EasyChair Preprint

№ 10908

Logical Infrastructure for Internet-Wide Artificial Intelligence Based on Artificial Neural Network and Multi-Agent Models

Olegs Verhodubs

September 15, 2023

Logical Infrastructure for Internet-wide Artificial Intelligence Based on Artificial Neural Network and Multi-Agent Models

Olegs Verhodubs
oleg.verhodub@inbox.lv

Abstract. The Internet is a huge infrastructure of electronic devices with the ability to communicate with each other. The possibility of communication between individual devices creates the preconditions for the transformation of the entire Internet into a single whole. The Internet as a single whole will increase the intellectual possibilities available to the average user. Despite significant progress in this direction, the potential is still not used to its full extent. With enormous computing power, the Internet-wide artificial neural network is what realizes its full potential. Previously designed multi-agent model is used to interconnect artificial neurons in Internet-wide artificial neuron network. This paper describes the architecture of Internet-wide artificial intelligence, built on the basis of an artificial neural network and a multi-agent system.

Keywords: Internet-wide Artificial Intelligence, Artificial Neural Network, Multi-Agent Systems

I. Introduction

Human-computer interaction can have different psychological connotations. Some people perceive the computer as a super-executive slave, ready to carry out any order of the owner. Such people revel in their power, realizing their secret and unfulfilled desire to command. Other people have found a true friend in the face of a personal computer that will never betray and will always be there, despite the vicissitudes of real life. Another category of people tends to identify the computer with God, or a super-skilled being with extraordinary abilities. Such people expect to receive help, advice or motivation from the computer, mistakenly endowing it with skills that it does not yet possess. The key word here is the word "yet", because the capabilities of the computer are increasing year by year. Of course, a computer will never replace God from real life; however, it (a computer) can become an electronic super-intelligent being as something common to all humankind, that is, a manifestation of all its knowledge, skills and mental states. One of the most important questions is whether a computer can become such a super intelligent being, given its technical capabilities and limitations. The situation is similar to the "Binding of Isaac" situation in reverse. According to the Bible, God tells Abraham to sacrifice his son, Isaac [1]. As Abraham begins to comply, he is stopped by the Angel of the Lord; a ram is sacrified instead of Isaac. Thus, God tested the strength of Abram's love for himself with this test. The reverse situation, or an alternative understanding of this situation, is described in Dan Simmons' novel "Hyperion". There, one of the main characters concluded that in fact it was not God, but Abraham who tested God, that is, whether this God is capable of being God. And since God nevertheless stopped the sacrifice of Isaac, showing his mercy, Abraham was convinced that such a God was worthy to be the God of Abraham and his people [2]. Similarly, we must decide whether it is possible to create a super intelligent being, based on a computer, accumulating the intelligence of all humankind. When we say whether, or not we can, we mean, first, the technical aspect, namely, whether there will be enough technical capabilities for the practical implementation of the plan.

There are only a few technologies, which can claim to be the foundation of a computer-based super intelligent electronic being. The first one is a technology of the Semantic Web [3], which

is the successor of hypertext markup language (HTML) and which allows you to markup information on the Web semantically. Semantically marked information on the Web becomes available for processing by machines, which turns the task of integrating information from various sources into a routine task that does not cause any difficulties. In addition, logical conclusions can be made, based on semantically marked information [4]. Another technology with a claim to create a global intelligent electronic structure is artificial neural networks. The areas of application of neural networks are very diverse - these are classification, forecasting, recognition, etc. The implementation of artificial neural networks requires large computing power, so it is logical to combine many computers to implement an artificial neural network using the communication infrastructure of the Internet. The implementation of artificial neural networks requires large computing power, so it is logical to combine many computers to implement an artificial neural network using the communication infrastructure of the Internet. The software layer of communication between computers, as parts of one artificial neural network, can be implemented using a ready-made library for creating multi-agent systems. Thus, this paper describes the architecture of a distributed system for creating an artificial neural network based on multi-agent technology.

This paper has several sections. The next section describes the architecture of the Internet-wide artificial intelligence based on artificial neural network and multi-agent system. After that, multi-agent subsytem of the designed sytem is discussed. The last section “Conclusion” is intended to sum up the results of the work.

II. Requirements and architecture

It was necessary to describe the work with the being designed system, as if it had already been implemented. This helps to develop requirements for the designed system, from which, in turn, it is possible to form the system architecture.

A: Expectations

The work with the system is as follows. The user launches an application in which he or she sets the parameters of an artificial neural network that will be distributed on many computers on the Internet. In general, there are three fundamentally different classes of artificial neuron network architectures. They are single-layer feedforward network, multilayer feedforward networks and recurrent networks [5]. The configurable artificial neural network, distributed on many computers, parameters are the following:

- number of used computers,
- type of artificial neural network,
- topology of artificial neural network:
    - number of artificial neurons in a layer (one computer is one layer),
    - connections between artificial neurons in artificial neuron network.

The setting of these parameters should be organized through visual capabilities through a graphical interface, since visual information is more demonstrative and better absorbed by a person.

After the parameters are set, the program creates a connection between computers that form a distributed artificial neural network, and only then, you can start working directly with this distributed artificial neural network.

B: Requirements

There are a lot of ready-to-use libraries and applications for creating an artificial neural network. The key feature of the developed system is its distribution on a variety of computers connected to the Internet. Thus, a technology capable of running applications independent of hardware and system software (operating system) is necessary. The choice is not great, because there are only two possible technologies, namely .NET and Java technologies. Without delving into a thorough comparison of these technologies, we will opt for Java technology, which has a longer history of its development, has a huge number of successful examples of implementing network applications (for example, servlets), and also has good documentation and a large number of free libraries [6].

Another requirement follows from the structure of an artificial neuron, which is a building block of an artificial neural network. It is known that the structure of an artificial neuron has the following form:

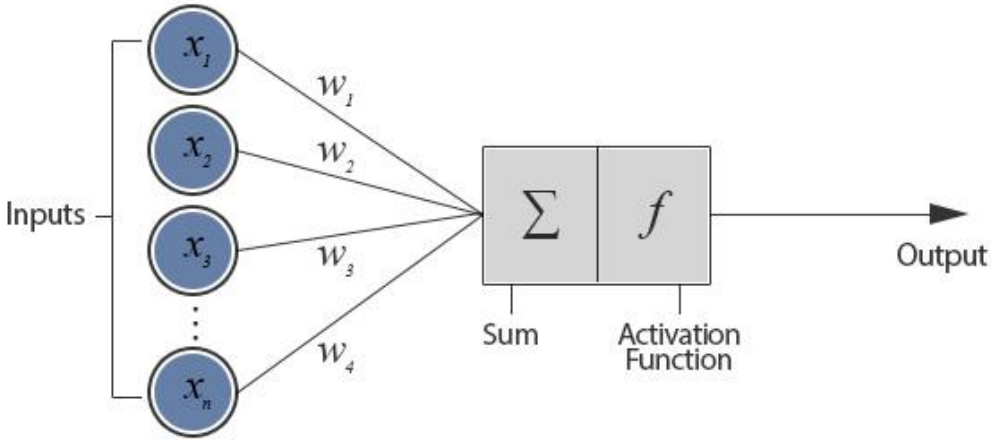

Fig.1. Model of artificial neuron [7].

Every artificial neuron has two components that need to be calculated. These components are summing function (sum in Fig.1.) and activation function. Nevertheless, arithmetic operations allowed in Java programming language and its Math class has has everything for programmer to calculate these functions: multiplication, division and addition operations, exponentiation, trigonometric functions and also many other possibilities [8].

Another requirement is related to the need to exchange values between artificial neurons, whether they are neurons of the same layer or neighboring layers. The exchange of values between neurons is necessary for learning and using an artificial neural network. There are two possible ways here. The first way is to use standard features of the Java programming language [9], [10]. The second way involves using one of the third-party libraries. The second way is preferred in our case, and a third-party library that provides communication between neurons in a distributed artificial neural network is the library for creating a multi-agent application. There are plenty of different third-party libraries for implementing multi-agent application, but this task

will be discussed further in the “Multi-agent subsystem” section, which is specifically dedicated to this purpose.

C: Architecture

Designed system is a software that simultaneously includes both client and server parts. The client part is necessary to create your own distributed artificial neuron network via the Internet, and the server part is necessary to provide your computer resources for distributed artificial neural networks of other users on the Internet (Fig.2). Here computer resources means memory and processor performance in terms of either percentage of load, or the number of artificial neurons in the layer (in the designed system, one computer is one layer of neurons in an artificial neural network), which you are ready to provide for the neural networks of other users.

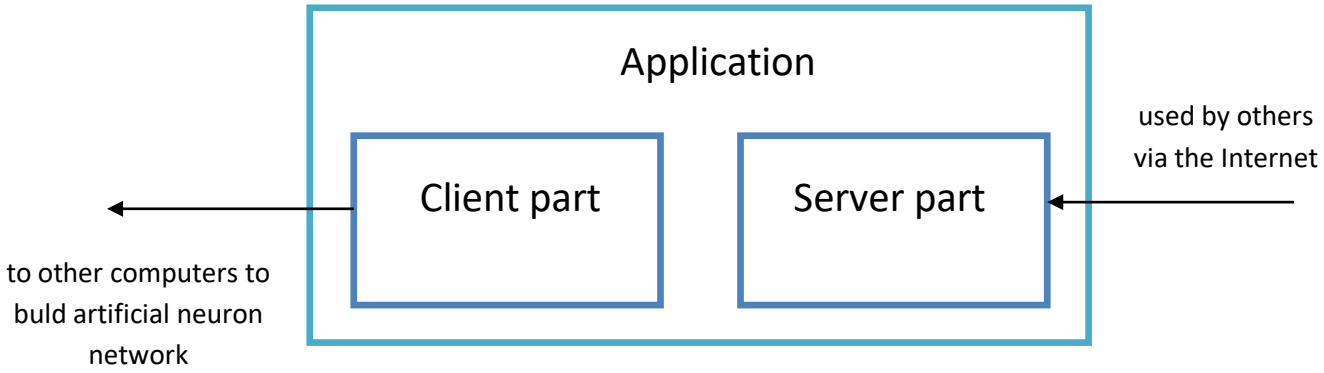

Fig. 2. General architecture of the application.

The client part (Fig.3) of the application consists of several blocks. The first block is an interface block, and this is for user interaction. This block allows you to customize your distributed artificial neural network by specifying its characteristics. The specified characteristics are the same as described in Section 2. The second block is a network-forming block, which is necessary to form a list of computers on the Internet that will participate in the implementation of a distributed artificial neural network. The third block is a neural network use block, which is necessary to exploit the created artificial neuron network via the Internet, including learning and use.

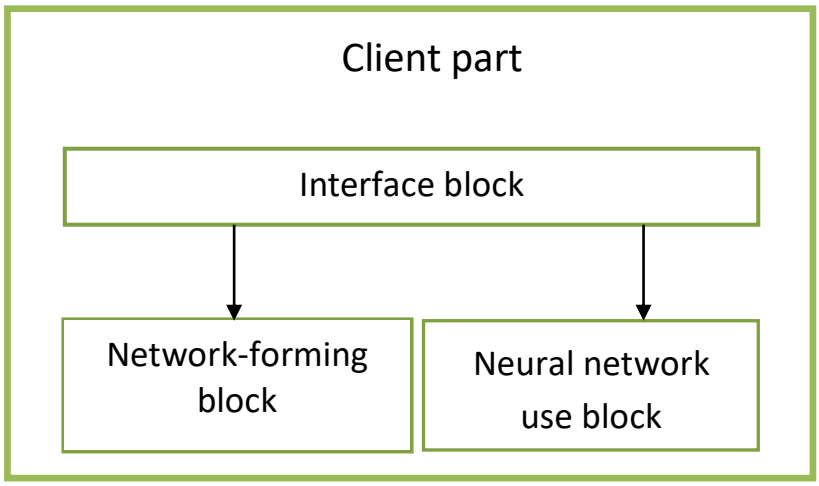

Fig. 3. Architecture of the client part of the application.

The server part (Fig.2) of the application consists of two blocks. The first block is a multi-agent subsystem, which is a container with agents, and the second block is a control block, which builds work with the multi-agent subsystem. Each agent in the multi-agent subsystem is an artificial neuron.

III. Multi-agent subsystem

Multi-agent subsystem is the key part in the designed system. Structurally, multi-agent subsystem is included in the server part of the designed application, and this multi-agent subsystem realizes one layer of the artificial neural network distributed on the Internet. This refers to the artificial neural network that other users create over the Internet and use your computer to create one layer of artificial neurons of the entire artificial neural network.

In general, the creation of multi-agent systems is subject to the section of science, which is included in the field of artificial intelligence, therefore we will consider some knowledge from this area.

A: Some theory

An agent is anything that can be viewed as perceiving its environment through sensors and acting upon that environment through actuators [11]. A distinction should be made between agents and intelligent agents. An intelligent agent is an autonomous entity, which act upon an environment using sensors and actuators for achieving goals [12]. A rational agent is one that acts so as to achieve the best outcome or, when there is uncertainty, the best expected outcome [11]. There are several classifications of intelligent agents based on their behavior and internal structure, but more important than this is a system of agents in which the boundaries between the differences between the characteristics of intelligent agents are erased.

Multiagent systems are systems composed of multiple interacting computing elements, known as agents [13]. The key moment of this definition of multiagent systems is “interacting..elements” regardless of the internal structure of intelligent agents. It is this property namely the property of the interaction of intelligent agents that should be used to implement an artificial neural network distributed on the Internet. Speaking about the interaction between agents, there is exactly communication or information interaction (that is, the exchange of information) between agents. Apparantly, the exchange of information can be implemented in several ways. The first way of the exchange of information is when one agent sends a message to another agent directly. The second way of the exchange of information is when one agent sends a message to another agent by means of one more intermediate agent. That is, if there are three agents A, B, C, and it is necessary to send some message from A agent to the agent C, then this message is sent from A agent to B agent and then this message is sent from the B agent to the agent C. The third way of the exchange of information implies the presence of not only agents, but also the environment. The environment can be defined as follows: environment is everything, except the observer. So, the third way of of the exchange of information is when one agent sends some message to the environment and another agent reads it. In this case, agents should have sensors to send messages to the environment and also to extract messages from the environment. Only the first and second methods have to be used in the designed system, because building interaction with the environment in this case is problematic.

B: Comparison

Requirements to the multi-agent subsystem follow from the requirements for the Internet-wide artificial intelligence, built on the basis of an artificial neural network and a multi-agent system as a whole. These requirements are:

- the possibility of using the Java programming language;
- the ability to place agents on different computers on the Internet;
- the ability to communicate agents (including those located on different computers) with each other;

Additional requirements can be added to these requirements, based on common sense:

- ongoing library support;
- good documentation;
- free and open-source licence.

A lot of multi-agent systems and libraries are developed, and our task is to choose the most appropriate of them. In turn, a large amount of something gives rise to the possibility of classification or classifications. It is possible to divide multi-agent systems or platforms into several groups [14]:

- general purpose platforms,
- special purpose platforms,
- platforms with unclear status or no longer under development.

Exactly general purpose platforms are interesting for the designed system, but they also can be divided into some groups [15], if any selection criterion is applied to them. Such a selection criterion can be one of the requirement, listed above.

So, the best choice is Jade platform [16], based on [14], [15] and considering all of the above requirements.

C: Use

One of the main tasks when working with artificial neural networks is their training. In the case of distributed neural network, where each container of agents is a layer of neurons, it is important to send messages with commands from one agent to another, regardless of which computer the destination neuron is located on. Jade enables sending messages from one agent, where each agent is an artificial neuron, to another, for example:

```
ACLMessage msg = new ACLMessage(ACLMessage.INFORM);
msg.addReceiver(new AID("Airport", AID.ISLOCALNAME));
msg.setLanguage("English");
msg.setOntology("Weather-ontology");
msg.setContent("Today it's raining");
send(msg);
```

This code creates a message to inform an agent whose nickname is Airport that today it is raining. Here the content of the message is "`Today it's raining`", but it can be anything. In our case, by means of a message, it is necessary to transmit service information, such as numerical values and something else. Technically, a lot of data can be encoded in a transmitted text

message. To do this, you need to know the amount of this data and how they are encoded. The task of extracting a substring from a given string is trivial in almost any programming language, including Java.

An agent can receive messages from the queue by means of the receive() method. This method returns the first message in the message queue (thus causing it to be removed), or null if the message queue is empty, and immediately returns [16]. For example:

```
ACLMessage msg = receive();
if (msg != null) {
    // . . . . . .
}
```

It is possible to use this code to process received messages.

In theory, agents are something more than entities which exchange information only. In theory, agents have their own behaviour, but behaviour of agents is redundant capability here that is why it is not use.

IV. Conclusion and future work

This paper has demonstrated the possibility of creating an Internet-wide artificial intelligence using artificial neural network and multi-agent system. The architecture in general terms and requirements of being designed system are described in this paper, too. Different multi-agent platforms have been reviewed and evaluated for their applicability to the designed system in the paper, and exactly Jade platform is chosen the most suitable.

The main limitation for the day-to-day use of the system in the future is the critical dependence on the reliability of the communication means. A broken connection can interrupt the work of an artificial neural network distributed over various computers. And even if the communication channel does not cease to perform its function, the speed of information exchange between nodes (computers) can be significantly reduced, which will be a serious obstacle to efficient operation. A possible solution in this case may be to maintain statistics on the reliability of nodes (computers), their operating time and their throughput. This statistics will be useful at the stage of building an artificial neural network distributed on the Internet, when the most appropriate node will be selected for each layer of neurons. Apparently, to maintain such statistics, one cannot do it for a separate server. Another solution here is possible, but either more difficult to implement or not as effective.

The implementation of the system described above will give every user with access to the Internet a tool comparable in power to a supercomputer and even more powerful. Of course, it is not necessary to use all computers on the Internet to create your own artificial neural network, but it is possible to use a lot of computers what increases the available power, than when only one computer is used. In turn, the transfer of the ability to experiment with artificial neural networks to a wide, public access holds many discoveries in the future.

Acknowledgments

This paper as previous papers is prepared only by the efforts of the author.

References

[1] Bible

[2] Dan Simmons, Hyperion,, 1989.

[3] John Davies, Rudi Studer, Paul Warren, Semantic Web Technologies, 2006.

[4] Dieter Fensel and others, Spinning the Semantic Web, 2005.

[5] Haykin, Neural Networks, 3d edition, 2009

[6] Bruce Eckel, Thinking in Java 4th edition, 2006.

[7] https://www.theprojectspot.com/tutorial-post/introduction-to-artificial-neural-networks-part-1/7 . Accessed: 29.05.2023

[8] https://docs.oracle.com/javase/8/docs/api/java/lang/Math.html Accessed: 30.05.2023

[9] https://www.geeksforgeeks.org/java-networking/ Accessed: 30.05.2023

[10] https://www.tutorialspoint.com/java/java_networking.htm Accessed: 30.05.2023

[11] Russel, Norvig, Artificial Intelligent. Modern Approach

[12] https://www.javatpoint.com/agents-in-ai Accessed: 08.06.2023

[13] M. Wooldridge, An introduction to multi-agent systems, 2001.

[14] Constantin-Valentin Pal, Florin Leon, Marcin Paprzycki, Maria Ganzha, A Review of Platforms for the Development of Agent Systems, 2020.

[15] Constantin-Valentin Pal, Florin Leon, Marcin Paprzycki, Maria Ganzha, A Review of Agent Platforms, 2018.

[16] Fabio Bellifemine, Giovanni Caire, Dominic Greenwood, Developing multi-agent systems with Jade, 2004.