



UNIVERSIDAD NACIONAL AUTONOMA DE MÉXICO

**FACULTAD DE ESTUDIOS SUPERIORES
ARAGÓN**

INGENIERIA EN COMPUTACION

DISEÑO Y ANÁLISIS DE ALGORITMOS

GRUPO: 1509

PROFESOR: MIGUEL ÁNGEL SÁNCHEZ HERNANDEZ

TERCER PARCIAL

**BALCAZAR GONZÁLEZ LUIS ENRIQUE
JIMÉNEZ SANDOVAL JORGE ÁNGEL
PÉREZ ALVAREZ LUIS GONZALO
SOSA DELGADILLO LUIS RUBEN**

28 - NOVIEMBRE - 2023

Contenido

Redes Neuronales	3
Objetivo	5
Programa	6
Recolección de datos.....	6
Entrenamiento de los perceptrones.....	6
Clasificador por Arduino	8
Conclusión	12
Referencias.....	14

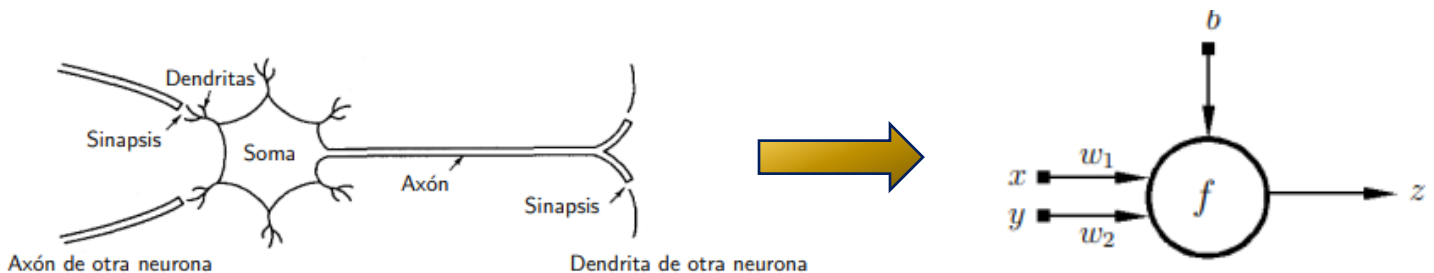
Redes Neuronales

En computación cuando hablamos de una red neuronal hacemos referencia al término de la inteligencia artificial, ya que con esto se enseña a las computadoras a procesar datos de manera inspirada en la que lo hace el cerebro humano. Se trata de un tipo de proceso de *Machine Learning* (Aprendizaje de Máquina), que utiliza los nodos o las neuronas interconectados en una estructura de capas que intentan simular el funcionamiento del cerebro humano.

Con esa intención se crea un sistema adaptable que las computadoras utilizan para aprender de sus errores y mejorar continuamente, de esta forma las redes neuronales artificiales intentan resolver problemas que están presentes en varios casos de uso y en muchos sectores, como los siguientes:

- **Visión artificial:** con la visión artificial se tiene la capacidad para extraer información y conocimientos de imágenes y videos que con las redes neuronales las computadoras pueden distinguir y reconocer imágenes de forma similar a los humanos.
- **Reconocimiento de voz:** Las redes neuronales pueden analizar el habla humana a pesar de los diferentes patrones de habla como el tono, el idioma y el acento y con los asistentes virtuales y el software de transcripción automática utilizan el reconocimiento de voz para realizar algunas tareas.
- **Procesamiento de lenguaje natural:** El procesamiento de lenguaje natural es la capacidad de procesar texto natural creado por humanos con él, las redes neuronales ayudan a las computadoras a obtener información y significado a partir de los datos y los documentos de texto.
- **Motores de recomendaciones:** gracias a las redes neuronales se puede hacer seguimiento de la actividad de usuario para elaborar recomendaciones personalizadas, además de analizar todo el comportamiento de los usuarios y descubrir productos o servicios nuevos que interesen a un usuario específico.

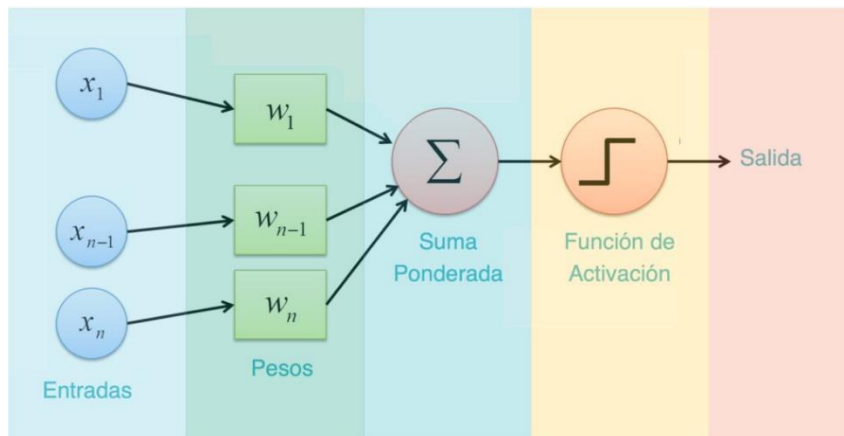
Como se mencionó anteriormente las redes neuronales suelen simular el funcionamiento de aprendizaje del ser humano, es por eso que se utiliza a la neurona como modelo de funcionamiento, solo que en el área de la computación se conoce como perceptrón y mientras más capas y perceptrones se usan son más parecidas a lo que son los grafos, pero cada perceptrón puede ser descrito como una función que tiene datos de entrada y debe generar datos de salida a través del proceso con el que este configurado.



Las entradas x e y son la entrada que la neurona artificial (perceptrón) recibe del entorno, y la salida z es la respuesta a la función planteada, la neurona se adapta y aprende modificando el valor de sus pesos sinápticos w_1 y w_2 y su Bias o sesgo b . Estos son los parámetros libres del modelo, pues los mismos pueden ser modificados y adaptados para realizar una tarea determinada, en este modelo la salida de la neurona z se da por $z = f(w_1 x + w_2 y + b)$ y la función de activación f es seleccionada de acuerdo con la tarea que quiera ser realizada por la neurona.

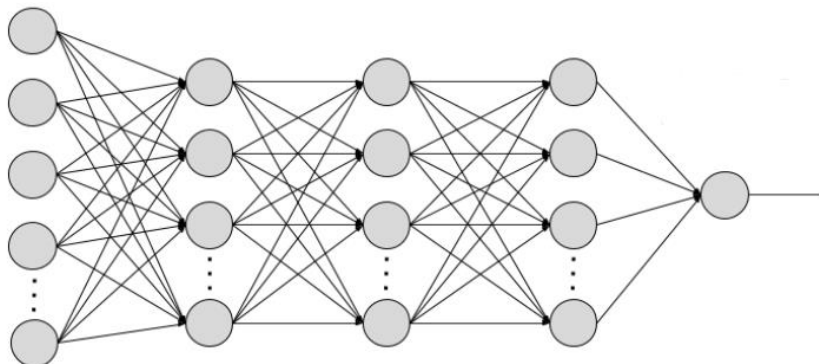
Con los datos obtenidos por el perceptrón clasificaremos clases dentro de un plano, aplicando el modelo neuronal, consideraremos como función de activación definida por:

$$f(s) = \begin{cases} 1, & \text{si } w_1 x + w_2 y + b \geq 0 \\ 0, & \text{si } w_1 x + w_2 y + b < 0 \end{cases}$$



Ya que se describió una neurona y como funciona, podemos definir lo que es una red neuronal, ya que consiste en una serie de capas de neuronas, específicamente, todas las neuronas de una capa se conectan a las neuronas de la siguiente capa, es decir que las entradas a la red se conectan con los nodos de la primera capa y su salidas con los siguientes nodos y así de manera sucesiva hasta las neuronas que constituyen la salida final de la red, las capas ocultas formadas por las neuronas que se encuentran entre los nodos de entrada y la capa de salida se denominan capas ocultas y una red neuronal puede tener varias capas ocultas o no tener ninguna de ellas.

Las flechas indican el flujo de la red y tienen asociado un peso sináptico correspondiente, si la salida de una neurona va dirigida hacia dos o más neuronas de la siguiente capa, cada una de estas últimas recibe la salida de la neurona anterior. De manera general podemos decir que el algoritmo de la red neuronal es determinar el peso que deben tener los pesos de las redes neuronales para obtener la salida deseada o esperada.



Objetivo

Aprender a usar Arduino para poder escribir información en tiempo real, mediante la recepción de información por un sensor RGB.

Entrenar un perceptrón de manera correcta con los datos recolectados para que nos ayude a identificar entre tres posibles frutas diferentes.

Poder hacer uso del perceptrón para que con los pesos calculados y los datos cargados en un Arduino podamos identificar las tres diferentes frutas en tiempo real.

Programa

Recolección de datos

```
import os
import serial
import time

def arduino_read_data(file):
    encoding = "utf-8"

    arduinoPort = serial.Serial('COM7',9600,timeout=1)
    f = open(file,"w")

    time.sleep(.3)

    for i in range(500):
        getSerialValue = arduinoPort.readline()
        print(getSerialValue)
        f.write(getSerialValue.decode(encoding))

    arduinoPort.close()
    f.close()

arduino_read_data("Limon2.txt")
```

Entrenamiento de los perceptrones

```
import pandas as pd
def escalon(k,pesos,b):
    n=-b
    for i in range(len(k)):
        n=n+(k[i]*pesos[i])
    if(n>=0):
        return 1
    else:
        return 0
def entrenar_perceptron(datos_ent,pesos,b,lamda):
    epocas=0
    errores=True
    while errores and epocas<4000:
        epocas=epocas+1
        print()
        print("Epoca: ",epocas)
        errores=False
        for k,y in datos_ent.items():
            n=escalon(k,pesos,b)
            if(n!=y):
                print("Hay ajuste")
```

```

        print("p1:",k[0],"p2:",k[1],"p3:",k[2])
        errores=True
        e=(y-n)
        delta_b=- (lamda*e)
        print("b:",b,"e",e,"delta_b",delta_b)
        for i in range(len(k)):
            delta_w=lamda*e*k[i]
            pesos[i]=pesos[i]+delta_w
            print("Pesos",i,pesos[i])
    else:
        print("No hay Ajuste")
        print("p1:",k[0],"p2:",k[1],"p3:",k[2])
        e=(y-n)
        delta_b=- (lamda*e)
        print("b:",b,"e",e,"delta_b",delta_b)
    return pesos,b

def clasificar(entrada,pesos,b):
    return escalon(entrada,pesos,b)

def insertar_datos(archivo,archivo2,archivo3):
    datos=pd.read_csv(archivo, sep="," ,
                      header=None)
    datos2=pd.read_csv(archivo2, sep="," ,
                      header=None)
    datos3=pd.read_csv(archivo3, sep="," ,
                      header=None)

    lst = datos.values.tolist()
    lst2 = datos2.values.tolist()
    lst3 = datos3.values.tolist()
    #Se guardan los datos en una lista.

    new_lst = [tuple(x) for x in lst]
    new_lst2 = [tuple(x) for x in lst2]
    new_lst3 = [tuple(x) for x in lst3]
    dataa={}

    for i in new_lst:
        dataa[i]=0
    for i in new_lst2:
        dataa[i]=1
    for i in new_lst3:
        dataa[i]=0
    return dataa

if __name__=="__main__":
    datos_ent=insertar_datos("./tratamientoDatos/csv/NuevaManzana1.csv",
                             "./tratamientoDatos/csv/Mandarina.csv",
                             "./tratamientoDatos/csv/Limon1.csv")

    pesos=[.2,.6,.1]
    b=0.4
    lamda=0.2

```

```

    pesos,b=entrenar_perceptron(datos_ent,pesos,b,lamda)
    print("-----")
    print("Pesos finales: ",pesos)
    print("Bias Final: ",b)
    prueba=(45,60,92)
    print("Probando perceptron: ",str(prueba)+str(clasificar(prueba,pesos,b)))

prueba=(132,152,189)
print("UNO: ",str(prueba)+str(clasificar(prueba, [42.800000000000001,
4.6000000000000016, -28.899999999999984]
,0.4)))
print("DOS: ",str(prueba)+str(clasificar(prueba, [-225.60000000000002,
123.80000000000007, -1.6999999999999886],0.4)))

```

Clasificador por Arduino

```

#include <Wire.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>

#define ANCHO 128
#define ALTO 64

#define OLED_RESET 4
Adafruit_SSD1306 oled(ANCHO, ALTO, &Wire, OLED_RESET);

#define S0 8
#define S1 9
#define S2 12
#define S3 11
#define salidaSensor 10
int Lectura[3];
int R,G,B,F;

float p1p1 = 42.800000000000001;
float p1p2 = 4.6000000000000016;
float p1p3 = -28.899999999999984;

float p2p1 = -225.60000000000002;
float p2p2 = 123.80000000000007;
float p2p3 = -1.6999999999999886;

int P1,P2;

void setup() {
    pinMode(S0,OUTPUT);
    pinMode(S1,OUTPUT);
    pinMode(S2,OUTPUT);
    pinMode(S3,OUTPUT);

    pinMode(salidaSensor,INPUT);

```



```

digitalWrite(S0,HIGH);
digitalWrite(S1,LOW);

Serial.begin(9600);
Wire.begin();
oled.begin(SSD1306_SWITCHCAPVCC, 0x3C);

}

void Dibujar(int R, int G, int B, int F){
    String red = String(R);
    String green = String(G);
    String blue = String(B);
    oled.clearDisplay();
    oled.setTextCursor(WHITE);
    oled.setCursor(0, 0);
    oled.setTextSize(1);
    Fruit(F);

    oled.drawRect(1, 5, 36, 13, WHITE);
    oled.drawRect(1, 21, 36, 13, WHITE);
    oled.drawRect(1, 37, 36, 13, WHITE);
    oled.setCursor(6, 14-7);
    oled.print("Rojo");
    oled.setCursor(4, 32-8);
    oled.print("Verde");
    oled.setCursor(6, 48-8);
    oled.print("Azul");
    oled.setCursor(42, 14-7);
    oled.print(red);
    oled.setCursor(42, 32-8);
    oled.print(green);
    oled.setCursor(42, 48-8);
    oled.print(blue);
    oled.display();
}

void Fruit(int F){
    oled.drawRect(64, 11, 60, 37, WHITE);

    if(F == 0){
        oled.setCursor(68, 33-8);
        oled.print("Mandarina");
    }
    if(F == 1){
        oled.setCursor(79, 33-8);
        oled.print("Limon");
    }
    if(F == 2){
        oled.setCursor(74, 33-8);
        oled.print("Manzana");
    }

    if(F == 3){

```

```

        oled.setCursor(79, 33-8);
        oled.print("Error");
    }

}

int LeerColor(int tiempo_de_espera){
    digitalWrite(S2,LOW);
    digitalWrite(S3,LOW);

    Lectura[0]=pulseIn(salidaSensor,LOW);
    delay(tiempo_de_espera);

    digitalWrite(S2,HIGH);
    digitalWrite(S3,HIGH);

    Lectura[1]=pulseIn(salidaSensor,LOW);
    delay(tiempo_de_espera);

    digitalWrite(S2,LOW);
    digitalWrite(S3,HIGH);

    Lectura[2]=pulseIn(salidaSensor,LOW);
    delay(tiempo_de_espera);

    return Lectura;
}

int Perceptron1(int R, int G, int B ){
    float n = 0.4 + R*p1p1 + G*p1p2 + B*p1p3;
    if (n>=0){
        return 1;
    }else{
        return 0;
    }
}

int Perceptron2(int R, int G, int B ){
    float n = 0.4 + R*p2p1 + G*p2p2 + B*p2p3;
    if (n>=0){
        return 1;
    }else{
        return 0;
    }
}

int Clasificar(int R, int G, int B){
    P1 = Perceptron1(R,G,B);
    P2 = Perceptron2(R,G,B);

    if(P1 == 1){
        if(P2 == 1){
            Serial.println("Fuera de Rango");
            return 3;
        }
    }
}

```

```

    }
    if(P2 == 0){
        Serial.println("Es una limon!!");
        return 1;
    }
}

if(P1 == 0){
    if(P2 == 1){
        Serial.println("Es un mandarina!!");
        return 0;
    }
    if(P2 == 0){
        Serial.println("Es una manzana");
        return 2;
    }
}

}

void loop() {
    int* lectura;
    lectura = LeerColor(200);
    Serial.print(lectura[0]);
    Serial.print(" ");
    Serial.print(lectura[1]);
    Serial.print(" ");
    Serial.println(lectura[2]);

    F = Clasificar(lectura[0],lectura[1],lectura[2]);
    Dibujar(Lectura[0],Lectura[1],Lectura[2],F);
    delay(1000);
}

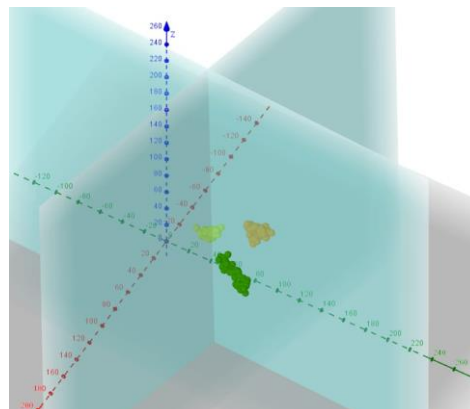
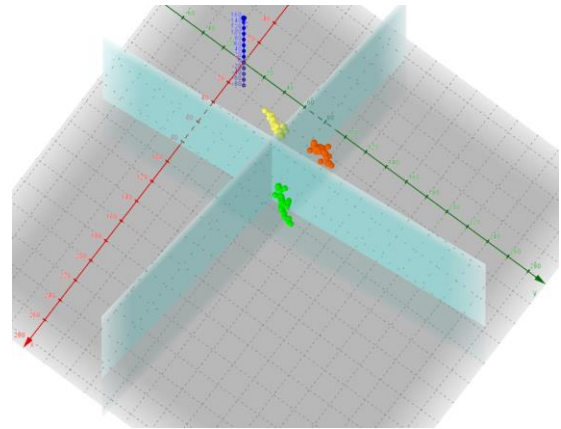
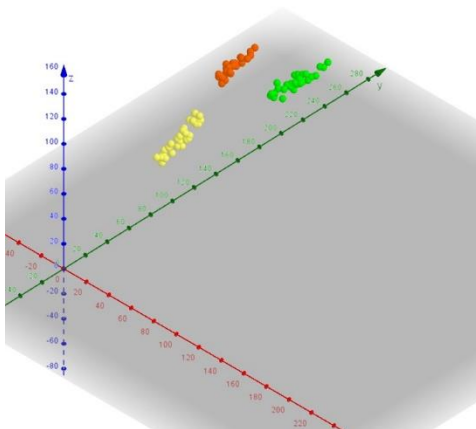
```

Conclusión

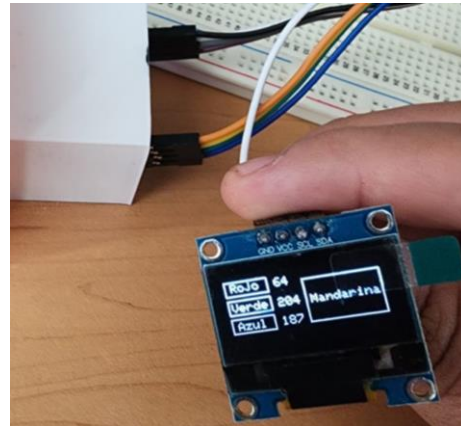
Podemos ver que con los perceptrones estamos haciendo uso del aprendizaje de maquina (Machine Learning) ya que estamos creando un algoritmo que nos ayude a identificar objetos, esto lo hacemos mediante aprendizaje supervisado, con esto nos referimos a que en el mundo general nosotros etiquetamos a los objetos dependiendo del enfoque que se le dé o se necesite.



El algoritmo que ocupamos es para este caso es uno encargado clasificar objetos mediante códigos de color en la escala de colores principales RGB, se le da una entrada a la máquina un conjunto de datos que están previamente etiquetados para que el modelo de entrenamiento aprenda generalizaciones o patrones de las escalas, en este caso usamos una manzana, una mandarina y limones.



Ya en la etapa fina llegamos al periodo de predicción, de manera exitosa, ya que logramos que los perceptrones logran reconocer dichos patrones, haciendo que con los datos que se le dieron para aprender, logra diferenciar o clasificar de manera eficiente los nuevos datos que se le otorguen mediante el sensor, ya que en los resultados de la salida se puede ver que nos logra decir que es la fruta que tiene en el sensor.



Referencias

Soria, Emiliano. (2022). *Inteligencia Artificial*. España: RA-MA S.A. Editorial y Publicaciones. Recuperado 24 de Noviembre de 2023.

Restrepo Leal, D. A., Vilorio Porto, J. P., Robles Algarín, C. A. (2021). *El camino a las redes neuronales artificiales*. Colombia: Editorial Unimagdalena. Recuperado 24 de Noviembre de 2023.

Tablada Claudio Javier, Ariel Torres Germán. (s.f.). *Redes Neuronales Artificiales*. Argentina. Artículo de la Facultad de Matemática, Astronomía y Física (FaMAF) – CIEM (CONICET) Universidad Nacional de Córdoba. Recuperado 25 de Noviembre de 2023.

Amazon Web Services. (s. f.). ¿Qué es una red neuronal?. Amazon Web Services. <https://aws.amazon.com/es/what-is/neural-network/#:~:text=Una%20red%20neuronal%20es%20un,lo%20hace%20el%20cerebro%20humano>. Recuperado 25 de Noviembre de 2023

Arduino. (s. f.). Guía de Referencia de Arduino. Arduino. <https://www.arduino.cc/reference/es/>. Recuperado 16 de Noviembre de 2023.