

Minimizing Output Error in Multi-Layer Perceptrons

Jonathan P. Bernick

Department of Computer Science

Coastal Carolina University

I. Abstract

It is well-established that a multi-layer perceptron (MLP) with a single hidden layer of N neurons and an activation function bounded by zero at negative infinity and one at infinity can learn N distinct training sets with zero error. Previous work has shown that the input weights and biases for such a MLP can be chosen in an effectively arbitrary manner; however, this work makes the implicit assumption that the samples used to train the MLP are noiseless. We demonstrate that the values of the input weights and biases have a provable effect on the susceptibility of the MLP to noise, and can result in increased output error. It is shown how to compute a quantity called Dilution of Precision (DOP), originally developed for the Global Positioning System, for a given set of input weights and biases, and further shown that by minimizing DOP the susceptibility of the MLP to noise is also minimized.

II. Introduction

Consider a multi-layer perceptron (MLP) with M input neurons, a single hidden layer of N neurons, one linear output neuron, and an activation function $g(a)$. The output $O(\mathbf{x})$ of this network may be written as

$$O(\mathbf{x}) = \sum_{i=1}^N \beta_i g(\mathbf{w}_i \cdot \mathbf{x} + b_i), \quad (1)$$

where β_i is the weight between the i th hidden neuron and the output neuron, w_{ij} the weight between the j th input neuron and the i th hidden neuron, $\mathbf{w}_i = [w_{i1} \ w_{i2} \ \dots \ w_{iM}]^T \in \mathfrak{R}^M$ the vector of weights between the input layer and the i th hidden neuron, $\mathbf{x} = [x_1 \ x_2 \ \dots \ x_M]^T \in \mathfrak{R}^M$ the vector of input values, and b_i the input bias of the i th hidden neuron.

For a set of distinct unweighted training samples (\mathbf{x}_i, t_i) , $\mathbf{x}_i \in \mathfrak{R}^M$, $t_i \in \mathfrak{R}$, $i = 1, \dots, P$, we may write the output of the MLP as

$$t_j = \sum_{i=1}^N \beta_i g(\mathbf{w}_i \cdot \mathbf{x}_j + b_i), \quad j = 1, \dots, P. \quad (2)$$

In training such a network, we seek to find \mathbf{w}_i , b_i , and β_i such that the output error $E(O)$, expressed as

$$E(O) = \frac{1}{P} \sum_{i=1}^P (O(\mathbf{x}_i) - t_i)^2, \quad (3)$$

is minimized.

Let us now specify that our activation function $g(a)$ is continuous and nonlinear, and that

$$g(a) \rightarrow \begin{cases} 0 & a \rightarrow -\infty \\ 1 & a \rightarrow \infty \end{cases} \quad (4)$$

(e.g., the sigmoid function). It is well-known that a MLP of this type can model a set of N distinct unweighted training samples with zero output error. Previous work has shown that the values of the input weights \mathbf{w}_i and biases b_i for such a MLP can be chosen in an effectively arbitrary manner, and algorithms for computing these values are detailed in [4], [5], and [10]. For this assertion of

arbitrariness to be true, though, our training samples must be noise-free. In this paper, we will examine the effects of noise on the weights and biases of the trained neural net, and define a criterion called Dilution of Precision; this criterion, originally developed for the Global Positioning System, will allow us to choose the neural net parameters so as to minimize the output error.

III. Noise Propagation in Multilayer Perceptrons

Let us consider the MLP from the introduction; i.e., one consisting of M input nodes, single hidden layer of N neurons, one linear output neuron, and an activation function $g(a)$ bounded as per equation (4). Given a set of N distinct unweighted training samples (\mathbf{x}_i, t_i) , $i = 1, \dots, N$, once we have chosen values for the input weights \mathbf{w}_i and biases b_i of the hidden layer, the output layer weights β_i , $i = 1, \dots, N$, may be found by solving the linear equation

$$\mathbf{H}\boldsymbol{\beta} = \mathbf{t}, \quad (5)$$

where $\mathbf{t} = [t_1 \ t_2 \ \dots \ t_N]^T \in \mathfrak{R}^N$, $\boldsymbol{\beta} = [\beta_1 \ \beta_2 \ \dots \ \beta_N]^T \in \mathfrak{R}^N$, and the hidden layer output matrix \mathbf{H} is defined as

$$\mathbf{H} \equiv \begin{bmatrix} g(\mathbf{w}_1 \cdot \mathbf{x}_1 + b_1) & \dots & g(\mathbf{w}_N \cdot \mathbf{x}_1 + b_N) \\ \vdots & \ddots & \vdots \\ g(\mathbf{w}_1 \cdot \mathbf{x}_N + b_1) & \dots & g(\mathbf{w}_N \cdot \mathbf{x}_N + b_N) \end{bmatrix}, \quad (6)$$

where w_{ji} is the weight between the j th input node and the i th hidden node, b_i the bias on the i th hidden node, and $\mathbf{w}_i = [w_{1i} \ w_{2i} \ \dots \ w_{Mi}] \in \mathfrak{R}^M$ the input weight vector for hidden node i . It has been shown ([4], [7]) that in this case \mathbf{w}_i and b_i may always be chosen such that \mathbf{H} is invertible, and accordingly $\boldsymbol{\beta} = \mathbf{H}^{-1}\mathbf{t}$. In the absence of output noise in our training samples, there is no inherent reason to prefer one set of parameters $(\mathbf{w}_1, \dots, \mathbf{w}_N, b_1, \dots, b_N, \boldsymbol{\beta})$ which accurately models the training

set over another; we shall demonstrate, though, that in the presence of output noise this is not the case.

Let us now consider a set of N distinct unweighted noisy training samples (\mathbf{x}_i, T_i) , $\mathbf{x}_i \in \mathfrak{R}^N$, $T_i \in \mathfrak{R}$, $i = 1, \dots, N$, such that

$$T_i = t_i + \delta t_i, \quad (7)$$

where t_i is the correct output value for the i th sample, δt_i the error term for that sample, and δt_i is small relative to t_i . If we let $\mathbf{T} = [T_1 \ T_2 \ \dots \ T_N]^T \in \mathfrak{R}^N$ replace \mathbf{t} in (5), then simple algebra shows that

$$\delta \boldsymbol{\beta} = \mathbf{H}^{-1}(\mathbf{t} + \boldsymbol{\delta t}) - \mathbf{H}\boldsymbol{\beta}, \quad (8)$$

where $\delta \beta_i$ is the error in the calculated value of the i th output weight, $\delta \boldsymbol{\beta} = [\delta \beta_1 \ \delta \beta_2 \ \dots \ \delta \beta_N]^T \in \mathfrak{R}^N$, and $\boldsymbol{\delta t} = [\delta t_1 \ \delta t_2 \ \dots \ \delta t_N]^T \in \mathfrak{R}^N$.

It is apparent from equation (8) that error in the training sample output values will result in an error in the calculated values of the output weights, and likewise apparent that the magnitude of this error will depend on the hidden layer output matrix \mathbf{H} . Accordingly, given flexibility in choosing the parameters of our MLP, it is obviously desirable to choose our input weights \mathbf{w}_i and biases b_i so as to minimize $\delta \boldsymbol{\beta}$.

It is well-known that, for an exactly determined or overdetermined linear system, the optimal solution for (5) may be obtained by a least-squares fit; i.e.,

$$\boldsymbol{\beta} = (\mathbf{H}^T \mathbf{H})^{-1} \mathbf{H}^T \mathbf{t}. \quad (9)$$

Let us consider a training set of N distinct unweighted noisy training samples (\mathbf{x}_i, T_i) , $\mathbf{x}_i \in \mathfrak{R}^N$, $T_i \in \mathfrak{R}$, $i = 1, \dots, N$, $(\mathbf{x}_i, T_i) = (\mathbf{x}_j, T_j)$ iff. $i = j$. If the noise components of all T_i are assumed to be

uncorrelated and generated by a normal random variable of mean zero and standard deviation σ , then the covariance law requires ([6], [11]) that

$$\mathbf{C}_{\delta\beta} = (\mathbf{H}^T \mathbf{H})^{-1} \mathbf{H}^T \mathbf{C}_{\delta\alpha} (\mathbf{H}^T \mathbf{H})^{-1} \mathbf{H}^T, \quad (10)$$

which simplifies to

$$\mathbf{C}_{\delta\beta} = (\mathbf{H}^T \mathbf{C}_{\delta\alpha}^{-1} \mathbf{H})^{-1}, \quad (11)$$

where $\mathbf{C}_{\delta\beta} \in \mathfrak{R}^{N \times N}$ and $\mathbf{C}_{\delta\alpha} \in \mathfrak{R}^{N \times N}$ are the covariance matrices for $\delta\beta$ and $\delta\alpha$, respectively. For the training set given above,

$$\mathbf{C}_{\delta\alpha} = \mathbf{I} \sigma^2, \quad (12)$$

where \mathbf{I} is the identity matrix, and thus

$$\mathbf{C}_{\delta\beta} = \sigma^2 (\mathbf{H}^T \mathbf{H})^{-1}. \quad (13)$$

Accordingly, if we use the notation

$$(\mathbf{H}^T \mathbf{H})^{-1} = \begin{bmatrix} h_{11} & \dots & h_{1N} \\ \vdots & \ddots & \vdots \\ h_{N1} & \dots & h_{NN} \end{bmatrix}, \quad (14)$$

then

$$\mathbf{C}_{\delta\beta} = \begin{bmatrix} \sigma^2 h_{11} & \dots & \sigma^2 h_{1N} \\ \vdots & \ddots & \vdots \\ \sigma^2 h_{N1} & \dots & \sigma^2 h_{NN} \end{bmatrix}. \quad (15)$$

The trace elements of $\mathbf{C}_{\delta\beta}$ represent the variances of the normally-distributed $\delta\beta_i$, where

$$\text{var}(\delta\beta_i) = \sigma^2 h_{ii}, \quad (16)$$

with $i = 1, \dots, N$. It thus follows that the standard deviation σ_i of output weight error $\delta\beta_i$ is

$$\sigma_i = \sigma \sqrt{h_{ii}}, \quad (17)$$

and the standard deviation $\sigma_{\delta\beta}$ of $|\delta\beta|$ is

$$\sigma_{\delta\beta} = \sigma \sqrt{h_{11} + \dots + h_{NN}} . \quad (18)$$

IV. Dilution of Precision

The quantity Dilution of Precision was developed as a means to estimate the quality of location coordinates obtained from the Global Positioning System (GPS), and was first described in [9]. We will now adapt it to serve a similar purpose for our calculated β .

Examining equation (18), we notice that the values of the trace elements h_{ii} of $(\mathbf{H}^T\mathbf{H})^{-1}$ have a multiplicative effect on the standard deviation of the error in $\delta\beta$; i.e., although not a source of error themselves, they amplify any error present in the MLP. Accordingly, it is apparent that, given two or more differing MLP's modeling the same function, it is preferable to use the MLP for which the square root of the sum of the trace elements h_{ii} of $(\mathbf{H}^T\mathbf{H})^{-1}$ is the smallest. We refer to this quantity as *Dilution of Precision* (abbreviated DOP), and thus define

$$\text{DOP} \equiv \sqrt{h_{11} + \dots + h_{NN}} . \quad (19)$$

The existence of DOP gives us a criterion by which to judge the desirability of one set of MLP weights and biases $(\mathbf{w}_1, \dots, \mathbf{w}_N, b_1, \dots, b_N, \beta)$ over another; i.e., the set with the lowest DOP value should be selected. Given that observed real-world training data is invariably noisy, the existence of DOP represents a valuable tool for minimizing the effects of such noise.

V. Discussion

The case of a MLP with N neurons being trained with N distinct unweighted training samples was chosen for the derivation of DOP because the hidden layer output matrix \mathbf{H} has been proven to be invertible ([4], [5], [7], [10]). Given that the cross-product matrix $\mathbf{H}^T\mathbf{H}$ is guaranteed to be invertible if \mathbf{H} is of full rank [11], DOP may be calculated for any MLP satisfying this criterion; i.e., the number of distinct training samples used to train the MLP is equal to or greater than the number of neurons in the hidden layer of the MLP, and the rows of \mathbf{H} are linearly independent. Accordingly, DOP may be used as a criterion with which to choose between two or more trained MLP's to use in modeling a function approximated by noisy training samples; i. e., the MLP with the lowest DOP will be the MLP whose weights and biases were least affected by the noise, and should thus be chosen. An obvious next research step would be develop or adapt an algorithm to generate MLP parameters (e.g., the algorithm detailed in [4]) to do so in a manner that minimizes DOP; intuition suggests that this task can most easily be accomplished with a Monte Carlo or genetic algorithm.

When \mathbf{H} is overdetermined (i.e., when the number of distinct training samples used to train the MLP is greater than the number of neurons in the hidden layer of the MLP), an additional criterion becomes available. From [3], we define the quantity DOP_{MAX} such that, for an MLP with P training samples,

$$\text{DOP}_{MAX} \equiv \max\left\{\sqrt{\text{DOP}^2 - \text{DOP}_i^2}, i = 1, \dots, P\right\}, \quad (20)$$

where DOP_i is the DOP calculated from \mathbf{H} when the i th row is removed. DOP_{MAX} has been shown to be a more effective indicator of system susceptibility to error than simple DOP [3] for GPS, and

it would be of worth to determine whether this improvement also applies to MLP's, and whether such improvement justifies the increased computation time.

We note that we have made the assumption that the errors in our training sample output values are small relative to those values. If this should turn out not to be the case for some training sample, it may be desirable to exclude that training sample from our system. GPS literature details numerous algorithms by which this can be accomplished (e.g., [12]), and the adaptation of these algorithms to MLP's would be desirable.

Finally, it is of worth to mention that the DOP_{MAX} criterion discussed above is a part of a family of GPS integrity algorithms known collectively as Receiver Autonomous Integrity Monitoring (RAIM). RAIM algorithms work by using redundant satellite data to estimate the quality of a position calculated with GPS ([1]), and it is reasonable to expect that other RAIM techniques might also be adapted to minimizing noise susceptibility in MLP's. An overview and comparison of some major RAIM methods may be found in [8], and an analysis of RAIM theory is available in [2].

VI. Conclusions

In this paper, it has been proved that MLP with M input neurons, a single hidden layer of N neurons, one linear output neuron, and a continuous activation function $g(a)$ bounded by zero at negative infinity and one at infinity, being trained by a set of N distinct unweighted noisy training samples, should have its parameters $(\mathbf{w}_1, \dots, \mathbf{w}_N, b_1, \dots, b_N, \beta)$ chosen such that its DOP is minimized. We have established DOP as a criterion by which to select between different MLP's implementing

the same function, and have generalized DOP to the larger class of all MLP's. Finally, we have suggested several future research directions and unsolved problems.

With development, DOP and related metrics can serve as invaluable tools for MLP construction, and it is recommended that such development take place.

VII. Acknowledgements

My deepest thanks to Jean-Louis Lassez for his meaningful comments concerning this problem.

VIII. References

- [1] Bernick, J., *Computational Analysis of GPS UDSRAIM Algorithms*, Ph.D. Dissertation at New Mexico Institute of Mining and Technology, December 1998.
- [2] Brown, R. G., "A Baseline GPS RAIM Scheme and a Note on the Equivalence of Three RAIM Methods," *Navigation: Journal of the Institute of Navigation*, Vol. 39, No. 3, Fall 1992.
- [3] Chin, G., Kraemer, J., and Brown, R., "GPS RAIM: Screening Out Bad Geometries Under Worst-Case Bias Conditions," *Proceedings of the 48th Annual Meeting of the Institute of Navigation*, June 29 - July 1, 1992.
- [4] Huang, G. B., and Babri, H. A., Upper Bounds on the Number of Hidden Neurons in Feedforward Networks with Arbitrary Bounded Nonlinear Activation Functions," *IEEE Trans. Neural Networks* 9, pp. 224 - 229 (1998).
- [5] Ito, Y., and Saito, K., "Superposition of Linearly Independent Functions and Finite Mappings by Neural Networks," *Math. Scientist* 21, pp. 27 - 33 (1986).

- [6] Langley, Richard B., "Dilution of Precision," *GPS World*, 2000 (journal - no month or volume given). Reprinted at http://www.gpsworld.com/0800/0800_2_dop.html
- [7] Li, X., "Training Multilayer Perceptrons Via Interpolations by Superpositions of an Activation Function," *Proceedings of The 2001 International Conference on Mathematics and Engineering Techniques in Medicine and Biological Sciences*, June 25 - 28, Monte Carlo Resort & Casino, Las Vegas, Nevada, USA.
- [8] Michalson, W., Bernick, J., Levin, P., and Enge, P., "RAIM Availability for Augmented GPS-Based Navigation Systems," *Proceedings of the Seventh International Technical Meeting of the Satellite Division of the Institute of Navigation (ION GPS-94)*.
- [9] Milliken, R. J., and Zoller, C. J., "Principle of Operation of NAVSTAR and System Characteristics," in *Global Positioning System*, Vol. 1, Institute of Navigation, Alexandria, 1980.
- [10] Sartori, M. A., and Antsaklis, P. J., "A Simple Method to Derive Bounds on the Size and to Train Multilayer Neural Networks," *IEEE Trans. Neural Networks* 2, pp. 467 - 471 (1991).
- [11] Strang, G., *Linear Algebra and Its Applications*, 3rd edition, Academic Press, Inc., New York, 1988.
- [12] van Graas, F., and Farrell, J., "Baseline Fault Detection and Exclusion Algorithm," in *Proceedings of the 49th Annual Meeting of the Institute of Navigation*, pp. 413-420.