The Impact of Irrationals on the Range of Arctan Activation Function for Deep Learning Models

Talya Tümer Sivri*, Nergis Pervan Akman**, Ali Berkol***

* Middle East Technical University, Ankara, Türkiye, E-mail: talyatumer@gmail.com

ORCID: 0000-0003-1813-5539

** BITES Defence & Aerospace, BITES-ASELSAN, Ankara, Türkiye, E-mail: nergis.pervan@bites.com.tr

ORCID: 0000-0003-3241-6812

*** Dr., BITES Defence & Aerospace, BITES-ASELSAN, Ankara, Türkiye, E-mail: ali.berkol@bites.com.tr

ORCID: 0000-0002-3056-1226

Received: 21.09.2023 Accepted: 01.02.2025

Abstract- Deep learning has been applied in numerous areas, significantly impacting applications that address real-life challenges. Its success across a wide range of domains is partly attributed to activation functions, which introduce non-linearity into neural networks, enabling them to effectively model complex relationships in data. Activation functions remain a key area of focus for artificial intelligence researchers aiming to enhance neural network performance. This paper comprehensively explains and compares various activation functions, particularly emphasizing the arc tangent and its specific variations. The primary focus is on evaluating the impact of these activation functions in two different contexts: a multiclass classification problem applied to the Reuters Newswire dataset and a time-series prediction problem involving the energy trade value of Türkiye. Experimental results demonstrate that variations of the arc tangent function, leveraging irrational numbers such as π (pi), the golden ratio (ϕ), Euler number (e), and a self-arctan formulation, yield promising outcomes. The findings suggest that different variations perform optimally for specific tasks: arctan ϕ achieves superior results in multiclass classification problems, while arctan e is more effective in time-series prediction challenges.

Keywords: Deep neural networks, Activation functions, Multiclass classification, Time-Series prediction, Reuters data, Energy trade value data

Derin Öğrenme Modelleri için İrrasyonellerin Arctan Aktivasyon Fonksiyonunun Aralığı Üzerindeki Etkisi

Öz- Gerçek hayattaki çözümü zorlu uygulamalarda, derin öğrenme modelleri birçok alanda önemli başarı sergilemiştir. Bu başarının önemli bir kısmını, sinir ağlarındaki doğrusal olmayan yapılar aracılığı ile verideki karmaşık ilişkileri etkili bir şekilde modellemelerini sağlayan aktivasyon fonksiyonlarına dayanmaktadır. Aktivasyon fonksiyonları, sinir ağlarının performansını artırmayı hedefleyen yapay zeka araştırmacıları için hala önemli bir odak alanıdır. Bu makale, özellikle arktanjant ve onun belirli varyasyonlarına vurgu yaparak çeşitli aktivasyon fonksiyonlarını kapsamlı bir şekilde açıklamakta ve karşılaştırmaktadır. Ana odak noktası, bu aktivasyon fonksiyonlarının iki farklı bağlamdaki etkilerinin değerlendirilmesidir: Reuters Newswire veri kümesine uygulanan çok sınıflı sınıflandırma problemi ve Türkiye'nin enerji ticaret değerini içeren bir zaman serisi tahmini problemidir. Deneysel sonuçlar, π (pi), altın oran (ϕ), Euler sayısı (e) gibi irrasyonel sayıları ve yeni formüle edilmiş kendine ark tanjant formülasyonunu kullanan arktanjant fonksiyonu varyasyonlarının dikkate değer sonuçlar verdiğini göstermektedir.

Bulgular, farklı varyasyonların belirli görevler için en iyi performansı sergilediğini öne sürmektedir: arctan ϕ çok sınıflı sınıflandırma problemlerinde üstün sonuçlar elde ederken, arctan e zaman serisi tahmini problemlerinde daha etkili olmaktadır.

Anahtar Kelimeler: Derin sinir ağları, Aktivasyon fonksiyonları, Çok sınıflı sınıflandırma, Zaman serisi tahmini, Reuters verisi, Enerji ticaret değeri verisi

1. Introduction

Deep learning is a powerful and versatile methodology that has a wide range of applications in various fields, such as healthcare [8], natural language processing applications [7], drug discovery [5], autonomous vehicles [6], finance [9], cyber security [2], and many others. It belongs to a category of machine learning algorithms that go beyond traditional methods, such as multilayer neural networks with many hidden units, to learn complex predictive models [20]. When attempting to model complex problems, activation functions are pivotal as they introduce nonlinearity, which is essential for many tasks. Some examples of activation functions include sigmoid, ReLU, and tanh, which have been widely used in deep learning due to their ability to model complex and nonlinear relationships between input and output data [19]. Overall, deep learning has become a promising approach for solving many real-world problems thanks to activation functions.

By the adaptations made to the universal approximation theorem for neural networks, under certain conditions on the activation function, a single hidden layer neural network can approximate any continuous function arbitrarily well [12, 13, 14, 15, 16]. This means that, given enough data and suitable network architecture, a neural network can learn to approximate complex functions and make accurate predictions for a wide range of tasks. However, the effectiveness and performance of a neural network can vary depending on the specific problem and the characteristics of the data, as well as the choice of the activation function [18]. For example, some activation functions may be more effective for certain tasks or data types, while others may perform better for other tasks or data. Therefore, selecting the appropriate activation function is an important aspect of neural network design and can significantly impact the accuracy and efficiency of the model. In our work, we experimented with time series prediction and multiclass classification problems to find out how activation functions range affects the training time and success of models. Different ranges were obtained by dividing irrational numbers like the Euler number, the golden ratio, and pi, specifically applied to the arctan function. As a result of the experiments, we observed remarkable results.

2. Related Work

While developing more optimized algorithms, many researchers are focusing on activation functions. It is a common research problem that develops optimized models with the help of activation functions. There is a wide range of research that presents the efficiency of activation functions on different datasets and problems. Firstly, we proposed variations of the arctan function with irrational numbers such as the golden ratio and pi and the self-arctan function applied to the multiclass classification problem with the Reuters news wire classification dataset [1]. Skoundrianos and Tzafestas (2004) [24] developed a sigmoidal activation function for modeling dynamic, discrete-time systems. Efe (2008) [23] proposed activation functions which are sinc and cosc experimented on eight different datasets.

An activation function was developed by Liew et al. (2016) [22] which sets a boundary from the positive side of the ReLU applied on the MNIST handwritten digit dataset. Sharma (2019) [21] proposed an activation function that is a combination of the sigmoid and ReLU (Rectified Linear Unit) applied to the ionosphere dataset which is a radar system dataset with continuous values. Misra et al. (2020) [17] proposed a non-monotonic self-regularized activation function which is a composition of tanh and softplus functions and a multiplication of input values. It experimented on CIFAR-10, ImageNet-1k, and MS-COCO Object Detection. The recent research made by Jin et al. (2022) [28] focuses on solving the Time-varying Sylvester equation using Zeroing Neural Network based on the Versatile Activation Function (VAF) variations which include exponential convergence characteristics and adjustable parameters. Another recent work proposed a function called Smish by Wang et al. (2022) [29]. It was experimented on the CIFAR-10 dataset with the EfficientNetB3 network, the MNIST dataset with the EfficientNetB5 network, and the SVHN dataset with the EfficientNetB7 network. Furthermore, Shui-Long et al. (2022) [25] developed an activation function, named tanhLU which uses the symmetry feature of tanh and the unbounded feature of ReLU. It experimented on seven different benchmark datasets which are MNIST, CASIA-webface, Penn Treebank Dataset (PTB), LFW, CIFAR-10, CIFAR-100, datasets and neural network architectures like Fully Connected Neural Network (FCNN), Long Short-Term Memory (LSTM), and Convolutional Neural Network (CNN).

In this article, we present a novel approach that focuses on using variations of the arctan function, specifically those that are based on the combination of input and tanh function and the irrational numbers pi, the golden ratio, and the Euler number. Our experiments demonstrate that this approach can significantly improve training time compared to the topic classification problem. Furthermore, experiments show that promised activation functions produced better and closer results for both problems. Overall, our results indicate that this new approach has great potential for improving the performance of machine learning models.

3. Activation Functions

Activation functions are used in artificial neural networks to determine whether a neuron should be activated or not. These functions take the input from the previous layer of the neural network, sum the weighted input, and then transform it into an output that will be passed to the next layer of the network. Activation functions are often referred to as transfer functions in deep learning because they help to transfer the input to the next layer of the network [26]. Put differently, they serve as a mathematical barrier regulating the flow of information between the input and the neuron at hand, allowing the network to decide which information is relevant and should be passed on to the next layer. Overall, the activation function plays a crucial role in the process of deep learning, as it determines which information is important and should be used to make predictions or decisions.

The activation function is a key component of a neural network. Its primary role is to introduce non-linearity into the network [19], which is necessary for the network to be able to learn complex tasks. Without non-linearity, the network would only be able to perform linear transformations and would not be able to tackle more complicated problems. Non-linearity allows the network to make decisions based on conditional relationships, rather than just relying on a linear function to solve all problems. In addition to adding non-linearity, the activation function must also be differentiable [11]. This property is important because it allows the network to use gradient-based optimization algorithms, which are essential for training the network. Differentiability also enables the network to learn by making small adjustments to the weights and biases of the network, rather than having to start from scratch every time. Overall, the activation function plays a critical role in enabling neural networks to learn and make decisions in a way that is flexible and adaptable to different types of tasks. It allows the network to process and analyze complex data, and to find solutions that are numerically close to the original function, even if the exact function is unknown.

There are three types of activation functions used in Deep Neural Networks (DNN): binary step functions [4], linear activation functions, and nonlinear activation functions. First of all, binary step functions are not useful for solving deep learning problems because they only output a single value (either 0 or 1). They also have a zero gradient, which can cause problems when using backpropagation to update the weights in the network. The second is linear activation functions. They are not effective at approximating complex, nonlinear relationships. This is because their derivative is constant, meaning that they will not learn any new information during training. Lastly, nonlinear activation functions, on the other hand, are able to approximate a wide range of complex, nonlinear relationships and can be used for multiclass classification and regression tasks. Some examples of nonlinear activation functions include sigmoid, tanh, and ReLU. Each function has its pros and cons according to the problem it considers. The objective of this study is to compare the arctangent function and its variations with other functions, elucidating their efficacy while manipulating the range of the arctangent function. In the following sections, activation functions and experiments are explained in detail.

3.1. Rectified Linear Unit (ReLU)

Rectified Linear Unit (ReLU) [4] is a widely used activation function in deep learning models. It is defined in Table 1. In other words, the output of the ReLU function is the maximum of 0, and the input value is x. If x is greater than 0, the output is equal to x. If x is less than or equal to 0, the output is 0. The ReLU function has a simple, piecewise linear form and has several desirable properties for use in neural networks. It and its gradient, see Table 1, are computationally efficient because they do not require any expensive operations such as exponentiation or logarithms, and ReLU has a fast convergence rate during training. It also does not saturate for large input values, which can lead to vanishing gradients and slow training. ReLU is often used as an activation function in hidden layers of deep neural networks and has been found to work well in a wide range of applications. However, it can sometimes produce outputs that are "dead" (i.e., the output is always 0) if the network is not properly initialized or trained, or if the input data is not properly normalized. Variations like Leaky ReLU, Exponential Linear Unit (ELU), Scaled Exponential Linear Unit (SELU), parametric ReLU, etc. can be a solution to the vanishing gradient problem depending on the problem.

3.2. Leaky ReLu

Leaky ReLU [4] is a variant of the standard ReLU activation function that has a small, non-zero slope for negative input values. This helps address the issue of "dying ReLUs," which is a problem that can occur when using the standard ReLU function in neural networks. In other words, neurons that have negative weights will be updated unlike ReLU, and it allows a small, non-zero gradient for negative input values, see Table 1. For positive input values, leaky ReLU behaves like the standard ReLU function, mapping the input to itself. For negative input values, leaky ReLU maps the input to a small, non-zero value determined by the constant α, see Table 1. Leaky ReLU has been shown to improve the performance of neural networks in certain tasks and is often used as an alternative to the standard ReLU function. However, it is important to carefully tune the value of the constant a to ensure that the model is not overfitting the training data.

3.3. Sigmoid

The sigmoid function [4] is a mathematical function that maps values from an input space to a range between 0 and 1. It is often used as an activation function in neural networks, particularly in binary classification tasks, where the output of the network is interpreted as a probability. The sigmoid function is defined as in Table 1, where x is the input value and e is the base of the natural logarithm. It has an S-shaped curve, with an output of 0.5 when the input is 0. One of the main advantages of the sigmoid function is that its output is always bounded between 0 and 1, which makes it well-suited for modeling probabilities. However, the sigmoid function can also suffer from some limitations. For example, the output of the sigmoid function is not zero-centered, which can make it difficult to model certain types of relationships. In addition, the derivative of the sigmoid function (see Table 1) becomes very small for large positive or negative input values, which can make it difficult to train deep neural networks using the sigmoid function. Despite these limitations, the sigmoid function remains a popular choice for activation functions in neural networks and is often used in conjunction with other activation functions to achieve good performance on a wide range of tasks.

3.4. Hyperbolic Tangent (tanh)

The hyperbolic tangent (tanh) function [4], shown in Table 1, is a mathematical function that maps values from an input space to a range between -1 and 1. It is often used as an activation function in neural networks, particularly in tasks where the output of the network needs to be interpreted as a continuous value. One of the main advantages of the tanh function is that its output is zero-centered, which makes it well-suited for modeling certain types of relationships. In addition, the derivative of the tanh function, see Table 1, is relatively large for most input values, which can make it easier to train deep neural networks using the tanh function. However, the tanh function can also suffer from some limitations. For example, the output of the tanh function is not bounded, which can make it difficult to model probabilities. In addition, the tanh function can saturate for very large positive or negative input values, which can slow down the training process.

3.5. Swish

Swish [4] is a modified sigmoid function introduced by Google researchers, see Table 1. It has no upper limit but a lower limit. It is smooth and continuously differentiable, which can make it easier to optimize the model using gradient-based methods, see Table 1. Moreover, it captures negative values and prevents significant negative values from influencing the pattern, introducing a sense of sparsity. However, it can saturate and produce very small gradients when the input is very large or very small, which can slow down the training process. It provides sparsity.

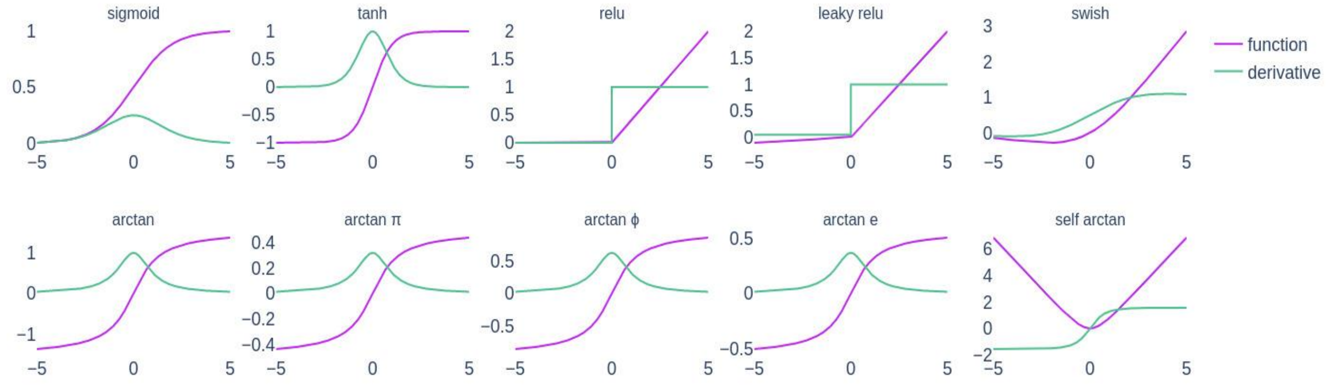


Fig. 1 Illustration of activation functions and derivatives are shown separately.

3.6. Arctanh

The arctan function [3], also known as the inverse tangent function, is a mathematical function that takes a value as input and returns the angle in radians that has a tangent equal to that value, see Table 1. It is also an S-shaped function like the sigmoid; however range of arctan is wider, i.e., $(-\frac{\pi}{2}, \frac{\pi}{2})$. Additionally, a convex error surface is provided by its monotonicity (faster backpropagation).

4. Promised Activation Functions

Activation functions are powerful gates for solving a problem with deep learning models. In this section, we provide the promised activation functions in detail. Specifically, the variations of the arctangent function are explained. These functions use irrational numbers like the golden ratio, the Euler number, and pi. Also, we defined a self-arctan function that uses the input directly in computations. Promised activation functions are designed to show the effect of range of functions manipulation with irrational numbers and the power of symmetry. Variations of arctan functions and their derivatives can be observed in Figure 2(a) and 2(b) respectively.

The variations of the arctan function leverage the unique properties of irrational numbers to adjust the range and behavior of the base arctan function. For example, the division of arctan by π, the golden ratio (ϕ), or Euler number (e) is designed to compress or shift the output range in ways that exploit the symmetry and distribution properties of these constants. This manipulation ensures that the resulting functions maintain smoothness while introducing subtle changes to the gradient flow, which can positively impact training dynamics. Similarly, the self-arctan function adds an innovative approach by directly multiplying the input with the arctan function, leveraging its symmetry around the y-axis to prevent the loss of negative weights—an issue seen in functions like ReLU—and to provide a differentiable, smooth alternative with polynomial-like behavior near the origin. These mathematical properties are not arbitrary but are grounded in the intention to address specific challenges in deep learning, such as vanishing gradients, non-linearity, and weight symmetry. Exploring these derivations and their implications in greater depth helps to clarify the design rationale and solidify the theoretical foundation of the proposed activation functions.

4.1. Arctanπ

Arctan π is a variation of arctan where arctan is divided with π, see Table 1. With purpose is to give a narrower range to arctan using the irrationality of π. So, the range becomes $(-\frac{1}{2}, \frac{1}{2})$. Other properties are the same as the arctan function.

4.2. Arctan Golden Ratio (ϕ)

Arctan with golden ratio is a variation of arctan where arctan is divided with $\frac{1+\sqrt{5}}{2}$, see Table 1. In this activation function, a narrower range is the purpose according to arctan using the irrationality of the golden ratio. Thus, the range becomes $(-\frac{\pi}{1+\sqrt{5}}, \frac{\pi}{1+\sqrt{5}})$. Other properties are the same as the arctan function.

4.3. Arctan Euler

Arctan with Euler number is another variation that we used to experiment with how the Euler number affects the range of arctan to train a neural network architecture, see Table 1 for details of it. Dividing the arctan function with Euler number stacks the range between $(-\frac{\pi}{2e}, \frac{\pi}{2e})$. Other properties are the same as the arctan function.

4.4. Self Arctan

The input itself with arctan is used to obtain symmetry according to the y-axis, see Table 1. Thus, negative and positive outputs will be treated similarly. Also, negative weights will not be lost like ReLU. Furthermore, it is smooth, i.e., differentiable at every point. It acts like a second-degree polynomial around zero.

5. Experiment

As we mentioned in the previous sections, deep learning can be used for solving different problems. While training the models, the effectiveness of activation functions can be different according to the problem and the data. To see the impacts of promised activation functions, we experimented with two different problems which are topic classification and time series prediction. While solving the problem we implemented two different models for each problem. The details are explained in the following sections. For both models, the hyperparameters are chosen using the Keras Tuner library. Specifically, this includes optimizing parameters such as the number of hidden layers, the number of neurons in each hidden layer, dropout probability, and the learning rate.

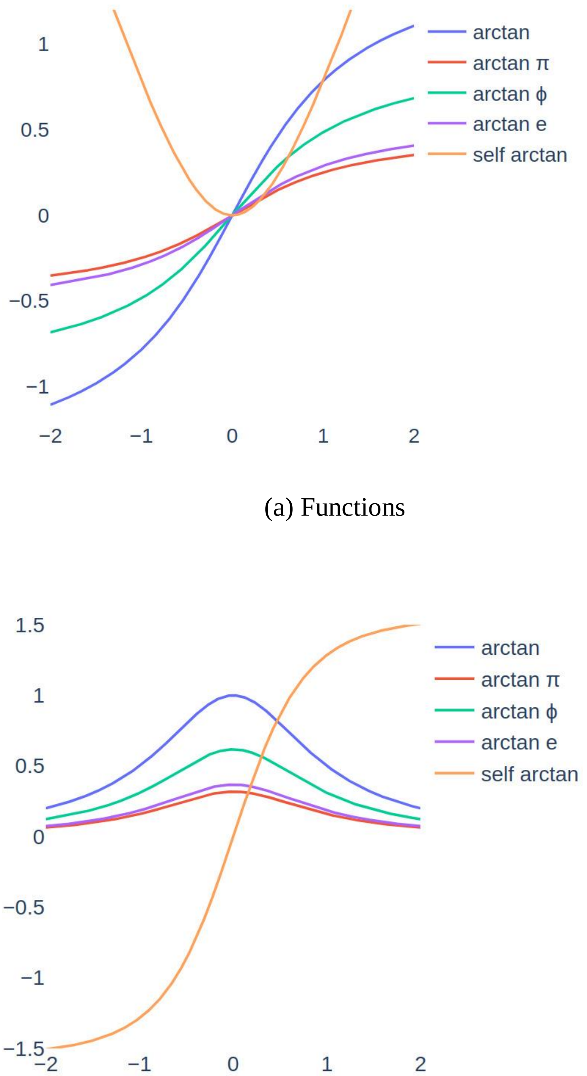
5.1. Topic Classification

Topic classification is a widely used problem in machine learning. Reuters news wire classification dataset was used in the experiments. It is a multiclass dataset with 46 different topics. It contains 11.228 news-wires from Reuters. The dataset was generated by preprocessing and parsing the classic Reuters-21578 dataset. Furthermore, it is an unbalanced dataset, i.e., the class example numbers are not close. For example, there are approximately 4000 samples for one label and approximately 20 samples for more than one label.

Data Preprocessing: Data preprocessing is an essential step in the data analysis and modeling processes as it ensures that the data is in the computable format for the model to analyze. This is particularly important for text data, which can be highly unstructured and contain noise such as emotions, punctuation, and misspellings. To address this, various natural language processing techniques can be applied, such as stemming, tokenization, lemmatization, removal of stop words, punctuation, extra spaces, and letter lowering. The specific techniques used will depend on the nature of the data and the problem being addressed. The Reuters Newswire Classification dataset [10] was used to perform a solution to the topic classification problem. Since the dataset was generated from the Reuters-21578 dataset, tokenization was applied only as a preprocessing step. After that list of sequences was converted to the array.

Models: There are two main components for training high-performance models which are data and hyperparameters. Preparing data for the model is explained in section 5.2. Regarding the performance of the model, fine-tuning hyperparameters is a critical stage for achieving an optimal fit. The selection of hyperparameters to be tuned should be done judiciously, taking into account the specific characteristics of the problem and the data at hand.

In this experiment, two deep learning models were trained for the classification problem for observing the effects of activation functions. The first model (Model 1) is a deep neural network that has two hidden layers with 64 neurons. The second model (Model 2) is also a deep neural network that has one hidden layer with 512 neurons. As the experiments detailed in the paper focus on the efficacy of activation functions, the remaining hyperparameters were kept constant. Moreover, since the problem itself is a multiclass classification, the activation function of the output layer was set to softmax. Some important hyperparameters used in the Model 1 are batch size is 512, and dropout probability is 0.4. In Model 2, the batch size is 32, and the dropout probability is 0.5. Both models have early stopping and also learning rates. The primary distinction between the models lies in the size of the layers and neurons. In Model 1, the neural network is narrower compared to Model 2. Employing early stopping is crucial as it serves to prevent the model from over-fitting.



(a) Functions

(b) Derivatives

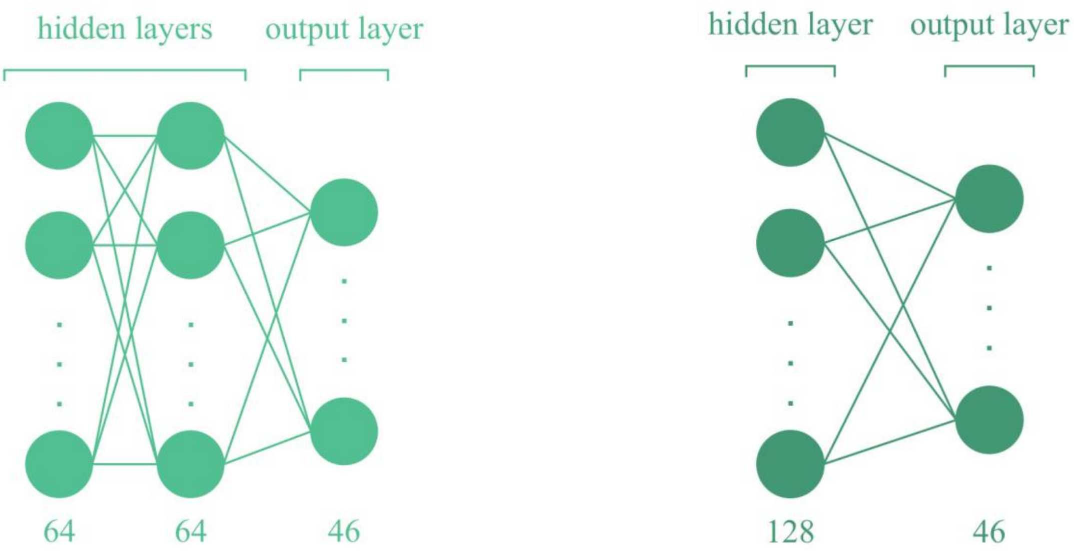
Fig. 2 Graphs of functions and derivatives of arctan and its variations. Figure 2(a) shows the function graphs, and Figure 2(b) shows the function derivatives. These graphs enable us to compare the ranges of variation

Table 1 This table summarizes activation functions with their mathematical definitions, derivatives, and ranges.

Function Name	**Function**	**Derivative**	**Range**
ReLU	$max(0, x)$	$1\ if\ x \geq 0$ $0\ if\ x\ < 0$	$[0, \infty)$
leaky ReLU	$max(\alpha x, x), \alpha\ \in\ (0, 1)$	$1\ if\ x\ \geq 0$ $\alpha\ if\ x\ < 0, \alpha \in\ (0,1)$	$(-\infty, \infty)$
sigmoid	$\frac{1}{1 + e^{-x}}$	$\frac{e^{-x}}{(1 + e^{-x})^2}$	$(0,1)$
tanh	$\frac{(e^x - e^{-x})}{(e^x + e^{-x})}$	$\frac{4e^{2x}}{(e^{2x} + 1)^2}$	$(-1,1)$
swish	$x \cdot sigmoid(x)$	$tanh(x) + x \cdot sech^2(x)$	$(-0.2, \infty)$
arctan	$tan^{-1}(x)$	$\frac{1}{x^2 + 1}$	$(-\frac{\pi}{2}, \frac{\pi}{2})$
arctan π	$\frac{tan^{-1}(x)}{\pi}$	$\frac{1}{\pi(x^2 + 1)}$	$(-\frac{1}{2}, \frac{1}{2})$
arctan ϕ	$\frac{tan^{-1}(x)}{\frac{1 + \sqrt{5}}{2}}$	$\frac{1}{\frac{1 + \sqrt{5}}{2} x(x^2 + 1)}$	$(-\frac{\pi}{1 + \sqrt{5}}, \frac{\pi}{1 + \sqrt{5}})$
arctan e	$\frac{tan^{-1}(x)}{e}$	$\frac{1}{e \cdot (x^2 + 1)}$	$(-\frac{\pi}{2e}, \frac{\pi}{2e})$
self arctan	$x \cdot tan^{-1}(x)$	$\frac{x}{x^2 + 1} + tan^{-1}(x)$	$[0, \infty)$

5.2. Time Series Prediction

Time series prediction is an important problem for researchers and sector employees. Since modeling the time series is useful for future planning, budget planning, etc. In this work, we used day ahead market/trade value [30] in Türkiye, where the data is available at the EXIST Transparency Platform. The dataset has an hourly frequency and its range is starting on October 1st, 2022 00:00, and ending on 12th December 2022 23:00. To sum up, the problem we consider here is the time-dependent univariate series. Before moving on to the model, the dataset needs to be preprocessed for the prediction problem since the features depend on the time and seasonality.



(a) Model 1

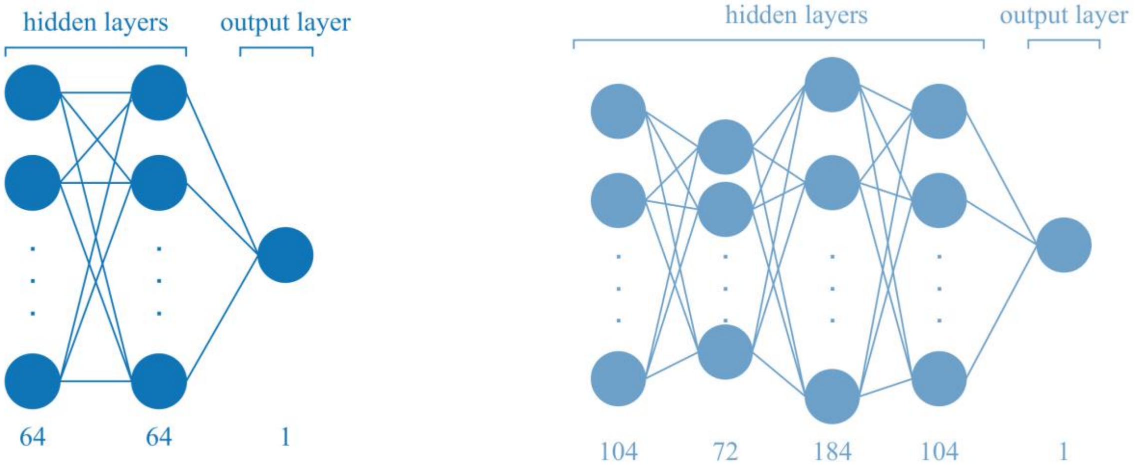
(b) Model 2

Fig. 3 Two different models for topic classification are presented. 3(a) shows the architecture of Model 1 and 3(b) shows the architecture of Model 2.

Data Preprocessing: Univariate time series problems generally need to extract features such as date-time, lags, etc. In our problem, we need to extract serial dependence and time-dependent properties. Serial dependence properties are obtained using the past values of the target variable. PACF and ACF plots are useful for understanding consecutive relationships between variables. It is observable that the data have a 24-hour cyclic pattern at the ACF plot. Also, we observe that the variable of time t+1 is highly correlated with the variable of time t. Moreover, one of the main feature extraction is lagging which is shifting values forward one or more time steps. Lagging is applied 36 times to the trade value to capture and learn hidden patterns. Another main feature extraction is time-dependent variables. Hour, month, year, day, week, and dayofweek independent variables are extracted from corresponding datetime information. Thus, 42 features were obtained. Since all features cannot be a helper for predictive models, feature selection is needed. Boruta, a feature selection library, was used for dimensionality reduction and feature selection. A Random Forest regression model was built with 100 estimators, and depth is 5 for the selection of features. As a result of 100 iterations, 19 features were confirmed, 3 were tentative and 20 were rejected. Features found important by Boruta are as follows: lags are 1, 2, 3, 4, 5, 7, 8, 9, 18, 21, 22, 23, 24, 26, 27, 28, 30, and time-dependent features are hour and day of the week. Finally, the MinMax scaling is applied to the dataset which is the version of the selected features since the problem we consider is a regression problem, i.e., scaling is vital. To sum up, time and serial-dependent features are extracted and selected for a univariate time series prediction.

Models: Time series prediction problems are based on AutoRegressive (AR) models. AR, ARMA (Auto Regressive Moving Average), ARIMA (Auto Regressive Integrated Moving Average), and MA (Moving Average) are some examples. In our case, we modeled the problem using DNN architecture to show that our promising activation functions are applicable to time series prediction problems.

In the previous section, data preprocessing steps were explained in detail. In this section, we will give details of prediction models. We applied two different models. Firstly, the dataset was divided into the train, validation, and test sets. The train set has approximately 55% set of attributes, and validation and test sets have approximately 22% of a set of attributes. The first model is designed as two hidden layers with 0.2 dropout probability and 64 neurons in each and an output layer with one neuron DNN model, see Fig. 4(a). It trained with a 0.01 learning rate. The second model has 4 hidden layers with 104, 72, 184, and 104 neurons and 0.1, 0.2, 0.5, and 0.3 dropout probabilities respectively and one output layer with 1 neuron, see Fig. 4(b). It trained with a 0.005 learning rate. Adam as an optimizer and mean squared error as a loss function were used for both models. Also, early stopping with 5 patience was used in both models.



(a) Model 1

(b) Model 2

Fig. 4 Two different models for trade value prediction are presented. 4(a) shows the architecture of Model 1 and 4(b) shows the architecture of Model 2

6. Results

In this section, we will discuss the results of our experimentation with various activation functions applied to two different problems, including their impact on model training time, model performance, and the trade-offs between them.

First of all, evaluating training time is important for deep learning problems since less training time is preferable in terms of optimized solutions and energy requirements. In the topic classification problem, all promised variations of arctan are faster than sigmoid for both models, see Fig. 5. For Model 1, the best epoch size occurs for self-arctan which is faster than sigmoid, the same with tanh, and slower than the other existing activation functions. On the other hand, promised activation functions, other than self-arctan, are slower than the existing ones. For Model 2, arctan π and arctan e are the slowest after sigmoid. However, arctan ϕ and self-arctan have the same epoch size as ReLU, swish, and arctan. Next, in the time series prediction problem, the slowest one is arctan π for both models. For Model 1, self-arctan is the fastest between promised activation functions and it is faster than leaky ReLU and sigmoid. For Model 2, arctan ϕ is the fastest between promised ones and it is faster than only sigmoid.

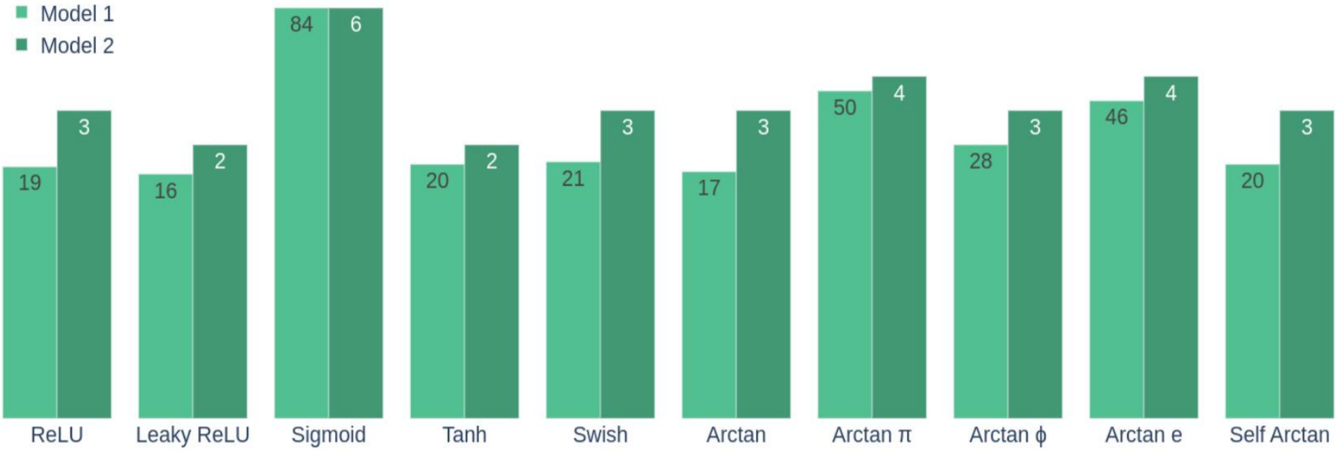


Fig. 5 Epoch sizes of topic classification models are summarized.

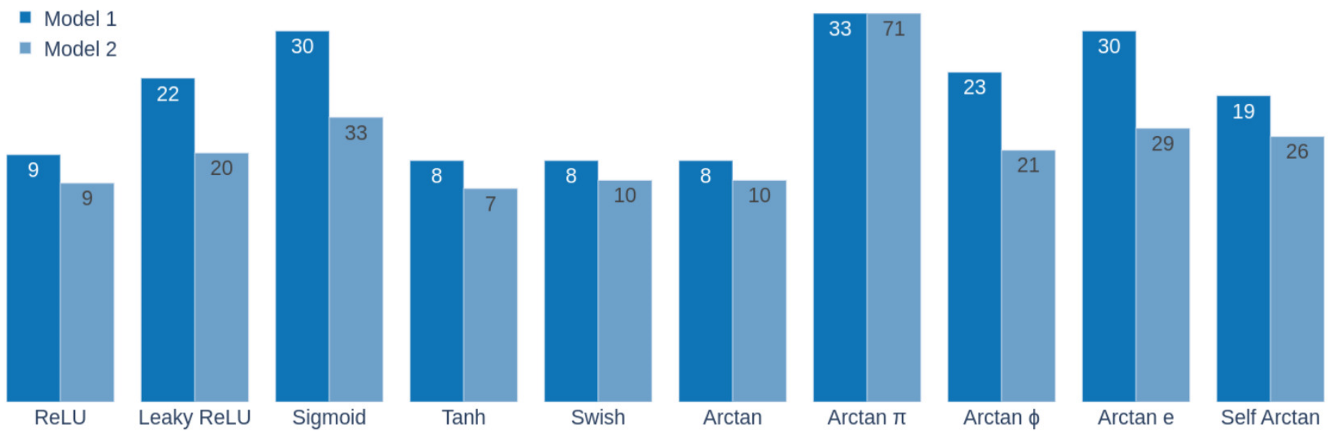


Fig. 6 Epoch sizes of energy trade value models are summarized.

Secondly, model performances are evaluated. For the first problem, topic classification, macro, and weighted average scoring methods were considered. Weighted and macro averages are considered separately since the dataset used in the problem has an unbalanced structure. For both models, promised activation functions to perform that they can compete with widely used activation functions. More specifically, in Table 2 and Table 3 precision, recall, and F1 scores are shown. For Model 1, it can be observed that arctan ϕ is the most competitive among the promised functions. It shares the best score in weighted recall with the evaluation according to two decimal truncation. However, arctan π is the worst among them and it is better than only sigmoid. For Model 2, arctan e has the best precision according to the macro average, arctan ϕ has the best recall, and F1 score according to the macro average, and precision and recall according to the weighted average. Furthermore, promised activation functions are mostly better than sigmoid. As a second problem, we trained the energy trade value and evaluated them according to RMSE, MSE, and R2 score; Table 4 and Table 5. Similarly, with the first problem, we obtained competitive results. For Model 1, arctan e is the most successful among the promising activation functions. Also, it performed better than sigmoid, tanh, ReLU, and arctan for all evaluation metrics except for arctan in MSE, where they are equal at the second decimal. For Model 2, similar results were obtained, which is arctan e is the best between arctan variations.

Table 2 Topic classification Model 1 results are summarized. *: Macro average. ‘: Micro average.

Evaluation Metrics	sigmoid	tanh	relu	leaky relu	swish	arctan	arctan π	arctan ϕ	arctan e	self arctan
Precision*	0.29	0.65	0.41	0.61	0.62	0.67	0.36	0.61	0.48	0.40
Recall*	0.25	0.45	0.32	0.42	0.43	0.45	0.31	0.44	0.38	0.29
F1Score*	0.25	0.50	0.35	0.46	0.47	0.50	0.31	0.48	0.40	0.31
Precision’	0.71	0.78	0.74	0.77	0.78	0.78	0.73	0.77	0.75	0.74
Recall’	0.76	0.79	0.77	0.78	0.79	0.79	0.76	0.79	0.78	0.77
F1Score’	0.72	0.77	0.75	0.77	0.77	0.78	0.74	0.77	0.76	0.77

Table 3 Topic classification Model 2 results are summarized. *: Macro average. ‘: Micro average.

Evaluation Metrics	sigmoid	tanh	relu	leaky relu	swish	arctan	arctan π	arctan ϕ	arctan e	self arctan
Precision*	0.72	0.71	0.71	0.72	0.71	0.69	0.74	0.72	0.75	0.71
Recall*	0.49	0.54	0.54	0.52	0.55	0.55	0.53	0.56	0.53	0.53
F1Score*	0.55	0.58	0.57	0.58	0.59	0.59	0.59	0.60	0.58	0.57
Precision’	0.79	0.80	0.80	0.79	0.80	0.80	0.80	0.81	0.80	0.80
Recall’	0.79	0.80	0.80	0.80	0.81	0.80	0.80	0.81	0.80	0.80
F1Score’	0.78	0.79	0.78	0.79	0.80	0.79	0.79	0.80	0.79	0.79

Table 4 Time series prediction Model 1 results are summarized.

Evaluation Metrics	sigmoid	tanh	relu	leaky relu	swish	arctan	arctan π	arctan ϕ	arctan e	self arctan
RMSE	0.17	0.19	0.18	0.12	0.12	0.17	0.43	0.25	0.14	0.39
MSE	0.03	0.03	0.03	0.01	0.01	0.02	0.19	0.06	0.02	0.15
R2 Score	0.63	0.54	0.61	0.82	0.82	0.66	-1.20	0.22	0.74	-0.76

Table 5 Time series prediction Model 2 results are summarized.

Evaluation Metrics	sigmoid	tanh	relu	leaky relu	swish	arctan	arctan π	arctan ϕ	arctan e	self arctan
RMSE	0.18	0.20	0.23	0.13	0.12	0.17	0.42	0.27	0.14	0.38
MSE	0.03	0.04	0.05	0.01	0.01	0.02	0.18	0.07	0.02	0.15
R2 Score	0.61	0.50	0.34	0.79	0.80	0.65	-1.12	0.14	0.76	-0.74

Overall, we can conclude that:

- At least one of the promised activation functions beat the sigmoid in every case;
- It is significant to gain attention to the range of functions for solving problems with deep learning;
- Arctan ϕ is the best for the multiclass classification problem;
- Arctan e is the best for the time series prediction problem;
- Promising activation functions are shown that they can learn stably with different neuron numbers and model depths.

7. Discussion and Limitations

The field of neural networks has seen many advancements in the last decade, one of which is the introduction of new activation functions. These functions play a crucial role in determining the output of a neuron, and as such, have a significant impact on the performance of a neural network. Many studies have focused on developing activation functions by applying different strategies like a combination of multiple functions. In this work, we mainly focused on a unique strategy which is changing the range of arctan functions with irrationals, pi, the golden ratio, the Euler number, and multiplying input with the arctan function. Furthermore, two different datasets and two different models in each are taken into consideration. One of them is the REUTERS Newswire classification dataset and the other is the Türkiye energy trade market value. Each problem has its challenges in terms of training and data preprocessing steps. Promised activation functions are experimented on wider and narrower DNNs to see how the results are affected by the range of functions. As a result, it is observed that promised activation functions showed stable results. Arctan ϕ shows the most promising results on the topic classification problem and arctan e shows on the time series prediction problem. Additionally, we can mention that although there is a small difference between the ranges of arctan π and e, arctan e is preferable for time series prediction. Also, arctan π which has a wider range according to variations and a narrower range according to arctan itself is more successful for topic classification.

However, this study has several limitations. The first limitation is that the experiments were restricted to only two datasets, limiting the generalizability of the results. The effectiveness of these activation functions in deeper networks or large-scale applications was not explored. Additionally, the underlying mathematical reasons for their success in specific tasks still need to be clarified and require further theoretical analysis. Finally, computational efficiency and convergence rates were not systematically studied, which is important for real-world scalability.

To sum up, these new activation functions have shown promising results and have the potential to improve the performance of neural networks. Further research and experimentation are needed to fully understand their advantages and limitations, as well as to determine their suitability for various tasks.

8. Conclusion and Future Work

In this study, we explored the impact of various activation functions, with a particular emphasis on arctangent variations, in addressing multiclass classification and time-series prediction problems. By adapting irrationals such as pi, the golden ratio, the Euler number, and a self-arctan formulation, we demonstrated the potential of these variations to enhance neural network performance in different contexts. Our experimental results indicate that arctan ϕ is particularly effective for multiclass classification tasks, while arctan e is successful in prediction problem. These findings emphasize the importance of modification of activation functions to the specific requirements of the problem, offering new insights into activation function design and optimization.

For future work, several paths can be taken in advance of this research. Firstly, additional experiments could be conducted across a wide range of datasets and different domains to validate the generalizability of the proposed arctangent variations. Secondly, combining these activation functions with advanced neural network architectures, such as transformers or graph neural networks, could provide deeper insides. Furthermore, the mathematical properties and optimization behaviors of these activation functions can be analyzed to better understand their underlying mechanisms. Finally, exploring new activation functions derived from other irrational numbers or mathematical constants could reveal further promising candidates for improving neural network performance across diverse applications.

Acknowledgements

Thank you for the support to BITES Defence and Information System Department.

References

[1] T.T. Sivri, N.P. Akman, A. Berkol, "A.: Multiclass Classification Using Arctangent Activation Function and Its Variations.", 2022 14th International Conference on Electronics, Computers and Artificial Intelligence (ECAI), Ploiesti, pp. 1-6, 30 June 2022 - 01 July 2022.

[2] T.T. Sivri, N.P. Akman, A. Berkol, C. Peker, "Web Intrusion Detection Using Character Level Machine Learning Approaches with Upsampled Data", Annals of Computer Science and Information Systems, DOI: http://dx.doi.org/10.15439/2022F147, Vol. 32, pp. 269–274.

[3] J. Kamruzzaman, "Arctangent Activation Function to Accelerate Backpropagation Learning", IEICE TRANSACTIONS on Fundamentals of Electronics, Communications and Computer Sciences, Vol. E85-A, pp. 2373–2376, October 2002.

[4] S. Sharma, S. Sharma, A. Athaiya, "Activation Functions In Neural Networks", International Journal of Engineering Applied Sciences and Technology, Vol. 4, pp. 310–316, April 2020.

[5] D. Paul, G. Sanap, S. Shenoy, D. Kalyane, K. Kalia, R.K. Tekade, "Artificial intelligence in drug discovery and development", Drug Discovery Today, DOI: 10.1016/j.drudis.2020.10.010, Vol. 26, No. 1, pp. 80–93.

[6] B. Kisačanin, "Deep Learning for Autonomous Vehicles", 2017 IEEE 47th International Symposium on Multiple-Valued Logic (ISMVL), Novi Sad, pp. 142, 22-24 May 2017.

[7] D.W. Otter; J.R. Medina; J.K. Kalita, "A Survey of the Usages of Deep Learning for Natural Language Processing", IEEE Transactions on Neural Networks and

Learning Systems, DOI: 10.1109/TNNLS.2020.2979670, Vol. 32, No. 2, pp. 604–624.

[8] R. Miotto, F. Wang, S. Wang, X. Jiang, J.T. Dudley, "Deep learning for healthcare: review, opportunities and challenges", Briefings in Bioinformatics, Vol. 19, pp. 1236–1246, November 2018.

[9] S. I. Lee, S. J. Yoo, "Multimodal deep learning for finance: integrating and forecasting international stock markets", The Journal of Supercomputing, DOI: https://doi.org/10.1007/s11227-019-03101-3, Vol. 76, pp. 8294–8312.

[10] Team, K.: Keras Documentation: Reuters Newswire Classification Dataset. Keras. Retrieved December 2022, from https://keras.io/api/datasets/reuters/

[11] T. Kim, T. Adali, "Complex backpropagation neural network using elementary transcendental activation functions", 2001 IEEE International Conference on Acoustics, Speech, and Signal Processing Proceedings, Utah, pp. 1281–1284, 07-11 May 2001.

[12] G. Cybenko, "Approximation by superpositions of a sigmoidal function.", Mathematics of Control, Signals and Systems, DOI: https://doi.org/10.1007/BF02551274, Vol. 2, No. 4, pp. 303–314.

[13] K. Hornik, M. Stinchcombe, H. White, "Multilayer feedforward networks are universal approximators", Neural Networks, Vol. 2, pp. 359–366, 1989.

[14] K.-I. Funahashi, "On the approximate realization of continuous mappings by neural networks", Neural Networks, Vol. 2, pp. 183–192, 1989.

[15] A.R. Barron, "Universal approximation bounds for superpositions of a sigmoidal function", IEEE Transactions on Information Theory, DOI: 10.1109/18.256500, Vol. 39, No. 3, pp. 930–945.

[16] M. Leshno, V.Y. Lin, A. Pinkus, S. Schocken, "Multilayer feedforward networks with a nonpolynomial activation function can approximate any function", Neural Networks, Vol. 6, pp. 861–867, 1993.

[17] D. Misra, "Mish: A Self Regularized Non-Monotonic Activation Function", British Machine Vision Conference, 7–10 September 2020.

[18] R. Parhi, R.D. Nowak, "The Role of Neural Network Activation Functions", IEEE Signal Processing Letters, DOI: 10.1109/LSP.2020.3027517, Vol. 27, pp. 1779–1783.

[19] N. Kulathunga, N. R. Ranasinghe, D. Vrinceanu, Z. Kinsman, L. Huang, Y. Wang, "Effects of the Nonlinearity in Activation Functions on the Performance of Deep Learning Models", CoRR, 2020.

[20] Y. LeCun; B. Boser, J.S. Denker, D. Henderson, R.E. Howard, W. Hubbard, L.D. Jackel, "Backpropagation applied to handwritten zip code recognition", Neural Computation, DOI: 10.1162/neco.1989.1.4.541, Vol. 1, No. 4, pp. 541–551.

[21] O. Sharma, "A new activation function for deep neural network", 2019 International Conference on Machine Learning, Big Data, Cloud and Parallel Computing (COMITCon), Faridabad, pp. 84 – 86, 14 – 16 February 2019.

[22] S.S. Liew, M. Khalil-Hani, R. Bakhteri, "Bounded activation functions for enhanced training stability of deep neural networks on visual pattern recognition problems", Neurocomputing, Vol. 216, pp. 718–734, December 2016.

[23] M.Ö. Efe, "Novel neuronal activation functions for Feedforward Neural Networks", Neural Processing Letters, DOI: https://doi.org/10.1007/s11063-008-9082-0, Vol. 28, No. 2, pp. 63–79.

[24] E.N. Skoundrianos, S.G. Tzafestas, "Modelling and FDI of Dynamic Discrete Time Systems Using a MLP with a New Sigmoidal Activation Function", Journal of Intelligent and Robotic Systems, DOI: https://doi.org/10.1023/B:JINT.0000049175.78893.2f, Vol. 41, No. 1, pp. 19–36.

[25] S.-L. Shen, N. Zhang, A. Zhou, Z.-Y. Yin "Enhancement of neural networks with an alternative activation function tanhLU", Expert Systems with Applications, Vol. 199, pp. 117181, August 2022.

[26] M.F. Augusteijn, T.P. Harrington, "Evolving transfer functions for artificial neural networks", Neural Computing & Applications, DOI: https://doi.org/10.1007/s00521-003-0393-9, Vol. 13, No. 1, pp. 38–46.

[27] J.M. Benitez, J.L. Castro, I. Requena, "Are artificial neural networks black boxes?", IEEE Transactions on Neural Networks, DOI: 10.1109/72.623216, Vol. 8, No. 5, pp. 1156–1164.

[28] J. Jin, J. Zhu, J. Gong, W. Chen, "Novel activation functions-based ZNN models for fixed-time solving dynamic Sylvester equation", Neural Computing and Applications, DOI: https://doi.org/10.1007/s00521-022-06905-2, Vol. 34, No. 17, pp. 14297–14315.

[29] X. Wang, H. Ren, A. Wang, "Smish: A Novel Activation Function for Deep Learning Methods", Electronics, Vol. 11, pp. 540, February 2022.

[30] Trade Value - Day Ahead Market - Electricity Markets| EPIAS Transparency Platform. (n.d.). Retrieved December 20, 2022, from https://seffaflik.epias.com.tr/transparency/piyasalar/gop/islem-hacmi.xhtml