IMPROVING PERFORMANCE OF DEEP NEURAL NETWORKS BY DEVELOPING NOVEL ACTIVATION FUNCTIONS

PhD Thesis – Abstract

To obtain a PhD degree at

Politehnica University Timișoara

In Computers and Information Technology

author ing. Marina Adriana MERCIONI

Thesis supervisor Prof.univ.dr.ing. Ștefan HOLBAN

In the thesis entitled „Improving performance of deep neural networks by developing novel activation functions" are presented a multivariate series of activation functions developed along time in Deep Learning [1] domain. The thesis is structured in five parts having a total of eight chapters and contains more subchapters (Fig.1. and Fig.2.) divided as follows:

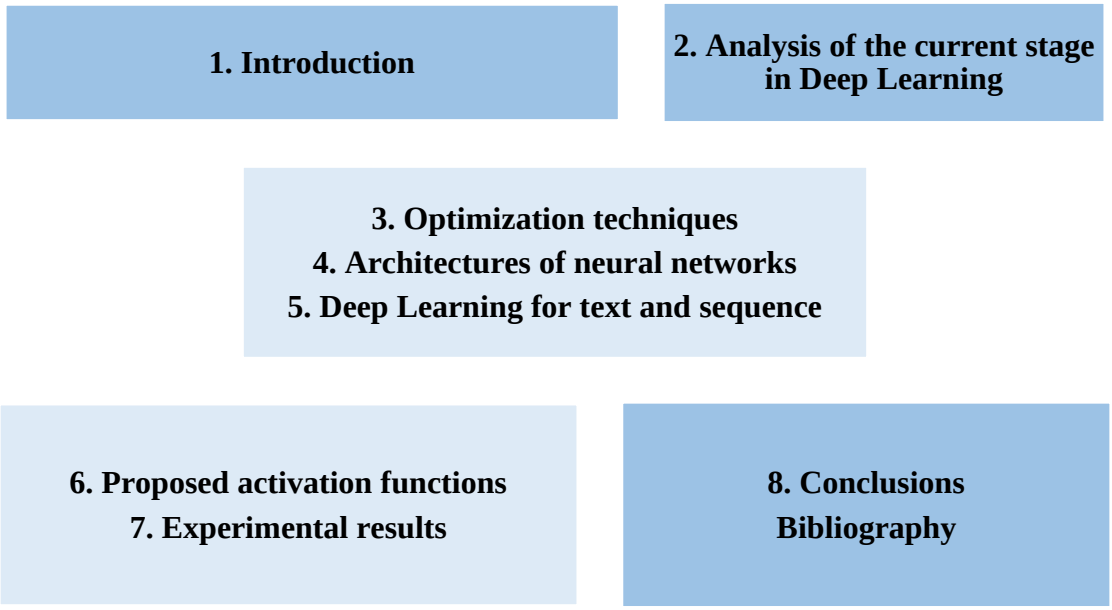

Fig.1. The architecture of the thesis

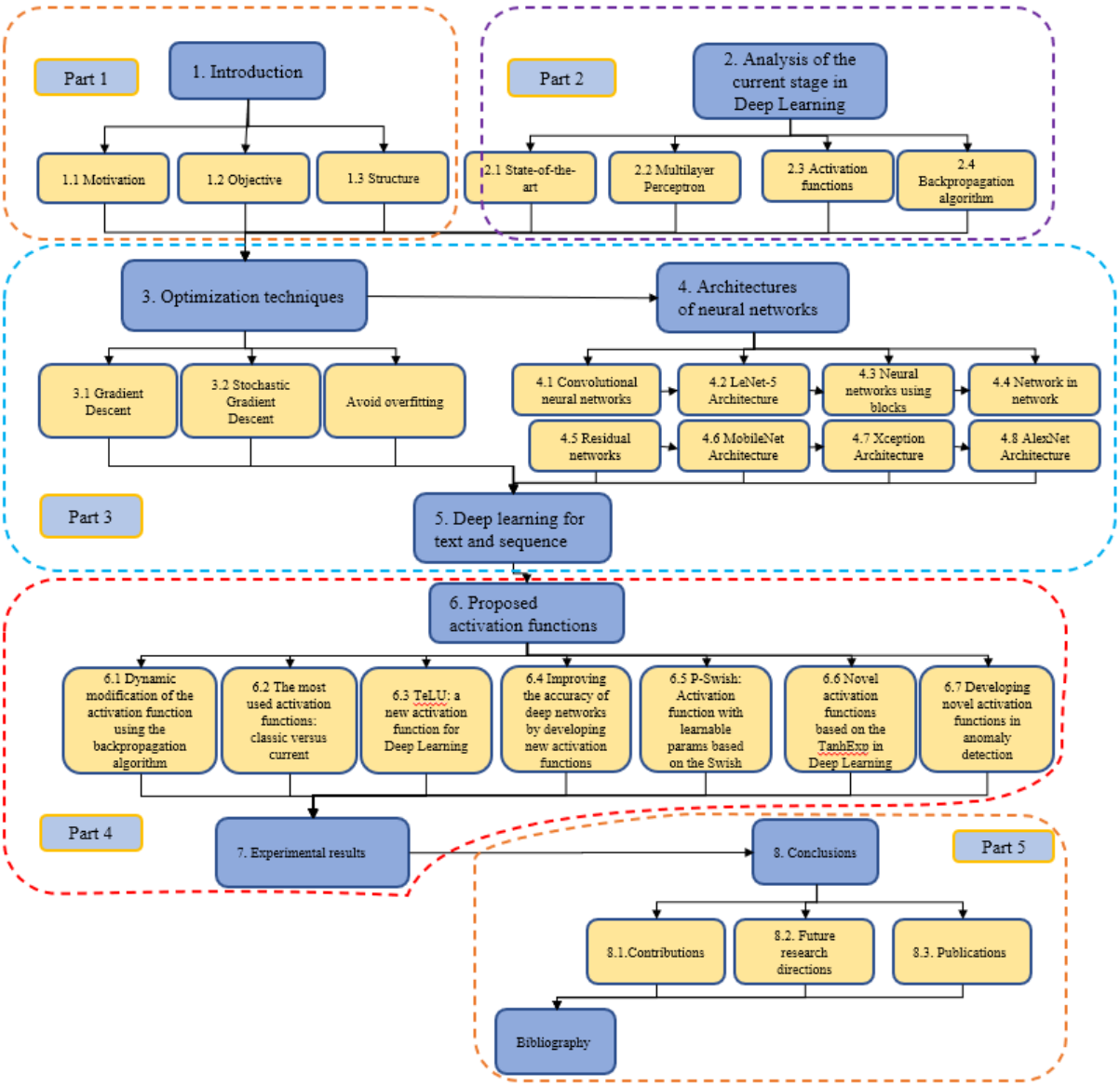

Fig.2. The architecture of the paper by subchapters

The thesis structure:

The first part contains chapter 1 in order to present the motivation, the objective of the research treated in the paper. The second part contains the analysis of the current state in the field of activation functions. The third part consists of chapters 3, 4 and 5 in which are presented the optimization techniques used in the paper, different architectures of neural networks based on which I did experiments and a short presentation of Deep Learning for text and sequences. The fourth part is the main part of the paper, it contains chapters 6 and 7 with proposed activation functions and experimental results. The last part contains chapter 8 on the final conclusions on the issues presented in the paper as well as personal contributions. The paper concludes with future research directions, publications and bibliography.

The aim of the paper is to analyze in detail and develop new activation functions in the field of Deep Learning that are able to increase performance tasks on both Computer Vision [2] and Natural Language Processing fields [3] but also other types of tasks such as detection anomalies [4] and predictions in time series.

The interest for the analysis of Artificial Neural Networks through the use of an

activation function has increased rapidly in recent years, thus becoming a highly studied field of research, as evidenced by existing articles in this direction. That is why in this thesis I want to emphasize the importance of the activation function in the analysis of several datasets for different tasks, to see how the activation function impacts the training process. I will continue to present, very succinctly, the essential aspects that are key points throughout each chapter of the thesis.

The ***first chapter*** presents the motivation of choosing the topic for the thesis, the objective pursued and the structure of the thesis. My motivation starts from the fact that without activation functions, the neural network can only learn basic tasks, so by introducing the activation function the network is able to learn more complex tasks. So, the activation function is a key point in defining the architecture of the neural network and at the same time the activation function is one of the most important parameters that we must choose to successfully obtain a better performance in an artificial neural network. Starting from its role, which it plays in a neural network, my focus was on the study, analysis of how it works, the advantages and disadvantages of activation functions to find the function that best maps to the type of task, bringing the best performance. A decisive property that determines the performance of an activation function is given by whether or not it is smooth. [5] Property also analyzed by me in this thesis in comparison with other functions such as: tangent activation function (*tanh*) [6], ReLU function (*rectified linear unit*) [7], Swish activation function [8], and so on. This property gives to the function the capacity to have continuous derivatives, up to a certain specified order. This implies that the function is continuously differentiable, in other words the first derivative exists everywhere and is continuous. Also, due to the fact that this field is constantly evolving, another direction was the development of novel activation functions that would bring improvements to the architecture of artificial neural networks.

This thesis aims to investigate the use of different activation functions that are an integral part of artificial neural networks in the field of Deep Learning and the development of novel activation functions to bring improvements during network training. This field has been very successful and has been widely studied in recent years due to its availability in terms of hardware, having graphics processing units (*GPUs*) and increasing volumes of data (*Big Data*). To achieve this goal, I have defined the following tasks:

- analysis of the current state in the field and identification of new research directions in the context of activation functions, which have developed over time, being a research area that is constantly evolving and how to correlate the functions of activation with the requirements of the task, the architecture's type but also the data's type.
- identification of activation functions that bring substantial improvements and lead to a rapid convergence of artificial neural networks. This task is based on the analysis of the activation functions and the selection of the most appropriate function based on the data, which also constitutes the exploitation and impact potential.
- implementation of several types of models and their evaluation according to certain specific metrics such as precision, accuracy, cost function, mean absolute error, etc.

In the ***second chapter***, which also corresponds to the ***second part***, I focus on the presentation of the state of the art and the current situation in the field, the Multilayer Perceptron architecture, activation functions and the Backpropagation algorithm. Artificial Intelligence (AI) is the key point in technological development, allowing computers to model the real world to a very high degree, obtaining results comparable to reality. Significant progress has been

made in the field of neural networks - a large and sufficient number to drive my attention in this direction. To achieve this, we need a large amount of information about everything around us, information that must be stored in the computer. Basically, a certain form is given to this data that computers can use to answer questions, providing better answers as the model has a growing volume of information. To generalize this context, many researchers have turned to learning algorithms to store a quantity of information in a short time. Lots of progress has been made in understanding and improving these learning algorithms, but Artificial Intelligence still remains a challenge.

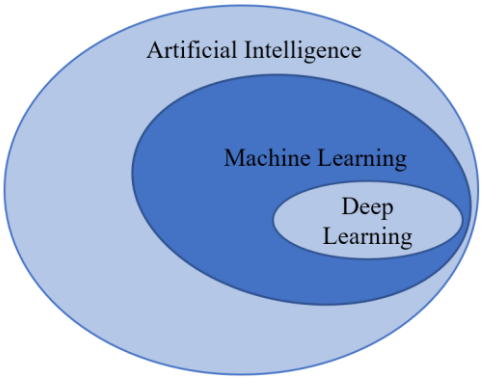

Fig.3. Artificial Intelligence, Automated Learning and Deep Learning - Venn Chart [9]

The history of Deep Learning began in 1943, when a model based on the neural networks of the human brain was created. Since then, Deep Learning has constantly evolved, there were only two significant breaks in its development, encountered in the literature as the ugly winters of Artificial Intelligence. In 1960, the basic elements of a continuous pattern of Backpropagation were defined. [10] Although so many algorithms have been developed over time, activation functions have not been ignored either. Starting from the role of the activation function to filter information, I want to have functions with equally specific properties, as it is based on updating the weights in the neural network. Based on this consideration, several activation functions have been developed that allow us to use them in different ways and that help neural networks achieve convergence faster or give them the ability to use fewer layers in defining their architecture. With fewer layers in the architecture, we have fewer parameters in our network so it is a good way to optimize the network. The main purpose of the activation function is to introduce nonlinearity into the result of a neuron. Also, the activation function known as the *transfer function* can decide whether or not a neuron should be activated by a weighted amount to which bias is added. A neural network without activation functions is just a linear regression model. But the activation function gives the network the ability to learn more complex tasks.

Another important aspect is that the activation function and the determination of the initialization of the weights was given by the space in which the optimization algorithm will take values. This algorithm comes as a solution for creating the neural network which consists of a non-convex optimization problem. Selecting an activation function is an important issue because it can affect how input data changes. It was shown that a novel type of activation function known as the Rectified Linear Unit (ReLU) improves the performance of the neural network in terms of time efficiency and spatial complexity. It was also studied the impact of using nonlinear behavior instead of sigmoid function or tangent function (also known as *tanh*)

by using the controller to prevent possible numerical problems with unlimited activations. Also in this chapter I presented *Perceptron* and *Multi-layer perceptron* architectures.

Because the activation function is a fundamental element in Deep Learning I will mention only the names of the functions that are presented in detail in the thesis. These activation functions are:

- Sigmoid activation function
- Tangent function (tanh)
- Rectified Linear Unit (ReLU) activation function
- Parametric Rectifier (PReLU) activation function
- Leaky ReLU activation function (LReLU)
- Exponential Linear Units (ELU) activation function
- Self-Normalizing Neural Networks (SELU) activation function
- Gaussian Error Linear Unit (GELU)
- Swish activation function
- E-Swish activation function
- EliSH and HardELiSH activation functions
- Flatten-T Swish (FTS) activation function
- Softplus activation function
- Mish activation function
- ARiA2 activation function (Adaptive Richard's Curve weighted Activation)
- SiLU and dSiLU activation functions
- RadBas (RadialBasis), LogSig (Logarithmic-Sigmoid) and TanSig (Tangent-Sigmoid) activation functions
- ElliotSig activation function (Elliot Sigmoid)
- SQNL (Square-Law Non-Linear) activation function
- Inverse square root linear unit (ISRLU) and ISRU activation functions
- Soft Clipping (SC) activation function
- SReLU activation function (S-shaped Rectified Linear Activation Unit)
- BReLU activation function (Bipolar Rectified Linear Activation Unit)
- Concatenated Rectified Linear Unit (CReLU) activation function
- Maxout activation function
- Orthogonal Permutation Linear Unit Activation Functions (OPLU)
- Adaptive Piecewise Linear units (APL) activation function
- Softmax, Large-margin Softmax, Noisy Softmax, SparseMax, Dropmax activation functions
- Softsign activation function
- Activation functions: KAFs, SLAF, Sinusoid, PELU (Parametric Exponential Linear Units), BLU (Bendable Linear Unit), SL-ReLU, DP ReLU and Dual Line, Hard tanh

(Hard Hyperbolic), Hard sigmoid, FReLU (flexible rectified linear unit), Snake.

As a learning strategy, the Backpropagation algorithm has proven to be effective in providing a classification whose accuracy is generally satisfactory. The main disadvantage of this learning technique is that it involves repeated attempts to determine the network architecture, the number of hidden layers and the number of neurons in each hidden layer, in the context where training requires a lot of resources such as memory and running time.

The ***third part*** contains Chapters 3, 4 and 5.

Chapter 3 presents concepts related to optimization techniques such as *Gradient Descent* (GD), *Stochastic Gradient Descent* (SGD), *Overfitting* [11], *Early Stopping* [12], *Regularization*, *Dropout* [13], which are used in the evaluation of my proposed functions in order to optimize the model to combat overfitting.

Chapter 4 contains the fundamental concepts that underlie the understanding and definition of the architectures of deep neural networks, a chapter that presents in detail the types of architectures that I used in Chapter 7 dedicated to experiments. This chapter is also divided into subchapters that briefly describe the architectures used in the thesis. These architectures are:

- Convolutional Neural Network (CNN)
- LeNet-5 architecture
- Neural networks using blocks (Visual Geometry Group - VGG)
- Network in Network (NiN)
- Residual networks (Residual networks - ResNet)
- MobileNet architecture
- Xception architecture
- AlexNet architecture

In ***Chapter 5*** I focus my attention on a *Sentiment Analysis* task, using CNNs for Natural Language Processing task.

The ***fourth part***, which is the most important part of the thesis, contains chapters 6 and 7.

Chapter 6, which is also one of the main chapters of the thesis, describes my predefined and learnable activation functions proposed for deep networks.

As seen in Chapter 2, the choice of activation function has an overwhelming influence on the performance of the neural network. It was also shown that performance is influenced by depth of the network and by the initialization of weights using the uniform Gaussian distribution. In this chapter, I will present several proposals for activation functions that bring an improvement of performance in the training of neural networks.

Chapter 6 is structured as follows:

- In *subchapter 6.1. Dynamic modification of the activation function using the back propagation algorithm in artificial neural networks*, I propose the dynamic modification of the activation function using a known learning technique, namely a Backpropagation algorithm (*Backpropagation-BP*). The change consists in the dynamic change of the slope for the sigmoid activation function based on the increase or decrease of the error in a learning epoch. The study was done using the WEKA (*Waikato Environment for*

Knowledge Analysis) platform by adding this function to the Multi-layer Perceptron (MLP) class. This study aims at the dynamic modification of the activation function that changed according to the relative error of the gradient and an aspect to be mentioned is that in the definition of the neural network architecture no hidden layers were used for this study. The activation functions that have been proposed over time have been analyzed in terms of the applicability of the BP algorithm. The main purpose of the activation function is to scale the outputs of neurons in neural networks and to introduce a nonlinear relationship between the input and output of the neuron. On the other hand, the sigmoid function is usually used for hidden layers, because it combines linear, curvilinear, constant behavior and depends on the input value. Sigmoid function has also been shown not to be effective for a single hidden unit, but when multiple hidden units are involved, it becomes more useful. [14] Which is why in the following studies I will use complex architectures, with several hidden layers (*deep neural networks*).

My approach, in this paper, is to dynamically modify the activation function using the BP algorithm to drive an artificial neural network. I propose to modify the equation of sigmoid function with a β parameter that is dynamically modified during training, in other words this parameter is a learnable β parameter (equation 1).

$$f(x) = \frac{1}{1+e^{-\beta x}} \tag{1}$$

- In *subchapter 6.2. The most used activation functions: classic versus current*, through this study I aim to provide an overview of the most used activation functions, classical functions and current functions. When I say classic, I mean the first activation functions, the most popular and used in the past. But because of their disadvantages, other novel activation functions have appeared that I call current. These functions are among the best known activation functions of Artificial Intelligence, Machine Learning and Deep Learning. With each function, I provide a brief description of the activation function, discuss its impact and show where it is applicable, its advantages and disadvantages, and more details for further clarification. These functions cover several problems such as the vanishing gradient, the exploding gradient, when we use GD and so on. These solutions to these problems are one of the most important topics in the area of research and development of Artificial Intelligence. In this subchapter I presented only the functions presented in this study, because the detailed presentation of the advantages and disadvantages of these functions was made in Chapter 2, Subchapter 2.3 at the presentation of the current status of activation functions. Among the classical activation functions I had as an object of study: the sigmoid activation function and the tangent activation function (*tanh*). The two functions are very similar, both require exponential computation, the exploding/vanishing gradient, the difference is given by the interval of the activation output, in the case of the sigmoid it is [0,1], and in the case of the tanh function [-1,1]. Among the current activation functions I had as an object of study: ReLU, PReLU, LReLU, ELU, SELU and Softmax.

- In *subchapter 6.3. TeLU: a new activation function for Deep Learning*, through this study I aim to develop two novel activation functions derived from ReLU, *tanh* and, the

ELU activation function, they are quite similar to the TanhExp function [15], but the main difference consists of using the ELU function [16] instead of the exp argument and also the property of being learnable introduced by setting a learnable parameter α. The equation of the new activation function called TeLU learnable is given by:

$$f(x) = x \cdot \tanh(elu(\alpha \cdot x)) \tag{2}$$

Where α is a learnable parameter, which I initialized to 0.1. The TeLU function is only a particular case of the TeLU learnable function with the parameter set to 1. Where ELU in turn can be written as follows:

$$f(x) = elu(x) = \alpha \cdot e^{x} - 1 \tag{3}$$

Where α> 0 (Equation 3) is a hyperparameter that controls the value at which the function saturates for negative inputs. When designing the TeLU and *TeLU learnable* functions, I was inspired by the TanhExp function and the Mish function. For training, I will consider the architectures ResNet18, LeNet-5, MobileNet, AlexNet, with different depths. From the point of view of the properties of the activation functions, TeLU and TeLU learnable are centered functions in zero, continuous, smooth, non-monotonic and bounded below. TeLU and TeLU learnable being bounded below add value through a strong regularization that drives to an optimization of the network. On the positive side, TeLU is almost equal to a linear transformation, when the input is greater than 1, also valid on the negative side when the input is smaller than -1, the function almost becomes a linear transformation. TeLU and TeLU learnable show a steeper gradient close to zero, which can speed up the updating of network parameters. During the backpropagation, the network updates its parameters, it has no problems of stopping the training.

- In *subchapter 6.4 Improving the accuracy of deep neural networks by developing new activation functions*, I propose four activation functions that bring improvements for different datasets in Computer Vision tasks. These functions are a combination of popular activation functions, such as sigmoid, bipolar sigmoid, rectified linear unit (ReLU), and tangent (*tanh*). By allowing activation functions to be learnable, I get more robust models. To validate these functions, I also compared them with other powerful activation functions to see how my proposals impact performance improvement. I used several datasets, such as the Dogs and Cats, CIFAR-10, CIFAR-100. Among the architectures used I mention ResNet18, ResNet34, I also used Transfer Learning (TL) technique in the case of VGG16 architecture, pre-trained architecture with data augmentation (DA).
    - The TSReLU activation function consists of a combination of ReLU function, *tanh* function and sigmoid function, which I called *TSReLU* (Tangent-Sigmoid-ReLU). So, this function is a combination of two classical functions (sigmoid and *tanh*) and a current function, given by ReLU. Its equation is given by:

$$f(x) = x \cdot tanh\ (sigmoid(x)) \tag{4}$$

    - The *TSReLU learnable* activation function is quite similar to TSReLU, the only difference is given by a parameter α that is learnable. Its equation is given by:

$$f(x) = x \cdot tanh\,(\alpha \cdot sigmoid(x)) \tag{5}$$

Both *TSReLU* and *TSReLU learnable* are functions with smooth curves, which means that their output will also be smooth. This property offers many advantages when optimizing neural networks to achieve convergence towards the minimum loss.

- The *TBSReLU* activation function is given by the combination of the ReLU function, the tangent function and the bipolar sigmoid function, a function called *TBSReLU* (Tanh-Bipolar-Sigmoid-Relu). Its equation is given by:

$$f(x) = \mathrm{x} \cdot \tanh\left(\frac{1-e^{-x}}{1+e^{-x}}\right) \tag{6}$$

- The *TBSReLU learnable* activation function is quite similar to TBSReLU, the only difference is a parameter α that is a learnable parameter. Its equation is given by:

$$f(x) = \mathrm{x} \cdot \tanh\left(\alpha \cdot \frac{1-e^{-x}}{1+e^{-x}}\right) \tag{7}$$

The TBSReLU learnable activation function has the same properties as the nonparametric TBSReLU activation function.

- In *subchapter 6.5. P-Swish: Activation function with learnable parameters based on the Swish activation function in Deep Learning*, I propose a novel activation function called *P-Swish* (*Parametric Swish*), which is able to bring performance improvements on *object classification* tasks using datasets such as CIFAR-10, CIFAR-100, but we will see that I also used datasets for Natural Language Processing (*NLP*). To test this novel function, I used several types of architectures, including LeNet-5, Network in Network (*NiN*) and ResNet34 compared to popular activation functions such as sigmoid, ReLU, Swish and my proposals. The APL (*Adaptive Piecewise Linear units*) activation function developed from the fact that artificial neural networks usually have a fixed, non-linear activation function in each neuron. APL is a novel form of linear activation function, which is learned independently for each neuron, using GD.

  However, while activation functions have been intensively explored to see their impact on learning, the piecewise activation function has been less explored, so my focus is on developing such an activation function. In this subchapter, I will present a novel activation function that I call *Parametric Swish* (*P-Swish*), it is an activation function derived from the Swish activation function. P-Swish is a piecewise activation function, it is a combination of the ReLU function and the Swish function. Its equation is given by:

$$f(x) = \begin{cases} \alpha \cdot x \cdot sigmoid(\delta \cdot x) & x \leq \beta \\ x & x > \beta \end{cases} \tag{8}$$

  This activation function is a function derived from Swish with 3 parameters that can be predefined or learnable α, β and δ when x≤β, and for x> β we have a function that derives from the ReLU function, but with the parameter β learnable, but when β = 0 the function becomes the ReLU function itself. This proposal benefits from the

properties of the functions from which Swish and ReLU derive, P-Swish being continuous, non-monotonic, unbounded above and bounded below. To validate this function I used several architectures, including *Transfer Learning* technique, applied on several variate tasks, including tasks for NLP. The main advantages of the *P-Swish* function are the improved simplicity and accuracy and the fact that it has no problems of vanishing gradient, but offers a good propagation of information during the training in Deep Learning. Regarding this activation function, I have defined several possibilities for representing this function presented in the thesis. I also tested the P-Swish function on a semantic segmentation task using a fully connected U-Net type architecture.

- In *subchapter 6.6 Novel activation functions based on the TanhExp activation function in Deep Learning*, I propose three novel activation functions (called *TanhExp learnable*, *a_TanhExp* and *a_TanhExp learnable*), which are able to bring performance enhancements to classification tasks on datasets such as MNIST, Fashion-MNIST, CIFAR-10, CIFAR-100, but we will see that I also used a dataset to detect anomalies in time series. To test them, I used several types of architectures. I compared them with the ReLU, tangent (*tanh*) and TanhExp activation functions.
  - *TanhExp learnable* activation function: it is a novel activation function that I developed inspired by the TanhExp activation function, it is a continuous, non-monotonic, unbounded above and bounded below. To validate this function we used several architectures that I mapped to different types of tasks. The *TanhExp learnable* activation function is quite similar to TanhExp, but which brings as a novelty its learnable parameter, which is able to bring improvements in accuracy. The *TanhExp learnable* function equation is given by:

$$f(x) = x \cdot \tanh(e^{\alpha \cdot x}) \tag{9}$$

    Where α is a parameter that can be predefined or learnable.
  - The *a_TanhExp* activation function is a parametric activation function inspired by the TanhExp activation function. The parameter *a* controls the concavity of the global minimum of the activation function, when $a = 0$ this function is the TanhExp activation function itself. The variation of a negative scale reduces the concavity and on the positive scale increases the concavity. *a* is introduced to combat "*dead*" gradient scenarios due to the sharp global minimums of the TanhExp activation function. Its equation is defined by:

$$f(x) = x \cdot \tanh(e^{x} + a) \tag{10}$$

  - The *a_TanhExp learnable* activation function is quite similar to the a_TanhExp activation function. In this case, the parameter *a* becomes a parameter that is able to be learnable.

- In *subchapter 6.7 Developing novel activation functions in time series anomaly detection with LSTM autoencoder*, I propose two novel activation functions in time series anomaly detection, which have the ability to reduce cost functions on the validation set. To achieve this goal, I used an LSTM (*Long Short-Term Memory*)

autocoder [17]. The key point of my proposal is given by the parameter that can be learnable. I tested my proposal compared to other popular functions such as ReLU (*Linear Rectifier Unit*), hyperbolic tangent (*tanh*) and TaLu activation function. Also, the novelty of this proposal is to take into account the piecewise behavior of an activation function to increase the performance of a neural network in Deep Learning. My proposals were inspired by the TaLu (*linear tangent unit*) activation function [18] which is a novel activation function based on hyperbolic tangent and ReLU, developed for neural networks and has been shown to perform better than ReLU, LReLU [19] and ELU activation functions. The equation of the TaLu function is given by:

$$f(x) = y_i = \begin{cases} x_i & \text{if } x_i \geq 0 \\ \tanh(x_i) & \text{if } \alpha < x_i < 0 \\ \tanh(\alpha) & \text{if } x_i \leq \alpha \end{cases} \tag{11}$$

Where α is a fixed parameter with negative values. It was tested from -0.50 to -0.01. As we have seen, the TaLu function is a combination of two functions, being a piecewise function. My proposal is to develop novel functions that derive from TaLu, but the main difference is to use the property that gives to the novel functions the ability to be learnable, a property that is missing in the design of the TaLu function.

- *Talu learnable* activation function is an activation function that is quite similar to TaLu, but brings as a novelty its learnable parameter. The equation of the *Talu learnable* activation function is given by:

$$f(x) = Talu \text{ learnable } (x) = y_i = \begin{cases} x_i & \text{if } x_i \geq 0 \\ \tanh(x_i) & \text{if } \alpha < x_i < 0 \\ \tanh(\alpha) & \text{if } x_i \leq \alpha \end{cases} \tag{12}$$

  Where α is a learnable parameter.
- The second proposal is given by the activation function called *P_Talu learnable* which is similar to *Talu learnable*, but this time I introduce two parameters that can be learned. Its equation is defined as follows:

$$f(x) = P\_Talu(x) = y_i = \begin{cases} x_i & \text{if } x_i \geq b \\ \tanh(x_i) & \text{if } a < x_i < b \\ \tanh(a) & \text{if } x_i \leq a \end{cases} \tag{13}$$

  Where *a* and *b* are two learnable parameters. My approach was based on state-of-the-art strategies, which consist of activation functions with a single scalable parameter that is learned during training. [20]

- In *subchapter 6.8 Soft Clipping Mish – a novel activation function*, I propose two compositional novel activation functions: *Soft Clipping Mish* (SC Mish) *şi Soft Clipping Mish learnable* (SCL Mish), tested on a prediction task of the air pollution for multivariate time series. For modelling this problem, I used two architectures: LSTM (*Long Short Term Memory*) and GRU (*Gated Recurrent Unit*), offering a comparative study of results from point of loss function and RMSE (*root mean square error*) metric. Also the key point in this proposal is given by learnable parameter.
  - *Soft Clipping Mish* (SC Mish) is an activation function that derives from Mish, it is given by:

$$f(x) = max(0, x \cdot tanh(softplus(x))) \tag{14}$$

- In addition, starting from this proposal, I also proposed *Soft Clipping Mish learnable* (SCL Mish), which basically derives from *Soft Clipping Mish,* the major difference being the α learnable parameter, offering more flexibility to the network.

$$f(x) = max(0, x \cdot tanh(softplus(\alpha \cdot x))) \quad (15)$$

Where α is a learnable parameter.

The ***last part*** of the paper contains chapter 7, 8 and the bibliography.

In ***Chapter 7*** I present the results and analysis of deep networks with the proposed activation functions on the selected datasets that I analyzed from a theoretical point of view in Chapter 6.

Chapter 8 contains the conclusions of the thesis and future research directions on this topic. The paper concludes with the publications presented at international conferences and published during doctoral study. At the same time, in the thesis are presented **10** works published by the thesis author as first author. Briefly, according to the level of impact, the grouping of published works is as follows:

- 1 paper in indexed journal Web of Science (ISI) - Impact Factor 1.324
- 7 papers in volumes of international scientific events (proceedings) indexed Web of Science (ISI)
- 2 papers in volumes of BDI indexed scientific manifestations (IEEE Xplore) - submitted for integration in Web of Science (ISI)
- 3 papers in the volumes of international scientific events of which 3 will be submitted for integration in the Web of Science (ISI), one of them having Impact Factor 2.71.

The thesis has 243 pages, of which: 205 pages structured in 8 chapters, 12 pages of bibliography and 16 pages dedicated to the annexes. The paper contains 176 figures and 301 bibliographic titles.

My contributions, given by novel activation functions, highlighted comparatively with the existing activation functions are given by:

(i) Adaptation of sigmoid function,

(ii) Most used activation functions: classical versus current,

(iii) TeLU: a new activation function for Deep Learning,

(iv) Improving accuracy of deep neural networks by developing novel activation functions: TSReLU, TSReLU learnable, TBSReLU and TBSReLU learnable,

(v) P-Swish: Activation function with learnable parameters based on the Swish activation function in Deep Learning,

(vi) Novel activation functions based on the TanhExp activation function in Deep Learning: TanhExp learnable, a_TanhExp and a_TanhExp learnable.

(vii) Developing novel activation functions in the detection of anomalies in univariate time series with LSTM autoencoder: Talu learnable and P_Talu learnable.

(viii) Soft Clipping Mish – a novel activation function: with predefined parameter but also with learnable parameter.

Currently, I have *the following papers* accepted and presented at international conferences and international journal in the field of Automation and Computer Science:

1. *Marina Adriana Mercioni*, Ștefan Holban, "*The recognition of the architectural style using Data Mining techniques*", Conferința: 12th IEEE International Symposium on Applied Computational Intelligence and Informatics (SACI), Timisoara, Romania, 17-19 mai 2018, Pg: 331-337 Publicată: 2018, indexată ISI, Scopus.
2. *Marina Adriana Mercioni*, Ștefan Holban, "*Evaluating hierarchical and non-hierarchical grouping for develop a smart system*", Conferința: 13th International Symposium on Electronics and Telecommunications (ISETC), Timisoara, Romania, 8-9 noiembrie 2018, Pg: 114-117 Publicată: 2018, indexată ISI, Scopus.
3. *Marina Adriana Mercioni*, Alexandru Tiron, Ștefan Holban, "*Dynamic Modification of Activation Function using the Backpropagation Algorithm in the Artificial Neural Networks*", Jurnal: International Journal of Advanced Computer Science and Aplications, Volum: 10 Issue: 4 Pg: 51-56 Publicată: Aprilie 2019, indexată ISI.
4. *Marina Adriana Mercioni*, Nina Holban, Vlad Virgiliu Todea, "*Wireless Routers and their impact on the environment*", Global and Regional in Environmental Protection, Conferența GLOREP 2018, 15- 17 Noiembrie 2018, Timisoara, Romania.
5. *Marina Adriana Mercioni*, Ștefan Holban, "*A survey of distance metrics in clustering data mining techniques*", ICGSP '19 Proceedings of the 2019 3rd International Conference on Graphics and Signal Processing, Pages 44-47, Hong Kong, Publicată: 01 – 03 Iunie 2019, indexată Scopus.
6. *Marina Adriana Mercioni*, Ștefan Holban, "*A study on hierarchical clustering and the distance metrics for identifying architectural styles*", 9 th International Conference on Energy and Environment 2019, 17-18 Octombrie 2019, Timisoara Romania.
7. *Marina Adriana Mercioni*, Ștefan Holban, "*The Most Used Activation Functions: Classic Versus Current*", 15th International Conference on Development and Application Systems, Suceava, Romania, May 21-23, 2020.
8. *Marina Adriana Mercioni*, Angel Marcel, Tat, Ștefan Holban, "*Improving the Accuracy of Deep Neural Networks Through Developing New Activation Functions*", 2020 IEEE 16th International Conference on Intelligent Computer Communication and Processing (ICCP 2020), Cluj-Napoca, Romania, September 3-5, 2020.
9. *Marina Adriana Mercioni*, Ștefan Holban, "*Novel Activation Functions Based on TanhExp Activation Function in Deep Learning*", 2020 International Conference on Data Science, Machine Learning and its Applications (ICDML 2020), Sridevi Women's Engineering College (SWEC), New Delhi, India, October 9-10, 2020.
10. *Marina Adriana Mercioni*, Ștefan Holban, "*P-Swish: Activation Function With Learnable Parameters Based on Swish Activation Function in Deep Learning*", International Symposium on Electronics and Telecommunications 2020, Timisoara, Romania, November 05 - 06 2020.
11. *Marina Adriana Mercioni*, Ștefan Holban, "*TeLU: A New Activation Function for Deep Learning*", International Symposium on Electronics and Telecommunications 2020, Timisoara, Romania, November 05 - 06 2020.
12. *Marina Adriana Mercioni*, Ștefan Holban, „*Soft Clipping Mish – A Novel Activation Function for Deep Learning*", The 4th International Conference on Information and Computer Technologies (ICICT 2021), Kahului, Maui Island, Hawaii, United States, March 11-14, 2021.
13. *Marina Adriana Mercioni*, Ștefan Holban, „*Developing Novel Activation Functions in Time Series Anomaly Detection with LSTM Autoencoder*", IEEE 15th International Symposium on Applied Computational Intelligence and Informatics, May 19-21, 2021.

Selective Bibliography

[1] A. Krizhevsky et al., *ImageNet Classification with Deep Convolutional Neural Networks*, 2012.
[2] G. Koch et al., *Siamese Neural Networks for One-shot Image Recognition*, 2015.
[3] A. Turing, *Computing Machinery and Intelligence*, 1950.
[4] Zimek, Arthur; Schubert, Erich, *Outlier Detection*, 2017.
[5] A. Panigrahi et al., *Effect of activation functions on the training of overparametriezed neural nets*, 2020.
[6] Y. LeCun, Y. Bengio, and G. Hinton, *Deep learning*, 2015.
[7] V. Nair et al., *Rectified linear units improve restricted boltzmann machines*, 2010.
[8] P. Ramachandran et al., *Searching for Activation Functions*, 2017.
[9] C. E. Nwankpa et al., *Activation Functions: Comparison of Trends in Practice and Research for Deep Learning*, 2018.
[10] H. J. Kelley, *Gradient theory of optimal flight paths*, 1960.
[11] P. Bühlmann et al., *Statistics for High-Dimensional Data*, 2011.
[12] L. Prechelt et al., *Early Stopping — But When?*, 2012.
[13] G. E. Hinton, *System and method for addressing overfitting in a neural network*, 2016.
[14] K. Hara et al., *Comparison of activation functions in multilayer neural network for pattern classification*, 1994.
[15] Xinyu Liu et al., *TanhExp: A Smooth Activation Function with High Convergence Speed for Lightweight Neural Networks*, 2020.
[16] D. A. Clevert et al., *Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs)*, 2015.
[17] A. Pulver et al., *LSTM with Working Memory*, 2017.
[18] M. Jain, *A New Hyperbolic Tangent Based Activation Function for Neural Networks*, 2018.
[19] K. He et al., *Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification*, 2015.
[20] A. Turner et al., Julian Francis, *Neuroevolution: Evolving heterogeneous artificial neural networks*, 2014.