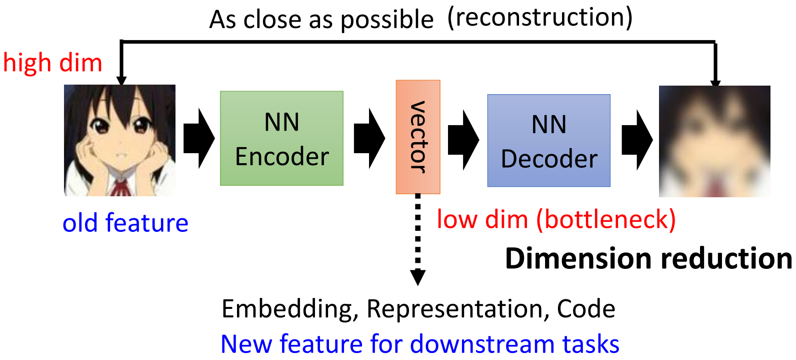
Auto-Encoder - unsupervised learning

Auto encoder is a subfield of self-supervised learning.
Sounds familiar? We have seen the same idea in Cycle GAN.

Use Image as an example

Encoder: (with high dimensional input)
Convert an image to vector and send it to decoder
Export of Encoder is called **Embedding.**
Function of Encoder is to convert high dimensional data to low dimensional data. - **Dimension reduction.**
Decoder: (with low dimensional input)
Export an image


As close as possible (reconstruction)
high dim
NN Encoder
vector
NN Decoder
old feature
low dim (bottleneck)
Dimension reduction
Embedding, Representation, Code
New feature for downstream tasks

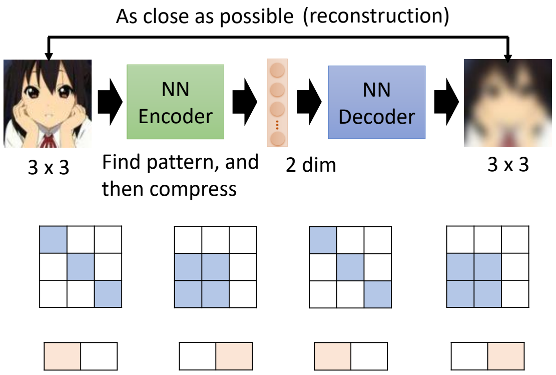
Training:
Let output approach input. (Reconstruction)

The concept is the same as Cycle-GAN

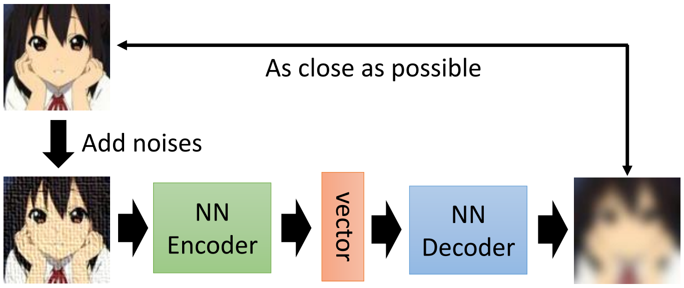
Why Auto-encoder?

The diversity of images are limited. Maybe we can use lower dimension to describe high dimensional data.

$$high\ dim \rightarrow Encoder \rightarrow low\ dim$$


As close as possible (reconstruction)
NN Encoder
NN Decoder
3 x 3
Find pattern, and then compress
2 dim
3 x 3

De-noising Auto-encoder



Vincent, Pascal, et al. "Extracting and composing robust features with denoising autoencoders." *ICML,* 2008.

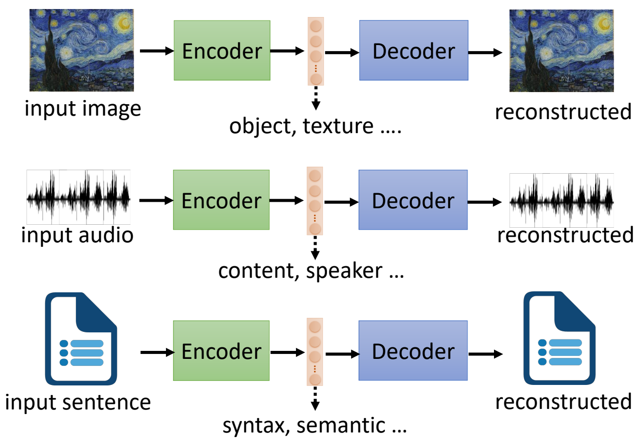
BERT can be seen as a De-noising Auto-encoder.

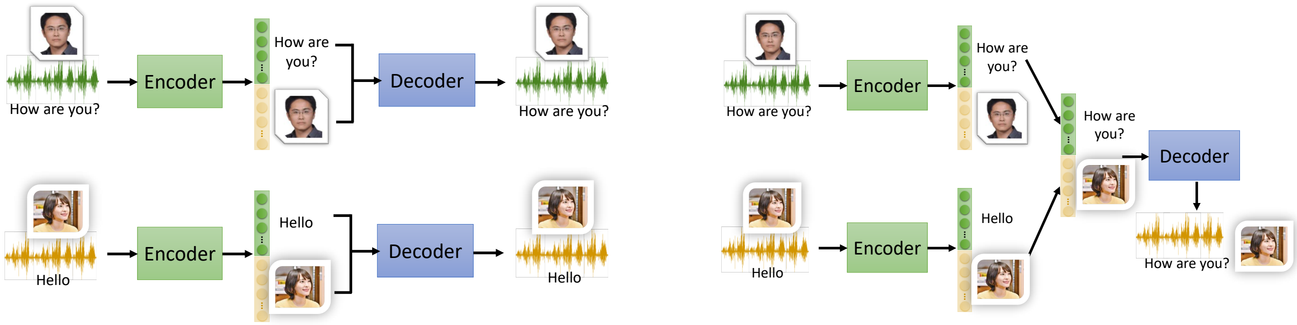
Feature Disentanglement

Auto-encoder can extract features from input data. But we do not know the meaning of those features.

Therefore we want to know the dimensional meanings of features.

Application:
Voice Conversion

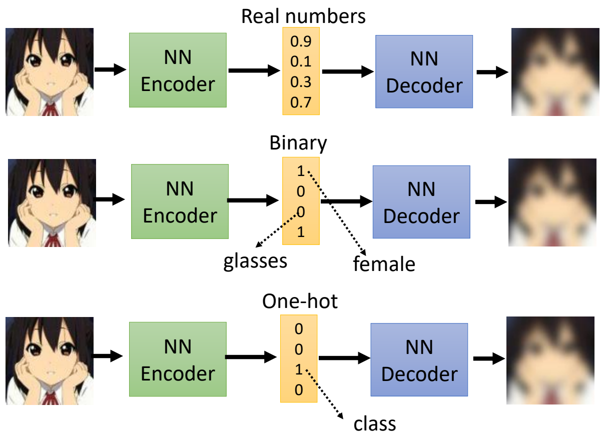




Discrete (Latent) Representation

Until now we assume the output of encoder is a vector (Real numbers), but it also can be binary or one-hot.

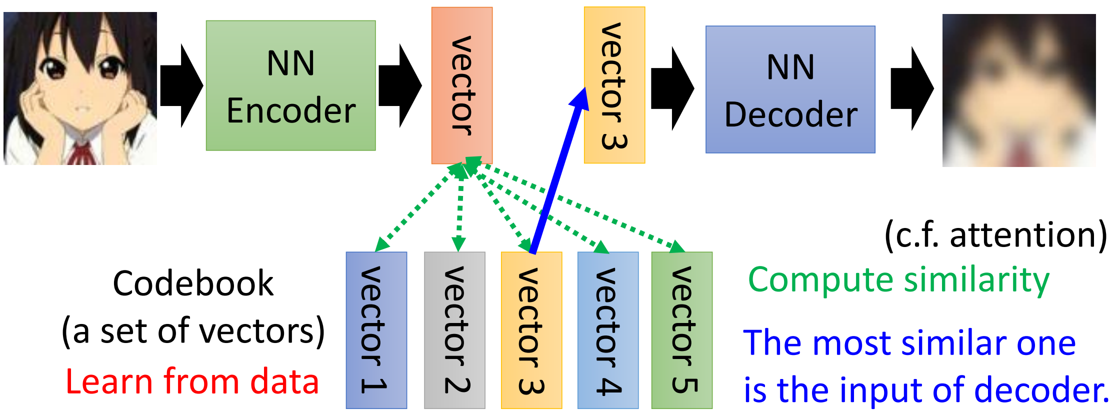
Which can make unsupervised classification- **it might be possible to train a classifier without labeled data.**



$$\text{Discrete Representation} \begin{cases} \text{Real numbers, e.g.}[0.9,0.1,0.3,0.7] \\ \text{Binary, e.g.}[1,0,0,1] \rightarrow [female, a, b, glasses] \\ \text{One-hot, e.g.}[0,0,1,0], class \end{cases}$$

Vector Quantized Variational Auto-encoder (VQVAE)

https://arxiv.org/abs/1711.00937



For speech, the codebook represents phonetic information

Codebook: a set of vectors by learning from data (can be seen as Query in self-attention)
Compared with vector (can be seen as Key in self-attention)
Now we compute similarity of vector and codebook (c.f. attention)
The most similar one is the input of decoder.

Training: $input \sim output$
If there are only 32 vectors in codebook, the input of decoder has only 32 possible inputs, which **makes the input discrete not continuous.**

When we apply VQVAE on speech, the machine will learn the basic components, phonetics.

Other Representation types

Text - seq2seq2seq auto-encoder

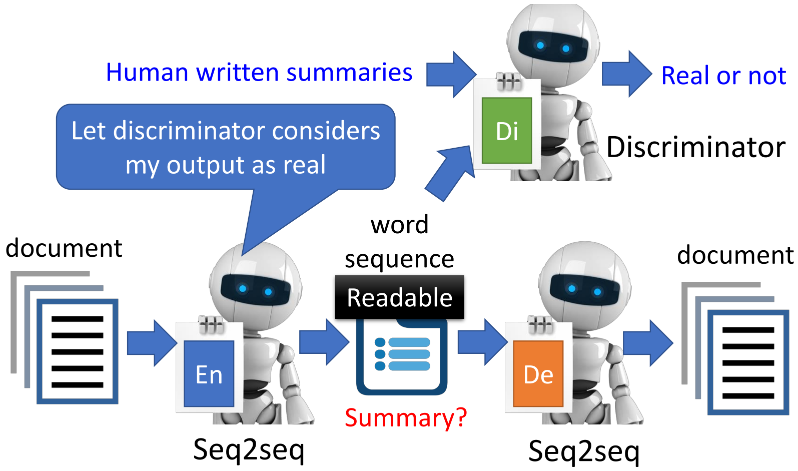
Application: Generate a summary of an article.

$$Document \xrightarrow[seq2seq]{} \underset{summary}{word\ sequence} \xrightarrow[seq2seq]{} document$$

If we only have this one, the word sequence will become a jargon that only could be understood by discriminator. Thus we need to add a discriminator to judge the word sequence is written by human.
It is a cycle-GAN!

Tree structure as embedding.

https://arxiv.org/abs/1806.07832



Generator

Use Decoder as a generator.

We can randomly generate a vector from a distribution.

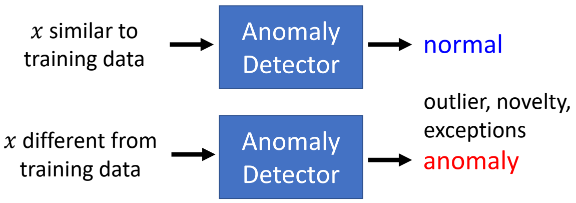
variational auto-encoder (VAE)

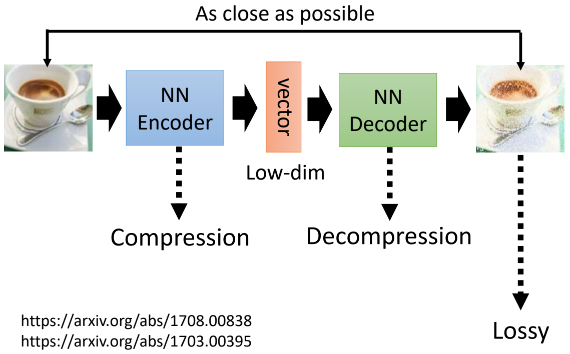
Use decoder as a generator

Compression Images - Lossy

Anomaly Detection

Given a set of training data
$\{x^1, x^2, \ldots, x^N\}$
Now we can detecting input x is similar to training data or not.


x similar to training data
Anomaly Detector
normal
x different from training data
Anomaly Detector
outlier, novelty, exceptions
anomaly



https://arxiv.org/abs/1708.00838
https://arxiv.org/abs/1703.00395

Applications of Anomaly Detection

Binary Classification?

We only have one class. Training auto-encoder

Fraud Detection

Training data: credit card transactions, x: fraud or not • Ref:https://www.kaggle.com/ntnu-testimon/paysim1/home
Ref:https://www.kaggle.com/mlg-ulb/creditcardfraud/home

Network Intrusion Detection

Training data: connection, x: attack or not
Ref: http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html

Cancer Detection

Training data: normal cells, x: cancer or not?
Ref: https://www.kaggle.com/uciml/breast-cancer-wisconsin-data/home

The difficulty is data collection. Because it's difficult to collect abnormal data. We usually make assumption that most of data are normal.
This is called **One class** problem.

Approach: Auto-encoder

Training:
Using real human faces to learn an *auto-encoder*

Testing:
If input can be reconstructed, it is true.
If input can not be reconstructed, it is abnormal.
We can check reconstruction loss.
Large reconstruction loss → abnormal.

There are more about anomaly detection…