

# Assigning different activation functions in artificial neural networks with the goal of achieving higher prediction accuracy<sup>\*</sup>

Gytis Baravykas<sup>1,\*</sup>, Justas Kardoka<sup>1,†</sup>, Domas Grigaliunas<sup>1,†</sup> and Darius Naujokaitis<sup>1,2,†</sup>

<sup>1</sup> Faculty of Informatics, Kaunas University of Technology, Studentu 50, 51368 Kaunas, Lithuania

<sup>2</sup> Smart Grids and Renewable Energy Laboratory, Lithuanian Energy Institute, 44403 Kaunas, Lithuania

## Abstract

The research paper explores the concept of using multiple activation functions in artificial neural networks and investigates their impact on model performance. The experiments conducted on various models such as AlexNet, ResNet50, TuNet, and SimpleNN reveal insights into the effectiveness of different activation function combinations. The results indicate that using multiple activation functions can lead to modest improvements in model performance, particularly in image segmentation tasks where modifications to the UNet architecture show significant enhancements. However, for time series regression/forecasting tasks, the experiments demonstrate that using multiple activation functions does not significantly improve prediction accuracy. Therefore, the paper concludes that while there are some benefits to using multiple activation functions in certain scenarios, the choice of activation function should be based on the specific task and dataset.

## Keywords

Activation functions, artificial neural networks, machine learning

## 1. Introduction

Artificial neural networks (ANNs) are becoming increasingly more relevant. Although the idea of ANNs spans multiple decades, various ANN architectures are still widely being developed to this day. One of the most important components of ANNs is activation functions. They are often used for introducing non-linearity, and in turn, allow ANNs to understand intricate features in the data. Although different activation functions have been developed and studied, there exists no body of work in which the choice of activation functions would be considered in the case of solar power generation forecasts. In this paper, we propose a new approach for improving the results of ANN predictions via changing the activation functions in the ANN. We have chosen to test our approach on a range of different machine learning tasks, with the goal of introducing a new, alternative hyper-parameter that would work for different ANN architectures.

## 2. Literature review


Activation functions in an ANN are used to introduce non-linear relations to the data, so that the network would better fit the results and improve the accuracy of a given task. It is a very common part of ANNs and often omitted from neural network structure diagrams. Many mathematical functions have been introduced to achieve non-linearity, such as ReLU, Tanh, Sigmoid and others, each tailored to specific tasks. In this paper we entertain the idea of using no one activation function per layer or network, but multiple, assigning a different one for each neuron.

\* IVUS2024: Information Society and University Studies 2024, May 17, Kaunas, Lithuania

<sup>1,\*</sup> Corresponding author

<sup>†</sup> These authors contributed equally.

✉ gytis.baravykas@ktu.lt (G. Baravykas); justas.kardoka@ktu.lt (J. Kardoka); domas.grigaliunas@ktu.lt (D. Grigaliunas);  
darius.naujokaitis@ktu.lt (D. Naujokaitis)

 0000-0002-8548-5056 (D. Naujokaitis)

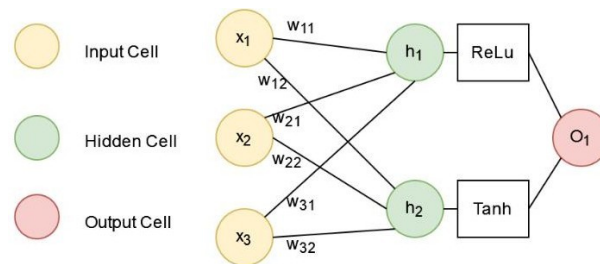


© 2024 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

The importance of activation functions is discussed in many recent works. Their importance is based on their wide-spread usage in ANN architectures. Dubey has published a comprehensive overview of the most common activation functions, along with their characteristics and a performance comparison between them [1]. They have found that different activations functions are more suited for certain machine learning tasks, and that in certain cases, alternative choices must be considered. Although there are some common choices, new activation functions are constantly being developed [2,3,4,5,6]. Yu has created a modified activation function based on ReLU, with the goal of increasing the accuracy of classification tasks [2]. Wang developed a activation function as a better alternative to other commonly used activation functions [3]. The developed activation function, Smish, performed better than other common activation functions in classification tasks on open datasets. Wuraola has developed a family of activation functions that are to be used in embedded systems [4]. The proposed activation functions were shown to be computationally faster, and their use resulted in higher accuracy results than other common activation functions in recurrent neural networks and logistic regression models. Kaytan has introduced a new non-monotonic activation function capable of achieving higher results than other activation functions like Swish, Mish and others for image classification tasks [5]. Chai developed a new model based on LSTM capable of achieving higher accuracy for short-term PV generation forecasts [6]. The model uses a newly proposed activation function that helps solve the gradient disappearance problem and ensures a high accuracy of the prediction results for the task of short-term PV generation. There are also works in which the activation functions of the default implementation of model architectures are switched with other, alternative activation functions. Anami had performed experiments in which they had tried to compare prediction results by switching the default activation function with other different, common activation functions [7]. Wang has performed experiments in which they tried to use alternative activation functions in VGG16, ResNet50 and LeNet architectures, achieving superior results [8]. Essai Ali has tried to modify a LSTM by changing its' Tanh functions to different activation functions [9]. The author has achieved his aim of increasing the classification accuracy from 86% to 88% using the Weather Reports dataset, and from 93% to 97 % using the Japanese Vowels dataset.

### 3. Methodology

#### 3.1. Activation functions



**Figure 1.** A simple neural network with different activation functions per neuron

Let's review the concept displayed in Figure 1. In this example we have an input layer, hidden layer of 2 neurons and one output layer. Each neuron has a different function applied to it. Calculations for such a network is as follows:

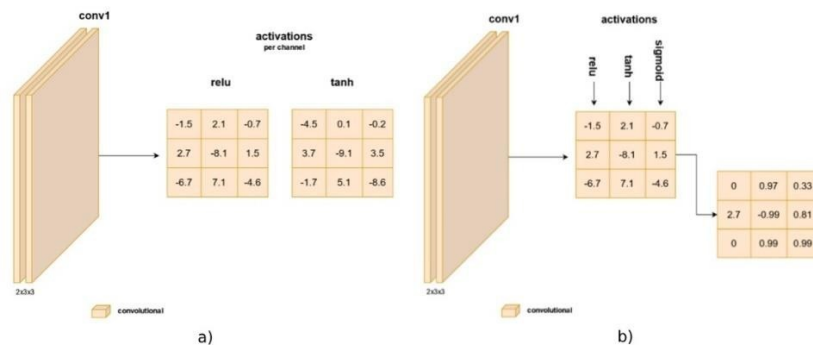
$$h_i = \sum_{j=1}^n w_{ij} \cdot x_j + b_i \quad (1)$$

$$z_1 = \text{relu}(h_1) \quad (2)$$

$$z_2 = \text{tanh}(h_2) \quad (3)$$

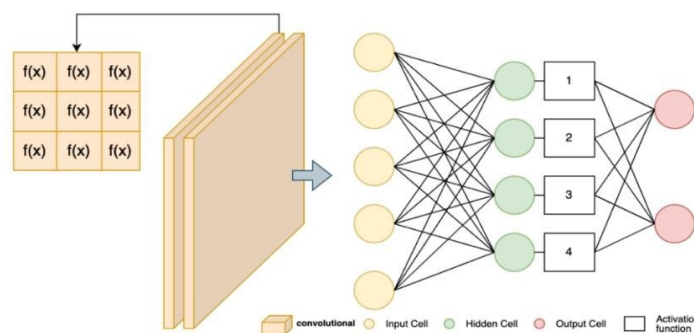
$$o_1 = z_1 w_{\text{relu}} + z_2 w_{\text{tanh}} \quad (4)$$

where  $h$  – hidden layers,  $w$  – weights,  $x$  – inputs,  $b$  – bias,  $z$  – activation function results and  $o$  – outputs. In an artificial convolutional neural network activations play a similar role, but because there are no actual neurons in a convolutional layer, different application is required. For the convolution layer 2 approaches were introduced.



**Figure 2.1.** Different activation functions per channel, 2.2. Different activation function for each matrix column.

In regular CNN architectures there is often only one activation function in a convolution layer. As displayed in the diagram Figure 2.1. different activation function can be applied to each channel after the convolution layer. Second diagram Figure 2.2. refers to another idea to apply multiple activation functions for each matrix column. In this case 3x3 matrix there are 3 columns in each channel. Every slice has a specific activation applied to it.



**Figure 3.** One activation for convolution layers and different activation functions in linear layer.

Some CNN architectures have a linear neuron layer which typically have only one activation function. The idea displayed on Figure 3 is to leave one activation in convolution layers and only have multiple activation functions in linear neuron layers, specifically an activation function for each neuron. As displayed in the diagram boxes (1-4) can each have a specific function assigned creating a spectrum of variations: (1-tanh, 2-relu, 3-sigmoid, 4-softmax), (1-relu, 2-tanh, 3-sigmoid, 4-relu) and so on.

For linear layers it is also possible to have a complete list of activation functions assigned. This idea is later experimented in this paper. Combinations of this list can be calculated as such. In this case 2 activation functions (ReLU, Tanh) power by 4 neurons equal to 16 variations:

$$v = e^n \quad (5)$$

where  $v$  – variations,  $e$  – elected activations and  $n$  – number of neurons.

It must also be noted that various activation functions can be used, and it is not limited to the most used activation functions such as ReLU, Tanh, Sigmoid, etc. The range of activation functions that were tested in this work are detailed in the experiments section.

## 3.2. Models

There has been a vast selection of CNN models proposed for image classification, a lot of those have complex implementations and long training hours. The models chosen for this paper are a low to mid- range complexity to test out the theory. Starting with SimpleNN, a simple neural network with one hidden layer of  $N$  neurons. TuNet – a CNN with 2 convolutions, 2 pooling layers and 3 linear layers [10]. AlexNet is a convolutional neural network (CNN) architecture that consists of five convolutional layers, three fully connected layers, and two pooling layers [10]. The convolutional layers extract features from the input images, while the pooling layers reduce the dimensionality of the feature maps. The fully connected layers learn a mapping from the extracted features to the output classes. Some of the key innovations introduced by AlexNet include the use of rectified linear unit (ReLU) activation functions, dropout regularization, and data augmentation techniques.

ResNet50 derives its name from its depth, incorporating 50 layers [11]. Notably, ResNet50 addresses the challenge of training deep networks by introducing residual connections that enable the direct flow of information across layers. This innovation mitigates the vanishing gradient problem, allowing for the successful training of extremely deep networks.

The architecture comprises building blocks known as residual blocks, each containing skip connections that bypass one or more layers. These skip connections facilitate the smooth propagation of gradients during backpropagation, enhancing the model's ability to capture intricate features. Additionally, ResNet50 employs batch normalization to accelerate training convergence and improve generalization performance.

UNet was used for image segmentation tasks [12]. It is a popular model with several modifications over the years [13,14,15]. The model has improved on the results of previous image segmentation models by its' architecture consisting of a contracting path used for capturing context and a symmetric expanding path used that enables precise localization [12]. The resulting architecture consists of 23 convolutional layers and the architecture utilizes the ReLU activation function. The model also heavily utilizes image augmentation, which enables it to achieve high accuracy without relying on many training images.

## 3.3. Datasets

### 3.3.1. Images

Several image datasets are popular for testing performance of CNN models. The CIFAR-100 is a dataset containing 60 000 32x32 color images with 100 classes (600 images per class). It is a subset of the Tiny Images dataset and is commonly used for fine-grained image classification [16]. The dataset contains a wide variety of images of objects, animals, and textures. The images are labeled with both fine-grained and coarse labels. The fine-grained labels correspond to the specific object or scene in the image, while the coarse labels correspond to the superclass of the object or scene.

The German Traffic Sign Benchmark is a multi-class, single-image classification challenge held at the International Joint Conference on Neural Networks (IJCNN) 2011 [17]. The following dataset includes 43 classes of traffic signs and more than 50,000 images.

Cityscapes dataset is a popular image segmentation dataset that consists of 25 000 such images captured from a moving vehicle [13,14,15]. The images were taken in different cities in Germany during different weather conditions. The dataset consists of 50 different classes. Each dataset item consists of a horizontally joined image, in which the left image is the original photograph, meanwhile the right image is the semantically segmented version of the image.

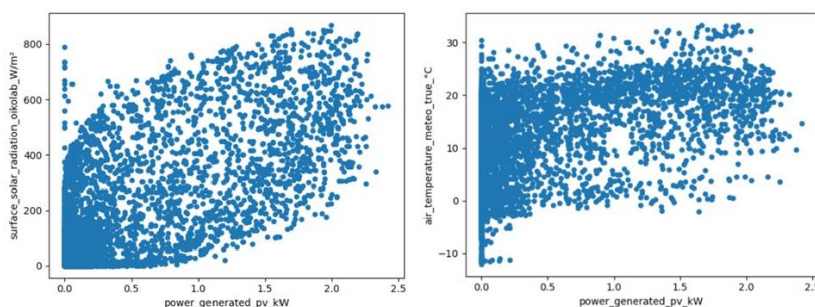
### 3.3.2. Tabular

Two tabular datasets were incorporated in this paper: breast cancer and iris flower classification. Breast cancer dataset features are computed from a digitized image of a fine needle aspirate (FNA) of a breast mass [18]. They describe characteristics of the cell nuclei present in the image. A few of the images can be found at <http://www.cs.wisc.edu/~street/images/>.

Iris flowers dataset is one of the earliest datasets used in literature on classification methods and widely used in statistics and machine learning [19]. The data set contains 3 classes of 50 instances each, where each class refers to a type of iris plant. One class is linearly separable from the other 2; the latter are not linearly separable from each other. When performing experiments, Obaid's work was used as a benchmark for the comparison of results [20].

### 3.3.3. Timeseries

Timeseries data for amazon stocks with stock price, closing price and other attributes was used [21]. Additionally, a custom photovoltaic (PV) panel generation dataset was used. The data consists of about a year of meteorological and PV generation data. The PV generation data was retrieved from a PV station in Kaunas, Lithuania, meanwhile the publicly available meteorological data was retrieved from Oikolab and from the Lithuanian Hydrometeorological Service. It was also attempted to include METAR data on cloud conditions at different altitudes, but utilizing this data did not provide any improvement to the results, so it was left out from the dataset. Based on the observed linear relationships between different meteorological features and PV generation, certain meteorological features were chosen to be used in the experiments (see Figure 4).



**Figure 4.** Scatter plots between PV generation data and surface solar radiation and air temperature.

As can be seen from the relationships between different features, a strong linear relationship between PV generation and air temperature, surface solar radiation has been observed. It was noted that using other meteorological data improved the results, although these features did not seem to have a linear relationship with the PV generation data. In total, the dataset consists of the following 11 features (see Table 1).

**Table 1.** Features used in the dataset, their data providers and measurement units

Feature name	Data provider	Measurement units
Generated power	-	kW
Air temperature	LHS	°C
Sea level pressure	LHS	hPa
Relative humidity	LHS	%
Wind speed	LHS	m/s
Wind gust speed	LHS	m/s
Is wind from north (true / false)	LHS	-
Is wind from south (true / false)	LHS	-
Is wind from west (true / false)	LHS	-
Surface solar radiation	Oikolab	W/m <sup>2</sup>
Total cloud cover	Oikolab	%

As it can be seen from the table, a wide range of different meteorological variables were used.

### 3.4. Environment

Google Collab environment with a single NVIDIA Tesla T4 GPU was used for experimentations of AlexNet and ResNet50 on CIFAR100. For GTSRB, UNet and LSTM experiments, the models were trained on two Tesla T4 GPU setup. Amazon stock close predictions were performed on a Kaggle provided CPU.

## 4. Experiments and results

### 4.1. Image classification

#### 4.1.1. CIFAR-100 with AlexNet

Inspired by Sharma’s work [22], we choose AlexNet as the primary target. Main reasons for choosing this architecture were that it had linear layers aside convolution blocks. We began experiments with the OriginalAlexNet implementation as a baseline with Tanh. Next, we experimented with changing only linear layers - changing one layer then changing both. The change was that instead of applying a single activation function, we applied 2 or 3 in cyclic order. The best results were with Tanh and Softmax combination of functions – 1.14% improvement in testing accuracy compared to the ReLU baseline, however, Tanh baseline was still more superior.

Later, we expanded experimentation with modifying Convolution Neural Network layers (CNN). Here implementation consisted of changing activation functions per channel. This showed marginally better results than the OriginalAlexNet with ReLU - 0.36% improvement.

For experimentation, hyper parameters were the following: learning rate – 0.0001, batch size – 256 and number of epochs – 40.

**Table 2.** Results from AlexNet experiments.

Training	Activations	Trainin g time min	Training accuracy	Validation accuracy	Testing accuracy
OriginalAlexNet	ReLU	34.75	81.209	36.64	36.95
OriginalAlexNetb	Tanh	26.11	84.216	43.060	<b>43.18</b>
AlexNetCustomLinear2a	Tanh,	35.03	81.473	37.84	37.21
	Softmax				
AlexNetCustomLinear2b	Tanh,	36.77	82.46	36.68	38.36
	Softmax				
AlexNetCustomLinear2r	random list	36.11	82.316	37.2	37.41
AlexNetCustomCNNa	Tanh,	35.76	82.427	37.32	37.31
	Softmax				
AlexNetCustomCNNb	Tanh,	35.73	81.502	36.62	37.31
	Softmax				
AlexNetCustomCNNr	random list	35.26	80.767	38.16	37.17

#### 4.1.2. CIFAR-100 with ResNet50

We have also investigated Residual networks block, using ResNet50 architecture (see Table 3). Hyperparameters used for the experiment: learning rate – 0.0001, batch size – 256 and number of epochs

– 12.

**Table 3.** Results from ResNet50 experimentations.

Training	Activations	Trainin g time, min	Trainin g accuracy	Validatio n accuracy	Testin g accuracy
----------	-------------	---------------------	--------------------	----------------------	-------------------

ResNet50	ReLU	12.66	78.907	44.06	44.33
ResNet50	Tanh	11.54	66.111	43.58	41.29
ResNet50Cus tomResiduala	ReLU, Tanh	70.56	81.193	43.02	42.61
ResNet50Cus tomResidualb	ReLU, SoftMax	78.33	76.469	42.44	43.32
ResNet50Cus tomResidualc	Tanh, Softmax, ReLU	78.1	81.342	43.66	<b>44.99</b>
ResNet50Cus tomResidualr	random list	75.25	72.767	40.52	41.76

As can be seen from results, only a combination of three functions - Tanh, Softmax and ReLU - managed to outperform baseline model with ReLU by 0.66% margin. Other combinations were below.

### 4.1.3 GTSRB with TuNet

Classifying images are pre-processed in the same manner and on the same training parameters as in the previous experiments, meanwhile the fixed size image is 32 by 32 pixels. The training parameters for TuNet are as follows: optimizer – Adam, learning rate – 0.001, loss function – cross entropy and batch size – 32. As can be seen in Table 4, the results of the TuNet baseline are generally worse than of the modified architecture:

**Table 4.** Results from TuNet experimentations.

Model	Activations	Epoch	Training time (1 epoch), ms	Training accuracy	Validation accuracy
TuNet (baseline)	Tanh	8	7007.23	0.9973	0.9834
TuNet	ReLU	10	7066.44	0.9721	0.9599
TuNetOnlyNN(Tanh)	ReLU, Tanh	10	16265.21	0.9990	<b>0.9863</b>
TuNetOnlyNN(Tanh)	Tanh, Softplus	9	18699.11	0.9961	0.9837
TuNetOnlyNN(Tanh)	ReLU, Tanh, Softplus	10	18615.31	0.9943	<b>0.9851</b>
TuNetOnlyNN(Tanh)	ReLU, Tanh, ELU	10	16559.02	0.9945	<b>0.9849</b>
TuNetPerNeuronAndChannel	ReLU, Tanh	8	18736.37	0.9945	0.9800
TuNetPerNeuronAndChannel	Tanh, Sigmoid	10	17864.02	0.9939	0.9809
TuNetPerNeuronAndChannel	Tanh, Softplus	10	21888.24	0.9929	0.9813
TuNetPerNeuronAndChannel	ReLU, Tanh, ELU	9	19664.91	0.9931	0.9836

In the table, several different models can be seen:

- TuNet – baseline model.
- TuNetOnlyNN – a model, where convolution has one activation function and neuron linear layers have specific activation function for each neuron.
- TuNetPerNeuronAndChannel – a model, where convolution layers have a specific activation function for each channel and a specific activation for each neuron in linear layer.

We can see a very slight improvement when different activations are applied to only the linear layer.

## 4.2. Cityscapes with UNet

For the image segmentation task, the popular Cityscapes dataset was chosen alongside the UNet model. The following parameters were the same for all the experiments using UNet: Adam optimizer with a learning rate of 0.001, the mean-squared error as the loss function, a batch size of 4 and 20 as the number of epochs for training.

As it can be seen from the results of the experiments, a significant Dice metric increase of about 10% was achieved by various activation function combinations (see Table 5).

**Table 5. Results from UNet experimentations**

Model	Activations	Epoch s	Trainin g time, ms	Train. dice	Valid. dice
UNet	ReLU	10	1378448.12	0.4700	0.4062
UNet	Tanh	10	1380602.75	0.4680	0.4334
UNetPerNeuron	ReLU, Tanh	10	4429268.50	0.4747	0.4293
UNetPerNeuron	Tanh, ReLU	10	4430903.50	0.4656	0.4884
UNetPerNeuron	Tanh, Softmax	10	4487534.50	0.3716	0.3389
UNetPerNeuronAnd Channel	ReLU, Tanh	10	4487183.00	0.4714	0.5013
UNetPerNeuronAnd Channel	ReLU, Softmax	10	4600614.00	0.3733	0.4442
UNetPerNeuronAnd Channel	Tanh, Softmax	10	4539303.00	0.3696	0.4242
UNetPerNeuronAnd Channel	Tanh, Softplus	10	4526773.00	0.4697	0.4453
UNetPerNeuronAnd Channel	Tanh, Softplus	8	3621696.25	0.4685	0.4958
UNetPerNeuronAnd Channel	Tanh, ReLU, Softplus	10	4516755.50	0.4709	0.4468
UNetPerNeuronAnd Channel	Tanh, ReLU, Softplus	9	4065430.75	0.4700	0.5081
UNetPerNeuronAnd Channel	ReLU, Tanh, ELU	10	4525098.50	0.4696	0.4339
UNetPerNeuronAnd Channel	ReLU, Tanh, ELU	7	3169012.25	0.4646	0.4654

As can be seen from the table, using almost any combinations of activation functions can result in better prediction results in the case of UNet. It is also observed that even changing the activation in the baseline model from ReLU to Tanh has improved the results by a significant amount as well.

### 4.3 Time series regression/forecasting

#### 4.3.1 Simple NN on Amazon stock prediction

Experiments were performed on Amazon stock timeseries data predict the closing price for the next day. An architecture named SimpleNN was used. It is a neural network with 1 input cells, 14 hidden layer cells and 1 output. The following parameters were used in the experiment: optimizer – Adam, learning rate – 0.001, loss function – mean-squared error, batch size – 16, lag values – 7 and number of training epochs – 5.

The experiment compares the same model and its architecture, the only difference is activations per neuron and one activation for the whole network (see Table 6).

**Table 6. Testing results of SimpleNN and PerNeuron models.**

Model	Activations	MAE	RMSE	RMSLE
SimpleNN	ReLU (baseline)	2.8582	3.7894	0.0312
SimpleNN	Tanh	2.8583	3.9185	0.0316
PerNeuron	Tanh, ReLU	3.0003	4.0790	0.0332
PerNeuron	ReLU, Tanh	3.0899	4.1825	0.0343
PerNeuron	ReLU, ReLU, Sigmoid	2.7314	3.6951	0.0301



PerNeuron	ReLU, Softmax	2.9816	3.9862	0.0323
<b>PerNeuronList</b>	<b>ReLU, ReLU, ReLU, ReLU, ReLU,</b> <b>Sigmoid, ReLU, ReLU, Sigmoid, ReLU,</b> <b>ReLU, Sigmoid, ReLU, Sigmoid</b>	<b>2.6980</b>	<b>3.6736</b>	<b>0.0298</b>

Additionally, all possible combinations of different activation functions sets have been tested (see model *PerNeuronList*).

As can be seen from the results, there is an increase in accuracy in certain cases, and it can also be observed that finding the best possible set of activation functions yielded the best results out of the experiments.

### 4.3.2 Custom PV dataset with LSTM

Experiments were performed using a time-series dataset for forecasting PV generation. An LSTM model was used, as it is often utilized for solving PV generation forecast tasks [23,24,25,26,27]. For performing the forecasts, the output of the previous step is used as the input of the following training step. The following parameters were used for the experiments: Adam optimizer with a learning rate of 0.001, mean-squared error for the error metric, a batch size of 8, 12 lag values for the PV data, and 20 training epochs.

The parameters for the experiments were chosen based on experiments performed using different sets of parameters. The batch size refers to the number of predictions retrieved from the model output and the lag values refers to the number of previous predictions to use as input of the next prediction. Based on tests using different lag values, a value of 12 was noticed to be one of the best values for this parameter, although this parameter did not seem to have much impact on the accuracy of predictions. Regarding transformations of data, the training data has been standardized so that the ranges of values would be the same for all features.

**Table 7. Results from UNet experimentations**

Model	Activations	Epochs	Training MAE	Test MAE	Test RMSE	Test RMSLE	Time (ms)
LSTM	Default (Tanh, Sigmoid)	20	0.0563	0.0757	0.1262	0.070	197461.00
LSTM	Tanh, Softmax	20	0.0565	0.0867	0.1412	0.0806	3275714.00
LSTM	ELU, Sigmoid	20	0.2056	0.2113	0.2882	0.1734	3259991.50
LSTM	Sigmoid, ELU	20	0.1792	0.1863	0.2516	0.1727	3271329.50
LSTM	Sigmoid, Tanh	20	0.0533	0.0857	0.1420	0.0783	3096815.50
LSTM	Sigmoid, Tanh	8	0.0693	0.0782	0.1305	0.0721	1248338.12
LSTM	Sigmoid, Softmax	20	0.0740	0.0798	0.1317	0.0741	3708114.00
LSTM	ELU, Sigmoid, Tanh	20	0.1748	0.1823	0.2469	0.1615	2843427.75
LSTM	ELU, Tanh, Sigmoid	20	0.1817	0.2553	0.1796	0.1542	2818499.50
LSTM	Softmax, Sigmoid, Tanh	20	0.0606	0.0791	0.1315	0.0726	3155361.75
LSTM	Softmax, Tanh, Sigmoid	20	0.0604	0.0814	0.1331	0.0756	3171810.00

As can be seen from Table 7, there is no significant improvement based on testing RMSLE. Although many experiments yielded similar results to the baseline, there was not a single experiment which yielded better results than the baseline. It can also be observed that an increase in the number of different activation functions used does not improve the forecast results either.

## 4.4 Tabular

Tabular data is still widely used in machine learning tasks. In this paper we choose two datasets to experiment with the changes on Iris flowers and Breast cancer classifications. Both experiments have the following training parameters: optimizer – SGD, learning rate – 0.01, loss function – cross entropy loss and number of training epochs – 200.

From results displayed in Table 8 comparing one activation versus multiple for this Iris flowers classification task, there is no improvement compared to best suited activation function.

**Table 8.** Iris flower results of SimpleNN vs PerNeuron models. Both models architecture (4 input cells, 6 hidden cells, 3 output cells)

Model (Iris)	Activations	Test Accuracy
SimpleNN	Tanh (baseline)	0.93
SimpleNN	Relu	0.70
PerNeuron	Relu, Tanh	0.90
PerNeuron	Softmax, ELU	0.93
PerNeuron	Tanh, Sigmoid, Softmax, Softplus	0.93
PerNeuron	Tanh, Sigmoid, Softmax, ELU	0.93
PerNeuron	Tanh, Sigmoid, Softmax, Softplus, ELU	0.93

Experiments performed on breast cancer dataset can be visible in Table 9. After training testing results, can be viewed in the table below. As we can see there is slight improvement with model having multiple activation functions.

**Table 9.** Breast cancer results of SimpleNN vs PerNeuron models. Both models architecture (30 input cells, 20 hidden cells, 2 output cells).

Model	Activations	Test Accuracy
SimpleNN	Tanh (baseline)	0.9649
SimpleNN	Relu	0.9649
PerNeuron	ReLU, Tanh	0.9739
PerNeuron	ReLU, ELU - less epoch (150/200)	0.9739
PerNeuron	Sigmoid, ELU	0.9739
PerNeuron	ReLU, Tanh, Sigmoid	0.9739
PerNeuron	ReLU, Tanh, Softmax - less epoch (150/200)	0.9739
PerNeuron	ReLU, Tanh, Softplus	0.9739
PerNeuron	ReLU, Tanh, Sigmoid, ELU	0.9739
PerNeuron	ReLU, Tanh, Softmax, ELU - less epoch (150/200)	0.9739
PerNeuron	ReLU, Sigmoid, Softmax, ELU	0.9739
PerNeuron	Tanh, Sigmoid, Softmax, Softplus	0.9739
PerNeuron	Tanh, Softmax, Softplus, ELU	0.9739
PerNeuron	ReLU, Tanh, Sigmoid, Softmax, Softplus	0.9739
<b>PerNeuronList</b>	<b>ReLU, ReLU, ReLU, ReLU, ReLU, Tanh, Tanh, ReLU, Tanh, ReLU, ReLU, Tanh</b>	<b>0.9825</b>

Additionally, a activation function set from a large number of combinations was selected and the accuracy using it is better compared to one activation function (see Table 10).

**Table 10.** Breast cancer results of SimpleNN using all possible activation function combinations with 12 neurons in a hidden layer.

Model	Activations	Test Accuracy
PerNeuron	ReLU, ReLU, ReLU, ReLU, ReLU, Tanh, Tanh, ReLU, Tanh, ReLU, ReLU, Tanh	0.9825

It should also be noted that better results were achieved than from the SVM described in Obaid's work. As can be seen from the results, there is a significant accuracy increase for the PerNeuron models, whilst the most significant increase can be seen when finding the best activation function list from all possible combinations.

## 5. Conclusions and discussion

The research paper explores the concept of using multiple activation functions in artificial neural networks. It discusses the role of activation functions in introducing non-linear relations to improve the accuracy of tasks. The paper investigates different approaches to incorporating multiple activation functions, including assigning a different function to each neuron or channel.

The experiments included using models such as AlexNet, ResNet50, TuNet, and SimpleNN. In the AlexNet experiment, different activation function combinations were tested in both linear layers and convolutional neural network (CNN) layers. The results showed that using Original AlexNet with Tanh activation function yielded the best overall performance. The ResNet50 experiments resulted in one combination performing marginally better than any of single function baselines. The TuNet and SimpleNN experiments aimed to evaluate the performance of these specific architectures on their respective datasets. Overall, the experiments provided insights into the impact of activation function combinations on model performance, with modest improvements observed compared to using a single activation function. The datasets used in the experiments included CIFAR-100, GTSRB, Breast Cancer Wisconsin (Diagnostic), Iris flowers, and Amazon stocks. In image segmentation tasks, modifying the UNet architecture with different activation function combinations leads to significant improvements in the Dice metric. Even changing the activation function in the baseline model from ReLU to Tanh shows improved results. For time series regression/forecasting tasks, the experiments show that using multiple activation functions does not significantly improve the accuracy of predictions. This paper also hints into an idea of full list of activation functions, which would learn relation with the specific data neuron is receiving. An idea which requires further analysis.

Overall, the paper concludes that while using multiple activation functions can have some benefits in certain scenarios, the improvements are not substantial compared to using a single activation function. The choice of activation function should be based on the specific task, dataset and its features.

## References

- [1] S.R. Dubey, S.K. Singh, B.B. Chaudhuri, Activation functions in deep learning: A comprehensive survey and benchmark, *Neurocomputing (Amsterdam)*. 503 (2022) 92–108. <https://doi.org/10.1016/j.neucom.2022.06.111>.
- [2] Y. Yu, K. Adu, N. Tashi, P. Anokye, X. Wang, M.A. Ayidzoe, RMAF: Relu-Memristor-Like Activation Function for Deep Learning, *IEEE Access*. 8 (2020) 72727–72741. <https://doi.org/10.1109/ACCESS.2020.2987829>.
- [3] X. Wang, H. Ren, A. Wang, Smish: A Novel Activation Function for Deep Learning Methods, *Electronics (Basel)*. 11 (2022) 540-. <https://doi.org/10.3390/electronics11040540>.
- [4] A. Wuraola, N. Patel, S.K. Nguang, Efficient activation functions for embedded inference engines, *Neurocomputing*. 442 (2021) 73–88. <https://doi.org/10.1016/j.neucom.2021.02.030>.
- [5] M. Kaytan, İ.B. Aydilek, C. Yeroğlu, Gish: a novel activation function for image classification, *Neural Comput & Applic*. 35 (2023) 24259–24281. <https://doi.org/10.1007/s00521-023-09035-5>.
- [6] M. Chai, F. Xia, S. Hao, D. Peng, C. Cui, W. Liu, PV Power Prediction Based on LSTM With Adaptive Hyperparameter Adjustment, *IEEE Access*. 7 (2019) 115473–115486. <https://doi.org/10.1109/ACCESS.2019.2936597>.
- [7] B.S. Anami, C.V. Sagarnal, Influence of Different Activation Functions on Deep Learning Models in Indoor Scene Images Classification, *Pattern Recognition and Image Analysis*. 32 (2022) 78–88. <https://doi.org/10.1134/S1054661821040039>.
- [8] W. Hao, W. Yizhou, L. Yaqin, S. Zhili, The Role of Activation Function in CNN, in: 2020 2nd International Conference on Information Technology and Computer Application (ITCA), 2020: pp. 429–432. <https://doi.org/10.1109/ITCA52113.2020.00096>.
- [9] M.H. Essai Ali, A.B. Abdel-Raman, E.A. Badry, Developing Novel Activation Functions Based Deep Learning LSTM for Classification, *IEEE Access*. 10 (2022) 97259–97275. <https://doi.org/10.1109/ACCESS.2022.3205774>.
- [10] A. Krizhevsky, I. Sutskever, G. Hinton, ImageNet Classification with Deep Convolutional Neural Networks, *Neural Information Processing Systems*. 25 (2012).

- <https://doi.org/10.1145/3065386>.
- [11] K. He, X. Zhang, S. Ren, J. Sun, Deep Residual Learning for Image Recognition, (2015). <https://doi.org/10.48550/arXiv.1512.03385>.
  - [12] O. Ronneberger, P. Fischer, T. Brox, U-Net: Convolutional Networks for Biomedical Image Segmentation, (2015). <https://doi.org/10.48550/arXiv.1505.04597>.
  - [13] H. Bai, L. Liu, Q. Han, Y. Zhao, Y. Zhao, A novel UNet segmentation method based on deep learning for preferential flow in soil, *Soil & Tillage Research*. 233 (2023) 105792-. <https://doi.org/10.1016/j.still.2023.105792>.
  - [14] K.K. Wong, A. Zhang, K. Yang, S. Wu, D.N. Ghista, GCW-UNet segmentation of cardiac magnetic resonance images for evaluation of left atrial enlargement, *Computer Methods and Programs in Biomedicine*. 221 (2022) 106915–106915. <https://doi.org/10.1016/j.cmpb.2022.106915>.
  - [15] G. Rani, P. Thakkar, A. Verma, V. Mehta, R. Chavan, V.S. Dhaka, R.K. Sharma, E. Vocaturo, E. Zumpano, KUB-UNet: Segmentation of Organs of Urinary System from a KUB X-ray Image, *Computer Methods and Programs in Biomedicine*. 224 (2022) 107031–107031. <https://doi.org/10.1016/j.cmpb.2022.107031>.
  - [16] A. Krizhevsky, Learning Multiple Layers of Features from Tiny Images, in: 2009. <https://www.semanticscholar.org/paper/Learning-Multiple-Layers-of-Features-from-Tiny-Krizhevsky/5d90f06bb70a0a3dced62413346235c02b1aa086> (accessed January 17, 2024).
  - [17] J. Stallkamp, M. Schlipsing, J. Salmen, C. Igel, Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition, *Neural Networks*. 32 (2012) 323–332. <https://doi.org/10.1016/j.neunet.2012.02.016>.
  - [18] W.N. Street, W.H. Wolberg, O.L. Mangasarian, Nuclear feature extraction for breast tumor diagnosis, in: R.S. Acharya, D.B. Goldgof (Eds.), San Jose, CA, 1993: pp. 861–870. <https://doi.org/10.1117/12.148698>.
  - [19] A. Unwin, K. Kleinman, The Iris Data Set: In Search of the Source of *Virginica*, *Significance*. 18 (2021) 26–29. <https://doi.org/10.1111/1740-9713.01589>.
  - [20] O. Ibrahim Obaid, M. Mohammed, M.K. Abd Ghani, S. Mostafa, F. Al-Dhief, Evaluating the Performance of Machine Learning Techniques in the Classification of Wisconsin Breast Cancer, *International Journal of Engineering and Technology*. 7 (2018) 160–166. <https://doi.org/10.14419/ijet.v7i4.36.23737>.
  - [21] Amazon, Inc., Amazon.com, Inc. (AMZN) Stock Historical Prices & Data - Yahoo Finance, Amazon. (2024). <https://finance.yahoo.com/quote/AMZN/history/> (accessed January 17, 2024).
  - [22] N. Sharma, V. Jain, A. Mishra, An Analysis Of Convolutional Neural Networks For Image Classification, *Procedia Computer Science*. 132 (2018) 377–384. <https://doi.org/10.1016/j.procs.2018.05.198>.
  - [23] T. Limouni, R. Yaagoubi, K. Bouziane, K. Guissi, E.H. Baali, Accurate one step and multistep forecasting of very short-term PV power using LSTM-TCN model, *Renewable Energy*. 205 (2023) 1010–1024. <https://doi.org/10.1016/j.renene.2023.01.118>.
  - [24] L. Wang, M. Mao, J. Xie, Z. Liao, H. Zhang, H. Li, Accurate solar PV power prediction interval method based on frequency-domain decomposition and LSTM model, *Energy (Oxford)*. 262 (2023) 125592-. <https://doi.org/10.1016/j.energy.2022.125592>.
  - [25] X. Huang, Q. Li, Y. Tai, Z. Chen, J. Liu, J. Shi, W. Liu, Time series forecasting for hourly photovoltaic power using conditional generative adversarial network and Bi-LSTM, *Energy (Oxford)*. 246 (2022) 123403-. <https://doi.org/10.1016/j.energy.2022.123403>.
  - [26] H. Gao, S. Qiu, J. Fang, N. Ma, J. Wang, K. Cheng, H. Wang, Y. Zhu, D. Hu, H. Liu, J. Wang, Short-Term Prediction of PV Power Based on Combined Modal Decomposition and NARX-LSTM-LightGBM, *Sustainability (Basel, Switzerland)*. 15 (2023) 8266-. <https://doi.org/10.3390/su15108266>.
  - [27] J. Ospina, A. Newaz, M.O. Faruque, Forecasting of PV plant output using hybrid wavelet-based LSTM-DNN structure model, *IET Renewable Power Generation*. 13 (2019) 1087–1095. <https://doi.org/10.1049/iet-rpg.2018.5779>.