ARTIFICIAL INTELLIGENCE STUDIES
2022, 5(1): 1-13

www.aistudies.org

Gazi AKADEMİK YAYINCILIK

Çok-Katmanlı Ağlar İçin Swish Tabanlı Aktivasyon Fonksiyonlarının Performans Değerlendirmesi

Yılmaz KOÇAK[*, a], Gülesen ÜSTÜNDAĞ ŞİRAY[b]
[a,*] Çukurova Üniversitesi Adana Meslek Yüksekokulu, ADANA 01160, TÜRKİYE
[b] Çukurova Üniversitesi Fen-Edebiyat Fakültesi İstatistik Bölümü, ADANA 01330, TÜRKİYE

MAKALE BİLGİSİ
Alınma: 11.12.2021
Kabul: 23.02.2022

Anahtar Kelimeler
YSA, aktivasyon fonksiyonu, Exp-Swish AF, Swish AF, çok-katmanlı perseptron

[*]Sorumlu Yazar:
e-posta:
ykocak@cu.edu.tr

ÖZET

Yapay Sinir Ağları (YSA), birçok disiplinde evrensel olarak kullanılan ve regresyon, sınıflandırma, fonksiyon yaklaşımı, tanılama, kontrol, örüntü tanıma ve tahmin gibi karmaşık gerçek dünya problemlerini modellemek için kullanılan hesaplamalı modelleme araçlarıdır. YSA'nın önemi doğrusal olmama, hata toleransı, paralellik ve öğrenme gibi biyolojik sistemlerin bilgi işleme özelliklerinden kaynaklanmaktadır. Aktivasyon fonksiyonu (AF), ağlara doğrusal olmama özelliği kazandırdığı için YSA'nın çok önemli bir özelliğidir. Literatürde kullanılan en çok bilinen AF'lerden birisi genellikle derin ağlarda kullanılan Swish AF'dir. Bu çalışmada Swish AF'ye ek olarak çok katmanlı algılayıcı için Mish, E-Swish ve üstel Swish (Exp-Swish) AF gibi Swish tabanlı üç farklı AF kullanılmıştır. Çok katmanlı ağlar için farklı AF'ler kullanan YSA modellerini karşılaştırmak için California Irvine Üniversitesi makine öğrenmesi veritabanından dört farklı kıyaslama veri seti kullandık ve Exp-Swish AF'nin en iyi performansa sahip olduğu sonucunu elde ettik.

DOI: 10.30855/AIS.2022.05.01.01

Performance Evaluation of Swish-Based Activation Functions for Multi-Layer Networks

ARTICLE INFO

Received: 11.12.2021
Accepted: 23.02.2022

Keywords
ANN, activation function, Exp-Swish AF, Swish AF, multi-layer perceptron

[*]Corresponding Authors
e-mail:
ykocak@cu.edu.tr

ABSTRACT

Artificial Neural Networks (ANNs) are computational modelling implements that are universally endured in many disciplines and utilized to model complicated real-world problems such as regression, classification, function approximation, identification, control, pattern recognition, and forecasting. The importance of the ANN originates from the information processing properties of biological systems like nonlinearity, fault tolerance, parallelism, and learning. Activation function (AF) is a crucial characteristic of an ANN since it gains nonlinearity to the networks. One of the most well-known AF used in the literature is Swish AF which is generally used in deep networks. In this study, in addition to the Swish AF we use three different AFs based on the Swish AF which are Mish, E-Swish, and Exponential Swish (Exp-Swish) AFs for multi-layer perceptron. In order to compare the ANN models using different AFs for multi-layer networks, we use four different benchmark datasets from the University of California Irvine (UCI) Machine Learning Repository and get the result that Exp-Swish AF has the best performance.

DOI: 10.30855/AIS.2022.05.01.01

1. INTRODUCTION (*GİRİŞ*)

McCulloch, a neurophysiologist, and Pitts, a mathematician, took place the first step on neural networks by explaining how neurons do work in 1943. They modelled a simple neural network with electrical circuits [1]. Great progress on deep networks has been made since then.

Basically, an ANN is a computational modelling method that learns from examples by means of iterations without prior information about the relationships of parameters. Weights, which are modifications of internal parameters, provide the network to learn any function mapping from the input to the output of the network. Since AFs acquire nonlinearity to the networks, their choosing can seriously affect the performance of the ANN.

The selection of AFs has a considerable effect on the network performance and plays a major role in the success of training. Sigmoid and tanh AFs are mostly used in the ANN and are widely known in the literature. In addition, sigmoid AF, also called logistic function, can be used in logistic regression, as a membership function in fuzzy theory and as an approximation of Gaussian probability distribution [2-4]. However, these AFs are rarely employed in deep networks since their saturation is also called soft saturation. Because of saturation, they have zero gradient in the limit. Soft saturation has some difficulties of the training multi-layer perceptron and deep network, such as the derivative of sigmoid function goes to zero during saturation area and this called vanishing gradient. In deep and multi-layer networks, the sigmoid AF is frequently used in the output layer owing to its value distribution. Because of the limitation of sigmoid AF several advancements and new AFs have been proposed.

There exist some articles that proposed new AFs and investigated their properties of them in the literature. The primary of these articles can be given as follows: Glorot et al. [5] suggested Rectified Linear Unit (ReLU) AF, by demonstrating that ANNs with rectifying neurons give equal or superior performance than hyperbolic tangent networks despite the hard non-linearity and non-differentiability at zero. Agostinelli et al. [6] introduced adaptive piecewise linear AF that learns independently for each neuron employing gradient descent and can represent both convex and non-convex functions of the input and can bring about noteworthy performance enhancements in the deep neural network (DNN). Ramachandran et al. [7] proposed Swish AF in their study to address the deficiencies of producing zero results against negative inputs, and they show that the Swish AF works better than the ReLU on deeper models. Alcaide [8] defined E-Swish AF, which is a generalized version of the Swish AF, and illustrated that the E-Swish is superior to the ReLU and Swish AFs even if the hyperparameters are designed for the ReLU AF. Misra [9] described Mish AF for the purpose of dealing with the disadvantage of the ReLU AF known as the dying-ReLU that is a gradient information loss via collapsing negative inputs to zero. Koçak and Şiray [10] proposed new AFs, generalized swish, mean-swish, ReLU-swish, triple-state swish, sigmoid-algebraic, triple-state sigmoid, exponential swish, sinc-sigmoid and derivative of sigmoid AFs, that combine the advantages of predefined AFs and outperform them.

In this study, to compare the performances of Swish-based AFs for multi-layer networks we used four different benchmark datasets published on the UCI Machine Learning Repository. The first of these datasets, 2-Dimensional Simultaneous Optical Multiprocessor Exchange Bus, is a real data set, which is created by Acı and Akay [11]. In [11], the performance of the prediction models is assessed by using various criteria and 10-fold cross-validation besides in [12,13] it is shown that the Support Vector Regression (SVR) model with the radial-based function has the least predictive error among all models. The second dataset, which is about city-cycle fuel consumption in miles per gallon (MPG) originally got from StatLib library that is kept going on at Carnegie Mellon University, and also used by Quinlan [14]. In the used third dataset, energy efficiency dataset, energy analysis is done by using 12 different building shapes simulated in Ecotect Software [15]. In [16], as a result of planning and developing an ANN model to predict heating and cooling loads of a building based on this dataset for building energy performance, the most important factors affecting heating and cooling load were identified, and the validation accuracy was obtained as 99.06%. The last dataset produced by I-Cheng [17] and used by Asteris and Mokos [18] is about concrete compressive strength. In [18] to predict the compressive strength of concrete in existing structures the application of ANNs is investigated by using both the ultrasonic pulse velocity and the Schmidt rebound hammer experimental results and it was concluded that the ANNs have the ability to estimate the compressive strength of concrete reliably and robustly than the experimental findings.

The paper is organized as follows: The AFs used in this study are presented in Section 2. Experimental studies and comparisons of each model using datasets mentioned above are done in Section 3. Lastly, we give some conclusions in Section 4.

2. ACTIVATION FUNCTION *(AKTİVASYON FONKSİYONU)*

AFs are used to compute the weighted sum of inputs and biases which is assisted to determine whether the neuron can be fired or not. AFs are also called transfer functions in some literature. They can be linear or non-linear depending on function represents. AFs can be characterized by different features regarded for successful learning such as differentiability, monotonicity or non-monotonicity, and if their range is finite or not. While some AFs like the sigmoid and tanh are equivalent in theoretical level, different AFs show very different behavior in practice. The sigmoid and tanh AFs mostly used in the past turned out less suitable because their derivatives have vanishing gradients. In the use of the sigmoid and tanh AFs that are squashed from top and bottom limits, the initial value of the network must be attentively specified to stay in the linear regions of the functions. Glorot et al. [5] prove that the ReLU AF is much more suitable than squashing functions. The ReLU has an identity derivative in the positive part so it can be less sensitive to vanishing gradients. To date, several AFs such as sigmoid, hard-sigmoid, tanh, SiLU, dSiLU, softplus, softmax, ReLU, LReLU, SReLU, PReLU, ELU, SELU, Elish, Swish, etc. are examined [19-21].

The Swish AF, presented as follows, proposed at [7] is formulated by multiplying the input and sigmoid function;

$$y = \frac{x}{1+e^{-x}} = x.sigmoid(x) \qquad (1)$$

The AF given in (1) is limited from below and unlimited from the above like the ReLU AF. Unlike the ReLU AF, the Swish AF is not monotonous. Non-monotonicity is the most prime characteristic that distinctive the Swish AF from other AFs. The accomplishment of the Swish AF shows that the gradient preserving of the ReLU AF is no longer a significant advantage in modern network architecture. The gradient preserving property means that derivative of the ReLU AF is equal to 1 for $x > 0$.

Actually, the Swish AF can be considered as a smooth function interpolating between linear and ReLU AF. The functions that approach zero at the limit generate a larger normalization effect because of the forgetting of large negative inputs. The ReLU, Swish, and Softplus AFs approach to zero at the limit. Swish AF is self-gating which is the gating mechanism by using the same value to get itself. While self-gating only demands a single scalar, normal gating demands multiple scalar inputs. [19,22].

One of the disadvantages of the ReLU AF is known as dying ReLU, which is a gradient information loss originated by going the negative inputs to zero. The Swish and Swish-like AFs will overcome this problem. E-Swish AF is proposed inspired by the Swish AF [8]. It is shown that using E-Swish provided 1.5% and 4.6% accuracy improvements respectively on Cifar10 and Cifar100 when compared to ReLU, and 0.35% and 0.6% respectively when compared to Swish. It is a generalized version of the Swish AF and expressed multiplying by a parameter as seen in (2).

$$y = \frac{\beta x}{1+e^{-x}} = \beta x Sigmoid(x) = \beta Swish(x) \qquad (2)$$

Choosing large values of β may cause gradient exploding problems, so it is suitable in the range $1 \le \beta \le 2$. Alcaide [8] shows that the E-Swish AF is systematically superior to the other well-known AFs, such as the Swish and ReLU AFs. The E-Swish AF is bounded below and unbounded above like the ReLU and Swish AFs. On the other hand, like the Swish AF, it has non-monotonicity and smoothness. Non-monotonicity favors the performance of AF. For the negative inputs, the Swish and E-Swish can output small negative values which give non-monotonicity property.

Misra [9] proposed a new AF called Mish which is a self-regularized non-monotonic function motivated by the self-gating feature of the Swish AF and can be defined mathematically as in (3).

$$y = x\tanh(\ln(1+e^{x})) = x\tanh(softplus(x)) \qquad (3)$$

The Mish AF is similar to the Swish AF which has smoothness, non-monotonicity, and ability to preserve negative weights. In addition to this, it is bounded below and unbounded above likewise the ReLU and Swish AFs. As said before, the Mish AF has self-gating property in which non-modulated input is multiplied by the output of non-linear function of the input. Eliminating dying ReLU property assists in better information flow and expressivity. Unlike the ReLU but like the Swish, Mish is continuously differentiable and so prevents undesired side effects and singularities during gradient-based optimization. The author demonstrated that Mish outperformed Leaky ReLU on YOLOv4 on average precision by 2.1% object detection and ReLU on ResNet-50 on accuracy by 1%.

The performance comparisons of some AFs of the ANN with single hidden layer are done by experimental studies in Koçak and Şiray [10]. The authors compared statistical regression and neural regression with different AFs for regression problem. They showed that all the ANN models outperform statistical regression. One of the AFs defined in the paper is the Exp-Swish which is calculated by multiplying e^{-x} and the sigmoid AF inspired by the Swish AF, below:

$$y = \frac{e^{-x}}{1+e^{-x}} = e^{-x}sigmoid(x) \qquad (4)$$

The Exp-Swish AF is another AF we examine in this paper.

3. EXPERIMENTAL STUDY *(DENEYSEL ÇALIŞMA)*

In this section, an experimental study for each aforementioned AF is carried out by using four different benchmark datasets. The performances of AFs are evaluated and compared with each other on the regression problems using four different datasets that are frequently utilized in machine learning studies. These datasets are got from the UCI repository of machine learning and are known as Optical Interconnection Network [11], Auto MPG [14], Energy Efficiency [15], and Concrete Compressive Strength [17]. The inputs and ouputs are normalized into the interval [-1,1] for all datasets. Each data set is divided into the training and test data randomly. Data characteristics, sample size, and the size of test and training are given in Table 1. The codes written in the Matlab Software environment are run Intel(R) Core (TM) i7-10750H CPU 2.60 GHz, 16 GB RAM X64 based processor.

Table 1. Benchmark datasets for experimental study *(Deneysel çalışma için kıyas verileri).*

Datasets	Property	Sample size	Training size	Test size
Optical Interconnection Network	5	640	448	192
Auto Fuel Consumption	6	392	274	118
Concrete Compressive Strength	8	1030	721	309
Energy Efficiency	7	768	538	230

The values of the parameters in the training of ANN are 1000 for maximum epoch count, 10^{-7} for minimum performance gradient. Layer-by-layer initialization which initializes the weights and biases for each layer with the Nguyen-Widrow initialization method is utilized for network parameters [23]. This algorithm is used in order to speed up the training process. Small random initial weights are set up so that each hidden node is appointed to approximate a part of the range desired function at the start of the training. The weights need to move towards the region of interest are splitted into small intervals. In the training process, each hidden node is appointed its own interval at the beginning of training by adjusting the initial weights of the hidden layer [24].

Mean Squared Error (MSE), coefficient of determination (R^2), and simulation time (in seconds) metrics are used for the comparison. MSE is calculated by means of squared of difference between real and estimated values as in (5). MSE can take values from 0 to ∞, and a smaller MSE value means better performance. The coefficient

of determination is the coefficient of how well the values fit relative to the original values. R^2, in (6), which is interpreted as a percentage, can take the values between 0 and 1, and the higher the value the better it represents the model. If the coefficient of determination equals 1, it indicates a perfectly linear relationship between actual and target output, and if the coefficient of determination is close to zero, it means that there is no linear relationship between actual and target output.

$$MSE = \frac{1}{N}\sum_{i=1}^{N}\left(y_i - \hat{y}_i\right)^2 \quad (5)$$

$$R^2 = 1 - \frac{\sum_{i=1}^{N}\left(y_i - \hat{y}_i\right)^2}{\sum_{i=1}^{N}\left(y_i - \bar{y}\right)^2} \quad (6)$$

3.1. Optical Interconnection Network Dataset *(Optik Enterkonnekte Ağ Veriseti)*

The first dataset, got from UCI Machine Learning Repository and generated with 32-OPNET Modeler, is utilized to simulate the message passing the 2D SOME-Bus (2D Simultaneous Optical Multiprocessor Exchange Bus) multiprocessor architecture. Acı and Akay [11] measured the performance of the message passing architecture interconnected with 2D SOME-Bus by using multiple linear regression, SVR, and multilayer feedforward ANN. They used 32-OPNET Modeler to simulate the message passing 2D SOME-Bus multiprocessor architecture, created training, and test datasets. In this dataset, the number of nodes, number of threads, spatial distribution, transient distribution, and the ratio of the mean message channel transfer time to the mean thread run time (T/R) are explanatory variables, and average channel waiting time, average processor utilization, average network latency (network response time), average input waiting time, and average channel utilization are output variables [12,13]. Acı and Akay [11] employed 10-fold cross-validation and assessed the performance of the prediction models by utilizing various criteria. The SVR model with the radial-based function has the least predictive error among all models [11,12]. Akay et al. [13] used Multi-layer Feedforward ANN (MFANN) with sigmoid AF in the hidden layer and linear function in the output layer. They obtained R^2=0.9216 using network response time as a response variable.

In this study, we take network response time as an output variable for multiple linear regression and get the regression model as in (7) and regression plot as in Figure 1. As seen in Figure 1, there is a small overfitting.

$$\hat{y} = -0.00039934 + 0.16346\,x_1 + 0.30369\,x_2 - 0.11463\,x_3 - 0.44333\,x_4 + 0.64038\,x_5 \quad (7)$$

Figure 1. Statistical regression plot for the optical interconnected network *(Optik enterkonnekte ağ için istatistiksel regresyon eğrisi)*.

To compare statistical and neural regression, we used the ANN architecture with three hidden layers given in Figure 2. The neuron number of the input layer is equal to the number of explanatory variables, output neuron number is equal to the number of the response variable. We arbitrarily choose the number of hidden layers and neuron number in each layer by experience. So, the architecture model of the neural network is ANN(5,10,20,10,1),

where the first parameter is the number of input neurons, the last parameter is the number of output neurons and the others are the number of hidden neurons, respectively.

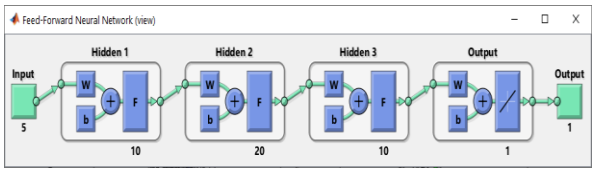

Figure 2. Architecture of the ANN for optical interconnection network dataset *(Optik enterkonntekte ağ için YSA mimarisi).*

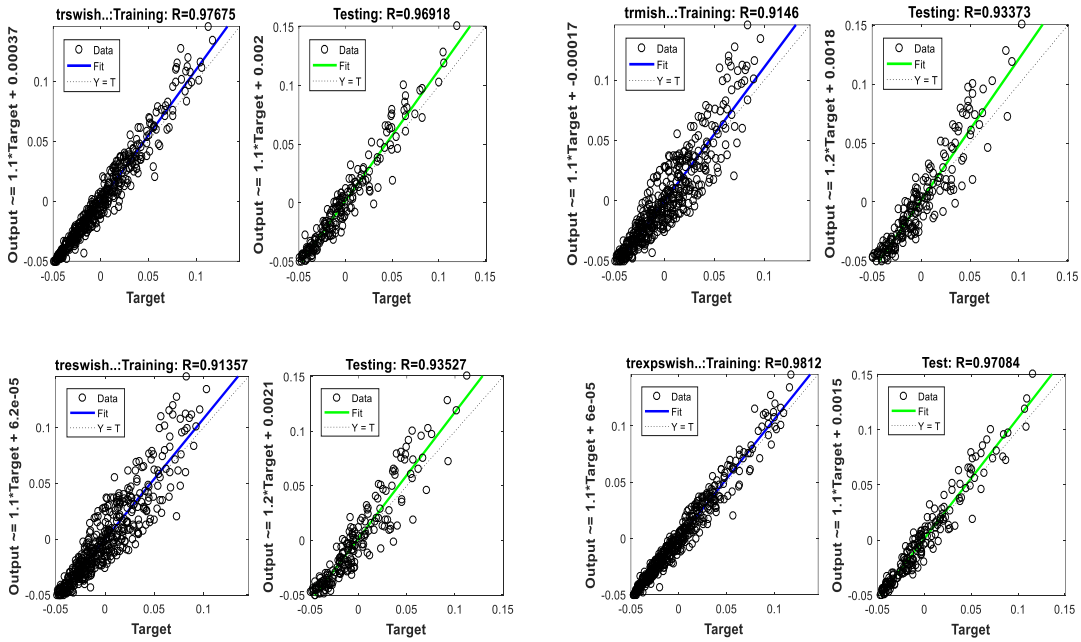

Figure 3. Regression plots of the ANN with the Swish, Mish, E-Swish, and Exp-Swish AFs, respectively *(Swish, Mish, E-Swish ve Exp-Swish AF'ları kullanan YSA regresyon eğrileri).*

Table 2**.** Results of statistical and neural regression for the optical interconnected network dataset *(Optik enterkonnekte ağ verisi için istatistiksel ve sinir ağı regresyonlarının sonuçları)*

Method		Training MSE	Test MSE	Training R^2	Test R^2	Simulation Duration
Statistical Regression		0,000367	0,000350	0,758527	0,793417	**1,810425**
AFs for ANN	Swish	0,000081	0,000120	0,954038	0,939317	47,804599
	Mish	0,000261	0,000254	0,836490	0,871859	52,441510
	E-Swish	0,000258	0,000237	0,834611	0,874739	60,886624
	Exp-Swish	**0,000061**	**0,000109**	**0,962755**	**0,942533**	39,026620

As shown in regression plots of the ANN models given in Figure 3, there is a small overfitting for the ANN models with the Mish and E-Swish AFs, because the determination coefficient of test data is bigger than for training data. Results of statistical and neural regression for the optical interconnected network dataset are given in Table 2. According to the Table 2, all the ANN models overperformed statistical regression for all metrics except simulation time. Among the ANN models, the model used Exp-Swish has the best performance in terms of all metrics.

3.2. City-Cycle Fuel Consumption Dataset *(Şehir İçi Yakıt Tüketimi Veriseti)*

The dataset got from the UCI Machine Learning Repository includes the technical specifications of the cars in miles per gallon in city-cycle. This dataset has the variables such as acceleration, cylinder size, displacement, horsepower, model year, weight, and fuel consumed in miles per gallon (MPG). MPG is the output or response variable, the others are explanatory or input variables. The regression model and plot are given in (8) and Figure 4, respectively. As seen in Figure 5, the architecture model is ANN(6,12,24,12,1). Regression results are got by 100 times Monte Carlo simulation.

$$\hat{y} = -0.00052 + 0.01979x_1 - 0.10756x_2 + 0.19810x_3 + 0.00686x_4 + 0.33906x_5 - 0.81013x_6 \quad (8)$$

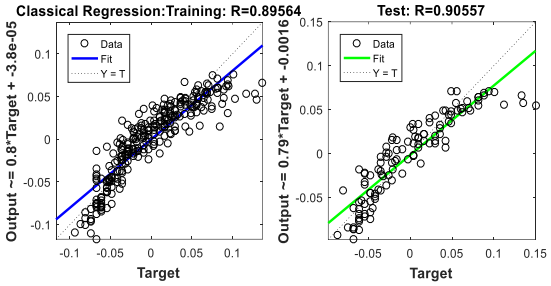

Figure 4. Statistical regression plot for the Auto MPG dataset *(Oto MPG veriseti için istatistiksel regresyon eğrisi)*.

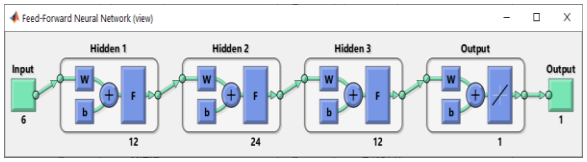

Figure 5. ANN architecture for the Auto MPG dataset *(Oto MPG veriseti için YSA mimarisi)*.

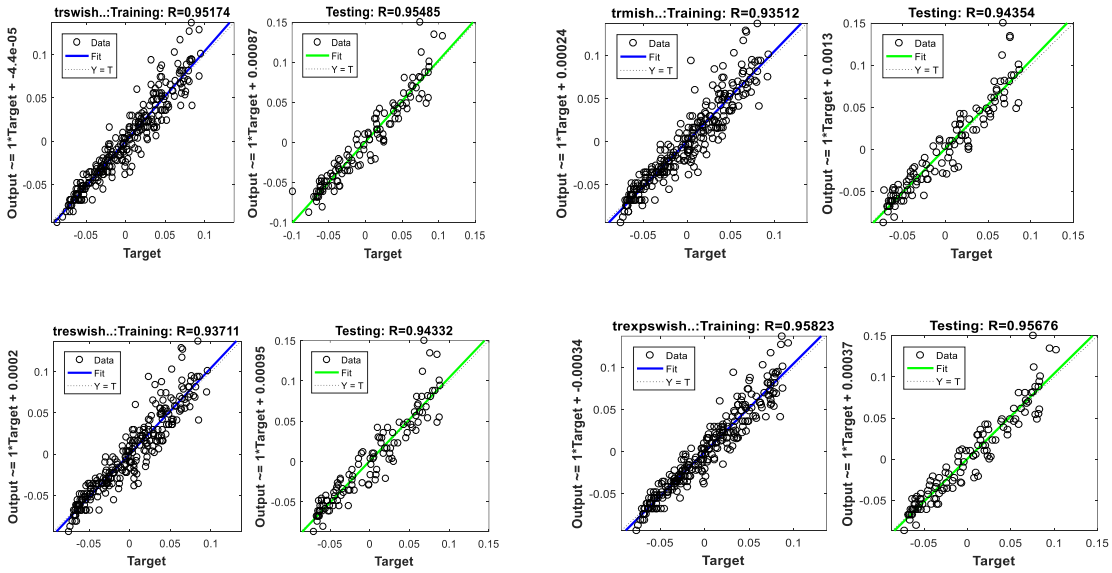

Figure 6. Neural regression plots with Swish, Mish, E-Swish and Exp-Swish AFs, respectively, for the Auto MPG *(Oto MPG veriseti için Swish, Mish, E-Swish ve Exp-Swish AF'ları kullanan sinir ağlarının regresyon eğrileri)*

Figure 6 shows regression plots of the ANN models which architecture is given in Figure 5. Results of the statistical and neural regression for the Auto MPG dataset are given in Table 3. According to Table 3, all the ANN

models overperformed the statistical regression for all metrics except simulation time. Among the ANN models, the model used the Exp-Swish AF has the best performance in terms of all metrics.

Table 3. Results of the statistical and neural regression for the Auto MPG dataset *(Oto MPG veriseti için istatistiksel ve sinir ağı regresyon sonuçları)*

Method		Training MSE	Test MSE	Training R^2	Test R^2	Simulation Duration
Statistical Regression		0,0004798	0,0005173	0,8021787	0,8200622	**1,7838312**
AFs for ANN	Swish	0,0002322	0,0002521	0,9058120	0,9117361	45,758643
	Mish	0,0003092	0,0003179	0,8744422	0,8902766	63,624691
	E-Swish	0,0002992	0,0003164	0,8781696	0,8898449	108,103219
	Exp-Swish	**0,0002034**	**0,0002441**	**0,9182136**	**0,9153895**	45,080864

3.3. Energy Efficiency Dataset *(Enerji Verimliliği Veriseti)*

Energy efficiency dataset, which contains 768 samples and 8 features aimed at predicting two real-valued responses, is shared in the UCI Machine Learning Repository. Energy analysis is performed employing 12 different building shapes simulated in Ecotect Software. The goal here is to use the 8 independent variables to predict the two response variables [13]. However, for multiple linear regression analysis, we take just one response variable and 7 explanatory variables, like this the response variable is heating load explanatory variables are relative compactness, surface area, wall area, overall height, orientation, glazing area, glazing area distribution. For this dataset, regression results are got by 100 times Monte Carlo simulation and given model in (9) and plots are given in Figure 7.

$$\hat{y} = 0.00019 - 0.67397x_1 - 0.74641x_2 + 0.25278x_3 + 0.73814x_4 + 0.00140x_5 + 0.25833x_6 + 0.02520x_7 \quad (9)$$

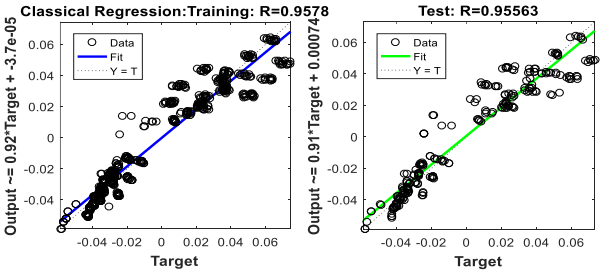

Figure 7. Statistical regression plot for the energy efficiency dataset *(Enerji verimliliği veriseti için istatistiksel regreyon eğrisi).*

As shown in Figure 7, obtained R-value is 0.95563 and the determination coefficient is 0.9132, so the explanatory variables can explain 91% of the model.

For the energy efficiency dataset, we used the ANN architecture with three hidden layers in Figure 8 in which neuron size of the input layer is equal to the number of explanatory variables, output neuron size is equal to the number of the response variable. The architecture model designed by taking 3 hidden layers and arbitrarily chosen hidden neurons is ANN(7,7,14,7,1).

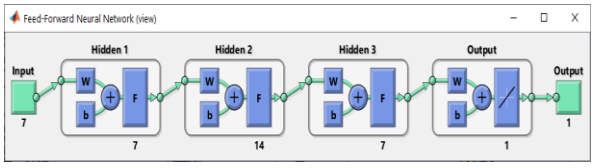

Figure 8. ANN architecture for the energy efficiency dataset *(Enerji verimliliği veriseti için YSA mimarisi).*

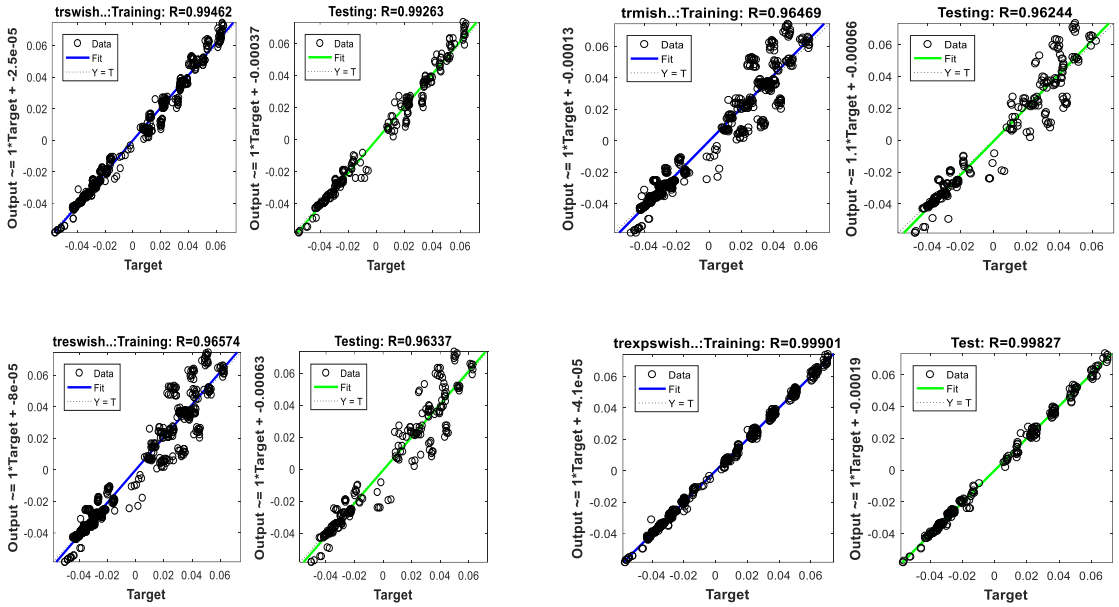

Figure 9. Neural regression plots with the Swish, Mish, E-Swish and Exp-Swish AFs, repectively, for the energy efficiency dataset *(Enerji verimliliği veriseti için Swish, Mish, E-Swish ve Exp-Swish AF'ları kullanan sinir ağlarının regresyon eğrileri).*

Table 4. Results of statistical and neural regression for the energy efficiency dataset *(Enerji verimliliği veriseti için)*

Method		**Training MSE**	**Test MSE**	**Training R^2**	**Test R^2**	**Simulation Duration**
Statistical Regression		0,0001065	0,0001160	0,9173884	0,9132361	**1,8352960**
AFs for ANN	Swish	0,0000148	0,0000208	0,9892765	0,9853182	41,4122524
	Mish	0,0000921	0,0001015	0,9306218	0,9262983	37,9792852
	E-Swish	0,0000881	0,0000974	0,9326525	0,9280743	47,2971815
	Exp-Swish	**0,0000026**	**0,0000048**	**0,9980228**	**0,9965492**	44,2108330

As a result of neural regression analysis, as in Figure 9 and Table 4, all the ANN models outperformed statistical regression for all metrics except simulation time. Also, the Exp-Swish AF overperformed on the ANN models by the criteria of the MSE and R^2 for training and test data. However, the Mish AF is the fastest one in the ANN models. The determination coefficient of the ANN model with the Exp-Swish AF is 0.9965 which means input variables can explain 99% of the model.

3.4. Concrete Compressive Strength Dataset *(Beton Basınç Dayanımı Veriseti)*

In civil engineering, one of the most important materials is concrete. Concrete compressive strength is a function of age and components which comprise cement, blast furnace slag, fly ash, water, superplasticizer, coarse aggregate, and fine aggregate [17]. This dataset has 8 explanatory variables and one response variable and using this dataset regression model and regression plot are obtained as in (10) and Figure 10 using multiple linear regression.

$$\hat{y} = 0.00029 + 0.79762x_1 + 0.55737x_2 + 0.3564x_3 - 0.13396x_4 + 0.12806x_5 + 0.1136x_6 + 0.14616x_7 + 0.45259x_8 \quad (10)$$

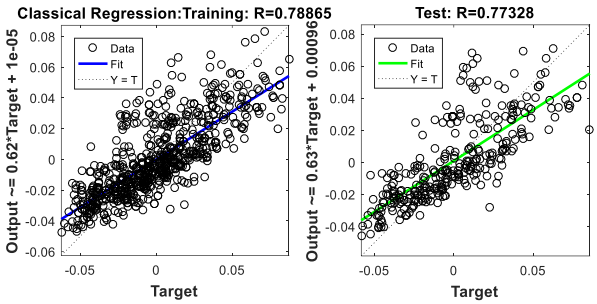

Figure 10. Statistical regression plot for the concrete compressive strength dataset *(Beton basınç dayanımı veriseti için istatistiksel regresyon eğrisi).*

As shown in Figure 10, we obtained the values of R is 0.77328 and R^2 =0.5980, so the explanatory variables can explain 59% of the model.

For the concrete compressive strength, we used the ANN architecture with three hidden layers in Figure 11 in which neuron size of the input layer is equal to 8 which is the number of explanatory variables, output neuron size is equal to, the number of the response variable, 1. Hidden neuron sizes are chosen by trial and error. So, the network architecture is ANN(8,16,32,16,1).

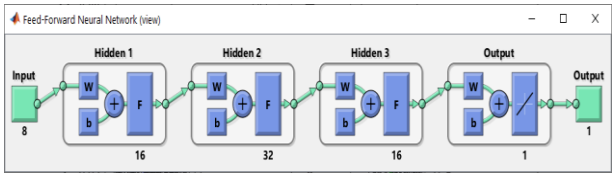

Figure 11. ANN architecture for the concrete compressive strength dataset *(Beton basınç dayanımı veriseti için YSA mimarisi)*

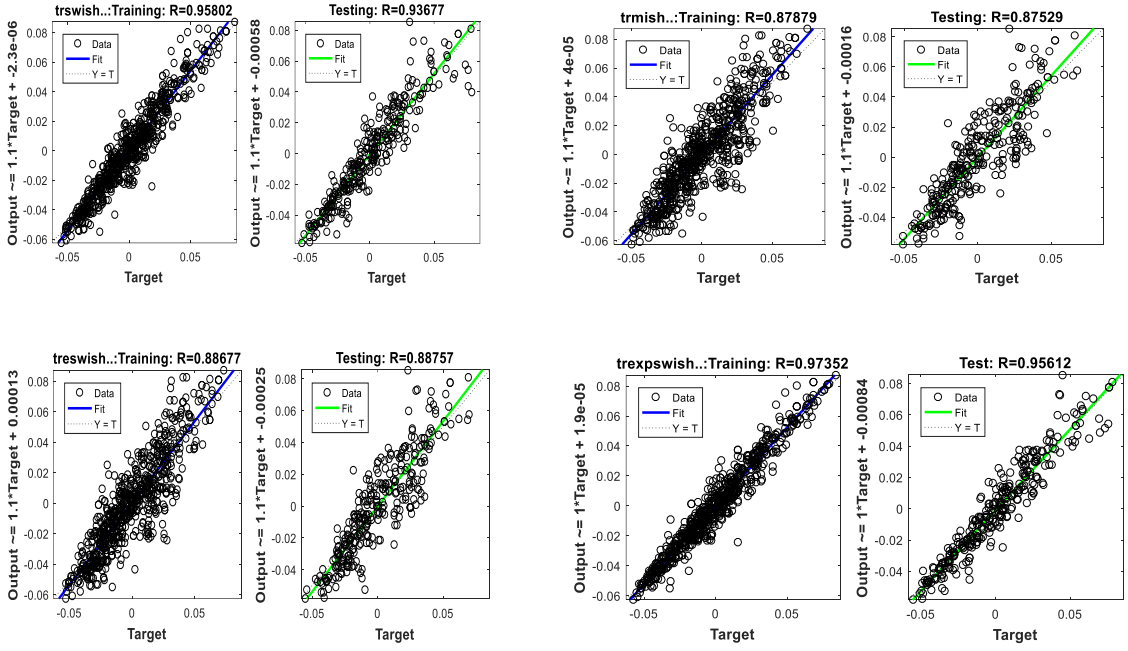

Figure 12. Neural regression plots with the Swish, Mish, E-Swish and Exp-Swish AFs, repectively, for the concrete compressive strength dataset *(Beton basınç dayanımı veriseti için Swish, Mish, E-Swish ve Exp-Swish AF'ları kullanan sinir ağlarının regresyon eğrileri).*

Table 5. Results of statistical and neural regression for the concrete compressive strength dataset *(Beton basınç dayanım verisi için istatistiksel ve sinir ağı regresyon sonuçları)*

Method		Training MSE	Test MSE	Training R^2	Test R^2	Simulation Duration
Statistical Regression		0,0003653	0,0003979	0,6219762	0,5979694	**1,7194565**
AFs for ANN	Swish	0,0000846	0,0001225	0,9177933	0,8775451	407,1412921
	Mish	0,0002267	0,0002343	0,7722698	0,7661350	261,1855047
	E-Swish	0,0002101	0,0002118	0,7863527	0,7877891	406,3351229
	Exp-Swish	**0,0000524**	**0,0000860**	**0,9477376**	**0,9141708**	202,4940290

Neural regression plots for the concrete compressive strength are given in Figure 12 with the Swish, Mish, E-Swish, and Exp-Swish AFs, respectively, and results of the statistical and neural regression for the concrete compressive strength dataset are given in Table 5. As shown in Table 5 and Figure 12, all the ANN models outperformed statistical regression for all metrics except simulation time as expected. Moreover, the Exp-Swish AF has the best performance among the ANN models in terms of the MSE and R^2 criteria.

4. CONCLUSIONS *(SONUÇLAR)*

Since 1943, the ANN has contributed to the science and technology world. ANN technologies especially deep networks recently go forward on the application of all sciences. In order to keep up with developments in science and technology, there should be advancements in ANN models. One of these advancements is about AFs. In this context, we handled the AFs, which crucially affect the performance of the ANN. The sigmoid and tanh AFs were mostly used in applications of neural networks. However, they have some disadvantages such as saturation effects in the limits, the slope of these functions near the origin causes difficulty during training. To overcome these problems, several researchers proposed new activation functions, such as the ReLU, LReLU, PReLU, RReLU, ELU, Maxout, Swish, Mish, E-Swish, Exp-Swish, Elish, etc. The common feature of recently proposed AFs is that they are non-saturated functions. The Swish and most of the Swish-based functions are limited below but unlimited above as in ReLU.

In this study, we examined Swish-based AFs, which are Swish, Mish, E-Swish, and Exp-Swish AFs. We compare the results of the statistical regression and neural regression for multi-layer networks with respect to metrics MSE, R^2, and simulation duration for different benchmark datasets. Although there is a small overfitting for the ANN models with the Mish and E-Swish AFs for the optical interconnected network dataset, all ANN models overperformed statistical regression for all metrics except simulation time. Among the ANN models, the model used Exp-Swish has the best performance in terms of all metrics. Unlike in the optical interconnected dataset, for the other datasets explained above, there is no overfitting. However, as in the optical interconnected dataset, for other datasets, all ANN models outperformed statistical regression for MSE and R^2 metrics. Among ANN models, Exp-Swish AF has the best performance according to MSE and R^2 metrics. On the other hand, as expected statistical regression is the fastest model for the simulation time. As a result, the Exp-Swish AF proposed earlier by us outperformed the other AFs for used datasets and ANN architectures.

Publication Ethics *(Yayın Etiği)*

The shortened version of this paper was presented as an oral paper at the International Conference on Engineering Technologies 2021 (ICENTE'21), Konya, Turkiye.

REFERENCES *(KAYNAKLAR)*

[1] W.S. McCulloch and W. Pitts, "A logical calculus of the ideas immanent in nervous activity." *Bulletin of Mathematical Biophysics 5*, 115–133, 1943, https://doi.org/10.1007/BF02478259.

[2] P. Sibi, S.A. Jones and P. Siddarth, "Analysis of Different Activation Functions Using Backpropagation Neural Networks", Journal of Theoretical and Applied Information Technology, Vol.47, No.3, 1264-1268, 2013.

[3] I.S. Isa, Z. Saad, M.K. Osman, K.A. Ahmad and H.A.M. Sakim, "Suitable MLP Network Activation Functions for Breast Cancer and Thyroid Disease Detection", Second International Conference on Computational Intelligence, Modelling and Simulation, pp.39-44, 2010.

[4] K.V.N. Babu and D.R. Edla, "New algebraic activation function for multi-layered feed forward neural networks", IETE Journal of Research, 4, 70-79, 2017.

[5] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," *In Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics, volume 15 of Proceedings of Machine Learning Research*, Fort Lauderdale, FL, USA, 2011, p.315–323.

[6] F. Agostenelli, M. Hoffman, P. Sadowksi and P. Baldi, "Learning activation functions to improve deep neural networks", arXiv:1412.6830v3 [cs.NE], 2015.

[7] P. Ramachandran, B. Zoph and V.Q. Le, "Swish: A self-gated activation function," *arXiv:1710.05941v1 [cs.NE]*, 2017a.

[8] E. Alcaide, "E-Swish: Adjusting activations to different network depths", *arXiv:1801.07145 [cs.CV]*, 2018.

[9] D. Misra, "Mish: A self-regularized non-monotonic activation function", *arXiv:1908.08681 [cs.LG]*, 2020

[10] Y. Koçak and G.Ü. Şiray, "New activation functions for single layer feedforward neural network", *Elsevier, Expert Systems with Applications*, Vol. 164, 113977, 2021.

[11] Ç. İ. Acı and M.F. Akay, "A hybrid congestion control algorithm for broadcast-based architectures with multiple input queues", *J Supercomput (2015) 71: 1907*, 2015.

[12] M.F. Akay and İ. Abasıkeleş, "Predicting the performance measures of an optical distributed shared memory multiprocessor by using support vector regression", *Expert Systems with Applications* 37(9) pp. 6293–6301, 2010, doi: 10.1016/j.eswa.2010.02.092

[13] M.F. Akay, İ. Abasıkeleş and F. Abut, "Predicting the performance measures of a 2-dimensional message passing multiprocessor architecture by using machine learning methods", *Neural Network World* vol.25, pp.241-265, 2015.

[14] R. Quinlan, "Combining Instance-Based and Model-Based Learning", *In Proceedings on the Tenth International Conference of Machine Learning*, 236-243, University of Massachusetts, Amherst. Morgan Kaufmann.

[15] A. Tsanas and A. Xifara, "Accurate quantitative estimation of energy performance of residential buildings using statistical machine learning tools", *Energy and Buildings*, Vol. 49, pp. 560-567, 2012.

[16] A.J. Khalil, A.M. Barhoom, B.S. Abu-Nasser, M.M. Musleh, and S.S. Abu-Naser, "Energy Efficiency Prediction using Artificial Neural Network", *International Journal of Academic Research (IJAPR)*, ISSN:2643-9603, Vol.3 Issue 9, pp. 1-7, September 2019.

[17] Y. I-Cheng, "Modelling of strength of high-performance concrete using artificial neural networks", *Cement and Concrete Research,* Vol. 28, No. 12, pp. 1797-1808, 1998.

[18] P.G. Asteris, V.G. Mokos, "Concrete compressive strength using artificial neural networks", *Neural Computing & Applications* 32, 11807–11826 (2020). https://doi.org/10.1007/s00521-019-04663-2

[19] C. E. Nwanka, W. Ijomah, A. Gachagan, and S. Marshall, "Activation functions: Comparison of trends in

practice and research for deep learning," *arXiv:1811.003378v1 [cs.LG]*, 2018.

[20] S. Eger, P. Youssef, and I. Gurevych, "Is it time to Swish? Comparing deep learning activation functions across NLP tasks," *Proceedings of the 2018 Conference on Empirical Methods in Natural Processing, Brussels, Belgium,* 4415-4424. 2018.

[21] M. Sipper, "Neural Network with Â la carte selection of activation functions", *SN Computer Science*, 2021, DOI:10.1007/s42979-021-00885-1.

[22] P. Ramachandran, B. Zoph and V.Q. Le, "Searching for activation functions", *arXiv:1710.05941v2 [cs.NE]*, 2017b

[23] D. Nguyen and B. Widrow, "Improving the learning speed of 2-layer neural networks by choosing initial values of the adaptive weights" *IJCNN International Joint Conference on Neural Networks* 3:21–26, 1999.

[24] A. Pavelka and A. Prochazka, "Algorithms for initialization of neural network weights", https://www.researchgate.net/publication/228402422, November 2004.

Yılmaz KOÇAK*

Yılmaz KOÇAK was born in Silifke, Mersin, Turkey, on 20 January 1972. He received the B.Sc. from the Electrical and Electronics Engineering Department in Istanbul Technical University in 1993. In 1999, he received the M.Sc. degree from the Electrical and Electronics Engineering Department in Kahmaranmaraş Sütçü İmam University. He has going on studies of Ph.D. at the Department of Statistics, Çukurova University. He has been working as a lecturer at the Department of Computer Programming in Adana Vocational School of Higher Education at Cukurova University since January 1996. His research interests are computer science, programming, electrical and electronics, artificial intelligence, and statistics.

……

Gülesen ÜSTÜNDAĞ ŞİRAY

Gülesen ÜSTÜNDAĞ ŞİRAY received the B.Sc. degree in 2002 from the Department of Mathematics in Çukurova University, Turkey. She completed her M.Sc. and Ph.D. in the Department of Statistics at the same university in 2005 and 2011, respectively. She is currently an Associate Professor at the Department of Statistics, Çukurova University. Her research interests are linear models, regression analysis, measurement error models, bias and restricted regression estimators and also genetic algorithms, artificial neural networks.