VARIANTS OF NEURAL NETWORKS: A REVIEW

Bahera H. Nayef [1]***, Siti Norul Huda Sheikh Abdullah*** [2]***, Rossilawati Sulaiman*** [3]***, Zaid Abdi Al Kareem Alyasseri*** [4]

[1,2,3,4]Faculty of Information Science and Technology, Universiti Kebangsaan Malaysia, 43600 Bangi, Selangor, Malaysia

[1]Radiology Techniques Department, Ibn Khaldun University College, Iraq

[4]Information Technology Research and Development Center (ITRDC), University of Kufa, Najaf, Iraq

Email: bahera.hani@ik.edu.iq[1*](corresponding author), snhsabdullah@ukm.edu.my[2], rossilawati@ukm.edu.my[3], zaid.alyasseri@uokufa.edu.iq[4]

DOI: https://doi.org/10.22452/mjcs.vol35no2.5

ABSTRACT

Machine learning (ML) techniques are part of artificial intelligence. ML involves imitating human behavior in solving different problems, such as object detection, text handwriting recognition, and image classification. Several techniques can be used in machine learning, such as Neural Networks (NN). The expansion in information technology enables researchers to collect large amounts of various data types. The challenging issue is to uncover neural network parameters suitable for object detection problems. Therefore, this paper presents a literature review of the latest proposed and developed components in neural network techniques to cope with different sizes and data types. A brief discussion is also introduced to demonstrate the different types of neural network parameters, such as activation functions, loss functions, and regularization methods. Moreover, this paper also uncovers parameter optimization methods and hyperparameters of the model, such as weight, the learning rate, and the number of iterations. From the literature, it is notable that choosing the activation function, loss function, number of neural network layers, and data size is the major factor affecting NN performance. Additionally, utilizing deep learning NN resulted in a significant improvement in model performance for a variety of issues, which became the researcher's attention.

Keywords: Artificial intelligence, Machine learning, Object Detection, Image Classification, Optimizing Parameters

1.0 INTRODUCTION

Artificial intelligence science is a field of computer science that mimics humans' behavior in solving problems that learn from predefined collected databases with mathematical rules and operations. Machine Learning (ML) is a part of artificial intelligence that uses algorithms to find patterns in predefined data and make predictions for later processing of undefined events [1, 2]. On top of that, ML algorithms can either be supervised learning, unsupervised learning, semi-supervised learning, or reinforcement learning, as shown in Table 1. The dataset consists of a set of labeled instances. Each instance *(x)* includes a set of independent attributes (*y*). Therefore, an instance can be represented as *(x,y)*. Both the instances and attributes could be continuous or discrete values. In the case of supervised algorithms, *y* represents the target attribute (data classification). In unsupervised algorithms, there is no target attribute (clustering data). Hence, the dataset is divided into clusters according to their similarities [3]. While the semi-supervised is the in-between case where most of the labels are present, but some are not. Contrarily, reinforcement learning is based on learning by interaction with the environment [4].

Table 1: Types of machine learning algorithms. Source: [5]

Supervised ML algorithms	Unsupervised ML algorithms	Reinforcement Learning algorithms
Support vector machine	K-means	Dynamic programming
Decision tree	Mean shift	Monte Carlo methods
Linear regression	Affinity propagation	Temporal difference
Logistic regression	Hierarchical clustering	
Naïve Bayes	Gaussian mixture modeling	
K-nearest neighbor	Markov random field	
Random forest	Iterative self-organizing data	
AdaBoost	Fuzzy C-means systems	
Neural networks	Density-based spatial clustering applications	

Neural networks can be categorized into feedforward neural networks or recurrent neural networks. The information flows straightforwardly from the input layer to the output layer via several hidden layers in the first type. The information enables to traverse forward and backward in recurrent or feedback neural networks to pass several interconnections [6]. The applications of Neural networks have been expanded in recent decades to various domains such as Medical imaging using MRI and CT scan images, credit risk analysis, customer profiling, market segmentation, targeted marketing retail management, fraud detection, smart production, Image recognition, voice recognition, handwriting recognition, biometric systems as in applications and age assessment [6-8]. Any NN with three layers and more forms a deep neural network.

Neural networks tolerate various data types such as text from images, audio signals, and video. In addition to :

1- Different dataset sizes,
2- Binary and multiple numbers of classes,
3- Noisy data with outliers,
4- Accepts raw data and preprocessed datasets,
5- Allows deep learning to extract the best feature maps, and
6- Allows dropping features with low probabilities with a predefine dropout rate.

Some terms and definitions used in machine learning systems must be learned to understand neural networks, as shown in Table 2.

Table 2: Neural network terminology. Source: [5]

No.	Term	Definition
1	Classification	Is the process of assigning a class or a label to a set of input vectors (or pixels)?
2	Model	The machine learning system creates a model by learning a set of weights and decision rules. This model is used to evaluate and test examples of anonymous data.
3	Labeled data	A dataset contains its ground truth labels.
4	Training set	A process to update the weights and the decision rules of the labeled data using machine learning. It stops updating when the local optima have attained and no more improvement in the performance.
5	Validation set	A dataset uses in the training phase.
6	Testing set	An unseen test dataset to justify the performance confidence of the machine algorithm with real data.
7	Node	A neural network unit. It accepts input units with an activation function.
8	Activation function	It is a function that sums the inputs and uses a threshold to generate the outputs.
9	Loss function	It is a function that calculates the difference between the true and the predicted output of the NN

10	Layers	A set of nodes that calculate at each layer. Three types of layers are input layer, hidden layer, and output layers
11	Weights	Each feature in the input layer multiplies with a small real parameter value or weight. These weights later are updated during the training step to build the best model.
12	Segmentation	An input image divides into several parts that illustrate the available object of the input image.
13	Feature	A set of numerical values compounded with input examples such as pixel values, a person's age, length, etc.
14	Overfitting	A classifier is familiar with a specific type of training data but less flexible to additional data.
15	Regularization	The change is made to the weight range of the fully-connected layers to enhance the overall classification accuracy, and the best technique is the dropout.
16	Dropout	A regularization technique addresses the overfitting problem by randomly setting some nodes (such as 50% of the nodes) weights to zero value.

The use of neural networks has expanded exponentially in the last few years. Depending on the cited research in this paper, a small survey is conducted to demonstrate the most NN interested fields. Table 3 and Fig.1 show the number of researches on developing and improving Activation functions, which got a large percentage (48%) in comparison to loss functions (38%) researches, overcoming the overfit and underfit problems (8%) and regularization (8%). From these percentages, it is clear that most of the research is conducted on either developing or enhancing activation functions or loss functions. The total number of research papers used in this statistics is 50 using searching keywords such as activation functions, Relu, loss functions, NN, Convolution Neural network, NNs overfit and underfit, regularization, and data augmentation. Google scholar and research gates were used for finding and downloading these papers.

Table 3: The number of researches in NN fields from the year 2005 until 2020

Year	Loss Functions	Activation Functions	Over / Underfitting	Regularization
2020	2	-		
2019	1	7	4	3
2018	5	3		1
2017	1	3	-	-
2016	0	4	-	-
2015	1	3	-	-
2014	4	1	-	-
2013	4	1	-	-
2009	1	1	-	-
2005	1	1	-	-
Total	18	24	4	4

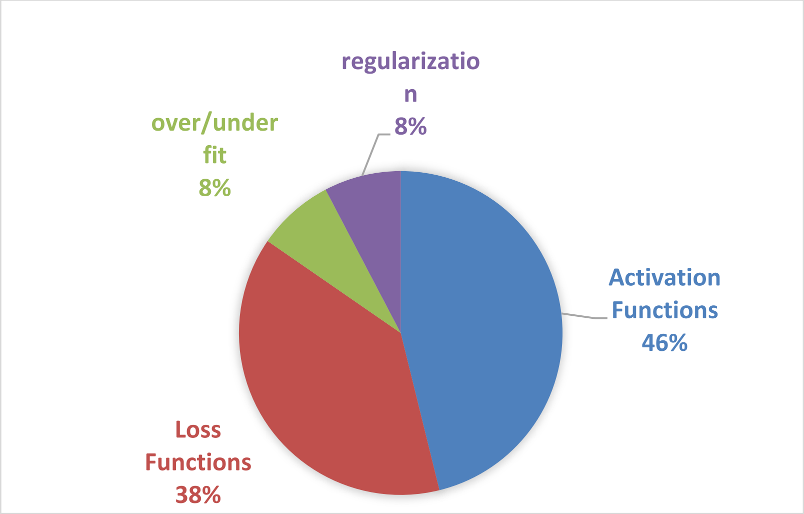


Fig. 1: The research fields related to the activation function, loss function, over/underfitting, and regularization in percentages from 2005-2020.

Neural Networks encompass a large number of methods and techniques. These methods and techniques enable the machine to learn patterns from data with less human cooperation. It depends on human knowledge to work efficiently via transfer learning, feature selection, and multitasking. So, this paper attempts to assemble and present all the fundamentals, components, and techniques of neural networks in short notes. It motivates the researcher to pursue a new research direction to enrich their knowledge about the aspects of NN. There are a few research questions that need to uncover and responded them in this paper, such as:

1) What is the NN?
2) Which activation function is better than the others?
3) How to choose an activation function?
4) What are the different types of loss functions?
5) What is data augmentation?

This paper is organized as follows: SECTION 2 addresses the NN structure, activation functions, and loss function, SECTION 3 is related to NN optimization, and SECTION 4 discusses the regularization technique. Then, the related work is in SECTION 5.

2.0 NEURAL NETWORK (NN) OVERVIEW

Over many years, humans have tried to build intelligent machine systems that simulate the human brain in solving problems. The NN history began with the attempt of Marvin Minsky on artificial intelligence research in solving the exclusive OR (XOR) function [9] that became the base for solving more complicated problems with a larger size of data in conjunction with the development of computational processing and storage units.

The NN structure resembles the biological human brain structure. This structure enables the computer to acquire human knowledge in solving problems. It provides magnificent solutions to various problems such as character recognition [10], image recognition [11], handwriting recognition [12, 13], speech recognition [14, 15], and natural language processing [9].

2.1 Neural Network Structure

A simple supervised NN comprises three layers: the input, hidden, and output layers. The input layer has several input neurons (perceptron) (x_1, $x_{2,}$ $x_{3,}$ $x_{4,\ \dots,}$ $x_{n),}$ and the number depends on the dataset required in the learning process. The output layer contains several output neurons (or one neuron), representing a class of the output dataset. Between the input and output layers, a hidden layer is also made up of several hidden neurons. The NN could contain one or more hidden layers, as shown in Fig. 2. According to the given weight, all the input layer neurons are evaluated, a real value (w_1,w_2,w_3,…w_n). All the neurons in the input layer are fully connected to the hidden layer's neurons. The computational processing of the data in the hidden layer is very complex and unknown to us.

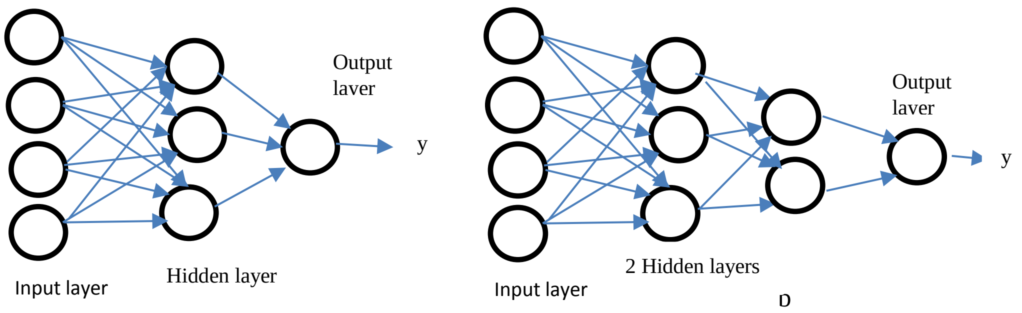


Fig. 2 The neural network structure : (a) contains one input layer, one hidden layer, and one output layer. (b) one input layer, two hidden layers, and one output layer.

NN calculates the output y by summing up the multiplication of the weight vectors (w_i) and the input vectors (x_i), and a predefined threshold, t determines the output or the predicted value, y. The mathematical representation is shown in Eq.1 [16].

$$y_j = \begin{cases} 1 & if \quad \sum_i w_i x_i > t, \\ 0 & if \quad \sum_i w_i x_i \leq t. \end{cases} \tag{1}$$

The threshold condition is replaced with a constant called bias (b) which is equivalent to (-threshold, t), so Eq.2 becomes as follows:

$$y_j = \begin{cases} 1 & if \quad \sum_i w_i x_i + b > 0, \\ 0 & if \quad \sum_i w_i x_i + \mathrm{b} \leq 0. \end{cases} \tag{2}$$

When the bias value $b > 0$ then the output is 1 and when $b \leq 0$, then the output, y_j is 0. Any small changes in the weight or bias will change the output and result in a closer outcome to the desired behavior.

2.2 Activation Function

The activation function or transfer function is set in between the NN layers. The prediction quality depends on the nonlinear or linear activation function used in the NN structure. This prediction quality can increase the strength of a multilayer NN

The activation function (AF) calculates the summation of the weight and bias of a neuron. The summation result helps decide whether the neuron is valuable or invaluable to the NN The AF can also update data using the gradient descent or stochastic gradient descent algorithm to create an output of the NN [17]. Regardless of using an activation function, NN becomes a linear regression model. The predicted output is like the fed input data accompanied by an error. The linear activation function adapts effectively if the input produces a linear change [18]. The NN uses the nonlinear activation function for real problems since the network cannot learn from errors[18]. Moreover, AF acts as a regulator for the resulting output in various domains such as speech recognition, image detection, and classification [19, 20], biometric detection, and recognition [21, 22], and other researchers. More discussion in the next subsection about function types.

2.2.1 Linearly Act Of Nonlinear Activation Function

This section discusses some nonlinear activation functions with linearity properties, such as the Relu versions. These are derivative functions and can perform backpropagation to update weights and biases.

1. Rectified linear unit (Relu)

The rectified linear unit is one of the most popular in DNN because it is easy to optimize due to its linear similarity to the linear units [23]. Relu acts as a linear function because it approximates the negative values into zero and preserves the positive values [24]. The Relu sparsity characteristic has motivated the researchers to prefer Relu over the sigmoid activation function. Besides that, it has various other advantages such as:

a) Relu function is non-saturating and does not have the fading gradient issue as encountered with other deep learning neural network architectures [25].
b) It ensures faster computation since it does not involve the computation of exponentials and divisions [17].

For the above reasons, the researchers have conducted various studies [26] in developing and enhancing Relu and have therefore developed Leaky Relu, PRelu, Elu, and many other updated versions of Relu. This section presents an abstract on the use of versions of the Relu. The Relu output is equal to zero in its half active domain, which maintains large and consistent derivatives while active. The output of the Relu operation is 0 for most of the second derivatives and is equal to 1 when the Relu operation is active [27]. The Rectified Linear Unit (Relu) function keeps all x≤0 equal to 0 and the slope above 1. The function formula is shown in Eq. 3 and Fig.3[9].

$$z_j = f(x_j) = Max\ (0, x_j) \tag{3}$$

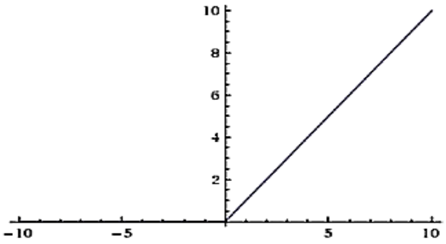


Fig. 3: Relu function. Source: [9]

2. Leaky Relu

The Leaky Relu (Fig.4) has proposed a slight improvement in its function compared to Relu by reducing the vanishing gradient. This is achieved by approximating the slope for the x value, which is less than 0 to be (-0.01). It has introduced good performance but has limited uses [26].

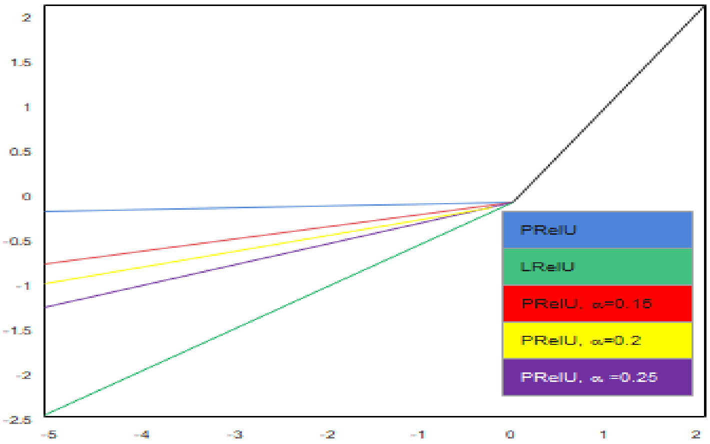


Fig. 4: LRelu, PRelu functions with different α values

Source: [28]

Apart from the function shown above, numerous other activation functions are derived from the same function, including the following:

a) Parameter Relu is a generalized Leaky Relu that allows learning negative slopes [29].

b) The *maxout* function works by using more parameters and more than one input vector. It takes a maximum of two inputs, three inputs, and four inputs. According to [30], the max-out function can approximate any convex function.
c) Panelized *tanh* is similar to LRelu [24].
d) Cubic (x^3) developed by [31] is used in MLP.
e) Selu is an updated Elu used in self-normalizing nets [32].

3. SoftMax activation function

The output vector is the probability distribution between certain labels (classes). The summation of all label probabilities is equal to 1. The probability distribution shows the level of confidence for each label. The label with a probability close to 1 represents the output label or the predicted class [33]. All softmax neurons are normalized using the Softmax function shown in Eq. 4.

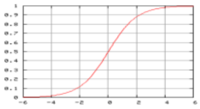
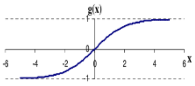
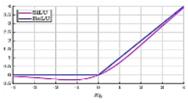
$$softmax(x_i) = \frac{e^{x_i}}{\sum_j^n e^{x_j}} \tag{4}$$

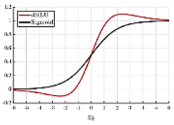
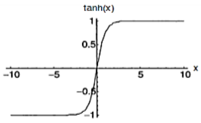
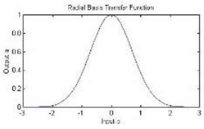
Based on the discussion of the above activation functions, a question has been raised. Which activation function is better? The answer is that it depends on the type of problem to be solved and the used datasets.

2.2.2 Nonlinear Activation Function

The nonlinear characteristic which identifies the nonlinear activation function is that it has rounded margins. Any nonlinear variation in the input leads to a nonlinear change in the output. In this way, an NN can learn more complicated input features dealing with rounded margins. A nonlinear activation function is a good simulation of real-life data. This rounded mechanism reflects that real-life data are nonlinear [18]. Some of The nonlinear activation functions are presented in Table 4.

Table 4: Nonlinear activation functions formula, representation, and description

Activation function	Function formula	Function representation diagram	Description
A unipolar sigmoid Activation function	$g(x) = \frac{1}{1 + e^{-x}}$ Source: [34]	 [34]	4. Works with backpropagation NN 5. Performs easy way to distinguish different patterns 6. Reduces the computational time. 7. Good performance with deep learning. 8. Good for binary output. Source: [33]
Vehbi Bipolar sigmoid Activation function	$g(x) = \frac{1 - e^{-x}}{1 + e^{-x}}$ Source: [34]	 Source: [33]	1- It performs well with problems that generate outputs in range. 2- Same as unipolar sigmoid. Source: [33]
Sigmoid weighted linear unit (SilU)	$a_k\ (z_k) = z_k\sigma(z_k)$ Source: [35]	 Source: [13]	1- Overcomes the Relu performance. 2- The SilU function can be used only in a reinforcement learning-based system. Source: [35]

Derivative SiLU	$a_k(s)$ $= \alpha(z_k)\left(1 + z_k\left(1 - \alpha(z_k)\right)\right)$ Source: [13]	 Source: [13]	9. It is used for updating the gradient descent learning of the NN weight parameters. 10. The dSilU performance is significantly better than the sigmoid function. Source: [35]
A hyperbolic tangent activation function	$g(x) = Tanh(x)$ $= \frac{sinh\,(x)}{cosh\,(x)} = \frac{e^x - e^{-x}}{e^x + e^{-x}}$ Source: [33]	 Source: [33]	1- The formula can be explained as the ratio between the half difference and half sum of two exponential functions in units x and –x. Source: [33] 2- The Sigmoid and Tanh functions are equivalent in performance. Both can approximate ununiformed continuous functions. Source: [36]
Radial Basis Function	$y(x) = \sum_{i=1}^{N} w_i\, g(\lVert x - c_i \rVert)$ Source: [33]	 Source: [37]	1- It is based on the Gaussian curve principle. The radial basis function accepts real values. 2- It works by measuring the distance of each value from the origin or a certain point using the Euclidean distance. Source: [38]

A small comparison can be performed from the above linear and nonlinear activation functions discussion to show their differences, as shown in Table 5.

Table 5: General comparison between a sigmoid activation function, Relu activation function, and Leaky Relu

Factors	**Sigmoid Activation Function**	**Rectified linear unit (Relu)**	**LeakyRelu**
Performance with classification neural network	High performance but with a limited number of hidden layers	High performance but has regression problems when presented with a large number of hidden layers	High performance within a limited range of negative input vectors
Output range	[0,1]	[0, input value]	[0, input value]
Saturation problem when presented with increasing weighted inputs	Yes	No	No
Vanishing Gradient	During saturation	Dying Relu happens when weighted input to Relu is negative affects stop learning.	Less than Relu
Complexity	Saturation leads to a reduced gradient	Gradient = 0 leads to a sparse network, keeping it less dense	Denser than Relu
Learning time	Less	Less	Less

2.3 Loss Functions

Classification and regression are two main tasks in machine learning techniques. In a supervised classification process, a set (n) of input data samples (x_i) where $(i = 1 \dots n)$ and the target is y_i (where $i = 1 \dots . n$). The main goal is to train a model or a function $f(x)$ that can predict the target vector y from a new input vector x (unseen). In regression models, the predicted target value of the input vector x is continuous. However, in classification models, the predicted target value of x is discrete. The main goal in training the function $f(x)$ is to obtain the best prediction with a minimum loss for all training data. The formula of the regression model function f(x) is shown in Eq. 5 below [39]:

$$f(x)\ min\ \sum_{i=1}^{n} l(f(x_i) - y_i) + R_\lambda(f), \tag{5}$$

where:

f(x$_i$)-y$_i$: represents the difference between the learning function and the target y_i .
l: represents the loss function.
$R_\lambda(f)$: represents the regularization to avoid overfitting.

The formula is shown in Eq. 6, [39]: for a classification model.

$$f(x)\ min\ \sum_{i=1}^{n} l(y_i f(x_i)) + R_\lambda(f)\ , \tag{6}$$

where:

y_i is the label of x_i.
$y_i f(x_i)$ is the difference between the learning function *f(x$_i$)* and the hyperplane.

In recent years [39-42]. have developed numerous loss functions that suit their classification tasks, such as the capped SVR, RSVR-GL, Capped SVM, ramp loss, truncated pinball loss, and C-Loss for support vector classification problems.

In general, the loss function can be formulated as in Eq. 7, [39]:

$$min_L \| P(L) - X \|_l + R_\lambda(L), \tag{7}$$

where:

P is an operator,
l is the loss function, and
$R_\lambda(L)$ is the regularization factor.

All loss functions are used to describe the quality of the learning function in both the classification and regression tasks. The main task in calculating the loss is by computing the difference between the true label and the predicted or target label. If the difference value is close to 0, then the learning function is valuable; otherwise, it needs to be enhanced. Table 6 presents some of the loss functions and their mathematical representations. There are many other loss functions, but most are derived from the functions presented in Table 4.

Table 6: Mathematical representations of some well-known Loss functions. Source: [43-46]

Tasks	**Loss function**	**Math. Representation**
Classification	Square loss	$L(w.y) = (w-y)^2 = (1-wy)^2$
	Mean of Max Square error (MMaSE)	$MMaSE = \frac{1}{N}\sum_{n=1}^{N} max_{1,2}\ ((T-P)^2)$
	Max Absult Error (MaAE)	$MaAE = \max(\lvert T-P\rvert$
	Perceptual loss function	$\mathcal{L}(\theta) = \lambda_{mse}\mathcal{L}_{mse}(\theta) + \lambda_{\mathrm{p}}\mathcal{L}_p(\theta)$ Where: λ_{mse} and λ_p are weighting scalars square error loss and perceptual loss.
	Hinge loss	$L(w.y) = \{1-wy, 0\} \ =:\ \lvert 1-wy\rvert_+$
	Squared hinge loss	$L(w.y) = \{1-wy, 0\}^2 \ =:\ \lvert 1-wy\rvert^2{}_+$
	Cubed hinge loss	$L(w.y) = \{1-wy, 0\}^3 \ =:\ \lvert 1-wy\rvert^3{}_+$
	Logistic loss (cross-entropy)	$L(w.y) = -(ylog(p) + (1-y)log(1-p))$
	Angular-Softmax Loss	$L_{AS} = -\frac{1}{N}\sum_{i=1}^{N} \log(\frac{e^{\lVert x_i\rVert\ \Psi(\theta_{y_i,i})}}{e^{\lVert x_i\rVert\ \Psi(\theta_{y_i,i})} + \sum_{j\neq y_i} e^{\lVert x_i\rVert\ \cos(\theta_{j,i})}}$
	Additive-Margin Softmax Loss	$L_{AM} = -\frac{1}{N}\sum_{i=1}^{N} \log(\frac{e^{s.(cos\theta_{y_i}-\mathrm{m})}}{e^{s.(cos\theta_{y_i}-\mathrm{m})} + \sum_{j=1,j\neq y_i}^{c} e^{s.cos\theta_j}}$
	ArcFace Loss	$L_{AF} = -\frac{1}{N}\sum_{i=1}^{N} \log(\frac{e^{s.(cos\theta_{y_i}+\mathrm{m})}}{e^{s.(cos\theta_{y_i}+\mathrm{m})} + \sum_{j=1,j\neq y_i}^{n} e^{s.cos\theta_j}}$
	Marginal Loss	$L_M = \frac{1}{N^2-N}\sum_{i,j,i\neq j}^{N} \xi - y_{ij}(\theta - \lVert \frac{x_i}{\lVert x_i\rVert} - \frac{x_j}{\lVert x_j\rVert} \rVert_2^2))$
Regression	Square loss	$L(w.y) = (w-y)^2$
	Absolute value loss	$L(w.y) = \lvert w-y\rvert$
	ϵ-insensitive loss	$L(w.y) = \{\lvert w-y\rvert - \epsilon. 0\} \ =: \lvert w-y\rvert_\epsilon$ where ϵ is a constant ≥0

3.0 NEURAL NETWORKS OPTIMIZATION

Many NN techniques have been adapted in various f research fields such as (image processing, text recognition, biometric security systems such as fingerprint detection, signal processing, and so forth). Optimization algorithms are an important component of the NN Model Optimization aims to learn the parameters from the input data using the activation function [47]. Excellent and effective optimization algorithms influence the performance improvement of the NN model. The optimization techniques can be categorized into the following types [48]:

1. First-order optimization methods such as the stochastic gradient method (SGM). It is used to update the weights of the NN model. This update leads to changing and updating the model parameters to minimize the loss rate.
2. High order optimization methods such as Newton's method.
3. Heuristic derivative-free optimization such as the coordinate descent method, global optimization methods like tabu, and annealing search to produce networks with less complexity and a high classification rate

Many improved SGM versions represent the first-order optimization methods that are widespread and fast-evolving. However, they do not attract users and are occasionally used as a black box optimizer. The high-order optimization methods are very fast in converging, and the information extracted from the curve could empower the search to be more effective. The negative aspect of a high-level optimizer is the storage operation of the inverse Hessian matrix. The solution to overcome this problem is the Newton method. Derivative optimizations are used in some instances, such as when the activation function's derivative is unavailable or is problematic to compute. This problem can be solved using the heuristic search-based rules method or supporting the activation function with examples [48].

4.0 REGULARIZATION

Regularization can be explained as a technique used to improve the training model for the unseen data or the test data when training a limited size dataset or the incongruous optimization method [49].

The main task in data classification and supervised classifier is to find a good learning function or learning model f which can approximate the target output from the input samples accurately. The neural network training function $[f_w: x \rightarrow y\ ]$ aims to allocate the weight, which minimizes the loss function as in Eq.8.

$$w^* = argmin_w L_{min}(f(x; w)) \qquad (8)$$

There are different regularization techniques such as [49]:

1. Data regularization: To transform the training dataset D to a new dataset DR. Transforming data could be done by performing feature extraction, preprocessing, redistributing the dataset, or generating a new sample such as resizing, rotating, side shifting, which are called data augmentation.

2. Network work architecture regularization: This type considers improving the function f_w input-output mapping properties to suit the input data such as weight sharing, activation function selection, noise models, multitask learning, dropout, and model selection.

3. Loss function regularization: This includes, for example, cross-entropy and mean square error. More details are presented in the study.

5.0 OVERFITTING AND UNDERFITTING

The error bias generated from inaccurate assumptions used in building a model is like predicting a nonlinear function from a linear model, or the learning model is too complex and needs infinite training data. High bias indicates that the learning model cannot detect the pattern between the input data and the target label called underfitting. On the other hand, high variance indicates that the learning model trains all data, even the noise, outliers, and replicated data. This indication leads to perfect training, which is too good to be true, called overfitting [50] see Fig.5.

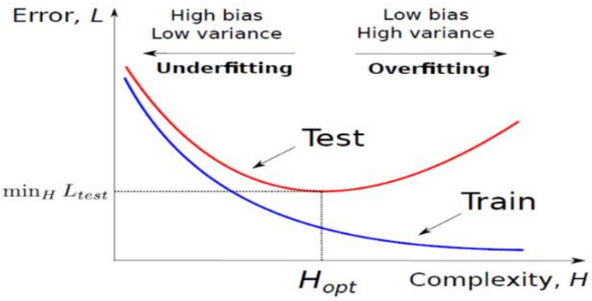


Fig. 5: Explanation of model overfitting and underfitting. Source: [51]

6.0 NEURAL NETWORK-RELATED WORK

6.1 Linear And Non-Linear Activation Function Related Work

This section presents some studies that employed or developed activation functions in various applications.

Relu is considered the most common activation function in deep learning. The researchers in [52] proposed parametric Relu for segmenting and classifying pulmonary lobes. They were also proposed using coordination-guided CNN with Vnet architecture. In [53] study, another improved Relu is proposed and called Mexican Relu (MeLU) which relied on partial derivatives. Moreover, they proposed using an ensemble of different activation functions with small size datasets provided by the internet.

Another use for Relu is proposed by [54] and [55]. The studies discussed using the Relu activation function as the output classifier instead of the Softmax classifier. In the study, two models were used: the feedforward NN and the VGG-like deep neural network. The proposed approach of [54] tested CIFAR10, MNIST, and SVHM with ResNet50, PreAct RestNet 18, and PreAct RestNet34 architectures. With a similar motivation to [54], the proposed approach is tested using the MNIST, fashion-MNIST, and the Wisconsin Diagnose Breast Cancer [55] However, Relu with CNN showed a lower performance level than Softmax during the training phase. This situation is stated as being related to the slow divergence of Relu.

Also, based on Relu, a novel activation function was proposed by [56] called CRELU. The proposed activation scheme simultaneously preserves positive and negative phase information while enforcing non-saturated non-linearity characteristics. The main advantage of CRELU, it enables the reconstruction property of the CNN, which empowers the generalization of the CNN features. CRELU outperformed the other activation functions using the top-1 and top-5 measurements after testing on CIFAR10, CIFAR101, and Imagenet.

Recently, [57] presented a novel activation function called the S-Shaped Rectified Linear Unit (SRelu) based on two laws. The first is the Webinar-Fechner law ($s = k \log p$) and the second (Stevens law ($s = kp^e$). The results showed excellent performance compared to Relu and LRelu with four benchmark datasets, which are CIFAR10, CIFAR100, MNIST, and ImageNet. In continuation, [58] proposed a novel probabilistic activation function named ProAct that utilized a normalized distribution with a mean value like Relu and a fixed or trained variance value using backpropagation. The stochastic noise played a valuable role in preventing overfitting like a dropout.

Another study [59] was conducted to propose an activation function that does not base on Relu. The proposed design is based on applying the optimizer search space of [60]. They employ a mixture of a full and reinforced learning-based search called the switch function. Swish function outperformed Relu with top1 classification accuracy of about 90% with ImageNet and 60% with Mobile NASNet-A and Inception-ResNet-v2. The researchers tested the proposed activation functions using the resNet-164(RN) by [19], Wide ResNet 28-10 (WRN) by [61], and DenseNet 100-12 (DN). The proposed method produced an acceptable performance which is like Relu. However, the proposed activation function required extensive computational processes and was time-consuming. Moreover, another study proposed the Differential Equation unit (DEU) activation function by [25]. The proposed DEU aimed to decrease the number of neurons by reforming the existing neurons. The study tested the proposed DEU by using two punch mark datasets, MNIST, and fashion MNIST, they claimed that DEU outperformed the Relu activation in terms of the accuracy rate and the smaller number of NN parameters/neurons used in the training phase.

From the above-presented studies, we can answer questions 2 and 3 from section 1. Some studies proposed an improvement to Relu to activate all negative and positive feature maps, especially with small size datasets. Other studies proposed an activation function to overcome the proportional increment in the number of parameters versus the increment of the number of neurons units and reduce the model complexity. Another study proposed improving the sigmoid activation function as a learning and searching-based activation function.

Choosing the appropriate activation function depends on the quality of the used dataset, the number, and the distribution of class labels during the training process. Moreover, the size of the used dataset is important in deep learning since we need to extract only the high-value features and then pass them to multiple layers of training with different kernels sizes. Another factor that must be considered in choosing the activation function is the cost of calculation complexity and time-consuming. However, using deep learning techniques improved the Neural network model's performance. Deep searching for the best feature is represented by the increasing number of CNN blocks. That could improve the model performance. Relu is the most common activation function that activates positive features and discards the negative features. However, when dealing with small-sized datasets, utilizing Relu results in rapid dimensionality reduction, resulting in poor model performance. That is why researchers keep developing new activation functions or improving them to work on their datasets and gain the best models performance.

6.2 Regularization

Regularization techniques are used in all Neural networks to enhance the performance of the NN. This section aims to demonstrate some research related to data augmentation, one of the regularization techniques.

In [62], The researchers have studied the use of augmented data in training CNN Samples of stationary or moving target images taken from synthetic aperture radar (SAR) were used in this study. An augmentation algorithm was introduced to extract the attributed scattered centers (ASC), and depending on the electromagnetic characteristic of the SAR targets, the Sparse Representation (S.R.) algorithm was applied in reconstructing the target images. Due to the extraction of high-value characteristics from the original image, the proposed method is claimed to have achieved high results due to the minor difference between the tested and trained samples. This condition could be explained by the reconstructed target samples sharing the same vector features as the original images.

Furthermore, the researchers in [63] proposed using data augmentation with CNN architecture to classify the Drum transcription sound data. This study used transcriptions from three drum instruments and fed them into CNN. This study showed that the CNN performance with dropout and Gaussian noise was comparable and no significant difference for the three drum instruments. Another novel study was published by [64]. This study has proposed a new image augmentation algorithm called Part Analysis of the shape. The research was based on the idea of creating new animal images by altering the original image (crop, flip, rotate and resize). The main objective is to maintain the integrity of the original image features. This method has improved the performance of the CNN architecture by increasing the number of samples.

Also, research by [65] discusses the classification of multilabel land cover scenes from the satellite. The research discussed two problems the first the small size of the collected data and the network overfitting. Data augmentation techniques such as rotation in (45^o,-45^o), denoising, rescale, and cropping was proposed to overcome these two problems. Besides the data augmentation techniques, the researcher applied different rates of dropouts. They believed that the improvement could be caused due to the use of dropouts rather than only data augmentation.

On top of that, shuffling and mixing training samples were applied in [66] to increase the training dataset. The results showed better CNN model performance. In the [67] study, the researchers attempted to use CNN for Electroencephalography (EEG)-based brain-computer interface (BCI). The EEG signals were first transformed into band powers using Morlet wavelet transformation and then passed to CNN for features extraction. This research aimed to enable disabled users to use the computer by using brain signals.

In the case of the unavailability of a dataset with proper size miscellaneous, the above researchers attempted various data augmentation approaches to enhance their model performance. Data augmentation is essential to increase the training samples size and improve their quality by denoising, cropping, shuffling, mixing, transforming, and rotating at different angles. As a result, these determinants improve the performance of the classification model. Deep learning requires a huge dataset size with various samples. However, increasing the number of classes complicates the model training process and increases the training time too. Also, some data augmentations like constructing target samples from training samples led to a high accuracy rate because of the likeness between target samples and training samples.

6.3 Loss Function Related Work

For the past few decades, many researchers have tried to present loss functions that minimize the performance error of classification algorithms. In this section, some of the latest research reviews conducted on loss functions and their results are presented and discussed.

Twin SVM (TWSVM) has been proposed with a rescaled hinge (Rhinge) by [68] in coping with large datasets with noise and unbalanced data while maintaining the generalization ability. They tested the performance of the proposed TWSVM and Rhinge with benchmark datasets from UCI. The proposed approach results showed higher performance than with Least square loss, Cross-entropy loss, and hinge loss. Next, a study [69] proposed a novel loss function to discriminate images and classify them as fake or real. The images were taken from the ImagNet dataset by first using the convolutional (VGG16) and encoding techniques to extract the embedded features of the source and target images. Both encoded source and target images were reconstructed. A discriminator was used to classify the reconstructed images as real and fake. Unlike previous studies, [70] proposed a novel adversarial loss function composed of generator loss (L.G.), discriminator loss (L.D.), and Encoder loss (L.E.). These three loss functions were summated for the source and target images. In the same study, another loss function called the conservation loss function was proposed to overcome the overfitting problem. The results showed impressive results, especially for a large dataset size.

The author in [71] found certain differences, suggesting that a novel loss function enhances the accuracy of the information retrieval system. The proposed loss function combines the Average Precision (A.P.) and the Normalized Discounted Cumulative Gain (NDCG) to enhance the training time. The above findings are consistent with the study by the author [72] proposed a novel loss function that combined the categorical cross-entropy (CCE) loss function by imposing Mean Absolute Error (MAE). The proposed deep learning neural networks with the proposed loss function improved the model performance for the dataset with different noise rates. In a nutshell, their reported results showed better results than applying CCE and MAE individually.

The study by [73] contradicts the study by [74] concerning the weakness of RMSE and its inappropriateness to evaluate the average error of the estimation models with normal distribution problems and for triangle inequality. Instead, other researchers have been encouraged to use MAE over RMSE for their statistical evaluation, as found in [75-77]. This evaluation mechanism is further supported by [73], who have presented, contrary to [74], the advantages of RMSE over MAE. It is stated that RMSE is not ambiguous and is more appropriate to be used in models with the normal distribution and triangle inequality. In the final discussion, the researchers explained that MAE and RMSE are important in different fields that suit their functionality.

Replacing the softmax activation function layer with the linear Support vector machine proposed by [78]. Besides replacing the cross-entropy loss function introduced by [79] with margin-based loss function for facial expression and recognition problems in deep convolutional learning NN. The experiment was conducted on MNIST and CIFAR-10. The results showed that the use of SVM in deep learning for classification tasks proved to be superior over softmax with cross-entropy due to the ability of SVM in detecting the maximum margins among data points of various classes. On the other hand, the softmax function minimizes the cross-entropy or maximizes the likelihood of the log.

Despite the numerous work carried out on the same subject, another study [74] has conducted a comparative study between the measurements of the root mean square error (RMSE) and the mean absolute error (MAE). The main objective of this study is to explain the advantages of using MAE over RMSE in estimating the average error of the environment climate data in model performance evaluation and inter-comparison models. This study proved that the MAE error estimation is far better than RMSE because it can maximize the average error by squaring the total errors and calculating the square root subsequently. On the other hand, MAE calculates the mean of the total errors in an unambiguous way. In the conclusion of the study, it is recommended that RMSE should no longer be used in estimation models.

Even though there have been several known quantitative analyses, the study by [80] has proposed an energy-based model to discriminate input images for classification and verification purposes such as face recognition and verification. This technique was suggested when the training dataset contains many categories with few training samples. The minimized loss function was proposed in that the difference of the weighted energy between two images is calculated using two convolution networks. If the difference is equal to 0, these two images belong in the same category. On the other hand, if the difference is greater than 0, they belong to different categories. However, the proposed loss function was unable to estimate the probabilities of the categories. Two combined datasets were used; the first was the A.R. database of faces from Purdue University and the second set was the grayscale Feret database. It is concluded that the similarity metrics can be used in many different applications, such as in building invariant kernels, for instance, the support vector machine kernel and other kernel-based methods. Another study [90] proposed a combination of Modified Particle Swarm Optimization (MPSO) and Conjugate Gradient (CG) algorithm. The proposed hybrid techniques aim to optimize CNN performance and overcome the trapping in the local minima. The reported results showed a clear improvement in CNN performance due to well-updated weights. According to the researchers, the model showed good convergence and less computational cost.

The above studies proposing and developing loss functions aim to update the training model parameters to produce the least error rate and high accuracy performance. Nevertheless, choosing the appropriate loss function depends on the type of model. Some of the proposed loss functions are used for regression models and the other for classification models. Another critical issue related to choosing the loss function is the amount of outlier and noise samples in the dataset. Some loss functions such as cross-entropy cannot handle outliers and noisy data contrary to the mean square loss function. Although, mean square error suffers from the fast vanishing. So, proposing a novel or improved loss function depends on the dataset and methods used to collect and present it.

6.4 Overfitting And Underfitting Related Work

This section spots light on some research studies that discussed the overfitting and underfitting problems using different approaches and presented as follow:

Interestingly, the study by [81] was proposed a new approach and algorithm called the consensus-based classification (CCS) to overcome overfitting. Their approach depended on training many (k) models with different thresholds. In this study, the approach was applied to individual models and combined models. It was found that the number of samples which was classified correctly had increased with high threshold values (greater than 0.8) with a high accuracy rate. Finding by Author [81] also points toward the results of the study [82] with the introduction of a new path in overcoming network overfitting when using UAV remote sensing images. A group of CNN architectures was applied, such as Alex Net, Caffinet, VGG Net, and Google Net. In addition to the proposed sparse coding technique for dimensionality reduction. This paper used the Support Vector Machine (SVM), the Random Forest algorithm, and the Extreme Learning Machine (ELM). The proposed approach took less time in training CNN using the VGG16 Net. The GAP algorithm performance with the VGG16 Net also showed high accuracy and less time consumption in training the CNN due to removing the F.C. layer.

Despite the extensive evidence provided by [81], there appears to be a wide difference in [83] study proposing the use of the ISING dropout technique in solving two main problems in deep neural networks: overfitting and the increase in the number of parameters. The first experiment was conducted to train the MNIST dataset using the MLP classifier with and without a dropout, and the results showed less accuracy rate (90%) with the proposed ISING than without (94.4%). On the other hand, the result showed a noticeable reduction in the number of parameters used while training by 41% with Adam optimizer. The proposed ISING dropout decreased the accuracy rate with all datasets and decreased the number of model parameters, leading to faster training. However, the researchers in [84] have presented in their study a combination of the CNN and LSTM with batch normalization techniques with the VGG16 network architecture in detecting images of fake faces. The face data were pre-trained using the VGG-face (2.6M images with 2622 classes) to initialize the parameters. Then, transfer learning was used to transfer the knowledge from the VGG to the proposed face-anti spoofing. The main purpose of transfer learning is to overcome overfitting problems. Two datasets were used in this study: the MSU-MFSD and the reply–Attacks.

The above-discussed studies presented different methods to overcome the overfitting problem. These methods include using probabilities threshold, data augmentation, features normalization, and multi-classifiers with various CNN architecture. Mixing different techniques to overcome the overfitting problem could complicate the model architecture, increase the training time complexity, and reduce the model performance accuracy.

The final outputs from the discussed studies in the related work section can be summarised as follow :

Neural networks expansion led to open unlimited research fields using various datasets. However, the researchers have to improve the available techniques or develop new approaches that suit their requirements, such as improving or developing activation functions that work well with their dataset. Then, enhancing or proposing a novel loss function to enhance the model performance. Finally, applying different data augmentation techniques to manipulate and increase the training samples, using regularization techniques such as dropout, and increasing the NN layers for deep learning training.

7.0 DISCUSSION

Neural networks are a machine learning technique based on human knowledge in which a machine learns human experience using special techniques for extracting features, selecting features, filtering data, training classifiers with binary and multiple labels, and various other applications. The essential advantages of Neural networks are their efficiency in processing data of different sizes, reducing computational complexity, reducing the consumption of training time, and processing raw data such as 2D and 3D images, audios, and videos. The neural network training process starts with computing the feature vectors of an image that have more influence on the performance of the prediction process. The best mixture of these features is chosen to gain the most accurate classification results [4] and [85].

NN algorithms have been a great aid in various fields such as medical image detection and diagnosis, for example, brain and breast images fed from MRI, CT, and mammogram techniques, character detection and recognition, object detection, speech recognition, natural language processing, vehicle navigation, product recommendation and many other fields [4].

This paper discusses the main aspects of NN techniques and deep learning applications. It also presents the main components of deep learning techniques and explains the importance of applying the convolution Neural Network since it is the most popular neural network in Deep Learning. The paper explains the techniques that empower better performance over the traditional procedures, such as high extraction feature values, faster processing, and less requested techniques, in addition to the generalization aptitude of the Neural networks. Furthermore, it sheds light on the problems that might affect the classification performance and provides the methods to overcome them. Besides that, this paper includes reviewing some of the latest research that uses a neural network, particularly the CNN layers and applications. In a nutshell, deep learning becomes the preferred choice for researchers compared to conventional methods. It provides a wide field of research using fewer algorithms with significant performance accuracy and minimal loss errors. Also, it opens a wide range of creativity in designing CNN architectures.

7.1 Neural network disadvantages

Besides the enormous advantages to the use of Neural networks, there are a set of disadvantages such as:

1. Training consumes a long time, especially with a large size dataset, and the duration cannot be predicted.
2. Requires a large number of training samples to train the designed model due to the use of activation functions that prune samples.
3. There is no condition or rule to decide the suitable artificial NN structure for a problem because it depends on a trial and error basis.
4. NN can work with numerical values, so all the inputs of NN should be in numerical form.
5. Optimizing the parameters like the learning rate, number of nodes in the hidden layers, and number of the hidden layers.
6. The way that Artificial NN processes input data and produces the requested results is still ambiguous [86].

7.2 Challenges

Although there is significant progress in the neural networks with deep learning techniques, there is still more to explore and research to improve ANN performance. This progress is due to the accelerated expansion in information technology. It leads to developing and improving ANN parameters, such as designing new activation functions and loss functions that suit various data types and sizes. Moreover, increasing the hidden layer or the number of neurons consider the best approach to improve the quality of the extracted features, but this leads to an increase in the computational time. Another issue related to creating deep networks which is the unstable gradient. It could be either vanishing which indicates the model has not converged so far or exploding gradient resulting from large weights. Besides, for deep NN the challenge is to search for the hardware techniques that decrease the processing complexity and speed up the processing at a low cost.

7.3 Limitations

The rapid progress of NN makes it an uneasy task to prepare a to date review paper. Many studies are published yearly about NN applications, functions, and newly developed architectures. So, surveying all these papers from different databases was a challenge. Other constraints related to the dataset are the data collecting process, applying real or synthetic, and the limitation of sample size. To create an effective NN model with a high result, the dataset must contain a large and a various number of samples. The quality of the collected samples has the most impact on designing the NN model and choosing the appropriate activation and loss functions.

7.4 The future direction of Neural networks

Recently deep learning neural networks have become an excellent choice for solving various problems in different fields. The reason is that the deep learning techniques guarantee very high performance for the models even when training raw data. Also, deep learning is powerful to control the number of the model parameters, layers, and the number of feature extraction filters [13]; [52]; [54]; [87], [88] etc.

ACKNOWLEDGMENT

The author would like to thank UKM for providing the facilities and financial support under FRGS/1/2019/ICT02/UKM/02/9 entitled "Convolution neural network enhancement based on adaptive convexity and regularization functions for fake video analytics". Not forgetting the research team members at the Digital Forensic Lab and Medical and Health Informatics Lab at the Faculty of Information Science and Technology, Universiti Kebangsaan Malaysia.

REFERENCES

[1] V. Dunjko and H.J. Briegel, "Machine learning & artificial intelligence in the quantum domain: a review of recent progress", *Reports on Progress in Physics*, Vol. 81 No. 7,2018, p. 074001.

[2] T. Hong Khai, et al., “Underwater Fish Detection and Counting Using Mask Regional Convolutional Neural Network”, *Water*, 2022. Vol. 14 No**.** 2, p. 222.

[3] M.D. Rechenthin, "Machine-learning classification techniques for the analysis and prediction of high-frequency stock direction". *University of Iowa*, 2014, DOI: 10.17077/etd. fvui75i2.

[4] P. Henderson et al., "Deep reinforcement learning that matters", *arXiv preprint arXiv*:1709.06560, 2017.

[5] B. J. Erickson et al., "Machine Learning for Medical Imaging*", RadioGraphics*, Vol.37 No. 2, 2017, p. 160130, DOI:10.1148/rg.2017160130.

[6] M. Mijwil," Overview of Neural Networks", Computer Engineering Techniques Department, 2019, Vol.1, p.2.

[7] G. Tzanis et al., "Modern applications of machine learning". *in Proceedings of the 1st Annual SEERC Doctoral Student Conference–DSC*, 2006, Vol. 1, No. 1, p. 1-10.

[8] P. Goyal, et al.," Deep Learning for Natural Language Processing: Creating Neural Networks with Python", *Berkeley, CA: Apress*, 2018.

[9] S.M.M.N. Kahaki, et al., “Deep convolutional neural network designed for age assessment based on orthopantomography data”. *Neural Computing & Applications*, 2020 Vol. **32 No.** 13: p.9357-9368.

[10] J. Bai, et al., "Image character recognition using deep convolutional neural network learned from different languages", *IEEE International Conference on Image Processing (ICIP)*, Paris, France,27-30 Oct. 2014.

[11] H. Kagaya et al.," Food detection and recognition using convolutional neural network". in *Proceedings of the 22nd ACM international conference on Multimedia*, New York, United States, November 2014, 1085–1088.

[12] A. A. A. Ali and S. M, "Arabic Handwritten Character Recognition Using Machine Learning Approaches," *2019 Fifth International Conference on Image Information Processing (ICIIP)*, 2019, p. 187-192, DOI: 10.1109/ICIIP47207.2019.8985839.

[13] N. Altwaijry et al., "Arabic handwriting recognition system using convolutional neural network*". Neural Computing and Applications*, 2020: p. 1-13.

[14] T. Yoshioka et al., "Far-field speech recognition using CNN-DNN-HMM with convolution in time". in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP),* South Brisbane, QLD, Australia, 19-24 April 2015, DOI**:** 10.1109/ICASSP.2015.7178794.

[15] D. Palaz et al., "Convolutional neural networks-based continuous speech recognition using raw speech signal". in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, South Brisbane, QLD, Australia, 19-24 April 2015, DOI**:** 10.1109/ICASSP.2015.7178781.

[16] M.A. Nielsen, "Neural networks and deep learning", *Determination press USA*, Vol. 25. 2015:

[17] C. Nwankpa et al., "Activation functions: Comparison of trends in practice and research for deep learning*", arXiv preprint arXiv*, 2018,1811.03378.

[18] K. Dhana Sree, "Data Analytics: Role of Activation function In Neural Net*", International Journal of Innovative Technology and Exploring Engineering*, Vol. **8**, 2019, p. 299-302.

[19] K. He, et al.," Deep residual learning for image recognition". in *Proceedings of the IEEE conference on computer vision and pattern recognition*, Las Vegas, NV, USA, 27-30 June 2016.

[20] A. Krizhevsky et al., "Imagenet classification with deep convolutional neural networks", in *Advances in neural information processing systems*, *Communications of the ACM,* Vol. 60 No.6, 2017, p.84-90.

[21] E. Park et al., "Patch-based Fake Fingerprint Detection Using a Fully Convolutional Neural Network with a Small Number of Parameters and an Optimal Threshold," *arXiv preprint arXiv*, 2018, 1803.07817.

[22] R.D. Labati et al., "A novel pore extraction method for heterogeneous fingerprint images using convolutional neural networks*", Pattern Recognition Letters*, Vol. 113**,** 2018: p. 58-66.

[23] R. H. Hahnloser, et al., "Digital selection and analogue amplification coexist in a cortex-inspired silicon circuit*", Nature*, Vol. 405 No.6789, 2000, p. 947.

[24] B. Xu et al., "Empirical evaluation of rectified activations in convolutional network*", arXiv preprint arXiv*:1505.00853, 2015.

[25] M. Torkamani, et al., "Learning Compact Neural Networks Using Ordinary Differential Equations as Activation Functions*", arXiv preprint arXiv*, 2019, 1905.07685.

[26] K. Kakuda, et al., "Nonlinear Activation Functions in CNN Based on Fluid Dynamics and Its Applications*", Computer Modeling in Engineering & Sciences*, Vol.118 No. 1, 2019, p. 1-14.

[27] I. Goodfellow, et al., "Deep learning", *MIT Press*, 2016.

[28] B. Ding, et al., "Activation functions and their characteristics in deep neural networks, in *2018 Chinese Control And Decision Conference (CCDC),* Shenyang, China, 9-11 June 2018.

[29] K. He, et al., "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification", in *Proceedings of the IEEE international conference on computer vision*, Santiago, Chile, 7-13 Dec. 2015.

[30] I. J. Goodfellow et al., "Maxout networks*". arXiv preprint arXiv*, 2013, 1302.4389.

[31] D. Chen et al., "A fast and accurate dependency parser using neural networks". in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), Association for Computational Linguistics,* Doha, Qatar, 2014, p- 740–750.

[32] G. Klambauer, et al., "Self-normalizing neural networks". in Advances in neural information processing systems, 31st Conference on Neural Information Processing Systems (NIPS 2017), Long Beach, CA, USA 2017.

[33] Vehbi Olgac et al., "Performance Analysis of Various Activation Functions in Generalized MLP Architectures of Neural Networks". International Journal of Artificial Intelligence And Expert Systems, Vol.1, 2011, p. 111-122.

[34] T. Kocak, "Sigmoid functions and their usage in artificial neural networks*", School of Electrical Engineering and Computer Science*, presentation, 2015.

[35] S. Elfwing et al., "Parallel reward and punishment control in humans and robots: Safe reinforcement learning using the MaxPain algorithm", *Joint IEEE International Conference on Development and Learning and Epigenetic Robotics (ICDL-EpiRob)* IEEE, Lisbon, Portugal, 18-21 Sept. 2017.

[36] S. Eger et al., "Is it time to swish? comparing deep learning activation functions across NLP tasks*". arXiv preprint arXiv*, 2019, 1901.02671.

[37] "RBF Neural Networks". Access time 12:23 am. Date: 14/02/2021

[38] J. Christensen et al., Chapter ten – "Optimization for Refinement of Vehicle Safety Structures, in Nonlinear Optimization of Vehicle Safety Structures", Butterworth-Heinemann: Oxford, 2016, p. 415-450.

[39] F . Nie et al.," An investigation for loss functions widely used in machine learning". Communications in Information and Systems, 2018, Vol 18 No 1, p. 37-52.

[40] F. Nie et al., "Joint capped norms minimization for robust matrix recovery". *in The 26th International Joint Conference on Artificial Intelligence (IJCAI 2017),* 2017.

[41] K. Wang et al., "Robust support vector regression with generalized loss function and applications". *Neural Processing Letters*, 2015, Vol. 41 No**.**1, p. 89-106.

[42] H. Masnadi-Shirazi et al., "On the design of loss functions for classification: theory, robustness to outliers, and savage boost", *in Advances in neural information processing systems,* 2009.

[43] M. Edalatifar, et al., "Using deep learning to learn physics of conduction heat transfer". *Journal of Thermal Analysis and Calorimetry*, 2021, Vol.146, p.1435–1452.

[44] M. Gholizadeh-Ansari et al.," Deep Learning for Low-Dose CT Denoising Using Perceptual Loss and Edge Detection Layer", *Journal of Digit Imaging*, 2020, Vol 33 No**.** 2, p. 504-515.

[45] L. Rosasco et al., "Are loss functions all the same?", *Neural Computation*, 2004, Vol. 16 No. 5, p. 1063-1076.

[46] K. Janocha et al., "On loss functions for deep neural networks in classification", *arXiv preprint arXi,* 2017,1702.05659.

[47] S. Sun et al., "A Survey of Optimization Methods from a Machine Learning Perspective", arXiv preprint arXiv, 2019,1906.06821.

[48] A. Abdullah, et al., "Ting, Orientation and Scale Based Weights Initialization Scheme for Deep Convolutional Neural Networks". *Asia-Pacific Journal of Information Technology and Multimedia*, 2020, Vol. **9 No.** 2, p. 103-112.

[49] J. Kukačka, et al., "Regularization for deep learning: A taxonomy", *arXiv preprint arXiv,* 2017, 1710.10686.

[50] A. Tharwat, "Classification error: Bias and variance, Underfitting and Overfitting", DOI: 10.13140/RG.2.2.15956.91523, 2018.

[51] S. Demyanov," Regularization methods for neural networks and related models", *A gateway to Melbourne's research publication*,2015.

[52] W. Wang et al., "Automated Segmentation of Pulmonary Lobes using Coordination-Guided Deep Neural Networks*". arXiv preprint arXiv,* 2019,1904.09106.

[53] G. Maguolo, et al., "Ensemble of Convolutional Neural Networks Trained with Different Activation Functions". *arXiv preprint arXiv,* 2019, 1905.02473.

[54] B. Wang et al., "Deep neural nets with interpolating function as output activation", *in Advances in Neural Information Processing Systems,* Montréal, Canada, 2018.

[55] A. F. Agarap, "Deep learning using rectified linear units (Relu)*". arXiv preprint arXiv,* 2018, 1803.08375.

[56] W. Shang et al., "Understanding and improving convolutional neural networks via concatenated rectified linear units", *in the international conference on machine learning,* New York, NY, USA, Vol 48, p. 2217–2225 2016.

[57] X. Jin et al., "Deep learning with s-shaped rectified linear activation units". in *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

[58] J. Lee et al., "ProbAct: A Probabilistic Activation Function for Deep Neural Networks*". arXiv preprint arXiv*, 2019, 1905.10761.

[59] P. Ramachandran et al., "Searching for activation functions*", arXiv preprint arXiv,*2017,p.1710.05941.

[60] I. Bello et al., "Neural optimizer search with reinforcement learning", *in Proceedings of the 34th International Conference on Machine Learning,* Vol. 70, 2017, p. 459–468.

[61] S. Zagoruyko et al., "Wide residual networks*", arXiv preprint arXiv*, 2016, p.1605.07146.

[62] J. Lv et al.,", Data Augmentation Based on Attributed Scattering Centers to Train Robust CNN for SAR ATR", *IEEE Access,* 2019, Vol **7,** p. 25459-25473.

[63] C. Jacques et al.," Data Augmentation for Drum Transcription with Convolutional Neural Networks", *arXiv preprint arXiv,* 2019, 1903.01416.

[64] N. Nayef et al., "ICDAR2019 Robust Reading Challenge on Multi-lingual Scene Text Detection and Recognition-RRC-MLT", *arXiv preprint arXiv, 2019,* 1907.00945.

[65] R. Stivaktakis et al., "Deep learning for multilabel land cover scene categorization using data augmentation", *IEEE Geoscience and Remote Sensing Letters,* 2019, Vol 16 No.7, p. 1031-1035.

[66] T. Inoue et al., "Domestic activities classification based on CNN using shuffling and mixing data augmentation", *Detection and Classification of Acoustic Scenes and Events 2018 Challenge*, 2018.

[67] S. Ullah. Halim, “Imagined character recognition through EEG signal using deep convolution neural network”. *Medical & Biological Engineering & Computing*, 2021, Vol. 59, pp.1167–1183.

[68] L.W. Huang et al., "Robust Rescaled Hinge Loss Twin Support Vector Machine for Imbalanced Noisy Classification", *IEEE Access,* 2019, Vol. 7, p. 65390-65404.

[69] X. Zhu et al., "Penalizing top performers: Conservative loss for semantic segmentation adaptation", *in Proceedings of the European Conference on Computer Vision (ECCV), Springer International Publishing*, 2018.

[70] I. Goodfellow et al., "Generative adversarial nets", *in Advances in neural information processing systems*., 2014.

[71] P. Mohapatra et al., "Efficient optimization for rank-based loss functions", *in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition,* 2018.

[72] Z. Zhang et al., "Generalized cross-entropy loss for training deep neural networks with noisy labels". *in Advances in neural information processing systems*, 2018.

[73] T. Chai et al.," Root mean square error (RMSE) or mean absolute error (MAE)? Arguments against avoiding RMSE in the literature", *Geoscientific model development*, 2014, Vol. 7 No. 3, p. 1247-1250.

[74] C. J. Willmott et al.," Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance", *Climate research*, 2005. Vol. 30 No.1, p. 79-82.

[75] M. Taylor," Partial differential equations II: Qualitative studies of linear equations", *Springer Science & Business Media*, 2013, Vol. 116, p. 525.

[76] S. Chatterjee et al., "Virtual wallet card selection apparatuses, methods and systems", 2013.

[77] A. Jerez et al., "STAT3 mutations indicate the presence of subclinical T-cell clones in a subset of aplastic anemia and myelodysplastic syndrome patients", *Blood*, 2013, Vol. 122 No.14, p. 2453-2459.

[78] Y. Tang," Deep learning using linear support vector machines", *arXiv preprint arXiv*:1306.0239, 2013.

[79] C. E. Shannon, "A Mathematical Theory of Communication", *Bell System Technical Journal*, 1948, Vol. 27 No**.** 3, p. 379-423.

[80] S. Chopra et al., "Learning a similarity metric discriminatively, with application to face verification", *in 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05*, San Diego, CA, USA, 20-25 June 2005.

[81] S. Salman, "Overfitting Mechanism and Avoidance in Deep Neural Networks", *arXiv preprint arXiv*, 2019, 1901.06566.

[82] A. Qayyum et al., "Designing deep CNN models based on sparse coding for aerial imagery: a deep-features reduction approach", *European Journal of Remote Sensing*, 2019, Vol. 52 No. 1, p. 221-239.

[83] H. Salehinejad et al.," Ising-dropout: A regularization method for training and compression of deep neural networks", *in ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Brighton, UK, 12-17 May 2019.

[84] X. Tu et al., "Enhance the Motion Cues for Face Anti-Spoofing using CNN-LSTM Architecture", *arXiv preprint arXiv*, 2019, 1901.05635.

[85] G. S. Fu et al., "Machine Learning for Medical Imaging. Journal of healthcare engineering", *Journal of Healthcare Engineering*, 2019.

[86] M. Cilimkovic, "Neural networks and backpropagation algorithm", *Institute of Technology Blanchardstown, Blanchardstown Road North Dublin*, 2015, Vol. 15.

[87] J. Gu et al., "Recent advances in convolutional neural networks". *Pattern Recognition*, 2018, Vol. 77, p. 354-377.

[88] B.H. Nayef et al., “Optimized leaky ReLU for handwritten Arabic character recognition using convolution neural networks”, *Multimedia Tools and Applications*, 2022. Vol. 81 No**.**2, p. 2065-2094