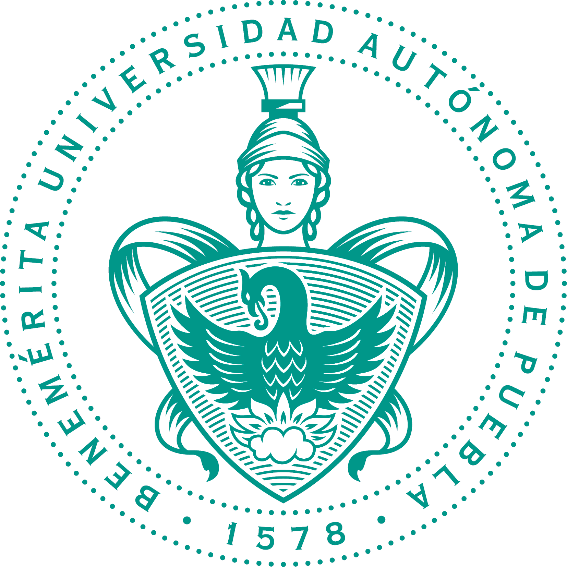
BENEMERITA UNIVERSIDAD AUTONOMA DE PUEBLA



Faculta de Computación

**Ingeniería en Ciencias de la Computación**

Tratamiento de Información

**Dr. Arturo Olvera López**

Redes Neuronales Multicapa

Reporte de Proyecto

**Ivan Bravo Mendoza 201210213**

**Otoño 2017**

Indice

1. Herramientas de desarrollo para la aplicación
   1. Implementación de la red
   2. Interfaz
   3. Desarrollo
2. Descripción de la aplicación
   1. Crear Red
      1. Crear
      2. Cargar
      3. Información
   2. Entrenar Red
      1. Entrenamiento
      2. Resultados
   3. Pruebas
      1. Validar la Red
      2. Resultados
      3. Probar la Red
      4. Resultados

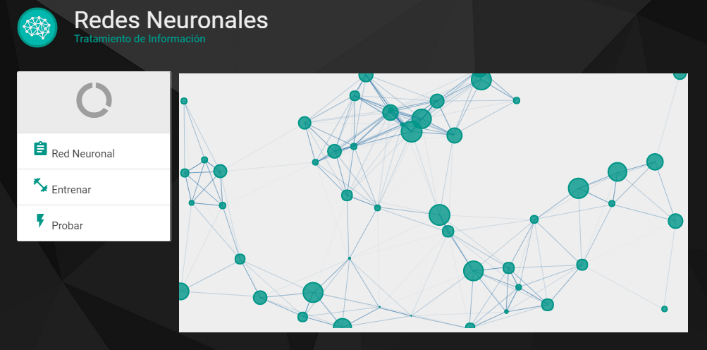
Herramientas de desarrollo para la aplicación

# Implementación de la red

La implementación de la red neuronal fue desarrollada utilizando la librería FANN (Fast Artificial Neural Network Library) [1], la cual proporciona las funciones necesarias para crear una red multicapa.

* La versión utilizada de esta Liberia es 2.2 para JavaScript.

# Interfaz

El desarrollo de la interfaz fue realizado utilizando HTML 5 y CSS 3, integrando frameworks para el diseño Materialize y graficas con Highcharts.

# Desarrollo

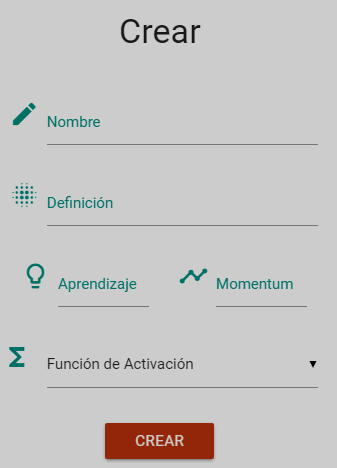
Se compone por 3 partes esenciales: creación, entrenamiento y prueba, esto enfocado directamente a la red neuronal. Cada una de estas partes engloban los conceptos generales de la implementación real de una red neuronal.

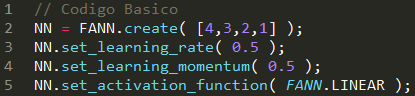
Descripcion de la aplicacion

# Crear Red

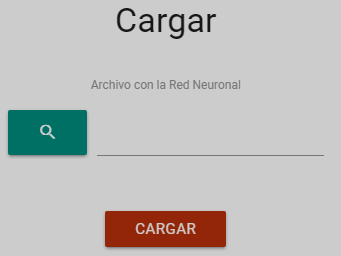
# Crear

Se proporciona una interfaz donde el usuario pueda ingresar los datos necesarios para crear una red nueva. El usuario debe ingresar:

* Nombre de la red. Esto con el fin de poder guardar la red e identificarla si se desea acceder a los datos de entrenamiento y compararlos con otras implementaciones.
* Definición de la red. Esta se especifica mediante capas y neuronas por capas. Por ejemplo, se crea una red con la siguiente definición [4,3,2,1], esto indica que la primera capa (capa de entrada) tendrá cuatro atributos como entrada, la segunda capa tendrá tres neuronas, la tercera capa tendrá dos neuronas y finalmente una neurona como capa de salida (cuarta capa).
* Aprendizaje o tasa de aprendizaje. Es un valor entre [0,1], indica que tan estricto se realizar el aprendizaje de la red.
* Momentum. Valor entre [0,1].
* Función de Activación. Enlista las diferentes opciones de función que se pueden aplicar a la red.



# Cargar

Permite al usuario cargar una red anteriormente creada y entrenada. La aplicación se encarga de procesar la información para reconstruir la red neuronal, de la misma manera los datos obtenidos durante el entrenamiento. Una vez más, esto permitiría al usuario comparar la información con diferentes redes.

* El tipo de archivo aceptado tiene la extensión “. ibm”

# Información

Muestra la información general de la red creada. Cuando es entrenada la red, en esta sección se muestra el nombre del archivo con los datos de entrenamiento utilizados, estos se pueden visualizar en la misma interfaz, además de los datos de forma normalizada. Se especifica la forma en que los datos fueron utilizados durante el entrenamiento de la misma red (Originales, Normalizados).

# Entrenar Red

# Entrenamiento

Para el entrenamiento se necesitan los campos:

* Error Mínimo. Es el que esperamos obtener al final del entrenamiento, es una referencia para determinar la efectividad de la red. Como consideración este valor no influye durante el entrenamiento. El valor debe estar entre [0,1].
* Épocas. En las que la red será entrenada.
* Archivo. Este debe contener los datos de los casos de entrenamiento, el formato es el siguiente:

Número de elementos en el conjunto de entrenamiento

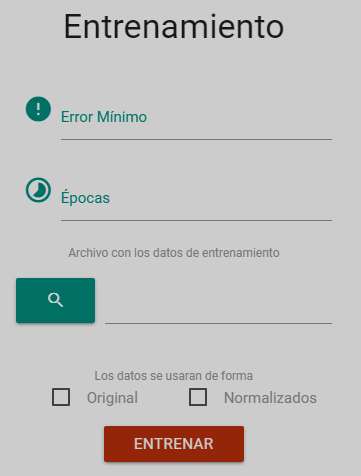
Número de atributos

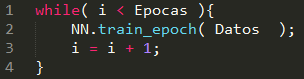
Número de clases

Atributo1, Atributo2, …, AtributoN, Clase

…

Atributo1, Atributo2, …, AtributoN, Clase

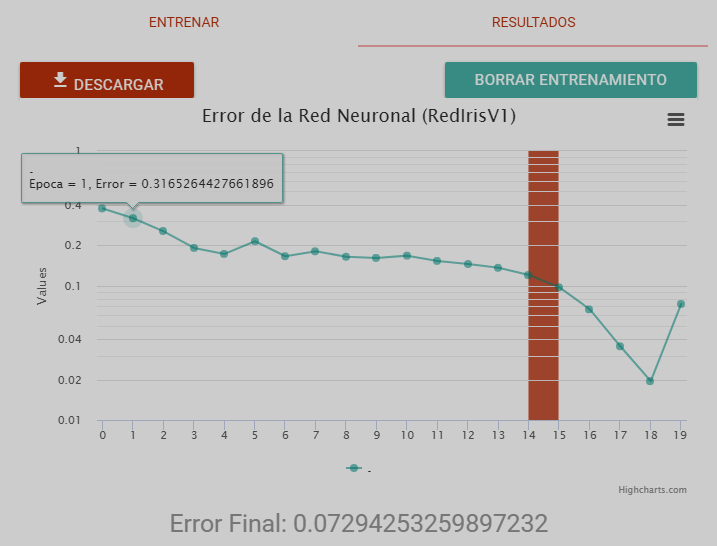
* Casillas. Debe seleccionar una casilla para especificar la forma en que se usaran los datos de entrenamiento: original o normalizados.



# Resultados

Muestra una gráfica con el resultado del entrenamiento donde los valores de x representan las épocas y los valores de y el error obtenido en dicha época. En la parte inferior de la gráfica se muestra el error obtenido en la última época. Se puede verificar el valor de cada una solo seleccionando el punto graficado. La línea color naranja representa la época en la que se obtuvo el error mínimo indicado por el usuario.

Las opciones adicionales disponibles son:

* Descargar. Proporciona un archivo con extensión “.ibm” el cual contiene la red con todos los parámetros especificados, así como el entrenamiento (resultados del entrenamiento).
* Borrar Entrenamiento. Si el resultado obtenido no fue el esperado, la red se podrá entrenar nuevamente.

# Pruebas

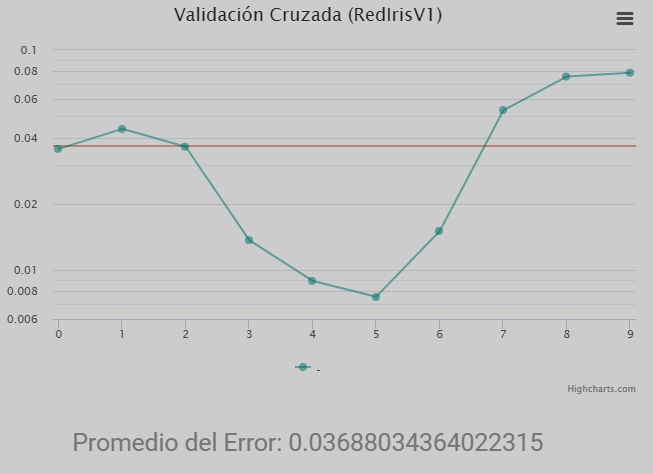
# Validar la Red

El usuario podrá especificar el valor que se usará para la validación cruzada. Este valor será utilizado para dividir el total de datos, en N bloques con (total de datos / N) elementos por cada uno.



# Resultados

Muestra una gráfica con el resultado de la validación cruzada donde los valores de x representan el bloque evaluado, el valor de error obtenido es representado por y.

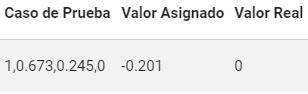


# Probar la Red

Se evalúan los casos de prueba o nuevos casos, para ello se da la opción a modo de casillas especificando: “Datos de Entrenamiento”, en donde evalúa los atributos del archivo de entrenamiento y “Casos Específicos”, el cual despliega una caja de texto en donde se puede ingresar los casos nuevos.



# Resultados

Muestra una tabla con: el caso de prueba utilizado, el valor asignado y la clase a la que pertenece dicho caso, esto es solo si se selecciona “Datos de Entrenamiento”, ya que el archivo contiene la clase a la que pertenece cada caso. Cuando se ingresan los casos de prueba nuevos, solo mostrara el caso de prueba y el valor asignado.



ExperimentacioN

# Objetivo

Visualizar los resultados que se pueden obtener con diferentes redes neuronales multicapa, variando los parámetros como: la definición de la red, el porcentaje de aprendizaje, momentum, función de activación, épocas, datos (originales, normalizados). Obtener un criterio para poder implementar un clasificador y justificar con base a pruebas la efectividad que se puede obtener.

# Análisis de los datos

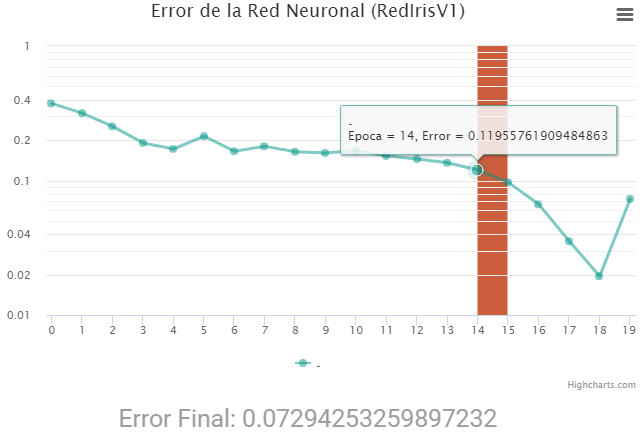
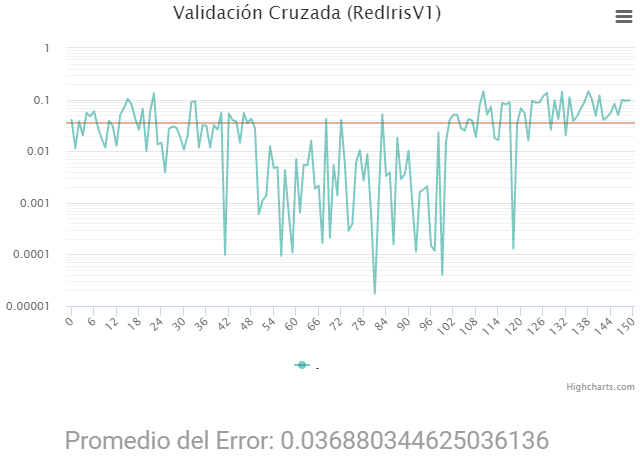
Los datos que se utilizaran para la experimentación es el conocido conjunto de datos “Iris”. Este conjunto ha sido utilizado en múltiples experimentaciones, y se puede encontrar información al respecto en internet. Edgar Anderson coleccionó la data usada para cuantificar la variación morfológica del Iris con las flores de tres especies relacionadas.

Las características de Iris (Fisher’s Iris) son:

* El conjunto contiene 150 casos.
* 3 clases (Iris setosa, iris virginica, iris versicolor).
* 50 elementos para Iris setosa, 49 para Iris virginica y 51 para Iris versicolor.
* 4 Atributos (Largo de sépalo, Ancho de sépalo, Largo de pétalo, Ancho de pétalo).
* Basado en la combinación de estos cuatro atributos, se puede obtener una región de separación lineal, la cual permite distinguir entre una especie y otra.

# Primer Implementación

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| RedIrisV1 | | | | | | | | | | | |
| Definición | | Aprendizaje | | Momentum | | Función | Error | | Épocas | Datos | |
| 4,3,2,1 | | 0.5 | | 0.5 | | LINEAR | 0.1 | | 20 | Normalizados | |
| Entrenamiento | | | | | | Validación | | | | | |
| Error | **Mejor** | | **Peor** | | **Mínimo** | **15** | | **50** | | | **150** |
| 0.07294 | 0.0194 | | 0.3741 | | 14-15 | 0.0368803 | | 0.0368803 | | | 0.0368803 |

# Segunda Implementación

|  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| RedIrisV1 | | | | | | | | | | | |
| Definición | | Aprendizaje | | Momentum | | Función | Error | | Épocas | Datos | |
| 4,3,2,1 | | 0.5 | | 0 | | LINEAR | 0.1 | | 100 | Normalizados | |
| Entrenamiento | | | | | | Validación | | | | | |
| Error | **Mejor** | | **Peor** | | **Mínimo** | **15** | | **50** | | | **150** |
| 0.01851 | 0.01851 | | 0.4275 | | 15-16 | 0.0185063 | | 0.0185063 | | | 0.0185063 |

