Neural Comput & Applic (2019) 31:2507–2521
DOI 10.1007/s00521-017-3210-6

ORIGINAL ARTICLE

A comparative performance analysis of different activation functions in LSTM networks for classification

Amir Farzad[1] · **Hoda Mashayekhi**[2] · **Hamid Hassanpour**[2]

Received: 3 August 2016 / Accepted: 4 October 2017 / Published online: 19 October 2017

Abstract In recurrent neural networks such as the long short-term memory (LSTM), the sigmoid and hyperbolic tangent functions are commonly used as activation functions in the network units. Other activation functions developed for the neural networks are not thoroughly analyzed in LSTMs. While many researchers have adopted LSTM networks for classification tasks, no comprehensive study is available on the choice of activation functions for the gates in these networks. In this paper, we compare 23 different kinds of activation functions in a basic LSTM network with a single hidden layer. Performance of different activation functions and different number of LSTM blocks in the hidden layer are analyzed for classification of records in the IMDB, Movie Review, and MNIST data sets. The quantitative results on all data sets demonstrate that the least average error is achieved with the *Elliott* activation function and its modifications. Specifically, this family of functions exhibits better results than the *sigmoid* activation function which is popular in LSTM networks.

Keywords LSTM · Neural network · Activation function · Sigmoidal gate

✉ Hoda Mashayekhi
hmashayekhi@shahroodut.ac.ir

Amir Farzad
amir.farzad@shahroodut.ac.ir

Hamid Hassanpour
h.hassanpour@shahroodut.ac.ir

[1] Kharazmi International Campus, Shahrood University of Technology, Shahrood, Iran

[2] Department of Computer Engineering, Shahrood University of Technology, P.O. Box: 3619995161, Shahrood, Iran

1 Introduction

LSTM, introduced by Hochreiter and Schmidhuber [1], is a recurrent neural network (RNN) architecture shown to be effective for different learning problems especially those involving sequential data [2]. The LSTM architecture contains blocks which are a set of recurrently connected units. In RNNs, the gradient of the error function can increase or decay exponentially over time, known as the vanishing gradient problem. In LSTMs, the network units are redesigned to alleviate this problem. Each LSTM block consists of one or more self-connected memory cells along with input, forget, and output multiplicative gates. The gates allow the memory cells to store and access information for longer time periods to improve performance [2].

LSTMs and bidirectional LSTMs [3] are successfully applied in various tasks especially classification. Different applications of these networks include online handwriting recognition [4, 5], phoneme classification [2, 3, 6], and online mode detection [7]. LSTMs are also employed for generation [8], translation [9], emotion recognition [10], acoustic modeling [11], and synthesis [12] of speech. These networks are also employed for language modeling [13], protein structure prediction [14], analysis of audio and video data [15, 16], and human behavior analysis [17].

Generally, the behavior of neural networks relies on different factors such as the network structure, the learning algorithm, the activation function used in each node, etc. However, the emphasis in neural network research is on the learning algorithms and architectures, and the importance of activation functions has been less investigated [18–20]. The value of the activation function determines the decision borders and the total input and output signal strength of the node. The activation functions can also affect the complexity and performance of the networks and also the

convergence of the algorithms [19–21]. Careful selection of activation functions has a large impact on the network performance.

The most popular activation functions adopted in the LSTM blocks are sigmoid (*log-sigmoid*) and hyperbolic tangent. In different neural network architectures, however, other kinds of activation functions have been successfully applied. Among the activation functions are the complementary log–log, probit and log–log functions [22], periodic functions [23], rational transfer functions [24], Hermite polynomials [25], non-polynomial functions [26, 27], Gaussians bars [28], new classes of sigmoidals [20, 29], and also combination of different functions such as polynomial, periodic, sigmoidal, and Gaussian [30]. These activation functions upon application in LSTMs may demonstrate good performance. In this paper, a total of 23 activations including the just mentioned functions are analyzed in LSTMs.

The properties that should be generally fulfilled by an activation function are as follows: The activation function should be continuous and bounded [31, 32]. It should also be sigmoidal [31, 32], or the limits for infinity should satisfy the following equations [33]:

$$\lim_{x\to-\infty} f(x) = \alpha \tag{1}$$

$$\lim_{x\to+\infty} f(x) = \beta \tag{2}$$

$$\text{with } \alpha < \beta \tag{3}$$

The activation function's monotonicity is not a compulsory requirement for the existence of the Universal Approximation Property (UAP) [32].

In this paper we investigate the effect of the 23 different activation functions, employed in the input, output, and forget gates of LSTM, on the classification performance of the network. To the best of our knowledge this is the first study to aggregate a comprehensive set of activation functions and extensively compare them in the LSTM networks. Using the IMDB and Movie Review data sets, the misclassification error of LSTM networks with different structures and activation functions are compared. The results specifically show that the most commonly used activation functions in LSTMs do not contribute to the best network performance. Accordingly, the main highlights of this paper are as follows:

1. Compiling an extensive list of applicable activation functions in LSTMs.
2. Applying and analyzing different activation functions in three gates of an LSTM network for classification.
3. Comparing the performance of LSTM networks with various activation functions and different number of blocks in the hidden layer.

The rest of the paper is organized as follows: in Sect. 2, the LSTM architecture and the activation functions are described. In Sect. 3, the experimental results are reported and discussed. The conclusion is presented in Sect. 4.

2 System model

In this section, the LSTM architecture and the activation functions employed in the network are described.

2.1 LSTM architecture

We use a basic LSTM with a single hidden layer with an average pooling and a logistic regression output layer for classification. The LSTM architecture, illustrated in Fig. 1, has three parts, namely the input layer, a single hidden layer, and the output layer. The hidden layer consists of single-cell blocks which are a set of recurrently connected units. At time t, the input vector x_t is inserted in the network. Elements of each block are defined by Eqs. 4–9.

$$f_t = \sigma\left(W_f x_t + U_f h_{t-1} + b_f\right) \tag{4}$$

$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i) \tag{5}$$

$$o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o) \tag{6}$$

$$\tilde{C}_t = \tanh(W_C x_t + U_C h_{t-1} + b_C) \tag{7}$$

$$C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t \tag{8}$$

$$h_t = o_t \odot \tanh(C_t) \tag{9}$$

The forget, input, and output gates of each LSTM block are defined by Eqs. 4–6, respectively, where f_t, i_t, and o_t are the forget, input, and output gates, respectively. The input gate decides which values should be updated, the forget gate allows forgetting and discarding the information, and the output gate together with the block output selects the outgoing information at time t. $\tilde{C}_t$ defined in Eq. 7 is the block input at time t which is a tanh layer and with the input gate, the two decides on the new information that should be stored in the cell state. C_t is the cell state at time t which is updated from the old cell state (Eq. 8). Finally, h_t is block output at time t.

The LSTM block is illustrated in Fig. 2. The three gates (input, forget, and output gates), and block input and block output activation functions are displayed in the figure. The output of the block is recurrently connected back to the block input and all of the gates. W and U are weight matrices, and b is the bias vector. The $\odot$ sign is the point-wise multiplication of two vectors. Functions σ and tanh are point-wise nonlinear logistic sigmoid and hyperbolic tangent activation functions, respectively.

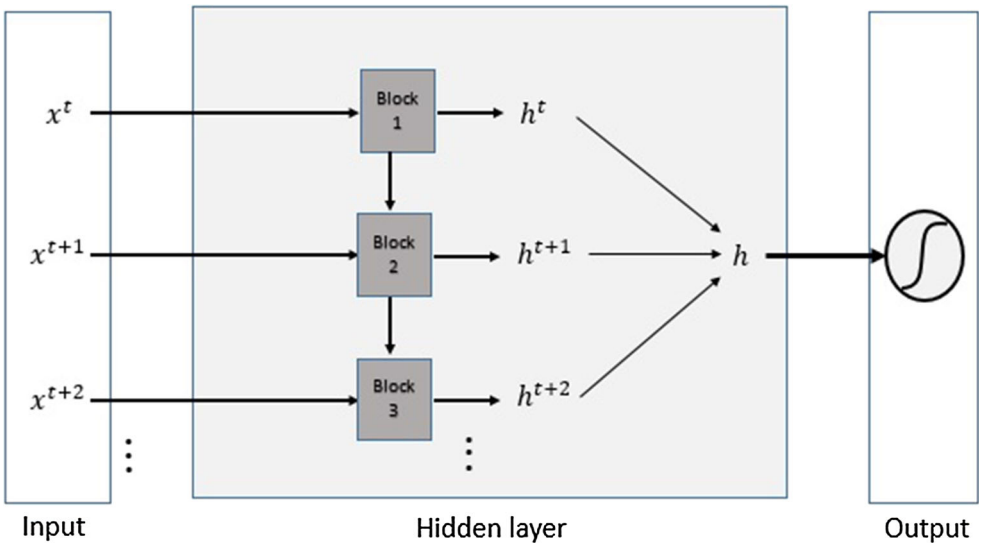

Fig. 1 The LSTM architecture consisting of the input layer, a single hidden layer, and the output layer [2]

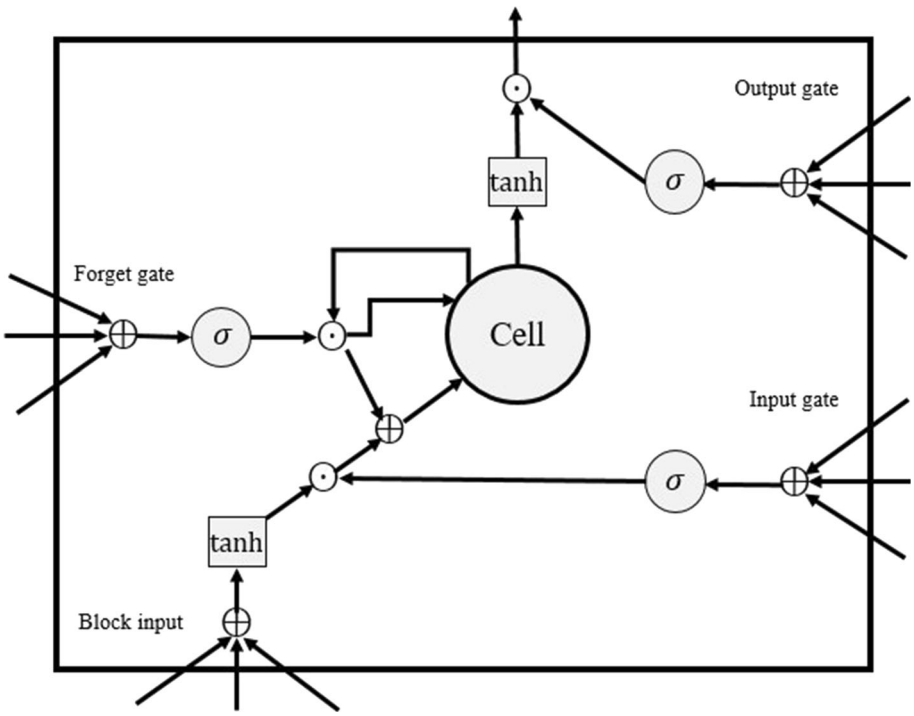

Fig. 2 A single LSTM block with tanh block input and output and with the sigmoidal gates shown with σ [2]. The $\odot$ sign is the point-wise multiplication

2.2 Activation functions

Three main aspects of neural network have important roles in network performance: the network architecture and the pattern of connections between units, the learning algorithm, and the activation functions used in the network. Most of the researches on analysis of the neural networks have focused on the importance of the learning algorithm, whereas the importance of the activation functions used in the neural networks has been mostly neglected [18–20].

We analyze the LSTM network in this paper by changing the activation functions of the forget, input, and output gates (sigmoidal gates of Eqs. 4, 5, and 6). We compare 23 different activation functions in terms of their effect on network performance when employed in sigmoidal gates of a basic LSTM block for classification.

Sigmoid and hyperbolic tangent functions are the most popular activation functions used in the neural networks. However, some individual studies have considered other activation functions in their research. We have compiled a comprehensive list of 23 such functions as shown in Table 1 and discussed below. We experimentally observed that adding a value of 0.5 to some functions makes them applicable as activation functions in the network. Changing

Table 1 Label, definition, corresponding derivative and range of each activation function

No.	Label	Activation function	Derivative function	Range
1	Aranda [18]	$f(x) = 1 - (1 + 2e^{x})^{-1/2}$	$f'(x) = e^{x}(2e^{x} + 1)^{-3/2}$	[0, 1]
2	Bi-sig1 [36]	$f(x) = \frac{1}{2}\left(\frac{1}{1+e^{-x+1}} + \frac{1}{1+e^{-x-1}}\right)$	$f'(x) = \frac{\frac{e^{1-x}}{(e^{1-x}+1)^2} + \frac{e^{-x-1}}{(e^{-x-1}+1)^2}}{2}$	[0, 1]
3	Bi-sig2 [36]	$f(x) = \frac{1}{2}\left(\frac{1}{1+e^{-x}} + \frac{1}{1+e^{-x-1}}\right)$	$f'(x) = \frac{\frac{e^{-x}}{(e^{-x}+1)^2} + \frac{e^{-x-1}}{(e^{-x-1}+1)^2}}{2}$	[0, 1]
4	Bi-tanh1 [36]*	$f(x) = \frac{1}{2}\left[\tanh\left(\frac{x}{2}\right) + \tanh\left(\frac{x+1}{2}\right)\right] + 0.5$	$f'(x) = \frac{\mathrm{sech}^2\left(\frac{x+1}{2}\right) + \mathrm{sech}^2\left(\frac{x}{2}\right)}{4}$	[− 0.5, 1.5]
5	Bi-tanh2 [36]*	$f(x) = \frac{1}{2}\left[\tanh\left(\frac{x-1}{2}\right) + \tanh\left(\frac{x+1}{2}\right)\right] + 0.5$	$f'(x) = \frac{\mathrm{sech}^2\left(\frac{x+1}{2}\right) + \mathrm{sech}^2\left(\frac{x-1}{2}\right)}{4}$	[− 0.5, 1.5]
6	Cloglog [22]	$f(x) = 1 - e^{-e^{x}}$	$f'(x) = e^{x - e^{x}}$	[0, 1]
7	Cloglogm [21]*	$f(x) = 1 - 2e^{-0.7e^{x}} + 0.5$	$f'(x) = 7e^{x - 0.7e^{x}}/5$	[− 0.5, 1.5]
8	Elliott [39]	$f(x) = \frac{0.5x}{1+\lvert x\rvert} + 0.5$	$f'(x) = \frac{0.5}{(1+\lvert x\rvert)^2}$	[0, 1]
9	Gaussian	$f(x) = e^{-x^2}$	$f'(x) = -2xe^{-x^2}$	[0, 1]
10	Logarithmic*	$f(x) = \begin{cases} \ln(1+x) + 0.5 & x \geq 0 \\ -\ln(1-x) + 0.5 & x < 0 \end{cases}$	$f'(x) = \begin{cases} \frac{1}{x+1} & x \geq 0 \\ \frac{1}{1-x} & x < 0 \end{cases}$	[− ∞, +∞]
11	Loglog [21]*	$f(x) = e^{-e^{-x}} + 0.5$	$f'(x) = e^{-e^{-x} - x}$	[0.5, 1.5]
12	Logsigm [20]*	$f(x) = \left(\frac{1}{1+e^{-x}}\right)^2 + 0.5$	$f'(x) = \frac{2e^{-x}}{(e^{-x}+1)^3}$	[0.5, 1.5]
13	Log-sigmoid	$f(x) = \frac{1}{1+e^{-x}}$	$f'(x) = \frac{e^{-x}}{(e^{-x}+1)^2}$	[0, 1]
14	Modified Elliott [41]	$f(x) = \frac{x}{\sqrt{1+x^2}} + 0.5$	$f'(x) = \frac{1}{(x^2+1)^{3/2}}$	[− 0.5, 1.5]
15	Rootsig [19]*	$f(x) = \frac{x}{1+\sqrt{1+x^2}} + 0.5$	$f'(x) = \frac{1}{\sqrt{x^2+1}+x^2+1}$	[− 0.5, 1.5]
16	Saturated*	$f(x) = \frac{\lvert x+1\rvert - \lvert x-1\rvert}{2} + 0.5$	$f'(x) = \frac{\frac{x+1}{\lvert x+1\rvert} - \frac{x-1}{\lvert x-1\rvert}}{2}$	[− 0.5, 1.5]
17	Sech	$f(x) = \frac{2}{e^{x}+e^{-x}}$	$f'(x) = -\frac{2(e^{x}-e^{-x})}{(e^{x}+e^{-x})^2}$	[0, 1]
18	Sigmoidalm [20]*	$f(x) = \left(\frac{1}{1+e^{-x}}\right)^4 + 0.5$	$f'(x) = \frac{4e^{-x}}{(e^{-x}+1)^5}$	[0.5, 1.5]
19	Sigmoidalm2 [42]*	$f(x) = \left(\frac{1}{1+e^{-x/2}}\right)^4 + 0.5$	$f'(x) = \frac{2e^{-x/2}}{(e^{-x/2}+1)^5}$	[0.5, 1.5]
20	Sigt [37]	$f(x) = \frac{1}{1+e^{-x}} + \frac{1}{1+e^{-x}}\left(1 - \frac{1}{1+e^{-x}}\right)$	$f'(x) = \frac{2e^{x}}{(e^{x}+1)^3}$	[0, 1]
21	Skewed-sig [38]*	$f(x) = \left(\frac{1}{1+e^{-x}}\right)\left(\frac{1}{1+e^{-2x}}\right) + 0.5$	$f'(x) = \frac{(e^{2x}+2e^{x}+3)e^{3x}}{(e^{x}+1)^2(e^{2x}+1)^2}$	[0.5, 1.5]
22	Softsign [39]*	$f(x) = \frac{x}{1+\lvert x\rvert} + 0.5$	$f'(x) = \frac{1}{(1+\lvert x\rvert)^2}$	[− 0.5, 1.5]
23	Wave [40]	$f(x) = (1 - x^2)e^{-x^2}$	$f'(x) = 2x(x^2 - 2)e^{-x^2}$	[− 0.055, 1]

* 0.5 added to the original function

the range of the activation functions is previously observed in other studies [41].

In Table 1, the first activation function is Aranda-Ordaz introduced by Gomes et al. [18] which is labeled as Aranda. The second to fifth functions are the bimodal activation functions proposed by Singh et al. [36] and labeled as *Bi-sig1*, *Bi-sig2*, *Bi-tanh1*, and *Bi-tanh2*, respectively. The sixth function is the complementary log–log [22]. The next function presents a modified version of cloglog, named *cloglogm* [21]. Next come the *Elliott, Gaussian, logarithmic*, and *log–log* functions, the 12th function is a modified logistic sigmoid function proposed by Singh and Chandra [20] labeled as *logsigm*. The logistic sigmoid comes next as called *log-sigmoid*, followed by the *modified Elliott* function. The 15th function is a sigmoid function with roots [19], called *rootsig*. The 16th to 19th functions are the *Saturated*, the hyperbolic secant (*Sech*), and two modified sigmoidals labeled as *sigmoidalm* and *sigmoidalm2*. The tunable activation function proposed by Yuan et al. [37] and labeled as *sigt* is the 20th function. Next is a skewed derivative activation function proposed by Chandra et al. [38] labeled as *skewed-sig*. The *softsign* function proposed by Elliott [39] and the *wave* function proposed by Hara and Nakayamma [40] come last. Some other activation functions such as rectifier [43] were applied in the network but turned out to be ineffective due to the exploding gradient problem.

Table 2 Average train errors per each activation function for the Movie Review data set

Activation function	No. of hidden blocks						
	2	4	8	16	32	64	Average test error (95% CI)
Aranda	11.8	8.43	7.16	**4.06**	6.03	7	23.24 (22.45–24.02)
Bi-sig1	11.53	9.1	6.66	5.53	6.1	**5.3**	23.85 (23.45–24.25)
Bi-sig2	9.03	8	7.93	**6.96**	8.16	7.16	23.72 (22.9–24.53)
Bi-tanh1	7.4	5.96	6.86	**4.73**	6.2	7.76	23.1 (22.44–23.76)
Bi-tanh2	9.86	8.33	7.53	**7.26**	8.1	8.13	23.1 (22.8–23.41)
Cloglog	8.26	7.46	6.13	5.4	**4.56**	8.73	23.3 (22.66–23.95)
Cloglogm	9.7	**7.26**	7.83	8.33	9.1	7.32	22.62 (21.11–24.13)
Elliott	9.36	10.5	**5.86**	6.7	7.56	7.46	23.2 (22.02–24.37)
GAUSSIAN	6.83	6.03	8.26	**4.66**	8.2	6.15	23.53 (23.13–23.93)
Logarithmic	11.23	9.4	7.7	**5.93**	7.13	8.53	23.12 (22.33–23.9)
Loglog	8.56	6.6	6.76	6.43	5	**2.96**	23.9 (23.65–24.15)
Logsigm	7.86	7.63	7.03	**4.53**	5.3	5.26	23.01 (22.46–23.56)
Log-sigmoid	10.7	10.06	6.56	4.73	**3.93**	4.46	23.52 (23.16–23.87)
Modified Elliott	9.46	9.06	6.5	**5.53**	5.66	6.12	*22.52* (21.05–23.98)
Rootsig	11.53	9.5	6.53	6.43	**6.16**	6.84	23.22 (22.17–24.27)
Saturated	10.26	10.43	7.23	7.83	**4.56**	7.5	23.06 (22.25–23.87)
Sech	10.53	7.43	8.1	**5.03**	5.1	5.83	23.85 (23.04–24.66)
Sigmoidalm	9.46	9	**6.53**	7.13	8.1	8.1	23.33 (23.04–23.62)
Sigmoidalm2	9.06	7.53	7.2	8.66	5.76	**5.63**	24.48 (24.04–24.91)
Sigt	8.3	7.23	**6.06**	6.16	6.26	6.83	23.24 (22.63–23.84)
Skewed-sig	8.83	8.1	7.63	**6.8**	6.86	7.06	23.09 (22.97–23.20)
Softsign	11.86	8.26	6	**4.93**	5	6.66	22.85 (22.34–23.36)
Wave	10.93	7.6	5.56	7.66	7.23	**4.53**	23.9 (22.98–24.82)

The minimum train error for each function is shown in bold face. The last column shows the test error for the network configuration which produced the least training error in each row. The minimum test error is italic

2.3 Methodology

To evaluate the effect of different activation functions on the classification performance, we vary the activation of the input, output, and forget gates which we refer to as sigmoidal gates, and keep the tanh units unchanged. In each configuration, all the three sigmoidal gates are identical and chosen from the set of activation functions introduced in Table 1.

To train the network, the back propagation through time algorithm (BPTT) [34] is used with either ADADELTA [35] or RMSprop [44] as the optimization method. ADADELTA is a gradient descent-based learning algorithm which is proposed as an improvement over Adagrad [45] and adapts the learning rate per parameter over time. RMSprop is also an extension of Adagrad that deals with its radically diminishing learning rates. The two optimization methods are popular for LSTM networks and achieve faster convergence rates [46, 47].

The mini-batch method is used for the training and test phases. The network is trained and tested three times for each activation function with the same train and test data. The initial network weights and the batches are chosen randomly in each experiment. The error interval is reported using the results of the three experiments of each configuration. The train and validation errors are measured at the end of each batch. The dropout method with probability of 0.5 is used to prevent overfitting [48]. The network is trained until a low and approximately constant classification error based on training data is observed, and also the validation error is stable for 10 consecutive batches. The test errors at this stage are reported.

3 Experimental results

To analyze the performance of the LSTM network, two sets of experiments are designed with different types of data sets. In both set of experiments different architectures of LSTM are evaluated and in each configuration the input, output, and forget gates of the LSTM blocks use an identical activation function from Table 1. In what follows, we describe the analysis results.

Table 3 Average train errors per each activation function for the IMDB data set

Activation function	No. of hidden blocks							
	4	8	16	32	64	128	256	Average test error (95% CI)
Aranda	3.46	1.3	2.23	**1**	1.03	1.1	3.16	13.6 (12.28–14.91)
Bi-sig1	1.9	1.4	1.2	1.3	1.3	1.23	**1.16**	13.93 (12.68–15.18)
Bi-sig2	1.73	2.4	1.76	1.43	1.46	1.63	**1.4**	13.93 (11.48–16.38)
Bi-tanh1	2.1	1.16	1.93	1.13	1.96	**1.03**	4.03	13.93 (12.18–15.67)
Bi-tanh2	4.9	1.86	1.23	1.66	1.83	**1**	1.2	14.06 (12.81–15.31)
Cloglog	2.13	2.7	1.53	1.36	2.66	1.5	**1.23**	13.4 (11.67–15.12)
Cloglogm	4.33	2.5	2.4	2.63	1.4	**1.16**	1.33	13.13 (12.37–13.89)
Elliott	1.53	1.5	**0.93**	1.2	1.06	1.26	4.16	14.06 (12.91–15.21)
Gaussian	3.13	**1.6**	2.66	2.36	5.23	18.46	11.7	14.8 (10.38–19.21)
Logarithmic	1.56	3.06	1.5	1.13	**1**	1.06	2.73	13.6 (13.6–13.6)
Loglog	4.73	**2.4**	2.5	3.3	3.2	22.33	24.3	19.6 (7.76–31.43)
Logsigm	1.8	3.86	1.63	1.26	**1.16**	3.16	11.3	14.53 (13.38–15.68)
Log-sigmoid	1.3	1.5	1.4	2.46	2.56	1.43	**1.23**	13.6 (12.73–14.46)
Modified Elliott	2.3	3.43	1.66	5.1	1.66	1.56	**1.43**	*12.46* (10.22–14.7)
Rootsig	1.76	1.83	1.23	1.63	1.06	1.36	**1.06**	13.4 (12.08–14.71)
Saturated	3.86	1.9	1.63	1.83	1.6	**1.15**	1.16	13.06 (10.71–15.41)
Sech	**1.1**	2.1	2.26	1.23	3.1	6.36	22.06	17.73 (14.99–20.46)
Sigmoidalm	1.36	1.66	2.63	2.3	3.16	**1.03**	2.53	13.6 (12.1–15.09)
Sigmoidalm2	1.6	2.76	2.13	1.1	**1.06**	1.96	1.36	14.13 (13.37–14.89)
Sigt	1.73	1.53	**1.03**	4.16	1.23	2.13	3.13	15.2 (12.71–17.68)
Skewed-sig	3.3	2.36	2.43	**1.66**	1.73	3.23	17.7	16.06 (12.87–19.26)
Softsign	2.5	1.26	2.4	1.8	1.3	**1.23**	3.5	13.13 (12.09–14.16)
Wave	7	8.73	10.36	**4.6**	12.8	35.2	31.06	16.2 (13.92–18.47)

The minimum train error for each function is shown in bold face. The last column shows the test error for the network configuration which produced the least training error in each row. The minimum test error is italic

3.1 First set of experiments

In the first set of experiments, we use two movie review data sets. The first one [49] is referred to as Movie Review[1] in this paper, and the other is the IMDB large movie review data set[2] [50]. The Movie Review data set consists of 10,662 review sentences, with equal number of positives and negatives. From this data set, we use a total of 8162 sentences in the training and the rest are used in the test phase. Both sets contain equal number of positive and negative sentences. From the IMDB data set, we use 2000 sentences for training the network (with portion of 5% for validation set) and 500 sentences for testing the performance. Again, the number of positive and negative sentences is equal and the sentences have a maximum length of 100 words.

The mini-batch method is used with the batch size for the training and test phases set to 16 and 64, respectively. The batch sizes have been chosen based on experiment for producing a better performance. We use the backpropagation through time algorithm (BPTT) with ADADELTA as the optimization method, with the epsilon parameter set to 1e−6. Hyperparameters are not tuned specifically for each configuration of LSTM and are identical in all experiments. The test misclassification error is used to rank the activation functions. Each experiment is repeated three times.

Table 2 illustrates the average training error values on the Movie Review data set for different configurations. In these set of experiments, the number of LSTM blocks in the hidden layer increases exponentially from 2 to 64 and the number of epochs on each run is set to 20 (in each epoch, all the training data are exposed to the network in mini-batches). For each activation function, the average test error for the configuration which has produced the least train error is shown in the last column. The test errors for all configurations are presented in "Appendix." As observed, on this data set the modified Elliott has the least average test error (22.52%). Overall, the *modified Elliott* (with range of [− 0.5, 1.5]), *cloglogm* ([− 0.5, 1.5]),

[1] http://www.cs.cornell.edu/people/pabo/movie-review-data/rt-polaritydata.tar.gz.

[2] http://ai.stanford.edu/amaas/data/sentiment/.

Table 4 Average test and train errors for Movie Review data set, and average number of convergence epochs for 16 blocks in hidden layer of LSTM

Activation function	Average test error	Average train error	Average convergence epochs
Aranda	23.24	4.06	17.66 (144,140.92)
Bi-sig1	23.52	5.53	16.66 (135,978.92)
Bi-sig2	23.72	6.96	9.66 (78,844.92)
Bi-tanh1	23.1	4.73	10 (81,620)
Bi-tanh2	23.1	7.26	12.33 (100,637.46)
Cloglog	23	5.4	10.33 (84,313.46)
Cloglogm	23.01	8.33	12.33 (100,637.46)
Elliott	23.33	6.7	15.33 (125,123.46)
Gaussian	23.53	4.66	13.66 (111,492.92)
Logarithmic	23.12	5.93	14.33 (116,961.46)
Loglog	23.1	6.43	10.66 (87,006.92)
Logsigm	23.01	4.53	11.66 (95,168.92)
Log-sigmoid	23.78	4.73	16.33 (133,285.46)
Modified Elliott	22.52	5.53	12.33 (100,637.46)
Rootsig	23.1	6.43	12.66 (103,330.92)
Saturated	22.8	7.83	13 (106,106)
Sech	23.85	5.03	14.66 (119,654.92)
Sigmoidalm	23.7	7.13	10.66 (87,006.92)
Sigmoidalm2	23.88	8.66	10 (81,620)
Sigt	23.36	6.16	10.33 (84,313.46)
Skewed-sig	23.09	6.8	8.66 (70,682.92)
Softsign	22.85	4.93	15.33 (125,123.46)
Wave	22.82	7.66	12 (97,944)

The number of iterations reported in parenthesis

softsign ([− 0.5, 1.5]), *logsigm* ([0.5, 1.5]), and *saturated* ([− 0.5, 1.5]) functions when used as activation present the least average error values which are 22.52, 22.62, 22.85, 23.01, and 23.06%, respectively. The optimum number of LSTM blocks on the hidden layer when using *modified Elliott* was 16 units, while for *cloglogm*, *softsign*, *logsigm*, and saturated it was 4, 16, 16, and 32 units, respectively. Interestingly *log-sigmoid* stands in rank 17 among all functions, with the average error of 23.52%. For the Movie Review data set, training errors have negative correlation with number of units for most functions (although the correlations are mostly weak). Most activations perform poorly with a very low number of units (e.g., 2). But, as number of units increase, the sensitivity of error values to number of units is less observed and the standard deviation of error values for most activations is less than 2.

Results of the training error values for the IMDB data set are illustrated in Table 3 including the average test error for the best configuration of each activation. The number of LSTM units in the hidden layer was modified exponentially from 4 to 256. The number of epochs on each run was 50. On this data set, similar to the first data set, *modified Elliott* had the least average error (12.46%). After *modified Elliott* (with range of [− 0.5, 1.5]), the *saturated* ([− 0.5, 1.5]), *cloglogm* ([− 0.5, 1.5]), and *softsign* ([− 0.5, 1.5]), functions have the least average error values of 13.06, 13.13, and 13.13%, respectively. The optimum number of LSTM blocks on the hidden layer for *modified Elliott* was 256 units, while for *cloglogm*, *saturated*, and *softsign* it was 128, 128, and 128 units, respectively. Interestingly again, the *log-sigmoid* does not appear in the four best results and rank of this activation function is 10 out of 23 with the average error of 13.6%. For the IMDB data set, no solid pattern of correlation is observed between error values and number of units.

Most functions produced best training results with 16 and 256 blocks for the Movie Review and IMDB data sets, respectively. Average test and train errors, and average number of convergence epochs for Movie Review and IMDB data sets are represented in Tables 4 and 5, respectively. Note that in each epoch all of the train data are represented to the network in a sequence of mini-batches. The number of actual (average) iterations is reported in parentheses. To test the significance of the results, ANOVA tests were conducted for results of 16 and 256 blocks for the Movie Review and IMDB data sets, respectively. The obtained p values were 3.33e−6 and 7.61e−11, which being less than 0.05 indicate the significance of the results.

Results show that for both data sets, the activation function *modified Elliott* has the best performance. Using this activation function for large data sets, we may need to

Table 5 Average test and train errors for IMDB data set, and average number of convergence epochs for 256 blocks in hidden layer of LSTM

Activation function	Average test error	Average train error	Average convergence epochs
Aranda	13.93	3.16	23 (46,000.0)
Bi-sig1	13.93	1.16	28 (56,000.0)
Bi-sig2	13.93	1.40	23 (46,000.0)
Bi-tanh1	15.2	4.03	35 (70,000.0)
Bi-tanh2	13.8	1.20	29 (58,000.0)
Cloglog	13.4	1.23	26 (52,000.0)
Cloglogm	13	1.33	27 (54,000.0)
Elliott	13.73	4.16	25 (50,000.0)
Gaussian	20.93	11.70	45 (90,000.0)
Logarithmic	13.33	2.73	22 (44,000.0)
Loglog	35.93	24.3	44 (88,000.0)
Logsigm	25.93	11.30	47 (94,000.0)
Log-sigmoid	13.6	1.23	25 (50,000.0)
Modified Elliott	12.46	1.43	28 (56,000.0)
Rootsig	13.4	1.06	23 (46,000.0)
Saturated	12.86	1.16	29 (58,000.0)
Sech	32.2	22.06	47 (94,000.0)
Sigmoidalm	14.2	2.53	19.66 (39,320.0)
Sigmoidalm2	14.6	1.36	24 (48,000.0)
Sigt	15.53	3.13	46 (92,000.0)
Skewed-sig	29.93	17.70	47 (94,000.0)
Softsign	13	3.50	20 (40,000.0)
Wave	34.93	31.06	45 (90,000.0)

The number of iterations reported in parenthesis

tune the hyperparameters such as using a smaller epsilon parameter in ADADELTA optimization method. Interestingly, the *log-sigmoid* activation function which is commonly used in neural networks and in LSTM networks does not produce the best results and the *modified Elliott* function demonstrates better results when employed in the sigmoidal gates. Additionally, it was observed that sigmoidal activations with range of $[-0.5, 1.5]$ result in a more accurate network than those in the range of [0, 1] in LSTM network. The maximum length of sentences for the Movie Review and IMDB data sets used in the experiments were 64 and 100, respectively. When applied on the IMDB data set, LSTM network required more hidden blocks and even more epochs per run. This can be justified by the greater complexity of this data set.

The error levels measured in current study are consistent with some other studies in the literature. Lenc and Hercig [51] report a 38.3% error for classification of Movie Review with LSTM. Dai and Le [52] report an error of 13.5% for classification of IMDB data with LSTM. The overall error difference for all functions is at most 2 and 5% in Movie Review and IMDB data sets, respectively. In the experiments, the difference in the best measured error values of the *modified Elliott* function and the popular *log-sigmoid* function is 0.36 and 1.14 for the Movie Review and IMDB data sets, respectively. Although being small, these values can be meaningful according to the specific application.

3.2 Second set of experiments

For the second experiment we use the MNIST[3] data set of handwritten digits. The mini-batch method is again used with the batch size for the training and test phases set to 128. The batch sizes have been chosen based on experiment. We use the RMSprop as the optimization method, with the learning rate set to 0.001.

The MNIST data set of handwritten digits has a training set of 60,000 examples (with 5000 examples for validation), and a test set of 10,000 examples. The image sizes are 28×28 pixels. We use the one-hot method to predict 10 digits (0–9) or equivalently 10 classes.

Table 6 illustrates the average training error values for the MNIST data set with the test error being reported for the best configuration of each activation function. In these set of experiments, two configurations of 64 and 128 LSTM blocks in the hidden layer are considered, and the number of epochs for each run is set to six. As observed, on

[3] http://yann.lecun.com/exdb/mnist/.

Table 6 Average train errors per each activation function for the MNIST data set

Activation function	No. of hidden blocks		
	64	128	Average test error (95% CI)
Aranda	2.11	**1.82**	2.10 (1.85–2.34)
Bi-sig1	1.95	**1.76**	2.00 (1.75–2.24)
Bi-sig2	1.86	**1.73**	2.03 (1.65–2.41)
Bi-tanh1	**2.48**	2.56	3.03 (2.74–3.32)
Bi-tanh2	1.93	**1.72**	1.93 (1.55–2.31)
Cloglog	2.23	**1.88**	2.26 (1.88–2.64)
Cloglogm	2.18	**2.06**	2.13 (1.75–2.51)
Elliott	1.75	**1.62**	*1.66* (1.28–2.04)
Gaussian	2	**1.75**	1.96 (1.67–2.25)
Logarithmic	**2.6**	3.14	2.80 (2.14–3.45)
Loglog	**4.81**	7.68	5.46 (4.94–5.98)
Logsigm	4.25	**4.21**	4.53 (3.59–5.47)
Log-sigmoid	2.15	**2**	2.16 (1.59–2.74)
Modified Elliott	2.21	**2.1**	2.03 (1.74–2.32)
Rootsig	2	**1.9**	1.90 (1.65–2.14)
Saturated	**3.32**	3.76	3.43 (3.14–3.72)
Sech	2.3	**2.22**	2.06 (1.68–2.44)
Sigmoidalm	**2.13**	2.22	2.23 (1.47–2.99)
Sigmoidalm2	2.56	**2.42**	2.66 (2.14–3.18)
Sigt	2.41	**2.37**	2.76 (2.47–3.05)
Skewed-sig	4.89	**4.56**	4.73 (4.35–5.11)
Softsign	1.7	**1.62**	*1.66* (1.52–1.81)
Wave	2.37	**2.25**	2.26 (1.88–2.64)

The minimum train error for each function is shown in bold face. The last column shows the test error for the network configuration which produced the least training error in each row. The minimum test error is italic

this data set *Elliott* and *softsign* have the least average error (1.66%). Overall, the *softsign* (with range of [− 0.5, 1.5]) and *Elliott* (with range of [0, 1]), *rootsig* ([− 0.5, 1.5]), *Bi-tanh2* ([− 0.5, 1.5]), *Gaussian* ([0, 1]), *Bi-sig1* ([0, 1]), *Bi-sig2* ([0, 1]), and *modified Elliott* ([− 0.5, 1.5]) functions when used as activation present the least average error values which are 1.66%, 1.66, 1.9, 1.93, 1.96, 2, 2.03, and 2.03% respectively. The optimum number of LSTM blocks on the hidden layer for *softsign, Elliott, rootsig, Bi-tanh2, Gaussian, Bi-sig1, Bi-sig2*, and *modified Elliott* was 128 units, and it seems that most of the activation functions worked better with this number of units Interestingly, *log-sigmoid* stands in rank 10 with the average error of 2.16%.

Average test and train errors, and average number of convergence epochs for the MNIST data set, for 128 blocks in the hidden layer are represented in Table 7. The ANOVA result for all experiments with 128 blocks is 3.7e−21 which shows the results are significant. The error levels are consistent with the study of Arjovsky et al. [53]. which report a classification error of 1.8% for MNIST data with LSTM.

3.3 Discussion

In this paper we aggregated a list of 23 applicable activation functions that can be used in place of the sigmoidal gates in a LSTM network. We compared performance of the network using these functions with different number of hidden blocks, in classification tasks. The results showed the following:

1. Overall, the results on both data sets suggest that less-recognized activation functions (such as *Elliott*, *modified Elliott*, and *softsign* which are interestingly all in the Elliott family) can produce more promising results compared to the common functions in the literature.
2. Activation functions with the range of [− 0.5, 1.5] have generally produced better results, and this indicates that a wider range of codomain (than the sigmoidal range of [0, 1]) can yield a better performance.
3. The *log-sigmoid* activation function which is mostly used in LSTM blocks produces weak results compared to other activation functions.

Burhani et al. [41] in their study on denoising autoencoders reported a similar result that the modified Elliott has a better performance and less error than log-sigmoid activation function. In addition, in the first set of experiments we found cloglogm to be the second best activation which is consistent with Gomes et al. [21] stating that cloglogm shows good results for forecasting financial time series. The top activations (*Elliott family and cloglogm*) along with the popular *log-sigmoid* activation are displayed in Fig. 3. According to the diagram, *modified Elliott, softsign*, and *cloglogm* are much steeper than *log-sigmoid* around zero and also have a wider range. In Fig. 4 performance comparison of these five functions in term of the average error value for the Movie Review, IMDB, and MNIST data sets, respectively, for 16, 256 and 128 blocks is illustrated.

There are two widely known issues with training the recurrent neural networks, the vanishing and the exploding gradient problems [54]. The LSTM networks alleviate the gradient vanishing problem by their special design. The gradient exploding problem can, however, still occur. A gradient norm clipping strategy is proposed by Pascanu et al. [55] to deal with exploding gradients. Gradient clipping is a technique to prevent exploding gradients in very deep networks. A common method is to normalize the gradients of a parameter vector when its L2 norm exceeds a certain threshold [55]. Although we have not performed gradient norm clipping in training the LSTM network, the method suggests that gradient exploding problem is closely related to norm of the gradient matrix and smaller norms are preferred.

Table 7 Average test and train errors for MNIST data set, and average number of convergence epochs for 128 blocks in hidden layer of LSTM

Activation function	Average test error	Average train error	Average convergence epochs
Aranda	2.1	1.82	3.55 (195,413.3)
Bi-sig1	2.00	1.76	3.95 (217,600.0)
Bi-sig2	2.03	1.73	3.94 (216,746.7)
Bi-tanh1	2.50	2.56	4.01 (220,586.7)
Bi-tanh2	1.93	1.72	3.43 (183,893.3)
Cloglog	2.26	1.88	3.90 (214,613.3)
Cloglogm	2.13	2.06	4.12 (226,560.0)
Elliott	*1.66*	1.62	3.97 (218,453.3)
Gaussian	1.96	1.75	3.50 (192,853.3)
Logarithmic	2.70	3.14	4.25 (234,240.0)
Loglog	8.16	7.68	4.19 (230,400.0)
Logsigm	4.53	4.21	4.22 (232,533.3)
Log-sigmoid	2.16	2.00	3.14 (172,800.0)
Modified Elliott	2.03	2.10	4.15 (228,266.7)
Rootsig	1.90	1.90	3.39 (186,880.0)
Saturated	3.83	3.76	4.11 (226,133.3)
Sech	2.06	2.22	3.79 (208,640.0)
Sigmoidalm	2.36	2.22	3.54 (194,986.7)
Sigmoidalm2	2.66	2.42	3.27 (180,053.3)
Sigt	2.76	2.37	3.90 (215,040.0)
Skewed-sig	4.73	4.56	4.26 (234,666.7)
Softsign	*1.66*	1.62	4.16 (229,120.0)
Wave	2.26	2.25	3.48 (191,573.3)

The number of iterations reported in parenthesis

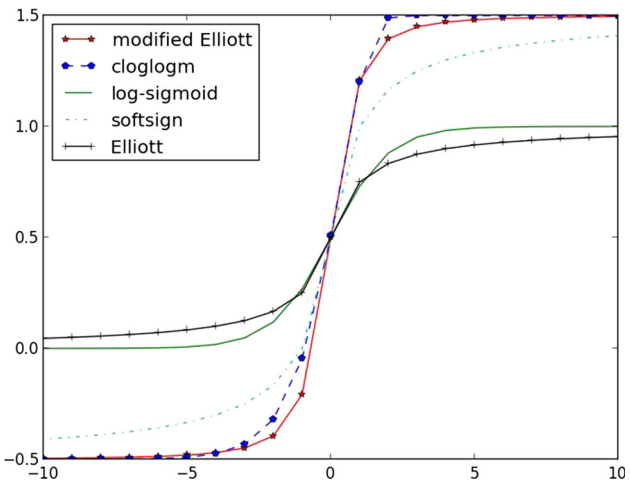

Fig. 3 The modified Elliott, cloglogm, log-sigmoid, softsign, and Elliott activation functions

We evaluated the norm of the gradient matrix in the second set of experiments, and interestingly observed that the norm of the gradient matrix for the Elliott activation was low and in fact the second lowest among all the activations. The norms of the gradient matrix after convergence are presented in Table 8. As observed, the norm of gradient matrix for most of the activation functions achieving lower classification errors is considerably low (less than 0.1).

The tanh function is one of the most popular activation functions which is widely used in LSTM networks [2]. From a conceptual point of view, two tanh activations in LSTM blocks squash the block input and output and can be considered to have a different role from the three gates. However, they can indeed have a significant effect on the overall network performance and can be replaced by other activations which fulfill the properties mentioned in Sect. 1 of the manuscript. This change will specifically affect the gradient, range, and derivative of the activation functions and blocks. Analyzing the effect of other activation functions when used in place of tanh activations is left for future work.

Some follow-up studies have proposed modifications on the initial LSTM architecture. Evaluating different activation functions on these architectures can serve as an interesting future study. Gers and Schmidhuber [56] introduced peephole connections that cross directly from the internal state to the input and output gates of a node. According to their observations, these connections improve performance on timing tasks where the network must learn to measure precise intervals between events. Another line of research is the alternate and similar architectures which are popular along with the LSTM. The bidirectional

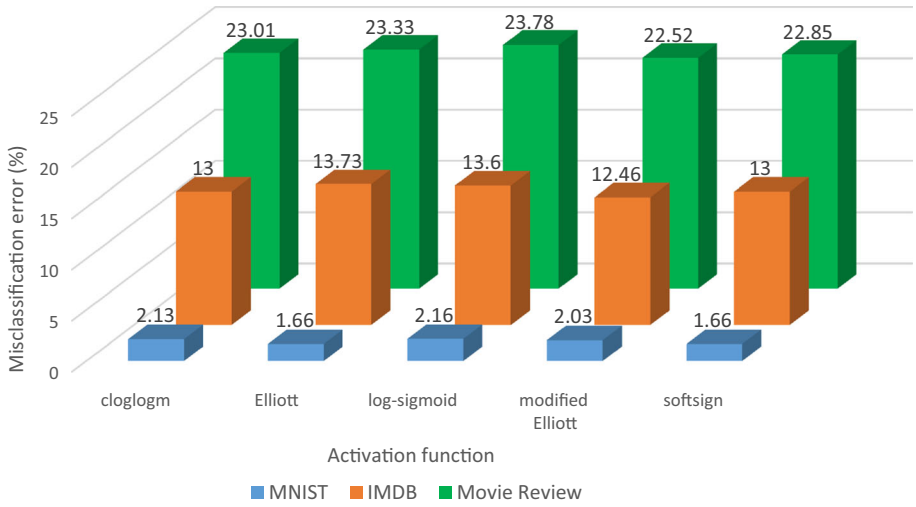

Fig. 4 Comparison of the average error values for the cloglogm, Elliott, log-sigmoid, modified Elliott, and softsign activation functions for MNIST, IMDB and Movie Review data sets with 128, 256, and 16 blocks, respectively

Table 8 Norm of gradient matrix for MNIST data set in increasing order

Activation function	Gradient norm	Activation function	Gradient norm
Sigmoidalm2	0.0172	Gaussian	0.1206
Elliott	0.0234	Bi-sig2	0.1228
Bi-tanh2	0.0346	Logarithmic	0.1304
Aranda	0.0346	Sech	0.1646
Saturated	0.0443	Sigmoidalm	0.2171
Modified Elliott	0.047	Skewed-sig	0.2885
Sigt	0.0548	Logsigm	0.4368
Rootsig	0.0634	Cloglogm	0.4824
Bi-sig1	0.0798	Cloglog	0.5623
Softsign	0.0964	Loglog	0.8145
Log-sigmoid	0.1004	Wave	0.9476
Bi-tanh1	0.1098		

recurrent neural network (BRNN) is first proposed by Schuster and Paliwal [57]. This architecture involves two layers of hidden nodes, both of which are connected to input and output. The first hidden layer has recurrent connections from the past time steps, and in the second layer direction of recurrent of connections is flipped. A gated recurrent unit (GRU) was proposed by Cho et al. [58] to make each recurrent unit to adaptively capture dependencies of different time scales. These modifications can improve performance of the network.

4 Conclusions

In LSTM blocks, the two most popular activation functions are sigmoidal and hyperbolic tangent. In this study we evaluated the performance of a LSTM network with 23 different activation functions that can be used in place of the sigmoidal gates. We varied the number of hidden blocks in the network and employed three different data sets for classification. The results exposed that some less-recognized activations such as the Elliott function and its modifications can yield less error levels compared to the most popular functions.

More research is needed to study other parts and details of an LSTM network such as the effect of changing the hyperbolic tangent function on the block input and block output. Variants of the LSTM network can also be analyzed. Additionally, larger data sets and different tasks can be employed to further analyze the network performance considering different configurations.

Compliance with ethical standards

Conflict of interest We declare that we do not have any commercial or associative interest that represents a conflict of interest in connection with the work submitted.

Appendix

See Tables 9, 10 and 11.

Table 9 Average test error values per each activation function for the Movie Review data set

Activation function	No. of hidden blocks					
	2	4	8	16	32	64
Aranda	23.56 (23.03–24.08)	23.68 (22.28–25.07)	23.62 (22.27–24.98)	**23.24** (22.45–24.02)	23.88 (23.28–24.47)	23.74 (22.78–24.71)
Bi-sig1	23.97 (22.26–25.67)	23.45 (22.36–24.54)	**23.33** (23.01–23.65)	23.52 (22.51–24.52)	23.86 (23.51–24.21)	23.85 (23.45–24.25)
Bi-sig2	**23.01** (22.16–23.85)	23.09 (21.13–25.05)	23.24 (22.8–23.67)	23.72 (22.9–24.53)	23.92 (22.52–25.31)	23.86 (22.58–25.14)
Bi-tanh1	22.81 (21.69–23.93)	23.01 (22.11–23.91)	**22.8** (21.59–24)	23.1 (22.44–23.76)	23.26 (22.11–24.41)	23.2 (22.76–23.63)
Bi-tanh2	23.26 (22.32–24.21)	**22.92** (22–23.83)	23.06 (22.23–23.9)	23.1 (22.8–23.41)	23 (22.64–23.35)	23.4 (22.53–24.26)
Cloglog	23.37 (22.45–24.29)	23.06 (22.36–23.77)	**22.98** (22.37–23.59)	23 (22.39–23.6)	23.3 (22.66–23.95)	23.5 (22.77–24.23)
Cloglogm	22.93 (22.61–23.25)	**22.62** (21.11–24.13)	22.98 (22.37–23.59)	23.01 (22.63–23.38)	22.89 (22.14–23.63)	22.84 (21.88–23.79)
Elliott	23.8 (22.05–25.54)	23.97 (20.85–27.09)	**23.2** (22.02–24.37)	23.33 (22.48–24.17)	24.17 (23.77–24.57)	24.05 (23.67–24.42)
Gaussian	**22.88** (21.25–24.51)	23.58 (23.29–23.87)	24.13 (22.73–25.52)	23.53 (23.13–23.93)	23.85 (23.27–24.43)	24.22 (23.11–25.33)
Logarithmic	23.69 (22.24–25.14)	22.9 (21.91–23.9)	22.94 (22.54–23.34)	23.12 (22.33–23.9)	**22.88** (22.78–22.97)	22.94 (22.54–23.34)
Loglog	23.16 (21.78–24.53)	23.16 (22.6–23.71)	23.48 (23.21–23.74)	**23.1** (22.85–23.35)	23.38 (23.03–23.73)	23.9 (23.65–24.15)
Logsigm	23.16 (22.55–23.76)	23.36 (22.6–24.11)	23.24 (22.03–24.44)	**23.01** (22.46–23.56)	23.54 (22.94–24.14)	24.08 (23.29–24.86)
Log-sigmoid	**22.74** (22.27–23.21)	23.68 (21.29–26.06)	23.52 (22.15–24.88)	23.78 (23.08–24.49)	23.52 (23.16–23.87)	23.65 (23.5–23.8)
Modified Elliott	***22.36*** (21.41–23.3)	23.33 (22.27–24.39)	22.68 (21.53–23.82)	22.52 (21.05–23.98)	23.06 (22.65–23.48)	23.57 (22.97–24.17)
Rootsig	**22.84** (21.53–24.14)	23.12 (22.34–23.89)	23.45 (23.2–23.7)	23.1 (22.99–23.22)	23.22 (22.17–24.27)	23.32 (22.01–24.62)
Saturated	24.09 (21.02–27.16)	23.72 (20.83–26.6)	23.04 (21.79–24.28)	**22.8** (22.7–22.89)	23.06 (22.25–23.87)	23.29 (22.37–24.21)
Sech	24.24 (22.65–25.82)	**23.62** (22.88–24.37)	23.85 (22.81–24.88)	23.85 (23.04–24.66)	23.89 (23.32–24.45)	24.52 (22.62–26.41)
Sigmoidalm	**23.22** (22.55–23.9)	23.4 (22.68–24.11)	23.33 (23.04–23.62)	23.7 (22.87–24.54)	23.66 (22.85–24.47)	24.25 (23.68–24.81)
Sigmoidalm2	23.56 (22.15–24.96)	**23.49** (23.11–23.86)	23.85 (23.28–24.41)	23.88 (22.57–25.18)	24.14 (23.25–25.03)	24.48 (24.04–24.91)
Sigt	**23.12** (22.14–24.09)	23.17 (21.89–24.45)	23.24 (22.63–23.84)	23.36 (22.92–23.79)	23.6 (23.08–24.11)	23.82 (23.21–24.43)
Skewed-sig	23.26 (22.1–24.42)	**22.92** (22.65–23.18)	23.09 (21.34–24.84)	23.09 (22.97–23.20)	23.09 (21.06–25.12)	23.34 (21.99–24.7)
Softsign	23.81 (22.36–25.26)	23.14 (21.55–24.74)	22.94 (21.79–24.09)	**22.85** (22.34–23.36)	23.14 (22.22–24.06)	23.12 (21.97–24.26)
Wave	23.69 (19.46–27.92)	23.3 (22.25–24.35)	23.18 (22.78–23.58)	**22.82** (21.98–23.67)	23.72 (22.23–25.2)	23.9 (22.98–24.82)

The 95% CI are reported in parentheses. The least average errors are shown in boldface for each activation function. The overall minimum value is italic

Table 10 Average test error values per each activation function for the IMDB data set

Activation function	No. of hidden blocks						
	4	8	16	32	64	128	256
Aranda	15.2 (10.78–19.61)	**13.6** (13.1–14.09)	14 (13–14.99)	**13.6** (12.28–14.91)	13.66 (12.41–14.91)	14.06 (13.77–14.35)	13.93 (11–92–15.94)
Bi-sig1	15 (14.5–15.49)	14.46 (13.03–15.9)	14.2 (13.33–15.06)	**13.33** (12.57–14.09)	14.4 (12.12–16.67)	14 (13.5–14.49)	13.93 (12.68–15.18)
Bi-sig2	14.66 (14.09–15.24)	13.8 (11.21–16.38)	14.2 (12.88–15.51)	13.66 (12.14–15.18)	13.73 (12.97–14.49)	**12.86** (11.61–14.11)	13.93 (11.48–16.38)
Bi-tanh1	16.53 (11.75–21.3)	16.13 (13.68–18.58)	15.33 (13.73–16.93)	14.53 (12.93–16.13)	14 (12.68–15.31)	**13.93** (12.18–15.67)	15.2 (11.61–18.78)
Bi-tanh2	15.8 (12.81–18.78)	13.66 (11.78–15.54)	13.6 (11.61–15.58)	13.2 (12.7–13.69)	**12.93** (11.89–13.96)	14.06 (12.81–15.31)	13.8 (12.48–15.11)
Cloglog	15.13 (12.83–17.42)	13.93 (12.05–15.81)	14.06 (11.82–16.3)	13.8 (12–15.59)	13.46 (12.7–14.22)	**13.26** (12.11–14.41)	13.4 (11.67–15.12)
Cloglogm	13.66 (12.06–15.26)	14.8 (10.49–19.1)	13.33 (12.57–14.09)	13.53 (12.77–14.29)	**12.86** (10.85–14.87)	13.13 (12.37–13.89)	13 (10.72–15.27)
Elliott	14.4 (11.91–16.88)	**13.66** (11.59–15.73)	14.06 (12.91–15.21)	14.13 (12.53–15.73)	14.2 (12.03–16.36)	14.53 (13.95–15.1)	13.73 (10.42–17.04)
Gaussian	17.86 (3.77–31.95)	14.8 (10.38–19.21)	15.8 (10.26–21.33)	**14.66** (14.37–14.95)	16 (12.24–19.75)	24.8 (17.58–32.01)	20.93 (11.19–30.67)
Logarithmic	14.93 (13.05–16.81)	14.86 (13.26–16.46)	14.06 (11.19–16.93)	13.6 (12.6–14.59)	13.6*	**13** (12.13–13.86)	13.33 (10.98–15.68)
Loglog	20 (16.52–23.47)	19.6 (7.76–31.43)	18.86 (10.14–27.59)	**16.8** (9.48–24.11)	17.13 (9.89–24.37)	30.2 (28.88–31.51)	35.93 (32.13–39.72)
Logsigm	17.4 (15.23–19.56)	17.06 (11.41–22.71)	15.53 (11.67–19.39)	14.93 (12.92–16.94)	**14.53** (13.38–15.68)	17.66 (10.7–24.62)	25.93 (1.93–49.92)
Log-sigmoid	14.8 (13.93–15.66)	14.13 (13.37–14.89)	14.2 (12.88–15.51)	14.06 (11.61–16.5)	13.73 (12.97–14.49)	**13.6** (12.1–15.09)	**13.6** (12.73–14.46)
Modified Elliott	17.33 (10.62–24.04)	14.6 (12.43–16.76)	13.26 (9.74–16.79)	14.4 (12.23–16.56)	13.46 (11.22–15.7)	12.93 (10.02–15.84)	***12.46*** (10.22–14.7)
Rootsig	15.6 (11.26–19.93)	14.06 (11.82–16.3)	**13.26** (11.66–14.86)	13.53 (12.77–14.29)	13.86 (13.57–14.15)	13.53 (11.46–15.6)	13.4 (12.08–14.71)
Saturated	16.6 (12.71–20.48)	14.13 (12.88–15.38)	15 (12.51–17.48)	14.73 (9.42–20.04)	13.13 (9.97–16.28)	13.06 (10.71–15.41)	**12.86** (11.83–13.9)
Sech	17.73 (14.99–20.46)	17.26 (10.87–23.65)	**14.4** (11.81–16.98)	14.6 (14.1–15.09)	15.33 (13.32–17.34)	18.8 (13.56–24.03)	32.2 (27.78–36.61)
Sigmoidalm	13.93 (11.69–16.17)	14.33 (13.57–15.09)	14.13 (11.68–16.58)	13.8 (12.07–15.52)	13.73 (13.15–14.3)	**13.6** (12.1–15.09)	14.2 (13.33–15.06)
Sigmoidalm2	14.06 (12.91–15.21)	14.33 (13.18–15.48)	14 (12.01–15.98)	**13.8** (12.93–14.66)	14.13 (13.37–14.89)	13.86 (13.1–14.62)	14.6 (13.6—15.59)
Sigt	17.06 (13.33–20.79)	14.26 (13.69–14.84)	15.2 (12.71–17.68)	14.46 (11.43–17.5)	**14.06** (11.61–16.51)	14.66 (13.06–16.26)	15.53 (12.7–18.35)
Skewed-sig	15.8 (10.07–21.52)	17.93 (14.62–21.24)	17 (12.75–21.24)	16.06 (12.87–19.26)	**14.93** (13.41–16.45)	16.53 (13.66–19.4)	29.93 (1.82–58.03)
Softsign	14.06 (11.99–16.13)	14.46 (13.21–15.71)	14.06 (13.03–15.1)	**12.93** (12.35–13.5)	13.86 (11.36–16.36)	13.13 (12.09–14.16)	13 (10.72–15.27)
Wave	19.46 (12.22–26.7)	18.33 (7.62–29.03)	18.06 (15.99–20.13)	**16.2** (13.92–18.47)	19.4 (16.29–22.5)	38.33 (27.65–49.01)	34.93 (4.58–65.27)

The 95% CI are reported in parentheses. The least average errors are shown in boldface for each activation function. The overall minimum value is italic

* All results for logarithmic function with 64 units was the same

Table 11 Average error values per each activation function for the MNIST data set

Activation function	No. of hidden blocks	
	64	128
Aranda	2.13 (1.84–2.42)	**2.10** (1.85–2.34)
Bi-sig1	2.6 (2.35–2.84)	**2.00** (1.75–2.24)
Bi-sig2	2.36 (1.98–2.74)	**2.03** (1.65–2.41)
Bi-tanh1	3.03 (2.74–3.32)	**2.50** (2.25–2.74)
Bi-tanh2	2.13 (1.5–2.75)	**1.93** (1.55–2.31)
Cloglog	2.30 (2.05–2.54)	**2.26** (1.88–2.64)
Cloglogm	2.70 (2.26–3.13)	**2.13** (1.75–2.51)
Elliott	2.16 (1.29–3.03)	***1.66*** (1.28–2.04)
Gaussian	2.16 (1.64–2.68)	**1.96** (1.67–2.25)
Logarithmic	2.80 (2.14–3.45)	**2.70** (2.2–3.19)
Loglog	5.46 (4.94–5.98)	**8.16** (3.79–12.54)
Logsigm	3.90 (3–4.79)	**4.53** (3.59–5.47)
Log-sigmoid	2.30 (1.8–2.79)	**2.16** (1.59–2.74)
Modified Elliott	2.43 (1.91–2.95)	**2.03** (1.74–2.32)
Rootsig	2.13 (1.98–2.27)	**1.90** (1.65–2.14)
Saturated	**3.43** (3.14–3.72)	3.83 (1.82–5.84)
Sech	2.33 (1.95–2.71)	**2.06** (1.68–2.44)
Sigmoidalm	**2.23** (1.47–2.99)	2.36 (1.74–2.99)
Sigmoidalm2	2.83 (2.31–3.35)	**2.66** (2.14–3.18)
Sigt	**2.66** (2.52–2.81)	2.76 (2.47–3.05)
Skewed-sig	5.43 (3.07–7.79)	**4.73** (4.35–5.11)
Softsign	1.83 (1.68–1.97)	***1.66*** (1.52–1.81)
Wave	2.56 (2.04–3.08)	**2.26** (1.88–2.64)

The 95% CI are reported in parentheses. The least average errors are shown in boldface for each activation function. The overall minimum value is italic

References

1. Hochreiter S, Schmidhuber J (1997) Long short-term memory. Neural Comput 9:1735–1780. doi:10.1162/neco.1997.9.8.1735
2. Graves A (2012) Supervised sequence labelling with recurrent neural networks. Springer, Berlin
3. Graves A, Schmidhuber J (2005) Framewise phoneme classification with bidirectional lstm and other neural network architectures. Neural Netw 18(5):602–610
4. Liwicki M, Graves A, Bunke H, Schmidhuber J (2007) A novel approach to on-line handwriting recognition based on bidirectional long short-term memory networks. In: Proceedings of the 9th international conference on document analysis and recognition, ICDAR 2007
5. Graves A, Liwicki M, Fernández S et al (2009) A novel connectionist system for unconstrained handwriting recognition. IEEE Trans Pattern Anal Mach Intell 31:855–868. doi:10.1109/TPAMI.2008.137
6. Graves A, Fernández S, Schmidhuber J (2005) Bidirectional LSTM Networks for Improved Phoneme Classification and Recognition. In: Duch W, Kacprzyk J, Oja E, Zadrożny S (eds) Artificial neural networks: formal models and their applications—ICANN 2005. Springer, Berlin, pp 799–804
7. Otte S, Krechel D, Liwicki M, Dengel A (2012) Local feature based online mode detection with recurrent neural networks. In: 2012 international conference on frontiers in handwriting recognition (ICFHR). pp 533–537
8. Graves A, Mohamed A, Hinton G (2013) Speech recognition with deep recurrent neural networks. arXiv:1303.5778 [cs]
9. Thang Luong IS (2014) Addressing the rare word problem in neural machine translation. doi:10.3115/v1/P15-1002
10. Wöllmer M, Metallinou A, Eyben F, et al (2010) Context-sensitive multimodal emotion recognition from speech and facial expression using bidirectional LSTM modeling. In: Proceedings of interspeech, Makuhari. pp 2362–2365
11. Sak H, Senior A, Beaufays F (2014) Long short-term memory recurrent neural network architectures for large scale acoustic modeling. In Proceedings of the annual conference of international speech communication association (INTERSPEECH). pp 338–342
12. Fan Y, Qian Y, Xie F, Soong FK (2014) TTS synthesis with bidirectional LSTM based recurrent neural networks. In: Proceedings interspeech. pp 1964–1968
13. Zaremba W, Sutskever I, Vinyals O (2014) Recurrent neural network regularization. arXiv:1409.2329 [cs]
14. Sønderby SK, Winther O (2014) Protein secondary structure prediction with long short term memory networks. arXiv:1412.7828 [cs, q-bio]
15. Marchi E, Ferroni G, Eyben F, et al (2014) Multi-resolution linear prediction based features for audio onset detection with bidirectional LSTM neural networks. In: 2014 IEEE international conference on acoustics, speech and signal processing (ICASSP). pp 2164–2168
16. Donahue J, Hendricks LA, Guadarrama S, et al (2014) Long-term recurrent convolutional networks for visual recognition and description. arXiv:1411.4389 [cs]
17. Wollmer M, Blaschke C, Schindl T et al (2011) Online driver distraction detection using long short-term memory. IEEE Trans Intell Transp Syst 12:574–582. doi:10.1109/TITS.2011.2119483
18. da Gomes GSS, Ludermir TB (2013) Optimization of the weights and asymmetric activation function family of neural network for time series forecasting. Exp Syst Appl 40:6438–6446. doi:10.1016/j.eswa.2013.05.053
19. Duch W, Jankowski N (1999) Survey of neural transfer functions. Neural Comput Surv 2:163–213
20. Singh Sodhi S, Chandra P (2003) A class +1 sigmoidal activation functions for FFANNs. J Econ Dyna Control 28:183–187
21. da Gomes GSS, Ludermir TB, Lima LMMR (2010) Comparison of new activation functions in neural network for forecasting financial time series. Neural Comput Appl 20:417–439. doi:10.1007/s00521-010-0407-3
22. Gomes GS d S, Ludermir TB (2008) Complementary log-log and probit: activation functions implemented in artificial neural networks. In: Eighth international conference on hybrid intelligent systems, 2008. HIS'08. pp 939–942
23. Michal Rosen-Zvi MB (1998) Learnability of periodic activation functions: general results. Phys Rev E 58:3606–3609. doi:10.1103/PhysRevE.58.3606
24. Leung H, Haykin S (1993) Rational function neural network. Neural Comput 5:928–938. doi:10.1162/neco.1993.5.6.928
25. Ma L, Khorasani K (2005) Constructive feedforward neural networks using Hermite polynomial activation functions. IEEE Trans Neural Netw 16:821–833. doi:10.1109/TNN.2005.851786
26. Hornik K (1993) Some new results on neural network approximation. Neural Netw 6:1069–1072. doi:10.1016/S0893-6080(09)80018-X
27. Hornik K (1991) Approximation capabilities of multilayer feedforward networks. Neural Netw 4:251–257. doi:10.1016/0893-6080(91)90009-T

28. Hartman E, Keeler JD, Kowalski JM (1990) Layered neural networks with Gaussian hidden units as universal approximations. Neural Comput 2:210–215. doi:10.1162/neco.1990.2.2.210
29. Skoundrianos EN, Tzafestas SG (2004) Modelling and FDI of dynamic discrete time systems using a MLP with a new sigmoidal activation function. J Intell Robot Syst 41:19–36. doi:10.1023/B:JINT.0000049175.78893.2f
30. Pao Y-H (1989) Adaptive pattern recognition and neural networks. Addison-Wesley Longman Publishing Co. Inc, Boston
31. Carroll SM, Dickinson BW (1989) Construction of neural nets using the radon transform. In: International joint conference on neural networks, 1989, vol 1. IJCNN. pp 607–611
32. Cybenko G (1989) Approximation by superpositions of a sigmoidal function. Math Control Signal Syst 2:303–314. doi:10.1007/BF02551274
33. Chandra P, Singh Y (2004) Feedforward sigmoidal networks—equicontinuity and fault-tolerance properties. IEEE Trans Neural Netw 15:1350–1366. doi:10.1109/TNN.2004.831198
34. Williams RJ, Zipser D (1995) Gradient-based learning algorithms for recurrent networks and their computational complexity. In: Chauvin Y, Rumelhart DE (eds) Back-propagation: theory, architectures and applications. L. Erlbaum Associates Inc., Hillsdale, pp 433–486
35. Zeiler MD (2012) ADADELTA: an adaptive learning rate method. arXiv:1212.5701 [cs]
36. Singh Sodhi S, Chandra P (2014) Bi-modal derivative activation function for sigmoidal feedforward networks. Neurocomputing 143:182–196. doi:10.1016/j.neucom.2014.06.007
37. Yuan M, Hu H, Jiang Y, Hang S (2013) A new camera calibration based on neural network with tunable activation function in intelligent space. In: 2013 6th international symposium on computational intelligence and design (ISCID). pp 371–374
38. Chandra P, Sodhi SS (2014) A skewed derivative activation function for SFFANNs. In: Recent advances and innovations in engineering (ICRAIE). IEEE, pp 1–6
39. Elliott DL (1993) A better activation function for artificial neural networks. Technical Report ISR TR 93–8, University of Maryland
40. Hara K, Nakayamma K (1994) Comparison of activation functions in multilayer neural network for pattern classification. In: 1994 IEEE international conference on neural networks, 1994. IEEE world congress on computational intelligence, vol 5. pp 2997–3002
41. Burhani H, Feng W, Hu G (2015) Denoising autoencoder in neural networks with modified Elliott activation function and sparsity-favoring cost function. In: 2015 3rd international conference on applied computing and information technology/2nd international conference on computational science and intelligence (ACIT-CSI). pp 343–348
42. Chandra P, Singh Y (2004) A case for the self-adaptation of activation functions in FFANNs. Neurocomputing 56:447–454. doi:10.1016/j.neucom.2003.08.005
43. Nair V, Hinton GE (2010) Rectified linear units improve restricted boltzmann machines. In: Proceedings of the 27th international conference on machine learning (ICML 2010). pp 807–814
44. Tieleman T, Hinton G (2012) Lecture 6.5-rmsprop: divide the gradient by a running average of its recent magnitude. COURSERA Neural Netw Mach Learn 4:26–31
45. Duchi J, Hazan E, Singer Y (2011) Adaptive Subgradient Methods for Online Learning and Stochastic Optimization. Journal of Machine Learning Research 12:2121–2159
46. Lipton ZC, Berkowitz J, Elkan C (2015) A critical review of recurrent neural networks for sequence learning. arXiv:1506.00019 [cs]
47. Ruder S (2016) An overview of gradient descent optimization algorithms. arXiv:1609.04747 [cs]
48. Hinton GE, Srivastava N, Krizhevsky A, et al (2012) Improving neural networks by preventing co-adaptation of feature detectors, vol abs/1207.0580. arXiv preprint arXiv:1207.0580. The Computing Research Repository (CoRR)
49. Pang B, Lee L (2005) Seeing stars: exploiting class relationships for sentiment categorization with respect to rating scales. In: Proceedings of the 43rd annual meeting on association for computational linguistics. Association for Computational Linguistics, Stroudsburg, PA, USA. pp 115–124
50. Maas AL, Daly RE, Pham PT, et al (2011) Learning word vectors for sentiment analysis. In: Proceedings of the 49th annual meeting of the association for computational linguistics: human language technologies, vol 1. Association for Computational Linguistics, Stroudsburg, PA, USA. pp 142–150
51. Lenc L, Hercig T (2016) Neural networks for sentiment analysis in Czech. In: ITAT 2016 proceedings, CEUR Workshop Proceedings, vol 1649. pp 48–55
52. Dai AM, Le QV (2015) Semi-supervised sequence learning. In: Cortes C, Lawrence ND, Lee DD et al (eds) Advances in neural information processing systems, vol 28. Curran Associates Inc, Red Hook, pp 3079–3087
53. Arjovsky M, Shah A, Bengio Y (2016) Unitary evolution recurrent neural networks. In: Proceedings of the 33rd international conference on machine learning. pp 1120–1128
54. Bengio Y, Simard P, Frasconi P (1994) Learning long-term dependencies with gradient descent is difficult. IEEE Trans Neural Netw 5:157–166. doi:10.1109/72.279181
55. Pascanu R, Mikolov T, Bengio Y (2013) On the difficulty of training recurrent neural networks. In: Proceedings of the 30th international conference on machine learning. pp 1310–1318
56. Gers FA, Schmidhuber J (2000) Recurrent nets that time and count. In: Proceedings of the IEEE-INNS-ENNS international joint conference on neural networks. IJCNN 2000. Neural computing: new challenges and perspectives for the new millennium, vol 3. pp 189–194
57. Schuster M, Paliwal KK (1997) Bidirectional recurrent neural networks. IEEE Trans Signal Process 45:2673–2681. doi:10.1109/78.650093
58. Cho K, van Merrienboer B, Bahdanau D, Bengio Y (2014) On the properties of neural machine translation: encoder–decoder approaches. arXiv:1409.1259 [cs, stat]