Tarea
usando PCA determinar las entradas $x \in \mathcal{X}$, $y \in \mathcal{Y}$, la salida $\hat{y} \in ?$, el loss $\frac{1}{\ }\{\cdot,\cdot\}$, hay que optimizarlo y programarlo como red neuronal

PCA es un modelo generativo, ej: permite limpiar ruido de imagenes creando imagenes nuevas en un subespacio de menos dimensiones y compararlo con las imagenes originales

la entrada es la imagen un x que viven cualquier espacio $\mathcal{X}$, y la salida es la imagen proyectada y que vive en $\mathcal{Y}$, $\mathcal{Y} \subseteq \mathcal{X}$, ya que es un subespacio

la funcion Loss se da comparando la imagen original con la generada $\|x - f(y)\|$, $f(y)$ regresa la imagen generada en el subespacio al espacio original $\mathcal{X}$.

y como y se genera desde x, entonces $y = g(x)$ en un espacio $\mathcal{Y}$ de dimension $m \leq$ a P, la dimension de $\mathcal{X}$

entonces $\text{Loss} = \mathcal{E}_{x, \hat{x}}\{\|x - f(g(x))\|_2\} = \mathcal{E}_{x,\hat{x}}\{\|x - \hat{x}\|_2\}$

con $\hat{x} = f(g(x))$, si se aproxima usando la media muestral $= \frac{1}{N}\sum_{N}\|x_n - \hat{x}_n\|_2^2$. El objetivo es encontrar los parametros Θ, en $\|x_n - x_n\Theta\Theta^T\|_2^2$, que son los pesos que permiten generar las imagenes (Θ) y con Θ^T regresamos al número original de dimensiones para comparar x_n con el $\hat{x}_n$ generado. Se hace necesario entonces calcular $\|x_n - \hat{x}\|_2^2$ y minimizarlo en Θ

$$\min_{\Theta} \mathcal{E}_{x,\hat{x}}\left\{ \| x_n - x_n\Theta\Theta^T \|_2^2 \right\} = \min_{\Theta} \mathcal{E}_{x,\hat{x}}\left\{ \langle x_n - x_n\Theta\Theta^T, x_n - x_n\Theta\Theta^T \rangle \right\}$$

$$= \min_{\Theta} \mathcal{E}_{x,\hat{x}}\left\{ x_n x_n^T - x_n(x_n\Theta\Theta^T)^T - x_n(x_n\Theta\Theta^T)^T + x_n\Theta\Theta^T(x_n\Theta\Theta^T)^T \right\}$$

$$= \min_{\Theta} \mathcal{E}_{x,\hat{x}}\left\{ x_n x_n^T - 2x_n(x_n\Theta\Theta^T)^T + x_n\Theta\Theta^T(x_n\Theta\Theta^T)^T \right\}$$

Si x_n viene del espacio $\mathcal{X}$ de dimension p $\mathcal{X} \in \mathbb{R}^p$, x_n sea un vector fila tal que $x_n \in \mathbb{R}^{1\times p}$ y w un vector columna tal que $\Theta \in \mathbb{R}_{p\times 1}$

$$= \min_{\Theta} \mathcal{E}_{x,\hat{x}}\left\{ x_n x_n^T - 2x_n\Theta\Theta^T x_n^T + x_n\Theta\boxed{\Theta^T\Theta}\Theta^T x_n^T \right\}$$

Considerando que vamos a regresar al mismo numero de dimensiones inicial

$\Theta^T\Theta = I$, para que sea posible comparar x_n con $\hat{x}_n$

$$= \min_{\Theta} \mathcal{E}_{x,\hat{x}}\left\{ x_n x_n^T - 2x_n\Theta\Theta^T x_n^T + x_n\Theta\Theta^T x_n^T \right\}$$

$$= \min_{\Theta} \mathcal{E}_{x,\hat{x}}\left\{ x_n x_n^T - x_n\Theta\Theta^T x_n^T \right\}$$

Como la minimizacion es en Θ, el termino $x_n x_n^T$ no aporta nada

$$= \min_{\Theta} \mathcal{E}_{x,\hat{x}}\left\{ -x_n\Theta\Theta^T x_n^T \right\} = \min_{w} -\mathcal{E}_{x,\hat{x}}\left\{ x_n\Theta\Theta^T x_n^T \right\}$$

Considerando la restriccion $\Theta^T\Theta = I$, y definimos una variable $z_n = x_n\Theta$, para realizar una sustitucion es importante porque como ya notamos $x_n \in \mathbb{R}^{1\times p}$ y $\Theta \in \mathbb{R}^{p\times 1}$, dando como resultado que $z_n \in \mathbb{R}$

$= \min\limits_{\theta} - \mathcal{E}_{x,\hat{x}} \{ z_n z_n^T \}$ Como z_n es real $= \min\limits_{\theta} - \mathcal{E}_{x,\hat{x}} \{ z_n^T z_n \}$

$= \min\limits_{\theta} - \mathcal{E}_{x,\hat{x}} \{ \theta^T x_n^T x_n \theta \}$ Como el operador esperanza es en

x y $\hat{x}$. $= \min\limits_{\theta} - \theta^T \mathcal{E}_{x,\hat{x}} \{ x_n^T x_n \} \theta$ y $\mathcal{E}_{x,\hat{x}} \{ x_n^T x_n \} = R_{xx} \in \mathbb{R}^{p \times p}$

$= \min\limits_{\theta} - \theta^T \boxed{\Sigma_x} \theta = \max\limits_{\theta} \theta^T \boxed{\Sigma_x} \theta$ Considerando $\theta^T \theta = 1$

No recuerdo el nombre de esta expresion, creo que es la matriz de covarianza pero no es un concepto con el que este familiarizado

a partir de aqui es necesario proponer un lagrangiano $L(\theta, \lambda) = \theta^T \Sigma x \theta - \lambda(\theta^T \theta - 1)$ y calcular su derivada igualada a cero para minimizar θ

$$\frac{\partial L}{\partial \theta} = 2 \Sigma_x \theta - 2 \lambda \theta^T \theta = 0 \rightarrow \Sigma_x \theta = \lambda \theta$$

que es el problema de valores y vectores propios

a partir de aqui seguiria implementar todo en una red neuronal y programarlo, pero tengo muy poca experiencia y demasiadas dudas sobre el uso de Keras y como crear la red en python.

¿Como asegurarme de que $\theta^T \theta = I$? ¿como se cuantas neuronas ponerle a cada layer? ¿cual debe ser la funcion de activacion?