PCA steps

$$\underline{\underline{X}} = [\underline{x}_1, \underline{x}_2, \cdots, \underline{x}_N] \quad , \quad \underline{x}_i \in \mathbb{R}^D \quad , D = 784$$

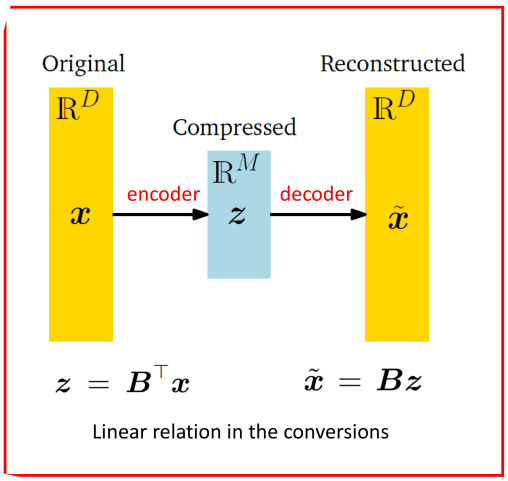
$$\underline{\underline{S}} = \frac{1}{N} \sum_{n=1}^{N} \underline{x}_n \underline{x}_n^T$$

$$\underline{x}_n \; (D \times 1) \cdot \underline{x}_n^T \; (1 \times D) \Rightarrow (D \times D)$$

$$\begin{bmatrix} | & | & & | \\ \underline{x}_1 & \underline{x}_2 & \cdots & \underline{x}_N \\ | & | & & | \end{bmatrix}_{D \times N} \begin{bmatrix} — \; \underline{x}_1 \; — \\ — \; \underline{x}_2 \; — \\ \vdots \\ — \; \underline{x}_N \; — \end{bmatrix}_{N \times D} = \begin{bmatrix} & \\ & \end{bmatrix}_{D \times D}$$

$$\Rightarrow \begin{bmatrix} | \\ \underline{x}_1 \\ | \end{bmatrix}_{D \times 1} [— \; \underline{x}_1 \; —]_{1 \times D} + \begin{bmatrix} | \\ \underline{x}_2 \\ | \end{bmatrix} [— \; \underline{x}_2 \; —] + \cdots \begin{bmatrix} | \\ \underline{x}_N \\ | \end{bmatrix} [— \; \underline{x}_N \; —]$$

$$\Rightarrow \begin{bmatrix} \cdots \\ \cdots \\ \cdots \end{bmatrix}_{D \times D} + \begin{bmatrix} \cdots \\ \cdots \\ \cdots \end{bmatrix}_{D \times D} + \cdots \begin{bmatrix} \cdots \\ \cdots \\ \cdots \end{bmatrix}_{D \times D} \Rightarrow \sum_{n=1}^{N} \underline{x}_n \underline{x}_n^T$$

$\underline{x}$ → Normalize → 

→ De-normalize → $\tilde{\underline{x}}$

$\tilde{\underline{x}} \cdot \sqrt{\sigma(\underline{x})} + \overline{\underline{x}}$

$\underline{x} - \overline{\underline{x}}$ / $\sqrt{\sigma(\underline{x})}$

→ $\left[— \lambda_i — \right], \left[— \underline{v}_i — \right]$ = eigendecomposition ($\underline{\underline{\Sigma}}$)

→ rearrange eigenvalues in descending order

$[\lambda_1, \lambda_2, \dots \lambda_D]$ $\quad \lambda_1 > \lambda_2 > \cdots > \lambda_D$

→ rearrange eigenvectors in the same order

$[\underline{v}_1, \underline{v}_2, \cdots, \underline{v}_D]$, $\underline{v}_D \in \mathbb{R}^D$

→ Collect first M eigenvectors. (M = no. of Principal Components = reduced dimension)

$\underline{\underline{B}} = [\underline{v}_1, \underline{v}_2, \cdots, \underline{v}_M]$, $M \leq D$

$\underline{x} \rightarrow \underline{\underline{B}}^T \underline{x} \rightarrow \underline{z} \rightarrow \underline{\underline{B}}\,\underline{z} \rightarrow \tilde{\underline{x}}$

encode decode

$\begin{bmatrix} — \underline{v}_1 — \\ — \underline{v}_2 — \\ \vdots \\ — \underline{v}_M — \end{bmatrix} \begin{bmatrix} | \\ \underline{x} \\ | \end{bmatrix} = \begin{bmatrix} | \\ \underline{z} \\ | \end{bmatrix}$ $M \times D$ $D \times 1$ $M \times 1$

$\underline{\underline{X}} \rightarrow \underline{\underline{B}}^T \underline{\underline{X}} \rightarrow \underline{\underline{Z}} \rightarrow \underline{\underline{B}} \cdot \underline{\underline{Z}} \rightarrow \tilde{\underline{\underline{X}}}$

$\begin{bmatrix} — \underline{v}_1 — \\ — \underline{v}_2 — \\ \vdots \\ — \underline{v}_M — \end{bmatrix} \begin{bmatrix} | & | & & | \\ \underline{x}_1 & \underline{x}_2 & \cdots & \underline{x}_N \\ | & | & & | \end{bmatrix} = \begin{bmatrix} | & | & & | \\ \underline{z}_1 & \underline{z}_2 & \cdots & \underline{z}_N \\ | & | & & | \end{bmatrix}$ $M \times D$ $D \times N$ $M \times N$

$\begin{bmatrix} | & | & & | \\ \underline{v}_1 & \underline{v}_2 & \cdots & \underline{v}_M \\ | & | & & | \end{bmatrix} \begin{bmatrix} | & | & & | \\ \underline{z}_1 & \underline{z}_2 & \cdots & \underline{z}_N \\ | & | & & | \end{bmatrix} = \begin{bmatrix} | & | & & | \\ \tilde{\underline{x}}_1 & \tilde{\underline{x}}_2 & \cdots & \tilde{\underline{x}}_N \\ | & | & & | \end{bmatrix}$ $D \times M$ $M \times N$ $D \times N$

→ Recall linear regression

$h_{\underline{w},b}(\underline{x}) = \underline{w} \cdot \underline{x} + b$, $\underline{w} \in \mathbb{R}^d$, $b \in \mathbb{R}$

↳ Used for regression., ie. to predict values.

How to use it for classification?

Sigmoid function

$$sig(x) = \frac{1}{1+e^{-x}} = \frac{e^x}{1+e^x}$$

If $x \in \mathbb{R}$, $sig(x) \in (0, 1)$

Thus, we can convert a predicted value from regression to a probability value.

→ Logistic Regression

$h_{\underline{w},b}(\underline{x}) = sig(\underline{w} \cdot \underline{x} + b)$, $\underline{w} \in \mathbb{R}^d$, $b \in \mathbb{R}$

$h(\underline{x}) = \mathbb{P}$(object with features $\underline{x} \in \mathbb{R}^d$ has label +1)

$\Rightarrow \mathbb{P}$(object with features $\underline{x} \in \mathbb{R}^d$ has label 0) $= 1 - h(\underline{x})$

object with $\underline{x} \in \mathbb{R}^d$ (Our object can only belong to category 1 or 0)

label = +1 : $h(\underline{x})$

label = 0 : $1 - h(\underline{x})$

→ Likelihood

$\mathbb{P}(Y = 1 \mid X = \underline{x}) = h_{\underline{w},b}(\underline{x})$

$\Rightarrow \mathbb{P}(Y = 0 \mid X = \underline{x}) = 1 - h_{\underline{w},b}(\underline{x})$

$\mathbb{P}(Y = y_i \mid X = \underline{x})$

$= \mathbb{P}(Y=1 \mid X=\underline{x}) \cdot \mathbb{P}(Y=0 \mid X=\underline{x})$

$= h_{\underline{w},b}(\underline{x}) \cdot (1 - h_{\underline{w},b}(\underline{x}))$

Example: Coin tossing → $y \in \{H, T\}$, $\mathbb{P}(Y = H \mid X = \text{fair coin}) = 1/2$, $\mathbb{P}(Y = T \mid X = \text{fair coin}) = 1/2$

for two toss, $\mathbb{P}$(getting H atleast once)

$= \mathbb{P}(H,H) + \mathbb{P}(H,T) + \mathbb{P}(T,H)$

$= \frac{1}{2} \cdot \frac{1}{2} + \frac{1}{2} \cdot \frac{1}{2} + \frac{1}{2} \cdot \frac{1}{2} = \frac{3}{4}$

→ Maximum Likelihood Estimation

Given: sample with m no. of datapairs $(\underline{x}_i, y_i) \in \mathbb{R}^d \times \{1, 0\}$

Goal: Find the weight vector $\underline{w}_s$, and bias b_s that best explains the data

OR: Find the pair $(\underline{w}_s, b_s)$ which yields the longest likelihood for observing all the data pairs $(\underline{x}_i, y_i)$

Assuming that our data pairs are i.i.d, we can say,

$$\mathcal{L}_{\underline{w},b}(s) = \mathbb{P}_{\underline{w},b}\left(Y_1 = y_1, \dots, Y_m = y_m \mid X_1 = \underline{x}_1, \dots, X_m = \underline{x}_m\right)$$

Probability of finding the correct labels $y_1 \dots y_m$ given all the samples $\underline{x}_1, \dots, \underline{x}_m$

Since our samples are i.i.d, and $P(A \cap B) = P(A) \cdot P(B)$ for independent A and B,

$$\mathcal{L}_{\underline{w},b}(s) = \prod_{i=1}^{m} \mathbb{P}_{\underline{w},b}\left(Y = y_i \mid X = \underline{x}_i\right)$$

$$\mathcal{L} = \prod_{i=1}^{m} h_{\underline{w},b}(\underline{x}_i)\left(1 - h_{\underline{w},b}(\underline{x}_i)\right) \xrightarrow{!} \max$$

$$h_{\underline{w},b}(\underline{x}) = \frac{1}{1+\exp(-\underline{w}\cdot\underline{x}+b)} \qquad \text{let } \underline{w}\cdot\underline{x}+b = t;$$

$$h(\underline{x}) = \frac{1}{1+e^{-t}}, \quad 1-h(\underline{x}) = 1-\frac{1}{1+e^{-t}} = \frac{e^{-t}}{1+e^{-t}} = \frac{1}{1+e^{t}}$$

$$\therefore 1-h_{\underline{w},b}(\underline{x}) = \frac{1}{1+\exp(\underline{w}\cdot\underline{x}+b)}$$

$\rightarrow$ **Log-likelihood**

$$L = \prod_{i=1}^{n} h_{\underline{w},b}(\underline{x}_i)\left(1-h_{\underline{w},b}(\underline{x}_i)\right)$$

$$\ln(L) = \sum_{i=1}^{m}\left[y_i \ln\left(h_{\underline{w},b}(\underline{x}_i)\right) + (1-y_i)\left(1-\ln\left(h_{\underline{w},b}(\underline{x}_i)\right)\right)\right]$$

$$\stackrel{!}{=} \max$$

Aim: $$(\underline{w}, b) = \underset{\underline{w},b}{\operatorname{argmax}}\left(\ln(L)\right)$$

OR equivalently:

$$(\underline{w}, b) = \underset{\underline{w},b}{\operatorname{argmin}}\left(-\ln(L)\right) \quad !!!$$

Many optimization algorithms to minimize

Thus, we can find $\underline{w}_s, b_s$ as follows:

$$\underline{w}_s, b_s = \underset{\underline{w},b}{\operatorname{argmin}}\left[\underbrace{\sum_{i=1}^{m} -y_i \ln\left(h_{\underline{w},b}(\underline{x}_i)\right) - (1-y_i)\left(1-\ln\left(h_{\underline{w},b}(\underline{x}_i)\right)\right)}_{L(\underline{w},b)}\right]$$

$$\frac{\partial}{\partial \underline{w}} \mathcal{L}(\underline{w}, b) = \sum_{i=1}^{m} \left(h_{\underline{w},b}(\underline{x}_i) - y_i \right) \underline{x}_i$$

$$\frac{\partial}{\partial b} \mathcal{L}(\underline{w}, b) = \sum_{i=1}^{m} \left(h_{\underline{w},b}(\underline{x}_i) - y_i \right) \cdot 1$$

Update rule by gradient descent:

$$\underline{w}^{(k+1)} = \underline{w}^{(k)} - \alpha \frac{\partial}{\partial \underline{w}} \mathcal{L}(\underline{w}, b)$$

$$b^{(k+1)} = b^{(k)} - \alpha \frac{\partial}{\partial b} \mathcal{L}(\underline{w}, b)$$