

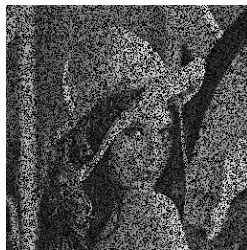
CSE414 딥러닝 과제#2

- 1. 할 일:** Matrix Factorization을 구현하여, 아래 주어진 Image들 각각을 R Matrix라 하고, 각 각의 이미지에 대해 Matrix Factorization을 사용하여 P, Q 두 개의 Matrix로 분해한 다음, 다시 $P \times Q^T$ 를 계산하여 원래 이미지를 복원하여라.

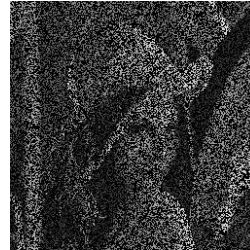
대상 이미지:



Lena_00c.png



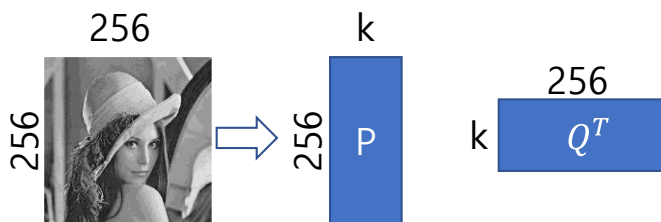
Lena_40c.png



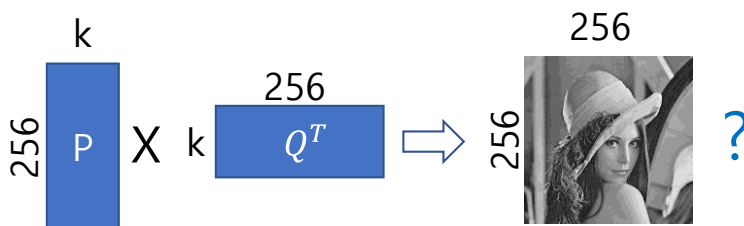
Lena_60c.png

Lena_00c.png가 원본 이미지, Lena_40c.png는 원본 이미지 40%의 pixel을 random하게 골라 값을 0으로 만든 이미지, Lena_60c.png는 원본 이미지 60%의 pixel을 random하게 골라 값을 0으로 만든 이미지이다.

Step 1. Factorization



Step 2. Reconstruction



2. 구현:

- Matrix Factorization의 설명 및 알고리즘은", 그리고 Web에서 검색할 수 있는 문서를 참고한다. (알고리즘은 수업자료 "[Material]딥러닝_07_MLDM_MF.pdf" 혹은 아래 그림 참조)
- 주어진 이미지를 256 by 256의 R Matrix라 두고 이를 Matrix Factorization을 사용하여 각각 256 by k인 P Matrix와 Q Matrix로 분해한다.

Input : Rating Matrix R (m by n matrix), feature vector dimension k , Regularization parameter: λ , learning rate: α , max_iterations: i_{max} , epsilon : ϵ
Output : Matrix P (m by k matrix) and Q (n by k matrix).

```
{
  Randomly initialize the elements of  $P$  and  $Q$ .
  for 1 to  $i_{max}$  :
    for each  $\langle u, i \rangle \in R$ 
       $p_u \leftarrow p_u - \alpha \frac{\partial l_{u,i}}{\partial p_u}$ 
       $q_i \leftarrow q_i - \alpha \frac{\partial l_{u,i}}{\partial q_i}$ 
    endfor
    if  $|l_{cur} - l_{prev}| < \epsilon$  :
      break;
    endif
  endfor

  Return P, Q
}
```

l_{cur} : the value of l that calculated after current iteration.

l_{prev} : the value of l that calculated after previous iteration.

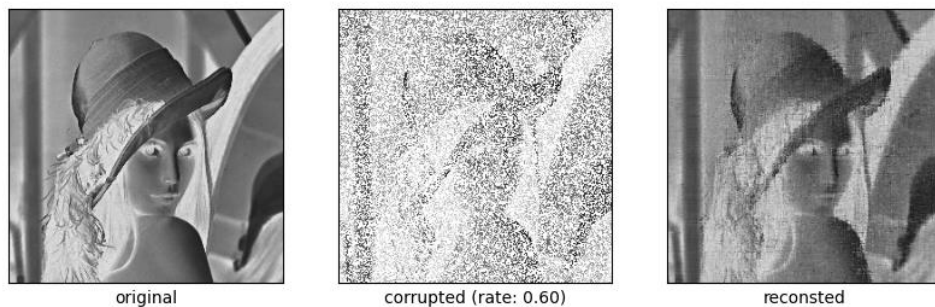
$$l = \sum_{\langle u, i \rangle \in R} l_{u,i} \quad l_{u,i} = (r_{u,i} - p_u \cdot q_i)^2 + \lambda \|p_u\|^2 + \lambda \|q_i\|^2$$

- P, Q matrix의 element값의 초기값은 평균 0.0, 표준편차 1/k의 Gaussian Distribution에서 random sampling하라.
- 추천에서 $\langle u, i \rangle \in R$ 는 Rating matrix에서 실제 유저 u 가 i 에 준 Rating값, 즉 관측된 rating 값을 가리키나 (위치는 R matrix의 u 행, i 열에 해당하는 element 값), 본 과제에서는 R matrix가 이미지이므로, $\langle u, i \rangle \in R$ 는 R matrix의 u 행, i 열의 값이 0 보다 큰 값을 가지는 (0은 랜덤으로 삭제한 픽셀 값)인 element를 가리킨다.
- 추천이 아니므로 λ_u, λ_i 는 같은 값 λ 을 사용한다. 즉 $\lambda_u = \lambda_i = \lambda$ 이다.
- Learning rate α 는 0.1을 사용한다.
- max_iterations i_{max} 는 100을 사용한다.
- epsilon ϵ 는 나름의 작은 값을 정해 보라.

이미지 취급 시 주의점:

- 읽은 이미지의 각 pixel은 0~255의 값을 가지는데, Matrix Factorization하기 전에 255로 나누어 0.0~1.0 사이 값을 가지도록 Normalize한 다음에 Matrix Factorization을 적용하여라.

실행 예)



왼쪽: Original (Lena_00c.png) 이미지,

가운데: Corrupted (Factorization 대상이 되는 Image. 즉 분해하기 위한 Rating matrix R에 해당하는 이미지. 이 예에서는 Lena_60c.png),

오른쪽: Reconsted: R을 분해하여 계산한 P, Q를 사용하여 $P \times Q^T$ 로 계산한 이미지.

즉, 왼쪽 이미지는 원본이 원래 어떤 이미지인지 비교용. 가운데의 오염된 이미지를 Matrix Factorization 후 P, Q를 사용하여 재구성한 오른쪽 이미지가 얼마나 오염되기 전의 왼쪽 이미지에 가까워지는지 비교할 목적으로 표시해 두는 것.

기타 구현 시 유의점:

- Python으로 작성할 것.

동봉된 A02_MF.py의 내용을 채워 넣을 것

그럼에도 불구하고 새로 짜고 싶다면 짜도 됨.

- Matrix Factorization은 상용 Library 사용하지 말고 직접 구현할 것.
- Matrix Factorization 구현은 직접해야 하지만, 코딩 시 웹에서 찾을 수 있는 문서 및 소스 코드 참조 가능. 단, 복사 (ctrl-c+ctrl-v, 눈으로 보고 손으로 쓰기 모두 해당.) 하지 말고 본인이 이해하는데 참고만 하고 프로그램 짜는 것은 스스로 생각해서 할 것.

3. 평가: Matrix Factorization을 구현하여 다음 설정에서 실행한 후 결과를 snapshot으로 제시 하여라.

평가 1. K 에 따른 Reconstruction 성능.

No.	Target	k	λ
1-1	Lena_00c.jpg	16	0.0
1-2	Lena_00c.jpg	32	0.0
1-3	Lena_00c.jpg	64	0.0

평가 2. λ 의 기능.

	Target	K	λ
2-1	Lena_00c.jpg	64	0.0
2-2	Lena_00c.jpg	64	0.001
2-3	Lena_00c.jpg	64	0.01

평가 3. Reconstruction Performance

No.	Target	K	λ
3-1	Lena_40c.jpg	64	0.0
3-2	Lena_40c.jpg	64	0.001
3-3	Lena_40c.jpg	64	0.01
3-4	Lena_60c.jpg	64	0.0
3-5	Lena_60c.jpg	64	0.001
3-6	Lena_60c.jpg	64	0.01

Lena_80c.jpg, Lena_90c.jpg는 평가에 포함하지는 않으나 흥미 있으면 테스트 가능.

4. 기한: 2023-06-07 (수) 23:59 분까지

5. 제출물: 아래 (1), (2)을 "한 deeplearn_hw2_학번.zip" 으로 압축하여 제출

(1) Source code 파일

A02_image_reconst_main.py, A02_MF.py

(2) 리포트 문서 한글 or 워드

- 자신이 짠 소스 코드를 왜 그렇게 짰는지 이유를 설명할 것
- "3. 평가"에서 요구하는 Matrix Factorization을 실행한 결과에 대한 스크린 샷을 모두 첨부할 것.
- 다음에 대해 고찰할 것.
 - Feature vector dimension k 가 클 경우와 작을 경우의 각각의 장단점을 고찰하라.
 - Regularization weight λ 가 클 경우와 작을 경우 어떤 효과가 있는지 고찰하라.
- Matrix Factorization을 작성해서 테스트해 보면서 느낀 점이나 의문점등을 자유롭게 기술할 것.

6 제출: LMS로 제출

LMS 과제2에 파일 업로드