



“CONGRESO INTERNACIONAL DE INVESTIGACIÓN E INNOVACIÓN 2014”
Multidisciplinario
10 y 11 de abril de 2014, Cortazar, Guanajuato, México
ISBN: 978-607-95635

Módulo-RNA-Algorithm Backpropagation-DaaCIA

Norma Verónica Ramírez Pérez¹, Martín Laguna Estrada²
Cecilia Malagón Girón³, Edgar Yair Rodríguez Robles⁴, José Francisco Yepez Carrillo⁵

¹norma.ramirez@itcelaya.edu.mx

²martin.laguna@itcelaya.edu.mx

³ceciliamalagon.14@hotmail.com

⁴vairr_134@hotmail.com

⁵francisco_23191_acuario@hotmail.com

^{1,2,3,4,5}Instituto Tecnológico de Celaya

Resumen

En este trabajo se presenta una propuesta denominada Módulo-RNA-Tipo Backpropagation-DaaCIA, que formará parte del proyecto integral: “Demostración Académica de algoritmos computacionales de Inteligencia Artificial”, que servirá para realizar prácticas de los diferentes algoritmos de clasificación y reconocimiento de imágenes por alumnos de Ingeniería en Sistemas Computacionales e Ingeniería en informática que cursan asignaturas terminales.

Palabras clave: RNA, Backpropagation, Inteligencia Artificial

1. Introducción

En la actualidad existen diversos sistemas de demostración del funcionamiento de algoritmos de clasificación y reconocimiento de patrones, uno de ellos es el WEKA[1], el cual es muy usado en universidades europeas, sin embargo nunca sobra tener una herramienta que concentre las diferentes técnicas que se utilizan en el área de inteligencia artificial. En este trabajo se presenta el Módulo-RNA-Tipo Backpropagation-DaaCIA, el cual formará parte del proyecto denominado “Demostración Académica de Algoritmos computacionales de inteligencia artificial”, que tiene como finalidad ayudar de una manera didáctica y visual a los alumnos de las Carreras de ingeniería en Sistemas Computacionales e Ingeniería Informática principalmente, sin embargo, también puede ser ocupado en otras carreras pues el ser multidisciplinario

1 | “Congreso Internacional de Investigación e Innovación 2014” Multidisciplinario, 10 y 11 de abril de 2014.
México



“CONGRESO INTERNACIONAL DE INVESTIGACIÓN E INNOVACIÓN 2014”

Multidisciplinario

10 y 11 de abril de 2014, Cortazar, Guanajuato, México

ISBN: 978-607-95635

enriquecerá más al mismo, donde no sólo abarca las redes neuronales, sino que también otras técnicas de clasificación como SVM(Máquinas de Soporte Vectorial), KNN(Algoritmos de Vecinos más Cercanos), Algoritmos genéticos, Average (Promedio). Se cuenta además con Algorithm Conformal prediction (Algoritmo de Predictores Conformales), que sirve para comprobar la calidad y la predicción de la clasificación. En este trabajo se detalla una breve introducción de las redes neuronales y el funcionamiento de la red neuronal tipo Backpropagation.

2. Marco teórico

2.1 Redes Neuronales

Una de las teorías relacionales que permiten la optimización del recurso de hardware son las redes neuronales, dichos algoritmos pretenden, a partir de su similitud con el sistema nervioso y las conexiones entre las neuronas, pretenden optimizar el desarrollo de los sistemas que impliquen el uso de dispositivos programables. En la teoría de las redes neuronales, se expone que todo sistema tiene la capacidad de realizar procedimientos que puedan aprovechar el aprendizaje partir de soluciones en las cuales el sistema interactúa [2]. Una red neuronal artificial es una estructura compuesta de un número de unidades interconectadas (neuronas artificiales), cada unidad posee una característica de entrada/salida e implementa una computación local o función. La salida de cualquier unidad está determinada por su característica de entrada/salida, su interconexión con otras unidades y (posiblemente) de sus entradas externas[3]. El cerebro humano contiene más de 100 billones de elementos de procesos llamados neuronas, que se comunican a través de conexiones llamadas sinapsis. Cada neurona está compuesta por tres partes fundamentales: el cuerpo, dendritas y axón. El cuerpo en sí, o capa externa, tiene la capacidad única de generar impulsos nerviosos. Las dendritas que son como las ramas que salen del cuerpo, poseen algunas conexiones sinápticas que reciben señales que generalmente vienen de otros axones. El axón se



“CONGRESO INTERNACIONAL DE INVESTIGACIÓN E INNOVACIÓN 2014”

Multidisciplinario

10 y 11 de abril de 2014, Cortazar, Guanajuato, México

ISBN: 978-607-95635

encarga de activar o inhibir otras neuronas las cuales a su vez son activadas por cientos o miles de otras neuronas. El funcionamiento de una red neuronal artificial está basado en este diseño, básicamente consiste en aplicar un conjunto de entradas, cada una representando una salida de otra neurona o una entrada del medio externo para realizar una suma ponderada con estos valores y filtrar este valor con una función, como se puede observar en la figura 1.

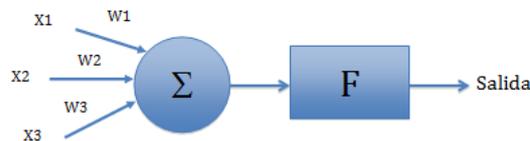


Figura 1. Cálculo de la salida de una neurona artificial

Cada neurona recibe un vector X de entrada que corresponde a todas aquellas señales que llegan a la sinapsis. Cada una de estas señales se multiplica por 1 que tiene asociado $W_1, W_2, W_3, \dots, W_n$, es decir al conjunto de pesos que se le denomina vector W . Cada peso representa la “intensidad” o fuerza de conexión de una sinapsis en una neurona biológica. Los resultados de estas multiplicaciones se suman. Esta sumatoria simula vagamente el cuerpo de una neurona biológica.

$$net = \sum x_j W_{ij}$$

2.2 Función de activación

Una neurona biológica puede estar activa (excitada) o inactiva (no excitada), es decir, que tiene un “estado de activación”. Las neuronas artificiales también tienen diferentes estados de activación, algunas de ellas solamente dos, al igual que las biológicas, pero otras pueden tomar cualquier valor dentro de un conjunto determinado. La función activación calcula el estado de actividad de una neurona; transformando la entrada global (menos el umbral, θ_i) en un valor



“CONGRESO INTERNACIONAL DE INVESTIGACIÓN E INNOVACIÓN 2014”

Multidisciplinario

10 y 11 de abril de 2014, Cortazar, Guanajuato, México

ISBN: 978-607-95635

(estado) de activación, cuyo rango normalmente va de (0 a 1) o de $(-1 a 1)$. Se presenta de esta manera, porque una neurona puede estar totalmente inactiva (0 o -1) o activa (1). La función activación, es una función de la entrada global (gx_i) menos el umbral (θ_i).

2.3 Función de salida

El último componente que una neurona necesita es la función de salida. El valor resultante de esta función es la salida de la neurona i (s_i); por ende, la función de salida determina que valor se transfiere a las neuronas vinculadas. Si la función de activación está por debajo de un umbral determinado, ninguna salida se pasa a la neurona subsiguiente normalmente. No cualquier valor es permitido como una entrada para una neurona, por lo tanto, los valores de salida están comprendidos en el rango $[0,1]$, $[-1,1]$. También pueden ser binarios $\{0,1\}$ o $\{-1,1\}$. Existen otros tipos de redes neuronales, una de las más conocidas fue presentada por Rosenblatt en 1962[4], que fue el perceptrón, una red neuronal con aprendizaje supervisado cuya regla de aprendizaje era una modificación de la propuesta de Hebb[2]. El perceptrón trabaja con patrones de entrada binarios, y su funcionamiento por tratarse de una red supervisada, se realiza en dos fases: la primera es donde se presentan las entradas y salidas deseadas, en esta fase aprende la salida que debe dar para cada entrada. La principal aportación del perceptrón es que la adaptación de ellos se realiza teniendo en cuenta el error entre la salida que da la red y la salida que se desea. La segunda fase es cuando la red es capaz de responder adecuadamente cuando se le vuelven a presentar patrones de entrada. En 1960 Widrow[5] publica una teoría sobre la adaptación neuronal y unos modelos inspirados en esa teoría, la Adaline (Adaptative linear neuron) y el Madaline (múltiple Adaline). Estos modelos fueron usados en numerosas aplicaciones y permitieron usar, por primera vez, una red neuronal en un problema importante del mundo real: filtros adaptativos para eliminar ecos en las líneas telefónicas, también existe la red de hopfield[6], donde este tipo de



“CONGRESO INTERNACIONAL DE INVESTIGACIÓN E INNOVACIÓN 2014”

Multidisciplinario

10 y 11 de abril de 2014, Cortazar, Guanajuato, México

ISBN: 978-607-95635

respuesta es recurrente y completamente conectada, funciona como una memoria asociativa no lineal que puede almacenar internamente patrones presentados de forma incompleta o con ruido, de esta forma puede ser usada como una herramienta de optimización, mediante la cual, el estado de una neurona puede ser actualizado un número indefinido de veces, independientemente del resto de las neuronas de la red pero en paralelo. En este proyecto nos enfocaremos en la red de tipo backpropagation que se resume en el siguiente apartado.

3. Metodología de análisis de las RNA-Backpropagation

Los autores Rumelhart, Hinton y Williams [7], basándose en los trabajos que generaron Werbos y Parker en 1982[8], formalizaron un método para que una red neuronal aprendiera la asociación que existe entre los patrones de entrada a la misma y las clases correspondientes, utilizando más niveles de neuronas que utilizó Rosenblatt para desarrollar el perceptrón. Este método, conocido en general como backpropagation (propagación del error hacia atrás), está basado en la generalización de la regla delta y, a pesar de sus propias limitaciones, ha ampliado de forma considerable el rango de aplicaciones de las redes neuronales. En la figura 2 se muestra la estructura de una red neuronal tipo backpropagation.

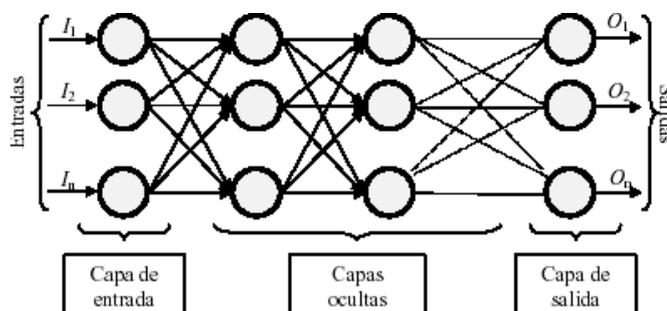


Figura 2. Estructura de una red neuronal tipo bakpropagation.



“CONGRESO INTERNACIONAL DE INVESTIGACIÓN E INNOVACIÓN 2014”

Multidisciplinario

10 y 11 de abril de 2014, Cortazar, Guanajuato, México

ISBN: 978-607-95635

El algoritmo de propagación hacia atrás, o retropropagación (backpropagation), es una regla de aprendizaje que se aplica en modelos de redes con más de dos capas de células. Una característica importante de este algoritmo es la representación interna del conocimiento que es capaz de organizar en la capa intermedia de las células para conseguir cualquier correspondencia entre la entrada y la salida de la red. De forma simplificada, el funcionamiento de una red backpropagation consiste en el aprendizaje de un conjunto predefinido de pares de entradas y salidas previamente dados. Un ejemplo sería, empleando un ciclo propagación y adaptación: primero se aplica un patrón de entrada para estimular la primera capa de las neuronas de la red y se va propagando a través de todas las capas superiores hasta generar una salida deseada y calculando un valor del error para cada neurona de salida, después estos errores se transmiten hacia atrás, partiendo de la capa de salida hacia todas las neuronas de la capa intermedia en la salida original. Este proceso se repite, capa por capa, hasta que todas las neuronas de la red hayan recibido un error que describa su aprobación relativa al error total. Basándose en el valor del error recibido, se reajustan los pesos de conexión de cada neurona, de manera que la siguiente vez que se presente el mismo patrón, la salida esté más cercana a la deseada; es decir, el error disminuya. La red backpropagation tiene la capacidad de autoadaptar los pesos de las neuronas de las capas intermedias para aprender la relación que existe entre un conjunto de patrones dados y sus salidas correspondientes para poder aplicar esa misma relación después del entrenamiento, a nuevos vectores de entrada con ruido, dando una salida activa si la nueva entrada es parecida a las presentadas durante el aprendizaje. Esta característica es importante porque exige a los sistemas de aprendizaje, la capacidad de generalización, extendida como la facilidad de dar salidas satisfactorias a entradas que el sistema no ha visto nunca en su fase de entrenamiento. Puede aplicar además a entradas no presentadas durante la etapa de aprendizaje para clasificarlas según las características que compartan con los ejemplos de entrenamiento.

4. Aplicación de una red neuronal tipo Backpropagation al procesado de imágenes

Antes de iniciar un reconocimiento de imágenes debemos empezar con el procesamiento de las mismas que nos permita realizar las entradas a la red backpropagation.

Las imágenes a reconocer que se presentan en este ejemplo son: una manzana, una naranja y una pera como se puede apreciar en la figura 3.



Figura 3. Imágenes a reconocer

Como primer paso para poder realizar el reconocimiento, debemos de realizar alguna modificación en el tamaño de las imágenes según se requiera por si alguna de ellas no cumple con las dimensiones requeridas. Para este proyecto las imágenes utilizadas son de tamaño 156 pixeles por 250 pixeles de ancho, por lo que se requirió hacer un redimensionamiento a todas ellas de la siguiente manera:

```
%Redimensionamos las imagenes a un mismo tamaño  
ima_rez_manzana = imresize(img_manzana, [20 35]);  
ima_rez_naranja = imresize(img_naranja, [20 35]);  
ima_rez_pera = imresize(img_pera, [20 35]);
```

Una vez realizado el redimensionamiento, debemos de convertir cada una de las imágenes a escala de grises:

```
ima_manzana_gris = rgb2gray(ima_rez_manzana);
```

“CONGRESO INTERNACIONAL DE INVESTIGACIÓN E INNOVACIÓN 2014”

Multidisciplinario

10 y 11 de abril de 2014, Cortazar, Guanajuato, México

ISBN: 978-607-95635

Ya teniendo la imagen en escala de grises, se puede proceder a realizar la obtención del contorno de la imagen. En este paso se utilizó la extracción de bordes tipo “canny” por ser la mejor técnica después de probar con: “sobel”, “prewit”, “roberts”, “log” y “zerocross”, que son otro tipo de extracción de bordes que maneja Matlab, y se le dio un valor de 0.5. Para ajustar el umbral y obtener una mejor calidad en la imagen, se puede observar en la figura 4 como cada uno de los bordes de las imágenes fueron tratados para una mejor visualización de la imagen mediante una inversión de colores, donde se utiliza el símbolo ~ renombrando nuevamente una nueva variable en la que se almacenará la inversión de colores con el siguiente código:

```
%Generacion de bordes para la imagen de la manzana  
ima_pera_BIN = edge(ima_pera_gris,'canny',0.5);  
ima_pera_BIN = ~ima_pera_BIN;
```

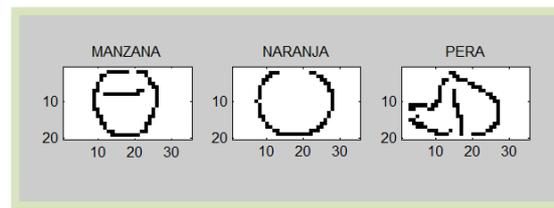


Figura 4. Extracción de bordes tipo ‘canny’

Después se procedió a convertir las matrices en vectores, de tamaño “700 x 1”, estos vectores representarán cada una de las imágenes a reconocer. Para esto se utiliza lo siguiente:

```
%Generamos los vectores correspondientes a cada imagen  
v_manzana = reshape(ima_manzana_BIN,700,1);  
v_naranja = reshape(ima_naranja_BIN,700,1);  
v_pera = reshape(ima_pera_BIN,700,1);
```

Una vez obtenidos los vectores, se creó la matriz que representa cada una de las imágenes a reconocer. Sin embargo, aún no tenemos preparadas las entradas debido a que al extraer los bordes nos arroja una matriz lógica que no



“CONGRESO INTERNACIONAL DE INVESTIGACIÓN E INNOVACIÓN 2014”

Multidisciplinario

10 y 11 de abril de 2014, Cortazar, Guanajuato, México

ISBN: 978-607-95635

puede ser leída por la red, a lo que se procede a convertirla en una matriz binaria, como se muestra a continuación.

```
Entradas = [v_manzana v_naranja v_pera];
Entradas = double(Entradas);
```

Posteriormente se definieron las clases con 1, 2 y 3 para cada una de las representaciones de las imágenes:

1 = Manzana, 2 = Naranja, 3 = Pera

Para obtener el vector objetivo:

```
Objetivo = [1 2 3];
```

Ahora sí, se tienen las condiciones necesarias para crear nuestra red:

```
RNA = newff(minimax(Entradas),[3,1],{'logsig','purelin'});
```

Como primera instancia de este reconocimiento, se procedió a hacer una simulación sobre la red creada para ver el comportamiento de ella y analizar las salidas de dicha simulación

```
Simulacion_1 = sim(RNA, Entradas)
```

En la figura 5 se pueden observar los resultados que arroja nuestra primera simulación y apreciar que los datos no son los esperados.

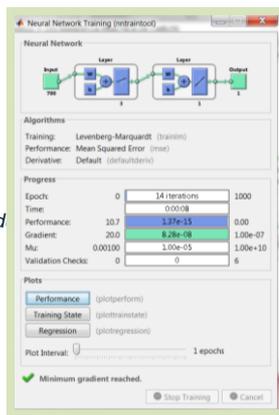
Simulacion_1 <1x3 double>						
	1	2	3	4	5	6
1	-0.6257	-1.3298	-1.2903			

Figura 5. Resultados de la primera simulación, sin entrenar la red.

Por lo que se procedió a entrenar la red relacionando las entradas y los objetivos respectivamente:

```
RNA = train(RNA, Entradas, Objetivo);
```

Se puede observar en la figura 6 el entrenamiento en la Toolbox.



“CONGRESO INTERNACIONAL DE INVESTIGACIÓN E INNOVACIÓN 2014”
 Multidisciplinario
 10 y 11 de abril de 2014, Cortazar, Guanajuato, México
 ISBN: 978-607-95635

Figura 6. Lanzamiento del Toolbox de Matlab, de entrenamiento de la red. Mediante la Toolbox, es posible extraer información importante del comportamiento de la red, como se puede apreciar en la figura 7.

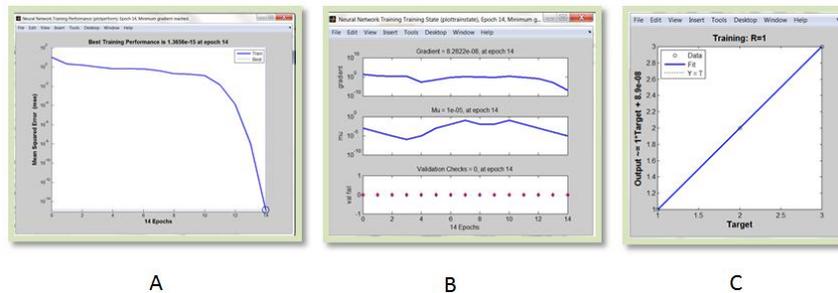


Figura 7. Gráficas integradas en la Toolbox de Matlab.

Se puede observar en la gráfica “A” que se obtiene el mejor rendimiento o aprendizaje de la red con 1.36 mse (error cuadrático medio) en 14 épocas (ciclos de aprendizaje de la red). En la gráfica “B” el gradiente tiende a bajar, mientras que en la gráfica “C”, la interpolación de los datos con la salida de la red, se puede apreciar que están totalmente correlacionados con los objetivos.

Se realizó nuevamente la simulación para observar los resultados arrojados por la red y compararlos nuevamente con las entradas que se muestran en la figura 8.

$$\text{Simulacion_2} = \text{sim}(\text{RNA}, \text{Entradas})$$

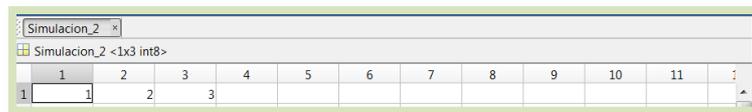


Figura 8. Simulación de la red con entrenamiento

“CONGRESO INTERNACIONAL DE INVESTIGACIÓN E INNOVACIÓN 2014”

Multidisciplinario

10 y 11 de abril de 2014, Cortazar, Guanajuato, México

ISBN: 978-607-95635

Se puede ver claramente que la simulación realizada arroja ya los objetivos deseados, como se puede observar en la figura 9, que contiene las entradas y salidas a la red.

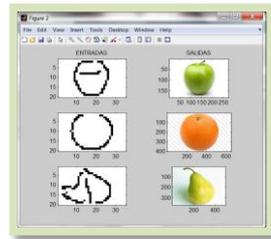


Figura 9. Imágenes reconocidas.

5. Conclusiones

De acuerdo al objetivo propuesto en este módulo, se tiene el resultado del prototipo con tendencia a mejorar, de tal manera que exponga cada uno de los pasos que se deben seguir para aprender el área de redes neuronales; si bien es cierto, que existen otros tipos de redes como la de perceptron, la de adaline/madeline, o bien la de hopfield, este es un paso inicial para que el alumno vea las potencialidades que se tienen de esta área y como trabajo futuro, se tiene proyectado terminar el módulo completo de tal manera que se ponga en marcha el proyecto integral e incentivar al alumno con estas herramientas docentes a que pregunte y participe en clases, a dar respuesta a las dudas que surjan al realizar las prácticas, y que supongan una motivación para un estudio y profundización del área de inteligencia artificial.

- [1] <http://weka.waikato.ac.nz>
- [2] Sierra, B., Aprendizaje Automático: conceptos básicos y avanzados, Editorial Pearson, Prentice Hall, pp.101-130, 2006.
- [3] Shalkoff, Robert J. Neural networks (Computer science), McGraw-Hill (New York), xxi, 422 p., 1997.
- [4] Rosenblatt, F. Principles of neurodynamics Spartan, 1962.
- [5] Widrow, B., Hopf. M. Adaptative Switching Circuits. IRE WESCON Convention Record, part 4.96, 104(1960).



“CONGRESO INTERNACIONAL DE INVESTIGACIÓN E INNOVACIÓN 2014”

Multidisciplinario

10 y 11 de abril de 2014, Cortazar, Guanajuato, México

ISBN: 978-607-95635

- [6] Hopfield, J.J.: "Neural Networks and Physical Systems with Emergent Collective Computational Abilities". En Proceedings of the National Academy of Sciences, USA 79, 2554-2558, 1982.
- [7] Rumelhart, D. Hinton, G. & Williams, R. "Learning representations by backpropagation errors". Nature. 1986.
- [8] Werbos, P.J. Beyond Regression: New tools for prediction and Analysis in the behavioral Sciences, PhD thesis, Harvard university, Cambridge, MA., 1974