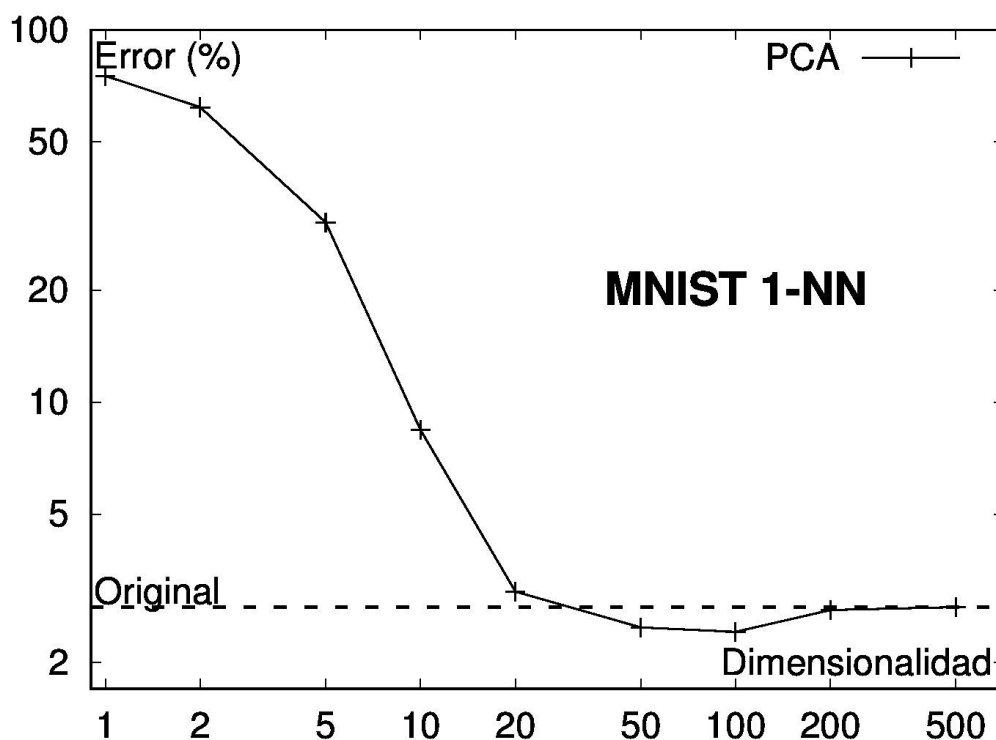


Memoria Entrega 1 Proyecto PER - José Miguel Acosta Triana



Esta es la gráfica que he obtenido que relaciona el número de dimensiones proyectadas con el porcentaje de error al clasificar del conjunto de entrenamiento de MNIST a través de PCA y luego aplicando 1-NN para clasificar.

Se puede observar que cuando proyectamos a muy pocas dimensiones (entre 1 y 20 aproximadamente), no se puede discriminar correctamente entre las clases, ya que muchos datos serán clasificados en una clase incorrecta debido a que faltan dimensiones con datos descriptivos que, de estar, cambiarían la clasificación.

También podemos ver que, a partir de 200 dimensiones, el error se aproxima bastante al error con las dimensiones originales de MNIST, es decir, sin aplicar PCA, con 500 dimensiones, ya tenemos el mismo error que el original (2,817%)

Tras haber anteriormente observado cómo varía el error según la cantidad de dimensiones a las que hemos proyectado, debemos encontrar un número de dimensiones que consiga minimizar el error al clasificar, en la tabla de abajo están los valores más óptimos que encontré, al final decidí coger 89 dimensiones. La tabla de abajo representa los valores de error tomando únicamente el conjunto de entrenamiento.

Dimensiones	% Error
784 (Original)	2,817
86	2,400
87	2,383
88	2,383
89	2,367
90	2,367
91	2,383
92	2,400

Ahora usaremos para comprobar el conjunto de test de MNIST. Clasificando con PCA y k=89 dimensiones, obtenemos que el porcentaje de error al clasificar con 1-NN es de **2,69%**, y clasificando con las dimensiones originales sin aplicar PCA, obtenemos un error de **3,09%**. Este error coincide con el error que muestra MNIST en su página web cuando es clasificado con una de las implementaciones de K-NN usando distancia euclídea (L2), que es lo que estamos usando en esta práctica, así que considero que la implementación realizada es correcta.