

DISEÑO Y DESARROLLO DE UNA PRÓTESIS MIOELÉCTRICA.

Introducción.

Este proyecto propone el diseño e implementación de una prótesis de brazo robótico controlada por sensores de electromiografía para resolver el problema de limitación de personas que carecen de una extremidad y la resolución de un problema con impacto social al proporcionar Una prótesis funcional que genera inclusión laboral y social para las personas que no tienen una mano.

Problema.

En la actualidad en México, los índices de personas que han perdido algún brazo por accidente o por causas congénitas va en aumento, la pérdida de un brazo ocasiona problemas para reintegrarse con la sociedad de una manera adecuada, debido a eso existen métodos para intentar sustituir la función de los miembros con la finalidad de dar una mejor calidad de vida.

Sin embargo, en Jalisco, aún es reducido el número de persona que tienen la posibilidad de tener una, ya que estas son muy costosas, existen asociaciones como el DIF donde ayudan a la obtención de ella, pero existe una lista de espera muy larga, que además se debe someter a ciertos estudios socioeconómicos como médicos para ver si son aptos o no.

Justificación.

Las prótesis disponibles en el mercado son limitadas, existen ya empresas mexicanas y extranjeras trabajando en ello en jalisco, pero estas aún siguen siendo elevadas en precio e invasivas en el cuerpo, donde las microelectricas aún son nuevas y como tal no se han implementado como tal, en las personas que sufren de esta discapacidad.

Nosotros proponemos el desarrollo y diseño de una prótesis mioeléctrica funcional la cual pueda ejecutar movimientos básicos de la mano de los dedos para cumplir con alguna función ya sea sujetar algún objeto y que éste tenga cierto grado de precisión a la hora de tomar objetos o ser utilizado ya sea de manera funcional o estética y que el sistema de control no sea tan invasivo para el usuario es decir que todo sea embebido en la misma prótesis y que no requiera de conexión a un computador. Además, propone ser de código abierto, bajo costo y fácil de implementarlo con la finalidad de satisfacer las necesidades de la persona y favorecerle su calidad de vida.

Objetivo General:

- Diseñar una prótesis semejante a la morfología y con la locomoción esencial de la mano humana.
- Crear un dispositivo de adaptación humano-maquina mioeléctrico-no invasivo.

Objetivos particulares:

- Controlar la mano robótica, con redes neuronales artificiales.
- Desarrollar una red neuronal para identificar conjuntos de señales bioeléctricas provenientes del brazo.

Módulos del proyecto:

Módulo Electrónica Digital

- Procesamiento de señales mioeléctricas usando el microcontrolador ESP32.
- Implementación y configuración de una red neuronal usando las herramientas de desarrollo de Tensor Flow Lite.
- La idea es implementar la red neuronal de 1 entrada a 5 salidas con una función de activación sigmoide, la red estará embebida en el código del microcontrolador, con esto los pesos de dicha neurona y nuestros clasificadores así como la obtención de datos serán con ayuda de Matlab y Python para finalmente exportar dichos pesos a el ESP32.

Módulo Electrónica Analógica

- Diseño y fabricación de PCB para las etapas de alimentación de motorreductores.
- Implementación y ajuste de motorreductores de 6v GA12-N20.
- Diseño y fabricación de PCB para las etapas de recarga y protección de las baterías recargables DTP de 1850mA/h de 3.7V.
- Implementación de etapas de regulación de voltaje para los diferentes niveles requeridos por los componentes del sistema (3.3V, 6V, 12V y -12V) usando fuentes buck boost.

Módulo de Comunicaciones

- Configuración de un servidor remoto donde se almacenarán datos de usuario y biométricos para la prótesis con el fin de poder realizar entrenamientos posteriores de la red neuronal del sistema para adecuar el control y renar detalles.

Módulo de Instrumentación y control

- Control de los actuadores de brazo y dedos en respuesta a los impulsos de las señales mioeléctricas proporcionadas por el muñón del brazo, los actuadores serán motorreductores de 6v con un encoder para la retroalimentación del mismo, ya que contará con un procedimiento de calibración y, de posición precisos debido al encoder empleado así que las posiciones de dedo deseadas sean representadas correctamente.

Estado del Arte

Prótesis I-Limb Ultra.

Una de las más utilizadas en la actualidad en diferentes países cuenta con la mayoría de los movimientos básicos de la mano humana, permitiendo la integración del amputado a la vida laboral y mejorando su condición de vida. Los dedos de la prótesis se controlan de manera independiente, el dedo pulgar puede rotar hasta 90° , realiza sujeciones de alta precisión y fuerza.



Figura 1: Prótesis I-Limb

Prótesis Michelangelo.

Esta prótesis se caracteriza por tener una gran gama de movimientos de precisión, debido al control que implementa y a su mecanismo de locomoción que aportan gran fuerza de sujeción, sus dedos son controlados al igual que la anterior, de manera independiente, además cuenta con los grados de libertad de la muñeca, en su exterior está recubierta por elastómero de silicona dándole

resistencia al agua, en su interior por acero y duraluminio de alta resistencia.

Figura 2: Prótesis Michaelangelo



Prótesis Bebionic

La Bebionic es muy semejante a la I-Limp figura; Desarrollada por Steeper RS, con la finalidad de emular a esta última, pero a “bajo costo”, aproximadamente \$10,000 dólares. En su última versión, la Bebionic tiene movimientos de articulación en todos los dedos, el pulgar en oposición se coloca de manera manual en extensión y flexión, dando quince diferentes posiciones de sujeción.

La prótesis es controlada por señales bioeléctricas obtenidas a partir de la contracción de los músculos del paciente, por tal motivo entra en la clasificación de prótesis “mioeléctrica”, por último, tiene cinco actuadores con cinco velocidades fijas cada uno, diseñados para bajo consumo de energía.



Figura 3: Prótesis Bebionic

Prótesis CyberHand

Es la prótesis de mayor tecnología moderna y consta, como su nombre lo indica, es una prótesis de conexión invasiva [4], obteniendo la información directa desde el cerebro mediante electrodos instalados en el paciente por medio de cirugía, y de igual forma recoge información externa como presión o temperatura a la que está sometida la prótesis.

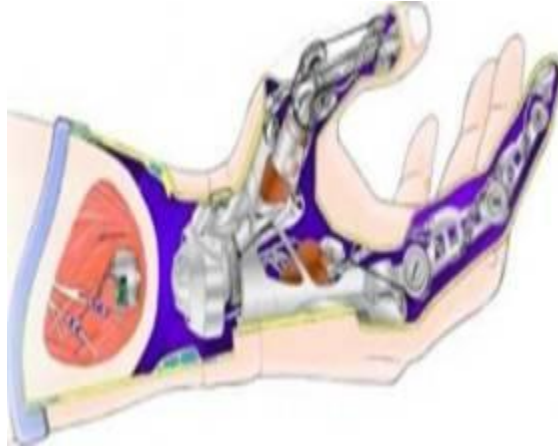


Figura 1.9: prótesis CyberHand

Solución propuesta

Implementación de una prótesis mioeléctrica a un bajo costo, con la finalidad de ayudar a complementar funciones básicas, que ayuden a mejorar la calidad de vida.

Mediante el uso de una red neuronal para mejorar el sistema de control mecánico de la mano.

Esta se pretende que sea más ligera y menos invasiva con material PLA, el cual es más ligero y fácil de montar en el brazo donde será puesta.

Desarrollo e implementación

Recursos y presupuesto

COSTOS											
Itm	Description	Supplier	Location	Unit Cost USD	Quantity	Sub-total Cost USD	Shipping Cost USD	Total Cost USD	Total Cost MXN	Total Cost MXN per person	
1	1000:1 Micro	Pololu	USA	\$ 25.95	6	\$ 155.70	\$ 20.45	\$ 253.73	\$ 5,678.48	3	
2	Magnetic Encoder	Pololu	USA	\$ 7.95	6	\$ 47.70					
3	6-Pin Female	Pololu	USA	\$ 1.49	6	\$ 8.94					
4	2990 DRV8833	Pololu	USA	\$ 3.49	6	\$ 20.94					
5	Servomotor	Ali Express	China	\$ 10.12	1	\$ 10.12	\$ 5.24	\$ 15.36	\$ 343.76		
6	Servomotor	Ali Express	China	\$ 9.50	1	\$ 9.50	\$ 3.07	\$ 12.57	\$ 281.32		
7	Electrodos El	Ali Express	China	\$ 17.31	3	\$ 51.93	\$ 2.54	\$ 54.47	\$ 1,219.04		
8	Sensor Presi	Ali Express	China	\$ 1.58	2	\$ 3.16	\$ 0.86	\$ 4.02	\$ 89.97		
9	ESP 32 Micro	Ali Express	China	\$ 3.57	3	\$ 10.71	\$ 3.65	\$ 14.36	\$ 321.38		
10	Ligas ortodo	Mercado Lib	México	\$ 8.02	1	\$ 8.02	\$ -	\$ 8.02	\$ 179.49		
11	1 Caja de tor	Mercado Lib	México	\$ 27.78	1	\$ 27.78	\$ -	\$ 27.78	\$ 621.72		
						\$ 354.50	\$ 35.81	\$ 390.31	\$ 8,735.14	\$ 2,911.71	

Desarrollo

10 octubre 2020 Se toman las medidas necesarias para el diseño de prótesis

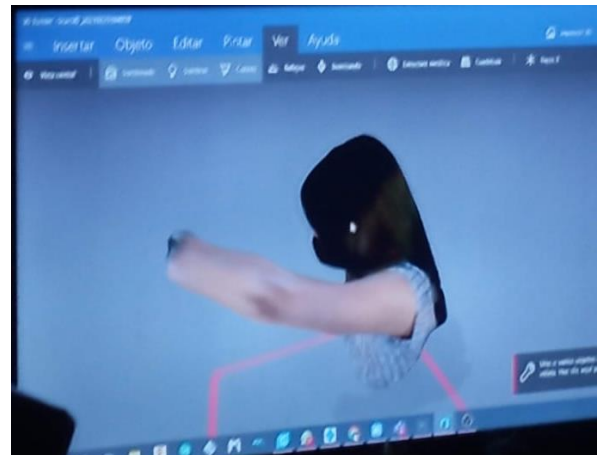
Se toman las medidas necesarias de la persona voluntaria, se comienza a dimensionar y pensar como quedaría el diseño y que no sea tan invasiva para ella. Y quede de cierta manera práctico y menos rígido a comparación del resto de prótesis.

**24 de
octubre del
2020**



Dimensiones del diseño en 3D de la mano

Con las medidas obtenidas se comienza a diseñar el soporte del brazo , para que este sea propio de ella , se hace el diseño en 3d , mediante el software de autocad.

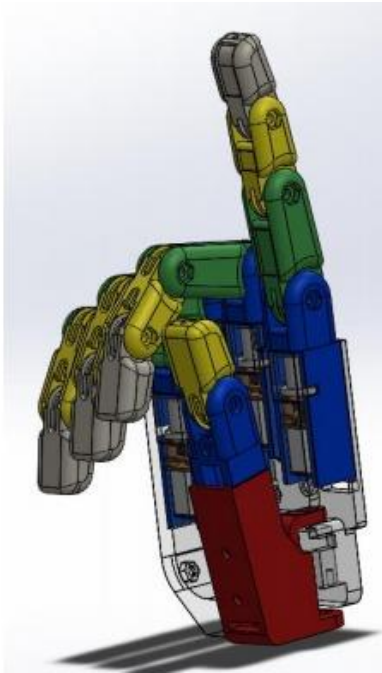


3 de noviembre del 2020 Diseño en Autocad

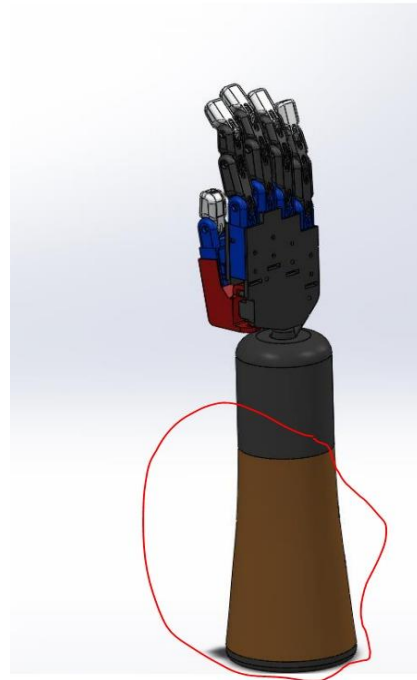
A partir de las dimensiones y medidas obtenidas en el escaneo 3d para adecuarlos a el antebrazo de la voluntaria, junto con los actuadores que llevará cada dedo , las adecuaciones que tendrá para el cableado correspondiente del resto del circuito electrónico.

25 noviembre del 2020 Diseño de prótesis terminado

Queda listo el diseño en FreeCAD, para proceder con la impresión

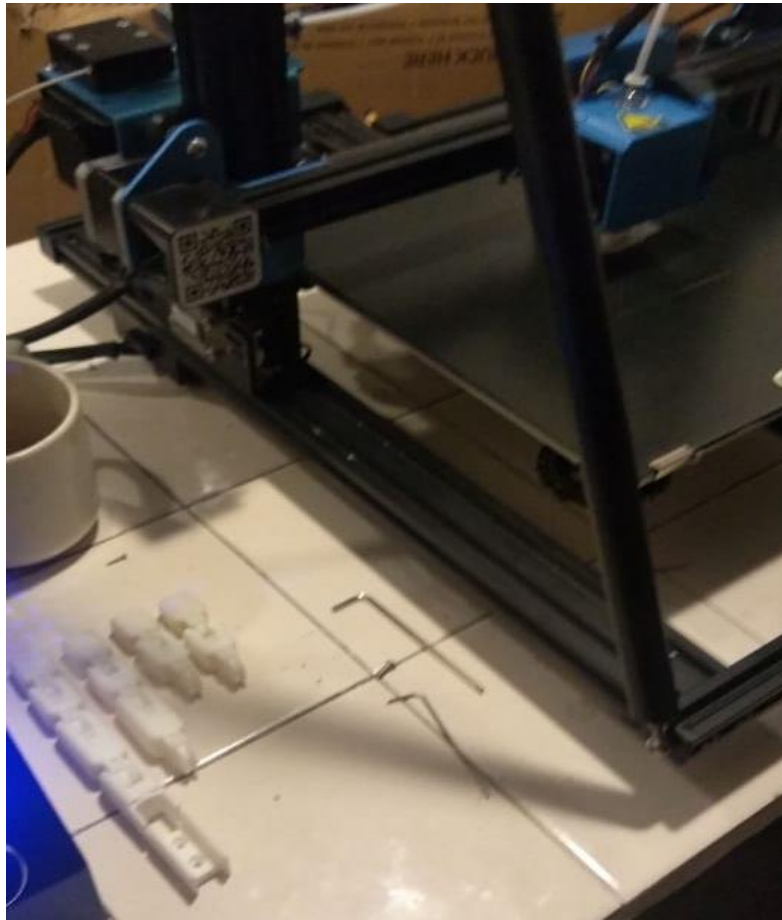


30



noviembre del 2020

Se realiza las primeras impresiones



cabe mencionar que para tener la mano completa duró una semana aproximadamente.

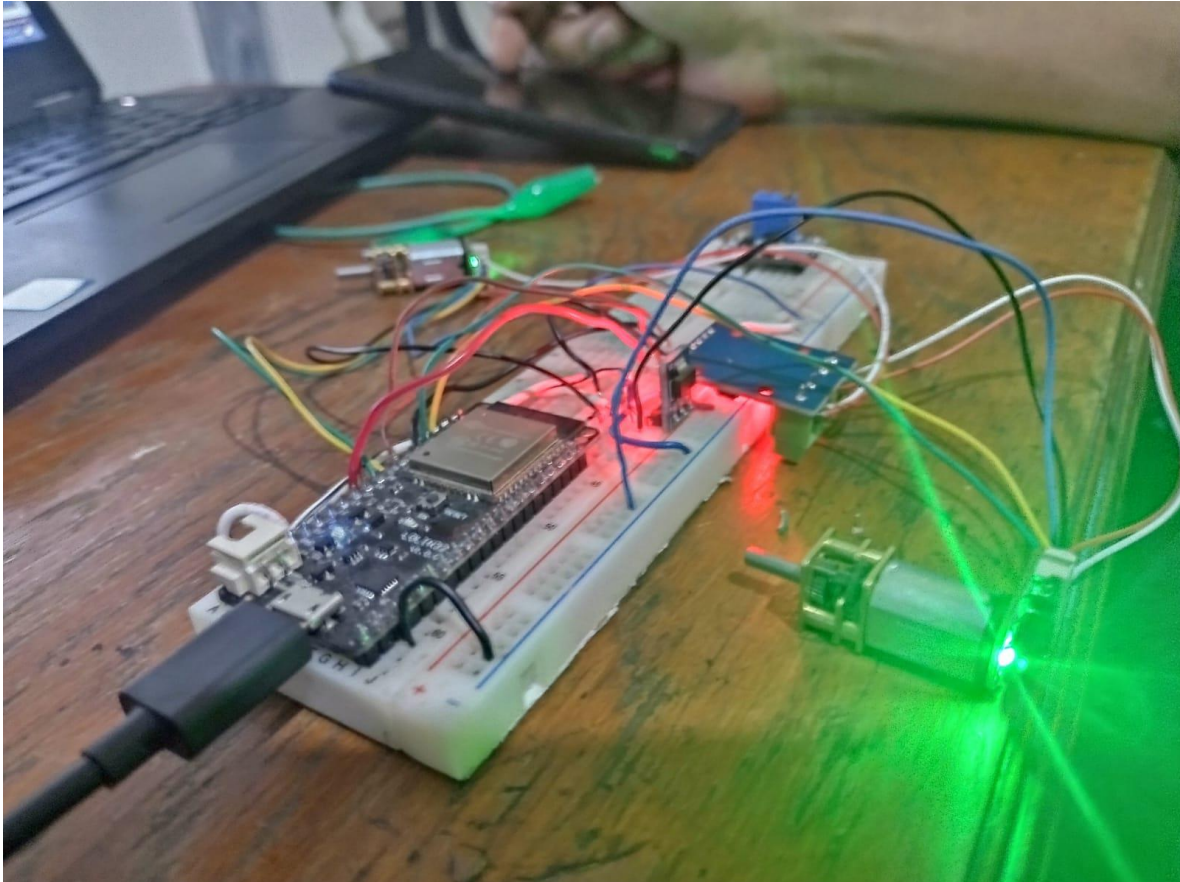
15 enero del 2021.

Se realiza el ensamble de la mano completa con los actuadores.



16 de abril de 2021 Desarrollamos un sistema de control para cada uno de los motores

El programa se realiza en lenguaje C para el microcontrolador Esp32, con el que controlaremos el giro y movimiento de los motores que tendrá cada dedo, se realizan las primeras pruebas con los motorreductores.



30 abril del 2021 Continuación programación del Esp32.

Se desarrolla el programa para controlar los 5 motores con la detección de posición por encoder se logra controlar los 5 motores a la vez superando la objeción de tener múltiples interrupciones a la vez.

10 mayo del 2021 Continuación programación del Esp32.

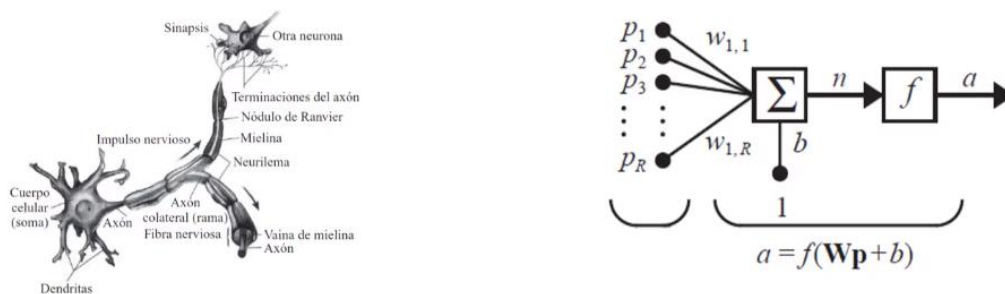
Se agrega la función de calibración de los motores que lleguen a la posición 0 y guardar la posición exacta la cual podrá ser ejecutada para resolver futuros problemas en la posición de los dedos en la prótesis al reiniciar el sistema.

15 mayo del 2021 Continuación programación del Esp32.

Se añade el protocolo de comunicación al Esp32 debido a que este hará las veces de esclavo recibirá las órdenes del ESP32 Maestro el cual contiene la red neuronal, de esta forma se potencia la capacidad de procesamiento de la red neuronal limitándose a enviar las instrucciones de posición en los motores de manera serial y el esclavo se limitará a administrar el manejo de los motores de manera independiente al recibir un valor individual para cada motor. El protocolo consiste en recibir los valores individuales separados por comas.

18 mayo del 2021 Modelado de la red neuronal, función de activación sigmoideal y clasificación de las señales obtenidas por el Electro miógrafo.

El modelado de las redes neuronales son una simplificación de una neurona biológica,



La suma neta: $n = w_{11}p_1 + w_{12}p_2 + \dots + w_{1R}p_R + b$

Forma matricial: $n = \mathbf{Wp} + b$

La salida de la neurona se calcula como: $a = f(\mathbf{Wp} + b)$

Las entradas de la neurona son representadas por $P_1, P_2 \dots P_r$ y es la información que recibe la red.

Los pesos representados por $w_{1,1}$ hasta $w_{1,R}$ si el sistema es de una neurona, dichos pesos son simples valores numéricos que establecen una influencia de la entrada en una salida deseada, dichos pesos también son considerados como el conocimiento adquirido de nuestra red y, por último, b representa el bias que en inglés significa sesgo, esto quiere decir que nosotros aplicamos un sesgo a la red para que actúe de una u otra manera, de igual forma que influyen los pesos se podría decir.

Para el modelado de la red neuronal en una arquitectura de Arduino programada en el ESP32 haremos una prueba con valores arbitrarios y someteremos dichos códigos a una simulación tanto en Arduino como en Python para comprobar si se logra llegar a los mismos resultados en un tiempo considerablemente similar.

$$R = 3$$

$$\begin{array}{ll} p_1 = 0.51 & w_{11} = 0.57 \\ p_2 = 0.59 & w_{12} = -0.79 \\ p_3 = -0.15 & w_{13} = 0.75 \\ & b = 0.8 \end{array} \quad \mathbf{p} = \begin{bmatrix} p_1 \\ p_2 \\ p_3 \end{bmatrix} \quad \mathbf{W} = [w_{11} \quad w_{12} \quad w_{13}] \quad \mathbf{b} = [0.8]$$

$$a = f(\mathbf{Wp} + b)$$

La suma neta: $n = (0.57 \times 0.51) + (-0.79 \times 0.59) + (0.75 \times -0.15) + 0.8$

$$n = 0.2907 - 0.4661 - 0.1125 + 0.8$$

$$n = 0.5121$$

La salida de la neurona se calcula como: $a = f(0.5121)$

Cálculo de salida de una neurona de 3 entradas en Arduino:

```
double p1 = 0.51;
double p2 = 0.59;
double p3 = -0.15;

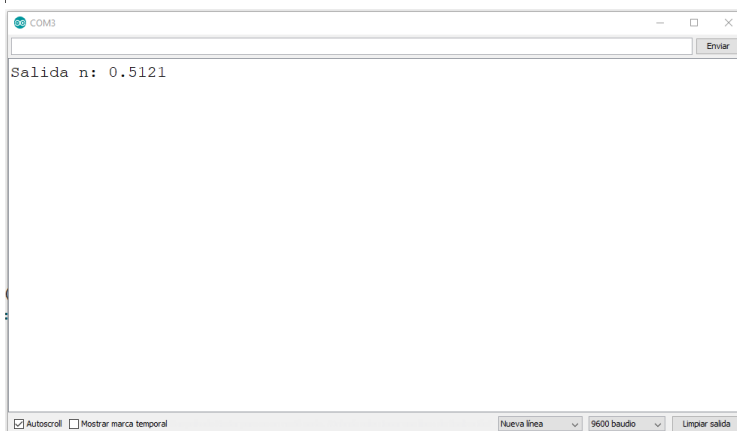
double w11 = 0.57;
double w12 = -0.79;
double w13 = 0.75;
double b = 0.8;

double n = 0;

void setup() {
  Serial.begin(9600);

  n = (w11*p1)+(w12*p2)+(w13*p3)+b; // <--- Red neuronal de 3 entradas
  Serial.print("Salida n: ");
  Serial.println(n,4) ;
}

void loop() {
}
```



Cálculo de salida de una neurona de 3 entradas en Python:

```
*Modelo_simplificado_red_neuronal_Tensorflow.py - D:\D Desktop\CUCEI\B202...
File Edit Format Run Options Window Help

import tensorflow as tf

p = tf.constant([0.51, 0.59, -0.15], shape=[3, 1])

W = tf.constant([0.57, -0.79, 0.75], shape=[1, 3])

b = tf.constant([0.8], shape=[1, 1])

n = tf.matmul(W, p)+b # n = Wp+ b

print(f"Salida n: {n}")

Ln: 14 Col: 0

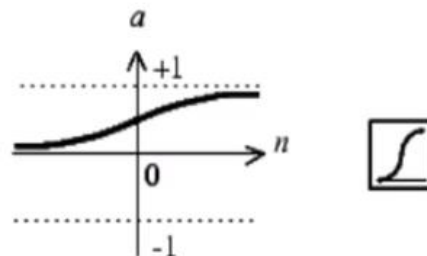
Python 3.7.8 Shell
File Edit Shell Debug Options Window Help
Python 3.7.8 (tags/v3.7.8:4b47a5b6ba, Jun 28 2020, 08:53:46) [MSC v.1916 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: D:\D Desktop\CUCEI\B2020B\Modular\udem\Modelo_simplificado_red_neuronal_Tensorflow.py
Salida n: [[0.5121]]
>>>

Ln: 6 Col: 4
```

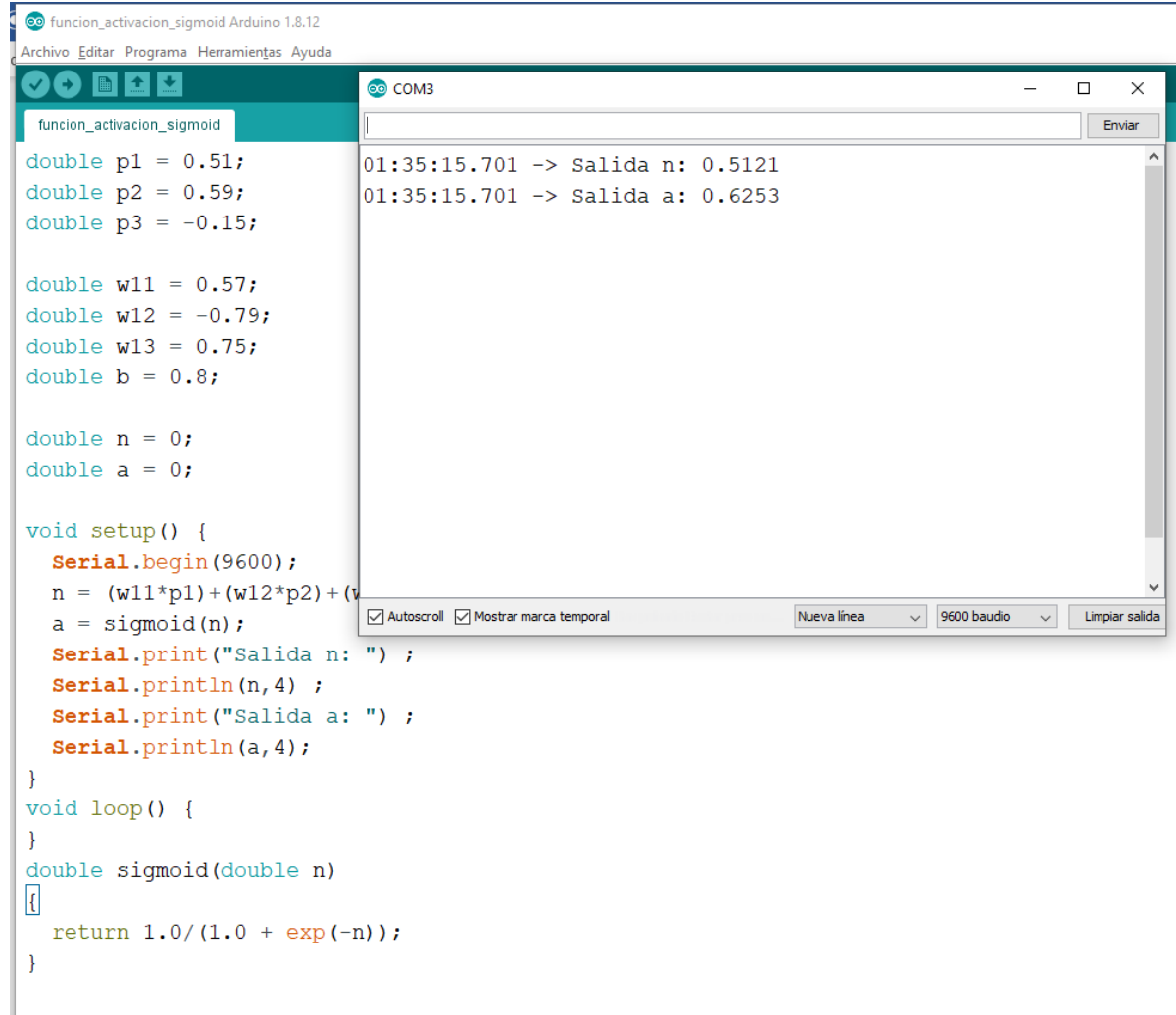
Como logramos observar nuestra salida calculada es totalmente igual tanto en Python como en Arduino, siendo favorable proceder.

Como función de transferencia o función de activación hemos decidido utilizar la función sigmoideal ya que nos presenta la ventaja de reducirnos la salida a un rango de 0 a 1 siendo que de entrada se le pueden insertar valores de menos infinito a infinito.

$$a = \frac{1}{1 + e^{-n}}$$



Implementación de función sigmoïdal en Arduino



The screenshot shows the Arduino IDE interface. The left pane displays the code for a file named 'funcion_activacion_sigmoid'. The code defines three input variables (p1, p2, p3) and three weight variables (w11, w12, w13), along with a bias variable (b). It then calculates a weighted sum 'n' and passes it to a 'sigmoid' function. The 'sigmoid' function is defined as $1.0 / (1.0 + \exp(-n))$. The 'setup' function initializes the serial port at 9600 baud and prints the calculated values of 'n' and 'a' (the result of the sigmoid function) to the serial monitor. The 'loop' function is empty. The right pane shows the serial monitor output, displaying the calculated values of 'n' and 'a' at the time 01:35:15.701.

```
funcion_activacion_sigmoid

double p1 = 0.51;
double p2 = 0.59;
double p3 = -0.15;

double w11 = 0.57;
double w12 = -0.79;
double w13 = 0.75;
double b = 0.8;

double n = 0;
double a = 0;

void setup() {
  Serial.begin(9600);
  n = (w11*p1)+(w12*p2)+(w13*p3)+b;
  a = sigmoid(n);
  Serial.print("Salida n: ") ;
  Serial.println(n,4) ;
  Serial.print("Salida a: ") ;
  Serial.println(a,4);
}

void loop() {
}

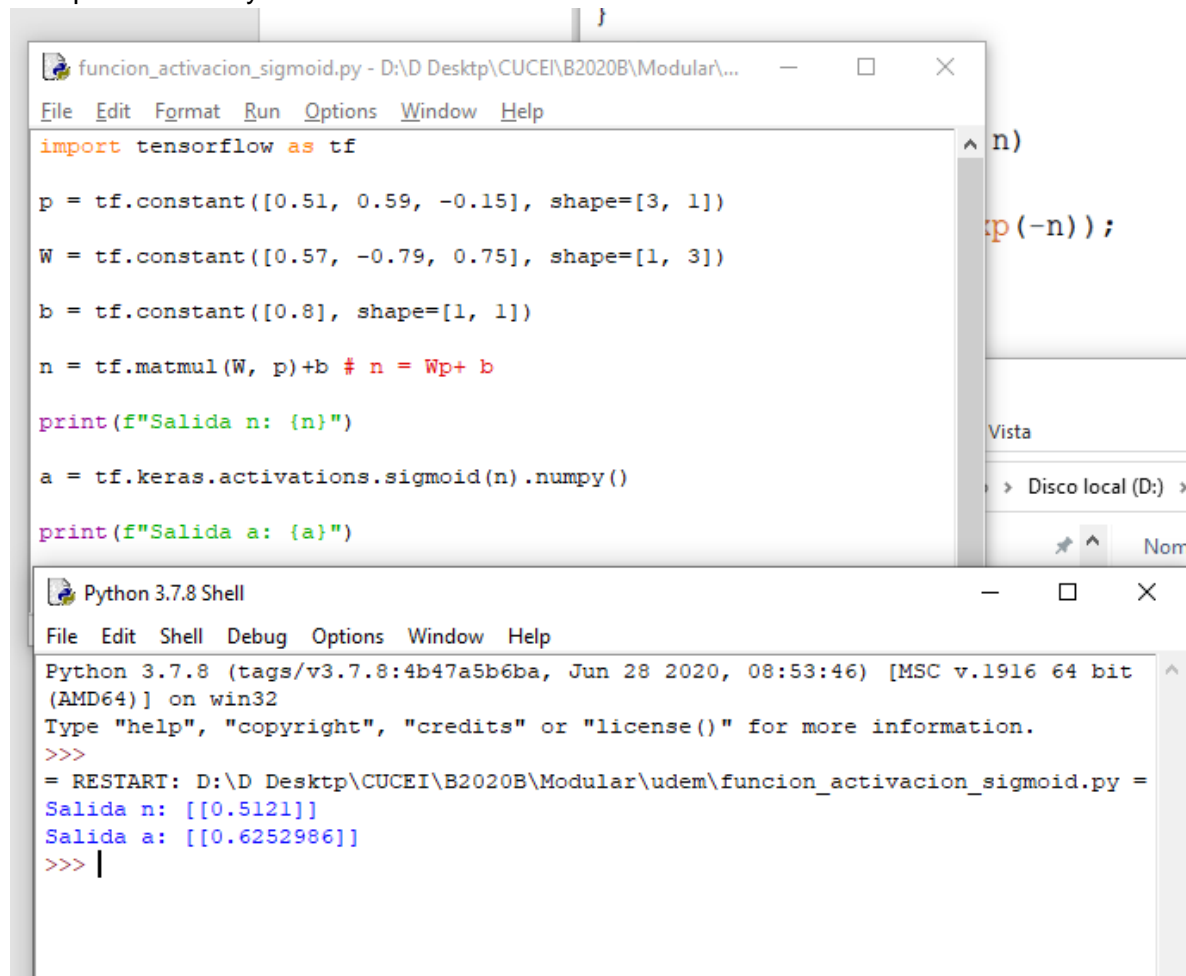
double sigmoid(double n)
{
  return 1.0/(1.0 + exp(-n));
}
```

COM3

01:35:15.701 -> Salida n: 0.5121
01:35:15.701 -> Salida a: 0.6253

☒ Autoscroll ☒ Mostrar marca temporal Nueva línea 9600 baudio Limpiar salida

Comparativa en Python



The image shows a screenshot of a Python script and its execution. The script, named `funcion_activacion_sigmoid.py`, is located in the directory `D:\D Desktop\CUCEI\B2020B\Modular\...`. It imports `tensorflow` as `tf` and defines three constant tensors: `p` (shape [3, 1]), `W` (shape [1, 3]), and `b` (shape [1, 1]). It then calculates `n = tf.matmul(W, p) + b` (commented as `n = Wp + b`) and prints the result. Finally, it applies the sigmoid activation function using `tf.keras.activations.sigmoid(n).numpy()` and prints the result.

```
funcion_activacion_sigmoid.py - D:\D Desktop\CUCEI\B2020B\Modular\...
File Edit Format Run Options Window Help

import tensorflow as tf

p = tf.constant([0.51, 0.59, -0.15], shape=[3, 1])
W = tf.constant([0.57, -0.79, 0.75], shape=[1, 3])
b = tf.constant([0.8], shape=[1, 1])

n = tf.matmul(W, p)+b # n = Wp+ b

print(f"Salida n: {n}")

a = tf.keras.activations.sigmoid(n).numpy()

print(f"Salida a: {a}")
```

The execution window, titled "Python 3.7.8 Shell", shows the output of the script. It indicates that the script was restarted and provides the following output:

```
Python 3.7.8 (tags/v3.7.8:4b47a5b6ba, Jun 28 2020, 08:53:46) [MSC v.1916 64 bit
(AMD64)] on win32
Type "help", "copyright", "credits" or "license()" for more information.
>>>
= RESTART: D:\D Desktop\CUCEI\B2020B\Modular\udem\funcion_activacion_sigmoid.py =
Salida n: [[0.5121]]
Salida a: [[0.6252986]]
>>> |
```

Se logra observar que ambas salidas y funciones de activación tienen el mismo resultado por lo tanto son bastante similares.

Bibliografía

- [1] J. Pedreño Molina, A. Guerrero González y J. López Coronado, "Estudio de los Sensores táctiles artificiales Aplicados a la robótica de agarre", pág. 9. [En línea] .Available: <http://www.ceautomatica.es/old/actividades/jornadas/XXI/documentos/003.pdf>
- [2] P. Dudley S., Childress, "Aspectos históricos de prótesis del miembro accionado," pp. 2-13. [En línea]. Available: <http://www.oandplibrary.org/cpo/pdf/198501002.pdf>
- [3] J. Pelletier, "Touch bionics i-limb prostheses," Biomedical Engineering, vol. Second Presentation, April 2013. [Online]. Available: <http://www.ele.uri.edu/Courses/bme181/S13/1JustinP2.pdf>
- [4] MEC 11 Raising the Standard, "Bebionic prosthetic design," p. 4, August 2011. [Online]. Available: <http://dukespace.lib.duke.edu/dspace/bitstream/handle/10161/4733/4320Medynski.pdfsequence=1>
- [5] C. A. Quinays Burgos, "Diseño y construcción de una prótesis robotica de mano funcional adaptada a varios agarres," Tesis de Maestría, Universidad del cauca, Facultad de Ingeniería Electrónica y Telecomunicaciones, Popayan, Colombia, 2010.
- [6] J. L. Loaiza and N. Arzola, "Evolución y tendencias en el desarrollo de prótesis de mano," no. 169, pp. 191–200, Agosto 2011.[Online]. Available: <http://www.scielo.org.co/pdf/dyna/v78n169/a22v78n169.pdf>

[7] C. A. Quinays Burgos, "Diseño y construcción de una prótesis robotica de mano funcional adaptada a varios agarres," Tesis de Maestría, Universidad del cauca, Facultad de Ingeniería Electrónica y Telecomunicaciones, Popayan, Colombia, 2010.