Discovery of Relevant Weights by Minimizing Cross-validation Error

Kazumi Saito[1] and Ryohei Nakano[2]

[1] NTT Communication Science Laboratories
2-4 Hikaridai, Seika, Soraku, Kyoto 619-0237 Japan
saito@cslab.kecl.ntt.co.jp
[2] Nagoya Institute of Technology
Gokiso-cho, Showa-ku, Nagoya 466-8555 Japan
nakano@ics.nitech.ac.jp

Abstract. In order to discover relevant weights of neural networks, this paper proposes a novel method to learn a distinct squared penalty factor for each weight as a minimization problem over the cross-validation error. Experiments showed that the proposed method works well in discovering a polynomial-type law even from data containing irrelevant variables and a small amount of noise.

1 Introduction

Neural networks can be utilized as a core technique in some KDD (Knowledge Discovery and Data mining) applications such as scientific discovery [2, 1]. One important research subject of neural networks is to improve the generalization performance. Here the generalization means the performance on new data. It is widely known that adding some penalty term to a standard training error term can lead to significant improvements in network generalization. As for squared penalty, a single penalty factor is often conveniently used. If we can develop a method that automatically adjusts a distinct penalty factor for each weight, several advantages can be expected, i.e., the generalization performance will be still more improved; the readability of discovered laws will be improved; such a squared penalty term is consistent with any linear scaling of variables; and suitable penalty factors can be determined without inaccurate estimation.

2 Optimal Penalty Factor Calculation

Let $(x_1, \cdots, x_K, y)$ be a vector of variables describing each example, where x_k is a numeric or nominal explanatory variable and y is a numeric target variable. Here we assume that each nominal explanatory variable is described as a *dummy variable*. As a class of numeric formula $y(\boldsymbol{x}; \boldsymbol{\Theta})$, we consider a generalized polynomial expressed by

$$y(\boldsymbol{x}; \boldsymbol{\Theta}) = w_0 + \sum_{j=1}^{J} w_j \prod_{k=1}^{K} x_k^{w_{jk}} \;=\; w_0 + \sum_{j=1}^{J} w_j \exp\left(\sum_{k=1}^{K} w_{jk} \ln x_k\right), \quad (1)$$

where each parameter w_j or w_{jk} is an unknown real number, and J is an unknown integer corresponding to the number of terms. $\boldsymbol{\Theta}$ is an M-dimensional parameter vector constructed by arranging parameters $w_j, j = 0, \cdots, J$, and $w_{jk}, j = 1, \cdots, J, k = 1, \cdots, K$. Let $D = \{(\boldsymbol{x}^\mu, y^\mu), \mu = 1, \cdots, N\}$ be a set of training examples, where N is the number of examples. Here we assume that each training example $(\boldsymbol{x}^\mu, y^\mu)$ is independent and identically distributed. Now, our ultimate goal of the law discovery is defined as a problem of minimizing the generalization error, that is, to find the optimal estimator $\boldsymbol{\Theta}^*$ that minimizes

$$\mathcal{G}(\boldsymbol{\Theta}^*) = E_D E_T \left(y^\nu - y(\boldsymbol{x}^\nu; \boldsymbol{\Theta}^*(D))\right)^2, \tag{2}$$

where $T = (\boldsymbol{x}^\nu, y^\nu)$ denotes test data independent of the training data D. The *least-squares estimate* of $\boldsymbol{\Theta}^*$, denoted by $\widehat{\boldsymbol{\Theta}}$, minimizes the error sum of squares

$$\mathcal{E}_1(\boldsymbol{\Theta}) = \frac{1}{2} \sum_{\mu=1}^{N} \left(y^\mu - y(\boldsymbol{x}^\mu; \boldsymbol{\Theta})\right)^2. \tag{3}$$

However, this estimation is likely to over-fit to the training data; thus, we cannot usually obtain good results in terms of the generalization performance.

As we have already mentioned, it is widely known that adding some penalty term to Eq. (3) can lead to significant improvements in network generalization. Here a simple penalized target function using a single factor is given as below.

$$\mathcal{E}_2(\boldsymbol{\Theta}) = \mathcal{E}_1(\boldsymbol{\Theta}) + \frac{1}{2} \exp(\lambda) \sum_{m=1}^{M} \theta_m^2, \tag{4}$$

where $\exp(\lambda)$ is a penalty factor and $\theta_m \in \boldsymbol{\Theta}$. Here since the penalty factor must be non-negative, we adopted $\exp(\lambda)$, instead of a standard parameterization λ.

To improve both the generalization performance and the readability, we consider a distinct penalty factor for each weight. Let $\boldsymbol{\lambda}$ be an M-dimensional vector $(\lambda_1, \cdots, \lambda_M)^T$, and $\boldsymbol{\Lambda}$ be an M-dimensional diagonal matrix whose diagonal elements are defined by $\Lambda_{mm} = \exp(\lambda_m)$ for $m = 1, \cdots, M$, where $\boldsymbol{a}^T$ denotes a transposed vector of $\boldsymbol{a}$. Then, the discovery of laws subject to Eq. (1) can be defined as the following learning problem in neural networks. That is, the problem is to find the $\boldsymbol{\Theta}$ that minimizes the following objective function for weights

$$\mathcal{E}(\boldsymbol{\Theta}) = \mathcal{E}_1(\boldsymbol{\Theta}) + \frac{1}{2} \boldsymbol{\Theta}^T \boldsymbol{\Lambda} \boldsymbol{\Theta}. \tag{5}$$

Now, we introduce an objective function for penalty factors derived from the procedure of *cross-validation*, and propose *MCV (Minimum Cross-Validation) regularizer*. The procedure of cross-validation divides the data D at random into S distinct segments ($G_s,\ s = 1, \cdots, S$), and uses $S-1$ segments for training, and uses the remaining one for the test. This process is repeated S times by changing the remaining segment, and the generalization performance is evaluated by using the following MSE (mean squared error) over all S test results.

$$MSE_{CV} = \frac{1}{N} \sum_{s=1}^{S} \sum_{\nu \in G_s} \left(y^\nu - y(\boldsymbol{x}^\nu; \widehat{\boldsymbol{\Theta}}_s)\right)^2. \tag{6}$$

Here $\widehat{\boldsymbol{\Theta}}_s$ denotes the optimal weights obtained by minimizing the following objective function for weights

$$\mathcal{E}_s(\boldsymbol{\Theta}_s) = \frac{1}{2} \sum_{\mu \notin G_s} (y^\mu - y(\boldsymbol{x}^\mu; \boldsymbol{\Theta}_s))^2 + \frac{1}{2} \boldsymbol{\Theta}_s^T \boldsymbol{\Lambda} \boldsymbol{\Theta}_s. \tag{7}$$

The extreme case of $S = N$ is known as the *leave-one-out* method, which is often used for a small size of data. Note that Eq. (6) is regarded as a reasonable approximation to Eq. (2) for a given data set D. According to the *implicit function theorem*, since $\widehat{\boldsymbol{\Theta}}_s$ can be regarded as a vector consisting of implicit functions of $\boldsymbol{\lambda}$, Eq. (6) can be defined as the objective function for penalty factors. Thus, we can calculate $\widehat{\boldsymbol{\lambda}}$ which minimizes Eq. (6). Then, by using $\widehat{\boldsymbol{\lambda}}$, we can calculate $\widehat{\boldsymbol{\Theta}}$ which minimizes Eq. (5). Finally, $\widehat{\boldsymbol{\Theta}}$ is adopted as the final weight vector of the discovered law.

3 Evaluation by Experiments

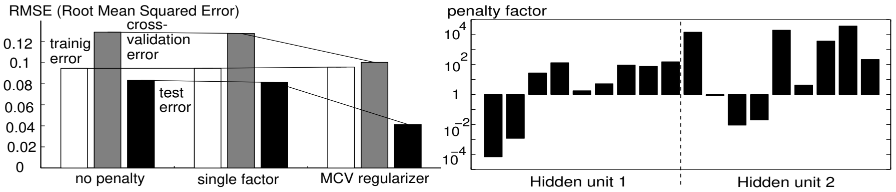
We consider an artificial law (function) described by

$$y = 2 + 3x_1^{+1} x_2^{-0.02} + 4x_3^{-1} x_4^{+0.02} \tag{8}$$

where we have 9 numeric explanatory variables. Clearly, variables $x_5, \cdots, x_9$ are irrelevant to Eq. (8). Each example is generated as follows: each value of numeric variables $x_1, \cdots, x_9$ is randomly generated in the range of $(0, 1)$, and we get the corresponding value of y by calculating Eq. (8) and adding Gaussian noise with a mean of 0 and a standard deviation of 0.1. The number of examples is set to 200 ($N = 200$). Before the analysis, the following scaling was applied to the variables: $\widetilde{y} = (y - mean(y))/std(y)$, and $\ln \widetilde{x}_k = \ln x_k - mean(\ln x_k),\ k = 1, \cdots, 9$.

In the experiments, the initial values for the weights w_{jk} were independently generated according to a normal distribution with a mean of 0 and a standard deviation of 1; the initial values for the weights w_j were set to 0. The initial values for the penalty factors $\boldsymbol{\lambda}$ were set to $\mathbf{0}$, i.e., $\boldsymbol{\Lambda}$ was set to the identical matrix. The iteration was terminated when the gradient vector was sufficiently small, i.e., $\max_m\{\|\partial/\partial\theta_m\ \mathcal{E}(\boldsymbol{\Theta})\|\} < 10^{-6}$ for learning over $\boldsymbol{\Theta}$; $\max_m\{\|\partial/\partial\lambda_m\ MSE_{CV}(\boldsymbol{\lambda})\|\} < 10^{-6}$ for learning over $\boldsymbol{\lambda}$.

MCV regularizer was compared with two conventional methods, no-penalty method and single-factor method, where the objective functions of these conventional methods are Eq. (3) and Eq. (4), respectively. Figure 1(a) shows the learning results of these three methods, where the RMSE (root mean squared error) was used for evaluation; the number of hidden units J was fixed at the correct number 2; the cross-validation error was calculated by using the leave-one-out method, i.e., $S = N$; and the generalization performance was measured by using a set of noise-free $10,000$ test examples generated independently to the training examples. This figure shows that the RMSE for the training data was almost the same for each method; both the RMSE for the cross-validation and



(a) performance of three methods

(b) learning result of penalty factors

Fig. 1. Experimental results of artificial data

the RMSE for the test data were clearly decreased by using MCV regularizer; and the performance of the single factor method was almost comparable to those of the no penalty method.

An example of the laws discovered by the no penalty method was as follows:

$$
\begin{aligned}
y \;=\; & 2.0306 \\
&+2.9791 x_1^{+1.0024} x_2^{-0.0203} x_3^{+0.0035} x_4^{-0.0029} x_5^{+0.0073} x_6^{+0.0056} x_7^{+0.0010} x_8^{+0.0022} x_9^{-0.0036} \\
&+3.9993 x_1^{+0.0008} x_2^{+0.0004} x_3^{-1.0003} x_4^{+0.0201} x_5^{-0.0005} x_6^{-0.0011} x_7^{-0.0002} x_8^{-0.0002} x_9^{+0.0011}
\end{aligned}
$$

where the weight values were rounded off to the fourth decimal place. Note that these weight values were transformed so as to correspond to the original scale of variables. Although a law almost equivalent to the true one was found, it is difficult to select only the relevant weights from this result. While an example of the laws discovered by MCV regularizer was as follows:

$$
\begin{aligned}
y \;=\; & 2.0118 \\
&+2.9792 x_1^{+0.9941} x_2^{-0.0190} x_3^{-0.0000} x_4^{-0.0000} x_5^{+0.0019} x_6^{+0.0007} x_7^{+0.0000} x_8^{+0.0000} x_9^{+0.0000} \\
&+3.9987 x_1^{+0.0000} x_2^{+0.0001} x_3^{-0.9999} x_4^{+0.0197} x_5^{-0.0000} x_6^{-0.0006} x_7^{-0.0000} x_8^{-0.0000} x_9^{+0.0003}
\end{aligned}
$$

Clearly, the irrelevant weight values were greatly suppressed.

Figure 1(b) shows the learning result of the penalty factors. This figure indicates that only the penalty factors for the relevant weights became small enough, i.e., we can easily select only the relevant weights. Therefore, it was shown that the MCV regularizer simultaneously improves the generalization performance and readability, without care of variable scaling and a candidate determination for the penalty factors.

References

1. R. Nakano and K. Saito. Discovery of a set of nominally conditioned polynomials. In *Proc. 2nd Int. Conf. on Discovery Science, LNAI 1721*, pages 287–298, 1999.
2. K. Saito and R. Nakano. Law discovery using neural networks. In *Proc. 15th Int. Joint Conf. on Artificial Intelligence*, pages 1078–1083, 1997.