

MEMORIA (Práctica 1B)

“Reconocimiento de dígitos utilizando PCA”

Ingeniería del conocimiento

Índice de la memoria

1. Introducción

- a. Problema a resolver
- b. Método utilizado

2. Explicación del problema

- a. Conjuntos de datos que usaremos
- b. Pruebas a realizar

3. Resultados

- a. Explicación de los resultados
- b. Conclusiones (Preguntas de la Sección 5)

4. Conclusión final

Introducción

El objetivo de esta práctica es reconocer un dígito a través de una imagen. Para ello utilizaremos el Análisis de las Componentes Principales (PCA) ya que transforma los datos originales en unos nuevos que muestran mejor las características del problema y facilita la clasificación.

En este método un clasificador realizará un proceso de aprendizaje que permita obtener una función a partir de los ejemplos dados. El clasificador propuesto creará prototipos de cada clase cogiendo todas las imágenes y calculará la media para cada variable, obteniendo una nueva imagen con las medias de todas las de esa clase.

Posteriormente se calcularán distancias que hay desde el ejemplo hasta cada uno de los prototipos y la respuesta será al prototipo que más se acerque.

Explicación del problema

Para llevar a cabo la práctica hemos utilizado el dataset MNIST (Tabla 1), un conjunto de números manuscritos en forma de imagen con tamaño 28x28 píxeles. Cuenta con 10 clases (números del 0 al 9), 60000 imágenes de entrenamiento (Train) y 10000 para comprobar el rendimiento (Test). Al conjunto inicial de datos lo llamaremos XTrain, la salida de cada ejemplo lo llamaremos YTrain y sus respectivos test serán muestras de los conjuntos mencionados anteriormente.



Tabla 1. Ejemplo de distintas imágenes contenidas en MNIST.

Se realizarán 6 pruebas.

En primer lugar, compararemos las clases 0 y 5 y el resultado esperado es alto dado a la poca similitud entre ambas clases.

Las siguientes pruebas serán la comparación entre números similares, como es el caso del 1 y el 7, el 2 y el 7 y el 4 y el 9. En este caso se desea evaluar la efectividad de los prototipos en clases conflictivas, donde se espera un porcentaje más bajo de acierto.

Seguidamente, compararemos todas las clases del dataset EMNIST y evaluaremos como afecta a la precisión de los prototipos el número de ejemplos de test.

Probaremos con 100 imágenes de test y posteriormente con 200, esperando que cuanto mayor sea la muestra, mayor sea el porcentaje de acierto.

Por último, compararemos usando el dataset EMNIST (Figure 2), las 26 clases de letras mayúsculas y determinaremos el porcentaje de acierto con 1000 imágenes de entrenamiento y 104 de test



Figure 2: Clases del conjunto extendido inicial EMNIST

Resultados

Explicación de resultados

Al ejecutar el programa obtendremos 6 resultados.

El primero es 95 y será el porcentaje de acierto de nuestro clasificador entre las clases 0 y 5, es decir, de 100 veces el clasificador distinguirá correctamente entre la clase 0 y la 5 en 95 y fallará en 5. El resultado es el esperado ya que ambos prototipos no tienen muchas similitudes.

Los tres resultados siguientes son 100, 95 y 75 y muestran el porcentaje de acierto en tres pruebas donde comparamos números parecidos (clases conflictivas). El porcentaje será del 100% cuando comparamos el 1 con el 7. En caso de que la comparación sea entre el 2 y el 7 el porcentaje de acierto será del 95%. Por último al comparar las clases 4 y 9 la precisión bajará y el clasificador será capaz de distinguir correctamente el número 75 veces por cada 100 comparaciones realizadas.

El cuarto y quinto resultado muestran como con un menor número de ejemplos de prueba la precisión baja. Cuando hemos probado las 10 clases del dataset MNIST con 100 imágenes de test, la precisión ha sido de un 65%. Al duplicar el número de imágenes de test el resultado ha sido de un 80%, es decir, concluimos que en el dataset MNIST el número de imágenes de test y la precisión de acierto están directamente relacionadas.

Por último, el sexto resultado muestra el porcentaje de acierto de las 26 clases mayúsculas del dataset EMNIST contando con 1000 imágenes de entrenamiento y 104 de test. Este porcentaje es igual a 48.55, un registro muy bajo que se debe al bajo número de ejemplos de test dados y a la peor creación de prototipos derivada de una falta de imágenes de entrenamiento en comparación con las anteriores pruebas.

Conclusiones

*¿Cómo clasifica el algoritmo realizado la distinción entre las clases “más conflictivas”?
¿Por qué consideras que clasifica bien a pesar de que sean números parecidos?*

El algoritmo realizado, una vez ha creado una nueva base y posteriormente los prototipos de cada una de las clases, compara las clases a dadas con los prototipos creados. En base a los resultados obtenidos, podemos asegurar que el algoritmo clasifica bien ya que sin unos prototipos bien distinguidos la comparación entre clases conflictivas no debería tener el porcentaje de acierto alto, como es el caso. Evaluando los resultados entre las clases más conflictivas, el porcentaje baja en el caso del 4 y el 9 pues es del 75%, un resultado bastante poco preciso y que debería mejorar. Cuando comparamos las demás clases conflictivas confirmamos que el algoritmo clasifica bien, es decir, tiene prototipos bien distinguidos y no baja del 95% de acierto (es el caso de las clases 1,7 y 2,7).

¿Qué ocurre cuándo tiene que clasificar 10 clases? ¿Cambia algo el algoritmo cuando tenemos el doble de datos de entrenamiento?

Cuando intentamos clasificar los 10 números, los resultados obtenidos disminuyen al 65%. Esto se debe a un bajo número de ejemplos de test pues contamos con 10 clases

y tan solo con 100 ejemplos de test. El prototipo tan solo se está creando con 10 imágenes y es por ello por lo que la precisión cae.

Al duplicar los datos de entrenamiento y mantener las clases obtenemos un porcentaje más alto (80%). Por ello podemos afirmar que el número de muestras de test y el porcentaje de acierto está directamente relacionado.

¿Qué ocurre cuando tratamos de clasificar 26 clases con un entrenamiento realizado con 1000 ejemplos? ¿Por qué crees que funciona de esa manera? ¿Has probado a realizar un entrenamiento con más ejemplos (2000, 5000, 10000, etc.? Si lo has probado, comenta qué es lo que ocurre.

Cuando tratamos de clasificar las 26 clases de letras mayúsculas del dataset EMNIST, el resultado obtenido es del 48.55% de aciertos sobre los prototipos creados con las 1000 imágenes de entrenamiento. En un principio parecía que aumentando el número de imágenes de entrenamiento el porcentaje subiría. Por ejemplo, con 2000 imágenes el resultado obtenido es del 49% de aciertos. Sin embargo, al probar con 3000 imágenes más, es decir 5000, la precisión volvió a bajar a 47%, algo que no ocurría con el dataset MNIST. Una explicación puede ser que aunque los prototipos creados sean bien distinguidos, contamos con muchas clases similares que hacen no aumentar mucho el porcentaje. Asimismo, cuando el número de imágenes es de 10000 vuelve a aumentar al 50%, confirmando que no hay mucha diferencia entre pruebas y no afecta casi el tamaño de entrenamiento.

Conclusión

Como hemos observado, el algoritmo PCA crea prototipos bien distinguidos que permiten identificar correctamente la clase a la que pertenecen en un porcentaje muy alto de casos. También hemos probado que el número de ejemplos influye directamente en el porcentaje de acierto de nuestro clasificador, siendo menor en caso de que los ejemplos de test también lo sean. Por todo esto y las demás conclusiones extraídas durante este informe, podemos concluir que el algoritmo PCA funciona bien con en el dataset MNIST pero los resultado son todavía mejorables ya que al comparar algunas clases conflictivas o al hacerlo con varias clases a la vez, los porcentajes no son todo lo altos que cabría esperar.