UNIVERSIDAD NACIONAL MAYOR DE SAN MARCOS
FACULTAD DE INGENIERÍA DE SISTEMAS E INFORMATICA

Reconocimiento de placas de autos mediante redes neuronales utilizando el algoritmo de aprendizaje RPROP

CURSO

Redes Neuronales

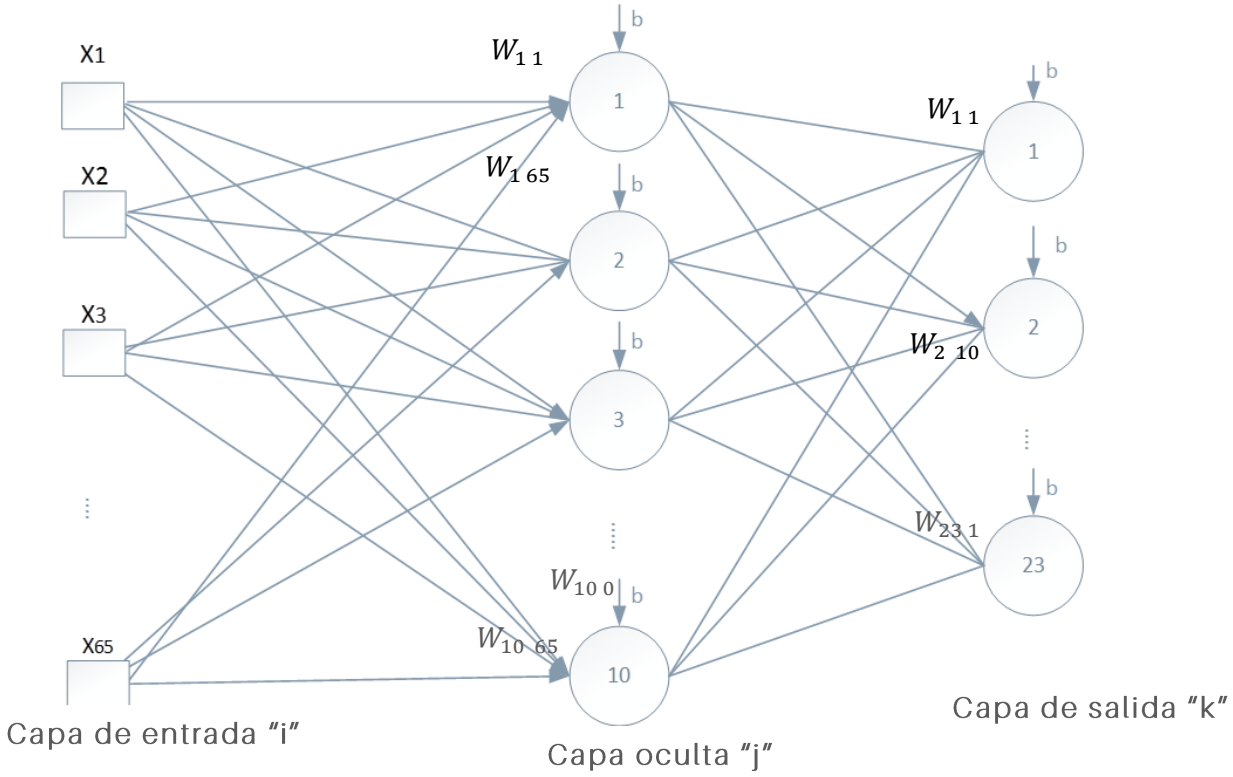
PROFESOR

Rolando A. Maguiña Pérez

INTEGRANTES	CODIGO
Gian Carlos Huachin Sairitupac	11200117
Jorge Luis De la Cruz Espinoza	11200065

Síntesis de la neurona

TOPOLOGIA DE LA RED



La primera capa es representada por las "i" la segunda por "j" y la última por "k".

Función sigmoide simétrica:

$$f(x) = \beta\left(\frac{1 - e^{-\alpha x}}{1 + e^{-\alpha x}}\right)$$

Donde:

$\alpha = 1 \quad \beta = 1$

$$f(x) = \frac{1 - e^{-x}}{1 + e^{-x}}$$

REGLA DE PROPAGACION

$$u_j = \sum_{0}^{i} x_i w_{ji}$$

$$f(u_j) = O_j$$

$$u_k = \sum_{0}^{j} O_j w_{kj}$$

Reemplazando:

$$u_k = \sum_{0}^{j} f(\sum_{0}^{i} x_i w_{ji}) w_{kj}$$

REGLA DE ACTIVACION

$$f(u) = \frac{1 - e^{-u}}{1 + e^{-u}} = \frac{2}{1 + e^{-u}} - 1$$

REGLA DE EQUIVALENCIA

$$y_k = f(u_k)$$

REGLA DE APRENDIZAJE

$$w_{ji}^{(t+1)} = w_{ji}^{(t)} + \Delta w_{ji}^{(t)}$$

$$\Delta w_{ji}^{(t)} \begin{cases} -\Delta_{ji}^{(t)}, & si \quad \frac{\partial E}{\partial w_{ji}}^{(t)} > 0 \\ +\Delta_{ji}^{(t)}, & si \quad \frac{\partial E}{\partial w_{ji}}^{(t)} < 0 \\ 0, & si \quad \frac{\partial E}{\partial w_{ji}}^{(t)} = 0 \end{cases}$$

$$\Delta_{ji}^{(t)} \begin{cases} \eta^+ * \Delta_{ji}^{(t-1)}, & si \quad \frac{\partial E}{\partial w_{ji}}^{(t-1)} * \frac{\partial E}{\partial w_{ji}}^{(t)} > 0 \\ \eta^- * \Delta_{ji}^{(t-1)}, & si \quad \frac{\partial E}{\partial w_{ji}}^{(t-1)} * \frac{\partial E}{\partial w_{ji}}^{(t)} < 0 \\ \Delta_{ji}^{(t-1)}, & si \quad \frac{\partial E}{\partial w_{ji}}^{(t-1)} * \frac{\partial E}{\partial w_{ji}}^{(t)} = 0 \end{cases}$$

Donde:

$$\eta^+ = 1.2$$
$$\eta^- = 0.5$$
$$\Delta_0 = 0.1$$

ALGORITMO DE APRENDIZAJE

For all weights and biases{
  if $(\frac{\partial E}{\partial w_{ij}}(t-1) * \frac{\partial E}{\partial w_{ij}}(t) > 0)$ **then** {
    $\triangle_{ij}(t) =$ **minimum** $(\triangle_{ij}(t-1) * \eta^+, \triangle_{max})$
    $\triangle w_{ij}(t) =$ **-** **sign** $(\frac{\partial E}{\partial w_{ij}}(t)) * \triangle_{ij}(t)$
    $w_{ij}(t+1) = w_{ij}(t) + \triangle w_{ij}(t)$
  }
  else if $(\frac{\partial E}{\partial w_{ij}}(t-1) * \frac{\partial E}{\partial w_{ij}}(t) < 0)$ **then** {
    $\triangle_{ij}(t) =$ **maximum** $(\triangle_{ij}(t-1) * \eta^-, \triangle_{min})$
    $w_{ij}(t+1) = w_{ij}(t) - \triangle w_{ij}(t-1)$
    $\frac{\partial E}{\partial w_{ij}}(t) = 0$
  }
  else if $(\frac{\partial E}{\partial w_{ij}}(t-1) * \frac{\partial E}{\partial w_{ij}}(t) = 0)$ **then** {
    $\triangle w_{ij}(t) =$ **-** **sign** $(\frac{\partial E}{\partial w_{ij}}(t)) * \triangle_{ij}(t)$
    $w_{ij}(t+1) = w_{ij}(t) + \triangle w_{ij}(t)$
  }
}

Fuente: A Direct Adaptive Method for Faster Backpropagation Learning:
The RPROP Algorithm
Martin Riedmiller Heinrich Braun
Institut fur Logik, Komplexitat und Deduktionssyteme
University of Karlsruhe

Desarrollo para hallar $\frac{\partial E}{\partial w_{kj}}$:

$$E = \frac{1}{2}\sum_{1}^{k}(t_k - Y_k)^2$$

Reemplazamos la regla de cadena en ∂u_k

$$\frac{\partial E}{\partial w_{kj}} = \frac{\partial E}{\partial u_k} * \frac{\partial u_k}{\partial w_{kj}} = \frac{\partial E}{\partial u_k} * \frac{\partial \sum w_{kj} * y_j}{\partial w_{kj}}$$

Reemplazamos $u_k => \sum w_{kj} * y_i$

$$\frac{\partial E}{\partial u_k} * y_i = \frac{\partial E}{\partial y_k} * \frac{\partial y_k}{\partial u_k} * y_j$$

$$\delta_k = \frac{\partial E}{\partial u_k}$$

Reemplazamos la regla de cadena en ∂y_k

$$\frac{\partial E}{\partial y_k} * \frac{\partial f(u_k)}{\partial u_k} * y_j$$

Reemplazamos y_k => $f(u_k)$

$$\frac{\partial \frac{1}{2}\sum(t_k - Y_k)^2}{\partial y_k} * f'(u_k) * y_j \ldots \text{ (a)}$$

Reemplazamos E

$$\frac{\partial \frac{1}{2}\sum(t_k - Y_k)^2}{\partial y_k} = -(t_k - Y_k) \ldots \text{ (b)}$$

Agregamos (b) en (a):

$$\frac{\partial E}{\partial w_{kj}} = -(t_k - Y_k) * f'(u_k) * y_j$$

$$\delta_k = \frac{\partial E}{\partial u_k} = -(t_k - Y_k) * f'(u_k) \qquad \dots (\theta)$$

Podemos derivar

$$f(u) = \frac{1 - e^{-u}}{1 + e^{-u}} = \frac{2}{1 + e^{-u}} - 1$$

$$f'^{(u)} = 2 * f(u) * (1 - f(u))$$

Desarrollo para hallar $\frac{\partial E}{\partial w_{ji}}$:

$$E = \frac{1}{2}\sum_{1}^{k}(t_k - Y_k)^2$$

Reemplazamos la regla de cadena en ∂u_k

$$\frac{\partial E}{\partial w_{ji}} = \frac{\partial E}{\partial u_j} * \frac{\partial u_j}{\partial w_{ji}} = \frac{\partial E}{\partial u_j} * \frac{\partial \sum w_{ji} * x_i}{\partial w_{ji}}$$

Reemplazamos $u_j => \sum w_{ji} * x_i$

$$\frac{\partial E}{\partial u_j} * x_i = \frac{\partial E}{\partial y_j} * \frac{\partial y_j}{\partial u_j} * x_i$$

Reemplazamos la regla de cadena en ∂y_j

$$\frac{\partial E}{\partial y_j} * \frac{\partial f(u_j)}{\partial u_j} * x_i$$

Reemplazamos y_j => $f(u_j)$

$$\frac{\partial E}{\partial y_j} * f'(u_j) * x_i \ \ldots \ (c)$$

Resolvemos $\frac{\partial E}{\partial y_j}$

$$\frac{\partial E}{\partial y_j} = \sum_k (\frac{\partial E}{\partial u_k} * \frac{\partial u_k}{\partial y_j})$$

$$\sum_k (\frac{\partial E}{\partial u_k} * \frac{\partial \sum w_{kj} y_j}{\partial y_j}) = \sum_k (\frac{\partial E}{\partial u_k} * w_{kj})$$

Reemplazamos $\boldsymbol{\theta}$

$$\delta_k = \frac{\partial E}{\partial u_k} = -(t_k - Y_k) * f'(u_k)$$

$$\sum_k (\frac{\partial E}{\partial u_k} * w_{kj}) = \sum_k (-(t_k - Y_k) * f'(u_k) * w_{kj})$$

$$\frac{\partial E}{\partial y_j} = \sum_k (-(t_k - Y_k) * f'(u_k) * w_{kj}) \ \ldots \ (d)$$

Reemplazamos $(\mathbf{d})$ **en** $(\mathbf{c})$:

$$\frac{\partial E}{\partial y_j} * f'(u_j) * x_i$$

$$\sum_k (-(t_k - Y_k) * f'(u_k) * w_{kj}) * f'(u_j) * x_i$$

$$\frac{\partial E}{\partial W_{jI}} = \sum_k (-(t_k - Y_k) * f'(u_k) * w_{kj}) * f'(u_j) *$$

DESARROLLO DE LA APLICACIÓN

1 SEGEMENTACION

Se captura la imagen, luego lo procesamos para resaltar las caracteristicas para poder calcular las posibles regiones que puedan contener la placa, por ultimo se hace un procesamiento extra para asegurar que las regiones sea correspondientes al de una placa.Al termino del procesamiento se corta y almacena dicha region.

2 SVM

A partir de las regiones cortadas y almacenadas, donde algunas pueden contener placas y otras no, se implementa un algoritmo para obtener las caracteristicas de estas regiones y poder almacenarlas en un archivo .XML.Posteriormente en la aplicacion Final principal se utiliza un SVM para poder diferenciar entre las regiones que contienen placa y las que no contienen, utilizando el archivo XML.

3 OCR

En esta etapa se segmenta los simbolos de la region reconocida como placa, se hace un procesamiento para resaltar dichos simbolos, luego se verfica si el area, que enmarca dichos simbolos, se encuentra entre los limites definidos.Despues se realiza un corte del area y se almacena en un tamaño 20x20 pixeles.

Por ultimo se a partir de los simbolos almacenados se extraen sus caracteristicas y se guardan en un archivo XML.

4 TRAIN OCR

En esta etapa se define la Red Neuronal, su topologia,funcion de activacion,algoritmo de aprendizaje, entre otros. Despues pasamos a la etapa de entrenamiento donde utlizamos como data de entrada las caracterisiticas almacenadas en el archivo XML de la etapa del OCR.

Por ultimo una vez entrenada la red neuronal se define un algoritmo para la predicion del simbolo a partir de la RN.

5 INTEGRACION Y APP FINAL

En esta etapa juntamos todos los algoritmos y funciones definidos anteriormente, de tal manera que podamos ingresar una imagen nueva y poder reconocer la placa que se encuentra en ella.

SEGMENTACION

1°Procesamiento

Se transforma la imagen a escala de grises, luego se realza las lineas verticales y horizontales.Posteriormente se binariza y se hace un barrido de las lineas tanto vertical como horizontal quedando formas en la imagen en blanco y negro.

2°Procesamiento

A partir de las ubicaciones de las regiones obtenidas anteriormente, se utiliza un metodo de realze de caracteristicas usando "semillas" de opencv en las cuales a partir del centro de la region va coloreando del mismo color pixel por pixel que se encuentra juntos. generando una imagen, donde se calcula su area y de nuevo se verifica si esta en los limites

Obtencion de regiones

A partir de las formas se obtiene los contornos de las formas para luego calcular su area y verificar si se encuentran dentro de los limites definidos.En esta fase se hace un primer filtro de regiones de acuerdo a su area.

Corte y Almacenamiento

Las nuevas regiones filtradas se calcula el rectangulo que encierra la region como tambien el angulo de inclinacion, posteriormente se corta el rectangulo de la region y luego se redimensiona a un tamaño de 144x33 Pixeles .
Una vez cortada la region se almacena, para luego sirva de entrada a la siguiente fase llamada SVM

Carácteristicas Para SVM

Segmentación Caracteres

Características Para SVM

Argumentos: 18 20 tmp/plate tmp/noplate

Lectura imagenes

En esta fase se define los siguientes argumentos:
-Numero de Placas
-Numero de No Placas
-Nombre de archivos Placas.
-Nombre de archivos No Placas
Todo esto con la finalidad de leer las imagenes iterativamente.

Transformacion

Se transforma las imagenes a un solo canal y una fila tanto las correspodientes a placas como a las No placas y se almacena todas en una sola variable, al mismo tiempo tambien se almacena etiquetas respectivamente al orden de las imagenes transformadas para una facil ubicacion e identificacion.

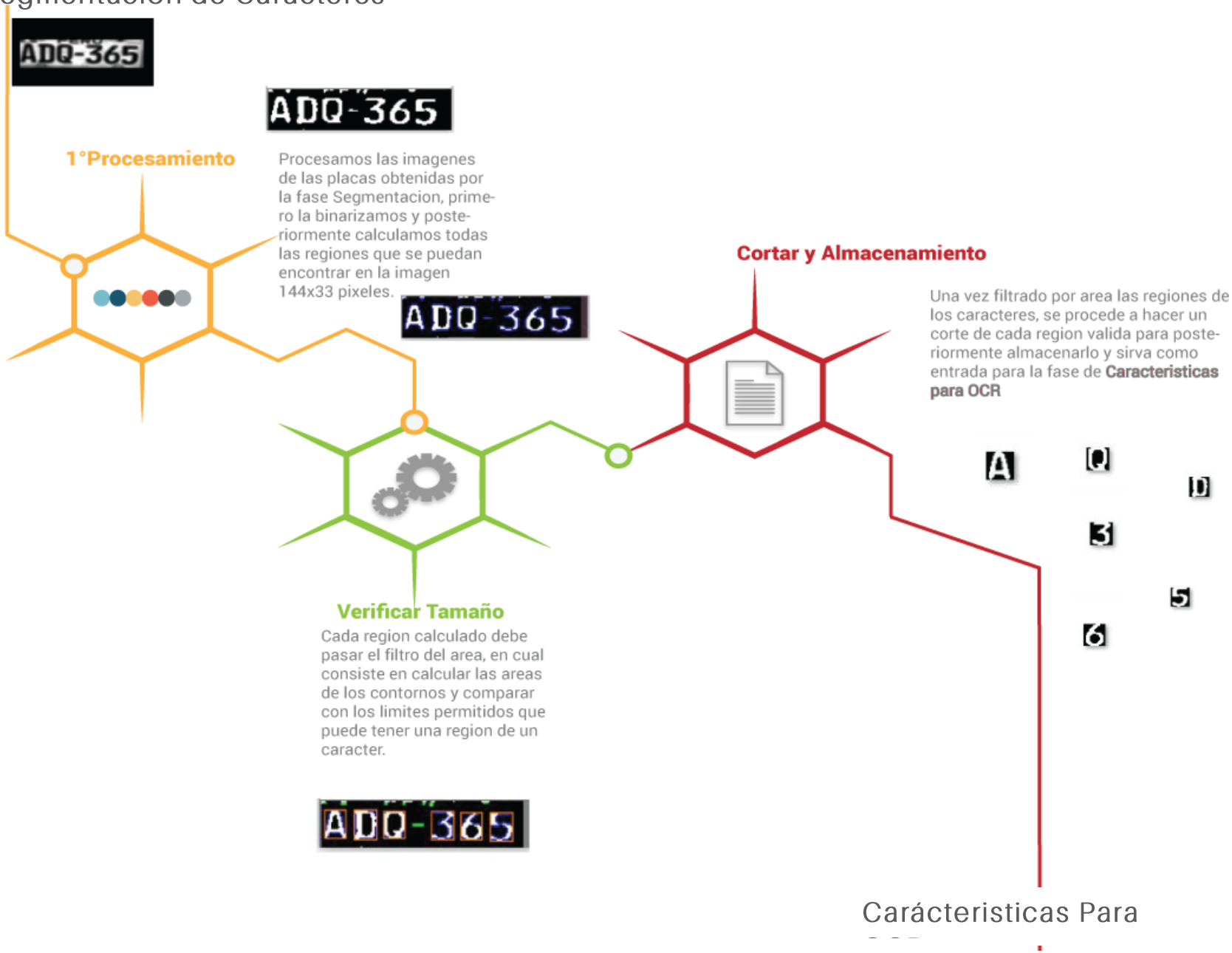
Almacenamiento

Las variables que contienen las imagenes transformadas y las etiquetas correspondientes, todo esto viene a ser las caracteristicas de entrada para la utilizacion del SVM en el programa principal, por lo tanto se procede a almacenarlas en el archivo **SVM.xml**

SVM. xml

Aplicación Final

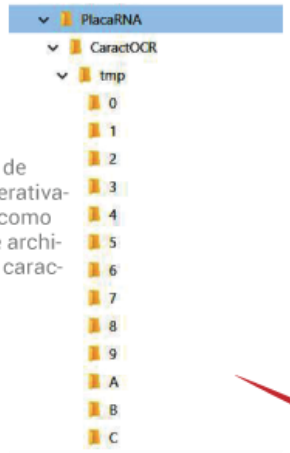
Segmentación de Carácteres



Características Para OCR

Lectura Imagenes

Se indica la direccion de donde se van a leer iterativamente las imagenes, como tambien el numero de archivos que hay por cada caracter.



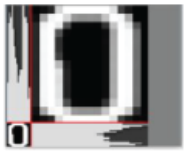
Almacenamiento

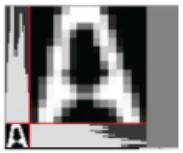
Una vez que obtengas las caracteristicas vamos a transformarlos al tipo CV_32FC1 y a partir de ello vamos a escribirlo en un archivo **OCR.xml**

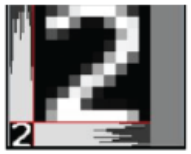
OCR.xml

Obtencion de Caracteristicas

Se Obtienes las caracteristicas de la imagen de cada caracter, se almacena en un array los valores normalizados de las columnas y de los horizontales de la imagen, como tambien los valores propios sin normalizar de la imagen. Se define tambien el tamaño de la imagen de cual desean extraer las caracteristicas, por motivos de costo computacional se opto por una imagen 5x5 pixeles.

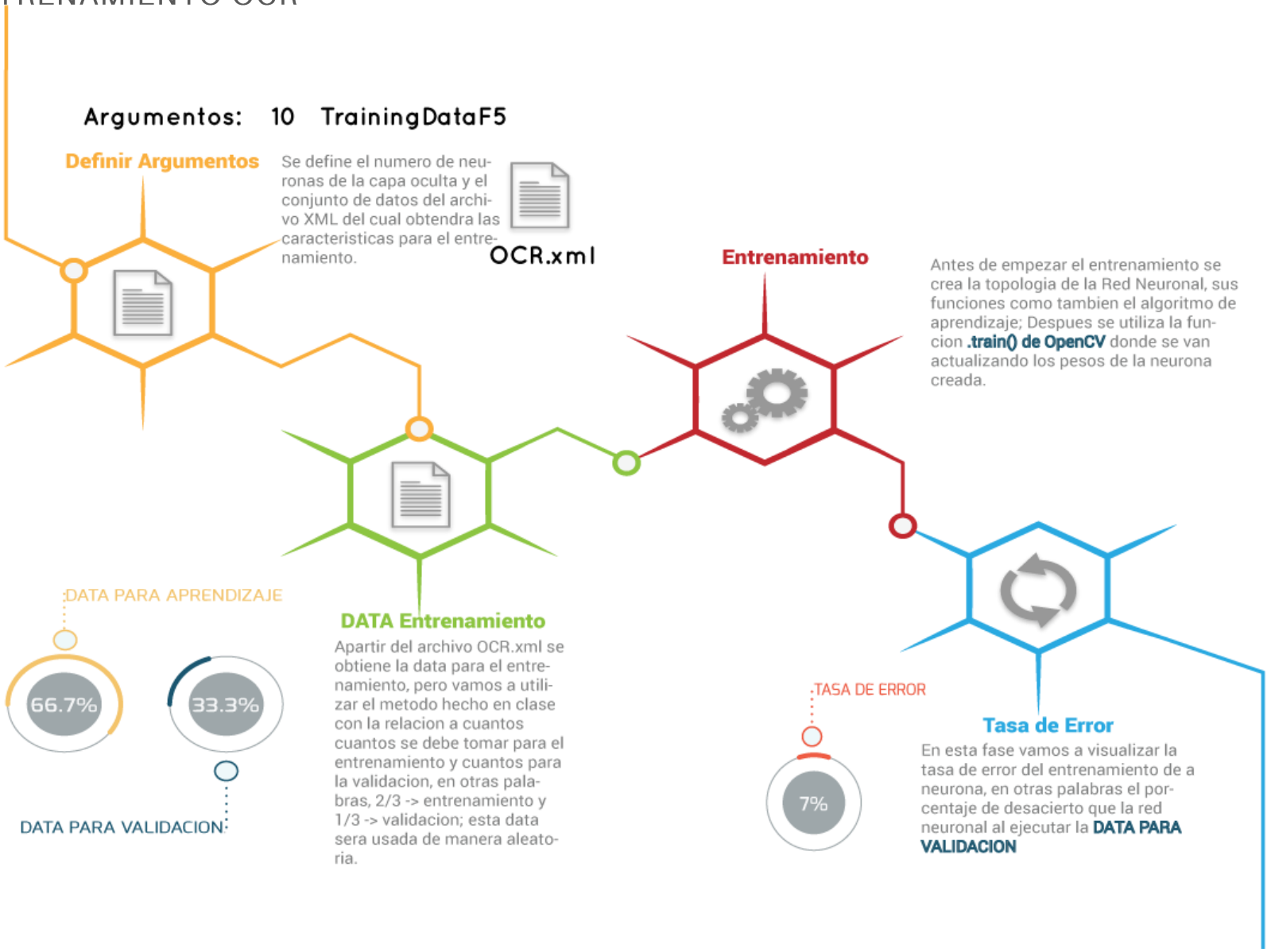






Entrenamiento del OCR

ENTRENAMIENTO OCR



APLICACIÓN FINAL E INTEGRACION

Segmentacion

Se lee la imagen, luego se utiliza las funciones de segmentacion de la 1°fase donde tendremos como resultado regiones posibles de placas.

Clasificacion SVM

Creamos un objeto de SVM con la ayuda de **OpenCV** donde utilizara como data ,para poder clasificar las nuevas regiones, el archivo **SVM.xml.** Luego clasificamos las regiones posibles en Placas y No placas, de las cuales solo se guardaran las regiones de placas.

Clasificacion OCR

Una vez obtenido las regiones que contienen la placa, se segmentara para encontrar las regiones de caracteres. Donde cada caracter sera clasificado mediante el uso de la red neuronal, dando como resultado el nombre del caracter en **CHAR 'D'.**

VISUALIZACION

Por ultimo se ordena las letras clasificadas de acuerdo a la posicion horizontal que se encuentran en la imagen y seran imprimidas en la imagen principal.

FIN DEL PROYECTO

CONCLUSIONES

La red neuronal RPROP es la más eficaz que el backpropagation, ya que es la que se entrena a menor iteraciones respecto a la demás.

La actualización de los pesos se realiza en función al signo de la gradiente descendiente.

El SVM agilizo de manera considerable la clasificación de las imágenes recortadas en placas y no placas.

OpenCV proporciona herramientas de **Machine Learning** muy fáciles de implementar en cualquier aplicación.

La aplicación de detección de placas sirve para distintos objetivos como el de seguridad, automatización de una playa de estacionamiento, búsqueda de autos robados y entre otros.