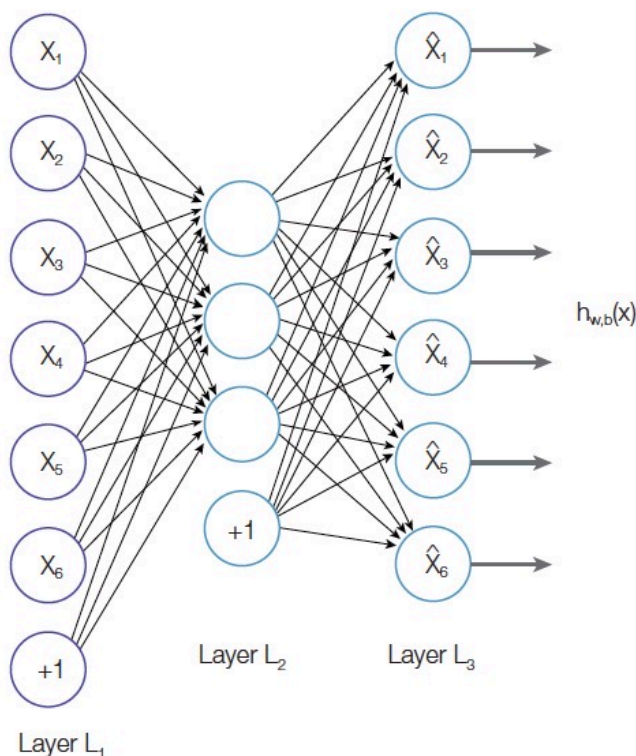

강의 목표

1. 오토인코더(AutoEncoder)의 개념을 이해합니다.
2. TensorFlow 2.0을 이용해서 MNIST 데이터 재구축을 위한 오토인코더(AutoEncoder)를 구현해봅시다.

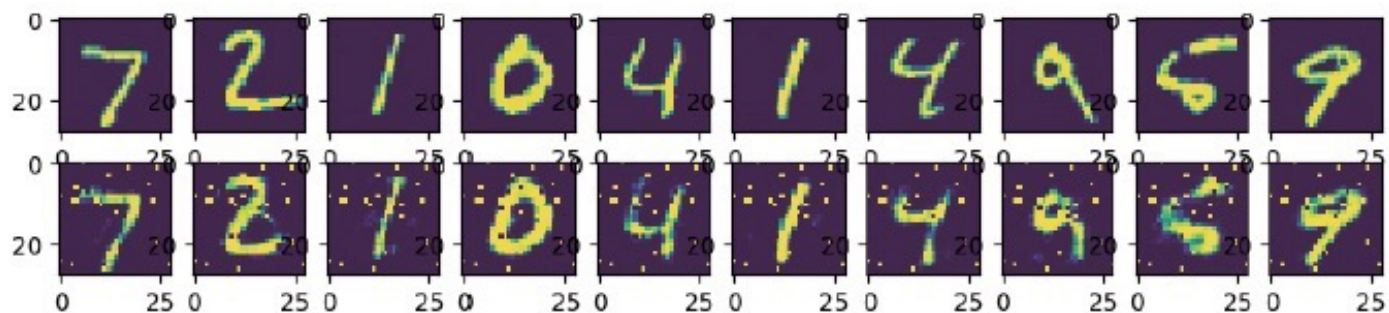
오토인코더 Autoencoder의 개념

- 오토인코더 Autoencoder는 대표적인 비지도 학습 Unsupervised Learning을 위한 인공신경망 구조 중 하나입니다. 비지도 학습은 1강에서 살펴본 것과 같이 어떤 값을 예측하거나 분류하는 것이 목적인 지도 학습 Supervised Learning과는 다르게 데이터의 숨겨진 구조를 발견하는 것이 목표인 학습 방법입니다.
- 구체적으로 오토인코더는 출력층 Output Layer의 노드 개수와 입력층 Input Layer의 노드 개수가 동일한 구조의 인공신경망입니다. 아래 그림은 1개의 은닉층 Hidden Layer을 가진 오토인코더 구조를 나타냅니다.



오토인코더 Autoencoder의 개념

- 따라서, 오토인코더의 출력은 **원본 데이터 Raw Data**를 **재구축 Reconstruction**한 결과가 됩니다. 아래 그림은 오토인코더를 이용해서 MNIST 데이터를 재구축한 결과를 보여줍니다. 재구축 결과가 완벽하지 않아서 약간의 노이즈가 포함된 모습을 볼 수 있습니다.



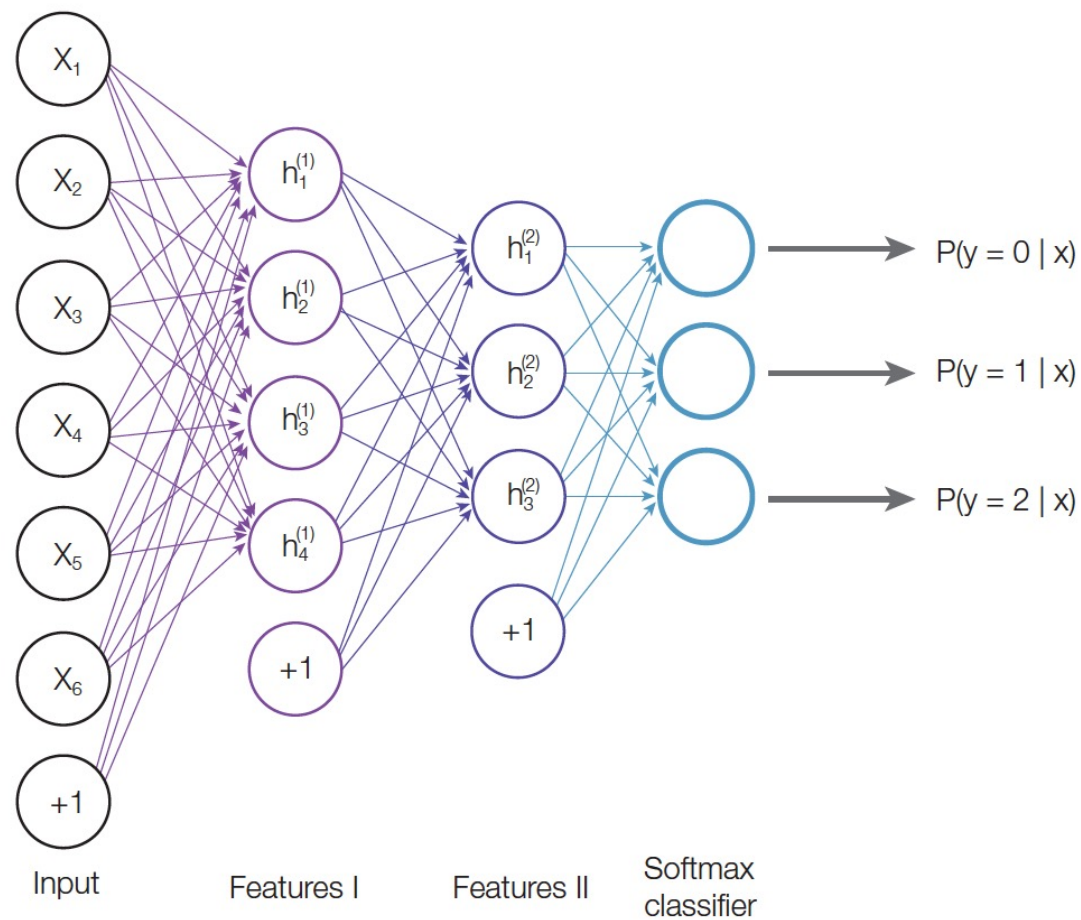
- 그렇다면 이런 데이터 재구축을 어떤 용도로 활용할 수 있을까요? 사실 오토인코더의 핵심은 재구축된 출력층의 출력값이 아니라, **은닉층의 출력값**입니다. 오토인코더의 구조를 자세히 보시면 은닉층의 노드 개수가 입력층과 출력층의 노드 개수보다 적다는 사실을 관찰할 수 있습니다.
- 따라서, 은닉층은 더 작은 표현력으로 원본 데이터의 모든 특징들 Features를 학습해야 합니다. 결과적으로 은닉층의 출력값은 **원본 데이터에서 불필요한 특징들을 제거한 압축된 특징 Compressed Features**들을 학습하게 됩니다.

오토인코더 Autoencoder의 장점

- 예를 들어, 우리가 어떤 데이터로부터 이 데이터가 나타내는 동물이 개인지 고양이인지를 예측해야만 하는 상황을 가정해봅시다. 이때 데이터로부터 주어진 동물의 특징은 다음과 같이 4가지라고 가정해봅시다.
- 데이터의 구성 : ‘동물의 길이, 동물의 몸무게, 동물 주인의 나이, 동물 주인의 몸무게’
- 이 데이터로부터 이 동물이 개인지 고양이인지를 분류하고자 하는 경우를 생각해보면, ‘동물 주인의 나이, 동물 주인의 몸무게’의 2개의 특징은 이 동물이 개인지 고양이인지를 분류하는데 도움이 되지 않는 **잉여 특징 Redundant Feature**임을 알 수 있습니다. 따라서 ‘동물의 길이, 동물의 몸무게’의 2개의 특징만을 이용해서 분류를 수행하는 것이 합리적 일 것입니다.
- 다소 거칠게 표현하면, 오토인코더 은닉층의 출력값은 학습을 통해서 이 과정을 자동적으로 수행하고 남은 특징들입니다. 즉, 오토인코더 은닉층의 출력값은 위의 가정에서 ‘동물 주인의 나이, 동물 주인의 몸무게’라는 불필요한 특징들을 자동적으로 제거하고 남은 ‘동물의 길이, 동물의 몸무게’ 2개의 특징만을 가지고 있을 것입니다.
- 이렇게 압축된 특징을 나타내는 은닉층의 출력값을 원본 데이터 대신에 분류기 Classifier의 입력으로 사용하면 더욱 좋은 분류 성능을 기대할 수 있을 것입니다.

오토인코더 Autoencoder의 장점

- 다음 그림은 원본 데이터 대신에 오토인코더의 2번째 은닉층의 출력값을 분류기의 입력으로 사용하는 예시를 나타냅니다.



TensorFlow 2.0을 이용한 오토인코더를 이용한 MNIST 데이터 재구축

- 오토인코더를 이용해서 MNIST 데이터를 재구축하는 예제를 살펴봅시다.
- https://github.com/solaris33/deep-learning-tensorflow-book-code/blob/master/Ch06-AutoEncoder/mnist_reconstruction_using_autoencoder_v2_keras.py

Chapter 4 - 머신러닝 기초 이론들

- 소프트맥스 회귀(Softmax Regression)를 이용한 MNIST 숫자분류기 ([Code](#)) ([TF v2 Code](#)) ([TF v2 Keras Code](#))
- tf.nn.sparse_softmax_cross_entropy_with_logits API를 사용한 소프트맥스 회귀(Softmax Regression)를 이용한 MNIST 숫자분류기 ([Code](#)) ([TF v2 Code](#))

Chapter 5 - 인공신경망(Artificial Neural Networks) - ANN

- ANN을 이용한 MNIST 숫자분류기 구현 ([Code](#)) ([TF v2 Code](#)) ([TF v2 Keras Code](#))

Chapter 6 - 오토인코더(Autoencoder)

- 오토인코더를 이용한 MNIST 데이터 재구축 ([Code](#)) ([TF v2 Code](#)) ([TF v2 Keras Code](#))
- 오토인코더 + 소프트맥스 회귀를 이용한 MNIST 숫자 분류기 ([Code](#)) ([TF v2 Code](#)) ([TF v2 Keras Code](#))

Thank you!
