

PERCEPTRON EN POO Y ARDUINO

Castillo Molano Sergio de Jesus¹[scastillom2200@alumno.ipn.mx] Flores Sotelo Miguel Alejandro²[mflo-
ress2200@alumno.ipn.mx] Flores Mora Juan Carlos³[jfloresm2200@alumno.ipn.mx]

¹ INSTITUTO POLITECNICO NACIONAL

² Unidad Profesional Interdisciplinaria de Ingenieria Campus Tlaxcala
uppit@ipn.com.mx

1 Indice

- Introduccion
- Objetivo
- Marco teorico
- Desarrollo
- Resultados POO y Arduino (mostrados en clase)
- Conclusion

1.1 Introduccion

El perceptrón es un algoritmo de aprendizaje supervisado utilizado en el campo del machine learning. Fue propuesto por Frank Rosenblatt en 1957 y es considerado uno de los primeros modelos de redes neuronales artificiales.

Funciona tomando las entradas, multiplicándolas por pesos asociados y luego sumándolas. Esta suma ponderada se pasa a través de una función de activación, comúnmente una función de paso, para producir la salida.

1.2 Objetivo

El objetivo de una práctica de un perceptrón que implementa la función de la compuerta lógica OR en Arduino es diseñar y construir un circuito electrónico que simule el comportamiento de una compuerta lógica OR utilizando un perceptrón implementado en el microcontrolador de la placa Arduino y potenciómetros para variar los pesos y rectificar-comparar con los resultados del programa implementado en POO.

1.3 Marco teorico

Perceptron: Clasificador binario que toma un vector de entrada y produce una única salida binaria, que puede ser interpretada como una clase o una decisión.

POO: Paradigma de programación que se basa en el concepto de "objetos". Estos objetos son entidades que combinan datos, en forma de atributos o propiedades, y código, en forma de métodos o funciones, que operan sobre esos datos.

Arduino: Plataforma de hardware de código abierto que incluye hardware y software diseñados para facilitar el desarrollo de proyectos electrónicos y la creación de prototipos. Consiste en una placa de circuito impreso con un microcontrolador y un entorno de desarrollo integrado (IDE) que se utiliza para escribir y cargar código en la placa.

Potenciometro: Componente electrónico que tiene la capacidad de variar su resistencia eléctrica en función de la posición de un cursor o una perilla. Esto permite ajustar manualmente el valor de la resistencia dentro de un rango determinado.

Resistencia: componente pasivo diseñado para limitar o controlar el flujo de corriente eléctrica en un circuito. Se compone de un material conductor que se opone al paso de la corriente, lo que genera una caída de voltaje en el componente.

Boton pull-up: Dispositivo de entrada que permite al usuario enviar una señal a un sistema mediante una acción física, generalmente presionando o pulsando el botón.

1.4 Desarrollo

Para implementar esta práctica, necesitarás los siguientes materiales:

Placa Arduino

4 Resistencias

2 Potenciómetros 5Kohms

Protoboard

Cables de conexión (macho-macho)

2 Botones pull-up

Conectamos los componentes según el siguiente esquema:

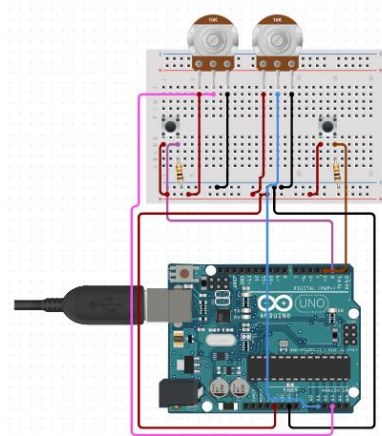


Figura 1.1 Diagrama de Conexiones Arduino-protoboard

Se conectan los potenciómetros a los pines analógicos de la placa Arduino para ajustar los pesos del perceptrón.

Se conecta los botones a los pines digitales de entrada de la placa Arduino para mostrar y obtener las entradas de la compuerta lógica OR.

Paso 2: Programación del perceptrón en Arduino

Paso 3: Verificación y ajuste

Verificamos que el circuito esté funcionando correctamente ajustando los potenciómetros para simular diferentes combinaciones de entradas.

Compara los resultados mostrados por el led integrado en la placa con resultados esperados de la compuerta lógica OR.

Si es necesario, ajustamos los pesos del perceptrón mediante los potenciómetros hasta que los resultados coincidan con las expectativas.

Paso 4: Análisis de resultados

1.5 Resultados POO

```
PS C:\Users\sergi\OneDrive\Documentos\PARADIGMAS DE PROGRAMACION> & C:/Users/sergi/AppData/Local/Programs/Python/Python312/python.exe "c:/Users/sergi/OneDrive/Documentos/PARADIGMAS DE PROGRAMACION/perceptron.py"
[-1.  0.5  1.5  1. ]
[[-0.  0.  0.015  0.01 ]
 [-0.  0.005  0.  0.01 ]]
[0.025 0.015]
[-0.5, 0.5]
[-0.475 0.515]
running one iteration ...
predicted: [-1, 1, -1, 1]
real: [-1  1  1  1]
current weights [-0.475  0.515]
new weights [-0.2315  0.6595]
```

```
predicted: [-1, 1, 1, 1]
real: [-1  1  1  1]
current weights [0.37665392 0.85016785]
new weights [0.41630635 0.84246889]
-----
running one iteration ...
predicted: [-1, 1, 1, 1]
real: [-1  1  1  1]
current weights [0.41630635 0.84246889]
new weights [0.44879819 0.83234448]
-----
running one iteration ...
predicted: [-1, 1, 1, 1]
real: [-1  1  1  1]
current weights [0.44879819 0.83234448]
new weights [0.47580411 0.82099576]
-----
PS C:\Users\sergi\OneDrive\Documentos\PARADIGMAS DE PROGRAMACION>
```

1.6 Conclusiones

En esta práctica, implementamos un perceptrón en Arduino para simular una compuerta lógica OR. A través de la conexión de potenciómetros para ajustar los pesos del perceptrón y botones pull-up para proporcionar las entradas, pudimos observar cómo el perceptrón aprende y toma decisiones en función de las entradas recibidas.

Al analizar los resultados obtenidos durante la práctica, pudimos comprobar la capacidad del perceptrón para aprender y ajustar sus pesos en función de las entradas proporcionadas. La interacción entre el hardware (Arduino) y el software nos permitió comprender de manera práctica los principios detrás del funcionamiento de los algoritmos de aprendizaje automático.

Además, al comparar los resultados obtenidos con los resultados esperados de la compuerta lógica OR, pudimos verificar la eficacia y precisión de nuestro modelo de perceptrón implementado en Arduino.

Referencias

1. Rosenblatt, F. (1958). The perceptron: A probabilistic model for information storage and organization in the brain. *Psychological Review*, 65(6), 386-408.
2. Minsky, M., & Papert, S. A. (1969). *Perceptrons: An introduction to computational geometry*. MIT Press.
3. Rosenblatt, F. (1957). The perceptron--a perceiving and recognizing automaton (Report No. 85-460-1). Cornell Aeronautical Laboratory. (Technical Report).