

설계 제안서

과제	이미지 노이즈 제거 클래스 분류기
제출일자	2023.06.05
연구참여자	조장 김찬영(20190895) : 프로젝트 총괄 및 보고서 작성
	팀원 허진환(20190954) : 프로그래밍 총괄
	팀원 지성원(20190948) : 프로그래밍, 자료조사
	팀원 우상욱(20190919) : 프로그래밍, 회의록 작성, ppt 제작
팀명 / 수업명	셀프러닝 / 머신러닝(4-6)

1. 설계과제 소개

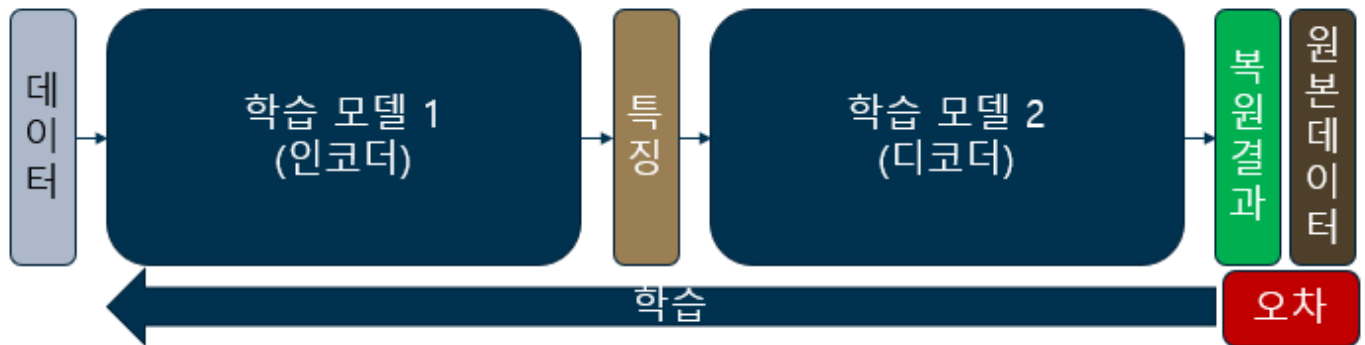
노이즈가 포함된 이미지를 분류하는 모델!



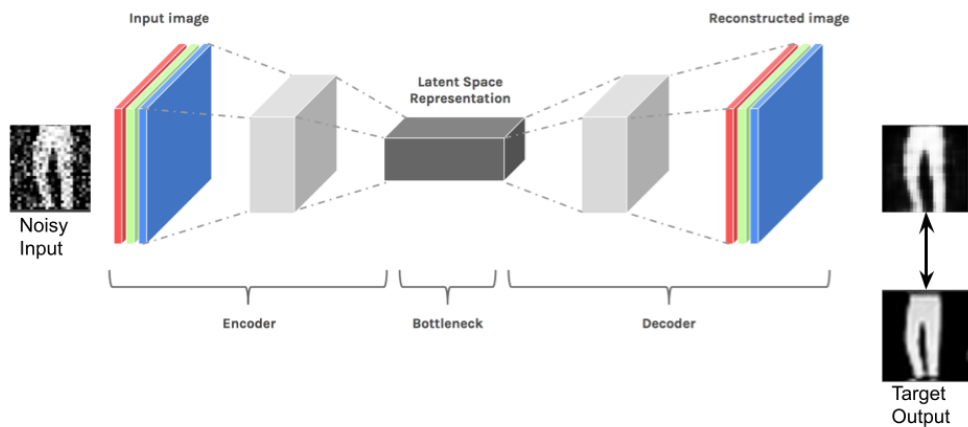
머신 러닝을 통하여 노이즈가 포함된 이미지를 classification 하는 모델을 설계한다. 성능의 증가를 위해 오토인코더를 사용하여 이미지의 노이즈 제거 후, '헬리콥터', '비행기', '얼굴'과 같은 101 가지 다른 클래스의 이미지를 분류하기 위한 알고리즘을 개발한다. Sequential 모델을 사용하여 인공적으로 특징을 추출하고, 모델을 훈련하여 새로운 입력 이미지의 클래스를 분류하고 예측하는 것을 목표로 한다. 이미지의 특징을 추출하고 학습하기 위해 합성곱 신경망(CNN)을 사용하고, 모델을 분류하기 위해 훈련시킨다. 두 가지 다른 활성화 함수의 영향을 비교할 예정으로, Rectifier (relu) 활성화 함수와 tanh 활성화 함수의 정확도를 비교할 예정이다.

2. 설계과제 내용

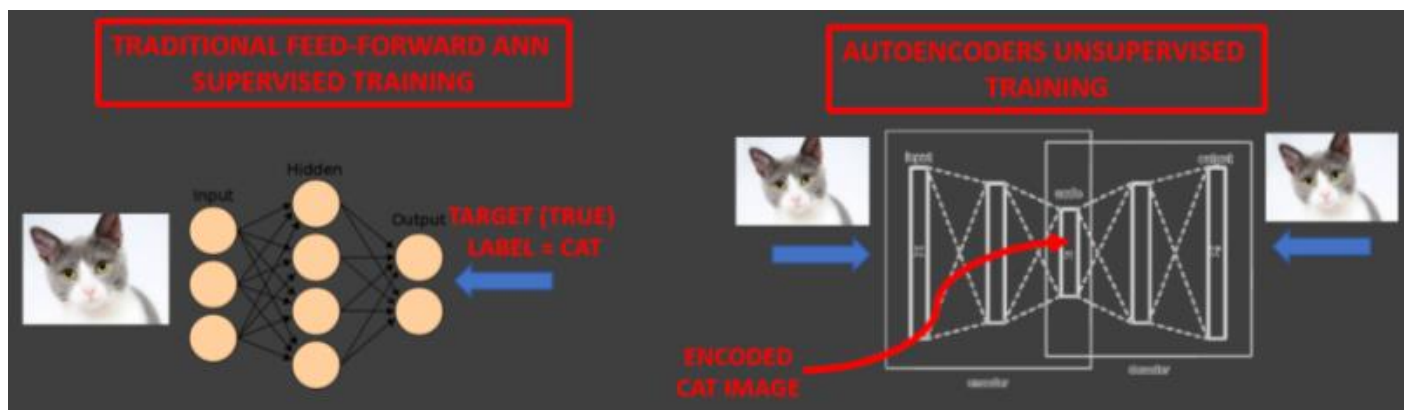
2.1 오토인코더를 통하여 이미지의 노이즈를 제거



2.1.1 오토인코더: 오토인코더는 데이터 인코딩(representation learning) 작업을 수행하는 인공 신경망의 한 유형으로, 오토인코더의 특징으로는 입력 데이터를 입력 및 출력으로 사용한다.



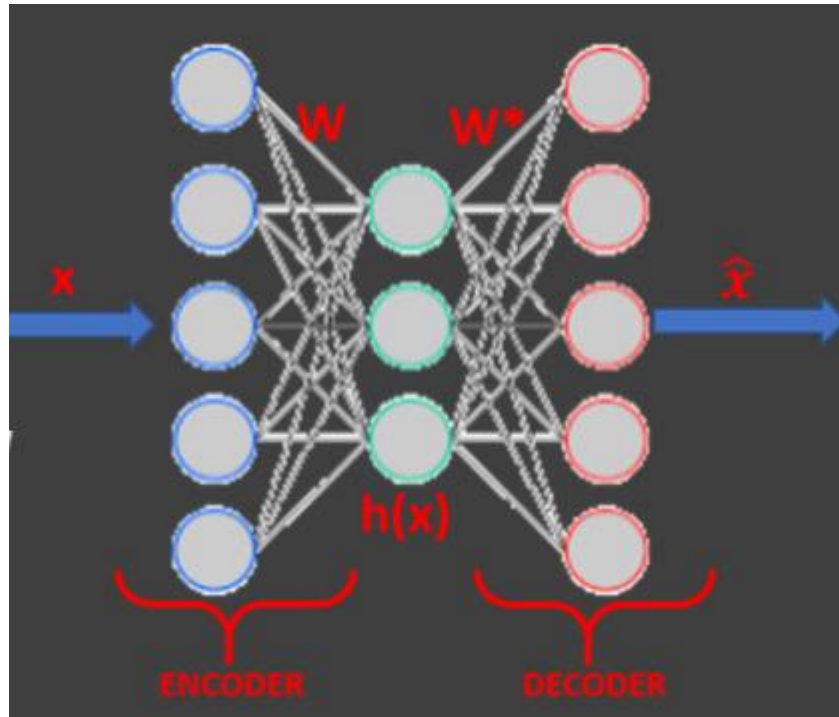
2.1.2 코드 레이어: 오토인코더는 네트워크에 병목 현상을 만들어 줌으로서 동작한다. 이 병목 현상은 네트워크가 원래 입력의 압축된(encoded) 버전을 생성하도록 한다. 오토인코더는 입력 데이터와 상관관계가 있는 경우에 잘 작동한다.(입력 데이터가 모두 독립적인 경우에는 잘 작동하지 않을 수 있음)



2.1.3 오토인코더의 수학적 원리:

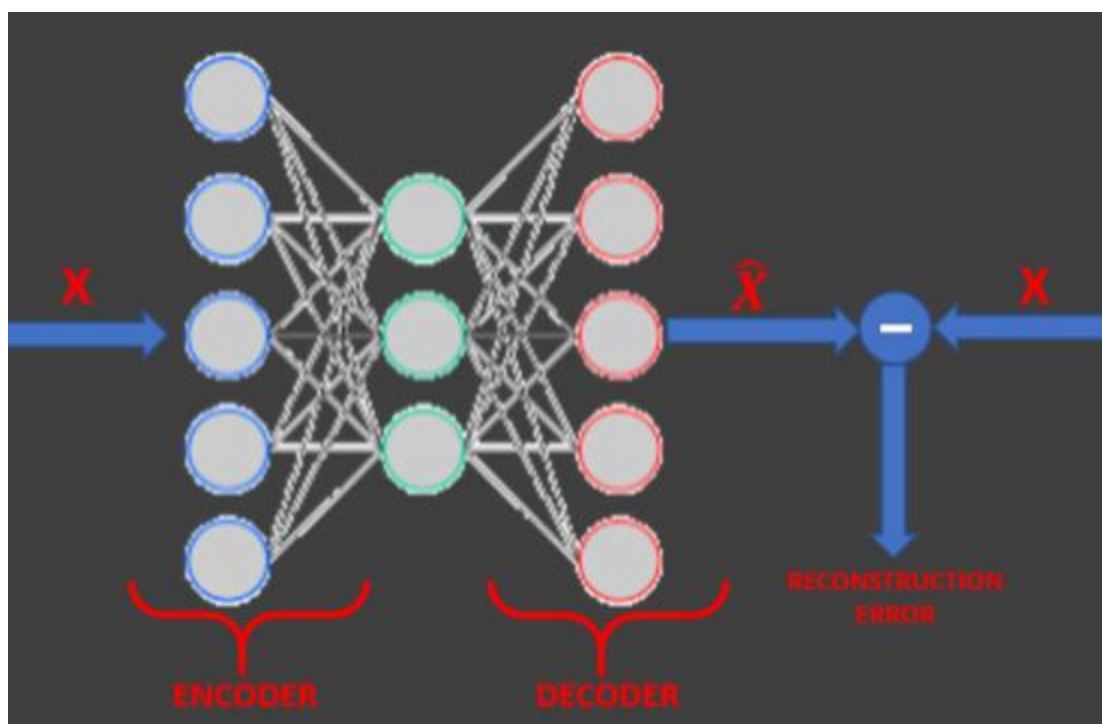
Encoder : $h(x) = \text{sigmoid}(W * x + b)$

Decoder : $\hat{x} = \text{sigmoid}(W^* * h(x) + c)$

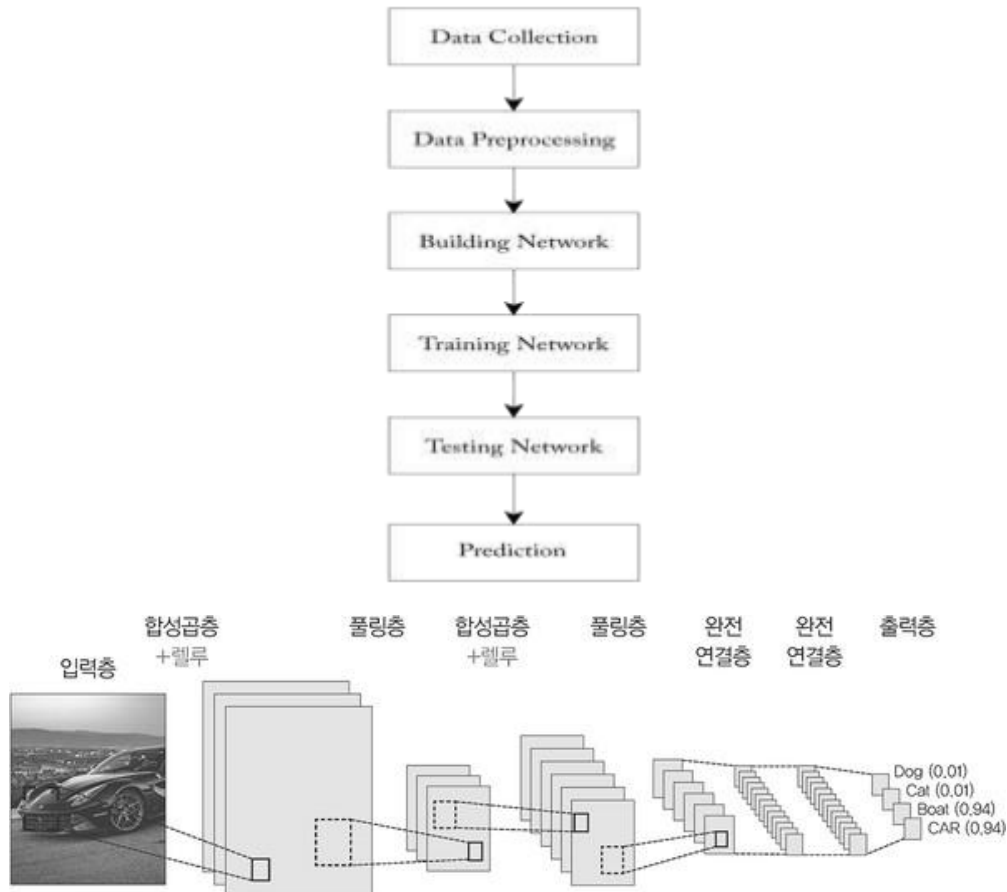


2.1.4 재구성 오차: 오토인코더의 목표는 입력 X 와 네트워크 출력 \hat{X} 사이의 재구성 오차를 최소화하는 것이다.

오토인코더의 차원 축소(잠재 공간)는 선형 활성화 함수를 사용하는 경우 PCA(주성분 분석)와 매우 유사하다.



2.2 합성곱 신경망(CNN)을 사용한 이미지 Classification



2.2.1 신경망: 신경망의 기본 단위는 퍼셉트론이다. 입력 레이어에서 이 퍼셉트론으로 입력이 들어가고, 이 퍼셉트론은 결과를 출력한다. 본 프로젝트에서의 입력은 이미지로 들어가게 되는데, 이 이미지는 훈련 및 테스트 데이터셋으로 나뉜다. 그런 다음 훈련 데이터셋으로 모델을 학습시킨다. 그 후 테스트 데이터셋으로 모델의 성능을 테스트한다. 프로세스의 마지막 부분은 임의의 입력 데이터의 클래스를 예측하는 것이다.

2.2.2 입력층에서의 이미지 정규화: 첫 번째 단계는 입력 이미지를 정규화하는 것으로, 이미지는 RGB 값의 집합으로 이루어져 있으며, 이 값들은 채널로써 주어진다. 채널의 값이 1 이면 그레이스케일, 3 이면 RGB 이다. 따라서 이러한 값을 공통 범위로 정규화해야 한다. 첫 번째 레이어에서 이미지를 RGB 로 정규화한다.

2.2.3 활성화 함수: 입력을 임계값으로 처리하기 위해 ReLU 를 사용한다. 이는 일반적으로 합성곱 계층에서 정의된다. 텐서플로우에서는 두 가지 방법으로 정의할 수 있는데, 첫 번째는 합성곱 계층에서 정의하는 것이고, 두 번째는 활성화 함수를 포함하는 다른 계층을 추가하는 것이다. 본 프로젝트에서는 첫 번째 방법인 합성곱 계층에서 정의하는 옵션을 사용하기로 하였다. ReLU 의 공식은 다음과 같다:

$$f(x) = \max(0, x)$$

2.2.4 합성곱 계층: 합성곱 계층은 입력 위에 일련의 필터를 합성한다. 높은 필터 응답은 필터와 입력 간의 유사성을 나타내고, 그 반대도 마찬가지이다. 이 계층에서 얻은 필터 출력을 통해 입력 이미지의 클래스에 대한 결정을 내릴 수 있다! 이 계층은 입력 이미지의 차원은 변경하지 않으며 출력 이미지의 채널 수를 변경하는 선형 변환을 수행한다. 합성곱 계층은 일정한 바이어스로 입력을 받고 각 입력에 대하여 가중치가 정의된다. 이 가중치는 여러 값을 역전파하여 모델을 개선하기 위해 조정할 때 오류를 줄이는 데 도움이 된다.

2.2.5 Pooling 층, Maxpool 계층: Maxpool 작업은 이미지를 다운샘플링하기 위해 사용된다. 이는 입력을 가져와 원하는 크기로 변환한다. 이 단계에서는 행과 열의 수를 변경하지만 깊이는 동일하게 유지된다. Maxpool 작업은 데이터에 과적합되지 않도록 도움이 된다.

2.2.6 완전 연결 계층: 이 계층은 모든 합성곱, Maxpool 및 ReLU 작업이 완료된 후 사용된다. 이 계층은 위의 작업이 출력인 주어진 입력의 숫자 값을 계산한다.

2.2.7 softmax 계층: 합성곱 신경망의 마지막은 일반적으로 softmax 분류기 또는 SVM(Support Vector Machine)이다. 본 프로젝트에서는 softmax 를 사용할 예정이다. softmax 분류기는 배열을 입력으로 받고, 데이터셋의 다른 범주에 대한 출력을 제공한다.

3. 프로젝트 진행 현황

STEP #1: 라이브러리 및 데이터셋 불러오기

```
import tensorflow as tf
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
import random
```

[31] ✓ 0.4s

```
# 이미지 분류기를 위해 다양한 카테고리를 가진 cifar10 데이터셋 사용
(X_train, y_train), (X_test, y_test) = tf.keras.datasets.cifar10.load_data()
```

[32] ✓ 1.6s

```
X_train.shape
```

[65] ✓ 0.2s

... (50000, 32, 32, 3)

```
X_test.shape
```

[66] ✓ 0.3s

... (10000, 32, 32, 3)

이미지 분류를 위해 다양한 카테고리를 가진 이미지 dataset 인 cifar10 을 사용하고, 훈련용/검증용 데이터로 구분하였다.

STEP #2: 데이터 이미지 출력

```
# 이미지 여러 개 출력
# 15 * 15 형식으로 출력
W_grid = 15
L_grid = 15

# subplots()함수 fig, axes 객체 리턴
fig, axes = plt.subplots(L_grid, W_grid, figsize = (17,17))

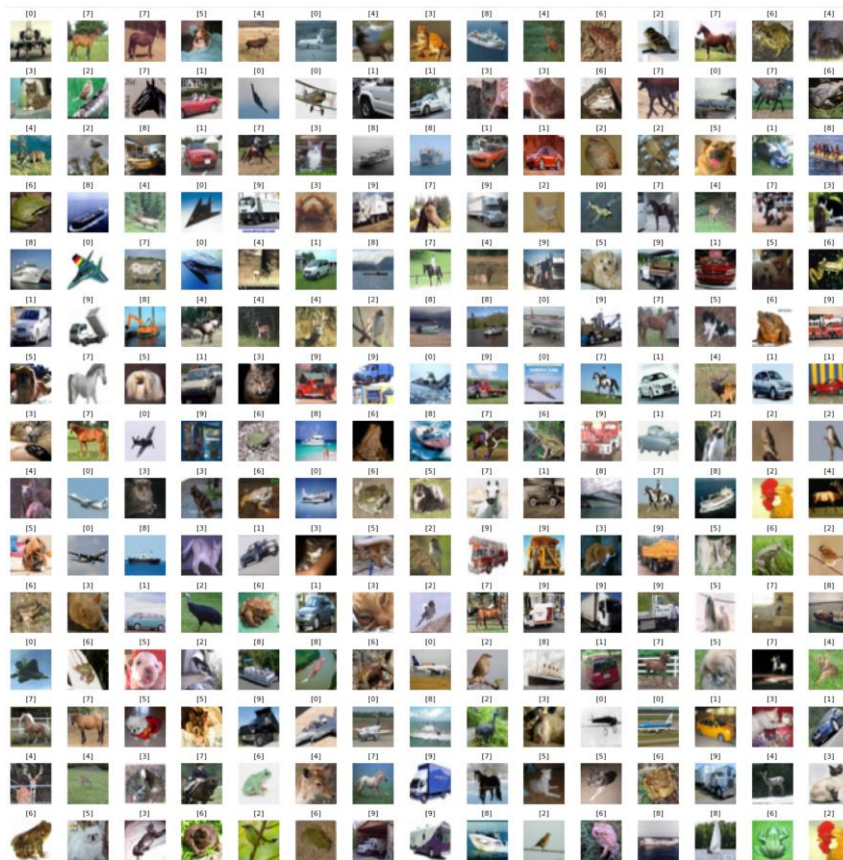
# 15 * 15 행렬을 255크기의 배열로 바꿈
axes = axes.ravel()

# 훈련용 데이터셋 크기 반환
n_training = len(X_train)

# 0 ~ n_training 까지 중 랜덤한 인덱스의 데이터 이미지 출력
for i in np.arange(0, W_grid * L_grid):
    # 랜덤한 인덱스 지정
    index = np.random.randint(0, n_training)
    # 인덱스에 해당하는 데이터 이미지 출력
    axes[i].imshow( X_train[index] )
    axes[i].set_title(y_train[index], fontsize = 8)
    axes[i].axis('off')

plt.subplots_adjust(hspace=0.4)
```

✓ 15.3s



이미지 데이터셋이 제대로 로드되어 훈련용 데이터로 들어갔는지 확인하기 위해 데이터 이미지를 랜덤한 인덱스로 접근하여 위와 같이 15*15 형식으로 출력(출력 화면 밑으로 스크롤 시 15 * 15 형식)하였다.

STEP #3: 이미지 노이즈 추가

```
# 노이즈 강도 0.3
noise_factor = 0.3

# 노이즈 추가된 이미지 저장할 배열
noise_dataset = []

# 훈련용 데이터 노이즈 추가
for img in X_train:
    noisy_image = img + noise_factor * np.random.randn(*img.shape) # 이미지 픽셀에 노이즈 추가
    noisy_image = np.clip(noisy_image, 0., 1.) # 이미지 클리핑
    noise_dataset.append(noisy_image) # 노이즈 추가된 이미지 noise_dataset 배열에 추가

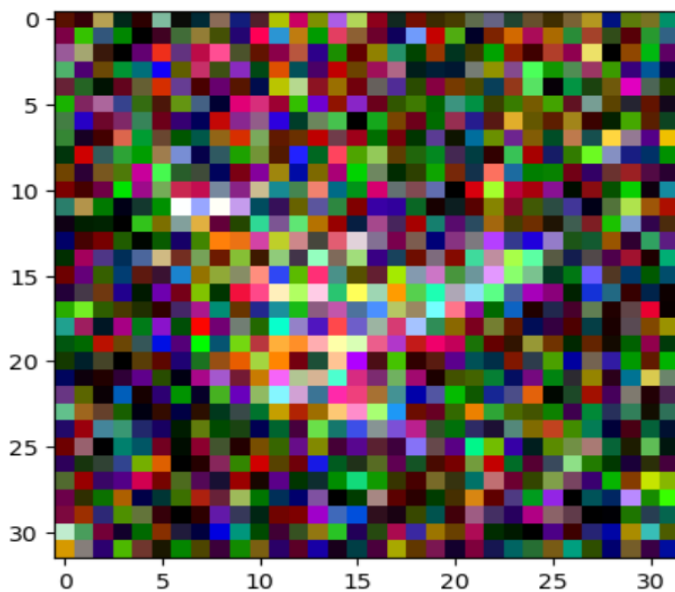
noise_dataset = np.array(noise_dataset)
noise_dataset.shape
```

✓ 26.8s

(50000, 32, 32, 3)

```
# 노이즈 추가된 이미지 하나 임의 출력
plt.imshow(noise_dataset[22], cmap="gray")
```

<matplotlib.image.AxesImage at 0x1710664af40>



```
# 검증용 데이터 노이즈 추가
noise_test_set = []
for img in X_test:
    noisy_image = img + noise_factor * np.random.randn(*img.shape) # 이미지 픽셀에 노이즈 추가
    noisy_image = np.clip(noisy_image, 0., 1.) # 이미지 클리핑
    noise_test_set.append(noisy_image) # 노이즈 추가된 이미지 noise_dataset 배열에 추가

noise_test_set = np.array(noise_test_set)
noise_test_set.shape
```

(10000, 32, 32, 3)

이미지 노이즈 추가를 위해 노이즈 강도 설정 후 각 이미지 데이터 픽셀에 노이즈를 추가하고, 노이즈가 잘 추가되었는지 확인하기 위해 노이즈 추가된 이미지 한 장을 임의 출력하였다.

STEP #4: 오토인코더 모델 설계 및 훈련(.fit())

```
# 입력 이미지의 형태 저장
input_shape = X_train.shape[1:]

# 오토인코더 모델 생성
autoencoder = tf.keras.models.Sequential([
    ### 인코더 부분
    # 이미지 특성 추출
    tf.keras.layers.Conv2D(32, (3, 3), activation='relu', padding='same', input_shape=input_shape),
    # 2x2 최대 풀링을 사용하여 이미지 크기를 1/2배 (차원 축소 및 압축)
    tf.keras.layers.MaxPooling2D((2, 2), padding='same'),
    # 이미지 특성 추출
    tf.keras.layers.Conv2D(32, (3, 3), activation='relu', padding='same'),

    ### 디코더 부분
    # 2x2 업샘플링을 사용하여 이미지 크기를 2배 (차원 복원 및 복원)
    tf.keras.layers.UpSampling2D((2, 2)),
    # 이미지 특성 추출
    tf.keras.layers.Conv2D(3, (3, 3), activation='sigmoid', padding='same')
])
```

```
# 오토인코더 모델 컴파일
autoencoder.compile(loss='binary_crossentropy', optimizer=tf.keras.optimizers.Adam(lr=0.001))
# 오토인코더 모델 요약 출력
autoencoder.summary()
```

WARNING:absl:lr is deprecated in Keras optimizer, please use learning_rate or use the legacy optimizer: "sequential_2"

Layer (type)	Output Shape	Param #
=====		
conv2d_6 (Conv2D)	(None, 32, 32, 32)	896
max_pooling2d (MaxPooling2D)	(None, 16, 16, 32)	0
conv2d_7 (Conv2D)	(None, 16, 16, 32)	9248
up_sampling2d (UpSampling2D)	(None, 32, 32, 32)	0
conv2d_8 (Conv2D)	(None, 32, 32, 3)	867

오토 인코더 모델의 인코더 부분을 설계할 때 이미지 데이터 특성을 추출하기 위해 Conv2D 함수를 이용하고, MaxPooling2D() 함수를 사용하여 차원 축소 및 압축을 시행하도록 하였다. 디코더 부분을 설계할 때는 그의 역순으로 Upsampling2D() 함수를 이용하여 차원 복원 및 압축으로부터의 복원을 시행하도록 하였다.

```
# 데이터 훈련
autoencoder.fit(noise_dataset, X_train, epochs=10, batch_size=128, validation_data=(noise_test_set, X_test))
```

설계한 오토 인코더 모델을 통해 데이터 훈련을 fit() 함수를 통해 진행하였다. 학습 반복 횟수인 epochs 를 10 으로 두었고, 노이즈 추가 데이터 noise_dataset 입력에 대해 정답 레이블인 원본 데이터 X_train 과 비교하여 학습하도록 하였다.

STEP #5: 모델 성능 측정 및 노이즈 제거 예시 출력

```
# 모델 성능 측정 (예측 정확도 파라미터)  
evaluation = autoencoder.evaluate(noise_test_set, X_test)  
print('Test Accuracy : {:.3f}'.format(evaluation))
```

```
313/313 [=====] - 4s 14ms/step - loss: 0.5659  
Test Accuracy : 0.566
```

```
# 오토인코더 노이즈 제거 출력을 위한 예측 결과 저장  
predicted = autoencoder.predict(noise_test_set[:10])
```

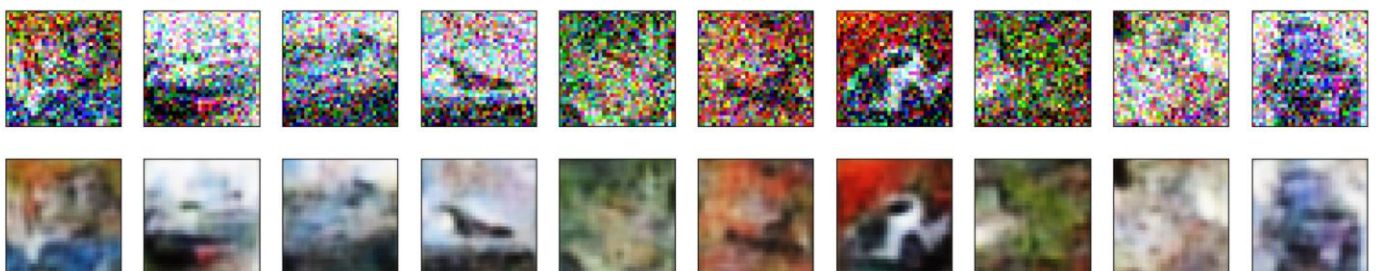
```
1/1 [=====] - 0s 59ms/step
```

```
predicted.shape
```

```
(10, 32, 32, 3)
```

모델 성능 측정을 위해 정확도 파라미터를 사용하여 모델 성능을 측정하였고, 해당 모델에서 0.566 의 정확도를 보였다. 또한, predict()함수를 통해 입력데이터 noise_test_set[:10] (10 개)에 대한 오토인코더 모델 출력의 예측 결과를 predicted 에 담고 아래와 같이 예측 결과를 출력하였다.

```
# 예측 결과 출력  
fig, axes = plt.subplots(nrows=2, ncols=10, sharex=True, sharey=True, figsize=(20,4))  
for images, row in zip([noise_test_set[:10], predicted], axes):  
    for img, ax in zip(images, row):  
        ax.imshow(img.reshape((32, 32, 3)), cmap='Greys_r')  
        ax.get_xaxis().set_visible(False)  
        ax.get_yaxis().set_visible(False)
```



noise_test_set[:10] (10 개)에 대한 오토인코더 모델 출력의 예측 결과인 predicted 를 위와 같이 출력하였다. 1 행에는 노이즈 추가된 이미지인 noise_test_set[:10] (10 개)가 출력되고, 2 행에는 그에 대한 오토인코더 모델 출력의 예측 결과인 predicted 이미지 10 개가 출력된다.

4. 추진 체계

과제를 진행하기 위한 팀원은 4 명으로 다음과 같은 역할을 담당한다. 목표의 구상 단계에서부터 개발 완료시까지 토론 및 협력을 통하여 개발 목표를 달성할 수 있도록 한다.

팀장	김찬영	프로젝트 총괄 및 보고서 작성
팀원	허진환	프로그래밍 총괄
팀원	지성원	프로그래밍, 자료조사
팀원	우상욱	프로그래밍, 회의록 작성, ppt 제작

5. 설계 추진 일정

2023 년 5 월 22 일 ~ 2023 년 6 월 12 일

수행 내용		일정							
		1	2	3	4	5	6	7	8
목표와 기준 설정	-설계목표 설정 -목표 달성 방법에 대한 논의								
합성	-기능별 구현 방법 결정 -적용할 기술								
분석	-세부적인 기능 구현 방법 결정 -목표달성 가능성 확인								
제작	-분류기 프로그래밍								
시험/평가	-시험 및 검증 -재설계								
결과	-결과보고 및 시연								

회 의 록				
회의명	주제 선정 및 목표 설정			
일 시	2023.05.28		장 소	온라인 회의
의 제	주제 선정 및 목표 설정을 위한 안건 회의			
회의 내용	<div>1.과제 수행을 위한 주제 및 목표 설정</div> <div>-머신러닝에 대한 실습을 직접 여러 번 반복하면서 머신러닝에 대한 이해도를 높일 수 있는 과제를물을 선정하면 좋겠다는 의견</div> <div>-교재에서 배운 내용을 최대한 활용하는 것을 목표로 하면 좋겠다는 의견</div> <div>-실생활에 활용 할 여지가 있는 과제를물을 선정하는 것이 좋겠다는 의견</div> <div>-조원 모두가 배경지식이 있거나 공통적으로 관심이 있는 주제를 선정하자는 의견</div> <div>-기존의 교재에서 수행했던 내용,코드를 바탕으로 기능추가,개선을 목표로 하는 과제를물을 선정 하자는 의견</div> <div>-이전에 중간고사 대체 과제로 나왔던 노이즈 제거를 통한 이미지 구별 코드를 바탕으로 과제를 진행하는 것이 좋겠다는 의견</div> <div>-이전에 진행했던 이미지 분류기 과제를물을 서로 공유하고 피드백 하면서 목표 설정을 하는 게 좋겠다는 의견</div>			
결의 사항	<div>조원 모두가 배경지식이 있는 주제물중 실제로 실습,과제를 진행했던 이미지 분류기를 주제로 선정함</div> <div>조원 모두가 이전에 각자 이미지 분류기를 바탕으로한 중간고사 대체 과제를 진행한적이 있기에 서로 공유하고 참고할 내용이 많다 여겨 의견교환이 원활이 여겨지고</div> <div>교재에서 배운 내용을 최대한 활용하기에도 적합하다는 의견에 모두 동의해 이번 과제목표에 적합하다고 여김</div>			
이견 사항				
참석 현황	대상	성명	학번	서 명(자필)
	참석자	김찬영	20190895	
		우상욱	20190919	
		지성원	20190948	
		허진환	20190954	
불참자				
작성자	우상욱	작성일	2023.05.28	

회 의 록				
회의명	과제 수행을 위한 목표 구체화			
일 시	2023.05.30		장 소	온라인 회의
의 제	목표 구체화를 위한 의견 수립			
회의 내용	1.이미지 분류기를 활용한 과제물의 목표 구체화 -교재에 있는 알고리즘,이론을 바탕으로 이미지 분류기에 새로운 기능을 추가하자는 의견 -기존 이미지 분류기에 성능에서 아쉬웠던 부분을 토의하고 그것을 개선시키는 방안을 찾아보자는 의견 -생소하거나 관련 지식 자체가 적은 기능,목표보다는 관련 자료를 얻기 쉽고 직관적인 기능을 추가하자는 의견 -이전에 수행했던 이미지 분류기에는 노이즈 제거에 대한 성능이 아쉽다는 의견에 조원 모두가 동의하고 노이즈 제거 기능에 대한 개선,성능향상에 집중하자는 의견 -이미지 정교화,오류 개선 등을 위한 알고리즘 중 교재에 있는 이론 중 합성 신경망 알고리즘을 구현해 과제물에 사용하자는 의견 -노이즈 제거를 위한 알고리즘 중 오토 인코더를 사용해 노이즈를 제거하는 것이 기존보다 기능 향상에 도움이 된다는 의견			
결의 사항	조원모두가 이전에 했던 이미지 분류기 과제물을 서로 공유하고 의견을 나누던 중 기존에 교재에 코드에는 노이즈 제거 기능에 대한 성능의 아쉬움이 있다는 의견이 공통적으로 나왔으며 이를 개선하는 것을 중점적으로 과제목표를 설정하기로 함 노이즈 제거를 위해 신경망 알고리즘 중 하나인 오토 인코더를 사용해 기능 구현을 목표로 하였으며 이미지 정교화,오류개선 등을 위해 합성신경망 구현에도 집중, 이외에도 다른 이론,알고리즘 등을 이용해 과제물의 완성도를 높이는 것을 목표로 함			
이견 사항				
참석 현황	대상	성명	학번	서 명(자필)
	참석자	김찬영	20190895	
		우상욱	20190919	
		지성원	20190948	
		허진환	20190954	
불참자				
작성자	우상욱		작성일	2023.05.30

회 의 록				
회의명	목표 달성을 위한 역할 수행 및 설계 제안서 작성			
일 시	2023.06.02		장 소	온라인 회의
의 제	설계보고서 작성 및 발표 준비			
회의 내용	<p>1.각자 분담했던 코딩 진행도에 대해 공유 및 앞으로의 계획 토의</p> <p>-기초적인 라이브러리,데이터셋 준비 및 데이터 이미지에 노이즈를 추가하는 기능구현 완료</p> <p>-오토인코더 모델 생성 및 훈련 실시 및 학습 진행중</p> <p>-모델 성능 측정 및 예시 출력 기능 구현</p> <p>-세부적인 기능 구현을 위한 의견을 토의하고 실제 실현 가능성에 대해 검토하기로함</p> <p>-분류기 프로그래밍 완성을 위한 알고리즘 이해도 향상 및 실제 구현을 위한 역할 수행을 할 것</p> <p>-반복적으로 성능 실험을 통해 그 과정에서 문제점,피드백할 내용 지속적으로 검토 예정</p> <p>2.설계 보고서 작성 및 발표 자료 준비</p>			
결의 사항	설계보고서 및 발표준비를 조원모두가 각자 역할을 맡아 작성 후 기초적인 분류기 프로그래밍을 위한 코딩을 진행하였으며 실제 성능 측정에 대해서도 테스트를 진행, 과제를 수행하면서 지속적으로 개선사항을 피드백 함으로서 과제를 진행하기로 함			
이견 사항				
참석 현황	대상	성명	학번	서 명(자필)
	참석자	김찬영	20190895	
		우상욱	20190919	
		지성원	20190948	
		허진환	20190954	
	불참자			
작성자	우상욱	작성일	2023.06.02	