# Proyecto machine learning segundo corte

Para el proyecto del segundo corte, se planteó el entrenamiento de un sistema el cual contiene 10 clases de imágenes distintas (aviones, carros, barcos, pájaros, etc.), el objetivo de la práctica es entrenar un sistema de machine learning el cual nos permita identificar las imágenes de “botes” de nuestro dataset, utilizando el método de entrenamiento de redes neuronales.

# Desarrollo

Se parte que, a diferencia de otros proyectos, se nos brinda 2 data sets uno dedicado a entrenar al sistema, y otro dedicado a probarlo, para esto se utilizan los siguientes códigos:

# Código dataset de entrenamiento

clear all

load('train.mat')

w=y;%%%%% prueba

cla=9;%%%% clase seleccionada

f=5000;%# filas

c=96\*96\*3;%# columnas en GRB

c2=65\*65;%#columnas en Gray y con bajo resolución

%%%%%%%%%convertidor de X %%%%%%

for h=1:1:f

img=X(h,1:c);% selecciona la primera imagen fila 1

imgrgb=reshape(img,96,96,3);% lo convierte en matrix de 96\*96 cada 3 columnas

imgf=rgb2gray(imgrgb);%convierte la imagen de color a gris

B = imresize(imgf, [65 65]);% baja la resolución

imggray=reshape(B,1,c2);%convierte la matriz en un vector

p(h,1:c2)=imggray;% acopla en la matriz

end

%%%%%%%% convertir los demás valores en 0 y 1 donde uno solo es para la

%%%%%%%% clase 1 aviones

for i=1:1:f

if y(i,1) == cla;

w(i,1)=1;

else

w(i,1)=0;

end

end

%%%%%concatenar las targets and sources

p1=[w p];

%%%%%%%% guardar nuestro dataset

csvwrite('train.csv',p1);

en resumen, estos comandos nos permiten convertir los valores de salida del dataset de 10 clases a dos clases las que son botes y las que no son botes, a su vez de que se convierten las imágenes a color, en imágenes a escalas de grises con resolución de 65 x 65 de allí el comando “rgb2gray” y el comando “reshape”.

# Código dataset de prueba

clear all

load('test.mat')

w=y;%%%%% prueba

cla=9;%%%% clase seleccionada

f=8000;%# filas

c=96\*96\*3;%# columnas en GRB

c2=65\*65;%#columnas en Gray y con bajo resolución

%%%%%%%%%convertidor de X %%%%%%

for h=1:1:f

img=X(h,1:c);% selecciona la primera imagen fila 1

imgrgb=reshape(img,96,96,3);% lo convierte en matrix de 96\*96 cada 3 columnas

imgf=rgb2gray(imgrgb);%convierte la imagen de color a gris

B = imresize(imgf, [65 65]);% baja la resolución

imggray=reshape(B,1,c2);%convierte la matriz en un vector

p(h,1:c2)=imggray;% acopla en la matriz

end

%%%%%%%% convertir los demas valores en 0 y 1 donde uno solo es para la

%%%%%%%% clase 1 aviones

for i=1:1:f

if y(i,1) == cla;

w(i,1)=1;

else

w(i,1)=0;

end

end

%%%%%concatenar las targets and sources

p1=[w p];

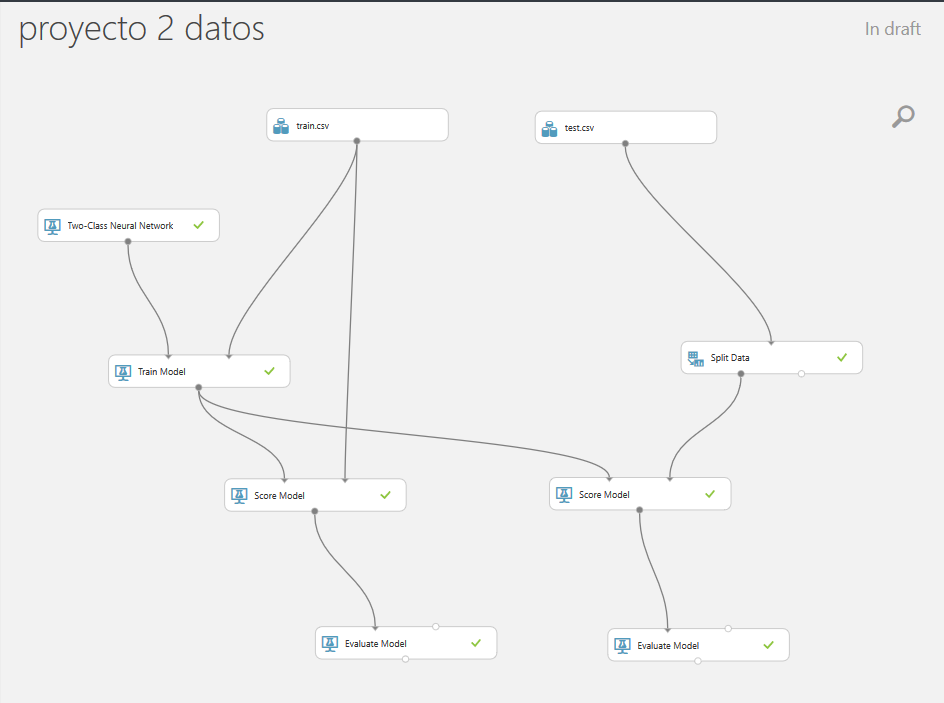
%%%%%%%% guardar nuestro dataset

csvwrite('test.csv',p1);

Este proceso se hace tanto para el dataset de entrenamiento como para el dataset de prueba, con la única diferencia es que la cantidad de datos varían, ya que ay más datos de prueba que de entrenamiento.

# Modelo de simulación

Para nuestra simulación y obtención de los valores a desarrollar, se implementó el siguiente modelo:



En este modelo se puede observar la integración de los dos datasets (entrenamiento y prueba) los cuales generan dos salidas:

Primera salida: corresponde al entrenamiento y evaluación con el mismo dataset, esto nos permite saber que tan bien se entrena el sistema, comparando y poniendo a prueba el aprendizaje adquirido con el mismo dataset

Segunda salida: el sistema se entrena con la data set de entrenamiento, pero se evalúa con el 50% del dataset de prueba, parte la cual se denomina como validación cruzada

Este tratamiento nos permite mejorar la precisión del sistema, además de realizarle un diagnostico al algoritmo.

Para esto nuestro valor de referencia será el AUC ( área under curve )la cual nos permite evaluar el Rendimiento en todos los umbrales de calificación posibles , haciéndola un valor clave de referencia.

# Proceso de extracción de valores

Antes de evaluar nuestro sistema, requerimos identificar que valores de nuestra red neuronal nos genera una predicción más cercana a los datos verdaderos, para ello se vario dato por dato, siguiendo el modelo anterior planteado y realizando diversas graficas en Excel comparando los diversos valores generados.

Los valores señalados con verde en las tablas , corresponden a los mejores valores generados.

# # de neuronas

|  |  |  |
| --- | --- | --- |
| neuronas | train | cv |
| AUC | AUC2 |
| 10 | 81,11% | 78,80% |
| 30 | 75,30% | 75,00% |
| 50 | 74,00% | 71,30% |
| 70 | 75,80% | 75,10% |
| 85 | 73,20% | 70,10% |
| 90 | 79,30% | 77,10% |
| 100 | 71,80% | 69,50% |
| 105 | 76,70% | 74,80% |
| 110 | 75,40% | 71,80% |
| **115** | **77,90%** | **76,20%** |
| 117 | 69,70% | 67,40% |
| 120 | 50,30% | 49,80% |
| 150 | 50,30% | 50,20% |

# The initial learning weights diameter

|  |  |  |
| --- | --- | --- |
| valor pesos | train | cv |
| AUC | AUC2 |
| 0,1 | 77,90% | 76,20% |
| 1 | 78,30% | 74,60% |
| 10 | 80,30% | 76,60% |
| 100 | 81,20% | 76,10% |
| 115 | 81,50% | 77,70% |
| 120 | 81,50% | 77,20% |
| 130 | 81,90% | 77,00% |
| 150 | 80,40% | 77,77% |
| 200 | 75,20% | 72,30% |
| 500 | 69,30% | 65,40% |
| 1000 | 60,90% | 61,40% |

# Number of learning iterations

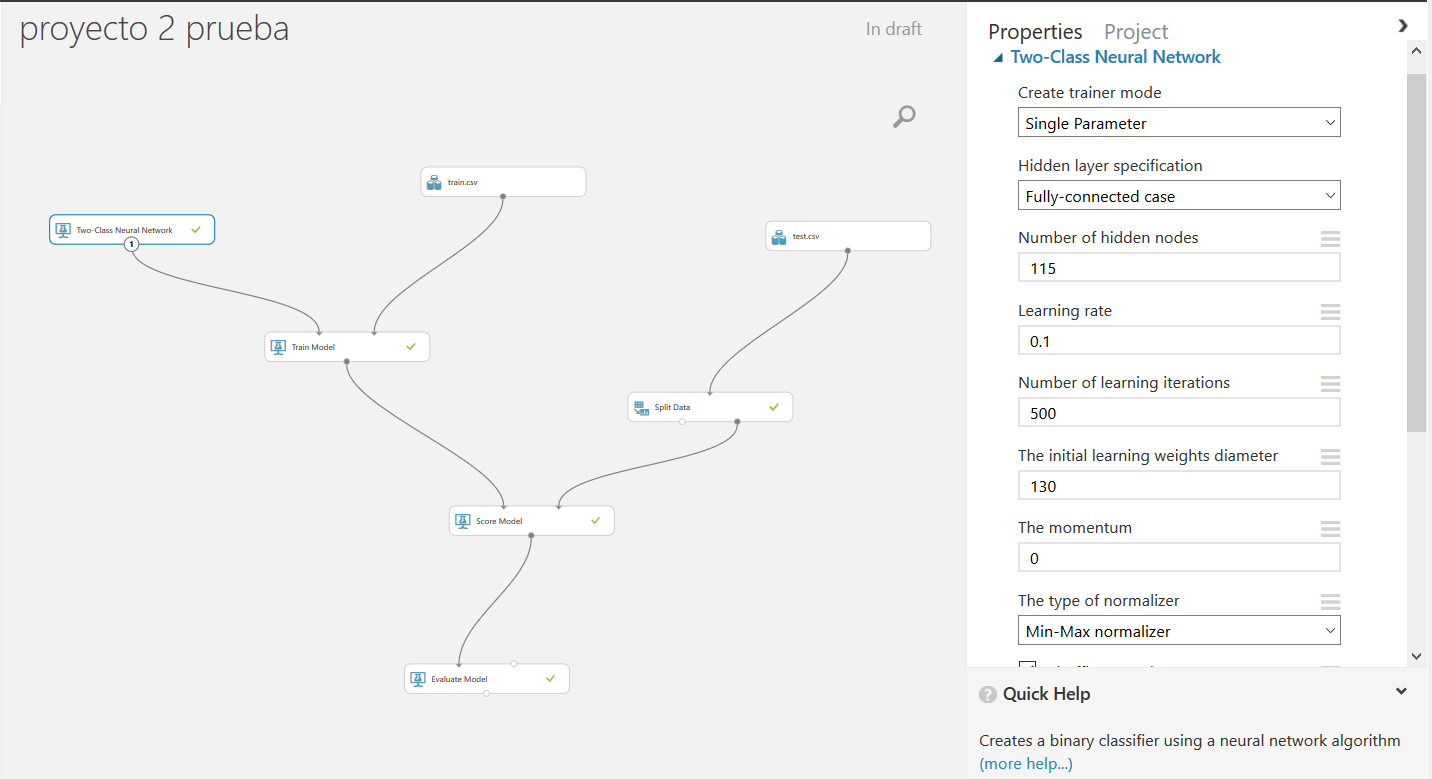
|  |  |  |
| --- | --- | --- |
| # iteraciones | train | cv |
| AUC | AUC2 |
| 10 | 66,40% | 67,70% |
| 50 | 78,80% | 75,10% |
| 70 | 77,77% | 76,00% |
| 100 | 81,90% | 77,00% |
| 200 | 82,40% | 77,70% |
| 500 | 85,00% | 80,50% |
| 700 | 84,80% | 77,70% |

# Learning rate

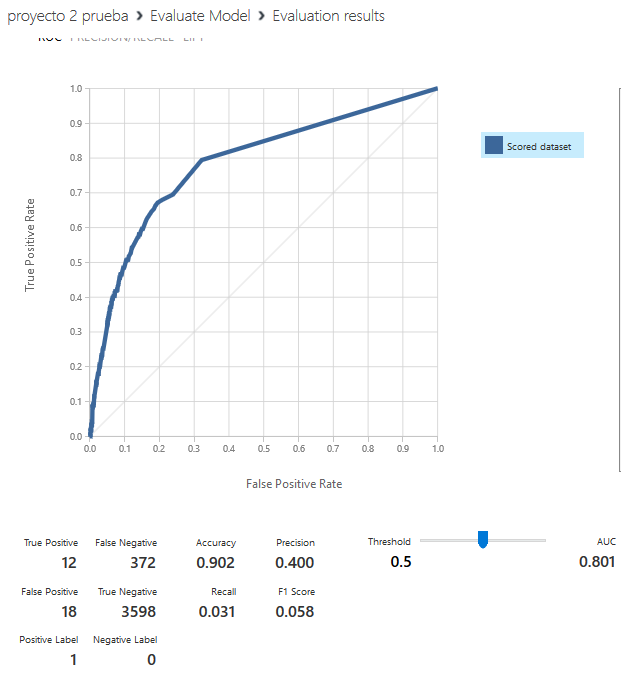
|  |  |  |
| --- | --- | --- |
| Learning rate | train | cv |
| AUC | AUC2 |
| 0,001 | 66,20% | 63,60% |
| 0,01 | 79,50% | 73,40% |
| 0,05 | 84,80% | 76,80% |
| 0,07 | 85,10% | 77,10% |
| 0,085 | 86,60% | 77,70% |
| 0,09 | 86,10% | 79,30% |
| 0,1 | 85,00% | 80,50% |
| 0,3 | 51,60% | 50,10% |
| 0,5 | 50,60% | 50,00% |
| 1 | 50,20% | 50,00% |

# Realización de la prueba

Para la realización de la prueba se implementó el siguiente esquema, con la siguiente configuración de los parámetros de la red neuronal.



Se puede observar en el modelo que lo que varía con respecto a modelo de entrenamiento validación cruzada, es que se toma el otro 50% de la data set de prueba, dando como resultado los siguientes valores:



# Análisis de resultados:

Como se pudo observar en la gráfica anterior se obtuvieron muy buenos valores de AUC, precisión y f1 score, esto debido al proceso minucioso de selección de valores , pero aun asa aunque estos valores representen un valor alto , al visualizar las tablas de verdad , nos damos cuenta que no se están identificando los verdaderos positivos como corresponden , esto debido a que a diferencia de los datos de prueba y de validación cruzada , estos datos son muy pocos representando solo el 10% de los datos de entrenamiento , de allí la falta de valores positivos, todo esto a diferencia de los verdaderos negativos , cuyo valor se acerca mucho a los valores originales planteados para el dataset.