

Red neuronal de Kohonen para trazado de trayectoria de robot móvil, y aplicación de algoritmo Back-Propagation para reconocimiento de caracteres.

Cristian Ruiz 1802566, David Galvis 1802584, Eric Buitrago 1802410

Universidad Militar Nueva Granada

Resumen -- En este documento se lleva a cabo la implementación de una red neuronal mediante el algoritmo de back-propagation con el fin de reconocer caracteres ingresados a través de una interfaz gráfica de usuario, para esto se debió entrenar la red con 4 patrones diferentes para cada carácter y al finalizar el programa debe indicar qué carácter se acaba de ingresar. Además se desarrolla un mapa auto-organizativo de Kohonen con el fin de clasificar los caracteres sobre un campo de trabajo, sobre el cual se desplazará un robot móvil, para llevar a cabo este desarrollo fue necesario implementar las 3 etapas fundamentales, fase competitiva, fase cooperativa y fase adaptativa.

Abstract -- In this document the implementation of a neural network is carried out by means of the back-propagation algorithm in order to recognize characters entered through a graphical user interface, for this the network had to be trained with 4 different patterns for each character and at the end of the program you must indicate which character has just been

entered. In addition, a self-organizing map of Kohonen is developed in order to classify the characters on a work field, on which a mobile robot will move. To carry out this development it was necessary to implement the 3 fundamental stages, competitive phase, phase cooperative and adaptive phase.

Palabras clave: Red neuronal, aprendizaje, adaptación, reconocimiento.

Key words: Neural network, learning, adaptation, recognition.

INTRODUCCIÓN

Las redes neuronales artificiales son redes interconectadas masivamente en paralelo de elementos simples (usualmente adaptativos) y con organización jerárquica, las cuales intentan interactuar entre sí y con objetos del mundo real. Debido a su constitución y a sus fundamentos, presentan un gran número de características semejantes a las del cerebro. Son capaces de aprender de la experiencia, de generalizar de casos anteriores a nuevos casos,

de abstraer características esenciales a partir de entradas que representan información irrelevante. Entre las ventajas que estas ofrecen se incluyen[1]:

- Aprendizaje adaptativo
- Auto-organización
- Tolerancia a fallos
- Operación en tiempo real

Modelado estándar de las redes neuronales

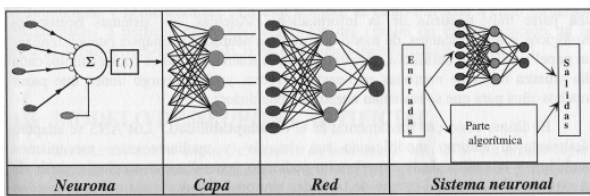


Figura 1: Sistema global de proceso de una red neuronal [2]

Objetivos

- General

Diseñar y construir un sistema de reconocimiento de caracteres basado en redes neuronales.

-Específicos

Implementar un software para la creación, entrenamiento (algoritmo de Back-propagation) y uso de redes neuronales.

MATERIALES

- Matlab R2017a

PROCEDIMIENTO

Algoritmo Back-Propagation

Para la implementación del reconocimiento de patrones (caracteres) utilizando una red neuronal se definen primero los parámetros del algoritmo, el error mínimo deseado, un factor de aprendizaje inicial (ya que para esta aplicación este será dinámico) y un factor de momento. Luego de estos parámetros básicos se configuran los parámetros del error y se inicializa un error por época de cero.

Se definen 4 patrones diferentes para 10 caracteres distintos, en base a esto y al definir una cantidad de capas se puede continuar con la inicialización de la matriz de pesos, que al principio son pesos aleatorios que posteriormente se guardarán en vectores diferentes. Se iguala cada salida de las neuronas a cero al igual que las entradas de cada capa. También se inicializan los cambios de los pesos (ΔW delta pesos) en cero.

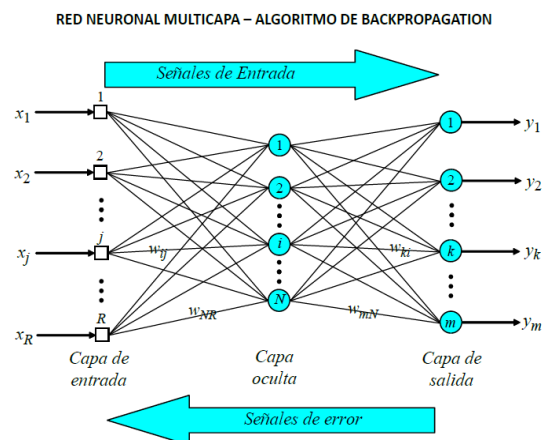


Figura 2: Algoritmo Back-Propagation [2]

Luego de la inicialización de los valores de salida en cero, se procede a realizar el algoritmo back-propagation primero haciendo un recorrido hacia adelante.

```

2. Recorrido hacia adelante y backpropagation
while (mse > errorMIN) && (epoch <= 5000-1) %Dos parametros de stop (error y epoca)
    e = zeros(Q,P);
    for p = 1:P
        %Recorrido forward
        aux{1}(1:end-1) = X(:,p);
        for i = 1:depth-1
            y[i] = W[i]*aux[i];
            if i < depth-1
                aux[i+1](1:end-1) = tansig(y[i]); %Capas ocultas con funcion de activacion tanh
            else
                aux[i+1] = logsig(y[i]); %Capa de salida con funcion de activacion log
            end
        end
    end
end

```

En la imagen anterior se observa que para las capas ocultas se utiliza la función de activación tangente hiperbólica que en Matlab corresponde al comando `<tansig>` y para las capas de salida se usa la función de activación logaritmo. Y teniendo esto se procede a hallar la sensibilidad de la capa de salida, sobre la cual se trabaja al momento de ajustar los pesos con el momento definido anteriormente, se hace el recorrido hacia atrás `<Backpropagation>`. Posterior al ajuste de pesos, se guardan los pesos anteriores ya que con estos se halla nuevamente el momento. La expresión para la actualización de pesos se da de la siguiente manera.

$$\omega_{ij}^{(l)}(k+1) = \omega_{ij}^{(l)}(k) + \alpha * \delta_i^{(l)}(k) * y_j^{(l-1)}(k)$$

Para las iteraciones, se calcula el error cuadrático medio y se guarda en una variable, para luego aumentar otra época, cabe resaltar que para este programa se define un máximo de 5000 épocas. El alfa es dinámico y va cambiando en función de cómo aumentan las épocas, definido para que cuando el error aumenta, el alfa disminuye un 70% y para el caso contrario cuando el error disminuye, el alfa aumenta su valor a un 105%. Ya que el factor de aprendizaje solo puede tomar valores entre 0.1 y 0.9 se hace una saturación de estos dos valores críticos para que el sistema no falle.

Red de Kohonen

Para esta aplicación se simula un espacio de trabajo de dos metros de largo por dos metros de ancho, sobre la cual el robot móvil se moverá, se ubican aleatoriamente 40 puntos y un número de neuronas que generalmente es 3 veces el número de puntos, para este caso será entonces de 120, estos 120 se pueden ubicar también aleatoriamente sobre el campo. Pero para este algoritmo se ubican sobre una línea diagonal en el campo de trabajo. Luego de esto inicia la fase competitiva entre neuronas, donde se calcula la distancia de cada uno de los nodos y ganará la neurona la cual su vector de pesos sea más parecido al vector de entrada, para esto se utiliza la distancia entre puntos sobre un plano.

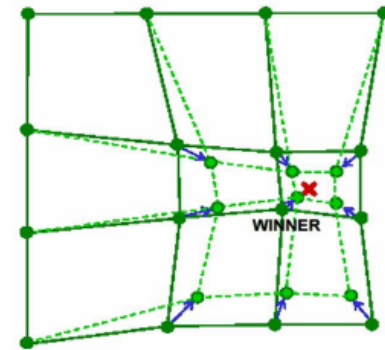
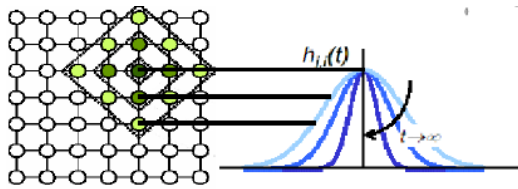


Figura 3: Red de Kohonen

Al finalizar la fase competitiva, se sigue con la fase cooperativa, en la cual alrededor de la neurona ganadora se define una vecindad de cooperación, donde las otras neuronas se "acercan" a la ganadora, esta función se expresa de la siguiente manera:

$$h(k) = e^{-\left(\frac{d^2}{2\sigma^2}\right)}$$

Donde d es el índice de distancia entre la neurona ganadora y sus vecinas.



Y finalmente la fase adaptativa en la cual los pesos ahora se actualizan en función de la neurona ganadora y con ella toda la vecindad, la actualización se da a través de la siguiente expresión vectorial:

$$\omega_j(k+1) = \omega_j(k) + \mu(k) * h(k) * (x - \omega_j(k))$$

Y a su vez, en esta misma fase ocurre que el tamaño de la vecindad y el factor de aprendizaje disminuyen a medida que avanzan las iteraciones, el factor de aprendizaje varía en función de las iteraciones k .

$$\mu(k) = \mu_0 * e^{\left(-\frac{k}{\tau_2}\right)}$$

RESULTADOS

Posición aleatoria de puntos y neuronas.

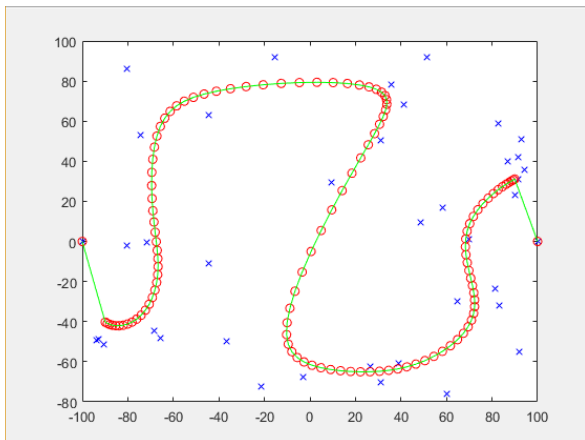


Figura 4: Posición aleatoria de entradas y neuronas

Luego de un tiempo transcurrido y de iteración tras iteración, las neuronas (círculos) se adaptan a las entradas (presentadas con x azules).

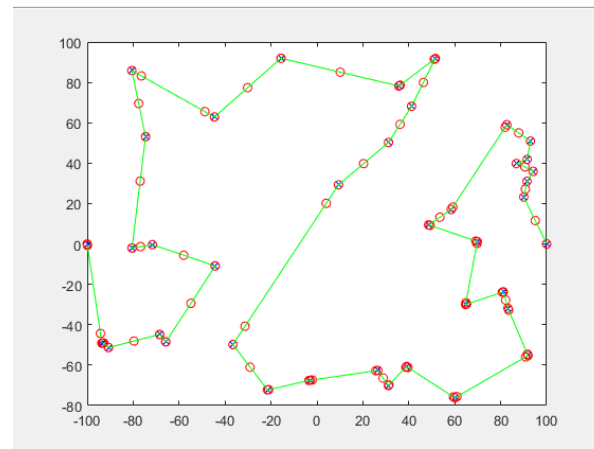


Figura 5: Neuronas adaptadas

CONCLUSIONES

Se logra crear la red neuronal de manera que reconozca y entrene distintos patrones de caracteres ingresados, y que con ayuda de un alfa dinámico y un momento en los pesos se logró la disminución de iteraciones en comparación con un sistema que no tenga estas características.

REFERENCIAS

- [1]D. Matich, "Redes neuronales: conceptos básicos y aplicaciones", Informática aplicada a la ingeniería de procesos, Universidad Tecnológica Nacional (Regional Rosario), Argentina, 2001.
- [2]P. Larrañaga, I. Inza, A. Moujahid, "Redes Neuronales", Universidad del País Vasco - Euskal Herriko Unibertsitatea.
- [3] REDES NEURONALES, Ricardo Andrés Castillo MSc, Material de clase. Universidad Militar Nueva Granada