

## Image denoising with a denosing autoencoder

2020-08-18 노정현

기간 : 2020-08-11 ~ 2020-08-17

### ● autoencoding

- autoencoding은 데이터 압축 알고리즘으로 압축과 압축 해제가 자동으로 학습되는 알고리즘

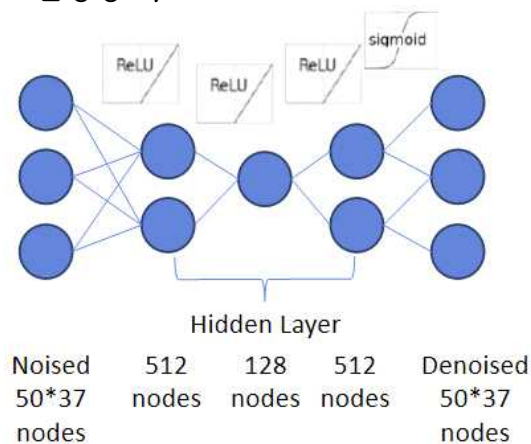
- 학습에 사용된 유형의 데이터에 대해서만 데이터 압축이 가능
- 손실 압축
- 자동으로 학습이 되어 올바른 학습 데이터만 사용하면 됨

### ● Denosing autoencoder

- 신경망을 사용한 비지도학습으로 노이즈가 발생된 이미지를 원본 영상으로 복구하는 autoencoding의 애플리케이션 중 하나

- 입력 dataset은 노이즈된 이미지
- 라벨 dataset은 노이즈가 없는 원본 이미지

### ● 신경망 구조



- ReLU : 0보다 작은 값은 0을, 0보다 큰 값은 그대로 반환
- Sigmoid : 함수값이 (0, 1), 중간값이 1/2인 활성화 함수
- 입력 이미지의 크기 : (50,37)
- Optimizer : Adam(learning rate=0.001, weight\_decay=0.00001)  
학습률 : 경사하강법 알고리즘에서 다음 지점을 결정하기 위한 값  
가중치 감쇠 : 오버피팅을 방지하기 위한 학습률 조절
- Loss : Binary Cross Entropy(선택지가 두 개인 손실함수)

- **pytorch -> keras**

Sequential 모델 : input과 output이 한 개씩만 존재(단일 출력 모델)

계층을 선형적으로 쌓음

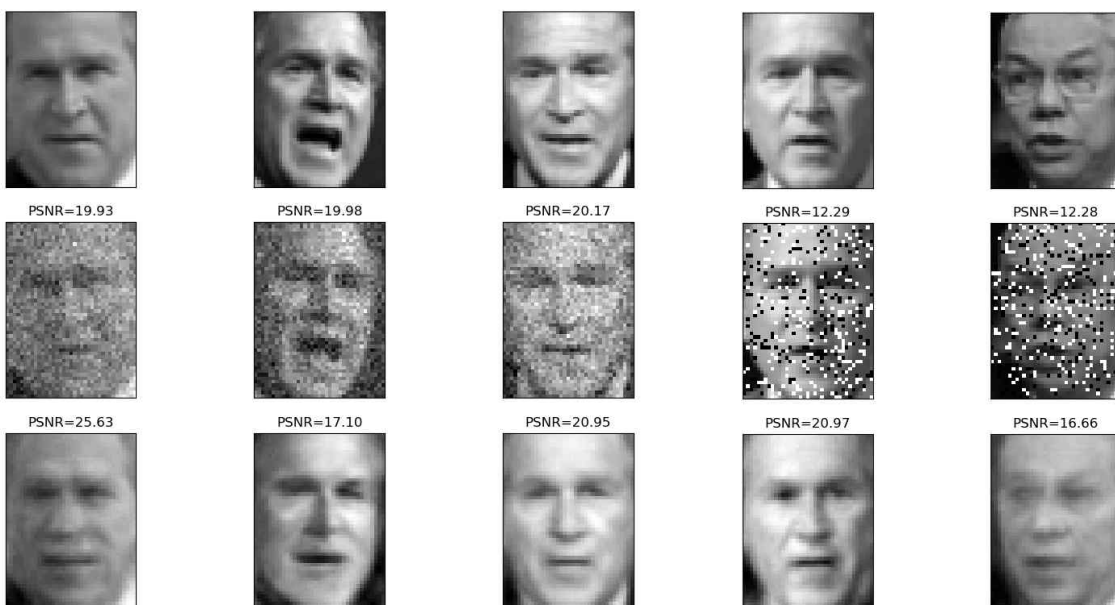
함수형 API : 다중 출력 모델

방향성 비순환 그래프

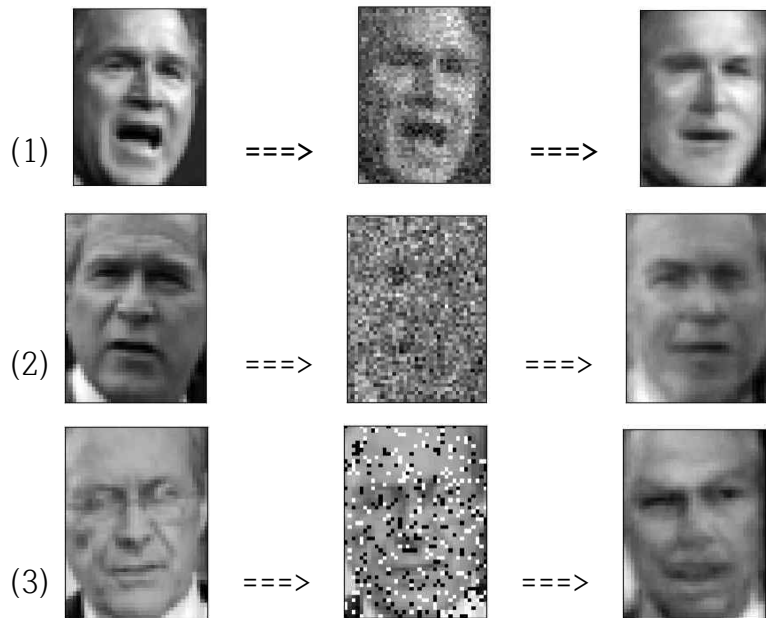
공유 계층

- `keras.layers.Input(shape=(n,))` : (n, 1)의 입력 계층 정의
- `keras.layers.Dense(512, activation='relu')(input)` : 입력을 input, 노드 수가 512, 활성화 함수가 relu인 계층 정의
- `keras.models.Model(input, output)` : 입력이 input, 출력이 output인 모델 정의
- `optimizer = keras.optimizers.adam(lr=learning_rate, decay=1e-5)`  
학습률이 learning\_rate이고, 가중치 감쇠가 1e-5인 최적화 함수 정의
- `autoencoder.compile(optimizer=optimizer, loss='binary_crossentropy')`  
최적화 함수가 optimizer이고,  
손실 함수가 binary\_crossentropy인 모델 구성
- `autoencoder.fit(noised, flat, batch_size=batch_size, epochs=num_epochs, shuffle=True)` : 훈련 데이터와 목표 데이터에 대해 batch\_size, epochs만큼의 학습을 무작위로 실시한다.
- Input Image는 1차원으로 flatten()

- **결과(batch\_size = 16, epochs = 100)**



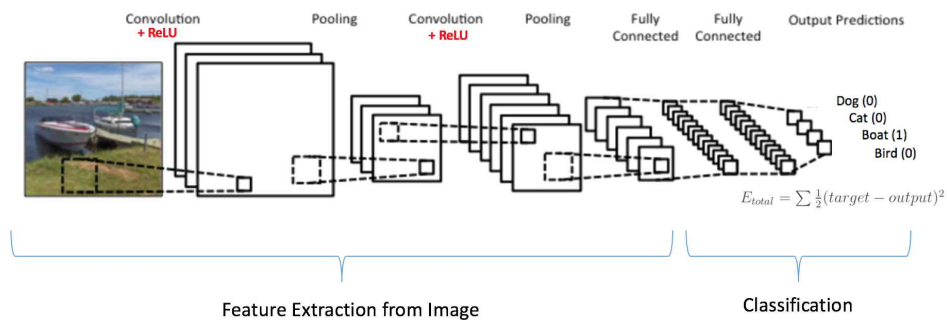
## ● 특징



- (1) : Denosing은 이루어졌지만, 입 모양에서의 변화가 생김
- (2) : noise가 심한 영상에는 전혀 다른 얼굴이 검출됨
- (3) : salt&pepper 노이즈 영상에 적용 시 오차가 큼

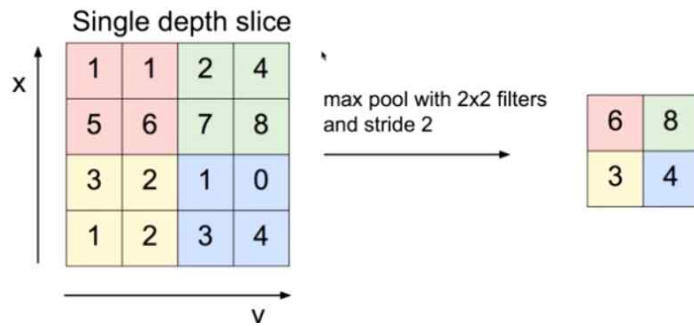
## ● Convolution autoencoder (CNN)

- 이미지에 일정 크기의 필터를 convolution 연산을 하여 필터링을 하면서 신경망을 이용한 학습 (예제 : convolution-maxpooling-upsampling 사용)
- 이미지의 특징을 자동으로 검출
- 입출력 데이터의 형상유지 => flatten 불필요



Max pooling :

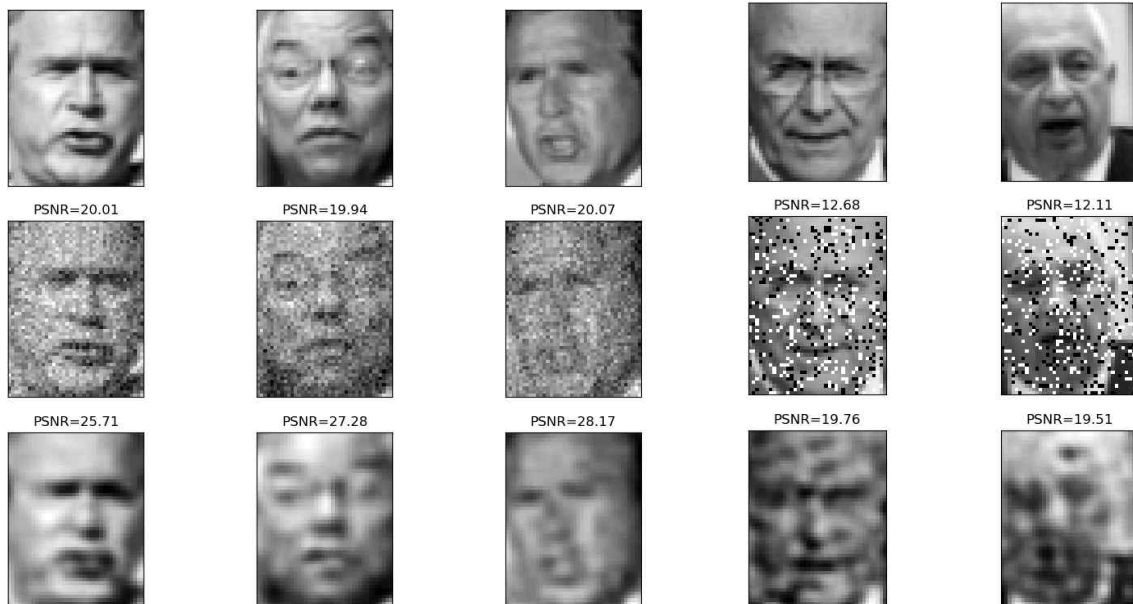
정해 범위의 픽셀중 큰 값만  
출력 => 전체 크기가 작아짐  
오버피팅 방지



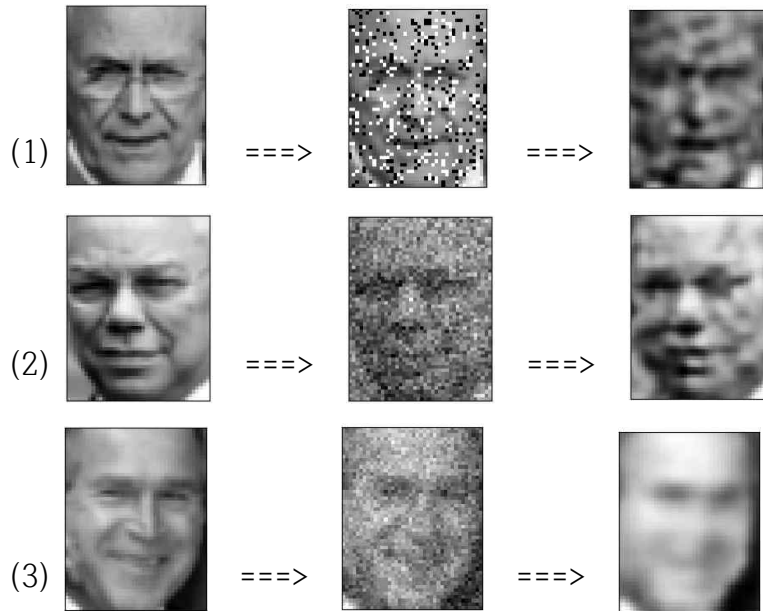
Upsampling : Transposed convolution 사용, 가중치는 학습가능  
저해상의 이미지를 고해상의 이미지로 변환하는 작업

- `keras.layers.Conv2D(32, (3, 3), activation='relu', padding='same')`  
output : 32, 커널 : (3, 3), 활성화함수 : relu, padding : same
- `keras.layers.MaxPooling2D((2, 2), padding='same')`  
pool size : (2, 2), padding : same
- `keras.layers.UpSampling2D((2, 2))`  
(2, 2)로 upsampling

● 결과(batch\_size=128, epochs=100)



● 특징



(1) Gaussian 노이즈로 학습한 모델에 salt&pepper 노이즈가 적용된 이미지에 denoising이 제대로 적용되지 않는다.

(2) 노이즈가 심한 영상에 적용하면 왜곡이 되지만 원형은 어느 정도 유지

(3) 노이즈가 심한 영상으로 학습 후 노이즈가 적은 영상에 적용 시 흐릿해짐