프로젝트 보고서

**프로젝트 명: 딥러닝을 활용한 창원시 주택가격지수 예측**

작성자 : 최진서

[1. 요약 2](#_Toc68595007)

[2 서론 3](#_Toc68595008)

[**2.1 추진배경 및 목적** 3](#_Toc68595009)

[3. 연구동향 5](#_Toc68595010)

[**3.1 개발연구 동향** 5](#_Toc68595011)

[**3.2 기존시스템과의 비교** 5](#_Toc68595012)

[4. 설계 5](#_Toc68595013)

[**4.1 개발환경** 5](#_Toc68595014)

[**4.2 시스템 설계** 6](#_Toc68595015)

[5. 구성 요소별 기능 구현 7](#_Toc68595016)

[**5.1 학습모델 선정** 7](#_Toc68595017)

[**5.3 데이터 전처리** 10](#_Toc68595018)

[**7.1참고문헌 출처** 13](#_Toc68595019)

# **1. 요약**

부동산은 국내에서 규제와 세금이 가장 강력한 시장이다. 부동산 시장의 가격 등락에 따라서 국가경제에 큰 영향을 끼치는데 부동산 가격이 폭등하게 되면 사회 초년생 같은 보유자산 규모가 작은 집단의 내집마련, 영세 사업자의 사업장 확보가 어려워지면서 등 서민경제가 악화될 위험이 있고 폭락하게 되면 부동산 시장이 위축되면서 건설업체들과 부동산으로 자산을 보유하고 있는 개인이 피해를 입고 가격 폭락 전 부동산을 담보로 금융기관으로 대출을 받은 개인 또는 기업들이 부동산을 팔아서 채무를 이행할 능력이 떨어지게 되며 수많은 사람들이 빚더미에 올라 국가적인 경기침체로 이어질 수 있다. 정부에서는 부동산 시장의 중요성을 인지하고 여러가지 규제나 정책 시행을 통해 부동산 가격을 조절하는데 이는 정부 경제능력을 평가하는 중요한 잣대가 되기도 한다. 적절한 정책 또는 규제시행을 위해선 여러가지 경제지표의 통계분석을 통한 부동산 가격 등락을 예측하는 과정이 필요하다.

본 문서에서는 딥러닝을 사용하여 여러가지 경제지표 데이터 학습을 통해 창원시 주택가격지수 예측하는 모델을 설계하는 과정과 정확성 평가를 진행한 결과를 제공한다.

주택가격지수 예측은 시계열 데이터 학습에 적합하다고 알려진 LSTM 모델을 사용하고 5가지 경제지표 시계열 데이터를 학습하여 입력 데이터의 마지막 시점으로부터 다음 6개월의 주택가격지수를 예측한다. 2008-01-01~2020-12-01 사이 기간의 한국은행, 한국 부동산원에서 제공하는 시계열 경제지표 데이터셋을 사용하였으며 2018~2020년 사이 데이터를 평가데이터로 분리하여 예측모델의 성능평가를 진행했을 때 validation\_loss가 0.23의 성능을 보인다.

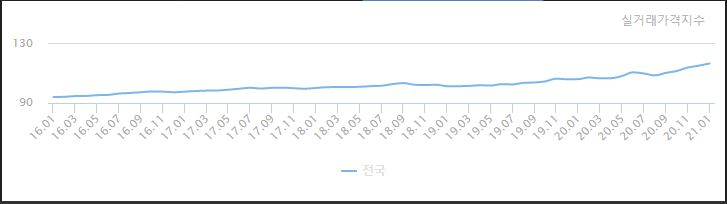
기존에 이미 존재하는 시계열 예측모델인 VAR, ARIMA 모형은 데이터셋의 계절성, 추세 등의 데이터 특성을 제거하여 편향을 막는 전처리 과정이 필요하지만, LSTM 모델은 전처리 과정이 없이도 비슷하거나 더 좋은 성능을 나타낸다는 장점이 있다.

창원시 주택가격지수 예측모델을 사용해 지자체에서는 선제적인 정책, 규제시행을 통해 부동산 시장 가격을 안정적으로 유지할 수 있을 것으로 기대된다.

# **2 서론**

## **2.1 추진배경 및 목적**

최근 5년동안 전국 주택가격은 꾸준히 상승세를 보이고 있다. 주택가격의 급등은 주택 실수요자, 특히 신혼부부 같은 젊은 세대의 주거불안정과 소비심리 위축을 야기하면서 출산율 감소, 빈부격차 확대와 같은 여러 사회문제의 원인으로 작용하고 있다.



[그림 2.1 전국 주택 실거래가격지수]

인구감소 국면으로 접어든 우리나라에서 특히, 지방에서의 주거불안정 문제는 젊은 세대 인구유출의 원인이 되는 민감한 사안이다. 주택가격의 급등의 원인으로는 수요-공급의 불균형, 특정지역 투기과열 등 여러가지가 존재하는데, 이에 정부에서는 주택 공급량 증가, 투기과열지구 지정 등 여러가지 규제와 정책을 시행하여 상승률 감소에 나서고 있다. 최근 경남에서는 창원시 의창구가 투기과열지구로 지정되면서[[[1]](#footnote-2)] 주택법에 따른 분양권 전매제한, 주택청약 1순위 자격제한, 주택담보대출 조건강화 등의 규제가 적용되었다.

지도이(가) 표시된 사진

자동 생성된 설명

[그림 2.1 투기과열지구 지정현황]

하지만 과도한 규제는 주택가격의 폭락으로 이어질 수 있는데 주택가격이 폭락하게 되면 부동산 시장이 위축되면서 건설업체들과 부동산으로 자산을 보유하고 있는 중산층이 경제적인 피해를 입고 가격 폭락 전 부동산을 담보로 금융기관으로 대출을 받은 개인 또는 기업들이 부동산을 팔아서 채무를 이행할 능력이 떨어지게 되며 이는 국가적인 경기침체로 이어질 수 있다.

앞서 서술한 바와 같이 주택가격의 등락폭을 최소화하면서 적절한 가격을 유지하는 것이 국가경제를 유지하는데 있어서 정부 및 지자체에 요구되는 중요한 능력이자 현 정부의 역량을 평가하는 지표가 되기도 한다[]. 따라서 적절한 주택가격 유지를 위해서는 현재 경제상황을 파악하고 여러가지 경제지표를 분석하여 예상되는 주택가격지수를 산정하여 이에 맞는 대책을 적절한 규제나 정책을 시행해야 한다. 이때 주택가격지수 예측을 위해 딥러닝 모델을 사용하면 비교적 신속하고 정확한 예측으로 주택정책 전반에 대한 적절한 대책을 강구할 수 있으리라 사료된다.

**2.3 개발목표**

본 문서는 창원시 주택가격지수와 관련성이 있는 경제지표 데이터를 학습한 모델을 사용하여 딥러닝 관련 지식이 없는 사람들도 신뢰성 있는 주택가격지수를 예측 값을 제공하는 프로그램을 제작하는 것이 목표이다. 경제지표 데이터를 월별 시계열 데이터셋으로 확보하고 예측모델은 시계열데이터 학습에 적합하다고 알려진(링크) LSTM 모델을 사용한다. 창원시에 소속된 각 지역구 주택가격지수 예측에 최적화된 모델이 존재하고 사용자는 주택가격지수를 예측할 지역구를 선택할 수 있다, 반복구동을 통해 가장 높은 예측성능을 보이는 하이퍼파라미터를 선정하여 최적화된 모델을 사용하여 예측 프로그램을 제작한다. 최종적으로는 GUI환경에서 경제지표 데이터를 입력하면 예측치를 출력할 수 있도록 구현하고자 한다.

# **3. 연구동향**

## **3.1 개발연구 동향**

과거 경제지표 데이터를 활용하여 주택가격지수를 예측하려는 연구는 예전부터 많이 진행되어 왔다. (서승환, 1994; 박헌수, 2003; 정규일, 2006; 김세완, 김은미, 2009; 이준용·손재영, 2010; 김문성·배형, 2013; 전해정, 2017) 하지만 최근 딥 러닝을 활용한 주택가격지수 예측모형이 기존 예측모형보다 예측성능이 뛰어나다는 것이 확인되면서(민성욱, 2017; 배성완·유정석2017; 이창로·김세형2018) 딥 러닝을 활용한 예측연구가 활발하게 진행되고 있다. 전해정 ∙ 양혜선(2019)는 시계열 데이터 예측에 적합하다고 평가받는 RNN, LSTM, GRU 모형의 주택가격지수 예측결과를 비교분석하여 RNN, GRU 모델의 예측성능이 LSTM보다 다소 뛰어남을 확인하였으나 하이퍼파라미터 설정에 의해 평가 결과가 뒤바뀔 수 있다는 한계점을 시사했다. 이태형 ∙ 전명진(2018) 6가지 거시경제지표 데이터를 사용했고 모형 안정성 확보를 위해 예측모형의 반복 구동 후 예측 값의 평균과 표준편차를 통해 전통적인 벡터자기회귀모형VAR과 RNN, LSTM 모형의 주택가격지수 예측성능을 분석한 결과 VAR 모형의 RMSE값이 LSTM모다 2배가량 더 높아 전통적인 다변량 시계열 통계분석모형보다 딥러닝 모형의 성능이 더 우수함을 보였고 RNN에 비해 LSTM의 예측 값에 대한 표준편차 값이 작아 LSTM 모형이 재현성과 안정성 측면에서 더 나은 성능을 나타낸다는 것을 보였다.

## **3.2 기존시스템과의 비교**

현재 국내에서 관련 연구는 활발한 것에 반해 아직까지 주택가격지수 예측에 관한 서비스는 존재하지 않는 것으로 확인된다. 본 문서에서는 기 서술한 연구자료를 참고하여 입력데이터셋을 산정하고 모형을 제작한 뒤 최적화된 예측모형을 누구나 쉽게 사용할 수 있는 프로그램을 구현하고자 한다.

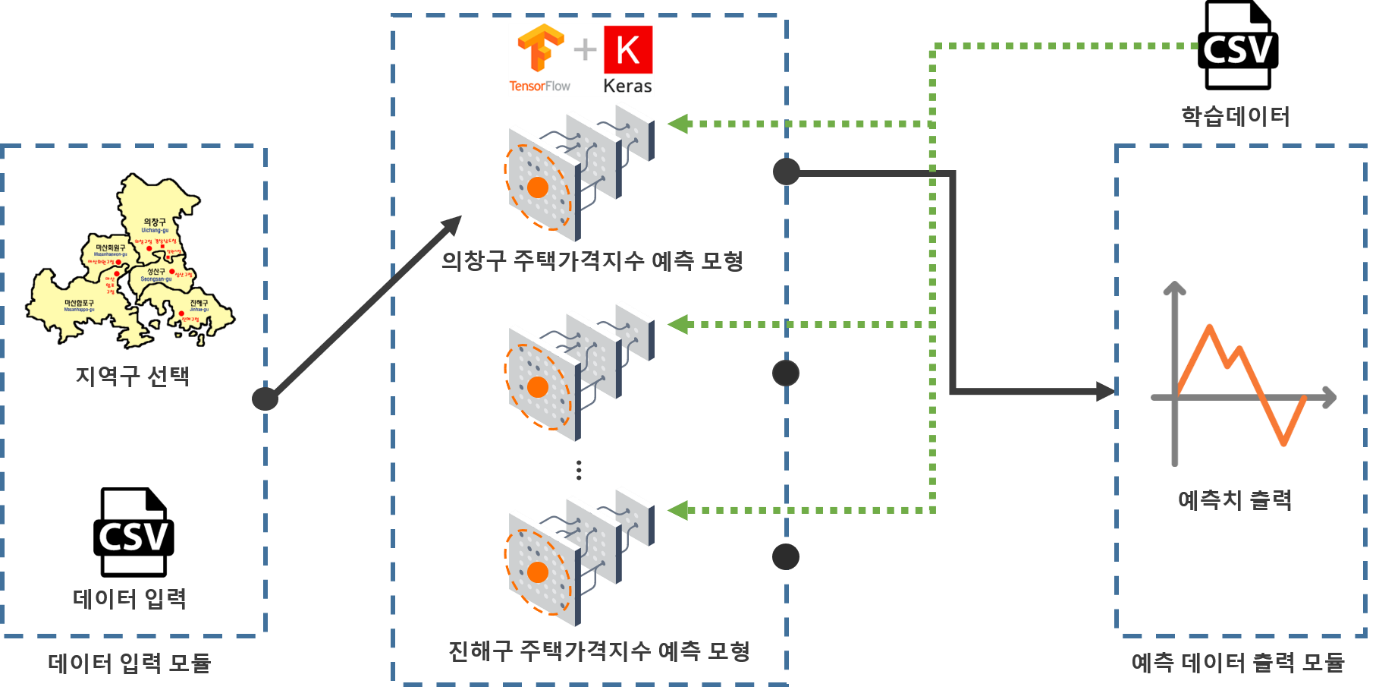
# **4. 설계**

## **4.1 개발환경**

|  |  |
| --- | --- |
| **H/W** | * Google Colabortory(GPU runtime) * PC   + CPU: 3rd Gen AMD Ryzen 3500x 6-core 3.6 GHz   + GPU: Nvidia GeForce RTX 2060 6GB   + Memory: DDR4 16GB   + Storage: Nvme m.2 512 GB |
| **S/W** | * OS: MicroSoft window 10 pro 64bit * IED: VisualStudio Code, jupyter notebook * Language: Python * Framework: TensorFlow 2.0, Keras, * Version Control: Git |

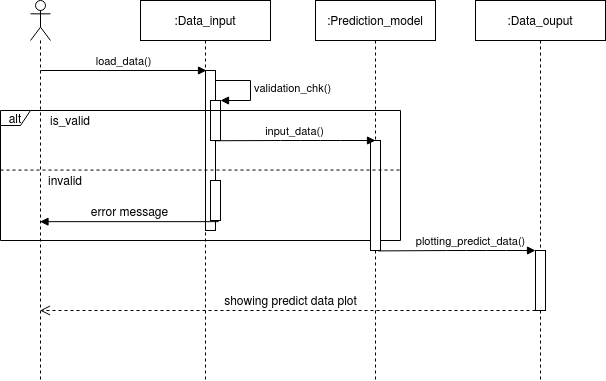
[표 4.1 개발환경]

## **4.2 시스템 설계**



[그림 4.2 시스템 설계]

시스템은 크게 데이터 입, 출력 모듈과 데이터 예측 모듈로 나뉜다. 사용자는 창원시에 소속되어 있는 지역구 들 중 하나를 선택할 수 있으며 선택한 지역구에 대한 경제지표 데이터를 파일 형식으로 로드 할 수 있다. 로드 된 데이터는 특정 포맷을 만족하는지 여부를 검사하며 만족하지 않을 시 재입력을 요구한다. 입력데이터는 선택된 지역구의 주택가격지수를 예측하는 모델로 전달되며 미리 학습된 예측모형에 의해 입력데이터 시점으로부터6개월간의 예측치를 출력한다. 출력 값은 출력 모듈로 전달되어 선형그래프 형태로 출력한다.



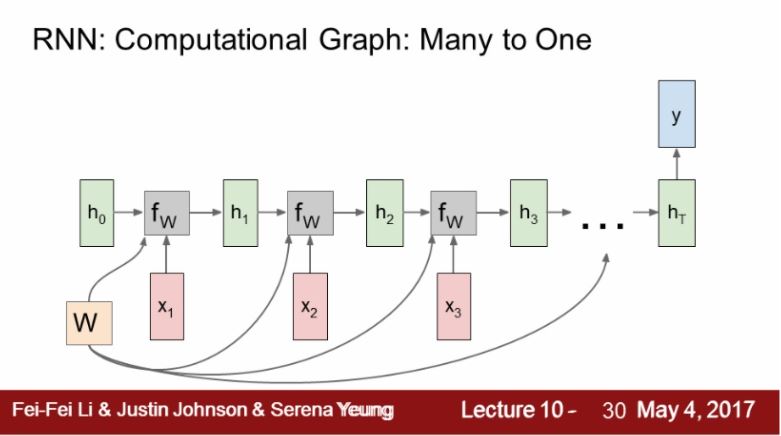
[그림 4.2 시스템 시퀀스 다이어그램]

[그림 4.2]는 시스템의 순차 다이어그램으로 각 모듈의 기능과 데이터의 흐름을 나타낸 것이다.

# **5. 구성 요소별 기능 구현**

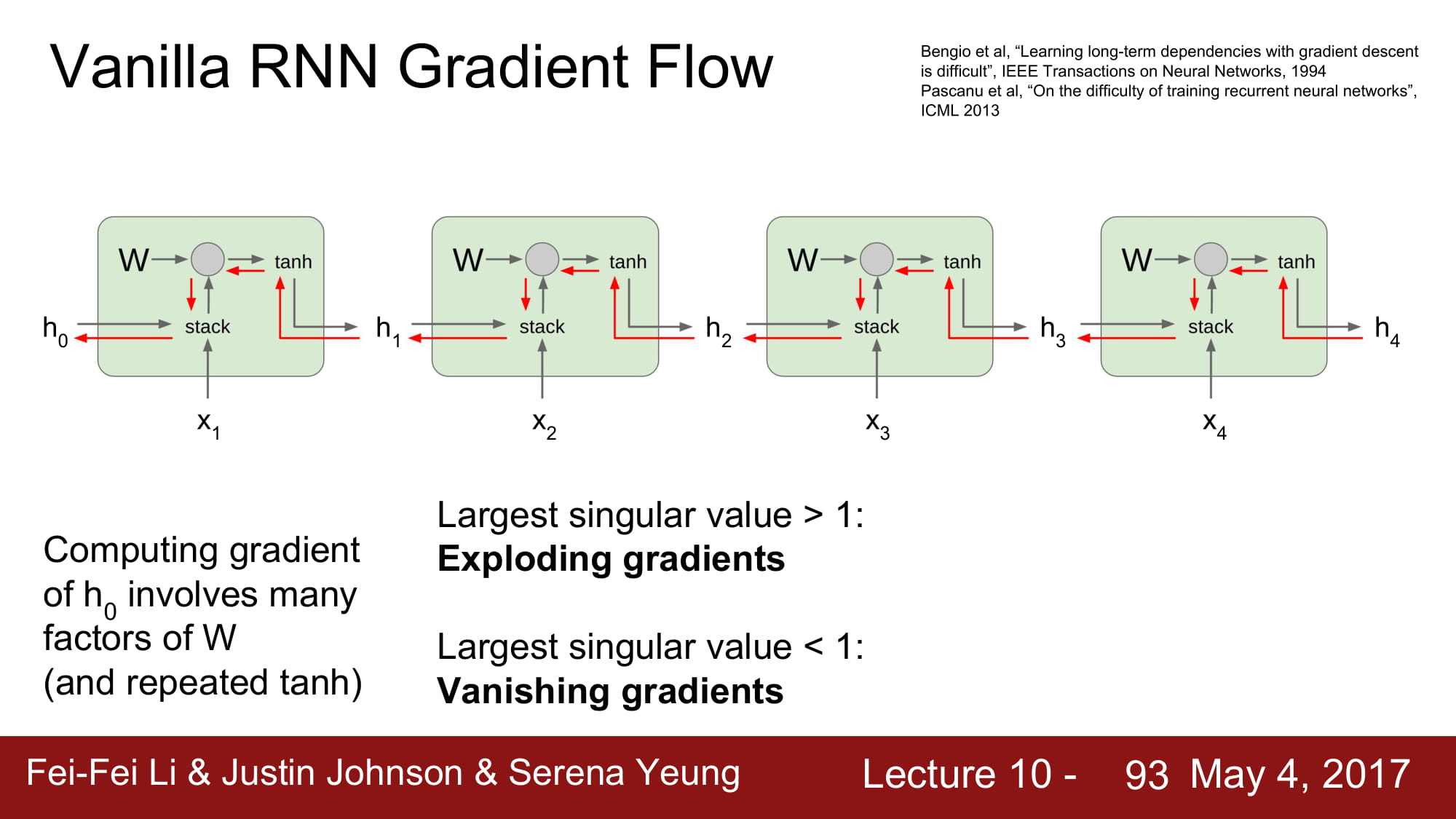
## **5.1 학습모델 선정**

본 문서에서는 주택가격예측을 위해 다변량 시계열 데이터를 학습하는 LSTM 모형을 사용한다. LSTM(Long Short-Term Memory)은 순환신경망(Recurrent Neural Network)의 한 종류이다. 순환신경망은 은닉층이 순환구조를 띄는 신경망으로 아래 그림은 여러 개의 데이터를 학습하여 하나의 값을 예측하는 RNN모델을 단순하게 표현한 것이다. 그림을 보면 각 레이어 마다 이전 레이어의 hidden state(hn-1) 값을 반영하여 현재 입력값과 가중치를 계산한 hidden state(hn)를 다음 레이어로 전달하는 것을 확인할 수 있다. 즉, 과거의 데이터가 미래에 영향을 끼치는 구조를 가진 모델이다. RNN은 음성인식, 번역, 감정분석, 자연어 처리 등에서 높은 성능을 보이며 또한 시계열데이터 학습에도 적합하다고 알려져 있다.



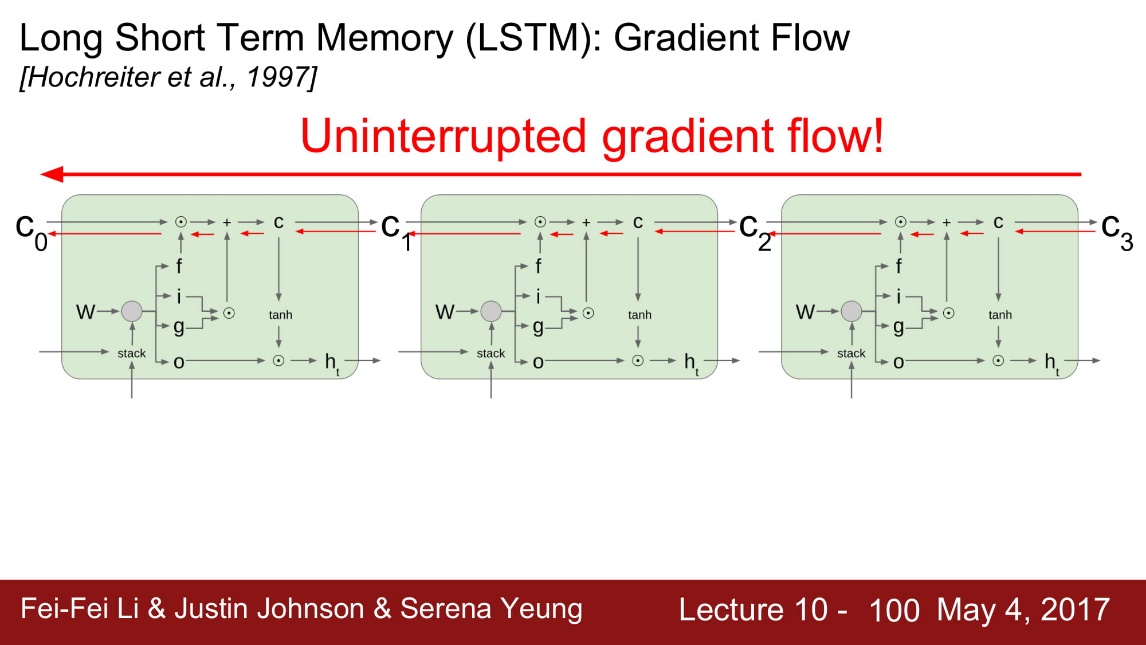
[그림 5.1 RNN Computational Graph]

하지만 장기적인 데이터 학습에 대해서 효율이 떨어지고 학습능력이 저하된다는 단점이 있는데 이는 학습 과정 중 모델의 예측값에 대한 실제값의 loss를 반대로 전달하면서 각 레이어의 파라미터를 업데이트하는 역전파(back propagation) 과정에서 발생한다. 아래 그림에서는 RNN 모델의 역전파 과 나타내고 있다. 이 과정에서 행렬 곱을 수행하게 되는데 이는 RNN 유닛이 많을수록, 수많은 연산 오버헤드가 발생하면서 효율이 떨어지고 수많은 연산에 의해 변질된 값은 gradient vanishing, 또는 exploding을 야기하여 파라미터를 제대로 업데이트 할 수 없어 장기적인 데이터에 대한 학습성능이 떨어지게 된다. (이태형·전명진, 2018).



[그림 5.1 RNN Gradient Flow]

이러한 문제를 해결하기 위해 LSTM모형이 등장했다. LSTM은 cell state(Cn)를 도입하여 역전파 시 gradient를 cell state를 통해 전달하여 불필요한 연산을 줄여 연산 오버헤드를 줄이고 안정적으로 gradient를 전달하여 장기 데이터 학습에도 높은 성능을 보인다. 아래 그림을 보면 역전파 과정이 기본RNN에 비해 상당히 단순하게 묘사된 것을 확인할 수 있다.



[그림 5.1 LSTM Gradient Flow]

본 문서에서는 RNN의 단점을 개선한 LSTM 모형을 사용하여 Many-to-one, 여러가지 시계열 경제지표 데이터를 학습하여 주택가격지수를 예측하는 학습모델을 구현한다.

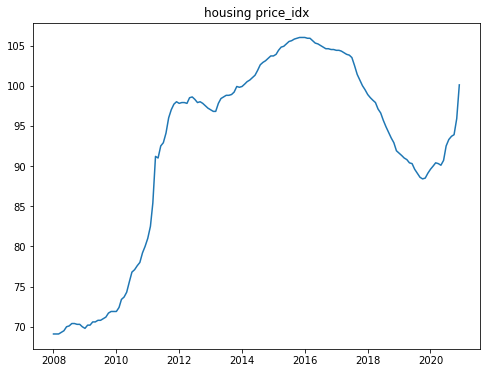
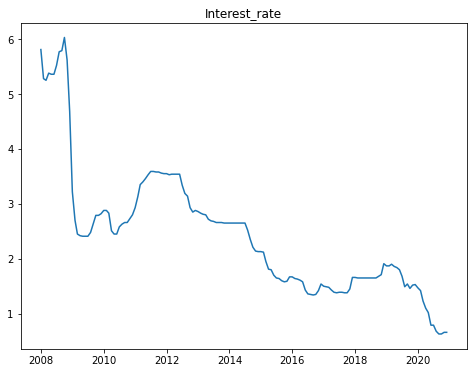
**5.2 데이터 선정**

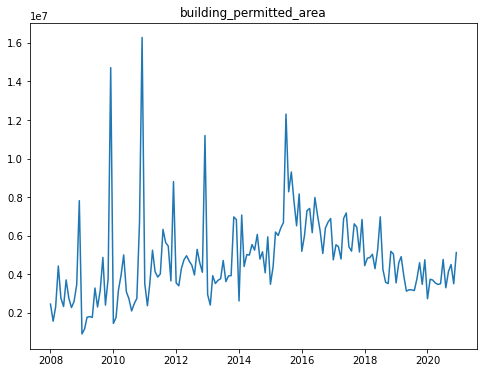
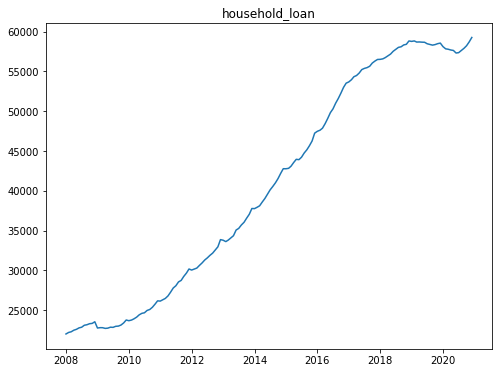
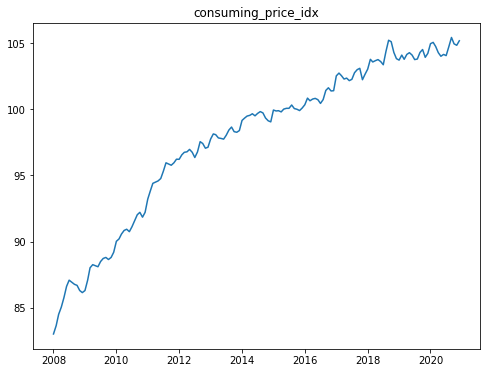
학습데이터셋은 기존 선행연구들을 참고하여 주택가격지수과 연관성이 있는 실증분석이 진행된 데이터셋 5가지를 선정하였다. (전해정 ∙ 양혜선, 2019), (이용만, 이상한, 2004) 목록은 아래와 같다

|  |  |  |  |
| --- | --- | --- | --- |
| **변수명** | **설명** | **단위** | **출처** |
| housing price\_idx() | 지역구별 주택가격지수 | 지수 | 한국 부동산원 |
| Interest\_rate() | 전국CD금리 | 연% | 한국은행 |
| household\_loan() | 경남 가계대출금 | 십억원 | 한국은행 |
| building\_permitted\_area  () | 창원시 건축허가면적 | m3 | 창원시 |
| consuming\_price\_idx() | 경남 소비자물가지수 | 지수 | 한국은행 |

[표 5.2 데이터셋 목록]

5가지의 경제지표 데이터를 독립변수로 지정했으며 그 중 주택가격지수는 독립변수이자 종속변수로 지정하여 각 지역구별 주택가격지수 특성을 반영할 수 있도록 하였다. 각 데이터셋은 2008년 1월 1일부터 2020년 12월1일까지 월별로 확보하였으며 각 데이터별 변화 추이는 아래와 같다.

[그림 5.2 데이터셋 추이 그래프]

## **5.3 데이터 전처리**

import os

import datetime

import IPython

import IPython.display

import matplotlib as mpl

import matplotlib.pyplot as plt

%matplotlib inline

import numpy as np

import pandas as pd

import seaborn as sns

import tensorflow as tf

from keras import backend as K

from google.colab import files

mpl.rcParams['figure.figsize'] = (8, 6)

mpl.rcParams['axes.grid'] = False

데이터셋 전 처리 및 시각화를 위해 IPython, matplotlib, pandas, seaborn 라이브러리를 사용한다.

uploaded = files.upload()

for fn in uploaded.keys():

  print('User uploaded file "{name}" with length {length} bytes'.format(name=fn, length=len(uploaded[fn])))

data = pd.read\_csv('/content/integrated\_data.csv')

data['new\_Date'] = pd.to\_datetime(data['date'])

print(data.info())

print('\n')

#첫번째 컬럼을 날짜형식 컬럼으로 변경

print(type(data['new\_Date'][0]))

data.drop('date', axis = 1, inplace=True)

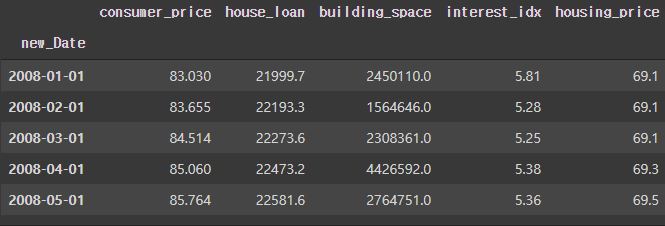
data.set\_index('new\_Date', inplace=True)

print(data.head())

print('\n')

print(data.info())

Pandas 라이브러리를 사용하여 데이터셋 파일을 로드하여 데이터셋 객체를 생성한다. 객체를 통해 날짜 컬럼을 조회하여pandas에서 인식할 수 있는 날짜형식의 컬럼으로 변경한다. 변경된 데이터셋의 형식은 아래와 같다.



[그림 5.3 데이터셋 출력값]

2008년1월 1일부터2018년 12월1일사이의 120행 데이터를 학습데이터로 분리하고 나머지 36행 데이터를 모델의 성능평가를 위한 평가데이터로 사용한다. 이때 모델 재현성 유지를 위해 랜덤시드를 설정한다. 또한 데이터셋의 각 컬럼의 단위가 서로 다르기 때문에 데이터셋을 표준화한다.

#훈련데이터 분할

#2008-01-01 ~ 2018-01-01 120개월을훈련데이터로 분할

# 각 데이터 원시값-평균/표준편차 => 데이터셋 표준화

TRAIN\_SPLIT =120

tf.random.set\_seed(13)

dataset = features.values

data\_mean = dataset[:TRAIN\_SPLIT].mean(axis=0)#각 열에 대한120인덱스 까지의  평균값

data\_std = dataset[:TRAIN\_SPLIT].std(axis=0)# 각 열에 대한 표준편차차

print(data\_std)

dataset = (dataset-data\_mean)/data\_std

print(dataset)

LSTM모델 훈련에 사

#훈련에 사용한 특정시간 윈도우 데이터 확보를 위한 메소드

#history: 과거데이터 크기

#target size 모델이 미래를 예측할 크기

# step값이 주어지면 step에 맞게 데이터를 샘플링

*def* multivariate\_data(*dataset*, *target*, *start\_index*, *end\_index*, *history\_size*, *target\_size*, *step*, *single\_step*=False):

    data = []

    labels = []

    start\_index = start\_index + history\_size

    if end\_index is None:

        print("dataset len %d",len(dataset))

        end\_index = len(dataset) - target\_size

    for i in range(start\_index, end\_index+1):

        indices = range(i - history\_size, i, step)

        print(indices)

        data.append(dataset[indices])#

        if single\_step:

          labels.append(target[i + target\_size])

          print(label)

          print('타겟 데이터')

          print(target[i + target\_size])

        else:

          labels.append(target[i:i + target\_size])

    return np.array(data), np.array(labels)

## **7.1참고문헌 출처**

이태형, 전명진. (2018). 딥러닝 모형을 활용한 서울 주택가격지수 예측에 관한 연구. 주택도시연구, 8(2), 39-56.

전해정, 양혜선. (2019). 딥 러닝을 이용한 주택가격 예측에 관한 연구. 주거환경, 17(2), 37-49.

민성욱 (2017) 딥러닝(Deep Learning)을 이용한 주택가격 예측모형 연구, 강남대학교 대학원 박사학위

논문

Lim, S. S. (2014). A study on the forecasting models using housing price index. Journal of the Korean Data and Information Science Society, 25(1), 65–76. <https://doi.org/10.7465/jkdi.2014.25.1.65>

정원구ㆍ이상엽(2007), “인공신경망을 이용한 공동주택 가격지수 예측에 관한 연구 –서울지역을 중심 으로-”, 주택연구, 15(3), pp.39~64.

한국부동산원 http://www.kab.co.kr/ 24.

한국은행 경제통계시스템 <http://ecos.bok.or.kr/>

Fei-Fei Li, Justin Johnson, Lecture 10: Recurrent Neural Networks, http://cs231n.stanford.edu/slides/2017/

1. 국가법령정보센터, 투기과열지구, 국토교통부공고 제2020-1649호 지정, https://www.law.go.kr [↑](#footnote-ref-2)