



Auto Encoder

오토인코더란 신경망의 한 종류로 인풋을 모델에 넣으면 이를 압축하여 유의미한 representation을 생성하고 이것을 다시 디코딩하여 가능한 원본과 유사한 상태로 재구축하는 것입니다.

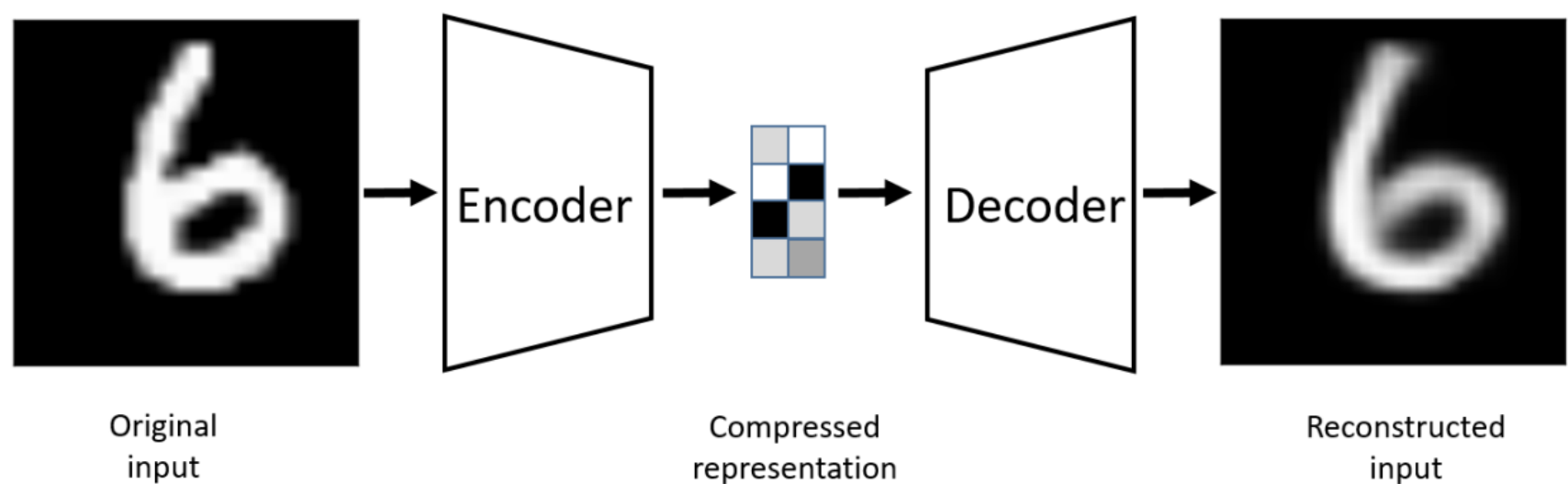


Fig. 1: An autoencoder example. The input image is encoded to a compressed representation and then decoded.

오토 인코더는 1986년에 처음 소개되어졌습니다. 이 때의 main contribution은 비지도 학습을 통해서 정보가 담긴 representation을 만들어내는 것이었습니다. 이 문제를 해결하는 방법은 다음과 같이 정의됩니다.

Encoder A : $R^n \gg R^p$

입력값을 받아 특징값으로 변환하는 신경망

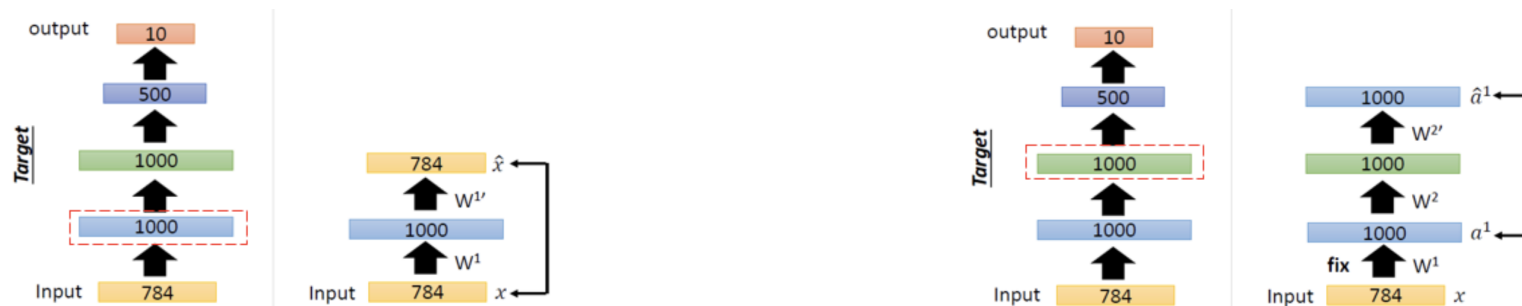
Decoder B : $R^p \gg R^n$

특징값을 받아 출력값으로 변환하는 신경망

목적 함수 $\arg \min_{A,B} E[\Delta(x, B \circ A(x))]$

여기서 x 는 input을 뜻하며, Δ 는 loss를 의미합니다. loss는 l2 norm을 따릅니다. 이 loss를 줄이기 위해 학습이 진행되며 Encoder의 Input과 Decoder output을 같게 만들기 위해 학습이 진행되어집니다.

오토인코더는 주로 input data feature를 추출하는 방식으로 사용되며 차원 축소에 활용되어집니다. Batch-Norm, parameter Initialization기법이 나오기 전 Network parameter초기화, pre-training에 많이 사용되어졌다고 합니다.



초기화 기법이 나오기 전, Auto encoder는 학습을 위해 하나씩 떼어서 따로 따로 학습을 진행했습니다. 그 다음 각각의 pretrained된 weight를 이용하여 전체적인 모델 학습을 진행합니다.

Encoder와 Decoder가 activation function없이 작동되는 경우도 존재합니다. 이와 같은 경우도 마찬가지로 latent representation을 PCA와 같이 얻을 수 있습니다. 그러므로 사실상 autoencoder는 PCA의 일반화라고도 볼 수 있으며 데이터의 저차원 설명을 가능케 해줍니다. 오토인코더는 end-to-end로 학습이 가능하며 layers가 stack된 형태로 볼 수 있습니다.

LATENT FEATURE

신경망 내부에서 추출된 특징점 값들

표현학습/특징학습이란?

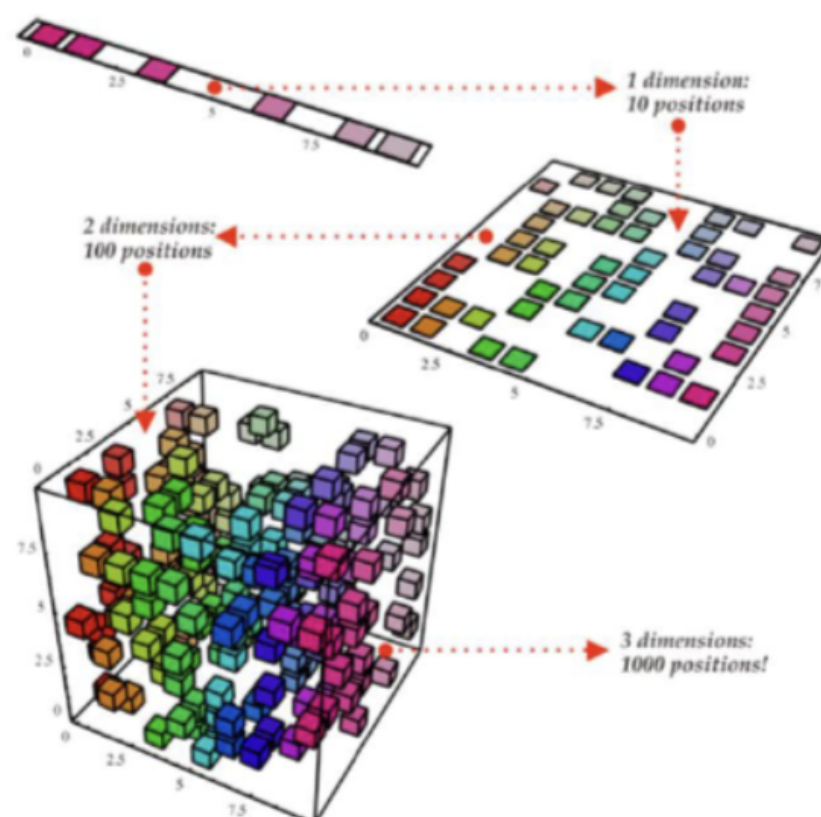
특징 탐지나 분류를 위해 필요한 특징점들을 자동으로 발견하는 시스템적 기법들의 총체를 일컫는다.

차원을 줄이는 이유가 무엇일까? → 차원의 저주

차원의 저주

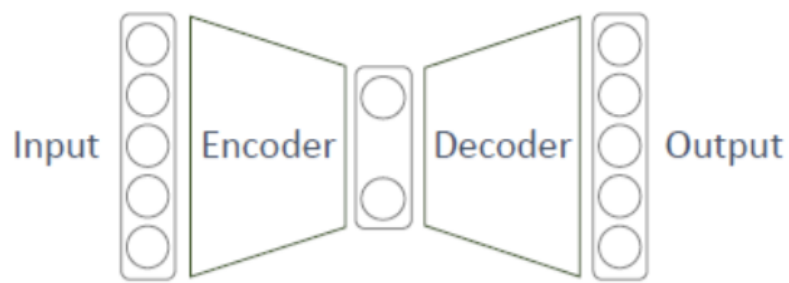
데이터의 차원이 증가할수록 해당 공간의 크기(부피)가 기하급수적으로 증가하기 때문에 동일한 개수의 데이터의 밀도는 차원이 증가할수록 급속도로 희박해진다.

따라서, 차원이 증가할수록 데이터의 분포 분석 또는 모델추정에 필요한 샘플 데이터의 개수가 기하급수적으로 증가하게 된다.



http://videolectures.net/kdd2014_bengio_deep_learning/
<http://darkpgmr.tistory.com/145>

고차원 공간의 데이터를 분석하거나 측정할 때 저차원 공간에서는 나타나지 않았던 여러 문제들이 발생하는 것으로, 머신러닝에서는 일반적으로 차원이 증가할 경우 기하급수적인 데이터가 요구되는 현상을 말한다.



[4 MAIN KEYWORDS]

1. Unsupervised learning
2. Manifold learning
3. Generative model learning
4. ML density estimation

오토인코더를 학습할 때:

학습 방법은 비교사 학습 방법을 따르며, -----> Unsupervised learning
 Loss는 negative ML로 해석된다. -----> ML density estimation

학습된 오토인코더에서:

인코더는 차원 축소 역할을 수행하며, -----> Manifold learning
 디코더는 생성 모델의 역할을 수행한다. -----> Generative model learning

추가적인 관련

모델의 학습 방법

loss를 줄이는 것이다!

아래는 한개의 레이어를 가지는 모델을 통해 학습을 진행하는 것이다.

OBJECT

$$C = (a - y)^2 / 2 = a^2 / 2$$

$$\nabla_a C = (a - y)$$

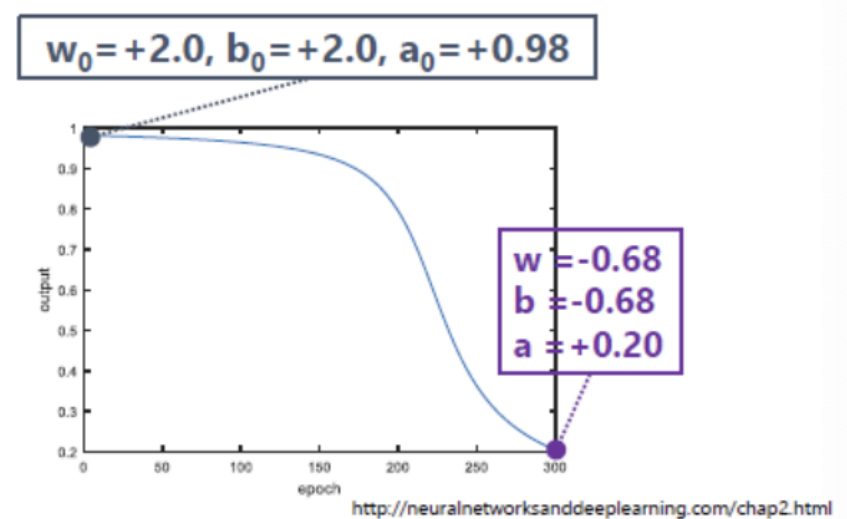
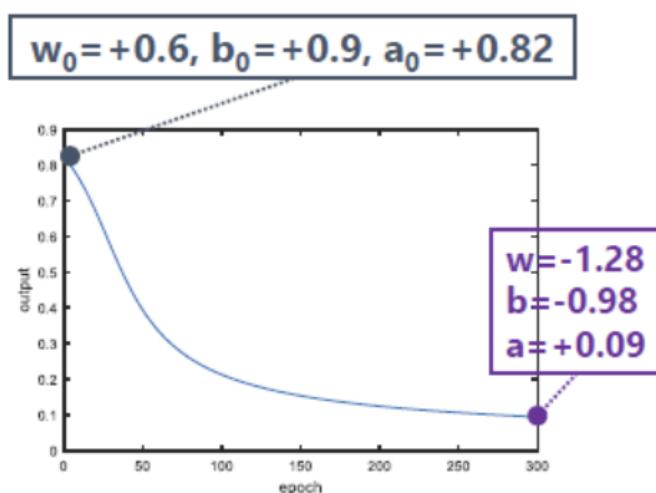
$$\delta = \nabla_a C \odot \sigma'(z) = (a - y) \sigma'(z)$$

$$\frac{\partial C}{\partial w} = x\delta = \delta$$

$$w = w - \eta\delta$$

$$\frac{\partial C}{\partial b} = \delta$$

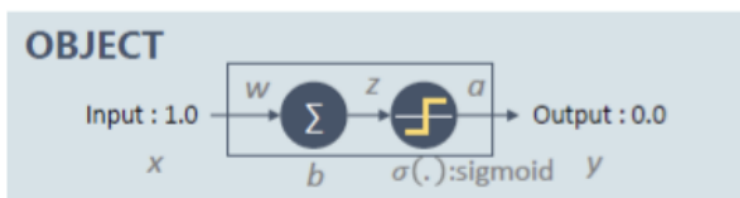
$$b = b - \eta\delta$$



Learning Slow & Learning Fast...!

단 한개의 레이어를 가지며 같은 값으로 추론을 하며 모델은 같은데 왜 학습의 속도가 다를까?

그 이유는 바로 초깃값이다.



$$C = (a - y)^2 / 2 = a^2 / 2$$

$$\nabla_a C = (a - y)$$

$$\delta = \nabla_a C \odot \sigma'(z) = (a - y) \sigma'(z)$$

$$\frac{\partial C}{\partial w} = x \delta = \delta$$

$$\frac{\partial C}{\partial b} = \delta$$

$$w = w - \eta \delta$$

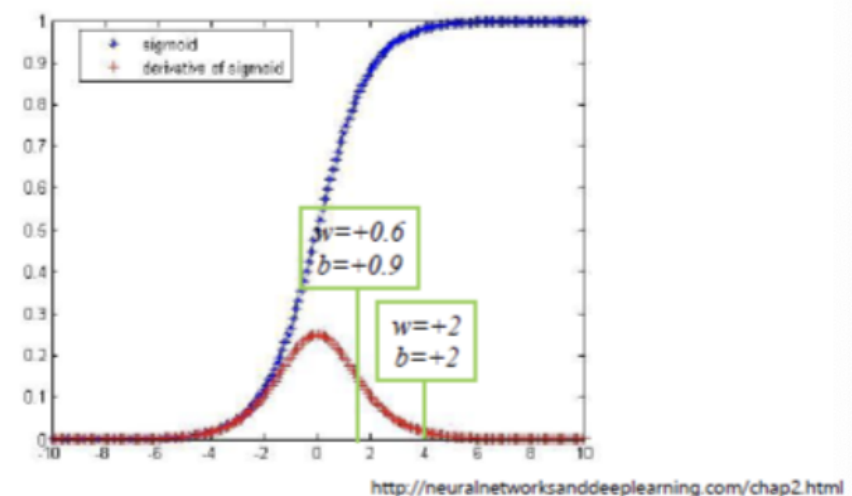
$$b = b - \eta \delta$$

Learning slow means are $\partial C / \partial w, \partial C / \partial b$ small !!

Why they are small???

$$\frac{\partial C}{\partial w} = x \delta = x a \sigma'(z) = a \sigma'(z)$$

$$\frac{\partial C}{\partial b} = \delta = a \sigma'(z)$$



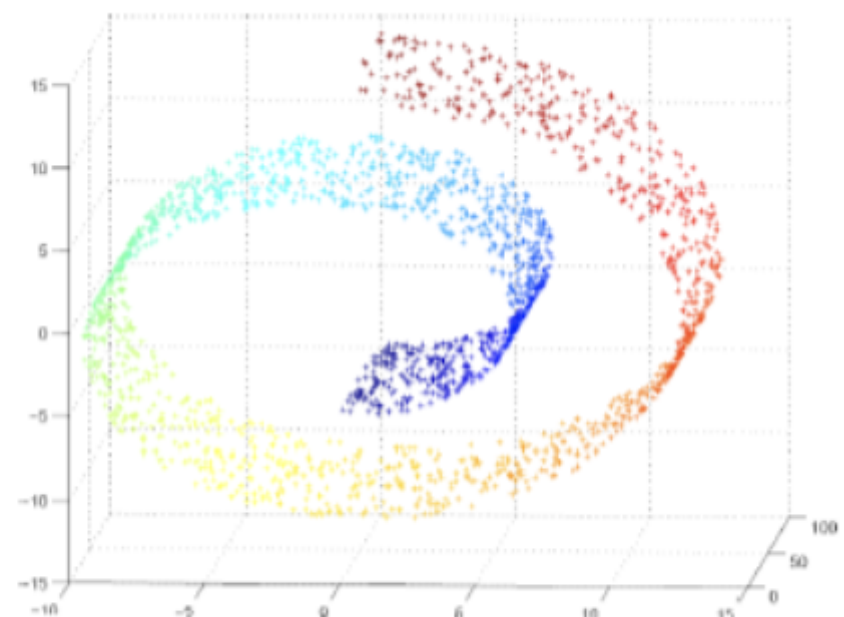
왼쪽의 학습 초깃값의 기울기가 약 0.2, 오른쪽의 학습 초깃값의 기울기가 약 0.01 정도로 이 차이는 크며 이에 따라 전파되는 기울기의 크기가 달라 학습 속도가 달라진 것이다. 그만큼 초깃값이 중요하며 잘 샘플링된 데이터는 좋은 결과를 가져온다는 것을 확인할 수 있다. 우리는 이를 생각해 보면 초깃값을 레이턴트 벡터라고 생각하면, 오토 인코더의 인코더 부분은 초깃값을 결정하는 모델이라 볼 수 있을 것이다. (분포) → 이는 이후 VAE에서 중요한 부분으로 다뤄진다.

Manifold Learning

Manifold란 고차원 데이터를 공간에다가 뿌리면 sample들을 잘 아우르는 subspace가 있을 거라는 가정하에서 학습이 진행되며, 데이터의 차원을 축소시키는 학습 방식입니다.

Manifold는 데이터 압축, 데이터 시각화, 차원의 저주, 중요 피쳐 파악에서 사용됩니다.

Manifold는 고차원의 데이터는 밀도는 낮지만, 고차원의 데이터를 포함하는 저차원의 매니폴드가 있다고 생각하는 것입니다.



http://videlectures.net/deeplearning2015_vincent_autoencoders/?q=vincent%20autoencoder

우리는 이러한 매니폴드를 찾아서, 샘플 데이터를 추출할 수 있습니다. 그리고 매니폴드 내에서 좌표들이 조금씩 변할 때 원 데이터도 유의미한 변화를 보인다고 합니다. 이러한 경우 우리는 원본 데이터와 유사한 sample을 추출할 수 있게 됩니다.

→ 저는 여기서 manifold란 AutoEncoder의 Encoder의 파라미터 과정 자체를 manifold차원이라 생각했습니다.

DAE (Denoising Auto Encoder)

denoising AE는 input data의 random noise를 추가하는 방식입니다. output data인 원본은 그대로 두고 학습을 진행합니다. DAE는 noise가 추가되어 있는 data에서부터 학습을 하기 때문에 모델은 중요한 패턴들을 파악해내며 원본 데이터 복구를 더 강하게 학습합니다. 이에 따라 더욱 더 좋은 성능을 갖게 됩니다.

결론 AutoEncoder는 Manifold Learning을 위함!

참고

<https://www.youtube.com/watch?v=YxtzQbe2UaE>

Autoencoders <https://arxiv.org/abs/2003.05991>

<https://deepinsight.tistory.com/123>

<https://www.slideshare.net/NaverEngineering/ss-96581209>