Prediction of maize phenotype based on whole-genome single nucleotide polymorphisms using deep belief networks

H Rachmatia[*], W A Kusuma, and L S Hasibuan

Department of Computer Science, Bogor Agricultural University, Indonesia

E-mail: hikmat_rac12t@apps.ipb.ac.id

Abstract. Selection in plant breeding could be more effective and more efficient if it is based on genomic data. Genomic selection (GS) is a new approach for plant-breeding selection that exploits genomic data through a mechanism called genomic prediction (GP). Most of GP models used linear methods that ignore effects of interaction among genes and effects of higher order nonlinearities. Deep belief network (DBN), one of the architectural in deep learning methods, is able to model data in high level of abstraction that involves nonlinearities effects of the data. This study implemented DBN for developing a GP model utilizing whole-genome Single Nucleotide Polymorphisms (SNPs) as data for training and testing. The case study was a set of traits in maize. The maize dataset was acquisitioned from CIMMYT's (International Maize and Wheat Improvement Center) Global Maize program. Based on Pearson correlation, DBN is outperformed than other methods, kernel Hilbert space (RKHS) regression, Bayesian LASSO (BL), best linear unbiased predictor (BLUP), in case allegedly non-additive traits.DBN achieves correlation of 0.579 within -1 to 1 range.

1. Introduction

Traditional genetic improvement of plant breeding, using information on phenotypes and pedigrees to predict trait values, has been very successful. However, trait values should be able to predict more accurately by using information on variation in DNA sequence between plants. Marker-assisted selection (MAS) has been used in plant breeding improvement program since the 1990s, after promising research results for tagging genes or mapping quantitative trait loci (QTL). Currently, MAS has failed to significantly improve polygenic traits. While MAS has been effective for the manipulation of large effect alleles with known association to a marker, it has been at a deadlock when many alleles of small effect separate and no substantial, reliable effects can be identified.

The introduction of Genomic Selection (GS) has paved the way to overcome these limitations. GS has shifted paradigm in MAS that seeking to identify individual loci significantly associated with trait into paradigm that uses all marker data as predictors of performance and consequently delivers more accurate predictions. Selection can be based on the output of genomic prediction (GP) models. GP uses a training population of individuals that have been both genotyped and phenotyped to develop a model that takes genotypic data from a candidate population of untested individuals and produces estimated trait values. This approach potentially leading to more rapid and lower cost gains from plant breeding [1].

[*] Corresponding author

Building an appropriated GP models poses several statistical and computational challanges, such as how models can cope with the curse of dimensionality, colinearity between markers, or the complexity of quantitative traits that involves non-additive effects. Linear model in parametric methods (i.e Bayessian LASSO (BL) [2], and best linear unbiased predictors (BLUP) [3]), that are frequently used in developing GP model typically ignore gene by gene interactions, as well as higher order non-linearities [4]. To meet this challange, and to take possible non-linearities into account in prediction, there has been a growing interest in the use of semiparametric and nonparametric methods. In this context, machine learning (as one of nonparametric methods) have been considered to be promising predictive machineries.

Deep learning is a branch of machine learning, currently has been being the most popular topic in area of machine learning research in a last decade. Deep learning is a set of algorithm that attempt to model high level abstractions in data by using a deep graph with multiple linear and non-linear tranformation. This method succeeded to solve perceptual problems such as image recognition [5] and speech recognition [6]. Deep learning has been characterized as a buzzword, because it is just rebranding of artificial neural network.

The study about GP that involve deep learning approach have explored before. Reference [7] constructed deep learning architecture with multi-layer restricted Boltzmann machines (RBM) deep network to disease diagnosis of schizoprenia with single nucleotide polymorpishms (SNPs) molecular marker data as predictors. This method can obtain the average accuray is over 93% and outperformed than other method such as k-nearest neighbours and support vector machines (SVMs). However, that proposed method slightly down in the performance compared with deep belief networks (DBN), which is another deep learning architecture [8].

In this paper, we propose a 4-layer DBN which composed by three RBM to develop GP models. We modified the DBN such that it addresses regression problem instead of classification problem that is usually DBN used to. We used maize dataset from CIMMYT's (International Maize and Wheat Improvement Center) Global Maize program. Reference [9] used the same data to develop GP models with parametric and semiparametric methods. Parametric method consist of BL and BLUP, and then semiparametric method consist of repoducing kernel Hilbert spaces (RKHS) method [10]. To measure the performance of DBN GP models, we compared DBN and these three methods based on Pearson's correlation in the end.

2. Methodology

2.1 SNPs Data and Preprocessing

The maize data set is from the Dought Tolerance for Africa project of CIMMYT's Global Maize Program. The original data set included 300 tropical lines genotyped with 1148 SNPs (referred to as markers). For each marker, the alleles with lowest frequeny was coded as one.

Trait analyzed for this study were grain yield (GY), female flowering (FFL) (or days to silking), male flowering (MFL) (or days to anthesis), and the anthesis-silking interval (ASI), each evaluated under severe drought stress (SS) and well-watered (WW) conditions. Hereinafter we refer to these data sets as maize-grain yield (M-GY) and maize-flowering (M-F), repectively. In all environments, the response variable was standardized to a sample variance equal to one.

The number of lines in the M-F data set was 284, whereas 264 lines were available in M-GY. The average minor allele frquency in these data sets was 0.20. Marker with allele frequency less than 0.05 or greater than 0.95 were removed. Missing genotypes were imputed using samples from the marginal distribution of marker genotypes, that is, $x_{ij} \sim Bernoulli(\hat{p}_j)$, where $\hat{p}_j$ is the estimated allele frequency computed from nonmissing genotypes [9]. After edition, the numers of markers available for analysis were 1148 and 1135 in M-F and M-GY, respectively.

2.2 Deep Belief Networks

Deep belief network (DBN) was introduced by Hinton [8] as a probabilistic generative model which contain multi layers of hidden variable, and these layers detect the higher order of strong corelation between activity of hidden feature in its above layer. State of the art from DBN is applying greedy layer-wise unsupervised learning algorithm to pre-training the initial weights of the networks, and then with global supervised learning for fine-tuning the networks. We used RBM as pre-training algorithm for the DBN.

An RBM is bipartite graph in which visible units that represents observations are connected to binary, stochastic hidden units using undirected weighted connections. They are restricted in the sense that there are no visible-visible or hidden-hidden connections. RBM has an efficient training procedure which makes them suitable as building blocks for DBN.

The input of the first layer in the DBN is the markers data, i.e, the information is coded as -1, 0, or 1. Denote the length of each sample as N_v. In this paper, $N_v = 1148$ for M-F data set and $N_v = 1135$for M-GY data set. In order to obtain more accurate feature selection than generally used softmax RBM for the multi-states sequences, firstly, the marker sequences are changed into binary sequences [7].

We developed a 4-layer DBN which was contained of a visible layer and 3-layer RBM. SNPs data are available from data set that are used as the raw input for the visible layer. Next, each RBM layer can find a feature representation of the current input data, and finally the last layer of the model provides abstract features of the raw features for the regression. The overall approach to building and training a DBN to predict quantitative trait value from markers data is shown in Figure 1.

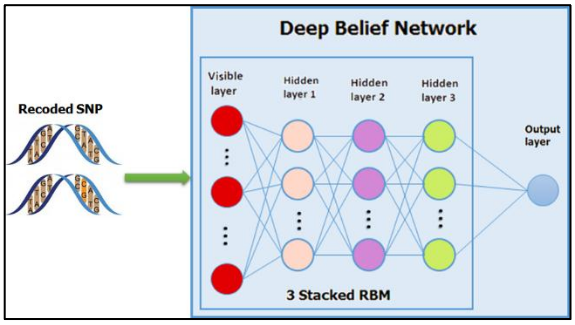

Figure 1. The DBN architecture for genomic prediction

In an RBM, thereare are two layers; one is the layer of visible units, and the other is the layer of hidden units. There are connections between the layer but no connection between units within each layer. In order to get beter performance, we do not choose the general used RBM with softmax and multinomial units, instead we use the Bernoulli RBM as in [7] and define the energy function as follows:

$$E_\theta(v,h) = -v^T W h - \alpha^T v - \beta^T h \tag{1}$$

where the visible units $v = (v^{(-1)}, v^{(0)}, v^{(1)})$, and $v^{(i)} = (v_1^{(i)}, v_2^{(i)}, \ldots, v_{N_v}^{(i)})^T$ $(i = -1, 0, 1)$ is a N_v-dim vector with each $v_j^{(i)} \in \{0, 1\}$. Hidden units denoted as $h = (h_1, h_2, \ldots, h_{N_h})^T$with each $h_k \in \{0, 1\}$. Bias of the visible units is $\alpha = \left(a_1, a_2, \ldots, a_{3*N_v}\right)^T$, bias of the hidden units is $\beta = \left(b_1, b_2, \ldots, b_{N_h}\right)^T$, and $W = \{w_{jk}\}_{4*N_v x N_h}$with each w_{jk} is the connection of v_j and h_k. Here θ is the parameters of the RBM, especially refering to the connection weight matrix W between the visible units v and hidden units h, the bias α of v, and the bias β of h. For every line sample, denote the SNPs sequence as S, i.e, S is a N_v-dim vector and each $S_j \in \{-1, 0, 1\}$. Define $v_j^{(i)}$ $(i = -1, 0, 1;\ j =$

$1, 2, \ldots, N_v)$ as follows:

$$v_j^{(i)} = \begin{cases} 1, & S_j = i \\ 0, & S_j \neq i \end{cases} \tag{2}$$

Thus, we have the joint probability distribution of (v, h) as follows:

$$P_\theta(v,h) = \frac{1}{Z_\theta} exp(-E_\theta(v,h)) \tag{3}$$

where $Z_\theta = \sum_{v,h} exp(-E_\theta(v,h))$ is known as the partition function or normalizing constant.

The distribution of the observed data v, $P_\theta(v)$, is

$$P_\theta(v) = \frac{1}{Z_\theta} \sum_h exp(-E_\theta(v,h)) \tag{4}$$

which has another name, i.e, log-likelihood function. Let $\mathcal{L}(\theta) = -\log P_\theta(v)$. Minimizing the log-likelihood function is equal to determining the parameters to fit the given training samples, i.e, to find a θ^*, such that

$$\theta^* = \arg\underset{\theta}{min}\, \mathcal{L}(\theta) \tag{5}$$

By minimizing $\mathcal{L}(\theta)$, we get the parameters of an RBM, and this can be achieved by performing stochastic steepest descent. The RBM parameters (θ) are updated by

$$\Delta\theta = \epsilon \cdot \frac{\partial \mathcal{L}(\theta)}{\partial \theta} \tag{6}$$

in which, ϵ is the learning rate.

In this study, we use the free energy function in [11] to estimate the gradient of log-likelihood. This function defined as follows:

$$F(v) = \alpha^T v - \sum_j \log(1 + \exp(W_{j\circ}\mathrm{v} + \beta_j)) \tag{7}$$

then the gradient can be written as follows:

$$\frac{\partial \mathcal{L}(\theta)}{\partial \theta} \approx \frac{\partial F(v_{data})}{\partial \theta} - \frac{\partial F(v_{model})}{\partial \theta} \tag{8}$$

Variable v_{data} is an input vector and v_{model} is a reconstructed input that obtained by use contrastive divergence (CD) with Gibbs sampling procedure.

The first step we implement DBN is intialize every weights of hidden layer. The initial values for this weights are uniformly sampled from a symmetric interval that depends on the activation function. We use sigmoid function as an activation, therefore the interval is $[-4\sqrt{6/p_{in} + p_{out}}, 4\sqrt{6/p_{in} + p_{out}}]$, where p_{in} is the number of units in the $(i-1)$-th layer, and

p_{out} is the number of units in the i-th layer. This initilization ensures thta, early in training, each neuron operates in a regime of its activation function where can easily be propagated both upward (activation flowing from inputs to outputs) and backward (gradient flowing from outputs to inputs) [12].

We do some modification to DBN such that the architecture can models the regression problem of genomic prediction. We do not use softmax function as activation function in output layer, instead we use linear function ($f(x) = x$). Besides, we change the basic cost function from logistic loss to mean squared error (MSE) as follows:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2 \tag{9}$$

where n is the number of sample in mini-batch training, $\hat{y}_i$ is the predicted value of sample i by DBN, and y_i is the true value of phenotype from the sample i.

Furthermore, in order to prevent the model from overfitting we add a regularization term to the basic cost function. The regularizer is L2 regularization and also known as weight decay. Hence, our cost function is defined as follows:

$$cost = MSE + \lambda\|\theta\|_p^p \tag{10}$$

where $\|\theta\|_p$ is

$$\|\theta\|_p = \left(\sum_{j=0}^{|\theta|}|\theta_j|^p\right)^{\frac{1}{p}} \tag{11}$$

with $p = 2$ in L2 regularization term. Beside that, we applied early stoping procedure when fine-tuning the model in training process. We used mini-batch stochastic gradient descent optimization in [13] to fine-tune the model.

We implemented all our pre-training and training algorithm concept for DBN by using Theano library for Python programming [14]. Theano is a Python library that can used to define, optimize, and evaluate mathematical expressions, especially ones with multi-dimensional arrays. Using Theano, it is possible to attain speeds rivaling hand-crafted C programming implementations for problems involving large amounts of data.

2.3 SNPs Data and Preprocessing

Prediction of performance of lines whose phenotypes are yet to be observed is a central problem in plant breeding. Such prediction can be used, for example, to decide which of the newly generated lines will be evaluated in field trials. Cross-validation (CV) methods were used to evaluate the ability of model to predict future outcomes. To this end, data were divided into 10 folds; this was done by using an index variable, $I_i \in \{1, \dots, 10\}$, $i = 1, \dots, n$, that randomly assigns observations to 10 disjoint folds, $F_j \in \{i : I_i = j\}, j = 1, \dots, 10$. CV predictions of the observations in the first fold, $F_1 = \{i : I_i = 1\}$, are obtained by omitting phenotypic data on all lines in the first fold. This yields CV predictions of lines in the first fold, that is, $\{\hat{y}_i : I_i = 1\}$. Repeating this task for the second, third, ..., 10th folds fields a whole set of CV predictions $\{\hat{y}_i\}_{i=1}^{n}$ that can be compared with actual observations $\{y_i\}_{i=1}^{n}$ to assess precision. We use Pearson's correlation between phenotypes and their respective predictions from the model as precision measurement. Reference [9] also used the same CV mechanism to obtain precision of the models.

3. Results and Discussions

The DBN that we used to develop GP models are consist of 3 RBM layer for pre-training and 1 output layer for regression, in other words the DBN has 3 hidden layers. Each hidden layer consist of 512 neurons. In order to train the RBM we use CD-k with parameter $k = 1$, and number of epoch is 500.

After pre-training the DBN, we train the DBN with feed forward-backpropagation and use mini-batch stochastic gradient descent as optimization algorithm. In this study, we compare the DBN architecture based on learning rate in fine-tune step. The highest average correlation value we choose from each learning rate variation to represent the precision of the model. Here, we use learning rate 0.01 and 0.001 for training the model.

It can be seen from Figure 2, the best correlation value is obtained from learning rate 0.001 for all data set phenotype category except data set GY-SS that has the best correlation value from learning rate 0.01.

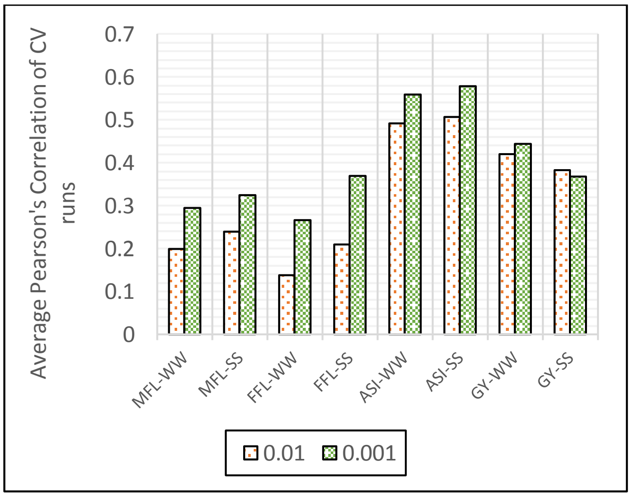

Figure 2. Comparison of precision for two learning rates

Therefore, we have precision of DBN for each phenotype category as follows: 0.295 (MFL-WW), 0.325 (MFL-SS), 0.267 (FFL-WW), 0.370 (FFL-SS), 0.559 (ASI-WW), 0.579 (ASI-SS), 0.445 (GY-WW), and 0.368 (GY-SS). This leads to the last step, the result will be compared with the existing research which is using BL, BLUP, and RKHS methods in [9]. The comparison results can be seen in Table 1.

Table 1. Cross-validation (cv) correlation between predicted and observed phenotypes

Trait–environment	Model[a]			
	RKHS	BL	BLUP	DBN
MFL – WW	0.607	0.790	—[b]	0.295
MFL – SS	0.674	0.778	0.464	0.325
FFL – WW	0.588	0.781	—[b]	0.270
FFL – SS	0.648	0.774	0.521	0.370
ASI – WW	0.547	0.513	0.469	**0.559**
ASI – SS	0.572	0.517	0.481	**0.579**
GY – WW	0.514	0.525	0.515	0.445
GY – SS	0.453	0.415	0.442	0.383

Four models were fitted to each trait (FFL, MFL, ASI, and GY) and environment (SS, severe drought stress; WW, well watered) combination.

[a] Models were molecular marker (SNPs) using reproducing kernel Hilbert space (RKHS) regression, Bayesian LASSO (BL), best linear unbiased predictor (BLUP), and our proposed method deep belief network (DBN).

[b] BLUP were not computed because the estimated genetic variances were negligible [9].

Generally, DBN outperformed than other method and similarly or better than RKHS for anthesis-silking interval trait. DBN get a poor result for female and male flowering traits. This is because DBN is nonparametric method that build non-additive model for GP. Reference [15] provide evidence suggesting that female and male flowering traits in maize are additive traits. Therefore, one could expect the DBN to perform well in traits where epitasis plays a central role. RKHS is semiparametric method that may be able to capture epistatic interactions better than parametric method.Therefore, we can conclude that anthesis-silking trait in maize is a non-additive trait.

4. Conclusion

This research has succesfully modified and implemented DBN method for develop GP models. We also have compared the precision of DBN with other method that mostly used in GS. DBN outperforms than other methods, kernel Hilbert space (RKHS) regression, Bayesian LASSO (BL), best linear unbiased predictor (BLUP),in trait that we expect as non-additive traits. In this study, we have not done hyper-parameter optimization for training the DBN. We belief that DBN performance can be better with hyper-parameter optimization and apply advance method to prevent overfitting in neural network models.

References

[1] Jannink J L, Lorenz A J and Iwata H 2010 *Brief. Funct. Genomics.* **9** 166-77

[2] Campos G D L, Naya H, Gianola D, Crossa J, Legarra A, Manfredi E, Weigel K and Cotes J M. 2009 *Genetics* **182** 375-85.

[3] Meuwissen T H, Hayes B J and Goddard M E 2001 *Genetics* **157** 1819–29.

[4] Ehret A, Hochstuhl D, Gianola D and Thaller G 2015 *Genet. Sel. Evol.* **47** 22.

[5] Ghahramani Z, Welling M, Cortes C, Lawrence N D and Weinberger K Q 2014 *Proc. Advances in Neural Information Processing Systems* (Montreal: NIPS Conference)

[6] Sainath T N, Mohammed A, Kingsbury B and Ramabhadran B 2013 *IEEE International Conference on Acoustics, Speech and Signal Processing* (ICASSP) May 26-31 (Vancouver: IEEE)

[7] Chen Q, Dong-Dong L, Shao-Long C, Yu-Ping W 2015 *Proceedings of the 2015 IEEE International Conference on Bioinformatics and Biomedicine* (BIBM) (USA :Washington DC)

[8] Hinton G E, Osindero S and Teh Y W 2006 *Neural Comput.* **18** 1527-54.

[9] Crossa J, De Los Campos G, Pérez P, Gianola D, Burgueño J, Araus J L, Makumbi D, Singh R P, Dreisigacker S, Yan J, Arief V, Banziger M and Braun H J 2010 *Genetics* **186** 713-24

[10] Gianola D and Kaam J B C H M V 2008 *Genetics* **178** 2289-303

[11] Bengio Y 2009 *FNT in Machine Learning* **2** 1-127.

[12] Glorot X and Bengio Y 2010 *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics* (AISTATS-10) (Brookline: Microtome)

[13] Mu L, Tong Z, Yuqiang C, Smola A J 2014 *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (New York: Association for Computing Machinery)

[14] Theano Development Team 2016 Theano: A Python framework for fast computation of mathematical expressions.; arXiv *e-prints*. abs/1605.02688.

[15] Buckler E S, Holland J B, Bradbury P J, Acharya C B, Brown P J, Browne C *et al* 2009 *Science* **325** 714-8