On the PDF Estimation for Information Theoretic Learning for Neural Networks

Tokunbo Ogunfunmi and Manas Deb
Signal Processing Research Lab (SPRL), Department of Electrical Engineering,

Santa Clara University, Santa Clara, CA 95053, USA
togunfunmi@scu.edu, mdeb@scu.edu

Abstract **Information-Theoretic Learning (ITL) is one of the new methods gaining popularity used for adaptive signal processing learning algorithms and has many advantages compared to traditional method which minimizes the mean square error (MSE). Previously [12], we described a method based on the backpropagation algorithm to train a type of neural network called the multi-layer perceptron (MLP) using Information Theoretic Learning (ITL) techniques. Our method was developed to train MLPs by utilizing the minimum error entropy (MEE) of the error samples. The MSE is a second order statistic whereas the MEE uses the probability density function of the error samples. Therefore, the MEE technique uses higher order statistical information from the error samples to adapt the weights of the neural network. When the error distribution is non-gaussian, higher order statistical information can lead to faster training and smaller residual training error. The Probability Density Function (PDF) estimation using the Parzen window could affect the accuracy of the Back-Propagation training.**

In this paper, we investigate the effects of the Parzen Window estimator on the efficacy of the ITL training using Renyi's Entropy and Shannon's Entropy. Using different estimators and simulations, we compare MLP using the typical backpropagation algorithm (using MSE and cross-entropy) and also one using ITL methods in terms of convergence speed of the weights, PDF estimator and the residual error. We use standard data sets (like the MNIST handwriting data set available on the Internet) to train and test the MLP using all these methods. Simulation results compare the prediction accuracy of the three different types of backpropagation algorithms (MSE, Shannon's cross-entropy, Renyi's quadratic entropy) in the paper.

Keywords— Information Theoretic Learning (ITL); multi-layer perceptorn (MLP); neural network; backpropagation; kernel function; minimum error entropy (MEE), Rényi's entropy; Parzen window estimator.

I. INTRODUCTION

Traditional adaptive systems use second order statistics (such as variance, correlation etc.) of the input signal to adapt a set of weights based on a cost function. The cost functions in traditional adaptive systems use a second order statistical metric such as the mean square error (MSE) to derive the adaptive system. The second order statistics are sufficient if the input signal and the error signal are Gaussian. However, there are a number of challenging signal processing problems such as blind separation of sources and blind deconvolution of linear channels, where second order statistics are not sufficient. Also, in applications like machine learning, the data does not have a Gaussian distribution and the adaptive system may be non-linear. Such applications require higher order statistics of the input data signal.

A new approach to adaptive signal processing is where second order statistics such as variance, correlation and mean square error are respectively replaced by Information Theoretic metrics such as entropy, correntropy and minimum error entropy. By using these Information Theoretic learning metrics, the adaptive system can use higher order statistics of the data to adapt its weight vector. Information Theoretic Learning can be used in both linear and non-linear adaptive signal processing systems and also in supervised and unsupervised machine learning applications

A multilayer perceptron (MLP) is a type of feedforward artificial neural network that consists of an input layer, one or more hidden layers and an output layer. Each layer consists of multiple artificial neurons laid out like a directed acyclic graph where neurons from one layer connect to one or more neurons in the next layer in a feedforward manner. MLPs use supervised learning to learn a non-linear approximation function that can be used for classification or regression. For example, for classification problems, the features of the input data are transformed by the non-linear activation function of each neuron in each layer into a space where they can be linearly separated and identified as a specific class.

The backpropagation algorithm is used to train MLPs [1]. This is a supervised learning algorithm where the weights of the neurons are initially set to random values and the training data vector is input to the neural network to generate the output vector. The error vector is generated from the difference of the target vector and the output vector and an error gradient is computed based on the mean square error (MSE) criterion. The weights of the neurons in each layer are iteratively modified along the direction of the gradient that minimizes the MSE.

Information Theoretic Learning (ITL) comprises of a set of adaptive signal processing algorithms where the cost function is based on entropy and divergence of the error sample distribution. Instead of using a second-order statistic like the MSE to adapt the weights of a linear or non-linear system, ITL algorithms use the minimum error entropy (MEE) as the metric to converge to the optimal solution. Since the computation of the entropy involves the probability density function of the error samples, ITL techniques extract higher order statistics from these samples during system adaptation. Entropy is a measure of the uncertainty of a random variable. ITL algorithms train neural

networks by minimizing the entropy of the error samples. Instead of Shannon's entropy, which is the popular information measuring metric for data communication applications, ITL techniques employ Rényi's entropy which results in simple numerical algorithms to compute the entropy.

We investigate the effects of the parameters of the Parzen Window estimator on the efficacy of the ITL training using Renyi's Entropy and Shannon's Entropy. We compare MLP using the typical backpropagation algorithm (using MSE and cross-entropy) and also one using ITL methods in terms of convergence speed of the weights, PDF estimator and the residual error. The application was for standard data sets (like the MNIST handwriting data set available on the Internet) to train and test the MLP using all these methods. Simulation results compare the Information Potential and PDF using Parzen window estimator with and without sigma update.

The paper is organized as follows: In Section II, we review the Information Theoretic Learning concepts and in Section III, we present the Backpropagation algorithm for the ITL. Then in Section IV, we present results of our exploration of the choice of the parameters for the Parzen Window estimator using simulation. We discuss our results in Section V and present Conclusions in Section VI.

II. INFORMATION THEORETIC LEARNING CONCEPTS

The genesis of ITL is Alfréd Rényi's pioneering work on generalized measures of entropy and information [2]. At the core of Rényi's work is the concept of generalized mean or the Kolmogorov-Nagumo (K-N) mean [3][4]. For numbers $x_1, x_2, \quad x_N$, the K-N mean is expressed as:

$$\psi^{-1}\left(\frac{1}{N}\sum_{i=1}^{N}\psi\left(x_i\right)\right) \tag{1}$$

where $\psi(\ )$ is the K-N function which is continuous and strictly monotonic. In the general theory of means, the quasi-linear mean of a random variable X which takes the values $x_1, x_2, \quad x_N$ with probabilities $p_1, p_2, \quad p_N$ is defined as:

$$E_\psi\left[X\right] = \left\langle X\right\rangle_\psi = \psi^{-1}\left(\sum_{k=1}^{N} p_k \psi\left(x_k\right)\right) \tag{2}$$

From the theorem on additivity of quasi-linear means [5], if $\psi(\ )$ is a K-N function and c is a real constant, then:

$$\psi^{-1}\left(\sum_{k=1}^{N} p_k \psi\left(x_k + c\right)\right) = \psi^{-1}\left(\sum_{k=1}^{N} p_k \psi\left(x_k\right)\right) + c \tag{3}$$

if and only if $\psi(\ )$ is either linear or exponential.

A. Rényi's Entropy

Consider a random variable X which takes the values $x_1, x_2, \quad x_N$ with probabilities $p_1, p_2, \quad p_N$. The amount of information generated when X takes the value x_k is given by the Hartley information measurement function $I\left(x_k\right)$ [6]:

$$I\left(x_k\right) = \log_2\left(\frac{1}{p_k}\right) \text{ bits} \tag{4}$$

The expected value of $I\left(x_k\right)$ yields the expression for Shannon's entropy [7]:

$$H\left(X\right) = \sum_{k=1}^{N} p_k I\left(x_k\right) = \sum_{k=1}^{N} p_k \log_2\left(\frac{1}{p_k}\right) \tag{5}$$

Rényi replaced the linear mean in (5) with the quasi-linear mean (2) to obtain a generalized measure of information:

$$H_\psi\left(X\right) = \psi^{-1}\left(\sum_{k=1}^{N} p_k \psi\left(\log_2\left(\frac{1}{p_k}\right)\right)\right) \tag{6}$$

For $H_\psi\left(X\right)$ to satisfy the entropy additivity of independent events, it must satisfy $\left\langle X + c\right\rangle_\psi = \left\langle X\right\rangle_\psi + c$ where c is a constant. From (3), this implies that $\psi\left(x\right) = cx$ (linear) or $\psi\left(x\right) = c2^{(1-\alpha)x}$ (exponential). Setting $\psi\left(x\right) = cx$ reduces (6) to the linear mean and yields Shannon's equation (5). Substituting $\psi\left(x\right) = c2^{(1-\alpha)x}$ and $\psi^{-1} = \frac{1}{(1-\alpha)}\log_2$ in (6) yields the expression for Rényi's entropy:

$$H_\alpha\left(X\right) = \frac{1}{(1-\alpha)}\log_2\left(\sum_{k=1}^{N} p_k^\alpha\right) \text{ where } \alpha > 0 \text{ and } \alpha \neq 1 \tag{7}$$

The equation for Rényi's entropy is therefore a general expression for entropy and comprises of a family of entropies for different values of the parameter α. Shannon's entropy is a special case of Rényi's entropy in the limit as $\alpha \to 1$. The argument of the logarithm function in (7) is called the Information Potential:

$$V_\alpha\left(X\right) = \sum_{k=1}^{N} p_k^\alpha \tag{8}$$

Error samples at the output of the adaptive system are viewed as information particles by ITL algorithms. These information particles are analogous to charged particles in physics. Each information particle has an information potential associated with it. By substituting (8) in (7), Rényi's entropy can be expressed as:

$$H_\alpha\left(X\right) = \frac{1}{(1-\alpha)}\log_2\left(V_\alpha\left(X\right)\right) \tag{9}$$

The Information Potential is the expected value of the probability density function of the samples raised to $\alpha - 1$:

$$V_\alpha\left(X\right) = \sum_{k=1}^{N} p_k^\alpha = \sum_{k=1}^{N} p_k p_k^{\alpha-1} = E\left[p_k^{\alpha-1}\right] \tag{10}$$

For $\alpha = 2$ in (7), we get Rényi's quadratic entropy, which has the useful property that it allows us to compute the entropy directly from the error samples. The equations for Rényi's quadratic entropy and quadratic information potential (QIP) are obtained by substituting $\alpha = 2$ in (9) and (10):

$$H_2(X) = \frac{1}{(1-2)}\log_2\left(V_2(X)\right) = -\log_2\left(V_2(X)\right)$$
$$V_2(X) = E\left[p_k^{2-1}\right] = E\left[p_k\right] \quad (11)$$

From (11), we can see that since the logarithm function is continuous and strictly monotonic, the quadratic entropy $H_2(X)$ is minimized by maximizing $V_2(X)$. Also, the QIP is simply the expected value of the probability density function (PDF) which is a scalar. It is for these reasons that Rényi's entropy is used in ITL algorithms instead of Shannon's entropy.

B. *Rényi's Quadratic Entropy Estimator*

In this paper, ITL techniques to train the MLP aim to maximize Rényi's quadratic information potential (i.e. minimize Rényi's quadratic entropy) of the error samples. Form (11), it is evident that to compute the QIP requires the knowledge of the PDF of the error samples. Since an analytical expression of the PDF is rarely available, computation of the QIP requires a non-parametric estimator of the PDF directly from the error samples.

The Parzen window estimator is a simple non-parametric estimator of the PDF of a random variable from its sample values. This estimator places a kernel function with its center at each of the sample points. The resulting output values are averaged over all the samples to estimate the PDF. A popular choice for the kernel function is the Gaussian kernel. The Gaussian kernel Parzen window estimator for samples $x_1, x_2, \quad x_N$ is expressed as:

$$p(x) = \frac{1}{N}\sum_{i=1}^{N} G_\sigma(x - x_i)$$

where: (12)

$$G_\sigma(u) = \frac{1}{\sqrt{2\pi}\sigma}\exp\left[-\frac{1}{2}\left(\frac{u}{\sigma}\right)^2\right]$$

σ is the window length of the estimator and it has to be carefully chosen to obtain an accurate estimate of the PDF. It has been proved [8] that the entropy estimated by a Parzen window estimator using a Gaussian kernel has a global minima in the direction where all the error samples are identical over the whole data set. In other words at the global minima, the PDF of the error samples is a dirac delta function.

Rényi's quadratic entropy for a continuous random variable is expressed as:

$$H_2(X) = -\log\int_{-\infty}^{\infty} p^2(x)dx \quad (13)$$

Substituting $p(x)$ from (12) for $p(x)$ in (13) as described in [9], we get:

$$H_2(X) = -\log\left[\frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N} G_{\sigma\sqrt{2}}(x_j - x_i)\right] \quad (14)$$

This expression shows that we can compute Rényi's quadratic entropy directly from the samples without first computing the PDF. Since, $H_2(X) = -\log_2\left(V_2(X)\right)$, the expression for the QIP estimator is:

$$V_{2,\sigma}(X) = \frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N} G_{\sigma\sqrt{2}}(x_j - x_i) \quad (15)$$

C. *Information Potential and Information Force between samples*

For samples $x_1, x_2, \quad x_N$, the estimate of the quadratic information potential between any 2 samples x_i and x_j is defined as:

$$V_2(x_i, x_j) = G_{\sigma\sqrt{2}}(x_i - x_j) \quad (16)$$

The derivative of the quadratic information potential is the information force between the samples x_i and x_j:

$$\begin{aligned} F_2(x_i, x_j) &= \frac{\partial}{\partial x_i}V_2(x_i; x_j) \\ &= \frac{\partial}{\partial x_i}G_{\sigma\sqrt{2}}(x_i - x_j) \\ &= -\frac{1}{2\sigma^2}G_{\sigma\sqrt{2}}(x_i - x_j)(x_i - x_j) \end{aligned} \quad (17)$$

In ITL based training of the MLP, the information forces between the error samples at the output of the neural network are backpropagated to update the weights of the neurons in each layer of the network.

III. Training the MLP using ITL techniques

Consider the MLP with one input layer (layer h), two hidden layers (layers i and j) and one output layer (layer k) as shown in Fig 1.

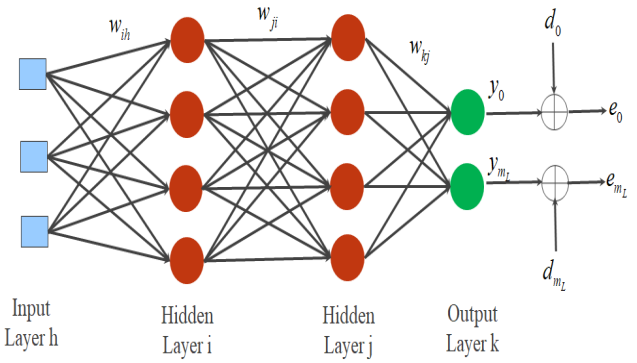

Figure 1: An MLP with one input, two hidden and one output layer

Each neuron of the MLP has one or more inputs with weights associated with them as shown in Fig 2. One of the inputs is a bias with a fixed value of 1.

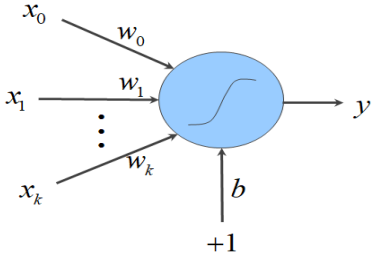

Figure 2: Structure of an artificial neuron

The sum of the weighted inputs and the bias is called the *net*. The *net* is passed through the activation function to generate an output y :

$$\begin{aligned} net &= \sum_{k=0}^{N} x_k w_k + b; \\ y &= \varphi(net) \end{aligned} \tag{18}$$

The activation function is a non-linear function and in this paper it is defined as:

$$\varphi(x) = \frac{1}{1+\exp(-x)} \tag{19}$$

The derivative of this activation function is:

$$\varphi'(x) = \varphi(x)\left[1-\varphi(x)\right] \tag{20}$$

Consider an MLP with L layers with the input layer being index $l=0$. Let there be m_l neurons in layer l. Therefore, the output layer has m_L neurons. Let w_{ji}^{l} be the weight associated with the connection from the i^{th} neuron of layer $l-1$ to the j^{th} neuron of layer l. Assume that there are N training samples. Let s and t be sample indices. The output y of the k^{th} neuron of the l^{th} layer is:

$$\begin{aligned} &y_k^l = \varphi\left(net_k^l\right) \\ &\text{where:} \\ &net_k^l = \sum_{i=0}^{m_{l-1}} w_{ki}^l y_i^{l-1} \end{aligned} \tag{21}$$

The total information potential of the error samples $\left[e(1), e(2), \ \ , e(s), \ \ e(t), \ \ e(N)\right]$ is expressed as [9]:

$$V = \sum_{s=1}^{N}\sum_{t=1}^{N}\sum_{k=1}^{m_L} V\left(e_k(s) - e_k(t)\right) \tag{22}$$

Due to the Parzen window estimation of the error PDF, the information potential computation is done over all error sample pairs. The total information potential of the output errors summed over all the output neurons for a given sample pair (s,t) is defined as:

$$J(s,t) = \sum_{k=1}^{m_L} V\left(e_k(s) - e_k(t)\right) \tag{23}$$

Taking the derivative of $J(s,t)$ with respect to the output layer weights:

$$\begin{aligned} \frac{\partial J(s,t)}{\partial w_{ki}^l} &= V'\left(e(s)-e(t)\right)\left[\frac{\partial e(s)}{\partial w_{ki}^l} - \frac{\partial e(t)}{\partial w_{ki}^l}\right] \\ &= V'\left(e(s)-e(t)\right)\left[\begin{matrix} \frac{\partial e(s)}{\partial y_k^l(s)}\frac{\partial y_k^l(s)}{\partial net_k^l(s)}\frac{\partial net_k^l(s)}{\partial w_{ki}^l} - \\ \frac{\partial e(t)}{\partial y_k^l(t)}\frac{\partial y_k^l(t)}{\partial net_k^l(t)}\frac{\partial net_k^l(t)}{\partial w_{ki}^l} \end{matrix}\right] \end{aligned}$$

$$\begin{aligned} \frac{\partial J(s,t)}{\partial w_{ki}^l} &= V'\left(e(s)-e(t)\right)\left[\begin{matrix} -\varphi'\left(net_k^l(s)\right)y_i^{l-1}(s) + \\ \varphi'\left(net_k^l(t)\right)y_i^{l-1}(t) \end{matrix}\right] \\ &= V'\left(e(s)-e(t)\right)\varphi'\left(net_k^l(t)\right)y_i^{l-1}(t) - \\ &\qquad V'\left(e(s)-e(t)\right)\varphi'\left(net_k^l(s)\right)y_i^{l-1}(s) \\ &= \delta_k^l(t,s)\,y_i^{l-1}(t) - \delta_k^l(s,t)\,y_i^{l-1}(s) \end{aligned} \tag{24}$$

The sensitivities of the local information potentials are denoted as $\delta_k^l(s,t)$ and $\delta_k^l(t,s)$

For the hidden layer $l-1$ and for any arbitrary weight w_{ji} the error gradient can be written as:

(25)

$$\begin{aligned} \frac{\partial J(s,t)}{\partial w_{ji}^{l-1}} &= V'\left(e(s)-e(t)\right)\left[\frac{\partial e(s)}{\partial w_{ji}^{l-1}} - \frac{\partial e(t)}{\partial w_{ji}^{l-1}}\right] \\ &= V'\left(e(s)-e(t)\right)\left[\begin{matrix} \frac{\partial e(s)}{\partial y_j^{l-1}(s)}\frac{\partial y_j^{l-1}(s)}{\partial net_j^{l-1}(s)}\frac{\partial net_j^{l-1}(s)}{\partial w_{ji}^{l-1}} - \\ \frac{\partial e(t)}{\partial y_j^{l-1}(t)}\frac{\partial y_j^{l-1}(t)}{\partial net_j^{l-1}(t)}\frac{\partial net_j^{l-1}(t)}{\partial w_{ji}^{l-1}} \end{matrix}\right] \\ &= V'\left(e(s)-e(t)\right)\left[\begin{matrix} -\sum_{k=1}^{m_l}\varphi'\left(net_k^l(s)\right)w_{kj}^l\varphi'\left(net_j^{l-1}(s)\right) + \\ y_i^{l-2}(s) \\ \sum_{k=1}^{m_l}\varphi'\left(net_k^l(t)\right)w_{kj}^l\varphi'\left(net_j^{l-1}(t)\right) \\ y_i^{l-2}(t) \end{matrix}\right] \\ &= \sum_{k=1}^{m_l}\delta_k^l(t,s)\,w_{kj}^l\varphi'\left(net_j^{l-1}(t)\right)y_i^{l-2}(t) - \\ &\qquad \sum_{k=1}^{m_l}\delta_k^l(s,t)\,w_{kj}^l\varphi'\left(net_j^{l-1}(s)\right)y_i^{l-2}(s) \\ &= \delta_k^{l-1}(t,s)\,y_i^{l-2}(t) - \delta_k^{l-1}(s,t)\,y_i^{l-2}(s) \end{aligned}$$

The backpropagation algorithm to train the MLP using ITL techniques is as follows [9]:

1. For $s,t = 1, \ \ , N$ and $k = 1, \ \ , m_L$, compute the sensitivities of the local information potential using the equations:

$$\begin{aligned}\delta_k^l(s,t) &= V'\left(e_k(s)-e_k(t)\right)\varphi'\left(net_k^l(s)\right)\\ &= F\left(e_k(s),e_k(t)\right)\varphi'\left(net_k^l(s)\right)\\ \delta_k^l(t,s) &= V'\left(e_k(s)-e_k(t)\right)\varphi'\left(net_k^l(t)\right)\\ &= F\left(e_k(s),e_k(t)\right)\varphi'\left(net_k^l(t)\right)\end{aligned} \quad (26)$$

2. For layer indices $l = L, \quad ,1$ compute the sensitivities of the local information potential of the hidden layers using the equations:

$$\begin{aligned}\delta_j^{l-1}(s,t) &= \varphi'\left(net_j^{l-1}(s)\right)\sum_{k=1}^{m_l}\delta_k^l(s,t)w_{kj}^l\\ \delta_j^{l-1}(t,s) &= \varphi'\left(net_j^{l-1}(t)\right)\sum_{k=1}^{m_l}\delta_k^l(t,s)w_{kj}^l\end{aligned} \quad (27)$$

3. After all the sensitivities of the local information potentials have been computed, the weights of the neural network are updated using the equation:

$$\Delta w_{ji}^l = \mu\left[\delta_j^l(t,s)y_i^{l-1}(t)-\delta_j^l(s,t)y_i^{l-1}(s)\right] \quad (28)$$

In the neuron weight update equation, μ is the learning rate of the neural network. The backpropagation of the information forces through the neural network is done using a gradient ascent method, since the goal is to maximize the information potential (i.e. minimize the entropy) of the error samples. After the training is completed the error of the neural network may have a non-zero mean. This is because minimizing the entropy results in minimizing the uncertainty in the error which is achieved even when the error has a constant non-zero value. In order to drive the error to the desired level, different bias values may be added to each output node of the MLP.

IV. SIMULATION RESULTS

The ITL based MLP training algorithm was tested on the Iris dataset [10] and the MNIST handwriting dataset [11]. The Iris dataset contains 50 sample vectors from each of the 3 species of the Iris flower (*Iris setosa, Iris virginica* and *Iris versicolor*). Each sample vector consists of measurements of 4 features: length and width of the sepals and petals of the flower in centimeters. A 3-layer MLP was used to train and classify the Iris species type from its input feature vector. The MLP had 4 input nodes (one for each input feature), 6 hidden nodes and 3 output nodes (one for each classification species). The MLP was trained in a batch sequential manner where the batch size was 150, the learning rate μ was 0.0275. The length of the Parzen window estimator for each output node was initially set to different values given by the vector $\sigma = [2.8, 5.5, 5.2]$.

The MLP was trained without varying the initial σ value for any of the nodes and also by varying the σ value for the nodes depending on the variance of the error of each node. The algorithm that varied the σ value, averaged the variance of the error at each output node over 5 epochs. If the variance of the error decreased from the last average value, the value of σ for that node was multiplied by a factor of 0.99. If the variance of the error increased from the last average value, the value of σ for that node was multiplied by a factor of 1.01. The sensitivity of the MSE to the estimator's window length can be seen in Figure 3. For the training example which kept the value of σ constant throughout the training, the MSE is a smooth curve. For the training algorithm which varied the value of σ, the MSE curve shows temporary instability as the change in σ results in a rapid increase in the information potential. This is turn causes a significant perturbation in the weights of the neural network resulting in temporary instability in the MSE.

In order to understand the reason for the instability of the MSE in Figure 3 at Epoch 664, the Information Potential and the delta between the error samples are plotted in Figure 4 for output node 2 centered at error sample 75. Here you see for epoch 664 where there is an instability in the MSE. The red dots are the MLP output node error sample pair deltas plotted on the same plot as the Information potential. You can see the peak of this curve is close to zero (which is expected) but the error sample pair deltas are clustered far from 0. From this figure it can be seen that at Epoch 664, the delta between the sample pairs is clustered at locations which are far from the peak of the Information Potential curve. At locations where the MSE is stable, the error sample deltas are clustered at the peak of the Information Potential curve as shown in Figure 5 at Epoch 500. Similar results are shown in Figure 6 for Epoch 680.

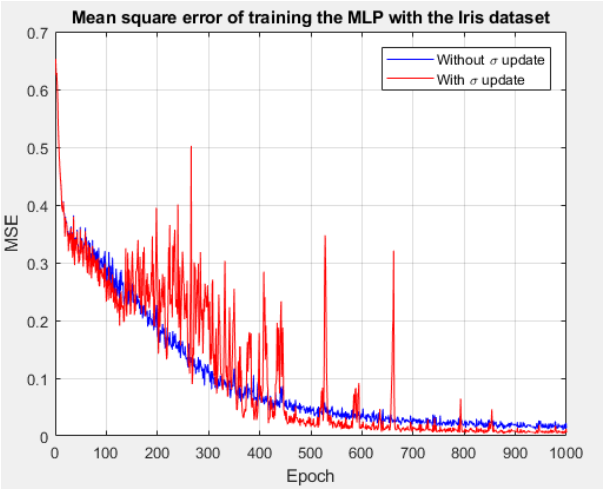

Figure 3: MSE of the MLP training with and without σ update

Figure 3 shows that the residual MSE is smaller for the training algorithm that varies the σ of the PDF estimator. As the training progresses the error PDF tends to a delta function. Therefore, reducing the value of σ as the variance of the error reduces, results in a better estimate of the PDF. Figure 7 shows that the information potential of the error samples increases as σ is reduced in response to a reduction of the error variance. If the window length is kept constant during the training, the information potential remains constant and results in inaccurate estimate of the error pdf.

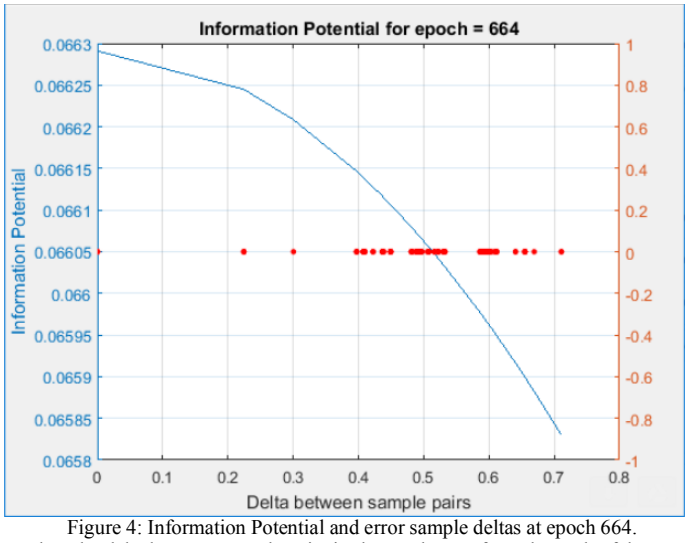

Figure 4: Information Potential and error sample deltas at epoch 664. When the delta between sample pairs is clustered away from the peak of the Information Potential curve, there is an instability in the MSE

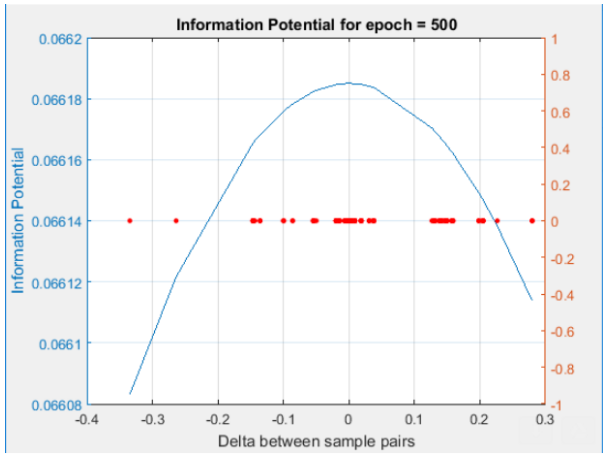

Figure 5: Information Potential and error sample deltas at epoch 500. When the delta between sample pairs is clustered at the peak of the Information Potential curve the MSE is stable

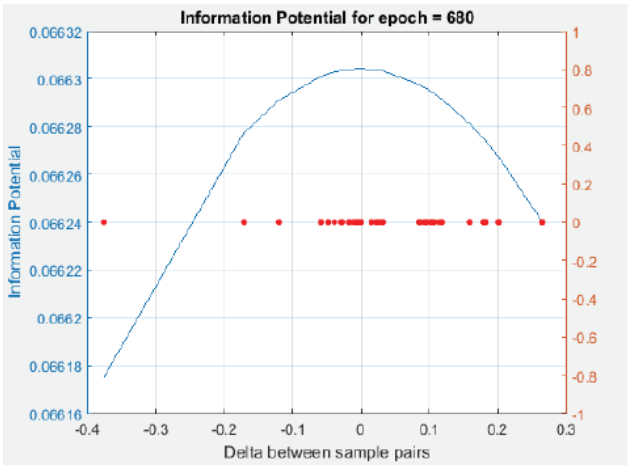

Figure 6: Information Potential and error sample deltas at epoch 680. When the delta between sample pairs is clustered at the peak of the Information Potential curve the MSE is stable

At the beginning of the MLP training the PDF of the error does not have a Gaussian distribution. As the training progresses, the PDF becomes Gaussian and eventually converges to a delta function as shown in Figure 8. The Parzen estimate of the error (red curve in Figure 8) closely matches the shape of the PDF.

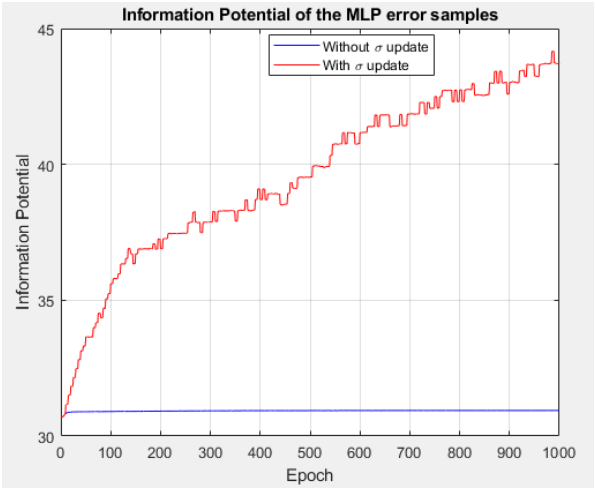

Figure 7: Information Potential of the error samples of the MLP

The 3-layer MLP for the MNIST database consists of 784 input neurons, 900 neurons in the hidden layer and 10 neurons in the output layer. The accuracy of the ITL trained MLP is 97% for the Iris dataset and 90% for the MNIST dataset.

Figure 8: PDF of the error and the error estimate for different epochs at output node #1 of the MLP

V. Discussions

It looks like the instability in the MSE occurs when the cluster of MLP output node error sample pair deltas are not close to 0. The Information Potential (which is the PDF of the error sample pairs) between samples pairs is always highest close to zero. However, if most of the error sample pair deltas are not

around zero, there seems to be a corresponding instability in the MSE.

From Figure 4, the plot for epoch 664 where there is an instability in the MSE, you can see the peak of this curve is close to zero (which is expected) but the error sample pair deltas are clustered far from 0.

When there is no instability in the MSE, the Information Potential curve looks like Figures 5 and 6. In these plots, the error sample pair deltas are clustered around the peak of the Information Potential curve.

VI. Conclusions

ITL-based techniques to train MLPs converge to an optimal solution by maximizing the information potential or minimizing the entropy of the error samples. The MEE-based backpropagation algorithm is extremely sensitive to the Parzen window size σ. Adapting the value of σ results in temporary instability but eventually results in lower residual error.

The temporary instability is caused by the output node error delta samples moving away from the peak of the Information Potential curve. The PDF of the error samples of the MLP tends towards a delta function as the weights converge to the optimal values.

Future work will focus on tracking the locations of the error sample delta clusters and moving them to the peak of the Information Potential curve. One of the ideas that we are exploring to curtail this potential instability is to use the BFGS (Broyden–Fletcher–Goldfarb–Shanno) of unconstrained non-linear optimization [13][14] on the Information Potential surface. This will likely introduce additional computational complexity into the entire ITL based multi-layer perceptron algorithm, however, the potential instability will be curtailed. Results of this work will be presented in the future.

References

[1] R. O. Duda, P. E. Hart, D. G. Stork, Pattern Classification 2nd edition, New York: John Wiley & Sons Inc, 2001.

[2] A. Rényi, "On measures of entropy and information", Proc. Of the 4th Berkeley Symposium on math, statsitics and probability, vol 1, Berkeley, California: University of California Press, 1961, pp 547-561.

[3] A. N. Kolmogorov, "Sur la notion de la moyenne", Atti Accad. Naz. Lincei. Rend. 12:9 (1930), pp 388-391.

[4] M. Nagumo, Über eine klasse von mittelwerte, Japanese Journal of Mathematics 7 (1930) pp 71–79.

[5] G. H. Hardy, J. E. Littlewood, G. Pólya, Inequalities, Cambridge, 1934

[6] R. V. L. Hartley, Transmission of information, Bell System Technical Journal 7 (1928) p 535

[7] C. E. Shannon, A mathematical theory of communication, Bell System Technical Journal 27 (1948) p 379.

[8] D. Erdogmus D. and J. Principe, "An Error-Entropy Minimization Algorithm for Supervised Training of Nonlinear Adaptive Systems", Trans. On Signal Processing, Vol. 50, No. 7, pp 1780-1786.

[9] J. Principe, Information Theoretic Learning, New York, Springer, 2010.

[10] Iris dataset, Univeristy of California Irvine, Machine Learning Repository, http://archive.ics.uci.edu/ml/datasets/Iris [Accessed: 10-Oct- 2017]

[11] Modified National Institute of Standards and Technology (MNIST), http://yann.lecun.com/exdb/mnist [Accessed: 10- Oct- 2017]

[12] Manas Deb and Tokunbo Ogunfunmi,"Using Information Theoretic Learning Techniques to Train Neural Networks", Proceedings of the 2017 Asilomar Conference on Circuits, Syetsm and Computers, Nov. 2017.

[13] Igor Griva, Stephen Nash and Areila Sofar, Linear and Noninear Optimization, SIAM Publisher, 2008.

[14] Amir Beck, Introduction to Nonlinear Optimization: Theory, Algorithms, and Applications with MATLAB, SIAM Publisher, 2014.