8조 최현율 김명소 황정임 조 억

언론기사를 활용한 부동산 가격 흐름 분석 및 예측

최종발표

2018.06.14

고려대학교
KOREA UNIVERSITY

Contents

1 Introduction

2 Data

3 Article Selection

4 Prediction Model

1. Text CNN [비정형 데이터]
2. Bidirectional LSTM [비정형 데이터]
3. 정형 데이터 모델 (월별, 전국)

5 Conclusions

6 Appendix

Introduction

- Article Selection ：부동산 기사를 토픽별로 분류하고 성격을 파악하여 예측에 도움이 되는 기사 선별
- Prediction : 아파트 전세가격지수 예측

Data Crawling

- Making crawlers
- Collecting news articles

Pre-processing

- Analyzing Morpheme
- Removing stop words
- Analyzing term-document frequency

Article Selection

- Classifying Documents
  - LDA (Latent Dirichlet Allocation)
- Analyzing Topics

Prediction

- Fitting Prediction Models
  - Text CNN
  - Bidirectional LSTM
  - Temporal CNN (정형 데이터)

Dataset

KB 부동산 (http://nland.kbstar.com)

- KB부동산 -> KB종합뉴스
- 기간 : 2007.04 ~ 2018.04

번호	분류	제목	출처	등록일	조회수
418	시장분석	(재공지)4월 청담동 삼성청담(래미안) 116㎡ 2억5천만...	부동산정보팀	2018.04.30	55
417	시장분석	4월 청담동 삼성청담(래미안) 116㎡ 2억5천만원 증가 (...	부동산정보팀	2018.04.30	48
416	시장분석	서대문구 4월 주택가격 2.04% 전국 최고 상승(지역 주...	부동산정보팀	2018.04.30	73
415	시세동향	이번에 이사가볼까 전셋값 안정	조인스랜드	2018.04.30	43
414	시세동향	아파트값 상승률 '제자리 걸음마' 수준	조인스랜드	2018.04.30	43
413	기타	잠실엘스 84.8㎡ 보유세 45% 올라 327만원..."종부세...	조선닷컴	2018.04.30	90
412	시세동향	강남4구 집값 8개월 만에 첫 '동반 하락'	조인스랜드	2018.04.27	122
411	기타	남북 화해 무드에 빛 보는 파주 땅 ... '자경' 원칙 잊지...	조인스랜드	2018.04.27	108
410	분양/청약	청약개편-선거-월드컵 3중변수... 건설사들 줄줄이 분...	동아일보	2018.04.27	319
409	시세동향	"정부규제 약발" 서울 집값 하향 안정세...11주 연속 둔화	동아일보	2018.04.27	41

Article Selection Outline

Process

Topic Allocation	Topic Analysis	Article Selection
• LDA를 통해 각 기사들에 토픽 할당 및 토픽 별 주요단어 추출	• 토픽의 주요 단어들을 분석하여 선/후행 토픽을 분류	• 선행 토픽의 기사들은 train set에 포함시키고 후행 토픽의 기사들은 제거

- 명사, 동사, 형용사로 corpus 생성 후 토픽 개수 (k)를 10으로 하여 기사마다 하나의 토픽을 할당

Topic Analysis

1 LDA - Topic 3 Analysis

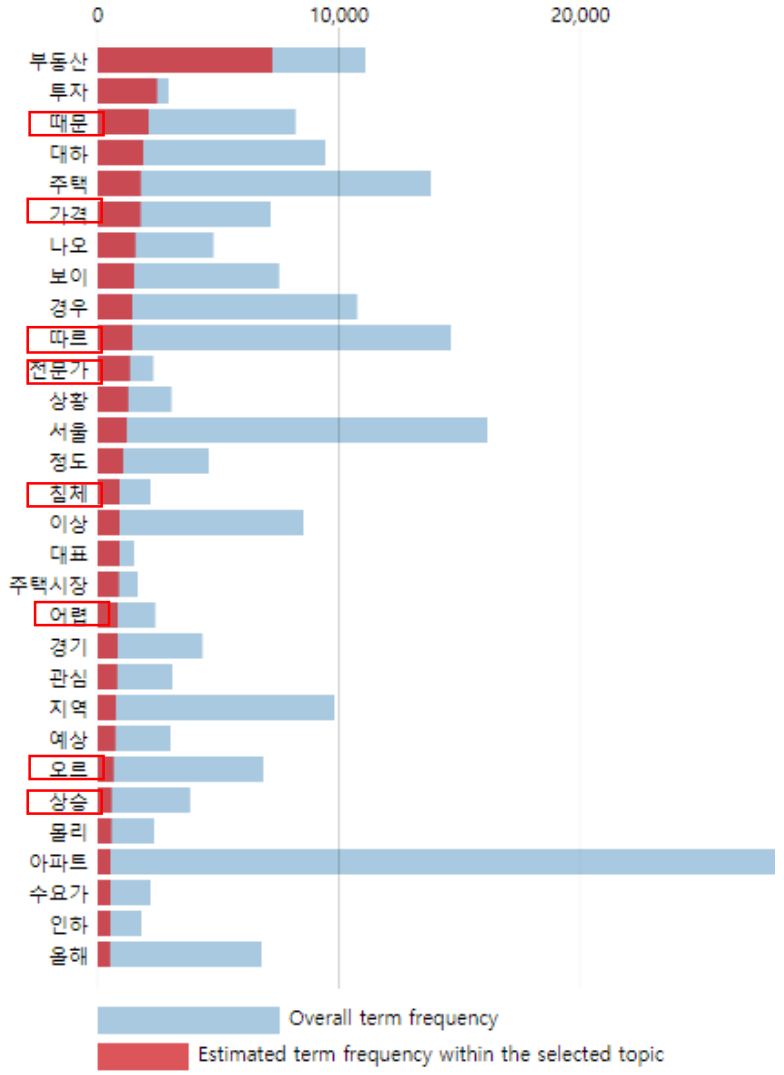
- “침체”, “어렵”, “상승”, “오르”, “인하”, “전문가”, “때문” 과 같은 시세 동향과 부동산 투자에 관련된 단어들이 자주 등장

- Topic3는 후행적 특성이 강해 해당 토픽에 할당된 기사는 이후 분석에서 제거

(예시) Topic 3

지난해는 청담동 청담마크힐스 183.5㎡ 면적 아파트가 50억원에 팔리면서 1위에 올랐으나...

그 동안 교육특구로 대표돼 새 학기를 앞두고 전셋값이 치솟았던 서울 강남구와 양천구 등 이 올해에는 가격 약세를 면하지 못해...



Topic Analysis

2 LDA - Topic 6 Analysis

- “예정”, “예상”, “시작”, “보이”, ”앞두” 와 같은 신규 아파트 단지 조성 계획과 관련된 단어들이 자주 등장

- Topic6 는 선행적 성격이 강해 이후 해당 토픽에 할당된 기사는 향후 분석에 사용

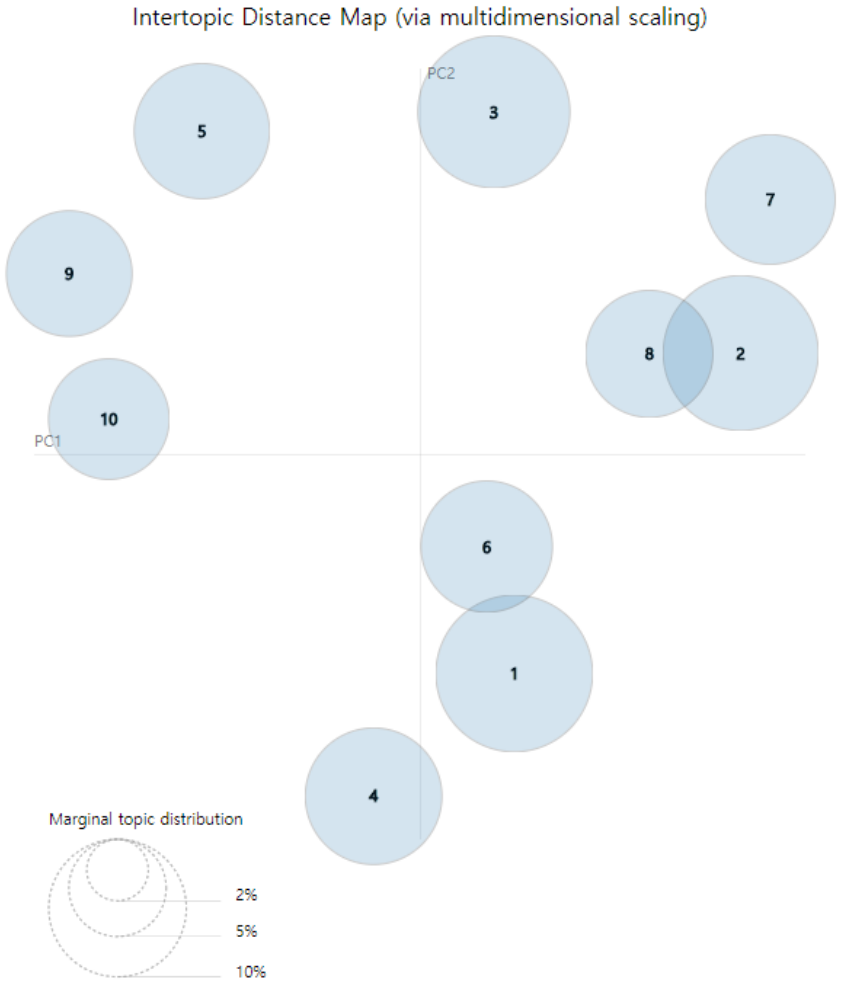
(예시) Topic 6

... 공장 증설이 올 해 말까지 한시적으로 허용된다 정부는 ...산업 집적 활성화 및 공장설립법 시행령 개정안 등을 의결했다

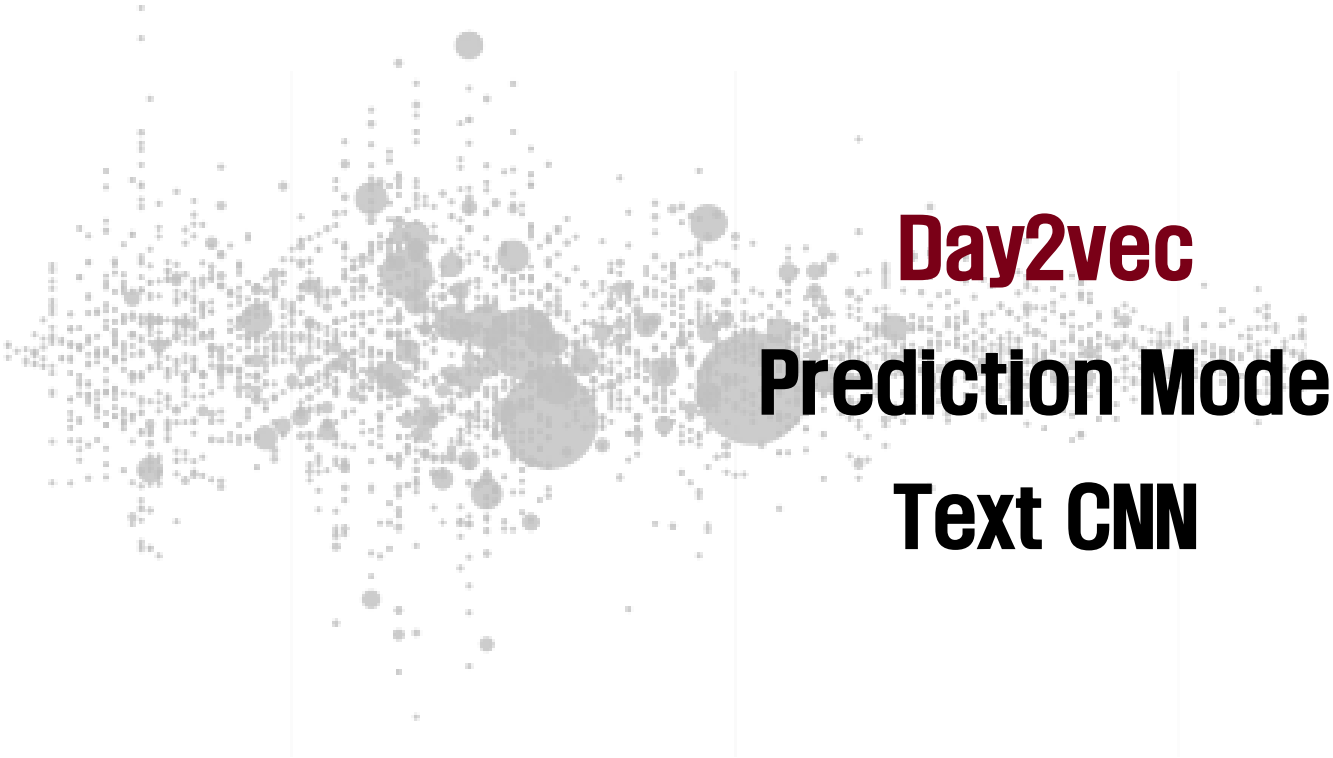
조세연구원은 미국일본 등 주요국의 부동산 보유세 실태 분석을 토대로 한국의 보유세 제에 대해 일부 조정이 필요하다는 결론을...

Article Selection

Selection Result



- 위와 같은 Topic 분석과 Topic PCA 결과를 참고하여 Topic 3과 7을 제거하고 나머지 8개 토픽에 할당된 기사들을 선택
- Topic의 선/후행 분류가 애매한 경우 데이터셋에 포함하여 이후 분석 진행
- 총 12,000건의 기사 중 8,800개의 기사 선택



Day2vec

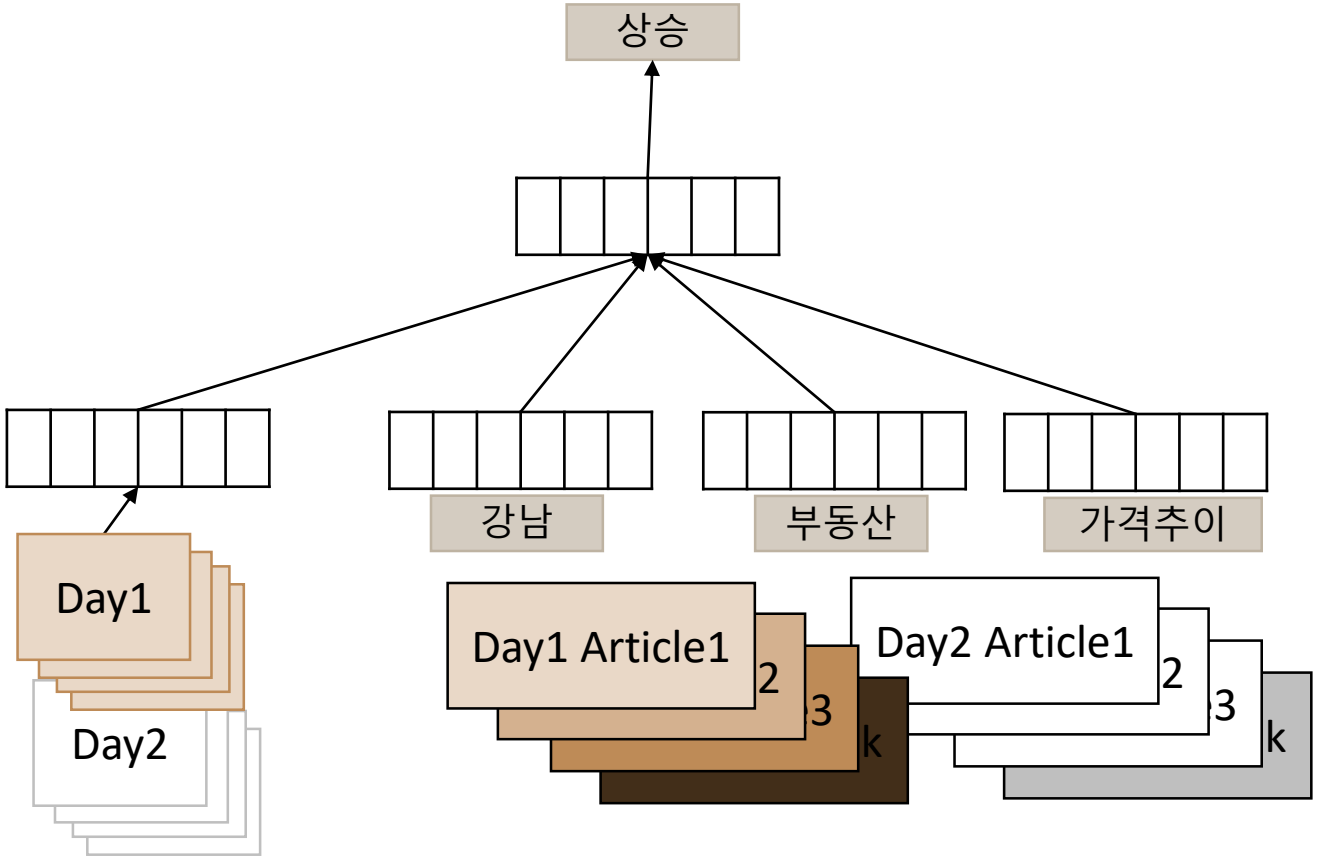
Prediction Model

Text CNN

Prediction

Day 2 Vector

- 개별 Article을 Doc2vec의 Document, Document Label을 Day Label처럼 사용하여 Day를 벡터로 임베딩.
- PV-DM, PV-DBOW를 통해 만든 128차원의 벡터를 concatenate 하여 256차원의 Day vector 사용



종속변수(상승 보합 하락) 기준 정립

- 집값이 10년 동안 2배 오르고 소비자 물가가 같은 기간동안 3배 올랐다면 이득? 손해?

- 대한민국의 집값과 물가는 최근 10년간 꾸준히 증가 추세였으므로 ***상승/보합/하락의 기준은 실질가격의 반영이 필요.*** *(명목주택가격은 물가상승률 반영이 된 것)*

- 집값의 기준으로 '아파트 전세가격지수' 를 사용하였으며 '전국' 기준 2007년 3월부터 2018년 4월까지 (133개월) 60->105로 상승-> 평균적으로 6개월에 2.04 상승 -> **'보합' 의 기준으로 사용**

Prediction

Input/Output Description

Train data : 2007년 3월 1일을 기준으로 3,000일간 (약 8.2년)의 Day_vector

Test data : 2007년 3월 1일을 기준으로 3001일 ~ 3900일 (약 2.4년)의 Day_vector

Y label : '아파트 전세가격지수' k월 1일부터 30일까지의 Label 모두 **k+' 예측기간'** 개월 이후 지수의 **'상승/보합/하락' 여부(One-hot)로 사용**

❖예측기간이 6개월일 때 상승/보합/하락의 기준, **2.5, 1.7은 상승/보합/하락일 수가 최대한 균등하게 나뉘는 숫자로 선정.**

상승 : 전세 가격지수가 6개월 전보다 **2.5이상 상승**

보합 : 전세 가격지수가 6개월 전보다 **1.7 초과, 2.5 미만 상승**

하락 : 전세 가격지수가 6개월 전보다 **1.7 이하 상승**

지역 : 전국

예측기간 : 3,6,9,~~12~~ 개월을 사용

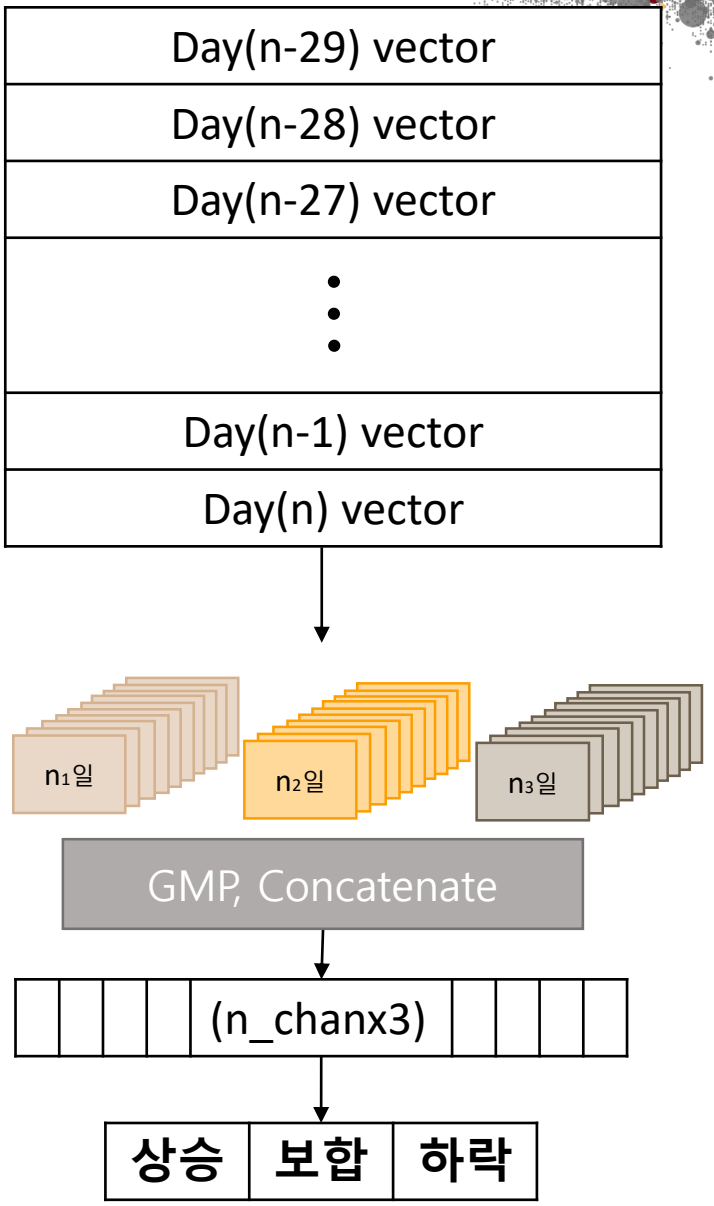
Prediction

1. Text CNN

- Day_vector_size : 128*2 (DM+DBOW)
- Input_shape : (256x30)
- Filter size : (7,14,21) / (4,8,12) x 256
- N_channel : 16, 32, 64
- Dropout : 0.4
- Optimizer : Adam(0.0001)

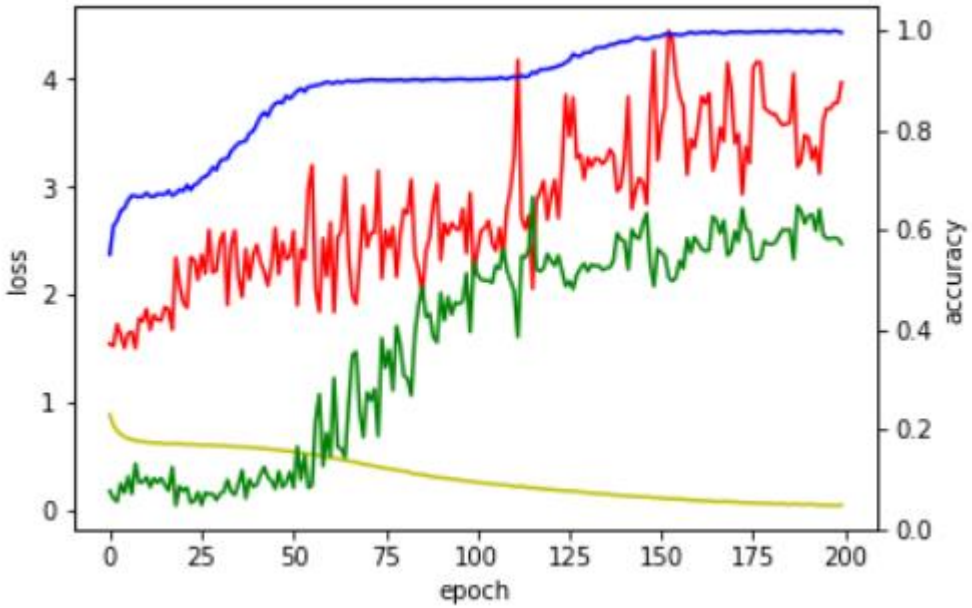
* Day vector는 CNN 학습 중 변하지 않는 Static Model을 사용

* 평가 척도는 마지막 10 epoch중 최소,최대 Val_accuracy로 사용



Prediction Result

- Parameter의 변화에 따라 성능의 차이가 심하고 Val_Accuracy는 상승하여도 Val_loss가 함께 증가하는 추세로 모델의 안정성이 부족
- 200 epoch도 되지 않아 Train_Accuracy가 거의 1에 근접하고 Train_Loss가 0에 근접하는걸 보아 데이터의 개수가 충분하지 않았음을 유추해 볼 수 있음



N_channel \ Filter size	16	32	64
(4,8,12)	50.4~55.7	57.1~64.7	29.5~33.3
(7,14,21)	41.7~48.1	30.1~34.9	44.1~53.3

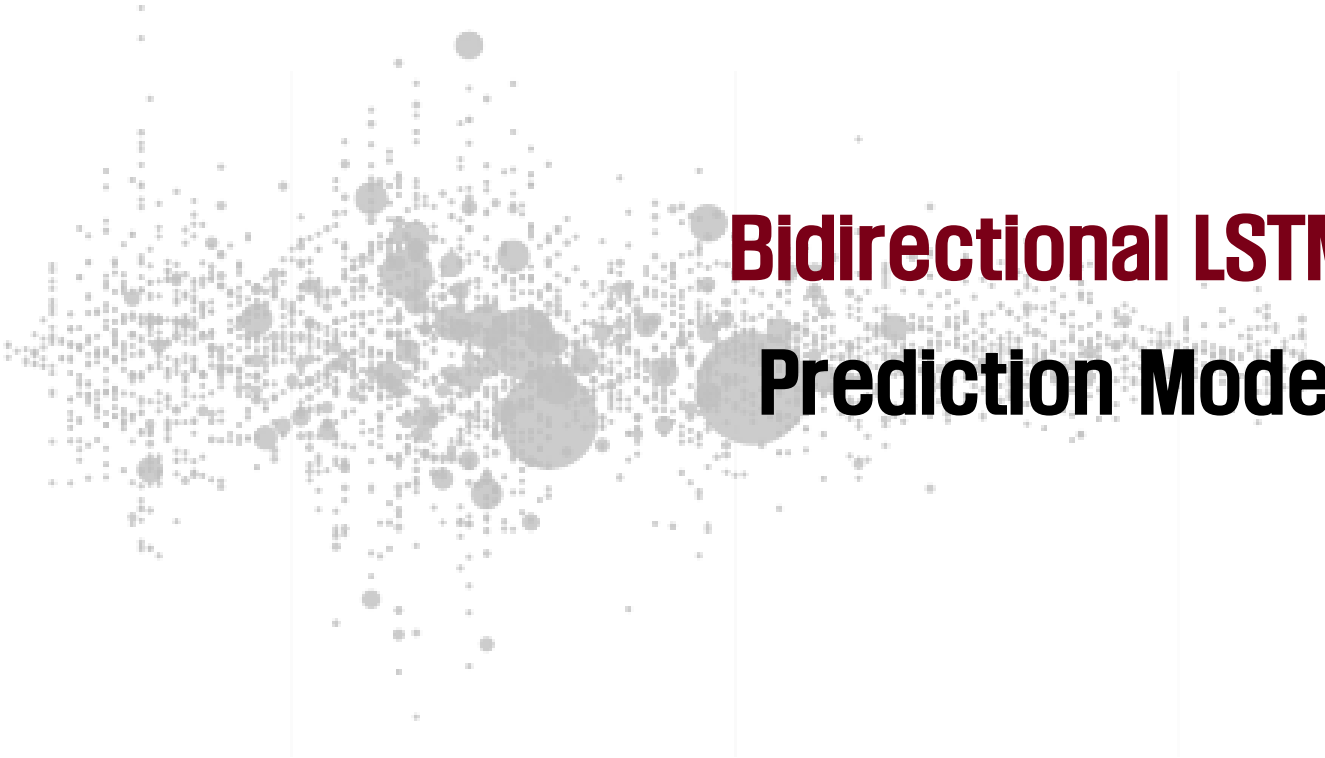
Prediction Result

- 일반적인 CNN 모델이 Channel의 개수가 변한다고 해서 정확도의 편차가 크지 않기에,
  다른 parameter는 고정 후 더 작은 크기의 Day vector size, Random seed를 바꿔가며 재실험.

- 아래 실험 결과를 볼 때 최적의 Parameter를 찾아 60%의 성능을 나타내었다 단정짓기 보다,
  성능이 나쁜 모델은 학습된 결과물이 상승을 더 많이 예측하도록, 좋은 모델은 하락을 더 많이
  예측하도록 편향되었을 가능성이 높음

- **평균적으로 약 45%의 예측 정확도**를 보여 만약 충분히 많은 모델을 앙상블 한다면 그 정도의 예측
  정확도를 기대

Random_Seed / Day_Vector	1	2	3	4	5	Average
64x2	43.5	39.4	50.3	60.8	35.1	**45.82**
32x2	60.8	49.4	34.0	40.7	32.2	**43.42**



Bidirectional LSTM Prediction Model

Prediction Result

Computing Power
CPU: i5-7500 @ 3.40GHz
RAM : DDR4 16.0GB
GPU: NVIDIA 1060

2. Bidirectional LSTM

Hyperparameter & Layer wrappers

- **Day_Vector : 32x2**
- Hidden_Size : 64
- Flow_Matrix : (4043, 30, 64)
- Input Shape : (3000, 30, 64)
- **Batch size : 5**
- N_timesteps : 256
- Epoch : 10
- Activation Function :
  - Softmax, Sigmoid, ReLU
- Optimizer : RMSprop, Adam
- Loss : Categorical-crossentropy

Day_Vector : 32x2
RMSprop

Activation Function	Val_accuracy	Val_loss
Softmax	**27.33%**	**6.0661**
Sigmoid	24.67%	2.6402
ReLU	24.67%	nan

Day_Vector : 32x2
Adam

Activation Function	Val_accuracy	Val_loss
Softmax	26.00%	5.0867
Sigmoid	**28.33%**	**3.2260**
ReLU	**29.00%**	**11.0068**

✓ 선행연구[1] **에 의하면 Batch size에 따라 학습 속도 및 성능이 달라짐을 직접 확인함.**

1) **Batch size : 5 , Val_acc : 24.67%** (ReLU, RMSprop)
2) **Batch size : 1 , Val_acc : 30.67%, Val_loss : 11.1752** (ReLU, RMSprop)

You, Y.; Zhang, Z.; Hsieh, C.; Demmel, J.; Keutzer, K.; 2018. ImageNet Training in Minutes. Computer Vision and Pattern Recognition. arXiv preprint arXiv:1709.05011v10

Alex Graves and Jurgen Schmidhuber, Framewise Phoneme Classification with Bidirectional LSTM and Other Neural Network Architectures, 2005

Prediction Result

2. Bidirectional LSTM

Hyperparameter & Layer wrappers

- **Day_Vector : 64x2**
- Hidden_Size : 64
- Flow_Matrix : (4043, 30, 128)
- Input Shape : (3000, 30, 128)
- **Batch size : 5**
- N_timesteps : 256
- Epoch : 10
- Activation Function :
  - Softmax, Sigmoid, ReLU
- Optimizer : RMSprop, Adam
- Loss : Categorical-crossentropy

Day_Vector : 64x2
RMSprop

Activation Function	Val_accuracy	Val_loss
Softmax	24.67%	2.8301
Sigmoid	**27.67%**	**3.2003**
ReLU	24.67%	11.9211

Day_Vector : 64x2
Adam

Activation Function	Val_accuracy	Val_loss
Softmax	**28.00%**	**3.6295**
Sigmoid	26.67%	4.2174
ReLU	**30.67%**	**11.1752**

You, Y.; Zhang, Z.; Hsieh, C.; Demmel, J.; Keutzer, K.; 2018. ImageNet Training in Minutes. Computer Vision and Pattern Recognition. arXiv preprint arXiv:1709.05011v10

Alex Graves and Jurgen Schmidhuber, Framewise Phoneme Classification with Bidirectional LSTM and Other Neural Network Architectures, 2005

Prediction Result

2. Bidirectional LSTM

Hyperparameter & Layer wrappers

- Day_Vector : 64x2
- Hidden_Size : 64
- Flow_Matrix : (4043, 30, 128)
- Input Shape : (3000, 30, 128)

✓ **Batch size : 1**

- N_timesteps : 256
- Epoch : 10
- Activation Function :
  - Softmax, Sigmoid, ReLU
- Optimizer : **Adam**
- Loss : Categorical-crossentropy

Day_Vector : 64x2
Adam (Epoch 10)

Activation Function	Val_accuracy	Val_loss
Softmax	**30.67%**	**5.3342**
Sigmoid	30.33%	2.8683
ReLU	24.67%	12.1423

Epoch : **30**, Softmax, Adam
Val_accuracy : 29.67% Val_loss : 3.6778

Epoch : **30**, **Sigmoid**, Adam
Val_accuracy : **33.67%** Val_loss : **3.7994**

Epoch : **30**, ReLU, Adam
Val_accuracy : 24.67 % Val_loss : 1.1921e-07

정형 데이터

Prediction Model

(Monthly, 전국 시·도)

3. 정형 데이터 모델

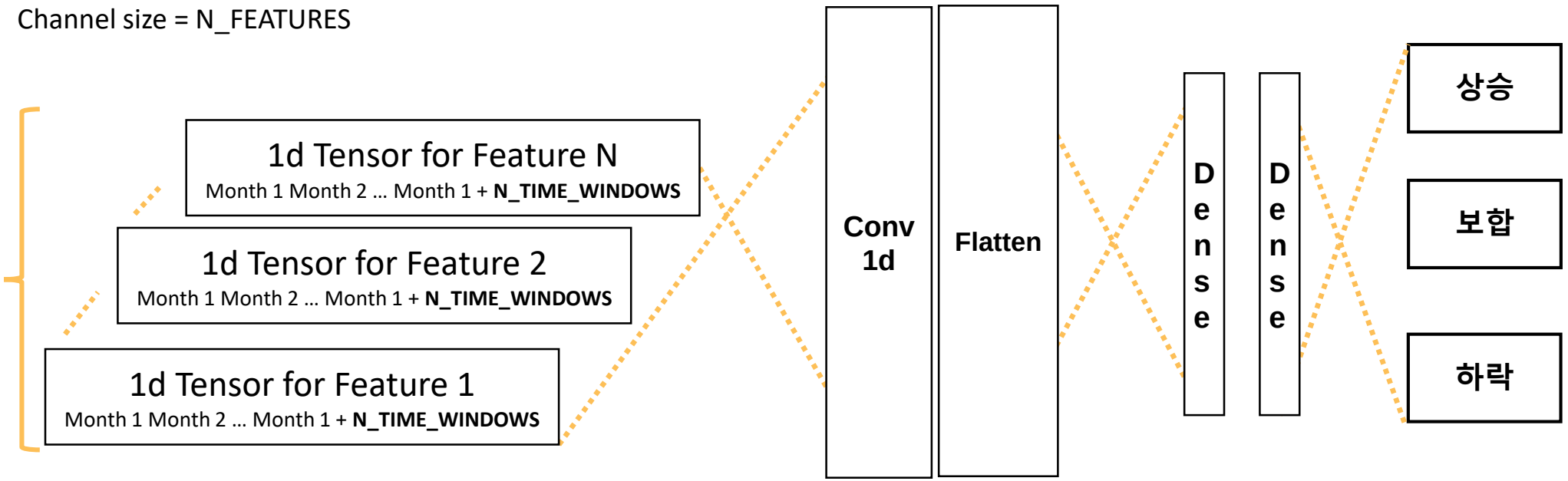
Data Description

Train : Test = 7:3

- **전국** 월별 기준 시계열 데이터 (기간 : 2003년 10월~현재)
- Input = 은행 대출 , 은행 예/적금 , 주택 Bubble Index, CD금리, M2통화량, 신용경색INDEX , 매매거래지수, 전세거래지수, 대출위험INDEX, 미분양율 등
- Output= 아파트 전세가격지수 (2015.12년 가격을 기준(=100)으로 Averaging 한 지수)

3. 정형 데이터 모델 #1

Simple Temporal CNN based by Monthly Feature Flow



❖ Multilayer Perceptron(MLP)로 Month Feature Vector를 Concatenate해서 예측하여 했으나, 학습이 잘 되지 않았음.

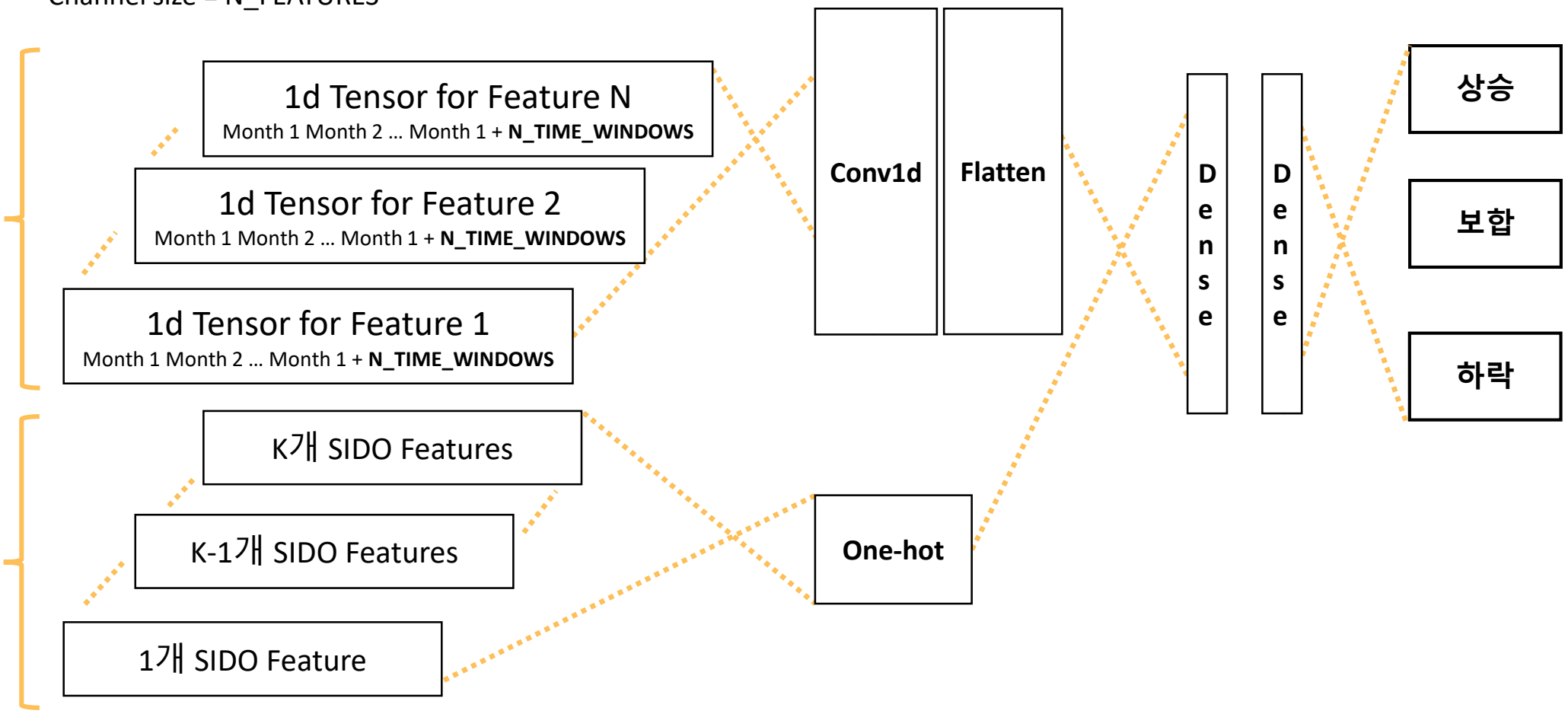
✓ **차후 비정형 데이터 모델과의 통일성도 고려했을 경우 CNN 이 적합하다고 판단**

- Conv1d를 이용 Flatten 한 후의 Embedding value와 그리고 dense layer를 한번 더 거치도록 처리

3. 정형 데이터 모델 #2

Conditional CNN based on Sido(전국 시도)

Channel size = N_FEATURES



Prediction Result

2. 정형데이터 모델 – 결과

Hyperparameter

- N_Features = 22
- N_Time_Windows = 50
- N_Month_Period_To_Predict = 6
- Activation Function(Layer) : ReLU
- Activation Function(Dense) : Softmax
- Optimizer : Adadelta
- CNN
  - Layer count size
  - Kernel size
  - Filter dimensionality size
- Dense
  - Hidden Node size
  - Layer count size

〈모델 성능 결과〉

✓ Simple Temporal CNN based by Monthly Feature Flow

Val_acc: 0.3000 **(30.00 %),** Val_loss : 11.2827

✓ Conditional CNN based on SIDO (전국 시도)

Val_acc: 0.6666 **(66.66%),** Val_loss : 5.3727

Conclusions

1. 필요한 데이터의 개수가 예상보다 매우 많이 필요

- Day를 벡터화 시키려면 충분한 기사가 필요하고, CNN을 잘 학습 시키려면 많은 Day 벡터가 필요
- Doc2vec과 CNN을 모두 잘 학습할 정도의 양의 데이터 확보가 어려우며 컴퓨팅 파워도 뒷받침 되어야 함

2. Text CNN을 사용한다고 해서 꼭 범주형으로만 예측을 해야 했는지 의문이 들었음

- 과거에는 집값이 가파르게 상승했기에 실질 가격을 고려해도 상승세인 구간이 많았지만
  최근 (Test set)에서는 실질 가격 측면에서 하락세인 구간이 많음 -〉 train, test set이 동질적이지 않음
- Loss function인 'Categorical cross-entropy'는 상승한 집값을 보합으로 예측할 때와
  하락으로 예측할 때의 penalty가 같음

Conclusions

3. Embedding된 Day vector로 RNN-Bidirectional LSTM 모델에 적용했으나 학습이 잘 되지 않음. 비록 RNN이 시계열 데이터의 더 복잡한 구조를 표현할 수 있다는 장점이 있으나 실제 적절한 weight로 학습시키는데 더 어려움이 있었음

4. 정형 데이터와의 예측 결과를 비교해보면 비정형 데이터를 통해 예측한 모델로도 비슷하거나 때로는 더 나은 정확도를 얻었음

5. 예측에 불필요한 기사(Topic 3)를 제거 한 후 RNN 적용했을 때 (Test acc 8% → 18% (10%p ↑)) 을 보였음.

6. 부동산 경기에 대한 심리적인 영향과 정책에 대한 효과를 고려할 때 비정형 데이터의 활용은 매우 중요하며 연구의 필요성이 있음

Bidirectional LSTM

Computing Power

CPU: i5-7500 @ 3.40GHz

RAM : DDR4 16.0GB

GPU: NVIDIA 1060

embedding 후의 Bidirectional LSTM Model

model.add(Embedding(max_features, embedding_size, input_length=maxlen)

<Hyperparameter>

- Hidden_Size : 64
- Flow_Matrix : (4073, 30, 256)
- max_features: 4073
- Batch size : 1
- N_timesteps : 256
- Dropout : 0.6
- Epoch : 100
- Activation Function : Sigmoid
- Optimizer : Adam

Layer (type)	Output Shape	Param #
embedding_2 (Embedding)	(None, 30, 256)	1042688
bidirectional_2 (Bidirection	(None, 30, 128)	164352
dropout_2 (Dropout)	(None, 30, 128)	0
dense_2 (Dense)	(None, 30, 3)	387

Total params: 1,207,427
Trainable params: 1,207,427
Non-trainable params: 0

Train...

✓ 에러 해결하지 못 함

Train on 3000 samples, validate on 866 samples
Epoch 1/100
32/3000 [.....] - ETA: 3:23 - loss: 1.0957 - acc: 0.4333

```
InvalidArgumentError: indices[26,3] = -1 is not in [0, 4073)
[[Node: embedding_2/Gather = Gather[Tindices=DT_INT32, Tparams=DT_FLOAT, validate_indices=true,
_device="/job:localhost/replica:0/task:0/device:CPU:0"](embedding_2/embeddings/read, embedding_2/Cast)]]
```

Bidirectional LSTM

❖ 앞선 Text CNN 모델의 절차에서 합성곱신경망이 아닌 RNN 모델 적용 구조 : 3-dimenssions

[Samples, Timesteps, Features] (e.g. one sequence = 1 sample)

일방통행이 아닌 양방향 Receiver

✓ TimeDistributed layer : 256 timesteps of 20 outputs, it will now receive 256 timesteps of 40 (20 units + 20 units) outputs.

<Hyperparameter>

- Flow_Matrix : (4043, 30, 256)
- Batch size : 1
- N_timesteps : 256
- Epoch : 1,000
- Activation Function : Sigmoid
- Optimizer : Adam

X_train_shape	(3000, 30, 256)
Y_train_shape	(3000, 3)
X_test_shape	(896, 30, 256)
Y_test_shape	(896, 3)

X_train_shape	(1, 256, 1)
Y_train_shape	(1, 256, 1)
X_test_shape	(896, 30, 256)
$\hat{Y}$_shape	(1, 256, 1)

Layer (type)	Output Shape	Param #
bidirectional	(Bidirection (None, 256, 40)	3520
time_distributed	(TimeDist (None, 256, 1)	41

Total params: 3,561
Trainable params: 3,561
Non-trainable params: 0

*Source: Alex Graves and Jurgen Schmidhuber, Framewise Phoneme Classification with Bidirectional LSTM and Other Neural Network Architectures, 2005

Appendix

Bidirectional LSTM

〈모델 성능 결과〉

Train loss : 0.0343

Train Accuracy : around 95.70% ~ 99.22%

```
Epoch 1/1
 - 0s - loss: 0.0454 - acc: 0.9805
Epoch 1/1
 - 0s - loss: 0.0807 - acc: 0.9609
Epoch 1/1
 - 0s - loss: 0.0386 - acc: 0.9883
Epoch 1/1
 - 0s - loss: 0.0873 - acc: 0.9570
Epoch 1/1
 - 0s - loss: 0.0510 - acc: 0.9766
Epoch 1/1
 - 0s - loss: 0.0343 - acc: 0.9922
```

✓ Only have four errors

```
Expected: [0] Predicted [0]     Expected: [0] Predicted [1]
Expected: [0] Predicted [0]     Expected: [0] Predicted [1]
Expected: [0] Predicted [0]     Expected: [0] Predicted [1]
Expected: [0] Predicted [0]     Expected: [0] Predicted [1]
Expected: [0] Predicted [0]     Expected: [1] Predicted [1]
Expected: [0] Predicted [0]     Expected: [1] Predicted [1]
Expected: [0] Predicted [0]     Expected: [1] Predicted [1]
Expected: [0] Predicted [0]     Expected: [1] Predicted [1]
Expected: [0] Predicted [0]     Expected: [1] Predicted [1]
Expected: [0] Predicted [0]     Expected: [1] Predicted [1]
Expected: [0] Predicted [0]     Expected: [1] Predicted [1]
Expected: [0] Predicted [0]     Expected: [1] Predicted [1]
Expected: [0] Predicted [0]     Expected: [1] Predicted [1]
Expected: [0] Predicted [0]     Expected: [1] Predicted [1]
Expected: [0] Predicted [0]     Expected: [1] Predicted [1]
Expected: [0] Predicted [0]     Expected: [1] Predicted [1]
Expected: [0] Predicted [0]     Expected: [1] Predicted [1]
Expected: [0] Predicted [0]     Expected: [1] Predicted [1]
Expected: [0] Predicted [0]     Expected: [1] Predicted [1]
Expected: [0] Predicted [0]     Expected: [1] Predicted [1]
Expected: [0] Predicted [0]     Expected: [1] Predicted [1]
Expected: [0] Predicted [0]     Expected: [1] Predicted [1]
Expected: [0] Predicted [0]     Expected: [1] Predicted [1]
Expected: [0] Predicted [0]     Expected: [1] Predicted [1]
Expected: [0] Predicted [0]     Expected: [1] Predicted [1]
Expected: [0] Predicted [0]     Expected: [1] Predicted [1]
Expected: [0] Predicted [0]     Expected: [1] Predicted [1]
Expected: [0] Predicted [0]     Expected: [1] Predicted [1]
Expected: [0] Predicted [0]     Expected: [1] Predicted [1]
Expected: [0] Predicted [0]     Expected: [1] Predicted [1]
Expected: [0] Predicted [0]     Expected: [1] Predicted [1]
Expected: [0] Predicted [0]     Expected: [1] Predicted [1]
Expected: [0] Predicted [0]     Expected: [1] Predicted [1]
Expected: [0] Predicted [0]     Expected: [1] Predicted [1]
Expected: [0] Predicted [0]     Expected: [1] Predicted [1]
Expected: [0] Predicted [0]     Expected: [1] Predicted [1]
Expected: [0] Predicted [0]     Expected: [1] Predicted [1]
```

*Source: Alex Graves and Jurgen Schmidhuber, Framewise Phoneme Classification with Bidirectional LSTM and Other Neural Network Architectures, 2005

QUESTION & ANSWERS SESSION