

# AE

## 도입

- Feature를 복제하는 방법
  - 1 to 1 으로 신경망을 구성해준다.
- Feature Extraction 해주는 방법
  - 제약을 주고 Feature를 복제한다. → 중요 요소들을 스스로 판별해준다.

## AutoEncoder

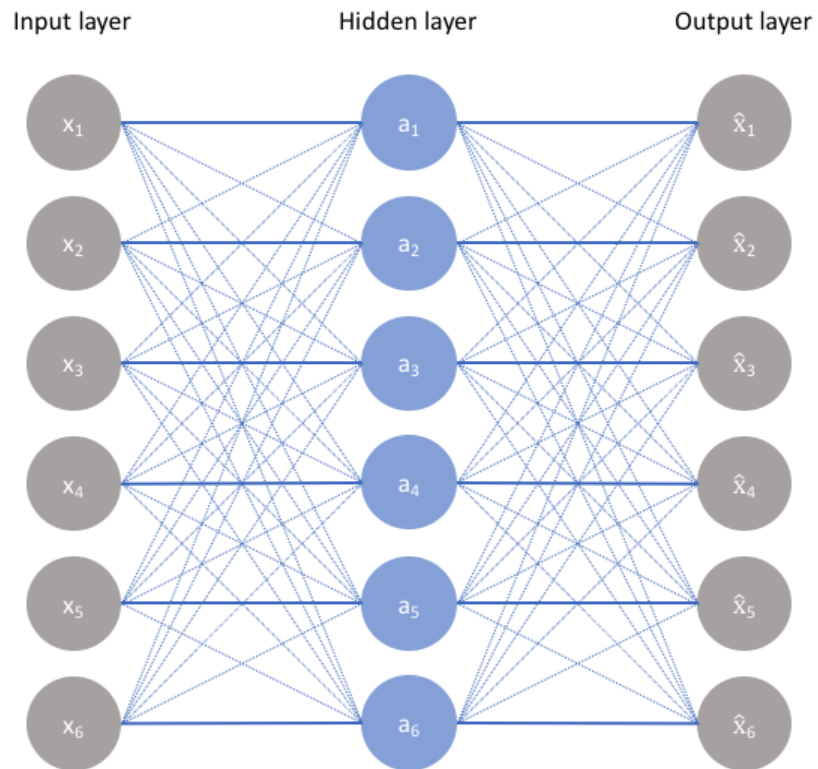
- 구성
  - Encoder
    - 입력 featureSet을 표현 학습을 통해 다른 표현으로 변환
    - 특징을 추출(**Feature Extraction**)
    - 네트워크는 고차원 입력 데이터를 저차원 표현 벡터로 압축합니다.
  - Decoder
    - 새로 학습한 표현을 원본의 형식으로 재구성(**Feature Construction / Reconstruction**)
    - 주어진 표현 벡터를 원본 차원으로 다시 압축 해제
- 특징
  - 입력의 shape와 출력의shape이 같다.
  - 가운데 계층의 노드수가 입력값보다 작다.
    - 데이터를 압축함
    - 차원을 축소함 (Featuer 를 요약하여, 필요한 값들로 정리해 놓는다.)
  - 입력 → encoding → 차원 축소 → decoding → 출력

## 설계

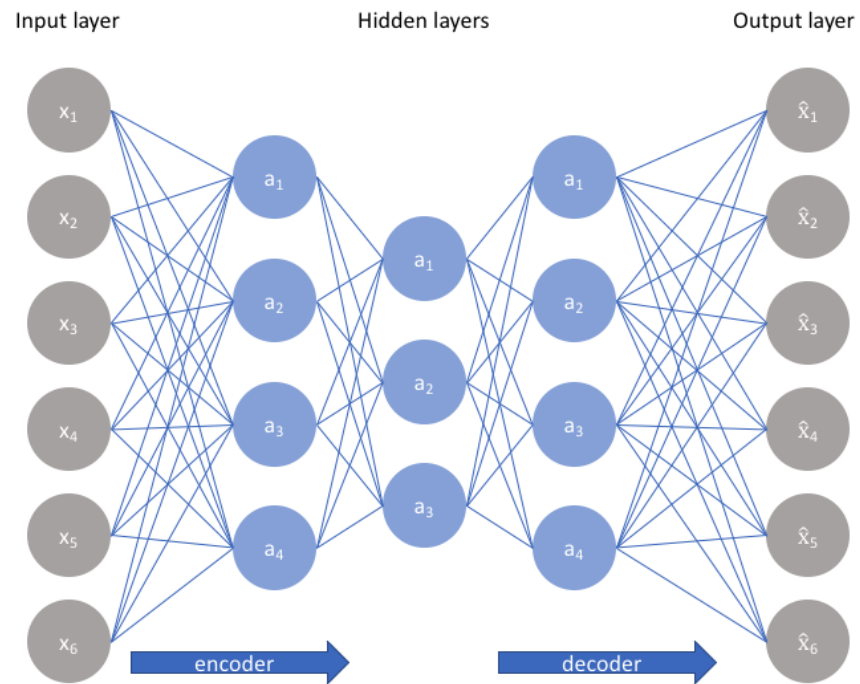
- feature를 압축하여 가능한 feature의 갯수를 줄인다.
- 많이 줄이지만 원본이 의미하고 있는 핵심은 포함해야함
- feature의 성질은 신경망에 의해 변경됨.
- Function
  - Encoder function :  $z = f(x)$
  - Decoder function :  $x = g(z)$
  - $x \rightarrow f(x) \rightarrow z \rightarrow g(z) \rightarrow \hat{x}$

## 동작

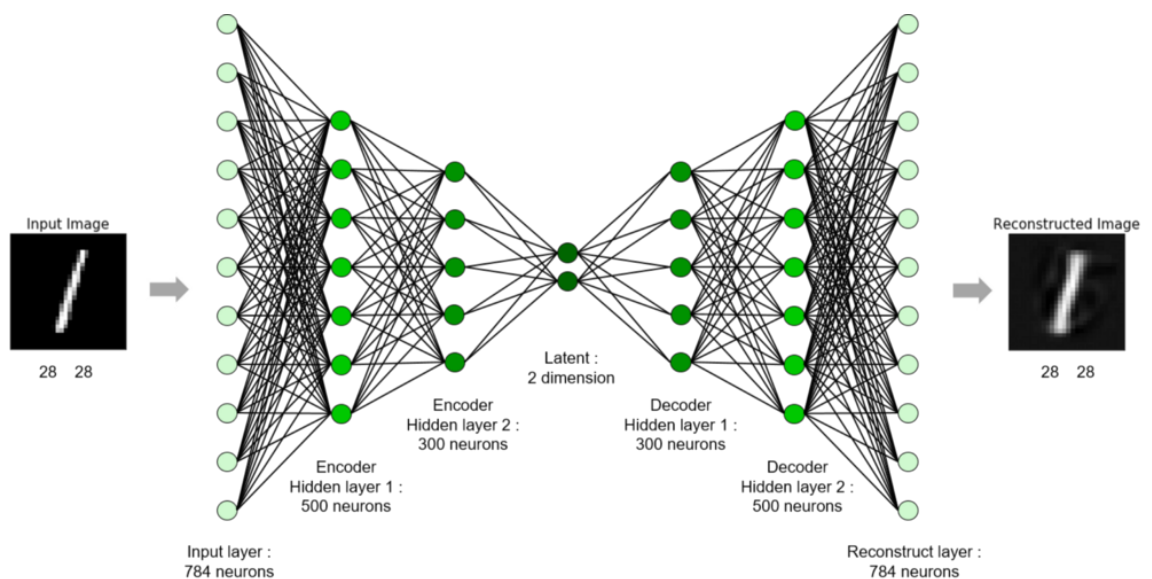
- 학습 원리 : input Data를 output Data 로 학습하는 방식으로 학습시킨다.
- 가장 쉬운 방법
  - 항등함수로 생성한 신경망
  - $f(x) = x$
  - AE가 항등함수를 학습하는것은 비효율 적이다.
  - 100% 복사 되지만 차원이 들어들지 않는다.



- Dimension reduction 이 포함 된 방식을 차용한다면?
  - 예) 6개의 Input Data 정보를 유지할수 있도록 Key가 되는 정보를 Feature Extraction 해준다.
  - Hidden Layers 에 제약조건을 주고 데이터를 입력한다면 중요한 요소들만 판별해 낼수 있다.(6 → 3 → 6)



- 3개의 데이터를 이용해 6개의 데이터를 유추한다. (즉, 3개는 각각 독립적인 요소들만 남게된다. 원진 모르겠지만...)
- Latent Space 는 입력 된 Data를 표현하는 가장 핵심적인 값들을 갖고있는 차원이다.



원본 그림

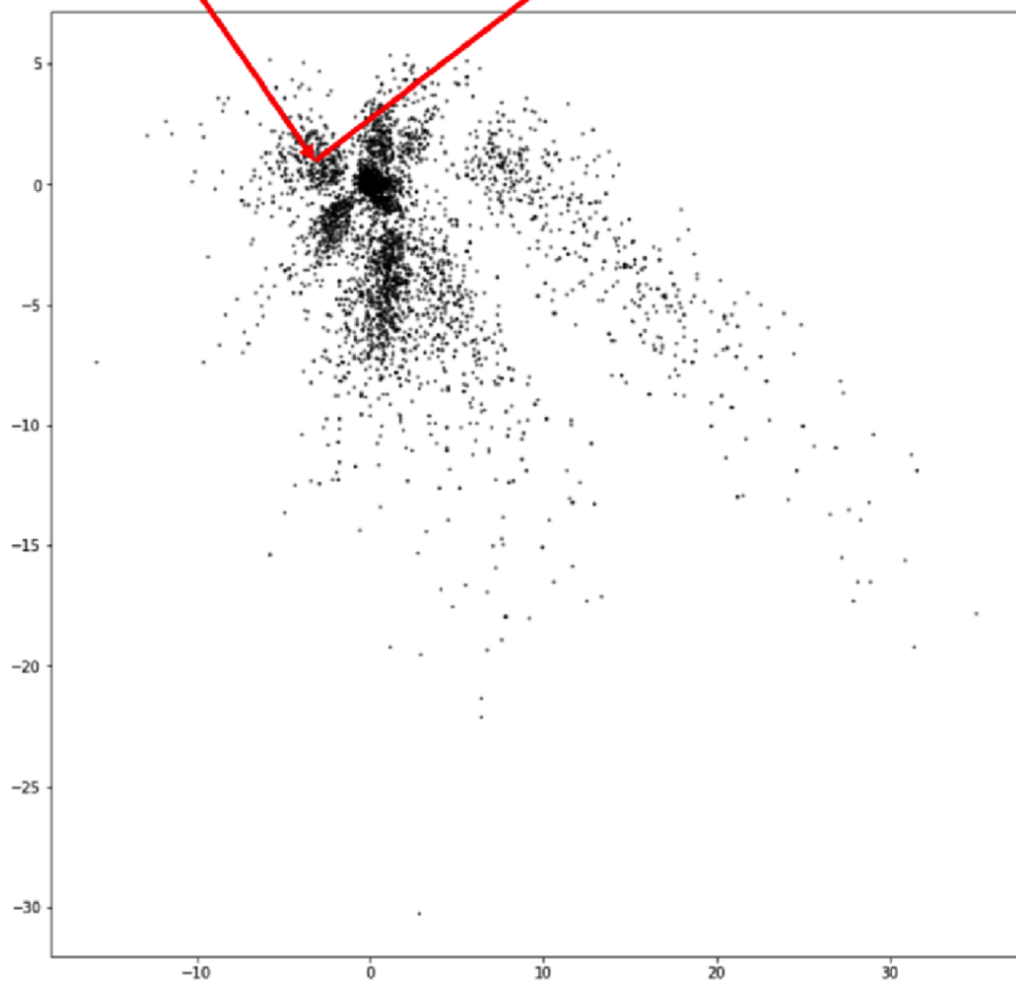


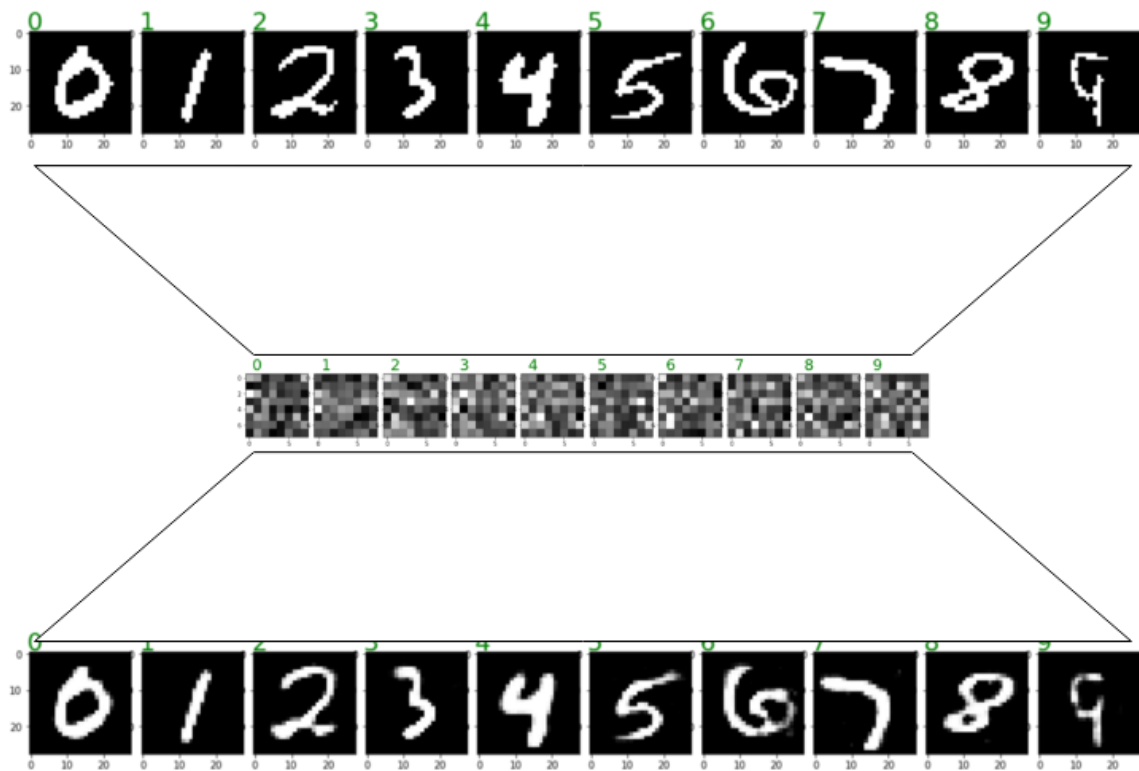
엔 코더 씨

재구성한 그림



디 코더 씨





## 손실함수

- $E[\|X - g \circ f(X)\|^2] \approx 0$ 
  - Encoder function :  $z = f(x)$
  - Decoder function :  $x = g(z)$
  - $x \rightarrow f(x) \rightarrow z \rightarrow g(z) \rightarrow \hat{x}$
- 입력과 Encoding → Decoding 된 결과에 차이에 대한 평균이 0에 가까워 지도록 설계하면 된다.

## 참고자료

<https://www.jeremyjordan.me/autoencoders/>

## 과소완전 AE

- AE 에서 Encoder의 역할은 원본데이터의 새로운 표현을 학습하는 구성요소 이다.
- 과소완전(undercomplete) 이란 내부의 표현이 입력 데이터보다 저차원이기 때문에 사용하는 용어 이다.
- Feature는 원본 피쳐 및 관측치 셋에서 파생된 새로운 피쳐셋이다.
  - Encoder :  $h = f(x)$ 
    - $x$  : 원본 관측치
    - $f$  : 학습되는 값
    - $h$  : 새롭게 학습된 feature
  - Decoder :  $r = g(h)$
  - 제대로 수행된다면  $g(f(x))$ 는 모든  $x$ 와 정확히 같지 않지만 충분히 가까운 값을 갖는다.
    - 즉,  $g(f(x)) \neq x$

---

## 과대완전 AE

- 인코더가 원본 입력 차원보다 더 많은 수의 차원으로 표현을 학습하는 경우
  - 신경망학습시 정규화를 이용해

### ▼ 질문

- 어떻게 인코더 함수를 근사 $x$ 로 제한해  $x$ 를 정확히 복사하지 않고  $x$ 의 핵심 특성만 학습하도록 할수 있을까?
  - 인코더 함수의 출력인  $h$ 를 원본 관측치  $x$  보다 더 작은 차원으로 제한한다.
  - 인코더의 차원 수가 원본 입력 차원 수보다 작기때문에 가능해 진다.
  - 차원 축소와 비슷한 과정
- 구조는 꼭 대칭이어야 할까?

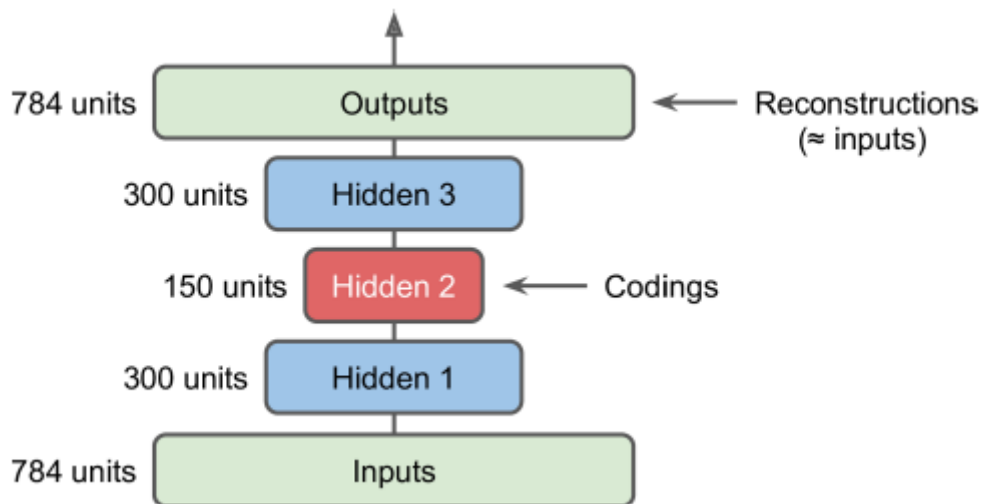
- 그렇지 않다.

## 손실함수

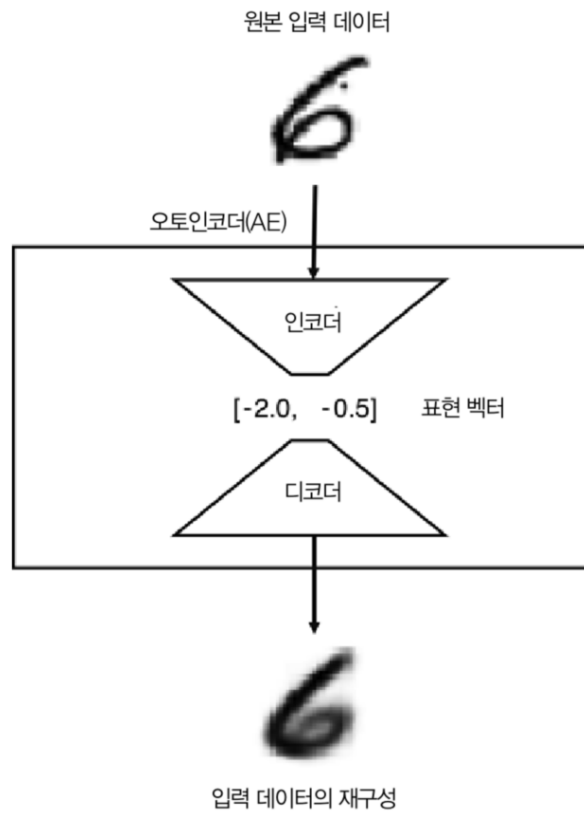
- 손실함수는 원본에 대한 재구성에 대해 각각의 Feature 들에 대해 평균 제곱근 오차 (RMSE) 나 binary-cross-entropy 를 사용한다.
  - 입력된  $x_1$  값과 출력된  $g(f(x_1))$  의 값이 최대한 같게 만들어 준다.
- 디코더가 선형이고 손실 함수가 평균제곱 오차인 경우, 과소완전 인코더는 PCA와 동일한 종류의 새로운 표현을 학습한다.
- 그러나 인코더 및 디코더 함수가 비선형인 경우 AE는 훨씬 더 복잡한 비선형 표현을 학습할 수 있다.

## 적층 AE

- 여러 은닉층을 가진 AE를 Stacked AE(또는 Deep) 라고 한다.
- MNIST 를 활용한 Stacked AE

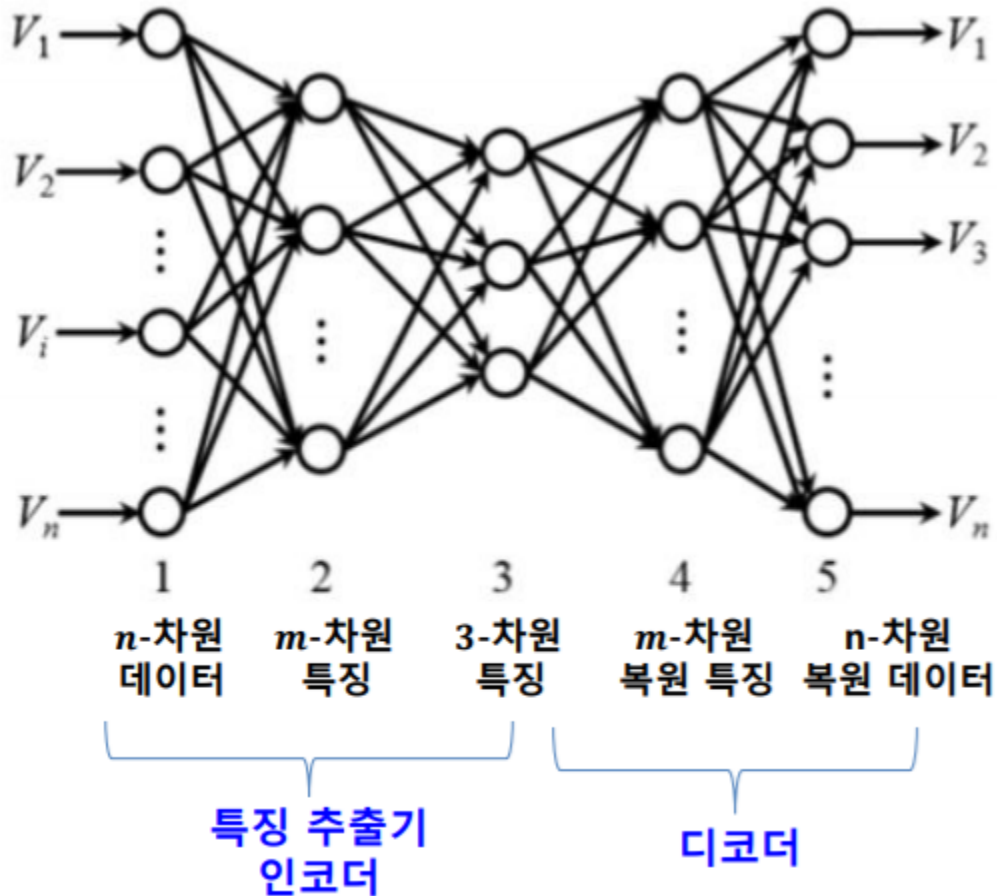






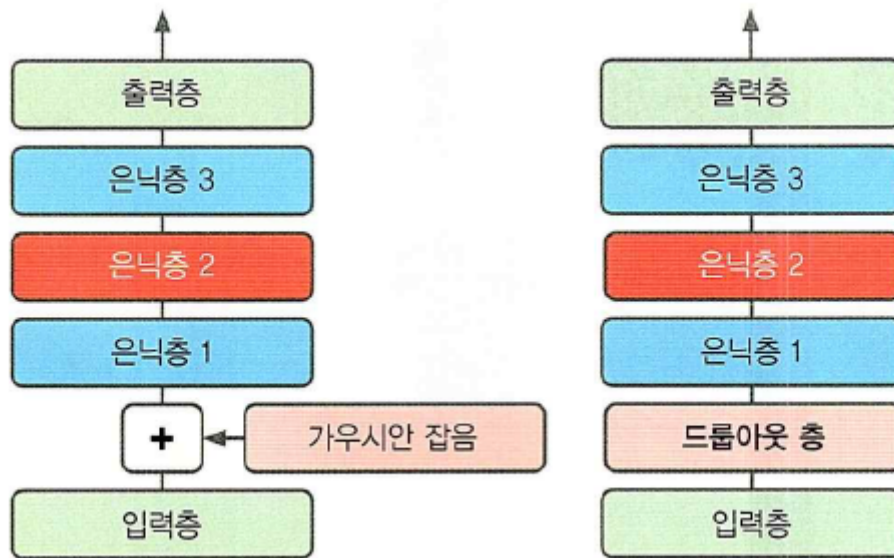
## Stacked AE 를 이용한 비지도 사전 훈련

### AE를 통한 특징 추출



## Denoising autoencoders

- 오토인코더는 원본 입력 데이터로부터 새로운 표현을 학습 가장 핵심적인 요소를 추출 원본 데이터의 노이즈는 무시.
- 배경에서 들려오는 잡음(노이즈)으로부터 대화(신호)를 분리할수 있다.
- 잡음은 입력에 추가된 순수한 가우시안 잡음 이거나 드롭아웃처럼 무작위로 입력을 꺼서 발생시킬 수도 있다.



- $\hat{x} = x + r$  :  $x$ (원본 데이터),  $r$ (무작위 잡음)
- 입력된 정보를 유지할수 있는 특징을 추출할수 있지만, 에 포함된 잡음을 제거 할수 있다.

