1. 사용 환경

2. 데이터 로드

2. 데이터 셋 설명

사용 환경

Python Version	3.8.3
Tensor Flow Version	2.3.0

데이터 불러오기

MNIST Data Set을 사용한다.

Auto-Encoder (AE) 기반 차원 축소

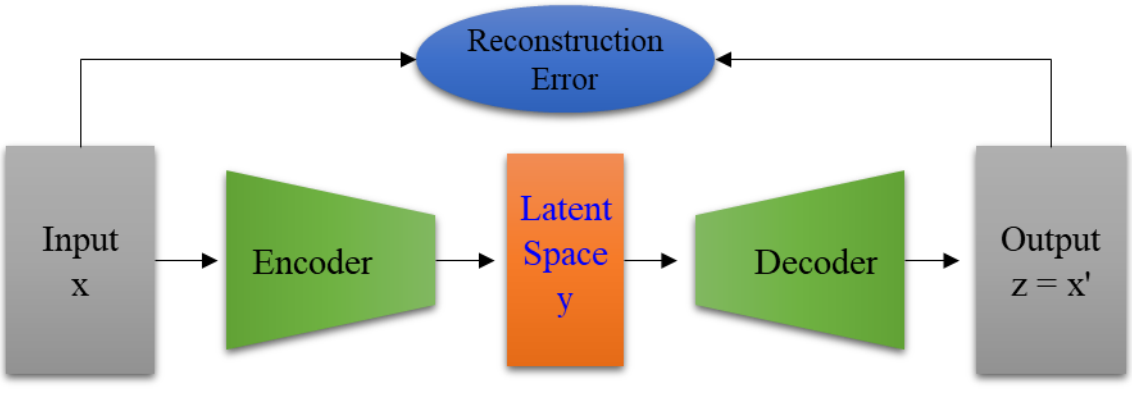
◆ 개요

- 차원 축소를 위한 AE의 구성
- TensorFlow를 통한 코드 구현
- AE 성능 확인
- 느낀 점

◆ 차원 축소를 위한 AE의 구성

- Unsupervised Learning 모델
- Input Data와 Out Data 구조가 동일
- Reconstruction Loss가 최소화 되도록 모델 학습 진행
- Input Data의 특징이 압축된 Latent Space가 핵심
- Latent Space라는 개념을 통해 차원 축소 가능

AE는 **고차원의 정보를 압축해 주는 Encoder**와 **압축된 정보를 다시 원래 정보로 돌려주는 Decoder**로 이루어져 있다.



【 Auto-Encoder (AE) 구조 】

AE 모델은 Encoder-Decoder의 결합으로 구성된다.

응용 방안으로는 학습이 완료된 Encoder-Decoder 구조에서 Decoder만 따로 사용이 가능하다. 압축된 정보를 Decoder의 입력으로 사용하면 알아서 원본 이미지와 유사한 가짜 이미지를 만드는 것이 가능한다.

- Encoder

$$y = f_\theta(x) = s(Wx + b)$$

- Decoder

$$z = g_{\theta'}(y) = s(W'y + b')$$

- 손실 함수

$$\theta^*, \theta'^* = \arg min \frac{1}{N} \sum_{i=1}^{n} L\left(x^{(i)}, z^{9i)}\right)$$

기본적인 Auto-Encoder 의 Weight를 학습하는 방법은 Backpropagation (BP) 알고리즘과 Stochastic Gradient Descent (SGD) 알고리즘을 활용하였다. Input으로 주어진 값에 대하여, hidden과 input 간의 weight 값을 계산하여, sigmoid 함수에 넣고, 다시 그 값을 한번 더 hidden과 output에 연결되어 있는 weight와 계산하여 추정된 값을, 최초 input data와 지속적을 비교하여 Reconstruction Error에 대해 update를 진행하는 Unsupervised Learning이다.

◆ 논문

Auto-encoderBasedDimensionalityReduction(2016, Neurocomputing)

AE가 쌓일 대 축적되어 딥러닝의 성공에 기여할 수 있는 좋은 속성 있는지 여부에 대한 아이디어에서 나왔다.해당 논문은 Buliding Block인 AE의 차원 축소 능력에 중점을 두고 있다. AE의 원래 입력 노드 수보다 적은 수의 히든 레이어 노드를 제한하면 원하는 차원 축소 효과를 얻을 수 있따.

MNIST와 Olivetti 얼굴 데이터 세트로 평가를 진행하였다.

차원 축소 과정에서 일부 차원을 버리면 필연적으로 정보 손실이 발생한다. 따라서 해결해야 할 주요 문제는 원본 데이터의 주요 및 중요한 특성을 가능한 한 많이 유지하는 것이다.
따라서 차우너 축소 과정은 원래 데이터 특성과 밀접한 관련 이 있다고 말한다.
고차원 데이터의 중요한 고유 특성인 latent space는 데이터의 필수 차원을 잘 반영 할 수 있다. 고차원 공간의 샘플 데이터는 일반적으로 전체 공간에서 확산 될 수 없다. 그것들은 실제 고차원 공간에 내장 된 저 차원 다양체에 놓여 있으며 다양체의 차원은 데이터의 고유한 차원이다.

대표적인 차원 축소 방법인 PCA, LDA, LLR, Isomap에 대한 분류 성능 비교 수행하였다.
선형 방법 : PCA, LDA
비 선형 방법 : LLE, Isomap

MNIST 데이터 셋은 60,000개의 학습 이미지와 10,000개의 테스트 이미지가 포함된 필기 숫자의 고전적인 데이터 세트이다. 모든 이미지는 표준화되고 통합된 크기로 된다. 이미지 크기는 28x28이가. 데이터 셋은 회식조 이미지다.

히든 레이어 수의 영향

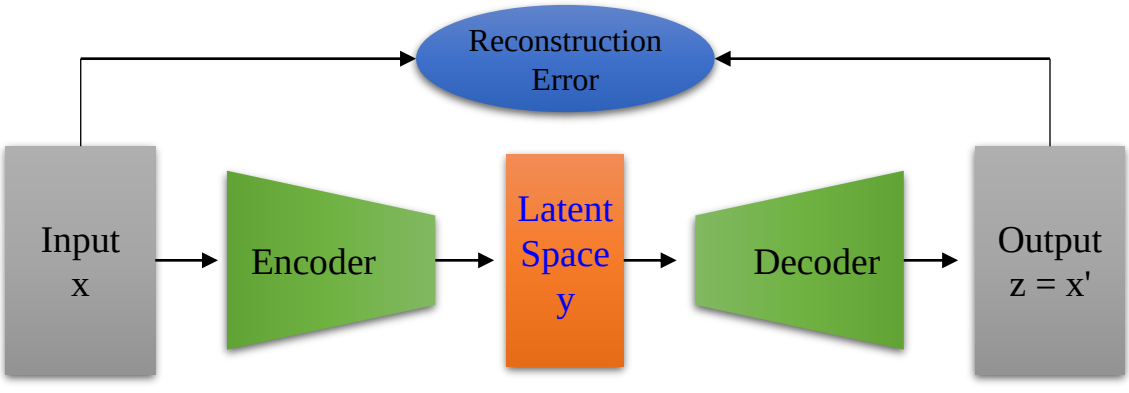
MNIST data set을 통한 차원 축소에서
- 차원 축소를 위한 AE의 구성
- TensorFlow를 통한 코드 구현
- AE 성능 확인
- 느낀 점

◆ 코드

Auto-encoderBasedDimensionalityReduction
- 차원 축소를 위한 AE의 구성
- TensorFlow를 통한 코드 구현
- AE 성능 확인
- 느낀 점

◆ 실험

Auto-encoderBasedDimensionalityReduction
- 차원 축소를 위한 AE의 구성
- TensorFlow를 통한 코드 구현
- AE 성능 확인
- 느낀 점



【 Auto-Encoder (AE) 구조 】