

Lecture 24: Autoencoder

[SCS4049-02] Machine Learning and Data Science

Seongsik Park (s.park@dgu.edu)

AI Department, Dongguk University

Tentative schedule

week	topic	date (수 / 월)
1	Machine Learning Introduction & Basic Mathematics	09.02 / 09.07
2	Python Practice I & Regression	09.09 / 09.14
3	AI Department Seminar I & Clustering I	09.16 / 09.21
4	Clustering II & Classification I	09.23 / 09.28
5	Classification II	(추석) / 10.05
6	Python Practice II & Support Vector Machine I	10.07 / 10.12
7	Support Vector Machine II & Decision Tree and Ensemble Learning	10.14 / 10.19
8	Mid-Term Practice & Mid-Term Exam	10.21 / 10.26
9	Dimensional Reduction I	(휴강) / 11.02
10	Dimensional Reduction II & Neural networks and Back Propagation I	11.04 / 11.09
11	Neural networks and Back Propagation II & III	11.11 / 11.16
12	AI Department Seminar II & Convolutional Neural Network	11.18 / 11.23
13	Python Practice III & Recurrent Neural network	11.25 / 11.30
14	Autoencoder	(휴강) / 12.07
15	Final Exam Practice & Final Exam	12.09 / 12.14

Final Exam: 비대면

- 일시
 - 날짜: 12월 14일 (월)
 - 시간: 10:00 - 11:50 (110분, 연장 없음)
 - 미팅룸을 먼저 나갈 수 없음: 동시 시작, 동시 종료
 - 자세한 일정은 [Table 1: 시험 일정] 참조
- 장소: 비대면 webex (수업 미팅룸)
- 범위
 - 중간고사 이전: 중간고사 문제 중에서
 - 중간고사 이후: 강의자료, 수업필기, 실습수업, 숙제 등
 - 프로그래밍 포함, 학과세미나 제외
- 방식 및 배점
 - Closed book, 중간고사와 동일한 방식
 - 100점 만점, 별도의 카메라 모범 설치 보너스 +5점
 - 2지선다: 출제 중
 - 단답형: 출제 중
 - 서술형: 출제 중

Table 1: 시험 일정

시간	해야할 일	비고
시험 시작 전부터 10시 00분까지	카메라 설정 완료 미팅룸 입장 완료	준비에 도움이 필요하면 사전에 담당 교수에게 문의
10시 00분부터 카메라 설정 확인시까지	카메라 설정 확인	수강생 본인 책임 하에 준비되지 않은 사람은 시험 자격 박탈 최대 10시 10분을 넘기지 않으며 그때까지 협조하지 않는 경우 자격 박탈
카메라 설정 확인시부터 11시 35분까지	시험 응시	
11시 35분부터 11시 40분까지	답안을 카메라에 차례로 보여줌	시험 일괄 종료
11시 40분부터 11시 50분 59초까지	답안을 스캔하여 메일 전송	메일 도착 시간 기준 11시 51분 00초부터 0점 처리

Autoencoder

Autoencoder

- artificial neural networks capable of learning efficient representations of the input data, called *codings*, without any supervision
- these codings typically have a much lower dimensionality than the input data, making autoencoders useful for dimensionality reduction
- autoencoders act as powerful feature detectors, and they can be used for unsupervised pretraining of deep neural networks
- they are capable of randomly generating new data that looks very similar to the training data; this is called a *generative model*

Autoencoder

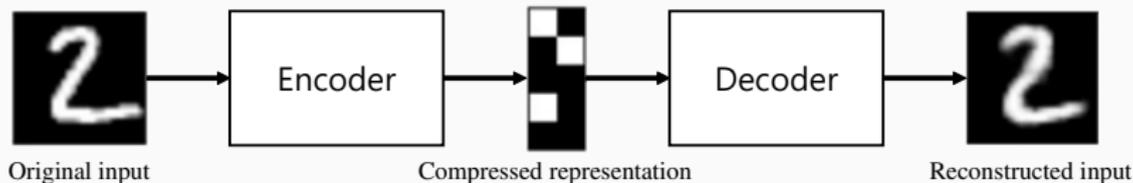
Surprisingly, autoencoders work by simply learning to copy their inputs to their outputs. This may sound like a trivial task, but we will see that constraining the network in various ways can make it rather difficult. For example,

- you can limit the size of the internal representation
- you can add noise to the inputs and train the network to recover the original inputs

These constraints prevent the autoencoder from trivially copying the inputs directly to the outputs, which forces it to learn efficient ways of representing the data. In short, the codings are byproducts of the autoencoder's attempt to learn the identity function under some constraints.

Efficient data representations

- 체스 판의 말 위치 기억하기
 - 기억, 인지, 패턴 매칭 간의 관계
 - 실제 경기의 배치 vs 무작위적인 배치
- 단순 기억 보다는 패턴에 의한 기억이 유용
- Autoencoder는 입력의 표상을 만들고(encoding), 이것을 재현(decoding)하도록 학습



Efficient data representations

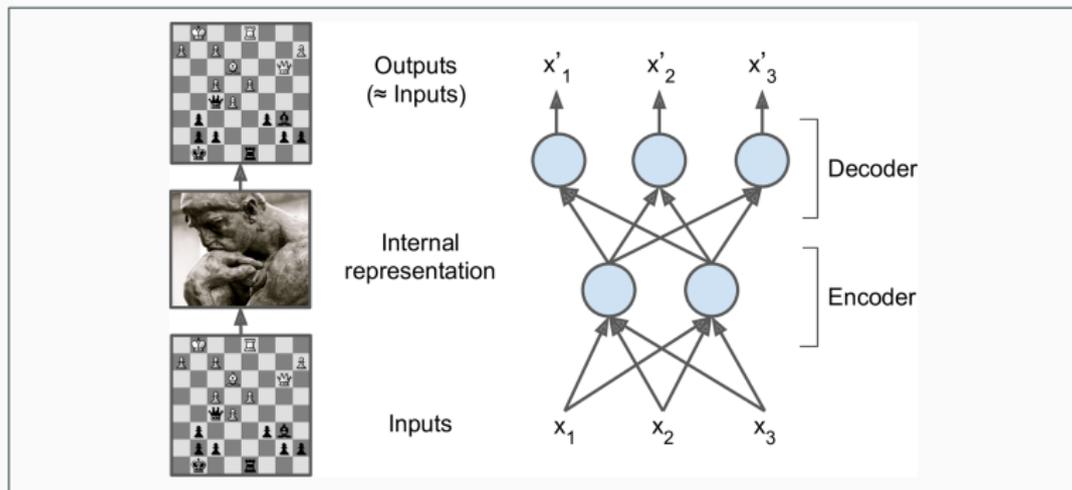
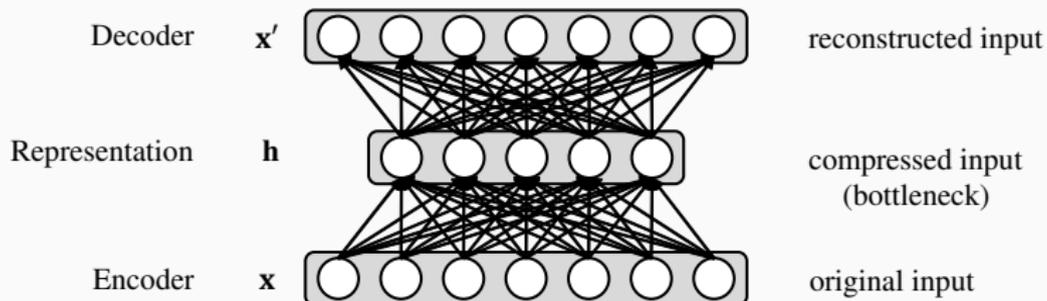


Figure 15-1. The chess memory experiment (left) and a simple autoencoder (right)

Basic autoencoder



$$\text{Encoder: } \mathbf{h} = s_1(\mathbf{W}\mathbf{x}) \quad (1)$$

$$\text{Decoder: } \mathbf{x}' = s_2(\mathbf{W}\mathbf{h}) \quad (2)$$

- activation function, s_1, s_2 (e.g., element-wise sigmoid)
- parameters of encoder: weight, \mathbf{W}
- parameters of decoder: *tied weight*, \mathbf{W}

Basic autoencoder

For MNIST dataset having 784 inputs (28×28 image),

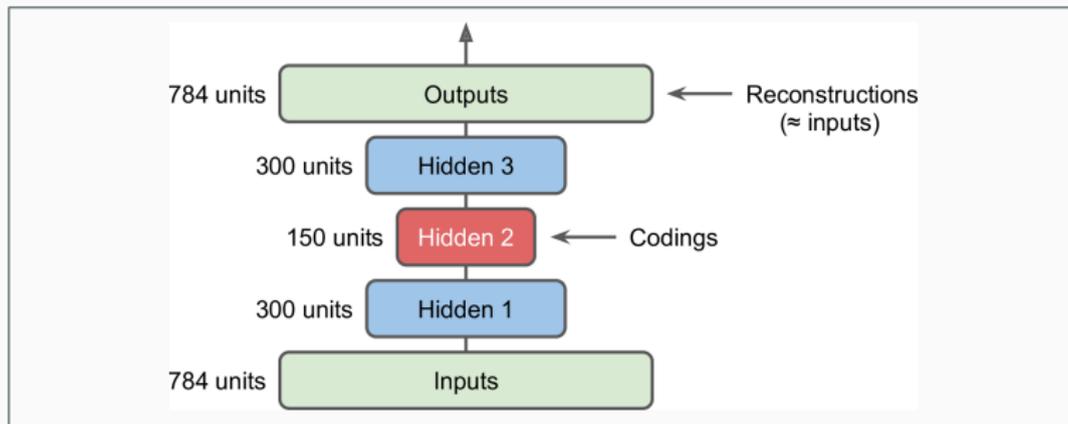


Figure 15-3. Stacked autoencoder

Basic autoencoder

Loss function: reconstruction error

- Squared error (i.e., Euclidean distance² between \mathbf{x} and \mathbf{x}')

$$L(\mathbf{x}, \mathbf{x}') = \|\mathbf{x} - \mathbf{x}'\|^2 \quad (3)$$

- Bernoulli cross entropy

$$L(\mathbf{x}, \mathbf{x}') = - \sum_{i=1}^D x_i \log x'_i + (1 - x_i) \log(1 - x'_i) \quad (4)$$

Dimension of hidden layer

- Hidden layer의 차원이 입력/출력 layer보다 작음 (undercomplete)
⇒ dimensionality reduction
- Hidden layer의 차원이 더 큰 경우 (overcomplete)
⇒ sparse autoencoder
 - 특별한 조치가 없다면 identity function을 학습하게 될 것
 - 활성화되는 hidden unit의 수를 적어지도록 규제

Sparse autoencoder

Hidden unit의 수가 input unit의 수보다 큰 경우

- Hidden unit에 sparsity constraint를 부가하여 가능한 많은 hidden unit의 activation이 0이 되도록 함

$$\hat{\rho}_j = \frac{1}{N} \sum_{i=1}^N h_j^{(i)} \quad (5)$$

- $h_j^{(i)}$: i 번째 input vector $\mathbf{x}^{(i)}$ 에 대한 j 번째 hidden unit의 activation값
- Training dataset $\mathcal{D} = \{\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \dots, \mathbf{x}^{(N)}\}$

Sparsity parameter ρ

$$\hat{\rho}_j = \rho \quad (6)$$

- 즉, 모든 hidden unit h_j 에 대해서 training set 전체에서의 평균값이 근사적으로 작은 수인 ρ 에 가깝게 제약함
- 이 제약을 만족시키려면 $h_j^{(i)}$ 는 대부분 0에 가까운 값을 가져야함

Regularization term

$$\sum_{j=1}^M \text{KL}(\rho \parallel \hat{\rho}_j) = \sum_{j=1}^M \rho \log \frac{\rho}{\hat{\rho}_j} + (1 - \rho) \log \frac{1 - \rho}{1 - \hat{\rho}_j} \quad (7)$$

- $\text{KL}(\rho \parallel \hat{\rho}_j)$: Kullback-Leibler divergence
- M : hidden unit의 수

Loss function

$$L_{\text{sparse}}(\mathbf{x}, \mathbf{x}') = L(\mathbf{x}, \mathbf{x}') + \beta \sum_{j=1}^M \text{KL}(\rho \parallel \hat{\rho}_j) \quad (8)$$

$$\text{where hyperparameter } \beta \quad (9)$$

Sparse autoencoder

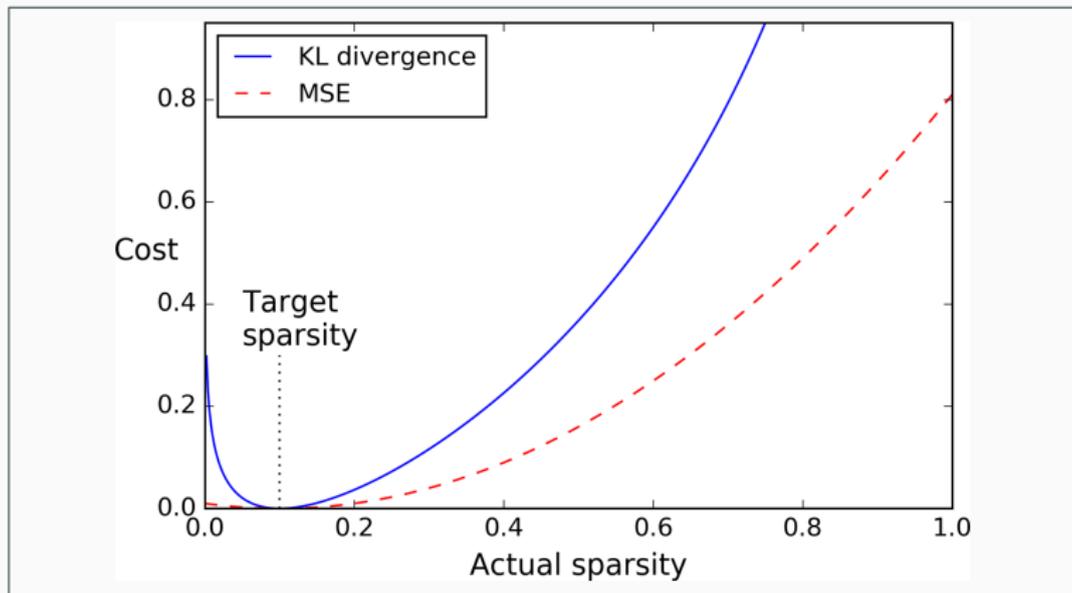


Figure 15-10. Sparsity loss

Unsupervised pretraining using autoencoder

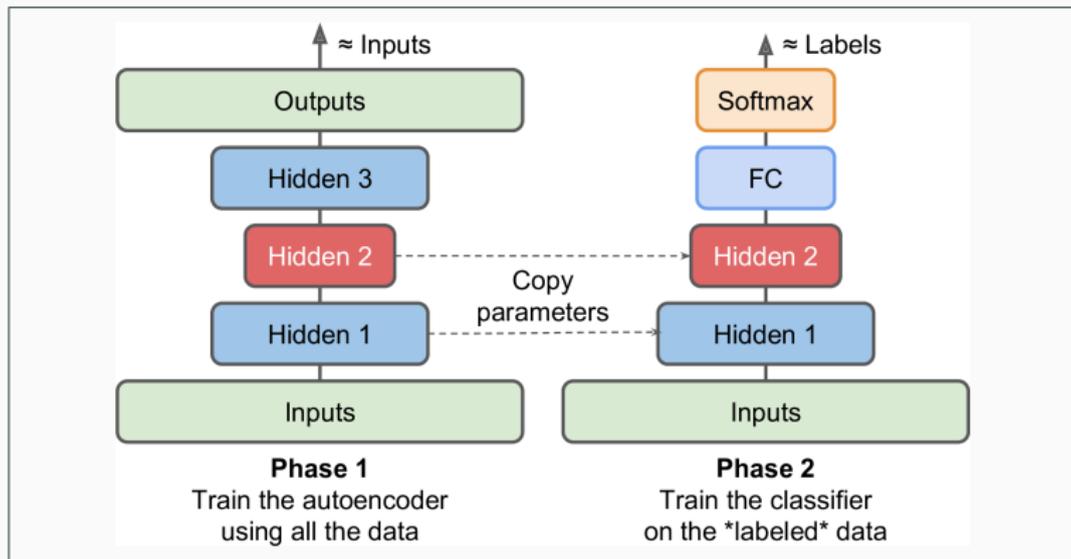


Figure 15-8. Unsupervised pretraining using autoencoders

Denoising autoencoder

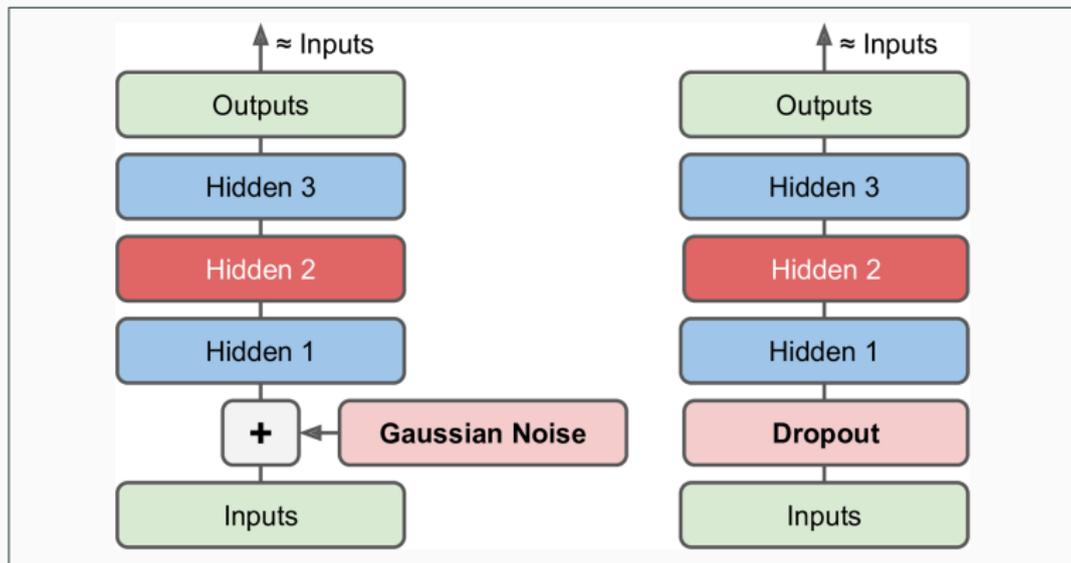


Figure 15-9. Denoising autoencoders, with Gaussian noise (left) or dropout (right)

Variational autoencoder

Variational autoencoder

- Probabilistic autoencoders: denoising autoencoder에서는 훈련하는 동안만 무작위성(randomness)을 이용하지만 variational autoencoder는 훈련 이후에도 무작위성을 이용
- Generative autoencoders: training set에서 뽑은 것 같은 새로운 샘플들을 생성할 수 있음

Variational autoencoder의 학습

- 주어진 입력에 대한 코딩을 직접 출력하기 보다는 코딩의 평균 μ , 표준편차 σ 를 출력
- 그러면 실제 코딩은 Gaussian 분포 $\mathcal{N}(\mu, \sigma^2)$ 에서 무작위로 추출

Variational autoencoder

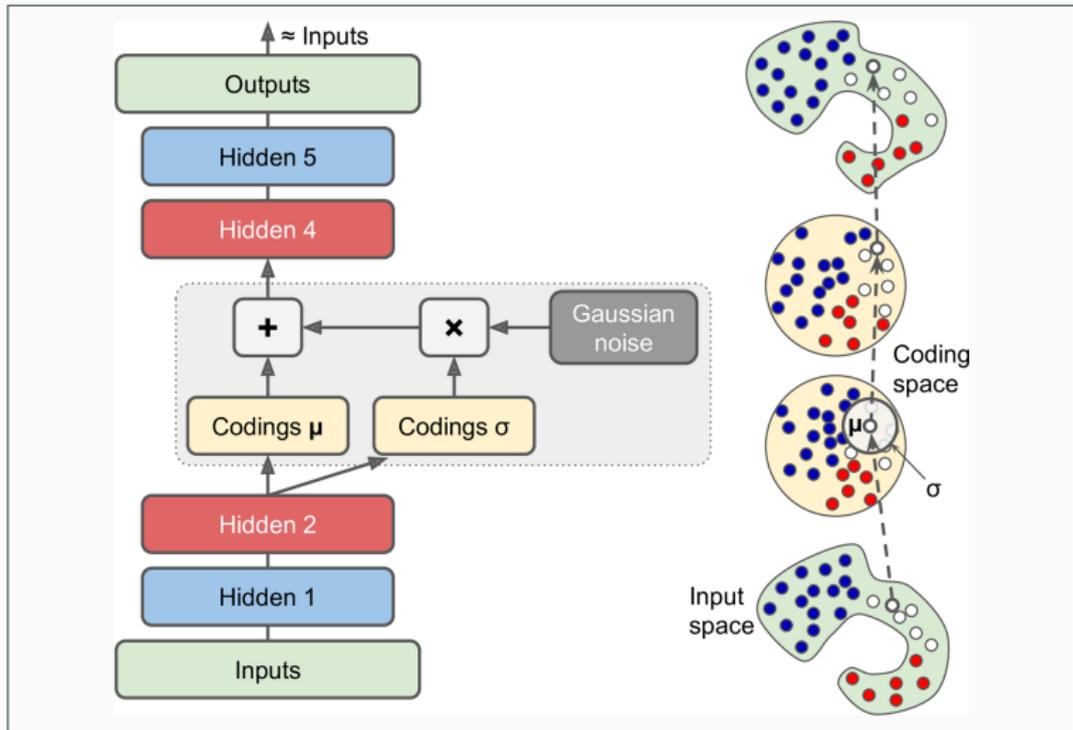


Figure 15-11. Variational autoencoder (left), and an instance going through it (right)

Appendix

Reference and further reading

- “Chap 15 | Autoencoders” of A. Geron, Hands-On Machine Learning with Scikit-Learn and TensorFlow
- “Lecture 14 | Autoencoders” of Kwang Il Kim, Machine Learning (2019)