Conference on
Information Technology and
Data Science

Neuron Network Model in the Study of Smart City Ideas

Mátyás Varga[a], Bence Soltész[a], Norbert Béla Fiedler[a], Anikó Apró[a], Balázs Borsos[a], Gábor Kiss[b], Zoltán A. Godó[a]

[a]Department of Information Technology, Faculty of Informatics,
University of Debrecen, Hungary
Corresponding author e-mail: godo.zoltan@inf.unideb.hu

[b]Institute of Machine Design and Safety Engineering,
University of Óbuda, Budapest, Hungary

Proceedings of the 1st Conference on Information Technology and Data Science
Debrecen, Hungary, November 6–8, 2020
published at http://ceur-ws.org

Abstract

With neural networks, computer science has made tremendous progress in the field of artificial intelligence. The breakthrough is provided by the closest possible analogy with the living nervous system, as it provides the most efficient processing of real-world data. In our work, we are looking for an even closer analogy with the living nervous system by building a neural network. We build the network hardware and emulate the nodes with multiprocessors. We implement actual concurrency, real-time task runs. We enable communication with both analog and digital features, thus mimicking the operation of natural systems. A unique interpreter at the nodes of the neural network provides controllable stream signal processing. Thus, the system is able to receive and process any data stream. By implementing cascade programming, the interpreter can also be developed. That is, the entire control language can be replaced according to the desired task. The system is not only suitable for Smart City or traffic modeling, but also for direct neuroinformatics or didactic research.

Keywords: Smart City, neuron-network, massively parallel system

1. Problem Statement

In today's most popular personal computers, which are based on the von Neumann architecture, true parallelism is not yet implemented [19]. Despite modern processors having multiple cores, the concept of per-core task execution points beyond the complexity of current personal computer architectures. The basic advantage of multicore processors is that they are capable of running multiple threads in parallel at the same time. With multiple cores built into a single processor, the overall performance is multiplied, thanks to calculations running parallel to each other. However, processors that are capable of processing parallel instructions are called superscalar processors.

Typically, the real world can be characterized by data traffic that consists of unprocessably many, parallel signals. This is further complicated by the fact that everything around us is analog. However, traditional computer processing works with quantized numbers, which are digitized with a high degree of loss, and move in a discrete range of values.

The world around us, physical and chemical phenomena, or the usage of a man-made environment such as a Smart City, can be most effectively interpreted by the central nervous system of living beings [8]. While computers are more suitable for data processing, the data will be used by an individual with a nervous system. Vital processes, orientation, usage of the environment, social interactions etc. are all handled by the brain as a supercomputer in the most effective way.

However, the nervous system is typically an analog, massively parallel neuron system [7]. To be exact, the living nervous system is not the one that is similar to the artificial model, but it is the artificial model that was used to try and copy the natural structure of the brain. It is quite clear, that if the central nervous system is such an efficient user system, similar artificial structures have to be built if we want to approximate the efficiency of the natural system. While neural networks are software emulated parallel systems on a von Neumann architecture machine, the goal of neuron networks is to achieve true, full, hardware architecture based parallel task execution. Neural networks already provide some amazing results, for example with learning algorithms or different levels of artificial intelligence. Therefore, we can expect even more exciting results from neuron networks, which are much closer to the structure of the natural nervous system.

2. Levels of Parallelism

Multiple levels of parallelism can be implemented. On the highest level, we have parallelism between processes. Then comes parallelism between jobs, parallelism between procedures (macro instructions), parallelism between instructions, parallelism on the level of processed data, parallelism within the execution of an instruction, and on the lowest level, parallelism within a hardware unit. Our goal was to implement parallelism on the lowest level, which has the closest analogy to the 'evolution-developed' central nervous system.

Several scientific and engineering problems can be better approached in a parallel neuron network model. With the system that we developed, we would like to model [14] Smart City dataflow, the propagation of information and the city traffic as well [4]. Due to its universal structure, any other field can be modelled with it, with the change of the processor's program. For example, an obvious use case would be neuro-informational application, due to the similarities with the nervous system [13]. In this case we would connect the system with a living nervous system and implement two-way communication. In the past, we have already achieved connection with a living nervous system, using only 9 processors and 256 microelectrodes.

In our current research, 216 high performance processors are available, which makes it possible to build a 6 x 6 x 6 sized, massively parallel neuron system cube.

Taking use of modern technological opportunities, nodes are represented by complete microcontrollers. This way, each node in the neuron network contains a combined processor, memory, and I/O unit. This means that it does not simply work as a Kirchhoff node, like in the CNN model, but as a complete, program-controlled, data and program processing unit.

Gustav Kirchhoff's Current Law is one of the fundamental laws used for circuit analysis. This law states that for a parallel path the total current entering a circuits junction is exactly equal to the total current leaving the same junction.

The name of CNN stands for Convolutional Neural Network. The model is the most popular among the other deep neural networks (for example: Deep Neural Network(DNN) or Artificial Neural Network(ANN)). The model takes its name from a mathematical linear operation between matrixes called convolution. This type of model has multiple layers just like our microcontroller system. The model itself has a great performance in machine learning problems. We will use this performance to analyze a city's traffic and more [1].

Shared nothing systems are concerned with access to disks, not access to memory. Nonetheless, adding more CPUs and disks can improve scaleup.

3. Massively Parallel Systems

Massively Parallel (MP) systems have the following characteristics:

We do not need thousands of nodes to build a working parallel system from only few nodes we can set up a whole system which contain the characteristic of these kind of systems. Even if we can afford an MP system we do not need to spend a lot of money because the cost of the nodes can be extremely low.

Each node has a non-shared memory so the memory capability of the system is grow with the amount of nodes equally. Every microcontroller is a unique device so they do different tasks, but even if a controller is wrong the other nodes can access to the failed device and take it's task or compute with the controller's data.

There is multiple organization type of the nodes. We can build a grid topology where the controllers close a square shape on the same level (for example a table) or a mesh organization type where nodes will form a hexagon shape. If we do not want the processors to be on the same level we can build a cube form where every node will form a three dimensional cube formation which is called hypercube arrangement.

The software can potentially reside on all nodes but in a tokenized form so every controller has a unique ID.

A massively parallel system may have as many as several thousand nodes [16]. Each node may have its own software instance, with all the standard facilities of an instance.

An MP has access to a huge amount of real memory for all database operations (such as sorts or the buffer cache), since each node has its own associated memory. To avoid disk I/O, this advantage will be significant in long running queries and sorts.

Examples of massively parallel systems are the inCUBE2 Scalar Supercomputer, the Unisys OPUS, Amdahl, Meiko [9], and the IBM SP.

The Unisys OPUS is a parallel computer system. This system was a low-cost and high-speed machine that was gaining acceptance fastly in the corporate. This computer was built to make processes and computations fast instead of scientist [12]. We would like to earn the same result with our neural network.

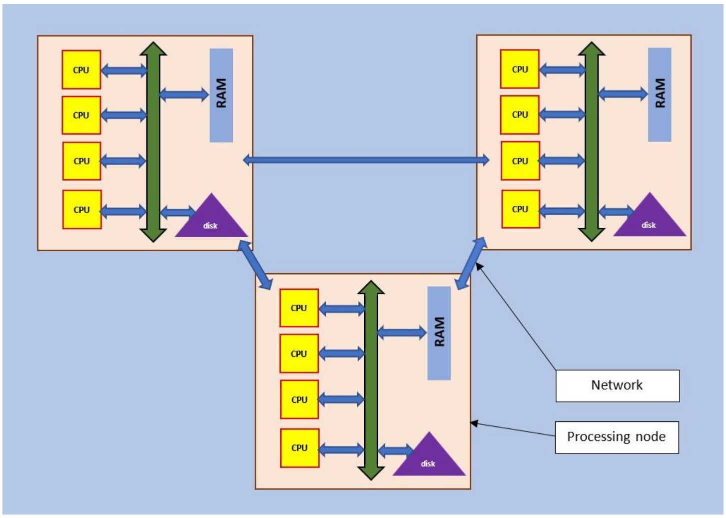

Figure 1. Massively parallel processing.

The Amdahl is a law that based on parallelism. The theory of doing computational work in parallel has some fundamental laws that place limits on the benefits one can derive from parallelizing a computation [18]. This law helps us to speed up the process time of different computations which would be great for our system.

The IBM SP (SP stands for Scalable POWERparallel) is a series of supercomputers from IBM. The SP was introduced back in 1993. This computer is a distributed memory system, consisting of multiple RS/6000-based nodes interconnected by an IBM-proprietary switch called the High Performance Switch (it is also called HPS).

Figure 1 of ours is representing a shared memory and CPU system where one Processing node can give its computations through a network to another node. Because of this 'sharing system', one node can only work with one process and another controller with another computation. As a result the problem solving time will be reduced dramatically because if we give one complex problem to one node, then it has to be solve every part of the problem. So this is the layer structure we talked about before in the CNN model. We 'break' the problem into pieces and then because all node is connected (just like in the picture) to each other, one node has only one job to be done.

4. Analog Implementation

Our system under construction is capable of performing tasks of any complexity due to intelligent nodes [17]. The program uploaded to the nodes is suitable for controlling the data flow, which is realized by the connections of the I/O PINs of the node microcontrollers. Because the nodes are separate processing units, it is possible to implement analog data traffic in addition to, or instead of digital.

Analog stream processing adds new possibilities to the system. Because signals from the natural environment are typically analog, the architecture of our neural network system is also closer to the nature of the data to be processed (see Figure 2).

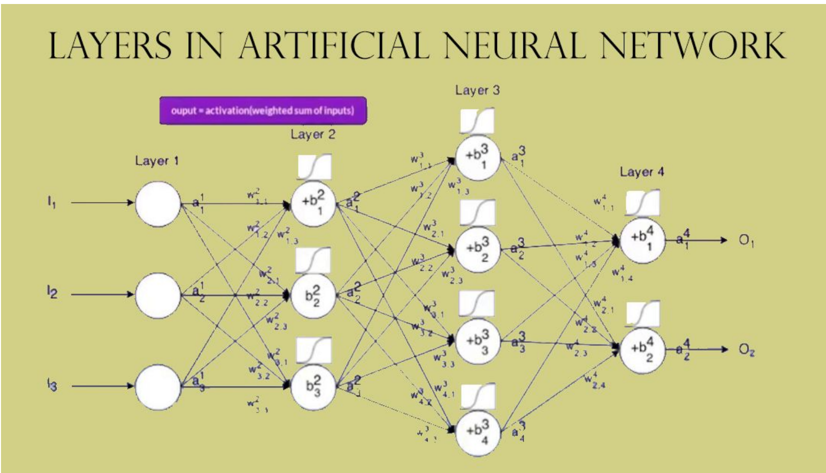

Figure 2

If we analyze Smart City problems, a number of analog signals need to be processed [20]. Analog communication is typically represented by voltage levels in similar systems. A refined model of this is when a so-called fill factor (PWD) provides the voltage signal level. This way, the analog signal can be produced with the frequency of the digital signal. This solution is also interesting because in the living nervous system, the flow of information between living neurons takes place on a similar principle. The signal is binary, where 1 is the value of the action potential and 0 is the value of the resting potential. In addition, due to the 'all or nothing' law, neurons cannot take on an intermediate value. So they are discrete or stable at 1 or 0. If that were all the 'science of neurons' we could already connect to our digital computers. However, the information in the nervous system is carried by frequency, i.e., the time course of states 0 and 1.

The Artificial Neural Network (ANN) is a computing system inspired by biological neural network [2] that constitute animal brain but it can be easily implemented in other areas such as Smart City. The system contains one input layer (picture layer 1) where the data which has to be processed flows in then there are multiple hidden layers (picture layer 2 and 3) where the actual computations will be done. The last layer is the output layer (picture layer 4) where the processed result is given by the computation. In our model it is representing the position and tasks of the micro controllers. The input layer will be our main controller (see Figure 3) and the hidden layers will be the other nodes. And in our implementation the result (the output layer) will be a picture or some information on our computer.

5. Software Levels

Node microcontrollers require multi-level program execution [6]. The most basic program is the bootloader, which is responsible for uploading and executing the running main program. We can upload our own main program to the nodes running below this. However, the running main program is an interpreter that will process the data stream. The stream contains special control information that gives instructions on how to process the stream. This allows for extremely complex processing and wide applicability [10].

As a first step, the bootloader must be rewritten before the 216 fixed circuit processors are installed. This is because replacing the main program cannot be solved by programming the processors one by one. Instead, we need to use something called cascade programming (See Figure 3). The point is that the first processor receives the new master program, and then the modified bootloader passes the master program to the next program via an I/O PIN. This way, programming takes place automatically in a cascaded system. Each node has its own ID. This allows the main program to adapt to the physical location of the node. As a result, the interpreter program processing the data stream can be developed at any time after the neuron network has been assembled. The scripting language that controls the interpreter can also be developed and expanded.

The main program then becomes capable of receiving and sending data streams to neighboring nodes via dedicated I/O PINs [3]. The instructions placed in the data stream are interpreted and executed by the main program as an interpreter. Follow the instructions to switch to analog or digital data transmission. Traffic can be controlled by telling which node or nodes we want to forward the data stream to [15]. It can be transformed, processed and also multiplied or reduced. All this can be controlled by instructions placed in the datastream. So the universality of the system is very extensive.

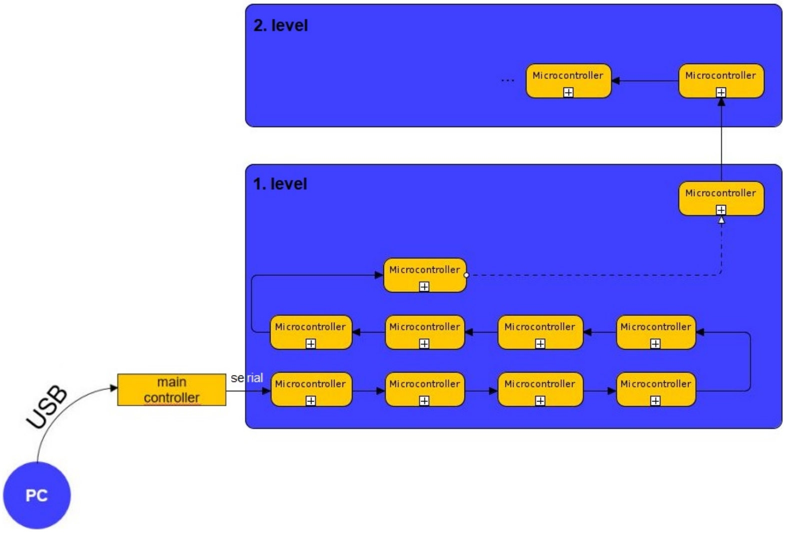

Figure 3. Cascade programming visualized.

Figure 3 is about the method we programming all microprocessors. The main thing about this that we do not program up every controller one by one. We can see that the program starts at the computer where we wrote a code in a TOKEN form which means there are parameters in the code that will be overwritten if a condition becomes true. We upload our program through and USB serial cable to the main controller (this will be the only controller that we will program up manually) and then the given code will travel through all the other processors and by the end of the process all controllers have a different ID. With the identifier then we can easily give command to a processor which computation shall it done when the data 'flows in'. We must make sure that every controller is successfully connected with the other otherwise the system will not work because of processing error or the time for the problem will radically grows.

6. Summary

A stream-driven, massively parallel neuron network with such a structure, with both analog and digital features, with such a large number of intelligent processors, running an intelligent interpreter [11]. It is completed with serious and very thoughtful planning. Therefore, its presentation and expected results may be of great interest to the scientific world. It is universal analog-to-digital architecture and data stream interpreter control open up exciting modeling opportunities in Smart City modeling [5].

References

[1] S. Albawi, T. A. Mohammed, S. Al-Azawi: *Understanding of a convolutional neural network*, in: 2017 International Conference on Engineering and Technology (ICET), Antalya, Turkey: IEEE, 2017,
DOI: https://doi.org/10.1109/ICEngTechnol.2017.8308186.

[2] M. Anthony, P. L. Bartlett: *Neural Network Learning*, in: Australian National University, Canberra: Theoretical Foundations, 2009.

[3] S. V. Bykovsky, Y. G. Gorbachev, A. E. Platunov, A. O. Kluchev, A. V. Penskoi: *Hardware/software Co-design*, in: St. Petersburg, Russia: University of Accurate Mechanics and Optics, 2016, part 1,
DOI: https://doi.org/10.1007/978-94-009-0187-2.

[4] N. Chen, Y. Chen: *Smart city surveillance at the network edge in the era of IoT: opportunities and challenges*, in: Smart Cities, Netherlands: Springer, Berlin, 2018, pp. 153–176,
DOI: https://doi.org/10.1007/978-3-319-76669-0_7.

[5] S. Furber, S. Temple, A. Brown: *On-chip and inter-chip networks for modelling large-scale neural systems*, in: Procedural International Symposium on Circuits and Systems, Kos, Greece: ISCAS-2006, 2006,
DOI: https://doi.org/10.1109/ISCAS.2006.1692992.

[6] A. Gaur, B. Scotney, G. Parr, S. McClean: *Smart city architecture and its applications based on IoT*, in: Procedia Computer Science. 52, 2015, pp. 1089–1094,
DOI: https://doi.org/10.1016/j.procs.2015.05.122.

[7] K. Gautam, V. Puri, J. G. Tromp, N. G. Nguyen, C. V. Le: *Internet of Things (IoT) and Deep Neural Network-Based Intelligent and Conceptual Model for Smart City*, in: Singapore: Springer, 2019,
DOI: https://doi.org/10.1007/978-981-32-9186-7_30.

[8] F. Gil-Castineira, E. Costa-Montenegro, F. Gonzalez-Castano, C. López-Bravo, T. Ojala, R. Bose: *Experiences inside the ubiquitous oulu smart city*, in: Computer, vol. 44, 6, IEEE, pp. 48–55,
DOI: https://doi.org/10.1109/MC.2011.132.

[9] A. Holman: *The meiko computing surface: A parallel & scalable open systems platform for Oracle*, in: Berlin, Heidelberg: Springer, 2005,
DOI: https://doi.org/10.1007/3-540-55693-1_34.

[10] J. Jin, J. Gubbi, S. Marusic, M. Palaniswami: *An information framework for creating a smart city through internet of things.* In: IEEE Internet Things Journal 1(2), 2014, pp. 112–121,
DOI: https://doi.org/10.1109/JIOT.2013.2296516.

[11] J.-K. Kim, J.-H. Choi, S.-W. Shin, C.-K. Kim, H.-Y. Kim, W.-S. Kim, C. Kim, S.-I. Cho: *A 3.6 Gb/s/pin simultaneous bidirectional (SBD) I/O interface for high-speed DRAM*, in: San Francisco, CA, USA: IEEE, 2004,
DOI: https://doi.org/10.1109/ISSCC.2004.1332770.

[12] D. B. Kirk, W.-m. W. Hwu: *Programming Massively Parallel Processors: A Hands-on Approach*, in: Elsevier Inc., 2017, p. 576,
DOI: https://doi.org/10.1016/C2015-0-02431-5.

[13] A. Kolesenkov, B. Kostrov, E. Ruchkina, V. Ruchkin, in: Anthropogenic situation express monitoring on the base of the fuzzy neural networks, Budva, Montenegro: IEEE,
DOI: https://doi.org/10.1109/MECO.2014.6862684.

[14] S. Latre, P. Leroux, T. Coenen, B. Braem, P. Ballon, P. Demeester: *City of things: An integrated and multi-technology testbed for iot smart city experiments*, in: Smart Cities Conference (ISC2) 2016, IEEE International, 2016, pp. 1–8, doi: https://doi.org/10.1109/ISC2.2016.7580875.

[15] S. Paul, V. Honkote, R. G. Kim, T. Majumder, P. A. Aseron, V. Grossnickle, R. Sankman, D. Mallik, T. Wang, S. Vangal, J. W. Tschanz, V. De: *A Sub-cm3 Energy-Harvesting Stacked Wireless Sensor Node Featuring a Near-Threshold Voltage IA-32 Microcontroller in 14-nm Tri-Gate CMOS for Always-ON Always-Sensing Applications*, in: vol. 52, IEEE, 2017, pp. 961–971, doi: https://doi.org/10.1109/JSSC.2016.2638465.

[16] J. L. Potter, D. B. Gannon: *The Massively Parallel Processor*, in: Harward St.Cambridge MA United States: The MIT Press, 1985, doi: https://doi.org/10.7551/mitpress/4468.001.0001.

[17] L. Sanchez, L. Muñoz, J. A. Galache, P. Sotres, J. R. Santana, V. Gutierrez, R. Ramdhany, A. Gluhak, S. Krco, E. T. et al.: *Smart-santander: Iot experimentation over a smart city testbed*, in: Computer Networks, vol. 61, pp. 217–238, doi: https://doi.org/10.1016/j.bjp.2013.12.020.

[18] S. Tsutsui, P. Collet: *Massively Parallel Evolutionary Computation on GPGPUs*, in: Berlin, Heidelberg: Springer, 2013, doi: https://doi.org/10.1007/978-3-642-37959-8.

[19] A. Zanella, N. Bui, A. Castellani, L. Vangelista, M. Zorzi: *Internet of things for smart cities*, in: IEEE Internet Things J. Vol. 1(1), pp. 22–32, doi: https://doi.org/10.1109/JIOT.2014.2306328.

[20] Y. Zou, B. Jolly, R. Li, M. Wang, R. Kaur: *The internet of things: nervous system of the smart city*, in: Smart Cities, Berlin: Springer, pp. 75–96, doi: https://doi.org/10.1007/978-3-319-59381-4_5.