ISSN:1738-6764, eISSN:2093-7504

Statistical Analyses of Cross-Entropy Error Function with Probabilistic Target Encoding for Training Neural Networks

Sang-Hoon Oh [1,*]

[1] Mokwon University; Professor; ohsanghoon16@gmail.com
[*] Correspondence

https://doi.org/10.5392/IJoC.2025.21.1.088

Manuscript Received 26 September 2024; Received 1 December 2024; Accepted 27 December 2024

Abstract: *Training neural networks with softmax outputs requires assigning target values to output nodes. Due to its simplicity, we often use one-hot encoding, which adopts "one" or "zero" as the desired output values. However, when training neural networks to minimize the cross-entropy error function between the desired and actual output node values, overfitting of neural networks to training samples becomes a significant issue. A probabilistic target encoding has been proposed to mitigate the overfitting. In this paper, we derive the optimal solutions for output nodes that minimize the cross-entropy error function using the probabilistic target encoding. In the extreme case of the probabilistic target encoding, the analysis corresponds to the cross-entropy error function with one-hot encoding. The statistical analyses conducted to derive the optimal solutions provide considerable insights, including the interval of target values for the Bayes classifier.*

Keywords: Cross-entropy Loss Function; Probabilistic Target Encoding; Optimal Solution of Output Values; Softmax Function; Neural Networks

1. Introduction

Pattern classifiers can be implemented based on the posteriori probabilities that an input sample belongs to a certain class, which provides the Bayes classifier [1]. The Bayes classifier essentially requires estimating the probability distribution or the p.d.f.(probability density function) of samples, which is a very difficult task. Parzen's window provides a method to estimate the p.d.f. of samples by locating the window function at each sample. However, the Parzen's window method requires a sufficient number of samples for accurate estimation of the p.d.f. [2].

Alternatively, we can implement pattern classifiers with a discriminant function, which provides the class boundary of patterns without estimating the p.d.f. or probability distribution of pattern samples. The decision rule of classification is to select the class with the maximal discriminant value, which corresponds to the confidence degree that an input sample belongs to a certain class [3]. In real environments with finite training samples, the discriminant function approach attains better performance than the posteriori probability approach. From this point of view, NNs (neural networks) are trained based on discriminant functions [4].

Conventionally, NNs are trained to minimize MSE (mean-squared error) function between desired and actual output node values, which correspond to the discriminant values [5]. Additionally, there is a variant of MSE that add noise to the desired values of output nodes to anticipate performance improvement [6]. To suppress the large amount of weight updating caused by outliers of training data, MLS (mean-log square) error function was proposed by Liano [7]. The binary CE (cross-entropy) error function [8] can accelerate the learning convergence of neural network classifiers, and the n-th order extension of binary CE [9, 10] attains better classification performance than MSE and binary CE error functions. Contrary to the above error functions, which are minimized during training, Hampshire and Waibel proposed CFM (classification figure of merit) function to be maximized for training NNs [11].

When training neural network classifiers with the above discriminant functions, we need to assign desired values of neural network outputs. The most popular method is one-hot encoding of the desired output vector, which contains all zeros except for a single 1 at the index corresponding to a class. Assuming that desired values of NNs are encoded with the one-hot vector, learning of NNs based on the above various functions had been analyzed from a statistical perspective under certain regularity conditions [12-15].

DNNs (deep neural networks) is a breakthrough of NNs for real world applications. In classification applications, there are three key substitutions between two-layer NNs and DNNs. The sigmoid activation function of neural network output is replaced with the softmax function in DNNs [16]. The second is the use of ReLU(Rectified Linear Unit) activation function for hidden layer of DNNs [17]. The third is the use of CE error function as a training criterion of DNNs [16]. Additionally, probabilistic target encoding has been proposed to alleviate the overfitting of DNNs to training samples [18].

In this paper, we analyze DNNs from a statistical perspective to provide considerable insights into the properties and advantages of the learning method based on the cross-entropy error function with the probabilistic target encoding. In section 2, we briefly review the one-hot encoding and the probabilistic target encoding. In section 3, we derive the optimal solution of output nodes to minimize the CE error function with the probabilistic target encoding. Additionally, we discuss considerable insights including the interval of target values with the probabilistic target encoding. Finally, section 4 concludes this paper.

2. One-Hot Encoding vs. Probabilistic Target Encoding

In classifications, we usually train DNNs to minimize the CE error function defined by

$$E_{CE} = -\sum_{k=1}^{M} t_k \ log \ y_k, \tag{1}$$

where M is the number of output nodes, y_k denotes the k-th output node's value, and t_k is its desired value [16]. We adopt the softmax activation function for output nodes of DNNs, given by

$$y_k = \frac{e^{\hat{y}_k}}{\sum_{j=1}^{M} e^{\hat{y}_j}} \ (k = 1,2,\dots,M) \tag{2}$$

where $\hat{y}_j$ is the weighted sum or net input to y_j [16].

Let $\boldsymbol{x}$ be an input vector and $\boldsymbol{t} = [\ t_1, t_2, \dots, t_M]^T$ be the target vector of output values corresponding to an input $\boldsymbol{x}$. In the scheme of one-hot encoding for classification applications of DNNs, the target vector is coded as follows:

$$t_k = \begin{cases} 1, & \text{if } \boldsymbol{x} \text{ originates from class } k \\ 0, & \text{otherwise.} \end{cases} \tag{3}$$

The desired values 1 or 0 in (3) are the extreme values of the softmax function. Thus, continuous updating of neural networks parameters to decrease the CE error function can cause neural networks to overfit to training samples, consequently degrading the classification performance to test samples [9].

After successful training of neural networks in the limit that the number of training samples goes to infinity, the neural network output y_k can be interpreted as a posterior probability that an input vector belongs to class k [12-15]. Therefore, it is better to use a posterior probabilistic value instead of the one-hot encoding. In this sense, a probabilistic encoding of the target vector was proposed as follows [18]:

$$t_k = \begin{cases} q_k(\boldsymbol{x}), & \text{if } \boldsymbol{x} \text{ originates from class } k \\ \dfrac{1 - q_k(\boldsymbol{x})}{M-1}, & \text{otherwise.} \end{cases} \tag{4}$$

Here, $q_k(\boldsymbol{x})$ is a real value between 0 and 1. Since it is very difficult to estimate the posterior probability that an input vector belongs to class k, we use a real value between 0 and 1 as a substitution of the posterior probability. Also, $q_k(\boldsymbol{x}) = 1$ corresponds to the one-hot encoding. That is, the one-hot encoding is the extreme case of the probabilistic target encoding. Unlike other regularization methods such as drop-out [19, 20], weight decay [21], and penalizing confident output distributions [22], the probabilistic target encoding method achieves the effect of preventing overfitting with a simple target encoding scheme, without any additional burden on the training algorithm of neural networks [18]. The effectiveness of the probabilistic target encoding was demonstrated in [18] through simulations of multi-layer perceptrons and convolutional neural networks for various classification problems such as handwritten-digit recognition, isolated-word recognition, image classification, and object recognition tasks. The simulation results showed that the probabilistic target encoding is superior to one-hot encoding as it prevents the overfitting of neural networks to training samples.

3. Statistical Analyses of Cross-Entropy Error Function

Let $Q_k(\boldsymbol{x})$ denote the posterior probability that $\boldsymbol{x}$ originates from class k. In the limit that the number of samples goes infinity, the minimizer of CE (1) with the probabilistic target encoding converges towards the minimizer of the function

$$E\{E_{CE}\} = -\int \sum_{k=1}^{M}[Q_k(\boldsymbol{x})q_k(\boldsymbol{x}) + (1 - Q_k(\boldsymbol{x}))\frac{1-q_k(x)}{M-1}]log\ \ y_k f(\boldsymbol{x})d(\boldsymbol{x}), \tag{5}$$

where $E\{E_{CE}\}$ is the expectation operator and $f(\boldsymbol{x})$ is the p.d.f. of input samples. Let us seek the function $\boldsymbol{b} = [\, b_1, b_2, \ldots, b_M]^T$ minimizing the criterion (5) in the space of all functions taking values in (-1, +1). For fixed $Q_k(\boldsymbol{x})$, the optimal solution $b_i(\boldsymbol{x})$ $(i = 1,2, \ldots, M)$ can be derived by

$$\frac{\partial \sum_{k=1}^{M}[Q_k(\boldsymbol{x})q_k(\boldsymbol{x})+\left(1-Q_k(\boldsymbol{x})\right)\frac{1-q_k(x)}{M-1}]log\ \ y_k}{\partial \hat{y}_i} = 0. \tag{6}$$

By substituting $\frac{\partial y_k}{\partial \hat{y}_i} = y_i(\delta_{ki} - y_k)$,

$$\sum_{k=1}^{M}[Q_k(\boldsymbol{x})q_k(\boldsymbol{x}) + (1 - Q_k(\boldsymbol{x}))\frac{1-q_k(x)}{M-1}]\ \ \frac{1}{y_k}y_i(\delta_{ki} - y_k) = 0 \tag{7}$$

Above equation can be reorganized as follows:

$$\left[Q_i(\boldsymbol{x})q_i(\boldsymbol{x}) + (1 - Q_i(\boldsymbol{x}))\frac{1-q_i(x)}{M-1}\right](1 - y_i) + \sum_{k\neq i}\left[Q_k(\boldsymbol{x})q_k(\boldsymbol{x}) + (1 - Q_k(\boldsymbol{x}))\frac{1-q_k(x)}{M-1}\right](-y_i) = 0 \tag{8}$$

Thus, we can get the optimal solution $b_i(\boldsymbol{x})$ which corresponds to y_i in $\frac{\partial E\{E_{CE}\}}{\partial \hat{y}_i} = 0$

$$b_i(\boldsymbol{x}) = \frac{Q_i(\boldsymbol{x})q_i(\boldsymbol{x})+\left(1-Q_i(\boldsymbol{x})\right)\frac{1-q_i(x)}{M-1}}{Q_i(\boldsymbol{x})q_i(\boldsymbol{x})+\left(1-Q_i(\boldsymbol{x})\right)\frac{1-q_i(x)}{M-1}+\sum_{k\neq i}\left[Q_k(\boldsymbol{x})q_k(\boldsymbol{x})+\left(1-Q_k(\boldsymbol{x})\right)\frac{1-q_k(x)}{M-1}\right]}. \tag{9}$$

Let's consider the two-class classification case, where M=2 and $1 - Q_i(\boldsymbol{x}) = Q_j(\boldsymbol{x})$. Simplifying (9) gives

$$b_i(\boldsymbol{x}) = \frac{Q_i(\boldsymbol{x})q_i(\boldsymbol{x})+\left(1-Q_i(\boldsymbol{x})\right)(1-q_i(\boldsymbol{x}))}{Q_i(\boldsymbol{x})[1+q_i(\boldsymbol{x})-q_j(\boldsymbol{x})]+\left(1-Q_i(\boldsymbol{x})\right)(1-q_i(\boldsymbol{x})+q_j(\boldsymbol{x}))}\ . \tag{10}$$

If $q_i(\boldsymbol{x}) = 1$ and $q_j(\boldsymbol{x}) = 1$, which corresponds to the one-hot encoding, simplifying (10) results in

$$b_i(\boldsymbol{x}) = \frac{Q_i(\boldsymbol{x})}{Q_i(\boldsymbol{x})+\left(1-Q_i(\boldsymbol{x})\right)} = Q_i(\boldsymbol{x}). \tag{11}$$

Hence, the optimal solution of output nodes trained with the CE error function under the one-hot encoding scheme gives the Bayes classifier by the decision of maximum output value. Considering the case of $q_i(\boldsymbol{x}) = q_j(\boldsymbol{x})$, (10) is simplified to

$$b_i(\boldsymbol{x}) = Q_i(\boldsymbol{x})\left(2q_i(\boldsymbol{x}) - 1\right) + 1 - q_i(\boldsymbol{x}). \tag{12}$$

From Figure 1 which is the plot of $b_i(\boldsymbol{x})$ vs. $Q_i(\boldsymbol{x})$, the target value should be $0.5 < q_i(\boldsymbol{x}) \leq 1$ so that $b_i(\boldsymbol{x})$ is the strictly increasing function of $Q_i(\boldsymbol{x})$. If $0 < q_i(\boldsymbol{x}) \leq 0.5$, $b_i(\boldsymbol{x})$ is not strictly increasing and NNs cannot be the Bayes classifier.

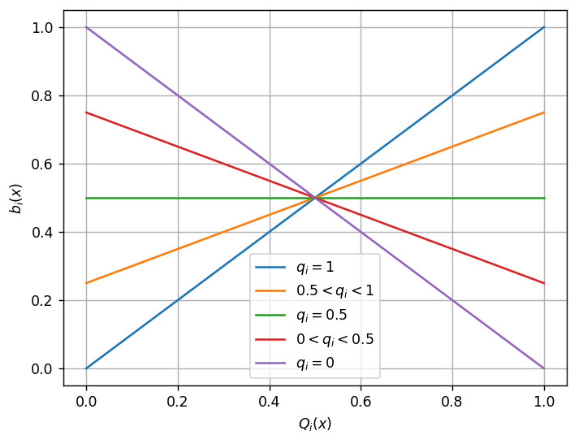

Figure 1. The optimal solution of output node for minimizing the expectation of CE error function using the probabilistic target encoding in the two-class classification case $(M = 2)$. $b_i(x)$ denotes the optimal solution of the ith output node and $Q_i(x)$ denotes the posterior probability that x originates from class i

Next, let's consider the case of M>2. If $q_i(\boldsymbol{x}) = q_k(\boldsymbol{x}) = \alpha$ in (9), we can get

$$b_i(\boldsymbol{x}) = \frac{Q_i(\boldsymbol{x})\alpha + \left(1-Q_i(\boldsymbol{x})\right)\frac{1-\alpha}{M-1}}{Q_i(\boldsymbol{x})\alpha + \left(1-Q_i(\boldsymbol{x})\right)\left(\frac{1-\alpha}{M-1}+\alpha\right) + \frac{1-\alpha}{M-1}\sum_{k\neq i}\left(1-Q_k(\boldsymbol{x})\right)}. \quad (13)$$

By substituting $\sum_{k=1}^{M}\left(1-Q_i(\boldsymbol{x})\right) = M-1$, (13) is rearranged into

$$b_i(\boldsymbol{x}) = \left(\alpha - \frac{1-\alpha}{M-1}\right)Q_i(\boldsymbol{x}) + \frac{1-\alpha}{M-1}. \quad (14)$$

Here, $\alpha = 1$ corresponds to the one-hot encoding and (13) is simplified to $b_i(\boldsymbol{x}) = Q_i(\boldsymbol{x})$, which is the same with the two-class classification case. Additionally, we can consider special cases of $\alpha = \frac{1}{M}$ and $\alpha = 0$ which are summarized as follows:

$$b_i(\boldsymbol{x}) = \begin{cases} Q_i(\boldsymbol{x}), & \text{if } \alpha = 1 \\ \frac{1}{M}, & \text{if } \alpha = \frac{1}{M} \\ -\frac{1}{M-1}Q_i(\boldsymbol{x}) + \frac{1}{M-1}, & \text{if } \alpha = 0. \end{cases} \quad (15)$$

If $\alpha = \frac{1}{M}$, $b_i(\boldsymbol{x})$ has a constant value of $\frac{1}{M}$, which cannot be a Bayes classifier. When $\alpha = 0$, $b_i(\boldsymbol{x})$ is a decreasing function of $Q_i(\boldsymbol{x})$, and this is also not a Bayse classifier. Thus, the condition of Bayes classifier is $\frac{1}{M} < \alpha \leq 1$, which guarantees that $b_i(\boldsymbol{x})$ is a strictly increasing function of $Q_i(\boldsymbol{x})$. Figure 2 clarifies these arguments. Additionally, the probabilistic target encoding can be interpreted as a weighted mixture of the one-hot encoding targets and a uniform distribution [23]. There is a possibility that $q_i(\boldsymbol{x}) \neq q_k(\boldsymbol{x})$ for performance improvement. However, analyzing the case of $q_i(\boldsymbol{x}) \neq q_k(\boldsymbol{x})$ is another challenging subject because of complexity.

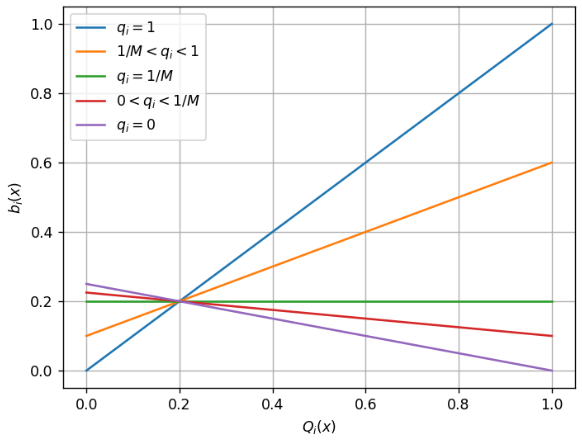

Figure 2. The optimal solution of output node for minimizing the expectation of CE error function using the probabilistic target encoding in the multi-class classification case. Here, $b_i(x)$ denotes the optimal solution of the ith output node, $Q_i(x)$ denotes the posterior probability that x originates from class i, and the num of class $M = 5$

4. Conclusions

In classification applications, we usually train deep neural networks using the cross-entropy error function. Conducting statistical analyses of learning procedure is strongly recommended to determine what neural networks eventually learn. From this point of view, this paper analyzed the cross-entropy error function with statistical perspective within the scheme of the probabilistic target encoding. Firstly, this paper has demonstrated a strong relationship between the optimal solution and Bayes probability. Specifically, Bayes probabilities are estimated by minimizing the cross-entropy error function with the probabilistic target encoding. Secondly, the analysis results indicate that the target values should lie between 1 and the reciprocal of the number of classes, ensuring that the classifier must be a Bayes classifier. Additionally, in extreme cases, the analysis corresponds to the cross-entropy error function under the one-hot encoding.

Conflicts of Interest: The author declares no conflict of interest.

References

[1] K. Fukunaga and D. Kessel, "Nonparametric Bayes error estimation using unclassified samples," IEEE Trans. Inf. Theory, vol. 19, pp. 434-439, Jul. 1973, doi: https://doi.org/10.1109/TIT.1973.1055049.

[2] E. Parzen, "On estimation of a probability density function and mode," Ann. Math. Statist. vol. 33, no. 3, pp. 1065-1076, Sep. 1962, doi: https://doi.org/10.1214/aoms/1177704472.

[3] W. J. Park and R. M. Kil, "Pattern Classification with Class Probability Output Network," IEEE Trans. Neural Network, vol. 20, pp. 1659-1673, Oct. 2009, doi: https://doi.org/10.1109/TNN.2009.2029103.

[4] J. B. Hamshire II and B. Pearlmutter, "Equivalence Proofs for Multilayer Perceptron Classifier and the Bayesian Discriminant Function," in Proc. 1990 Connectionist Model Summer School, pp. 159-172. 1990, doi: https://doi.org/10.1016/B978-1-4832-1448-1.50023-8.

[5] D. E. Rumelhart and J. L. McClelland, Parallel Distributed Processing, MIT Press, Cambridge, MA, 1986, doi: https://doi.org/10.7551/mitpress/5236.001.0001.

[6] C. Wang and J. C. Principe, "Training Neural Networks with Additive Noise in the Desired Signal," IEEE Trans. Neural Networks, vol. 10, pp. 1511-1517, Nov. 1999, doi: https://doi.org/10.1109/72.809097.

[7] K. Liano, "Robust Error Measure for Supervised Neural Network Learning with Outliers," IEEE Trans. Neural Networks, vol. 7, pp. 246-250, Jan. 1996, doi: https://doi.org/10.1109/72.478411.

[8] A. van Ooyen and B. Nienhuis, "Improving the Convergence of the Backpropagation Algorithm," Neural Networks, vol. 4, pp. 465-471, 1992, doi: https://doi.org/10.1016/0893-6080(92)90008-7.

[9] S. H. Oh, "Improving the error back-propagation algorithm with a modified error function," IEEE Trans. Neural Networks, vol. 8, pp. 799-803, May. 1997, doi: https://doi.org/10.1109/72.572117.

[10] S. H. Oh, "Error Back-Propagation Algorithm for Classification of Imbalanced Data," Neurocomputing, vol. 74, pp. 1058-1061, 2011, doi: https://doi.org/10.1016/j.neucom.2010.11.024.

[11] J. B. Hampshire and A. H. Waibel, "A Novel Objective Function for Improved Phoneme Recognition Using Time-Delay Networks," IEEE Trans. Neural Networks, vol. 1, pp. 216-218, June 1990, doi: 10.1109/72.80233.

[12] H. White, "Learning in Artificial Neural Networks: A Statistical Perspective," Neural computation, vol. 1, pp. 425-464, Dec. 1989, doi: https://doi.org/10.1162/neco.1989.1.4.425.

[13] M. D. Richard and R. P. Lippmann, "Neural Network Classifiers Estimate Bayesian a Posteriori Probabilities," Neural Computation, vol. 3, no. 4, pp. 461-483, Dec. 1991, doi: https://doi.org/10.1162/neco.1991.3.4.461.

[14] S. H. Oh, "A Statistical Perspective of Neural Networks for Imbalanced Data Problems," International Journal of Contents, vol. 7, no. 3, pp. 1-5, Sep. 2011, doi: https://doi.org/10.5392/ijoc.2011.7.3.001.

[15] S. H. Oh, "Statistical Analyses of Various Error Functions for Pattern Classifiers," in Proc. Convergence on Hybrid Information Technology, Daejon, Korea, vol. 206, pp. 129-133, Sep. 22-24, 2011, doi: https://doi.org/10.1007/978-3-642-24106-2_17.

[16] S. Horiguchi, D. Ikami, and K. Aizawa, "Significance of Softmax-Based Features in Comparison to Distance Metric Learning-Based Features," IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 42, no. 5, pp. 1279-1285, May. 2020, doi: https://doi.org/10.1109/TPAMI.2019.2911075.

[17] X. Glorot, A. Bordes, and Y. Bengio, "Deep Sparse Rectifier Neural Networks," in Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics, PMLR, vol. 15, pp. 315-323, 2011.

[18] S. H. Oh, "Probabilistic Target Encoding of Neural Networks with Softmax Output Nodes," International Journal of Contents, vol. 18, no. 4, pp. 102-108, Dec. 2022, doi: https://doi.org/10.5392/IJoC.2022.18.4.102.

[19] K. Baldi and P. Sadowski, "The Dropout Learning Algorithm," Artificial Intelligence, vol. 210, pp. 78-122, 2014, doi: https://doi.org/10.1016/j.artint.2014.02.004.

[20] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A Simple Way to Prevent Neural Networks from Overfitting," Journal of Machine Learning researches, vol. 15, pp. 1929-1958, 2014.

[21] G. Gnecco and M. Sanguineti, "The Weight-Decay Technique in Learning from Data: An Optimization Point of View," Computational Management Science, vol. 6, pp. 53-79, 2009, doi: https://doi.org/10.1007/s10287-008-0072-5.

[22] G. Pereyra et al., "Regularizing Neural Networks by Penalizing Confident Output Distributions," arXiv:1701.06548.

[23] R. Müller, S. Kornblith, and G. Hinton, "When Does Label Smoothing Helps?," arXiv.1906.02629.