Krzysztof TURCHAN[1], *Emil MICHTA*[2], *Krzysztof PIOTROWSKI*[1]
[1]*IHP – Leibniz-Institut für innovative Mikroelektronik, Frankfurt (Oder), Germany*
[2]*Uniwersytet Zielonogórski, Instytut Metrologii Elektroniki i Informatyki*

LOW-POWER ARTIFICIAL INTELLIGENCE BASED SENSORS FOR WIRELESS SENSOR NETWORK APPLICATIONS

This work describes the progress towards creating a framework for applying AI techniques on low-power sensor nodes. The aim of the approach is to make use of the AI operations on the nodes to allow moving the data processing as close as possible to the place the data is collected. This requires the processing to be low power but reliable, since the raw data will not be transferred from the source node.

ENERGOOSZCZEDNE SENSORY BAZUJĄCE NA SZTUCZNEJ INTELIGENCJI DO ZASTOSOWAŃ W SIECIACH SENSORÓW BEZPRZEWODOWYCH

Artykuł przedstawia postępy prac w kierunku stworzenia zbioru zasad i narzędzi ułatwiających użycie technik SI na węzłach sieci bezprzewodowych sensorów. Proponowane podejście ma na celu realizacje operacji SI na węzłach, i poprzez to, przeniesienie przetwarzania danych jak najbliżej miejsca, w którym są one zbierane. Przetwarzanie musi być w takich przypadkach energooszczędne ale również niezawodne, ponieważ surowe dane nie będą przesyłane poza źródłowy węzeł sieci sensorów.

1. MOTIVATION

AI-based solutions are used nowadays in many application scenarios. They approach the wireless sensor network area. Intelligent solutions like machine learning and object recognition raise sensors to a higher level. Such sensors can identify specific events in the data streams and by that allow observations, but also emulate behavior of the human brain. However, in many cases these solutions require either processing power and/or permanent Internet access. Both requirements cause a large power consumption. Therefore, new approaches to solve this issue are needed. The scope of this paper is to present the concept of a framework that defines the aspects of the problem and that suggests how to minimize power consumption through data preprocessing as close to the source as possible. In general, the processing should happen before transferring any data, but it needs a careful planning considering the energy budget of the specific sensor node, while taking advantage of the non-linear and abstract data processing offered by the AI techniques.

2. ARTIFICIAL INTELLIGENCE

Artificial Intelligence (AI) is a field of computer science that deals with creating models of intelligent behaviors and with computer applications simulating these behaviors [1]. These approaches do not follow the usual digital and organized programming approach, but add some abstract and non-linear aspects typical for the human brain. For that reason, a neural network is the most prominent example of an AI structure, but there are also many other structures and techniques used in this area.

Neural networks are pieces of hardware and/or software that model the operation of neurons in the human brain. The simplest neural network is a perceptron. It consists of exactly one artificial neuron. The perceptron has several inputs with assigned weights that determine how much the value of a given input affects the overall state of the neuron. The sum of the weighted data from all inputs, together with the optional bias value added to it, drives the activation function that generates the output value. In the simplest example – the perceptron – this function returns only two values, i.e., if the sum is greater than zero, the perceptron will generate a positive answer (Boolean true or “1”) and a negative answer (Boolean false or “0”), otherwise. By that, the perceptron allows to split the combinations of input values into two classes, i.e., the *positive* and the *negative* ones. This classification feature allows solving some tasks that are hard to program in the typical way. Further, it allows adapting to the changing context by updating the weights and the bias. This update can be done either by online training, or by replacing the values with ones from another trained neuron (identic structure).

To implement more complex classifiers other types of activation functions are used and their output is usually a real number. Further, a single artificial neuron is not enough for complex tasks. For this reason, the neurons are combined into networks with multiple layers. Each neural network consists of an input layer, hidden layers and an output layer. There may be any number of hidden layers, but the more hidden layers the more computing power is required. The first layer (the input layer) receives raw data from all kinds of sensors. Every following layer (the hidden layer) receives data resulting from data processing in the previous layer. The last layer produces the output data. The simplest neural network structures pass the data between their layers in one direction only (feedforward neural network), which means that each neuron processes a single set of input data only once. Such a structure can be easily represented in a software or hardware implementation as a series of vector operations. By that, a relatively simple sequential program can perform classification or pattern recognition tasks that could be hardly implemented using classical programming methods.

On the other hand there are complex networks consisting of a large number of layers that are associated with the Deep Learning (DL) concept. Standard neural networks in most cases require human support in the learning phase. The difference between neural networks and DL is that in the latter the computer learns how to perform tasks that are natural to the human brain. The computer collects basic data parameters and prepares these for self-learning by recognizing patterns using multiple processing layers. Which, in turn, causes a significant computational burden and as a result, significant energy consumption.

3. EMBEDDED ARTIFICIAL INTELLIGENCE

Examples from section 2 provide different possibilities with respect to the tasks they can realize, but also have different energy and computational power requirements. On embedded devices, due to their rather limited computational power, too complex calculations cannot be performed, but there are options to use AI on these devices. For embedded PC devices, like the Raspberry Pi, there are AI software libraries, like Caffe or TensorFlow. These libraries allow implementing AI solutions that can run on such embedded hardware. To improve the energy-to-performance ratio, there are AI hardware accelerators that allow reducing the performance limitations of pure software [2], [3], [4].

The learning phase needs high computing power, so for embedded devices it is usually transferred to devices that are more powerful. Then the trained neural networks are copied to the embedded devices and are executed in the application phase or the inference phase. This approach can be used for tasks, where the model is static, i.e., the patterns to be recognized do not change, or do not change frequently and the costs for the model update are acceptable.

3.2 Use cases

Embedded Artificial Intelligence can be used in every field of life. First example of its use is to monitor the operation of an electric motor. Such a system can detect possible engine malfunction. The system compares data from some sensors with the trained model and returns output data like information if the parts of motor are still operational.

Another example is an automatic temperature setting. In such a system, the input data are the current air temperature and the current season. On this basis, system can decide how high temperature is necessary to be set.

Yet another example is an autonomous car. Tesla Inc. has presented its autonomous car based on the AI. The operation of Artificial Intelligence must take place directly in the car, because it is possible for the car to lose the Internet connection. Tesla uses their own AI chip that performs up to 72 trillion operations per second. The system, as a whole, is capable of analyzing 2,100 frames of video each second, which is 21 times faster than previous generation of hardware [5].

These examples show that it is possible to operate embedded AI without a continuous connection to the Internet. However, most of these examples do not have limited resources, like computing power, but mainly energy. The proposed approach aims at investigating the applicability of AI in devices that are limited in both, computational power and available energy – the sensor nodes.

4. INTELLIGENT SENSORS

The proposed solution is a work in progress in its initial conceptual phase and with several steps defined. Its aim is to enhance the sensor nodes with features provided by the AI techniques. These features include a broad set of useful tools that are not easily implementable traditionally. Examples here are measurement classification and compression, detecting anomalies and recognizing defined relations between different measured parameters, or the same parameter in the time domain. The first task is to identify the most relevant applications of AI techniques in the area of wireless sensor networks and to investigate the most suitable structures to implement these applications.

Further, an analysis of the costs related to the learning phase and the application phase for the different implementation options (hardware and software) for each structure will be performed. Based on that results a framework that supports the implementation will be proposed. This framework will help to define and dimension the AI structure to be applied in the application, as well as to estimate the related energy costs. The latter will be done considering the user preference regarding the device, on which the different phases on the AI structure are executed (learning and application phase). Suggestions will be provided, based on the defined parameters of the sensor hardware.

The following paragraphs use the image recognition application as an example to explain the steps and aspects considered by the methodology supported by the proposed framework. In the selected scenario, the task of the sensor is to recognize and store the license plates for vehicles that enter or leave a secured car park.

First, it is very important to define the AI structure or technology that suits the task that the system shall perform the most. In addition to the above-mentioned neural networks in all their forms, other techniques, like, fuzzy logic, expert systems or decision trees can be chosen, but the choice must be properly justified. After choosing the right structure, it is necessary to specify its dimensions and it has to be configured properly, according to the target task. The first basic rule is to select the right algorithm for the problem. Using complex algorithms that can solve more tasks than required is inappropriate. Systems that require continuous CPU activity will have the worst energy efficiency.

The first very important point in choosing the AI technique is the information if the data is forming a stream of continuously related data items. In case such a relation exists, recurrent neural networks (RNN) [6] may be a good choice. Such a solution will also be good if we want to predict future behavior on the base of the collected data. RNNs are called recurrent, because their output depends on previous state that the structure stores in its memory. Applications based on RNNs include, for instance autonomous vehicles, where they help to avoid a road collision, based on previously collected data. Another example is the sound recognition. In the field of sensor networks RNNs can also be used for forecasting future phenomena, e.g., atmospheric. They can also help to solve production line optimization problems.

The convolutional neural networks [7] fit perfectly to the example scenario. This type of network allows identifying the license plates on the picture. This system must recognize the license plate in the picture and read the characters on it. This process is quite complex, so it is necessary to properly dimension the network, to determine the required number of convolutional layers. In order to reduce the energy consumption by limiting the image analysis to the time it is really needed, it is possible to use another approach typical for sensor networks. A simple solution is to use a detector, like a coil below the asphalt to detect a car or even a button to open the barrier. When the car is detected or the button is pressed, then the image recognition is activated. Otherwise, the system is in a low-power mode. If a license plate is detected, the text from the plate is sent to the database. Due to the small amount of data, sending is very quick. This also reduces the power consumption.

Finally, it is also important to identify the phases of the neural network that are involved in the application. It is usually the case that the learning phase is much more energy expensive than the classification (recognition). The selected scenario allows using the second step only, reducing the complexity and energy consumption.

5. FURTHER STEPS

The presented topic plays a crucial role in the development of intelligent sensors. These devices are embedded devices that are required to be energy efficient but also accurate. The discussed problems have to be addressed properly, if the result shall be positive. Thus, the further work is to define a methodology that defines steps that lead to a good design of an intelligent sensor.

The energy efficiency in wireless sensor networks based on AI does not only depend on the applied AI technologies, but is also determined by the programmer's approach. The first author will further investigate these issues as his PhD Thesis.

ACKNOWLEDGMENTS

This work was supported by the European Regional Development Fund within the BB-PL INTERREG V A 2014-2020 Programme, “reducing barriers – using the common strengths”, project SmartRiver, grant number 85029892 and by the German government under grant 01LC1903M (AMMOD project). The funding institutions had no role in the design of the study, the collection, analyses, or interpretation of data, the writing of the manuscript, or the decision to publish the results.

REFERENCES

1. Nielsen M., *Using neural nets to recognize handwritten digit* http://neuralnetworksanddeeplearning.com/chap1.html, last viewed 20.04.2020.
2. Intel Corporation, *Intel® Neural Compute Stick 2 (Intel® NCS2),* https://software.intel.com/en-us/neural-compute-stick, last viewed 20.04.2020
3. NVIDIA Corporation, *Embedded Systems for Next-Generation Autonomous Machines,* https://www.nvidia.com/en-us/autonomous-machines/embedded-systems, last viewed 20.04.2020
4. Yau S., *Battle of Edge AI — Nvidia vs Google vs Intel* ,https://towardsdatascience.com/battle-of-edge-ai-nvidia-vs-google-vs-intel-8a3b87243028, last viewed 20.04.2020
5. Vincent J., *Tesla's new AI chip,* https://www.theverge.com/2019/4/24/18514308/tesla-full-self-driving-computer-chip-autonomy-day-specs, last viewed 20.04.2020.
6. Venkatachalam M., *Recurrent Neural Networks*, https://towardsdatascience.com/recurrent-neural-networks-d4642c9bc7ce, last viewed 20.04.2020.
7. Saha S., *A Comprehensive Guide to Convolutional Neural Networks — the ELI5 way,* https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53, last viewed 20.04.2020.