THE EFFECT OF AMPLITUDE MODIFICATION IN S-SHAPED ACTIVATION FUNCTIONS ON NEURAL NETWORK REGRESSION

F. Makhrus[*]

Abstract: Time series forecasting using multilayer feed-forward neural networks (FNN) is potential to give high accuracy. Several factors influence the accuracy. One of them is the choice of activation functions (AFs). There are several AFs commonly used in FNN with their specific characteristics, such as bounded type AFs. They include sigmoid, softsign, arctan, and tanh. This paper investigates the effect of the amplitude in the bounded AFs on the FNNs' accuracy. The theoretical investigations use simplified FNN models: linear equation and linear combination. The results show that the higher amplitudes give higher accuracy than typical amplitudes in softsign, arctan, and tanh AFs. However, in sigmoid AF, the amplitude changes do not influence the accuracy. These theoretical results are supported by experiments using the FNN model for time series prediction of 10 foreign exchanges from different continents compared to the US dollar. Based on the experiments, the optimum amplitude of the AFs should be high, that is greater or equal to 100 times of the maximum input values to the FNN, and the accuracy gains up to 3–10 times.

Key words: *feedforward neural network, activation functions' amplitude, time series prediction, increasing accuracy*

Received: February 1, 2023 **DOI:** 10.14311/NNW.2023.33.015
Revised and accepted: July 26, 2023

1. Introduction

Multilayer feed-forward neural network (FNN) can be applied for constructing data regression models, which are applicable for estimating unknown values or forecasting. The dataset of the regression models can be time series data or other multivariables data. There are several fields which implement it, such as chemistry [18], geography [15], medicine [27], pharmacy [4], and engineering [9]. Although FNN has been used for many years, it is still interesting for many researchers today. The researches focus on accelerating the training process [32] and increasing prediction accuracy.

[*]Faizal Makhrus; Algorithm and Computational Lab, Department of Computer Science and Electronics, Universitas Gadjah Mada, Sekip Utara, Bulaksumur, Sleman, DIY, Indonesia, E-mail: faizal_makhrus@ugm.ac.id

One of the methods to increase the accuracy is by modifying the activation functions (AFs). AFs play a vital role in FNN since AFs produce the outputs of the neuron in FNN. These AFs are able to make FNN to approximate several functions [8]. Therefore, FNN can be used as a regression or classification. However, there is no single AF that is suitable for all problems. There are five categories of AF based on [1], namely bounded, rectified, non-linear below, non-linear and unbounded above, and increasing and decreasing functions. The sigmoid function is a bounded function broadly used in many FNN applications [11, 16, 23, 28]. It is a smooth and continuous function that maps real value $[-\infty, \infty]$ into $[0, 1]$ [7]. Since the output of the sigmoid function is only a non-negative number, there are other alternative bounded functions, such as bipolar sigmoid, tanh, arctan, and softsign functions [30], which can output negative numbers. Those AFs are appropriate in simple FNN architecture (few layers) [3] but may suffer from gradient disappearance in complex FNN architecture such as deep neural network (DNN) during the training phase [25]. The sparsity of the model [19] or inappropriate weights initialization [34] causes this disappearance.

The commonly used AFs in complex FNN architecture, such as convolution neural network (CNN) or DNN, is ReLU [29] and its modifications. ReLU can handle the gradient disappearance problem in sigmoid AF [25], yet it may suffer from death neurons since it maps negative input into zero value. There are several modifications based on ReLU to improve its performance. One of them is leaky ReLU (LReLU) or parametric ReLU (PReLU) [14]. This AF converts negative input into linearly small negative values based on static (LReLU) or dynamic (PReLU) parameters. This AF shows better performance than ReLU. Another modification is the exponential linear unit (ELU) [6] which converts the negative input value of AF into a non-zero value based on the exponential function. The treatment of negative input in ELU gives more robust and faster training than ReLU and LReLU. ELU has a few improvements, such as adding a parameter to scale the ELU function (scaled ELU (SELU)) [13]. Other improvements are approximating the exponential function with floating point operations (fast exponential linear unit (FELU)) [25] and changing the original exponential term in ELU into parametric deformed exponential (parametric deformed ELU (PDELU)) [5]. Apart from ELU, [1] proposed ReLU modification called inverse polynomial linear unit (IpLU) and absolute linear unit (AbsLU). These AFs keep the linearity for positive input and treat negative input using a polynomial function for IpLU and an absolute function for AbsLU. The experiments showed that IpLU and AbsLU performed better than ReLU and LReLU.

A few ReLU modifications combine particular AFs, as done by [20, 24], and [17]. These combinations are purposed to obtain new AFs having advantages from each individual AF and eliminate the limitations. [20] proposed a new AF by combining tangent, ReLU, and sigmoid functions (TSReLU). This AF gives a slight improvement in accuracy compared to each individual function. SinLU AF is proposed by [24], who combines sine and sigmoid linear unit functions. This AF has better accuracy and is more stable to hidden layer addition than ReLU. [17] proposed a combination of ReLU, sigmoid and ELU called RSigELU. It performs better than ReLU and ELU, although it is slower on average for all experiments.

Trainable AFs are another type of AFs introduced by [31], followed by [12], that have become popular in nowadays research [2]. They have parameters that change during the network training. The basic functions of trainable AF can be either the previously explained AFs or any linear/non-linear functions. One of the trainable AF is sigmoid AF, with two added parameters for controlling its slope and amplitude, as done by [22, 31]. These two parameters have specific initialization values and are updated during the network training. These works do not compare the results of the proposed AFs with any existing AFs, yet they look for the best slope and amplitude parameters. Another trainable AF is introduced by [10]. This AF is a polynomial function where its coefficients are obtained using a linear regression. The linear regression is between the sum of the input neurons and the desired outputs. This AF has slightly higher accuracy in classification problems than the similar AFs with fixed parameters. On the other hand, it has promising higher accuracy in regression problems.

From all the AF modifications above, it can be concluded that AF modifications apply to the functions' shape, slope, and amplitude. In this paper, we put interest in the effect of the amplitude modification in bounded AFs regarding the accuracy of the FNN. Since bounded AFs have explicit amplitudes in their functions, the effect of the amplitude modification can be seen clearly. This research investigates four AFs, namely sigmoid, softsign, arctan, and tanh AFs. Although these AFs were founded in the past decades, yet they are still used until today. The characteristics, differences, and advantages of those AFs are explained in [21, 26, 30].

In order to observe the relationship between the AFs' amplitudes and FNNs' accuracy, there are theoretical analyses and some experiments. First, we test that relationship with a simple model, which is a single-variable linear model. This model can be considered as a single neuron. After that, we increase the model into a linear combination of two variables and a bias to know whether that relationship also holds in a more complex model. At last, that relation is tested using the FNN model to see that the relationship is consistent from a simple to a more complex model. However, in FNN model, there are no theoretical analyzes since they are too complex therefore we provide experiment-based analyzes only. The observed FNN model uses a simple architecture with one hidden layer. In addition, the experiments in this research use time series data that are foreign exchanges of 10 countries from different continents to US dollar (USD). The rest of this article is organized as follows. Section 2 provides the analyses of the relationship between AFs' amplitudes and the predictions' accuracy using a single-variable linear model. Section 3 discusses that relationship using a linear combination of two variables and a bias. Section 4 is similar to the previous sections but uses the FNN model. Section 5 provides the discussions, and the last section is the conclusions.

2. Single variable linear function model

In order to know the effect of AFs' amplitude modifications on the FNNs' accuracy, we analyze the model mathematically by simplifying the FNN into one variable linear function, which can be considered as a single neuron with single input and output.

Given a data series y_i where $y_i \in \mathbb{R}^+$ and $i = 1, 2, \ldots, n$, which can be regarded as time series data. This data becomes an input to the linear model with single variable as Eq. (1)

$$\bar{y}_j = a y_{j-1}, \tag{1}$$

with a is a coefficient and $j = 2, 3, \ldots, n$. Then, $\bar{y}_j$ becomes an input to AFs $f(\bar{y}_j)$ called the prediction value. There are three AFs, which are used, namely sigmoid Eq. (2), softsign (3), and arctan (4) AFs, with p is the amplitude parameter

$$f(\bar{y}) = \frac{p}{1 + e^{-\bar{y}}} \qquad \text{(sigmoid)}, \tag{2}$$

$$f(\bar{y}) = \frac{p\bar{y}}{1 + |\bar{y}|} \qquad \text{(softsign)}, \tag{3}$$

$$f(\bar{y}) = p \arctan(\bar{y}) \qquad \text{(arctan)}. \tag{4}$$

We only observe sigmoid, softsign, and arctan AFs for a few reasons. The first reason is that those AFs have different output ranges. The sigmoid AF has an output between 0 and p. On the other hand, the output of softsign and arctan AFs are $[-p, p]$ and $\left[-\frac{p\pi}{2}, \frac{p\pi}{2}\right]$, respectively. The second reason is that those three AFs have good mathematical properties to be derived. Another reason is that tanh AF has a similar shape and range to softsign AF. Therefore, the analysis of tanh AF can be approximated from it.

2.1 Sigmoid AF

This section observes the relation between amplitude p of sigmoid AF and the accuracy of model Eq. (1). The optimum coefficient a of Eq. (1) is obtained by minimizing error function Eq. (5)

$$E = \frac{1}{2} \sum_{j=2}^{n} (f(\bar{y}_j) - y_j)^2. \tag{5}$$

Consider the function $f(\cdot)$ is sigmoid AF as in Eq. (2). Take the derivative of Eq. (5) with respect to a and equal to zero, one will reach

$$\frac{dE}{da} = \sum_{j=2}^{n} (f(\bar{y}_j) - y_j) \frac{df(\bar{y})}{d\bar{y}} \frac{d\bar{y}}{da} = 0. \tag{6}$$

Simplify the Eq. (6) and obtain

$$\begin{aligned} 0 &= \sum_{j=2}^{n} \left(\frac{p}{1 + e^{-a y_{j-1}}} - y_j \right) \frac{p e^{-a y_{j-1}}}{(1 + e^{-a y_{j-1}})^2} \, y_{j-1}, \\ &= \sum_{j=2}^{n} \frac{p^2 y_{j-1} e^{-a y_{j-1}} - y_j y_{j-1} p e^{-a y_{j-1}} (1 + e^{a y_{j-1}})}{(1 + e^{-a y_{j-1}})^3}. \end{aligned}$$

Consider one of solutions from the above equation

$$\begin{aligned} p^2 y_{j-1} e^{-a y_{j-1}} &= y_j y_{j-1} p e^{-a y_{j-1}} (1 + e^{-a y_{j-1}}), \\ p &= y_j + y_j e^{-a y_{j-1}}, \\ \frac{p - y_j}{y_j} &= e^{-a y_{j-1}}, \\ \ln \frac{p - y_j}{y_j} &= \ln e^{-a y_{j-1}}, \\ a &= -\frac{\ln \frac{p-y_j}{y_j}}{y_{j-1}}. \end{aligned}$$

Plug the above a into Eq. (5), and we have this relation

$$E_{\min} = \sum_{j=2}^{n} \left(\frac{p}{1 + e^{\ln \frac{p-y_j}{y_j}}} - y_j \right)^2, \tag{7}$$

where $E_{\min}$ is the minimum error. The summation term can be neglected since this observation focuses on finding the relation between amplitude p and the accuracy of the model represented by $E_{\min}$. Therefore, by assuming $y = 1$, the growth of function $E_{\min}$ based on p is shown in Fig. 1a. From that figure, the minimum error $E_{\min}$ increases to be a constant when p increases, although the increment is relatively small.

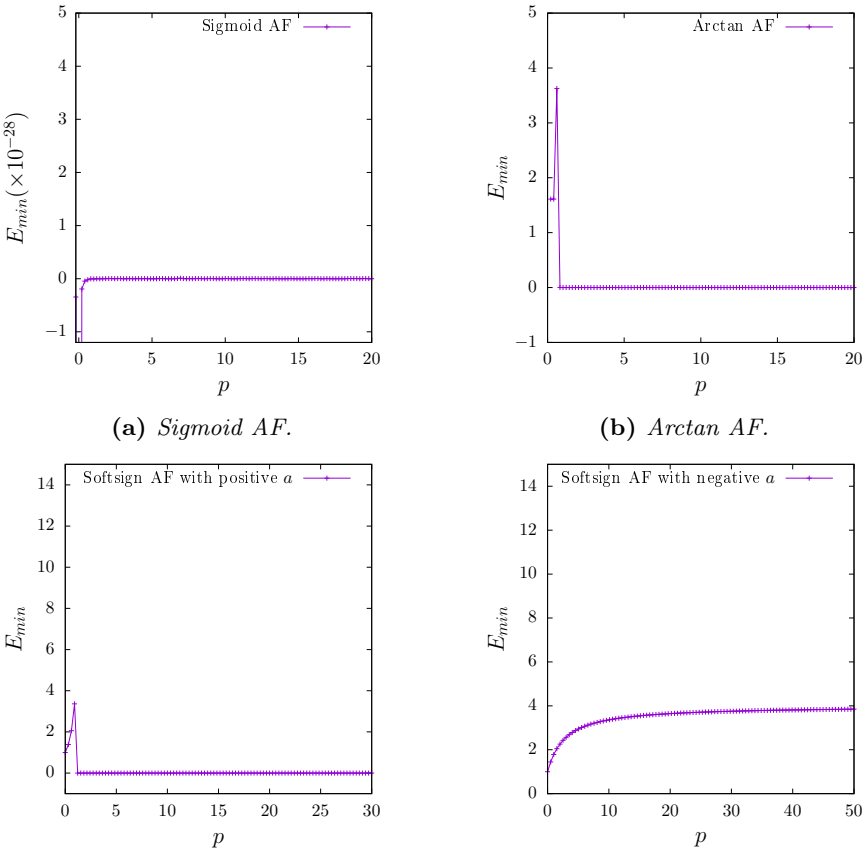
2.2 Softsign AF

Similar to the above method, we use Eq. (1) as the model, and the error function is Eq. (5). Take the derivative of Eq. (5) with respect to a and equal to zero to obtain the optimum a as shown in Eq. (6). Consider the function $f(\cdot)$ is softsign AF as in Eq. (3), then the derivation becomes

$$\begin{aligned} 0 &= \sum_{j=2}^{n} \left(\frac{p a y_{j-1}}{1 + |a y_{j-1}|} - y_j \right) \left(\frac{p y_{j-1}}{(1 + |a y_{j-1}|)^2} \right), \\ &= \sum_{j=2}^{n} \frac{p^2 a y_{j-1}^2 - p y_{j-1} y_j (1 + |a y_{j-1}|)}{(1 + |a y_{j-1}|)^3}, \\ \sum_{j=2}^{n} \frac{p^2 a y_{j-1}^2}{(1 + |a y_{j-1}|)^3} &= \sum_{j=2}^{n} \frac{p y_{j-1} y_j (1 + |a y_{j-1}|)}{(1 + |a y_{j-1}|)^3}. \end{aligned}$$

Take one solution of the above equation, and the following equations hold

$$\begin{aligned} p^2 a y_{j-1}^2 &= p y_j y_{j-1} (1 + |a y_{j-1}|), \\ a &= \begin{cases} \dfrac{y_j}{y_{j-1}(p - y_j)}, & \text{if } a \text{ is non-negative} \\ \dfrac{-y_j}{y_{j-1}(p + y_j)}, & \text{if } a \text{ is negative} \end{cases} \end{aligned}$$



(a) *Sigmoid AF.* **(b)** *Arctan AF.*

(c) *Softsign AF with positive a.* **(d)** *Softsign AF with negative a.*

Fig. 1 *Minimum error of Eq. (5) (E_{min}) with respect to p.*

Put the above a into Eq. (5), and we have these relations

$$E_{\min} = \begin{cases} \sum_{j=2}^{n} \left(\dfrac{\dfrac{py_j}{p - y_j}}{1 + \left| \dfrac{y_j}{p - y_j} \right|} - y_j \right)^2, \text{if } a \text{ is non-negative,} \\ \sum_{j=2}^{n} \left(\dfrac{\dfrac{-py_j}{p + y_j}}{1 + \left| \dfrac{-y_j}{p + y_j} \right|} - y_j \right)^2, \text{if } a \text{ is negative.} \end{cases} \tag{8}$$

The relation between $E_{\min}$ and p can be seen by neglecting the summation and considering $y = 1$ as shown in Fig. 1c and 1d. From those figures, if a is positive,

then $E_{\min}$ slowly decreases and becomes constant when p increases. However, if a is negative then $E_{\min}$ slowly increases until it becomes a constant, as p increases.

2.3 Arctan AF

Similar to the previous method, we use Eq. (1) as the model, and the error function is Eq. (5). Take the derivative of Eq. (5) and equal to zero to obtain the optimum a as shown in Eq. (6). Consider the function $f(\cdot)$ is arctan AF as in Eq. (4), then the derivation becomes

$$\begin{aligned} 0 &= \sum_{j=2}^{n} (p \arctan(ay_{j-1}) - y_j) \frac{py_{j-1}}{1 + a^2 y_{j-1}^2}, \\ &= \sum_{j=2}^{n} \frac{p^2 y_{j-1} \arctan(ay_{j-1}) - py_j y_{j-1}}{1 + a^2 y_{j-1}^2}. \end{aligned}$$

Take one of the solutions of the above equation, and we have these equations

$$\begin{aligned} p \arctan(ay_{j-1}) &= y_j, \\ \tan \arctan(ay_{j-1}) &= \tan \frac{y_j}{p}, \\ a &= \frac{1}{y_{j-1}} \tan \frac{y_j}{p}. \end{aligned}$$

Put the above a to the error function Eq. (5) to obtain $E_{\min}$, as shown below

$$E_{\min} = \sum_{j=2}^{n} \left(p \arctan \left(\tan \left(\frac{y}{p} \right) \right) - y_j \right)^2. \tag{9}$$

The relation between $E_{\min}$ and p can be seen in Fig. 1b by neglecting the summation term and assuming $y = 1$. In that figure, $E_{\min}$ is decreasing while p is increasing.

All the figures in Fig. 1, show that the relations between $E_{\min}$ and amplitude p for S-shaped AFs can be summarized into two patterns. The first pattern is that $E_{\min}$ increases and goes to a constant when p increases, as shown in Fig. 1a and 1d. The second pattern is that $E_{\min}$ decreases and goes to a constant when p increases, as shown in Fig. 1b and 1c.

3. Linear combination model

The previous section explains the relations of AFs' amplitude to the accuracy of a single neuron model. This section observes whether those relations also hold in a more complex model. Now, the used model is a linear combination which consists of two variables and an intercept, as shown in Eq. (10). This model can be regarded as a single neuron with two inputs and a bias.

$$\bar{y}_j = ay_{j-1} + by_{j-2} + c. \tag{10}$$

The optimum coefficients of the model Eq. (10) are obtained by minimizing the error function Eq. (5). However, solving a model analytically with higher dimensions is complex. Therefore we minimize the error function Eq. (5) with respect to $\bar{y}$. The below subsections show the minimization steps for sigmoid, softsign, and arctan AFs. In addition, some experiments are performed to support the theoretical results.

3.1 Sigmoid AF

Consider $f(\cdot)$ in Eq. (5) as sigmoid AF as in Eq. (2). Taking the derivative of Eq. (5) with respect to $\bar{y}$ and equalizing it to zero, one will reach

$$\begin{aligned} 0 &= \sum_{j=3}^{n} \left(\frac{p}{1+e^{-\bar{y}_j}} - y_j \right) \left(\frac{p e^{-\bar{y}_j}}{(1+e^{-\bar{y}_j})^2} \right), \\ &= \sum_{j=3}^{n} \frac{p^2 e^{-\bar{y}_j} - p y_j e^{-\bar{y}_j}(1+e^{-\bar{y}_j})}{(1+e^{-\bar{y}_j})^3}, \\ \sum_{j=3}^{n} \frac{p^2 e^{-\bar{y}_j}}{(1+e^{-\bar{y}_j})^3} &= \sum_{j=3}^{n} \frac{p y_j e^{-\bar{y}_j}(1+e^{-\bar{y}_j})}{(1+e^{-\bar{y}_j})^3}. \end{aligned}$$

Take one solution from the above equation. Hence we have this relation

$$\begin{aligned} p^2 e^{-\bar{y}_j} &= p y_j e^{-\bar{y}_j}(1+e^{-\bar{y}_j}), \\ \frac{p}{y_j} - 1 &= e^{-\bar{y}_j}, \\ \bar{y}_j &= -\ln\left(\frac{p}{y_j} - 1 \right). \end{aligned} \tag{11}$$

Plug Eq. (11) into the error function Eq. (5), and then the minimum error $E_{\min}$ is equal to Eq. (7). In other words, the effect of sigmoid AFs' amplitude on the accuracy of the linear combination model is equal to the effect in a single variable linear function.

3.2 Softsign AF

Similar to the above sigmoid AF, we minimize Eq. (5) with respect to $\bar{y}$ where $f(\cdot)$ is softsign AF as in Eq. (3) to obtain

$$\begin{aligned} 0 &= \sum_{j=3}^{n} \left(\frac{p\bar{y}_j}{1+|\bar{y}_j|} - y_j \right) \left(\frac{p}{(1+|\bar{y}_j|)^2} \right), \\ &= \sum_{j=3}^{n} \left(\frac{p^2 \bar{y}_j}{(1+|\bar{y}_j|)^3} \right) \left(\frac{p y_j}{(1+|\bar{y}_j|)^2} \right). \end{aligned}$$

Take one of the solutions of the above equation, then

$$\begin{aligned} p^2\bar{y}_j &= py_j(1+|\bar{y}_j|), \\ \bar{y}_j &= \begin{cases} \dfrac{y_j}{p-y_j}, \text{if } a \text{ is non negative,} \\ \dfrac{-y_j}{p+y_j}, \text{if } a \text{ is negative.} \end{cases} \end{aligned} \tag{12}$$

Plug Eq. (12) into the error function Eq. (5), then the relation between $E_{\min}$ and p are the same as Eq. (8).

3.3 Arctan AF

Take the derivative of Eq. (5) with respect to $\bar{y}$ where $f(\cdot)$ is arctan AF as in Eq. (4) to obtain

$$\begin{aligned} 0 &= \sum_{j=3}^{n} 2(p\arctan(\bar{y}) - y)\frac{p}{1+\bar{y}^2}, \\ &= \sum_{j=3}^{n} \frac{p^2\arctan(\bar{y})}{1+\bar{y}^2} - \frac{py}{1+\bar{y}^2}. \end{aligned}$$

Take one solution of the above equation, and the result is

$$\begin{aligned} p^2\arctan(\bar{y}) &= yp, \\ \tan\arctan(\bar{y}) &= \tan\left(\frac{y}{p}\right), \\ \bar{y} &= \tan\left(\frac{y}{p}\right). \end{aligned} \tag{13}$$

Plug Eq. (13) into the error function Eq. (5) to see the relation between p and $E_{\min}$, and we have Eq. (9). Therefore, a linear combination with two variables and a bias as a model has the same amplitude and minimum error relation as a linear function model.

3.4 Experiments

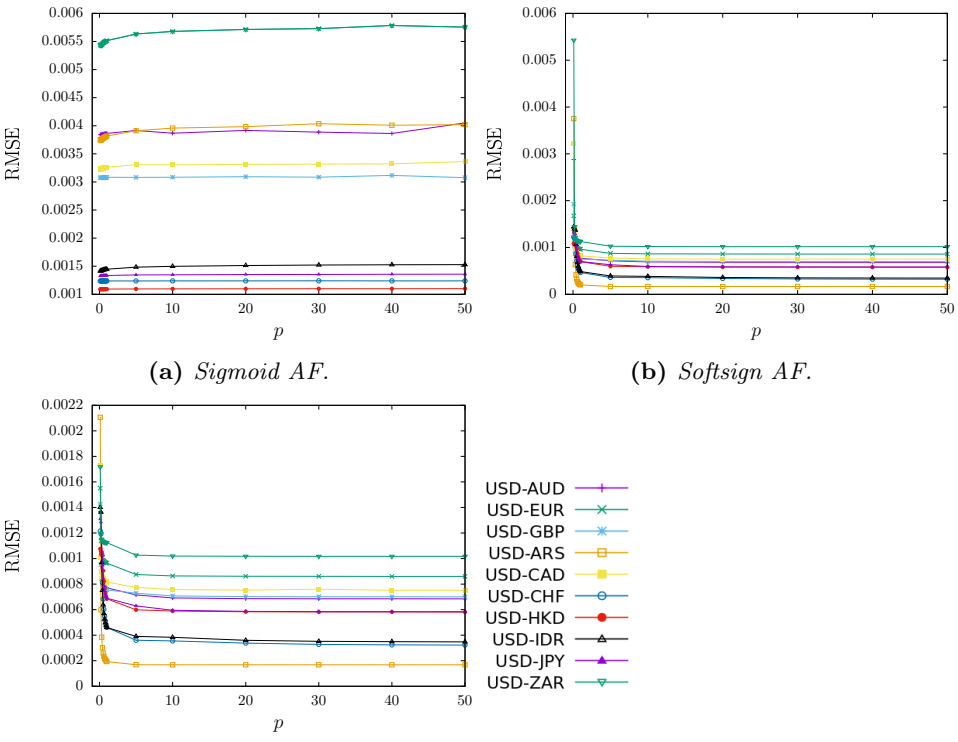
In addition to the above analytic solutions, several experiments are conducted to see whether the analytic solutions hold in real applications. These experiments use Eq. (10) as the model with $a = b = c = 0.1$, Eq. (5) as the objective function, Eqs. (2), (3), and (4) as the AFs, the gradient descent as the minimization method and ten foreign exchanges as the datasets, namely AUD-USD, EUR-USD, GBP-USD, ARS-USD, CAD-USD, CHF-USD, HKD-USD, IDR-USD, JPY-USD, and ZAR-USD. The stopping criterion of the gradient descent method is $||\nabla E||_2 < \epsilon$. The value of ϵ depends on the foreign exchanges since they have different condition of gradient magnitudes when near to the minimum. Therefore, we set ϵ in the range of $[2\times10^{-4}, 6\times10^{-4}]$. The stepsize α (learning rate) varies according to the

p	0.1	1	5	10	20	30	40	50
α	10^{-3}	10^{-4}	10^{-4}	10^{-5}	10^{-5}	10^{-6}	10^{-6}	10^{-7}

Tab. I *Stepsize α for each p.*

amplitude p, as shown in Tab. I. The smaller p uses the bigger α and vice versa, depending on the stability of the searching process.

The foreign exchange datasets are normalized using min-max normalization into values between $[0, 0.1]$. The datasets are obtained from Investing.com, which consist of 200 daily average prices from July 2019 until June 2020. The learning processes of these experiments minimize Eq. (5), and the approximated solutions are used to calculate the root mean of Eq. (5) or root mean square error (RMSE). We choose RMSE rather than Eq. (5) to describe the errors since they are close to the actual values of foreign exchanges. In addition, the relations between RMSE and p are similar to the relations between Eq. (5) and p. The results of these experiments can be seen in Fig. 2. It consists of 3 subfigures showing the relations between RMSE and amplitudes p of sigmoid (Fig. 2a), softsign (Fig. 2b), and arctan AFs (Fig. 2c) in numerical solutions using foreign exchange datasets.



(a) *Sigmoid AF.*

(b) *Softsign AF.*

(c) *Arctan AF.*

Fig. 2 *The RMSE of the linear combination model Eq. (10) with respect to amplitude p using foreign exchange datasets.*

From Fig. 2a, all the datasets have relations between RMSE and the amplitudes p of sigmoid AF similar to the theoretical observation, which are Eq. (7) or Fig. 1a. The relations are that the RMSE increase slightly, while p increase. Fig. 2b shows that the relations between RMSE and amplitudes p of softsign AF for all foreign exchanges follow the theoretical analysis in Eq. (8) or Fig. 1c. It means that the RMSE are decreasing to a constant when p are increasing. Similar to the results of softsign AF, from Fig. 2c, arctan AF also has the RMSE that decrease to constants when p increase as shown in Fig. 1b. Therefore, all the relations between RMSE of the linear combination model and the amplitudes p of sigmoid, softsign and arctan AFs match in both theoretical and experimental.

4. Feed-forward neural network model

This section explains the relations between the amplitudes of AFs and the accuracy of FNN model using some experiments without theoretical analyzes. The theoretical analyzes of FNN are assumed to be similar to the previous simple models. The FNN model consists of an input, a hidden, and an output layers. The input and hidden layers consist of three input neurons, and the output layer has only one neuron. The initial weights of all neurons and bias connections are 0.1. The objective function is Eq. (5), and the optimization process uses the gradient descent method with stopping criterion is $||\nabla E||_2 < \epsilon$ and $2 \times 10^{-4} < \epsilon < 6 \times 10^{-4}$. The stepsizes α (learning rates) for each amplitude p are shown in Tab. I. The datasets are similar to the previous section. They consist of 400 tuples divided into 200 tuples as training data and another 200 tuples as testing data. The datasets are normalized using min-max normalization into $[0, 0.1]$. The AFs in these experiments are sigmoid, softsign, arctan, and tanh AFs. We add tanh AF to see whether it has similar patterns to the other AFs, although there is no theoretical analysis for tanh AF.

The experiments consist of training and testing processes. The first experiments are training processes using 200 training data, where the results can be seen in Fig. 3. The figure consists of four subfigures showing the relation between RMSE and p for each AF. Fig. 3a shows that the RMSE-p relations in sigmoid AF are similar to the theoretical analysis in Fig. 1a, where the RMSE are increasing to constants when p are increasing although the increments are small. These patterns hold in all foreign exchanges.

The other result is Fig. 3b which shows the RMSE and p relations of softsign AF in log-scale. From that figure, it can be seen that all the foreign exchanges have RMSE and p relations which are similar to the theoretical analysis in Fig. 1c, where RMSE decrease to constants when the amplitudes p increase. These patterns in softsign AF are also applied in arctan and tanh AFs, as seen in Figs. 3c and 3d.

The second experiments are testing processes using 200 testing data, where the results can be seen in Fig. 4. The figure consists of four subfigures showing the errors of testing processes for each AF. From all the subfigures, it can be concluded that RMSE-p relations are similar to the relation in training processes Fig. 3. Therefore, the effects of the amplitudes p are not only in the training processes but also in the testing processes with similar patterns.

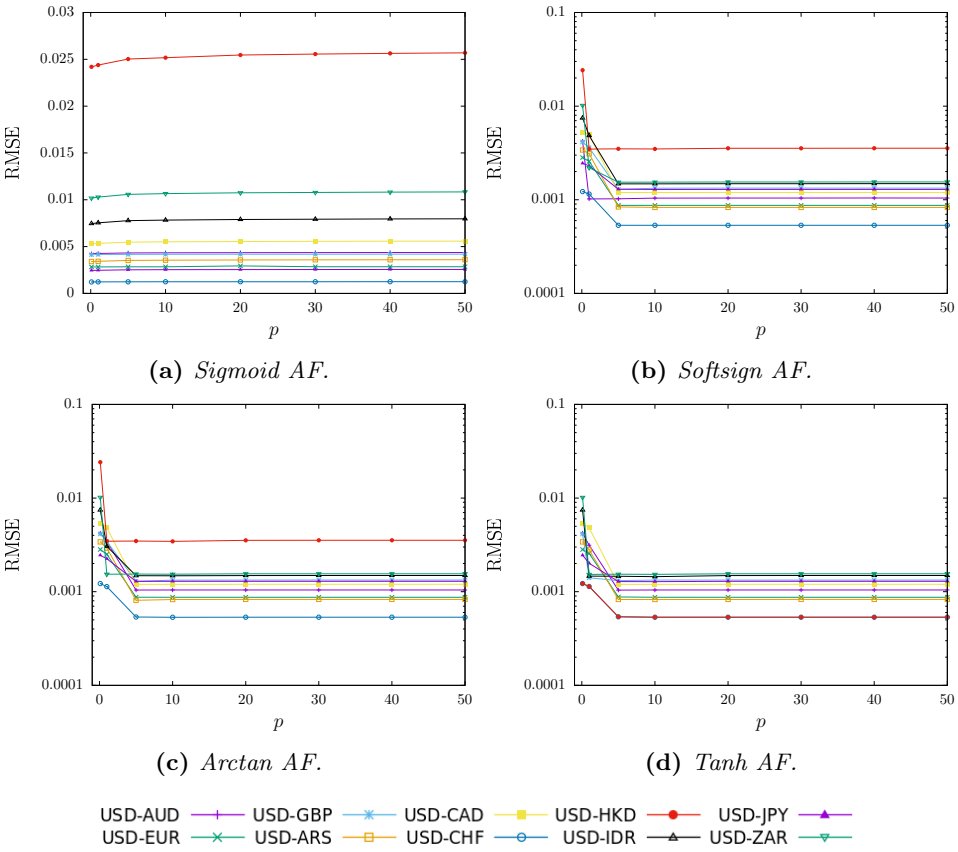


Fig. 3 *The RMSE of FNN model based on amplitudes p in the training processes.*

To see whether the minimization processes of Eq. (5) using the FNN model reach near the exact minimum, we need to see the convergence processes in each epoch given in Fig. 5–7. There are three sample figures showing the convergence processes in each epoch namely USD-AUD, USD-HKD, and USD-CHF datasets. The other datasets have similar convergence rates as those three datasets.

From Fig. 5a, using dataset USD-AUD, it can be seen that the amplitude p in sigmoid AF does not significantly influence the minimum RMSE. The minimum RMSE when $p = (0.1, 1.0, 10, 20)$ are RMSE $= (0.00423, 0.00426, 0.00433, 0.00435)$ respectively. Their differences are around 4–5 decimal places which are insignificant compared to the maximum value in the dataset, which is 0.1. The other results are shown in Fig. 5b, 5c, and 5d for softsign, arctan, and tanh AFs. Those three AFs have similar patterns of RMSE-p relations, where the RMSE are decreasing, when p are increasing. The minimum RMSE of those AFs are 0.00423 when $p = (0.1, 1)$, and 0.001044 when $p = (10, 20)$. In other words, the accuracy of high p is four times better than low p. In addition, the convergence rates of arctan and tanh AFs are similar, yet softsign AF has different convergence speeds.

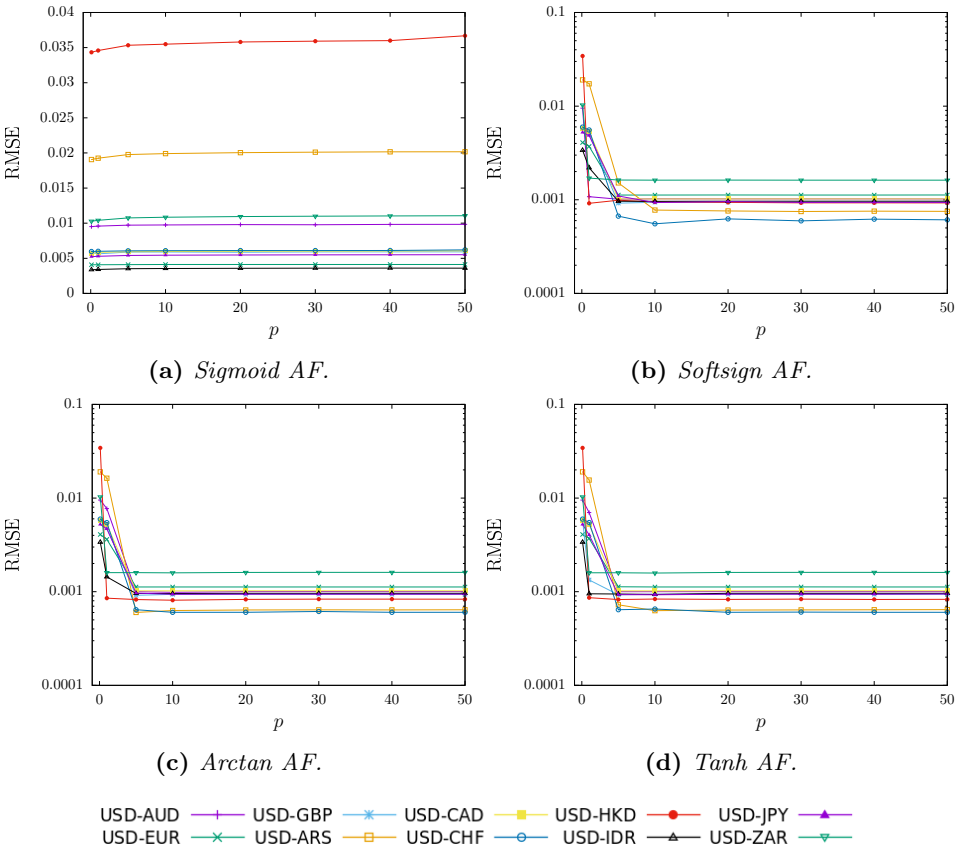
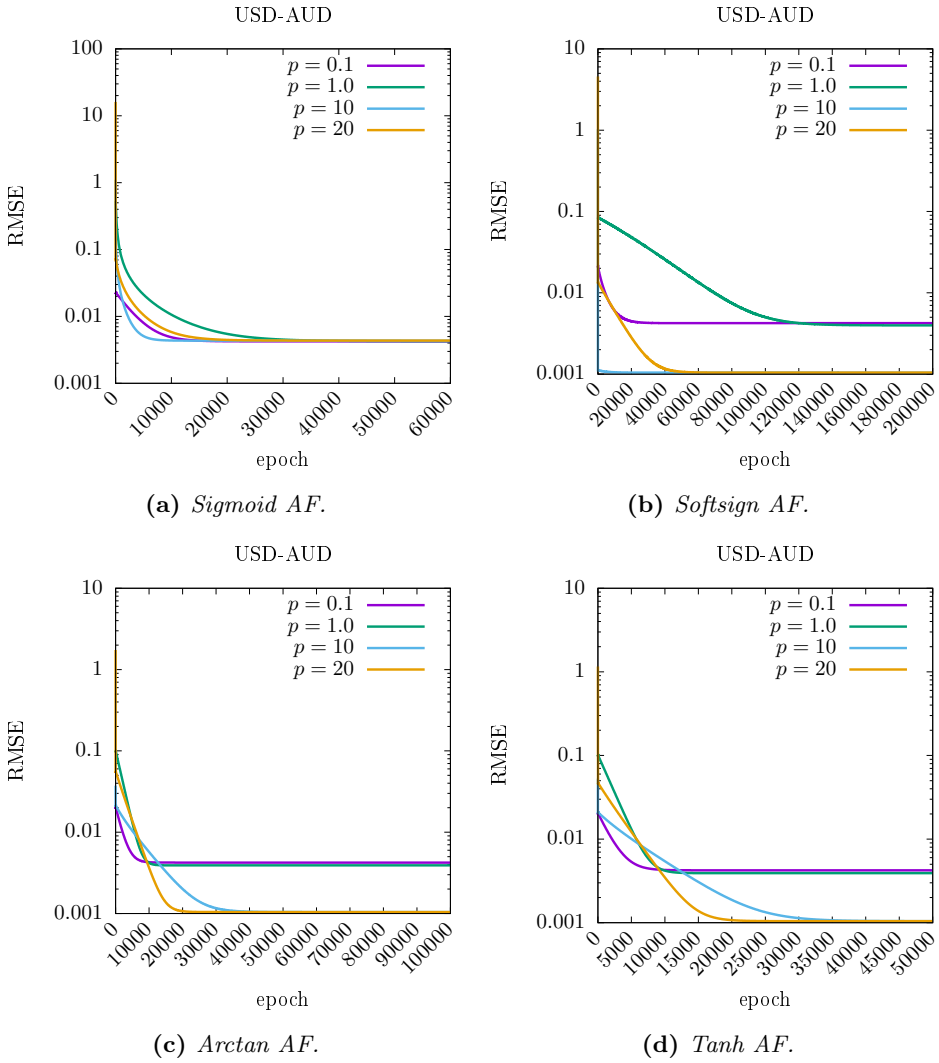


Fig. 4 *The RMSE of FNN model based on amplitudes p in the testing processes.*

Move to USD-HKD dataset. The results can be seen in Fig. 6. Similar to the results of USD-AUD dataset, in USD-HKD dataset, the amplitudes p in sigmoid AF do not influence the accuracy since the minimum RMSE of all p are similar in the range of [0.02441, 0.02522]. These results can be seen in Fig. 6a. On the other hand, the minimum RMSE of softsign, arctan, and tanh AFs are decreasing when p are increasing. These can be seen in Fig. 6b, 6c, and 6d. From those figures, it can be seen that when $p = (0.1, 1)$, the minimum RMSE are in the range of [0.02418, 0.02434] for those three AFs. However, when $p = (10, 20)$, the minimum RMSE are in the range of [0.00354, 0.00355]. Therefore, big p gives 14 times better accuracy than small p. In addition to the accuracy, the convergence rates of those three AFs are similar for the same amplitudes p.

The last example dataset is USD-CHF. The minimization process of this dataset can be seen in Fig. 7. As the results in USD-AUD and USD-HKD, the amplitudes p of sigmoid AF in USD-CHF dataset do not impact the minimum RMSE significantly, as can be seen in Fig. 7a. That minimum RMSE are in [0.00122, 0.00125] for all p. Although the minimum RMSE of each p are similar, convergence rates are different without specific patterns. Contrary to the sigmoid AF, the other three

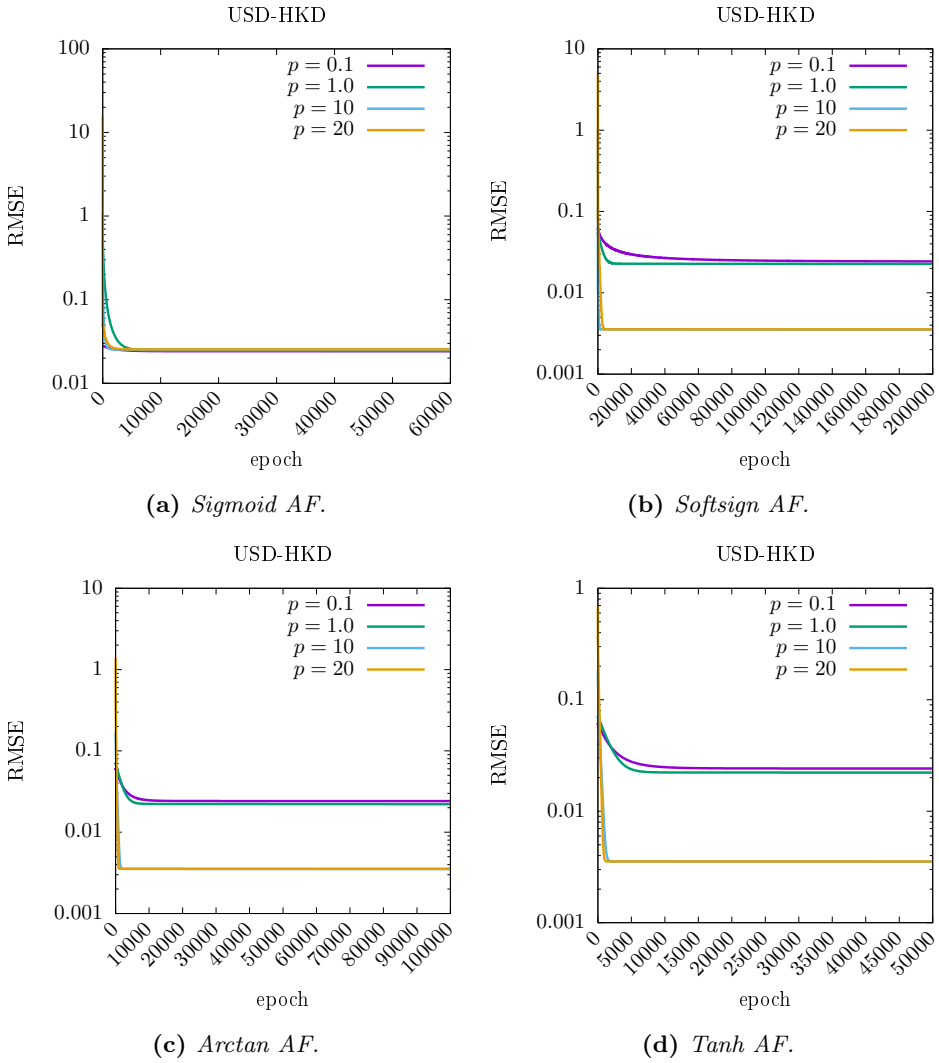


(a) *Sigmoid AF.* **(b)** *Softsign AF.*

(c) *Arctan AF.* **(d)** *Tanh AF.*

Fig. 5 *The convergence of RMSE in each epoch during the training processes using USD-AUD dataset and FNN model.*

AFs, namely softsign, arctan, and tanh AFs, have their minimum RMSE influenced by their amplitudes p, as seen in Fig. 7b, 7c, and 7d. That minimum RMSE are in [0.00113, 0.00115] when $p = (0.1, 1)$ and [0.000534, 0.000535] when $p = (10, 20)$. In other words, the big p has around two times better accuracy than the small p. Furthermore, the convergence rates of those three AFs when $p = (10, 20)$ are almost 15 times slower than when $p = (0.1, 1)$. The learning rate α causes this slow convergence for big p, as shown in Tab. I. The other cause is the characteristic of the USD-CHF dataset, which is more irregular than USD-AUD and USD-HKD.



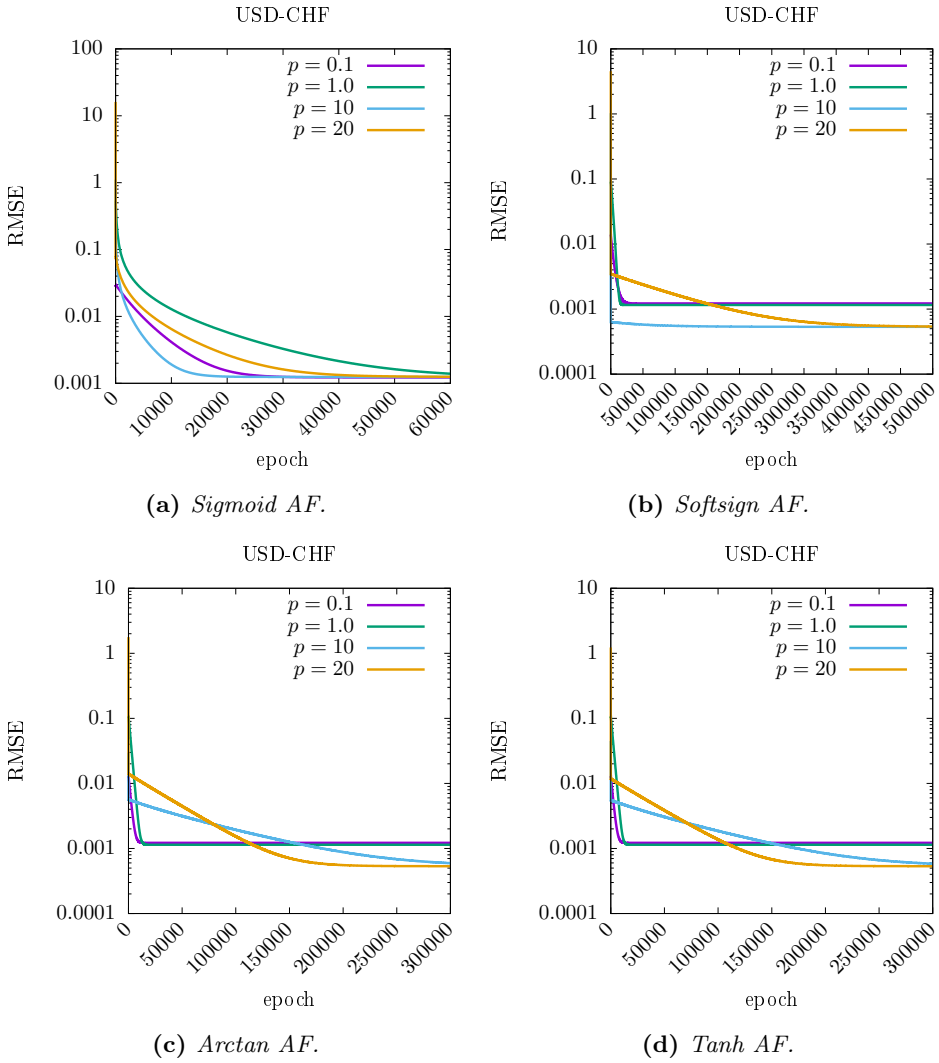
(a) *Sigmoid AF.* **(b)** *Softsign AF.*

(c) *Arctan AF.* **(d)** *Tanh AF.*

Fig. 6 *The convergence of RMSE in each epoch during the training processes using USD-HKD dataset and FNN model.*

5. Discussion

The previous FNN experiments show that the amplitude and forecasting accuracy relationship follows the theoretical analysis of simple models. However, those experiments use an initial weight, which is 0.1. Therefore, observing whether other initial weights have similar relations is necessary. In order to do this, a few experiments are conducted using four initial weights: 0.01, 0.05, 0.2, and 0.3. The FNNs' architecture and the stopping condition are similar to the previous FNN experiments, namely: three layers architecture (3-3-1), $||\nabla E||_2 < \epsilon$ and $2 \times 10^{-4} < \epsilon < 6 \times 10^{-4}$



(a) *Sigmoid AF.* **(b)** *Softsign AF.*

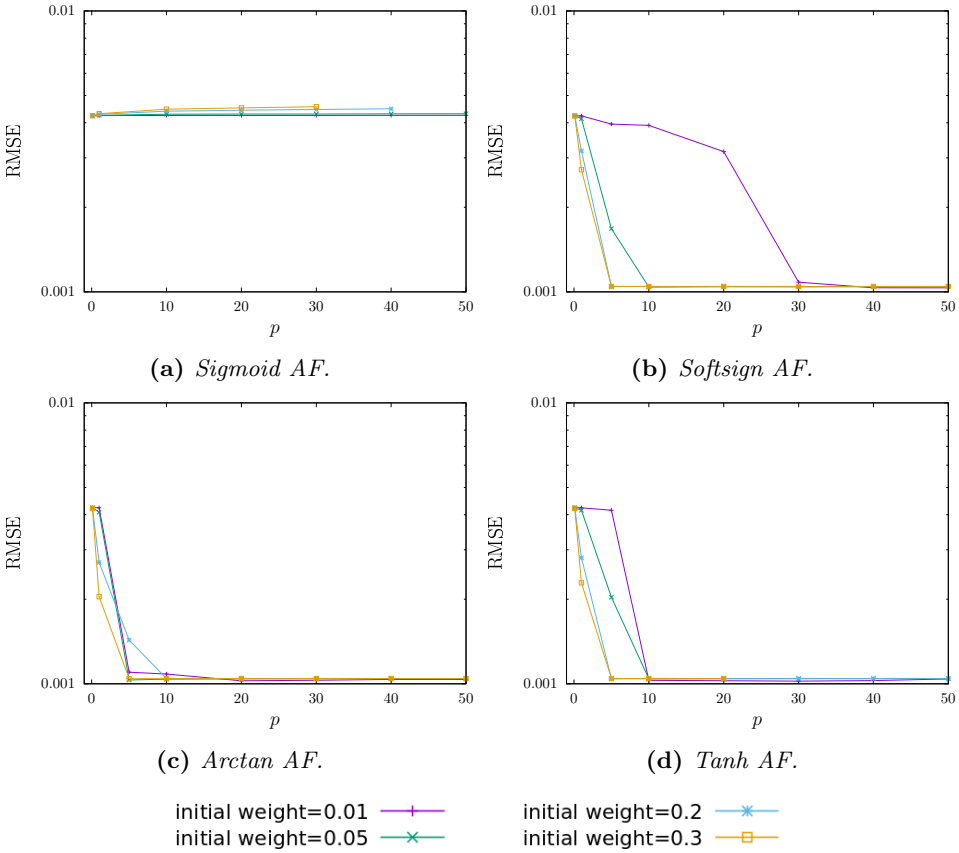
(c) *Arctan AF.* **(d)** *Tanh AF.*

Fig. 7 *The convergence of RMSE in each epoch during the training processes using USD-CHF dataset and FNN model.*

as the stopping criterion, 400 tuples divided into 200 tuples as training data and the remaining tuples as testing data, and normalized dataset using min-max normalization into $[0, 0.1]$.

The results can be seen in Fig. 8. The figure contains four subfigures showing the relation between the accuracy (RMSE) and the amplitude p using different initial weights and four AFs. Fig. 8a shows that the relation between RMSE and p using sigmoid AF and different initial weights is similar to the theoretical analysis presented in Fig. 1a. Meanwhile, when using softsign, arctan, and tanh AFs, those relations are similar to the theoretical analysis depicted in Fig. 1b and 1c

for all initial weights. Therefore, initial weights do not influence the RMSE and p relations. However, a few initial weights fail to reach the minimum of the objective function, e.g, Fig. 8a on initial weight $= 0.3$ and $p = (40, 50)$ and Fig. 8d on initial weight $= 0.3$ and $p = (30, 40, 50)$.



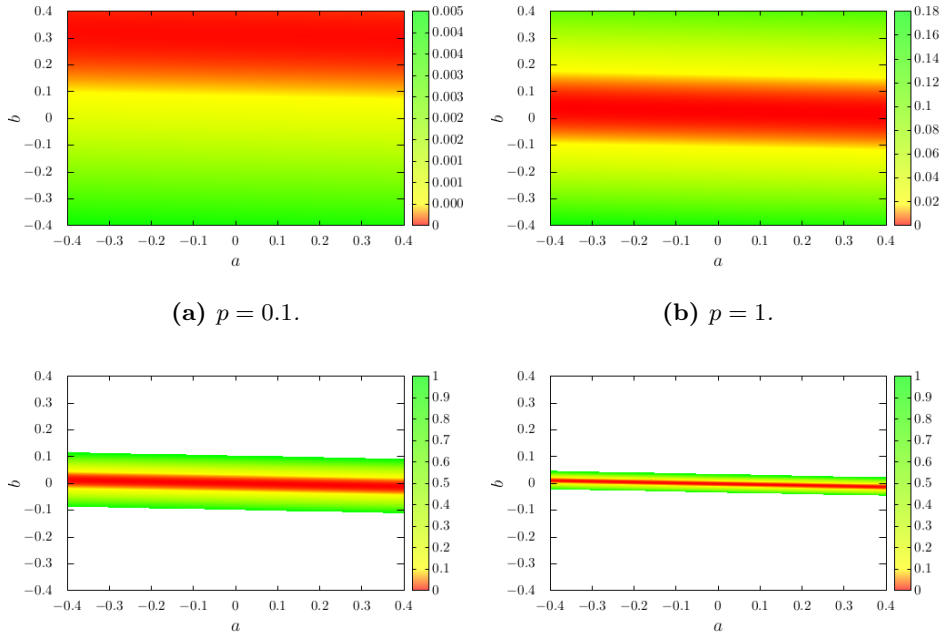
(a) *Sigmoid AF.* **(b)** *Softsign AF.*

(c) *Arctan AF.* **(d)** *Tanh AF.*

Fig. 8 *The RMSE of training FNN model based on amplitudes p and different initial values.*

This happen when amplitude p is high. A high amplitude makes the objective function steep near the minimum, and the minimum region shrinks, as shown in Fig. 9. Therefore, the convergence can fail using an initial weight far from the minimum region. From that figure, each subfigure shows the map of the objective function RMSE using a simple model Eq. (14) and tanh AF. We use a simple model of two variables since it can be represented using a 2D graph, and we choose tanh AF for giving an illustration as the representative of other AFs. The subfigures show that the red areas indicating the objective functions' minimum decrease when p increase. Since the initial weight ≥ 0.3 is far from the minimum region when using $p = 30$ as shown in Fig. 9d, it fails to converge.

$$\bar{y}_j = ay_{j-1} + b \tag{14}$$



(a) $p = 0.1$. **(b)** $p = 1$.

(c) $p = 10$. **(d)** $p = 30$.

Fig. 9 *The map of RMSE using model Eq. (14) and different amplitude p of tanh AF.*

In addition to the initial weight issues, observing the relation between FNNs' accuracy and the amplitude p using different FNN architectures is necessary. There are two architectures to observe for representation: Architecture A and B. A is an architecture with three neurons in the input layer, six neurons in the hidden layer, and one neuron in the output layer (3-6-1). At the same time, B is an architecture with four layers consisting of three input neurons, three neurons in the first hidden layer, two neurons in the second hidden layer, and one neurons in the output layer (3-3-2-1). The training parameters of these experiments are similar to the initial weight experiments above. The results of these experiments are shown in Fig. 10. In that figure, four subfigures describe the RMSE and p relations of four AFs. Each subfigure has training and testing results. Those subfigures show that all the AFs follow the RMSE and p relations as the theoretical analysis Fig. 2. All the AFs increase the accuracy when p increase, except the sigmoid AF. Therefore, based on these experiments, the FNN architectures do not influence the relations between RMSE and p.

Modifying the amplitudes of bounded AFs was introduced by [31], where the outputs of the AF are bounded dynamically. These boundaries increase or decrease based on the derivative of the cost function with respect to the amplitude during the network training. Therefore, these AFs are called trainable AFs. In general, bounded AFs use $[0, 1], [-1, 1]$, or $\left[\frac{\pi}{2}, -\frac{\pi}{2}\right]$ as their outputs, yet in trainable AFs,

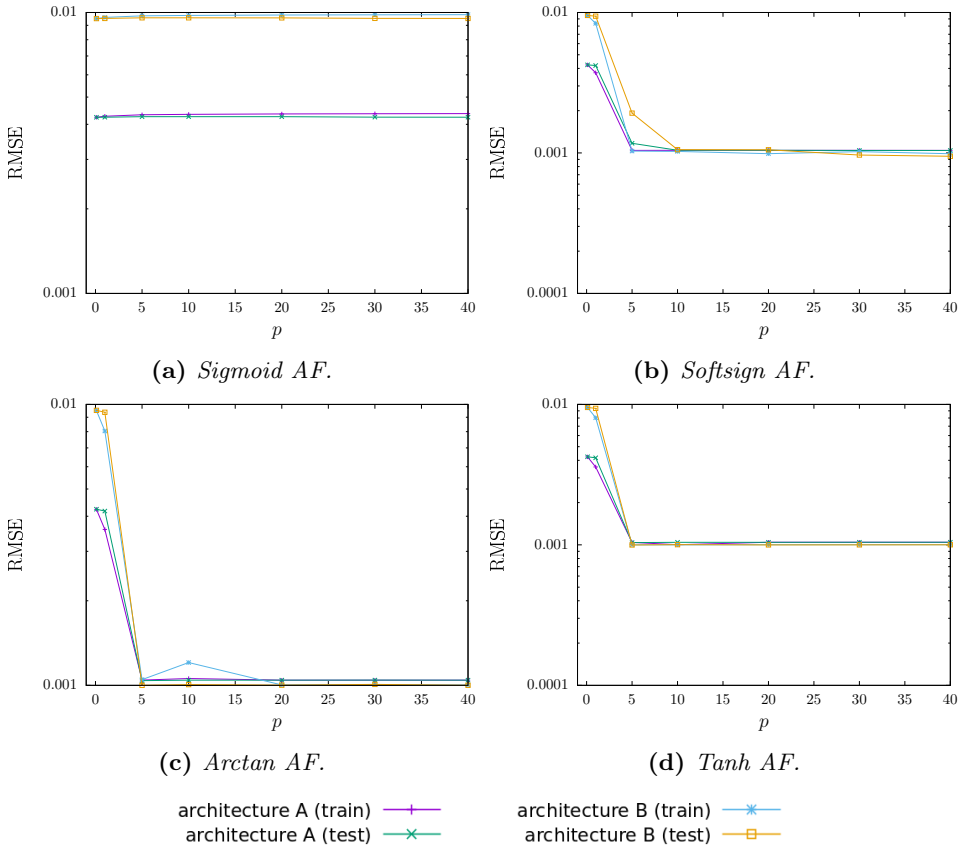


Fig. 10 *The RMSE of training FNN model using architecture A(3-6-1) and B(3-3-2-1).*

their outputs can be larger or smaller than those ranges. This paper observes a modification of the AFs' amplitudes with fixed numbers rather than trainable values. The amplitudes can be greater than the maximum input values of the neural network. Therefore, to compare the proposed modifications' performances, we provide the performance comparisons between trainable AFs and fixed amplitude modification AFs using the FNN model. The FNN model consists of an input layer with three neurons, a hidden layer with three neurons, and an output layer with one neuron. The initial weights of all neurons and bias connections are 0.1.

The FNN using trainable AFs works like regular FNN. However, in the training process, the amplitudes of the AFs are changed in every epoch. These changes are based on the derivative of the cost function Eq. (5) with respect to the amplitudes. On the other hand, the fixed amplitude AF uses $p = 10$. These experiments use the gradient descent method to look for the minimum error of function Eq. (5). The stopping criterion is $||\nabla E||_2$ does not change significantly ($<10^{-10}$) within 40 epochs. The stepsize α (learning rates) is 10^{-5}. The input data to the FNN are normalized using min-max normalization into $[0, 0.1]$. The results of these experiments can be seen in Tab. II and Tab. III.

Dataset	Sigmoid		Softsign		Arctan		Tanh	
	p_{tb}	$p = 10$	p_{tb}	$p = 10$	p_{tb}	$p = 10$	p_{tb}	$p = 10$
AUD-USD	**4.269**	4.338	4.228	**1.043**	4.225	**1.043**	4.225	**1.043**
EUR-USD	**2.838**	2.846	2.825	**0.872**	2.824	**0.872**	2.824	**0.872**
GBP-USD	**4.141**	4.179	4.131	**1.330**	4.131	**1.331**	4.131	**1.332**
USD-ARS	**3.402**	3.549	3.388	**0.829**	3.384	**0.826**	3.384	**0.826**
USD-CAD	**5.293**	5.509	5.274	**1.196**	5.266	**1.200**	5.266	**1.199**
USD-CHF	**1.224**	1.245	1.222	**0.533**	1.222	**0.533**	1.222	**0.534**
USD-HKD	**24.194**	25.283	23.968	**3.522**	23.632	**3.518**	23.632	**3.534**
USD-IDR	**7.459**	7.832	7.423	**1.482**	7.406	**1.488**	7.406	**1.488**
USD-JPY	**2.454**	2.535	2.447	**1.289**	2.444	**1.290**	2.444	**1.292**
USD-ZAR	**10.179**	10.671	10.130	**1.545**	10.105	**1.554**	10.105	**1.555**

Tab. II *The RMSE* ($\times 10^{-3}$) *of FNN model using trainable AF* (p_{tb}) *and fixed amplitude* $p = 10$ *in training processes.*

	Sigmoid		Softsign		Arctan		Tanh	
	p_{tb}	$p = 10$	p_{tb}	$p = 10$	p_{tb}	$p = 10$	p_{tb}	$p = 10$
AUD-USD	**9.598**	9.761	9.504	**1.025**	9.495	**1.002**	9.495	**1.000**
EUR-USD	**4.033**	4.118	4.085	**1.123**	4.084	**1.124**	4.084	**1.124**
GBP-USD	**5.828**	5.900	5.825	**0.943**	5.823	**0.939**	5.823	**0.942**
USD-ARS	**19.069**	19.912	18.980	**0.778**	18.957	**0.631**	18.957	**0.633**
USD-CAD	**5.728**	5.964	5.710	**1.022**	5.701	**1.023**	5.701	**1.023**
USD-CHF	**5.985**	6.086	5.957	**0.573**	5.954	**0.603**	5.954	**0.617**
USD-HKD	**34.337**	35.614	34.046	**0.953**	33.592	**0.822**	33.591	**0.833**
USD-IDR	**3.378**	3.552	3.360	**0.965**	3.352	**0.961**	3.352	**0.961**
USD-JPY	**5.227**	5.441	5.216	**0.933**	5.209	**0.938**	5.209	**0.960**
USD-ZAR	**10.291**	10.861	10.230	**1.622**	10.200	**1.608**	10.200	**1.582**

Tab. III *The RMSE* ($\times 10^{-3}$) *of FNN model using trainable AF* (p_{tb}) *and fixed amplitude* $p = 10$ *in testing processes.*

Tab. II shows RMSE comparisons of the training processes between FNN with trainable AF denoted by p_{tb} and fix amplitude $p = 10$. The first column of the table is the foreign exchanges. The second and third columns show the RMSE of sigmoid AF. The RMSE of p_{tb} are slightly better than $p = 10$, which are 0.25 times on average. The fourth and fifth columns are the results of softsign AF. From that columns, it can be concluded that static amplitude $p = 10$ has better RMSE than p_{tb}, which is seven times on average. The last four columns present the RMSE of arctan and tanh AFs, which have similar results as softsign AF. The arctan and tanh AFs using fixed amplitude $p = 10$ are more accurate seven times than trainable amplitude p_{tb}.

In addition, Tab. III compares RMSE in the testing processes. The results are similar to the training processes where the trainable AF gives slightly better accu-

racy than the fixed AF ($p = 10$) when implemented using sigmoid AF. However, trainable AFs do not perform better than fixed AFs ($p = 10$) when they are applied in softsign, arctan, and tanh AFs.

The results above show that softsign, arctan, and tanh AFs with fixed and high amplitudes are more accurate than sigmoid AF. Those three AFs have positive and negative output values compared to sigmoid AF, which only has positive output values. Therefore, it is interesting to see the performance comparison of those three AFs with linear AF. Linear AF is similar to those three AFs as it has positive and negative output values, but it is unbounded. Some experiments implement this comparison using the same FNN architecture and parameters as previous trainable AF experiments. The results can be seen in Tab. IV. From that table, it can be seen that softsign, arctan, and tanh AFs with amplitude $p = 10$ perform better than linear AF both in training and testing processes.

	Training				Testing			
Dataset	Linear	Softsign $p = 10$	Arctan $p = 10$	Tanh $p = 10$	Linear	Softsign $p = 10$	Arctan $p = 10$	Tanh $p = 10$
USD-AUD	3.906	**1.043**	**1.043**	**1.043**	8.769	**1.025**	**1.002**	**1.000**
USD-EUR	2.615	**0.872**	**0.872**	**0.872**	3.778	**1.123**	**1.124**	**1.124**
USD-GBP	3.826	**1.330**	**1.331**	**1.332**	5.375	**0.943**	**0.939**	**0.942**
USD-ARS	3.130	**0.829**	**0.826**	**0.826**	17.494	**0.778**	**0.631**	**0.633**
USD-CAD	4.876	**1.196**	**1.200**	**1.199**	5.271	**1.022**	**1.023**	**1.023**
USD-CHF	1.138	**0.533**	**0.533**	**0.534**	5.491	**0.573**	**0.603**	**0.617**
USD-HKD	22.070	**3.522**	**3.518**	**3.534**	31.261	**0.953**	**0.822**	**0.833**
USD-IDR	6.850	**1.482**	**1.488**	**1.488**	3.100	**0.965**	**0.961**	**0.961**
USD-JPY	2.293	**1.289**	**1.290**	**1.292**	4.812	**0.933**	**0.938**	**0.960**
USD-ZAR	9.353	**1.545**	**1.554**	**1.555**	9.384	**1.622**	**1.608**	**1.582**

Tab. IV *The RMSE* ($\times 10^{-3}$) *of FNN model using linear, softsign, arctan, and tanh AFs in the training and testing processes.*

All the above experiments use Eq. (5) as the cost function, which is a square error function. However, in FNN, it is common to use the absolute error function Eq. (15) as the cost function [33]. Therefore, in the subsequent experiments, there are some implementations of amplitude AF modification in the FNN model with the absolute error function as the cost function. The architecture and parameters of the FNN model are the same as previous. However, the learning rate is 10^{-8} as the most stable learning rate in this objective function, and the stopping criterion is that $||\nabla E||_2$ does not change significantly ($<10^{-10}$) within 40 epochs or the maximum epoch equals 10^5. The absolute error function is defined as follows

$$E = \sum_j |f(\bar{y}_j) - y_j|, \tag{15}$$

where $f(\bar{y}_j)$ is the output of FNN at time j, and y_j is the real value at time j. The results of these experiments can be seen in Tab. V for the training processes

and VI for the testing processes. Those tables show the MAE of the training and testing processes using softsign, arctan, and tanh AFs. We leave the discussion of sigmoid AF in this experiment since it does not give better accuracy when p increases. Tab. V shows that a higher amplitude $p = 10$ gives higher accuracy than low amplitudes $p = (0.1, 1)$. In addition, the results shown in Tab. VI are similar to Tab. V where the amplitude $p = 10$ has less MAE than low amplitudes $p = (0.1, 1)$. Therefore, the characteristics of the minimum error between two cost functions, namely square error and absolute error in the FNN model, are similar, although we do not observe amplitudes $p > 10$.

Dataset	Softsign			Arctan			Tanh		
	$p = 0.1$	$p = 1$	$p = 10$	$p = 0.1$	$p = 1$	$p = 10$	$p = 0.1$	$p = 1$	$p = 10$
USD-AUD	2.706	2.539	**0.712**	18.160	2.498	**0.717**	18.160	2.498	**0.717**
USD-EUR	2.526	2.192	**1.014**	2.707	2.164	**1.219**	2.707	2.164	**1.219**
USD-GBP	4.061	3.368	**0.940**	3.863	3.320	**0.944**	3.863	3.320	**0.944**
USD-ARS	52.095	2.386	**0.290**	50.492	2.334	**0.293**	50.492	2.334	**0.293**
USD-CAD	49.488	3.112	**0.734**	47.885	3.047	**0.733**	47.886	3.047	**0.733**
USD-CHF	12.647	0.918	**0.494**	11.056	0.904	**0.575**	11.057	0.904	**0.575**
USD-HKD	56.132	21.728	**2.267**	54.527	21.144	**2.262**	54.527	21.145	**2.262**
USD-IDR	65.690	4.204	**0.768**	64.082	4.102	**0.750**	64.083	4.102	**0.750**
USD-JPY	44.400	1.794	**0.835**	42.799	1.760	**0.827**	42.799	1.760	**0.827**
USD-ZAR	62.262	6.076	**1.118**	60.655	5.938	**1.120**	60.656	5.938	**1.120**

Tab. V *The MAE* $(\times 10^{-3})$ *of FNN model using softsign, arctan, and tanh AFs in the training processes.*

Dataset	Softsign			Arctan			Tanh		
	$p = 0.1$	$p = 1$	$p = 10$	$p = 0.1$	$p = 1$	$p = 10$	$p = 0.1$	$p = 1$	$p = 10$
USD-AUD	7.351	6.901	**0.729**	25.969	6.761	**0.719**	25.969	6.761	**0.719**
USD-EUR	3.206	3.199	**1.396**	3.242	3.158	**1.692**	3.242	3.158	**1.692**
USD-GBP	6.380	3.932	**0.750**	5.773	3.875	**0.759**	5.773	3.875	**0.759**
USD-ARS	69.932	17.637	**0.181**	68.323	17.214	**0.160**	68.323	17.214	**0.160**
USD-CAD	47.347	4.053	**0.812**	45.745	3.966	**0.810**	45.745	3.966	**0.810**
USD-CHF	7.178	5.216	**1.584**	5.589	5.134	**2.334**	5.589	5.134	**2.333**
USD-HKD	21.829	42.263	**0.522**	20.234	39.960	**0.446**	20.235	39.961	**0.446**
USD-IDR	66.144	3.179	**0.627**	64.537	3.101	**0.619**	64.537	3.101	**0.619**
USD-JPY	39.793	4.770	**0.833**	38.193	4.674	**0.735**	38.194	4.674	**0.731**
USD-ZAR	69.032	10.335	**1.292**	67.424	10.035	**1.290**	67.424	10.035	**1.290**

Tab. VI *The MAE* $(\times 10^{-3})$ *of FNN model using softsign, arctan, and tanh AFs in the testing processes.*

6. Conclusion

This paper observes the amplitude modification effect in S-shaped activation functions, namely sigmoid, softsign, arctan, and tanh, to the accuracy of feed-forward neural network (FNN) regression. We provide theoretical analyzes of that effect by using simplified FNN model. In addition, some experiments of time series prediction are conducted using ten foreign exchange datasets to verify the theoretical analyzes. The results show that higher amplitude gives higher accuracy up to 3–10 times than low amplitude. These occur in softsign, arctan, and tanh AFs, yet they do not apply in sigmoid AF. These happen in either square or absolute error as the cost function. They are not influenced by the FNNs' architecture and the choice of the initial weight. However, the choice of the initial weight influences the convergence process. Therefore, it is necessary to choose an initial weight that is near the optimum weight. The tested amplitudes are ≥ 100 times than the maximum input values to the FNN. Besides, the stable learning rate when using high amplitude AF tends to be smaller than using low amplitude since the condition of the objective function is steep.

Acknowledgement

This work was supported by Department of Computer Science and Electronics, Universitas Gadjah Mada, Indonesia.

References

[1] ALKHOULY A.A., MOHAMMED A., HEFNY H.A. Improving the Performance of Deep Neural Networks Using Two Proposed Activation Functions. *IEEE Access*. 2021, 9, pp. 82249–82271, doi: 10.1109/ACCESS.2021.3085855.

[2] APICELLA A., DONNARUMMA F., ISGRÒ F., PREVETE R. A survey on modern trainable activation functions. *Neural Networks*. 2021, 138, pp. 14–32, doi: 10.1016/j.neunet.2021.01.026.

[3] AYODELE O.F., AYODELE B.V., MUSTAPA S.I., FERNANDO Y. Effect of activation function in modeling the nexus between carbon tax, CO2 emissions, and gas-fired power plant parameters. *Energy Conversion and Management: X*. 2021, 12, 100111, doi: 10.1016/j.ecmx.2021.100111.

[4] BADURA A., KRYSIŃSKI J., NOWACZYK A., BUCIŃSKI A. Application of artificial neural networks to the prediction of antifungal activity of imidazole derivatives against Candida albicans. *Chemometrics and Intelligent Laboratory Systems*. 2021, 222, doi: 10.1016/j.chemolab.2022.104501.

[5] CHENG Q., LI H.L., WU Q., MA L., NGAN K.N. Parametric Deformable Exponential Linear Units for deep neural networks. *Neural Networks*. 2020, 125, pp. 281–289, doi: 10.1016/j.neunet.2020.02.012.

[6] CLEVERT D.A., UNTERTHINER T., HOCHREITER S. Fast and accurate deep network learning by exponential linear units (ELUs). *4th International Conference on Learning Representations, ICLR 2016*, San Juan, Puerto Rico. Arxiv, 2016, pp. 1-–14, doi: 10.48550/arXiv.1511.07289.

[7] COSTARELLI D. *Sigmoidal Functions Approximation and Applications*[online]. Roma, 2013. PhD Thesis, Universita Degli Studi Roma Tre [viewed 2021-07-20]. Available from: http://www.matfis.uniroma3.it/Allegati/Dottorato/TESI/costarel/Sigmoidal_Approximation_Costarelli.pdf.

[8] CYBENKO G. Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals and Systems*. 1989, 2(4), pp. 303–314, doi: 10.1007/BF02551274.

[9] EBRAHIMZADEH A., GHAFARI M., Moshkbar-Bakhshayesh K. Detection and estimation of faulty sensors in NPPs based on thermal-hydraulic simulation and feed-forward neural network. *Annals of Nuclear Energy*. 2022, 166, 108726, doi: 10.1016/j.anucene.2021.108726.

[10] ERTUĞRUL Ö.F. A novel type of activation function in artificial neural networks: Trained activation function. *Neural Networks*. 2018, 99, pp. 148–157, doi: 10.1016/j.neunet.2018.01.007.

[11] FADEL M.M., ELSEDDEQ N.G., ARNOUS R., ALI Z.H., ELDESOUKY A.I. A Fast Accurate Deep Learning Framework for Prediction of all Cancer Types. *IEEE Access*. 2022, 10, pp. 122586–122600, doi: 10.1109/ACCESS.2022.3222365.

[12] GOH S.L., MANDIC D.P. Recurrent neural networks with trainable amplitude of activation functions. *Neural Networks*. 2003, 16(8), pp. 1095–1100, doi: 10.1016/S0893-6080(03)00139-4.

[13] KLAMBAUER G., UNTERTHINER T., MAYR A., HOCHREITER S. Self-normalizing neural networks. In: *Proceedings of the 31st International Conference on Neural Information Processing Systems (NIPS'17)*, Red Hook, NY, USA. Curran Associates Inc, 2017, pp. 972–981, https://dl.acm.org/doi/10.5555/3294771.3294864.

[14] HE K., ZHANG X., REN S., SUN J. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. *Proceedings of the IEEE International Conference on Computer Vision, 2015 International Conference on Computer Vision, ICCV 2015*. IEEE, pp. 1026–1034, doi: 10.1109/ICCV.2015.123.

[15] JIANG F., LI C.Y. Study on Geographically Attribute Neural Network Weighted Regression. *2022 3rd International Conference on Geology, Mapping and Remote Sensing, ICGMRS 2022*, Zhoushan, China. IEEE, 2022, pp. 213–218. doi: 10.1109/ICGMRS55602.2022.9849396.

[16] KAMATH R.S., KAMAT R.K. Magnitude Prediction Model for Japan Seismic Tremors Using Artificial Neural Network. *2022 IEEE Pune Section International Conference (PuneCon)*, Pune, India. IEEE, pp. 1–4, doi: 10.1109/PuneCon55413.2022.10014739.

[17] KILIÇARSLAN S., CELIK M. RSigELU: A nonlinear activation function for deep neural networks. *Expert Systems with Applications*. 2021, 174, 114805. doi: 10.1016/j.eswa.2021.114805.

[18] KOLBADINEJAD S., MASHHADIMOSLEM H., GHAEMI A., BASTOS-NETO M. Deep learning analysis of Ar, Xe, Kr, and O2 adsorption on Activated Carbon and Zeolites using ANN approach. *Chemical Engineering and Processing – Process Intensification*. 2022, 170, 108662, doi: 10.1016/j.cep.2021.108662.

[19] LANGER S. Analysis of the rate of convergence of fully connected deep neural network regression estimates with smooth activation function. *Journal of Multivariate Analysis*. 2021, 182, 104695, doi: 10.1016/j.jmva.2020.104695.

[20] MERCIONI M.A., TAT A.M., HOLBAN S. Improving the Accuracy of Deep Neural Networks through Developing New Activation Functions. *Proceedings – 2020 IEEE 16th International Conference on Intelligent Computer Communication and Processing, ICCP 2020*, Cluj-Napoca, Romania. IEEE, pp. 385–391, doi: 10.1109/ICCP51029.2020.9266162.

[21] MERCIONI M.A., HOLBAN S. The Most Used Activation Functions : Classic Versus Current. *2020 International Conference on Development and Application Systems (DAS)*, Suceava, Romania. IEEE, 2020, pp. 141–145, doi: 10.1109/DAS49615.2020.9108942.

[22] MISHRA A., CHANDRA P., GHOSE U., SODHI S.S. Bi-modal derivative adaptive activation function sigmoidal feedforward artificial neural networks. *Applied Soft Computing Journal*. 2017, 61, pp. 983–994. doi: 10.1016/j.asoc.2017.09.002.

[23] OROUJI N., MOSAVI M.R., MARTIN D. A Lightweight and Real-Time Hardware Architecture for Interference Detection and Mitigation of Time Synchronization Attacks Based on MLP Neural Networks. *IEEE Access*. 2021, 9, pp. 142938–142949, doi: 10.1109/ACCESS.2021.3120668.

[24] PAUL A., BANDYOPADHYAY R., YOON J.H., GEEM Z.W., SARKAR R. SinLU: Sinu-Sigmoidal Linear Unit. *Mathematics*. 2022, 10(3), pp. 1–14, doi: 10.3390/math10030337.

[25] QIUMEI Z., DAN T., FENGHUA W. Improved Convolutional Neural Network Based on Fast Exponentially Linear Unit Activation Function. *IEEE Access*. 2019, 7(61305008), pp. 151359–151367, doi: 10.1109/ACCESS.2019.2948112.

[26] RAM S., KUMAR S., BARAN B. Neurocomputing Activation functions in deep learning : A comprehensive survey and benchmark. *Neurocomputing*. 2022, 503, pp. 92–108, doi: 10.1016/j.neucom.2022.06.111.

[27] RATNA PRAKARSHA K., SHARMA G. Time series signal forecasting using artificial neural networks: An application on ECG signal. *Biomedical Signal Processing and Control*. 2022, 76, 103705, doi: 10.1016/j.bspc.2022.103705.

[28] SHABBIR M.N.S.K., WANG C., LIANG X., ADAJAR E. A Novel Regression Model-Based Toolbox for Induced Voltage Prediction on Rail Tracks Due to AC Electromagnetic Interference of Adjacent Power Lines. *Conference Record – IAS Annual Meeting (IEEE Industry Applications Society)*, Detroit, MI, USA. IEEE. 2022, pp. 1–8, doi: 10.1109/IAS54023.2022.9939715.

[29] SONODA S., MURATA N. Neural network with unbounded activation functions is universal approximator. *Applied and Computational Harmonic Analysis*. 2017, 43(2), pp. 233–268, doi: 10.1016/j.acha.2015.12.005.

[30] SZANDAŁA T. Review and comparison of commonly used activation functions for deep neural networks. *Studies in Computational Intelligence*. 2021, 903, pp. 203–224, doi: 10.1007/978-981-15-5495-7.

[31] TRENTIN E. Networks with trainable amplitude of activation functions. *Neural Networks*. 2001, 14(4–5), pp. 471–493, doi: 10.1016/S0893-6080(01)00028-4.

[32] WANG J., LI H., ZHAO L. Accelerated Gradient-free Neural Network Training by Multi-convex Alternating Optimization. *Neurocomputing*. 2022, 487, pp. 130–143, doi: 10.1016/j.neucom.2022.02.039.

[33] WANG Q., MA Y., ZHAO K., TIAN Y. A Comprehensive Survey of Loss Functions in Machine Learning. *Annals of Data Science*. 2022, 9(2), pp. 187–212, doi: 10.1007/s40745-020-00253-5.

[34] YILMAZ A., POLI R. Successfully and efficiently training deep multi-layer perceptrons with logistic activation function simply requires initializing the weights with an appropriate negative mean. *Neural Networks*. 2022, 153, pp. 87–103, doi: 10.1016/j.neunet.2022.05.030.