

MATERIA DE EMBEBIDOS

Proyecto Comparacion del Sistema KN-N y CNN

Jonathan F. Garcia Navarro
Alexander.S Sinche Flores

14 de diciembre de 2018

1. Introducción

En la actualidad son muchos los problemas de reconocimiento de patrones con una gran cantidad de información, por lo que es necesario recurrir a técnicas que reduzcan la talla del conjunto de entrenamiento para la clasificación, en este trabajo, el objetivo es realizar una comparación de KNN y CNN y de acuerdo a las ventajas elegir cual es el mejor. El modelo de clasificación de los K vecinos más cercanos o K-NN (K Nearest Neighbours), es muy ideal y

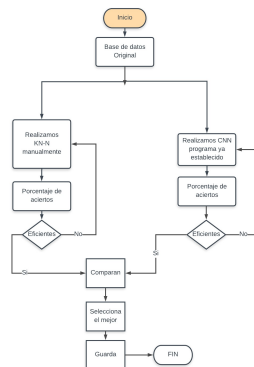
sencillo dependiendo de los datos que queramos procesar; el algoritmo clasifica cada dato nuevo en el grupo que corresponda, según tenga k vecinos más cerca de un grupo o de otro. Es decir, calcula la distancia del elemento nuevo a cada uno de los existentes, y ordena dichas distancias para seleccionar el grupo al que pertenecer. El método CNN se basa en lo mismo pero aquí en cambio se reutilizan los mismos datos los cuales son adecuados y erróneos.

2. Diseño del Sistema

2.1. Diagrama de Flujo

Se muestra el diagrama de flujo para uno mejor comprensión del proceso a realizar.

Figura 1: Diagrama de flujo para KN-N y CNN

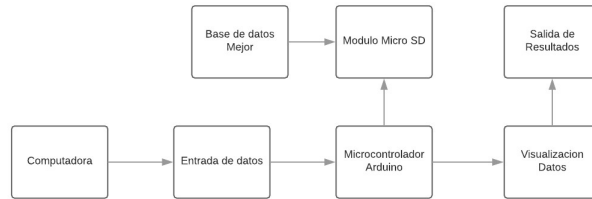


Basado en el diagrama de flujo de la Figura 1. Nos podemos guiar para ejecutar el proceso de selección de los datos ya establecidos por el medio de los algoritmos de KN-N y CNN, para al final realizar un proceso de comparación y seleccionar el proceso más adecuado con más eficiencia de aciertos.

Diagrama de bloques

En la Figura 2. Nos permite visualizar los pasos que va a realizar nuestro prototipo en su funcionamiento desde la comunicación serial hasta el proceso de guardar en el micro SD. Después de seleccionar el método que da más porcentaje de aciertos para proceder a grabarlo..

Figura 2: Diagrama de bloques



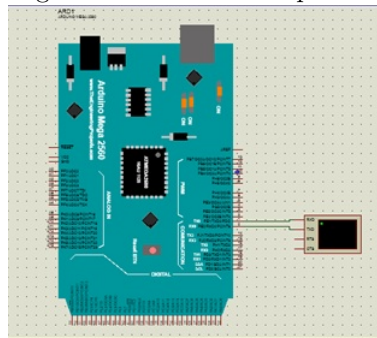
3. Desarrollo

3.1. Simulación

Simulación para los casos de KN-N y CNN.

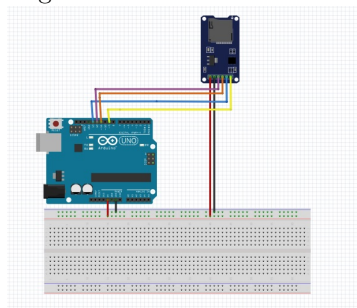
Las simulaciones que realizamos las hacemos con el arduino Mega 2560, debido a que nos ofrece más capacidad de memoria para poder guardar las bases de datos y que no nos ocasione problemas. Aquí podemos simular los algoritmos de KN-N y CNN y ver el porcentaje de eficiencia de cada uno para poder seleccionar el adecuado y posteriormente proceder a guardarlo en el armado de arduino uno ya que reducimos el espacio de almacenamiento requerido.

Figura 3: Simulación en proteus



Para el armado solo necesitaremos una placa de arduino uno y un módulo de micro SD. Lo cual lo podemos representar en la siguiente figura.

Figura 4: Armado del sistema



4. Análisis de Resultados

Se realizó la programación en el software de Arduino 1.8.5. realizando cada método por separado con su respectiva simulación para al final compararlos y escoger el mejor criterio.

4.1. Metodo-KNN

Para el método de KNN lo que hicimos fue obtener una matriz base en este caso una de 120 X 5 lo cual procedimos a obtener el promedio de cada grupo de etiquetas para realizar la obtención de las distancias

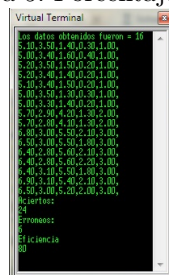
respectivamente, para después sacar el valor máximo de la distancia la cual procedemos a dividirla para cada una y obtener en escala de 0 a 1 los datos, después por un método obtenemos los puntos a tratar y así reducir el tamaño de la matriz de forma manual, para lo cual vamos a probar el porcentaje de error con una matriz de prueba a la cual le aplicamos el método de KNN y nos sale el porcentaje de eficiencia.

Cuadro 1: Base de datos seleccionada
Tabla realizada con centro $r=0.2$

Dato 1	Dato 2	Dato 3	Dato 4	Etiqueta
5,1	3,5	1,4	0,3	1
5	3,4	1,6	0,4	1
5,2	3,5	1,5	0,2	1
5,2	3,4	1,4	0,2	1
5,1	3,4	1,5	0,2	1
5	3,5	1,3	0,3	1
5	3,3	1,4	0,2	1
5,7	2,9	4,2	1,3	2
5,7	2,8	4,1	1,3	2
6,8	3	5,5	2,1	3
6,5	3	5,5	1,8	3
6,4	2,8	5,6	2,1	3
6,4	2,8	5,6	2,2	3
6,4	3,1	5,5	1,8	3
6,9	3,1	5,4	2,1	3
6,5	3	5,2	2	3

En la Figura 5 podemos ver el porcentaje de funcionamiento ya con el programa ejecutándose correctamente en la simulación con el $r=2$.

Figura 5: Porcentaje KNN



4.2. Metodo-CNN

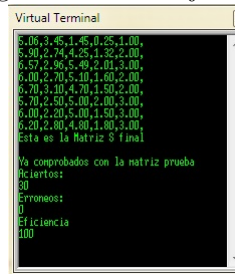
Para el método CNN lo hacemos con una base de 90 x 5 tuvimos que reducirlo ya que en nuestra programación tuvimos problemas de espacio al compilar. Sacamos los puntos medios de cada etiqueta los cuales les vamos a almacenar la primera instancia y realiza un primer barrido sobre el conjunto de datos, agregando a la bolsa almacenada aquellas instancias que no están clasificadas correctamente por K-NN tomando la bolsa almacenada como conjunto de entrenamiento. Luego, el proceso se itera hasta que todas las instancias no almacenadas se clasifican correctamente.

Cuadro 2: base de datos final CNN

Dato 1	Dato 2	Dato 3	Dato 4	Etiqueta
5.06	3.45	1.45	0.25	1
5.9	2.74	4.25	1.32	2
6.57	2.96	5.49	2.01	3
6	2.7	5.1	1.6	2
6.7	3.1	4.7	1.5	2
5.7	2.5	5	2	3
6	2.2	5	1.5	3
6.2	2.8	4.8	1.8	3

En la Figura 6 podemos ver el porcentaje de funcionamiento con el algoritmo CNN.

Figura 6: Porcentaje CNN



4.3. Comparacion de Resultados

CNN elimina todas aquellas instancias no críticas para la clasificación es decir reduce mucho la necesidad de almacenamiento, pero este método tiende a conservar aquellas instancias con ruido puesto que son mal clasificadas por las instancias de store además depende mucho del orden en que se toma las instancias.

KNN Clasifica nuevas instancias como la clase mayoritaria de entre los k vecinos más cercanos entre los datos de entrenamiento pero este es un algoritmo perezoso ya que durante el entrenamiento, solo guarda las instancias, no construye ningún modelo pero es método es muy sencillo.

Obtenido los resultados de los porcentajes procedemos a la comparación de KNN y CNN para determinar cuál es el mejor criterio, como podemos observar en los procesos anteriores CNN es el criterio que a usar ya que es un método sencillo y con muy buenos resultados del 100 en este caso. Ya que en el algoritmo KNN la matriz final no es muy eficiente debido a la cantidad de datos y el porcentaje de eficiencia del 80 “esto puede ser ajustado por el radio de selección”.

Cuadro 3: Comparación
Eficiencia de cada Algoritmo

KNN	80 %
CNN	100 %

5. Conclusiones y Recomendaciones

5.1. Conclusiones

El algoritmo de Kn-n que nosotros lo realizamos manualmente puede que sea mejor que el CNN pero en otros casos se usa mas el metodo de CNN debido a los datos que podriamos manejar.

La base de datos usada en Arduino uno ocupa mucho espacio lo cual no permitia compilar por medio de simulaciones vimos que el adecuado es el Arduino Mega.

El método más adecuado depende mucho de la reducción de datos y del porcentaje de funcionamiento comparado entre los dos métodos para así tener un programa óptimo.

5.2. Recomendaciones

Para la selección de datos manuales se debe tomar de muchas pruebas ya antes hechas para verificar cual es la mejor lógica de comparación.

Reducir de una manera considerable los datos y escoger adecuadamente el criterio de kn-n que vamos a usar para así ver la eficiencia final ya que todos los criterios de kn-n son distintos.

Antes de guardar información en la micro sd debemos escribir bien el código especificando los parámetros a guardar para evitar información errónea.