SISTEMA EMBEBIDO PARA EL RECONOCIMIENTO DE ESCRITURA SOBRE PLATAFORMA LINUX EMBEBIDO

EMBEDDED SYSTEM FOR THE RECOGNITION OF WRITING ON EMBEDDED LINUX

Freddy Rodrigo Mendoza Ticona[1]

[1.] Facultad de Ingeniería Eléctrica y Electrónica, Universidad Nacional de Ingeniería, Lima, Perú.
E-mail:freddy12120@gmail.com

Recibido: 3 Diciembre
Aceptado: 15 Diciembre
*Correspondencia del autor. Facultad de Ingeniería Eléctrica y Electrónica, Universidad Nacional de Ingeniería, Lima, Perú,
E-mail: freddy12120@gmail.com

RESUMEN

En este artículo se presenta un sistema embebido para el reconocimiento de escritura o caracteres utilizando redes neuronales artificiales, el tipo de RNA que se utiliza es el mapa auto-organizado (Self Organizing Map o SOM) porque consume menos recursos de memoria y es de bajo coste computacional. El hardware utilizado la tarjeta de desarrollo beaglebone black conectada a una pantalla táctil resistiva, la tarjeta beaglebone black cuenta con un procesador ARM Cortex-A8 y posee como sistema operativo linux embebido. Se realiza una interfaz gráfica mediante la biblioteca Qt para linux embebido, en esta interfaz se ingresan los caracteres por medio de un lápiz táctil (stylus) para posteriormente extraer las características de la imagen y procesarlas. El sistema implementado se prueba como parte de una aplicación de calculadora que realiza las cuatro operaciones elementales, y por último, se realizó la evaluación del desempeño del mapa auto-organizado (SOM).

Palabras claves: Beaglebone black, mapa auto-organizado (SOM), linux embebido, pantalla táctil resistiva, Qt embebido, Red neuronal artificial RNA, sistema de reconocimiento de caracteres.

ABSTRACT

In this article, an embedded system for handwriting recognition or characters using artificial neural networks is presented. The type of RNA that is used is the self-organizing map (SOM Self Organizing Map) because it consumes less memory and it is low computational cost. The hardware development board used was the BEAGLEBONE black connected to a resistive touch screen. The card has a black BEAGLEBONE ARM Cortex-A8 processor and has as Linux embedded operating system. A graphical interface using the Qt library for embedded linux. In this interface, characters were entered using a touch pencil and then extract features from the image and process them. The implemented system was tested as part of a calculator application that performs the four basic operations, and finally, the evaluation of the performance of self-organizing map (SOM) was held.

Keywords: BEAGLEBONE black, self-organizing map (SOM), embedded linux, resistive touch screen, embedded Qt Artificial Neural Network RNA, character recognition system.

INTRODUCCIÓN

El reconocimiento de escritura (Handwritten Recognition o HWR) consiste en la capacidad de interpretar de forma transparente la escritura hecha por el hombre, de diferentes fuentes como documentos, una fotografía o pantallas táctiles. Existen dos enfoques el reconocimiento en línea (Online) y fuera de línea (Offline) (1), en este último se escanea la escritura de una hoja de papel, esto es llamado reconocimiento óptico de caracteres (optical character recognition o OCR), y en el Online se ingresan los caracteres en una superficie táctil, donde la tarea consiste en interpretar el movimiento del lápiz táctil (stylus) a través de la superficie táctil y convertirlo en texto digital, lo último se implementa en tablets, celulares o algún otro dispositivo. Este tema ha sido extensamente estudiado por un gran número de investigadores en todo el mundo y varias técnicas han sido propuestas, se ha hecho el reconocimiento de caracteres y palabras en varias lenguas (2), (3). Sin embargo, el reconocimiento de escritura sigue siendo una tarea difícil de realizar, debido principalmente a los diferentes estilos en la escritura. Diferentes tipos de algoritmos han sido empleados para el reconocimiento de escritura como la Maquina de Soporte Vectorial (SVM), Perceptron Multicapa (4), Mapa auto-organizado (SOM) (5), lógica difusa y varias combinaciones de estos algoritmos (6), básicamente los algoritmos consisten en la clasificación de patrones o reconocimiento de patrones. Las aplicaciones son variadas, así por ejemplo en compañías postales con el propósito de ordenar automáticamente paquetes haciendo una lectura de sus direcciones de destino, en el manejo de votos electorales, el reconocimiento de firmas y también en aplicaciones de software para el reconocimiento óptico de una imagen de texto para convertirlo a formato digital con la finalidad de almacenarlos en formato digital.

ARQUITECTURA DEL SISTEMA EMBEBIDO

El principal objetivo de este trabajo es realizar una aplicación de software para el reconocimiento Online de números y símbolos matemáticos, y se prueba como una calculadora, es decir, implementa el reconocimiento de los símbolos de entrada y luego calcula la operación deseada. Esta aplicación es implementada en lenguaje C++ utilizando dos APIs conocidas, Qt library (7) para el manejo de la interfaz gráfica y OpenCV (8) para realizar los algoritmos de procesamiento de imágenes, esta aplicación se construye sobre Linux embebido y se ejecuta sobre una computadora embebida llamada Beaglebone Black (9), esta tarjeta de bajo coste posee como procesador un ARM Cortex-A8, 512MB de DDR3L RAM y Angstrom Linux (10) como sistema operativo. Angstrom Linux esta diseñada para dispositivos ARM y es ideal para este tipo de arquitectura.

Figura 1. Tarjeta de desarrollo Beaglebone black y Beaglebone black con la LCD 4DCAPE-43T

La tarjeta de desarrollo Beaglebone black es una computadora embebida que tiene Angstrom Linux (otras distribuciones pueden ser instaladas), cuenta con varios puertos GPIO, conexión USB, puerto Ethernet, etc. El linux embebido que posee soporta una variedad de librerías de varios lenguajes de programación ideal para el desarrollo de aplicaciones embebidas, como por ejemplo OpenCV library y Qt embedded library. Para poder ejecutar la tarea de reconocimiento de escritura, el sistema cuenta con una pantalla táctil resistiva 4DCAPE-43T de 4.3" y 480x272 de resolución como se puede ver en la Figura. 1, la aplicación en QT se muestra en esta pantalla.

Modelo del Sistema

La metodología usada para construir el sistema de reconocimiento se puede apreciar en la Figura. 2, consiste en la Adquisición de los datos, Segmentación de los datos, extracción de características de los símbolos y la clasificación de patrones. Posteriormente se evalúan las operaciones matemáticas necesarias para obtener el resultado final.

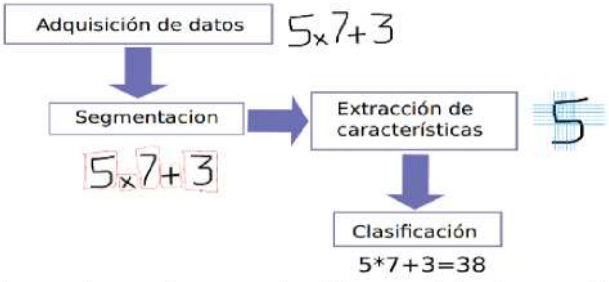

Figura 2. Diagrama de bloques del sistema de reconocimiento

Adquisición de datos

En esta parte se elabora la interfaz de usuario usando la biblioteca QT y la pantalla táctil, se ingresan los datos por medio de un lápiz táctil al área de trabajo, como se muestra en la Fig. 3, los datos se guardan como imágenes en escala de grises.

Figure 3. Adquisición de datos

Segmentación

Un segmento es un digito o en general un símbolo, en esta parte se separa cada símbolo, esto se consigue rastreando las coordenadas del lápiz táctil, se hace uso de los eventos de mouse que proporciona la librería Qt para obtener las coordenadas de los pixeles trazados, con esta información se encierra el símbolo en una área rectangular mínima dando flexibilidad al sistema de posicionar los símbolos en cualquier parte del área de trabajo. La imagen de cada símbolo se almacena temporalmente en memoria para luego realizar la extracción de características de cada uno.

Extracción de características

Existen principalmente dos formas de considerar las características de cada símbolo, una es tomando todos los pixeles de la imagen, esto da buenos resultados pero requiere más calculo computacional y por lo tanto mayor tiempo en obtener la respuesta, además la imagen tendría que ser de un tamaño fijo.

Por otro lado, se suele hacer una reducción en la cantidad de características mediante la codificación de cada símbolo, debido a que la imagen esta en formato binario, se extrae el número de transiciones de 0 a 1 (líneas negras vistas verticalmente y horizontalmente) como en la Figura. 4, con esta técnica los símbolos pueden ser de diferentes tamaños y formas, dando mayor flexibilidad al sistema embebido.

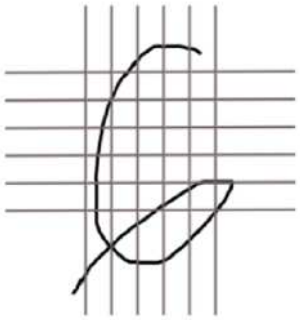

Figura 4. Posicionamiento de las líneas verticales y horizontales para la extracción de características de un símbolo.

Tanto las líneas verticales como horizontales están separadas de forma homogénea, se elige trazar 6 líneas tanto verticales como horizontales, teniendo un total de 12 características y el espaciado entre líneas que se elige es de 1/8 de la dimension vertical y horizontal (11), con esto se consigue buenos resultados en la clasificación.

$$P = [n_1\ n_2\ n_3\ n_4\ n_5\ n_6\ n_7\ n_8\ n_9\ n_{10}\ n_{11}\ n_{12}]^T \quad (1)$$

La Eq. 1 muestra un ejemplo de un patrón que representa un símbolo, esta forma de vector de características se utiliza en la etapa de clasificación.

MAPA AUTO-ORGANIZADO (SOM)

El mapa auto-organizado (self organizing map) o también llamado mapa de kohonen es un tipo de red neuronal de aprendizaje no supervisado competitivo, este algoritmo traslada un espacio de entrada de alta dimension a un espacio de salida de baja dimension conservando la topología del espacio de los datos de entrada, tienen la habilidad de encontrar similitudes en los datos y tienen la propiedad especial de crear de forma efectiva representaciones internas especialmente organizadas de varias características de las señales de entrada y sus abstracciones (12).

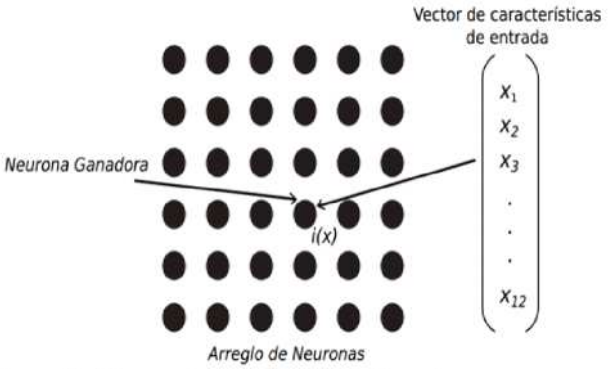

Figura 5. Mapa auto-organizado de Konohen.

En el mapa auto-organizado se tiene a las neuronas en los nodos de una rejilla que es usualmente de dos dimensiones que puede ser rectangular o hexagonal como se puede ver en la Fig. 5, un mapa topográfico es automáticamente organizado por un proceso cíclico de comparación de los patrones de entrada con el vector de pesos de cada neurona, la neurona que posee el vector de pesos más cercanos al vector de entrada es llamada neurona ganadora, el algoritmo consiste primero en inicializar los pesos de la red aleatoriamente, luego existen tres procesos esenciales involucrados en la formación del mapa auto-organizado (13).

Competencia

Dado un vector de entrada como en la Figura. 5 se encuentra la neurona más cercana al vector de entrada, el criterio de esta medida se basa en la distancia Euclidiana.

$$i(\mathbf{x}) = arg\ min\ |\mathbf{x} - \mathbf{w}_j|, j = 1,2,.....,1 \qquad (2)$$

Cooperación

La neurona ganadora (BMU) es el centro de una vecindad topológica de cooperación, se usa la función gausiana como funcion de vecindad, las neuronas que se encuentren dentro de la vecindad de la BMU son actualizadas.

$$h_{j,i(x)}(n) = \exp\left(-\frac{d_{j,i}^2}{2\nu^2(n)}\right) \qquad (3)$$

La función de vecindad decae con la distancia en el mapa de Kohonen y también con el tiempo.

$$\sigma(n) = \sigma_0 \exp\left(-\frac{n}{x_1}\right) \qquad (4)$$

Donde: $i(\mathbf{x})$, representa la ubicación de la neurona ganadora y j , representa las demás neuronas. La distancia entre neuronas tiene la siguiente expresión.

$$d_{j,i}^2 = |\mathbf{r}j - \mathbf{r}i|^2 \qquad (5)$$

Adaptación

Los pesos de las neuronas se actualizan en dirección a los patrones de entrada de la siguiente manera.

$$w_j(n+1) = w_j(n) + \eta(n)\, h_{j,i(x)}(n)(\mathbf{x} - \mathbf{w}_j(n)) \qquad (6)$$

Donde la tasa de aprendizaje disminuye con el tiempo.

$$\eta(n) = \eta_0 \exp\left(-\frac{n}{x_2}\right) \qquad (7)$$

El número de iteraciones varía de acuerdo al problema, pero no depende del número de muestras presentes en el espacio de entrada, además al inicio del algoritmo se debe tener una función vecindad que cubra casi todo el arreglo y la tasa de aprendizaje debe ser cercano a la unidad, en este artículo se eligió un arreglo rectangular de 15x15.

Clasificación

Una vez realizado el entrenamiento de la red neuronal se asigna la clase a que pertenece cada neurona, esta es la clase de la mayoría de muestras de entrada que son los más cercanos a dicha neurona, se puede ver un ejemplo en la Fig. 6, luego el mapa está listo para clasificar muestras desconocidas que no participaron en el entrenamiento (generalización) mediante el criterio de similitud o vecino más cercano.

4	9	4	4	7	7	7	9	9	9	8	8	8	4	4
4	4	4	4	9	7	7	9	9	9	8	8	1	1	4
4	4	4	4	9	7	7	7	9	7	9	1	1	1	7
4	4	4	4	4	2	7	7	9	7	1	1	1	4	2
5	5	4	4	4	9	7	7	7	1	1	1	8	1	5
5	5	8	9	8	8	7	7	7	1	1	1	9	8	0
5	5	5	8	8	8	8	7	1	1	1	6	5	6	0
0	5	8	8	8	8	8	2	2	8	6	6	6	0	0
0	0	0	8	8	8	2	2	2	6	6	6	0	0	0
0	0	5	5	8	8	2	2	2	2	6	6	0	0	0
0	0	0	5	5	3	3	2	2	2	6	6	6	0	0
0	0	0	5	3	3	3	3	2	2	2	6	6	6	6
6	0	5	3	3	3	3	3	2	2	2	2	6	6	6
6	8	9	3	3	3	3	3	3	2	2	2	2	6	4
4	9	9	7	7	3	3	3	8	2	2	2	2	2	4

Figura 6. Mapa de características entrenada usando el algoritmo de aprendizaje SOM.

RESULTADOS

Se entrena el mapa auto-organizado (SOM) con un total de 1288 muestras (train samples) y se evalúa el desempeño del mapa con 538 muestras (test samples). Las muestras de entrenamiento fueron capturadas mediante la interfaz gráfica y posteriormente enviadas a la PC por medio del protocolo SSH (Secure SHell File Transfer Protocol), el entrenamiento se puede hacer en la misma tarjeta de desarrollo pero por comodidad se realizó en la PC usando el entorno de desarrollo QT creator, luego la matriz de pesos obtenida se envía en formato YAML a la tarjeta Beaglebone Black, este formato es fácilmente interpretado por la biblioteca OpenCV. Una vez entrenada el mapa auto-organizado (SOM) el sistema funciona independientemente como una calculadora con las cuatro operaciones elementales, además se obtiene una precisión del 86.3 %

en la clasificación de los patrones de ensayo, el sistema embebido implementado en este articulo funciona correctamente y con retardo de procesamiento mínimo.

Tabla 1 Precisión obtenida en el reconocimiento de muestras de ensayo

	Muestras de entrenamiento	Muestras de ensayo
Técnica	Precisión(%)	Precisión(%)
SOM	96.1	86.3

Figura 7. Plataforma de hardware del reconocimiento de símbolos escritos a mano

CONCLUSIÓN

Una técnica de reconocimiento de escritura usando el mapa auto-organizado (Self-organizing maps o SOM) es presentado en este artículo, se implementa un sistema embebido que funcione como una calculadora usando la tarjeta de desarrollo Beaglebone black con una pantalla táctil, se realizó una aplicación de software en Qt para la interfaz de usuario y demás operaciones del sistema embebido, todo la aplicación esta hecha en lenguaje C++. La red neuronal SOM es usada para categorizar e identificar los dígitos y los operadores matemáticos ingresados por el usuario, la red neuronal fue hecha para trabajar con un número reducido de características dando mayor rendimiento al sistema. Por último, se obtiene buenos resultados en el entrenamiento de la red neuronal y en la precisión de clasificación de los patrones de ensayo. Trabajos futuros están enfocados en el aumento de la precisión obtenida con una mezcla de técnicas y también aumentar los símbolos que serán clasificados, como por ejemplo el reconocimiento de letras del alfabeto español, una mejor opción es diseñar una arquitectura de hardware en un FPGA para un reconocimiento de escritura más sofisticado pudiendo usar reconfiguración parcial dinámica, con la combinación de un procesador secuencial (ARM) y un FPGA.

BIBLIOGRAFÍA

1. J. Wang, Group for User Interface Research, University of California, Berkeley, “Online Handwriting Recognition Technology-State of the Art”, 2003.
2. B. Alijla and K. Kwaik, “OIAHCR: Online Isolated Arabic Handwritten Character Recognition Using Neural Network'", in The Int. Arab Journal of Information Technology, Vol.9, No.4, 2012, 343-351.
3. Qizhen He, “A Hybrid Language Model for Handwritten Chinese Sentence Recognition”, in Int. Conf. on Frontiers in Handwriting Recognition (ICFHR)}, Bari, 2012, 129-134.
4. B. K. Verma, “Handwritten Hindi character Recognition using multilayer Perceptron and radial basis function neural network”, Proc. of the IEEE Int. Conf. on Neural Networks, 2002, 2111-2115.
5. D. Gil and M. Jhonsson, “Supervised SOM Based Architecture versus Multilayer Perceptron and RBF Networks”, Proc. of the Linkoping Electronic Conf., 2010, 15-24.
6. Z. Chi, J. WU and H. Yan, “Handwritten Numeral Recognition using Self-organizing maps and fuzzy rules”, in Pattern Recognition, Elsevier Ltd., Vol. 28, No. 1 , 1995, 59-66.
7. Qt Digia, Documentation Qt 4.8 library. [Online]. Available: http://qt-project.org/doc/qt-4.8/classes.html
8. OpenCV community, OpenCV API Reference. [Online]. Available: http://docs.opencv.org/2.4.6
9. BeagleBoard.org Foundation. [Online]. Available: http://beagleboard.org/Products/BeagleBone Black
10. The Angstrom Distribution Project, http://www.angstrom-distribution.org/
11. E.~Bouvett, “An FPGA Embedded System Architecture for Handwritten Symbol Recognition”, in 16th Electrotechnical Conf. (MELECON), Yasmine Hammamet, 2012, 653 - 656.
12. T. Kohonen, “The Self-Organizing Map”, Proc. of IEEE, Vol.78, No. 9, 1990, 1464-1480.
13. S. Haykin, “Self-Organizing Maps”, in Neural Networks: A Comprehensive Foundation, 2th ed. Prentice-Hall, 1999, ch. 9, sec.3, 468-476.