2015 IEEE Symposium Series on Computational Intelligence

Measuring Saturation in Neural Networks

Anna Rakitianskaia
Department of Computer Science
University of Pretoria
Pretoria, South Africa
Email: annar@cs.up.ac.za

Andries Engelbrecht
Department of Computer Science
University of Pretoria
Pretoria, South Africa
Email: engel@cs.up.ac.za

***Abstract*—In the neural network context, the phenomenon of saturation refers to the state in which a neuron predominantly outputs values close to the asymptotic ends of the bounded activation function. Saturation damages both the information capacity and the learning ability of a neural network. The degree of saturation is an important neural network characteristic that can be used to understand the behaviour of the network itself, as well as the learning algorithm employed. This paper suggests a measure of saturation for bounded activation functions. The suggested measure is independent of the activation function range, and allows for direct comparisons between different activation functions.**

I. Introduction

Saturation occurs when the hidden units of a neural network (NN) predominantly output values close to the asymptotic ends of the activation function range. Saturation reduces the NN to a binary state, thus limiting the overall information capacity of the NN. Saturated units make gradient descent learning slow and inefficient due to small derivative values near the asymptotes. Glorot and Bengio [1] observed the activation function outputs in order to better understand the difficulty of training deep neural networks. It was recently shown that non-gradient descent learning can also be rather sensitive to the degree of saturation present in the NN [2]. In general, the degree of saturation is an important NN characteristic that can provide insight into both the model and the learning algorithm behaviour.

No standardised way to measure the degree of saturation exists, though. Glorot and Bengio [1] graphed the activation function output ranges over algorithm iterations to observe the level of saturation. In both [1] and [2] frequency distributions were used to compare the spread of activation function outputs across a selection of different activation functions. Even though graphical representations are easy to interpret, they provide no means of a statistical comparison between the degree of saturation as exhibited by the methods being compared.

This paper suggests a simple measure of saturation for bounded activation functions. The suggested measure generates a single value in a predefined range independent of the activation function range, allowing for statistical comparisons between different activation functions. The rest of the paper is structured as follows: Section II presents the problem of saturation. Section III presents the suggested numeric saturation measure. Section IV tests the suggested measure on a selection of problems. Section V summarises the paper and outlines the suggested measures necessary in a saturation behaviour study.

II. Saturation in Neural Networks

Activation functions used in NN hidden and output layers are usually chosen to be non-linear and bounded. Non-linearity of the hidden units allows a NN to approximate any non-linear mapping between inputs and outputs provided that enough neurons are used in the hidden layer [3]. Upper and lower bounds ensure that the signal does not grow uncontrollably as it propagates from one layer to the next. Functions with a sigmoidal curve such as the logistic function and the hyperbolic tangent are often used as activation functions. Sigmoidal functions exhibit linear behaviour in the active range determined by the function slope, and saturate (approach asymptotes) for large positive and negative input values.

Thus, if the magnitude of the input signal lies outside the active range of a sigmoidal activation function, the output signal will be close to an asymptotic value. The net input signal is a weighted sum of inputs from the previous NN layer:

$$net = \sum_{i=1}^{I+1} w_i z_i \tag{1}$$

where $I+1$ is the number of incoming connections plus the bias, w_i is the weight of the i-th connection, and z_i is the i-th input signal. Now, suppose that w_i for a certain i is set to a very large value, causing net to always lie outside of the activation function's active range. If a bounded sigmoidal activation function is used, the output for such net will be very close to either the lower or the upper asymptotic value, depending on the net's sign. In such case, *saturation* is observed: the phenomenon when a NN unit is reduced to a binary state, predominantly outputting values close to the asymptotic ends of the activation function.

Why is saturation undesirable? For a saturated unit, a small change in the incoming weights will have almost no influence on the output of the unit. Therefore, a training algorithm used for weight optimisation will struggle to determine whether the weight change had a positive or a negative effect on the NN's performance. As a result, the training algorithm will stagnate, and no further learning will occur.

If a sigmoidal function is used in the output NN units, and a classification problem with binary-coded targets is considered, then "binary" saturated outputs may seem appropriate. However, a saturated output unit does not indicate the "confidence" level of the NN [4], in other words, all patterns, even the ones not fitted very well by the NN, will be classified with the same "strength", preventing the training algorithm from refining the solution.

978-1-4799-7560-0/15 $31.00 © 2015 IEEE
DOI 10.1109/SSCI.2015.202

Even though excessive saturation is undesirable throughout the NN, a certain degree of saturation is in fact necessary in the output layer to accommodate the binary-coded classification targets. Overly linear hidden units will not compute a non-linear mapping, therefore some saturation is required in the hidden layers, too. Thus, we face a difficult problem of finding a balance between too little saturation (trivial model) and too much saturation (imprecise model immune to further training) [1]. It was also shown in [5] that there is a correlation between excessive saturation and overfitting, thus controlling saturation may be beneficial to the generalisation capabilities of the constructed model. In order to control saturation, a way of quantifying, or measuring saturation is necessary.

III. Measuring Saturation

An obvious way to check for the presence of saturation is to examine the activation function outputs in the non-linear layers of a NN. If the activation outputs on the given data set are concentrated around the asymptotic ends, then saturation is present. Raw activation output data can be analysed graphically to approximate the extent of saturation. Glorot and Bengio [1] graphed the activation function output ranges over algorithm iterations to observe the level of saturation. In both [1] and [2] frequency distributions were used to compare the spread of activation function outputs across a selection of different activation functions. However, graphical analysis provides no means of a statistical comparison between different models. A single-valued saturation measure would be much more convenient and meaningful.

In [5], a simple single-valued measure of saturation based on the magnitudes of net was proposed:

$$\varsigma_h = \frac{\sum_{i=1}^{P}\sum_{j=1}^{H}|net_{ij}|}{PH} \tag{2}$$

where h is the hidden layer index, P is the number of data patterns, and H is the number of units in the hidden layer. Saturation measure ς_h effectively measures the growth of the input signal magnitudes. As discussed in Section II, saturation occurs when the value of net lies outside the active range of the activation function. Thus, monitoring the growth of average net gives an indication of the extent of saturation present in the given hidden layer. The disadvantage of ς_h is that the same net value will yield different levels of saturation for different activation functions. Thus, ς_h can only be used to compare saturation level in layers that employ the same activation function.

Another disadvantage of using net magnitudes is that the resulting measure of saturation is unbounded, which is somewhat counterituitive given that the concept of saturation applies to bounded activation functions only. A single-valued saturation measure based on activation function outputs rather than inputs may be easier to interpret.

Consider the outputs of an arbitrary bounded activation function $g(net)$ for all values of net. If a frequency distribution of $g(net)$ for all values of net is constructed, the level of saturation can be approximated by observing the output frequencies. Figure 1 shows a typical histogram produced by a saturated unit over all patterns in the data set: highest frequencies are concentrated around the extremes of g's range

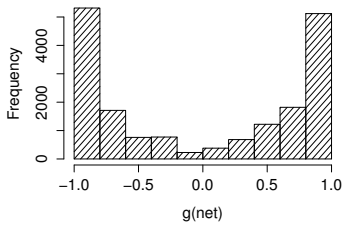


Fig. 1. Frequency histogram of a saturated NN unit

(in this case, $[-1, 1]$). The higher the frequency in the leftmost and the rightmost bin, the higher the saturation. A completely saturated unit would have a frequency of zero in all bins except the leftmost and the rightmost one. A non-saturated unit will have frequencies of similar magnitude across all the bins.

A single-valued saturation measure can be derived from a $g(net)$ frequency distribution. The average output signal value for each bin b can be calculated as follows:

$$\bar{g}_b = \begin{cases} \left(\sum_{k=1}^{f_b} g(net)_k\right)/f_b & \text{if } f_b > 0 \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

where f_b is the number of output signals $g(net)$ in bin b. If the range of g is centred around zero, the absolute average $|\bar{g}_b|$ will be higher for the bins closer to the asymptotic values, and lower for bins closer to the centre. If the range of g is $[g_L, g_U]$, $\bar{g}_b$ can be scaled to the $[-1, 1]$ range as follows:

$$\bar{g}'_b = \frac{2(\bar{g}_b - g_L)}{g_U - g_L} - 1 \tag{4}$$

A weighted mean magnitude is then calculated as

$$\varphi_B = \frac{\sum_{b=1}^{B}|\bar{g}'_b| f_b}{\sum_{b=1}^{B} f_b} \tag{5}$$

where B is the total number of bins, and f_b constitutes the weight of each bin. The weighted mean is the same as the arithmetic mean if all weights are equal. Thus, if $\bar{g}'$ is uniformly distributed in $[-1, 1]$, the value of φ_B will be close to 0.5, since absolute activation values are considered, thus all $\bar{g}'$ values are squashed to the $[0, 1]$ interval. For a normal distribution of $\bar{g}'$, the value of φ_B will be smaller than 0.5. The higher the assymptotic frequencies of $\bar{g}'$, the closer φ_B will be to 1. Thus, φ_B can be used as a measure of saturation that tends to 1 as the degree of saturation increases, and tends to zero otherwise. The relationship between the φ_B values and the different $g(net)$ histogram shapes are illustrated in Figure 2.

The rest of the paper presents the empirical study where φ_B is shown to be a valid NN saturation measure.

IV. Empirical Study

The purpose of the experiments was to test the saturation measure φ_B proposed in Section III. In order to test the robustness and universality of the measure, a number of benchmarks, activation functions, and training algorithm set-ups were considered. The proposed measure φ_B was compared with the previously used saturation measure ς_h. The rest of this

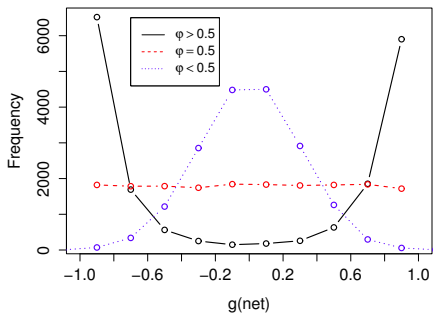


Fig. 2. Values of φ for different histogram shapes

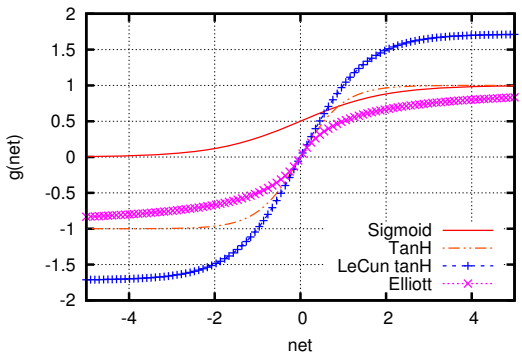


Fig. 3. Activation function slopes

section is structured as follows: Section IV-A summarises the benchmarks used. Section IV-B discusses the activation functions considered. Section IV-C presents the training algorithm used. Empirical results are discussed in Section IV-D.

A. Benchmarks

For the purpose of this study, four well-known benchmark classification problems were considered. Problems with a known optimal number of hidden units were chosen to simplify the parameter optimisation process. The benchmark problems along with the corresponding architectures are summarised in Table I. The specified sources point to papers from which the NN architectures were adopted.

TABLE I. BENCHMARK PROBLEMS

Problem	# Input	# Hidden	Source
Iris	4	4	Gupta and Lam [6]
Glass Identification	9	9	Gupta and Lam [6]
Heart	35	6	Carvalho and Ludermir [7]
Diabetes	8	6	Carvalho and Ludermir [7]

The data sets were pre-processed according to the suggestions given in [4]: The inputs were standardised such that the average of each input variable over the data set was close to zero. The targets were scaled to the appropriate activation function output ranges.

B. Activation Functions

Feedforward NNs with a single hidden layer were used in the experiments. The identity (linear) activation function was used in the input layer, while the hidden layer and the output layer both used a non-linear activation function $g(net)$. The proposed saturation measure φ_B is independent of the activation function parameters, and allows to directly compare saturation levels exhibited by different activation functions. To illustrate this quality, the following activation functions, shown in Fig.3, were considered:

1) Sigmoid: The sigmoid function is defined as

$$g(net) = \frac{1}{1 + e^{-net}} \tag{6}$$

The sigmoid function is the most commonly used activation function. The output of the sigmoid function is in the range $(0, 1)$.

2) TanH: The hyperbolic tangent function, further referred to as tanH, is defined as

$$g(net) = \frac{e^{net} - e^{-net}}{e^{net} + e^{-net}} \tag{7}$$

The output of tanH is in the range $(-1, 1)$.

3) LeCun tanH: A modified hyperbolic tangent activation function was suggested by LeCun et al in [4], and is defined by

$$g(net) = 1.7159 \tanh\left(\frac{2}{3} net\right) \tag{8}$$

Compared to the traditional sigmoid, LeCun tanH (shown in Fig. 3) has a softer slope and a wider output range, $(-1.7159, 1.7159)$.

4) Elliott: The last activation function used in this paper is the Elliott function [8], further referred to as Elliott and defined as

$$g(net) = \frac{net}{1 + |net|} \tag{9}$$

The output range of Elliott is also $(-1, 1)$, but it has a shallower gradient than TanH, and thus approaches the asymptotes slower.

C. Training Algorithm: Particle Swarm Optimisation

For the purpose of this study, particle swarm optimisation (PSO) was used to train the NNs. PSO, first introduced by Kennedy and Eberhart in [9], is a nature-inspired population-based optimisation technique. PSO works on a set of particles, referred to as a swarm, where every particle represents a candidate solution to the optimisation problem. For an n-dimensional optimisation problem, a particle is represented by an n-dimensional vector, $\vec{x}$, also referred to as the particle's position. Every particle has a fitness value, indicating the quality of the candidate solution, and a velocity vector, $\vec{v}$, which determines the step size and direction of the particle's movement. Social interaction is imitated by forming neighbourhoods within a swarm. Each particle remembers its own best position found so far, and can query the neighbourhood for the best position as discovered by the neighbour particles. PSO searches for an optimum by moving the particles through the search space. At each time step, t, the position $\vec{x}_i(t)$ of

particle i is modified by adding the particle velocity $\vec{v}_i(t)$ to the previous position vector:

$$\vec{x}_i(t) = \vec{x}_i(t-1) + \vec{v}_i(t) \tag{10}$$

Particle velocity determines the step size and direction of the particle. Velocity is updated using

$$\begin{aligned}\vec{v}_i(t) = \omega\vec{v}_i(t-1) + c_1\vec{r}_1(t)(\vec{x}_{pbest,i}(t-1) - \vec{x}_i(t-1)) \\ + c_2\vec{r}_2(t)(\vec{x}_{nbest,i}(t-1) - \vec{x}_i(t-1))\end{aligned} \tag{11}$$

where ω is the inertia weight [10], controlling the influence of previous velocity values on the new velocity; c_1 and c_2 are acceleration coefficients used to scale the influence of the *cognitive* (second term of Equation (11)) and *social* (third term of Equation (11)) components; $\vec{r}_1(t)$ and $\vec{r}_2(t)$ are vectors with each component sampled from a uniform distribution $U(0,1)$; $\vec{x}_{pbest,i}(t)$ is the personal best of particle i, in other words, the best position encountered by this particle so far; similarly, $\vec{x}_{nbest,i}(t)$ is the neighbourhood best of particle i, or the best position found by any of the particles in the neighbourhood of particle i. Thus, each particle is attracted to both the best position encountered by itself so far, as well as the overall best position found by the neighbourhood.

A particle's neighbourhood is determined topologically rather than spatially, meaning that the distance between particles is determined by the particle indices and not the actual position in the search space [9]. Two neighbourhood topologies were considered in this study: the gBest topology [9], and the Von Neumann (VN) topology [11]. In the gBest topology, the entire swarm constitutes the neighbourhood of a particle. In the VN topology, particles are connected in a grid-like structure, where every particle is directly connected to four neighbours: above, below, to the left and to the right. For all experiments, a swarm of 20 particles was used. PSO with the gBest topology is further referred to as gBest PSO, and PSO employing the VN topology is further referred to as VN PSO.

In all experiments, the inertia weight ω was set to 0.729844 while the values of the acceleration coefficients c_1 and c_2 were set to 1.496180. This choice is based on [12], where it was shown that such parameter settings give convergent behaviour.

A maximum velocity $\vec{V}_{max}$ [10] is sometimes used to limit (or clamp) particle velocity in every dimension. Velocity clamping is done to prevent particles from traversing the search space too fast, since unreasonably large steps prevent particles from exploiting good regions. It was observed in [13] that PSO tends to diverge on NN training problems unless swarm expansion is restricted. With velocity clamping, $\vec{V}_{max}$ is enforced by restricting $\vec{v}_i(t)$ per dimension j as follows:

$$v_{ij}(t) = \begin{cases} V_{max,j} & \text{if } v_{ij}(t) > V_{max,j} \\ -V_{max,j} & \text{if } v_{ij}(t) < -V_{max,j} \\ v_{ij}(t) & \text{otherwise} \end{cases} \tag{12}$$

For the purpose of this study, $\vec{V}_{max}$ was set to $\vec{1}$ for all experiments.

All reported results are averages over 30 simulations. Every simulation ran for 1000 iterations. Every data set was divided into a training set and a test set; 80% of data patterns constituted the training set, and the remaining 20% were used for testing. Test data used to calculate the final generalisation error values was not used for parameter optimisation.

D. Experimental Results

Table II summarises the average mean classification error (E_C), ς_h, and φ_B for different number of bins, B, obtained for the Iris data set. The largest values are shown in bold for each row, and the lowest values are shown in italics. For the sake of brevity, gBest PSO and VN PSO are referred to as gBest and VN, respectively.

According to the proposed saturation measure, φ_B, most saturation was observed in NNs that used the sigmoid activation function in the hidden layer, trained with VN PSO. This observation can be confirmed by looking at the frequency distribution of the hidden layer outputs obtained at the last iteration of the algorithm, shown in Figure 4(a). Indeed, most output signals fall into either the leftmost or the rightmost bin, indicating high saturation.

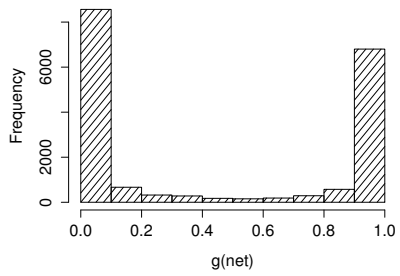
According to ς_h, most saturation was observed in NNs that used the Elliott $g(net)$ trained with gBest PSO. This disagrees with the φ_B result. To resolve the matter, consider Figure 4(b) that shows the frequency distribution of the Elliott $g(net)$ values obtained using gBest PSO as the training algorithm. Even though the highest frequencies are concentrated in the leftmost and the rightmost bins, the remaining bins in Figure 4(b) have higher frequencies than the corresponding bins in Figure 4(a). Thus, Elliott was less saturated than the sigmoid, as correctly indicated by φ_B. Conclusions based on ς_h will therefore be erroneous, because ς_h is based on net alone, and the same values of net yield different $g(net)$ values for the different activation functions.

According to both ς_h and φ_B shown in Table II, the least saturation was observed in NNs with the LeCun tanH activation function in the hidden layer. Figure 5 shows the corresponding frequency distributions for gBest PSO and VN PSO. Mid-range bin frequencies are higher for LeCun tanH than for the other activation functions as depicted in Figure 4. Evidently, LeCun tanH exhibited less saturation.

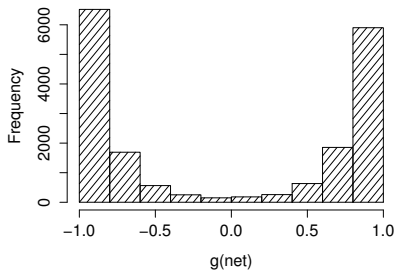
The two saturation measures agree on the least saturated activation function, but disagree on the algorithm yielding the least saturation: according to ς_h, LeCun tanH $g(net)$ trained with VN PSO saturated the least. According to φ_B, gBest

TABLE II. AVERAGE E_C, ς_h, AND φ VALUES FOR THE IRIS DATA SET, WITH CORRESPONDING STANDARD DEVIATION IN PARENTHESIS

g(net)	**Sigmoid**		**TanH**		**LeCun TanH**		**Elliott**	
Alg.:	gBest	VN	gBest	VN	gBest	VN	gBest	VN
E_C	0.0422	0.0322	0.0411	*0.0289*	**0.0467**	0.0367	0.0444	0.04
	(0.0315)	(0.0321)	(0.0358)	(0.0324)	(0.0416)	(0.0354)	(0.0343)	(0.0355)
ς_h	9.8375	8.2117	5.5728	5.1136	4.0721	*3.6347*	**11.4278**	9.4702
	(4.1496)	(2.3794)	(1.4583)	(1.5274)	(1.797)	(1.2032)	(4.3423)	(1.9462)
φ_5	0.88	**0.9065**	0.8945	0.9016	*0.7367*	0.7631	0.8158	0.8138
	(0.0887)	(0.0472)	(0.0595)	(0.049)	(0.1004)	(0.0951)	(0.0501)	(0.0337)
φ_{10}	0.8825	**0.9078**	0.8964	0.9039	*0.7414*	0.768	0.8174	0.8151
	(0.0864)	(0.0459)	(0.0585)	(0.0464)	(0.0968)	(0.0915)	(0.0491)	(0.0332)
φ_{20}	0.8825	**0.9078**	0.8964	0.9039	*0.7414*	0.768	0.8174	0.8151
	(0.0864)	(0.0459)	(0.0585)	(0.0464)	(0.0968)	(0.0915)	(0.0491)	(0.0332)
φ_{30}	0.8825	**0.9078**	0.8964	0.9039	*0.7421*	0.7681	0.8174	0.8151
	(0.0864)	(0.0459)	(0.0585)	(0.0464)	(0.0971)	(0.0916)	(0.0491)	(0.0332)
φ_{50}	0.8825	**0.9078**	0.8964	0.9039	*0.7421*	0.7681	0.8174	0.8151
	(0.0864)	(0.0459)	(0.0585)	(0.0464)	(0.0971)	(0.0916)	(0.0491)	(0.0332)



(a) Sigmoid, VN PSO



(b) Elliott, GBest PSO

Fig. 4. Hidden unit output values frequency distributions for the Iris data set

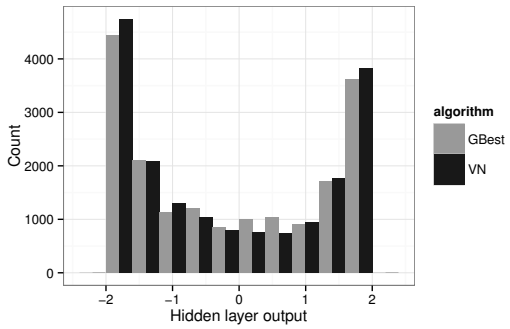


Fig. 5. Iris, LeCun TanH, last iteration frequency histograms

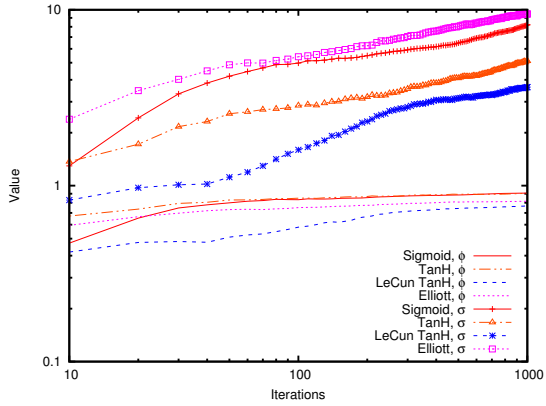


Fig. 6. φ_{10} and ς_h profiles for the Iris data set

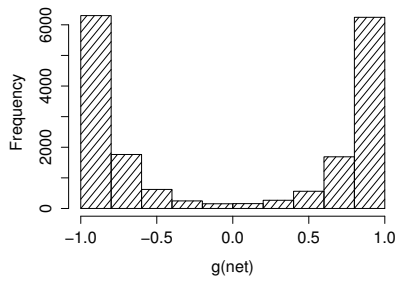
PSO with the same $g(net)$ achieved the lowest saturation. The matter can be resolved by putting the frequency distributions next to one another, as shown in Figure 5. Both histograms are similar in shape, but VN has a noticeably higher frequency in the leftmost and the rightmost bins, while gBest has higher frequencies in the middle bins. Thus, it is not unreasonable to conclude that VN saturated more than gBest. The proposed measure φ_B once again describes the data more correctly than ς_h in this example.

Figure 6 provides another graphical comparison between φ_B and ς_h. In Figure 6, profiles of φ_B for $B = 10$ (further referred to as φ_{10}) and ς_h over iterations 1 to 1000 are shown for the VN PSO algorithm. According to ς_h, Elliott exhibited the most saturation out of the four activation functions considered. According to φ_{10}, the most saturation was exhibited by sigmoid and tanH, and Elliott in fact saturated less than both these functions. Figures 7(a) and 7(b) show the frequency distributions for Elliott and tanH obtained under VN PSO training. Clearly, tanH exhibited more saturation than Elliott, as most output signals fall into either the leftmost or the rightmost bin on the corresponding histogram. Thus, once again φ_B provided a more precise description of the saturation level than ς_h. The superiority of φ_B in all of the above examples is due to scaling outputs of any $g(net)$ to the same range $[-1, 1]$, as well as using the $g(net)$ output values to determine the saturation instead of the input net values. Figure 6 illustrates that both φ_B and ς_h show similar dynamics in saturation growth, but φ_B makes the results comparable by scaling them to the same range.

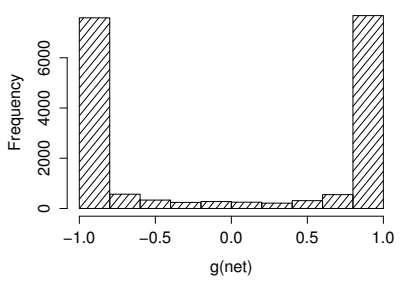
Table II lists values of φ_B obtained for $B \in [5, 10, 20, 30, 50]$. The first two significant digits of the φ_B values are identical for most values of B for all algorithms considered, indicating that φ_B is not very sensitive to the value of B. Using five bins seems to give a somewhat rough estimate of saturation, but ten bins or more converge on the same value. Thus, sufficient granularity was obtained with $B = 10$ on the Iris data set.

Table III summarises the average E_C, ς_h, and φ_B for different number of bins, B, obtained for the Glass data set. The largest values are shown in bold for each row, and the lowest values are shown in italics. According to ς_h, highest saturation on the Glass data set was observed with the Elliott $g(net)$ using the gBest PSO as the training algorithm. According to φ_B, tanH using the gBest PSO for training saturated the most. Figures 8(a) and 8(b) show frequency distributions for the corresponding hidden layer outputs. Figure 8(b) clearly shows higher frequencies in the leftmost and the rightmost bins, and lower frequencies in the mid-range bins than Figure 8(a). Thus, φ_B correctly indicated tanH as more saturated than Elliott, and ς_h provided misleading results.

Both ς_h and φ_B agree that LeCun tanH using the VN PSO for training saturated the least. The values of φ_B on

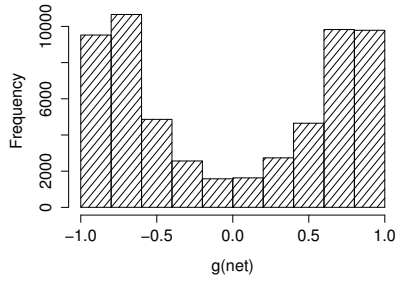


(a) Elliott, VN PSO

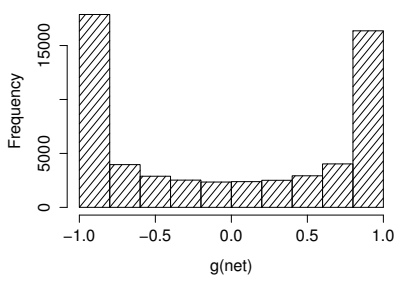


(b) TanH, VN PSO

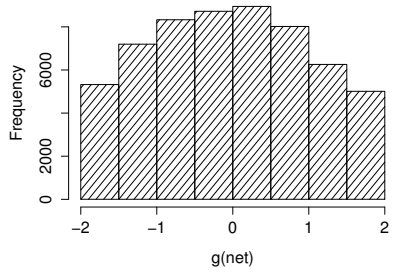
Fig. 7. Hidden unit output values frequency distributions for the Iris data set



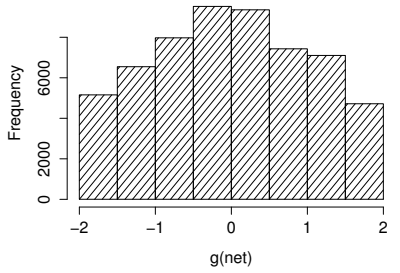
(a) Elliott, GBest PSO



(b) TanH, GBest PSO



(c) LeCun TanH, GBest PSO



(d) LeCun TanH, VN PSO

Fig. 8. Hidden unit output values frequency distributions for the Glass data set

LeCun tanH for both gBest and VN PSO are very close to 0.5, indicating no saturation, as discussed in Section III. Figures 8(c) and 8(d) show the corresponding frequency distributions. Indeed, highest frequencies are observed in the middle bins, indicating no saturation, and thus confirming the interpretation of the φ_B values.

The values of φ_B obtained on the Glass data set for different values of B once again converge on the same value for $B \geq 10$, indicating that $B = 10$ provides enough granularity.

Tables IV and V summarise the average E_C, ς_h, and φ_B for different number of bins B obtained for the Heart data set and the Diabetes data set, respectively. The largest values are shown in bold for each row, and the lowest values are shown in italics. There are once again discrepancies between the ς_h and the φ_B estimates of saturation, and φ_B once again describes the experimental data correctly, while ς_h provides ambiguous results. The corresponding frequency distributions are omitted for brevity, but the picture is quite similar to that observed on the Iris and the Glass data sets. Values of φ_B converge on the same value for $B \geq 10$ for both data sets. Thus, φ_B is not very sensitive to B, and B need not be optimised as a parameter. On all data sets considerd, $B \geq 10$ provided sufficient granularity. Therefore, this value is suggested as a default number of bins for the φ_B measure.

TABLE III. AVERAGE E_C, ς_h, AND φ VALUES FOR THE GLASS DATA SET, WITH CORRESPONDING STANDARD DEVIATION IN PARENTHESIS

g(net)	**Sigmoid**		**TanH**		**LeCun TanH**		**Elliott**	
Alg.:	gBest	VN	gBest	VN	gBest	VN	gBest	VN
E_C	0.438	0.4884	0.4372	0.455	0.4744	**0.4891**	*0.4357*	0.4535
	(0.0822)	(0.0728)	(0.0764)	(0.0648)	(0.0657)	(0.0653)	(0.078)	(0.0709)
ς_h	3.4534	2.8163	2.2161	1.9336	1.3895	*1.3452*	**3.8907**	3.3354
	(0.7454)	(0.4972)	(0.4478)	(0.4274)	(0.2983)	(0.351)	(0.8526)	(0.4817)
φ_5	0.7205	0.6841	**0.741**	0.7009	0.4901	*0.4769*	0.656	0.6357
	(0.059)	(0.0644)	(0.0543)	(0.0728)	(0.0548)	(0.0581)	(0.0454)	(0.04)
φ_{10}	0.729	0.693	**0.7483**	0.71	0.508	*0.4968*	0.6614	0.6417
	(0.0568)	(0.0613)	(0.052)	(0.0694)	(0.0507)	(0.0538)	(0.0436)	(0.0381)
φ_{20}	0.729	0.693	**0.7483**	0.71	0.508	*0.4968*	0.6614	0.6417
	(0.0568)	(0.0613)	(0.052)	(0.0694)	(0.0507)	(0.0538)	(0.0436)	(0.0381)
φ_{30}	0.729	0.693	**0.7483**	0.71	0.5082	*0.497*	0.6614	0.6417
	(0.0568)	(0.0613)	(0.052)	(0.0694)	(0.0507)	(0.0539)	(0.0436)	(0.0381)
φ_{50}	0.729	0.693	**0.7483**	0.71	0.5082	*0.497*	0.6614	0.6417
	(0.0568)	(0.0613)	(0.052)	(0.0694)	(0.0507)	(0.0539)	(0.0436)	(0.0381)

TABLE IV. AVERAGE E_C, ς_h, AND φ VALUES FOR THE HEART DATA SET, WITH CORRESPONDING STANDARD DEVIATION IN PARENTHESIS

g(net)	**Sigmoid**		**TanH**		**LeCun TanH**		**Elliott**	
Alg.:	gBest	VN	gBest	VN	gBest	VN	gBest	VN
E_C	0.2018	0.1918	**0.208**	0.1984	0.1915	0.1906	*0.1897*	0.1933
	(0.0308)	(0.0276)	(0.0245)	(0.0262)	(0.0243)	(0.0264)	(0.0269)	(0.0297)
ς_h	**7.5938**	6.6772	4.4627	3.771	3.9794	*3.7456*	7.0287	5.7845
	(1.3151)	(1.3077)	(0.8374)	(0.6072)	(0.6476)	(0.7286)	(1.6021)	(0.9499)
φ_5	0.8774	0.8636	**0.8891**	0.8709	0.8095	0.8008	0.7747	*0.752*
	(0.0205)	(0.0291)	(0.0204)	(0.0208)	(0.0349)	(0.0417)	(0.0315)	(0.0238)
φ_{10}	0.8806	0.8674	**0.8921**	0.8744	0.8148	0.8064	0.7772	*0.7548*
	(0.0199)	(0.0283)	(0.0198)	(0.0201)	(0.0338)	(0.0401)	(0.0309)	(0.0234)
φ_{20}	0.8806	0.8674	**0.8921**	0.8744	0.8148	0.8064	0.7772	*0.7548*
	(0.0199)	(0.0283)	(0.0198)	(0.0201)	(0.0338)	(0.0401)	(0.0309)	(0.0234)
φ_{30}	0.8806	0.8674	**0.8921**	0.8744	0.8151	0.8065	0.7772	*0.7548*
	(0.0199)	(0.0283)	(0.0198)	(0.0201)	(0.0337)	(0.0401)	(0.0309)	(0.0234)
φ_{50}	0.8806	0.8674	**0.8921**	0.8744	0.8151	0.8065	0.7772	*0.7548*
	(0.0199)	(0.0283)	(0.0198)	(0.0201)	(0.0337)	(0.0401)	(0.0309)	(0.0234)

TABLE V. AVERAGE E_C, ς_h, AND φ VALUES FOR THE DIABETES DATA SET, WITH CORRESPONDING STANDARD DEVIATION IN PARENTHESIS

g(net)	**Sigmoid**		**TanH**		**LeCun TanH**		**Elliott**	
Alg.:	gBest	VN	gBest	VN	gBest	VN	gBest	VN
E_C	0.2526	*0.2483*	0.2604	0.2574	0.2578	0.2541	**0.2682**	0.2569
	(0.0336)	(0.0263)	(0.0364)	(0.0325)	(0.0244)	(0.0225)	(0.0265)	(0.0348)
ς_h	**4.7613**	4.2917	2.729	2.3961	2.5968	*1.9721*	3.7401	3.1226
	(1.3284)	(1.1398)	(0.5767)	(0.4632)	(0.8283)	(0.5306)	(0.7939)	(0.5866)
φ_5	0.7499	0.7511	**0.7858**	0.7748	0.67	*0.6239*	0.6555	0.6359
	(0.0795)	(0.0593)	(0.0502)	(0.0391)	(0.0694)	(0.0689)	(0.0357)	(0.0329)
φ_{10}	0.7575	0.7584	**0.7922**	0.7814	0.6805	*0.6362*	0.6611	0.6418
	(0.0764)	(0.0572)	(0.0482)	(0.0378)	(0.0663)	(0.0657)	(0.0346)	(0.0318)
φ_{20}	0.7575	0.7584	**0.7922**	0.7814	0.6805	*0.6362*	0.6611	0.6418
	(0.0764)	(0.0572)	(0.0482)	(0.0378)	(0.0663)	(0.0657)	(0.0346)	(0.0318)
φ_{30}	0.7575	0.7584	**0.7922**	0.7814	0.6807	*0.6362*	0.6611	0.6418
	(0.0764)	(0.0572)	(0.0482)	(0.0378)	(0.0664)	(0.0657)	(0.0346)	(0.0318)
φ_{50}	0.7575	0.7584	**0.7922**	0.7814	0.6807	*0.6362*	0.6611	0.6418
	(0.0764)	(0.0572)	(0.0482)	(0.0378)	(0.0664)	(0.0657)	(0.0346)	(0.0318)

Algorithms were ranked based on their mean φ_{10} values, and the resulting ranks are reported in Table VI. The two-tailed non-parametric Mann-Whitney U test [14] was used to determine whether the difference in φ_{10} values between any two algorithms was statistically significant. The choice of the significance test is based on [15], where the authors showed that the Mann-Whitney U test is safer than the parametric tests such as the t-test, since the Mann-Whitney U test assumes neither normal distributions of data, nor homogeneity of variance. The null hypothesis $H_0 : \mu_1 = \mu_2$, where μ_1 and μ_2 are the means of the two samples being compared, was evaluated at a significance level of 95%. The alternative hypothesis was defined as $H_1 : \mu_1 \neq \mu_2$.

TABLE VI. AVERAGE ALGORITHM RANKS: φ_{10}

Algorithm	$g(net)$	Iris	Glass	Heart	Diabetes	Average Rank	
GBest PSO	Sigmoid	6.5	5.5	6.5	6.5	6.25	
	TanH	6.5	5.5	6.5	6.5	6.25	4.65625
	LeCun TanH	1.5	1.5	3.5	3.5	2.5	
	Elliott	3.5	5.5	2	3.5	3.625	
VN PSO	Sigmoid	6.5	5.5	6.5	6.5	6.25	
	TanH	6.5	5.5	6.5	6.5	6.25	4.34375
	LeCun TanH	1.5	1.5	3.5	1.5	2	
	Elliott	3.5	5.5	1	1.5	2.875	

Table VI shows that on average across all problems considered, LeCun tanH saturated the least. This observation agrees with the theory, as the whole idea of using the modified tanH was to improve NN learning ability and decrease saturation [4]. Elliott with its soft slope came second-lowest in terms of saturation, which corresponds to the observations made in [2]. Both sigmoid and tanH exhibited the same level of saturation, higher than that of LeCun tanH and Elliott. On average, VN PSO saturated less than gBest PSO. This makes sense, as gBest is a fully connected topology: all particles share the global attractor (the social component in Eq.(11)), which may result in faster algorithm convergence and overfitting. The behaviour of neighbourhood topologies is very problem-specific, though: Figure 5 confirms that on the Iris data set, it is the VN topology that yielded higher saturation.

No conclusions regarding the classification efficiency can be made at this point for two reasons: firstly, PSO parameters were not optimised for each problem, thus any comparison would be unfair; secondly, no statistically significant difference was observed between the E_C values. Determining the relationship between the saturation level and classification accurancy is out of the scope of this study.

V. CONCLUSIONS

This paper presented a simple single-valued saturation measure for NNs based on activation function outputs. The degree of saturation is an important characteristic of a trained NN that can provide insight into both the model and the training algorithm. The proposed measure is applicable to all bounded activation functions, is independent of the activation function output range, and allows direct statistical comparisons between NNs employing different activation functions. The proposed measure was tested on four different classification problems, four different activation functions, and two different training algorithms. The results were easy to interpret, and described the observed saturation behaviour well. The proposed saturation measure is bounded: it tends to 1 as the degree of saturation increases, and tends to zero otherwise. Compared to a saturation measure used previously in literature, the proposed measure is much more robust and unambiguous. The single tunable parameter of the proposed measure, number of bins B, converges for $B \geq 10$. Thus, $B = 10$ can be used without any further tuning.

Out of the four activation functions considered in the experiments, LeCun tanH saturated the least. This corresponds well with the original paper, where the modified version of tanH was suggested as a saturation-resistant activation function

[4]. Out of the two PSO topologies considered, VN PSO saturated less than GBest PSO.

Further studies will include an analysis of NN saturation in relation to such qualities of NNs as the ability to learn, the ability to generalise, and classification accuracy. Means of controlling saturation in PSO training, and using saturation as an extra NN learning guide will be considered. The suggested measure will also be used to quantify the propensity to saturate for different training algorithms.

REFERENCES

[1] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *International conference on artificial intelligence and statistics*, 2010, pp. 249–256.

[2] A. Rakitianskaia and A. P. Engelbrecht, "Training high-dimensional neural networks with cooperative particle swarm optimiser," in *International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2014, pp. 4011–4018.

[3] S. Lawrence, A. C. Tsoi, and A. D. Back, "Function approximation with neural networks and local methods: bias, variance and smoothness," in *Proceedings of the Australian Conference on Neural Networks*. Canberra, Australia: Australian National University, 1996, pp. 16–21.

[4] Y. A. LeCun, L. Bottou, G. B. Orr, and K.-R. Müller, "Efficient backprop," in *Neural networks: Tricks of the trade*. Springer, 2012, pp. 9–48.

[5] A. Rakitianskaia and A. Engelbrecht, "Saturation in PSO neural network training: Good or evil?" in *IEEE Congress on Evolutionary Computation (CEC)*. IEEE, 2015.

[6] A. Gupta and S. M. Lam, "Weight decay backpropagation for noisy data," *Neural Networks*, vol. 11, no. 6, pp. 1127–1138, 1998.

[7] M. Carvalho and T. B. Ludermir, "Particle swarm optimization of feed-forward neural networks with weight decay," in *Sixth International Conference on Hybrid Intelligent Systems (HIS'06)*. IEEE, 2006, pp. 5–8.

[8] D. L. Elliott, "A better activation function for artificial neural networks," Institute for Systems Research, University of Maryland, Tech. Rep. 93-8, 1993.

[9] J. Kennedy and R. C. Eberhart, "Particle swarm optimization," in *Proceedings of the IEEE International Conference on Neural Networks*, vol. IV. Piscataway, USA: IEEE, 1995, pp. 1942–1948.

[10] Y. Shi and R. Eberhart, "A modified particle swarm optimizer," in *Proceedings of the IEEE International Conference on Evolutionary Computation*. IEEE, 1998, pp. 69–73.

[11] J. Kennedy and R. Mendes, "Population structure and particle swarm performance," in *Proceedings of the Congress on Evolutionary Computation*. Piscataway, USA: IEEE, 2002, pp. 407–412.

[12] R. C. Eberhart and Y. Shi, "Comparing inertia weights and constriction factors in particle swarm optimization," in *Proceedings of the IEEE Congress on Evolutionary Computation*, vol. 1. Piscataway, USA: IEEE, 2000, pp. 84–88.

[13] A. B. van Wyk and A. P. Engelbrecht, "Overfitting by PSO trained feedforward neural networks," in *Proceedings of the IEEE Congress on Evolutionary Computation*, 2010, pp. 1–8.

[14] H. B. Mann and D. R. Whitney, "On a test of whether one of two random variables is stochastically larger than the other," *Annals of Mathematical Statistics*, vol. 18, no. 1, pp. 50–60, 1947.

[15] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *Journal of Machine Learning Research*, vol. 7, pp. 1–30, 2006.