Representations and Autoencoder

11-785 Introduction to Deep Learning

- lecture 19 -

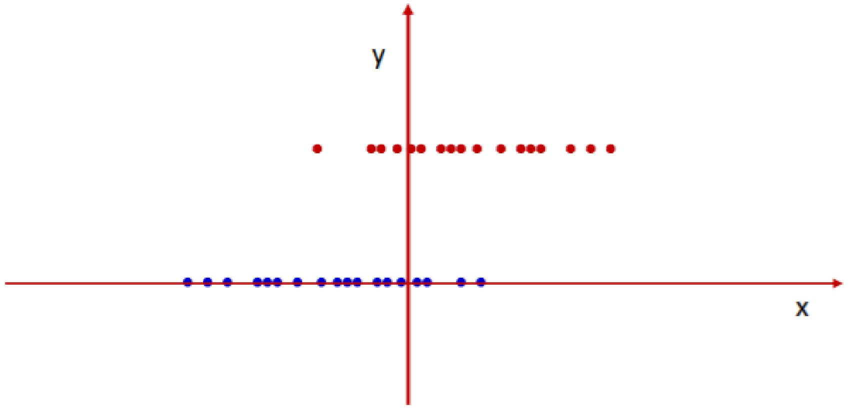
TAVE Research DL001
Heeji Won

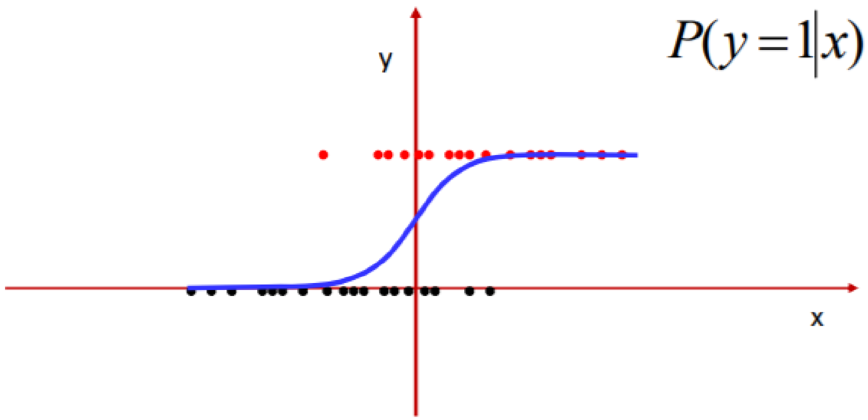
Contents

1. Estimate a classifier model
2. Role of the layers
3. Autoencoder

Contents

01. Estimate a classifier model





- to represent using a y=+1/−1 notation

$$P(y=1|x)=\frac{1}{1+e^{-(w_0+w_1x)}} \qquad P(y=-1|x)=\frac{1}{1+e^{(w_0+w_1x)}}$$

$$\downarrow$$

$$P(y|x)=\frac{1}{1+e^{-y(w_0+w_1x)}}$$

- Total probability of data

$$P\big((X_1,y_1),(X_2,y_2),\dots,(X_N,y_N)\big) = \prod_i P(X_i)P(y_i|X_i)$$

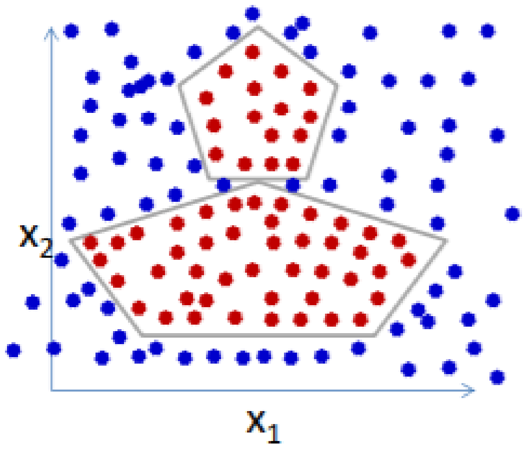
$$=\prod_i P(X_i)\prod_i \frac{1}{1+e^{-y_i(w_0+w^TX_i)}}$$

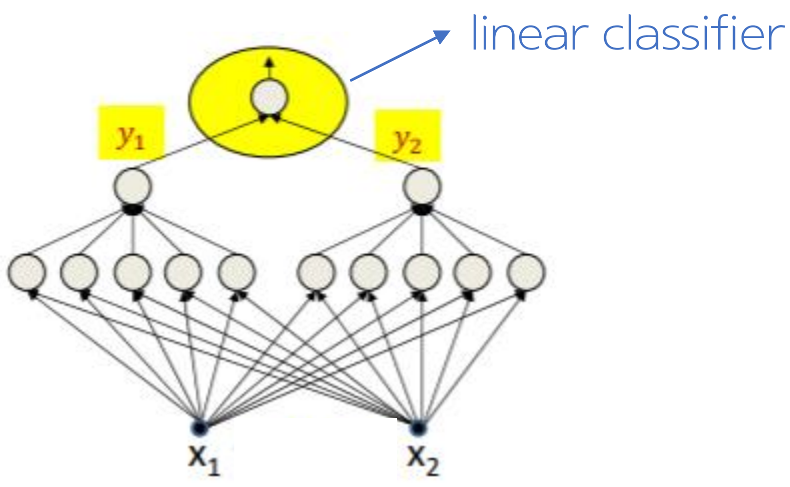
=> $$\boxed{\hat{w}_0,\hat{w}_1 = \underset{w_0,w}{\operatorname{argmin}} \sum_i \log\left(1+e^{-y_i(w_0+w^TX_i)}\right)}$$

Cannot be solved directly, needs gradient descent!

01. Estimate a classifier model

How non-linear classifier?





- y_1 and y_2 make a transformation that data from non-linear classes to linearly separable features

  => the role of **Feature Extraction**

- y_{out} compute a posterior probability $P(y|y_1, y_2)$

$$y_{out} = \frac{1}{1+\exp(b+W^T Y)} = \frac{1}{1+e\quad(b+W^T f(X))}$$

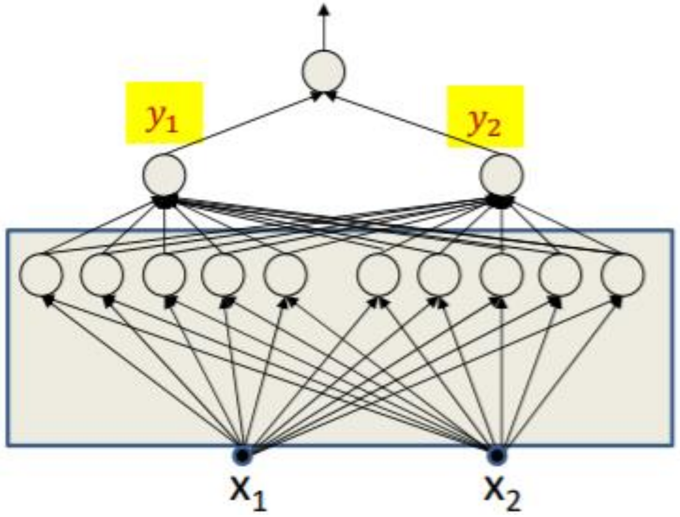
- transformation from input to y spaces is deterministic

  => So, $\boldsymbol{y_{out}}$ **virtually compute a posterior probability** $\boldsymbol{P(y|x)}$

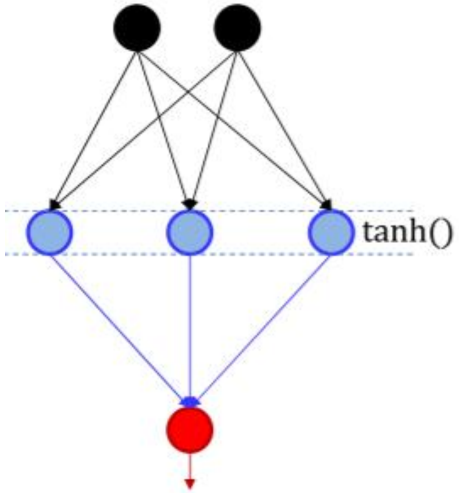
Contents

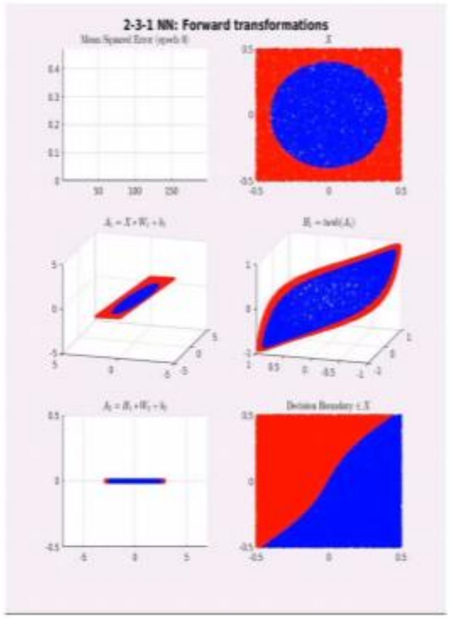
02. Role of the layers

How about the lower layers?

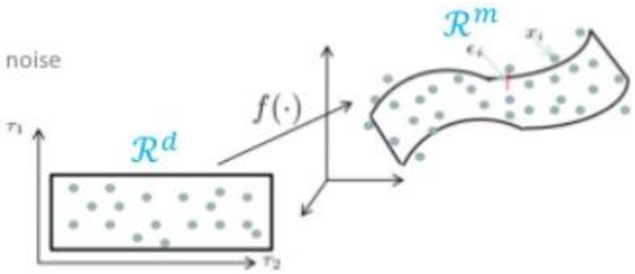


<the behavior of the layers>





- ✓ **Manifold hypothesis** : For separable classes, the classes are linearly separable on a non-linear manifold

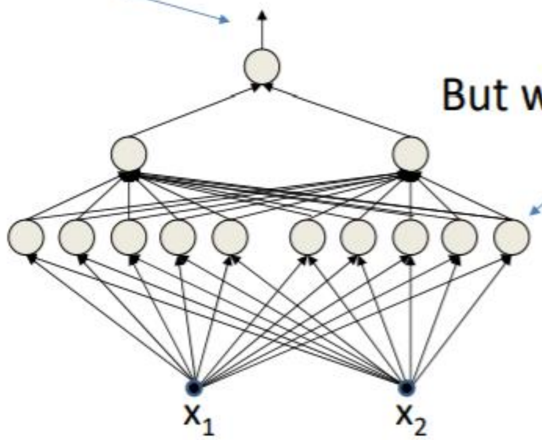


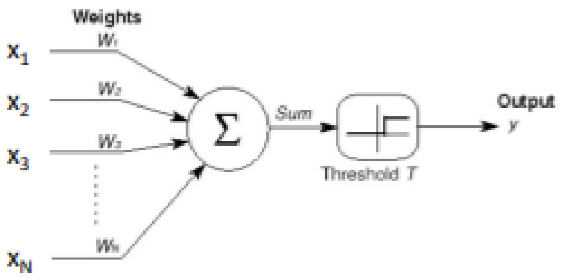
02. Role of the layers

The intermediate layers

We've seen what the network learns here

But what about here?



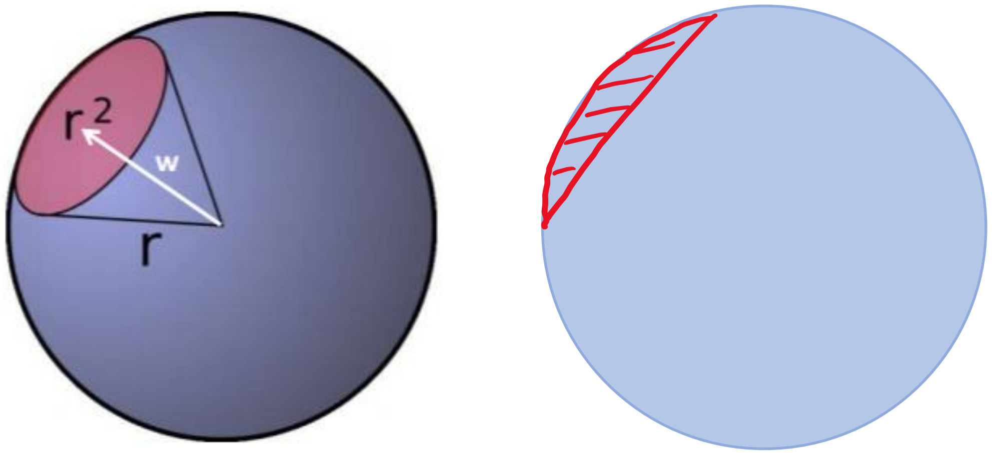


$$X^T W > T$$

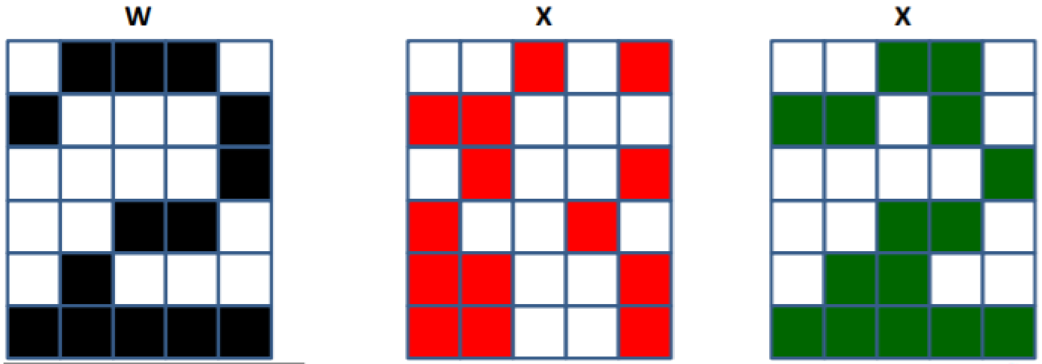
$$\cos\theta > \frac{T}{|W||X|}$$

$$\theta < \cos^{-1}\left(\frac{T}{|W||X|}\right)$$

- The weight as a 'template' !



- as correlation filters



$$y = \begin{cases} 1 \text{ if } \sum_i w_i x_i \geq T \\ 0 \text{ else} \end{cases}$$

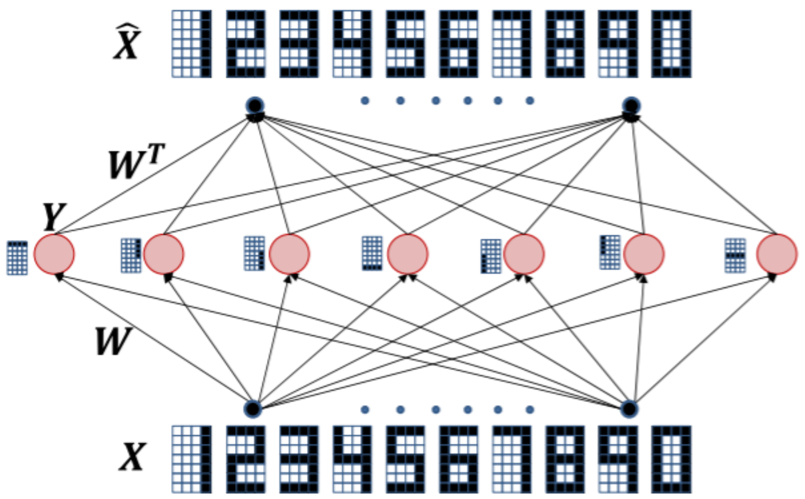
Correlation = 0.57

Correlation = 0.82

Contents

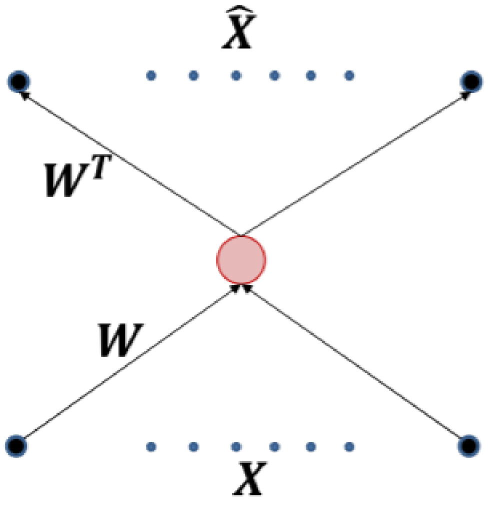
1. Estimate a classifier model
2. Role of the layers
3. Autoencoder

03. Autoencoder



- The signal could be reconstructed using these features
- by using W^T, simply recompose the detected features
- trained to predict the input itself

$\Rightarrow$ Autoencoder!



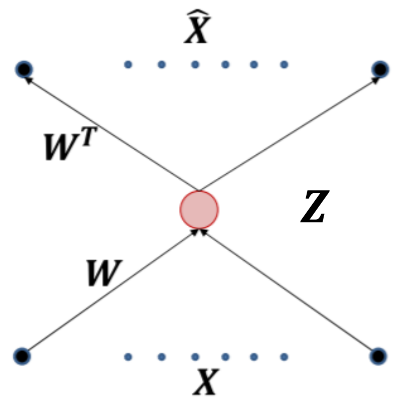
- This is a simplest autoencoder
- training by minimizing L2 divergence

$$\hat{\mathrm{x}} = w^T w\mathrm{x}$$
$$div(\hat{\mathrm{x}}, \mathrm{x}) = \|\mathrm{x} - \hat{\mathrm{x}}\|^2 = \|\mathrm{x} - \mathrm{w}^T w\mathrm{x}\|^2$$

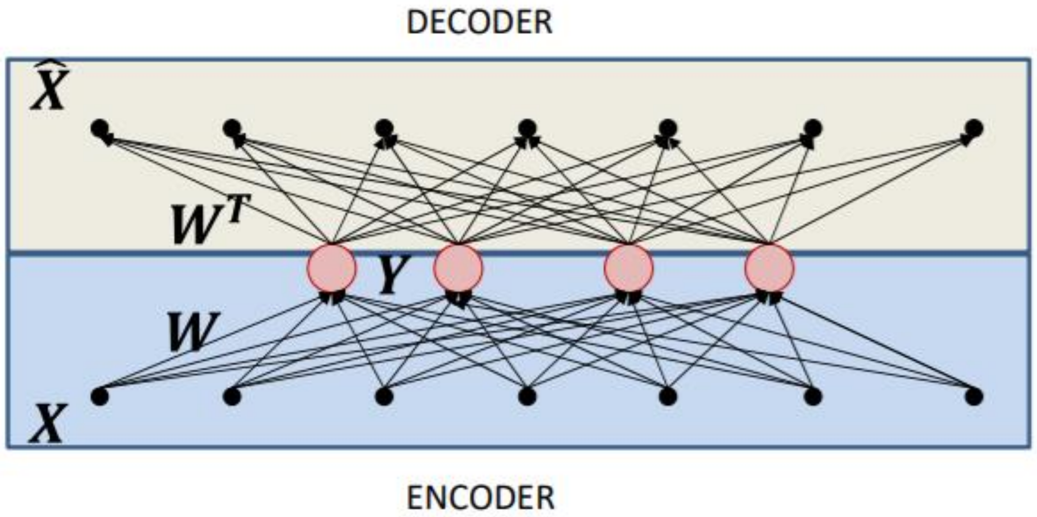
$$\widehat{W} = \underset{W}{\operatorname{argmin}}\, E[\|\mathrm{x} - \mathrm{w}^T w\mathrm{x}\|^2]$$

This is just PCA!

03. Autoencoder



- Z will always lie on W space!



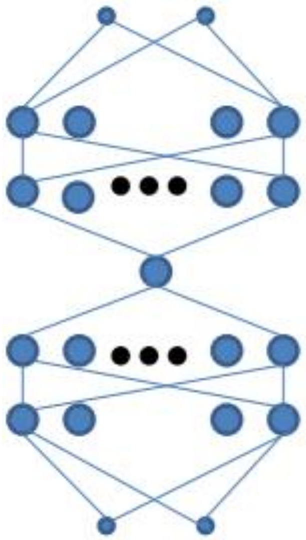
$\mathbf{Y} = \mathbf{W}\mathbf{X}$ $\quad\quad$ $\hat{\mathbf{X}} = \mathbf{W}^T\mathbf{Y}$

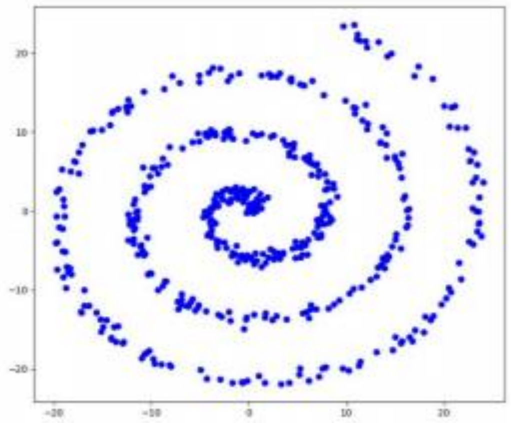
$$E = \|\mathbf{X} - \mathbf{W}^T\mathbf{W}\mathbf{X}\|^2$$

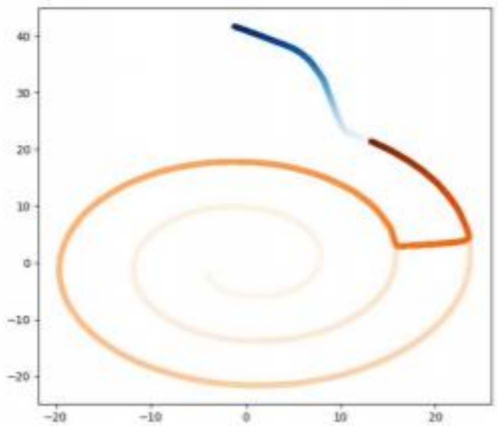
- This is still just PCA!
- The output of hidden layer will be in the principal subspace

- Encoder : The "Analysis" net which computes the hidden representation
- Decoder : The "Synthesis" which recomposes the data from the hidden representation

03. Autoencoder



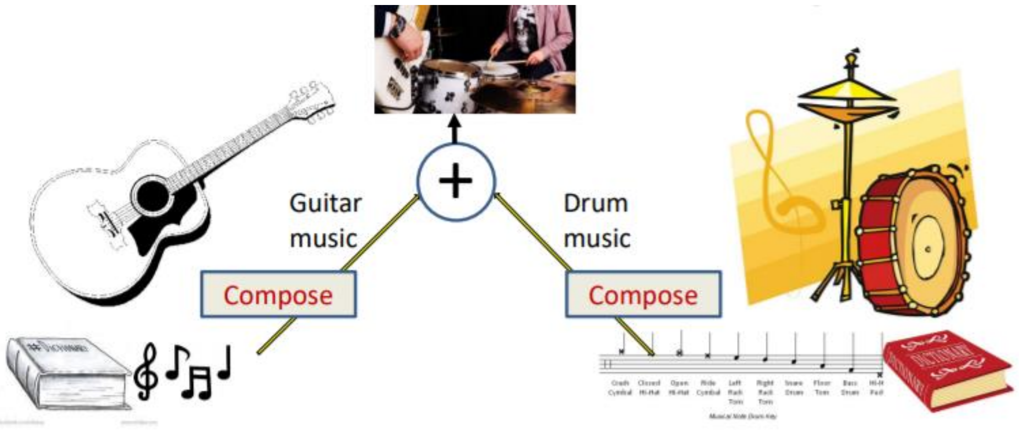




- When the hidden layer has non-linear activation, the net performs nonlinear PCA
- Varying the hidden layer value only generates data along the learned manifold
- But may not generalize beyond the manifold

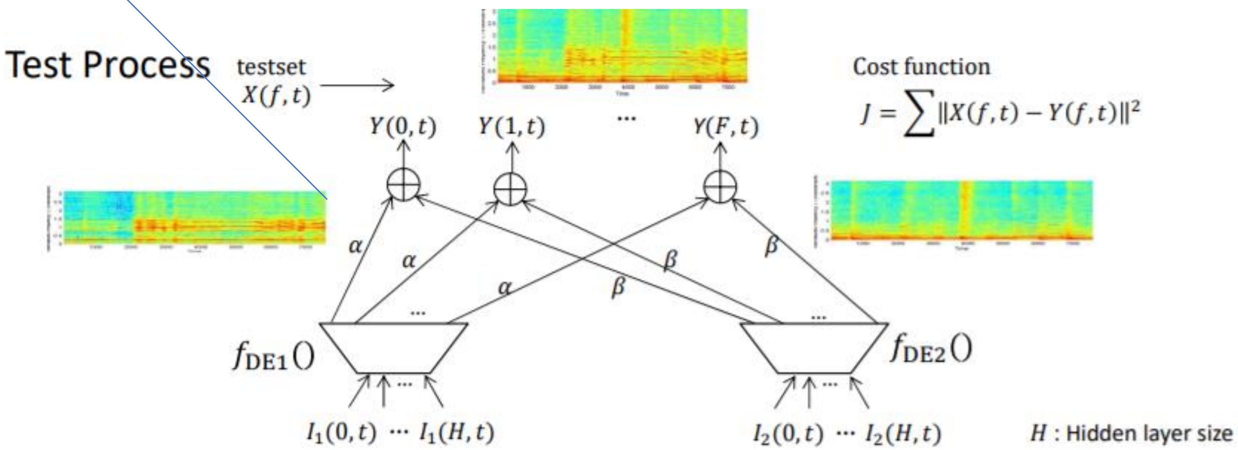
03. Autoencoder

Examples



Given a mixed sound from multiple sources, separate out the sources!

Learning AE dictionaries for each source



- ✓ Given mixed signal and source dictionaries, find excitation that best **recreates** mixed signal
- ✓ **Intermediate results** are separated signals

Thank you