NeuralMAS

Neural Network – MAS Hybrid Framework

by Agnese Salutari

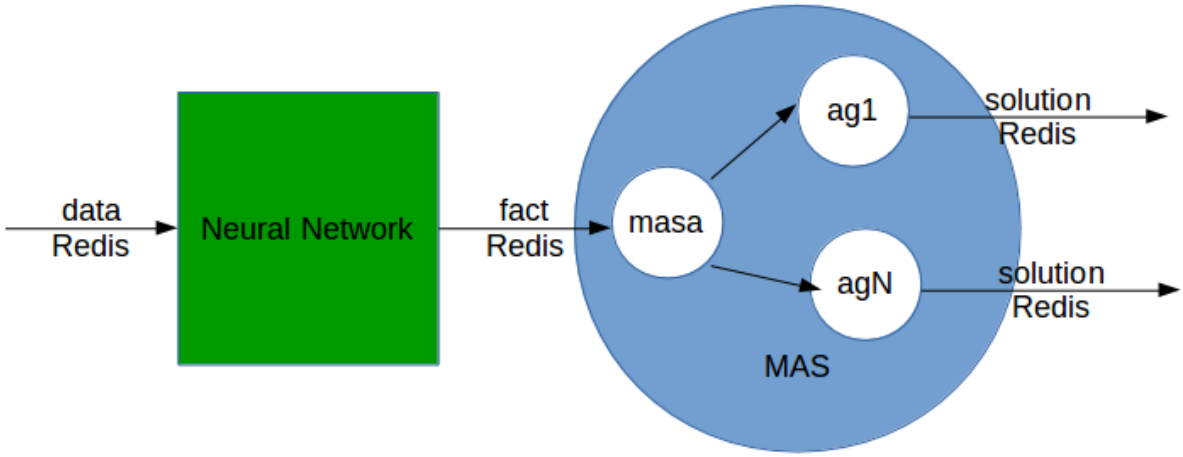
What it is

NeuralMAS is a Python3 framework that combine the power of Neural Networks with Multi-Agent Systems (MAS).

NeuralRedis is part of NeuralMAS and it is a special library that allows to create and model a daemon Python 3 program (similar to a MAS agent) that contains a Neural Network. That daemon program can make training on Datasets (the Dataset can be easily managed with NeuralMAS library) and can receive data, classify that data and send its response to the MAS (the communication too can be easily managed thanks to NeuralMAS library).

The MAS can finally elaborate the data coming from the Neural Net, according to its logic rules and Knowledge Base (KB) and sends its inferences as outputs.

A NeuralMAS project is easy to integrate with other modules, that can be programmed with different technologies too. The modularity comes from the communication channel that NeuralMAS uses: it uses Redis technology to store and deliver data and as an event handler too. In particular, the NerualMAS project receives data via Redis and sends its response to Redis, so that other modules can send queries to the NeuralMAS System and can get the NeuralMAS System solutions from Redis. Redis can be used (as well as a Dataset file) to store and/or send the Dataset to the Net too.



For more information about Redis, see: https://redis.io/

Thanks to its high modularity, a NeuralMAS project can be used on distributed Systems too. Even its modules can be distributed, we can put the Neural Net and the MAS in different machines, and even MAS agents can run on different machines.

NeuralMAS Projects are compatible with Docker (a very efficient virtual machine technology), so you can install them with all their dependencies on every machine that is compatible with it (on Servers and Robots too) very easily using Docker Containers.

For more information about Docker, see: https://www.docker.com/

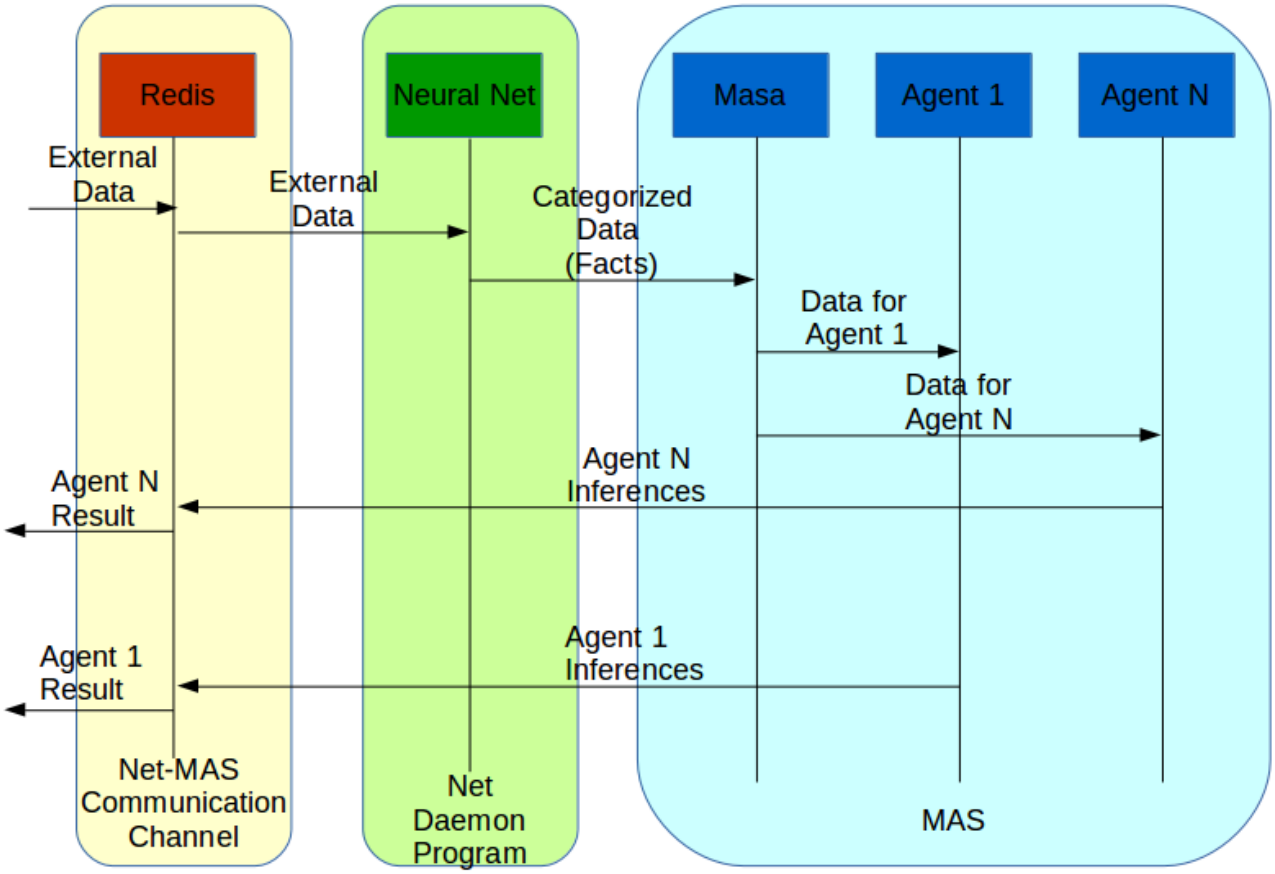
To have an example of Docker usage with real robots, see my work on a Kobuki robot:

https://github.com/agnsal/docker-IndigoROSdisPyPl

https://github.com/C-A-V-ROBOT/DR2

A NeuralMAS project can work on both synchronous and asynchronous events. It's useful to work on related events and big data management.

An event can be the arrive of some kind of data from sensors, for example.



Masa, MAS Administrator, is a particular agent that delivers data to the other agents of the MAS, depending on the agents skills (each agent works on a specific data category).

Technologies used in the example project

In the example I used a Keras Neural Net (running on top of Tensorflow), but in a NeuralMAS project, together with NeuralRedis, you can choose to use the Machine Learning technology you prefer instead of Keras Nets.

For more information about Keras, see: https://keras.io/

For more information about Tensorflow, see: https://www.tensorflow.org/

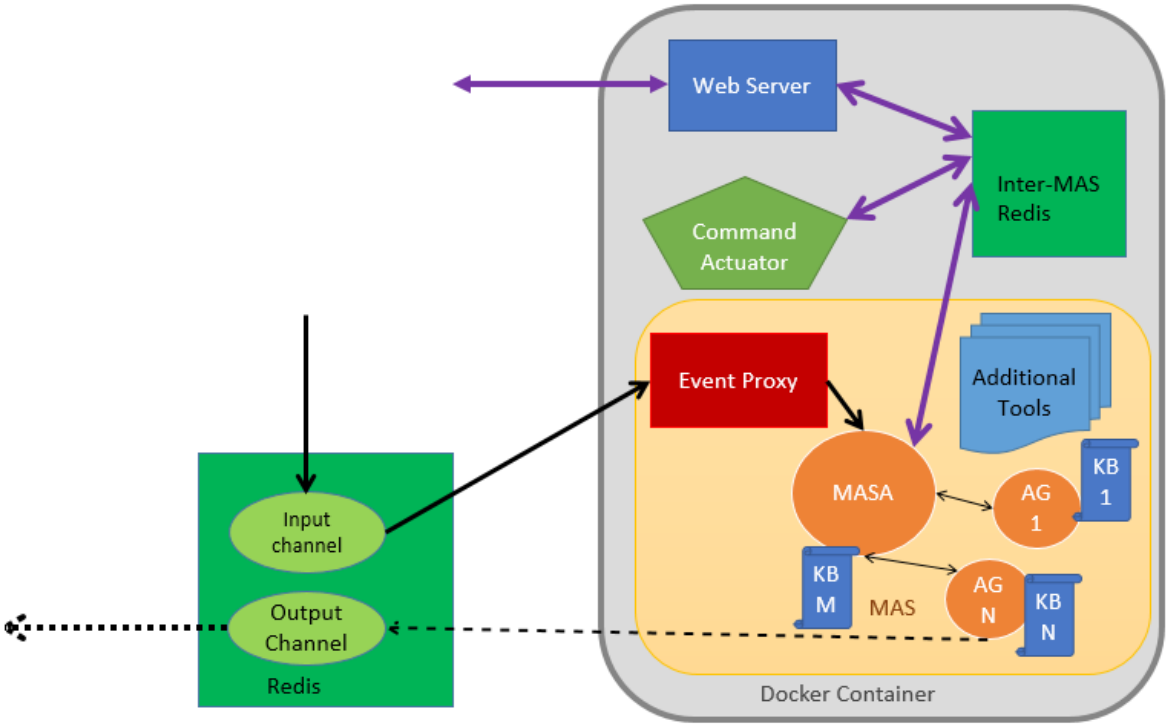
In the example I used a Koinè DALI MAS (running on top of SICStus Prolog), but you can choose the MAS or logic programming or programming technology you prefer instead of it to.

For more information about DALI, see: https://github.com/AAAI-DISIM-UnivAQ/DALI

For more information about Koinè DALI, see: https://github.com/agnsal/KOINE-DALI

http://ceur-ws.org/Vol-1949/CILCpaper05.pdf

For more information about SICStus Prolog, see: https://sicstus.sics.se/



In this image you can see how a Koinè MAS works at a lower point of view.

The violet rows are Koinè DALI modules I'm going to implement soon.

The other parts exist and I tested them in a Company, achieving the production phase with success.

The Event Proxy is a Python 3 program, called Redis2LINDA.py, for more information, see:

https://github.com/agnsal/Redis2LINDA-stringESE

https://github.com/agnsal/RedisClient

What happens in the example project

Suppose we are in a hospital, in the breast cancer ward.

Patients arrive randomly during the day.

We make them some analysis and put the obtained data in our NeuralMAS System using Redis as a communication channel.

The data, containing the patient identifier, arrive to the daemon program (BreastCanceNet.py) containing the Neural Network, that has been trained to distinguish between malignant and benign cancer. The daemon program, considering the Neural Net and its output correction function, has a reliability around 100%.

The result can be:

- 2 = benign
- 4 = malignant
- 0 = the Net can’t choose if 2 or 4

The daemon program sends its results, put into Prolog facts, to the MAS via Redis communication channel.

The Masa delivers the data to the competent agent, in the example we have BreastCancerAgent only.

BreastCancerAgent, who knows the patient clinic history, infers the risk band of every patient and sends its results (with every patient identifier) back via Redis communication channel.

The risk band will be:

- 1 – the highest:
  - If the net gave 4 as result
  - If the Net gave 2 as result but the patient had a cancer in the past
  - If the Net gave 0 as result and the patient had a cancer in the past
- 2 – the middle one:
  - If the Net gave 2 as result and the patient is a smoker
  - If the Net gave 0 as result and the patient is a smoker
- 3 – the lowest:
  - If the Net gave 2 as result and the patient has never had a cancer and the patient is not a smoker
  - If the Net gave 0 as result and the patient has never had a cancer and the patient is not a smoker

We can get the result in real time from Redis communication channel and we can decide what every patient need in a very short while.

There is protection against code injection.

NeuralMAS usage advantages

Why do we have to choose between Machine Learning and Logic Programming techniques when we can combine them, gaining the power of both and making each one rectify the other one weaknesses?

Machine Learning and Logic Programming are both very powerful:

- The first one allows to easily render every kind of information, so it allows to "see" clearly what's happening in the external world and read sensor data
- The second one allows to easily deduce new facts by using logic criteria from time to time, so it allows to "understand" what's happening in the external world and to take decisions about the action to perform next

NeuralMAS is a framework aiming to allow the cooperation of that 2 technologies, to create more efficient and complete systems.

In future I'd like to expand NeuralMAS in order to include more techniques and technologies. In particular I'm going to make it work with ROS to make it run on real robots and get sensor data in real time.

For more information about ROS, see: http://www.ros.org/

To see a ROS example, see my work on a real Kobuki robot:
https://github.com/agnsal/kobukiROSindigo

Contacts

Agnese Salutari

Email: agneses92@hotmail.it

Github: https://github.com/agnsal