Unsupervised learning :- we work with unlabelled data.

$$\bar{\mathcal{D}} = \{ \underbrace{x}_{\text{vector}} i \}_{i=1}^{N} = \begin{bmatrix} | & | & & | \\ x^1 & x^2 & \cdots & x^M \\ | & | & & | \end{bmatrix}$$

$\underbrace{N}_{\text{No. of data}} \times \underbrace{M}_{\text{No. of feature}}$

* What can be done with this data?

(1) Hidden structure in the data.

Reduced representation of the data.

Clustering.

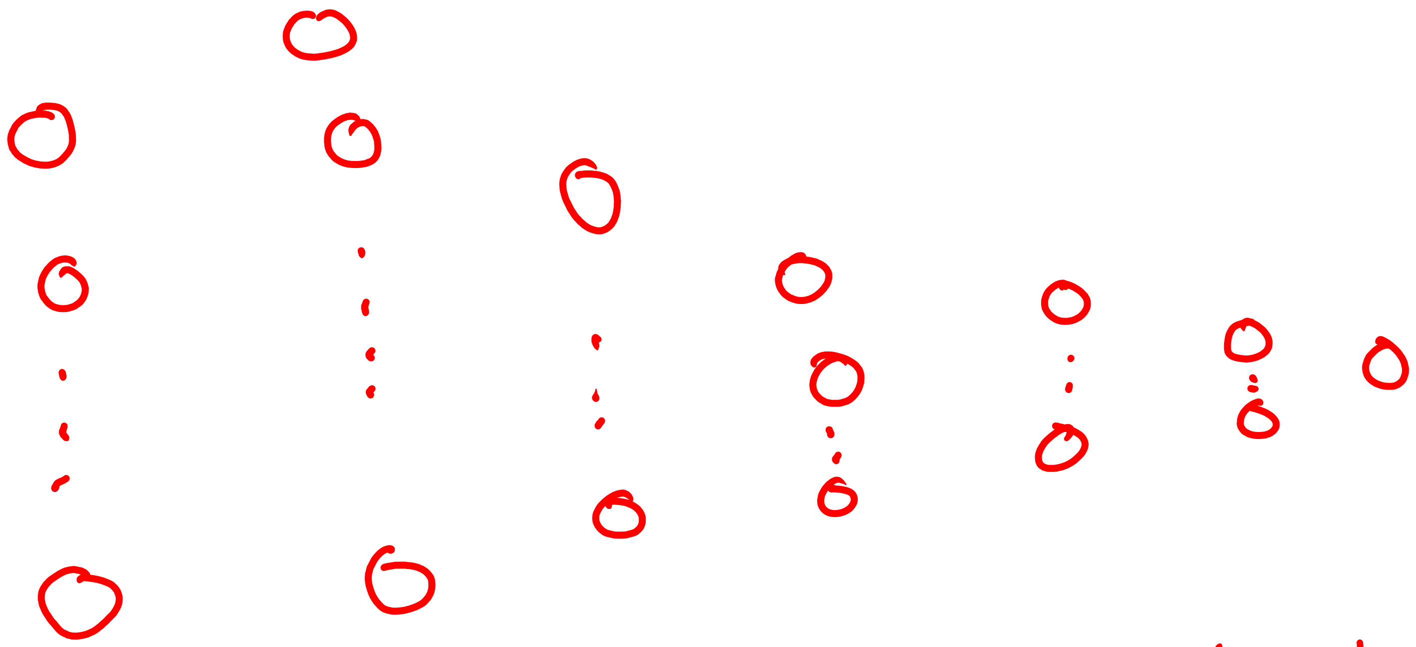
$$X \in \mathbb{R}^{N \times M} \longrightarrow Z \in \mathbb{R}^{N \times K}, \quad K << M$$

↳ Less memory

↳ $X \rightleftarrows Z$

Eg., PCA, Autoencoders

* One more application of reduced representation is development of semi-supervised learning algorithm.

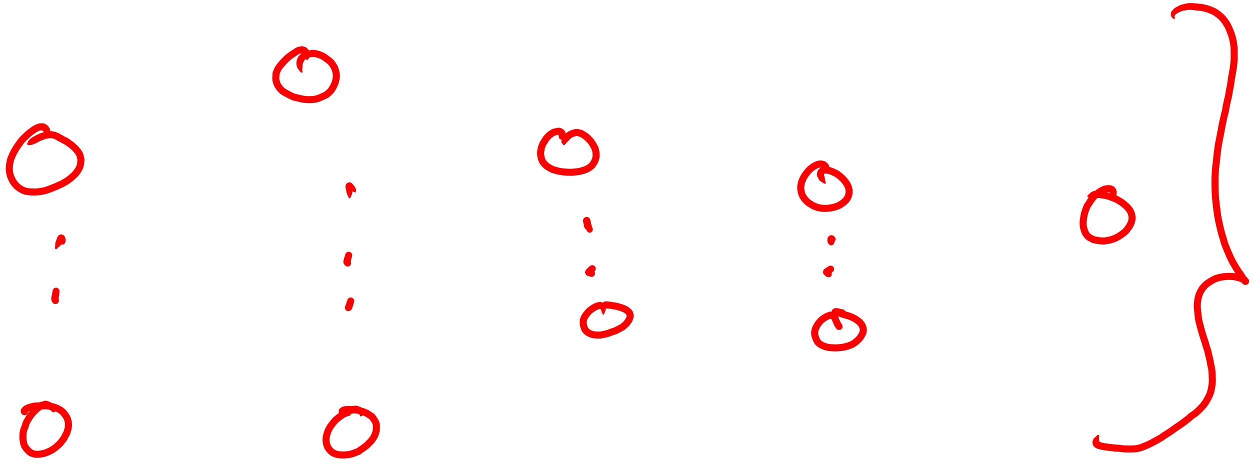


└→ Lots of parameters.

* We can use semi-supervised learning algorithm for efficiency.

⟨1⟩ $10000 \xrightarrow{\text{ROM}} 50$

⟨2⟩



$X \in \mathbb{R}^{50} \longmapsto Y \in \mathbb{R}$

* One of the primary challenges in Supervised learning is to label the data.

* Suppose we have $X \in \mathbb{R}^{N \times 10000}$ and $Y \in \mathbb{R}^{500}$

⟨1⟩ use ROM for $X \in \mathbb{R}^{N \times 10,000}$ to $Z \in \mathbb{R}^{N \times 50}$

⟨2⟩ $Z \in \mathbb{R}^{500 \times 50}$ to $Y \in \mathbb{R}^{500}$

* One application of clustering is fast labelling of data.

$X \in \mathbb{R}^{N \times 100000}$

→ (1) Cluster

(2) I will check few data from each cluster to determine potential label of data points in a cluster.

<2> Learning distribution of data.

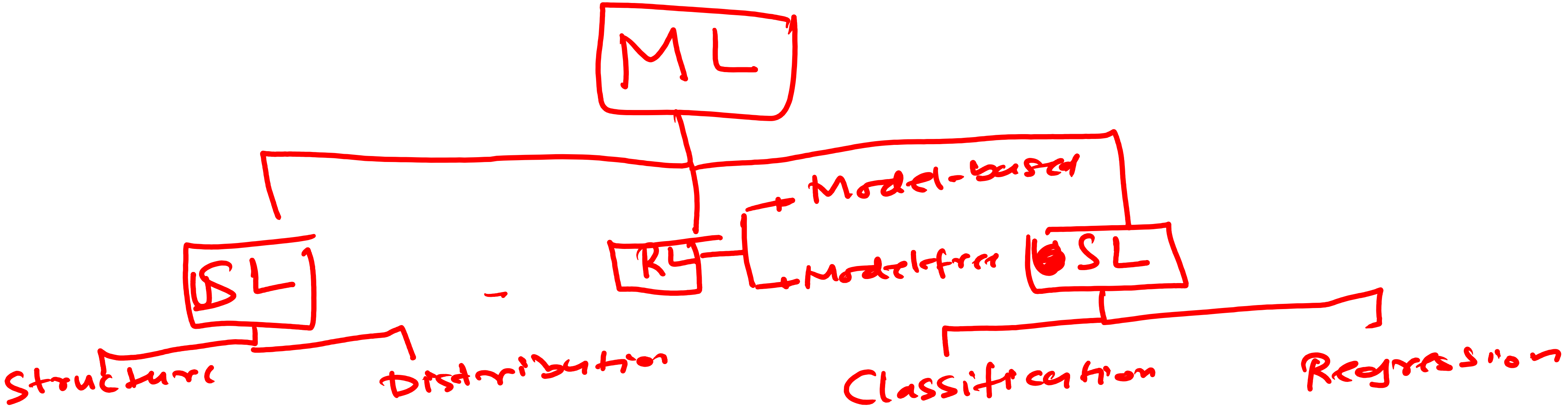
$$X \in \mathbb{R}^{N \times 10000} \rightsquigarrow P(X)$$

$\rightarrow$ 10,000

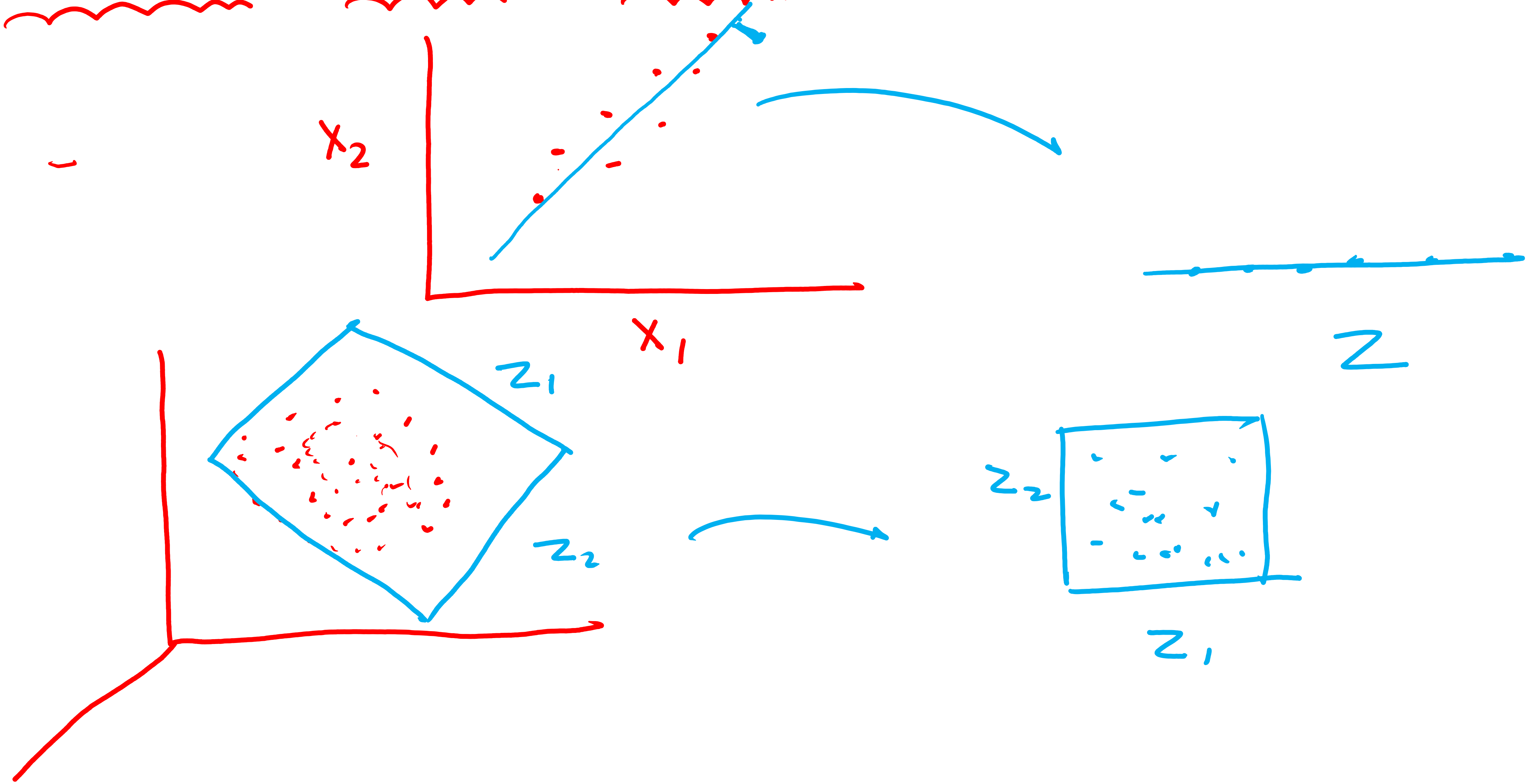
$\downarrow$ Sample it

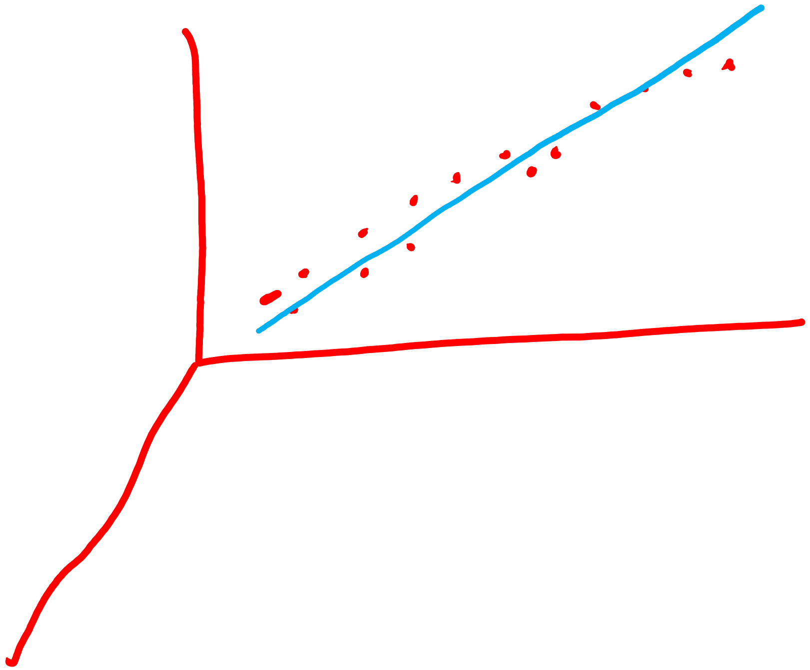
New data.

$$P(X \mid Y = C)$$



* Reduced order model :-



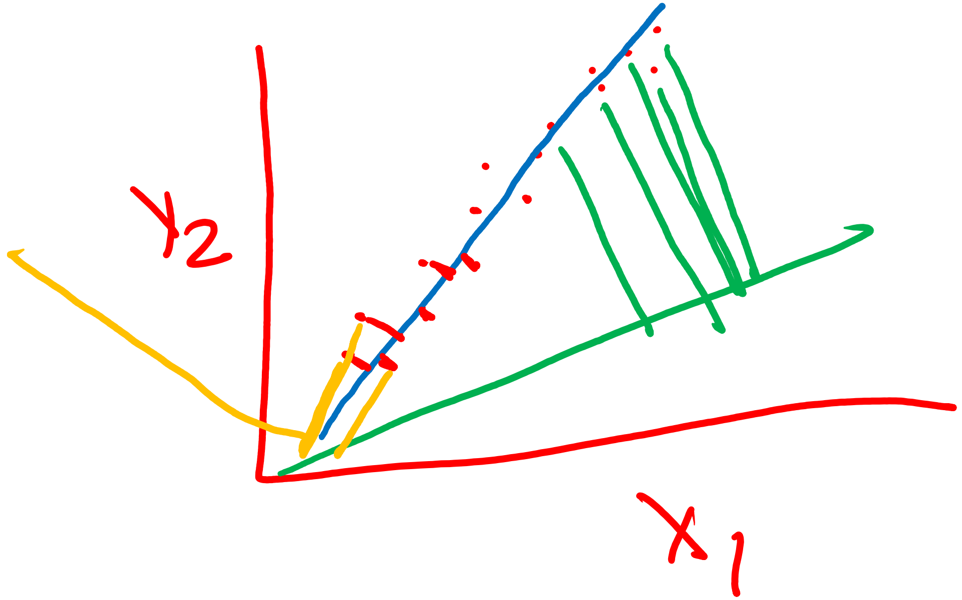


* Generally in ML, we go from 1000 – 100000 to 10 – 100

* The reduced coordinates may or may not have physical significance.

Principal Component Analysis :-

Principal Component ~~is~~ represents coordinate axes which capture maximum information, and PCA deals with finding the principal components.

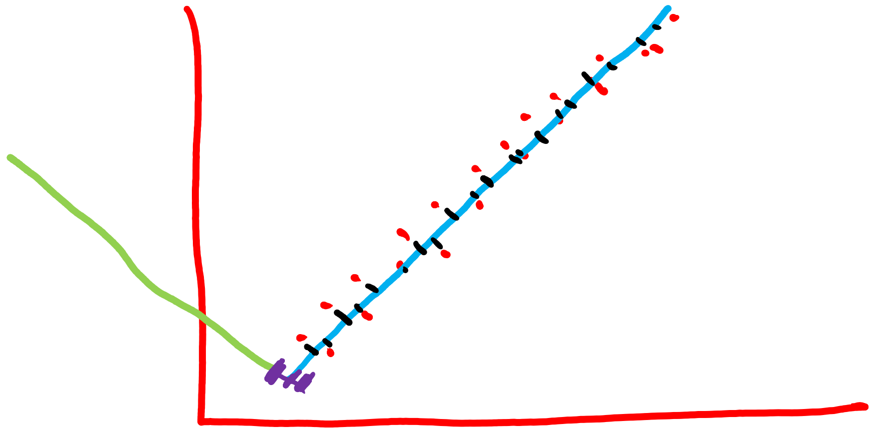


(1) Minimization of projection error.

↓

perpendicular distance b/w the data and the PC.

(2) Maximizing the variance



PCA

(1) Covariance matrix

$$\underbrace{\Sigma}_{\to M \times M} = \frac{1}{N} \sum_{i=1}^{N} \underbrace{(x_i)}_{M\times 1} \underbrace{(x_i)^T}_{1 \times M}$$

$$\Sigma = \begin{bmatrix} Var(x_1) & cov(x_1, x_2) & \cdots & cov(x_1, x_M) \\ & Var(x_2) & & \vdots \\ & & \ddots & \\ & & & Var(x_M) \end{bmatrix}$$

$\to$ $M \times M$ symmetric matrix.

<2> $[U, S, V] = \mathrm{SVD}(\Sigma)$

linear algebra

$\lambda, \phi = \mathrm{eig}(\Sigma)$

MxM diagonal matrix.

with diagonal elements as your eigen values

Eigen vector (MxM)

* We can now reduce the model by only considering the first K Eigen values and Eigen vectors.

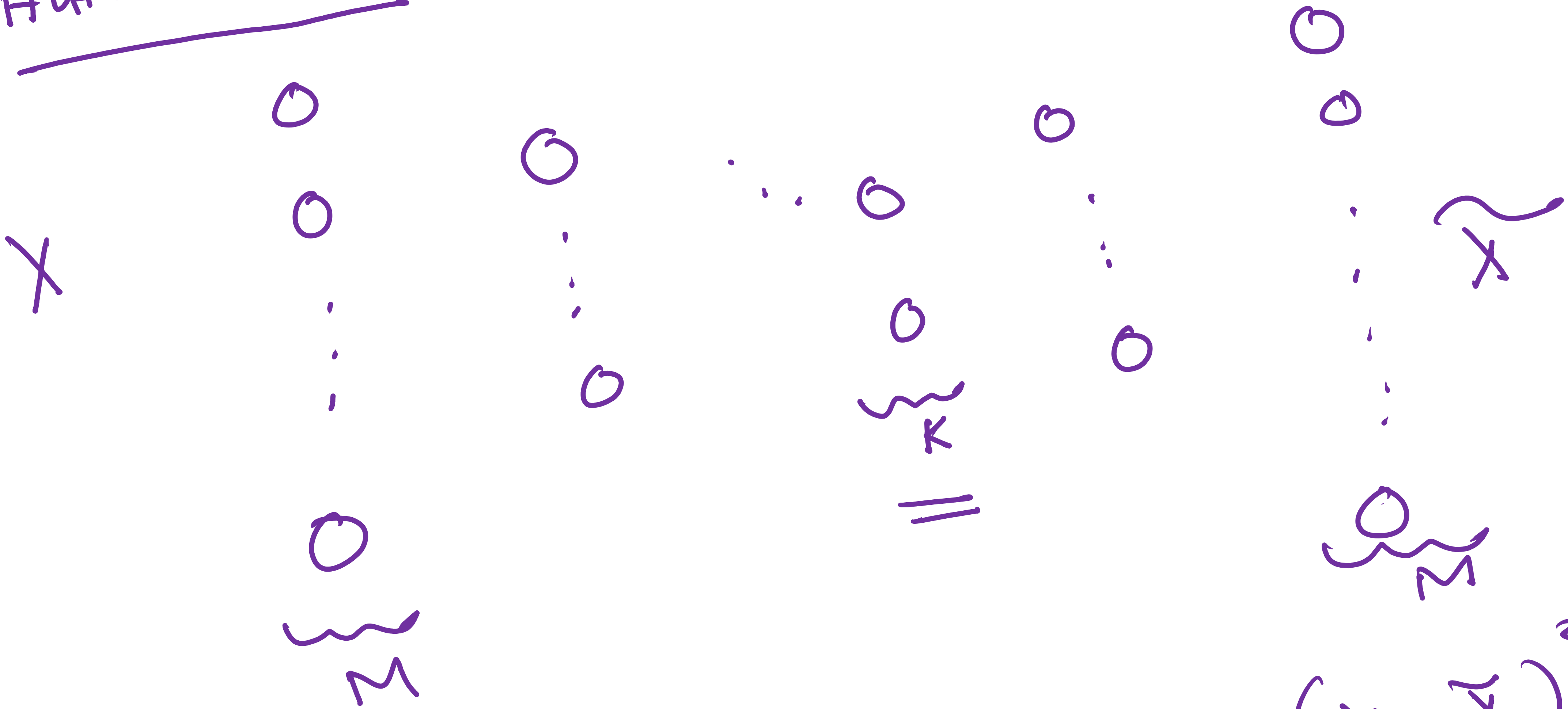
$$\underbrace{Z}_{N \times K} = \underbrace{X}_{N \times m} \underbrace{\phi_K}_{m \times K}$$

$K \leftarrow$ using Eigen value

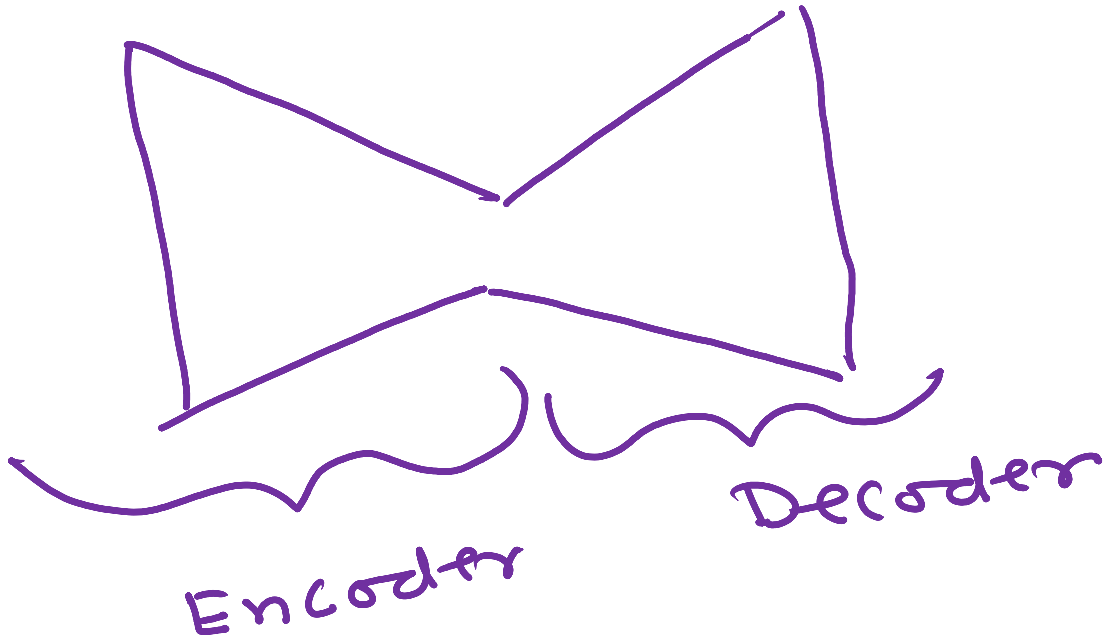
Energy captured

$$= \frac{\lambda_{1:K}^2}{\sum_{j=1}^{M} \lambda_j^2}$$

Autoencoder



$$E = (X - \tilde{X})^2$$


Encoder
Decoder