Postprint of Parallel Ensemble Learning Method for Small Datasets Based on Resource Allocation Networks

Authors: Zhang Anguo, Zhang Shuxun, Zhu Wei, Li Xiumin, Huang Jinlong

Date: 2018-05-02T00:00:00+00:00

Abstract

In the field of machine learning, obtaining an algorithmic model with stable high computational accuracy on small-scale training datasets has long been a challenging and formidable problem. From the perspective of algorithmic models, a parallel ensemble learning method based on Resource Allocating Networks with Extended Kalman Filter is proposed. The ensemble system comprises multiple Resource Allocating Networks with Extended Kalman Filter (RANEKF), where the input of each RANEKF subnet is derived from the inputs of the original dataset modified by random weights. Through experimental comparison with ensemble learning algorithms composed of other neural networks, the proposed method is found to possess higher computational accuracy and stability on small training sets.

Full Text

Parallel Ensemble Learning Method Based on Resource Allocating Networks for Small Dataset

Zhang Anguo[1], Zhang Shuxun[2],[3], , Zhu Wei[1], Li Xiumin , Huang Jinlong ,* [1] Research Institute of Ruijie, Ruijie Networks Co., Ltd., Fuzhou 350002, China [2] Xinjiang Technical Institute of Physics & Chemistry, Chinese Academy of Sciences, Ürümqi 830011, China [3] University of Chinese Academy of Sciences, Beijing 100049, China Xinjiang Laboratory of Minority Speech & Language Information Processing, Ürümqi 830011, China College of Automation, Chongqing University, Chongqing 400044, China Yangtze Normal University, Chongqing 408100, China

Abstract: Designing a training model with stable computational performance and high accuracy on small-scale datasets has long been a difficult and challenging problem in machine learning. This paper proposes a parallel ensem-

ble learning algorithm based on Resource Allocating Networks with Extended Kalman Filter (RANEKF). The learning system comprises multiple RANEKF units, where each unit' s inputs are generated by modifying the original dataset inputs with random weights. Experimental comparisons with ensemble learning algorithms constructed from other neural networks demonstrate that the proposed method achieves higher computational accuracy and stability on small training sets.

Keywords: resource allocating network; parallel ensemble learning; incremental learning; extended Kalman filter

0 Introduction

In recent years, artificial intelligence, particularly machine learning, has garnered widespread attention from both academia and industry. Machine learning training refers to the dynamic adaptive adjustment of free parameters in an algorithmic model within a certain data environment, thereby enabling the model to exhibit new input-output computational behaviors—this constitutes the essence of "learning." Consequently, the quantity and quality of training data profoundly influence model performance, making data collection and processing critically important. However, in many practical engineering applications, data collection is extremely time-consuming and labor-intensive, making it difficult to acquire sufficient data volume or sample values that meet required precision levels. For instance, disease data and aero-engine fault data are particularly challenging to collect due to their inherently low occurrence rates. Nevertheless, even small datasets possess inherent value and learnability, much like how humans can learn new concepts without requiring massive amounts of sample data.

Currently, there is no explicit definition for "small-scale" datasets. Typically, adequacy of dataset size is evaluated from two perspectives: the relationship between data volume and feature dimensionality, and the relationship between data volume and noise level. If the data volume is not substantially larger than—or is even smaller than—the feature dimensionality, or if the noise level is relatively high while the data volume is not sufficiently large, these scenarios can be considered as having small data scale.

Against this backdrop of small datasets, how to leverage limited data to obtain a high-precision, stable machine learning model has become a research focus. Existing approaches to address model training on small datasets primarily fall into three categories: first, manipulating the training dataset itself through oversampling or undersampling techniques [1,2]; second, algorithmic approaches such as semi-supervised learning [3,4] and input feature preprocessing [5,6]; and third, learning strategy-based methods like transfer learning [7,8] and ensemble learning [9].

Ensemble learning methods based on weak learner integration can ensure good generalization capability for the entire model system while avoiding overfitting phenomena caused by small training sample sizes. Consequently, ensemble learn-

ing has become a popular research direction, particularly achieving numerous research 成果 when using neural networks as weak learners in ensemble units [10-14]. For these small-scale neural network ensemble units, network structure and model complexity significantly impact the final system' s computational accuracy. However, no definitive method currently exists to determine the optimal neural network scale for a given training set size that maximizes computational performance, generalization, and overfitting avoidance. Beyond empirical values, designing an optimal network model based on training sample quantity and complexity remains challenging. Therefore, incremental learning methods capable of dynamically adjusting model structure and parameters as the training set expands and complexity increases have been proposed [15-17].

1 Related Work

1.1 Ensemble Learning

Ensemble learning leverages collaboration among multiple weak learners to solve the same problem, with system output determined collectively by these weak learners' outputs on current samples. Ensemble learning can significantly improve computational accuracy and robustness of learning systems on small training sets. Currently, common ensemble learning methods include Boosting and Bagging [23].

The main idea of Boosting is to adjust the influence weight of each sample in subsequent training rounds based on the current round' s training error. Given a weak learner h and training dataset $< \mathbf{X}, \mathbf{Y} > = \{< \mathbf{x} , y >, < \mathbf{x} , y >, \cdots, < \mathbf{x} , y >\}$ with n training samples, each sample is initially assigned equal weight $1/n$. During R training rounds, samples that are misclassified receive larger weights, ensuring they receive more attention in subsequent training.

Bagging' s primary concept involves randomly extracting a certain amount of training data from the dataset $< \mathbf{X}, \mathbf{Y} > = \{< \mathbf{x} , y >, < \mathbf{x} , y >, \cdots, < \mathbf{x} , y >\}$ in each round to train all weak learners in the ensemble. The ensemble's final output is determined by voting among weak learners (for regression problems, by averaging all weak learners' outputs).

1.2 Resource Allocating Networks

Resource Allocating Network (RAN) is an incremental learning model whose hidden layer node count can dynamically increase according to training sample complexity, thereby achieving higher computational capability. Consequently, RAN has found widespread industrial applications, including thermal process object identification in thermal power unit control [18], text classification in natural language processing [19], Alzheimer' s disease detection in healthcare [20], and robot manipulation in industrial production [21]. In academia, improved RAN models have been continuously proposed, such as Minimal RAN (MRAN) that prevents rapid hidden node growth [21,22] and RAN with Extended Kalman

Filter (RANEKF) that incorporates noise and variance estimation [18,19]. The RANEKF method employs an extended Kalman filter instead of the least mean squares algorithm for adjusting network free parameters, thereby improving convergence performance and accelerating learning speed—advantageous for quickly obtaining a stable and usable network model on small datasets.

Literature [24] addresses the poor stability of single RAN networks and their susceptibility to sample quality and training order by proposing a parallel computing structure composed of multiple minimal RAN network units, where each RAN network is trained using only single-class (pattern) sample data. This method effectively avoids overfitting caused by excessively large network scale due to rapid hidden layer node growth. However, since only a small portion of sample data is used to train individual RAN networks, the total training data requirement is substantial, making it inapplicable to small-sample datasets. Moreover, its effectiveness significantly decreases for XOR-like problems and cannot be applied to regression fitting tasks.

Literature [25] proposes a RAN ensemble learning method based on Long-Term Memory (LTM) using the AdaBoost.M1 algorithm to compute input sample weights for the next RAN network based on current RAN network output errors. The authors demonstrate that this method achieves higher computational accuracy than single-classifier algorithms. However, since training subsequent RAN classifiers requires current RAN output errors, only a single RAN network can be trained at each time point, preventing parallel training of the entire RAN ensemble and consequently consuming substantial training time. Additionally, because RAN networks lack random initial values internally, this method cannot effectively increase diversity among individual RANs in the ensemble during early training stages, resulting in low training efficiency.

Literature [26] proposes a RAN network ensemble method based on Bagging technology. To ensure diversity among integrated RAN networks, each RAN network is trained using only $N-1$ class sample data (N being the total number of sample label categories). For each RAN sub-network, if the distance between the input sample and the network' s nearest node exceeds a certain threshold, it indicates that the input sample label does not belong to the network' s training label space. Selecting this threshold becomes crucial and highly dependent on the feature value distributions of both the $N-1$ classes used for training and the remaining one class. Consequently, threshold selection becomes extremely difficult and cumbersome, sometimes even impossible.

In summary, existing RAN network ensemble learning methods perform poorly on small dataset training problems, require excessively long training times, or suffer from low training efficiency. To address these issues, this paper proposes the RANEKF-PEL (Parallel Ensemble Learning) algorithm with the following key characteristics: (a) easy construction and simple training; (b) full dataset training for ensemble units; (c) dynamic adjustment of network structure complexity; and (d) feasibility of parallel training for ensemble units. These features largely avoid the aforementioned problems, making our algorithm superior in

performance and applicability.

2 RANEKF-PEL Method

2.1 Network Model

The RANEKF-based ensemble learning method proposed in this paper is illustrated in [Figure 1: see original paper]. The input sample is an M-dimensional vector $\mathbf{x}$ $\hat{}M$, and the input weight $\mathbf{w}\_\text{in}\hat{}(i)$ $\hat{}M$ for the i-th (1 i p) RAN classifier is randomly initialized and remains fixed during subsequent training. The framework' s final output $\mathbf{y}$ $\hat{}L$ is generated by voting among the outputs $\mathbf{y}\hat{}(i)$ $\hat{}L$ (1 i p) from each RAN classifier in the ensemble.

[Figure 1: see original paper] Parallel ensemble learning framework using RAN as weak classifiers

[Figure 2: see original paper] Structure diagram of Resource Allocating Network (RAN)

The three-layer RAN neural network shown in [Figure 2: see original paper] has input and output dimensions of M and L, respectively. The number of hidden layer nodes N may increase during training as input sample novelty emerges. For a training set of T input samples $X = \{\mathbf{x}\_1, \mathbf{x}\_2, \cdots, \mathbf{x}\_T\}$ where each sample $\mathbf{x}\_i$ contains M features, i.e., $\mathbf{x}\_i = [x\_i\hat{}(1), x\_i\hat{}(2), \cdots, x\_i\hat{}(M)]\hat{}T$ $\hat{}(M\times1)$, the j-th hidden layer node has center $\mathbf{c}\_j = [c\_j\hat{}(1), c\_j\hat{}(2), \cdots, c\_j\hat{}(M)]\hat{}T$ $\hat{}(M\times1)$ and center width $\_j = [\ \_j\hat{}(1), \ \_j\hat{}(2), \cdots, \ \_j\hat{}(M)]$. The output layer bias is $\mathbf{b} = [b\hat{}(1), b\hat{}(2), \cdots, b\hat{}(L)]$. The network output corresponding to input $\mathbf{x}\_i$ is $\mathbf{y}\_i = [y\_i\hat{}(1), y\_i\hat{}(2), \cdots, y\_i\hat{}(L)]$.

During computation, the state of the j-th hidden layer node is:

$\Phi\_j(\mathbf{x}\_i) = \exp(-\|\mathbf{x}\_i - \mathbf{c}\_j\|^2 \ / \ \_j^2) = \exp(-\ \_\{k=1\}\hat{}M \ (x\_i\hat{}(k) - c\_j\hat{}(k))^2 \ / \ (\ \_j\hat{}(k))^2)$

The output of the l-th output layer node is:

$y\_i\hat{}(l) = f(\mathbf{x}\_i) = b\hat{}(l) + \ \{j=1\}\hat{}N \ w\{lj\} \ \Phi\_j(\mathbf{x}\_i), \ l = 1,2,\cdots,L$

where $w\_\{lj\}$ represents the connection weight between the j-th hidden layer neuron and the l-th output layer node.

For multi-class classification, the softmax algorithm is applied:

$o\_i\hat{}(l) = \exp(y\_i\hat{}(l)) \ / \ \_\{j=1\}\hat{}L \exp(y\_i\hat{}(j)), \ l = 1,2,\cdots,L$

The final classification output is:

$\hat{y}\_i\hat{}(l) = \{\ 1, \text{if } o\_i\hat{}(l) = \max\_\{j=1,2,\cdots,L\} \ o\_i\hat{}(j) \ 0, \text{otherwise}$

2.2 Network Training

Initially, the RANEKF network contains zero hidden layer neurons. During training, each sample pair $<\mathbf{x}\_i, \ d\_i>$ (1 i T, where d_i is the label

of sample **x**_i) is input to the RAN network, which first computes the actual output corresponding to **x**_i. By comparing with the expected label d_i, the network determines whether the sample pair satisfies novelty conditions. If satisfied, a new hidden layer neuron is added to the RAN; otherwise, the RAN' s connection weights, node centers, and center widths are adjusted.

Since this framework employs the Bagging ensemble method where each RANEKF learner can be trained in parallel, the entire training process requires minimal time.

The specific training procedure is:

a) Initialize RAN network parameters.

b) Calculate the nearest neighbor Euclidean distance between sample **x**_i and the current N hidden layer neurons:

dist(**x**_i) = min_{1 j N} $\|$**x**_i − **c**_j$\|$

c) Compute the error between the actual network output and expected label value d_i for sample **x**_i:

err(**x**_i) = |**y**_i − d_i|

d) If the novelty condition is satisfied, i.e., dist(**x**_i) _{dist} AND err(**x**_i) _{err}, add a new node to the RAN' s hidden layer and initialize it; otherwise, adjust network parameters. Here, _{dist} and _{err} are novelty determination thresholds for center distance and output error, respectively.

e) When adding a new hidden node:

Hidden node count: $N \leftarrow N + 1$

New hidden node center: **c**_N = **x**_i

Hidden node center width: ← [; dist(**x**_i)]

Network output weights: **w**_out ← [**w**_out ; err(**x**_i)]

where S is the training set size, ensuring uniform distribution of sample categories in small-scale training sets.

f) When adjusting network parameters, apply the Extended Kalman Filter method. Let:

= [**b**, **w**_out , **c** ^T, , , **w**_out_N, **c**_N^T, _N]

be the vector of all RAN network parameters requiring adjustment. The parameter update formula is:

(i) = (i − 1) + **k**(i) err(i)

where the gain vector **k**(t) is:

k(i) = **P**(i − 1) **B**(i) [R(i) + **B**^T(i) **P**(i − 1) **B**(i)] [1]

Here, $\mathbf{B}(i) = \_ f(\mathbf{x}\_\mathrm{i})$ is the gradient vector of function $f(\mathbf{x}\_\mathrm{i})$ with respect to at $(i-1)$, and $R(i)$ is the measurement noise variance. The covariance matrix $\mathbf{P} \hat{} \{z \times z\}$ $(z = L + N \times (M + L + 1))$ is updated as:

$$\mathbf{P}\_\{z \times z\}(i) = [\mathbf{I}\_\{z \times z\} - \mathbf{k}(i)\ \mathbf{B}\hat{}\mathrm{T}(i)]\ \mathbf{P}(i-1) + q\ \ \mathbf{I}\_\{z \times z\}$$

where $\mathbf{I}\_\{z \times z\}$ is the identity matrix, and positive scalar q determines the allowable random step size on the gradient vector. Note that when a new hidden layer node is added, matrix $\mathbf{P}$ becomes:

$$\mathbf{P}(i) = (\ \mathbf{P}(i-1)\ \mathbf{0}\ \mathbf{0}\ p\ \ \mathbf{I}\_\{z \times z\}\ )$$

where p estimates initial parameter uncertainty, and z is the number of parameters added by a new hidden layer node, i.e., $z = M + L + 1$.

g) Repeat steps b)-f) until $i = T$, i.e., all samples have been trained.

3 Experimental Results

This study uses the MNIST handwritten digit database as the experimental dataset. MNIST contains 60,000 training samples and 10,000 test samples of digits 0–9, with each sample being a 28×28 pixel image. To verify the proposed method' s effectiveness on small datasets, only a tiny fraction of the training dataset is used as the experimental training set.

To demonstrate the proposed method' s superior computational accuracy, we compare it against ensemble learning methods constructed from various weak classifiers, including feed-forward fully-connected neural networks (FCNs), convolutional neural networks (CNNs), deep belief nets (DBN), and sparse autoencoders (SAEs). These classifiers have been successfully and widely applied to MNIST handwritten digit recognition, achieving over 97% classification accuracy on the complete MNIST training set, confirming their strong learning capabilities and comparative value.

We compare classification accuracies of several ensemble networks across different training dataset sizes and different weak learner ensemble scales.

3.1 Training Dataset Scale

We test classification accuracies of several ensemble learning methods with ensemble size 20 on training sets ranging from 50 to 2000 samples, where each handwritten digit category (0–9) has a training sample quantity of:

$$S \times i,\ i = 0,1,2,\cdots,9$$

ensuring uniform distribution of sample categories in small-scale training sets.

Experimental results are shown in [Figure 3: see original paper]. Notably, with extremely small training sample sizes where each category contains only 5 samples, among the five model comparison results, only the proposed RANEKF-PEL achieves accuracy above 50%. As the training dataset increases, the mean

classification accuracies of all ensemble learning methods gradually improve, and accuracy fluctuations gradually decrease. The proposed RANEKF ensemble learning method achieves the highest classification accuracy and smaller accuracy fluctuations under equivalent training dataset sizes, demonstrating high computational stability.

Ensemble size significantly impacts system output accuracy and stability. Typically, smaller ensemble sizes yield lower accuracy and stability.

We compare classification output accuracy and fluctuations when the training dataset size is 100 across different ensemble sizes (1, 3, 5, 7, 10, 12, 15, 18, 20). In [Figure 4: see original paper], the RANEKF-PEL method consistently maintains classification accuracy above 50%, indicating that our proposed method achieves the highest accuracy and very small accuracy fluctuations—i.e., high computational stability—across various ensemble sizes.

[Figure 4: see original paper] Classification accuracy and fluctuation comparison of several ensemble learning methods under different ensemble sizes

3.2 Hidden Node Growth During Incremental Training

As shown in [Figure 5: see original paper], during the initial training stage, the RAN network has very few hidden layer nodes, resulting in poor output accuracy and consequently rapid hidden node addition. As the RAN network scale continuously grows and its computational capability for complex patterns strengthens, fewer new nodes need to be added. When the number of hidden layer nodes reaches between 50 and 70, the network already possesses good computational capability, so the hidden layer node count gradually converges to a fixed value and the network structure gradually reaches a steady state.

[Figure 5: see original paper] Hidden layer node variation during incremental RAN training

4 Conclusion

To achieve high computational accuracy and stability on small-scale training datasets, this paper proposes a parallel ensemble learning method based on Resource Allocating Networks with Extended Kalman Filter (RANEKF-PEL). In this method, the input signal for each Resource Allocating Network (RAN) unit is a weighted signal of various dimensions from the original sample data. Through experimental comparisons with ensemble systems constructed from multiple types of artificial neural network units, the proposed RANEKF-PEL method achieves the highest computational accuracy and robustness after training on small datasets, representing an effective and feasible machine learning model.

References

[1] Chen H Y, Li D C, Lin L S. Extending sample information for small data set prediction [C]// Proc of International Congress on Advanced Applied Informatics. 2016: 710-714.

[2] Li D C, Liu C W. Extending attribute information for small data set classification [J]. IEEE Trans on Knowledge and Data Engineering, 2012, 24(3): 452-464.

[3] Aydemir M S, Bilgin G. Semisupervised hyperspectral image classification using small sample sizes [J]. IEEE Geoscience and Remote Sensing Letters, 2017, 14(5): 621-625.

[4] Meng Jianjun, Sheng Xinjun, Zhang Dingguo, et al. Improved semisupervised adaptation for a small training dataset in the brain-computer interface [J]. IEEE Journal of Biomedical and Health Informatics, 2014, 18(4): 1461-1472.

[5] Da Silva I B V, Adeodato P J L. PCA and Gaussian noise in MLP neural network training improve generalization in problems with small and unbalanced data sets [C]// Proc of IEEE International Joint Conference on Neural Networks. 2011.

[6] Sasikala S, Appavu S, Balamurugan A, et al. An efficient feature selection paradigm using PCA-CFS-Shapley values ensemble applied to small medical data sets [C]// Proc of International Conference on Computing, Communications and Networking Technologies. 2013.

[7] Wang Tian, Chen Yang, Zhang Mengyi, et al. Internal transfer learning for improving performance in human action recognition for small datasets [J]. IEEE Access, 2017, 5: 17627-17633.

[8] Li Zengxi, Song Yan, Mcloughlin I, et al. Compact convolution neural network transfer learning for small-scale image classification [C]// Proc of IEEE International Conference on Acoustics, Speech and Signal Processing.

[9] Rani T S, Soujanya P V. An ensemble method using small training sets for imbalanced data sets: Application to drugs used for kinases [C]// Proc of International Conference on Contemporary Computing. 2013.

[10] Cheng Zezhou, Yang Qingxiong, Sheng. Bing Colorization using neural network ensemble [J]. IEEE Trans on Image Processing, 2017, 26(11):

[11] Zhou Zhihua, Jiang Yuan. NeC4.5: neural ensemble based C4.5 [J]. IEEE Trans on Knowledge on Data Engineering, 2004, 16(6): 770-773.

[12] Yu Shicai, Xia Giurong. Neural network ensemble method based on improved sort learning algorithm [C]// Proc of International Forum on Information Technology and Applications, IEEE, 2010.

[13] Liu Yong. Create weak learners with small neural networks by balanced ensemble learning [C]// Proc of IEEE International Conference on Signal Pro-

cessing, Communications and Computing. 2011.

[14] Lima T P F, Ludermir T B. Differential evolution and meta-learning for dynamic ensemble of neural network classifiers [C]// Proc of IEEE International Joint Conference on Neural Networks. 2015.

[15] Schaal S, Atkeson C G. Constructive incremental learning from only local information [J]. Neural Computation, 1998, 10(8): 2047-2084.

[16] Gao Yaozong, Zhan Yiqiang, Shen Dinggang. Incremental learning with selective memory (ILSM): towards fast prostate localization for image guided radiotherapy [J]. IEEE Trans on Medical Image, 2014, 33(2): 518-

[17] Ristin M, Guillaumin M, Gall J, et al. Incremental learning of random forests for large-scale image classification [J]. IEEE Trans on Pattern Analysis and Machine Intelligence, 2016, 38(3): 490-503.

[18] 杨世忠, 吕剑虹. 基于最小资源分配网络的热工对象辨识 [J]. 热能动力工程, 2007, 22(1): 91-95.

[19] 何晓亮, 宋威, 梁久帧. 基于资源分配网络和语义特征选取的文本分类 [J]. 计算机功课与科学, 2014, 36(2): 340-346.

[20] Mahanand B S, Suresh S, Sundararajan N, et al. Alzheimer' s disease detection using a self-adaptive resource allocation network classifier [C]// Proc of IEEE International Conference on Neural Networks. 2011.

[21] Corradini M L, Fossi V, Giantomassi A, et al. Minimal resource allocating networks for discrete time sliding mode control of robotic manipulators [J]. IEEE Trans on Industrial Informatics, 2012, 8(4): 773-745.

[22] CIabattoni L, Grisostomi M, Loppoliti G, et al. On line solar irradiation forecasting by minimal resource allocating networks [C]// Mediterranean Conference on Control & Automation. 2012.

[23] 周志华, 陈世福. 神经网络集成 [J]. 计算机学报, 2002, 25(1): 1-8.

[24] Lim W S, Rao M V C. Effective sonar target classification via parallel structure of minimal resource allocation network [J]. International Journal of Computational Intelligence, 2008, 18: 1109-1116.

[25] Kidera T, Ozawa S, Abe S. An incremental learning algorithm of ensemble classifier systems [C]// Proc of IEEE International Joint Conference on Neural Networks. 2006.

[26] Ji Genlin, Ling Xiaohan, Cheng Xueyun. Novel distributed intrusion detection method based on neural network ensemble [J]. Journal of Nanjing University of Aeronautics & Astronautics, 2007, 39(2): 231: 235.

Note: Figure translations are in progress. See original paper for figures.