

UNIVERSIDAD DE CONCEPCIÓN

FACULTAD DE INGENIERÍA

DEPARTAMENTO DE INGENIERÍA ELÉCTRICA



Tarea n°4: Sistemas Adaptativos – 543820.

Jocelyn Matus Ancavil.

Profesor, Daniel Sbarbaro.

Concepción, lunes 6 de diciembre 2021

Problema 1

Mediante la regla de Hebb (versión Oja) comprima la imagen cameraman.tif disponible en Matlab (esta imagen se lee desde Matlab con $A = \text{IMREAD}('cameraman.tif')$). Para realizar esta compresión considere la figura compuesta por bloques de 8×8 píxeles. Es decir el conjunto de entrenamiento contendrá 1024 vectores de dimensión 16

. a. Cuantifique la diferencia entre la imagen original y la reconstituida para diferentes números de componentes principales usados.

La imagen que se quiere comprimir es la siguiente:



Figura N°1, . Imagen original.

Se divide esta imagen en bloques de 8×8 píxeles para comprimir esta imagen. Al aplicar la regla de Hebb (versión Oja) se obtiene una matriz de peso de 8×8 , la cual se aplica a cada bloque de 8×8 que se encuentra en la imagen.

Esta matriz 8×8 obtenida es una matriz que contiene los componentes principales de la imagen que está dividida en bloques de 8×8 .

Al realizar la construcción de la imagen con distintos componentes principales, de manera que se agregan mas componentes principales a la reconstrucción, se obtiene lo siguiente:



Figura N°2, Imagen reconstruida con numero de componentes principales creciente.

Estas imágenes se ordenan de la siguiente manera: en la primera imagen, de la esquina arriba izquierda, se utiliza solo un componente principal para la reconstrucción de la imagen.

La segunda imagen, se utiliza dos componentes principales para la reconstrucción de la imagen, y así sucesivamente, con la imagen que se encuentra en la esquina izquierda abajo con 5 componentes principales, hasta utilizar los 8 componentes principales, que sería la imagen de la esquina de abajo derecha.

Se puede ver la diferencia de la imagen original con la reconstrucción con todos los componentes principales en la siguiente imagen:



Figura N°3, Comparación entre imagen comprimida e imagen original.

Se puede ver que la compresión dio resultado, y que resulto exitoso el calculo de los vectores principales.

Ahora, para poder obtener la diferencia entre la imagen real y cada una de las reconstrucciones, se analiza la diferencia entre cada componente de las imágenes, y se calcula su cambio relativo, y el promedio entre todos estos cambios relativos.

Al realizar esto, se obtuvo las siguientes diferencias:

- 1) 1 componente principal: 587%
- 2) 2 componentes principales: 993%
- 3) 3 componentes principales: 515%
- 4) 4 componentes principales: 417%
- 5) 5 componentes principales: 316%
- 6) 6 componentes principales: 237%
- 7) 7 componentes principales: 235%
- 8) 8 componentes principales: 0.913%

La diferencia se puede explicar por que el ultimo componente principal que se utilizo es que el que entrega el vector con más información disponible de la reconstrucción, mientras que los otros casi no entregan algún tipo de apoyo en esta reconstrucción.

b. Qué características tienen los vectores principales?

Los vectores principales tienen la característica que su norma es 1, o muy cercana a esta. Esto se explica por qué los componentes principales indican la dirección de la línea donde los datos están mejor representados.

c. Comprima la imagen del archivo galin.mat (el cual se carga con el comando load) usando los vectores principales encontrados con la imagen anterior. Comente los resultados.

La imagen que se quiere comprimir es la siguiente:



Figura N°4, Imagen original.

Y, al aplicar la compresión con vectores principales encontrados anteriormente, se obtiene la siguiente imagen:

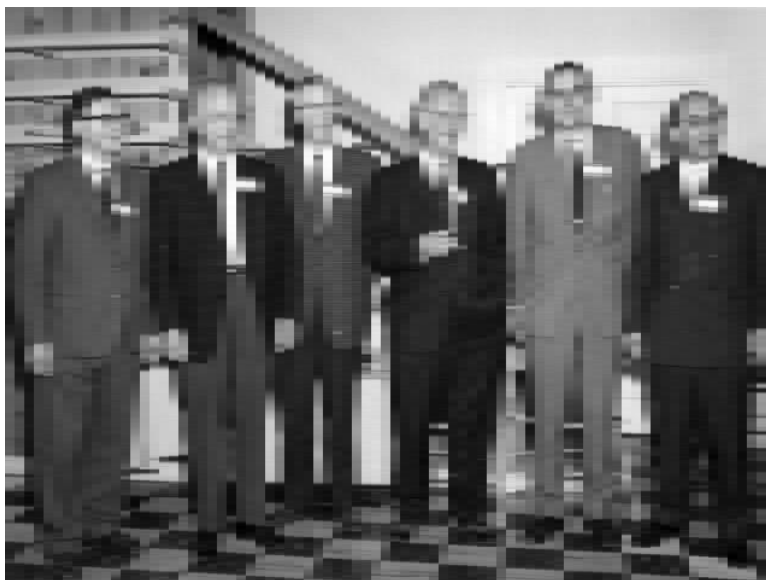


Figura N°5, Imagen reconstruida.

Se puede ver que se realizó la reconstrucción de manera correcta, y, además, al trabajar con más cantidad de píxeles, por la dimensión de la imagen, se puede ver que la imagen reconstruida se ve menos comprimida, que en comparación con la imagen reconstruida de los puntos anteriores.

Problema 2 (2.5)

Considere la base de rostros Olivetti, que puede obtenerse de la siguiente dirección: <http://www.cs.toronto.edu/~roweis/data.html> y desarrolle un sistema de clasificación de rostros que use PCA para reducir la dimensionalidad de las imágenes y realice la clasificación usando la mínima distancia euclidiana en el espacio reducido.

a. Genere una base de datos para clasificación calculando un vector promedio entre las fotos de cada individuo del conjunto de entrenamiento.

Al promediar los píxeles de las caras, de manera que los píxeles que se encuentran en cierta posición son promediados con el píxel de las otras caras que se encuentra en la misma posición, se obtiene la siguiente imagen:

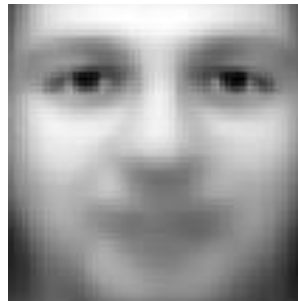


Figura N°6, Cara promediada.

b. Elija en forma aleatoria la mitad de los rostros de cada persona para entrenamiento y la otra mitad para evaluar su algoritmo. Obtenga los valores de los vectores principales mediante la regla de Hebb (versión Oja). Comente los resultados.

Se eligió de manera aleatoria los conjuntos de entrenamiento y validación, y con el conjunto de entrenamiento se calculó la cara promedio, obteniendo lo siguiente:



Figura N°7, Cara promedio de set de entrenamiento.

Al utilizar la regla de Hebb (versión Oja), se obtiene la matriz de pesos W , la cual contiene los vectores principales de la cara promediada mostrada anteriormente.

Para verificar que es correcto, se reconstruyo la imagen con esta matriz y se obtuvo lo siguiente:



Figura N°8, Imagen reconstruida.

Lo cual se verifica que la matriz de pesos w entregada es la correcta.

La norma de estos vectores principales es de 1, lo cual es una característica de los vectores principales tener una norma de 1, y estos vectores, al ser la imagen 64×64 , tienen un largo de 64.

c. Grafique la precisión de la clasificación versus cantidad de componentes utilizados en el reconocimiento en el espacio reducido. Evalúe la sensibilidad de sus resultados a la elección de los rostros en el conjunto de entrenamiento

Al evaluar la precisión vs el número de componentes, se obtiene lo siguiente:

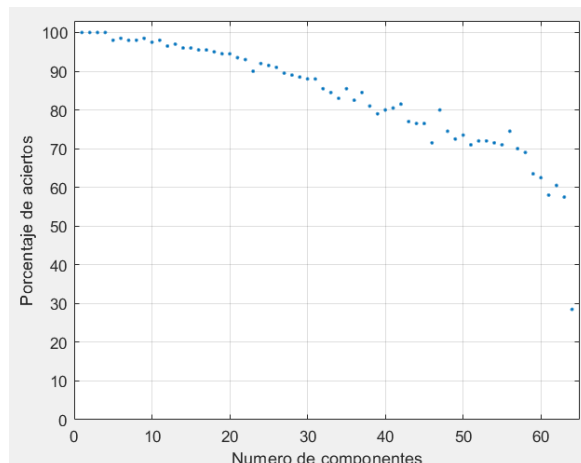


Figura N°9, Numero de componentes vs precisión, 200 caras de entrenamiento.

La explicación de por que se reduce la precisión cuando aumenta la cantidad de componentes es que los primeros componentes, los principales, en este caso, entregan la mayor información que se necesita para reconstruir las caras, mientras que, si se sigue aumentando de componentes, la información que entregan esos componentes no es tan alta, y hasta puede afectar el rendimiento de la reconstrucción.

En el grafico anterior se considero todo el set de entrenamiento para el calculo de la matriz de pesos, el cual son 200 caras. En los siguientes gráficos se presenta la precisión vs numero de componentes con distintas cantidades de caras de entrenamiento:

150 caras de entrenamiento:

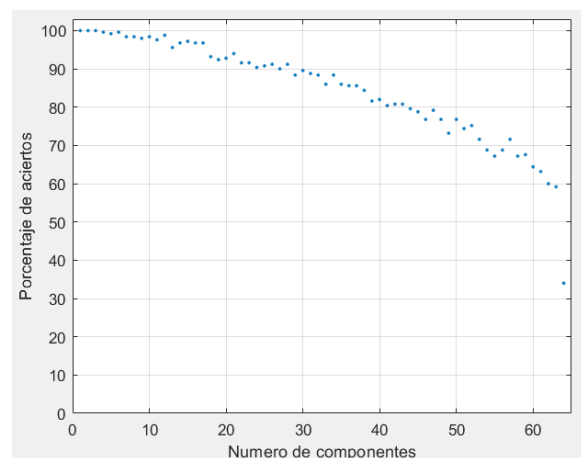


Figura N°10, Numero de componentes vs precisión, 150 caras de entrenamiento.

100 caras de entrenamiento:

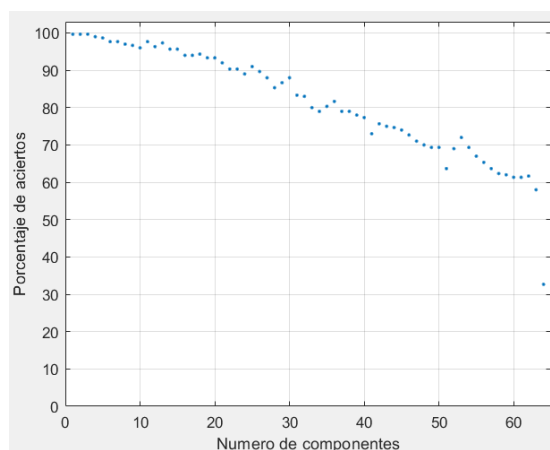


Figura N°11, Numero de componentes vs precisión, 100 caras de entrenamiento.

50 caras de entrenamiento:

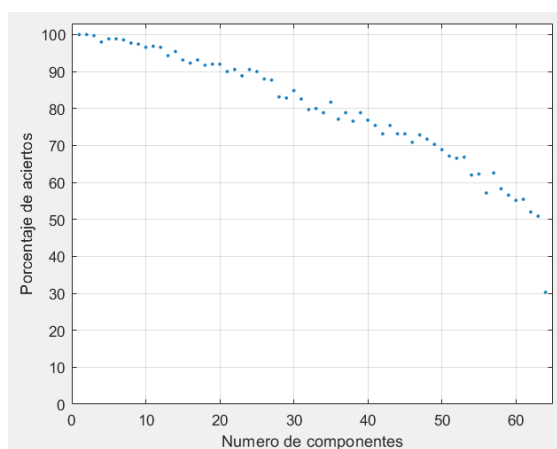


Figura N°12, Numero de componentes vs precisión, 50 caras de entrenamiento.

25 caras de entrenamiento:

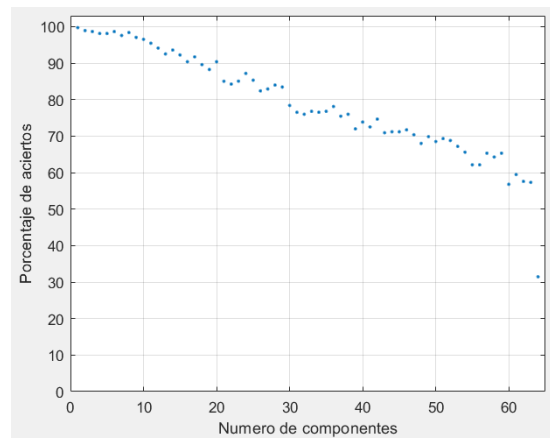


Figura N°13, Numero de componentes vs precisión, 25 caras de entrenamiento.

12 caras de entrenamiento:

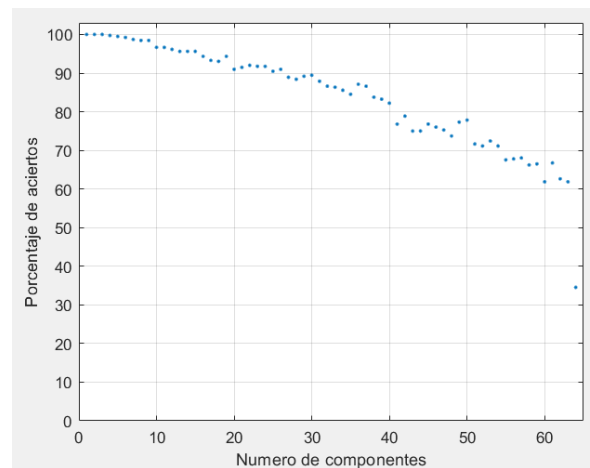


Figura N°14, Numero de componentes vs precisión, 12 caras de entrenamiento.

7 caras de entrenamiento:

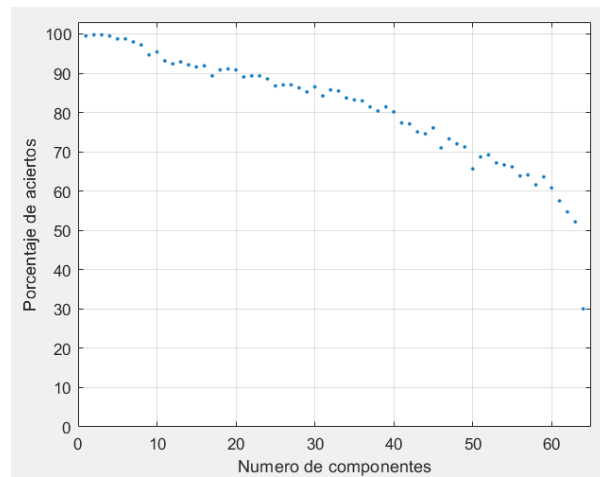


Figura N°15, Numero de componentes vs precisión, 7 caras de entrenamiento.

4 caras de entrenamiento:

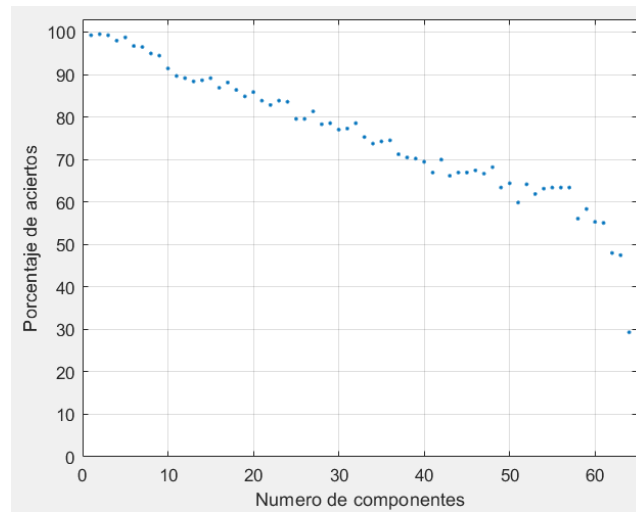


Figura N°16, Numero de componentes vs precisión, 4 caras de entrenamiento.

2 caras de entrenamiento:

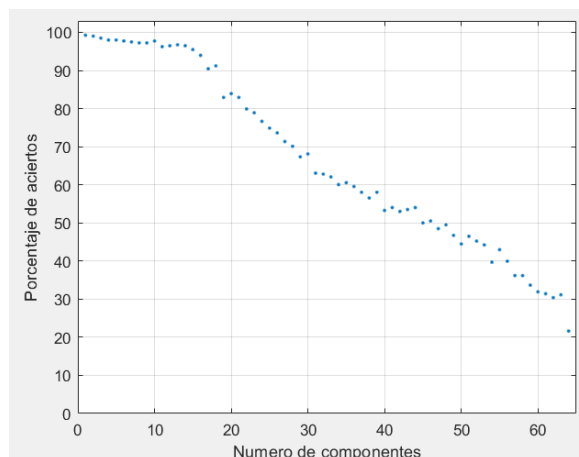


Figura N°17, Numero de componentes vs precisión, 2 caras de entrenamiento.

1 caras de entrenamiento:

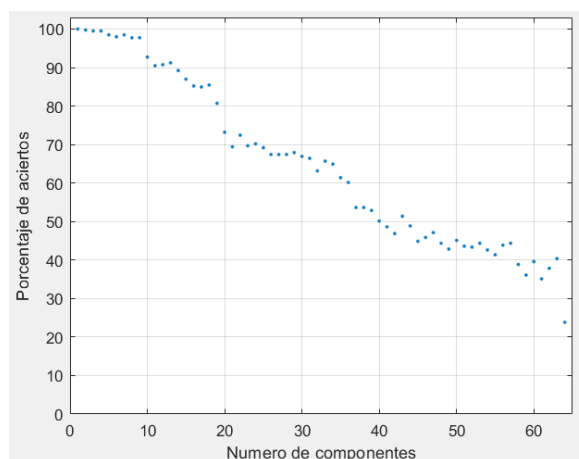


Figura N°18, Numero de componentes vs precisión, 1 cara de entrenamiento.

La sensibilidad de los resultados dependiendo de la elección de entrenamiento, para valores pequeños de componentes, no afecta tanto en la precisión, pero si afecta cuando se ocupan mas valores de componentes. Cuando la cantidad de caras de entrenamiento es alta, no afecta tanto, pero cuando va bajando, afecta más la precisión.

Esto se puede explicar con que las caras humanas, en general, son parecidas entre si, por como se componen las caras de manera biológica, por lo cual, el sistema puede ser entrenado con pocas caras.

. d. Realice un análisis de valores propios para calcular directamente los coeficientes de la red PCA en modo batch y compare los resultados con los obtenidos en b. Comente!

Al realizar un análisis de los valores propios y calcular los vectores principales de la imagen, se puede ver que son distintos en comparación a los obtenidos en el punto b). El calculo de esta diferencia es de 78.34%. Esto se puede explicar con que el algoritmo de Hebb (versión Oja) solamente converge a un mínimo local, el cual no necesariamente es el mínimo global, por lo que puede entregar resultados que están alejados del valor real

Problema 3 (1)

Estudie el artículo titulado "Self-Organizing Feature Maps with Self-Adjusting Learning Parameters" publicado en la IEEE Transaction on ANN Vol 9, No. 6, 1998.

a. Resuma y comente los aspectos principales propuestos en este artículo.

La idea principal del paper es que, según el método actual de algoritmo de aprendizaje de Kohonen, es cual es considerado una regla de aprendizaje para crear un self-organizing feature map (SOM), el cual es una función de mapeo el cual proyecta un set de datos hacia una red neuronal, realizando una reducción de dimensión de estos datos.

El algoritmo de Kohonen converge a una función de mapeo que mantiene las relaciones de vecindad que tienen los datos de entrada.

El problema que tiene este algoritmo es que se tiene que fijar dos parámetros para que este funcione correctamente y llegue a una solución correcta. La determinación de estos parámetros se realiza de manera práctica, sin tener una regla o proceso para obtenerlos, generalmente a través de prueba y error.

La solución que entrega este paper es la automatización de la obtención de estos parámetros, y la realización del algoritmo de manera correcta. Para esto, se divide en dos pasos la solución:

- 1) El algoritmo de Kohonen se puede transformar de manera equivalente a ecuaciones del filtro de Kalman. Con estas ecuaciones, se calculan los pesos de la red del algoritmo sin que se tenga que calcular los 2 parámetros.
- 2) Para que se pueda ser llevado a una red neuronal, para poder analizar las relaciones de vecindad que se tienen que preservar, se utiliza el filtro de Kalman extendido.

Al utilizar estos dos pasos, se puede utilizar este nuevo algoritmo y asegurar que funcione correctamente, sin tener que calcular los dos parámetros que se necesitaban para el algoritmo de Kohonen.

Una característica de esta nueva solución es que converge para todo tipo de datos que son lineales. Lamentablemente, no se puede asegurar lo mismo para datos que no tienen una forma lineal, pero, en el paper, se pudo constatar, de manera práctica, que si el número de vecinos elegido para el algoritmo es bajo, este puede converger de manera correcta y adaptar correctamente datos que no son lineales.

b. Desarrolle un pequeño ejemplo ilustrando lo publicado.
