

Redes Neuronales Competitivas. ("CLNN")

Autor. Omar Baruch Morón López, *Universidad de Guadalajara*

Resumen En esta práctica se explicará lo que es una red neuronal competitiva y se programara en Matlab para un problema de agrupación de datos de 3 dimensiones y ocho neuronas(grupos).

Palabras Clave— Red neuronal, inteligencia artificial, competitividad, agrupación, Multisalida.

I. INTRODUCCION

“Las redes de aprendizaje competitivo se diferencian de las otras redes neuronales en que en las anteriores redes las neuronas colaboran en la representación de los patrones, sin embargo, en este tipo de redes cada neurona compite con las otras neuronas para representar los patrones.

El aprendizaje de este tipo de redes es como su nombre indica, competitivo.

Las neuronas compiten en cual representa mejor al patrón y la ganadora se lleva todo el aprendizaje de ese patrón. El objetivo de este tipo de redes es que se formen grupos de patrones, categorías, que son representados por cada neurona.

Cuando ejecutamos un patrón en una red competitiva solamente se activa una neurona que es la que representa mejor el patrón.

Este tipo de redes fue desarrollado por Rumelhart y Zipser en 1985 aunque a partir de él se han diversificado sus aplicaciones y modificaciones dando lugar a redes tan interesantes como las redes de kohonen y otras” [1]

Arquitectura

Las redes neuronales competitivas son monocapa, y están basadas en tener una neurona por cada grupo que deseas tener de tus datos, con la siguiente estructura:

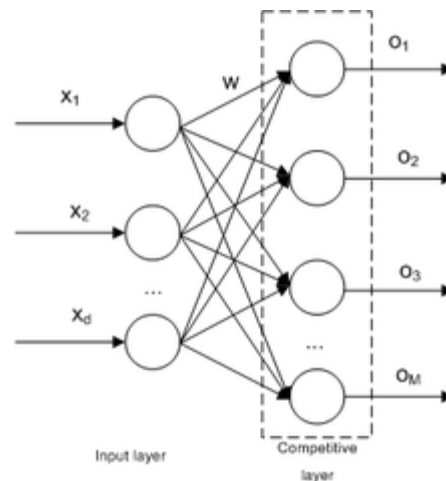


Figure 1: Arquitectura de las redes neuronales competitivas

Regla de entrenamiento

La regla de entrenamiento está dada por:

$$v = xTw - .5(wTw) + \sum \text{sombreroMexicano}(\text{distanciaNeuronal entre una neurona a todas})$$

Regla de actualización de pesos

Los pesos se actualizan solo para la neurona con la salida mas alta de entre todas las salidas de las neuronas. Y la actualización está dada por:

$$w(\text{nuevo}) = w(\text{actual}) + \eta(X(\text{actual}) - w(\text{actual}))$$

II. OBJETIVO DE LA PRACTICA

El objetivo de esta práctica es dividir en grupos un conjunto de datos. Para poder hacer una representación visual del problema se hará de 3 dimensiones, 3 variables, en otras palabras, más relacionadas con la estructura neuronal; 3 entradas.

Generare un conjunto de datos aleatorios en un plano tridimensional, por lo que por motivos didácticos planeo generar conjuntos de datos separados para cada uno de los 8 cuadrantes, y las neuronas las iniciare en el centro cercano al origen, pero de igual forma una en cada cuadrante.

Para la competencia entre neuronas se usará la función de sombrero mexicano con la que se castigara a las neuronas que están más lejos de la neurona actual y ayudara a las más cercanas.

Los valores de a y b que son coeficientes para la modificación de la función será de .01 y .06 respectivamente y eso debido a que nuestros datos y problemas se mantienen en valores menores a 1.5 y es algo pequeño.

III. ((EXTRA) COEFICIENTE DE APRENDIZAJE IDEAL

Un plus a la práctica es que deducir una forma de obtener un coeficiente de aprendizaje ideal sin que se salga de los parámetros del problema y no diverjan las neuronas. Por medio de la siguiente fórmula:

$$\text{Coeficiente de aprendizaje} = \frac{\sum_k \frac{\sum_i x_i}{n} k}{d}$$

IV. RESULTADOS

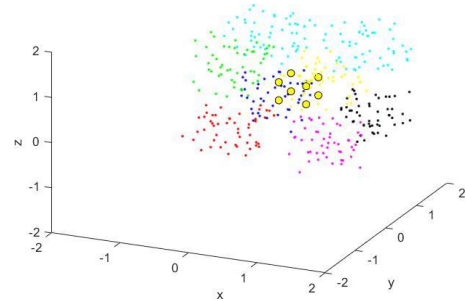


Ilustración 1: Datos dispersos y neuronas en posición inicial

A partir de este punto se calcula para cada dato la neurona ganadora y se le actualiza el peso de la neurona ganadora por lo que se ira acercando al grupo correspondiente, y esto se realizara el número de generaciones diseñadas.

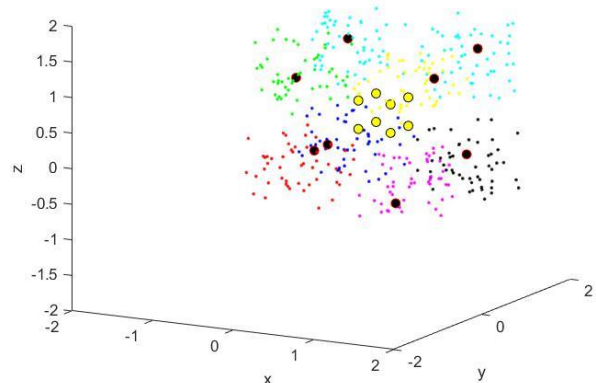


Ilustración 2: Neuronas reorganizadas.

Podemos ver en la ilustración 2 el resultado. En amarillo las neuronas iniciales, los datos con colores variados, y la posición final de las neuronas en color negro.

V. CONCLUSIÓN

La red neuronas competitiva es ideal para identificar patrones, aglomeración de datos, agrupar resultados de combinaciones.

Es muy útil he incluso podría llegar a realizar tareas de organización que ni un humano podría hacer, o agrupar datos de 4,5 6 o mas dimensiones que son imposibles para el humano de concebir.

Incluso pensé en un ejemplo, de agrupación de resultados de un experimento de 4 variables que podría realizar algún investigador en química, donde realiza muestras de combinaciones posibles de alguna solución del cual dependen 5 compuestos que su proporción varía, es imposible para el humano

poder organizar los resultados debido a que es de 5 dimensiones, sin embargo, la red neuronal con algún tiempo de cálculo podría agruparlas para así crear categorías nuevas.

VI. CÓDIGO

```
clear all; clc; clear all; cla;
%% Variables modificables
Numero_deDatos=50;
Neuronas=8;
a=.01;
b=.06;
Generaciones=10;
sep=.7;%Para ploteo
%%
hold on
axis([-2,2,-2,2,-2,2]);
xlabel('x');
ylabel('y');
zlabel('z');
g1=[rand(1,Numero_deDatos)+sep; rand(1,Numero_deDatos)+sep; rand(1,Numero_deDatos)+sep];
scatter3(g1(1,:),g1(2,:),g1(3,:),'.','MarkerFaceColor','r','MarkerEdgeColor','r');
g2=[rand(1,Numero_deDatos)+sep; rand(1,Numero_deDatos)+sep; rand(1,Numero_deDatos)+sep];
scatter3(g2(1,:),g2(2,:),g2(3,:),'.','MarkerFaceColor','g','MarkerEdgeColor','g');
g3=[rand(1,Numero_deDatos)+sep; rand(1,Numero_deDatos)+sep; rand(1,Numero_deDatos)+sep];
scatter3(g3(1,:),g3(2,:),g3(3,:),'.','MarkerFaceColor','b','MarkerEdgeColor','b');
g4=[rand(1,Numero_deDatos)+sep; rand(1,Numero_deDatos)+sep; rand(1,Numero_deDatos)+sep];
scatter3(g4(1,:),g4(2,:),g4(3,:),'.','MarkerFaceColor','c','MarkerEdgeColor','c');
g5=[rand(1,Numero_deDatos)+sep; rand(1,Numero_deDatos)+sep; rand(1,Numero_deDatos)+sep];
scatter3(g5(1,:),g5(2,:),g5(3,:),'.','MarkerFaceColor','m','MarkerEdgeColor','m');
g6=[rand(1,Numero_deDatos)+sep; rand(1,Numero_deDatos)+sep; rand(1,Numero_deDatos)+sep];
scatter3(g6(1,:),g6(2,:),g6(3,:),'.','MarkerFaceColor','y','MarkerEdgeColor','y');
g7=[rand(1,Numero_deDatos)+sep; rand(1,Numero_deDatos)+sep; rand(1,Numero_deDatos)+sep];
scatter3(g7(1,:),g7(2,:),g7(3,:),'.','MarkerFaceColor','k','MarkerEdgeColor','k');
g8=[rand(1,Numero_deDatos)+sep; rand(1,Numero_deDatos)+sep; rand(1,Numero_deDatos)+sep];
scatter3(g8(1,:),g8(2,:),g8(3,:),'.','MarkerFaceColor','y','MarkerEdgeColor','y');
XX=[g1,g2,g3,g4,g5,g6,g7,g8];
X=mix(XX);
%Inteligencia Ideal para un problema 3D
inte(1)=mean(X(1,:)); %x
inte(2)=mean(X(2,:)); %y
inte(3)=mean(X(3,:)); %z
Inteligencia=mean(inte)%Unificacion

%% Datos Leatorios
T=X;
%% Inicializar w
w=.8;
w1=[-.5+u;-.5+u;-.5+u];w1=2;
w2=[-.5+u;-.5+u;-.5+u];w2=3;
w3=[-.5+u;-.5+u;-.5+u];w3=3.5;
w4=[-.5+u;-.5+u;-.5+u];w4=4.5;
w5=[1.5+u;-.5+u;-.5+u];w5=5.6;
w6=[1.5+u;-.5+u;-.5+u];w6=7.8;
w7=[1.5+u;-.5+u;-.5+u];w7=8;
w8=[1.5+u;-.5+u;-.5+u];w8=9;
Neu=[w1 w2 w3 w4 w5 w6 w7 w8];
NeuInicial=Neu;
for i=1:length(Neu) %Pintar neuronas en posicion inicial
    scatter3(Neu(i,1),Neu(i,2),Neu(i,3),'MarkerFaceColor','y','MarkerEdgeColor','k');
end
for r=1:Generaciones
    for i=1:Numero_deDatos
        v(:,1)=Neu(:,1)*T(i,1)-(5*Neu(:,1)*Neu(:,1))+sombbrero(abs(wd1-wd1),a,b)+sombbrero(abs(wd1-wd2),a,b)+sombbrero(abs(wd1-wd3),a,b)+sombbrero(abs(wd1-wd4),a,b)+sombbrero(abs(wd1-wd5),a,b)+sombbrero(abs(wd1-wd6),a,b)+sombbrero(abs(wd1-wd7),a,b)+sombbrero(abs(wd1-wd8),a,b));
        v(:,2)=Neu(:,2)*T(i,1)-(5*Neu(:,2)*Neu(:,2))+sombbrero(abs(wd2-wd1),a,b)+sombbrero(abs(wd2-wd2),a,b)+sombbrero(abs(wd2-wd3),a,b)+sombbrero(abs(wd2-wd4),a,b)+sombbrero(abs(wd2-wd5),a,b)+sombbrero(abs(wd2-wd6),a,b)+sombbrero(abs(wd2-wd7),a,b)+sombbrero(abs(wd2-wd8),a,b));
        v(:,3)=Neu(:,3)*T(i,1)-(5*Neu(:,3)*Neu(:,3))+sombbrero(abs(wd3-wd1),a,b)+sombbrero(abs(wd3-wd2),a,b)+sombbrero(abs(wd3-wd3),a,b)+sombbrero(abs(wd3-wd4),a,b)+sombbrero(abs(wd3-wd5),a,b)+sombbrero(abs(wd3-wd6),a,b)+sombbrero(abs(wd3-wd7),a,b)+sombbrero(abs(wd3-wd8),a,b));
        v(:,4)=Neu(:,4)*T(i,1)-(5*Neu(:,4)*Neu(:,4))+sombbrero(abs(wd4-wd1),a,b)+sombbrero(abs(wd4-wd2),a,b)+sombbrero(abs(wd4-wd3),a,b)+sombbrero(abs(wd4-wd4),a,b)+sombbrero(abs(wd4-wd5),a,b)+sombbrero(abs(wd4-wd6),a,b)+sombbrero(abs(wd4-wd7),a,b)+sombbrero(abs(wd4-wd8),a,b));
        v(:,5)=Neu(:,5)*T(i,1)-(5*Neu(:,5)*Neu(:,5))+sombbrero(abs(wd5-wd1),a,b)+sombbrero(abs(wd5-wd2),a,b)+sombbrero(abs(wd5-wd3),a,b)+sombbrero(abs(wd5-wd4),a,b)+sombbrero(abs(wd5-wd5),a,b)+sombbrero(abs(wd5-wd6),a,b)+sombbrero(abs(wd5-wd7),a,b)+sombbrero(abs(wd5-wd8),a,b));
        v(:,6)=Neu(:,6)*T(i,1)-(5*Neu(:,6)*Neu(:,6))+sombbrero(abs(wd6-wd1),a,b)+sombbrero(abs(wd6-wd2),a,b)+sombbrero(abs(wd6-wd3),a,b)+sombbrero(abs(wd6-wd4),a,b)+sombbrero(abs(wd6-wd5),a,b)+sombbrero(abs(wd6-wd6),a,b)+sombbrero(abs(wd6-wd7),a,b)+sombbrero(abs(wd6-wd8),a,b));
        v(:,7)=Neu(:,7)*T(i,1)-(5*Neu(:,7)*Neu(:,7))+sombbrero(abs(wd7-wd1),a,b)+sombbrero(abs(wd7-wd2),a,b)+sombbrero(abs(wd7-wd3),a,b)+sombbrero(abs(wd7-wd4),a,b)+sombbrero(abs(wd7-wd5),a,b)+sombbrero(abs(wd7-wd6),a,b)+sombbrero(abs(wd7-wd7),a,b)+sombbrero(abs(wd7-wd8),a,b));
        v(:,8)=Neu(:,8)*T(i,1)-(5*Neu(:,8)*Neu(:,8))+sombbrero(abs(wd8-wd1),a,b)+sombbrero(abs(wd8-wd2),a,b)+sombbrero(abs(wd8-wd3),a,b)+sombbrero(abs(wd8-wd4),a,b)+sombbrero(abs(wd8-wd5),a,b)+sombbrero(abs(wd8-wd6),a,b)+sombbrero(abs(wd8-wd7),a,b)+sombbrero(abs(wd8-wd8),a,b));
        [~,ind]=max(v);
        Neu(:,ind)=(Neu(:,ind)+(Inteligencia*(T(i,1)-Neu(:,ind))));
    end
end
for i=1:length(Neu) %Neuronas finales
    scatter3(Neu(i,1),Neu(i,2),Neu(i,3),'MarkerFaceColor','k','MarkerEdgeColor','r');
end
function [X] = mix(XX)
    indices = randperm(length(XX));
    X = zeros(3,length(XX));
    for ind1 = 1:length(XX)
        X(:,ind1) = XX(:,indices(ind1));
    end
end
function s = sombrero(v,a,b)
    s=(b-a*v^2)*exp((-v^2));
end
```

REFERENCIAS

- [1] Alfonso Ballesteros Computing Ingenieer by the University of Malaga, Spain as a part of his final degree project with the supervision of D. Enrique Dominguez. Se puede encontrar en <http://www.redes-neuronales.com/es/tutorial-redes-neuronales/red-neuronal-competitiva-simple.htm>