

# BIGSCOIN

Explainable Bitcoin Pattern Alert and Forecasting Service



# Contents

**01 Motivation**

**02 Process**

**03 Data**

**04 Classification**

**05 Regression**

**06 Web**

**07 Improvement**

# Motivation

# Motivation



투자 수익률을 높이기 위해  
투자자는 차트를 본다

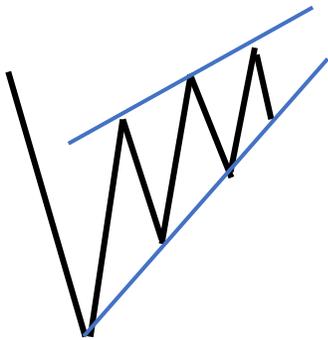
Why?

매매하기 적합한 패턴을  
찾아 투자를 하기 위함

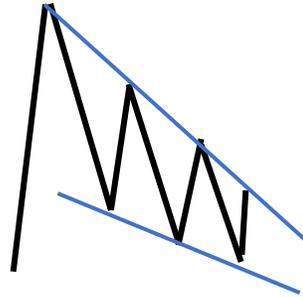
# Motivation

매매하기에 적합한 패턴이란?

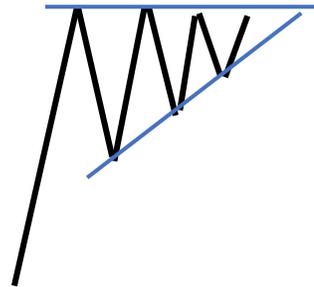
Rising Wedge



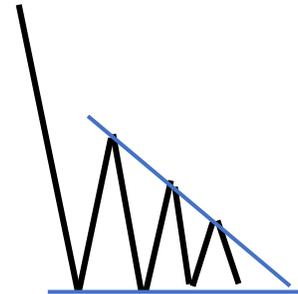
Falling Wedge



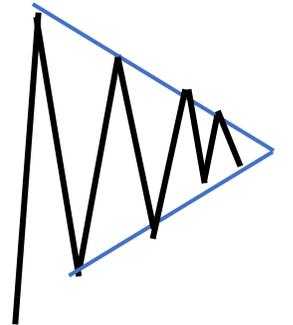
Ascending Triangle



Descending Triangle



Symmetric Triangle



# Motivation

바쁜 현대인들을 위해 해당 패턴이  
나타나는 시점을 자동으로 알려주자!

**패턴 인식**

해당 패턴 발생 시  
알람 제공

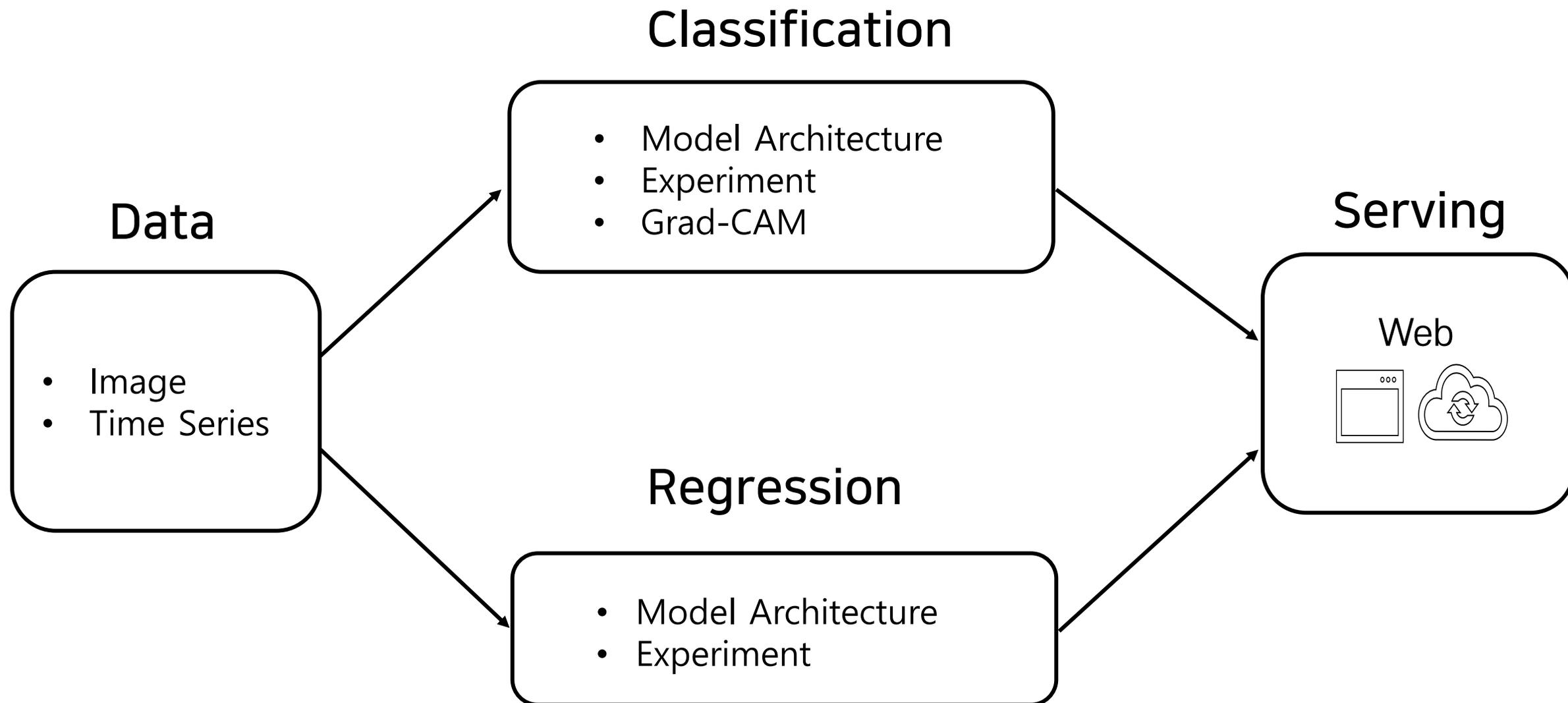
+

**코인 방향성 예측**

코인 가격이 상승할지  
하락할지 예측 지표 제공

# Process

# Process



**Data**

# Data

## Pyupbit API

- 기간: 2017년 8월 ~ 2022년 5월
- 5분봉 시가, 고가, 저가, 종가 데이터를 추출

## 학습 데이터 구축

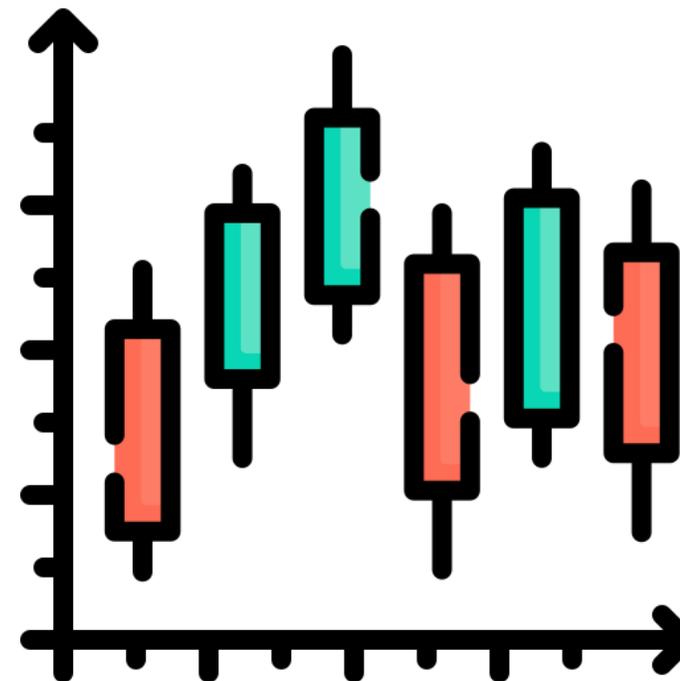
- 캔들스틱 차트에서 패턴의 시작점, 끝점, 패턴 종류 라벨링

The logo for UPbit, featuring the text "UPbit" in a bold, white, italicized sans-serif font. The "U" and "P" are connected, and the "i" has a dot. The logo is set against a dark blue rectangular background.

# Data

## 데이터 라벨링 예시

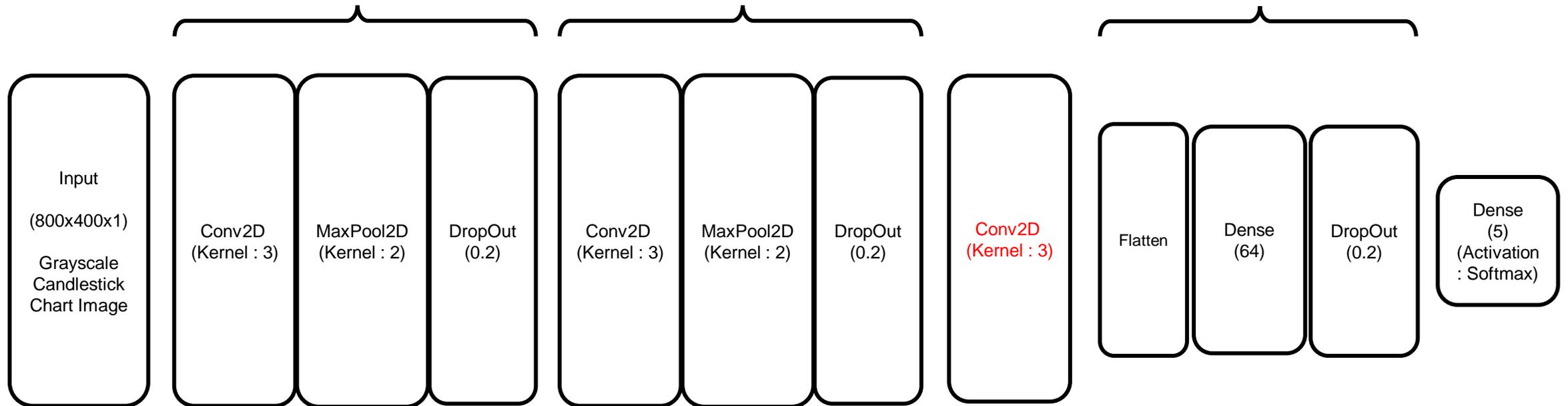
Start	End	Type
2018-10-16 08:30:00	2018-10-16 12:10:00	sym



# Classification

# Classification

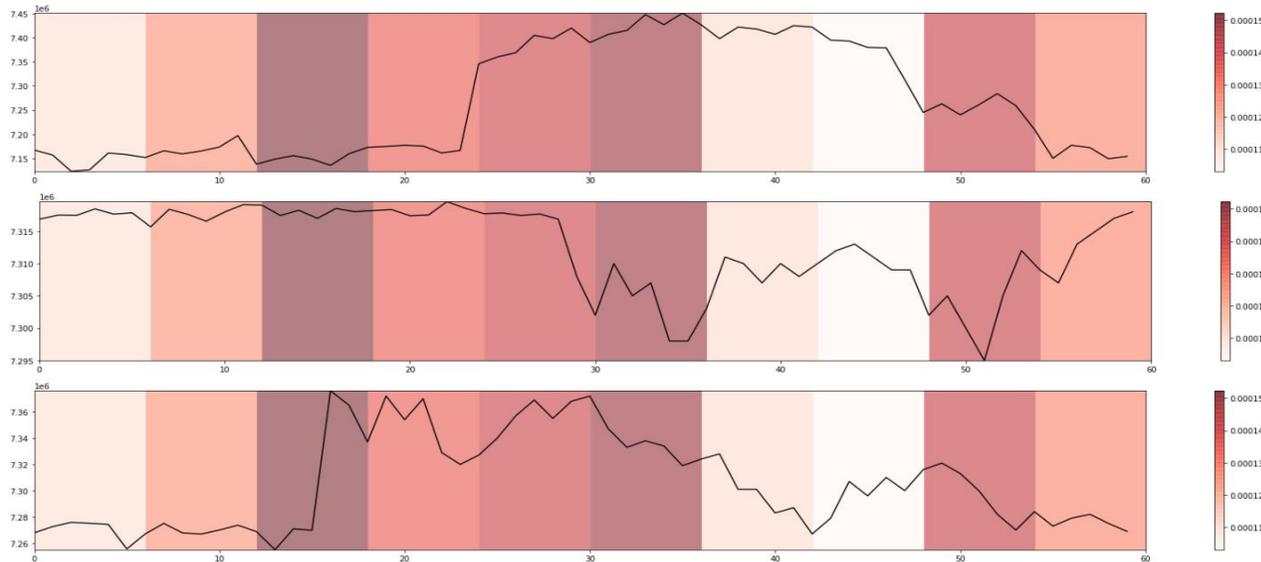
## Model Architecture



# Classification

## Experiment

- Chart Pattern recognition with Deep Learning (2018)
- Time Series를 Input으로 하는 LSTM-Conv1d 모델 구축 및 Grad-CAM 적용
- 마지막 Convolutional layer의 (feature map) gradient가 Input에 관계 없이 동일
- Train Dataset에서 가장 비중이 높은 label을 출력함 (local optimum)



LSTM-Conv1d에 Grad-CAM을 적용한 결과

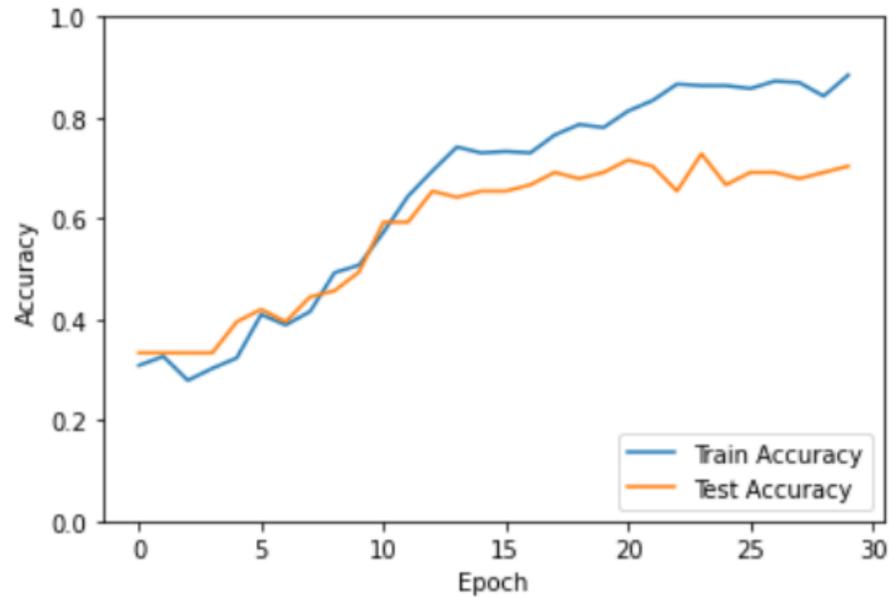


Image를 Input으로 하는  
Conv2d 모델 적용

# Classification

## Experiment

- Image : Conv2d
- Best Test Accuracy : 70.3704%



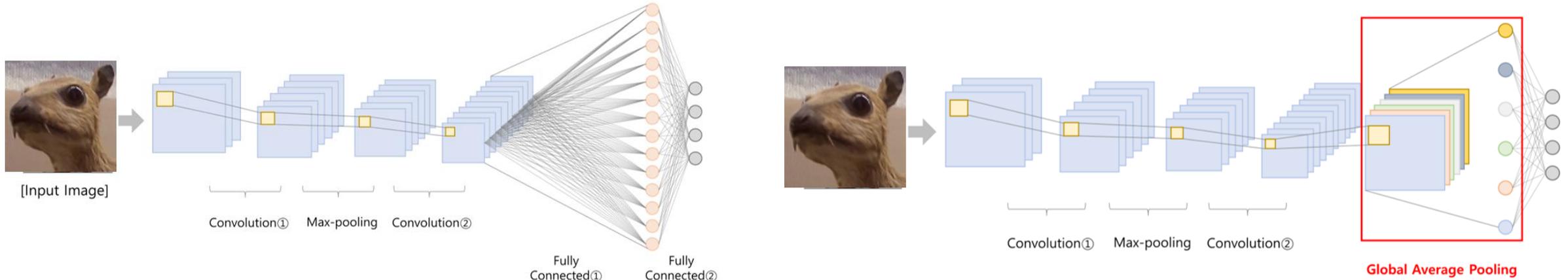
```
# Report Test Accuracy
```

```
test_loss, test_acc = model.evaluate(valDatagen)  
print(f'Test Accuracy : {round(100*test_acc, 4)}%')
```

```
6/6 [=====] - 1s 83ms/step - loss: 2.5675 - accuracy: 0.7037  
Test Accuracy : 70.3704%
```

# Classification

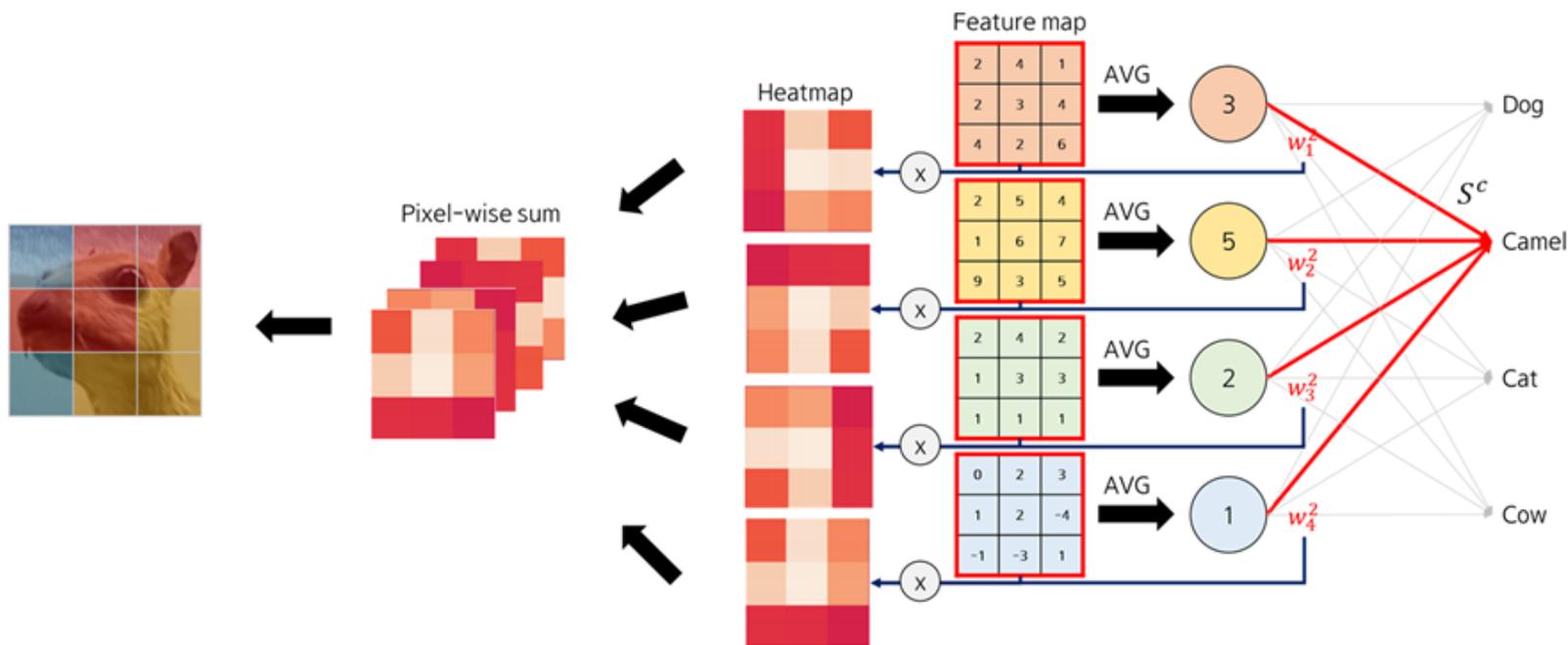
## CAM & Grad-CAM



- CAM(Class Activation Map) & Grad-CAM(Gradient-weighted CAM)
- CNN 기반 모델 출력을 가시적으로 해석하기 위한 방법론  
(‘A technique for producing ‘visual explanations’ for decisions from a large class of CNN-based models’)
- CAM : Fully-Connected Layer 대신 Global Average Pooling Layer 도입하여 weight 보존 (구조적 제약 존재)
- Grad-CAM : Fully-Connected Layer를 두 개 사용하고, 미분(gradient)을 통해 특정 Feature Map이 class에 미치는 영향 도출

# Classification

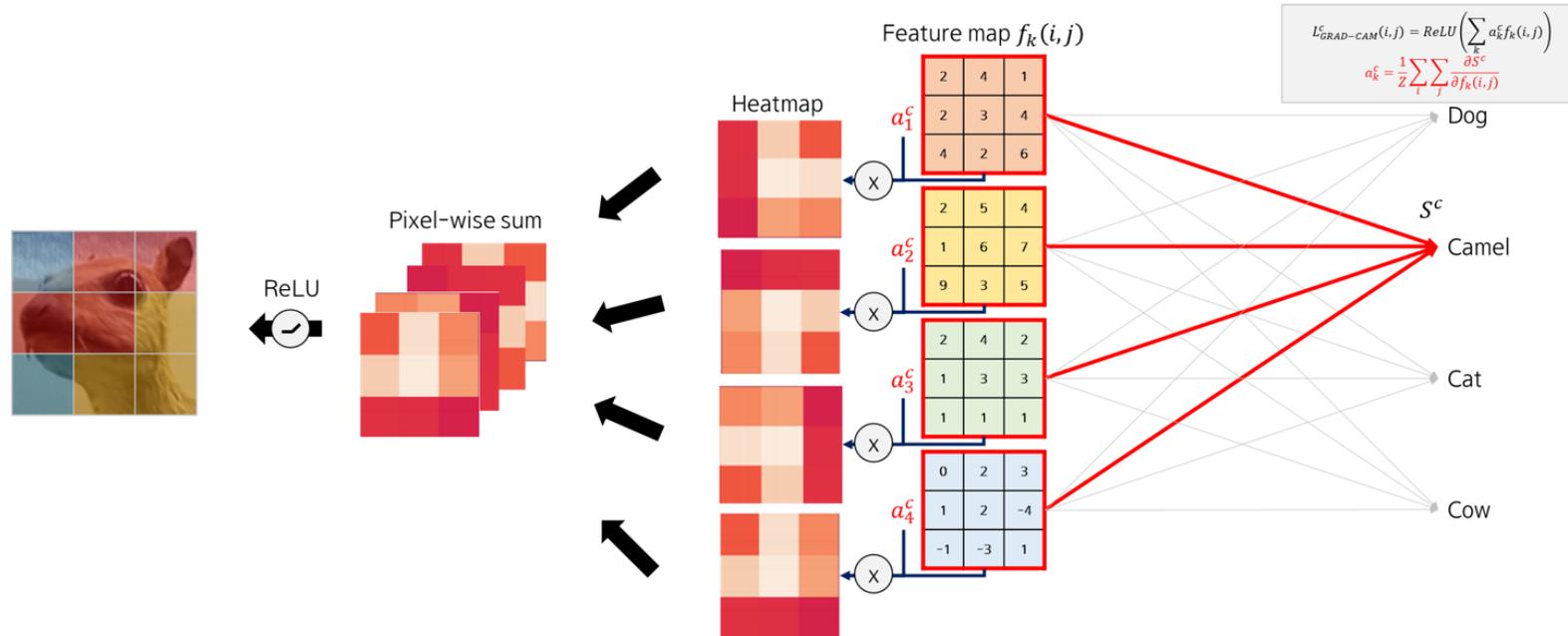
## CAM



$$L_{CAM}^c(i, j) = \sum_k w_k^c f_k(i, j)$$

# Classification

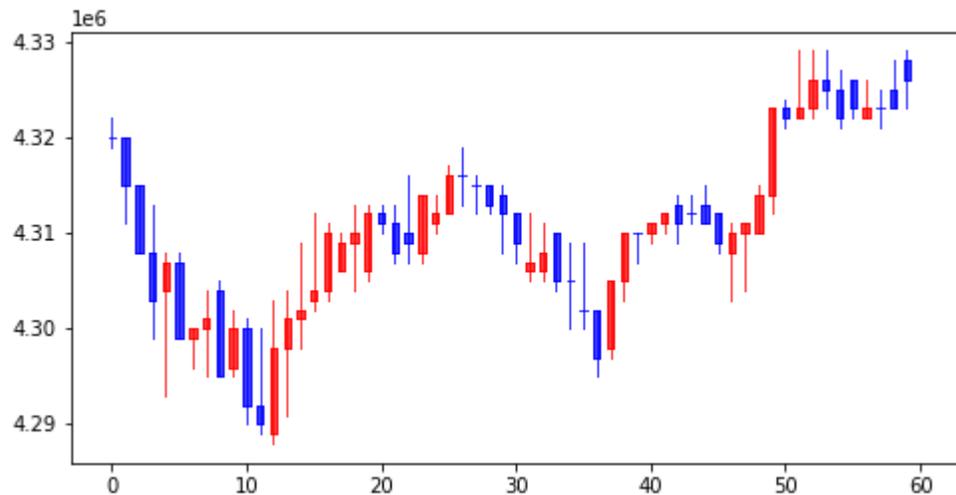
## Grad-CAM



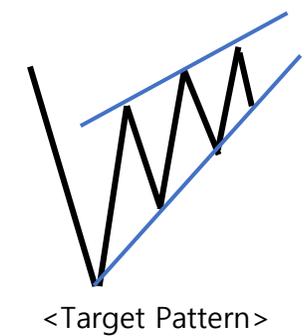
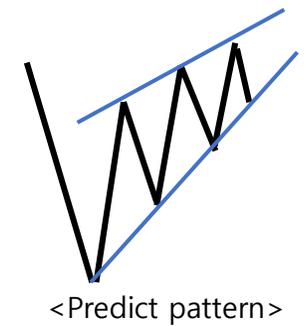
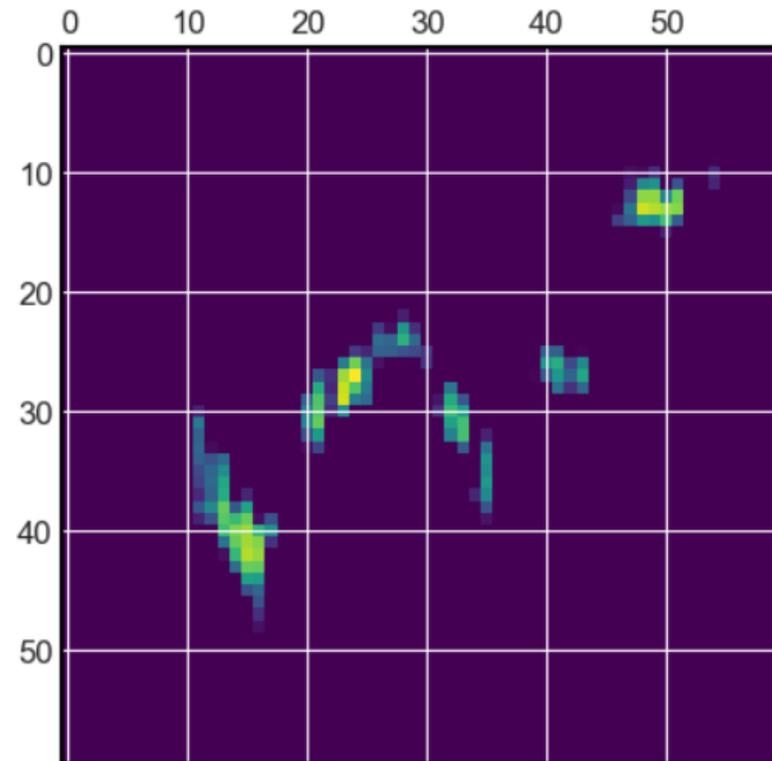


# Classification

Grad-CAM our case (1) : 예측이 올바른 경우

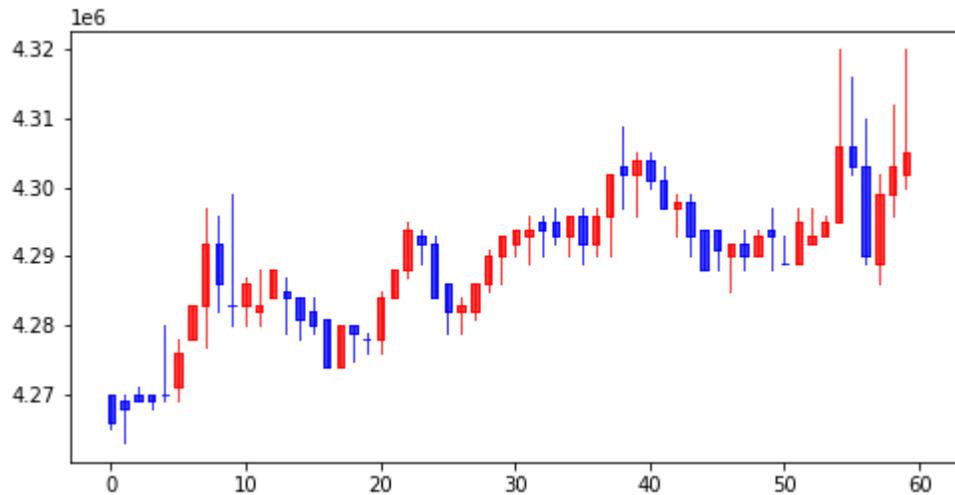


Plot : Plot 112.png | Predict : ris | Label : ris

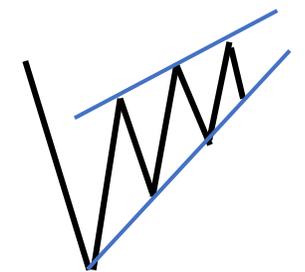
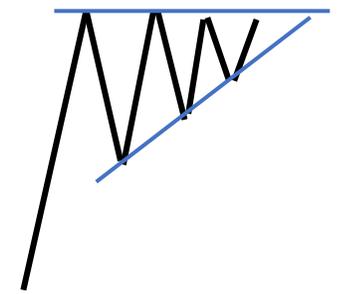
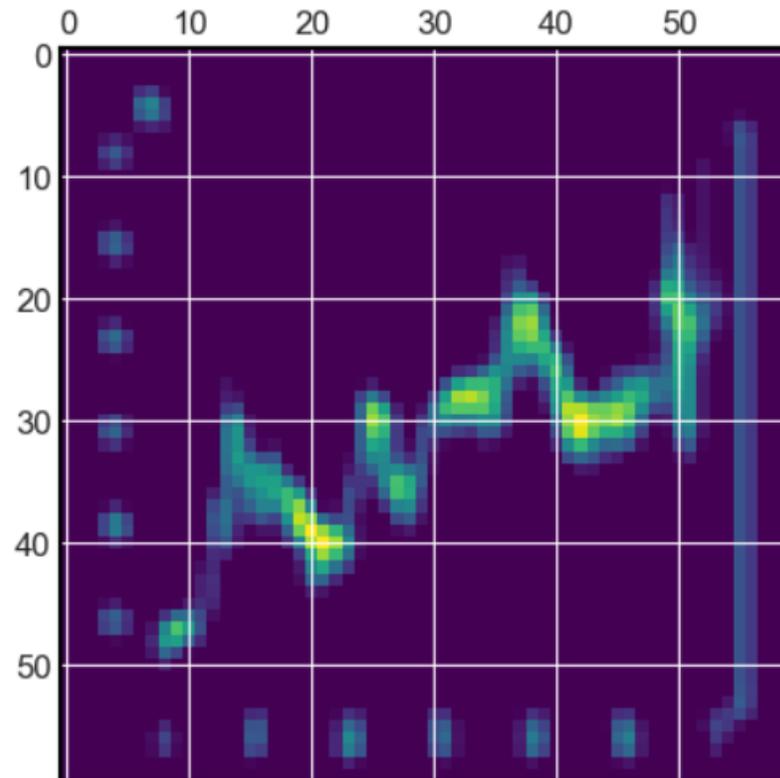


# Classification

Grad-CAM our case (2) : 예측이 틀린 경우

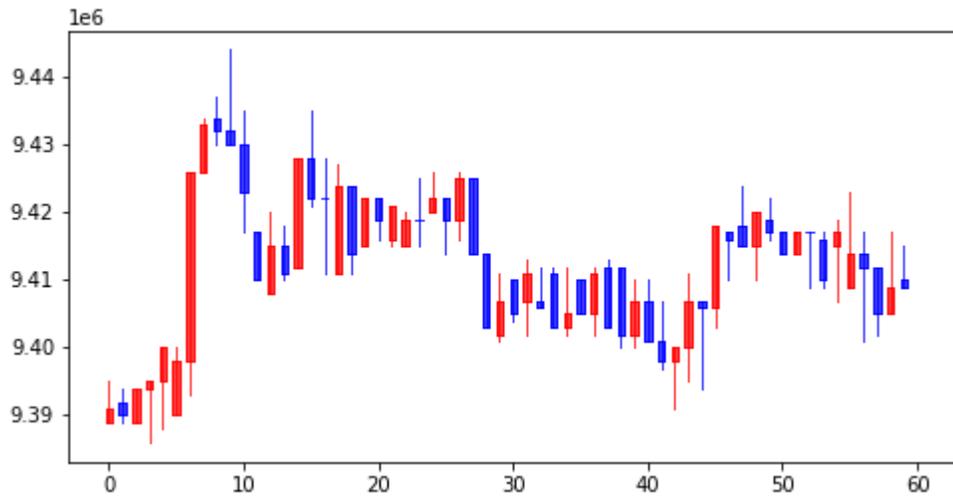


Plot : Plot 113.png | Predict : asc | Label : ris

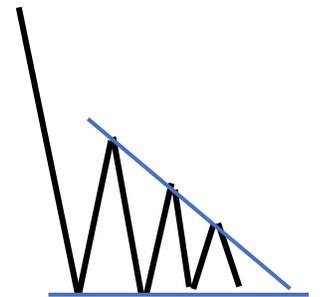
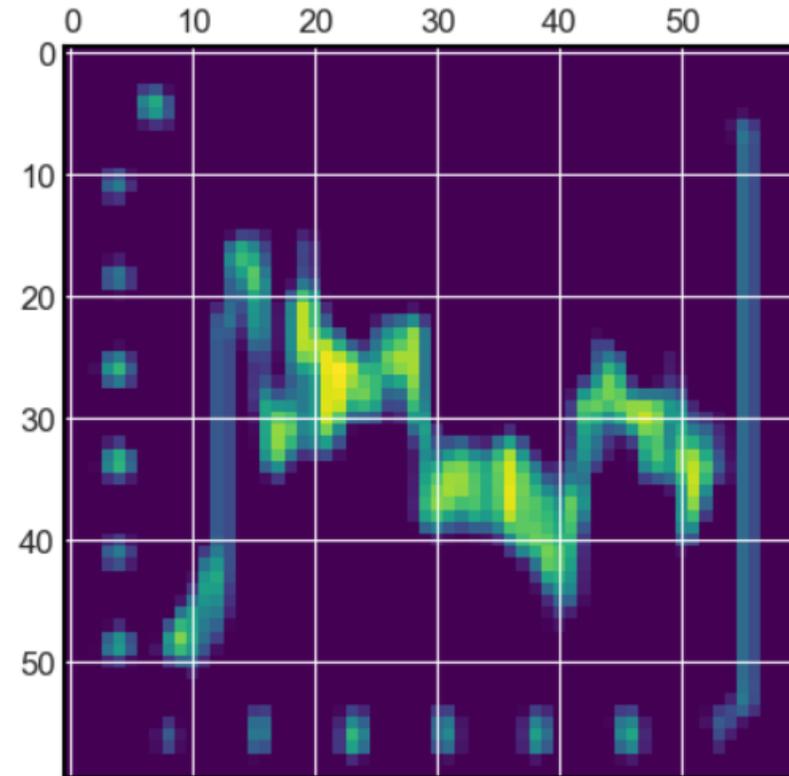


# Classification

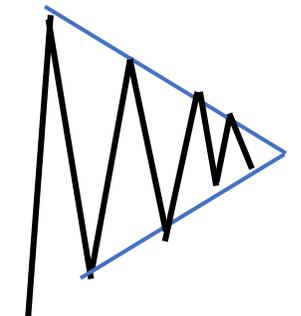
Grad-CAM our case (2) : 예측이 틀린 경우



Plot : Plot 12.png | Predict : des | Label : sym



<Predict pattern>

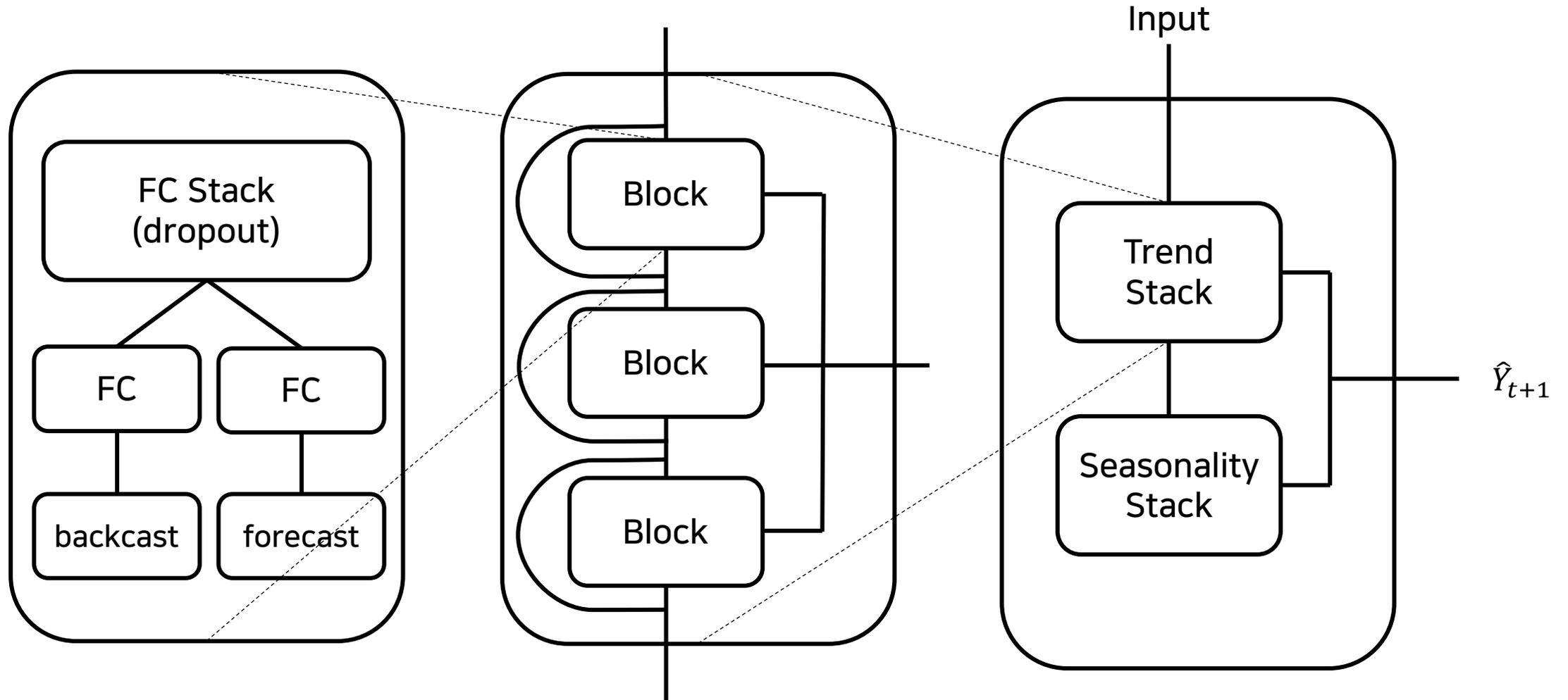


<Target Pattern>

# Regression

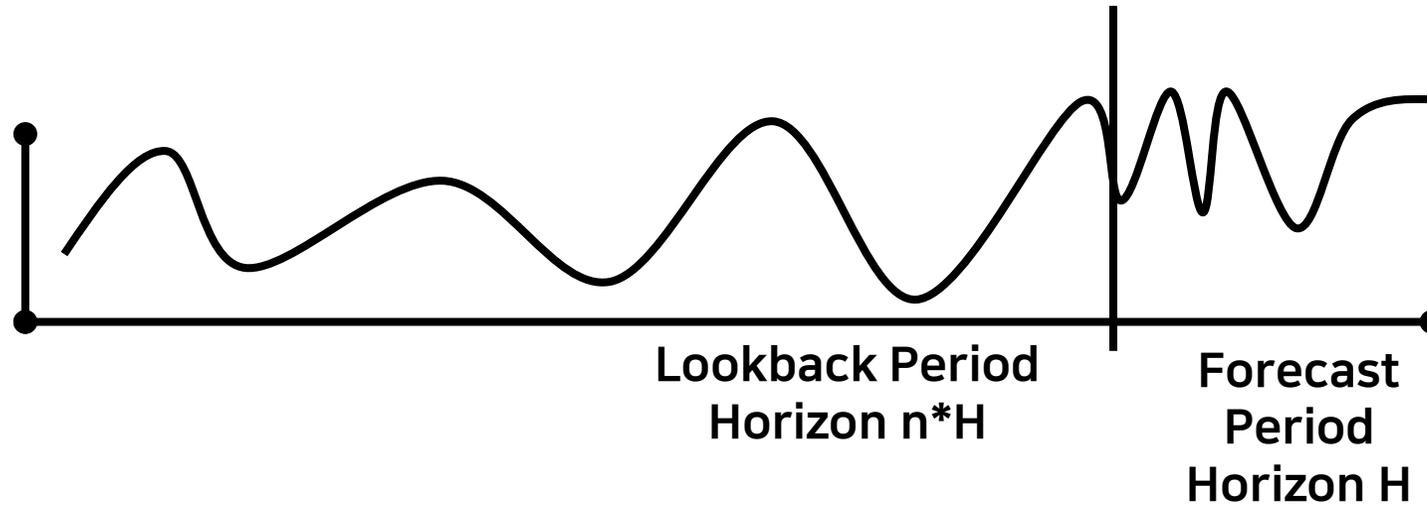
# Regression

## Model Architecture: NBEAT-S



# Regression

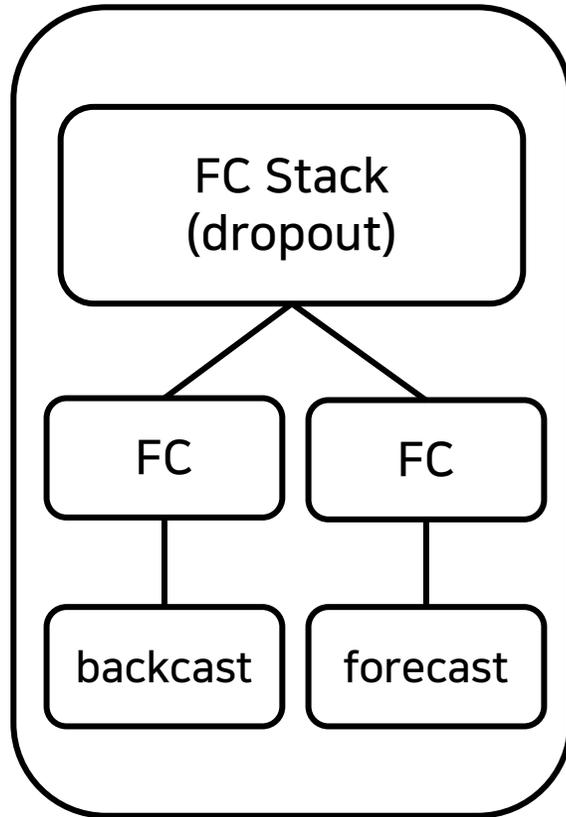
## NBEAT-S



- Trend 및 Seasonality로 분해하여 예측의 설명성을 제공하는 딥러닝 아키텍처
- 관측 시점  $t$ 로부터 길이  $H$  만큼의 prediction
- $n*H$  길이의 이전 시계열을 Input으로 활용

# Regression

## NBEAT-S: Block



- Trend는 단조 증가 또는 단조 감소의 성질을 지님.
- 이를 반영한 Output 생성을 위해 backcast, forecast 함수를 다음과 같이 지정

$$\hat{y}_t = T\theta_t^f$$

$$T = [1, t, \dots, t^p]$$

$$t = [0, 1, \dots, H-2, H-1]^T / H$$

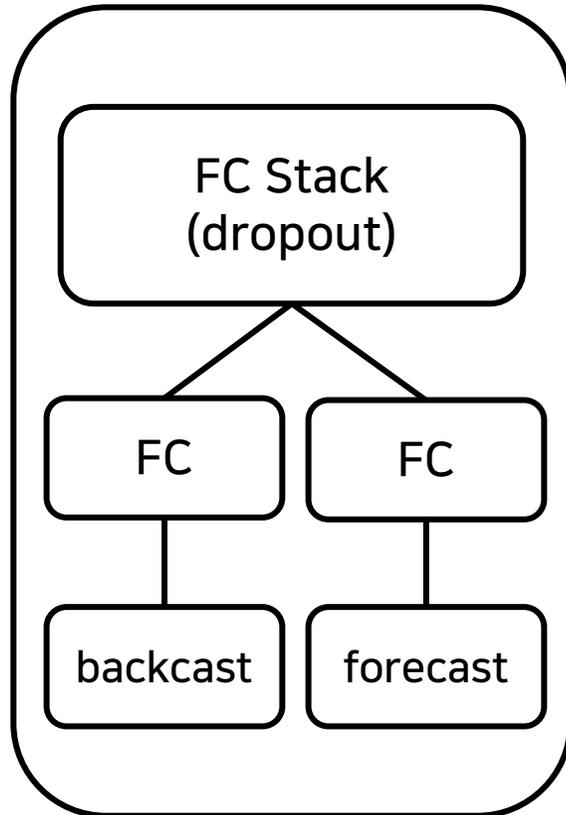
- Seasonality는 주기성을 지님.
- 푸리에 급수를 활용한 주기함수 형태로 backcast, forecast 함수 설정

$$\hat{y}_t = S\theta_t^f$$

$$S = [1, \cos(2\pi t), \dots, \cos\left(2\pi \left[\frac{H}{2} - 1\right] t\right), \sin(2\pi t), \dots, \sin\left(2\pi \left[\frac{H}{2} - 1\right] t\right)]$$

# Regression

## NBEAT-S: Block

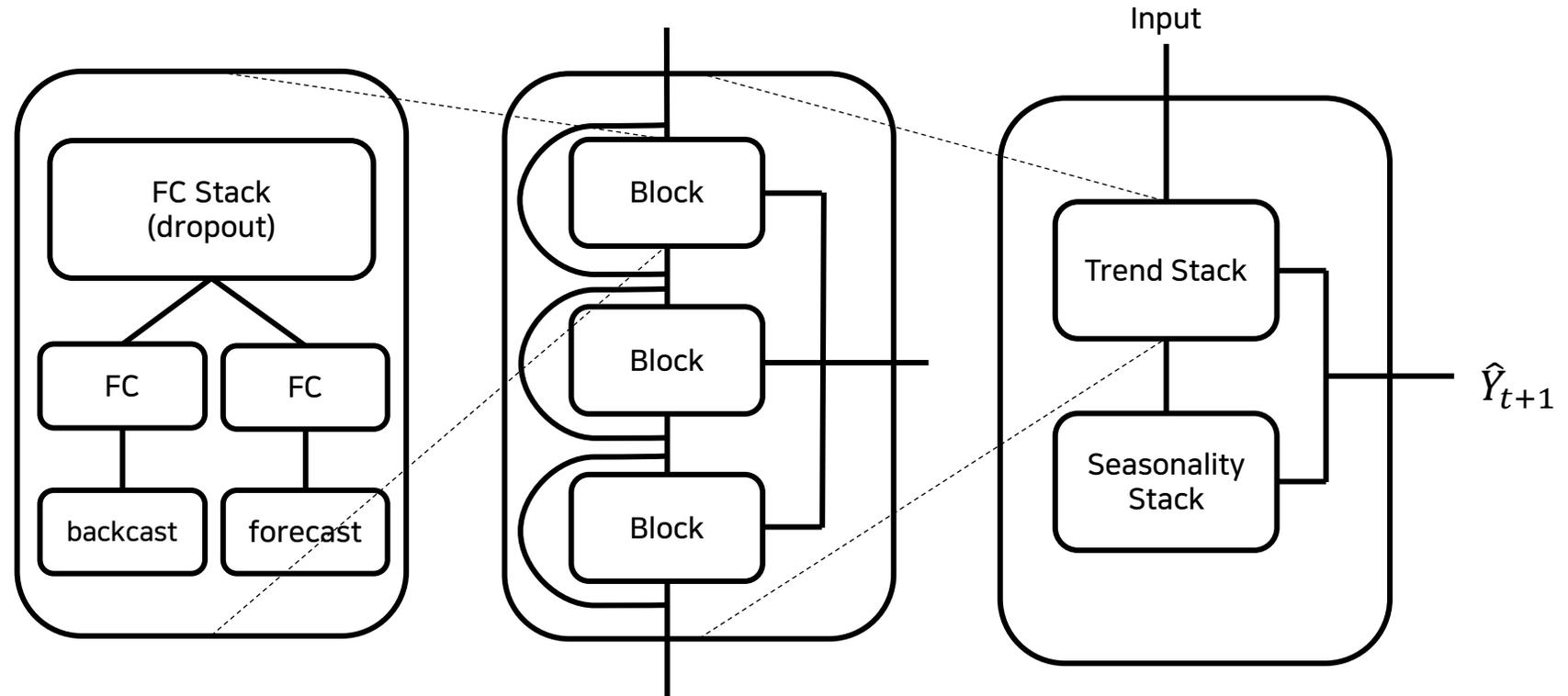


- Probabilistic Modeling : Drop-out을 통해 예측할 때마다 랜덤한 예측 값을 도출
- 신뢰구간 추정: 예측 값의 모평균을 신뢰수준 95% 하에서 구간 추정
- 50번의 Sampling을 사용, 정규분포 근사 후 신뢰구간 계산

$$\bar{X} - z_{\frac{\alpha}{2}} \frac{s}{\sqrt{n}} \leq \mu \leq \bar{X} + z_{\frac{\alpha}{2}} \frac{s}{\sqrt{n}}$$

# Regression

## NBEAT-S: Stack



- Block의 Backcast와 Forecast: Stack의 Output을 구성하거나 다음 Block Input으로 활용
- Backcast Residual Connection: 이전 Block이 Input의 일부 signal을 제거함으로써 다음 Block의 예측력 개선
- Forecast Summation Connection: Input을 계층적으로 분해하는 효과 (Interpretability)
- Stack Connection: 시계열의 trend 성분을 분해하고 남은 feature를 seasonality 성분에 사용해 시계열의 특성을 고려한 예측이 가능

# Regression

## NBEAT-S: Hyperparameter

Lookback	Forecast	Stride	Train-test ratio	Batch size	Hidden layer units	Drop-out rate	Learning rate	Epochs
144	48	1	9:1	64	128	0.2	0.001	200

한 종목 (KRW-BTC) 종가에 대한 시계열 패턴을 학습 (univariate)

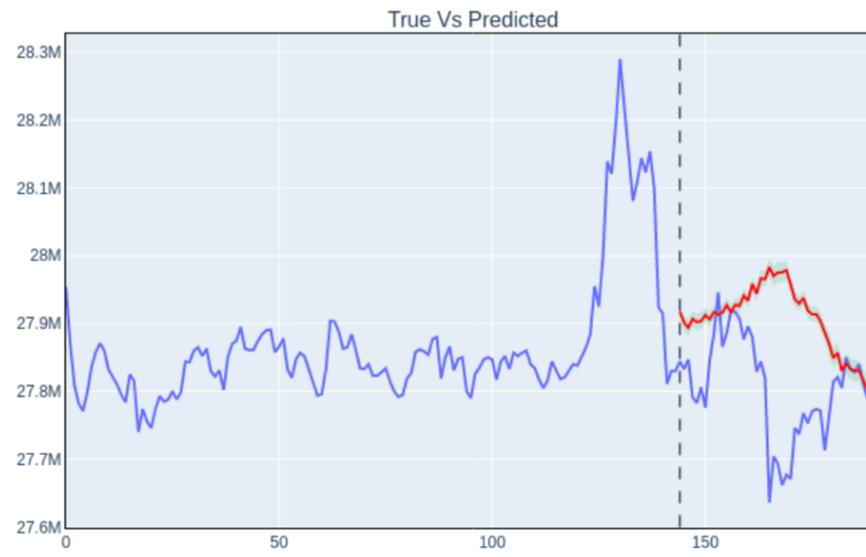
# Regression

## Experiment

- BTC 종목의 증가를 예측하는 univariate forecasting task
- train dataset size(20000개, 2500개)에 따라 실험



20000개



2500개

train dataset size 20000개의 경우 예측이 평균에 수렴,  
5분봉 2500개 data 최종 결정

# Regression

## Experiment

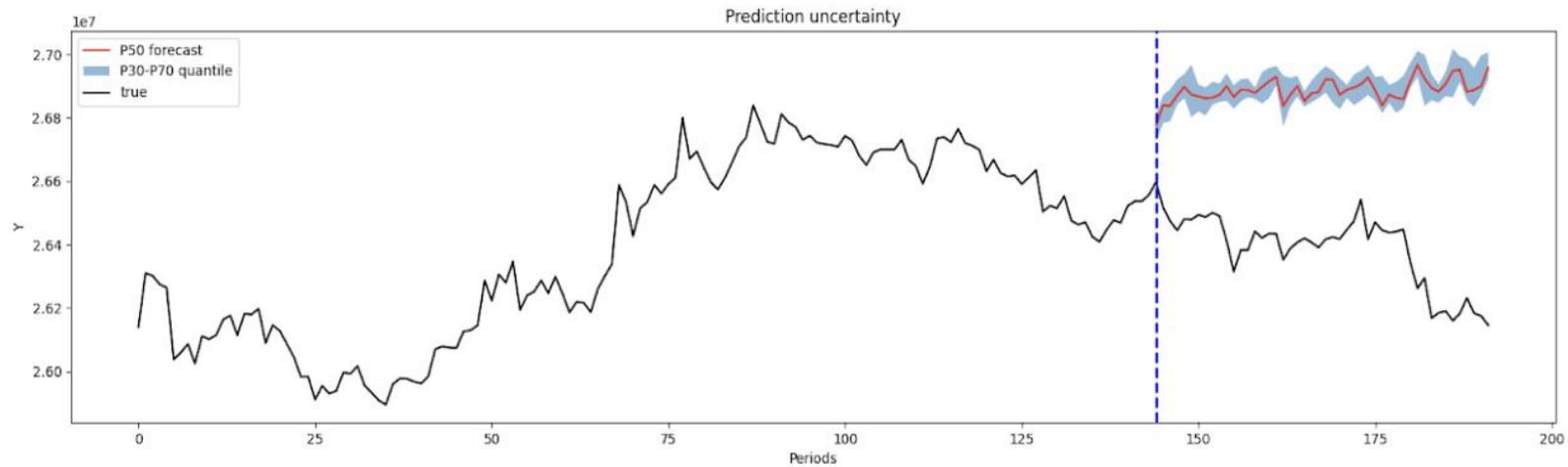
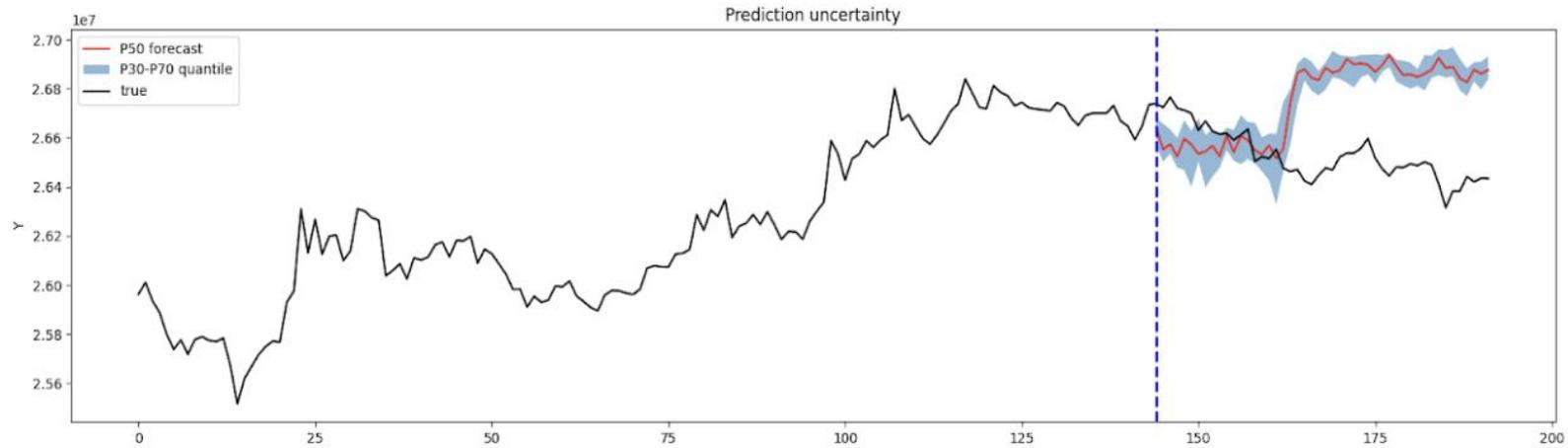
$$sMAPE = \frac{200}{H} \sum_{i=1}^H \frac{|y_{T+i} - \hat{y}_{T+i}|}{|y_{T+i}| + |\hat{y}_{T+i}|}$$

$$MAPE = \frac{100}{H} \sum_{i=1}^H \frac{|y_{T+i} - \hat{y}_{T+i}|}{|y_{T+i}|}$$

모델의 불확실성을 반영할 수 있는 DeepAR모델과  
Long Sequence Time series Forecasting (LSTF)에  
최적화된 Informer 모델의 test 예측 개형 및 성능 비교(MAPE, sMAPE, MAE)

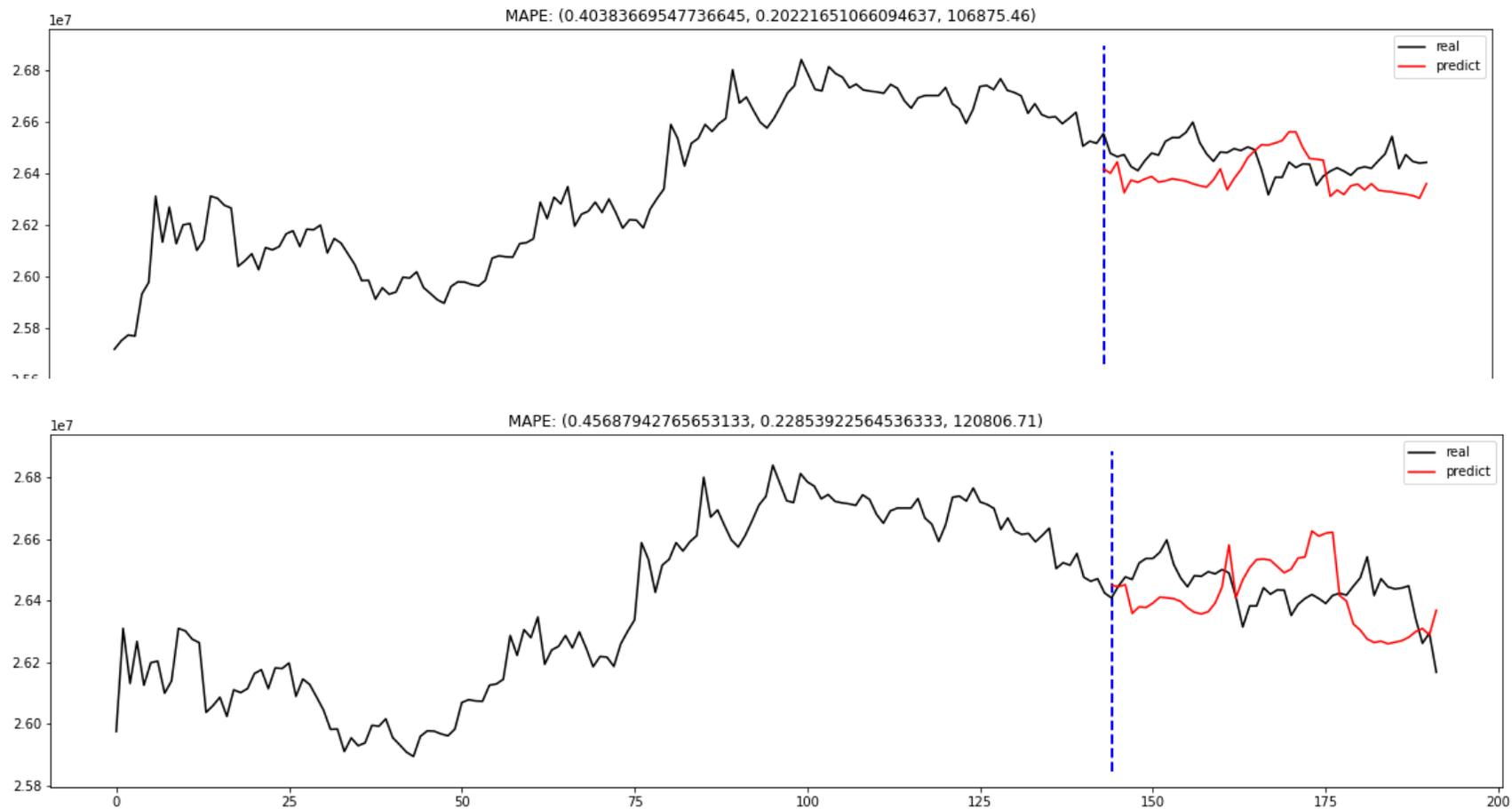
# Regression

## Experiment: DeepAR



# Regression

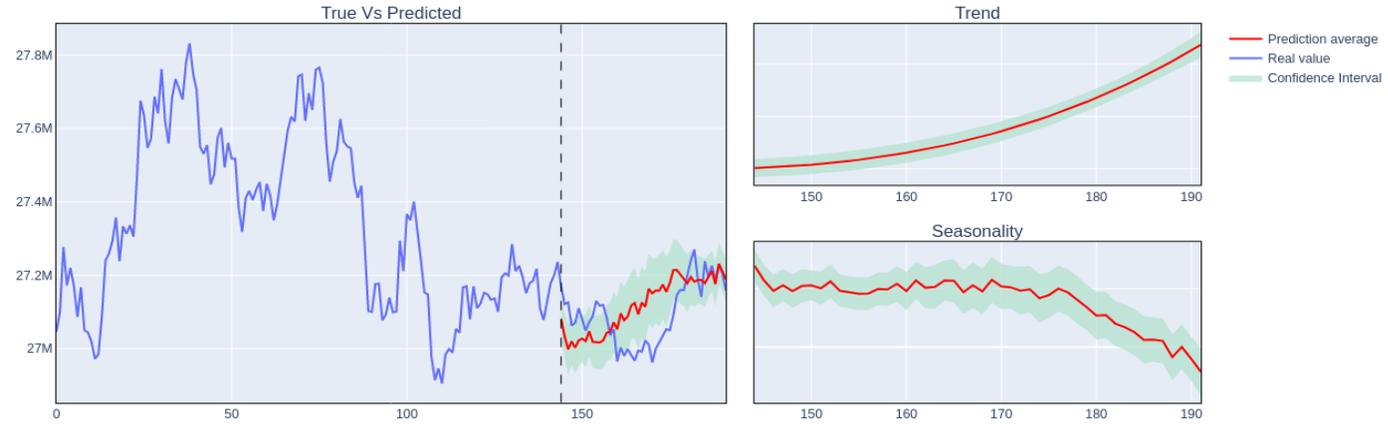
## Experiment: Informer



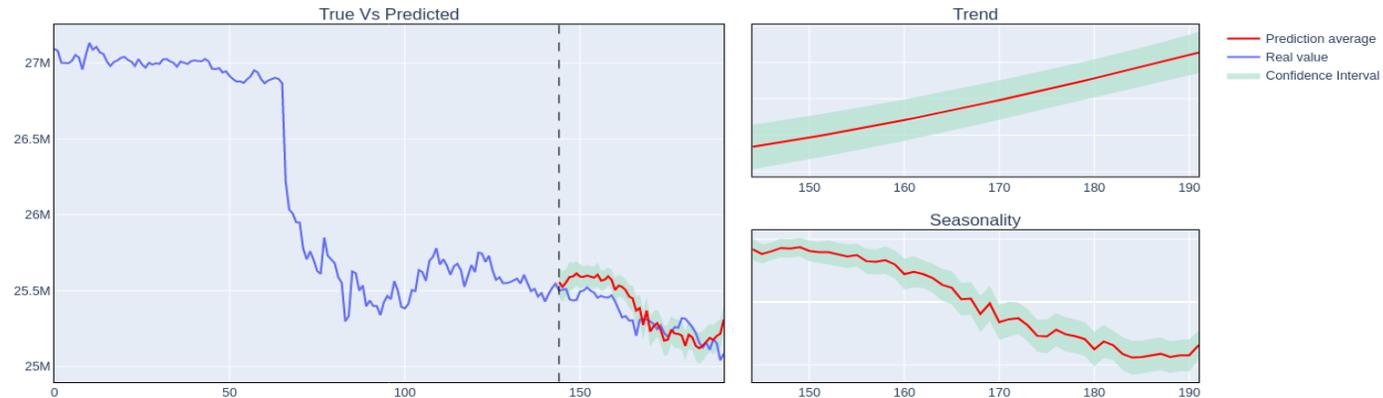
# Regression

## Experiment: NBEAT-S

Bitcoin Price Prediction



Bitcoin Price Prediction

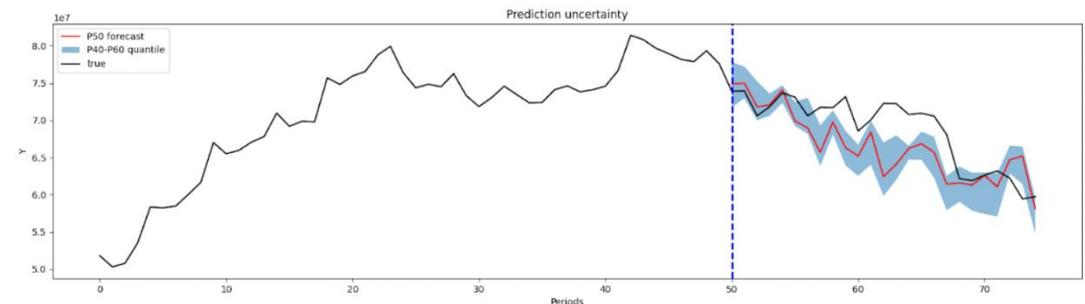
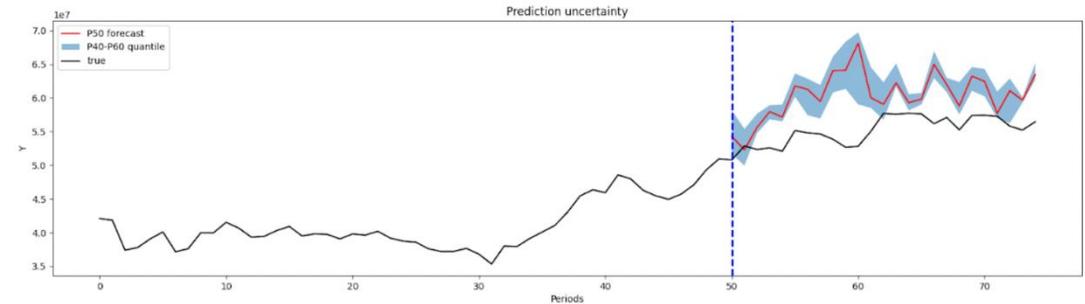


# Regression

## Experiment: NBEAT-S

	MAPE	sMAPE	MAE	Learning Time
Nbeats	0.656	0.326	173207	1'16"
DeepAR	1.808	0.89	476389	69'23"
Informer	0.475	0.237	125200	24'02"

- DeepAR 모델은 Amazon사에서 변동성과 스케일이 큰 제품 수요 예측을 위해 제안한 모델. granularity가 다른 5분봉 차트 데이터에 적합하지 않은 결과 도출.
- NBEAT-S 모델은 Informer 모델보다 성능은 조금 낮지만, 설명성을 제공하며 학습 효율 면에서 우위를 보임.



**Web**

# Web

×

Bigscoin

Services

## BIGSCOIN

Explainable Bitcoin Pattern Alert and Forecasting Service



### Motivation

주식 및 코인 투자자들은 차트에서 발생하는 패턴 중 매매하기 적합한 시점의 패턴을 관측하며 투자를 진행합니다. BIGSCOIN은 매매에 적합한 패턴이 등장하는 것을 자동으로 감지하여 투자자에게 유용한 정보를 제공합니다.

### Dataset

Pyupbit 패키지를 통해 2017년 8월부터 2022년 5월까지의 5분봉 시가, 고가, 저가, 종가 데이터를 추출하여 학습에 사용하였습니다.

### Pattern Recognition Forecasting Prices

Rising wedge, falling wedge와 같은 매매에 적합한 패턴 5가지를 탐지하기 위해 2D CNN을 사용하여 분류를 시행합니다.

경향성(Trend)과 계절성(Seasonality)과 같은 시계열 특징을 분석하는 해석 가능한 딥러닝 아키텍처인 Nbeats를 학습하여 향후 4시간을 예측합니다.

Developed By ToBigs 16&17 Time Series & XAI Team

김권호 김상윤 김윤혜 김주호 김현태 나세연 박한나 유현우 이예림

## Model page

### Select patterns

알림 받을 패턴을 선택해주세요. (다중 선택 가능)

- Rising Wedge
- Falling Wedge
- Ascending Triangle
- Descending Triangle
- Symmetric Triangle

## Model page

### Pattern Recognition

Rising Wedge, Falling Wedge와 같은 매매에 적합한 패턴 5가지를 탐지하기 위해 2D CNN을 사용하여 분류를 시행합니다.

pattern explanation +

### Bitcoin Prices



# Web

Bigscoin  
Services

Model page

Select patterns

알림 받을 패턴을 선택해주세요. (다중 선택 가능)

- Rising Wedge
- Falling Wedge
- Ascending Triangle
- Descending Triangle
- Symmetric Triangle

## XAI: Grad-CAM



Symmetric Triangle 패턴과 가장 유사합니다.

Show raw data

# Web

×

☰

Bigcoin  
Services

## Model page

### Select patterns

알림 받을 패턴을 선택해주세요. (다중 선택 가능)

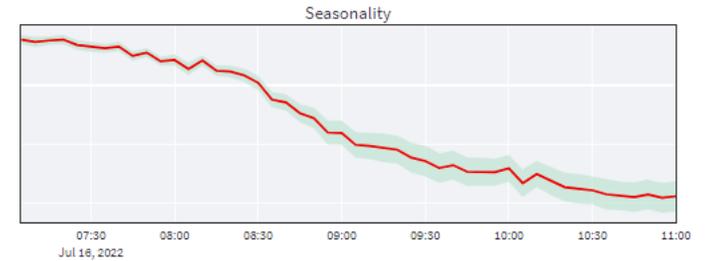
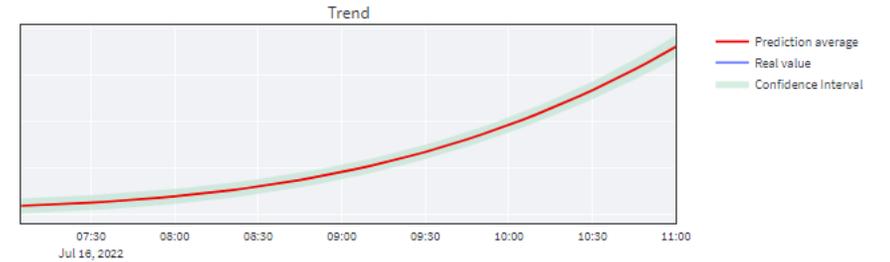
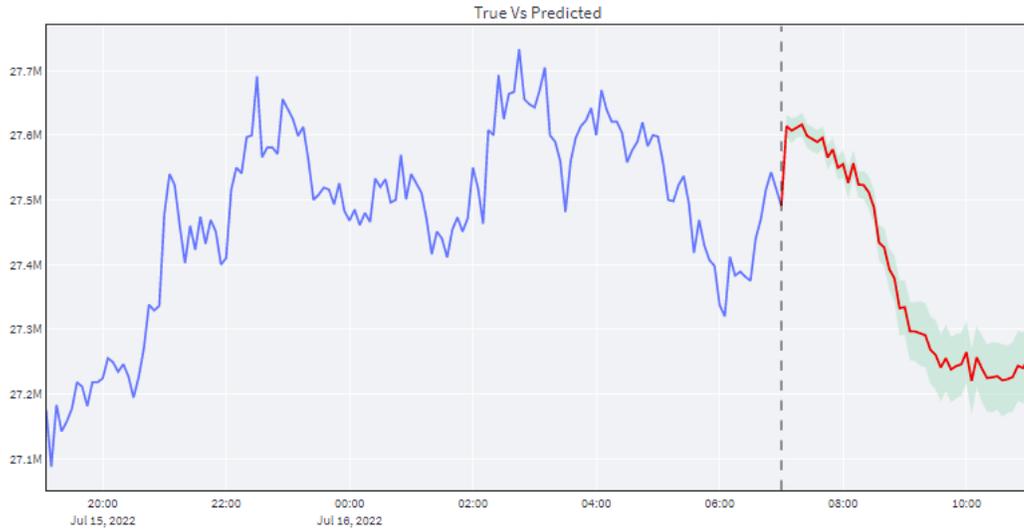
- Rising Wedge
- Falling Wedge
- Ascending Triangle
- Descending Triangle
- Symmetric Triangle

Symmetric Triangle 패턴과 가장 유사합니다.

Show raw data

## Forecasting Bitcoin Price

경향성(Trend)과 계절성(Seasonality)과 같은 시계열 특징을 분석하는 해석 가능한 딥러닝 아키텍처인 Nbeats를 학습하여 향후 4시간을 예측합니다.



**Improvement**

# Improvement

## Classification

- 'None' class 추가하여 성능 개선 가능
- Clean data의 부재

## Regression

- 5분봉 데이터 2500개를 일정 주기를 가지고 지속적으로 재학습한 경우가 제일 효과적임을 확인
- 재학습 가능한 모델을 배포하여 예측치를 보다 정확하게 개선 가능
- 비트코인 가격 외 파생변수들을 고려할 수 있도록 NBEAT-S 모델 개선

## Web

- 비트코인 외 타 종목도 선택하여 서비스를 제공할 수 있도록 배포
- 모바일에서 푸시 알림을 받을 수 있도록 서비스 개선

# 팀원 소개

# 팀원 소개 (분류 팀)



16기 이예림



17기 나세연



17기 유현우

# 팀원 소개 (회귀 팀)



16기 김주호



17기 김현태



17기 김상윤

# 팀원 소개 (웹 팀)



16기 김권호



16기 김윤희



16기 박한나

**감사합니다**