Complex Systems **6** (1992) 495–505

A Comparison between Squared Error and Relative Entropy Metrics Using Several Optimization Algorithms

Raymond L. Watrous
Siemens Corporate Research
755 College Road East, Princeton, NJ 08540

Abstract. Convergence rates and generalization performance are compared for the squared error metric and a relative entropy metric on a contiguity problem using several optimization algorithms. The relative entropy measure converged to a good solution slightly more often than the squared error metric given the same distribution of initial weights. However, where the results differed, the squared error metric converged on average more rapidly to solutions that generalized better to the test data. These results are not in complete agreement with some results previously published.

1. Introduction

A considerable reduction in the number of iterations required to reach good performance on a classification task was reported by Solla et al. for a relative entropy (RE) metric as compared to the standard squared error (SE) metric [2]. The improvement in convergence speed was attributed to differences in the contours of the error surfaces [2]; curiously, this improvement was termed "accelerated learning," although it resulted from a change in the error metric rather than in the optimization algorithm.

Different learning algorithms have different convergence properties. For example, in the quadratic case, first-order methods converge at a rate that depends on the condition number of the Hessian matrix, whereas the quasi-Newton methods converge in a number of steps determined by the *rank* of the Hessian [1]. Thus, the convergence rate of different learning algorithms can depend on different properties of the Hessian matrix. Clearly, different error metrics can have different Hessians; therefore, changing the error metric can have different consequences for different learning algorithms.

In this paper, several optimization algorithms are used to compare the SE and RE metrics on the same classification problem for which results were previously reported [2]. Section 2 restates the error metrics and the classification problem and describes the experimental conditions. In Section 3,

the metrics are compared using an online learning algorithm, to establish a baseline for comparison with the results of Solla. This section also investigates the match between the fixed step size used in the learning algorithm and the error metric. Optimization results using a quasi-Newton algorithm are presented in Section 4.

2. Problem Statement

2.1 Error Metrics

Using the notation of Solla, the squared error metric

$$E_Q = \frac{1}{2} \sum_{\alpha=1}^{m} \sum_{j=1}^{N_L} (\mathcal{O}_j^\alpha - \mathcal{T}_j^\alpha)^2 \tag{1}$$

and the relative entropy metric

$$E_L = \sum_{\alpha=1}^{m} \sum_{j=1}^{N_L} \left\{ \mathcal{T}_j^\alpha \ln \frac{\mathcal{T}_j^\alpha}{\mathcal{O}_j^\alpha} + (1 - \mathcal{T}_j^\alpha) \ln \frac{1 - \mathcal{T}_j^\alpha}{1 - \mathcal{O}_j^\alpha} \right\} \tag{2}$$

are compared, where $\alpha = 1 \ldots m$ is the index of the training token, $j = 1 \ldots N_L$ is the index of the output unit, $\mathcal{T}_j^\alpha$ is the target value for the jth output unit, and $\mathcal{O}_j^\alpha$ is the actual output unit value. Note that in the case of an exact solution, $E_Q = E_L = 0$, since $\mathcal{O}_j^\alpha = \mathcal{T}_j^\alpha$ for all α, j.

The faster convergence obtained by Solla using the RE metric was explained by appealing to its greater steepness [2, page 636]. However, the increased "steepness" could simply be the result of a scale factor, which could be compensated for by the learning constant. (Suppose, for example, that the RE metric were simply κ times the SE metric; then, using $\eta_{RE} = \eta_{SE}/\kappa$ would result in identical weight updates in the scaled space.) Thus, it becomes necessary to distinguish the effects of the shape of the error surface from the effects of step size in comparing error metrics. This distinction is pursued in what follows by

1. trying to separate scaling from shape, and
2. using an optimization method in which the step size is chosen optimally at each iteration.

Suppose the RE metric is related to the SE metric by a constant scale factor. We consider the contribution of a single observation to the metric under the condition of $0, 1$ targets. In this case,

$$e_L = -\ln(1 - z) \tag{3}$$

and

$$e_Q = \frac{1}{2} z^2 \tag{4}$$

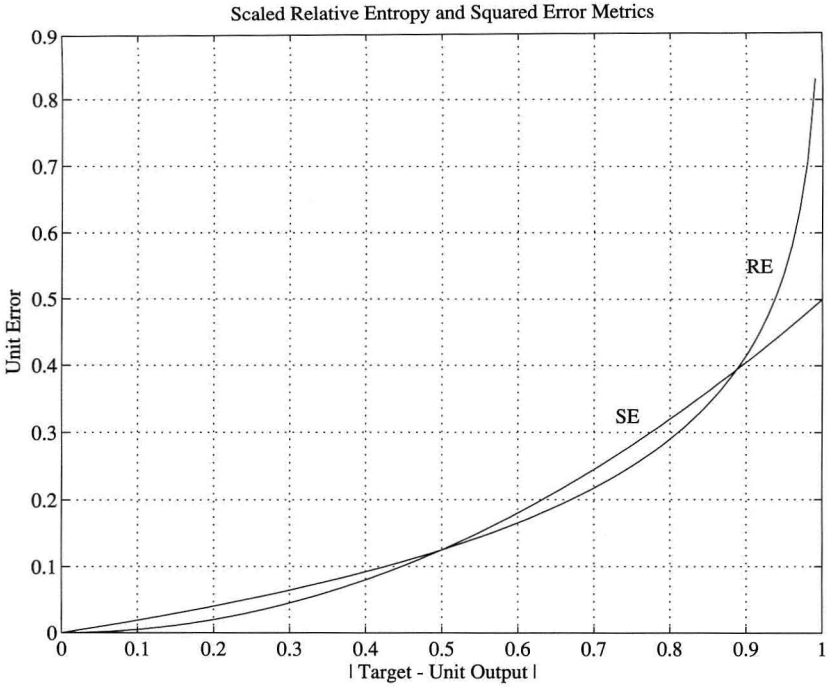

Figure 1: Comparison of scaled relative entropy and squared error metrics.

where $z = |\mathcal{O}_j^\alpha - \mathcal{T}_j^\alpha|$. Suppose further that the RE metric is scaled so that it has the same value as the SE metric for $z = 0.5$; that is, the metrics are identical for the case where the unit is maximally "uncertain."

$$e_Q = \frac{1}{2}(0.5)^2 = \kappa * (e_L = -\ln(0.5)) \tag{5}$$

whence

$$\kappa = 8 \ln 2 \tag{6}$$

Accordingly, we define the scaled relative entropy (SRE) metric as:

$$E_L = \frac{1}{8 \ln 2} \sum_{\alpha=1}^{m} \sum_{j=1}^{N_L} \left\{ \mathcal{T}_j^\alpha \ln \frac{\mathcal{T}_j^\alpha}{\mathcal{O}_j^\alpha} + (1 - \mathcal{T}_j^\alpha) \ln \frac{1 - \mathcal{T}_j^\alpha}{1 - \mathcal{O}_j^\alpha} \right\} \tag{7}$$

We now compare graphically the SE and scaled RE metrics for a single unit in Figure 1. It may observed that the metrics are equal for z values of 0, 0.5, and approximately 0.9. The scaled RE is increasingly greater than the SE metric as the absolute difference between target and actual exceeds 0.9. However, when the absolute difference is between 0.5 and 0.9, the SE

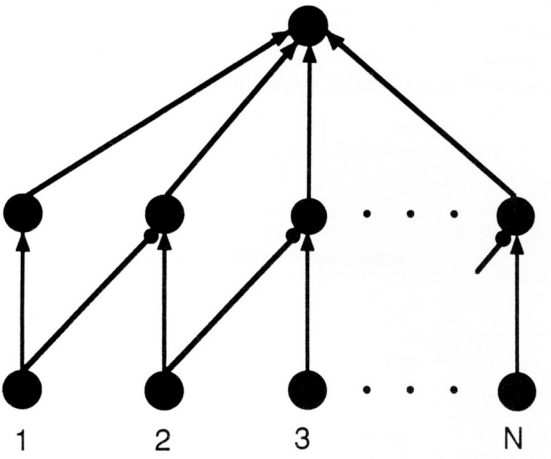

Figure 2: A solution to the contiguity problem with $L = 2$. All couplings $\{W_{ij}\}$ have absolute value of unity. Excitatory ($W_{ij} = +1$) and inhibitory ($W_{ij} = -1$) couplings are indicated by $\rightarrow$ and $—\bullet$, respectively. Intermediate units are biased by $W_i^{(1)} = -0.5$; the output unit is biased by $W_i^{(2)} = -(k_0 + 0.5)$.

is slightly greater, while for differences of 0 to 0.5, the scaled RE is again slightly greater.

This observation suggests that, apart from a constant scale factor, the effective difference between the SE and RE metrics might lie in the treatment of outliers, where the difference between actual and desired response is greater than 0.9. For other cases, the differences might be relatively small.

2.2 Contiguity Problem

Following Solla, the contiguity problem was used to compare the SE and RE metrics. The contiguity problem consists of finding the number k of clumps of contiguous bits of value 1. The simpler form of the contiguity problem used by Solla and here is a dichotomous classification task for $k \leq 2$ and $k > 2$.

An exact solution to the contiguity classification task is given by Solla using N hidden units [2, Figure 2], and is reproduced here as Figure 2. Note that the input is not treated as circular in the contiguity problem; thus, there are no links from the final input unit to the initial hidden unit.

The hidden units compute 0-1 transitions over pairs of adjacent bits, whereas the output unit simply counts the transitions and checks whether there are more than k. Note that this solution is exact for target values of $0, 1$ *only if the network consists of binary threshold units.* The binary

threshold function can be realized using sigmoidal units with infinite gain; this, however, implies unbounded weights.

With bounded weights, the hidden units will take on three values, depending on whether one input is equal, less than, or greater than the other input. Thus, the output unit must take on different values for some members within the same class. This means that there can be no zero-error solution, and a particular solution will represent the best compromise given the training set.

2.3 Experimental Conditions

The following experimental conditions were chosen to be as close as possible to those described by Solla [2].

The contiguity problem was investigated for $N = 10$ bits. Networks with 10 input units, 10 hidden units, each with a fan-in of p, and one output unit were evaluated for $p = 2 \ldots 6$. 50 strings randomly chosen from the 330 $k = 2$ negative examples, and 50 strings randomly chosen from the 462 $k = 3$ positive examples were used as training data; the remaining strings constituted the test set. The same training/test split was used in all the experiments reported.

The initial weight values were chosen from a normal distribution $\mathcal{N}(0, 1)$, subject to a hard limit of ± 1.

Following Solla, target values of 0 and 1 were used for training throughout. For classification, network responses $|\mathcal{O}_j^\alpha - \mathcal{T}_j^\alpha| \leq 0.1$ were considered correct; the classification accuracy was computed as the number of correct responses relative to the total number of examples.

In setting the stopping criterion, we guaranteed a *maximum* absolute difference of $|\mathcal{O}_j^\alpha - \mathcal{T}_j^\alpha| = 0.1$. Thus, similarly to Solla, we set

$$\epsilon_{SE} = \frac{1}{2} \sum_{\alpha=1}^{m} (\mathcal{O}^\alpha - \mathcal{T}^\alpha)^2 \leq \frac{1}{2}(0.1)^2 = 0.005. \tag{8}$$

However, in contrast to Solla, we did not use a squared error measure as a stopping criterion for the RE metric; instead, we used the RE metric itself, under the same assumption as before, and set

$$\epsilon_{RE} = \sum_{\alpha=1}^{m} \left\{ \mathcal{T}^\alpha \ln \frac{\mathcal{T}^\alpha}{\mathcal{O}^\alpha} + (1 - \mathcal{T}^\alpha) \ln \frac{1 - \mathcal{T}^\alpha}{1 - \mathcal{O}^\alpha} \right\} \leq \ln \frac{1}{0.9} \approx 0.10 \tag{9}$$

3. Online Optimization

3.1 Optimal Step Size

The relationship between the error metric and the learning constant η was explored in a small experiment in which the same randomly initialized network ($p = 2$) was optimized separately for each error metric for 100 epochs of the online algorithm using 15 values of the learning constant η from 0.001 to 10.0.

Step Size	Error Metric SE	RE	SRE
0.001	12.81	69.73	12.80
0.002	12.69	69.13	12.66
0.005	12.57	67.50	12.59
0.010	12.39	62.95	12.49
0.020	12.05	55.76	12.31
0.050	10.36	43.64	11.56
0.100	9.09	38.93	10.18
0.125	8.58	35.90	9.88
0.200	7.52	34.42	8.68
0.250	7.07	35.88	8.11
0.500	6.31	38.58	7.12
1.000	5.42	58.44	6.22
2.000	5.94	126.28	7.17
5.000	5.55	89.79	9.76
10.000	9.80	249.06	18.90

Table 1: Value of Error Metric after 100 Epochs of Online Optimization using Fixed Step Size Algorithm for Various Values of η.

A plot of the error metric, evaluated over the complete training set, as a function of complete epochs of the online algorithm, showed a monotonic decrease in the SE and SRE metrics for values of $\eta \leq 0.5$ and in the RE metric for $\eta \leq 0.1$. For larger values of η, there were oscillations in the decreasing function, which increased in magnitude with increasing η. The RE metric seemed to be unstable for $\eta \geq 2.0$.

The values of the SE, RE, and SRE metrics after 100 epochs are listed for each value of η in Table 1. The RE is much greater than the SE and SRE for all values of η. The values of the SE and SRE metrics are quite similar, especially for small values of η.

The minimum SE and SRE values were reached using $\eta = 1.0$, whereas the minimum RE value occurred at $\eta = 0.2$; thus, there was a difference in the optimal step size of a factor of 5. Note that this factor is not very different from the scale factor of $8 \ln 2 \approx 5.55$. Note also that the step sizes which led to minimal metric values after 100 epochs were slightly larger than the maximum step size for which the descent was monotonic.

The values used by Solla (0.25 for the SE metrics, 0.125 for the RE metric) are both smaller than the optimal values obtained by this experiment. However, the value of 0.125 used with the RE metric is slightly larger than the maximum value leading to monotonic descent (0.1) and may be significantly larger relative to the error surface than the value used for the SE.

3.2 Online Experiments

Following Solla [2], 10 optimizations for both the SE and RE metrics were carried out from different random initial conditions for values of $p = 2 \ldots 6$.

p	Total Error	Training (%)	Test (%)
2	1.27 ± 1.08	84.4 ± 11.6	70.5 ± 14.7
3	1.08 ± 0.94	83.8 ± 9.0	63.6 ± 6.1
4	0.59 ± 0.53	92.2 ± 6.3	63.0 ± 4.0
5	0.72 ± 0.65	92.8 ± 6.4	58.9 ± 4.0
6	0.17 ± 0.14	97.4 ± 2.0	58.5 ± 1.8

Table 2: Results of Online Optimization of Squared Error Metric Using Fixed Step Size Algorithm with $\eta = 0.25$.

In addition, the SRE metric was similarly evaluated; there were $10*5*3 = 150$ experiments.

The optimizations were carried out using a fixed step-size algorithm operated in online mode. Following Solla, a learning rate of 0.25 was used for the SE metric and 0.125 for the RE metric. The learning rate for the SRE metric was 0.25. Since the SRE metric is simply a scaled version of the RE, this latter experiment amounts to using a scaled step size of 0.045 for the RE metric.

After each pass through the training data, the objective function was evaluated over the corpus and recorded. The optimization was allowed to run until 2000 iterations were completed,[1] until the stopping criterion was met, or until the algorithm halted because it was unable to further reduce the function.

During optimization, the SE and SRE metrics decreased monotonically. The decrease in the RE metric was non-monotonic in 10 cases; generally, the non-monotonic cases were among the highest in final error values and lowest in training and test performances.

In no case was the minimum metric value achieved within 2000 iterations. However, the training performance achieved after 2000 iterations was perfect in 8 cases for the SE, 10 cases for the SRE, and 16 cases for the RE metric. The average test performance of the perfectly trained networks was 73.4% for the SE, 72.7% for the RE, and 67.9% for the SRE metric. In no case was the test performance perfect.

The average metric value at termination, and training, and test performances for the 10 optimizations are summarized in Tables 2, 3, and 4. In the case of the SE metric (Table 2), the average error decreased as the receptive field p increased, except for $p = 5$. There was a corresponding increase in the performance on the training data, except for $p = 3$, and a corresponding *decrease* in the performance on the test data.

The average RE (Table 3) similarly decreased as the size of the receptive field increased. There was a corresponding increase in the performance on the training data and, for $p > 2$, a decrease in performance on the test data

[1]This corresponds roughly to the number of total passes through the training data reported by Solla for networks to converge successfully for the SE metric.

p	Relative Entropy	Training (%)	Test (%)
2	6.47 ± 3.24	80.5 ± 9.3	66.2 ± 14.3
3	3.79 ± 4.14	90.2 ± 9.5	74.0 ± 11.8
4	1.67 ± 1.81	95.3 ± 5.3	69.9 ± 5.0
5	1.56 ± 1.00	96.0 ± 3.2	65.8 ± 2.5
6	0.76 ± 1.14	98.2 ± 3.5	63.4 ± 4.1

Table 3: Results of Online Optimization of Relative Entropy Metric Using Fixed Step Size Algorithm with $\eta = 0.125$.

p	Scaled Relative Entropy	Training (%)	Test (%)
2	2.20 ± 0.90	78.1 ± 8.6	63.5 ± 11.7
3	1.33 ± 1.07	86.9 ± 10.8	67.6 ± 8.0
4	0.79 ± 0.55	93.2 ± 5.7	67.3 ± 3.0
5	0.87 ± 0.70	93.2 ± 6.6	61.7 ± 4.0
6	0.58 ± 0.51	95.9 ± 4.6	60.1 ± 3.3

Table 4: Results of Online Optimization of Scaled Relative Entropy Metric Using Fixed Step Size Algorithm with $\eta = 0.25$.

with increasing p. Except for $p = 2$, the average performance on the training and test data was greater for the RE than for the SE.

The average SRE (Table 4) also generally decreased as the size of the receptive field increased (except for $p = 4$). There was a corresponding increase in the performance on the training data and, for $p > 2$, a decrease in performance on the test data with increasing p. The level of performance of the SRE was uniformly lower than the RE; since the SRE and RE metrics differ only by a scale factor, this demonstrates that the performance of the optimization depends critically on the step size. Like the RE, the SRE metric resulted in better performance than the SE for $p \geq 2$, although by a smaller margin.

Contrary to Solla, for $p = 2$, networks that converged to perfect training performance did not give perfect test performance. Also contrary to Solla, for $p = 2$, the average RE training and test performance was *worse* than the average SE training and test performance. Further contrary to Solla, for $p \geq 3$, learning was not always successful; although the average training performanced increased with p, perfect training performance was not always obtained within the limit of 2000 iterations.

4. Quasi-Newton Method

In comparing the SE and RE measures, it would be advantageous to remove the uncertainty about the optimal step size. This may be done using an optimization algorithm in which the step size is optimized at every point by a one-dimensional sub-optimization called a line search. This process is an integral part of the quasi-Newton methods, which have the additional

p	Total Error	Iterations	Training (%)	Test (%)
2	3.36 ± 1.12	492 ± 272	84.5 ± 6.1	65.3 ± 6.5
3	2.54 ± 0.94	589 ± 293	86.9 ± 6.9	66.1 ± 5.6
4	2.19 ± 1.14	668 ± 256	90.2 ± 6.1	66.2 ± 6.6
5	1.97 ± 0.80	534 ± 315	90.6 ± 5.7	62.8 ± 4.0
6	2.00 ± 0.73	568 ± 285	89.7 ± 4.5	60.3 ± 4.9

Table 5: Results of Squared Error Minimization Using BFGS Algorithm.

advantage that the search direction is oriented by an iteratively approximated inverse Hessian matrix. This serves to increasingly direct the line search toward the minimum of the function.

In the following experiments, the Broyden-Fletcher-Goldfarb-Shanno (BFGS) quasi-Newton method was used [1]. Optimizations were carried out from 20 different random initial conditions for both the SE and RE metrics for $p = 2 \ldots 6$; there were $20 * 5 * 2 = 200$ experiments. The initial networks formed a superset of the 10 used in previous experiments. The same 50 positive and 50 negative training strings were used as before.

The same termination criteria were also used, except that the maximum number of iterations was set at 1000; since each iteration typically involves at least 2 function evaluations, this limit seemed to be at least as generous as the one used for the fixed step size algorithms.

As guaranteed by the descent property of the quasi-Newton algorithms, the decrease in error was monotonic for all optimizations. In the SE case, the descent was often gradual to a plateau, with occasional abrupt down-steps, generally reaching the plateau value within 100 iterations; in no case did the algorithm terminate at the error criterion of 0.005. In the RE case, the descent similarly was often gradual to a plateau, with occasional abrupt down-steps; in fact, *the shape of the RE optimization curves were very similar to those for the SE metric for the same initial conditions* (see Figure 3). However, there were breakthroughs in 4 cases to the criterion value of 0.1. These breakthroughs, one at $p = 4$, two at $p = 5$ and one at $p = 6$, all occurred in less than 60 iterations. The performance of these networks on the training data was perfect and averaged 67.2% on the test data.

In 29 cases, 14 for the SE and 15 for the RE metric, the BFGS algorithm exhausted the preset limit of 1000 iterations. In the remaining 167 cases, the algorithm halted because it was unable to further reduce the metric.

The averages across all trials of the metric at termination, number of iterations, training and test accuracies are shown in Tables 5 and 6. It may be observed that the average squared error and average relative entropy decrease with increasing p, except for $p = 6$. The average number of iterations using SE is maximum at $p = 3$, and is less than the maximum for RE, which occurs at $p = 2$. The average training accuracy for the SE metric increases slightly with p until $p = 6$, where it dips slightly. The pattern for the RE metric is similar, although the value for $p = 2$ is much lower than the values for $p > 2$.

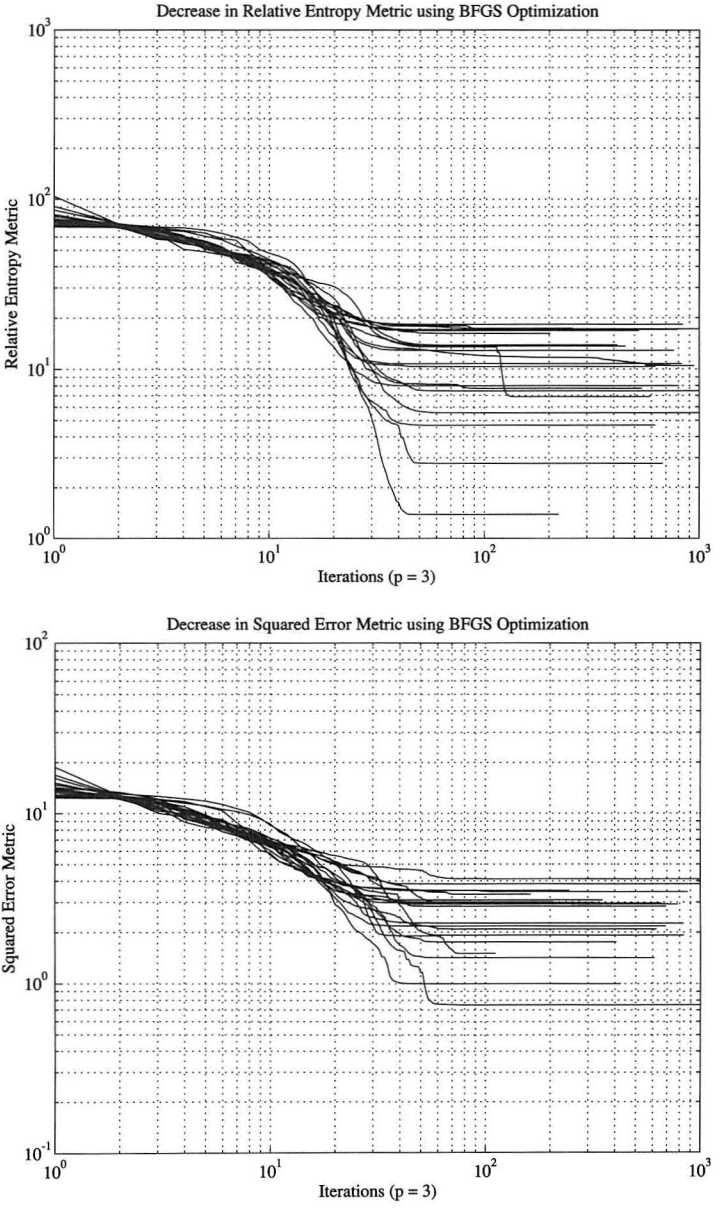

Figure 3: A comparison of optimizations from 20 different initial conditions for the squared error (lower panel) and relative entropy (upper panel) metrics using the BFGS algorithm. Iterations are shown up to 1000 on a log scale; note metrics shown on different log scales.

p	Relative Entropy	Iterations	Training (%)	Test (%)
2	19.95 ± 5.71	770 ± 261	70.1 ± 9.2	56.3 ± 7.9
3	10.96 ± 5.27	651 ± 256	83.0 ± 8.5	67.5 ± 6.5
4	9.20 ± 4.99	558 ± 255	85.3 ± 8.7	65.5 ± 7.5
5	6.64 ± 6.08	496 ± 314	89.6 ± 8.8	64.0 ± 6.4
6	7.94 ± 4.70	542 ± 305	86.7 ± 8.9	58.6 ± 7.0

Table 6: Results of Relative Entropy Minimization Using BFGS Algorithm.

The average training performance for the SE is greater than that for the RE for all p. Apart from $p = 2$, the difference between the SE and RE metrics in average test performance is not large; for $p = 2$, the SE leads to greater average test accuracy by 9%.

These results suggest that the RE metric does not accelerate learning with the BFGS algorithm. However, using RE may increase the likelihood of convergence to a global minimum; given the properties of the BFGS algorithm, this convergence can occur quite rapidly.

5. Conclusions

Several conclusions may be drawn from these experiments. First of all, it is clear that the choice of optimal step size is both problem- and metric-specific. Second, there seems to be a slightly higher probability of convergence to a good solution with the relative entropy metric for the same distribution of initial weights. However, for $p = 2$, the squared error leads (on average) to higher training set performance and better test set performance than the relative entropy metric.

It seems unlikely that these differences can be explained satisfactorily in terms of the relative steepness of the error surfaces, since that difference is taken into account by the quasi-Newton methods. It may be preferable to understand the differences in terms of the increased cost of extreme outliers obtained with the relative entropy metric.

Acknowledgements

The constructive comments of Marco Gori, Gary M. Kuhn, and an anonymous reviewer are gratefully acknowledged.

References

[1] D. G. Luenberger, *Linear and Nonlinear Programming* (Addison-Wesley, Reading, MA, second edition, 1984).

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