Proceedings of the 3rd International Conference on Statistics: Theory and Applications (ICSTA'21)
~~Prague, Czech Republic~~ Virtual Conference – August, 2021
Paper No. 134
DOI: 10.11159/icsta21.134

Sufficient Dimension Reduction with Deep Neural Networks for Phenotype Prediction

Siqi Liang[1], Wei-Heng Huang[2], Faming Liang[1]
[1] Department of Statistics, Purdue University
610 Purdue Mall, West Lafayette, USA
liang257@purdue.edu; fmliang@purdue.edu
[2] Department of Statistics, Feng Chia University
100 Wenhwa Road, Taichung, Taiwan
weihhuang@mail.fcu.edu.tw

Abstract **-** Phenotype prediction with genome-wide SNPs or biomarkers is a difficult problem in biomedical research due to many issues, such as nonlinearity of the underlying genetic mapping, high-dimensionality of SNP data, and insufficiency of training samples. To tackle this difficulty, we propose a split-and-merge deep neural network (SM-DNN) method, which employs the split-and-merge technique on deep neural networks to obtain nonlinear sufficient dimension reduction of the input data and then learn a deep neural network on the dimension reduced data. We show that the DNN-based dimension reduction is sufficient, which retains all information on response contained in the explanatory data. Our numerical experiments indicate that the SM-DNN method can lead to significant improvement in phenotype prediction for a variety of real data examples. In particular, with only rare variants, we achieved a remarkable prediction accuracy of over 74% for the Early-Onset Myocardial Infarction (EOMI) exome sequence data.

Keywords**:** deep learning, data integration, split and merge, high-dimensional data

1. Introduction

"Understanding the rules of life: predicting phenotype" is one of the big ten ideas that National Science Foundation (NSF) is to invest for the future. The goal of this idea is to elucidate the sets of rules that predict an organism's observable characteristics, i.e., its phenotype. Phenotype prediction promotes health managements for human and genomics selection in animal and plant populations in the future.

The modern biological research shows that single nucleotide polymorphisms (SNPs) might hold the promise of accurately predicting the phenotype of complex traits. Over the last decade, genome-wide association studies (GWAS) have successfully identified a large number of SNPs associated with complex traits [7]. However, phenotype prediction with SNPs has not been very successful. The variants identified from GWAS explain only a small fraction (10-30%) of the total genetic variation [17]. To address the low predictive power of GWAS- derived variants for phenotype, the so-called "missing" heritability, the analysis of many SNPs' joint effects is gaining attention, as each SNP may account for only a very low portion of phenotypic variance [2], but a large number of SNPs may jointly capture a high proportion. However, modeling genome-wide SNPs with standard analysis tools is either impossible or extremely inefficient. The major difficulty comes from the small-n-large-p nature of the problem, for which the number of SNPs (denoted by p in convention) is typically much larger than the number of subjects (denoted by n in convention). The standard statistical tool for dealing with small-n-large-p problems is variable selection by assuming that the underlying model is sparse. However, as mentioned previously, the sparsity assumption does not hold for the SNP-based phenotype prediction problem. To learn a dense model for the problem, the deep neural network (DNN) is a promising tool. The DNN has proven to exhibit better prediction performance over traditional models for many complex problems, such as computer vision and natural language processing. Recently, it has been applied to some biological problems such as inference of gene expression [3] and phenotype prediction in genomics [19]. However, for such a small- n-large-p problem, the DNN model is easy to be over-parameterized. As the consequence, the model tends to be over fitted and the predication can be severely biased. To make the DNN work well, it generally requires a large number of training samples and a relatively low dimension of the data.

To tackle this difficulty, we propose a split-and-merge deep neural network (SM-DNN) method, which is to use the split-and-merge technique on DNNs to obtain nonlinear dimension reduction of the original data, and then learn a DNN on the dimension reduced data. We show that the DNN-based dimension reduction is sufficient, which retains all the response information contained in the explanatory data. Our numerical experiments indicate that the SM-DNN method can lead to significant improvement in phenotype prediction for a variety of simulated and real data examples. The SM-DNN has a great potential for high-dimensional data modeling with deep neural networks and can be used as a tool for integration of multi-omics data.

2. Sufficient Dimension Reduction with DNNs

2.1. Sufficient Dimension Reduction

There is a rigorous statistical theory for sufficient dimension reduction (SDR) under the framework of regression analysis [14, 4]. Let $\boldsymbol{Y} \in \mathbb{R}^d$ be the response variable and $\boldsymbol{X} = (\boldsymbol{X_1}, \boldsymbol{X_2}, \cdots, \boldsymbol{X_p})^{\boldsymbol{T}} \in \mathbb{R}^{\boldsymbol{p}}$ be the explanatory variable with dimension p. The goal of SDR is to find a lower-dimensional representation of X satisfying the condition:

$$p(\boldsymbol{y}|\boldsymbol{x}) = p(\boldsymbol{y}|\boldsymbol{z}) \text{ or equivalently } \boldsymbol{Y} \perp\!\!\!\perp \boldsymbol{X}|\boldsymbol{Z}, \tag{1}$$

where $\perp\!\!\!\perp$ denotes conditional independence, $\boldsymbol{Z} = (\mathrm{Z}_1, \mathrm{Z}_2, \cdots, \mathrm{Z}_\mathrm{q})^{\mathrm{T}}$ denotes the dimension-reduced predictor, and $q < p$. Intuitively, condition (1) means that for predicting $\boldsymbol{Y}$, $\boldsymbol{Z}$ contains the same amount of information as $\boldsymbol{X}$. How to perform sufficient dimension reduction has been studied in the literature for both linear and nonlinear regression models.

For linear regression, the problem is to find a few number of linear combinations of $\boldsymbol{X}$ that are sufficient to describe the conditional distribution of $\boldsymbol{Y}$ given $\boldsymbol{X}$, i.e., finding a matrix $\boldsymbol{B} \in \mathbb{R}^{p \times q}$ such that

$$\boldsymbol{Y} \perp\!\!\!\perp \boldsymbol{X}|\boldsymbol{B}^{\boldsymbol{T}}\boldsymbol{X} \tag{2}$$

Towards this goal, [14] proposed the sliced inverse regression (SIR) method, and a variety of methods related to SIR came out afterwards, see e.g. [5]. [22] proposed an alternative method, minimum average variance estimation based on conditional density functions (dMAVE).

For nonlinear regression, the problem is to find a set of nonlinear functions $f_1(\boldsymbol{x}), f_2(\boldsymbol{x}), \cdots, f_k(\boldsymbol{x})$ such that

$$\boldsymbol{Y} \perp\!\!\!\perp \boldsymbol{X}\,|f_1(\boldsymbol{X}), f_2(\boldsymbol{X}), \cdots, f_k(\boldsymbol{X}) \tag{3}$$

Towards this goal, the kernel trick is often used [18, 21], where the variable $\boldsymbol{X}$ is first mapped to a high- dimensional feature space via kernels and then SIR or other linear SDR methods are performed. A drawback of these methods is that they require to compute the eigenvectors of an $n \times n$ matrix, where n denotes the total number of observations. Hence, the method is hard to be applied when n is large. A general theory for nonlinear sufficient dimension reduction is given in [13].

In what follows, we propose a nonlinear SDR method based on DNNs. The proposed method avoids the step of eigenvector computation, and can be applied to the problems where the sample size n is very large.

2.2. Nonlinear SDR with DNNs

This section shows how to utilize DNNs to achieve nonlinear sufficient dimension reduction. For simplicity, we consider only the cases that the response variable $\boldsymbol{Y} \in \mathbb{R}$ is scalar and distributed according to Gaussian, Bernoulli, and multinomial. For all of them, the density function can be expressed by

$$p(y|\boldsymbol{x}) = \exp\{a(\boldsymbol{\theta})y + b(\boldsymbol{\theta}) + c(y)\}, \tag{4}$$

where $a(\cdot)$ and $b(\cdot)$ are continuously differentiable functions of $\boldsymbol{\theta}$, $c(\cdot)$ is a function of y, $a(\cdot)$ has nonzero derivatives, and $\boldsymbol{\theta}$ is called the natural parameter that relates Y to the predictors $\mathbf{X} = (X_1, \dots, X_p)$ via a nonlinear continuous function

$$\boldsymbol{\theta} = g(\mathbf{x}, \mathbf{w}) = g(x_1, \dots, x_p, \mathbf{w}), \tag{5}$$

where $\mathbf{x} = (x_1, \dots, x_p)$, and $\mathbf{w}$ denotes other parameters of the function. The mean function $\mu = E(Y|\mathbf{X}) = -b'(\boldsymbol{\theta})/a'(\boldsymbol{\theta}) = \phi(\boldsymbol{\theta})$, where $\phi(\boldsymbol{\theta})$ is the inverse of a chosen link function. It is known that if Y is Gaussian, binary, and multinomial, the corresponding link function is linear, logit, and softmax. A dispersion parameter can also be included in [4], which can then include the Gaussian model with unknown variance.

Suppose that n independent random samples, $\{(\mathbf{X}_1, Y_1), \dots, (\mathbf{X}_n, Y_n)\}$, have been drawn from the nonlinear model as specified in [4] and [5]. For this model, if we interpret $\boldsymbol{\theta}$ as the "data" and y as the "parameter", and assume that $a(\boldsymbol{\theta})$ is an one-to-one function and the function $g(\cdot)$ is fully specified, then, by the property of exponential family distributions, $(\boldsymbol{\theta}_1, \dots, \boldsymbol{\theta}_n)$ is sufficient for inference of y, where $\boldsymbol{\theta}_i = g(\mathbf{X}_i, \mathbf{w})$ for $i = 1,2, \dots, n$. This inverse regression-based argument encompasses Fisher's classical definition of sufficiency, and has been used in [1]. Alternatively, we can express the generalized linear models as

$$y = h(g(\mathbf{x}, \mathbf{w}), \mathbf{u}), \tag{6}$$

where $\mathbf{u}$ is a random variable independent of $\mathbf{x}$, $\mathbf{w}$ are parameters, and $h(\cdot,\cdot)$ is a 1-1 mapping. For such a model, it is obvious that $Y \perp\!\!\!\perp \mathbf{X} | g(\mathbf{X}, \mathbf{w})$ and thus $g(\mathbf{X}, \mathbf{w})$ is a sufficient reduction of $\mathbf{X}$ for Y.

To estimate the function $g(\cdot)$, we employ a DNN by taking advantage of its universal approximation capability. The universal approximation capability of neural networks have been studied by multiple authors [6, 9, 11, 15]. In particular, [15] showed that a deep neural network, with appropriately bounded width and growing depth, can approximate any Lebesgue integrable function on d-dimensional input space with respect to L_1-distance. To be more precise, we approximate the function $g(\mathbf{x}, \mathbf{w})$ using a deep neural network by

$$g(\mathbf{x}, \mathbf{w}) = \psi_o \circ \psi_H \circ \cdots \circ \psi_1(\mathbf{x}, \mathbf{w}), \tag{7}$$

where $\mathbf{w}$ becomes the bias and weights of the deep neural network, $\circ$ denotes function composition, ψ_o denotes the activation function of the output layer, and ψ_i denotes the activation function of the i-th hidden layer for $i = 1,2, \dots, H$. Let $f_i(\cdot)$, $i = 1,2, \dots, k_H$ denote the output functions of the hidden layer H, where k_H is the number of hidden units at the hidden layer H. Then $g(\mathbf{x}, \mathbf{w})$ can be re-written as

$$g(\mathbf{x}, \mathbf{w}) = \psi_o(f_1(\mathbf{x}, \mathbf{w}), \dots, f_{k_H}(\mathbf{x}, \mathbf{w})). \tag{8}$$

In summary of the above arguments, we have the following proposition.

Proposition 1. *Consider a nonlinear Gaussian, logistic or multiclass logistic regression model as specified in (4) and (5). Suppose that $\boldsymbol{\theta}$ can be approximated by a deep neural network as specified in (7) and (8) when the sample size n becomes large. Then $\{f_1(\boldsymbol{x}, \boldsymbol{w}), \dots, f_{k_H}(\boldsymbol{x}, \boldsymbol{w})\}$ asymptotically satisfies the SDR condition (3).*

The proof simply follows the above arguments and the properties of the activation functions of the deep neural network, and is thus omitted. Note that all the activation functions used in the DNN, such as tanh, ReLu and softmax, are one-to-one continuous mappings. This implies that the output of any hidden layer satisfies the SDR condition [1], which is consistent with the Markovian structure of feed-forward DNNs.

3. Split-and-Merge DNNs for High-Dimensional Data

As mentioned previously, a direct application of the DNN to small-n-large-p data can lead to many serious issues, such as over-parameterization, over-fitting, and biased prediction. To address these issues, we propose the so-called split-and-merge DNN (SM-DNN) method. The basic idea of the method is to use the split-and-merge technique on DNNs to obtain nonlinear sufficient dimension reduction of the original input data, and then learn a DNN on the dimension reduced data. To be more precise, we first split the high-dimensional dataset into many low-dimensional subsets along the dimension of predictors. For each subset, we fit a DNN, which we call a "local" network, and extract the outputs of the last hidden layer of the "local" network to get dimension reduced subset data, which are called 'SDR subset data'. Then we combine the SDR subset data and repeat this procedure until the dimension of the combined dataset is manageable or it cannot be reduced further. Finally, we train a "global" network on the dimension reduced data. Figure 1 depicts the architecture of the SM-DNN network.

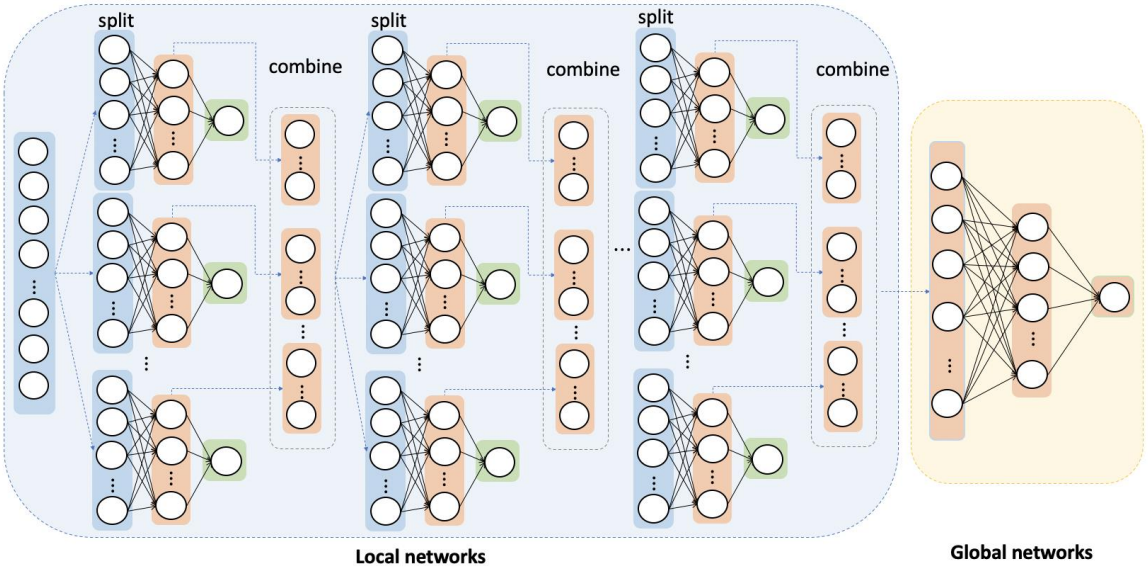

Fig. 1: The architecture of SM-DNN.

To justify the SM-DNN method, we assume that for each subset data, the number of features has been large enough such that each $\mathbf{x}_i$, for $i = 1,2,\ldots,n$, is identifiable based on only the features included in the subset data. That is, for any distinct pair (i,j), if $\mathbf{x}_i \neq \mathbf{x}_j$, then $\tilde{\mathbf{x}}_i \neq \tilde{\mathbf{x}}_j$, where $\tilde{\mathbf{x}}_i$ denotes a subvector of $\mathbf{x}_i$. Under this assumption, by the universal approximation capability of the DNN, one function $g'(\cdot,\cdot)$ can be learned for each subset data such that the nonlinear model (4) still holds. That is, in the form of [6], we have

$$y = h(g'(\tilde{\mathbf{x}}, \tilde{\mathbf{w}}), \mathbf{u}'), \tag{9}$$

for each subset model, where $(\tilde{\mathbf{x}}, y)$ denotes the subset data, $\tilde{\mathbf{w}}$ denotes the weights of the subset DNN model, and $\mathbf{u}'$ is random variable independent of $g'(\tilde{\mathbf{x}}, \tilde{\mathbf{w}})$. Note that existence of the model (9) is obvious. By construction, for each subset data, one can first sufficiently approximate $\tilde{\mathbf{x}}$ (viewing part of the network as an autoencoder) as well as the missed features, i.e., those in $\mathbf{x}$ but not in $\tilde{\mathbf{x}}$, at a hidden layer of the DNN and then continue to approximate the function $g(\cdot,\cdot)$ as with the full dataset. Following from Proposition 1, the SDR subset data has retained all y-related information contained in the subset data. Therefore, SM-DNN does not lead to any loss of y-related information contained in the predictors $\mathbf{X}$.

Proposition 2. *Assume the conditions of Proposition 1 hold. If we further assume that for each subset data, all* $\boldsymbol{x}_i$*'s for* $i = 1,2,\dots,n$ *are identifiable based on only the features of the subset data, then a model (9) can be learned by the DNN for each subset data as* $n \to \infty$*, and the combined SDR subset data retains all response information contained in the predictors.*

To ensure the identifiability condition required in Proposition 2 to hold, overlaps between different subsets are allowed. Note that SM-DNN does not put any strict constraints on the structure of the "local" and "global" networks. For the "local" network, in order to serve the purpose of dimension reduction, the number of hidden units of the last hidden layer is generally set to be smaller than the number of input units. If, for some subset, it cannot be well approximated by such a "local" network, we might consider to increase the depth or even the width of the "local" network until a good approximation of the data can be achieved.

4. Numerical Examples

4.1. A Simulated Example

Suppose that there is a binary outcome Y mimicking the phenotype of an individual, and p discrete variants $X_1, \dots, X_p$ with values $\{0,1,2\}$ mimicking genetic markers. We generated a dataset from a dense feed-forward neural network (FNN) with the structure 2000-500-300-100-1 and other parameters given by

$$\begin{aligned}
\mathbf{h}_i^{(1)} &= \tanh\left(\mathbf{W}^{(1)}\mathbf{x}_i + \mathbf{b}_i^{(1)}\right), \\
\mathbf{h}_i^{(2)} &= \tanh\left(\mathbf{W}^{(2)}\mathbf{h}_i^{(1)} + \mathbf{b}_i^{(2)}\right), \\
\mathbf{h}_i^{(3)} &= \tanh(\mathbf{W}^{(3)}\mathbf{h}_i^{(2)} + \mathbf{b}_i^{(3)}), \\
y_i &= \mathbf{1}_{[0.5,1]}(\sigma(\mathbf{W}^{(4)}\mathbf{h}_i^{(3)} + b_i^{(4)})) \\
b_{ij}^{(l)} &\sim N(0, 10^{-4}), \quad l = 1,2,3,4; i = 1,\dots,n; j = 1,\dots,n^l
\end{aligned}$$

where n^l denotes the number of nodes in l layer, $\tanh(\cdot)$ denotes a hyperbolic tangent function, σ denotes a sigmoid function, $\mathbf{1}_{[0.5,1]}(\cdot)$ denotes an indicator function and $\mathbf{h}^{(l)}$'s denote the outputs of l_{th} hidden layer. The weights in $\mathbf{W}^{(i)}$'s are generated from a Gaussian mixture model $p(w) = \sum_{i=1}^{2} \phi_i N(w: \mu_i, \sigma_i)$ with $\phi_1 = \phi_2 = 0.5$, $\mu_1 = -2$, $\mu_2 = 2$, and $\sigma_1 = \sigma_2 = 1$. The training dataset consists of 4,000 subjects and the test dataset consists of 1,000 subjects, each subject consists of 2,000 variants, and each variant has a minor allele frequency of 0.3. Ten simulated datasets were generated as described above. For both the training and test sets, there are half subjects with positive response.

The SM-DNN method was applied to the 10 simulated datasets. For each of them, we split the training set to 4 subsets and fit each subset by a local network with the structure 500-400-300-300-1, which reduced the dimension of the data to 1,200 after merging the outputs of the last hidden layer of the local networks; then we fit the dimension reduced data by a global network with the structure 1200-600-500-400-1. All the local and global networks were trained using the Adam method with a learning rate of $1e-4$ and a L_2-regularization parameter of $1e-4$ for the connection weights. The hyperparameters were empirically determined via a grid search approach. We measured the performance of the method using both prediction accuracy and AUC. The latter refers to the area under the Receiver Operating Characteristic (ROC) curves. The AUC measures the discriminatory power of the prediction model and can be interpreted as the probability that a randomly chosen positive sample have a higher predicted risk than that of a randomly chosen negative sample.

For comparison, we fit each training set by two FNNs. One is called "true FNN", which has the identical structure as that used for data generation. The other is called "large FNN", which has the structure 2,000- 1,000-500-300-1. The two FNNs were trained using the same learning rate and regularization parameter as those used in the SM-DNN method. Each network was trained for 10,000 epochs. The results were summarized in Table 1. The comparison shows that SM-DNN can achieve similar or even better performance than FNN, which supports our proposition that the last hidden layer of the network retains all the response information contained in the explanatory variables.

Table 1: Accuracy and AUC produced by the FNNs and SM-DNN methods for the simulated classification example, where the numbers in parentheses denote the standard deviation of the estimates. Best results are presented in bold font.

Method	Split	Accuracy		AUC	
		Train	Test	Train	Test
SM-DNN	500	1.0000 (0.0000)	0.5806 (0.0078)	1.0000 (0.0000)	**0.6116 (0.0117)**
True FNN	all	0.9968 (0.0010)	**0.5822 (0.0101)**	0.9984 (0.0006)	0.5959 (0.0118)
Large FNN	all	0.8334 (0.0963)	0.5459 (0.0225)	0.8223 (0.1103)	0.5574 (0.0300)

4.2. A Real Data Example

The Early-Onset Myocardial Infarction (EOMI) exome sequence Data (downloaded from dbGaP) is from the NHLBI Grand Opportunity Exome Sequencing Project (ESP). The dataset consists of 278,263 SNPs in 905 subjects (467 cases and 438 controls) with European origin. After removing the common variants (with minor allele frequency (MAF)> 5%) and the variants with zero MAFs, the number of variants is reduced to 113,438. We used the Variant Effect Prediction tool (VEP) [16] to map these 113,438 SNPs to the genome. This tool gives both the chromosomal location and the known consequences of the variants in the gene sequence (as defined by Sequence Ontology [8]), transcripts, proteins, and regulatory regions (http://www.ensembl. org/info/docs/tools/vep/index.html). We obtained 11,145 genes. Next, the curated human WikiPathways collection was used to retrieve pathway information on these genes [12]. In WikiPathway, genes and metabolites are connected by lines that shows meaningful interaction and/or chemical reactions between entities present in the pathway. We identified 662 different pathways.

To determine the structure of each "local" network, we propose a network architecture called SNP-gene- pathway-associated Neural Network (SNPNet) based on the structure of Pathway-Associated Sparse Deep Neural Network (PASNet) [10]. The SNPNet models a multilayered, hierarchical biological system of SNPs, genes and pathways on a disease, while leveraging the strengths of deep learning for competitive predictive performance. We followed a typical design of conventional DNNs for SNPNet. The cross-entropy was used for the cost function. The tanh was used as the activation function of the hidden layer. The softmax function was used in the output layer so that the probabilities of the output nodes add up to one. The "global" network is a FNN with one hidden layer and 20 hidden units and we use the same cost function and activation function for the hidden and output layer as in the "local" network. For comparison, we also used the convolutional neural network (CNN) as "local" networks in one experiment.

To assess the effect of data partition, we split the 113,438 SNPs into different subset sizes from 3,000 to 20,000 and by chromosome as well. For each subset, the genes were obtained by using VEP to map these SNPs to genome, and then the pathways were retrieved from the curated human WikiPathways collection. We chose 100 units in the hidden layer for each "local" network. Each "local" network is fully-connected and trained for 10,000 epochs. After combining these 100 units of the hidden layer from the "local" network, a "global" network with 20 units in the hidden layer is fitted and trained for 10,000 epochs. For simplicity, all the hyper-parameters were set to be the same for each network. The learning rate was 1e 4 and the L2-regularization parameter was set to 3e 4, which were empirically determined based on the validation set in a pre-experiment. Adam was used as the stochastic optimizer for training these networks. A tenfold cross-validation experiment was performed to evaluate the prediction performance of SM-DNN.

For comparison, we fit a single FNN and a single CNN for the whole dataset with architecture 113,438- 11,145-662-500-1. They were trained using the stochastic gradient descent (SGD) algorithm for 10,000 epochs, where the learning rate was set to 1e 4 as in training other networks. We also trained SVM, which delivers the state-of-the-art performance in a variety of biological applications [20], for comparison. We used the function svm() in the R packages e1071 (https://cran.r-project.org/web/packages/e1071/) to build the linear SVM model and the function ksvm() in the R package kernlab (https://cran.r-project.org/web/packages/kernlab/index.html) to build the kernel SVM model.

We evaluated the performances of these models in terms of prediction accuracy and AUC. The results are summarized in Table 2, which shows that SM-DNN via SNPNet significantly outperforms other benchmark classifiers in both prediction accuracy and AUC. In particular, with only the rare variants, we achieved a remarkable prediction accuracy of 74.14%. With the existing methods, the best prediction accuracy is only 66.24%. It is worth noting that the

architecture of the "local" network is important. For this example, we can achieve very good results with SNPNet but not with CNN. For the latter, the data cannot be even well fitted. Our numerical experience shows that to achieve good phenotype prediction, the local networks must be chosen such that the subset data can be well approximated and thus sufficient dimension reduction can be achieved. It is also interesting to note that the performance of SM-DNN is rather robust to the subset size. As shown in Table 2, although the prediction accuracy varies with the subset size, but is uniformly better than the existing methods. In practice, we may find a suitable data partition based on a validation set.

Table 2: The accuracy and AUC produced by different methods for the EOMI data, where the standard deviations of the estimates are given in the parentheses. Best results are presented in bold font.

Model	Split/Method	Accuracy		AUC	
		Train	Test	Train	Test
SM-DNN (SNPNet)	3000	1.0000 (0.0000)	0.7254 (0.0045)	1.0000 (0.0000)	0.9933 (0.0004)
	5000	1.0000 (0.0000)	0.7378 (0.0052)	1.0000 (0.0000)	0.9961 (0.0003)
	8000	1.0000 (0.0000)	**0.7414 (0.0051)**	1.0000 (0.0000)	0.9947 (0.0003)
	10000	1.0000 (0.0000)	0.7278 (0.0027)	1.0000 (0.0000)	**0.9969 (0.0006)**
	20000	1.0000 (0.0000)	0.7210 (0.0096)	1.0000 (0.0000)	0.9952 (0.0003)
	chrome	1.0000 (0.0000)	0.7155 (0.0019)	1.0000 (0.0000)	0.9940 (0.0004)
FNN	all	0.5628 (0.0043)	0.5232 (0.0094)	0.5892 (0.0063)	0.5385 (0.0079)
SM-DNN (CNN)	3000	0.5310 (0.0000)	0.0037 (0.0000)	0.5167 (0.0057)	0.5234 (0.0108)
	5000	0.5310 (0.0000)	0.0037 (0.0000)	0.5125 (0.0050)	0.4749 (0.0104)
	8000	0.5310 (0.0000)	0.0037 (0.0000)	0.5266 (0.0100)	0.4909 (0.0124)
	10000	0.5121 (0.0000)	0.0037 (0.0000)	0.5053 (0.0051)	0.5048 (0.0084)
	20000	0.5310 (0.0000)	0.0037 (0.0000)	0.5042 (0.0082)	0.4929 (0.0139)
CNN	all	0.5310 (0.0000)	0.0037 (0.0000)	0.5007 (0.0050)	0.4960 (0.0100)
SVM	linear	0.5153 (0.0021)	0.5188 (0.0084)	0.5000 (0.1667)	0.4881 (0.0800)
	kernel	1.0000 (0.0000)	0.6624 (0.0073)	1.0000 (0.0000)	0.7357 (0.0091)

Finally, we note that for the rare variant data, the identifiable condition required by Proposition 2 might be occasionally violated by some subsets. This example indicates that the proposed method is pretty robust to this occasional violation.

5. Discussion

This paper proposed the SM-DNN method to enhance phenotype prediction with genomic data. SM-DNN works based on the theory of sufficient dimension reduction. By using the split-and-merge technique, it successfully reduces the problem of DNN learning for high-dimensional data to a series of problems of DNN learning for low-dimensional data, while retaining all the information on response contained in the predictors.

The proposed method is scalable in both sample size and dimension, which gets around the issue of large sample size by subsampling in training DNNs, and gets around the issue of high dimensionality by the split-and-merge technique. However, for each subset, the number of features should be reasonably large such that the samples are identifiable, rendering the model (9) holds. Regarding the use of SM-DNN, we note that, as mentioned in multiple places of the paper, a good approximation of the local network to each subset data is essential, which ensures sufficiency of dimension reduction.

References

[1] Adragni, K. and Cook, R. (2009), "Sufficient dimension reduction and prediction in regression," *Philos. Trans. A Math. Phys. Eng. Sci.*, 367, 4385–4405.

[2] Balding, D. J. (2006), "A tutorial on statistical methods for population association studies," *Nature reviews genetics*, 7, 781.
[3] Chen, Y., Li, Y., Narayan, R., Subramanian, A., and Xie, X. (2016), "Gene expression inference with deep learning," *Bioinformatics*, 32, 1832–1839.
[4] Cook, R. (1998), *Regression Graphics: Ideas for Studying Regression through Graphics*, John Wiley & Sons.
[5] Cook, R. and Weisberg, S. (1991), "Discussion of 'Sliced inverse regression for dimension reduction,' by K.C. Li." *Journal of the American Statistical Association*, 86, 328–332.
[6] Cybenko, G. (1989), "Approximations by superpositions of sigmoidal functions," *Mathematics of Control, Signals, and Systems*, 2, 303–314.
[7] Donnelly, P. (2008), "Progress and challenges in genome-wide association studies in humans," *Nature*, 456, 728.
[8] Eilbeck, K., Lewis, S. E., Mungall, C. J., Yandell, M., Stein, L., Durbin, R., and Ashburner, M. (2005), "The Sequence Ontology: a tool for the unification of genome annotations," *Genome Biology*, 6, R44.
[9] Hanin, B. (2017), "Universal function approximation by deep neural nets with bounded width and ReLU activations," *ArXiv:*, 1708.02691.
[10] Hao, J., Kim, Y., Kim, T.-K., and Kang, M. (2018), "PASNet: pathway-associated sparse deep neural network for prognosis prediction from high-throughput data," *BMC bioinformatics*, 19, 510.
[11] Hornik, K. (1991), "Approximation Capabilities of Multilayer Feedforward Networks," *Neural networks*, 4, 251–257.
[12] Kutmon, M., Riutta, A., Nunes, N., Hanspers, K., Willighagen, E. L., Bohler, A., Mélius, J., Waagmeester, A., Sinha, S. R., Miller, R., Coort, S. L., Cirillo, E., Smeets, B., Evelo, C. T., and Pico, A. R. (2016), "WikiPathways: capturing the full diversity of pathway knowledge," *Nucleic Acids Research*, 44, D488– D494.
[13] Lee, K.-Y., Li, B., and Chiaromonte, F. (2013), "A general theory for nonlinear sufficient dimension reduction: Formulation and estimation," *The Annals of Statistics*, 41, 221–249.
[14] Li, K.-C. (1991), "Sliced inverse regression for dimension reduction," *Journal of the American Statistical Association*, 86, 316–327.
[15] Lu, Z., Pu, H., Wang, F., Hu, Z., and Wang, L. (2017), "The expressive power of neural networks: A view from the width," in *Advances in neural information processing systems*, pp. 6231–6239.
[16] McLaren, W., Gil, L., Hunt, S. E., Riat, H. S., Ritchie, G. R. S., Thormann, A., Flicek, P., and Cunning- ham, F. (2016), "The Ensembl Variant Effect Predictor," *Genome Biology*, 17, 122.
[17] Mihaescu, R., Meigs, J., Sijbrands, E., and Janssens, A. C. (2011), "Genetic risk profiling for prediction of type 2 diabetes," *PLoS Currents*, 3.
[18] Nilsson, J., Sha, F., and Jordan, M. I. (2007), "Regression on manifolds using kernel dimension reduction," in *Proceedings of the 24th international conference on Machine learning*, ACM, pp. 697–704.
[19] Pouladi, F., Salehinejad, H., and Gilani, A. M. (2015), "Deep Recurrent Neural Networks for Sequential Phenotype Prediction in Genomics," *arXiv preprint arXiv:1511.02554*.
[20] Wei, Z., Wang, K., Qu, H.-Q., Zhang, H., Bradfield, J., Kim, C., Frackleton, E., Hou, C., Glessner, J. T., and Chiavacci, R. (2009), "From disease association to risk assessment: an optimistic view from genome-wide association studies on type 1 diabetes," *PLoS genetics*, 5, e1000678.
[21] Wu, H.-M. (2008), "Kernel sliced inverse regression with applications to classification," *Journal of Com- putational and Graphical Statistics*, 17, 590–610.
[22] Xia, Y. (2007), "A constructive approach to the estimation of dimension reduction directions," *The Annals of Statistics*, 35, 2654–2690.