

PCA

La base de datos FASHION-MNIST posee imágenes de diferentes prendas de vestir, pertenecientes a 10 clases diferentes. Se desea entrenar un algoritmo de PCA.

(a) Cargar la base de datos utilizando `tensorflow.keras.datasets.fashion_mnist.load_data`. Utilizando `imshow` (matplotlib) represente 10 muestras del conjunto de testeo elegidas al azar. ¿Qué tipo de prenda representa cada categoría?

(b) Implementar el algoritmo de PCA utilizando `linalg.eig` (numpy). El código debe estar estructurado de la siguiente manera (implementar todos los métodos mencionados):

```
class PCA:

    # Inicializar atributos y declarar hiperparámetros. No aclarar aún la dimensión
    # del espacio latente
    def __init__(self,...

    # Etapa de entrenamiento. Se entrenan todos los autovectores, sin definir la
    # dimensión latente
    def fit(self,X):

    # Transformar del espacio original al espacio latente de dimensión k.
    def transform(self,X,k):

    # Transformar del espacio latente al espacio original
    def inverse_transform(self,U):

    # fit + transform
    def fit_transform(self,X,k):

    # transform + inverse_transform. Debe incluir la posibilidad de aprovechar
    # reconstrucciones ya efectuadas con menos cantidad de componentes.
    def transform_reconstruction(self,X,k,use_previous_results = False):

    # fit + transform_reconstruction. No es necesario utilizar resultados
    # anteriores pues entrenamiento hay uno solo.
    def fit_reconstruction(self,X,k):
```

(c) Calcular y graficar el porcentaje de energía en función del número de componentes principales.

(d) Graficar el error cuadrático medio de testeo en función del número de componentes principales.

🔗: Mucha atención a no repetir cálculos. Es decir, aprovechar la reconstrucción con k componentes principales para el cálculo de $k + 1$ componentes. Es importante también que el ECM no se vea afectado por normalizaciones.

(e) Graficar 10 imágenes reconstruidas de testeo (elegidas al azar) utilizando 1, 100 y 784 componentes principales.

(f) Se desea evaluar el desempeño del algoritmo como detector de anomalías. Para ello, construir una base de datos combinando el conjunto de datos de testeo con el conjunto de datos de testeo de la base de datos MNIST (base de datos de imágenes dígitos manuscritos del mismo tamaño). Dicha base de datos puede ser cargada con `tensorflow.keras.datasets.mnist.load_data`.

(g) Diseñar un detector de anomalías comparando el error cuadrático contra un umbral. Implementar y graficar la curva ROC para 1, 100 y 784 componentes principales.