Frontiers in Bioscience Landmark

Front. Biosci. (Landmark Ed) **2022**; 27(3): 078
http://doi.org/10.31083/j.fbl2703078

Original Research

A split-and-merge deep learning approach for phenotype prediction

Wei-Heng Huang[1], Yu-Chung Wei[2,*]

[1]Department of Statistics, College of Business, Feng Chia University, 407802 Taichung, Taiwan
[2]Graduate Institute of Statistics and Information Science, National Changhua University of Education, 500207 Changhua, Taiwan
*Correspondence: weiyuchung@cc.ncue.edu.tw (Yu-Chung Wei)

Academic Editor: Graham Pawelec
Submitted: 6 January 2022 Revised: 8 February 2022 Accepted: 10 February 2022 Published: 4 March 2022

Abstract

Background: Phenotype prediction with genome-wide markers is a critical but difficult problem in biomedical research due to many issues such as nonlinearity of the underlying genetic mapping and high-dimensionality of marker data. When using the deep learning method in the small-*n*-large-*p* data, some serious issues occur such as over-fitting, over-parameterization, and biased prediction. **Methods**: In this study, we propose a split-and-merge deep learning method, named SM-DL method, to learn a neural network on the dimension reduce data by using the split-and-merge technique. **Conclusions**: Numerically, the proposed method has significant performance in phenotype prediction for a simulated example. A real example is used to demonstrate how the proposed method can be applied in practice.

Keywords: deep learning; genomic prediction; high-dimensionality data; machine learning; neural networks

1. Introduction

During the past two decades, the dramatic improvement in data collection and acquisition technologies has enabled scientists to collect a great amount of high-dimensional data, for which the dimension p can be much larger than the sample size n, says small-n-large-p. For example, high-throughput genomic data are highly dimensional relative to their sample size. These data often make prediction models sensitive to noise and false positive associations, which consequently make predicting accurate prognoses difficult. The ability to predict complex traits from marker data is becoming increasingly important in plant breeding and association study [1,2].

There have been numerous studies in the genomic prediction models developed with conventional statistical algorithms, including the traditional hypothesis testing approaches [3,4], hidden Markov models [5,6], regression-based methods [7,8], and some Bayesian algorithms [9,10]. Some approaches used regularization methods to reduce the high-dimensional feature sizes, such as ridge regression best linear unbiased prediction (rrBLUP) [7], genomic relationship best linear unbiased prediction (GBLUP) [8], Bayes-*A*, Bayes-*B*, and Bayes LASSO [9,10]. However, among these different genomic prediction models, there was not frequently observed in variation of prediction accuracy. In addition, these prediction models typically make strong assumptions and perform linear regression analysis. For instance, in the rrBLUP model, the assumption is all the marker effects are normally distributed with a small but non-zero variance and it predicts phenotypes from a linear function of genotype markers [7]. Therefore, there has difficulty capturing complex relationship within genotype, and between genotypes and phenotypes in these genomic prediction models for the highly dimensional marker data.

Deep learning is a recently developed machine learning technique that builds multi-layered neural networks containing a large number of neurons to model complex relationship in big data [11]. Deep learning has emerged as a powerful tool to improve prediction performance over traditional models for speech recognition, image identification and natural language processing [11]. This advanced model has also been adopted in bioinformatics and genomics issues recently [12–15]. Many biologists have successfully applied it to several prediction problems including the gene expression inference [16,17], the functional annotation of genetic variants [18,19], phenotype identification from genetic variations [18–22], the recognition of protein folds [23,24], and the prediction of genome accessibility [25]. Pérez-Enciso and Zingaretti [26] provided a guide for using deep learning for complex trait genomic prediction.

Recently, some phenotype prediction and genomic selection methods adopted the deep learning model, such as DualCNN [27], G2PDeep [28], GenNet [29], and the comparative approach [30]. Ma, *et al*. [31] proposed a deep learning method, called DeepGS, to predict phenotypes from genotypes using a deep convolutional neural network. Unlike conventional statistical models, DeepGS automatically learns complex relationships between genotypes and phenotypes from training data, without pre-defined rules (e.g., normal distribution, non-zero variance) for the variables in the neural network. However, when using the DeepGS method in the small-*n*-large-*p* data, it can lead to over-fitting, over-parameterization, and biased prediction. Hence, a large number of training dataset and a low-dimensional subset data are required in order to overcome

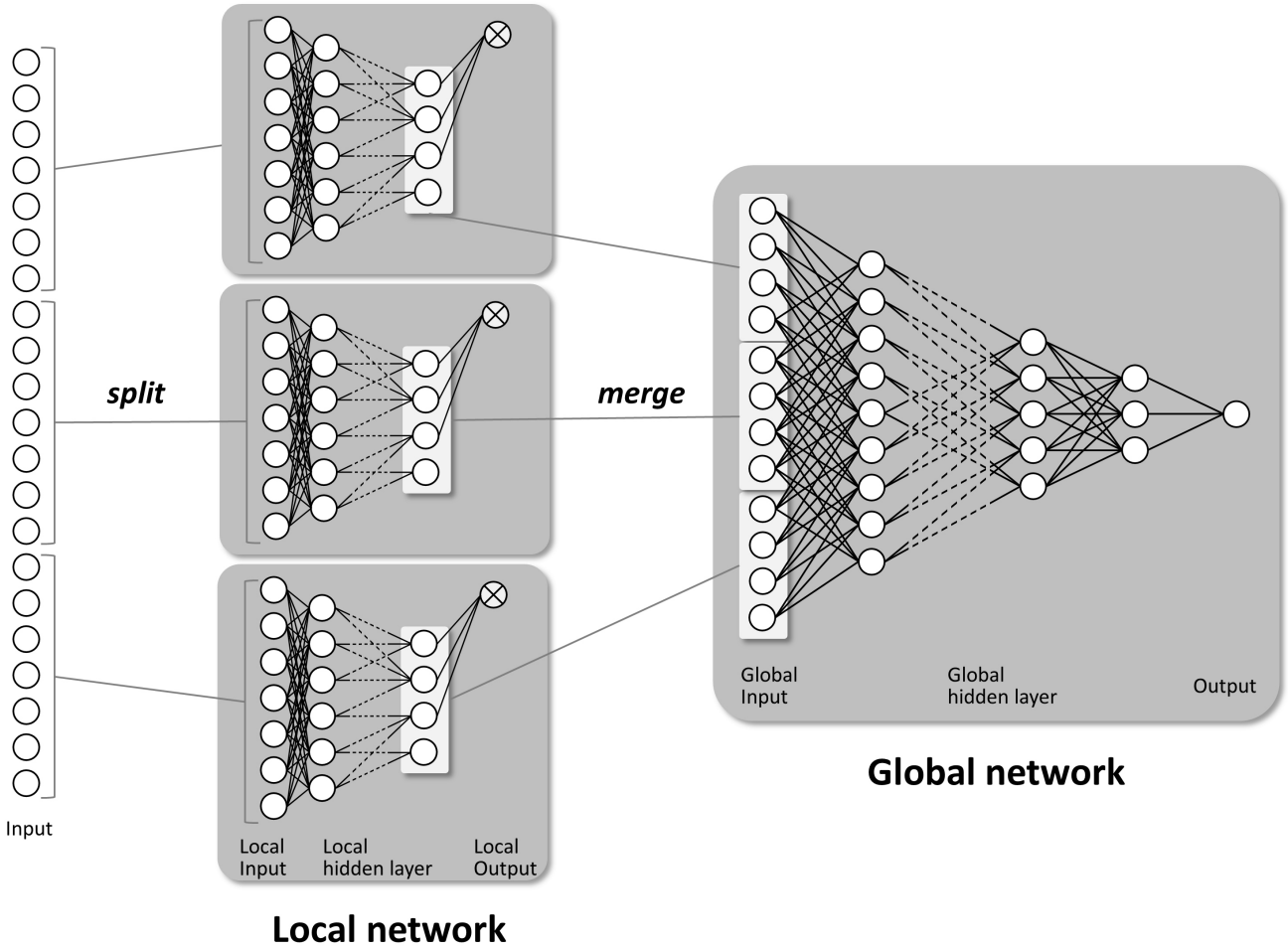

Fig. 1. The architecture of the SM-DL method. The left part presents the high-dimensional features as input; the middle part shows the local neural network; the right part presents the global neural network.

these issues. In this study, a split-and-merge deep learning method, named SM-DL method, is proposed to learn a neural network on the dimension reduced subset data by using the split-and-merge technique on deep learning to obtain nonlinear dimension reduction of the original data.

The rest of the paper is organized as follows. The split-and-merge deep learning method for high dimensional data is discussed in Section 2. Section 3 presents a simulated example. A real data example is given in Section 4 to demonstrate how the proposed method can be applied in practice. Section 5 concludes this paper with brief discussion. Finally, a conclusion is given in Section 6.

2. Split-and-merge deep learning method

In this section, we present a split-and-merge deep learning method, referred to as the SM-DL method, to address the previously mentioned issue, such as over-fitting, over-parameterization, and biased prediction. Based on the SM-DL strategy, the information of the original input features is reduced through deep learning. And then the nonlinear sufficient dimension reduced data is used for the following network algorithms.

To be more precise, the high-dimensional input features are split into several low-dimensional subsets dependent on the dimension of feature. Here, the ways of partition feature are not restricted, which means one can partition randomly or depending on the informative rules, even if the overlapped are also allowed. For simplicity, in this study, we partition features into the non-overlapping subsets. For each low-dimensional subset, a "local" neural network is fitted. The neurons of the last hidden layer for each local network are extracted, named dimension reduced subset data. Finally, all the dimension reduced subset data are merged as the input of the "global" network and then construct a global neural network. Both the local neural network and the global neural network are trained on the training set and validated on the testing set during each fold of cross-validation. We use a ten-fold cross-validations with ten replicates to evaluate the prediction performance of the SM-DL method. If it is needed, the split-and-merge procedure can be repeated in the local network until the dimension reduce sufficiently. Fig. 1 presents the structure of the SM-DL method. The method is summarized in Table 1.

Table 1. SM-DL algorithm.

Algorithm 1
1: Input data $\{(y_i, \boldsymbol{x}_i)\}^n$.
2: Partition the high dimensional dataset to many low-dimensional subsets.
3: For each low dimensional subset, do
◦ Fit a "local" neural network.
◦ Get the dimension reduced subset data by extracting the outputs of the last hidden layer.
4: Combine the dimension reduced subset datasets to form a new dataset, called a dimension reduced subset data.
If the dimension of the dimension reduced subset data is still higher than n, go to step 2.
5: Learn a "global" neural network on the combined dimension reduced dataset.

The structure of both "local" and "global" neural networks via SM-DL strategy are not any strict constraints. We just note that the number of neurons in the last hidden layer of the "local" network is generally set to be smaller than the number of input features in order to serve the purpose of dimension reduction. The source codes of the SM-DL method are available at GitHub (https://github.com/WeiHeng86/SM-DL).

3. A simulated example

In this section, we illustrate the use of the SM-DL method via a simulated example. We mimic the real data which included the markers with biallelic genotype as the input and the continuous phenotype as the output. The framework of the relationship between the input and output was referred to in the previous study [32]. Let Y be a continuous output to mimic the phenotype of an individual and $X_1, \ldots, X_p$ be the discrete variants with values {0, 1, 2} to mimic genotypes of markers. The true dense feed-forward neural network (FNN) with a 2000-500-300-100-1 structure, which includes one input layer, three hidden layers, and one output layer, is determined by

$$\boldsymbol{h}^{(\pounds+1)} = a(\boldsymbol{W}^{(\pounds)}\boldsymbol{h}^{(\pounds)} + \boldsymbol{b}^{(\pounds)}),\ \pounds = 0, 1, 2, 3,$$

where $a(\cdot)$ is an activation function, $\boldsymbol{h}^{(\pounds)}$ denotes an output vector on the $\pounds$-th layer, and $\boldsymbol{W}^{(\pounds)}$ and $\boldsymbol{b}^{(\pounds)}$ are a weight matrix and a bias vector, respectively. Note that the first layer $\boldsymbol{h}^{(0)}$ receives the variants (x's) as input and $\boldsymbol{h}^{(4)}$ is the output layer.

We first generated the 2000 biallelic markers with three genotypes represented as values {0, 1, 2}, and the minor allele frequency of each marker sets as 0.3. For the FNN, the activation function is set to a hyperbolic tangent function, the weights in $\boldsymbol{W}^{(\pounds)}$'s are generated from a Gaussian mixture model, which is $p(w) = \varphi_1 N(w{:}\mu_1, \sigma_1) + \varphi_2 N(w{:}\mu_2, \sigma_2)$ with the parameters $\varphi_1 = \varphi_2 = 0.5$, $\mu_1 = -2$, $\mu_2 = 2$, and $\sigma_1 = \sigma_2 = 1$, and the bias in $\boldsymbol{b}^{(\pounds)}$'s follows a normal distribution with mean 0 and standard deviation 10^{-4}. There are 4000 subjects in the training dataset and 1000 subjects in the test dataset, and there has 2000 variants in each subject.

For SM-DL method, we first split the variants of the training dataset into two non-overlapping subset each with size 1000, and a local network with a 1000-800-700-600-1 structure is fitted to each subset. After merging the output of the last hidden layer from the local network, which consists of 1200 units as the input of a global network, we fit a global network with a 1200-600-500-400-1 structure on the dimension reduced subset data. For all the local and global networks, the Adam method was used as the optimizer by setting the learning rate to 10^{-4} and the L_2-regularization parameter to 10^{-4} for the connection weights. The loss function is set to use mean square error (MSE).

For comparison, an FNN with the true structure 2000-500-300-100-1, referred as True FNN, identical to the data generated structure was used. Moreover, a more complex large network with the structure 2000-1000-500-300-1, referred as Large FNN, was fitted the simulation data. We applied the same learning rate and regularization parameter used in the SM-DL method in the FNNs.

Both Pearson correlation coefficients (PCC) and mean squared error (MSE) were measured the performance of these methods. A ten-fold cross-validation was implemented to evaluate the train performance of each model, and the results were shown in the "Train" columns of Table 2. After getting the optimal model, the predictive performance of testing dataset was shown in the "Test" columns of Table 2. We repeated the same process 10 times, and the average PCC and MSE from the 10 calculations was reported to measure model performance. The results summarized in Table 2. The PCC of the SM-DL method is slightly lower than those of the true FNN and large FNN method; however, the MSE of the SM-DL is lower than those of the other two methods, which shows the SM-DL method has the better prediction performance. This example supports that the last hidden layer of the network retains all the response information contained in the input data.

4. A real data example

In this section, we present a real data example to illustrate how the SM-DL method can be applied to predict phenotype using genome-wide biomarkers. The dataset used consists of 2000 Iranian bread wheat (Triticum aestivum) landrace accessions, each of which was genotyped with 33,709 Diversity Array technology (DArt) markers. For

Table 2. Comparison of performance in the simulated example.

Method	Subset size	PCC		MSE	
		Train	Test	Train	Test
SM-DL	1000	1 (0)	0.2542 (0.0341)	0.0018 (0.0011)	448.8200 (67.1627)
True FNN	all	1 (0)	0.2551 (0.0353)	0.2151 (0.1503)	563.6709 (85.6407)
Large FNN	all	1 (0)	0.2713 (0.0348)	0.3156 (0.2541)	536.3424 (75.6712)

The average Pearson correlation coefficients (PCC) and average mean squared error (MSE) with standard deviation of the estimates listed in parentheses for the SM-DL, True FNN and Large FNN methods. The "Train" presents the train performance of the training dataset and the "Test" indicates the predictive performance of the testing data set by using a ten-fold cross-validation.

Table 3. Comparison of performance in the real data example.

Model	Subset size	PCC		MSE	
		Train	Test	Train	Test
CNN	1000	0.9945 (0.0002)	0.8211 (0.0033)	0.0121 (0.0004)	0.3652 (0.0079)
	2000	0.9949 (0.0000)	0.8225 (0.0013)	0.0095 (0.0001)	0.3588 (0.0030)
	3000	0.9952 (0.0000)	0.8278 (0.0008)	0.0092 (0.0001)	0.3334 (0.0020)
	4000	0.9963 (0.0000)	0.8266 (0.0008)	0.0087 (0.0001)	0.3476 (0.0021)
	all	0.9986 (0.0000)	0.8161 (0.0011)	0.0077 (0.0003)	0.3762 (0.0031)
FNN	1000	0.4943 (0.0446)	0.4145 (0.0411)	0.9921 (0.0146)	1.0375 (0.0028)
	2000	0.4223 (0.0138)	0.3201 (0.0164)	0.9972 (0.0053)	1.0389 (0.0006)
	3000	0.3459 (0.0164)	0.2637 (0.0112)	1.0003 (0.0074)	1.0372 (0.0008)
	4000	0.3348 (0.0132)	0.2726 (0.0129)	0.9932 (0.0045)	1.0374 (0.0005)
	all	0.0336 (0.0175)	0.0524 (0.0312)	0.9911 (0.0000)	1.0383 (0.0000)
rrBLUP	all	1.0000 (0.0000)	0.7524 (0.0068)	0.0000 (0.0000)	0.4152 (0.0136)
GBLUP	all	0.9947 (0.0001)	0.7512 (0.0002)	0.0216 (0.0003)	0.4361 (0.0002)

The average Pearson correlation coefficients (PCC) and average mean squared error (MSE) with standard deviation of the estimates listed in parentheses for the different methods. The "Train" presents the train performance of the training dataset and the "Test" indicates the predictive performance of the testing data set by using a ten-fold cross-validation.

each DArT marker, the allele was encoded by either 1 or 0, to indicate its presence or absence, respectively. The phenotype used as the response variable is grain length (GL). These genotypic and phenotypic data can be downloaded from the International Maize and Wheat Improvement Center (CIMMYT) wheat gene bank (https://www.cimmyt.org/resources/data/). Detailed descriptions for this dataset can be found in [33]. For this dataset, 2000 Iranian bread wheat were divided into a training set with 1600 subjects and a testing set with 400 subjects.

To assess the effect of the data partition, four strategies for the SM-DL method with different split sizes were compared including subset size 1000, 2000, 3000, 4000. Here, the markers were sorted by the location on the genome and then divided into the non-overlapping subset with equal size. The additional strategy without split for total of 33,709 variants was also compared. For the local network of the SM-DL method, two types of neural networks, CNN and FNN, following the DeepGS structures were adopted [31]. The CNN model was trained using the Adam method as the optimizer with the number of epochs of 6000, the learning rate of 0.01, the momentum for moving average of 0.5, and the weight of 10^{-5}. For the FNN model, the parameters were optimized using the stochastic gradient descent (SGD) with the number of epochs of 10,000 and the learning rate of 10^{-4}. For simplicity, the activation function and all hyper-parameters for the local networks were set to the default values given in the R package "DeepGS" (https://github.com/cma2015/DeepGS). For the global network, we use a basic FNN with one hidden layer with size 500, and use the tanh function as the activation and the MSE as the loss function. For training the global network, we use the learning rate of 0.01 and the L2-regularization parameter of 0.02.

Also, two BLUP (best linear unbiased prediction)-based models, which included using ridge regression BLUP (rrBLUP) and genomic relationship BLUP (GBLUP), were constructed by using the R package "rrBLUP" (https://cran.r-project.org/web/packages/rrBLUP) and "BGLR" (https://cran.r-project.org/web/packages/BGLR), respectively.

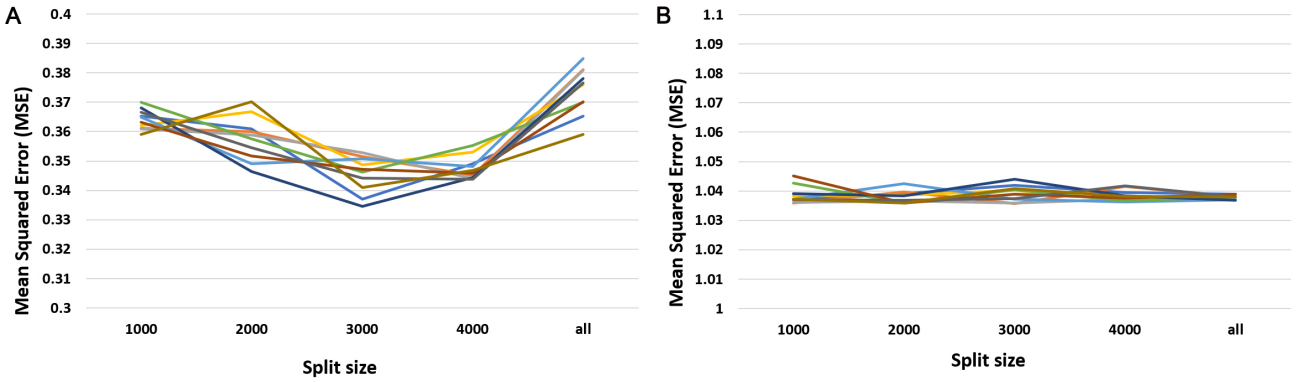

Fig. 2. The mean squared error (MSE) of the testing data set for ten-fold cross-validation, each fold with different colors. (A) CNN model. (B) FNN model.

Table 4. The Wilcoxon signed-rank tests (W statistic) with the corresponding *p*-value between the SM-DL method with the local CNN network and two other regression-based methods.

Methods	PCC		MSE	
	W Statistic	*p* - value	W Statistic	*p* - value
SM-DL (CNN) vs. GBLUP	100	9.234×10^{-5}	100	9.234×10^{-5}
SM-DL (CNN) vs. rrBLUP	100	9.234×10^{-5}	96	3.124×10^{-4}

A ten-fold cross-validation with 10 times was implemented to evaluate the train performance of each model, and the results were shown in the "Train" columns of Table 3. After getting the optimal model, the predictive performance of testing dataset was shown in the "Test" columns of Table 3. The average PCC and average MSE were used as metrics for measuring predictive performance of different models and the results were summarized in Table 3. Fig. 2A and Fig. 2B present the MSE of the testing data set for the CNN and FNN model for the ten-fold cross-validation, respectively. CNN model has a lower mean but a higher variance of MSE than the FNN model. For the model with the local CNN network, the SM-DL method significantly outperforms using all dataset without split in both PCC and MSE. Similarly, for the model with the local FNN network, the SM-DL method has better performance than the FNN model without split structure. However, the rrBLUP and GBLUP model has higher PCC than the SM-DL method with local FNN network. It indicates the architecture of the local networks plays an important role. Overall, the SM-DL method with local CNN network has the best performance among these methods in both PCC and MSE. We can have great improvement with local CNN networks, but not with local FNN networks. For the local FNN networks, we cannot achieve sufficient dimension reduction because the subset data cannot be well approximated. Therefore, it is important for SM-DL method that the local networks should be chosen such that sufficient dimension reduction can be achieved.

Wilcoxon signed-rank test was conducted to statistically assess the performance of the SM-DL method as compared to two other regression-based methods. The SM-DL method with the local CNN network for comparison was the one with the highest PCC value. Table 4 shows the performance of the SM-DL method with the local CNN network is significantly better than other methods based on the 5% significance level (p-value < 0.05). Hence, the outperformance of the SM-DL method with the local CNN network was statistically significant compared to two other regression-based methods.

5. Discussion

A split-and-merge deep learning method to learn a neural network on the dimension reduced subset data has been developed. Two neural networks, CNN and FNN, are applied to the local network. The CNN is regarded as a local connectivity algorithm that can integrate the information of adjacent features. This structure is helpful to take the relative location information of features (genetic markers) into consideration. For this reason, we adopted CNN and FNN models in the local neural network, and the performance of CNN is much better than FNN. However, the input (dimension reduced subset data) of the global network was combined from the local model, and the relative location of these neurons was not meaningful. For this reason, we adopted FNN rather than CNN in the global neural network. It would be worthwhile to apply different structures of deep learning algorithm to the local and global neural network. Moreover, the convolutional layer in CNN is regarded as a kind of dimension reduction strategy. The effect from the split-and-merge algorithm may be partially covered by the convolutional layer. On the contrary, there is no efficient

dimension reduction mechanism in the FNN model, and the advantage of the split-and-merge algorithm is shown apparently. For CNN, the optimal subset size is 3000 that based on the testing results in Table 3. For FNN, using the smaller subset size (says 1000) could help to efficiently reduce the dimension.

Here are few noteworthy perspectives for the SM-DL algorithm. First, the split and-merge procedure was efficient and effective to integrate the information of a great number of inputs such as genomic data. Considered all potential genetic variants into one algorithm to predict phenotype was the optimal circumstance. However, it is almost impossible and time-consuming because of too many genetic variants such as over 88 million identified genetic variants in the human genome [34]. In general procedures, the variants in one chromosome or a small genome fragment were considered at a time, and the variants in the different fragments were seemed independent without further integrated process. It is inappropriate because the mechanism of phenotype general resulted from the variants across several genome regions. The SM-DL algorithm used the split step to partition the genome into small fragments to construct the local network that make the computing more efficient. Also, the merge step effectively integrated the important information retrieved from the last hidden layer in each local network of fragments together to predict phenotype. The information of each fragment transferred from other research could also be adopted as the input in the merge step to make the procedure more efficient. For the split-and-merge deep learning strategy approach, the input variable as a number represents the information of a genetic molecular from a sample. We believe this approach could be applied to other types of genetic variants such as intensity from single nucleotide polymorphism microarray and read depth for copy number variation from sequencing with appropriate data preprocessing.

Second, the SM-DL algorithm could be taken as the parallel ensemble learning algorithm. Two steps for ensemble learning are essential, including several model construction parallelly and combination results of constructed models. For the split step of SM-DL, several local neural networks were built parallelly. Here, the methods of partition features are not restricted even if the overlapped features among local networks. These features have a similar mechanism but in different regions could be overlapped in several local networks. For example, the features/variants were partitioned based on the genetic location first. Then, these variants that belonged to the same gene or pathway were copied to all the corresponding networks as inputs. Alternatively, the randomly selected overlapping features were allowed to decorrelate the features in each local model such as random forest algorithm. In this study, we adopted a simple way to test the proposed SM-DL algorithm in order to focus on the performance of the split-and-merge strategy. It would be worthwhile to study the performance of the SM-DL method with the overlapping features among local networks. Moreover, a systematically integrated method via neural network model was adopted in the merge step of SM-DL. Different weights of the results from each local network can be considered via the network automatically rather than the equal weight such as generally used arithmetic mean and majority vote for ensemble learning algorithms.

Third, SM-DL can be generalized easily to different kinds of neural networks, such as FNN, CNN, and more complex structures. In the SM-DL algorithm, the neurons from the last hidden layer of each network were the only information kept to the next merge-split loop. It mentioned that SM-DL can be applied to any neural network in which information from the last hidden layer can be extracted. It is flexible to use the appropriate network and hyperparameter settings based on your data structure, application issue, and prior information. Through the SM-DL procedure with accurate prediction, the results could be further applied to select top-ranked individuals for animal breeding or find the important genetic variants for medical diagnosis.

6. Conclusions

In this research, we proposed a split-and-merge strategy for deep learning to treat the high-dimensional features problem. A large number of features were reduced to the lower-dimensional data while keeping the information on response contained in the features. In the simulated and real data example, the non-overlapping features among local networks were adopted and the results show the SM-DL method has the better performance. This strategy enhances the predictive performance of deep learning and can be applied to different structures of deep learning algorithms.

Author contributions

W-HH and Y-CW supervised each individual project and performed the entire research together. W-HH analyzed the data and Y-CW presented the results. Both W-HH and Y-CW contributed to write, read, and approved the submitted version of the manuscript.

Ethics approval and consent to participate

Not applicable.

Acknowledgment

The authors are grateful to the reviewers for their insightful comments and suggestions which help improve the presentation of the paper.

Funding

This work was partially supported by the Ministry of Science and Technology, Taiwan (MOST 108-2118-M-035-005-MY3, MOST 109-2118-M-018-005 & MOST 110-2118-M-018-002).

Conflict of interest

The authors declare no conflict of interest.

References

[1] Bhat JA, Ali S, Salgotra RK, Mir ZA, Dutta S, Jadon V, *et al*. Genomic Selection in the Era of next Generation Sequencing for Complex Traits in Plant Breeding. Frontiers in Genetics. 2016; 7: 221.

[2] Bhering LL, Junqueira VS, Peixoto LA, Cruz CD, Laviola BG. Comparison of methods used to identify superior individuals in genomic selection in plant breeding. Genetics and Molecular Research. 2015; 14: 10888–10896.

[3] Cardon LR, Bell JI. Association study designs for complex diseases. Nature Reviews Genetics. 2001; 2: 91–99.

[4] Wei YC, Wen SH, Chen PC, Wang CH, Hsiao CK. A simple Bayesian mixture model with a hybrid procedure for genome-wide association studies. European Journal of Human Genetics. 2010; 18: 942–947.

[5] Cheng Y, Dai JY, Kooperberg C. Group association test using a hidden Markov model. Biostatistics. 2016; 17: 221–234.

[6] Wang P, Zhu W. Replicability analysis in genome-wide association studies via Cartesian hidden Markov models. BMC Bioinformatics. 2019; 20: 146.

[7] Endelman JB. Ridge Regression and other Kernels for Genomic Selection with R Package rrBLUP. The Plant Genome. 2011; 4: 250–255.

[8] VanRaden PM. Efficient methods to compute genomic predictions. Journal of Dairy Science. 2008; 91: 4414–4423.

[9] de los Campos G, Naya H, Gianola D, Crossa J, Legarra A, Manfredi E, *et al*. Predicting Quantitative Traits with Regression Models for Dense Molecular Markers and Pedigree. Genetics. 2009; 182: 375–385.

[10] Meuwissen TH, Hayes BJ, Goddard ME. Prediction of total genetic value using genome-wide dense marker maps. Genetics. 2001; 157: 1819–1829.

[11] LeCun Y, Bengio Y, Hinton G. Deep learning. Nature. 2015; 521: 436–444.

[12] Koumakis L. Deep learning models in genomics; are we there yet? Computational and Structural Biotechnology Journal. 2020; 18: 1466–1473.

[13] Min S, Lee B, Yoon S. Deep learning in bioinformatics. Briefings in Bioinformatics. 2017; 18: 851–869.

[14] Piccialli F, Somma VD, Giampaolo F, Cuomo S, Fortino G. A survey on deep learning in medicine: why, how and when? Information Fusion. 2021; 66: 111–137.

[15] Schmidt B, Hildebrandt A. Deep learning in next-generation sequencing. Drug Discovery Today. 2021; 26: 173–180.

[16] Chen Y, Li Y, Narayan R, Subramanian A, Xie X. Gene expression inference with deep learning. Bioinformatics. 2017; 32: 1832–1839.

[17] Singh R, Lanchantin J, Robins G, Qi Y. DeepChrome: deep-learning for predicting gene expression from histone modifications. Bioinformatics. 2016; 32: i639–i648.

[18] Quang D, Chen Y, Xie X. DANN: a deep learning approach for annotating the pathogenicity of genetic variants. Bioinformatics. 2015; 31: 761–763.

[19] Quang D, Xie X. DanQ: a hybrid convolutional and recurrent deep neural network for quantifying the function of DNA sequences. Nucleic Acids Research. 2016; 44: e107.

[20] Chen SY. Predict and visualize the association between small genetic variants and phenotype via deep Learning convolutional neural networks. Master Thesis, National Changhua University of Education. 2021.

[21] Su SY. Using the convolution neural network to predict and visualize the association between structural variations and binary phenotypes. Master Thesis, National Changhua University of Education. 2021.

[22] Liu Y, Qu H, Chang X, Nguyen K, Qu J, Tian L, *et al*. Deep learning prediction of attention-deficit hyperactivity disorder in African Americans by copy number variation. Experimental Biology and Medicine. 2021; 246: 2317–2323.

[23] Jo T, Hou J, Eickholt J, Cheng J. Improving Protein Fold Recognition by Deep Learning Networks. Scientific Reports. 2015; 5: 17573.

[24] Wang S, Peng J, Ma J, Xu J. Protein Secondary Structure Prediction Using Deep Convolutional Neural Fields. Scientific Reports. 2016; 6: 18962.

[25] Kelley DR, Snoek J, Rinn JL. Basset: learning the regulatory code of the accessible genome with deep convolutional neural networks. Genome Research. 2016; 26: 990–999.

[26] Pérez-Enciso M, Zingaretti LM. A guide on deep learning for complex trait genomic prediction. Genes. 2019; 10: 553.

[27] Liu Y, Wang D, He F, Wang J, Joshi T, Xu D. Phenotype prediction and genome-wide association study using deep convolutional neural network of soybean. Frontiers in Genetics. 2019; 10: 1091.

[28] Zeng S, Mao Z, Ren Y, Wang D, Xu D, Joshi T. G2PDeep: a web-based deep-learning framework for quantitative phenotype prediction and discovery of genomic markers. Nucleic Acids Research. 2021; 49: W228–W236.

[29] van Hilten A, Kushner SA, Kayser M, Ikram MA, Adams HHH, Klaver CCW, *et al*. GenNet framework: interpretable deep learning for predicting phenotypes from genetic data. Communications Biology. 2021; 4: 1–9.

[30] Sandhu KS, Lozada DN, Zhang Z, Pumphrey MO, Carter AH. Deep learning for predicting complex traits in spring wheat breeding program. Frontiers in Plant Science. 2021; 11: 2084.

[31] Ma W, Qiu Z, Song J, Li J, Cheng Q, Zhai J, *et al*. A deep convolutional neural network approach for predicting phenotypes from genotypes. Planta. 2018; 248: 1307–1318.

[32] Liang S, Huang WH, Liang F. Sufficient Dimension Reduction with Deep Neural Networks for Phenotype Prediction. Proceedings of the 3rd International Conference on Statistics: Theory and Applications (ICSTA'21), 2021; 134.

[33] Crossa J, Pérez-Rodríguez P, Cuevas J, Montesinos-López O, Jarquín D, de los Campos G, *et al*. Genomic Selection in Plant Breeding: Methods, Models, and Perspectives. Trends in Plant Science. 2017; 22: 961–975.

[34] The 1000 Genomes Project Consortium. A global reference for human genetic variation. Nature. 2015; 526: 68.