AutoEncoder

https://youtu.be/EjK2QAmUxNI

Autoencoder

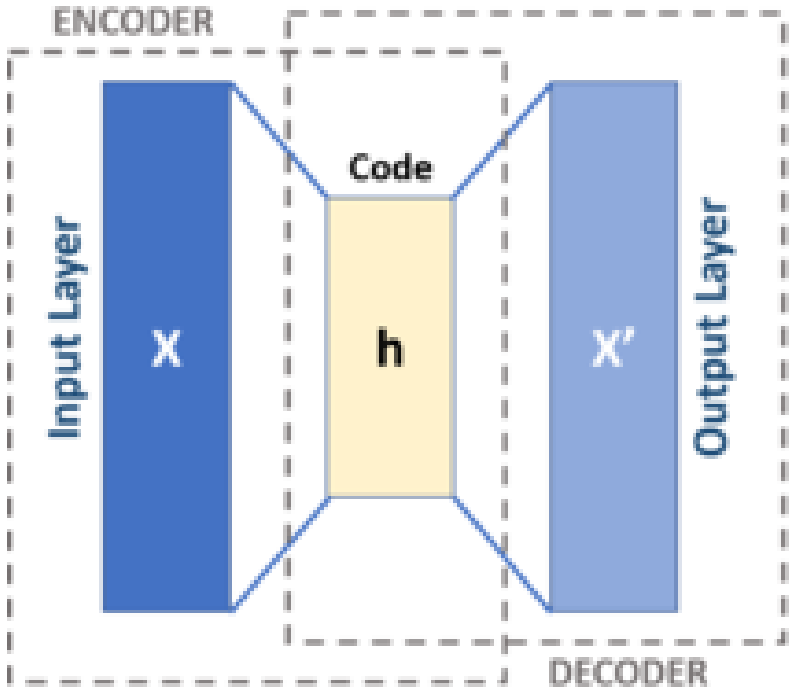
DNN-AutoEncoder

CNN-AutoEncoder

계획

AutoEncoder

- Keywords
  - Unsupervised Learning
  - Representation Learning
  - Dimensionality Reduction
  - Generative Model Learning

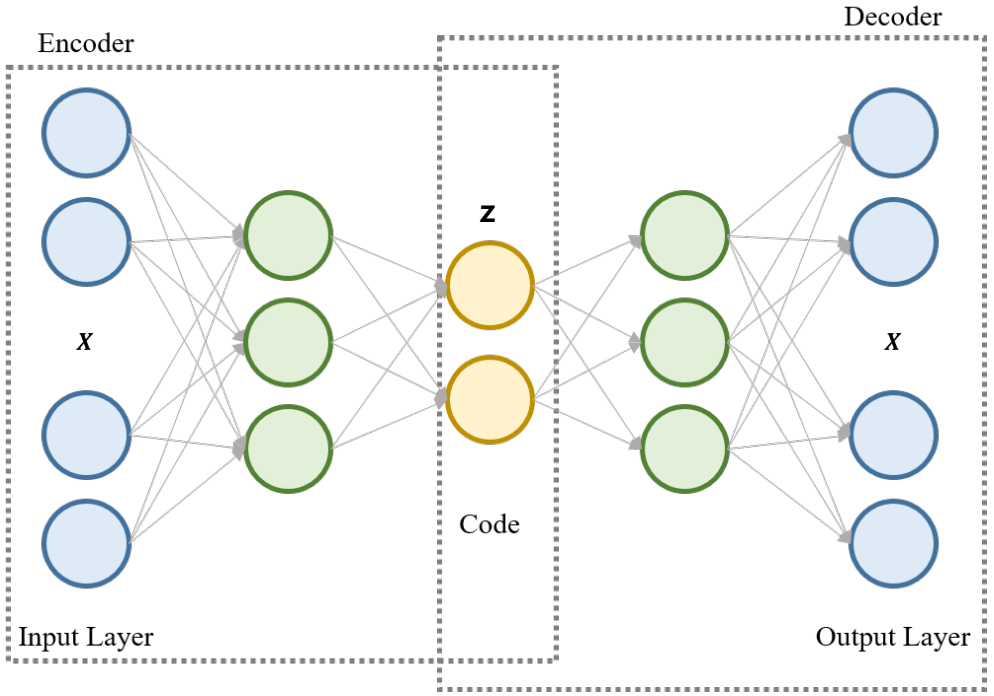


AutoEncoder

- 특징
  - 입력과 출력이 같다.
  - Latent Variable(z)이 존재한다.

- 과정
  - Input Data를 Encoder Network에 통과시켜 압축된 z값을 얻는다.
  - 압축된 z vector로부터 input data와 같은 크기의 출력값을 생성한다.
  - 이때, 입력값 x와 Decoder를 통과한 y를 통하여 Loss를 계산한다.

- 목적
  - Encoder는 Input Data의 특징을 잘 추출해내는 것
  - Decoder는 추출된 z vector로부터 최대한 Input Data와 유사한 출력값을 생성해내는 것



AutoEncoder

- 전처리
  - 음성 -> Numpy array로 변환.
  - 데이터를 0~10 사이의 값으로 Normalization
  - Data Slicing을 하여 사용.
  - Data loader

```python
class PandasDataset(data.Dataset):
    def __init__(self, dataframe):

        self.dataframe = dataframe

    def __len__(self):
        return len(self.dataframe)

    def __getitem__(self, index):
        x = self.dataframe[index]
        y = self.dataframe[index]
        return x, y
```

```python
# Definite Train & Evaluate
def train(model, train_loader, optimizer):
    model.train()
    train_loss = 0

    for step, (x, y) in enumerate(train_loader):
        x = x.to(args.device)
        y = y.to(args.device)

        optimizer.zero_grad()

        encoded, decoded = model(x)

        loss = criterion(decoded, y)
#        print("x:₩n", x)
#        print("encoded:₩n", (encoded))
#        print("y:₩n", y)
#        print("decoded:₩n", (decoded))
        loss.backward()
        optimizer.step()


        train_loss += loss.item()

    train_loss = train_loss / len(train_loader)

    return model, train_loss
```

```
(tensor([4.5235, 4.5269, 4.5189,  ..., 5.7309, 5.5064, 5.2279]), tensor([4.5235, 4.5269, 4.5189,  ..., 5.7309, 5.5064, 5.2279]))
```

DNN-AutoEncoder

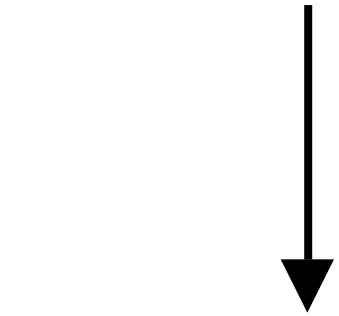
- Model Architecture

```
Model:  AE(
  (encoder): Sequential(
    (0): Linear(in_features=20000, out_features=4096, bias=True)
    (1): ReLU()
    (2): Linear(in_features=4096, out_features=1024, bias=True)
    (3): ReLU()
    (4): Linear(in_features=1024, out_features=128, bias=True)
    (5): ReLU()
  )
  (decoder): Sequential(
    (0): Linear(in_features=128, out_features=1024, bias=True)
    (1): ReLU()
    (2): Linear(in_features=1024, out_features=4096, bias=True)
    (3): ReLU()
    (4): Linear(in_features=4096, out_features=20000, bias=True)
    (5): ReLU()
  )
)
Device:  cuda
```

DNN-AutoEncoder

- Result

```
|| TRAINING - Epoch 1 || Loss[train]: 7.71295 ||
```



```
|| TRAINING - Epoch 100 || Loss[train]: 1.89651 ||
```

CNN-AutoEncoder

- Model Architecture

```
Model:  AE(
  (encoder): Sequential(
    (0): Conv1d(256, 256, kernel_size=(3,), stride=(1,), padding=(1,))
    (1): ReLU()
    (2): Conv1d(256, 256, kernel_size=(3,), stride=(1,), padding=(1,))
    (3): ReLU()
  )
  (decoder): Sequential(
    (0): Conv1d(256, 256, kernel_size=(3,), stride=(1,), padding=(1,))
    (1): ReLU()
    (2): Conv1d(256, 256, kernel_size=(3,), stride=(1,), padding=(1,))
    (3): ReLU()
  )
)
```

CNN-AutoEncoder

- Result

```
|| TRAINING/VALIDATION - Epoch 1 || Loss[train/val]: 1.68690/1.43272 ||
```

```
|| TRAINING/VALIDATION - Epoch 100 || Loss[train/val]: 0.72578/0.74998 ||
```

계획

- Denoising AutoEncoder

- 유저 분류 + 딥보이스 탐지 통합 데이터셋

Q & A