1. **OPCIÓN**

Se puede utilizar el protocolo Modbus para comunicarte entre tu implementación en SIMATIC Step 7 y tu modelo en Python. El protocolo Modbus es un estándar de comunicación ampliamente utilizado en la automatización industrial.

Aquí hay un ejemplo de cómo podrías implementar la comunicación utilizando Modbus entre SIMATIC Step 7 y tu modelo en Python:

**En SIMATIC Step 7:**

* Configura un dispositivo Modbus maestro en tu programa Step 7 para comunicarte con tu modelo en Python. Esto implica establecer la configuración de comunicación, como la dirección IP y el puerto del dispositivo Modbus esclavo (que sería tu computadora que ejecuta Python con el servidor Modbus).
* Define variables en tu programa Step 7 para almacenar los valores de entrada que serán enviados al modelo y las variables para almacenar los valores de salida que serán recibidos desde el modelo.

**En Python:**

* Utiliza una biblioteca de Python compatible con Modbus (por ejemplo, pymodbus, modbus-tk, minimalmodbus) para crear un servidor Modbus que esté a la escucha de las solicitudes del dispositivo Modbus maestro (que sería tu implementación en SIMATIC Step 7).
* Configura el servidor Modbus para recibir los valores de entrada desde SIMATIC Step 7 y enviar los valores de salida calculados por el modelo de regreso a SIMATIC Step 7.
* En la lógica del servidor Modbus, puedes utilizar tu modelo previamente entrenado para procesar los valores de entrada y obtener los valores de salida correspondientes.
* Utiliza las funciones proporcionadas por la biblioteca Modbus para manejar las solicitudes y respuestas Modbus, y para leer y escribir registros Modbus.

Ten en cuenta que la implementación detallada de la comunicación Modbus dependerá de la biblioteca específica que elijas y de los requisitos de tu entorno de automatización industrial. Es importante consultar la documentación de la biblioteca Modbus que utilices para comprender cómo configurar el servidor Modbus y manejar las solicitudes y respuestas.

Recuerda que, en este enfoque, necesitarás asegurarte de que tanto tu implementación en SIMATIC Step 7 como tu servidor Modbus en Python estén en ejecución y puedan comunicarse entre sí a través de la red. Además, deberás tener en cuenta los aspectos de seguridad y control de acceso en tu implementación para proteger tus sistemas y datos industriales.

pip install pycomm3

from pycomm3 import LogixDriver

# Crear una instancia del driver Logix para comunicarse con un controlador Logix

with LogixDriver('192.168.1.1') as plc:

# Definir las variables de entrada y salida en el controlador Logix

input\_tags = ['Input1', 'Input2', 'Input3']

output\_tags = ['Output1', 'Output2']

# Leer los valores de entrada desde SIMATIC Step 7

input\_values = plc.read(input\_tags)

# Procesar los valores de entrada utilizando tu modelo

# (aquí debes agregar la lógica específica de tu modelo)

# Escribir los valores de salida de regreso a SIMATIC Step 7

plc.write(output\_tags, output\_values)

**En SIMATIC Step 7:**

* Configura tu dispositivo Modbus esclavo para comunicarse con el servidor Modbus en Python. Esto implica establecer la dirección IP y el puerto del servidor Modbus en Python.
* Define las variables en tu programa Step 7 para almacenar los valores de entrada que se enviarán al modelo y las variables para almacenar los valores de salida que se recibirán desde el modelo.
* Configura tu implementación en Step 7 para leer los valores de entrada desde el dispositivo Modbus esclavo y escribir los valores de salida en el dispositivo Modbus esclavo.

Recuerda que los detalles exactos de implementación dependerán de tu entorno y requisitos específicos. Es importante consultar la documentación de pycomm3 y de SIMATIC Step 7 para comprender cómo configurar la comunicación Modbus y utilizar las funciones y métodos proporcionados por estas herramientas. Además, asegúrate de considerar aspectos de seguridad y control de acceso para proteger tus sistemas y datos industriales.

1. **OPCIÓN**

Obtener las matrices de pesos y sesgos de todas las capas del modelo. Se utilizan variables locales para almacenar los pesos y los sesgos del modelo. Luego, se realizan las operaciones de multiplicación y suma necesarias para calcular las salidas del modelo.

1. **OPCIÓN**

Opción para comunicarte con tu modelo desde SIMATIC Step 7 es utilizando una Raspberry Pi. Puedes utilizar la Raspberry Pi como un intermediario entre SIMATIC Step 7 y tu modelo en Python.

Aquí tienes un ejemplo de cómo podrías implementar esta opción:

**En la Raspberry Pi:**

Configura tu Raspberry Pi con el sistema operativo y las librerías necesarias, como TensorFlow u otras dependencias específicas de tu modelo.

Crea un script en Python en la Raspberry Pi que establezca un servidor de socket para recibir solicitudes desde SIMATIC Step 7 y enviar las respuestas correspondientes:

import socket

# Configurar el servidor de socket

HOST = '' # Dirección IP de la Raspberry Pi

PORT = 12345 # Puerto para la comunicación

# Función para procesar los datos recibidos y obtener la respuesta del modelo

def process\_data(data):

# Procesar los datos utilizando tu modelo

# (aquí debes agregar la lógica específica de tu modelo)

response = 'Respuesta del modelo'

return response

# Crear el socket del servidor

with socket.socket(socket.AF\_INET, socket.SOCK\_STREAM) as server\_socket:

# Vincular el socket a la dirección y puerto especificados

server\_socket.bind((HOST, PORT))

# Escuchar conexiones entrantes

server\_socket.listen()

print('Esperando conexiones...')

while True:

# Aceptar una nueva conexión

client\_socket, addr = server\_socket.accept()

print('Conexión establecida desde:', addr)

# Recibir los datos enviados por SIMATIC Step 7

data = client\_socket.recv(1024).decode('utf-8')

print('Datos recibidos:', data)

# Procesar los datos y obtener la respuesta del modelo

response = process\_data(data)

# Enviar la respuesta de vuelta a SIMATIC Step 7

client\_socket.sendall(response.encode('utf-8'))

# Cerrar la conexión con el cliente

client\_socket.close()

**En SIMATIC Step 7:**

* Configura tu dispositivo de comunicación en SIMATIC Step 7 para establecer una conexión con la dirección IP y puerto de la Raspberry Pi.
* Define las variables en tu programa Step 7 para almacenar los valores de entrada que se enviarán al modelo y las variables para almacenar los valores de salida que se recibirán desde el modelo.
* Configura tu implementación en Step 7 para enviar los valores de entrada a la dirección IP y puerto de la Raspberry Pi y recibir los valores de salida de vuelta.

Recuerda que este es solo un ejemplo de implementación y que los detalles exactos pueden variar según tus necesidades y configuración específica. Asegúrate de consultar la documentación de SIMATIC Step 7 y de la Raspberry Pi para comprender cómo configurar la comunicación y utilizar las funciones y métodos proporcionados por estas herramientas. Además, ten en cuenta aspectos de seguridad y control de acceso para proteger tus sistemas y datos industriales.

1. **OPCIÓN**

**OPC**