.1647	1.0976
롤링윈도우 예측				
y(t)	0.8220	0.9587	0.7619	1.1327
y(t+1)	1.1024*	1.0927	1.1500	1.0818
y(t+2)	1.1645	1.0735	1.1741	1.0752
y(t+3)	1.1295	1.0816	1.1761	1.0723

(d) 종합

<표 19> 종합 RMSE (2020년도 포함)

	GRU	LSTM	DFM	AR(1)
축차 예측				
y(t)	0.8668	0.9931**	0.8531	1.1177
y(t+1)	1.1546	1.0427	1.1426	1.1049
y(t+2)	1.1227	1.0459	1.1322	1.0932
y(t+3)	1.1375	1.0530	1.1266	1.0952
롤링윈도우 예측				
y(t)	0.9210	0.9826***	0.8510	1.0875
y(t+1)	1.1408	1.0700	1.1470	1.0781

y(t+2)	1.1140	1.0702	1.1412	1.0714
y(t+3)	1.0994	1.0527	1.1394	1.0698

주1: GRU: Gated Recurrent Unit. LSTM: Long-Short Term Memory. DFM = Dynamic Factor Model, AR = Autoregressive Model.

주2: *: Diebold-Mariano (1995) 검정 결과 DFM 모형과의 예측오차 제곱항의 차이가 10% 유의수준에서 유의함을 지칭.

주3: **: Diebold-Mariano (1995) 검정 결과 AR(1) 모형과의 예측오차 제곱항의 차이가 10% 유의수준에서 유의함을 지칭.

주4: ***: Diebold-Mariano (1995) 검정 결과 DFM 및 AR(1) 모형과의 예측오차 제곱항의 차이가 10% 유의수준에서 유의함을 지칭.

<표 20 > GDP 성장률 표본 외 예측 (2020년도 포함): MAE

(a) 초월

<표 21> 초월 기준 MAE (2020년도 포함)

	GRU	LSTM	DFM	AR(1)
축차 예측				
y(t)	0.7070	0.6389	0.6669	0.7002
y(t+1)	0.8210	0.6745	0.7842	0.7204
y(t+2)	0.7754	0.6785	0.7759	0.6900
y(t+3)	0.7772	0.6757	0.7722	0.7014
롤링윈도우 예측				
y(t)	0.7024	0.6717	0.6656	0.6836
y(t+1)	0.8258	0.7182	0.8078	0.7108
y(t+2)	0.7715	0.7371	0.7925	0.6892
y(t+3)	0.7948	0.6819	0.7746	0.6852

(b) 중월

<표 22> 중월 기준 MAE (2020년도 포함)

	GRU	LSTM	DFM	AR(1)
축차 예측				
y(t)	0.6394*	0.6249	0.5292	0.7002
y(t+1)	0.7694	0.6796	0.7225	0.7204
y(t+2)	0.7635	0.6811	0.7112	0.6900
y(t+3)	0.7787*	0.6747	0.7119	0.7014
롤링윈도우 예측				
y(t)	0.6358	0.6074**	0.5439	0.6836
y(t+1)	0.7251	0.7031	0.7352	0.7108
y(t+2)	0.7220	0.7104	0.7263	0.6892

y(t+3)	0.7006	0.6649*	0.7394	0.6852

(c) 말월

<표 23> 말월 기준 MAE (2020년도 포함)

	GRU	LSTM	DFM	AR(1)
축차 예측				
y(t)	0.6072	0.6133**	0.5367	0.7623
y(t+1)	0.7739	0.6710**	0.7833	0.7240
y(t+2)	0.7655	0.6648**	0.7627	0.7034
y(t+3)	0.8006	0.6791	0.7935	0.7036
롤링윈도우 예측				
y(t)	0.6125	0.6330**	0.5399	0.7437
y(t+1)	0.7461	0.7324	0.8106	0.7167
y(t+2)	0.7789	0.7271	0.7757	0.6914
y(t+3)	0.7824	0.7317	0.8316	0.6962

(d) 종합

<표 24> 종합 MAE (2020년도 포함)

	GRU	LSTM	DFM	AR(1)
축차 예측				
y(t)	0.6512*	0.6257**	0.5776	0.7209
y(t+1)	0.7881	0.6750	0.7633	0.7216
y(t+2)	0.7681	0.6748	0.7499	0.6944
y(t+3)	0.7855	0.6765	0.7592	0.7021
롤링윈도우 예측				
y(t)	0.6502*	0.6374**	0.5832	0.7037
y(t+1)	0.7657	0.7179	0.7845	0.7127
y(t+2)	0.7575	0.7249	0.7649	0.6900
y(t+3)	0.7593	0.6928*	0.7819	0.6888

주1: GRU: Gated Recurrent Unit. LSTM: Long-Short Term Memory. DFM = Dynamic Factor Model, AR = Autoregressive Model.

주2: *: Diebold-Mariano (1995) 검정 결과 DFM 모형과의 예측오차 절대값의 차이가 10% 유의수준에서 유의함을 지칭.

주3: **: Diebold-Mariano (1995) 검정 결과 AR(1) 모형과의 예측오차 절대값의 차이가 10% 유의수준에서 유의함을 지칭.

주4: ***: Diebold-Mariano (1995) 검정 결과 DFM 및 AR(1) 모형과의 예측오차 절대값의 차이가 10% 유의수준에서 유의함을 지칭.

4. 결론

❑ 본 연구는 현대 사회의 여러 분야에서 활발하게 응용되고 있는 대표적인 기계학습 방법론 중 하나인 딥러닝 알고리즘 계열의 순환신경망을 이용하여 우리나라 단기 경제성장률 예측과 관련된 모형의 예측력에 대해서 살펴보았음

○ LSTM 기반 예측모형의 경우 벤치마크 모형인 DFM 모형과 비교하여 예측력에 있어서 아주 큰 차이를 만들어 내지 못하고 있으나, AR 모형과 비교해서는 예측력이 상대적으로 뛰어난 것으로 평가됨

○ 벤치마크 모형 대비 예측력이 크게 증가하지 못한 원인은 아래와 같이 정리할 수 있음

- 첫째, 학습 데이터의 양이 절대적으로 부족한 현상에 기인. 일반적인 기계학습 환경과 대비해서 인공신경망의 학습에 사용할 수 있는 데이터의 양이 매우 적음. 샘플 개수가 최대 80개 이내라, 과적합(overfitting)의 문제가 완전히 해결되기 어려운 상황

- 둘째, 결측치 다루는 과정에서 DFM 모형에 기반한 추측치를 사용함에 가중된 불확실성. 신경망 학습은 물론, 예측치를 산출하기 위한 작업에도 설명변수들의 예측치가 투입되는 이중 불확실성 존재. 따라서 동 모형 예측력이 DFM의 예측력에 크게 의존하고 있는 상황

○ 그럼에도 불구하고, 문헌 내 존재하는 여러 예측모형 대비 예측오차가 양호한 수준으로 평가됨

- 아래 <표 25>는 문헌 내 기계학습을 이용한 다양한 모형들의 당분기 성장률에 대한 예측오차를 나타내주고 있음

<표 25> 문헌 내 모형과 예측오차 비교

모델	RMSE
ForecastModel_NABO	0.47~
Biau and D' Elia (2012)	0.66~
Tiffin (2016)	0.69~
Jung et al. (2018)	0.17~
Medeiros et al. (2021)	0.73~

주: ForecastModel_NABO는 2020년을 제외한 표본 외 예측결과임

- 동 모형의 예측오차는 앙상블 기반의 예측모형인 Jung et al. (2018)보다 높은 수치를 보여주고 있지만, 다른 모형 대비 낮은 예측오차를 기록

○ 2022년도 2분기 실질 경제성장률에 대한 동 모형의 예측치(2022년 6월말 기준)와 시장 예측치를 살펴보면 아래 <표 26>과 같음

<표 26> 2022년 2분기 한국 경제성장률 예측치 (단위: %)

모델	예측치
실현 성장률	0.70
LSTM_mq	0.7019
GRU_mq	0.4520
LSTM_m	0.8585
GRU_m	0.3547
노무라증권[30]	1.60
BNP 파리바[31]	0.40
시장 컨센서스	0.40

주: mq = [부록 1]의 혼합주기 및 하위 그룹 요인 설정 모형, m = 본문의 공통요인 및 월별자료 사용 모형

30) https://www.mk.co.kr/news/economy/view/2022/07/658262/
31) https://news.einfomax.co.kr/news/articleView.html?idxno=4226006

- 위 결과는 한 시점의 예측치에 대한 정보를 담고 있고 전반적인 예측력을 평가하기에 무리가 있지만, 동 모형의 지속적인 수정·보완을 통해 예측모형의 예측력을 전반적으로 향상시킬 수 있는 가능성이 있음을 보여줌[32]

❑ 향후 성장률 예측모형의 개선 및 발전방향은 다음과 같이 정리할 수 있음

○ 동 모형은 예측시계가 길어질수록 벤치마크 모형과의 예측력에 있어서 큰 차이를 만들어내지 못한다는 점에서, 학습 데이터의 구조 상 목표 변수를 미래 시점의 성장률로 대체하여 학습 진행 후 예측오차의 개선이 있는지 검토

- 다만, 이러한 접근 방법도 DFM 예측치에 기반하고 있기 때문에, 모형의 최종 예측력은 DFM 모형의 예측력에 의존할 수 밖에 없는 상황

- 따라서 [부록 2]에서 고려한 것과 같이, 발표 시차가 없는 고빈도 자료가 중심이 되는 예측모형에 대한 추가 연구 필요

○ 하나의 기계학습 알고리즘에 의존하는 예측 모형이 아니라, 여러 알고리즘들의 예측치를 결합하여 최종적인 예측치를 산출하는 앙상블(ensemble) 기법에 대한 연구 필요

- 예를 들어, Super Learner 알고리즘 (e.g., van der Laan et al., 2007)[33]의 경우 여러 예측 모형이 산출해는 예측결과의 최적의 결합비중을 학습을 통해 도출

- Super Learner 내 LSTM 또는 GRU 기반 예측모형을 하위 예측모형으로 설정하여, 기타 다른 예측모형들과 결합한 앙상블 기법을 적용

32) [부록 4]는 본 연구에서 사용한 모형의 API documentation을 담고 있어, 모형의 실제 적용, 결측치 추정 방법 및 하위 기계학습 알고리즘 변화 등에 참고.
33) Super Learner의 파이썬 내 구현에 대한 자세한 내용은 http://ml-ensemble.com/참고

5. 참고문헌

강창구 (2010), “상태공간모형을 이용한 월별 GDP 추정”, 한국은행 계간 국민계정 , 2010년 제3호, 49～79.

김덕파·강성진 (2020), “월별 지표를 이용한 GDP추정을 위한 계량모형 구축”, 정책연구용역 보고서, 국회예산정책처

김윤영 (2010), “혼합주기자료 VAR모형을 이용한 통화정책의 월별 동태효과 분석”, 경제학연구 , 제58권 제1호, 87～111.

김윤영·박준용 (2007), “혼합주기자료 VAR모형을 이용한 분기 GDP성장률의 최적예측”, 경제학연구 , 제55권 제3호, 189～217.

오현희 (2022), “단기 GDP 예측모형 연구”, 경제현안분석 104호, 국회예산정책처.

최병재·한진현 (2014), “월별 자료를 이용한 예측조합에 의한 GDP 전망”, 한국은행 조사통계월보 , 2014년 10월호, 16～48.

황선웅·박문수 (2012), “대규모 혼합주기 자료를 이용한 단기 경제전망 모형 구축”, 산업연구원, 연구보고서, 2012-642.

황종률 (2012), “월간경제지표를 이용한 실질 GDP성장률 예측”, 예산정책연구 , 제1권 제2호, 103～133.

Aastveit, K., K. R. Gerdrup and A. S. Jore (2011), “Short-term forecasting of GDP and Inflation in real-time: Norges Bank’s System for Averaging Models”, *Norges Bank Staff Memo*, No. 09.

Angelini, E., G. Camba-Mendez, D. Giannone, G. R ´ unstler, and L. Reichlin ¨ (2008), “Short-term forecasts of euro area GDP growth,” Working Paper Series 949, European Central Bank

Banbura, M., and M. Modugno (2010), “Maximum likelihood estimation of large factor model on datasets with arbitrary pattern of missing data” Working Paper Series 1189, European Central Bank.

Banbura, M., D. Giannone and L. Reichlin (2010), “Nowcasting” Working Paper Series 1275, European Central Bank.

Barhoumi, K., S. Benk, R. Cristadoro, A. D. Reijer, A. Jakaitiene, P. Jelonek, A. Rua,

G.Runstler, K. Ruth and C. Van Nieuwenhuyze (2008), “Short-term forecasting of GDP using large monthly datasets: A pseudo real-time forecast evaluation exercise”, European Central Bank, *ECB Working Paper Series*, No. 84.

Biau, O. and A. D’Elia (2012). “Euro area GDP forecasting using large survey datasets: A random forest approach,” mimeo.

Chernis, Tony and Sekkel, Rodrigo, (2017), “A dynamic factor model for nowcasting Canadian GDP growth,” *Empirical Economics*, 53, issue 1, p. 217-234.

Cho, Kyunghyun, B. van Merrienboer, D. Bahdanau and Y. Bengio (2014), “On the properties of neural machine translation: Encoder-decode approach,” In Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation, pages 103-111.

Clements, M. P. and D. I. Harvey (2009), “Forecast combination and encompassing”, *Palgrave handbook of econometrics*, *Palgrave MacMillan, London*, 169～198.

Diron, M. (2008): “Short-term forecasts of euro area real GDP growth: an assessment of real-time performance based on vintage data,” Journal of Forecasting, 27(5), 371-390.

Diebold, F.X. and R.S. Mariano (1995). “Comparing predictive accuracy,” Journal of Business and Economic Statistics, 13, 253-263.

Elger, Thomas, Binner, Jane, Nilsson, Birger, & Tepper, Jonathan (2006). “Predictable non-linearities in U.S. Inflation,” Economics Letters, 93(02), 323-328.

Elman, Jeffrey L. (1990), “Finding structure in time”, *Cognitive Science*, 14(2), 179-211.

Foroni, C. and M. Marcellino (2012), “A comparison of mixed frequency approaches for modelling Euro Area macroeconomic variables”, European University Institute, *EUI Working Papers*, Eco 2012/07.

Giannone, D., L. Reichlin and D. Small (2008), “Nowcasting: The real-time informational content of macroeconomic data,” *Journal of Monetary Economics*, Vol. 55(4), 665～676.

Hecht-Nielsen, Robert (1992), “Theory of the backpropagation neural network,” *Neural Networks for Perception*, 65～93.

Hochreiter S, Schmidhuber J (1997), “Long short-term memory,” Neural Computation, 15;9(8):1735-80.

Jung, J.-K., M. Patnam, and A. Ter-Martirosyan (2018), “An algorithmic crystal ball: Forecasts-based on machine learning,” IMF working paper.

Kitchen, J., and R. M. Monaco (2003), “Real-time forecasting in practice: The U.S. Treasury Sta.s real-time GDP forecast system,” Business Economics, pp. 10-19.

Kaiser, H. F. (1960), “The application of electronic computers to factor analysis,” Educational and Psychological Measurement, 20, 141-151.

Nakamura, Emi (2005), “Inflation forecasting using a neural network,” Economics Letters, 86(3), 373-378.

Nunes, L.C. (2005), “Nowcasting quarterly GDP growth in a monthly coincident indicator model”, *Journal of Forecasting,* Vol. 24(8), 575～592.

Mariano, R., and Y. Murasawa (2003), “A new coincident index of business cycles based on monthly and quarterly series,” Journal of Applied Econometrics, 18, 427-443.

Marcellino, Massimiliano and Schumacher, Christian, (2010), “Factor MIDAS for nowcasting and forecasting with ragged-edge data: A model comparison for German GDP,” *Oxford Bulletin of Economics and Statistics*, 72, issue 4, p. 518-550.

Matheson, T. D. (2006), “Factor model forecasts for New Zealand,” *International Journal of Central Banking*, vol. 2(2), May.

Medeiros, M. C., G. F. R. Vasconcelos, A. Veiga, and E. Zilberman (2021), “Forecasting inflation in a data-rich environment: The benefits of machine learning methods.” *Journal of Business & Economic Statistics* 39(1), 1-22.

Mitchell, J., R. J. Smith, M. R. Weale, S. Wright, and E. L. Salazar (2005), “An Indicator of monthly GDP and an early estimate of quarterly GDP growth,” Economic Journal, 115(501), F108-F129.

Mittnik, S. and P. Zadrozny (2004), “Forecasting quarterly German GDP at monthly intervals using monthly IFO business conditions data”, *CESIFO Working Paper Series*, No. 1203.

Parigi, G. and G. Schlitzer (1995), “Quarterly forecasts of the Italian business cycle by means of monthly economic indicators”, *Journal of Forecasting,* Vol. 14, 117～141.

Parigi, G., and R. Golinelli (2007), “The use of monthly indicators to forecast quarterly GDP in the short run: an application to the G7 countries,” Journal of Forecasting, 26(2), 77-94.

Proietti, T. (2008), “Estimation of common factors under cross-sectional and temporal aggregation constraints: Nowcasting monthly GDP and its main components,” MPRA Paper 6860, University Library of Munich, Germany.

Rathjens, P. and R. P. Robins (1993), “Forecasting quarterly data using monthly information”, *Journal of Forecasting*, Vol. 12, 321~330.

Runstler, G., K. Barhoumi, R. Cristadoro, A. D. Reijer, A. Jakaitiene, ¨P. Jelonek, A. Rua, K. Ruth, S. Benk, and C. V. Nieuwenhuyze (2008): “Shortterm forecasting of GDP using large monthly data sets: a pseudo real-time forecast evaluation exercise,” Occasional Paper Series No 84, European Central Bank.

Schumacher, C., and J. Breitung (2008), “Real-time forecasting of German GDP based on a large factor model with monthly and quarterly Data,” International Journal of Forecasting, 24, 386-3

Stock, J. H., and M. W. Watson (2002), “Macroeconomic forecasting using diffusion indexes,” Journal of Business and Economics Statistics, 20(2), 147-162

Stock, J.H. and Watson, M.W. (2004), “Combination forecasts of output growth in a seven country data set,” *Journal of Forecasting,* Vol. 23, 405-430.

Swanson, Norman and White, Halbert, (1997), “A model selection approach to real-time macroeconomic forecasting using linear models And artificial neural networks,” The Review of Economics and Statistics, 79(4), 540-550.

Tashman, Leonard J., (2000), “Out-of-sample tests of forecasting accuracy: an analysis and review,” International Journal of Forecasting, 16, issue 4, p. 437-450

Trehan, B. (1989), “Forecasting growth in current quarter real GNP,” Economic Review, 39-52.

Tiffin, A. (2016), “Seeing in the dark: A machine-learning approach to nowcasting in Lebanon,” *IMF working paper*.

Tkacz, G. (2001), “Neural network forecasting of Canadian GDP growth.” *International Journal of Forecasting*, 17(1), 57-69.

van der Laan M.J., Polley EC, Hubbard A.E. (2007), “Super learner,” *Statistical Application in Genetics and Molecular Biology*, 6(1):Article25.

Zheng, Y. and J. Rossiter (2006), “Using monthly indicators to predict quarterly GDP,” *Bank of Canada Working Paper*, 2006-26.

[부록 1: 혼합주기 자료 예측모형 결과]

❑ 본문의 모형에서는 결측치 추정을 위한 DFM 적용시 공통요인의 개수를 주성분분석을 통해 결정하였지만, 본 부록의 모형에서는 공통요인의 임의 지정 및 혼합주기(월별, 분기별) 예측 모형의 예측성능을 점검함

(1) 모형 셋업

○ 본 부록의 모형에서는 Banbura et al. (2010)을 따라 모든 변수에 영향을 미치는 공통요인 하나와 각 그룹별 하위 요인들을 상정하여 DFM을 구성하였음[34)]

- 예를 들어, 모든 변수에 영향을 미치는 공통요인이 존재하고, 실물변수 그룹(R)과 명목변수 그룹(N)으로 나누어 각 그룹별 하위 공통요인을 상정

- DFM을 구성하는 본문 식 (7) 주요 모수는 아래 식 (27)과 같이 수정될 수 있음[35)]

34) 아래 설명은 Banbura et al. (2010) 및 Mariano and Murasawa (2003)의 일부 내용을 요약한 것임.
35) 단, y_t 벡터상에서 명목변수가 실질변수보다 앞서 있다는 가정 하에 설명된 것임.

$$\Lambda = \begin{bmatrix} \Lambda_{N,G} & \Lambda_{N,N} & 0 \\ \Lambda_{R,G} & 0 & A_{R,R} \end{bmatrix}$$

$$f_t = \begin{bmatrix} f_t^G \\ f_t^N \\ f_t^R \end{bmatrix}$$

$$A_i = \begin{bmatrix} A_{i,G} & 0 & 0 \\ 0 & A_{i,N} & 0 \\ 0 & 0 & A_{i,R} \end{bmatrix}$$

$$Q = \begin{bmatrix} Q_G & 0 & 0 \\ 0 & Q_N & 0 \\ 0 & 0 & Q_R \end{bmatrix}$$

..... (27)

- 단, $\Lambda_{i,j}$는 $j = \{Global, Nominal, Real\}$ 요인이 $i = \{Nominal, Real\}$ 그룹의 변수에 미치는 영향을 나타내는 factor loading을 나타냄

○ 분기 자료의 경우 월별 자료와는 다르게 자료의 분절이 규칙적으로 일어나게 됨

- Mariano and Murasawa (2003)는 이를 고려한 DFM 구조를 제시함. 가령 분기별 소비(CON_t^Q) 자료가 관측 불가능한(latent) 월별 소비(CON_t^M) 자료의 합이라고 가정하면, 분기 변수는 아래 식 (28)과 같이 표현될 수 있음

$$CON_t^Q = CON_t^M + CON_{t-1}^M + CON_{t-2}^M$$

..... (28)

- $C_t^Q = 100 \times \ln(CON_t^Q)$, $C_t^M = 100 \times \ln(CON_t^M)$, $c_t = \triangle C_t^M$으로 정의하고, c_t의 확률과정이 월별 자료와 동일하게 DFM 구조를 따른다고 가정하면,[36] 아래 식 (29)과 같이 표현될 수 있음

$$
\begin{aligned}
c_t &= \mu_Q + \Lambda_Q f_t + \epsilon_t^Q \\
\epsilon_t^Q &= \alpha_Q \epsilon_{t-1}^Q + e_t^Q \\
e_t^Q &\sim i.i.d.\, N(0, \sigma_Q^2) \\
\Lambda_Q &= \begin{bmatrix} \Lambda_{Q,G} & 0 & \Lambda_{Q,R} \end{bmatrix} \\
f_t &= \begin{bmatrix} f_t^G \\ f_t^N \\ f_t^R \end{bmatrix}
\end{aligned}
$$

..... (29)

- 월별 주기로 표현한 분기 자료(c_t^Q)는 아래와 같이 표현될 수 있음(즉, 분기 말월에는 관측 가능하나, 분기 말월 이외의 시점에서는 관측 불가능한 것으로 상정)

$$
c_t^Q = \begin{cases} C_t^Q - C_{t-3}^Q & , t = 3, 6, 9, \cdots \\ NaN & , otherwise \end{cases}
$$

..... (30)

- 식 (7)과 C_t^Q에 대한 정의를 결합하면, 관측 가능한 분기 변수와 월별 latent 변수간의 연결관계를 아래 식 (31)과 같이 정의할 수 있음

$$
\begin{aligned}
c_t^Q = C_t^Q - C_{t-3}^Q &\approx (C_t^M + C_{t-1}^M + C_{t-2}^M) - (C_{t-3}^M + C_{t-4}^M + C_{t-5}^M) \\
&= c_t + 2c_{t-1} + 3c_{t-2} + 2c_{t-3} + c_{t-4}
\end{aligned}
$$

..... (31)

36) 단, 여기서 분기 변수는 실물변수 그룹에 포함되었다고 가정하였음.

○ 식 (27) - 식 (31)을 이용하여, 예측에 사용할 월별 변수(x_t) 변수와 분기 변수(c_t)를 아래 식 (32)와 같은 상태공간모형(SSM)으로 나타낼 수 있음

$$
\begin{aligned}
z_t &= \mu_z + Z(\theta)s_t \\
s_t &= T(\theta)s_{t-1} + \eta_t
\end{aligned}
$$

$$
\begin{aligned}
z_t &= ({x_t}', c_t^Q)' \\
\mu_z &= (\mu', \mu_Q)'
\end{aligned}
$$

..... (32)

- 상기한 SSM에서 상태변수 벡터(state variable vector)는 공통요인과 월별 변수와 분기 변수의 idiosyncratic shock으로 이루어져 있음: [37]

$$s_t = ({f'}_t, {f'}_{t-1}, {f'}_{t-2}, {f'}_{t-3}, {f'}_{t-4}, \epsilon_{1t}, \cdots, \epsilon_{nt}, \epsilon_t^Q, \epsilon_{t-1}^Q, \epsilon_{t-2}^Q, \epsilon_{t-3}^Q, \epsilon_{t-4}^Q)'$$

- 가령, 공통요인(factor; ‘GLOBAL’, ‘REAL’, ‘NOMINAL’)이 3개이고, $p \leq 5$ 일 경우 상태공간모형의 구체적인 형태는 아래 식 (33)-식 (34)과 같이 표현될 수 있음

37) 단, $p \leq 5$ 일 경우임.

(Measurement Equation)

$$\begin{bmatrix} x_t \\ c_t^Q \end{bmatrix} = \begin{bmatrix} \Lambda & 0 & 0 & 0 & 0 & I_n & 0 & 0 & 0 & 0 & 0 \\ \Lambda_Q & 2\Lambda_Q & 3\Lambda_Q & 2\Lambda_Q & \Lambda_Q & 0 & 1 & 2 & 3 & 2 & 1 \end{bmatrix} \begin{bmatrix} f_t \\ f_{t-1} \\ f_{t-2} \\ f_{t-3} \\ f_{t-4} \\ \epsilon_t \\ \epsilon_t^Q \\ \epsilon_{t-1}^Q \\ \epsilon_{t-2}^Q \\ \epsilon_{t-3}^Q \\ \epsilon_{t-4}^Q \end{bmatrix} \quad \cdots\cdots\cdots\cdots\cdots\cdots\cdots \quad (33)$$

(Transition Equation)

$$\begin{bmatrix} f_t \\ f_{t-1} \\ f_{t-2} \\ f_{t-3} \\ f_{t-4} \\ \epsilon_t \\ \epsilon_t^Q \\ \epsilon_{t-1}^Q \\ \epsilon_{t-2}^Q \\ \epsilon_{t-3}^Q \\ \epsilon_{t-4}^Q \end{bmatrix} = \begin{bmatrix} A_1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ I_r & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & I_r & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & I_r & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & I_r & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & diag(\alpha_1, \cdots, \alpha_n) & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & \alpha_Q & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} f_{t-1} \\ f_{t-2} \\ f_{t-3} \\ f_{t-4} \\ f_{t-5} \\ \epsilon_{t-1} \\ \epsilon_{t-1}^Q \\ \epsilon_{t-2}^Q \\ \epsilon_{t-3}^Q \\ \epsilon_{t-4}^Q \\ \epsilon_{t-5}^Q \end{bmatrix} + \begin{bmatrix} u_t \\ 0 \\ 0 \\ 0 \\ 0 \\ e_t \\ e_t^Q \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad \cdots\cdots\cdots\cdots\cdots\cdots\cdots \quad (34)$$

- 위 논의는 분기 변수(quarterly variable)가 하나인 경우를 상정하여 기술한 것인지만, 분기 변수의 factor loading (Λ^Q)과 α^Q를 수정함으로써 다수의 분기별 자료가 존재하는 경우도 상태공간모형으로 쉽게 표현 가능

(2) 추가 자료

❑ 본 부록의 모형에서 추가된 분기자료는 GDP 지출 측면을 구성하는 주요 요소인 소비, 투자, 수출 자료를 추가함

<표 27> 혼합주기 모형 설명 변수

변수명	변수 설명	LAG	DIFF	GROUP	FREQ
CONSUMPTION	민간소비 증가율	2	0	REAL	Q
PINVEST	설비투자 증가율	2	0	REAL	Q
CINVEST	건설투자 증가율	2	0	REAL	Q
EXPORT	수출 증가율	2	0	REAL	Q
PROD	산업생산지수 (전산업, 농림업제외)	2	1	REAL	M
PRODMAN	산업생산지수 (제조업)	2	1	REAL	M
PRODSER	산업생산지수 (서비스업)	2	1	REAL	M
PRODMIN	광업생산지수	2	1	REAL	M
MAFSHIP	제조업출하지수	2	1	REAL	M
MAFINVEN	제조업재고지수	2	1	REAL	M
INTENSITY	제조업가동률지수	2	0	REAL	M
LEADINDEX	경기선행지수 (순환변동치)	2	0	REAL	M
COINDEX	경기동행지수 (순환변동치)	2	0	REAL	M
RETAILSALES	소매판매지수	2	1	REAL	M
RETAILSALESD	소매판매지수 (내구재)	2	1	REAL	M
RETAILSALESND	소매판매지수 (비내구재)	2	1	REAL	M
INVESTIND	설비투자지수	2	1	REAL	M
CONSAMT	건설기성액 (불변)	2	0	REAL	M
CPI	소비자물가지수	1	1	NOMINAL	M
PPI	생산자물가지수	1	1	NOMINAL	M
IPI	수입물가지수	1	1	NOMINAL	M
EXPI	수출물가지수	1	1	NOMINAL	M
CPI_EXAGRI	소비자물가지수(농산물석유제외)	1	1	NOMINAL	M
CPI_EXFOODS	소비자물가지수(식료품에너지제외)	1	1	NOMINAL	M
EMP_NUM	취업자수	1	1	LABOR	M
UEMPR	실업률	1	0	LABOR	M
EMPR	고용률	1	0	LABOR	M

ECONPART_RATE	경제활동참가율	1	0	LABOR	M
M1END	M1말잔	2	1	MONEY	M
M2END	M2말잔	2	1	MONEY	M
LFEND	LF 말잔	2	1	MONEY	M
EXP	수출금액 (통관기준)	1	1	REAL	M
IMP	수입금액 (통관기준)	1	1	REAL	M
BSI_SALES	매출 BSI (실적기준, 전산업)	0	0	SENTIMENT	M
BSI_ENV	업황 BSI (실적기준, 전산업)	0	0	SENTIMENT	M
BSI_MAF_EX	수출 BSI (제조업)	0	0	SENTIMENT	M
BSI_MAF_INTENSITY	가동률 BSI (제조업)	0	0	SENTIMENT	M
BSI_MAF_DOMESTIC	내수판매 BSI (제조업)	0	0	SENTIMENT	M
BSI_MAF_ENV	업황 BSI (제조업)	0	0	SENTIMENT	M
SENT	경제심리지수 (순환변동치)	0	0	SENTIMENT	M

○ <표 27>의 「GROUP」은 해당 설명변수가 DFM의 다중 요인중 어떤 하위 그룹별 요인에 영향을 받는지 나타내는 변수임

- 즉, 모든 변수에 영향을 미치는 'GLOBAL' 공통요인을 상정 후, 각 변수의 하위 그룹별 해당 공통요인을 설정: 'REAL', 'NOMINAL', 'LABOR', 'MONEY', 'SENTIMENT' 등 5개의 하위그룹별 공통요인을 상정함

(3) 표본 외 예측 결과 (2020년도 제외)

(RMSE 기준)

❑ 당분기 GDP성장률 표본 외 예측에서 전반적으로 축차예측 기반 LSTM 모형의 RMSE는 0.45, GRU의 RMSE는 0.52를 나타냄 (<표 32> 참조)

○ 롤링윈도우 예측의 경우 GRU의 RMSE는 약 0.65, LSTM의 RMSE는 약 0.47을 기록하여, 전반적으로 당분기 경제성장률 예측에는 LSTM 기반 모형이 혼합주기 자료 하에서도 상대적으로 더 우수한 것으로 평가됨

- 당분기 축차예측 기준, LSTM모형은 AR모형과 비교해서 예측력에 유의한 차이가 있으나, GRU 모형은 벤치마크 모형 대비 예측력의 유의한 차이를 보이지

않음

- GRU 모형 축차예측의 경우, 전 예측시계에 걸쳐 DFM 보다 예측오차가 상대적으로 더욱 높게 나타나는 경향이 관찰되며, 롤링윈도우 기준 당분기 및 1분기 후 성장률 예측에서 벤치마크 모형보다 더 낮은 예측력을 가진 것으로 나타남

- LSTM 모형은 DFM 모형 예측 결과와 비교해 전 예측시계에 걸쳐 크게 다르게 나타나지 않은 것으로 관찰되며, 예측력에 있어서 통계적으로 유의한 차이는 발견되지 않음

○ 예측 시계가 길어질수록 예측오차가 확대되는 경향을 보이고 있으며, 2분기 이후 성장률에 대한 예측력은 LSTM 모형을 제외하고는 AR 모형과도 큰 차이를 보이지 않는 것으로 관찰됨

○ LSTM 모형의 예측오차는 예측 시점(초월, 중월, 말월)에 따라 크게 변동하지 않으나, GRU 모형의 예측오차는 중월에서 말월로 갈수록 확대되는 경향이 있음

- LSTM 모형의 경우, 말월의 당분기 성장률 예측 기준 가장 낮은 예측 오차를 나타냄

(MAE 기준)

❑ 당분기 GDP성장률 표본 외 축차예측에서 전반적으로 LSTM 모형의 MAE는 약 0.34, GRU의 RMSE는 약 0.42를 나타냄 (<표 37> 참조)

○ 롤링윈도우 예측의 경우 GRU의 MAE는 약 0.53, LSTM의 MAE는 약 0.37을 나타냄

- 혼합주기 자료 하에서도 GRU 기반 예측의 경우 벤치마크 모형인 DFM 보다 예측오차가 상대적으로 더욱 높게 나타나는 경향이 있고, 다수의 예측시계에 걸쳐 DFM 모형보다 예측력이 낮은 것으로 평가됨

- LSTM 기반 예측의 경우 벤치마크 모형인 DFM의 예측 결과와 비교해 전 예측

시계에 걸쳐 큰 차이를 보이고 있지 않지만, 롤링윈도우 1분기 후 성장률 예측 기준, DFM 모형보다 예측력이 높은 것으로 평가됨

- 축차예측 기준, LSTM 모형은 AR 모형보다 예측력이 높은 것으로 평가됨

○ 모든 예측 모형에 있어서 예측 시계가 길어질수록 예측오차가 확대되는 경향을 보이고 있으며, 전 예측시계에 걸쳐 GRU 모형은 LSTM 모형의 예측오차 보다 높은 값을 기록함

○ 예측 시점(초월, 중월, 말월)에 따라 예측 오차가 크게 변동하지 않는 것으로 관찰됨

- 초월 축차예측 당분기 성장률 예측 기준, LSTM 모형은 AR 모형보다 예측력이 높은 것으로 평가되나, DFM과는 예측력에 있어서 유의미한 차이가 없음

- 초월 롤링윈도우 예측 1분기 후 성장률 예측 기준, LSTM 모형은 DFM 모형 대비 예측력이 높은 것으로 평가됨

- 중월 축차 예측 1분기 후 성장률 예측 기준, LSTM 모형은 AR 모형 대비 예측력이 높은 것으로 평가됨. 또한 중월 롤링윈도우 1분기 후 성장률 예측 기준, DFM 모형 대비 예측력이 높은 것으로 평가됨

- 말월 기준, LSTM 모형과 DFM 모형 간의 예측력의 차이는 존재하지 않는 것으로 평가되나, LSTM 모형은 AR 모형 대비 예측력이 높은 것으로 평가됨

<표 28> 혼합주기 모형 GDP 성장률 표본 외 예측(2020년도 제외): RMSE

(a) 초월

<표 29> 초월 기준 혼합주기 모형 RMSE (2020년도 제외)

	GRU	LSTM	DFM	AR(1)
축차 예측				
y(t)	0.4915	0.4727	0.5416	0.5124
y(t+1)	0.6605**	0.6350**	0.6396	0.7299
y(t+2)	1.0574	1.0567	1.0301	1.1408
y(t+3)	1.0479	1.1019**	1.0995	1.1771

롤링윈도우 예측				
y(t)	0.5757	0.5080	0.5222	0.4912
y(t+1)	0.7428	0.6185**	0.6278	0.6944
y(t+2)	1.0335*	1.0054	1.0005	1.1010
y(t+3)	1.0480	1.0802	1.0835	1.1464

(b) 중월

<표 30> 중월 기준 혼합주기 모형 RMSE (2020년도 제외)

	GRU	LSTM	DFM	AR(1)
축차 예측				
y(t)	0.4691	0.4240	0.3926	0.5124
y(t+1)	0.6997*	0.6604**	0.6468	0.7299
y(t+2)	1.0081	1.0437	1.0191	1.1408
y(t+3)	1.0884**	1.1197**	1.1086	1.1771
롤링윈도우 예측				
y(t)	0.5906*	0.4429	0.4005	0.4912
y(t+1)	0.7924***	0.6233	0.6394	0.6944
y(t+2)	0.9874	0.9828	0.9879	1.1010
y(t+3)	1.0975	1.1099	1.0980	1.1464

(c) 말월

<표 31> 말월 기준 혼합주기 모형 RMSE (2020년도 제외)

축차 예측				
y(t)	0.6036*	0.4531**	0.4977	0.5737
y(t+1)	0.7359*	0.6488	0.6673	0.7115
y(t+2)	1.0110	1.0011	1.0151	1.1376
y(t+3)	1.1186	1.1261**	1.1165	1.1776
롤링윈도우 예측				
y(t)	0.7596***	0.4594**	0.4823	0.5401
y(t+1)	0.8421*	0.6616	0.6697	0.6863
y(t+2)	1.0451	0.9758	0.9818	1.0993
y(t+3)	1.1175	1.0798*	1.0941	1.1470

(d) 종합

<표 32> 종합 기준 혼합주기 모형 RMSE (2020년도 제외)

	GRU	LSTM	DFM	AR(1)
축차 예측				

y(t)	0.5247	0.4504**	0.4814	0.5336
y(t+1)	0.6994	0.6482	0.6514	0.7238
y(t+2)	1.0258	1.0341**	1.0214	1.1398
y(t+3)	1.0854	1.1160	1.1082	1.1773
롤링윈도우 예측				
y(t)	0.6473***	0.4709	0.4710	0.5080
y(t+1)	0.7935*	0.6348	0.6459	0.6917
y(t+2)	1.0223	0.9881	0.9901	1.1005
y(t+3)	1.0881	1.0901	1.0919	1.1466

주1: GRU: Gated Recurrent Unit. LSTM: Long-Short Term Memory. DFM = Dynamic Factor Model, AR = Autoregressive Model.

주2: *: Diebold-Mariano (1995) 검정 결과 DFM 모형과의 예측오차 제곱항의 차이가 10% 유의수준에서 유의함을 지칭.

주3: **: Diebold-Mariano (1995) 검정 결과 AR(1) 모형과의 예측오차 제곱항의 차이가 10% 유의수준에서 유의함을 지칭.

주4: ***: Diebold-Mariano (1995) 검정 결과 DFM 및 AR(1) 모형과의 예측오차 제곱항의 차이가 10% 유의수준에서 유의함을 지칭.

<표 33> 혼합주기 모형 GDP 성장률 표본 외 예측(2020년도 제외): MAE

(a) 초월

<표 34> 초월 기준 혼합주기 모형 MAE (2020년도 제외)

	GRU	LSTM	DFM	AR(1)
축차 예측				
y(t)	0.3867	0.3770	0.4151	0.4106
y(t+1)	0.5007**	0.4630**	0.4759	0.5318
y(t+2)	0.6751	0.6408**	0.6280	0.7063
y(t+3)	0.6639**	0.6980**	0.7204	0.7554
롤링윈도우 예측				
y(t)	0.4451	0.4067	0.4057	0.4028
y(t+1)	0.6077**	0.4652*	0.4962	0.5115
y(t+2)	0.7557	0.6270	0.6227	0.6810
y(t+3)	0.7524	0.7139	0.7114	0.7350

(b) 중월

<표 35> 중월 기준 혼합주기 모형 MAE (2020년도 제외)

	GRU	LSTM	DFM	AR(1)
축차 예측				
y(t)	0.3700	0.3293	0.3269	0.4106

y(t+1)	0.5435*	0.4815**	0.4806	0.5318
y(t+2)	0.6659*	0.6191**	0.6075	0.7063
y(t+3)	0.7031	0.6980	0.7006	0.7554
롤링윈도우 예측				
y(t)	0.5052*	0.3540	0.3451	0.4028
y(t+1)	0.6223*	0.4717***	0.4918	0.5115
y(t+2)	0.7570**	0.6058**	0.5953	0.6810
y(t+3)	0.7822	0.7081	0.7173	0.7350

(c) 말월

<표 36> 말월 기준 혼합주기 모형 MAE (2020년도 제외)

	GRU	LSTM	DFM	AR(1)
축차 예측				
y(t)	0.4962*	0.3226**	0.3742	0.4838
y(t+1)	0.5577	0.4733	0.5013	0.5128
y(t+2)	0.6424	0.6024**	0.6060	0.7028
y(t+3)	0.7605	0.7063	0.7174	0.7557
롤링윈도우 예측				
y(t)	0.6340***	0.3528**	0.3614	0.4545
y(t+1)	0.7065***	0.4792**	0.5307	0.5025
y(t+2)	0.7925*	0.6081**	0.5869	0.6770
y(t+3)	0.8278	0.7037	0.7319	0.7368

(d) 종합

<표 37> 말월 기준 혼합주기 모형 MAE (2020년도 제외)

	GRU	LSTM	DFM	AR(1)
축차 예측				
y(t)	0.4177	0.3430**	0.3721	0.4350
y(t+1)	0.5340	0.4726	0.4859	0.5255
y(t+2)	0.6611	0.6208**	0.6138	0.7051
y(t+3)	0.7092	0.7008**	0.7128	0.7555
롤링윈도우 예측				
y(t)	0.5281***	0.3712	0.3707	0.4201
y(t+1)	0.6455*	0.4720*	0.5062	0.5085
y(t+2)	0.7684*	0.6136	0.6016	0.6797
y(t+3)	0.7875	0.7085	0.7202	0.7356

주1: GRU: Gated Recurrent Unit. LSTM: Long-Short Term Memory. DFM = Dynamic Factor Model, AR =

Autoregressive Model.

주2: *: Diebold-Mariano (1995) 검정 결과 DFM 모형과의 예측오차 절대값의 차이가 10% 유의수준에서 유의함을 지칭.

주3: **: Diebold-Mariano (1995) 검정 결과 AR(1) 모형과의 예측오차 절대값의 차이가 10% 유의수준에서 유의함을 지칭.

주4: ***: Diebold-Mariano (1995) 검정 결과 DFM 및 AR(1) 모형과의 예측오차 절대값의 차이가 10% 유의수준에서 유의함을 지칭.

(4) 코로나-19 팬데믹 기간 모형의 예측 양상

❑ 코로나-19와 같은 예외적인 이벤트 상황을 동 모형이 어떻게 판단하는지 알아보기 위해 <그림 18>, <그림 19>는 예측 기간을 2020년으로 한정하여 당시 실제 실현된 GDP성장률과 모형의 예측치를 같이 표현하고 있음

○ 축차 예측 및 롤링윈도우 예측 모두 GDP성장률의 전반적인 경로와 유사한 흐름을 보여주고 있지만, 구체적인 수치에 있어서는 실제 성장률과의 간극이 존재함

- 다만, 축차/롤링 예측의 경우 2020년 1월 기준 1분기 성장률을 약 0.7%로 예상하였고, 모형은 말월로 갈수록 예측치를 하향조정하는 패턴을 보임

- 축차 예측의 경우, 초월 기준 2020년 1분기 성장률을 0.74%로 예측, 말월에는 마이너스 성장인 -0.09%로 예측함. 팬데믹이 확산되는 2분기에 들어서는 점차 예측치를 하향 조정하여 모형은 6월말 기준 -1.45% 성장률을 예측함

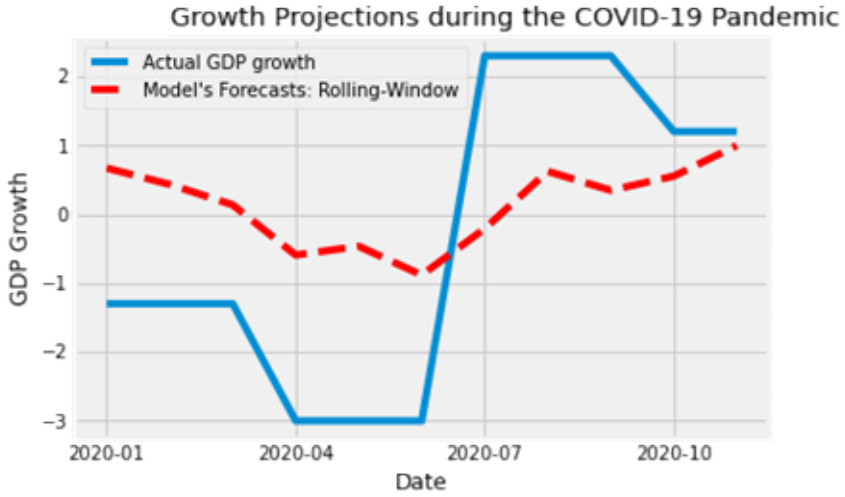
- 롤링윈도우 예측 경우, 2020/01 기준 1분기 성장률 예측을 0.66%로 기록하였다가, 말월인 3월에도 여전히 플러스 성장인 0.13%로 예측함. 팬데믹이 확산되는 2분기에 들어서는 점차 예측치를 하향 조정하여 6월말 기준 -0.88% 성장률을 예측함

○ 두가지 예측 방법 모두 2분기 이후 경기반등을 예상하는 예측치를 산출하고 있으나 실제 실현된 값과는 차이를 보임

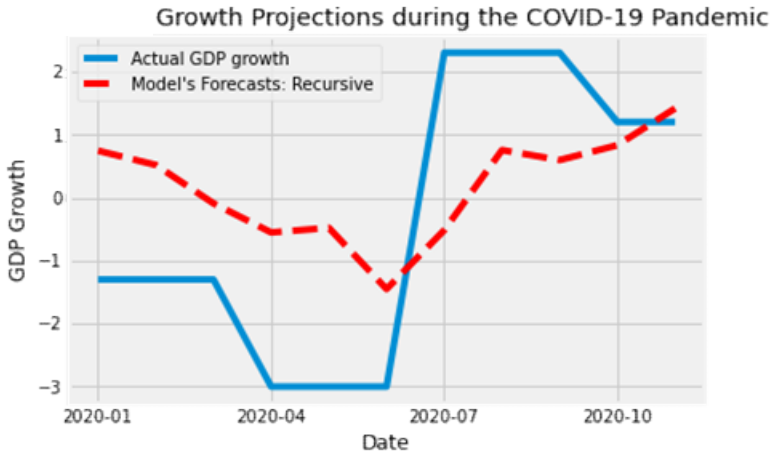
- 다만, 2020년 3분기 성장률은 기저효과로 인해 2.3%대의 분기 성장률을 기록하고 이후 마지막 분기에는 1% 성장률로 안정화되는데, 모형은 12월 기준 실제

값과 가장 유사한 예측치를 산출하는 경향을 보임

<그림 18> 2020년도 예측 양상: 롤링윈도우 예측



<그림 19> 2020년도 예측 양상: 축차예측



○ 모형은 팬데믹 기간 성장률의 실제 경로와 유사한 패턴을 보이는 예측치를 산출해내고 있지만, 관측된 성장률과 차이를 보이고 있음

- 이는 코로나-19 팬데믹과 같은 이벤트가 인공신경망의 학습 자료(training data)에 다수 포함되어 있지 않은 아주 희귀한 사건(rare event)에 속한 것에 기인하는 것으로 보임

- 과거 글로벌 금융위기(GFC) 여파로 인해 우리나라는 2008년 4분기 -3.3% 분기 성장률을 한 차례 기록했음

- 또한 동 예측모형의 경우 예측치를 산출하는데 있어 DFM이 예측한 미래의 월별/분기별 자료를 사용하기 때문에 불확실성이 더욱 가중된 것에 기인할 수도 있음

- 향후 모형개발에는 코로나-19 팬데믹 기간이 인공신경망의 학습자료로 포함될 수 있기 때문에 모형의 예측력이 향상될 것으로 기대

[부록 2: 나우캐스팅(nowcasting) 변수 활용 모형 결과]

❑ 본 부록에서는 통계청에서 발표하는 속보성 사회경제지표 관련 변수를 예측모형에 추가할 시 모형의 예측력에 미치는 영향력을 점검 (<표 38> 참고)

○ 통계청이 발표하는 속보성 사회경제지표는 일반적인 자료에 비해 발표 주기가 짧아 나우캐스팅에 적절히 활용될 가능성이 높음

- 다만, 해당 자료는 2020년 1월부터 존재하기 때문에 학습데이터의 양이 적은 문제가 있음

- 따라서 본 부록 모형의 결과는 실험적 성격의 결과로 이해하는 것이 적절함

<표 38> 통계청 발표 속보성 사회경제지표 관련 변수

변수명	변수 설명	LAG	DIFF	GROUP	FREQ
CARD_SALES	가계 신용카드 이용금액	0	0	REAL	W
CASH_SALES	가맹점 현금매출액	0	0	REAL	W
CREDIT_EXP	가맹점 카드매출액	0	0	REAL	W
UNEMP_PAY	실업급여지급건수	0	0	LABOR	W
JOB_AD	온라인채용 모집인원수	0	0	LABOR	W
MOBILE	모바일통신이동량	0	1	REAL	W

주: 모바일통신 이동량을 제외하고는 2020년 1월 대비 증가율을 나타냄

❑ 본 부록에서의 예측모형은 [부록 1]의 모형의 세부적인 구조는 동일하고, 주간단위로 발표되는 속보성 사회경제지표를 월별 주기로 전환하여 설명변수 집합에 포함시킴

○ 또한, 표본 외 예측 기간은 자료의 제약으로 인해, 2020/02~2021/12로 한정함

- 따라서 표본 외 예측의 RMSE, MAE 계산에 표본 수가 상대적으로 적어, 본문 모형의 결과와 [부록 1]의 모형 결과와의 직접적인 비교는 적절치 않음

❑ 속보성 사회경제지표가 설명 변수에 포함된 예측 모형의 경우 전반적으로 본문의 모형에 비해 RMSE와 MAE가 비교적 높게 관측됨 (<표 40>, <표 41> 참조)

○ 당분기 축차 예측의 경우, 전반적으로 GRU와 LSTM간 예측 성과에서 큰 차이는 관찰되지 않음

- 다만 흥미로운 점은, 예측 시계가 멀어질수록 GRU, LSTM 예측 모형 모두 RMSE가 개선되는 양상을 보임

- GRU 모형의 경우 3분기 이후 성장률 예측에 대한 축차예측 RMSE가 약 0.7, LSTM의 축차예측 RMSE가 약 0.75로써 본문의 모형과 비교해 예측성과가 크게 개선되었음

- GRU 모형의 경우 3분기 이후 성장률 예측에 대한 축차예측 MAE가 약 0.6, LSTM의 축차예측 MAE가 약 0.68로써 본문의 모형과 비교해 예측 성과가 크게 개선되었음

- 다만, 비교적 짧은 예측 시계의 예측 오차는 본문의 모형과 비교해 크게 증가함

○ 동 모형의 표본 외 예측 기간은 코로나-19 기간만을 포함하여 단기 예측 시계 관점에서 속보성 사회경제지표가 예측성과에 유의미한 개선을 가져오는데 어느 정도 제약이 존재함을 확인 할 수 있음

- 인공신경망 학습 과정에서 동 자료의 학습 데이터(training data)의 다수가 결측치로 처리되어 있어 불확실성이 더욱 가중된 사실에 기인하는 것으로 보임

- 다만, 비교적 장기 시계에서의 예측성과를 크게 개선시킨다는 현상에 비추어 보면, 동 변수들이 담고 있는 정보가 어느 정도 시차를 가지고 국민경제활동에 영향을 미치는 것으로 해석할 수 있음

- 따라서 향후 관련 데이터가 더욱 축적됨에 따라 속보성 사회경제지표와 전통적인 저빈도 자료를 결합한 예측모형에 대한 연구가 필요해 보임

<표 39 > 속보성 지표 포함 모형 표본 외 예측:

(a) RMSE

<표 40> 속보성 지표 포함 모형 예측오차: RMSE

	GRU	LSTM
축차 예측		
y(t)	1.6251	1.7110
y(t+1)	1.4614	1.5299
y(t+2)	0.9069	1.0396
y(t+3)	0.6971	0.7489
롤링윈도우 예측		
y(t)	1.6307	1.6094
y(t+1)	1.3958	1.6044
y(t+2)	0.8515	1.1165
y(t+3)	0.7291	0.8538

(b) MAE

<표 41> 속보성 지표 포함 모형 예측오차: MAE

	GRU	LSTM
축차 예측		
y(t)	1.1062	1.2681
y(t+1)	1.0463	1.1611
y(t+2)	0.7576	0.8063
y(t+3)	0.5343	0.6363
롤링윈도우 예측		
y(t)	1.1264	1.2804
y(t+1)	0.9272	1.2447
y(t+2)	0.7114	0.8959
y(t+3)	0.6065	0.6772

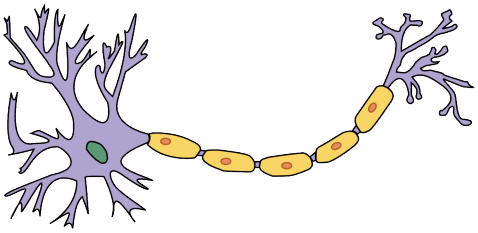
주: GRU: Gated Recurrent Unit. LSTM: Long-Short Term Memory.

[부록 3: 인공신경망에 대한 이해]

가. 생물학적 뉴런과 생물학적 신경망(Biological Neural Network; BNN)

(1) 생물학적 신경세포(neuron)

<그림 20> 생물학적 뉴런



❑ 짧은 전기 자극인 신호을 생성하여, 화학적 신호인 신경 전달물질(neurotransmitter)을 발생하는 역할로, 하나의 생물학적 뉴런은 단순히 신호 생성 및 전달을 수행

(2) 생물학적 신경망(BNN)

❑ 수십억 개의 뉴런들이 서로 네트워크를 형성하여 생물학적 신경망을 이루면, 매우 복잡한 계산 가능

나. 인공뉴런(Artificial neuron)과 인공 신경망 (Artificial Neural Network; ANN)

(1) 인공 뉴런

❑ 하나 이상의 이진 입력과 하나의 이진 출력으로 작동

<그림 21> 뇌 신경망과 인공 신경망 구조

(2) 인공 신경망

❑ 뇌 신경세포인 뉴런의 네트워크에서 영감을 받은 모델로, 딥러닝의 핵심이며 확장성이 좋아 복잡한 대규모 머신러닝 문제를 다루는 데 적합

❑ 대표적인 사용 예시로 유튜브 동영상 추천 알고리즘, 구글 이미지 분류 등

(3) 인공 신경망의 역사

❑ 1943년 신경생리학자 워런 매컬러와 수학자 월터 피츠가 최초의 인공 신경망 구조를 발표[38] 했지만, 발표 이후 인공 신경망 모델은 오랫동안 침체기였다가 최근에 다시 부흥

❑ 부흥 요인으로는 모델을 훈련시킬 방대한 데이터 축적, 하드웨어 발달, 알고리즘의 향상 등

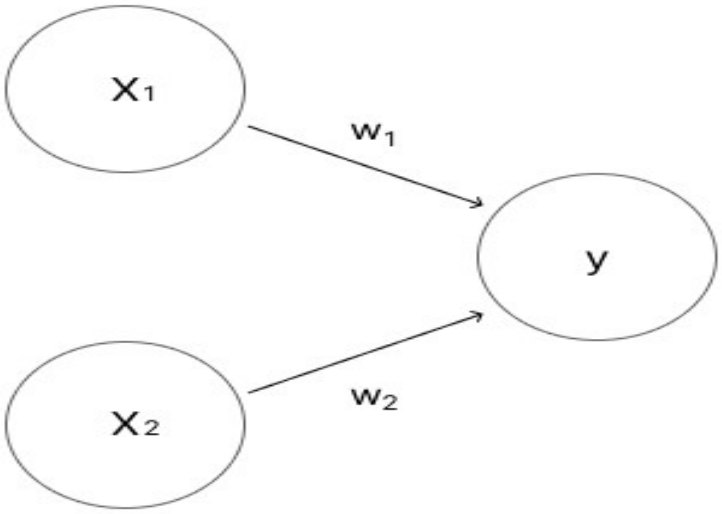
다. 퍼셉트론

(1) 퍼셉트론 (Perceptron)

❑ 인공 신경망의 기원이 되는 알고리즘으로서, 다수의 신호를 입력으로 받아 하나의 신호를 출력하고 퍼셉트론 신호는 '흐른다/흐르지 않는다'로 나뉘고 각각 1과 0으로 표현

38) "A logical calculus of the ideas immanent in nervous activity" (Warren S. McCulloch & Walter Pitts, 1943)

<그림 22> 입력 신호가 두 개인 퍼셉트론 예시



❑ 위 그림에서 입력된 다수의 신호는 x_1과 x_2이고, 출력신호는 y, 가중치는 w_1과 w_2이며 가중치는 각각 고유한 값을 부여받고, 각 신호의 영향력을 조정하는 모델 훈련의 핵심적인 역할

○ 가중치 값이 커질수록 해당 신호의 중요도가 커지는데, 예를 들어 가중치 $w_1 = 0.00001$이라고 한다면, 입력값 x_1이 아무리 크더라도 가중합 w_1x_1은 0에 가까운 값

❑ 위 그림에서 원을 뉴런 또는 노드(node)이며, 출력 뉴런 y는 이진 판단을 통해 출력 결정

❑ 신호 흐름 구조

○ 입력 신호 x_1와 x_2가 다음 뉴런으로 흘러갈 때, 가중치 w_1과 w_2와 각각 곱하여 x_1w_1과 x_2w_2 형태로 신호 도착하고, 이 x_2w_2 가중치와 신호를 곱한 것을 가중합이라고 지칭. 입력 뉴런에서 흘러온 신호의 총합이 임계값(θ)에 따라 양성 클래스(1) 또는 음성 클래스(0)를 출력

$$y = \begin{cases} 0 \ (w_1x_1 + w_2x_2 \le \theta) \\ 1 \ (w_1x_1 + w_2x_2 > \theta) \end{cases}$$

····· (35)

○ 위의 퍼셉트론 로직에서 입계값 θ을 b로 치환하면, b는 가중합에 더해지는 상수인 편향

$$y = \begin{cases} 0 & (b + w_1x_1 + w_2x_2 \leq 0) \\ 1 & (b + w_1x_1 + w_2x_2 > 0) \end{cases}$$

····· (36)

○ 즉, 입력 신호에 가중치(w) 곱하고 편향(b)을 더한 값이 0보다 크면 1을 출력하고, 작으면 0을 출력하는 로직으로 작동

(2) 퍼셉트론 논리회로

❑ 퍼셉트론 논리회로는 가중치와 임계값(편향)에 따라 출력이 결정됨. 즉, 페셉트론의 구조는 동일하고, 매개변수 값 조정에 따라 게이트가 교체

<그림 23> 퍼셉트론 진리표 예시 1: AND 게이트

x_1	x_2	y
0	0	0
1	0	0
0	1	0
1	1	1

<그림 24> 퍼셉트론 진리표 예시 2: NAND 게이트

x_1	x_2	y
0	0	1
1	0	1
0	1	1
1	1	0

<그림 25> 퍼셉트론 진리표 예시 3: OR 게이트

x_1	x_2	y
0	0	0
1	0	1
0	1	1
1	1	1

<그림 26> 퍼셉트론 진리표 예시 4: XOR 게이트

x_1	x_2	y
0	0	0
1	0	1
0	1	1
1	1	0

❑ AND 게이트는 두 입력 x_1, x_2가 모두 1인 경우, 1을 출력, 그 외는 0을 출력하는 조건을 충족시키는 다양한 매개변수(가중치, 편향)의 조합 설정

○ 예시

조합을 $(x_1,\ x_2,\ \theta)$ = (0.3, 0.3, 0.5)라고 설정하면,

$0.3*0 + 0.3*0 \leq 0.5$ => 0 출력

$0.3*1 + 0.3*0 \leq 0.5$ => 0 출력

$0.3*0 + 0.3*1 \leq 0.5$ => 0 출력

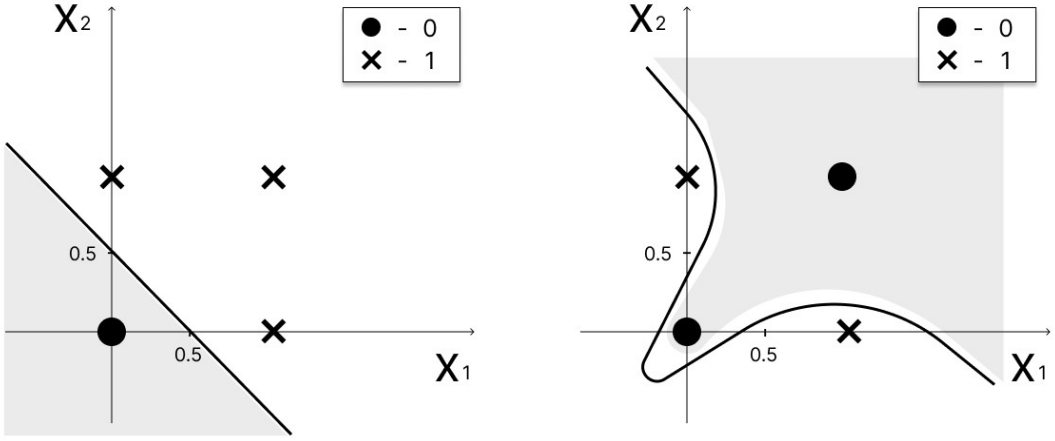
$0.3*1 + 0.3*1 > 0.5$ => 1 출력이므로 이 조합은 AND 게이트 조건을 만족

❑ NAND 게이트는 AND 게이트 출력 반전의 형태로, 두 입력 x_1, x_2가 모두 1인 경우, 0을 출력, 그 외는 1을 출력

❑ OR 게이트는 두 입력 중 하나 이상이 1인 경우, 1을 출력, 그 외는 0을 출력

❑ XOR 게이트는 두 입력 중 하나만 1인 경우, 1을 출력, 그 외는 0을 출력하는 조건이며, 단층 페셉트론 구조에서는 구현 불가능

<그림 27> 퍼셉트론 시각화 (왼: OR게이트 조건, 우: XOR게이트 조건)

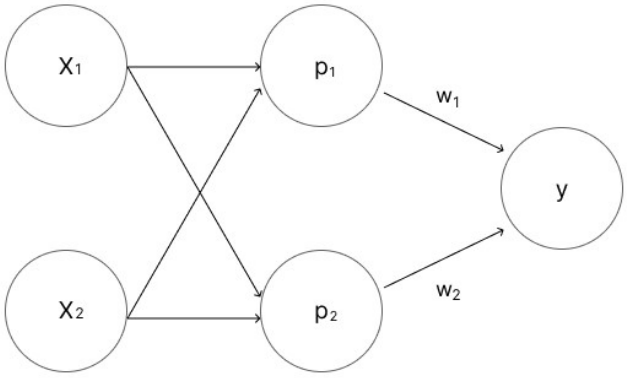


○ 단층 페셉트론을 시각화했을 때, OR 게이트와 달리 XOR 게이트는 선형으로 출력 0과 1을 영역 분리 불가능하지만, 비선형적으로 XOR 게이트 출력 영역 구분 가능

(2) 다층 퍼셉트론 (Multi-Layer Perceptron; MLP)

❑ 퍼셉트론을 순차적으로 여러 층 쌓은 구조로, 단층 퍼셉트론을 여러 층 쌓아 비선형적으로 분리되는 XOR 게이트를 표현할 수 있는 구조

<그림 28> XOR 퍼셉트론



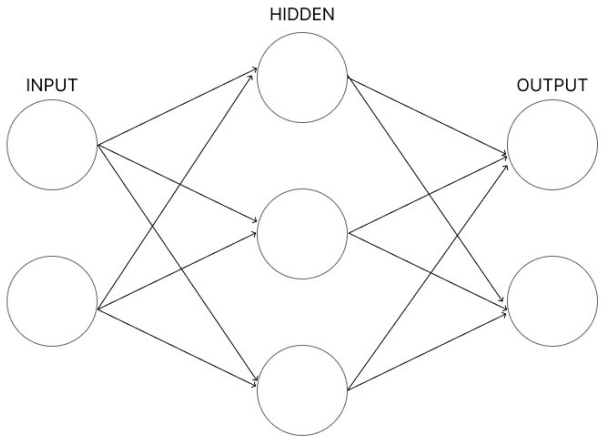
❑ x층에서 데이터 입력이 들어오고 p층을 거쳐 y층에서 출력하는 작동하며, 층을 더 늘리면서 단층 퍼셉트론에서 구현하지 못한 것을 표현 가능. 즉, 층이 여러 겹일수록 더 큰 비선형성을 구현 가능

라. 신경망

(1) 신경망 층 구조

❑ 다층 퍼셉트론 구조와 유사한 형태를 가짐. 인접한 층의 뉴런과 연결되어있으며, 동일한 층의 뉴런과의 연결되지 않음. 심층 신경망에서는 x층은 입력층, p층은 은닉층, y층은 출력층이 됨.

<그림 29> 신경망 구조 예시



❑ 입력층, 은닉층, 출력층으로 구성되어 있음. 입력층은 입력 데이터를 받아들이는 한 겹의 층이며, 은닉층은 입력 뉴런으로부터 받은 입력 데이터가 출력층으로 가기 전에 거치는 모든 층, 그리고 출력층은 은닉층에서 전달된 최종 결과값을 포함하는 한 겹의 층

(2) 신경망 신호 흐름 구조

❑ 다층 퍼셉트론과 가장 큰 차이점인 활성화 함수가 존재

○ 페셉트론 신호 흐름 수식

$$y = \begin{cases} 0 & (b + w_1x_1 + w_2x_2 \le 0) \\ 1 & (b + w_1x_1 + w_2x_2 > 0) \end{cases}$$

····· (37)

○ 신경망 신호 흐름 수식

$$y = h(b + w_1x_1 + w_2x_2)$$

$$h(x) = \begin{cases} 0\,(x \le 0) \\ 1\,(x > 0) \end{cases}$$

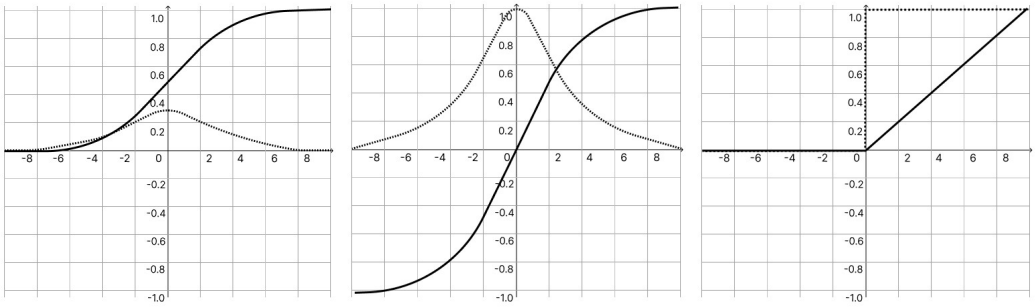
····· (38)

❑ 활성화 함수 (Activation Function)

○ 가중합 wx에 편향 b를 더한 값을 출력신호로 변환하는 비선형 함수 h, 즉 합성함수 형태. 가중합에 활성화 함수를 적용하는 이유는 모델의 비선형성 극대화 목적

(3) 대표적인 활성화 함수

<그림 30> 활성화 함수 형태: 시그모이드, 하이퍼볼릭 탄젠트, ReLu (왼쪽부터)



39)

❑ 시그모이드 함수(sigmoid; σ)는 출력 범위가 0에서 1 사이이며, 확률 표현에 적합한 함수

$$\sigma(z) = \frac{1}{1+e^{-z}} \qquad \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \quad (39)$$

❑ 하이퍼볼릭 탄젠트 (hyperbolic tangent; tanh) 함수는 출력 범위가 -1에서 1 사이이며, 주로 시계열 모델의 은닉층 활성화 함수로 사용

$$\tanh(z) = 2\sigma(2z) - 1 \qquad \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \quad (40)$$

❑ ReLU 함수는 출력 범위가 양수 신호는 그대로 출력, 음수 신호는 0으로 출력함. 작동 원리가 간단하고, 기울기가 1로 일정하기 때문에 학습속도가 빠르고, 다양한 ReLU 함수의 변형된 함수 존재 Leaky ReLU, LRel, GeLU 등

39) 점선은 활성화 함수의 도함수이다.

$$ReLU(z) = \max(0, z)$$

····· (41)

(4) 신경망 각 층에 따른 신호 흐름 구조

❑ 입력층(Input Layer)은 입력 뉴런으로 형성되어있으며, 입력 뉴런은 데이터 형태와 상관없이 바로 다음 층(layer)으로 흘려보냄

❑ 은닉층(Hidden Layer)은 주로 다수의 층으로 구성되고, 각 은닉층은 고유의 노드 개수 가짐. 은닉층 수와 노드 개수가 많을수록, 모델의 복잡성(complexity) 증가

○ 은닉층의 계산 수식

$$h_1 = \sigma_1(x\ W^1 + b^1)$$

$$h_2 = \sigma_2(x\ W^2 + b^2)$$

····· (42)

○ 은닉층 계산은 가중합과 편향의 합이 활성화 함수 σ를 거쳐 다음 층으로 넘겨짐

❑ 출력층(Output Layer)은 은닉층에서 넘겨받은 가중합을 출력. 일반적인 회귀문제에서는 출력층에 활성화 함수를 사용하지 않지만, 분류 문제에서는 이를 사용하여 최종값을 출력

$$\hat{y} = h_2\ W^3 + b^3$$

····· (43)

(3) 심층 신경망 (Deep Neural Network; DNN)과 딥러닝(Deep Learning)

❑ 여러 개의 은닉층으로 쌓은 인공 신경망을 의미하며, 신경망의 은닉층이 1개 이하이면 shallow network, 2개 이상이면 Deep network라고 지칭

❑ 딥러닝은 심층 신경망을 학습시키는 과정으로, 넓은 의미로는 심층 신경망을 연구하는 분야

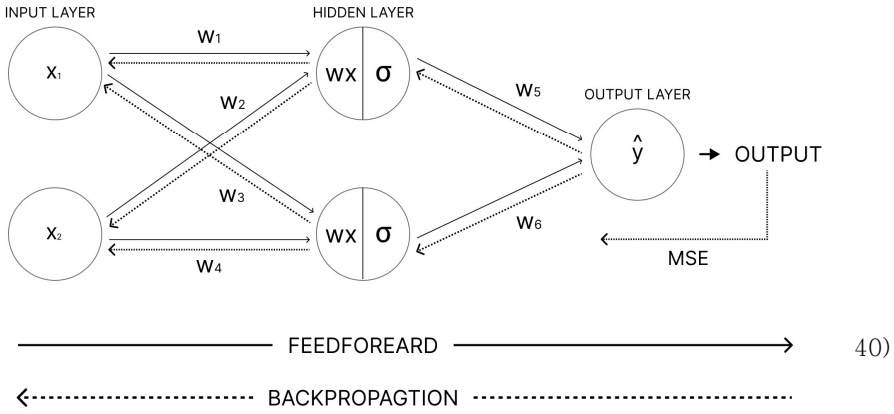
라. 신경망 학습

(1) 의미

❑ 수동으로 지정한 퍼셉트론 가중치(매개변수)를 자동으로 최적의 가중치를 찾는 과정. 자동으로 적절한 매개변수를 찾게 하기 위해서는 학습 데이터를 통한 기계학습이 필요. 즉, 주어진 훈련 데이터로부터 특징을 추출하고, 그 특징의 패턴을 찾아나가는 과정

(2) 순전파(Feedforward)와 역전파(Backpropagation)

<그림 31> 순전파 학습과 역전파 학습방향 및 과정



40)

40) wx는 입력 신호인 가중합이며, 활성화 함수 σ를 취한 후 y를 출력한다.

❑ 신경망 학습은 순전파와 역전파 두 단계로 진행

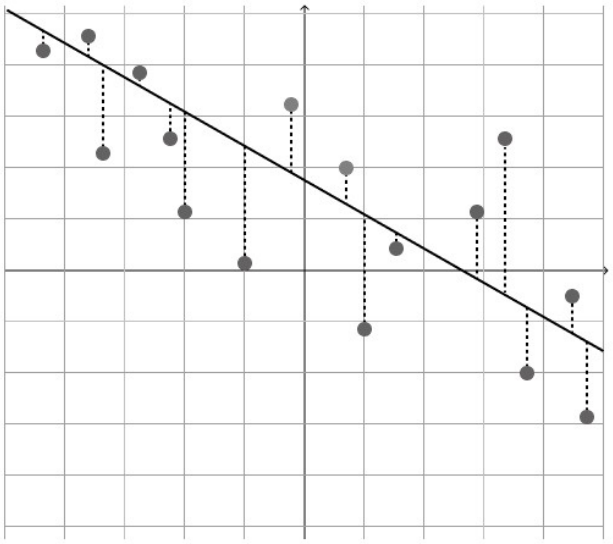
- ㅇ 순전파는 모델에 훈련 데이터가 입력되고 전체 신경망을 지나가면서 모든 뉴런이 계산을 완료하면 예측값이 출력되는 과정으로, 데이터를 기반으로 예측값(output)을 추론하는 과정
- ㅇ 역전파는 순전파 과정이 진행되고 나서, 모델의 예측값과 실제값의 차이인 예측오차(Loss)를 계산하고, 예측오차를 최소화하기 위해 모델의 훈련을 반복하면서 '가중치를 조정'

❑ 즉, 순전파는 입력층 → 은닉층 → 출력층 순이라면, 역전파는 출력층 → 은닉층 → 입력층 순으로 작동

(3) 손실함수(Loss Function)

❑ 여러 가중치의 조합을 가진 각 모델의 최적화 정도를 평가하는 기준이면서, 모델의 성능을 측정하는 지표. 실제값인 데이터와 예측함수의 차이를 잔차(error)라고 하는 데, 이 잔차의 합을 최소화는 예측함수를 구하는 것이 목적. 즉, 각 데이터에 대한 잔차의 합을 가장 작게 만드는 예측 함수 내의 모수(parameter)를 탐색

<그림 32> 실제 값과 예측값 차이



❑ 잔차의 합은 보통 각 잔차를 제곱하여 양수화[41]하고 모두 더하는 방식인 RSS(Residual Sum of Square)를 취함. RSS는 아래와 같은 수식으로 표현

$$
\begin{aligned}
RSS &= \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y})^2 \\
&= \frac{1}{N}\sum_{i=1}^{N}(y_i - (w_0 + w_1 x_i))^2
\end{aligned}
\qquad \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots \quad (44)
$$

○ 여기서 y는 실제값인 데이터, 예측함수 $\hat{y}$는 절편 w_0, 기울기 w_1와 데이터 x_i로 구성. 학습을 통해 RSS를 최소화하는 w_0와 w_1를 구하는 것이 목적이며, RSS는 손실함수라고 함.

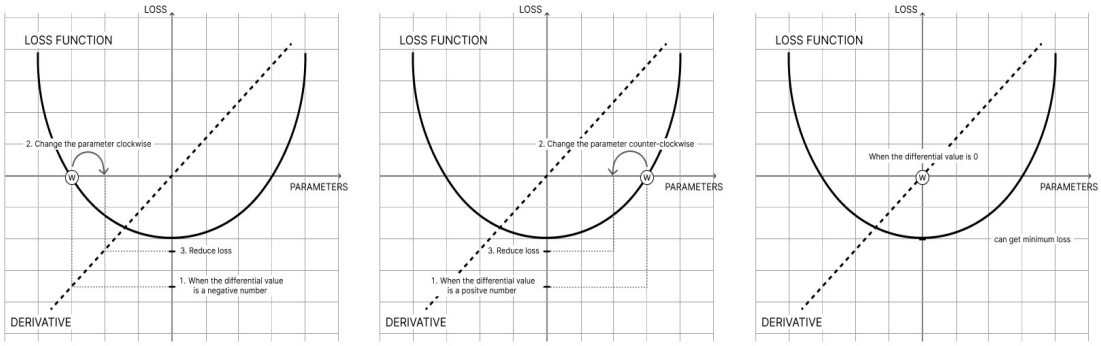
○ 손실 함수값(Loss)을 최소로 만들는 모델의 매개변수(가중치와 편향) 조합이 데이터에 최적화된 상태이고, 그 매개변수 조합을 찾는 것이 역전파 학습의 목적

❑ 가중치 매개변수의 손실함수 미분을 단서로, 매개변수 값의 조금씩 수정하며 최적의 조합을 탐색. 즉, 가중치 매개변수의 값을 매우 조금 변화시켰을 때, 손실함수는 어떻게 변하는가가 열쇠가 됨

○ 예를 들어 미분 값이 음수이면, 양의 방향으로 가중치 매개변수를 갱신하여 손실 함숫값을 줄일 수 있으며, 미분 값이 양수이면, 음의 방향으로 가중치 매개변수를 갱신하여 손실 함숫값을 줄일 수 있음. 미분 값이 0이라면, 최소 손실값을 얻고 가중치 매개변수 갱신 중단

41) 제곱을 하지 않으면 상쇄되어 잔차의 합은 0이 된다.

<그림 33> 손실함수의 도함수와 매개변수 관계

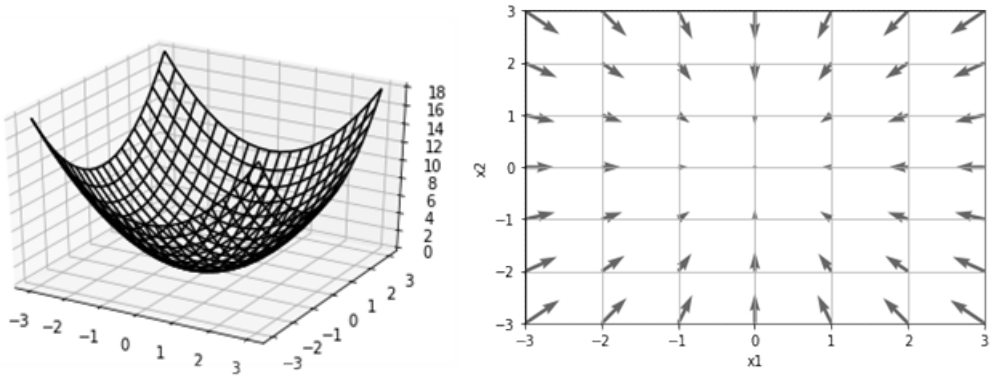


○ 모든 데이터별로 편미분을 통해 최소 손실 함숫값을 찾을 수 있지만, 변수들 하나하나의 편미분 값을 구하는 것은 불가능

❑ 동시에 모든 데이터의 편미분을 동시에 계산하기 위해, 모든 변수의 편미분을 벡터로 정리한 값을 기울기 벡터(gradient vector)[42)]

○ 예를 들어 $f(x_1, x_2) = x_1^2 + x_2^2$ 함수를 그래프로 나타내면 <그림 11>의 좌측과 같은 형태이며, <그림 11>의 우측은 각 데이터에서 편미분을 통해 구한 그래디언트(gradient)를 음의 방향으로 표현한 벡터장

<그림 34> $f(x_1, x_2) = x_1^2 + x_2^2$ 시각화와 벡터장



42) 기울기 벡터와 기울기(gradient)를 구분하지 않고 사용한다.

❑ 각 그래디언트(gradient)의 음의 방향이 각 데이터에서 함수의 출력값을 최소화하는 방향. <그림 34> 우측에서 각 벡터의 방향이 가리키는 함수의 최솟값 방향이며, 최솟값에서 가까워질수록 화살표의 크기가 작아짐

○ 즉, 손실 함숫값이 최소가 되는 매개변수를 찾기 위해서는 그래디언트 벡터의 음의 방향으로 이동하면 효율적인 탐색이 가능

❑ 대표적인 오차제곱합(MSE)과 교차 엔트로피(cross entropy)를 손실함수로 사용함. 오차 제곱합은 회귀 문제에서 가장 많이 사용되는 손실함수

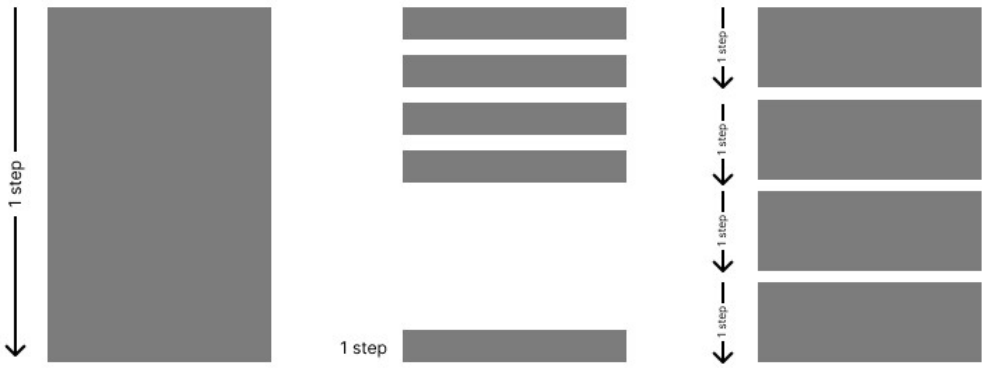
(4) 반복법 (Iterative method)

❑ 최소의 손실함수(loss function) 값을 갖는 가중치 조합을 탐색하기 위해 여러 경우의 수를 실행하기 위한 방법으로, 손실함수(loss function) 값을 기준으로 반복적으로 가중치를 업데이트하여, 점차 모델의 성능을 개선

❑ 신경망 모델을 학습하기 위해서는 반복법을 사용한 수치최적화 모델 선택을 통해 최적화 단계 수행

(5) 경사 하강법 유형과 배치 학습

<그림 35> 경사하강법, 확률적 경사하강법, 미니배치(mini batch) 방법 원리



❑ 배치 경사 하강법(Batch Gradient Descent)은 전체 훈련 데이터에 대한 가중치를 편미분을 하여 전체 데이터에 대한 손실함숫값 계산 후, 기울기를 한 번 구하여 모델의 파라미터를 갱신하는 방법. 정확한 손실함수의 그래디언트(gradient)를 구할 수 있다는 장점이 있지만, 높은 computation 시간 소요

❑ 확률적 경사 하강법(Stochastic Gradient Descent)은 최소한의 데이터로 최대한 정확한 손실함수의 그래디언트(gradient)를 구하는 방법. 임의로 선택한 일부 데이터만 사용하여 계산하므로 비용은 낮지만, 정확도가 상대적으로 낮음

❑ 미니 배치 방법(Mini-batch Gradient Descent)에서 배치(Batch)는 그래디언트(gradient) 업데이트를 위해 사용될 데이터 개수 또는 사이즈. 전체 데이터를 미니 배치 사이즈로 여러 개로 나눈 후, 각 미니 배치마다 기울기를 계산하고 평균 기울기로 매개변수 갱신

○ 배치의 크기가 클수록 그래디언트(gradient) 계산 정확도는 높아지지만, 속도는 느려지므로 적절한 배치크기로 설정하여 미니배치 학습을 실시하는 것이 핵심

마. 신경망 시계열 분석

❑ 시간 연속성을 가진 데이터를 처리하기 위해 고안된 인공 신경망으로, 대표적인 신경망 모델로는 순환 신경망 (Recurrent Neural Network; RNN)와 장단기 메모리 (Long Short-Term Memory; LSTM)가 있음

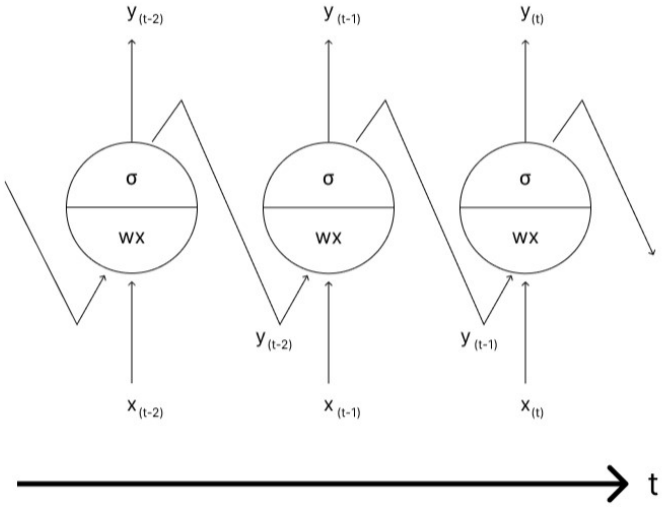
(1) 순환 뉴런

❑ 일반적인 신경망은 입력층에서 출력층으로 일방통행으로 흐르는 연결 구조이지만, 순환 신경망은 같은 층의 뉴런 사이에 연결도 존재하는데 순환 뉴런을 통해 이뤄짐

○ 아래 그림에서 순환 뉴런은 입력 신호 x를 계산하여 출력신호 y를 다음 층에 전달하는 동

시에, 다음 timestep에도 자신에게 출력을 전달

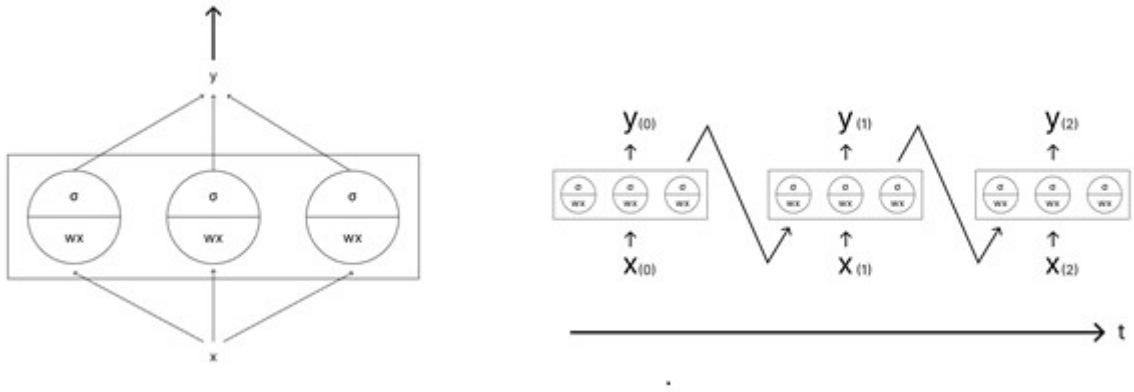
<그림 36> Timestep에 따른 순환뉴런 흐름



○ 즉, 순환 뉴런은 각 타임스텝에 따라 입력 신호 x와 이전 타임 스텝의 동일한 뉴런에게 전달되는 출력신호 받음. 순환 뉴런은 입력 신호 x의 가중치인 w_x와 이전 시간의 출력신호의 가중치인 w_y를 보유

(2) 순환 층

<그림 37> 순환층과 Timestep에 따른 순환층 흐름



❑ 순환 뉴런으로 이루어진 층으로, 각 순환 뉴런들의 출력 y가 모여 순환 층 하나의 벡터 y를 출력함. 순환 층은 입력 벡터 x의 가중치인 W_x와, 이전 시간의 출력신호의 가중치인 W_y를 보유

○ 하나의 데이터에 대한 순환 층 출력

$$y_t = \sigma(W_x^T x_t + W_y^T y_{t-1} + b) \quad \cdots\cdots (45)$$

○ 하나의 미니배치 전체 데이터에 대한 순환 층의 출력

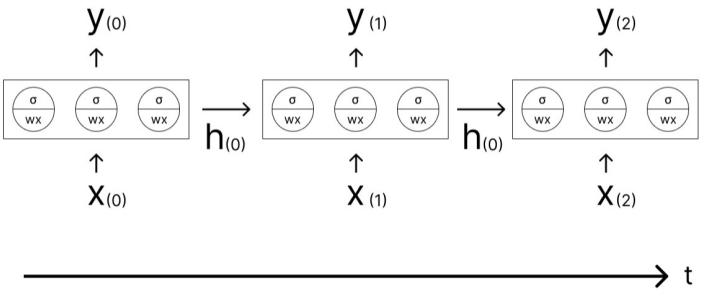
$$Y_t = \sigma([X_t \quad Y_{t-1}]\, W + b)$$
$$W = \begin{bmatrix} W_x \\ W_y \end{bmatrix} \quad \cdots\cdots (46)$$

(3) 메모리 셀 (Memory Cell)

❑ 현재 timstep ‘t’ 입력에서 이전 timestep ‘t-1’의 출력은 과거에 대한 정보를 담고 있는 기억(memory)이라고 볼 수 있음. 즉, 여러 timestep 거쳐 현재까지 입력 데이터를 요약한 정보를 메모리 셀(h)이라 지칭함

❑ 시간순에 따라 새로운 입력이 모델에 입력되면 메모리 셀은 점차적으로 수정되고, 최종 메모리 셀은 전체 입력 데이터를 요약한 정보라고 볼 수 있음

<그림 38> Timestep에 따른 메모리 셀 흐름



❑ h는 현재 timsetep의 입력과 이전 timestep 메모리 셀로 이루어진 함수로, RNN에서는 현재 타임스텝에서 $y_t = h_t$이지만, LSTM에서는 $y_t \neq h_t$

$$h_t = f(h_{t-1}, x_t) \qquad \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots (47)$$

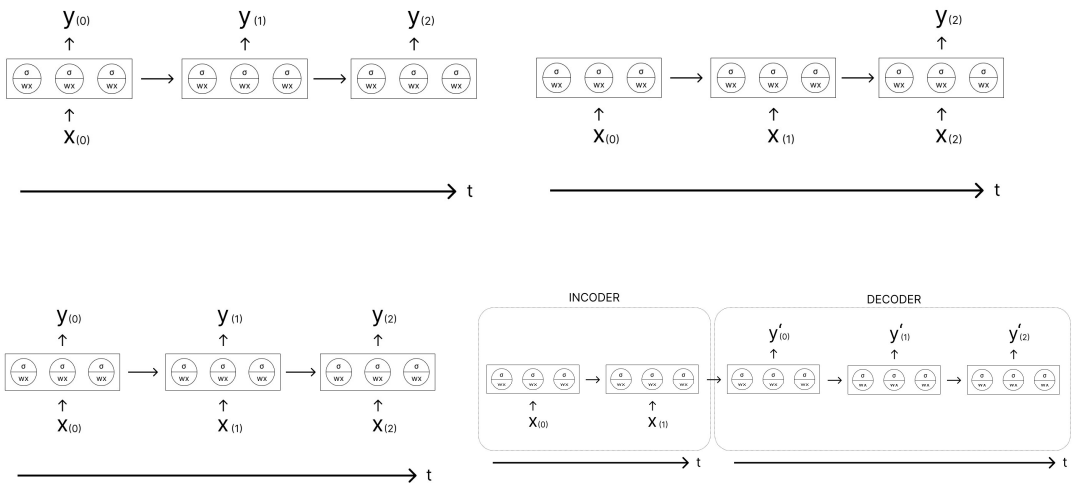
바. RNN (Recurrent Neural Network)

(1) 의미

❑ ‘Recurrent(반복되는)’는 이전 은닉층 노드가 현재 은닉층 노드 입력에 영향을 주면서, 반복적인 순환 구조를 갖는 모델임을 의미함. 시계열 데이터를 처리하는 인공 신경망으로, 회귀문제와 분류 문제를 다루고 가장 기본적인 순환 신경망 구조

(2) 입력과 출력에 따른 유형

<그림 39> 일대다, 다대일, 다대다, 동기화 다대다 구조 (시계방향)



❑ 일대일(vector-to-vector Network) 구조는 순환할 수 없으므로, RNN 구조보다는 순방향 구조에 가까움

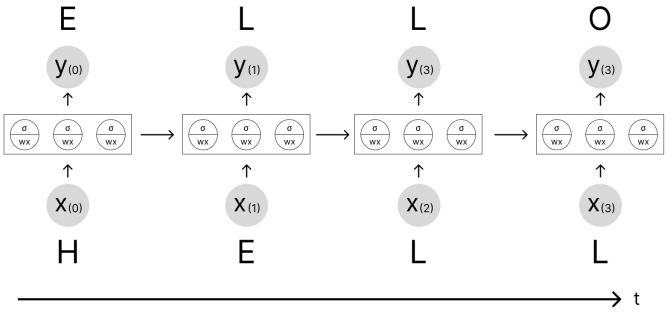
❑ 일대다(vector-to-Sequence Network) 구조는 입력이 하나, 출력이 다수인 구조. 입력 벡터 하나를 반복하여 모델에 넣고 하나의 시퀀스를 출력하며, 대표적인 예시로 이미지를 입력하면 자동으로 이미지 캡션문장 출력

❑ 다대일(vector-to-Sequence Network) 구조는 입력이 다수, 출력이 하나인 구조. 시퀀스(sequence)를 입력으로 넣고 마지막 출력을 제외한 모든 출력은 무시하는 방식으로 작동되며, 대표적인 예시로 자연어 처리에서 문장 감성 분석을 통한 긍정/부정을 출력

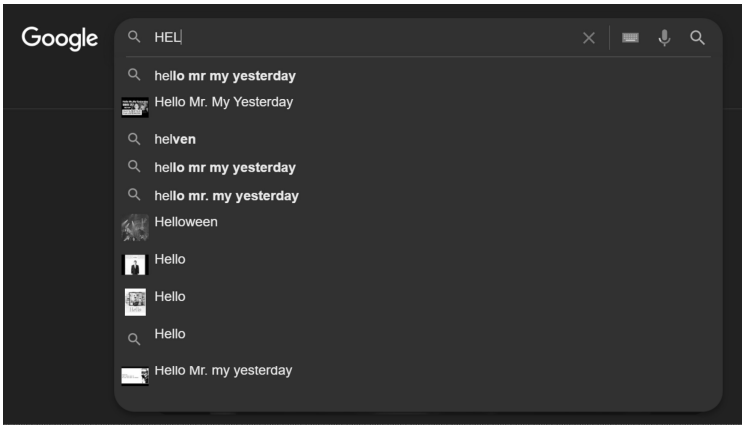
❑ 다대다(Sequence-to-Sequence Network) 구조는 입력과 출력이 다수인 구조. 대표적인 예시로 자동 번역기

❑ 동기화 다대다(Encoder-decoder) 구조는 입력과 출력이 다수인 다대다와 유사한 구조. 인코더인 다대일 구조와 디코더인 일대다 구조가 결합한 형태이며, 대표적인 예시로 자동 검색어 완성

<그림 40> RNN 다대다 구조



<그림 41> 자동검색어 완성 예시

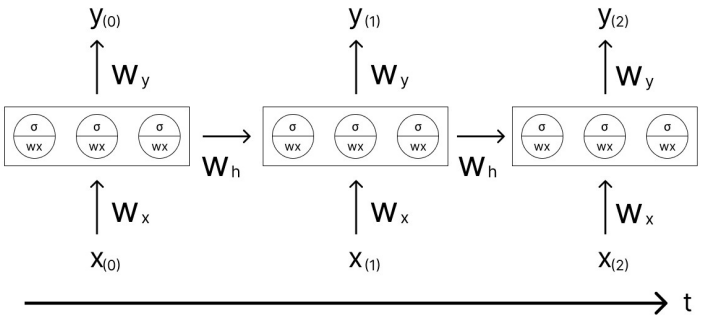


(3) RNN 훈련

❑ RNN의 가중치는 입력층에서 출력층으로 전달과 관련된 가중치 W_x, 이전 시점 은닉층의 메모리셀의 현재 시점 은닉층으로 전달과 관련 있는 W_h, 그리고 은닉층에서 출력층으로 전달과 관련된 가중치인 W_y 세 가지로 분류됨

○ 아래 그림과 같이 각각의 가중치들은 모든 시점에서 동일

<그림 42> RNN 모델에서의 가중치



❑ t 시점에서 은닉층 활성화 함수는 주로 아래 식 (48)과 같이 하이퍼볼릭 탄젠트(hyperbolic tangent; tanh) 함수가 사용됨

$$h_t = \tanh(W_{hh}h_{t-1} + W_{xh}x_t)$$

····· (48)

❑ t 시점에서 RNN 출력층 계산은 분류 문제일 때 일반적으로 소프트맥스(soft max) 활성화 함수가 사용됨

$$\hat{y}_t = softmax(W_{hy}h_t)$$

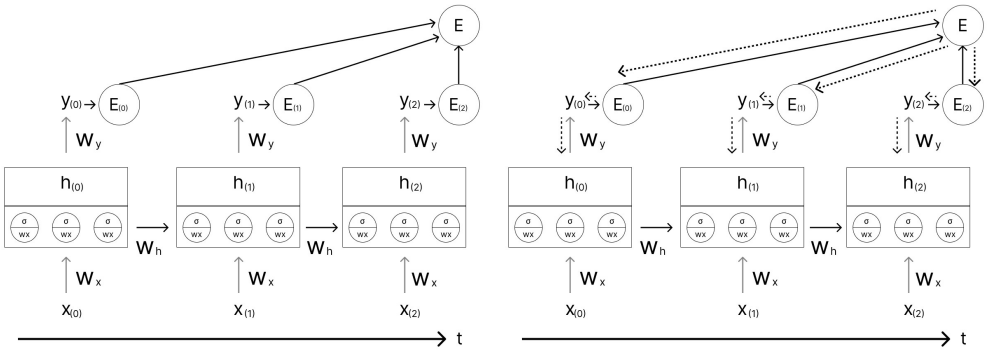
····· (49)

❑ BPTT (Backpropagation through time) 방식

○ 일반적인 심층 신경망과 동일하게 데이터가 정방향으로 전체 신경망을 통과하고, 각 단계 t마다 실제 값(t)과 예측값($\hat{y}_t$)으로 MSE를 사용하여 오차 측정. 즉, 마지막 출력을 포함하여 사용한 모든 출력에서 역방향으로 전파되고, 측정된 오차를 이용하여 가중치 및 편향을 업데이트하는 과정

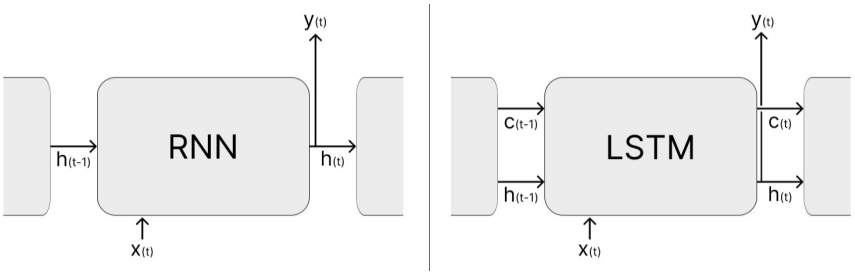
○ 오차가 멀리 전파될수록 “폭주와 기울기 소실문제(exploding and gradient vanishing problem)” 발생 가능성 증가함. 기울기 소실 문제 원인은 가중치 업데이트를 위한 행렬 곱이 역전파 과정에서 반복적으로 수행되기 때문임

<그림 43> RNN 순전파 학습과 역전파 학습



사. LSTM (Long Short-Term Memory)

<그림 44> RNN과 LSTM 출력구조 차이점

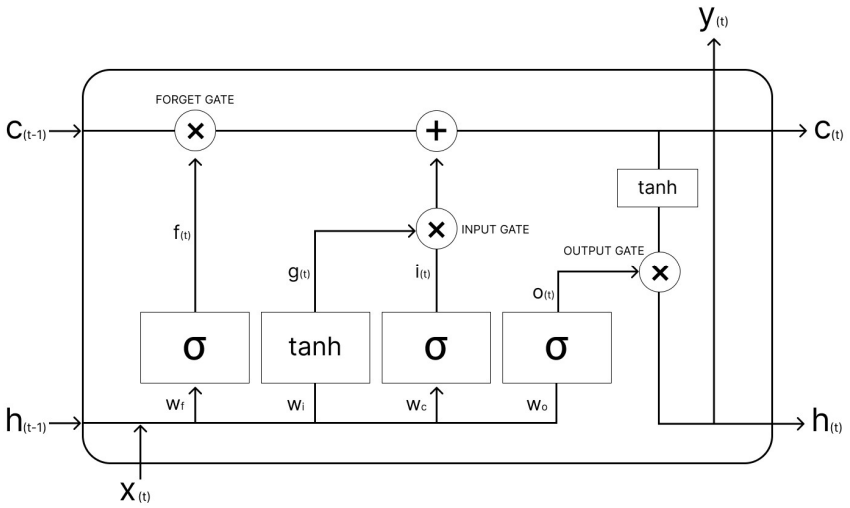


(1) 의미

❑ RNN은 매 단계를 거치면서 과거 정보에 새로운 현재 데이터가 결합하므로, 일부 과거 정보는 소멸하여 초기의 정보들은 마지막 단계에서는 사실상 남아 있지 않음. 즉, 역전파 학습에서 초기 단계일수록 기울기 소멸문제가 발생함. 이러한 단점을 보완하기 위해 초기 정보까지 보존하는 '장기 메모리 셀'을 사용한 모델

(2) LSTM 순전파

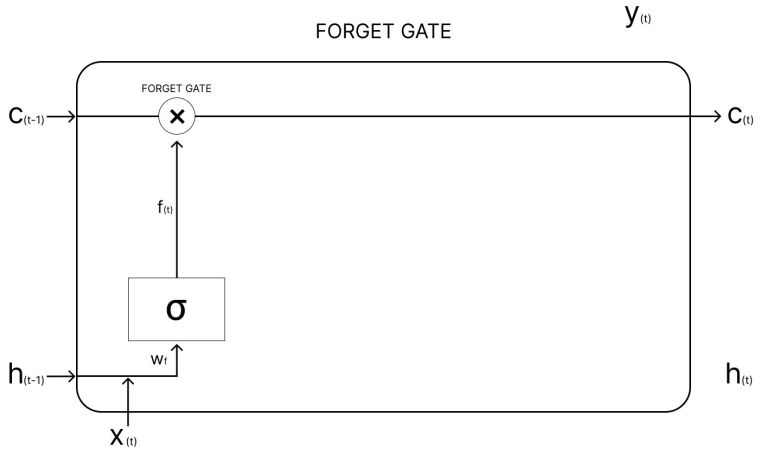
<그림 45> LSTM 계층 구조



❑ 메모리 셀(memory cell)은 각 단계에 대한 은닉 노드이며, 장기 과거정보(또는 장기 메모리 셀; c_t)와 단기 과거정보(또는 단기 메모리 셀; h_t)로 나누어져 있음

❑ LSTM의 각 은닉층(hidden layer)에는 각 뉴런(neuron)에 데이터의 흐름을 제어[43]하는 망각 게이트, 입력 게이트, 출력 게이트가 존재함

<그림 46> LSTM 망각 게이트 구조



43) 흐름을 열기/닫기 또는 열림상태에서 흘려보낼 양을 제어하는 역할이다. 게이트가 제어하는 정도 또한 학습을 통해 자동으로 설정된다.

❑ 망각 게이트 (Forget gate)

○ 망각 게이트는 장기 메모리 셀(c_t)에 이전 시점의 단기 메모리 셀(h_{t-1})의 정보를 어느 정도 반영할지 결정하여, 불필요한 정보를 삭제하는 역할을 수행. 즉, 이전 시점의 과거 단기 메모리 셀(h_{t-1})과 현재시점의 데이터(x_t)를 입력받아 활성화 함수를 취한 값을 이전 시점의 장기 메모리셀(c_{t-1})에 곱하여 현재 시점의 임시 장기 메모리 셀 생성 (식 (50) 참고)

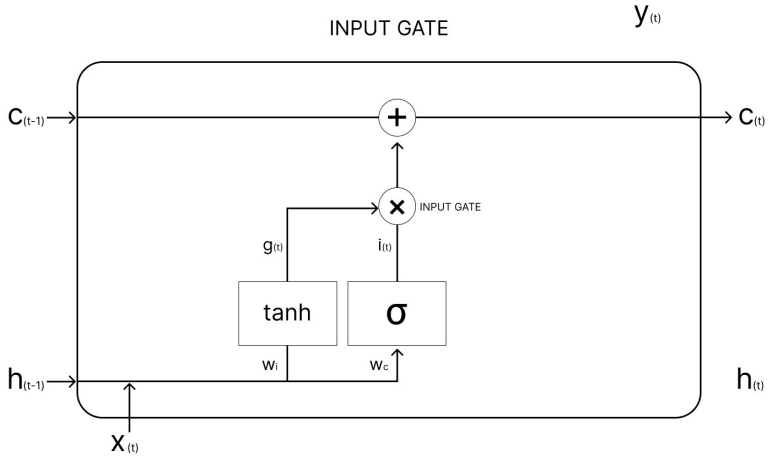
- 여기서 활성화 함수는 sigmoid 활성화 함수를 사용함. 동 활성화 함수 출력값이 0에 가까울수록 *임시* 장기 메모리 셀(즉, 망각 게이트를 거친 장기 메모리 셀; $\tilde{c}_t$)의 생성 과정에 이전시점 장기 메모리 셀(c_{t-1})에 포함된 정보는 적게 반영되고, 1에 가까울수록 많이 반영됨

$$f_t = \sigma(W_{xf} x_t + W_{hf} h_{t-1} + b_f)$$

$$\tilde{c}_t = f_t \, c_{t-1}$$

····· (50)

<그림 47> LSTM 새로운 기억 셀과 입력 게이트(input gate)구조



❑ Main Layer

○ Main Layer에서는 현재 시점의 입력값(x_t)에 포함된 정보를 장기 메모리 셀에 저장하는 역할

을 함으로써, 들어오는 정보의 통과 비율을 정하는 망각 게이트와 다른 기능을 수행

○ Main Layer에 적용된 tanh 활성화 함수는 -1과 1 사이의 값을 갖는 출력값을 도출함으로써, 현재 시점에서 들어오는 정보의 강약을 표시하는 역할로 해석됨. 현재 시점의 입력값은 tanh 활성화 함수를 거쳐 장기 메모리 셀(c_t)에 반영됨 (식 (52) 참조)

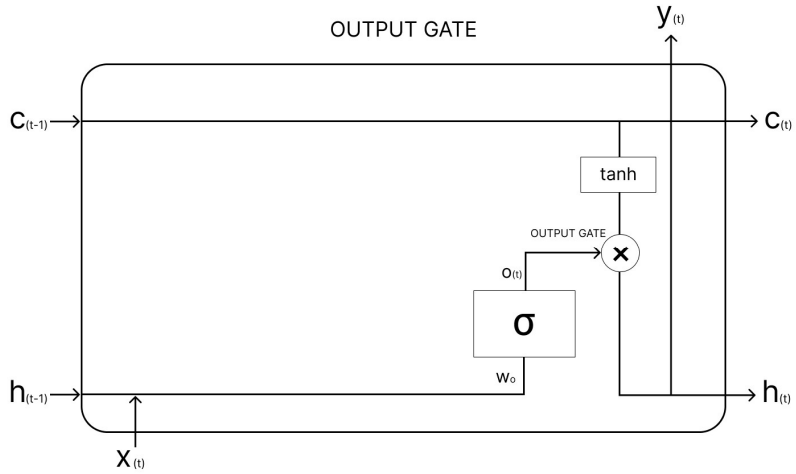
$$g_t = \tanh(W_{xg}x_t + W_{hg}h_{t-1} + b_g) \quad \cdots\cdots (51)$$

❑ 입력 게이트 (Input gate)

○ 입력 게이트(input gate)는 현재 시점 데이터에 포함된 정보를 장기 메모리 셀에 반영하는 정도를 제어하는 역할로, Main Layer에서 도출된 g_t에서 일부 정보만 장기 메모리 셀(c_t)에 반영. 동 게이트의 산출값(i_t)이 1에 가까울수록 현재 시점의 입력 정보(x_t)가 장기 메모리 셀(c_t)에 더 많은 비중으로 반영

$$\begin{aligned} i_t &= \sigma(W_{xi}x_t + W_{hi}h_{t-1} + b_i) \\ c_t &= \tilde{c}_t + i_t g_t \end{aligned} \quad \cdots\cdots (52)$$

<그림 48> LSTM 출력 게이트 구조



❑ 출력 게이트 (Output gate)

○ 최종적인 단기 메모리 셀(h_t) 값의 출력 및 뉴런의 출력값(y_t)를 결정 ($h_t = y_t$)하는 역할 수행. 단기 메모리 셀 (h_{t-1})과 현재 데이터 x_t를 입력받아 sigmoid 활성화 함수를 취한 값과, tanh 활성화 함수를 거쳐 온 임시 장기 메모리 셀($\tilde{c}_t$)의 정보가 결합되어 최종 산출값 도출

- 예를 들어, sigmoid의 출력값이 0.75인 경우는 $\tanh(c_t)$의 3/4 정도의 기억이 y_t 또는 h_t가 됨

$$o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + b_o)$$

$$y_t = h_t = o_t \tanh(\tilde{c}_t)$$

..... (53)

❑ 요약하자면, 장기 메모리 셀(c_t)의 최종 출력값은 망각 게이트(forget gate)와 입력 게이트(input gate)를 통과하면서 총합(sum)을 통해 갱신

$$c_t = f_t \odot c_{t-1} + i_t \odot g_t$$

····· (54)

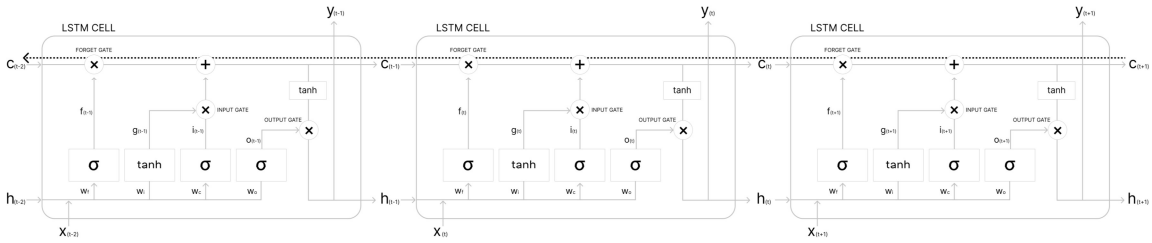
○ 요약하자면, 단기 메모리 셀(h_t)의 최종 출력값은 출력 게이트(output gate)에서 결정

$$h_t = o_t \odot \tanh(\tilde{c}_{t-1})$$

····· (55)

(3) LSTM 역전파

<그림 49> LSTM 역전파 학습



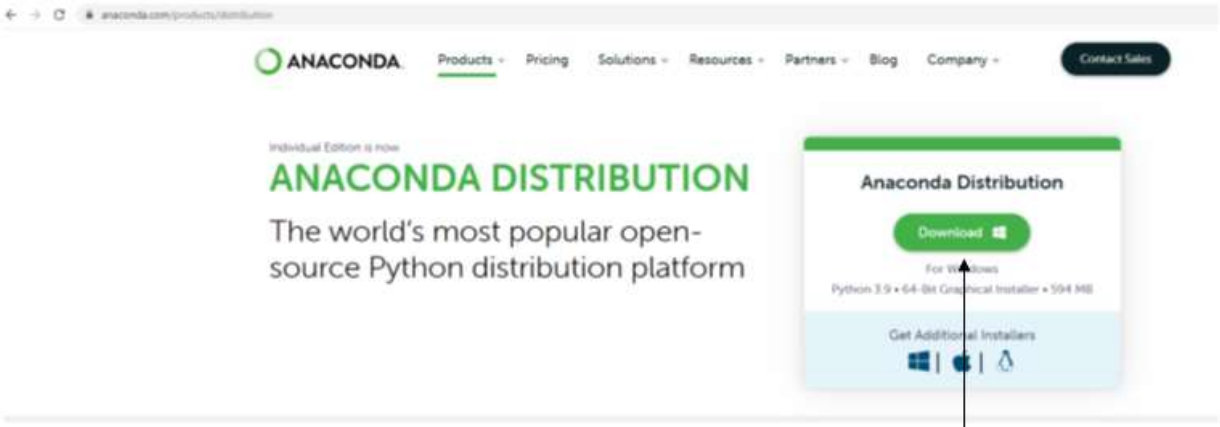
❑ 최종 오차가 메모리 셀 단위로 모든 노드로 진행되는 역전파 학습이며, 입력 게이트에서의 덧셈 연산은 그래디언트 소실(gradient vanishing) 문제를 발생시키지 않고, 망각 게이트에서 곱셈 연산은 그래디언트 소실 발생 원인인 행렬 곱 연산과 다른 점에서 RNN에 비해 안정적으로 신경망내 모수를 추정할 수 있는 장점이 있음

[부록 4: 모형 사용 매뉴얼 및 함수 설명]

<그림 50> Anaconda 다운로드 및 설치

1. 파이썬(Python) 프로그램 설치

- https://www.anaconda.com/products/distribution



- 아나콘다 설치 파일 다운로드 하여 설치

다운로드 버튼

- Python을 이용한 예측모형의 효과적인 모델링을 위해 객체지향프로그래밍(Object Oriented Programming)에 대한 개념[44]을 숙지할 필요가 있음. 관련 내용은 아래 도서 참고.

  Weisfeld, Matt. (2019). *The Object-Oriented Thought Process* (5th Edition), Addison-Wesley Professional

- 본 연구에서 제시하고 있는 모형의 모듈 내 함수들은 파이썬 언어에 대한 기본적인 지식만 있다면 이해할 수 있는 수준으로 작성되었기 때문에, 사용자의 요구에 맞게 쉽게 변형 및 가공 할 수 있을 것으로 기대. 파이썬 언어 기초 문법에 대한 입문서 중 하나는 아래와 같음

  Matthes, Eric. (2019). *Python Crash Course: A Hands-On, Project-Based Introduction to Programming* (2nd Edition), No Starch Press

44) 이는 파이썬 언어에만 해당하는 것이 아니라, Java, C, R 등 다양한 computing language를 사용한 모델링에도 적용됨.

<그림 51> 주요 라이브러리 설치 1

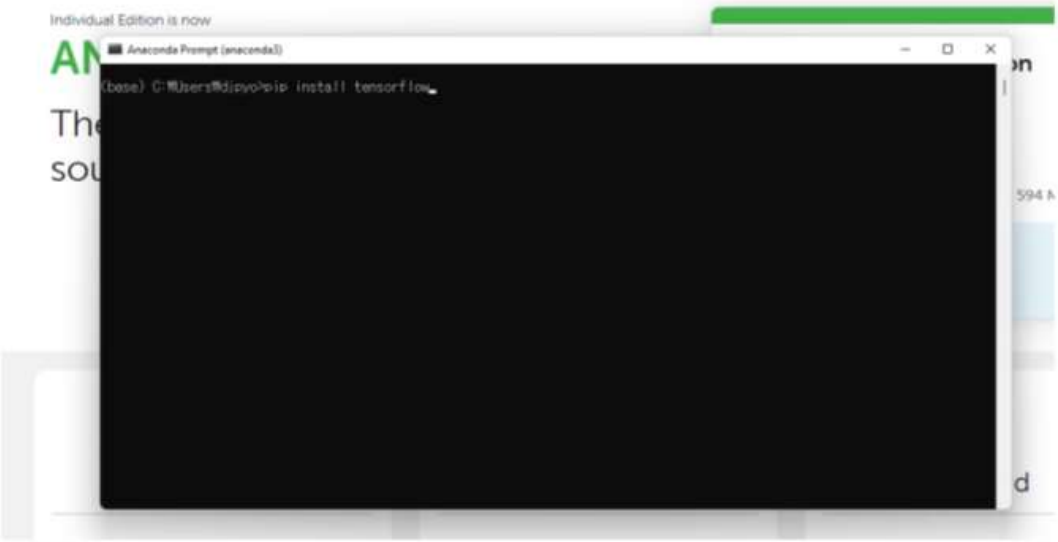
2. 주요 라이브러리 설치

- 아래와 같이 윈도우 검색에서 Anaconda Prompt 검색하여 실행

<그림 52> 주요 라이브러리 설치 2

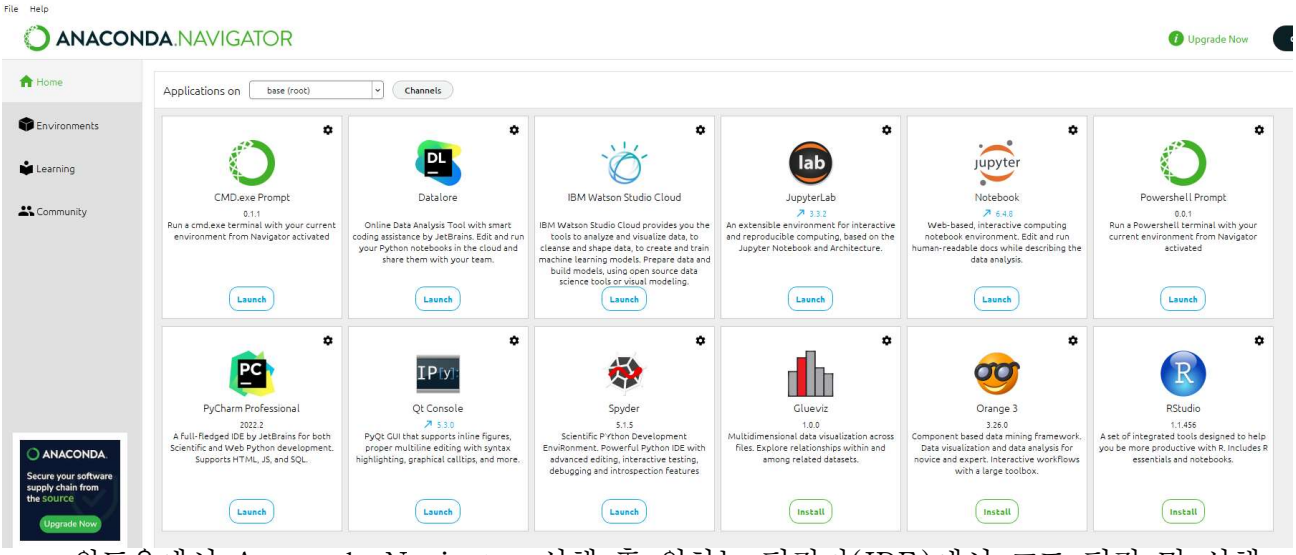
2. 주요 라이브러리 설치 (계속)

- 아래 Anaconda Prompt 명령창에서 아래 명령어 차례로 실행



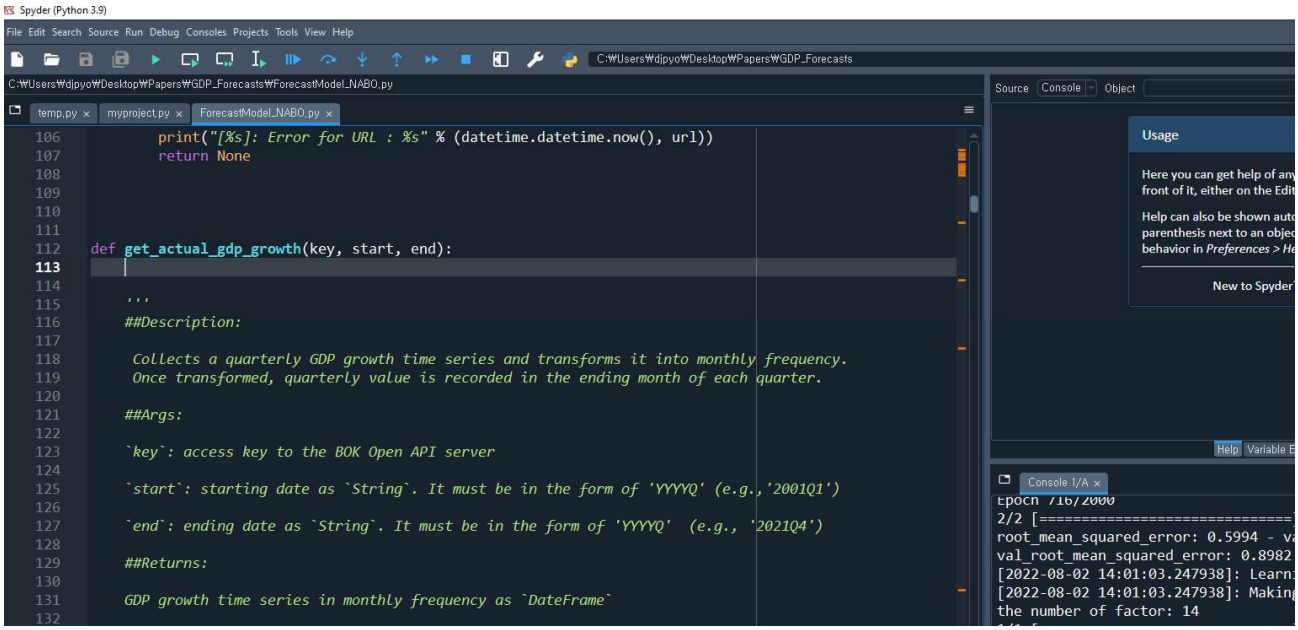

```
pip install tensorflow
pip install keras_tuner
pip install xlwings
```

<그림 53> 편집기 (IDE) 선택

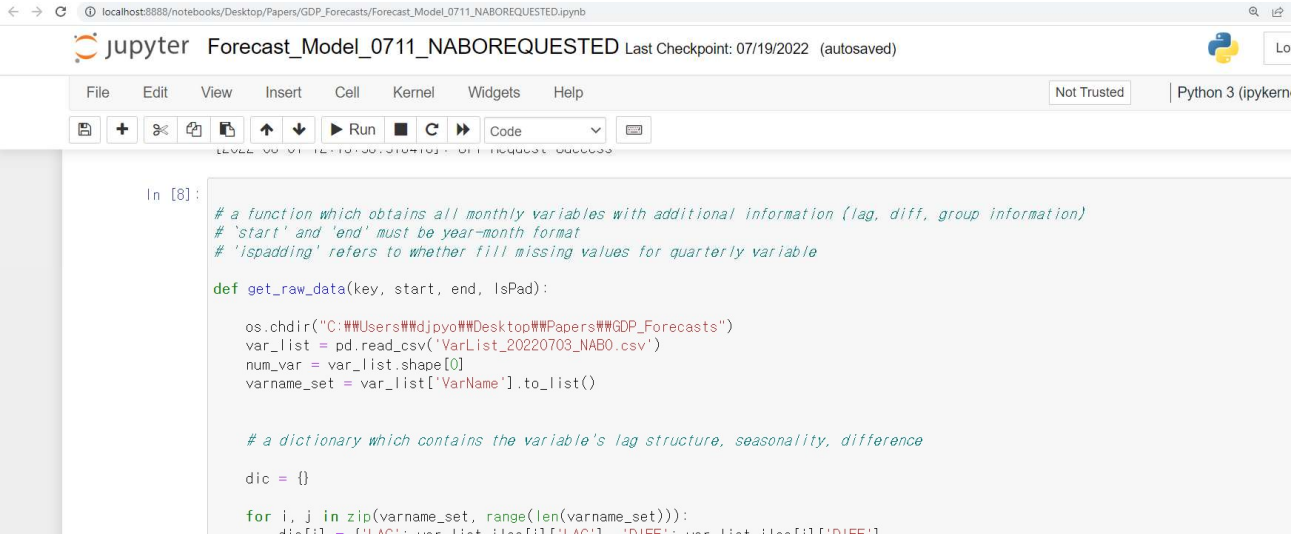


– 윈도우에서 Anaconda Navigator 실행 후 원하는 편집기(IDE)에서 코드 편집 및 실행

<그림 54> 편집기(IDE) 예시: Spyder



<그림 55> 편집기(IDE) 예시: Jupyter Notebook



<그림 56> 편집기에서 함수 실행

- cell에서 원하는 함수 또는 명령어 실행

Module **`ForecastModel_NABO`**

Created on Mon Aug 1 10:03:47 2022

This Python module contains essential functions for the short-term GDP growth forecasting of Korea. For the seasonal adjustment of variables, the module uses the U.S. Census Bureau's `X-13ARIMA-SEATS` library. So users are required to obtain the library to run `transform()` function in the module.

@author: Dong-Jin Pyo & Deung-Yong Heo

Functions

`def build_gru_model(hp)`

Description:

Generates a generic GRU model for hyperparmeters tuning.

Args:

`hp` : hyperparameter space, which is a `HyperParameter` object

Returns:

a generic GRU model, which is a `tensorflow.keras.Sequential` object.

`def build_lstm_model(hp)`

Description:

Generates a generic LSTM model for hyperparmeters tuning.

Args:

`hp` : hyperparameter space, which is a `HyperParameter` obejct

Returns:

a generic LSTM model, which is a `tensorflow.keras.Sequential` object.

`def create_timestep_data(data, timestep)`

Description:

Generates a 3-dimensional dataset. This is an auxiliary function for `get_new_instance()` .

Args:

`data` : dataset to consider

`timestep` : timestep

Returns:

`NumPy` object of feature variables with timestep.

Functions that use this function in the module:

`get_new_instance()`

`def df_model(data, dic, num_factor, path)`

Description:

Constructs a dynamic factor model and fits the model to the given data. This function is based on Python's `statmmodels.statespace.dynamic_factor_mq` library. For the details of this library, see the following:

https://www.statsmodels.org/devel/generated/statsmodels.tsa.statespace.dynamic_factor_mq.DynamicFactorMQ.html (https://www.statsmodels.org/devel/generated/statsmodels.tsa.statespace.dynamic_factor_mq.DynamicFactorMQ.html)

Args:

`data` : dataset upon which the dynamic factor model is built

`dic` : `Dictionary` object which contains variables' information

`num_factor` : the number of common factors in the model

`path` : directory path of where the variable list file is located, as `String` . For example, `C:\Users\djpyo\Desktop\Papers\GDP_Forecasts`

Returns:

`DynamicFactorMQ` object for the given dataset.

Functions which use this function in the module:

`fill_missing()`

`def fill_missing(data, dic, path)`

Description:

Fills missing values in the dataset with the DFM's predictions.

Args:

`data` : dataset to be filled

`dic` : `Dictionary` object which contains variables' information on factor groups

`path` : directory path of where the variable list file is located. For example, `C:\Users\djpyo\Desktop\Papers\GDP_Forecasts`

Returns:

filled dataset as `DataFrame` type.

`def get_actual_gdp_growth(key, start, end)`

Description:

Collects a quarterly GDP growth time series and transforms it into monthly frequency. Once transformed, all monthly entries in the series are filled with the growth rate of the corresponding quarter.

Args:

`key` : access key to the BOK Open API server

`start` : starting date, as `String` . It must be in the form of 'YYYYQN' (e.g.,'2001Q1')

`end` : ending date, as `String` . It must be in the form of 'YYYYQN' (e.g., '2021Q4')

Returns:

GDP growth time series in monthly frequency, as `DateFrame` .

`def get_factor_num(data)`

Description:

Calculates the optimal number of common factors among variables via PCA (Principal Component Analysis). Kaiser (1960) criterion is used to determine the appropriate number of common factors.

Args:

`data` : dataset to be analyzed, as `DataFrame`

Returns:

The number of common factors, as `Int` type.

Functions that use this function in the module:

`df_model()`

`def get_individual_var(key, start, end, stat_code, item_code1, item_code2, freq)`

Description:

Collects an individual feature variable. For the details of stat codes and item codes, refer to the BOK's API documentation:

https://ecos.bok.or.kr/api/#/DevGuide/StatisticalCodeSearch (https://ecos.bok.or.kr/api/#/DevGuide/StatisticalCodeSearch)

Note that the Open API service is not available for the nowcasting-related variables used in Appendix 2. Thus, you need to manually download the data from the following link and merge them with an existing dataset:

https://data.kostat.go.kr/nowcast/main.do (https://data.kostat.go.kr/nowcast/main.do)

Args:

`key` : access key to the BOK Open API Server

`start` : starting date, as `String` . It must be in the form of 'YYYYMM' (e.g.,'200101')

`end` : ending date, as `String` . It must be in the form of 'YYYYMM' (e.g.,'202001')

`stat_code` : stat code, as `String`

`item_code1` : item code 1, as `String`

`item_code2` : item code 2, as `String`

`freq` : data frequency (e.g., 'Q', 'M')

Returns:

individual variable time series as `DataFrame` type.

Functions that use this function in the module:

`get_raw_data()`

`def get_new_instance(start_m, end_m, timestep, dic, step, path, s_path)`

Description:

Generates a new instance of feature variables to make a prediction. The unseen future values of feature variables are generated by the pre-specified DFM model.

Args:

`start_m` : starting date of the training dataset. It must be in the form of 'YYYYMM' (e.g.,'200101')

`end_m` : current date at the time of forecasting. It must be in the form of 'YYYYMM' (e.g.,'200101')

`timestep` : timestep

`dic` : `Dictionary` object which contains variables' related information

`step` : forecasting horizon as `Int` type (this must be a nonnegative integer)

- step = n: n-quarter ahead forecasting horizon

`path` : directory path of where the variable list file is located. For example, `C:\Users\djpyo\Desktop\Papers\GDP_Forecasts`

`s_path` : directory path of where `X-13ARIMA-SEATS` library is located. For example, `C:\Users\djpyo\Desktop\Papers\GDP_Forecasts\x13as`

Returns:

`NumPy` object of feature variables with the timestep structure.

`def get_optimal_hyperparam(start_m, end_m, timestep, tuner, path, s_path)`

Description:

Conducts grid search for hyperparameters of neural networks. This function is based on `tensorflow`'s `keras_tuner` class. For the details of `keras_tuner`, refer to the following:

https://keras.io/keras_tuner/ (https://keras.io/keras_tuner/)

Once you find the optimal hyperparameters, you can build the model with the given tuner and the tuned hyperparameters by running, for example,

```
model = tuner.hypermodel.build(best_hps)
```

Args:

`start_m` : start date of a vintage dataset. It must be in the form of 'YYYYMM' (e.g.,'200101')

`end_m` : end date of a vintage dataset. It must be in the form of 'YYYYMM' (e.g.,'201201')

`timestep` : timestep

`tuner` : `keras_tuner` object for a specific model

`path` : directory path of where the variable list file is located. For example, `C:\Users\djpyo\Desktop\Papers\GDP_Forecasts`

`s_path` : directory path of where `X-13ARIMA-SEATS` library is located. For example, `C:\Users\djpyo\Desktop\Papers\GDP_Forecasts\x13as`

Returns:

`HyperParameter` object which contains the best hyperparameters for the given model and the data.

`def get_raw_data(key, start, end, IsPad, path)`

Description:

Collects a raw dataset that contains the requested variables in monthly frequency. Running this function automatically generates 'raw_data.csv' file in your currently working directory.

Args:

`key` : access key to the BOK Open API

`start` : starting date as `String` . It must be in the form of 'YYYYMM' (e.g.,'200101')

`end` : ending date as `String` . It must be in the form of 'YYYYMM' (e.g.,'201201')

`IsPad` : Refers to whether you want to fill missing values of the quarterly variable for monthly entries: (1) 1: to fill missing values with quarterly values for monthly entries. (2) 0: not to fill missing values with quarterly values for monthly entries.

`path` : directory path where the variable list file is stored, as `String` . For example, `C:\Users\djpyo\Desktop\Papers\GDP_Forecasts`

Returns:

A raw dataset which contains requested variables as `DataFrame` type.

`def get_request_url(url)`

Description:

Receives a response object from the BOK's server when given a proper url.

Args:

`url` : URL address to a specific Open API server, as `String` .

Returns:

JSON object that contains requested information.

Functions which use this function in the module:

`get_actual_gdp_growth()` , `get_individual_var()`

`def get_training_data(data, target, timestep)`

Description:

Generate a training dataset.

Args:

`data` : feature data

`target` : a target variable

`timestep` : timestep

Returns:

- training data of feature variables: 3-dimensional `NumPy` array (dimension = (sample_number, timestep, feature_number))
- training data of a target variable

`def get_vintage_data(start_vintage, end_vintage, path)`

Description:

Collects a vintage dataset and other additional information of variables (publication lag, whether to difference series, factor group information). To run this function, you need to have a 'raw_data.csv' file in your local drive.

Args:

`start_vintage` : starting date of a vintage dataset as `String` . It must be in the form of 'YYYYMM' (e.g.,'200101')

`end_vintage` : ending date of a vintage dataset as `String` . It must be in the form of 'YYYYMM' (e.g.,'200101')

`path` : directory path where both the variable list file and the raw data file are stored, as `String` . For example, `C:\Users\djpyo\Desktop\Papers\GDP_Forecasts`

Returns:

- vintage dataset for the periods you've specified as `DataFrame` type
- `Dictionary` object which contains the additional information (i.e., lag, difference, seasonality, factor group) of variables

Functions that use this function in the module:

`transform()` , `get_new_instance()`

`def make_prediction(start_m, end_m, timestep, step, path, s_path)`

Description:

Makes predictions for step-ahead GDP growth

Args:

`start_m` : start date of a vintage dataset. It must be in the form of 'YYYYMM' (e.g.,'200101')

`end_m` : end date of a vintage dataset. It must be in the form of 'YYYYMM' (e.g.,'201201')

`timestep` : timestep

`step` : forecasting horizon (this must be a nonnegative integer)

- 0: for the current quarter
- 1: 1-quarter ahead

- and so on....

`path` : directory path of where the variable list file is located. For example, `C:\Users\djpyo\Desktop\Papers\GDP_Forecasts`

`s_path` : directory path of where `X-13ARIMA-SEATS` library is located. For example, `C:\Users\djpyo\Desktop\Papers\GDP_Forecasts\x13as`

Returns:

- LSTM model's prediction
- GRU model's prediction

`def train_gru_network(X, y, timestep)`

Description:

Constructs a GRU network model and fits it to the given data.
In the training process, L2 regularization for kernel and recurrent is applied. For the details of the GRU layer implementation, see the following:

https://www.tensorflow.org/api_docs/python/tf/keras/layers/GRU
(https://www.tensorflow.org/api_docs/python/tf/keras/layers/GRU)

Note that, in neural network training, batch size refers to the number of training samples before the model's internal parameters are updated, while the number of epochs refers to the number of complete passes through the training dataset.

Args:

`X` : training data of feature variables

`y` : training data of a target variable

`timestep` : timestep

Returns:

a pre-trained GRU model, which is a `tensorflow.keras.Sequential` object.

`def train_network(X, y, timestep)`

Description:

Constructs a LSTM network model and fits it to the given data.
In the training process, L2 regularization for kernel and recurrent is applied. For the details of the LSTM layer implementation, see the following:

https://www.tensorflow.org/api_docs/python/tf/keras/layers/LSTM
(https://www.tensorflow.org/api_docs/python/tf/keras/layers/LSTM)

Note that, in neural network training, batch size refers to the number of training samples before the model's internal parameters are updated, while the number of epochs refers to the number of complete passes through the training dataset.

Args:

`X` : training data of feature variables

`y` : training data of a target variable

`timestep` : timestep

Returns:

a pre-trained LSTM model, which is a `tensorflow.keras.Sequential` object.

`def transform(vintage_data, dic, path, s_path)`

Description:

Transforms a raw dataset into stationary and seasonally-adjusted series. For the seasonal adjustment, this function uses the U.S. Census Bureau's `X-13ARIMA-SEATS Seasonal Adjustment Program` library. Users are required to install the library before running it.

Args:

`vintage_data` : `DateFrame` object which contains variables

`dic` : `Dictionary` object which contains the variables' lag structure, seasonality, stationary and group information

`path` : directory path of where the variable list file is located. For example, `C:\Users\djpyo\Desktop\Papers\GDP_Forecasts`

`s_path` : directory path of where `X-13ARIMA-SEATS` library is located. For example, `C:\Users\djpyo\Desktop\Papers\GDP_Forecasts\x13as`

Returns:

`DataFrame` object of the transformed dataset.

Index

Functions

build_gru_model
build_lstm_model
create_timestep_data
df_model
fill_missing
get_actual_gdp_growth
get_factor_num
get_individual_var
get_new_instance
get_optimal_hyperparam
get_raw_data
get_request_url
get_training_data
get_vintage_data
make_prediction
train_gru_network
train_network
transform