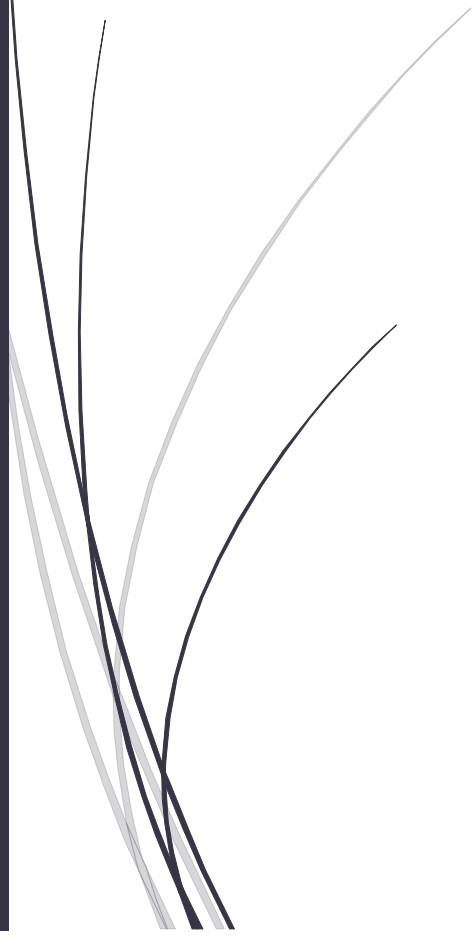
Curso 2021-2022

Proyecto de prácticas

Reconocimiento de dígitos manuscritos: MNIST

Parte opcional



Iñaki Diez Lambies y Manuel Diaz Pastor
PERCEPCIÓN

1 CONTENIDO

Entrega 1

2 EJERCICIO OPCIONAL – ALGORITMO DE WILSON

Hemos escogido implementar este ejercicio siguiendo las recomendaciones dadas en el anexo. Así pues, en el archivo *wilson.py* hemos realizado la implementación de los tres métodos descritos en el boletín.

2.1 MNN

En primer lugar, realizamos *mnn*. Esta función tiene como objetivo que, dada una matriz con los datos de entrenamiento por filas *X*, un vector columna *xl* con las etiquetas de estos y el número *m* de vecinos más cercanos que queremos almacenas; calcular la matriz *V* que almacena por columnas los índices de los *m* vecinos más cercanos de cada prototipo (fila) de *X*.

El funcionamiento de esta es el siguiente:

1. Inicializamos la matriz *V* a una matriz de ceros con la dimensionalidad deseada (*m*, *nº prototipos*).
2. Para cada muestra realizamos el calculo de la distancia L2 adaptando una de las implementaciones dadas en el documento *L2dist.py* a nuestro calculo iterativo.
    a. Después de este cálculo utilizamos la función *argsort* para conseguir los índices de las distancias más pequeñas a la muestra en cuestión.
    b. Guardamos los *m* resultados más cercanos en la matriz *V*. Esto lo hacemos obviando el elemento más cercano (distancia 0, una muestra consigo misma).
3. Después de realizar el cálculo devolvemos *V* como resultado.

En cuanto al cálculo de la distancia, hemos realizado una aproximación iterativa (frente a la directa ya propuesta y más eficiente temporalmente) debido a la incapacidad de almacenar una matriz de tamaño (60000, 60000). Nuestros sistemas no han podido con esa cantidad de datos y es por ello por lo que hemos optado por sacrificar eficiencia temporal a cambio de una eficiencia espacial del algoritmo.

2.2 KNNV

Seguidamente realizamos la implementación del método *knnV*. Este nos permite devolver la clase a la cual pertenece el prototipo *i*. Esto lo hace a partir de la columna con los índices de los prototipos más cercanos *Vi*, el conjunto de índices aún disponibles de Wilson *ind*, las clases de las muestras *xl* y el número de vecinos más cercanos a tener en cuenta *k*.

El funcionamiento de este es el siguiente:

1. Filtramos en *idx* los índices de *Vi* que aún no han sido eliminados.
2. Escogemos los *k* primeros a considerar.
3. Realizamos la clasificación (siguiendo lo propuesto en el clasificador *knn*).
4. Devolvemos la clase más probable.

2.3 WILSON

Por último, nos enfrentamos a aplicar todo esto en la función *wilson* la cual implementa el algoritmo deseado. Nuestro enfoque tuvo como objetivo replicar lo visto en teoría de la forma más fidedigna. En nuestro caso esta función recoge el conjunto de muestras *X* y sus etiquetas *xl*

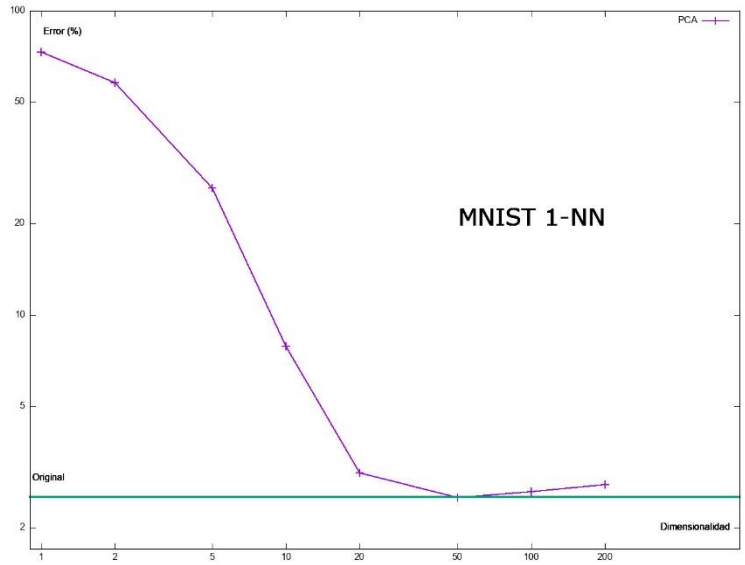
así como el número *k* de vecinos más cercanos a tener en cuenta y devuelve los índices de las muestras

Su funcionamiento es tal que:

1. Inicializamos *ind* con los índices de todas las muestras que permanecerán en el conjunto.
2. Calculamos la matriz *V* para las muestras que
3. Realizamos lo siguiente siempre que eliminemos alguna muestra en el bucle.
    a. Para cada muestra de las disponibles comprobamos que su clase coincidiría con la más cercana por *knn*
    b. En caso de que esto no sea así la eliminamos del conjunto de muestras

2.4 PCA+KNN+WILSON-EXP.PY

Hemos realizado una implementación paralela a la del ejercicio obligatorio. Indicando el porcentaje de muestras de entrenamiento, las de test y las dimensionalidades a probar.



2.4.1 Resultados

Como podemos observar en la imagen los resultados son muy similares a los obtenidos sin aplicar el algoritmo de Wilson. Aún así cabe destacar que de forma general presenta una mayor tasa de error a partir de la dimensionalidad óptima.

Esto nos permite confirmar que aún presentando una mayor una tasa de error, nos permite tener un menor número de muestras de entrenamiento (dependiendo de la dimensionalidad entre un 7.36% y un 26.74%). Esto tendrá un efecto significativo para entrenamientos con un gran número de prototipos sin sacrificar en gran medida el error de nuestro clasificador.

2.5 PCA+KNN+WILSON-EVA.PY

De nuevo una aproximación similar a la mostrada en el ejercicio obligatorio. La única diferencia en que aquí hemos permitido la posibilidad de incluir múltiples dimensionalidades.

2.5.1 Resultados

Los resultados son realmente similares entre las muestras de test y las de evaluación. No hay nada especialmente a destacar sobre estos resultados aportados.