Complex Systems **9** (1995) 287–304

Information-theoretics Based Error-metrics for Gradient Descent Learning in Neutral Networks

Joseph C. Park
Atlantic Undersea Test and Evaluation Center (AUTEC),
West Palm Beach, FL 33401, USA

Perambur S. Neelakanta
Salahalddin Abusalah
Dolores F. De Groff
Raghavan Sudhakar
Department of Electrical Engineering,
Florida Atlantic University,
Boca Raton, FL 33431, USA

Abstract. Conventionally, square error (SE) and/or relative entropy (RE) error functions defined over a training set are adopted towards optimization of gradient descent learnings in neural networks. As an alternative, a set of divergence (or distance) measures can be specified in the information-theoretic plane that functionally have pragmatic values similar to (or improved upon) The SE or RE metrics. Kullback-Leibler (KL), Jensen (J), and Jensen-Shannon (JS) measures are suggested as possible information-theoretic error-metric candidates that are defined and derived explicitly. Both conventional SE/RE measures, as well as the proposed information-theoretic error-metrics, are applied to train a multilayer perceptron topology. This is done in order to elucidate their relative efficacy in deciding the performance of the network as evidenced from the convergence rates and training times involved. Pertinent simulation results are presented and discussed.

1. Introduction

In typical learning algorithms of neural networks (NNs), an error-metric is usually specified and minimized towards the optimization of network performance [1]. That is, in pattern categorization efforts, NNs are trained conventionally from examples using the analog attributes of the activity of the output units. In the relevant procedure, the analog output (activity) parameter of a neuron presented with an input pattern (specified by a set of parametric values), is compared with the teacher parameters. The resulting mean square of the error between the value(s) is minimized to achieve the network training. The quadratic error-metric and the relative entropy (RE) error-metric are based on the values of output and target parameters and their logarithmic values, respectively. Network training with such parametric values is referred to in this paper as a goal-seeking task envisaged in a parametric spread space. Instead of analog values of the parameters, a probablistic interpertation of the of target activities can be specified to define error measure(s) for the gradient descent learning in the information-theoretic domain. Such informatic specifications are implicitly negative entropy entities, and the corresponding error-metrics developed presently refer to conditional information measures. Elucidation of such cross-entropy parameters, based on the principle of discrimination-information (or conditional information), leads to what are known as the divergence or distance measures. The basis for using such cross-entropy based paradigms (*in lieu* of the conventional sqare error (SE)/RE error metrics) refers to deciding the extent of discrimination between the information associated with the statistical distributions of the network output and the target being pursued. Relevant underlying considerations are based on the minimum entropy principle instead of the maximum entropy concept built on Shannon's information measure.

As is well known, a major function of a neural complex is the goal-related, self-organizing (or self-regulating) effort dictated by an objective function (or teacher value) and viewed within a set of bounds. The associated randomness of disorderliness due to the presence of noise would cause the neural system parameters (specified by a vector set) to veer from the system objective. This deviatory response in a NN can be quantified by an ensemble of diversion factors *vis-a-vis* the neural environment.

Disorderliness in a NN can be defined by measuring the deviation of a selected variable, say the output $\mathbf{y}_i$, with a specified target standard $\mathbf{y}_T$. In a geometrical representation such as Figure 1, $\mathbf{y}_T$ can be denoted by a vector corresponding to the center of a region of orderliness wherein a stipulated stochastic extent of orderliness is dictated within certain bounds. The disorderliness at the ith realization in the parameter spread space Ω_S of Figure 1 can be written as in [2]:

$$Y_i = |y_i - y_T| - D(y_i) \tag{1}$$

where $|y_i - y_T|$ refers to the magnitude of the error vector and $D(\mathbf{y}_i)$ is the distance from the center to the boundary of a quasi-ordered region close to the target or goal. Equivalently, a goal-associated positional entropy can be specified by H_{yi} at the ith elementary subspace in the entropy space Ω_H. Elsewhere, say at the jth subspace, let H_{yj} represent the goal-associated positional entropy vector perceived. Now, one can seek information in the sample space of $\mathbf{y}$ for discrimination in favor of H_yi against H_{yj}, or symmetrically for discrimination in favor of H_{yj} against H_{yi}. Physically, these conditions represent whether the ith realization (respectively) would enable achieving the goal being sought [2].

The entropy space Ω_H is affinely similar to the parameter spread space Ω_S such that each value of H_y in the entropy space could be mapped onto

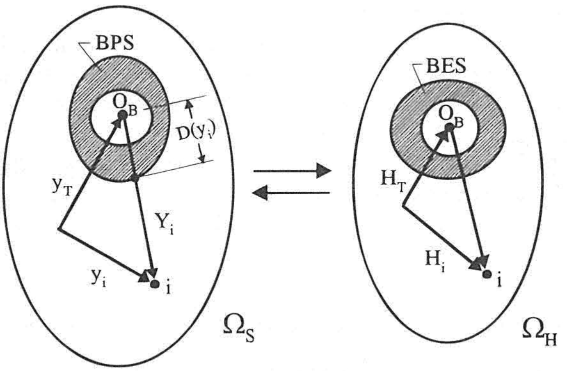

Figure 1: Parameter spread space and entropy space of the neural complex. O_B: locale of the objective function; BPS: boundary of the quasi-order of parameter space; BES: boundary of the quasi-ordered region of entropy space.

the parameter spread space on a one-to-one basis. In the entropy subspace of the neural complex, the probability of encountering the ith cell wherein the disorganization is observed can be denoted by p_i. With this prescription of a probabilistic attribute to the goal-related entropy, H_y should satisfy certain conditions stipulated by the lemmas presented in the appendix.

Consistent with the aforesaid description of the neural complex in the information-theoretic plane, the development of an optimization algorithm as done in this paper refers to the minimization of the error-metric estimated in respect to the objective or target value. Such an effort also determines the convergence rate and the performance aspects of the error-metrics in the training cycles as well as in achieving the stability of the network.

In the relevant pursuits, the RE measure *versus* the SE metric in parametric spread space has been addressed in [3]. Further, in [4] a comparison between SE and RE errors is presented using several optimization algorithms. Improvements towards the convergence specified in terms of iterations required to reach an acceptable performance (i.e., output being close to the target value) has been observed with RE measures. Such improvements have been regarded as the realization of an "accelerated learning" that results from the use of the RE metric *in lieu* of the SE measure.

Apart from the conventional RE measure specified in the parametric space, it is also possible to define several distance measures in the information-theoretic plane based on the divergence concept indicated before (e.g., [5]). In this paper, a set of such distance measures are defined explicitly and the corresponding error-metrics are elucidated. The distance measures are applied to a multilayer perceptron network and simulation results pertinent to the convergence rate (or the number of iterations warranted towards convergence), the network stability, and the accuracy of the output are compared with those of the SE and RE based simulations.

In section 2, explicit definitions of the proposed distance measures (error-metrics) are furnished, relevant expressions are derived and elaborated. Section 3 describes a multilayer perceptron network implemented for the experimental simulations. Details on the initial condtions and the experimental protocols are given. The results of the experiments with the defined error-metrics are presented in section 4 along with the corresponding results obtained *via* SE and RE measures. Hence, the relative performance aspects are discussed. Section 5 enumerates the conclusions drawn from the experimental results.

2. Information-theoretic based divergence measures

2.1 Kullback-Leibler and Jensen measures

The basic set of elements $\mathbf{y}_i$ that deviate from the target value $\mathbf{y}_T$ (Figure 1) have probability measures p_i that are absolutely continuous with respect to each other. These probabilities can be specified by generalized probability densities $f(y_i)$ such that $0 < f(y_i) < \infty$ and $p_i = \int f(y_i)dy_i$ with $0 \leq p_i \leq 1$. The average information for discrimination in favor of H_{yi} against H_{yj} can be written in terms of a divergence measure known as the Kullback-Leibler (KL) measure. It is given by

$$I(i:j,y) = (1/p_i)\int \log[f(y_i)/f(y_j)]dp_i. \tag{2}$$

Considering also the average information for discrimination in favor of H_{yj} against H_{yi}, namely $I(j:i,y)$, a symmetrical measure of divergence (known as the Jensen or J-measure) can be written as

$$J(i:j,y) = I(i;j,y) + I(j:i,y) = \sum_{y_i}(p_i - p_j)\log(p_i/p_j). \tag{3}$$

This J-divergence represents the divergence of disorganization associated with the subspace regions of the ith realization and that of the jth realization [7].

Additionally, each of the realizations can be weighted with respect to their probability distribution to specify its individual strength in the goal-seeking endeavor. Suppose Π_i and Π_j ($\Pi_i, \Pi_j \geq 0$ and $\Pi_i + \Pi_j = 1$) are the weights of the two probabilities p_i and p_j respectively. Then a generalized divergence measure (known as the Jensen-Shannon (JS) measure), can be stipulated as follows from [7]:

$$JS_\Pi(p_i:p_j) = H(\Pi_i p_i + \Pi_j) - \Pi_i H(p_i) - \Pi_j H(p_j). \tag{4}$$

This measure is nonnegative and equal to zero when $p_i = p_j$. It also provides the upper and lower bounds for the Bayes' probability of error. The JS divergence is ideal for describing the variations between the subspaces or the goal-seeking realizations as in the entropy space of the neural complex (Figure 1). It also measures the distance between the random-graph depictions of such realizations pertinent to the entropy plane Ω_H.

General characteristics of the distance measures are as follows.

1. $I(p_i : p_j)$ continuous of p_i and p_j.
2. When $I(p_i : p_j) = I(p_j : p_i)$, the divergence refers to the symmetric property of the error-metric.
3. $I(p_i : p_j) \geq 0$ is the nonnegativity property of the error-metric. Equality occurs if and only if $p_i = p_j$, in which case the corresponding property is known as the identity property.
4. $I(p_i : p_j) + I(p_j : p_k) \geq I(p_i : p_k)$ is the triangle inequality property.
5. $I(p_i : p_j)$ is a concave function of $(p_1, p_2, \ldots, p_n)$.
6. When $I(p_i : p_j)$ is minimized, subject to known linear constraints, none of the resulting minimized probabilities should be negative.

Within the framework of the various properties enumerated above, the cross-entropy based distance measures can be derived for a NN as indicated in section 3.

3. Distance measures as error-metrics of a neural network

Considering the NN depicted in Figure 2, let O_i represent the output at the ith cell, the corresponding target sought is specified as T_i. Then the following KL cross-entropy can be written:

$$I(p_i : q_i) = \sum_{i=1}^{N} p_i \log(p_i/q_i), \tag{5}$$

where $i = 1, 2, \ldots, N$ enumerates the number of cells or offers an index for the output units; and p_i is the probability of O_i which complies with the following hypothesis: In terms of learning, probabilities of a set of hypotheses are represented by output limits using p_i and q_i. That is, for a hypothesis

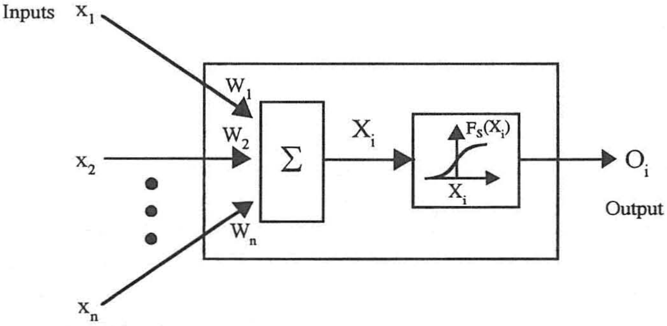

Figure 2: An artificial neuron.

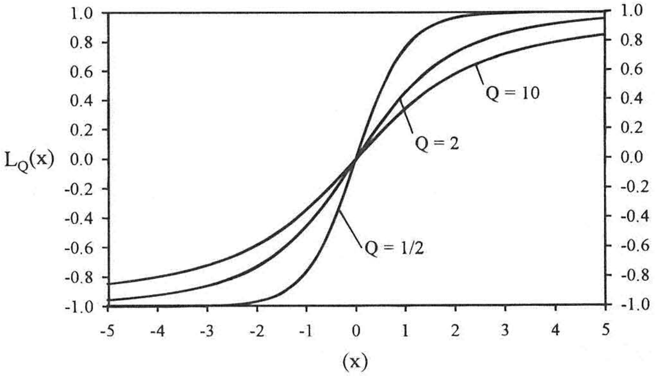

Figure 3: The sigmoidal Bernoulli activation function $L_Q(x)$ with values of $Q = 1/2$, 2, and 10.

represented by the ith unit, $p_i = 0$ means definitely false, and $p_i = +1$ means definitely true. Similarly, q_i refers to a target set of probabilities such that $q_i = 0$ and $q_i = +1$ set the false-true limits of the target value.

Let $X_i = \sum W_{ij}x_j$ represent the weighted sum of the multi-inputs $\{x_j\}$ with W_{ij} being the weighting factor across the interconnection between the ith and jth cells. This summed input is processed by a nonlinear activation function F_S to produce the neuron's output signal O_i as depicted in Figure 2. That is, each neuron evaluates its inputs and "squashes" its permissible amplitude range of output to some finite value $O_i = F_s(X_i)$ in a nonlinear fashion. As indicated in [8], the Bernoulli function $L_Q(z) = a[\coth(az)] - b[\coth(bz)]$ (with $a = 1 + 1/2Q$ and $b = 1/2Q$) is a stochastically justifiable sigmoidal function to represent the nonlinear activation F_S of a neuron (Figure 3). Here, Q is a single parameter that controls the slope of F_S at the origin. Further, $L_Q(z)$ is a monotonic function, differentiable everywhere and squashed between the limits -1 and $+1$. When $Q = 1/2$, the Bernoulli function becomes $\tanh(z)$, which is the conventional sigmoid adopted in NN algorithms. $L_Q(z)$ can be approximated as:

$$L_Q(z) = \begin{cases} +1; & z > +(1/\alpha) \\ \alpha; & -(1/\alpha) < z + (1/\alpha) \\ -1; & z < -(1/\alpha) \end{cases} \tag{6}$$

where $\alpha = (2a - 1)/3 = (2b + 1)/3$.

$p(O_i)$ can be deduced in terms of the Bernoulli function chosen as the sigmoid as follows. Let the probability density function (f) of the weighted sum of the inputs, namely X_i, be uniform (of constant value $\alpha/2$) over the interval $-1/\alpha$ to $+1/\alpha$. That is, $f(X_i)$ is equal to $\alpha/2$ in the interval $-1/\alpha \leq X_i \leq +1/\alpha$ or zero otherwise. Hence, the probability density function of the otuput $f(O_i)$ can be obtained by the following transformation:

$$f(O_i) = [f(X_i)/F_s'(X_i)]_{X_i = F_s^{-1}(O_i)}, \tag{7}$$

where the prime denotes the differentiation with respect to the argument. With the assumed uniform distribution of X_i, $f(O_i)$ reduces to the following uniform distribution:

$$f(O_i) = \begin{cases} 1; & 0 \leq O_i \leq 1, \\ 0; & \text{otherwise.} \end{cases} \tag{8}$$

Therefore, $p_i(O_i)$ can be deduced as:

$$p_i(O_i) = \int_0^{O_i} f(x)dx = (1 + \alpha X_i)/2 \tag{9}$$

which guarantees that $p_i(O_i) = 0$ at $X_i = -1/\alpha$, and that $p_i(O_i) = 1$ at $X_i = +1/\alpha$.

Assuming that the target T_i can be obtained from the following transformation:

$$f(T_i) = K[f(X_i)/G'(X_i)_{X_i=G^{-1}(T_i)}, \tag{10}$$

where K is a normalization constant required to realize the following identity on total probability:

$$K \int f(T_i)dT_i \equiv 1. \tag{11}$$

Denoting $G'(X_i)|_{X_i=G^{-1}(T_i)} = D_1(T_i)$ and $D_2(T_i) = \int_0^{T_i} dy/[D_1(y_i)]$, the probability of T_i is obtained as

$$q_i(T_i) = [D_2(T_i) - D_2(0)]/[D_2(1) - D_2(0)]. \tag{12}$$

Now, inasmuch as $p_i(O_i)$ and $q_i(T_i)$ are explicitly known in terms of X_i and $G(X_i)$, the other distance measures in the informatic plane can be written as follows.

1. KL Measure:

$$\varepsilon_{KL} = \sum p_i \log(p_i/q_i). \tag{13}$$

2. J Measure:

$$\varepsilon_J = \sum p_i \log(p_i/q_i) + \sum q_i \log(q_i/p_i). \tag{14}$$

3. JS Measure:

$$\varepsilon_{JS} = \Pi_1 \sum p_i \log(p_i/q_i) + \Pi_2 \sum q_i \log(q_i/p_i), \tag{15}$$

where Π_1 and Π_2 are weights such that $\Pi_1 + \Pi_2 = 1$.

For simulation purposes and to compare results, SE and RE measures of a NN with O_i and T_i as the output and target values respectively can be written as

$$\varepsilon_{SE} = 1/2 \sum (T_i - O_i)^2 \tag{16}$$

and

$$\begin{aligned}\varepsilon_{RE} = &(1/2)(1+q_i)\log[(1+q_i)/(P_i+p_i)] \\ &+(1/2)(1-q_i)\log[(1-q_i)/(P_i+p_i)]\end{aligned} \tag{17}$$

where $P_i^2 = a^2 + b^2[\coth^2(bX_i) + \operatorname{cosech}^2(bX_i)] - 2ab/\coth(aX_i)\coth(bX_i)$.

In all of the preceding cases, the sigmoidal function refers to $L_Q(X_i)$. For the comparison of the aforesaid error-metrics in dictating the performance characteristics of a NN each error-metric is applied in training a mulitlayer perceptron network. Its description is presented in section 4.

4. Computer simulations with a multilayer perceptron network

4.1 Description of the network

A multilayer perceptron is implemented to evaluate the training effectiveness of backpropagation in the prediction of a sine wave function $[\sin(m\pi\alpha x_i/2)+1]/2$, where $m = (-1)^n(2n+1), n = 0, 1, 2, \ldots$ using the various distance measures specified previously to calculate the synaptic weight modifications.

The test network is depicted schematically in Figure 4 and consists of nine input units, ten units in the hidden layer, and a single output unit. The activation functions in the input-to-hidden layer are Bernoulli sigmoids

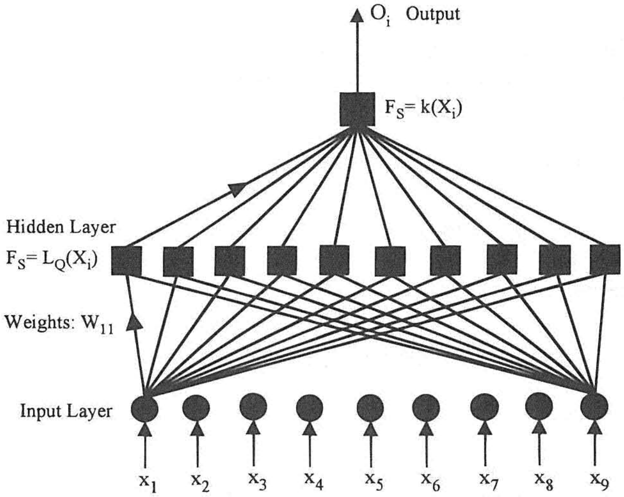

Figure 4: Test NN topology of the multilayer perceptron.

$L_Q(x)$, and are linear in the output layer. The nine input units are trained at x values over the interval $x_i \epsilon [-1, 1]$ at increments of 0.25, while the output unit is evaluated at a x value of $x_T = 0.375$. The weights and thresholds are initialized to be uniformly distributed pseudorandom values over $[-1, 1]$, the input and hidden layers have an additional bias unit clamped to -1 which is connected through a trainable weight to each unit in the hidden and output layers respectively. These bias units provide a trainable offset to the activation function origins for the units in the hidden and output layers, thereby enhancing the convergence rate.

The network is sequentially presented seventy-five sine wave training sets at the nine inputs (x_i) during each training epoch. The training sets are specified by $T_\gamma = [\sin(m\pi\alpha\gamma x_i/2) + 1]/2$, where γ is a uniformly distributed pseudorandom variate in the range $[-1, 1]$. The output of the network specified at x_T is used in the backpropagation mode with the gradient descent in order to adjust the weights over 150 training epochs. After the training, the network is set to compute the values of the sine function $[\sin(m\pi\alpha\gamma x_T/2) + 1]/2$ at fifty equally spaced points over the interval $\gamma \in [-1, 1]$.

4.2 Backpropagation algorithm

The essence of the backpropagation algorithm is that the synaptic weights are adjusted to minimize the local error of the network with a given knowledge of the target output T_i. The fundamental quantity used in the determination of the weight-states is the error, or the distance (ε_i) of the network output (O_i) of the ith unit from the target value T_i.

The basic prescription to adjust the weights at the nth training step follows the well-known Widrow-Hoff algorithm, namely, $w_{ij}(n) = w_{ij}(n - 1) + \Delta w_{ij}$ where $\Delta w_{ij} = \eta\delta_j O_i$ and w_{ij} is the weight from unit i to j, η is the learning rate, and δ_j is the effective gradient. The effective gradient has two distinct definitions depending on whether or not a target value is available for a particular unit.

In the case of network output units for which a target is known, δ_j is defined as the deviation of the jth unit multiplied by the derivative of the activation function evaluated at the output value of the ith unit. That is, $\delta_j = (\partial O_j/\partial\sigma_j)\varepsilon_j$, where σ_j represents the summed input to the activation function, or $O_j = F_S(\sigma_j)$.

When the unit resides in a hidden or input layer, the target value is not available for the computation of the effective gradient. Therefore, a modified definition in which the product of cumulative effective gradients with the interconnection weights are back-propagated to these units *via* the relation specified as $\delta_j = (\partial O_j/\partial\sigma_j)\sum_j \delta_j w_{ij}$. In the case of the conventional SE metric, the sign of δ is decided by the simple arithmetic difference between the target and output. The direction of gradient descent is controlled by the feedback obtained from the comparison of target *versus* output difference. The cross-entropy metrics involving logarithmic functions are, however, strictly nonnegative; and therefore would not allow for δ to change its sign in response to the target *versus* output differences resulting in a loss of feedback control in the weight changing algorithm. To remedy this situation, the calculation of effective gradient while using the cross-entropy error-metrics is multiplied by ±1, depending upon the sign of the target-output difference. That is, the value of delta is specified by $\delta_i = \delta_i \operatorname{signum}(T_i - O_i)$.

In regions of the error surface where large gradients exist, the δ terms may become inordinately large. The resulting weight modifications will also be extensive, leading to large oscillations of the output, bypassing the true error minimum. The learning coefficient can be set to an extremely small value to counteract this tendency; however, this would drastically increase the training time. To avoid this problem, the weight modification can be given a memory so that it will no longer be subject to abrupt changes. That is, the weight-change algorithm is conventionally specified by $\Delta w_{ij}(n) = \eta\delta_j O_i + \lambda[\Delta w_{ij}(n-1)]$, where λ is known as the momentum parameter. If λ is set to a value close to 1, the search in the parameter space will be determined by the gradient accumulated over several epochs instead of a single iteration, thereby improving the stability of the network towards the convergence.

5. Results and discussions

The multilayer perceptron described in section 3 was implemented to predict values of the target function at fifty equally spaced points over the interval $[-1, 1]$ using the five error-metrics defined by equations (13) through (17). The value of the network parameter Q is adjusted to control the training effectiveness and stability of the network. Further, the value of the learning coefficient η can also be altered to influence the convergence rates and the ultimate accuracy of the performance of the network. To assay the consequences of varying η and Q in deciding network performance, each of the distance measures (namely KL, J, and JS) was implemented in training the network with values of η ranging from 0.0001 to 0.2 and for Q values of 1/2, 2, and 10. The same random initializations of the interconnection weights for each set of η and Q was used.

After training, the root mean-square (rms) deviation of the functional prediction over the fifty points x_n was recorded. Table 1 lists the results for the set of parameters ($Q = 1/2$, $m = 2$), ($Q = 2$, $m = 4$), and ($Q = 10$, $m = 8$). When the error-metric magnitude exceeded 10^4 during training, the network was considered to have diverged and the result is indicated in Table 1 as DIV.

Relevant to these test studies, the highlighted vales presented in Table 1(c) represent the minimum rms errors at the network output for the various test error-metrics under consideration with $Q = 10$ and indicate the corresponding η values to be used for training. Examination of Table 1 reveals that for all of the error-metrics considered (and in particular for the RE measure), increasing the value of Q enhances the tolerance of the network optimization to larger learning rates. It is also observed that for each error-

Table 1: Root mean-square (rms) deviation of the network output for the SE, RE, KL, J, and JS distance measures *versus* the learning coefficient η. (a): ($Q = 1/2$, $m = 2$). (b): ($Q = 2$, $m = 4$). (c): ($Q = 10$, $m = 8$).

(a)

	η	SE	RE	KL	J	$\mathrm{JS}_{\Pi=0.5}$
$Q = 1/2$ $m = 2$	0.0001	0.0924	0.5205	1.5708	0.3700	0.4584
	0.0010	0.0366	0.2467	3.3927	0.1402	0.2189
	0.0100	DIV	0.0289	6.6301	0.2629	0.1436
	0.1000	DIV	DIV	95.4445	57.5541	46.5342
	0.2000	DIV	DIV	105.7281	53.0585	57.5541

(b)

	η	SE	RE	KL	J	$\mathrm{JS}_{\Pi=0.5}$
$Q = 2$ $m = 4$	0.0001	0.0531	0.4572	1.13710	0.4242	0.4964
	0.0010	0.0295	0.0868	6.2400	0.0935	0.0886
	0.0100	0.0254	0.0579	16.1989	0.2538	0.0945
	0.1000	DIV	0.0985	25.9595	83.3622	187.5787
	.2000	DIV	2.0794	110.5492	91.9989	83.3622

(c)

	η	SE	RE	KL	J	$\mathrm{JS}_{\Pi=0.5}$
$Q = 10$ $m = 8$	0.0001	0.1372	0.5135	0.7664	0.4766	0.4993
	0.0010	0.0240	0.2182	5.3636	0.0508	0.1654
	0.0100	0.0181	0.0275	28.6065	0.0980	0.0677
	0.1000	DIV	0.2005	150.1648	16.2894	0.1195
	0.2000	DIV	0.3980	73.8275	49.4355	16.2894

metric, the minimum rms error in predicting the target function is achieved at $Q = 10$.

For the performance comparison of the error metrics, the values of the learning rate η corresponding to these minimum values of the rms error (at the output) are considered in realizing the network output (optimized towards the target function) in each case of the error-metric under discussion. Figure 5 presents the network convergence data and the output for the SE error-metric predicted with the value of η decided by the minimum rms error of the output indicated in Table 1(c). The left plot (a) is the magnitude of the SE *versus* training epoch, normalized with respect to the maximum error value. The right plot (b) depicts the target and network output values of the sine function. Convergence is achieved by the 80th training epoch. It can be noted that the error in the prediction of the sine function is appreciably small over the entire interval.

The results for the RE error-metric are shown in Figure 6. Again, the η value used corresponds to the minimum rms error in the output (as in Table 1(c)). In this case, the network converges after 60 training epochs to an almost constant value. The error in predicting the sine function is negligibly small, as in the SE case, over the interval $[-1, 1]$.

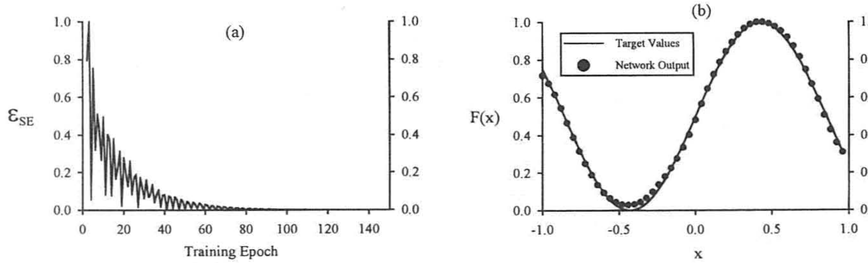

Figure 5: Network training and prediction of the target function with the SE error-metric ($Q = 10$, $\eta = 0.01$). (a) Distance measure $\varepsilon_{\mathrm{SE}}$ versus training epoch. (b) Target and predicted values of the sine function.

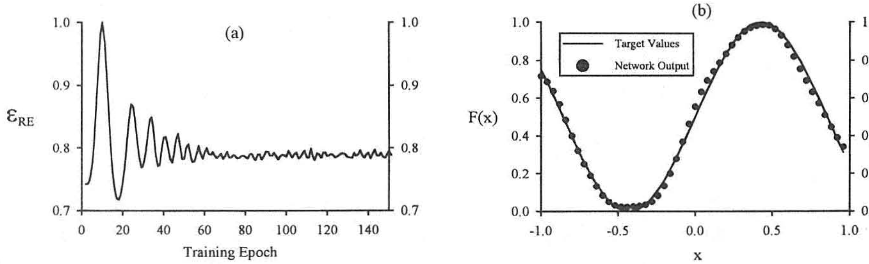

Figure 6: Network training and prediction of the target function with the RE error-metric ($Q = 10$, $\eta = 0.01$). (a) Distance measure $\varepsilon_{\mathrm{RE}}$ versus training epoch. (b) Target and predicted values of the sine function.

The KL error-metric performance is shown in Figure 7, where it is observed that the distance measure does not reach an equilibrium; hence, network convergence appears to be unstable with respect to the training epochs. Network performance under the control of the J-measure is presented in Figure 8, with $Q = 10$ and η dictated by the minimum rms error of the output as given in Table 1(c). It can be seen that the convergence occurs after approximately 50 epochs and the network output is comparable to the target objective over the prediction interval.

Figures 9, 10, and 11 depict the effectiveness of the JS-measure in training the network under similar conditions of Q and η (as per Table 1(c)). The functional output is predicted for the weighting values of $\Pi_1 = 0.25$, 0.5, and 0.75 respectively, with $\Pi_2 = (1 - \Pi_1)$. In Figures 9, 10, and 11 the only stable configuration pertinent to these data is observed only in the case of symmetrical weighting, namely, $\Pi_1 = \Pi_2 = 0.5$, where network performance is seen acceptable over the entire interval.

To demonstrate the utility of the Bernoulli activation function parameter Q in desensitizing the error-metric oscillations during training, as well as in-

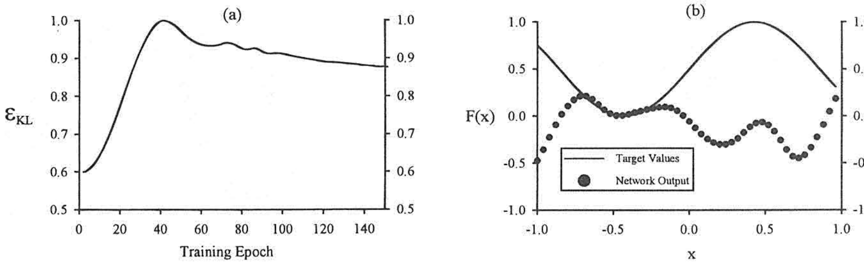

Figure 7: Network training and prediction of the target function with the KL error-metric ($Q = 10$, $\eta = 0.0001$). (a) Distance measure ε_{KL} versus training epoch. (b) Target and predicted values of the sine function.

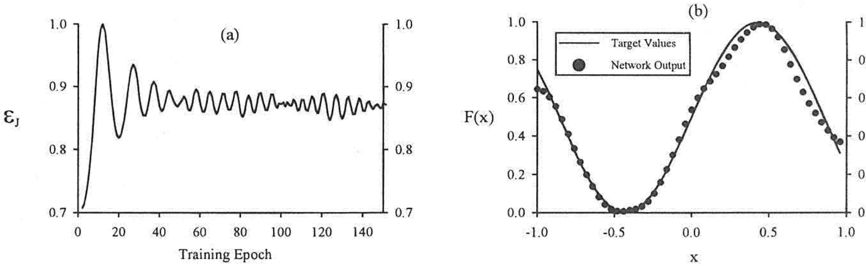

Figure 8: Network training and prediction of the target function with the J error-metric ($Q = 10$, $\eta = 0.001$). (a) Distance measure ε_J versus training epoch. (b) Target and predicted values of the sine function.

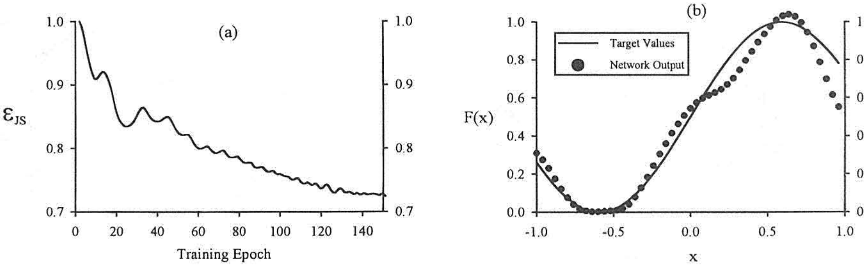

Figure 9: Network training and prediction of the target function with the JS error-metric ($\Pi_1 = 0.25$, $\Pi_2 = 0.75$, $Q = 2$, $\eta = 0.001$). (a) Distance measure ε_{JS} versus training epoch. (b) Target and predicted values of the sine function.

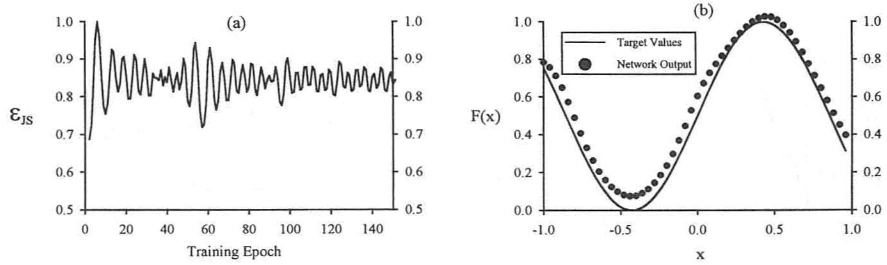

Figure 10: Network training and prediction of the target function with the JS error-metric ($\Pi_1 = 0.5$, $\Pi_2 = 0.5$, $Q = 10$, $\eta = 0.01$). (a) Distance measure ε_{JS} versus training epoch. (b) Target and predicted values of the sine function.

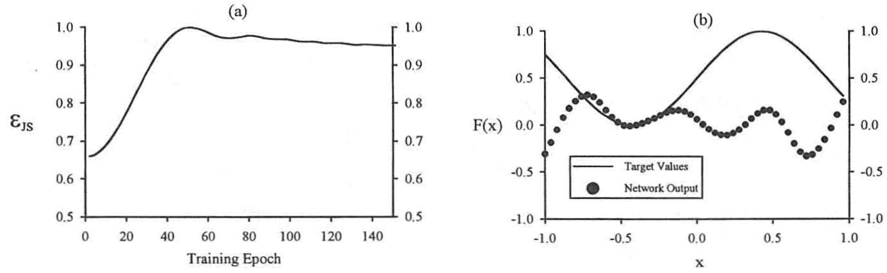

Figure 11: Network training and prediction of the target function with the JS error-metric ($\Pi_1 = 0.75$, $\Pi_2 = 0.25$, $Q = 10$, $\eta = 0.0001$). (a) Distance measure ε_{JS} versus training epoch. (b) Target and predicted values of the sine function.

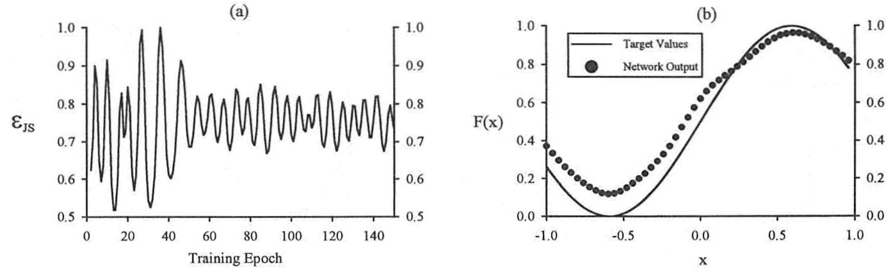

Figure 12: Network training and prediction of the target function with the JS error-metric ($\Pi_1 = 0.5$, $\Pi_2 = 0.5$, $Q = 2$, $\eta = 0.01$). (a) Distance measure ε_{JS} versus training epoch. (b) Target and predicted values of the sine function.

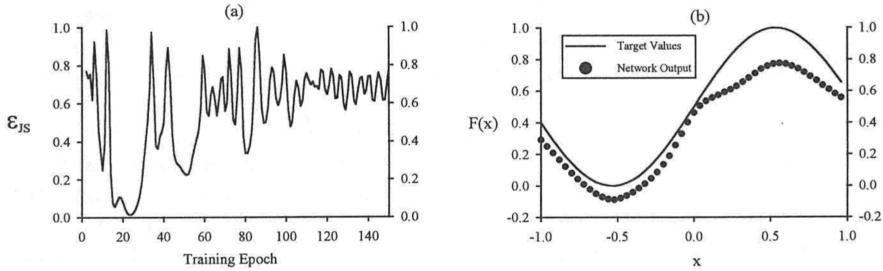

Figure 13: Network training and prediction of the target function with the JS error-metric ($\Pi_1 = 0.5$, $\Pi_2 = 0.5$, $Q = 1/2$, $\eta = 0.01$). (a) Distance measure ε_{JS} versus training epoch. (b) Target and predicted values of the sine function.

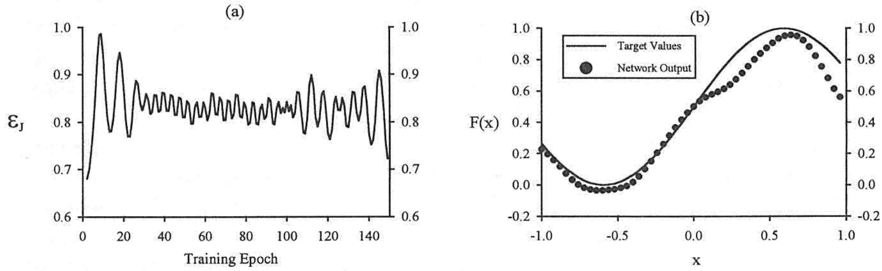

Figure 14: Network training and prediction of the target function with the J error-metric ($Q = 2$, $\eta = 0.001$). (a) Distance measure ε_J versus training epoch. (b) Target and predicted values of the sine function.

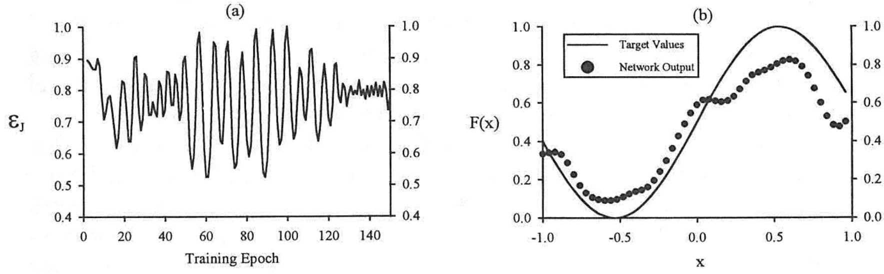

Figure 15: Network training and prediction of the target function with the J error-metric ($Q = 1/2$, $\eta = 0.001$). (a) Distance measure ε_J versus training epoch. (b) Target and predicted values of the sine function.

creasing the ultimate prediction accuracy of the network, Figures 12 through 15 are presented. They depict the predictions for the JS and J measures with values of η fixed (corresponding to the minimum rms error conditions of Table 1(c)) while decreasing the value of Q. Figures 12 and 13 are results for the symmetric JS measure with $\eta = 0.01$ and $Q = 2$ and $1/2$ respectively. In comparison to the result of Figure 10 (with $Q = 10$), it is clear that as Q decreases, the magnitude of the error-metric oscillations increases, thereby jeopordizing the stability of training optimization, as well as simultaneously decreasing the accuracy of target prediction. Figures 14 and 15 depict the results relevant to the J measure of Figure 8 with $\eta = 0.001$ and values of Q equal to 2 and $1/2$, respectively. Again it is observed that decreasing the value of Q increases the error-metric oscillations and degrades the accuracy of the functional prediction.

6. Conclusions

The present study indicates the feasibility of prescribing an error measure for the NN optimization algorithms in terms of the divergence associated with the statistical distributions of the network's output and the target values. Such a divergence measure specifies implicitly the conditional entropy pertinent to the statistics involved.

In specifying a divergence measure based on information-theoretic considerations, three error-metrics, namely the Kullback-Leibler (KL), the Jensen (J), and the Jensen-Shannon (JS) measures are defined and their relative efficacies are elucidated in reference to a multilayer perceptron trained via a backpropagation algorithm. Computer simulations indicate that not all such error-metrics (defined on the basis of the conditional entropy) are however, useful. Specifically, the KL measure poses convergence problems during training and exhibits significant deviation of the network end-results versus the target values. However, balanced (symmetric) measures defined by J- and/or JS-metrics offer both rapid convergence during training and network output deviations from the target that are comparable to those of conventional square-error and relative-entropy measures. But, before using the J and/or JS measures, the choice of the network parameter Q and the learning coefficient η have to be preevaluated such that the pair of the values (Q and η) correspond to the minimum rms output error of the network.

In the case of an asymmetric (unbalanced) error-metric such as the KL measure, the error deviations ($\varepsilon_{\mathrm{KL}}$) are one-sided as can be seen in Figure 7(a) along the training epochs. In symmetric (balanced) error-metrics (such as J-measure or JS measure with $\Pi_1 = \Pi_2 = 0.5$), the respective error deviations ε_{J} and $\varepsilon_{\mathrm{JS}}$ (in Figures 8 and 10) are symmetric about the converged value specified at the terminal epochs. Again, when the JS measure becomes asymmetric (with $\Pi_1 \neq \Pi_2$), the performance of the network deteriorates, as can be seen in Figures 11 and 12. Further, the criterion of using the pair of values of Q and η (which correspond to the minimum rms output error as in Table 1(c) is rather essential to obtain acceptable network performance, even if the symmetric error-metrics are adopted. A degradation in the performance prediction is imminent as can be observed from the results shown in Figures 13 through 15.

Also indicated in this paper is the significance of using a single parameter sigmoidal activation function (known as the Bernoulli function) in training the network. The relevant controlling parameter (designated as Q) has an influence on network performance towards convergence by suppressing the magnitude of the allowable neuronal state-transitions, thereby desensitizing the oscillations of the network output to increased learning rates.

To conclude, this paper presents a new strategy of NN optimization under the constraint of conditional entropy error-metrics. Apart from the three distance measures indicated here, there are also other measures such as Csiszer's family of directed divergence [5] which can be exploited in the NN vis-a-vis the conditional entropy error-metrics on the basis of similar considerations outlined in this work.

Appendix

Lemma 1.

$$\begin{aligned} &H_y = 0, \quad \textit{if all } |y_i - y_T| \le D(y_i) \textit{ or if} \\ &\;p_i = 0 \quad \textit{for all } |y_i - y_T| - D(y_i) > 0. \end{aligned} \tag{A.1}$$

Lemma 2.

$$\begin{aligned} &H_y \to 0 \quad \textit{for the ensemble } p_i > 0, \textit{ if } |y_i - y_T| - D(y_i) \to 0, \textit{ and} \\ &H_y \to \infty, \quad \textit{if } |y_i - y_T| - D(y_i) \to \infty. \end{aligned} \tag{A.2}$$

Lemma 3. *With* $p_i = 1/\kappa$,

$$H_y = -(1/\kappa)\sum_{i=1}^{\kappa} \log\{1/[|y_i - y_T| - D(y_T)]\} + e_\kappa \tag{A.3}$$

where

$$e_\kappa = (1/\kappa)\sum_{i=1}^{\kappa} \log\{[|y_i - y_T| - D(y)]/[|y_i - y_T| - D(y) + 1]\}$$

and

$$e_\kappa \to 0, \textit{ if } |y_i - y_T| - D(y) \gg 1.$$

Lemma 4. *Sum of two entropies satisfying the conditions of independence and summation in the spread space of the state vector leads to:*

$$\begin{aligned} H_{y(1,2)} = \log H_{y1}[&|y_1 - y_T| - D(y_i) \\ &+ H_{y2}[|y_2 - y_T| - D(y_i) + \varepsilon_{(1,2)} \end{aligned} \tag{A.4}$$

where

$e_{(1,2)} = \log\{[H_{y_1}(Y_1) + H_{y_2}(T_2)]/[H_{y_1}(Y_1) + H_{y_2}(Y_2) + 1]\} \to 0$ *for* $H_y(Y_i) \gg 1$ *with* $Y_i = |y_i - y_T|$, $(i = 1, 2)$.

Lemmas 1 and 2 represent the intuitive concept of neural disorganization in the state of being controlled towards the goal. If an ideal control is perceived, it strikes the well-ordered target domain in all realizations specified by $H_y = 0$. Diversions from the ideality of the ensemble with an increasing or decreasing trend enable H_y to increase or decrease respectively.

Lemma 3 stipulates that, in the event of equiprobable diversions, the relation between the spread space of the state vector and the entropy space is logarithmic with an err $e_\kappa \to 0$ for $|y_i| \ll 1$.

Entropy associated with target seeking is not additive. That is, goal-associated entropies cannot be added or subtracted directly in the entropy space. However, these superposition operations can be performed in the parameter spread space and the consequent effects can be translated to the entropy space.

Lemma 4 specifies the rule of additivity in the parameter spread space pertaining to independent goal-associated position entropies with an accuracy set by $e_{(1,2)} \to 0$ with $H_y(Y_{1,2}) \gg 1$.

References

[1] H. Hertz, A Krough, and R. Palmer, *Introduction to the Theory of Neural Networks*, Lecture Notes Volume I, Sante Fe Institute, Studies in the Science of Complexity (Addison Wesley, Reading, MA, 1991).

[2] P. S. Neelakanta and D. De Groff, *Neural Network Modeling: Statistical Mechanics and Cybernetic Perspectives*, (CRC Press, Boca Raton, FL, 1994).

[3] S. A. Solla, E. Levin, and M. Fleisher, "Accelerated Learning in a Layered Neural Network," *Complex Systems*, **2** (1988) 625–640.

[4] R. L. Watrous, "A Comparison between Squared Error and Relative Entropy Metrics Using Several Optimization Algorithms," *Complex Systems*, **6** (1992) 495–505.

[5] J. N. Kapur and H. K. Kesavan, *Entropy Optimization Principles with Applications*, (Academic Press/Harcourt Brace Jovanovich, Publishers, Boston, MA, 1992).

[6] T. M. Cover and J. A. Thomas, *Elements of Information Theory*, (John Wiley and Sons Inc., New York, 1991).

[7] J. Lin, "Divergence Measures Based on the Shannon Entropy," *IEEE Transactions on Theory*, **17** (1991) 145–151.

[8] P. S. Neelakanta, R. Sudhakar, and D. De Groff, "Langevin Machine: A Neural Network Based on Stochastically Justifiable Sigmoidal Function," *Biological Cybernetics*, **65** (1991) 331–338.