SVD 알고리즘

SVD를 사용해서 아이템 기반 평점을 행렬 기반으로 예측해보자

사용자/아이템 기반 협업 필터링의 문제점

- 확장성(scalability): 큰 행렬 계산은 여러모로 쉽지 않음
  - 하지만 아이템 기반으로 가면 계산량이 줄어듬
  - Spark을 사용하면 큰 행렬 계산도 얼마든지 가능
- 부족한 데이터(sparse data)
  - 많은 사용자들이 충분한 수의 리뷰를 남기지 않음
- 해결책 -> 모델 기반 협업 필터링
  - 머신 러닝 기술을 사용해 평점을 예측. 입력은 사용자-아이템 평점 행렬
    - 행렬 분해(Matrix Factorization) 방식
    - 딥러닝 방식

행렬 분해 방식

- 협업 필터링 문제를 사용자-아이템 평점 행렬을 채우는 문제로 재정의
  - 아래 그림에서 ?를 채우는 문제!
  - 사용자 혹은 아이템을 적은 수의 차원으로 기술(차원수 축소)함으로써 문제를 간단화
- 가장 많이 사용되는 행렬 분해 방식
  - PCA(Principal Component Analysis)
  - SVD (Singular Vector Decomposition) 흔히 SVD라고

	scifi1	scifi2	scifi3	comedy1	comedy2	comedy3
user1	4.0	5.0	3.0	NaN ?	2.0	1.0
user2	5.0	3.0	3.0	2.0	2.0	NaN ?
user3	1.0	NaN ?	NaN ?	4.0	5.0	4.0
user4	NaN ?	2.0	1.0	4.0	NaN ?	3.0
user5	1.0	NaN ?	2.0	3.0	3.0	4.0

PCA (Principal Component Analysis)

- 차원을 축소(Dimensionality Reduction)하되 원래 의미는 최대한 그대로 간직

	scifi-1	scifi-2	comedy-1	comedy-2	action-1
user1	3.0	5.0	1.0	2.0	4.0
user2	4.0	4.0	2.0	3.0	4.0
user3	1.0	2.0	4.0	5.0	2.0

PCA

숨어있는 특징 (Latent Factors)을 찾아내기

	scifi	comedy	action
user1	4.0	1.5	4.0
user2	4.0	2.5	4.0
user3	1.5	4.5	2.0

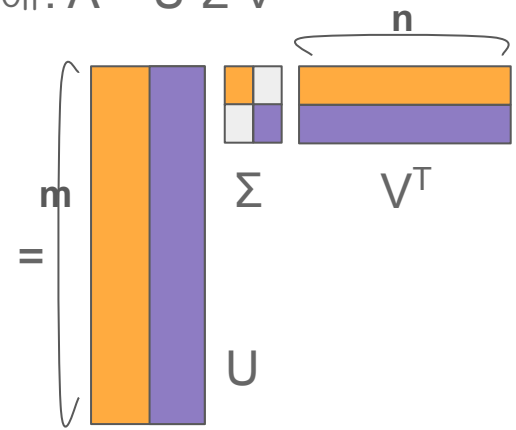
SVD (Singular Vector Decomposition)

- 2개 혹은 3개의 작은 행렬의 곱으로 단순화 (일종의 소인수 분해)
  - PCA와 같은 차원축소 알고리즘이지만 다른 방식
- SVD vs. SVD++

사용자/영화 행렬의 분해: $A = U \Sigma V^T$

	scifi1	scifi2	scifi3
user1	4.0	5.0	3.0
user2	5.0	3.0	3.0
user3	1.0	NaN	NaN
user4	NaN	2.0	1.0
user5	1.0	NaN	2.0

m, n



- 사용자 대 아이템 행렬을 훨씬 작은 3개의 행렬(U, Σ, V^T)로 나눠 표현
  - U: 사용자를 나타내는 훨씬 작은 행렬
  - Σ: U의 특정 부분을 시그널을 증폭시켜주는 행렬 (대각 행렬). 여기를 보면 덜 중요한 부분 파악 가능
  - V^T: 아이템을 나타내는 훨씬 작은 행렬 (치환)

SVD++ (Singular Vector Decomposition)

- SVD++: 넷플릭스 컨테스트때 고안된 추천방식
  - SVD나 PCA는 완전하게 채워져있는 행렬의 차원수를 줄이는 방식
  - SVD++는 sparse 행렬이 주어졌을 때 비어있는 셀들을 채우는 방법을 배우는 알고리즘
    - 채워진 셀들의 값을 최대한 비슷하게 채우는 방식으로 학습 (에러률을 최소화)
      - 보통 RMSE의 값을 최소화하는 방식으로 학습하면서 SGD를 사용
    - A(사용자 대 아이템 평점 행렬 u x n)를 입력으로 주면 U, Σ, V^T를 찾아냄
      - U: 사용자 행렬 (u x r)
      - Σ: 스케일 행렬 (r x r)
      - V^T: 아이템 행렬 (r x n)

    $A = U\Sigma V^T$
- surprise 라이브러리를 사용하거나 scikit-learn의 TruncatedSVD를 사용

SVD 기반 추천 엔진 실습

Surprise 라이브러리를 이용해서
SVD 기반 추천 엔진을 만들어보자

실습 소개

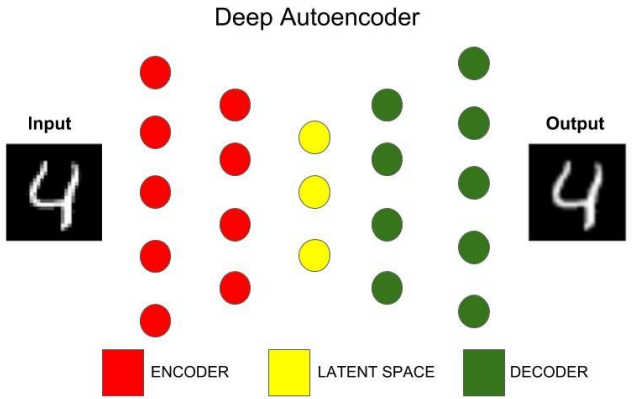
- 앞서 소개한 Surprise 라이브러리와 무비렌즈 데이터셋을 이용
  - 평점을 예측하는 모델 개발
- 모델의 성능 평가
  - Test/Train 방식
  - https://colab.research.google.com/drive/1ivoRqcwTjIwiUPlHuwsCoUdAoth9JAK7?usp=sharing

오토인코더 소개

딥러닝 오토인코더가 무엇인지 배워보자

오토 인코더란 (1)

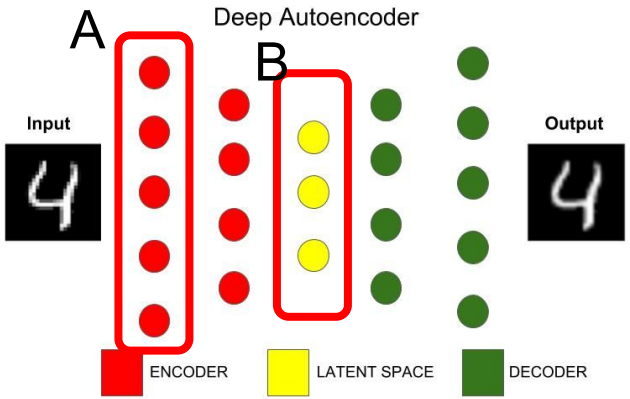
- 대표적인 비지도학습을 위한 딥러닝 모델
  - 데이터의 숨겨진 구조를 발견하면서 노드의 수를 줄이는 것이 목표
  - 입력 데이터에서 불필요한 특징들을 제거한 압축된 특징을 학습하려는 것
  - 오토인코더의 출력은 입력을 재구축한 것임
    - 최대한 비슷하게 나오도록 학습
    - 입력 데이터와 예상 출력 데이터가 동
      - 입력 == 레이블



https://socr.umich.edu/HTML5/ABIDE_Autoencoder/

오토 인코더란 (2)

- 오토인코더의 구조
  - 출력층의 노드 개수와 입력층의 노드 개수가 동일해야함
  - 은닉층의 노드 개수가 출력층/입력층의 노드 개수보다 작아야함
- 이렇게 학습된 은닉층의 출력을 입력을 대신하는 데이터로 사용
  - 데이터의 크기 축소. 그림에서 A 대신 B를 사용

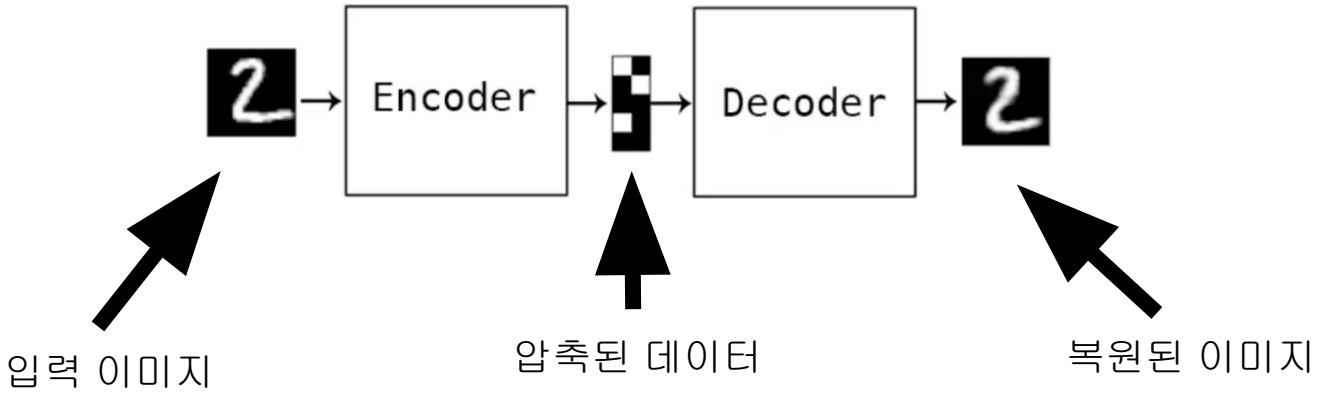


오토 인코더 실습

딥러닝 오토인코더를 만들어보자

실습 소개 (1)

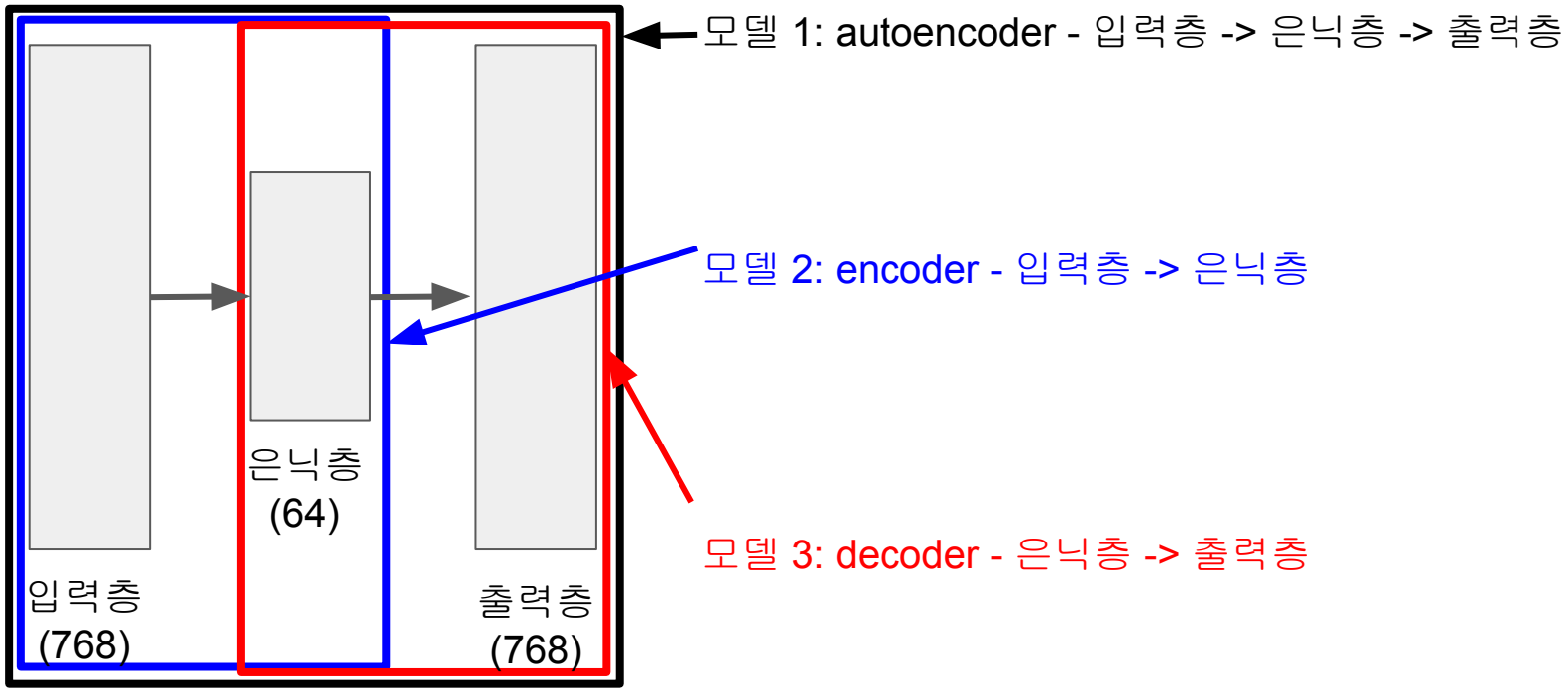
- 케라스를 사용하여 숫자 이미지를 인코딩했다가 디코딩하는 오토인코더 만들기



https://blog.keras.io/building-autoencoders-in-keras.html

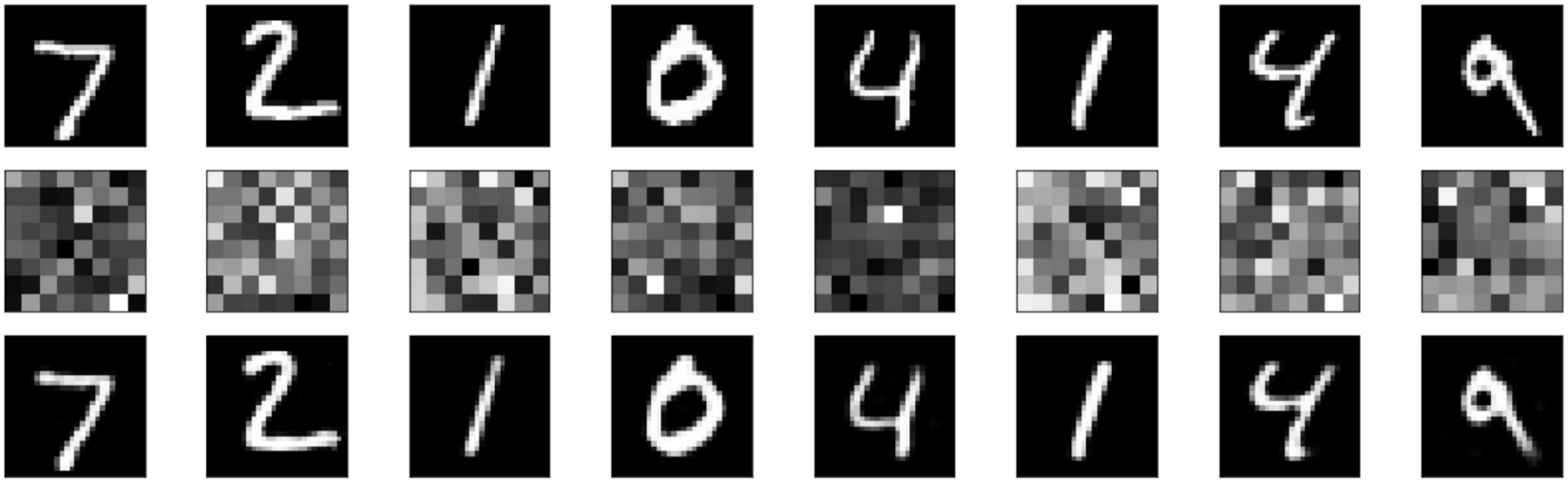
실습 소개 (2)

- 모델 소개


모델 1: autoencoder - 입력층 -> 은닉층 -> 출력층
모델 2: encoder - 입력층 -> 은닉층
모델 3: decoder - 은닉층 -> 출력층
은닉층
(64)
입력층
(768)
출력층
(768)

실습 소개 (3)

- 입력 이미지
- 중간 이미지
- 복원된 이미지



케라스 (Keras) 소개

- 파이썬으로 작성된 오픈소스 Deep Learning 라이브러리 (구글에서 시작)
- 다양한 프레임웍 위에서 동작하는 상위레벨 딥러닝 프레임웍
  - TensorFlow, MXNet, CNTK, Deeplearning4j, Theano 등 지원
- TensorFlow위에서만 동작하는 라이브러리도 있음
  - from tensorflow import keras vs. import keras
  - TensorF TF의 상위레벨 라이브러리로 공식 확정됨

케라스 API를 사용하는 세 가지 방법

- Sequential 모델 API
  - 가장 간단하며 가장 많이 사용됨 (70+%)
  - 하나의 입력 데이터, 하나의 출력 데이터, 순차 레이어 스택을 지원
- Functional API
  - 레고블록 모델
  - 다중 입력 데이터, 다중 출력 데이터, 임의의 그래프 구조 지원 (TensorFlow와 흡사)
  - Sequential 모델에 비해 복잡
- Model Subclassing
  - 가장 Flexible하지만 가장 복잡
  - 개인적으로는 써 본적이 없음

숫자 이미지 오토인코더 실습

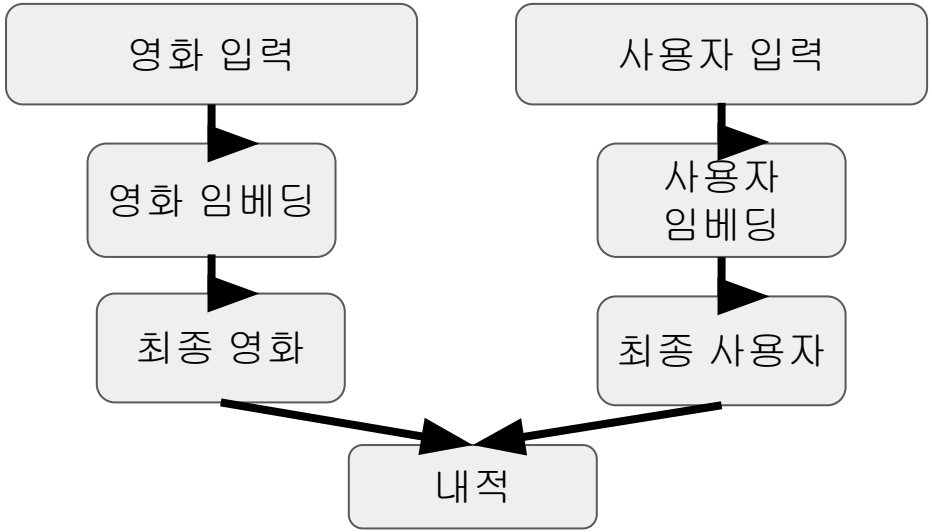
- https://colab.research.google.com/drive/1WmM0LviGcq4t4rYOMjcNQcAk-vJtlOpN?usp=sharing

오토 인코더 기반 추천 엔진 실습

딥러닝 오토인코더를 기반으로 추천 엔진을 만들어 보자

실습 소개

- 무비렌즈 데이터셋의 평점 정보 예측하는 딥러닝 모델 생성
- 무비렌즈 데이터에서 영화의 수는 9,066개
  - 하지만 영화ID의 범위는 1부터 163949가 됨
  - 무비렌즈 영화ID의 값을 순차적으로 변경가능 (0부터 9065)



Input 레이어

Embedding 레이어

Flatten 레이어

Dot 레이어 (평점 예측을 배우게 됨)

무비렌즈 영화 평점 예측 오토인코더 실습

- https://colab.research.google.com/drive/1Ux3elIhFxCRXtG3A0g6OAV_RlVmYZFF-?usp=sharing