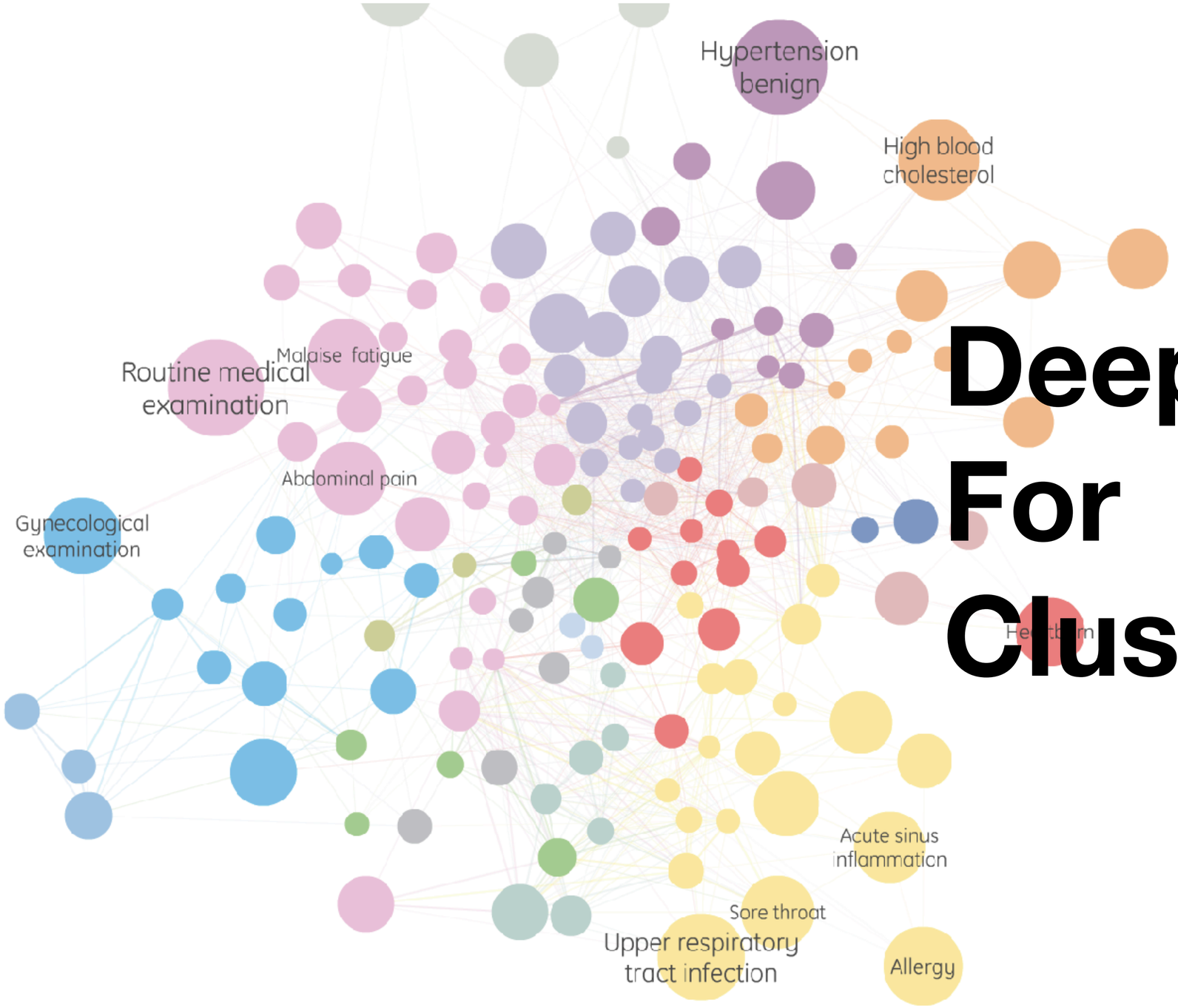
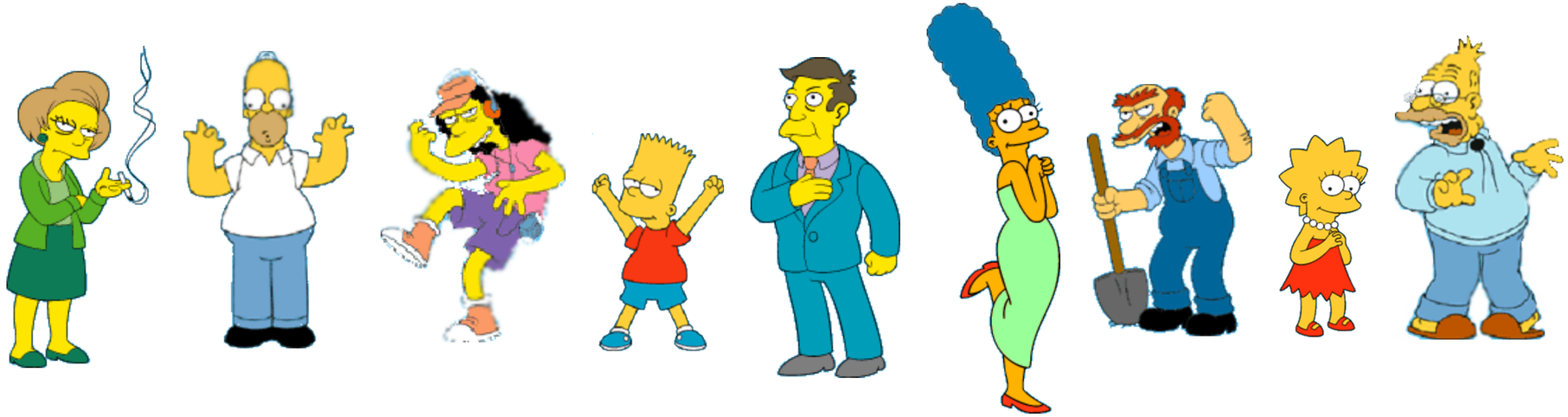
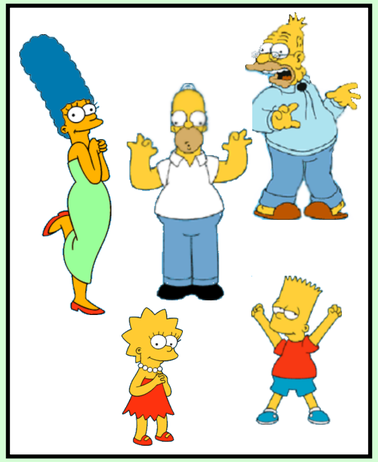


Deep Learning For Clustering

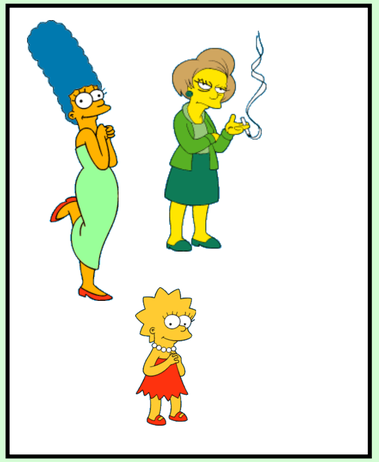
What is a natural grouping among these objects?





Simpson's Family

School Employees



Females

Males

Cluster Analysis

- 將相似的事物歸類。它的原則是同一類中有較大相似性，不同類有較大差異性。
- 非監督式學習
- 常用在發現不同的客戶群，了解客戶群的內在特點和規律

Agenda

Introduction
methods
issues

Traditional Cluster

Deep Embedding Cluster

Introduction
Flow

Introduction
AE methods

Deep Cluster

Hands On

Traditional Cluster

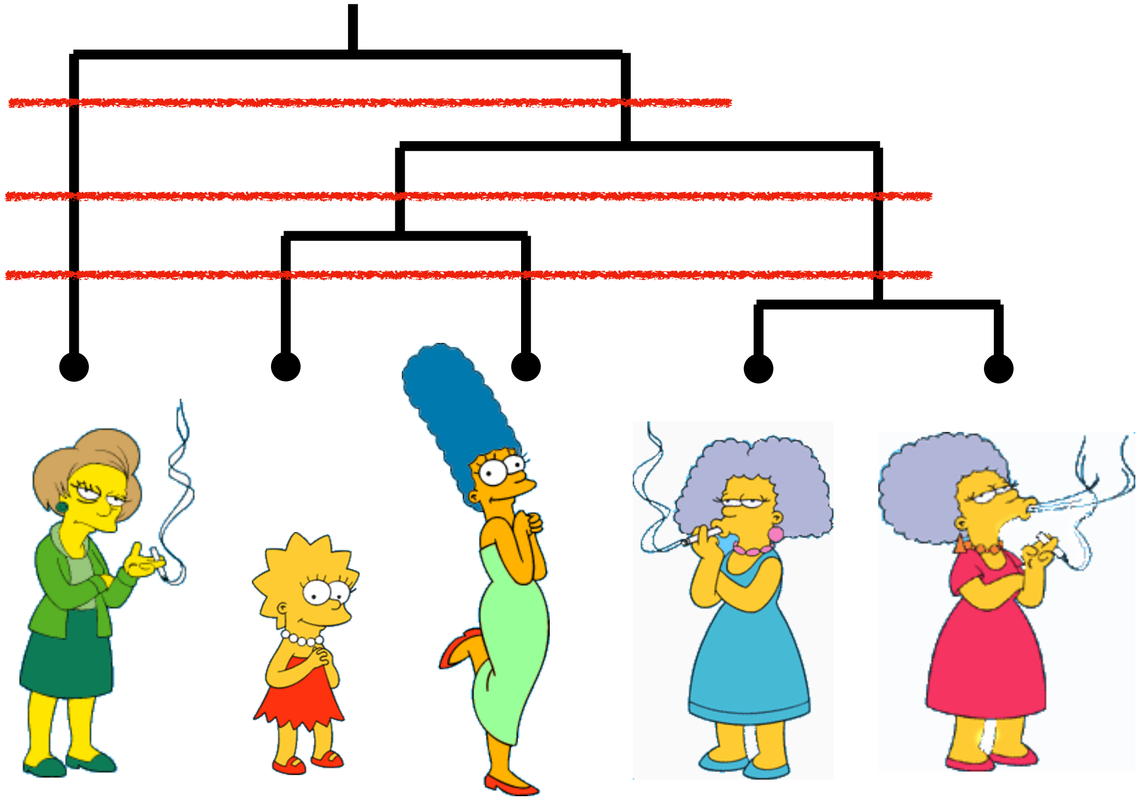
- 距離 -

  歐式距離、曼哈頓距離、馬氏距離....

- 階層式分群法 -

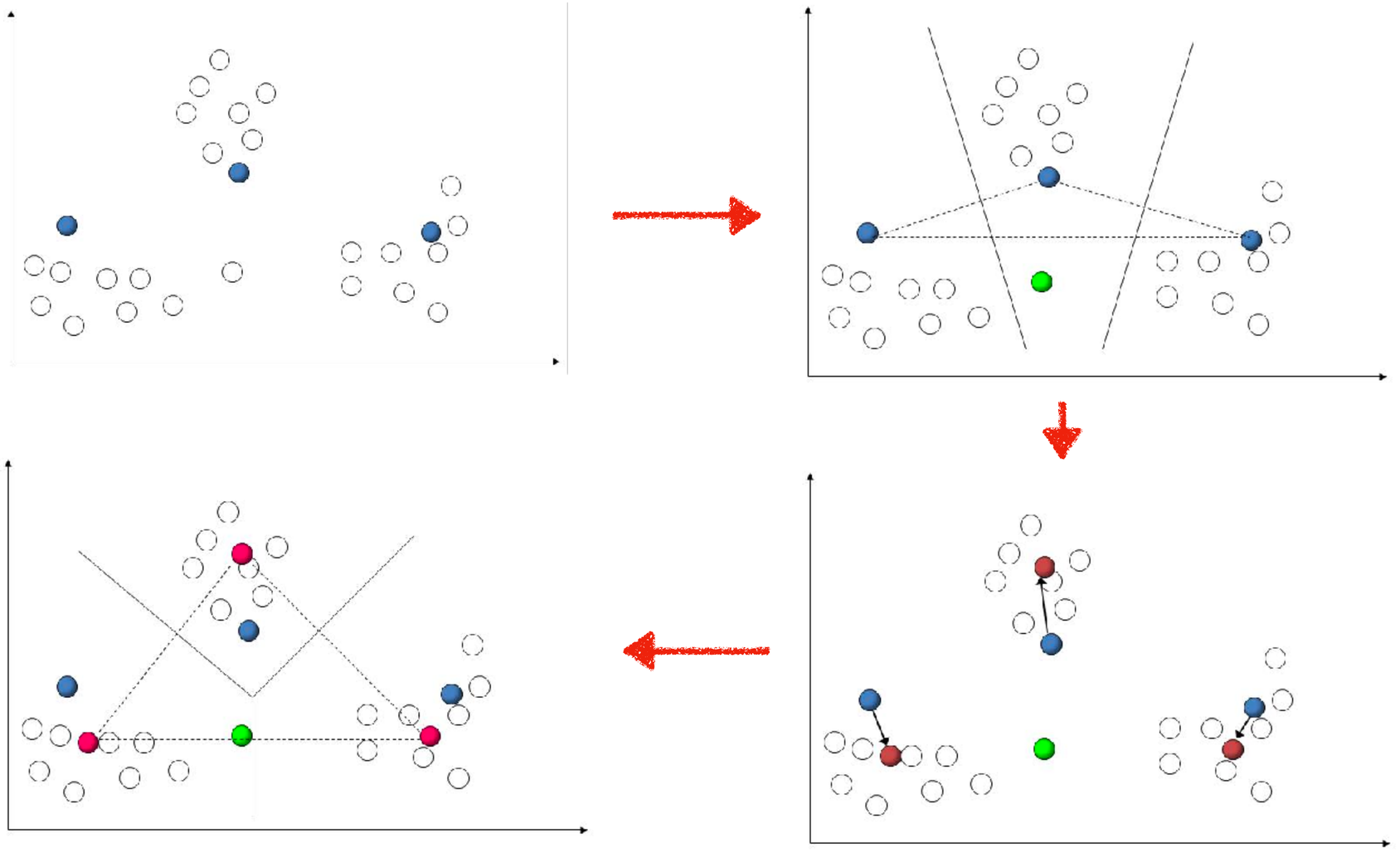
  凝聚層次法 - 自我一群，逐漸聚合

  分離分層法 - 全部一群，直到完全分離



Traditional Cluster

- 分割式分群法 - KMeans

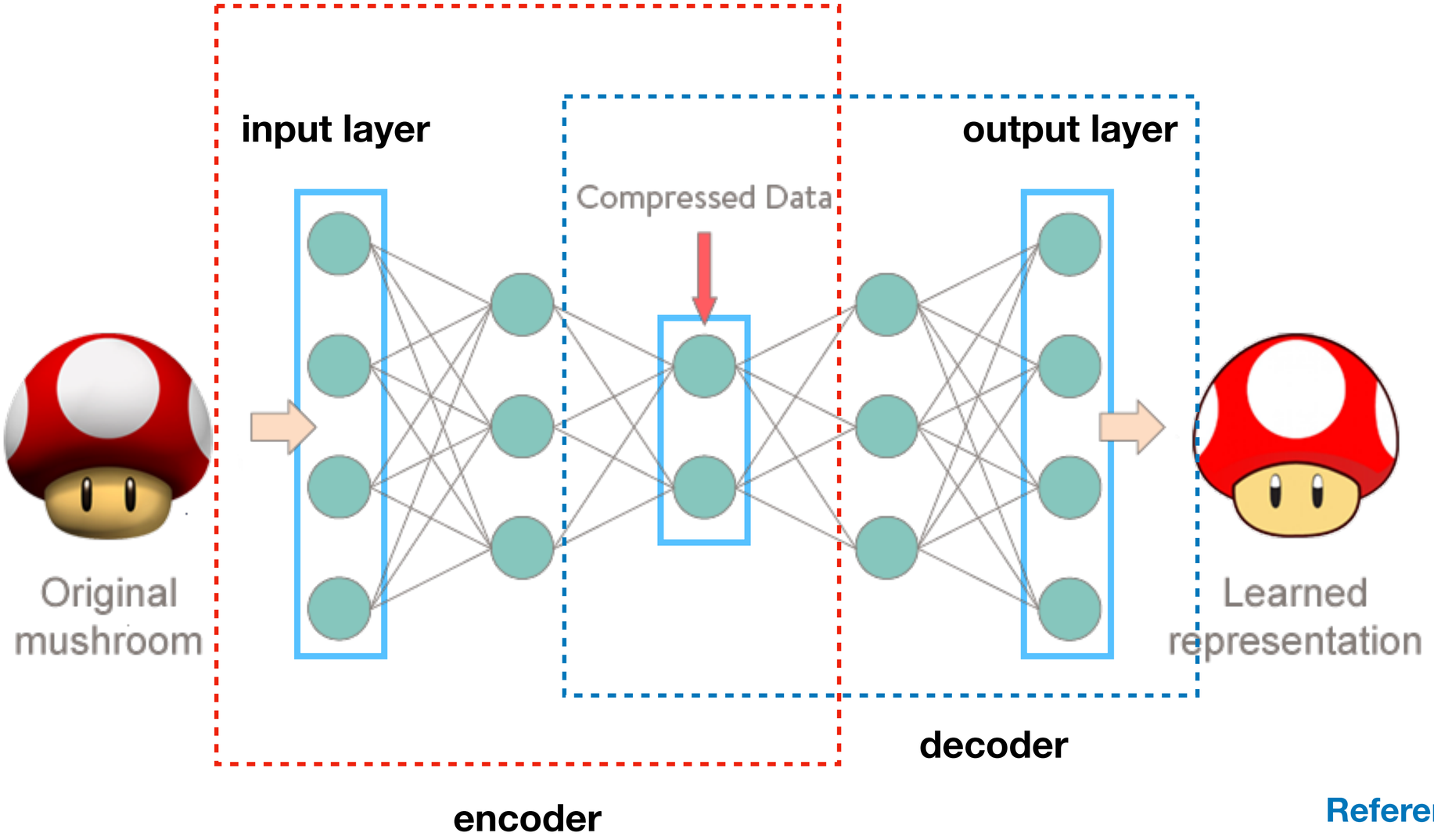


Traditional Cluster

- 高維度和大型數據表現不佳 （大部分我們遇到的都是高維度）
- 高維空間數據分佈稀疏，數據間距離幾乎相等
- 高維空間存在大量無關屬性

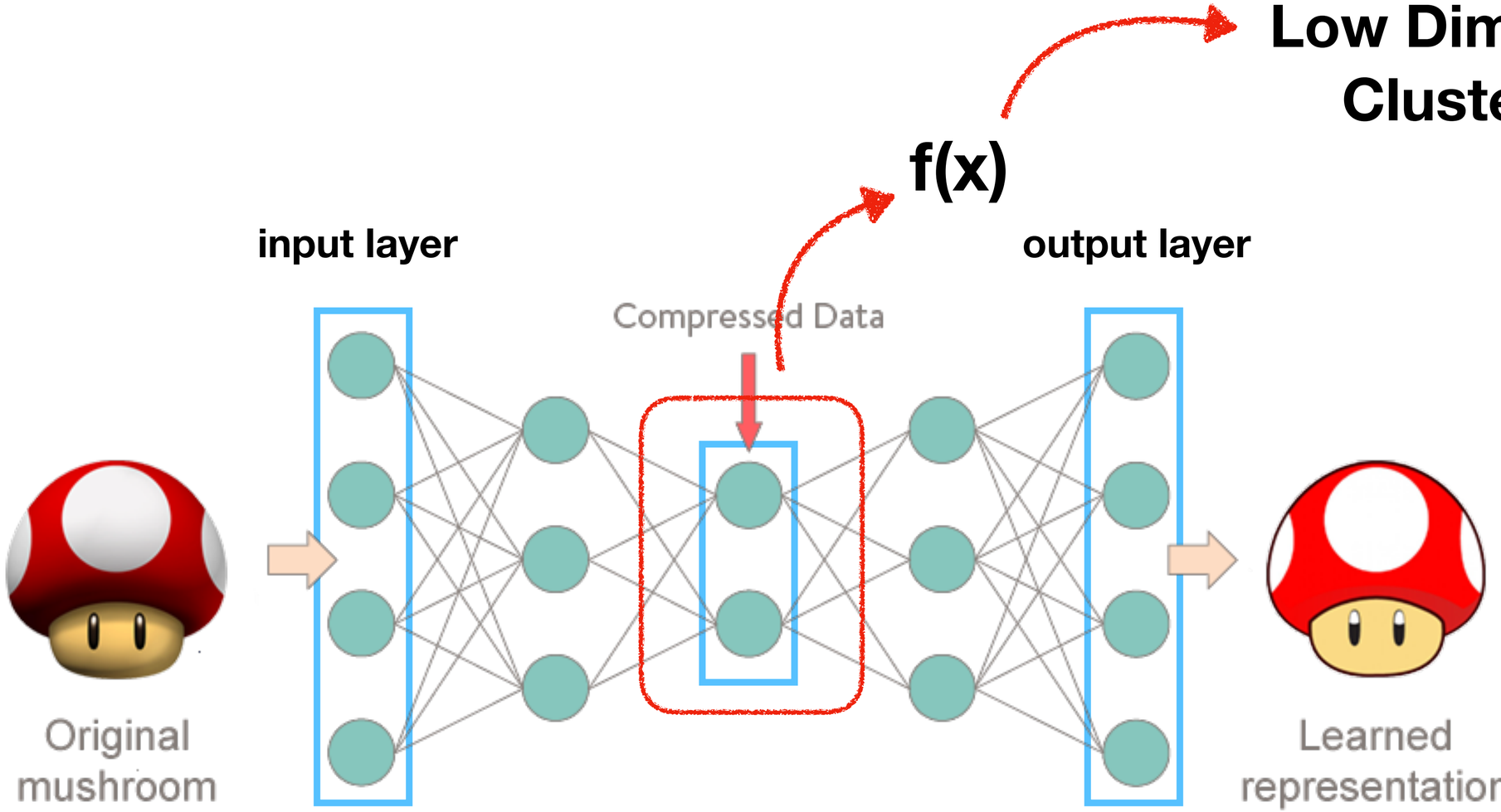
實際遇到的狀況都是高維度
要怎麼解決？

Autoencoder = Encoder + Decoder



Reference from BGG

Autoencoder = Encoder + Decoder



Reference from BGG

Deep Cluster = AutoEncoder + Traditional Cluster

有那麼簡單就好了！！！

AutoEncoder 跟數據相關程度很高

僅適用於與訓練集相似的樣本

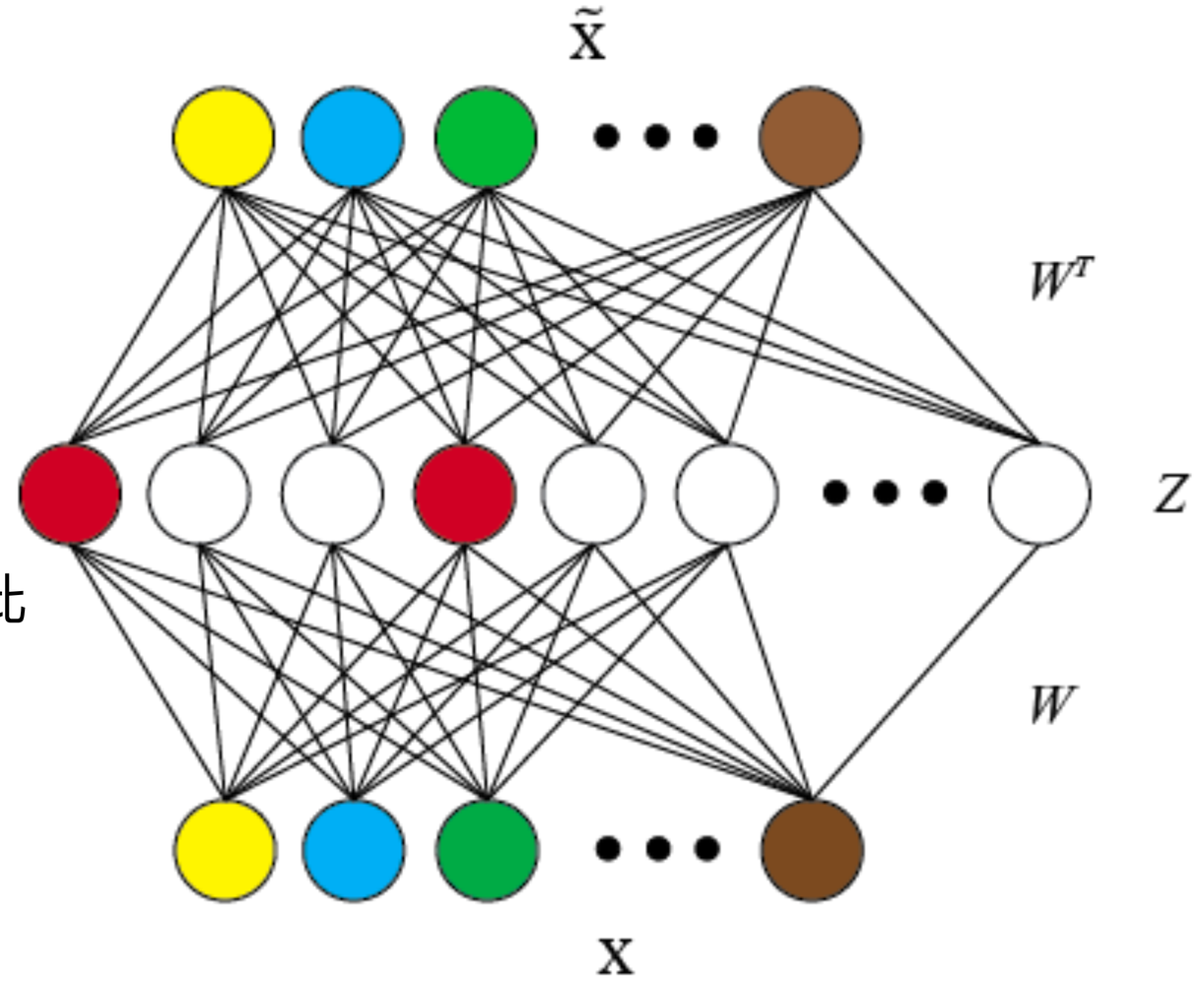
Sparse AE

L2 Regularizer

如果W的分量大的話就抑制多一點，如果分量小就抑制少一點，會留下很多不為0的微小分量

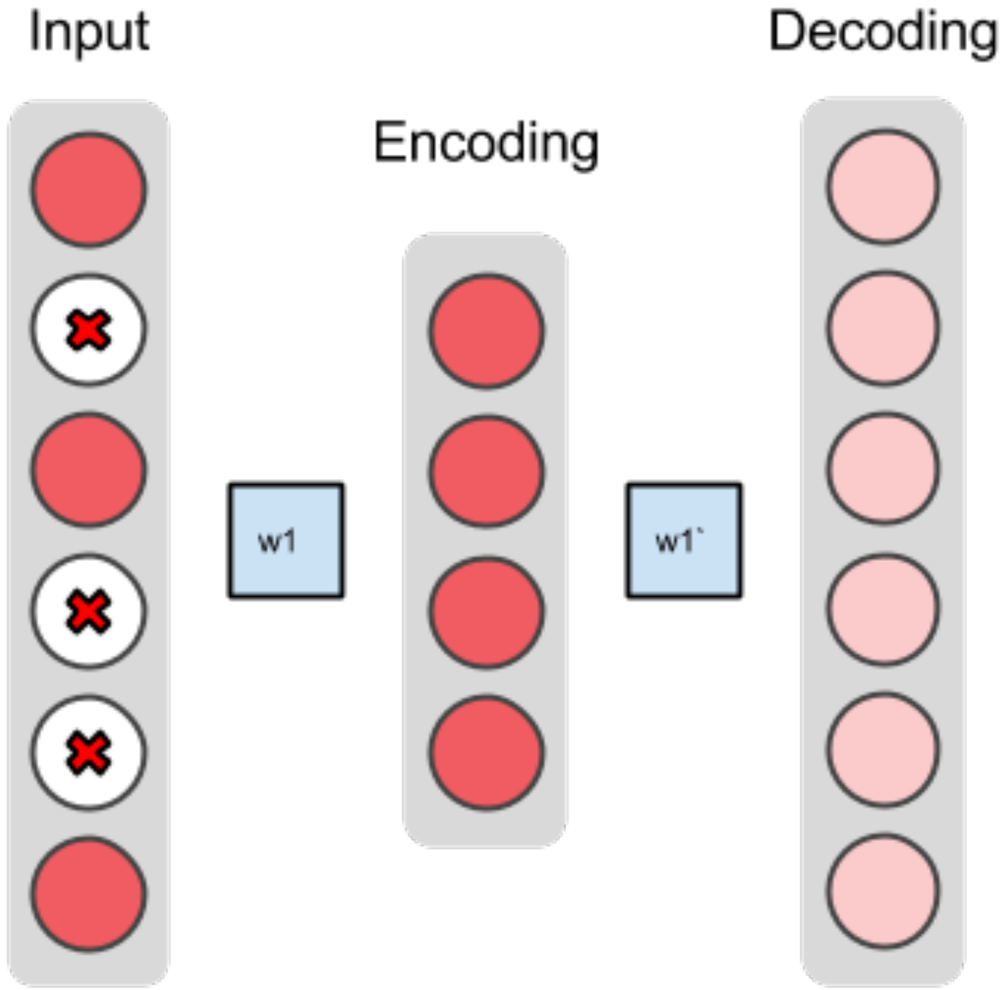
Weight-Elimination L2 Regularizer

不管W大或小，它受到抑制的值大小接近的，因此就可以使得部分W可以為0，達成Sparse的目的。

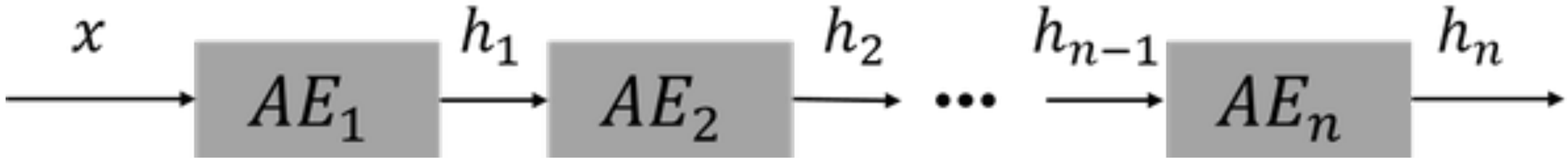


Denoising AE

- 故意破壞原始數據
- 盡可能的逼近還原未被污染的原數據
- 能够在一定程度上對抗原始數據的污染、缺失等情況



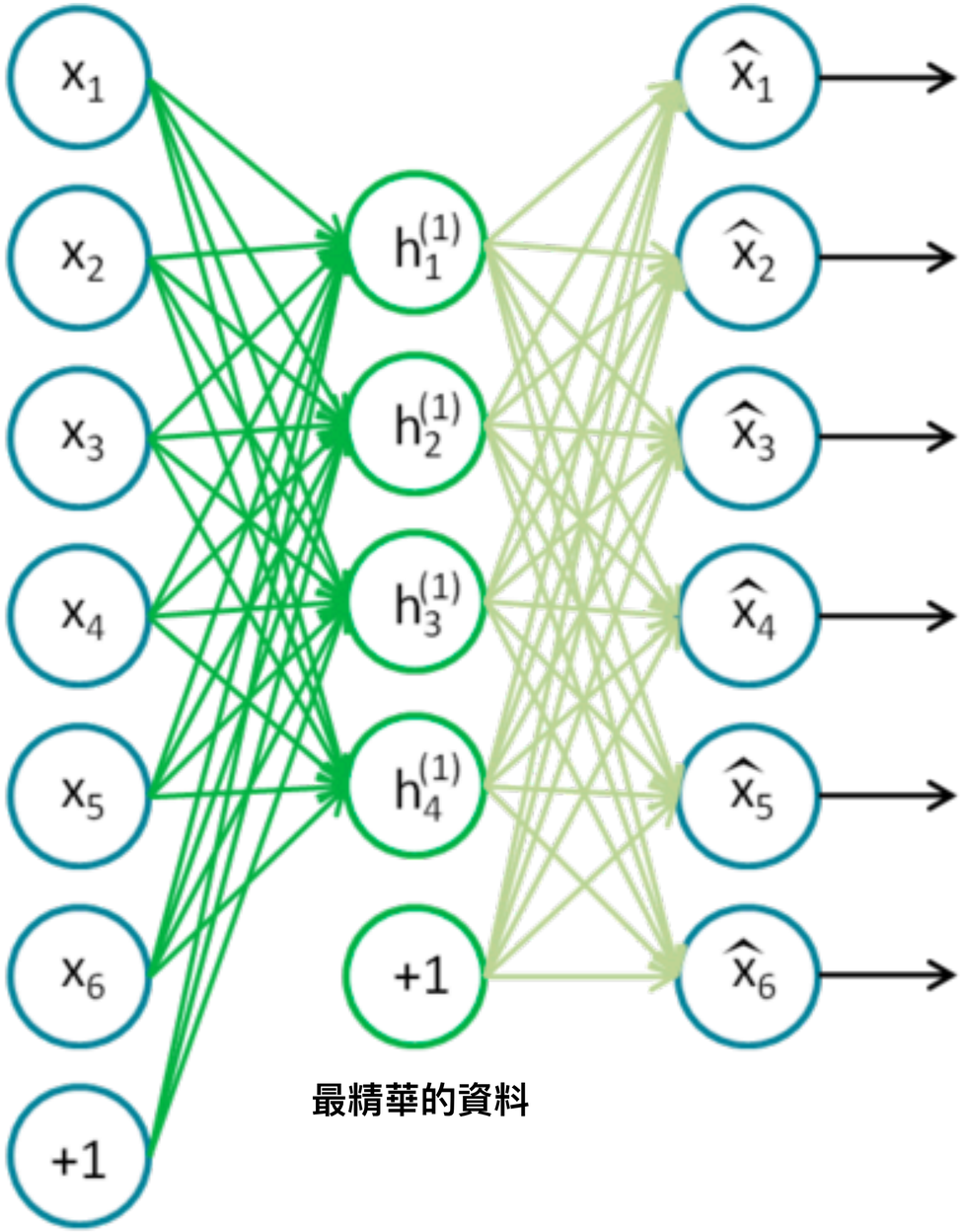
Stacked AE



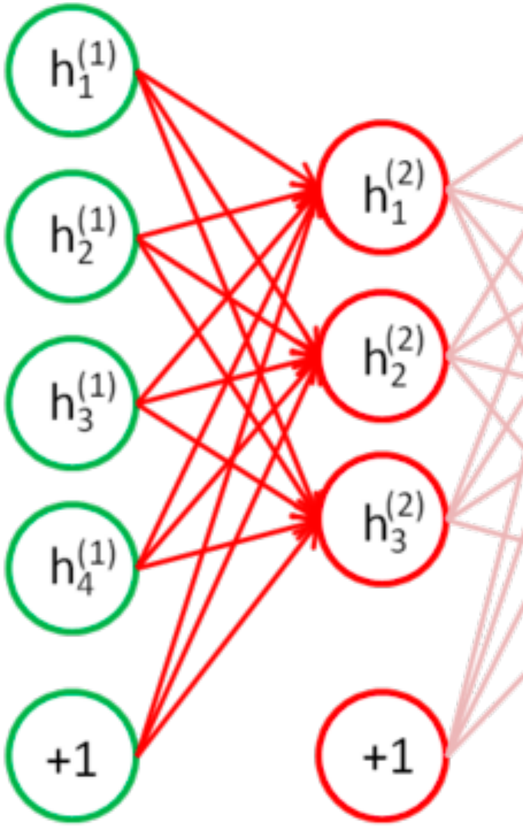
- 逐層進行
- 上一層Input是下一層Output
- 每層Input Output相同

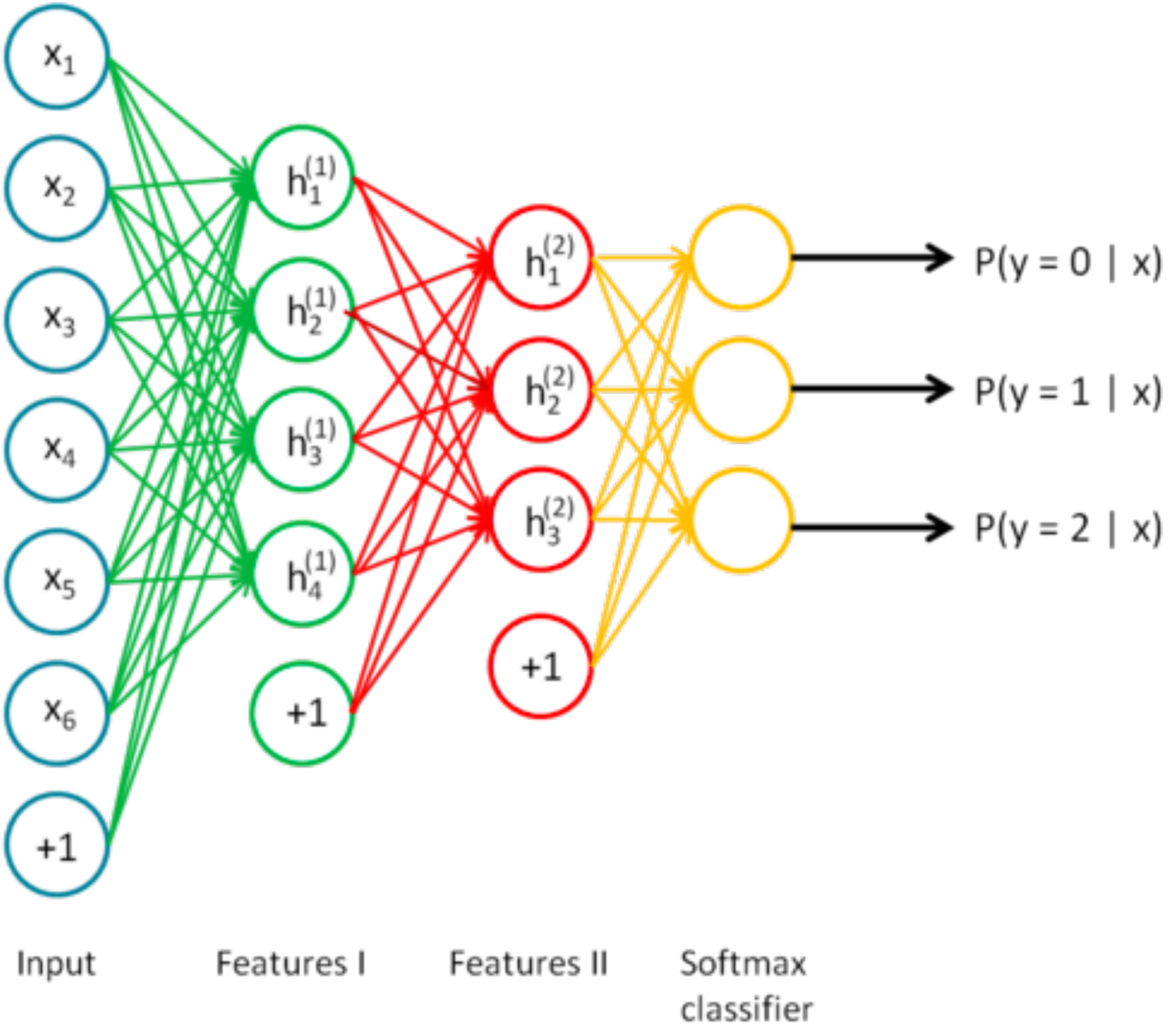
- 舉例來說要做n -> m -> k 的網絡
  實作上先做 n -> m -> n 得到 n -> m的變換
  再做 m -> k -> m 得到 m -> k的變換

Stack 將AutoEncoder改成深度結構


x_1
x_2
x_3
x_4
x_5
x_6
+1
$h_1^{(1)}$
$h_2^{(1)}$
$h_3^{(1)}$
$h_4^{(1)}$
+1
$\hat{x}_1$
$\hat{x}_2$
$\hat{x}_3$
$\hat{x}_4$
$\hat{x}_5$
$\hat{x}_6$
Input
Features I
Output

最精華的資料


$h_1^{(1)}$
$h_2^{(1)}$
$h_3^{(1)}$
$h_4^{(1)}$
+1
$h_1^{(2)}$
$h_2^{(2)}$
$h_3^{(2)}$
+1
Input
(Features I)
Features II


x_1
x_2
x_3
x_4
x_5
x_6
+1
$h_1^{(1)}$
$h_2^{(1)}$
$h_3^{(1)}$
$h_4^{(1)}$
+1
$h_1^{(2)}$
$h_2^{(2)}$
$h_3^{(2)}$
+1
P(y = 0 | x)
P(y = 1 | x)
P(y = 2 | x)
Input
Features I
Features II
Softmax classifier

Deep Embedded Clustering

將上述集大成

Two phases:

(1) parameter initialization with a **Deep Autoencoder** (Vincent et al., 2010)

(2) parameter optimization (i.e. **KL divergence** clustering)

KL Divergence

兩個機率分佈P和Q差別的非對稱性的度量，也就是**機率分布的差異**

量化機率分布的訊息

$$H(X)=\sum_{x\in X}P(x)\log[1/P(x)]$$

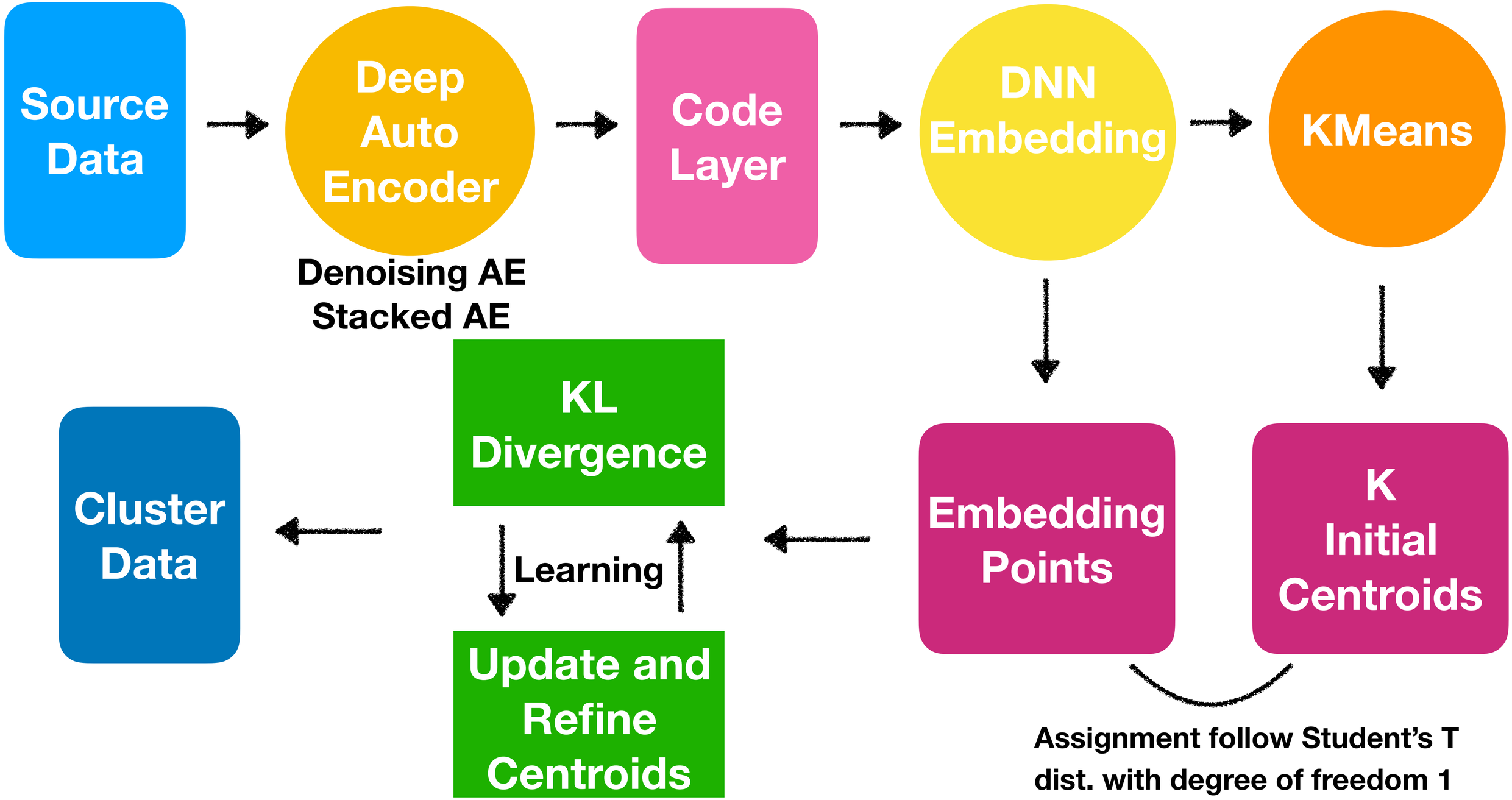
兩個分布的距離

$$D_{KL}(Q||P)=\sum_{x\in X}Q(x)[\log(1/P(x))] - \sum_{x\in X}Q(x)[\log[1/Q(x)]]=\sum_{x\in X}Q(x)\log[Q(x)/P(x)]$$

方法A得到四個類別的機率為0.4、0.3、0.2、0.1

方法B得到四個類別的機率為0.1、0.2、0.3、0.4

$$\text{Distance(A,B)}$$
$$=0.1*\log(0.1/0.4)+0.2*\log(0.2/0.3)+0.3*\log(0.3/0.2)+0.4*\log(0.4/0.1)$$


Source Data
Deep Auto Encoder
Denoising AE
Stacked AE
Code Layer
DNN Embedding
KMeans
Embedding Points
K Initial Centroids
Assignment follow Student's T dist. with degree of freedom 1
KL Divergence
Learning
Update and Refine Centroids
Cluster Data

Dataset	# Points	# classes	Dimension	% of largest class
MNIST (LeCun et al., 1998)	70000	10	784	11%
STL-10 (Coates et al., 2011)	13000	10	1428	10%
REUTERS-10K	10000	4	2000	43%
REUTERS (Lewis et al., 2004)	685071	4	2000	43%

Method	MNIST	STL-HOG	REUTERS-10k	REUTERS
k-means	53.49%	28.39%	52.42%	53.29%
AE+k-means	81.84%	33.92%	66.59%	71.97%
AE+LDMGI	83.98%	32.04%	42.92%	N/A
AE+SEC	81.56%	32.29%	61.86%	N/A
DEC (ours)	**84.30%**	**35.90%**	**72.17%**	**75.63%**

Reference

- https://morvanzhou.github.io/tutorials/machine-learning/keras/2-6-autoencoder/
  AutoEncoder
- https://allenlu2007.wordpress.com/2017/07/12/sparse-autoencoder/
  http://www.ycc.idv.tw/YCNote/post/43
  Sparse AutoEncoder
- http://blog.csdn.net/u010089444/article/details/52618864
  http://www.jmlr.org/papers/volume11/vincent10a/vincent10a.pdf
  Denoising AutoEncoder
- https://www.jianshu.com/p/51d5639c2c71
  Stacked AutoEncoder
- https://www.youtube.com/watch?v=ci0xtJwZdzk
  https://www.youtube.com/watch?v=_DPLpV-vZT4
  AutoEncoder Clustering
- http://www.jmlr.org/papers/volume11/vincent10a/vincent10a.pdf
  Deep AutoEncoder
- https://arxiv.org/pdf/1511.06335.pdf
  https://github.com/XifengGuo/DEC-keras
  Deep Embedding Cluster