The 32nd Annual Conference of the Japanese Society for Artificial Intelligence, 2018

補助重み法によるニューラルネット上での特徴選択と過適応防止

Feature selection and over-adaptation prevention in neural networks using the auxiliary weight method

野田 陽 *1
NODA Akira

*1島津製作所 基盤技術研究所
Technology Research Laboratory, Shimadzu Corporation

In this study, we propose a method called auxiliary weight (AW) for neural networks in which each input value is weighted according to its contribution to the input dimension. AW is similar to Lasso regularization in the sense that it can extract features; however, AW is faster than Lasso in processing data that contains a several contributing dimensions and massive non-contributing dimensions, such as the data of medical mass spectrometry.

1. Introduction

In applications of discrimination using a chemical analyzer, such as in disease diagnosis using a mass spectrometer (MS), the input data generally contains a large number of dimensions and a high level of independence. Therefore, feature extraction is often performed by the partial differential of a neural network (NN) [**?**] or by Lasso regularization. In the field of medicinal chemistry, feature extraction is applied for biomarker candidate discovery. It is important not only to understand the reactions that occur in disease but also to prevent over-adaptation. In this study, we propose a novel method, which is called the auxiliary weight (AW*1) method in which each input dimension is multiplied by a weight based on its degree of contribution to the output of NN.

2. About Input Data

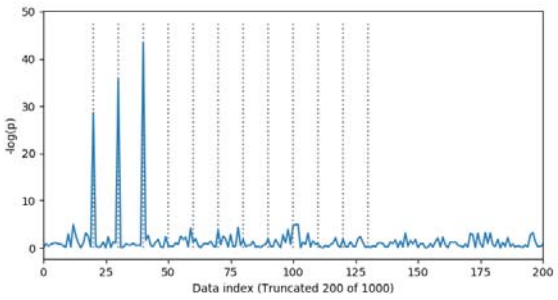
MS is an analytical device that is used to output an intensity value that correlates with the concentration of molecules having a certain mass-to-charge ratio (m/z). *2 Thus, in the obtained data, the concentration information regarding one or few molecular species is provided in a single dimension. Therefore, the independence of each dimension is observed to be high. The MS spectrum contains approximately 1000-100000 dimensions and one to several tens of these dimensions are observed to contribute to the diagnosis of a disease. In the diagnosis of several diseases, the discriminant function is nonlinear. Hence, discrimination is often performed using SVM(Support Vector Machine), and feature selection is often performed using SVMRFE(SVM recursive feature elimination) [**?**].

2.1 Simulation data

In this study, we use relatively simple simulated data. The input data is a 1000-dimensional vector of random numbers that depict a normal distribution. The training labels are provided on the basis of an equation, 1. Twelve of these dimensions of simulation data are contributing to the training label.

$$\begin{aligned} & Label = (Boolean)(y < 0) \\ y \quad = \quad & (x_1 + 0.5)(x_2 - 0.4)(x_3 + 0.3) \\ & +(x_4 + 0.25)(x_5 - 0.15)(x_6 + 0.05) \\ & +(x_7 + 0.3)(x_8 - 0.2)(x_9 + 0.1) \\ & +(x_{10} - 0.2)(x_{11} + 0.1)x_{12} \end{aligned} \tag{1}$$

Fig. 1 depicts the logarithmic p-value using the t-test for each dimension. Although twelve dimensions ($20, 30, \cdots, 130$ on the horizontal axis) contribute to the discrimination process, it can be observed that only three of them (indices 20, 30, and 40) can be identified using the t-test.



The dashed line indicates the contributing dimension.

Fig. 1: Log p-value of each dimension

3. Method

3.1 AW method

In the AW method, weights are applied to each dimension of the input data. The outputs of the AW layer are inputted to the NN, $f(x)$ (Fig.2). The AW method is similar to the Lasso method such that the weights of the dimensions that provide no contributions are zero; however,

連絡先: a-noda@shimadzu.co.jp

∗1 Code: `https://bitbucket.org/akira_you/awexperiment`.

∗2 MS-data is expressed as a vector in which the length covers the m/z range, whereas the elements denote the signal intensities at each value of m/z.

the AW method contains an additional loop to update each weight based on the degree of contribution. The degree of contribution, g_i, of each dimension is calculated using the norm of $\partial f(x_i)/\partial x_i$, which is obtained using a mini-batch. Further, g_i is normalized such that $|G| = 1$. Each weighting coefficient, a_i, of the AW layer is updated by $a_i* = g_i^u$, where u is the update rate parameter and is usually set to approximately 0.1. The AW layer and $f(x)$ are updated alternately. For the dimensions that depict large contributions, the input value is increased during the AW-update. Further, in the next batch, an NN-update of $f(x)$ is performed so that the contribution degree of the dimension becomes smaller. Finally, the contribution degrees of all the dimensions become equal to one. Thus, the AW-weights of the dimensions with no contributions become zero. AW can also be considered to be dynamic normalization. In several situations, medical MS data is normalized according to the variance, which may be caused by a disease (the target of interest in this technique) or may be caused due to the natural differences between various patients. Before conducting learning on the network, there is no information that which variance is caused by disease. In AW, data is normalized to the contribution degrees rather than to the variances. The contribution degree is defined using the NN and the learned parameters. Thus, the weights and NN must be updated alternately.

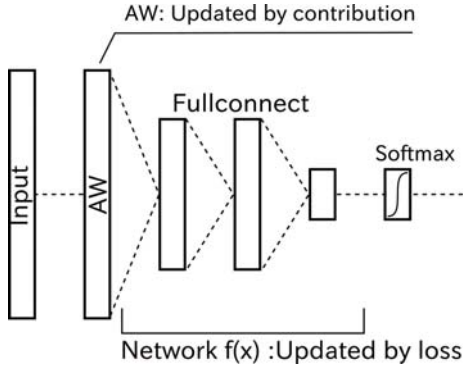


Fig. 2: The AW network model

3.2 Stochastic invalidator

The contributing dimensions can be extracted using the AW method. However, if there are massive random-number dimensions in the input, it is difficult to discern their contributions due to over-adaptation and contribution due to true discrimination. Several dimensions are required for over-adaptation by random values, the number of which is correlated with the numbers of the training data. However, in case of disease diagnosis and similar applications, only a few dimensions contribute because a disease may be caused due to a partially abnormal metabolism or because only a few chemical substances are affected by the condition.

Therefore, we intended to limit the dimensions that were contributed by the entanglement of multiple dimensions by implementing stochastic invalidation (SI) in the AW method. SI stochastically gates the output of the AW layer according to the value of a_i for each mini-batch. In this study, the value of a_i is compared with a random number, $N(0, 2.5\sigma_i)$, which was generated according to a normal distribution, where σ_i is the average of all the values of a_i. Further, a_i is replaced by zero if it is smaller than the random number. For example, consider a set of dimensions, $d_1, d_2, ..d_N$, with over-adaptation and a gate-open probability of $p_1, p_2, ..p_N$. The joint probability, $\prod p_i$, decreases exponentially with respect to the required number of dimensions, thereby sharply decreasing the learning opportunities of these dimensions.

Algorithm 1 AW update during the back-propagation of NN

while all epochs are incomplete **do**
  $G \leftarrow 0$;
  for i **in** range(B_a/B_n)
    //Normal back-propagation
    $x \leftarrow A \circ D[i]$;
    $y \leftarrow forward(x)$;
    $grad \leftarrow backword(\partial|Softmax(y) - L[i]|/\partial y))$;
    [update the NN parameters by grad]

    //Summing the degrees of contribution
    $G+ = |\partial f(x)/\partial x|$;
  end for
  $G/ = |G|$;
  $a_i* = g_i^u$;
end while

Where: B_a is the batch size for the AW, B_n is the batch size for the NN, D is the input data batch array, L contains the labels, a_i is the weight of the AW layer, A is is a vector of a_i u is the update rate, $\circ$ is the element-wise product, and grad represents the gradient for all variables.
Note:$\partial f(x)/\partial x$ can be obtained by $backword(1)$

Algorithm 2 Stochastic invalidator

$P \leftarrow 1$;
while all epochs are incomplete **do**
  for i **in** range(B_a/B_n)
    $x \leftarrow P \circ A \circ D[i]$;
    [NN update]
    [Summing the degrees of contribution]
  end for
  [Update A]
  $ave \leftarrow \sum_i a_i/N$
  $h_i \leftarrow N(0, ave * 2.5)$
  $p_i \leftarrow 0 \quad if(a_i < h_i) \quad else \quad 1$
end while

where P is the 0/1 filter for each input dimension, A is the weight for the AW, D is the input data batch array, and N is a random vector that is generated according to a normal distribution.

4. Experiment

Three analytical methods - no AW, with AW, and with AW and SI - are compared on fully connected NNs that depict the following grid-searched hyper-parameters:

- Number of middle layers: $(5, 6)$
- Dimension of the middle layer: $(100, 300)$
- Dimension of the first layer of the middle layers: $(1000, 2000)$
- Lasso: $(0, 1e-5, 1e-6)$

- Lasso (AW) :$(0, 1e-3, 1e-4, 1e-5)$
- Batch size: 800 / AW batch size: 2400

Lasso (AW) was applied to weights of AW when AW was valid and was applied to the first layer of the middle layers for the 'No AW' case. The number of training data was 27000, and the number of test data was 3000. Fig.3 depicts the accuracy that was observed during the three repeats of the experiment.

'AW+SI' provides high accuracy during the early epoch. Fig.4 depicts the results of feature extraction. These are the AW weights (or the weights of the L2 norms of the first layers). These observations illustrate that the use of AW alone or AW with SI results in the successful extraction of all twelve features, whereas the use of Lasso alone, without using AW, results in the extraction of only six features.

5. Conclusion and Future work

AW can be used with or without SI to perform feature extraction and prevent over-adaptation more effectively than the performance of Lasso when there are massive non-contributing dimensions. This technique will be useful for searching the biomarkers and for conducting diagnostics using medical MS.

In this study, only the simulated data was analyzed. However, in actual medical data, the number of data will be low. Therefore, the extracted features will not be very accurate. It is necessary to consider these differences while observing the extraction results of each cross-validation and conduct verification process based on the knowledge of human physiology.

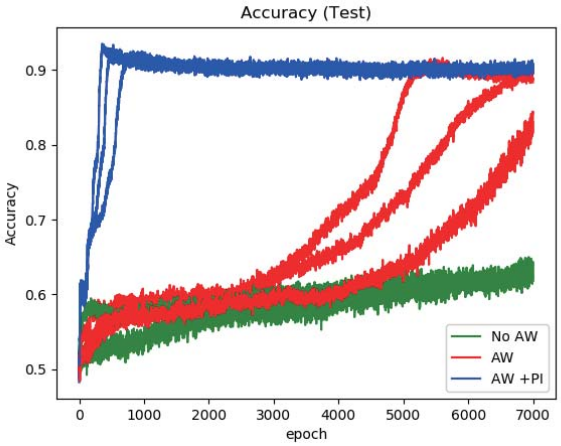


Fig. 3: Accuracy (Test)
Parameters (#of Mid, Mid.Order, 1stMid.Order, Lasso, LassoAW) are
No AW: (5, 300, 2000, 1e-6, 1e-5), (5, 100, 2000, 1e-5, 1e-3), (6, 100, 2000, 1e-5, 1e-4)
AW: (5, 300, 2000, 1e-6, 1e-4), (6, 100, 1000, 1e-5, 1e-4), (5, 300, 2000, 1e-6, 0)
AW+SI: (6, 300, 2000, 1e-5, 1e-3), (6, 300, 1000, 1e-5, 1e-3), (6, 300, 2000, 1e-5, 1e-3)

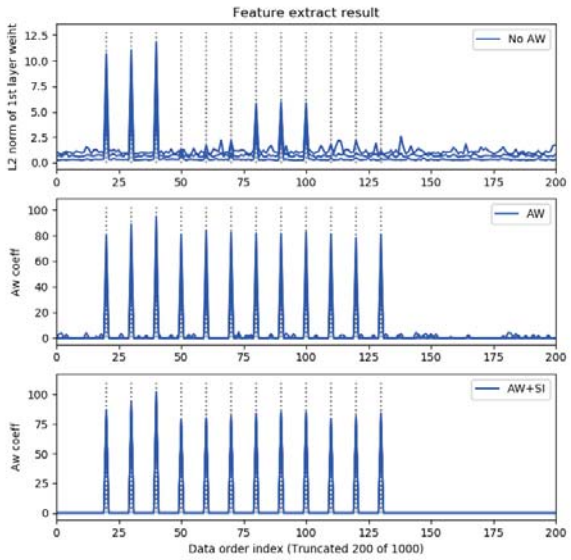


Fig. 4: Feature extraction result