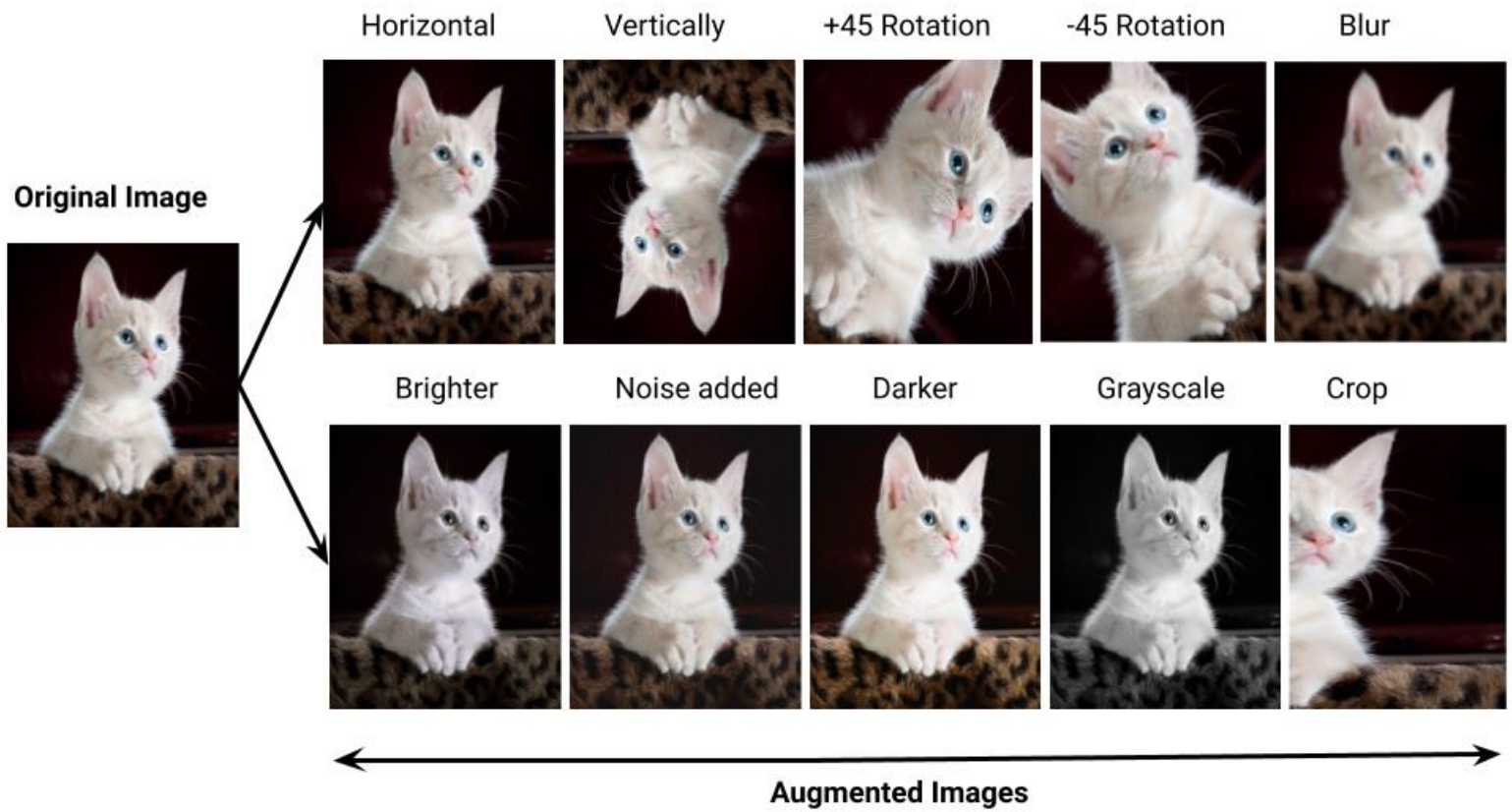
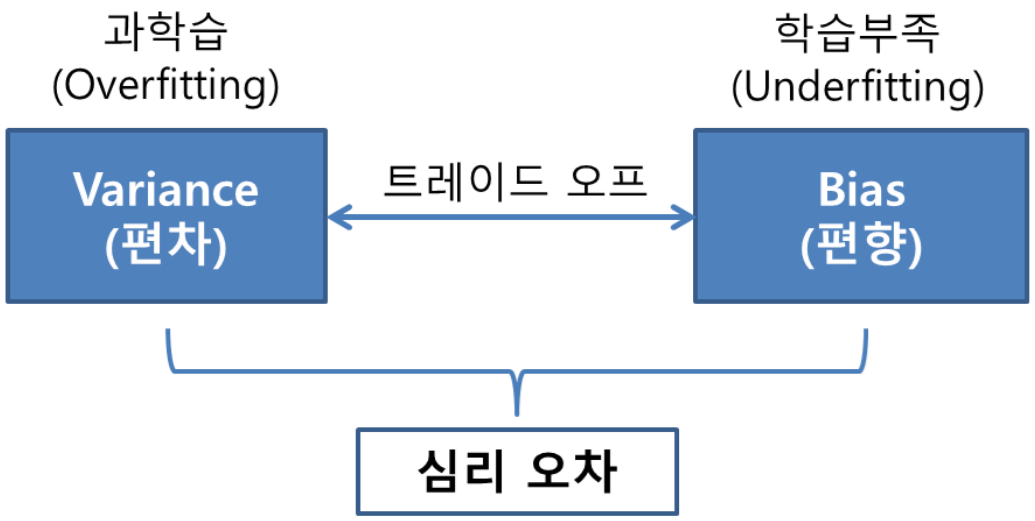
AI 실전 4주차

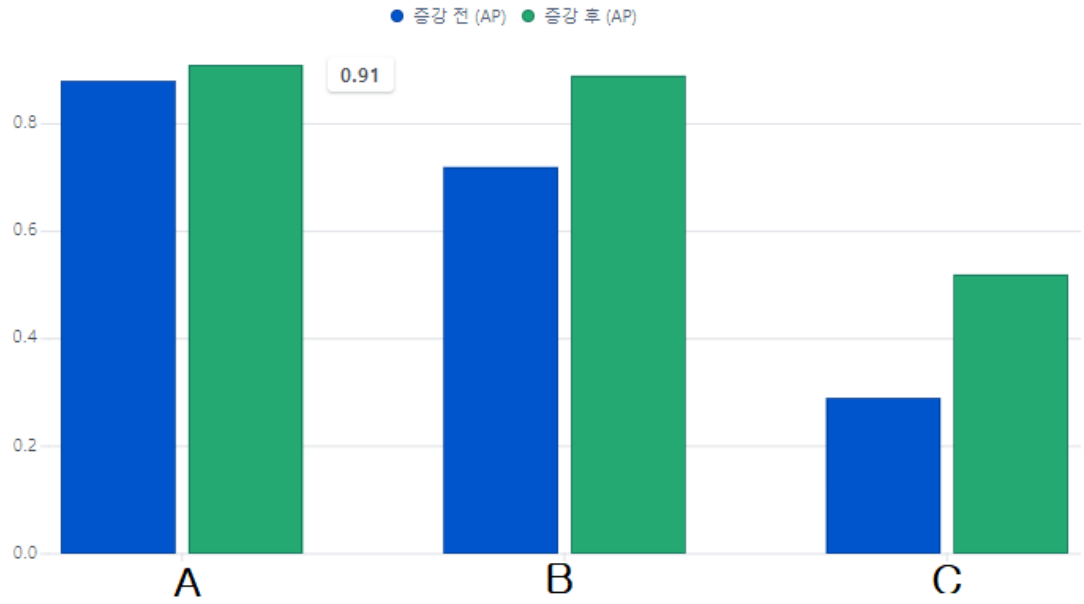
어떻게 데이터를 늘려볼까?
데이터 증강

데이터 증강?

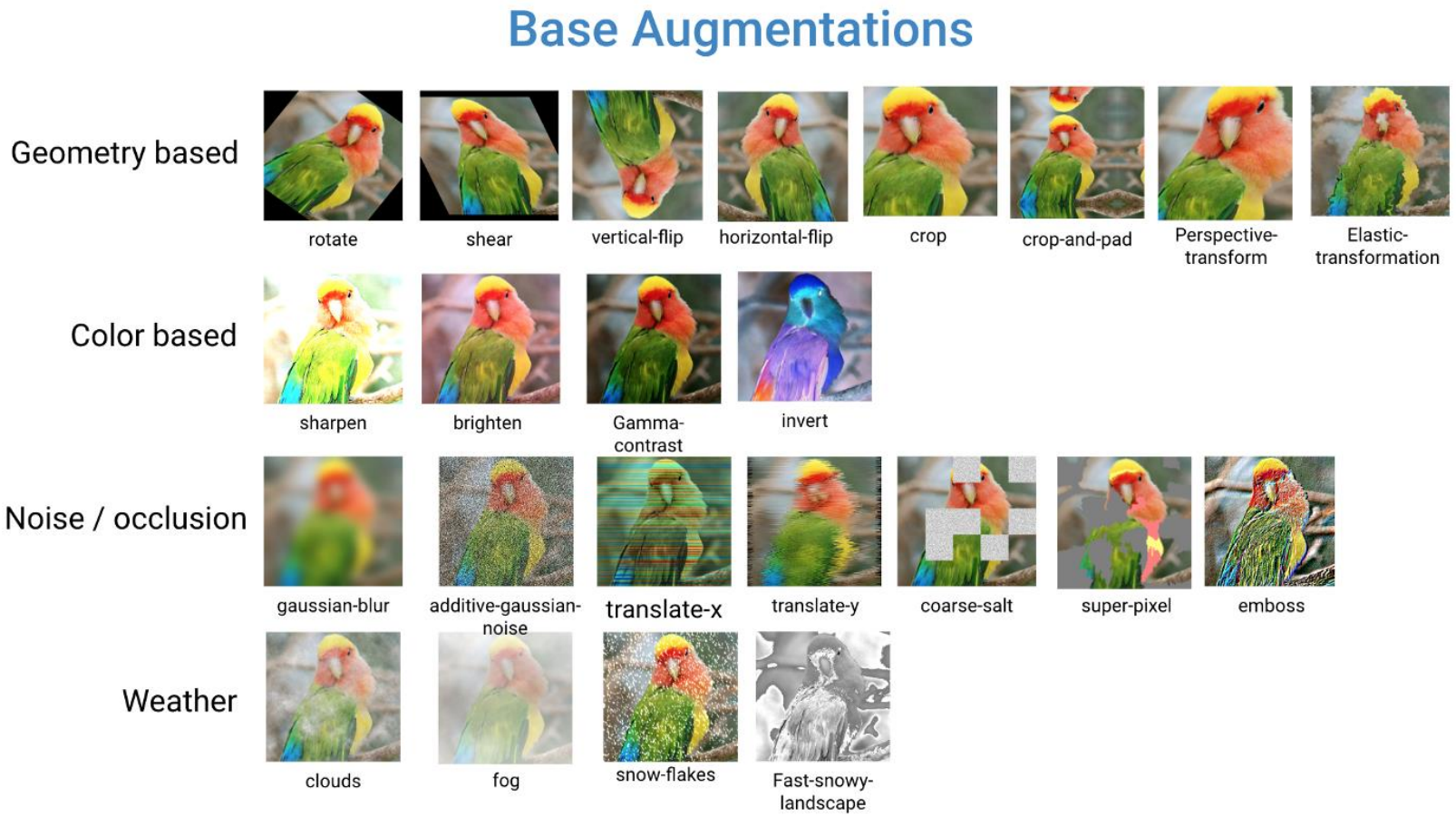

Original Image
Horizontal
Vertically
+45 Rotation
-45 Rotation
Blur
Brighter
Noise added
Darker
Grayscale
Crop
Augmented Images

Why? 데이터 증강 필요

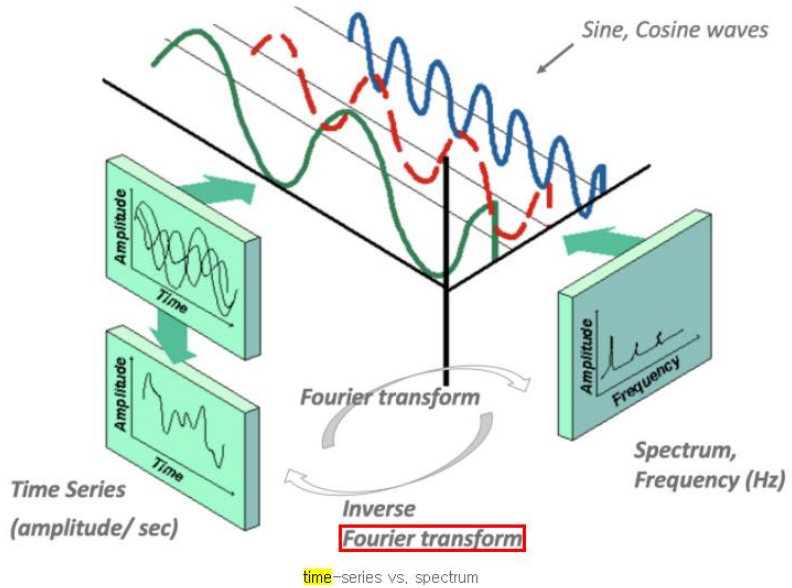

과학습
(Overfitting)
학습부족
(Underfitting)
Variance
(편차)
트레이드 오프
Bias
(편향)
심리 오차


증강 전 (AP)
증강 후 (AP)
0.91
0.8
0.6
0.4
0.2
0.0
A
B
C

데이터 증강 종류(이미지)


Base Augmentations
Geometry based
rotate
shear
vertical-flip
horizontal-flip
crop
crop-and-pad
Perspective-transform
Elastic-transformation
Color based
sharpen
brighten
Gamma-contrast
invert
Noise / occlusion
gaussian-blur
additive-gaussian-noise
translate-x
translate-y
coarse-salt
super-pixel
emboss
Weather
clouds
fog
snow-flakes
Fast-snowy-landscape

시계열 데이터 증강



time-series vs. spectrum

- 데이터의 일정 시간 간격으로 기록됨.
- 시간의 종속성으로 인해 데이터를 뒤죽박죽 조합하면 안된다!

시계열 데이터 증강 방법

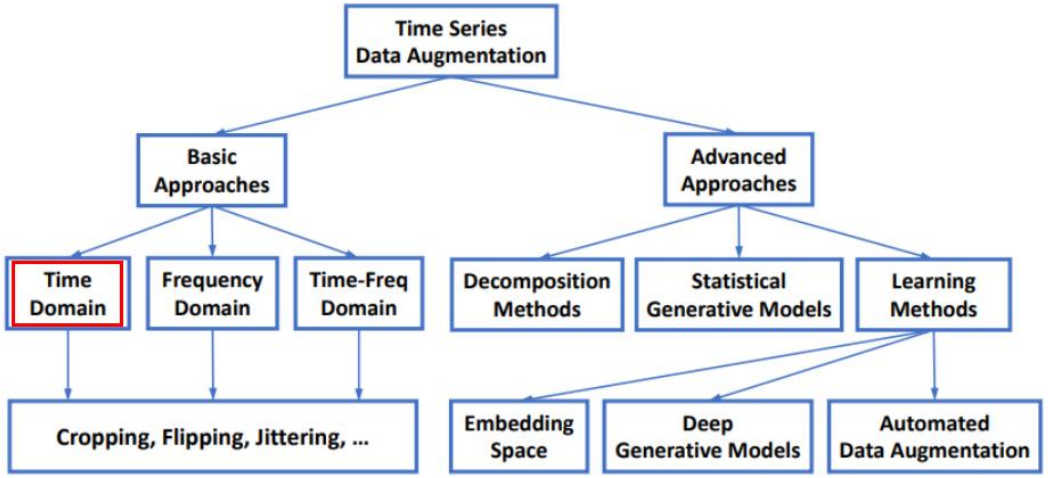


Figure 1: A taxonomy of time series data augmentation techniques.

- **Window cropping or slicing**
  - 원본 데이터에서 랜덤하게 연속적인 slice를 추출
  - 이미지를 랜덤하게 잘라 데이터를 증강하는 것과 유사
- **Window warping**
  - 랜덤한 구간을 선정하여 압축 or 확장하는 방법이다.
  - 원본 시계열의 전체 길이를 바꿀 수 있다.
- **Flipping**
  - 원본 데이터의 부호를 바꿔 새로운 시퀀스를 만드는 방법
  - 부호를 바꿔도 라벨 동일
- **Noise injection**
  - 라벨을 바꾸지 않은 채 원본 데이터에 노이즈를 삽입
  - 가우시안 노이즈(정규분포를 갖는 잡음) 또는
  - spike, step-like trend, and slope-like trend 등 적용
- **Label expansion**
  - 이상치 감지를 위한 증강 기법이다.
  - 거리와 값 거리가 이상치와 가까운 이상치로 분류하여 라벨을 확장

자연어 데이터 증강

Operation	Sentence
None	A sad, superior human comedy played out on the back roads of life.
SR	A ***lamentable***, superior human comedy played out on the ***backward*** road of life.
RI	A sad, superior human comedy played out on ***funniness*** the back roads of life.
RS	A sad, superior human comedy played out on ***roads*** back ***the*** of life.
RD	A sad, superior human out on the roads of life.

Table 1: Sentences generated using EDA. SR: synonym replacement. RI: random insertion. RS: random swap. RD: random deletion.

- Synonym Replacement, SR
  - 문장에서 랜덤으로 stop words가 아닌 n 개의 단어들을 선택해 임의로 선택한 동의어들 중 하나로 바꾸는 기법.
- Random Insertion, RI
  - 문장 내에서 stop word를 제외한 나머지 단어들 중에서, 랜덤으로 선택한 단어의 동의어를 임의로 정한다. 그리고 동의어를 문장 내 임의의 자리에 넣는걸 n번 반복한다.
- Random Swap, RS
  - 무작위로 문장 내에서 두 단어를 선택하고 위치를 바꾼다. 이것도 n번 반복
- Random Deletion, RD
  - 확률 p를 통해 문장 내에 있는 각 단어들을 랜덤하게 삭제한다.

EDA: Easy Data Augmentation Techniques for Boosting Performance on Text Classification Tasks

자연어 처리 증강 성능 비교

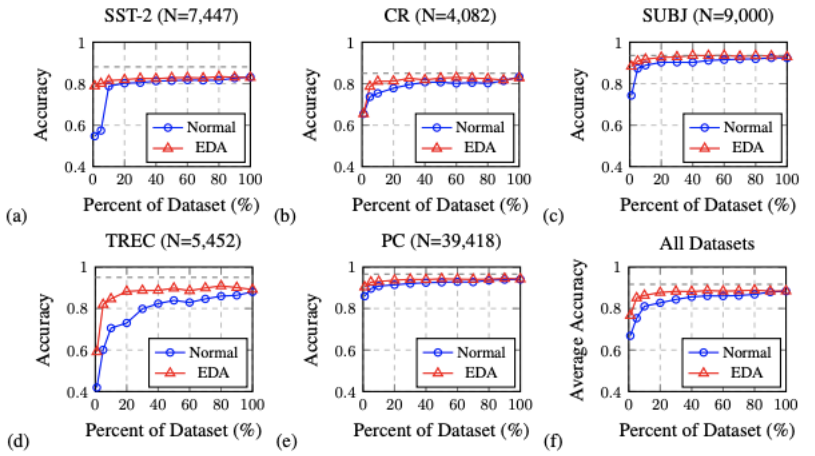


Figure 1: Performance on benchmark text classification tasks with and without EDA, for various dataset sizes used for training. For reference, the dotted grey line indicates best performances from Kim (2014) for SST-2, CR, SUBJ, and TREC, and Ganapathibhotla (2008) for PC.

	Training Set Size			
Model	500	2,000	5,000	full set
RNN	75.3	83.7	86.1	87.4
+EDA	79.1	84.4	87.3	88.3
CNN	78.6	85.6	87.7	88.3
+EDA	80.7	86.4	88.3	88.8
Average	76.9	84.6	86.9	87.8
+EDA	79.9	85.4	87.8	**88.6**

Table 2: Average performances (%) across five text classification tasks for models with and without EDA on different training set sizes.

필요한 라이브러리 불러오기

```
import tensorflow as tf
from tensorflow.keras import datasets, layers, models

import matplotlib.pyplot as plt
```

데이터 불러오기

```
(train_images, train_labels),(test_images,test_labels) =datasets.cifar10.load_data()
```

ImageDataGenerator 클래스 불러오기

```
from tensorflow.keras.preprocessing.image import ImageDataGenerator

datagen = ImageDataGenerator(
    rescale=1./255,
    rotation_range=30,
    zoom_range=0.2,
    width_shift_range=0.1,
    height_shift_range=0.1,
    horizontal_flip=True,
    vertical_flip=True,
    shear_range=0.2
)
```

이미지 데이터 미리보기

```
class_names = ['airplane','automobile','brid','cat',\
               'deer','dog','frog','horse','ship','truck']

plt.figure(figsize=(20,20))

for i in range(25):
    plt.subplot(5,5,i+1)
    plt.grid(False)
    plt.imshow(train_images[i])
    plt.xlabel(class_names[train_labels[i][0]])

plt.show()
```

CNN 모델 만들기

```
model = models.Sequential()
model.add(layers.Conv2D(32,(3,3),activation='relu',input_shape=(32,32,3)))
model.add(layers.MaxPooling2D((2,2)))
model.add(layers.Conv2D(64,(3,3),activation='relu'))
model.add(layers.MaxPooling2D((2,2)))
model.add(layers.Conv2D(64,(3,3),activation='relu'))

model.add(layers.Flatten())
model.add(layers.Dense(64,activation='relu'))
model.add(layers.Dense(10))
```

모델 설정하고, 데이터 증강 포함 학습

```
train_gen = datagen.flow(train_images, train_labels, batch_size=128)

model.compile(optimizer='adam',
              loss=tf.keras.losses.SparseCategoricalCrossentropy(from_logits=True),
              metrics=['accuracy'])


history = model.fit(train_gen, epochs=100, validation_data=(test_images, test_labels))
```

자연어 한국어 전용 EDA

```
pip install ktextaug
```

Getting Started

ktextaug를 사용하는 간단한 예제입니다.

패키지 0.1.9 버젼부턴 기본적으로 TextAugmentation() 을 사용하여 처리하는 것을 권장합니다. multiprocessing 을 이용하여 대용량의 데이터를 빠르게 처리할 수 있도록 만들었습니다.

```
from ktextaug import TextAugmentation

sample_text = '달리는 기차 위에 중립은 없다. 미국의 사회 운동가이자 역사학자인 하워드 진이 남긴 격언이다.'
sample_texts = ['프로그램 개발이 끝나고 서비스가 진행된다.', '도움말을 보고 싶다면 --help를 입력하면 된다.']
agent = TextAugmentation(tokenizer="mecab",
                         num_processes=1) # num_process 가 -1 일시 자동으로 가능한 process의 절반으로
print(agent.generate(sample_text))     # default is back_translation
print(agent.generate(sample_texts))
```

https://github.com/jucho2725/ktextaug

한국어 전용 EDA 적용

```
from ktextaug import TextAugmentation

sample_text = '달리는 기차 위에 중립은 없다. 미국의 사회 운동가이자 역사학자인 하워드 진이 남긴 격언이다.'
agent = TextAugmentation(tokenizer="komoran",
                         num_processes=-1)

print(agent.generate(sample_text))
```