



## 저작자표시-비영리-변경금지 2.0 대한민국

이용자는 아래의 조건을 따르는 경우에 한하여 자유롭게

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

다음과 같은 조건을 따라야 합니다:



저작자표시. 귀하는 원저작자를 표시하여야 합니다.



비영리. 귀하는 이 저작물을 영리 목적으로 이용할 수 없습니다.



변경금지. 귀하는 이 저작물을 개작, 변형 또는 가공할 수 없습니다.

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

저작권법에 따른 이용자의 권리는 위의 내용에 의하여 영향을 받지 않습니다.

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

[Disclaimer](#)

석사학위논문

모기 활동성 예측을 위한 기상  
데이터에서 인접성 향상을 위한  
연구

A Study on Improving Adjacency in Weather Data for  
Predicting Mosquito Activity



국민대학교 일반대학원  
컴퓨터공학과 컴퓨터공학전공

이 동 우

2018

# 모기 활동성 예측을 위한 기상 데이터에서 인접성 향상을 위한 연구

A Study on Improving Adjacency in Weather  
Data for Predicting Mosquito Activity

지도교수 황 선 태

이 論文을 工學碩士學位 請求論文으로 提出함

2019 년 2 월

국민대학교 일반대학원  
컴퓨터공학과 컴퓨터공학전공  
이 동 우  
2018

이 동 우 의

工學碩士學位 請求論文을 認准함

2019 년 2 월



審査委員長    강 승 식    ㉠

審査委員        김 준 호    ㉠

審査委員        황 선 태    ㉠

국민대학교 일반대학원

# 차 례

그림 차례 .....	iii
표 차례 .....	v
국문요약 .....	vii
제 1장 서론 .....	1
제 2장 관련 연구 .....	3
2.1 경험적 모델을 활용한 환경 분야 예측/예보 사례 .....	3
2.1.1 미세먼지 수치 예보 .....	3
2.1.2 태양 입자 유입 예측 .....	4
2.2 시계열 데이터에서의 CNN적용 .....	5
제 3장 이전 연구 .....	6
3.1 데이터의 구조 .....	6
3.2 신경망 모델 설계 .....	13
3.3 모델을 통한 학습 및 결과 .....	23
3.4 데이터의 전처리 방안 .....	27
제 4장 전처리 과정을 통한 실험 .....	28
4.1 누적 기상 요소 선택 .....	28
4.2 인접성 향상을 위한 전처리 .....	30
4.3 데이터 구성조합의 선택 및 수치 비교 .....	35
제 5장 기존의 신경망을 통한 학습 .....	43
5.1 선행 연구와의 결과 비교 .....	43
제 6장 결론 및 향후과제 .....	44

참 고 문 헌 .....	45
Abstract.....	47



## 그림 차례

Figure 1- 1	모기 예보 서비스 예시	1
Figure 2-1	환경부 미세먼지 예보 서비스	3
Figure 2- 2	우주 전파 환경 정보 등급표	4
Figure 2- 3	Facebook 의 기계번역의 신경망 작동 개념	5
Figure 3- 1	Convolutional Neural Network 과정	13
Figure 3- 2	Google Inception V4 신경망의 구조	15
Figure 3- 3	Stem 구조	17
Figure 3- 4	Inception-A 구조	18
Figure 3- 5	Reduction-A 구조	19
Figure 3- 6	Inception-B 구조	20
Figure 3- 7	Reduction-B 구조	21
Figure 3- 8	Inception-C 구조	22
Figure 3- 9	데이터 분리 후 학습 과정 구조	26
Figure 4- 1	규칙적 누적 값 추출	28
Figure 4- 2	18 개의 기상 요소 누적 값	31
Figure 4- 3	18 개의 누적 값들 중 인접 개수 2 개의 예시	31
Figure 4- 4	18 개의 누적 값들 중 인접 개수 3 개의 예시	31
Figure 4- 5	Google Inception 신경망의 간소화	35
Figure 4- 6	Random Feature r-squared score 값 비교	39
Figure 4- 7	Regular Feature r-squared score 값 비교	39
Figure 4- 8	총 누적 개수(n)와 인접 기상 요소 개수(r) 선택	41

Figure 4- 9 [Figure4-8]에 대한 수도코드 .....	42
--	----





## 표 차례

Table 3- 1 기상자료 CSV 데이터 원본 예시 .....	7
Table 3-2 모기 포집 지역(19 곳).....	8
Table 3-3 2 차원 CSV 데이터 예시(5 월 28 일).....	9
Table 3- 4 각 기상 요소 별 정규화 변환 수식 .....	12
Table 3-5 예측 결과의 쏠림 현상 .....	23
Table 3- 6 섞이는 부분의 예시 .....	24
Table 3- 7 기상 요소 분리 후 예측 결과 .....	25
Table 4- 1 누적 값 추출 .....	30
Table 4- 2 18 C 3 행의 예시 .....	32
Table 4- 3 18 C 2 행의 예시 .....	33
Table 4- 4 Regular Feature r-squared score .....	37
Table 4- 5 Random Feature r-squared score .....	37
Table 5- 1 완전한 Goolge Inception V4 를 통한 학습의 결과.	43

## 국문 요약

# 모기 활동성 예측을 위한 기상 데이터에서 인접성을 향상하는 전처리에 대한 연구

국민대학교 대학원 컴퓨터공학과

이동우

모기 활동성 예측을 위해서는 기상 데이터와 모기 포집량 데이터를 활용한다. 기상 데이터와 모기 포집량 데이터는 수치 데이터로 인접성과 지역 정보가 없고, 매일 측정 한 시계열 데이터이다. 딥러닝 알고리즘은 크게 2가지가 있다. RNN(Recurrent Neural Network) 알고리즘과 CNN(Convolutional Neural Network) 알고리즘이다. 일반적으로 시계열 데이터의 학습을 위해서는 RNN 알고리즘을 사용하고 이미지 데이터의 학습을 위해서는 CNN 알고리즘을 사용한다. CNN 알고리즘은 데이터에 Filter를 적용하고 특징을 추출하는 과정을 반복적으로 진행해 지역 정보와 Locality 정보를 추출해내는데 용이하기 때문이다. 하지만 최근 시계열 데이터에 CNN 알고리즘을 적용하는 연구가 활발히 진행되고 있다. CNN 알고리즘을 사용하기 위해 시계열 데이터에 전처리가 필요하다.

주제어 : 딥러닝, CNN, 전처리, 기상 데이터

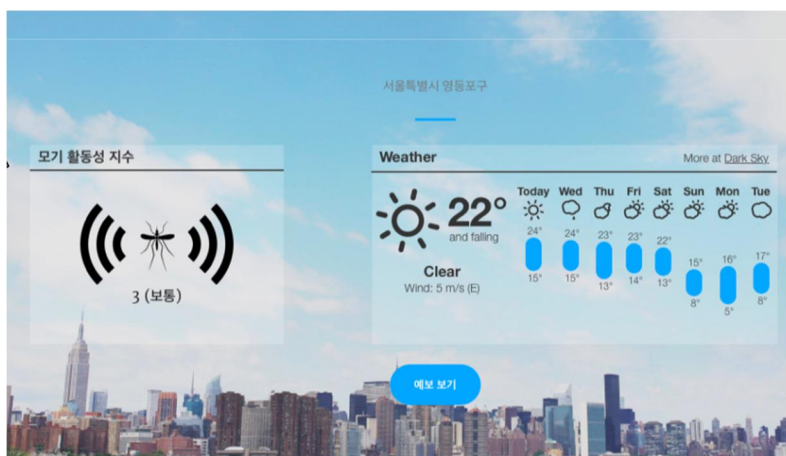
본 논문에서는 실험에서 다루는 시계 열 데이터에서 CNN 알고리즘을 활용하여 좋은 결과를 도출할 수 있는 인접성을 향상하는 데이터의 전처리 과정을 제시한다. 그리고 기존 선행 연구와 비교하여 성능 개선을 보여주고자 한다.



## 서론

시민들은 삶의 질 향상에 관심이 높아지고 있다. 이를 위해 서울시 영등포구에서는 모기 활동성 지수를 예보하고 있다.[1][2] 이 예보는 [Figure 1-1]처럼 오른쪽의 각 요일 별 최저기온, 최고기온, 운량, 현재 날씨 그리고 왼쪽의 모기 지수를 1~4단계로 나누어서 보여주고 있다 이를 토대로 사람들에게 단계별 행동 요령을 전달하고 있다.

이 시스템의 예보 모델은 과거(2011년~2015년) 기상 자료와 모기 포집량을 토대로 만들었다. 모기 포집 수는 영등포구의 포집 지역 총 19곳의 자료를 사용했고, 특정 수식으로 1~8단계로 등급화를 했다. 기상자료에는 평균 습도, 강수량, 강수일 수, 평균 온도, 최고 온도, 최저 온도가 있다. 기존 연구에서는 모기 관련 전문가의 연구 결과와 Random Forest 알고리즘을 사용하여 각 기상 요소별로 모기 활동성에 영향을 가장 많이 미치는 누적 일 수를 추출하였다.[1] 추출된 기상 요소 누적 값들은 평균 습도 5일 누적, 강수량 6일 누적, 강수일 29일 누적, 최고 온도 19일 누적, 평균 온도 16일 누적, 최저 온도 2일 누적이다.



[Figure 1-1 모기 예보 서비스 예시]

모기 포집 량 데이터는 정답 데이터에 해당한다. 선행연구에서 모기 포집 량을 마리 수 별로 등급화를 진행했다.[1] 20마리 이하면 1단계 20마리 이상일 경우  $\rightarrow \text{Ceiling}(\text{Log2}(\text{모기 포집 량}/10))$ 의 수식을 통해 2~8단계로 등급화 했다.

모기 예보 시스템은 이미 배포된 시스템이다. 따라서 시간이 흐를수록 데이터가 누적 됨에 따라 개선이 이루어져야 한다. 전문가의 연구 결과 없이 시스템 예측 율을 개선하고자 본 논문에서는 CNN알고리즘을 사용한다.

인접성이 뚜렷한 사진과 같은 데이터에 잘 맞는 CNN의 특성에 맞추려면 데이터의 인접성을 향상 시킬 필요가 있는데 이를 위해 본 논문에서는 가시적 인접성이 없는 시계 열 데이터인 기상 데이터에서 인접성이 향상된 조합을 찾아내는 전처리 방법을 제안한다. 이 전처리에서는 여러 데이터 조합의 예측 율을 간소화된 CNN 모델로 계산하여, 예측 율 향상 정도가 목표 오차율에 도달하면 멈추는 방식으로 적절한 데이터 조합을 찾아내도록 하였다. 그리고 완전한 CNN모델을 사용하여 적절한 데이터 조합에 대한 정확도를 이전 연구와 비교한다.

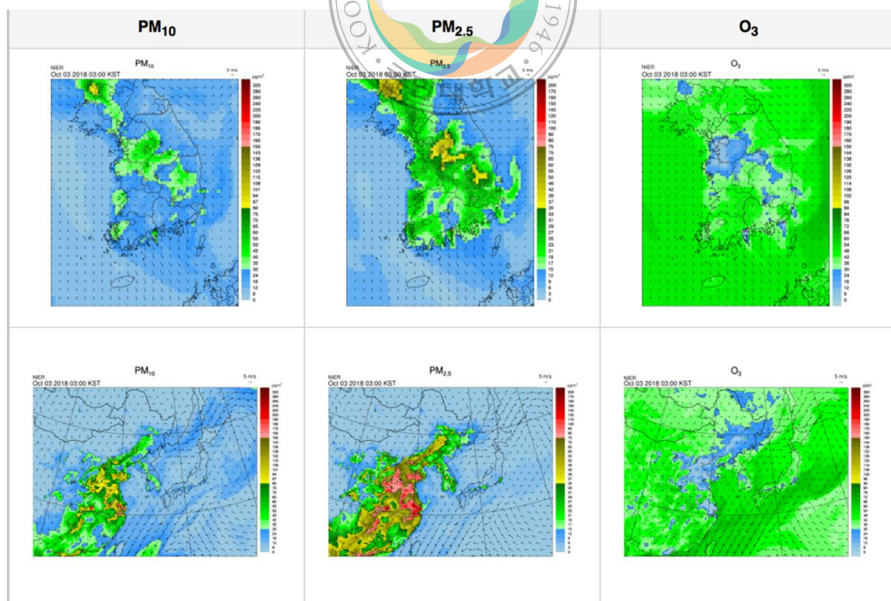
## 제 2장 관련 연구

### 2.1 경험적 모델을 이용한 예보 사례

#### 2.1.1 미세먼지 수치 예측

최근 봄, 가을 철 미세먼지 수치가 급격히 상승함에 따라 이에 대한 국민의 관심도가 굉장히 높아지고 있다. 하지만 현대인의 삶에서 미세먼지 노출은 불가피하다. 노출을 국민들로 하여금 최소화 하기 위하여

[Figure 2-1]처럼 환경부에서는 대기오염 농도를 예보를 매일 4회에 걸쳐 실시하고 있다.[3] 미세먼지를 비롯 여러 오염물질을 전국 총 330여 개의 측정소에서 수치를 측정해 모델을 개발하였다. 모기 예보 시스템 모델과 같이 과거 데이터 기반 경험적 모델을 사용했다.








[Figure 2-1 환경부 미세먼지 예보 서비스]

## 2.1.2 경험적 태양 입자 유입 예보

태양에서의 흑점은 11년 주기로 수가 증감한다. 태양의 흑점이 증가하는 시기에는 이 흑점이 폭발하여 태양폭풍이 발생한다. 이럴 경우 전파 방해가 발생해 휴대폰이 필수인 현대인의 경우 많은 불편함이 일어날 수 있다.

이에 국립전파연구원에서는 인공 신경망 모형을 기반으로 태양 흑점 폭발에 따른 태양 입자 경보 발생 확률을 예측하고 있다. 이 예측도 과거 경험적 데이터를 사용하고 있다. 또한 과거 데이터의 유사 패턴을 분석하여 단기적인 예측 데이터를 산출하는 방식으로 태양 양성자 입자를 현재시점으로 1일, 2일, 3일 예보 정보를 제공하고 있고 실측 데이터와 비교하여 정확도도 보여주고 있다. 태양 입자 경보 발생 확률 및 태양흑점 폭발 등급을 1일 /3일 /27일 간격으로 예보 서비스를 시행하고 있다. [4]

▶ 태양흑점폭발 (태양 X-ray 방출)

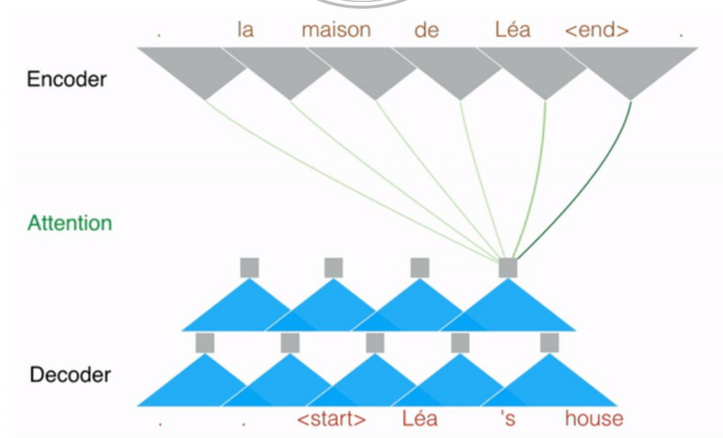
등급		예상되는 피해	관측값*	평균발생횟수
R1		<ul style="list-style-type: none"> <li>HF 통신 : 낮지역에서 HF 통신에 약한 장애가 발생하며 가끔씩 신호감쇄 현상이 나타남</li> <li>항법시스템 : 저주파 항법신호의 강도가 짧은 기간동안 약해짐</li> </ul>	$10^{-5}$	태양활동 1주기 당(11년) 약 2000회
R2		<ul style="list-style-type: none"> <li>HF 통신 : 낮지역에서 제한적으로 HF 통신이 두절되거나 수 심분 동안 신호감쇄현상이 나타남</li> <li>항법시스템 : 수 심분 동안 저주파 항법신호의 강도가 약해짐</li> </ul>	$5 \times 10^{-5}$	태양활동 1주기 당(11년) 약 350회
R3		<ul style="list-style-type: none"> <li>HF 통신 : 낮지역에서 HF 통신이 두절되거나 수 시간 동안 신호감쇄현상이 나타남</li> <li>항법시스템 : 수 심분 동안 저주파 항법신호의 강도가 약해짐</li> </ul>	$10^{-4}$	태양활동 1주기 당(11년) 약 175회
R4		<ul style="list-style-type: none"> <li>HF 통신 : 낮시간 광범위한 지역에 걸쳐 수 시간동안 HF 통신두절이 나타날 수 있음</li> <li>항법시스템 : 수 시간동안 항법데이터에 오류가 증가하여 저주파 항법 시스템이 두절될 수 있음</li> </ul>	$10^{-3}$	태양활동 1주기 당(11년) 약 8회
R5		<ul style="list-style-type: none"> <li>HF 통신 : 상당한 시간 낮지역에서 HF 통신 두절 발생(항공 및 해상 항해 시 HF 통신 불가)</li> <li>항법시스템 : 수 시간 동안 저주파 항법시스템이 두절될 수 있으며, GPS위성신호의 오류증가로 발지역까지 항법시스템 두절 현상이 영향을 줄 수 있음</li> </ul>	$2 \times 10^{-3}$	태양활동 1주기 당(11년) 약 1회

[Figure 2-2 우주 전파 환경 경보 등급 표]

## 2.2 시계열 데이터에서의 CNN 적용

FaceBook 기계 번역 시스템[5][6][Figure2-3]은 시계열 데이터인 자연어 처리 과정에서 CNN(Convolution Neural Network) 알고리즘을 썼다. 보통 시계열 데이터에서 RNN(Recurrent Neural Network) 알고리즘을 쓰는 것이 일반적인 방법인데, 최근 이처럼 CNN 알고리즘을 써서 좋은 결과를 내고 있다.

기존 기계 번역에서는 RNN을 사용한다. 지금까지 많은 연구자들이 RNN기반 기계 번역 기술에 시간과 노력을 투자했다. RNN은 문장을 번역하기 위해 문장을 왼쪽에서 오른쪽으로 순차적으로 분석할 수 있기에 뛰어난 알고리즘이기 때문이다. 하지만 CNN은 문장 의미와 구조를 전체적으로 분석할 수 있으며, 속도 또한 GPU 처리에 용이해 RNN보다 빠르다. FaceBook이 CNN으로 개발한 기계 번역 시스템은 이전 방법보다 11%가량 정확도가 향상되었다.



[Figure 2-3 FaceBook 기계 번역의 신경망 작동 개념]



## 제 3장 이전 연구

### 3.1 데이터의 구조

- 데이터의 수집

영등포구에 있는 DMS(Digital Mosquito Monitoring System, 디지털 모기 포집 시스템)와 AWS(Automatic Weather System, 자동 기상 관측 시스템)로부터 2011년~2015년 총 5년간의 모기 포집량과 기상자료를 얻었다.[1] 모기 포집량은 영등포구에 DMS가 존재하는 19곳으로부터 각 년도 별로 5월 1일부터 10월 31일까지 존재하고 기상 자료는 각 년도 별 1월 1일부터 12월 31일까지 습도, 강수량, 평균 온도, 최고 온도, 최저 온도 총 5가지 요소의 자료가 존재한다. 모기 포집량과 기상 자료 모두 엑셀 csv데이터 형식이다. 특정 지역은 모기 포집량이 일부 년도만 존재한다. 모기 포집량 데이터와 기상 데이터 모두 매일매일 존재하는 시계열 데이터이다. 또한 두 데이터 모두 값의 범위가 다르다.

[Table 3-1 기상자료 csv 데이터 원본 예시]

	습도(%)	강수량 (mm)	평균 온도(℃)	최고 온도(℃)	최저 온도(℃)	모기포집
2013-06-19	53.4	4.8	23.2	26.5	19.9	20
2013-06-20	59.2	8.8	23.6	25.8	21.4	30
2013-06-21	48.5	22.5	24.8	27.0	22.6	199
2013-06-22	64.2	70.4	24.8	27.7	21.9	520
2013-06-23	40.8	5.2	23.9	26.8	21	400
2013-06-24	50.2	0	23.9	26.9	20.9	328
2013-06-25	55.2	0	24.6	27.5	21.7	140
2013-06-26	54.2	2.7	24.7	27.8	21.6	128
2013-06-27	30	1	24.5	27.8	21.2	4
2013-06-28	29.8	102	23.9	28.0	19.8	28
2013-06-29	34.5	40	24.0	28.6	19.4	32
2013-06-30	62	5	24.1	28.8	19.4	45
2013-07-01	45.2	0	25.2	29.5	20.9	153
2013-07-02	70	0	25.0	28.4	21.6	842
2013-07-03	40.9	0	26.0	29.6	22.4	1000

[Table 3-2 모기 포집 지역(19 곳)]

모기 포집 지역
국회의사당, 대림 유수지, 당산 중학교, 동아 에코빌 아파트, 두암 어린이 공원, KBS, 김안과, 문래근린공원, 문래빗물펌프장, 살레시오, 신길근린공원, 신길어린이공원, 영등포보건소, 양평빗물펌프장, 양평1동유수지, 양평동노인복지회관, 여의도고등학교, 여의도공원, 윤중초등학교

### ● 데이터의 전처리

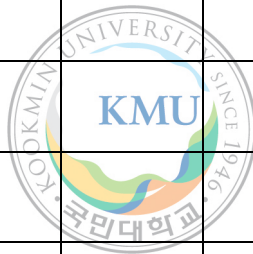
기존 연구는 기계 학습 알고리즘 중 하나인 Random Forest를 사용하여 모기 활동성에 가장 영향을 많이 주는 요소를 추출했다. 각 요소별로 누적(1~30)값들이 후보다. 추출한 결과 습도 5일 누적, 강수량 6일 누적, 강수일 29일 누적, 평균 온도 16일 누적, 최고기온 19일 누적, 최저기온 2일이 요소로 추출되었다. 강수일은 비가 온 날만 기록하여 계산해주었다.

그리고 CNN알고리즘에 알맞은 2차원 Matrix 형태의 CSV파일로 데이터 형식을 변형한다. [Table 3-3] 처럼 CSV파일의 가로는 당일 기준 30일 전까지의 날짜가 기록 되어 있고 세로는 평균 습도 1~30일 누적, 강수량 1일~30일 누적, 누적 강수일 1~30일 누적, 평균 온도 1~30일 누적, 최고 온도 1~30일 누적, 최저 온도 1~30일 누적 값이 차례로 기록돼있어 180개의 행, 30개열 총 180\*30의 누적 기상 데이터의 2차원 CSV데이터 파일로 데이터 구조를 변형한다. 강수량이 0인 경우는 비가 온 날이므로 누적 강수일에 포함시킨다.

[Table 3-3 2 차원 CSV 데이터 예시(5 월 28 일)]

	1일전	...	27일전	28일전	29일전	30일전
습도1일누적						
습도2일누적						
습도3일누적						
습도4일누적						
...						
습도30일누적					B	
강수량1일누적						
강수량2일누적						
...						
강수량30일누적						
강수일1일누적						
강수일2일누적						
...						
강수일29일누적						
강수일30일누적						

평균온도 1일 누적						
평균온도 2일 누적						
평균온도 3일 누적						
평균온도 4일 누적						
...						
평균온도 30일 누적						
최고온도 1일 누적						
최고온도 2일 누적						
...						
최고온도 30일 누적						
최저온도 1일 누적						
최저온도 2일 누적						
...						
최저온도 28일 누적						
최저온도 29일 누적						
최저온도 30일 누적						



‘A’같은 경우에는 27일전의 강수량 2일 누적이므로 예를 들어 5월 28일의 27일 전 즉, 5월 1의 강수량 2일 누적이므로 5월1일과 4월 30일의 강수량의 합이 입력 된다. ‘B’와 같은 경우에는 29일전 즉, 4월 30일의 습도 30일 누적이므로 4월 1일부터 4월 30일의 습도를 전부 더한 값이 입력 된다. 이렇게 9400개의 데이터가 존재한다.



- 데이터의 정규화

입력 데이터의 정규화[7]는 입력 데이터의 범위를 모두 같게 만드는 것을 말한다. 각 기상 요소별로 [Table 3-4]의 수식으로 0~1 사이로 정규화를 진행한다. 최고 온도, 평균 온도, 최저 온도 같은 경우는 하루 동안의 온도 범위를 -15도~45도로 설정하고 +15도씩 Shift를 해주어 범위를 0~60으로 수정했고 0~1로 범위를 맞춰주기 위해 60으로 나눴다. 0보다 작으면 0, 60보다 크면 1이다. 강수량은 하루에 최고로 올 수 있는 강수량을 300mm로 설정했다.

[Table 3-4 각 기상 요소 별 정규화 변환 수식]

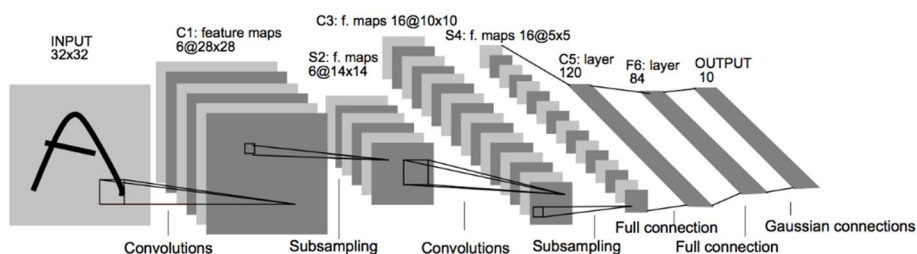
	정규화 변환 수식
평균 습도	습도 누적 / 일 수 / 100
강수량	강수량 누적 / 일 수 / 300
강수일	강수 일 누적 / 일 수
평균 온도	(온도 누적/일 수 + 15) / 60
최고 온도	(온도 누적/일 수 + 15) / 60
최저온도	(온도 누적/일 수 + 15) / 60

## 3.2 신경망 모델 설계

- CNN (Google Inception)

CNN (Convolutional Neural Network)[8][Figure3-1]는 딥 러닝 대표적인 알고리즘 중 하나로 이미지의 지역 정보와 인접성을 이용해 이미지 인식에 탁월한 성능을 발휘하고 있다. 일반적으로 특정 크기의 Filter를 이미지에 적용해 특징을 추출한다. 그 후, Subsampling 을 통해 수 많은 Feature를 줄여주는 과정을 거친다. 마지막으로 나온 Feature들로 분류를 하게 된다. CNN은 이와 같이 3개의 층으로 나뉘어진다.

1. Convolution layer : Convolution feature를 추출하는 layer로 의미 있는 특징들을 추출하는 과정이다.
2. Pooling Layer : 이미지의 특성 상 많은 pixel이 존재하기 때문에 feature를 줄여주기 위해 subsampling을 하는 층이다.
3. Feedforward layer : convolution layer와 pooling layer를 거쳐 나온 feature를 이용해 분류를 하는 층이다.



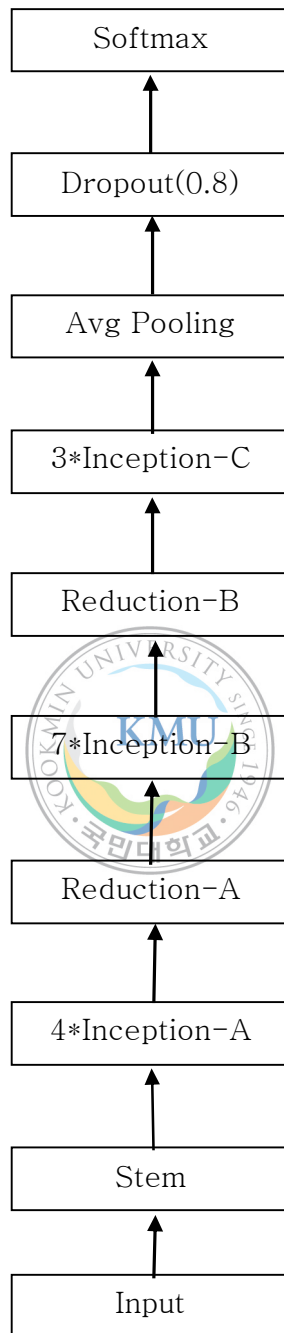
[Figure 3-1 Convolutional Neural Network 과정]

CNN에서도 많은 알고리즘들이 존재한다. 그 중 Google Inception모델 [9][10][11]은 2014년 ILSVRC에서 VGG모델을 근소한 차이로 누르고



1등을 차지한 모델이다. 이 후 3번에 걸쳐 Inception모델을 수정한 여러 논문들을 Google에서 발표했다. 본 논문에서는 Inception V4 모델을 사용한다. Google Inception V4는 [Figure 3-2]처럼 신경망의 구조가 구성되어있다.

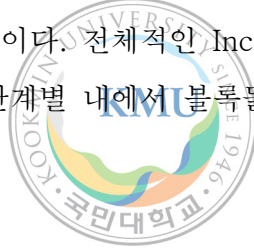




[Figure3-2 Google Inception-V4 신경망의 구조]

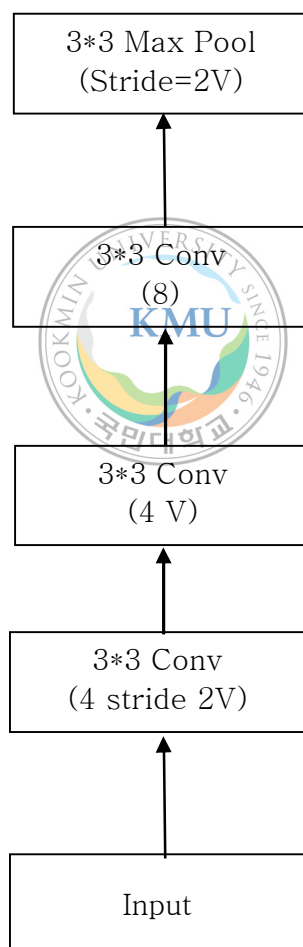
Input 데이터가 들어오게 되면 순서대로 Stem과정을 거쳐 4번의 Inception-A과정의 반복, Reduction-A과정, 7번의 Inception-B과정의 반복, Reduction-B 과정, 3번의 Inception-C과정의 반복을 하고, 그 다음 Average Pooling을 하게 되고 Softmax를 거쳐 최종 분류를 하게 된다. Google Inception-V4망은 여러 과정이 반복되는 만큼 넓고 깊은 망이기 때문에 Vanishing Gradient[12] 문제가 발생한다. 이를 해결하기 위해 중간에 Reduction-B과정을 마치고 Softmax 층을 추가 시켜주었다. 최종 Cost값의 설정을 위해 직관적으로 여러가지 조합을 시도했다.

Reduction-B과정 후에 Cost값의 0.3을 곱하고 마지막 Cost값의 0.7을 곱한 값이 가장 성능이 좋았다. 실질적으로 Feature가 줄어드는 과정은 모두 Reduction 과정에서 일어나게 된다. Stem과정과 각 Inception과정은 특징을 추출하는 부분이다. 전체적인 Inception망 구조는 참고하지만 filter 수를 줄이고 각 단계별 내에서 블록들을 조정했다.



## -Stem 과정

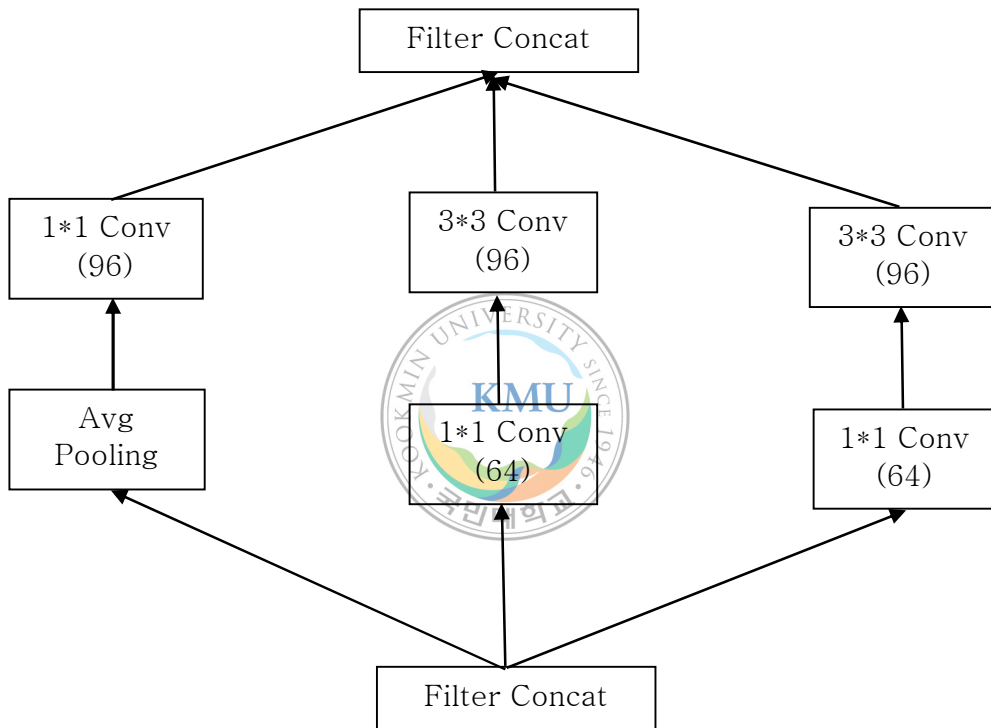
Stem과정은[Figure3-3]과 같이 구성되어 있다. 기본적으로 9개의 layer를 거쳐 Feature-map을 만들어낸다. Stem도 Feature를 추출하는 과정으로 Inception 과정과 비슷하지만 Input과 가까운 부분에서 Inception 과정은 효과가 없기 때문에 Stem과정으로 먼저 Feature를 추출해준다.



[Figure3-3 Stem 구조]

## -Inception-A

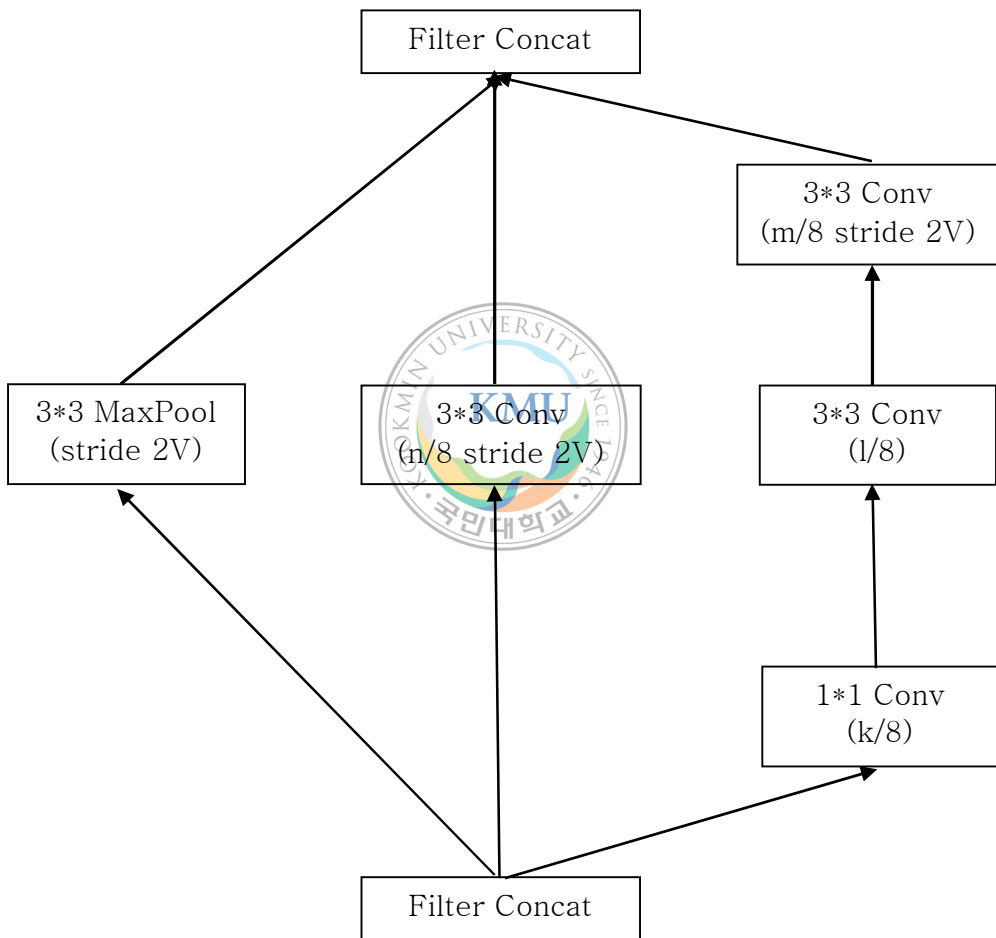
Stem 과정을 거친 후 Inception-A 과정이 4번 반복이 된다. Inception 과정은 특징을 추출하는 과정이다. 구조는 [Figure3-4]와 같다. Inception 과정에서는 Feature의 크기는 변함이 없다.



[Figure3-4 Inception-A 구조]

## -Reduction-A

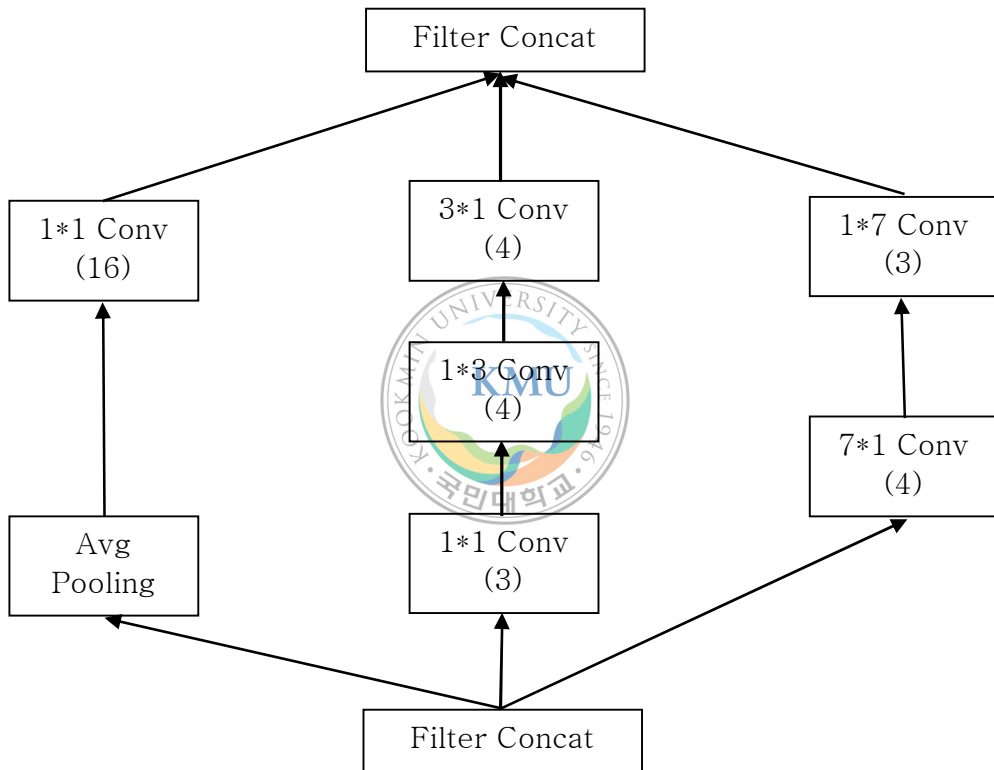
Inception-A를 4번 반복 후 Reduction-A과정을 거친다. Reduction-A과정 [Figure 3-5]에서 Feature의 크기가 줄어든다. Reduction-A과정에서  $k=192$ ,  $l=224$ ,  $m=256$ ,  $n=384$ 의 feature 개수를 의미한다.



[Figure 3-5 Reduction-A 구조]

## -Inception-B

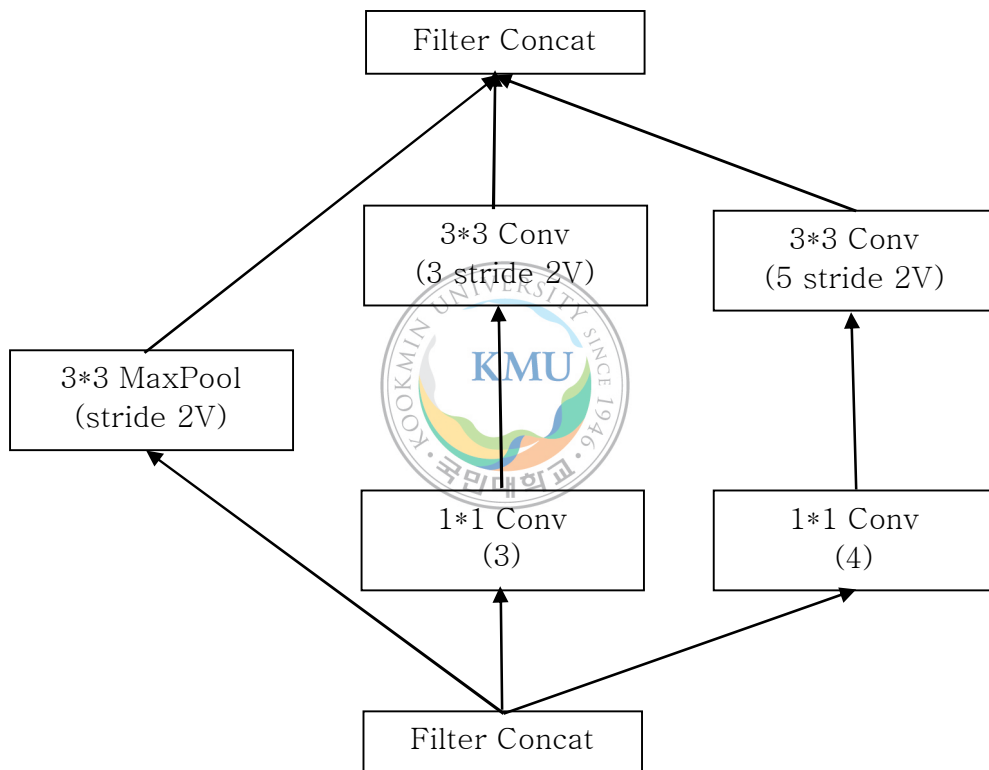
Feature의 크기가 줄어드는 Reduction-A의 과정을 거친 후 Inception-B의 과정을 7번 반복한다. Inception-B과정도 Inception-A과정과 동일하게 Input과 Output크기의 변화가 없다. 구조는 [Figure3-6]과 같다.



[Figure 3-6 Inception-B 구조]

## -Reduction-B

Inception-B 과정을 7번 거치고 난 후 다시 Reduction-B의 과정을 거치게 된다. Reduction-B의 과정도 Reduction-A과정과 동일하게 Feature의 크기가 줄어든다. 구조는 [Figure 3-7]과 같다.

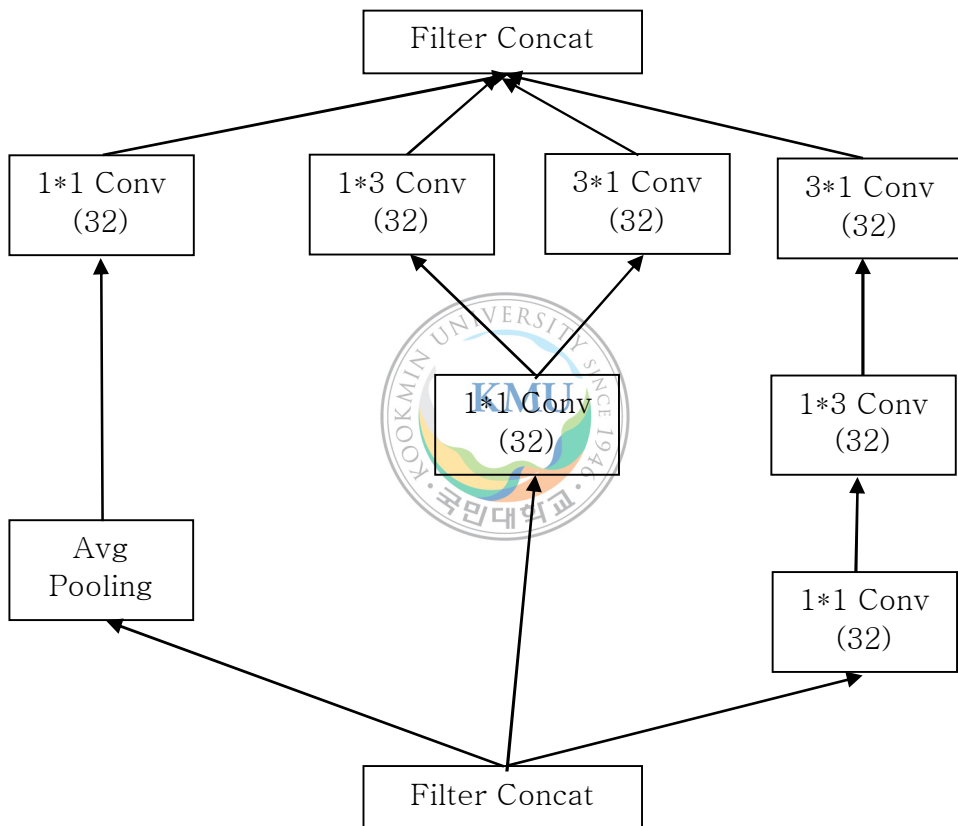


[Figure3-7 Reduction-B 구조]



## -Inception-C

Reduction-B의 과정 후에 Inception-C의 과정을 3번 반복한다. 이 과정을 마지막으로 Inception 과정은 끝이 난다. 구조는 [Figure3-8]과 같다.



[Figure3-8 Inception-C 구조]

Inception-C의 망을 3번 거치고 난 후에 Dropout[13]을 적용한다. Dropout은 신경망의 overfitting[14]을 방지하기 위한 기법이다. 본 논문에서는 0.8로 설정했다. 즉, layer에 포함 된 weight 중 80%만 학습에 쓰인다. 최종적으로 softmax를 거쳐 분류를 하게 된다.

### 3.3 모델을 통한 학습 및 결과

- 기존 데이터의 학습

학습 데이터로 [Table3-3]의 csv데이터를 사용했다. Test 데이터의 예측 결과가 [Table3-5]처럼 한 쪽으로 편향(1단계)되었다. [Table 3-6]과 같이 Filter가 학습을 하는 과정에서 평균 습도 30일과 강수량 1일이 섞이게 된다. 이처럼 각 기상<요소별 경계면에서 섞이기 때문에 기상 요소의 특성을 강하게 나타내지 못하기 때문이라고 예상했다.

[Table3-5 예측 결과의 쏠림 현상]

날짜	예측 결과	정답
06-01	1	2
06-02	1	4
06-03	1	3
06-04	1	7

[Table3-6 섞이는 부분의 예시(테두리 부분)]

평균 습도 30일	0.5055	0.5085	0.5086
강수량 1일	0	0	0
강수량2일			
...			
강수량30일	0.2456	0.43	0.32
강수일1일	0	0	0
강수일2일			
...			
강수일30일	0.7	0.624	0.135
평균온도1일	0.42	0.66	0.789
평균온도2일			
...			
평균온도30일	0.23	0.32	0.278
최고온도1일	0.3	0.44	0.31
최고온도2일			

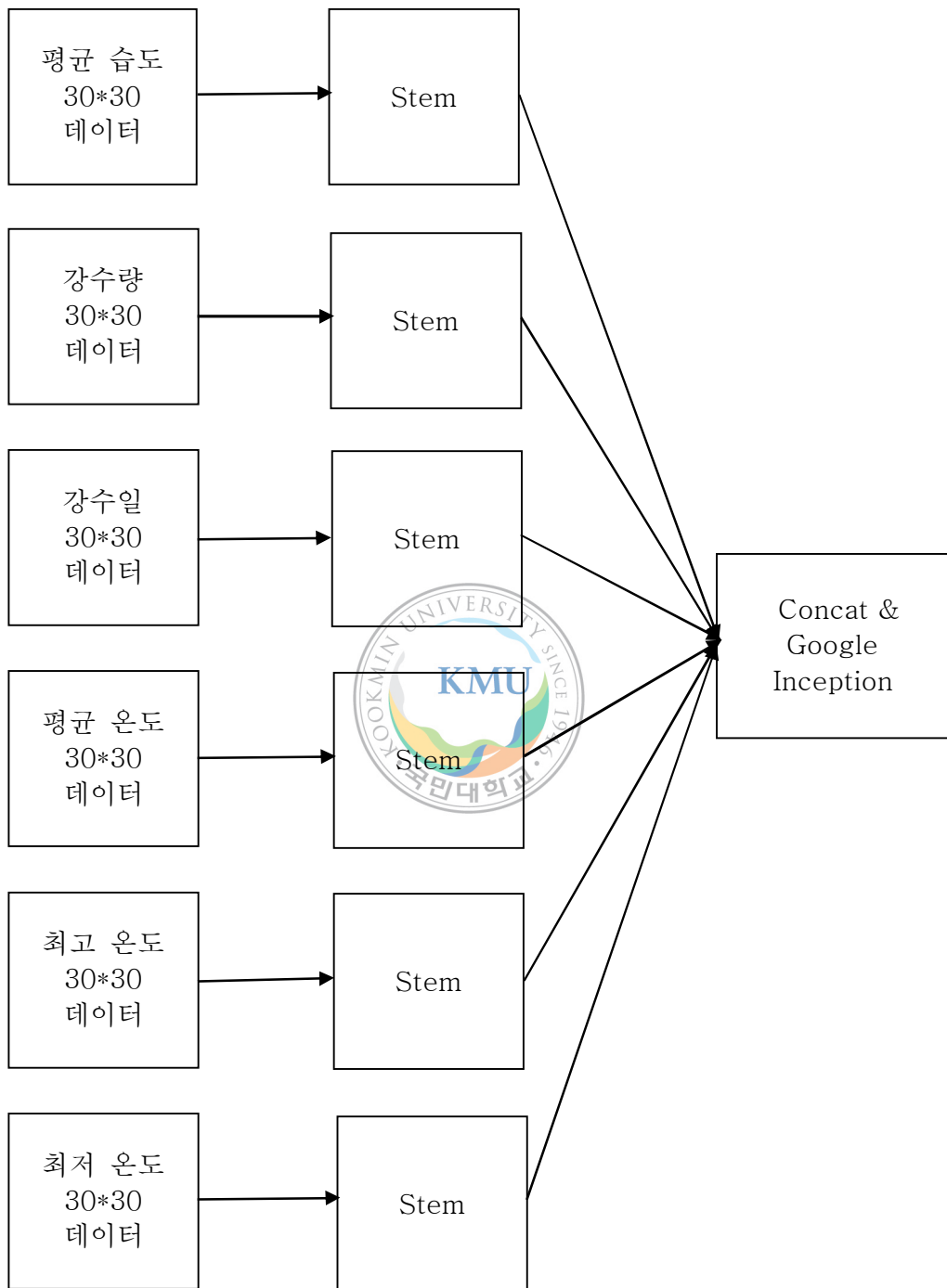
각 기상 요소 별 경계 면에서  
전혀 다른 속성이 섞이면서  
학습이 된다.

- 기상 요소 분리 후 학습

각 기상 요소 경계면 마다 섞이는 것을 피하기 위해 학습 망을 통해 기상 요소를 분리하는 전처리를 한다. 데이터를 평균 습도, 강수량, 강수 일, 평균 온도, 최고 온도, 최저 온도 각각의 데이터(30\*30)를 stem과정을 적용해주고, 적용해 준 결과 나온 Tensor 6개를 붙여준 다음 Google-Inception-V4과정 그대로 진행한다. 이 실험은 [Table3-7처럼] 예측 단계가 한 쪽으로(1단계) 쏠리는 현상은 피했다. 하지만 정확도가 47.8%로 기존 연구 결과인 50.94% 보다 낮았다. 과정은 [Figure3-9]과 같다.

[Table3-7 기상 요소 분리 후 예측 결과]

날짜	예측 결과	정답
06-01	2	2
06-02	4	4
06-03	1	3
06-04	1	7



[Figure3-9 데이터 분리 후 학습 과정 구조]

### 3.4 데이터의 전처리 방안

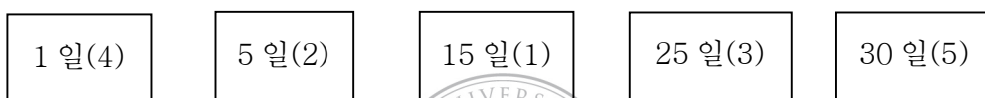
기상 데이터의 인접성 향상을 위한 전처리를 진행한다. 신경망의 Filter 사이즈를 바꾸는 것, 혹은 Pooling 방법이나 Pooling의 Filter 사이즈를 바꾸는 것도 방법이 될 수 있다. 하지만 이 Hyperparameter tuning과정에서 결국 데이터는 고정되어 있어 여전히 각 기상 요소 별로 독립적으로 특성이 강하게 나타날 수 있다. 따라서 기상 데이터를 전처리 한다. 기상 데이터의 요소별 조합을 여러 가지로 구성하면서 인접성이 가장 뛰어난 조합을 찾아 데이터의 구조를 변경한다.



## 제 4장 전처리 과정을 통한 실험

### 4.1 누적 기상 요소 값 선택

3.4장에서 제시한 데이터의 전처리 방안을 적용한다. 기존 2차원 csv 데이터는 각 기상 요소 6가지의 1일 누적부터 30일 누적까지 총 180개의 행이 존재한다. 각 기상 요소 1가지는 30개의 행을 차지한다. 여기서 각 요소별로 특정 누적 값을 선택한다. 누적 값 선택은 총 2가지 방법으로 진행한다. 첫번째는 [Figure4-1]처럼 규칙적으로 추출하는 것이다.



[Figure 4-1 규칙적 누적 값 추출]

규칙적으로 누적 값을 추출하는 방법은 첫번째로 1과 30의 중간 값인 15일, 15일 기준으로 -10, +10을 한 5일과 25일, 양 끝 값인 1, 30일을 추출했다. 괄호 안에 값이 순서이다. 추가로 Random으로 추출하는 방법도 진행을 했다. Random으로 추출하는 것은 기존 서론에서 언급했던 Random Forest로 연구한 기상 요소 누적 값들을 포함하고 추가로 임의로 누적 값들을 뽑았다. 뽑은 누적 값들은 [Table 4-1]과 같다.

[Table 4-1 누적 값 추출]

	각 기상 요소별 누적 값 1개		각 기상 요소별 누적 값 2개		각 기상 요소별 누적 값 3개		각 기상 요소별 누적 값 4개	
Random Forest로 추출한 기상요소 별 누적 값 포함	평균 습도	5	평균 습도	5,9	평균 습도	5,9,13	평균 습도	5,9,13,17
	강수량	6	강수량	6,22	강수량	6,22,30	강수량	1,6,22,30
	강수일	29	강수일	22,29	강수일	5,22,29	강수일	1,5,22,29
	평균 온도	16	평균 온도	16,20	평균 온도	16,20,28	평균 온도	8,16,20,28
	최고 온도	19	최고 온도	15,19	최고 온도	5,15,19	최고 온도	1,5,15,19
	최저 온도	2	최저 온도	2,8	최저 온도	2,8,20	최저 온도	2,8,20,25
Random Forest로 추출한 기상요소 별 누적 값 포함	평균 습도	15	평균 습도	5,15	평균 습도	5,15,25	평균 습도	5,15,25,1
	강수량	15	강수량	5,15	강수량	5,15,25	강수량	5,15,25,1
	강수일	15	강수일	5,15	강수일	5,15,25	강수일	5,15,25,1
	평균 온도	15	평균 온도	5,15	평균 온도	5,15,25	평균 온도	5,15,25,1
	최고 온도	15	최고 온도	5,15	최고 온도	5,15,25	최고 온도	5,15,25,1
	최저 온도	15	최저 온도	5,15	최저 온도	5,15,25	최저 온도	5,15,25,1



## 4.2 인접성 향상을 위한 전처리

4.1에서 추출한 누적 값 들을 인접성 향상을 위한 전처리를 진행한다. 인접성 향상을 위해서 각 기상 요소 별 누적 값들을 추출한다. 각 기상 요소 별(평균 습도, 강수량, 강수 일 수, 평균 온도, 최고 온도, 최저 온도)로 누적 값을  $n$ 개 추출하면 총 누적 값의 개수는  $6*n$ 개이다. 예를 들어 각 기상 요소 별로 3가지를 뽑으면 총 18가지의 누적 값들이 추출된다. 그 중 3개의 기상 요소를 인접하게 둘 경우, 행의 개수는  $3*18 C 3$ 이고 [Table4-2]처럼 데이터의 구조가 바뀌게 된다.

이전 연구에서 데이터의 구조는 각 기상 요소 별로 평균 습도, 강수량, 강수 일 수, 평균 온도, 최고 온도, 최저 온도가 차례대로인  $180*30$ 의 2차원 Matrix 형태의 구조였던 반면, 위와 같이 전처리를 할 경우, 기상 요소가 다양하게 서로 인접하게 된다.



습도 5, 습도 15, 습도 25, 강수량 5, 강수량 15, 강수량 25, 강수일 5, 강수일 15, 강수일 25, 평균 온도 5, 평균 온도 15, 평균 온도 25, 최고 온도 5, 최고 온도 15, 최고 온도 25, 최저 온도 5, 최저온도 15, 최저온도 25

[Figure4-2 18개의 기상 요소 누적 값]

(습도 5, 습도 15), (습도 15, 강수량 5), (습도 15, 강수량 15), (습도 15, 강수량 25), (습도 5, 강수일 5), (습도 5, 강수일 15), ..... , (습도 15, 습도 25), (습도 25, 최저 온도 25), (강수량 5, 강수량 15), (습도 25, 강수량 25),....., (최저 온도 5, 최저 온도 15), (최저 온도 15, 최저 온도 25), (최저 온도 15, 최고 온도 5), (최저 온도 15, 최고 온도 15), (최저 온도 15, 최고 온도 25), (평균 온도 15, 최고 온도 5), (평균 온도 5, 최고 온도 5), (평균 온도 15, 최고 온도 15),....., (최저 온도 15, 최저 온도 25)

[Figure4-3 18개의 누적 값들 중 인접 개수 2개의 예시]

(습도 5, 습도 15, 습도 25), (습도 5, 습도 15, 강수량 5), (습도 5, 습도 15, 강수량 15), (습도 5, 습도 15, 강수량 25), (습도 5, 습도 15, 강수일 5), (습도 5, 습도 15, 강수일 15), ..... , (습도 15, 습도 25, 최저온도 15), (습도 15, 습도 25, 최저 온도 25), (습도 25, 강수량 5, 강수량 15), (습도 25, 강수량 5, 강수량 25),....., (최저 온도 5, 최저 온도 15, 최고 온도 5), (최저 온도 15, 최저 온도 25, 최고 온도 15), (최고 온도 25, 최저 온도 5, 최저 온도 15),....., (최저 온도 5, 최저 온도 15, 최저 온도 25)

[Figure4-4 18개의 누적 값들 중 인접 개수 3개의 예시]

[Table 4-2 18 C 3 행의 예시]

평균 습도 5				
평균 습도15				
강수량 15				
평균 습도 5				
평균 습도15				
강수량 25				
...				
평균 습도 5				
평균 온도15				
최고 온도25				
강수량25				
평균 습도 5				
최고 온도15				

각 기상 요소 별로 [누적 5일, 누적 15일, 누적 25일] 총 18가지를 추출 한 후 인접 기상 요소 3개로 행을 구성하여 데이터를 전처리 한 구조

[Table 4-3 18 C 2 행의 예시]

평균 습도 5				
평균 습도15				
평균 습도 5				
최고 온도15				
...				
평균 습도 5				
평균 온도15				
평균 습도5				
강수량 5				
평균온도 5				
최고온도 15				
평균온도 25				
최고온도 5				
최고온도 5				
최저온도 15				

각 기상 요소 별로 [누적 5일, 누적 15일, 누적 25일] 총 18가지를 추출 한 후 인접 기상 요소 2개로 행을 구성하여 데이터를 전처리 한 구조이다.

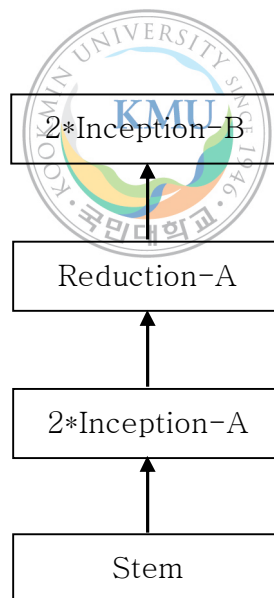
위의 [Table4-2]와 [Table 4-3]은 [Figure 4-3], [Figure 4-4]의 순서쌍을 차례대로 나열해서 생성한 데이터 구조다.



### 4.3 신경망 학습 및 수치 비교

- 신경망의 간소화

Google Inception-V4망은 매우 깊고 넓은 망이다. 본 실험에서는 데이터의 용량이 매우 크므로 실험 시간도 매우 많은 시간이 소요된다. 따라서  $r$ 의 값을 순차적으로 증가할 때  $r$ -squared score의 추세를 보기 위해 backbone 신경망은 Google Inception-V4망을 그대로 쓰지만 간소화했다. Google Inception-V4의 전체 신경망 중 Reduction-A과정까지 진행해 주었다. Inception-A 과정은 기존 4번 반복에서 2번 반복으로 감소시켜주었다. 학습 시 최적화 함수는 AdamOptimizer[15]를 사용했고 learning rate는 0.0001로 설정했다.



[Figure 4-5 Google Inception 신경망의 간소화]

- r-squared score의 비교

r-squared score는 결정 계수라고 하며 0~1사이의 범위를 가진다. 추정한 예측 모델이 자료에 적합한 정도를 재는 척도이다.[16] r-squared score값은 1에 가까울수록 예측 모델이 신뢰성이 높다는 것을 의미하고 0에 신뢰성이 낮다는 것을 의미한다. r-squared score값은 scikit learn 라이브러리에 정의가 되어있다. 본 논문에서 Test 모델을 통한 예측 데이터와 정답 데이터 총 2800개를 비교해 r-squared score를 구한다. 방법은 아래[Algorithm 1]와 같다.

---

**Algorithm 1** r-squared score를 구하는 방법

---

```
1: from sklearn.metric import r2score
2: y_true=[2,4,5,3,7,...] <- mosquito label data
3: y_prediction=[2,3,4,5,6,...] <- mosquito prediction data
4: print(r2_score(y_true, y_prediction))
```

---

[Table 4-1]에 제시된 2가지 방법으로 실험을 진행한다. 각 기상 요소별로 누적 값을 1가지를 추출하면 총 누적 값은 6개, 2가지를 추출하면 총 12개, 3가지를 추출하면 총 18개, 4가지를 추출하면 총 24개가 된다. 기존 선행 연구가 포함되고 임의로 추출 한 것을 Random Feature, 기존 선행연구가 포함 되지 않고 규칙적으로 추출 한 것을 Regular Feature라고 한다. 총 기상 요소 개수 6, 12, 18, 24가지는  $n C r$ 에서  $n$ 에 해당 된다. 몇 개의 기상 요소를 인접하게 설정할 것인지에 대한 값은  $r$ 에 해당 된다.  $r$ 의 값을 순차적으로 증가 시켜 그래프를 그려본다. 아래 [Table4-4]과 [Table4-5]는 위에서 언급한 Random Feature와 Regular Feature의 r-squared score표이다. 표의 가로는 총 누적 값들의 개수( $n$ ), 세로는 누적 값들의 인접 개수( $r$ ) 이다.

	6	12	18	24	30	36
1	0.19	0.21	0.22	0.227	0.233	0.237
2	0.25	0.259	0.261	0.27	0.278	0.284
3	0.271	0.28	0.287	0.298	0.302	0.305
4	0.288	0.295	0.307	0.318	0.322	0.325
5	0.295	0.304	0.321	0.331	0.336	0.338
6		0.309	0.33	0.34	0.3455	0.347
7		0.311	0.335	0.3465	0.3518	0.353
8		0.3118	0.337	0.3478	0.357	0.358
9		0.3121	0.3378	0.3503	0.361	0.3622

[Table 4-4 Regular Feature의 r-squared score]

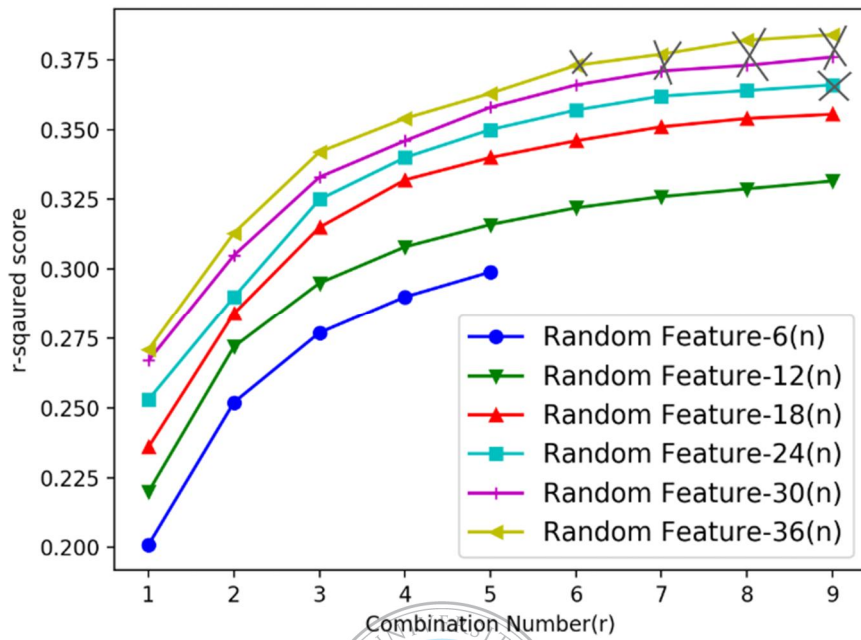
	6	12	18	24	30	36
1	0.201	0.22	0.236	0.253	0.267	0.271
2	0.252	0.272	0.284	0.29	0.308	0.315
3	0.277	0.295	0.313	0.325	0.336	0.345
4	0.29	0.308	0.328	0.346	0.35	0.358
5	0.299	0.316	0.335	0.358	0.362	0.366
6		0.322	0.346	0.364	0.371	0.375
7		0.326	0.351	0.369	0.378	0.381
8		0.3288	0.354	0.372	0.382	0.384
9		0.3316	0.3575	0.3742	0.3835	0.383

[Table 4-5 Random Feature의 r-squared score]

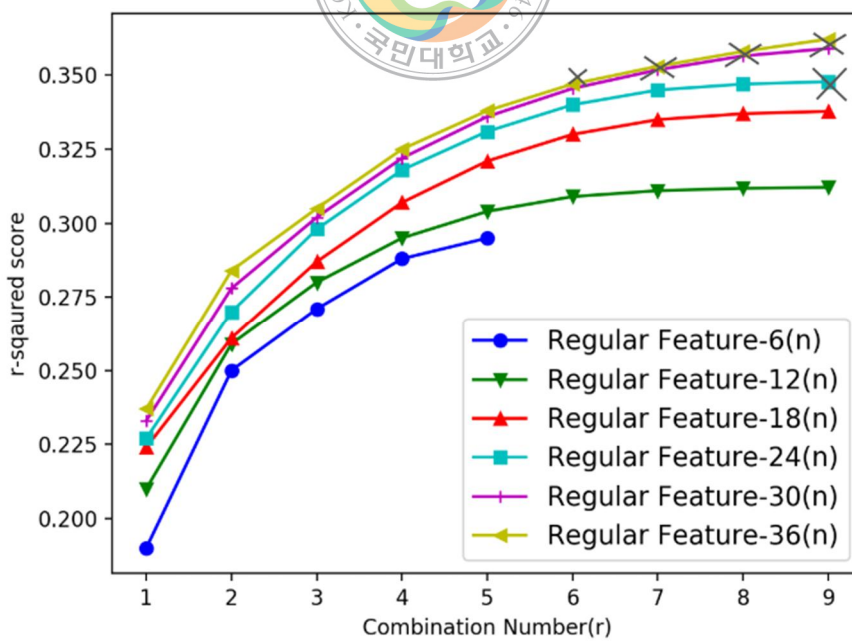


다음은 [Table 4-4]과 [Table 4-5]에 해당하는 그래프를 그린다.  
[Figure4-6]는 [Table4-5]의 해당 되고 [Figure4-7]는 [Table4-4]에  
해당 된다. 총 누적 값의 개수가 증가 할수록 r-squared score는 증가하  
면서 수렴하고 있고 인접 기상 요소 개수 또한 증가 할수록 r-squared  
score가 증가하면서 수렴하고 있다. 24C9, 30C8, 30C9, 36C7, 36C8,  
36C9는 메모리 에러가 발생해 실험을 하지 못하였다.





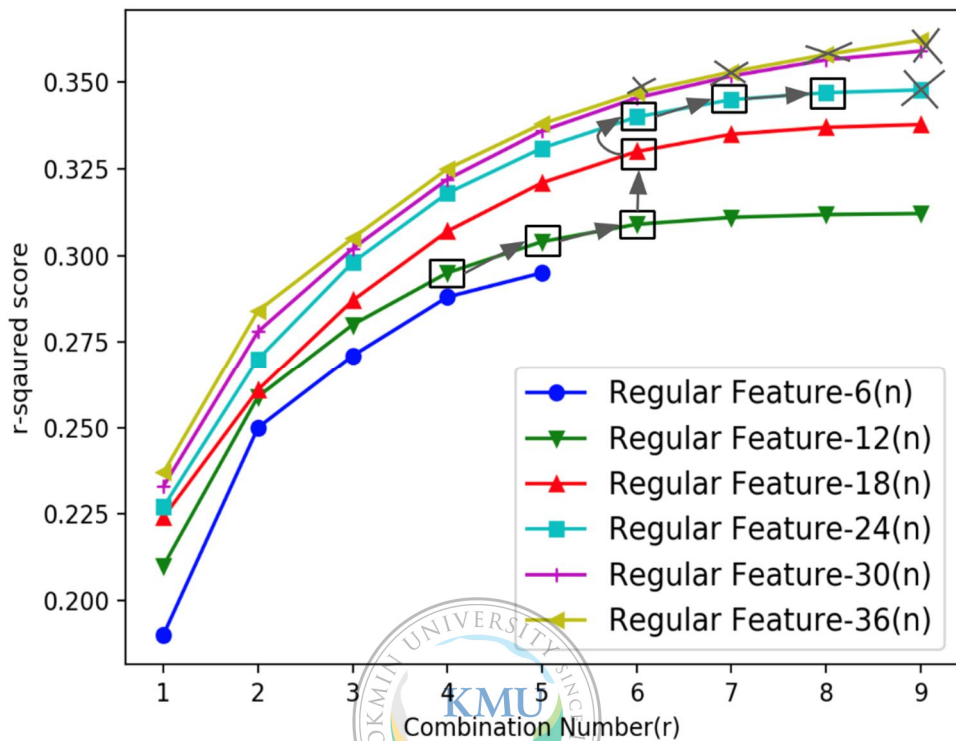
[Figure4-6 Random Feature r-squared score값 비교]



[Figure4-7 Regular Feature r-squared score값 비교]

[Figure4-6]와 [Figure4-7]을 보면 Random Feature와 Regular Feature 모두  $r$ 의 값을 증가 시켜줄수록  $r$ -squared score가 급격히 증가하다가 수렴하는 양상을 띄는데 인접 누적 값의 개수가 1 → 2, 2 → 3, 3 → 4, 4 → 5, 5 → 6으로 증가할 때 마다 증가 폭이 작아진다. 또한 2개의 그래프 모두 비슷한 양상을 띄고 있다. 이 부분에서 전문가의 의견을 반영한 것과 그렇지 않은 것이 서로 차이가 없다는 것을 알 수 있다. 이 과정에서  $r$ -squared score의 증가 폭을 threshold value로 적절하게 설정해 [Figure4-8]처럼  $n$  값과  $r$  값을 선택 할 수 있다.

각 그래프에서 적절한  $r$ 을 선택할 수 있는 방법은 [Figure4-8]에서 Elbow Method[17]를 통해 보면 알 수 있다. 총 기상 요소별 누적 값의 개수에서 인접 값의 개수가 늘어날 때  $r$ -squared score의 증가 폭을 확인한다. 총 누적 값이 12인 경우 인접 기상 요소 개수가 5에서 6으로 갈 때  $r$ -squared score의 증가 폭이 2% 미만이다. 따라서 6개의 누적 값이 인접하게 나열 된다. 이 과정에서 행의 개수는  $nC6$ 이 된다. 6을 고정 한 상태에서 총 누적 값을 증가시킨다. 총 누적 값이 12에서 18, 18에서 24로 증가 할 때  $r$ -squared score의 증가 폭을 본다. 18에서 24로 갈 때 2% 미만의 범위에 만족해 24를 선택한다. 따라서 총 누적 값 24개와 인접 기상 요소 개수 6개를 선택할 수 있다. 마지막으로 인접 누적 기상 요소 개수를 6부터 1씩 증가하면서  $r$ -squared score의 증가 폭을 본다. 7에서 8로 갈 때 2% 미만의 조건을 만족해 최종 인접 기상 요소 개수는 8개이다. 본 실험에서는 모든 경우의 수를 실험해 주었지만 조건에 맞는 총 누적 값과 인접 기상 요소 개수 추출을 위해 실험을 전부 할 필요는 없다.



[Figure4-8 총 누적 개수(n)와 인접 기상 요소 개수(r) 선택]

[Figure4-9]는 총 누적 값의 개수와 인접 기상 요소 개수를 추출하는 방법에 대한 수도 코드이다.

```

def choose_n_r(threshold):
    aDay=[6,12,18,24,...,180] ##총 누적 값 개수
    n,r = None,None ##n,r값이 저장될 변수
    for i in range(0,len(aDay)):
        p=None ##n값의 index가 저장 될 변수
        for j in range(2,aDay[i]):
            if((r2score(aDay[i],Cj)-r2score(aDay[j],Cj-1))
                /r2score(aDay[j],Cj-1)<threshold):
                r=j, n=aDay[i], p=i
                ##i값은 고정하고 j값이 증가하면서 r2score의 증 가폭
                이 threshold보다 작으면 j값을 r값으로 선택하고 그때의
                n값과 n값의 index를 저장.
        for k in range(p+1,len(aDay)):
            if((r2score(aDay[k],Cr)-r2score(aDay[k-1],Cr))
                /r2score(aDay[k-1],Cr)<threshold):
                n=aDay[k] ##선택된 r값은 고정하고 k값이 증가하면서
                r2score의 증가폭이 threshold보다 작으면
                aDay[k]를 n값으로 선택
        for q in range(r+1,n):
            if((r2score(n,Cq)-r2score(n,Cq-1))
                /r2score(n,Cq-1) <threshold):
                r=q ##n,r값이 선택된 상태에서 q값이 다시 증가하면
                서 r2score의 증가폭이 threshold보다 작으면 q값
                을 최종 r로 선택한다.
    return n,r

```

[Figure4-9 [Figure4-8]에 대한 수도 코드]

## 제5장 기존 신경망을 통한 학습

### 5.1 선행 연구와의 결과 비교

4장에서 언급한 기상 요소 별 누적 총 개수(n)와 누적 값의 인접 개수(r)를 추출한 후 간소화한 신경망이 아닌 기존 깊고 넓은 Google Inception-V4 망을 통해 실험을 진행한다. 24 C 6, 24 C 7, 24 C 8의 총 3가지 경우를 실험했다. 기존 Random Forest와 이 3가지를 비교한다.

[Table5-1 Google-Inception-V4을 통한 학습 결과]

	Random Forest	Google Inception -V4		
		24 C 6	24 C 7	24 C 8
정확도	50.94%	56.8%	58%	58.8%
r-squared score	0.53	0.55	0.556	0.559

실험 결과, 24 C 6, 24 C 7, 24 C 8의 결과는 기존의 Random Forest 알고리즘을 사용했을 때 보다 정확도 뿐만 아니라 r-squared score에서도 좋은 결과를 보여주고 있다. 그리고 r의 값이 증가 할수록 정확도와 r-squared score모두 더 좋은 결과를 나타낸다.

## 제 6장 결론 및 향후 과제

실험 결과 모기 활동성을 예측하기 위해서 모든 기상 요소가 서로 영향을 주는 것을 알 수 있다. 이전 연구에서는 평균 습도, 강수량, 누적 강수일, 평균 온도, 최저 온도, 최고 온도 각각 서로 간섭이 일어나는 부분을 학습하려고 해 좋지 않은 결과가 발생했다.

각 기상 요소 누적 값의 총 개수가 더 많으면 많을수록 그리고 인접 누적 기상 요소 개수를 더 많이 설정할수록 r-squared score가 높아지는 것을 확인 할 수 있다. 각 기상 요소가 서로 영향을 주는 것으로 예상할 수 있다. CNN(Convolutional Neural Network)은 이미지 형식처럼 지역 정보와 인접성이 강하게 형성되어 있는 데이터에 알맞은 인공지능 알고리즘이다. 하지만 기상 데이터 같은 지역 정보나 인접성이 없는 일반적인 수치 데이터에서 데이터의 전처리 결과에 따라서 충분히 좋은 학습 결과를 얻을 수 있다. 또한 기존 연구의 Random Forest로 추출한 기상 요소가 필요 하지 않았다. 사용자가 임의로 추출해도 큰 성능 차이가 발생하지 않았다.

3가지 경우(24C6, 24C7, 24C8)만 실험을 진행해주었다. 하지만 더 좋은 컴퓨터와 서버가 갖추어 졌을 때, 더 많은 누적 값들을 추출해서 실험을 진행하면 지금 보다 더 좋은 정확도와 r-squared score 수치를 확인할 수 있다.

## 참 고 문 헌

- [1] 기상 정보를 이용한 도시생태정보 생산 기발 기술 개발, 2017, pp.1~129
- [2] 서울시 영등포구 모기 활동성 지수 예보 [Online]. Available: <http://mosq-forecast.cs.kookmin.ac.kr>
- [3] 환경부 미세먼지 예보 서비스 [Online]. Available: <http://www.air-korea.or.kr/dustForest/>
- [4] 국립 전파 연구원 경험적 태양 입자 유입 예측 [Online]. Available: <http://spaceweather.rra.go.kr/models/spef>
- [5] Gehring, Jonas, and Auli, Michael and Grangier, David and Dauphin, Yann N. "A Convolutional Encoder Model for Neural Machine Translation, ArXiv e-prints, 2016
- [6] Gehring, Jonas, and Auli, Michael and Grangier, David and Yarats, Denis and Dauphin, Yann N. "Convolutional Sequence to Sequence Learning", ArXiv e-prints, 2017
- [7] J.sola, J.Sevilla. "Importance of input data normalization for the application of neural networks to complex industrial problems", IEEE Transactions on Nuclear Science, Jun 1997
- [8] Yann Lecun, Leon Bottou, YoshuaBengio, and Patrick Haffner. "Gradient-Based Learning Applied to Document Recognition", In proc. 1998
- [9] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, Alex Alemi. "Inception-v4, Inception-ResNet and the Impact of Residual



Connections on Learning” , ArXiv:1602.07261, 2016

- [10] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, Andrew Rabinovich. “Going Deeper With Convolutions” , arXiv:14-09.4842, 2014
- [11] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, Zbigniew Wojna. “Rethinking The Inception Architecture for Computer Vision.” arXiv:1512.00567, 2015
- [12] Sepp Hochreiter. “The Vanishing Gradient Problem during learning recurrent neural nets and problem solutions” , International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems
- [13] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Suts-  
kver, Ruslan Salakhutdinov. “Dropout: A Simple Way to Prevent  
Neural Networks from Overfitting” , Journal of Machine learning  
Research, June 2014. 1929-1958
- [14] Rich Caruana, Steve Lawrence, Lee Giles, “Overfitting in Neural  
Nets: Backpropagation, Conjugate Gradient, and Early Stopping”  
 , Advances in Neural Information Processing Systems 13, NIPS,  
2000
- [15] Diederik P. Kingma, Jimmy Ba, “Adam: A method for Stochastic Op-  
timization ” , arXiv:1412.6980
- [16] A. Colin Cameron, Frank A.G Windmeijer. “R-squared Measures for  
Count Data Regression Models With Applications to Health Care

utilization” , Journal of Business and Economic Statistics(forthcoming), April 1995

- [17] Purnima Bholowalia, Arvind Kumar. “EBK-Means: A clustering T-  
echnique based on Elbow Method and K-Means in WSN”,  
International Journal of Computer Applications(0975-8887),  
Volume 105-No.9, November 2014

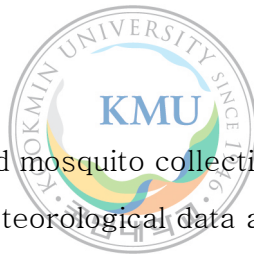


## Abstract

# A Study on Preprocessing for Adjacency Improvement in Weather Data for Predicting Mosquito Activity

*by Lee, Dong Woo*

Department of Computer Science Graduate School,  
Kookmin University, Seoul, Korea



Meteorological data and mosquito collection data are used to predict mosquito activity. The meteorological data and the mosquito collection data are numerical data with no adjacency and local information, and are daily time series data. There are two kinds of deep learning algorithms. RNN (Recurrent Neural Network) algorithm and CNN (Convolutional Neural Network) algorithm. Generally, RNN algorithm is used for learning time series data and CNN algorithm is used for image data learning. The CNN algorithm is useful for extracting local information and locality information by repeatedly applying the filter to the data and extracting features. Recently, however, researches have been actively applied to apply the CNN algorithm to the time series data.

In this paper, we propose a preprocessing process of data that improves the adjacency to obtain good results by using the CNN

algorithm in the time series data handled in the experiment. And we show the performance improvement compared with previous studies.

