



Unsupervised Learning **AutoEncoder**

“진짜를 제대로 알아야 가짜가 보입니다”



출처 : https://bravo.etoday.co.kr/view/atc_view.php?varAtclId=9114

진폐의 무엇을 알아야 할까요?

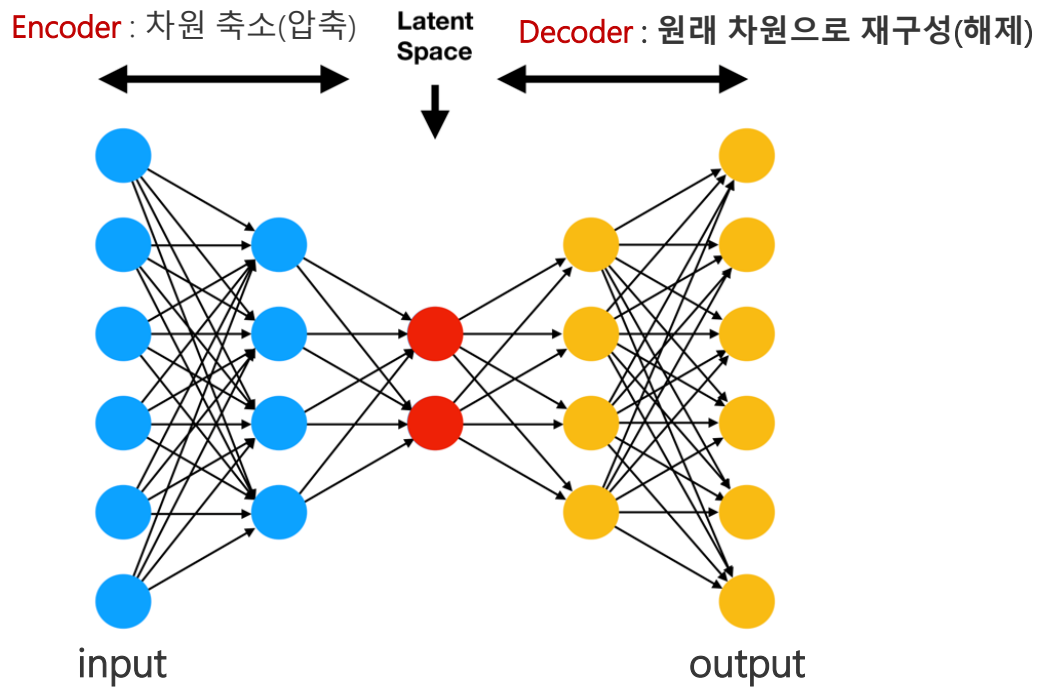


위폐 감별사

- ① **진폐 만**을 가지고 Training 합니다.
- ② 진폐의 **중요한 특징**을 추출해서 기억합니다.

Autoencoder?

✓ 두 개의 부분으로 구성되어 있습니다.

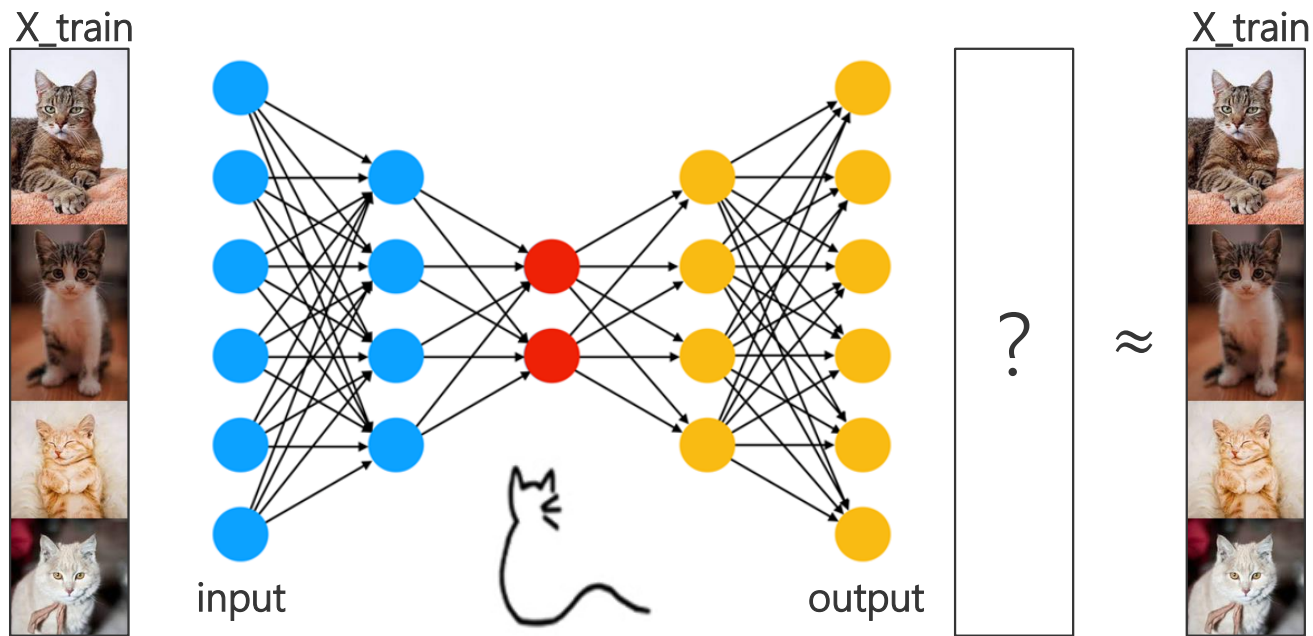


- **차원을 압축**을 하면 어떤 일이 일어날까요?
- 차원을 압축한 후 **input과 최대한 유사한 Output을 만들어야** 한다면 압축으로 얻게 되는 효과는?

Autoencoder?

✓ 예를 들어, 고양이 사진들로 **학습**을 시킨다고 해 봅시다.

- Input과 Output의 차이(오차, Error)를 최소화 시키기 위해 학습(최적화)됩니다.

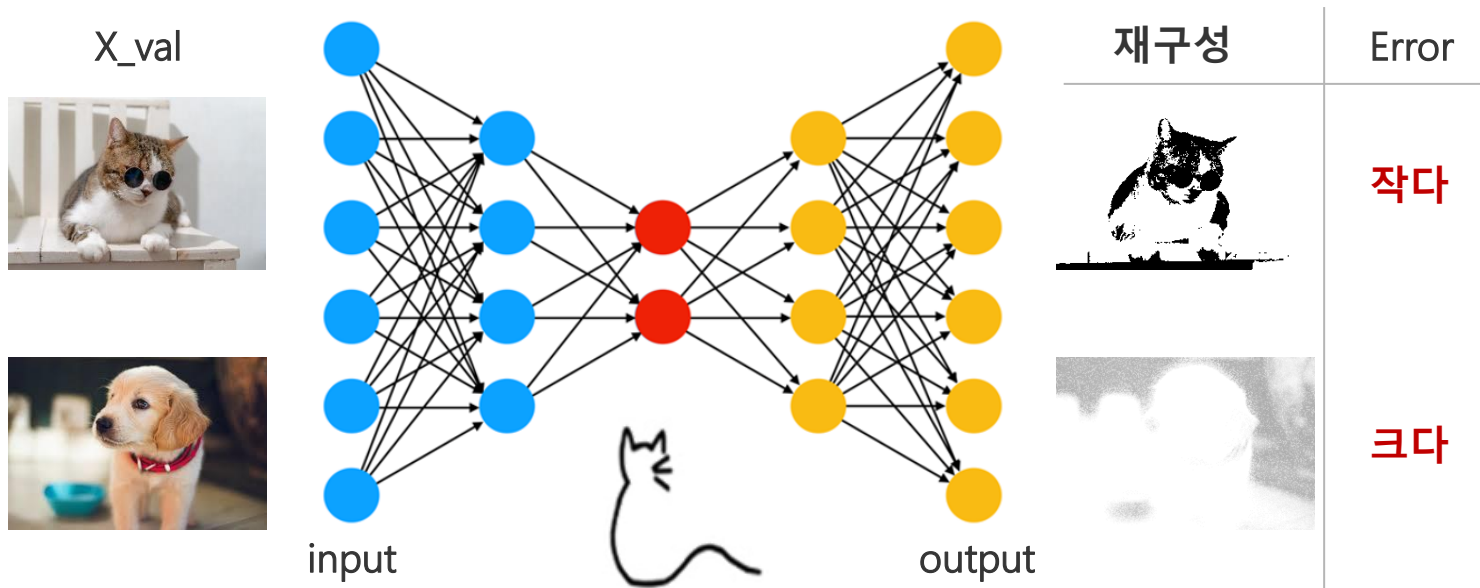


- 학습이 되면, 모델은 고양이의 **공통된 중요한 특징**만을 담게 됩니다. (노이즈는 제거)

Autoencoder?

✓ 모델이 만들어 진 후, 예측을 할 때

- X_{val} 이 입력되면, 결과로 X_{val} 과 같은 차원(변수의 수)으로 결과를 만들어 냅니다.
- 이를 “**재구성(Reconstruction)** 한다.”라고 합니다.



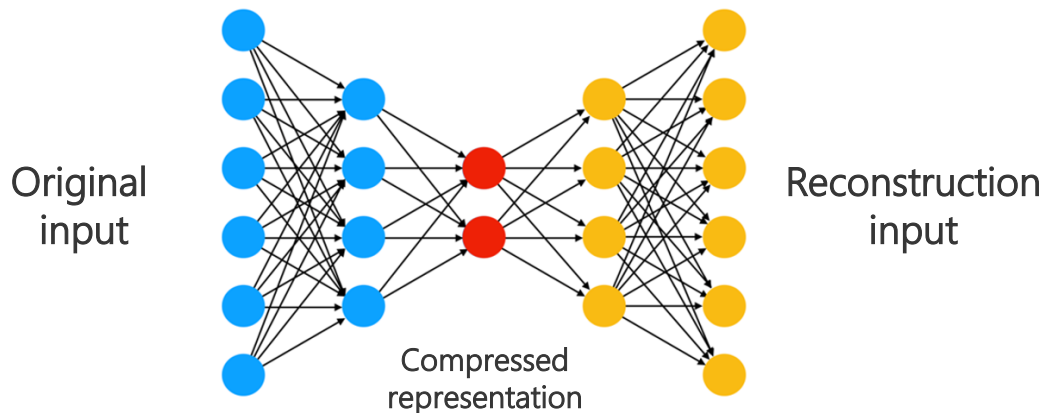
Autoencoder

① Compressed representation

- 데이터를 압축 → 보존되는 것과 손실되는 것
- 어떤 데이터가 손실되고, 어떤 데이터가 보존될까요?

② Original Input → Reconstructed Input

- 학습할 때 데이터를 그대로 복원(재구성, reconstruction)하도록 학습.



Autoencoder for Anomaly Detection

- ✓ 그렇다면 AE를 이상탐지에 어떻게 적용해 볼 수 있을까요?
- ✓ 학습할 때는,
 - Normal Data만 학습
 - 정상 데이터들 만의 고유한 특징을 담아내고, 불필요한 특징, 노이즈 등은 걸러지도록
- ✓ 검증할 때는,
 - Validation / test는 Normal & Abnormal 모두 포함
 - **재구성 오차(Reconstruction Error)**를 계산
 - Normal : 오차가 적을 것이고(Original Input과 Reconstruction Input의 차이가 적음)
 - Abnormal : 오차가 클 것이다.
 - 재구성 오차가 큰 데이터를 Abnormal 이라고 예측!
- ✓ 재구성 오차는 **어떻게 계산? 얼마나 크면 Abnormal?**

모델링을 위한 전처리

✓ 스케일링

- 딥러닝은 feature에 대한 스케일링 필요 → 모델의 최적화를 위해서

✓ 학습 시 Normal Data만 사용

- One Class SVM과 마찬가지로 Normal Data만 사용해서 학습합니다.

```
x_train0 = x_train_s[y_train == 0]
```

AE 모델 구조 예

- ✓ Encoder : Input(52) → layer1(32) → layer2(16)
- ✓ Decoder : layer3(32) → layer4(52)

```
autoencoder = Sequential([Dense(32, input_shape = (52,)), activation = 'relu'),  
                          Dense(16, activation="relu"),  
                          Dense(32, activation='relu'),  
                          Dense(52, activation='relu')])
```

Layer (type)	Output Shape	Param #
dense (Dense)	(None, 32)	1696
dense_1 (Dense)	(None, 16)	528
dense_2 (Dense)	(None, 32)	544
dense_3 (Dense)	(None, 52)	1716

예측 및 평가 절차

✓ 모델 생성 후 다음의 절차로 예측 및 평가 진행

① 예측

- `x_val`로 예측
- 예측 결과는 `x_val`과 동일한 구조(shape)

② 재구성 오차 계산

- 생성한 재구성 오차 함수 사용
- **`recon_err_plot`**(`x_val`, `pred`, `y_val`,)

③ threshold 찾기

- 평가지표를 최대화 하는 threshold 찾기
- **`prec_rec_f1_curve`**(`y_val`, `recon_err`)

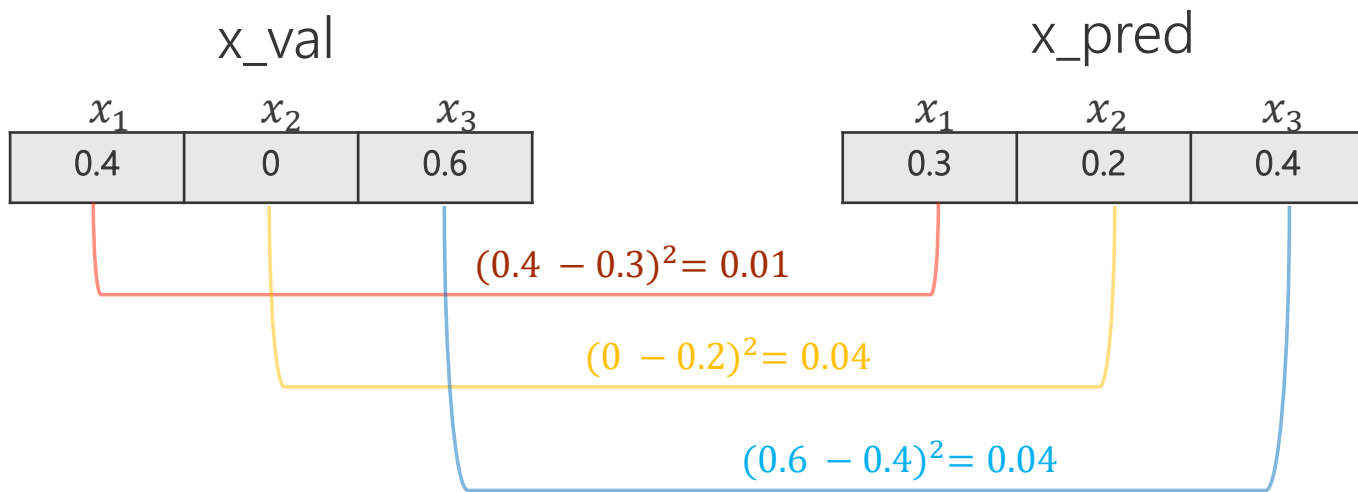
④ 잘라서 평가

- threshold로 잘라서 평가하는 함수
- **`classification_report2`**(`y_val`, `recon_err`, `thres`)

재구성 오차(Reconstruction Error) 계산

✓ 실제 Data point와 예측 값에 대한 MSE 계산

- 분석 단위 별로 MSE 계산



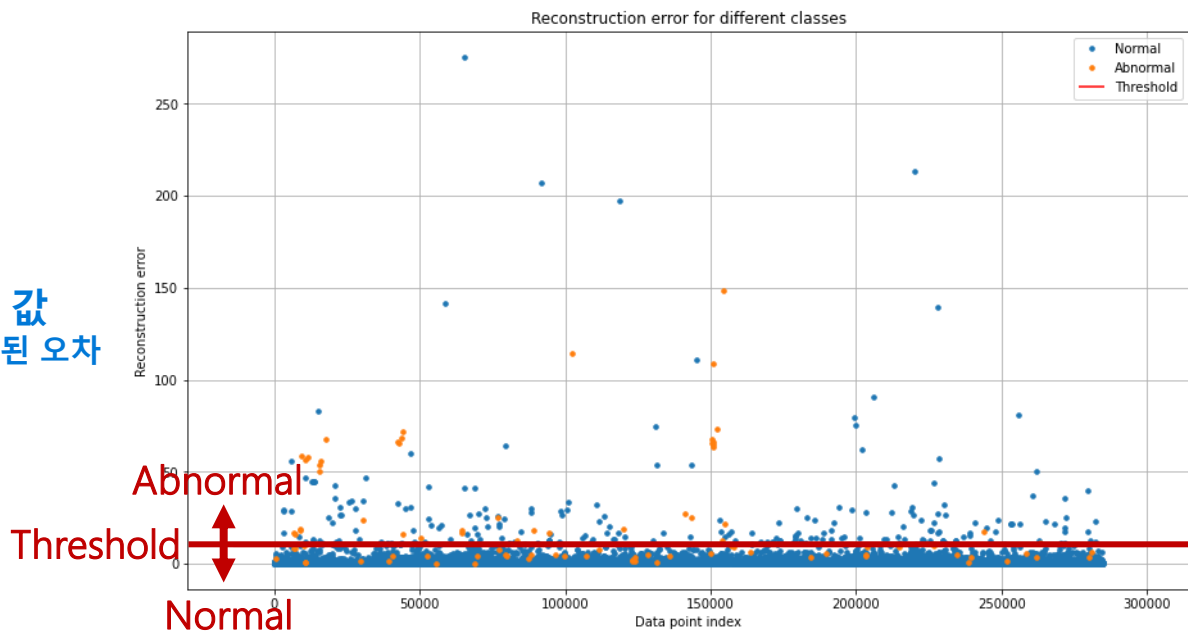
$$Reconstruction Error = \frac{(0.01 + 0.04 + 0.04)}{3}$$

재구성 오차가 얼마나 크면 Abnormal일까?

✓ Threshold는 우리가 조절해야 합니다.

- Threshold 보다 크면 Abnormal, 작으면 Normal 로 예측
- 아래 차트는 재구성 오차 plot 입니다.

예측 값
으로 계산된 오차



실제값

예측한 데이터 index

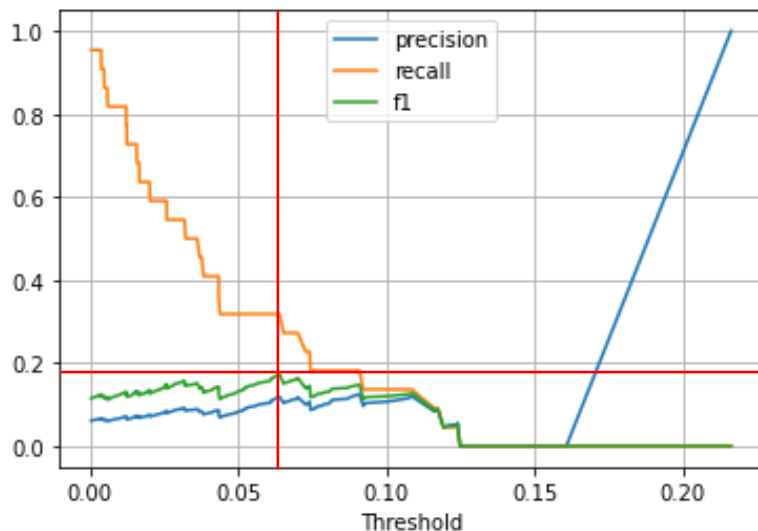
재구성 오차가 얼마나 크면 Abnormal일까?

✓ Threshold 값을 잘 조절하려면,

- 모델을 무엇으로 평가할지 지표를 결정한 후,
- Threshold 값에 따라 그 지표가 어떻게 달라지는지 확인해봐야 합니다.

✓ 그래서 아래와 같은 그래프를 그려볼 겁니다.

- Threshold를 조금씩 조절할 때마다 그때의 Precision, Recall, F1 커브 그리기.
- F1을 최대화 하는 Threshold를 찾습니다.



[추가]재구성 에러에 가장 영향을 많이 주는 변수는?

✓ 중요한 변수를 찾기 위한 아이디어

- 모델에 의해 계산된 [재구성 에러]가 큰 데이터(instance, 행)는 Abnormal 이라고 볼 수 있다.
- 그렇다면, [재구성 에러] 계산 시 값이 큰 변수가 이상탐지에 중요한 변수로 볼 수 있다.

Train (Original input)					Prediction (Reconstruction input)				오차 제공				재구성에러
	Age	Distance	Income	Single	Pr_Age	Pr_Distance	Pr_Income	Pr_Single	SE_Age	SE_Distance	SE_Income	SE_Single	Recon_Error
0	0.404762	0.607143	0.239705	0.0	0.458374	0.285538	0.325915	0.0	0.002874	0.103430	0.007432	0.0	0.028434
1	0.738095	0.607143	0.973828	0.0	0.915366	0.475978	0.821785	0.0	0.031425	0.017204	0.023117	0.0	0.017937
2	0.523810	0.035714	0.128436	0.0	0.355599	0.242709	0.214397	0.0	0.028295	0.042847	0.007389	0.0	0.019633
3	0.523810	0.000000	0.339547	0.0	0.447760	0.281115	0.314397	0.0	0.005784	0.079026	0.000633	0.0	0.021360
4	0.452381	0.285714	0.167457	0.0	0.389328	0.256765	0.250995	0.0	0.003976	0.000838	0.006979	0.0	0.002948

- 그래서 변수 별 오차 제공의 평균(MSE), 표준편차 비교

```
for v in list(x) :  
    print(v.ljust(10, ' '), round(np.mean(reconErr['SE_'+v]),4), round(np.std(reconErr['SE_'+v]),4))
```

```
Age      0.0379 0.056  
Distance 0.0883 0.1208  
Income   0.0602 0.1003  
Single   0.0063 0.0158
```

정리

✓ Autoencoder은 딥러닝을 이용한 비지도 학습 방식 알고리즘

- Encoder 와 Decoder로 구성되어 있습니다.
- Original Input을 **차원축소(압축)**하고 **Reconstruction Input** 으로 만들어 냅니다.
- 둘의 차이를 최소화 하도록 학습이 되기에,
- **중요한 정보는 보존**되고, **불필요한 정보는 제거**될 것이라는 기대(?)가 있습니다.

✓ 이상탐지를 위해

- 재구성 오차를 계산하고
- Threshold로 잘라서
- Normal / Abnormal로 분류하게 됩니다.

✓ Threshold의 결정

- 모델을 무엇으로 평가할지 지표를 결정하고,
- Threshold를 조금씩 조절하면서 지표가 어떻게 달라지는지 그래프를 그리고,
- 최적의 Threshold 값을 찾습니다.