SLOVAK UNIVERSITY OF TECHNOLOGY IN BRATISLAVA

FACULTY OF MATERIALS SCIENCE AND TECHNOLOGY IN TRNAVA

Power efficient deep learning on dedicated resource constrained devices

Offprint of Dissertation Thesis

Trnava 2023

Ing. Roman Budjač

SLOVAK UNIVERSITY OF TECHNOLOGY IN BRATISLAVA

FACULTY OF MATERIALS SCIENCE AND TECHNOLOGY IN TRNAVA

Ing. Roman Budjač

Offprint of Dissertation Thesis

Power efficient deep learning on dedicated resource constrained devices

for being awarded the academic degree: Doctor ("philosophiae doctor", abbrev. as "PhD.")

in PhD. study programme: 2621 Process Automation and Informatization

in study branch: Cybernetics

form of study: full-time

Trnava 29.5.2023

STU

Dissertation Thesis was elaborated at: the full-time form of doctoral studies at the Institute of Applied Informatics, Automation and Mechatronics, Faculty of Materials Science and Technology in Trnava on Slovak University of Technology in Bratislava

Submitted by:	Ing. Roman Budjač Institute of Applied Informatics, Automation and Mechatronics Faculty of Materials Science and Technology in Trnava Slovak University of Technology in Bratislava Jána Bottu 2781/25 917 24 Trnava
Supervisor:	doc. Ing. Juraj Ďuďák, PhD. Institute of Applied Informatics, Automation and Mechatronics Faculty of Materials Science and Technology in Trnava Slovak University of Technology in Bratislava Jána Bottu 2781/25 917 24 Trnava
Reviewers:	

Offprint was sent on:

Defence of Dissertation Thesis will be on............................ **at**.......................... **o'clock**.

At Faculty of Materials Science and Technology in Trnava, Jána Bottu 25, 917 24 Trnava, T02 pavilion.

....

Prof. Ing. Miloš Čambál, CSc.
MTF STU Faculty Dean

Content

Introduction

At present, it is popular to implement artificial intelligence in all technical devices wherever possible. The first prerequisite for using artificial intelligence is enabling data collection in the application domain. Based on this data, it is possible to create datasets and solve specific problems using AI algorithms, particularly deep neural networks. Such a position for artificial intelligence is reasonable.

Considering the European Commission's suggestion, which identified Artificial Intelligence technologies as key to the development and business competitiveness in the upcoming years [1]. The European Commission also considers Artificial Intelligence the most critical component of the Fourth Industrial Revolution. The European Commission estimates that the deployment of AI in key sectors of the European Union economies will increase the value of GDP by 1.8% by 2025. In the long term, the estimated impact of the implementation of AI in key industries is estimated to result in a cumulative increase in the contribution to GDP of 13.5%, depending on the industry [2]. Artificial intelligence (AI) is anticipated to have a significant positive influence on the manufacturing industry, particularly in the areas of the Industrial Internet of Things (IIoT), mobility, and intelligent healthcare.

Artificial Intelligence (AI) has increasingly become an integral component within various industrial sectors, offering innovative solutions and significant advancements in operational efficiency. Initially, its unique capacity for data interpretation, pattern recognition, and predictive decision-making has transformed traditional business models and strategic approaches.

The transformative potential of Artificial Intelligence for reshaping future industrial processes is considerable and, indeed, compelling. Significantly, AI is anticipated to enhance efficiency, productivity, safety, and decision-making prowess within industrial contexts.

Concurrently, supply chain and logistical sectors have benefitted immensely from AI's ability to optimise routes, accurately forecast demand, and improve inventory management systems. On a larger scale, AI is an essential component in the paradigm of Industry 4.0, providing the underpinnings for smart factories through autonomous systems and interconnected devices.

As of now, the field of deep learning has achieved impressive success in many visual recognition tasks. Undoubtedly, all progress achieved in this research area has been conducted with high computational expense and memory-intensive processes. This condition has resulted in the unfeasibility of devices to deploy deep learning models on devices with limited computation resources due to strict requirements. Specifically, convolution neural network models are generally over-parametrised where the model size exceeds the size of the training dataset. Therefore, a model that is neither over-parameterised nor under-parameterised but has a reasonable number of parameters represents the suitable solution for a task. As we have already stated, overparameterised neural models, which have many more parameters than necessary to solve a particular problem, entail a certain degree of redundancy. The existence of redundancy enables the application of neural network compression techniques. By compressing these models, it has the potential to significantly reduce their size without a significant decrease in model performance.

Consequently, it paves the way for research and exploration in this area. This research gap is of particular interest because it opens up the possibility of implementing deep neural networks on devices where this has been possible only with limitations.

Based on the presented motivation and the high relevance of the subject of Artificial Intelligence in this thesis, we have discussed the problem of reducing neural network models and resource-constrained devices with the potential for use in Industry 4.0 applications.

1 Objectives of the dissertation thesis

The aim of this dissertation can be summarised in the following points:

1. Elucidate the concepts of neural networks, machine learning, and deep learning.
2. Assess different compression techniques for deep neural networks.
3. Investigate the implementation of power-constrained devices within the Industry 4.0 paradigm.
4. Employ selected compression approach and propose an innovative strategy for training neural network.

The first objective of the dissertation was to clarify the concept of artificial intelligence, machine learning and the concept of neural networks. In the first part of the thesis, the aim was to introduce the problem of artificial intelligence. Next, in this part of the chapter, we focused on an essential subfield of machine learning, namely deep learning. We then established the definition of deep learning and explained the motivation behind deep learning and deep neural networks. Then we described the different categories of deep learning techniques, namely discriminative, generative, and hybrid learning. The next topic we addressed in this chapter was a detailed explanation of the basic concept of neural networks. In addition, we explained the basic concepts of neural networks and discussed the description of the essential components of neural network architecture. In addition, concepts such as weights, bias, neuron, and activation function and their role have been described. Subsequently, the basic concept of the neural network learning process, represented by two algorithms, was discussed. First is the Forward Pass, and second is the Backpropagation algorithm. Finally, the learning process was demonstrated on a complex example of forward transition and backpropagation computation. Then, some basic neural network training techniques were identified to serve as a prevention against neural network overlearning.

The second objective of this thesis was to conduct a literature review of neural network compression techniques. Particular emphasis was placed on two specific methods: pruning and quantisation. The theory underlying the complexity of neural network compression and the theory associated with the specific methods were also explained in detail. This comprehensive approach allowed a deep understanding of the topic and laid the foundation for further study in the field of neural network compression. In this thesis, the theoretical concepts underlying the various compression techniques have been elucidated in addition to a comprehensive literature review. In the case of pruning, the fundamentals were explored in detail, allowing for a deep understanding of this particular technique, also in the case of quantisation. A detailed review was given of the different types of quantisation, namely uniform and non-uniform. This theory analysis and a review of the existing literature have played a key role in forming a comprehensive view of neural network compression. In addition, the significant differences inherent in each technique have been described in detail.

The third aim of this thesis was to explore the potential of resource limited devices as part of the Industry 4.0 paradigm. Then, we discussed resource constrained embedded devices. After this, we focused on the importance of deep learning technologies and their current deployment in various industrial applications. Subsequently, we explained the importance of these devices in the context of deep learning and artificial intelligence in general as part of modern industrial processes. We then reviewed potentially suitable devices for application using deep learning. To address this, we presented methods and hardware to enhance the effectiveness of deep learning. Finally, we focused on the procedures involved in applying the quantisation technique and examined the hardware parameters of a particular device. Special attention is provided to Industrial IoT- IIoT devices and IoT devices designed to work with Artificial Intelligence - AIoT. We explained their application potential in the industry and the position of these devices in the role of IoT, IIoT and AIoT and discussed their application possibilities. We concluded that devices that would meet industry standard designs in terms of design and have hardware would greatly benefit industrial processes. In particular, they would significantly enhance the Edge Computing paradigm, which is part of the Industry 4.0 concept. As a result, these resource-constrained devices can be placed in a different category, namely AIoT. We have also peripherally mentioned the potential of power-limited devices for SMEs concerning their specific business needs.

This dissertation's fourth and final objective was to propose a neural network training strategy to improve the training process in the neural network quantisation problem. First, we proposed a modification of the neural network training process, with the addition of an index, that uses data from the original neural network during the training of the quantised neural model. This method is based on statistical methods to compare the distribution of the weights of the original and quantised models. This information is then provided during training the quantised model as an indicator that can be further manipulated. We then discuss the possibilities of comparing the distribution of weights in the context of neural network training and its quantisation.

2 Artificial intelligence

Artificial intelligence, also called machine intelligence, is the ability of the computer to simulate the human ability to solve problems based on mathematical rules while using computational power thorough the programming language. In other words, AI can be a number of explicitly programmed if-then statements or complex machine learning model mapping sensory data as input from, for example, an electric device. The study of knowledge engineering is an essential aspect of AI research. Software needs to have bountiful information related to the world, with often an expectation to be precise in decision-making like human beings. AI must have access to properties, categories, objects and relations between all of them to implement knowledge engineering. AI initiates common sense, problem-solving and analytical reasoning power in machines, which is more complex and tedious [3]. Figure 1 shows subfields of artificial intelligence.

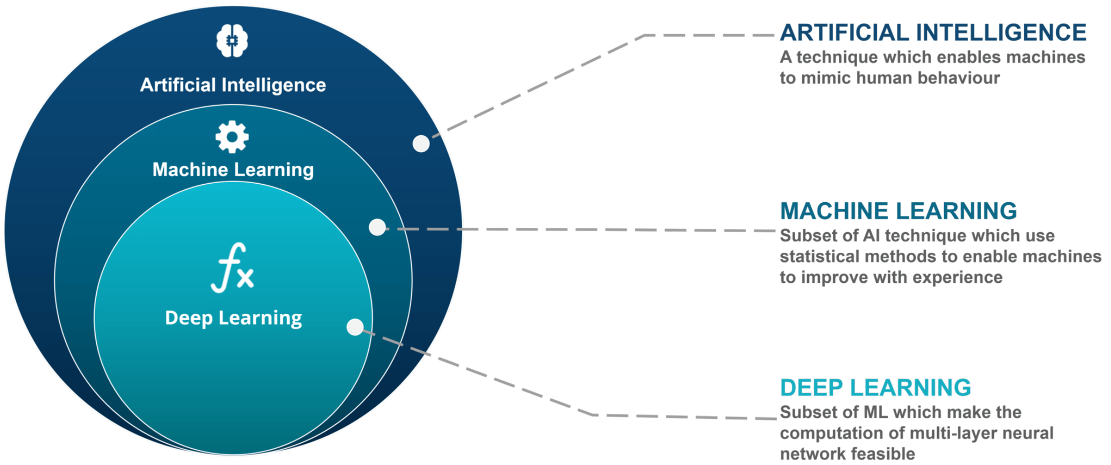

Artificial Intelligence
Machine Learning
Deep Learning
ARTIFICIAL INTELLIGENCE
A technique which enables machines to mimic human behaviour
MACHINE LEARNING
Subset of AI technique which use statistical methods to enable machines to improve with experience
DEEP LEARNING
Subset of ML which make the computation of multi-layer neural network feasible

Figure 1 Field of artificial intelligence [4]

2.1 Neural networks' basic concept

The concept of a neural network was inspired by the neuron cell in the human brain. A neuron is a core component of the human neural system [5]. Information is transmitted via a structure called the synapsis, which enables the transmission of neural signals from one interconnected neuron to another. Neurons have four essential functions: receiving information, transmitting information, storing information and communicating signals to target cells. It is particularly important to realise that artificial neurons do not function in the same way as biological neurons. The idea is inspired by triggering the neurons but is not identical to it.

The core idea behind neural networks is to develop a computational model that emulates the structure and functionality of biological neural networks, such as those present in the human brain. The primary objective is to empower machines with the ability to learn, identify patterns, and make informed decisions or predictions using input data [6]. A neural network comprises multiple interconnected layers of artificial neurons or units, also called nodes. Each neuron accepts data inputs from its antecedent neurons in the neural system. Following data processing, the neuron conveys the resultant output to the succeeding layer within the network. The connections between neurons are characterised by weights, which dictate the intensity of influence that one neuron exerts on another. Neurons employ activation functions to apply the non-linearity concept to the model, allowing neural networks to learn intricate patterns and relationships within the data. The functions mentioned previously are utilised to operate on the summation of inputs, weights, and biases that have been accumulated within each neuron. This process finally decides the output that will be released and transmitted to other neurons. The learning process within a neural network revolves around adjusting the weights and biases based on the discrepancies between the network's actual output and the desired output. This is typically achieved using an optimisation algorithm, such as gradient descent, in conjunction with a process known as backpropagation. Backpropagation computes the gradients of the error concerning each weight and bias by applying the chain rule, iteratively calculating the gradient from the output layer back to the input layer. This gained information is subsequently used to update the weights and biases, directly leading to minimising the overall error. Various types of neural networks exist. These include feed-forward networks, recurrent networks, and convolutional networks. A unidirectional flow of information characterises feed-forward networks, while recurrent networks allow feedback loops and are especially useful for processing sequential data. Convolutional networks are particularly suited for image recognition and computer vision tasks, employing convolutional layers that can automatically learn spatial hierarchies from the input data. Neural networks have gained widespread adoption across diverse fields, such as computer vision, natural language processing, speech recognition, and decision-making systems, owing to their flexibility and effectiveness in handling complex tasks. The ongoing progress and implementation of neural networks are propelling substantial breakthroughs in the fields of artificial intelligence and machine learning.

2.2 Deep Learning Models

Different learning paradigms in machine learning are being recognised, including discriminative, generative, and hybrid learning. s an overview of each in Figure 2. Discriminative models achieve their results by a model architecture that accurately represents. the decision boundary between different classes. Examples of such discriminative models include Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) [7]. The second technique is generative. Generative models pivot their primary focus towards unravelling the intrinsic distribution of the data [8]. The third techniques are hybrid models. As suggested by the name, endeavour to fuse the strengths of both the discriminative and generative models [9].

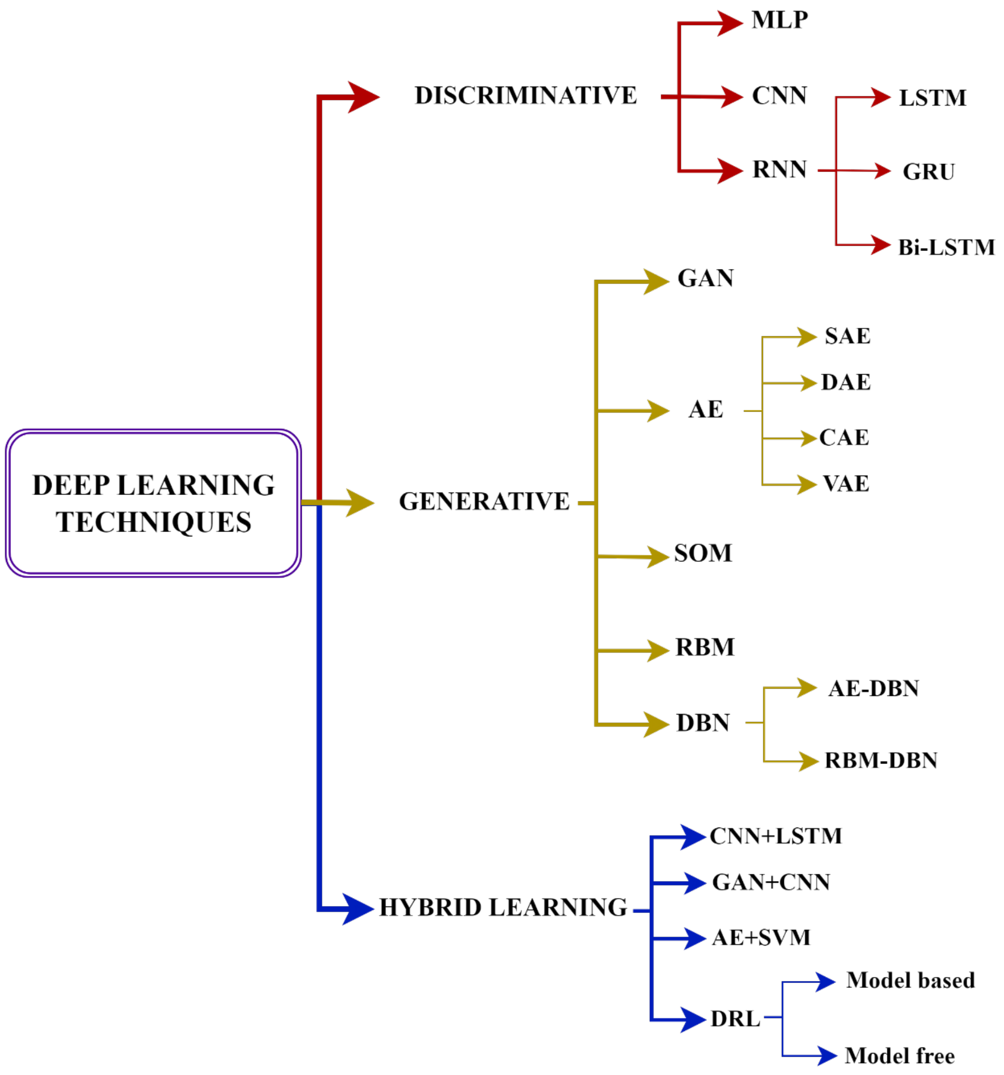


Figure 2 Deep learning techniques

3 Compressing neural network review

According to a literature review, neural network models are mostly over-parametrised. Furthermore, it is even desirable for the initial stages of researching complex tasks. In complex tasks, it is essential that the model is large enough to solve the problem. Especially in the case of a higher number of deep layers, a large number of parameters is inevitable. A high number of parameters allows us to train neural networks more efficiently. Another advantage of an over-parameterised neural network is due to the presence of more suitable landscape loss structures. Thirdly, over-parametrised models may have fewer "erroneous" local minima that could degrade the optimisation process. However, despite these advantages, over-parametrisation is not an optimal situation, even if the computational power is not considered constrained. Therefore, we consider over-parametrised networks as a limiting factor. Especially when attempting to extend the application of deep neural networks to resource-constrained devices such as IoT, IIoT and AIoT devices.

Compression of neural networks seemed to be a possible method to solve this problem. The outcome of the conducted research was identifying several neural network reduction methods and confirming their validity based on a literature search.

Neural network compression has enormous potential and great importance in the field of artificial intelligence. Specifically in the problem of optimising deep learning models. The primary purpose of such compression is its ability to reduce the computational resource requirements that these neural models need. Optimally, we can reach a reduction of the impact on their performance to a minimum. This reduction in model complexity correspondingly reduces the memory footprint and computational demands. This enables the deployment of deep learning models on devices with limited computational capabilities, such as IoT, IIoT, AIoT and thus supports the field of edge computing. In addition, compressed neural networks can alleviate power consumption and latency issues.

Consequently, it enhances the scalability and responsiveness of applications. For example, in real-time applications. Latency is often a factor, and deploying compressed models can speed up inference times. Similarly, for battery-dependent devices, reducing power consumption through model compression can extend the operational runtime of the device.

Following a literature review, two leading methods for neural network compression have been identified for the purposes of this thesis. Specifically, pruning and quantisation. The principle of pruning is to trim off individual neurons with the intention of trimming specific neurons or weights with the objective of removing neurons that do not contribute to the learning of the network. The process of determining which parameters to trim involves optimisation elements. This is one of the disadvantages of pruning. In addition, the search algorithm during the implementation of pruning requires the design of a search strategy and extra computational power to implement the pruning. Based on the above, we consider pruning to be a suitable method with convincing results. Figure 3 clearly demonstrates a fully connected network and a pruned network.

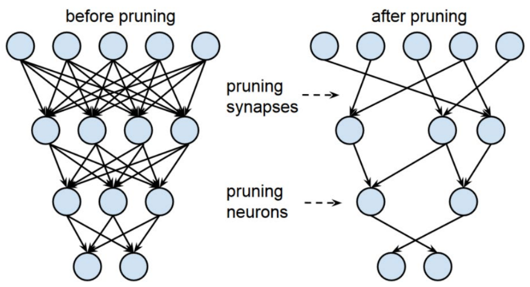


Figure 3 Synapses and neurons before and after pruning [10]

Therefore, the second leading method is quantisation, shown in Figure 4. Fundamentally, quantisation encompasses the deliberate reduction in the precision of numerical representations that are used for the weights within the network. To illustrate, a model that has been trained initially employing 32-bit floating point numbers might subsequently undergo quantisation to use 8-bit. Two types of quantisation exist, uniform and non-uniform, with the latter adjusting quantisation levels according to value distribution and providing better accuracy, though with increased implementation complexity. Weight and activation values can be quantised separately or simultaneously, with the latter option yielding maximum savings. Quantisation can also be applied during training, allowing the model to learn to compensate for quantisation-induced errors or after training, which is a simpler and faster process but may cause more significant accuracy loss.

In contrast to pruning, we identify that quantisation has several advantages:

- Straightforward implementation,
- greater reduction in model size compared to pruning,
- computational operations with lower precision are less demanding,
- quantisation does not modify the architecture configuration of the network,
- modern hardware accelerators have native support for quantised operations.

Despite the undeniably significant potential of neural network compression, it should be mentioned that the process is not without its limitations. Compression methods' effectiveness and the subsequent effect on the neural model is a significant subject for discussion. The efficiency is indisputable; we can select between aggressive compression and more subtle methods. Particular compression methods or selection of compression parameters can result in a significant loss of model accuracy on the test set. This leads to designing and refining compression techniques capable of effectively reducing the model size without significant performance degradation. Currently, these problems remain active areas of research.

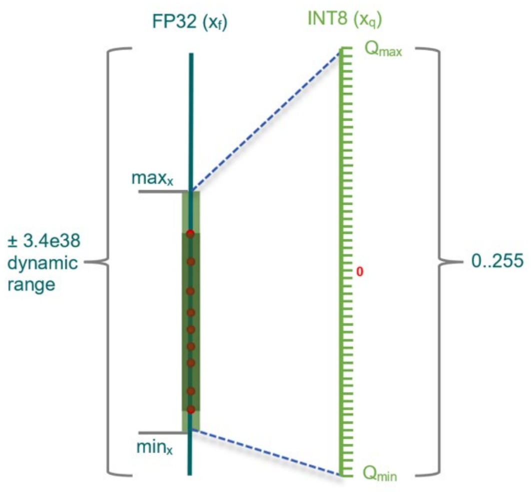


Figure 4 FP32 to INT8 quantisation schema [11]

Overall, the potential and importance of neural network compression underline its vital role in advancing artificial intelligence technologies and expanding the potential applications of deep neural networks to resource-constrained devices.

4 Resource-constrained devices

Based on our analysis, it is clear that deploying AI-based tasks using deep learning requires specialised hardware. This dissertation thesis analyses and discusses the relation between IoT devices, IIoT devices and AIoT devices [12].

As we have found in this thesis, such hardware is not commonly available in IoT devices. This stems from the nature of the nature of their activities has not yet led to such a requirement. With the advancing efforts to deploy artificial intelligence through deep learning as part of the Industry 4.0 transformation, it is also expected to process deep network algorithms at the location of the application. From our perspective, this is precisely the place for the demand for compact embedded devices that could leverage deep learning algorithms. In this thesis, we have focused on understanding IoT devices in Industry 4.0 and their adaptation in industrial environments. In particular, certified hardware production and adaptation to external environmental conditions are required to implement IoT devices in the industry successfully. By adapting to the environment, we mean adapting to dust conditions and water resistance standards concerning international standards. IoT devices that meet the industrial design standard for industrial applications can be defined as Industrial IoT devices or IIoT devices. Let us move forward with the development of these devices. If we extend the idea of these IIoT devices to include the requirements for implementing deep neural network algorithms, it would involve a condition that such devices are equipped with the proper hardware for neural network acceleration. Therefore, a new term for such devices is established as Artificial Intelligence IoT devices or AIoT. We have found the relationship between the IoT industrial standards and the capability of accelerating neural networks to be synergistic, as Figure 5 shows. As a result, each technological feature overlap enhances the benefit of the previous technology. This further highlights the requirement for multi-disciplinarity in device deployment when AIoT equipment meets industry standards. From our perspective, there will be a strong demand for these devices in the following years, and we will see a great effort to implement these devices in Industry 4.0. For now, the market of AIoT devices that would meet the industrial standard is not large enough. In this thesis, we have selected potential devices which serve as neural network accelerators. We have described the selected devices and their technical parameters. Most of the devices had one common problem.

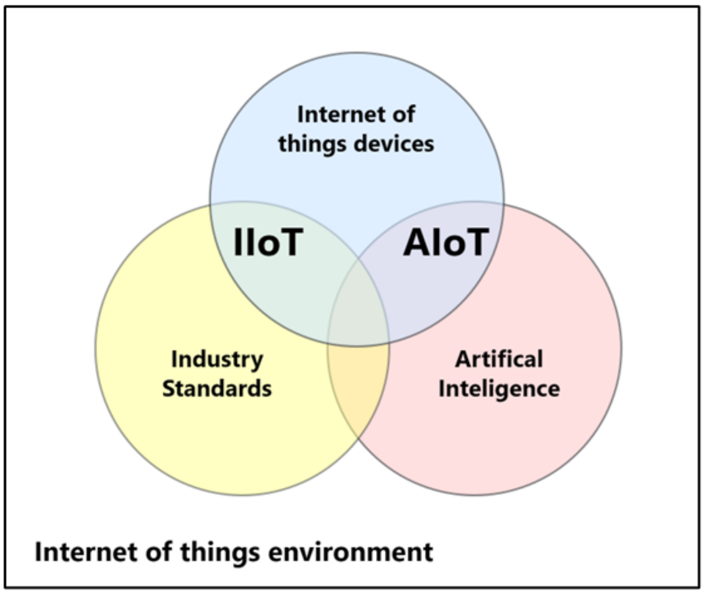


Figure 5 Internet of things environment

The majority of the devices do not meet the industrial standard. These devices, such as google coral or Nvidia Jetson devices, have proper software support by the manufacturer. They also support frameworks like Tensorflow, PyTorch and Keras. They have excellent connectivity and interface support.

As a result of our efforts, a comparison with the key attributes of Industry 4.0 utilising the Jetson device was provided. However, manufacturers recommend using them for development purposes. The exception is some Nvidia Jetson devices. For example, the Jetson NX Xavier comes in a System on Module (SoM) version. In this case, the computing unit is in the form of a card that can be embedded in its own device - a board carrier. The second device chosen was the Jetson Orin. This device is the only one in the Jetson series that has the ability to adhere to industry standards of design while being fully capable of accelerating neural networks. Its features allow to produce autonomous machines at the edge. The result of our efforts was a comparison with the key attributes of Industry 4.0. Additionally, our research discussed resource-constrained devices from small and medium businesses' points of view. We explored their application potential in the industry, the position of these devices in the role of IoT, IIoT and AIoT, and discussed the possibilities of their application. Small and medium businesses can particularly benefit from the fact that these facilities, even when using AIoT devices, have a relatively low cost. According to our review, in the case of SMEs, the purchase price is one of the essential aspects of the decision-making process. We concluded that devices that would meet the industrial design standard and, at the same time, have deep learning-ready hardware would greatly benefit industrial processes. In particular, they would significantly enhance the Edge Computing paradigm, which is part of the Industry 4.0 concept. This paradigm involves performing computation at or near the data source instead of relying on a centralised location away from the data source, such as cloud servers. In this case of deploying high-performance AIoT devices, advanced deep learning algorithms can be used at the point of first data acquisition. In addition, this concept involves performing computations at or near the data source instead of relying on a centralised location outside the data source, such as cloud servers. All of this is to promote IoT as one of the cornerstones of Industry 4.0, with an emphasis on expanding support for AI applications.

To clarify the issue, it is necessary to distil two other terms from the term IoT, namely Artificial Intelligence of Things and Industrial Internet of Things. These two concepts fully complement the potential of IoT in industry, where using AI on edge makes it possible to replace conventional methods with AI methods on the devices whose hardware architecture and software allow it. Several devices can be categorised as AIoT. Since no standard definition of such devices exists, we have based our definition on the attributes. Among the devices we have turned much of our attention to is Nvidia's Jetson category of devices. These devices feature robust hardware architectures for the acceleration of neural networks. Nvidia Jetson devices are designed to infer deep neural networks. They can also be categorised as AIoT based on their Internet connection or internal industrial network connection. The Jetson Nano and Jetson Xavier NX support the network's latest connectivity standards. Such specialised architecture and connectivity enable the use of advanced real-time neural network model architectures. Thus, in Table 1 we compare some of the characteristics of the devices from the perspective of Industry 4.0, whether they are promising for the fourth generation of industry, and if they meet the requirements for Industry 4.0 IoT, Big Data, AI, Augmented Reality, Simulation, Horizontal and Vertical Integration, Additive Manufacturing, Autonomous Robots, Cloud Computing and Cybersecurity [1].

Table 1 Evaluation of selected devices in terms of Industry 4.0 attributes [1]

Industry 4.0 Technologies	**Jetson AGX**	**Jetson Xavier NX**	**Jetson Nano**
IoT	Yes	Yes	Yes
Big Data	Yes	Yes	Yes
Artificial Intelligence	Yes	Yes	Yes
Augmented Reality	Yes	Yes	Yes
Autonomous Robots	Yes	Yes	Yes
Simulation	Yes	Yes	Yes
Horizontal and Vertical Integration	No	No	No
Additive Manufacturing	No	No	No
Cloud Computing	Yes	Yes	Yes
Cybersecurity	Yes	Yes	Yes

5 Proposal of strategy for training neural network based on weight distribution comparison

Comparing distributions is a fundamental aspect of statistical analysis that is applied to various domains, including industrial processes. This comparison helps to detect patterns, understand the underlying structure, and draw inferences about the data [13]. By comparing distributions, it is possible to quantify differences, monitor performance, optimise processes, and assess risks [14]. This introduction will discuss the importance of comparing distributions in industrial settings and provide examples of how it is applied in practice.

Nowadays, statistics is an indispensable tool for analysing and interpreting the collected data results. A fundamental step in this analysis is comparing distributions between different groups or within a single group over time. The methods for comparing distributions can be different. Each offers its own unique advantages and limitations. In what follows, we will attempt to compare selected methods to gain a deeper understanding of their characteristics and to decide in which situations their use is most appropriate.

In industrial processes, probability distributions can be used to model various aspects of the system, such as the quality of products, sensor readings, equipment lifetimes, production times, and failure rates. Comparing these distributions can provide valuable insights into the performance of the process and identify potential areas for improvement or risk mitigation [15].

One common application of comparing distributions in industrial processes is quality control. For example, consider a manufacturing plant that produces ball bearings. The diameters of the bearings should adhere to a specific tolerance range. By comparing the distribution of the diameters of the manufactured bearings with the expected distribution, deviations from the standard can be detected [16]. Identifying these deviations allows the manufacturer to take corrective action, ensuring the products meet the required quality standards. Another application is process optimisation. For example, in a chemical plant, the distribution of the output product's purity might be of interest. The optimal conditions that maximise product purity can be determined by comparing the purity distribution under different process conditions, such as varying temperature, pressure, or reactant concentration.

In predictive maintenance, comparing distributions can help detect anomalies in sensor data that may indicate equipment failure. For instance, suppose a machine's temperature readings typically follow a specific distribution. In that case, a sudden shift in the distribution might indicate a malfunction that requires maintenance to prevent unexpected breakdowns and production downtime. Risk management is another area where comparing distributions plays a crucial role. In an industrial setting, comparing the distribution of losses under different scenarios can help identify the situations that represent the most significant risk. This information can be used to develop strategies for risk mitigation, such as increasing safety measures, adjusting production schedules, or purchasing insurance coverage.

In conclusion, comparing distributions is a vital tool for analysing and improving industrial processes. By quantifying differences between distributions, it is possible to identify patterns, assess risks, and optimise performance. Applications include quality control, process optimisation, predictive maintenance, and risk management, among others. Companies can enhance their operations and make more informed decisions by understanding and applying the principles of comparing distributions.

5.1 Proposal of implementations statistics methods

The comprehension of weight distribution is of vital significance in the domain of machine learning, particularly in neural networks. Various weight distributions can result in distinct model efficacy, robustness, and convergence rate degrees.

The comparison of distributions holds a vital position in machine learning, wherein the weight distribution selection can substantially influence the training procedure and the comprehensive efficacy of the model. Therefore, this research aims to analyse and compare the properties and effectiveness of different distributions used in neural network training in the context of quantising neural network weights.

Various methods exist for measuring statistics to compare:

- Kullback-Leibler divergence,
- Jensen-Shannon Divergence.

Kullback-Leibler divergence

In this case, the KL divergence is utilised to measure the difference between the probability distributions of the weights in the initial model and the quantised model. The statement denotes the degree of dissimilarity among the learned parameters of the two models. The Kullback-Leibler divergence is a non-negative real number, whereby a value of zero signifies that the two probability distributions are indistinguishable, and higher values indicate a greater extent of divergence [17]. When implementing quantisation on a neural network model, it is preferable to maintain the model's performance as closely as possible to its original state. Consequently, conducting an analysis of the KL divergence can prove to be beneficial in assessing the impact of the quantisation process on the model parameters. In this instance, the KL divergence is employed to quantify the disparity between the probability distributions of the weights in the original model and the quantised model. It indicates how dissimilar the two models' learned parameters are.

The KL divergence is a positive number, where a value of 0 indicates that the two probability distributions are identical, and more significant values indicate a greater degree of dissimilarity. In summary, the KL divergence exhibits a range of values between 0 and positive infinity. The following is an explanation of the meaning towards those values. The KL divergence is zero if and only if the two distributions being compared are the same [18]. In other words, every possible event has the same probability in both distributions. This represents a perfect match. With an increase in divergence, it becomes clear that the distinction between the distributions also escalates. It is important to note that this value is not capped, meaning there is no upper boundary. Consequently, this divergence could manifest as a huge value in cases where the two distributions are remarkably different. [19]. When applying quantisation to a neural network model, keeping the model's performance as near to the original as feasible is desirable. Therefore, it is helpful to analyse the KL divergence to determine how much the quantisation procedure has affected the model parameters.

Hence, analysing the KL divergence can prove advantageous in comprehending how much the quantisation process has impacted the model parameters. Nevertheless, it is essential to note that the KL divergence does not have symmetric properties, as shown by the existing inequality below.

$$\mathrm{KL}(p, q) \neq \mathrm{KL}(q, p). \tag{1}$$

This is because the measurement solely pertains to the quantised model's deviation from the original model, not vice versa. A reduced KL divergence value indicates that the parameters of the quantised model are close to those of the original model, indicating that the quantisation procedure has effectively maintained the model's properties. Conversely, a higher KL divergence value signifies a substantial deviation of the quantised model's parameters from the original model, which may impact the model's efficacy.

Although KL divergence offers valuable insights into the dissimilarity between the model parameters, it does not provide direct information on the effect on the model's performance. Therefore, evaluating the performance of the quantised model on a test dataset is essential in determining its effectiveness relative to the original model.

$$R(d;q) = KL(M_d \parallel M_q) = \sum_{t \in V} P(t|M_q) \log \frac{P(t|M_q)}{P(t|M_d)} \tag{2}$$

Jensen-Shannon Divergence

The Jensen-Shannon Divergence (JSD) is a mathematical metric that evaluates the similarity between two probability distributions in a symmetric manner. For example, the quantity mentioned above results from the Kullback-Leibler Divergence (KLD) and is both non-negative and bounded [20]. Therefore, the Jenson-Shannon Divergence (JSD) is a viable method for evaluating the similarity between probability distributions in various fields such as information theory, natural language processing, and machine learning [21].

The formula for JSD is as follows:

$$\mathrm{JSD}(P, Q) = (1/2) * (\mathrm{KLD}(P, M) + \mathrm{KLD}(Q, M)) \tag{3}$$

Where P and Q are the two probability distributions being compared, KL divergence is the Kullback-Leibler Divergence, and M is the average distribution defined as:

$$M = (1/2) * (P + Q) \tag{4}$$

JS divergence holds a fundamental characteristic wherein it consistently assumes a finite numerical value that stays within the range of 0 to 1. This approach is highly beneficial in distributions with non-overlapping supports, as KL divergence would result in an unbounded outcome [22].

A key aspect, and indeed a crucial advantage, of the Jensen-Shannon (JS) divergence lies in its bounded nature. Notably, it consistently produces values confined within the interval [0,1] when represented in bits. This specific characteristic proves to be particularly beneficial under certain conditions. For instance, when comparing probability distributions with non-overlapping supports is necessary, a scenario presents a challenge for Kullback-

Leibler (KL) divergence as it would lead to an undefined or infinite outcome. Hence, the finite and bounded nature of JS divergence provides a robust solution in such circumstances.

5.2 A proposed method for implementing weight distribution comparison of quantised and non-quantised neural network

The primary purpose of this section is to understand the properties and performance of two comparing different weight distributions in our neural network models. Comparing distributions is particularly important because we will know what happens to the model during or after training. In our design, we thus have information on how the weight distribution changes during training. This is essential information because the values of the weights change significantly during backpropagation. Consequently, even more, when in our case, we use the comparison of weight distributions to compare the original trained model and the model after quantising the weights. In our case, we have designed a new neural network training strategy designed for comparing the weight distributions of the original and the quantised model. In this way, we are able to obtain information about the effect of quantisation directly on the model weights in an independent way. In this case, we used the architecture of convolutional neural networks and fully connected layers.

We considered two ways. The first way was to train the neural network on an unconstrained dataset. This involved the standard procedure of pre-processing the dataset if the data in it had not been modified previously. Subsequently, split the dataset into a training set and a test set. The next step was to train the neural network. After training, we stored the model and weights of a separate dataset.

The second step was to quantise the trained model. After several experiments, we decided to choose uniform quantisation as the quantisation method. Non-uniform quantisation proved to be ineffective during our experimentation for several reasons. We did achieve a first-time random distribution of numbers in the quantisation range, but there was an intense degradation in the prediction of the neural network output. Further use of neural network regeneration techniques was also unsuccessful. Thus, there was an irreversible degradation of the model, which led to the need to re-train the neural network from scratch. The next logical choice was to use uniform quantisation. Accordingly, this is based on the fact that all quantisation levels are equally distributed. Uniformity refers to the fact that these levels are equally distributed across the entire range of values that can be quantised. Consequently, the difference between two adjacent quantisation levels is always constant.

This quantisation method has proven to be significantly better for our purposes, especially regarding the impact of the quantisation process on the model.

Figure 6 shows a conceptual diagram of the procedure of the first proposed method of comparing the distribution of weights of a quantised and unquantised neural network. The diagram shows the dataset from which we make a copy during training. One for the original model without quantisation and the other for the quantised model. The following is a typical portion of training a convolutional neural network on the dataset. It includes the following procedure:

The experimentation requires a certain duration until the desired network performance is achieved. Since it is an iterative process, it is necessary to change the hyperparameters. After successfully completing the convolutional neural network training, the next step was to save the model in the Keras framework. In this way, it is possible to recover the saved model and continue with predictions or further training without losing the learned weights.

As part of this algorithm, in the beginning, it is possible to choose the quantisation degree, the quantisation parameter we can choose from 4 to 32 bits. Since non-uniform quantisation has not been successful in our implementation, we only use uniform quantisation in this process. In our opinion, uniform quantisation best represents the original distribution of weights and can also produce the expected quantisation effect. After the quantisation parameters are set, the next step is to create a copy of the original (reference) model. This copy will be the input for the uniform quantisation process according to the quantisation parameters.

In order to avoid the early degradation effects of quantisation, we further retrained the model after this process. This model was no longer trained from the beginning, instead we used only a small number of iterations for retraining. The number of iterations of retraining is not fixed in any way due to it is based on empirical rules. The number of iterations of retraining depends on the number of epochs needed to train the whole network. It can be in units of repetitions or even in tens, depending on the complexity of the model. However, this step is optional. In fact, ideally, after quantisation, the model would not need any further processing. Then the effect of exploiting the quantisation potential would be most significant.

The quantisation process is followed by the storage of the quantised weights. At this point, we have stored the original and quantised weights. The following is the part where we used statistical methods. We have stored their final distributions from the originally stored weights and the new quantised weights. We then used these distributions as input to the comparison algorithm. In this section, we compared the distributions of the original model before quantisation and after quantisation with the selected statistical method. Namely, we tested the Kullback-Leibler divergence. Nevertheless, using the Jensen-Shannon divergence or the Cosine similarity is also possible. This information is represented by a number – an index, which is provided on the screen and can be evaluated afterwards. This index describes the similarity of the tested distributions. If the distributions are significantly different, it is possible to assume a degradation of the weights in the model; hence, the quantisation had an undesirable impact on the network. We used the Keras framework for the implementation. In the next section, we propose a modified version of this training strategy to use the similarity index directly during training.

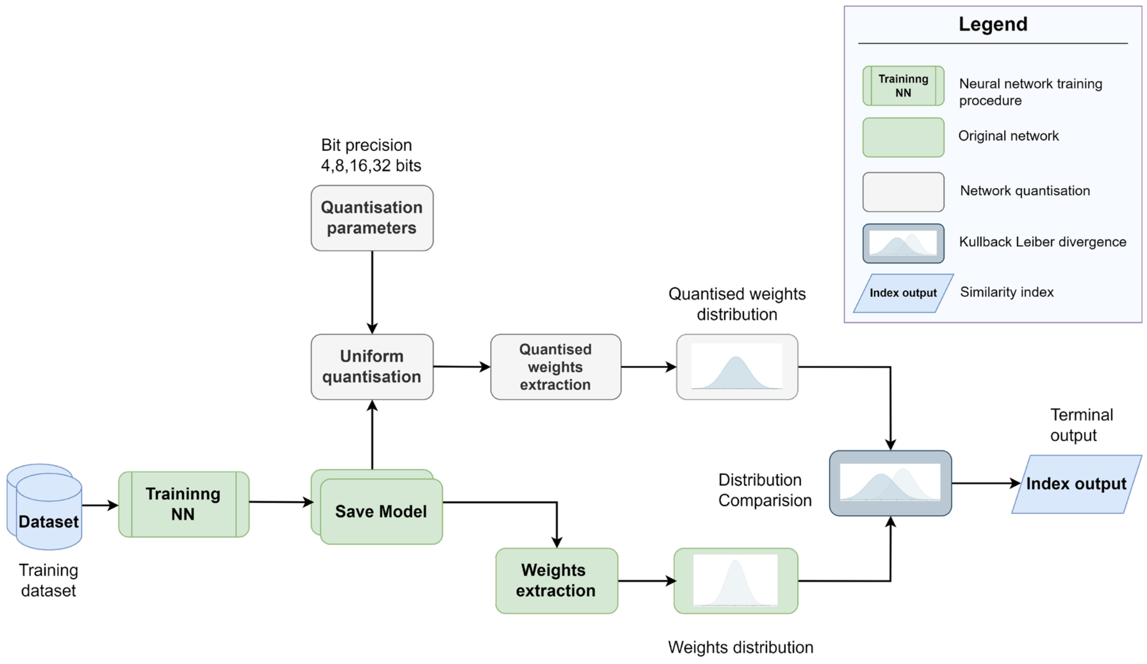


Figure 6 Proposed method for training neural networks based on weight distribution comparison after quantisation

Figure 7 shows a conceptual diagram of the procedure of the enhanced technique proposed method of comparing the distribution of weights of a quantised and non quantised neural network. The diagram shows the dataset from which we make a copy during training. One for the original model without quantisation and the other for the quantised model. The following is a typical portion of training a convolutional neural network on the dataset. It includes the same training procedure as we mentioned above.

This strategy differs from the previous one mainly in the aspect of utilising information about the change in distribution between the quantised and the unquantised model. In this proposed training strategy, we used the similarity index directly during training. This technique aims to provide the user with information during training about the similarity of the distribution of the weights in the quantised and in the original unquantised one. Each epoch similarity index is calculated during the quantised neural network training. Hence, a similarity index between the original weight distribution and the current weight distribution will be available every epoch.

The procedure presented in this context helps interpret and facilitates the comprehension of the model's behaviour throughout the training process. Further, it provides information directly during training about the dissimilarity of the weight distributions during training each epoch (each iteration of training). Obtaining such information can help us, for example, in deciding to terminate the training of the quantised network early. In this case, if it is clear that the model's accuracy does not grow during the training process. At the same time, if the difference between the distributions is significant, we can change the quantisation parameters and repeat the whole process.

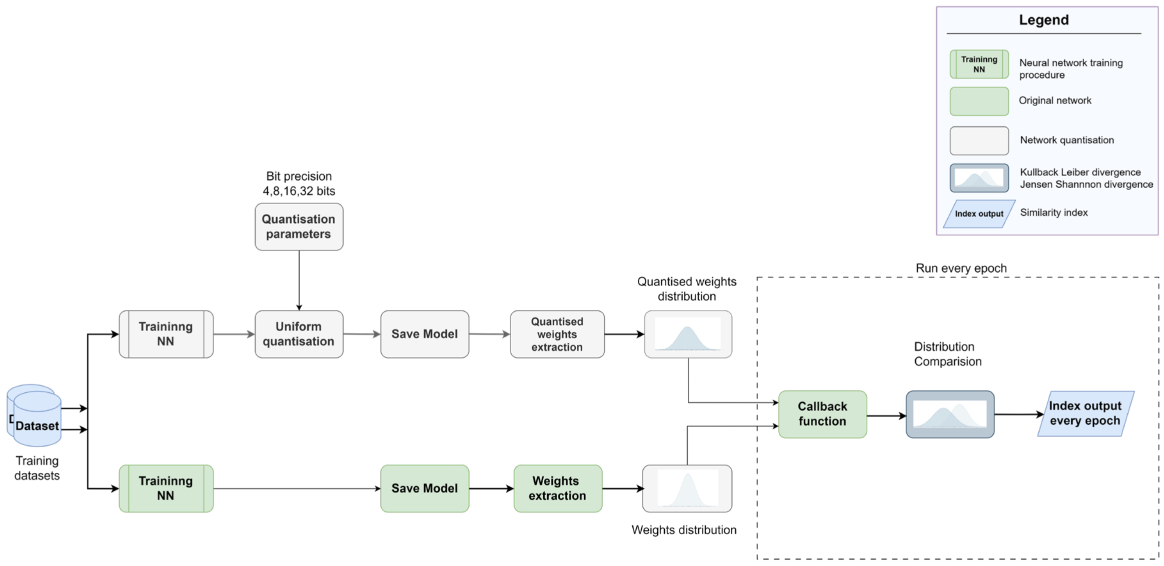


Figure 7 Proposed method for training neural networks based on weight distribution comparison during training

Especially in the case of a considerable number of epochs (e.g. 50 or more), it is a helpful tool for analysing the training status. Moreover, it provides interesting additional information, which can be useful during the training of neural networks using quantisation. The method call to compute the selected statistical method is provided by the CallBack class. Callbacks are set of methods called at various stages of training, testing, and predicting. Callbacks are useful to obtain a view of internal states and statistics of the model during training.

The values of the resulting index depend on the selected statistical method. Among the methods presented in the previous chapter, the Kullback-Leibner divergence was the most interesting. Although the latter is not bounded, it is relatively intuitive to infer the magnitude of the difference between the distributions as an auxiliary index from the magnitude of the difference. One advantage of applying these statistical methods, such as KL divergence and JS divergence, is that they are not computationally intensive. From our observations, we did not observe a significant impact on the training progress, even in the case of computing multiple similarity indexes.

This strategy is intended to support the process of reducing the model size for deployment on devices with constrained computational resources that involve rounding or reducing the accuracy of the weights. Furthermore, comparing the weight distributions of the quantised and original model can help in deciding how to use quantisation techniques use in order to determine the impact of quantisation on the model in terms of the possibility of its degradation. In addition, this technique was designed to facilitate the training and quantisation of neural networks, particularly concerning use in training in order to further utilise the models on resource-constrained devices.

5.3 Experiments and results

In this section, we have chosen to perform experiments based on the proposal in the section above. This series of experiments was conducted on various datasets, aiming to deepen our understanding of their inherent structures and characteristics. We used convolutional networks to classify object types for our purposes.

At the beginning of the experiments, we trained the selected neural networks without constraints and used a simple custom or convolution neural network architecture. Generally, we used a network with varying depths depending on the depth of the network needed, based on the complexity of the dataset. After training the neural network, we stored the parameters of the trained neural model. These parameters contained all the parameters found in the model.

Upon completing our neural network model training phase, an essential next step that naturally follows is saving the model's learned parameters. Therefore, we extracted the weights from the stored model parameters from each layer of this model. We believe that the weight distribution of the neural model can provide valuable insights into the learning process and the model's overall behaviour. Specifically, this analysis can elucidate whether the learning process exhibits signs of convergence and to what extent the weight updates during training contribute to the meaningful progress of the model's learning.

Following this initial assessment, we focus on implementing a quantisation process on the corresponding networks derived from these datasets as an essential technique within the realm of compression. Quantisation effectively simplifies the network, reducing its complexity while preserving its essential features. Furthermore, through this process, we can transform higher-dimensional data into a more manageable format, thereby rendering it more suitable for detailed analysis and interpretation.

The primary aim underlying these experiments was twofold. Firstly, we sought to devise a technique tailored explicitly towards the training and quantisation of neural networks. Secondly, the objective leads to support facilitating the training of neural networks designed for resource-constrained devices. In our experimentation, we used a feedback approach. During training, we used information from the original trained neural network.

Procedure of experiments

In the following tables, the enhanced second proposal results are represented. We have focused on implementing the Kullback-Leibler divergence and Jensen-Shannon divergence into single epochs during neural network training. Thus, we output the Kullback-Leibler divergence and Jensen-Shannon divergence in each iteration for comparing the weight distribution of the trained quantised model and the original model's weight distribution. As a result, we output an index that serves as an indicator during training. We used uniform quantisation for these experiments. In this case, an eight-bit degree of quantisation is implemented. The difference between original and quantised neural networks was calculated using Kullback-Leibler divergence and Jensen-Shannon divergence.

We performed an experiment calculating KL and JS divergence in the following tables. Then, we applied our proposal to the MNIST dataset with seven hidden neural layers. The aim was to verify the concept.

Architecture:

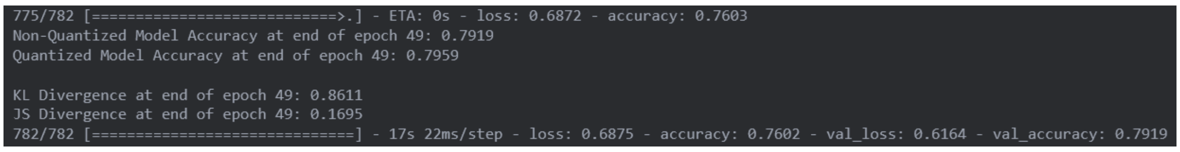
- Conv2D Layer with 32 filters (First Hidden Layer)
- MaxPooling2D Layer
- Dropout Layer
- Conv2D Layer with 64 filters (Second Hidden Layer)
- MaxPooling2D Layer

- Dropout Layer
- Conv2D Layer with 128 filters (Third Hidden Layer)
- MaxPooling2D Layer
- Dropout Layer

Table 2 Results of an experiment implementing KL divergence and JS divergence in the process of training neural networks - CIFAR10

Dataset	**Model**	**Epoch**	**Quantisation**	**KL divergence [-]**	**JS divergence [-]**	**Accuracy quantised [%]**	**Accuracy original [%]**
CIFAR 10	CNN	0	8 bit	0.6203	0.1261	56.92	49.99
		1	8 bit	0.6681	0.1355	62.18	58.35
		2	8 bit	0.7023	0.1417	65.75	62.76
		…	…	…	…	…	…
		40	8 bit	0.8516	0.1680	77.41	78.70
		41	8 bit	0.8512	0.1681	78.70	78.06
		42	8 bit	0.8527	0.1685	78.68	78.64
		43	8 bit	0.8567	0.1689	77.99	77.66
		44	8 bit	0.8582	0.1692	77.74	78.43
		45	8 bit	0.8585	0.1693	79.19	78.93
		46	8 bit	0.8586	0.1693	78.79	79.26
		47	8 bit	0.8570	0.1691	78.46	79.97
		48	8 bit	0.8579	0.1692	79.85	79.39
		49	8 bit	0.8611	0.1695	79.59	79.19

In the Figure 8 we can see the training results, which are listed in the table above (Table 2).



```
775/782 [============================>.] - ETA: 0s - loss: 0.6872 - accuracy: 0.7603
Non-Quantized Model Accuracy at end of epoch 49: 0.7919
Quantized Model Accuracy at end of epoch 49: 0.7959

KL Divergence at end of epoch 49: 0.8611
JS Divergence at end of epoch 49: 0.1695
782/782 [==============================] - 17s 22ms/step - loss: 0.6875 - accuracy: 0.7602 - val_loss: 0.6164 - val_accuracy: 0.7919
```

Figure 8 Results of an experiment implementing KL divergence and JS divergence in Python - CIFAR10

We performed an experiment calculating KL and JS divergence in the following tables. Then, we applied our proposal on the MNIST dataset with seven hidden neural layers. The aim was to verify the concept.

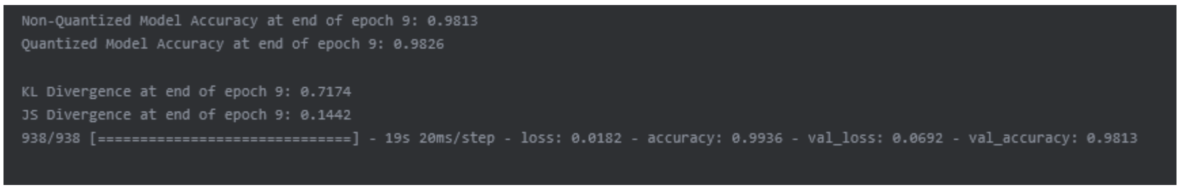
Architecture:

- Flatten Layer (Input Layer)
- Dense Layer with 512 neurons (First Hidden Layer)
- Dropout Layer
- Dense Layer with 256 neurons (Second Hidden Layer)
- Dropout Layer
- Dense Layer with 10 neurons (Output Layer)

Table 3 Results of an experiment implementing KL divergence and JS divergence in the process of training neural networks - MNIST

Dataset	**Model**	**Epoch**	**Quantisation**	**KL divergence [-]**	**JS divergence [-]**	**Accuracy quantised [%]**	**Accuracy original [%]**
MNIST	Custom	0	8 bit	0.6144	0.1248	97.25	96.64
		1	8 bit	0.6523	0.1320	97.53	97.47
		2	8 bit	0.6739	0.1361	97.88	97.94
		3	8 bit	0.6880	0.1386	98.14	98.17
		4	8 bit	0.6985	0.1405	97.97	98.08
		5	8 bit	0.7024	0.1414	98.20	97.98
		6	8 bit	0.7072	0.1422	98.02	98.16
		7	8 bit	0.7099	0.1429	98.16	98.00
		8	8 bit	0.7127	0.1435	98.31	98.17
		9	8 bit	0.7174	0.1442	98.26	98.13

In the Figure 9 we can see the results of the training, which are listed in the table above (Table 3).



```
Non-Quantized Model Accuracy at end of epoch 9: 0.9813
Quantized Model Accuracy at end of epoch 9: 0.9826

KL Divergence at end of epoch 9: 0.7174
JS Divergence at end of epoch 9: 0.1442
938/938 [==============================] - 19s 20ms/step - loss: 0.0182 - accuracy: 0.9936 - val_loss: 0.0692 - val_accuracy: 0.9813
```

Figure 9 Results of an experiment implementing KL divergence and JS divergence in Python - MNIST

We performed an experiment calculating KL and JS divergence in the following tables. Then, we applied our proposal to the FMNIST dataset with seven hidden neural layers. The aim was to verify the concept.

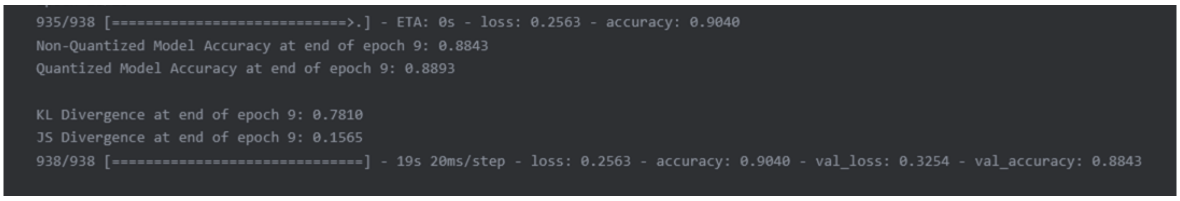
Architecture:

- Flatten Layer (Input Layer)
- Dense Layer with 512 neurons (First Hidden Layer)
- Dropout Layer
- Dense Layer with 256 neurons (Second Hidden Layer)
- Dropout Layer
- Dense Layer with 10 neurons (Output Layer)

Table 4 Results of an experiment implementing KL divergence and JS divergence in the process of training neural networks - FMNIST

Dataset	**Model**	**Epoch**	**Quantisation**	**KL divergence [-]**	**JS divergence [-]**	**Accuracy quantised [%]**	**Accuracy Original [%]**
FMNIST	Custom	0	8 bit	0.6144	0.1248	86.24	83.61
		1	8 bit	0.6523	0.1320	86.45	86.43
		2	8 bit	0.6739	0.1361	87.32	86.13
		3	8 bit	0.6880	0.1386	87.45	87.52
		4	8 bit	0.6985	0.1405	87.20	87.55
		5	8 bit	0.7024	0.1414	87.30	87.27
		6	8 bit	0.7072	0.1422	88.19	87.21
		7	8 bit	0.7099	0.1429	87.51	88.17
		8	8 bit	0.7127	0.1435	88.33	87.87
		9	8 bit	0.7174	0.1442	88.93	88.43

In the Figure 10 we can see the results of the training, which are listed in the table above (Table 4).



```
935/938 [============================>.] - ETA: 0s - loss: 0.2563 - accuracy: 0.9040
Non-Quantized Model Accuracy at end of epoch 9: 0.8843
Quantized Model Accuracy at end of epoch 9: 0.8893

KL Divergence at end of epoch 9: 0.7810
JS Divergence at end of epoch 9: 0.1565
938/938 [==============================] - 19s 20ms/step - loss: 0.2563 - accuracy: 0.9040 - val_loss: 0.3254 - val_accuracy: 0.8843
```

Figure 10 Results of an experiment implementing KL divergence and JS divergence in Python - FMNIST

The above experiments were made mainly to prove the concept. In addition, we show that the proposed training strategy is implementable in a conventional convolutional network training procedure.

During training, this method is advantageous when the resulting iteration accuracy of training the neural network does not grow several iterations ahead of time, and our indicator shows a significant difference between the iteration distribution from the original (unquantised) neural network.

Since we interfere with the training process and compute the dissimilarity of the weight distribution during each iteration, it can potentially slow down the training. However, there was no significant slowing down of the training process during the execution of the experiments. Problems could arise with large data sets. As we can see by observing the values of KL and JS indices with the increasing number of iterations, the distributions change. We can also observe that as the number of iterations increases during training, the values of the KL divergence and JS divergence indexes also increase in all models. This indicates that the distributions changed during the training of the quantised neural model. The continuous increase of these indices without a significant leap in values means that there were no rapid and drastic changes in the model caused by quantisation. The advantage of using the index for similarity distributions of the weight distributions is that we can verify the changes during the training of the quantised model and track them independently with the two indexes. Thus, we can track the changes in the model and detect any significant effects that may cause model degradation promptly. We can use this to terminate training earlier and subsequently change the quantisation parameters or other neural network training parameters. Thus, this is an additional measurement parameter. We have provided this index as part of a new neural network training strategy to quantise neural networks. We have tested this proposal's feasibility and tested this concept on three datasets.

Summary

In our dissertation, we addressed the topic of deep learning and resource-constrained devices. First, we analysed the current state of the art of neural networks from the perspective of the needs of resource-constrained devices. The higher degree of over-parameterisation of neural network models has created room for increasing their efficiency. One of the tools to address redundancies in neural network models is neural network compression. Reducing over-parameterised models has opened the way to using deep neural networks on resource-constrained devices. In this context, we have addressed the issue of resource-constrained devices and their potential in Industry 4.0. We have identified the industry requirements for such devices in the industry. Ideally, these IoT devices would meet the industrial standard while having hardware capable of accelerating neural networks. The result would be an industrial AIoT device capable of participating in the implementation of AI algorithms in an industrial environment. Our research has identified potentially suitable devices for implementing deep learning in the industry based on the characteristics expected for the transformation to Industry 4.0 within the IoT concept. Subsequently, we addressed neural network compression techniques, namely pruning and quantisation. We performed a theoretical analysis of these compression methods and conducted a literature review where we evaluated the different approaches. This comprehensive review provided a complete perspective of the problem and was a prerequisite for further work.

The last part of this thesis was devoted to designing a strategy for training neural networks based on statistical methods. We performed this comparison to use the distribution of the weights of the original model and the quantised model as an indicator of the state of the network during training. First, a theoretical analysis was performed in which different methods of comparing distributions based on statistical methodologies were presented. Then, based on this theoretical foundation, we proposed a novel strategy for training neural networks and comparing weight distribution. In particular, we used this comparison of distributions as an indicator of training status during neural network quantisation. This procedure gave us insight into the current state of the neural network and allowed us better to understand the impact of quantisation on the training process. Finally, throughout the quantised neural network training process, we extracted data regarding the distribution of weights from the original network. We then used statistical methods to compare the weight distributions of the original unquantised neural network and the ongoing training epoch of the quantised network. The result was a distribution similarity index that we can work with and evaluate. This metric proved particularly valuable in facilitating the decision to terminate training early, especially in cases where the difference in the network was insignificant. In conclusion, this dissertation's results support the field of automation while utilising resource-limited devices in industrial processes as part of the Industry 4.0 paradigm.

References

[1] M. Barton, R. Budjac, P. Tanuska, G. Gaspar, a P. Schreiber, “Identification Overview of Industry 4.0 Essential Attributes and Resource-Limited Embedded Artificial-Intelligence-of-Things Devices for Small and Medium-Sized Enterprises”, *Applied Sciences*, roč. 12, č. 11, Art. č. 11, jan. 2022, doi: 10.3390/app12115672.

[2] “Industrial applications of artificial intelligence and big data”. https://ec.europa.eu/growth/industry/strategy/advanced-technologies/industrial-applications-artificial-intelligence-and-big-data_en (cit 03. marec 2022).

[3] P. Ariwala, “Artificial Intelligence and Machine Learning made simple”, *Maruti Techlabs*. https://marutitech.com/artificial-intelligence-and-machine-learning/ (cit 22. január 2023).

[4] T. Shieh-Newton, PhD, a M. McConihe, “Patenting Considerations for Artificial Intelligence in Biotech and Synthetic Biology – Part 2: Key Issues in Patent Subject Matter Eligibility | Mintz”. https://www.mintz.com/insights-center/viewpoints/2231/2020-01-30-patenting-considerations-artificial-intelligence-biotech (cit 02. apríl 2020).

[5] T.-H. Huynh a M. Yoo, “A Taillight Matching and Pairing Algorithm for Stereo-Vision-Based Nighttime Vehicle-to-Vehicle Positioning”, *Applied Sciences*, roč. 10, č. 19, Art. č. 19, jan. 2020, doi: 10.3390/app10196800.

[6] W. S. McCulloch a W. Pitts, “A logical calculus of the ideas immanent in nervous activity”, *Bulletin of Mathematical Biophysics*, roč. 5, č. 4, s. 115–133, dec. 1943, doi: 10.1007/BF02478259.

[7] G. Huang, Z. Liu, L. Van Der Maaten, a K. Q. Weinberger, “Densely Connected Convolutional Networks”, v *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017, s. 2261–2269. doi: 10.1109/CVPR.2017.243.

[8] Z. Zhang, M. Li, a J. Yu, “D2PGGAN: Two Discriminators Used in Progressive Growing of GANS”, v *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, máj. 2019, s. 3177–3181. doi: 10.1109/ICASSP.2019.8683262.

[9] Z. Dai, Z. Yang, F. Yang, W. W. Cohen, a R. Salakhutdinov, “Good semi-supervised learning that requires a bad GAN”, v *Proceedings of the 31st International Conference on Neural Information Processing Systems*, v NIPS’17. Red Hook, NY, USA: Curran Associates Inc., dec. 2017, s. 6513–6523.

[10] S. Han, H. Mao, a W. J. Dally, “Deep Compression: Compressing Deep Neural Network with Pruning, Trained Quantization and Huffman Coding”, v *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*, Y. Bengio a Y. LeCun, Ed., 2016. [Online]. Available at: http://arxiv.org/abs/1510.00149

[11] N. Zmora, G. Jacob, L. Zlotnik, B. Elharar, a G. Novik, “Neural Network Distiller: A Python Package For DNN Compression Research”, *ArXiv*, roč. abs/1910.12232, 2019.

[12] H. Boyes, “The industrial internet of things (IIoT)_ An analysis framework | Elsevier Enhanced Reader”. https://reader.elsevier.com/reader/sd/pii/S0166361517307285?token=CCD4C935CF533E9D04A0C18DB7FCDA780ACC95074E448D703BB615B13B183ECD0C7A0C4011D51766B72FC0F53099699E&originRegion=eu-west-1&originCreation=20220304133828 (cit 04. marec 2022).

[13] D. C. Montgomery, *Statistical quality control: a modern introduction*, Seventh edition, International student version. Hoboken, NJ: Wiley, 2013.

[14] X. Chen, B. L. Nelson, a K. K. Kim, "Stochastic kriging for conditional value-at-risk and its sensitivities: 2012 Winter Simulation Conference, WSC 2012", *Proceedings of the 2012 Winter Simulation Conference, WSC 2012*, 2012, doi: 10.1109/WSC.2012.6465096.

[15] M. King, *Process Control: A Practical Approach*, 1st ed. Wiley, 2016. doi: 10.1002/9781119157779.

[16] R. K. Mobley, *An introduction to predictive maintenance*, 2nd ed. Amsterdam ; New York: Butterworth-Heinemann, 2002.

[17] T. M. Cover a J. A. Thomas, *Elements of Information Theory*, 1st ed. Wiley, 2005. doi: 10.1002/047174882X.

[18] S. Kullback a R. A. Leibler, "On Information and Sufficiency", *Ann. Math. Statist.*, roč. 22, č. 1, s. 79–86, mar. 1951, doi: 10.1214/aoms/1177729694.

[19] I. Sason a S. Shamai, "Performance Analysis of Linear Codes under Maximum-Likelihood Decoding: A Tutorial", *FNT in Communications and Information Theory*, roč. 3, č. 1/2, s. 1–222, 2006, doi: 10.1561/0100000009.

[20] Zhibo Fan a Kai-Fong Lee, "Hankel transform domain analysis of dual-frequency stacked circular-disk and annular-ring microstrip antennas", *IEEE Trans. Antennas Propagat.*, roč. 39, č. 6, s. 867–870, jún. 1991, doi: 10.1109/8.86891.

[21] L. Zinchenko a S. Sorokin, "Fitness estimations for evolutionary antenna design", v *NASA/DoD Conference on Evolvable Hardware, 2003. Proceedings.*, Chicago, IL, USA: IEEE Comput. Soc, 2003, s. 155–164. doi: 10.1109/EH.2003.1217661.

[22] M. Oizumi, L. Albantakis, a G. Tononi, "From the Phenomenology to the Mechanisms of Consciousness: Integrated Information Theory 3.0", *PLoS Comput Biol*, roč. 10, č. 5, s. e1003588, máj. 2014, doi: 10.1371/journal.pcbi.1003588.

List of publications

ADC Scientific papers in foreign peer-reviewed journals

ADC01 BARTOŇ, Martin - BUDJAČ, Roman - TANUŠKA, Pavol - GAŠPAR, Gabriel - SCHREIBER, Peter. Identification Overview of Industry 4.0 Essential Attributes and Resource-Limited Embedded Artificial-Intelligence-of-Things Devices for Small and Medium-Sized Enterprises. In *Applied Sciences*. Vol. 12, iss. 11 (2022), s. 1-26. ISSN 2076-3417 (2021: 2.838 - IF, Q2 - JCR Best Q, 0.507 - SJR, Q2 - SJR Best Q). In database: DOI: 10.3390/app12115672 ; SCOPUS: 2-s2.0-85131736165 ; WOS: 000808625000001 ; CC: 000808625000001. [Faculty category: M*A]. Output Type: article; Outcome: foreign; Publication category since 2022: V3; Publication category to 2021: ADC

ADC02 GAŠPAR, Gabriel - ĎUĎÁK, Juraj - BEHÚLOVÁ, Mária - STRÉMY, Maximilián - BUDJAČ, Roman - ŠEDIVÝ, Štefan - TOMAŠ, Boris. IoT-Ready Temperature Probe for Smart Monitoring of Forest Roads. In *Applied Sciences*. Vol. 12, iss. 2 (2022), s. 1-21. ISSN 2076-3417 (2021: 2.838 - IF, Q2 - JCR Best Q, 0.507 - SJR, Q2 - SJR Best Q). In database: DOI: 10.3390/app12020743 ; SCOPUS: 2-s2.0-85122865683 ; WOS: 000747021200001 ; CC: 000747021200001. [Faculty category: M*A]. Output Type: article; Outcome: foreign; Publication category since 2022: V3; Publication category to 2021: ADC

ADF Scientific papers in other national journals

ADF01 BARTOŇ, Martin - BUDJAČ, Roman - TANUŠKA, Pavol - SCHREIBER, Peter - HORÁK, Tibor. Industry communication based on TCP/IP protocol. In *Vedecké práce MtF STU v Bratislave so sídlom v Trnave. Research papers Faculty of Materials Science and Technology Slovak University of Technology in Trnava*. Vol. 29, no. 49 (2021), s. 59-66. ISSN 1336-1589. In database: INSPEC ; DOI: 10.2478/rput-2021-0025. [Faculty category: M*B]. Output Type: article; Outcome: national; Publication category since 2022: V3; Publication category to 2021: ADF

ADF02 BUDJAČ, Roman - NIKMON, Marcel - SCHREIBER, Peter - ZAHRADNÍKOVÁ, Barbora - JANÁČOVÁ, Dagmar. Automated machine learning overview. In *Vedecké práce MtF STU v Bratislave so sídlom v Trnave. Research papers Faculty of Materials Science and Technology Slovak University of Technology in Trnava*. Vol. 27, no. 45 (2019), s. 107-112. ISSN 1336-1589. In database: DOI: 10.2478/rpu-2019-0033 ; INSPEC. [Faculty category: M*B]. Publication category since 2022: V3; Publication category to 2021: ADF

ADF03 NIKMON, Marcel - BUDJAČ, Roman - KUCHÁR, Daniel - SCHREIBER, Peter - JANÁČOVÁ, Dagmar. Convolutional networks used to classify video and audio data. In *Vedecké práce MtF STU v Bratislave so sídlom v Trnave. Research papers Faculty of Materials Science and Technology Slovak University of Technology in Trnava*. Vol. 27, no. 45 (2019), s. 113-120. ISSN 1336-1589. In database: DOI: 102478/rput-2019-0034 ; INSPEC. [Faculty category: M*B]. Publication category since 2022: V3; Publication category to 2021: ADF

AFC Published papers at foreign scientific conferences

AFC01 BUDJAČ, Roman - BARTOŇ, Martin - SCHREIBER, Peter - SKOVAJSA, Martin. Analysing Embedded AIoT Devices for Deep Learning Purposes. In *Artificial Intelligence Trends in Systems : 11th Computer Science On-line Conference 2022 (CSOC 2022), Vol.2*. 1. ed. Cham : Springer Nature, 2022, S. 434-448. ISSN 2367-3370. ISBN 978-3-031-

09075-2. In database: DOI: 10.1007/978-3-031-09076-9_39 ; SCOPUS: 2-s2.0-85135064552 ; WOS: 000893642100039. [Faculty category: M*B]. Output Type: Conference paper; Outcome: foreign; Publication category since 2022: V2; Publication category to 2021: AFC

AFC02 ĎUĎÁK, Juraj - GAŠPAR, Gabriel - ŠEDIVÝ, Štefan - BUDJAČ, Roman. Utilisation of RTOS Solutions in IoT Modules based on RISC Microcontrollers. In *Cybernetics Perspectives in Systems : 11th Computer Science On-line Conference 2022, Vol. 3 (CSOC 2022)*. 1. ed. Cham : Springer Nature, 2022, S. 80-93. ISSN 2367-3370. ISBN 978-3-031-09072-1. In database: DOI: 10.1007/978-3-031-09073-8_8 ; SCOPUS: 2-s2.0-85135074870 ; WOS: 000892638700008. [Faculty category: M*B]. Output Type: Conference paper; Outcome: foreign; Publication category since 2022: V2; Publication category to 2021: AFC

AFC03 HORÁK, Tibor - ŠIMON, Marek - HURAJ, Ladislav - BUDJAČ, Roman. Vulnerability of smart IoT-based automation and control devices to cyber attacks. In *Applied Informatics and Cybernetics in Intelligent Systems. CSOC 2020 : 9th Computer Science On-line Conference 2020 (CSOC 2020), April 23,2020 - April 26,2020*. 1. ed. Cham : Springer, 2020, S. 287-294. ISSN 2194-5357 (print). ISBN 978-3-030-51973-5 (print). In database: DOI: 10.1007/978-3-030-51974-2_27 ; SCOPUS: 2-s2.0-85089620744. [Faculty category: M*B]. Publication category since 2022: V2; Publication category to 2021: AFC

AFC04 NIKMON, Marcel - BUDJAČ, Roman. Classification of motor imagery tasks for brain-computer interface with SVM classifiers. In *Proceedings of the 11th International Doctoral Seminar (IDS 2019). 30th Central European Conference on Information and Intelligent Systems : 11th International Doctoral Seminar, Varaždin, Croatia, 3rd October 2019*. 1. ed. Varaždin : University of Zagreb, 2019, S. 26-28. ISBN 978-953-6071-67-8. [Faculty category: M*C]. Publication category since 2022: V2; Publication category to 2021: AFC

AFC05 ZAHRADNÍKOVÁ, Barbora - BUDJAČ, Roman - SCHREIBER, Peter - RYDZI, Štefan. Directed Evolution of Human Facial Images. In *IFAC-PapersOnLine*. Vol. 53, iss. 2: IFAC World Congress, Berlín, Germany, 12-17 July 2020 (2020), s.16518-16523. ISSN 2405-8963 (2020: 0.308 - SJR, Q3 - SJR Best Q). In database: DOI: 10.1016/j.ifacol.2020.12.768 ; WOS: 000652593600524 ; SCOPUS: 2-s2.0-85119594922. [Faculty category: M*A]. Output Type: Conference paper; Outcome: foreign; Publication category since 2022: V2; Publication category to 2021: AFC

AFD Published papers at national scientific conferences

AFD01 BUDJAČ, Roman - MARKECHOVÁ, Iveta - STÚPALOVÁ, Hana. A notelet to enrichment of elementary calculus via sigmoid & elliptic curve. In *XXXII Didmattech 2019 [International Scientific and Professional Conferece, Trnava 20th - 21st June 2019]*. 1. ed. Trnava : Trnavská univerzita, 2019, S. 1-7. ISBN 978-80-568-0398-1. [Faculty category: M*C]. Publication category since 2022: V2; Publication category to 2021: AFD

AFH Abstracts of papers from national conferences

AFH01 BUDJAČ, Roman - MARKECHOVÁ, Iveta - STÚPALOVÁ, Hana. Activation function as an inspiration for metamaterial design and gyroid as inspiration for activation function design. In *DMSRE 2022 : Development of materials science in research and education, 31st joint seminar, 05. - 08. 09. 2022, Nová Lesná, SR*. 1. ed. Bratislava : FCHPT STU, 2022, S. 32. ISBN 978-80-8208-086-8. [Faculty category: M*D].

Output Type: abstract ; Outcome: national; Publication category since 2022: V2; Publication category to 2021: AFH

BCI Scripts and textbooks

BCI01 SCHREIBER, Peter - BUDJAČ, Roman - NIKMON, Marcel. *Inteligentné metódy riadenia : Návody na cvičenia*. 1. ed. Trnava : AlumniPress, 2021. 110 s. Edícia skrípt. ISBN 978-80-8096-278-4. [Faculty category: M*D]. Output Type: skriptum; Outcome: national; Publication category since 2022: P1; Publication category to 2021: BCI

Statistics: Publication category since 2021

ADC	Scientific papers in foreign peer-reviewed journals	2
ADF	Scientific papers in other national journals	3
AFC	Published papers at foreign scientific conferences	5
AFD	Published papers at national scientific conferences	1
AFH	Abstracts of papers from national conferences	1
BCI	Scripts and textbooks	1
Sum		**13**

Statistics: Citations category from 2022

1 Citation in a publication registered in citation indexes				**19**
	Foreign		19	
2 Citation in a publication, including a citation in a publication registered in databases other than citation indexes				**6**
	Foreign		6	
Sum				**25**