



Title: Algoritmo Perceptrón de Reconocimiento de Imágenes para Detección de Problemas en Cultivos de la Uva

Author: Luis Fernando, RUIZ-AVILA, José Luis, SIERRA-LEYVA, Jhonnathan
Alexis, FUENTES-VÁZQUEZ, Cinthia Valeria, MENDIVIL-REYES

Editorial label ECORFAN: 607-8534
BCIERMMI Control Number: 2018-03
BCIERMMI Classification (2018): 251018-0301

Pages: 13
RNA: 03-2010-032610115700-14

ECORFAN-México, S.C.

244 – 2 Itzopan Street
La Florida, Ecatepec Municipality
Mexico State, 55120 Zipcode
Phone: +52 | 55 6159 2296
Skype: ecorfan-mexico.s.c.
E-mail: contacto@ecorfan.org
Facebook: ECORFAN-México S. C.

Twitter: @EcorfanC

www.ecorfan.org

Holdings

Mexico	Colombia	Guatemala
Bolivia	Cameroon	Democratic
Spain	El Salvador	Republic
Ecuador	Taiwan	of Congo
Peru	Paraguay	Nicaragua



Antecedentes

En la actividad agrícola se presentan daños causados por enfermedades, falta o exceso de humedad, mal control de plagas, entre otros factores.

Al tener el conocimiento de los problemas es posible redimensionar en datos numéricos estos diversos problemas, en nuestro caso en patrones de color y textura que pueden ser obtenidos de una imagen.



Antecedentes

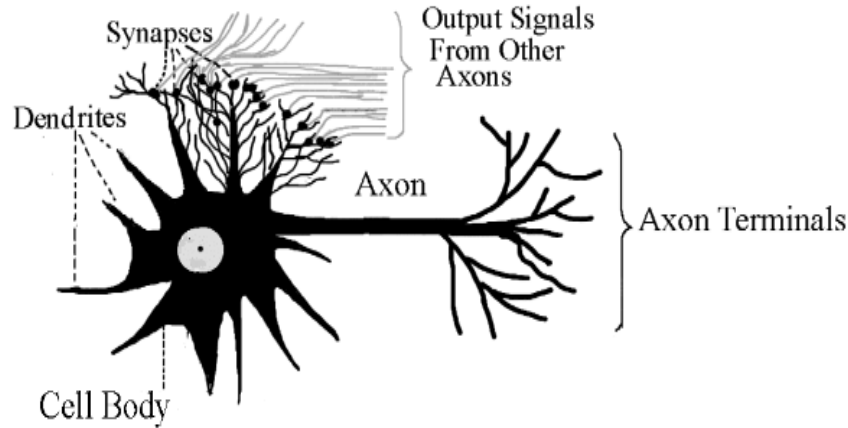


Figura 1. *Neurona Biológica.* Recuperado de: *Intelligent Control Systems...* Pág. 43

La neurona por el proceso de sinapsis es capaz de emitir una respuesta por axones, en analogía, una función matemática con ciertas entradas es capaz de producir diversas respuestas. (Ramesh, Rangachar, & Brian G., 1995).



Antecedentes

La visión artificial por computadora en base a redes neuronales, modela mediante técnicas matemáticas la forma tradicional y apariencia de los objetos para recuperar la forma física y de color de los objetos (González G. F., 2009).

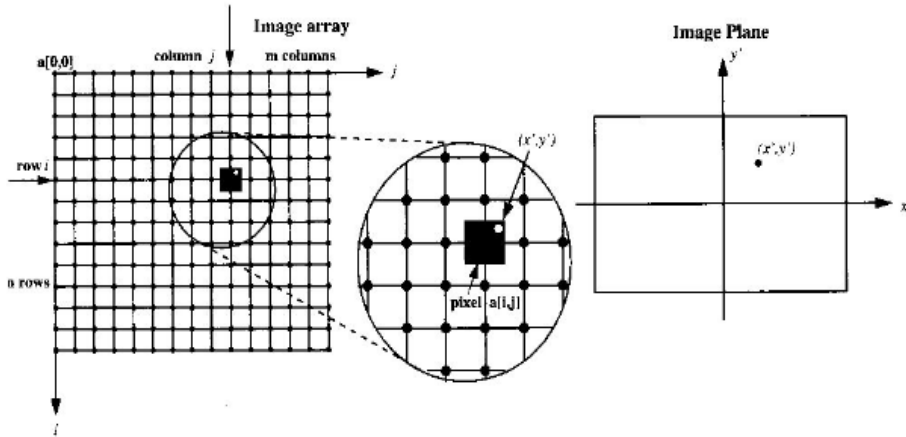


Figura 2. Visión de máquina. Recuperado de *Relationship between image plane and image array indices*.



Antecedentes

$$n = x_1w_1 + x_2w_2 + x_3w_3 + \dots + x_iw_i$$

Un Perceptrón multicapa tiene tres diferentes zonas en las cuales procesa la información de diferentes maneras, en las capas de entrada, se reciben las variables de entrada, capas ocultas donde ocurre la mayor parte del procesamiento y la capa de salida que es la obtención de resultados. (García, 2006)

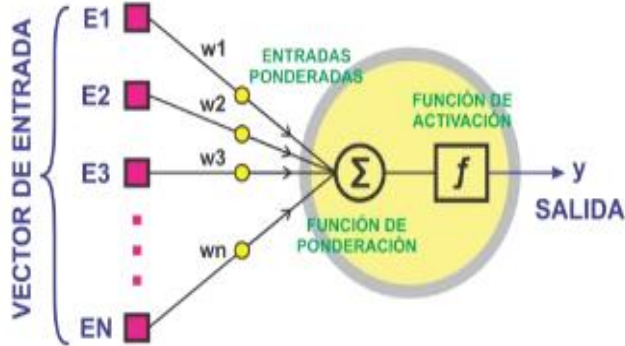


Figura 3. Modelo De Neurona Artificial. Recuperado de: Redes Neuronales Artificiales. Morse Blas, et al. Santa Fe 2015



Metodología

Un Perceptrón multicapa tiene tres diferentes zonas en las cuales procesa la información de diferentes maneras, en las capas de entrada, se reciben las variables de entrada, capas ocultas donde ocurre la mayor parte del procesamiento y la capa de salida que es la obtención de resultados. (García, 2006)

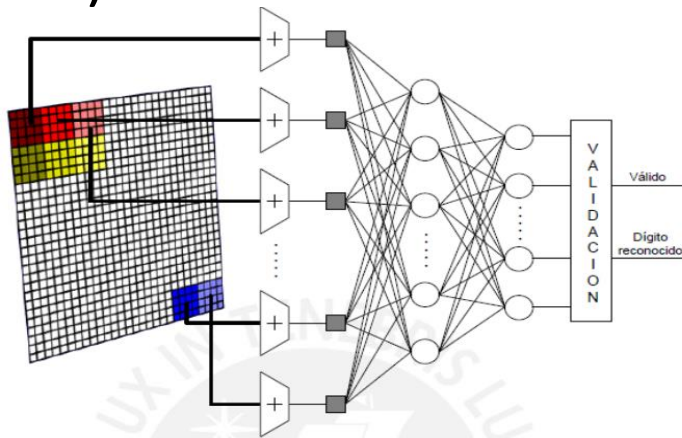


Figura 4. Representación De Colores. Recuperado de: (Hernández, 2014)

```
for (int i = 0; i < mapa.Height; i++)
{
    for (int j = 0; j < mapa.Width; j++)
    {
        colores = mapa.GetPixel(j, i);
        f.Add(new int[] {colores.R,colores.G,colores.B});
        f2.Add(new int[] { Convert.ToInt32(mapa.GetPixel(j, i).GetSaturation()),
            Convert.ToInt32(mapa.GetPixel(j, i).GetBrightness()),
            Convert.ToInt32(mapa.GetPixel(j, i).GetHue()) });
        Color nuevo = colores;
        nuevo= Color.FromArgb(205,nuevo);
        nuevo = Color.FromArgb(100,nuevo.B,nuevo.B,nuevo.B);
        mapa.SetPixel(j, i, nuevo);
        mapa.SetResolution(10f,10f);
        mapa.MakeTransparent();
    }
}
```

Figura 5. Obtención de color natural de la imagen, para la aplicación de técnicas de procesamiento de donde formaremos matrices de datos.



Metodología

Tratamiento sistemático de los pixeles de la imagen con filtros y transformaciones geométricas para realzar datos específicos, incluyen la mejora del contraste, eliminar el brillo, iluminación y restauración de la imagen (La Serna Palomino & Román Concha, 2009)

	B	A	B	
	A	p	A	
	B	A	B	

Figura 6. Vecindario del punto p. Recuperado de: *Visión Artificial*. José F. Vélez S., et al. Madrid. 2003

$$q = q \in N_4(p)$$

$$(x - 1y) \text{ o } (xy - 1) \text{ o } (x + 1y) \text{ o } (xy + 1)$$

$$q = q \in N_8(p)$$

$$(x - 1y) \text{ o } (xy - 1) \text{ o } (x + 1y) \text{ o } (xy$$



Metodología

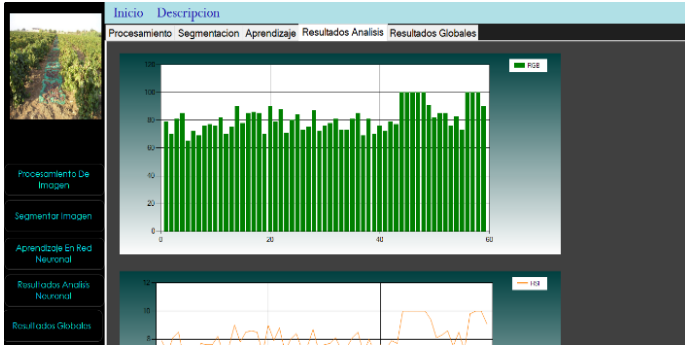


Figura 8. Resultados de datos obtenidos por patrón de color.

$$F = \{f(0), (f1), \dots, f(N - 1)\}$$

$$I_c(n, m) = F(I(x, y))$$

$$= \frac{1}{N} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} I(x, y) e^{-j^2 \frac{\pi x n}{N}} e^{-j^2 \frac{\pi m y}{N}}$$

$$V_{n,m} = 0, 1., N - 1$$

La segmentación de imágenes a partir transformada de Fourier de imágenes digitales con los datos obtenidos y registrados en las matrices de datos nos permitió formar un modelo matemático de los datos más completo de las capturas ya tomadas, en escalas de grises.



Metodología

- Una vez obtenidos los datos se crea un patrón, clasificando los patrones de acuerdo a si tienen la afección distintiva, al ser procesada, la neurona definirá el patrón de error y la correlación con la función de activación definida por la constante variante de umbral en cada neurona generalizando y disminuyendo el error hasta un mínimo aceptable.

Pixel	RGB	HSL	HSB	CMYK	Porcentaje RGB	Frecuencias Relativas	Cambios De Tipo En Pixeles
0.0	(53.17.8)	(12.73%.12%)	(19.100%.21%)	(0%.68%.85%.79%)	0.23%	0.25%	0.14%

Figura 8. Imagen representativa de resultados obtenidos en la cuarta fase.



Metodología

Este algoritmo se basa en la función de error, arroja datos en una curva, que deben alcanzar un mínimo y un máximo de error, el propósito, mediante el descenso del gradiente, es evitar mínimos locales de la curva. (Anil K., Jianchang, & K. M., 1996).

$$f(x) = \frac{1}{1 + e^{-ax}} \quad 0 \leq f(x) \leq 1$$

$$\delta_i^L = f'(net_i^L) [d_i^L - o_i^L]$$

$$f(x) = f(x) [1 - f(x)]$$

$$\delta_j^L = [z_j^{(q)} (1 - z_j^{(q)})] (t_j^{(q)} - z_j^{(q)})$$

$$\delta_i^L = f'(net_i^L) \sum_k v_{ik}^{l+1} \delta_k^{l+1}$$

```

public bool AprendizajeBackPropagacion(List<double[]> input, List<double[]> desiredOutput, double alpha, double maxError, int maxIterations)
{
    double err = 99999;
    log = new List<string>();
    while (err > maxError)
    {
        maxIterations--;
        if (maxIterations <= 0)
        {
            return false;
        }

        if (Console.KeyAvailable)
        {
            System.IO.StreamWriter w = new StreamWriter(@"C:\Users\User\Desktop\bin\Debug\log.txt");

            string pasar = "";
            foreach (var item in log)
            {
                pasar = item + "\n";
                w.WriteLine(pasar);
            }
            return true;
        }

        AplicarBackPropagacion(input, desiredOutput, alpha);
        err = ErrorGenerico(input, desiredOutput);
        log.Add(err.ToString());
        Console.WriteLine(err);
    }

    System.IO.File.WriteAllLines(@"C:\Users\User\Desktop\bin\Debug\logtail.txt", log.ToArray());
}

```

Figura 12. Método genérico para actualizar valores en las funciones de cada neurona de la red neuronal.

```

static void LeerPatrones()
{
    string data = System.IO.File.ReadAllText(dataPath).Replace("\n", "");
    string[] row = data.Split(Environment.NewLine.ToCharArray());
    for (int i = 0; i < row.Length; i++)
    {
        string[] rowData = row[i].Split(',');

        double[] inputs = new double[inputCount];
        double[] outputs = new double[outputCount];

        for (int j = 0; j < rowData.Length; j++)
        {
            if (j < inputCount)
            {
                inputs[j] = Normalizacion(double.Parse(rowData[j]), inputMin, inputMax);
            }
            else
            {
                outputs[j - inputCount] = Normalizacion(double.Parse(rowData[j]), outputMin, outputMax);
            }
        }

        input.Add(inputs);
        output.Add(outputs);
    }
}

```

Figura 10. Obtención de valores de los archivos.





Metodología

- Se procede a iterar con los campos y con el algoritmo de backpropagation para determinar un nuevo aprendizaje a partir de detectar máximos y mínimos en la evaluación de los datos generalizados en las distintas neuronas. Así disminuyendo el grado de error para las capas de neuronas internas y a su vez mejorando el conocimiento.

```

1 referencia
void AplicarBackPropagation(List<double> input, List<double> desiredOutput, double alpha)
{
    GuardarDeltas();
    for (int i = 0; i < input.Count; i++)
    {
        Activacion(input[i]);
        GuardarSigmas(desiredOutput[i]);
        ActualizarBias(alpha);
        AgregarDeltas();
    }
    ActualizarPesos(alpha);
}

```

Figura 13. Aplicación del algoritmo.

```

1 referencia
double ErrorIndividual(double[] realOutput, double[] desiredOutput)
{
    double err = 0;
    for (int i = 0; i < realOutput.Length; i++)
    {
        err += Math.Pow(realOutput[i] - desiredOutput[i], 2);
    }
    return err;
}

1 referencia
double ErrorGenerico(List<double> input, List<double> desiredOutput)
{
    double err = 0;
    for (int i = 0; i < input.Count; i++)
    {
        err += ErrorIndividual(Activacion(input[i]), desiredOutput[i]);
    }
    return err;
}

```

Figura 14. Calculo del error.

```

void ActualizarPesos(double alpha)
{
    for (int i = 0; i < Capas.Count; i++)
    {
        for (int j = 0; j < Capas[i].numberOfNeurons; j++)
        {
            for (int k = 0; k < Capas[i].neurons[j].weights.Length; k++)
            {
                Capas[i].neurons[j].weights[k] -= alpha * deltas[i][j, k];
            }
        }
    }
}

```

Figura 14. Actualización de pesos.



Resultados

Los resultados obtenidos en la simulación de aprendizaje del Perceptrón y en la utilización del programa, fueron satisfactorios, al probarlo con imágenes a un cultivo de uva, arrojó con éxito la diferencia entre fotografías de hojas que padecían necrosis en la epidermis y las que no lo tenían, con una eficiencia del 95 %.

datos: Bloc de notas				
Archivo	Edición	Formato	Ver	Ayuda
4.9,3.1,1.5,0.1,0				
5.0,3.2,1.2,0.2,0				
5.5,3.5,1.3,0.2,0				
4.9,3.1,1.5,0.1,0				
4.4,3.0,1.3,0.2,0				
5.1,3.4,1.5,0.2,0				
5.0,3.5,1.3,0.3,0				
4.5,2.3,1.3,0.3,0				
4.4,3.2,1.3,0.2,0				
5.0,3.5,1.6,0.6,0				
5.1,3.8,1.9,0.4,0				
4.8,3.0,1.4,0.3,0				
5.1,3.8,1.6,0.2,0				
4.6,3.2,1.4,0.2,0				
5.3,3.7,1.5,0.2,0				

5.0,3.3,1.4,0.2,0
7.0,3.2,4.7,1.4,1
6.4,3.2,4.5,1.5,1
6.9,3.1,4.9,1.5,1
5.5,2.3,4.0,1.3,1
6.5,2.8,4.6,1.5,1
5.7,2.8,4.5,1.3,1
6.3,3.3,4.7,1.6,1
4.9,2.4,3.3,1.0,1
6.6,2.9,4.6,1.3,1
5.2,2.7,3.9,1.4,1
5.0,2.0,3.5,1.0,1
5.9,3.0,4.2,1.5,1
6.0,2.2,4.0,1.0,1
6.1,2.9,4.7,1.4,1
5.6,2.9,3.6,1.3,1
6.7,3.1,4.4,1.4,1

Figura 9. Archivo datos.txt de una de las pruebas obtenidas

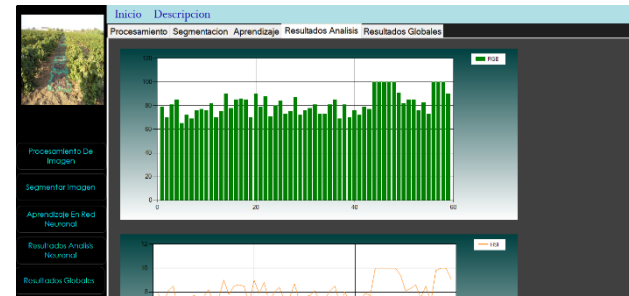


Figura 8. Resultados de datos obtenidos por patrón de color.



Conclusiones

- A través de la investigación desarrollada, en la simulación del prototipo de la implementación del software, pruebas concluyentes a partir de imágenes tomadas a cultivos de uva se obtuvo la disminución del error, obtenido de 2.123 que resulta de la evaluación de la función cálculo de error con el aprendizaje obtenido en 30 pruebas, se redujo a 0.70 el grado de error lo que da una aceptabilidad del 93% para el algoritmo, que puede disminuir mucho más a través del aprendizaje.

```
0,709669736987337
0,709669731248186
0,709669725509032
0,709669719769872
0,709669714030711
0,709669708291545
0,709669702552374
0,7096696968132
0,709669691074021
0,709669685334839
0,709669679595653
0,709669673856464
0,709669668117268
0,709669662378071
0,709669656638868
0,709669650899663
```

Figura 7. *Disminución del grado de error*



Conclusiones

Herramientas como estas son fiables y seguras en su utilización, su aplicabilidad es basta en diversos cultivos de uva, al poder determinar si la planta tiene plagas en nuestro caso necrosis en la epidermis, seria de vital importancia para el productor en etapas tempranas de desarrollo, para poder combatir la plaga.

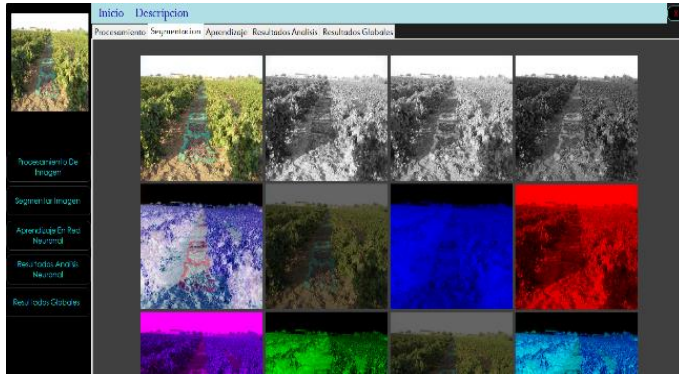


Figura 7. Captura. Etapa De Segmentación.



ECORFAN®

© ECORFAN-Mexico, S.C.

No part of this document covered by the Federal Copyright Law may be reproduced, transmitted or used in any form or medium, whether graphic, electronic or mechanical, including but not limited to the following: Citations in articles and comments Bibliographical, compilation of radio or electronic journalistic data. For the effects of articles 13, 162,163 fraction I, 164 fraction I, 168, 169,209 fraction III and other relative of the Federal Law of Copyright. Violations: Be forced to prosecute under Mexican copyright law. The use of general descriptive names, registered names, trademarks, in this publication do not imply, uniformly in the absence of a specific statement, that such names are exempt from the relevant protector in laws and regulations of Mexico and therefore free for General use of the international scientific community. BCIERMMI is part of the media of ECORFAN-Mexico, S.C., E: 94-443.F: 008- (www.ecorfan.org/ booklets)