2021 BIG CONTEST

댐 유입 수량 예측을 통한 최적의 수량 예측 모형 도출

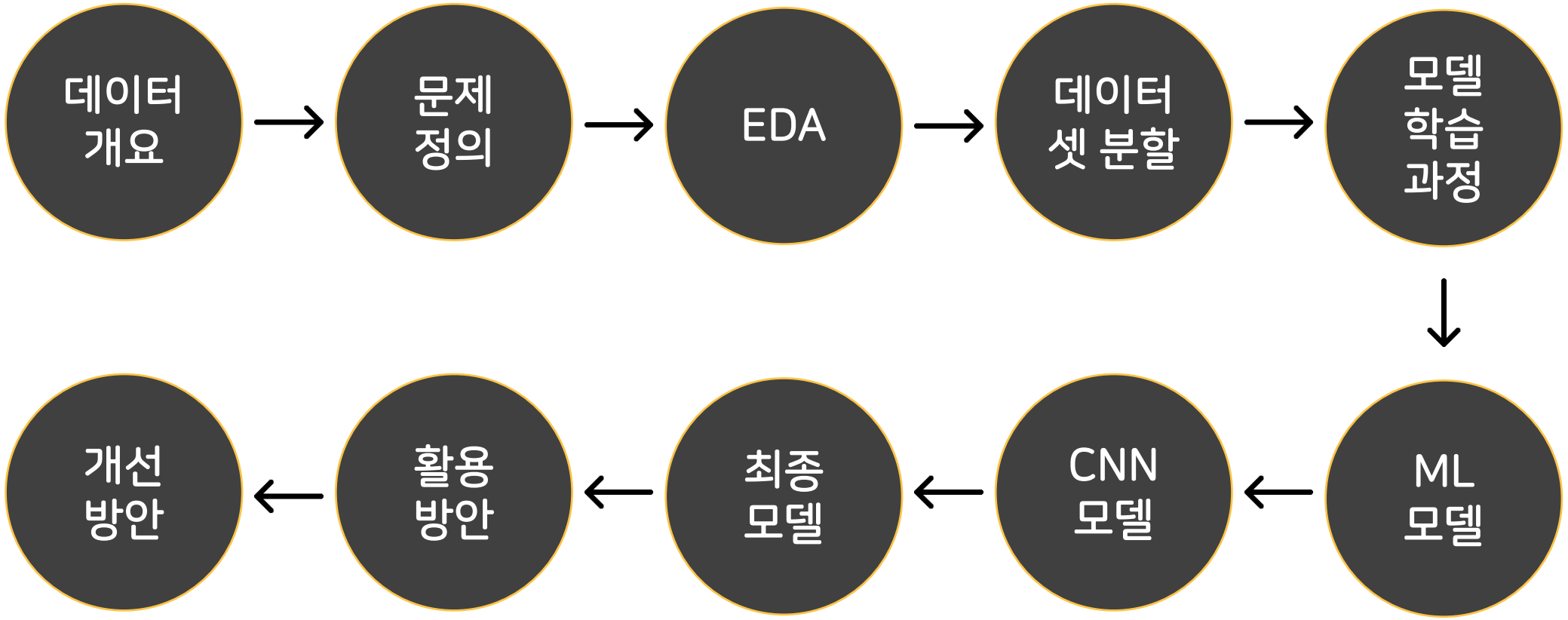
: 데이터 집단과 변수를 효과적으로 학습하는 CNN 모델을 활용한 댐 유입량 예측

팀명 : 범호

이성범 (팀장) : 2712qwer@naver.com

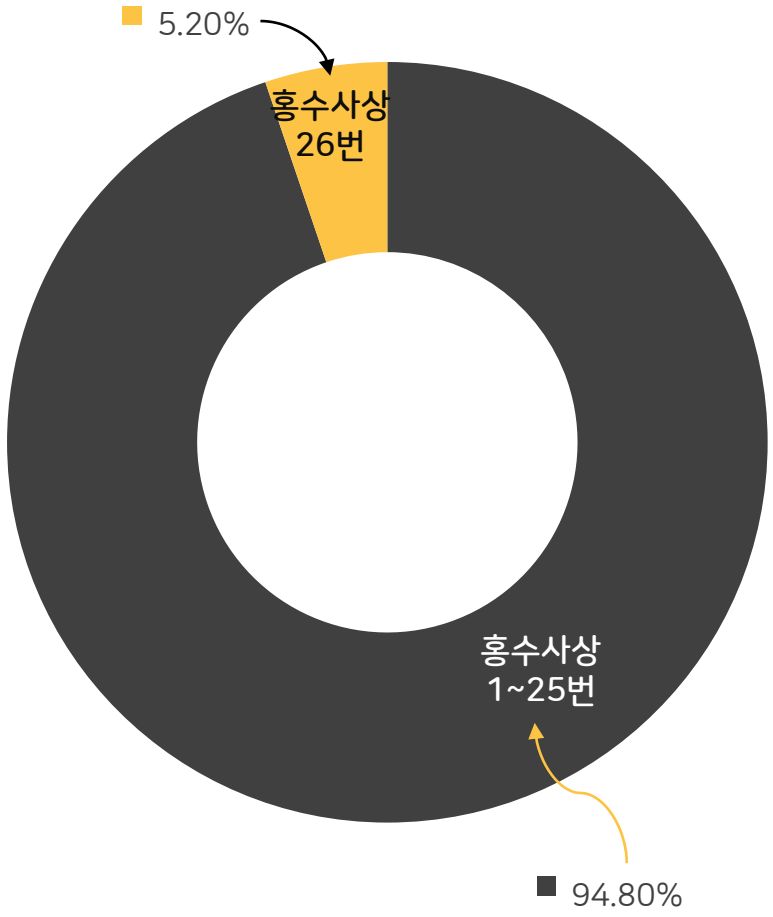
김주호 : jooho991122@gmail.com

목차



01
part

데이터 개요



제공 데이터 특성

① 25개의 홍수사상을 학습하여 26번째 홍수사상을 예측

② 각 홍수사상은 시계열로 이루어진 데이터

③ 홍수사상끼리는 시계열적으로 독립

01
part

데이터 개요

홍수사상번호	연	월	일	시간	유입량	데이터집단 1							데이터집단 2						
						유역평균강수	강우(A지역)	강우(B지역)	강우(C지역)	강우(D지역)	수위(E지역)	수위(D지역)	유역평균강수	강우(A지역)	강우(B지역)	강우(C지역)	강우(D지역)	수위(E지역)	수위(D지역)
1	2006	7	10	8	189.1	6.4	7.0	7.0	7.0	8.0	2.5	122.6	6.3	7.0	7.0	7.0	8.0	2.5	122.5
1	2006	7	10	9	217.0	6.3	7.0	8.0	7.0	8.0	2.5	122.6	6.4	7.0	8.0	7.0	8.0	2.5	122.6
1	2006	7	10	10	251.4	6.4	7.0	9.0	7.0	8.0	2.5	122.6	7.3	7.0	9.0	7.0	8.0	2.5	122.6
1	2006	7	10	11	302.8	7.3	7.0	10.0	7.0	8.0	2.5	122.6	8.2	7.0	10.0	8.0	8.0	2.5	122.6
1	2006	7	10	12	384.8	8.2	7.0	12.0	8.0	10.0	2.5	122.6	11.3	9.0	12.0	10.0	10.0	2.5	122.6
1	2006	7	10	13	512.5	11.3	7.0	14.0	10.0	11.0	2.5	122.6	14.4	12.0	14.0	10.0	11.0	2.5	122.6
1	2006	7	10	14	701.5	14.4	9.0	17.0	10.0	14.0	2.5	122.6	16.9	14.0	17.0	15.0	14.0	2.5	122.6

① K-댐 주변 지역 (A, B, C, D, E)의 유역평균강수, 강우량, 수위 데이터로 이루어짐

② 하나의 Target 값에 독립적인 데이터 집단 6개 존재

02

part

문제 정의

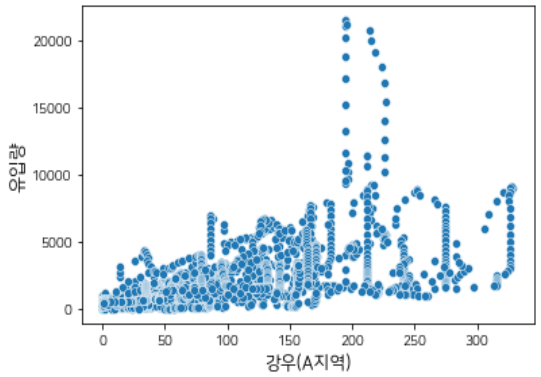
자연현상 속 유입량 예측

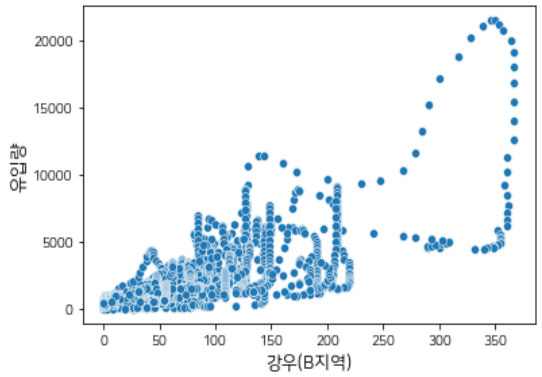
6개의 데이터 집단 활용

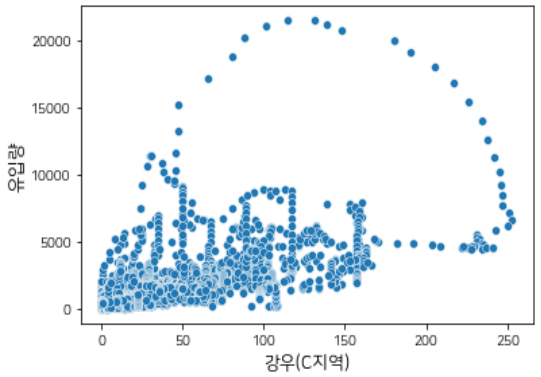
독립 변수간 높은 상관성

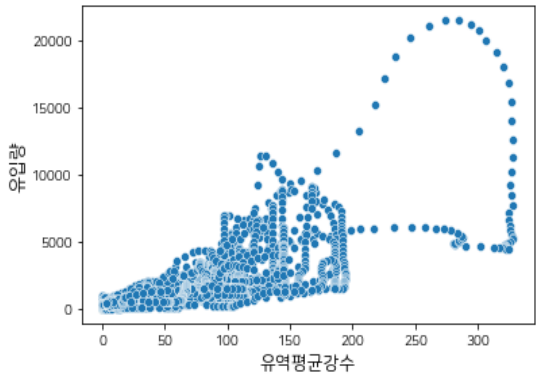
비선형적인 패턴

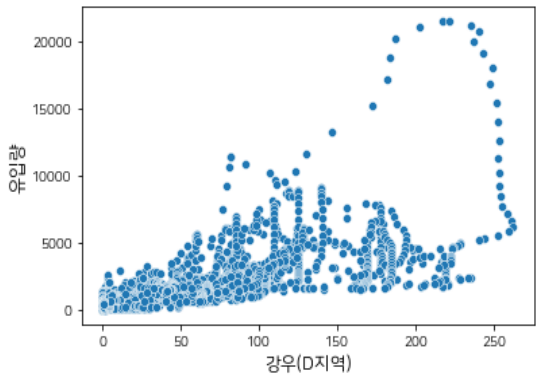
자연적인 변수(노이즈)로 인한 예측의 어려움

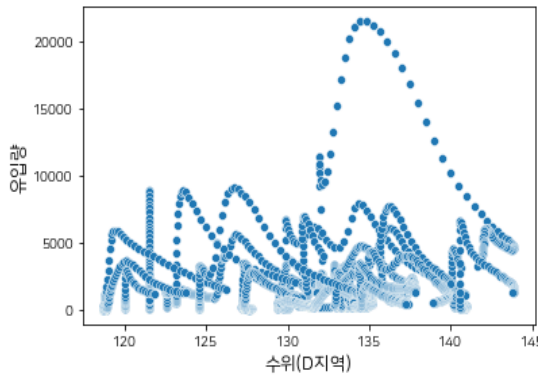


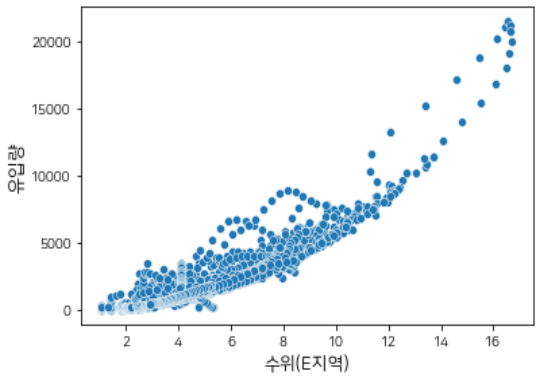












02
part

문제 정의

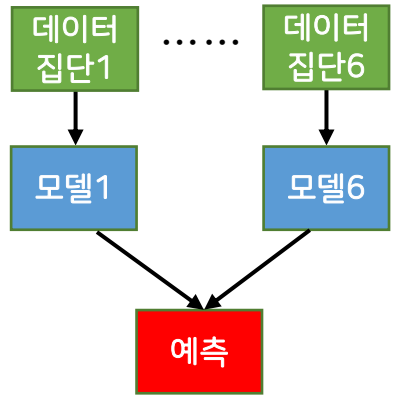
자연현상 속 유입량 예측

6개의 데이터 집단 활용

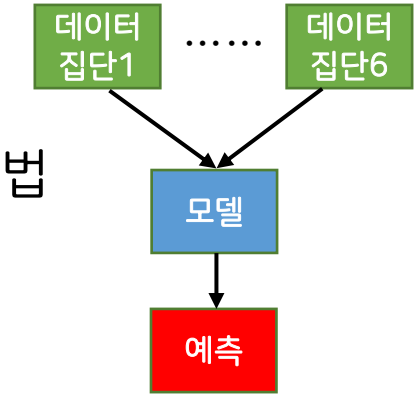
독립 변수간 높은 상관성

관측소와 댐 구간 거리 및 시간을 달리하여 얻은
6개의 데이터 집단의 활용이 본 과제의 핵심

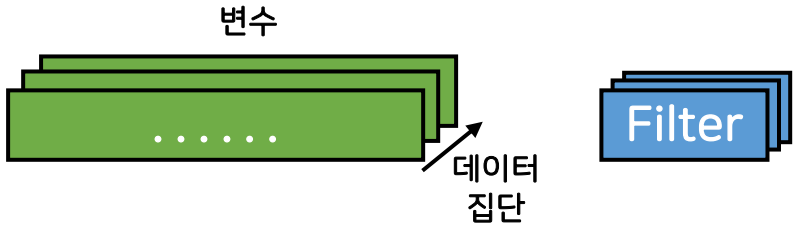
① 모델을 각 데이터 집단으로 학습시켜 6개의 결과값을 Ensemble



② 6개 데이터 집단을 변수로 넣고 학습시켜 예측하는 방법



③ 데이터 집단을 CNN 모델의 채널로 생각하여 결합하는 방법



02

part

문제 정의

자연현상 속 유입량 예측

6개의 데이터 집단 활용

독립 변수간 높은 상관성

다중공선성으로 인한 회귀계수 불안정

다중공선성이란?

➡ 독립 변수간 강한 상관관계가 나타나 회귀계수가 불안정해지는 현상

다중공선성에 따른 문제점:

➡ 회귀계수 추정 및 변수 선택에 왜곡을 불러와 모델 학습이 제대로 이루어지지 않음

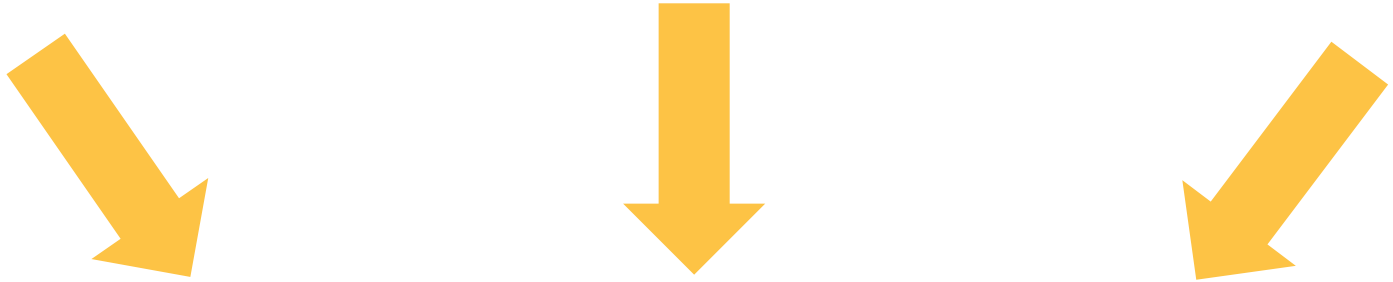
02
part

문제 정의

자연현상 속 유입량 예측

6개의 데이터 집단 활용

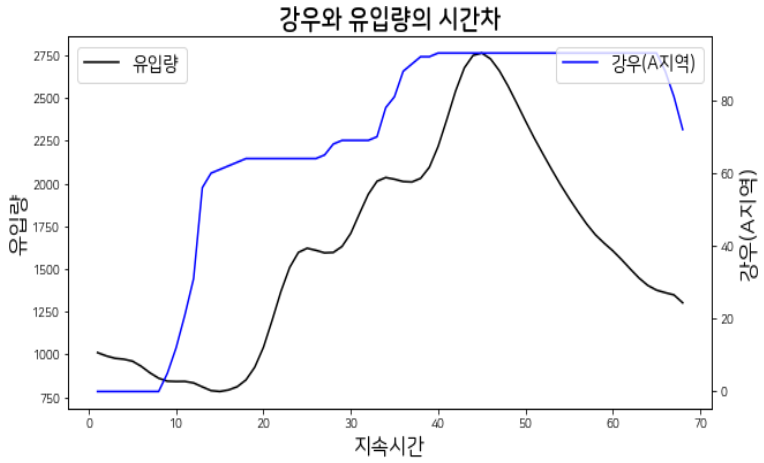
독립 변수간 높은 상관성

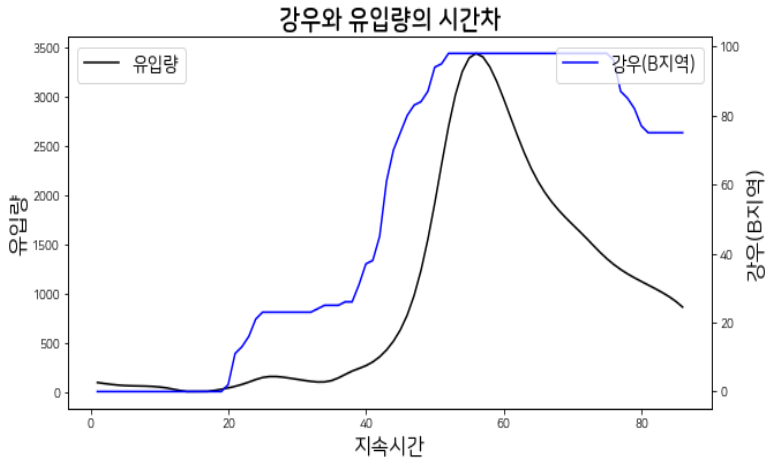


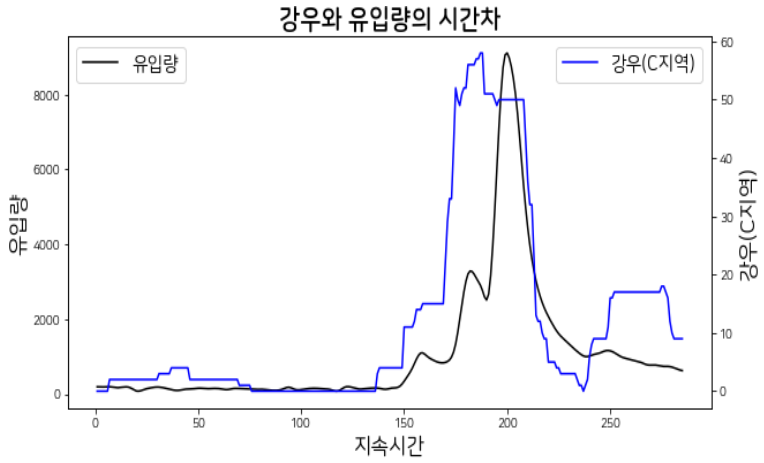
합성곱 신경망 모델
(Convolutional Neural Network)

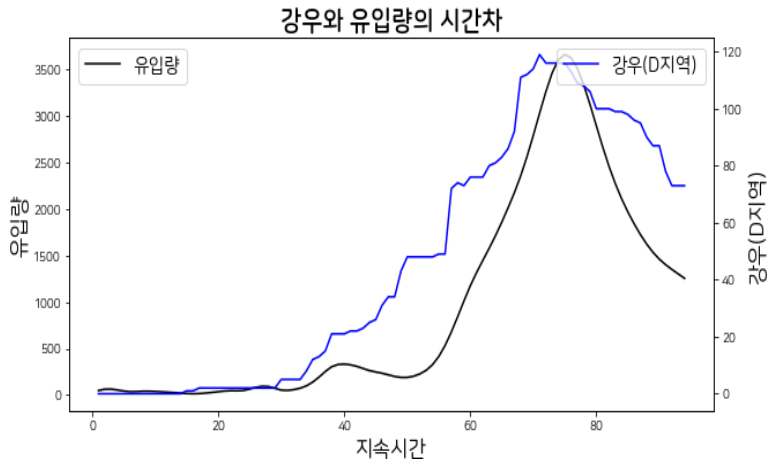
탐색적 데이터 분석 (Exploratory Data Analysis)

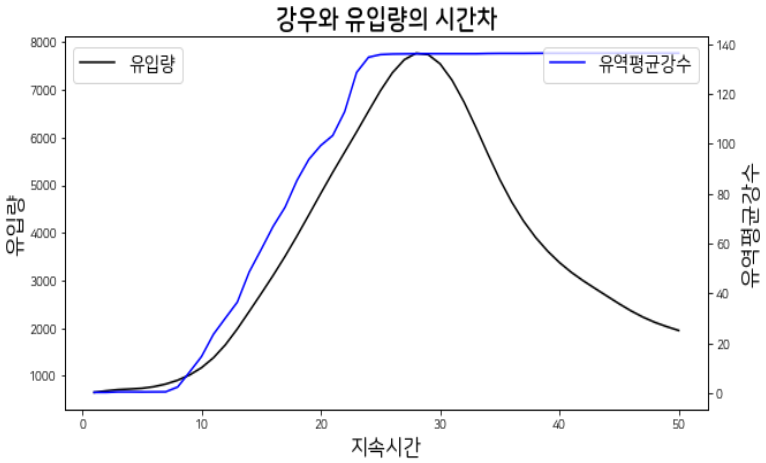
파생 변수 생성











강우 관련 파생 변수 생성

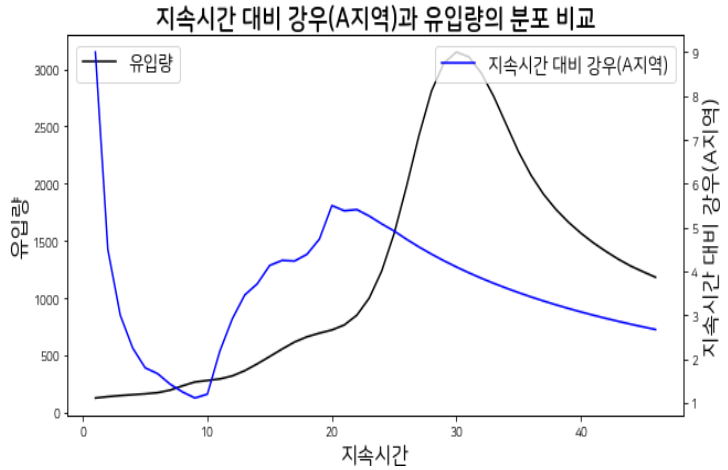
① 과거의 강수는
현재의 유입량에 영향을 줄 것

② 여러 홍수사상에서
강우 데이터 shift의 필요성 확인

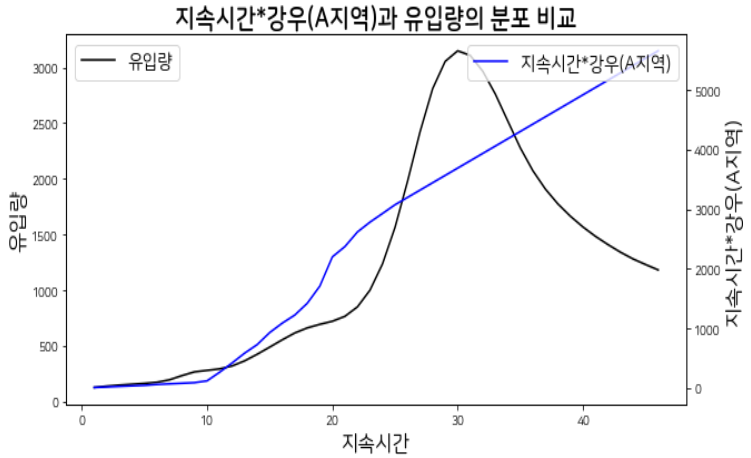
③ EDA와 논문에 근거하여
A, B, C, D지역 강우와 유역평균강수를
<u>1~6시간씩 shift</u>

탐색적 데이터 분석 (Exploratory Data Analysis)

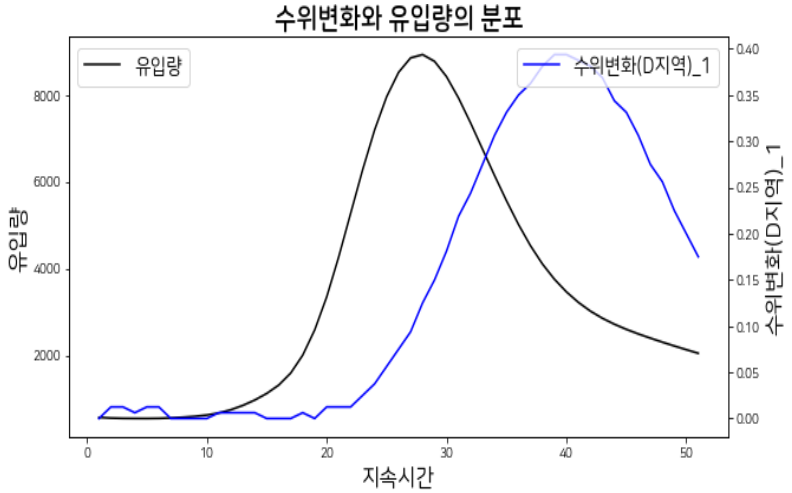
기타 파생 변수 고민



① 지속시간 대비 강우 : 초반 강우의 분포 **과대 평가**



② 지속시간 x 강우 : 후반 강우 분포가 **과대평가**

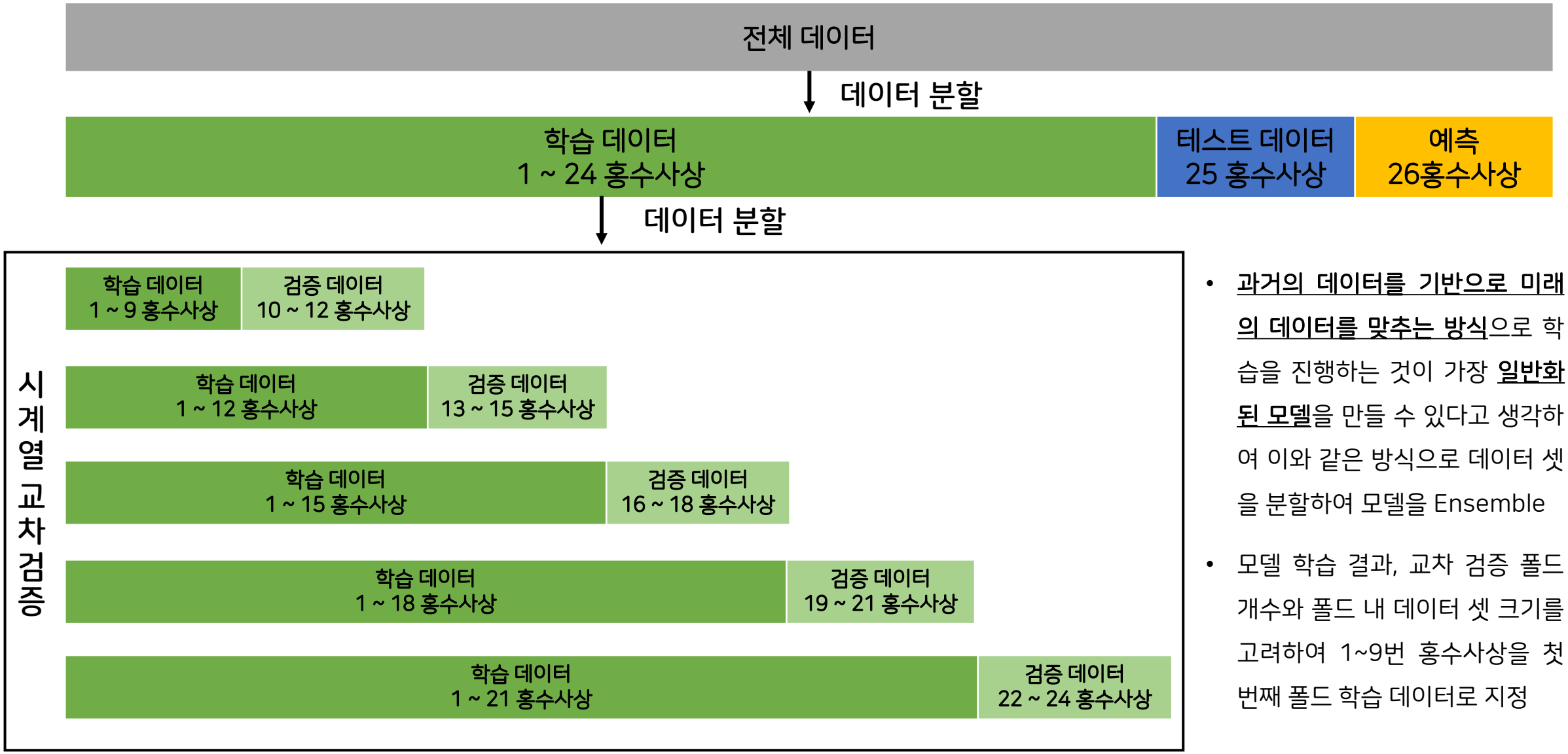


③ 수위변화량 (D지역) :

- 전체적인 분포는 다른 홍수사상에 대해서도 유입량의 분포와 비슷하나, 댐 유입 이후 D지역 수위 변화가 발생
  → 미래의 수위 변화량으로 유입량 **예측 불가능**
- 음수의 shift 없이 그대로 모델에 넣을 경우 **성능 하락**

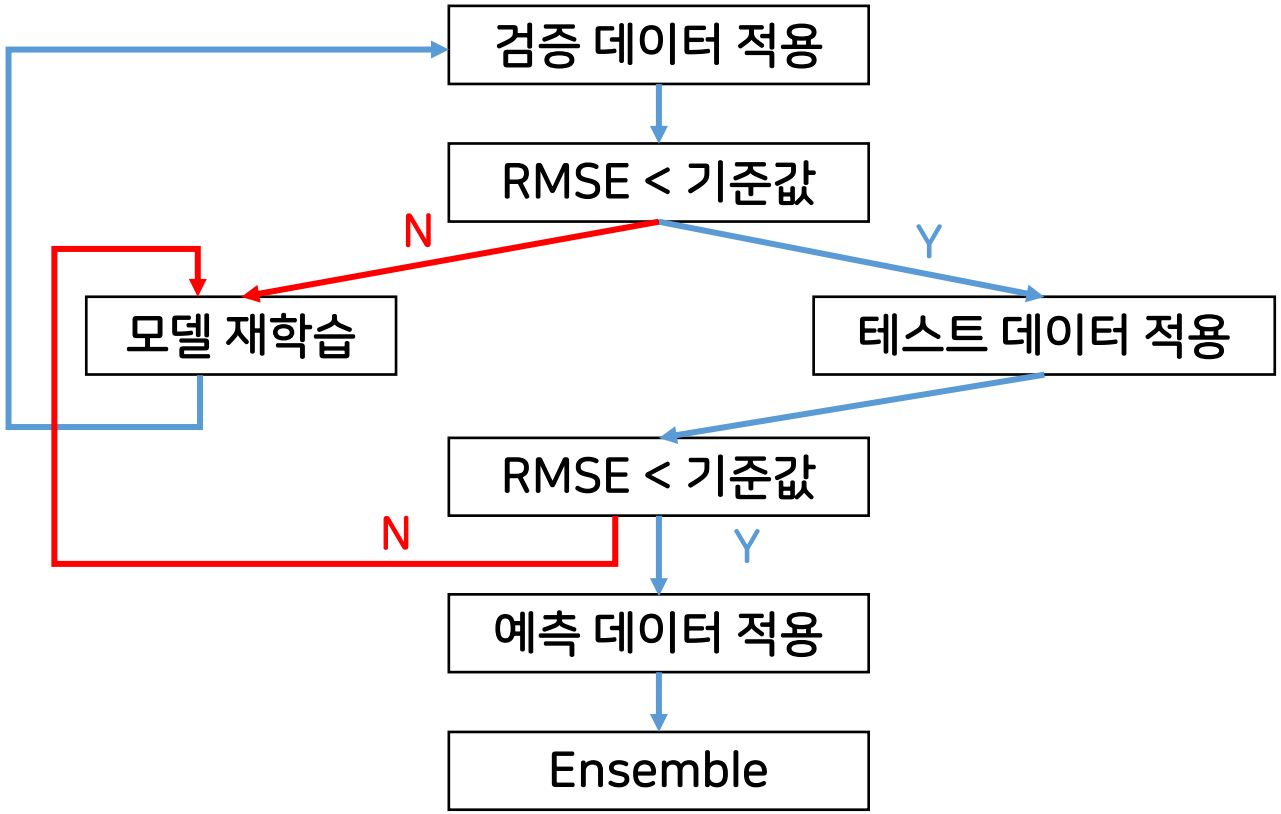
04
part

데이터 셋 분할



- 과거의 데이터를 기반으로 미래의 데이터를 맞추는 방식으로 학습을 진행하는 것이 가장 **일반화된 모델**을 만들 수 있다고 생각하여 이와 같은 방식으로 데이터 셋을 분할하여 모델을 Ensemble
- 모델 학습 결과, 교차 검증 폴드 개수와 폴드 내 데이터 셋 크기를 고려하여 1~9번 홍수사상을 첫 번째 폴드 학습 데이터로 지정

모델 학습 과정

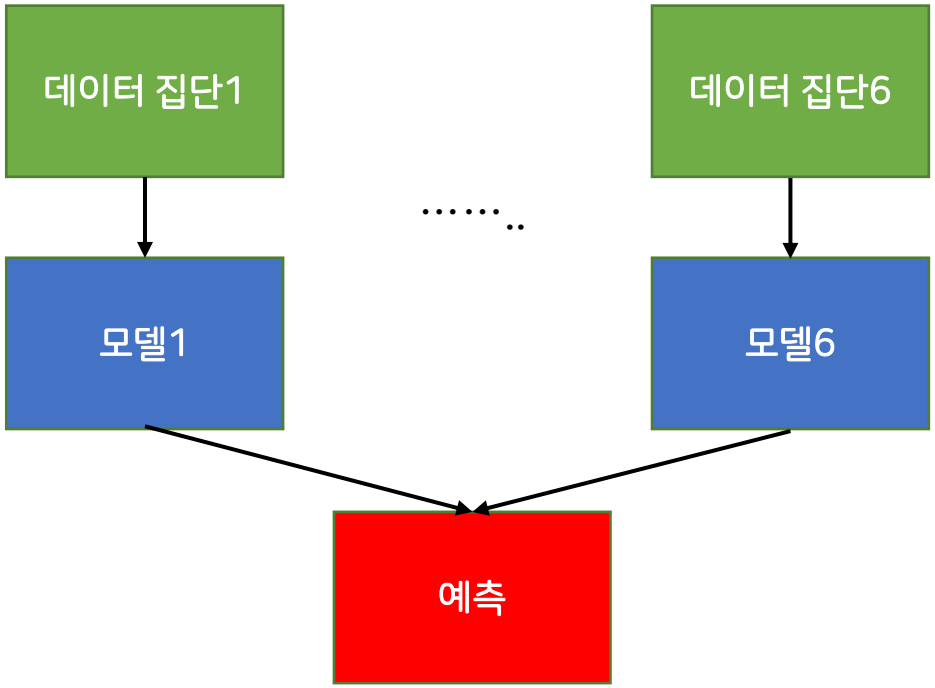


- 각 k-fold set을 기준으로 모델이 검증 데이터에 대하여 RMSE가 기준 값 보다 크다면 모델을 재학습
- 검증 데이터에 대하여 RMSE가 기준 값보다 작다면 테스트 데이터에 적용한 후, 테스트 데이터에 대하여 RMSE가 기준 값보다 크다면 모델을 재학습
- 모든 기준을 통과하면 예측 데이터에 적용하여 5개의 fold set으로 학습시킨 5개의 모델을 Ensemble

06
part

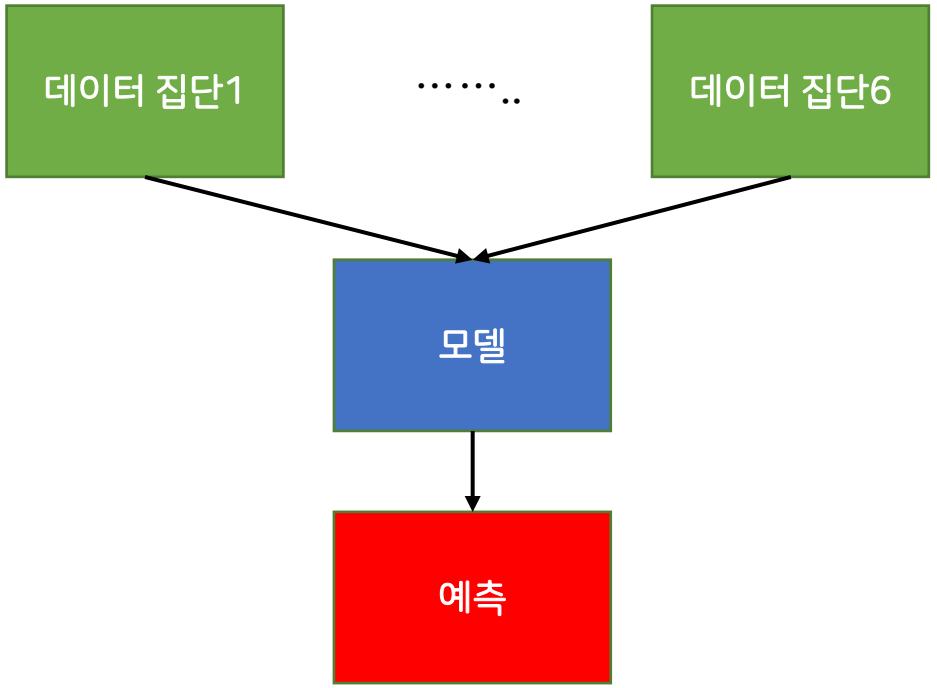
Machine Learning 모델

머신러닝 모델 학습방법



방법1

각 데이터 집단을 하나의 데이터 셋으로 보아
각각의 데이터 셋을 통해 모델을 학습시켜 Ensemble 하는 방법



방법2

각 데이터 집단에 존재하는 변수를 서로 다른 변수로 생각하여
한번에 모든 데이터 집단을 학습시켜 예측하는 방법

06

part

Machine Learning 모델

모델 후보

- Regression 모델: Linear, Ridge, Lasso, ElasticNet
- K-NN Regression
- SVR
- Bagging 기반 Tree : RandomForest
- Boosting 기반 Tree : XGBoost, LGBM, CatBoost

Machine Learning 모델

Regression 모델

회귀 분석의 기본 가정

- 선형성 : 독립변수와 종속변수간의 선형관계가 존재해야 함
- 정규성 : 오차는 정규분포를 따라야 함
- 독립성 : 오차는 서로 독립의 관계를 가져야 함

회귀 분석의 기본 가정을 확인 후

Linear(규제 X), Lasso(L1 규제), Ridge(L2 규제), ElasticNet(L1 + L2 규제)

등의 Regression 모델로 유입량을 예측

06
part

Machine Learning 모델

Regression 모델

선형성

	유입량	유역평균강수	강우(A지역)	강우(B지역)	강우(C지역)	강우(D지역)	수위(E지역)	수위(D지역)
유입량	1.00000	0.74227	0.61742	0.73995	0.60559	0.70057	0.90384	0.10947
유역평균강수	0.74227	1.00000	0.82363	0.95487	0.81020	0.82286	0.82007	0.07740
강우(A지역)	0.61742	0.82363	1.00000	0.80805	0.56174	0.61950	0.73794	-0.12065
강우(B지역)	0.73995	0.95487	0.80805	1.00000	0.75240	0.77505	0.78194	0.01888
강우(C지역)	0.60559	0.81020	0.56174	0.75240	1.00000	0.84339	0.67988	0.05587
강우(D지역)	0.70057	0.82286	0.61950	0.77505	0.84339	1.00000	0.77450	0.08974
수위(E지역)	0.90384	0.82007	0.73794	0.78194	0.67988	0.77450	1.00000	0.15014
수위(D지역)	0.10947	0.07740	-0.12065	0.01888	0.05587	0.08974	0.15014	1.00000

	컬럼	VIF
0	유역평균강수	42.783555
1	강우(A지역)	8.199935
2	강우(B지역)	23.255878
3	강우(C지역)	7.759979
4	강우(D지역)	9.233687
5	수위(E지역)	17.244936
6	수위(D지역)	5.701089

수위(D지역) 변수를 제외하고 유입량과의 선형 관계가 존재하여 선형성 만족

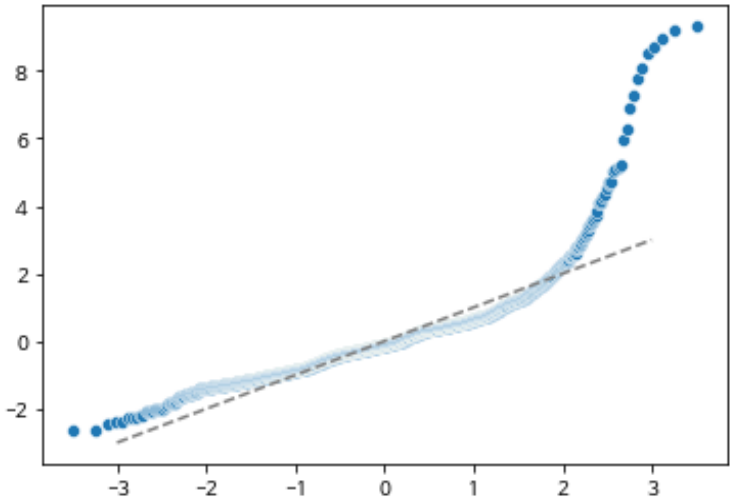
But, 독립변수간의 강한 상관관계가 존재 → 다중공선성 문제 발생(VIF >= 10) → 회귀계수가 불안정

따라서 변수에 규제를 가하는 Regression 모델을 사용하는 것이 좋음

Machine Learning 모델

Regression 모델

정규성




```
1 import scipy.stats
2 scipy.stats.shapiro(residual)
```

```
(0.8293247222900391, 0.0)
```

Q-Q 플롯과 Shapiro wilk test 결과

오차는 정규분포를 따르지 않는다는 것이 확인됨

따라서 정규성을 만족하지 않음

Machine Learning 모델

Regression 모델

독립성

OLS Regression Results

Dep. Variable:	유입량	R-squared (uncentered):	0.905
Model:	OLS	Adj. R-squared (uncentered):	0.905
Method:	Least Squares	F-statistic:	9929.
Date:	Thu, 09 Sep 2021	Prob (F-statistic):	0.00
Time:	00:04:59	Log-Likelihood:	-23642.
No. Observations:	2891	AIC:	4.730e+04
Df Residuals:	2884	BIC:	4.734e+04
Df Model:	7		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
유역평균강수	-4.9524	1.154	-4.293	0.000	-7.214	-2.691
강우(A지역)	-7.3613	0.425	-17.317	0.000	-8.195	-6.528
강우(B지역)	12.8127	0.849	15.087	0.000	11.148	14.478
강우(C지역)	-3.9031	0.734	-5.320	0.000	-5.342	-2.465
강우(D지역)	-0.1592	0.626	-0.254	0.799	-1.387	1.069
수위(E지역)	871.4139	12.853	67.801	0.000	846.213	896.615
수위(D지역)	-15.0313	0.290	-51.797	0.000	-15.600	-14.462

Omnibus:	1879.958	Durbin-Watson:	0.042
Prob(Omnibus):	0.000	Jarque-Bera (JB):	41705.647
Skew:	2.726	Prob(JB):	0.00
Kurtosis:	20.791	Cond. No.	173.

OLS 모델의 결과, R-squared의 값은 높으나

Durbin-Watson 검정 결과 0.042의 값을 가짐에 따라 오차들은 강한 양의 자기 상관성 존재

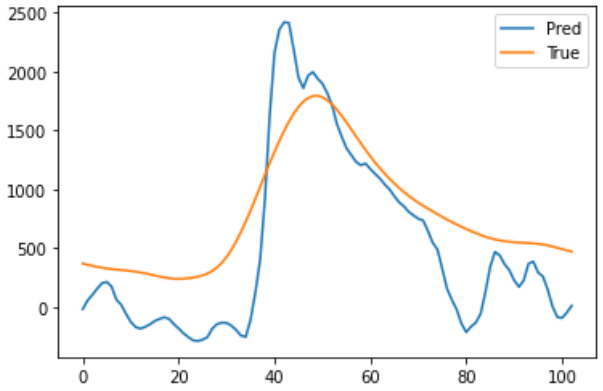
따라서 독립성을 만족하지 않음

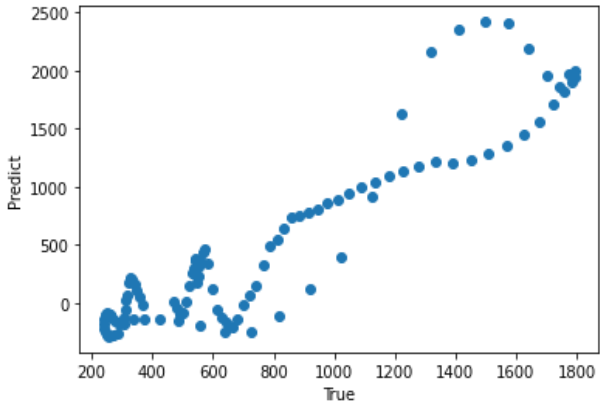
Machine Learning 모델

Regression 모델

학습 결과

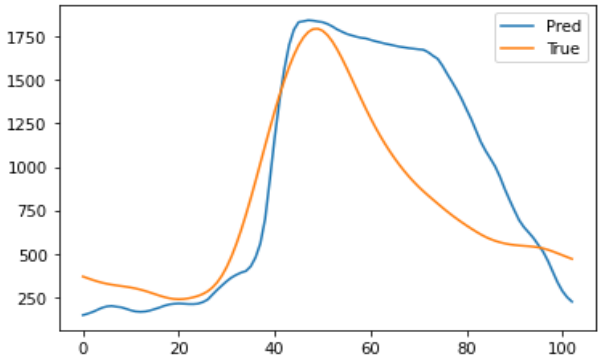
Regression 모델의 경우,
회귀분석의 기본 가정을 모두 만족하지
않으며 모델의 성능 또한 매우 좋지 않음

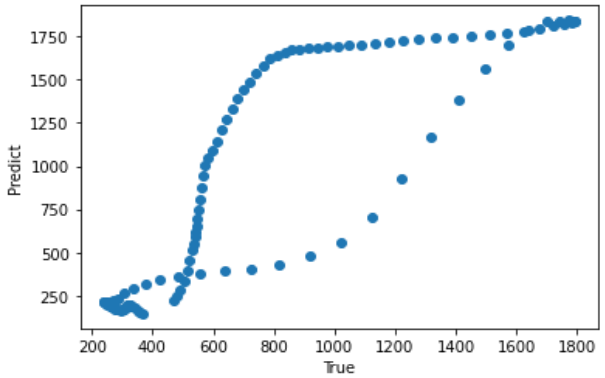




MAE : 383.7430585083823
MSE : 208025.511281365
RMSE : 456.09813777449807
MAPE : 76.06466438480601
MPE : 68.8052251708619 5
r2 : 0.11212127352515266

방법 1





MAE : 283.0086412325318
MSE : 145177.36782464405
RMSE : 381.02147947936487
MAPE : 40.593031733544 47
MPE : -11.413141647694149
r2 : 0.3803649578211018

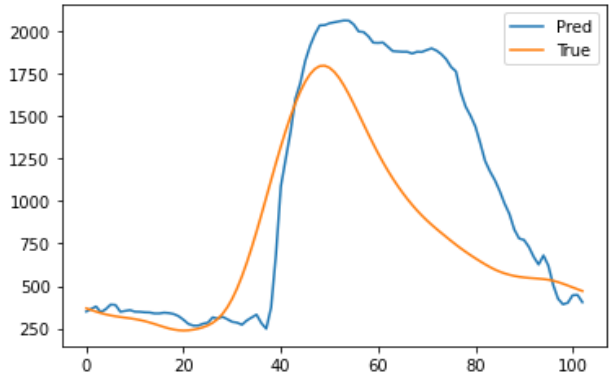
방법 2

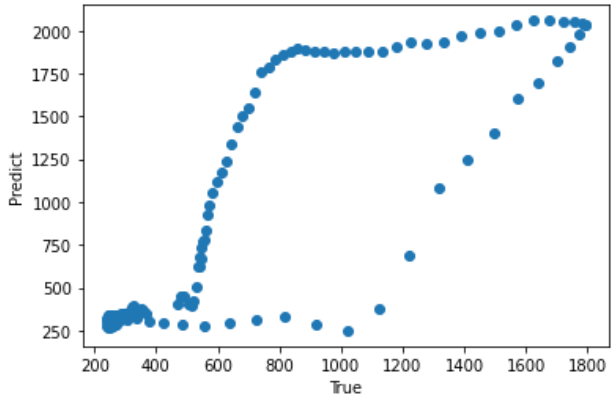
Machine Learning 모델

K-NN Regression

학습 결과

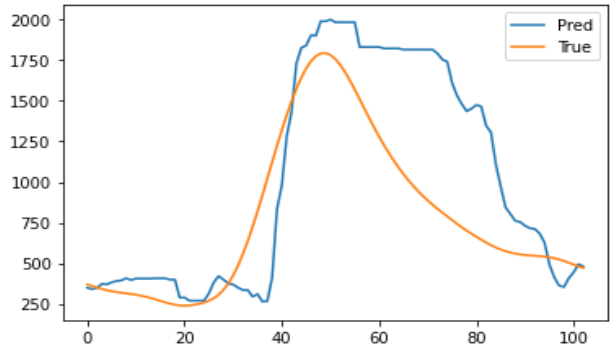
Regression 모델에 비하여
시작 시점, 유입량이 증가하는 시점,
종료 시점 등은 잘 맞추나,
여전히 모델의 성능이 좋지 않음

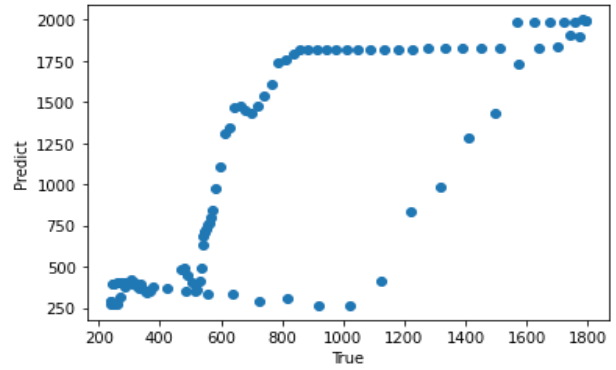




MAE : 348.52155600754816
MSE : 233934.06867662256
RMSE : 483.6673119786188
MAPE : 43.53348378752131
MPE : -29.72724142238496
r2 : 0.0015403317781177428

방법 1





MAE : 327.4369872277243
MSE : 197732.981901021
RMSE : 444.67176872500124
MAPE : 43.169005596564574
MPE : -30.299796877777695
r2 : 0.1560510676265332

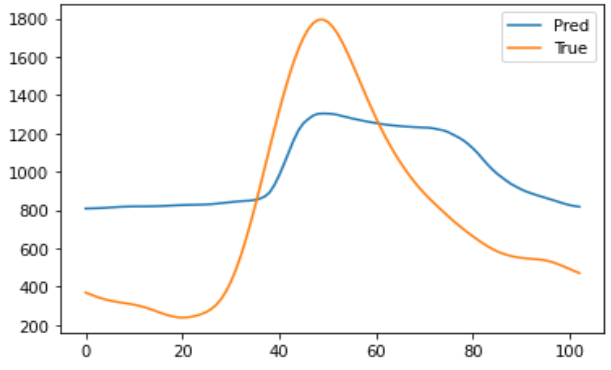
방법 2

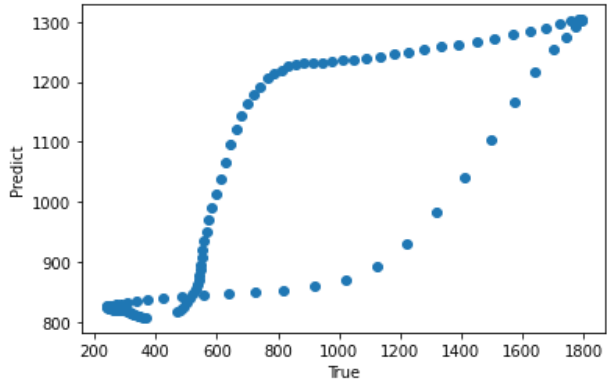
Machine Learning 모델

SVR

학습 결과

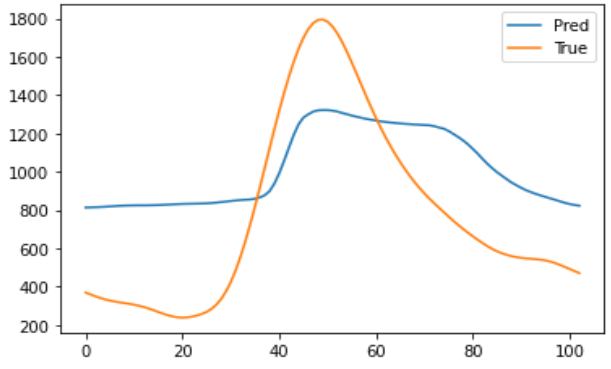
SVR의 경우 본 과제에서
매우 좋지 못한 성능을 보임

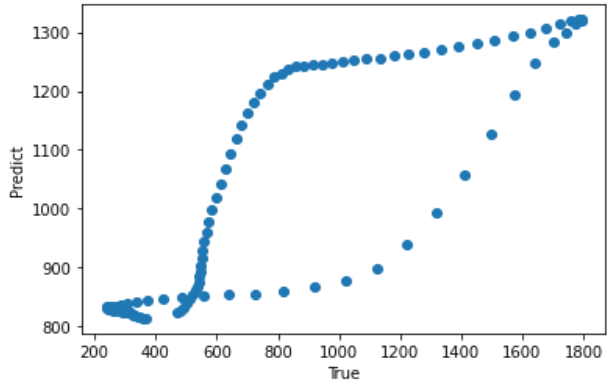




MAE : 384.5906074847475
MSE : 167744.6663549481
RMSE : 409.5664370464798
MAPE : 82.90118304955597
MPE : -72.88123584511467
r2 : 0.28404492401542625

방법 1





MAE : 385.583028469313
MSE : 168628.7587771581
RMSE : 410.6443214963018
MAPE : 83.6726606494701
MPE : -74.16461138218368
r2 : 0.2802715077210364

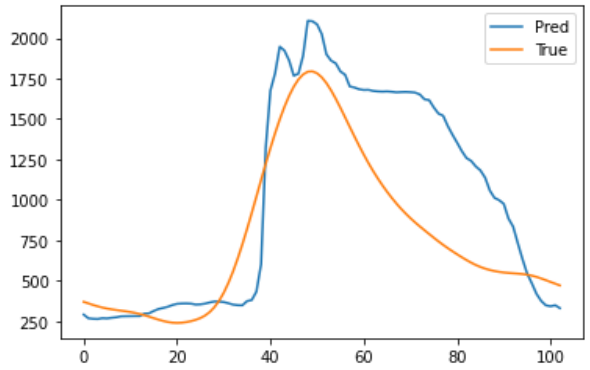
방법 2

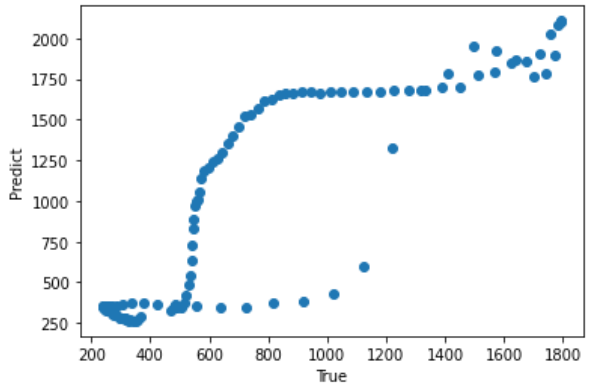
Machine Learning 모델

Bagging 기반 Tree

학습 결과

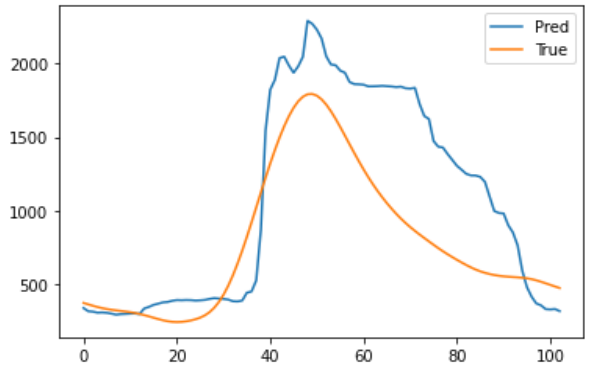
현재까지의 모델은 대부분
Underfitting 상태,
따라서 모델의 Variance를 줄여
Overfitting을 해소하는 것에 적합한
Bagging 기반 Tree 모델은
본 과제에 적합하지 않은 모델로 판단됨

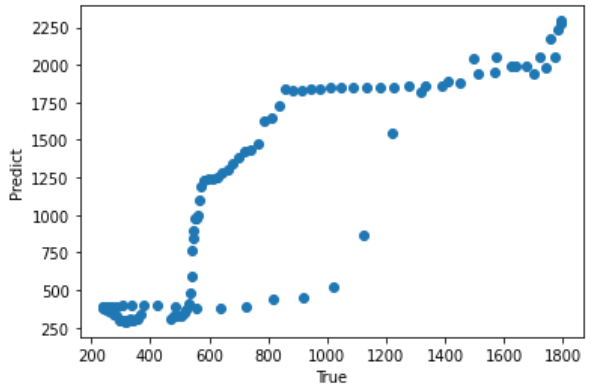




MAE : 311.5466705609095
MSE : 164048.35188047102
RMSE : 405.02882845603847
MAPE : 42.52890777467944
MPE : -27.563242609063533
r2 : 0.29982125340903765

방법 1





MAE : 359.847225510912
MSE : 206219.55427664384
RMSE : 454.11403223930864
MAPE : 47.081071198345775
MPE : -34.641333102438715
r2 : 0.11982932238677224

방법 2

Machine Learning 모델

Boosting 기반 Tree

학습 결과

지금까지의 모델은

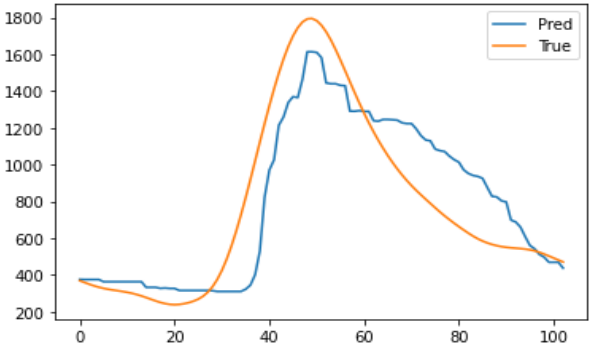
대부분 Underfitting(High Bias) 상태

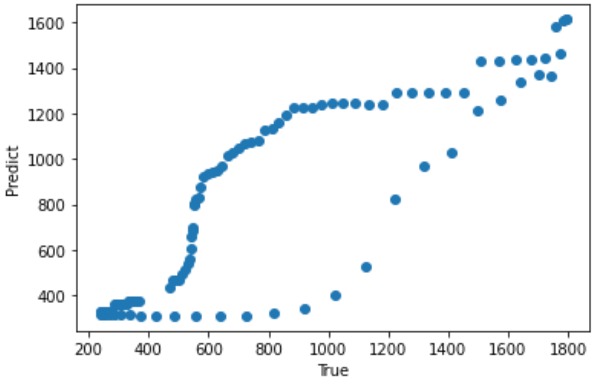
→ 모델의 Bias를 줄여주는

Boosting 기반의 Tree 모델이

머신러닝 모델 중

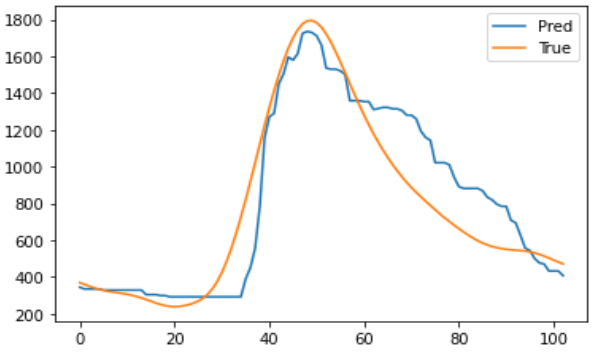
<u>본 과제에 가장 적합한 모델</u>

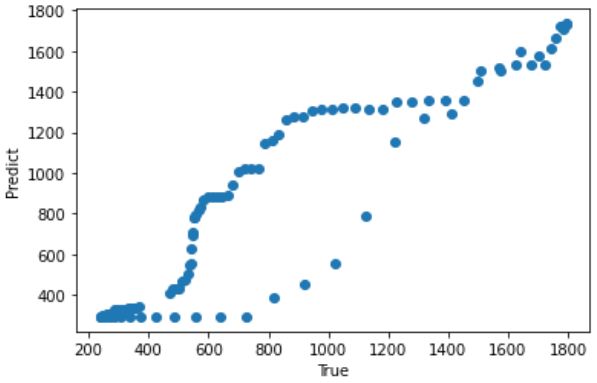




MAE : 185.96503762644292
MSE : 56351.44612427726
RMSE : 237.38459538116044
MAPE : 25.52161018701658
MPE : -9.549647866674896
r2 : 0.759485027044751 4

방법 1





MAE : 143.85518188693302
MSE : 37909.89382810763
RMSE : 194.70463227182765
MAPE : 20.49797614454558
MPE : -7.985609697954514
r2 : 0.8381958633555725

방법 2

Machine Learning 모델

종합

머신러닝 모델 중 본 과제에서 가장 적합한 모델은 XGBoost 인 것으로 확인됨

하지만 해당 모델의 경우 **유입량을 과소 추정하는 특징**을 보여줌

또한 해당 모델의 경우, 깊이가 얕은 Decision Tree를 앙상블하는 방식이기에

변수 간의 강한 상관 관계가 존재하는 현재 데이터로는

모든 변수를 효과적으로 활용하지 못함

따라서 본 과제에서 머신러닝 모델의 사용은 적합하지 않다고 판단함

07

part

Convolutional Neural Network 모델

개요

① 유입량 예측을 위해서는 변수 간의 상호작용을 반영하는 것이 중요하며, 제공된 데이터는 변수 간 상관계수가 높아 회귀 모델이나 얕은 트리 기반의 ML 모델이 변수 간 상호작용을 제대로 학습하기 어려움

② ML 모델의 데이터 집단의 활용 방법에 따른 모델 학습 결과를 보면 데이터 집단 간의 학습 방식도 중요하게 작용하는 것으로 보임

→ **본 과제에서는 변수 간의 상호작용과 데이터 집단 간의 결합을 효과적으로 학습하는 모델이 필요**

07

part

Convolutional Neural Network 모델

개요

머신러닝 모델과 달리,

변수 간의 상호 작용과 데이터 집단 간의 결합을

효과적, 효율적으로 학습할 수 있는

WaveNet을 변형한

Convolutional Neural Network 모델을 구현

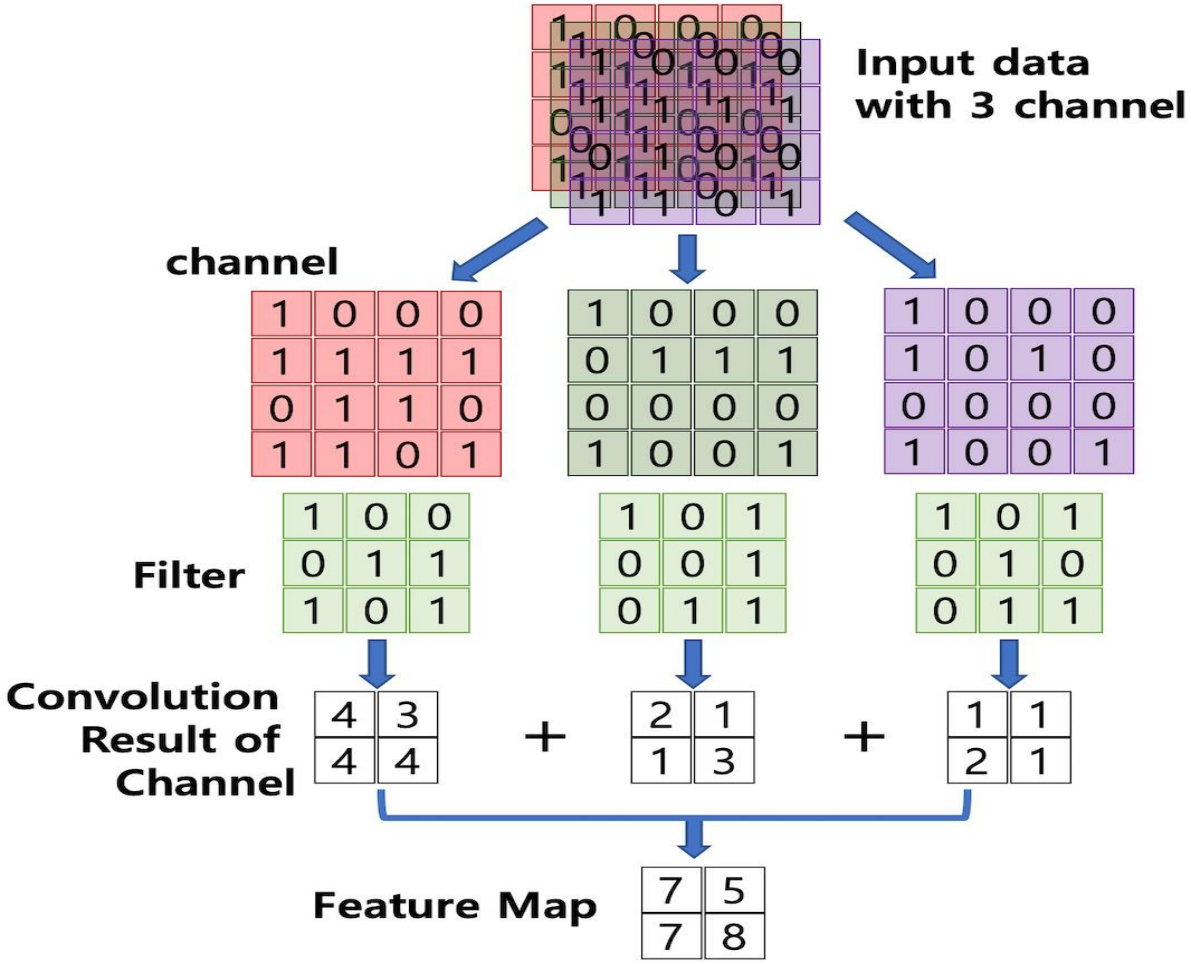
07

part

Convolutional Neural Network 모델

개요

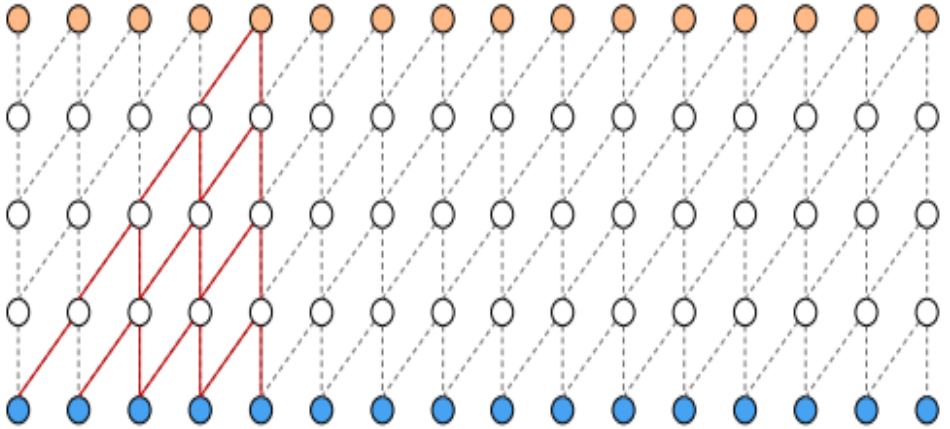
- Convolutional Neural Network은 다음과 같이 3개의 채널 데이터가 들어오면 채널마다 서로 다른 필터가 Stride하며 변수 간 상호작용을 학습하고, 각 채널의 합성곱 결과값을 합하여 하나의 Feature Map을 생성
- **본 과제에서는 채널을 데이터 집단으로 구성**하여 **Filter를 통해 변수 간의 상호작용을 학습**하고, **각 데이터 집단의 결과 값을 하나로 합쳐 Feature Map을 생성**하도록 구현
- CNN 기반의 모델은 각 데이터 집단이 서로 다른 모델을 학습시키는 ML 모델과 달리 **데이터 집단의 결합과 변수와의 상호작용을 End-to-End 과정으로 학습** 가능

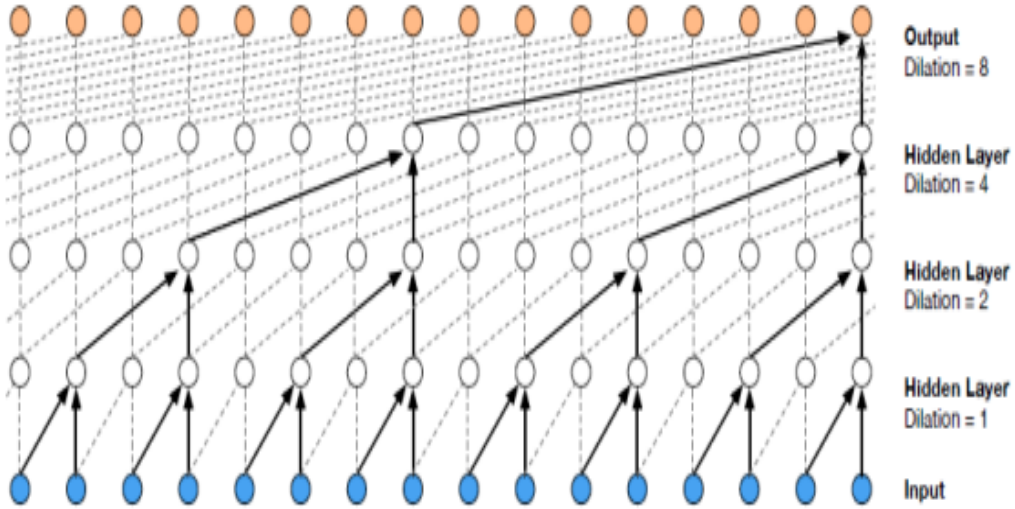


Convolutional Neural Network 모델

개요

- 단순한 CNN 모델은 위 그림과 같이 인접 변수와의 상호작용만을 학습하기 때문에 멀리 있는 변수와의 상호작용을 학습시키기 위해서는 Layer를 깊게 쌓아야 함
- 하지만 Layer를 깊게 쌓게 된다면 기울기 소실, 파라미터 증가에 따른 과적합이 발생할 수 있어 모델의 성능이 감소하게 됨
- 따라서 **변수와의 상호작용을 조금 더 효율적으로 학습시키기 위해서 WaveNet의 Dilated Convolution을 활용**함
- Dilated Convolution은 아래의 그림과 같이 학습이 진행되어 **Layer를 깊게 쌓아도 멀리 있는 변수와의 상호작용을 효율적으로 학습**을 할 수 있음





Convolutional Neural Network 모델

모델 후보

- WaveNet : 기존 WaveNet을 본 과제에 맞게 변형한 형태
- CNN1D : 집단마다 서로 다른 필터가 변수 간의 상호작용을 학습
- CNN2D : 집단마다 동일한 필터가 변수 간의 상호작용을 학습
- Time_CNN2D : 집단마다 서로 다른 필터가 시간의 인과성과 변수 간의 상호작용을 학습

Convolutional Neural Network 모델

데이터 전처리

- 유역평균강수, 강우 변수를 1~6시간 shift한 변수를 추가 (결측치의 경우 첫 시간대의 값으로 채움)

: 강수의 시간에 따른 영향력을 반영

- 최소 값은 0, 최대 값은 1.2를 곱한 값을 활용하는 변형 MinMax 정규화를 통해서 Scaling을 진행

→ X = (X – 0) / ((X_max * 1.2) – 0)

: 단순한 MinMax 정규화는 최소, 최대 값이 제한되어 일반화된 모델을 만들 수 없음

- 데이터를 각 모델의 특성에 맞게 변형하여 학습

: ex) CNN1D의 경우 (전체 데이터의 수, 변수, 데이터 집단)으로 변형

Convolutional Neural Network 모델

모델 학습 방법

- 시계열 교차 검증을 통한 Ensemble 진행

: 모델의 분산을 줄이고(Overfitting 방지) 일반화된 모델을 만듦

- 활성화 함수의 경우 ELU 함수를 사용

: 활성화 함수를 통해서 변수 간의 복잡한 상호작용을 조금 더 효과적으로 학습함

- BatchNormalization과 Dropout 사용

: 모델의 일반화 성능이 향상됨

- Optimizer의 경우 Adam을 사용

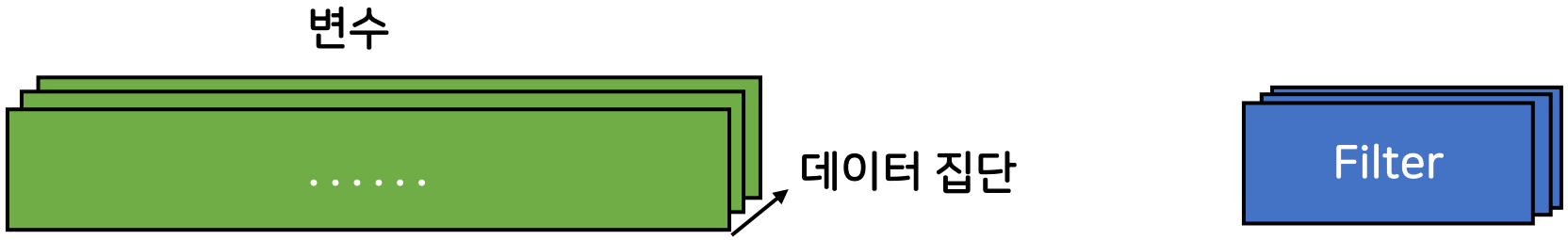
- Scheduler을 사용하여 검증 데이터 셋에 대하여 더 이상의 성능 향상이 없으면 학습률을 감소시킴

: Overshooting을 방지하고 학습을 안정적으로 할 수 있음

Convolutional Neural Network 모델

WaveNet

모델 구조



WaveNet은 (전체 행 데이터, 변수, 데이터 집단)으로 구성된 데이터를 입력 받아

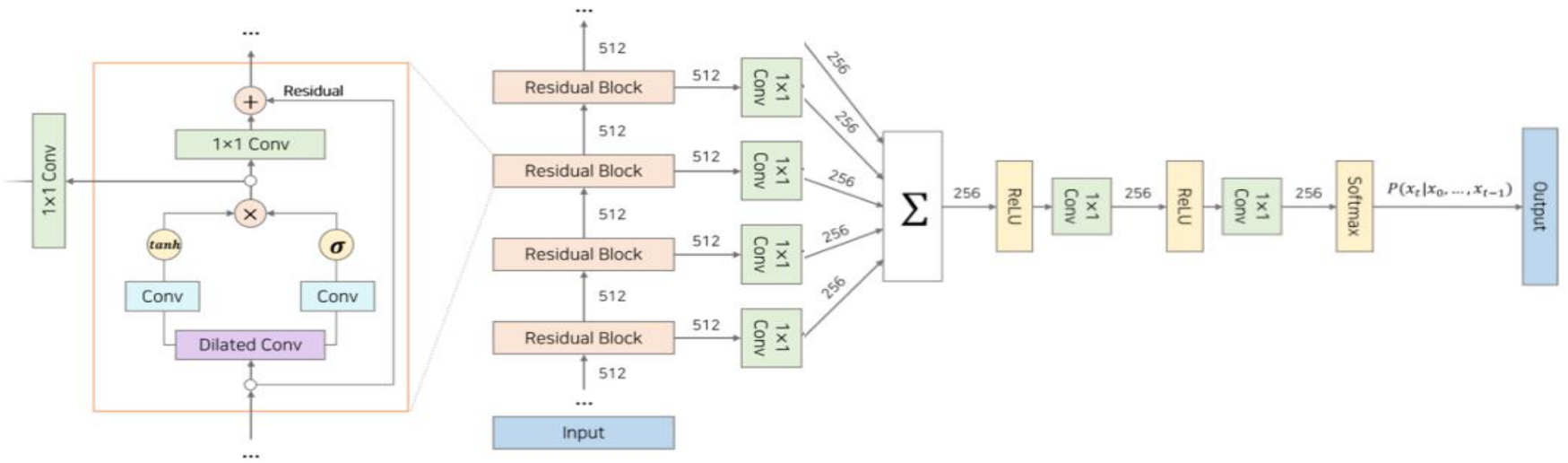
집단마다 서로 다른 필터가 stride하며 변수 간의 상호작용을 학습

각 필터에 따른 결과 값은 하나로 합쳐져 집단 간의 결합을 이룸

Convolutional Neural Network 모델

WaveNet

모델 구조



WaveNet은 그림과 같이 여러 개의 Residual Block으로 이루어져

조금 더 효과적으로 Layer을 깊게 쌓을 수 있다는 특징 있음

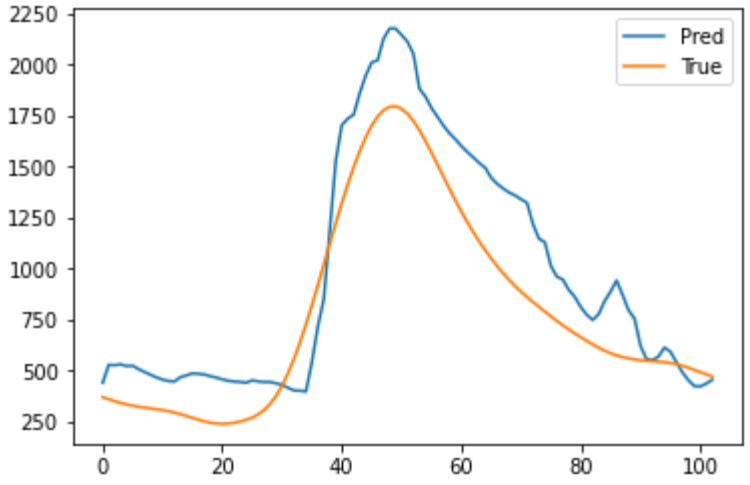
본 과제의 경우 <u>3개의 Residual Block</u> 쌓음

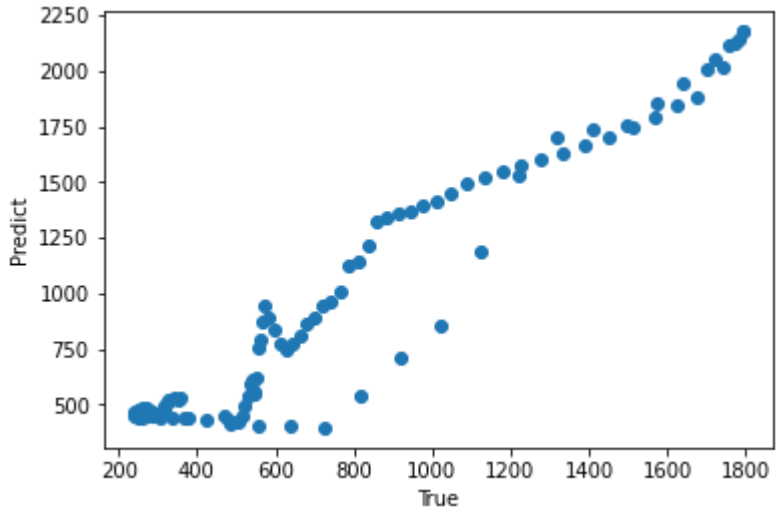
07
part

Convolutional Neural Network 모델

WaveNet

학습 결과






```
=================================
Total params: 4,246,105
Trainable params: 4,238,481
Non-trainable params: 7,624
---------------------------------

MAE : 221.2141504097239
MSE : 62311.876230848684
RMSE : 249.62346891037447
MAPE : 35.94690187845539
MPE : -30.923012535796822
r2 : 0.7340451708479454
```

전체적인 유입량의 추세를 잘 맞추나

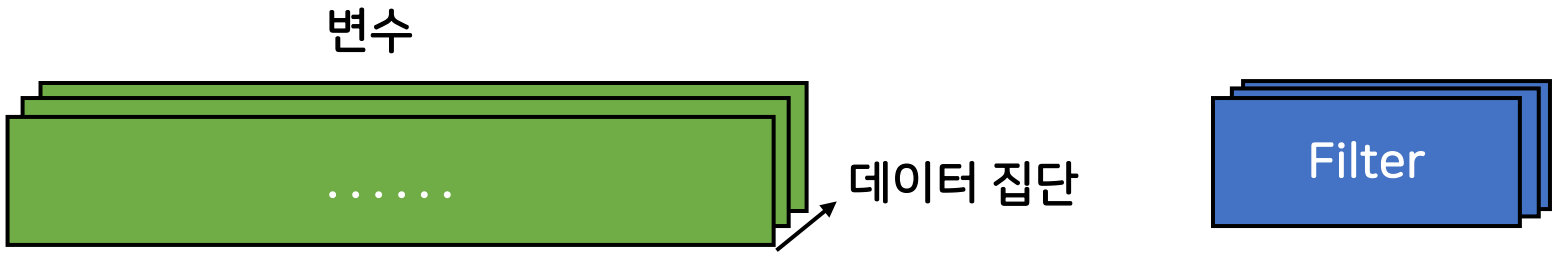
Residual Block 에 의하여 **Param의 수가 증가하여 과적합이 발생**

이에 모델이 과대 추정을 하는 것으로 판단됨

Convolutional Neural Network 모델

CNN1D

모델 구조



CNN1D는 (전체 행 데이터, 변수, 데이터 집단)으로 구성된 데이터를 입력 받아

집단마다 서로 다른 필터가 stride하며 변수 간의 상호작용을 학습

각 필터에 따른 결과 값은 하나로 합쳐져 집단 간의 결합을 이룸

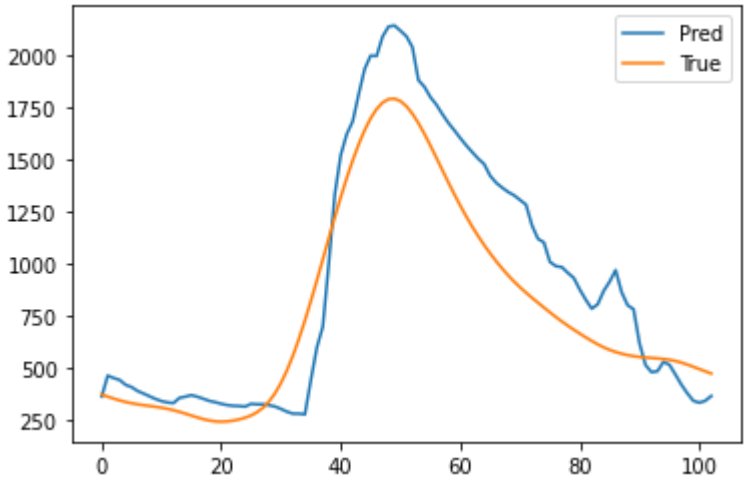
WaveNet과 달리, **Residual Block이 존재하지 않아 Param의 수가 적음**

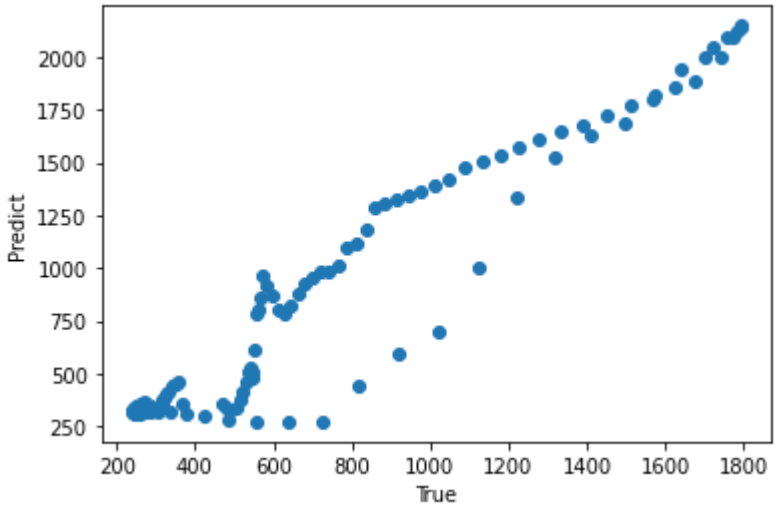
07
part

Convolutional Neural Network 모델

CNN1D

학습 결과






```
=================================================================
Total params: 1,620,793
Trainable params: 1,619,633
Non-trainable params: 1,160
_________________________________________________________________
```

```
MAE : 199.91738204291565
MSE : 55991.71130748098
RMSE : 236.62567761652787
MAPE : 27.124652636046903
MPE : -15.402254154010603
r2 : 0.7610204199349712
```

전체적인 유입량의 추세와 정점을 잘 맞추며

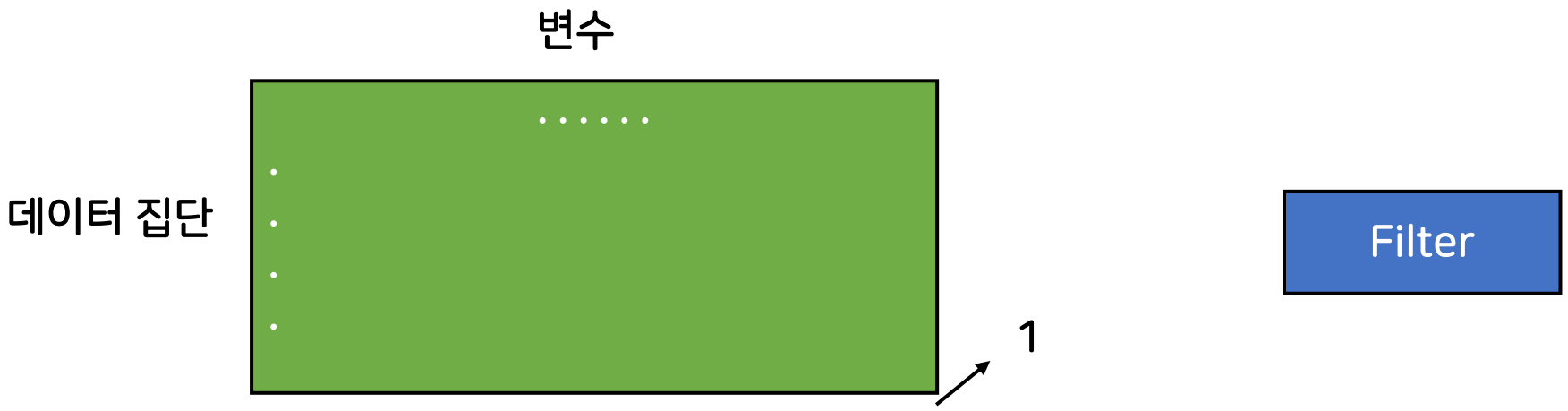
유입량이 많은 시점만 과대 추정하고 있음

<u>Param의 수가 다른 모델에 비해 상대적으로 적어 과적합 발생 가능성 낮음</u>

Convolutional Neural Network 모델

CNN2D

모델 구조



CNN2D는 (전체 행 데이터, 데이터 집단, 변수, 1)로 구성된 데이터를 입력 받아

집단마다 서로 동일한 필터가 변수 간의 상호작용과 집단 간의 결합을 학습

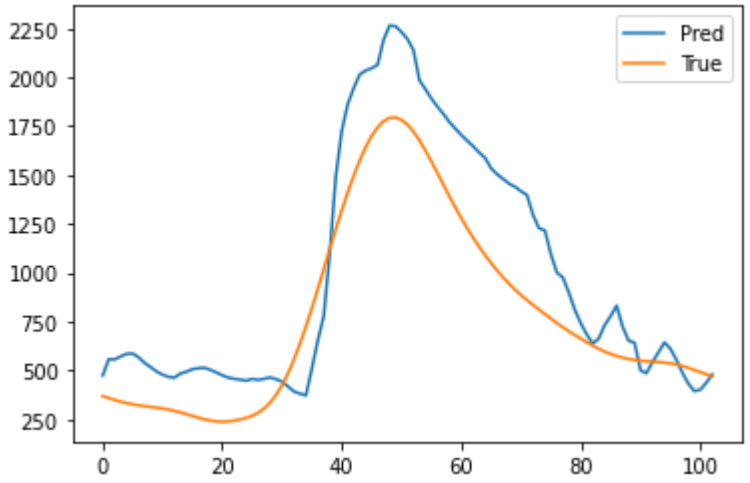
데이터가 3D의 구조를 가짐에 따라 **param의 수가 매우 많다는 단점** 존재

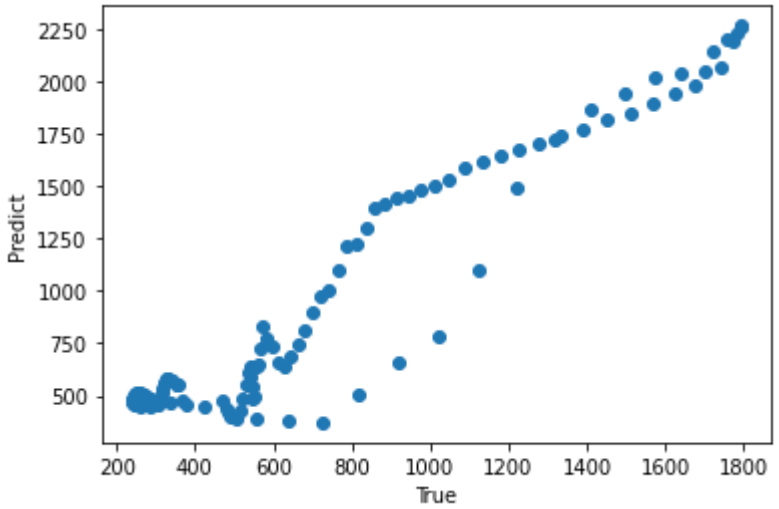
07
part

Convolutional Neural Network 모델

CNN2D

학습 결과






```
=================================
Total params: 6,354,553
Trainable params: 6,353,393
Non-trainable params: 1,160
---------------------------------

MAE : 253.98032539073503
MSE : 87350.1996822451
RMSE : 295.5506719367173
MAPE : 39.8944731428141
MPE : -33.77172068160748
r2 : 0.6271784956879809
```

Param의 수가 다른 모델에 비해 상대적으로 많아

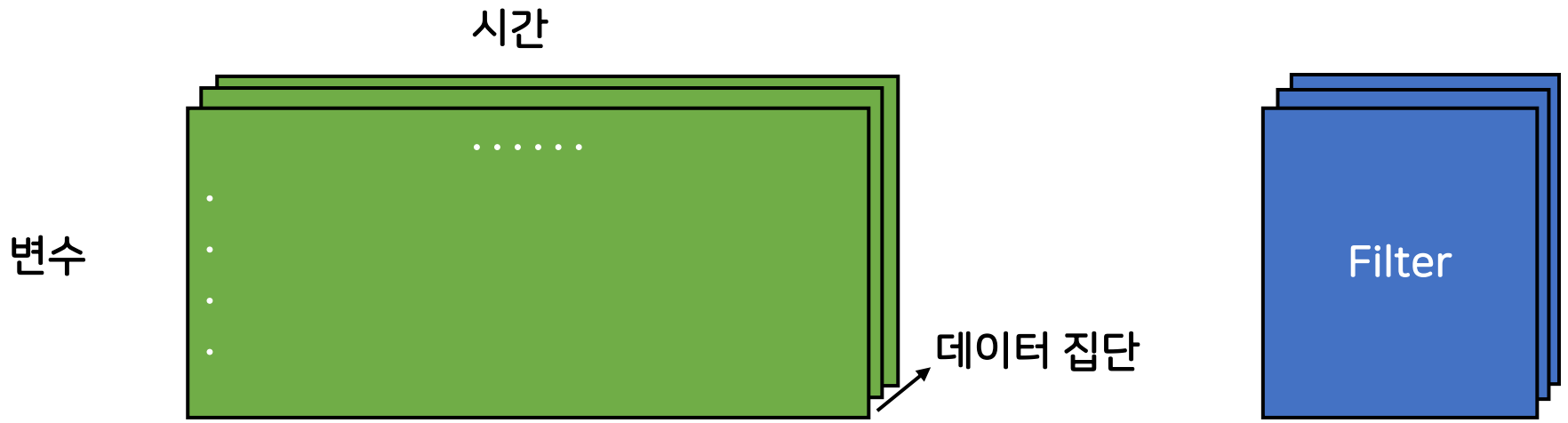
과적합이 발생하여 전체적으로 유입량을 과대 추정하는 경향 있음

학습 시간이 다른 모델에 비해서 매우 느리다는 단점 존재

Convolutional Neural Network 모델

Time_CNN2D

모델 구조



Time_CNN2D는 (전체 행 데이터, 변수, 시간, 데이터 집단)으로 구성된 데이터를 입력 받아

<u>집단마다 서로 다른 필터가 stride하며 시간의 인과성과 변수간의 상호작용을 학습</u>

각 필터에 따른 결과 값은 하나로 합쳐져 집단 간의 결합을 이룸

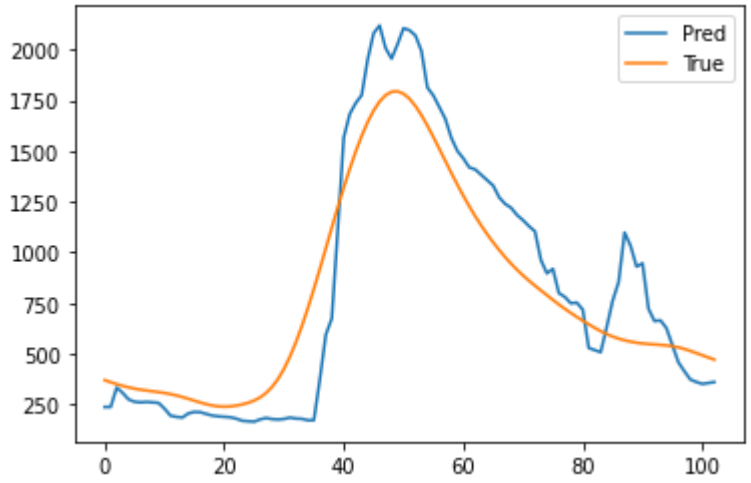
좌측에만 Zero-padding을 하는 Causal padding을 통하여 시간의 인과성을 반영 가능

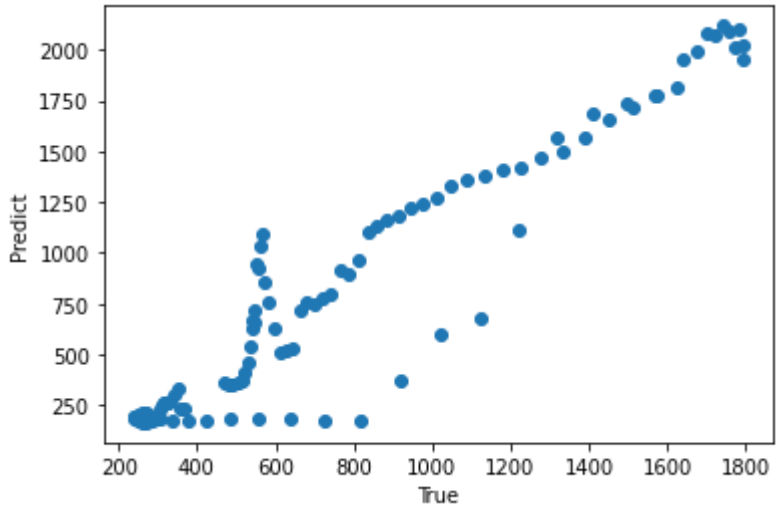
07
part

Convolutional Neural Network 모델

Time_CNN2D

학습 결과






```
Total params: 1,623,577
Trainable params: 1,621,457
Non-trainable params: 2,120
```

```
MAE : 189.9909988250853
MSE : 54457.19461548657
RMSE : 233.36065352900985
MAPE : 27.467385975332594
MPE : 3.4885246056880246
r2 : 0.7675699278191248
```

유입량의 전체적인 추세와 정점을 다른 CNN 모델에 비하여 잘 맞추지 못함

모델의 학습 결과 또한 매우 불안정

최종 모델

최종 모델 선정

- ML 모델과 달리, 데이터 집단과 변수 간의 상호작용을 효과적으로 학습할 수 있음
- WaveNet, CNN2D에 비해 param의 수가 적어 모델 학습 속도가 빠르며, 과적합 발생 가능성 낮음
- 다른 모델에 비해 유입량이 많은 시점만을 과대 추정

  → 홍수 피해를 막기 위해서는 유입량의 과소 추정보다 과대 추정이 더 안전하다고 판단

→ CNN1D 모델을 최종 모델로 선정

08
part

최종 모델

모델 구조

```
nf = 32
fs = 14
padding = 'same'
activation = 'elu'

model = Sequential()

model.add(keras.layers.InputLayer((X_train.shape[1], X_train.shape[2])))

model.add(Conv1D(filters = nf, kernel_size = fs, padding = padding))
model.add(BatchNormalization())
model.add(Activation(activation = activation))
model.add(Dropout(0.2))

model.add(Conv1D(filters = nf * 2, kernel_size = fs, padding = padding, dilation_rate=2))
model.add(BatchNormalization())
model.add(Activation(activation = activation))
model.add(Dropout(0.2))

model.add(Conv1D(filters = nf * 4, kernel_size = fs, padding = padding, dilation_rate=4))
model.add(BatchNormalization())
model.add(Activation(activation = activation))
model.add(Dropout(0.2))

model.add(Conv1D(filters = nf * 8, kernel_size = fs, padding = padding, dilation_rate=8))
model.add(BatchNormalization())
model.add(Activation(activation = activation))
model.add(Dropout(0.2))

model.add(Conv1D(filters =  nf * 8, kernel_size = 1))
model.add(Flatten())
model.add(Dense(100))
model.add(BatchNormalization())
model.add(Activation(activation = activation))
model.add(Dropout(0.2))
model.add(Dense(1))# output size

optimizer = keras.optimizers.Adam(lr=0.01)

model.compile(loss = 'mse', optimizer = optimizer, metrics=[tf.keras.metrics.RootMeanSquaredError()])
```

- 파라미터 튜닝을 통하여 최적의 필터의 개수와 필터의 크기를 찾음
  : 변수 간의 상호작용 학습에 중요한 요소
- Padding과 활성화 함수의 경우 각각 same과 ELU를 사용

- 총 4개의 Layer를 쌓음
  : 적절한 param을 가짐
- 필터의 개수와 Dilation_rate 모두 2의 배수로 증가하도록 설정
  : 변수 간의 상호작용과 데이터 집단 간의 결합을 효율적으로 학습
- Batch Normalization, Dropout 층을 쌓음
  : 모델의 일반화 성능 향상

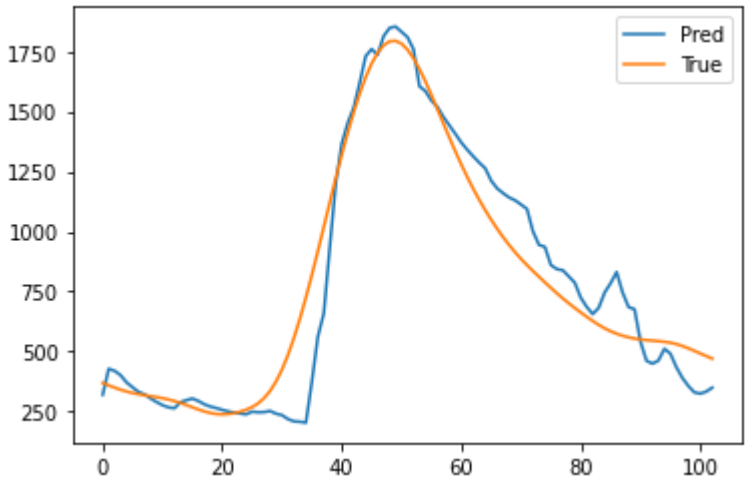
- 필터의 크기가 1인 Convolution를 통하여 완전 연결층을 구현
  : 보다 효과적으로 학습 진행

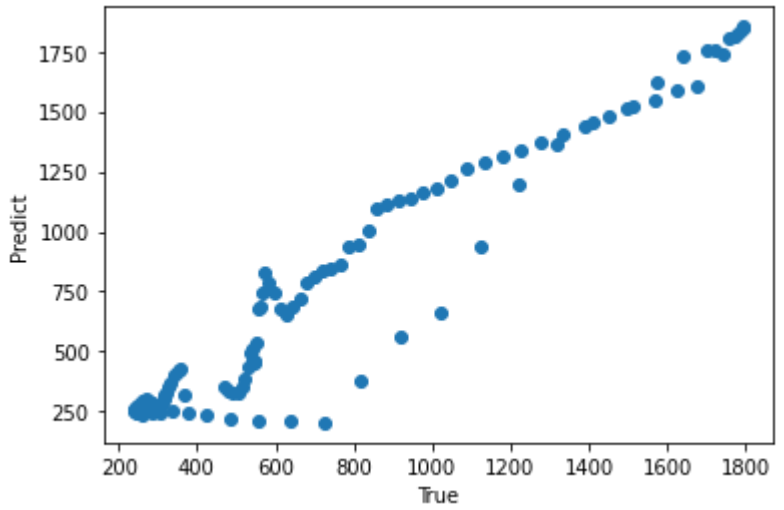
- Adam을 사용
- 검증 데이터에 대하여 RMSE가 가장 낮은 Model을 선택함

08
part

최종 모델

학습 결과





MAE : 103.34554615362751
MSE : 20915.53939501696
RMSE : 144.62205708334037
MAPE : 15.907976042196392
MPE : 1.2344191048670972
r2 : 0.9107298793922218

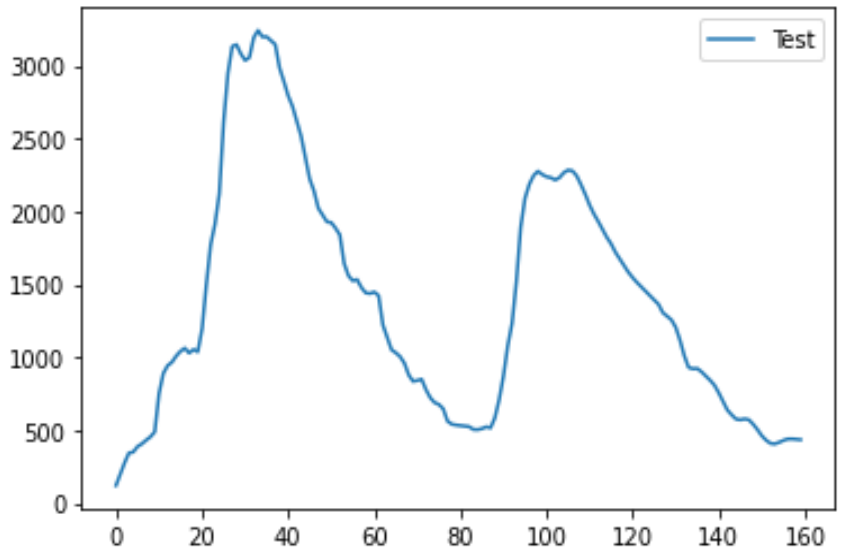
테스트 데이터에 대하여 전체적인 유입량의 추세와 정점을 잘 맞추고,

RMSE 값도 약 144로 지금 까지의 모델들 중 **가장 높은 성능**을 보여주며,

모델의 학습 결과 또한 매우 안정적임

최종 모델

예측 결과



26번 홍수 사상에 대한 CNN1D 모델의 예측 결과

09
part

활용방안

현업에서의 어려움 | 가설 및 실험 구상 | 실험 결과

국지성 호우란?

→ 특정 지역에 집중적으로 많은 비가 내리는 현상

현업에서의 문제점

→ 특정 지역에 비가 많이 왔음에도 불구하고 **관측되지 않는 경우가 있어** 댐의 유입량을 예측하는 데에 어려움이 존재

활용방안

가설

: 최종 모델은 모든 변수와, 데이터 집단을 조화롭게 사용하기 때문에 관측 오류에 강건할 것이다

실험 구상

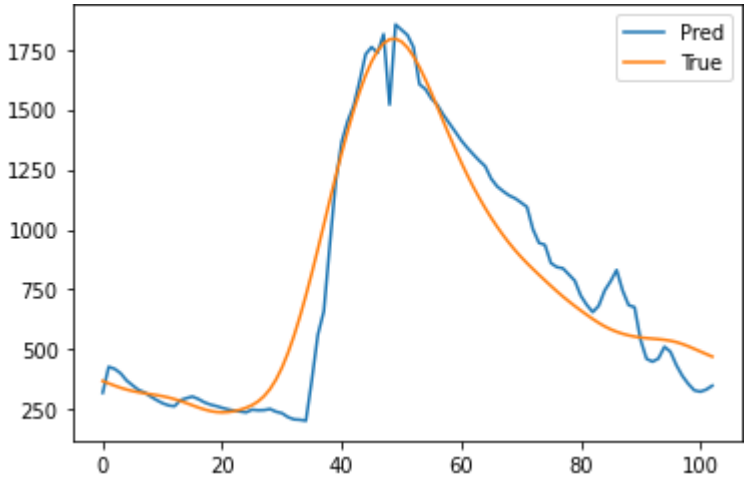
① 테스트 데이터의 유입량의 정점에서 가장 중요한 변수로 작용하는 데이터 집단 1의 “수위(E지역)”를 0으로 수정 (모델의 강건함을 보다 뚜렷하게 시각화하기 위해 강우량 대신 “수위(E지역)”에 0 대입)

② XGBoost와 최종 모델에 수정된 데이터를 넣어 예측 결과값 도출

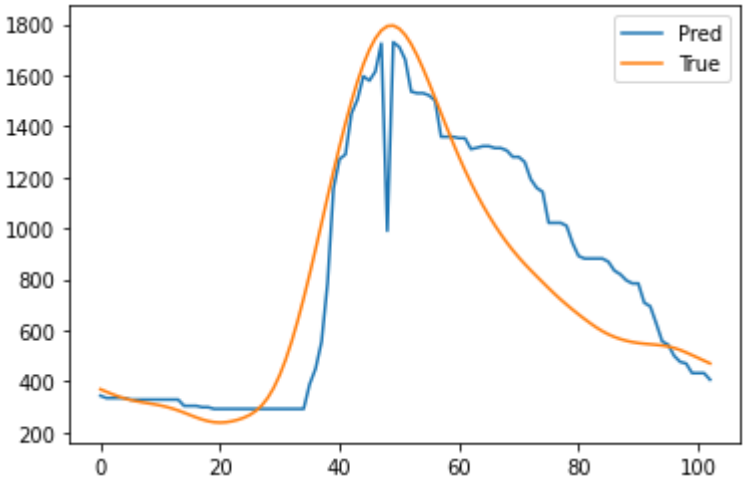
③ 두 모델의 성능 변화 비교

활용방안

현업에서의 어려움 | 가설 및 실험 구상 | **실험 결과**



CNN1D 최종 모델



XGBoost 모델

실험 결과: XGBoost 모델에 비해 최종 모델이 더 강건함

해석: 최종 모델은 필터의 striding을 통해 변수간, 집단간 상호작용을 반영하여 **관측 오류에도 강건**함

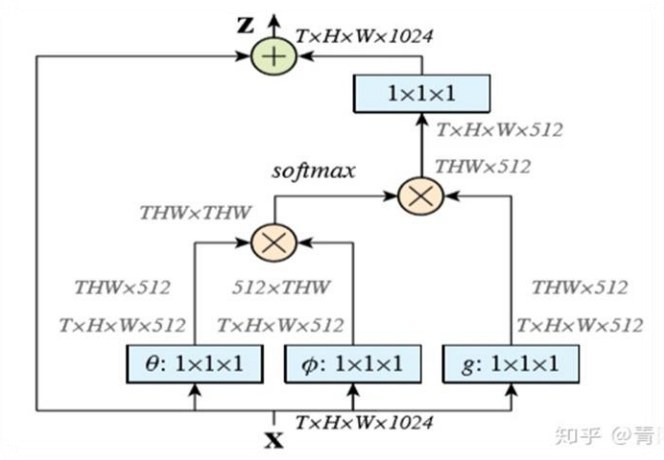
결론: **국지성 호우로 인한 강우 관측 오류에 대해 강건한 성능을 보일 것으로 예상되며 현업 문제 해결에 기여할 수 있음**

개선방안

K-댐을 특정하여
더 많은 학습 데이터 확보

비정형 데이터까지 활용하는
Multimodal 모델 개발

강우 shift로 인한 소실 데이터를
실제 관측치로 대체



Non-Local Block을 추가하여
CNN의 Locality 개선

참고문헌

- Sequence to Sequence based LSTM (LSTM-s2s)모형을 이용한 댐유입량 예측에 대한 연구, 한희찬 외 3명, 2021
- 홍수량 예측 인공신경망 모형의 활성화 함수에 따른 영향 분석, 김지혜 외 5명, 2021
- 강우자료 형태에 따른 인공신경망의 일유입량 예측 정확도 평가, 김석현 외 5명, 2019
- 수문학적 예측을 위한 딥러닝기반 인공신경망의 최적화 알고리즘 비교: 남강댐 일유출량을 사례를 중심으로, 마샤 모라디 외 1명, 2018
- WAVENET: A GENERATIVE MODEL FOR RAW AUDIO, Aaron van den Oord 외 8명, 2016
- 입력 강우 형태에 따른 수위예측 회귀모형의 성능 비교분석, 최승용 외 1명, 2011
- 실측자료를 이용한 하천의 수위변화 분석 - 강원도를 사례로 -, 배선학, 2011
- 신경망을 이용한 낙동강 유역 홍수기 댐유입량 예측, 윤강훈 외 2명, 2004
- 머신러닝 모델링의 흔한 실수들, 이제현, 2021
- http://taewan.kim/post/cnn/
- http://dmqm.korea.ac.kr/activity/seminar/242

감사합니다

팀명 : 범호

이성범 (팀장) : 2712qwer@naver.com

김주호 : jooho991122@gmail.com