Symmetrization of Information-theoretic Error-measures Applied to Artificial Neural Network Training

Joseph C. Park[*]
2d3D Incorporated,
2003 North Swinton Avenue,
Delray Beach, FL 33444

Salahalddin T. Abusalah[†]
University of West Florida,
Department of Electrical and Computer Engineering,
Pensacola, FL 32514

The typical training scenario for an artificial neural network involves minimization of a cost-function in terms of the network output variables. Alternatively, the minimization may be done on the basis of the probability distributions of the network output, specifically, in terms of an informational entropy. A large class of such entropy based cost-functions are not suitable for training as they provide a directed divergence of mutual information between the network output and the desired behavior, that is, they are one-sided or asymmetric divergence functions. It is shown that a "symmetrization" of such divergence functions can transform them into suitable cost-functions for gradient-descent based optimizations. Nine such divergence measures are explicitly detailed and employed in training a multilayer perceptron to demonstrate their utility as pragmatic cost-functionals.

1. Introduction

Artificial neural networks (ANNs) have developed from precursory structures intended to mimic the gross operational features of biological neurons, into an important and useful class of computational architectures capable of performing many pragmatic functions not limited to pattern recognition and model-free nonlinear estimation. With respect to the latter class of computational tasks, feedforward ANNs have demonstrated a robust and computationally efficient architecture (once the ANN is trained) for realizing otherwise intractable numerical solutions. In a general sense, ANNs are simply implementations

[*]Electronic mail address: jpark@emi.net.
[†]Electronic mail address: sabusala@uwf.edu.

of computational algorithms that provide a functional mapping from an n-dimensional input to an m-dimensional output space. Typically, ANNs operate in two distinct phases.

1. The *training* phase, where the network adjusts its internal parameters in response to training or environmental data.

2. A *predictive* phase, where the trained network responds to input data and produces a functional mapping.

ANNs can be generally classified as either supervised, or unsupervised paradigms. The former class requires a "teacher" to produce an error output in response to training data, minimization of the error directs the learning process and adjustment of the network internal parameters. An unsupervised network does not rely on an external error source, but utilizes a rule base to adjust the internal parameters in response to the network output during training. ANNs can be further classified as either feedforward or feedback types. This refers to the flow of information during the predictive phase of operation.

The training process is a crucial step in producing a useful ANN for a problem at hand, as this is the stage wherein an ANN organizes its internal structure. This process of a directed organization based on an error minimization, when viewed from an information-theory perspective, is a process of entropy minimization. That is, the network adjusts itself from an essentially random, unusable state, through the minimization of an error (or information) measure, into an ordered structure capable of useful inference and generalization. This entropy of course is not the thermodynamic entropy, but Shannon's entropy. The emergence of Shannon's concept of an information measure [1] was arguably one of the most profound and useful ideas to emerge in the field of communications engineering, and has recently been applied towards precisely this problem of organizing ANNs into pragmatic computational machines [2–6]. Prior to this application of entropy minimization in ANN training, the fiducial process was to train the network based on some measure of error-difference between the desired and current state of the output solely in the output domain. That is, if the network output represented a thermodynamic temperature, then some form of temperature difference between the desired and actual output was used as the error-correction term.

An entropy-based network organization departs from this approach by training the network based on the minimization of a cross-entropy (the mutual information content) between the desired output distribution and the current distribution of the network output. It has been shown that training of a feedforward perceptron ANN in the information-theoretic domain results in a more robust training phase than training in the conventional output domain error-measures [4], in that convergence of the network training is desensitized to increases in

the learning rate. This is a welcome feature, as the "art" of parameter selection for learning rates can become a tiresome and repetitive chore in the training of conventional perceptron ANNs.

The authors have developed, applied, and evaluated a variety of such information-theoretic error-measures in order to assay their relative efficacy in the training of ANNs [6]. During this work, it was found that certain classes of information-theoretic error-measures were unsuitable for training purposes, these error measures always caused a network divergence, irrespective of the learning rates. Examination of these ill-conditioned measures revealed that they were not balanced in the information-theoretic sense, they directed the network organization based on a one-sided information flow between the desired and actual output.

To facilitate understanding of this concept, we can fix ideas based on the classic definition of Shannon's entropy which can be defined as:

$$H(x) = \sum_{n=1}^{N} p_n \log(p_n) \tag{1}$$

for a random variable with sample space $X = \{x_1, x_2, \ldots, x_N\}$ and associated probability measure $P(x_n) = p_n$. This represents the average information of the random variable x, or in the case that x_i represents an individual symbol from a sequence of symbols intended to convey information, the information per symbol. If one of the $p_i = 1$, then $H(x) = 0$, and there is no information conveyed, as there is no uncertainty about which symbol is transmitted. In the case that all $p_i = 1/N$, then $H(x) = \log N$, with an upper bounding value of $H(x)$, as the uncertainty about which symbol will be transmitted is maximum since they are all equally probable. It follows that the reception of a symbol will transfer maximum information, as there is maximal uncertainty about which symbol would be received. The Shannon entropy then describes the uncertainty associated with the outcome of a stochastic experiment and maximization of this entropy, subject to the applied constraints, provides a method for determination of probability distributions.

In the case that two independent probability distributions are involved, denoted as p and q, an information measure is proposed in [7] based on the Shannon entropy that would quantify the average information for discrimination between the two distributions:

$$D_{\mathrm{KL}}(p : q) = \sum_i p_i \log\left(\frac{p_i}{q_i}\right). \tag{2}$$

When p and q are equivalent distributions, then $D = 0$, there is no information contained in p that q does not have. Alternatively, the "distance" or difference between the two distributions is zero. As the two distributions diverge, then the corresponding value of D increases.

However, this D represents a *directed divergence* of the probability p *from* q, and not *vice versa*. This is a consequence of the fact that in general $D(p : q) \neq D(q : p)$. That is, the measure is one-directional or one-sided.

The basic tenet of applying information-theoretic divergence measures as a means of error-feedback for training ANNs is to ascribe the desired output probabilities associated with the target values T_j to the distribution q, and output probabilities arising from the actual network output O_j to the distribution p. The divergence D is used at each training iteration in place of the conventional error measure based on $f(T_j - O_j)$. However, when a one-sided error-measure such as D_{KL} is employed the network will quickly converge to a state that is not necessarily representative of the desired behavior. It has been shown that symmetric, two-sided information error-measures have demonstrated a pragmatic utility in allowing for a wider range of learning rates to be tolerated [4]. Therefore, it is useful to effect a "symmetrization" of the available one-sided error-metrics so that they may be investigated for enhanced network optimization characteristics and profitably employed in ANN learning. The focus of this paper is the explicit symmetrization of a set of one-sided information-theoretic error-measures which otherwise would be useless as tools for gradient-descent learning in ANNs. Before proceeding to elaborate the individual error measures and their symmetrization, it is appropriate to detail the mechanics of applying an information-theoretic error measure in the learning algorithm of an ANN.

2. Application of information-theoretic error-measures to artificial neural network training

Consider a general feedforward ANN as depicted in Figure 1. The ANN consists of an agglomeration of individual information processing units, referred to as neurons, arranged in successive layers with complete cross-neuron interconnections between adjacent layers. The interconnections are numerical weights, denoted as w_{ij}, between the ith and jth neurons. This weight is multiplied by the output of the ith neuron, and is then presented as one of the multiinputs to the jth unit. Each weight is modified during the training process to produce a minimum error output from the network. The input layer receives the network stimulus and serves as a multiplexer to the first "hidden layer" of neurons. Successive neural layers propagate the incrementally processed stimuli until the network output layer is reached. Each neuron is a nonlinear processor which takes the weighted sum of the multiinputs x_j: $X_i = \sum w_{ij} x_j$ and then processes this value by a (typically sigmoidal) activation function $F_S(X_i)$ to produce the neuron output signal O_i.

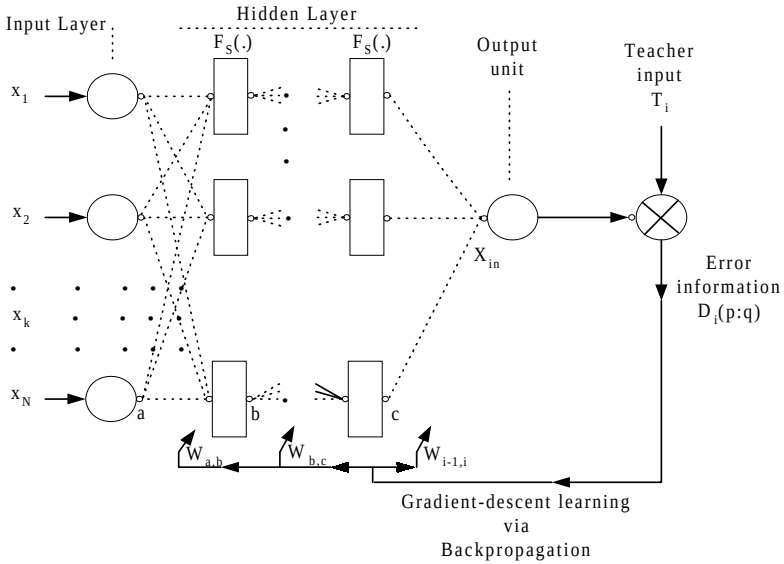

Figure 1. Schematic of the multilayer perceptron ANN.

The ANN will be trained via the "backpropagation" [8] gradient-descent algorithm. However, a departure from the conventional parametric output-domain error gradient is that we employ the cross-entropy between the actual neuron output and the specified training value as the representation of the divergence of the actual output from the desired one. This requires delineation of the probability distributions associated with both the actual output O_i and the desired output T_i. In the general case of arbitrary training sets which may not be conveniently expressable in closed-form, the training distributions $q_i(T(X_i))$ may be obtained from explicit relative-frequency distributions extracted from the observable training set over the network output range. Likewise, the distribution of the actual output $p_i(O(X_i))$ can be ascertained from a relative-frequency distribution evaluated at each discrete training cycle over the output range of the neurons. Alternatively, in cases where the training sets are expressable in a closed-form, it is possible to analytically determine the associated distributions as detailed in [4].

Concerning the backpropagation algorithm, the fundamental entity used in the weight adjustment process is the error ϵ_i of the network output O_i at the ith unit, with respect to the target value T_i. This error is used to calculate the effective gradient δ_j of the weight modification term. In order to effect gradient-descent backpropagation in the information-theoretic plane, the classical parametric euclidian error: $\epsilon = (T_i - O_i)$, is replaced with the information-theoretic cross-entropy $\epsilon = D(p : q)$ between the actual and target distributions. 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 error of the jth unit times the derivative of the activation function evaluated at the output value of the ith unit. That is, $\delta_j = (\partial O_i/\partial X_i)\epsilon_j$ where X_i represents the ith unit input to the activation function. When the unit resides in a hidden or input layer, a target value is not available for computation of the network error ϵ. Therefore, the definition is modified such that the product of cumulative effective gradients from the next layer and the interconnection weights are backpropagated to these units. In other words:

$$\delta_i = \frac{\partial O_i}{\partial X_i} \sum_j \delta_j w_{ij}.$$

In the case of the conventional euclidean metric, the sign of δ is determined by the simple arithmetic difference between the target and output, so that the direction of the gradient-descent is controlled by feedback from the comparison of the target *versus* output difference. However, the cross-entropy metrics involving logarithmic functions are strictly nonnegative, and therefore would not allow for δ to change its sign in response to the target *versus* output differences changing sign, thereby resulting in a loss of feedback control in the weight change algorithm. To remedy this situation, the calculation of the effective gradient with the cross-entropy error-metrics is multiplied by ± 1, depending on the sign of the target-output difference. That is, the value is specified by

$$\delta_i = \delta_i \,\mathrm{signum}(T_i - O_i).$$

With the appropriate expression for the error gradient in hand, the basic prescription for adjustment of weights at the nth training step is given by the well known Widrow–Hoff delta rule [8]:

$$w_{ij}(n) = w_{ij}(n-1) + \eta\delta_j O_i = w_{ij}(n-1) + \Delta w_{ij}(n)$$

where η is the learning rate.

In regions of the error surface where large gradients exist, the δ terms may become inordinately large. The resulting weight modifications will also be large, leading to extensive oscillations of the network output, preventing convergence to the true error minimum. The learning coefficient could be set to an extremely small value to counteract this tendency; however, this would drastically increase the training time. To avoid this situation, 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 specified by:

$$\Delta w_{ij}(n) = \eta\delta_j O_i + \lambda[\Delta w_{ij}(n-1)]$$

where λ is the momentum parameter. If λ is set to a value close to 1, the search in error space will be determined by the gradient accumulated

over several epochs instead of a single iteration, improving the stability of the network.

2.1 Artificial neural network implementation details

Specific to the ANN utilized in this work, the network architecture consists of 30 neurons in the input layer, two hidden layers comprised of 15 neurons each, and a single output layer neuron. The activation functions used in the hidden layers are sigmoidal Bernoulli functions $L_Q(x)$ [9], with $Q = 1/2$, and a linear function in the output layer. The linear output activation functions allow the network output to converge to values outside the ± 1 interval set by the Bernoulli function bounds. The input and hidden layers also have an additional bias unit clamped to a fixed output of -1, connected to each unit in the succeeding layer through a trainable weight.

The 30 input units are trained at equally-spaced x values over the interval $x_j \in [0, \pi]$, where each input unit corresponds to a single x_j, while the output unit is evaluated at an x value of $x_i = 0.3$. The weights and thresholds are initialized to uniformly distributed pseudorandom values over $[-1, 1]$. A learning rate coefficient of 0.001 is used unless otherwise specified, along with a momentum value of $\lambda = 0.9$.

The network is sequentially presented with 100 sinc wave training sets over the 30 points x_j during each training epoch. The training sets are specified by $T_\gamma = |\sin(m\pi\gamma x)/m\pi\gamma x|$, where γ is a uniformly distributed pseudorandom variate in the range $[-1, 1]$ and m is the frequency control integer. The output of the network at x_i is used in the backpropagation mode with the gradient-descent to adjust the weights for 500 training epochs. After the training, the network is set to compute the values of the sinc function (with $\gamma = 1$) at 50 equally spaced points x_n over the interval $[0, \pi]$.

3. Symmetrization of one-sided error-measures

A wide range of candidate error-metrics exists in the information-theoretic domain for application to ANN optimization [6]. Many of these measures have been profitably applied to constrained optimization problems [10], such as the derivation of statistical mechanics distributions, resulting in equivalent realizations with substantially reduced effort. Here we concentrate on several such measures that were adapted to the training of ANNs, yet proved ill-conditioned for successful ANN convergence due to their inherently one-sided nature, and show that it is possible to symmetrize each of these unstable error-metrics into a form that allows for robust network training.

First, we introduce a nomenclature to facilitate description of the various error measures. It was pointed out in [10] that information-theoretic

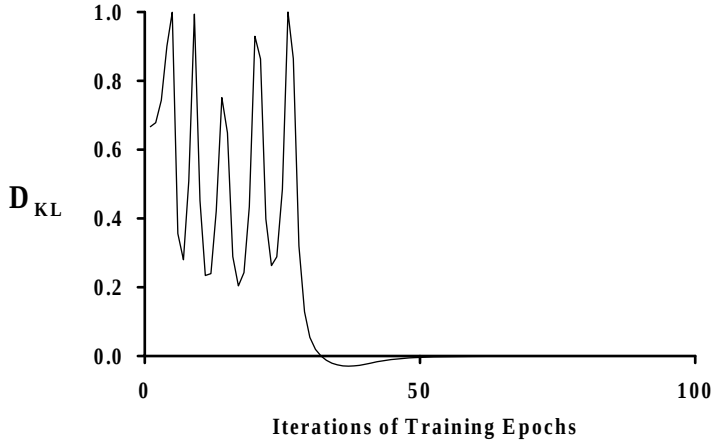

Figure 2. ANN training evolution exhibiting one-sided divergence of the Kullback–Leibler error measure.

divergence measures D can take the form of an arbitrary functional, as long as the function meets the following criteria.

1. D is continuous in p and q.
2. $D(p, q) \geq 0$.
3. $D(p_i : p_j) + D(p_j : p_k) \geq D(p_i : p_k)$.
4. $D(p : q)$ is a convex function of p and q.

Therefore, a generalized functional Φ can be used to provide a convenient framework for discussing the various error measures and their symmetrization. The canonical form for a directed divergence measure may be written as:

$$D(p : q) = q \cdot \Phi(p : q). \tag{3}$$

For example, the Kullback–Leibler error measure of equation (2) arises if $\Phi(x) = x \ln(x)$, with $x = p/q$. However, as ascribed earlier, this constitutes a one-sided error measure unsuitable for convergence of neural network error spaces. To illustrate this ill-conditioned behavior, the feedforward neural network described in section 2 was used to learn the nonlinear function $|\sin(x)/x|$ with the Kullback–Leibler divergence of equation (2). The resulting temporal evolution of the error is shown in Figure 2, with the error normalized to a maximum value of 1.

It is observed that the ANN initially oscillates in error-space about the $D_{\mathrm{KL}} = 0.5$ value, then migrates to a negative terminal divergence value approaching zero. Subsequent to this point, the error gradient never again oscillated. This behavior indicates that the gradient-descent

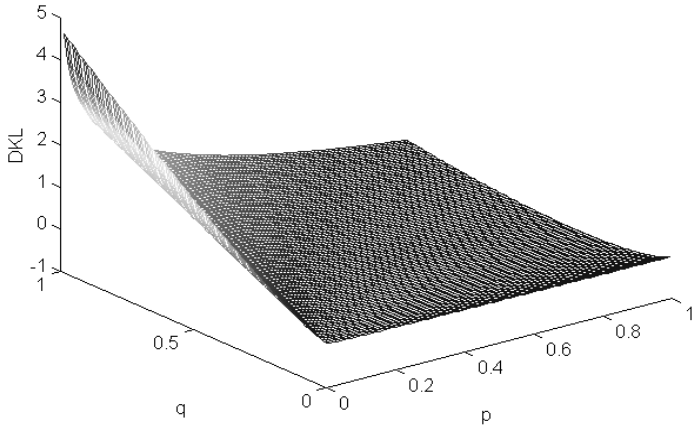

Figure 3. Kullback–Leibler error measure, exhibiting the unbalanced nature of the information for discrimination with respect to the divergence of distribution p from q.

has found a smooth (locally flat) region in the information divergence space which has reduced the mutual information between the teacher and predicted distributions to zero. That is, the network is perfectly trained and the information-theoretic entropy of the network is zero. This is however an erroneous interpretation which loses sight of the fact that the divergence measure is strictly nonnegative. The network has not in fact converged, but has evolved into an unbalanced state where the information for discrimination as a measure of divergence between the teacher and network predictions has lost meaning. In order to visualize the one-sided and unbalanced nature of such a divergence function, the D_{KL} measure is depicted in Figure 3.

Therefore, it is sensible to balance the information discrimination symmetrically between p and q to avoid the one-sided behavior. This is achieved in the canonical sense *via*

$$D(p : q) = q \cdot \Phi(p : q) + p \cdot \Phi(q : p) \tag{4}$$

so that the symmetrized form of the Kullback–Leibler divergence (denoted by the trailing S subscript and enumerated as Measure I) becomes:

$$D_{\mathrm{KLS}}(p : q) = \sum_i p_i \log\left(\frac{p_i}{q_i}\right) + \sum_i q_i \log\left(\frac{q_i}{p_i}\right). \tag{5}$$

A plot of the symmetrized Kullback–Leibler divergence is shown in Figure 4, and it clearly demonstrates the balanced nature of information for discrimination between the distributions.

The symmetrized version was employed with the exact same weight initializations, and network parameters in the ANN to learn the same

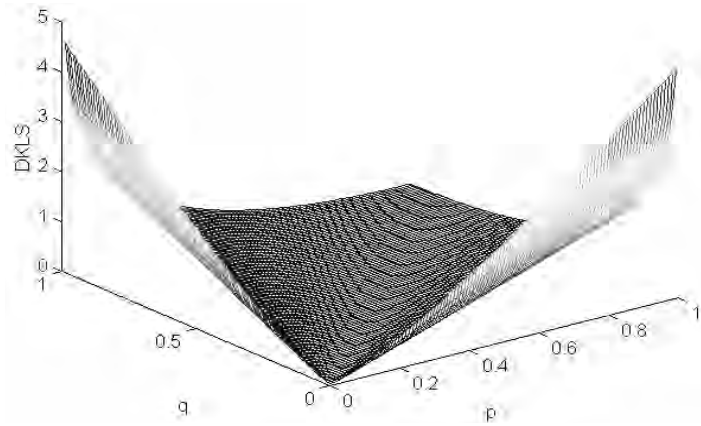

Figure 4. Symmetrized Kullback–Leibler error measure (h), demonstrating the balanced nature of the information for discrimination with respect to the divergence of distribution p from q.

nonlinear function as in the unsymmetrized case. The resulting network outputs are shown in Figure 5.

Figure 5(a) shows that the symmetrized error measure allows the network to converge, and that the error values are symmetrically distributed about the converged value. The network prediction of the learned function is depicted in Figure 5(b), verifying that the network converged to a reasonable approximation of the sinc function. One could argue that predicted values are practically useless for a mathematical representation, however, it is not the aim of this paper to fine-tune multilayer perceptron performance to particular nonlinear functional representations. Rather, we are concerned with the ability to convert divergent information-based error metrics into usable gradient-search directives.

In addition to the Kullback–Leibler measure, several one-sided divergence measures were found which could be symmetrized into useful ANN error measures. Each of these measures are enumerated below in both the generalized functional Φ form, the explicit discrete probability measure form $D(p : q)$, and their symmetrized form $D_S(p : q)$.

Measure II. Havrda and Charvát [11] proposed a "structural α-entropy" which incorporates an order parameter α. As $\alpha \to 1$, this function reduces to Shannon's definition of entropy or information content:

$$\Phi(x) = \frac{(p/q)^{\alpha} - (p/q)}{\alpha - 1}; \qquad \alpha > 0 \text{ and } \alpha \neq 1 \tag{6}$$

$$D_{\mathrm{HC}} = \frac{1}{\alpha - 1} \sum_i q_i [p_i^{\alpha} q_i^{(1-\alpha)} - p_i] \tag{6a}$$

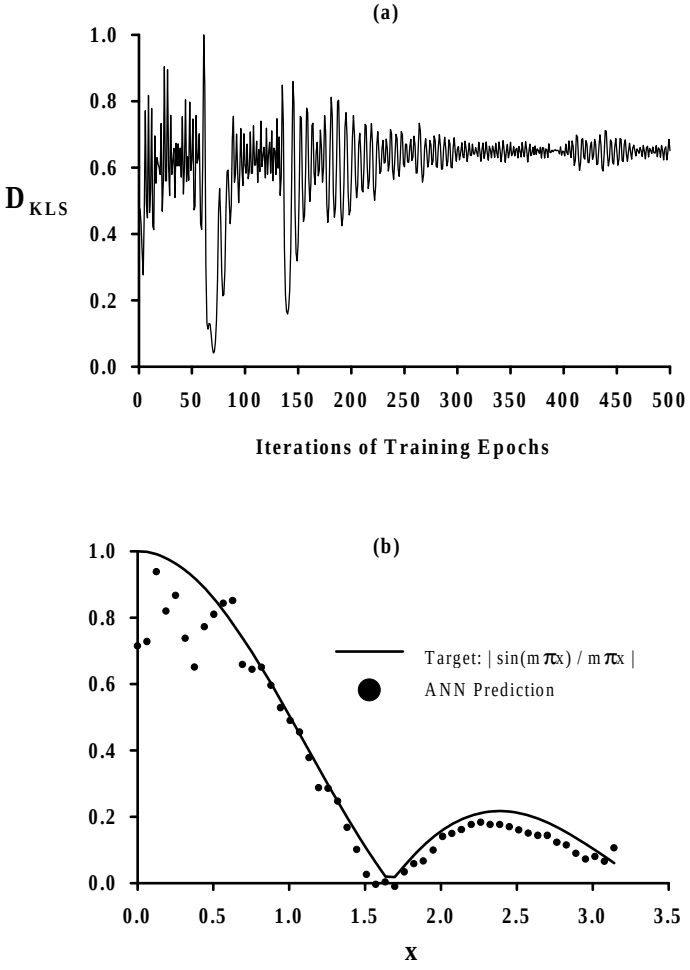

Figure 5. ANN training evolution and functional prediction with the symmetrized Kullback–Leibler error measure.

which in symmetrized form is:

$$D_{\mathrm{HCS}} = \frac{1}{\alpha - 1}\left\{\left(\sum_i p_i^{\alpha} q_i^{(1-\alpha)} - p_i\right) + \left(\sum_i q_i^{\alpha} p_i^{(1-\alpha)} - q_i\right)\right\}. \qquad (6b)$$

Measure III. Sharma and Mittal [12] developed a power-law functional applicable to sum generalizations as:

$$\Phi(x) = \frac{(p/q)^{\alpha} - (p/q)^{\beta}}{\alpha - \beta}; \qquad \begin{array}{l} \alpha > 1, \beta \le 1; \text{ or} \\ \alpha < 1, \beta \ge 1, \text{ with } \alpha \text{ and } \beta > 0 \end{array} \qquad (7)$$

$$D_{\mathrm{SM}} = \frac{1}{\alpha-\beta}\left(\sum p_i^{\alpha} q_i^{(1-\alpha)} - \sum p_i^{\beta} q_i^{(1-\beta_i)}\right) \tag{7a}$$

$$D_{\mathrm{SMS}} = \frac{1}{\alpha-\beta}\left\{\left(\sum_i p_i^{\alpha} q_i^{(1-\alpha)} - \sum_i p_i^{\beta} q_i^{(1-\beta)}\right) + \left(\sum_i q_i^{\alpha} p_i^{(1-\alpha)} - \sum_i q_i^{\beta} p_i^{(1-\beta)}\right)\right\}. \tag{7b}$$

Measures IV through IX were proposed by Kapur and Kesevan [10] as generalized entropy functionals based on a class of information measures proposed by Csiszár [13], and are known collectively as forms of Csiszár's generalized measures.

Measure IV. Generalized Csiszar Type 1:

$$\Phi = \left(\frac{p+\alpha}{q}\right)\log\left(\frac{p+\alpha}{q+\alpha}\right); \quad \alpha > 0 \tag{8}$$

$$D_{\mathrm{CZ1}} = \sum (p_i+\alpha)\log\left(\frac{p_i+\alpha}{q_i+\alpha}\right) \tag{8a}$$

$$D_{\mathrm{CZ1S}} = \sum (p_i+\alpha)\log\left(\frac{p_i+\alpha}{q_i+\alpha}\right) + \sum (q_i+\alpha)\log\left(\frac{q_i+\alpha}{p_i+\alpha}\right). \tag{8b}$$

Measure V. Generalized Csiszár Type 2:

$$\Phi = \left(\frac{1+\alpha p}{q}\right)\log\left(\frac{1+\alpha p}{1+\alpha q}\right); \quad \alpha > 0 \tag{9}$$

$$D_{\mathrm{CZ2}} = \sum (1+\alpha p_i)\log\left(\frac{1+\alpha p_i}{1+\alpha q_i}\right) \tag{9a}$$

$$D_{\mathrm{CZ2S}} = \sum (1+\alpha p_i)\log\left(\frac{1+\alpha p_i}{1+\alpha q_i}\right) + \sum (1+\alpha q_i)\log\left(\frac{1+\alpha q_i}{1+\alpha p_i}\right). \tag{9b}$$

Measure VI. Generalized Csiszár Type 3:

$$\Phi(x) = \frac{1}{\alpha}\left[\alpha\left(\frac{p}{q}\right)+\beta\right]\log\left[\alpha\left(\frac{p}{q}\right)+\beta\right] - \frac{p}{q} - \frac{1}{\alpha}(\alpha+\beta)\log(\alpha+\beta) + 1; \quad \alpha > 0 \tag{10}$$

$$D_{\mathrm{CZ3}} = \frac{1}{\alpha}\sum (\alpha p_i + \beta q_i)\log\left[\frac{\alpha p_i + \beta q_i}{q_i}\right] - \frac{1}{\alpha}(\alpha+\beta)\log(\alpha+\beta) \tag{10a}$$

$$D_{\mathrm{CZ3S}} = \frac{1}{\alpha}\left\{\sum (\alpha p_i + \beta q_i)\log\left(\frac{\alpha p_i + \beta q_i}{q_i}\right) + \sum (\alpha q_i + \beta p_i)\log\left(\frac{\alpha q_i + \beta p_i}{p_i}\right) - \frac{2}{\alpha}(\alpha+\beta)\log(\alpha+\beta)\right\}. \tag{10b}$$

Measure VII. Generalized Csiszár Type 4:

$$\Phi\left(\frac{p}{q}\right) = \frac{p}{q}\log\left(\frac{(p/q)}{\mu(p/q)+(1-\alpha)}\right); \quad 0 \le \alpha \le 1 \tag{11}$$

$$D_{\mathrm{CZ4}} = \sum p_i \log\left(\frac{p_i}{\alpha p_i + (1-\alpha)q_i}\right) \tag{11a}$$

$$D_{\mathrm{CZS4}} = \sum p_i \log\left(\frac{p_i}{\alpha p_i + (1-\alpha)q_i}\right) + \sum q_i \log\left(\frac{q_i}{\alpha q_i + (1-\alpha)p_i}\right). \tag{11b}$$

Measure VIII. Generalized Csiszár Type 5:

$$\Phi\left(\frac{p}{q}\right) = \frac{p}{q} - \ln\left(\frac{p}{q}\right) - 1 \tag{12}$$

$$D_{\mathrm{CZ5}} = q\left[\frac{p}{q} - \log\left(\frac{p}{q}\right) - 1\right] \tag{12a}$$

$$D_{\mathrm{CZS5}} = q\left[\frac{p}{q} - \log\left(\frac{p}{q}\right) - 1\right] + p\left[\frac{q}{p} - \log\left(\frac{q}{p}\right) - 1\right]. \tag{12b}$$

Measure IX. Generalized Csiszár Type 6:

$$\Phi\left(\frac{p}{q}\right) = \frac{(p/q)^\alpha - (p/q)}{\alpha(\alpha-1)}; \quad \alpha \ne 0, 1 \tag{13}$$

$$D_{\mathrm{CZ6}} = \frac{1}{\alpha(\alpha-1)}\left[\left(\frac{p}{q}\right)^\alpha - \left(\frac{p}{q}\right)\right] q \tag{13a}$$

$$D_{\mathrm{CZS6}} = \frac{1}{\alpha(\alpha-1)}\left[\left(\frac{p}{q}\right)^\alpha - \left(\frac{p}{q}\right)\right] q + \frac{1}{\alpha(\alpha-1)}\left[\left(\frac{q}{p}\right)^\alpha - \left(\frac{q}{p}\right)\right] p. \tag{13b}$$

4. Evaluations *via* the multilayer perceptron

In order to verify the utility of the symmetrized divergence measures, they are employed to direct the backpropagation gradient-descent training of the multilayer perceptron in learning the teacher function $|\mathrm{sinc}(x)|$. Each ANN realization is then used to predict the learned function over 50 equally-spaced points in the interval $[0, \pi]$. A measure of the prediction veracity, and an indication that the network has converged to a stable solution, is to assess the root mean-square (RMS) error of the prediction from that of the teacher function over the 50 points. For example, the ANN prediction using the unsymmetrized Kullback–Liebler measure (corresponding to the temporal divergence evolution of Figure 2) produces an RMS error of 5.933. Therefore, the average magnitude of the error at any of the 50 prediction points was near 6, confirming that the network did not converge to a pragmatically stable solution. When the ANN is trained with the symmetrized Kullback–Liebler measure, the resulting RMS error is 0.087, verifying that the gradient-search sucessfully converged to a useful stable mean.

Divergence Number	Divergence Function	RMS Error	Divergence Parameters
	D_{KL}	5.9325	$\alpha = 0.5$
I	D_{KLS}	0.0874	
II	D_{HCS}	0.0988	$\alpha = 0.5$
III	D_{SMS}	0.2193	$\alpha = 0.5, \beta = 1.5$
IV	D_{CZS1}	0.1984	$\alpha = 0.5$
V	D_{CZS2}	0.0213	$\alpha = 0.5$
VI	D_{CZS3}	0.1840	$\alpha = 0.5, \beta = 0.5$
VII	D_{CZS4}	0.2215	$\alpha = 0.5$
VIII	D_{CZS5}	0.1191	
IX	D_{CZS6}	0.0955	$\alpha = 0.5$

Table 1. RMS error of the ANN prediction of the |sinc| function over 50 equally spaced points in the interval $[0, \pi]$.

Table 1 lists the results of training the ANN with each of the symmetrized measures. Prior to symmetrization, none of the divergence measures were capable of successfully training the ANN to a useful organization. The results of Table 1 clearly demonstrate that symmetrization of the one-sided information measures transform the unusable divergence forms into usable information domain cost-functions applicable to gradient-descent learning in ANNs. It is again emphasized that the results are not intended to establish the accuracy of a multilayer perceptron as a nonlinear functional estimator. Certainly, predictions with a smaller RMS error are possible with an investment in selection of network parameters and architecture. Rather, the results are intended to verify the success of the network organization under the direction of the symmetrized information-theoretic cost functions.

5. Conclusions

The use of information-theoretic cost functions based on Shannon's concept of information content has emerged as a powerful tool in the derivation and analysis of probability distributions arising from constrained optimization problems. The recent application of such information-theoretic divergence measures to the goal-directed organization of ANNs has demonstrated an increased tolerance to accelerated learning rates in comparison to the usual network output error minimization, and demonstrated the utility of a gradient-descent optimization in the cross-entropy (mutual information) domain. The cross-entropy between two distributions may be viewed as a measure of divergence between the distributions, and can therefore be used as the cost-function for minimization. In general, such cross-entropy func-

tions are not symmetric, that is $D(p : q) \neq D(q : p)$, as can be observed from the generalized divergence $D(p : q) = q\Phi(p : q)$. Such divergence measures are inherently directed-divergences, and so do not symmetrically balance the flow of information between the stochastic state of an ANN during training and the desired goal. This limitation renders such unsymmetric divergences unsuitable as cost-functions for ANN optimization.

The work described in this paper details a procedure for symmetrization of the basic one-sided algorithms of the Kullback–Leibler and/or Csiszár family of cross-entropy measures so as to make them useful for neural network applications. The basic procedure is to balance the divergence symmetrically with mutual-information between both the p and q, as well as the q and p as: $D(p : q) = q\Phi(p : q) + p\Phi(q : p)$. With this modification the divergence changes from a one-sided function with a locally flat region (in the p, q space) encompassing one of the p or q domains, into a symmetrically balanced function with global minimums along the desired $p = q$ boundary. This boundary delineates the $D = 0$ region, analogous to the $T_i - O_i = 0$ regions in conventional output variable error-space. The smoothly rising amplitude on either side of the $p = q$ boundary ensure that the global minimums are accessible to the gradient-descent algorithm as stable convergence points. Whereas for the unsymmetric functions, the locally flat $D = 0$ regions extend over a large area with $p \neq q$, allowing for network convergence into a region far removed from a pragmatic network organization with $D = 0$.

References

[1] Shannon, C. E., "A Mathematical Theory of Communication," *Bell System Technical Journal*, **27** (1948) 623–659.

[2] Solla, S. A., *et al.*, "Accelerated Learning in a Layered Neural Network," *Complex Systems*, **2** (1988) 625–640.

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

[4] Park, J. C., *et al.*, "Information-theoretic Based Error-metrics for Gradient Descent Learning in Neural Networks," *Complex Systems*, **9** (1995) 287–304.

[5] Neelakanta, P. S., *et al.*, "Dynamic Properties of Neural Learning in the Information-theoretic Plane," *Complex Systems*, **9** (1995) 349–374.

[6] Neelakanta, P. S., *et al.*, "Csiszár's Generalized Error Measures for Gradient-descent-based Optimizations in Neural Networks Using the Backpropagation Algorithm," *Connection Science*, **8**(1) (1996) 79–114.

[7] Kullback, S. and Leibler, R. A., "On Information and Sufficiency," *Annals of Mathematical Statistics*, **22** (1951) 79–86.

[8] Wasserman, P. D., *Neural Computing* (Van Nostrand Reinhold, New York, 1989).

[9] Neelakanta, P. S., *et al.*, "Langevin Machine: A Neural Network Based on a Stochastically Justifiable Sigmoidal Function," *Biological Cybernetics*, **65** (1991) 331–338.

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

[11] Havrda, J. and Charvát, F., "Quantification Method of Classification Processes," *Kybernetika*, **3** (1967) 30–35.

[12] Sharma, B. D. and Mittal, D. P., "New Non-additive Measures of Entropy for Discrete Probability Distributions," *Journal of Mathematical Sciences*, **10** (1975) 28–40.

[13] Csiszár, I., "A Class of Measures of Informativity of Observation Channels," *Periodica Mathematica Hungarica*, **2** (1972) 191–213.